

ct RASPI

Die Toolbox für Nerds

Hard- und Software ausreizen

GPIO-Pins in Python programmieren
Messen und steuern mit dem Raspi
Raspberry Pi OS und Updates im Griff

Raspi im Wohnzimmer

Spotify und AirPlay abspielen
4K-Filme in HDR gucken

Verstehen und loslegen

Wieso Millionen Menschen Raspis kaufen
Blitzschnell einrichten durch Vorkonfiguration
Betriebssystem aus dem Internet booten

Cooler Projekte mit dem Raspi

PC und Smartphone automatisch sichern, WLAN-Angriffe erkennen,
Passwörter auf dem Raspi-Speicher synchronisieren



€ 14,90

CH CHF 27,90

AT € 16,40

LUX € 17,10



WISSEN SCHÜTZT

heise Security
TOUR

Angriffe auf die IT erfolgreich abwehren

23.06. KÖLN | 28.06. STUTTGART | 05.07. ONLINE

Angesichts eines realen Kriegs gewinnt IT-Sicherheit eine neue Bedeutung. Es geht nicht mehr „nur“ darum, gut organisierte Kriminelle abzuwehren; auch Cyber-Angriffe durch staatliche Akteure werden zur realen Gefahr.

Die heise Security Tour hilft Ihnen dabei, den Überblick zu bewahren, Maßnahmen zu priorisieren und diese in der Praxis richtig umzusetzen.

DIE THEMEN

- Lagebild IT-Security: Auswirkungen des Kriegs, Ransomware, Cybercrime & Co.
- Datenschutz: US-Datentransfer, Cookies, Google Analytics
- Erfahrungen eines CISOs mit Anti-Phishing
- Sichern der Supply Chain
- Sicheres Backup: Immutable Storage, Air Gaps und andere Konzepte
- Die Wahrheit des Patchens

Frühbucherpreis
Köln oder Stuttgart: **549,-**

Frühbucherpreis Online-Konferenz: 449,-
Sie sind Mitglied von heise Security Pro?
Dann ist die Teilnahme für Sie kostenlos.

Jetzt 50 Euro Frühbucher-Rabatt sichern:
heise-security-tour.de



Editorial

Liebe Leserin, lieber Leser,

vor zehn Jahren wurde der Raspberry Pi vorgestellt und seitdem hat sich viel geändert – nur eins nicht: Der Raspi ist immer noch ein kleiner und günstiger Bastelrechner, der für viele Projekte ausreichend Rechenleistung hat. Seine Fangemeinde wächst von Jahr zu Jahr und auch in der Redaktion kommen uns immer wieder neue Ideen, was man mit dem Raspi alles erledigen kann.

Gehören Sie auch zu den Raspi-Fans? Dann dürfen Sie unsere kleine Geschichtsstunde über die Entwicklung vom ersten Raspberry Pi bis zu den heutigen Modellen nicht verpassen; dazu gehören auch zwei aktuelle Ableger des Raspi. Dann geben wir Tipps zur komfortablen Erstinstallation – dazu braucht man nicht mal mehr einen PC, der Raspi holt sich sein Betriebssystem neuerdings einfach selbst aus dem Netz.

Diverse Raspi-Projekte aus diesem Sonderheft vereinfachen den Alltag. So lässt sich der Einplatinencomputer etwa als Server für ein Backup aller PCs im heimischen Netz einrichten, auch Android- und iOS-Smartphones sichern Fotos dort. Oder Sie speichern darauf Ihre Passwörter, statt sie einem anonymen Dienst im Internet anzuvertrauen. Auch im Wohnzimmer macht der Raspi übrigens eine gute Figur, sei es als Grundlage für einen Streaming-Client oder eine Retro-Spielkonsole.

Wer lieber selbst programmiert, findet eine Einführung, wie sich die GPIO-Pins des Raspi einfach in Python ansprechen lassen. Und damit bei all dem die Sicherheit nicht zu kurz kommt, zeigen wir, wie automatische Updates funktionieren. Viel Spaß bei alten und neuen Projekten wünscht

Lutz Labs

Lutz Labs

Inhalt

DAS RASPI-UNIVERSUM VERSTEHEN

Raspi 1 bis Raspi 4, Raspi Zero und seine Ableger: Grundlage aller Projekte ist die Hardware. Wir geben einen Überblick und erklären die Technik unter der Haube.

- 6 10 Jahre Raspberry Pi
- 14 Zahlen, Daten, Fakten zum Raspi
- 18 Test Raspberry Pi Zero 2 W
- 20 Der Mikrocontroller RP2040 im Detail

MIT DEM RASPI LOSLEGEN

Die Einrichtung des Raspi-Betriebssystems ist viel einfacher als eine Windows-Installation. Dazu braucht man nicht mal mehr einen PC mit SD-Kartenleser.

- 26 Raspi superschnell einrichten
- 34 Autonome Linux-Installation
- 38 FAQ: Raspberry Pi

NÜTZLICHES ZUM MITMACHEN

Ein Teil unserer Raspi-Projekte dient der Sicherheit: Der Raspi schützt vor Einbrüchen ins WLAN und als Backup- oder Passwort-Server. Er hilft aber auch beim Streaming oder dem 3D-Druck.

- 42 Raspberry Pi für den Alltag
- 44 Passwortmanager in Eigenregie hosten
- 52 Backups mit Syncthing
- 56 Syncthing-Backup einrichten
- 64 Raspi meldet Angriffe auf Ihr WLAN
- 72 Fernbedienung fürs Game-Streaming
- 78 Octoprint bringt 3D-Drucker ins Netz

HARD- UND SOFTWARE AUSREIZEN

Zur flexiblen Anbindung externer Hardware dienen die GPIO-Pins. Ihre Ansteuerung programmiert man am besten in Python.

- 84 GPIO-Pins mit Python programmieren
- 92 Servos und Sensoren ansteuern
- 102 Automatische Paket-Updates anpassen

RASPI IM WOHNZIMMER

Der Raspi kann nicht nur Nützliches, er hilft auch bei der Unterhaltung. Er spielt Audio-Streams und 4K-Filme ab oder dient als Ersatz für alte Spielkonsolen.

- 106 Retro-Spiele auf dem Raspi Zero 2 zocken
- 110 Raspi als Client für Spotify und AirPlay
- 112 4K-Filme in HDR auf dem Raspi 4

ZUM HEFT

- 3 Editorial
- 101 Impressum



10 Jahre Raspberry Pi

Der Einplatinencomputer Raspberry Pi begeistert Bastler, verbessert die IT-Ausbildung, dient Profis als Entwicklungswerkzeug und gehört zu den prominentesten Linux-Rechnern. Wir blicken auf wichtige Raspi-Meilensteine zurück, nehmen das Erfolgskonzept unter die Lupe und analysieren seine wirtschaftliche Basis.

Von **Christof Windeck**

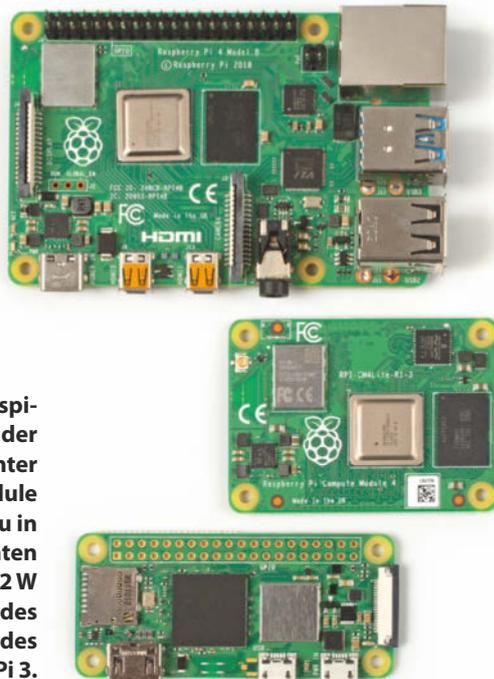


| | |
|--|----|
| 10 Jahre Raspberry Pi | 6 |
| Zahlen, Daten, Fakten: 10 Jahre Raspberry Pi | 14 |
| Test: Raspberry Pi Zero 2 W | 18 |
| Der Microcontroller Raspberry Pi RP2040 | 20 |

Er hat am Schalttag Geburtstag: Am 29. Februar 2012 stellte die Raspberry Pi Foundation ihren ersten Einplatinencomputer vor. Seither wurden mehr als 40 Millionen Raspis gekauft. Alleine 2020 waren es über 7 Millionen, was rund 2,5 Prozent der im selben Jahr verkauften PCs und Notebooks entsprach. Nach Chromebooks sind Raspis wohl die meistverkauften Linux-Rechner. Seit 2012 kamen 18 Raspi-Versionen auf den Markt, die Millionen Bastler inspirierten und eine Horde von Nachahmern auf den Plan riefen. Der kleine Computer hat riesigen Erfolg: Schauen Sie mit uns hinter die Kulissen.

SBC für alle!

Der Raspi ist ein sogenannter Einplatinencomputer mit einem billigen ARM-Chip als Hauptprozessor, auf dem (meistens) Linux läuft. Auf den ersten Blick ähnliche Single Board Computer (SBC) gibt es schon seit Jahrzehnten, aber bis 2012 nur mit Pferdefüßen: Entweder waren sie für professionelle Entwickler gedacht, also teuer und für Privatleute kaum zu beschaffen. Oder sie waren krötenlahm und wegen frickeliger Linux-Unterstützung nur von Nerds benutzbar.



Die aktuellen Raspi-Bauformen: oben der Raspberry Pi 4, darunter das Compute Module CM4 für den Einbau in andere Geräte und unten der Raspberry Pi Zero 2 W mit einer Variante des BCM2837-Prozessors des älteren Raspberry Pi 3.



Bild: Raspberry Pi Foundation/YouTube

Dr. Eben Upton leitet die Raspberry-Pi-Handels-sparte seit ihrer Gründung und spielt auch in der Stiftung eine wichtige Rolle.

Der Raspi traf mit seinen positiven Eigenschaften einen Nerv: billig, ausreichend leistungsfähig, viele Anschlüsse – aber trotzdem nicht zu kompliziert, vor allem dank guter Linux-Unterstützung und Dokumentation. Der Raspi macht Tüftler glücklich, weil er leicht in Betrieb zu nehmen ist; das wiederum motiviert zu Experimenten. Man kann sich auf sein Projekt konzentrieren, statt viel Zeit mit der Konfiguration der Hardwareplattform zu verschwenden.

Erklärtes Vorbild des Raspberry Pi 1 war der „BBC Micro“ aus den 1980er-Jahren, der als Lerncomputer in einer Sendereihe der britischen BBC große Bekanntheit erreichte. Auch der Raspi zielte vor allem auf britische Schüler und Studenten, nämlich als unkomplizierte und bezahlbare Hardwareplattform für die IT-Ausbildung.

Die gemeinnützige Raspberry Pi Foundation steckt viel Arbeit in Software, Dokumentation, (Lehrer-) Schulungen und Support über Online-Foren. In der hauseigenen Linux-Distribution auf Debian-Basis – früher Raspbian genannt, heute Raspberry Pi OS – steckt viel Feinschliff. Kompatibilität hat höchste Priorität: Softwareprojekte laufen ohne Änderungen auf unterschiedlichen Raspi-Varianten. Das seit Jahren weiterentwickelte Betriebssystem läuft stabil genug für manche Industrieanwendungen.

Das Bootmedium ist clever gewählt: Die meisten Raspi-Versionen booten von einer MicroSD-Karte, die man an einem anderen PC (mit Windows, macOS oder Linux) vorbereiten kann. Die präparierte Speicherkarte lässt sich in einen Raspi aus einer anderen Generation oder mit anderer Bauform umstecken.

Dank offenem Betriebssystem und riesiger Entwicklergemeinde eignet sich der Raspi auch für Anwendungen, die viele Jahre lang benutzt werden sollen: Es ist kaum zu befürchten, dass der Support ausläuft, im Unterschied zu vielen sonstigen Smart-Home- und IoT-Geräten. Anders als in aktuellen x86-Rechnern steckt in Raspis zudem sehr wenig proprietäre Firmware. Damit bieten sie wenig Angriffsfläche und Verstecke für Hintertüren – gute Voraussetzungen zur Verarbeitung sensibler Daten.

Sahnehäubchen

Die Raspi-Entwickler haben Maßstäbe gesetzt. Zum großen Erfolg beigetragen hat die 40-polige Pfostensteckerleiste mit konfigurierbaren Allzweckanschlüssen, die beim ersten Raspi noch 26 Kontakte hatte. Dieser simple GPIO-Header (GPIO steht für General Purpose I/O) im gängigen 2,5-Millimeter-Raster ermöglicht es, eine Fülle anderer Hardware wie Sensoren, Displays, Servo- und Schrittmotoren, Schalter, Encoder, Relais und LED-Streifen anzuschließen und unter Linux zu nutzen, wie wir auf Seite 92 ausführlich vorstellen.

Der billige und flexible GPIO-Anschluss entwickelte sich zum Quasistandard. Eine Fülle von Softwareprojekten nutzt ihn und es gibt eine breite Auswahl an aufsteckbaren Erweiterungsplatinen, genannt „Hardware Attached on Top“ (HAT): zum Beispiel für Hi-Fi-Audio oder mit Display, Schrittmotortreibern, Sensoren oder Analog-Digital-Messwandlern. Die 40-polige GPIO-Pinleiste findet sich mittlerweile auch auf vielen konkurrierenden SBCs mit ARM-, RISC-V- und x86-Chips, sogar auf FPGA-Entwicklerboards für Profis.

Der Raspi taugt auch bestens als Open-Source-Netzwerkgerät, etwa als Webcam, Firewall oder WLAN-Hotspot. Praktisch für diese Zwecke ist die optionale Speisemöglichkeit per Power-over-Ethernet (PoE). Eine clevere und finanziell einträgliche Idee der Raspi-Handelssparte war es, kompatibles Zubehör zu verkaufen: Kameras und Touchdisplays mit jeweils passendem (CSI-/DSI-)Flachbandkabel, Tastaturen, Mäuse, Netzteile, Gehäuse. Das erleichtert Laien den Einstieg, weil sie sich mit weniger Problemen herumschlagen müssen. Das Zubehör ist zudem für die Belieferung von Schulen, Unis und anderen Bildungseinrichtungen wichtig.

Raspi-Konkurrenten wie Orange Pi, Banana Pi und Pine A64 mögen für bestimmte Anwendungen besser geeignet oder schlichtweg billiger sein. Doch die Summe seiner guten Eigenschaften macht den Raspi



Der Schulcomputer Raspberry Pi 400 zeigt, dass die Raspi-Macher auch aufs Äußere achten.

zur ersten Wahl als Hardware-Unterbau für IT-Basteleien. Dadurch kommen die verschiedenen Raspi-Varianten auf gewaltige Stückzahlen. Das wiederum verschafft den Raspi-Entwicklern die finanziellen Mittel, ihr Produkt weiter zu verbessern. Bisher schaukelt sich dieser selbstverstärkende Kreislauf immer weiter hoch – bis die Chipkrise aufzog. Aktuell sind viele Raspi-Varianten entweder gar nicht zu bekommen oder nur zu überzogenen Preisen.

Raspi-Varianten

Die Raspberry-Pi-Familie umfasst vier verschiedene Bauformen, nämlich den „normalen“ Raspi mit einer Platine im Bezahlkartenformat (8,5 cm × 5,6 cm), den weniger als halb so großen und billigeren Zero (6,6 cm × 3 cm), das Compute Module (CM) für Industriekunden und den Minicomputer Raspi 400 im Tastaturegehäuse [1]. Im Vergleich zur Standardausführung fehlen dem Zero einige Schnittstellen, um Platz zu sparen, und für die GPIO-Leiste sind nur Lötungen vorhanden – wer Pins benötigt, muss selbst löten. Das CM wiederum hat keine direkt nutzbaren Anschlüsse, sondern ist als Steckmodul zum Einsatz in anderen Geräten gedacht, etwa als „Rechengehirn“ einer Industriesteuerung. Auf einem kompatiblen I/O-Board lässt sich ein CM aber ähnlich wie ein normal großer Raspi nutzen.

Der 2020 eingeführte, eng mit dem Raspi 4 verwandte Raspi 400 ist ein bezahlbarer Schülercomputer und auch als Kit inklusive Netzteil, Maus, HDMI-Kabel, MicroSD-Karte mit vorinstalliertem Linux und gedrucktem Handbuch erhältlich. Anders als die Raspberry Pi Foundation andeutet, ist ein Raspi 400 zwar kein vollwertiger Ersatz für einen Mini-PC, dazu ist der Prozessor schlichtweg zu schlapp. Doch er taugt für einfache Aufgaben wie Textverarbeitung und steuert auch 4K-Monitore an, bei Bedarf gleich zwei davon.

Bis auf den Raspi 400 gibt es die drei anderen Bauformen jeweils in mehreren Generationen. Der Urvater Raspi 1 aus dem Jahr 2012 erschien mit einem ziemlich lahmen System-on-Chip (SoC) Broadcom BCM2835, in dessen einzigem Prozessorkern noch ARMv6-Technik aus dem Jahr 2002 rechnete. Drei Jahre später brachte der Raspi 2 eine Leistungsexplosion: Vier Cortex-A7-Kerne im BCM2836 mit 30 Prozent höherem Takt und jüngerer ARMv7-Architektur. 2016 folgte mit dem Raspi 3 ein kleinerer Schritt zum BCM2837: Dessen vier Cortex-A53-Kerne gehören zur ARMv8-Generation und sind 64-Bit-tauglich. Letzteres bringt allerdings nur in wenigen Spezialfällen Vorteile – erst vor einigen Wochen erschien

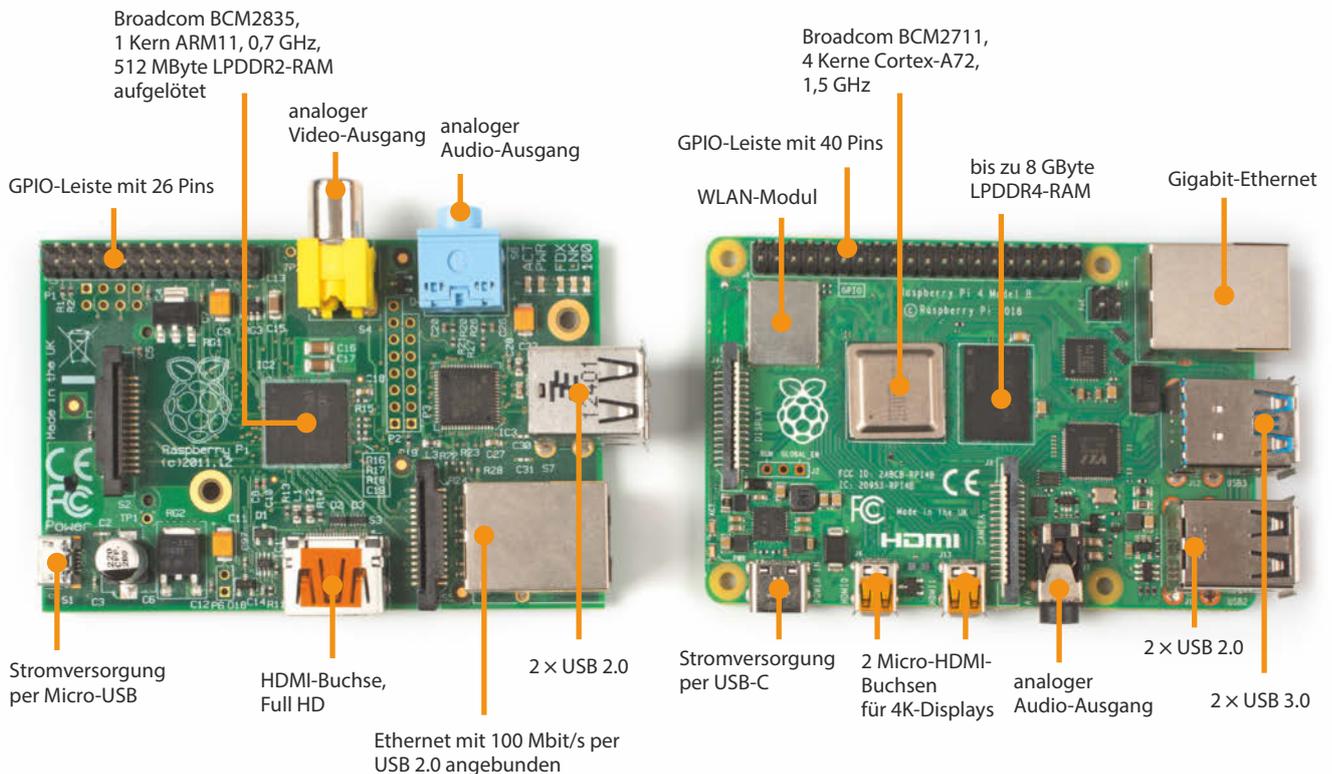
die finale 64-Bit-Version von Raspberry Pi OS. 2019 brachte der Raspi 4 den BCM2711 mit vier deutlich stärkeren Cortex-A72-Kernen und 1,4 GHz Takt (beim jüngeren BCM2711C0 1,8 GHz).

Wichtiger als der Zuwachs an Rechenleistung ist beim BCM2711, dass er endlich mehr als 1 GByte RAM anbindet. Nebenbei ist der Arbeitsspeicher auch noch schneller (LPDDR4 statt LPDDR2, [2]). Den Raspi 4 gibt es mit 2, 4 oder 8 GByte RAM. Damit stemmt er erstaunlich viele Serverdienste, auch in Docker-Container verpackte [3].

Für viele Bastelprojekte genügt locker noch ein Raspi 3 beziehungsweise sein billigerer Verwandter Zero 2 W (siehe Seite 18). Auf letzterem sitzt ein

Generationenvergleich: Raspberry Pi 1 und 4

Zwischen dem Raspberry Pi 1 und dem Raspberry Pi 4 liegen sieben Jahre. Der Blick auf die wichtigsten Komponenten zeigt, was sich verändert hat.



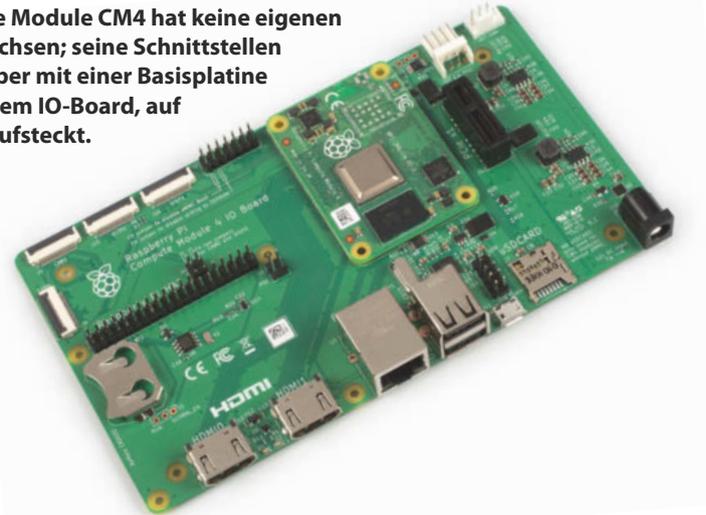
Kombichip, der komprimierte Raspi-3-Technik enthält: den BCM2837B0 sowie einen 512-MByte-Speicherchip. Das erste Raspi-SoC BCM2835 nutzte einen Vorläufer dieser Technik, genannt Package-on-Package (PoP): Dabei sitzt der RAM-Chip in einem eigenen Gehäuse, das oben auf das SoC gelötet ist. Man sieht daher auf dem Raspi 1 nur den Speicherchip, unter dem sich das ARM-SoC verbirgt.

Die Platinen der Compute Modules CM1, CM3 und CM3+ haben die Abmessungen von SODIMM-Speichermodulen. Das CM4 [4] ist noch kompakter und hat andere Steckverbinder. Um Industriekunden anzulocken, sind die CMs nach einschlägigen Standards für Embedded Systems zertifiziert und besonders lange lieferbar; außerdem gibt es sie in Varianten mit aufgelötetem eMMC-Flashspeicher anstelle einer MicroSD-Kartenfassung.

Klare Linie

Die Grundidee des Raspi ist außer vom eingangs erwähnten BBC Micro auch vom sieben Jahre älteren Arduino inspiriert: Der Arduino zeigte, wie sich die zuvor praktisch nur von Profis nutzbaren Mikrocontroller auf einem „Universalsteuerplatinchen“ demokratisieren lassen. Der Raspi-Erfolg beruht ganz wesentlich auf der klaren Linie, die das Team um Dr. Eben Upton seit zehn Jahren hält. Ein Eckpfeiler ist das KISS-Prinzip: „Keep it simple, stupid“ – also Vereinfachung, wo immer möglich. Das hält Kosten und Fehlerquote niedrig, fördert die Kompatibilität und erleichtert es, das Produkt zu verstehen. Bei der Fokussierung auf wenige Gerätevarianten, die sich jedoch für viele Einsatzbereiche eignen und möglichst reibungslos funktionieren, schimmert Apple als Vorbild durch. Vielleicht gilt das auch für die Bemühungen um schöne Gestaltung: das Himbeer-Logo

Das Compute Module CM4 hat keine eigenen Anschlussbuchsen; seine Schnittstellen lassen sich aber mit einer Basisplatine nutzen wie dem IO-Board, auf das man es aufsteckt.



(Raspberry), die himbeerrote Farbgebung von Raspi-Gehäusen, -Mäusen und -Tastaturen, das Layout von Website und Dokumentation.

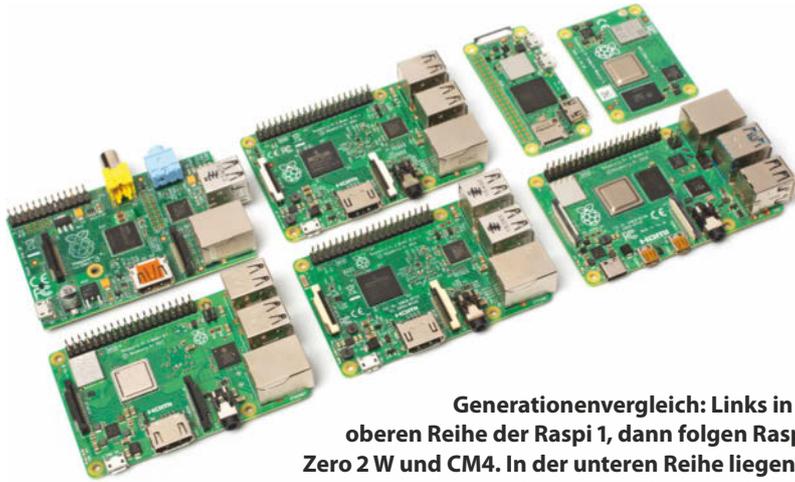
Alle Raspis haben dasselbe Platinenformat und seit dem Model 2 den berühmten GPIO-Pfostenstecker. Das bringt nicht nur Kompatibilität zu vorhandenem Zubehör und Software, sondern auch Wiedererkennungswert. Wichtige Rollen spielen auch Nachhaltigkeit, Fairness und Offenheit: Die Raspi-Platinen werden im Wesentlichen von einer Sony-Fertigungssparte in Wales bestückt, der Heimat von Raspi-Mastermind Eben Upton. Von Anfang an wandten sich Upton und seine Mitstreiter per Blog direkt an die Raspi-Nutzer, Supportforen dienen zur Pflege der Community.

Doch es gibt auch Kritiker, die sich zusätzliche Funktionen wünschen oder mehr Offenheit fordern. So fehlen den Raspi-SoCs bislang (AES-)Rechen-

Systems-on-Chip (SoCs) bisheriger Raspberry Pis

| Broadcom-SoC | ARM-Kerne / Typ | CPU-Takt | Grafik ¹ | max. RAM | Raspi-Typen | Jahr |
|------------------------|--------------------------|-------------|---------------------|----------------------|------------------------------|------|
| BCM2835 | 1 / ARM1176JZF-S (ARMv6) | 0,7–1 GHz | VC IV | 0,5 GByte | 1A(+), 1B, CM1, Zero, Zero W | 2012 |
| BCM2836 | 4 / Cortex-A7 (ARMv7) | 0,9 GHz | VC IV | 1 GByte | 2B | 2015 |
| BCM2837 | 4 / Cortex-A53 (ARMv8) | 1,2 GHz | VC IV | 1 GByte | 3B, CM3, 2B v1.2 | 2016 |
| BCM2837B0 ² | 4 / Cortex-A53 (ARMv8) | 1–1,4 GHz | VC IV | 1 GByte | 3A+, 3B(+), CM3+ | 2018 |
| BCM2710A1 ³ | 4 / Cortex-A53 (ARMv8) | 1 GHz | VC IV | 0,5 GByte | Zero 2 W | 2021 |
| BCM2711 (B0, C0) | 4 / Cortex-A72 (ARMv8) | 1,5–1,8 GHz | VC VI | 8 GByte ⁴ | 4A, CM4, 400 | 2019 |

¹ VC steht für Broadcom VideoCore ² optimierte Variante des BCM2837 mit Metalldeckel für bessere Kühlung ³ Variante des BCM2837B0 mit 512 MByte LPDDR2-RAM im Gehäuse integriert ⁴ LPDDR4, bei allen anderen LPDDR2; der BCM2711 kann theoretisch 16 GByte ansteuern



Generationenvergleich: Links in der oberen Reihe der Raspi 1, dann folgen Raspi 2, Zero 2 W und CM4. In der unteren Reihe liegen die Raspi 3B+, 3B V1.2 und 4B nebeneinander.

werke für schnelle Ver- und Entschlüsselung. Mehr PCI-Express-Lanes wären nötig, um außer USB 3.0 auch SATA oder M.2 anzubinden. Anders als beispielsweise der Arduino ist der Raspberry Pi keine Open-Source-Hardware. Zudem kommt proprietäre Firmware zum Einsatz. Das hat auch damit zu tun, dass Broadcom nicht jedes Detail der SoCs offenlegt und Funktionsblöcke von Dritten zukaufte, beispielsweise die ARM-Kerne.

Stiftung und Kommerz

Der Raspi-Erfolg ruht auf einem geschickt konstruierten wirtschaftlichen Unterbau. Denn es gibt einerseits die Raspberry Pi Foundation als gemeinnützige („charitable“) Stiftung und andererseits die Handelssparte Raspberry Pi Ltd., die vor 2021 Raspberry Pi (Trading) Ltd. hieß. Die Handelssparte übernimmt Verkauf und Entwicklung der Raspis, gehört aber der Stiftung, der sie einen erheblichen Teil ihrer Erlöse für die Finanzierung spendet. 2020 waren es 3 Millionen britische Pfund (GBP), also knapp 3,6 Millionen Euro. Die Summe sinkt von Jahr zu Jahr: 2019 waren es 4 und davor noch 5 Millionen Pfund gewesen. Geschäftsberichte für 2021 liegen noch nicht vor.

Über die Stiftung wacht ein Rat aus 27 ehrenamtlichen Mitgliedern (darunter Eben Upton), der zehn Treuhänder wählt. 111 bezahlte Angestellte plus ehrenamtliche Helfer setzen die gemeinnützigen Ziele der Stiftung um, also vor allem IT-Bildung. 2020 gab die Stiftung rund 9,5 Millionen Pfund aus für Lehr- und Lernmaterial, Schulungen und (digitale) Zeitschriften wie das MagPi Magazine. Die Stiftung

erhält auch Spenden und Fördermittel und konnte ihr Vermögen 2020 um 7,3 auf insgesamt 36,7 Millionen Pfund deutlich steigern.

Die Raspi-Handelssparte – hier kurz Raspi Ltd. genannt – arbeitet hingegen gewinnorientiert und zahlt an leitende Mitarbeiter auch hohe Gehälter, 27 ihrer Spitzenkräfte beziehen zwischen 120.000 und 230.000 Euro jährlich. Sie arbeiten höchst erfolgreich: Der Umsatz verdoppelte sich von 2019 auf 2020 fast auf 71,7 Millionen Pfund (85 Millionen Euro), wovon knapp 15 Prozent beziehungsweise 10,7 Millionen Pfund (13 Millionen Euro) als Bruttoertrag übrigblieben. Der durchschnittliche Ertrag pro Raspi stieg dank der teureren Raspi-4-Versionen um 13 Prozent auf 3,93 Euro. 17 Prozent des Umsatzes entfiel auf 2,9 Millionen verkaufte Zubehörteile wie Netzteile, Mäuse, Kameras und Kabel.

Die Handelssparte hat 73 Mitarbeiter, darunter 10 „Directors“ und 34 Entwickler. 2020 flossen umgerechnet 3,7 Millionen Euro in die Entwicklung, 15 Prozent mehr als 2019 (3,2 Millionen Euro). Die Stiftung hat 2021 ein Lizenzabkommen für nicht näher genannte Prozessortechnik abgeschlossen, vermutlich geht es um die ARM-Kerne des selbst entwickelten Raspberry Pi RP2040 (siehe Seite 20). Die jährlichen Lizenzgebühren dafür betragen rund 1,2 Millionen Euro.

Um weiter wachsen zu können, hat sich die Raspi Ltd. Ende 2021 knapp 40 Millionen Euro von zwei Investoren beschafft. Diese erhielten im Gegenzug Firmenanteile. Wegen dieser Transaktion kamen Spekulationen auf, dass die Raspi Ltd. ihren Börsengang vorbereite, was Eben Upton aber dementierte. Allerdings erhalten auch einige leitende Angestellte Firmenanteile, die treuhänderisch verwahrt werden. Kurzum: Die Raspi Ltd. arbeitet wie ein schnell wachsendes Wirtschaftsunternehmen mit gut bezahlten Fachkräften.

Enge Kontakte

Raspi-Stiftung samt Handelssparte sind in der britischen Universitätsstadt Cambridge ansässig. Dort befindet sich nicht nur der Hauptsitz der Firma ARM, sondern auch eine Niederlassung der US-Firma Broadcom. Für letztere arbeitete Eben Upton von 2006 bis 2020 als „SoC Architect“ und war insbesondere für den Grafikern VideoCore zuständig. Alle Raspis nutzen Broadcom-SoCs mit ARM-Kernen und VideoCore. Die Entwicklung quelloffener Linux-Grafiktreiber dafür hat mit den guten Kontakten der Raspi-Entwickler zu tun.

Die Firma Broadcom wurde seit 2012 mehrfach umstrukturiert. Schaut man auf ihre aktuelle Produktpalette, erschließt sich kaum noch, weshalb der Einsatz eines Broadcom-SoC für den Raspi vor zehn Jahren nahelag. Doch in Vor-Smartphone-Zeiten gehörte Broadcom zu den wichtigsten Lieferanten billiger Prozessoren sowohl für Handys als auch für Set-Top-Boxen für den TV-Empfang per Kabel und Satellit. Der BCM2835 war für Set-Top-Boxen gedacht, daher auch die Auslegung (und der Name) des VideoCore fürs Decoding von Videodaten. Auch das trug zum Raspi-Erfolg bei, denn schon früh gab es Projekte, die den Raspi in eine Medienzentrale verwandelten, etwa Kodi.

Auch die Zusatzchips für Ethernet, WLAN und Bluetooth stammten ursprünglich von Broadcom, doch deren „Wireless“-Sparte wechselte seit 2016 zweimal den Besitzer: Zunächst kaufte sie die Firma Cypress, die jedoch 2020 selbst von Infineon geschluckt wurde. Auf dem Raspi kommen keine nackten WLAN-Chips zum Einsatz, sondern mit einem Blechdeckel gekapselte Module, die gegen elektromagnetische Störungen geschirmt sind.

Kostendruck und Linux-Pflege

In jeder Raspi-Generation gibt es eine mager ausgestattete Basisversion, die hierzulande für 35 bis 40 Euro erhältlich ist (falls nicht gerade Chipkrise herrscht). Der Raspi Zero kostet sogar nur zwischen 5 und 20 Euro – trotz europäischer Fertigung. Um die Herstellungskosten zu reduzieren, sind die Platinen kompakt, haben wenige Lagen (nämlich sechs) und sind möglichst nur einseitig bestückt. Das wiederum erschwert das sogenannte „Routing“, also die Anordnung von Leiterbahnen auf der Platine, vor allem für hochfrequente Signale wie bei USB 3.0, HDMI und LPDDR4-RAM. Deshalb sind Controller und Buchsen dicht nebeneinander platziert.

Die Raspi-Entwickler bauen ihre Schaltungen aus möglichst wenigen unterschiedlichen elektronischen Bauteilen auf, die sich für sämtliche Raspi-Varianten eignen. Das vereinfacht die Beschaffung und führt zu größeren Stückzahlen, also höheren Rabattstaffeln.

Eine geringe Anzahl an Produktvarianten trägt zu höherer Zuverlässigkeit bei und reduziert den Aufwand zur Pflege von Firmware und Software. Dass neue Versionen von Raspberry Pi OS auch auf alten Raspis laufen, erleichtert Bastlern und industriellen Nutzern die Arbeit. Dafür nimmt die Raspi-Stiftung kleine Nachteile und zusätzlichen Aufwand in Kauf. Als mit der jüngsten Raspi-OS-Version auf Basis von

Bild: Kumbus



Debian 11 Bullseye Probleme auftraten, schob man innerhalb weniger Wochen eine „Legacy“-Version mit Debian 10 Buster nach. Mit der sorgfältigen Linux-Pflege hebt sich der Raspi von vielen konkurrierenden Einplatinencomputern ab. Weshalb es bei solchen ARM-Systemen so aufwendig ist, verbreitete Linux-Distributionen zum Fliegen zu bringen, erläutern [5, 6] im Detail.

Außer Raspberry Pi OS lassen sich auch Raspi-Ableger anderer Linux-Distributionen installieren, darunter Ubuntu, Arch Linux, Kali Linux und OpenWRT. Auch NetBSD, FreeBSD, RISC OS und sogar Windows 11 laufen auf Raspis, letzteres per UEFI-Bootloader. Industriekunden finden Echtzeitbetriebssysteme wie eCos RTOS.

Ausblick

In ihren ersten zehn Jahren eilte die Raspberry Pi Foundation von einem Erfolg zum nächsten. Erschwingliche Preise, möglichst leicht verständliche Technik, Kompatibilität und offene Software haben im Raspiversum höchste Priorität. Auf einer soliden finanziellen Basis erschließt sich die Raspi-Stiftung neue Betätigungsfelder, beispielsweise mit dem Schülercomputer Raspi 400 und dem Mikrocontroller RP2040.

Doch was kommt als Nächstes? Ob sich die Raspi-Macher an Tablets oder Notebooks wagen, ist ungewiss. Denn bei europäischer Fertigung würden solche Geräte wohl mindestens 150 bis 200 Euro kosten. Damit schießen sie weit über den Preisbereich hinaus, der den Raspi überhaupt erst attraktiv

Raspi-Technik gibt es auch in robustem Gehäuse zur Montage auf einer Hutschiene im Schaltschrank, etwa den Revolution Pi als Steuerungscomputer.

Literatur

[1] Christof Windeck, **Tastenaspi**, Raspberry Pi 400 im Tastaturgehäuse, c't 25/2020, S. 82

[2] Dr. Maik Merten, **Raspi-Kernschau**, Das Prozessor-Innenleben des Raspberry Pi 4 im Detail, c't 20/2019, S. 164

[3] Christof Windeck, **NAS-Pi**, Der Raspberry Pi 4 als NAS-Basis, c't 12/2020, S. 46

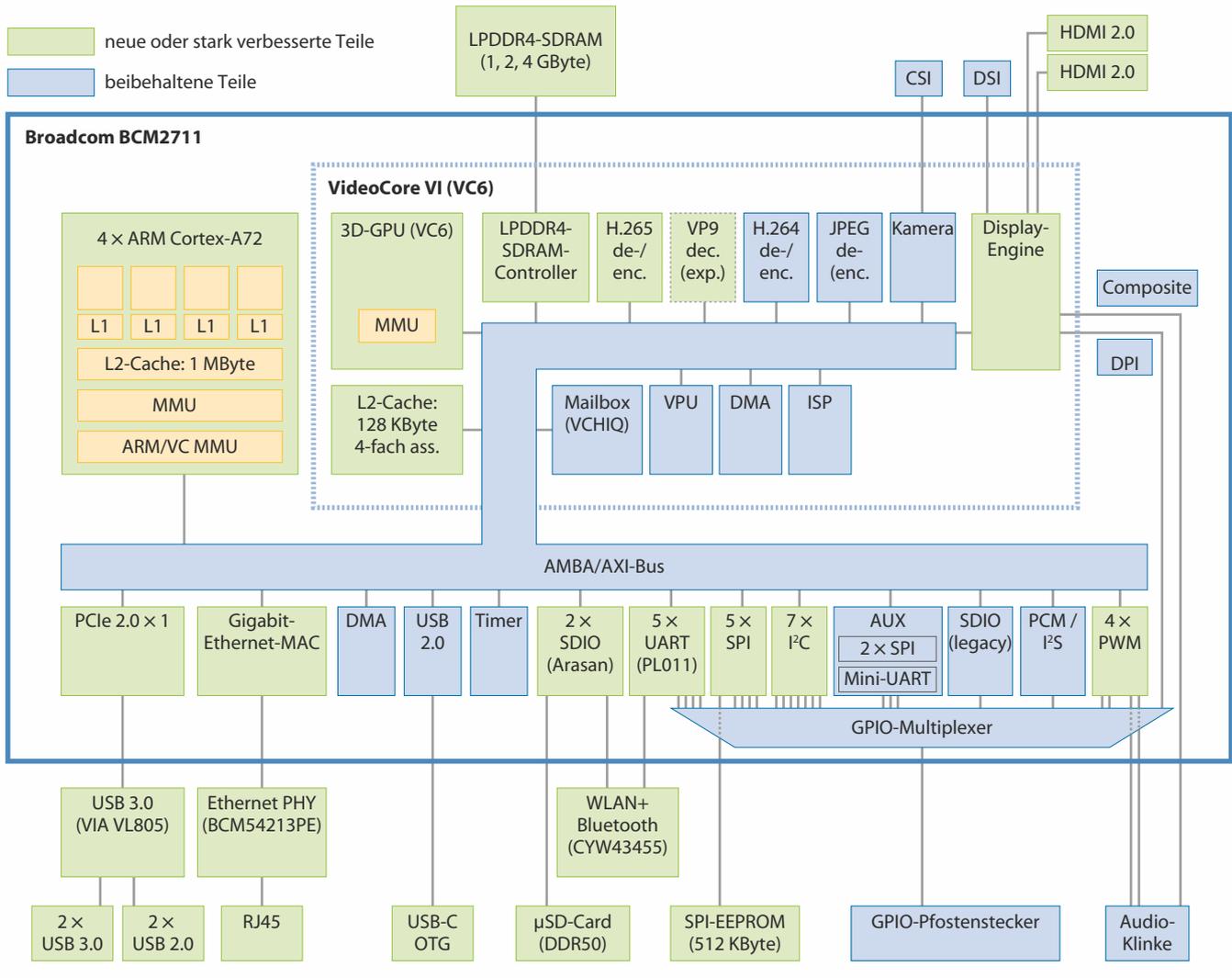
[4] Christof Windeck, **Himbeerchen**, Mini-Rechenmodul Raspberry Pi Compute Module 4 (CM4), c't 5/2021, S. 112

[5] Thorsten Leemhuis, **Für einander gemacht**, Warum es mit Upgrades bei Android hakt, erläutert an Linux beim Raspberry Pi, c't 8/2020, S. 120

[6] Thorsten Leemhuis, **Jenseits von Raspbian**, Mainstream-Distributionen und Vanilla-Linux für aktuelle Raspberry-Pi-Modelle, c't 8/2020, S. 128

Herz des Raspberry Pi 4: Broadcom BCM2711

Das System-on-Chip (SoC) BCM2711 vereint nicht nur vier CPU-Kerne mit einer GPU, sondern enthält auch Controller für viele Schnittstellen.



gemacht hat. Außerdem würde ein solcher Raspi-Mobilrechner mit Android-Tablets sowie den billigsten Windows- und ChromeOS-Geräten konkurrieren.

Ein komplett offengelegter Raspi mit RISC-V-Technik ist wohl noch Jahre entfernt, bis der dazu nötige Billigprozessor mit genügend Rechen- und Grafikleistung auf den Markt kommt und RISC-V-Linux

weiter gereift ist. Näher liegt ein Raspi 5 mit stärkeren ARM-Kernen und mehr PCIe-Schnittstellen, etwa zum Eigenbau von NAS und Routern. Ganz sicher bleibt die Raspi-Stiftung aber ihren Bildungszielen treu - und es wäre zu wünschen, dass auch Schulen in Deutschland, Österreich und Schweiz mehr Raspis nutzen. Happy Birthday, Raspi! (ciw) **ct**

Zahlen, Daten, Fakten: 10 Jahre Raspberry Pi

Weil er sich mehr Informatik-Bewerber wünschte, entwickelte der Informatiker Eben Upton während seiner Zeit an der Uni Cambridge einen Minicomputer für Schulkinder. Anders als die ebenfalls für den Unterricht entwickelten Mikrocontroller wie BBC micro:bit oder Caliope mini ist der Raspi ein vollständiger Computer, auf dem sich Betriebssysteme und Anwendungen installieren lassen. Das macht ihn auch für Maker und Tüftler interessant, was den unerwarteten kommerziellen Erfolg begründete.

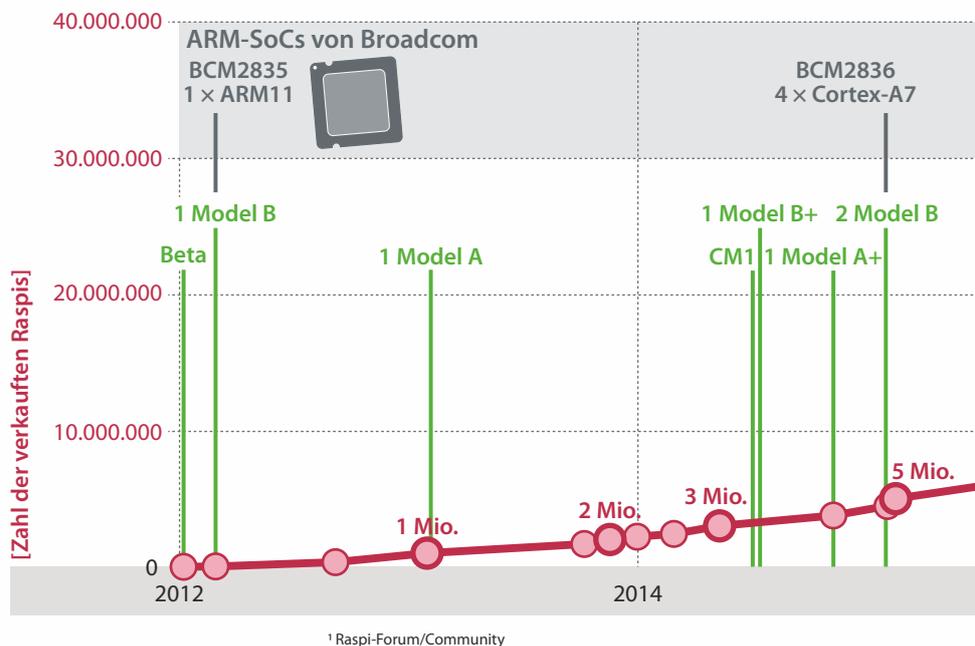
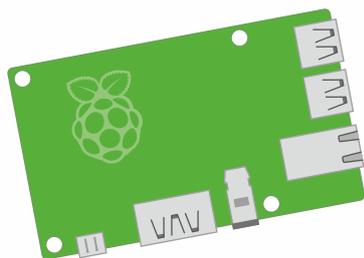
Um Bildungsprojekte aller Art kümmert sich die gemeinnützige Raspberry Pi Foundation. In Programmierclubs und Wettbewerben lernen und tüfteln

Jugendliche auf der ganzen Welt mit dem Raspi. Im Heimatland Großbritannien ist der Alleskönner an Schulen besonders verbreitet – sicher auch wegen der nationalen Bildungsinitiative „Coding at School“, die 2014 startete.

Für die Vermarktung ist die „Raspberry Pi Ltd.“ zuständig. Deren Umsatz stieg in den vergangenen zehn Jahren stetig, allerdings expandierte das Unternehmen auch stark. So meldete die Ltd. 2018 und 2019 höhere Personal- und Verwaltungskosten, der Gewinn ging vorübergehend zurück. Rund 3 Millionen britische Pfund fließen jährlich in die Weiterentwicklung. Bis Februar 2022 wurden über 45 Millionen Raspis verkauft. (dwi) **ct**

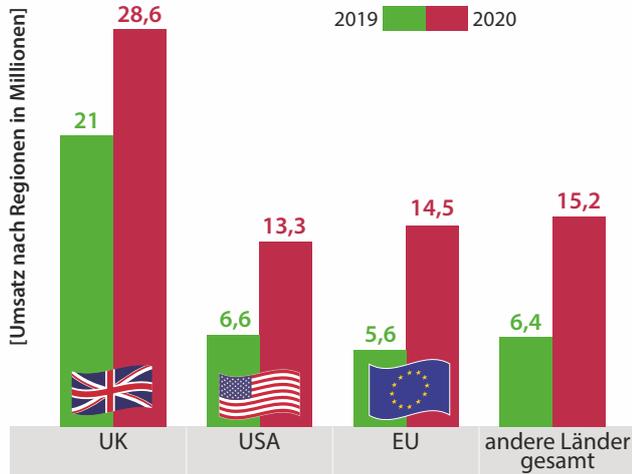
► Verkaufsschlager

Raspi-Erfinder Eben Upton habe die Nachfrage auf höchstens 10.000 Stück geschätzt, berichtete CNN Anfang 2013. 2021 dürfte die 40-Millionen-Marke deutlich überschritten worden sein. Auf allen Raspis rechnen System-on-Chip (SoCs) der Firma Broadcom mit ARM-Rechenkernen.¹



Umsatz nach Region

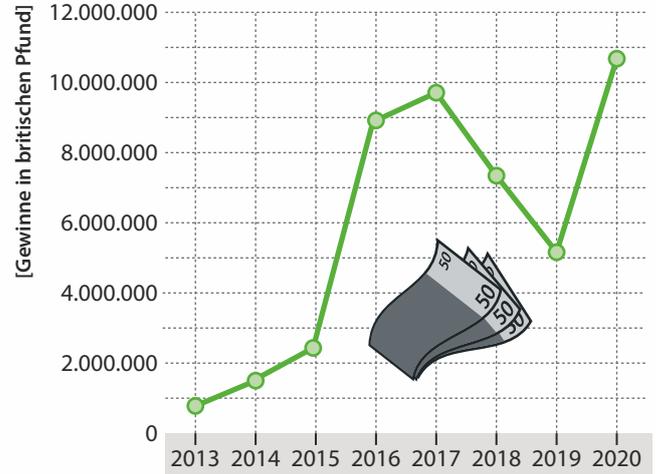
Nach wie vor ist die Nachfrage nach Raspis in Großbritannien besonders hoch.⁴



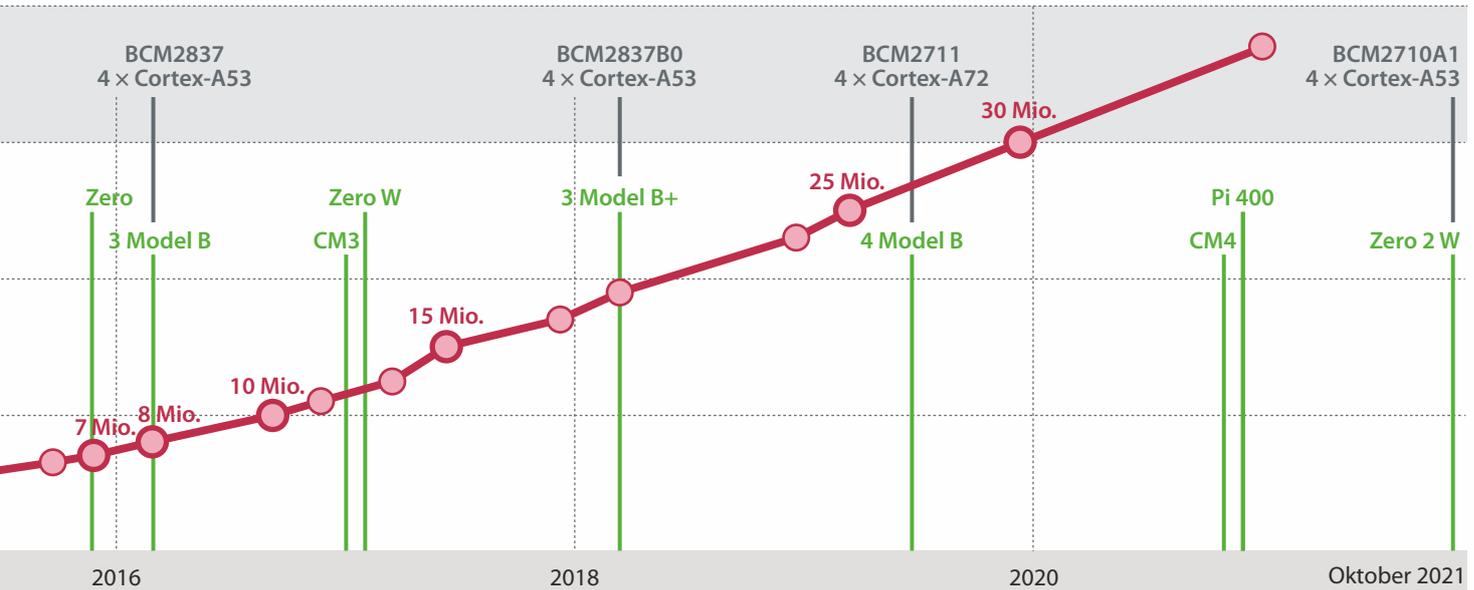
⁴ Raspberry Pi Ltd

Gewinne weltweit

Die „Raspberry Pi (Trading) Limited“, seit Dezember 2021 „Raspberry Pi Ltd“, erzielte in den vergangenen Jahren tendenziell steigende Gewinne.⁴

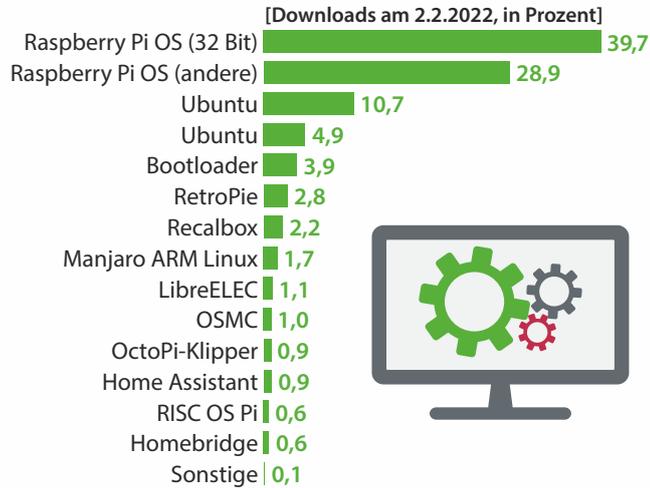


⁴ Raspberry Pi Ltd



► Betriebssysteme-Mix

Raspi-Nutzer laden am häufigsten Raspberry Pi OS herunter, experimentieren aber auch mit vielen anderen Systemen.²

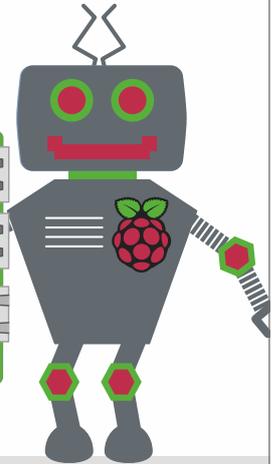


² rpi-imager Stats

► Code Clubs

Code Clubs sind ein Angebot an Kinder zwischen 7 und 13 Jahren. In den Clubs programmieren sie Scratch, HTML, CSS oder Python.³

13.000 Clubs weltweit
in **160 Ländern**,
180.000 Kinder und Jugendliche



³ Raspberry Pi Foundation

► Coolest Projects

Unter dem Namen „Coolest Projects“ bietet die Raspi Foundation jungen Mätern bis 18 Jahren eine Plattform im Web.³



³ Raspberry Pi Foundation

► Astro Pi

Seit 2015 gibt es eine jährliche Challenge für junge Coder, deren Programme auf Raspis auf der Weltraumstation ISS ausgeführt werden.³



³ Raspberry Pi Foundation



building **IoT**

Die Konferenz zu IoT und Industrie 4.0

10. und 11. Mai 2022 - Karlsruhe

Highlights aus dem Programm:

- Gestaltung und Entwurf eines IoT-Edge-Geräts
- Eingebettete Systeme programmieren mit Rust
- Test-Driven IoT mit Testcontainers
- Lean Prototyping für IIoT-Projekte
- Objekterkennung an der Edge
- Die Supply Chain im (I)IoT erfolgreich absichern

**Jetzt
Tickets
sichern!**

www.buildingiot.de

Workshops am 9. Mai: »IoT Supply Chain Security« und »Unüberwachtes Lernen im IoT-Umfeld«

Goldsponsoren

 **codecentric**

 **com2m**
connecting software solutions

 **HIVEMQ**

 **NOVATEC**

Silbersponsor

**Maiborn
Wolf**
Mensch IT

Veranstalter

 **heise Developer**

 **dpunkt.verlag**

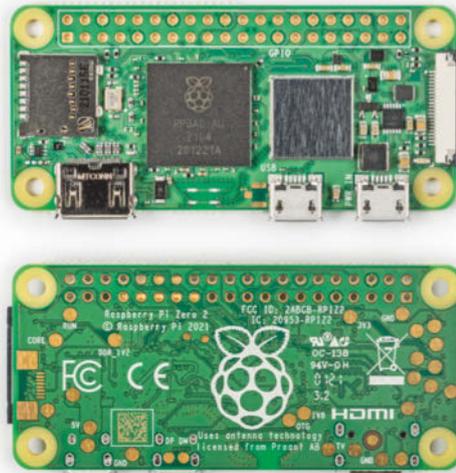
Test Raspberry Pi Zero 2 W

Der neue Raspi Zero 2 W ist um ein Mehrfaches schneller als sein Vorgänger Zero W von 2016, bleibt aber trotzdem sparsam.

Von **Christof Windeck**

Für manches Bastelprojekt ist ein normaler Raspberry Pi zu teuer oder zu groß. Das winzige „Compute Module“ CM4 ist wiederum nur auf einem zusätzlichen Basisboard nutzbar. Besser für kleine Basteleien geeignet sind die abgespeckten „Zero“-Raspis im Platinenformat 6,5 × 3 Zentimeter. 2015 erschien der erste noch ohne WLAN, 2016 sein Nachfolger „Zero W“ mit WLAN und Bluetooth für 10 Euro. Nun folgt mit dem 5 Euro teureren Raspberry Pi Zero 2 W endlich eine frische Obstlieferung mit deutlich mehr Rechenleistung.

Auf dem Neuling rechnet nicht der vergleichsweise starke Broadcom BCM2711 des aktuellen Raspi 4, sondern eine Variante des schwächeren BCM2837, mit dem der Raspi 3 ab 2016 ins Rennen ging. Dessen vier ARM-Kerne vom Typ Cortex-A53 liefern mehr als die vierfache Rechenleistung des BCM2835 auf dem alten Zero W. Denn der BCM2835 hat nur einen einzigen CPU-Kern mit der 19 Jahre



alten Mikroarchitektur ARMv6. Der Zero W wird mindestens bis 2026 weiter produziert. Von Zero W wurden bisher insgesamt rund 4 Millionen Stück verkauft, der Löwenanteil von der „W“-Version.

Im Vergleich zu einem normal großen Raspi 3 taktet der Prozessor beim Zero 2 W niedriger und nutzt nur halb soviel RAM (512 MByte statt 1 GByte). Der Zero 2 W hat auch keinen Ethernet-Adapter, sein WLAN-Modul funkt ausschließlich im 2,4-GHz-Band und es gibt weder USB-A- noch Audio-Buchsen. Der Flachbandkabelanschluss für ein DSI-Display fehlt ebenfalls, es gibt nur einen für eine CSI-2-Kamera. Um Kosten und Bauhöhe zu sparen, lässt die Raspberry Pi Foundation den GPIO-Pfostenstecker weg – wer ihn braucht, kann ihn selbst auflöten. Ansonsten hat der Hersteller beim Zero 2 W auf Rückwärtskompatibilität zum Zero W geachtet: Das Platinenformat

Raspberry Pi Zero 2 W

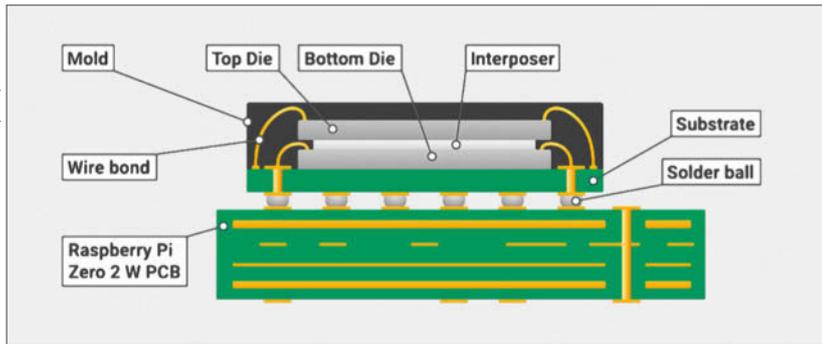
Einplatinencomputer mit ARM-Soc

| | |
|---------------------------------|--|
| Hersteller, URL | Raspberry Pi Foundation, raspberrypi.com |
| Prozessor | RP3A0 (Broadcom BCM2837, 4 × ARM Cortex-A53, 1,0 GHz) |
| GPU / RAM | VideoCore IV, in CPU integriert / 512 MByte (4 GBit) LPDDR2, integriert in RP3A0 |
| WLAN/Bluetooth / Platinenformat | 802.11b/g/n, 2,4 GHz, 1x1, Bluetooth 4.2 / Kantenlänge 65 mm × 30 mm |
| externe Anschlüsse | 1 × Mini-HDMI, 1 × Micro-USB 2.0, 1 × CSI-2, 1 × MicroSD-Kartenleser |
| sonstige Anschlüsse | Lötäugen für 40-poligen GPIO-Pfostenstecker |

Leistungsaufnahme (angeschlossen an Full-HD-Display)

| | |
|--------------------------------|----------------------------------|
| Leerlauf mit USB-Tastatur/Maus | 1 / 1,5 Watt ¹ |
| Leerlauf ohne Eingabegeräte | 0,7 / 0,85 Watt ¹ |
| Preis | 6 € (zurzeit schlecht lieferbar) |

¹ ohne/inklusive Netzteil gemessen



Der Chip RP3A0 auf dem Raspberry Pi Zero 2 W vereint einen Broadcom BCM2837 im untenliegenden Bottom Die mit 512 MByte LPDDR2-RAM im obenliegenden Top Die.

ist identisch und es gibt weiterhin eine Mini-HDMI- und zwei Micro-USB-Buchsen, davon eine für die Stromversorgung sowie eine als USB-Host. Der Zero 2 W passt in die meisten Zero-W-Gehäuse.

Chip-Paket

Auf dem Zero 2 W sitzt ein Bauelement mit der Bezeichnung RP3A0, das zwei Chips vereint: den BCM2837 und ein LPDDR2-SDRAM mit 512 MByte Kapazität.

Raspberry Pi OS erkennt die vier Kerne des RP3A0 genau wie beim BCM2835 als „ARMv7l“-Kerne. Sie

takten mit höchstens 1 GHz statt wie bei Raspi 3B und 3B+ mit 1,2 beziehungsweise 1,4 GHz. Der Zero 2 W rechnet deshalb etwas langsamer als der 3B und kommt bei Weitem nicht an einen Raspi 4 heran: Der ist im 7-Zip-Benchmark um 70 Prozent schneller und hat bis zu 8 GByte RAM, also die 16-fache Kapazität.

In unseren Tests war der Raspi Zero 2 W sparsamer als ein Raspi 3B: Er brauchte im Leerlauf mit USB-Eingabegeräten 1,0 statt 1,35 Watt, gemessen an 5 Volt. Ohne USB-Geräte waren es 0,7 statt 1,15 Watt. Jeweils inklusive Netzteil – also an 230 Volt gemessen – waren es 1,5 versus 1,9 Watt beziehungsweise 0,85 zu 1,7 Watt. Unter hoher Last im 7-Zip-Benchmark schluckte der Zero 2 W höchstens 2,9 Watt (3,5 Watt mit Netzteil). Anders gesagt braucht ein Zero 2 W inklusive Netzteil im Leerlauf weniger als 1 Watt und ist ähnlich sparsam wie sein Vorgänger Zero W.

Fazit

Der Raspi Zero 2 W soll keine Alternative zum deutlich stärkeren Compute Module CM4 sein. Und im Vergleich zu einem Raspi 3 hat er weniger Anschlüsse, nur halb so viel Arbeitsspeicher und etwas weniger Rechenleistung. Doch der Zero 2 W ist eben auch kleiner, billiger und sparsamer. Er glänzt in Projekten, wo es auf niedrigen Preis und kompakte Bauform ankommt, beispielsweise als portable Retro-Spielkonsole (siehe S. 106) oder als Streaming-„Adapter“ für Spotify und AirPlay (siehe S. 110). (ciw) **ct**

TECHNIKUNTERRICHT MACHT ENDLICH SPÄß!



Make:Education

Mit **Make Education** erhalten Sie jeden Monat kostenlose Bauberichte und Schritt-für-Schritt-Anleitungen für einen praxisorientierten Unterricht:



Für alle weiterführenden Schulen



Fächerübergreifend

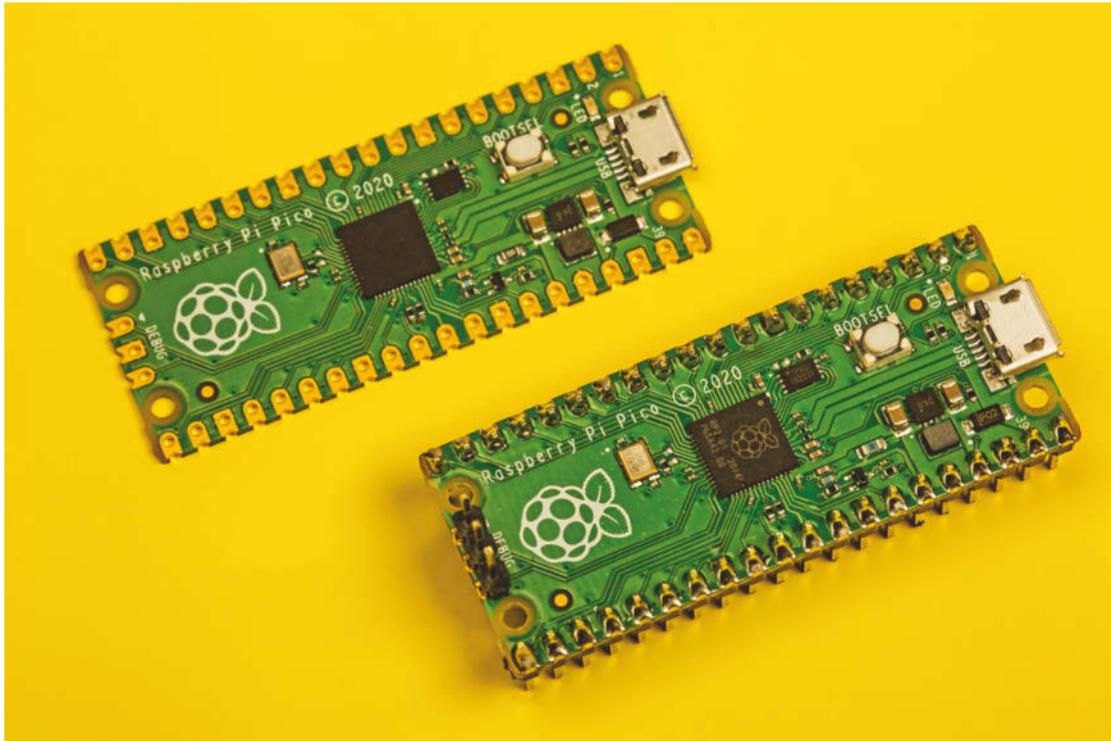


Digital zum Downloaden



Monatlicher Newsletter

Jetzt kostenlos downloaden: make-magazin.de/education



Der Mikrocontroller Raspberry Pi RP2040

Der RP2040 ist der erste selbst entwickelte Chip der Raspberry Pi Foundation. Er macht einiges anders als andere Mikrocontroller und eignet sich damit für besondere Projekte. Ein Überblick.

Von **Maik Merten** und **Christof Windeck**

Mancher mag sich fragen, ob die Menschheit wirklich noch einen weiteren Mikrocontroller braucht. Schließlich gibt es bereits eine riesige Vielfalt, die vermeintlich jeden erdenklichen Anwendungsbereich abdeckt [1]. Doch der RP2040 der Raspberry Pi Foundation hat einige attraktive Besonderheiten. Anfang 2021 erschienen der neue

Mikrocontroller und die damit bestückte Entwicklerplatine Raspberry Pi Pico. Der RP2040 erschließt der Raspberry Pi Foundation andere, neue Einsatzbereiche und Geschäftsfelder.

Der mit 40-Nanometer-Technik von TSMC gefertigte Mikrocontroller RP2040 ist auch einzeln erhältlich und kostet weniger als 1 Euro. Kurz nach

dem Start des etwa 4 Euro teuren Platinchens Raspi Pico, das an einen Arduino Nano erinnert, kamen zahlreiche andere Mikrocontrollerboards und Steuermodule mit RP2040 heraus, siehe Tabelle auf Seite 24.

RP2040-Besonderheiten

Der RP2040 hat vier wesentliche Besonderheiten: Zwei Rechenkern statt nur einem, spezielle Ein-/Ausgabefunktionen namens Programmable I/O (PIO), offengelegter Boot-ROM-Code und kein eingebauter Flash-Speicher. Dazu kommt noch eine Kombination weiterer Eigenschaften, die in dieser Preisklasse nicht häufig sind.

Aber der Reihe nach: Die beiden 32-Bit-Rechenkern des RP2040 sind vom ARM-Typ Cortex-M0+. Sie laufen mit bis zu 133 MHz Taktfrequenz, also schneller als in vielen anderen Mikrocontrollern. Dadurch steht relativ viel Rechenleistung bereit, allerdings keine Hardware-Einheiten für die Verarbeitung von Gleitkommazahlen (Floating Point, FP) und zur Division ganzer Zahlen (Integer). Die Raspi Foundation packt deshalb einerseits optimierte FP-Routinen ins Boot-ROM und hat andererseits eine Hardware-Einheit für Integer-Division in den RP2040 eingebaut.

Die beiden Rechenkern lassen sich parallel nutzen, aber man kann den zweiten auch für Spezialaufgaben verwenden, unter anderem fürs Debugging. Auch die Verknüpfung der beiden Rechenkern untereinander und mit den Ein-/Ausgabekontakten (General Purpose I/O, GPIO) hat die Raspi Foundation sehr geschickt gelöst, dazu später mehr.

Die zweite Besonderheit des RP2040 sind die beiden PIO-Funktionseinheiten. Sie lassen sich dazu nutzen, vergleichsweise leistungsfähige Schnittstellen sozusagen „in Software“ nachzubilden, sogar einen VGA- oder HDMI-Ausgang für ein Display. Die PIOs taugen ferner zur Ansteuerung von LED-Streifen, Barcode-Scannern, I2S-Soundchips, (Micro-)SD-Karten oder als CAN-Bus-Interface. In manchen Fällen braucht man noch einen passenden Transceiver-Baustein, aber einer für den CAN-Bus kostet nur 1,50 Euro.

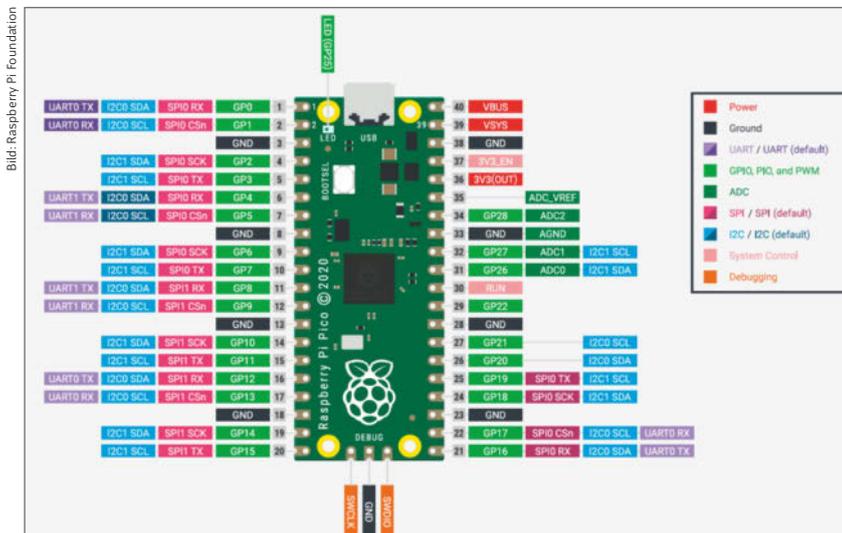
Der Verzicht auf eingebauten Flash-Speicher macht den RP2040 billiger und erspart es der Raspi Foundation, mehrere Chipvarianten mit unterschiedlicher Flash-Ausstattung zu verkaufen. Der Nachteil liegt auf der Hand: Anders als etwa beim Arduino muss zusätzlich ein separater NOR-Flash-Chip auf die Boards. Die kleinsten 16-MBit-Chips (2 MByte) kosten kaum 50 Cent, der RP2040 kann bis zu 128 MBit (16 MByte) anbinden.

Im RP2040 steckt auch ein USB-1.1-Controller; den haben zwar andere Mikrocontroller ebenfalls, aber der im RP2040 kann auch als Host arbeiten. Im Device-Modus wiederum lässt sich der RP2040 sehr leicht mit Software versorgen, indem er sich als Massenspeichergerät an einem PC anmeldet (der nötige Code steckt im Boot-ROM). Den Programmcode für sein Projekt kopiert man dann via USB in den Flash-Chip, nämlich als Image im Format „USB Firmware Flash“ (U2F).

Eine weitere Besonderheit ist die gute öffentliche Dokumentation des RP2040 und des Raspberry Pi Pico; es gibt auch viele Code-Beispiele bei GitHub, wo das Boot-ROM ebenfalls offengelegt ist (siehe ct.de/wrkw). Der Raspi Pico lässt sich mit MicroPython programmieren, wozu man etwa die IDE Thonny oder Microsoft Visual Studio Code einspannen kann. Zudem stellt die Raspi Foundation ein Software Development Kit (SDK) für C/C++ mit umfangreicher Dokumentation bereit (siehe ct.de/wrkw).

Speicherei

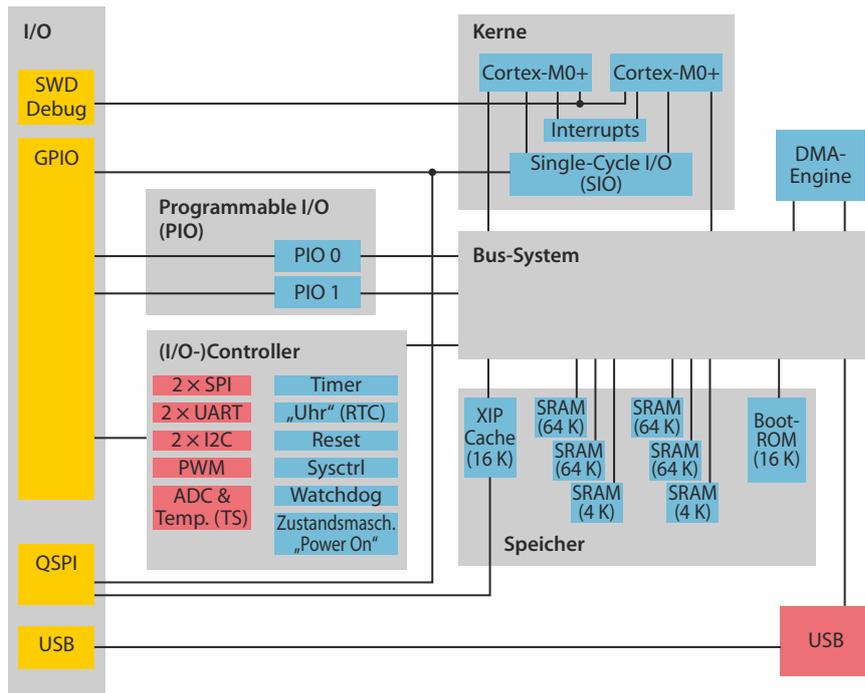
Die beiden Rechenkern können insgesamt 264 KByte eingebautes SRAM als Arbeitsspeicher nutzen; es



Der Raspberry Pi Pico mit dem Mikrocontroller RP2040 hat 2 MByte Flash-Speicher und 26 GPIO-Kontakte.

Blockschaltbild Raspberry Pi RP2040

Der von der Raspberry Pi Foundation entwickelte Mikrocontroller RP2040 hat als Besonderheit zwei Programmable-I/O-(PIO-)Blöcke, um schnelle Schnittstellen bereitzustellen. Außerdem hat er zwei 32-Bit-ARM-Kerne von Typ Cortex-M0+ und 264 KByte SRAM, aber keinen Flash-Speicher. Aus einem per QSPI angeschlossenen Flash-Chip kann er Daten direkt ausführen (Execute-in-Place, XIP). 16 KByte XIP-Cache puffern Flash-Zugriffe.



verteilt sich auf sechs „Bänke“ (Banks), die mit kleinen Einschränkungen parallel arbeiten: vier mit je 64 KByte, zwei mit je 4 KByte. Ebenfalls eingebaut und bei der Herstellung unveränderlich programmiert sind 16 KByte ROM für den Bootcode.

Auf einen extern angeschlossenen Flash-Speicherchip, der typischerweise den Programmcode enthält – bei Mikrocontrollern spricht man oft auch von Firmware –, greift der RP2040 per Serial Peripheral Interface (SPI) zu, genauer per Quad-SPI (QSPI).

QSPI hat im Vergleich zu SPI vier Datenleitungen statt nur eine und ist deshalb schneller.

Code aus dem Flash-Speicher kann der RP2040 direkt ausführen, ohne ihn zuvor in sein SRAM kopieren zu müssen: Für diese (gängige) Funktion namens Execute-in-Place (XIP) sind 16 KByte SRAM-Cache eingebaut. Schaltet man XIP ab, lässt sich dieser Cache als zusätzlicher Arbeitsspeicher verwenden. Weitere 4 KByte SRAM lassen sich „befreien“, wenn man den USB-Funktionsblock abschaltet.

RP2040: Programmable I/O (PIO)

Jeder PIO-Block des Raspberry Pi RP2040 enthält vier programmierbare Zustandsmaschinen mit je zwei 32-Bit-Schieberegistern: Je eines für Ein- und Ausgang. Per PIO lassen sich auch schnelle Schnittstellen wie HDMI nachbilden, bei denen es auf hohen Takt und wenig Jitter ankommt.



Schnittstellen

Für Steuerungen gedachte Mikrocontroller haben zahlreiche eingebaute Schnittstellen, so auch der RP2040. 30 Kontakte (für 1,8 oder 3,3 Volt Spannung, aber nicht 5-Volt-tolerant) dienen als „Allzweck-Ein- und Ausgänge“ (General Purpose I/O, GPIO), viele lassen sich mit Spezialfunktionen beschalten. Dazu gehören je zwei Funktionsblöcke für serielle Kommunikation (UART), SPI und I2C. Ein Analog-Digital-Wandler (ADC) lässt sich mit bis zu vier GPIO-Pins verbinden und misst als fünften Kanal den eingebauten Temperatursensor (TS). Zum ADC-Funktionsblock gehört auch ein FIFO-Register, um Werte per Direct Memory Access (DMA) ins SRAM zu schreiben.

Zum Debugging ist die von ARM spezifizierte Schnittstelle Serial Wire Debug (SWD) eingebaut. Die beiden Prozessorkerne sind intern über einen sogenannten Single-Cycle-I/O-(SIO-)Block mit den GPIO-Pins verbunden, also sehr flink. Der SIO-Block hilft außerdem bei der Kommunikation der Rechen-

kerne untereinander sowie bei der Synchronisation von Threads, die auf beiden Kernen laufen, und zwar durch FIFO-Register und Spinlock-Hardware; letztere gewährleistet exklusiven Zugriff auf Ressourcen. Im SIO-Block ist zudem die oben erwähnte Divisions-einheit eingebaut sowie pro Kern zwei Interpolatoren, vor allem für Audio-Anwendungen.

Für Steuerungsfunktionen sind außer den passenden I/O-Schnittstellen vor allem Timer-Funktionen nötig. Der RP2040 hat einen 64-bittigen und Interrupt-fähigen Haupttimer, der im Mikrosekundentakt läuft. Einstellbare Alarme beachten davon aber nur die unteren 32 Bit, weshalb es für Zeiträume von mehr als einigen Minuten noch eine „Echtzeituhr“ (RTC) gibt, die ebenfalls Interrupts auslösen kann. Für Pulsweitenmodulation (PWM) sind acht weitere 16-Bit-Timer vorhanden. Außerdem gibt es in der DMA-Einheit noch vier Timer sowie die „Sys-Tick“-Timer mit je 24 Bit in den Rechenkernen. Falls das nicht reicht, lassen sich auch die PIO-Blöcke als 32-Bit-Timer programmieren.

Platinen und Geräte mit Raspberry Pi RP2040

| Hersteller/Anbieter | Produkt | Eigenschaften |
|-------------------------|---------------------|---|
| Raspberry Pi Foundation | Pico | 2 MByte Flash, LED, Micro-USB, 4 €; mit Pfostenstecker 9 € |
| Adafruit | Feather RP2040 | 8 MByte Flash, RGB-LED, Qwiic, LiPo-Akkulader, USB-C, 20 € |
| Adafruit | QT Py RP2040 | 8 MByte Flash, winzig, RGB-LED, Qwiic, USB-C, 11 € |
| Arduino | Nano RP2040 Connect | 16 MByte Flash, WLAN & BLE via ESP32 (µBlox Nina W102), Mikrofon, Gyrosensor, Micro-USB, 25 € |
| Cytron | Maker Pi RP2040 | 2 MByte Flash, 15 (RGB-)LED, Motorsteuerung, Summer, LiPo-Akkulader, Grove, Micro-USB, 20 € |
| Melopero | Shake RP2040 | 16 MByte Flash, RGB-LED, Qwiic, LiPo-Akkulader, USB-C, 25 € |
| Pimoroni | Tiny 2040 | 8 MByte Flash, winzig, RGB-LED, USB-C, 10 € |
| Pimoroni | Pico Lipo | 4/16 MByte Flash, Qwiic, LiPo-Akkulader, USB-C, 12/16 € |
| Pimoroni | PicoSystem | akkubetriebene „Spielkonsole“ mit RP2040 und Display, 55 € |
| Seedstudio | Seeed XIAO RP2040 | 2 MByte Flash, winzig, LED, USB-C, 8 € |
| Sferalabs | Iono RP | Hutschienenmodul, 16 MByte Flash, 4 230-V-Relais, 190 € |
| Sparkfun | Pro Micro | 16 MByte Flash, RGB-LED, Qwiic, USB-C, 11 € |
| Sparkfun | Thing Plus | 16 MByte Flash, RGB-LED, MicroSD-Leser, LiPo-Akkulader, USB-C, 17 € |
| Breakout-Boards | | |
| Cytron | Maker Pi Pico Base | MicroSD-Leser, Taster, Klinkebuchse, Grove, Pfostenstecker, 9 € |
| DiP-Pi | PiIoT | WLAN-Modul (ESP8266), MicroSD-Leser, LiPo-Akkulader, Spannungsregler 6 – 18 V, 16 € |
| Kitronik | Pico Robotics Board | 2 Motortreiber, I2C-Treiber, Schraubklemmen für Servos, 20 € |
| Pimoroni | Pico RG Keypad Base | 16 Silikon Tasten mit eingebauten RGB-LEDs, 25 € |
| Pimoroni | Pico Explorer | Steckbrett, Display, Motortreiber und mehr, 26 € |

an Qwiic-Anschluss (I2C, Sparkfun) passt auch Stemma-QT (Adafruit), für Grove (Seeed/Cytron) gibt es Adapter

PIO-„Prozessoren“

Die beiden PIO-Blöcke kann man sich wie I/O-Spezialprozessoren vorstellen, die unabhängig von den Cortex-M0+-Kernen arbeiten. Das Konzept erinnert an die Universal Digital Blocks (UDBs) der Cypress-PSoc-Mikrocontroller. Wie eingangs erklärt, sind die PIOs vor allem dazu gedacht, schnelle oder latenzkritische digitale Schnittstellen umzusetzen. Das kann man zwar im Prinzip auch mit Code für die normalen Mikrocontroller-Rechenkerne erledigen, Stichwort „Bit-Banging“. Doch das erzeugt hohe Last auf den Kernen und stößt an Grenzen, wenn höhere Frequenzen nötig sind. Deshalb hat jeder PIO-Block vier programmierbare Zustandsmaschinen: Insgesamt gibt es also acht „I/O-Prozessoren“.

Zu Nachbildung einer digitalen Schnittstelle ist es typischerweise nötig, an einem oder mehreren Kontakten (also GPIO-Pins) bestimmte Bitmuster mit präzisen Zeitbezügen (Frequenz, Jitter) auszugeben oder einzulesen. Jede PIO-Zustandsmaschine verfügt deshalb über zwei 32-bittige Schieberegister für Ein- und Ausgabe (In/Out Shift), deren Inhalte in

beide Richtungen um beliebige Stellen verschoben werden können. Hinzu kommen zwei 32-bittige Hilfsregister X und Y (Scratch X/Y), die Daten mit den Schieberegistern austauschen oder etwa als Zähler dienen können. Ein Befehlszähler (Program Counter, PC) gibt an, welche Befehle aus dem Befehlsspeicher ausgeführt werden. Der Takt der Zustandsmaschine lässt sich über einen Taktteiler einstellen, um auch Protokolle zu „sprechen“, die nicht ins Taktraster des Mikrocontroller-Systems passen.

Jeder PIO-Block hat einen Befehlsspeicher für 32 Befehle und mit vier Ports, den alle vier Zustandsmaschinen dieses Blocks gemeinsam nutzen. Der Datenaustausch zwischen dem Mikrocontroller-System und der einzelnen Zustandsmaschine geschieht über zwei FIFO-Puffer mit Platz für vier 32-Bit-Werte. Üblicherweise verwendet man je einen dieser FIFOs für die Sende- und Empfangsrichtung, die Puffer lassen sich jedoch auch zusammenschalten, sodass ein Puffer mit Platz für acht Werte entsteht, entweder zum Senden oder Empfangen.

Die Ein- und Ausgaben einer Zustandsmaschine in Richtung Außenwelt lassen sich auf GPIO-Pins

abbilden. Dabei gibt man einen GPIO-Pin an, den die Zustandsmaschine als „Pin 0“ auffasst. Möchte man mehrere Pins nutzen, zählt sie diese ab Pin 0 hoch. Zählt die PIO über den letzten GPIO-Pin hinaus, landet sie per Überlauf wieder beim ersten.

Pro Zustandsmaschine sind vier GPIO-Abbildungen definierbar: Input, Output, Set und Sideset. Per „Set“ weist man dem GPIO-Pin einen festen Wert zu (High oder Low, also 1 oder 0). „Sideset“ setzt einen Wert als Nebenfunktion zusätzlich zur Hauptfunktion. GPIO-Abbildungen dürfen überlappen, sowohl zwischen einzelnen Abbildungen als auch über mehrere Zustandsmaschinen hinweg. Falls mehrere davon den Wert derselben Ausgabe-Pins innerhalb eines Taktes verändern, so werden die Werte der höchstnummerierten Zustandsmaschine übernommen.

Zur Programmierung der PIOs dient eine spezielle Assembler-Sprache, die sich über Bibliotheken in den C- oder MicroPython-Code einbetten lässt. Bei MicroPython heißt die Bibliothek „rp2“; sie ermöglicht es, PIO-Assemblerprogramme aus Python-Funktionsaufrufen zusammensetzen. Für C/C++ gibt es das Programm „pioasm“ im RP2040-SDK, das beispielsweise C-Kopfdateien mit PIO-Assemblercode ausgibt. Diese Kopfdateien bindet man in ein C-Programm ein.

RP2040-Boards

Das für unter 4 Euro erhältliche Raspberry Pi Pico ist sozusagen das offizielle RP2040-Entwicklerboard, das aber auch in großen Stückzahlen für andere Systeme bestellbar ist. Eine doppelt so teure Version kommt mit bereits aufgelöteten Pfostensteckerleisten, die billigere nur mit Lötaugen. Noch billiger kommt man an den RP2040 als (SMD-)Einzelchip.

Teurer, aber entweder viel kleiner als das Pico oder besser ausgestattet sind RP2040-Boards etwa von Adafruit, Arduino, Seed Studio und Sparkfun. Manche haben Qwiic- beziehungsweise Stemma-QT-Steckverbinder zum lötfreien Anschluss von Sensoren per I2C.

Der RP2040 braucht nicht viel Strom, auch wenn es noch sparsamere Mikrocontroller gibt. Damit eignet sich der RP2040 auch für Projekte mit Akkustromversorgung. Einige RP2040-Boards enthalten dazu gleich die Ladeelektronik für einen Lithium-(Polymer)-Akku. Im Standby begnügt sich der RP2040 mit wenigen Milliwatt, unter Volllast bleibt er unter 0,5 Watt. Er kann mit einer einzigen Gleichspannung auskom-

men, weil ein eingebauter Wandler seinen Digitalern versorgt. Die 3,3 Volt sind nur für den USB-Controller Pflicht und für ADC empfohlen, den GPIOs reichen auch 1,8 Volt. Braucht man kein USB, kann man also auf 3,3 Volt verzichten, um die Leistungsaufnahme zu senken.

Ausblick

Der RP2040 lässt sich nicht nur mit MicroPython und C/C++ programmieren, sondern längst auch mit CircuitPython oder der Arduino-IDE. PlatformIO kennt den Chip ebenfalls, ebenso wie die Visual-Studio-Code-Extension PicoGo. Mehrere Projekte passen ihre Echtzeitbetriebssysteme (Real Time Operating Systems, RTOS) für den RP2040 an, darunter Amazon FreeRTOS, NuttX und RT-Thread OS.

Eben Upton, Chef der Raspi Foundation, erwartet, dass der RP2040 in großen Stückzahlen Abnehmer findet. Firmen sollen ihn in ihre eigenen Produkte einbauen. Dazu soll beitragen, dass der RP2040 bis mindestens 2028 lieferbar ist; voraussichtlich wird er aber so lange angeboten, wie TSMC 40-Nanometer-Chips produziert.

Vermutlich bleibt der RP2040 auch kein einmaliger Versuchsballon, sondern die Raspi Foundation dürfte die Mikrocontrollerfamilie künftig erweitern. In diesem Zusammenhang ist es interessant, dass sich die Raspi Foundation für ARM-Kerne entschieden hat und nicht etwa für 32-Bit-RISC-V-Kerne. Allerdings sitzt die Foundation in Cambridge, quasi nebenan vom ARM-Hauptquartier.

Bei der Frage, ob auch zukünftige Systems-on-Chip für die „großen“ Raspberry Pis hausintern entwickelt werden, winkte Raspi-Chef Upton jedoch ab: Die Entwicklung solcher SoCs sei zu aufwendig, um sie alleine zu stemmen, weshalb man auch in Zukunft mit Partnern zusammenarbeite. Aber man kann spekulieren, ob vielleicht der PIO-Block auch bei kommenden Raspis Einzug hält.

Mit dem RP2040 liefert die Raspberry Pi Foundation einen 1-Euro-Mikrocontroller, der sich dank seiner üppigen Ressourcen flexibel einsetzen lässt. Seine PIO-Blöcke ermöglichen es, viele Schnittstellen quasi „in Software“ umzusetzen.

Der RP2040 eignet sich besonders gut für die komfortable Programmierung mit MicroPython, ist aber gleichzeitig offen für C/C++ und Echtzeitbetriebssysteme. Um den RP2040 wächst bereits ein vielfältiges Ökosystem aus Software und Hardware, mit dem sich Produktideen und Projekte effizient und schnell umsetzen lassen. (ciw) **ct**

Literatur

[1] Christof Windeck, **Grundlagen Mikrocontroller**, Was Mikrocontroller von Mikroprozessoren unterscheidet, c't 24/2021, S. 136

Datenblätter, Boot-ROM

ct.de/wrkw

Raspi superschnell einrichten

Einschalten und wohlfühlen: Mit dem Raspberry Pi Imager geben Sie Ihrem Raspi schon vor dem ersten Hochfahren alle wichtigen Einstellungen mit auf den Weg, einschließlich der WLAN-Konfiguration. So ist Ihr Raspi sofort startklar.

Von **Ronald Eikenberg**

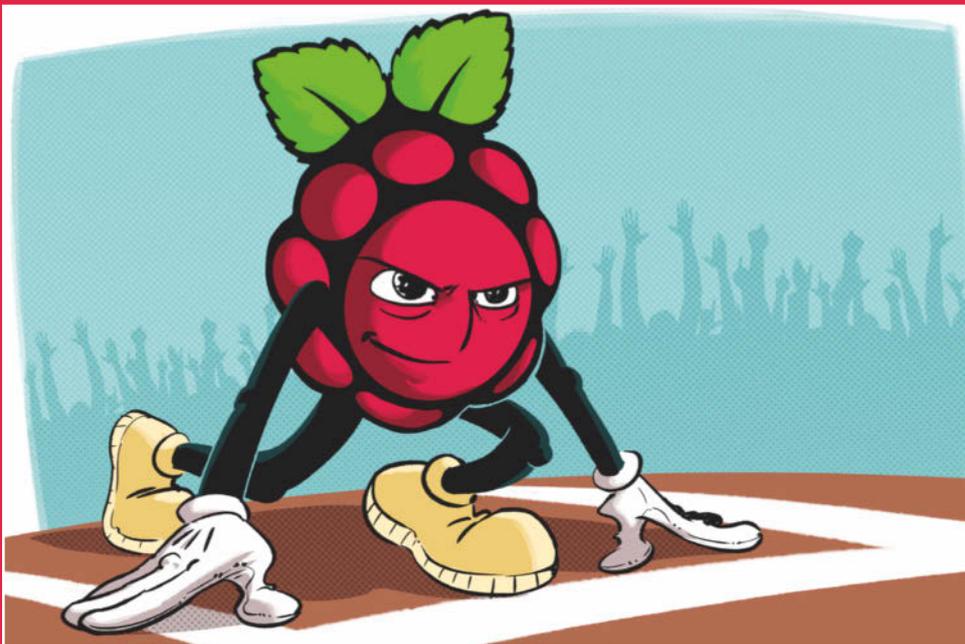


Bild: Albert Hulm

| | |
|--|----|
| Raspi superschnell einrichten | 26 |
| Autonome Linux-Installation | 34 |
| FAQ zu Betriebssystem, Hardware und Kamera | 38 |

Die Ersteinrichtung eines Raspis haben wir schon oft beschrieben. Und doch gibt es einen weiteren Artikel dazu: Denn dank einer recht neuen Funktion des offiziellen Image-Tools „Raspberry Pi Imager“ geht das leichter denn je von der Hand. So schreiben Sie die wichtigsten Einstellungen direkt mit auf die Micro SD-Karte. Anschließend verbindet sich Ihr Raspi schon beim ersten Start wie von Geisterhand mit Ihrem WLAN und nimmt Befehle per SSH entgegen. Die passende Tastaturbelegung ist auch schon eingestellt und das unsichere Standardpasswort „raspberrypi“ geändert. Sie können direkt loslegen. Wer häufiger neue Raspis in Betrieb nimmt, kann so richtig viel Zeit sparen – und Einsteiger kommen leichter ans Ziel.

Das offizielle Imager-Tool der Raspberry Pi Foundation hat im Wesentlichen einen Zweck und diesen erfüllt es mit Bravour: Es schreibt Betriebssystem-Images auf Micro SD-Karten, und zwar genau so, dass sie vom Raspi gebootet werden können. Um diesen Prozess so einfach wie möglich zu gestalten, bietet der Imager neben dem Standardbetriebssystem „Raspberry Pi OS“ eine Auswahl weiterer Images an, etwa die genügsame Lite-Variante ohne grafische Oberfläche, LibreELEC für den Einsatz als Medienabspieler und RetroPie, das den Raspi in eine Retro-Spielekonsole verwandelt. Die Images lädt das Tool bei Bedarf automatisch aus dem Internet nach. Zudem kann man eigene Images wählen oder die



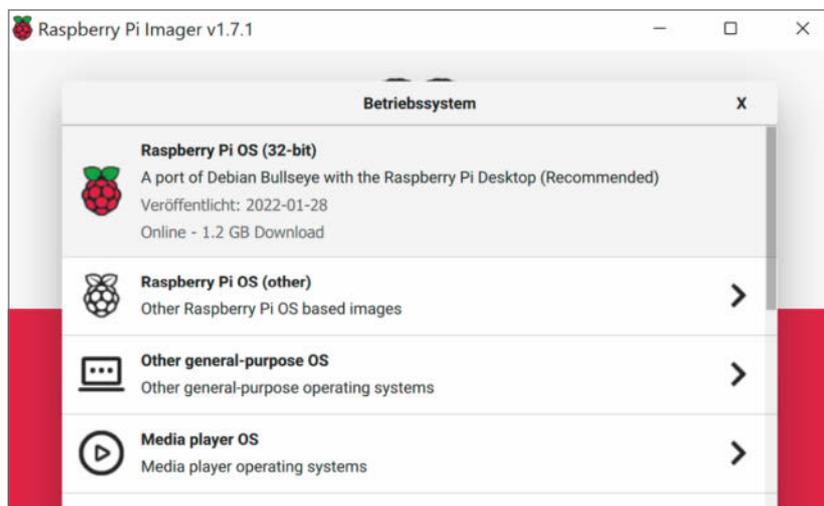
Der Raspberry Pi Imager bespielt SD-Karten mit dem gewünschten Betriebssystem.

SD-Karte einfach nur FAT32-formatieren, falls sie nach dem Einsatz im Raspi wieder einem anderen Zweck dienen soll.

Normalerweise fängt die Arbeit nach dem Beschreiben der Speicherkarte jedoch erst an: Je nachdem, welche Aufgabe der vielseitige Mini-PC erfüllen soll, sind einige weitere Schritte nötig, um ihn betriebsbereit zu machen; etwa die Einrichtung von WLAN und SSH-Server, das Ändern des Passworts für den „pi“-User und die Wahl von Tastaturlayout und Zeitzone. All das ist nicht sehr kompliziert, jedoch etwas fummelig. Und es kostet Zeit. Seit Version 1.6.1 des Raspberry Pi Imager fällt das alles flach, denn das Tool schreibt die wichtigsten Einstellungen auf Wunsch gleich mit auf die SD-Karte. Sie müssen lediglich einmalig die gewünschten Vorgaben einstellen und können anschließend beliebig viele SD-Karten damit bespielen. So können Sie nicht nur für sich selbst, sondern auch für Familie und Freunde individuelle Raspis vorbereiten, die vom ersten Start an ohne weitere Handgriffe einsetzbar sind.

Installieren Sie zunächst die aktuelle Version des Raspberry Pi Imager für Windows, macOS oder Linux (siehe ct.de/w924). Mit dem Befehl `sudo apt install rpi-imager` können Sie es gar auf einem Raspi installieren, um SD-Karten für andere Raspis an den Start zu bringen. Falls Sie das ausprobieren möchten und Ihren Raspi von einer SD-Karte booten, benötigen Sie hierfür einen externen (Micro) SD-Kartenleser, da Sie die Speicherkarte mit dem laufenden Betriebssystem nicht entfernen sollten.

Nach dem Programmstart klicken Sie links auf „OS wählen“, um das gewünschte Betriebssystem zu selektieren. Wenn Sie den Raspi wie einen Desktop-PC mit Monitor, Maus und Tastatur nutzen



Es stehen diverse Images zur Auswahl, etwa für den Einsatz als Desktop-PC, Medienabspieler oder Spielekonsole.



Über die erweiterten Optionen können Sie dem Raspi viele wichtige Einstellungen wie WLAN-Konfiguration und Tastaturlayout mit auf den Weg geben, wodurch er sofort einsatzbereit ist.

möchten, dann wählen Sie etwa ganz oben „Raspberry Pi OS (32-bit)“ aus, für den Einsatz als Server ohne Monitor eignet sich das ressourcenschonende „Raspberry Pi OS Lite (32-bit)“, das Sie unter „Raspberry Pi OS (other)“ finden. Zahlreiche weitere Images wie Ubuntu oder für den Einsatz als Medienabspieler oder 3D-Drucker stehen ebenfalls zur Auswahl. Haben Sie eine Wahl getroffen, legen Sie im Hauptfenster über den Knopf „SD-Karte wählen“ fest, wohin das Image geschrieben werden soll. Beachten Sie, dass der Zieldatenträger komplett überschrieben wird.

Einstellungen auf Karte schreiben

Damit sind die Möglichkeiten der grafischen Oberfläche schon fast ausgereizt. Die erweiterten Optionen erreichen Sie nun entweder über die Tastenkombination Strg+Umschalt+X oder den Zahnrad-Knopf, der erst nach der Auswahl des Raspi-Betriebssystems erscheint.

Es öffnet sich eine lange Liste mit allem, was das Herz begehrt. Zunächst stellen Sie ganz oben unter „OS-Modifizierungen“ ein, ob Sie die Einstellungen nur einmalig („Nur für diese Sitzung“) oder dauerhaft („Immer verwenden“) mit auf die SD-Karte schreiben möchten. Die erste Option „Overscan deaktivieren“ ist dann interessant, wenn Sie einen Monitor an Ihren Raspi anschließen wollen: Sie schalten damit den schwarzen Rahmen ab, der standardmäßig rund um das ausgegebene Bildsignal angezeigt wird. Dieser soll garantieren, dass sämtliche Inhalte auf allen Bildschirmen angezeigt werden. Das ist insbesondere bei alten Röhrenmonitoren und -TVs relevant, bei modernen Displays hingegen können Sie auf den Trauerand verzichten.

Darunter legen Sie mit „Hostname“ einen individuellen Hostnamen fest. Standardmäßig horcht der Raspi auf den Namen „raspberrypi“, beim Einsatz mehrerer Raspis im gleichen Netz führt das jedoch schnell zu Verwirrung. Legen Sie am besten einen einsatzbezogenen Namen fest, etwa „raspi-smart-home“, „raspi-server“ oder raspikodi“. Wenn Sie Ihren Raspberry übers Netzwerk steuern möchten, dann wollen Sie wahrscheinlich „SSH aktivieren“ über das gleichnamige Häkchen, um den SSH-Server dauerhaft auf dem Standardport 22 zu aktivieren.

Standardmäßig ist auf dem System ein Nutzer „pi“ mit dem Passwort „raspberrypi“ eingerichtet, was natürlich alles andere als sicher ist, weil das jeder kennt. Sie sollten daher unter „Password zur Authentifizierung verwenden“ ein individuelles Passwort vorgeben. Selbst wenn Sie dieses Passwort für alle Ihre Raspis nutzen, ist das allemal sicherer als das vorgegebene Standardpasswort.

Login ohne Passwort

Wenn Sie auf Nummer sicher gehen wollen, dann nutzen Sie unterschiedliche Passwörter, wenn Sie mehrere Raspis betreiben. Noch besser ist die darunterliegende Option „Authentifizierung via Public-Key“. Damit fragt der Raspi nicht länger nach einem Passwort und Sie nehmen Angreifern somit die Chance, Ihr Passwort zu erraten. Zur Authentifi-

zierung wird stattdessen ein Krypto-Schlüsselpaar genutzt, dessen geheimen Schlüssel nur Sie besitzen. Der öffentliche Schlüssel muss im Raspi hinterlegt sein, damit die Anmeldung funktioniert.

Mit dem Imager-Tool ist das besonders einfach: Kopieren Sie den Inhalt der Schlüsseldatei (Public Key, der Dateiname lautet etwa „id_rsa.pub“) in das Feld „authorized_keys für 'pi'“. Falls das Public-Key-Verfahren neu für Sie ist und Sie Starthilfe benötigen, finden Sie in der offiziellen Raspberry-Dokumentation eine Anleitung, die erklärt, wie Sie mit ssh-keygen ein geeignetes Schlüsselpaar generieren (siehe ct.de/w924). Bei der Public-Key-Authentifizierung haben Sie die Wahl zwischen maximalem Komfort und maximaler Sicherheit: Wenn Sie beim Erstellen des Schlüsselpaars kein Passwort für den privaten Schlüssel setzen, dann können Sie sich auch ganz ohne Passwordeingabe am Raspi anmelden, solange sich der private Schlüssel auf Ihrem Rechner befindet, mit dem Sie die Verbindung herstellen. Für maximale Sicherheit können Sie den privaten Schlüssel mit einem Kennwort verschlüsseln. Sie müssen ihn in diesem Fall vor der Nutzung durch die Passwordeingabe entschlüsseln.

Aber zurück zum Raspi-Imager und seinen erweiterten Optionen. Deren Möglichkeiten sind nämlich noch längst nicht ausgereizt. Nachdem Sie sich um die Authentifizierung gekümmert haben, können

Sie Ihrem Raspi gleich darunter bei „Wifi einrichten“ eine WLAN-Konfiguration mit auf den Weg geben. Dies haben die Entwickler besonders charmant gelöst: Sofern Ihr PC gerade per WLAN mit dem Netz verbunden ist, versucht der Raspi-Imager selbstständig die Daten der gerade genutzten WLAN-Verbindung einzutragen, einschließlich des Passworts. Auf unserem Windows-System hat das auf Anhieb geklappt. Falls sich der Raspi mit einem anderen Netz verbinden soll, dann geben Sie Netzwerknamen und Passwort einfach von Hand ein.

Sie müssen unter dem Passwortfeld bei „Wifi-Land“ noch den Ländercode Ihres aktuellen Aufenthaltsorts einstellen (etwa „DE“ für Deutschland), damit der Raspi auch auf den jeweils erlaubten Frequenzen funkt. Beachten Sie bei der WLAN-Konfiguration, dass sich die Raspis erst ab dem Raspberry Pi 3 Model B+ mit 5-GHz-Netzen verbinden können. Bezieht sich also die durch das Tool vorgefüllte WLAN-Konfiguration auf ein 5-GHz-Netz, wird sich ein älteres Raspi-Modell nicht damit verbinden können. In diesem Fall können Sie zu einem WLAN-Stick greifen, der die eingegebenen WLAN-Parameter ebenfalls automatisch nutzen würde. Falls Sie Ihren Raspi per Netzkabel verbinden möchten, können Sie sich die WLAN-Einstellungen sparen. In diesem Fall sind keine weiteren Schritte nötig, um den Raspi ins Netz zu hieven.

**Der plappernde
Einrichtungssassistent
piwiz wird
überflüssig, wenn
Sie die Einstellungen
gleich mit auf
die SD-Karte
schreiben.**



Tastaturlayout einstellen

Weiter unten können Sie Ihren Raspi über die „Spracheinstellungen“ mit der passenden Zeitzone (in hiesigen Gefilden „Europe/Berlin“) sowie dem gewünschten Tastaturlayout (etwa „de“ für Deutsch) impfen. Darunter können Sie über „Einrichtungsassistent überspringen“ noch den grafischen Konfigurationshelfer piwiz aus dem Weg räumen, der sich möglicherweise beim ersten Start Ihres Raspi-Images zu Wort meldet. Er fragt im Wesentlichen die gleichen Einstellungen ab, die Sie gerade getätigt haben und ist somit potenziell überflüssig. Ist dieser in Ihrem Image nicht als Autostart eingerichtet, hat dieses Häkchen keine Auswirkungen – Sie machen damit aber auch nichts kaputt.

Am unteren Rand der Einstellungen finden Sie bei „Dauerhafte Einstellungen“ noch drei Häkchen, die sich nur auf Imager-Tool selbst beziehen: Sie können ein Tonsignal abspielen lassen, wenn die SD-Karte fertig ist und das Tool zudem anweisen, die SD-Karte nach dem Bespielen auszuwerfen, damit Sie die Karte angstfrei aus dem Speicherkartenleser ziehen können. Ist „Telemetrie aktivieren“ eingeschaltet, dann fließen Ihre Image-Downloads in die öffentlich einsehbare Statistik ein. Wenig überraschend machen dort derzeit die verschiedenen Raspbian-Editionen den Löwenanteil aus, allerdings hat auch Ubuntu auf dem Raspi an Beliebtheit gewonnen.

Image schreiben

Nach getaner Arbeit schließen Sie die erweiterten Einstellungen über „Speichern“ und schließen das Fenster, anschließend klicken Sie auf „Schreiben“, um das Image samt Voreinstellungen auf die zuvor gewählte Speicherkarte zu befördern. Das Schreiben dauert einige Minuten, abhängig von der Geschwindigkeit der Speicherkarte und des Kartenlesers. Bei der darauffolgenden Verifizierung überprüft der Imager, ob der Schreibvorgang fehlerfrei war. Dies dauert weitere Minuten. Falls Sie mal in Eile sind – und ein gesundes Grundvertrauen gegenüber Speicherkarte und Leser mitbringen – können Sie die Verifizierung auch abbrechen, um etwas Zeit zu sparen. Anschließend wartet der Raspi-Imager, bis die FAT32-Partition der Speicherkarte vom Betriebssystem gemountet wurde und schreibt dort schließlich die Voreinstellungen hinein.

Nach dem Schreiben stecken Sie die Speicherkarte wie gewohnt in den Raspi und versorgen ihn

anschließend mit Strom. Er wird jetzt mit der gewünschten Konfiguration booten. Haben Sie etwa WLAN und den SSH-Server eingerichtet, können Sie kurz darauf ohne weitere Handgriffe übers Netzwerk auf den Raspi zugreifen. Beachten Sie, dass der erste Start des Raspi-Betriebssystems stets etwas länger dauert, weil der Raspi noch ein paar Dinge erledigen muss, etwa die Anpassung der Partitionstabelle samt eines daran anschließenden Neustarts. Erst danach ist der SSH-Server erreichbar.

Ist der Raspi schließlich betriebsbereit, sollten Sie das System noch auf den aktuellen Stand bringen, um von sämtlichen verfügbaren Bugfixes und Sicherheits-Patches zu profitieren. Loggen Sie sich also per SSH in den Raspi ein und tippen die folgenden Befehle ein:

```
sudo apt update
sudo apt full-upgrade -y
```

Falls Sie Bildschirm und Tastatur angeschlossen haben, können Sie das direkt am Raspi erledigen. Dazu starten Sie das Terminal (schwarze Box in der Menüleiste am oberen Bildschirmrand). Die Aktualisierung nimmt je nach Alter des Images, Raspi-Generation und Internetanbindung einige Zeit in Anspruch.

Blick hinter die Kulissen

Das Imager-Tool kürzt die Inbetriebnahme der Raspis erheblich ab und man gewöhnt sich schnell an den Komfort. Aber was passiert da hinter den Kulissen? Wer schon dem ein oder anderen Raspi auf die Füße geholfen hat, der hat vermutlich mitbekommen, dass man bestimmte Dinge über die FAT32-formatierte Boot-Partition der SD-Karte einstellen kann.

Liegt hier eine Datei namens „ssh“, dann aktiviert das Raspberry-OS den SSH-Server und löscht die Datei anschließend. Auch eine WLAN-Konfiguration in Form einer wpa_supplicant.conf-Datei saugt der Raspi auf diese Weise auf. Unsere erste Vermutung war, dass der Imager einfach diese Mechanismen nutzt, um die Einstellungen auf der SD-Karte zu platzieren. Doch es gibt nicht für alle der angebotenen Voreinstellungen eine Automatik im Betriebssystem.

Eine weitere Möglichkeit wäre, die Linux-Partition zu modifizieren, doch dies wäre insbesondere unter Windows aufwendig, da es mit ext4-Partitionen nichts anfangen kann. Ein Blick auf die fertige

SD-Karte löste das Rätsel recht schnell: Auf der boot-Partition befinden sich zwei Dateien, die neuer als alle anderen sind: cmdline.txt und firstrun.sh.

In cmdline.txt sorgt der Eintrag `systemd.run=/boot/firstrun.sh` offenbar dafür, dass das Bash-Skript firstrun.sh beim Booten aufgeführt wird. Im Erfolgsfall ist ein Neustart vorgesehen: `systemd.run_success_action=reboot`

Die gesamte Magie steckt in dem Bash-Skript. Es ist gut lesbar und emsig bemüht, die Konfigurationswünsche des Raspi-Besitzers in die Tat umzusetzen. Los geht es mit der Änderung des Hostnamens, sofern eingestellt:

```
CURRENT_HOSTNAME=␣
  ↵cat /etc/hostname | tr -d " \t\n\r"
echo raspberrypi >/etc/hostname
sed -i "s/127.0.1.1.*$CURRENT_␣
  ↵HOSTNAME/127.0.1.1␣
  ↵raspberrypi/g" /etc/host
```

Das Skript fragt zunächst den aktuellen Hostnamen über `cat /etc/hostname` ab und merkt sich diesen in der Variable `CURRENT_HOSTNAME`. Danach schreibt es den neuen Namen, in diesem Beispiel `raspi-smartphone`, in `/etc/hostname` und ersetzt mit `sed` in `/etc/host` den alten durch den neuen. Das ist etwas komplizierter als nötig, dafür aber universell: Da zunächst der bisherige Hostname abgefragt wird, kann das Skript den Hostnamen auch dann ersetzen, wenn im Image ein anderer als `raspberrypi` vorgegeben ist. Dies ist bei Spezial-Images für Smart-Home-Anwendungen, Spieleemulation et cetera häufig der Fall. Den SSH-Server aktiviert das Skript recht unspektakulär über `systemctl enable ssh`. Interessanter wird da schon wieder die WLAN-Einrichtung:

```
cat >/etc/wpa_supplicant/␣
  ↵wpa_supplicant.conf <<WPAEOF
country=DE
ctrl_interface=DIR=/var/run/␣
  ↵wpa_supplicant GROUP=netdev
```



Sichern Sie sich jetzt 20%
Rabatt auf Ihre Bestellung

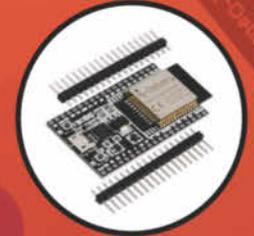
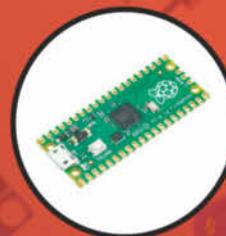
Stellen Sie jetzt Ihre
Business Anfrage



Az-Delivery

Ihr Experte für Mikroelektronik!

- Große Auswahl
- Kurze Lieferzeiten
- Business Konditionen
- Versand aus Deutschland



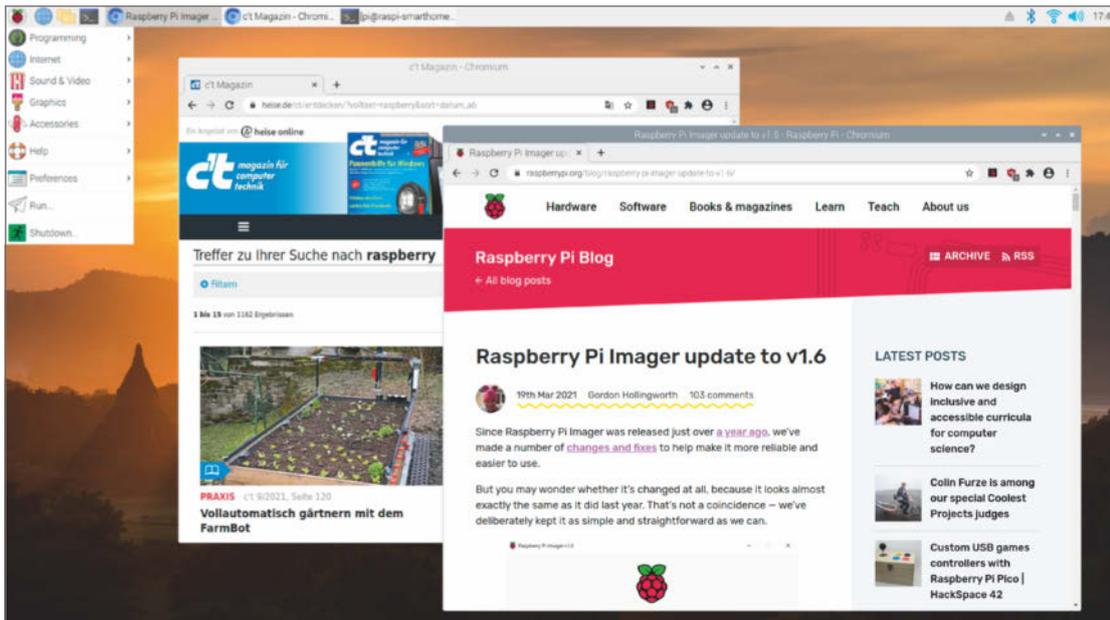
www.az-delivery.de



businesskunden@az-delivery.de

Ihr Rabattcode: CT-SOHE-RASPI
Das Angebot ist gültig bis zum 08.Mai 2022





Haben Sie den Raspi über den Imager vor-konfiguriert, können Sie beim ersten Start direkt lossurfen.

```
ap_scan=1
update_config=1
network={
    ssid="WLAN-Name"
    psk="WLAN-Passwort"
}
WPAEOF
chmod 600 /etc/wpa_supplicant/
    wpa_supplicant.conf
...
```

Das Skript schreibt die WLAN-Konfiguration mit `cat` in die Datei `/etc/wpa_supplicant/wpa_supplicant.conf` und setzt anschließend mit `chmod` die passenden Zugriffsrechte. Es fällt auf, dass das WLAN-Passwort im Klartext geschrieben wird, dies ist jedoch ein vertretbares Sicherheitsrisiko: Wer physischen Zugriff auf den Router oder einen der angemeldeten Clients hat, der kann es mit einer guten Wahrscheinlichkeit ohnehin im Klartext auslesen. Das neue Nutzerpasswort für den „pi“-User steht hingegen nur als Hash in dem Einrichtungsskript.

Dem Einrichtungsassistenten `piwiz` macht das Skript den Garaus, indem es die Autostart-Datei löscht:

```
rm -f /etc/xdg/autostart/piwiz.desktop
```

Ist die Datei nicht vorhanden, scheitert der Löschversuch schlicht. Es ist interessant und lehrreich, welche Lösungen die Entwickler des Raspi-Imagers gefunden haben und welche Bash-Tricks konkret zum Einsatz kommen. Aktivieren Sie doch einmal alle erweiterten Optionen und inspizieren Sie die daraus resultierende Datei `/boot/firstrun.sh` direkt nach dem Beschreiben der Speicherkarte.

Selbsterstörung

Die letzte Amtshandlung des Skripts ist übrigens die Selbsterstörung, anschließend entfernt es den Verweis auf sich selbst in `cmdline.txt`:

```
rm -f /boot/firstrun.sh
sed -i 's| systemd.run.*||g'
    /boot/cmdline.txt
```

So ist gewährleistet, dass das Skript nur einmalig beim ersten Start und nicht auch noch bei jedem Neustart ausgeführt wird.

Raum für Experimente

Der eingesetzte Mechanismus über das Startskript regt zu eigenen Experimenten an. Wer mit der Bash vertraut ist, kann das Skript leicht um weitere, indi-

viduell nötige Einrichtungsaufgaben ergänzen. Das Risiko hält sich in Grenzen: Da der Raspi ohnehin neu eingerichtet wird, steht nicht viel auf dem Spiel. Wenn das System nicht startet, unternimmt man einfach mit einem korrigierten Skript einen neuen Anlauf. Zur Fehlersuche können Sie in der Datei /boot/cmdline.txt behutsam den Befehl quiet entfernen, um die Kernelnachrichten beim Hochfahren live mitverfolgen zu können. In der Datei /var/log/boot.log können Sie den Bootvorgang zudem in aller Ruhe nachvollziehen.

Die Bash-Befehle können Sie vorher leicht auf einem bereits lauffähigen Test-Raspi ausprobieren. Führen die Befehle zum gewünschten Erfolg, können Sie Ihr Glück mit der Datei firstrun.sh versuchen. Wenn Sie häufig neue Raspis in Betrieb nehmen, kann Ihnen das viel Zeit sparen, da die Rechner auf Anhieb und ohne weitere Einrichtungsschritte so funktionieren, wie Sie es gerne hätten. Beachten Sie, dass während der Einrichtung möglicherweise noch nicht alle Ressourcen wie gewohnt zur Verfügung stehen, so etwa die Netzwerkverbindung über WLAN.

firstrun.sh statt Image

Eine weitere naheliegende Möglichkeit, den Raspi möglichst schnell in einen betriebsbereiten Zustand zu bringen, wäre, ihn wie gewohnt komplett einzurichten und anschließend ein Image von der Speicherkarte zu ziehen, das bei weiteren Einrichtungen einfach komplett auf die neue Karte geschrieben wird. Dies hat jedoch mehr als einen entscheidenden Nachteil: Das Betriebssystem wird auf einem alten Stand eingefroren. Zudem überspringen Sie mit dem Einsatz eines solchen Komplettabbilds die automatische Einrichtung, die dafür sorgt, dass die Partitionstabelle passend zur Speicherkarte geändert wird und individuelle Krypto-Schlüssel generiert werden.

Bei der Konfiguration per Startskript können Sie das Betriebssystem-Image hingegen jederzeit durch eine aktuelle Version oder gar eine andere Variante austauschen, solange sich die zu ändernden Ressourcen noch an denselben Stellen befinden. Auch spätere Konfigurationsänderungen sind so einfach möglich. (rei) **ct**

Raspi-Imager, SSH-Einrichtung und mehr

ct.de/w924



Heft + PDF
mit 29% Rabatt

Schlaues Gärtnern

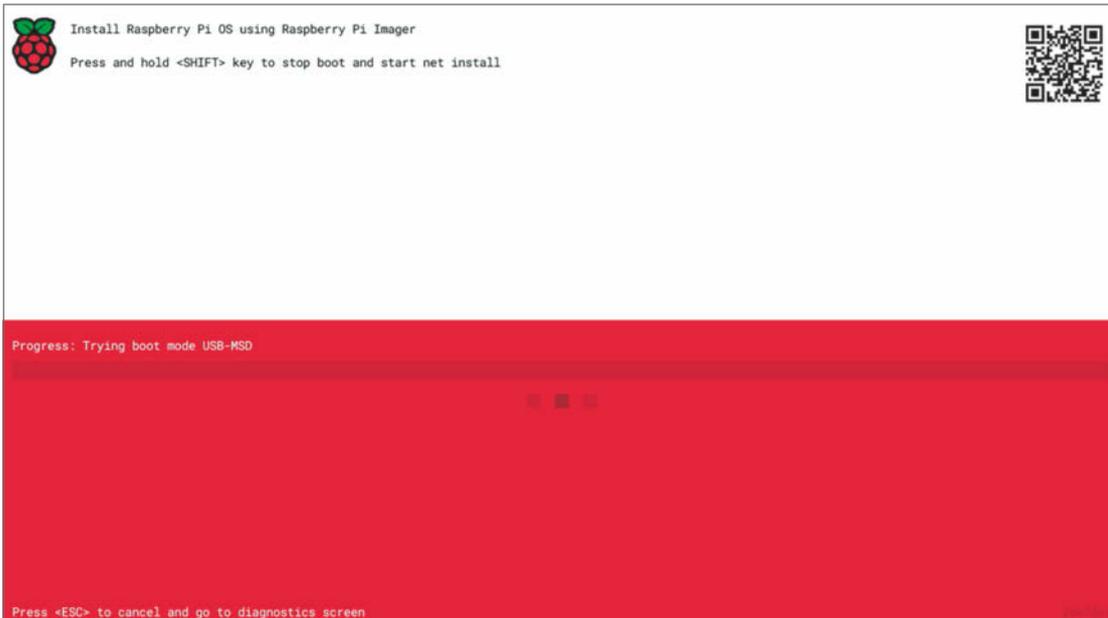
In diesem **heise online-Sonderheft** erfahren Sie, wie Ihnen smarte Helfer die Gartenarbeit abnehmen:

- ▶ Garten smart machen: Das brauchen Sie, bevor Sie starten
- ▶ Bewässerungssysteme: Sechs im Vergleich
- ▶ Intelligente Mähroboter: Fünf smarte Rasenmäher im Test
- ▶ **Auch im Set mit Gartenbewässerung WLAN Ventil zum Sonderpreis**

Heft für 14,90 € • PDF für 12,99 € • Bundle Heft + PDF 19,90 €



shop.heise.de/ho-smartgardening22



Autonome Linux-Installation

Fabrikneue Raspberry Pi sollen ihr Betriebssystem künftig ohne fremde Hilfe direkt aus dem Internet installieren können. Wir haben die Beta des neuen Netzwerk-Installers getestet und auf Angriffsmöglichkeiten hin abgeklopft.

Von **Mirko Dölle**

Auspacken, einschalten und schon betankt sich der Raspberry Pi selbst mit dem gewünschten Betriebssystem. Die Raspberry Pi Foundation hat eine Betaversion der künftigen Raspfi-Firmware veröffentlicht, die schon in wenigen Monaten ab Werk auf allen neuen Raspis installiert sein soll. Bei alten Raspis wird ein Update von Raspberry Pi OS die neue Firmware automatisch nachrüsten, sobald sie ins Stable Release aufgenommen wird.

Damit entfällt künftig der Umweg, Raspberry Pi OS auf einem anderen Rechner herunterladen und vorab

auf der MicroSD-Karte installieren zu müssen. Mit der neuen Firmware mit Netzwerkinstallation benötigen Sie künftig nur noch Tastatur, Maus, Monitor und eine Internetverbindung via LAN, damit der Raspi das gewünschte Betriebssystem selbst herunterlädt und auf einer MicroSD-Karte oder einem USB-Laufwerk installiert. Wir haben uns angesehen, was der neue Netinstaller leistet und welche Risiken damit verbunden sind.

Ein mit der neuen Firmware ausgestatteter Raspberry Pi prüft unmittelbar nach dem Einschalten, ob

sich im MicroSD-Kartenschacht ein bootbares Medium befindet. Falls nicht, erscheint ein Bootmenü, das den Benutzer auffordert, die Umschalttaste gedrückt zu halten, um die Netzwerkinstallation zu starten. Parallel dazu prüft die Firmware, ob sie an einem der USB-Anschlüsse oder im Netzwerk ein Bootmedium findet.

Dass die Netzwerkinstallation nicht automatisch anläuft, ist eine gute Versicherung, denn der Net-Installer ist nicht in der Firmware enthalten, sondern wird jedes Mal von einer in der Firmware hinterlegten URL (siehe ct.de/wrkm) nachgeladen. Würde jeder Raspi, der kein geeignetes Bootmedium vorfindet, vollautomatisch ein Programm aus dem Internet herunterladen und ausführen, könnten Angreifer etwa durch Spoofing der URL des Net-Installers unbemerkt unzählige Raspis übernehmen und in ein Botnetz verwandeln.

In einer früheren Beta-Version der Firmware tauchte der Dialog für die Netzwerkinstallation erst nach gut einer Minute auf, wenn sämtliche anderen Bootversuche fehlgeschlagen waren. Außerdem musste man dann nach der Umschalt-Taste auch noch die Leertaste betätigen. Diese Vorgehensweise war etwas komplizierter, aber auch sicherer, denn

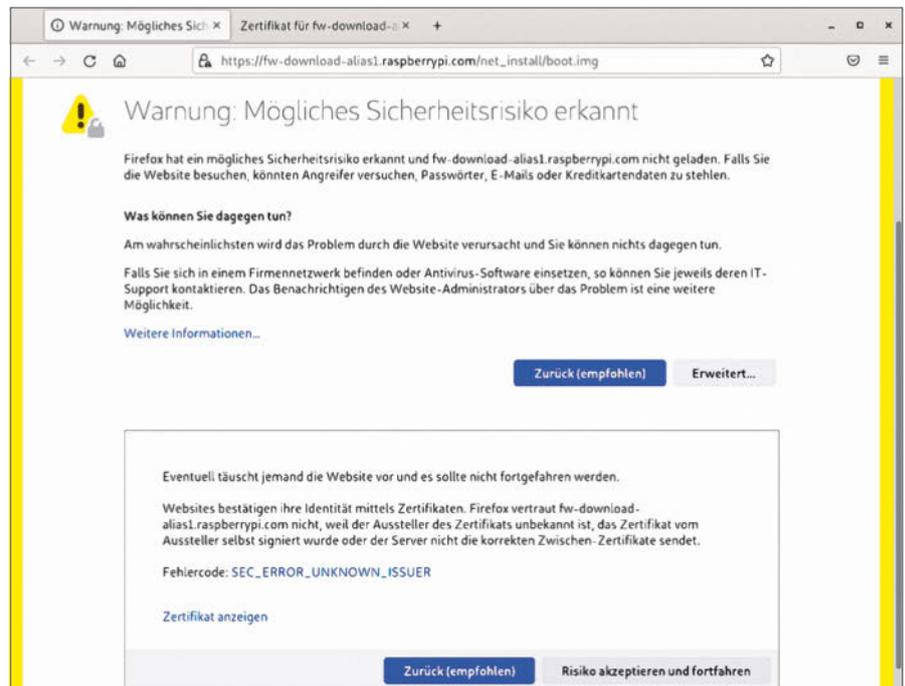
so lief die Installation nicht schon an, wenn versehentlich etwas auf der Tastatur lag.

Drum prüfe!

Wer die URL des Net-Installers im Browser aufruft, stolpert automatisch darüber, dass die Foundation ein selbst signiertes SSL-Zertifikat benutzt, dessen Authentizität der Browser nicht überprüfen kann. Dennoch haben Angreifer keine Chance, die Verbindung zu einem eigenen Server umzuleiten – die Firmware des Raspi überprüft den Fingerabdruck des Zertifikats und verweigert Verbindungen zu Servern mit anderen Zertifikaten. Zudem ist die Datei signiert.

Allerdings muss die Raspberry Pi Foundation peinlich genau darauf achten, dass ihr der private Schlüssel nie abhandenkommt – das Zertifikat wurde gleich mit 25 Jahren Gültigkeit ausgestellt. Um ein anderes einzutragen, müsste man wiederum die Firmware aller Raspis über ein Update von Raspberry Pi OS aktualisieren, aber natürlich auch die Firmware, die die Fabriken einspielen. Andererseits muss die Raspberry Pi Foundation so nicht darauf vertrauen, dass sich niemand über eine Certificate

Versucht man, den neuen Net-Installer im Browser herunterzuladen, stolpert man über ein selbst signiertes SSL-Zertifikat, dessen Authentizität der Browser nicht überprüfen kann. Die Raspi-Firmware hingegen kennt das Zertifikat und bemerkt Manipulationsversuche.



Authority ein Zertifikat erschleicht oder sie zwingt, ein gefälschtes Zertifikat auszustellen.

Mehr Komfort

Bei der nachgeladenen Datei handelt es sich um ein rund 23 MByte großes bootbares Dateisystem mit einem Mini-Linux und dem Raspberry Pi Imager. Nach etwa 30 Sekunden Lade- und Bootzeit, abhängig von der Internetverbindung, bootet das Mini-Linux aus einer RAM-Disk heraus und startet automatisch den Imager. Der hat den gleichen Umfang wie die Version aus dem Paket `rpi-imager` aus dem Repository von Raspberry Pi OS: Er bietet über einen Dutzend Varianten von Raspberry Pi OS, 32 und 64 Bit,

Ubuntu, Manjaro, LibreELEC, OSMC, RetroPie und Recalbox zur Auswahl an, aber auch, das Stable-Release sowie etwaige Beta-Varianten des Raspi-Bootloaders auf die MicroSD-Karte zu schreiben.

Das eröffnet künftig eine interessante Update-Möglichkeit für die Raspi-Firmware: Man benötigt nur noch eine leere Karte, bootet den Raspi, schreibt mit dem Net-Imager die neueste Firmware auf die Karte und bootet erneut. Anschließend kann man das neueste Raspberry Pi OS auf derselben Karte installieren. Übrigens lässt sich die MicroSD-Karte problemlos entfernen und wieder einstecken, der Raspi erkennt den Medienwechsel anstandslos. Ebenso unterstützt er es, Raspberry Pi OS auf einem USB-Datenträger zu installieren und anschließend

Selbst ausprobiert

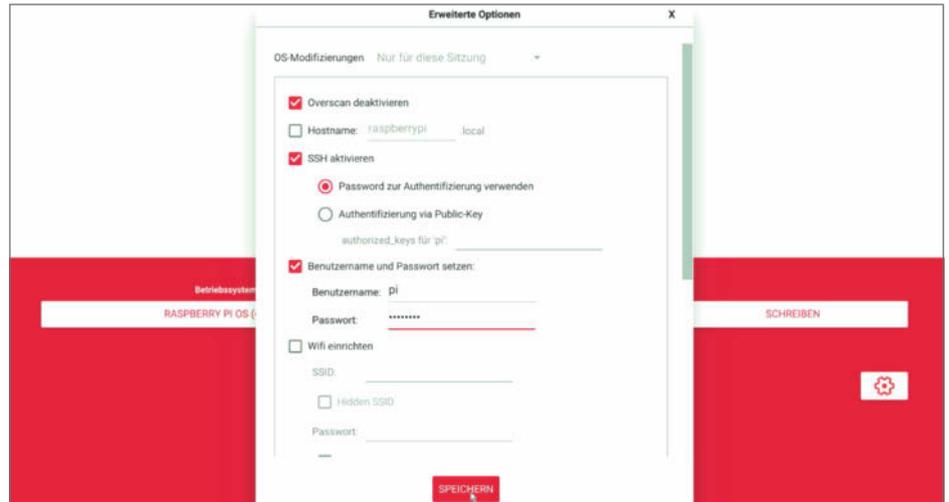
Um die Betaversion der Firmware mit dem neuen Bootloader auf Ihrem Raspberry Pi zu testen, verwenden Sie am besten den Raspberry Pi Imager und die MicroSD-Karte, auf der Sie anschließend Raspberry Pi OS installieren wollen. Den Imager finden Sie auf raspberrypi.com/software für Windows, macOS und Ubuntu zum Download. Haben Sie ihn installiert und aufgerufen, finden Sie die Beta-Firmware unter „Misc utility images“, „Beta Test Bootloader“. Typischerweise nimmt man die Firmware für „SD Card Boot“: Die versucht zuerst, von der SD-Karte zu booten, bevor sie andere Möglichkeiten ausprobiert.

Können Sie den Raspi Imager nicht verwenden, etwa weil Sie ein anderes Linux auf Ihrem PC installiert haben, finden Sie den Link zum Download des Zip-Archivs der Beta-Firmware unter ct.de/wrkm. Der Dateiname enthält unter anderem das Release-Datum und die Versionsnummer sowie am Ende das primäre Bootmedium – Standard ist „sd“ für die MicroSD-Karte. Haben Sie das Image aus dem Zip-Archiv entpackt, können Sie es mit `dd` oder einem beliebigen anderen Imager, etwa dem Balena Etcher von etcher.io, auf Ihre MicroSD-Karte übertragen.

Mit der auf der MicroSD-Karte installierten Beta-Firmware starten Sie Ihren Raspberry Pi anschließend, Monitor und Tastatur benötigen Sie dazu nicht. Nach einigen Sekunden beginnt die grüne LED des Raspi schnell zu blinken und zeigt damit ein erfolgreiches Update der Firmware an. Entfernen Sie die Karte wieder, damit der Raspi nicht beim nächsten Einschalten erneut davon bootet und die Firmware überschreibt – das Firmware-Update ist persistent, Sie müssen diesen Schritt bei jedem Raspi nur ein einziges Mal durchführen.

Sollte das Update fehlschlagen, immerhin befindet sich die Firmware noch im Beta-Stadium, wird Ihr Raspi dadurch aber nicht gleich zum Briefbeschwerer: Sie müssen dann lediglich die aktuelle Stable-Firmware einspielen. Dazu wählen Sie im Raspberry Pi Imager unter „Misc utility images“ den Menüpunkt „Bootloader“ und lassen dieses Image auf die gerade benutzte MicroSD-Karte schreiben. Das eigentliche Firmware-Update läuft dann genauso ab wie zuvor beschrieben, indem Sie den Raspi wiederum von der Karte booten und warten, bis die grüne LED rhythmisch blinkt. Danach ist Ihr Raspi wieder einsatzbereit.

Im Konfigurationsmenü lässt sich das neu installierte Raspberry Pi OS gleich konfigurieren und dort etwa SSH aktivieren, der Hostname und das Passwort ändern oder die WLAN-Zugangsdaten hinterlegen.



davon zu booten – egal, ob es sich dabei um einen USB-Stick oder eine USB-SSD handelt.

Über das Konfigurationsmenü gibt es außerdem die Möglichkeit, das neu installierte Raspberry Pi OS gleich einzurichten: den Hostnamen anpassen, SSH aktivieren, den Benutzernamen pi und sein Passwort ändern und ein WLAN konfigurieren. Allerdings gibt es keine Auswahl der vorgefundenen WLANs, die SSID muss von Hand eingegeben werden.

Fazit

Der neue Net-Installer erlaubt es künftig, den Raspberry Pi gleich nach dem Auspacken ohne fremde Hilfe und ohne einen zusätzlichen Computer einzurichten. Das ist nicht nur für Schulen ein Segen, wo

die Vorbereitung von dutzenden MicroSD-Karten entfällt, sondern senkt allgemein die Einstiegschwelle. Kenntnisse von Windows oder eines anderen Betriebssystems, um den Raspberry Pi Installer herunterzuladen und einzurichten, sind nicht länger erforderlich.

Es genügt, Tastatur, Maus, Monitor und ein Netzwerkkabel anschließen zu können, um den Raspberry Pi in Betrieb zu nehmen. Und das ist auch der einzige Wunsch, der noch offen bleibt: die Möglichkeit, die WLAN-Zugangsdaten beim Bootloader eingeben zu können, damit auch das Netzwerkkabel entfallen kann. Bis die neue Firmware mit Net-Installer auf allen fabrikneuen Raspis zum Einsatz kommt, dürfte es allerdings noch einige Monate dauern. (mid) **ct**

Download-URL Firmware
und Net-Installer

ct.de/wrkm

Mehr **wissen** –
besser **verstehen**

Heft + PDF
mit 29 % Rabatt



- ▶ Docker verstehen und richtig loslegen
- ▶ Mikrocontroller versus Mikroprozessoren
- ▶ Windows-Basics: Explorer, Dateisysteme, Registry
- ▶ Das eigene Netzwerk richtig ausrüsten

Heft für 14,90 € • PDF für 12,99 € • Bundle Heft + PDF 19,90 €



shop.heise.de/ct-knowhow22

Raspberry Pi

Zum beliebtesten Einplatinencomputer sammeln sich viele Fragen von interessierten Bastlern in unseren Postfächern. Wir haben Antworten auf die häufigsten zu Hardware, Betriebssystem, Kameramodul und Spannungsversorgung zusammengestellt.

Von **Jan Mahn, Pina Merkert und Peter Siering**



Das Betriebssystem

? **Raspberry Pi OS gibt es in einer 32- und einer 64-Bit-Variante. Wann brauche ich ein 64-Bit-Betriebssystem auf dem Raspi?**

! Das hängt vom verbauten RAM Ihres Raspi-Modells ab. Das klassische 32-Bit-Raspberry-Pi-OS kann die vollen 8 GByte RAM verwenden. Es kann einem einzelnen Prozess aber maximal nur 4 GByte spendieren. Anfang Februar 2022 ist die 64-Bit-Version fertig geworden und aus der Beta-Phase entlassen (zum Download über ct.de/wbxa). Damit darf ein Prozess den kompletten Arbeitsspeicher okkupieren.

? **Mit welchen Problemen muss ich rechnen, wenn eine Anleitung für eine Raspi-Basterei schon einige Jahre alt ist und sich auf eine ältere Version von Raspbian und nicht mal Pi OS bezieht?**

! Das hängt sehr davon ab, welche Funktionen ein Projekt nutzt. Mit jeder Version von Raspbian oder Pi OS beziehungsweise dem darunterliegenden Debian haben sich Details geändert. So sind seit einiger Zeit die Funkschnittstellen WLAN und Bluetooth standardmäßig deaktiviert. Wenn die Installationsroutinen das (noch) nicht berücksichtigen, kann man solche Konfigurationskniffe aber meist von Hand nachziehen, etwa mithilfe von `raspi-config` oder mit `rfkill unblock wifi`.

Ältere Anleitungen nennen obendrein oft Paketquellen, die überholt sind. In der Datei `/etc/apt/`

`sources.list` (und manchmal in den Unterverzeichnissen) steht, woher ein Raspi Pakete bezieht. Das derzeit aktuelle Pi OS trägt den Namen „Bullseye“, das vorherige „Buster“. Für beide bietet die Raspberry Foundation weiterhin gepflegte Paketquellen an.

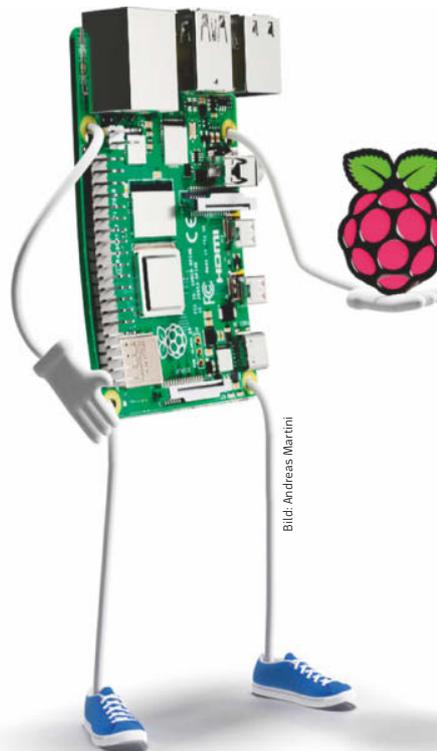


Bild: Andreas Martini

Für ältere Versionen (Stretch, Jessie ...) gibt es die nicht mehr. Damit ist die Gefahr groß, dass ein Projekt mit aktuellen Paketquellen „entgleist“.

Oft bedienen sich Projekte aber auch Quellen zukünftiger Debian-Versionen: „Unstable“ oder „Backports“. Wenn das bei einem älteren Projekt der Fall ist, stehen die Chancen gut, dass ein Rückgriff auf diese Quellen gar nicht mehr nötig ist. Ein Beispiel: Die c't-Artikel zum Einrichten eines WireGuard-VPN-Servers haben diese Quellen bemüht, weil WireGuard noch nicht in der regulären Distribution angekommen war. Heute kann man auf den Rückgriff auf „Unstable“ verzichten und die entsprechenden Paketquellen in der sources.list austragen.

? Warum gelingt es nicht, auf einem Raspi Zero (W) Pakete aus den Debian-ARM-Repositories zu beziehen?

! Die Debian-Entwickler stellen derzeit nur ARM-Pakete bereit, die auf Raspis mit CPU-Kernen des Typs Cortex-A funktionieren. Die Urversion des Raspi Zero (und alle Modelle bis 1B+) verwendet aber noch ARM11 als Kern und kann den Binärcode nicht ausführen. Ganz ohne ist es ohnehin nicht, die Pakete aus dem Debian-Repository in Pi OS zu mischen – gerade für Prototypen kann das aber ein pragmatischer Weg sein, an Software zu kommen, die es im Raspi-Universum noch nicht gibt.

Die Hardware

? Was ist dran, dass der Raspi SD-Karten frisst? Man liest immer wieder von Fällen, bei denen die Speicherkarten regelrecht verschleifen.

! In den Kindertagen des Kleincomputers stiegen SD-Karten bei intensivem Gebrauch gern irgendwann aus. Seit dem Umstieg auf das MicroSD-Format und schlichte Klemmsokkel ohne mechanische Gimmicks wie eine Feder (bis Raspi 2) haben solche Aussetzer nach unserer Erfahrung rapide abgenommen. Wenn Sie ausschließlich Karten renommierter Flash-Hersteller einsetzen, nicht die günstigste Klasse erwerben, dann halten die auch intensiver Belastung stand. Wir haben so bestückte Geräte über Monate intensiver Schreib- und Leselast ausgesetzt, etwa als Datenbankserver oder für Monitoringsysteme. Wenn Sie dennoch skeptisch sind: Moderne Raspis mit USB3-Ports können auch von einer dort angeschlossenen SSD starten. Sie können also auch ganz auf SD-Karten verzichten.

? Eignet sich ein Raspi für den dauerhaften Betrieb?

! Aus unserer nunmehr zehnjährigen Erfahrung mit zahlreichen Raspis aller Baureihen seit dem Raspberry Pi 1 können wir sagen: ja! Raspis dienen bei uns als Heimserver, Smart-Home-Zentralen, Sensordatensammler und Überwachungskamerarekorder. Sie hängen in feuchten Kellern und auf heißen Dachböden, ohne irgendwann hardwarebedingt auszusteigen. Wenn man auf eine funktionierende Smart-Home-Zentrale angewiesen ist, legt man sich am besten einen Zweit-Raspi in die Schublade, den man für gelegentliche Basteleien und als Ersatzgerät nutzen kann, wenn der Dauerläufer den nächsten Blitzeinschlag mit Spannungsspitze nicht überleben sollte.

? Was will mir der Blitz sagen, der immer wieder oben rechts auf dem Bildschirm erscheint?

! Wenn Sie den Raspi im Desktop-Betrieb mit angeschlossenem Bildschirm einsetzen und oben rechts einen Blitz sehen, mangelt es dem Kameraden an Energie – die Versorgungsspannung ist auf ein kritisches Level abgesunken. Das passiert, wenn das Netzteil unterdimensioniert ist, das Kabel vom Netzteil zur USB-Buchse aus zu dünnen Drähten besteht oder Sie zu viele Geräte per USB angeschlossen haben. Probieren Sie es mit anderer Spannungsversorgung oder schließen Sie energiehungriges USB-Zubehör über einen aktiven USB-Hub an, der eine eigene Spannungsversorgung mitbringt.

Falls Sie den Raspi per SSH als Server bedienen, sehen Sie den Blitz als Warnhinweis nicht. Um herauszufinden, ob Sie Spannungsprobleme haben, geben Sie `dmesg` ein und suchen in der Ausgabe nach dem Hinweis „Under-voltage detected!“ Systematisch sammeln können Sie entsprechende Einträge mit `dmesg | grep "Under-voltage"`.

? Welches Netzteil brauche ich?

! Die Raspis arbeiten mit der „USB-Spannung“ von 5 Volt und haben je nach Generation eine Micro-USB- oder USB-C-Buchse (Raspi 4) zur Speisung. Allerdings liefert nicht jedes beliebige USB-Netzteil oder Handy-Ladegerät genügend Strom, denn manche sind nur mit 0,5 Ampere belastbar. Das offizielle Micro-USB-Netzteil der Raspi-Entwickler liefert bis zu 2,5 A (12,5 Watt), die für den



Das mitgelieferte Flachbandkabel der Raspberry-Kamera ist kurz. Ist es für ein Projekt sinnvoll, kann man es durch längere Flachbandkabel austauschen – weil sie ungeschirmt sind, sind Störungen je nach Umgebung möglich.

Raspi 4 gedachte USB-C-Version sogar 3 A (15 Watt). Dermaßen viel Strom ist aber nur nötig, wenn man viel (USB-)Zubehör an den Raspi anschließt oder etwa auch ein Display speist. Für einfache Projekte und einen Raspi Zero kann ein schwächeres Netzteil genügen, mindestens 1 A Belastbarkeit (5 Watt) sind aber ratsam – sonst wird man, wie oben beschrieben, das Blitzsymbol aufleuchten sehen.

Raspi 4 der ersten Fertigungscharge haben einen Fehler beim USB-C-Anschluss: Es fehlt ein Widerstand, der nach USB-Standard vorgeschrieben ist. In der Folge können ausgerechnet hochwertige, sogenannte intelligente USB-C-Kabel („Electronically Marked Cables“) nicht wie vorgesehen Versorgungsspannung und -strom mit dem USB-C-Netzteil aushandeln. Billige USB-C-Kabel ohne integrierte Chips oder das Netzteil der Raspberry Pi Foundation schaffen Abhilfe.

? Der Raspi hat Löcher zur Befestigung, was für ein geplantes Projekt sehr hilfreich ist. Kann ich ihn mit M3-Schrauben fixieren?

! In die Löcher an den Ecken vom Model B passen leider keine M3-Schrauben. Direkt an den Löchern verlaufen aber keine Leiterbahnen, weshalb wir die Löcher schon erfolgreich auf 3 Millimeter

aufbohren konnten. Eine schönere Lösung sind aber Schrauben der Größe M2,5, die perfekt passen.

Die Kamera

? Neben dem offiziellen Kameramodul RPI-CAM-V2 gibt es zahlreiche Kameras von anderen Anbietern, die als Raspi-tauglich beworben werden. Funktionieren die?

! Treiberseitig gibt es da keine Probleme. Die Stecker sind aber nicht immer gleich breit. Einige nutzen den schmaleren Anschluss vom Pi Zero, andere den breiten vom Model B. Es gibt aber Adapterkabel zum Nachkaufen.

? Ich plane ein Projekt mit einem Kameramodul, die Kamera soll aber nicht direkt am Raspi hängen. Muss das Kabel der Raspi-Kamera so kurz sein?

! Nein. Es gibt längere Kabel zum Austauschen. Wir haben schon 1,5 Meter lange Kabel getestet und konnten keine Nachteile feststellen. Einstreuungen sind bei ungeschirmten langen Kabeln aber immer möglich, weshalb Sie vor Ort testen sollten, ob ein langes Kabel noch funktioniert. (jam)

Raspberry Pi OS
ct.de/wbxa

Für Maker und Nerds

 shop.heise.de/shop-maker



ParkLite

ParkLite denkt mit. Die elektronische Parkscheibe stellt automatisch nach ca. 20 Sekunden die Parkzeit ein. Damit ist Schluss

mit Bußgeldern! Hitze- und kältebeständig, inklusive Reinigungstuch und Klebepads.

29,90 €



Aluminium-Case FLIRC

Das hochwertige Gehäuse aus stabilen

Aluminium ist ideal, um den Raspberry Pi 4 als Media Center zu verwenden. Das elegante Design integriert sich optimal in jede Wohnumgebung. **Auch im Set mit Raspi 4 Model B 2GB erhältlich.**

23,90 €



musegear® finder Version 2

Finden Sie Schlüssel, Handtasche oder Geldbeutel bequem wieder statt ziellos zu

suchen. Mit dem Finder können Sie z.B. das Smartphone klingeln lassen oder Wertgegenstände einfach tracken und noch mehr.

24,90 €

NEU



Raspberry Pi Pico Starter Kit

Mit Zubehör: Board, Micro-USB-Kabel, Stiftleisten, Breadboard, Jumperkabel.

Raspberry Pi Pico ist ein kostengünstiges, leistungsstarkes Mikrocontroller-Board mit flexiblen digitalen Schnittstellen.

19,90 €



Make Family + Makey-Paket

Das PDF-Magazin mit 21 Anleitungen zum kreativen Basteln mit Kids auf über 200 Seiten.

Dazu: Makey-Plüschroboter und Makey-Lötbausatz mit LEDs und Batterie.

~~27,70 €~~ 19,90 €



PokitMeter – Multimeter, Oszilloskop und Logger

PoKit misst, zeigt und protokolliert eine Vielzahl von Parametern wie Spannung, Strom, Widerstand und Temperatur mittels Verbindung via Bluetooth mit Ihrem Smartphone oder Tablet.

94,90 €



Joy-IT OR750i: Freifunk- & OpenWrt-Dual-Band-Router

Der Einstieg in die Freifunk- und OpenWrt-Welt kann oft schwierig sein. Deshalb hat Joy-IT in Zusammenarbeit mit Freifunk Hannover und c't den OR750i entwickelt.

Dank Webinterface kann man beliebige Firmwares einfach hochladen – ohne komplizierte Kommandos oder inkompatible Hardware-Revisionen; ideal für OpenWrt-Einsteiger und solche, die Freifunk einfach nur nutzen wollen.

39,90 €

BEST-SELLER



Tasse „Kein Backup? Kein Mitleid!“

Wer kennt es nicht? Die lieben Kollegen haben wieder mal die Datensicherung vergessen und betteln bei Ihnen in der IT-Abteilung um Hilfe. Sie denken sich dabei nur:

„Kein Backup? Kein Mitleid!“ Platzieren Sie die schicke Keramiktasse, außen matt-

schwarz und innen rot, einfach demonstrativ auf Ihrem Schreibtisch. Der praktische Holzdeckel kann gleichzeitig als Untersetzer verwendet werden und verhindert somit nervige Kaffeeflecken auf dem Schreibtisch.

17,90 €



shop.heise.de/shop-maker

PORTOFREI AB 20 € BESTELLWERT

Raspberry Pi für den Alltag

Als Bastelplattform hat der Raspi einen festen Platz nicht nur in den Herzen der Maker. Wer professionell in der IT arbeitet, lernt den Minicomputer schnell für besondere Angelegenheiten zu schätzen. Mal ersetzt er teure Profitechnik und mal erfüllt er Aufgaben, für die es keine fertigen Lösungen gibt.

Von **Peter Siering**



| | |
|--------------------------------------|----|
| Raspi statt Profi-Technik | 42 |
| Passwortmanager in Eigenregie hosten | 44 |
| Backups mit Syncthing | 52 |
| Syncthing-Backups einrichten | 56 |
| Raspi meldet Angriffe auf Ihr WLAN | 64 |
| Fernbedienung fürs Game-Streaming | 72 |
| Octoprint bringt 3D-Drucker ins Netz | 78 |

Die Liste der skurrilen Projekte aus der Maker-szene rund um den Raspi ist lang: Witzig ist zum Beispiel eine Ausgabe der Lobbyistenaktivität in den USA über ein analoges Zeigerinstrument. Jüngst geriet ein Projekt in die Schlagzeilen, das einen Raspi per Laser Mücken eliminieren ließ. Sie finden diese und alle im Folgenden vorgestellten Projekte unter ct.de/wt8e. Die in den folgenden Artikeln vorgestellten Projekte erfüllen höhere Ansprüche und sind ausgesprochen nützlich. Sie wandeln den Minicomputer in ein Utensil für IT-Administratoren und -Enthusiasten. Dieser Artikel liefert Inspiration, was Sie mit Ihren Raspis noch anstellen könnten.

Ad-hoc starten

Dass ein Raspi für ITler, Netzwerkbetreuer und PC-Enthusiasten attraktiv ist, liegt daran, dass er wenig kostet und leistungsmäßig an einen PC von vor einigen Jahren heranreicht. Je nach Version verbrät er trotzdem nur wenig Energie. Obendrein hat er viele Schnittstellen, um ihn mit anderen Geräten zu verbinden, zum Beispiel Temperatursensoren. Er eignet sich etwa als günstige Messsonde, die sich problemlos und sicher mit einem bestehenden Netzwerk verknüpfen lässt.

Hinzu kommt ein stabiles Linux-Betriebssystem, das eine Fülle von Standards unterstützt. Software sprudelt aus einem schier unerschöpflichen Vorrat. Die Dokumentation profitiert von fleißigen Communities. So genügt für die Inbetriebnahme gängiger Projekte für den Raspi eine SD-Speicherkarte mit der Lite-Ausgabe von Raspberry Pi OS. Typischerweise übernehmen einfache Shell-Skripte die Regie beim Einrichten der oft anspruchsvollen Software-Stacks der Raspi-Projekte. Der Nutznießer ist der ITler, der binnen Minuten ein nützliches Gerät in der Hand hält, zum Beispiel mit Pi-hole oder AdGuard. Sie filtern Werbung und Malware nicht nur für einen einzelnen PC, sondern auch für kleine Netze.

Im Windschatten anderer erfolgreicher Projekte wie zum Beispiel Nextcloud wachsen Ableger heran, die ihre Macher auf den Raspi-Betrieb anpassen. Das erleichtert nicht nur die Inbetriebnahme, sondern auch den dauerhaften Einsatz. Obendrein bringen solche Projekte oft Extras mit, die man sonst von Hand in eine selbst gepflegte Installation einbauen müsste. Meist sind die Projekte leicht im Internet zu finden: Hängen Sie einfach „pi“ vor oder hinter einschlägige Namen, wie bei Nextcloudpi oder PiVPN.

Eine weitere Quelle für nützliche Fertigsoftware für den Raspi erschließt sich der Einplatinencomputer über Container. Wenn erst mal Docker eingerichtet ist (die nötigen Schritte dazu erklärt am Beispiel Bitwarden der Artikel ab Seite 44), steht die gesamte Docker-Welt offen: Die E-Mail-Stacks Mailu und mailcow sind Beispiele für solche Docker-Umgebungen, die Raspis im Handumdrehen in einen vollwertigen E-Mail-Server verwandeln.

Ausreizen

Für passionierte ITler liegt der besondere Reiz des Raspis jenseits fertiger Projekte: Mit einer Minimalinstallation plus VPN-Client und SSH-Server schaffen sie sich vorübergehend ein Sprungbrett in fremde Netze. Sie werfen den Raspi dort ab. Der VPN-Client verbindet sich automatisch mit einer Gegenstelle unter Ihrer Kontrolle. Per SSH springen Sie dann ins fremde Netz.

Bei SSH führen mehrere Wege zum Ziel: Es kann einzelne TCP-Ports durch einen Tunnel weiterleiten. SSH kann aber als SOCKS-Proxy auch das fremde Netz zum Gateway für den lokalen Browser oder andere Dienste machen. Das ist sehr nützlich, wenn man dort Geräte warten soll und deren Webinterface nicht ins Internet öffnen möchte. Solcherlei Treiben setzt freilich das Wissen und Einverständnis des Betreibers des fremden Netzes voraus.

Ein Raspi eignet sich als kleiner Server, wie openmediavault als Projekt für den NAS-Selbstbau zeigt. Der Kleincomputer kann aber auch Protokollbrücken für Dateidienste auf Basis von SMB, also Windows-Freigaben bilden. So kann er veraltete, nur SMB1-taugliche Geräte gefahrlos an ein längst auf SMB2 umgestelltes Netz ankoppeln: Der Raspi spricht dabei in einem privaten Netz mit dem gefährdeten SMB1-Altgerät. Er reicht dessen Dateien schließlich per SMB2 oder 3 ins externe Netz weiter. Ins private Netz gerichtet arbeitet er als Server, ins externe Netz als Client.

Generell glänzt der Raspi im Netz: Mit WLAN- und Ethernet-Schnittstelle ausgerüstet kann er als Router zwischen Funk- und Drahtnetz vermitteln. Oder Sie lassen ihn als WLAN-Access-Point arbeiten, der eine Brücke zwischen den beiden Medien spannt. Seine Stärke dabei liegt nicht beim Durchsatz, sondern in den Filterfähigkeiten: Per Firewall lässt sich fein dosieren, was durchkommt. Ebenso fein lässt sich auch beobachten, was Geräte untereinander austauschen oder was ins Internet abfließt. Der Raspi ist eben mehr als ein Spielzeug für Admins. (ps) **ct**

**Erwähnte Projekte im Web,
Einführungen in die
Raspi-Welt aus c't**

ct.de/wt8e

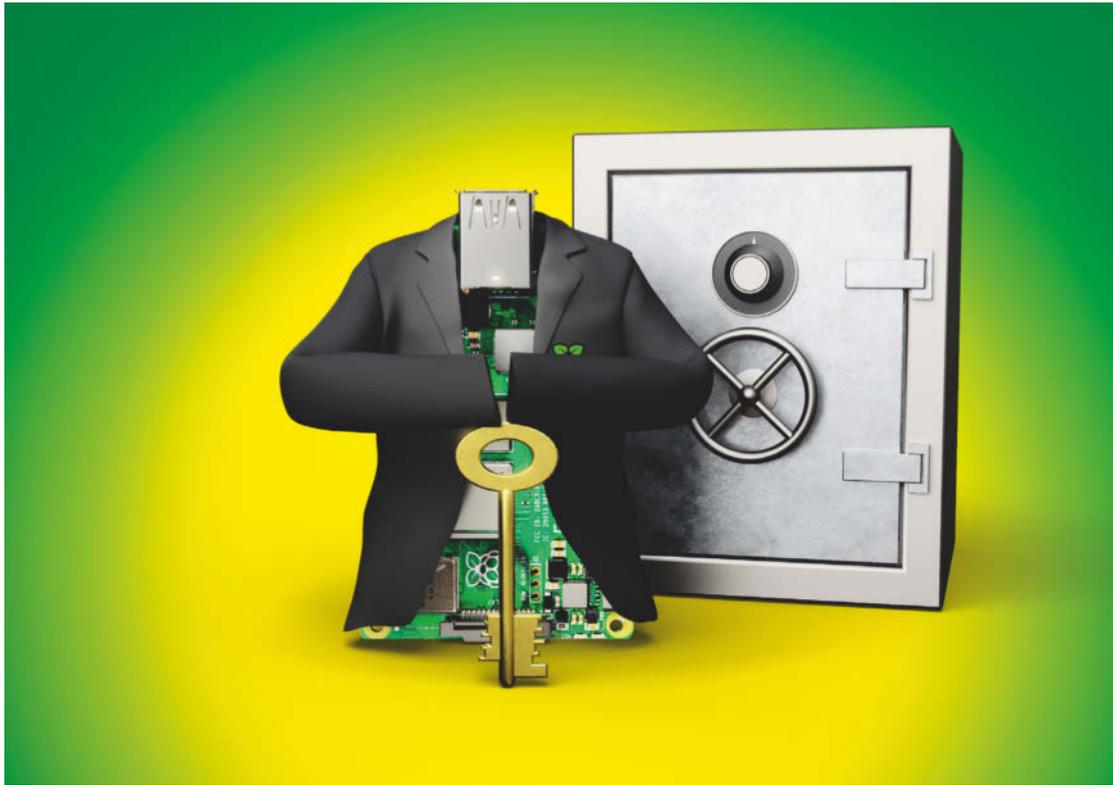


Bild: Andreas Martini

Passwortmanager in Eigenregie hosten

Mit Vaultwarden betreiben Sie Ihren eigenen Passwort-Tresor und -Server. Es synchronisiert Zugangsdaten zwischen mehreren Geräten, ohne Ihre Datenbank in fremde Hände zu geben. Das klappt auf einem (Heim-) Server oder mit dem Raspberry Pi.

Von **Niklas Dierking**

Ein neues Kundenkonto hier, ein Social-Media-Login da. Bei vielen Internetnutzern türmen sich Zugangsdaten zu Portalen und Diensten zu einem bedrohlichen Haufen auf. Bei jeder Registrierung ein einzigartiges und sicheres Passwort zu

überlegen und jedes einzelne zu merken, das ist eine schier unlösbare Aufgabe. Passwortmanager versprechen Abhilfe, aber sie hängen üblicherweise an der Cloud des Anbieters und synchronisieren die Passwortsammlung zwischen mehreren Geräten

nur, wenn man einen monatlichen Obolus zahlt. Außerdem brauchen Nutzer ein hohes Maß an Vertrauen in den Betreiber: Der virtuelle Schlüsselbund liegt immerhin in einem fremden Schließfach.

Den Open-Source-Passwortmanager Bitwarden können Sie in Eigenregie hosten. Bitwarden besteht aus zwei Teilen: dem Server, der Passwörter verwaltet und bereitstellt, und den Clients, die sie abrufen. Das Community-Projekt Vaultwarden hat eine schlankere Variante des offiziellen Bitwarden-Servers geschaffen, die das Bitwarden-API implementiert. Damit kann ihr (Heim-)Server oder ein Raspberry Pi die Rolle des Passwort-Servers übernehmen, der zu den offiziellen Bitwarden-Anwendungen kompatibel bleibt. Nach der Installation können Sie mit mehreren Accounts über die offiziellen Bitwarden-Clients oder mittels Weboberfläche („Web-Vault“) auf den Server zugreifen. Die Lösung eignet sich damit auch für Familien, Freunde und kleine Teams.

Für Vaultwarden empfiehlt sich eine Docker-fähige Linux-Distribution, beispielsweise ein Ubuntu LTS oder Raspberry Pi OS. Falls Sie den Passwort-Server auf dem Raspi aufsetzen, brauchen dazu mindestens einen Raspberry Pi 3. Ob Sie das System mit grafischer Oberfläche an einem Monitor mit Maus und Tastatur bedienen oder sich mittels SSH von einem anderen Gerät aufschalten, bleibt Ihnen überlassen.

Für die Installation braucht es lediglich den Zugriff auf die Kommandozeile. Nach der Einrichtung können Sie Vaultwarden mit einem Web-Browser von einem beliebigen Gerät aus konfigurieren. Die Weboberfläche enthält eine Web-App („Vault“), in der Nutzer ihren Account verwalten, und stellt Werkzeuge zur Administration über ein Admin-Panel bereit. Die Entwickler von Vaultwarden empfehlen, den Server als Docker-Container aufzusetzen. Diese Lösung hat den Vorteil, dass der Passwortmanager anderen Diensten auf dem System nicht in die Quere kommt.

Als Zutaten für eine solche Installation haben sich die folgenden Komponenten bewährt: Ein Reverse-Proxy ist der bequemste Weg, die Weboberfläche und das API verschlüsselt ins Web zu bekommen. Der Proxy nimmt Anfragen entgegen und liefert die Daten über eine HTTPS-Seite aus. In den nachfolgenden Beispielen wird angenommen, dass die TCP-Ports 80 und 443 des Systems nicht bereits belegt sind. Für diesen Zweck eignet sich der containerisierte Edge-Router Traefik, der sich die nötigen TLS-Zertifikate selbst besorgt und automatisch erneuert.

Docker-Compose erspart lange `docker run`-Befehle. Die Konfigurationen der Container sind dabei in einer Compose-Datei zusammengefasst, die durch eine `.env`-Datei mit Umgebungsvariablen ergänzt wird. Beispieldateien stellen wir in einem GitHub-Repository zur Verfügung, das Sie über [ct.de/w7gs](https://github.com/ctde/w7gs) einsehen können.

Wenn Sie Vaultwarden im Heimnetz betreiben und von außen erreichen wollen, benötigen Sie einen festen Hostnamen, der auf die WAN-IPv4-Adresse des Routers verweist. Hier hilft ein DynDNS-Dienst. Dieser leitet Anfragen, die beim konfigurierten DynDNS-Hostnamen eingehen, an die jeweils aktuelle WAN-IPv4-Adresse des Routers weiter. Eine Anleitung für den kostenlosen DynDNS-Anbieter FreeDNS finden Sie online im Artikel „Mit DDNS auf’s Heimnetzwerk zugreifen“ (ct.de/w7gs). Sie haben die Wahl, welchen DynDNS-Dienst Sie verwenden. In dieser Anleitung dient die Adresse `bitwarden.example.com` als Beispiel.

Das Betriebssystem vorbereiten

Im ersten Schritt bringen Sie das System auf den neuesten Stand und installieren mit einem Skript die Docker-Engine für die Container-Laufzeitumgebung. Wenn es sich um eine frische Raspberry-Pi-OS-Installation handelt, dann starten Sie das

Der Login-Bildschirm der Bitwarden-Weboberfläche („Web-Vault“). Nach der Installation von Vaultwarden genügt beispielsweise ein Browser, um jederzeit Zugriff auf Ihre Passwörter zu haben. Um den Passwortmanager zu benutzen, sind keine Spezialkenntnisse mehr nötig.

System nach dem Einspielen der Updates neu, bevor Sie Docker installieren.

```
sudo apt update
sudo apt upgrade
sudo reboot
curl -fsSL https://get.docker.com \
-o get-docker.sh
sudo sh get-docker.sh
```

Falls auf Ihrem System die Apache2-Utilities und das Versionsverwaltungswerkzeug Git nicht vorhanden sind, dann installieren Sie die Pakete ebenfalls.

```
sudo apt install apache2-utils
sudo apt install git
```

Docker-Compose beziehen Sie am besten aus dem Python-Paketmanager „pip“.

```
sudo apt install python3-pip
sudo pip3 install docker-compose
```

Eine Einführung in die Funktionsweise von Docker-Compose finden Sie im Artikel „Container-Komponente“ [1]. Um mit dem Docker-Daemon zu interagieren, brauchen Sie Systemverwalterrechte. Wenn Sie Docker-Befehle ohne vorangestelltes sudo verwenden möchten, müssen Sie Ihren Benutzer zur Gruppe „docker“ hinzufügen und sich erneut einloggen.

```
sudo usermod -aG docker user
passwd
```

Die nötige Software ist jetzt installiert und Docker einsatzbereit. Bevor es jedoch an das Einrichten der Container geht, muss sichergestellt werden, dass ihr Server – und damit Weboberfläche sowie API – auch aus dem Internet erreichbar ist. Wenn Sie Vaultwarden auf einem angemieteten Server betreiben wollen, überspringen Sie den nächsten Schritt.

Netzwerkverkehr dirigieren

Damit Anfragen zuverlässig beim Passwort-Server und nicht bei einem anderen Netzwerkgerät landen, sollte eine feste IP-Adresse vergeben werden. Handelt es sich um einen Raspi, dann tragen Sie diese für das aktive Netzwerk-Interface in der Datei /etc/dhcpd.conf ein. Wenn der Raspi via Ethernet mit dem Netzwerk verbunden ist, dann wählen Sie das

Interface eth0 (bei WLAN entsprechend wlan0). Passen Sie in der Datei die folgenden vier Zeilen an Ihr lokales Netzwerk an, etwa so:

```
interface eth0
static ip_address=192.168.0.184/24
static routers=192.168.0.1
static domain_name_servers=192.168.0.1
```

Starten Sie das System mit `sudo reboot` im Anschluss neu, damit die Änderungen wirksam werden. Richten Sie nun im Web-Interface Ihres Routers eine Port-freigabe (Port-Forwarding) ein. Anfragen von außen, die an TCP-Port 80 oder 443 eingehen, sollen an den Server durchgereicht werden und beim Reverse-Proxy Traefik landen. Als Protokoll wählen Sie TCP und tragen Port 80 ein. Der Name, der für den Dienst vergeben wird, ist beliebig. Den gleichen Vorgang wiederholen Sie für den Port 443.

Damit der DynDNS-Dienst Ihrer Wahl in der Lage ist, auf die aktuelle WAN-IPv4-Adresse des Routers zu verweisen, benötigt er einen Gegenpart, der ihn mit dieser Information versorgt. Diese Aufgabe können die meisten modernen Router übernehmen. Die Konfiguration erledigen Sie ebenfalls im Webinterface des Routers. Wählen Sie im entsprechenden Menü den Anbieter aus und tragen Sie die abgefragten Daten, meist Login-Informationen und eine Update-URL, in die Maske ein. Wird Ihr DynDNS-Anbieter nicht gelistet, dann setzen Sie einen benutzerdefinierten Eintrag.

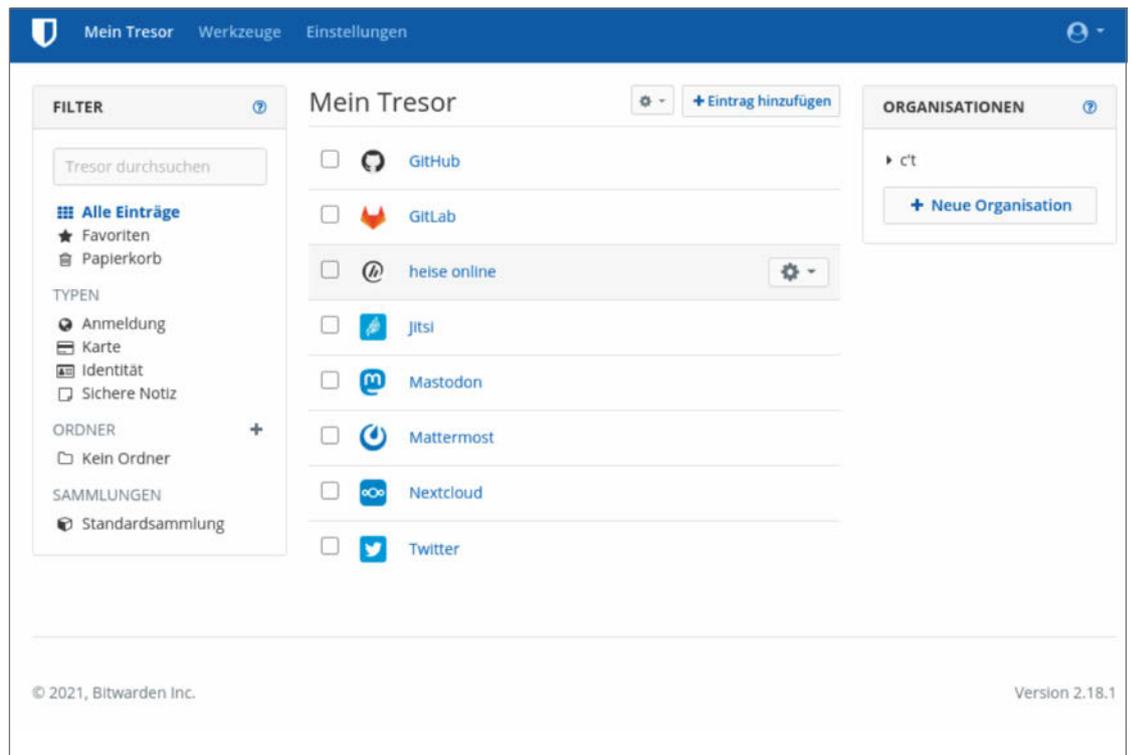
Transportverschlüsselung dank Traefik

Nach diesen Vorbereitungen geht es nun an die Container-Architektur. Als Speicherort für Container-Konfigurationen und Compose-Files auf dem Linux-Dateisystem hat sich das Verzeichnis /srv bewährt. Wenn Sie dort Dateien (be)schreiben wollen, müssen Sie mit sudo arbeiten oder sich Systemverwalterrechte einräumen. Klonen Sie das GitHub-Repository nach /srv und wechseln Sie danach in den Ordner.

```
sudo git clone \
https://github.com/ndi-ct/raspi-pwm \
/srv/raspi-pwm
cd /srv/raspi-pwm
```

Als Container-Blaupause steht im heruntergeladenen Repository eine YAML-Datei bereit. Die schreibt

Willkommen im Vault: In der Web-App können Sie Ihre Passwörter verwalten und Einstellungen vornehmen, beispielsweise Zwei-Faktor-Authentifizierung aktivieren.



fest, welche Container-Images aus dem Docker Hub geladen werden und wie diese konfiguriert sind. Umgebungsvariablen sind in die zugehörige .env-Datei ausgelagert. Diese wird standardmäßig nicht angezeigt. Bei Bedarf verschaffen Sie sich mit `ls -la` einen Überblick. Der Reverse-Proxy benötigt darüber hinaus noch die Dateien `traefik.toml` und `dynamic.yml`, die Anweisungen zur Beschaffung von Zertifikaten und zur Regelung des Netzwerkverkehrs geben. Diese finden Sie im Ordner `config-files`.

Öffnen Sie `traefik.toml` mit dem Texteditor Ihrer Wahl und tragen Sie in der Zeile `email = "mail@example.com"` eine gültige Mail-Adresse ein (die Anführungszeichen bleiben bestehen). An diese Adresse sendet Let's Encrypt Warnungen, falls es Probleme mit dem erteilten Zertifikat gibt. Traefik legt die Zertifikate, die es automatisiert mittels ACME-Protokoll besorgt, in der Datei `acme.json` ab. Diese darf nur für Root lesbar sein. Erstellen Sie im Ordner `config-files` eine leere Datei mit dem Namen `acme.json` und passen Sie die Rechte an.

```
sudo touch acme.json
sudo chmod 600 acme.json
```

Danach legen Sie ein virtuelles Netzwerk für den Reverse-Proxy an.

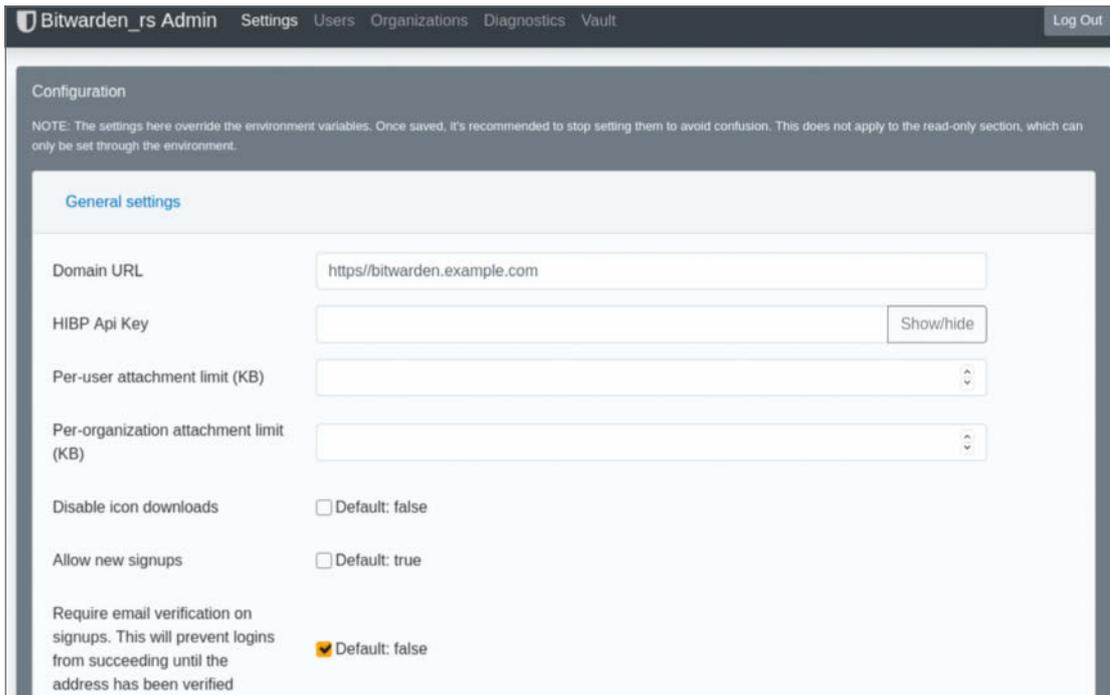
```
sudo docker network create \
traefik_reverse_proxy
```

Tresor aufstellen

Wechseln Sie jetzt in das Verzeichnis `/srv/raspi-pwm`. Um sich einen Überblick über den Container-Aufbau zu verschaffen, können Sie sich den Inhalt des `Compose-File` ausgeben lassen:

```
cat docker-compose.yaml
```

Die Images „`vaultwarden/server`“ und „`traefik`“ sind mit Versionsnummern versehen, um Ihnen unschöne Überraschungen zu ersparen. Ohne dieses Tag lädt



Das Admin-Panel ist die Schaltzentrale von Vaultwarden. Hier haben Sie Zugriff auf wichtige Einstellungen und können die Nutzer verwalten.

Docker stets das neueste Image vom Docker Hub herunter. Das kann dazu führen, dass der Container plötzlich nicht mehr startet, weil die neue Version Änderungen an den Konfigurationsdateien erfordert. Prüfen Sie regelmäßig, ob für Ihre Images (Sicherheits-)Updates vorliegen. Wenn das der Fall ist, dann ersetzen Sie das ausgediente Image durch ein neueres. Die verwendeten Images haben wir unter [ct.de/w7gs](https://github.com/ct-wg/w7gs) verlinkt.

Der Parameter `restart: unless-stopped` sorgt dafür, dass die Container auch nach einem Neustart des Raspi wieder anlaufen, es sei denn, Sie haben zuvor einen Container manuell gestoppt. Der `volumes`-Abschnitt legt fest, wo der Bitwarden-Container persistente Daten speichert:

```
volumes:
  - /var/docker/bitwarden:/data
```

Das Verzeichnis `/data` im Container-Dateisystem entspricht `/var/docker/bitwarden` im Host-Dateisystem. Hier liegt auch die SQLite-Datenbankdatei `db.sqlite3`.

Sie sollten das Verzeichnis `/var/docker/bitwarden` regelmäßig sichern.

Im Compose-File finden Sie die Variablen `BITWARDEN_URL`, `WEBSOCKET_ENABLED` und `ADMIN_TOKEN`. Die Werte holt Docker-Compose sich aus der `.env`-Datei. Öffnen Sie die Datei mit einem Texteditor.

```
BITWARDEN_URL=bitwarden.example.com
WEBSOCKET_ENABLED=true
ADMIN_TOKEN=
```

Tragen Sie hinter `BITWARDEN_URL=` Ihre DynDNS-Adresse ein. Die Adresse wird als Host an Traefik übergeben. Der Eintrag `WEBSOCKET_ENABLED=true` sorgt dafür, dass Passwörter und andere Einträge im Vault bei jeder Änderung synchronisiert werden, sofern der entsprechende Client dies unterstützt („Live-Sync“). Wir empfehlen, Vaultwarden über den Admin-Bereich der Weboberfläche zu konfigurieren. Zum Aktivieren des Admin-Panel setzen Sie hinter `ADMIN_TOKEN=` ein Token ein. Erzeugen Sie dazu auf der Kommandozeile einen langen String mit folgendem Befehl.

```
openssl rand -base64 48 | tr -d /=
```

Mit dem eingetragenen Token können Sie sich später im Admin-Panel der Weboberfläche authentifizieren. Es steht Ihnen frei, die Einstellungen, die das Admin-Panel bietet, manuell vorzunehmen, indem Sie in der `.env`-Datei weitere Umgebungsvariablen setzen und diese in der Datei `docker-compose.yml` referenzieren. Die Entwickler von Vaultwarden stellen im GitHub-Wiki des Projektes eine umfangreiche Beispieldatei zur Verfügung, die Sie über `ct.de/w7gs` finden. Löschen Sie in diesem Fall die Zeile `ADMIN_TOKEN=` in der `.env`-Datei und die zugehörige Referenz aus dem Compose-File.

In unserem Beispiel haben wir das Admin-Panel zusätzlich mit einer Basisauthentifizierung, bestehend aus Username und Passwort, abgesichert. Dazu dient eine Middleware des Reverse-Proxy Traefik, die bereits in der Compose-Datei angelegt ist. Wer im Browser `bitwarden.example.com/admin` ansteuert, muss zunächst die Login-Daten eingeben, um die Seite zu sehen und kann sich erst im zweiten Schritt mit dem Token als Admin authentifizieren. Die nötigen Zugangsdaten erzeugen Sie mit folgendem Befehl:

```
htpasswd -nb user password\  
| sed -e s/\\$/\\$\\$/g
```

Dieser Befehl erzeugt das Zugangsdaten-Paar „user:password“, wobei „password“ als Hash ausgegeben wird. Ersetzen Sie „user“ und „password“ durch sichere Zugangsdaten. Öffnen Sie dann die Datei `docker-compose.yml` mit einem Texteditor und ersetzen Sie in der folgenden Zeile „user:password“ durch die Ausgabe des Befehls.

```
- "traefik.http.middlewares.admin-auth.basicauth.  
users=user:password"
```

Speichern Sie im Anschluss die Datei `docker-compose.yml` und starten Sie die beiden Container:

```
sudo docker-compose up -d
```

Docker lädt nun zunächst die Images aus der Container-Registry. Der Reverse-Proxy kümmert sich um das Zertifikat von Let's Encrypt und liefert Web-Vault und Admin-Panel von Vaultwarden über die konfigurierte DynDNS-Adresse aus. Geben Sie den Containern dabei etwas Zeit zum Starten. Auf einem Raspberry Pi 4 mit 4 GByte Arbeitsspeicher nimmt der erste Startvorgang einige Minuten in Anspruch. Den

Status der Container können Sie mit dem Befehl `docker ps` einsehen. Sobald der Vaultwarden-Container den Status „up x minutes (healthy)“ ausgibt, können Sie das Web-Vault in einem Browser unter der Adresse `bitwarden.example.com` aufrufen. Für das Admin-Panel hängen Sie `/admin` an die URL an. Sollte Ihnen irgendwo ein Fehler unterlaufen sein, dann hilft der Befehl `docker logs` gefolgt vom Namen des Containers bei der Diagnose.

Tresortür abschließen

Registrieren Sie einen neuen Nutzer und vergeben Sie dabei ein sicheres Master-Passwort. Jetzt können Sie sich in das Web-Vault einloggen und Ihr Nutzerkonto einrichten. Wer zuvor einen anderen Passwortmanager genutzt hat, findet sich hier schnell zurecht. Die Kategorie „Mein Tresor“ listet Einträge wie gespeicherte Anmeldedaten und sichere Notizen auf. Diese können Sie als Favoriten markieren oder in Ordner sortieren. In „Werkzeuge“ finden Sie einen Passwortgenerator, Berichte über schwache oder kompromittierte Passwörter und die Funktionen zum Importieren und Exportieren von Passwort-Sammlungen. Bitwarden beherrscht die meisten geläufigen Formate (`.csv`, `.json`, `.xml`, `.txt`). Hier können Sie auch die eigene Passwortdatenbank exportieren. Wählen Sie dabei vorzugsweise das Format „json (encrypted)“, sonst exportiert Bitwarden die Zugangsdaten im Klartext.

Sie sollten für den Login unbedingt einen zweiten Faktor verlangen. Andernfalls könnte es Angreifern potenziell gelingen, sich mit Brute-Force-Angriffen Zugriff auf Ihre Passwortdatenbank zu verschaffen. Die nötigen Handgriffe nehmen Sie in den Einstellungen im Untermenü „Zwei-Faktor-Authentifizierung“ vor. Der Passwortmanager bietet Zwei-Faktor-Authentifizierung mittels 2FA-Apps wie Authy, Duo oder Microsoft Authenticator, via E-Mail und mit FIDO-kompatiblen Hardware-Sicherheitsschlüsseln an.

Sobald ein zweiter Faktor eingerichtet ist, können Sie im Menü „Zwei-Faktor-Authentifizierung“ einen Wiederherstellungsschlüssel anzeigen lassen. Falls Sie beispielsweise Ihr Mobiltelefon und damit Zugriff auf den zweiten Faktor verlieren, ist der Wiederherstellungsschlüssel die letzte Möglichkeit, erneut Zugang zum Vault zu bekommen. Notieren Sie den Schlüssel und verwahren Sie ihn an einem sicheren Ort. Tritt das Szenario ein, dann rufen Sie im Browser die Adresse `bitwarden.example.com/#/recover-2fa/` auf und folgen Sie den Anweisungen. Wenn Sie direkt mehrere Methoden zur Zwei-Faktor-Authentifi-

zierung einrichten, dann kommen Sie erst gar nicht in diese Situation.

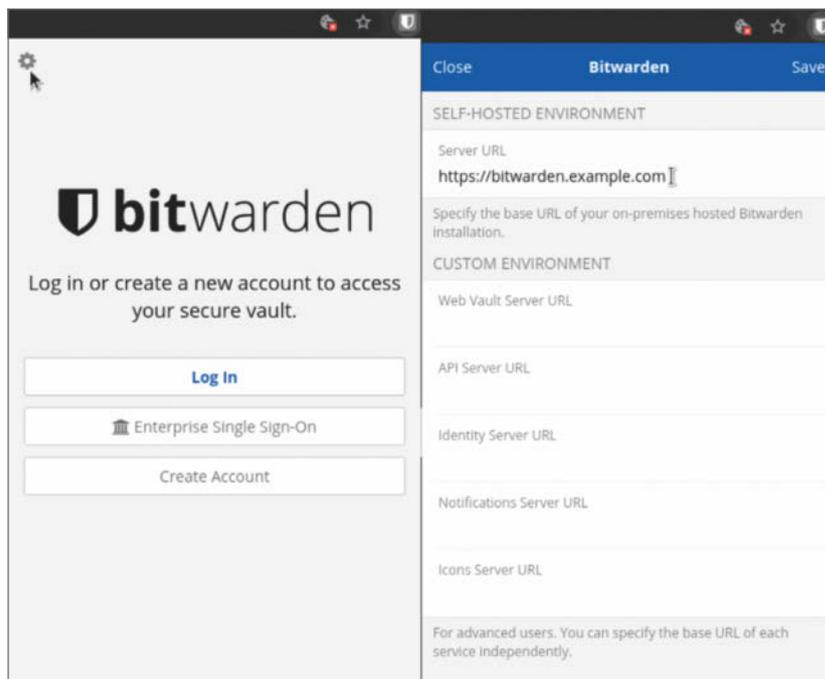
Aktuell kann noch jede Person, die die URL Ihrer Web-Vault kennt, ein Konto anlegen. Wer also keinen Gratis-Passwortmanager ins Internet hängen möchte, tut gut daran, neue Registrierungen zu deaktivieren. Rufen Sie das Admin-Panel im Browser unter der Adresse `bitwarden.example.com/admin` auf. Wir empfehlen, im Admin-Panel einige Einstellungen vorzunehmen, um die Angriffsfläche Ihres Passwort-Servers zu verringern.

Im Bereich „General“ müssen Sie den korrekten Wert im Feld „Domain URL“ eintragen, damit automatisch versandte Mails die richtigen Links enthalten. Verboten Sie Registrierungen auf der Hauptseite, indem Sie den Haken beim Eintrag „Allow new signups“ entfernen. Die Schaltfläche und das Eingabefeld zum Registrieren bleiben zwar weiterhin sichtbar, Registrierungen sind aber nicht mehr möglich. Im Bereich „SMTP Email Settings“ sollten Sie einen SMTP-Postausgangsserver konfigurieren, damit der Passwort-Server Mails verschicken kann, etwa, um Accounts zu verifizieren oder zu informieren, falls eine Anmeldung in Ihrem Account von einem zuvor unbekanntem Gerät erfolgt ist. Im Admin-Panel können Sie auch dann weitere Nutzer via Mail einladen, wenn die Registrierung auf der Hauptseite deaktiviert ist. Teilen Sie den Passwort-Server beispielsweise mit Freunden, sollten Sie darüber nachdenken, ihnen zu verbieten, weitere Nutzer einzuladen. Dazu entfernen Sie den Haken bei „Allow Invitations“ im Bereich „General“.

Wem das nicht genügt, der deaktiviert das Web-Vault mit der Umgebungsvariable `WEB_VAULT_ENABLED=false` oder betreibt Vaultwarden ausschließlich im Heimnetz mit selbsterzeugten Zertifikaten. Um dann trotzdem von unterwegs auf den Passwort-Server zuzugreifen, müssten Sie sich mit einer VPN-Lösung behelfen. Die Entwickler raten davon ab, die integrierte Methode `ROCKET_TLS` für HTTPS zu nutzen. Stattdessen empfehlen sie, private Instanzen mit dem Reverse-Proxy Caddy und Let's Encrypt-Zertifikaten abzusichern. Eine Anleitung für das nicht ganz triviale Verfahren steht auf GitHub im Wiki des Projekts zur Verfügung, das wir unter `ct.de/w7gs` verlinkt haben.

Den Passwortmanager nutzen

Der Passwort-Server ist jetzt einsatzbereit. Vaultwarden bietet viele Features, die für gewöhnlich einen Bitwarden-Premium-Account voraussetzen. Einige



Bitwarden-Clients existieren für diverse Plattformen, hier sehen Sie das Browser-Add-On in Chromium. Damit der Client den Passwort-Server ansteuern kann, braucht er die Server-URL.

Funktionen fehlen jedoch, etwa Push-Benachrichtigungen in iOS und Android, Single-Sign-On oder Notfall-Vollmachten (Emergency Access). Welche Unterschiede zum offiziellen Bitwarden-Server bestehen, haben die Entwickler im Wiki des Projekts in einer Liste festgehalten, die Sie in der Linksammlung unter `ct.de/w7gs` abrufen können.

Sie können zum Speichern, Verwalten und Synchronisieren von Passwörtern neben dem Web-Vault auch die Bitwarden-Browser-Erweiterungen sowie die Apps für Android und iOS nutzen. Auch Desktop-Clients für Windows, Linux, macOS und sogar Tools für die Kommandozeile sind verfügbar. Eine vollständige Liste aller Clients finden Sie auf der offiziellen Bitwarden-Website, die wir unter `ct.de/w7gs` verlinkt haben. Damit die Anwendungen mit der Server-API kommunizieren können, müssen Sie die URL Ihrer Vaultwarden-Instanz eintragen. Das vollständige Menü verbirgt sich hinter dem Zahnrad in der linken oberen Ecke des Anmeldefensters.

Literatur

[1] Jan Mahn,
Container-Komponist,
Docker-Container mit
Docker-Compose
einrichten,
c't 26/2018 S. 148

Linksammlung zum Projekt

ct.de/w7gs

Doch Vorsicht: Die Schaltfläche „abmelden“ in den Clients ist irreführend und bedeutet „vom Server abmelden“. Wenn Sie sich vollständig abmelden und die Server-API nicht erreichbar ist, dann verweigert der Bitwarden-Client einen neuen Login. Um sich dann erneut anzumelden, muss eine Verbindung zum Server bestehen. Wählen Sie darum stets die Option „Jetzt sperren“, um Ihr Nutzerkonto zu sperren, es sei denn, Sie beabsichtigen, das Nutzerkonto zu wechseln oder den Client an einem anderen Server anzumelden.

Sie können jetzt loslegen und den Passwort-Treasure befüllen. Wenn Sie zukünftig im Web eine Login-Seite ansteuern und Anmeldedaten eingeben, bietet beispielsweise die Bitwarden-Browsererweiterung an, die Login-Daten im Vault zu speichern. Für neue Logins kann der integrierte Passwortgenerator zufällige, sichere Passwörter erzeugen. Falls bereits Anmeldedaten im Vault gespeichert sind, kann Bit-

warden Login-Formulare automatisch ausfüllen. Das Auto-Fill-Verhalten können Sie für einzelne Seiten an ihre Vorstellungen anpassen. Gespeicherte Passwörter werden von nun an automatisch synchronisiert.

Fazit

Mit Vaultwarden verwandeln Sie ihr System im Handumdrehen in einen Passwortspeicher mit Cloud-Funktionalität, müssen dafür jedoch Ihre Passwortsammlung nicht aus der Hand geben. Mit dem Zugewinn an Freiheit kommt allerdings auch Verantwortung: Nehmen Sie regelmäßig Backups vor und prüfen Sie die Container-Images auf Sicherheitsupdates. Richtig angewendet ersetzt Vaultwarden einen kostenpflichtigen Passwortmanager und eignet sich optimal für die Nutzung in kleinen Teams oder gemeinsam mit Freunden oder der Familie. (ndi) **ct**

Hilfreiche Tool-Tipps für Ihren Job!

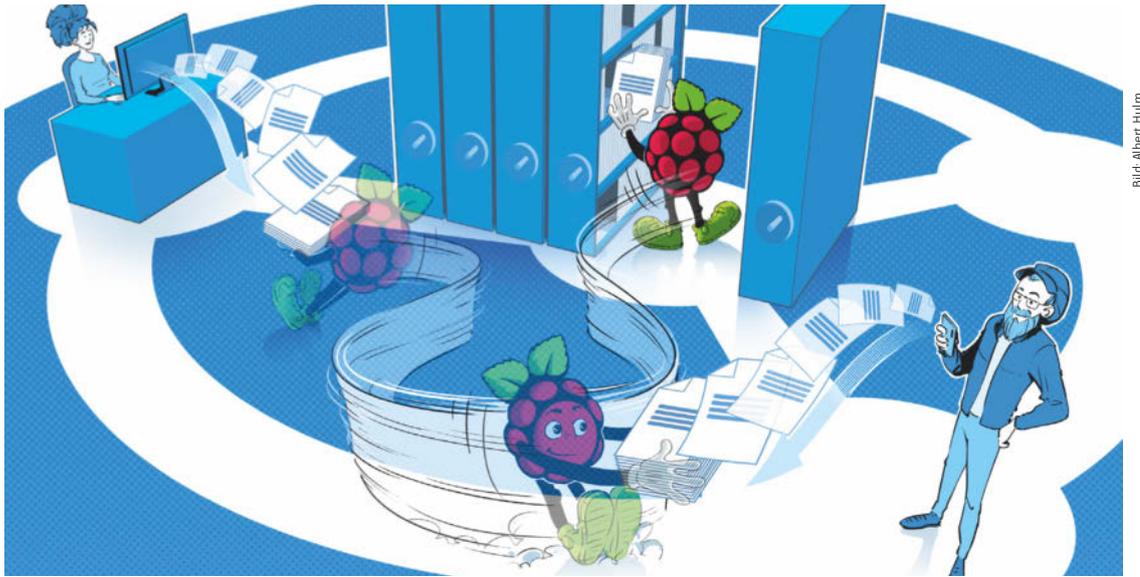


Steigern Sie Ihre Effizienz im Homeoffice, unterwegs oder im Büro. Das **Sonderheft c't @work** zeigt Ihnen, welche nützlichen Tools Ihnen einen echten Mehrwert im Job bringen:

- ▶ Abhilfe für zähe **Videokonferenzen** schaffen
- ▶ Mit **Threadit** kurze Video-Tutorials erstellen
- ▶ **Online-Interaktionsplattformen** beruflich nutzen
- ▶ Rechtssicher **digital unterzeichnen**

Heft für 14,90 € • PDF für 12,99 €
Bundle Heft + PDF 19,90 €

 shop.heise.de/ct-work22



Backups mit Syncthing

Die quelloffene Software Syncthing gleicht Ordnerinhalte zwischen Geräten ab und lässt sich auch hervorragend zur Datensicherung einsetzen – zum Beispiel auf einem Raspberry Pi mit großer Festplatte als zentraler Backup-Speicher für all Ihre Rechner, Smartphones & Co.

Von **Jan Schübler**

Syncthing eignet sich, um etwa die mit dem Smartphone geschossenen Fotos automatisch und ganz ohne Cloud auf den Rechner zu befördern, auf einem Raspi taugt es aber auch als zentraler Backup-Speicher für das digitale Hab und Gut der ganzen Familie. Nach der Einrichtung muss man sich um Syncthing nicht mehr kümmern, es erledigt still und zuverlässig seinen Job. Die Datenübertragung erfolgt stets Ende-zu-Ende-verschlüsselt.

Im Gegensatz zum funktional vergleichbaren Resilio Sync handelt es sich bei Syncthing um quelloffene Software, und anders als bei Tools wie Robocopy oder Synchredible geschieht die Synchronisierung nicht durch direktes Kopieren zwischen Dateisystemen, sondern via Peer-to-Peer-Übertragung (P2P). Es muss auf jedem beteiligten Gerät laufen.

Um was für welche es sich dabei handelt, ist allerdings beinahe egal: Weil Syncthing quelloffen ist, gibt es die Software für die meisten gängigen Plattformen wie Windows, Linux und macOS und für ein paar weniger gängige Systeme. Auch für Android und iOS gibts passende Apps, und sogar für die NAS-Betriebssysteme von QNAP und Synology.

Das Charmante an Syncthing ist außerdem, dass man einen synchronisierten Speicherplatz auf einem Netzwerkgerät bekommt, ohne dafür groß mit NAS-Diensten, Netzwerkkonfigurationen und Benutzerrechten hantieren zu müssen – dank grafischer Oberfläche ist die Einrichtung sehr einfach und schnell erledigt. Wer will, kann Syncthing nicht nur im lokalen Netz nutzen, sondern auch übers Internet: Praktischerweise existiert ein Community-

betriebenes Netz von Discovery- und Relay-Servern, damit Daten auf Wunsch auch über NAT-Grenzen hinweg synchronisiert werden können. Relays speichern Inhalte bei Bedarf zwischen, sodass die Synchronisierung zwar verzögert wird, nicht aber komplett abbricht, sollte ein Gerät zeitweilig die Internetverbindung verlieren. Die Verschlüsselung bleibt dabei bestehen.

Ideal ist ein Syncthing-gestütztes Backup für Nutzdaten wie Dokumente, Musik und Fotos. Das gilt vor allem, wenn man die integrierte Versionierung einschaltet: Dann speichert Syncthing Dateien beim Ändern oder Löschen in einem Versionsarchiv, aus dem sich ältere Versionen bei Bedarf wiederherstellen lassen. Das ist nicht nur ein Sicherheitsnetz gegen versehentliche Änderungen, sondern auch gegen Verschlüsselungstrojaner.

Bedenken Sie aber, dass zu einer umfassenden Backup-Strategie mitunter weitere Aspekte gehören können. Bei Windows-PCs mit lange gewachsenen, individuellen Installationen etwa empfehlen sich zusätzlich komplette System-Backups, mit denen man den kompletten Installations- und Konfigurationszustand eines Rechners im Handumdrehen wiederherstellen kann. Dafür eignet sich zum Beispiel unser Tool c't-WIMage [1], aber auch andere

Software wie Drive Snapshot, Acronis True Image und Ähnliche [2]. Ein gutes Backup folgt zudem der 3-2-1-Regel: Drei Kopien (das Original zählt mit), davon zwei auf verschiedenen Medientypen, und eines davon außer Haus – mehr dazu lesen Sie in [3].

In diesem Artikel erklären wir die Installation von Syncthing auf Windows 10, Ubuntu Linux, Android und iOS sowie auf einem Raspi 4 mit dem regulären Raspberry Pi OS mit Desktopumgebung – macOS haben wir hier ausgeklammert, weil es mit Time Machine bereits eine eigene, ähnlich funktionierende Lösung mitbringt. Der Raspi 4 eignet sich dank USB 3.0 zusammen mit einer externen Festplatte hervorragend als dauer verfügbarer Backup-Speicher. Im nachfolgenden Artikel lesen Sie, wie Sie Syncthing auf den Clients für die Datensicherung auf einem zentralen Backup-Raspi konfigurieren und wie Sie Daten daraus im Falle eines Falles wiederherstellen.

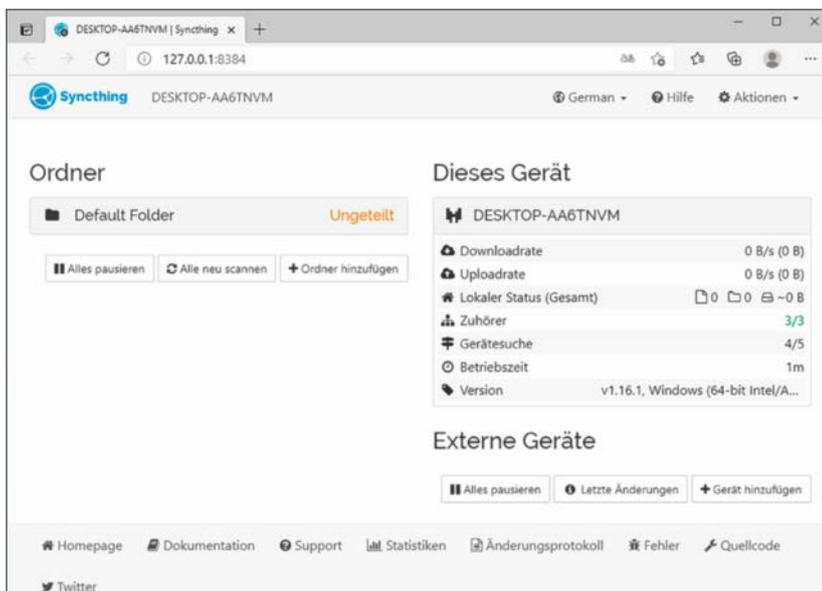
Allgemeines

Syncthing arbeitet im Hintergrund und stellt zur Konfiguration ein Web-Frontend auf HTTP-Port 8384 bereit. Das heißt: Läuft Syncthing, lässt sich die Oberfläche im lokalen Webbrowser durch Eingabe der Adresse 127.0.0.1:8384 aufrufen. Ein separates Konfigurationsprogramm wird nicht mitgeliefert. Es ist nicht nötig und auf dem Desktop zunächst auch nicht enthalten. Wer trotzdem eins haben will: Für Linux gibt es Tools wie Syncthing-GTK, das auch in den Repositories einiger Distributionen enthalten ist. Auf Windows-Systemen empfiehlt sich SyncTrayzor (siehe ct.de/wm7d). Es bringt die Syncthing-Engine gleich mit und zeigt die bekannte Oberfläche in einem eigenen Programmfenster. Außerdem legt es ein Status-Icon für den Infobereich in der Taskleiste ab.

Auf Windows und Linux hat Syncthing die Eigenart, nicht automatisch zu starten, weshalb dafür ein paar zusätzliche Handgriffe nötig sind.

Installation auf Windows

Für Windows gibt es keinen Installer, sodass Sie Syncthing händisch in einen Programmordner kopieren müssen, den Sie zunächst erstellen. Öffnen Sie den Datei-Explorer und dann Laufwerk C: per Doppelklick. Mit einem Rechtsklick in den leeren Bereich des Laufwerks und Auswahl von „Neu/Ordner“ erstellen Sie ein Verzeichnis für die Programmdateien. Nennen Sie ihn zum Beispiel „Syncthing“. Darin legen Sie nun die zuvor heruntergeladenen, entpackten Programmdateien ab.



Läuft Syncthing im Hintergrund? Das können Sie einfach durch Eingabe der Adresse 127.0.0.1:8384 in einem beliebigen Browser prüfen. Es öffnet sich das Webinterface, über das Syncthing auch konfiguriert wird.

Um Syncthing nach einem Neustart des Rechners automatisch starten zu lassen, empfiehlt sich die Windows-Aufgabenplanung: Drücken Sie die Windows-Taste, tippen Sie `taskschd.msc` ein und bestätigen Sie mit der Eingabetaste. Klicken Sie rechts auf „Aufgabe erstellen“. Auf der Registerkarte „Allgemein“ vergeben Sie einen Namen wie „Syncthing automatisch starten“ und markieren die Sicherheitsoption „Unabhängig von der Benutzeranmeldung ausführen“.

Auf der Registerkarte „Trigger“ klicken Sie auf „Neu“. Für „Aufgabe starten“ wählen Sie „Beim Start“ und bestätigen mit OK. Wechseln Sie nun zur Registerkarte „Aktionen“ und klicken Sie auch hier auf „Neu“ und dann bei „Programm/Skript“ auf „Durchsuchen ...“. Wählen Sie die zuvor nach `C:\Syncthing` kopierte Programmdatei namens „syncthing.exe“ per Doppelklick aus. Ins Feld „Argumente hinzufügen (optional)“ tragen Sie

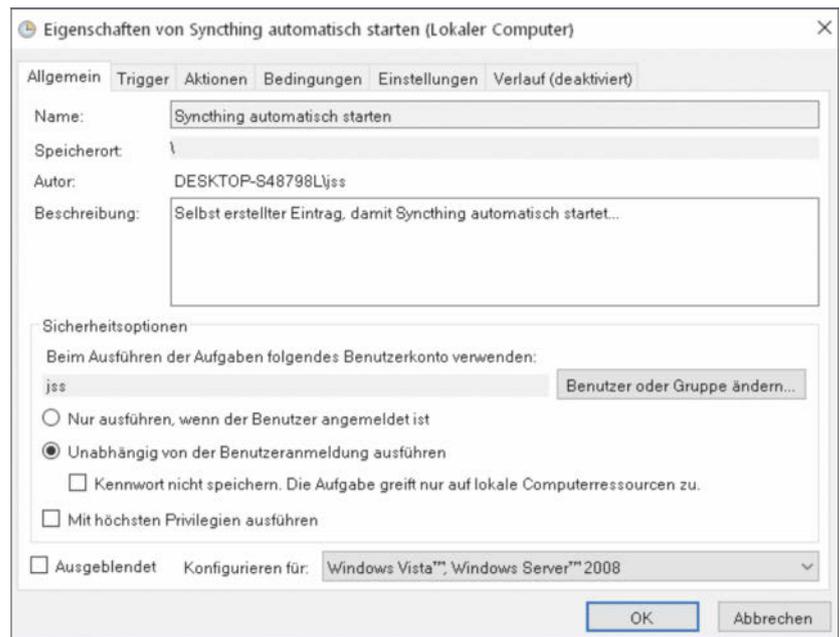
```
--no-console --no-browser
```

ein. Entfernen Sie dann noch auf der Registerkarte „Einstellungen“ das Häkchen vor „Aufgabe beenden, falls Ausführung länger als ...“. Wenn Syncthing auf Notebooks auch im Akkubetrieb funktionieren soll, entfernen Sie außerdem auf der Registerkarte „Bedingungen“ noch die Häkchen vor „Beenden, wenn Computer in den Akkubetrieb wechselt“ und vor „Aufgabe nur starten, falls Computer im Netzbetrieb ausgeführt wird“. Prinzipiell ist das sinnvoll, es kann beim Sync größerer Datenmengen allerdings zu einem höheren Stromverbrauch führen - und auch das Datenvolumen einer eventuellen Mobilfunkverbindung stark belasten (Tipps zur Schonung des Datenvolumens finden Sie in [4]).

Bestätigen Sie nun den Aufgabendialog mit OK - nach Bestätigung durch Eingabe Ihres Windows-Kennworts ist der Syncthing-Autostart eingerichtet. Testen können Sie das ganz einfach, indem Sie nach einem Windows-Neustart die Syncthing-Oberfläche unter `127.0.0.1:8384` aufrufen. Beim ersten Start von Syncthing bekommen Sie im Regelfall auch eine Firewall-Abfrage - erlauben Sie dem Programm die Kommunikation.

Linux

Syncthing ist in den Repositories vieler Desktop-Linux-Distributionen enthalten, etwa bei Ubuntu und seinen Derivaten, bei OpenSUSE Leap und bei Debian - es lässt sich einfach über die Software-



Damit Syncthing mit Windows startet, müssen Sie eine passende Aufgabe in die Aufgabenplanung eintragen.

verwaltung finden und installieren. Alternativ installieren Sie es unter Ubuntu & Co. auch per Konsole mit dem Bash-Kommando

```
sudo apt install syncthing
```

Leider startet Syncthing damit noch nicht automatisch. Das ändern Sie per

```
sudo systemctl enable syncthing@benutzername
```

gefolgt von

```
sudo systemctl start syncthing@benutzername
```

Ersetzen Sie „benutzername“ dabei durch den Namen Ihres Benutzerkontos. Nach einem Neustart sollte Syncthing automatisch laufen und im Browser unter `127.0.0.1:8384` aufrufbar sein.

Raspberry Pi OS

Im regulären Software-Repository der Linux-Variante Raspberry Pi OS ist Syncthing zwar enthalten. Wir empfehlen allerdings die Installation aus dem APT-

Repository von Syncthing selbst – die Einrichtung des Autostarts lief damit glatter ab. Zunächst importiert der Konsolenbefehl

```
sudo curl -s -o /usr/share/keyrings/
syncthing-archive-keyring.gpg
https://syncthing.net/release-key.gpg
```

den GPG-Schlüssel des Repositorys. Fügen Sie dann die Paketquelle hinzu:

```
echo "deb [signed-by=/usr/share/keyrings/
syncthing-archive-keyring.gpg]
https://apt.syncthing.net/ syncthing stable" |
sudo tee /etc/apt/sources.list.d/syncthing.list
```

Nun sorgen Sie noch dafür, dass Syncthings eigenes Repository bei der Installation bevorzugt wird:

```
printf "Package: *\nPin: origin
apt.syncthing.net\nPin-Priority: 990\n" |
sudo tee /etc/apt/preferences.d/syncthing
```

Nun aktualisieren Sie die Paketquellen per

```
sudo apt-get update
```

und installieren Syncthing mit

```
sudo apt-get install syncthing
```

Jetzt müssen Sie noch dafür sorgen, dass Syncthing beim Systemstart automatisch geladen wird. Das lässt sich einfach mit `systemd` erledigen, und zwar mit

```
sudo systemctl enable syncthing@pi
```

sowie

```
sudo systemctl start syncthing@pi
```

Nach einem Neustart sollte Syncthing automatisch starten, was Sie einfach prüfen können, indem Sie versuchen, die Weboberfläche von Syncthing im Browser via `127.0.0.1:8384` aufzurufen.

Android

Für Android steht Syncthing sowohl im Play Store als auch in F-Droid zur Verfügung. Beim ersten Start verlangt es nach der Berechtigung, auf den Speicher

und optional auf den Standort zuzugreifen – letzteres, damit es auch im Hintergrund erkennen kann, ob und mit welchem WLAN das Smartphone verbunden ist, um nach Wunsch nur in bestimmten WLANs zu synchronisieren. Je nach Android-Version und -Anbieter zeigt die Erlaubnis-Abfrage für den Standortzugriff allerdings nicht die erforderliche Option „Immer zulassen“ an. Setzen Sie die Erlaubnis stattdessen von Hand – bei einem Google Pixel 6 mit Android 12 in den Einstellungen unter „Apps/Alle Apps ansehen/Syncthing/Berechtigungen/Standort“.

Außerdem empfiehlt die App, auch die Ausführung im Hintergrund zu erlauben. Sinnvoll ist diese Einstellung, um Syncthing nicht jedes Mal extra öffnen zu müssen, um die Synchronisierung zum Beispiel neuer Fotos anzustoßen; sie kann sich allerdings negativ auf die Akkulaufzeit auswirken, wenn Syncthing viel und oft synchronisieren muss. Je nach Hersteller und Android-Version finden Sie diese Einstellung an einer anderen Stelle, bei unserem Pixel 6 zum Beispiel in den Einstellungen unter „Apps/Alle Apps ansehen/Syncthing/Akku“.

iOS

Eine offizielle Syncthing-App existiert für iOS und iPadOS nicht, stattdessen gibt es Möbius Sync. Die App fußt auf der quelloffenen Syncthing-Engine, bringt aber eine kleine Modifikation mit: Übersteigt der synchron zu haltende Datenbestand eine Gesamtgröße von 20 MByte, wird eine Freisaltung des „Unlimited File Sync“ per In-App-Kauf nötig. Derzeit kostet das einmalig 20 Euro. In Sachen Bedienung gleichen sich Syncthing und Möbius Sync ansonsten weitgehend.

Unter iOS beziehungsweise iPadOS muss man allerdings damit leben, dass die Synchronisierung im Hintergrund stark eingeschränkt ist, weil das Betriebssystem Hintergrundaufgaben restriktiv und in nicht genau vorhersehbarer Weise einschränkt. Die FAQ zur App (siehe ct.de/wm7d) nennen rund ein bis zwei Stunden Sync-Aktivität pro Tag als realistischen Wert.

Neben den genannten Betriebssystemen ist Syncthing auch für weitere Systeme wie macOS, FreeBSD und OpenBSD zu haben – und natürlich als Quellcode. Auch für NAS-Betriebssysteme von QNAP und Synology stehen Pakete bereit – die Downloads haben wir unter ct.de/wm7d verlinkt.

Nachdem Sie Syncthing auf allen gewünschten Geräten installiert haben, ist es bereit zur Einrichtung als Backup-Lösung. Wie das geht, steht im nachfolgenden Artikel. (jss) **ct**

Literatur

[1] Axel Vahldiek, **Ersatzrad**, c't-WIMAge erstellt Windows-Backups, c't 10/2021, S. 18

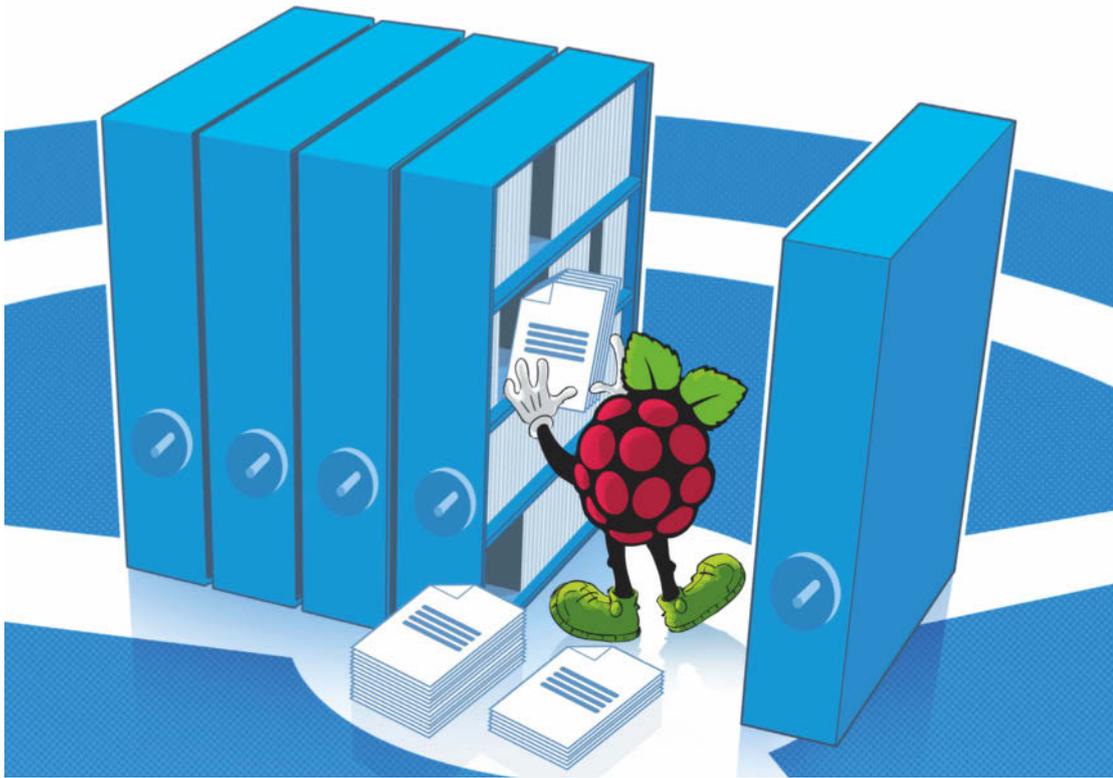
[2] Jan Schüßler, **Sicherheitsnetze**, Backup-Programme für Windows, c't 8/2018, S. 122

[3] Axel Vahldiek, **Emotet: Hält Ihr Backup?**, So sichern Sie Ihre Daten wirklich zuverlässig, c't 10/2020, S. 16

[4] Jan Schüßler, **Sparsam im Grünen**, Wie Sie unter Windows das Inklusivvolumen Ihres Mobilfunkvertrags schonen, c't 11/2021, S. 150

Syncthing-Downloads

ct.de/wm7d



Synching-Backup einrichten

Einmal eingerichtet, läuft Synching unauffällig im Hintergrund. Doch das allein ergibt noch keine Datensicherung – in diesem Artikel lesen Sie, wie Sie Synching unter Verwendung eines Raspberry Pi als zentralen Speicher dafür einrichten. So sind Ihre Daten sogar vor Erpressungs-Trojanern geschützt.

Von **Jan Schübler**

Wir gehen in diesem Artikel davon aus, dass Sie wie im vorigen Artikel beschrieben Synching auf den zu sichernden Geräten sowie auf einem Raspi mit einem externen Laufwerk als zentralen Backuprechner installiert haben. Eine

schöne Idee ist auch das Gehäuse mit SSD-Steckplatz, das wir im Kasten auf Seite 61 vorstellen.

Hier geht es nun um die Einrichtung des Backups am Beispiel eines Datenordners, der von einem Windows-Rechner auf den Backupraspi synchronisiert

werden soll. Das kann zum Beispiel der Ordner „Dokumente“ sein, in dem sich wichtige Dateien befinden, die sich häufig ändern. Genauso gut können Sie Ihre Fotos, Projekte und vieles mehr sichern lassen. Auch wenn Sie Daten von Linux-Rechnern, Macs oder Smartphones/Tablets sichern wollen, hilft Ihnen diese Anleitung weiter. Die zur Konfiguration genutzte Weboberfläche von Syncthing ist immer dieselbe. Ebenso können Sie auch eine andere Hardware als einen Raspi als Backupziel verwenden, zum Beispiel einen alten Linux- oder Windows-PC. Bedenken Sie aber den Stromverbrauch – 3 Watt im Leerlauf schafft kein normaler PC.

Die Bedienoberfläche eines installierten und laufenden Syncthing erreichen Sie auf Windows- und Linux-Systemen im Browser durch Aufruf von 127.0.0.1:8384. Bevor es an die Konfiguration der Datensicherung geht, sollten Sie aber ein paar grundlegende Einstellungen vornehmen. So ist die Syncthing-Oberfläche serienmäßig nicht kennwortschutzgeschützt. Das ändern Sie in den Einstellungen, die Sie über das Aktionen-Menü oben rechts öffnen, und dort auf die Registerkarte „GUI“ klicken.

Als Gerätenamen übernimmt Syncthing automatisch den Netzwerknamen des Rechners. Je nach PC ist das mitunter kryptisch anmutender Zeichensalat – um die Einrichtung zu erleichtern, sollten Sie all Ihren Syncthing-Instanzen sprechende Namen geben, wie zum Beispiel „Desktop-PC“, „Arbeitsnotebook“ oder „Backup-Raspi“. Das erledigen Sie ebenfalls in den Einstellungen: Tragen Sie auf der Registerkarte „Allgemein“ bei „Gerätename“ den Namen ein.

Außerdem kann es sinnvoll sein, die Syncthing-Oberfläche des Raspi im internen Netzwerk verfügbar zu machen, sodass Sie die Konfiguration auch von einem anderen Rechner aus vornehmen können. Dafür tragen Sie auf der Registerkarte „GUI“ als „Adresse der Benutzeroberfläche“ 0.0.0.0:8384 ein. Nun können Sie die Oberfläche auch von anderen Geräten aus über den Hostnamen oder die IP-Adresse des Raspi aufrufen – im einfachsten Fall also über `raspberrypi:8384`, sofern Sie den Hostnamen des Raspi nicht geändert haben. Klappt das nicht, verwenden Sie stattdessen die IP-Adresse Ihres Raspi, die Sie mit dem Konsolenbefehl `ifconfig` ermitteln können.

Datenträger vorbereiten

Kommt ein Raspi als Backupziel zum Einsatz, gehören die Daten auf eine externe USB-Festplatte



Hängt das zu sichernde Gerät nicht im gleichen Netz wie der Raspi, müssen Sie einmalig die Geräte-ID eintippen – oder auf einem Smartphone einfach den QR-Code scannen.

oder -SSD – die SD-Karten, auf denen bei Raspis das Betriebssystem läuft, sind bei Dauerbelastung weniger zuverlässig und oft auch langsamer. Im besten Fall kommt ein Raspi 4 zum Einsatz, dessen USB-3.0-Anschluss deutlich schneller überträgt als die USB-2.0-Schnittstelle älterer Raspis.

Das Formatieren der Platte ist schnell mit dem Standardwerkzeug GParted erledigt. Falls es nicht installiert ist, ändern Sie das mit dem Konsolenbefehl

```
sudo apt install gparted
```

Starten Sie das Programm dann aus dem Startmenü unter „Systemwerkzeuge“ oder per `sudo gparted`. In GParted wählen Sie oben rechts die externe Platte aus (meist `/dev/sda`) und verpassen ihr unter „Laufwerk/Partitionstabelle erstellen“ eine neue Partitionstabelle vom Typ `gpt`. Achtung: Alle auf der Platte befindlichen Daten werden dabei gelöscht.

Nun legen Sie über „Partition/Neu“ eine Partition für die Backupdaten an. Im Prinzip genügt es, im Partitionsdialog einfach nur im Feld „Bezeichnung“ einen Namen wie „Backup-Platte“ zu vergeben, der Rest passt so. Geschmackssache ist, ob Sie das Dateisystem von `Ext4` (dem Debian-Linux-Standard) auf Microsofts `NTFS` umstellen. Das hat den Vorteil, dass Sie die externe Festplatte bei einem Ausfall des Raspi problemlos an einen Windows-PC anklammern und ohne weitere Tools auslesen können. Die Performance ist mit `Ext4` allerdings besser – deshalb haben wir uns dafür entschieden.

Nach einem Klick auf „Hinzufügen“ und auf das grüne Häkchen wird die neue Partition auf der Festplatte erstellt. Nun sollte das Laufwerk „Backup-Platte“ im Datei-Explorer des Raspi links oben auftauchen; ein Klick darauf mountet die Partition nach `/media/pi/Backup-Platte`.

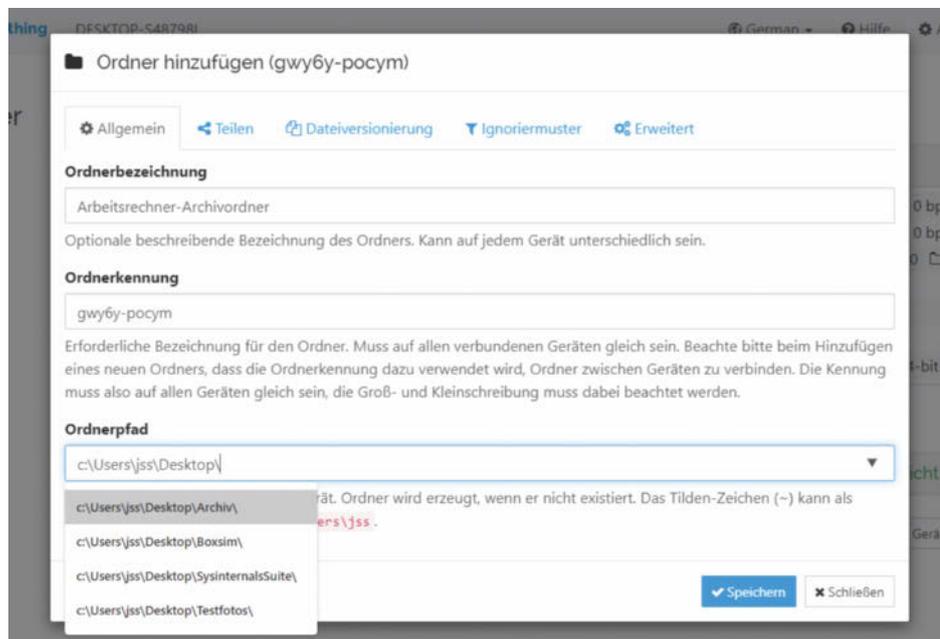
Die frisch formatierte und eingebundene `Ext4`-Partition lässt sich nun zwar lesen, doch zum Schrei-

ben, Ändern oder Löschen von Dateien bräuchten Sie Root-Rechte. Das ändern Sie mit dem Konsolenbefehl

```
sudo chmod 777 /media/pi/Backup-Platte
```

In unseren Versuchen erwies sich manche Hardware als bockig. Die Festplatten-Adapterbox Quickport XT von Sharkoon zum Beispiel wurde nach einem Neustart des Betriebssystems oftmals nicht mehr erkannt, und wenn doch, dann nicht mehr automatisch gemountet. Offenbar gelangt der Controller mitunter in einen Zustand, in dem er nicht mehr mit dem Raspi reden kann. Es hilft dann, die Stromzufuhr des Quickport für ein paar Sekunden zu unterbrechen.

Es ist sinnvoll, auf dem Backuplaufwerk eine Struktur mit Ordnern für die einzelnen Sync-Aufträge Ihrer Geräte anzulegen. Erstellen Sie also Ordner mit sprechenden Namen wie „Arbeitsnotebook“, „Desktop-PC“ oder „Smartphone“. Das geht recht simpel über den Datei-Explorer von Raspberry Pi OS per Rechtsklick in den leeren Bereich des Laufwerks und „New Folder“. Falls Sie mehrere Ordner eines Gerätes sichern möchten, bei einem Desktop-PC also zum Beispiel den Dokumenten- und den Bilderordner, erstellen Sie dafür wiederum Unterordner.



Um den Überblick zu behalten, geben Sie den synchronisierten Ordnern auf dem Raspi sprechende Bezeichnungen und eigene Unterverzeichnisse.

Einander vorstellen

Bevor Sie Synchronisierungsaufträge einrichten können, müssen Sie die Geräte, zwischen denen Daten synchronisiert werden sollen, miteinander bekannt machen. In diesem Fall bedeutet das: Der Raspi muss alle zu sichernden Geräte kennen, die aber nur jeweils den Raspi.

Um den Raspi dem Quellgerät vorzustellen, klicken Sie in der Syncthing-Oberfläche auf letzterem im Abschnitt „Externe Geräte“ auf „Gerät hinzufügen“. Auf der Registerkarte „Allgemein“ sollte Syncthing nun bereits Geräte-IDs vorschlagen, die es im lokalen Netzwerk findet und auf denen ebenfalls Syncthing läuft. Im Regelfall sind das alle, die per WLAN oder LAN am selben Router hängen. Wählen Sie den Raspi per Klick auf seine ID aus. Vergeben Sie unter „Gerätename“ noch einen eindeutigen Namen wie „mein Backup-Raspi“ und bestätigen Sie mit „Speichern“. In Syncthing auf dem Raspi erscheint nach ein paar Sekunden bis Minuten eine Benachrichtigung, dass er hinzugefügt wurde. Klicken Sie in dieser Meldung auf „Gerät hinzufügen“ und verfahren Sie wie eben beschrieben, um auch das Quellgerät dem Raspi bekannt zu machen.

Falls Sie mehr als nur einen PC und den Raspi mit Syncthing verwenden, erkennen Sie nicht auf An-

hieb, welches Gerät welches ist. In diesem Fall öffnen Sie Syncthing auf dem Raspi, klicken im Aktionen-Menü auf „Eigene Kennung“ und merken sich die ersten drei, vier Ziffern der ID – dann gelingt die Identifizierung.

Ab und an erleben wir es beim Hinzufügen einer neuen Geräteverbindung, dass Syncthing auf dem einen oder anderen Rechner die IDs anderer lokaler Syncthing-Geräte partout nicht anbietet. Das ist nicht tragisch: Findet etwa Ihr Laptop den Raspi nicht, schauen Sie einfach auf dem Raspi nach, ob Syncthing dort den Laptop findet – die Reihenfolge, in der Sie die Geräte hinzufügen, ist egal. Einzig dann, wenn sich die Syncthing-Installationen auf beiden Geräten dauerhaft nicht finden, müssen Sie die Geräteerkennung des einen auf dem anderen Rechner eintippen. Auf Geräten, die eine Kamera haben (etwa Smartphones), können Sie dafür auch den QR-Code benutzen, den Sie im Aktionen-Menü unter „Eigene Kennung“ angezeigt bekommen.

Syncs einrichten

Klicken Sie in Syncthing auf dem zu sichernden Gerät nun auf „Ordner hinzufügen“. Auf der Registerkarte „Allgemein“ geben Sie im Feld „Bezeichnung“ eine Klartext-Bezeichnung für das ein, was gesichert

Sobald eine neue Synchronisierung eingerichtet ist, legt Syncthing mit der Arbeit los – der Raspi bekommt die freigegebenen Dateien vom PC.

The screenshot displays the Syncthing web interface. On the left, the 'Ordner' (Folders) section shows a folder named 'Arbeitsrechner-Archivordner' with a synchronization status of 53% and 408 MiB. Below this, various settings like 'Ordnererkennung', 'Globaler Status', and 'Lokaler Status' are visible. On the right, the 'Dieses Gerät' (This Device) section shows system metrics such as 'Downloadrate', 'Uploadrate', 'Lokaler Status (Gesamt)', 'RAM-Auslastung', and 'Prozessorauslastung'. At the bottom, the 'Externe Geräte' (External Devices) section lists the 'Arbeitsrechner' device, which is currently 'Aktuell' (Current), with its own set of system metrics and folder information.

werden soll, zum Beispiel „Arbeitsrechner-Archivordner“. Die zufällig generierte „Ordnerkennung“ können Sie übernehmen. Als „Ordnerpfad“ geben Sie das Verzeichnis mit den zu sichernden Daten an. Leider gibt es hier keine Funktion zum Durchsuchen der Festplatte, mit der Sie sich zum gewünschten Ordner durchklicken könnten. Das Eingabefeld unterstützt aber immerhin automatisches Vervollständigen, wie es man es auch von modernen Suchmaschinen kennt – so müssen Sie nicht den kompletten Pfad von Hand eintippen. Setzen Sie noch auf der Registerkarte „Teilen“ ein Häkchen vor den Raspi und bestätigen Sie den Dialog abschließend mit „Speichern“.

Wenn Sie das Webinterface von Syncthing auf dem Raspi aufrufen, sollten Sie spätestens nach ein paar Minuten den Hinweis bekommen, dass ein Rechner einen Ordner teilen möchte. Klicken Sie innerhalb dieser Benachrichtigung auf „Hinzufügen“. Auch hier müssen Sie ein paar Dinge anpassen, als Erstes natürlich den Speicherort. Hier tragen Sie den für die jeweiligen Daten vorgesehenen Unterordner ein, den Sie zuvor auf der externen Festplatte angelegt haben, in unserem Beispiel `/media/pi/Arbeitsrechner/Archivordner`.

Auf der Registerkarte „Dateiversionierung“ sollten Sie die Versionierung einschalten. Ansonsten hätten Sie keinen Schutz vor versehentlichen Dateiänderungen und vor Erpressungstrojanern, die den Inhalt Ihrer Dateien verschlüsseln – wegen seiner Echtzeit-Synchronisierung würde Syncthing auch die Inhalte Ihres Backups unwiederbringlich durch die verschlüsselten Inhalte ersetzen. Sinnvoll ist die Einstellung „Einfache Dateiversionierung“ mit den vorausgewählten fünf alten Versionen. Klicken Sie auf „Speichern“ und warten Sie ein wenig; spätestens nach ein paar Minuten sollte die Sicherung

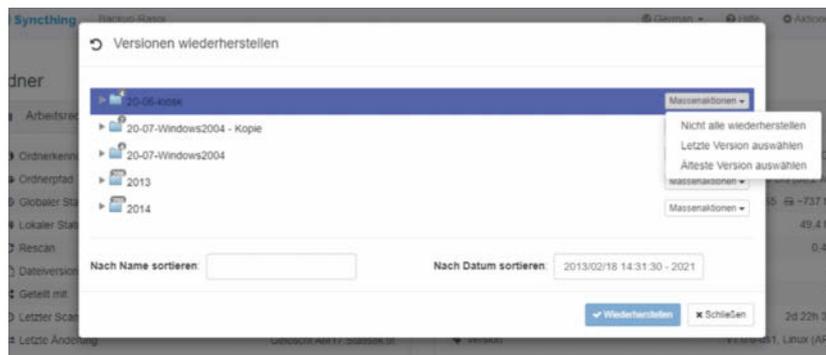
beginnen. Mit weiteren Ordnern und/oder Geräten verfahren Sie nun ebenso.

Wiederherstellung nach Löschung oder Änderung

Daten aus dem Backup wiederherzustellen läuft je nach Szenario etwas unterschiedlich ab. Für einiges hilft der Versionsverlauf. So ist es recht einfach, eine frühere Version einer Datei wiederherzustellen. Klicken Sie in Syncthing auf dem Raspi auf den Ordner eintrag und dort auf „Versionen“, und Sie bekommen eine Liste zu sehen, die alle im Versionsarchiv liegenden älteren Dateiversionen umfasst. Navigieren Sie zur gewünschten Datei, wählen aus dem Pull-down-Menü rechts eine ältere Version aus und klicken Sie am Fuß der Liste auf „Wiederherstellen“.

Ähnlich können Sie vorgehen, um gelöschte Dateien wiederherzustellen, denn die Versionierung greift auch hier. Entfernen Sie etwa auf Ihrem Arbeitsrechner Dateien, werden sie nur scheinbar auch auf dem Raspi gelöscht – tatsächlich liegen sie als Versionen im Archiv. Wurden ganze Unterordner gelöscht, können Sie die betroffenen Objekte rechts in der Versionstabelle über das Pull-down-Menü „Massenbearbeitung“ auf die „Letzte Version“ wiederherstellen. Achtung: Auf schwachen Systemen wie einem Raspberry Pi kann es nach dem Klick auf „Wiederherstellen“ so wirken, als sei Syncthings Bedienoberfläche bei der Meldung „Daten werden geladen...“ abgestürzt – lassen Sie dem Kleinen Zeit, um die ganzen Wiederherstellungen zu verarbeiten. Früher oder später sollten die gelöschten Dateien auf Ihren PC zurück synchronisiert sein.

Ein Sonderfall ist es, wenn ein Erpressungstrojaner Ihre Daten verschlüsselt hat: Auch eine solche massenhafte Änderung Ihrer Dateien wird zwar an



Mit der Versionierung kann Syncthing gelöschte Dateien oder ganze Ordner aus dem Archiv wiederherstellen.

Raspi-Backup angehübscht



Das Raspberry-Pi-4-Gehäuse Argon One M.2 ist ein Rundum-Sorglos-Paket: Außer einem aktiven Lüfter und einem HDMI-Adapter enthält es auch noch eine Schnittstelle für schnelle M.2-SSDs.

50 Euro kostet das Argon One M.2 – ganz schön heftig, schließlich gibts andere Gehäuse schon für einen Zehner. Dafür kühlt das Alugehäuse den Raspi nicht nur passiv, sondern bei Bedarf auch mit einem 30-mm-Lüfter, der über die GPIO-Pins gesteuert und mit Strom versorgt wird. Damit das klappt, installiert man mit einem Einzeiler auf der Kommandozeile ein Shell-Skript. Voreingestellt dreht der Lüfter mit 10 Prozent Leistung, sobald der Temperaturfühler auf dem Raspi-Prozessor mehr als 55 Grad Celsius meldet. Im Test passierte das nur selten; meist reichte die passive Kühlung aus und das Gehäuse blieb lautlos. Sinnvoll ist der Lüfter vor allem, um ein Heruntertakten bei zu hohen Temperaturen zu vermeiden. Für längere Zeit sprang der Lüfter während unseres Tests nur bei hoher Übertaktung und sehr leistungsintensiven Anwendungen an.

Das Lüfter-Skript installiert auch ein Konfigurationstool für den eingebauten Infrarot-Receiver – man kann den Raspi so mit der optional für 10 Euro erhältlichen Argon-Fernbedienung an- und ausschalten sowie in LibreELEC und Kodi bedienen.

Ebenfalls sehr praktisch: Das Gehäuse führt alle Schnittstellen nach hinten heraus – auf der blanken Raspi-Platine finden sich die Anschlüsse an zwei Seiten. Die beiden MicroHDMI-Anschlüsse werden direkt auf konventionelle HDMI-Buchsen geführt, sodass man nicht mehr mit Adap-

tern hantieren muss. Das Highlight ist aber die Unterstützung von M.2-SSDs (nur SATA mit B-Key oder B+M-Key-Anschluss, kein NVMe). Eine M.2-SSD wird über einen kleinen USB-Dongle außen am Gehäuse mit dem Raspi verbunden und aktiviert. Ändert man die Boot-Reihenfolge über `sudo raspi-config` und wählt unter „Advanced Options“ sowie „Boot Order“ dann „USB Boot“, startet der Raspi bei abgezogenem Dongle von der bei geschlossenem Gehäuse nicht mehr zugänglichen MicroSD und bei eingestecktem Dongle von der SSD. Der Geschwindigkeitsvorteil ist immens, im Test haben wir mit einer (schnellen!) MicroSD 46 und 37 MByte/s Lese- und Schreibgeschwindigkeit gemessen, mit einer Samsung-860-Evo-M.2 im Argon-Gehäuse waren es 171 MByte/s Lesen und 207 MByte/s Schreiben.

Den Einbau ins Gehäuse hat Argon gut dokumentiert und er gelingt auch Menschen mit unterdurchschnittlichen Bastelfähigkeiten. Man benötigt lediglich einen kleinen Kreuzschlitzschraubenzieher, alles weitere, zum Beispiel ein Silikon-Kühlpad zum Aufkleben, liegt in der Packung.

Das Argon One M.2 ist zwar teuer, aber dafür auch ein Rundum-Sorglos-Gehäuse für den Raspi 4, mit durchdachten Funktionen, das obendrein auch noch gut aussieht – und zwar sowohl auf dem Schreibtisch als auch im Wohnzimmer neben dem Fernseher. Übrigens: Wer die M.2-SSD-Funktion nicht benötigt, bekommt das Argon-One-Gehäuse für 30 Euro auch ohne. (jki)



Die Buchsen der Raspi-Platine werden allesamt nach hinten herausgeführt.

Argon One M.2

Gehäuse für Raspberry Pi 4

| | |
|-----------------|---|
| Hersteller, URL | Argon Forty, www.argon40.com |
| Systemanf. | Raspberry Pi 4, M.2-SATA-SSD (B-Key oder B+M-Key) |
| Preis | 50 € |

den Raspi übertragen, lässt sich dank Versionierung aber rückgängig machen. Allerdings sollten Sie davon absehen, die heilen Versionen Ihrer Dateien direkt auf den infizierten Rechner zurück zu synchronisieren, denn der Trojaner würde sie früher oder später abermals verschlüsseln. Besser ist es, den befallenen PC offline zu nehmen und ihn sauber neu zu installieren. Während das läuft, können Sie auf dem Raspi Ihre Dateien auf einen unverschlüsselten Zustand zurücksetzen. Dann installieren Sie Syncthing auf dem frisch installierten System und holen die Daten zurück.

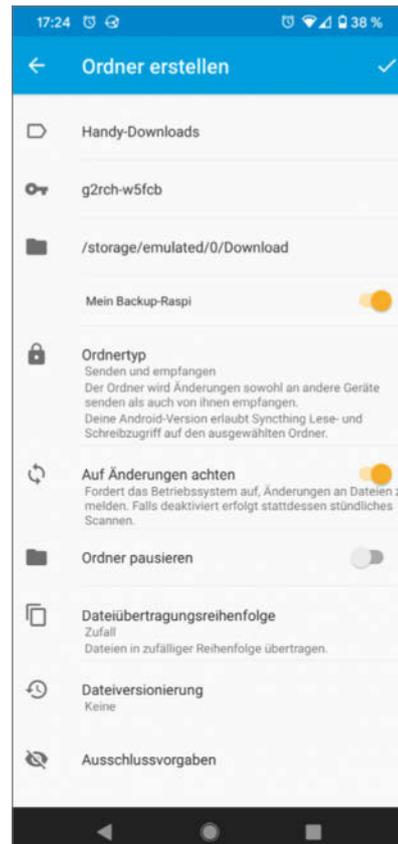
Wiederherstellung nach Hardware-Ausfall

Etwas anders läuft die Wiederherstellung ab, wenn Daten auf einem Rechner nicht gelöscht oder verändert wurden, sondern der PC komplett ausfällt. Installieren und konfigurieren Sie Syncthing wie zuvor beschrieben und klicken Sie dann in Syncthing auf dem Raspi auf den Ordner, der auf den neuen Rechner synchronisiert werden soll und dann auf „Bearbeiten“. Setzen Sie in der Registerkarte „Teilen“ ein Häkchen vor den neuen Rechner und bestätigen Sie mit „Speichern“. Nach kurzer Zeit sollten Sie in Syncthing auf dem neuen System wieder einen Hinweis auf einen geteilten Ordner bekommen, den Sie dort wie gewohnt einbinden können. Den noch bestehenden Eintrag des alten Rechners auf dem Raspi können Sie entfernen, indem Sie ihn unter „Externe Geräte“ anklicken, dann „Bearbeiten“ und links unten „Entfernen“.

Ähnlich ist die Vorgehensweise, wenn auf dem PC bloß die Festplatte mit Ihren Daten ausgefallen ist. Die einfachste Lösung ist es dann, in Syncthing auf dem PC auf den betreffenden Ordner eintrag zu klicken, dann auf „Bearbeiten“ und links unten auf „Entfernen“. Nach kurzer Wartezeit bekommen Sie den auf dem Raspi gesicherten Ordner wieder angeboten und können die Synchronisierung per Klick auf „Hinzufügen“ neu anlegen.

Mobilgeräte

Auf Android sieht Syncthing etwas anders aus: Zwar gibt es auch hier eine Web-Oberfläche, viel praktischer ist allerdings das in der App direkt enthaltene Frontend. Das sieht zwar auf den ersten Blick sehr anders aus als die Oberfläche auf Windows & Co., folgt aber der gleichen Logik: Es gibt zwei Registerkarten für „Ordner“ und „Geräte“; die Funktionen des



Ordner hinzuzufügen sieht auf Android etwas anders aus, folgt aber dem gleichen Prinzip wie auf Desktop-Rechnern.

Aktionsmenüs und die Einstellungen erreichen Sie über das Drei-Strich-Menü oben links.

Gerade für Smartphones ist die Option wichtig, dass die App unter Mobilfunkverbindungen keine Daten synchronisiert, um das Transfervolumen des Datentarifs nicht unnötig zu belasten. Dafür hat die App ein Menü: In den Einstellungen können Sie unter „Betriebsbedingungen“ festlegen, ob Syncthing auch Mobilfunkverbindungen nutzen darf. Außerdem können Sie einstellen, dass Syncthing nur in ausgewählten WLANs funktionieren soll. In Möbius Sync auf iOS/iPadOS gibt es solch ein Menü nicht – das macht nichts, denn Sie können in den App-Optionen in den iOS-Einstellungen festlegen, ob die App Mobilfunkdaten nutzen darf. (jss)



14.06. + 08.11.

Internetausfälle kompensieren

Fällt die Internetanbindung im Unternehmen aus, „steht“ oftmals der ganze Betrieb. Dieser Workshop vermittelt einen Überblick über aktuelle Techniken zu redundanten Internet-Anbindungen von Firmenstandorten.



21.– 22.06.

Post-Quantum Cryptography

In diesem Workshop tauchen wir in das Feld der Post-Quantum Cryptography ein und zeigen sowohl theoretisch als auch praktisch, wie Hersteller bereits jetzt verschlüsselnde Systeme gegen Quantencomputer absichern können.



23.06. + 06.09.
+ 15.11.

WordPress für Einsteiger

Der praxisorientierte Workshop richtet sich an Neu- und Quereinsteiger in WordPress und bietet eine grundlegende und fundierte Einarbeitung in die aktuelle Version des populären CMS.



18.10.

Dienste mit SELinux absichern

SELinux einfach abzuschalten, wenn es Probleme gibt, ist üblich, aber unklug. Der Workshop zeigt, wie man das System stattdessen so nutzt, dass alles besser abgesichert ist und trotzdem funktioniert.

Sichern Sie sich Ihren Frühbucher-Rabatt:

www.heise-events.de/workshops



Bild: Sven Hautth

WLAN-Alarmanlage

WLAN ist praktisch, birgt aber auch viele Risiken: Jeder in Funkreichweite kann Router und Clients attackieren – und zwar weitgehend unbemerkt. Ein Raspi mit Nzyme schlägt Alarm, wenn sich jemand an Ihrem Netz zu schaffen macht.

Von **Tomas Jakobs**

Drahtlose Netzwerke sind vielen Gefahren ausgesetzt: Mittels Deauthentication können Angreifer WLAN-Clients beispielsweise zwingen, die Verbindung zur Basisstation zu trennen. Das geschieht entweder aus Spielerei oder um sie zu einer Verbindung mit einem Zugangspunkt unter Kontrolle der Angreifer zu animieren, der das WLAN-Netz der ursprünglichen Basisstation imitiert. Das Ziel ist, den Datenverkehr auszulesen oder zu manipulieren (Man-in-the-middle-Attacke). All das ist längst kein Hexenwerk mehr, es gibt sogar einsatzfertige Hacking-Gadgets, die Standardangriffe auf Knopfdruck ausführen [1].

Normalerweise bekommt man von WLAN-Attacken bestenfalls etwas mit, wenn die Angreifer längst ins Netzwerk eingestiegen sind. Doch Sie können vorbeugen und mit Nzyme aktiv nach auf-

fälligen WLAN-Aktivitäten scannen. So bekommen Sie schnell mit, wenn etwa ein neues Netz in der Umgebung auftaucht, das Ihrem verdächtig ähnelt oder Ihre Geräte fremdgesteuert angewiesen werden, die Verbindung zu kappen. Die nötige Hardware kostet nur ein paar Euro und einen passenden Raspi haben Sie vielleicht sogar schon in der Schublade.

Das Open-Source-Projekt Nzyme fungiert als Wireless Intrusion Detection System (WIDS), also als Alarmanlage für Drahtlosnetzwerke. Ende vergangenen Jahres erschien die Version 1.2.2. Das Release bietet eine gute Gelegenheit zu erklären, wie Sie Nzyme in Betrieb nehmen und das eigene WLAN vor neugierigen Augen schützen. Entdeckt Nzyme Anomalien im Funkverkehr, warnt es in der Weboberfläche und versendet auf Wunsch E-Mails an den Administrator.

Frame-Sammler

Nzyme arbeitet im Hintergrund als Sniffer für die sogenannten WLAN-Management-Frames. Drahtlosgerätee senden Management-Frames meist unverschlüsselt, um Informationen auszutauschen. Sie machen so beispielsweise auf sich aufmerksam oder bereiten einen WPA-Handshake vor. Der Unterschied zu den bekannten Sniffern aus Pentesting-Werkzeugsammlungen liegt in der konsequenten Ausrichtung auf automatisiertes Schnüffeln und Warnen. Im Idealfall wird Nzyme stationär auf die Lauer gelegt und sich selbst überlassen. Fachkenntnisse über den Aufbau von WLAN-Paketen sind für den Betrieb von Nzyme nicht nötig. Die Software überwachte für diesen Artikel über einen Monat das heimische Netzwerk des Autors und konnte die eigenen Hackingversuche erfolgreich aufspüren.

Der Nzyme-Entwickler Lennart Koopmann hat das Tool für den Raspi konzipiert. Der Dokumentation zufolge reicht bereits ein Raspi 3 für kleine Heimnetze. Für diesen Artikel diente ein Raspi 4 mit 4 GByte Arbeitsspeicher als Testgerät. Bei Mesh-Drahtlosnetzwerken in Unternehmen empfehlen sich mehrere Raspis mit Nzyme für eine bessere Abdeckung der größeren Fläche. Diese können ihre Daten zur Auswertung an einen zentralen Log-Server im internen Netz schicken, beispielsweise an Gray-



Für Nzyme genügt ein Raspberry Pi 3 oder neuer. Dazu brauchen Sie einen WLAN-USB-Stick, der den Monitor-Modus beherrscht.

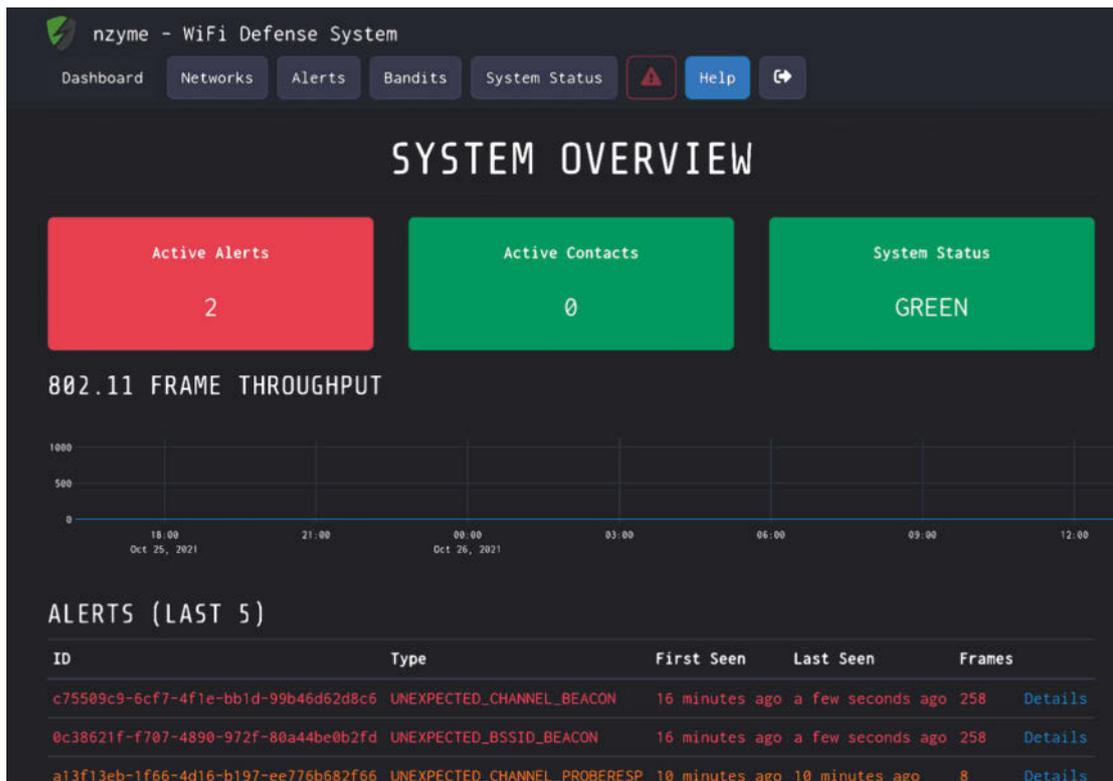
log, das vom gleichen Entwickler stammt. So kann man kostengünstig auch große Netze scannen.

Zur Überwachung der Funkfrequenzen ist ein WLAN-Modul nötig, das den sogenannten Monitor

WLAN-Management-Frames

Zur Überwachung des eigenen Netzes sammelt Nzyme WLAN-Management-Frames ein und meldet verdächtige Vorkommnisse, siehe dazu den Kasten „Verdächtige WLAN-Aktivitäten“ auf Seite 70. Neben Management-Frames gibt es auch weitere Arten von Frames, wie „control“, „extension“ und „data“, die für Nzyme nicht relevant sind. Management-Frames werden in der Regel unverschlüsselt gesendet und lassen sich in folgende Subtypen einteilen. Die Nzyme-Dokumentation (ct.de/wfrn) listet die sicherheitsrelevanten Management-Frames auf:

- **Authentication** ist der erste Schritt, um einem Drahtlosnetzwerk beizutreten. Je nach Verschlüsselungsverfahren (WEP, WPA) wird eine bestimmte Authentifizierungssequenz eingeleitet.
- Der **Association Request** folgt auf Authentication und fragt Ressourcen an der Basisstation an.
- Darauf folgt die **Association Response**. Wenn erfolgreich, dann darf das assoziierte Gerät weitere Frame-Arten senden, beispielsweise „data“ für Datenübertragung.
- Mit **Deauthentication** verlangt ein Drahtlosgerät die Trennung der Verbindung. Deauthentication muss von den anderen Geräten akzeptiert werden.
- **Beacon**: Basisstationen senden durchgängig solche Frames, um auf sich aufmerksam zu machen.
- **Probe Request**: Ein Drahtlosgerät sucht damit nach bekannten WLAN-Netzen.
- Zugangspunkte senden eine **Probe Response** als Antwort auf einen Probe Request und bestätigen ihre Anwesenheit.



In der intuitiv bedienbaren Weboberfläche ist das Dashboard die erste Anlaufstelle. Dort erfassen Sie auf einen Blick den Zustand des Systems. Verdächtige Aktivitäten springen direkt ins Auge.

Mode unterstützt. Ein preisgünstiger WLAN-USB-Stick, etwa mit Ralink-RT-Chipsatz, reicht schon aus. Wer nach einem Verwendungszweck für ältere, nicht mehr gebrauchte WLAN-Sticks sucht, kann mehrere davon an seinem Raspi für Nzyme nutzen. Das bringt mehr WLAN-Management-Frames in kleineren Zeitabständen und führt zu einem besseren Lagebild, da Nzyme auf mehreren Frequenzen parallel schnüffeln kann. Im Praxistest kamen ein älterer Panda-Wireless-USB-Adapter (PAU06) sowie ein etwas moderneres Handelsware-No-Name-Gerät mit RT5572-Chipsatz für gerade einmal 15 Euro zum Einsatz. Darüber hinaus sollte Nzyme via Ethernetkabel mit dem lokalen Netzwerk verbunden sein, um seine Weboberfläche bereitzustellen und vor Anomalien im Drahtlosnetzwerk zu warnen.

Installation

Nzyme installieren Sie wahlweise unter Debian, Ubuntu Server oder Raspberry Pi OS. Der Fokus liegt

hier auf der Installation unter Debian 10 „Buster“. Mit Debian 11 „Bullseye“, kam es im Testlauf zu Problemen mit dem Channel-Hopping. Bei anderen Distributionen oder WLAN-Hardware können die einzelnen Installationsschritte abweichen und müssen angepasst werden. Grundkenntnisse und Transfervermögen im Umgang mit einem Linux-System werden vorausgesetzt. Debian-Images für den Raspberry Pi finden Sie im Downloadbereich der Debian-Webseite, die wir unter ct.de/wfrn verlinkt haben.

Nach der Debian-Grundinstallation und Sprachanpassungen bietet es sich an, den SSH-Server in Betrieb zu nehmen und von einem anderen Rechner aus mit dem Raspi zu arbeiten. Dazu sollten Sie unter Debian ein Passwort für root vergeben und in der Datei `/etc/ssh/sshd_config` den Login via Passphrase erlauben (`PermitRootLogin yes`), falls Sie sich nicht mit SSH-Schlüsseln authentifizieren. Für Raspberry Pi OS konfigurieren Sie den SSH-Zugang mit dem Raspi-Imager-Tool vor [2]. Das Debian-Raspi-Image

bringt nur den Nutzer root mit, unter Raspberry Pi OS setzen Sie „sudo“ vor die Befehle, um sie mit Systemverwalterrechten auszuführen. Besorgen Sie mit dem nachfolgenden Befehl zunächst die nötigen Software-Pakete und die passende Firmware:

```
apt install libpcap0.8 sudo wget iw \
openjdk-11-jre-headless \
wireless-tools \
postgresql-13 iw \
firmware-ralink apt-transport-https -y
```

Falls Sie einen anderen WLAN-Chipsatz anstelle des hier verwendeten Ralink-Chipsatzes nutzen, dann ersetzen Sie „firmware-ralink“ durch das Paket des Herstellers. Für Realtek wäre das beispielsweise „firmware-realtek“.

Stellen Sie im Anschluss sicher, dass das WLAN-Interface des Sticks, mit dem Sie scannen wollen, mit einem eindeutigen Namen ansprechbar ist. Einen Überblick über relevante Netzwerkinterfaces verschaffen Sie sich mit `iwconfig`. Tauchen in dessen Ausgabe die generischen Interface-Bezeichnungen `wlan0` und `wlan1` auf, dann müssen Sie Hand anlegen und feste Gerätenamen setzen. Vergeben Sie dazu zwei udev-Regeln im Verzeichnis `/etc/udev/rules.d`:

```
cat <<EOF >>/etc/udev/rules.d/\
70-persistent-net.rules
SUBSYSTEM=="net", ACTION=="add", \
DRIVERS=="?* ", \
ATTR{address}=="XX:XX:XX:XX:XX:XX", \
ATTR{dev_id}=="0x0", \
ATTR{type}=="1", \
KERNEL=="wlan*", NAME="IDSWLAN"
EOF
```

Falls Sie Nzyme unter Raspberry Pi OS installieren, benötigen Sie die udev-Regeln nicht und es genügt, den Symlink `/etc/systemd/network/99-default.link` zu löschen und den Raspi neu zu starten. Das System generiert dann automatisch einen festen Namen mittels der MAC-Adresse des Adapters, beispielsweise „wlx00c0ca971201“. Ersetzen Sie unter Debian im vorigen Befehl die XX-Platzhalter durch die MAC-Adresse des eigenen WLAN-Sticks, die Sie mit `ip link show` ermitteln. Der Name des Gerätes sollte einmalig sein und nicht in Konflikt mit anderen Geräten stehen. Verwenden Sie einen Namen ohne Bindestriche, Unterstriche und andere Sonderzeichen, denn die stören Nzyme beim Kanalwechsel. Als Beispielname

für das Interface dient hier der Name „IDSWLAN“, mit dem Sie auf der sicheren Seite sind.

Nachdem Sie die erste Regel gesetzt haben, starten Sie den Raspi neu und vergeben Sie im Anschluss die zweite Regel:

```
cat <<EOF >>/etc/udev/rules.d/\
75-persistent-net-generator.rules
KERNEL!="wlan*|ath*|msh*|ra*|sta*|\
ctc*|lcs*|hsi*", \
GOTO="persistent_net_generator_end"
EOF
```

Nach einem weiteren Neustart sollte das WLAN-Interface unter dem Namen „IDSWLAN“ ansprechbar sein.

Im Anschluss gilt es, die PostgreSQL-Datenbank einzurichten, die Nzyme benötigt. Legen Sie dazu eine Datenbank mit Namen „nzyme“ und den Datenbank-User „nzyme“ an. Die Sternchen ersetzen Sie durch ein eigenes Passwort, das Sie sicher aufbewahren sollten.

```
sudo -Hsu postgres psql -c ↵
↳"create database nzyme;"
sudo -Hsu postgres psql -c ↵
↳"create user nzyme with encrypted ↵
↳password '*****';"
sudo -Hsu postgres psql ↵
↳-c "grant all privileges on ↵
↳database nzyme to nzyme;"
```

PostgreSQL bestätigt Ihnen die jeweilige Anweisung mit den Ausgaben „CREATE DATABASE“, „CREATE ROLE“ und „GRANT“. Als letzten Schritt vor der Installation des Nzyme-Paketes erstellen Sie mit dem Befehl `echo -n XXXXXX | sha256sum` ein Passwort-Hash zur späteren Anmeldung am Web-Interface. Ersetzen Sie XXXXXX durch das gewünschte Passwort. Legen Sie den Hash (ohne den Bindestrich am Ende) für späteres Copy & Paste in eine temporäre Datei im Home-Verzeichnis ab.

Laden Sie nun mittels `wget` das Debian-Paket der aktuellen Nzyme-Version herunter. Bei Redaktionsabschluss war das die Version 1.2.2 „Peck Slip“:

```
wget https://assets.nzyme.org/\
releases/nzyme-1.2.2.deb
```

Gleichen Sie die SHA256-Prüfsumme des Pakets mit der Zahlenfolge auf der Nzyme-Projektseite auf (`ct.de/wfrn`), um die Echtheit des Pakets zu prüfen.

Fahren Sie mit der Installation via `dpkg` nur fort, wenn die Prüfsumme stimmt:

```
sha256sum nzyme-1.2.2.deb
dpkg -i nzyme-1.2.2.deb
```

Nzyme konfigurieren

Nzyme kommt mit einer kommentierten Konfigurationsvorlage, die Sie einfach übernehmen und als Grundlage nutzen können:

```
cp /etc/nzyme/nzyme.conf.example \
/etc/nzyme/nzyme.conf
```

Ersetzen Sie nun mit einem Texteditor Ihrer Wahl die Beispielwerte in der Datei `/etc/nzyme/nzyme.conf` durch Ihre eigenen.

```
general: {
  role: LEADER
  id: nzyme-node-01
  admin_password_hash:
  database_path: "postgresql://localhost:5432/nzyme?user=nzyme&password=YOUR_PASSWORD"

  python {
    path: /usr/bin/python3.8
    script_directory: /tmp
    script_prefix: nzyme_
  }
}
```

Geben Sie im Abschnitt `general` Ihrer Nzyme-Instanz bei `id` einen eindeutigen Namen, der nur aus alphanumerischen Zeichen besteht. Tragen Sie bei `admin_password_hash` Ihren zuvor erstellten Password-Hash ein, den Sie in einer temporären Datei abgelegt haben. Der Punkt `database_path` zeigt den PostgreSQL-Verbindungsstring. Ersetzen Sie dahinter den Platzhalter `YOURPASSWORD` durch das eigene Datenbank-Passwort.

Gleichen Sie im Abschnitt `python` bei `path` die Vorgabe `/usr/bin/python3.8` mit Versionsnummer und Pfad der installierten Python-Version ab und korrigieren Sie falls nötig. Viele Distributionen setzen hier symbolische Verknüpfungen. Es ist ratsam, besser direkt auf das Python-Executable zu verweisen.

Den Abschnitt `alerting` mit den Angaben für E-Mail-Benachrichtigungen überspringen Sie zunächst. Bevor Nzyme bei jedem Alert Mails versendet, sollten Sie das System erstmal feintunen. Bei der ersten

Inbetriebnahme kommt es üblicherweise zu vielen False-Positives.

```
interfaces: {
  rest_listen_uri:
  http_external_uri:
  http_listen_uri:
}

802_11_monitors: [
  {
    device: wlan0
    channels:
  }
]
```

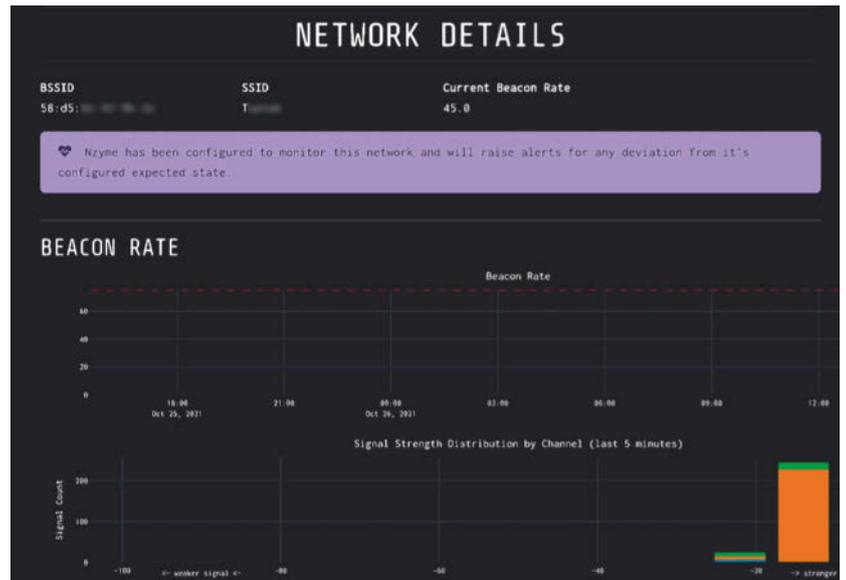
Ersetzen Sie im Abschnitt `interfaces` die Vorgabe `127.0.0.1` durch die IP-Adresse des Raspi-LAN-Interfaces, unter der er im lokalen Netz erreichbar ist. Prüfen Sie außerdem, ob der Raspi von einem DHCP-Server im Netzwerk eine fest zugewiesene IP-Adresse erhält. Von besonderem Interesse ist der Abschnitt `802_11_monitors`. Überschreiben Sie dort bei `devices` den Platzhalter „`wlan0`“ mit dem fest gesetzten Namen des Interfaces, beispielsweise „`IDSWLAN`“. Die Liste der abzusuchenden Kanäle steht kommasepariert unter `channels`. Für das 2,4-GHz-Band sind das zum Beispiel die Kanäle von 1 bis 14. Eine vollständige Liste aller verfügbaren Kanäle eines WLAN-USB-Sticks verschafft Ihnen der Befehl `iwlist channel`.

Zum Schluss machen Sie Nzyme `systemd` als Dienst bekannt und führen die Software aus. Zukünftig startet Nzyme nach einem Neustart des Raspi automatisch:

```
systemctl enable nzyme
systemctl start nzyme
```

Nach einer kurzen Wartezeit steht das Nzyme-Webinterface unter der IP-Adresse des Raspi und dem Port 22900 zur Verfügung. Die Anmeldung erfolgt mit dem Benutzernamen „`admin`“ und dem zuvor bestimmten Passwort. Falls Nzyme nicht startet, besteht wahrscheinlich ein Problem mit der Konfigurationsdatei. Der Befehl `journalctl -xe` und

Der Menüpunkt „Network Details“ listet Informationen wie Beacon-Rate und Signalstärke auf. Wenn Sie von hier die Werte für SSID, BSSID, Beacon-Rate und Network-Fingerprint in die Nzyme-Konfigurationsdatei kopieren, beginnt Nzyme, das Netzwerk zu überwachen.



ein Blick in die Logdatei `/var/log/nzyme/nzyme.log` helfen bei der Fehlersuche.

Nzyme auf das eigene Netzwerk ansetzen

Nun gilt es, Nzyme mit dem eigenen Drahtlosnetzwerk bekannt zu machen, damit es überwacht werden kann. Wechseln Sie dazu nach der Anmeldung im Web-Interface zum Hauptmenüpunkt „Networks“ und öffnen dort die „Network Details“ für Ihre Drahtlosnetzwerke. Dort lesen Sie die Angaben wie SSID, BSSID, Beacon-Rate und Netzwerk-Fingerprint ab und speichern sie zwischen. Damit ergänzen Sie die Informationen in der Nzyme-Konfigurationsdatei im Abschnitt `802_11_networks`.

```

ssid: mywifinetwork
channels: 3
[1,2,3,4,5,6,7,8,9,10,11,12,13]
security: [WPA2-PSK-CCMP]
beacon_rate: 40
bssids: [
  {
    address: "f0:9f:c2:dd:18:f6",
    fingerprints: [ 45933326ca0 ]
  }
]

```

Nach einem Neustart des Dienstes kennt Nzyme fortan ihr eigenes Netzwerk und dessen „Soll-Zustand“. Registriert Nzyme Abweichungen davon, zeichnet es diese auf und schlägt Alarm. Registriert Nzyme beispielsweise eine erhöhte Beacon-Rate, könnte dies darauf hindeuten, dass ein Angreifer versucht Ihren Access-Point zu imitieren (Spoofing). Zu jedem Alert informiert Nzyme über das mögliche Angriffsszenario, liefert aber zur besseren Einordnung auch Erklärungen für „natürliche Ursachen“ der Anomalie.

Alles im Blick

Das Dashboard ist die erste Anlaufstelle im Nzyme-Webinterface und zeigt den Status des Systems sowie mögliche Probleme an. Farbliche Statusanzeigen, Schnellübersichten der letzten Ereignisse und ein Graph über die vergangenen 24 Stunden verschaffen einen schnellen Situationsüberblick. Detailinformationen zu erkannten Drahtlosnetzwerken und Basisstationen finden Sie unter dem Hauptmenüpunkt „Networks“. Nzyme erstellt zum Unterscheiden einen möglichst eindeutigen Fingerprint. Dabei kann es vorkommen, dass einem Gerät mehrere davon zuteil werden, je nachdem, auf welchen Kanälen es sendet oder für welche Drahtlosnetzwerke es verantwortlich ist.

nzyme - WiFi Defense System

Dashboard Networks Alerts Bandits System Status Help

NETWORKS

Filter Reset Networks

| BSSID | Advertised Networks | OUI | SEC | FP | WPS |
|------------------|---------------------|--|---------------|----|-----|
| 58:d5 -15 dBm | T | D-Link International | WPA1, WPA2 | 1 | ☑ |
| 58:d5 -30 dBm | T | D-Link International | WPA1, WPA2 | 1 | ☑ |
| 44:4e -59 dBm | F | AVM Audiovisuelles Marketing und Computersysteme GmbH | WPA2 | 1 | ☑ |
| dc:15 -61 dBm | F | AVM Audiovisuelles Marketing und Computersysteme GmbH | WPA2 | 1 | ☑ |
| 5c:49 -66 dBm | S | AVM Audiovisuelles Marketing und Computersysteme GmbH | WPA2 | 1 | ☑ |

Unter dem Menüpunkt „Networks“ listet Nzyme alle drahtlosen Netzwerke in Reichweite.

Eine Gesamtübersicht aller von Nzyme registrierten „Alerts“ findet sich unter dem gleichnamigen Hauptmenüpunkt. Lassen Sie sich anfangs von den vielen Alarmmeldungen nicht verunsichern. Wie jedes Messinstrument braucht ein frisch installiertes

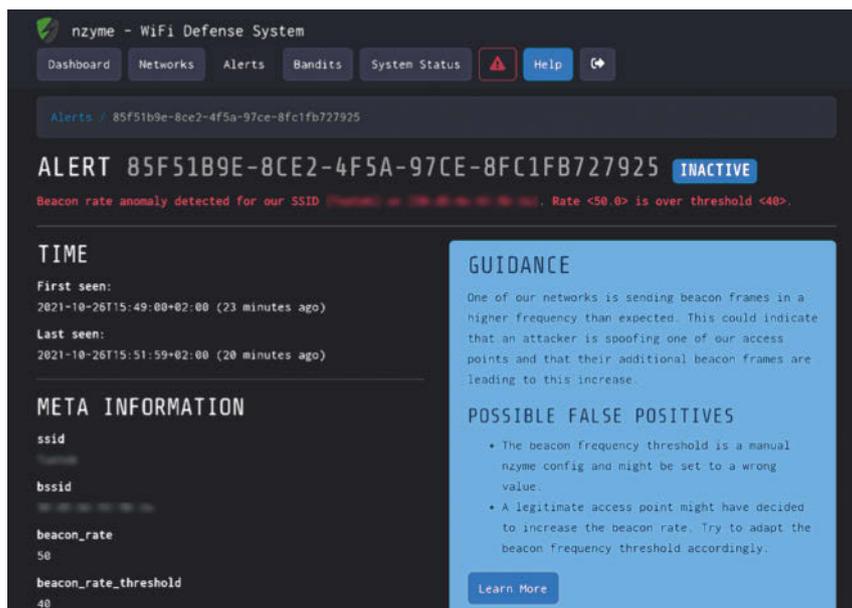
Nzyme ein wenig Feinkalibrierung und Zeit, bis die Anzahl falsch positiver Alarme sinkt. Dazu passen Sie beispielsweise den Schwellenwert der Beacon-Rate, den Sie als üblich beobachtet haben, in der Konfigurationsdatei entsprechend an. Mit etwas

Verdächtige WLAN-Aktivitäten

Nzyme behält die eigenen Drahtlosnetzwerke im Blick und sammelt möglichst viele WLAN-Management-Frames ein. Die folgenden Anomalien weisen auf einen möglichen Angriffsversuch hin:

- Neue Zugangspunkte, die legitime Access-Points nachahmen (Rogue-AP), fallen durch minimale Unterschiede in den Zeitstempeln auf.
- Nutzt das eigene Netzwerk plötzlich abweichende Kanäle, bricht die Verbindung ab oder ändert den Verschlüsselungsmodus, kann das auf einen aktuellen Angriff hindeuten.
- Durch die Organizationally Unique Identifier (OUI), die ersten Bytes der MAC-Hardware-Adresse, werden populäre Hacker-Werkzeuge wie Pineapples oder ESP8266-Deauther identifiziert.
- Deauth-Frames, die Endgeräte von Basisstationen trennen.
- Anomalien bei Signalstärken, etwa zusätzlich aufgestellte Access-Points oder physisch manipulierte Basisstationen, beispielsweise abmontierte Antennen.
- Unterschiedliche oder unerwartete Probe Responses bekannter Geräte.

Nzyme schlägt Alarm: Zu jedem Alert liefert die Software eine Erklärung, warum die Anomalie auf einen Angriff hindeutet. Zur besseren Einordnung gibt es auch einen Hinweis, warum es sich um einen False Positive handeln könnte.



Herumprobieren sind die idealen Schwellenwerte für die Beacon-Rates der eigenen Basisstationen nach einigen Tagen gefunden, die naturgemäß etwas schwanken. Erst wenn die Werte sich etabliert haben und False-Positive-Alerts zurückgehen, sollten Sie die E-Mail-Benachrichtigungen einrichten.

Wanted: Hacking-Gadgets und andere Banditen

Als Banditen bezeichnet Nzyme Geräte, die anhand ihrer Hersteller-Identifikation als typische Hacker-Werkzeuge wie beispielsweise Wifi-Pineapple oder Deauther erkannt werden. Eine Reihe davon haben wir in c't bereits unter die Lupe genommen [1]. Neben der vorgefertigten Liste von Nzyme können Sie aber auch selbst Banditen definieren und nachverfolgen.

Bei der Kombination unterschiedlicher WLAN-USB-Sticks kann es zu Problemen bei dem automatischen Kanalwechsel und dem Versetzen in den Monitor-Mode kommen. Ist das bei Ihnen der Fall, meldet Nzyme dies sowohl im Log als auch im Web-Interface. Ein Wechsel des WLAN-Dienstes von wpa_supplicant zu iwfd kann helfen. Die Dokumentation des Nzyme-Projektes, die wir unter ct.de/wfrn verlinkt haben, bietet Hilfestellung bei weiteren gängigen Problemen.

Weitere Anpassungen

Nzyme schützt Ihre Drahtlosnetzwerke, indem es verdächtige Aktivitäten aufspürt. Vergessen Sie jedoch nicht, auch Nzyme selbst zu schützen. Web-Interface und SSH-Zugang sollten nur aus einem verkabelten Management-Segment des Netzwerks erreichbar sein. Außerdem sollten Sie Transportverschlüsselung mittels TLS aktivieren. Dazu hinterlegen Sie die Pfade zu den Zertifikaten in der Konfigurationsdatei.

Zur großflächigen Abdeckung stellen Sie mehrere Nzyme-Raspis auf, die ihre gesammelten Daten einem zentralen Log-Management melden.

Fazit

Nzyme ist als Wifi-Intrusion-Detection-System (WIDS) eine sinnvolle Ergänzung zu bewährten IDS-Lösungen wie Snort oder Suricata. Es wird zum Auge und Ohr für jeden Admin von Drahtlosnetzwerken. Im Selbsttest konnte Nzyme sowohl einen simulierten Lauschangriff als auch die Imitation des eigenen Access-Point mittels SSID-Spoofing (Rogue-AP-Angriff durch Wifi Pineapple) erkennen und per Mail Alarm schlagen. Seine Stärke spielt Nzyme als automatisierte „Datenpumpe“ für eine zentrale Log-Auswertungssoftware wie Graylog aus. (ndi) **ct**

Literatur

[1] Ronald Eikenberg und David Wischnjak, **Böse und billig**, Hacking-Gadgets, Gefahr durch angriffslustige Hardware, c't 18/2017, S. 64

[2] Ronald Eikenberg, **Raspi-Schnellstart**, Raspberry Pi superschnell einrichten durch Vorkonfiguration, c't 11/2021, S. 132

Nzyme-Projektseite, Dokumentation und Downloadbereich der Debian-Webseite

ct.de/wfrn



Fernbedienung fürs Game-Streaming

Mit der Tastenkonzole Stream Deck können Sie bequem Game-Streams steuern – und schnell reagieren, wenn irgendwelche Spinner Hass und Hetze in Chats verbreiten. Ein Raspi vermittelt dabei zwischen Tastenkonzole und PC oder Spielekonsole.

Von **Mirko Dölle**

Es ist eben eine Frage des Gefühls: Gaming-Tastaturen haben einen fühlbaren Druckpunkt, bei Joysticks vermitteln die Mikroschalter die Rückmeldung und Game-Controller-Tasten lassen sich mitunter an ihrer Form erkennen – schließlich soll niemand den Blick vom Spielgeschehen abwenden müssen, um die richtige Taste zu finden.

Mit der Tastenkonzole Stream Deck von Elgato gibt es ein Werkzeug, das beim Game-Streaming eine

spürbare Verbesserung verspricht: Die verschiedenen Modelle der Konsole stellen bis zu 32 frei programmierbare, mit einem TFT-Display hinterleuchtete Tasten bereit, mit denen Sie während eines Game-Streams die Kameraperspektive wechseln, das Mikrofon stumm oder den Chat in den Emoji-Modus schalten können. Ähnlich wie bei einer Tastatur sind die Tasten erhaben, sodass Sie nach etwas Eingewöhnung gar nicht mehr hinsehen müssen, sondern

sich ganz auf Ihr Fingerspitzengefühl verlassen können. Mit einem Raspberry Pi als Steuerungscomputer wird diese Lösung plattformunabhängig und lässt sich mit Gaming-PCs, Spielekonsolen und in Videostudios einsetzen.

Die hier vorgestellte Stream-Fernbedienung besteht aus den drei Komponenten Raspberry Pi 4B (ab 4 GByte RAM, Speicherkarte mit 16 GByte oder mehr), der Tastenkonsole Elgato Stream Deck und dem Steuerungsprogramm Bitfocus Companion. Falls Sie zunächst nicht so viel Geld ausgeben möchten – ein Stream Deck kostet zwischen 80 und 225 Euro – können Sie Companion mit dem integrierten Emulator etwa auf einem Tablet im Browser ausprobieren – dann allerdings ohne die Haptik.

Bei Gaming-PCs ist der Raspi verzichtbar, Sie können Companion auch direkt auf Ihrem Rechner installieren. Die Windows- und macOS-Varianten bekommen Sie nach der Registrierung auf bitfocus.io kostenlos zum Download; unter Linux verwenden Sie die nachfolgende Anleitung für den Raspberry Pi als Blaupause, um Companion aus dem GitHub-Repository zu installieren. Der Vorteil eines Raspi ist, dass er autark von Ihrem Gaming-Rechner arbeitet – Sie können selbst dann im Handumdrehen

eine Werbepause auf Twitch schalten, wenn sich Ihr Gaming-PC aufgehängt hat.

Umfangreich

Welchen Funktionsumfang Sie mit Companion nutzen können, hängt von der Spieleplattform ab: Wenn Sie direkt von der PlayStation oder Xbox streamen, können Sie die Mikrofon- und Kameraeinstellungen nicht mit dem Stream Deck anfassen – die Konsolen bieten hierfür keine Fernsteuerungsfunktionen. Dann bleibt Ihnen nur, den YouTube-Stream oder den Twitch-Chat fernzubedienen. Das können Sie auch aus dem YouTube-Live-Studio oder dem Twitch-Stream-Manager heraus erledigen, doch die Bedienung des Browser-Frontends mit der Maus dauert deutlich länger und lenkt viel stärker ab, als auf dem Stream Deck eine dedizierte Taste zu drücken.

Besser sieht es bei Gaming-PCs und im Videostudio aus: Das OBS-Plug-in von Companion hat außerordentlich viele Funktionen, Sie können damit nicht nur zwischen den vordefinierten Szenen umschalten, sondern auch die Lautstärke von Audioquellen anpassen, Videoquellen ab- und zuschalten, Medien abspielen und über zahlreiche Feedback-Funktionen den tatsächlichen Zustand von Quellen, Streams und anderem ermitteln.

Die Bildmischer in Videostudios lassen sich mit dem Atem-Plug-in weitreichend ansteuern, auch für digitale Mischpulte von Allen & Heath, Behringer, Midas und Yamaha gibt es etliche Plug-ins – aktuell gehören knapp 300 zum Standardumfang von Companion. Ist der Raspberry Pi via WLAN mit dem Studionetzwerk verbunden, kann das Stream Deck auch vor der Kamera etwa als Räuspertaste benutzt werden oder falls man mal niesen muss. Genügend Tasten vorausgesetzt können Sie sogar einen Studio-Livestream ohne den sonst obligatorischen Videoproducer bestreiten.

Damit wäre das Thema angeschnitten, welches Stream-Deck-Modell man denn braucht. Es gibt drei Varianten mit 6, 15 und 32 Tasten für rund 80, 130 und 225 Euro. Die sechs Tasten der Mini-Version sind eigentlich immer zu wenig, außer das Stream Deck soll etwa nur als Mikrofontaste an einem Interviewtisch dienen. Mit 15 Tasten kommt man in der Regel gut aus – eigentlich. Denn die zahlreichen Feedbacks der Plug-ins erlauben es, das Stream Deck auch als Kontrolltafel für die wichtigsten Systemparameter mitzubeneutzen. Sogar die Tastenbeschriftung lässt sich dynamisch anpassen. Dann wird es jedoch schnell eng mit nur 15 Tasten respektive Displays.



Dem Stream-Deck-Emulator fehlt auf einem Tablet zwar die Haptik, erspart einem aber die Investition in eine der Tastenkonsolen, falls man Companion erst einmal nur ausprobieren möchte.

Die XL-Version mit 32 Tasten bietet hingegen genug Platz, um Statusanzeigen und Bedientöpfe sogar thematisch zu gruppieren.

Ein Clou von Companion: Die Software erlaubt es, mehrere Stream Decks, auch unterschiedlicher Größe, gleichzeitig anzuschließen. So kann der Moderator einer Talkrunde das Standardmodell mit 15 Tasten vor sich stehen haben, während sich die Gäste ein oder zwei Stream Deck Mini teilen und dementsprechend nur einen begrenzten Funktionsumfang nutzen können. Über das Web-Frontend kann außerdem gleichzeitig jemand hinter den Kulissen den Emulator mit einem virtuellen Stream Deck XL auf seinem Tablet aufrufen und hat so noch mehr Eingriffsmöglichkeiten.

Frei und unabhängig

Damit Sie die Fernsteuerung möglichst flexibel einsetzen können, empfehlen wir eine WLAN-Anbindung für den Raspi. Companion gibt es auch als fertiges Image, wir haben uns aber für die manuelle Installation unter Raspberry Pi OS mit grafischem Desktop entschieden. Das lässt mehr Möglichkeiten für eigene Anpassungen und erhöht die Flexibilität: So können Sie jederzeit Tastatur, Maus und Monitor anschließen und Companion über den Chromium-Browser direkt auf dem Raspi anpassen. Das ist zum Beispiel für das YouTube-Plug-in erforderlich, doch dazu später mehr.

Beginnen Sie damit, Raspberry Pi OS herunterzuladen und mittels Balena Etcher von etcher.io oder dem Raspberry Pi Imager auf eine MicroSD-Karte zu schreiben. Karte raus und wieder rein, damit die neue Partitionierung erkannt wird, danach können Sie den Zugang zum WLAN konfigurieren. Eine Vorlage für die Datei `wpa_supplicant.conf` haben wir Ihnen auf ct.de/w1w9 zum Download bereitgestellt, Sie müssen nur noch die SSID und die Passphrase Ihres WLANs eintragen und die Datei dann auf der ersten Partition der MicroSD-Karte speichern. Zur Orientierung: Sie trägt das Label `boot` und enthält unter anderem die Datei `config.txt`. Bei dieser Gelegenheit sollten Sie auch gleich den SSH-Zugang zum Raspi aktivieren, indem Sie im gleichen Verzeichnis eine leere Datei mit dem Namen `ssh` anlegen (Details zur Einrichtung finden Sie auch ab Seite 26).

Den ersten Systemstart des Raspi führen Sie idealerweise mit Tastatur, Maus und Monitor durch und nutzen die grafische Konfiguration, um Sprache und Tastaturbelegung anzupassen. Verpassen Sie Ihrem Raspi auch einen neuen Hostnamen, zum

Beispiel `companion` – moderne (DSL-)Router beherrschen die lokale Namensauflösung, sodass Sie den Raspi einfach über seinen Namen ansprechen und nicht umständlich die lokale IP-Adresse heraussuchen und eingeben müssen. Nach Abschluss der Erstkonfiguration startet der Raspi neu und Sie können es sich aussuchen, ob Sie die nächsten Schritte per SSH von einem anderen Rechner aus oder weiterhin direkt am Raspi erledigen.

Aufgezogen

Die Installation von Companion erfolgt aus dem GitHub-Repository von Bitfocus. Eine Liste aller auszuführenden Befehle finden Sie auch auf ct.de/w1w9 zum Download. Zunächst müssen Sie einige Entwicklungspakete installieren und außerdem die Paketquellen für Node.js 14 hinzufügen:

```
sudo apt-get install libgusb-dev \
git build-essential cmake \
libudev-dev libusb-1.0-0-dev curl
curl -fsSL https://deb.nodesource.com/setup_14.x | sudo -E bash -
```

Das Setup-Skript legt die Paketquellen für Node.js im Verzeichnis `/etc/apt/sources.list.d` ab und aktualisiert die Paketlisten, sodass Sie unmittelbar Node.js und dann den Paketmanager Yarn installieren können:

```
sudo apt-get install nodejs
sudo npm install yarn -g
export PATH="$HOME/.yarn/bin:$
$HOME/.config/yarn/global$
node_modules/.bin:$PATH"
```

Die nächsten Schritte sind, Companion aus dem GitHub zu klonen, zu übersetzen und zu installieren:

```
git clone https://github.com/
bitfocus/companion.git
cd companion
yarn update
./tools/build_writefile.sh
```

Damit Companion künftig automatisch bei jedem Neustart des Raspi ebenfalls anläuft, kopieren Sie den Systemd-Job `companion.service`, den wir Ihnen unter ct.de/w1w9 bereitgestellt haben, in das Verzeichnis `/etc/systemd/system`. Anschließend aktivieren und starten Sie den Job:

```
sudo systemctl daemon-reload
sudo systemctl enable companion
sudo systemctl start companion
```

Oberflächenstruktur

Damit ist die Installation von Companion abgeschlossen, ab sofort können Sie das Web-Frontend über die Adresse <http://companion:8000> öffnen, um Plug-ins zu verwalten und Tasten zu belegen. Da Companion mit den Rechten des Benutzers `pi` läuft, müssen Sie `pi` noch Zugriff auf das Stream Deck gewähren, indem Sie die Datei `50-companion.rules` von `ct.de/w1w9` in das Verzeichnis `/etc/udev/rules/` kopieren und die Regeln anschließend neu laden:

```
sudo udevadm control --reload-rules
```

Sollten Sie das Stream Deck bereits zuvor angeschlossen haben, müssen Sie es einmal ab- und wieder anstecken, damit die Rechte entsprechend gesetzt werden und Companion es finden kann, wenn Sie im Web-Frontend im Reiter „Surfaces“ auf „Rescan USB“ klicken. Dort finden Sie alle gefundenen Stream Decks aufgelistet.

Companion verwendet stets die gleiche Tastenkonfiguration für alle angeschlossenen Stream Decks, unabhängig von ihrer Größe. Die Basis ist dabei stets das größte Modell mit vier Reihen à acht Tasten, die auch der Emulator anzeigt und die von links oben nach rechts unten fortlaufend durchnummeriert sind – 1 bis 8, 9 bis 16, 17 bis 24 und 25 bis 32. Auf kleineren Modellen wird nur ein Ausschnitt des Tastenfelds dargestellt – das Stream Deck mit 3 Reihen à 5 Tasten benutzt die Tasten 1 bis 5, 9 bis

13 und 17 bis 21. Beim Stream Deck Mini stehen Ihnen nur die Tasten 2 bis 5 und 10 bis 12 zur Verfügung.

Indem auf den kleineren Stream Decks nur ein Teil des Layouts abgebildet wird, haben Sie die Möglichkeit, allein durch die Wahl des Konsolenmodells den Anwendern einen unterschiedlichen Funktionsumfang zur Verfügung zu stellen. Für Interview-Gäste zum Beispiel ist ein Stream Deck Mini, mit dem jeder sein Mikrofon selbst stumm schalten kann, völlig ausreichend. Diese Möglichkeit hat der Moderator beim größeren Stream Deck dann auch, mit den zusätzlichen Tasten können Sie es ihm aber auch erlauben, Clips selbst einzuspielen – für die Gäste sind diese Tasten unerreichbar. Der Videoproducer hingegen hat über den Emulator Zugriff auf gleich 32 Tasten und kann darüber auch den Stream oder den Chat steuern, was durch geschickte Platzierung außerhalb des Zugriffsbereichs des Moderators liegt.

Stecksystem

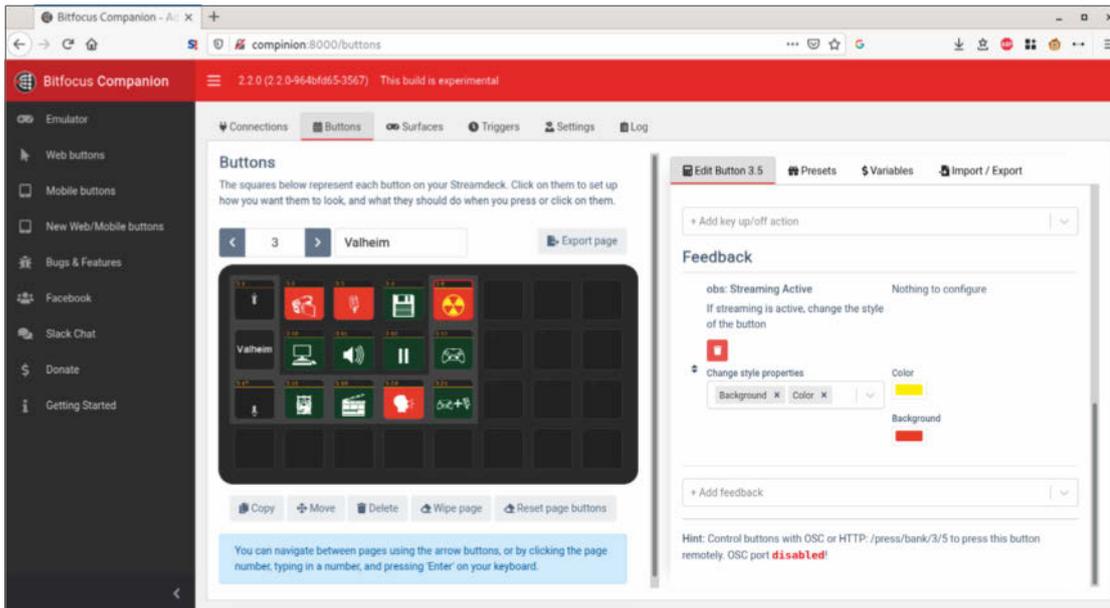
Damit Sie mit Companion zum Beispiel den Stream in OBS starten und stoppen, beim Bildmischer zwischen verschiedenen Kameraeinstellungen wechseln oder einen Mikrofonkanal auf einem Mischpult stumm schalten können, müssen Sie zunächst das für die jeweilige Funktion zuständige Plug-in über den Reiter „Connections“ hinzufügen. Für OBS ist das gleichnamige Plug-in zuständig, für das c't-Videostudio verwenden wir stattdessen die Plug-ins „Blackmagic Design ATEM“ für den Bildmischer und „Behringer/Midas X32/M32“ für das Audio-Mischpult. Vergessen Sie nicht, die Plug-ins durch einen Klick

Es gibt 10 Arten von Menschen. iX-Leser und die anderen.

Jetzt Mini-Abo testen:
3 Hefte + Tastatur nur 19,35 €

www.iX.de/testen





Dank einer Feedback-Funktion erkennen Sie an der Hintergrundfarbe der zugehörigen Taste sofort, ob der Stream live ist.

auf „Edit“ zu konfigurieren und etwa die IP-Adresse Ihres Streamingrechners für OBS oder die des Bildmischers anzugeben.

Im Reiter „Buttons“ können Sie anschließend die Tasten einzeln konfigurieren. Basis ist stets das Stream Deck XL mit seinen 32 Tasten; welche Ausschnitte das kleinere Stream Deck und das Stream Deck Mini anzeigen, ist durch die Gehäusefarbe des virtuellen Stream Deck XL angedeutet. Grundsätzlich gilt: Was Sie im Web-Frontend sehen, wird auch genau so auf den Tasten der Stream Decks angezeigt.

Für mehr Flexibilität können Sie in Companion insgesamt 99 verschiedene Layouts erstellen und zwischen ihnen hin und her blättern. Die Spezialfunktionen Seitennummer sowie „Page up“ und „Page down“ befinden sich standardmäßig in der linken Spalte und belegen die Tasten 1, 9 und 17. Alle anderen Tasten sind am Anfang nicht belegt, weshalb sie komplett schwarz sind.

Um eine Taste mit einer Funktion zu belegen, klicken Sie sie an und wählen zunächst aus, ob es sich um eine reguläre Taste handelt oder Sie eine der drei Spezialfunktionen zum Blättern verwenden wollen. Reguläre Tasten erhalten oben einen Tastenkopf mit ihrer eindeutigen Nummer, die aus der Seitennummer und Position besteht. Los geht es mit Taste 1.1 links oben auf der ersten Seite bis hin zu Taste 99.32 rechts unten auf der letzten Seite. Ist eine Taste gedrückt, so wird der Tastenkopf als Rück-

meldung gelb unterlegt. Unterhalb des Kopfes können Sie Text eingeben oder eine 72 x 56 Pixel große Grafik hochladen, außerdem die Hintergrund- und Textfarbe auswählen. Bevor Sie anfangen zu zeichnen: Der Companion-Font unterstützt viele UTF-8-Zeichen und -Symbole, etwa Kameras, Mikrofone und vieles mehr, die Sie stattdessen ausprobieren können.

Rückkanal

Jede Taste kennt drei Funktionsgruppen: Was beim Drücken geschehen soll, was beim Loslassen, und ob ihre Darstellung von einer Funktion (Feedback) bestimmt wird. In jeder dieser Gruppen können Sie einen oder mehrere Befehle einfügen, zum Beispiel für eine Stream-Taste unter „Press Action“ die Funktion „obs: Start Streaming“ und als „Release Action“ „obs: Stop Streaming“. Außerdem sollten Sie „Latch/ Toggle“ aktivieren, damit der Stream bei der ersten Betätigung startet – womit der Tastenkopf dann gelb unterlegt wird – und beim nächsten Mal stoppt. Starten Sie den Stream ausnahmsweise manuell in OBS, so bekommt Companion das nicht mit – schließlich haben Sie die Taste nicht gedrückt – und reagiert deshalb bei der ersten Betätigung erneut mit Stream starten anstatt wie gewünscht mit Stream beenden.

Um solche Probleme zu vermeiden, sollten Sie den Status wann immer das möglich ist durch Feedback-Funktionen ermitteln lassen. Die sinnvolle Kon-

figuration der Stream-Taste ist deshalb, „Latch/ Toggle“ zu deaktivieren und als „Press Action“ die Funktion „obs: Toggle Streaming“ zu wählen. Damit die Taste rot hinterlegt wird, sobald der Stream läuft, fügen Sie unter „Feedback“ die Funktion „obs: Streaming Active“ hinzu und wählen Rot als Hintergrundfarbe aus. Damit wird die Taste sofort rot, sobald der Stream startet, gleichgültig, ob Sie dies per Tastendruck oder manuell in OBS veranlasst haben.

Sie können auch gleich mehrere Feedbacks pro Taste einrichten, die dann nacheinander ausgeführt werden. Sie könnten etwa bei einem Kameraeingang des Atem-Bildmischers zunächst das Feedback „atem: ME: One ME preview source“ hinzufügen und Grün als Hintergrundfarbe wählen. Dann würde die Taste grün, sobald die Kamera in der Vorschau auf den nächsten Übergang aktiv ist. Indem Sie als zweites Feedback „atem: ME: One ME program source“ hinzufügen und Rot als Hintergrundfarbe, wird die Taste rot, sobald die Kamera auf Sendung geht – selbst dann, wenn die Kamera sowohl in der Preview als auch in der aktuellen Übertragung aktiv ist. Das zweite Feedback überschreibt die Hintergrundfarbe des ersten.

Chat im Griff

Für Twitch-Streamer ist das gleichnamige Companion-Plug-in besonders interessant: Aktuell häufen sich die Vorfälle, bei denen größere Gruppen Störer in einen Stream hineinplatzen und im Chat Hass und Hetze verbreiten. Bis Moderatoren oder der Streamer sie einzeln hinausbefördern können, dauert es eine Weile. Die Erste-Hilfe-Maßnahme ist deshalb, den Chat in den „Subscribers only“-Modus zu versetzen, in dem nur noch zahlende Abonnenten chatten dürfen. Dazu können Sie eine Taste mit der Funktion „twitch: Toggle sub only mode“ versehen, das zugehörige Feedback heißt „twitch: Chat Status“ und der Modus „Sub“. Wer kein Twitch-Affiliate oder gar -Partner ist, kann alternativ den „Slow Mode“, den „Emoji Mode“ oder den „Followers only Mode“ aktivieren.

Damit das Twitch-Plug-in die nötigen Zugriffsrechte auf Ihren Kanal bekommt, müssen Sie sich zunächst einmal auf twitch.tv anmelden. Dann wechseln Sie zu den Einstellungen des Twitch-Plug-ins im Web-Frontend von Companion und klicken auf den Link „Local Token“. Der Link führt Sie zur Website des Plug-in-Entwicklers, dort wählen Sie aus, welche Funktionen das Plug-in künftig steuern darf – standardmäßig sind alle Funktionen aktiviert. Mit einem

Klick auf „Connect with Twitch“ landen Sie bei Twitch, wo Sie noch einmal bestätigen müssen, dass das Plug-in die ausgewählten Funktionen fernbedienen darf. Sie werden schließlich wieder auf die Website des Entwicklers weitergeleitet, die Ihnen das von Twitch vergebene Token anzeigt. Dieses tragen Sie in der Konfiguration des Twitch-Plug-ins unter „Access Token“ ein. Derzeit sind diese Token bis zu 60 Tage gültig, Sie müssen also etwa alle zwei Monate ein neues abrufen, damit Sie weiterhin den Twitch-Chat auf Knopfdruck zügeln können.

YouTube angeströmt

Auch das YouTube-Plug-in benötigt ein Token, um auf Ihren Kanal zugreifen zu dürfen. Eine Web-App der Entwickler, die Ihnen das Token liefert, gibt es aber nicht. Stattdessen müssen Sie auf <https://console.developers.google.com/> ein externes Projekt anlegen und dafür das „YouTube Data API v3“ einbinden. Die Berechtigungen lassen Sie von <https://www.googleapis.com/auth/youtube.force-ssl> übernehmen.

Beim Einrichten der OAuth-Authentifizierung Ihrer neuen Webanwendung müssen Sie <http://localhost:3000> als Weiterleitungs-URI angeben. In den Plug-in-Einstellungen selbst geben Sie lediglich die Client-ID und den Schlüssel an. Sobald Sie auf „Save“ klicken, öffnet sich ein neues Browser-Fenster, das die Zugangsdaten Ihres Google-Accounts und den zu benutzenden YouTube-Kanal abfragt. Anschließend leitet Sie Google zum Plug-in weiter, wobei das Token automatisch eingetragen wird.

Die Einbindung des Stream-Status funktioniert genauso wie weiter oben für OBS beschrieben; die Feedback-Funktion für YouTube lautet „yt: Broadcast status“. So konfiguriert wird die Taste nur dann rot, wenn Sie live sind – selbst dann, wenn Sie den Stream klassisch per Maus im Browser beendet haben.

Fazit

Mit einem Stream Deck und Bitfocus Companion haben Sie es selbst in der Hand, Ihren Stream ohne große Ablenkung zu steuern: Dank der Haptik der Tastenkonsolle können Sie die Funktionen notfalls sogar blind auslösen, in jedem Fall aber schneller als mit einer Maus in einer grafischen Oberfläche. Wegen des Raspberry Pi mit WLAN-Anbindung ist das Gerät klein und portabel genug, um auch vor der Kamera stehen zu können. (mid) **ct**

**Konfigurationsdateien
und Konsolenbefehle**

ct.de/w1w9

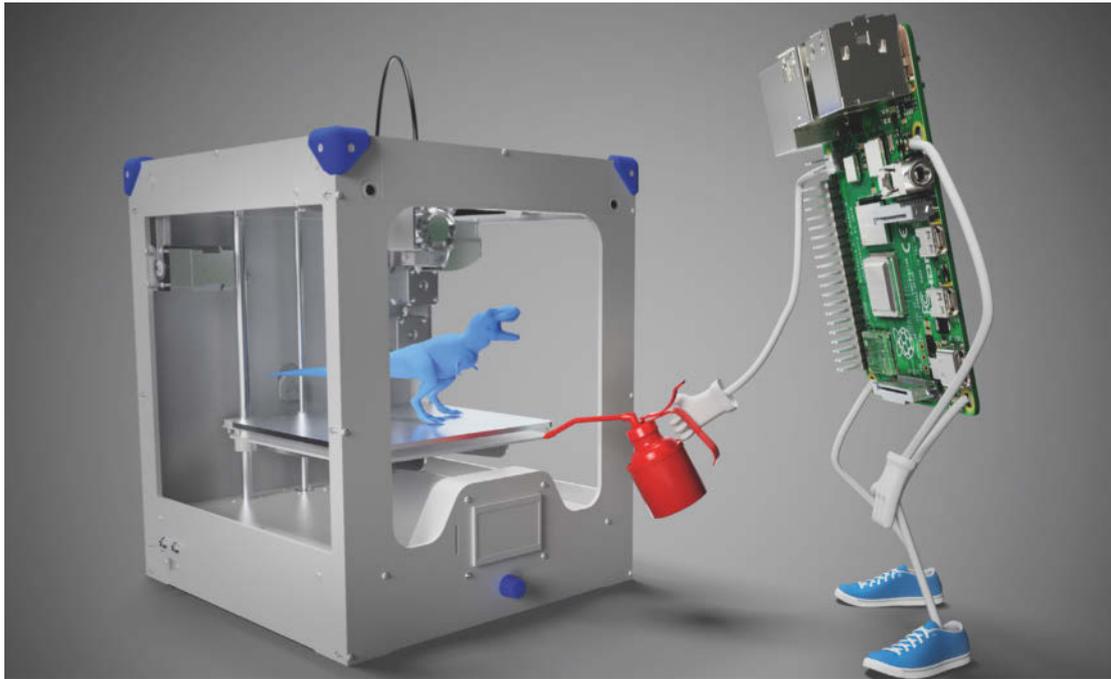


Bild: Andreas Martini

Octoprint bringt 3D-Drucker ins Netz

3D-Drucker machen über viele Stunden Lärm, weshalb man sie meist in eine entlegene Ecke der Wohnung verbannt. Damit man da nicht für jede Temperaturanpassung hinlaufen muss, gehört der Drucker ins Heimnetz. Mit Octopi lässt sich das leicht und günstig nachrüsten.

Von **Pina Merkert**

Beim 3D-Druck gibt es immer wieder Wartezeiten, beispielsweise, um das Druckbett vorzuheizen. Muss man für jede Einstellung zum Drucker laufen, der aus Lärmschutzgründen im Keller oder in einem Abstellraum steht, fallen viele Wege an. Möchte man den Druck zudem im Auge behalten, beispielsweise weil man mit einer neuen Filamentsorte experimentiert und Zweifel hat, ob die

Betthaftung ausreicht, fallen noch viel mehr Wege an und jeder zusätzliche Meter nervt.

Abhilfe schafft da ein Raspberry Pi (gern auch ein älteres Modell) mit Octoprint, einer browserbasierten 3D-Drucker-Oberfläche, die den Drucker von überall im Heimnetz steuerbar macht. Weil die Kombination so naheliegend ist, gibt es sogar eine Rasp-Distribution namens „Octopi“. Drucker-Mainboard

per USB-Kabel mit Raspi verbinden – und los geht's. Eine Steuerung über USB unterstützen nahezu alle jemals verkauften 3D-Drucker.

Octoprint erlaubt, Bett und Düse vorzuheizen, stellt die aktuelle Temperatur als attraktives Diagramm dar, bietet Tasten, um den Drucker manuell zu bewegen und erlaubt G-Code hochzuladen und Drucke zu starten. Wer neu kalibrieren muss, freut sich über eine integrierte Konsole, ein gut bestücktes Plug-in-System rüstet diverse Spezialfunktionen nach. Besonders attraktiv ist das Plug-in „Octolapse“, das mit einer Raspi-Kamera schicke Zeitraffer-Videos vom Verlauf der Drucke aufzeichnet. Mit diesem Artikel gelingt Ihnen die Einrichtung in wenigen kurzen Minuten.

Setup Wizard

Start
Sicherung wiederherstellen?
Zugangsbeschränkung
Anonymous Usage Tracking
Onlineprüfung
Plugin Blackliste
Default Druckerprofil
Beenden

Konfiguriere dein Druckerprofil

Bitte stelle sicher, dass die folgenden Einstellungen deinem Drucker entsprechen. Falls du planst, mehr als einen Drucker mit OctoPrint zu verbinden, kannst du weitere Druckerprofile unter Einstellungen > Druckerprofile konfigurieren

Allgemein **Druckbett & -volumen** Achsen Hotend & Extruder

Formfaktor Rechteckig
 Rund

Ursprung Unten links

Heizbett

Beheizte Kammer

Bitte definiere das Druckvolumen.

Breite (X) 250 mm
Tiefe (Y) 210 mm
Höhe (Z) 210 mm

Wenn der Druckkopf des Druckers sich außerhalb des Druckvolumens bewegen darf (z.B. für Nozzlereinigungs-routinen) kannst du hier eine als gefahrlos geltende Bounding Box für seine Bewegung definieren.

Individuelle Bounding Box

Diese Informationen werden für das Temperaturtab, den Bounding-Box-Check, den GCODE Viewer und/oder beim Slicen aus OctoPrint heraus verwendet.

Zurück Falls nicht anders vermerkt, kannst du jede Wizardseite durch Klicken auf "Weiter" oder "Beenden" überspringen. Weiter

Ein Einrichtungsassistent hilft beim ersten Start, die wichtigsten Einstellungen vorzunehmen. Das „Druckerprofil“ enthält beispielsweise Angaben zur Größe des Druckbetts.

Installation

Octopi laden Sie als gezippte IMG-Datei von octoprint.org herunter (siehe ct.de/wftz). Auf der Webseite finden Sie auch eine Anleitung, um Octoprint auf anderer Hardware wie einem Heimserver zu installieren. Für den Raspi ist die Installation aber am einfachsten: Archiv auspacken, Raspi-Imager öffnen, „Betriebssystem/eigenes Image“ auswählen, IMG-Datei im Downloads-Ordner auswählen und die SD-Karte beschreiben. Octopi basiert auf dem offiziellen Raspberry Pi OS, weshalb die mit dem Raspi-Imager einstellbaren Voreinstellungen funktionieren. Das Menü mit den Voreinstellungen öffnet sich mit der Tastenkombination Strg + Umschalt + X (wie das funktioniert haben wir auf Seite 26 beschrieben).

Statt des Raspi-Imagers können Sie auch Balena Etcher oder den „Schreiber von Laufwerksabbildern“ beziehungsweise das „Festplattendienstprogramm“ benutzen. Dann müssen Sie die Einstellungen fürs WLAN aber nach dem ersten Start mit angeschlossenen Display und Tastatur nachholen oder in die Datei octopi-wpa-suppllicant.txt im Verzeichnis /boot eintragen.

Die SD-Karte stecken Sie anschließend in den Raspi und verbinden auch gleich den Drucker mit dem Minirechner. Meist ist beim Drucker beziehungsweise dessen Mainboard ein Kabel mit USB-A und USB-B dabei. Manche Drucker können Sie auch mit einem alten Handy-Ladekabel (USB-A- und Micro-USB-Stecker) mit dem Raspi verbinden.

Drucker verbinden

Nach dem ersten Start sollte sich der Raspi in Ihrem Heimnetz melden und vom Router eine IP bekommen. Diese können Sie im Webinterface vom Router nachsehen. Statt der IP funktioniert meist auch der Hostname, den Sie in den Raspi-Voreinstellungen vergeben haben. Wir haben für den Artikel einen Prusa i3 netzwerkfähig gemacht, weshalb wir als Hostname „prusai3“ eingestellt haben. Unser Router ließ sich für die Namensauflösung einspannen, weshalb der Raspi nach wenigen Sekunden unter <http://prusai3> erreichbar war. Klappt die Namensauflösung nicht, nutzen Sie die IP-Adresse aus der Heimnetz-Übersicht im Router-Interface, beispielsweise <http://192.168.178.68>.

Beim ersten Aufruf begrüßt Sie ein Assistent, der diverse Fragen zum Druckernamen et cetera stellt. Die Erklärtexte sind automatisch auf Deutsch, wenn der Browser in deutscher Sprache installiert ist. Da

Verbindung

Serialport
/dev/ttyACM0

Baudrate
115200

Druckerprofil
Prusa i3 mk3

Verbindungseinstellungen speichern
 Automatisch bei Serverstart verbinden

Verbinden

Steuern kann Octoprint den 3D-Drucker erst, wenn die Serielle Verbindung über das USB-Kabel klappt. Für Port und Baudrate schlägt Octoprint die üblichen Werte vor und speichert sie auch für später.



Der kleine Raspi ist unter dem 3D-Drucker-Gehäuse versteckt. Seine Kamera hängt als kleines Rechteck an der Vorderkante der Seitenscheibe.

Gina Häußge, die Hauptentwicklerin von Octoprint, Deutsche ist, gibt es keine Übersetzungsfehler und die Beschreibungen sind gut verständlich.

Im Einrichtungsassistent legen Sie auch ein Profil für Ihren Drucker an, in dem Sie den Namen und die Größe des Druckraums festlegen. Octoprint nutzt diese Angaben beispielsweise für eine Prüfung, ob geladener G-Code überhaupt in den Druckraum passt. Alle Angaben können sie später selbstverständlich in den Einstellungen anpassen. Die Verbindung zum Drucker nehmen Sie erst auf, nachdem Sie den Assistenten fertig durchgeklickt haben.

Dann nämlich erscheint in der linken oberen Ecke der Weboberfläche ein Kasten mit Verbindungseinstellungen. Die bestehen aus Drop-down-Menüs für den Port, die Baudrate und das im Assistenten erstellte Druckprofil. Der Name des Ports ist der Pfad zum seriellen Interface in üblicher Linux-Benennung. Zur Auswahl steht üblicherweise `/dev/ttyUSB0` oder `/dev/ttyACM0` wie bei unserem Prusa. Stecken noch mehr serielle USB-Geräte am Raspi (der hat ja neben Octoprint noch Kapazitäten für andere Aufgaben frei) wechselt die Nummerierung der Devices je nach-

dem, welches vom Kernel beim Boot zuerst erkannt wird. Sollte Sie das stören, können Sie wie in [1] beschrieben eine UDEV-Regel für den Drucker erstellen. Eigene Namen für Devices müssen Sie in den Einstellungen im Bereich „Drucker“ auf der Seite „Serielle Verbindung“ im Textfeld „Zusätzliche serielle Ports“ hinzufügen.

Die Baudrate, mit der Ihr Drucker kommunizieren möchte, sollte in der Beschreibung zum Gerät benannt werden. Falls Sie dort keine Angabe finden, probieren Sie einfach mal 115200 aus. Das ist die Voreinstellung in der Konfiguration der Marlin-Firmware und auf der basieren die meisten Firmwares günstiger Drucker.

Das Druckerprofil ist normalerweise schon vorausgewählt, weshalb Sie anschließend nur noch den Haken bei „Verbindungseinstellungen speichern“ setzen wollen und anschließend auf den großen „Verbinden“-Knopf drücken. Nach ein paar Sekunden wechselt dessen Beschriftung auf „Trennen“ und im Temperatur-Reiter füllt sich das Diagramm mit Messwerten der Temperaturfühler von Heizbett und Hotend.

„Automatisch bei Serverstart verbinden“ sollten sie nur auswählen, wenn Sie immer nur mit Octoprint drucken und nie von SD-Karte und außerdem Drucker und Raspi stets gemeinsam einschalten. Dann spart das Häkchen dort einen Knopfdruck. Für mehr Übersicht in der linken Leiste können Sie den Kasten mit den Verbindungseinstellungen mit einem Klick auf die Überschrift einklappen.

Das Webinterface

Im Kasten „Status“ in der linken Leiste finden Sie Infos zum Fortschritt des aktuellen Drucks und Schätzungen zur Dauer. Darunter können Sie G-Code-Dateien hochladen, die Octoprint auf der SD-Karte des Raspi zwischenspeichert. Steckt eine SD-Karte im Drucker, erscheinen auch die G-Code-Dateien auf dieser Karte in der Liste. Hinter dem Schraubenschlüssel-Symbol finden Sie eine Einstellung, um nur eine der beiden G-Code-Quellen anzuzeigen.

Rechts von der Seitenleiste finden Sie einen Bereich mit mehreren Reitern. Die Standardansicht ist „Temperatur“ mit einem Diagramm mit den Thermistor-Messwerten der letzten Minuten. An dem können Sie leicht erkennen, ob Ihr Drucker ein PID-Tuning vertragen könnte. Dann schwingt die Temperaturkurve im Diagramm in Form einer Wellenlinie über, was bedeutet, dass die Parameter der Temperaturregelung nicht optimal eingestellt sind. Unter dem Diagramm können Sie Bett und Düse vorheizen. Wer Energie sparen möchte, legt eine Styrodur-Platte aufs Heizbett, was Staub fernhält und das Aufheizen beschleunigt. Die Platte müssen Sie aber wegnehmen, bevor Sie den Druck starten, weshalb ein manuelles Vorheizen hier zwingend zum Workflow gehört.

Den Reiter „Steuerung“ brauchen sie immer dann, wenn der Drucker manuelle Eingriffe benötigt. Dort können Sie die den Nullpunkt anfahren und die Achsen in großen oder kleinen Schritten bewegen, ohne die passenden G-Codes auswendig kennen zu müssen. Falls eine Kamera angeschlossen ist, sehen

Octoprint zeigt im Browserfenster links Statistiken zum laufenden Druck und darunter die bereit gelegten G-Code-Dateien. Rechts davon zeigen mehrere Reiter die Temperaturen von Düse und Bett, erlauben manuelle Steuerbefehle und zeigen Bilder von einer Webcam, falls angeschlossen.

The screenshot shows the Octoprint web interface. On the left, the 'Status' section displays: Status: Bereit, Resendverhältnis: 0 / 9 (0%), Datei: Hochgeladen: Zeitraffer: -, Ungefähre Dauer: -, Dauer: -, Verbleibend: -, Gedruckt: -. Below this are buttons for 'Drucken', 'Pause', and 'Abbruch'. The 'Dateien' section has a search bar and a 'Neues Verzeichnis...' button. At the bottom, it shows 'Frei: 26.0GB / Gesamt: 29.0GB' and 'Upload' buttons for 'Upload' and 'Upload (SD)'. On the right, the 'Temperatur' tab is active, showing a temperature graph with a green frog watermark. The graph plots 'Ist T: 166.1°C' and 'Soll T: 160.0°C' for the tool, and 'Ist Bett: 42.0°C' and 'Soll Bett: 60.0°C' for the bed. Below the graph is a table for temperature control:

| | Ist | Soll | Offset |
|------|---------|--------|--------|
| Tool | 164.3°C | 180 °C | 0 °C |
| Bett | 42.0°C | 60 °C | 0 °C |

sie dort auch aus der Ferne, was der Drucker tut. Wer stattdessen lieber G-Code tippt, nutzt den Reiter „Terminal“, das einen direkten Zugriff auf die serielle Verbindung zum Drucker erlaubt.

Zur Kontrolle während des Drucks dient der Reiter „GCode Viewer“. Darin zeigt Octoprint grafisch an, wie die per G-Code versendeten Bewegungsbahnen aussehen. Zuerst erscheinen sie im Browserfenster, gleich darauf sollte der Drucker die Bewegung in der echten Welt genau nachmachen. Da sich hier immer etwas bewegt, ist das der Reiter mit dem größten Unterhaltungswert.

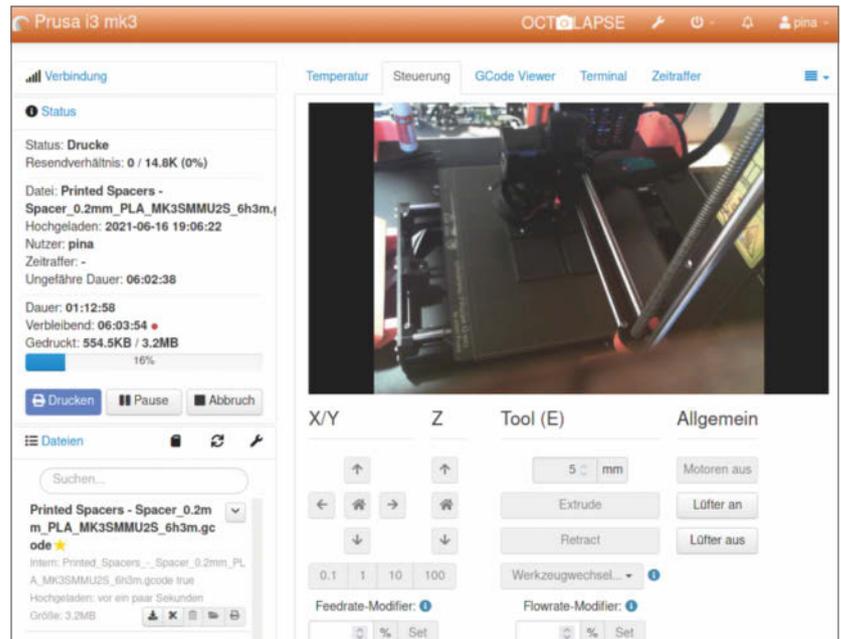
Den Reiter „Zeitraffer“ brauchen Sie nur, wenn eine Kamera angeschlossen ist. Octoprint kann von Haus aus Zeitrafferaufnahmen von den Drucken machen. Wie gut die mit den Bordmitteln von Octoprint aussehen, hängt mit dem Konstruktionsprinzip Ihres Druckers zusammen. Drucker nach dem Ultimaker-Prinzip [2] oder Core-XY fahren die Plattform nach unten weg und bewegen sie nicht in X- oder Y-Richtung. Bei solchen Druckern entstehen direkt hübsche Zeitraffer-Videos. RepRap-artige Maschinen wie der i3 oder Ender 3 bewegen aber das Bett in Y-Richtung, weshalb Zeitraffer-Videos zitterig werden. Für schöne Zeitraffer-Aufnahmen brauchen Sie bei solchen Maschinen ein Plug-in, das Fotozeitpunkt und Bettposition synchronisiert.

Octolapse

Octolapse durchforstet den G-Code nach geeigneten Stellen, um ein Einzelbild für eine Zeitrafferaufnahme aufzunehmen. Wenn der Druckkopf und vor allem das Druckbett nämlich bei jeder Zeitrafferaufnahme an den gleichen X- und Y-Koordinaten stünden, sähe es im Video so aus, als würde der Druckkopf ohne zittern nach oben wegfahren, während magisch unter ihm ein Plastikteil entsteht.

Octolapse sucht dafür nicht nur die ideale Stelle im G-Code, sondern fügt auch bewusst zusätzliche Bewegungen ein, falls der Druckkopf nicht ohnehin an der gewünschten Fotoposition vorbeikommt. Wo Kopf und Bett stehen, wenn Octolapse knipst, lässt sich in einer Registerkarte einstellen, die rechts neben der Zeitraffer-Karte erscheint, sobald man das Plug-in installiert hat.

Erfordert die Fotoposition bei jedem Schichtwechsel zusätzliche Bewegungen, dauert der Druck ein wenig länger und die Schichten haben jeweils eine Sekunde oder zwei mehr Zeit, sich abzukühlen. Für PLA spielt das meist keine große Rolle, bei heißer verarbeiteten Kunststoffen kann die Pause aber die



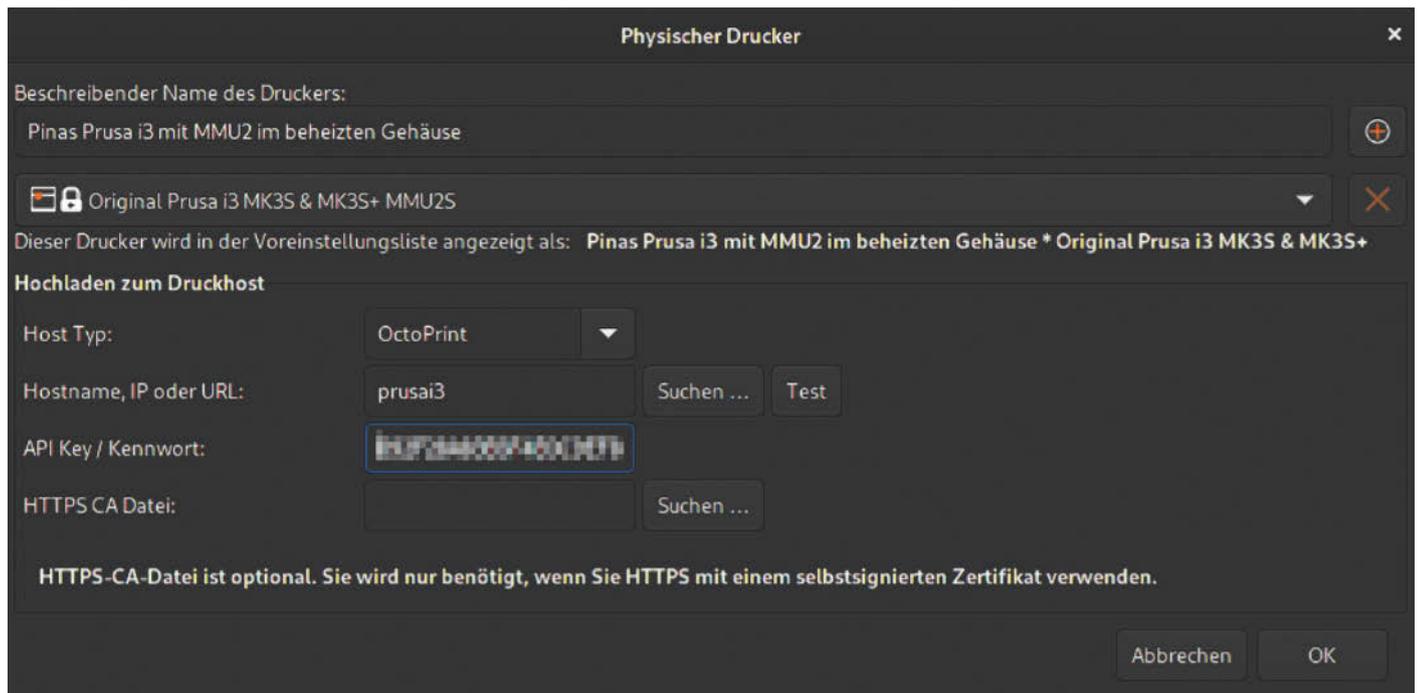
Eine Kamera am Raspi ermöglicht es, den Drucker bei der Arbeit zu überwachen. Neben der Raspi-Kamera funktionieren auch USB-Webcams. Die gleiche Kamera schießt auch die Bilder für Zeitraffer-Videos.

Druckqualität beeinflussen (manchmal auch zum Besseren).

In den Einstellungen können Sie auch die Videoqualität und die Bildfrequenz einstellen. HEVC funktioniert auf dem Raspi leider nicht, da er dafür keinen Hardware-Encoder hat. Die Videodateien sind zwar auch mit H.264 angenehm klein, zeigen aber sichtbare Kompressionsartefakte.

Plug-ins

Neben Octolapse gibt es Dutzende weitere Plug-ins, mit denen Sie Octoprint an Ihre Bedürfnisse anpassen können. Beispielsweise können Sie mit dem Firmware-Updater die Software auf dem Drucker-Mainboard austauschen, ohne das USB-Kabel umstecken zu müssen. Die Plug-ins installieren Sie über den „Pluginmanager“ unter der Überschrift „Octoprint“ in den Einstellungen. Nach der Installation bringen viele Plug-ins eine Konfigurationsseite mit, die in den Einstellungen unter der Überschrift „Plug-ins“ auftaucht.



Prusa Slicer kann die Octoprint-API benutzen, um G-Code direkt in den Druckserver zu laden.

Literatur

[1] Pina Merkert, **Funkbrücke für Thermostate**, Homegear bindet MAX! und HomeMatic-BidCoS zeitgemäß ins Smart Home ein, c't 19/2020, S. 160

[2] Pina Merkert, **Ultimaker gegen seine Klone**, 3D-Drucker nach dem Ultimaker-Prinzip schon ab 350 Euro, c't 1/2019, S. 130

Downloadlinks und Doku

ct.de/wftz

Wir haben für unseren Prusa i3 neben dem „Firmware Updater“ noch „Octolapse“ und den „Print Time Genius“ installiert. Letzterer nutzt statt Octoprints eigener Schätzung die Restzeit-Angaben, die der Prusa Slicer im G-Code hinterlassen kann. Damit das klappt, müssen Sie in den Plugin-Einstellungen den Haken bei „Use Slic3r PE M73 time remaining“ setzen und im Prusa Slicer unter „Druckereinstellungen/Allgemein“ im Kasten „Firmware“ den Haken bei „Unterstützt Restzeit“ setzen.

Prusa Slicer kann auch das Octoprint-API benutzen, um Druckaufträge mit einem Knopfdruck an den Drucker zu versenden. Erzeugen Sie dafür zunächst in den Octoprint-Einstellungen unter der Überschrift „Funktionen“ bei „Application Keys“ einen API-Key für den Prusa Slicer. Kopieren Sie die kryptische Zeichenfolge in die Zwischenablage und wechseln Sie in den Slicer. Gehen Sie dort in die Druckereinstellungen und klicken oben in der Leiste auf das Zahnrad-Symbol. Es öffnet sich der Einstellungsdialog zum „Physischen Drucker“, wo Sie zunächst einen neuen Namen für Ihren real existie-

renden 3D-Drucker vergeben müssen. Als „Host Typ“ wählen Sie dann OctoPrint, als Hostnamen den Namen beziehungsweise die IP-Adresse des Druckers im Heimnetz (Angabe ohne „http://“) und im Feld „API Key“ fügen Sie die Zeichenkette aus der Zwischenablage ein. Die „HTTPS CA Datei“ darf leer bleiben, wenn beide Geräte im gleichen Netz stehen. Danach erscheint nach dem Slicen unten rechts ein Knopf, der den G-Code direkt zum Drucker schickt.

Dank solcher Kleinigkeiten kann Octoprint die Arbeit mit dem 3D-Drucker erleichtern und beschleunigen: Wir konnten dank Octoprint mehr als die Hälfte der Laufwege zum Drucker einsparen. Voraussetzung ist allerdings ein 3D-Drucker, der bereits von der SD-Karte ordentliche Drucke abgeliefert hat. Als Bonus macht Octoprint tolle Zeitrafferaufnahmen, mit denen wir ab sofort den Freundes- und Bekanntenkreis per Videonachricht im Messenger beeindrucken. Beispiele finden Sie über ct.de/wftz. Nur Katzenbesitzer mit Laserpointern kriegen noch mehr Klicks. (pmk) **ct**

GPIO-Pins mit Python programmieren

Den Erfolg im Bastel-Business verdankt der Raspberry Pi seinem GPIO-Anschluss. Damit programmieren Sie leicht eine Mini-Lightshow oder fragen einen Buzzer ab und lassen dem Kandidaten ein Licht aufgehen. Mit der richtigen Python-Bibliothek und unserer Anleitung gelangen auch komplexe Aufgaben und der Raspi steuert bei Ihrem nächsten Live-Auftritt Kamera und Mikrophon.

Von **Mirko Dölle**



| | |
|-------------------------------------|-----|
| GPIO-Pins mit Python programmieren | 84 |
| Servos und Sensoren ansteuern | 92 |
| Automatische Paket-Updates anpassen | 102 |

Ein Raspberry Pi, eine LED, ein Widerstand und wenige Zeilen Python-Code sind alles, was Sie für eine blinkende Lightshow benötigen. Möglich macht das der 40-polige GPIO-Anschluss (General Purpose Input/Output, universelle Ein-/Ausgabe), an dem Sie ohne großen Aufwand LEDs, aber auch Taster, Schalter, Servomotoren, Sensoren und sogar TFT-Displays anschließen können. Die zugehörige Python-Bibliothek GPIO ist auf Raspberry Pi OS bereits vorinstalliert. Es gibt aber auch Bibliotheken für etliche andere Programmiersprachen. Wir haben uns für Python entschieden, weil die Programmiersprache leicht zu erlernen und dank zahlloser Bibliotheken nahezu grenzenlos erweiterbar ist. Damit sind die Möglichkeiten keineswegs auf triviale Dinge wie eine blinkende LED beschränkt. Mit dem nachfolgenden Beispielprogramm steigen Sie gleich ins Show-Business ein und steuern über einen zweckentfremdeten Gitarrenfußschalter für wenige Euro Kamera, Mikrofon, Aufnahme oder gar einen kompletten Live-Stream in der Streaming-Software OBS Studio.

Von den 40 Pins des GPIO-Ports stehen Ihnen 28 für Schalt- und Steueraufgaben zur Verfügung, die übrigen sind Masseanschlüsse oder dienen der Spannungsversorgung. Bei letzteren lauert auch gleich die erste Stolperfalle: An den Pins 2 und 4 liegt die Eingangsspannung (Vin) des Raspi von 5 Volt an. Hierüber können Sie die Eingangsspannung des Raspi für eigene Schaltungen abgreifen oder aber den Raspi mit Strom ohne ein zusätzliches Netzteil versorgen – so arbeitet zum Beispiel das PoE-Hat (Power over Ethernet). Die Schaltpins und Eingänge arbeiten jedoch nur mit der Versorgungsspannung 3,3 Volt (VCC), die an den Pins 1 und 17 zur Verfügung steht – bei mehr gehen sie kaputt! Trotz der niedrigen Spannung dürfen Sie eine LED nicht direkt an den GPIO-Pins betreiben, Sie benötigen zusätzlich einen Vorwiderstand zwischen 220 und 470 Ohm, um den Stromfluss zu begrenzen.

Licht an

Prinzipiell können Sie die LED an jeden der 28 schaltbaren GPIO-Pins anschließen, doch einige besitzen Zusatzfunktionen wie Taktgenerator, Modulator oder sind Teil eines Bussystems wie SPI oder I2C. Im Kasten „Raspberry Pi als Schaltzentrale“ haben wir diese Besonderheiten zusammengefasst und die GPIO-Pins farblich so gekennzeichnet, dass Sie etwaige Sonderfunktionen leicht erkennen können. Um eine Schaltung später nicht umgestalten zu müssen, weil Sie zusätzlich ein bestimmtes Bussystem brauchen,

sollten Sie für einfache Schaltaufgaben wie eine LED einen der grünen GPIO-Schaltpins ohne besondere Zusatzfunktion benutzen. In den nachfolgenden Beispielen ist es Pin 37.

Die nächste anstehende Entscheidung ist, welche Schaltlogik Sie verwenden wollen. Zur Wahl stehen Active High und Active Low. Bei Active High liefert der GPIO-Schaltpin die Versorgungsspannung für die LED. LED und Vorwiderstand werden dann zwischen Pin 37 und Masse geschaltet. Wird der Pin auf High-Pegel geschaltet, also 3,3 Volt, leuchtet die LED. Bei Active Low hingegen werden LED und Vorwiderstand an Pin 1 oder 17 angeschlossen und darüber mit Spannung versorgt – damit die LED leuchtet, müssen Sie den Pin 37 auf Low-Potenzial schalten, also Masse. Beim nachfolgenden Beispiel haben wir uns für Active High entschieden, deshalb schließen Sie die LED nebst Vorwiderstand zwischen Pin 37 und Pin 39 des GPIO-Ports an.

Für die Ansteuerung in Python genügen wenige Kommandos, die Sie direkt in den Python-Interpreter eingeben können. Dazu starten Sie ihn als Benutzer pi mit dem Befehl `python3` und erhalten dann das Python-Prompt, bestehend aus drei Großbuchstaben. Im ersten Schritt binden Sie die GPIO-Bibliothek des Raspi ein:

```
import RPi.GPIO as GPIO
```

Wie im Kasten „Raspberry Pi als Schaltzentrale“ beschrieben sollen die Pins über ihre Pin-Nummer auf der Platine des Raspi referenziert werden, dies legen Sie mit folgendem Befehl fest:

```
GPIO.setmode(GPIO.BOARD)
```

Pin 37 alias GPIO26 ist für die LED-Ansteuerung vorgesehen, es handelt sich also um einen Ausgang. Da wir uns für Active High entschieden haben und die LED im Normalfall nicht leuchten soll, initialisieren Sie den Pin mit Low-Potenzial:

```
GPIO.setup(37, GPIO.OUT,
           initial=GPIO.LOW)
```

Um die LED einzuschalten, müssen Sie nur noch den Pin auf High-Potenzial setzen:

```
GPIO.output(37, GPIO.HIGH)
```

Zum Ausschalten setzen Sie den Ausgang wieder auf Low:

```
GPIO.output(37, GPIO.LOW)
```

Bevor Sie den Python-Interpreter mit Strg+D verlassen, sollten Sie mit `GPIO.cleanup()` noch „aufräumen“ und den GPIO-Anschluss wieder in den Grundzustand zurückversetzen.

Auf und ab

Alle GPIO-Schaltpins lassen sich darüber hinaus auch als digitale Eingänge benutzen. Sie können also feststellen, ob an einem Pin High-Potenzial (3,3 Volt) oder Low-Potenzial (0 Volt, Masse) anliegt. Mit den folgenden Zeilen initialisieren Sie Pin 5 als Eingang:

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(5, GPIO.IN)
```

Anschließend können Sie abrufen, welcher Pegel an dem Pin anliegt:

```
while True:
    if GPIO.input(5):
        print("Pin 5: High")
    else:
        print("Pin 5: Low")
        break
```

Schließen Sie einen Taster an Pin 5 und Pin 6 (Masse) an, so ist Pin 5 im Normalfall High und Low, wenn Sie den Taster gedrückt halten – und die Schleife endet.

Zugegeben, mit diesen Zeilen brechen Sie die zuvor verkündete Regel, für einfache Schaltaufgaben keinen Pin eines Bussystems zu verschwenden. Denn Pin 5 gehört zum zweiten I2C-Bus des Raspi. Er hat aber noch eine weitere Sonderfunktion: Ein dort angeschlossener Taster startet den Raspi neu, wenn er zuvor heruntergefahren wurde. Damit das auch klappt, während der Raspi heruntergefahren ist, wurde an Pin 5 werksseitig ein sogenannter Pull-up-Widerstand angeschlossen und mit VCC verbunden.

Der Pull-up-Widerstand bewirkt, dass Pin 5 im Normalfall stets auf High-Potenzial liegt. Erst wenn Sie den Pin mittels Taster gegen Masse kurzschließen, erkennt der Raspi Low – dieser Eingang arbeitet also mit der Logik Active Low. Dabei verhindert ein Pull-up-Widerstand von 1,8 kOhm bei den Pins 3 und 5, dass während des Kurzschlusses zu viel Strom fließt und der Raspi beschädigt wird. Wenn Sie also

einen Taster zum Starten und Herunterfahren Ihres Raspi haben wollen, ist Pin 5 trotz Sonderfunktion der Richtige. Anstelle von `break` müssten Sie zum Herunterfahren Python nur den Linux-Befehl `shutdown -h now` ausführen lassen.

Bei anderen Pins gibt es keine fest verdrahteten Widerstände, sondern schaltbare: Hier können Sie bei der Initialisierung des Pins festlegen, ob Sie einen Widerstand brauchen und ob er als Pull-up-Widerstand gegen VCC oder als Pull-down-Widerstand gegen Masse geschaltet werden soll. Die folgende Zeile aktiviert Pin 16 als Eingang und schaltet einen Pull-up-Widerstand gegen VCC, ähnlich wie er bei Pin 5 zum Einsatz kommt:

```
GPIO.setup(16, GPIO.IN,
           pull_up_down=GPIO.PUD_UP)
```

Hitzkopf

Den Zustand eines Pins mit einer `while`-Schleife abzufragen hat eine ganz erhebliche Nebenwirkung: Sie bringt den SoC (System on Chip) des Raspi regelrecht zum Glühen und lastet das System vollständig aus – der Raspi ist nur noch damit beschäftigt, den Pin abzufragen. Ein naheliegender, aber sehr verpöner Weg ist es, mittels `time.sleep()` die Schleife zu bremsen. Doch dann reagiert der Raspi nur noch verzögert auf den Tastendruck. Die sinnvolle Lösung ist, dass sich der Raspi selbst ganz nebenbei um die Überwachung kümmert und das Programm bei jeder Zustandsänderung des Pins benachrichtigt. Dazu müssen Sie für den gewünschten Pin ein Event registrieren und eine Funktion bereitstellen, die aufgerufen wird, sobald sich an dem Pin etwas tut. Das folgende Beispiel fährt den Raspi herunter, wenn Sie die an Pin 5 angeschlossene Power-Taste drücken:

```
import RPi.GPIO as GPIO
import time
from subprocess import call

def buttonPress(pin):
    if not GPIO.input(5):
        print("Shutdown")
        call(['shutdown', '-h',
             'now'], shell=False)

GPIO.setmode(GPIO.BOARD)
GPIO.setup(5, GPIO.IN)
GPIO.add_event_detect(
```

Raspberry Pi als Schaltzentrale

Der 40-polige GPIO-Anschluss ist es, was den Raspberry Pi vom herkömmlichen PC unterscheidet und für viele Basteleien besonders interessant macht: Über die Pins lassen sich nicht nur LEDs und andere Komponenten ein- und ausschalten, sie können auch als Eingänge für Taster und Schalter dienen, modulierte Signale erzeugen und über die Bussysteme SPI (Serial Peripheral Interface) und I2C (Inter-Integrated Circuit) Displays ansteuern oder Messwerte komplexer Sensoren auslesen. Dafür besitzen etliche Pins bis zu fünf Funktionen: Pin 3 und 5 zum Beispiel können LEDs oder Optokoppler schalten, als Taster-Eingänge fungieren oder als I2C-Bus Sensoren auslesen.

In der Übersicht der GPIO-Pins haben wir nur die wichtigsten Sonderfunktionen aufgeführt. Diese soll Ihnen helfen, etwa für eine LED den geeigneten GPIO-Pin zu finden: Zwar

könnten Sie sie an Pin 3 oder 5 betreiben, doch damit wäre der I2C-Bus blockiert für den Fall, dass Sie Ihre Schaltung später noch erweitern wollen. Besser geeignet sind dafür die grün markierten Pins, etwa 11, 16 oder 37, die keine häufig benötigte Zusatzfunktion haben.

Manchmal überschneiden sich auch Anforderungen: Pin 5 ist nicht nur die Taktleitung des I2C-Busses. Schließt man diesen Pin gegen Masse kurz, so schaltet er den Raspi wieder ein, nachdem er heruntergefahren wurde. Möchte man den Raspi also mit einem Power-Taster versehen, muss man auf den I2C-Bus verzichten oder getrennte Tasten zum Ein- und Ausschalten verwenden mit dem Risiko, dass bei Fehlbedienung die I2C-Buskommunikation zusammenbricht. Auf den zweiten I2C-Bus auf den Pins 27 und 28 (Gelb) sollte man aber nicht ausweichen, denn diese Pins sind für die Kommunikation mit dem EPROM eines HATs (Hardware Attached on Top, Huckepackplatine) reserviert, um falls notwendig automatisch zusätzliche Konfigurationsparameter oder Treiber zu laden. Manchmal gibt es auch Alternativen: So lassen sich die PWM-Generatoren auf den Pins 12 und 33 auch auf die Pins 32 und 35 verlegen, indem Sie in der Datei config.txt der Boot-Partition des Raspi zusätzliche Parameter angeben.

Eine häufige Stolperfalle ist, dass manche Softwareprojekte die internen GPIO-Port-Nummern des Broadcom-Chips (BCM) und nicht die auf der Platine aufgedruckten Pin-Nummern verwenden. GPIO 17 am Chip etwa ist mit Pin 11 der 40-poligen Steckerleiste verbunden - das führt zu Verwirrungen. Bei eigenen Projekten sollten Sie die Pin-Nummern von der Platine bevorzugen, denn das erleichtert den Nachbau.

Etliche Pins des GPIO-Anschlusses haben wichtige Sonderfunktionen, weshalb man einfache Schaltaufgaben den grün markierten Pins übertragen sollte.

| Pin-Belegung des Raspi-GPIO-Ports | | | |
|--|--------------|---------|---------------------------------------|
| Sonderfunktion | Bezeichnung | Pin-Nr. | Sonderfunktion |
| | +3,3 V (VCC) | 1 | |
| I2C 1, SDA (I ² C 1, Daten) | GPIO2 | 3 | |
| I2C 1, SCL (I ² C 1, Takt) | GPIO3 | 5 | |
| GPCLK 0 (Taktgenerator) | GPIO4 | 7 | |
| Masse (GND) | | 9 | |
| | GPIO17 | 11 | |
| | GPIO27 | 13 | |
| | GPIO22 | 15 | |
| +3,3 V (VCC) | | 17 | |
| SPI 0, MOSI | GPIO10 | 19 | |
| SPI 0, MISO | GPIO9 | 21 | |
| SPI 0, SCLK (Takt) | GPIO11 | 23 | |
| Masse (GND) | | 25 | |
| I2C 0, SDA (I ² C 0, Daten) | GPIO0 | 27 | |
| GPCLK 1 (Taktgenerator) | GPIO5 | 29 | |
| GPCLK 2 (Taktgenerator) | GPIO6 | 31 | |
| PWM 1 (Modulator) | GPIO13 | 33 | |
| SPI 1, MISO | GPIO19 | 35 | |
| | GPIO26 | 37 | |
| Masse (GND) | | 39 | |
| | | 2 | +5 V (VIN) |
| | | 4 | +5 V (VIN) |
| | | 6 | Masse (GND) |
| | GPIO14 | 8 | UART 0, TXD (seriell, senden) |
| | GPIO15 | 10 | UART 0, RXD (seriell, empfangen) |
| | GPIO18 | 12 | PWM 0 (Modulator) |
| | Masse (GND) | 14 | |
| | GPIO23 | 16 | |
| | GPIO24 | 18 | |
| | Masse (GND) | 20 | |
| | GPIO25 | 22 | |
| | GPIO8 | 24 | SPI 0, CE0 (Chip Select 0) |
| | GPIO7 | 26 | SPI 0, CE1 (Chip Select 1) |
| | GPIO1 | 28 | I2C 0, SCL (I ² C 0, Takt) |
| | Masse (GND) | 30 | |
| | GPIO12 | 32 | |
| | Masse (GND) | 34 | |
| | GPIO16 | 36 | SPI 1, CE2 (Chip Select 2) |
| | GPIO20 | 38 | SPI 1, MOSI |
| | GPIO21 | 40 | SPI 1, SCLK (Takt) |

```
5, GPIO.BOTH,
callback=buttonPress
)
while True:
    time.sleep(5)
```

Der Aufruf der Funktion `GPIO.add_event_detect()` registriert durch den Parameter `GPIO.BOTH` die Funktion `buttonPress()` beim Event-Handler für den Fall, dass der Raspi an Pin 5 entweder einen fallenden (`GPIO.FALLING`) oder einen steigenden (`GPIO.RISING`) Span-

nungspegel erkennt. Dabei erhält `buttonPress()` die Nummer des Pins, an dem die Pegeländerung erkannt wurde. Wenn Sie dieses Beispielprogramm unter dem Namen `pishutdown.py` speichern und dann mit `sudo python3 pishutdown.py` aufrufen, können Sie Ihren Raspi herunterfahren und durch erneuten Tastendruck wieder aufwecken.

Abgeprallt

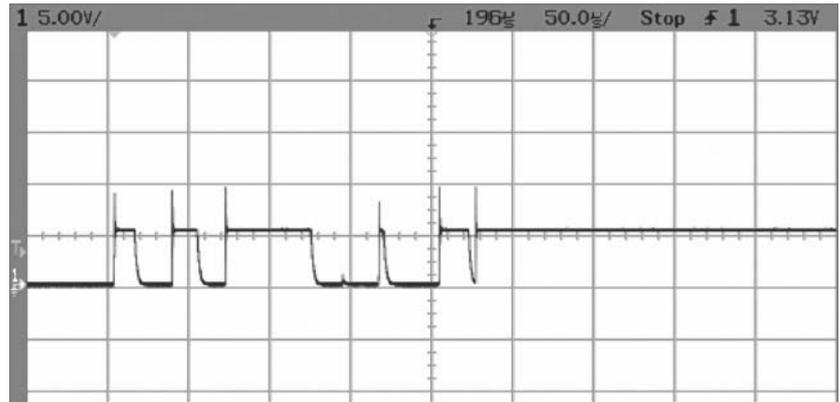
Ein generelles Problem von mechanischen Tastern und Schaltern ist das sogenannte Prellen: Der Schließkontakt überbrückt nicht sofort und dauerhaft die Ruhekontakte. Der bewegliche Schließer unterbricht und schließt die Kontakte mehrfach hintereinander, bis sich am Ende ein stabiler Zustand einstellt. Der Raspberry Pi fragt die GPIO-Pins so schnell ab, dass er diese Zustandswechsel als mehrfache schnelle Betätigung interpretiert. Das Prellen dauert üblicherweise nur wenige Millisekunden und kann schaltungstechnisch zum Beispiel dadurch verhindert werden, dass man einen Kondensator mit 10 bis 100 nF parallel zum Taster oder Schalter anlötet. Für die Funktion `add_event_detect()` gibt es den Parameter `bouncetime`, mit dem Sie angeben können, für wie viele Millisekunden der Raspi solch einen schnellen Zustandswechsel ignorieren soll – im nachfolgenden Beispiel 500 Millisekunden:

```
GPIO.add_event_detect(
    5, GPIO.BOTH,
    callback=buttonPress,
    bouncetime=500
)
```

Da wohl niemand den Power-Taster im Stakkato benutzt, ist eine `bouncetime` von einer halben Sekunde kein Problem. Bei anderen Anwendungen wie Benutzereingaben sollten Sie den Wert aber senken, etwa auf 100 Millisekunden – und dann zusätzlich einen Kondensator anschließen. Bei Tastern sollten Sie außerdem überlegen, ob Sie durch die Angabe von `GPIO.BOTH` wirklich auf beide Signalflanken reagieren wollen oder nicht nur auf eine. Dabei müssen Sie wiederum die Schaltlogik beachten: Bei Active Low bedeutet `GPIO.FALLING`, dass eine Taste gedrückt wird, und `GPIO.RISING`, dass Sie sie loslassen.

Event-Schleifen

Das vorangegangene Beispiel schummelt aber noch immer, schließlich enthält es am Ende eine `while-`



Mechanische Taster und Schalter stellen den Kontakt nicht sofort stabil her. Vielmehr gibt es beim Betätigen eine Reihe von Kontaktunterbrechungen, sogenanntes Prellen, das der Raspi als schnelles Umschalten interpretiert.

Endlosschleife mit einem `sleep()`-Befehl. Die dadurch erzeugte Last ist zwar verschwindend gering, die saubere Lösung ist jedoch, die `while`-Schleife durch folgende Konstruktion zu ersetzen:

```
try:
    loop = asyncio.get_event_loop()
    loop.run_forever()
except:
    loop.close()
```

Am Anfang des Programms, das Sie vollständig auf `ct.de/wfqr` zum Download finden, müssen Sie außerdem noch mit `import asyncio` die Bibliothek für asynchrone Ein-/Ausgabe laden. Diese Bibliothek ist der Schlüssel für eventgesteuerte Programmierung, bei der Funktionen asynchron etwa auf Prozessoren mit mehreren Kernen parallel ausgeführt werden können.

Typischerweise enthalten solche Programme lediglich einige Zeilen zur Initialisierung und unzählige Funktionen, die als Event-Handler dienen. Während der Initialisierung wird der Event-Loop angelegt, etwa mit der Funktion `get_event_loop()` und dann falls nötig der Event-Handler registriert. Anschließend sorgt die Funktion `run_forever()` dafür, dass das Programm weiter läuft, selbst wenn aktuell keine Befehle auszuführen sind. Sollten später Aufgaben hinzukommen, so werden diese zum laufenden Event-Loop hinzugefügt und abgearbeitet, sobald sie an die Reihe kommen.

Fußschalter für OBS Studio

```
import RPi.GPIO as GPIO
import asyncio
import simpleobsws
import logging

Switch1 = 16
Switch2 = 18
BounceTime = 500

ObsIP = '192.168.178.20'
ObsPort = 4444
ObsPass = '****'
SceneName = ['Welcome', 'Gameplay']
AudioSource = 'Mikrofon'

logging.basicConfig(level=logging.INFO)
loop = asyncio.get_event_loop()

async def toggleAudioMute():
    global ObsIP, ObsPort, ObsPass
    global AudioSource
    ws = simpleobsws.obsws(host=ObsIP,
port=ObsPort, password=ObsPass)
    await ws.connect()
    result = await ws.call('GetSourcesList')
    if result.get('status') == 'ok':
        for src in result.get('sources'):
            if src.get('name') == AudioSource:
                break
    if src.get('name') != AudioSource:
        logging.error("No audio source '%s'.",
AudioSource)
    return
    data = {'source':AudioSource}
    result = await ws.call('ToggleMute', data)
    if result.get('status') == 'ok':
        logging.info("Toggled mute on '%s'.",
AudioSource)
    await ws.disconnect()

async def switchToScene(SceneNr):
    global ObsIP, ObsPort, ObsPass
    global SceneName
    ws = simpleobsws.obsws(host=ObsIP,
port=ObsPort, password=ObsPass)
    await ws.connect()

    result = await ws.call('GetCurrentScene')
    if result.get('name') ==
SceneName[SceneNr]:
        logging.info("Scene '%s' already
active.", SceneName[SceneNr])
        return
    data = {'scene-name':SceneName[SceneNr]}
    result = await ws.call('SetCurrentScene',
data)
    if result.get('status') == 'ok':
        logging.info("Switched to scene
'%s'.", SceneName[SceneNr])
        await ws.disconnect()

def buttonPress(pin):
    global loop
    if loop is None:
        return
    state = GPIO.input(pin)
    if pin == Switch1:
        asyncio.run_coroutine_threadsafe(switchToScene(state), loop)
    if pin == Switch2:
        asyncio.run_coroutine_threadsafe(toggleAudioMute(), loop)

try:
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(Switch1, GPIO.IN, pull_up_
down=GPIO.PUD_UP)
    GPIO.setup(Switch2, GPIO.IN, pull_up_
down=GPIO.PUD_UP)
    GPIO.add_event_detect(Switch1, GPIO.BOTH,
callback=buttonPress, bouncetime=Bounc
eTime)
    GPIO.add_event_detect(Switch2, GPIO.BOTH,
callback=buttonPress, bouncetime=Bounc
eTime)
    state = GPIO.input(Switch1)
    loop.run_until_
complete(switchToScene(state))
    loop.run_forever()
except:
    loop.close()
    GPIO.cleanup()
```

Vorhang auf

Viele moderne Python-Bibliotheken arbeiten asynchron, so auch die Bibliothek „simpleobsws“, mit der Sie Befehle an den WebSocket-Server von OBS Studio schicken können. Wenn Sie die Bibliothek mit den Befehlen

```
sudo apt-get install python3-pip
pip3 install simpleobsws
```

installiert haben, kann der Raspi Ihren Auftritt im Internet künftig mit einem Gitarrenfußschalter unterstützen, sodass Sie auf Tritt zwischen zwei Kameraeinstellungen um- oder Ihr Mikrofon stumm-schalten können.

Doppel-Gitarrenfußschalter wie der Lead Foot FS-2 für knapp 15 Euro haben einen 3-poligen 1/4-Zoll-Klinkenstecker, bei denen jeweils ein Schalter mit der Spitze respektive dem ersten Kontaktring verbunden ist, den Masseanschluss nutzen beide gemeinsam. Um den Stecker nicht abzuschneiden zu müssen, haben wir in ein GeekPi Metal Case für ebenfalls knapp 15 Euro eine Stereo-Klinkenbuchse eingebaut. Das GeekPi-Gehäuse ist dafür besonders gut geeignet, da die Seitenwand mit dem MicroSD-Slot ein massiver, vier Millimeter starker Aluminiumblock ist. Hier können Sie gut ein Loch für die Klinkenbuchse BKL 1109029 und ein Feingewinde M12 1 mm bohren oder, falls Sie keinen passenden Gewindebohrer haben, die Buchse mit der beiliegenden Scheibe und Mutter verschrauben. Die Klinkenbuchse von BLK ist zwar mit rund 3 Euro teurer als andere, sie ist aber auch besonders robust und umschließt den Klinkenstecker auf voller Länge. Wenn Sie sie nach dem Einbau mit Schrumpfschlauch ummanteln, gibt es selbst dann keine Kurzschlüsse mit der Raspi-Platine, wenn Sie bei einem besonders beeindruckenden Auftritt versehentlich auf den Stecker oder den Raspi treten und das Gehäuse eindellen.

Für den Anschluss benötigen Sie ein dreipoliges Kabel, ideal sind Servo-Kabel aus dem Modellbau, und drei GPIO-Pins: einmal Masse und zweimal einen der grün markierten Schaltpins. Wir haben uns für die Schaltpins 16 und 18 entschieden; ob Sie als Masseleitung Pin 14 oder 20 benutzen, bleibt Ihnen überlassen – so können Sie leicht durch Umdrehen des Servo-Kabels die Schalter 1 und 2 vertauschen.

Welche Pins Sie als Schalt-Pins benutzen, ist im Listing auf Seite 89 (Download auf ct.de/wfqj) in den

Die gut vier Millimeter dicke, massive Alu-Seitenwand des GeekPi-Gehäuses eignet sich gut, um eine 1/4-Zoll-Klinkenbuchse zum Anschluss eines Gitarrenfußschalters stabil einzubauen.



globalen Variablen `Switch1` und `Switch2` (Zeile 6 und 7) festgelegt. Gleich darunter steht die `BounceTime` zum Entprellen. Die IP-Adresse Ihres Streaming-Rechners und das Passwort für den WebSocket-Server von OBS stehen gleich darunter.

Die Initialisierung der GPIO-Schnittstelle erfolgt am Ende des Listings in den Zeilen 64 bis 70. Der größte Unterschied zu den bisherigen Beispielen ist, dass `GPIO.add_event_detect()` zweimal hintereinander aufgerufen wird, einmal für `Switch1` und einmal für `Switch2`. Als Event-Handler dient in beiden Fällen die Funktion `buttonPress()` ab Zeile 53.

Direkt nach der GPIO-Initialisierung wird die Funktion `switchToScene()` aufgerufen: Damit ist sichergestellt, dass OBS Studio auf jene Kameraszene umschaltet, die vom Fußschalter gerade ausgewählt ist. Da es sich um eine asynchrone Funktion handelt, zu erkennen am Prefix `async def` in Zeile 38, erfolgt der Aufruf in Zeile 72 mittels `run_until_complete()` aus dem Event-Loop heraus. Anschließend startet in Zeile 73 die Endlosschleife.

Asynchronisiert

Die GPIO-Bibliothek des Raspi ist allerdings noch nicht auf asynchrone Arbeitsweise ausgelegt, folglich darf die beim Event-Handler registrierte Funktion `buttonPress()` in Zeile 53 auch nicht als asyn-

chron definiert sein. Die Funktionen `switchToScene()` und `toggleAudioMute()` hingegen nutzen etliche asynchrone Funktionen der WebSocket-Bibliothek von OBS Studio, weshalb sie in den Zeilen 19 und 38 als asynchron definiert sind. Nicht-asynchrone Funktionen können jedoch nicht ohne Weiteres asynchrone aufrufen – die Lösung ist, die Funktion `asyncio.run_coroutine_threadsafe()` zu benutzen, damit sie in der Event-Loop loop ausgeführt, aber nicht mittendrin von einem konkurrierenden Aufruf unterbrochen werden.

Die Arbeitsweise der beiden asynchronen Funktionen `toggleAudioMute()` und `switchToScene()` ist sehr ähnlich: Sie nehmen Kontakt zu OBS Studio auf dem Streaming-Rechner auf, überprüfen, was der aktuelle Zustand ist, senden dann falls nötig den Befehl und schließen die Verbindung wieder.

Der Verbindungsaufbau zum WebSocket-Server erfolgt in den Zeilen 22 und 41, indem eine neue Instanz des WebSocket-Objekts erzeugt wird. IP-Adresse, Port und Passwort sind dabei die notwendigen Parameter. In Zeile 24 ruft `toggleAudioMute()` mittels `ws.call('GetSourcesList')` die Liste der Audioquellen ab, die in der gerade aktiven Szenensammlung von OBS Studio gibt, und prüft für jede, ob es sich um die gesuchte handelt. Falls ja, bricht die Schleife ab – dann enthält die Variable `src` den Namen der gesuchten Audioquelle. Gibt es die gesuchte Audioquelle nicht, so läuft die Schleife in Zeile 26 bis zum letzten Element durch und der Namensvergleich in Zeile 29 schlägt fehl.

Hat `toggleAudioMute()` die richtige Audioquelle gefunden, bittet sie deren Namen in Zeile 32 in ein Dictionary ein und ruft in Zeile 33 den Befehl

`ws.call('ToggleMute', data)` mit dem Dictionary als zweiten Parameter auf. Die Zeile 34 überprüft, ob OBS Studio den Befehl akzeptiert und verarbeitet hat, in Zeile 35 wird dann die Verbindung zum WebSocket-Server geschlossen.

Die Funktion `switchToScene()` ruft in Zeile 44 mittels `ws.call('GetCurrentScene')` ab, welche Szene aktuell in OBS Studio läuft. Da Sie die Szenen bereits per Maus in OBS umgeschaltet haben könnten, ist dieser Schritt erforderlich. Nur dann, wenn die zu aktivierende Szene eine andere ist (Zeile 44), sendet `switchToScene()` wiederum per `ws.call()` in Zeile 48 den Umschaltbefehl an OBS Studio. Die Zeile 49 ermittelt, ob OBS den Befehl akzeptiert hat, anschließend wird die Verbindung getrennt.

Anstatt die Szene zu wechseln oder das Mikrofon stummzuschalten, könnten Sie auf Tritt zum Beispiel auch den Stream starten oder beenden. Welche Befehle der WebSocket-Server von OBS Studio unterstützt, finden Sie in der Protokollspezifikation (Link siehe ct.de/wfqr).

Fazit

Mit der GPIO-Bibliothek und wenigen Zeilen Python-Code wird der Raspberry Pi zur universellen Schaltzentrale. Dank umfangreicher Zusatzbibliotheken sind Sie nicht darauf beschränkt, etwa nur eine LED blinken zu lassen, sondern können auf Tastendruck sogar Schaltaufgaben in Drittprogrammen wie OBS Studio auf anderen Rechnern übernehmen. So wird der Raspi zur universellen Netzwerk-Fernbedienung, die Sie für den perfekten Auftritt individuell anpassen können. (mid) **ct**

Listings und
Dokumentation

ct.de/wfqr

Heft + PDF
mit 29% Rabatt



Hoch hinaus...

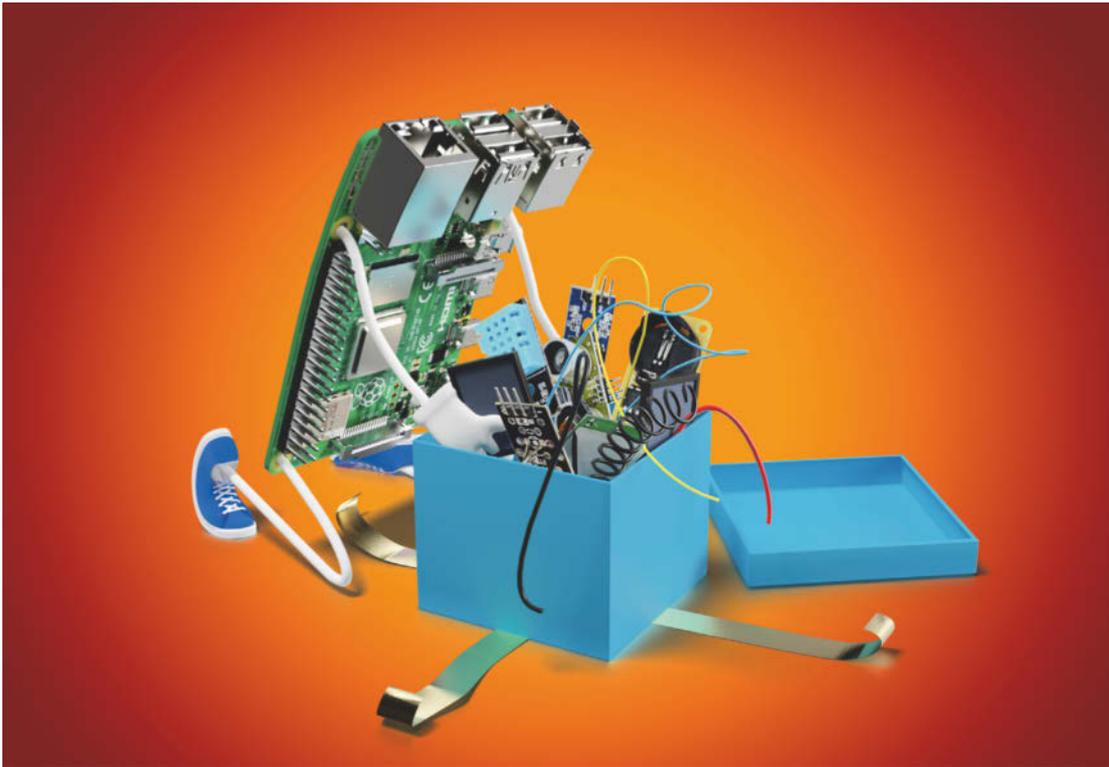
Wie Cloud-Dienste die Arbeit erleichtern

Das c't-Sonderheft zeigt Ihnen Wege, wie Sie sich die Arbeit mit Servern aus der Cloud erleichtern und Neues ausprobieren können.

Heft für 14,90 € • PDF für 12,99 € • Bundle Heft + PDF 19,90 €



shop.heise.de/ct-cloud21



Servos und Sensoren ansteuern

Der GPIO-Anschluss des Raspberry Pi kann mehr, als nur LEDs zum Leuchten bringen. Schon mit wenigen Zeilen Python können Sie Ultraschallsensoren einbinden, Signallaufzeiten messen oder Servos und andere Elektronikkomponenten ansteuern. Denn genau dafür wurde der Mini-Computer gemacht.

Von **Mirko Dölle**

Zum Basteln ist er da: Mit dem Raspberry Pi entwarf die gleichnamige Foundation einen günstigen Mini-Computer als Lernplattform, um mehr Menschen in aller Welt an die Computertechnik heranzuführen. Der ursprünglich 26-polige

GPIO-Anschluss, der später auf 40 Pins erweitert wurde, bietet zahlreiche Anschlussmöglichkeiten für zusätzliche Hardware: LEDs, Taster und Schalter, aber auch komplexe Komponenten wie Sensoren, Displays oder RFID-Leser. Dazu haben viele der insge-

samt 29 „Allzweck“- (General-Purpose-/GP-)Schaltpins Zusatzfunktionen, etwa die Pins 3 und 5, die auch als I2C-Bus (Inter-Integrated Circuit) fungieren, oder die Pins 35, 36, 38 und 40, die als zweiter SPI-Bus (Serial Peripheral Interface) dienen.

Weil Hardware-Bus-Controller Geld respektive Platz auf dem System on Chip (SoC) des Raspi kosten, gibt es davon nur wenige - beim Ur-Raspi mit nur 26 Pins waren jeweils nur ein I2C- und ein SPI-Bus über den GPIO-Anschluss verfügbar. Erst mit der Erweiterung auf 40 Pins kamen ein weiterer I2C- und SPI-Bus hinzu, die jedoch für Aufsteckplatinen alias Hats gedacht sind. Um manche andere Bussysteme wie zum Beispiel CAN-Bus zu nutzen, benötigen Sie zusätzliche Controllerchips oder teure Hats. Für ein-

fachere Bus-Systeme wie den 1-Wire-Bus, der zum Beispiel für Temperatur- oder Feuchtigkeitssensoren benutzt wird, ist das nicht erforderlich. Der Raspi ist leistungsfähig genug, um das jeweilige Busprotokoll in Software nachzubilden. Fachleute nennen das „Bit Banging“.

Bit für Bit

Beim Bit Banging wird ein Schaltausgang in den richtigen Zeitabständen ein- und ausgeschaltet oder an einem Eingang die Zeit gemessen, die zwischen Pegeländerungen liegt. Beim 1-Wire-Bus übernimmt das Modul `w1-gpio` des Linux-Kernels die Aufgabe, standardmäßig den Pin 7 des Raspi mit dem richti-

Der Raspberry Pi verfügt lediglich über zwei I2C- und SPI-Busse sowie zwei PWM-Modulatoren. Andere Bussysteme wie 1-Wire müssen per Bit Banging in Software implementiert werden.

| Sonderfunktion Bezeichnung Pin-Nr. | | | Pin-Nr. Bezeichnung Sonderfunktion | | | |
|--|------------------|----|------------------------------------|--|----|---|
| | +3,3 V (VCC) | 1 | | | 2 | +5 V (VIN) |
| I2C 1, SDA (I ² C 1, Daten) | GPIO2 | 3 | | | 4 | +5 V (VIN) |
| I2C 1, SCL (I ² 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 ² C 0, Daten) | GPIO0 | 27 | | | 28 | GPIO1 I2C 0, SCL (I ² 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) |

gen Timing ein- und auszuschalten, und belastet damit die CPU. Sie müssen allerdings zuvor den 1-Wire-Bus auf der Kommandozeile mit Befehl `raspi-config` oder über die grafische Raspi-Konfiguration aktivieren und den Raspi neu starten.

Gibt es kein fertiges Kernel-Modul für ein bestimmtes Protokoll, so können Sie dieses per Bit Banging selbst implementieren. Für die Programmierung der GPIO-Anschlüsse eignet sich die Programmiersprache Python gut, die in diesem Artikel vorgestellten Beispiele sind allesamt auf Python Version 3 ausgelegt. Mit der unter Raspberry Pi OS vorinstallierten Bibliothek `RPi.GPIO` haben Sie direkten Zugriff auf die einzelnen Pins. In der Python-Einführung (siehe Seite 84) haben wir gezeigt, wie Sie schon mit wenigen Zeilen Python-Code eine LED zum Blinken bringen – das war bereits eine sehr einfache Form des Bit Banging.

Da ein Python-Programm im Userspace ausgeführt wird und nicht wie ein Kernel-Modul im Kernel-Space, sollte das Timing des Protokolls nicht zu anspruchsvoll sein – schließlich kann das Python-Programm jederzeit von privilegierten Prozessen unterbrochen werden. Das kann im Extremfall zu so großen zeitlichen Abweichungen (Jitter) führen, dass die Kommunikation zwischen Raspi und der angesteuerten Komponente zusammenbricht. Bit Banging per Python eignet sich vor allem für niedrige Frequenzen und geringe Datenraten.

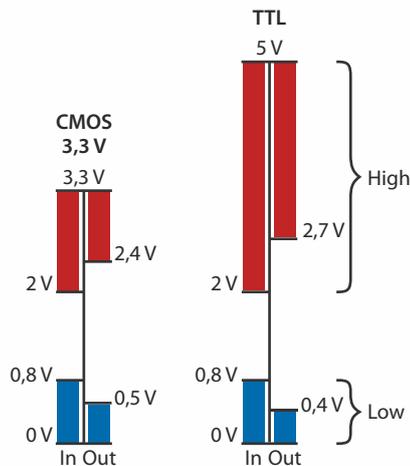
Als Beispiel für ein per Bit Banging in Python implementiertes Protokoll dient der Ultraschall-Entfernungsmesser HC-SR04, den Sie bei verschiedenen Elektronik-Shops für unter 5 Euro bekommen. Das Modul arbeitet autark, Sie müssen sich also nicht darum kümmern, einen Ultraschallton zu erzeugen. Stattdessen müssen Sie gemäß dem Protokoll, das in den Datenblättern aufgeführt ist, über den Trigger-Pin eine Messung anstoßen und erhalten dann am Echo-Pin ein Signal zurück, anhand dessen Sie die Entfernung ablesen können.

Hoch spannend

Dieser Sensor wird unter anderem auch in Arduino-Projekten eingesetzt, was ein wichtiger Hinweis ist: Der Arduino Uno und viele andere Arduino-Boards arbeiten mit TTL-Pegeln (Transistor-Transistor-Logic), die bis zu 5 Volt betragen. Zu viel für den Raspberry Pi, dessen Eingänge nur 3,3 Volt verkraften. Wenn Sie also eine Komponente wie einen Sensor oder ein Display aus einem Arduino-Projekt für den Raspi verwenden, müssen Sie prüfen, ob es davon even-

Logikpegel

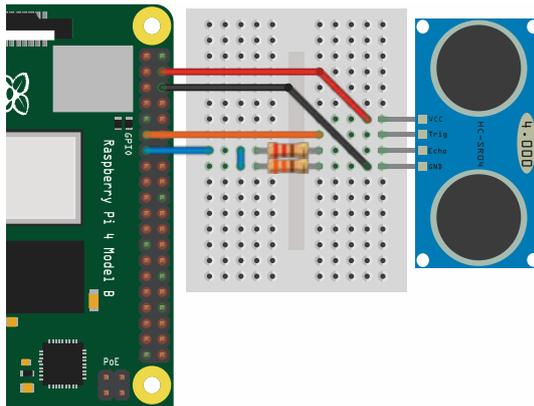
Viele Komponenten und Module etwa aus Arduino-Projekten arbeiten mit TTL-Pegeln. Der Raspberry Pi verträgt allerdings nur CMOS-Logikpegel bis maximal 3,3 Volt, weshalb an Eingängen ein Spannungsteiler erforderlich ist. Die Spannung an den Ausgängen des Raspi hingegen genügt, um auch TTL-Komponenten direkt anzusteuern.



tuell eine Version mit 3,3 Volt gibt oder die Logik-Pegel an den Raspi anpassen. Dafür gibt es Pegelwandler als fertige Baugruppen, doch manchmal geht es auch einfacher.

Beim Ultraschall-Entfernungsmesser HC-SR04 genügen zwei Widerstände für die Anpassung an den Raspi. Dabei können Sie ausnutzen, dass High- und Low-Pegel über eine gewisse Bandbreite erkannt werden. Bei TTL etwa an einem Eingang ein Pegel von 0 bis 0,8 Volt als Low-Signal erkannt, Spannungen von 2 bis 5 Volt als High-Signal. Bei Werten dazwischen ist nicht definiert, ob sie von einer Schaltung als High- oder als Low-Signal erkannt oder idealerweise als Störsignal interpretiert und ignoriert wird, der Störabstand beträgt an TTL-Eingängen

Der Ultraschall-Entfernungsmesser HC-SR04 arbeitet mit TTL-Pegeln, den die Eingänge des Raspberry Pi nicht vertragen. Deshalb müssen Sie die Spannung am Echo-Ausgang mit einem Spannungsteiler aus einem 220- und einem 330-Ohm-Widerstand reduzieren.



also 1,2 Volt. An TTL-Ausgängen ist der Störabstand noch größer, ein Low-Signal hat hier höchstens 0,4 Volt und ein High-Signal mindestens 2,7 Volt. Die 3,3-Volt-CMOS-Logikpegel liegen bei ebenfalls 0 bis 0,8 Volt am Eingang für ein Low-Signal und 2 bis 3,3 Volt für ein High-Signal, die Ausgänge liefern 0 bis 0,5 Volt bei einem Low-Signal und 2,4 bis 3,3 Volt für ein High-Signal.

Damit genügt das 3,3-Volt-Signal am Ausgang eines Raspi auch, um einen Sensor mit TTL-Eingang wie den HC-SR04 anzusteuern. Eine Anpassung für den Trigger-Eingang des Ultraschallsensors ist also unnötig. Am Echo-Ausgang hingegen benötigen Sie einen Spannungsteiler etwa aus einem 220- und einem 330-Ohm-Widerstand, der den TTL-Pegel von 5 Volt auf Raspi-freundliche 3 Volt herunterteilt. Dabei sind die konkreten Widerstandswerte nicht entscheidend, Sie könnten genauso gut einen 2,2-kOhm- und einen 3,3-kOhm- oder einen 5,1-kOhm- und einen 10-kOhm-Widerstand verwenden. Wichtig ist nur, dass die Widerstandswerte im Verhältnis von ungefähr 1 zu 2 stehen, denn dann fallen ein Drittel der Spannung am kleineren und zwei Drittel am größeren Widerstand ab – womit der Signalpegel von 5 Volt auf 3,3 Volt am größeren der beiden Widerstände sinkt. Da der Raspi ein High-Signal auch noch bei 2,5 Volt erkennt, könnten Sie sogar zwei gleiche Widerstände als Spannungsteiler benutzen.

Welche GPIO-Pins des Raspi Sie für Trigger und Echo benutzen, steht Ihnen grundsätzlich frei. Sie sollten für solche einfachen Aufgaben aber keine höherwertigen Anschlüsse von Bussystemen blockieren. Die Pins 11 als Schaltausgang für Trigger und 13 als Eingang für das Echo eignen sich prima

für den Ultraschallsensor. Um flexibel zu bleiben und den Python-Code leicht verständlich zu halten, tragen Sie die Pin-Nummern in Variablen ein und initialisieren damit die Pins:

```
import RPi.GPIO as GPIO
from time import sleep, time

triggerPin = 11
echoPin = 13
GPIO.setmode(GPIO.BOARD)
GPIO.setup(triggerPin, GPIO.OUT,
            initial=GPIO.HIGH)
GPIO.setup(echoPin, GPIO.IN)
```

Sie müssen den Code nicht abtippen, wie immer finden Sie alle Beispiele auf ct.de/wpra zum Download. Mit `initial=GPIO.HIGH` in der vorletzten Zeile legen Sie fest, dass der Trigger-Ausgang bei der Initialisierung auf High-Pegel gelegt wird. Prinzipiell ist es eine gute Idee, Ausgänge bei der Initialisierung auf einen definierten Pegel zu legen, ob High oder Low hängt aber ganz von der Anwendung ab. Der Entfernungsmesser wird getriggert, indem man die Leitung kurzzeitig auf Masse zieht – daher ist es sinnvoll, dass der Pin ansonsten auf High verbleibt und deshalb auch mit High-Pegel initialisiert wird.

Dauerlauf

Um den Raspi als digitalen Zollstock einzusetzen, soll die Messung im Sekunden-Intervall wiederholt werden. Dazu eignet sich eine `while`-Schleife mit Fehlerbehandlung, die Sie mit der Tastenkombination `Strg+C` jederzeit unterbrechen dürfen:

```
try:
    while True:
        print("Entfernung: %.1f cm"
              % Dist())
        sleep(1)
except KeyboardInterrupt:
    GPIO.cleanup()
```

Python strukturiert den Code durch Einrückungen, Standard sind vier Leerzeichen pro Ebene. Damit die Listings nicht ausufernd werden, verwenden wir in den Beispielen nur zwei Leerzeichen pro Ebene, was ebenfalls erlaubt ist. Drücken Sie später im laufenden Programm `Strg+C`, so erkennt die Fehlerbehandlung das Signal `KeyboardInterrupt` und schließt den GPIO-Anschluss des Raspi ordnungsgemäß, bevor das

Programm endet. Für die Entfernungsmessung selbst ist die Funktion `Dist()` zuständig:

```
def Dist():
    GPIO.output(triggerPin, GPIO.LOW)
    sleep(0.000010)
    GPIO.output(triggerPin, GPIO.HIGH)

    while GPIO.input(echoPin) == 0:
        t0 = time()
    while GPIO.input(echoPin) == 1:
        t1 = time()
    return (t1-t0)*34300/2
```

Die ersten drei Zeilen der Funktion ziehen den Trigger-Pin gemäß der Spezifikationen des HC-SR04 für mindestens zehn Mikrosekunden auf Low-Pegel, bevor er für die nächste Messung wieder auf High-Pegel zurückgesetzt wird. Voraussetzung ist, dass der Pin schon zuvor auf High lag – was die Initialisierung sichergestellt hat.

Der Rest der Funktion wertet die Antwort des Ultraschall-Entfernungsmessers auf dem Echo-Pin aus. Dieser hält die Echo-Leitung auf Low-Pegel, bis das Ultraschallsignal ausgesendet wird, und zieht es dann so lange auf High, bis das Echo empfangen wurde. Den Beginn der Messung signalisiert also der Wechsel von Low auf High, das Ende der Messung umgekehrt der Wechsel von High auf Low.

Dementsprechend aktualisiert die erste `while`-Schleife den Start-Zeitpunkt so lange, wie das Echo-Pin noch auf Low liegt. Die Schleife endet genau dann, wenn die Messung beginnt; in der Variablen `t0` steht somit der Zeitpunkt knapp vor Beginn der Entfernungsmessung. Die zweite `while`-Schleife verwendet die umgekehrte Logik, sie aktualisiert die Variable `t1`, solange die Messung läuft. Ist die Messung beendet, endet auch die Schleife und in `t1` steht der letzte Zeitpunkt, an dem die Messung noch lief.

Aus der Differenz von `t1` und `t0` lässt sich nun die Laufzeit des Ultraschallsignals berechnen. Multipliziert man das Ergebnis mit der Schallgeschwindigkeit von 34.300 Zentimetern pro Sekunde, erhält man die gesamte vom Ultraschallsignal zurückgelegte Strecke von Sender zum reflektierenden Objekt und zurück zum Empfänger. Das Objekt befindet somit sich auf halber Distanz.

Maß genommen

Speichern Sie das Listing unter dem Namen `zollstock.py` oder laden Sie es über ct.de/wpra herunter

und starten Sie es mit dem Befehl `python3 zollstock.py`. Wie das vom Raspi erzeugte Trigger-Signal und das Echo des Moduls in Wirklichkeit aussehen, zeigt unsere Oszilloskopaufnahme: Das in Türkis dargestellte Trigger-Signal hat eine Breite von gut 100 Mikrosekunden statt der programmierten 10 Mikrosekunden – dafür ist der Raspi offenkundig nicht schnell genug. Da das Ultraschallmodul lediglich eine Mindest-Signalbreite von 10 Mikrosekunden verlangt, es aber auch mehr sein dürfen, wird die Messung trotzdem angestoßen.

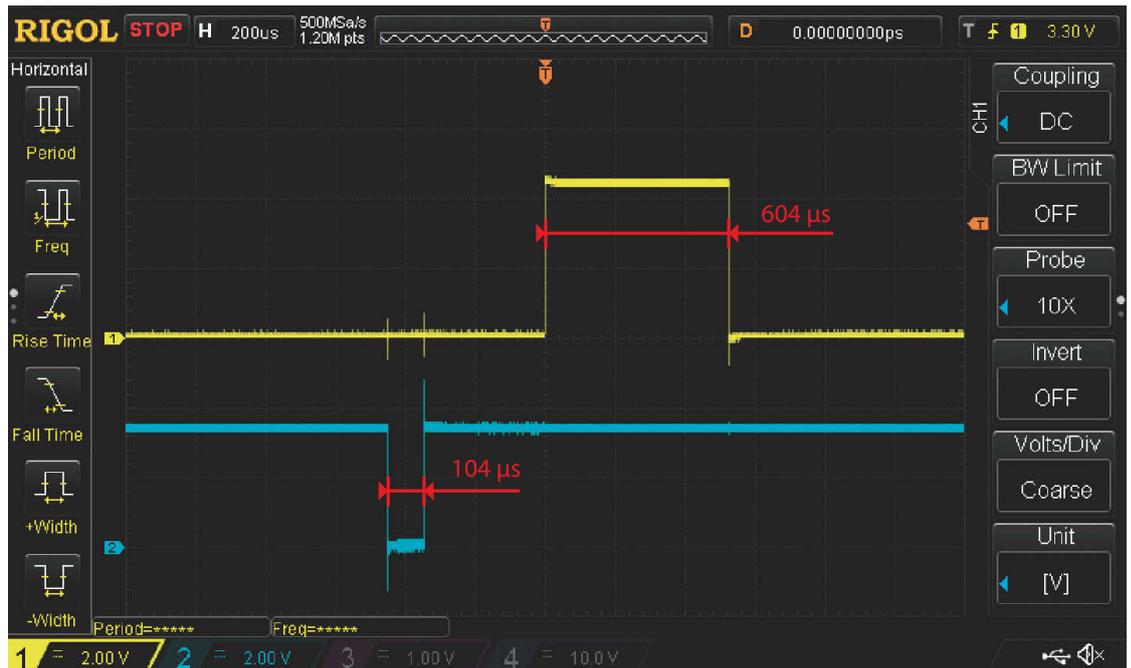
Rund 300 Mikrosekunden nach Ende des Trigger-Signals beginnt das Modul damit, die Ultraschalltöne auszusenden und setzt deshalb den Echo-Pin auf High (gelber Signalverlauf). Die horizontale Auflösung des Oszilloskops beträgt 200 Mikrosekunden pro Kästchen („H 200us“ links oben). Nach 3 Kästchen oder umgerechnet 600 Mikrosekunden fällt der Echo-Pin wieder auf Low zurück – das Echo wurde empfangen, die Messung ist beendet. Daraus ergibt sich eine Entfernung von rund zehn Zentimetern zum gemessenen Objekt.

Wichtig ist, dass die Verkabelung stimmt: Haben Sie das Modul falsch angeschlossen, blockiert das Programm in einer der beiden `while`-Endlosschleifen der Funktion `Dist()` und lastet den Raspi schlimmstenfalls voll aus. Ein entlastendes `sleep()` fehlt mit voller Absicht in beiden `while`-Schleifen; damit die Messung möglichst präzise ist, soll der Raspi so oft wie möglich den Status des Echo-Pins überprüfen und die Zeiten aktualisieren.

Weil die `while`-Schleifen zur Messung der Signalbreite in einem Python-Anwendungsprogramm im sogenannten Userspace laufen, müssen sie sich die Ressourcen des Raspi mit anderen Anwendungen sowie den Aufgaben des ohnehin privilegierten Kernels teilen. Bei einem stark belasteten System kann das zu Abweichungen im Zentimeter-Bereich führen. Präziser messen Sie, indem Sie den Raspi selbst den Echo-Pin überwachen lassen und lediglich bei einer Pegeländerung von Low nach High oder High nach Low `t0` respektive `t1` erfassen. Dazu fügen Sie unmittelbar nach der Initialisierung des Echo-Pins folgende Zeile ein:

```
GPIO.add_event_detect(echoPin,
                      GPIO.BOTH, callback=echoEvent)
```

Damit startet ein sogenannter Event-Handler die Funktion `echoEvent()`, sobald der Raspi eine ansteigende oder fallende Signalflanke (`GPIO.BOTH`) am Echo-Pin erkennt – also sobald das Ultraschallmo-



Für ein Trigger-Signal (Türkis) von nur zehn Mikrosekunden ist der Raspberry Pi nicht schnell genug; 100 Mikrosekunden sind dem Ultraschall-Entfernungsmesser aber auch recht. Die Antwort, ein 600 Mikrosekunden breites Rechtecksignal (Gelb), zeigt ein Objekt in zehn Zentimetern Entfernung an.

dul den Beginn oder das Ende einer Messung signalisiert:

```
t0 = None
t1 = None

def echoEvent(pin):
    global t0, t1
    if (GPIO.input(pin)
        and not t0):
        t0 = time()
    elif (not GPIO.input(pin)
          and t0
          and not t1):
        t1 = time()
    print("Entfernung: %.1f cm"
          % ((t1-t0)
             *34300/2))
    t0 = None
    t1 = None
```

Der Event-Handler ruft die Funktion mit der Nummer des betreffenden Pins als Parameter auf. Bei steigender Flanke wurde eine neue Messung gestartet, bei fallender eine laufende beendet. Diese beiden Fälle lassen sich leicht unterscheiden, indem Sie den aktuellen Pegel des Pins überprüfen: Ist der Pin High, so wurde die Funktion wegen einer steigenden Signalflanke am Anfang einer Messung aufgerufen und Sie speichern die Zeit in der globalen Variable t0. Ist der Pegel des Pins Low, war es die fallende Signalflanke am Ende der Messung, dann landet die aktuelle Zeit in der Variablen t1.

Stückwerk

Doch es ist nicht garantiert, dass die Funktion `echoEvent()` am Stück abgearbeitet wird, denn der Event-Handler ist währenddessen weiter scharf geschaltet. Sollte der Raspi, während `echoEvent()` gerade ausgeführt wird, wieder eine Signalflanke entdecken,

so würde er die aktuell laufende Funktion unterbrechen und sie neu starten – wodurch die Variablen `t0` oder `t1` überschrieben würden. Deshalb prüft die Funktion, ob bereits eine `t0` gesetzt ist – falls ja, läuft bereits eine Messung und die steigende Signalflanke wird ignoriert.

Bei einer fallenden Signalflanke, die das Ende einer Messung anzeigt, muss zuvor eine `t0` gesetzt worden sein – `t1` hingegen muss leer sein, denn andernfalls läuft noch die Auswertung einer früheren Messung. Erst wenn das Ergebnis der früheren Messung angezeigt wurde, leert `echoEvent()` die Variablen `t0` und `t1`, wonach die nächste Messung angestoßen werden kann. Das geschieht über eine verkürzte Variante der Funktion `Dist()`, die nur noch das Trigger-Signal sendet, aus der bekannten `while`-Schleife heraus:

```
def Dist():
    global triggerPin
    GPIO.output(triggerPin, GPIO.LOW)
    sleep(0.000010)
    GPIO.output(triggerPin, GPIO.HIGH)

try:
    while True:
        Dist()
        sleep(1)
except KeyboardInterrupt:
    GPIO.cleanup()
```

Servo unter Kontrolle

Auch Modellbau-Servos und Fahrtenregler für Motoren kann der Raspberry Pi per Bit Banging direkt ansteuern, Sie benötigen dafür im Idealfall nicht einmal zusätzliche Elektronikkomponenten. Gerade Miniatur-Servos arbeiten oft schon bei 3,3 Volt Versorgungsspannung, die Sie an Pin 1 des Raspi abgreifen und am roten Servo-Kabel einspeisen können. Ihre volle Kraft entfalten Servos aber erst bei 5 Volt, die an den Pins 2 und 4 des Raspi anliegen – Servos mit größeren Stellkräften und besonders schnelle Digital-Servos sollten Sie aber besser über ein externes 5-Volt-Netzteil versorgen, da diese sehr hohe Spitzenströme ziehen. Der Masseanschluss (Braun oder Schwarz) eines Servos gehört in jedem Fall, gegebenenfalls zusammen mit dem Masseanschluss des externen Netzteils, an Pin 6 des Raspi oder an einen der anderen Masseanschlüsse. Die Steuerleitung (Gelb oder Weiß) können Sie mit einem beliebigen GPIO-Ausgang verbinden, zum

Beispiel mit Pin 11. Die maximal 3,3 Volt der Raspi-Schaltausgänge genügen den meisten Servos.

Die Ansteuerung von Modellbau-Servos und Motorreglern erfolgt über Rechtecksignale mit definierter Pulsbreite und Pulsabständen. Bei einer Pulsbreite von 1,5 Millisekunden dreht sich der Servo in die Mittelstellung. Verkürzt man die Pulsbreite auf nur 0,5 Millisekunden, so dreht sich der Servo um 90 Grad in die eine Richtung, verlängert man die Pulsbreite auf 2,5 Millisekunden, dreht er sich um 90 Grad in die andere Richtung – jeweils von der Mittelstellung aus betrachtet. Bei Reglern bestimmt die Pulsbreite die Geschwindigkeit und die Drehrichtung des Motors, 1,5 Millisekunden Pulsbreite bedeuten hier Stopp. Ein Servo hält die zuvor eingenommene Position so lange, wie sich das Signal mindestens alle 25 Millisekunden (40 Hz) wiederholt – die meisten Servos unterstützen aber auch kürzere Abstände bis hinunter zu 5 Millisekunden (200 Hz), wodurch der Servo deutlich schneller reagieren kann.

Der Bereich für die Pulsbreite von 0,5 bis 2,5 Millisekunden bleibt allerdings stets gleich, unabhängig von der Pulshäufigkeit, also der Pulsfrequenz. Um einen Servo mit dem Raspi in die Mittelstellung zu bringen, müssen Sie lediglich den GPIO-Pin für 1,5 Millisekunden auf High schalten und anschließend zum Beispiel für 9 Millisekunden auf Low. Das erledigt folgendes Programm:

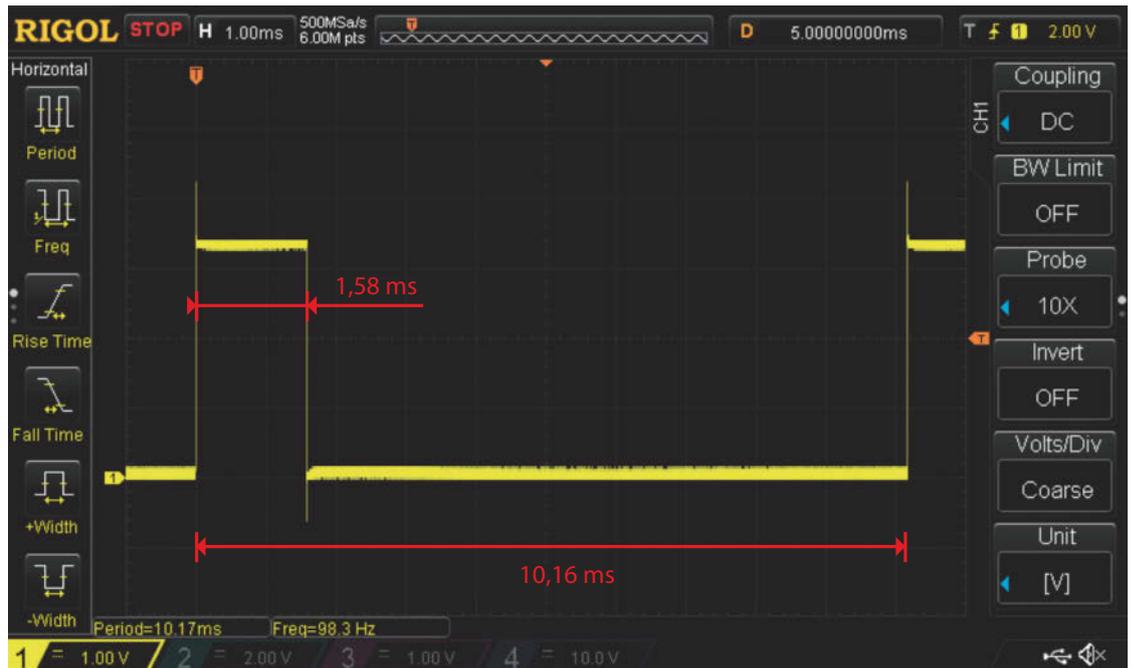
```
import RPi.GPIO as GPIO
from time import sleep

servopin = 11

GPIO.setmode(GPIO.BOARD)
GPIO.setup(servopin, GPIO.OUT,
            initial=GPIO.LOW)

for i in range(2):
    GPIO.output(servopin, GPIO.HIGH)
    sleep(0.0015)
    GPIO.output(servopin, GPIO.LOW)
    sleep(0.009)
GPIO.cleanup()
```

Da der GPIO-Pin mit einem Low-Signal initialisiert wurde, genügen bestenfalls zwei Pulse, also zwei Durchläufe (`range(2)`) der `for`-Schleife, um den Servo in die Mittelstellung zu bringen. Da es in der Praxis gelegentlich zu Störungen kommt, empfehlen wir mindestens fünf Schleifendurchläufe. Damit der



Wie präzise das Timing des Servo-Signals ist, hängt beim Bit Banging wesentlich von der Systemauslastung des Raspi ab. Schon ohne Last ist das Rechtecksignal fast 0,1 Millisekunden länger als programmiert.

Servo die Position auch dann hält, wenn beispielsweise die Kraft eines Ruders auf den Servoarm einwirkt, müssen Sie das Pulssignal hingegen ständig wiederholen – etwa mit vom Ultraschall-Entfernungsmesser bekannten `while`-Endlosschleife mit Abbruchmöglichkeit per `Strg+C` anstelle der `for`-Schleife:

```
try:
    while True:
        ...
except KeyboardInterrupt:
    GPIO.cleanup()
```

Der Raspberry Pi ist schnell genug, um so kurze Schlafzyklen von 1,5 Millisekunden einigermaßen genau einhalten zu können: Wie die Abbildung zeigt, hat das Rechtecksignal tatsächlich eine Breite von 1,58 Millisekunden; die horizontale Auflösung des Oszilloskops beträgt 1 Millisekunde pro Kästchen („H 1.00ms“ links oben). Auch die Wiederholung

weicht mit 10,16 Millisekunden nur geringfügig vom Soll ab.

Da es sich bei Raspberry Pi OS aber nicht um ein Echtzeit-Linux handelt (Real-Time Linux), sind je nach Auslastung des Minirechners sehr viel größere Abweichungen möglich. Denn während das Python-Programm schläft, kümmert sich das Betriebssystem um andere Prozesse, die im Hintergrund laufen, um Dateisystemoperationen, Datenverkehr über Ethernet und WLAN und vieles mehr.

Modellbau-Servos sind aber empfindlich genug, um auf Abweichungen im Bereich von Mikrosekunden zu reagieren. Deshalb kommt es abhängig von der Systemauslastung des Raspi dazu, dass der Servoarm zittert. In seltenen Fällen kann der Servo sogar von einem Anschlag zum anderen fahren, also in Bruchteilen einer Sekunde anschlagen, wenn es durch Multitasking zu größeren Timing-Problemen kommt. Gerade bei den schnellen Digital-Servos kann das ein Modell oder andere Hardware beschädigen.

PWM statt Bit Banging

Die Lösung ist, anstelle von Bit Banging einen der beiden PWM-Signalgeneratoren (Pulse Width Modulation) für die Ansteuerung der Servos zu benutzen, beispielsweise den an Pin 12 des Raspi. Einmal programmiert, wiederholt der PWM-Generator das Signal selbsttätig so lange, bis Sie es ändern oder wieder abschalten. Der PWM-Generator kennt zwei Parameter: die Frequenz, mit der er das Signal wiederholt, und die Einschaltdauer (Duty Cycle), die angibt, wie viel Prozent des Signaldurchgangs ein High-Pegel ausgegeben werden soll. Eine Frequenz von 100 Hz und ein Duty Cycle von 50 Prozent bedeuten, dass während einer Periodendauer von 10 Millisekunden die ersten 5 Millisekunden ein High-Pegel ausgegeben wird, danach ein Low-Pegel. Das ist zu viel für einen Servo. Für die Mittelstellung muss die Einschaltdauer, also Pulsbreite, bei einer Frequenz von 100 Hz genau 15 Prozent betragen:

```
import RPi.GPIO as GPIO
from time import sleep

servopin = 12
GPIO.setmode(GPIO.BOARD)
GPIO.setup(servopin, GPIO.OUT)
servo = GPIO.PWM(servopin, 100)
servo.start(15)
sleep(0.1)
servo.stop()
GPIO.cleanup()
```

Durch den Aufruf der Funktion `GPIO.PWM()` legen Sie fest, welchen Pin und welche Frequenz Sie verwenden wollen. Für Servos haben Sie die Wahl zwischen 40 Hz, was einem Pulsabstand von 25 Millisekunden entspricht, und 200 Hz, wodurch der Abstand auf 5 Millisekunden schmilzt. Häufig findet man in Beispielen eine Frequenz von 50 Hz, diese waren früher bei Modellbauempfängern üblich. Mit 100 Hz reagiert das Servo nicht nur schneller, es erleichtert auch die Berechnung des Duty Cycle – 5 Prozent entsprechen 0,5 Millisekunden, 15 Prozent 1,5 Millisekunden, also Mittelstellung, und 25 Prozent 2,5 Millisekunden. Die Prozentangabe muss übrigens nicht ganzzahlig sein, 10.51 ist ebenfalls zulässig. Das Beispiel ist ausreichend präzise, um frisch ausgepackte Servos vor dem Einbau zuverlässig in die Mittelstellung zu fahren.

Der `sleep()`-Aufruf vor `servo.stop()` sorgt dafür, dass der PWM-Generator überhaupt anläuft: Der

Raspi ist so schnell, dass er das PWM-Signal wieder abschalten würde, noch ehe das Signal einmal vollständig ausgesendet wurde. Der Servo würde also ohne `sleep()` nicht reagieren. Das hat auch Auswirkungen für die korrekte Ausnahmebehandlung von `Strg+C`, falls bei einem solchen Programmabbruch das Servo wieder in die Mittelstellung zurückgefahren werden soll, bevor der PWM-Generator abgeschaltet und der GPIO-Anschluss wieder freigegeben werden:

```
try:
    ...

except KeyboardInterrupt:
    servo.ChangeDutyCycle(15)
    sleep(0.1)
    servo.stop()
    GPIO.cleanup()
```

Der Vorteil von PWM gegenüber Bit Banging ist, dass Sie mehrere Servos gleichzeitig ansteuern können: Einmal programmiert laufen die PWM-Generatoren eigenständig und halten das Timing auch dann ein, wenn der Raspi belastet wird. Allerdings ist damit die Zahl der Servos auf zwei begrenzt, mehr Hardware-PWM-Generatoren besitzt der Raspberry Pi nicht.

Beim Bit Banging könnten Sie theoretisch mit allen 29 GPIO-Schaltpins Servos ansteuern, in der Praxis würde das Timing dabei aber viel zu ungenau. Die Lösung für mehr Servos am Raspberry Pi ist deshalb ein externer Servo-Controller, den Sie über den I2C-Bus des Raspberry Pi ansteuern. Passende Zusatzplatten für 8 oder gar 16 Servos gibt es für 10 bis 20 Euro.

Fazit

Der Raspi hat nur wenige eingebaute Hardware-Controller für externe Bussysteme. Per Bit Banging kann das leistungsfähige SoC des Raspberry Pi jedoch viele gebräuchliche Bussysteme emulieren, indem er einzelne Pins des GPIO-Anschlusses schnell genug ein- und ausschaltet oder einen der PWM-Generatoren benutzt, um das Timing des Bus-Protokolls zu emulieren. Man darf es aber damit nicht übertreiben und mit zu vielen Pins Bit Banging betreiben, sonst fällt der Raspi mit zunehmender Systemlast schnell aus dem Rahmen dessen, was die Protokolle erlauben. Für einzelne Sensoren oder Servos reicht die Leistung allemal. (mid) 

Listings zum Download

ct.de/wpra

IMPRESSUM

Redaktion

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.heise.de

Leserbriefe und Fragen zum Heft:
sonderhefte@ct.de

Die E-Mail-Adressen der Redakteure haben die Form xx@ct.de oder xxx@ct.de. Setzen Sie statt „xx“ oder „xxx“ bitte das Redakteurs-Kürzel ein. Die Kürzel finden Sie am Ende der Artikel und hier im Impressum.

Chefredakteur: Jobst-H. Kehrhahn (keh)
(verantwortlich für den Textteil)

Konzeption: Lutz Labs (ll)

Koordination: Pia Ehrhardt (pia), Angela Meyer (anm)

Redaktion: Niklas Dierking (ndi), Mirko Dölle (mid), Ronald Eikenberg (rei), Jan-Keno Janssen (jki), Lutz Labs (ll), Jan Mahn (jam), Pina Merkert (pmk), Dennis Schirmmacher (des), Jan Schüßler (jss), Peter Siering (ps), Dorothee Wiegand (dwi), Christof Windeck (ciw)

Mitarbeiter dieser Ausgabe: Thomas Jakobs, Maik Merten

Assistenz: Susanne Cölle (suc), Tim Rittmeier (tir), Christopher Tränkmann (cht), Martin Triadan (mat)

DTP-Produktion: Lara Bögner, Beatrix Dedek, Madlen Grunert, Lisa Hemmerling, Cathrin Kapell, Paula Krause, Kirsten Last, Martina Lübke, Steffi Martens, Marei Stade, Matthias Timm

Digitale Produktion: Christine Kreye (ltg.), Melanie Becker, Kevin Harte, Thomas Kaltschmidt, Martin Krefit, Pascal Wissner

Fotografie: Andreas Wodrich, Melissa Ramson

Illustration: Sven Hauth, Schülpi; Albert Hulm, Berlin; Andreas Martini, Wettin

Titel: Steffi Martens, www.freepik.com

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 Mühle, Falko Ossmann

Anzeigenleitung: Michael Hanke (-167)
(verantwortlich für den Anzeigenteil),
www.heise.de/mediadaten/ct

Anzeigenverkauf: Verlagsbüro ID GmbH & Co. KG,
Tel.: 05 11/61 65 95-0, www.verlagsbuero-id.de

Leiter Vertrieb und Marketing: Andre Lux (-299)

Service Sonderdrucke: Julia Conrades (-156)

Druck: Firmengruppe APPL Druck GmbH & Co. KG,
Senefelder Str. 3-11, 86650 Wemding

Vertrieb Einzelverkauf:
DMV DER MEDIENVERTRIEB GmbH & Co. KG
Meßberg 1
20086 Hamburg
Tel.: 040/3019 1800, Fax: 040/3019 145 1815
E-Mail: info@dermedienvertrieb.de
Internet: dermedienvertrieb.de

Einzelpreis: € 14,90; Schweiz CHF 27,90;
Österreich € 16,40; Luxemburg € 17,10

Erstverkaufstag: 26.04.2022

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 Verfügungsrecht des Verlages über. Sämtliche Veröffentlichungen erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes.

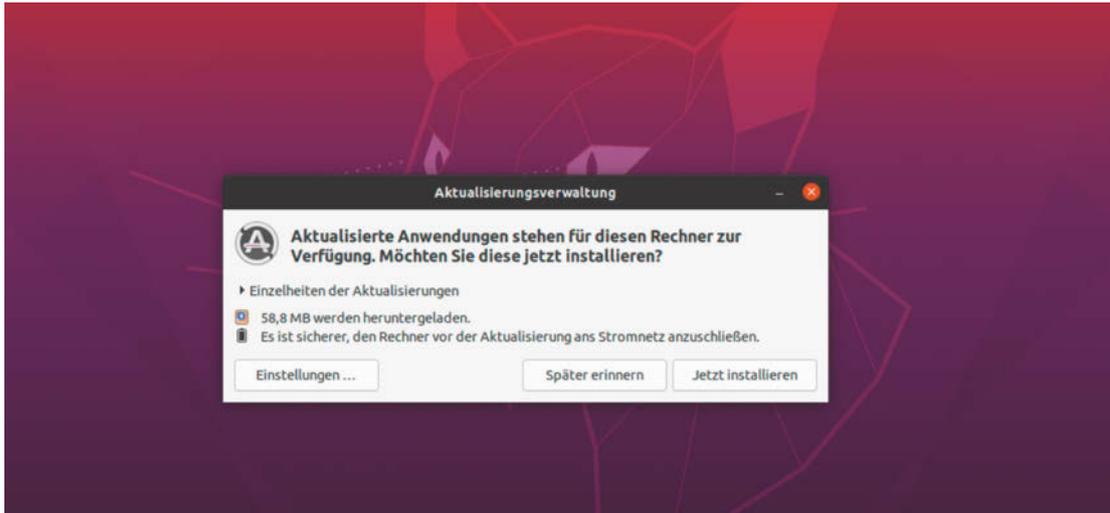
Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Hergestellt und produziert mit Xpublisher:
www.xpublisher.com

Printed in Germany.

Alle Rechte vorbehalten.

© Copyright 2022 by
Heise Medien GmbH & Co. KG



Automatische Paket-Updates anpassen

Immer auf dem neuesten Stand ohne einen Finger dafür zu rühren, das versprechen die automatischen Paket-Updates bei Debian-basierten Linux-Distributionen. Damit dabei keine Pakete unter den Tisch fallen, sollten Sie die Konfiguration individuell anpassen.

Von **Mirko Dölle**

Ganz so einfach ist es dann doch nicht: Das Paket `unattended-upgrades`, das es in Debian, Ubuntu, Raspberry Pi OS und praktisch allen anderen Debian-basierten Linux-Distributionen gibt, soll täglich verfügbare Updates bereits installierter Softwarepakete ohne Zutun des Anwenders einspielen. Doch je nach Distribution sind Anpassungen erforderlich, damit tatsächlich alle Aktualisierungen durchgeführt werden. Bei Raspberry Pi OS etwa funktioniert die vom Debian-Projekt übernommene Konfiguration nicht, ohne Anpassungen werden hier gar keine Pakete aktualisiert. Andererseits sind nicht immer alle Updates erwünscht; Server-Admins etwa sind nur auf Sicherheitsaktua-

lisierungen aus – Feature-Updates wollen sie erst auf ihre Stabilität und Kompatibilität überprüfen, sofern sie sie überhaupt in Betracht ziehen.

Allen Distributionen gleich ist, dass `unattended-upgrades` nach der Installation täglich alle Paketquellen aktualisiert. Welche Paket-Updates das Programm anschließend installiert, unterscheidet sich von Distribution zu Distribution. Bei Debian und Ubuntu etwa werden Pakete und Sicherheitsupdates der Standarddistribution automatisch eingespielt, nicht aber solche aus anderen Quellen. Dies ist in der zentralen Konfigurationsdatei `/etc/apt/apt.conf.d/50unattended-upgrades` festgelegt, allerdings in unterschiedlicher Art und Weise.

Neu geregelt

Debian verwendet bereits in Version 10 das aktuelle Format zur Angabe von Paketquellen, hier als Beispiel die Regel für Standardpakete der Distribution:

```
Unattended-Upgrade::Origins-Pattern {
    "origin=Debian,codename=${distro_codename},label=Debian";
    "origin=Debian,codename=${distro_codename},label=Debian-Security";
}
```

Die Origin-Angabe bezieht sich auf den Betreiber des Repositories, während das Label für das Team steht, das die Pakete bereitstellt. Das Debian Security Team etwa verwendet das Label „Debian-Security“, während das „Debian“-Team die Basispakete bereitstellt. Deshalb sind zwei Einträge erforderlich, um Standard- und Sicherheitsupdates abzudecken. Möchten Sie als Server-Admin lediglich die Sicherheitsaktualisierungen automatisch einspielen, können Sie den anderen Eintrag einfach entfernen oder auskommentieren.

Die Angabe des Codenames bewirkt, dass nur Pakete aktualisiert werden, die auch für die aktuell installierte Debian-Version vorgesehen sind. Dabei wird die Variable `${distro_codename}` unter Debian automatisch durch den Codename „bullseye“ ersetzt, bei Ubuntu 20.04 entsprechend durch „focal“. Es gibt

noch eine zweite Variable, `${distro_id}`, die mit dem Namen der Distribution ersetzt wird, also bei Debian durch „Debian“ und bei Ubuntu durch „Ubuntu“.

Ubuntu verwendet übrigens noch eine veraltete Schreibweise der Upgrade-Regeln, hier der Auszug für Standardpaket- und Sicherheitsupdates:

```
Unattended-Upgrade::Allowed-Origins {
    "${distro_id}:${distro_codename}";
    "${distro_id}:${distro_codename}-security";
}
```

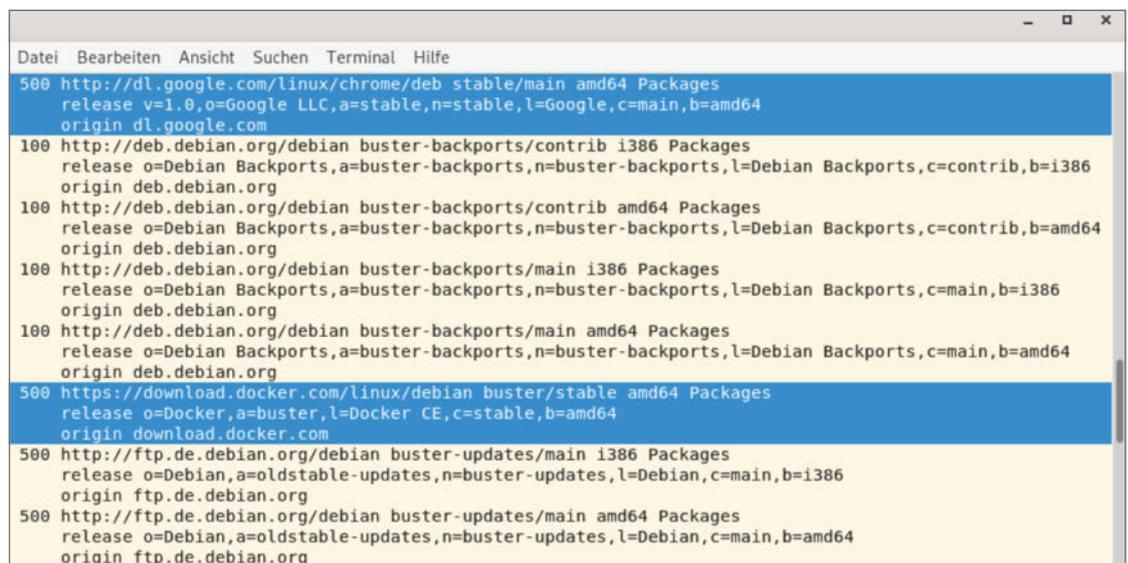
Dies ist gleichbedeutend mit folgender Konfiguration:

```
Unattended-Upgrade::Origins-Pattern {
    "origin=${distro_id},archive=${distro_codename}";
    "origin=${distro_id},archive=${distro_codename}-security";
}
```

Entwirrspiel

Die Felder für Codename und Archive werden von den Repositories höchst unterschiedlich benutzt, selbst innerhalb einer Distribution. Bei Debian etwa ist der Codename mit dem Namen der Distribution plus etwaiger Repository-Zusätze belegt, etwa

In welchen Feldern die jeweiligen Paketquellen Angaben zum Distributionsnamen oder zu den Repositories unterbringen, ist höchst unterschiedlich. Aus der Ausgabe des Befehls `apt policy` lassen sich aber leicht alle nötigen Informationen ablesen.



```
500 http://dl.google.com/linux/chrome/deb stable/main amd64 Packages
release v=1.0,o=Google LLC,a=stable,n=stable,l=Google,c=main,b=amd64
origin dl.google.com
100 http://deb.debian.org/debian buster-backports/contrib i386 Packages
release o=Debian Backports,a=buster-backports,n=buster-backports,l=Debian Backports,c=contrib,b=i386
origin deb.debian.org
100 http://deb.debian.org/debian buster-backports/contrib amd64 Packages
release o=Debian Backports,a=buster-backports,n=buster-backports,l=Debian Backports,c=contrib,b=amd64
origin deb.debian.org
100 http://deb.debian.org/debian buster-backports/main i386 Packages
release o=Debian Backports,a=buster-backports,n=buster-backports,l=Debian Backports,c=main,b=i386
origin deb.debian.org
100 http://deb.debian.org/debian buster-backports/main amd64 Packages
release o=Debian Backports,a=buster-backports,n=buster-backports,l=Debian Backports,c=main,b=amd64
origin deb.debian.org
500 https://download.docker.com/linux/debian buster/stable amd64 Packages
release o=Docker,a=buster,l=Docker CE,c=stable,b=amd64
origin download.docker.com
500 http://ftp.de.debian.org/debian buster-updates/main i386 Packages
release o=Debian,a=oldstable-updates,n=buster-updates,l=Debian,c=main,b=i386
origin ftp.de.debian.org
500 http://ftp.de.debian.org/debian buster-updates/main amd64 Packages
release o=Debian,a=oldstable-updates,n=buster-updates,l=Debian,c=main,b=amd64
origin ftp.de.debian.org
```

„bullseye“ für das aktuelle Debian Stable Release, aber „bullseye-backports“ für Pakete aus dem Backports-Repository. Das Debian-Security-Repository hingegen verwendet nicht den Distributionsnamen, sondern „oldstable“, „stable“ und „unstable“ für Debian Buster, Bullseye und Sid. Bei Ubuntu 20.04 steht im Archive-Feld für Sicherheits-Updates „focal-security“ und bei Backports „focal-backports“, im Codename-Feld hingegen steht stets nur der Name des Releases, also „focal“.

Bei Raspberry Pi OS steht im Archive-Feld nur „stable“ oder „unstable“, ohne weitere Zusätze, und im Codename der Name des Debian-Releases, von dem Raspberry Pi OS abgeleitet wurde, also aktuell „buster“ und demnächst „bullseye“. Und das Docker-Repository benutzt beim Archive-Feld „buster“ und beim Codename „stable“, also genau anders herum wie das Debian-Projekt. Auch das Component-Feld wird nicht immer einheitlich verwendet, hier sollte bei Debian-Paketen von lizenzrechtlich problematischer Software „non-free“ stehen, bei Ubuntu ist es „multiverse“. Kurz: Es gibt keine Regel, die nicht gebrochen wird. Aber es gibt eine einfache Methode, auf Anhieb die richtige Regel zu formulieren.

Der Schlüssel zur eigenen Upgrade-Regel ist der Befehl `apt policy`, der alle eingerichteten Paketquel-

len mit den für die Regeln notwendigen Daten auflistet. Die Abbildung auf Seite 103 zeigt einen Auszug dieser Liste eines Debian 10 Buster, bei dem die offiziellen Paketquellen für Google Chrome und Docker hinzugefügt wurden. Die jeweils zweite Zeile eines Repository-Eintrags enthält alle relevanten Angaben für `unattended-upgrades` – wobei die Feldnamen wie `origin` oder `label` zu `o` respektive `l` abgekürzt wurden, was `unattended-upgrades` ebenfalls unterstützt. Die einzige Stolperstelle ist `c`, das für `component` steht, `codename` wird mit `n` abgekürzt. Ansonsten ist die Kurzform stets der erste Buchstabe der Langform.

Allerdings unterstützt `unattended-upgrades` nicht alle Felder, die `apt policy` auflistet, Sie müssen sich in Upgrade-Regeln auf die fünf Felder `a`, `c`, `l`, `n` und `o` beschränken. Damit künftig auch Docker automatisch aktualisiert wird, reduzieren Sie die Ausgabezeile

```
release o=Docker,a=buster,
l=Docker CE,c=stable,b=amd64
```

von `apt policy` auf folgende Regel:

```
"o=Docker,a=buster,l=Docker CE,
c=stable";
```

```

pi@westpi:~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
pi@westpi:~ $ apt policy
Paketdateien:
 100 /var/lib/dpkg/status
      release a=now
   90 http://deb.debian.org/debian unstable/main armhf Packages
      release o=Debian,a=unstable,n=sid,l=Debian,c=main,b=armhf
      origin deb.debian.org
 500 http://archive.raspberrypi.org/debian buster/main armhf Packages
      release o=Raspberry Pi Foundation,a=testing,n=buster,l=Raspberry Pi Foundation,c=main,b=armhf
      origin archive.raspberrypi.org
 500 http://raspbian.raspberrypi.org/raspbian buster/rpi armhf Packages
      release o=Raspbian,a=stable,n=buster,l=Raspbian,c=rpi,b=armhf
      origin raspbian.raspberrypi.org
 500 http://raspbian.raspberrypi.org/raspbian buster/non-free armhf Packages
      release o=Raspbian,a=stable,n=buster,l=Raspbian,c=non-free,b=armhf
      origin raspbian.raspberrypi.org
 500 http://raspbian.raspberrypi.org/raspbian buster/contrib armhf Packages
      release o=Raspbian,a=stable,n=buster,l=Raspbian,c=contrib,b=armhf
      origin raspbian.raspberrypi.org
 500 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages
      release o=Raspbian,a=stable,n=buster,l=Raspbian,c=main,b=armhf
      origin raspbian.raspberrypi.org
Mit Pinning verwaltete Pakete:
pi@westpi:~ $

```

Die Raspberry Pi Foundation verwendet gleich zwei verschiedene Herkunfts- und Label-Angaben für ihre Paket-Repositories. Diese passen allerdings nicht zu den Standard-Upgrade-Regeln, die nur Pakete des Debian-Projekts aktualisieren würden.

Entscheidend ist, dass Sie keine zusätzlichen Leerzeichen einfügen: Das im Label `Docker CE` enthaltene Leerzeichen ist das einzige, das innerhalb der Anführungszeichen der Regel erlaubt ist.

Raspi repariert

Mit diesem Wissen gelingt es Ihnen auch, die Regeln in Raspberry Pi OS zu korrigieren, die standardmäßig nur die beiden vom Debian-Projekt übernommenen Regeln enthält.

Die Pakete der Raspberry Pi Foundation tragen als Herkunftsangabe sowie als Label jedoch die Angaben `Raspberry Pi Foundation` oder `Raspbian`. Daher kommen die beiden Standard-Regeln niemals zur Anwendung. Die richtigen Angaben bekommen Sie auch in diesem Fall mit dem Befehl `apt policy`. Daraus können Sie die korrekten Upgrade-Regeln destillieren:

```
"o=Raspberry Pi Foundation,a=testing,↵
↵n=buster,L=Raspberry Pi Foundation";
"o=Raspbian,a=stable,n=buster,↵
↵L=Raspbian";
```

Haben Sie die Regeln im Abschnitt `Unattended-Upgrade::Origins-Pattern` der Datei `/etc/apt/apt.conf.d/50unattended-upgrades` eingetragen, können Sie mit dem Befehl `sudo unattended-upgrades --dry-run` einen Probelauf starten. Das Ergebnis finden Sie

am Ende der Log-Datei `/var/log/unattended-upgrades/unattended-upgrades.log` – im Erfolgsfall werden dort alle Pakete zur Aktualisierung vorgeschlagen, die mit dem Befehl `sudo apt upgrade --dry-run` ebenfalls aktualisiert würden. Der Parameter `--dry-run` bewirkt bei beiden Kommandos, dass die Aktualisierung nicht tatsächlich erfolgt. So brauchen Sie nicht abzuwarten, bis es wieder neue Updates gibt, um Änderungen an Ihrer Konfiguration zu testen.

Vertrauen ist gut ...

Nach der Einrichtung ist `unattended-upgrades` bereits scharf geschaltet. Wenn Sie die Arbeiten an der Konfiguration erst am nächsten Tag oder noch später fortsetzen wollen, sollten Sie die automatische Aktualisierung mit dem Befehl `sudo dpkg-reconfigure unattended-upgrades` zunächst abschalten, bis Sie mit der Konfiguration zufrieden sind und Sie die Automatik auf dem gleichen Weg wieder aktivieren. Anschließend sollten Sie von Zeit zu Zeit die Log-Datei von `unattended-upgrades` auf aktualisierte Pakete kontrollieren. So verhindern Sie, dass Sie Opfer eines Angriffs über eine vermeintlich längst geschlossene Sicherheitslücke werden – weil bei Ihnen etwa aufgrund eines falschen Leerzeichens oder Tippfehlers keine Updates automatisch eingespielt wurden, obwohl Sie dafür eigens `unattended-upgrades` installiert haben und sich deshalb in Sicherheit wähnten. (mid) **ct**

Lift-off oder Bruchlandung?

Heft + PDF
mit 29% Rabatt



- ▶ Erste Handgriffe und Einrichten von Windows 11
- ▶ c't-Notfall-Windows hilft bei Virensuche, Datenrettung und Hardware-Diagnose

- ▶ Hardware-Anforderungen umsetzen und Setup ohne Hardware-Prüfung
- ▶ Grafische Oberfläche für Android-Apps und Linux-Anwendungen

Heft für 14,90 € • PDF für 12,99 € • Bundle Heft + PDF 19,90 €

 shop.heise.de/ct-windows11

Retro-Spiele auf dem Raspi Zero 2 zocken

Der Raspberry Pi Zero 2 W ist deutlich leistungsfähiger als sein Vorgänger. Wir haben eine Emulator-Sammlung darauf installiert und geprüft, wie gut Retro-3D-Spiele am Fernseher und unterwegs in einem Game-Boy-Case laufen.

Von **Dennis Schirmmacher**



| | |
|--|-----|
| Retro-Spiele auf dem Raspi Zero 2 zocken | 106 |
| Raspi als Client für Spotify und AirPlay | 110 |
| 4K-Filme in HDR auf dem Raspi 4 | 112 |

Der Kleincomputer Raspberry Pi hat von Anfang an die Herzen der Retro-Gamer erobert. Schon die erste Version des Zero mit seiner Single-Core-CPU konnte viele Super-Nintendo-Titel flüssig emulieren. Für anspruchsvollere Playstation-1-Titel mit 3D-Grafik reichte die Leistung allerdings nicht aus und aus dem Spielvergnügen wurde eine Diashow.

Ein Blick auf das Datenblatt des Raspberry Pi Zero 2 W weckt Hoffnung: Auf dem Papier zieht er mit seiner Quad-Core-CPU (1 GHz) fast mit dem Raspberry Pi 3B gleich. Wir haben verschiedene Emulatoren und Spiele im Wohnzimmer am Fernseher ausprobiert, den Einplatinencomputer sogar übertaktet und ihn zum Zocken für unterwegs in das Game-Boy-Gehäuse GPi Case verbaut (siehe S. 109).

Die Qual der Wahl

Wie für andere Raspis gibt es auch für den Zero 2 verschiedene Linux-Distributionen wie RetroPie und Recalbox, die eine ganze Palette an Emulatoren für Retro-Konsolen und -Computer mitbringen. Wir spielen die Einrichtung anhand der Recalbox-Distribution durch. Da sich die Installation und Konfiguration von Recalbox für den Betrieb im GPi Case im Detail von der Nutzung an einem TV-Gerät unterscheidet, gehen wir darauf in einem gesonderten Abschnitt ein.

Am einfachsten gelingt die Installation mit dem kostenlosen Tool „Raspberry Pi Imager“ für Linux, macOS und Windows (siehe ct.de/w5qy). Führen Sie die Anwendung aus und klicken Sie auf „OS WÄHLEN“, dann weiter auf „Emulation and game OS/Recalbox/Recalbox - Raspberry Pi Zero 2“. Schließen Sie nun eine microSD-Karte mit mindestens 4 GByte an den Computer an und wählen Sie diese im Installer aus. Im Anschluss wird die Karte komplett gelöscht und Recalbox darauf installiert.

Der erste Start am TV-Gerät

Wollen Sie an einem Fernseher zocken, stecken Sie die microSD-Karte nun in den passenden Slot des Zero 2 und schließen Sie ihn an eine Stromversorgung und einen Monitor an. Beim ersten Start wird die Installation der Emulator-Sammlung automatisch abgeschlossen. Währenddessen kann man sich ein paar nützliche Tipps zur Bedienung und Infos zum Funktionsumfang anschauen.

Erst nach der vollständigen Installation taucht die Partition SHARE auf der microSD-Karte auf. Schließen Sie die Karte erneut an einen Computer an und

kopieren Sie die Sicherheitskopien Ihrer Spiele (ROMs) darauf. Unter Windows schleicht sich manchmal ein Bug ein und die Partition erscheint nicht im Explorer. Ist das der Fall, rufen Sie in der Computerverwaltung unter „Datenspeicherverwaltung“ den Punkt „Datenspeicher“ auf. Nach einem Rechtsklick auf „SHARE“ wählen Sie die Option „Laufwerksbuchstaben und -pfade ändern“ aus und klicken auf „Hinzufügen“. Nun sollte die Partition auch unter Windows sichtbar sein. Alternativ können Sie ROMs auf einen am Raspi angeschlossenen USB-Stick kopieren oder eine Netzwerkfreigabe mit Spielen einrichten oder die Spiele per WLAN auf die SD-Karte transferieren.

Mit aktiver Internetverbindung aktualisieren Sie Recalbox in den Einstellungen. Die Entwickler veröffentlichen immer wieder neue Versionen etwa mit zusätzlichen Funktionen. Über das Menü ist ein Scraper erreichbar, der aus frei zugänglichen Spiele-Datenbanken Screenshots und weitere Infos lädt. Damit können Sie das Auswahlmenü für Spiele anreichern.

Die Bedienung von Recalbox ist voll auf Gamepads ausgelegt, zum Beispiel über einen PS4-Controller, der via Bluetooth oder USB Anschluss findet. In den Einstellungen konfigurieren Sie die Tastenbelegungen. Ein PS5-Controller machte im Test allerdings Probleme: Der Sound kam aus dem Lautsprecher des Gamepads, außerdem wurden Eingaben oft nicht erkannt.

Stilgerecht zocken

Wer das GPi Case besitzt, muss bei der Installation mit dem „Raspberry Pi Imager“ den Eintrag „Recalbox - GPi Case + Raspberry Pi Zero“ auswählen und das System auf einer SD-Karte installieren. Die Bedienung, das Kopieren von ROMs et cetera ist identisch mit der normalen Recalbox-Version. Bei der speziellen Distribution sind unter anderem die Bildausgabe und die Tastenbelegungen für das Gehäuse optimiert. Wer das Case bereits mit einem Zero 1 nutzt und auf den Zero 2 umsteigen will, muss Recalbox zwingend komplett neu installieren.

Um den Zero 2 im Spielmodulgehäuse des GPi Case zu platzieren (siehe S. 108), müssen Sie das Modul aufschrauben und den Kleincomputer mit der darin liegenden Platine verbinden. Dafür richten Sie den Raspi so über der GPIO-Schnittstelle aus, dass Sie problemlos das Micro-USB-Kabel anschließen können. Platzieren Sie den Verbund nun im Gehäuse und schrauben Sie es zu. Die SD-Karte schieben Sie seitlich am Gehäuse ein.



Beeindruckend: Der PS1-Titel „Tekken 3“ mit 3D-Grafik läuft auf dem Raspi Zero 2 mit butterweichen 60 Bildern pro Sekunde.

Damit die Erstinstallation nach dem ersten Einschalten fehlerfrei abläuft, sollten Sie eine stabile Stromversorgung über das mitgelieferte USB-Kabel und ein Netzteil oder eine Powerbank sicherstellen. Beim Batteriebetrieb muss man beachten, dass der Zero 2 keine Alkaline-Batterien mag und oft gar nicht erst startet oder beim Spielen einfriert. Im Test haben wir das GPi Case mit dem Zero 2 problemlos mit NiMH-Akkus (2500 mAh) betrieben. Damit konnten wir, je nachdem wie viel Rechenleistung ein Spiel einfordert und wie hell der Bildschirm ist, zwischen drei und fünf Stunden spielen. Selbst mit dem im Gehäuse verbauten übertakteten Zero 2 konnten wir „Tekken 3“ über vier Stunden problemlos zocken, bevor den Akkus die Puste ausging. Die erhöhte Hitze-

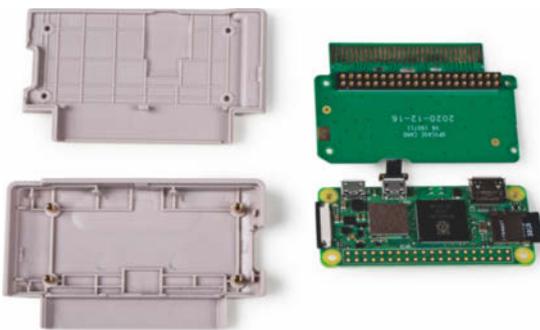
entwicklung aufgrund der Übertaktung war kein Problem und das System lief stabil.

Los! Zocken!

Spiele gibt der Zero 2 mit maximal 1920 × 1080 Bildpunkten mit 60 Bildern pro Sekunde (frames per second, fps) aus. Schon der Zero 1 konnte viele Retro-Konsolen wie das SNES von Nintendo emulieren. Titel wie „Super Mario World“ laufen bereits mit dem Vorgängermodell flüssig. Anspruchsvolle Spiele wie „Super Mario World 2: Yoshi’s Island“ ruckeln jedoch und mehr als 40 Bilder pro Sekunden sind nicht drin. Das Problem bei diesem Spiel ist, dass der Emulator noch den zusätzlich im Spielmodul verbauten Super-FX-Chip emulieren muss.

Hiermit ist selbst der Zero 2 überfordert und kommt nur auf 50 Bilder pro Sekunde. Im Test konnten wir unser Modell jedoch stabil auf 1,35 GHz übertakten und der Titel lief fortan nahezu perfekt spielbar mit 60 Bildern pro Sekunde. Das funktionierte bei uns sogar ohne Kühlung und selbst nach mehreren Stunden lief die Emulation noch stabil. Das Übertakten geschieht natürlich auf eigene Gefahr. Auch bei „Donkey Kong Country 2“ und „Star Fox“ war die Steigerung der Taktrate das Zünglein an der Waage für flüssigen Spielgenuss.

Überraschenderweise laufen PS1-Spiele mit 3D-Grafik wie „Tekken 3“ bereits mit dem Standardtakt flüssig. Mit dem Zero 1 ist das Kampfspiel unspielbar. Auch grafisch anspruchsvolle PS1-Spiele wie „Einhander“, „Colony Wars“ und „Metal Slug X“ laufen flüssig. Der N64-Titel „Super Mario 64“ lief bei uns zumindest im ersten Level mit relativ stabilen



Für den Betrieb mit dem GPi Case steckt der Raspi Zero 2 in einem Spielmodulgehäuse, das eine Platine mit GPIO- und Micro-USB-Schnittstelle mitbringt.

Download Raspberry Pi Imager

ct.de/w5qy

Neuer Spieljunge

Das GPI Case sieht nicht nur aus wie ein Game Boy, sondern fühlt sich auch so an.

Das Raspi-Gehäuse ist eine detaillierte Nachbildung des Original-Game-Boys. Darin steckt ein mit einer GPIO- und Micro-USB-Schnittstelle versehenes einsteckbares Spielmodul, in dem ein Raspi Zero oder Zero 2 Platz findet. Ist der Raspi angesteckt, muss man nur noch eine Retro-Emulator-Sammlung installieren.



Die Bildqualität des blickwinkelstabilen IPS-Bildschirms überzeugt. Über ein seitliches Rädchen passt man die Leuchtkraft an. Der interne Lautsprecher klingt extrem blechern und der Kopfhörerausgang ist schlecht abgeschirmt; mit empfindlichen In-Ear-Kopfhörern wird man von Störgeräuschen geplagt. Das Steuerkreuz fühlt sich wie beim

Original leicht schwammig an. Der Druckpunkt der Knöpfe gefällt. Insgesamt lassen sich die meisten Spiele gut bis sehr gut steuern, außer man spielt Titel wie Star Fox, die Schultertasten auf der Oberseite von Gamepads voraussetzen. Diese Tasten sind beim GPI Case mittig auf der Rückseite platziert, was zu Handkrämpfen führt.

Die Stromversorgung übernehmen ein USB-Netzteil oder drei AA-Batterien. Für den stabilen Betrieb sind NiMH-Akkus dringend empfohlen. Je nach CPU-Last und Bildschirmhelligkeit konnten wir unterwegs zwischen drei und fünf Stunden zocken.

GPI Case

Game-Boy-Case für Raspberry Pi Zero

| | |
|-----------------------------|-----------------------------------|
| Hersteller, URL | Retroflag, retroflag.com |
| Abmessungen / Gewicht | 13,5 cm × 8,1 cm × 3,2 cm / 183 g |
| Bildschirmgröße / Auflösung | 2,8 Zoll / 320 × 240 Pixel |
| Kompatibilität | Raspberry Pi Zero (W) 1 und 2 |
| Preis | 60 € |

30 Bildern pro Sekunden. Die Engine des Spiels ist nicht für mehr fps ausgelegt. Perfekt läuft das Nintendo 3D-Hüpfspiel aber noch nicht und kleinere Ruckler trüben immer mal wieder den Spielspaß.

Fazit

Für Retro-Gamer ist der Zero 2 ein äußerst lohnendes und mit rund 16 Euro (plus Versand) vergleichsweise günstiges Upgrade. Alle Fans von 3D-Spielen auf etwa der PS1 können ohne nachzudenken zugreifen. Man darf aber nicht vergessen, dass längst nicht alle Spiele fehlerfrei und flüssig laufen. Außerdem fehlt dem Zero 2 die Leistung, um die Konsolengeneration nach dem N64 und der PS1 wie Segas Dreamcast zu emulieren. Da beißt sich aber selbst ein Raspi 4 an vielen Titeln die Zähne aus.

Für ältere Konsolen reicht die Leistung des Zero 2 aber oft dicke aus und in Full-HD-Auflösung mit 60 Bildern pro Sekunde kommt jede Menge Spielspaß auf. Weil der Zero 2 so kompakt gebaut ist, zockt man mit einem geeigneten Gehäuse wie dem GPI Case sogar unterwegs. (des) **ct**

Ist der Zero 2 via WLAN mit dem Internet verbunden, kann man Informationen zu Titeln inklusive Bildern aus diversen freien Spiele-Datenbanken herunterladen.



Raspi als Client für Spotify und AirPlay

Selbst ein Raspi Zero 2 reicht aus, um unsmarten AV-Receiver oder analogen Verstärkern Musikstreaming-Fähigkeiten zu verleihen. Mit Raspotify und Shairport Sync klappt das mit wenig Aufwand.

Von **Dennis Schirmacher**

Wenn der AV-Receiver (AVR) seinen Dienst noch zuverlässig verrichtet, aber kein Audiostreaming beherrscht, müssen Sie das Gerät nicht gleich durch ein aktuelles ersetzen. Sogar die Leistung eines Raspberry Pi Zero 2 W reicht aus, um einen AVR oder Stereoverstärker aufzurüsten und Musik via WLAN zu streamen. Mit dem Raspi 2 bis 4 funktioniert das ebenfalls problemlos.

Damit Smartphones den Raspi als Spotify-Connect-Client im Netzwerk erkennen, gibt es die Open-Source-Software Raspotify. Auf einem Raspberry Pi Zero W oder dem Raspberry Pi 1 läuft der Spotify-Client aber nicht mehr ohne Weiteres, da der ARMv6-Support mittlerweile gestrichen wurde. Wer Raspotify dennoch auf der Hardware nutzen will, muss die letzte mit ARMv6 kompatible Version 0.31.8.1 installieren. Dafür gibt es aber keinen Support mehr. Zu empfehlen ist das nicht: In unserem Test mit einem Raspi Zero W kam es in Verbindung mit einem DAC (ADI-2 Pro FS) zu Knacksern bei der Musikwiedergabe.

Vorbereitungen

Als Basis genügt das schlanke Betriebssystem Pi OS Lite (Bullseye) in der 32- oder 64-Bit-Version. Für die Nutzung von Raspotify ist ein Spotify-Premium-Account nötig. Raspotify installieren Sie mit

```
sudo apt-get -y install curl && -sL ↵  
chttps://dtcooper.github.io/raspotify/install.sh | sh
```



Das Skript installiert nicht nur Raspotify, sondern fügt auch das Repository des Entwicklers hinzu. Das hat den Vorteil, dass die Paketverwaltung neben dem Betriebssystem auch den Spotify-Client aktualisiert. In der Spotify-App taucht der Raspi sofort nach der Installation als „raspotify (raspberrypi)“ auf und lässt sich als Ziel für die Wiedergabe auswählen.

Damit der Raspi als AirPlay-Client „Raspberrypi“ im Netzwerk erscheint, müssen Sie Shairport Sync installieren. Das ist etwas aufwendiger, da dafür benötigte Pakete in Pi OS veraltet sind. Auf GitHub erklärt der Entwickler verständlich, wie das Übersetzen und die Installation Schritt für Schritt ablaufen (siehe [ct.de/wxeb](#)).

Klang-Tuning

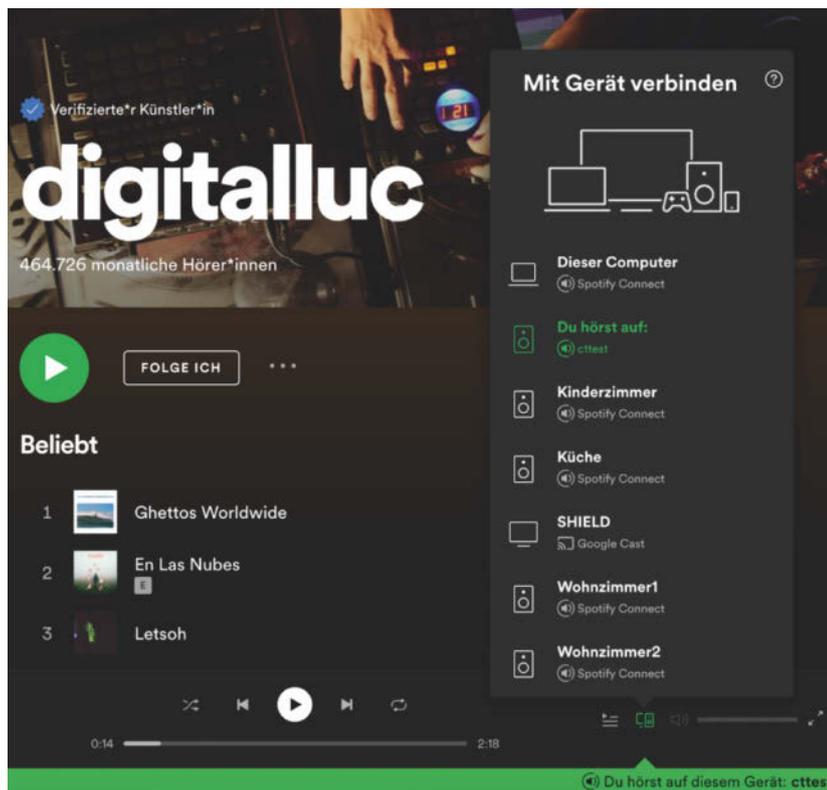
Um die Soundqualität für Spotify zu optimieren, müssen Sie die Konfigurationsdatei `/etc/raspotify/conf` bearbeiten. Die folgenden Zeilen aktivieren Sie,

indem Sie das Doppelkreuz-Zeichen am Zeilenanfang löschen. Ändern Sie in der Zeile `LIBRESPOT_BITRATE=` den Wert von 160 auf 320. Erst dann streamt Spotify Musik in der bestmöglichen Qualität mit einer Bitrate von 320 kbit/s.

Standardmäßig spielt Raspotify alle Songs mit einer gleichbleibenden Lautstärke ab (Normalisation). So muss man nicht ständig zum Lautstärkeregler greifen. Leider beeinflusst das die vom Künstler beabsichtigte Dynamik eines Songs negativ. Um das abzuschalten, müssen Sie die Zeile `LIBRESPOT_ENABLE_VOLUME2_E_NORMALISATION=` mit einem Doppelkreuz auskommentieren. Außerdem empfiehlt es sich, in der Zeile `LIBRESPOT_VOLUME_CTRL=` die digitale Lautstärkeregelung zu deaktivieren, indem Sie den Wert `linear` durch `fixed` ersetzen. Das hat zwar den Nachteil, dass man die Lautstärke nur noch am AVR und nicht mehr am Smartphone regeln kann, dafür kommt das Musiksinal, wenn man zusätzlich `LIBRESPOT_INITIAL_VOLUME="100"` aktiviert, mit vol-

Installationsanleitung AirPlay

ct.de/wxeb



Direkt nach der Installation von Raspotify spielt der Raspi ohne weitere Konfiguration Musik aus dem Streamingdienst ab.

lem Dynamikumfang am AVR an. Das sollte man unter `/etc/shairport-sync.conf` mit der Option `ignore_volume_control="yes"` auch für AirPlay machen. Dafür muss man in der Zeile nur die Zeichen `//` am Anfang entfernen.

Dank bei beiden Clients standardmäßig aktivem Gapless Play laufen Live-Alben ohne Unterbrechung zwischen Songs. Um den Namen des Spotify-Clients anzupassen, tippen Sie die gewünschte Bezeichnung hinter `LIBRESPOT_NAME=` ein. Bei AirPlay geht das bei `name=`.

Richtig anschließen

Um das Audiosignal vom Raspi Zero 2 und Raspi 2 bis 4 digital an einen AVR zu leiten, benötigen Sie lediglich ein HDMI-Kabel. Steht im Wohnzimmer noch ein analoger Stereo-Verstärker, kommt das Audiosignal beim Raspi 2 bis 4 nach einer Digital-Analog-Konvertierung (DAC) via 3,5-mm-Klinke aus dem Kleinstcomputer in den Verstärker. Die Audioqualität des Ausgangs ist aber nicht sehr gut und anspruchsvolle Ohren sollten einen externen DAC einsetzen. Beim Raspi Zero 2 W ist ein DAC obligatorisch, da er keinen analogen Audioausgang hat.

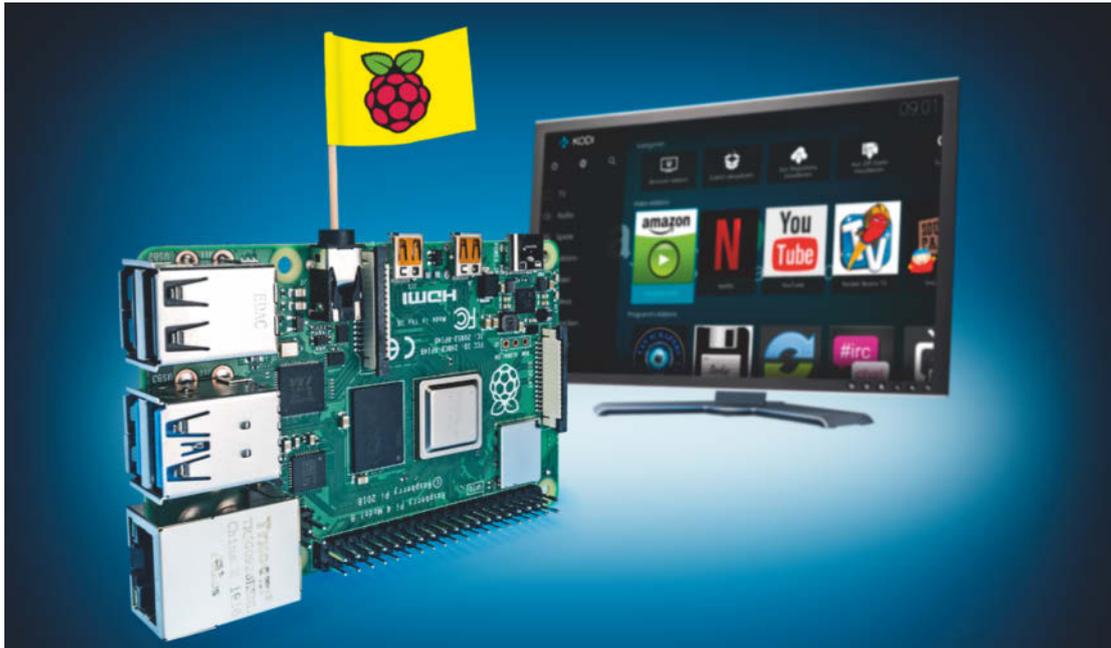
Das Aufrüsten gelingt kostengünstig für rund 15 Euro mit dem HifiBerry DAC+, den man via 40-Pin-GPIO-Anschluss mit einem Raspi verbindet. Wer noch hochwertigeren Sound möchte, muss mehr Geld ausgeben. Wir haben das erfolgreich mit dem Qudelix 5K (120 Euro) und dem RME ADI-2 DAC FS (960 Euro) ausprobiert.

Da viele DACs USB Class Compliant sind, erkennen Raspis sie ohne Treiberinstallation. Damit die Wiedergabe klappt, schließen Sie den DAC via USB als externe Soundkarte an und ändern unter `/usr/share/alsa/alsa.conf` das Ausgabegerät. Passen Sie dafür die Einträge wie folgt an: `defaults.ctl.card 1, defaults.pcm.card 1`

Falls die Lautsprecher stumm bleiben, ziehen Sie das HDMI-Kabel ab und probieren Sie es noch mal. In unserem Versuch konnten wir mit beiden DACs Musik ohne Störungen wiedergeben.

Fazit

Der Spotify- und AirPlay-Client auf einem Raspberry Pi Zero 2 W laufen in unserem Test seit mehreren Monaten absolut stabil. Die Musikübertragung via WLAN gelingt stets ohne Aussetzer. Es muss also nicht immer gleich ein neuer AVR sein, wenn man smarte Funktionen vermisst. (des) **ct**



4K-Filme in HDR auf dem Raspi 4

Die aktuelle Version des Open-Source-Mediacenters Kodi spielt Filme auf dem Raspi 4 nicht nur in 4K ab, sondern auch mit HDR. Sogar 3D-Sound geht.

Von **Dennis Schirmmacher**

Das Mediacenter Kodi ist seit Erscheinen des Raspi 4 im Jahr 2019 kompatibel mit dem Einplatinencomputer. Filme liefen bereits in einer frühen Entwicklerversion mit 3840×2160 Bildpunkten. Blockbuster mit erhöhtem Kontrastverhältnis und erweitertem Farbumfang (High Dynamic Range, HDR) spielte der Raspi 4 trotz kompatibler HDMI-2.0-Ausgänge aber nur in SDR ab. Nun haben die Kodi-Entwickler nachgelegt und die HDR-Kompatibilität für das übliche Format HDR10 implementiert. Die dynamische HDR-Variante Dolby Vision wird nicht

unterstützt. Besitzer älterer Raspi-Modelle kommen nicht in den Genuss ultrahochoflösender Bilder, denn das Abspielen von 4K-Inhalten klappt ausschließlich mit dem Raspi 4.

Als Basis für Kodi auf einem Raspi dient die auf die Medienwiedergabe spezialisierte Linux-Distribution LibreELEC. Während des Abspielens eines mit HEVC/H.265 kodierten 4K-Films mit HDR im MKV-Container auf unserem Raspi 4 mit 2 GByte RAM waren rund 30 Prozent des Arbeitsspeichers belegt. Demzufolge sollte Kodi auch problemlos auf einem Raspi 4

mit 1 GByte RAM laufen. Maximal findet die Bildausgabe in 4K mit 60 Bildern pro Sekunde statt.

Installation

Wer auf dem Raspi 4 bereits Kodi nutzt, sollte aufgrund von massiven Änderungen im Kernel eine Neu-Installation vornehmen. Am einfachsten gelingt die frische Installation auf einer SD-Karte mit dem offiziellen Installations-Tool „LibreELEC USB-SD Creator“ für Linux, macOS und Windows. Alternativ kann man das Image beispielsweise unter Windows mit dem kostenlosen Win32 Disk Imager auf eine SD-Karte kopieren.

Nach dem Start des offiziellen Tools wählen Sie bei „Version“ im linken Reiter „Raspberry Pi 4 and 400“ und im rechten Feld „LibreELEC-RPi4.arm-10.0.2.img.gz“. Nun müssen Sie im mit „3“ gekennzeichneten Feld noch die am Computer angeschlossene SD-Karte auswählen. Nach einem Klick auf „Schreiben“ startet der Kopiervorgang. Im Anschluss stecken Sie die SD-Karte in den Raspi 4.

Anschließen

Vorsicht beim Anschluss des Raspis an einen Fernseher beziehungsweise AV-Receiver: Die Micro-HDMI-Buchsen des Kleinstcomputers sind nur aufgelötet

und sie reagieren sehr empfindlich auf ruckelnde Stecker. Wir haben trotz aller Vorsicht beim Hantieren mit einem Micro-HDMI-auf-HDMI-Adapter einen der zwei HDMI-Ausgänge zerstört. Dank standardmäßig aktivem HDMI-CEC-Plug-in gelingt die Bedienung von Kodi bequem über die Fernbedienung des Fernsehers.

Nicht wundern: Sollte sich Ihr TV-Gerät wie von Geisterhand einschalten, ist Kodi dafür verantwortlich. Das passierte bei uns einige Male. Aufgrund von Marotten des CEC-Standards kommt es bei der Kommunikation von verschiedenen Geräte via CEC manchmal dazu, dass aus unerfindlichen Gründen ein TV-Einschaltbefehl gesendet wird. Leider konnten wir das Problem in unseren Tests trotz verschiedener Einstellungen unter System/Eingabe/Peripheriegeräte/CEC Adapter nicht beheben.

Nach dem ersten Einschalten des Raspis schließen Sie die Ersteinrichtung ab und verbinden ihn unter anderem mit Ihrem Netzwerk. Via WLAN war eine flüssige 4K-HDR-Wiedergabe nicht möglich und Kodi warnte vor einer „zu langsamen Quelle“. Deswegen wählten wir eine LAN-Verbindung – und erlebten tatsächlich keinerlei Aussetzer beim Abspielen von 4K-Filmen von einem NAS.

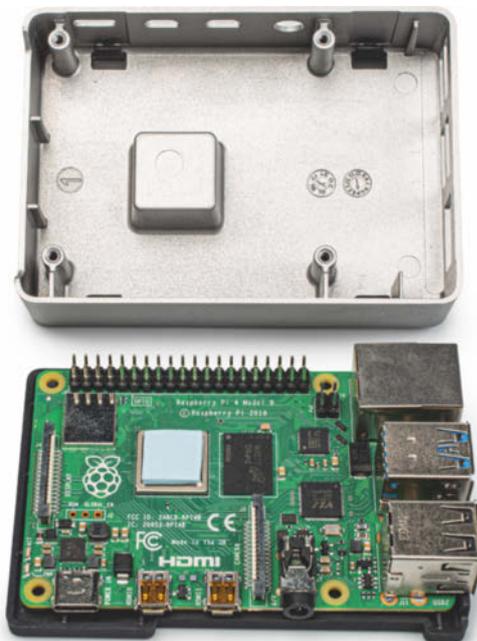
Bild und Ton

Damit die Bedienung im Kodi-Menü so flüssig wie möglich von der Hand geht, stellen Sie unter Einstellungen/System die Auflösung auf 1920 x 1080 Pixel und die Bildwiederholrate auf 60 Bilder pro Sekunde ein. So muss der Raspi weniger Bildpunkte berechnen, was der Performance zugutekommt. Angst vor unscharfen Filmen müssen Sie wegen der Einstellungen nicht haben: Sie betreffen nur die Darstellung der Benutzeroberfläche, die Filme selbst laufen in 4K-Auflösung.

Damit Kinofilme nicht ruckeln, aktivieren Sie in den Einstellungen ganz unten durch mehrmaliges Klicken auf das Zahnradchen die Expertenansicht. Nun wählen Sie unter „Player“ den Punkt „Bildwiederholrate anpassen immer“ aus. Dadurch passt Kodi automatisch die Ausgabe der Bildwiederholrate an die Fähigkeiten der HDMI-Ausgänge an, wodurch beispielsweise Filme von einer Blu-ray mit nativen 23,976 Bildern pro Sekunde laufen.

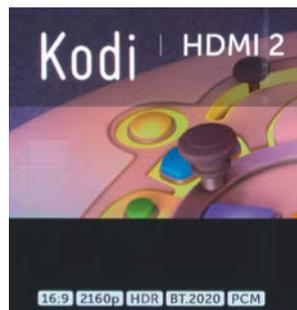
Für eine optimale Mehrkanaltonwiedergabe navigieren Sie in den Einstellungen zu „Audio“ und aktivieren die „Passthrough“-Option. Nun wählen Sie alle von Ihrem AV-Receiver unterstützten Surround-Sound-Formate aus. Selbst ältere AV-Receiver sollten

Das Flirc Case kühlt den Raspi passiv und somit komplett lautlos. Dafür verteilt ein auf dem SoC anliegender aus dem Deckel wachsender Zapfen die Abwärme auf das gesamte Alugehäuse, welches die Wärme an die Umgebung abgibt. Das Gehäuse gibt es auch in einer Kodi-Edition mit Logo auf dem Deckel.



Was rauskommt, kommt auch an

Dem Statusmonitor von Kodi zufolge (links) läuft 4K-Material mit 3840 × 1608 Bildpunkten (reduzierte vertikale Auflösung wegen 21:9-Bildseitenverhältnis) und 23,976 Bildern pro Sekunde. Aufgestockt um die Bildpunkte der schwarzen Balken gibt der Raspi 3840 × 2160 Pixel aus. Einen Hinweis auf HDR zeigt Kodi derzeit nicht an. Bei unserem OLED-Fernseher von LG kommt laut Anzeige (rechts) die 2160p-Auflösung in 16:9 nebst HDR und erweitertem BT.2020-Farbraum an.



alle aufgeführten Formate wie Dolby True HD beherrschen. Erst mit diesen Einstellungen leitet Kodi den Mehrkanalton korrekt über HDMI an einen Heimkinoverstärker.

So werden auch die 3D-Sound-Formate Dolby Atmos und DTS:X mit zusätzlichen Informationen für Höhenkanäle an AV-Anlagen weitergeleitet. Kodi unterstützt dabei als Grundcodec sowohl das komprimierte Format Dolby Digital Plus, das man von Videostreamingdiensten kennt, als auch das angesprochene verlustfreie Dolby TrueHD, das Blu-ray Discs und Ultra HD Blu-rays nutzen.

Bei unseren Versuchen kam es vor, dass beim Wechsel des AV-Receiver trotz aktiviertem Passthrough plötzlich Mehrkanal-PCM-Ton statt des jeweiligen Audio-Codecs ausgegeben wurde. Hier mussten wir aber nur einmal Passthrough in den Einstellungen deaktivieren und danach wieder aktivieren. Im Anschluss lief die Ausgabe einwandfrei.

Film ab!

Wenn Sie einen 4K-Film mit HDR10 starten, sollte der Fernseher automatisch in den HDR-Modus schal-

ten. So war es bei unserem OLED-Fernseher von LG aus dem Jahr 2019 der Fall. Während der Wiedergabe konnten wir keine Probleme feststellen und unsere Testfilme liefen alle problemlos. Kodi gibt nach unserer Erfahrung alle Farben und Kontraste korrekt aus. Eine Gegenprobe mit dem gleichen 4K-HDR-Film in einem Streaming-Portal bestätigte diese Beobachtung.

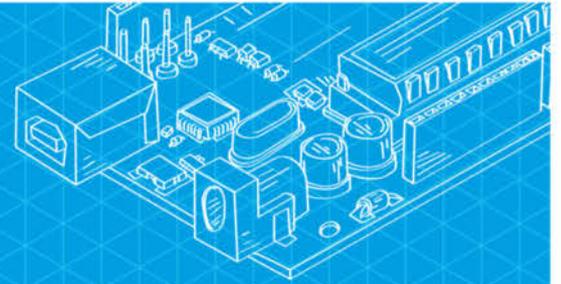
Bei der 4K-HDR-Wiedergabe pendelte sich die Temperatur des Raspis ohne Kühler bei rund 60 Grad Celsius ein. Das ist ein vergleichsweise hoher, in der Praxis aber unbedenklicher Wert: Der Einplatinen-Computer drosselt seine Leistung erst ab 80 Grad Celsius. Im Flirc Case, einem schicken kleinen Gehäuse, das auch als Kühlkörper dient, erreichte der ARM-Chip nur 53 Grad Celsius.

Der einzige Wermutstropfen ist, dass das WLAN-Modul im Kleinstcomputer zu schwach ist, weshalb man 4K-HDR-Filme nur via LAN-Kabel unterbrechungsfrei schauen kann. Insgesamt avanciert der Raspi 4 mit HDR-Unterstützung nichtsdestotrotz zum ausgewachsenen Heimkino-Player, der kaum noch Wünsche offen lässt – zumal auch 3D-Sound voll nutzbar ist. (des) 

LibreElec-Version mit HDR

ct.de/wk5b

Make:



DAS KANNST DU AUCH!



GRATIS!



2x Make testen und über 9 € sparen!

Ihre Vorteile:

- ✓ **GRATIS dazu:** Make: Tasse
- ✓ Jetzt auch im Browser lesen!
- ✓ Zugriff auf Online-Artikel-Archiv*
- ✓ Zusätzlich digital über iOS oder Android lesen

Für nur 16,10 € statt 25,80 €

*Für die Laufzeit des Angebotes.

Jetzt bestellen: make-magazin.de/miniabo



**WIR MACHEN
KEINE WERBUNG.
WIR MACHEN EUCH
EIN ANGEBOT.**

ct

ct.de/angebot

Jetzt gleich bestellen:

 ct.de/angebot

 +49 541/80 009 120

 leserservice@heise.de

ICH KAUF MIR DIE c't NICHT. ICH ABONNIER SIE.

Ich möchte c't 3 Monate lang mit 35 % Neukunden-Rabatt testen.
Ich lese 6 Ausgaben als Heft oder digital in der App, als PDF oder direkt im Browser.

**Als Willkommensgeschenk erhalte ich eine Prämie nach Wahl,
z. B. einen RC-Quadrocopter.**

