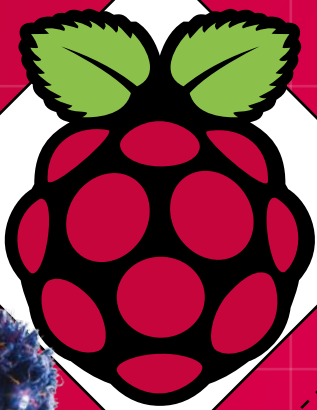
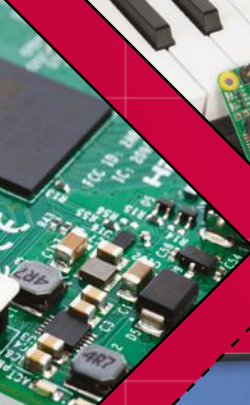
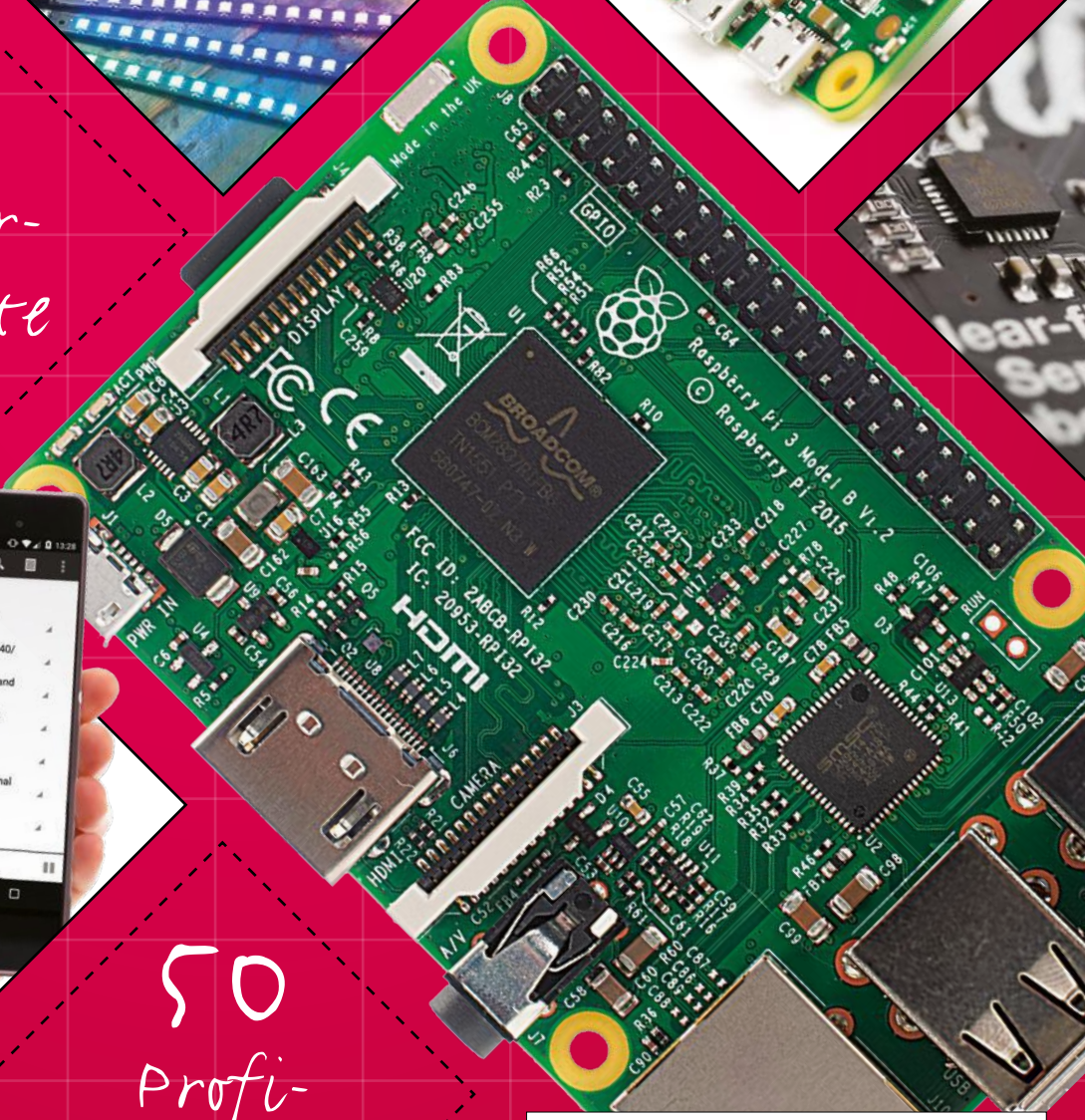
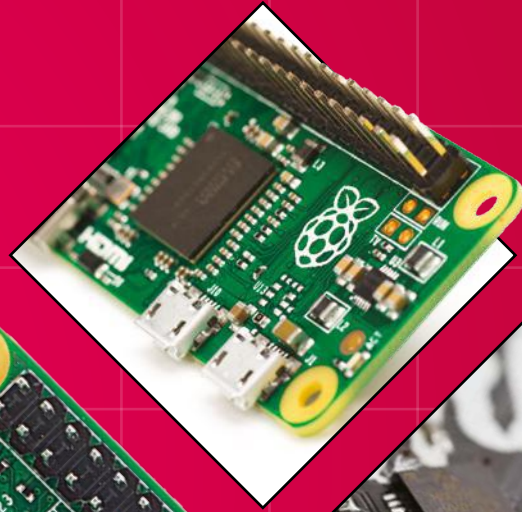


Raspberry Pi

Tipps, Tricks & Hacks



25
Maker-
Projekte



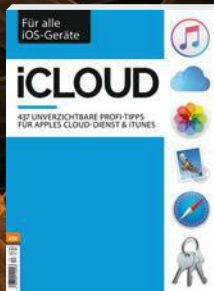
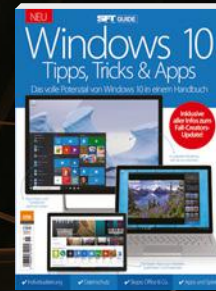
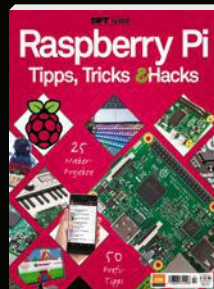
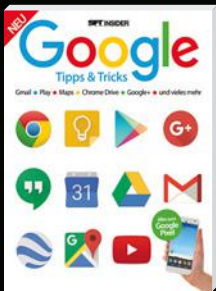
50
Profi-
Tipps

CMG

Computec Media Group

EDITION

DIE AKTUELLEN PREMIUM-BOOKKAZINES IM ÜBERBLICK



Bequem online bestellen:
shop.computec.de/edition



Oder einfach digital lesen:
epaper.computec.de



Raspberry Pi

Tipps, Tricks & Hacks

Stellen Sie sich vor, Anfang der 2000er-Jahre hätte Ihnen jemand erzählt, dass es gut ein Jahrzehnt später einen voll funktionsfähigen PC geben würde, der so groß ist wie eine Kreditkarte, individuell angepasst werden kann, mit dem Sie Spiele spielen, Roboter steuern und die Temperatur und Luftfeuchtigkeit messen können, und das alles zu einem Preis von etwa 35 Euro – hätten Sie es geglaubt?

Nun, inzwischen ist all das Realität – in Gestalt des Raspberry Pi, der für Programmierer (auf jedem Wissensniveau und in jedem Alter) ebenso interessant ist wie für Bastler, Tüftler und sogar Cosplayer (die damit ihre Gewandungen elektrifizieren können). Der durchschlagende Erfolg des Mini-Rechners spiegelt sich in mittlerweile 14 Millionen verkauften Exemplaren – auch das ist eine beinahe unglaubliche Zahl.

In diesem Heft haben wir eine Auswahl spannender Projekte aus unterschiedlichen Bereichen zusammengetragen. Konstruieren Sie eine automatische Kamera für Aufnahmen der nächtlichen Tierwelt, stellen Sie ein synchronisiertes Netzwerk-Audiosystem zusammen oder steuern Sie Ihre Zimmerbeleuchtung – alles mit dem Raspberry Pi und ein bisschen Zubehör.

Inhalt

08 25 Maker-Projekte für Pi 3

- Dritte Generation: Die Neuerungen des Raspberry Pi 3
- 3D-Drucker mit dem Pi steuern
- WLAN-Hotspot einrichten
- Fotofalle in der Natur
- Arcade-Automat bauen
- NERF-Waffe als Fernbedienung
- Smart-Home-Anwendungen

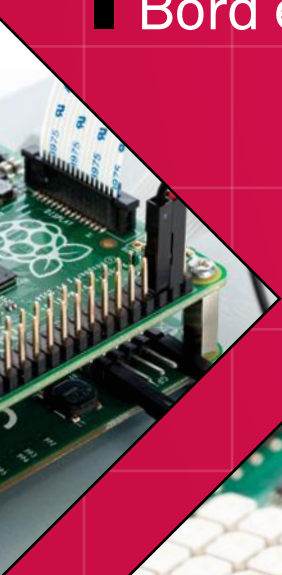
Tipps

18 So richten Sie den Pi Zero erfolgreich ein

- Welches Zubehör brauchen Sie?
- Richtig löten – so geht's
- WLAN konfigurieren
- Fernzugriff auf den Pi Zero
- VNC-Server einrichten

- 22 Entwickeln mit Python
- 30 Der neue Pi-Desktop
- 34 Multitasking mit dem Pi
- 36 Wie der Raspberry Pi auf Ihr Baby aufpasst
- 40 RISC OS als Betriebssystem für den Raspberry Pi
- 42 Die Grundlagen des Sense HAT (Teil 1)
- 46 Die Grundlagen des Sense HAT (Teil 2)
- 50 Umweltwissenschaft mit dem Sensly HAT
- 52 Der Pi als Synthesizer
- 58 Vernetzte Sensoranzeige

Ein Roboter auf Basis des Raspberry Pi, das ist schon was Feines. Aber einer mit Bluetooth und WLAN an Bord erst mal!



42



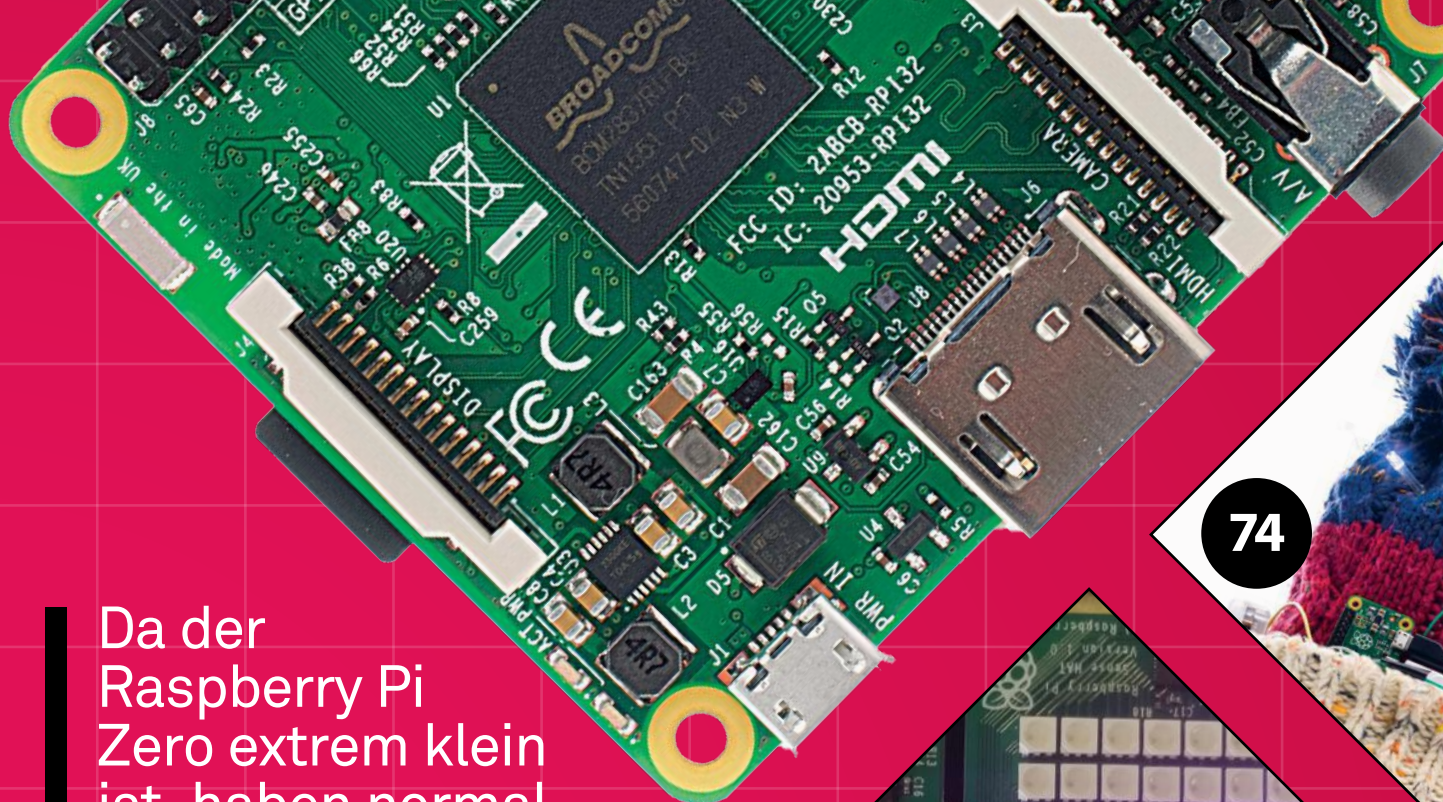
22

52



58





Da der Raspberry Pi Zero extrem klein ist, haben normal-große USB- und HDMI-Anschlüsse an ihm keinen Platz gefunden.

Tricks

64 Verbindungs-Geheimnisse

- 74 Wie Sie ein interaktives Wearable bauen
- 78 Der Pi als Minecraft-Konsole
- 84 Ein Minesweeper-Spiel mit Minecraft erstellen
- 88 Streamen Sie Internet-TV über den Raspberry Pi
- 92 Streamen Sie Internet-TV über den Pi
- 94 Web-Traffic mit Pi-Tor-Router anonymisieren
- 98 Mit Fritzing eigene Schaltpläne erstellen
- 102 Pi 2 als Desktop-PC
- 106 Dank Pi: Sound in jedem Raum



84

Hacks

114 50 Wege, um den Pi zu hacken

- | | |
|------------------------------------|---|
| 122 Zehn geniale Pi-Upgrades | 144 Beleuchtung mit Pi und Echo |
| 128 Roboter steuern | 148 Home-Verzeichnis des Pi verschlüsseln |
| 132 Selbstfahrendes RC-Auto | 152 Nächtliche Fotos mit der Kamera Pi NoIR |
| 134 Motion Tracking per Pi | 156 Ihr Telefon mit dem Pi finden |
| 136 Erstellen Sie einen Pi-Verbund | |
| 140 VPN-Zugang per Raspberry Pi | |



74



114



88

★ KOMPETENT ★ VIELSEITIG ★ KRITISCH ★

DEUTSCHLANDS ERSTAUNLICH EHRLICHES FILMMAGAZIN



WIDESCREEN – DAS BLU-RAY-, DVD- UND KINO-MAGAZIN

AUCH DIGITAL ERHÄLTlich | WWW.WIDESCREEN-ONLINE.DE

Raspberry Pi

Tipps, Tricks & Hacks



Ein Unternehmen der MARQUARD MEDIA INTERNATIONAL AG
Verleger Jürg Marquard

Verlag	Computec Media GmbH Dr.-Mack-Straße 83, 90762 Fürth Telefon: +49 911 2872-100 Telefax: +49 911 2872-200 E-Mail: bookazines@computec.de www.computec.de
Geschäftsführer	Hans Ippisch (Vorsitzender), Rainer Rosenbusch
Redaktionsleiter (V.i.S.d.P.) Mitarbeiter dieser Ausgabe Lektorat Layoutkoordination	Lars Craemer, verantwortlich für den redaktionellen Inhalt, Adresse siehe Verlagsanschrift MDV Textdienste, Marc Brehme, Frank Neupert-Paries, Andreas Spies, Andrea Reiss Claudia Brose (Ltg.), Birgit Bauer, Stephanie Kamm Albert Kraus
Vertrieb, Abonnement Marketing Produktion	Werner Spachmüller Jeanette Haag Martin Clossmann
Head of Online Entwicklung SEO/Produktmanagement Webdesign	Christian Müller Markus Wollny (Ltg.), Aykut Arik, Ruben Engelmann, René Giering, Tobias Hartlehnert, Christian Zamora, David Turkadze Stefan Wölfel Tony von Biedenfeld, Daniel Popa
Anzeigen Verantwortlich für den Anzeigenteil	CMS Media Services GmbH, Dr.-Mack-Straße 83, 90762 Fürth Annett Heinze, Adresse siehe Verlagsanschrift
Anzeigenberatung Print Bernhard Nusser: Tel.: 0911-2872-254; bernhard.nusser@computec.de Judith Gratias-Klamm: Tel.: 0911-2872-252; judith.gratias-klamm@computec.de Alto Mair: Tel.: 0911-2872-144; alto.mair@computec.de	
Head of Digital Sales Jens-Ole Quiel: Tel.: +49 (0) 911-2872-253; jens-ole.quiel@computec.de	
Head of B2B-Sales Peter Elstner: Tel.: +49 (0) 911-2872-152; peter.elstner@computec.de	
Anzeigenberatung Online Weischer Online GmbH Elbberg 7, 22767 Hamburg Tel.: +49 40 809058-2239 Fax: +49 40 809058-3239 www.weischeronline.de info@weischer.net	
Anzeigendisposition: anzeigen@computec.de Datenübertragung: via E-Mail: anzeigen@computec.de Es gelten die Mediadaten Nr. 31, gültig ab 01/2018.	
Abonnement Die Abwicklung (Rechnungsstellung, Zahlungsabwicklung und Versand) erfolgt durch DPV Deutscher Pressevertrieb GmbH als leistender Unternehmer: Post-Adresse: Leserservice Computec, 20080 Hamburg, Deutschland	
Ansprechpartner für Reklamationen ist Ihr Computec-Team unter: Deutschland: E-Mail: computec@dpv.de, Tel.: 0911-99399098, Fax: 01805-8618002* Support: Montag 07:00–20:00 Uhr, Dienstag–Freitag 07:30–20:00 Uhr, Samstag 09:00–14:00 Uhr * (14 Cent/Min. aus dem dt. Festnetz, max. 42 Cent/Min. aus dem dt. Mobilfunk)	
Österreich, Schweiz und weitere Länder: E-Mail: computec@dpv.de, Tel.: +49-911-99399098, Fax: +49-1805-8618002 Support: Montag 07:00–20:00 Uhr, Dienstag–Freitag 07:30–20:00 Uhr, Samstag 09:00–14:00 Uhr	
Vertrieb und Einzelverkauf: DPV Deutscher Pressevertrieb GmbH, Am Sandtorkai 74, 20457 Hamburg, Internet: www.dpv.de	
Druck: LSC Communications Europe, ul. Obroncow Modlina 11, 30-733 Krakau, Polen	
COMPUTEC MEDIA ist nicht verantwortlich für die inhaltliche Richtigkeit der Anzeigen und übernimmt keinerlei Verantwortung für in Anzeigen dargestellte Produkte und Dienstleistungen. Die Veröffentlichung von Anzeigen setzt nicht die Billigung der angebotenen Produkte und Service-Leistungen durch COMPUTEC MEDIA voraus. Sollten Sie Beschwerden zu einem unserer Anzeigenkunden, seinen Produkten oder Dienstleistungen haben, möchten wir Sie bitten, uns dies schriftlich mitzuteilen. Schreiben Sie unter Angabe des Magazins, in dem die Anzeige erschienen ist, inkl. der Ausgabe und der Seitennummer an: CMS MEDIA SERVICES GmbH, Annett Heinze, Anschrift siehe oben.	
Einsendungen, Manuskripte und Programme: Mit der Einsendung von Manuskripten jeder Art gibt der Verfasser die Zustimmung zur Veröffentlichung in den von der Verlaggruppe herausgegebenen Publikationen. Urheberrecht: Alle veröffentlichten Beiträge bzw. Datenträger sind urheberrechtlich geschützt. Jegliche Reproduktion oder Nutzung bedarf der vorherigen, ausdrücklichen und schriftlichen Genehmigung des Verlags.	



Lizenz

Die Artikel in diesem Bookazine sind übersetzt oder reproduziert und unterliegen dem Copyright von Future plc, UK 2017. Inhalte wurden als Lizenz erworben. Alle Rechte vorbehalten.
„Bookazines“ ist ein eingetragenes Warenzeichen von Future Publishing Limited. Weitere Informationen über diese und andere Zeitschriften der Future plc Gruppe:
www.futureplc.com

Licence

Articles in this bookazine are published under licence from Future Publishing Limited. All rights in the licensed material belong to Future Publishing Limited and it may not be reproduced, whether in whole or in part, without the prior written consent of Future Publishing Limited. ©2017 Future Publishing Limited.
www.futureplc.com



MARQUARD MEDIA INTERNATIONAL AG

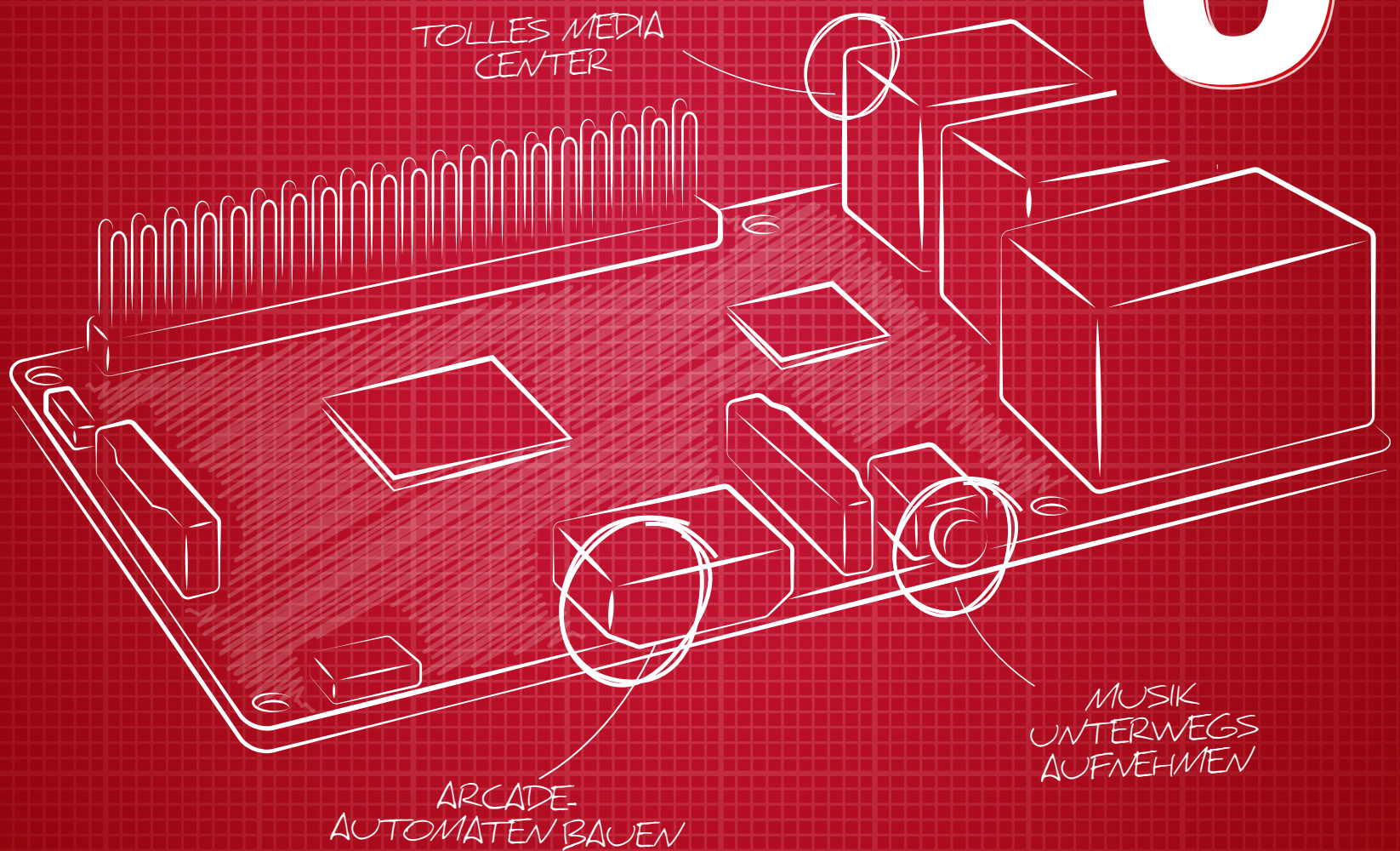
Deutschsprachige Titel: SFT, WIDESCREEN, PC GAMES, PC GAMES MMORE, PC GAMES HARDWARE, BUFFED, PLAY 4, GAMES AKTUELL, N-ZONE, XBIG GAMES, MAKING GAMES, LINUX-MAGAZIN, LINUXUSER, EASYLINUX, RASPBERRY PI GEEK

Internationale Zeitschriften: Polen: COSMOPOLITAN, JOY, SHAPE, HOT, PLAYBOY, CKM, VOYAGE, HARPER'S BAZAAR; Ungarn: JOY, SHAPE, EVA, IN STYLE, PLAYBOY, CKM, MEN'S HEALTH



25 Maker Projects für Raspberry Pi 3

25 MAKER PROJECTS FÜR RASPBERRY PI 3



DER PI 3 IST **50% SCHNELLER** ALS DER PI 2 UND HAT **WLAN UND BLUETOOTH** EINGEBAUT. WIR HABEN MIT HARDWARE-DIREKTOR **JAMES ADAMS** GESPROCHEN, UM MEHR ZU ERFAHREN. PLUS: WIR STELLEN **25 AUFREGENDE NEUE PROJEKTE VOR**, DIE DIE WIRELESS-TECHNIK DES PI 3 NUTZEN.



WIFI &
BLUETOOTH

GLEICHER
FORMFAKTOR

4X ARM CORTEX-A53,
1.2GHZ

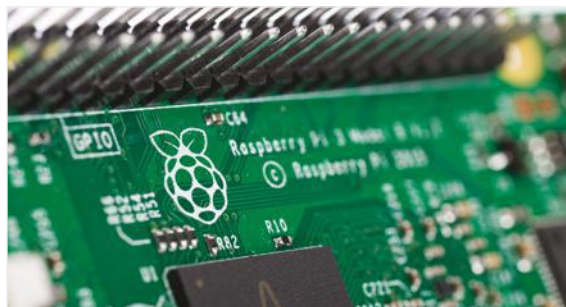
Was sind die großen Neuerungen?

Vieles, was wir im Raspberry Pi 3 finden, gab es auch schon im Pi 2. Ab dem B+ war es unserer Strategie, immer denselben Formfaktor für die Pis beizubehalten. Der B+ war tatsächlich der erste Pi an dem ich gearbeitet habe, seit ich 2013 zu Raspberry Pi kam. Die Idee war es, dass wir diesen schönen Formfaktor hatten und dass wir ihn unbedingt beibehalten wollten. Also haben wir für den Bau des Pi 3 einfach das Design des Pi 2 genommen und zusätzliche Dinge hinzugefügt.

Wir haben zunächst einen neuen Prozessor bekommen – danke an dieser Stelle an Broadcom – der ein schönes Leistungsplus bietet. Er ist dem BCM2836 sehr ähnlich. Der Cortex A7 wurde herausgenommen und der A53s eingebaut. Alles andere blieb eigentlich gleich, daher haben Sie auch die Abwärtskompatibilität. Die Taktrate der CPU wurde leicht von 250 auf 400 MHz erhöht, auf die Platine selbst bauten wir noch in die linke obere Ecke ein Bluetooth- und WLAN-Modul ein. Es gab noch einige kleinere Anpassungen aber im Grunde handelt es sich um einen Pi 2 mit WLAN und Bluetooth. Das klingt jetzt zwar trivial, aber Bluetooth und WLAN waren durchaus eine technische Herausforderung.

Wie hat sich die Geschwindigkeit mit dem BCM2837 verändert?

Wir sprechen von 50 Prozent gegenüber dem Pi 2. In der Praxis stellt sich der Pi



3 sogar noch etwas schneller dar. Es hängt hauptsächlich davon ab, welche Funktionen am meisten von der Prozessorleistung profitieren. Im alltäglichen Gebrauch erhält man 50 Prozent mehr Geschwindigkeit, resultierend aus der 33 Prozent höheren Taktfrequenz des Kernels plus der Anhebung von 900 auf 1.2 GHz. Aber die NEON-Einheit im A53 ist, glaube ich, doppelt so groß als die im A7. Also wenn Sie Prozesse haben, die von einem ARM NEON profitieren, dann arbeiten diese deutlich schneller. Ich glaube, es gibt noch einige weitere Verbesserungen im Kern, die zum Beispiel bei Dingen wie Websurfen einen größeren Unterschied machen. Es hat wirklich etwas mit der Arbeitsauslastung zu tun. Der A53 ist einfach ein viel modernerer, fetterer Kern.

Sie bekommen 64 Bit, was wir im Moment aber nicht benutzen. Aber es kamen schon Leute zu uns, die mit vorläufigen Linux-Kerneln im 64-Bit-Modus arbeiteten. Ich glaube Eben sagte öffentlich, dass wir das beobachten werden und wenn wir sehen, dass es funktioniert, machen wir entweder unser eigenes 64-Bit-OS, dass vollkommen getrennt von der 32-Bit-Version läuft, oder wir entwickeln das 32-Bit-System weiter, das ja eine schöne Abwärtskompatibilität mit allen vorangegangenen Pis hat

Links Der neue 1.2GHz BCM2837 enthält vier ARM Cortex-A53 Kerne.



James Adams ist Director of Hardware bei der Raspberry Pi Foundation, wo er die aktuelle Produktionshardware verwaltet und neue Raspberry Pi Produkte entwickelt. Er trat erstmals im Jahr 2013 ins Team ein und hat seitdem das Design des Raspberry-Pi-Modells B+, des 2B und des neuen 3B begleitet. Gerüchte sagen, das James auch Dämonenjäger und Bierbrauer ist.

Kein Funk

Es gibt Spekulationen, dass der Pi 3 einen FM-Empfänger hat, aber das ist leider nicht der Fall. „FM ist effektiv deaktiviert“, bestätigt James. „Der Wi-Fi-Chip ist ein BGA-Gerät, also hat man darunter kleine Lötstellen und diese müssen auf die Platine gelötet werden. Auf so einem Chip beträgt der Abstand zwischen ihnen 0,4 m, was man eigentlich nur auf hochmodernen Handy-Platinen findet. Diese sind mehr Hightech als das Board, das der Pi nutzt. Ein Grund, weshalb wir die Kosten so drücken konnten, ist die Tatsache, dass wir Lowtech-Boards nutzen. Man könnte also sagen, dass wir es mit ein paar pfiffigen Tricks geschafft haben, Hightech-Teile auf Lowtech-Boards zu verbauen. Aber FM wäre zu teuer.“

Rechts James ist auch Teil des Five Ninjas Teams, das einen Computer-Modul-basierten Media Player erschaffen hat.



Warum wurde die 64-Bit-Funktionalität hinzugefügt?

Zu dem Zeitpunkt, als wir einen neuen Kern in den 2837 setzten, überdachten wir alle Optionen. Alle neuen ARM-Kerne haben grundsätzlich schon 64-Bit-Unterstützung. Es ist ein nettes Feature für die Zukunft, aber uns liegt sehr viel daran, Formfaktoren und Software rückwärtskompatibel zu halten, zumindest so weit wie möglich. So gibt es derzeit keine Pläne, eine 64-Bit-Version von Raspbian zu machen. Es werden zwei völlig getrennte Betriebssysteme benötigt, die wir dann verwalten müssen. Wir müssen alles, was bei den Usern im Einsatz ist, neu kompilieren, um 64-Bit sein zu können. Das ist es eine Menge Arbeit und man muss sich ja auch noch um den Support kümmern. Technisch ist es möglich und wir werden sehen, welche Leistung 64-Bit-Linux bringt. Dann werden wir versuchen abzuschätzen, ob es sich lohnt oder nicht. Offensichtlich ist 64-Bit-ARMv8 die Art und Weise, wie ARM-Kerne am besten funktionieren und alle zukünftigen ARM-Kerne werden es unterstützen. Das ist für uns also schon ein Thema, aber wir haben noch keine Entscheidung getroffen. Mal sehen was passiert ...

Ein paar Leute, wie auch unser Rezensent Gareth Halfacree, haben bemerkt, dass der Pi 3 heißer läuft als seine Vorgänger. Sollte man jetzt vorsichtiger sein, wenn es um den Dauereinsatz geht?

Das Hitzeproblem ist etwas, das wir unter bestimmten Bedingungen reproduzieren können. Eins kann ich sagen: Beim Pi 2 verhält es sich ganz ähnlich – es dauert nur ein bisschen länger, bis er sich erwärmt. Das hat etwas mit den Veränderungen zu tun, die wir vorgenommen haben: Die Taktfrequenz hat sich erhöht und der ARM-Kern ist größer, also produziert

er auch mehr Wärme. Im Vergleich zum Pi 1 produziert der Pi 2 immer noch eine Menge Hitze, wenn man ihn so benutzt, wie Gareth ihn benutzt. Aber wird das Ihren Pi umbringen? Absolut nicht. Der 2837 SoC ist für 125° C ausgelegt und das bei einer Lebensspanne von zehn Jahren. Wir machen und also keine Sorgen, dass die Hardware plötzlich ausfällt.

Der Pi drosselt die CPU-Geschwindigkeit, wenn er zu heiß wird. Die Grenztemperatur ist auf 85° C in der Firmware eingestellt. Wir reden mal mit Gareth, wie er es geschafft hat, den Pi auf eine noch heißere Temperatur zu bringen. Es gibt ja auch leichte Temperaturgefälle quer über den Chip – der Temperatursensor befindet sich aber nur in einer Ecke des Chips. Wir sind aber auf jeden Fall nicht besorgt darüber und wir können unsere Temperaturalgorithmen aktualisieren, wenn uns Gareth seine Daten gibt.

Aber natürlich können diese Chips heiß werden, besonders dann, wenn sie stark beansprucht werden. Dann drosseln sie sich, indem sie Spannung und Taktfrequenz herunterregeln. So würde ein Pi 2 oder 3 von einem Kühlkörper profitieren, um den Grad der Drosselung zu reduzieren, wenn man den Pi kontinuierlich unter hoher Last laufen lassen würde. Auf der anderen Seite könnte man einen Pi 1 den ganzen Tag unter Volllast laufen lassen ohne dass er groß warm werden würde. Es gibt jetzt vier deutlich dickere Kerne, die mit einer viel schnelleren Rate laufen, also stießen wir an physikalische Grenzen. Ihre Handys funktioniert auf die gleiche Art und Weise – sie machen diese Performance-Sprints, wo sie den Kern so schnell wie möglich hochfahren. Ihr Telefon wird warm, und sobald es an den Punkt kommt, wo es zu heiß ist, wird die Frequenz und die Spannung heruntergefahren.



Die Herausforderung war etwas größer als sonst. Alles in allem ist es eine Menge Arbeit, HF auf etwas zu verbauen.

Aber bei rechenintensiven Standard-Aufgaben wie im Web Surfen kühlt sich der Chip auf einem Raspberry Pi ziemlich schnell ab, wenn man aufhört ihn weiter zu fordern. Das Board selbst ist also bereits ein guter Kühlkörper. Hitze wird durch die Unterseite der Löt kugeln und in die Grundebenen des Boards geleitet. Und das Board verfügt über ziemlich viel draufgelötetes Metall: USB-Verbindungsstücke und so. Es ist uns wichtig zu betonen, dass sich das Hitzeproblem nicht auf die Lebensdauer des Pis auswirken wird!

Als wir vor einer Weile nach Wi-Fi und Bluetooth gefragt haben, sagte die Foundation, dass es im Wesentlichen zu teuer wäre, diese Komponenten zu inkludieren. Was hat sich inzwischen geändert?

Wir wollten schon immer Bluetooth und WLAN auf den Pi setzen, weil es heutzutage einfach was ganz Normales ist. Die meisten Leute, die einen Laptop kaufen, haben jetzt WLAN. Verdrahtetes Ethernet wird manchmal nicht einmal auf Laptops unterstützt, wenn man einen von Apple kauft. Die offensichtliche Herausforderung liegt in der Preisgestaltung. Wir haben sehr hart daran gearbeitet, die Materialkosten so gering wie möglich zu halten. Am Ende hatten wir dann sogar Luft für ein WLAN-Modul und Broadcoms WLAN-Lösung ist wirklich toll, weil es sehr günstig ist und nur wenige externe Komponenten verwendet.

Auf der anderen Seite sind Dinge wie Funkwellen-Design, Entwicklung und Konformitätsprüfung sehr große technische Aufgaben. Sobald Ihr Produkt anfängt auszustrahlen, ste-



Links Zu den kleineren Änderungen gehört die Rückkehr zum alten Slip-Lock-MicroSD-Kartensteckplatz - kein Push-to-Eject mehr!

hen Sie vor einem großen Haufen von Konformitätsproblemen. Tatsächlich verlangt jedes Land von Ihnen, dass Sie Tests durchführen, ob Ihr Gerät mit den landesspezifischen Funkfrequenzregeln konform geht. Also mussten wir sehr eng mit Broadcom zusammenarbeiten, um den Chip in alle benötigte Testmodi für die Testhäuser einzustellen, was eine Menge Aufwand war. Und selbst das allgemeine HF-Design ist schwieriger als das Standard-Design, denn wenn Sie es mit elektrischem Übersprechen zu tun haben oder die Stromquelle in der Nähe von etwas Lärmendem ist, kann das die Funkperformance beeinflussen.



U.FL-Antenne

„Es geht nicht darum, hochfrequente Radiowellen (HF) in die Luft zu versprühen“, erklärt James uns, „Sie stecken ein Kabel in den Tester. Dafür müssen Sie irgendeine Art von Steckverbinder haben. Es gibt einen kleinen Kurzschlusswiderstand, den Sie mit dem U.FL-Anschluss verbinden und Sie können dann eine Antenne anschließen oder in unserem Fall alle Testgeräte, die wir brauchten. Allerdings würde ich sagen, wenn Sie die U.FL-Antenne benutzen wollen, tun Sie das nicht – die FCC mag es nicht. Aber theoretisch können Sie die Lötmaske abkratzen, eine U.FL anlöten und eine Antenne anschließen.“

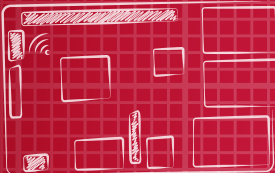


25 MAKER PROJECTS

NUTZEN SIE IHR ON-BOARD-WLAN UND -BLUETOOTH MIT DIESEN NEUEN UND AUFGERÜSTETEN PROJEKTIDEEN.

1

Wildtierkamera



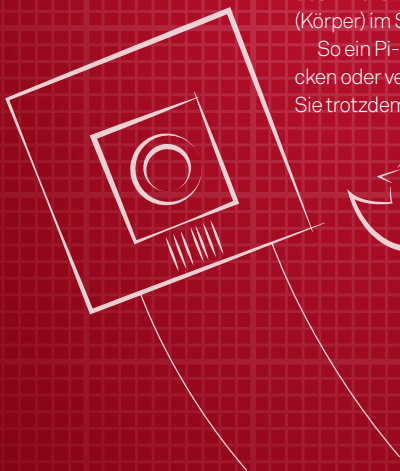
Zwar kann man dafür auch ein Kit kaufen, wie zum Beispiel jenes von Naturebytes (bit.ly/1P-dUNue), aber es ist auch relativ einfach, seine eigene Wildtierkamera zusammenzubauen.

Stellen Sie sicher, dass Sie ein wasserdichtes Gehäuse zum Schutz Ihres Pis, der portablen Stromversorgung und allen anderen elek-

trischen Komponenten vor Regenwasser zur Hand haben. Dafür gibt es 3D-Modelle (STL-Files), die Sie mit Programmen wie 123D Design leicht so anpassen können, dass Ihr Pi hineinpasst. Dann müssen Sie nur noch dafür sorgen, dass sämtliche Verbindungsstücke und das Loch, durch das die Kamera filmt, eine wasserdichte Abdeckung bekommen. Sehen Sie sich diesen Leitfaden an: bit.ly/1mgUA2X.

Sie können Ihren Pi so einstellen, dass die Kamera während bestimmter Zeiten durch das Auslösen eines Python-Skripts mit einem Cron-Job aktiviert wird – fügen Sie einfach ein Uhr-Modul über Ihren GPIO-Header hinzu, damit der Pi weiß, wie spät es ist. Wenn Ihr Pi in WLAN-Reichweite ist, kann er die Bilder automatisch hochladen oder für Sie tweeten. Wenn Sie es vorziehen, dass Ihre Kamera nur dann auslöst, wenn sich ein Tier vor der Linse befindet, richten Sie einen passiven Infrarot-Sensor (PIR) ein. Er reagiert auf Änderungen des Infrarotbildes, also wenn sich eine Wärmequelle (Körper) im Sichtbereich der Kamera bewegt.

So ein Pi-Setup lässt sich leicht in Büschen verstecken oder vergraben und dank WLAN und SSH haben Sie trotzdem jederzeit Zugriff auf Ihre Skripte.



Immer im Bilde

Verbinden Sie sich mit einem PC über Ethernet und verwenden Sie den Remote Desktop. Via WLAN erhalten Sie Zugang zum Internet.

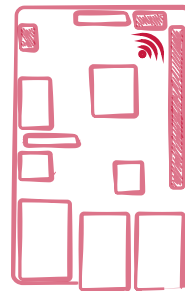
2

Wetterstation

Zubehör

- DHT22-Sensor
- BMP085-Sensor
- TGS 2600
- UVI-01-Sensor
- LDR (lichtabhängiger Widerstand)
- ADC (Analog-zu-digital-Konverter)
- Windfahne
- Windsensoren

Während ein wasserdichtes Gehäuse auch hier eine gute Idee ist, würden einige der Sensoren davon profitieren, außerhalb des Gehäuses zu sein, wie der Temperatursensor – für einige von ihnen gibt es wasserdichte Varianten! Sie könnten immer eine Kombination von Sensoren verwenden. Damit lassen sich Durchschnittswerte ermitteln und Sie haben auch einige eingebaute Redundanz. Hier geht es darum, über die GPIOs viele Sensoren anzuschließen, um die Wetterlage in Ihrem Bereich zu lesen. Mit dem WLAN und dem tragbaren Netzteil können Sie die Wetterstation in Ihrem Garten mit Ihrem Heim-Router (oder einem WLAN-Repeater) für einen Internetzugang verbinden und die gesammelten Daten auf eine Wordpress-Seite hochladen.



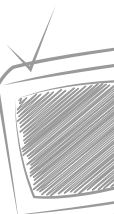
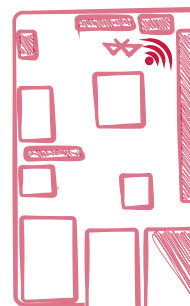
3

Roboter

Jetzt haben wir schon seit einiger Zeit Raspberry-Pi-Roboter gebaut – aber mit On-Board-Bluetooth und Wi-Fi? Für den Anfang brauchen Sie keinen Wi-Fi-Dongle, um auf Ihren Roboter zuzugreifen und irgendwelche Änderungen an dem Steuerskript vorzunehmen, was bedeutet, dass Sie die Standfläche Ihres Designs reduzieren können. Und wenn Sie einen Dongle benutzt haben, dann könnten Sie sogar Ihren Roboter herumfahren, um ihn als mobilen WLAN-Signal-Repeater zu verwenden! Durch die Einrichtung des BlueZ-Stacks können Sie Ihren Roboter auch benutzen, um Bluetooth-LE-Geräte zu steuern. Vielleicht montieren Sie ein Paar Lautsprecher auf die Rückseite des Roboters? Genauso können Sie Bluetooth-Geräte wie die Wiimote zur Bewegungssteuerung verwenden (siehe bit.ly/1RRh4Qu). Holen Sie sich ein Bluetooth-Headset und Sie können sogar sprachgesteuerte Anwendungen für Sprachbefehle nutzen!

Zubehör

- WLAN-Dongle
- Bluetooth-Lautsprecher
- Bluetooth-Headset





4

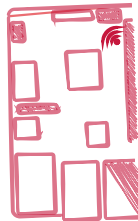
Kabellos

Investieren Sie in eine Bluetooth-Tastatur/Maus-Kombi und Sie haben alle vier USB-Ports wieder frei.

3D-Drucker

Sie sind in der Lage einen 3D-Drucker über einen Pi 3 zu steuern. Die extra Leistung macht es zudem möglich, das Zuschneiden mit dem Pi vorzunehmen. Zuerst müssen Sie sich ein entsprechendes Gerät kaufen oder Sie steigen in das RepRap-Projekt ein. RepRaps sind 3D-druckbare 3D-Drucker. Sie brauchen also einen Freund, der einen guten 3D-Drucker hat, um sich die Teile auszudrucken, die nichtdruckbaren Elemente finden Sie hier http://reprap.org/wiki/RepRapPro_Huxley. Wenn Ihr 3D-Drucker eingerichtet ist, installieren Sie das OctoPi-Image auf Ihre SD-Karte und verwenden dann

die enthaltene Software wie OctoPrint und CuraEngine, um Ihren G-Code zu bearbeiten und das Modell zurechtzuschneiden – all dies machen Sie direkt über WLAN. Sie können auch eine Kamera in Ihrem Drucker platzieren, um Zeitrafferaufnahmen Ihres Drucks zu machen!



Zubehör

- 3D-Drucker
- Kameramodul
- Externe Festplatte

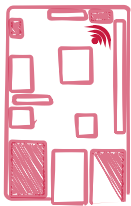
5

WLAN-Hotspot

Richten Sie Ihren Pi als Wireless Access Point für andere Geräte ein, der IP-Adressen zuweist. Verbinden Sie ihn mit dem Internet über Ethernet und er dient

Zubehör

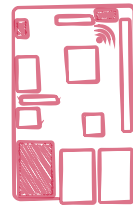
- Tragbares Akku-Pack
 - WLAN-Dongle
- als Brücke für die Geräte, die sich über das WLAN-Modul des Pis verbinden. Wenn Sie versuchen, eine Verbindung in Ihren Garten zu bekommen, verwenden Sie ein tragbares Netzteil und stecken Sie einen USB-WLAN-Dongle ein. Dann nutzen Sie einfach das On-Board-WLAN-Modul.



6

Super-Server

Der Pi 3 kann mit riesigen Mengen an Seitenaufrufen Schritt halten – die Pi-Foundation hat während der Pi 3-Einführung ein Experiment durchgeführt, um zu sehen, ob er mit einer großen Besucherzahl (<http://bit.ly/22mCjRR>) zurechtkommt. Über 12 Stunden lang bediente er 1,5 Millionen Anfragen! Also, auch wenn Ihre Website riesig ist, können Sie Ihren Pi als Server nutzen – und dank WLAN lässt er sich von überall aus ansteuern!



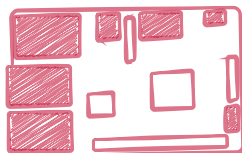
7

Mega-Media-Centre

Schließen Sie Monitor, Lautsprecher, Tastatur und Maus an, dann installieren Sie OSMC und Sie haben einen Heimkino-PC. Konfigurieren Sie eine Bluetooth-Fernbedienung oder eine Fernbedienungs-App auf Ihrem Smartphone oder Tablet.

Zubehör

- Heimkino-Zubehör
- Bluetooth-Fernbedienung
- Externe Festplatte



Verbinden Sie eine externe Festplatte mit Ihrem Raspberry Pi, auf der all Ihre Lieblingsmusik, Filme und TV-Serien gespeichert sind. Richten Sie Ihr OSMC-Mediencenter ein, indem Sie den auf Ihrer externen Festplatte gespeicherten Inhalt hinzufügen. Stellen Sie Ihren Pi als Medienserver für Ihre mobilen Geräte ein. Schließlich durchsuchen Sie den OSMC App Store nach Apps wie einen Torrent-Client und einen TV-Tuner.

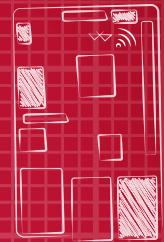
Praktische Gadgets

8

PiBook



Passen Sie ein Display in eine 3D-gedruckte Laptop-Tasche ein, zusammen mit Tastatur und Trackpad. Für Audio verwenden Sie kleine Bluetooth-Lautsprecher. Verbinden Sie sich mit dem Internet über WLAN. Wechseln Sie zu einem tragbaren Netzteil, wenn Sie unterwegs sind. Installieren Sie Ubuntu MATE.

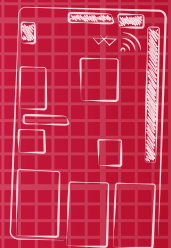


9

PiPad



Montieren Sie einen Pi auf die Rückseite eines Touchscreen-Displays und schließen Sie alles an. Verwenden Sie eine Standardstromversorgung und, wenn Sie ungebunden sein wollen, wechseln Sie zu einem tragbaren Akku-Pack. Koppeln Sie noch eine Bluetooth-Tastatur an.



10

Smart TV

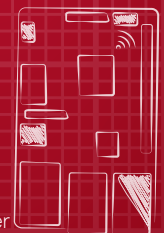
Installieren Sie Kodi und befestigen Sie Ihren Pi auf der Rückseite des Fernsehgeräts. Verbinden Sie Ihren Fernseher mit dem Pi über HDMI. Schließen Sie die externe Festplatte an, um Videos hinzuzufügen und rufen Sie eine Streaming-Webseite auf. Konfigurieren Sie noch eine Bluetooth-Fernbedienung.



11

Lautsprecher

Verbinden Sie Ihre Lautsprecher über einen Class-D-Verstärker mit dem Pi. Legen Sie das Setup in ein Gehäuse, so dass Platz für Netzkabel oder Power-Pack ist. Richten Sie einen Logitech-Media Server auf Ihrem Pi ein und steuern Sie die Wiedergabe mit der Squeezer App. Bluetooth-Lautsprecher-Besitzer können BlueZ verwenden.



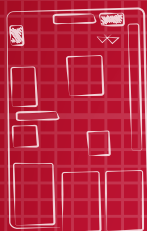


Smart-Home Technologie

Rufen Sie Ihren Pi an

Sie müssen etwas checken, aber Ihr Pi steckt unzugänglich in einem Projekt? Nutzen Sie JuiceSSH!

12 Lichtspiele



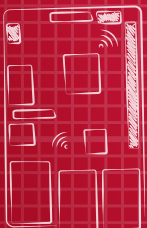
Viele Bluetooth-LEDs können dazu gebracht werden, mit BlueZ zusammenzuarbeiten – schauen Sie hier, ob Sie eine Bibliothek für Ihr Modell finden: bit.ly/1U-jxGq8. Sie können auch Bluetooth-Heimautomations-Schalter verwenden. Schließen Sie hier Ihre nicht smarten Lichter an und steuern Sie sie über den Schalter.

13 Pflanzenmonitor



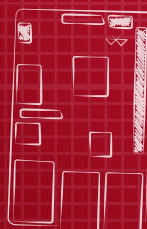
Verwenden Sie einen SHT10-Boardmonitor, um die Temperatur und Feuchtigkeit der Erde um der Wurzeln zu erfassen, dann laden Sie diese Daten auf einen Webserver auf Ihrem Pi, den Sie über Ihr Handy checken können. Mit Akku-Pack und wasserdichtem Gehäuse können Sie den Pi auch im Garten lassen.

14 Temperaturkontrolle

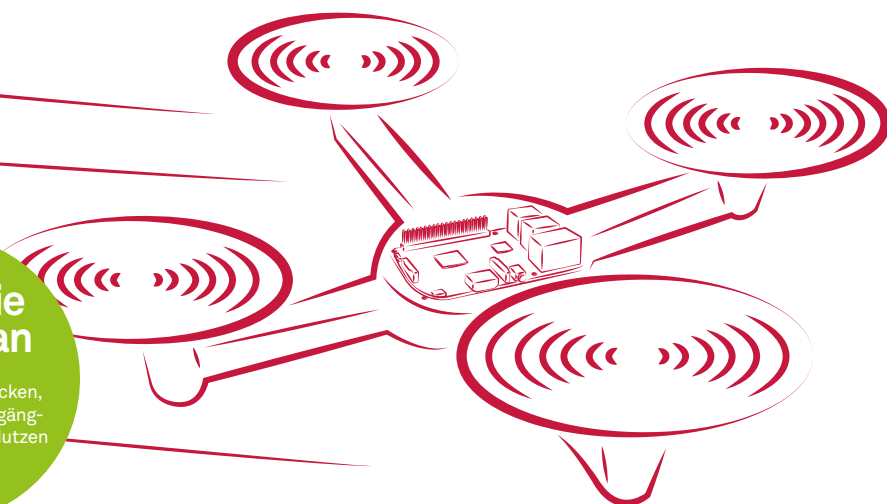


Verbinden Sie Sensoren wie den DHT22 mit Ihrem Pi, stecken Sie alles in ein Gehäuse, dann laufen Sie damit ums Haus. Laden Sie alle Daten in eine Datenbank auf Ihrem Pi-Webserver, um darauf von einer Web-Seite aus zugreifen zu können. Finden Sie heraus, wo die heißen und kalten Stellen in Ihrem Haus sind.

15 RC-Steckdosen



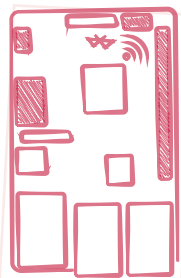
WeMo-Schalter werden direkt in Ihre Wandsteckdose gestöpselt. Sie können Ihre nicht smarten Geräte in diese Dosen stecken und sie per Bluetooth-Befehl von Ihrem Pi aus über BlueZ ein- und ausschalten. Sie können dies dann mit Cronjobs oder einem Mikrophon und PocketSphinx kombinieren.



16 Drone

Zubehör

- Droner-Kit
- PXFmini Autopilot Shield



Es gibt einige großartige Ressourcen, wenn Sie Ihre eigene Drone bauen wollen. Auf Seiten wie diydrones.com und instructables.com finden Sie alles, was Sie wissen müssen, inklusive aller Motoren und 3D-Modelle für ein Gehäuse. Sie können stattdessen auch ein komplettes Dronen-Kit kaufen. Um die Drone zu betreiben und sie zu fliegen, eignet sich Ihr Pi 3 mit dem neuen PXFmini Autopilot Shield. Dabei handelt es sich um ein exzellentes Gerät, das für den Pi Zero entwickelt wurde. Aber es ist auch 100 Prozent kompatibel mit dem Pi 3 und seine CPU gibt einer Pi 2- oder Pi Zero-Drone noch mal einen extra Performance-Schub. Das PXFmini beinhaltet ein Gyroskop sowie Kompass, Beschleunigungsmesser, Magnetometer, Druck- und Temperatursensor plus ein ADC. Und dank WLAN und Bluetooth brauchen Sie keine extra Fernsteuerung.

17

Vorort-Aufnahme-Studio

Zubehör

- Cirrus Logic Audio Card
- Mikrofone
- Instrumente

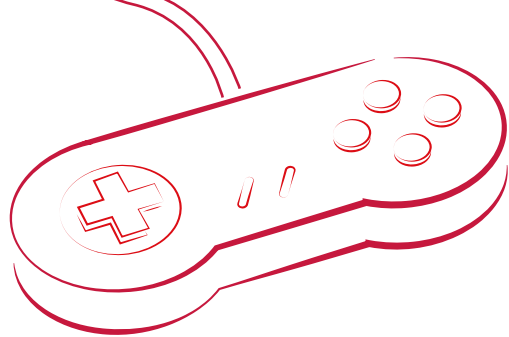
Add-ons wie die Wolfson Audio Card und die Pi 3-kompatible Cirrus Logic Audio Card ermöglichen es Ihnen, Audio viel besser auf Ihrem Pi zu handhaben. Mit einer Audiokarte können Sie die Vorteile von winzigen On-Board-Mikrofonen nutzen oder eigene Audio-Hardware anschließen. Es gibt auch Buchsen für Stereo-Line-Eingänge, in die Sie Ihre Instrumente stecken können, um hochwertige Audioaufnahmen zu machen. Sie können auch externe Verstärker oder Lautsprecher anschließen. Die Leistung des Pi 3 ermöglicht es Ihnen, dass Sie die Aufzeichnung ohne Verzögerungen verarbeiten können. Mit einem Programm wie Audacity lassen sich dann Aufnahmen dann bearbeiten.

18

Live-Audio-Performance

Zubehör

Richten Sie Ihre Lautsprecher mit dem Class-D-Verstärker ein und für die Audioausgabe ist gesorgt. Verbinden Sie Ihr Instrument über die USB-Soundkarte und richten Sie dann den Guitarix ein: Dies ist ein virtueller Gitarrenverstärker, der das pure Signal abgreift, einen Effekt hinzufügt und dann ein verarbeitetes Stereosignal an Ihre Lautsprecher weitergeben kann. Wenn Sie Effektivoreinstellungen vornehmen, können Sie dafür sorgen, dass sie von Ihren GPIO-verbundenen Tasten oder einem Tastenfeld-Add-On ausgelöst werden. Mit dem Pi lässt sich die Audioverarbeitung weiter anpassen. Es ist sogar möglich, mit dem Pi aufgenommene Tracks automatisch auf Ihre Dropbox über WLAN hochzuladen.

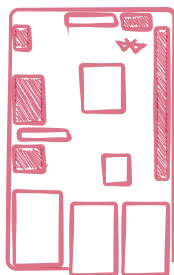


19 Arcade-Automat

Zubehör

- Schrank (z. B. selbstgebast)
- Monitor
- Lautsprecher
- Joystick & Tasten
- Gamepads

Nehmen Sie einen HDMI-Monitor (oder DVI mit einem Adapter) und bauen Sie ihn zusammen mit Lautsprechern in ein Gehäuse. Setzen Sie Joystick und die Tasten ein und verdrahten Sie sie dann mit den GPIO-Pins. Installieren Sie RetroPie auf dem Pi, fügen Sie die ROMs hinzu, die Sie besitzen, konfigurieren Sie den Joystick und die Tasten, dann konfigurieren Sie alle Bluetooth- oder USB-Gamepads, die Sie für zusätzliche Spieler haben. Setzen Sie den Pi in einen Schrank und schon haben Sie einen Retro-Arcade-Automaten!



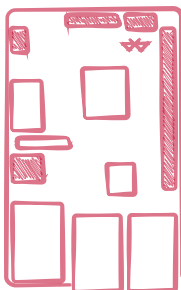
20

Arcade-Touch-Tisch

Zubehör

- Kaffeetisch (z. B. selbstgebast)
- 7-Zoll Touchscreen
- Lautsprecher
- 6-8 kapazitive Touch-Sensoren (Pi Desk)

Verkabeln Sie einen Touchscreen über die GPIOs und konfigurieren Sie das Display für die Touch-Eingabe. Bauen Sie es in die Oberfläche eines Tisches ein. Fügen Sie kapazitive Touch-Sensoren in einem D-Pad-Tasten-Layout hinzu und installieren Sie Lautsprecher. Nun folgen Sie der Anleitung von Frederick Vandenbosch für die Oberfläche (siehe bit.ly/1lwEMJ8). Legen Sie eine Plexifolie über Ihre kapazitiven Touch-Sensoren, fügen Sie ein paar Schichten Papier hinzu, um die Komponenten zu verdecken, dann eine weitere Schicht Folie. Die kapazitiven Touch-Sensoren sollten empfindlich genug sein, um sie auszulösen, wenn Sie die Hände auf die Oberfläche legen.



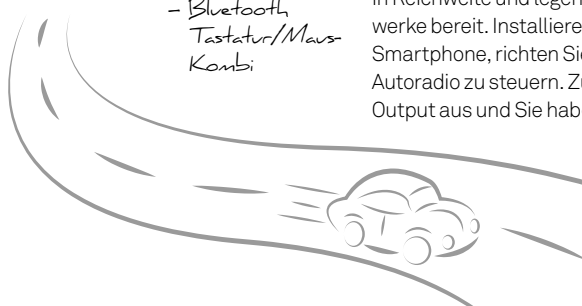
21

Car Entertainment

Zubehör

- Touchscreen Display
- Rücksitz-Monitore
- Lautsprecher
- Bluetooth Tastatur/Maus Kombi

Richten Sie ein Kodi Media Center auf Ihrem Raspberry Pi ein und fügen Sie ein Touchscreen-Display hinzu, um die Bedienung während des Fahrens zu ermöglichen. Installieren Sie das Gerät in Ihrem Auto im Radio-Steckplatz und fügen Sie verdrahtete oder drahtlose Lautsprecher hinzu (sie haben ja On-Board-WLAN). Sehen Sie sich bit.ly/1jdL8rQ für eine kleine Inspiration an. Platzieren Sie eine Bluetooth-Tastatur in Reichweite und legen Sie Ihre mit Medien bespielten Laufwerke bereit. Installieren Sie eine Kodi Remote App auf Ihrem Smartphone, richten Sie es ein und verwenden Sie es, um das Autoradio zu steuern. Zu guter Letzt: Führen Sie Navit mit TTS Output aus und Sie haben ein echtes Navigationssystem.



Praktische Gadgets

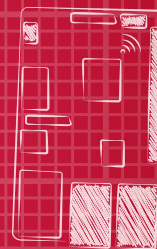
22 Smartes Spielzeug

Nehmen Sie ein mittelgroßes Spielzeug und machen Sie ein Loch hinein, damit Sie den Pi darin verstecken können. Jetzt verdrahten Sie oben Ihre LEDs, das Kameramodul und die Minilautsprecher, dann versiegeln Sie das Spielzeug oben wieder. Mit dem Pi können Sie auch ein Interface programmieren, um z. B. die Farbe der LEDs zu steuern.



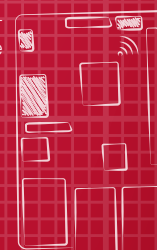
23 Nerf Shuffle

Schreiben Sie ein einfaches Skript, um Ihre Musik-Playlist im Shuffle-Modus wiederzugeben, was in diesem Fall mit einem Schuss aus einer Nerf-Pistole passieren kann. Verdrahten Sie einen mittleren Vibrationssensor, so dass er einen Treffer auf Ihrem Ziel erkennen kann, dann übergeben Sie diese Eingabe an Ihr Skript. Feuer frei!



24 Drahtloser Projektor

Holen Sie sich ein Gerät wie den Brookstone Pocket Projector und verbinden Sie ihn mit Ihrem Pi, stecken Sie das Setup in ein Gehäuse und projizieren Sie die Anzeige irgendwo hin. Wenn die Auflösung nicht zufriedenstellend ist, kann der Pi 3 eine VNC-Verbindung zu Ihrem Computer über Wi-Fi herstellen.



25 Minecraft-Zauberstab

Verbinden Sie Ihren Pi mit der Wiimote via Bluetooth und konfigurieren Sie sie dann mit dem CWiid Python Modul (bit.ly/1Wtzb2Z). Mit dem Wiimote-Setup können Sie jetzt die Wiimote-Buttons benutzen, um vorgefertigte Python-Skripts auszulösen, um Ihre Minecraft-Welt zu verändern.



Tipps

Lernen Sie den Raspberry Pi Zero kennen. Dieser Rechner ist sogar noch kleiner als der gewöhnliche Pi.

18 Pi Zero einrichten

- Welches Zubehör brauchen Sie?
- Richtig löten – so geht's
- WLAN konfigurieren
- Fernzugriff auf den Pi Zero
- VNC-Server einrichten

18

22 Entwickeln mit Python

30 Der neue Pi-Desktop

34 Multitasking mit dem Pi

36 Wie der Raspberry Pi auf Ihr Baby aufpasst

40 RISC OS als Betriebssystem für den Raspberry Pi

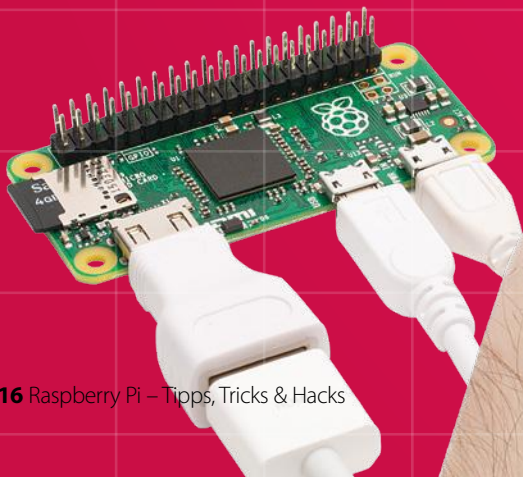
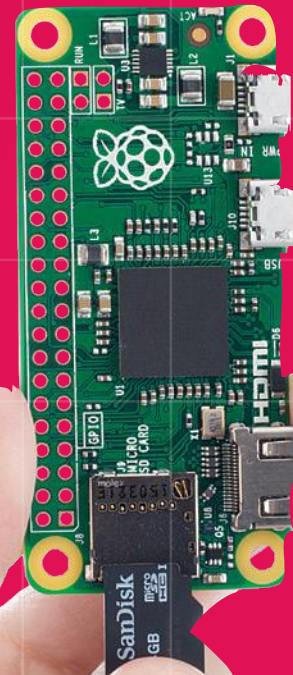
42 Die Grundlagen des Sense HAT (Teil 1)

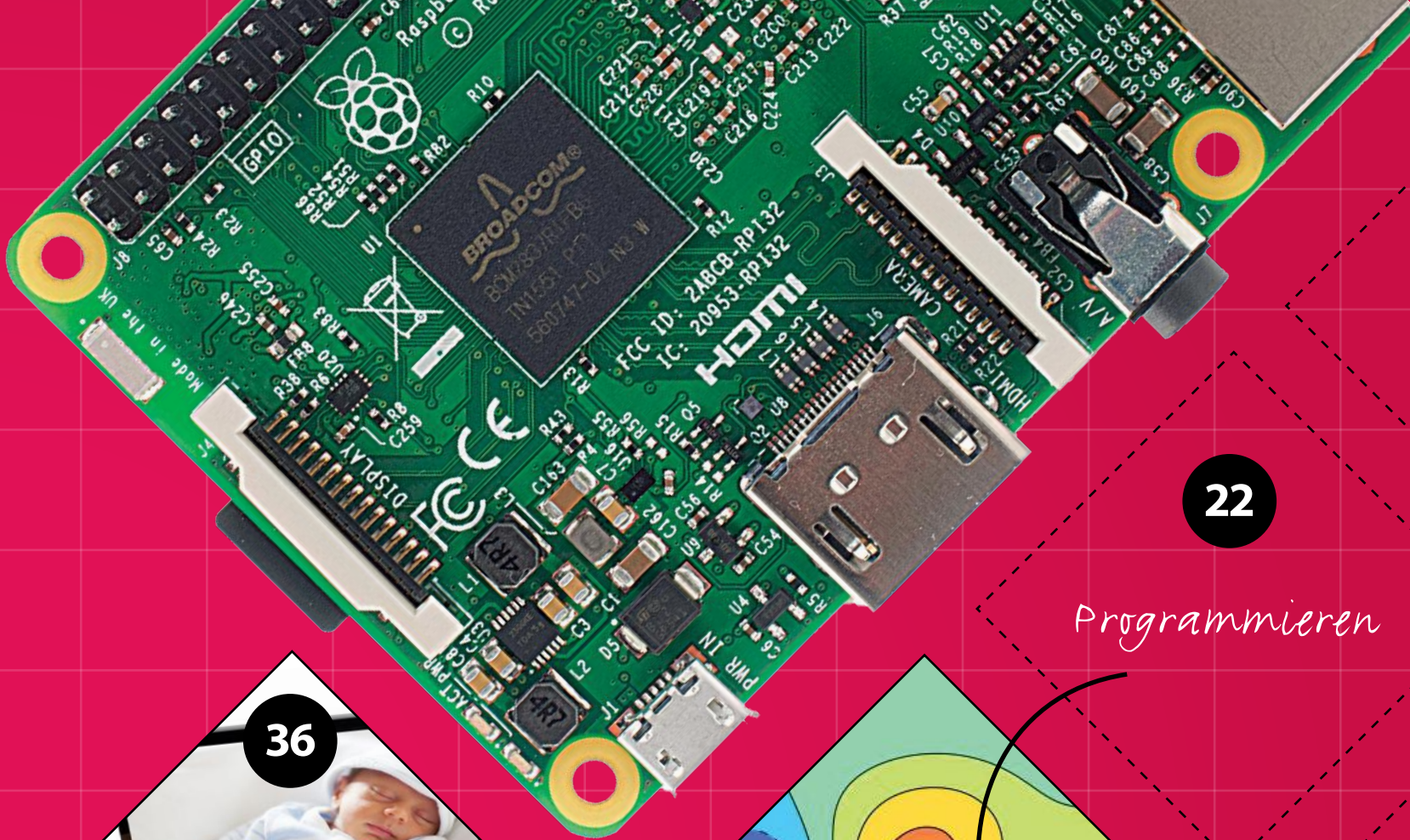
46 Die Grundlagen des Sense HAT (Teil 2)

50 Umweltwissenschaft mit dem Sensly HAT

52 Der Pi als Synthesizer

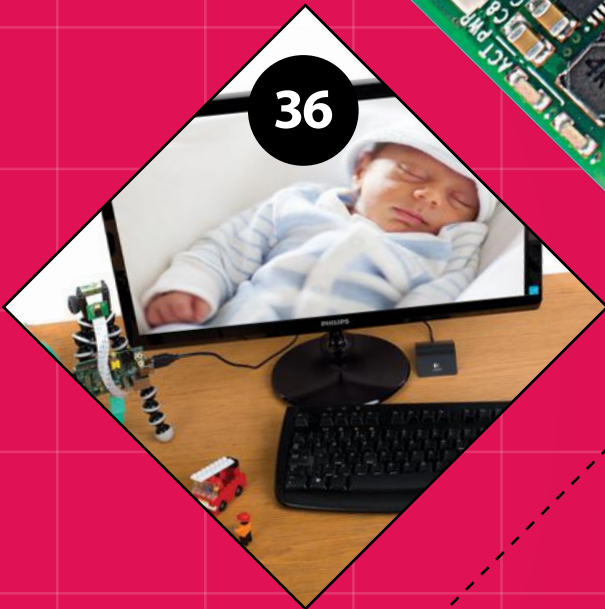
58 Vernetzte Sensoranzeige



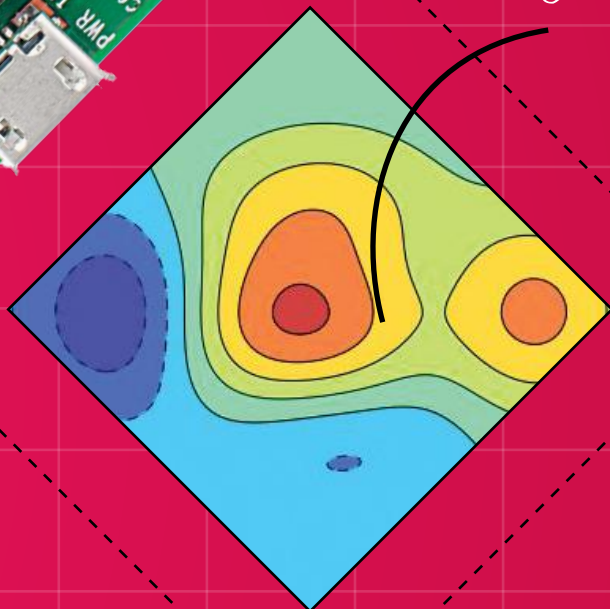


22

Programmieren



36

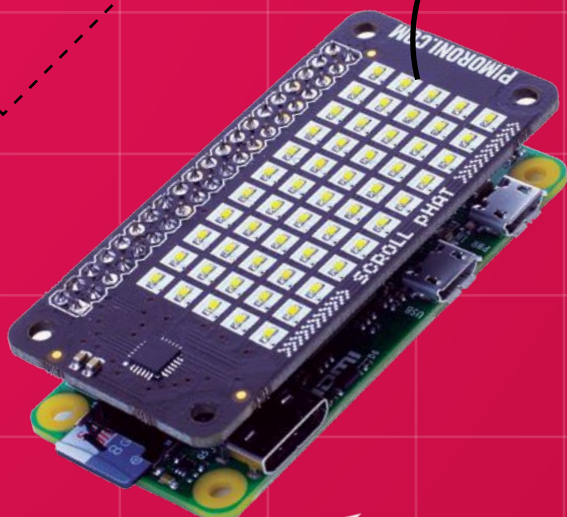
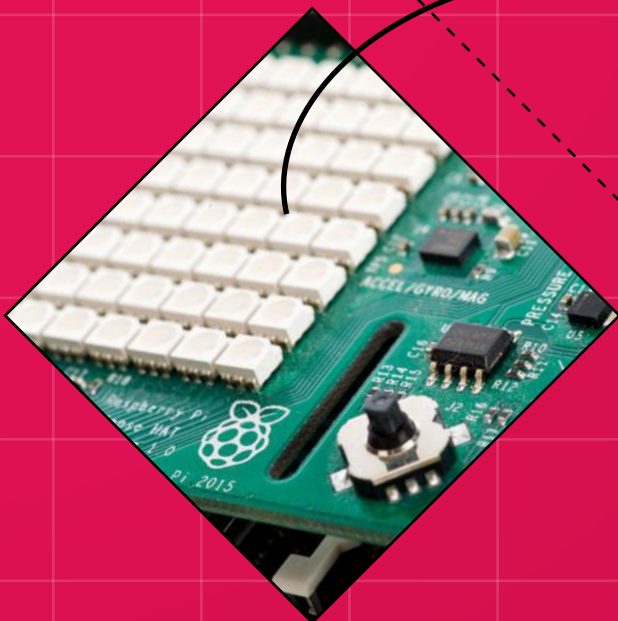


42

sense HAT

50

sensly HAT



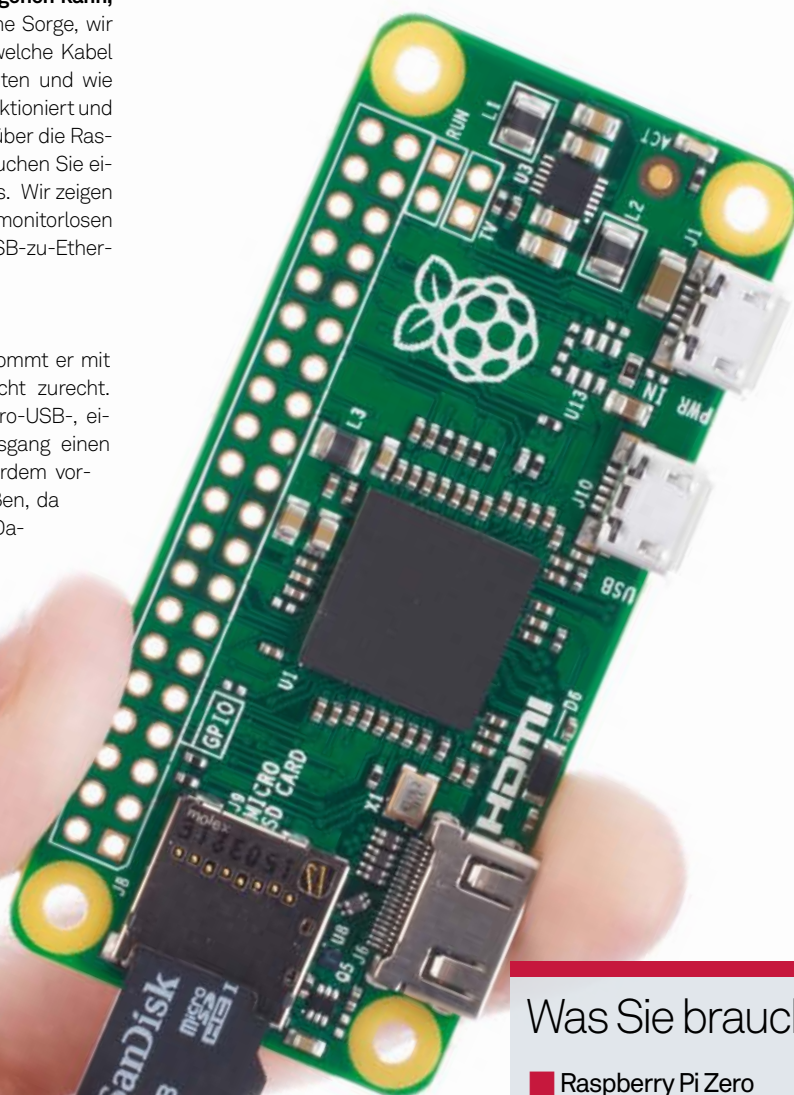
Pi Zero einrichten

Machen Sie sich mit dem Pi Zero vertraut – entweder als Gerät mit Monitor und Tastatur oder auch ohne.

Sie haben sich also einen dieser winzigen, aber leistungsstarken Zeroes geholt, aber bevor der Programmierspaß losgehen kann, müssen Sie sich mehr damit vertraut machen? Keine Sorge, wir gehen mit Ihnen den Raspberry Pi komplett durch, welche Kabel man benötigt, wie Sie eine NOOBS-SD-Karte einrichten und wie man die GPIO-Header auf den Pi lötet. Sobald der Pi funktioniert und hochgefahren ist, zeigen wir Ihnen, wie Sie mit WLAN über die Raspbian-Benutzeroberfläche arbeiten können. Dazu brauchen Sie einen USB-Hub oder einfach nur eine Tastatur und Maus. Wir zeigen Ihnen auch, wie Sie eine Raspbian-SD-Karte für den monitorlosen Gebrauch mit nur einem WLAN-Adapter oder einem USB-zu-Ethernet-Adapter vorbereiten können.

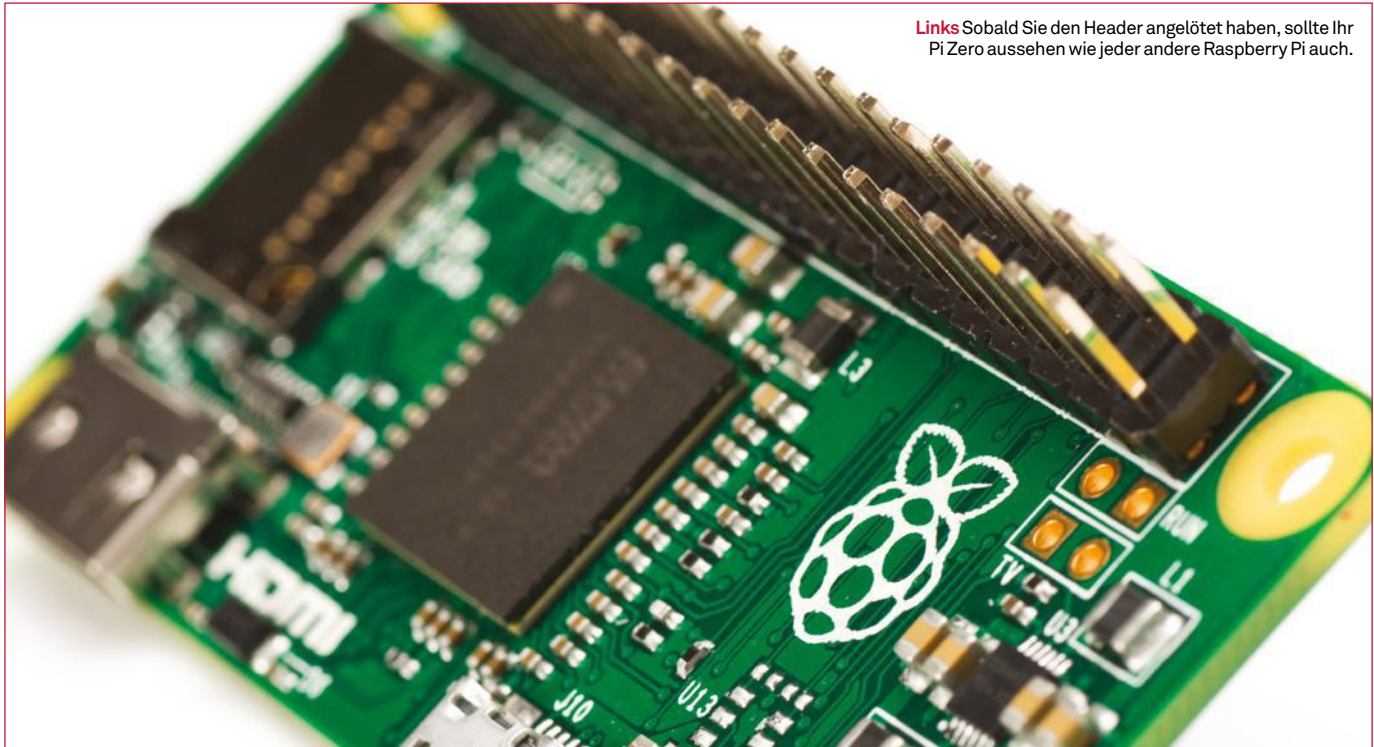
01 Raspberry Pi Zero: Kabelübersicht

Der Raspberry Pi Zero ist sehr klein, daher kommt er mit normal bemessenden USB- und HDMI-Buchsen nicht zurecht. Was Sie brauchen, sind Adapter, die aus einem Micro-USB-, einen normalen USB- sowie aus dem Mini-HDMI-Ausgang einen normalen HDMI-Ausgang machen. Sie müssen außerdem vorsichtig sein, wenn Sie die Micro-USB-Kabel anschließen, da das Micro-USB-Netzkabel in den Anschluss für die Daten-USB-Buchse passen würde. Das Auseinanderhalten ist aber einfach, da die Kabel beschriftet sind.



Was Sie brauchen

- Raspberry Pi Zero
- Micro-USB-Netzteil
- Lötkolben und Lötzinn
- Pi Zero-Adapter-Paket
- Monitor, Maus und Tastatur (optional)
- USB-WLAN- oder USB-Ethernet-Adapter (optional)
- USB-Hub (optional)



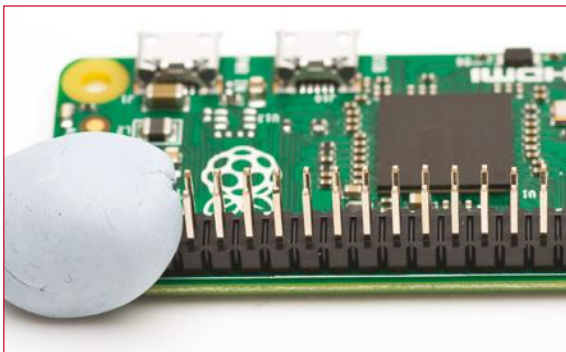
Links Sobald Sie den Header angelötet haben, sollte Ihr Pi Zero aussehen wie jeder andere Raspberry Pi auch.

02 GPIO-Header

Es mag einem etwas Angst einflößen, wenn man das erste Mal auf dem Raspberry Pi lötet – aber es ist gar nicht so schwer! Schwierigkeiten bereitet es lediglich, die richtige Anzahl an GPIO-Header-Pins (40) abzuknippen, da das Kit ja mehr als 40 bereitstellt. An dieser Stelle sollten wir noch erwähnen, dass es nicht schlimm ist, wenn Sie mal ein paar zu viel abklemmen.

03 Lötkits

Lötkolben kosten dieser Tage nicht viel. Wenn Sie vorhaben, viel zu löten, dann sollten Sie sich einen temperaturgesteuerten Kolben zulegen, bei dem man die Spitze wechseln kann. Wir haben hier ein Kit verwendet, das aus Kolben, Löt-sauger und etwas bleifreiem Lötzinn besteht und bei Amazon für unter zehn Euro zu haben ist, und hatten keine Probleme.



04 GPIO-Header an der richtigen Stelle

Bevor Sie die GPIO-Header löten können, müssen Sie in der Lage sein, sie richtig zu halten. Wir empfehlen, hierzu etwas Blu-Tack auf beiden Seiten der Stifte aufzutragen. Das hat auch den Vorteil, dass Sie den Pi umdrehen und den Blu-Tack so auftragen können, dass er fest auf dem Tisch steht, während Sie löten. Später schaben Sie den Kleber einfach ab.

Wenn Sie viel löten, dann lohnt es sich, einen temperaturgesteuerten Lötkolben zu kaufen.

05 Die GPIO-Header löten

Jetzt geht's ans Eingemachte, aber machen Sie sich keine Sorgen! Stellen Sie sicher, dass Sie den Schwamm im Lötkolbenhalter nass gemacht haben, da Sie ihn brauchen, um ab und zu die Spitze des Kolbens sauberzuwischen. Wenn Sie das erste mal mit dem Kolben löten, wird das Heizelement in den ersten Minuten sehr viel Rauch abgeben, aber machen Sie sich nicht, wenn das passiert. Achten Sie stets auf Ihre Sicherheit und löten Sie nur in gut belüfteten Räumen und versuchen Sie, keine Dämpfe einzuatmen.

Sobald der Kolben heiß ist, tragen Sie auf die Spitze etwas Lötmittel auf und wischen überschüssiges Lötmittel an dem Schwamm ab. Für jeden Pin berühren Sie mit der Kolbenspitze die Unterseite des GPIO-Headers sowie den Metallkontakt des Pis. Dann fügen Sie eine sehr kleine Menge Lötmittel hinzu. Sobald das Lötmittel auf den Stift und den Metallkontakt geflossen ist, könne Sie den Kolben wegnehmen.

Machen Sie beim Löten der GPIO-Header aus folgenden Gründen mehrere Pausen: 1) Sie wollen keine Komponenten auf dem Pi überhitzen. 2) Sie könnten das Plastik der GPIO-Header schmelzen und die Hardware würde nicht mehr halten. Wischen Sie die Spitze des Kolbens auf dem Schwamm ab, um sie während des Lötprozesses sauber zu halten. Ziehen Sie den Kolben abschließend unbedingt aus der Steckdose und lassen Sie ihn irgendwo sicher abkühlen.

WLAN-Fähigkeiten

Wenn das Löten der GPIO-Header gut geklappt hat, könnten Sie einen Schritt weitergehen und versuchen, die Interna eines USB-WLAN-Adapters direkt auf den Pi Zero zu löten. Das ist nützlich, wenn Sie Platz sparen wollen und den Pi Zero als Internet-PC nutzen wollen. Weitere Infos finden Sie auf: hackaday.com/2015/11/28/first-raspberry-pi-zero-hack-piggy-back-wifi/.

GPIO Sobald Sie auf einem männlichen 2x20 Header gelötet haben, werden Ihre GPIOs wie gewohnt funktionieren. Auf der rechten Seite sehen Sie die vier ungebrauchten Pins für Videoausgänge und einen Reset-Schalter.

Video Sie benötigen einen Mini-HDMI-zu-HDMI-Adapter, um diesen Audio-/Video-Port zu verwenden. Sie können auch den RCA-Composite-Video-Ausgang über den ungebrauchten Pin verwenden.

Daten Der Strom-Port auf der rechten Seite ist wie gewohnt Micro-USB. Der Datenport daneben ist jetzt auch Micro-USB, also benötigen Sie wahrscheinlich einen Micro-USB-zu-USB-Adapter

06 Bereiten Sie die NOOBS-SD-Karte vor

Besuchen Sie www.raspberrypi.org/help/noobs-setup für weitere Details. NOOBS benötigt eine SD-Karte, die mit FAT32 formatiert ist. Sie müssen dann das aktuelle NOOBS-Image von https://downloads.raspberrypi.org/NOOBS_latest herunterladen und dann auf die SD-Karte entpacken. Unter Linux sind die Schritte wie folgt:

```
sudo parted /dev/mmcblk0
(parted) mktable msdos
(parted) mkpart primary fat32 0% 100%
(parted) quit
sudo mkfs.vfat /dev/mmcblk0p1
cd /mnt
sudo mkdir pi
sudo mount /dev/mmcblk0p1 pi
cd pi
sudo unzip ~/Downloads/NOOBS_v1_5_0.zip
sync
cd ..
sudo umount pi
```

Retro-Gaming

Die kleine Größe des Pi Zero bedeutet, dass man ihn in einigen Spielcontrollern unterbringen kann. Tatsächlich wissen wir sicher, dass er in den ersten Xbox-Controller passt! Schauen Sie sich einige der anderen Projekte in diesem Buch zur Inspiration an. Es gibt eine ganze Reihe von Dingen, die Sie mit dem Pi Zero oder mit anderen Raspberry-Pi-Modellen machen können. Mehr dazu erfahren Sie auch in unserer Tricks-Sektion.

07 NOOBS starten und Raspbian installieren

Verbinden Sie Ihren Pi Zero wie im ersten Schritt gezeigt. Für eine NOOBS-Installation benötigen Sie auf jeden Fall einen Monitor und eine Tastatur. Allerdings sind eine Maus und ein Ethernet-Adapter bzw. Wi-Fi-Adapter auch sehr nützlich. Drücken Sie „Enter“, um Raspbian auszuwählen, und drücken Sie dann „I“, um zu installieren. Nach Abschluss drücken Sie „OK“ und Ihr Pi wird in Raspbian neu starten. Wenn Sie kein NOOBS verwenden möchten, können Sie Raspbian auf eine SD-Karte auf die üblichen Weise flashen.

08 WLAN konfigurieren

Wenn Sie einen USB-zu-Ethernet-Adapter verwenden, sollte der Pi bereits mit dem Internet verbunden sein. Wenn Sie



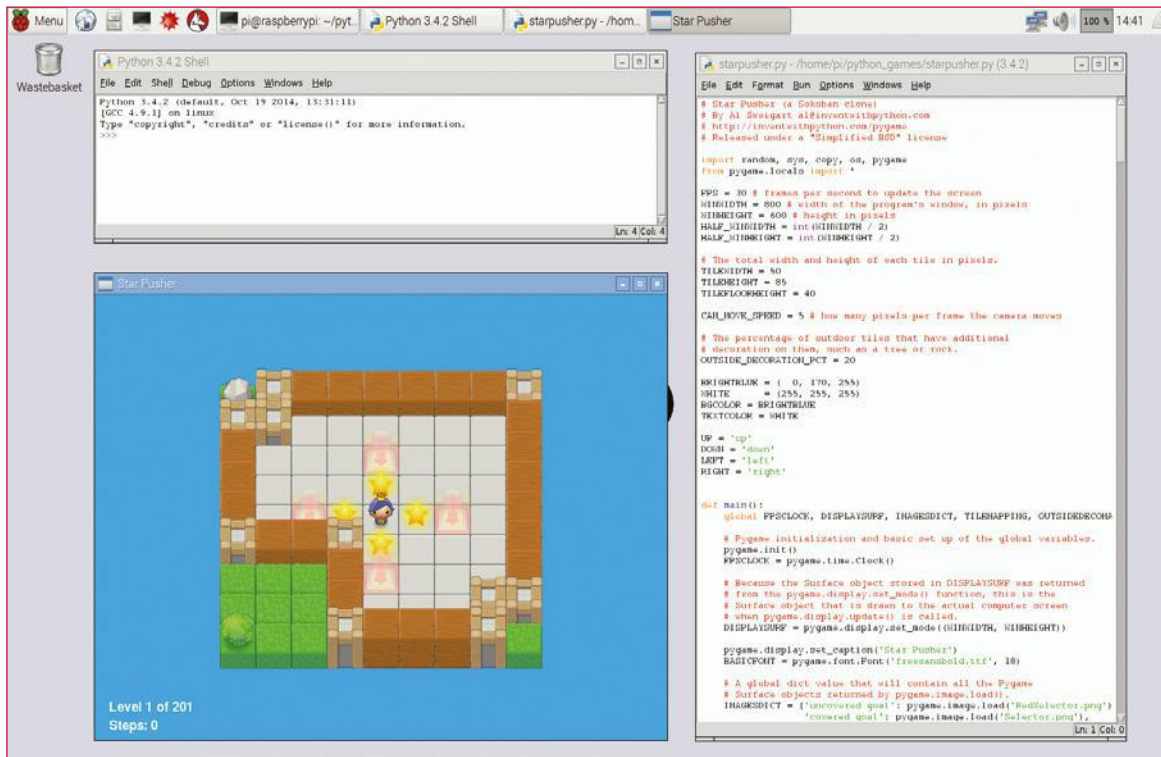
einen WLAN-Adapter verwenden, müssen Sie ihn konfigurieren, um eine Verbindung zu Ihrem drahtlosen Netzwerk herzustellen. Wir benutzen einen Edimax EW-7811UN, der perfekt mit dem Pi zusammenarbeitet. Klicken Sie auf dem Raspbian-Desktop auf das Netzwerk-Icon, um die verfügbaren drahtlosen Netzwerke zu sehen. Sobald Sie auf eines klicken, werden Sie nach dem Passwort gefragt. Nach dessen Eingabe sollten Sie verbunden sein.

09 WLAN von einem anderen Gerät aus einrichten

Wenn Sie den Pi Zero als monitorloses Gerät mit WLAN einsetzen, können Sie die SD-Karte mit einem anderen Linux-Rechner vorbereiten, der bereits mit dem richtigen WLAN-Netzwerk verbunden ist. Sie müssen die SD-Karte mounten und `/etc/wpa_supplicant/wpa_supplicant.conf` editieren. Das ist dieselbe Datei, die unter der Raspbian-Benutzeroberfläche im vorherigen Schritt konfiguriert wurde. Setzen Sie die SD-Karte in den Linux-Rechner. Wir brauchen den Gerätenamen:

```
dmesg | tail -n 3
[320516.612984] mmc0: new high speed SDHC card at address 0001
[320516.613437] mmcblk0: mmc0:0001 SD8GB 7.35 GiB
```

Das Gerät heißt `/dev/mmcblk0` – jetzt müssen wir herausfinden, auf welcher Partition sich das Stammverzeichnis (Root) befindet



Links Der Zero mag winzig sein, aber er lässt sich genauso gut programmieren.

(Beachten Sie, dass wir hier ein NOOBS-Image benutzen; bei einem Raspbian-Image ist das anders).

```
sudo parted /dev/mmcblk0 print
```

Sie erhalten dann eine Liste mit den Partitionen. Die größte Partition ist die Root-Partition. In diesem Fall ist es die Partition 7, also ist das Root-Dateisystem unter `/dev/mmcblk0p7` zu finden. Um die SD-Karte zu installieren und die Datei `wpa_supplicant.conf` zu bearbeiten, gehen Sie wie folgt vor:

```
cd /mnt
sudo mkdir pi
sudo mount /dev/mmcblk0p7 pi/
cd pi/
sudo nano etc/wpa_supplicant/wpa_supplicant.conf
```

Füllen Sie Ihre WLAN-Details aus:

```
network={
    ssid="your_wifi_network"
    psk="your_wifi_password"
    key_mgmt=WPA-PSK
}
```

Zum Schluss geben Sie Folgendes ein:

```
cd ..
sudo umount pi/
```

10 Fernzugriff auf Ihren Pi

Sie können `nmap` verwenden, um das lokale Netzwerk nach Ihrem Raspberry Pi zu scannen. Sie müssen den Adressbereich Ihres lokalen Netzwerks kennen (gemeinsame Netzwerke sind `192.168.1.0/24` und `192.168.2.0/24`). Sie können es mit dem Befehl `ip addr` finden. `Nmap -p22 -sV 192.168.157.0/24` sucht nach

einer Liste von Geräten mit SSH open. Beispielausgabe:

```
Nmap scan report for 192.168.157.29
Host is up (0.070s latency).
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      (protocol 2.0)
```

Dann können Sie SSH mit folgendem Befehl laufen lassen:

```
ssh pi@192.168.157.29
```

Das Passwort ist „raspberrypi“. Wenn Sie den Pi monitorlos verwenden, sollten Sie die Benutzeroberfläche deaktivieren.

```
sudo systemctl set-default multi-user.target
```

11 Einen VNC-Server aufsetzen

VNC steht für Virtual Network Computing. Mit VNC können Sie über das Netzwerk auf den Raspbian-Desktop zugreifen (Sie brauchen nur Strom und LAN/WLAN). Es gibt keine Audio-Unterstützung, aber für alle anderen Aufgaben sollte VNC eine akzeptable Leistung liefern. Sie können einen VNC-Server mit folgenden Befehlen installieren:

```
sudo apt-get update
sudo apt-get install tightvncserver
```

Es gibt viele kostenlose VNC-Clients. Eine Suchmaschine kann Ihnen helfen, den richtigen Client zu finden. Um eine VNC-Sitzung auf Ihrem Pi zu starten, melden Sie sich über SSH an und starten Sie dann `engvncserver`. Sie werden aufgefordert, ein Passwort einzugeben, wenn Sie es zum ersten Mal ausführen. Sie können eine Bildschirmauflösung mit der Option `-geometry` angeben, z.B. `-geometry 1024x768`. Eine bestehende VNC-Sitzung können Sie mit `tightvncserver -kill:1` beenden, wobei 1 die Sitzungsnummer ist.



ENTWICKELN MIT **PYTHON**

Mit Python wird auf der ganzen Welt Webentwicklung,
akademische Forschung und Ingenieursarbeit betrieben.
So wenden Sie Ihr Python-Wissen im Profibereich an.



Systemadministration

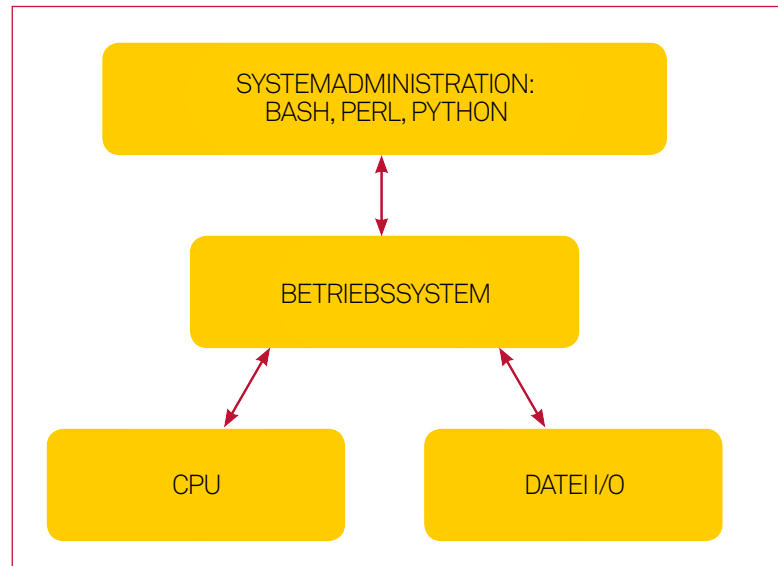
Holen Sie sich mit Python Hilfe bei alltäglichen Wartungsarbeiten, um Ihr System im Bestzustand zu erhalten.

Systemadministration gehört zu den lästigsten Aufgaben, die beim Betrieb eines Rechners anfallen. Es verwundert nicht, dass Systemadministratoren stets auf der Suche nach Automatisierungslösungen sind, um so Zeit für relevante Arbeit zu gewinnen. Den Anfang machten einfache Shell-Skripte, es folgten verschiedenste Programmiersprachen. Perl war lange Zeit das Mittel der Wahl für die Entwicklung von Wartungs-Tools, doch Python wird für diese Aufgaben immer beliebter. Inzwischen bringen fast alle Linux-Distros einen Python-Interpreter mit, der Skripte ausführen kann – es gibt also keinen triftigen Grund mehr, auf Automatisierungsskripte zu verzichten.

Der Großteil der Arbeit findet auf Systemebene statt, was einige zusätzliche Python-Module erfordert wie beispielsweise „os“. Das Modul übernimmt den Großteil der Kommunikationsaufgaben mit dem Betriebssystem. Üblicherweise ist der erste Schritt, die Systemumgebung zu erkunden, um herauszufinden, auf welche Informationen Ihr Skript zurückgreifen kann. Folgender Code erzeugt ein Mapping-Objekt, das es ermöglicht, mit den momentan aktiven Umgebungsvariablen zu interagieren:

```
import os
os.environ
```

Eine Liste der verfügbaren Umgebungsvariablen erhalten Sie mit der Funktion `os.environ.keys()`, wobei Sie auf einzelne Variablen mit `os.environ[key]` zugreifen können. Auch beim Initialisieren eines Unterprozesses kommen Umgebungsvariablen zum Einsatz. Hier können Sie also bei der Optimierung von Unterprozessen ansetzen, um etwa Pfade (mit PATH) oder das aktuelle Arbeitsverzeichnis anzupassen. Die Funktion `putenv` kann zwar Werte editieren, ist aber nicht auf allen Systemen verfügbar; daher bietet es sich an, alle Änderungen gleich beim Umgebungs-Mapping vorzunehmen.



Ein weiterer Automatisierungskandidat ist die Dateiverwaltung. Öffnen Sie das Arbeitsverzeichnis mit:

```
cwd = os.getcwd()
```

Lassen Sie dann eine Liste der enthaltenen Dateien anzeigen:

```
os.listdir(cwd)
```

Navigieren Sie mit der Funktion `os.chdir(new_path)` durch das Dateisystem und öffnen Sie relevante Dateien mit `os.open()` zur weiteren Bearbeitung. Mit den Funktionen `os.read()` und `os.write()` können Sie lesen und schreiben, nach Ende der Bearbeitung schließen Sie die Datei mit `os.close()`.

Oben Python-Skripte ermöglichen die Interaktion mit Ihrem Betriebssystem.



Unterprozesse in Python ausführen

Unix verfolgt die Philosophie, einzelne Aufgaben von kleinen, extrem spezialisierten Programmen erledigen zu lassen. Für komplexere Anforderungen werden diese dann einfach in Serie geschaltet. Auch Python-Skripting profitiert von dieser Philosophie. Es gibt einige Dienstprogramme, die ohne großen Mehraufwand direkt eingesetzt werden können. Früher war es üblich, Programme durch Funktionsaufrufe aus dem os-Modul wie etwa `popen()` oder `spawnl()` einzubinden, doch das subprocess-Modul eignet sich viel besser dazu. So öffnen Sie beispielsweise ls:

```
import subprocess
subprocess.run(['ls', '-l'])
```

Es wird eine lange Dateiliste des aktuellen Verzeichnisses ausgegeben. Die in Python 3.5 eingeführte Funktion `run()` wird dafür empfohlen. Für ältere Versionen oder weitere Anpassungen können Sie auf das zugrundeliegende `popen()` zurückgreifen. So erhalten Sie die Ausgabe:

```
cmd_output = subprocess.run(['ls', '-l'],
stdout=subprocess.PIPE)
```

Die Variable „cmd_output“ ist ein CompletedProcess-Objekt, das den Rückgabewert und eine Zeichenkette mit der Ausgabe von stdout enthält. Das mag ungewöhnlich wirken, aber die Methodik ist eigentlich dieselbe wie immer.

Zeitplanung mit cron

Fertige und funktionierende Skripte können Sie fortan automatisch und ohne weiteres Zutun starten lassen. Unix-Systeme erlauben es per cron, einen Zeitplan für Skriptausführung zu erstellen. Der Befehl `crontab -l` listet den aktuellen Inhalt der cron-Datei auf, `crontab -e` erlaubt die Bearbeitung der von cron regelmäßig auszuführenden Aufgaben.

Webentwicklung

Python bietet einige Frameworks für Webentwicklung. Wir stellen die beliebtesten vor.

Bei serverbasierten Inhalten und Berechnungen eignet sich eine Web-App für Endnutzer optimal. Das Django-Framework ist mit Plugins ausgestattet und arbeitet nach dem DRY-Prinzip (Don't Repeat Yourself), was für eine kurze Entwicklungsdauer förderlich ist. Da Django auf Python basiert, können Sie es zumeist direkt per `sudo pip install Django` installieren. Je nach Funktionsumfang Ihrer App brauchen Sie noch eine Datenbank wie MySQL oder PostgreSQL zur Speicherung von Programmdaten. Django bietet einige Möglichkeiten, den Ausgangspunkt für den Code eines neuen Projekts zu erstellen, wie etwa:

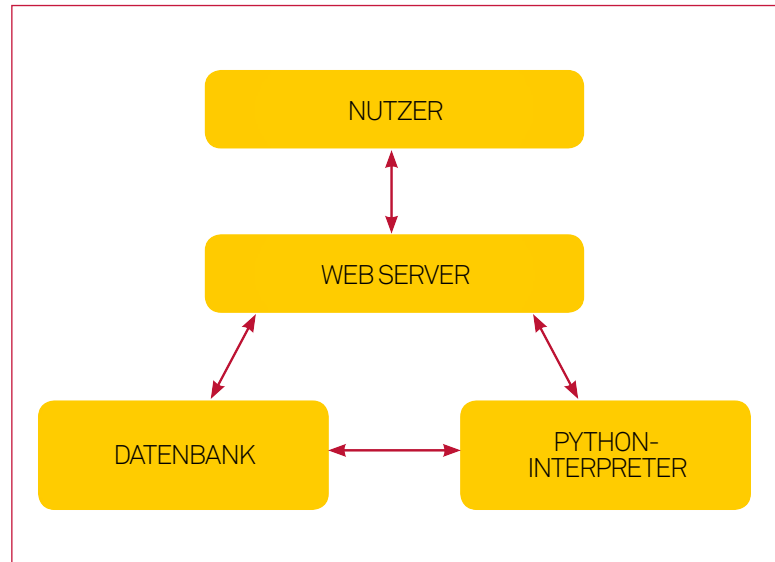
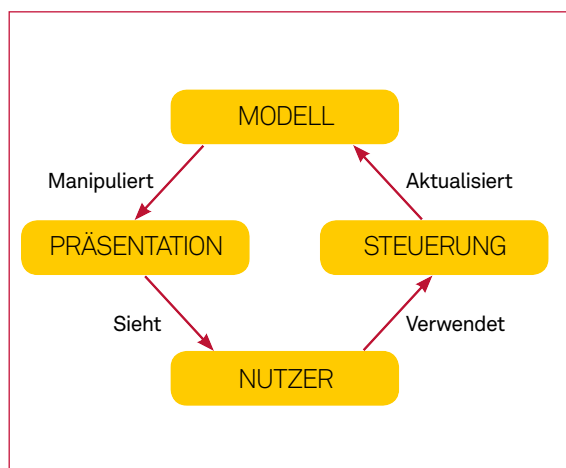
■ `django-admin startproject newsite`

Dieser Befehl erzeugt eine Datei namens „manage.py“ und ein Unterverzeichnis namens „newsite“. Die Datei „manage.py“ speichert einige Funktionen zur Verwaltung der App, das Unterverzeichnis enthält die Dateien „__init__.py“, „settings.py“, „urls.py“ und „wsgi.py“. Dateien und Unterverzeichnis stellen zusammen ein Python-Paket dar, das beim Aufruf der Website geladen wird. Grundsätzliche Einstellungen Ihrer Website werden in „settings.py“ gespeichert, während URL-Deklarationen – eine Tabelle von Website-Inhalten – in „urls.py“ hinterlegt sind. Der Einstiegspunkt für WSGI-kompatible Webserver wird in „wsgi.py“ definiert.

Ist die App fertig, sollte sie auf einem korrekt konfigurierten, gehärteten Webserver gehostet werden – während der Entwicklung ist das allerdings eher hinderlich. Als Lösung bietet Django einen eigenen Webserver, den Sie aktivieren, indem Sie in das Projektverzeichnis „newsite“ wechseln und dort folgenden Befehl ausführen:

■ `python manage.py runserver`

Damit starten Sie einen Server, der über den Systemport 8000 auf Ihrem Rechner erreichbar ist. Er ist für Entwicklungszwecke angepasst und aktualisiert bei jeder Abfrage den Python-Code, damit Sie den Server nicht bei jeder Änderung neu starten müssen. Ist das Projekt nach Ihren Vorstel-



lungen eingerichtet, können Sie mit der Entwicklung der App beginnen. Geben Sie innerhalb des Verzeichnisses „newsite“ Folgendes ein:

■ `python manage.py startapp newapp`

So erstellen Sie ein Unterverzeichnis namens „newapp“, das unter anderem die Dateien „models.py“, „tests.py“ und „views.py“ enthält. Die simpelste Ansicht wird folgendermaßen generiert:

```

from django.http import HttpResponse
def index(request):
    return HttpResponse("Hello world")
  
```

Sie ist dann allerdings noch nicht sichtbar – dazu bedarf es noch der Einrichtung einer URLconf. Existiert die Datei „urls.py“ noch nicht, erstellen Sie sie und fügen Sie hinzu:

```

from django.conf.urls import url
from . import views
urlpatterns = [ url(r'^$', views.index,
                  name='index'), ]
  
```

Registrieren Sie die URL dann mit folgendem Code in Ihrem Projekt:

```

from django.conf.urls import include, url
from django.contrib import admin
urlpatterns = [ url(r'^newapp/', include('newapp.urls')),
               url(r'^admin', admin.site.urls), ]
  
```

Der Code gehört in die Datei „urls.py“ im Hauptprojekt. Ihre neu erstellte App können Sie jetzt mit der URL `http://localhost:8000/newapp/` aufrufen.

Oben Python-Interpreter arbeiten mit Datenbanken und befeuern so Webserver.

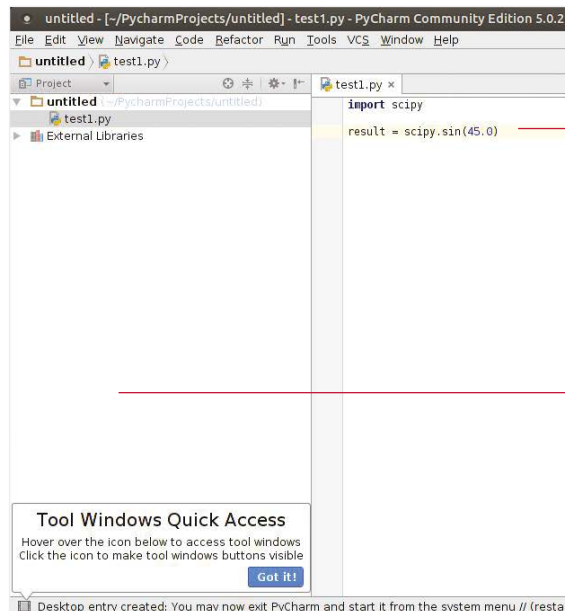
Links Für Nutzerinterfaces wird oft das Model-View-Controller-Konzept verwendet.

Virtuelle Umgebungen

Der Anfang der App-Entwicklung mit Python mag manchem wie ein Abstieg in die Hölle der Abhängigkeiten vorkommen: Mehrere Python-Pakete hängen wiederum von anderen Python-Paketen ab. Das kann eine Stärke, aber auch eine Schwäche sein. Mit `virtualenv` gibt es zum Glück Licht in diesem Tunnel. Sie können damit für jedes Projekt eine virtuelle Umgebung erstellen und so alle Abhängigkeiten für Ihr eigenes Paket genau festhalten.



Verwendung der PyCharm IDE



Editoransicht Das Hauptfenster des Editors kann nach Ihren Bedürfnissen angepasst und an andere Editoren wie emacs angelehnt werden. Es hebt die Syntax farbig hervor und zeigt sogar Fehlerstellen im Code an.

Projektansicht Dies ist die zentrale Ansicht für Ihr Projekt. Alle Dateien und Codebibliotheken werden hier angezeigt, per Rechtsklick können weitere hinzugefügt, Testläufe gestartet oder Debugging vorgenommen werden.

Statusleiste Hier finden Sie aktuelle Details zum gerade laufenden System.



Django verfügt sogar über einen Webserver innerhalb des Frameworks.

Die Datenbank kommt bei der App-Entwicklung meist als Letztes. Verbindungsdetails wie Benutzernamen und Passwörter sind in der Datei „settings.py“ gespeichert. Diese Informationen werden für alle im Projekt enthaltenen Anwendungen verwendet. So erstellen Sie die grundsätzliche Datenbankstruktur:

```
python manage.py migrate
```

Sie können für Ihre eigenen Anwendungen in der Datei „models.py“ Datenmodelle definieren. Sobald ein Datenmodell erstellt wird, können Sie die App zum Abschnitt INSTALLED_APPS in „settings.py“ hinzufügen, damit Django sie bei Datenbankinteraktionen mit einbezieht. Initialisieren Sie sie folgendermaßen:

```
python manage.py makemigrations newapp
```

Nehmen Sie an der erstellen Datenbank dann folgende Migration vor:

```
python manage.py migrate
```

Bei jeder Änderung des Datenmodells müssen Sie die Schritte makemigrations und migrate wiederholen. Sobald Ihre App fertig ist, können Sie sie auf den Webserver umziehen. Prüfen Sie immer, ob das Django-Framework vielleicht bereits für Sie nutzbaren Code beinhaltet, bevor Sie selbst viel Arbeit in die Entwicklung stecken.

Entwicklung im Terminal

Bei der Entwicklung haben Sie wahrscheinlich nicht selten mehrere Terminalfenster geöffnet, die zum Beispiel Ihren Code-Editor, Informationen zum Server und etwaige testweise generierte Ausgaben beinhalten. Auf dem eigenen Desktop ist das kein großes Problem, bei der Arbeit über Remote-Verbindung sollten Sie dafür die Verwendung von tmux in Erwägung ziehen. Damit erhalten Sie möglicherweise eine wesentlich stabilere Terminalumgebung.

Andere Python-Frameworks

Django gehört zu den beliebtesten Frameworks für die Webentwicklung, aber es ist bei weitem nicht das einzige. Seine Konkurrenten spielen ihre Stärken in anderen Feldern aus: Wenn Sie ein vollständig unabhängiges Framework brauchen, sollten Sie sich web2py ansehen – sie können ein komplettes System mitsamt Datenbanken, Webserver und Tracker mit Ticket-System auf Basis eines Frameworks erstellen, das sogar direkt von einem USB-Speichermedium starten kann.

Wer noch weniger Framework braucht, hat einige Mini-Frameworks zur Auswahl. CherryPy ist ein rein Python-basierter Multithread-Webserver zur Einbettung in Ihre Apps. Tatsächlich enthalten TurboGears und web2py diesen Server. Ein weiteres beliebtes Mikro-Framework ist flask, welches integrierte Komponententests, Jinja2-Template-funktionen und Dispatching von Verbuchungsaufträgen per REST unterstützt.

Eines der ältesten Frameworks ist zope, inzwischen bereits in Version 3 verfügbar und umgetauft in BlueBream. Zope selbst ist allerdings eher eingeschränkt nützlich – die vielen auf zope basierenden Frameworks sind für die Praxisanwendung interessanter. Pyramid ist beispielsweise ein schnelles, leicht verwendbares Framework, das sich auf die Kernfunktionen von Web-Apps beschränkt. Es beinhaltet unter anderem Template-Optionen, statische Inhalte und URL-Mapping in Code und bietet dabei noch zusätzliche Sicherheitsfunktionen.

Wenn Sie noch auf der Suche nach Ideen sind, können Sie sich verschiedene Open-Source-Projekte ansehen, die auf diesen Frameworks basieren, von Blogs bis hin zu Bugtrackern. Von deren Implementation können Sie vieles lernen, was für Ihre eigene Anwendung nützlich sein kann.

Rechnergestützte Wissenschaft

Dank der vielen verfügbaren Pakete wird Python zusehends zur wichtigsten Programmiersprache für wissenschaftliches Rechnen.

Python ist inzwischen eine der wichtigsten wissenschaftlich verwendeten Programmiersprachen. Es gibt enorm viele Pakete für fast alle Anwendungszwecke, und was Python nicht kann, wird einfach outgesourct: Um Unzulänglichkeiten der Programmiersprache auszugleichen, integriert Python einfach Code aus C oder FORTRAN und lagert so komplexe Berechnungen in effizienteren Code aus.

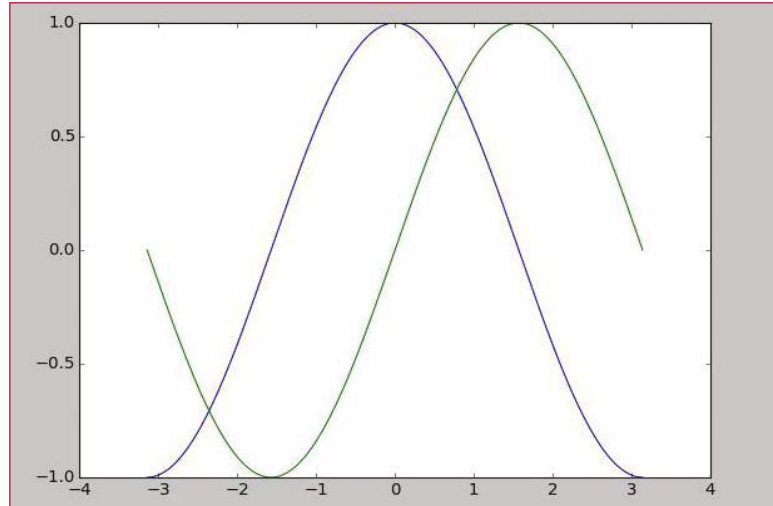
Im Zentrum der meisten Pakete für wissenschaftliche Berechnungen steht numpy. Python ist objektorientiert und daher ineffizient – ohne streng definierte Typen muss es bei jeder Rechenoperation alle Parameter prüfen. Numpy führt einen neuen Datentyp ein, das Array, welches einige Probleme löst. Arrays können nur eine Art von Objekt beinhalten, was Python erlaubt, seine Berechnungsgeschwindigkeit beinahe auf das Niveau von C oder FORTRAN zu heben. Zur Illustration ein Klassiker, die **for**-Schleife: Soll ein Vektor um einen gewissen Wert skaliert werden, also etwa $a \cdot b$, sähe dies so aus:

```
for elem in b:
    c.append(a * elem)
```

In numpy sieht es hingegen so aus:

```
a*b
```

Also nicht nur schneller, sondern auch kürzer und klarer. Neben dem neuen Datentyp bietet numpy auch überladene

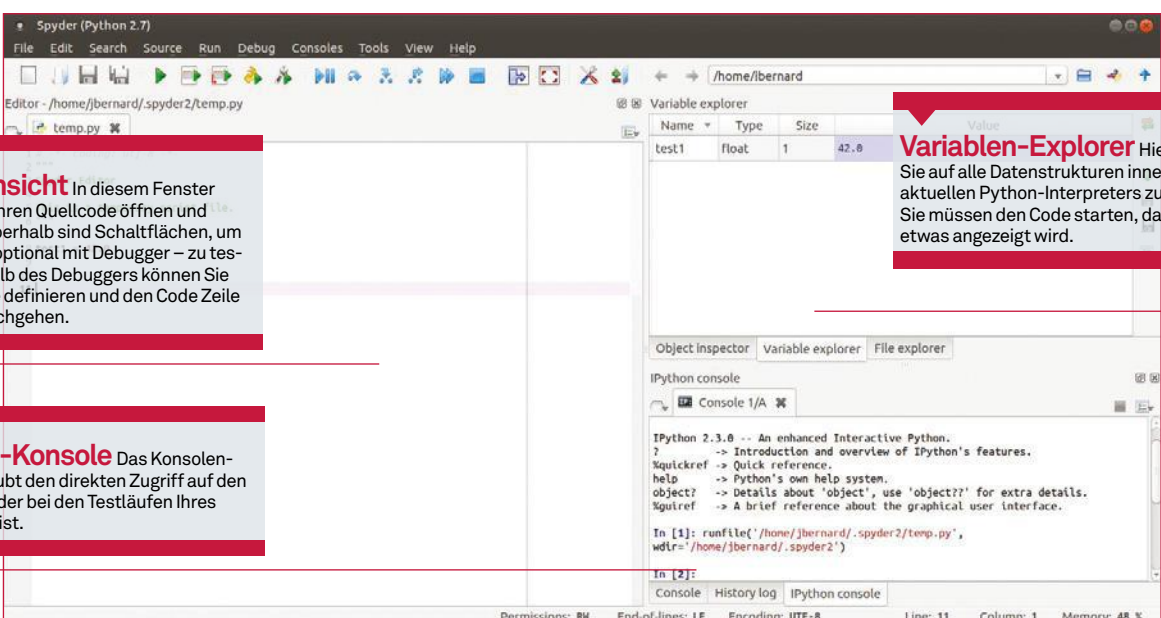


Versionen der nützlichsten Operatoren wie Multiplikation oder Division. Auch optimierte Versionen einiger Funktionen (zum Beispiel trig) stehen zur Verfügung, um den neuen Datentyp voll ausnutzen zu können. Das größte auf numpy aufbauende Paket ist scipy. Scipy bietet spezielle Unterfunktionen für verschiedene Bereiche, welche jeweils noch einzeln importiert werden müssen. Arbeiten Sie beispielsweise

Oben Mit dem numpy-Paket ist es ein Leichtes, Daten zu visualisieren.



Spyder, die IDE für Wissenschaftler



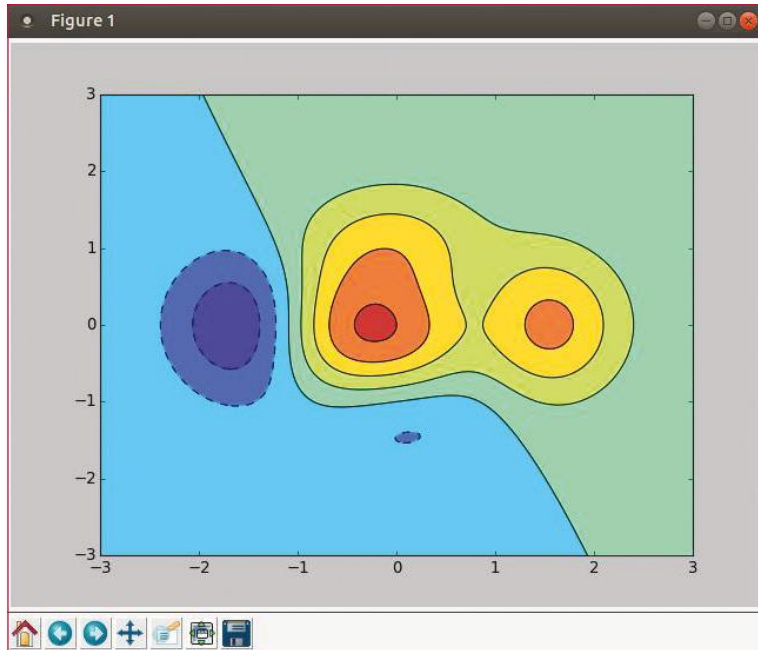
Editoransicht In diesem Fenster können Sie Ihren Quellcode öffnen und editieren. Oberhalb sind Schaltflächen, um den Code – optional mit Debugger – zu testen. Unterhalb des Debuggers können Sie Haltepunkte definieren und den Code Zeile für Zeile durchgehen.

IPython-Konsole Das Konsolenfenster erlaubt den direkten Zugriff auf den Interpreter, der bei den Testläufen Ihres Codes aktiv ist.

Variablen-Explorer Hier können Sie auf alle Datenstrukturen innerhalb des aktuellen Python-Interpreters zugreifen. Sie müssen den Code starten, damit hier etwas angezeigt wird.



Jupyter gibt mathematische Ausdrücke auf der erzeugten Webseite korrekt aus.



se mit Differentialgleichungen, können Sie zur Lösung die Sektion „integrate“ verwenden, indem Sie folgenden Code eingeben:

```
import scipy
import scipy.integrate
result = scipy.integrate.quad(lambda x: sin(x), 0, 4.5)
```

Differentialgleichungen sind in fast jeder wissenschaftlichen Disziplin nützlich. Statistische Analysen wiederum können Sie mit der Sektion „stats“ vornehmen. Alternativ können Sie mit den Sektionen „signal“ und „fftpack“ Signalverarbeitung betreiben – dieses Paket ist sicherlich das wichtigste für wissenschaftliche Datenverarbeitung.

Haben Sie alle Daten beisammen, wollen Sie sie wahrscheinlich für einen besseren Überblick als Diagramm visualisieren. Das wichtigste Paket hierfür ist matplotlib. Nutzer des Grafik-Pakets von R werden sich gleich wie zuhause fühlen: matplotlib hat sich für sein Design einige Ideen davon geborgt. Es gibt zwei Kategorien von Graphen-Funktionen: low-level und high-level. High-level-Funktionen übernehmen den Großteil der untergeordneten Arbeiten wie Erstellung der Ausgabeansicht, Zeichnen der Achsen oder Auswahl des Koordinatensystems. Low-level-Funktionen erlauben die Kontrolle über fast alle Komponenten der Ausgabe, von der Anzeige einzelner Pixel bis zum gesamten Ausgabefenster. Graphen werden zudem in ein speicherbasiertes Fenster ausgegeben, sodass die Ausgabe kontinuierlich erfolgen kann, während das Programm auf einem Speicher-Cluster läuft.

Oben Komplexe Ausgabemöglichkeiten sind essenziell.

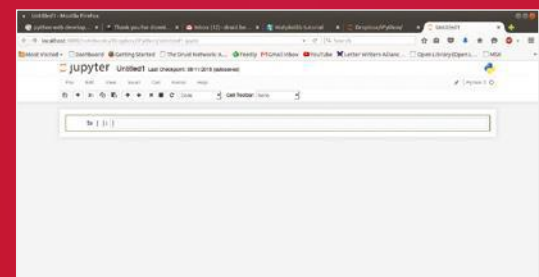
Schneller, höher, weiter

Manchmal benötigen Sie so viel Rechengeschwindigkeit, wie Ihre Hardware überhaupt hergibt. Dafür können Sie auf Cython zurückgreifen, um optimierten C-Code aus einem anderen Projekt zu entleihen und in Ihr Python-Programm einzubauen. Gerade bei der wissenschaftlichen Arbeit haben Sie mit dieser Strategie häufig Zugriff auf jahrzehntlang optimierten, hochgradig spezialisierten Code – Sie müssen meist nicht alles eigens entwickeln und das Rad mit jedem Projekt neu erfinden.

Interaktive Wissenschaft mit Jupyter

Viele wissenschaftliche Probleme erfordern einen interaktiven Ansatz zur Lösung. Das IPython Web Notebook bot als Erstes diese Möglichkeit und wurde inzwischen in Jupyter umgetauft. Wer Mathematica oder Maple kennt, wird das Interface sofort verstehen. Jupyter startet einen Server-Prozess auf Port 8888 und öffnet anschließend ein Browserfenster, in dem Sie ein Arbeitsblatt erstellen können. Wie bei vielen ähnlichen Programmen werden die Einträge chronologisch abgearbeitet, nicht in der Anzeigereihenfolge auf dem Arbeitsblatt. Das kann zunächst verwirren, aber es bedeutet auch, dass nach Änderungen in vorangegangenen Einträgen alle folgenden Einträge noch einmal manuell aufgerufen werden müssen, damit alle Änderungen bei den weiteren Berechnungen berücksichtigt werden.

Jupyter gibt mathematische Ausdrücke auf der erzeugten Webseite korrekt aus und unterstützt die nötige Formatierung. Sie können auch Dokumentation und Code auf einer Seite mischen und dadurch sehr anschauliches Lehrmaterial erstellen, das es Studierenden erlaubt, sich mit Techniken vertraut zu machen und sie direkt auszuführen und in Aktion zu sehen. Standardmäßig bettet Jupyter auch matplotlib-Ausgaben im gleichen Arbeitsblatt in eine Ergebnissection ein, sodass Sie einen Graphen und den Code, mit dem er generiert wurde, abgleichen können. Das ist ein großer Schritt in Richtung Reproduzierbarkeit wissenschaftlicher Arbeit. So können Sie stets rekapitulieren, wie die Analyse vorgenommen wurde, und jedes Ergebnis nachvollziehen.



Jupyter Notebook ist eine Webanwendung, die Live-Ansichten von Code und Berechnungen generiert.

Für symbolische Mathematik sind einfachere Lösungen wie Mathematica oder Maple geeigneter. Zum Glück gibt es sympy, das ähnlich funktioniert wie die beiden. Sie können Python damit für symbolische Infinitesimalrechnung oder Algebra nutzen – etwas seltsam ist bei sympy nur, dass Sie zunächst in der Funktion symbols() definieren müssen, welche Variablen in Ihren Berechnungen überhaupt gültig sind. Mit gültigen, registrierten Variablen können Sie anschließend Berechnungen anstellen.

Haben Sie große Datenmengen zur Verarbeitung oder Analyse vorliegen, kann das Paket pandas helfen. Pandas unterstützt Dateiformate wie CSV, Excel-Tabellen oder HDF5. Sie können Datensätze zusammenführen, trennen und unterteilen. Für größtmögliche Geschwindigkeit werden komplexe Berechnungen mit Cython-Code vorgenommen, der C-Funktionen beinhaltet. Auch hier wurden wieder einige Ideen zur Datenverarbeitung aus R entlehnt.

Es spricht also nichts dagegen, Python für wissenschaftliche Berechnungen zu verwenden – gehen Sie doch direkt das nächste mathematische Problem damit an!

Robotik und Elektronik

Bringen Sie Ihren Code in die reale Welt und hauchen Sie ihm mithilfe von Roboter-technologie Leben ein.

Robotik ist für Ihren Code der direkteste Weg in die reale Welt. So können Sie auf Basis von echtem Sensoren-Input echte Aktoren antreiben und echte Arbeit verrichten.

Zunächst einmal muss ein Roboter die umgebende Welt wahrnehmen können. Für Menschen ist der wichtigste Sinn sicherlich das Sehen. Webcams sind inzwischen so günstig und leicht einzurichten, dass es kein allzu großer Schritt ist, Ihrem Roboter das Augenlicht zu verleihen. Schwieriger ist es allerdings, die gesammelten Daten dann zu verarbeiten. Das OpenCV-Paket springt hier in die Bresche und bietet von einfachen Bildaufnahme- und Verarbeitungsfunktionen bis hin zu komplexen Anwendungen wie Gesichtserkennung oder 3D-Objekterkennung fast alles. Sie können bewegte Objekte im Gesichtsfeld der Kamera identifizieren und, wenn Sie wollen, Ihrem Roboter mit OpenCV sogar Entscheidungsgewalt einräumen – OpenCV bringt einige Optionen für maschinelles Lernen



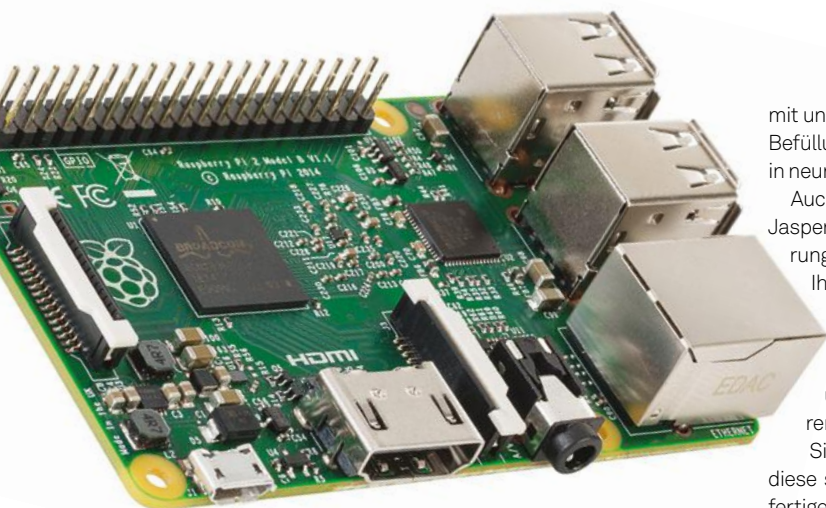
Arduino

Im Gegensatz zum Raspberry Pi ist der Arduino kein echter Rechner mit vollständigem Betriebssystem, sondern ein Mikrocontroller, der Code mithilfe seiner Firmware interpretiert und ausführt. Er wird hauptsächlich als Interface für Motoren, Sensoren und vieles mehr verwendet.

mit und kann statistische Klassifikation, Daten-Clustering, die Befüllung von Entscheidungsbäumen oder sogar Funktionen in neuronalen Netzwerken übernehmen.

Auch das Hören kann äußerst nützlich sein. Das Projekt Jasper entwickelt ein vollständiges System zur Sprachsteuerung, mit dem Sie die grundsätzliche Möglichkeit besitzen, Ihren Roboter auf gesprochene Kommandos reagieren zu lassen. Es ist inzwischen so weit fortgeschritten, dass es Spracheingaben in Text umwandeln kann. Sie müssen dann noch ein Kommando-Mapping vornehmen und festlegen, welcher Textbaustein welchen auszuführenden Aufgaben entspricht.

Sie können noch viele andere Sensoren verwenden, aber diese sind eher selten als bequem verwendbare, anschlussfertige Hardware im Handel verfügbar. Meist brauchen solche Komponenten, etwa zur Messung von Temperatur, Druck, Aus-



Raspberry Pi

Welchen Rechner Sie für Ihr Robotikprojekt verwenden, steht Ihnen frei; der berühmte Raspberry Pi ist jedoch eine gute Wahl. Er ist klein genug, um in fast jedem robotischen Konstrukt sein Plätzchen zu finden. Da er bereits serienmäßig mit Linux und Python ausgestattet ist, dürften Sie Ihren Code ganz einfach auf den Pi kopieren können. Er hat zudem eine I/O-Schnittstelle, an die Sie direkt Sensoren anschließen können.

ROS – Robot Operating System

Wenn Sie auf einem normalen Rechner mit Linux-Betriebssystem programmieren, kann die Art der Ausführung Ihres Codes für die Datenverarbeitung bei Echtzeitanwendungen in der Robotik schlichtweg ungeeignet sein. An diesem Punkt lohnt es sich, ein spezielles Betriebssystem in Betracht zu ziehen, nämlich das Robot Operating System (ROS). ROS bietet ein einheitliches Interface zwischen Ihrem Code und der Hardware, die ihn ausführt, mit minimalen Leistungsverlusten. ROS brilliert dabei, die Kommunikation zwischen verschiedenen Prozessen auf dem Rechner oder sogar mehreren in einem Netzwerk verbundenen Computern zu koordinieren. Einzelne Prozesse werden nicht isoliert betrieben und voneinander abgeschottet, sondern stehen in einer Art Systemgraph in ständigem Austausch.

Der Ansatz ist hier allerdings nicht, ROS wie eine Code-Bibliothek in Ihrem Python-Code anzusteuern; es ist

ein vollständiges Betriebssystem. Vielmehr schreiben Sie Code in Python, den Sie anschließend in ROS ausführen können. Das System sollte prinzipiell so wenig wissen wie nötig; ein gutes Interface sollte schlicht und einfach funktionieren, ohne unnötige Zusatzinformationen über das Wer und Warum Ihres Codes. Nur so kann es sich gut in den ROS-Prozessgraphen einordnen. Es gibt einige Standardbibliotheken für Koordinatenberechnungen, was für die Arbeit mit Sensoren und Robotergliedmaßen sehr nützlich ist. Auch für präemptives Multitasking zur Datenverarbeitung oder zur Erstellung und Verwaltung von Nachrichten zur Kommunikation zwischen Prozessen gibt es Bibliotheken. Für zeitkritische Aufgaben existiert eine Plugin-Bibliothek, mit der Sie Plugins in C++ schreiben und danach in ROS-Paketen laden können.



Untergeordnetes mit Arduino erledigen

Editorfenster Sie haben Zugriff auf viele Code-Bibliotheken und Unterstützung für diverse Arduino-Platinen. Programmiert wird quasi in C – wer Python beherrscht, wird hier also nicht ins kalte Wasser geworfen.

Ausgabefenster Hier erscheinen die Ausgabewerte verschiedener Aufgaben, etwa nach dem Kompilieren des Quellcodes oder nach der Übertragung von Code auf den Arduino.

Statusleiste Die Statusleiste zeigt an, für welche Art Platine Sie aktuell programmieren und an welcher Schnittstelle die Arduino-Entwicklungsumgebung andockt. Prüfen Sie vor jeder Übertragung, ob die Werte korrekt sind.

```
File Edit Sketch Tools Help
AnalogInOutSerial
/*
  Analog input, analog output, serial output

  Reads an analog input pin, maps the result to a range from 0 to 255
  and uses the result to set the pulsewidth modulation (PWM) of an output pin.
  Also prints the results to the serial monitor.

  The circuit:
  * potentiometer connected to analog pin 0.
    Center pin of the potentiometer goes to the analog pin.
    side pins of the potentiometer go to +5V and ground
  * LED connected from digital pin 9 to ground

  created 29 Dec. 2008
  modified 9 Apr 2012
  by Tom Igoe

  This example code is in the public domain.

  */

// declare constants for pins used
const int potPin = 0;    // the analog input pin that the potentiometer is attached to
const int ledPin = 9;    // the output pin that the LED is attached to

// setup variables for the pot value and the led on/off state
int potVal = 0;          // variable to store the value coming from the pot
int ledState = LOW;       // variable for LED state: HIGH = on, LOW = off

void setup() {
  // initialize serial communication at 9600 baud rate
  Serial.begin(9600);

  // initialize the LED pin as an output
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // read the value of the potentiometer (0 to 1023)
  potVal = analogRead(potPin);

  // convert the potentiometer value to a PWM value (0 to 255)
  int val = map(potVal, 0, 1023, 0, 255);

  // set the PWM value to the LED pin
  analogWrite(ledPin, val);

  // print the results to the serial monitor
  Serial.print("Potentiometer value: ");
  Serial.println(potVal);
}
```

Binary sketch size: 3,426 bytes (of a 32,256 byte maximum)

1 Arduino Uno on COM1



richtung oder Standort, spezielle Hardware, die mit dem Elektronengehirn Ihres Roboters zusammenarbeitet – nehmen Sie also schon einmal den Lötkolben zur Hand. Um die Daten einzulesen, reicht eine einfache serielle Verbindung zunächst aus – es genügt, das Modul pySerial mit der seriellen Schnittstelle kommunizieren zu lassen. Das Modul laden und mit dem Sensor verbinden können Sie mit dem Befehl:

```
import serial
```

Das Problem ist, dass die bloße Verbindung noch nicht sehr nützlich ist. Sie müssen als Programmierer selbst alle Verbindungsdetails festlegen wie etwa Geschwindigkeit, Bytegröße, Datenfluss; eigentlich alles. Hier ist also durchaus einige Programmierarbeit und vor allem Debugging nötig.

Wozu verwenden Sie nun die ankommenden Daten? Sie möchten sicher Antriebselemente bewegen und anderweitig Effekt auf Ihre Umgebung haben. Ob es nun Motoren zur Fortbewegung sind oder sogar ganze Körperteile wie Arme oder Beine – um Sie zu bewegen, fehlt es den Schnittstellen Ihres Rechners wahrscheinlich an Leistung. Sie brauchen somit eine zusätzliche Platine, mit der Sie Ihren Roboter schalten können.

Einer der beliebtesten Kandidaten für den Job ist sicher der Arduino. Er kommuniziert glücklicherweise mit der seriellen Schnittstelle des Rechners, pySerial kann also weiterhin verwendet werden. Sie können Befehle an den Code senden, den Sie zuvor geschrieben und auf den Arduino geladen haben, und

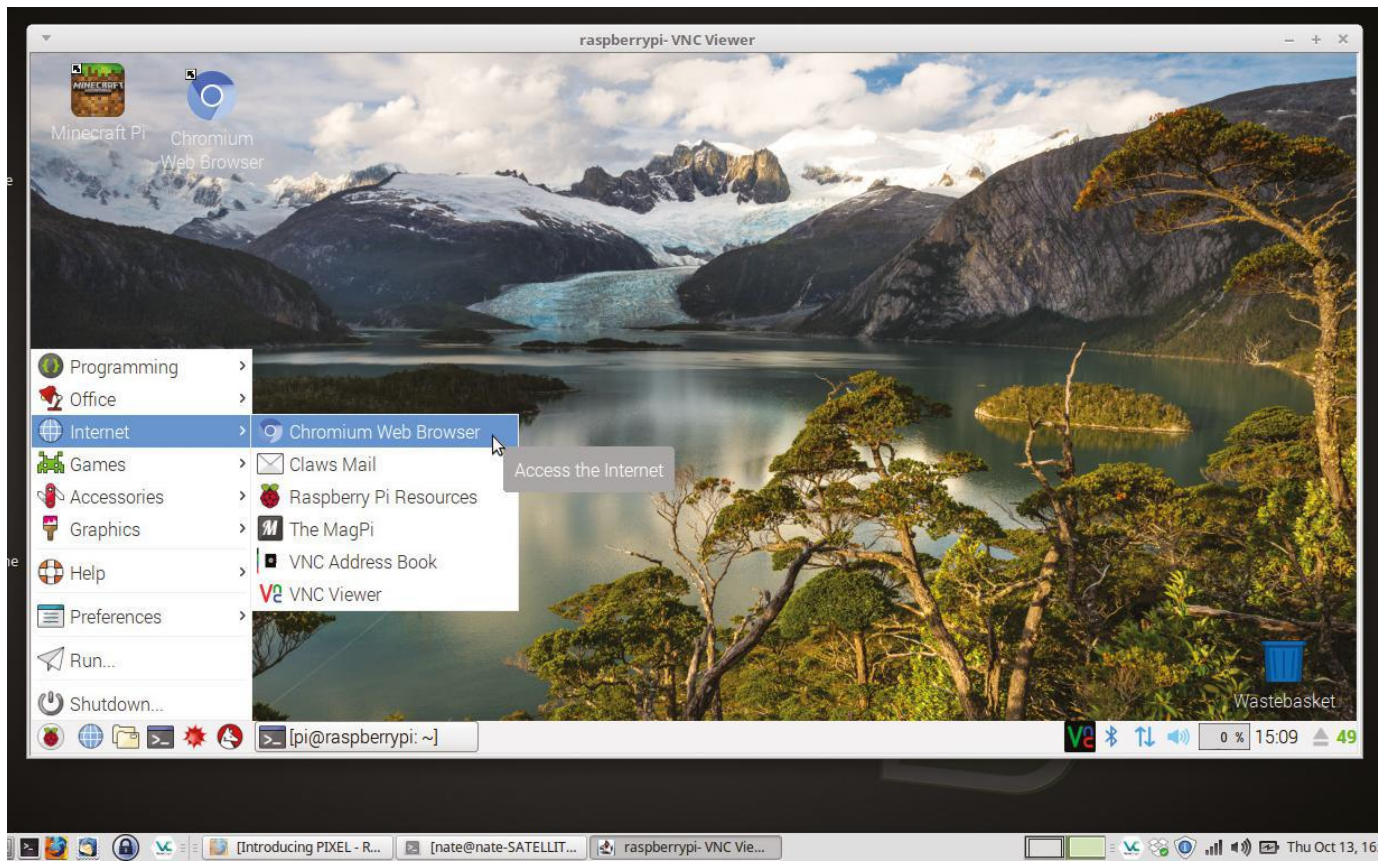
Viele Sensoren brauchen spezielle Hardware.

so Ihre Aktoren steuern. Und Sie bekommen sogar Rückmeldung, denn der Arduino gibt Feedback über die Effekte der ausgeführten Aktionen: Wurden die Räder weit genug gedreht? Der Arduino kann auch als Interface zwischen Sensoren und Computer dienen, was den Python-Code noch weiter reduziert. Es gibt zudem zahlreiche Erweiterungen, mit denen Sie möglicherweise genau die benötigten Sensoren sofort startklar in einem Paket bekommen. Wählen Sie aus den verschiedenen Versionen des Arduino diejenige, die am besten zu Ihren Anforderungen passt.

Sie können nun also Daten auslesen und darauf wiederum mit Aktionen in der realen Welt reagieren – jetzt braucht Ihr Roboter nur noch ein Gehirn. Hier kann der Stand der Technik nicht mit Fantasien von R2-D2 oder gar C-3PO mithalten. Der Großteil Ihrer Neuprogrammierung wird vermutlich in diesem Bereich stattfinden. Generell geht es dabei um die Arbeit mit AI, also Artificial Intelligence (auch künstliche Intelligenz oder KI genannt), wofür einige Projekte bereits Vorarbeit geleistet haben und als Ausgangspunkt Ihres vernunftbegabten Roboters dienen können, wie etwa SimpleAI oder PyBrain.

GIL umgehen

Die Arbeit mit Robotern kann erfordern, Code parallel auf mehreren CPUs laufen zu lassen. Python hat mit dem GIL (Global Interpreter Lock) allerdings an dieser Stelle ein Nadelöhr im Interpreter eingebaut. Eine Möglichkeit, dieses zu umgehen, ist, für jeden Thread eine eigene Interpreter-Instanz zu starten. Eine weitere wäre der Wechsel von CPython zu Jython oder IronPython, die nicht durch einen GIL gehemmt werden.



Der neue Pi-Desktop

Die grafische Benutzeroberfläche für den Raspberry Pi wurde überarbeitet. PIXEL ist ihr neuer Name.

Das offizielle Betriebssystem für den Pi, Raspbian, besitzt neben der Kommandozeilenumgebung auch einen Desktop, also eine grafische Benutzeroberfläche. Die Gestaltung des Desktops wurde grundlegend überarbeitet und kommt nun mit dem etwas sperrigen Namen „Pi Improved Xwindows Environment, Lightweight“ daher; die Kurzform ist jedoch deutlich griffiger: PIXEL.

Bereits beim Hochfahren des Rechners bemerkt man die erste Neuerung von PIXEL: Statt einer Abfolge von Bootmeldungen wird ein schlichter Splash Screen (Startbildschirm) mit der Pi-Himbeere angezeigt. Bei der Benutzeroberfläche selbst haben Sie die Wahl zwischen 16 Hintergrundbildern, darunter Landschaftsaufnahmen. Die Icons im Dateimanager und in der Taskleiste sehen schärfer und professioneller aus. Die Menüs sind sauberer und übersichtlicher gestaltet. Die Fenster haben nun leicht abgerundete Ecken.

Unter der Haube ist der neue Pi-Desktop direkt mit dem RealVNC-Server gekoppelt, damit Sie VNC einfach aus dem Schnittstellenmenü auswählen und

sich per Viewer verbinden können. Überdies ist eine provisorische Version des Google-Browsers Chromium integriert, die in Kombination mit dem Plugin h264ify zum Video-Streaming genutzt werden kann.

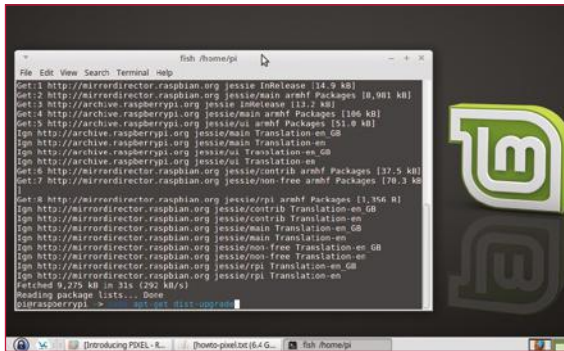


01 Installationsmethode wählen

Sie haben mehrere Möglichkeiten, die Raspbian-Version zu installieren, die den PIXEL-Desktop enthält. Falls Sie den NOOBS-Installer benutzen, können Sie beim Hochfahren des Pi die Shift-Taste gedrückt halten und Raspbian neu installieren. Alternativ können Sie das aktuelle Raspbian-Image von raspberrypi.org/downloads/raspbian/ herun-

Was Sie brauchen

- **Raspberry Pi**
(am besten Generation 2 oder 3, falls Chromium für Video-Streaming genutzt werden soll)
- **microSD-Karte mit 8 GB**
(falls Raspbian Jessie mit PIXEL von Grund auf neu installiert werden soll)
- **Zugang zu einem weiteren Computer with Pixel**
(falls eine SSH-Verbindung benötigt wird)



terladen. Falls Sie eine dieser beiden Optionen wählen, gehen Sie direkt weiter zu Schritt 05.

02 Manuelles Upgrade, sofern erforderlich

Haben Sie Raspbian Jessie bereits ohne PIXEL installiert, können Sie ein Upgrade auf die neueste Version durchführen. Öffnen Sie hierzu das Terminal oder starten Sie eine SSH-Verbindung und geben Sie Folgendes ein:

```
sudo apt-get update
sudo apt-get dist-upgrade
```

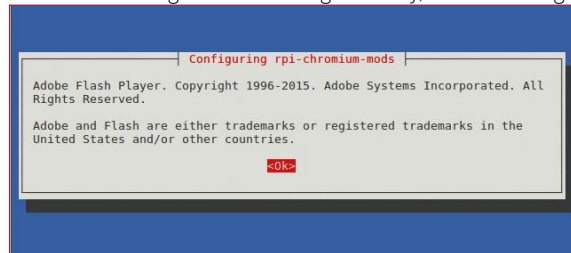
Die Aktualisierung wird einige Zeit in Anspruch nehmen. Lehnen Sie im Verlauf die Nachricht betreffend das Plymouth-Framework mit „Q“ ab.

03 Chromium installieren

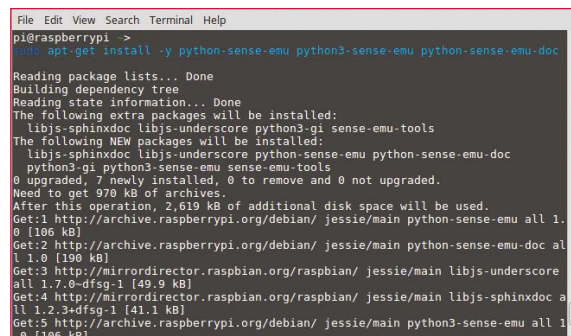
Die neue Raspbian-Version enthält eine Vorabversion des Browsers Chromium. Im Rahmen der manuellen Aktualisierung können Sie diese so hinzunehmen:

```
sudo apt-get install -y rpi-chromium-mods
```

In Chromium integriert ist das Plugin h264ify, das dafür sorgt,



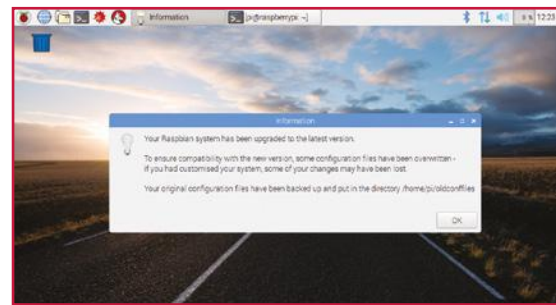
dass YouTube beim Streamen von Videos die Hardwarebeschleunigung des Pi verwendet. Ebenfalls enthalten ist der sehr gute Werbeblocker uBlock Origin.



04 Sense HAT Emulator installieren

Wenn Sie möchten, können Sie den Sense HAT Emulator installieren, der ebenfalls in die aktuelle Version von Raspbian aufgenommen wurde. Das Sense HAT ist eine Zusatzplatine für den Raspberry Pi, die mit diversen Sensoren ausgestattet ist, darunter ein Thermometer, ein Barometer und ein Kreiselinstrument. Mithilfe des Emulators lässt sich Code für die Zusatzplatine testen, ohne diese tatsächlich zur Hand zu haben. Geben Sie zur Installation ein:

```
sudo apt-get install -y python-sense-emu python3-sense-emu python-sense-emu-doc
```

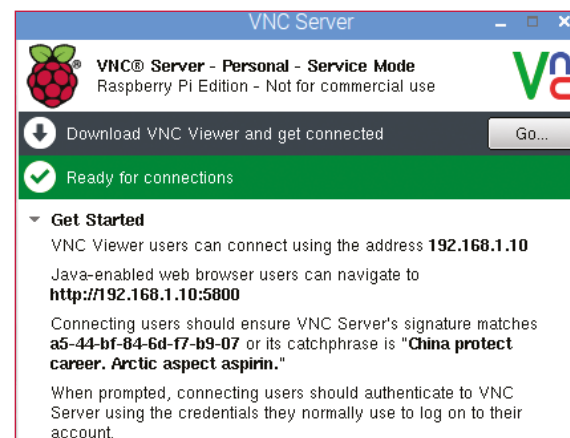


05 PIXEL-Desktop laden

Führen Sie einen Neustart des Pi durch, um PIXEL zu laden. Eventuell wird Ihnen eine Nachricht angezeigt, dass Ihre bisherigen Konfigurationsdateien gelöscht wurden. Bestätigen Sie dies mit „OK“. An dieser Stelle können Sie auch das Hintergrundbild ändern, wenn Sie möchten: Rechtsklicken Sie dazu irgendwo auf den Desktop und wählen Sie „Desktop Preferences“, dann das Wallpaper-Menü aus.

06 Schnittstellen festlegen

Bluetooth und WLAN lassen sich direkt vom Desktop aus deaktivieren; klicken Sie hierzu auf die zugehörigen Symbole. VNC können Sie ebenso leicht aktivieren: Gehen Sie im Menü zu Preferences > Raspberry Pi Configuration > Interfaces und klicken Sie „Enabled“ beim Punkt VNC. Daraufhin erscheint der VNC-Button auf dem Desktop. Ein Klick darauf offenbart die private IP-Adresse des Pi für VPN-Clients.



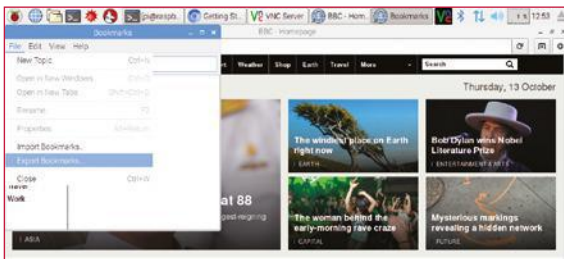
07 Vorhandene Lesezeichen in Chromium übertragen

Öffnen Sie Chromium per Klick auf das zugehörige Icon auf dem Desktop. Wenn Sie Ihre Lesezeichen aus dem Epiphany-Browser importieren möchten, tippen Sie ins Terminal:

```
epiphany-browser.
```

Versteckte Icons

Einige der neu gestalteten Icons werden nicht von vornherein angezeigt, da keine der Standardanwendungen den Bereichen Entwicklung und Lernen zugehörig sind. Falls Sie neugierig sind, gehen Sie einfach zu /usr/share/icons/PiX, dort finden Sie alle Icons. Alternativ können Sie auch im Desktop-Menü den Punkt Preferences > Add/Remove software aufrufen, um die Kategorien und deren jeweilige Symbole zu sehen. Für einzelne Apps können die Icons auch modifiziert werden (siehe Schritt 15).



Klicken Sie das Symbol für die Einstellungen oben rechts und wählen Sie „Edit Bookmarks“. Im sich daraufhin öffnenden Fenster haben Sie die Möglichkeit, Ihre Lesezeichen im HTML-Format zu exportieren. Schließen Sie Epiphany und klicken Sie in Chromium den blauen Link „Import Bookmarks“, dann werden die Lesezeichen hinzugefügt.

08 Desktop anpassen

Der Desktop ist in der Standardeinstellung schon ziemlich gut eingerichtet, aber falls Sie kleine Veränderungen vornehmen möchten, geht das auch. Rufen Sie im Hauptmenü den Bereich Preferences > Appearance Settings auf. Sollten Sie beispielsweise an eine Menüleiste am unteren Bildschirmrand gewöhnt sein, legen Sie die Position der Leiste entsprechend fest. Auch die Größe der Menüleiste lässt sich anpassen.



09 Sense HAT Emulator testen

Wenn Sie den Sense HAT Emulator mit dem Gesamtpaket oder in Schritt 04 separat installiert haben, können Sie ihn aus dem Bereich Programmieren im Anwendungsmenü heraus starten. Der Emulator enthält etwa ein Dutzend Beispiel-Skripte, um Ihnen den Einstieg zu erleichtern. Die Zusatzplatine selbst, also das Sense HAT, ist für etwa 35

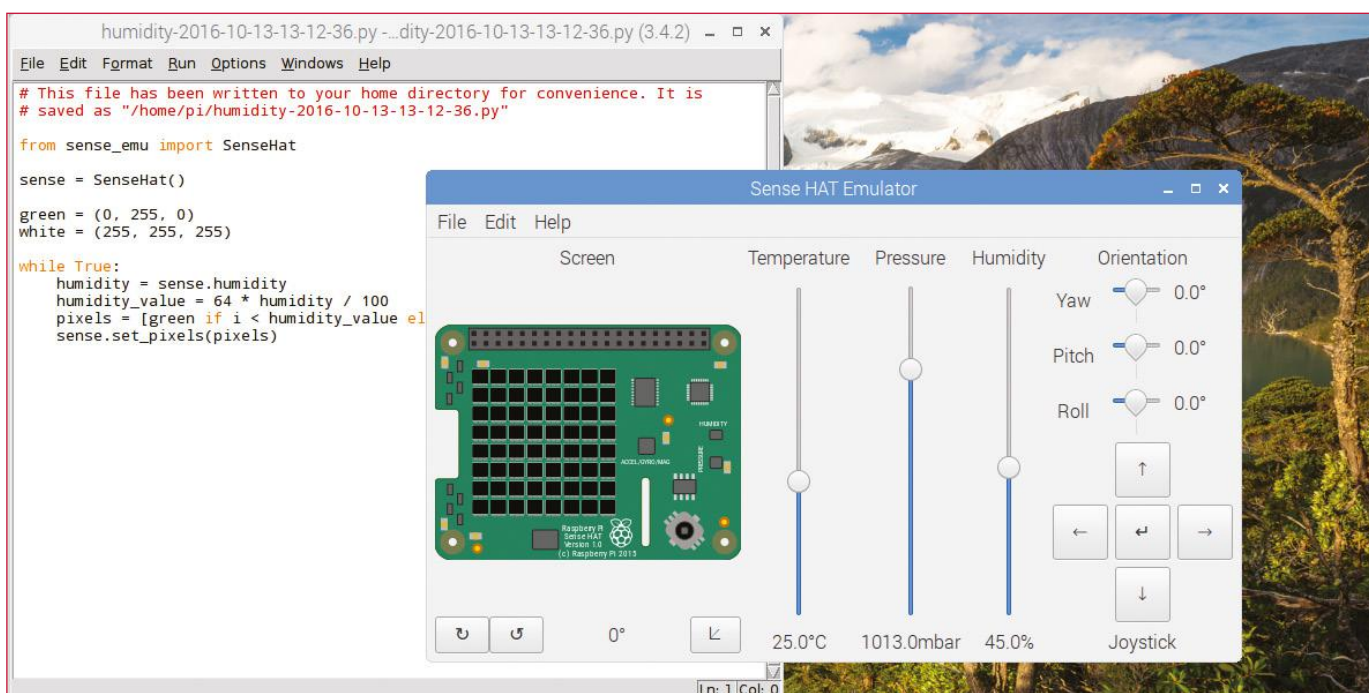
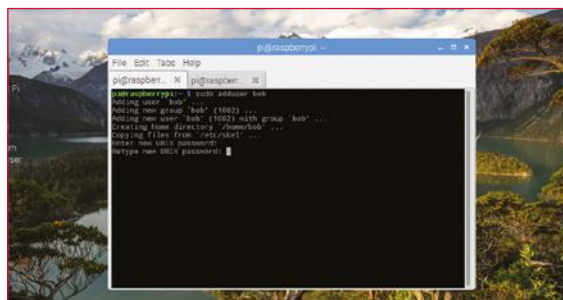
Euro auf thepihut.com/products/raspberry-pi-sense-hat-astro-pi zu haben.

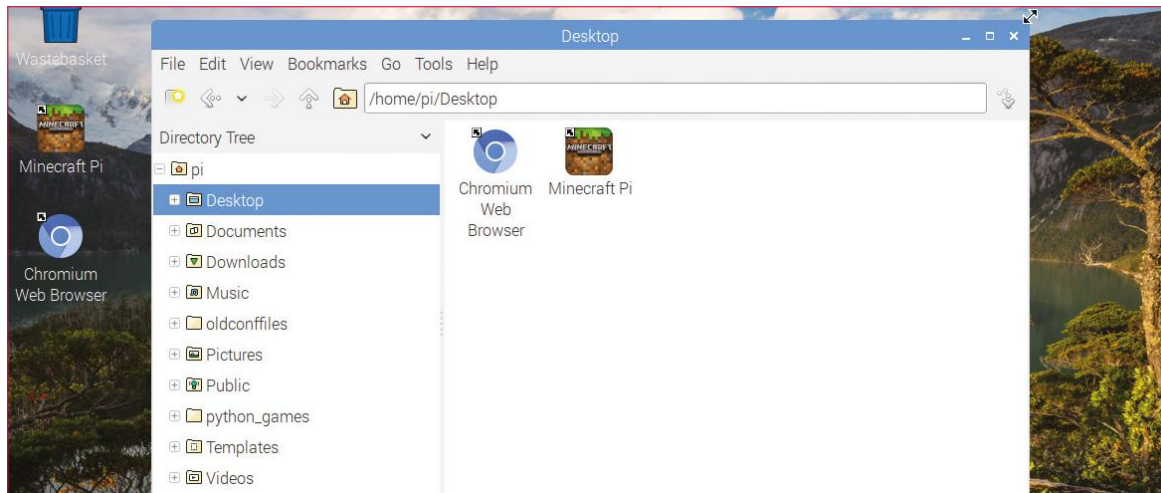
10 Login-Bildschirm aktivieren

Nun da Ihr Raspberry Pi eine reichhaltige grafische Benutzeroberfläche und einen modernen Webbrowser vorweisen kann, möchten Sie ihn womöglich als Arbeits-PC oder Heimcomputer verwenden. In diesen Fällen empfiehlt sich ein Passwortschutz. Um in den Bootvorgang eine Login-Abfrage zu integrieren, öffnen Sie zunächst folgende Datei:

sudo nano /etc/lightdm/lightdm.conf

Suchen Sie darin die Zeile **autologin-user=pi** und setzen Sie ein # davor. Drücken Sie anschließend Strg+X, Y und Enter, um zu speichern und zu schließen.





11 Nutzer hinzufügen

Nun sollten Sie noch das Standardpasswort „raspberry“ löschen und ein eigenes vergeben. Öffnen Sie hierzu das Terminal oder eine SSH-Verbindung und tippen Sie:

```
passwd
```

Wenn Sie die Login-Abfrage wie beschrieben umgesetzt haben und außer Ihnen noch andere Personen Zugang zum Pi haben sollen, dann richten Sie am besten weitere Benutzerkonten ein. Das Kommando dafür lautet:

```
sudo adduser name
```

Ersetzen Sie dabei „name“ durch den tatsächlichen Benutzernamen. Danach geben Sie zweimal das Passwort für den neuen Benutzer ein, jeweils nach Aufforderung.

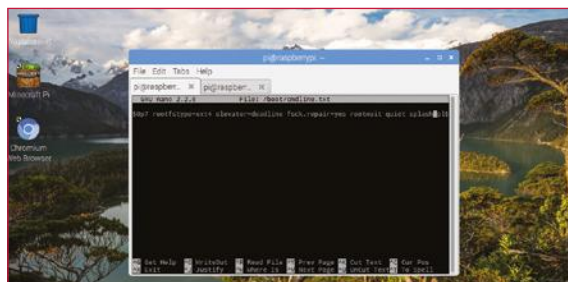
12 Größenanpassung der neuen Fenster

Das neue Design der Fenster ist zwar stilvoller, macht es aber möglicherweise anfangs etwas schwieriger, deren Größe zu justieren, wenn man die dicken Fensterrahmen des früheren Raspbian-Desktops gewohnt ist. Immerhin wird durch die Cursorform ziemlich präzise angezeigt, wann sich der Mauszeiger über dem Fenster befindet und wann außerhalb.

13 Splash Screen deaktivieren

Der Splash Screen von PIXEL mag nicht jedem zusagen. Falls Sie lieber die früher übliche Abfolge von Bootmeldungen wiederherstellen möchten, geben Sie das folgende Kommando im Terminal ein:

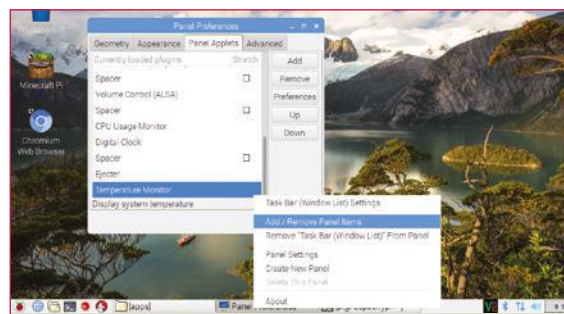
```
sudo nano /boot/cmdline.txt
```



Benutzen Sie die Pfeiltaste nach rechts, um den Text „quiet splash“ zu finden. Löschen Sie diesen. Speichern und schließen Sie die Datei.

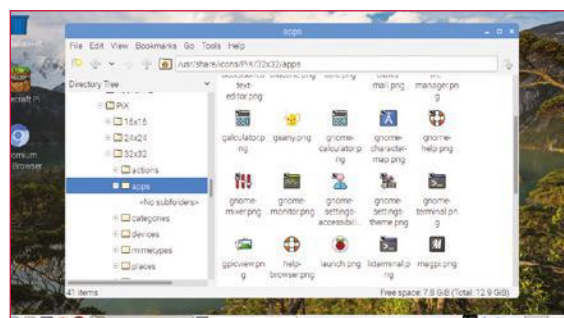
14 Spannungs- und Temperaturwächter

Bei früheren Versionen von Raspbian erschienen manchmal gelbe und rote Quadrate am Rand des Bildschirms, wenn der Pi überhitzte oder zu wenig Strom bekam. Stattdessen werden bei der neuen Version des Betriebssystems Warnsymbole in Form eines Thermometers beziehungsweise eines Blitzes angezeigt. Nach einem Rechtsklick auf die Taskleiste können Sie bei Bedarf festlegen, dass Spannung und Temperatur des Systems angezeigt werden.



15 Icons begutachten

Sam Alder und Alex Carter haben die Desktop-Icons für PIXEL sorgfältig überarbeitet. Sie finden alle Systemsymbole in `/usr/share/icons/PIX`. Wenn Sie ein Icon für ein bestimmtes Programm ändern möchten, dann erstellen Sie eine neue PNG-Grafik und versehen Sie sie mit dem identischen Namen wie das zu ersetzende Icon.



Multitasking mit dem Pi

Wie Sie mit Python Multitasking betreiben –
perfekt, um mehrere Projekte gleichzeitig durchzuführen.

Warum Python?

Das ist die offiziell unterstützte Programmiersprache für den Raspberry Pi. Unter python.org/doc finden Sie eine umfassende Dokumentation, Tutorials und weitere Hilfen.

Die meisten Programmierer lernen das Single-Threading als ihr erstes Rechenmodell kennen. Beim Single-Threading werden die Befehle vom Computer sequenziell, also einer nach dem anderen, abgearbeitet. Das funktioniert meistens ganz gut, doch irgendwann ist der Punkt erreicht, an dem man mit Multitasking beginnen sollte.

Klassischerweise werden Anwendungen mit Multi-Threading geschrieben, wenn Sie auf einem leistungsstarken Multi-Prozessor-Computer ausgeführt werden. In diesen Fällen läuft auf jedem Prozessor ein schwerer, rechenintensiver Prozess. Da der Raspberry Pi keine große 16-kernige Desktopmaschine ist, würde man vielleicht annehmen, dass man mehrere Threads nicht gleichzeitig durchführen kann. Aber das geht.

Allerdings gibt es viele Probleme, die es beim Multi-Threading zu beachten gilt. Eventuell brauchen I/O-Operationen relativ lange in der Durchführung. In diesem Fall lohnt es sich, Ihr Problem in mehrere Threads aufzubrechen und diese parallel laufen zu lassen. Da Python die bevorzugte Programmiersprache für den Pi ist, werden wir uns ansehen, wie Sie Threads zu Ihrem eigenen Python-Code hinzufügen können. Wenn Sie schon einmal in Python mit Multi-Threading programmiert haben, sind Sie eventuell schon auf GIL (Global Interpreter Lock) gestoßen. GIL bedeutet, dass tatsächlich immer nur ein Thread bearbeitet wird und keine wirkliche Parallelisierung vorliegt. Aber auf dem Pi ist das in Ordnung. Es kommt uns nur darauf an, auf bestimmte Probleme ein sinnvolles Programmierparadigma anzuwenden.

Mit dem ersten Befehl importieren wir das richtige Modul. In diesem Artikel wenden wir das Modul `threading` an. Nach dem Import haben Sie Zugriff auf alle Funktionen und Objekte, die Sie für das Schreiben des Codes brauchen. Als Erstes erschaffen wir ein neues Thread-Objekt:

```
t = threading.Thread(target=my_func)
```

Das Thread-Objekt übernimmt als auszuführendes Zielcode eine Funktion, die Sie erzeugt haben (im obigen Beispiel `my_func`). Nach seiner Initialisierung existiert das Thread-Objekt zwar, aber es läuft noch nicht. Dafür müssen Sie explizit die `start()`-Methode des Threads aufrufen. Dadurch beginnt der Code innerhalb der dem Thread übergebenen Funktion zu laufen.

Sie können überprüfen, ob der Thread existiert und aktiv ist, indem Sie die `is_alive()`-Methode aufrufen. Normalerweise läuft dieser neue Thread so lange, bis die Funktion endet. Andererseits können Threads auch enden, wenn eine unbehandelte Ausnahme auftritt. Je nachdem, wie versiert Sie mit parallel laufenden Programmen sind, haben Sie vielleicht schon eine Ahnung davon, welche Arten von Code Sie schreiben möchten. Zum Beispiel wird in MPI-Programmen üblicherweise derselbe Code auf mehreren Threads durchgeführt. Mithilfe der Thread-ID und einer Reihe von if-oder case-Anweisungen wird festgelegt, welcher Thread welchen Teil des Codes ausführt. Um etwas Derartiges durchzuführen, könnten Sie folgendermaßen vorgehen:

```
def my_func():
    id = threading.get_ident()
    if (id == 1):
        do_something()
thread1 = threading.Thread(target=my_func)
thread1.start()
```

Dieser Code funktioniert in Python 3, nicht jedoch in Python 2, da es hier die `get_ident()`-Funktion nicht gibt. Das Threading gehört zu jenen Modulen, die sich stark verändern, wenn man von einer Python-Version auf eine andere umsteigt. Werfen Sie diesbezüglich also sicherheitshalber einen Blick in die Dokumentation Ihrer Python-Version.

Beim parallelen Programmieren werden oft zeitintensive I/Os an mehrere separate Threads vergeben. So kann sich Ihr Hauptprogramm weiterhin um die Hauptarbeit kümmern, und alle Rechenressourcen werden maximal genutzt. Aber wie finden Sie heraus, ob der untergeordnete Thread bereits zu Ende ist oder nicht?

Sie können die oben genannte `is_alive()`-Funktion dafür nutzen. Doch was passiert, wenn Sie ohne die Ergebnisse des untergeordneten Threads nicht weitermachen können? In diesen Fällen können Sie die `join()`-Methode auf das Objekt anwenden, auf das Sie warten. Diese Methode blockiert so lange, bis der betreffende Thread zurückkehrt. Sie können einen zusätzlichen Parameter hinzufügen, damit die Methode nach einer bestimmten Zeit ausläuft. Auf diese Weise werden Sie nicht in einem Thread gefangen sein, der aufgrund eines Fehlers oder Code-Bugs nie wieder zurückkehrt.

Da nun mehrere Threads parallel durchgeführt werden, taucht eine Reihe neuer Probleme auf, über die wir uns Gedanken machen müssen. Das erste Problem stellt der Zugriff auf globale Datenelemente dar. Was passiert, wenn zwei verschiedene Threads gleichzeitig dieselbe Variable im globalen Speicher auslesen oder sogar überschreiben wollen? Es können sich Situationen ergeben, in denen die Veränderungen der Werte nicht mehr aufeinander abgestimmt sind und sich somit unerwartete Ergebnisse einstellen.

Dieses Problemfeld wird Wettlaufsituation genannt, da unterschiedliche Threads darum wetten, in welcher Reihenfolge sie Variablen updaten. Für dieses Problem gibt es zwei Lösungen. Die erste besteht darin, den Zugriff auf die globalen Variablen zu kontrollieren und immer nur einem Thread zu erlauben, mit der Variablen zu arbeiten. Meist wird hierfür ein wechselseitiger Ausschluss verwendet, auch Mutex genannt. Ein Mutex-Objekt muss von einem Thread blockiert werden, bevor dieser mit den dazugehörigen Variab-



len arbeiten kann. Im Threading-Modul von Python wird dieses Objekt „lock“ genannt. Zunächst wird ein neues Lock-Objekt erschaffen:

```
lock = threading.Lock()
```

Dieses neu erschaffene Objekt ist noch nicht gesperrt und daher frei verfügbar. Der Thread, der das Objekt nutzen möchte, muss dessen `acquire()`-Methode aufrufen. Wenn es verfügbar ist, dann ändert es seinen Status auf gesperrt, und Ihr Thread kann ungestört zugreifen. Wenn das Objekt bereits gesperrt ist, dann wartet Ihr Thread so lange, bis das Objekt wieder frei ist. Sobald Ihr Thread mit der Bearbeitung des geschützten Codes fertig ist, müssen Sie die `release()`-Methode ausführen, um das Objekt für die nächsten Threads zu entsperren. Beispielsweise könnten Sie eine Variable, die die Summe einer Reihe von Ergebnissen beinhaltet, mit folgendem Code kontrollieren:

```
lock.acquire()
sum_var += curr_val
lock.release()
```

Dies kann zu einem weiteren bekannten Problem von parallelen Programmen führen, nämlich Deadlocks (Verklemmungen). Diese treten auf, wenn Sie mehrere Sperren haben, die unterschiedlichen globalen Variablen zugeordnet sind. Nehmen wir an, Sie haben die Variablen A und B sowie die zugehörigen Sperr-Objekte lockA und lockB. Wenn Thread Nr. 1 versucht, zunächst auf lockA und dann auf lockB zuzugreifen, während Thread Nr. 2 versucht, zunächst lockB und dann lockA zu bekommen, dann könnte es Ihnen passieren, dass beide ihre ersten Objekte bekommen und dann ewig auf die zweiten warten.

Am besten vermeiden Sie diesen Bug, indem Sie Ihr Programm sehr achtsam erstellen. Leider sind Programmierer aber auch nur Menschen und kann sich chaotischer Code einschleichen. Sie können versuchen, diesen Fehler abzufangen, indem Sie beim Aufrufen der `acquire()`-Methode eine optionale Zeitbeschränkung hinzufügen. Somit wird nur für eine bestimmte Anzahl Sekunden versucht, auf das Objekt zuzugreifen. Nach Ablauf der Zeit kehrt die `acquire()`-Methode zurück. Durch Überprüfung des ausgegebenen Wertes können Sie feststellen, ob die Methode erfolgreich war. Falls ja, dann wird als Ergebnis „True“ zurückkehren. Falls nicht, dann ist das Ergebnis „False“.

Zweitens können Sie den Datenzugriff regeln, indem Sie alle verschiebbaren Variablen in den jeweiligen Bereich einzelner Threads bewegen. Die Grundidee besteht darin, dass jeder Thread seine eigene lokale Version von jeder benötigten Variable besitzt, die außer ihm keiner sehen kann. Hierfür wird ein lokales Objekt erstellt, dem Sie Attribute zuordnen können, die Sie wiederum als lokale Variablen verwenden. Innerhalb der Funktion, die von Ihrem Thread ausgeführt wird, hätten Sie folgenden Code:

```
my_local = threading.local()
my_local.x = 42
```

Das letzte Thema, das wir hier behandeln, ist die Synchronisierung Ihrer Threads, damit diese effektiv zusammenarbeiten können. Mehrere Threads müssen miteinander reden, nachdem Sie Ihren jeweiligen Teil eines Problems gelöst haben. Threads können Ergebnisse nur miteinander teilen, wenn alle ihre jeweiligen Ergebnisse fertig berechnet haben. Man kann dieses Problem mit einer Barriere lösen, an welcher jeder Thread stoppt, bis alle anderen auch an dieser Stelle angekommen sind. Bei Python 3 gibt es ein Barriere-Objekt, welches für eine bestimmte Anzahl an Threads erschaffen werden kann. Es erzeugt einen Punkt, an dem Threads anhalten, wenn Sie die `wait()`-Methode der Barriere aufrufen.

Dies kann jedoch eine Fehlerquelle sein, weil Sie der Barriere genau sagen müssen, wie viele Threads an ihr anhalten sollen. Wenn Sie fünf Threads anlegen, Ihre Barriere aber zehn Threads erwartet, dann werden Sie nie den Punkt erreichen, an dem die Barriere gelöst wird.

Ein weiteres Werkzeug zur Synchronisierung ist das Timer-Objekt. Ein Timer ist eine Untergruppe der Thread-Kategorie und führt nach Verstreichen einer bestimmten Zeit eine Funktion aus. Wie bei einem Thread müssen Sie zunächst die `start()`-Methode des Timers aufrufen, um den Countdown zu starten, nach dessen Ablauf die Funktion ausgeführt wird. Die neue `cancel()`-Methode ermöglicht es Ihnen, den Countdown des Timers anzuhalten, wenn er noch nicht bei null angelangt ist.

Nun sollten Sie in der Lage sein, Ihren eigenen Code effizienter ausführen zu lassen, indem Sie zeitintensive Teile zur Durchführung an andere Threads übergeben. Auf diese Weise kann der Hauptteil Ihres Programms so schnell wie möglich auf Interaktion mit dem Benutzer reagieren, während Sie gleichzeitig alle Teile Ihres Raspberry Pi maximal auslasten.

Prozesse oder Threads?

Was aber, wenn Sie tatsächlich parallel arbeitenden Code brauchen, der auf mehreren Kernen gleichzeitig läuft? Da Python GIL hat, müssen Sie sich von Threads verabschieden und zur Verwendung von separaten Prozessen übergehen, um unterschiedliche Aufgaben zu erfüllen. Zum Glück verfügt Python über ein Multiprocessing-Modul, das dem Threading-Modul für Prozesse entspricht. Wie beim Threading-Modul können Sie Prozessobjekte erstellen und ihnen eine auszuführende Zielfunktion übergeben. Anschließend müssen Sie die `start()`-Methode aufrufen, damit der Prozess beginnt. Bei Threads ist das Teilen von Daten einfach, weil der Speicher global ist und jeder alles sehen kann.

Unterschiedliche Prozesse befinden sich aber in unterschiedlichen Speicherbereichen. Um Daten auszutauschen, müssen wir ausdrücklich eine bestimmte Art der Kommunikation einrichten. Sie können eine Warteschlange erzeugen, auf welche Sie Objekte übertragen können. Prozesse speichern mit der `put()`-Methode Objekte auf die Warteschlange. Andere Prozesse ziehen über die `get()`-Methode Objekte aus der Warteschlange heraus. Wenn Sie etwas mehr Kontrolle darüber haben möchten, wer mit wem kommuniziert, dann können Sie eine Pipe erzeugen, um einen bidirektionalen Kommunikationskanal zwischen zwei Prozessen einzurichten. Wenn Sie Pipes oder Warteschlangen nutzen, müssen Sie diese als Argumente an Ihre Zielfunktion übergeben.

Sie können Informationen auch teilen, indem Sie einen gemeinsamen Speicherbereich schaffen. Man kann mit dem Wertobjekt einen einzigen variablen Speicherplatz zum Teilen von Informationen erstellen. Wenn Sie mehrere Variablen übergeben wollen, können Sie diese in ein Feldobjekt legen. Wie bei Pipes und Warteschlangen müssen diese als Parameter an Ihre Zielfunktion übergeben werden. Wenn Sie auf die Ergebnisse eines Prozesses warten müssen, verwenden Sie die `join()`-Methode, damit der Hauptprozess wartet, bis der untergeordnete Prozess abgeschlossen ist.

Anders als das Threading-Modul beinhaltet das Prozess-Modul auch das Konzept des Prozesspools. Mit einem Pool können mehrere Prozesse im Vorfeld erzeugt werden, die im Anschluss in einer Map-Funktion genutzt werden können. Dieses Vehikel ist hilfreich, wenn Sie die gleiche Funktion auf mehrere verschiedene Inputwerte anwenden. Wenn Sie mit Konzepten wie Map-Funktionen oder der Anwendung von Funktionen aus R oder Hadoop vertraut sind, liegt Ihnen dies eventuell für den Einsatz in Ihrem Python-Code mehr.

Wie der Raspberry Pi auf Ihr Baby aufpasst

Mit diesem Baby-Monitor haben Sie den schlafenden Nachwuchs immer im Blick.

Die nächste Staffel Ihrer Lieblingsserie liegt bereit, doch Sie können sich nicht so richtig entspannen – schläft das Baby, oder liegt es wach oder gar weinend in seinem Bettchen? Die Tonspur aus dem Surroundsystem macht die Sache nicht einfacher. Ein Baby-Monitor ist die Lösung.

Ein normales Babyfon ist nützlich, aber der Trend geht in Richtung videobasierte Lösungen, besonders in Kombination mit einer App oder einer Browserschnittstelle. Mit Bewegungsmelder und Alarmpunktion hat man ein unheimlich nützliches Instrument, dessen Aufgaben auch der Raspberry Pi problemlos und zu einem Bruchteil des Preises übernehmen kann.

Was Sie dazu brauchen, ist ein Raspberry Pi (je neuer, desto besser), eine USB-Webcam oder die Pi-Kamera (die NoIR-Version mit Infrarotunterstützung ist noch besser geeignet) sowie ein Gerät für den Videoempfang. Und schon wissen Sie immer, was Ihr Nachwuchs gerade anstellt – ein Blick aufs Smartphone genügt!

01 Schlafzimmer begutachten

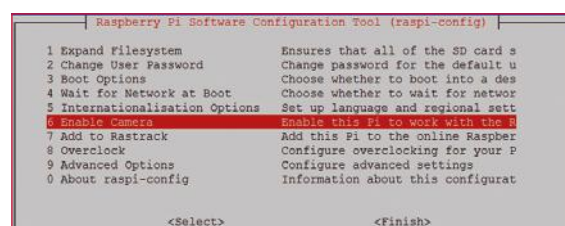
Sehen Sie sich vor der Installation und Konfiguration des Raspberry Pi das Schlafzimmer Ihres Babys genau an. Wo soll der Pi stehen, gibt es dort eine Stromquelle und einen LAN-Anschluss (oder ein ausreichend starkes WLAN-Sig-

nal)? Nehmen Sie sich Zeit, die perfekte Position für den Pi im Bezug auf Strom und Netzwerk zu finden.

02 Kamera aktivieren

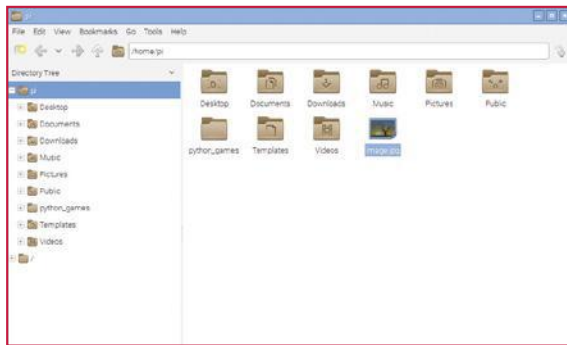
Die Pi-Kamera ist standardmäßig deaktiviert und kann mittels raspi-config aktiviert werden. Der Zugriff erfolgt über die Nutzoberfläche in Menu > Preferences > Raspberry Pi Configuration, wo Sie unter „Interfaces“ die Kamera auf „Enabled“ schalten müssen. Alternativ können Sie „sudo raspi-config“ starten und Option 6 wählen, „Enable Camera“. Für eine USB-Webcam verwenden Sie:

| `sudo apt-get install fswebcam`



Was Sie brauchen

| Pi-Kameramodul oder USB-Webcam



03 Kamera testen

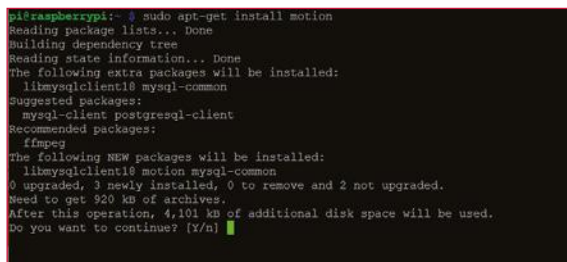
Wenn alles aufgebaut ist, aber die Kamera nicht funktioniert, ist die Enttäuschung groß. Ob Sie nun eine USB-Webcam oder die Pi-Kamera verwenden, Sie sollten das verwendete Gerät zunächst testen. Um das Testbild zu betrachten, navigieren Sie per Dateimanager ins Verzeichnis /home/pi und öffnen Sie image.jpg. So können Sie prüfen, ob das Bild korrekt gespeichert wurde.

04 Motion installieren

Die Motion-Capture-Software motion kann nach einem Update und Upgrade von Raspbian installiert werden.

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install motion
```

Ältere Versionen der Software starten nicht automatisch und zeigen die Fehlermeldung „Not starting motion daemon“ an. Beugen Sie dem vor, indem Sie zunächst update und upgrade ausführen. Das Pi-Kameramodul braucht zudem noch einen Treiber.



05 Treiber für Pi-Kamera aktivieren

Verwenden Sie folgenden Code, um den Kamerateiber zu aktivieren:

```
sudo modprobe bcm2835-v4l2
```

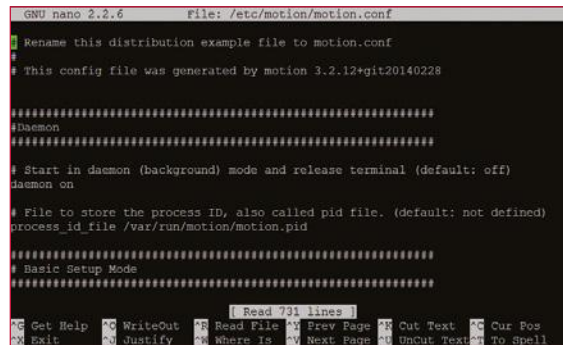
Fortan kann die Pi-Kamera mit Drittsoftware wie motion kommunizieren. Der Treiber muss bei jedem Neustart neu geladen werden, außer man fügt ihn der Datei rc.local hinzu. Öffnen Sie sie in nano:

```
sudo nano /etc/rc.local
```

Tragen Sie in eine freie Zeile vor dem Befehl „exit“ Folgendes ein:

```
modprobe bcm2835-v4l2
```

Beenden Sie mit Strg+X und speichern Sie mit „Y“.



06 Autostart von motion

Öffnen Sie dazu die Konfigurationsdatei von motion:

```
sudo nano /etc/default/motion
```

Das Programm soll beim Hochfahren des Raspberry Pi automatisch starten. Suchen Sie den Wert „daemon off“ und ändern Sie ihn in:

```
daemon on
```

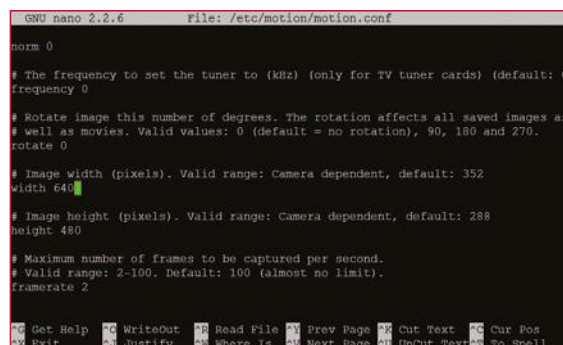
Beenden Sie mit Strg+X, speichern Sie mit „Y“, und bestätigen Sie mit der Eingabetaste. Starten Sie das System neu, um die Autostartfunktion zu testen.

```
sudo reboot
```

07 Motion konfigurieren

Konfigurieren Sie nun motion. Passen Sie die Frame-rate (wie oft Bilder aufgenommen werden), die Auflösung (eine höhere Auflösung erfordert mehr Rechenleistung, was den Baby-Monitor langsamer macht) und das Videoformat ein.

Die optimale Konfiguration hängt von Ihrem Raspberry Pi ab. Geräte der ersten Generation kommen mit niedriger Auflösung gut zurecht; für hochauflösende Bilder brauchen Sie wahrscheinlich einen Raspberry Pi 3.



08 Konfigurationsdatei editieren

Öffnen Sie also zunächst motion.conf mit:

```
sudo nano /etc/motion/motion.conf
```

Mit Strg+W starten Sie eine Volltextsuche. Passen Sie bei allen unten angegebenen Parametern die Werte wie folgt an:

```
daemon on
framerate 2
```

Welche Kamera?

Raspberry Pi-Projekte können sowohl mit dem Pi-Kameramodul als auch einer normalen USB-Kamera durchgeführt werden. Der Unterschied liegt in der problemlosen Treibersituation der Pi-Kamera, die bei USB-Webcams komplizierter sein kann. Ihre Wahl hängt letztlich von der Art des Projekts ab. Die Pi-Kamera ist schwerer zu positionieren, was manchmal für eine USB-Kamera mit langem Kabel spricht.



```
width 640
height 480
ffmpeg_video_codec mpeg4
stream_localhost off
control_localhost off
```

Ist es im Zimmer sehr dunkel, können Sie noch Kontrast (contrast) und Helligkeit (brightness) anpassen. Beenden Sie anschließend mit **Strg+X** und speichern Sie mit „Y“.

```
GNU nano 2.2.6      File: /etc/motion/motion.conf

# D = changed pixels, N = noise level,
# I and J = width and height of motion area,
# K and L = X and Y coordinates of motion center
# C = value defined by text event
# Quotation marks round string are allowed.
#####

# Target base directory for pictures and films
# Recommended to use absolute path. (Default: current working directory)
target_dir /var/lib/motion

# File path for snapshots (jpeg or ppm) relative to target_dir
# Default: %v-%Y%m%d%H%M%S-snapshot
# Default value is equivalent to legacy oldlayout option
# For Motion 3.0 compatible mode choose: %Y/%m/%d/%H/%M/%S-snapshot
# File extension can be automatically added so do not include this.
# Note: A symbolic link called lastsnap.jpg created in the target dir will always
# point to the latest snapshot, unless snapshot_filename is exactly 'lastsnap'
snapshot_filename %v-%Y%m%d%H%M%S-snapshot

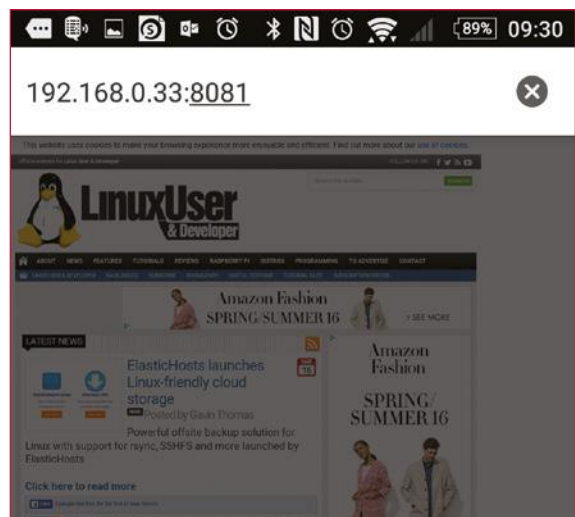
# Get Help  # WriteOut  # Read File  # Prev Page  # Cut Text  # Cur Pos
```

09 Berechtigungen anpassen

Es kann leider vorkommen, dass die Kamera nach kurzer Zeit keine Bilder mehr übergibt. Das liegt an fehlenden Berechtigungen für das Verzeichnis, die nach der Anzeige einiger Bilder für einen Timeout-Fehler sorgen. So lösen Sie das Problem:

```
sudo chown motion: /var/lib/motion
```

Sie können auch in der `motion.conf` einen eigenen Pfad eintragen. Ersetzen Sie „`target_dir`“ durch das gewünschte Verzeichnis. Speichern Sie die Datei und starten Sie `motion` neu.



10 Baby-Monitor starten und testen

10 Das Video-Babyfon ist nun fast einsatzbereit. Jetzt muss nur noch motion gestartet werden:

```
sudo service motion start
```

Den Videostream sollten Sie abrufen können, indem Sie im Browser Ihres Mobilgeräts die IP-Adresse des Raspberry Pi aufrufen. Typischerweise sieht diese etwa so aus:

192.168.0.10:8081

Prüfen Sie die Funktion am besten mit mehreren Geräten aus Ihrem Heimnetzwerk.

```
GNU nano 2.2.6 File: /etc/motion/motion.conf

# Number of frames to skip before each roundrobin step (default: 1)
roundrobin_skip 1

# Try to filter out noise generated by roundrobin (default: off)
switchfilter off

#####
# Motion Detection Settings:
#####

# Threshold for number of changed pixels in an image that
# triggers motion detection (default: 1500)
threshold 1500

# Automatically tune the threshold down if possible (default: off)
threshold_tune off

[ Search Wrapped ]
^G Get Help ^O WriteOut ^R Read File ^M Prev Page ^K Cut Text ^C Cur Pos
```

11 Bewegungserkennung einstellen

Die Bewegungserkennung wird nicht gleich beim ersten Versuch richtig funktionieren. Um sie für die jeweilige Umgebungssituation anzupassen, öffnen Sie folgende Datei:

```
sudo nano /etc/motion/motion.conf
```

Suchen Sie darin mit Strg+W nach „Motion Detection Settings“. Hier finden Sie einige justierbare Parameter wie „threshold“ oder „area_detect_value“. Testen Sie verschiedene Werte, bis das Ergebnis Sie zufriedenstellt.

12 Alarmton

12 Sie können einen Alarmton bei erkannter Bewegung aktivieren. Es lohnt sich, das Projekt so genau wie möglich zu kalibrieren. Die Einstellung finden Sie ebenfalls in der Datei `motion.conf`. Ändern Sie darin „quiet on“ in:

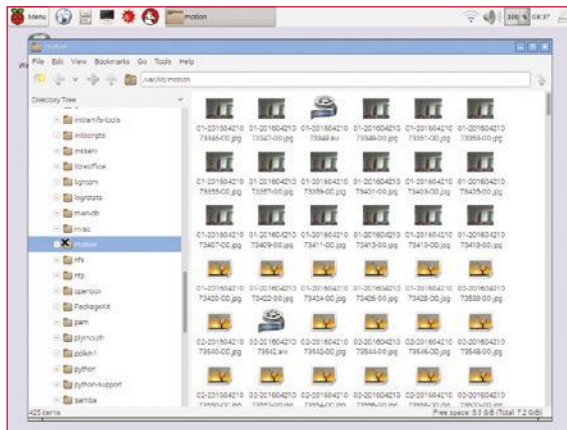
quiet off

Vergessen Sie nicht, den Alarmton nach erfolgter Kalibrierung wieder abzuschalten, um nicht versehentlich damit das Baby aufzuwecken.

[illegible]

12 Bildqualität anpassen

13 Sie können die Auflösung der aufgenommenen Bilder in der Datei `motion.conf` anpassen, aber nicht beliebig. Eine Bildgröße von 133 x 255 Pixeln funktioniert wahrscheinlich nicht, die Zahlen müssen jeweils Vielfache von vier sein. Hohe Auflösungen wären beispielsweise 1280 x 800 oder 1920 x 1080. Sie verlangsamen den Videostream und erhöhen die Größe der aufgenommenen AVI-Datei signifikant.



14 Bildqualität prüfen

Um die Qualität der aufgenommenen Bilder zu prüfen, öffnen Sie die Nutzeroberfläche (oder installieren Sie tightvncserver und öffnen Sie eine Remote-Verbindung), gehen Sie zum Verzeichnis `/var/lib/motion`, und sehen Sie sich die Bilder an. Passen Sie, falls nötig, die Auflösung an.

```
pi@raspberrypi:~$ dmesg | tail
[ 18.815042] cfg80211: (570000000 KHz - 660000000 KHz @ 2160000 KHz), (N/A, 40
[ 22.113760] smmc95xx 1-1.1:1.0 eth0: hardware isn't capable of remote wakeup
[ 22.114488] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 207.965707] media: Linux media interface: v0.10
[ 207.992283] Linux video capture interface: v2.00
[ 208.062384] bcm2835-v4l2: scene mode selected 0, was 0
[ 208.068577] bcm2835-v4l2: V4L2 device registered as video0 - stills mode > 12
[ 208.076533] bcm2835-v4l2: Broadcom 2835 MMAL video capture ver 0.0.2 loaded.
[ 322.391108] bcm2835-v4l2: error 0 waiting for frame completion
[ 2682.082108] bcm2835-v4l2: error 0 waiting for frame completion
pi@raspberrypi:~$
```

15 Kameraverbindung prüfen

Wenn motion nicht korrekt funktioniert, keine Bilder erzeugt werden oder keine Verbindung zum Videostream hergestellt werden kann, helfen diese beiden Befehle möglicherweise:

```
tail -f /var/log/syslog
```

und

```
GNU nano 2.2.6 File: /etc/motion/motion.conf
# Recommended to use absolute path. (Default: current working directory)
target_dir /var/lib/motion

# File path for snapshots (jpeg or ppm) relative to target_dir
# Default: %Y-%m-%d-%H-%M-%S-snapshot
# Default value is equivalent to legacy oldlayout option
# For Motion 3.0 compatible mode choose: %Y/%m/%d/%H/%M/%S-snapshot
# File extension .jpg or .ppm is automatically added so do not include this.
# Note: A symbolic link called lastsnap.jpg created in the target_dir will always
# point to the latest snapshot, unless snapshot_filename is exactly 'lastsnap'
snapshot_filename %Y-%m-%d-%H-%M-%S-snapshot

# File path for motion triggered images (jpeg or ppm) relative to target_dir
# Default: %Y-%m-%d-%H-%M-%S-lq
# Default value is equivalent to legacy oldlayout option
# For Motion 3.0 compatible mode choose: %Y/%m/%d/%H/%M/%S-lq
# File extension .jpg or .ppm is automatically added so do not include this
# Set to 'preview' together with best-preview feature enables special naming
# convention for preview shots. See motion guide for details

^G Get Help ^W WriteOut ^R Read File ^V Prev Page ^C Cut Text ^X Cut Pos
^X Exit ^J Justify ^M Where Is ^N Next Page ^U UnCut Text ^S To Spell
```

`dmesg | tail`

Sie zeigen mögliche Probleme an, was Ihnen dabei hilft, deren Ursachen zu diagnostizieren und sie zu lösen. Meist liegt es an fehlenden oder nicht funktionierenden Treibern, vor allem bei externen USB-Webcams. Beenden Sie mit `Strg+Z`.

16 Bilder benennen

Die aufgenommenen Videobilder können sehr spezifisch benannt werden, basierend auf Zeit- und Datumsangaben. Sie finden die Werte unter „target base directory“ in der Datei `motion.conf`. Sie können Ihre Bilder beispielsweise in nach Aufnahmedatum sortierten Ordnern speichern:

```
%Y_%m_%d/%v-%Y%m%d%H%M%S-%q1
```

Am „/“ erkennen Sie, dass sowohl Bilder als auch Ordner mit dem jeweiligen Datum benannt und Bilder zusätzlich mit der Aufnahmezeit versehen werden.

17 Außerhalb des Heimnetzwerks

Wie wäre es, auch beim Essen in Ihrem Lieblingsrestaurant noch hin und wieder einen Blick auf das schlafende Baby werfen zu können? Mit der Software No-IP ist das möglich. Sie erlaubt es, mithilfe einer Client-App auch dann von außerhalb des Heimnetzwerkes auf den Baby-Monitor zuzugreifen, wenn Sie über keine (teure) statische IP-Adresse verfügen. Auf noip.com finden Sie weitere Informationen.

Stromversorgung der Webcam

Verschiedene USB-Webcams benötigen auch verschiedene Arten der Stromversorgung. Manche sind mit der Versorgung des Raspberry Pi selbst zufrieden, andere brauchen zusätzlich Strom, etwa von einem USB-Hub mit Netzteil. Ist keine Steckdose in der Nähe Ihrer Wunschposition, sollte ein wiederaufladbarer Akku genügend Strom für 12 Stunden Betrieb beider Geräte bereitstellen.



RISC OS als Betriebssystem für den Raspberry Pi

Ein wenig Retro-Feel

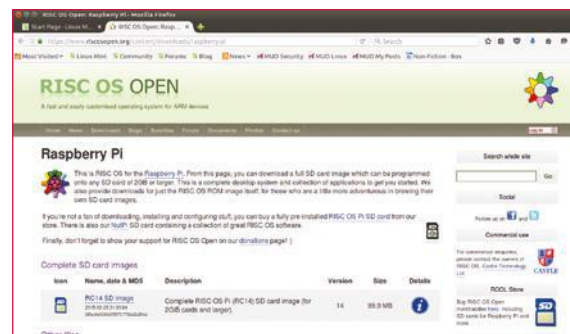
Die Version des Betriebssystems für den Raspberry Pi – RISC OS Open – ist sehr charmant gestaltet: Wenn Sie an die Computer-Benutzeroberflächen Anfang der 1990er-Jahre denken, sind Sie nicht weit vom aktuellen Look von RISC OS entfernt. Das Aussehen und der Aufbau der grafischen Benutzeroberfläche sind sehr einfach gehalten; sie versprühen Retro-Atmosphäre, kombiniert mit ein bisschen Verrücktheit.

Es muss nicht unbedingt Raspbian oder ein anderes Linux-Betriebssystem sein. Es gibt Alternativen.

Das offiziell empfohlene Betriebssystem für den Raspberry Pi ist Raspbian. Es gibt noch einige andere Optionen, aber die meisten davon basieren letztlich auf dem Linux-Kernel. Bei RISC OS handelt es sich jedoch um ein Betriebssystem ganz anderen Ursprungs. Das macht es noch interessanter, es einmal auszuprobieren.

RISC OS wurde 1987 in Cambridge entwickelt (wobei RISC für „Reduced instruction set computing“, also etwa „Datenverarbeitung mit reduziertem Befehlssatz“ steht), und seine Wurzeln reichen zurück bis in die Zeit des ersten ARM-Prozessors. Tatsächlich ist RISC OS ein Nachfahre des Betriebssystems, das auf dem Heimcomputer BBC Micro lief.

Das Betriebssystem RISC OS ist aber nach wie vor relevant, jedenfalls mit Blick auf den Raspberry Pi.



Oben Wählen Sie das SD-Karten-Image ganz oben.

01 RISC OS herunterladen

Holen Sie sich zunächst von riscosopen.org/content/downloads/raspberry-pi die aktuelle Version des Betriebssystems. Das Zip-Archiv mit dem SD-Karten-Image ist knapp 120 MB groß. Laden Sie dieses auf Ihren Computer herunter.

02 Archivdatei entpacken

Entpacken Sie das heruntergeladene Zip-Archiv. Im Terminal können Sie dies unter Angabe von Verzeichnispfad und Dateiname mit dem unzip-Befehl tun:

unzip Downloads/riscos-2015-02-17.14.zip

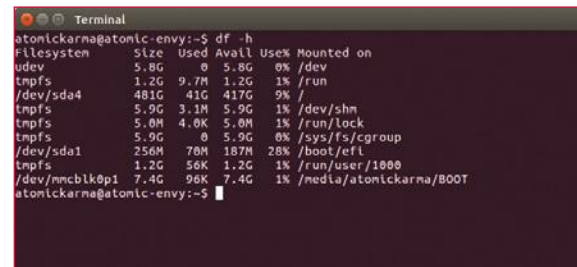
Ersetzen Sie dabei den Pfad durch Ihren. Das SD-Karten-Image (mit der Dateiendung .IMG) befindet sich im Home-Verzeichnis.



Oben Hier wird die Zip-Datei entpackt.

Was Sie brauchen

- RISC OS
bit.ly/2auOpYL
- Monitor
- Tastatur
- Drei-Tasten-Maus (ein klickbares Scrollrad kann ebenfalls als mittlere Taste dienen)



Oben SD-Karten sind oft am „mmc“ oder „sdd“ im Namen zu erkennen.

03 SD-Karte bespielen

Dieses Image von RISC OS sollten Sie nun auf eine formatierte SD-Karte kopieren. Stecken Sie die Karte in den Kartenleser, öffnen Sie das Terminal und rufen Sie dort die Liste der eingehängten Geräte ab:

df -h

Die SD-Karte müsste in dieser Liste erscheinen. Sie erkennen sie am Verzeichnispfad, dem Dateinamen und der Größe.

04 RISC OS auf dem Raspberry Pi installieren

Als Nächstes hängen Sie die SD-Karte wieder ab:

umount /dev/mmcblk0p1

Verwenden Sie anschließend den Befehl „dd“, um die Image-Datei zu überspielen. Seien Sie bei der Eingabe des Schreibziels besonders achtsam, denn eine falsche Angabe könnte hier Ihre Festplatte überschreiben!



```

etonic@atomic-envy:~$ sudo dd bs=4M if=riscos-2015-02-17-RC14.img of=/dev/mmcblk0
468+1 records in
468+1 records out
194688000 bytes (2.0 GB, 1.8 GiB) copied, 231.566 s, 8.5 MB/s
etonic@atomic-envy:~$

```

Oben Geben Sie den Dateipfad und das Zielgerät sorgfältig ein.

```

sudo dd bs=4M if=riscos-2015-02-17-RC14.img of /dev/mmcblk0

```

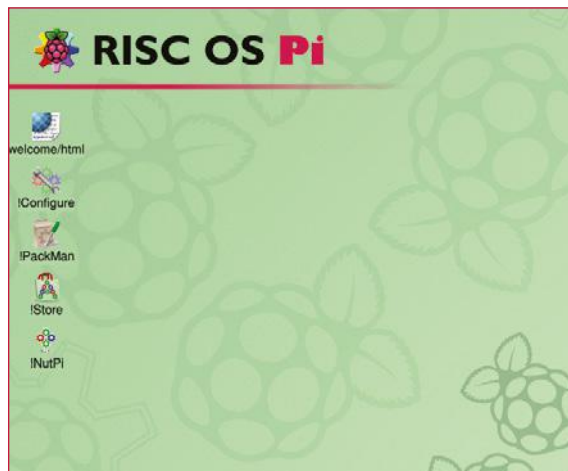
Wie Sie sehen, ist das „p1“ am Ende verschwunden. Es bezog sich auf eine Partition. Stecken Sie die fertige SD-Karte in den Raspberry Pi und booten Sie diesen.



Oben Das NOOBS-Paket dient als alternativer Installationsweg.

05 NOOBS als alternativer Installationsweg

Falls Sie das Ganze nicht in der Kommandozeile erledigen möchten, können Sie als Alternative die offizielle NOOBS-Software verwenden. Holen Sie sich diese von raspberrypi.org/downloads und überspielen Sie sie auf Ihre SD-Karte. Beim Hochfahren des Raspberry Pi können Sie dann RISC OS aus den enthaltenen Betriebssystemen auswählen.



Oben So sieht der Desktop von RISC OS aus.

06 RISC OS geladen

In beiden Fällen – mit dem Original-Image von RISC OS oder der NOOBS-Variante – ist RISC OS nun als Betriebssystem auf dem Pi geladen.



Oben Das Scrollrad dient als mittlere Maustaste. Diese benötigen Sie bei RISC OS zum Aufruf von Kontextmenüs.

07 Die richtige Maus

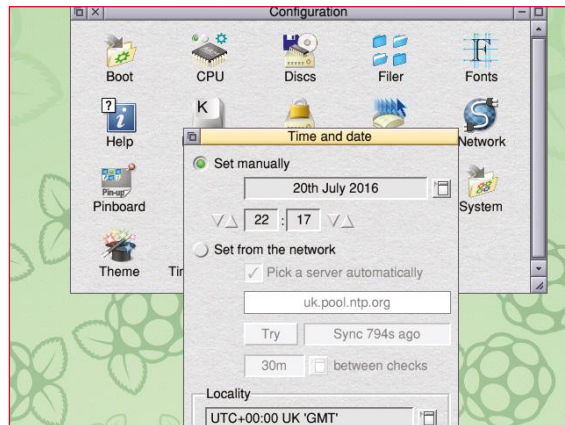
Um mit RISC OS arbeiten zu können, brauchen Sie Tastatur und Maus direkt am Raspberry Pi. Die Maus sollte eine Drei-Tasten-Maus sein, denn bei RISC OS ist traditionell die mittlere Taste dazu da, Kontextmenüs aufzurufen (wozu bei Linux, Windows und macOS die rechte Maustaste dient). Ein klickbares Scrollrad in der Mitte funktioniert auch. Des Weiteren benötigen Sie ein Ethernet-Kabel, um den Pi ans Netzwerk anzuschließen, denn RISC OS besitzt leider keine WLAN-Funktionalität.



Oben Die Ethernet-Funktionalität muss erst eingeschaltet werden.

08 Ethernetfunktion von RISC OS aktivieren

Standardmäßig ist die Ethernet-Funktion deaktiviert. Wie Sie sie einschalten, erfahren Sie nach einem Klick auf „welcome/html“ auf dem RISC-OS-Desktop. Falls dieser Bereich fehlt, doppelklicken Sie auf „iConfigure“, gehen Sie dann im Menü zu Network > Internet, setzen Sie ein Häkchen bei „Enable TCP/IP Protocol Suite“ und speichern Sie die Einstellungen. Führen Sie anschließend einen Neustart des Pi durch.



Oben Das Design ist ziemlich schlicht und ein bisschen retro.

09 RISC OS ausprobieren

Die Funktionsweise von RISC OS ist ziemlich selbst-erklärend. Spielen Sie einfach ein bisschen damit herum. Anwendungen sind meist durch ein „!“ (bei RISC OS mit „P!ng“ bezeichnet) vor dem Namen gekennzeichnet und werden durch Doppelklick gestartet.

Software installieren

Bei RISC OS werden zwei Paketmanager mitgeliefert. Der eine, Packman, installiert und aktualisiert Software. Der andere, iStore, bietet kommerzielle Lösungen an. Sie können Software auch manuell installieren, indem Sie Zip-Archive von Ihrem Desktop-Computer auf die SD-Karte kopieren und von dort wiederum den Inhalt der Archivdatei dem RISC-OS-Dateisystem hinzufügen. Sehr alte RISC-OS-Programme laufen jedoch möglicherweise nicht, denn die Architektur aktueller ARM-CPUs unterscheidet sich von der früherer Prozessoren.

Die Grundlagen des Sense HAT (Teil 1)

Mit Stein, Papier, Schere, Echse, Spock verbessern Sie Ihre Programmierkünste mit dem Sense HAT.

Ende 2015 wurden zwei Raspberry Pis mit je einem Astro Pi ausgestattet, einer Erweiterungsplatine voller Sensoren mit einer 8x8-LED-Anzeige. So ausgerüstet und mit Experimenten von Schulkindern ausgestattet, wurden sie zu Major Tim Peake auf die Raumstation ISS geschickt. Kurz darauf war der Astro Pi dann als Sense HAT für die Allgemeinheit käuflich zu erwerben. Diese Version gleicht dem Astro Pi in puncto Hardware und Sensoren exakt, verfügt jedoch über eine neue Sense-HAT-API.

Diese Anleitung stellt die Sense-HAT-Hardware vor und zeigt, wie man eine Sense-HAT-Version des modifizierten Klassikers Stein, Papier, Schere, Echse, Spock erstellt. Das dabei erworbene Wissen können Sie später auch auf eigene Projekte anpassen und übertragen.

Was Sie brauchen

- Raspberry Pi
- Sense HAT

Unten Der Sense HAT hat nicht nur eine LED-Anzeige, sondern auch viele nützliche Sensoren.

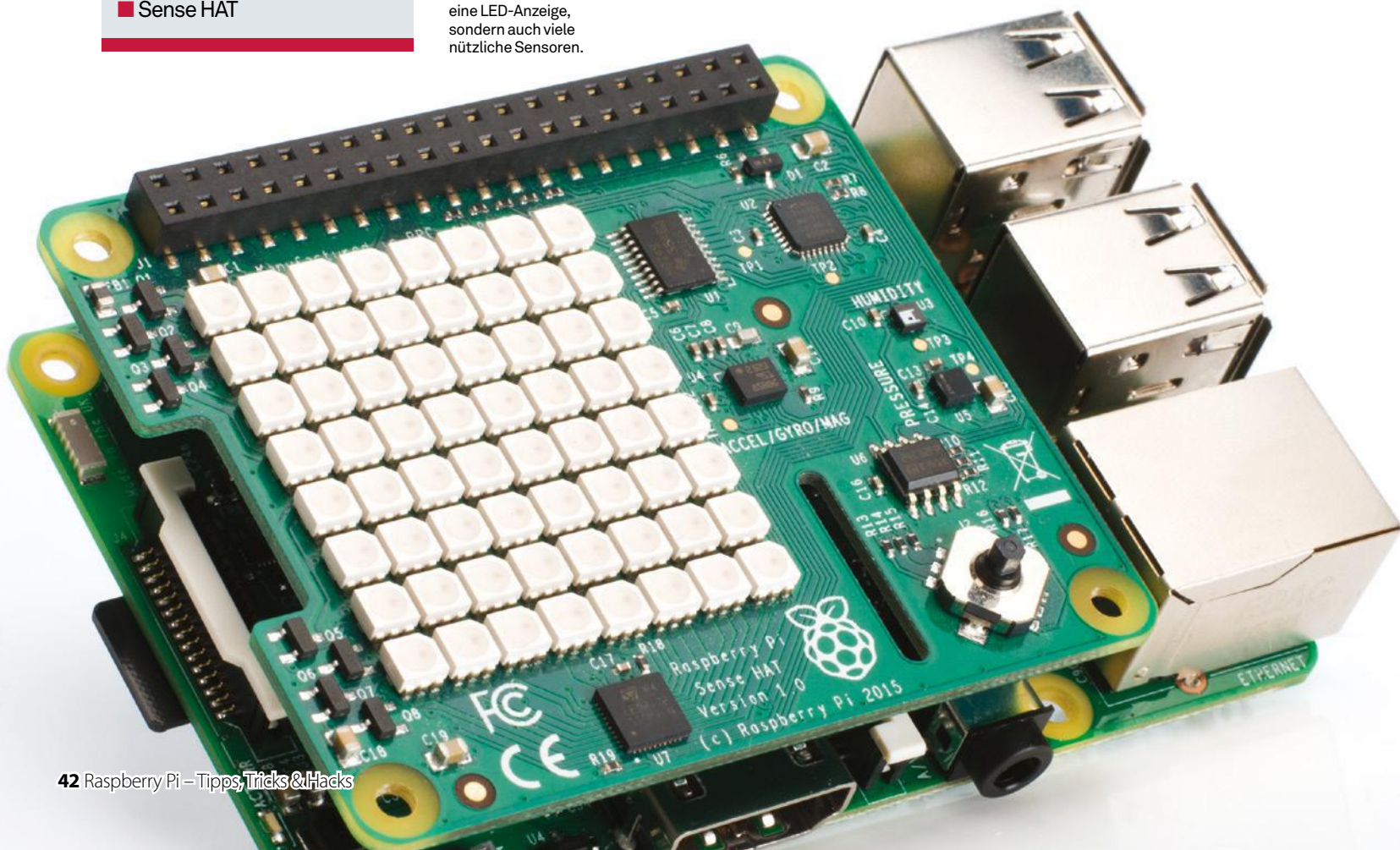
01 Sense-HAT-Software installieren

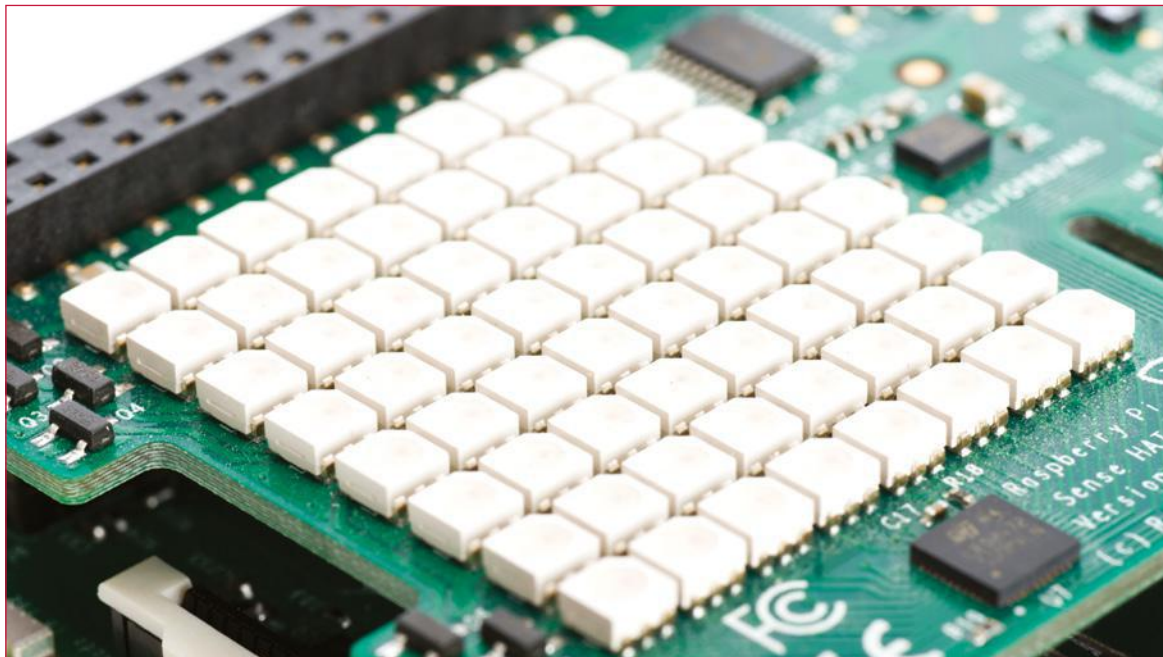
Verbinden Sie die Platine mit der GPIO-Leiste und installieren Sie die Sense-HAT-Software. Auf den neueren Raspbian-Distributionen ist sie vorinstalliert, für ältere müssen Sie sie herunterladen. Starten Sie den Pi und geben Sie zur Installation „sudo apt-get install sense-hat“ im Terminal ein. Starten Sie danach neu.

```
sudo apt-get update
sudo apt-get install sense-hat
sudo reboot
```

02 Lauftext für LEDs

Lauftext für LCD- oder LED-Displays zu programmieren, kann kompliziert und frustrierend sein. Die Sense-HAT-API reduziert die Komplexität auf eine einzige Zeile Code: „sense.show_message(„Das ist ein Lauftext“)“. Öffnen Sie Ihren Python-Editor und geben Sie den unten abgedruckten Code ein, speichern Sie und führen Sie das Skript aus. Ihr Text läuft sogleich über





Links Jede LED des Rasters kann einzeln angesteuert werden – so können Sie Bilder anzeigen.

die LED-Anzeige des Sense HAT. Passen Sie den Text im Code beliebig an. Auch die Anzeigefarbe und die Laufgeschwindigkeit lassen sich verändern, indem Sie „text_ colour=[255, 0, 0]“ (der RGB-Wert der Farbe) und „scroll_speed=(0.05)“ verwenden. Experimentieren Sie mit dem Codebeispiel:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.show_message("Raspberry Pi", text_colour=[255, 0, 0])
```

03 Temperaturmessung

Der Sense HAT hat einen Temperatursensor eingebaut, der messen und ausgeben kann (Zeile 3). Er ist recht nah an der CPU verbaut, was bei hoher Last durch Abwärme die Messung beeinträchtigen kann. Insgesamt ist er aber recht genau. Um die Temperatur zu messen, öffnen Sie den Python-Editor, tippen Sie den unten stehenden Code ein, speichern und schließen Sie. Das Ausführen des Skripts liest den aktuellen Messwert und gibt ihn aus:

```
from sense_hat import SenseHat
sense = SenseHat()
temp = sense.get_temperature()
print("Temperatur: %s C" % temp)
```

04 Kompassfunktion

Einer der raffinierten Sensoren ist das Magnetometer, das als Kompass verwendet werden kann. Es misst die Position des Sense HAT relativ zum magnetischen Nordpol. Auch dieser Code ist einfach anzuwenden: „sense.get_compass()“ in Zeile 3 gibt die Position aus, die in der Variablen north gespeichert ist. Der Messwert wird dann in Zeile 4 angezeigt. Mit folgendem Code können Sie den Kompass und das Messverfahren testen:

```
from sense_hat import SenseHat
sense = SenseHat()
north = sense.get_compass()
print("Norden: %s" % north)
```

05 Vom Bild zur LED-Grafik

Digitale Bilder sind aus Pixeln zusammengesetzt. Jede LED auf der Matrix-Anzeige kann anhand einer Bilddatei angesteuert werden. Sie können beispielsweise das Bild einer Echse laden, es werden Farben und Positionen berechnet und dann die entsprechenden LEDs aktiviert. Die Bilddatei muss 8x8 Pixel groß sein, damit die LED-Matrix sie auslesen kann. Laden Sie das Testbild – lizard.png – herunter und speichern Sie es in Ihrem Programmordner. Mit folgendem Code öffnen und laden Sie die Bilddatei einer Echse (Zeile 3) – der Sense HAT übernimmt die restliche Arbeit:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.load_image("lizard.png")
```

06 Ihr eigenes 8x8-Bild

Es gibt zwei weitere Methoden, ein Bild auf die LED-Anzeige zu bekommen. Die erste ist eine exzellente Software, mit der Sie die LEDs in Echtzeit schalten können. Sie können Farben wechseln, rotieren und das Bild als Code oder als 8x8 Pixel große PNG-Datei exportieren. Installieren Sie die Python PNG Library, indem Sie im Terminal „sudo pip3 install pypng“ eingeben. Führen Sie danach den Befehl „git clone github.com/jrobinson-uk/Rpi_8x8GridDraw“ aus. Ist die Installation beendet, öffnen Sie den zugehörigen Ordner per „cd Rpi_8x8GridDraw“ und geben Sie „python3 sense_grid.py“ ein, um das Programm zu starten.

07 Bild exportieren

Der Grid Editor bietet auf der rechten Seite des Fensters eine Auswahl verschiedener Farben. Wählen Sie die Farbe und anschließend die entsprechende LED auf dem Raster und klicken Sie „Play on LEDs“, um Ihre Auswahl an die LEDs des Sense HAT zu senden. Mit dem Befehl „Clear Grid“ können Sie alles löschen und von vorn beginnen. Ihren Entwurf können Sie entweder als PNG-Datei exportieren und dabei den im vorherigen Schritt gezeigten Code verwenden, um das Bild anzuzeigen, oder Sie lassen es direkt als Code ausgeben, den Sie in Ihrem Programm einfügen können.

SPSES: Die Ursprünge

Sheldon Cooper aus The Big Bang Theory brachte das Spiel in den Fokus der Öffentlichkeit, als er in einer Folge der TV-Serie dessen Regeln erklärte. Erfunden wurde es aber nicht in der Serie. Die Erweiterung des Klassikers Stein, Papier, Schere wird vielmehr Sam Kass und Karen Bryla zugeschrieben.

SPSES in Python

Wenn Sie eine Python-Version von SPSSES ausprobieren möchten, statten Sie der Seite trinket.io/python/46302ff1af einen Besuch ab. Die Version für Sense HAT verwendet ein ähnliches Verfahren zur Berechnung des Spielstands.

08 LED-Programmierung 1

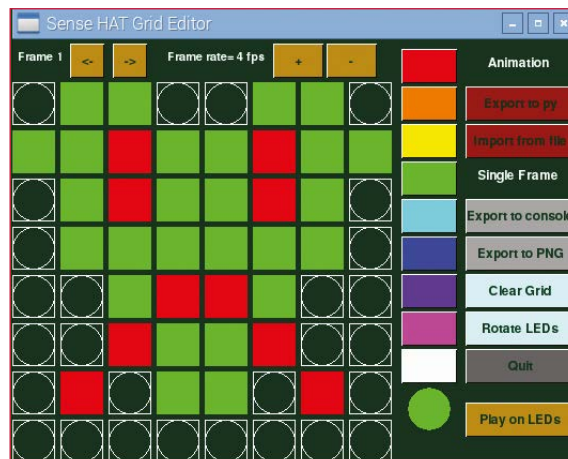
Die zweite Variante, ein Bild zu erzeugen, ist es, jede LED einzeln zu programmieren. Das funktioniert, indem Sie eine Variable für die Farbe der LED erstellen und ihr einen RGB-Wert zuweisen. Verwenden Sie so viele Farben, wie Sie brauchen. Jetzt können Sie das Bild in Code erzeugen, indem Sie die Namen der Variablen in ein Raster eintragen: In diesem Beispiel bilden die Variablen X und O ein Fragezeichen ab. Die LEDs werden mit dem Befehl „sense.set_pixels(question_mark)“ geschaltet:

```
from sense_hat import SenseHat
sense = SenseHat()
X = [255, 0, 0] # Rot
O = [255, 255, 255] # Weiß
question_mark = [
    0, 0, 0, X, X, 0, 0, 0,
    0, 0, X, 0, 0, X, 0, 0,
    0, 0, 0, 0, 0, X, 0, 0,
    0, 0, 0, 0, X, 0, 0, 0,
    0, 0, 0, X, 0, 0, 0, 0,
    0, 0, 0, X, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, X, 0, 0, 0, 0
]
sense.set_pixels(question_mark)
```

09 LED-Programmierung 2

Möchten Sie mehr Farbe in Ihr Bild bringen, erstellen Sie einfach weitere Variablen mit RGB-Werten analog zum obigen Beispiel. Ersetzen Sie den Code durch die untenstehende Sequenz, um ein neues Bild generieren zu lassen. RGB-Werte aller 16.581.375 verfügbaren Farben bekommen Sie beispielsweise auf colorpicker.com.

```
X = [255, 0, 0] # Rot
O = [255, 255, 255] # Weiß
B = [0, 255, 0] # Grün
new_image = [
    B, B, 0, 0, 0, 0, X, X,
    B, B, 0, 0, 0, 0, X, X,
    B, B, 0, 0, 0, 0, X, X,
    B, B, 0, 0, 0, 0, X, X,
    B, B, 0, 0, 0, 0, X, X,
    B, B, 0, 0, 0, 0, X, X,
    B, B, 0, 0, 0, 0, X, X,
    B, B, 0, 0, 0, 0, X, X,
    B, B, 0, 0, 0, 0, X, X
]
sense.set_pixels(new_image)
```



Rechts Mit dem Grid Editor ist es leicht, den Code für die Bilder zu erzeugen.

10 Das Bild einer Echse

Jetzt können Sie Ihre eigenen Bilder erstellen. Entwerfen Sie eine einfache Echse mit dem LED-Raster, dann verfeinern Sie Ihre Ideen und Konzepte – Sie brauchen schließlich noch Stein, Papier, Schere und Spock. Bei Johan Vinet ([johanvinet.tumblr.com/image/127476776680](https://www.tumblr.com/johanvinet)) finden Sie exzellente und inspirierende 8x8-Pixelgrafiken.

11 Zufällige Bildauswahl

Im nächsten Schritt geht es um die Auswahl der Bilder. Benennen Sie zunächst in aufsteigender Folge, beginnend bei 1 Ihre Dateien um – 1.png, 2.png zum Beispiel. In einer neuen Variablen namens playersChoice speichern Sie den Dateinamen des aktuellen Bildes. Laden Sie das Bild mit „sense.load_image(str(playersChoice) + “.png”)“ – playersChoice muss nun noch in einen String konvertiert werden. Wenn das Bild dargestellt wird, erhöht sich die Variable, und das nächste Bild wird geladen. Bauen Sie eine Bedingung ein, die prüft, ob das letzte Bild erreicht wurde. Wenn Sie fünf Bilder haben, geben Sie ein:

```
if playersChoice == 5:
    playersChoice = 0
```

Damit prüfen Sie, ob die Variable am letzten Bild angekommen ist, und setzen sie auf null zurück. Es wird wieder das erste Bild geladen und die Schleife beginnt von vorn:

```
import time
while True:
    sense.load_image(str(playersChoice) + “.png”)
    playersChoice = playersChoice + 1
    time.sleep(1)
    if playersChoice == 5:
        playersChoice = 0
```

12 Bildschirm löschen

Beim Bildwechsel werden die LEDs von Ihrem Code an- und abgeschaltet. Manchmal sollen jedoch auch alle LEDs gleichzeitig deaktiviert werden, was als Löschen des Bildschirms bezeichnet wird. Um alle LEDs zu „löschen“, schalten Sie sie mit dem Befehl „sense.clear()“ aus.

13 Einrichtung des Joysticks in PyGame

Der Sense HAT ist mit einem kleinen multidirektionalen Joystick ausgestattet, dessen Richtungseingaben programmiert werden können. Das Spiel SPSSES nutzt den Joystick für die Auswahl der Bilder. Richten Sie in einer neuen Python-Datei PyGame ein, indem Sie die PyGame-Module am Dateianfang importieren (Zeilen 1 und 2). Initialisiert wird mit „pygame.init()“. Nun öffnet PyGame ein neues Fenster, doch weil das Spiel auf dem Sense HAT läuft, ist dieses überflüssig und kann mit „pygame.display.set_mode((140, 180))“ minimiert werden.

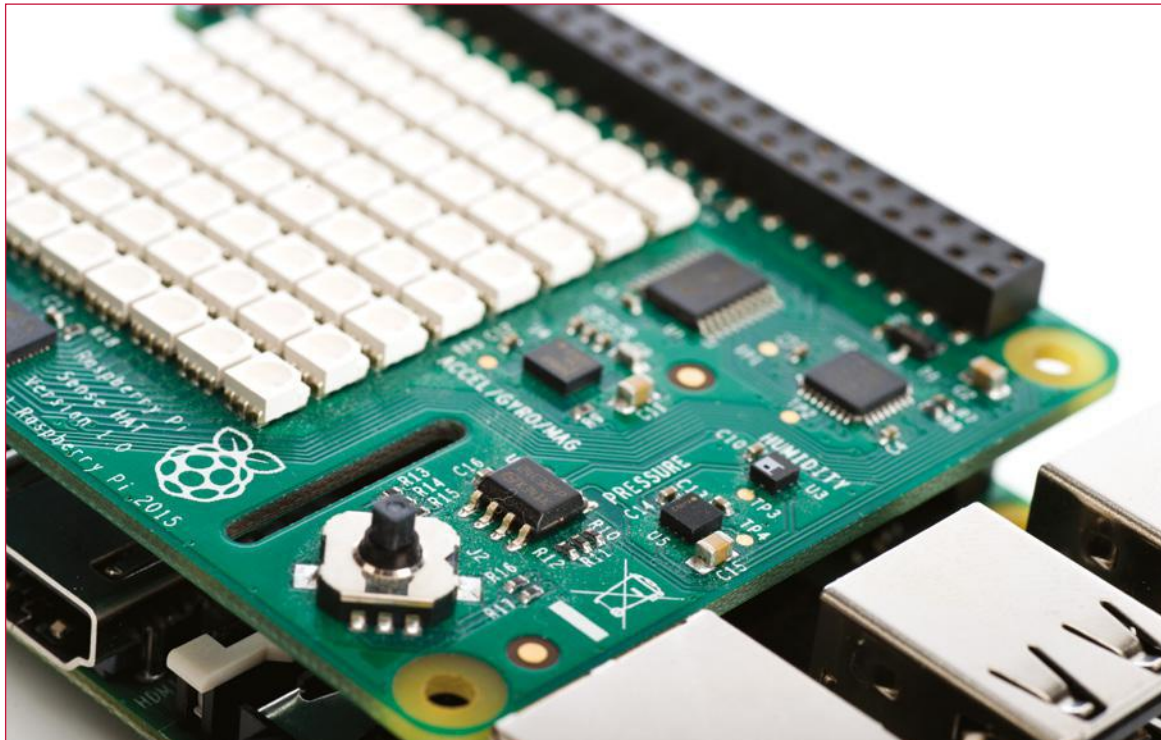
```
import pygame
from pygame.locals import *

###PyGame-Screen erstellen###

pygame.init()
pygame.display.set_mode((140, 180))
```

14 Konfiguration der Joystickeingabe

Um die Joystickeingabe einzurichten, erstellen Sie die Variable running und setzen sie auf True (Zeile 1). Fügen Sie eine



Links Der kleine Joystick im Vordergrund des Fotos dient als Eingabemethode für die Bildauswahl.

while-Sequenz (Zeile 2) hinzu, um durchgehend zu prüfen, ob der Joystick bewegt wurde. Das Programm fragt in Zeile 3 ein Event ab, die Bilderauswahl startet nach dem fünften Bild neu. Zeile 4–5 prüft zwei Bedingungen: ob eine Taste gedrückt wurde und ob der Wert der Bildauswahl unter fünf liegt. Ist die Bedingung erfüllt, wird die Joystickeingabe abgefragt. Pfeil nach oben auf der Tastatur entspricht K_UP und damit prinzipiell einer Joystickbewegung nach oben (Zeile 6). Fügen Sie schließlich einen print-Befehl hinzu, um zu prüfen, ob Ihr Programm auf die Joystickeingabe korrekt reagiert (Zeile 7).

```
running = True
while running == True:
    for event in pygame.event.get():
```

```
        if event.type == KEYDOWN and playersChoice < 5:
            if event.key == K_UP:
print("UP")
```

15 Manuelle Bildauswahl

PyGame wartet jetzt auf Events und fragt ab, ob der Joystick nach oben bewegt wird. Im SPSES-Spiel erlaubt dies dem Spieler, ein Bild auszuwählen, das seinen Zug darstellt, Papier beispielsweise. Ersetzen Sie den Befehl „print(„UP“)“ in Zeile 7 durch den Bildauswahl-Code aus Schritt 11. Laden Sie wie zuvor die Bilder mit „sense.load_image(str(playersChoice) + „.png“)“ auf die LED-Anzeige und erhöhen Sie die Variable playersChoice mit „playersChoice = playersChoice + 1“, um das nächste Bild anzuzeigen. Zeile 6 grenzt die Bildauswahl ein und setzt sie anschließend wieder auf null zurück (Zeile 8), damit die Auswahl von vorn beginnt.

```
if event.type == KEYDOWN and playersChoice < 5:
    if event.key == K_UP:
        print(playersChoice)
        sense.load_image(str(playersChoice) + “.png”)
```

Der Sense HAT ist mit einem multidirektionalen Joystick ausgestattet, dessen Eingaben programmiert werden.

```
        playersChoice = playersChoice + 1
    if playersChoice == 5:
        playersChoice = 0
```

16 Auswahl bestätigen

Wenn Sie das Bild Ihrer Wahl gefunden haben, müssen Sie es natürlich auch auswählen können. Das funktioniert, indem Sie den gesamten Joystick wie einen Knopf drücken, was wie ein Druck der Eingabetaste ausgelesen wird. In Zeile 2 fragt der Code „if event.key == K_RETURN:“ die Eingabe ab und unterbricht die Schleife. Das aktuell angezeigte Bild wird dann in der Variablen playersChoice gespeichert und später mit der Auswahl des Computergegners verglichen, um den Sieger zu ermitteln.

```
"""Checks for a 'select / Enter' Choice"""
if event.type == KEYDOWN:
    if event.key == K_RETURN:
        running = False
        break
"""Ends loop and moves onto main game"""
```

17 SPSES

Jetzt haben Sie die Grundstruktur des SPSES-Spiels. Damit können Sie aus fünf Bildern auswählen, die jeweils eine Handgeste darstellen: Stein, Papier, Schere, Echse, Spock. Einen Vorgeschmack auf das Spiel finden Sie auf YouTube ([youtube.com/watch?v=T_ZvWkMgVFM](https://www.youtube.com/watch?v=T_ZvWkMgVFM)). Vielleicht können Sie in der Zwischenzeit noch ein paar 8x8-Pixelgrafiken entwerfen?

Die Grundlagen des Sense HAT (Teil 2)

Mit dem Vorwissen aus dem letzten Teil erstellen Sie Ihre Sense-HAT-Version von SPSES.

Was Sie brauchen

- Raspberry Pi 2
- SenseHAT

Sicher haben Sie bereits einmal Stein, Papier, Schere gespielt.

Es gibt nur drei mögliche Ergebnisse, abgesehen von einem Unentschieden. Sam Kass und Karen Bryla haben dem Spiel in ihrer Alternativversion „Spock“ und „Echse“ hinzugefügt – „Spock“ wird vom vulkanischen Gruß symbolisiert, „Echse“ durch die „Krokodilgeste“, eine zum Mund geformte Hand. Spock zerstört die Schere und vaporisiert den Stein, wird allerdings von der Echse vergiftet und von Papier widerlegt. Die Echse vergiftet Spock und frisst Papier, wird vom Stein zermalmt und von der Schere enthauptet. Erstellen Sie also mit uns Ihre eigene Sense-HAT-Version des Spiels!

01 SPSES mit dem Sense HAT

Am Ende des ersten Teils hatten Sie einen Ordner mit fünf 8x8 Pixel großen Bildern, die jeweils eines der Spielobjekte – Stein, Papier, Schere, Echse, Spock – abbilden: 0.png, 1.png, 2.png, 3.png und 4.png. Bei Programmstart werden Sie aufgefordert, per Joystickbewegung nach oben durch die Objektliste zu navigieren und eines auszuwählen. Mit der Eingabetaste wählen Sie es aus, woraufhin der Computer seine eigene (zufällige) Wahl trifft. Jedes Bild hat einen Wert und eine Formel, um den Modulus (den Rest) aus der Division der beiden Werte zu ermitteln. Dieser Wert entscheidet über Sieg, Niederlage oder Unentschieden.

02 Module importieren

Starten Sie den Raspberry Pi und öffnen Sie das Terminal. Mit **sudo idle3** laden Sie den Python-3-Editor. Importieren Sie das PyGame- (Zeile 1) und das Sense-HAT-Modul (Zeile 3). Für die Spielzüge des Computers brauchen Sie das Random-Modul, importiert in Zeile 4.

```
import pygame
import pygame.locals import *
from sense_hat import SenseHat
import random
import time
```

03 Lauftext

Benachrichtigungen des Spiels werden als Lauftext auf der LED-Matrix ausgegeben. Die Sense-HAT-API braucht dazu nur eine einzige Zeile Code: **sense.show_message("Das ist ein Test.")**. Die Nachricht läuft dann über die LEDs des Sense HAT. Ersetzen Sie den Text zwischen den Anführungszeichen durch eigenen Text. Auch Farbe und Geschwindigkeit des Lauftextes können angepasst werden, und zwar mit **text_colour=[255, 0, 0]** (geben Sie den RGB-Wert an) und **scroll_speed=(0.05)**. Experimentieren Sie einfach mit dem Beispieldcode:



```
sense = SenseHat()
sense.show_message("Raspberry Pi",
text_colour=[255, 0, 0])
```

04 LED-Bildanzeige

Bilder bestehen aus Pixeln, die zu einem Gesamtbild kombiniert werden. Jede LED der Matrix kann automatisch aus einer Bilddatei geschaltet werden. Dazu muss das Bild lediglich der Größe der LED-Matrix von 8x8 Pixeln entsprechen. Laden Sie das Testbild einer Echse – lizard.png – herunter und speichern Sie es im Programmordner. Nach dem Laden werden Farben und Positionen berechnet und die entsprechenden LEDs aktiviert. Mit folgendem Code öffnen und laden Sie das Bild:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.load_image("lizard.png")
```

05 Variablen und Initialisierung von PyGame

Erstellen Sie nun die Variablen für die Wahl des Spielers (Zeile 1) und des Computers sowie für die Verwaltung des aktuell auf der LED-Matrix angezeigten Bildes (global count). Dies sind globale Variablen, die in allen Bereichen des Programms verfügbar sind und die gespeicherten Werte ausgeben. Initialisieren Sie mit **pygame.init()** und **pygame.display.set_mode((140, 180))** das PyGame-Fenster. Es wird fürs Spiel nicht benötigt, wählen Sie also eine kleine Fenstergröße. Laden Sie das erste Bild mit **sense.load_image("0.png")**, der Befehl öffnet Bild Nummer null in Ihrem Ordner. Falls Ihnen die LEDs zu hell sind, verringern Sie mittels **sense.low_light = True** die Helligkeit. Setzen Sie **playersChoice** auf 0 – das erste Bild – und starten Sie das Spiel mit **gameRunning = True**.

```
global playersChoice
global computer_choice
global count
###PyGame-Fenster einrichten###
pygame.init()
pygame.display.set_mode((140, 180))

###Prepare Sense Hat###
sense = SenseHat()
sense.load_image("0.png")
sense.low_light = True #Augen schonen!
playersChoice = 0
gameRunning = True
```




06 Werte/Zahlen in Objekte konvertieren

Im Spiel werden die einzelnen Objekte mit Zahlen identifiziert, nicht mit Namen. Sie können also ein Bild wählen und eine Modulus-Funktion anwenden, um das Resultat zu berechnen. Erstellen Sie eine Funktion, welche die jeweiligen Werte konvertiert und die passenden Namen zuordnet. Eine simple Bedingung prüft die Nummer und gibt den Namen aus. Der Wert von „Schere“ wird zweimal bestimmt – während des Spiels prüft eine Schleife, ob das fünfte Bild geladen wurde und setzt den Zähler auf 0 zurück, was andernfalls eine Anzeige des fünften Bildes verhindern würde.

```
def number_to_name(number):
    if number == 0:
        return "Stein"
    elif number == 1:
        return "Spock"
    elif number == 2:
        return "Papier"
    elif number == 3:
        return "Echse"
    elif number == -1: ### Wert ist 5, Zurücksetzung
    to 0, zero - 1 = -1 ###
        return "Schere"
    elif number == 4: ### Wert ist 5, Zurücksetzung
    to 0, zero - 1 = -1 ###
        return "Schere" ### für den Computer
```

07 Joystickbewegung erfassen

Erstellen Sie eine weitere Funktion für die Spielmechanik. Fügen Sie zunächst die globale Variable `sense.set_rotation(90)` hinzu, die den Anzeigewinkel des Bildes justiert. Mit `for event in pygame.event.get()`: fragen Sie Joystickbewegungen ab. Der Joystick dient zur Auswahl aus den fünf angezeigten Bildern auf der LED-Matrix. Die Zeile `if event.type == KEYDOWN and playersChoice < 5`: prüft, ob der Joystick nach oben bewegt wurde und die Bildnummer geringer als 5 ist; bei 5 wird auf 0 zurückgesetzt. Ist der Wert `playersChoice` geringer als 5, wird das entsprechende Bild mit dem Befehl `sense.load_image(str(playersChoice) + ".png")` in Zeile 13 an die LED-Anzeige gesendet und die Variable erhöht. Die Schleife startet von vorne, fragt die Joystickbewegung ab und zeigt die jeweilige Bilddatei an.

```
def mainGame():
    ###WAHL DES SPIELERS###
    ###Läuft in Schleife, solange running = True###

    running = True
    global playersChoice
    global computer_choice
    while running == True:
        sense.set_rotation(90)
        for event in pygame.event.get():
            if event.type == KEYDOWN and playersChoice < 5:
                if event.key == K_UP:
```

Inspirierende Pixel

Für das Spiel brauchen Sie gute Grafiken. Johan Vinet (johan-vinet.tumblr.com/image/127476776680) hat einige hervorragende, inspirierende Beispiele 8x8 großer Pixelgrafiken.

Vollständiger Code

```
import pygame
from pygame.locals import *
from sense_hat import SenseHat
import random
import time

global playersChoice
global computer_choice
global count

###Set up PyGame Screen###
pygame.init()
pygame.display.set_mode((140, 180))
```

```
###Prepare Sense Hat###
sense = SenseHat()
sense.load_image("0.png")
sense.low_light = True #save your eyes!

playersChoice = 0
gameRunning = True

'''Converts the Number into the choice i.e.
lizard, spock etc '''
def number_to_name(number):
    if number == 0:
        return "Rock"
    elif number == 1:
        return "Spock"
    elif number == 2:
        return "Paper"
    elif number == 3:
        return "Lizard"
    elif number == -1: ### because value is 5
    so re-sets to 0, zero - 1 = -1 ###
        return "Scissors"
```

```
elif number == 4: ### because value is 5 so
re-sets to 0, zero - 1 = -1 ###
    return "Scissors" ### for the computer

def mainGame():
    ###PLAYER SELECTION###
    ###Loops while running variable is True###
    running = True
    global playersChoice
    global computer_choice
    while running == True:
        sense.set_rotation(90)
        for event in pygame.event.get():

            if event.type == KEYDOWN and
            playersChoice < 5:
                if event.key == K_UP:
                    print (playersChoice)
                    sense.load_image(str(playersChoice) +
                    ".png")
                    playersChoice = playersChoice + 1
                if playersChoice == 5:
                    playersChoice = 0

            '''Checks for a 'select / enter' Choice
            '''
            if event.type == KEYDOWN:
                if event.key == K_RETURN:
                    running = False
                    break
            '''Ends loop and moves onto main
            game'''

            '''Message for player about their choice'''
            #print ("Your Choice is", playersChoice)
            #test
```

Bilder im Format 8x8

Am leichtesten erstellen Sie Ihre Grafiken mit dem Programm RPi Grid Draw, mit dem Sie die LEDs in Echtzeit schalten können. Sie können Farben anpassen, die Bilder drehen und anschließend als Code oder 8x8 große PNG-Dateien speichern. Installieren Sie die Python PNG Library im Terminal mit `sudo pip3 install pypng` und führen Sie dann `git clone https://github.com/jrobinson-uk/RPi_8x8GridDraw` aus. Öffnen Sie nach der Installation das Hauptverzeichnis und geben Sie zum Programmstart `cd Rpi_8x8GridDraw` und `python3 sense_grid.py` ein.

```
print (playersChoice)
sense.load_image(str(playersChoice) + ".png")
playersChoice = playersChoice + 1
if playersChoice == 5:
    playersChoice = 0
```

08 Objekt wählen

Bei der Bildauswahl müssen Sie natürlich nicht nur durchschalten, sondern auch auswählen können. Drücken Sie den Joystick nach unten, entspricht das dem Druck der Eingabetaste. Zeile 3 – `if event.key == K_RETURN`: – reagiert auf die Joystickausrwahl. Beenden Sie anschließend die Auswahlsschleife mit `running = False` und `break` und fahren Sie zum nächsten Element des Spiels fort.

```
"""Prüft, ob eine Auswahl erfolgt"""
if event.type == KEYDOWN:
    if event.key == K_RETURN:
        running = False
        break
```

09 Status aktualisieren

Zeigen Sie zur Bestätigung der Auswahl einen neuen Lauftext auf den LEDs an. Erstellen Sie eine neue Variable namens `number` und verknüpfen Sie sie mit dem Wert von `playersChoice`. Mit der Funktion aus Schritt 06 konvertieren Sie die Zahl, um den Objektnamen zu erhalten, `playerMessage = number_to_name(number)`. Generieren Sie dann den Lauftext mit `sense.show_message(playerMessage, text_colour=[0, 0, 255], scroll_speed = 0.08)`. Ändern Sie nach Belieben die Textfarbe (Werte zwischen 0 und 255) und die Laufgeschwindigkeit des Textes (Werte zwischen 0 und 1).

```
"""Rückmeldung zur Auswahl"""
number = playersChoice - 1
playerMessage = number_to_name(number)
print playerMessage
sense.set_rotation(0)
sense.show_message("Du = ", text_colour=[0, 255, 255], scroll_speed = 0.08)
sense.show_message(playerMessage, text_colour=[0, 0, 255], scroll_speed = 0.08)
```

10 Wahl des Computers

Nun ist der Computer an der Reihe. Er wählt eine zufällige Zahl zwischen 5 und 50, `count = random.randrange(5,50)`. So viele Befehlszyklen dauert es, bis der Computer eine Auswahl trifft – das vermittelt den Eindruck des „Nachdenkens“. Er wählt dann eine weitere Zufallszahl zwischen 0 und 5, `computer_choice = random.randrange(0,5)` und damit auch ein Objekt im Spiel. Der Zähler wird um 1 verringert, der Computer wählt ein neues Bild aus; das geht so weiter, bis der Zähler 0 erreicht. Mit `sense.load_image(str(computer_choice) + ".png")` wird immer das jeweilige Bild angezeigt.

```
###WAHL DES COMPUTERS
"""Computer wählt zufällig eine der
Optionen aus"""
count = random.randrange(5,50)
sense.set_rotation(90)
while count > 1:
    computer_choice = random.randrange(0,5)
    print computer_choice
    time.sleep(0.1)
    sense.load_image(str(computer_choice) + ".png")
    count = count - 1
```

11 Spielstand aktualisieren

Nun muss noch der Spieler informiert werden. Mit der Funktion aus Schritt 06 wird erneut Zahl zu Bild konvertiert: `computerMessage = number_to_name(number)`. Mithilfe von `sense.show_message("Computer = ", text_colour=[0, 150, 255], scroll_speed = 0.06)` zeigen Sie die endgültige Auswahl des Computers an:

```
#time.sleep(1)
"""Information zur Wahl des Computers"""
print ("Der Computer wählt", computer_choice)
number = computer_choice
computerMessage = number_to_name(number)
print computerMessage ##test
sense.set_rotation(0)
sense.show_message("Computer = ", text_colour=[0, 150, 255], scroll_speed = 0.06)
#sense.show_message(computerMessage, text_colour=[0, 0, 255], scroll_speed = 0.08)
sense.load_image(str(computer_choice) + ".png")
```

12 Wer hat gewonnen?

Der Wert der Auswahl des Computers wird nun von dem des Spielers abgezogen und durch fünf geteilt, der Restwert entscheidet das Resultat der Runde. Erstellen Sie eine Variable namens `result` und weisen Sie ihr die Berechnung zu, `result = (int(computer_choice - (playersChoice-1))) % 5`. Ist das Ergebnis 0, haben beide dasselbe Objekt gewählt, es steht unentschieden. Mit dem If-Statement in Zeile 5 prüfen Sie den Wert und zeigen gegebenenfalls die Benachrichtigung `sense.show_message("Unentschieden!", text_colour=[0, 0, 255], scroll_speed = 0.08)` an.

```
###SIEG-BERECHNUNG###
"""Ermittelt den Sieger"""
result = (int(computer_choice - (playersChoice-1))) % 5
#print result
if result == 0:
    sense.show_message("Unentschieden!",
        text_colour=[0, 0, 255], scroll_speed = 0.08)
```

Die Gewinnbedingung erfüllt ein Restwert von größer gleich 3. Mit einem ELSE IF-Befehl können Sie dies implementieren, `elif result >=3`. Jeder andere Wert ist gleichbedeutend mit einem Sieg des Computers. Fügen Sie eine else-Bedingung hinzu und zeigen Sie die Siegbenachrichtigung `sense.show_message("Computer gewinnt!", text_colour=[255, 0, 0], scroll_speed = 0.08)` an.

```
elif result >=3:
    sense.show_message("Spieler gewinnt!", text_
        colour=[0, 255, 0], scroll_speed = 0.08)
else:
    time.sleep(1)
    sense.show_message("Computer gewinnt!", text_
        colour=[255, 0, 0], scroll_speed = 0.08)##??
```

13 Spielintro

Jetzt fehlt noch das Intro, also Begrüßung und Anleitung. Sie könnten in der Hauptschleife Platz finden, aber dann würden Sie vor jedem neuen Spiel angezeigt. Geben Sie `sense.show_message("Hallo bei SPSES!", text_colour=[155, 100, 30], scroll_speed = 0.08)` ein, um den Lauftext zu aktivieren, und laden Sie mit `sense.load_image("0.png")` das erste Bild.



```
###SPIEL START##
```

```
sense.show_message("Hallo bei SPSES!", text_
colour=[155, 100, 30], scroll_speed = 0.08)
sense.show_message("Nach oben für Auswahl", text_
colour=[155, 255, 255], scroll_speed = 0.05)
sense.load_image("0.png")
```

14 Lläuft das Spiel?

Eine while-Schleife prüft, ob das Spiel gerade läuft. Wenn ja, findet momentan eine Partie statt. Setzen Sie die Variable `play_again` auf 1. Damit können Sie später bestimmen, ob es weitergeht oder das Spiel endet. Rufen Sie die in Schritt 07 erstellte Funktion mit der Spielmechanik auf, indem Sie in Zeile 3 `mainGame()` eintragen. Zeigen Sie am Ende der Runde Lauftext an, der nach einer weiteren Runde fragt, `sense.show_message("Noch eine Partie?", text_colour=[255, 255, 255], scroll_speed = 0.08)`.

```
while gameRunning == True:
    play_again = 1

    mainGame()
    sense.show_message("Noch eine Partie?", text_
colour=[255, 255, 255], scroll_speed = 0.08)
```

15 Noch eine Partie?

Eine If-Abfrage ist nötig, um auf diese Frage antworten zu können. Auch hier gilt eine Joystickbewegung nach oben als Bestätigung. Mit `for event in pygame.event.get():` in Zeile 2 fragen Sie einen erfolgten Tastendruck ab, also ob der Joystick nach oben bewegt wurde, `if event.type == KEYDOWN;` `if event.key == K_UP:` (Zeilen 3 und 4). Ist dies der Fall, wird die Variable auf 0 zurückgesetzt (Zeile 5), die If-Abfrage endet und die nächste Partie beginnt.

```
while play_again == 1:
    for event in pygame.event.get():
        if event.type == KEYDOWN:
            if event.key == K_UP:
                play_again = 0
```

16 Spielende

Beendet wird das Spiel folglich mit einer Joystickbewegung nach unten, `if event.key == K_DOWN:` Verabschieden Sie sich mit `sense.show_message("Bis bald!", text_colour=[255, 255, 255], scroll_speed = 0.08)` und setzen Sie die Variable `gameRunning` auf False. Das Spiel kehrt zum Anfang zurück. Da die Schleife aus Schritt 14 jetzt auf False gestellt ist, endet es hiermit, wie im letzten Code-Abschnitt ersichtlich. Viel Spaß beim Spielen!

Vollständiger Code (Fortsetzung)

```
number = playersChoice - 1
playerMessage = number_to_name(number)
print playerMessage
sense.set_rotation(0)
sense.show_message("You = ", text_colour=[0, 255,
255], scroll_speed = 0.08)
sense.show_message(playerMessage, text_colour=[0,
0, 255], scroll_speed = 0.08)

###COMPUTER SELECTION###
'''Computer selects a random choice from the
options'''
count = random.randrange(5,50)
sense.set_rotation(90)
while count > 1:
    computer_choice = random.randrange(0,5)
    print computer_choice
    time.sleep(0.1)
    sense.load_image(str(computer_choice) + ".png")
    count = count - 1

    '''Message for player about the computer's
choice'''
    print ("The computers choice is", computer_choice)
    number = computer_choice
    computerMessage = number_to_name(number)
    print computerMessage ##test
    sense.set_rotation(0)
    sense.show_message("Computer = ", text_colour=[0,
150, 255], scroll_speed = 0.06)
    #sense.show_message(computerMessage, text_
colour=[0, 0, 255], scroll_speed = 0.08)
    sense.load_image(str(computer_choice) + ".png")

    print computerMessage
    time.sleep(1)

###WINNER CALCULATED###
'''Calculates the Winner'''
result = (int(computer_choice - (playersChoice-1)))
% 5
```

```
if result == 0:
    sense.show_message("Player and Computer Tie!",
text_colour=[0, 0, 255], scroll_speed = 0.08)
    #print "tie"
elif result >=3:
    sense.show_message("Player Wins!", text_
colour=[0, 255, 0], scroll_speed = 0.08)
    #print "Player wins!"
else:
    time.sleep(1)
    sense.show_message("Computer Wins!", text_
colour=[255, 0, 0], scroll_speed = 0.08)###??
    #print "Computer wins!"
    print " "

###START THE GAME##
sense.show_message("Welcome to RPSLS!", text_
colour=[155, 100, 30], scroll_speed = 0.08)
sense.show_message("Pleases use 'Up' to select", text_
colour=[155, 255, 255], scroll_speed = 0.05)
sense.load_image("0.png")

while gameRunning == True:
    play_again = 1

    mainGame()
    sense.show_message("Play Again?", text_colour=[255,
255, 255], scroll_speed = 0.08)
    while play_again == 1:
        for event in pygame.event.get():
            if event.type == KEYDOWN:
                if event.key == K_UP:
                    play_again = 0

            if event.type == KEYDOWN:
                if event.key == K_DOWN:
                    print("Bye")
                    sense.show_message("Bye Bye", text_
colour=[255, 255, 255], scroll_speed = 0.08)
                    play_again = 0
                    gameRunning = False
```

Umweltwissenschaft mit dem Sensly HAT

Mit diesem cleveren kleinen Modul für Ihren Raspberry Pi können Sie Experimente durchführen, Schadstoffe überwachen und anderes.

Das Sensly ist ein smartes Modul, das am Raspberry Pi befestigt werden kann. Es besitzt einen eigenen Mikroprozessor, der in der Lage ist, analoge Daten zu verarbeiten und Proben von verschiedenen Sensoren aufzunehmen, ohne die Rechenleistung des Pi zu verschwenden. Am Modul selbst sind drei verschiedene Gassensoren angebracht, die eine Vielzahl von Gasen erkennen können. Auch Temperatur und Luftfeuchtigkeit können gemessen werden. Hinzu kommt, dass das HAT durch seine Vielzahl an analogen Eingängen und eine I2C-Schnittstelle leicht erweiterbar ist.

Wie vom Pi gewohnt, stellt das Modul eine Python-API bereit, welche die Arbeit für Sie erledigt. Die Berechnungen, Kalibrierungen und Fehlerbehebungen werden von der API selbst gelöst, Sie erhalten also sofort relevante Daten.

01 Sensly HAT einrichten

Diese Anleitung geht davon aus, dass Sie Ihren Pi eingerichtet haben und dieser mit dem Internet verbunden ist. Stecken Sie zunächst Ihr Sensly HAT in das Verteilerstück des Pi ein. Haben Sie dies getan, schalten Sie den Pi an und geben Sie die folgenden Befehle im Terminal ein:

```
# Installation von git, um API laden zu können
sudo apt-get install git
git clone https://github.com/Altitude-Tech/SenslyPi.git
cd SenslyPi
# Install the python API
sudo python setup.py install
```

Was Sie brauchen

■ Sensly HAT
altitude.tech

```
pi@ubuntu: ~
pi@ubuntu:~$ ./sensly.py
Running Sensly Test . . . . .
Sensly Was Detected!
Testing Sensors . . . . .
-OK-
```

02 Sensly HAT testen

Stellen Sie zunächst sicher, dass Ihr Pi mit dem Sensly HAT kommunizieren kann und richtig funktioniert. Geben Sie den nachfolgenden Befehl ein. Funktioniert alles, erhalten Sie die Ausgabe „OK“ und einen Bericht.

```
# Start der Test-Sequenz auf dem Sensly
sensly-test
```





03 API verwenden

Hier verwenden wir einen terminalbasierten Texteditor, um etwas Code einzugeben (wir benutzen Nano, aber jeder andere Editor tut auch seinen Zweck). Der folgende Befehl erstellt eine Datei namens `sensly.py` und öffnet diese zur Bearbeitung.

```
nano sensly.py
```

Mit folgendem Code erhalten Sie grundlegende Daten des Sensly:

```
import os
from sensly import Gases

atmosphere = Gases()

while True:
    print("Feuchtigkeit:")
    print( str(atmosphere.humidity()) )
    print("Temperatur:")
    print( str(atmosphere.temp()) )
    time.sleep(5)
```

04 Einsatz der Gas-Sensoren

Der Einsatz der Sensoren gestaltet sich komplizierter, da diese erst nach einer Aufwärmphase stabile Messwerte ausgeben. Wir bauen daher eine Schleife in den Code ein, die den Abschluss des Aufwärmvorgangs abwartet. Falls Sie das Sensly gerade erst angeschaltet haben, dauert dieser Vorgang zwei bis drei Minuten.

```
import os
from sensly import Sensly

sensly = Sensly()

# Ende des Aufwärmvorgangs abwarten
# für verlässliche Messwerte
While (sensly.preheated() == False):
    print("Heizt auf...")
    time.sleep(1)

print("Sensly kann jetzt Verschmutzungsdaten auslesen!")
```

05 Verschmutzungsdaten auslesen

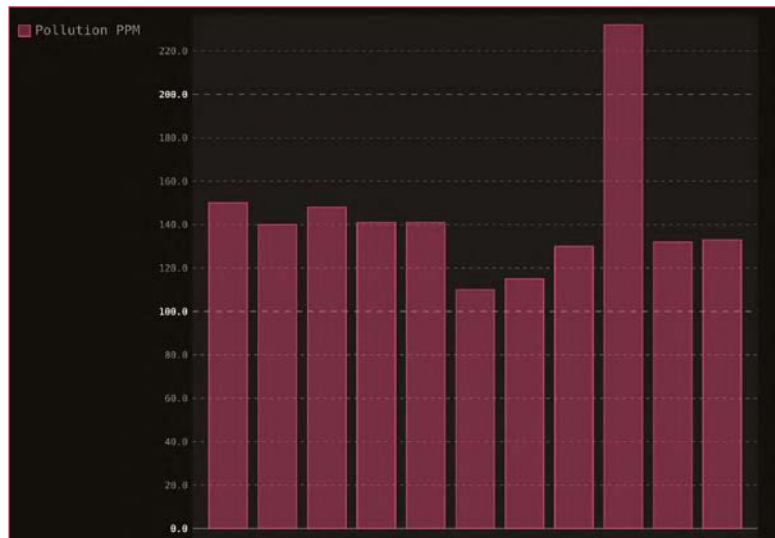
Der korrekte Einsatz der Gas-Sensoren wäre also sichergestellt – jetzt können Sie mit den Messungen beginnen. Fügen Sie diesen Code zum vorher eingegebenen Programm hinzu:

```
While(True):
    # Verschmutzungsgrad ausgeben
    print("Allgemeine Verschmutzung (PPM)")
    print( str( atmosphere.pollution() ) )

    #Industrielle Verschmutzung ausgeben
    # (z.B. Benzen, Stickoxid)
    print("Industrielle Verschmutzung (PPM)")
    print( str( atmosphere.pollution.industrial() ) )

    #Feuchtigkeit ausgeben
    print("Feuchtigkeit")
    print( str( atmosphere.humidity() ) )

    time.sleep(20)
```



06 Eigene Sensoren anschließen

Das Sensly stellt fünf Eingänge für zusätzliche Sensoren bereit, welche leicht mit einem simplen Python-Skript kontrolliert werden können. Diese Anschlüsse sind besonders nützlich, da der Pi selbst keine eigenen analogen Anschlüsse bereitstellt. So können Sie zusätzliche Sensoren hinzufügen, die sich an Ihr jeweiliges Projekt anpassen – beispielsweise einen Fotowiderstand, um Lichtverschmutzung zu messen. Mit dem folgenden Skript können Sie leicht analoge und digitale Daten von den Erweiterungsanschlüssen ablesen:

```
import sensly
from sensly import Port

sensor1 = Port(Sensly.port1)

print( str( sensor1.analog_read() ) )
```

07 Daten visualisieren

Hierfür verwenden wir ein Open-Source-Projekt namens `pygal`, um ganz einfach Vektordiagramme aus Ihren Daten zu generieren. Installieren Sie `pygal` und geben Sie dann Code ein, mit dem Sie weitere Informationen zur Luftverschmutzung gewinnen. Führen Sie dazu diese Befehle durch:

```
sudo apt-get install python-setuptools
sudo easy_install pygal
```

Der folgende Code misst zehn Stunden lang die Luft und generiert ein Vektordiagramm daraus. Die Gesamtzeit kann angepasst werden.

```
import pygal
import os
from sensly import Gases

atmosphere = Gases()
graph = pygal.Bar()

samples = []
for i in range(10):
    samples.append(atmosphere.pollution.industrial())

graph.add('pollution', samples)
graph.render_to_file("pollution.svg") # Graph erstellen
```

Oben Pygal ist eine dynamische Sammlung skalierbarer Vektordiagramme, die Python/CSS-Stile anbietet.

Freischalten des ARM-Kortex

Das Gehirn dieser Aufsteckplatine ist ihr ARM-Kortex, welcher alle analogen Signale, Datenproben und Kommunikationen zum Pi verarbeitet. Dies ist ein leistungsfähiger kleiner Chip mit vielen Funktionen und einer Open-Source-Firmware. Entsprechend schnell kann man das HAT vom Pi aus umprogrammieren, was Ihnen die Flexibilität eines Mikrocontrollers wie dem Arduino gibt, aber alle Vorteile eines kompletten Betriebssystems bietet – ein wirklich schönes Spielzeug für fortgeschrittene Basteleien.



Was Sie brauchen

- Raspberry Pi 2
- USB-Soundkarte (beispielsweise Behringer UCA202)

Der Pi als Synthesizer

Lernen Sie, einen einfachen polyphonen Synthesizer mittels Python und Cython zu programmieren.

Wir möchten Ihnen in diesem Artikel das Basiswissen der Wavetable-Synthese vermitteln und zeigen, wie Sie damit einen Echtzeit-Synthesizer in Python programmieren. Hierbei wird der Synthesizer über das Keyboard Ihres Computers angesteuert, aber es ist mit ein paar einfachen Schritten möglich, auch ein MIDI-Keyboard anzuschließen.

Python ist für einen polyphonen (also gleichzeitig mehrere Noten abspielenden) Synthesizer leider zu langsam, daher müssen wir auf Cython zurückgreifen, welches Python in C kompiliert, sodass Sie den Code letztendlich in Maschinsprache umwandeln, um die Performance zu verbessern. Unser Ziel ist die Polyphonie von drei Noten. Das reicht zwar nicht für einen voll ausgewachsenen Synthesizer, doch das Ziel dieses Tutorials ist es auch nur, Sie mit den grundlegenden Mechanismen der Wavetable-Synthese in einer zugänglichen Programmiersprache, nämlich Python, vertaut zu machen.

Haben Sie die ersten Schritte gemeistert, können Sie das Projekt erweitern, indem Sie das Tastenlayout an Ihr Keyboard anpassen oder den Code tweakten, um ein MIDI-Keyboard anzuschließen.

01 Pakete installieren

Installieren Sie mit dem aktuellen Raspbian-Image die benötigten Pakete mittels der folgenden Befehle:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install python-pip python2.7-dev
portaudio19-dev
sudo pip install cython pyaudio
```

Im letzten Schritt werden Cython und PyAudio kompiliert. Dies dauert eine gewisse Zeit.

02 Eingebaute Soundkarte deaktivieren

Da wir einige Probleme hatten, die eingebaute Soundkarte des Raspberry Pi mit unserem programmierten Code ans Laufen zu bringen, verwenden wir eine USB-Soundkarte und deaktivieren einfach die interne Karte:

```
sudo rm /etc/modprobe.d/alsa*
```




Cython

Cython ist ein Werkzeug, mit dem Sie Python in einen C-Code kompilieren können, der vom Interpreter benötigt wird, um das Programm auszuführen. Dies hat den Vorteil, dass Sie so Teile Ihres Python-Codes in puren C-Code optimieren können, was die Geschwindigkeit enorm steigert. Dies wird dadurch bewerkstelligt, dass C-Typen wie zum Beispiel int, float und char Python-Variablen zugeordnet werden.

Wenn Sie einmal den C-Code haben, können Sie ihn weiter mit einem C-Compiler (zum Beispiel GCC) bearbeiten, wodurch der Code sogar noch weiter optimiert wird. Ein Nachteil der Verwendung von Cython ist, dass ein gewöhnlicher Python-Interpreter keinen von Cython optimierten Code ausführen kann. Cython ist dennoch ein guter Kompromiss, da es die Einfachheit des Python-Codes mit einer höheren Performance verbindet. Cython hat einen Profiler, den Sie mit folgendem Befehl ausführen:

```
python -a synth.pyx
```

Der Profiler gibt eine HTML-Datei aus, die Ihnen einen Überblick gibt, wieviel Overhead Python generiert und wo Sie überall den Code optimieren können. Auf cython.org finden Sie weitere Infos.

```
sudo editor /etc/modules
```

Ändern Sie „snd-bcm2835“ zu „#snd-bcm2835“ und speichern Sie, danach:

```
sudo reboot
```

03 Soundkarte testen

Jetzt können wir die Funktionsfähigkeit der USB-Soundkarte überprüfen. Geben Sie „alsamixer“ ein und vergewissern Sie sich, dass die Lautstärke auf einen angenehmen Pegel eingestellt ist. Sollten Sie Lautsprecher anschließen wollen, ist eine Lautstärke von 100 % ratsam. Geben Sie im Anschluss „speaker-test“ ein, womit ein rosa Rauschen in den Lautsprechern erzeugt wird. Mit Strg+C wird die Tonausgabe unterbrochen, sobald Sie mit den Einstellungen zufrieden sind.

04 Projekt beginnen

Erstellen Sie nun ein Verzeichnis für Ihr Projekt. Laden Sie im Anschluss die Kurve einer rechteckigen Welle (Square Wave), die wir für den Wavetable brauchen, herunter:

```
mkdir synth
cd synth
wget liamfraser.co.uk/lud/synth/square.wav
```

05 Kompilations-Skript erstellen

Wir brauchen ein Skript, das Ihren Python-Code profilert, einen Cython-Code daraus generiert und schließlich in Binärsprache mit GCC umwandelt.

```
editor compile.sh:
#!/bin/bash
cython -a synth.pyx
cython --embed synth.pyx
gcc -march=armv7-a -mfpu=neon-vfpv4 -mfloat-abi=hard -O3 -I /usr/include/python2.7 -o synth.bin synth.c -lpython2.7 -lpthread
```

Beachten Sie die Einstellungen, mit denen der Compiler Fließkommastellen verwendet. Machen Sie das Skript ausführbar mit:

```
chmod +x compile.sh
```

Vollständiger Code

```
#!/usr/bin/python2
```

```
import pyaudio
import time
from array import *
from cpython cimport array as c_array
import wave
import threading
import tty,termios
```

Schritt 07

```
class MIDITable:
    # Generierungs-Code von
    # http://www.adambuckley.net/software/beep.c

    def __init__(self):
        self.notes = []
        self.fill_notes()

    def fill_notes(self):
        # Frequenz der MIDI-Note 0 in Hz
        frequency = 81.75799

        # Verhältnis: 2 hoch 1/12
        ratio = 1.0594631

        for i in range(128):
            self.notes.append(frequency)
            frequency = frequency * ratio

    def get_note(self, n):
        return self.notes[n]
```

Schritt 08

```
class ADSR:
    cdef float attack, decay, sustain, amplitude
    cdef float release, multiplier
    cdef public char state
    cdef int samples_per_ms, samples_gone

    def __init__(self, sample_rate):
        self.attack = 1.0/100
        self.decay = 1.0/300
        self.sustain_amplitude = 0.7
        self.release = 1.0/50
        self.state = A
        self.multiplier = 0.0
        self.samples_per_ms = int(sample_rate/1000)
        self.samples_gone = 0

    def next_val(self):
        self.samples_gone += 1
        if self.samples_gone > self.samples_per_ms:
            self.samples_gone = 0
        else:
            return self.multiplier

    if self.state == A :
        self.multiplier += self.attack
        if self.multiplier >= 1:
            self.state = D
    elif self.state == D :
        self.multiplier -= self.decay
        if self.multiplier <= self.sustain_amplitude:
            self.state = S
    elif self.state == R :
        self.multiplier -= self.release

    return self.multiplier
```

Vollständiger Code (Fortsetzung)

Schritt 09

```

cdef class Note:
    cdef int wavetable_len
    cdef float positionstep_size
    cdef c_array.array wavetable
    cdef public float freq
    cdef public object adsr
    cdef public int off

    def __init__(self, wavetable, samplerate, freq):
        # Referenz des genutzten Wavetables
        self.wavetable = wavetable
        self.wavetable_len = len(wavetable)
        # Frequenz in Hz
        self.freq = freq
        # Die Schritte durch den Wavetable um
        # die gewünschte Frequenz zu erhalten
        self.step_size = self.wavetable_len * \
            (freq / float(samplerate))
        # Position im Wavetable
        self.position = 0.0
        # ADSR-Instanz
        self.adsr = ADSR(samplerate)
        # Wird diese Note erzeugt mit
        self.off = 0

    def __repr__(self):
        return ( Note: Frequency = {0}Hz,
                Step Size = {1} ).format(self.freq,
                self.step_size)

    cpdef int next_sample(self):
        # Noch ein Beispiel
        cdef int pos_int, p2, interpolated
        cdef int out_sample = 0
        cdef float pos_dec
        cdef float adsr

        adsr = self.adsr.next_val()
        # Muss die Note stummschalten
        # Synthi entfernt nächstes Sample
        if adsr < 0:
            self.off = 1
            return out_sample

        pos_int = int(self.position)
        pos_dec = self.position - pos_int

        # Lineare Interpolation
        p1 = self.wavetable[pos_int]
        p2 = 0

        # Übertrag, falls erste Position
        # am Ende des Wavetables ist
        if pos_int + 1 == self.wavetable_len:
            p2 = self.wavetable[0]
        else:
            p2 = self.wavetable[pos_int + 1]

        # Zwischen p1 und p2 interpolieren
        interpolated = int(p1 + ((p2 - p1) * pos_dec))
        out_sample += int(interpolated * adsr)

        # Schrittgröße erhöhen und übertragen, falls
        # über Ende des Wavetables hinausgehend
        self.position += self.step_size
        if self.position >= self.wavetable_len:
            self.position -= self.wavetable_len

```

06 Fangen Sie an zu programmieren

Unsere Code-Datei erhält den Namen `synth.pyx`. Mit dieser Erweiterung zeigen Sie Cython an, dass es sich hier nicht um gewöhnlichen (und somit nicht im normalen Python-Interpreter ausführbaren) Python-Code handelt. Erstellen Sie die Datei mit einem Editor und fügen Sie die Importe hinzu.

07 MIDI-Tabelle

Um eine gewöhnliche Note eines Klaviers zu synthetisieren, brauchen wir eine Tabelle von MIDI-Werten. Noten in MIDI reichen von 0 bis 127. Der Wert 60 ist das mittlere C auf einem Klavier. Die Klasse für die MIDI-Tabelle hat eine „get note“-Funktion, welche die Frequenz einer Note zurücksendet, wenn Sie Ihr eine MIDI-Notennummer geben.



Oben Visuelle Darstellung einer Attack/Decay/Sustain/Release-Kurve.

08 Attack, Decay, Sustain, Release

Die ADSR-Klasse fügt der rohen Ausgabe eines Oszillators eine zeitlich begrenzte Volumenkurve hinzu. Dies wird durch das Hinzufügen eines Multiplikators im Bereich zwischen 0,0 und 1,0 zu der Note erreicht. In unserer Version liegt der Attackwert bei 100 ms, der Decaywert bei 300 ms und der Releasewert bei 50 ms. Sie können mit den Werten herumexperimentieren, um zu sehen, wie diese den Klang verändern. Die ADSR-Klasse führt eine Menge Berechnungen durch (44.100 pro Note und Sekunde). Daher wollen wir jede Variable einem Typ zuordnen, sodass die Berechnungen in einer C-Schleife, soweit möglich, optimiert werden, da Python im Vergleich zu C viel mehr Overhead hat. Hierfür wird das Schlüsselwort `cdef` benutzt. Mit `cdef public` kann die Variable auch von Python heraus verwendet werden.

09 Noten generieren

Die Noten-Klasse ist das Herz unseres Synthesizers. Sie verwendet den Wavetable, um Tonwellen einer spezifischen Frequenz zu generieren. Der Synthesizer fragt die Noten-Klasse nach einem Sample. Nachdem das Sample generiert worden ist, wird der ADSR-Multiplikator hinzugefügt und zum Synthesizer zurückgeschickt. Die Berechnungen werden in dem Kasten zur Synthesetheorie auf der gegenüberliegenden Seite erklärt.

Die Noten-Klasse benötigt genauso viele Berechnungen wie die ADSR-Klasse, sodass diese mit den `cdef`-Schlagwörtern so gut wie möglich optimiert wird. Das `cpdef`-Schlagwort, welches für die `next_sample`-Funktion verwendet wird, hat die Bedeutung, dass die Funktion auch von einer Nicht-`cdef`-Klasse aufgerufen werden kann. Die Main-Synth-Klasse ist leider zu kompliziert, als dass man statische Typen verwenden könnte.

10 Der Audiofluss

Die Synth-Klasse ist die wichtigste Klasse der Anwendung. Sie hat zwei Puffer für die Samples im Bereich der Puffergröße. Während einer der Puffer von der Soundkarte abgespielt wird, wird der andere von einem anderen Thread aufgefüllt. Hat die Soundkarte den Puffer abgespielt, wird eine callback-Funktion ausgeführt. Die Zuweisungen der beiden Puffer werden ausgetauscht und der gerade aufgefüllte Puffer wird wieder in die Audiobibliothek zurückgeschickt.

Je geringer die Größe des Puffers ist, desto geringer ist die Latenz. Da das Raspbian-Image nicht für die Echtzeitwiedergabe von Audio optimiert ist, können Sie Probleme mit kleinen Puffergrößen bekommen. Das hängt aber auch von der verwendeten USB-Soundkarte ab.



Vollständiger Code (Fortsetzung)

Schritt 09 `return out_sample`

```
class Synth:
    BUFSIZE = 1024
    SAMPLERATE = 44100

    def __init__(self):
        self.audio = pyaudio.PyAudio()

        # Output-Buffer erstellen
        self.buf_a = array( 'h', [0] * Synth.BUFSIZE )
        self.buf_b = array( 'h', [0] * Synth.BUFSIZE )
        # Oldbuf und curbuf sind Referenzen,
        # keine Kopien; newbuf wird befüllt,
        # während playbuf abgespielt wird
        self.playbuf = self.buf_b
        self.newbuf = self.buf_a

        self.load_wavetable()
        self.notes = []
        self.notes_on = []

        # Der Synthi-Loop läuft in einem eigenen
        # Thread; mithilfe dieser Bedingung wird er
        # benachrichtigt, wenn mehr Samples nötig sind
        self.more_samples = threading.Event()
        self.exit = threading.Event()

        # MIDI-Tabelle der Noten -> Frequenzen
        self.midi_table = MIDITable()

    def stop(self):
        print Exiting
        self.exit.set()
        self.stream.stop_stream()
        self.stream.close()

    def stream_init(self):
        self.stream = self.audio.open(
            format = pyaudio.paInt16,
            channels = 1,
            rate = Synth.SAMPLERATE,
            output = True,
            frames_per_buffer = Synth.BUFSIZE,
            stream_callback = self.callback)

    def load_wavetable(self):
        # Wavetable laden; sicherstellen, dass
        # das Format korrekt ist
        fh = wave.open( 'square.wav', 'r' )
        assert fh.getnchannels() == 1
        assert fh.getframerate() == Synth.SAMPLERATE
        assert fh.getsampwidth() == 2 # aka 16 bit

        # Wave-Daten als Byte String lesen; dann
        # in Sample Array umwandeln, das per Index
        # erreicht werden kann
        data = fh.readframes(fh.getnframes())
        # h steht für int16_t
        self.wavetable = array( 'h' )
        self.wavetable.fromstring(data)

    def swap_buffers(self):
        tmp = self.playbuf
        self.playbuf = self.newbuf
        self.newbuf = tmp
        # Mit der Bedingung entsteht der Loop
```

Synthesetheorie

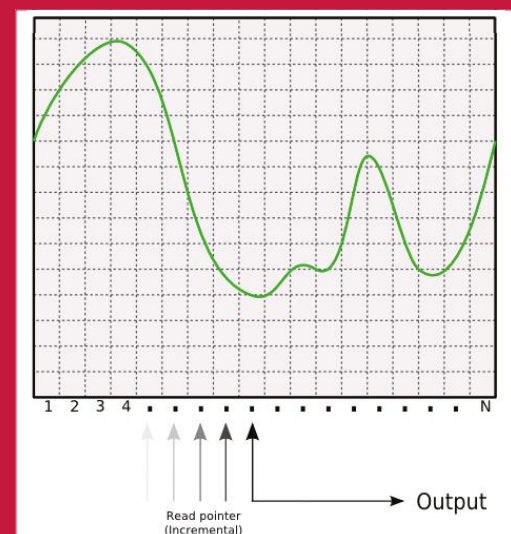
In der Wavetable-Synthese verwendet man eine einzelne Kurve einer Welle als Umsetzungstabelle, um einen Ton zu erschaffen.

In unserem Fall nehmen wir eine rechteckige Welle, Sie können sich jedoch auch für jede andere Wellenform entscheiden. Audio in CD-Qualität wird in 44.100 Hz gesampelt, was auch wir benutzt haben. Bei jedem Sample gibt der Synthesizer einen Wert der Tabelle aus und begibt sich zum nächsten Wert. Wenn der Wavetable eine Frequenz von 440 Hz hat, dann müssen wir in der Lage sein, ihn in willkürlichen Schritten zu durchschreiten. Um dies zu erreichen, nutzen wir lineare Interpolation. Ausgehend von einer Frequenz von 440 Hz wollen wir nun eine Frequenz von 220 Hz haben. Daher müssen wir den Wert mit 0,5 multiplizieren. Diese Vorgehensweise ist vergleichbar mit dem Zeichnen einer Linie zwischen den zwei Werten der Tabelle, wobei der Wert auf der Linie dann ausgegeben wird.

Hier ein Beispiel: Wenn Faktor 0 gleich 5 und Faktor 1 gleich 10 ist, dann würde Folgendes für den Faktor 0,5 gelten: $5 + ((10 - 5) * 0,5)$. Der gesuchte Wert wäre in diesem Fall 7,5. Erreichen Sie einen Punkt in der Tabelle, der über diese hinausgeht, fangen Sie wieder am Anfang an. Da Sie eine einzelnen Kurve der Welle im Wavetable gespeichert haben, gibt es keine Unterbrechungen. Die Gleichung für die Schrittgröße lautet:

$$\text{Schrittgröße} = \text{Tabellengröße} * (\text{Noten-Frequenz} / \text{Sample-Rate})$$

Der Wavetable-Oszillator erzeugt eine Note der gewünschten Frequenz, jedoch wird diese immer in der maximalen Amplitude ausgegeben und klingt daher roh und unnatürlich. Schneiden Sie die Welle in der Mitte ihrer Kurve ab, so werden Sie ein Klicken oder Ploppen hören. Aus diesem Grund werden Attack-, Decay-, Sustain- und Release-Envelopes verwendet. Hierdurch wird die Amplitude der rohen Ausgabe des Oszillators mit der Zeit verändert, um mehr nach einem natürlichen Instrument zu klingen. Dies wird durch das Hinzufügen eines fraktionellen Multiplikators zum ursprünglichen Sample bewerkstelligt, welches zum Wavetable-Oszillator zurückgesendet wird. Wird ein Release in einem Zeitraum programmiert, bei dem sich die Lautstärke von 100 % auf 0 % senkt, wird die Note langsam ausgeblendet. Mit den richtigen ADSR-Kurven und einem vernünftigen Wavetable ist es durchaus möglich, echte Instrumente nachzuempfinden. Unter bit.ly/1KgI9dp erhalten Sie weitere Informationen.



Oben Hier sehen Sie eine Kurve des Wavetable-Oszillators.

Leistungs- probleme

Python führt zu einigen Leistungsproblemen im Vergleich mit anderen nativen Synthesizern, die in C oder C++ geschrieben wurden. Bei unserem Programm haben wir Cython verwendet, um die Probleme so gut es geht zu entschärfen, trotzdem reicht es nicht ganz. Hier nur ein grobes Beispiel: Unser Experte arbeitete an einem Synthesizer-Projekt in C, welches auf einen 100-MHz-ARM-Prozessor ausgelegt war, und erreichte damit eine Polyphonie von 30 Noten. In unserem Projekt erreichten wir mit einem 900-MHz-ARM-Kern gerade mal 3 Noten.

Ein Hauptproblem ist, dass die Soundkarte 16-Bit-Werte verwendet, um ein Sample darzustellen, was von Python leider nicht unterstützt wird. Um die Daten an die Audiobibliothek weiterzuleiten, müssen diese von Integer Arrays in einen Byte String umgewandelt werden. Auf der anderen Seite muss jedoch Python die Daten wieder von Byte Strings in Integer Arrays konvertieren. Wäre der Code in einer Sprache wie C++ oder Rust geschrieben worden, wäre es möglich gewesen, die Daten nahezu direkt an die Soundkarte weiterzugeben.

Ein weiteres Problem ist, dass Python einen großen Overhead beim Aufrufen von Funktionen hat. In kompilierten Sprachen kann so etwas optimiert werden, indem die Funktion in die Zeile des Aufrufers kompiliert wird; der Code der Funktion wird hierfür einfach in den Aufrufer kopiert. Der Zugriff auf die Variablen erzeugt auch einen Overhead, da eine Überprüfung der Typen erforderlich ist. Weiterhin gibt es auch einen Overhead des Garbage Collectors, welcher Objekte löscht, die nicht mehr wichtig sind.

Ein großes Problem ist, dass eine Soundkarte 16-Bit-Integer verwendet, um ein Sample darzustellen. Python unterstützt diesen Typ jedoch nicht.

Vollständiger Code (Fortsetzung)

```
# Weitere Samples erzeugen
self.more_samples.set()

def callback(self, in_data, frame_count,
               time_info, status):
    # Soundkarte braucht mehr Samples, darum
    # Puffer tauschen und anschließend den
    # gerade befüllten Puffer abspielen
    self.swap_buffers()
    return (self.playbuf.tostring(),
            pyaudio.paContinue)

Schritt 11
def do_sample(self, int i):
    cdef int out_sample = 0
    # Durch alle Noten gehen; jede Note
    # dem Sample hinzufügen
    for note in self.notes:
        if note.off:
            self.notes.remove(note)
        else:
            out_sample += note.next_sample() >> 3

    self.newbuf[i] = out_sample

def synth_loop(self):
    cdef int i

    while self.exit.is_set() == False:
        # Für jedes Sample erzeugen
        for i in range(1, Synth.BUFSIZE):
            self.do_sample(i)

        # Auf Benachrichtigung warten, dass
        # neue Samples erzeugt werden sollen
        self.more_samples.clear()
        self.more_samples.wait()

def start(self):
    self.stream_init()
    # Synth-Loop-Thread starten
    t = threading.Thread(target=self.synth_loop)
    t.start()

Schritt 12
def freq_on(self, float freq):
    n = Note(self.wavetable.Synth.SAMPLERATE,
             freq)
    print n
    self.notes.append(n)

def freq_off(self, float freq):
    # ADSR-Status auf Release setzen
    for n in self.notes:
        if n.freq == freq:
            n.adsr.state = ord( R )

def note_on(self, n):
    self.freq_on(self.midi_table.get_note(n))
    self.notes_on.append(n)
```




11 Die Synth-Schleife

Die Start-Methode der synth-Klasse initialisiert die Audiohardware und startet dann die synth_loop-Methode in ihrem eigenen Thread. Wenn das Exit-Ereignis auf falsch gesetzt ist, wird die do_sample-Funktion aufgerufen.

Die do_sample-Funktion arbeitet die aktivierten Noten in einer Schleife ab und fragt nach einem Sample von jeder Note. Die Samples werden um 3 nach rechts verschoben (also durch 2^3 geteilt) und dann zu out_sample hinzugefügt. Die Division sorgt dafür, dass es im Output-Sample zu keinem Overflow kommt – wenngleich dies eine ziemlich primitive Herangehensweise ist, um Noten zusammenzusetzen, funktioniert sie dennoch.

Das daraus resultierende Sample wird im Anschluss in den Sample-Puffer geladen. Ist der Puffer voll, wird die more_samples-Kondition gelöscht und wartet der synth_loop-Thread darauf, darüber unterrichtet zu werden, dass der gerade erschaffene Puffer zur Soundkarte gesendet wurde. Jetzt kann der Synthesizer den Puffer füllen, der gerade eben abgespielt wurde, und der Kreislauf beginnt von vorn.

12 Noten einschalten

Es gibt sowohl note_on/off- als auch freq_on/off-Funktionen, mit denen MIDI-Noten oder willkürliche Frequenzen aktiviert werden. Zusätzlich gibt es noch eine toggle-note-Funktion. Diese behält den Überblick darüber, welche Noten aktiviert sind, und schaltet sie gegebenenfalls aus. Die toggle-note-Methode wird speziell für die Keyboardeingabe verwendet.

13 Ein Keyboard ansteuern

Für eine Keyboardeingabe benötigen wir die Fähigkeit, die Eingabe eines einzelnen Buchstabens vom Bildschirm zu bekommen. In Python's Eingabe-Code ist es vorgesehen, dass etwas eingegeben wird, bevor es zum Programm zurückgeschickt wird. Unser Code für diesen Zweck wurde von dem Modul unter code.activestate.com/recipes/577977-get-single-keypress/ inspiriert.

Es gibt eine Zuordnung von Buchstaben zu MIDI-Noten für den Bereich einer gesamten Oktave. Wir haben bei unserer Version versucht, den Abstand der Buchstabentasten zugunsten einer besseren Spielbarkeit dem der Klaviertasten nachzuempfinden.

14 Alles zusammenfügen

Die main-Funktion des Programms erstellt eine Instanz der Synth-Klasse und startet im Anschluss einen synth-loop-Thread und Audiostream. Die start-Funktion gibt die Kontrolle wieder an den main-Thread zurück. Jetzt erschaffen wir eine Instanz der KB-input-Klasse und gehen in eine Schleife, die einen Buchstaben bekommt und die dementsprechende MIDI-Note ein- oder ausschaltet. Drückt der Anwender auf Q, wird der Synthesizer gestoppt und die Eingabeschleife unterbrochen. Das Programm wird daraufhin geschlossen.

15 Code kompilieren

Verlassen Sie nun den Editor und starten Sie das compile-Skript, indem Sie Folgendes eingeben:

```
./compile.sh
```

Der Vorgang dauert ungefähr eine halbe Minute. Ist das Kompilieren abgeschlossen, führen Sie synth.bin aus:

```
./synth.bin
```

Wenn Sie die Tasten A bis K drücken, werden die weißen Tasten eines Klaviers emuliert. Drücken Sie die Tasten erneut, wird der Ton wieder ausgeschaltet.

Vollständiger Code (Fortsetzung)

Schritt 12

```
def note_off(self,n):
    self.freq_off(self.midi_table.get_note(n))
    self.notes_on.remove(n)

def toggle_note(self,n):
    if n in self.notes_on:
        print note('{} off'.format(n))
        self.note_off(n)
    else:
        print note('{} on'.format(n))
        self.note_on(n)
```

Schritt 13

```
class KBInput:
    def __init__(self,synth):
        self.synth = synth

    self.keymap = { a : 60, w : 61, s : 62,
                   e : 63, d : 64, f : 65,
                   t : 66, g : 67, y : 68,
                   h : 69, u : 70, j : 71,
                   k : 72 }
    self.notes_on = []

    @staticmethod
    def getch():
        fd = sys.stdin.fileno()
        old_settings = termios.tcgetattr(fd)
        try:
            tty.setraw(fd)
            ch = sys.stdin.read(1)
        finally:
            termios.tcsetattr(fd,termios.TCSADRAIN,
                             old_settings)
        return ch

    def loop(self):
        while True:
            c = self.getch()

            if c == 'q':
                self.synth.stop()
                return

            if c in self.keymap:
                n = self.keymap[c]
                self.synth.toggle_note(n)

if __name__ == '__main__':
    s = Synth()
    s.start()
    kb = KBInput(s)
    kb.loop()
```

```
note 60 on
Note: Frequency = 261.625640869Hz, Step Size = 0.599187970161
note 60 off
note 64 on
Note: Frequency = 329.627685547Hz, Step Size = 0.754929602146
note 64 off
note 65 on
Note: Frequency = 349.228363037Hz, Step Size = 0.799820065498
note 65 off
note 67 on
Note: Frequency = 391.995574951Hz, Step Size = 0.897767663002
note 67 off
Exiting
```

Oben Das Interface: Die Schrittgröße im Wavetable variiert mit der Frequenz.

Vernetzte Sensoranzeige mit Pimoroni Scroll pHAT

Verbinden Sie sich mit bis zu zwei Raspberry Pis, um ein Live-Netzwerk-Sensor-Display zu erstellen.

Wir werden mit dem Pimoroni Scrollphat eine vernetzte Sensoranzeige erstellen. Wir sehen uns an, welche Hardware enthalten ist und wie man die benötigten Python-Bibliotheken installiert. Dann arbeiten wir den Quellcode der Bibliothek durch und programmieren einige Beispiele, die die Open-Source-Community über GritHub beigesteuert hat. Sobald wir mit der Bibliothek vertraut sind, werden wir den ausgewählten Sensor auf dem ersten Raspberry Pi einrichten und den Bildschirm mit unserem Pi Zero verbinden. Wir lernen weiterhin, wie man Redis (eine Open-Source-Pub-Sub-NoSQL-Datenbank) verwendet.

Redis ist unser „Bindegewebe“, das Events veröffentlicht, mit denen sich unser Zero verbindet und dann auf den Scroll pHAT angezeigt wird. So müssen Sie nicht erst noch umständlich einen Webserver aufsetzen. Keine Sorge, wenn Sie keinen Scroll pHAT haben – wir zeigen Ihnen eine alternative Version des Codes, die den Text stattdessen auf Ihrem SSH-Terminal darstellt.

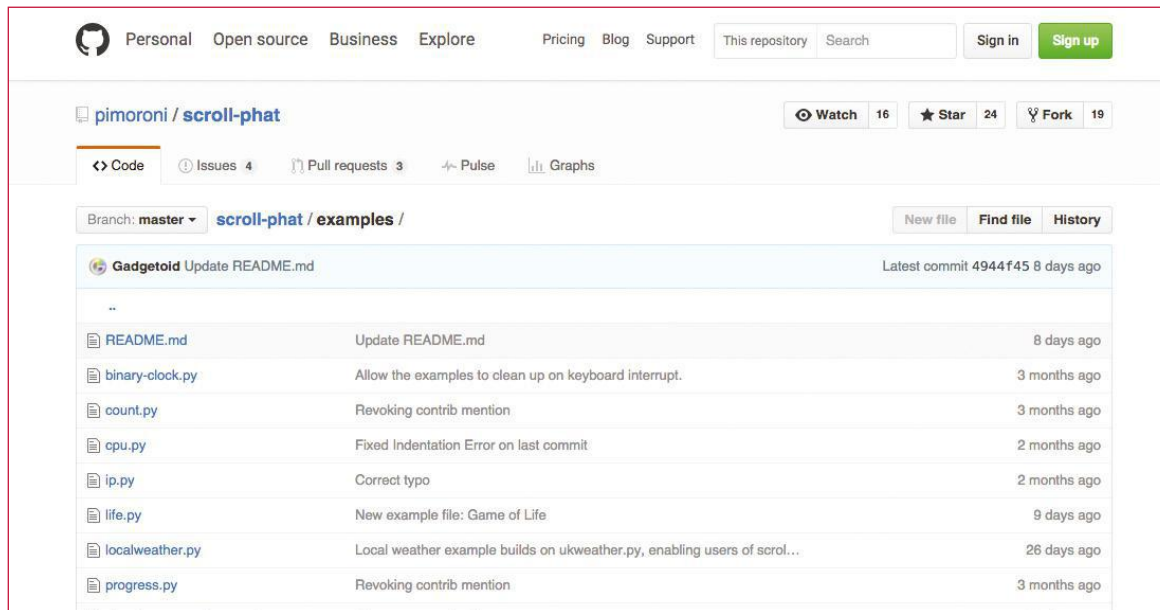
01 Betrachten Sie zuerst den Scroll pHAT

Der Scroll pHAT ist einer von mehreren pHATs (Hardware oben befestigt), die bei der Einführung der Raspberry Pi Zero erschienen. Pimoronis Gerät kostet über einen Online-Shop 10 Euro und ist auch in Bausatzform erhältlich. Das Preis-Leistungs-Verhältnis ist ausgezeichnet, wenn man bedenkt, dass der Preis für eine leere Platine um die 8 Euro liegen kann. Für Ihr Geld bekommen Sie eine schöne Leiterplatte in schwarzer Ausführung mit elf Reihen von fünf vormontierten LEDs. Das Herz des pHAT ist ein ISSI-IS31FL3730-LED-Matrix-Treiber. Der LED-Treiber versteht das I2C-Protokoll und kann sich mit fast jedem Mikrocontroller (z. B. den Arduino) verbinden.



Was Sie brauchen

- Github Repository (<http://github.com/alexellis/rpi-display>)
- Zwei Raspberry Pis
- Pimoroni Scrollphat (optional)
- Lötkolben, -material



Der Pi Zero kommt zurück!

Die ersten Chargen des Pi Zero waren sehr schnell ausverkauft. Die Raspberry-Pi-Stiftung gab bekannt, dass eine Bestellung für weitere 150.000 Pi Zeroes aufgegeben wurde. Die perfekte Zeit also, sich alles Zubehör und alle Erweiterungskarten zuzulegen, damit Sie für Ihren neuen Pi Zero gerüstet sind. Ausgestattet mit einer USB-Batterie-Bank, WLAN-Dongle und der Scroll-pHAT-Karte können Sie Ihren Fernbedienungssensor fast überallhin mitnehmen.

02 40-Pin-Header löten

Pimoronis Gerät ist fast einsatzbereit, aber wir müssen noch ein wenig löten. Wir werden die Header-Pins löten und uns entscheiden, ob wir die vertikale oder die 90-Grad-Version (enthalten) nutzen. Eine weitere Option ist, ein einziges Set Pins zu verwenden und den Scroll pHAT direkt auf die Oberseite des Pi Zero zu löten. Egal welche Option Sie wählen, stellen Sie sicher, dass Sie sich in einem gut belüfteten Raum mit guter Beleuchtung befinden. Hilfreich sind unserer Ansicht nach auch ein Flussmittel und ein dünnes Lötmittel.

03 Die Hardware verbinden

Stellen Sie den Zero und den Scroll pHAT sorgfältig auf und schieben Sie dann beides sorgfältig zusammen. Dann schalten Sie den Strom ein und booten Sie in Raspbian. Von hier aus sollten Sie ein Terminal-Fenster oder eine ssh-Verbindung öffnen und Folgendes eingeben, um die Installation aufzurufen:

```
curl -sSL get.pimoroni.com/scrollphat | bash
```

Profi-Tipp: Führen Sie ein Bash-Skript direkt über das Internet aus – das machen viele! Sie können das Skript immer in einem Webbrowser öffnen, um den Setup-Prozess zu verfolgen.

04 Führen Sie Ihr erstes Beispielskript aus

Jetzt, da Sie die Bibliothek installiert haben, klonen Sie das Git-Repository und schauen in den Beispielordner.

```
git clone https://github.com/pimoroni/scroll-phat
cd scroll-phat/examples/
```

Der Autor hat mehrere Codebeispiele zusammengetragen: Probieren Sie mal das progress.py-Beispiel mit: **sudo python progress.py**. Sie sollten eine Animation sehen, die an das klassische Snake-Game erinnert. Es gibt noch viele andere, die Sie einmal ausprobieren sollten.

05 LED mit der Bibliothek aufleuchten lassen

Nach dem Import des Scroll-pHAT-Moduls können wir Animationen mit einer Kombination aus **set_pixel()**, **update()** und **time.sleep()** erstellen. So blinkt die erste LED:

```
import scrollphat
scrollphat.set_brightness(2)
while(True):
    scrollphat.set_pixel(0, 0, True)
    scrollphat.update()
    time.sleep(0.5)

    scrollphat.set_pixel(0, 0, False)
    scrollphat.update()
    time.sleep(0.5)
```

Verwenden Sie **scrollphat.set_brightness()**, um eine passende Helligkeit für die Umgebungsbeleuchtung auszuwählen.

06 Zeigen Sie eine kurze Nachricht an!

Um eine statische Nachricht auszugeben, verwenden Sie **write_string(text)**, gefolgt von **update()**. Dies lädt eine Standard-Schriftart bei der jeder Buchstabe etwa 5 x 3 Pixel einnimmt, sodass Sie Platz für etwa drei Buchstaben haben.

```
scrollphat.write_string("F00")
scrollphat.update()
```

Das **count.py**-Beispiel verwendet **write_string**, um bis zu einer gegebenen Zahl zu zählen, wie eine Stoppuhr. Aber was ist, wenn Ihre Zeichenfolge nicht in drei Zeichen passt?

07 Längere Texte anzeigen lassen

Bei längeren Nachrichten können wir die Funktion **scroll()** aufrufen. Jedes Scrollen() verschiebt den Text nach links um ein Pixel. Um den Buchstaben „f“ aus dem Bildschirm zu verschieben, scrollen Sie() dreimal – oder in einer **while**-Schleife:

```
scrollphat.write_string("foo")
while(True):
    scrollphat.scroll()
    scrollphat.update()
```

08 Schirm säubern

Den Schirm aufzuräumen kann auf zwei Arten geschehen. Entweder durch die Methode **set_pixel(x,y, False)**,

Zur Scroll-pHAT-Bibliothek beitragen

Die Scroll-pHAT-Python-Bibliothek ist Open-Source und verfügbar unter der GP-Lizenz auf GitHub, was bedeutet, dass Sie das Repository abspalten können. Abspalten (Forking) bedeutet, dass Sie Änderungen auf eigene Faust vornehmen können und dann eine Pull-Request (PR) senden dürfen, die den Pi-moronis-Entwickler Phil darauf hinweist, einen Blick auf Ihre Arbeit zu werfen und diese (evtl.) mit dem Hauptprojekt zu verschmelzen. Die erste PR erhielt der Autor bereits vor 13 Jahren!

um einzelne Pixel auszuschalten, oder danach mittels `scrollphat.update()`. Hier eines von vielen Beispielen:

```
scrollphat.write_string("foo")
try:
    while(True):
        scrollphat.scroll()
        scrollphat.update()
except KeyboardInterrupt:
    scrollphat.clear()
    scrollphat.update()
```

Ein gutes Beispiel ist auch das `uptime.py` file.

09 Einen Sensor verbinden

Wir werden nun einen Sensor an unsere Sensor-RPi anschließen, dies kann jeder Sensor mit einem HIGH/LOW-GPIO-Ausgang oder auch etwas Komplexeres sein, solange es eine Python-Bibliothek dafür gibt. Das mit Fritzing erzeugte Diagramm zeigt einen PIR-passiven Infrarotempfänger – diese können für ein paar Euros gekauft werden und reagieren auf Hitzesignaturen, wie sie von Personen oder Tieren produziert werden. Verbinden Sie 5v an 5v, GND an GND und dann das Signal an Pin 17 (zum Beispiel).

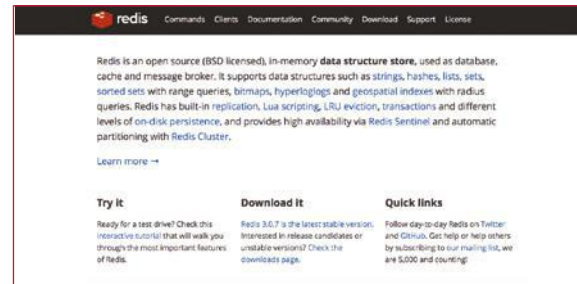
10 Python-Bibliothek installieren

Bei beiden Pis installieren Sie die Redis-Python-Bibliothek, gefolgt von redis selbst. Redis ist viel mehr als ein Key/Value-Pair-Store – es versteht sich mit Pub/Sub, Ranking sowie effizienten Set-basierten Operationen und ist wirklich schnell auf dem Pi. Der abtastende Pi wechselt mit der redis-Instanz über Ihr WLAN-Netzwerk auf den TCP-Port 6739. In unserem Beispiel verwenden wir pub/sub und ein key/value-Paar.

```
sudo pip install redis
sudo apt-get install redis-server
```

Auf dem Display konfigurieren Sie den Dienst fürs Booten:

```
sudo systemctl enable redis
sudo systemctl start redis
```



11 Redis-Befehle

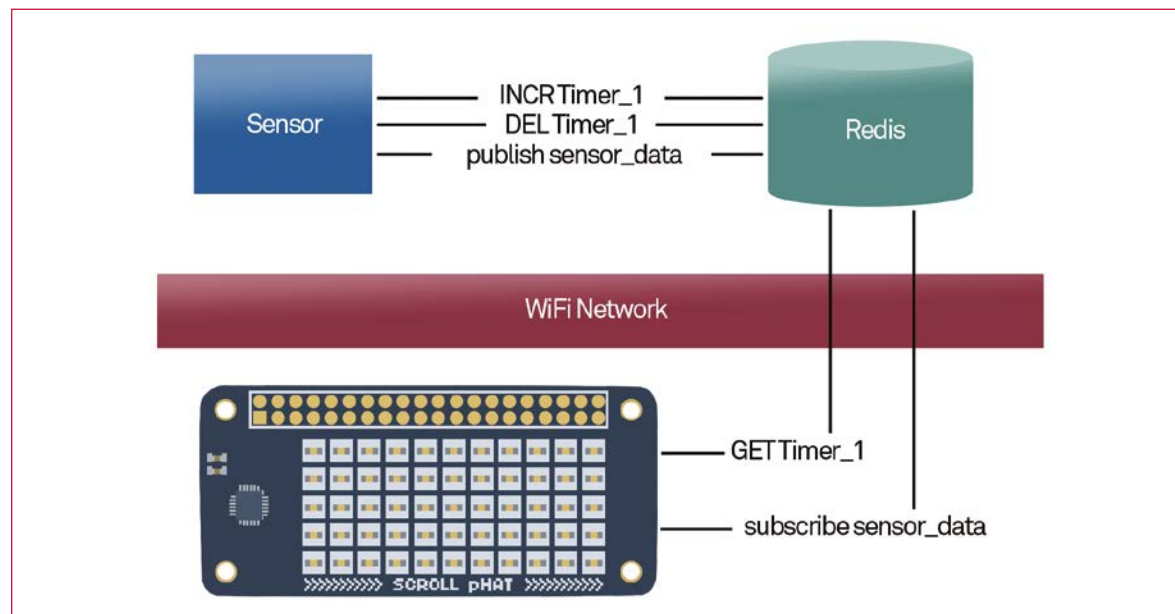
Die `sensor.py`-Software erhöht einen Zähler für jede Minute der Inaktivität, aber wenn der PIR die Bewegung feststellt, setzt er den Wert auf null zurück. Immer wenn sich der Zähler ändert, wird eine Meldung veröffentlicht, die die Anzeigeeinheit verwenden kann, um zu wissen, wann sie ihre Anzeige aktualisieren soll. Sie können auch das enthaltene `redis-cli`-Tool verwenden, um diese Befehle auszuprobieren.

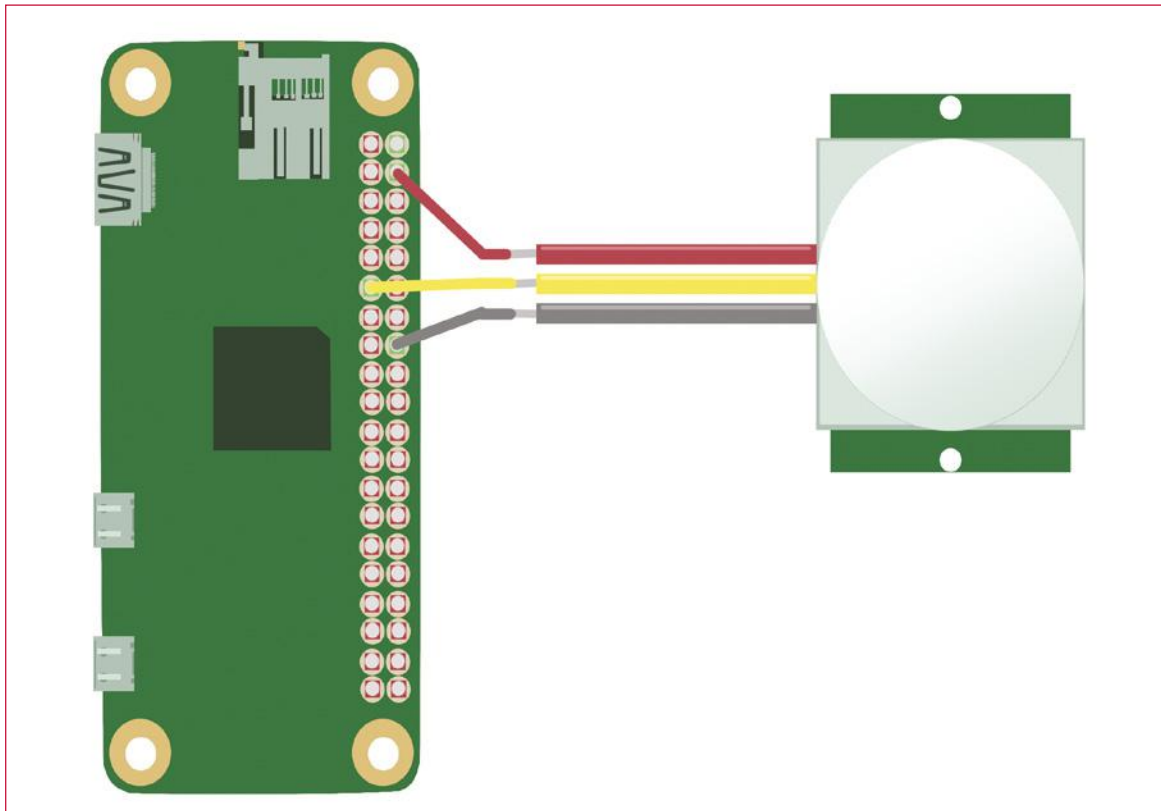
- * `INCR <key> <amt>` increments a key by one, or the amount given
- * `DEL <key>` removes and zeros a key
- * `PUBLISH <channel>` sends a message onto a channel
- * `SUBSCRIBE <channel>` subscribes to a channel for new messages

12 Redis-Bibliothek-Grundlagen

Um einen Client zu erstellen, importieren Sie die `redis`-Bibliothek und rufen Sie dann `redis.StrictRedis` auf. Übergeben Sie die IP-Adresse des Servers über die Variable `ip_address`. Methoden wie `incr` können dann auf dem Client benutzt werden.

```
>>> import redis
>>> client = redis.StrictRedis(ip_address)
>>> client.incr('counter', 10)
>>> client.incr('counter', 1)
>>> print(str(client.get('counter')))
>>> print("Counter: {}".format(client.get('counter')))
Counter: 11
```





Wir haben uns entschlossen, `redis_controller.py` zu erstellen. Sie finden diese sowohl in `sender.py` als auch in `display.py`.

13 Redis pub/sub

Aufgrund der Art und Weise, wie redis funktioniert, ist ein separater Client für das Anlegen eines Kanals erforderlich.

14 Demonstration vorbereiten

Starten Sie die Sensor-Software!

- Prüfen Sie mit `systemctl start redis`, ob der redis-Server gestartet ist.
- Klonen Sie Github-Repository und CD in den Ordner `rpi-display`
- Tippen Sie `sudo python sender.py` ein.

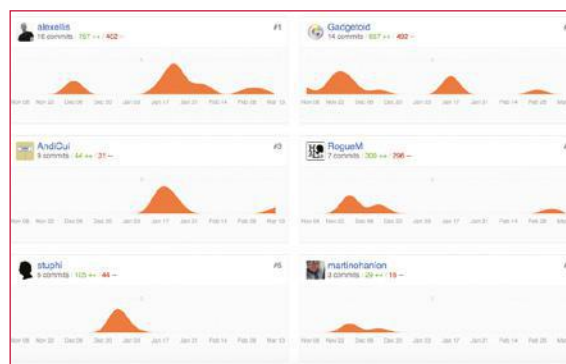
Starten Sie die Display-Software:

- Klonen Sie Github-Repository und CD in den Ordner `rpi-display`
- Tippen Sie `sudo python sender.py` ein.

Wenn Sie testen oder keinen Scroll-pHAT haben, dann bearbeiten Sie `display.py` und ändern Sie „from `pixel_scrollphat_display`“ zu „from `terminal_display`“.

15 Testen Sie die Demonstration

Jetzt, da Sie beide Teile der Demonstration gestartet haben, richten Sie den PIR von Ihnen weg und warten Sie. In jeder Minute, die vergeht, sollten Sie Punkte sehen, die sich auf dem Bildschirm des Scroll-pHAT genau wie bei einem Tetris-Spiel ansammeln. Wenn Sie mindestens einen Punkt auf dem Bildschirm haben, stehen Sie auf und stellen sich dem PIR in den Weg, um den Bildschirm zu löschen.



16 Daten von anderen Sensoren anzeigen lassen

Das `Sender.py`-Beispiel kann auch angepasst werden, um Daten von anderen Sensoren (z. B. die Umgebungstemperatur der Pi-CPU) zu sammeln. Anstatt einmal `INCR` aufzurufen, können Sie `DEL` und dann `INCR` einmal für jedes Grad anrufen, bevor Sie das Ergebnis anzeigen. Wenn Sie feststellen, dass die Punktanzeige unsinnig ist oder dass Ihre Ausgabe mehr als 5 x 11 Pixel abdecken muss, können Sie `numeric_scrollphat_display.py` als Ihre Anzeige verwenden, die eine Zahl zwischen 0 und 999, ohne Scrollen zu müssen, anzeigen kann.

17 Zusammenfassung

Wir haben gelernt, die Scroll-pHAT-Python-Bibliothek zu verwenden, um einzelne Pixelanimationen sowie statische Textnachrichten anzuzeigen. Wir nutzten Redis als Server und einen Client für die Vernetzung. Hier noch einige Ideen: Nutzen Sie PIR als stille Türklingel, die aufleuchtet, wenn sie Bewegung erkennt. Betreiben Sie einen Pi-Webserver und lassen Sie die CPU-Last anzeigen. Verbinden Sie einen magnetischen Sensor mit einem stationären Fahrradtrainer und lassen Sie die Radgeschwindigkeit anzeigen.

Tricks

64 Verbindungs- Geheimnisse

- Wie funktionieren die GPIO-Pins?
- Hardware sicher anschließen
- Lämpchen und Schalter
- Logik-Schaltzeichen

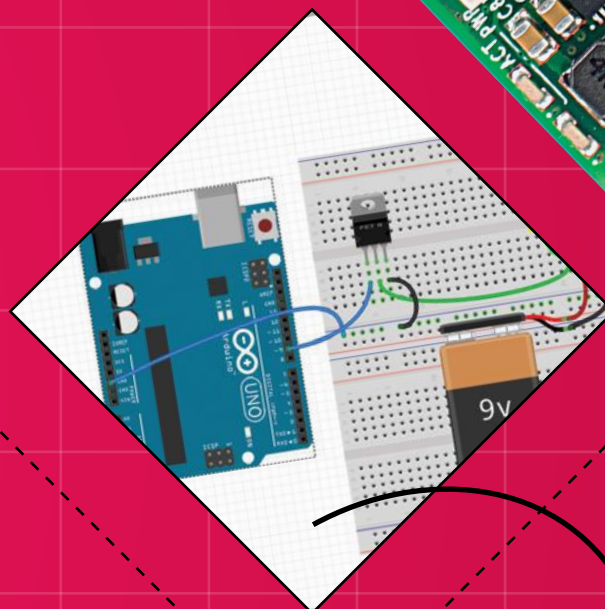
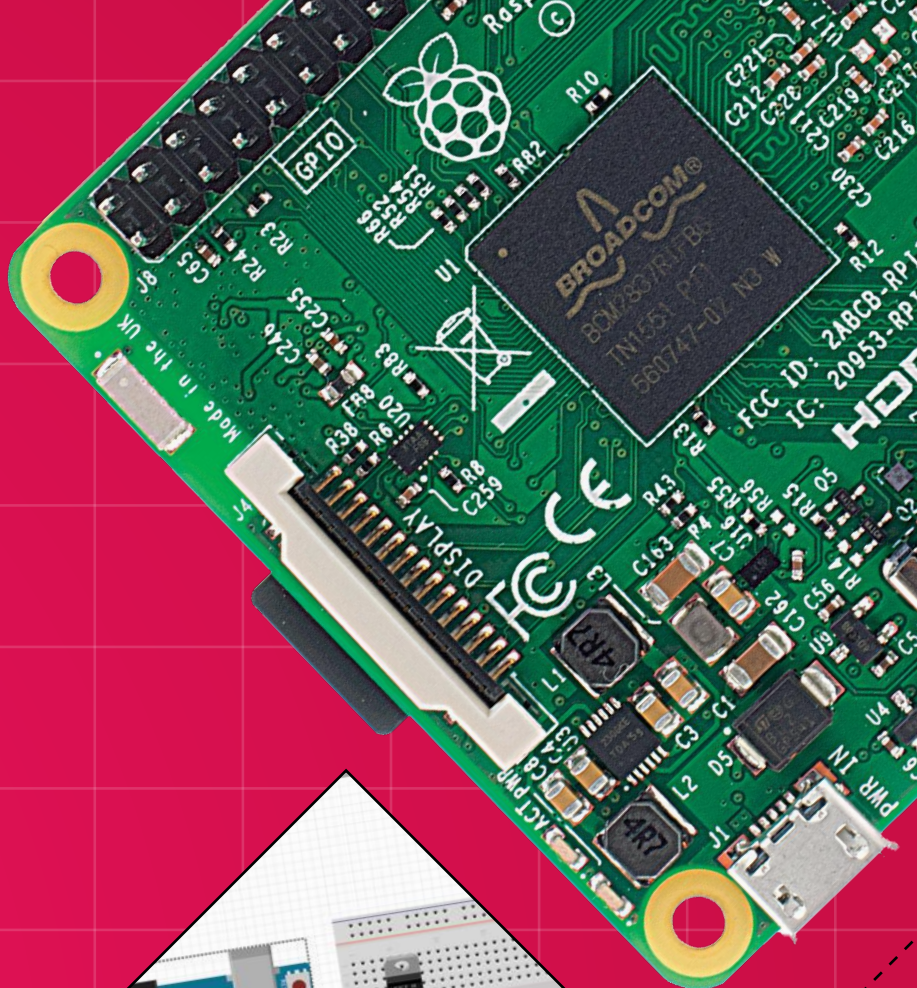
Sie können speziell für den Raspberry Pi entwickelte Zusatzplatinen verwenden, Schaltpläne von anderen nachbauen oder sogar Ihre eigenen Schaltkreise konstruieren!

64



Der Raspberry Pi bietet Ihnen die einfachste Möglichkeit, ein voll funktionsfähiges Minecraft-Spiel auf Ihr großes Fernsehgerät zu bringen.

- 74 Wie Sie ein interaktives Wearable bauen
- 78 Der Pi als Minecraft-Konsole
- 84 Minesweeper-Spiel mit Minecraft
- 88 Licht mit Ihrem Pi steuern
- 92 Streamen Sie Internet-TV über den Pi
- 94 Web-Traffic mit Pi-Tor-Router anonymisieren
- 98 Mit Fritzing eigene Schaltpläne erstellen
- 102 Pi 2 als Desktop-PC
- 106 Dank Pi: Sound in jedem Raum



88

Licht an per
smartphone

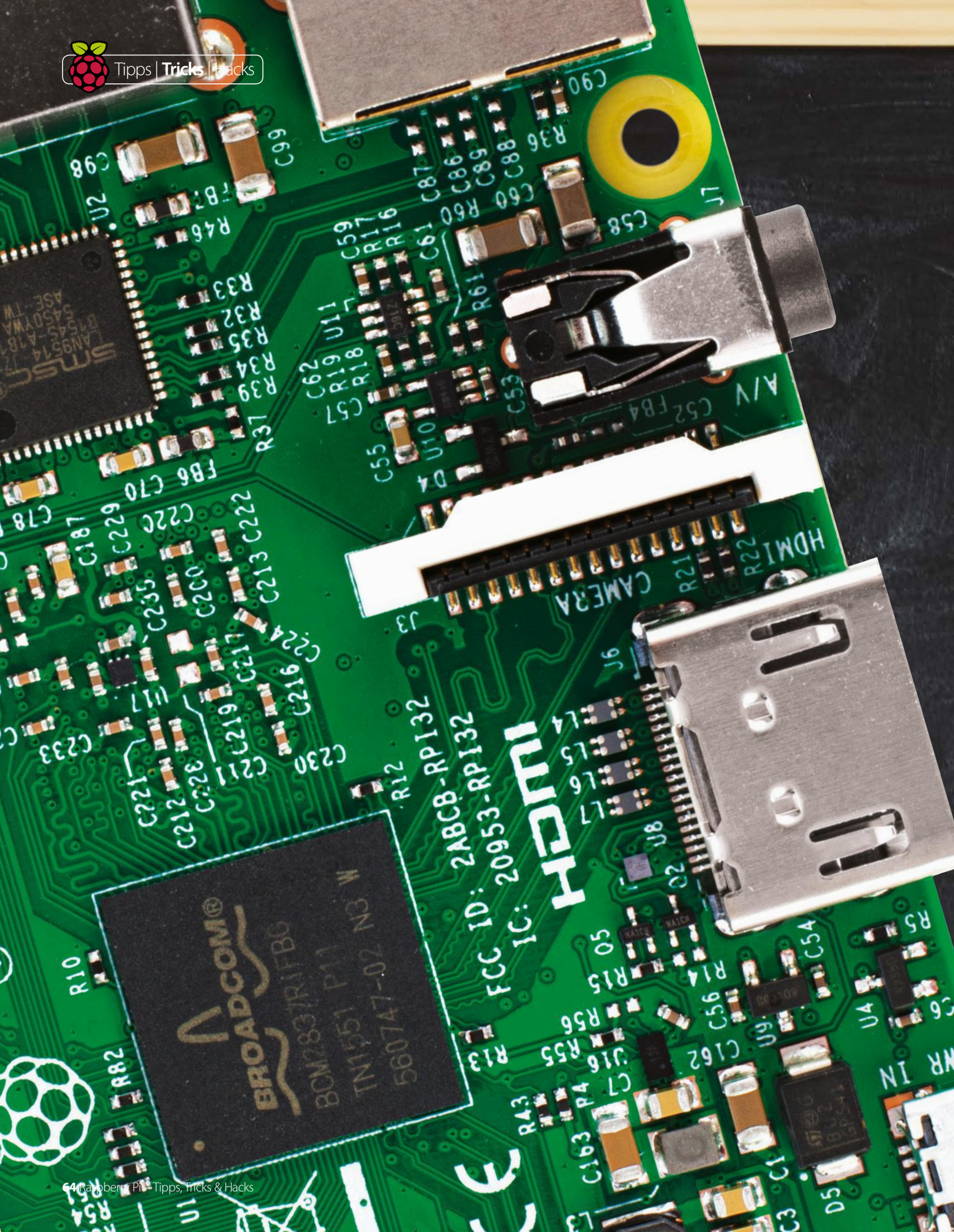
98

schaltpläne
zeichnen

78

spielkonsole
bauen







VERBINDUNGS- GEHEIMNISSE

Erfahren Sie, wie Sie einfache elektronische Schaltkreise entwerfen, um Ihren Pi an Geräte anzuschließen.

Damit Sie mit einem Single-Board-Computer wie dem Raspberry Pi reale Geräte steuern können, benötigen Sie zwei vollkommen verschiedene Fertigkeiten. Erstens müssen Sie in der Lage sein, Code zu schreiben, und zweitens müssen Sie externe Geräte anschließen können.

In diesem Artikel gehen wir auf das zweite Thema ein und erarbeiten uns insbesondere, wie man über die Verwendung von handelsüblichen Schnittstellen (z. B. HATs) hinausgeht und sogar Schaltkreise baut, die andere entworfen haben. Bald werden Sie dazu fähig sein, Schalter und LEDs miteinander zu verbinden und vieles mehr. Wir zeigen Ihnen hier Schaltpläne – falls Sie damit noch nicht vertraut sind, sehen Sie sich unseren Guide auf Seite 98 an, in dem Sie mehr darüber erfahren, wie Schaltpläne funktionieren, wie man sie zeichnet und benutzt.

Unser Hauptaugenmerk liegt auf dem Raspberry Pi, aber die meisten Schritte können auch auf anderen Mini-Computern wie dem Arduino angewendet werden, die auch für Steuerungsanwendungen beliebt sind.

Die GPIO-Hardware des Pi verstehen

Entdecken Sie die GPIO-Hardware des Pi – ihr Tor zur realen Welt.

Auch wenn Sie noch niemals externe Geräte an Ihren Pi angeschlossen haben, werden Sie bestimmt schon die doppelte Reihe von 40 Pins bemerkt haben. Dabei spricht man vom GPIO-Header, der auch als Allzweck-Input/Output-Connector bekannt ist. Er ist dafür da, reale Geräte an den Pi anzuschließen. Bevor Sie aber darüber nachdenken, Schaltkreise für die Anbindung an den Pi zu entwerfen, müssen Sie zuvor die Grundlagen der GPIO-Hardware verstehen.

Pins für Strom und Erdung

Obwohl als GPIO-Header bezeichnet, verbinden sich nicht alle Pins mit der GPIO-Hardware. Einige der anderen Pins bieten Strom- und Erdungs-Verbindungen, die auch von der Hardware verwendet werden, die mit dem Header verbunden ist.

Der GPIO-Header des Pi verfügt über acht Ground-Pins (GND), die Sie mithilfe der Dokumentation identifizieren können. „Ground“ (Erdung) ist gleichbedeutend mit der negativen Seite der Stromversorgung und wird oft als 0V (d.h. Null Volt) bezeichnet.

Der GPIO-Header hat auch vier Stromversorgungspins: zwei, die + 3.3 V, und zwei, die + 5 V liefern. Mit den Erdungs- und Stromversorgung-Pins können Sie Strom aus dem Pi für Ihre externen Schnittstellen-Schaltkreise beziehen, sodass Sie keine separate Stromversorgung benötigen.

GPIO-Pins

Mit zwei Ausnahmen sind die übrigen Pins auf dem GPIO-Header GPIO-Stifte, obwohl einige auch sekundäre Funktionen haben, die wir hier nicht weiter behandeln. Wie die Bezeichnung „Allzweck-Eingabe/Ausgabe“ nahelegt, können diese Pins in der Software so konfiguriert werden, dass Sie entweder als Ein- oder Ausgänge fungieren. Wenn sie als Eingänge programmiert werden, könnten diese Pins beispielsweise an einen Schalter angeschlossen werden und die Software könnte erkennen, ob der Schalter auf „Ein“ oder „Aus“ steht. Alternativ können diese Pins, wenn sie als Ausgänge programmiert werden, an eine LED angeschlossen werden, und die Software könnte sie ein- oder ausschalten.

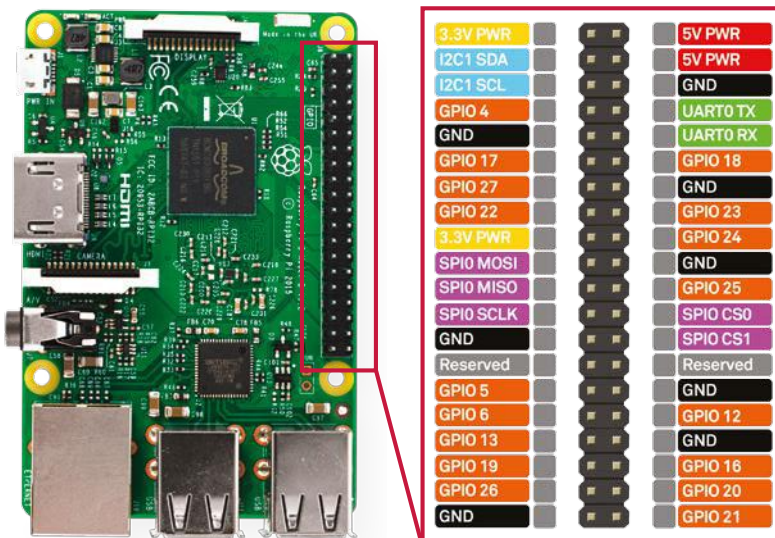
Maximale Werte

Die Raspberry-Pi-GPIOs arbeiten mit einer Versorgung von 3,3 V, sodass Sie keine höhere Spannung an einem

der Pins anbringen sollten – das könnte den Pi zerstören. Allerdings besteht die Möglichkeit, Verbindungen zu Geräten herzustellen, die höhere Spannungen benötigen, wie wir später im Abschnitt „Limits sicher überschreiten“ (auf der nächsten Seite) sehen werden. In der Tat gibt es sogar Anschlussmöglichkeiten an netzbetriebene Geräte. Die maximale Spannung ist nicht der einzige Weg, die GPIO auszureizen. Sie sollten auch den maximalen Strom von 16 mA (und insgesamt 50 mA für alle GPIO-Pins) einhalten. In der Praxis bedeutet dies, dass Sie mit LEDs keine Probleme kriegen werden, wohingegen Sie beim Betrieb eines Motors viel vorsichtiger sein sollten.

GPIO-Pin-Nummerierung

Es gibt zwei Nummerierungsschemata für GPIO-Pins. Zunächst gibt es die physische Nummerierung. Diese spiegelt die Position jedes Pin auf dem Header wieder, sie geht von 1 und 2 an einem Ende zu 39 und 40 am anderen. Für die tatsächlichen GPIO-Pins (im Gegensatz zu Netzteilen) gibt es GPIO-Nummern. Sie können in der Software wählen, welches Schema Sie verwenden möchten.





Limit sicher überschreiten

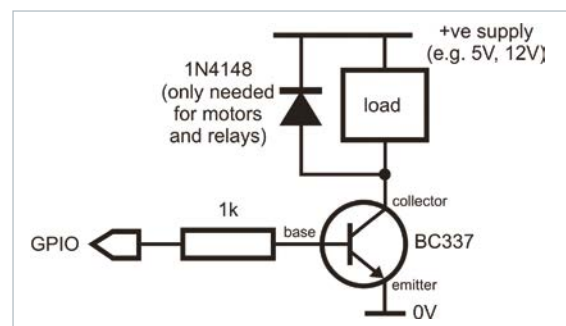
Verbinden Sie Geräte, die die maximale Spannung oder Stromstärke des GPIO überschreiten.

Das Anschließen eines Ausgabegeräts wie einer LED an einen GPIO-Pin erfordert, dass das Gerät nicht mehr als 3,3 V benötigt und es nicht mehr als einen GPIO-Pin mit maximal 16mA-Strom zieht. Geräte wie Elektromotoren und blaue oder weiße LEDs, die eine höhere Spannungsversorgung erfordern und/oder mehr Strom ziehen, benötigen eine spezielle Behandlung. Die Lösung besteht darin, einen Transistor zu verwenden, den man sich als elektronischen Schalter vorstellen kann. Ein kleiner Stromfluss von einem GPIO-Pin schaltet den Transistor ein oder aus, wodurch er einen separaten Schaltkreis mit einer höheren Spannung ein- oder ausschaltet und oft mehr Strom als ein GPIO-Pin liefern kann. Dieser Sekundärkreis kann die 5 V auf dem GPIO-Header als Versorgung verwenden, aber wenn eine höhere Spannung erforderlich oder der Strom höher ist als der, den der GPIO-Header liefern kann, wird eine externe Stromversorgung benötigt.

Der Schaltplan zeigt die allgemeine Konfiguration. Wenn 3,3 V an den Ausgang des Transistors über den Widerstand angelegt wird, schaltet er sich ein. Man kann sich das so vorstellen, als wäre der Emitter des Transistors direkt mit seinem Kollektor verbunden. Dies bedeutet, dass Strom von der Hochspannungsversorgung durch den Verbraucher (Motor, blaue LED usw.) auf 0V fließen kann und somit dreht sich der Motor oder die LED leuchtet. Die Wahl der Transistor-Art und der Wert des Widerstands ist knifflig, zumal

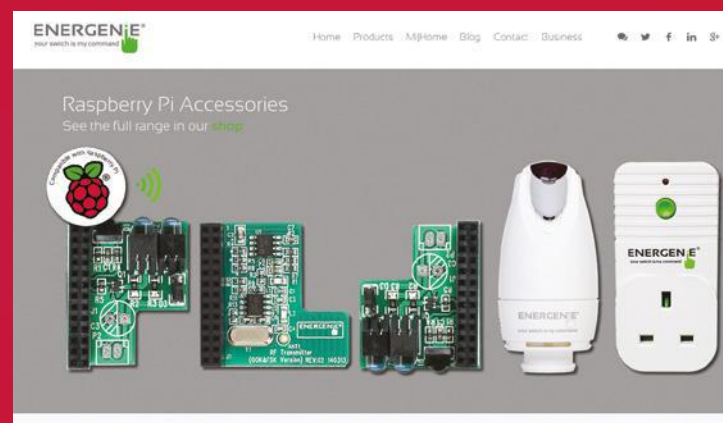
es so viele verschiedenen Transistoren gibt. Daher können wir lediglich ein paar Tipps geben. Zuerst muss in der dargestellten Schaltung der Transistor vom NPN-Bipolartransistor sein. Transistoren dieser Art werden auch durch ihre Verstärkung und den maximalen Strom und die Spannung, die Sie auf dem Emitter-Kollektor-Schaltkreis verwenden können, definiert. Und hier wird es kompliziert: Belassen wir es bei folgendem: Für Spannungen bis zu 24 V und Strom bis zu 250 mA (konservative Grenzen) wäre ein BC337 ideal. Bei dieser Art von Transistor und für diesen Strom wäre ein 1k-Widerstand geeignet.

Wenn es sich bei dem Verbraucher um einen Motor oder ein Relais handelt, ist es wichtig, eine Diode parallel zur Last (Kathode zur Versorgung) zu verdrahten, um den Transistor schädigende Rückströme zu unterdrücken,



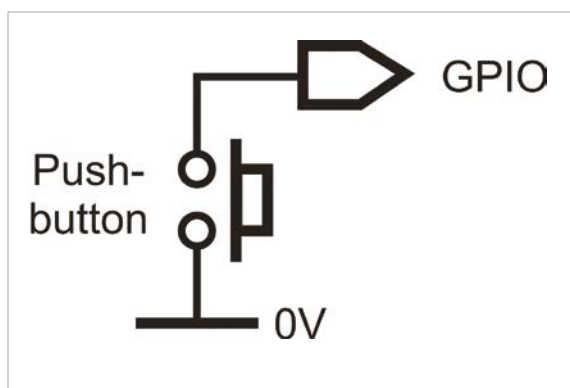
Schnittstelle zum Netzgerät

Das Entwerfen von Schaltkreisen zur Steuerung von Netzgeräten ist nicht schwierig, aber wenn etwas schiefgeht, könnten Sie Ihren Pi zerstören oder sich selbst einen Schlag versetzen. Es gibt HATs, die Sie zu diesem Zweck kaufen können, aber wir empfehlen diese nicht, da sich die Netzklemmen in der Nähe der Niederspannungsschaltkreise befinden. Wenn sich ein Draht lockert, könnte der Pi Feuer fangen, eine Komponente kaputtgehen, ihnen ein Schrapnell ins Gesicht fliegen oder hohe Spannungen in die Nähe Ihrer Finger bringen. Aus diesem Grund empfehlen wir Ihnen die Verwendung von handelsüblichen Schnittstellen, bei denen die einzige Anbindung an die netzbetriebenen Geräte über eine landestypische Steckdose erfolgt. Energenie (energenie4u.co.uk) bietet ferngesteuerte Steckdosen mit 13 A, die mit einer Fernbedienung genutzt werden können. Alternativ können sie mit einem Funksendermodul für den Pi verwendet werden, das ebenfalls bei Energenie erhältlich ist.



Wie man Schalter und LED an den Pi anschließt

Die Anbindung eines Schalters oder einer LED an den GPIO-Header könnte nicht einfacher sein.

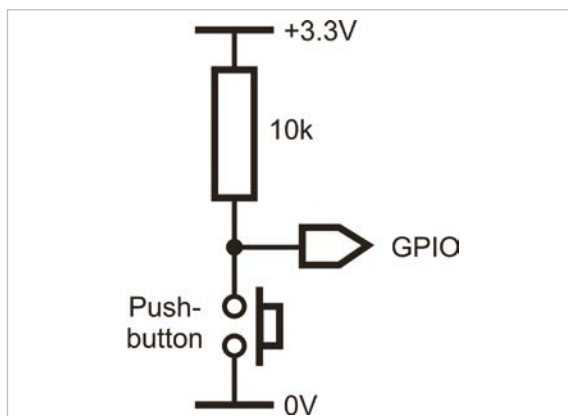


01 Den Schalter verdrahten

Als erstes müssen Sie, wenn Sie einen Schalter an den Pi anschließen, einen der beiden Terminals des Schalters mit einem GPIO-Pin (der als Eingang in der Software konfiguriert wird) anschließen und den anderen Terminal mit 0V (GND) verbinden. Danach wird der GPIO-Pin an 0V (GND) angeschlossen, ein Zustand, den die Software als Logik 0 wertet, wann immer der Schalter geschlossen ist, d. h. bei einem Tastendruck oder im „Ein“-Zustand mit einem mechanischen Kippschalter.

02 Pull-up-Widerstand hinzufügen

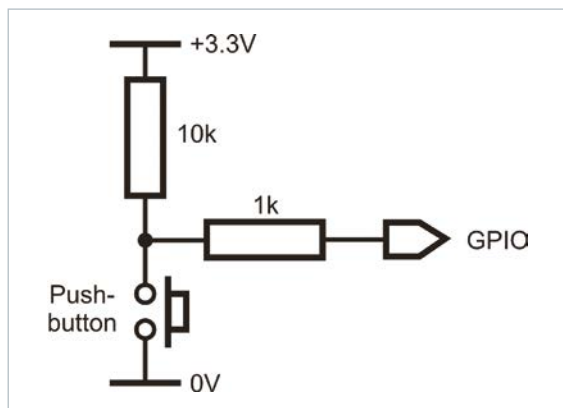
Obwohl ein GPIO-Pin an einen Schalter angeschlossen ist und 0 V auf Logik 0 steht, wenn der Schalter geschlossen ist, wird er „floating“, wenn er geöffnet ist. Mit anderen Worten: Es wäre nicht klar, ob er als 0 oder 1 anzusehen wäre. Um das Problem zu lösen, muss er über einen Widerstand, der als Pull-up-Widerstand bezeichnet wird, auf +3,3 V verdrahtet werden.



03 Verwenden Sie eingebaute Pull-ups

Wenn Sie den Schalter auf andere Single-Board-Computer oder auf die GPIO-Pins über einige Logik-Schaltungen verdrahten, ist ein externer Pull-up-Widerstand die einzige Lösung. Wenn Sie jedoch direkt an einen GPIO-Pin verbinden, können Sie als Option einen internen Pull-up-Widerstand in der Pi-Schaltung aktivieren. Der Code-Schnipsel hier zeigt, wie dies mit der RPi.GPIO-Python-Bibliothek geschieht. Die Schaltpläne in den übrigen Schritten übernehmen ein externes Pull-up, aber wenn Sie ein internes Pull-up verwenden, lassen Sie einfach den 10k-Widerstand aus.

```
GPIO.setup (2, GPIO.IN, pull_up_down=GPIO.PUD_UP)
# set GPIO 2 as input with pull-up
```



04 Den Strom limitieren

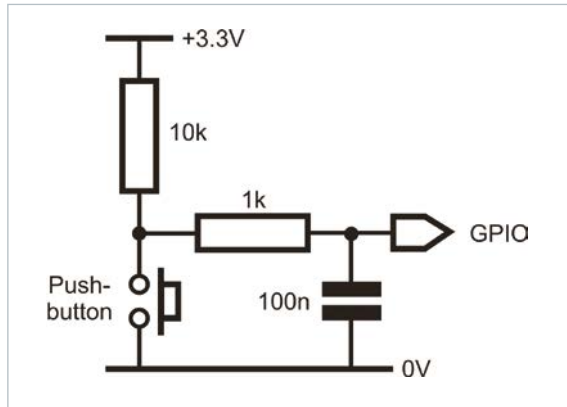
Da GPIO-Pins bidirektional sind, gibt es ein mögliches Problem, wenn ein Pin, der an einen Schalter angeschlossen ist, versehentlich als Ausgang konfiguriert ist und auf Logik 1 eingestellt ist. Damit werden 3,3 V auf den Pin geschickt, der, wenn der Schalter dann geschlossen ist, Direkt mit 0V verbunden würde. Dies würde dazu führen, dass ein starker Strom fließt und den Pi schädigen könnte. Das Einsetzen eines Serienwiderstandes für den Schalter wird dies verhindern, und 1k ist der empfohlene Wert. Der Serien-Widerstand ist auch im nächsten Schritt integral.

05 Den Schalter de-bouncen

Wenn ein Schalter betätigt wird, öffnen und schließen sich die Kontakte oftmals schnell für kurze Zeit. Der sogenannte Bounce könnte Probleme verursachen. Vielleicht soll ein Druckknopf eine LED ein- oder ausschalten. Nun, wenn die LED ausgeschaltet ist und Sie den Druckknopf drücken und loslassen, aber er schaltet geschlossen-offen-geschlossen-offen anstatt ge-

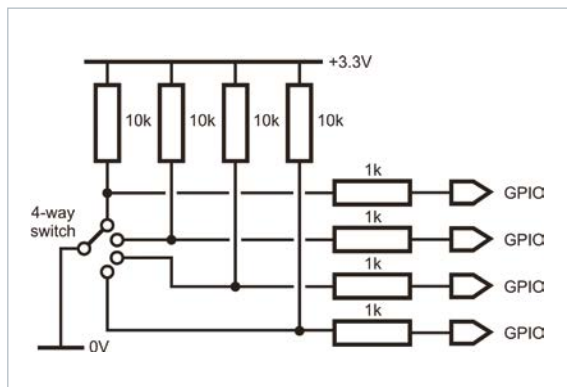


geschlossen-offen, schaltet sich die LED sehr schnell ein und aus, aber es scheint nichts passiert zu sein. Dies wird durch Hinzufügen eines Kondensators (wie gezeigt) behoben – ein Wert von 100 n ist typisch, wenn Sie einen 1k-Strombegrenzungswiderstand verwenden.



06 Mehrwegeschalter benutzen

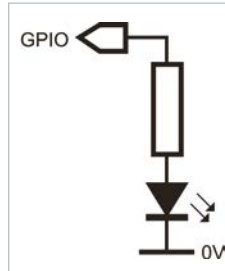
Ein Mehrwege-Drehswitcher wird auf die gleiche Weise wie ein Einwegeschalter gehandhabt, außer dass er mit mehreren GPIO-Stiften verbunden ist und die zuvor beschriebene Schnittstellenschaltung für jeden dieser Pins benötigt. Der Schaltplan zeigt, wie dies bei einem Vier-Wege-Schalter aussehen würde. De-Bounce-Kondensatoren sind nicht enthalten, weil Bounce bei Mehrwege-Schaltern kein Problem darstellt – genauso wie bei einem Druckknopf wegen seiner mechanisch verriegelnden Arbeitsweise.



07 LED-Grundlagen verstehen

Eine LED ist eine Komponente, die Licht erzeugt, wenn Strom durch sie fließt. Es ist ein polarisiertes Gerät, sodass seine Anode immer an die positive Versorgung und ihre Kathode an die negative Versorgung angeschlossen werden muss. Eine Grundeigenschaft einer LED ist die Durchlassspannung, die je nach Farbe und Art üblicherweise zwischen 1,8 V und 3,3 V liegt. Eine LED benötigt diese Spannung, um zu leuchten. Wichtig ist auch der empfohlene Strom, der wiederum die Helligkeit bestimmt. Alle diese Parameter werden im Spezifikationsblatt der LED angegeben.

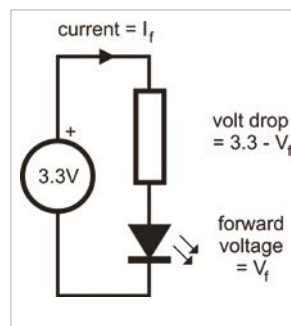
OPTICAL AND ELECTRICAL CHARACTERISTICS (T _{amb} = 25 °C, unless otherwise specified)						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Luminous intensity	I _f = 10 mA	I _v	1	4	-	mcad
Dominant wavelength	I _f = 10 mA	λ _d	612	-	625	nm
Peak wavelength	I _f = 10 mA	λ _p	-	635	-	nm
Angle of half intensity	I _f = 10 mA	θ	-	± 60	-	deg
Forward voltage	I _f = 20 mA	V _f	-	2	3	V
Reverse voltage	I _R = 10 μA	V _R	6	15	-	V
Junction capacitance	V _R = 0 V, f = 1 MHz	C _j	-	50	-	pF



08 Den Strom limitieren

Wenn Sie eine LED von einem GPIO-Pin einschalten, müssen Sie den Pin als Ausgang konfigurieren und dann eine Logik 1 ausgeben. Damit werden 3,3 V auf den Pin gelegt, aber dies wird wahrscheinlich die Vorwärtsspannung der LED übersteigen, wodurch der maximale

Strom überschritten wird und eventuell den Pi beschädigt. Dies wird durch Hinzufügen eines Serien-Widerstandes verhindert, der die überschüssige Spannung abfängt. Der Widerstand muss die Differenz zwischen 3,3 V und der Vorwärtsspannung der LED ausgleichen.



09 Ohmsches Gesetz nutzen

Zuerst müssen Sie den Strom für die LED festlegen. Dieser muss kleiner sein als der empfohlene Strom der LED und geringer als der des GPIO-Pin (maximal 16 mA). Auch der Gesamtstrom, der von allen GPIO-Stiften gezogen wird, muss kleiner als 50

mA sein. 10 mA wird meist genug Licht abgeben, auch wenn der empfohlene Strom größer ist. Das Ohmsche Gesetz wird als $V = I \cdot R$ zusammengefasst. Dieses kann als $R = V / I$ umgeordnet werden, um den Wert des Widerstands R anzugeben, wobei V die Spannung ist, die reduziert werden muss (d. h. 3,3 V minus der Vorwärtsspannung der LED), und I ist der Antriebsstrom. Zum Beispiel erfordert eine Durchlassspannung von 2,0 V und ein Strom von 10 mA (0,01 A) einen Wert von $(3,3 - 2,0) / 0,01 = 130 \text{ Ohm}$.

10 Weiße und blaue LEDs

Einige LEDs – vor allem Blau, Weiß und „reines Grün“ – haben eine höhere Vorwärtsspannung als 3,3 V, sodass sie nicht direkt von einem GPIO-Pin betrieben werden können. Auch wenn der Wert als 3,3 V angegeben ist, wäre der Betrieb von einem GPIO-Pin nicht sicher oder zuverlässig.

Widerstands- und Kondensatorwerte

Oft sind die Werte von Widerständen und Kondensatoren unkritisch und es können typische Werte verwendet werden. Wenn Sie jedoch eine LED anschließen, müssen Sie den Wert des Strombegrenzungswiderstandes herausfinden. Oft werden Sie feststellen, dass man keinen Widerstand zu dem Wert kaufen kann, den Sie berechnet haben, weil Widerstände (und Kondensatoren) nur in bestimmten Vorzugswerten verfügbar sind.

In der gängigen E-12-Reihe sind diese Werte 1,0, 1,2, 1,5, 1,8, 2,2, 2,7, 3,3, 3,9, 4,7, 5,6, 6,8 und 8,2. Allerdings können Ihnen zusätzliche Werte aus anderen Serien begegnen. Diese Werte können mit zehn Potenzen multipliziert werden. Also, im Falle von Widerständen, zusätzlich zu 4,7, zum Beispiel finden Sie 47, 470, 4,7 k, 47 k, 470 k und 4,7 M. Wenn ein Widerstand nicht in einem Wert verfügbar ist, sollten Sie auf Nummer sicher gehen und den nächstgrößeren Strombegrenzungswiderstand wählen.

Schaltkreise erklärt

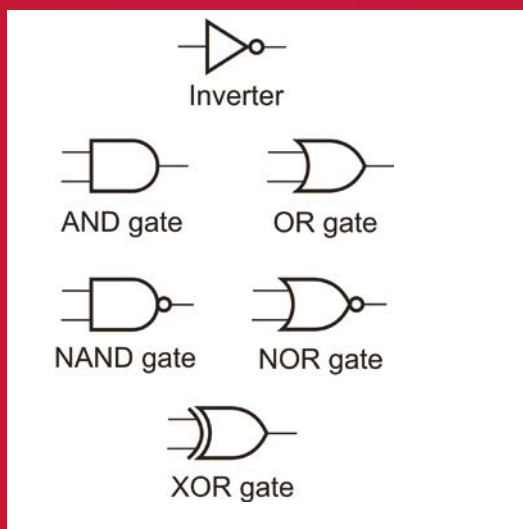
Grundlagen der Logikschaltung für zusätzliche Funktionalität zu Ihrer Schnittstelle.

Da der Prozessor im Raspberry Pi jede erdenkliche Logikoperation ausführen kann, möchte man meinen, dass es keinen Grund für die Verwendung von externen Logikschaltungen gibt.

Das wäre auch richtig, wenn der Pi ausreichend GPIO-Pins für Ihre Anwendung hätte. Wenn Sie sich dem Limit nähern, können Sie die Anzahl an benötigten Pins mittels externer Hardware-Logik reduzieren. Unsere Schritt-für-Schritt-Anleitung zeigt Ihnen, wie Sie das bewerkstelligen können.

Logik-Symbole verstehen

Das Diagramm zeigt die Standardsymbole für die verschiedenen Logikgatter. Jeder von ihnen hat einen oder mehrere Eingänge links und einen einzigen Ausgang rechts. Sie werden feststellen, dass einige Symbole kleine Kreise auf ihren Ausgängen haben. Das sind Gates, die invertierte Ausgänge haben. So ist beispielsweise das Symbol für ein NAND-Gate das gleiche wie das für ein UND-Gate – mit Ausnahme seines invertierten Ausganges. Die meisten anderen Logikvorrichtungen, z. B. ein 2-bis-4-Decoder, werden nur als quadratische oder rechteckige Boxen dargestellt, wieder mit den Eingängen links und den Ausgängen auf der rechten Seite. Weil so viele verschiedene Geräte ansonsten gleich aussehen würden, wenn man Boxen zur Darstellung nimmt, werden diese Symbole in der Regel mit ihrer Komponentennummer (z.B. 74HC138) und die verschiedenen Ein- und Ausgänge sowie deren Funktion mit ihren Pin-Nummern versehen. Logikgeräte verbinden sich mit 0V und einer Stromversorgung. Während diese Verbindungen in der Regel auf komplizierteren Logikgeräten erscheint, werden sie normalerweise nicht auf Gates angezeigt.



Logik-Ebenen

Logikkomponenten arbeiten auf zwei Spannungen, die die Binärwerte von 0 und 1 darstellen, obwohl es für einige Anwendungen besser ist, wenn man sie sich als „An“ und „Aus“ vorstellt. Im Fall der GPIO-Pins des Pi wird 0 durch 0V (GND) dargestellt, während 1 durch 3,3 V dargestellt wird. Bei anderen Singleboard-Computern, die eine 5V-Versorgung verwenden, wird 0 noch durch 0V dargestellt, während 1 durch 5V dargestellt wird. Sie sollten eine Familie von Logikchips wählen (siehe „IC-Logikfamilien“), um die Versorgungsspannung Ihres Computers anzupassen.

Wechselrichter (Inverter)

Die einfachste logische Komponente ist der Wechselrichter, der einen Eingang und einen Ausgang hat (siehe dazu die Diagramme für die Symbole aller Logikgatter). Wie der Name schon sagt, besteht seine Funktion darin, den Wert auf seine

Input	Output
0	1
1	0

Eingabe umzukehren. D.h. wenn der Eingang 0 ist, dann ist der Ausgang 1, wenn der Eingang 1 ist, dann ist der Ausgang 0. Der Vorgang einer Logikkomponente wird oft durch eine Wahrheitstabelle definiert. Hier die Wahrheitstabelle für einen Wechselrichter:

UND- und ODER-Gatter (AND and OR Gate)

Weiter oben nach dem Wechselrichter ist eine Gruppe von Logik-Komponenten als Gates (Gatter) bezeichnet. Gates können beliebig viele Eingaben aber nur eine Ausgabe haben. Betrachten wir das 2-Eingang-UND-Gate, dessen Funktion wie folgt zusammengefasst werden kann: Wenn Eingang 1, 1 ist UND Eingang 2, 1, ist der Ausgang 1. Alle anderen Kombinationen der Eingänge führen zu einer Ausgabe von 0. Mit einer ähnlichen Aussage können wir die Funktion des

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

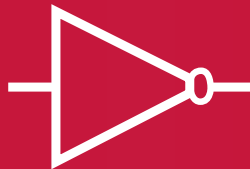
Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1



AND



OR



NOT



NAND



NOR



XNOR



XOR

IC-Logik-Familie

Einer der gebräuchlichsten Typen von Logikgeräten ist die Serie 7400. Diese haben Teilnummern der Form 74 <family> <id>, wobei <family> die Familie 7400 bezeichnet und <id> eine Zahl ist, die die Funktion definiert. Es könnte auch Buchstaben am Anfang und am Ende stehen, aber die kann man ignorieren. SN74LS00N ist zum Beispiel ein Low-Power-Schottky-Familiengerät (LS) und seine Funktion (00) ist ein Quad-2-Eingänge-NAND-Gate (d.h. jeder Chip enthält vier 2-Eingänge-NAND-Gates). Es gibt mehrere 74-Serien und die meisten sind nicht für den Anschluss direkt an die GPIO-Pins geeignet. Manche werden nicht mit 3,3V-Ein- und -Ausgängen funktionieren und viele sind nur in Upgrade-Paketen verfügbar, die für Anfänger nicht geeignet sind. Unsere Empfehlung ist die 74HC-Familie.

ODER-Gates wie folgt zusammenfassen: Wenn Eingang 1, 1 ist ODER Eingang 2, 1 ist, dann ist der Ausgang 1. Alle anderen Kombinationen ergeben einen Ausgang von 0.

NAND-, NOR- und XOR-Gates

NAND-Gate und NOR-Gate klingt zuerst komisch. Wenn wir Ihnen allerdings sagen, dass NAND „not AND“ (nicht UND), und dass NOR „not OR“ (nicht ODER) bedeutet, dann wird es klar. Ein NAND-Gate ist im Grunde ein UND-Gate mit einem Inverter, der mit seinem Ausgang verbunden ist und dessen Wahrheitstabelle die gleiche wie die für das UND-Gatter ist, aber in der Ausgangsspalte mit 0 s auf 1 s geändert wird und umgekehrt. Ähnlich ist ein NOR-Gate ein ODER-Gate mit einem Wechselrichter, der mit seinem Ausgang verbunden ist, also ist seine Wahrheitstabelle die gleiche wie die für das ODER-Gatter, wieder mit 0 s und 1 s, die in der Ausgangsspalte vertauscht wurden. Das übrige Gate ist das XOR-Gate, das für das exklusive ODER-Gate steht und dessen Wahrheitstabelle hier erscheint. Sie werden feststellen, dass es dieselbe wie die Wahrheitstabelle für das ODER-Gatter ist. Sie unterscheiden sich aber dadurch, dass die Ausgabe 0 ist, wenn beide Eingänge 1 sind. Eine andere Möglichkeit, ihre Funktion zu betrachten, ist, dass ihre Ausgabe 1 ist, wenn die Eingänge unterschiedlich sind, Sonst ist die Ausgabe 0.

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

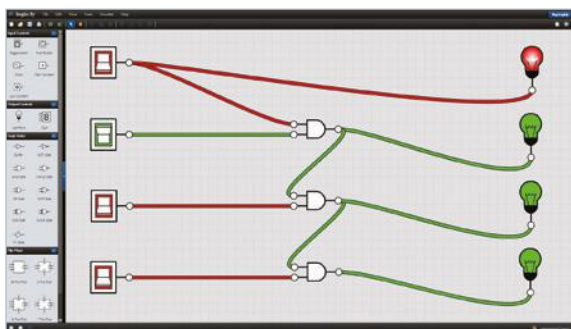
Andere Logik-Funktionen

Der Wechselrichter und die verschiedenen Gate-Arten sind die grundlegendsten Logikkomponenten, aber sie sind nur die Spitze des Eisbergs. Dutzende von anderen Komponenten sind verfügbar, obwohl in Wirklichkeit fast alle ihre Funktionalität durch eine Kombination der grundlegenden logischen Komponenten dupliziert werden könnte. Die meisten dieser Komponenten sind mit rechteckigen Symbolen inklusive Ein- und Ausgängen markiert, also müssen Sie die Wahrheitstabellen, die in ihren Spezifikationsblättern erscheinen, konsultieren, um ihre Funktion zu verstehen. Weil wir das später im Schritt-für-Schritt-Tutorial verwenden werden, sehen wir uns nur ein Beispiel an, das den Namen 2-zu-4-Decoder geht. Die Wahrheitstabelle eines der beiden 2-zu-4-Decoder im 74HC139-Chip erscheint hier (X bedeutet „entweder 0 oder 1“). Zuerst hat das Gerät einen sogenannten Freigabeeingang E. Das Gerät ist nur freigegeben, wenn dieser Eingang auf logisch 0 ist. Wenn das Gerät nicht freigegeben ist, sind alle Ausgänge hoch. Nach dem Aktivieren geht einer der vier Ausgänge je nach Binärzahl an den Eingängen A0 und A1 auf logisch 0. So werden zum Beispiel die Eingänge von 1 und 0 (binär 10 = dezimal 2) zu einer logischen 0 am Ausgang Y2 führen.

Input 1			Output			
E	A1	A0	Y3	Y2	Y1	Y0
1	X	X	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0

Mit Logik-Gates arbeiten, um GPIO-Pins zu sparen

Mit Logikschaltkreisen können Sie mehr Geräte an die begrenzten GPIO-Pins des Pi anschließen.

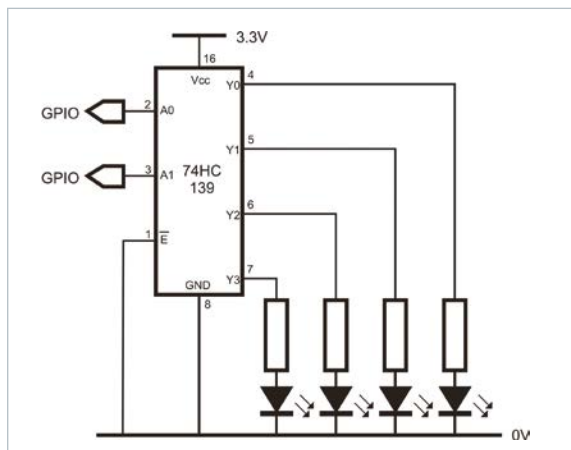


01 Verwenden Sie einen Logiksimulator

Bevor wir uns einige Beispiele für Logikschaltungen anschauen, hier ein Tipp, wie Sie etwas ausprobieren können, ohne dass Sie etwas verdrahten müssen. Wenn Sie nicht sicher sind, welche Komponenten Sie brauchen, kann Ihnen das vor dem Kauf helfen. Das Geheimnis ist, einen Logik-Simulator zu verwenden. Es stehen etliche zur Auswahl: einige laufen lokal unter verschiedenen Betriebssystemen, einige laufen online. Hier testen wir die Schaltung von Schritt 6 unter logic.ly/demo.

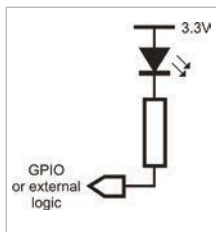
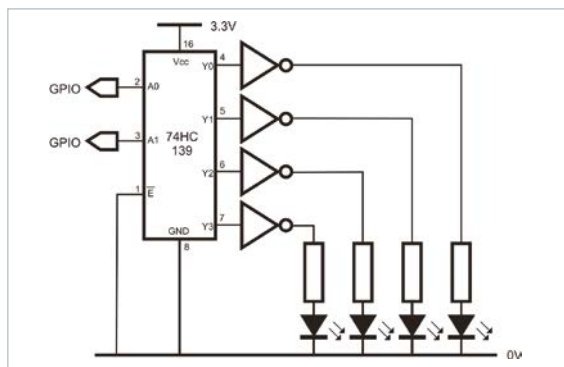
02 Einen 2-zu-4-Decoder verwenden

Das Betreiben von vier LEDs erfordert normalerweise vier GPIO-Pins. Wenn wir aber eine Anwendung haben, wo nur eine von ihnen zu einem beliebigen Zeitpunkt leuchten muss (z. B. bei einer Statusanzeige), reichen zwei GPIO-Pins. Dies wird mit einem 2-zu-4-Decoder erreicht. Wie wir bereits gesehen haben, gibt er je nach Binärwert seiner Eingänge eine logische 0 an einem seiner vier Ausgänge aus. Ein 74HC139 Chip enthält zwei 2-bis-4-Decoder:



03 Ausgänge umkehren

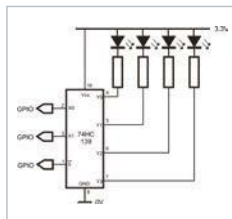
Das vorherige Diagramm zeigt, wie ein 2-zu-4-Decoder an zwei GPIO-Pins angeschlossen werden kann. Es liefert vier Signale. Allerdings sind die Ausgänge „active low“, d. h. wenn LEDs direkt an die Ausgänge geschaltet werden, würden alle leuchten, mit Ausnahme derjenigen, die durch den binären Eingangswert dargestellt werden. Damit nur eine LED leuchtet, benötigt man Wechselrichter:



04 LEDs mit 3,3 V

Als einfachere Alternative zur Umkehrung der Ausgänge von Logikgeräten mit aktiven Low-Ausgängen können LEDs an 3,3 V anstelle von 0V angeschlossen werden. Wir haben schon gesehen, wie man eine LED betreibt, die an 0V mit einem 3,3 V-Signal angeschlossen wird.

Das Diagramm zeigt eine LED, die in dieser Konfiguration verdrahtet ist. Sie leuchtet, wenn eine Logik 0 auf sie angelegt wird, entweder direkt von einem GPIO-Pin oder von einem externen Logikgerät.



05 Balkenanzeige implementieren (1)

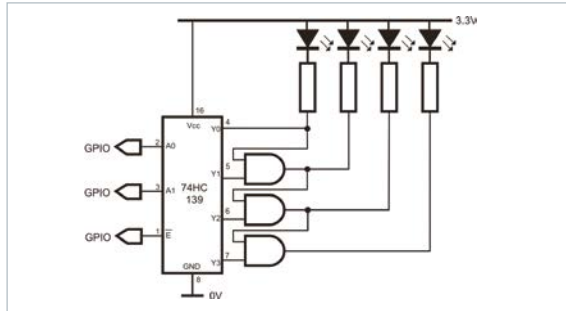
Eine Balkenanzeige, wie man sie manchmal bei Audiogeräten sieht, zeigt die Flexibilität der externen Logik. Wir müssen keine LEDs, eine LED, zwei LEDs und so weiter

bis zu vier LEDs entsprechend einer 2-Bit-Binärwertausgabe an den GPIO-Pins zum leuchten bringen. Ein 74HC139 kommt wieder zum Einsatz. Das Diagramm zeigt die erste Stufe bei der Bereitstellung dieser Funktionalität und Sie werden feststellen, dass, ein dritter GPIO-Pin benötigt wird.



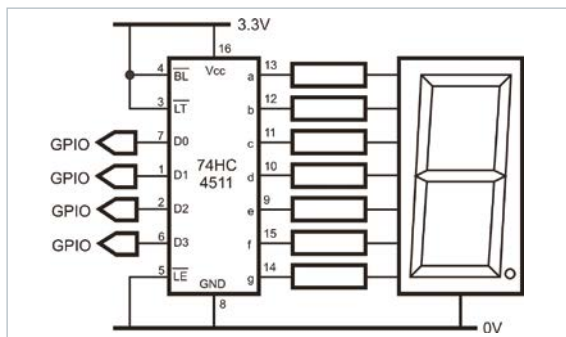
06 Balkenanzeige implementieren (2)

Bis jetzt haben wir eine Schaltung wie in Schritt 3, aber mit der Möglichkeit, alle LEDs auszuschalten. Die Schaltung in diesem Schritt zeigt, wie das Hinzufügen von drei UND-Gates bewirkt, dass binär 00 die LED 1 ansteuert, binär 01 die LEDs 1 und 2, binär 10 die LEDs 1, 2 und 3 und binär 11 die LEDs 1, 2, 3 und 4. Die Wahrheitstabelle für das UND-Gate macht klar, wie diese Schaltung das erreicht.



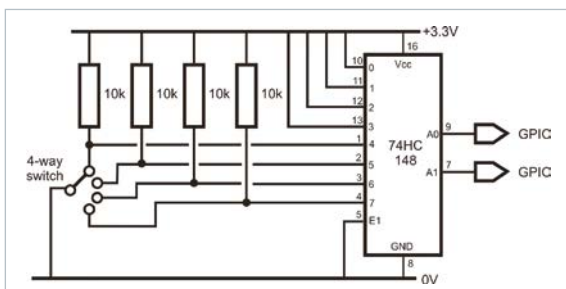
07 Eine siebensegmentige LED

Das Betreiben einer Siebensegment-LED ohne externe Logik erfordert sieben GPIO-Pins und eine achte, wenn man den Dezimalpunkt betreiben möchte. Dies kann durch die Verwendung eines 74HC4511 auf vier oder fünf reduziert werden. Dieser Siebensegment-Encoder schaltet die entsprechenden LEDs in der Siebensegment-Anzeige je nach 4-Bit-Binärwert an seinen Eingängen ein. Versuchen Sie mal herauszufinden, wie man eine 4-stellige Siebensegment-Anzeige fährt. Tipp: Sie multiplexieren die Pins.



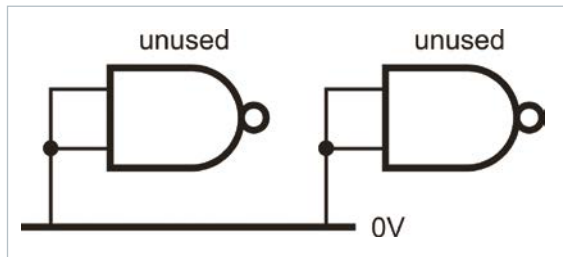
08 Einen Schalter kodieren

Es sind nicht nur Ausgänge, die LEDs leiten, die von der Logik profitieren können. Definitionsgemäß kann ein Mehrwege-Schalter nur eine seiner Positionen gleichzeitig geschlossen haben, so dass der Ausgang eines Vier-Wege-Schalters als 2-Bit-Binärzahl kodiert werden kann. Dies ist das Gegenteil von der Decodierung, die wir in Schritt 2 gesehen haben. Ein Chip, der dies tun kann, ist der 74HC148 (mit 8 Eingängen auf 3,3 V).



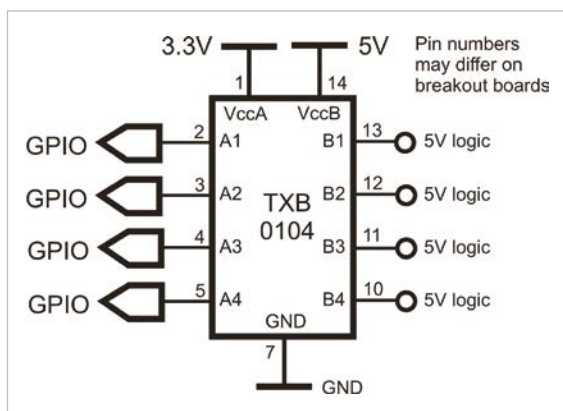
09 Verbinden Sie unbenutzte Eingänge

Bei Logikchips wird oft nur ein Teil des Chips verwendet, z.B. wenn Sie ein 74HC00 Quad NAND Gate haben und Sie nur zwei der vier Gates verwenden, während zwei unbenutzt bleiben. Dies ist kein Problem, aber Sie sollten unbenutzte Eingänge nicht unverbunden lassen, da sie oszillieren können, was dazu führt, dass der Chip übermäßige Strom zieht und überhitzen kann. Die Lösung besteht darin, alle nicht verwendeten Eingänge auf 3,3 V oder 0V zu verdrahten. Unbenutzte Ausgänge sind okay!



10 5V-Logik nutzen

Vielleicht möchten Sie mal eine 5V-Logikschaltung anschließen, die Sie nicht direkt mit dem Pi verbinden können. Es gibt mehrere Level-Converter-Chips, aber die meisten sind unidirektional. Allerdings sind die TXB0102-, TXB0104- und TXB0108-Chips von Texas Instruments (2, 4 und 8 Kanäle) bidirektional, was bedeutet, dass sie funktionieren, wenn die GPIO-Pins als Ein- oder Ausgänge konfiguriert sind. Die Texas-Geräte sind fummelige Oberflächenmontage-Chips, aber mehrere Unternehmen bieten die TXB0104 und TXB0108 auf einfacheren Breakout-Boards an.



Stromversorger

Wenn Ihre Schnittstellenschaltung und externen Geräte eine Versorgung von 3,3 V oder 5 V benötigen, können diese durch Pins am GPIO-Header bereitgestellt werden. Wenn Sie eine große Menge an externen Schaltkreisen haben, wäre es gut, Kondensatoren zwischen Versorger und Erdung zu schalten, um Schwankungen der Versorgung zu vermeiden, da dies zu Störungen des Stromkreises führen kann. Verwenden Sie einen 100µF-Kondensator plus mehrere 100nF-Keramik-Kondensatoren. Einen pro IC in Ihrem Schaltkreis und verdrahten Sie sie in der Nähe dieser ICs. Wenn Sie eine andere Versorgung als 3,3 V und 5 V benötigen, ist die einfachste Lösung, eine Batterie oder mehrere 1,5V-AA-Batterien zu verwenden. Wenn Sie eine bestimmte Spannung benötigen, kann ein Spannungsregler helfen.

Wie Sie ein interaktives Wearable bauen

Nutzen Sie die Twitter-Programmierschnittstelle (API), um ein Kleidungsstück bei bestimmten Tweets aufleuchten zu lassen.

Der Markt für Wearables explodiert mit Smartwatches, Fitness-Armbändern und dergleichen. Für viele ist diese Technologie nichts Neues, haben sie doch längst eigene Wearables entwickelt – so etwa Kleidung, die bei Berührung aufleuchtet, stimmverzerrende Masken oder leuchtenden Waffen. In diesem Tutorial werden Sie mit Ihrer alten LED-Lichterkette, Python und dem Pi Zero eine Mütze so verändern, dass sie leuchtet, wenn Sie bestimmte Tweets erhalten. Der Pi Zero ist mit seiner geringen Größe perfekt dafür geeignet. Sie können das Projekt auch für andere Kleidungsstücke anpassen, etwa Schuhe oder Handschuhe, die aufleuchten, oder einen Pulli, der auf das Wetter reagiert.

01 Twitter-API

Um mit Twitter interagieren zu können, benötigen Sie Schlüssel, die Sie von der Twitter-Website bekommen. Wenn Sie diese bereits haben, können Sie direkt zu Schritt 04 weitergehen. Andernfalls rufen Sie die Seite apps.twitter.com auf und loggen Sie sich ganz normal mit Ihren Zugangsdaten ein. Bestätigen Sie „Create New App“. Füllen Sie die geforderten Angaben aus und akzeptieren Sie die Nutzungsvereinbarung.

Was Sie brauchen

- Pi Zero
- USB-Powerbank
- USB-WLAN-Dongle
- Micro-USB-Konverter
- ein Kleidungsstück
- LED-Lichterkette

Links Sie können dies auf jede Art von Kleidung anwenden, was toll für Cosplay ist.



Your application has been created. Please take a moment to review and adjust your application's settings.

test for wearable tech

Test OAuth

Details Settings Keys and Access Tokens Permissions

test for setting up wearable tech application
http://www.tecoed.co.uk

Organization

Information about the organization or company associated with your application. This information is optional.

Organization None

Organization website None

Application Settings

Your application's Consumer Key and Secret are used to **authenticate** requests to the Twitter Platform.

Access level Read and write (modify app permissions)

Consumer Key (API Key) NPn0wKXmFJzS0l3BGnQyKz23 (manage keys and access tokens)

Callback URL None

Links Um die App zu nutzen, müssen wir sie zunächst bei Twitter registrieren.

02 Zugriffsrechte

Um den Vorgang zu vervollständigen, werden Sie auf die Übersichtsseite geführt. Wählen Sie „Access Level Permission“. Möglicherweise müssen Sie eine Mobilfunknummer eingeben, wie sie in den Einstellungen Ihres Twitter-Accounts steht. Folgende Auswahlmöglichkeiten gibt es:

Read: Die Tweets aus Ihrer Timeline werden gelesen.

Read and write: Die Einstellung ermöglicht es, sowohl Tweets zu lesen als auch Tweets zu schreiben und an die Timeline zu senden.

Read, write, direct: Hier wird das Lesen und Senden von Direktnachrichten erlaubt.

Für unser Projekt ist die Einstellung „Read and write“ erforderlich.

03 API-Schlüssel

Die API-Schlüssel werden gebraucht, um das Twitter-Konto über Python ansprechen zu können. Klicken Sie auf „Keys and access token“. Nun werden „consumer key“ und „consumer secret“ angezeigt. Notieren Sie beide, da Sie diese in Schritt 07 benötigen.

Am unteren Ende der Seite werden die Zugangsdaten angezeigt. Jedes Mal, wenn Sie den Button klicken, werden sie erzeugt. Das ist hilfreich, wenn unbefugte Dritte Kenntnis über Ihre Daten erlangt haben. Notieren Sie auch diese, da Sie sie ebenfalls in Schritt 07 benötigen.

04 WLAN einrichten

Der Pi Zero hat die optimale Größe für derartige Projekte und Hacks. Bei der Verwendung in Kleidungsstücken ist es unwahrscheinlich, einen Bildschirm für den Zugang zu haben. Eine einfache Möglichkeit ist der Zugang zum Pi per SSH. Dazu benötigt man WLAN. Am einfachsten ist es, den Pi an einen Monitor zu hängen, zu booten, den WLAN-Button rechts oben auszuwählen und den WEP-Key einzugeben. Speichern Sie die Einstellungen. Der Pi wird nun beim Start versuchen, sich in dieses Netzwerk einzuwählen.

Category:

- Session
- Logging
- Terminal
 - Keyboard
 - Bell
 - Features
- Window
 - Appearance
 - Behaviour
 - Translation
 - Selection
 - Colours
- Connection
 - Data
 - Proxy
 - Telnet

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address) raspberrypi Port 22

Connection type:

☐ Raw ☐ Telnet ☐ Rlogin ☒ SSH ☐ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings WinSCP temporary session

Load Save Delete

05 SSH zum Pi Zero

Downloaden und installieren Sie einen SSH-Client wie PuTTY auf Ihrem Laptop. Starten Sie PuTTY und geben Sie die IP-Adresse des Pi ein. Geben Sie nach Aufforderung den Nutzernamen und das Passwort ein, so kommen Sie zum Eingabefenster. Hier können Sie Code eingeben und ausführen und das Projekt aufsetzen.

06 Tweepy installieren

Die Twitter-App ist registriert, Sie haben die Keys und Tokens und den Zugang zu Ihrem Pi. Nun installieren Sie Tweepy, die Python-Twitter-API. Geben Sie Folgendes ins Terminal ein:

```
sudo apt-get install python-setuptools
sudo easy_install tweepy
```

Oder dies:

```
sudo pip install tweepy
```

Mit dem Befehl „sudo reboot“ den Pi neu starten. Nun können Sie über SSH Python-Code nutzen, um mit Ihrem Twitter-Feed zu interagieren.

Zero anbinden

Der Pi kann an ein Mobiltelefon angebunden werden, damit er auch unterwegs interaktiv ist. Dazu ist es empfehlenswert, das WLAN auszuschalten. Zur Installation tippen Sie „sudo nano /etc/network/interfaces“. Ergänzen Sie die Zeile „wireless-power off“. Die Datei speichern und den Pi neu starten.

07 Mit Twitter verbinden

Um Ihre Tweets zu streamen, müssen Sie mit Ihrem Consumer Key und Access Token aus Schritt 03 eine Verbindung genehmigen. Legen Sie durch Eingabe von „sudo nano [dateiname].py“ in die Kommandozeile eine neue Datei an. Fügen Sie die untenstehenden Befehlszeilen ein, damit Ihre Tweets zum Pi gestreamt werden. Zeile neun empfängt die Tweets Ihrer Timeline und „tweet_to_check = tweet.text“ greift sich die Tweets. Nutzen Sie „print tweet_to_check“, um Tweets anzuzeigen. Speichern Sie mit Strg + X und starten Sie dann das Programm mit „sudo python [dateiname].py“. Testen Sie, ob bis hierher alles funktioniert, bevor Sie mit Schritt 08 fortfahren.

```
import sys, subprocess, urllib, time, tweepy

consumer_key= 'xxxxxxxxxxxxx'
consumer_secret= 'xxxxxxxxxxxxx'

access_token= 'xxxxxxxxxxxxx'
access_token_secret= 'xxxxxxxxxxxxx'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(auth)

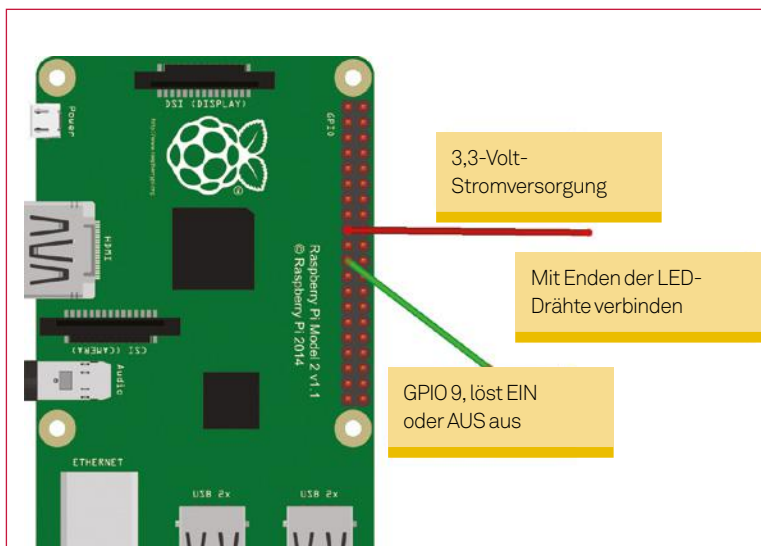
class Hat_Lights(tweepy.StreamListener):
    def on_status(self, tweet):
        tweet_to_check = tweet.text ###Holt Tweet
        print tweet_to_check

stream = tweepy.Stream(auth, Hat_Lights())
while True:
    stream.userstream()
```

08 LEDs vorbereiten

Je nach verwendetem Kleidungsstück gibt es zwei Arten, die LEDs mit Strom zu versorgen. Möchten Sie eine Mütze verwenden, gilt es, zusätzliches Gewicht zu vermeiden. Schneiden Sie ein Kabel zwischen LED und Akku durch. Trennen Sie auch ein Jumper-Kabel in der Mitte und löten es jeweils an.

Unten Dieses einfache Fritzing-Schaubild zeigt die Verbindungen.



Der Pi Zero kann 3,3 oder 5 Volt bereitstellen – befestigen Sie einen der Drähte zum Beispiel an dem GPIO-Pin 2, 4 oder 9 und den anderen an einen anderen Pin, etwa Pin 11. Sie können auch einen stromlosen Pin nehmen und mit der Masse verbinden, damit die Akkus die LEDs mit Strom versorgen. Es ist nicht ratsam, beides zu tun, da dann die LEDs bald durchbrennen würden.

09 LEDs testen

Testen Sie, ob LEDs und Verdrahtung funktionieren. Mit „sudo nano led_test.py“ eine neue Datei erstellen und den untenstehenden Code einfügen. Datei speichern und „sudo python led_test.py“ tippen. Die LEDs sollten für zehn Sekunden leuchten und dann ausgehen. Ersetzen Sie die GPIO-Pin-Nummern im Code durch diejenigen, die Sie tatsächlich verwendet haben:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(9, GPIO.OUT)

###Licht an
GPIO.output(9, GPIO.LOW)
time.sleep(10)

###Licht aus
GPIO.setup(9, GPIO.HIGH)
```

10 Module hinzufügen

Gehen Sie nun zurück zu dem Twitter-Testprogramm aus Schritt 07. Importieren Sie die geforderten Programme, die Python in die Lage versetzen, mit Twitter zu interagieren, und die LEDs zu steuern (Zeile drei bis sechs).

Die letzte Zeile fügt eine 15-Sekunden-Pause hinzu, damit der Pi genügend Zeit hat, vorher eine WLAN-Verbindung herzustellen. Andernfalls könnte es zu einer Fehlermeldung kommen.

```
import os
import time;
import sys, subprocess, urllib, time, tweepy
import RPi.GPIO as GPIO

time.sleep(15)
```

11 Keyword prüfen

Um festzustellen, ob ein Nutzer wirklich eine LED einschalten wollte, sucht das Programm nach einem speziellen Tweet, etwa „@ada_lovelace ON“. Fügen Sie die Programmzeile „tweet_to_check.find(\"@Ihre_Twitter_ID ON\")“ (Zeile 31) ein. Da die Phrase bei Position null startet, wird der Wert null sein. Ist der Wert nicht null, dann ist nicht genau diese Phrase gesendet worden. Eine einfache if-Anweisung kann dazu genutzt werden, um auf den Wert zu reagieren (Zeile 34).

```
does_the_tweet_contain_key_word = tweet_to_check.find("@Test_User ON")
```

12 Nutzernamen abfragen

Hat das Programm festgestellt, dass ein Twitternutzer die genaue Phrase getweetet hat, geht es darum, die Twitter ID dieses Nutzers zu ermitteln. (Zeile 38). Dies ermöglicht es Ihnen eine vorgefertigte Bestätigung zu senden, dass er die LEDs zum Leuchten gebracht hat. Fügen Sie die Zeile „user = str(tweet.user.screen_name)“ ein, um die ID einem wechselnden Nutzer hinzuzufügen.



Um Tweets zu streamen, müssen Sie eine Verbindung zulassen.

13 Zeit erfassen

Senden Sie denselben Tweet mehrfach, so vermutet Twitter, dass Sie ein Spam-Bot sind, und verhindert, dass die Nachricht gesendet wird. Um dies zu vermeiden, nehmen Sie die Zeit auf, zu der die LEDs geschaltet wurden, und fügen Sie sie der Nachricht hinzu (Zeile 39). So wird der Tweet auch in Echtzeit gesendet. Um den Zeitpunkt zu erfassen, verwenden Sie:

```
time_sent = time.asctime( time.localtime(time.time()) )
```

Die Ortszeit Ihres Pi wird ermittelt und als Sendezeit gespeichert. Stellen Sie sicher, dass die Uhr Ihres Pi richtig gestellt ist, bevor Sie das Programm starten.

14 Bild und Nachricht hinzufügen

Dieser Schritt kombiniert Ihre Antwort, ein Bild und die im vorherigen Schritt ermittelte Zeit, um dem Nutzer eine Nachricht zu senden, der Ihre LED zum Leuchten gebracht hat. Erstellen Sie eine neue Variable namens message und fügen Sie Text und Zeit hinzu: „message = "Die Mütze hat geleuchtet!", time_sent“ (Zeile 40). Suchen Sie ein passendes Bild, das sie mit der Nachricht senden wollen, und speichern Sie es ebenfalls als Variable: „pic = "/home/pi/lights.jpg““ (Zeile 37).

15 Nachrichtenteile zusammenfügen

Nehmen Sie die beiden Teile aus den vorangegangenen Schritten, um die endgültige Nachricht zu erzeugen. Sie wird als final_tweet gespeichert: „final_tweet = "@%s" %(user), message“ (Zeile 42). Der Codeteil „"@%s" %(user)“ fügt dem Tweet die Twitter-ID hinzu, damit diese auch in deren Timeline erscheint. Zuletzt nutzen Sie zum Senden den update-Code:

```
api.update_with_media(pic, final_tweet)
```

16 Licht an!

Dies ist derselbe Codeabschnitt wie in Schritt 09, der einen Stromkreis erzeugt. Er schaltet die LEDs an. Fügen Sie eine Zeitverzögerung hinzu, sodass die LEDs eine kurze Zeit anbleiben, bevor der Stromkreis wieder unterbrochen wird: „GPIO.setup(9, GPIO.HIGH)“ (Zeile 48).

```
###Licht an
GPIO.output(9, GPIO.LOW)
time.sleep(10)
###Licht aus
GPIO.setup(9, GPIO.HIGH)
```

17 Das Programm automatisch starten

Da das Programm in ein Kleidungsstück eingebaut ist und auf einem Pi Zero läuft, muss es automatisch gestartet werden, sobald Strom fließt. Damit das geschieht, tippen Sie „sudo nano /etc/rc.local“ ins Terminal. Fügen Sie am Ende der Datei den Pfad hinzu, etwa „python /home/pi/LED_Hat.py &“. Vergessen Sie nicht das & am Ende! Wenn Sie den Pi nun neu starten, wird das Programm automatisch ausgeführt. Jetzt heißt es Mütze auf und Ihre Follower dazu bringen, Ihnen Tweets zu senden!

Vollständiger Code

```
import os
import time;
import sys, subprocess, urllib, time, tweepy
import RPi.GPIO as GPIO

time.sleep(10)

####TWITTER SECTION###

# == OAuth Authentication ==#####
consumer_key= "xxxxxxxxxxxxxxxx"
consumer_secret= "xxxxxxxxxxxxxxxxxxxxxxxx"

access_token= "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
access_token_secret= "xxxxxxxxxxxxxxxxxxxxxxxx"

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(auth)

####CODE TO GET TWITTER TO LISTEN FOR Phrase###
class Hat_Lights(tweepy.StreamListener):
    def on_status(self, tweet):

        tweet_to_check = tweet.text ##gets the tweet
        print (tweet_to_check)

        ###Checks for tweets to @Your_Twitter_User_ID ###Add Yours
        does_the_tweet_contain_key_word = tweet_to_check.find("@Your_
                                                                Twitter_User_ID ON")

        ###Change to use code to find key word###
        print does_the_tweet_contain_key_word

        if does_the_tweet_contain_key_word == 0:
            GPIO.setmode(GPIO.BCM)
            GPIO.setup(9, GPIO.OUT)
            pic = "/home/pi/lights.jpg"
            user = str(tweet.user.screen_name)
            time_sent = time.asctime( time.localtime(time.time()) )
            message = "You turned on the LED Hat!", time_sent
            #print error_tweet
            final_tweet = "@%s" %(user), message
            #print type(error)

            ###Turn the lights on
            GPIO.output(9, GPIO.LOW)
            time.sleep(8)
            ###Turn them off
            GPIO.setup(9, GPIO.HIGH)
            #GPIO.output(10, GPIO.HIGH)lights_on()
            api.update_with_media(pic, final_tweet)

        else:
            print ("no lights")

stream = tweepy.Stream(auth, Hat_Lights())

while True:
    stream.userstream()
```



Der Pi als Minecraft-Konsole

Konstruieren Sie auf Basis des Raspberry Pi eine Konsole, mit der Sie Minecraft spielen können. Nebenbei üben Sie das Programmieren.

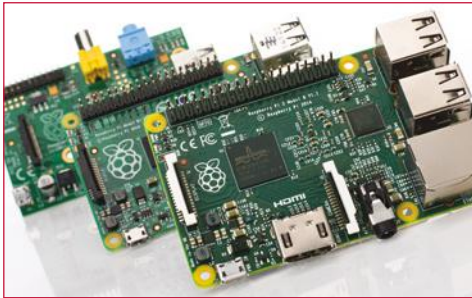
Wenn Pi-Besitzer von Minecraft sprechen, geht es oft darum, mithilfe dieses Spiels das Programmieren zu erlernen. Man kann jedoch auf dem Pi auch ganz normal Minecraft spielen und einfach nur Spaß haben.

Der Pi bietet Ihnen die einfachste Möglichkeit, ein voll funktionsfähiges Minecraft auf Ihren großen Fernseher zu bringen. Doch das Game ein-

fach nur auf dem TV-Gerät anzeigen zu lassen, ist natürlich noch nicht das Ende vom Lied. Wenn wir alle Funktionen nutzen, die der Pi bietet, machen wir aus ihm eine eigenständige Minecraft-Konsole mit Hack- und Modding-Qualitäten. In diesem Tutorial zeigen wir Ihnen, wie Sie sowohl Hardware als auch Software anpassen, einen Game-Controller anschließen und sogar modden können.

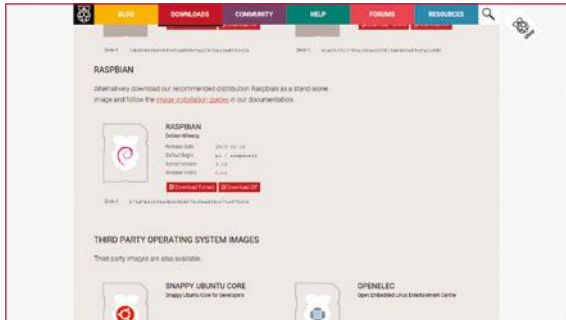
Was Sie brauchen

- Raspberry Pi 2
- Aktuelles Raspbian-Image
raspberrypi.org/downloads
- Minecraft: Pi Edition
pi.minecraft.net
- Raspberry-Pi-Gehäuse
- USB-Game-Controller (PS3 bevorzugt)



01 Ihren Raspberry Pi wählen

Bevor es losgeht: Alles, was wir in diesem Tutorial tun werden, funktioniert auf sämtlichen B-Modellen des Pi mit mindestens 512 MB RAM. Allerdings ist Minecraft: Pi Edition ziemlich rechenintensiv. Wir empfehlen Ihnen daher, für dieses Tutorial einen Raspberry Pi 2 zu verwenden.



02 Raspberry Pi vorbereiten

Minecraft: Pi Edition funktioniert derzeit nur mit Raspbian. Machen Sie eine frische Installation von Raspbian. Haben Sie bereits eine SD-Karte mit dem Betriebssystem, dann aktualisieren Sie es:

```
sudo apt-get update && sudo apt-get upgrade
```

03 Minecraft vorbereiten

Nutzen Sie die Standardinstallation von Raspbian, so ist Minecraft sogar schon vorinstalliert. Gehen Sie ins Menü und sehen Sie unter Games nach. Haben Sie gerade ein Update von Raspbian durchgeführt, können Sie das Spiel mit diesen Befehl installieren:

```
$ sudo apt-get install minecraft-pi
```

04 Ausprobieren

Mussten Sie Minecraft nachträglich installieren, sollten Sie zunächst testen, ob es auch funktioniert. Starten Sie den Desktop mit startx und rufen Sie Minecraft aus dem Menü auf. Minecraft: Pi Edition ist vom Funktionsumfang ziemlich limitiert, bietet aber immerhin Raum für Modding.

05 X-Setup

Läuft bei Ihnen eine frische Raspbian-Installation und/oder starten Sie Ihre Installation über die Kommandozeile, dann müssen Sie einstellen, dass auf den Desktop gebootet wird. Öffnen Sie das Terminal und tippen Sie raspi-config. Gehen Sie zur Option „Enable Boot Desktop“ und wählen Sie „Desktop“.



Oben Probieren Sie Minecraft: Pi Edition kurz aus, bevor Sie anfangen, Ihre Konsole zu bauen.

Nutzen Sie die Standard-Installation von Raspbian, so ist Minecraft sogar schon vorinstalliert.

06 Python einrichten

Wir werden Minecraft in einem späteren Teil dieses Tutorials ein wenig mit der Python-Programmiersprache verändern. Öffnen Sie das Terminal und tippen Sie:

```
$ cp /opt/minecraft-pi/api/python/mcpi ~/minecraft/
```

07 Mit Minecraft starten

Damit sich das Ganze auch wirklich nach Konsole anfühlt, sollte Minecraft direkt starten, nachdem man eingeschaltet hat. Um dies festlegen zu können, müssen wir das Autostart-Menü aufrufen:

```
$ sudo nano /etc/xdg/lxsession/LXDE-pi/autostart
```

08 Autostart-Sprache

Hier brauchen Sie einfach nur @minecraft-pi als letzte Zeile hinzuzufügen, abzuspeichern und noch einmal neu zu starten, um zu sehen, ob es funktioniert. Gut zu wissen, wenn Sie mal auch andere Programme während des Bootens starten wollen.

09 Ausschalten

Für den Moment können wir Maus/Keyboard nutzen, um den Pi ganz normal auszuschalten. In Zukunft werden Sie dafür aber die Stromzufuhr unterbrechen müssen. Das sollte kein Problem darstellen, solange Sie Ihre Minecraft-Welt zuvor gespeichert haben.

Updates zur Pi Edition?

Minecraft: Pi Edition hat eine ganze Weile schon keine Updates mehr bekommen. Was aber daran lag, dass das Spiel nur auf dem ursprünglichen Modell B lief. Jetzt, mit mehr Power unter der Haube, können wir vielleicht mehr Updates erwarten. Hinweise darauf gibt es allerdings noch keine. Sollte es so weit sein, updaten Sie Ihren Pi mit „sudo apt-get update && sudo apt-get upgrade“.



Es kann knifflig sein, die richtige Stromversorgung für den Raspberry Pi zu finden.



3D-Druck-Gehäuse

Aaron Hicks von Solid Technologies hat dieses Minecraft-Gehäuse für den Raspberry Pi entworfen und teilt es auf GrabCAD (bit.ly/2v1fRXh). Alles, was Sie machen müssen, ist, das STL-File an einen 3D-Drucker zu senden.

10 Das richtige Gehäuse

In diesem Szenario schließen wir den Raspberry Pi an einen Fernseher an. Wir benötigen darum ein Gehäuse, das die Elektronik vor Staub und Stößen schützt. Es gibt viele gute Gehäuse. Wir haben das PiBow von Pimoroni benutzt, da man es an die TV-Rückseite montieren kann. Sie können sich Ihr Gehäuse aber auch per 3D-Drucker ausdrucken.

11 Passende Stromversorgung finden

Es kann knifflig sein, die richtige Stromversorgung für den Raspberry Pi 2 zu finden, wenn Sie einen USB-Port oder ein mobiles Ladegerät verwenden möchten. Beiden fehlt es meist an Leistung. Holen Sie sich am besten die offiziell für den Pi erhältliche 2A-Stromversorgung.

12 Kabellos spielen

Uns ist bewusst, dass nicht jeder einen Ethernet-Anschluss in der Nähe seines Fernsehers hat. Daher ist es womöglich eine gute Idee, in einen WLAN-Adapter zu investieren. Eine Liste mit Modellen ist auf elinux.org/RPi_VerifiedPeripherals zu finden.

13 Maus und Keyboard

Jetzt da wir so weit sind, den Pi anzumontieren, sollten Sie sich mal Ihre Controller-Situation ansehen: Möchten Sie Kabelsalat oder soll gleich alles drahtlos funktionieren? Wir befassen uns auf der nächsten Seite mit Controllern, denken Sie aber schon jetzt darüber nach.

14 Bereit für SSH

Es ist einfacher, Skripte zu erstellen und auf Minecraft anzuwenden, wenn man sie via Netzwerk hochlädt, anstatt dies direkt auf dem Pi zu tun. Finden Sie über `iconfig` die IP-Adresse heraus. Sie können dann im Terminal eines anderen Netzwerk-Computers auf den Pi zugreifen. Tippen Sie dazu:

```
ssh pi@[IP address]
```

15 Ein Spielchen wagen

Zu diesem Zeitpunkt haben wir bereits eine voll funktionsfähige Minecraft-Konsole gebaut. Wenn Sie mögen, können Sie jetzt anfangen zu spielen, und einen Controller brauchen Sie auch nicht. Sie können auch zu Seite 82 springen, um zu erfahren, wie Sie Minecraft modden und besser Ihren Bedürfnissen anpassen. Möchten Sie einen Controller, lesen Sie hier weiter.



16 Controller-Unterstützung hinzufügen

Achten Sie darauf, dass Sie die Controller-Input-Funktionen auf dem Raspberry Pi installieren. Um das zu tun, greifen Sie wieder per SSH auf den Pi zu (das Passwort ist „raspberrypi“) und installieren folgendes Paket:

```
$ sudo apt-get install xserver-xorg-input-joystick
```

17 Controller-Mapping

Wir haben eine Controller-Map für den PS3-Controller, den Sie direkt auf Ihren Pi laden können. Mit ein paar Modifikationen lässt sich diese auch für andere USB-Controller verwenden. Gehen Sie in den Konfigurations-Ordner:

```
$ cd /usr/share/X11/xorg.conf.d/
```

18 Controller-Map ersetzen

Wir entfernen die aktuellen Joystick-Kontrollen mit „`sudo rm 50-joystick.conf`“ und ersetzen sie durch eine selbst erstellte Konfiguration mithilfe von:

```
$ sudo wget http://www.linuxuser.co.uk/wp-content/uploads/2015/04/50-joystick.conf
```



19 Erst rebooten, dann nutzen

Nach einem Reboot sollten Sie in der Lage sein, den Maus-Input auf der Konsole mit dem Gamepad zu steuern. R2 und L2 sind normale Mausklicks und können zur Navigation in den Minecraft-Menüs und im Spiel genutzt werden.



20 Vollbild-Ansicht

Wahrscheinlich ist Ihnen schon aufgefallen, dass Minecraft in einem Fenster läuft. Klicken Sie auf den Fullscreen-Button, damit sich der Bildschirm ganz füllt. Das kann aber die Maus-Nutzung erheblich einschränken. Zum Glück können Sie das Problem mit dem Gamepad umgehen. Im Spiel verwenden Sie die Sticks zum Bewegen und die Buttons für Inventar und anderes.

Xbox-Controller

Leider funktionieren Xbox-360-Controller mit Linux etwas anders. Sie nutzen ihre eigenen Treiber, die sich von den üblichen Joystick-Treibern, die wir für das Gamepad und andere USB-Controller benutzen, unterscheiden. Ein 360-Controller funktioniert nicht als Maus und es ist schwieriger, ihm spezielle Funktionen zuzuweisen. Das macht solche Aufgaben wie diese hier deutlich kniffliger.

Minecraft modifizieren

An dieser Stelle haben wir für Sie Beispiel-Code mit Erklärungen. So lernen Sie, wie man mit Python programmiert und Minecraft Pi moddet.

Wir programmieren Minecraft so, dass es auf die Python-Sprache reagiert. Dafür nutzen wir die API, die mit Minecraft: Pi Edition kommt. Diese haben wir an früherer Stelle in den Home-Ordner verschoben. Jetzt ist ein guter Zeitpunkt, sie zu testen. Das können wir etwa via SSH machen. Sie müssen nur in den Minecraft-Ordner gehen (cd), den wir im Home-Verzeichnis angelegt haben, und „nano test.py“ eingeben, um unser Testfile zu erstellen. Fügen Sie Folgendes hinzu:

```
from mcpi.minecraft import Minecraft
from mcpi import block
from mcpi.vec3 import Vec3
mc = Minecraft.create()
mc.postToChat("Hallo Minecraft!")
```

Abspeichern und dann starten mit:

```
$ python test.py
```

„Hallo Minecraft!“ sollte auf dem Schirm aufpoppen. Der Code importiert die Minecraft-Funktionen von den Files, die wir an anderer Stelle verschoben haben. Diese erlauben

es uns, Python zu nutzen, um mit Minecraft sowie mit den anderen Funktionen und importierten Modulen zu interagieren. Wir nutzen dann die mc-Instanz, damit wir über die postToChat-Funktion direkt etwas an Minecraft schicken können. Es gibt viele Möglichkeiten, wie Sie auf diesem Wege mit Minecraft interagieren können: zum Beispiel Blöcke platzieren, die dem Spieler folgen, oder ganze Strukturen erschaffen. Es gibt praktisch nichts, was Sie mit Python nicht machen können. Schauen Sie sich unter mcpipy.wordpress.com weitere Projekte an. Auf der nächsten Seite haben wir den kompletten Code für ein Spiel abgedruckt, bei dem der Spieler einen versteckten Diamanten suchen muss. Schreiben Sie es selbst oder laden Sie es sich auf dem Pi mit:

```
$ wget http://www.linuxuser.co.uk/tutorialfiles/
Issue134/ProgramMinecraftPi.zip
$ unzip ProgramMinecraftPi.zip
$ cp Program\MinecraftPi/hide_and_seek.py ~/minecraft
```

Wir programmieren Minecraft so, dass es auf Python reagiert.





Vollständiger Code

Importieren

Hier importieren wir alle nötigen Module und APIs, um Minecraft zu programmieren. Am wichtigsten sind die Files im mcpi-Ordner, die wir zuvor kopiert haben.

```
from mcpi.minecraft import Minecraft
from mcpi import block
from mcpi.vec3 import Vec3
from time import sleep, time
import random, math
```

Lokalisieren

Wir verbinden uns mit der ersten Zeile mit Minecraft. Dann suchen wir nach der Spielerposition und runden auf eine ganze Zahl auf.

```
mc = Minecraft.create()
playerPos = mc.player.getPos()

def roundVec3(vec3):
    return Vec3(int(vec3.x), int(vec3.y), int(vec3.z))
```

Bereichsfindung

Die Berechnung der Distanz zwischen Spieler und Diamant. Dies wird später im Code in Intervallen vollzogen, dabei werden einfach die Koordinaten der Positionen verglichen.

```
def distanceBetweenPoints(point1, point2):
    xd = point2.x - point1.x
    yd = point2.y - point1.y
    zd = point2.z - point1.z
    return math.sqrt((xd*xd) + (yd*yd) + (zd*zd))
```

Erstellen

Geben Sie eine zufällige Position des Diamanten in einem Umkreis von 50 Blöcken rund um die Spielerposition an.

```
def random_block():
    randomBlockPos = roundVec3(playerPos)
    randomBlockPos.x = random.randrange(randomBlockPos.x - 50, randomBlockPos.x + 50)
    randomBlockPos.y = random.randrange(randomBlockPos.y - 5, randomBlockPos.y + 5)
    randomBlockPos.z = random.randrange(randomBlockPos.z - 50, randomBlockPos.z + 50)
    return randomBlockPos
```

Start

Das ist die Eingangsschleife, die das Spiel startet. Bei jedem neuen Start wird die Position des Spielers ermittelt.

```
def main():
    global lastPlayerPos, playerPos
    seeking = True
    lastPlayerPos = playerPos
```

Benachrichtigung

Dieser Teil setzt den Block in die Umgebung und schleust eine Nachricht per postToChat zu der Minecraft-Instanz durch, damit der Spieler weiß, dass ein Mini-Game begonnen hat.

```
randomBlockPos = random_block()
mc.setBlock(randomBlockPos, block.DIAMOND_BLOCK)
mc.postToChat("A diamond has been hidden - go find!")
```

Prüfen

Wir timen den Spieler mit timeStarted und stellen die letzte Entfernung zwischen Spieler und dem Block ein. Jetzt starten wir die massive While-Schleife, die die Entfernung zwischen der sich ändernden Spielerposition und dem unbeweglichen Diamanten prüft. Befindet sich der Spieler innerhalb von zwei Blöcken vom Diamanten entfernt, hat er den Block gefunden und die Schleife endet.

```
lastDistanceFromBlock = distanceBetweenPoints(randomBlockPos, lastPlayerPos)
timeStarted = time()
while seeking:

    playerPos = mc.player.getPos()

    if lastPlayerPos != playerPos:
        distanceFromBlock = distanceBetweenPoints(randomBlockPos, playerPos)
        if distanceFromBlock < 2:
            seeking = False
```

Nachrichten erhalten

Sind Sie zwei oder mehr Blöcke vom Diamanten entfernt, teilt Ihnen das Programm mit, ob Sie näher dran oder weiter weg sind als bei Ihrer letzten geprüften Position. Dabei werden die neueste und die vorherige Position miteinander verglichen. Sind beide identisch, erscheint ein „Kälter“-Hinweis. Ist das erledigt, wird die letzte Position gespeichert.

```
else:
    if distanceFromBlock < lastDistanceFromBlock:
        mc.postToChat("Warmer " + str(int(distanceFromBlock)) + " blocks away")
    if distanceFromBlock > lastDistanceFromBlock:
        mc.postToChat("Colder " + str(int(distanceFromBlock)) + " blocks away")

    lastDistanceFromBlock = distanceFromBlock
```

Erfolg

Es bedarf einer Zwei-Sekunden-Pause, bevor die nächste Position über die Sleep-Funktion aktualisiert werden kann. Wird die Schleife unterbrochen, hält die Zeit an und Sie erfahren, wie lange Sie gebraucht haben, um den Diamanten zu finden. Das letzte Bit befiehlt Python, das Skript über die Hauptfunktion zu starten.

```
sleep(2)

timeTaken = time() - timeStarted
mc.postToChat("Well done - " + str(int(timeTaken)) + " seconds to find the diamond")

if __name__ == "__main__":
    main()
```

Ein Minesweeper-Spiel mit Minecraft erstellen

So programmieren Sie mit Ihrem Pi und Python-Wissen ein einfaches Mini-Game in Minecraft.

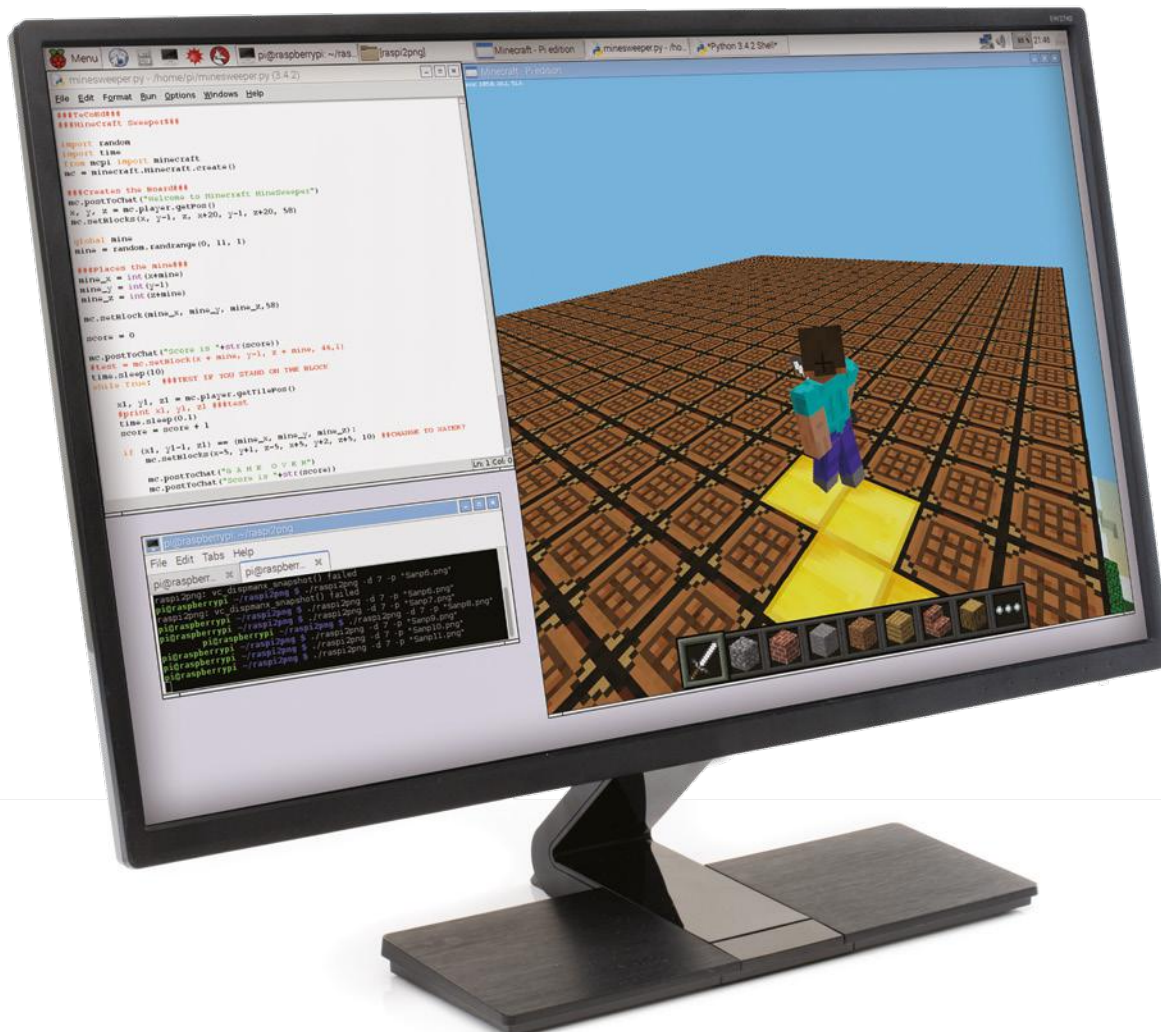
Sie erinnern sich vielleicht an das klassische *Minesweeper*-Spiel auf dem PC oder haben es sogar selbst schon gespielt. Über viele Jahre wurde es mit allen möglichen Betriebssystemen gebündelt, erschien für Handys und wurde sogar in einem Mini-Game in *Super Mario Brothers* featured.

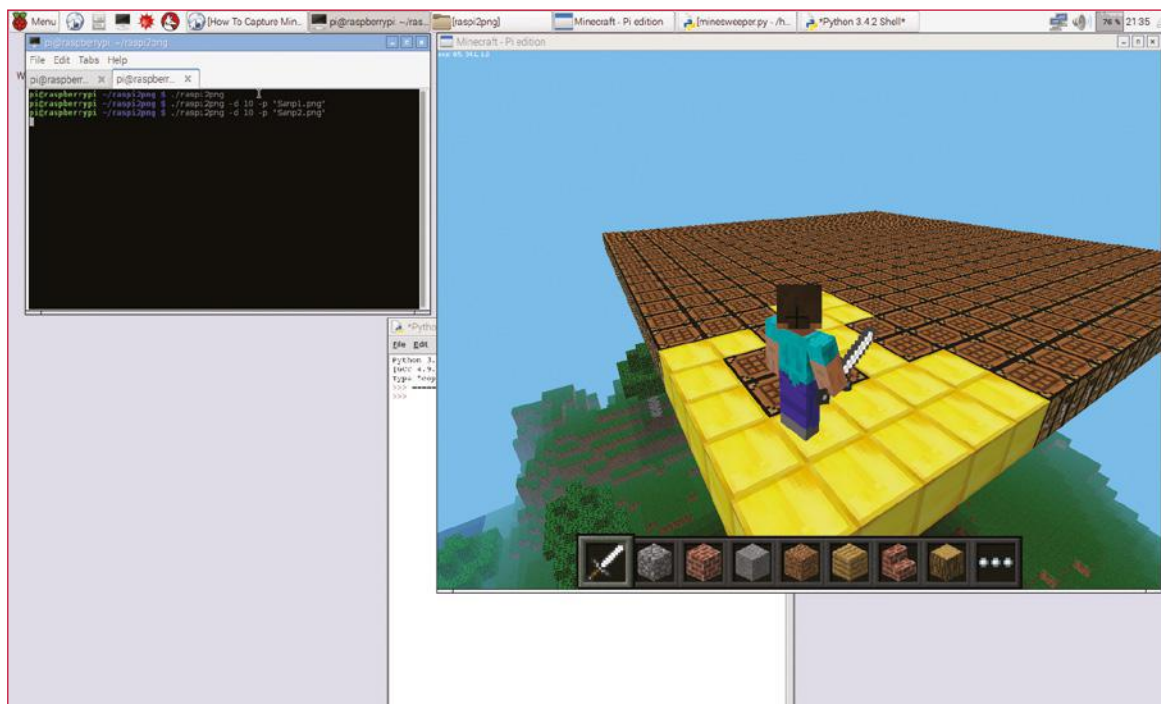
In diesem Projekt zeigen wir Ihnen, wie Sie eine einfache Version von *Minesweeper* in *Minecraft* erstellen. Sie werden ein Spiel programmieren, das in einer Arena aus Blöcken stattfindet. Das Spielekonzept besteht darin, dass Sie Ihren Spieler über das Spielfeld steuern. Jedes Mal, wenn Sie auf einem Block stehen, verwandelt sich dieser in Gold und Sie sammeln Punkte – aber passen Sie auf die Minen auf!

01 Aktualisieren und installieren
Zum Updaten Ihres Pis tippen Sie im Terminal

```
sudo apt-get upgrade
sudo apt-get update
```

ein. Das aktuelle Raspberry-Pi-Image verfügt bereits über *Minecraft* und Python. Die *Minecraft*-API, die es Ihnen ermöglicht, mit *Minecraft* über Python zu interagieren, ist ebenfalls mit an Bord. Wenn Sie eine alte Version des OS benutzen, dann lohnt sich der Download, um entweder auf das neue Jessie- oder Raspbian-Image upzudaten (www.raspberrypi.org/downloads).





Links Die sicheren Blöcke wurden in Gold verwandelt, die übrigen sind potenzielle Minen!

02 Module importieren

Laden Sie Ihren bevorzugten Python-Editor und öffnen Sie ein neues Fenster. Sie müssen die folgenden Module importieren: **import random**, um die zufällige Position der Mine zu bestimmen, und **import time**, um Pausen und Verzögerungen hinzuzufügen. Als nächstes fügen Sie zwei weitere Zeilen Code hinzu: **minecraft** and **mc = minecraft.Minecraft.create()**. Damit erstellen Sie den Programm-Link zwischen Minecraft und Python. Die MC-Variante erlaubt es Ihnen „mc“ anstatt „minecraft.Minecraft.create()“ zu schreiben.

```
import random
import time
from mcpi import minecraft
mc = minecraft.Minecraft.create()
```

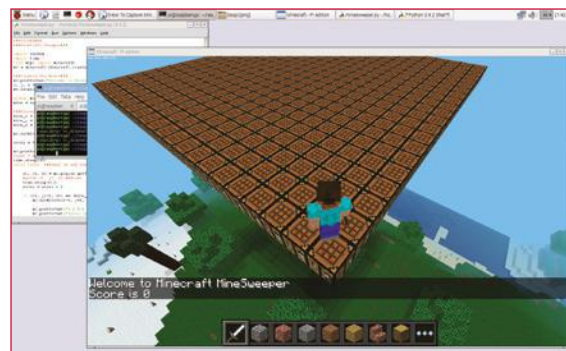
03 Blumen wachsen lassen

Erstellen Sie das unten stehende Programm und testen Sie, ob es funktioniert. Jeder Block hat seine eigene ID-Nummer; Blumen haben die 38. Die Zeile **x, y, z = mc.player.getPos()** bestimmt die aktuelle Position des Spielers in der Spielwelt mittels x-, y- und z-Koordinaten. Jetzt wissen Sie, wo in der Welt Sie stehen. Blöcke können mit **mc.setBlock(x, y, z, flower)** platziert werden. Speichern Sie, öffnen Sie MC und erstellen Sie eine neue Welt.

```
flower = 38
while True:
    x, y, z = mc.player.getPos()
    mc.setBlock(x, y, z, flower)
    time.sleep(0.1)
```

04 Code ausführen

Verringern Sie die Größe des MC-Fensters, damit Sie es leichter haben, Code und laufendes Programm gleichzeitig zu sehen. Starten Sie das Python-Programm und warten Sie, bis es geladen ist. Während Sie sich umherbewegen, lassen Sie Blumen fallen. Ändern Sie die ID-Nummer in Zeile 1, um den Block-Typ zu ändern. Versuchen Sie es mal mit Gold oder Wasser anstatt mit Blumen!



05 Nachricht in der Minecraft-Welt posten

Es ist auch möglich, Nachrichten in der *Minecraft*-Welt zu posten. Dies wird später im Spiel dafür benutzt, den Spieler darüber zu informieren, dass das Spiel gestartet ist, und um den Punktestand anzuzeigen. Fügen Sie Ihrem vorherigen Programm diese Code-Zeile bei **flower = 38** hinzu: **mc.postToChat("I grew some flowers with code")**. Speichern und starten Sie das Programm mit F5 – die Nachricht poppt auf! Ändern Sie die Nachricht oder gehen Sie zum nächsten Schritt, um das Spiel zu starten.

06 Spielfeld erstellen

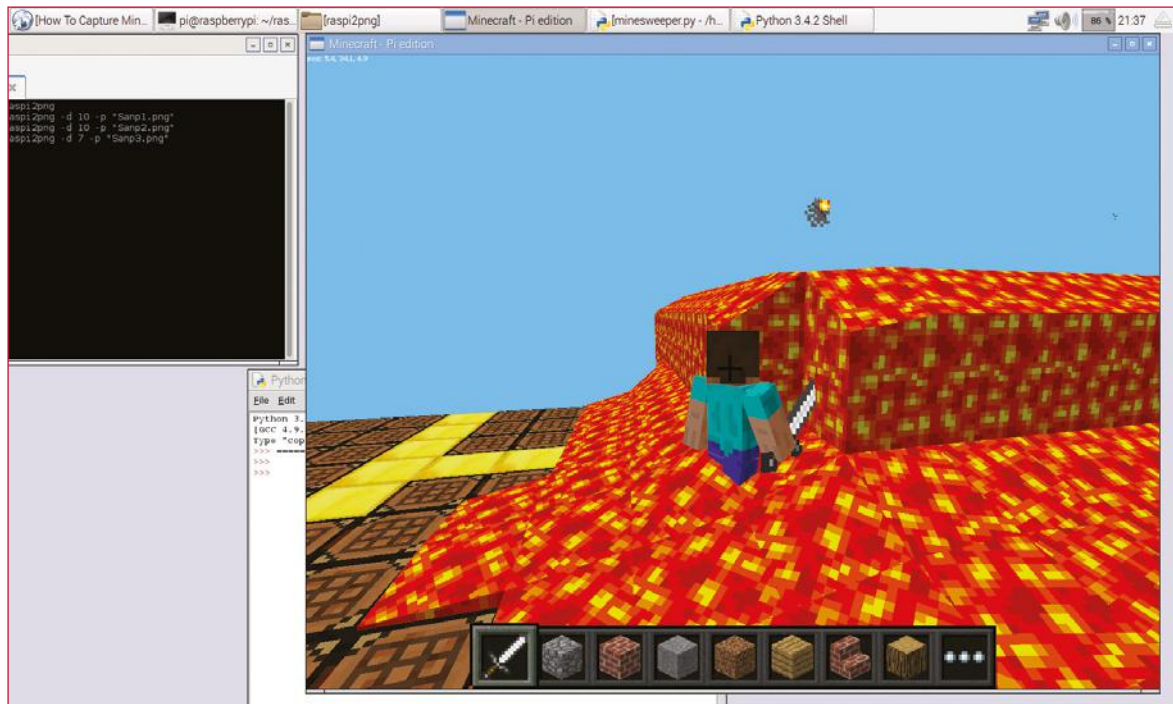
Das Spielfeld wird dort erstellt, wo der Spieler steht. Es empfiehlt sich, in die Luft zu fliegen und aus dieser Perspektive nach einem flachen Stück Landschaft zu suchen. Um das Spielfeld zu erstellen, müssen Sie die aktuelle Spielerposition mittels **x, y, z = mc.player.getPos()** festlegen. Nutzen Sie dann den Code **mc.setBlocks**, um die Blöcke für das Spielfeld zu erstellen.

```
mc.setBlocks(x, y-1, z, x+20, y-1, z+20, 58).
```

Die Nummer 58 ist die ID des Blocks, mit dem Sie ihr Spielfeld erstellen. Sie können Sie Ausmaße des Feldes vergrößern oder verkleinern, indem Sie **+20** ändern. Im Code-Beispiel oben beträgt die Feldgröße 20 x 20 Blöcke, also eine Arena aus 400 Blöcken.

Zur Shell wechseln

Es kann frustrierend sein, zwischen der Python-Shell und dem *Minecraft*-Fenster zu wechseln, besonders deshalb, weil MC das Python-Fenster überlagert. Am besten halbiert man beide Fenster auf dem Schirm (betreiben Sie MC nicht Fullscreen, da sonst die Mauskoordinaten nicht stimmen). Nutzen Sie die Tabulatur-Taste, um Keyboard und Maus freizugeben.



Rechts Nichts sagt „Game Over“ so deutlich, wie eine Lava-Eruption!

07 Die Mine erstellen

Im vorangegangenen Schritt haben Sie die Spielerposition auf dem Spielfeld bestimmt. Die x-, y- und z-Koordinaten können ebenfalls dazu benutzt werden, die Mine zu positionieren. Der Code `mine = random.randrange(0, 11, 1)` erstellt eine zufällige Anzahl zwischen 1 und 10. Kombinieren Sie das mit der aktuellen Spielerposition und fügen Sie die zufällige Anzahl seiner Position hinzu. Damit generieren Sie einen zufälligen Minen-Block.

```
mine_x = int(x+mine)
mine_y = int(y-1)
mine_z = int(z+mine)
```

Nutzen Sie `setBlock`, um die Mine zu platzieren: `mc.setBlock(mine_x, mine_y, mine_z, 58)`. Mit `y-1` stellen Sie sicher, dass der Block auf derselben Ebene, auf der sich auch das Spielfeld befindet, positioniert wird. Somit ist dieser unsichtbar. Die Nummer 58 ist die Block-ID. Diese können Sie ändern. Dadurch sehen Sie dann, wo sich der Block befindet. Das ist auch praktisch, um zu testen, ob der Rest des Codes funktioniert. Setzen Sie den Wert aber wieder zurück, bevor Sie spielen!

08 Punktestandvariable erstellen

Mit jeder Sekunde, in der es Ihnen in diesem Spiel gelingt, am Leben zu bleiben, erhalten Sie einen Punkt. Erstellen Sie eine Variable, um die Punkte zu speichern und den Anfangswert auf Null zu setzen. Verwenden Sie den Code `postToChat`, um den Punktestand zu Spielbeginn anzuzeigen. Seien Sie sich bewusst, dass MC keinen Wert an den Chat ausgeben kann, also wird der Punktestand zunächst in eine Zeichenfolge konvertiert.

```
score = 0
mc.postToChat("Score is "+str(score))
time.sleep(10)
```

09 Checken Sie die Spielerposition

Bis jetzt haben Sie ein Spielfeld sowie einer Mine in derselben Farbe erstellt, sodass Sie sie nicht sehen können. Als

nächstes müssen Sie prüfen, ob der Spieler zu Beginn nicht auf der Mine steht. Dafür brauchen Sie einen **while**-Loop, der kontinuierlich überprüft, ob die Spielerposition sicher ist. Mine meiden! Sonst ist das Spiel gleich vorbei! Da die Spielerkoordinaten dafür benutzt wurden, um das Ausgangsspielfeld zu erstellen, müssen wir die Spielerposition mit `x1, y1` und `z1` erneut bestimmen.

```
while True:
    x1, y1, z1 = mc.player.getTilePos()
```

10 Einen Punkt bitte

Jetzt, da der Spieler sich einen Block weiter bewegt hat, erhält er einen Punkt. Bei dieser simplen Aktion wird einfach der Wert 1 dem bereits bestehendem Punktwert hinzugefügt. Wir erreichen das mit `score = score + 1`. Da es sich um eine Schleife handelt, erhält der Spieler bei jeder Bewegung einen Punkt.

```
time.sleep(0.1)
score = score + 1
```

11 Die Spannung steigt ...

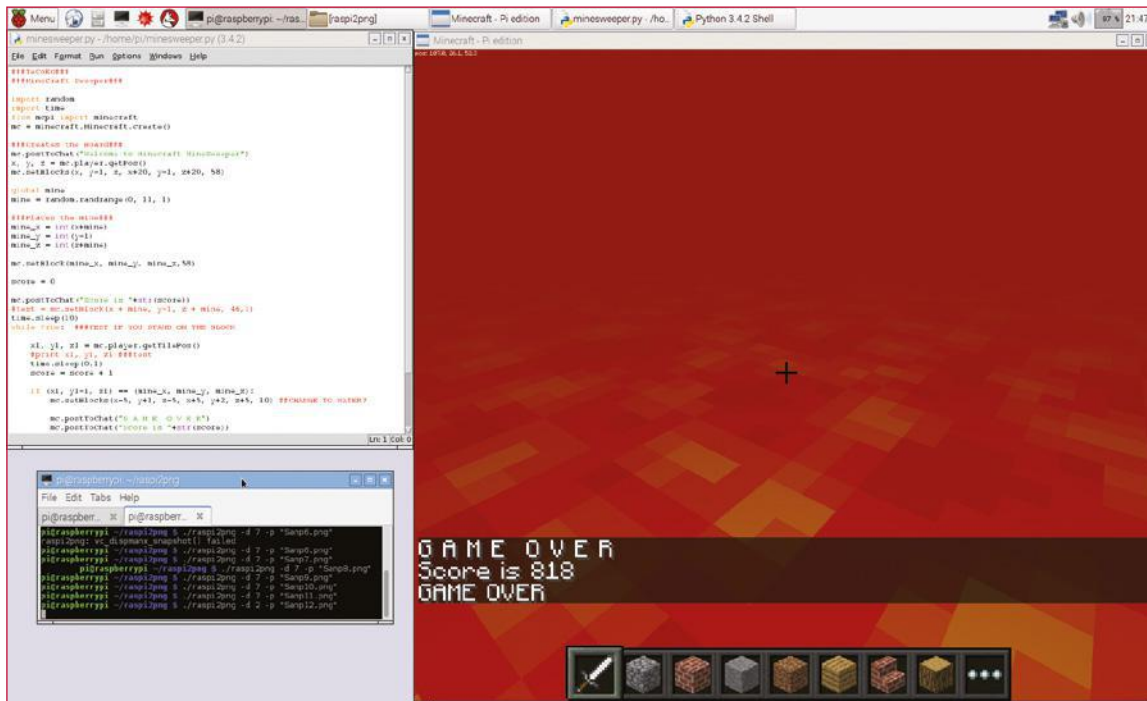
Nachdem Sie mit einem Punkt belohnt wurden, geht es im nächsten Schritt darum, zu prüfen, ob der Block, auf dem Sie stehen, eine Mine ist. Wir müssen also dafür sorgen, dass die Blöcke `x1, y1, z1` unter Ihnen, mit der Position `mine_x, mine_y, mine_z` der Mine abgeglichen werden. Sind die Werte gleich, stehen Sie auf einer Mine. Im nächsten Schritt programmieren Sie die Explosion.

```
if (x1, y1-1, z1) == (mine_x, mine_y, mine_z):
```

12 Mine auslösen

Stehen Sie auf der Mine, wird sie hochgehen. Für diese Effekt nutzen Sie die Lava-Blöcke. Sie können den Code `mc.setBlocks` verwenden, um Blöcke zwischen zwei Punkten festzulegen. Lava-Blöcke reagieren auf Gravitation. Wenn sie sie also über dem Spieler platzieren, fließt das Lava auf den Spieler herunter.

```
mc.setBlocks(x-5, y+1, z-5, x+5, y+2, z+5, 10)
```

Andere Minecraft-Hacks

Wenn Sie Spaß dabei haben, mit Minecraft zu programmieren, dann freuen Sie sich sicherlich über die Nachricht, dass es noch viele andere tolle Raspberry-Pi-basierte Projekte gibt, die Sie testen können. Unser Experte hat für Sie unter tecoed.co.uk/minecraft.html einige zusammengestellt. Auch die Jungs von Adventures in Minecraft haben einige tolle Guides: stuffaboutcode.com/p/minecraft.html.

Links Das fertige Spiel nutzt die Chat-Konsole, um Ihnen den Punktestand mitzuteilen.

13 Game Over

Wenn Sie auf der Mine stehen, ist das Spiel vorbei. Nutzen Sie den Post-to-chat-Code, um ein „Game Over“ in der Minecraft-Welt anzeigen zu lassen.

```
mc.postToChat("G A M E O V E R")
```

14 Finaler Punktestand

Der letzte Teil des Spiels besteht darin, einen Punktestand auszugeben. Dabei wird die Punktevariable genutzt, die Sie in Schritt 8 festgelegt haben, sowie der Befehl `mc.postToChat`. Konvertieren Sie den Punktestand zuerst in eine Zeichenfolge, damit er auf dem Schirm angezeigt werden kann. Da Ihre Runde zu Ende ist, fügen Sie ein Statement hinzu, damit der Loop endet und der Code nicht mehr ausgeführt wird.

15 Sicherer Block

Aber was ist, wenn Sie die Mine ausgelassen haben? Das Spiel geht weiter und Sie müssen wissen, wo auf dem Spielfeld Sie sich befinden haben. Nutzen Sie den Code `mc.setBlock(x1,y1-1,z1,41)`, um den Block, auf dem Sie stehen, in ein anderes Material wie z.B. Gold zu ändern. Im Code ist die Y-Position Y-1, was den Block unter den Füßen des Spielers auswählt.

16 Punkte hochzählen

Für jeden Goldblock, den Sie finden, bekommen Sie einen Punkt. Das erreichen Sie dadurch, dass die Punktevariable jedes Mal um 1 erhöht wird, wenn Sie einen Block in Gold verwandeln und der Loop von Neuem losgeht. Der Code `postToChat` sagt Ihnen, dass Sie einen weiteren Schritt überlebt haben!

```
score = score + 1
mc.postToChat("You are safe")
```

17 Das Spiel ausführen

Damit ist der Code für dieses Programm abgeschlossen. Speichern Sie ihn und starten Sie ein Minecraft-Spiel. Sobald die Welt erstellt wurde, starten Sie das Python-Programm. Gehen Sie zum Minecraft-Fenster zurück und Sie sehen das Spielfeld!

Vollständiger Code

```
import random
import time
from mcpi import minecraft
mc = minecraft.Minecraft.create()

###Creates the board###
mc.postToChat("Welcome to Minecraft MineSweeper")
x, y, z = mc.player.getPos()
mc.setBlocks(x, y-1, z, x+20, y-1, z+20, 58)

global mine
mine = random.randrange(0, 11, 1)

###Places the mine###
mine_x = int(x+mine)
mine_y = int(y-1)
mine_z = int(z+mine)
mc.setBlock(mine_x, mine_y, mine_z, 58)

score = 0
mc.postToChat("Score is "+str(score))

time.sleep(5)
while True: ###Test if you are standing on the mine

    x1, y1, z1 = mc.player.getTilePos()
    #print x1, y1, z1 ###test
    time.sleep(0.1)
    score = score + 1

    if (x1, y1-1, z1) == (mine_x, mine_y, mine_z):
        mc.setBlocks(x-5, y+1, z-5, x+5, y+2, z+5, 10)
        mc.postToChat("G A M E O V E R")
        mc.postToChat("Score is "+str(score))
        break
    else:
        mc.setBlock(x1, y1-1, z1, 41)

mc.postToChat("GAME OVER")
```

Licht mit Ihrem Pi steuern

Wenn die Winternächte länger werden, können Sie den Pi und ein mobiles Gerät nutzen, um Ihre Lichter fernzusteuern.

Die Leute bei Energie haben einige geniale Steckdosen erschaffen, die von Ihrem Raspberry Pi ein- und ausgeschaltet werden können. Sie können ein Starter-Kit kaufen, welches das Add-on-Board mit dem RF-Transmitter und zwei Steckdosen enthält und mit dem Sie gleich loslegen können. Das Add-on-Board verbindet sich direkt mit den GPIO-Pins und wird über eine Python-Bibliothek gesteuert. Nachdem alles installiert und eingerichtet ist, können Sie mittels der Pi-mote und einem einfachen Programm Ihren Raspberry Pi dazu nutzen, bis zu vier Energie-Dosen zu steuern. Dieses Tutorial erklärt, wie es geht.

01 Setup

Booten Sie Ihren Raspberry Pi, laden Sie das LX Terminal und updaten Sie dann Ihre Software, indem Sie Folgendes eingeben:

```
sudo apt-get update
sudo apt-get upgrade
```

Abhängig davon, welche Version des OS Sie nutzen, müssen Sie möglicherweise die Python-GPIO-Bibliotheken installieren. (Bei Raspbian Jessie ist diese Bibliothek bereits vorinstalliert, sodass Sie diesen Schritt überspringen können.) Tippen Sie nun diese Zeile ein:

```
sudo apt-get install python-rpi.gpio
```

Nach der Fertigstellung rebooten Sie Ihren Pi. Dies wird die Python GPIO-Bibliotheken installieren, d.h. Sie können auf die Pins nun mit Python-Code zugreifen und diese steuern.



Was Sie brauchen

■ Pi-mote-IR-Steuerplatine mit RC-Buchsen
bit.ly/1MdpFOU

■ Tischlampe

■ Zubehör

Oben Übernehmen Sie mit ihrem Smartphone die Kontrolle über Ihr Zuhause.



02 Energie-Bibliothek installieren

Installieren Sie nun die Energie-Bibliotheken. Mit diesen können das Pi-mote-Board und der Pi mit Python interagieren. Im LX Terminal tippen Sie – abhängig von Ihrer verwendeten Python-Version – entweder:

```
sudo apt-get install python3-pip
sudo pip-3.2 install energenie
```

... für Python 3, oder:

```
sudo apt-get install python-pip
sudo pip install energenie
```

... für eine ältere Version. Energenie wird seine Software zukünftig aktualisieren, sodass Sie möglicherweise überprüfen müssen, ob Sie die aktuelle Version verwenden. Zum Updaten der Software geben Sie Folgendes ein:

```
sudo pip install energenie --update
```

03 Pi-mote anbringen

Bevor Sie den Pi-mote-Sender anbringen, fahren Sie Ihren Raspberry Pi mit **sudo poweroff** herunter. Trennen Sie ihn von der Stromversorgung und stecken Sie das Modul an Ihren Pi an. Der „L“-Teil der Platine passt auf den gegenüberliegenden HDMI-Port. Schalten Sie den Pi ein und schließen Sie eine Ihrer Energenie-Steckdosen im gleichen Raum an, in dem sich auch Ihr Pi befindet. Die Reichweite ist ziemlich gut, aber Möbel und Wände können manchmal das Signal blockieren. Sie können prüfen, ob die Dose arbeitet, indem Sie zum Beispiel eine Tischlampe einstecken und dann auf die grüne Taste auf der Steckdose drücken. Das schaltet die Dosenleiste und damit auch die Lampe ein und aus.

04 Den Setup-Code herunterladen

Bevor der Raspberry Pi mit der Dose interagieren und sie ein- bzw. ausschalten kann, muss der Steuerungscode für den Transmitter programmiert werden. Jede Dose hat ihren eigenen einzigartigen Code, sodass Sie bis zu vier Stück individuell steuern können. Energenie liefert das Setup-Programm dafür mit.

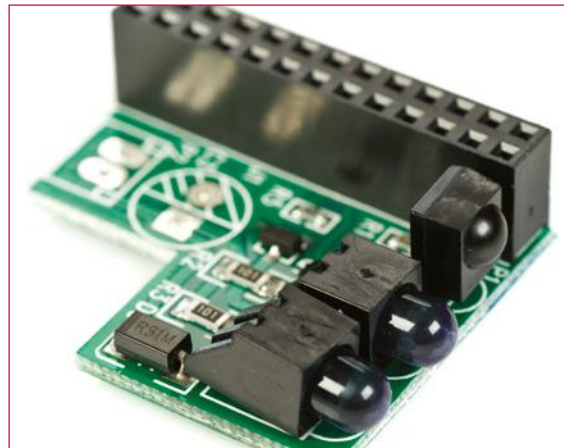
05 Steckdose einrichten

Starten Sie das Setup-Programm, wenn Sie es heruntergeladen haben. Damit geht die Dose in den „Lern“-Modus, der auch visuell signalisiert wird, wenn die LED vorn auf der Dose langsam blinkt. Passiert das nicht, dann drücken und halten Sie den grünen Knopf für mindestens fünf Sekunden und lassen ihn wieder los, wenn die LED beginnt, in einem Ein-Sekunden-Intervall zu blinken. Wenn Sie das Programm starten, wird es ein Signal aussenden. Folgen Sie den Anweisungen auf dem Bildschirm und drücken bei Bedarf die „Return“-Taste.

Wenn der Code akzeptiert wurde, wird das mit einem kurzen Blinken der LED am Gehäuse angezeigt. Wenn Sie mehr als eine Dose einrichten, verfahren Sie mit den anderen ebenso.

06 Ein schneller Test

Bevor Sie sich der Erstellung Ihres Python-Codes zuwenden, ist es eine gute Idee, zu überprüfen, ob die Steckdose ordnungsgemäß funktioniert. Stellen Sie die Stromversorgung der Wandsteckdose sicher und ebenso, dass die Lampe eingeschaltet ist. Sie können die Lampe ausschalten, indem Sie den grünen Knopf vorn an der Energenie-Steckdose drücken. Drücken Sie ihn erneut, sollte sich die Lampe wieder einschalten.



Links Der Pi-mote-Sender ist ganz einfach zu bedienen. Er wird vom Pi mit Strom gespeist und sendet in einem offenen Loop-System.

07 Code zum Einschalten der Steckdose

Die Python-Energenie-Bibliothek macht es unglaublich einfach, einen Code zu erstellen, der die Steckdose und damit auch Ihre Lampe einschalten wird. Und bald werden Sie dann mit dem Pi auch Heizkessel oder Fernseher ein- und ausschalten!

Öffnen Sie den Python-Editor und beginnen Sie ein neues Programm. Importieren Sie die Raspberry Pi GPIO-Library (Zeile 1 unten) und dann die Energenie-Library (Zeile 2 und 3). Fügen Sie nun den Code ein, der die Dose einschaltet (Zeile 4). Speichern Sie ab und starten Sie Ihr Programm. Die Dose wird eingeschaltet, Sie hören vielleicht einen Klick und dann geht Ihre Lampe an.

```
import RPi.GPIO as GPIO
import energenie
from energenie import switch_on
energenie.switch_on(1)
```

08 Dosen ein- und ausschalten

Da Sie der Dose nicht gesagt haben, sich abzuschalten, bleibt sie immer an! Um die Dose nach 5 Sekunden abzuschalten, importieren Sie zu Beginn des Programms die **time**-Funktion (Zeile 2 unten). Fügen Sie nun das Kommando zum Abschalten der Dose ein (Zeile 5). Aktivieren Sie nun mithilfe des **sleep**-Kommandos (Zeile 7) eine Pause und schalten Sie die Lampe dann ab (Zeile 8). Speichern Sie das Programm und starten Sie es.

```
import RPi.GPIO as GPIO
import time
import energenie
from energenie import switch_on
from energenie import switch_off
```

```
energenie.switch_on(1)
time.sleep(5)
energenie.switch_off(1)
```

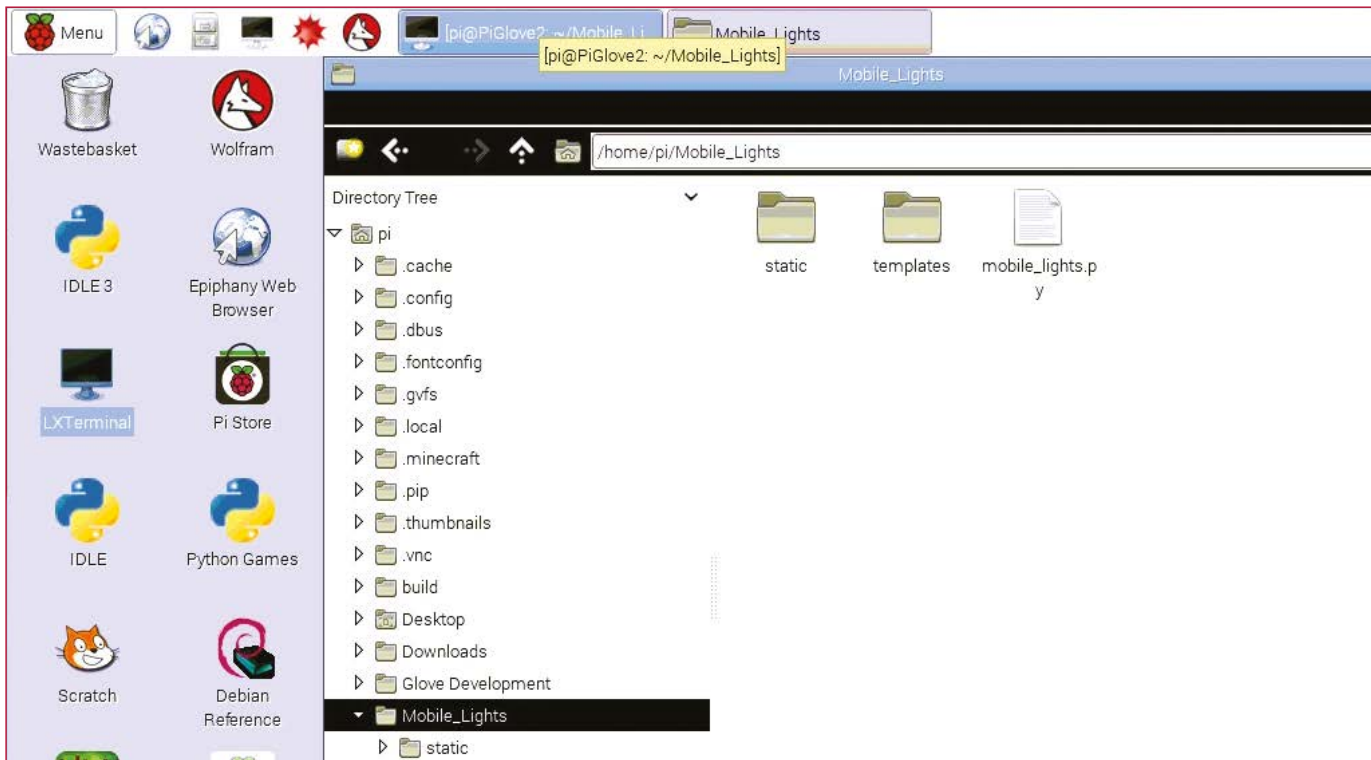
09 Webbasierte Anwendung erstellen

Es ist möglich, die Funktionsweise zu erweitern, sodass Sie die Lampe auch von einem mobilen Gerät wie Ihrem Handy, Notebook oder Tablet bedienen können. Der erste Schritt dazu ist, Ihren Raspberry Pi als Webserver einzurichten, der eine Webseite hostet, auf der eine EIN/AUS-Option angezeigt wird. Diese Buttons sind interaktiv und steuern die Steckdose. Öffnen Sie das LX Terminal und installieren Sie Pip und Flask:

```
sudo apt-get install pip
sudo pip install flask
```

IP-Adresse

Jedes Gerät im Internet hat eine Internetprotokolladresse (IP-Adresse). Das ist eine Zahlenkombination, mit deren Hilfe die Geräte in einem Netzwerk, das bis zu tausende Geräte enthalten kann, lokalisiert und identifiziert wird. Die meisten Heimnetzwerk-Adressen beginnen mit den Zahlen 192.168, wobei Ihr Router meist die 192.168.1.1 ist.



Oben Die Ordernamen müssen korrekt sein, damit Dateien ordnungsgemäß gespeichert werden.

10 CSS und HTML

Um die Webseite zu erstellen, müssen Sie je eine HTML- und CSS-Datei erstellen. HTML (HyperText Markup Language) ist die Programmiersprache für Internetseiten. Ihr Browser liest HTML-Dateien und konvertiert diese in Webseiten, sodass Bilder und Objekte in die Seiten eingebunden werden können. Cascading Style Sheets (CSS) ist der Code, der beschreibt, wie die Webseite aussehen wird, also die Präsentation des HTML-Inhalts. CSS enthält die Anweisung, wie die Elemente auf Ihrem Gerät gerendert werden. In diesem Tutorial steuert es, wie die Ein- und Aus-Optionen präsentiert werden und auf dem Bildschirm aussehen.

11 Neuen Ordner erstellen

Wenn Flask installiert ist, starten Sie Ihren Pi neu. Geben Sie **sudo reboot** ein. Erstellen Sie den neuen Ordner **Mobile_Lights** im Ordner **/home/pi**. Tippen Sie dazu im LX Terminal **mkdir Mobile_Lights** oder rechtsklicken Sie ins Fenster und wählen „Neuer Ordner“. Hier speichern Sie Python-Programm, CSS- und HTML-Datei.

12 Die HTML-Dateien

Öffnen Sie den **Mobile_Lights**-Ordner und erstellen Sie einen neuen Ordner namens „**templates**“. In diesem Ordner wird die HTML-Datei gespeichert, die die Struktur für das Webseiten-Layout enthält. Der Code benennt den Webseiten-Tab und – noch wichtiger – fügt die Links für „An“ und „Aus“ hinzu.

Öffnen Sie aus dem Startmenü einen Texteditor oder nutzen Sie **nano** und erstellen Sie eine neue Datei. Fügen Sie den HTML-Code von unten ein und speichern Sie die Datei als „**index.html**“ in den Template-Ordner. Denken Sie daran, dass eine HTML-Datei die Endung **.html** haben muss:

```
<!doctype HTML>
<HTML>
<head>
<title>Light Controller</title>
```

```
<link rel="stylesheet" href="/static/style.css" />
<meta name="viewport" content="width=device-width, user-scalable=no" />
</head>
<body>
<div class="on"><a href="/on/">ON</a></div>
<div class="off"><a href="/off/">OFF</a></div>
</body>
</HTML>
```

13 Etwas Stil hinzufügen

Das Cascading Style Sheet (CSS) wird genutzt, um auf einer Webseite einen „Button“-Stileffekt zu erstellen und anzuwenden. Erstellen Sie im **Mobile_Lights**-Ordner den neuen Unterordner „**static**“. Hier wird die CSS-Datei gespeichert. Erstellen Sie eine andere Textdatei und fügen Sie den Code unten ein, der den Stil der Webseite festlegt. Ab Zeile 20 können Sie die Farbe der Button anpassen. Speichern Sie die Datei als „**style.css**“ in den **Static**-Ordner. Beachten Sie, dass Sie eine CSS-Datei mit der Dateiendung **.css** abspeichern müssen:

```
body {
    position: absolute;
    margin: 0;
    top: 0;
    right: 0;
    bottom: 0;
    left: 0;
    font-family: Arial, sans-serif;
    font-size: 150px;
    text-align: center;
}

div {
    display: block;
```




```
pi@raspberrypi ~$ sudo hostname -I
192.168.1.125
pi@raspberrypi ~$
```

Oben Beim Befehl `hostname` empfiehlt es sich, als Operator lieber `-I` als `-i` zu verwenden, da Letzterer nur dann funktioniert, wenn der Hostname aufgelöst werden kann.

```
width: 100%;
height: 50%;
}

div a {
width: 100%;
height: 100%;
display: block;
}

div.on {
background: black;
}

div.off a {
color: white;
}

div.off a {
color: black;
}

a:link, a:visited {
text-decoration: none;
}
```

14 Alles zusammensetzen

Nun schreiben Sie noch das Python-Skript, das die `index.html` und die `style.css` mit dem Energie-Steckdosen-Steuerscode kombiniert, ähnlich dem in Schritt 7 verwendeten.

Öffnen Sie IDLE und darin ein neues Fenster, fügen Sie folgenden Code hinzu und speichern Sie als „`mobile_lights.py`“ in Ihren `Mobile_Lights`-Ordner. Zeile 4 nutzt `route()`, um Flask zu sagen, dass es das HTML-Template nutzen soll, um eine Webseite zu erstellen. Die Zeilen 7 und 11 `app.route('/on/')` und `app.route('/off/')` sagen Flask, dass die Funktion beim Klicken der URL ausgelöst werden soll. In Zeile 15 startet die `run()`-Funktion den lokalen Server mit Ihrer App. `if __name__ == '__main__':` stellt sicher, dass der Webserver nur läuft, wenn das Skript direkt vom Python-Interpreter ausgeführt wird – und nicht als importiertes Modul.

```
from flask import Flask, render_template
from energie import switch_on, switch_off

app = Flask(__name__)
```

```
@app.route('/')
def index():
    return render_template('index.HTML')

@app.route('/on/')
def on():
    switch_on()
    return render_template('index.HTML')

@app.route('/off/')
def off():
    switch_off()
    return render_template('index.HTML')

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0')
```

15 Ihre IP-Adresse finden

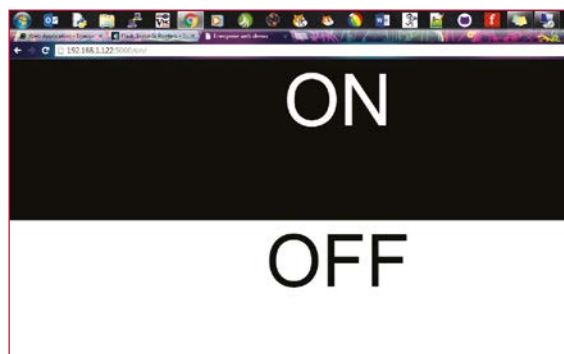
Bevor Sie den Webserver starten, müssen Sie Folgendes überprüfen:

- Sie haben einen Ordner namens `Mobile_Lights`
- Im `Mobile_Lights`-Ordner befindet sich eine Python-Datei namens `mobile_lights.py`
- Im `Mobile_Lights`-Ordner befinden sich zwei Unterordner: Zum einen „`templates`“ mit der Datei `index.HTML` und zum anderen der Ordner „`static`“, der die Datei `style.css` enthält.

Wenn das alles stimmt, geben Sie im LX Terminal den Befehl `sudo hostname -I` ein. Dadurch wird die IP-Adresse Ihres Raspberry Pi angezeigt – etwa `192.158.X.X`. Notieren Sie diese. Sie geben sie dann auf Ihrem Mobilgerät in den Webbrowser ein.

16 Webserver starten

Nun können Sie den Webserver starten. Lokalisieren Sie den `Mobile_Lights`-Ordner durch den Befehl `cd Mobile_Lights`. Starten Sie nun das Python-Programm `mobile_lights.py` durch den Befehl `sudo python mobile_lights.py`. Das startet den Webserver, der dann bereit ist, auf die Schaltflächen zu antworten, die auf der Webseite gedrückt werden.



17 Licht ein- und ausschalten

Rufen Sie auf Ihrem Mobilgerät den Webbrowser auf und geben Sie in der Adresszeile die IP-Adresse aus Schritt 15 ein. Fügen Sie am Ende der Adresse „`:5000`“ hinzu – etwa `192.168.1.122:5000`. Die 5000 ist die Nummer des Ports, der für die Kommunikation zwischen Ihrem Gerät und dem Raspberry Pi geöffnet wird. Ihnen werden die Optionen ON und OFF angezeigt und Sie können nun die Steckdose und die dort eingesteckten Geräte (Lampe, Wasserkocher, Radio etc.) von Ihrem Mobilgerät aus steuern, indem Sie einfach ON oder OFF klicken. Viel Spaß!

Flask

Flask ist ein leistungsfähiges Tool zur Erstellung interaktiver Webseiten und Apps. Wenn Sie mehr darüber lernen und andere Projekte ausprobieren möchten, sollten Sie <http://flask.pocoo.org> besuchen. Auf dieser Webseite finden Sie Beispiele, wie Flask und Python für Anwendungen und Lösungen genutzt werden. <http://flask.pocoo.org/community/poweredby>

Streamen Sie Internet-TV über den Pi

Mit Miro landen Ihre Lieblingssendungen und Podcasts automatisch auf Ihrem Fernseher.

Was Sie brauchen

- Raspbian Wheezy
- HDMI-Kabel
- Monitor oder Fernseher

An interessante Sendungen zu kommen, kann eine Weile dauern. Ob Sie nun Internet-TV-Programme, Video- oder Audio-Podcasts oder online verfügbare Serien suchen – bis Sie das Richtige gefunden und heruntergeladen haben, bleibt am Ende womöglich kaum noch Zeit, die Sachen überhaupt anzuschauen; vor allem, wenn Sie einen aktiven Lebensstil pflegen und viel zu tun haben.

Mit der Medienmanagement-Software Miro läuft all das automatisch ab, zumal auf dem Raspberry Pi. Eine Steilvorlage, um ein kompaktes Mediensystem zum Herunterladen und Abspielen interessanter Sendungen zu bauen. Gezielt geplantes Fernsehen auf Abruf ist das Schlagwort; das Projekt eignet sich also perfekt dazu, über ein bestimmtes Thema stets auf dem Laufenden zu bleiben.

01 Raspbian anpassen

Miro läuft leider nicht unter dem neuen Raspbian Jessie, sondern benötigt die vorherige Variante Wheezy, welche unter downloads.raspberrypi.org/raspbian/images noch verfügbar ist. Überprüfen Sie die HDMI-Verbindung zwischen Pi und Bild-

schirm. Miro ist eine Desktopanwendung und muss per Maus und Tastatur konfiguriert werden.

02 Miro installieren

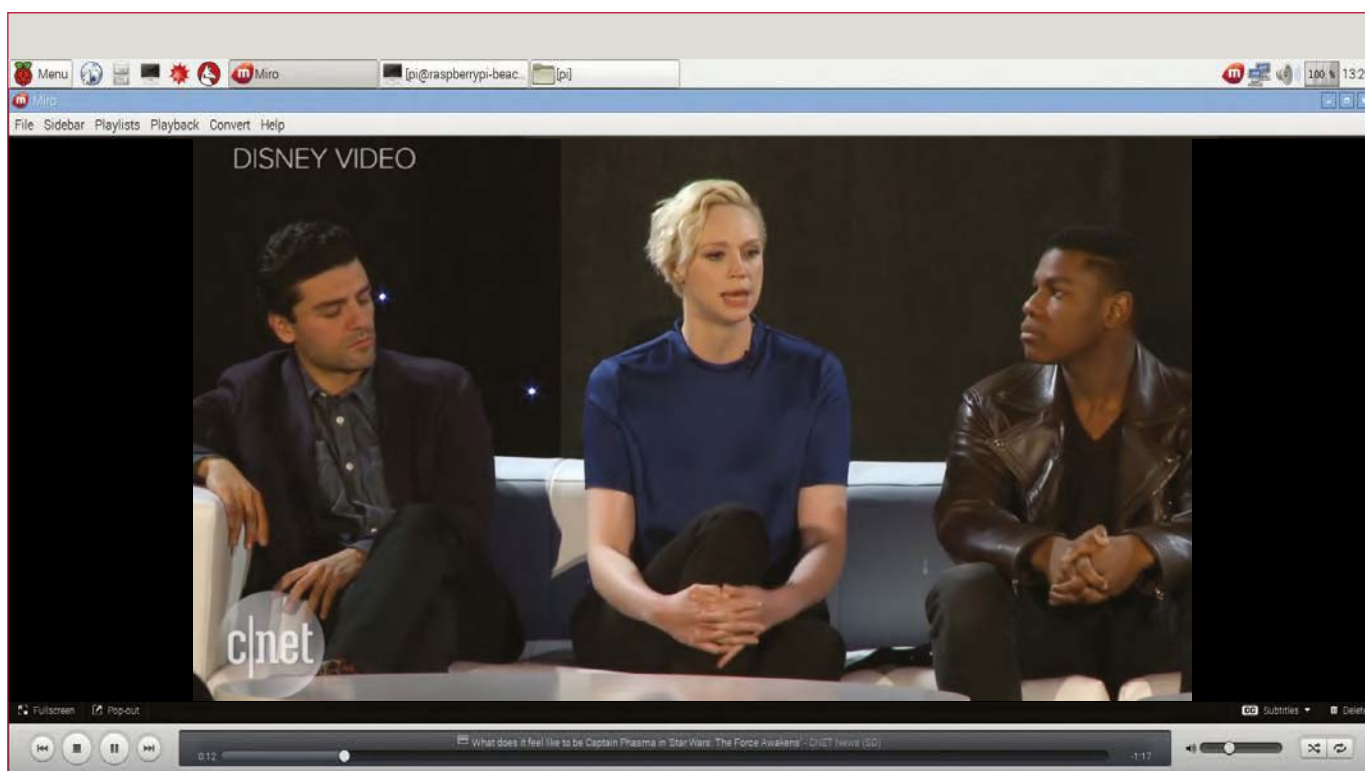
Wheezy ist auf der SD-Karte, der Pi läuft – öffnen Sie nun das Terminal und geben Sie ein:

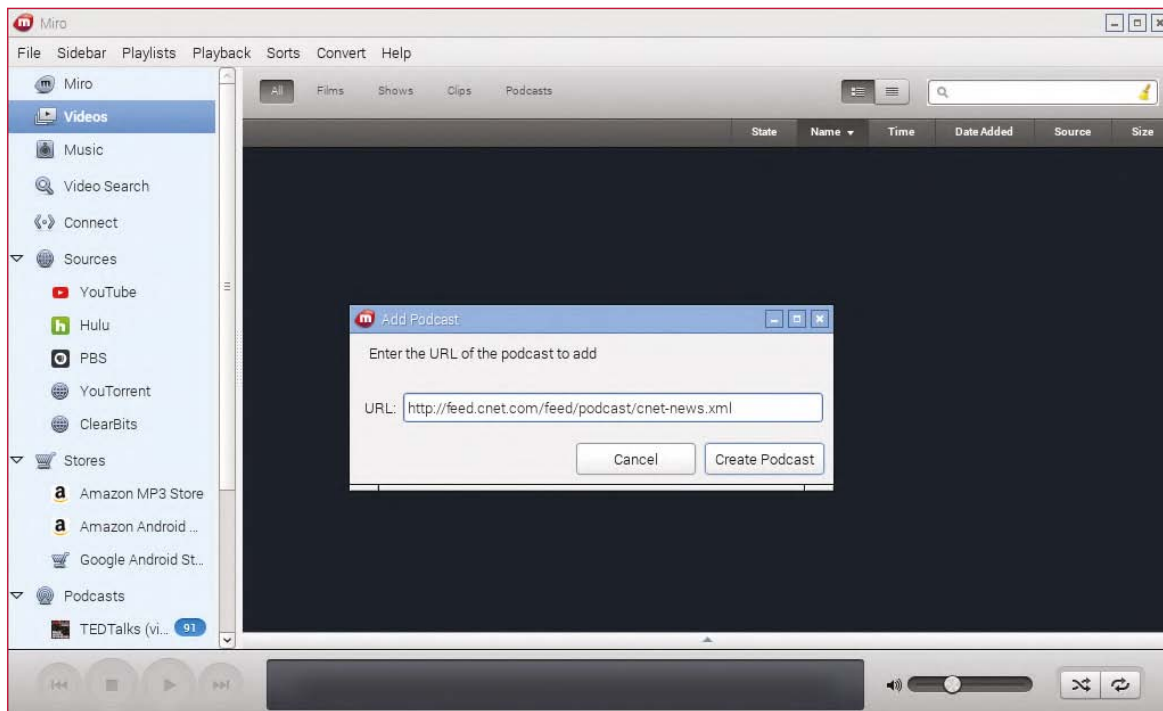
```
sudo apt-get install miro
```

Die Installation dauert ein wenig. Ist sie abgeschlossen, finden Sie Miro in Menu > Sound and Video. Starten Sie das Programm per Mausklick.

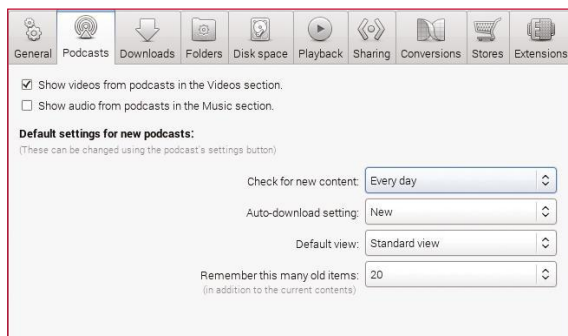
03 Autostart von Miro

Miro soll direkt nach dem Hochfahren starten. Unter File > Preferences > General können Sie dafür die Optionen „Automatically run Miro when I log in“ und „When starting up Miro remember what screen I was on when I last quit“ aktivieren. Ihr Pi sollte direkt in X booten; per raspi-config lässt sich das anpassen.





Links Sie können vielerlei Inhalte abonnieren, von Internetfernsehen bis zu Nachrichten-Podcasts.

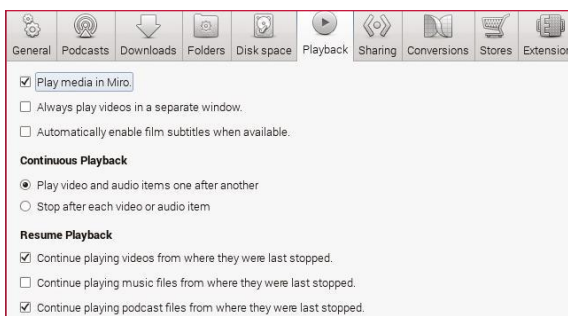


04 Inhalte aktualisieren

Wählen Sie im Reiter „Podcasts“ die Option „Show videos from podcasts in the Videos section“ an. Rechts im Fenster können Sie einstellen, wie oft Miro auf neue Inhalte prüfen soll – in dieser Frequenz werden dann die gespeicherten Feeds aktualisiert.

05 Wiedergabeeinstellungen

Im Reiter „Playback“ können Sie festlegen, dass alle Medien direkt in Miro abgespielt werden sollen. So beschränken Sie die Abhängigkeit von anderen Programmen und schonen Ressourcen. Empfehlenswert ist auch die Option „Play video and audio items one after another“ sowie die erste und dritte Option unter „Resume Playback“.



Je mehr Links Sie hinzufügen, desto mehr aktuelle Inhalte werden heruntergeladen.

06 Videos und Podcasts finden

Jetzt können die Inhalte also abgespielt werden – nur finden müsste man sie noch! Am besten geht das, indem Sie auf Websites, die Sie häufiger besuchen, Podcasts und Videos suchen und deren XML-Links kopieren.

07 Podcast-Feeds einrichten

Öffnen Sie File > Add Podcast in Miro und fügen Sie die URL des Podcasts in das Dialogfeld ein. Bestätigen Sie per Klick auf „Create Podcast“. Je mehr Links Sie hinzufügen, desto mehr aktuelle Inhalte werden auf Ihren mit Miro ausgestatteten Medien-Pi heruntergeladen und sind sofort bereit zum Anschauen.

08 Erste Nutzung

Zu Beginn hatten Sie eingestellt, dass Miro sich beim Start auf bestimmte Weise verhält. Jetzt können Sie dies in die Tat umsetzen, indem Sie im linken Seitenfenster den Bereich Videos öffnen und dort das erste abspielen. Jedes Mal, wenn Sie das System hochfahren, öffnet Miro diese Ansicht und beginnt mit der Wiedergabe.

09 Kein YouTube

Miro ist wirklich eine sehr gute Möglichkeit, sich ein benutzerdefiniertes Videoportal zum beliebigen Abruf zu bauen – leider ist es aber für YouTube-Videos ungeeignet. Das schränkt Sie nicht unbedingt drastisch ein, denn es gibt unzählige andere Kanäle für Medien, doch es sollte erwähnt werden. Es ist schade, beeinflusst aber nicht die Kernfunktion – Ihr Raspberry Pi ist jetzt ein für Sie maßgeschneidertes Videoportal!

Aktualisierungen abrufen

Eine hohe Aktualisierungsfrequenz einzustellen, scheint verlockend, doch zu häufige Überprüfungen gehen auch zulasten der Systemressourcen, was wiederum dazu führen kann, dass die gerade laufende Wiedergabe für eine Aktualisierung unterbrochen wird – unschön! Schränken Sie die Aktualisierungen besser auf stündliche oder tägliche Wiederholungen ein.

Web-Traffic mit Pi-Tor-Router anonymisieren

Verwenden Sie Ihren Raspberry Pi als Wireless Access Point und anonymisieren Sie Ihren Traffic.

Sind Sie wegen Ihrer Online-Privatsphäre besorgt? Sie befürchten, dass Geheimdienste Ihre Aktivitäten im Internet mitverfolgen? Oder sind Sie einfach nur kein Fan von Werbung, die speziell auf Sie zugeschnitten ist? Egal was Ihre Motivation ist: Sie können Ihren Raspberry Pi als anonymen Wireless Access Point (WAP) für das Tor-Netzwerk einrichten, um dann mit allen Ihren WLAN-Geräten darauf zuzugreifen.

Bei Tor handelt es sich um eine freie Software, die online Anonymität ermöglicht und den Internetverkehr durch ein anonymes Netz von über 7.000 Relais führt. Auf diese Weise können Ihre Standorte verborgen werden. Auch Analysen zu Ihrem Surfverhalten können so nicht mehr erstellt werden.

Hier geht es um den Schutz Ihrer Privatsphäre, indem Sie Besuche auf Websites, Instant Messages und E-Mails schwerer zu verfolgen machen. Alles, was dazu benötigt wird, ist ein Raspberry Pi, ein Drahtlos-Dongle, ein Ethernetkabel und etwas Konfigurationsarbeit.

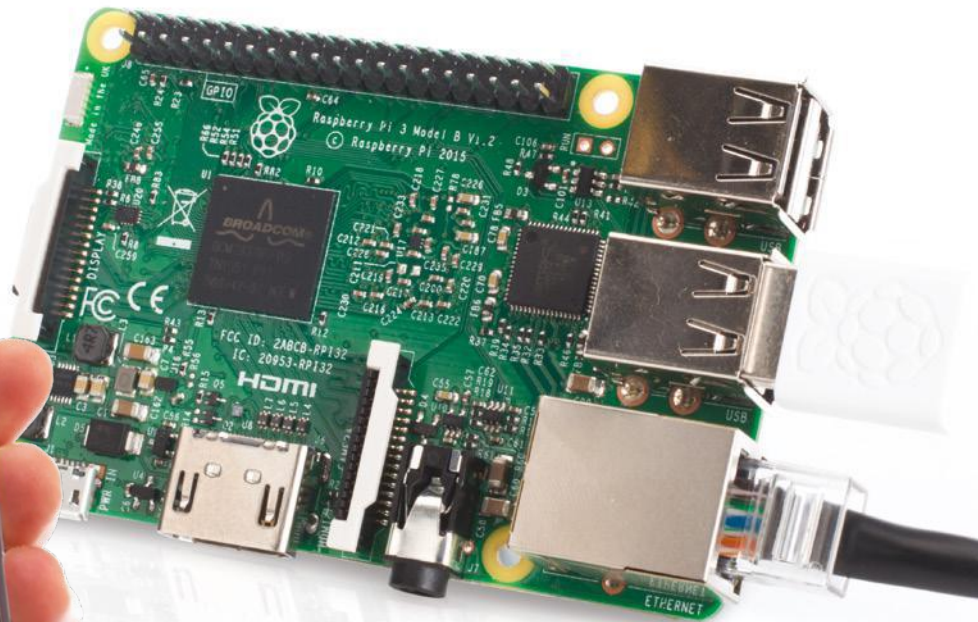
01 Alles verbinden

Wenn alles angeschlossen und eingeschaltet ist, können Sie mit der Konfiguration des Pi-Routers beginnen. Starten Sie damit, ein Ende des Ethernetkabels in den Raspberry Pi und das andere in einen freien Port an Ihrem Router zu stecken. Sollten Sie ein Pi-Modell vor 2016 verwenden, verbinden Sie einen Drahtlos-Dongle damit. Pi-3-Nutzer brauchen das nicht zu tun, da das Gerät bereits über WLAN verfügt.

02 Fernkonfiguration mittels SSH

Der nächste Schritt besteht darin, den Pi mit dem PC zu verbinden. Dazu verwenden wir einfach SSH. Sollte das nicht möglich sein, verbinden Sie den Mini-Rechner für den Rest der Anleitung mit einem Monitor sowie Maus und Tastatur. Öffnen Sie nun das Terminal und geben Sie ein:

```
ssh pi@192.168.0.27
```



Was Sie brauchen

- Ethernetkabel
- Wireless-Router
- USB-WLAN-Adapter oder Raspberry Pi 3

Links Sobald Sie Ihren Tor-Router eingerichtet haben, können Sie anonym mit Ihrem Smartphone surfen – sofern Sie mit dem WLAN des Routers verbunden sind.



```
pi@raspberrypi:~$ ssh pi@192.168.0.27
Warning: Permanently added '192.168.0.27' (ECDSA) to the list of known hosts.
pi@192.168.0.27:~$
```

Der Raspberry Pi hat den Standardbenutzernamen „Pi“ und die IP-Adresse 192.168.0.27. Wenn Ihre Distribution nicht über SSH verfügt, können Sie den openssh-Client oder PuTTY installieren. Sie dürfen aber auch über SSH bei Windows mit PuTTY kommunizieren.

Sie finden die IP-Adresse Ihres Raspberry Pi, indem Sie die Admin-Seite Ihres Routers überprüfen oder einen Monitor und eine Tastatur anschließen und ifconfig eingeben.

03 Zugangspunkt installieren

Als Nächstes verbinden Sie den Pi über SSH und geben `iwconfig` ein. So sehen Sie, ob der Wireless-Adapter erkannt wird. Dann aktualisieren Sie die Paketliste mit „`sudo apt-get update`“ und installieren die Wireless-Access-Point-Software:

```
sudo apt-get install hostapd isc-dhcp-server
```

```
pi@raspberrypi:~$ sudo nano /etc/dhcp/dhcpd.conf
# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.
```

04 DHCP-Einstellungen bearbeiten

Nun müssen wir die Datei `/etc/dhcp/dhcpd.conf` bearbeiten, um zu vermeiden, dass IP-Adressen zufällig zugewiesen werden:

```
sudo nano /etc/dhcp/dhcpd.conf
```

Suchen Sie nach den Zeilen, die mit „`option domain-name`“ beginnen, und kommentieren Sie sie mit einem `#` wie folgt aus:

```
#option domain-name-servers ns1.example.org, ns2.
example.org;
#option domain-name "internal.example.org";
```

Im Folgenden finden Sie „`authoritative`“. Löschen Sie hier das `#` aus der Zeile.

05 IP-Adressen angeben

Gehen Sie bis zum Ende des Dokuments und fügen Sie die folgenden Zeilen hinzu, um die IP-Adresse für den Access Point, den Adressbereich sowie die Domainnamen-Server (auch als DNS bezeichnet) zu definieren:

Bei Tor handelt es sich um eine freie Software, die online Anonymität ermöglicht und den Internetverkehr durch ein anonymes Netz leitet.

```
pi@raspberrypi:~$ nano /etc/dhcp/dhcpd.conf
# range 10.17.224.10 10.17.224.250;
# }
# pool {
#   deny members of "foo";
#   range 10.0.29.10 10.0.29.230;
# }
#}
subnet 192.168.12.0 netmask 255.255.255.0 {
  range 192.168.12.5 192.168.12.50;
  option broadcast-address 192.168.12.255;
  option routers 192.168.12.1;
  default-lease-time 600;
  max-lease-time 7200;
  option domain-name "local";
  option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

```
subnet 192.168.12.0 netmask 255.255.255.0 {
  range 192.168.12.5 192.168.12.50;
  option broadcast-address 192.168.12.255;
  option routers 192.168.12.1;
  default-lease-time 600;
  max-lease-time 7200;
  option domain-name "local";
  option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

Beachten Sie, dass die hier angegebenen Domainnamen-Server von Google bereitgestellt werden. Drücken Sie `Strg+X` zum Speichern und Beenden, folgen Sie danach den Aufforderungen:

```
pi@raspberrypi:~$ nano /etc/dhcp/dhcpd.conf
# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.
```

06 Statische IP zuweisen

Weil die WLAN-Karte des Pi standardmäßig eine dynamische IP-Adresse erhält, müssen wir eine statische IP angeben. Damit kann sie in ihrer Rolle als Tor-Anonymisierer jederzeit verwendet werden.

Öffnen Sie mit `nano` die Datei `/etc/default/isc-dhcp-server`, scrollen Sie zu der Zeile `INTERFACES=""` nach unten und ändern Sie sie in `INTERFACES="wlan0"`. Speichern und schließen Sie dann.

Fehler beheben

Bei diesem Projekt können verschiedene Fehler auftreten. Probleme mit Ihren Lokaleinstellungen sind nur ein Beispiel. Durch Bearbeiten der `ssh_config` können Sie Abhilfe schaffen:

```
sudo nano /etc/ssh/ssh_config
```

Finden Sie die Zeile „`SendEnv LANG LC_*`“ und kommentieren Sie sie mit einem `#` am Anfang aus. Möglicherweise müssen Sie den Pi neu starten, um die Fehler vollständig zu beseitigen.

Welchen Pi Sie nutzen können

Zwar lassen sich auch die älteren, mit einem doppelten USB-Port ausgestatteten Raspberry Pis für dieses Projekt verwenden, doch Sie erzielen die besten Ergebnisse mit den Raspberry-Pi-2- und -3-Modellen. Die Gründe sind einfach: Letztere besitzen vier USB-Ports und ermöglichen so den Anschluss von Peripheriegeräten wie Tastatur und Maus. Beide sind zudem schnell genug, um die benötigten Datenmengen zu verarbeiten. Ein Pi Zero wäre wahrscheinlich nicht so praktisch.

07 Wireless-Adapter einrichten

Nun da der WLAN-Adapter wlan0 eingerichtet ist, müssen wir ihn erst deaktivieren, bevor wir ihn konfigurieren. Dazu verwenden wir den Befehl „sudo ifdown wlan0“. Mit nano öffnen wir die Textdatei /etc/network/interfaces und setzen drei #:

```
# iface wlan0 inet manual
# wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
# iface default inet dhcp
```

Suchen Sie die Zeile „allow-hotplug wlan0“ und fügen Sie ein:

```
iface wlan0 inet static
address 192.168.12.1
netmask 255.255.255.0
```

Mit Strg+X speichern. Dann aktivieren Sie:

```
sudo ifconfig wlan0 192.168.12.1
```

08 Neuen WAP konfigurieren

Haben Sie den WLAN-Zugangspunkt aufgebaut, die Software installiert und dann definiert, ist es an der Zeit, mit der Konfiguration zu beginnen. Erstellen Sie in nano eine neue Datei:

```
sudo nano /etc/hostapd/hostapd.conf
```

Fügen Sie dann Folgendes hinzu:

```
interface=wlan0
ssid=TorHotSpot
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=$Your_Passphrase$
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Sie können Ihre eigene SSID angeben. Sie müssen auch Ihr eigenes sicheres Passwort bereitstellen.

09 Konfigurationsdatei finden

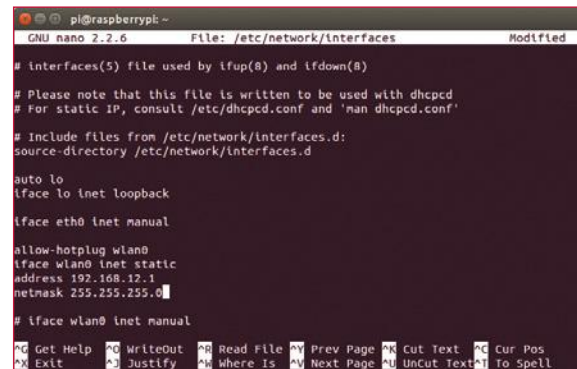
Standardmäßig ist Ihr Raspberry Pi nicht in der Lage, die neue Konfigurationsdatei zu erkennen. Damit sie erkannt werden kann, wenn der drahtlose Access Point bootet, müssen wir in der Datei /etc/default/hostapd darauf hinweisen. Dazu öffnen Sie die Datei in nano und suchen nach der Zeile #DAEMON_CONF="". Löschen Sie das # und fügen Sie folgenden Pfad hinzu:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Drücken Sie Strg+X zum Speichern und Beenden.

10 Konfigurieren der Netzwerkadressübersetzung

Network Address Translation oder NAT muss konfiguriert werden, um mehrere Geräte zuzulassen – Laptops, Smartphones, Tablets, Media-Streamer und mehr. Erst dann kann sich der Pi mit dem drahtlosen Zugangspunkt verbinden und den Traffic über eine einzelne IP weiterleiten. Öffnen Sie /etc/sysctl.conf mit nano und fügen Sie unten hinzu:



```
net.ipv4.ip_forward=1
```

Drücken Sie nun wieder Strg+X zum Speichern und Beenden. Geben Sie dann diese Befehlszeile ein:

```
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

11 Routing-Regeln angeben

Nun müssen Routingregeln hinzugefügt werden. Diese verbinden den Ethernet-Port eth0 und den Raspberry-Pi-Wireless-Port wlan0:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

12 Regeln beim Booten nutzen

Routing-Regeln werden beim Neustart des Pi gelöscht. Geben Sie daher im Terminal ein:

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

Damit wird sichergestellt, dass die Regeln bestehen bleiben. Als Nächstes öffnen Sie /etc/network/interfaces mit nano und fügen diese Zeile am Ende hinzu:

```
up iptables-restore < /etc/iptables.ipv4.nat
```

Speichern und Beenden Sie. Starten Sie den DHCP-Server neu:

```
sudo service isc-dhcp-server restart
```

Nun aktivieren wir den Zugangspunkt:

```
sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

13 Zugangspunkt bereitstellen

Jetzt ist es möglich, eine Verbindung zum Zugangspunkt herzustellen. Einige Einstellungen müssen garantiert werden, beginnend mit den DHCP- und Hostapd-Diensten:

```
sudo service hostapd start
sudo service isc-dhcp-server start
```

Aktualisieren Sie die Initialisierungsskripte (init) mittels:

```
sudo update-rc.d hostapd enable
sudo update-rc.d isc-dhcp-server enable
```




NAT muss konfiguriert werden, sodass Geräten die Verbindung zum Zugangspunkt ermöglicht wird.

Starten Sie mit „sudo shutdown -r“ jetzt neu und wiederholen Sie die Verbindung von Ihrem Mobilgerät.

14 Tor installieren

Nun fungiert der Pi als ein drahtloser Zugangspunkt. Es fehlt lediglich die Anonymisierung. Verbinden Sie sich wieder über SSH und geben Sie „sudo apt-get install tor“ ein, dann öffnen Sie `/etc/tor/torrc` in nano und fügen Folgendes hinzu, um Tor zu anonymisieren:

```
Log notice file /var/log/tor/notices.log
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsSuffixes .onion,.exit
AutomapHostsOnResolve 1
TransPort 9040
TransListenAddress 192.168.12.1
DNSPort 53
DNSListenAddress 192.168.12.1
```

15 SSH-Ausnahme hinzufügen

Bearbeiten Sie die IP-Tabellen so:

```
sudo iptables -F
sudo iptables -t nat -F
```

Im Anschluss ergänzen Sie jetzt eine Ausnahme für den Port 22:

```
sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp
--dport 22 -j REDIRECT --to-ports 22
```

Diese beiden Regeln ermöglichen DNS-Lookups und direkten TCP-Verkehr zum Tor-Port 9040.

```
sudo iptables -t nat -A PREROUTING -i wlan0 -p udp
--dport 53 -j REDIRECT --to-ports 53
sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp
--syn -j REDIRECT --to-ports 9040
```

So speichern Sie alles in der NAT-Datei ab:

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

16 Tor-Dienst aktivieren

Jetzt können Sie den Tor-Service auf Ihrem Wireless Access Point aktivieren:

```
sudo service tor start
```

Bootskripte aktivieren Sie wie folgt:

```
sudo update-rc.d tor enable
```

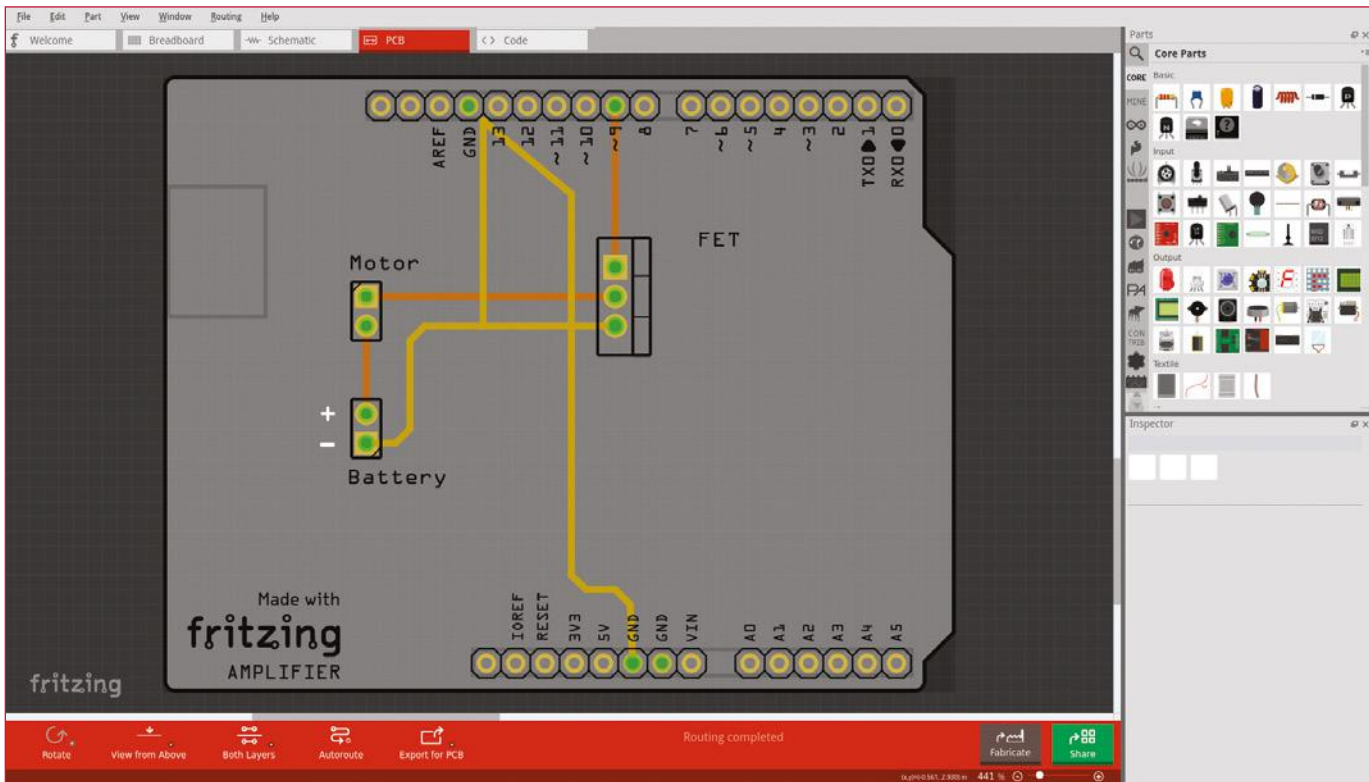
Starten Sie den Pi mit „sudo shutdown -r now“ neu.

17 Tor-Anonymität überprüfen

Haben Sie Ihren Zugangspunkt neu gestartet, wird es Zeit, ein WLAN-Gerät wie ein Smartphone zu verbinden. Die Verbindung sollte klappen. Aber surfen Sie nun auch anonym? Das finden Sie heraus, indem Sie von zwei Geräten auf torproject.org zugreifen und die IP-Adressen vergleichen. Sind sie unterschiedlich, dann bleiben Sie unerkannt!

Oben Sie können sogar das Bluetooth des Pi 3 verwenden, um Ihre Smart-Home-Gadgets zu kontrollieren oder Ihren Web-Traffic über Tor zu übermitteln.

Mit Fritzing eigene Schaltpläne erstellen



Fragten Sie sich schon, wie wir unsere Schaltpläne zeichnen? Mit einem Tool, das Sie ebenfalls verwenden können.

Schaltkreise zusammenzubauen, kann selbst für Profis kompliziert sein. Glücklicherweise gibt es diverse Notationen, die dabei helfen, schnell herauszufinden, wo Drahtenden, Bauelemente und Energiequellen angeschlossen werden sollen. Doch anstatt alles abstrakt mit Papier und Bleistift festzuhalten, bevorzugen Sie vielleicht etwas Realismus in der Abbildung. Das ist besonders für Anfänger nützlich, denn wenn man die Drahtverbindungen auf einem Breadboard (Steckplatine) mit einer Arduino-Schaltung zeigen will, ist es sinnvoll, dem Leser gleich die exakten Positionen auf der Platine und an den Bauteilen erkennbar zu machen. Hier ist es, wo Fritzing ins Spiel kommt.

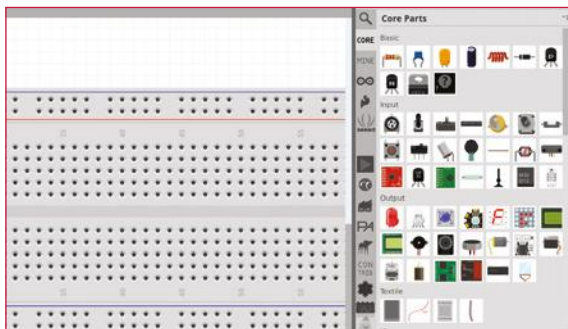
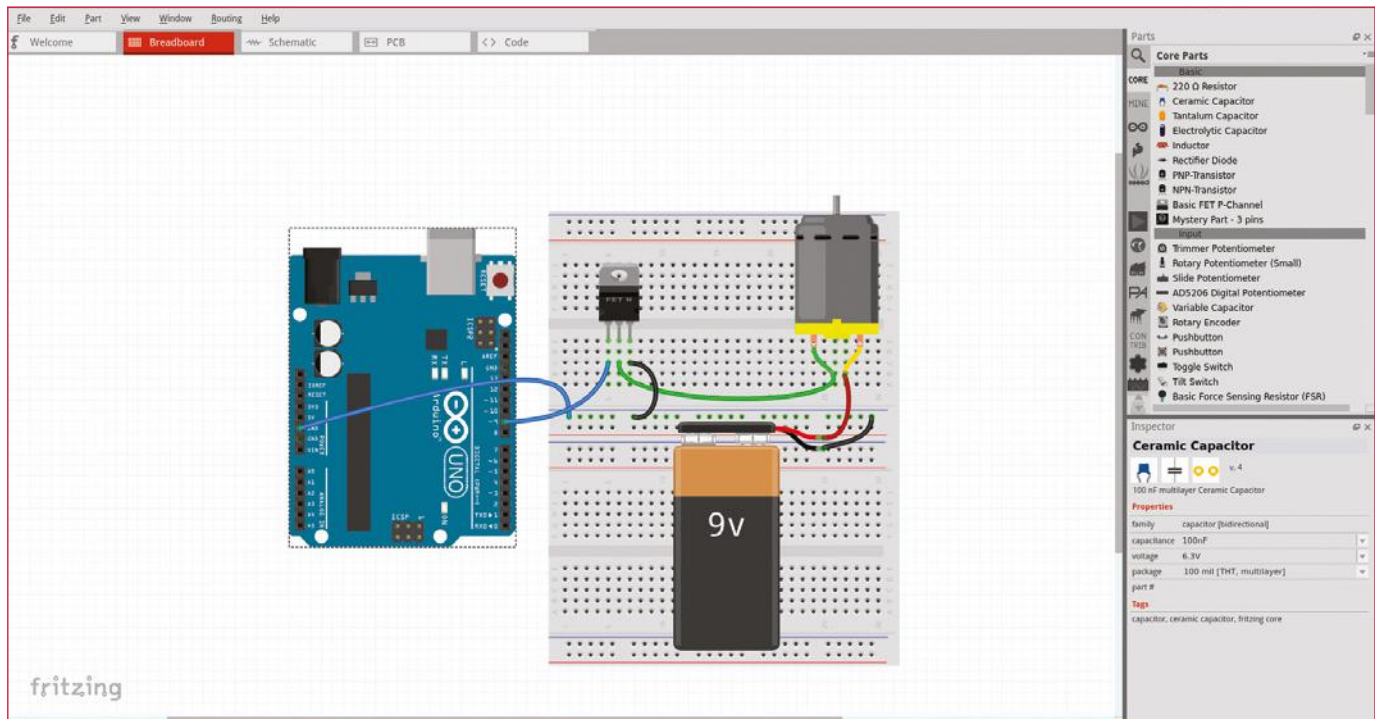
Sie haben über die Jahre vielleicht schon öfter Abbildungen von Schaltungen gesehen. Diese werden oft mit Fritzing erstellt, das Sie ebenfalls für diesen Zweck nutzen können, indem Sie im Tool einfach die Bauelemente an die richtige Stelle ziehen, verbinden, speichern und exportieren. Zusätzlich zur Verbildlichung von Breadboard-Schaltungen können Sie mit Fritzing auch ganz klassische Schaltpläne erstellen.

01 Fritzing installieren

Die schnellste Möglichkeit zur Installation ist der Befehl `sudo apt-get install fritzing` auf Ubuntu oder jeder anderen Debian-basierenden Distribution. Fedora-Nutzer tippen `yum install fritzing` in das Terminal. Falls Sie es wünschen, können Sie Fritzing aber auch als Archivdatei von der Website herunterladen (stellen Sie sicher, dass Sie die 32- oder 64-Bit-Version passend zu Ihrem Betriebssystem auswählen) und den Inhalt dann an die gewünschte Stelle hin entpacken. Sobald Sie das erledigt haben, navigieren Sie per Terminal zum Verzeichnis und starten es mittels Eingabe von `./Fritzing` in die Kommandozeile.

02 Art des Projekts wählen

Nach dem Starten der Anwendung müssen Sie den passenden Projekttyp auswählen. Davon gibt es bei Fritzing drei. Wir starten mit einem Breadboard-Projekt, wählen Sie darum bitte die entsprechende Option aus. Am unteren Rand des Bildschirms sehen Sie Schaltflächen zum Hinzufügen von Notizen oder dem Rotieren der Steckplatine. Rechts gibt es eine Liste diverser Komponenten, organisiert in Kategorien.



03 Bauteile auswählen

Der nächste Schritt ist das Finden der richtigen Bauteile, um Ihren Schaltplan aufzubauen. Der fertige Schaltplan soll dabei später so aufgeräumt wie möglich sein. Erstmal wollen wir aber nur mit Fritzing herumspielen, damit Sie das Programm kennenlernen. In der Komponentenliste oben rechts wählen Sie den passenden Tab aus und ziehen das gewünschte Bauteil von dort in die Mitte des Fensters.

04 Mit der Steckplatine jonglierend

Bevor Sie anfangen, dem Projekt weitere Teile hinzuzufügen, sollten Sie sich Zeit nehmen, Ihre Optionen kennenzulernen. Durch einen Linksklick auf das Breadboard können Sie dieses verschieben, ein Rechtsklick öffnet ein Menü mit Befehlen zum Drehen, Ändern der Sichtbarkeitsebene sowie Sperren von Bauteilen. Das Mausrad dient kombiniert mit dem Halten von Strg als Zoom.

05 Leitungen einfügen

Drahtverbindungen legen Sie an, indem Sie den Endpunkt eines Anschlusses anklicken und ihn an seinen Bestimmungsort ziehen. Die Pins auf der Steckplatine haben Beschriftungen, die erscheinen, wenn Sie die Maus über den zu identifizierenden Pin bewegen. Sie können den Leitungen Kurven und Eckpunkte zuweisen, indem Sie den Draht an einer Stelle greifen

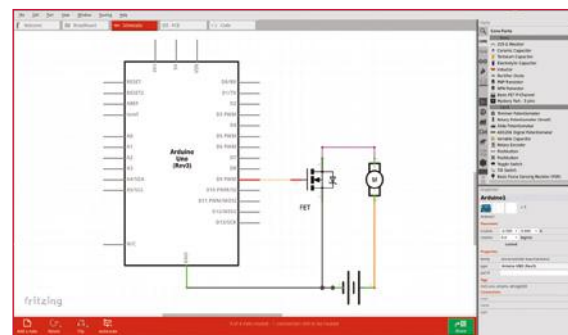
und an die gewünschte Stelle ziehen. Das ist besonders bei aufwendigen Projekten sinnvoll.

06 Komponenten einfügen

Ziehen Sie die benötigten Bauteile in den Bildschirm, um Ihr Projekt weiter aufzubauen. Achten Sie darauf: Fritzing hebt diejenigen Flächen des Breadboards, welche Verbindungen zur Komponente ermöglichen, hervor. Das hilft Ihnen beim korrekten Platzieren der Leitungen, auch wenn es keine perfekte Lösung zum Verbinden der Anschlüsse ist. Falls Ihnen die Verwendung von Breadboards neu ist, kann Ihnen das aber helfen, deren Funktionsweise zu verstehen.

07 Zwischen Ansichten umschalten

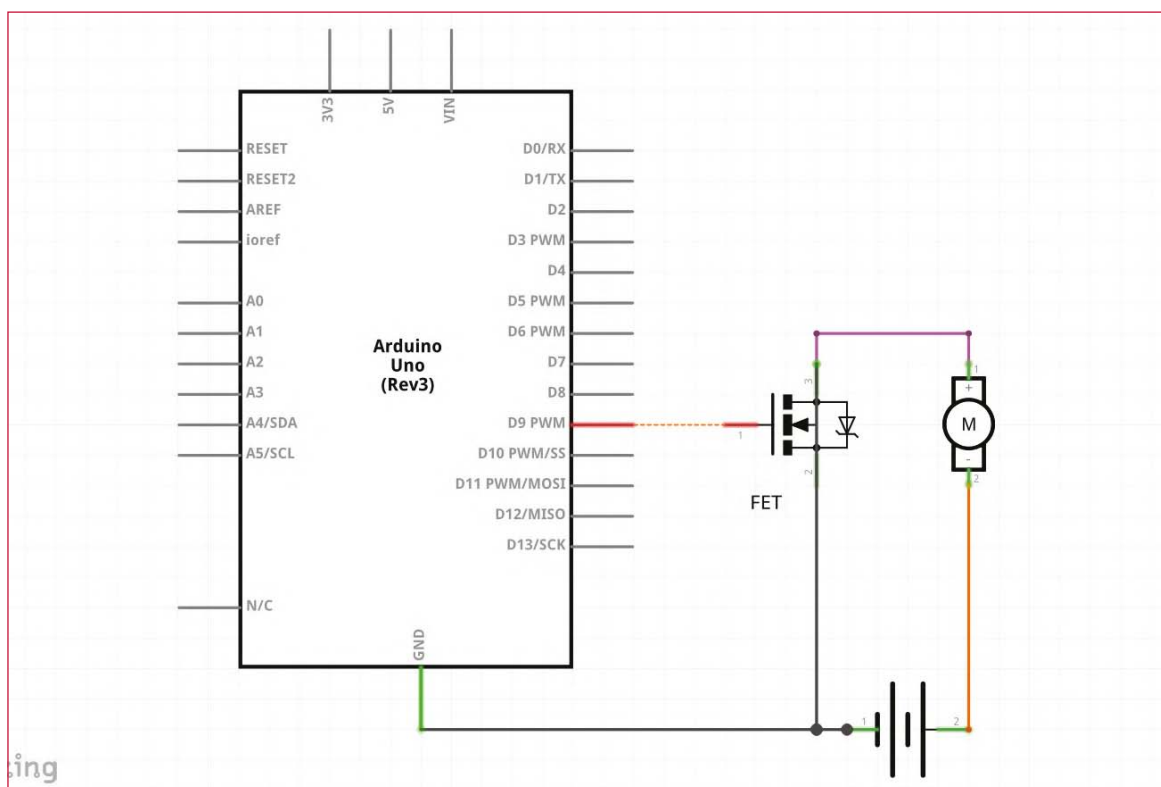
Steckplatinen sind eine einfache Möglichkeit, schnell Schaltungen aufzubauen. Die grafische Repräsentation trifft jedoch nicht jedermanns Geschmack. Glücklicherweise bietet Fritzing aber diverse Ansichten. Am oberen Rand des Fensters sind Ihnen vielleicht die Schaltflächen „PCB“ („Printed Circuit Board“, Leiterplatte) und „Schematic“ aufgefallen. Klicken Sie auf einen dieser Buttons, um die Ansicht umzustellen. Das Projekt, an dem Sie gerade arbeiten, wird dann in einer völlig anderen Weise dargestellt. Das ist vor allem später beim Exportieren nützlich.



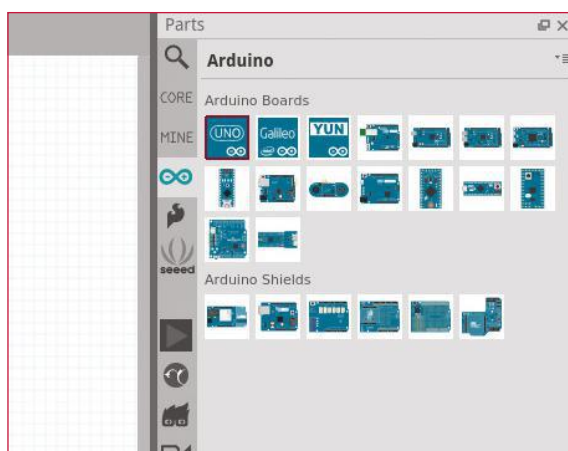
Oben Fritzing bietet eine überwältigende Anzahl verschiedener Bauteile, ebenso wie Platinen wie das Arduino.

Organisieren Sie sich

Sobald Sie Fritzing einige Zeit benutzt haben, werden Sie feststellen, dass es eine gute Idee ist, zunächst einmal alle Bauteile in das Fenster zu ziehen. Dadurch können Sie nach und nach alle Teile herauspicken, ohne viel Zeit für das Umschalten zwischen den Listen zu verwenden.



Rechts Sie können Ihre Schaltung auch in einem klassischen Schaltplan darstellen.



Meine Bauteile

In der Komponentenliste ist der zweite Tab zu Beginn leer. Diese Gruppe mit dem Namen MINE dient dazu, alle Bauteile zu sammeln, die Sie besonders häufig verwenden. Sie füllen diese Liste, indem Sie Komponenten aus anderen Kategorien auf diesen Tab ziehen.

08 Weitere Platinen einfügen

Bislang haben wir nur mit dem Steckbrett alleine herumgespielt. Aber was, wenn Sie zum Beispiel eine Arduino-basierte Schaltung entwickeln möchten? Die Antwort auf diese Frage finden Sie in der Komponentenliste rechts: Klicken Sie auf das Arduino-Symbol (die Plus-minus-Schleife) und ziehen Sie die passende Arduino-Platine in die Hauptansicht. Neben zahlreichen Arduino-Modellen enthält Fritzing noch eine Auswahl von Erweiterungsplatinen.

09 Sich mit den Komponenten vertraut machen

Fritzing enthält dermaßen viele Bauteile, dass man fast zwangsläufig von der Auswahl überfordert ist. Benutzen Sie die Tabs in der linken Spalte der Komponentenliste, um das gesuchte Bauteil zu finden. Sie finden hier Komponenten für Arduino, Intel Edison und Galileo, sowie SparkFun und viele mehr. Wenn Sie die Maus über ein Bauteil bewegen, bekommen Sie in der Detailansicht ausführliche Informationen.

Fritzing hilft beim Erstellen der Schaltpläne – und informiert Sie bei Problemen!

10 In anderen Ansichten arbeiten

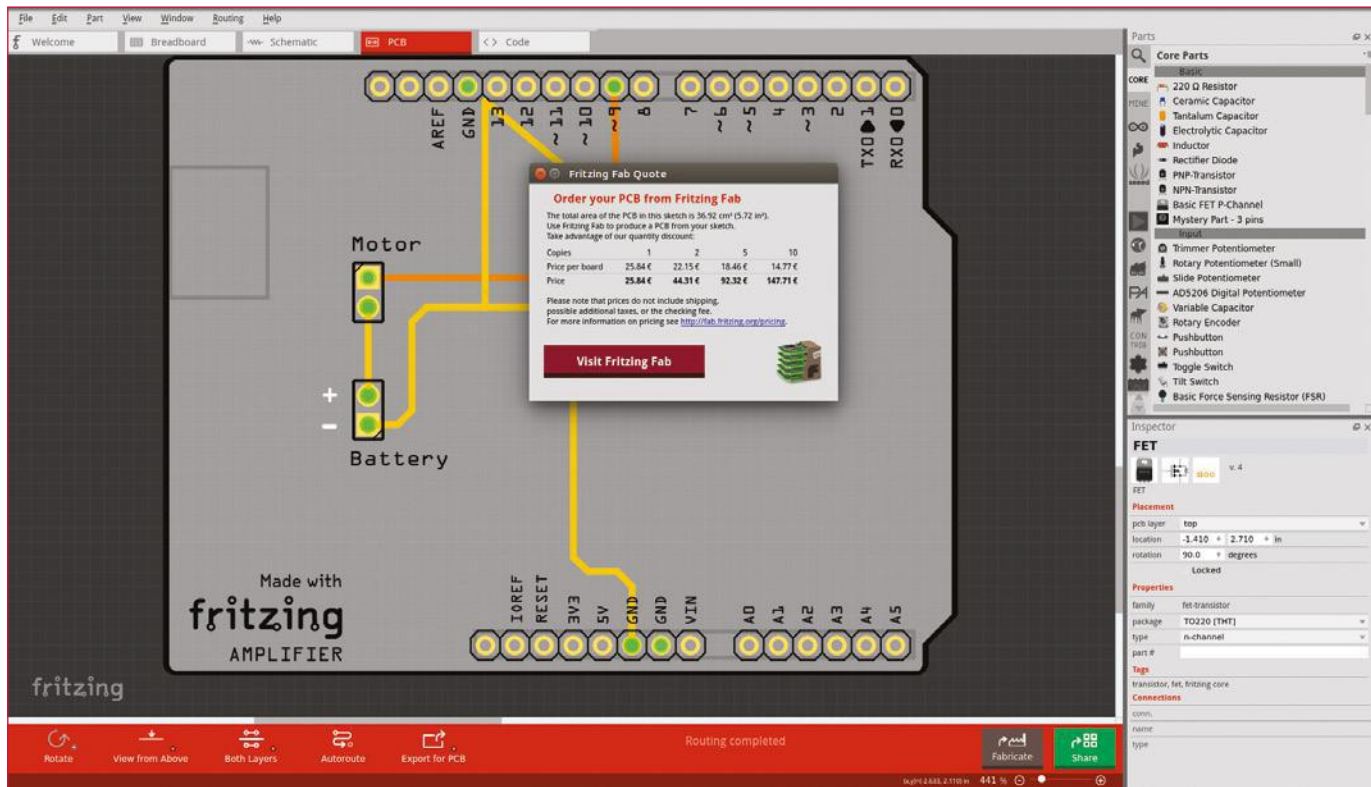
Sobald Sie die standardmäßige Ansicht verlassen, stellen Sie fest, dass die Möglichkeiten, Bauteile zu bewegen, zwar immer noch vorhanden, aber eingeschränkt sind. In der obigen Darstellung, in der das Arduino als Rechteck dargestellt wird, kann dieses ohne weiteres bewegt werden, während die Gruppe von Bauteilen, die auf dem Breadboard verbaut ist, sich nicht verschieben lässt. Es ist einfach nur eine Gruppe von Komponenten verbunden mit Drähten.

11 Mit der Platinenansicht spielen

Ähnlich funktioniert es auch in der Platinenansicht, wenn auch in einer anderen Darstellungsweise: Anstatt dass die Steckplatine und das Arduino als getrennte Einheiten dargestellt werden, sitzt das Breadboard direkt auf dem Arduino. Das Platzieren der Teile ist bei dieser Ansicht schwieriger, dafür erkennt man genau, wo was angeschlossen wird.

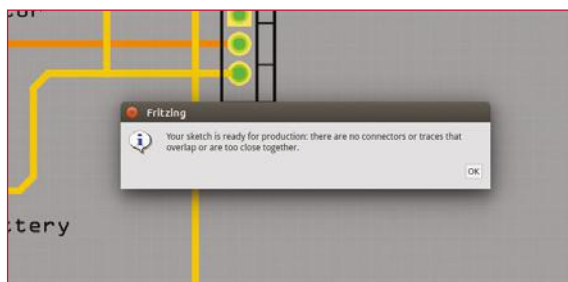
12 Raster verwenden

Es ist empfehlenswert, eigene Projekte übersichtlich zu gestalten. Eine Hilfestellung bietet hier das Raster, das Sie über View > Align to Grid aktivieren und per Show Grid einblenden können. Wie weit die einzelnen Teile bewegt werden können, hängt von der Ansicht und der Zoomstufe ab, die Sie gerade verwenden. Verwenden Sie die Maus zum groben Platzieren und die Pfeil-Tasten zur genauen Justierung.



13 Hilfe durch Autorouting

Während es nicht möglich ist, das Projekt an sich automatisch zu generieren, kann Ihnen Fritzing mit einem Autorouting (dem automatischen Verbinden von Anschlüssen) helfen. Klicken Sie hierfür, während Sie sich in der PCB-Ansicht befinden, auf Routing > Autoroute, und Fritzing wird, soweit möglich, alle Anschlüsse miteinander verbinden. Das gleichzeitige Festhalten der Tasten Strg+Umschalt+A erfüllt denselben Zweck. Nachdem Sie alle Komponenten platziert haben, kann Fritzing Ihnen also einen großen Teil der Arbeit abnehmen.



14 Ist Ihr Projekt bereit?

Fritzing hilft Ihnen nicht nur beim Erstellen der Schaltpläne, es kann Sie auch auf etwaige Probleme hinweisen! Sobald Sie alle Bauteile in die Ansicht gezogen und verdrahtet haben, klicken Sie auf Routing > Design Rules Check, um Ihren Aufbau einer Prüfung zu unterziehen. Sollten Probleme auftauchen, lässt Fritzing Sie das wissen. Die meisten Probleme lösen Sie durch das Verschieben der Bauteile und Leitungen.

15 Stückliste erstellen

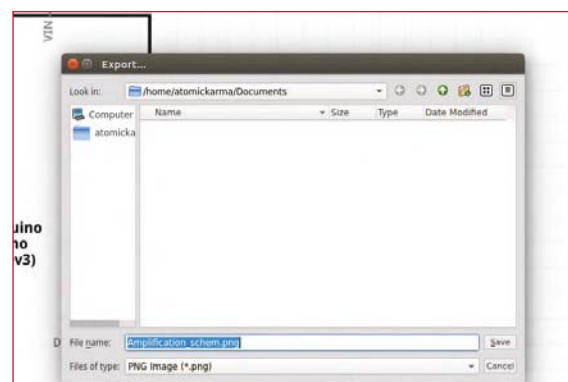
Große Projekte können durchaus das Erfassungsvermögen überfordern – so ist es leicht, auf der Steckplatine unauffällige Komponenten zu übersehen. Hier ist eine Stückliste sehr hilfreich. Unter File > Export gibt es die Option, eine HTML-Liste aller Bauteile zu erstellen, die Sie direkt zum Einkufen verwenden können.

16 Kostenvoranschlag einholen

Ihre Schaltung können Sie auch gleich in eine echte Platine überführen, indem Sie die Dienstleistung Fritzing Fab in Anspruch nehmen. Diesen finden Sie unter Routing > Fritzing Fab Quote. Im folgenden Dialog bekommen Sie alle Informationen darüber, was die Platine bei der Abnahme von einem, zwei, fünf oder zehn Stück kosten würde. Die Preise sind in Euro angegeben. Sie können mit Fritzing also nicht nur Ihre Schaltung designen, sondern sie auch gleich als echte Leiterplatte kaufen.

17 Fritzing-Projekt exportieren

Sobald Sie das Schaltungsdesign vollendet haben, möchten Sie es vielleicht mit anderen teilen. Das erreichen Sie über das Exportwerkzeug im „File“-Menü, wo Sie zwei Punkte zur Verfügung haben: als Bild oder für die Produktion. Wenn Sie Ihre Leiterplatte erstellen wollen, ist Letzteres die richtige Option. Um die Schaltung jedoch als JPG, PNG, SVG oder PDF zu exportieren, nutzen Sie die Möglichkeit „As Image“ und wählen Sie das gewünschte Format. Es lohnt sich im Übrigen wirklich, sich existierende Projekte anzusehen; denn Fritzing kann eine ganze Menge!



Oben Sobald Sie Ihre Leiterplatte entworfen haben, beispielsweise einen neuen Aufsatz für den Raspberry Pi, können Sie den Prototyp direkt aus der Software heraus ordern!

Genauer anpassen

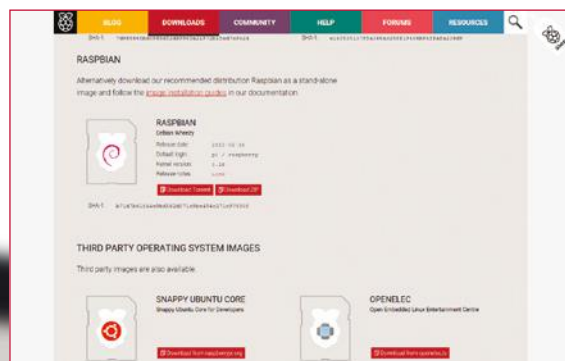
Möchten Sie die Position oder Farbe eines Bauteils verändern, benutzen Sie dazu das „Inspector Panel“. Ein ausgewähltes Bauelement können Sie mithilfe der dort im „Placement“-Bereich vorhandenen Dreiecke verschieben. Die Farbe der Drähte ändern Sie, indem Sie eine Leitung rechtsklicken und dann die Farbbox unter „Properties“ anpassen.

Pi 2 als Desktop-PC

Dank seiner gesteigerten Leistung eignet sich der Raspberry Pi 2 als Desktop-Rechner.

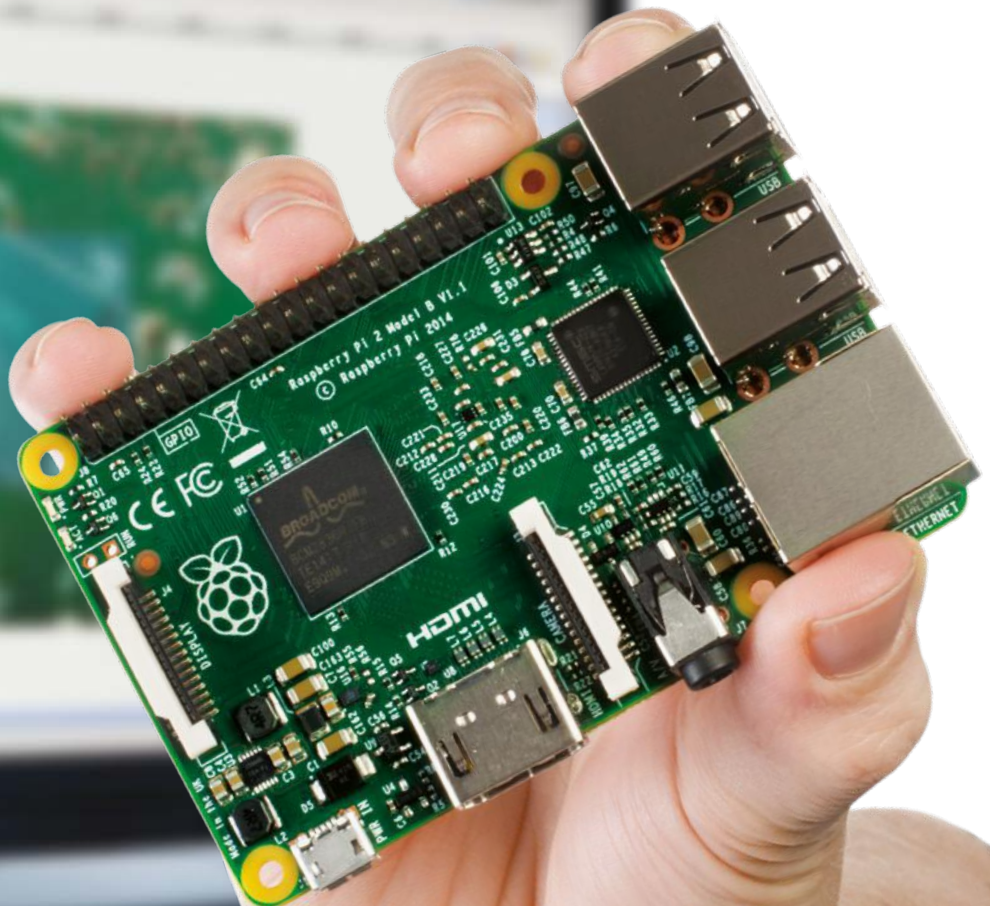
Der Raspberry Pi 2 Modell B hat deutlich mehr Leistung unter der Haube als seine Vorgängermodelle. Mehr CPU-Kerne und mehr RAM machen ihn sechsmal schneller, was wirklich eine beeindruckende Steigerung darstellt. Und Sie können sicher sein, dass Sie diesen Unterschied in der Praxis auch wirklich spüren werden. Diese Power ermöglicht es Ihnen, etwas zu tun, das mit dem ursprünglichen Raspberry Pi unmöglich war: Sie können ihn als Desktop-PC-Ersatz einsetzen. Die Komponenten als solche waren zwar schon immer vorhanden, allerdings mangelte es dem Pi an Leistung, um eine flüssige Arbeitsweise garantieren zu können. Doch mit der neuen Hardware gehören diese Einschränkungen der Vergangenheit an.

01 Raspbian beschaffen
Für unseren Desktop-PC benutzen wir das Betriebssystem Raspbian. Nicht nur, dass es einfach zu bekommen (raspberrypi.org/downloads) und zu benutzen ist, es wird außerdem von der Raspberry Pi Foundation und der Community unterstützt. Es ist sehr flexibel und die beste Wahl für einen Desktop-Rechner.



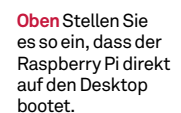
Was Sie brauchen

- Raspberry Pi 2 (oder 3)
- Raspbian
raspberrypi.org/downloads
- Tastatur
- Maus
- Wireless-Dongle
- Monitor
- Gehäuse
- USB-Hub (mit eigener Stromversorgung)





```
$ sudo dd bs=1M if=raspbian.img of=/dev/[Ort der SD-Karte]
```



03 Beim ersten Hochfahren gibt es bezüglich des Setups noch einiges zu tun. Das Wichtigste ist, „Enable Boot to Desktop“ zu aktivieren und dann die Installation so zu erweitern, dass sie die gesamte SD-Karte ausfüllt. Nachdem Sie damit fertig sind, booten Sie neu.

04 Sie werden in eine frische Version von Raspbian mit neuerem Interface und neueren Standard-Apps hineinbooten. Jetzt können Sie im Grunde schon normal arbeiten. Es gibt aber noch ein paar Dinge, die man machen kann, um die Desktop-Erfahrung zu verbessern.

05 Als Erstes werden wir das System auf den neuesten Stand bringen. Öffnen Sie das Terminal (die Kommandozeilenumgebung) vom Menü aus und tippen Sie Folgendes ein:

06 Während wir updaten, empfiehlt es sich, auch ein Firmware-Update durchzuführen. Im Terminal aktivieren Sie die Update-Software mit:

07 Zu diesem Zeitpunkt möchten Sie den Pi vielleicht noch ein bisschen anpassen. Rufen Sie den ursprünglichen Konfigurations-Screen im Terminal auf. Tippen Sie dafür:

8 Von hier aus können Sie einige Optionen aktivieren, die Sie vielleicht in Zukunft noch brauchen. Ein erster sinnvoller Schritt ist die Aktivierung des Pi-Kamera-Treibers. Und Sie können auch direkt in die Entwicklungsumgebung Scratch booten lassen, wenn Spieleprogrammierung Ihr Ding ist.

09 Peripherie

Die Einrichtung des Betriebssystems auf dem Raspberry Pi ist nur eine von vielen Aufgaben. Wir müssen uns auch um die Peripheriegeräte wie Tastatur und Maus kümmern.



10 Eingabegeräte

Standard-USB-Keyboards und -Mäuse eignen sich am besten für diese Aufgabe, viel besser nämlich als die kabellosen Maus/Tastatur-Kombos, die unter Pi-Nutzern beliebt sind. Versuchen Sie aber nicht, Ports zu sparen, indem Sie ein Keyboard mit eingebautem USB-Hub nehmen, denn der Pi kann solch einen USB-Hub nicht mit Strom versorgen.

11 Anzeige auf dem Monitor

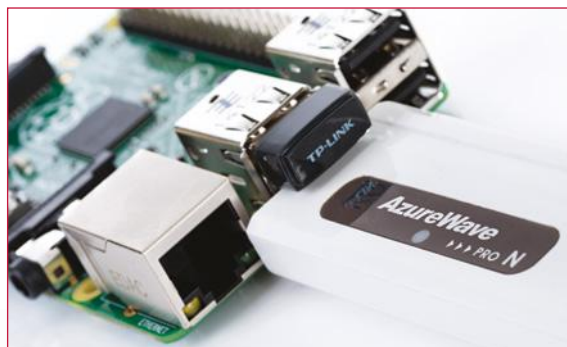
Der Pi kann maximal 1080p ausgeben, was einer Monitorauflösung von 1.920 x 1.080 Pixeln entspricht. Die Ausgabe erfolgt per HDMI. Falls Sie noch ein älteres Display haben, können Sie sich mit einem HDMI-auf-VGA- oder -DVI-Adapter behelfen.

Unten Sie können einen anständigen Minecraft-Server für sich und ein paar Freunde aufsetzen.



12 Ein Gehäuse zum Schutz

Der Pi ist ziemlich robust, doch unzerstörbar ist er auch wieder nicht. Während bei vielen Projekten schneller Zugriff auf die Komponenten vonnöten ist, brauchen wir dies bei einem Desktop-Pi nicht. Wir haben uns hier für ein sicheres Pimoroni-Pibow-Gehäuse entschieden.



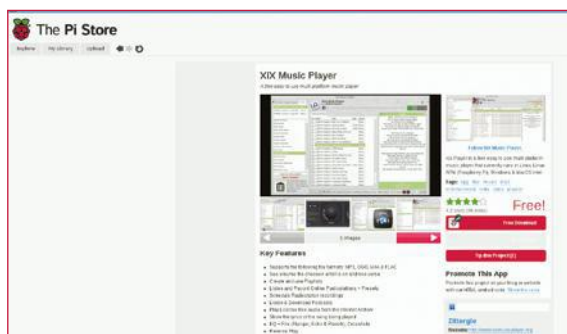
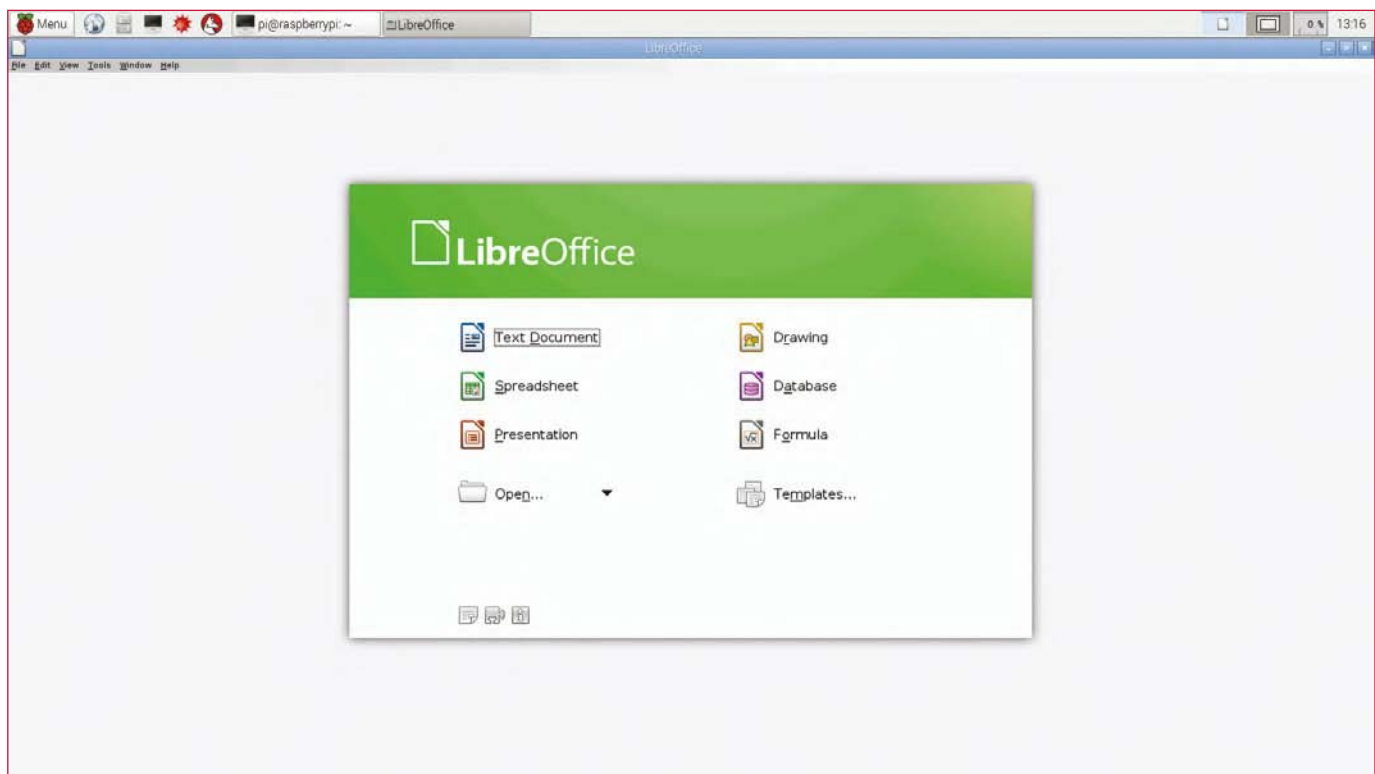
13 Kabellos ins Internet

Während einige Leute kein Problem mit Kabelverbindungen haben, können sich andere diesen Luxus jedoch nicht leisten und müssen auf Wireless-Dongles zurückgreifen. Allerdings arbeitet nicht jedes Modell mit dem Pi zusammen. Unter elinux.org/RPi_USB_Wi-Fi_Adapters finden Sie eine Liste mit kompatiblen Geräten.



14 Und was noch?

Unser Standard-Desktop-PC-Setup ist komplett, ein USB-Anschluss ist noch frei. Sie können diesen Single-Port für USB-Sticks oder portable Speicherlösungen verwenden, oder Sie investieren in einen USB-Hub mit eigener Stromversorgung, um das Anschlussangebot zu erweitern. Eine weitere gute Investition ist ein 2A-Netzteil, damit Sie immer für alles genügend Power haben.



LibreOffice ist – als führende Open-Source-Office-Suite – leicht für Raspbian zu bekommen.

15 Zusätzliche Software hinzufügen

Noch ist unser Raspberry-Pi-Desktop-PC nicht ganz perfekt. Wir müssen noch ein paar Programme hinzufügen, damit er sich wie ein echter Desktop anfühlt. Nachdem wir bereits einen Browser und einige Standards installiert haben, sollten wir als Nächstes eine Office-Suite hinzufügen.

16 Mit LibreOffice arbeiten

LibreOffice gehört nicht zur Standardinstallation, aber als führende Open-Source-Office-Suite ist sie in der Regel leicht für Raspbian zu bekommen. Öffnen Sie das Terminal und installieren Sie die Software mit:

```
$ sudo apt-get install libreoffice
```

17 GIMP für Fotos

Das ist ein großer Brocken. Der erste Pi hatte schon mit LibreOffice seine liebe Not, aber GIMP konnte man gleich ganz vergessen. Auf dem Raspberry Pi 2 läuft GIMP aber ganz ordentlich. Zum Installieren tippen Sie:

```
$ sudo apt-get install gimp
```



18 XiX für Musik

Möchten Sie während der Arbeit Musik hören, empfiehlt sich dafür die Software XiX. Sie bekommen diese auf xixmusicplayer.org kostenlos.

19 Pi für Desktop

Jetzt ist alles eingerichtet und Sie können den Pi als adäquates Desktop-System benutzen. Das Ganze funktioniert natürlich auch mit dem Raspberry Pi 3.

Oben Mit dem Pi 2 ist es möglich, mit einer vollwertigen Office-Suite zu arbeiten, ohne dass der Rechner in die Knie geht.





Dank Pi: Sound in jedem Raum

Kombinieren Sie den Raspberry Pi und den Logitech Media Server zu einem synchronisierten Netzwerk-Audiosystem.

Was Sie brauchen

- **1 Raspberry Pi für jeden Raum** (einer von diesen kann gleichzeitig als Server fungieren, oder es gibt einen separaten Server)
- **1 USB-Soundkarte für jeden Pi** (die Behringer UCA-202 ist eine gute Wahl)
- **Verstärker und Lautsprecher zur Verbindung mit dem Pi**
- **1 USB-WLAN-Adapter für jeden Pi im WLAN** (Edimax EW-7811UN)

In diesem Artikel werden wir eine Software namens Logitech Media Server (früher Squeezebox Server) und den Squeezebox-Client einsetzen, um ein Netzwerk-Lautsprechersystem einzurichten. Der Media Server kann Audio zu jedem Squeezebox-Client streamen und gegebenenfalls die Wiedergabe auf allen Clients synchronisieren. Nachdem Logitech seine Squeezebox-Produktreihe eingestellt hatte, spendierte es der zugehörigen Software eine Open-Source-Lizenz.

Der Media Server wird entweder über eine Webseite oder über diverse Handy-Apps angesteuert. Leider hatten wir Probleme mit dem Spotify-Plugin (genauer gesagt, das inoffizielle von Triode) beim Schreiben des Artikels. Die Suche funktio-

niert nicht, was sehr wahrscheinlich daran liegt, dass die Spotify-API ein Update erfahren hat und das Plugin noch nicht. Stattdessen kann der Server ein zugewiesenes Musikverzeichnis auslesen. Laden Sie hierfür Musik auf die SD-Karte, verwenden Sie einen USB-Stick oder schließen Sie eine Festplatte an. Da es schon 64-GB-Flashlaufwerke für knapp 15 Euro gibt, sollten Sie keine Platzprobleme bekommen. Wir richten einen Samba-Server ein, damit Sie Musik über das Netzwerk kopieren können, sodass Sie durch das Herausnehmen der Karte nicht die Wiedergabe unterbrechen müssen.

Wir empfehlen einen Raspberry Pi 2 oder 3 als Streaming-Server, jedoch sollte ein älterer Pi oder ein Pi Zero als Client ebenfalls ausreichen.

01 SD-Karten flashen

Wir verwenden Raspbian Jessie. Hier benutzten wir die Lite-Version, da sie kleiner und mit weniger Software ausgestattet ist. Es spielt jedoch keine Rolle, ob Sie die Vollversion oder die von NOOBS installierte Version einsetzen. Sie benötigen eine SD-Karte für jeden verwendeten Pi. In unserem Fall wäre das eine für den Server/Client und eine für den Nur-Client.

02 Rechner nacheinander einschalten

Es ist eine gute Idee, die Pis der Reihe nach einzuschalten, um herauszufinden, welcher davon welche IP besitzt. Loggen Sie sich in Ihren Router/DHCP-Server ein, um die jeweilige Adresse zu sehen, ansonsten lässt sich der Rechner auch per SSH unter dem Namen „raspberrypi“ finden. Alternativ können Sie `nmap` oder `arp -na | grep -i b8:27:eb` verwenden, um den Pi im Netzwerk zu ermitteln.

03 Loggen Sie sich ein

Loggen Sie sich in jeden Pi mit `ssh mailto:pi@172.17.173.58` „pi@172.17.173.58“ ein und ersetzen Sie die IP durch die tatsächliche IP Ihres Pi. Verwenden Sie das Passwort „raspberrypi“. Ändern Sie dann den Hostnamen in eine aussagekräftige Bezeichnung wie Wohnzimmer oder Musicserver: `echo ,musicserver' | sudo tee /etc/hostname`.

Geben Sie anschließend `sudo raspi-config` ein, um das Dateisystem zu erweitern. Wählen Sie danach Beenden aus und starten Sie neu. Hiernach sollten Sie in der Lage sein, sich mit dem eingestellten Namen in Ihren Pi einzuloggen, und wissen, um welchen es sich handelt.

04 WLAN einrichten

Falls Sie einen WLAN-Adapter verwenden, verbinden Sie diesen mit dem Pi und bearbeiten Sie `/etc/wpa_supplicant/wpa_supplicant.conf` mit `sudo` und einem Editor Ihrer Wahl. Beispiel: `sudo nano /etc/wpa_supplicant/wpa_supplicant.conf` – fügen Sie jetzt folgende Zeilen hinzu:

```
network={
    ssid="MYSSID"
    psk="passphrase"
}
```

Wenn Sie nun neu starten und das LAN-Kabel entfernen, sollte sich Ihr Pi über WLAN verbinden.

05 Sound card setup

Verwenden Sie eine USB-Soundkarte, dann sollte deren Priorität über jener der internen Soundkarte liegen (welche immer `card0` ist):

```
pcm.!default {
    type hw
    card 1
}
```

```
ctl.!default {
    type hw
    card 1
}
```

Stellen Sie mit `alsamixer` die Lautstärke auf 100 %, da die Karte als Line-out angesteuert wird.

06 Installieren Sie squeezelite auf jedem Client

Nachdem Sie WLAN und die USB-Soundkarten auf den Clients (und auf dem Server, wenn dieser gleichzeitig als Client fungiert) eingerichtet haben, wird es Zeit, Squeezelite zu installieren:

```
apt-get update
apt-get upgrade
apt-get install squeezelite
```

Mehr braucht man nicht, um einen Client einzurichten. Der Squeezelite-Client startet sich selbst und wird über den Server angesteuert, also muss dieser jetzt eingerichtet werden.

07 Musikspeicher vorbereiten

Wir speichern die Musik in `/mnt/music` auf dem Server-Pi. Möchten Sie eine USB-Festplatte oder ein Flash-Laufwerk verwenden, so müssen Sie diese formatieren und in diesem Verzeichnis mounten:

```
sudo parted /dev/sda
(parted) mktable msdos
(parted) mkpart primary ext2 0% 100%
(parted) quit
sudo mkfs.ext4 /dev/sda1
sudo mkdir /mnt/music
```

Unten Ein Raspberry Pi, der mit dem Logitech Media Server läuft, kann zu einem vollwertigen Streaming-Server umfunktioniert werden, der Musik und Internetradio wiedergibt.





```
sudo mount /dev/sda1 /mnt/music
```

```
sudo editor /etc/fstab, add:
/dev/sda1 /mnt/music ext4 defaults,noatime 0 2
```

08 Samba Share einrichten

Samba gibt Dateien und Drucker frei. Es ist es die beste Wahl, da es von Windows, macOS und Linux unterstützt wird. Hiermit können Sie per Drag & Drop einfach Dateien von jedem PC auf Ihren Streaming-Server kopieren.

```
sudo apt-get update
sudo apt-get install samba
```

```
sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.
original
sudo vim /etc/samba/smb.conf:
[global]
workgroup = WORKGROUP
server string = Music Server
security = user
guest account = nobody
map to guest = Bad User
```

```
[music]
path = /mnt/music
public = yes
browsable = yes
only guest = yes
writable = yes
```

```
sudo systemctl restart smbd
sudo chown nobody:nogroup /mnt/music/
```

Geben Sie /mnt/music Lese-, Schreib- und Zugriffsrechte für alle:

```
sudo chmod -R 777 /mnt/music/
```

09 Musik auf den Server kopieren

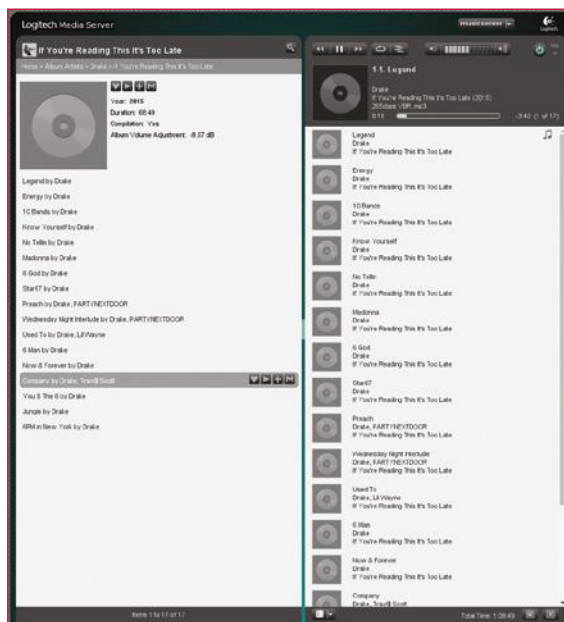
Kopieren Sie jetzt die Musik auf den Server. Bei Windows, macOS oder Linux sollte dieser im Netzwerkbereich zu sehen sein. Bei Linux fanden wir ihn unter Verwendung des Thunar-Dateibrowsers unter Network > Windows Network > WORKGROUP > MUSICSERVER. Jetzt kopieren wir die Musik unserer Wahl auf den Server. Eigentlich sollten alle Formate funktionieren, wir verwendeten aber 320k-MP3s mit Tags, sodass diese richtig angezeigt werden, nachdem sie der Logitech Media Server geladen hat.

10 Logitech Media Server konfigurieren

Nun da wir die Musik auf den Server kopiert haben, wird es Zeit, den Logitech Media Server einzurichten. Der Server läuft über Port 9000, daher mussten wir mit einem Browser über <http://musicserver:9000/> den Einrichtungsassistenten aufrufen. Abhängig von Ihrer Netzwerkkonfiguration brauchen Sie vielleicht die IP-Adresse Ihres Servers. Außerdem müssen Sie einen Account bei mysqueezebox.com erstellen, bevor es weitergeht. Navigieren Sie im Anschluss an den Ort, wo Ihre Musik gespeichert ist. Dieser sollte /mnt/music sein. Klicken Sie auf Weiter und legen Sie /mnt/music auch als Verzeichnis für Playlists fest, selbst wenn dort keine zu finden sind. Klicken Sie nun auf Beenden. Sollten Sie jemals Ihre Bibliothek neu scannen lassen müssen, geht dies in den Einstellungen (in der unteren rechten Ecke); klicken Sie auf den Rescan-Knopf.

11 Musik abspielen

Die Musik über den Browser abzuspielen, ist leicht. Links ist Ihre Musikbibliothek und auf der rechten Seite die aktuelle Playlist. In der unteren rechten Ecke können Sie zwischen verschiedenen Playern auswählen, bei uns music-server und musicclient. Am besten nennen Sie die Player aber nach deren jeweiliger Position in der Wohnung. Weiterhin gibt es auch einen Synchronisieren-Knopf, wodurch Sie verschiedene Räume zusammengruppieren können. Alles, was dann abgespielt wird, läuft gleichzeitig in diesen Räumen. Außerdem können Sie die Lautstärke für jeden Raum festlegen. Das bedeutet, dass Sie die Lautstärke an Ihrem Verstärker einigermaßen hoch einstellen und dann die tatsächliche Lautstärke mit Ihrem Smartphone oder Browser steuern können.



Bricht das Tonsignal ab?

Wenn der Ton hakt, liegt das sehr wahrscheinlich daran, dass in Ihrem Netzwerk nicht genügend Bandbreite zur Verfügung steht. Die einfachste Lösung ist dann, den Pi mit einem Netzwerkkabel zu verbinden, statt WLAN zu verwenden. Möchten Sie in Ihrem Haus keine Kabel verlegen, probieren Sie einen anderen WLAN-Adapter aus oder setzen Sie Powerline-Adapter ein; zwei Stück kriegen Sie schon für knapp 40 Euro. Kommerzielle Lösungen wie Sonos-Lautsprecher verwenden ihr eigenes Netzwerk über der Standardfrequenz von 2,4 GHz, sodass Ihr Netzwerk nicht überlagert wird. Das Netzwerk auf 5 GHz umzuschalten, ist eine weitere Option.

12 Musik vom Smartphone steuern

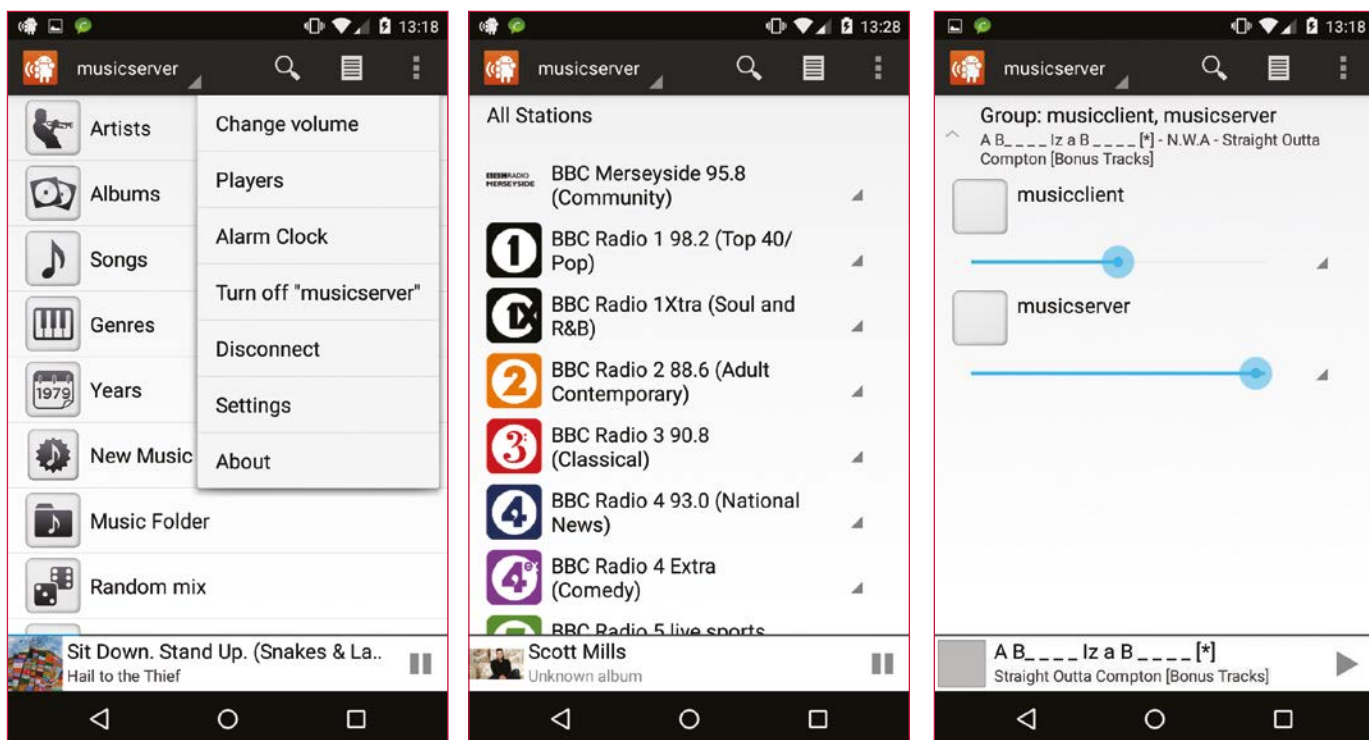
Unter Android empfehlen wir die Squeezer-App, welche Sie mit dem Suchbegriff „Squeezebox“ im Play Store finden. Eine ähnliche App gibt es auch für Apple-Geräte. Die App sollte den Server automatisch finden und sich mit diesem verbinden. Dort finden Sie eine Dropdown-Liste, mit der Sie den Player auswählen können. Die Lautstärketasten Ihres Telefons steuern die Lautstärke der Wiedergabe.

13 Radio hören

Der Logitech Media Server spielt auch Internetradio ab. Es stehen sehr viele Radiostationen – sortiert nach Genre, Land und weiteren Kriterien – zur Auswahl. Diese können sowohl über das Smartphone als auch über den Browser abgespielt werden.

14 Player per Smartphone gruppieren

Um Player über das Telefon zu gruppieren, drücken Sie auf die Menütaste oben rechts und wählen Sie Players aus. Markieren Sie nun die Player, die Sie gruppieren möchten, und klicken Sie auf Synchronisieren. Wählen Sie die Form der Gruppierung aus. Um die Synchronisierung wieder aufzuheben, verfahren Sie entsprechend, wählen aber „No grouping“ aus. Außerdem können Sie die Lautstärke für jeden Raum in diesem Menü festlegen.

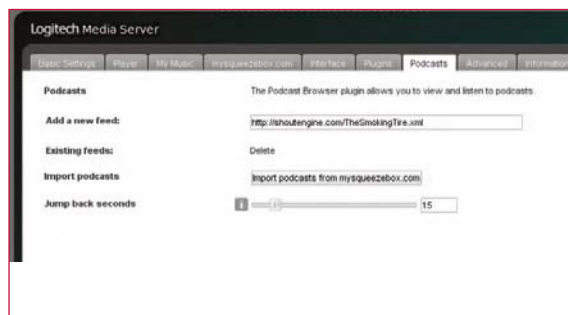


15 Podcasts konfigurieren

Wenn Sie Podcasts hören möchten, müssen Sie erst das Podcast-Plugin bei den Einstellungen im Logitech Media Server aktivieren. Wählen Sie das Plugin aus und klicken Sie auf Anwenden. Der Server muss hiernach neu gestartet werden:

| `sudo systemctl restart logitechmediaserver`

Das Interface braucht ein paar Sekunden, bis es wieder ange-steuert werden kann.

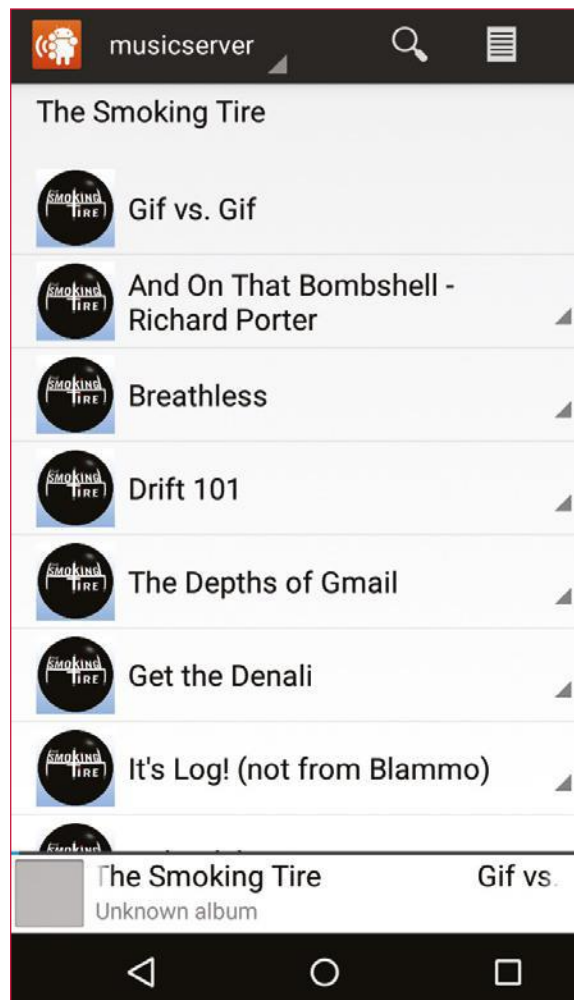


16 RSS-Feeds hinzufügen

Um einen Podcast hören zu können, braucht man des-sen RSS-Feed. Um einen Feed hinzuzufügen, gehen Sie auf die Einstellungsseite des Logitech Media Servers und dort auf den Podcasts-Reiter. Geben Sie die URL des RSS-Feeds des Pod-casts ein.

17 Wiedergabe der Podcasts überprüfen

Podcasts sind unter My Apps > Podcasts zu finden und funktionieren über das Webinterface und die Smartphone-App. Wie man hier sehen kann, werden die Podcasts nach Erschei-nungsdatum sortiert auf Ihrem Bildschirm aufgelistet. Klicken Sie einfach auf das Gewünschte und drücken Sie Play.





Den Media Server automatisieren

Es gibt eine Python-Bibliothek für den Logitech Media Server (github.com/jinglemansweep/PyLMS), mit dem Sie verschiedene Aufgaben automatisieren können. Schauen Sie sich den Code unten an, um eine Anwendung zu schreiben, die Ihr Smartphone anpingt und die Player anhält, sobald Ihr Telefon den Netzwerkbereich (als Hinweis darauf, dass Sie nicht mehr zu Hause sind) verlässt. Wenn Sie zurückkehren, startet die Anwendung die Wiedergabe erneut.

Skript starten

Es ist am sinnvollsten, dieses Skript auf dem Server laufen zu lassen:

```
sudo apt-get install python-pip
sudo pip install pylms
mkdir automation
cd automation
touch music_suspender.py
chmod +x music_suspender.py
editor music_suspender.py
Add code
```

Jetzt lassen Sie das Skript beim Booten starten:

```
sudo editor /etc/systemd/system/musicsuspender.service:
[Unit]
Description=Suspend Music

[Service]
Type=oneshot
ExecStart=/home/pi/automation/music_suspender.py

[Install]
WantedBy=multi-user.target
```

Starten Sie dann den Dienst und machen Sie einen Reboot, um diesen zu testen:

```
sudo systemctl daemon-reload
sudo systemctl enable musicsuspender.service
sudo reboot
```

Die Kosten

Angenommen, Sie haben schon einen oder zwei alte Raspberry Pis herumliegen, ist dies ein relativ kostengünstiges Projekt – es gibt eine Reihe guter und preiswerter T-Amps ab 60 Euro. Suchen Sie einfach nach „T-Amp“ bei Amazon, und Sie werden fündig. Diese Geräte funktionieren so lange gut, wie Sie nicht einen EQ vorschalten, und sind laut genug, dass sich Ihre Nachbarn beschweren, weit bevor der Verstärker verzerrt. Zusätzlich brauchen Sie kleine Standlautsprecher, eine USB-Soundkarte, Lautsprecherkabel und Cinchkabel, um die Soundkarte mit dem Verstärker zu verbinden.

Der Preis der Boxen hängt von deren Qualität ab. Alternativ können Sie sich 2.0- oder 2.1-Lautsprecher mit einem eingebauten Verstärker kaufen. Preislich bekommen Sie schon welche ab 50 Euro, was bedeutet, dass Sie nur noch einen Pi mit Netzwerkverbindung und eine Soundkarte brauchen.

Vollständiger Code

```
#!/usr/bin/env python2
```

```
from pylms.server import Server
from pylms.player import Player
import time
import os
```

Import Der Import benötigt Bibliotheken. Die ersten Imports sind die Server- und Player-Klassen von PyLMS. Time wird für Sleep verwendet, sodass die while-Schleife die CPU nicht vollständig auslastet. Mit „os“ wird der Ping-Befehl aufgerufen.

```
class MusicSuspender:
    def __init__(self, phone_ip):
        self.server = Server()
```

Class Erstellen Sie eine Klasse für die Anwendung. Verwenden Sie die IP Ihres Smartphones (diese muss statisch zugewiesen sein) für den Ping und um sich mit dem Server zu verbinden (läuft auf derselben Maschine). Es wird erwartet, bis die Verbindung aufgebaut wurde, für den Fall, dass das Skript vor dem Server startet. Außerdem überwacht eine Variable den derzeitigen Zustand, sodass nicht immerzu Start- oder Stopfbefehle gesendet werden, solange sich nichts ändert.

```
connected = False
while not connected:
    try:
        self.server.connect()
        connected = True
    except:
        pass
```

```
self.phone_ip = phone_ip
self.prev_state = "playing"
```

```
def stop(self):
    print "Stop"
    if self.prev_state == "stopped":
        return
```

```
for p in self.server.get_players():
    p.pause()
```

```
self.prev_state = "stopped"
```

```
def play(self):
    print "Play"
    if self.prev_state == "playing":
        return
```

```
for p in self.server.get_players():
    p.play()
```

```
self.prev_state = "playing"
```

Ping Die Ping-Funktion ruft den Ping-Befehl mit „-c 1“ auf, um nur einen Ping zu senden. Außerdem wird dieser zu /dev/null gesendet, sodass es keine Ausgabe in der Konsole gibt. Wird der Ping mit 0 zurückgegeben, war dieser erfolgreich, daher kommt der Vergleich mit 0 am Ende.

```
def ping(self):
    return os.system("ping -c 1 {} > /dev/null".format(
        self.phone_ip)) == 0
```

```
def loop(self):
    while True:
        if self.ping():
            self.play()
        else:
            self.stop()

        # Sleep for 5 seconds
        time.sleep(5)
```

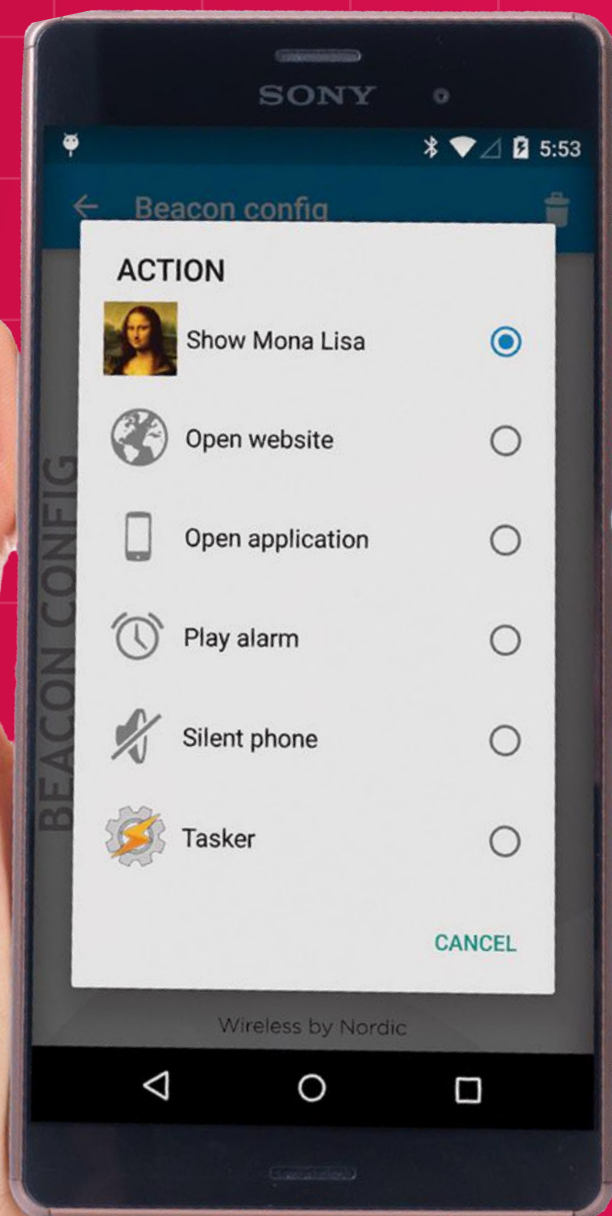
Schleife Erstellen Sie eine Instanz für die MusicSuspender-Klasse mit der IP Ihres Smartphones als Argument. Rufen Sie die Schleifenfunktion auf, welche unendlich lang läuft.

```
if __name__ == "__main__":
    ms = MusicSuspender('172.17.173.6')
    ms.loop()
```

Hacks

114 50 Wege, um den Pi zu hacken

- Luftfeuchtigkeit messen
- Melodien im 8-Bit-Sound
- Minecraft-Fotoautomat



- 122 Zehn geniale Pi-Upgrades
- 128 Roboter steuern
- 132 Selbstfahrendes RC-Auto
- 134 Motion Tracking per Pi
- 136 Erstellen Sie einen Pi-Verbund
- 140 VPN-Zugang per Raspberry Pi
- 144 Beleuchtung mit Pi und Amazon Echo
- 148 Home-Verzeichnis des Pi verschlüsseln
- 152 Nächtliche Fotos mit der Kamera Pi NoIR
- 156 Mit dem Pi Ihr Telefon finden

Geschmeidig: Wir bitten Alexa darum, das Mote darum zu bitten, das LED-Licht auf Blau zu stellen

144

Beleuchtung per Sprachbefehl steuern

152

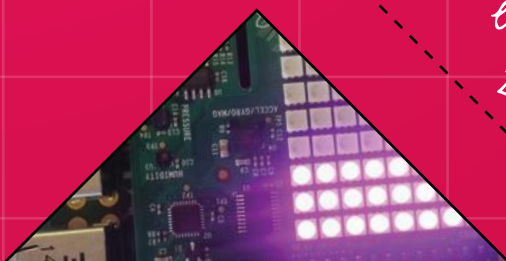
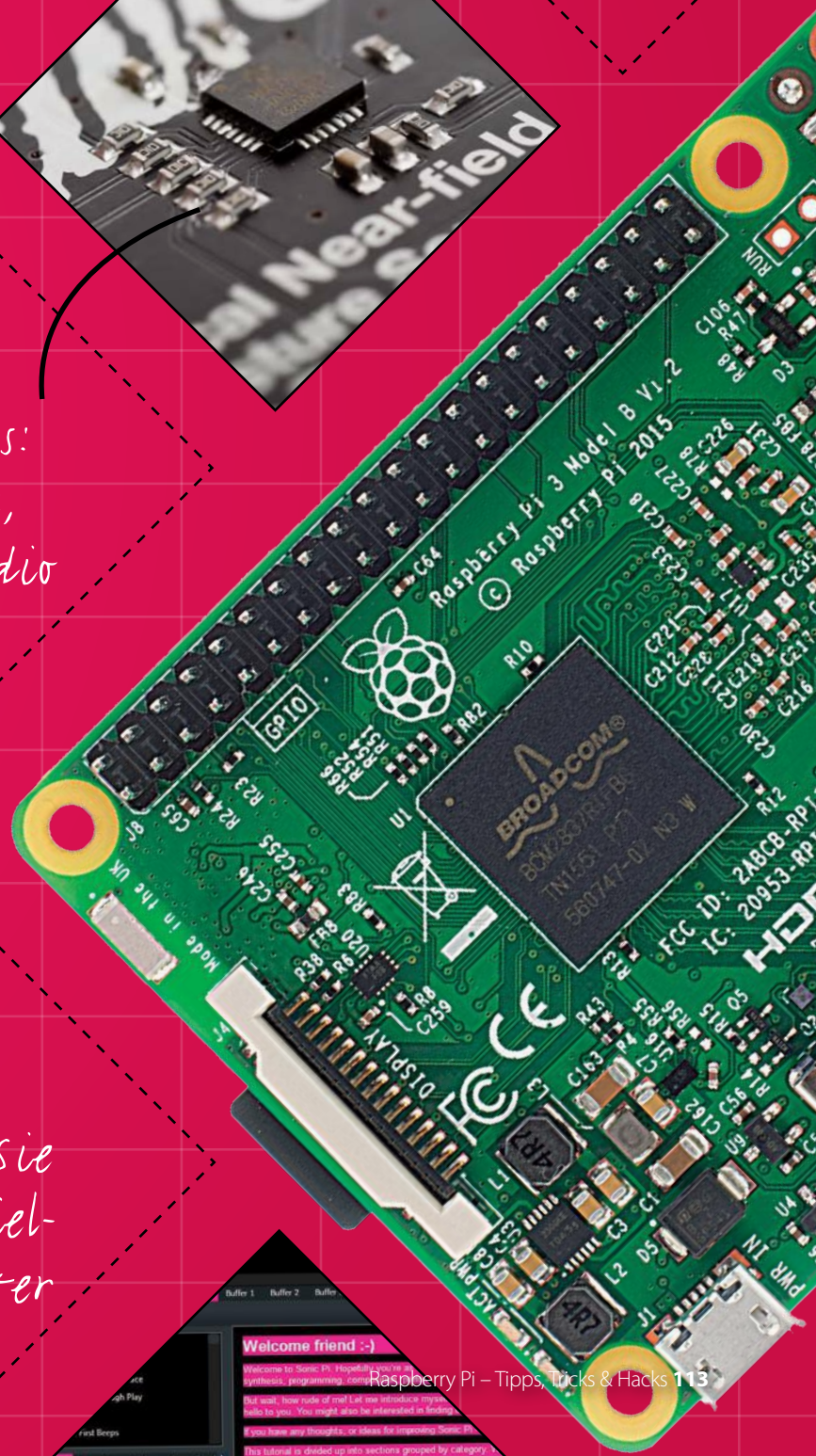
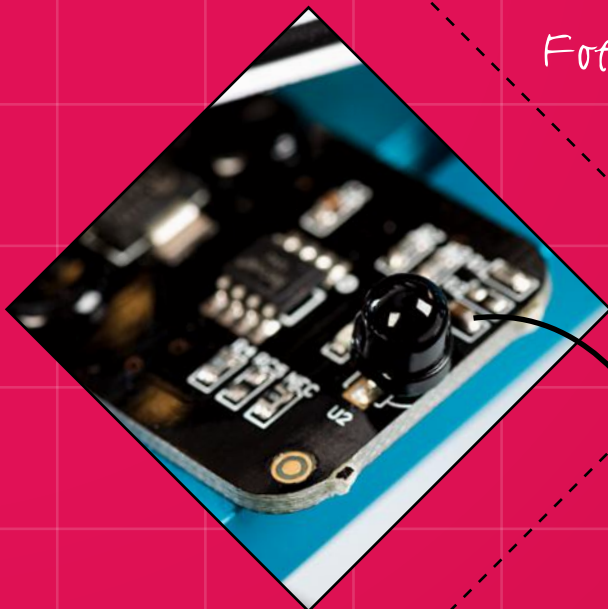
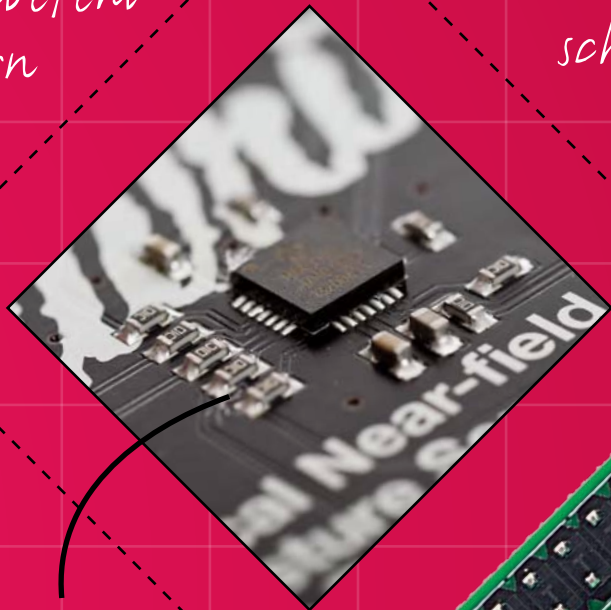
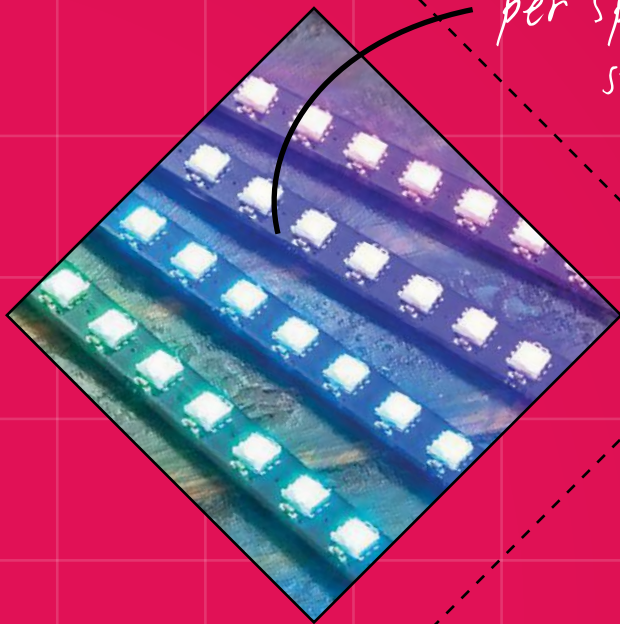
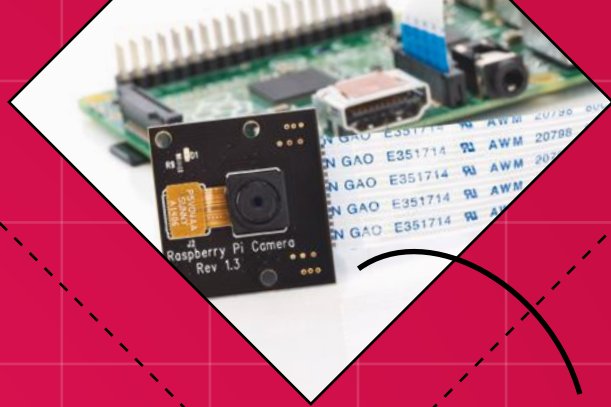
Ertappen sie Nachtschwärmer

122

Upgrades: Energie, Foto, Audio

128

Hacken sie einen spielzeugroboter





50

— WEGE, UM —

IHREN PI ZU HACKEN

Fünzig Wege, um mehr aus Ihrer Raspberry Pi
Hard- und Software herauszuholen

Sie haben einen Raspberry Pi und ihn erfolgreich eingerichtet, greifen auf den Desktop zu, gehen mit ihm online und laden die diversen vorinstallierten Programme.

In diesem Tutorial finden Sie 50 Skills, Codes und Hacks zum Ausprobieren mit Ihrem Pi, welche zwei Softwareteile und ein zusätzliches Hardwareteil umfassen. Zunächst geht es um den Raspberry Pi SenseHAT, der ursprünglich zur Internationalen Raumstation geschickt wurde, um Codes aus- und Experimente durchzuführen. Diese Zusatzplatine kostet um die 40 Euro und hat mehrere Sensoren, u. a. für Neigung, Druck, Feuchtigkeit und Temperatur sowie ein LED-Matrix-Display. Arbeiten Sie sich ein wenig in den notwendigen Programmiercode ein und Sie können schon bald einen eige-

nen Echtzeit-Feuchtigkeitssensor kreieren.

Als nächstes lernen Sie, wie Sie Musik mit dem Sonic Pi kodieren und Ihre eigenen Musikstücke komponieren. Sie können diese Klänge entweder von Grund auf neu erstellen oder Starterkits als Grundlage nehmen. Mit "live loop coding" lässt sich Musik, wie bei einer Live-Aufführung, auch in Echtzeit coden.

Dann lernen Sie alles über *Minecraft*-Hacks, um Ihr Spiel und die Fähigkeiten Ihrer Spielfigur anzupassen. Beamen Sie sich über große Entfernungen, wenn Sie auf einen bestimmten Block steigen oder entdecken Sie, wie Sie mit nur einem Mausklick Häuser jeder Größe erstellen. Wir zeigen Ihnen auch, was Sie alles mit einer in die *Minecraft*-Umgebung integrierten Pi Camera in der Spielwelt anstellen können.



Fünf wichtige Dinge zu Beginn

01 Ihren Pi updaten

Prüfen Sie vor einem Projekt, dass Betriebssystem und Software Ihres Pi aktuell sind. Öffnen Sie LX terminal:

```
sudo apt-get update
sudo apt-get upgrade
```

02 Software installieren und löschen

Online kann Ihr Raspberry Pi auf eine Vielzahl von Software und Programmen zugreifen. Viele davon sind in einer öffentlichen Repository gespeichert. Wenn Sie den Namen der Software kennen, die Sie suchen, etwa den Chrome-Browser, dann geben Sie ein:

```
sudo apt-get chrome
```

Zum Löschen unerwünschter Software nutzen Sie:

```
sudo apt-get purge chrome
```

Suche nach Software mit einem Schlüsselwort:

```
sudo apt-search cache chrome
```

03 Ihre IP-Adresse anzeigen

Um Ihre IP-Adresse zu finden, nutzen Sie ein einfaches Skript, das sie im Konsolenfenster ausgibt. Das geschieht nach dem Booten, noch bevor Ihre Anmeldeinformationen eingegeben werden. Starten Sie LX terminal und geben Sie ein:

```
sudo nano /etc/rc.local
```

Fügen Sie das folgende Skript zur Datei hinzu und speichern Sie:

```
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
printf "My IP address is %s\n" "$_IP"
```

04 Von Ihrem Notebook aus auf die Kommandozeile zugreifen

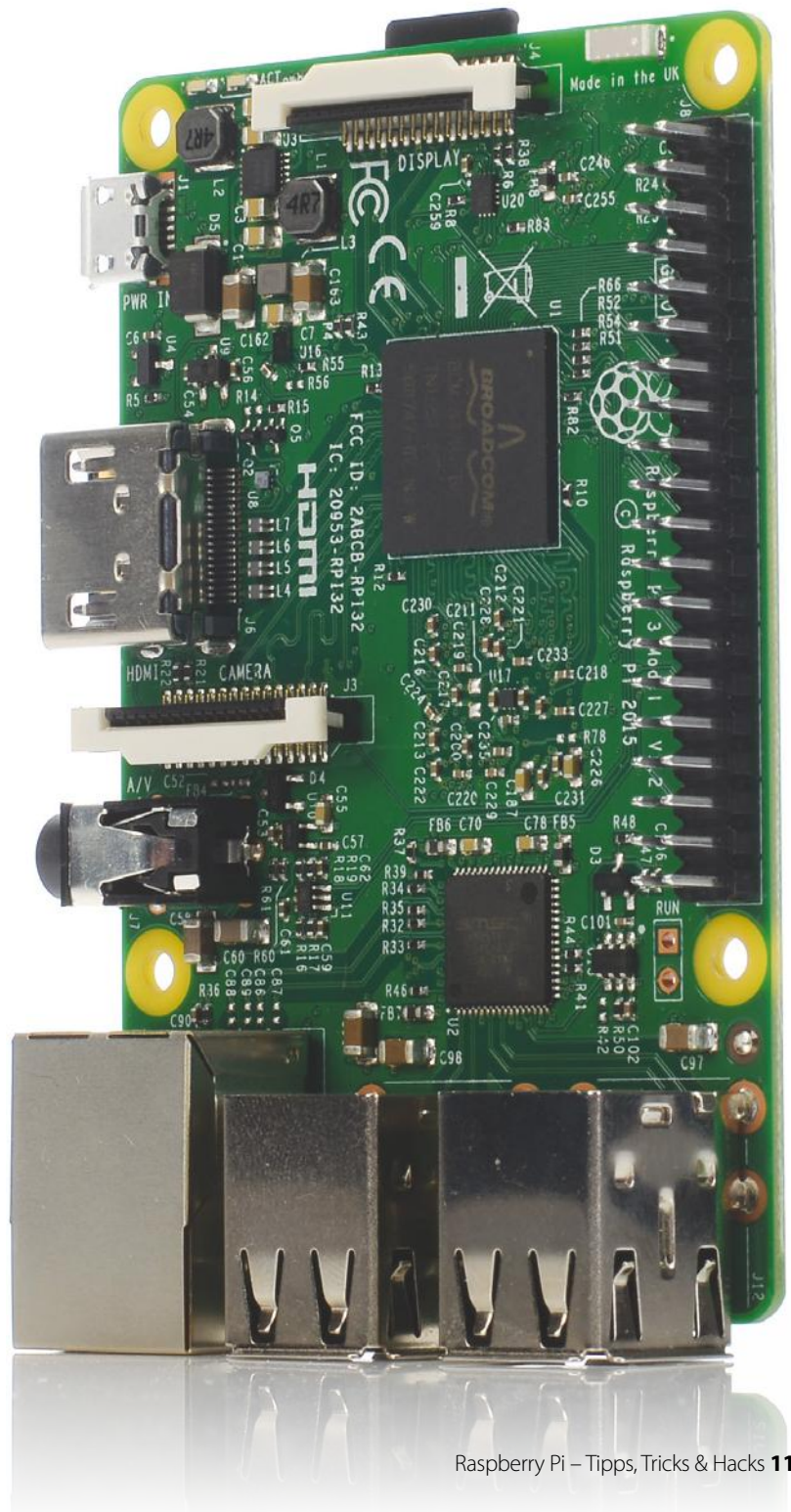
Manchmal haben Sie keinen Zugriff auf einen Bildschirm oder Ihr Pi ist vielleicht in ein Projekt eingebunden und Sie haben keinen Zugriff darauf. Nutzen Sie stattdessen Ihr Notebook. Installieren Sie zunächst ein Programm namens Putty: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Starten Sie Putty, geben Sie die IP-Adresse Ihres Pi sowie Ihren Benutzernamen und Ihr Passwort ein. Nun können Sie direkt auf das Terminalfenster Ihres Pi zugreifen.

05 Raspberry Pi 7"-Touch-Bildschirm

Sollten Sie einen Bildschirm benötigen, schauen Sie sich den offiziellen Pi Touchscreen an. Dieses 800 x 480-Display wird mittels Adapterplatine die Strom und Signalumwandlung sowie 10-Finger-Touch und eine Bildschirmstatur unterstützen. Nur zwei Verbindungen zum Pi werden benötigt: Strom aus dem GPIO-Port des Pi und ein Flachbandkabel, das mit dem DSI-Port verbunden ist. Das Display ist in mehreren Farben erhältlich.

Um Ihre IP-Adresse herauszufinden, lassen Sie sie mit einem einfachen Skript nach dem Booten im Konsolenfenster einblenden.



Lernen Sie den Raspberry Pi SenseHAT kennen

Hacken Sie die Raspberry Pi SenseHAT-Platine, die der britische Astronaut Major Tim Peake auf der Internationalen Raumstation verwendet hat.

- 06 Der Countdown hat begonnen**
Um einen einfachen Countdown von 10 bis 0 auf der LED-Matrix anzuzeigen, wandeln Sie die Zahlenwerte in einen String um. Öffnen Sie Python 3 und fügen Sie den folgenden Code ein:

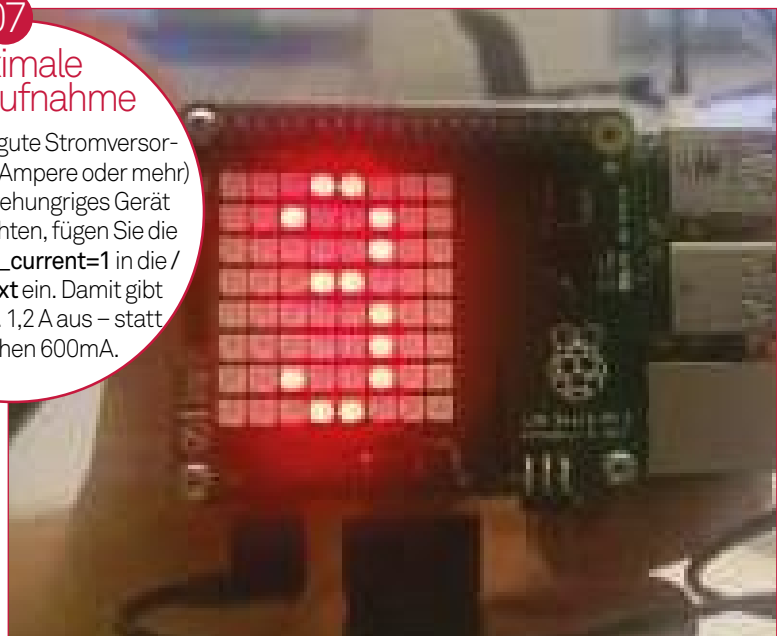
```
import time
from sense_hat import SenseHat
for i in reversed(range(0,10)):
    sense.show_letter(str(i))
    time.sleep(1)
```

- 08 Text scrollen**
Geben Sie im Python-Editor den Code ein, speichern und starten Sie ihn, um Ihre Nachricht über die SenseHAT LEDs zu scrollen. Passen Sie die Farbe des Textes und die Durchlaufzeit an, indem Sie Folgendes hinzufügen: `text_colour=[255, 0, 0]` (Einstellung des RGB-Farbwertes) und `scroll_speed=(0.05)`

```
from sense_hat import Sense HAT
sense = SenseHat()
sense.show_message("Raspberry Pi",
    text_colour=[255, 0, 0])
```

07 Maximale Stromaufnahme

Wenn Sie eine gute Stromversorgung haben (>2 Ampere oder mehr) und ein amperehungriges Gerät verbinden möchten, fügen Sie die Zeile `max_usb_current=1` in die `/boot/config.txt` ein. Damit gibt der USB max. 1,2 A aus – statt der üblichen 600mA.



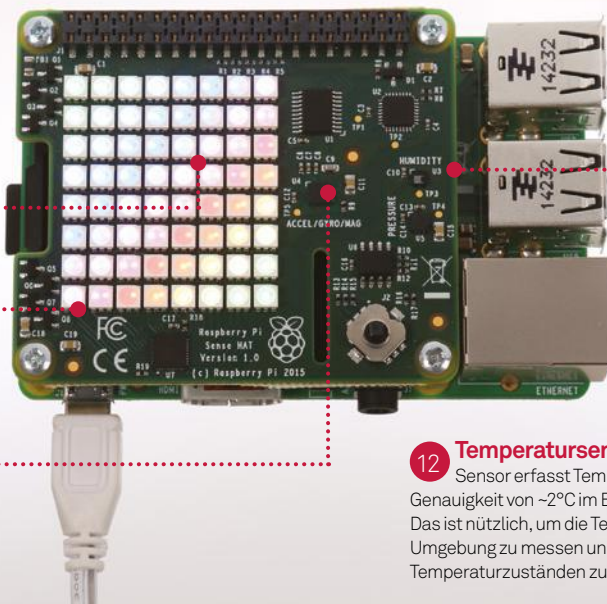
Oben Bauen Sie einen einfachen Countdownzähler oder (umgekehrt) einen Timer.

Der SenseHAT

- 09 Mehrfarbige 8x8 LED-Matrix**
64 vollständig anpassbare mehrfarbige LEDs, um die Daten der Sensoren, einfache Animationen und Spiele oder scrollende Textnachrichten anzuzeigen.

- 10 Magnetometer** Arbeitet wie ein Kompass, der Magnetfeldstärken erkennt. Durch Auslesen können magnetische Felder erkannt und Kompasspunkte relativ zu Nord gefunden werden.

- 11 Beschleunigungsmesser**
Misst die lineare Beschleunigung, etwa nach oben und unten. Diese Daten können verwendet werden, um Bewegungen zu erkennen und beispielsweise einen Controller für ein Spiel zu erstellen.



- 12 Temperatursensor** Der eingebaute Sensor erfasst Temperaturen mit einer Genauigkeit von -2°C im Bereich $0-65^{\circ}\text{C}$. Das ist nützlich, um die Temperatur Ihrer Umgebung zu messen und Veränderungen von Temperaturzuständen zu erkennen.



13 Ein Bild anzeigen

Mit der LED-Matrix können einfache Bilder erzeugt werden. Weisen Sie mittels RGB-Werten die Farben zu und erstellen Sie dann ein einfaches Muster des Bildes, z. B. ein Fragezeichen:

```
from sense_hat import SenseHat
sense = SenseHat()
X = [255, 0, 0] # Red
O = [255, 255, 255] # White
```

```
question_mark = [
    0, 0, 0, X, X, 0, 0,
    0, 0, X, 0, 0, X, 0,
    0, 0, 0, 0, 0, X, 0,
    0, 0, 0, 0, X, 0, 0,
    0, 0, 0, X, 0, 0, 0,
    0, 0, 0, X, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, X, 0, 0, 0,
    ]
sense.set_pixels(question_mark)
```

14 Wie feucht ist es?

Feuchtigkeit in Echtzeit auf der LED-Matrix anzeigen

► Schritt eins: Setup

Dieses Programm misst die Feuchtigkeit und zeigt das Ergebnis auf dem LED-Display. Importieren Sie zuerst die SenseHat-Bibliothek und erstellen dann zwei Variablen; je eine für die LED-Farbe bei ein und aus:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
on_pix = [255,0,0]
off_pix = [0,0,0]
```

► Schritt zwei: Feuchte auslesen

Lesen Sie die Feuchtigkeit aus und runden Sie sie auf eine Dezimalstelle. Erstellen Sie dann eine Liste namens "leds = []", um die Zahl der LEDs zu speichern, die eingeschaltet werden müssen. Da es nur 64 LEDs gibt, teilen Sie die Feuchte von 100 durch 64, um ein Verhältnis zurückzugeben.

```
while True:
    hum = sense.humidity
    hum = round(hum,1)
    if hum > 100:
        hum = 100
    leds = []
    ratio = 64 / 100.0
```

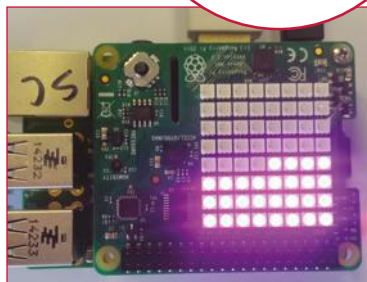
► Schritt drei: LEDs ein oder aus

Nun müssen Sie die Feuchte auslesen, um die Anzahl der einzuschaltenden LEDs zu berechnen. Das macht der Code `on_count = int(ratio*hum)`. Die Zahl der ausgeschalteten LEDs beträgt `64 - die Zahl der eingeschalteten`:

```
on_count = int(ratio*hum)
off_count = 64 - on_count
```

15

Mit der `sudo raspi-config` lassen sich etwa die Kamera aktivieren, der Pi übertakten oder auch die Boot-Optionen ändern.



► Step four: LED-Liste erstellen

Nun haben Sie die Gesamtzahl der LEDs, die eingeschaltet werden müssen. Erweitern Sie die Werte mit der in Schritt 2 erstellten Liste. Machen Sie das Gleiche für die LEDs, die ausgeschaltet werden müssen. Dann stellen Sie die Pixel auf der LED-Matrix ein:

```
leds.extend([on_pix]*on_count)
leds.extend([off_pix]*off_count)
sense.set_pixels(leds)
```

Starten Sie das Programm und beobachten Sie das Display. Pusten Sie auf den Feuchtigkeitssensor und schauen Sie, was passiert.

20

Mit den drei Farbwerten werden die Mengen an Rot, Grün und Blau festgelegt, wobei 255 der Maximalwert ist. Weiß ist beispielsweise [255, 255, 255].

Essenzielle SenseHAT-Messungen

Ihre Umgebung mit diesen SenseHAT-Sensoren vermessen

16 Unter Druck

Der atmosphärische Druck ist die Kraft, die Luft auf eine Oberfläche ausübt, in diesem Fall den SenseHAT. Er lässt sich mit dem Drucksensor auslesen. Je höher die Messung ausfällt, desto mehr Kraft wird ausgeübt:

```
from sense_hat import SenseHat
sense = SenseHat()
pressure = sense.get_pressure()
print("Pressure: %s Millibars" % pressure)
```

17 Temperaturmessung vornehmen

Der SenseHAT hat einen Wärmesensor, der die Temperatur auslesen und zurückgeben kann. Der Sensor befindet sich zwar ziemlich nah an der CPU (Restwärme!), ist aber dennoch recht genau:

```
from sense_hat import SenseHat
sense = SenseHat()
temp = sense.get_temperature()
print("Temperature: %s C" % temp)
```

18 Wo ist Norden?

Der Magnetometer liefert Ihre Position in Bezug auf Norden. Prüfen Sie Ihre Position mit einer while-Schleife und drehen Sie den senseHAT, um die Änderung zu sehen:

```
from sense_hat import SenseHat
sense = SenseHat()
import sleep
while True:
    north = sense.get_compass()
    print("North: %s" % north)
    time.sleep(1)
```

19 Beschleunigungsmesser

Mit ihm wird gemessen, wie sich Ihr SenseHAT bewegt und neigt, z. B. ein Projekt wie Apollo-Soyuz, in dem man ein 3D-Modell der Sojus-Rakete drehen kann (www.github.com/astr-pi/apollo-soyuz).

```
from sense_hat import SenseHat
sense = SenseHat()
while True:
    accel_only = sense.get_accelerometer()
    print("p: {pitch}, r: {roll}, y: {yaw}".format(**accel_only))
```

21 Feuchtigkeit überprüfen

Dieser einfache Code liefert die Feuchtigkeit der Umgebung. Pusten Sie auf den Sensor, um es zu testen:

```
from sense_hat import SenseHat
sense = SenseHat()
humidity = sense.get_humidity()
print("Humidity: %s %%rH" % humidity)
```

Musik mit dem Sonic Pi programmieren

Ein Synthesizer für Live-Code, der zahlreiche Samples, Noten und Klänge enthält

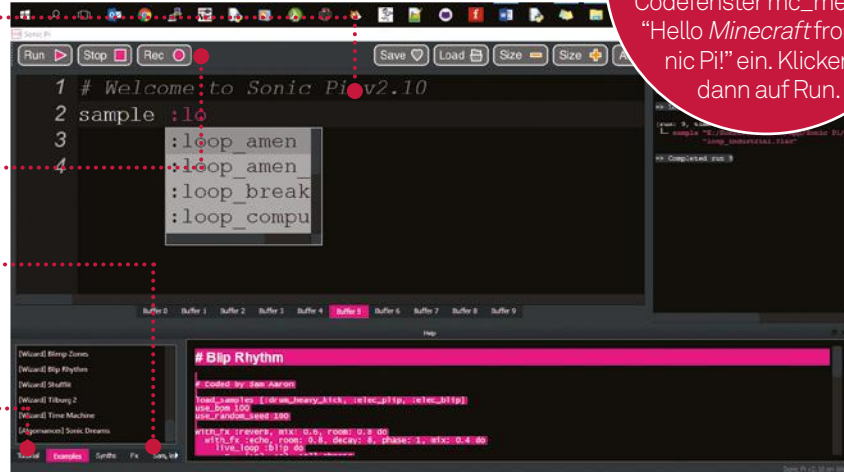
Das Interface des Sonic Pi

23 Das Codingfenster Es ist leicht, musikalische Kompositionen zu coden. Folgen Sie einfach einem Tutorial oder nehmen Sie eines der Beispiele als Grundlage.

24 Aufnehmen und exportieren Speichern und teilen Sie Ihre Werke als Code, damit Sie sie weiter bearbeiten, aufzeichnen und das Audio in eine WAV-Datei exportieren können.

25 Samples, Beispiele, Synths und mehr Bei mehr als 131 integrierten Samples, Synths und Soundeffekte ist für jeden etwas dabei. Zumal sich alles per Code an den eigenen Geschmack anpassen lässt.

26 Mehr erfahren Sonic Pi bietet eine riesige Auswahl an Tutorials, die alles von den Grundlagen für Anfänger bis hin zu fortgeschrittenen Fähigkeiten für erfahrene Profis abdecken.



22

Warum nicht mit *Minecraft* interagieren? Laden Sie beide Programme und geben Sie im Sonic Pi Codefenster `mc_message "Hello Minecraft from Sonic Pi!"` ein. Klicken Sie dann auf Run.

27

Der `sync`-Befehl sorgt dafür, dass alles aus dem Cache in den permanenten Speicher übertragen wird. Es ist nützlich, das etwa nach dem Update von Paketen laufen zu lassen.

28

Eine Note spielen

Melodien werden aus einzelnen Noten gebaut. Sonic Pi macht es sehr einfach, eine Note zu spielen. Um eine Note aufzurufen, tippen Sie im Codingfenster einfach `"play 60"` und drücken Run.

Diese Note ist C. Um eine andere Note zu spielen, ändern Sie den Wert der Zahl. Beispiele: `"play 70"` oder `"play 60"`. Versuchen es einfach. Anstatt der Notennummer können Sie auch den Buchstaben der Note eingeben, die Sie spielen möchten, also etwa `"play :C"` – das gibt die gleiche Note wieder wie das Kommando `"play 60"`. Um die Note höher zu machen, fügen Sie nach dem Buchstaben die Oktavenzahl ein. Ein Beispiel: Um die 4. Oktave D zu spielen, geben Sie `"play :D4"` ein. Sie können die Länge der Note mit dem `"release"`-Code verkürzen. Probieren Sie etwa einmal `"play :C6, release: 0.2"` aus. Erhöhen Sie die Dauer durch die Erhöhung der Zahl.



Oben Musiknoten entweder per Nummer oder Buchstaben erstellen

29

Noten zusammen spielen

Kombinierte Noten ergeben Akkorde. Das geht einfach, versuchen Sie es:

```
play 60
play 62
```

Der Akkord ist interessant und meint das gleichzeitige Erklängen unterschiedlicher Töne. Wenn das nicht Ihre musikalische Präferenz ist, können Sie auch einen traditionellen



Oben Akkorde durch die Kombination mehrere Noten programmieren



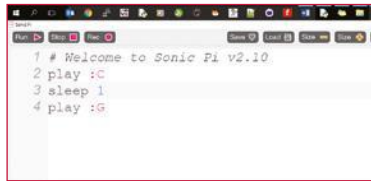
Klang wie C-Dur erstellen. Schlagen Sie Akkorde oder Noten online nach. Der Akkord von C-Dur entsteht aus den Noten C, E und einem G:

```
play :C, release: 1
play :E, release: 1
play :G, release: 1
```

30 Pause einfügen

Sie möchten in der Lage sein, Noten einzeln als Teil der Melodie zu spielen. Nutzen Sie die Sleep-Funktion, um anpassbare Pause zwischen Noten einzufügen. So machen Sie eine Pause von einer Sekunde vor der nächsten Note:

```
play :C
sleep 1
play :G
```



Oben Komponieren Sie Melodien, indem Sie durch Verzögerungen Rhythmus hinzufügen.

Abhängig von der Länge der ersten Note klingt die Pause vielleicht kürzer. Fügen Sie einen kurzen Release hinzu, um die Verzögerung zu verlängern und machen Sie die erste Note mehr stakkato. Standard für diese Release-Option ist eins:

```
play :C, release:
0.2
sleep 1
play :G
```

32 Themenmelodie

Dieser einfache Loop erinnert an die klassischen 8-Bit-Spiele

► Schritt 1: Schleife einrichten

In Zeile 1 richten wir die "live loop" genannte Schleife ein und fügen das Bass-Trance-Sample in Zeile 2 ein. Coden Sie eine kurze Verzögerung mittels sleep 0.8 in Zeile 3, beenden Sie Schleife dann in Zeile 4; klicken Sie Run. Das "Live looping" sorgt dafür, dass Sie während des Codens weiter die Musik hören. Lassen Sie das die nächsten Schritte lang so.

► Schritt 2: Schneller, mehr Noten

In Zeile 3 reduzieren Sie die sleep time auf 0.2, um den Beat zu beschleunigen. Mit dem "play choose"-Code in Zeile 4 wählen Sie aus einer Auswahl Noten aus. Experimentieren Sie mit den Werten, bis Sie etwas Passendes gefunden haben. Durch Hinzufügen der Zeile 5 "use_synth" wechseln Sie das Instrument. In diesem Beispiel bietet der Chiplead-Synthesizer ein schönes nostalgisches 8-Bit-Gefühl. Vielleicht möchten Sie jetzt die Noten in

der "play choose"-Zeile ändern, um die Melodie noch einmal zu ändern.

► Schritt 3: Zufallswahl

Sonic Pi kann mittels des Codes "if one_in(2)" ein Sample zum Abspielen auswählen. Das erzeugt ein Gefühl von Zufälligkeit. Sonic Pi wählt dafür entweder Sample eins oder Sample zwei. Fügen Sie zwei Samples und diese Zeilen hinzu, um Ihr Musikstück zu vervollständigen:

```
#8 bit music
live_loop :trance do
  sample :bass_trance_c
  sleep 0.2
  use_synth :chiplead
  play choose([40, 40, 60, 70, 73])
  sample :elec_snare
else
  sample :drum_tom_hi_hard
end
end
```



Fünf Tipps für das Coden von Musik

Instrumente, Töne & Beats

33 Das Instrument spielen und ändern

Zum Spielen einer Note nutzen Sie den Code "play 90". Um den Klang oder das Instrument zu ändern, fügen Sie den Code "use_synth :saw" wobei "saw" der Name des Synthesizers ist. (Andere sind unter dem Synth-Tab aufgeführt). Das betrifft nur die Aufrufe nach der Zeile, die erste Note bleibt der Standardton:

```
play 90
use_synth :piano
play 80
```

34 Samples

Unter dem Samples-Tab finden Sie zahlreiche Drum Beats, Riffs und Ambient Sounds. Scrollen Sie hinab bis "sounds for looping" und lokalisieren Sie ein Sample namens "sample :loop_mika". Nutzen Sie im Codingfenster die auto_complete-Funktion, die die Samples auflistet, und wählen Sie eine aus. Beginnen Sie mit der Eingabe von etwa "sample :loop_" und wählen Sie dann milka aus der Liste. Drücken Sie Run und versuchen Sie dann andere Beispiele.

35 Beispiele

Sonic Pi ist randvoll mit Beispielsamples für den Anfang. Wählen Sie ein Beispiel und fügen Sie es per Copy/Paste ins Codingfenster ein. Klicken Sie Run und dann können Sie es abspielen, modifizieren und tweaken wie Sie möchten.

36 Iterationen

Samples und Noten können mithilfe von Iterationen beliebig oft abgespielt werden. Wählen Sie aus den Beispielen select bd_haus, fügen Sie es hinzu und tippen Sie auf Run. Beachten Sie, dass es nur einmal abgespielt wird. Um eine mehrfache Wiedergabe zu erreichen, gehen Sie in Zeile 1 und fügen Sie hinzu "5.times do". Dann rücken Sie Zeile 2 ein und fügen in Zeile 3 eine kurze Pause ein: "sleep 0.5". Schlussendlich kommt noch ein "end" in Zeile 4:

```
5.times do
  sample :bd_haus
  sleep 0.5
end
```

37 Live Coding

Live coding ermöglicht es Ihnen, Musik in Echtzeit zu programmieren. Erstellen Sie einen einfachen Beat und fügen Sie ihn zum "live loop" hinzu. Die Musik beginnt und während Sie Werte in Ihrem Programmcode ändern, ändert sich die Musik in Echtzeit. Versuchen Sie den Code unten und ändern Sie die Rate; versuchen Sie 0.5, 2 oder sogar -1. Tippen Sie auf Run und hören Sie die Änderungen. Bearbeiten Sie auch die "sleep time":

```
live_loop :crazy_hiss do
  sample :vinyl_hiss, rate: 1
  sleep 2
end
```

Minecraft-Hacks meistern

Hacken Sie *Minecraft*, um das Spiel zu modifizieren, zu verbessern oder Ihre eigenen Spiele zu erstellen

38 Blöcke haben ID-Nummern

In der *Minecraft*-Umgebung hat jeder Block einen Namen und eine ID-Nummer als Referenz. Mit dem Befehl `mc.getBlock(p.x,p.y-1,p.z)` lässt sich die ID des Blocks auslesen, auf dem Sie gerade stehen. So bekommen Sie heraus, wo in der Welt Sie sich gerade befinden und wie nah Sie anderen Elementen sind. Sie können eine Antwort des Spielers programmieren. Etwa, dass die, die auf Gras stehen, in die Luft katapultiert werden:

```
from mcpi import minecraft
mc = minecraft.Minecraft.create()
while True:
    p = mc.player.getTilePos()
    b = mc.getBlock(p.x,p.y-1,p.z)
    if b == 2:
        mc.player.setPos(p.x, p.y+20, p.z)
```



Oben Finden Sie die Block-ID und springen Sie in die Luft.

39 Wissen, wo Ihr Spieler ist

In *Minecraft* ist Ihr Spieler in einer dreidimensionalen Umgebung. Die x-Achse verläuft von links und rechts, die y-Achse nach oben und unten und die z-Achse vorwärts und rückwärts. Während Sie sich bewegen, wird Ihre Position oben links am Bildschirms als ein Satz aus drei Koordinaten angezeigt. Die sind ä-

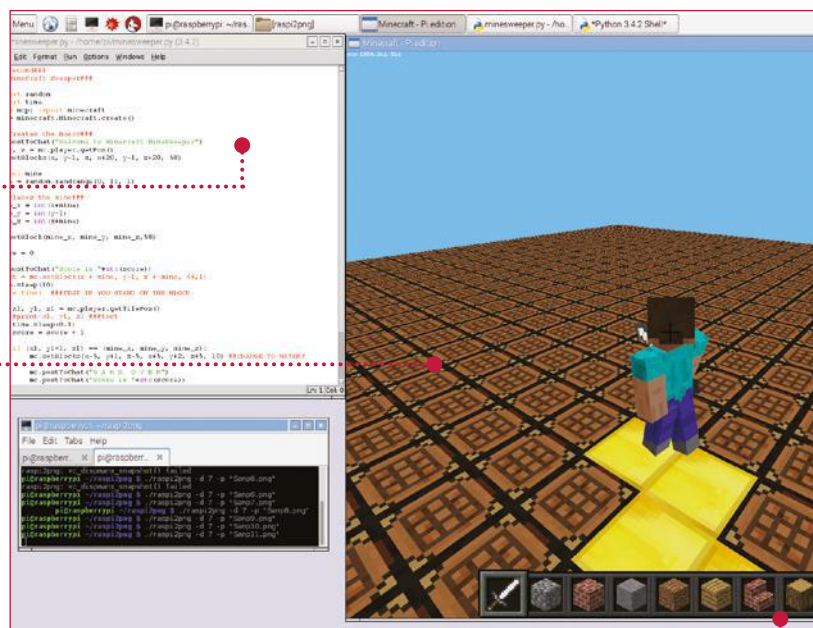
ußerst nützlich, um zu überprüfen, wo der Spieler ist, und können mittels `pos = mc.player.getPos()` gesammelt und gespeichert werden. Dieser Code liefert die Position des Spielers, kann aber auch jeden Block im Spiel finden. Mit dieser Methode können Sie Hacks programmieren, die die Entfernung des Spielers zu einem Ort messen, Ereignisse auslösen, wenn Sie auf einem bestimmten Block stehen, oder den Spieler sogar teleportieren.

Die Programmierumgebung von Minecraft

40 Python-Programme Nutzen Sie Python-Skripte, um die *Minecraft*-Welt zu manipulieren und interaktive Herausforderungen, responsive Umgebungen und übermenschliche Spieler-Mods zu entwickeln.

41 Minecraft Das Image des Raspberry Pi-Betriebssystems enthält eine grundlegende Pre-Alpha-Version von *Minecraft*. Diese ist vorinstalliert und bereit zum Spielen. Es handelt sich dabei um eine abgespeckte Version des klassischen Spiels, ist aber als Programmierspielplatz vollkommen ausreichend. Eine bessere Version erhalten Sie auf der Website von Minetest unter <http://www.minetest.net>.

42 Auf die Größe kommt es an *Minecraft* im Vollbild laufen zu lassen, versteckt Ihren Code und die Maus rendert nicht synchron. Wir empfehlen den Fenstermodus, sodass Sie gleichzeitig das Spiel wie auch Ihren Code sehen.





Oben Senden Sie im Spiel Nachrichten an andere Spieler.

44 Eine Meldung anzeigen

Nachrichten können in der *Minecraft*-Umgebung angezeigt werden, um die Spieler auf dem Laufenden zu halten, was passiert. Das geht mit dem Code `mc.postToChat()` – die Nachricht selbst schreiben Sie dann zwischen die Klammern:

```
from mcpi import minecraft
mc = minecraft.Minecraft.create()
mc.postToChat("Hello world")
```

Meldungen sind auch nützlich für die Weitergabe von Daten an den Spieler. Etwa Ihre aktuelle Position, die ID des Blocks, auf dem Sie stehen, oder wie weit weg von einem bestimmten Block Sie stehen.

45 Minecraft-Fotokabine

Fotos mit der PiCamera in *Minecraft* machen

► Schritt 1: Kabine erstellen

Erstellen Sie in *Minecraft* eine einfache Box als Fotokabine. Laufen Sie hinein und beachten Sie die x-, y- und z-Koordinaten, die oben links auf dem Bildschirm angezeigt werden. Das ist der Block, auf dem Sie stehen; der Triggerblock. Öffnen Sie Python und geben Sie den Code ein:

```
from mcpi.minecraft import Minecraft
from picamera import PiCamera
from time import sleep
mc = Minecraft.create()
camera = PiCamera()
mc.postToChat("Find the photobooth")
```



► Schritt 2: Bild aufnehmen

Fügen Sie nun den Programmcode hinzu, um Ihren aktuellen Standort zu überprüfen. Wenn Sie auf dem Triggerblock stehen, wird eine "Lächeln...!"-Nachricht angezeigt. Die Kamera zeigt eine Vorschau und dann wird das Foto gemacht. Es wird unter `/home/pi` als Datei `selfie.jpg` gespeichert. Lächeln nicht vergessen ...!

```
while True:
    x, y, z = mc.player.getPos()
    if x >= 10.5 and y == 9.0 and z == -44.3:
        mc.postToChat("You are in the photobooth!")
        sleep(1)
        camera.start_preview()
        sleep(2)
        camera.capture('/home/pi/selfie.jpg')
        camera.stop_preview()
        mc.postToChat("Check out your picture")
        sleep(3)
```

Verwenden Sie die Tasten, um Hacks auszulösen. Kombinieren Sie das mit einem Wearable wie z. B. dem *Minecraft Power Glove*:
www.tecoed.co.uk/piglove-minecraft-power-up.html

Fünf Minecraft-Hacks

Grundlagen für das Hacking der *Minecraft*-Welt

46 Finden Sie Ihren Standort

Um Ihre Umgebung zu manipulieren, müssen Sie zunächst wissen, wo Sie sich befinden. Mittels des Codes `mc.player.getPos()` sammeln Sie die Werte für X, Y, und Z Ihres Standorts und geben Ihre Position aus:

```
from mcpi import minecraft
mc = minecraft.Minecraft.create()
import time
while True:
    time.sleep(1.0)
    pos = mc.player.getPos()
    print pos.x, pos.y, pos.z
```

47 Einen Spieler an eine Position schicken

Jetzt, wo Sie Ihre Position kennen, können Sie den Spieler mittels `mc.player.setPos` automatisch an eine andere Position bewegen. Im unteren Beispiel transportieren wir die Spielfigur 100 Blöcke vorwärts:

```
from mcpi import minecraft
mc = minecraft.Minecraft.create()
x, y, z = mc.player.getPos()
mc.player.setPos(x, y+100, z)
```

48 Einen Block setzen

Ihre Figur kann mit dem Kommando `setBlock` einen einzelnen Block setzen. Lokalisieren Sie Ihre Position und setzen Sie dann einen Block dorthin. In diesem Beispiel ist 38 die Block-ID für Blumen:

```
while True:
    x, y, z = mc.player.getPos()
    mc.setBlock(x, y, z, 38)
    time.sleep(0.1)
```

49 Mehrere Blöcke setzen

Möchten Sie mehr als einen Block gleichzeitig setzen, nutzen Sie das Kommando `setBlocks`. Lokalisieren Sie Ihre Position, fügen Sie die Position hinzu, an der die Blöcke stoppen sollen und auch die ID der Blöcke, die Sie haben möchten. In diesem Beispiel ist Block Nummer eins ein Stein:

```
x, y, z = mc.player.getPos()
mc.setBlocks(x+1, y+1, z+1, x+11, y+11, z+11, 1)
```

50 Interagieren in der Welt eines anderen Spielers

Es macht Spaß, die eigenen Welt zu hacken, aber wie es wäre mit der eines anderen Spielers? Suchen Sie in einem Netzwerkspiel mittels `sudo ifconfig` die IP-Adresse eines anderen Spielers und notieren Sie diese. Fügen Sie in Ihrem Code diese IP-Adresse zur Codezeile `mc = minecraft.Minecraft.create(192.168.1.45)` hinzu. All Ihre Hacks erscheinen nun in der Welt des anderen Spielers.

10 GENIALE RASPBERRY PI UPGRADES

Von Solar-Power-Packs und E-Paper-Displays bis zur 3D-Gestensteuerung: Hier stellen wir zehn tolle Erweiterungen für den Pi vor.

In seinem relativ kurzen Leben von nur knapp fünf Jahren hat sich der Raspberry Pi zu einem wahren Alleskönner entwickelt. Das zeigt vor allem die Community, die mit dem Pi ein gemeinsames Ziel verfolgt und sich sowohl auf Lernprozesse fokussiert, am Ende aber auch tolle Ideen verwirklicht.

Wir können uns nicht erinnern, wie oft wir den Raspberry-Pi-Blog durchstöbert haben und einfach nur von den Projekten begeistert waren, die wir dort gefunden haben. Von der Sortierung der portugiesischen Post bis hin zur Herstellung von Bullet-Time-Rigs: Es gibt so viele tolle Projekte da draußen und jeden Tag werden es mehr! All das liegt an den schier unendlichen Möglichkeiten, die der Raspberry Pi mit sich bringt. So ziemlich jede Idee kann damit umgesetzt werden.

Vergleichen wir den Raspberry Pi mit einem durchschnittlichen Computer oder einem mobilen Gerät, liegt der Vorteil in seinen vielseitigen Upgrade- und Anpassungsmöglichkeiten. Bei einem Smartphone oder Tablet können Sie sich eine coole Hülle oder die neuesten Kopfhörer zulegen. Der Pi geht hier aber noch viel weiter – und das sowohl in Sachen Soft- als auch Hardware. Viele Projekte da draußen scheinen tatsächlich die Manifestierung eines Kindheitstraums zu sein. Das macht den Raspberry Pi aus. Denn nur mit ihm sind die wahnwitzigsten Projekte möglich.

Hier wollen wir Ihnen nun zehn unserer Lieblings-Raspberry-Pi-Upgrades vorstellen, mit denen Sie Ihre Ideen zum Leben erwecken oder sich einfach nur Inspirationen holen können.



Schützen Sie Ihren Pi!



Short Crust Plus

Preis: 10,92 € Hier kaufen:
bit.ly/1ICxbvw

Der Raspberry Pi ist ein langlebiger und zuverlässiger kleiner Computer – vor allem wenn Sie bedenken, dass es sich dabei einfach nur um eine Platine ohne wirklichen Schutz handelt. Um den Pi zu schützen, liegt es deshalb nahe, eine Art Gehäuse zu verwenden. Damit sieht der kleine Computer

für manche nicht nur besser aus, sondern wird auch vor lästigem Staub geschützt, wenn Sie ihn in Ihrer Tasche auf Reisen mitnehmen.

Das Short Crust Plus ist aufgrund des schmalen, tollen Designs und der guten Features unser Lieblingsgehäuse für das Model B+ und die 2B Raspberry Pis. Der Pi passt hier perfekt in das Gehäuse, das über Gummknöpfe an der Unterseite verfügt, um die Stabilität zu gewährleisten. Selbst bei Platinen-Upgrades kann das Short Crust Plus individuell angepasst werden, was es zu einem universellen Begleiter in allen Lebenslagen macht.

Tragbar & Solarenergie

PiJuice

Preis: 30,42 € Hier kaufen:
bit.ly/1Fb1ywy

Mit einem kleinen, eleganten Add-on-Board ist es nun möglich, Projekte abseits von Netzstromquellen zu realisieren. Das Upgrade nennt sich PiJuice, ist mit der Raspberry-Pi-HAT-Spezifikation kompatibel und macht sich einen mobilen Smartphone-Akku zunutze, mit dem Sie Ihre Arbeit auch unterwegs ausführen können. Es gibt sogar eine Version mit dem Namen PiJuice Solar. Damit dürfen Sie dann auf Solarenergie als Stromquelle zugreifen und auch andere erneuerbare Energiequellen dank der vielseitigen Anschlüsse verwenden.

PiJuice verfügt zudem über einen leistungsstarken ARM-Cortex-M0-Prozessor, der unter anderem eine Echtzeituhr bereitstellt. Die Firmware

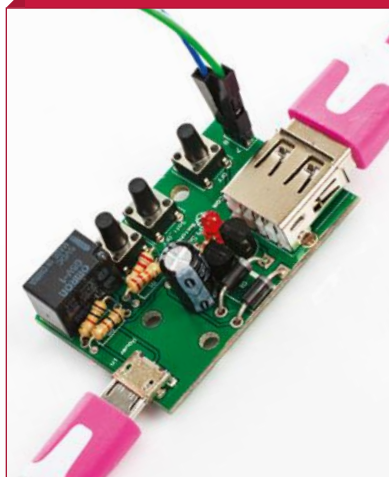
und das GUI (die grafische Benutzeroberfläche) des PiJuice kommunizieren mit der ACPI-Batterie (Advanced Configuration and Power Interface) und anderen APIs, was für eine direkte Integration in Raspbian sorgt. PiJuice nutzt lediglich 12C-Power und einen GPIO-Pin, sodass die meisten GPIO-Pins für andere Projekte frei bleiben. Mit dem PiJuice ist es darüber hinaus auch problemlos möglich, andere HATs oder Add-on-Boards einfach zu montieren. Weitere Informationen zu diesem wirklich nützlichen Add-on finden Sie auf dieser Webseite: bit.ly/1e2CoGE.



Netzschalter & sicher herunterfahren

Pi Supply Switch

Preis: 18,25€ Hier kaufen:
bit.ly/1RXHR0n



Der Raspberry Pi ist so beliebt, weil die Hardware so unglaublich günstig ist. Das liegt vor allem daran, weil der Computer von einer Wohltätigkeitsorganisation entworfen wurde. Hier gibt es also keine aufgeblästen Preise und Gewinnspannen wie bei kommerziellen Produkten, die bedient werden müssen. Aber wie bei jedem Produkt mit günstigen Preisen müssen auch beim Pi Zugeständnisse gemacht werden. Ansonsten wäre es nicht möglich, den Computer so klein und handlich zu halten.

Beim Vergleich mit Ihrem Desktop- oder Laptop-Computer fällt auf, dass der Pi über keinen Netzschalter oder eine Energiemanagement-Funktion verfügt. Es ist überraschend,

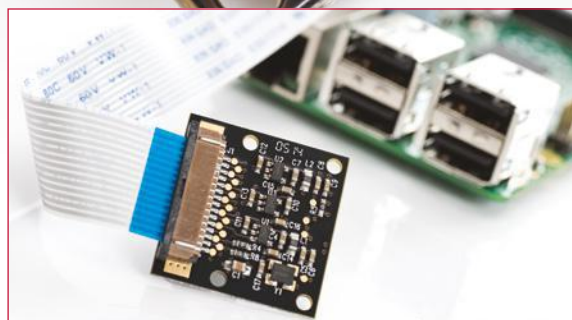
wie so etwas Einfaches so nützlich sein kann – zumindest dann, wenn Sie erst einmal in den Genuss dieser Funktion kommen.

Beim Pi Supply Switch handelt es sich um ein Selbstlötkit, das eine Ein-, Aus- und Soft-Aus-Taste zur Verfügung stellt. Letztere Funktion sorgt dafür, dass Sie Daten auf dem Pi beim Ausschalten nicht verlieren. Mit einigen implementierten Beispielskripts können Sie sicherstellen, dass Ihr Pi korrekt heruntergefahren wird, wenn Sie ihn ausschalten. Somit ist es also nicht mehr nötig, Menüs zu öffnen oder Befehle im Terminal auszuführen. Zudem stellt der Pi Supply Switch sicher, dass der Strom erst nach dem Herunterfahren gekappt wird.

Im Dunkeln mit Infrarot sehen

NoIR Infrarot-Kamera

Preis: 30, 41 € Hier kaufen:
bit.ly/2u0dBzt



Der CSI-Anschluss des Raspberry Pi (zwischen dem 3,5-mm-Klinkenstecker und dem HDMI-Anschluss auf den aktuellsten Modellen) ermöglicht es Ihnen, ein Kameramodul direkt anzuschließen. Die Kameramodule, die Sie hier verbinden können, verwenden weniger Strom und bieten – wie Sie es von der Raspberry Pi Foundation erwarten würden – einen beeindruckend kleinen Formfaktor. Wir sprechen hier von 25 x 24 x 9 mm und einem Gewicht von rund drei Gramm, allerdings ohne Kabel.

Hinsichtlich der Kameramodule können Sie selbstverständlich auf ein „normales“ Upgrade zurückgreifen. Also auf ein solches, das sichtbares Licht einfängt. Die Leistungsmerkmale sind beeindruckend: Es handelt sich hierbei um eine 5-Megapixel-Fokuskamera, die 1080p30, 720p60 und VGA90 unterstützt (weitere Informationen hier: bit.ly/1Gy3D8q). Mit der Veröffentlichung dieses Kameramoduls waren einige Nutzer aber nicht zufrieden, da sie andere Ideen verfolgten. Sie bauten die Kamera also auseinander und entfernten den Infrarot-Filter. Diese Methode fand in der Community schnell Zuspruch, sodass die Raspberry Pi Foundation einfach selbst eine Infrarot-Kamera veröffentlichte – die Pi NoIR.

Für Nachtaufnahmen ist die Pi NoIR wunderbar geeignet. Sie können dann in Kombination mit Infrarot-LEDs (sehen Sie sich dazu das Bright-Pi-Add-on-Board an) eine Sicherheitskamera entwickeln oder das Modul zur Tierüberwachung einsetzen. Ebenfalls bemerkenswert: Bei Tag lässt sich die Pi NoIR zur Überwachung von Pflanzen verwenden. Weitere Informationen dazu finden Sie hier: bit.ly/1QnZdFG.



Bewegung für das Kamera-Rig

Pi-Pan, ein Pan/Tilt-Kamera-Aufsatz

Preis: 55,97 € Hier kaufen: bit.ly/1dwpEr2

Das Kameramodul und die Pi NoIR, die wir Ihnen bereits vorgestellt haben, sind wesentliche Upgrades, die Sie in Ihrer Pi-Werkzeugkiste haben sollten. Aber wie sieht es aus, wenn Sie die Kamera bewegen wollen, um verschiedene Ansichtswinkel zu erhalten? Dafür gäbe es zumindest zahlreiche Einsatzgebiete, zum Beispiel für einen Roboter mit einer beweglichen Kamera oder eine mit dem Internet verbundene Webcam, die über ein Web-Interface gesteuert wird.

Hier präsentieren wir Ihnen deshalb das Pi-Pan. Mit diesem Pan/Tilt-Kamera-Aufsatz können Sie Ihre Kamera auf verschiedene Weise ausrichten – 110 Grad von oben nach unten und 180 Grad von links nach rechts. Der Bausatz verfügt zudem über eine durchdachte Hardware, einschließlich einer servobetriebenen Platine sowie der dazugehörigen Motoren. Auf der Software-Seite gibt es hingegen Bibliotheken in Python und Scratch, die für die meisten Projekte flexibel genug sein sollten.

Eine der beeindruckendsten Anwendungen, die Sie damit realisieren können, ist eine OpenCV-basierte Bewegungserkennung und Face-Tracking-Kamera. Beispielcode dazu finden Sie im openelectrons.com-Forum. Weitere Informationen gibt es hier: bit.ly/1JpXLe.

Bestes Audio für Ihren Pi

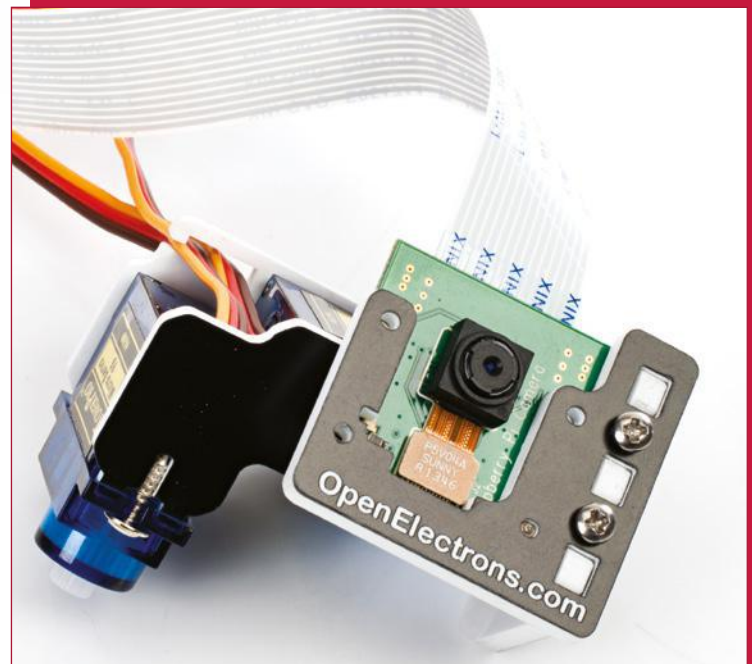
HiFiBerry DAC+

Preis: 36,51 € Hier kaufen: bit.ly/1L1hh4T

Aus pädagogischer Sicht ist der Raspberry Pi durch die Unterstützung der sehr großen Community ein unvergleichliches Werkzeug. Deshalb ist es auch kein Wunder, dass der Computer mit einem Upgrade zu einer wahren Klangmaschine avanciert. Der Standard-3,5-mm-Klinkenanschluss ist dazu von Haus aus nämlich nicht geeignet.

Aufgrund der geringen Kosten für einen Raspberry Pi, der kleinen Stellfläche und seiner Fähigkeit, ihn als Medienzentrum für die eigenen vier Wände zu betreiben, haben Sie in der Vergangenheit bestimmt auch schon davon geträumt, den Pi in seiner Klangqualität zu verbessern. Da Sie mit diesem Gedanken nicht alleine waren, haben sich die Männer der Raspberry Pi Foundation Gedanken gemacht und stellen bei der zweiten Revision des Model B Zugriff auf die I2S-Pins zur Verfügung. Auf den A+, B+ und 2B-Modellen besteht der Zugriff hingegen über die 10-Pin-GPIO-Header.

Bei I2S handelt es sich um ein Kommunikationsprotokoll, das speziell für Audiogeräte entwickelt wurde und einer Reihe von Firmen wie HiFiBerry und IQaudIO ermöglicht hat, hochwertige Audio-Add-ons für den Raspberry Pi zu entwickeln. Der HiFiBerry DAC+ ist beispielsweise ein Add-on, das einen qualitativ hochwertigen Digital-Analog-Wandler zu Ihrem Pi hinzufügt. Das HAT-kompatible Board verfügt zudem über Lautstärkenregelung mit Alsamixer. Auch sonst gibt es nicht viel zu beachten, denn die Platine ist mit den jüngsten Raspberry-Pi-Firmwares kompatibel und kann sofort verwendet werden. Wer also nach einer Audio-Lösung für den Pi sucht, der wird mit HiFiBerry DAC+ bestens bedient.





Adafruit 10.1" Display & Audio

Preis: 133,86 € Hier kaufen:
bit.ly/1HrfR1s

Mit einem Preis von etwa 130 Euro ist das Adafruit-Display sicher kein Schnäppchen, suchen Sie aber nach einem qualitativ hochwertigen Bildschirm, kommen Sie um dieses Produkt nicht herum. Stellen Sie sich nur einmal vor, was Sie damit alles machen können ...

Tolle Low-Power-Displays

PaPiRus ePaper/elnk HAT

Preis: 43,50 € Hier kaufen:
bit.ly/1f2Lzaj

Wollen Sie die ePaper-Display-Technologie auch mit Ihrem Raspberry Pi nutzen, sollten Sie sich das entsprechende Produkt von PaPiRus näher ansehen. Das HAT-kompatible Format kann mit Bildschirmgrößen zwischen 1,44 und 2,7 Zoll betrieben werden, die tatsächlich auf dem Material der bekannten eReader basieren. Besonders teuer ist das Upgrade auch nicht: Um in den Genuss der Technologie zu kommen, müssen Sie zwischen 43,80 € und 85,17 € investieren – je nach Bildschirmgröße.





Steckdosen kontrollieren

Energenie Pi-mote Funkschalter-Set

Preis: 24,33 € Hier kaufen: bit.ly/1L1kYHU

Hausautomation ist gerade in aller Munde – vielleicht weil Menschen einfach nur faul sind oder weil es einfach Spaß macht, diese Technik zu verwenden. Egal was die Intentionen sind, diese Technologie macht unser Leben definitiv einfacher. Nur so können wir schließlich das Licht vor dem Schlafen ausschalten, ohne noch einmal aus dem Bett aufstehen zu müssen, um den Schalter manuell zu betätigen.

Etwas, was uns schon unsere Eltern eingebracht haben, ist die Tatsache, elektrische Geräte immer aus der Steckdose zu ziehen, um nicht auch die Stromkosten für den Stand-by-Modus bezahlen zu müssen. Was aber, wenn die Steckdose nicht wirklich gut zu erreichen ist? In diesem Fall kommt das Energenie Pi-mote Funkschalter-Set zum Einsatz, das über zwei ferngesteuerte Steckdosen verfügt. Diese können dann per Fernbedienung aktiviert und deaktiviert werden. Und was hat das Ganze nun mit dem Raspberry Pi zu tun? Ganz einfach: Im Lieferumfang befindet sich auch ein Add-on-Board, mit dem Sie die Steuerung über den Pi durchführen können. Das bietet viele neue Möglichkeiten, beispielsweise können Sie so auch verschiedene Lampen ein- und ausschalten. Selbst eine Kombination mit dem Smartphone ist hier möglich. Auch in diesem Bereich spielt der Raspberry Pi also seine Trümpfe aus.

Gesten- & Touch-Steuerung

Pimoroni Skywriter HAT

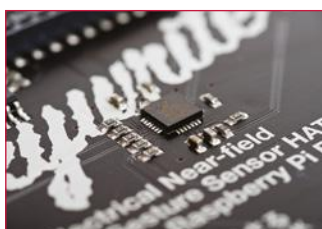
Preis: 19,47 € Hier kaufen: bit.ly/1lFt9cg

Bei vielen Projekten, die Sie mit dem Raspberry Pi realisieren wollen, ist irgendeine Art von Benutzerinteraktion nötig. Bei der Verwendung des Desktop-GUI wird das normalerweise mit einer Tastatur und einer Maus bewerkstelligt. Hierbei handelt es sich aber nicht immer um die intuitivste Eingabemethode, vor allem dann nicht, wenn keine vollständige Desktop-Umgebung vorhanden ist oder nichts eingetippt werden muss.

Die Piraten von Pimoroni haben deshalb ein HAT-Modul mit dem Namen Skywriter entwickelt, das Nahfeld-3D-Gesten und Touch-Steuerung zu einem günstigen Preis ermöglicht. Mit von der Partie ist eine Python-API, die vollständige 3D-Positionsdaten und Gesteninformationen bereitstellt (Wischen, Tippen und so weiter). Verwenden Sie das Upgrade, werden Sie feststellen, dass es sich dabei um eine tolle Eingabemethode handelt.

Zur Verfügung steht sogar eine größere Nicht-HAT-Version von Skywriter, die bis zu einem Abstand von 15 cm funktioniert. Sie können dieses Upgrade also sogar hinter einem Blatt Papier oder einer Acrylscheibe verstecken, um Ihre Projekte wie Magie aussehen zu lassen.

Der Preis für das Pimoroni Skywriter HAT beläuft sich auf lediglich knapp 20 Euro.



Roboter steuern

Schreiben Sie ein Programm, mit dem Sie einen Spielzeugroboter mithilfe des Raspberry Pi und des IR-Boards von Energenie steuern.

In der folgenden Anleitung zeigen wir Ihnen, wie Sie Signale von einem Spielzeugroboter aufnehmen und diesen durch Pygame-Befehle von einem Raspberry Pi aus steuern. Da der Code auf Python aufbaut, können Sie ihn später weiterentwickeln und zum Beispiel den Roboter mithilfe von Tweets oder über ein Web-Interface mit einem Smartphone steuern.

01 Bootdatei updaten

Schließen Sie das IR-Board Pi-Mote an die GPIO-Pins des Raspberry Pi an, bevor Sie diesen starten. Das Board passt auf die oberen Pins. Falls keine Pins anders konfiguriert worden sind, verwendet es die Pins wie folgt: Eingang für empfangenes Infrarotsignal = PIN12/GPIO18; Ausgang für gesendetes Signal = PIN11/GPIO17. Es empfiehlt sich, ein neues SD-Karten-Image zu verwenden, auf dem keine weitere Software installiert ist, um

Konflikte zu vermeiden. Starten Sie den Raspberry Pi und geben Sie im Terminal Folgendes ein:

```
sudo apt-get update  
sudo apt-get upgrade
```

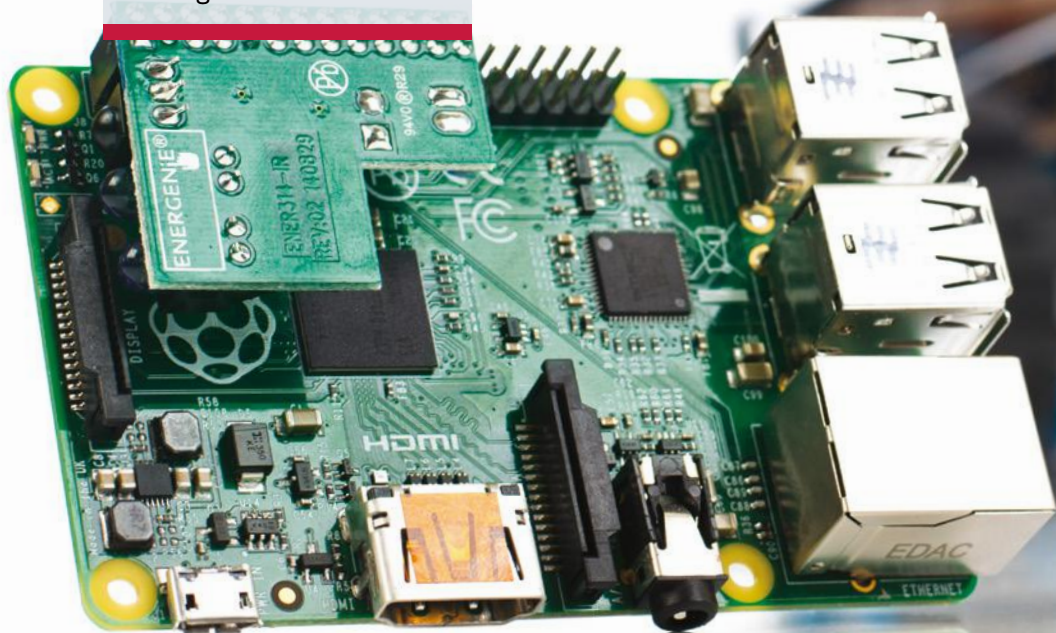
Fügen Sie danach eine Codezeile in die Datei `/boot/config.txt` ein, um die LIRC-IR-Software und das IR-Modul zu aktivieren. Geben Sie im Terminal „`sudo nano /boot/config.txt`“ ein. Damit wird die `config.txt`-Datei geladen. Scrollen Sie ans untere Ende des Textes und geben Sie ein:

```
dtoverlay=lirc-rpi-overlay
```

Speichern Sie die Datei und starten Sie den Pi durch die Eingabe von „`sudo reboot`“ neu.

Was Sie brauchen

- Pi-Mote
bit.ly/1MdpFOU
- Ferngesteuerter Roboter





Links Das Pi-Mote ist vielseitig einsetzbar und kostet nur etwa 12 €.

02 Software installieren

Installieren Sie nun die LIRC-Software (LIRC steht für Linux Infrared Remote Control). Sie ermöglicht es Ihnen, dem Roboter Befehle zu übermitteln. Öffnen Sie erneut die Kommandozeile:

```
sudo apt-get install lirc
sudo apt-get install lirc-x
```

Nach der Installation starten Sie den Pi mit „sudo reboot“ wieder neu.

03 Hardware-Datei ändern

Jetzt müssen wir die hardware.conf-Datei im /etc/lirc-Ordner anpassen. Geben Sie im Terminal ein:

```
sudo nano /etc/lirc/hardware.conf
```

Suchen Sie die DRIVER-, DEVICE- und MODULES-Zeilen und verändern Sie sie folgendermaßen:

```
DRIVER = "default"
DEVICE = "/dev/lirc0"
MODULES = "lirc_rpi"
```

Speichern Sie die Datei, ohne ihren Namen zu ändern. Starten Sie nun den LIRC Daemon mit „sudo /etc/init.d/lirc.restart“ neu.

04 IR-Empfänger testen

Um den Empfänger auf seine Funktionalität hin zu überprüfen, müssen Sie im Terminal den LIRC Daemon stoppen, den Testmodus starten und dann das Typ-2-Testverfahren aufrufen:

```
sudo /etc/init.d/lirc stop
sudo modprobe lirc_rpi
sudo mode2 -d /dev/lirc0
```

Hierdurch wird ein Programm gestartet, welches das Tastverhältnis des IR-Signals ausgibt. Es misst den Puls und die Länge

des Signals und sendet die Werte zum Terminal zurück. Nehmen Sie sich nun die Fernbedienung des Roboters, richten Sie sie auf den IR-Empfänger, und drücken Sie einige Knöpfe. Sie sollten nun etwas Ähnliches wie das hier sehen:

```
space 16300
pulse 95
space 28794
pulse 80
space 19395
space 28794
pulse 80
```

05 lircd.conf-Datei erstellen: Teil 1

In der lircd.conf-Datei stehen die Befehle, mit denen Ihr Gerät gesteuert wird. Auch wenn man diese online findet, ist es besser, sie von Grund auf neu zu erstellen. Dies beinhaltet das Ausführen des Programms Irrecord, das Ausrichten der Fernbedienung auf das IR-Board und dann das Drücken von Knöpfen. Dadurch werden die Signale aufgezeichnet, welche Sie dann Tasten zuordnen müssen, mit denen Sie die Signale an den Roboter schicken möchten. Stoppen Sie die LIRC-Software mit folgendem Befehl:

```
sudo /etc/init.d/lirc stop
```

06 lircd.conf-Datei erstellen: Teil 2

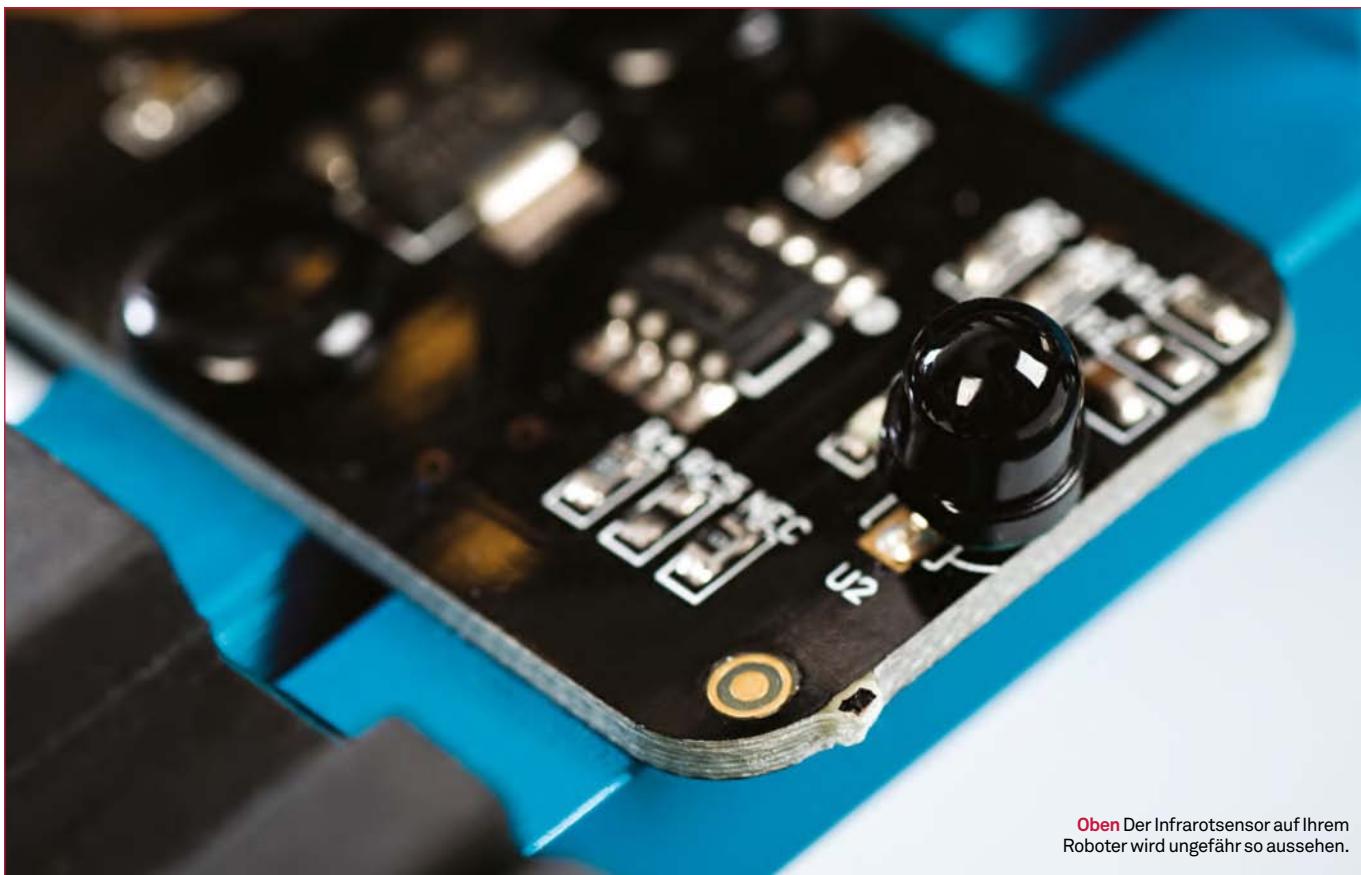
Erstellen Sie als Nächstes eine eigene lircd.conf-Datei und speichern Sie den Output. Geben Sie im Terminal ein:

```
irrecord -d /dev/lirc0 ~/lircd.conf
```

Dies öffnet das Programm, mit dem Sie die Signale Ihrer Fernbedienung aufnehmen können. Im ersten Teil drücken Sie wiederholt die Knöpfe Ihrer Fernsteuerung, bis Ihr Bildschirm zwei Punktzeilen anzeigt. Hiermit werden die Signale eingemessen und aufgezeichnet. Folgen Sie hierbei einer logischen Reihenfolge: Fangen Sie oben an und arbeiten Sie sich nach unten durch. Haben Sie die zwei Zeilen abgeschlossen, ist Ihre Fernbedienung erkannt worden.

Pygame-Knöpfe

Pygame ermöglicht es Ihnen, Knöpfe zu erstellen und diesen Handlungen zuzuordnen. Will heißen, Sie erstellen und weisen dem Pygame-Fenster Knöpfe zu. Sie können also den Roboter mit Knöpfen anstatt mit Tasten bedienen. Unter bit.ly/2m2b4jt finden Sie ein einfaches Tutorial. Dieses zeigt Ihnen, wie Sie einige Interaktionsmöglichkeiten mit verschiedenen Knöpfen erstellen können.



Oben Der Infrarotsensor auf Ihrem Roboter wird ungefähr so aussehen.

Roboter-variation

Sie können eine Vielzahl an Robotern mit diesem Tutorial verwenden, wie den abgebildeten MakeBlock oder den WowWee Roboquad, den unser Autor verwendet hat. Beachten Sie die verschiedenen Ereigniskonfigurationen in Ihrem Code – verschiedene Roboter haben unterschiedliche Funktionen. Das Roboquad hat einen „Tanz“-Befehl, welchen Sie mittig in der gegenüberliegenden Code-Auflistung sehen können.

07 lircd.conf-Datei erstellen: Teil 3

Der zweite Teil des Programms verlangt von Ihnen, die Namen der jeweiligen Tasten einzugeben, die von diesem aufgenommen wurden. Folgen Sie jeder der Bildschirmanweisungen, indem Sie einen passenden Namen für die Knöpfe/Tasten eingeben. Geben Sie zum Beispiel „KEY_UP“ ein, wenn Sie den entsprechenden Knopf auf der Fernbedienung drücken. Im Anschluss werden Sie aufgefordert, den Namen der nächsten Taste – in diesem Fall „KEY_DOWN“ – einzugeben, worauf Sie auf der Fernbedienung die Taste nach unten drücken. Führen Sie die Prozedur durch, bis Sie alle aufgezeichneten Knöpfe mit Namen versehen haben.

08 Fernbedienung umbenennen

Ist die Datei abgespeichert, suchen Sie nach dieser neuen lircd.conf-Datei im /home/pi-Ordner. Der voreingestellte Name der Bedienung in Zeile 14 wird vermutlich /home/pi/lircd.conf lauten. Ändern Sie diesen: Suchen Sie unter der Überschrift „begin remote“ nach der Bezeichnung „name“ und nennen Sie /home/pi/lircd.conf in etwas Passenderes wie zum Beispiel „Robot“ um. Hierdurch wird es wesentlich einfacher, sich im Programm auf die Fernbedienung zu beziehen, sobald Sie einen Befehl ausführen oder eine Taste aufrufen möchten.

09 Datei lircd.conf übertragen

Nun ist Ihre lircd.conf-Datei bereit, in den /etc/lirc-Ordner kopiert zu werden. Hier werden die Hardware- und Lirc-Dateien aufbewahrt. Die einfachste Methode ist es, den Code aus Ihrer neuen lircd.conf-Datei, welche im Order /pi/home gespeichert ist, per Copy & Paste zu übertragen. Hiermit wird jedoch jegliche alte Konfiguration unwiederbringlich gelöscht. Um die vorherige Datei zu erhalten, folgen Sie dem nächsten Schritt. Geben Sie ins Terminal ein:

```
sudo nano /etc/lirc/lircd.conf
```

10 Ohne Überschreiben übertragen

Falls Sie eine lircd.conf-Datei bereits eingerichtet haben und diese behalten möchten, erstellen Sie eine neue. Diese wird im Order /home/pi gespeichert und kann in /etc/lirc kopiert werden. Erstellen Sie ein Backup von der Originaldatei, indem Sie diese kopieren und im Anschluss als lircd_original.conf speichern. Geben Sie hierzu Folgendes ein:

```
sudo /etc/init.d/lirc start
sudo mv /etc/lirc/lircd.conf /etc/lirc/lircd_
original.conf
```

Kopieren Sie danach die neue Konfigurationsdatei:

```
sudo cp ~/lircd.conf /etc/lirc/lircd.conf
```

Die Originaldatei wird nun als lircd_original.conf gespeichert.

11 Übernehmen Sie die Kontrolle

Jetzt hat Ihr Roboter eine Konfigurationsdatei. Starten Sie LIRC mit „sudo /etc/init.d/lirc restart“. Überprüfen Sie die Funktionalität der lircd-Datei, indem Sie sich alle gespeicherten Tasten auflisten lassen:

```
irsend LIST Robot " "
```

Nun wird es Zeit, Ihr Gerät anzusteuern. Hierfür verwenden wir die Irsend-Anwendung, die mit LIRC mitgeliefert wird, um Befehle zu senden. Die Befehle sind einfach: „irsend SEND_ONCE Remote_Name Remote_Button“. Um den Roboter sich nach vorne bewegen zu lassen, geben Sie einfach dies ins Terminal ein: „irsend SEND_ONCE Robot KEY_UP“. Das Vorwärtssignal wird per Infrarot gesendet, und Ihr Roboter läuft nach vorne.



12 Python OS

Mit dem LIRC-Programm kontrollieren Sie das Pi-Mote über die Kommandozeile. Für das Coding ist dies jedoch unpraktisch, weil dadurch die Interaktionsmöglichkeiten mit anderer Hard- oder Software eingeschränkt wird. Im guten, alten Python gibt es jedoch ein OS-Modul, welches Sie die Befehlszeile steuern und Instruktionen der Befehlszeile aus einem Programm heraus ausführen lässt. Dadurch können Sie ein Programm schreiben, das durch Pygame gesteuert werden kann, was bedeutet, dass Sie den Roboter mit Ihrem Keyboard kontrollieren können.

13 Python OS verwenden

Öffnen Sie einen Python-Editor und importieren Sie mit „import os“ das OS-Modul. Geben Sie anschließend „os.system(„irsend SEND_ONCE Robot KEY_RIGHT“)“ ein. Dies erlaubt dem Python-Code Zugriff auf das Betriebssystem des Raspberry Pi. Die Codezeile sendet den „nach rechts bewegen“-Befehl zum IR-Board, welches das zugehörige Signal zum Roboter schickt, der sich daraufhin nach rechts bewegt. Überprüfen Sie die übrigen Bewegungsrichtungen, indem Sie die anderen Richtungsbezeichnungen am Ende des Codes einsetzen.

14 Machen Sie sich mit Pygame vertraut

Pygame ist eine plattformübergreifende Bibliothek, die zum Erstellen von einfachen Videospielen geschaffen wurde. Sie enthält eine Handvoll Grafik- und Soundbibliotheken, welche speziell für Python entwickelt wurden. Pygame ist auf dem Raspberry Pi vorinstalliert. Mehr Informationen zum Code und dem Erstellen von Spielen finden Sie in der offiziellen Dokumentation (pygame.org/docs).

15 Nicht überschreiben

Pygame läuft in einem Fenster, das vom User voreingestellt werden muss. Daher müssen Sie, bevor Sie Ihren Roboter mit Python steuern können, die Pygame-Struktur initialisieren. Auch wenn dieses Programm nur Ihr Keyboard verwendet, ist es notwendig, das klassische Pygame-Fenster zu erstellen. Sehen Sie sich den ersten Teil des Codes ab der ersten Zeile bis zu „runGame()“ an. Richten Sie die Dimensionen des Fensters zuerst ein (Zeile 7 und 8). Dann den Pygame-Clock: „FPSLOCK = pygame.time.Clock()“. Die Größe und Art der Schriftzeichen als Nächstes: „BASICFONT = pygame.font.Font(„freesansbold.ttf“, 18)“. Und zum Schluss die Beschriftung des Fensters: „pygame.display.set_caption(„ROBOT“)“.

16 Starten Sie LIRC neu

Für diesen Teil fahren Sie von „def runGame()“ nach unten fort. Sobald das Spiel initialisiert ist, wird der Code zur Robotersteuerung eingerichtet. Dieser verwendet das Pygame-Ereignis „get()“ (achten Sie auf das „for“ für die Schleife), um die gedrückte Taste zurückzuschicken. Das erste „elif“ veranlasst das Ereignis, auf einen Tastendruck zu warten, in unserem Beispiel „right“ in der nächsten Zeile. Da das OS importiert worden ist, verwenden wir „os.system(„irsend SEND_ONCE Robot KEY_RIGHT“)“, um das „nach rechts“-Signal zu senden. Vergewissern Sie sich, dass Sie das Pygame-Fenster ausgewählt haben, solange das Programm läuft, da dieses mit 100 x 100 Pixeln sehr klein ausfällt. Läuft alles, so fügen Sie die restlichen Bewegungen hinzu.

17 Noch mehr

Jetzt wo Sie mit Python Ihren Roboter steuern können, können Sie diesen mit einer Vielzahl von Interfaces ausstatten. Mit einem Makey Makey (makeymakey.com) steuern Sie Ihren Roboter mit Löffel und Gabel. Mit dem Tweepy-Modul (einer Twitter-API) reagiert Ihr Roboter sogar auf Tweets.

Vollständiger Code

```
import random, pygame, sys
from pygame.locals import *
import os
import time

FPS = 15
WINDOWWIDTH = 100
WINDOWHEIGHT = 100

UP = 'up'
DOWN = 'down'
LEFT = 'left'
RIGHT = 'right'

def main():
    global FPSLOCK, DISPLAYSURF, BASICFONT

    pygame.init()
    FPSLOCK = pygame.time.Clock()
    DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
    BASICFONT = pygame.font.Font('freesansbold.ttf', 18)
    pygame.display.set_caption('ROBOT')

    while True:
        runGame()

def runGame():
    # Set a random start point.
    while True: # main game loop
        for event in pygame.event.get(): # event handling loop
            if event.type == QUIT:
                terminate()

            elif event.type == KEYDOWN:
                if event.key == K_RIGHT:
                    print "right"
                    os.system("irsend SEND_ONCE Robot KEY_RIGHT")

                elif event.key == K_LEFT:
                    print "left"
                    os.system("irsend SEND_ONCE Robot KEY_LEFT")

                elif event.key == K_UP:
                    print "Forward"
                    os.system("irsend SEND_ONCE Robot KEY_UP")

                elif event.key == K_DOWN:
                    print "Back"
                    os.system("irsend SEND_ONCE Robot KEY_DOWN")

                elif event.key == K_SPACE:
                    print "STOP"
                    os.system("irsend SEND_ONCE Robot KEY_STOP")

                elif event.key == K_d:
                    print "Dance Baby"
                    os.system("irsend SEND_ONCE Robot KEY_D")

                elif event.key == K_y:
                    print "Say Yes"
                    os.system("irsend SEND_ONCE Robot KEY_R")

                elif event.key == K_l:
                    print "Say Yes"
                    os.system("irsend SEND_ONCE Robot KEY_L")

                elif event.key == K_n:
                    print "Say No"
                    os.system("irsend SEND_ONCE Robot KEY_P")

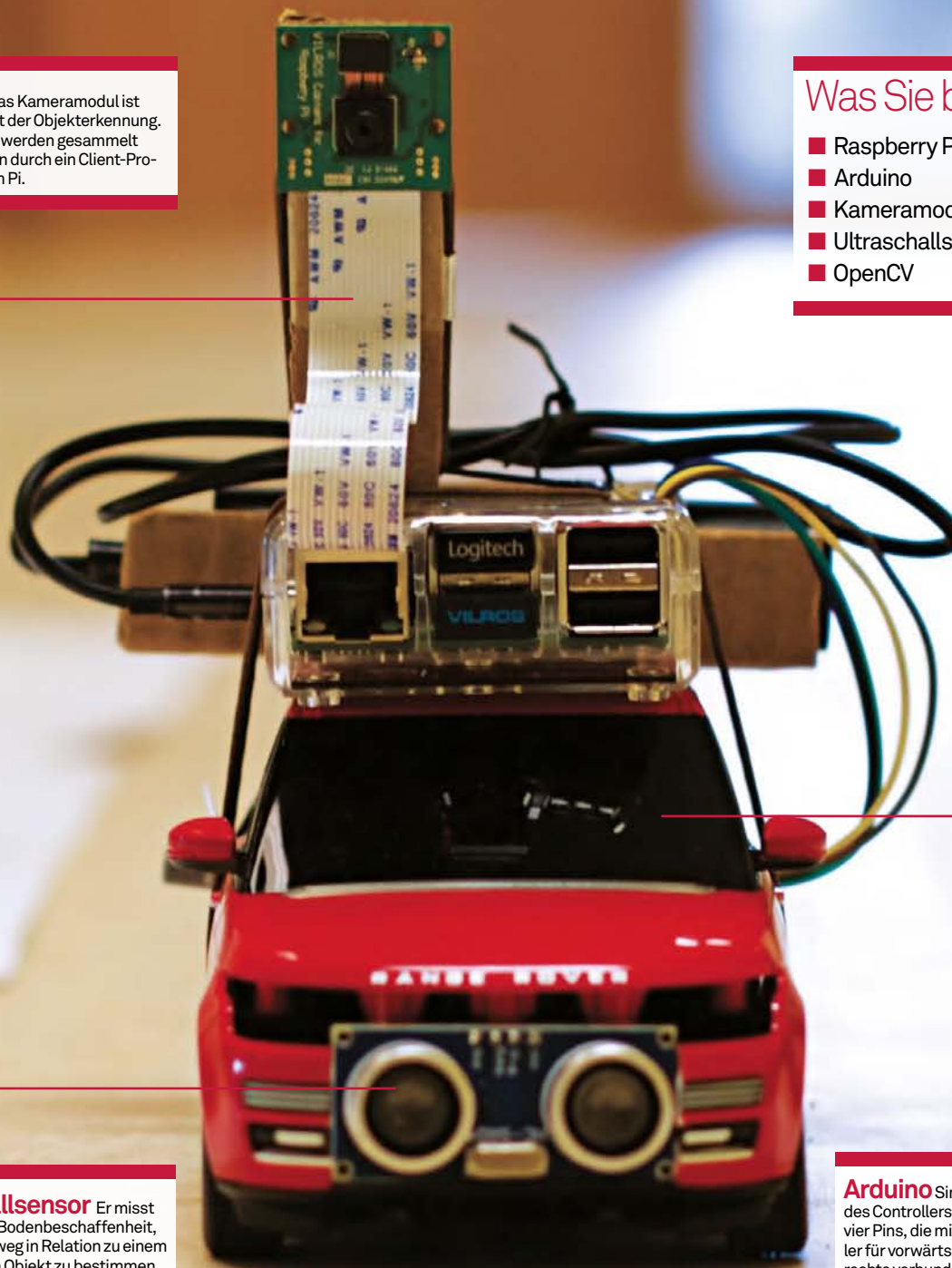
                elif event.key == K_ESCAPE:
                    terminate()

if __name__ == '__main__':
    main()
```

Kamera Das Kameramodul ist der Mittelpunkt der Objekterkennung. Die Inputdaten werden gesammelt und laufen dann durch ein Client-Programm auf dem Pi.

Was Sie brauchen

- Raspberry Pi B+
- Arduino
- Kameramodul
- Ultraschallsensor HC-SR04
- OpenCV

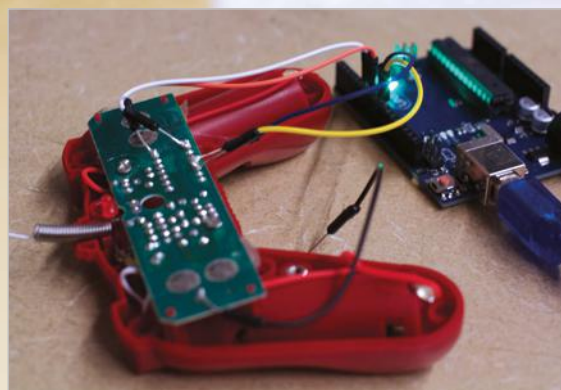


Ultraschallsensor Er misst Winkel und die Bodenbeschaffenheit, um den Bremsweg in Relation zu einem sich nähernden Objekt zu bestimmen.

Arduino Simuliert den Knopfdruck des Controllers des RC-Autos. Es gibt vier Pins, die mit Pins auf dem Controller für vorwärts, rückwärts, links und rechts verbunden sind.

Rechts Sobald ein Objekt entdeckt wird, überträgt der Ultraschallsensor die Infos und bringt so das RC-Auto zum Stehen.

Ganz rechts Das Arduino-Board simuliert den Knopfdruck, damit das RC-Auto selbstständig fahren kann.





Selbstfahrendes RC-Auto

Zheng Wang bietet Google mit seinem voll funktionsfähigen selbstfahrenden Auto die Stirn.

Woher kam die Idee zu einem selbstfahrenden Auto?

Ich habe es für ein Schulprojekt gebaut. Viele meiner Interessen drehen sich um maschinelles Lernen, also wollte ich unbedingt etwas in diese Richtung machen und mich mit den dahinterstehenden Konzepten auseinandersetzen. Ich habe online recherchiert, und das Projekt von David Singleton hat mich inspiriert, weil er so viel mit nur einem Arduino-Board und einigen anderen Dingen erreichen konnte. Es hat mich beeindruckt, dass das RC-Auto eine Strecke alleine fahren konnte, und ich habe mich gefragt, ob ich nicht etwas Ähnliches bauen könnte.

Danach habe ich mich mit meinem Raspberry Pi hingesetzt und überlegt, wie ich ein selbstfahrendes RC-Auto entwickeln könnte, das sogar noch mehr kann. Mein Ziel war es, Dinge wie Frontalkollisionen zu vermeiden und Stoppschilder oder Ampeln zu erkennen. Es dauerte seine Zeit, bis das Projekt mehr als eine Idee wurde, weil so viele Faktoren berücksichtigt werden mussten.

Wie funktioniert das selbstfahrende Auto?

Das Herzstück des Systems besteht aus drei Subsystemen, die nahtlos und synchron zusammenarbeiten. Diese Systeme bestehen aus einer Eingabeeinheit zur Steuerung der Kamera und des Ultraschallsensors, einer Verarbeitungseinheit und der Hauptkontrolleinheit für das RC-Auto.

Zunächst werden Live-Videodaten und Daten des Ultraschallsensors mittels einer starken WLAN-Verbindung vom Pi direkt auf den PC gestreamt. Ich habe schnell erkannt, dass es unerlässlich ist, so wenig Latenz wie möglich beim Streaming zu haben. Um das zu erreichen, musste die Auflösung drastisch auf

QVGA (320 x 240) gesenkt werden. Dies garantiert die flüssige Streamingqualität, die ich mir vorstellte. Danach müssen die Farbbilder, die auf dem PC ankommen, in Graustufen umgewandelt und in das neuronale Netz gespeist werden, um Prognosen für das Auto aufzustellen: ob es geradeaus oder im richtigen Moment nach links oder rechts fahren soll. Dieselben Bilder werden verwendet, um den korrekten Bremsweg zwischen Auto und Stoppschild zu berechnen, während der Pi das System vor einem nahenden Objekt warnt. Die Objekterkennung ist hauptsächlich ein Lernprozess.

Der letzte Teil des Systems besteht aus Outputs des künstlichen neuronalen Netzes, die per USB an den Arduino übermittelt werden, der wiederum direkt an den RC-Controller angeschlossen ist. Der Arduino liest die Befehle und gibt LOW- oder HIGH-Signale aus, die das Drücken der Knöpfe simulieren, die das Auto bewegen. Bei so vielen involvierten Sensoren und Datenfeeds musste ich viel herumprobieren, aber es hat nicht unverhältnismäßig lange gedauert, bis das Projekt eigenständig funktionierte.

Welche Rolle hat der Raspberry Pi für das selbstfahrende Auto gespielt?

Der größte Vorteil am Pi ist, dass er die perfekte Apparatur ist, um Input-Daten zu sammeln, die einen großen Teil dieses Projekts ausmachen. An den Raspberry Pi habe ich dann ein Pi-Kameramodul und einen Ultraschallsensor angeschlossen, die dem Pi zusammen bei der Datensammlung helfen.

Außerdem laufen zwei Client-Programme auf dem Pi, die beim Streaming hilfreich sind. Eines ist für das Videostreaming zuständig, das

andere für das Datenstreaming des Ultraschallsensors. Um ehrlich zu sein, habe ich mich da an die offizielle Pi-Kamera-Dokumentation gehalten, da es schon viele Richtlinien für Videostreaming gibt. Wenn ich Hilfe beim Messen der Distanz mit dem Ultraschallsensor brauchte, gab es nützliche Tutorials von Enthusiasten.

Können Sie uns mehr über den Ultraschallsensor erzählen? Kann er Kollisionen in einem Winkel von 360° erkennen?

Ich habe mich für den HC-SR04-Ultraschallsensor entschieden, weil er zu den kostengünstigsten und benutzerfreundlichsten Modellen auf dem Markt gehört. Er ist etwas knifflig, wenn man bei Null anfängt, aber wie schon gesagt, konnte ich mir bei jedem Problem Hilfe im Internet holen. In der Bedienungsanleitung dieses Sensors steht, dass die beste Erkennung in einem Winkel von 30° funktioniert, was mir anhand der Tests, die ich durchführte, richtig erschien. Es gibt zahlreiche Sensoren auf dem Markt, also erscheint mir eine komplette 360°-Erkennung durchaus plausibel.

Werden Sie das Projekt fortführen, vielleicht ein größeres Modell bauen?

Es gibt viele Bereiche, die ich erforschen möchte, um mein Projekt weiter voranzutreiben. Ich möchte auf jeden Fall den Ultraschallsensor durch eine Stereokamera ersetzen, um die Entfernungen zu messen. Die Resultate werden damit um einiges präziser sein. Wenn ich mehr freie Zeit zur Verfügung hätte, würde ich versuchen, weitere Verhaltensmerkmale zu integrieren. Es wäre faszinierend, zu sehen, ob ich Dinge wie Spurwechsel und Überholvorgänge einbauen könnte.

Motion Tracking per Pi

So können Sie mit dem Raspberry Pi, einer Kamera und etwas Python-Code Bewegungen erfassen.

In dieser Anleitung möchten wir Ihnen zeigen, wie man mit dem Raspberry Pi eine Bewegungserkennung ausführen kann. Diese Art fortgeschrittener Bildverarbeitung ist allerdings eine komplexe Angelegenheit, weswegen wir auf jedem Fall auf der Arbeit anderer aufbauen müssen. Hierzu nutzen wir das ausgezeichnete Python-Paket OpenCV. Dieses Paket wird kontinuierlich verbessert, mit jedem Update werden neue Funktionen hinzugefügt. Zuerst installieren Sie die verschiedenen Python-Pakete, die benötigt werden, um mit der Kamera kommunizieren und OpenCV nutzen zu können. So werden die Pakete installiert:

```
sudo apt-get install python-piamera python-opencv
```

Dadurch werden auch die nötigen Abhängigkeiten installiert. Das Projekt geht davon aus, dass das Kamera-Modul für den Raspberry Pi genutzt wird (auf der nächsten Seite finden Sie aber auch eine Anleitung für USB-Kameras). Um mit dem Kamera-Modul zu kommunizieren, importieren Sie die PiCamera-Klasse aus dem piamera-Python-Modul. Zudem benötigen Sie die PiRGBArray-Klasse, damit Sie die Rohdaten der Kamera speichern können. Erstellen Sie eine neue Instanz, damit Sie mit der Kamera „reden“ können. Bevor Sie Bilder aufzeichnen, haben Sie dann noch die Möglichkeit, die Auflösung und Bildrate einzustellen.

```
from picamera import PiCamera
from picamera import PiRGBArray
camera = PiCamera()
camera.resolution = tuple([640,480])
camera.framerate = 16
rawImage = PiRGBArray(camera, tuple([640,480]))
```

Nun ist Ihre Kamera bereit und ein Buffer im Speicher verfügbar, in dem die Bilder gespeichert werden. Es gibt verschiedene Methoden, Bewegungen zu erfassen. Eine einfache ist das Erkennen von Veränderungen innerhalb des Sichtfeldes. Das Python-Modul imutils verfügt

über einige grundlegende Bildverarbeitungsfunktionen, die bei den Preprocessing-Schritten nützlich sind. Das Paket ist nicht in Raspbian enthalten, weswegen Sie es zuerst mit diesem Befehl installieren müssen:

```
sudo pip install imutils
```

Um Bildveränderungen wahrzunehmen, müssen wir den Hintergrund betrachten. Sie können dafür eine Serie an Bildern aufzeichnen und diese überblicken, um einen Eindruck des durchschnittlichen Hintergrundes zu bekommen. Wenn sich dann ein Bild von diesem „Durchschnitt“ abhebt, wissen wir, dass sich etwas verändert hat. Dies liegt dann höchstwahrscheinlich daran, dass sich im Sichtfeld ein bisschen bewegt hat. Um den Prozess zu vereinfachen, wandeln wir das Bild in Graustufen um und verwischen es so, dass kontrastreiche Bereiche beseitigt werden. Starten Sie dazu eine Schleife, die ein Bild von der Kamera empfängt und diesen Prozess durchführt:

```
import imutils
import cv2
for f in camera.capture_continuous(rawImage, format='bgr', use_video_port=True):
    frame = imutils.resize(f.array, width=500)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray, (21, 21), 0)
```

Jetzt starten wir die für die Bildverarbeitung nötigen OpenCV-Funktionen. Sie werden bemerkt haben, dass wir bereits an der Reihendarstellung der Rohdaten des aufgezeichneten Frames arbeiten. Die Bilder werden nicht mit Metadaten ausgegeben, also liegt es in Ihrer Verantwortung, sich daran zu erinnern, womit Sie arbeiten. Der nächste Schritt innerhalb dieser Schleife ist, herauszufinden, ob wir bereits einen durchschnittlichen Hintergrund haben und initialisieren können oder nicht. Erstmals wird der folgende Code in der Schleife ausgeführt:

```
if avg is None:
    avg = gray.copy().astype("float")
    rawImage.truncate(0)
    continue
```

Mit unserem gewichteten Durchschnittsbild können wir jetzt jedes folgende Bild vergleichen. Wir müssen zudem herausfinden, wie stark sich das Bild vom Durchschnittsbild unterscheidet.

```
cv2.accumulateWeighted(gray, avg, 0.5)
imgDiff = cv2.absdiff(gray, cv2.convertScaleAbs(avg))
```

Dank des gewichteten Durchschnittsbildes sollten wir jetzt falsche positive Treffer aufgrund von Veränderungen in der Umgebung erhalten – wie etwa bei Schwankungen der Lichtverhältnisse. Wie werden die jetzt ersichtlichen Unterschiede zum Durchschnitt eingesetzt? Wie entscheiden Sie, wie sehr sich die Bilder vom Durchschnitt unterscheiden? Hierfür müssen wir einen Grenzbereich festlegen, der „echte“ Veränderungen auf dem Bild im Vergleich mit dem Durchschnittsbild ersichtlich macht. Wenn Sie das Bild mit dem Grenzwert erweitern, können Sie die findContours-Funktion nutzen, um die Umrisse von Objekten zu erkennen, die sich vom errechneten durchschnittlichen Hintergrund abheben:

```
imgThresh = cv2.threshold(imgDiff, 5, 255, cv2.THRESH_BINARY)[1]
imgThresh = cv2.dilate(imgThresh, None, iterations=2)
(conts, _) = cv2.findContours(
    (imgThresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```

Dadurch werden alle Konturen des aktuellen Bildes in der Liste „conts“ abgelegt. Sie interessieren sich wahrscheinlich nicht für winzige Objekte in der Liste mit den Konturen, vor allem nicht, wenn es sich dabei nur um Artefakte innerhalb der Bilddaten handelt. Sie sollten alle Bilder checken und diejenigen ignorieren, die unterhalb eines Grenzbereichs liegen. Sie können sämtliche verbliebene Konturen



Wir nutzen das kontinuierlich verbesserte OpenCV.

hervorheben, indem Sie Rahmen darum legen. OpenCV bietet hierfür eine Funktion, die die Eckkoordinaten sowie die Breite und Höhe anzeigt. Dann können Sie mittels folgender Funktion einen Rahmen auf dem Bild platzieren:

```
for c in conts:
    if cv2.contourArea(c) < 5000:
        continue
    (x, y, w, h) = cv2.boundingRect(c)
    cv2.rectangle(frame, (x, y),
                  (x+w, y+h), (0, 255, 0), 2)
```

Sie sollten jetzt ein Bild sehen, auf dem alle beweglichen Objekte durch rote Rahmen hervorgehoben werden. Doch was fangen Sie nun mit diesen „kommentierten“ Bildern an? Wenn Sie eine grafische Umgebung haben, können Sie die Ergebnisse direkt auf dem Display anzeigen lassen. OpenCV bietet verschiedene Möglichkeiten, die Ergebnisse der Bildanalyse zu zeigen. Die einfachste darunter ist `imshow()`, die das Bild öffnet, anzeigt und es mit einem Namen versieht.

```
cv2.imshow("Motion detected",
           frame)
```

Wenn Sie sich die Ergebnisse Ihrer Bewegungserkennung nicht in Echtzeit anzeigen lassen, möchten Sie vielleicht dennoch Bilder aufzeichnen, sobald sich etwas in der Umgebung bewegt. OpenCV bietet hierfür viele verschiedene IO-Funktionen. Zuerst sollten Sie die Bilder mit einer Zeitanzeige versehen. Mit dem Python-Modul `timestamp` und der Funktion „`putText()`“ können Sie dem Bild das aktuelle Datum und die Uhrzeit hinzufügen:

```
import timestamp
ts = timestamp.strftime("%A %d
%B %Y %I:%M:%S%p")
cv2.putText(frame, ts, (10, frame.
shape[0] - 10), cv2.FONT_HERSHEY_
SIMPLEX, 0.35, (0, 0, 255), 1)
```

Now you have an image with the current time and date on it, and the parts of the image that show up as having movement

bounded in red boxes. You can use the OpenCV IO functions to write out these images so that you can check them out later. The following code is an example:

```
cv2.imwrite("filename.jpg", frame))
```

Die Funktion `imwrite()` nutzt die Dateinamen-Endung, um das Format des zu speichernden Bildes herauszufinden, und kann mit JPEG, PNG, PBM, PGM, PPM und TIFF arbeiten. Wenn Sie in einem bestimmten Format mit unterschiedlichen Optionen speichern möchten, können Sie dies dem Befehl hinzufügen. Zum Beispiel können Sie die JPEG-Qualität mit der Zeile `CV_IMWRITE_JPEG_QUALITY` plus einem Wert zwischen 0 und 100 festlegen.

Bislang lag der Fokus auf dem Gedanken, Bilder in Echtzeit zu analysieren, was nur dann gut ist, wenn sich der Raspberry Pi am selben Ort wie die Kamera befindet. Wenn das nicht möglich ist, können Sie Ihr Wissen trotzdem dazu nutzen, das Video nachzubearbeiten, das die Kamera aufgenommen hat. Laden Sie die Videodatei mit denselben OpenCV-IO-Funktionen:

```
camera = cv2.
VideoCapture("filename.avi")
```

Jetzt können Sie denselben Vorgang wiederholen, um jedes Bild innerhalb des Videos zu analysieren. Die `VideoCapture()`-Funktion liest eine Serie von Bildern, wenn die Kamera anstatt eines Videos Einzelbilder aufnimmt. Nachdem das Programm beendet ist, sollten Sie noch kurz aufräumen. Entfernen Sie die benutzte Kamera und etwaige Bilder, die Sie auf dem Desktop haben anzeigen lassen.

```
camera.release()
cv2.destroyAllWindows()
```

Sie haben nun das Wissen, um einfache Bewegungserkennung zu Ihrem Python-Programm hinzuzufügen. In der OpenCV-Dokumentation finden Sie viele weitere und auch komplexe Bildverarbeitungs- und Analyse-Tools zum Ausprobieren.

Und Webcams?

Im Hauptartikel haben wir das Raspberry-Pi-Kameramodul via IO-Anschluss genutzt. Aber was tun, wenn das Modul nicht zur Hand ist? Dann hilft eine gewöhnliche alte Webcam, die wie der Raspberry Pi über eine USB-Schnittstelle verfügt. Die Qualität der Bilder und die Bildrate sind dann allerdings nicht ganz so gut wie mit dem aktuellen Pi-Modul. Der Schlüssel liegt darin, die Bilddaten von der Kamera in das Format zu bringen, das die OpenCV-Funktionen benötigen. Die `VideoCapture()`-Funktion kann dabei nicht nur Videodateien bearbeiten, sondern auch Geräte-IDs von Kameras, die mit dem Raspberry Pi verbunden sind. Wenn Sie nur eine Kamera anschließen, geht dies mit folgendem Befehl:

```
camera = cv2.VideoCapture(0)
```

Auch wenn OpenCV einige Möglichkeiten der Interaktion mit dem Benutzer bietet, können Sie auch ein anderes Framework verwenden, zum Beispiel das sehr schnelle `pygame`. Lassen Sie OpenCV die Bildverarbeitung übernehmen und bauen Sie Ihr eigenes Benutzer-Interface mit `pygame`. Das einzige Problem hierbei sind die unterschiedlichen internen Formate, die OpenCV und `pygame` nutzen, um Bilddaten zu speichern. Deshalb müssen die Daten in beide Richtungen übersetzt werden, wobei Sie sich nur um die Übersetzung von OpenCV zu `pygame` kümmern müssen, da die Daten nur in diese Richtung fließen. Es gibt einige Hilfsfunktionen, mit denen Sie OpenCV-Bilder in ein Zeichenkettenformat umwandeln, und außerdem eine Python-Funktion, um die Zeichenketten in ein Python-Bild zu importieren. Beispielsweise können Sie diesen Befehl nutzen:

```
pygameImg = pygame.image.frombuffer(cv2Img.
tostring(), cv2Img.shape[1::-1], "RGB")
```

So werden Bilder von einem OpenCV-Format (gespeichert in `cv2Img`) in ein `pygame`-Format (gespeichert in `pygameImg`) umgewandelt. Bei Bedarf können Sie die Daten auf ähnliche Weise mittels Zeichenketten von `pygame` ins OpenCV-Format übersetzen.

Erstellen Sie einen Pi-Verbund

Kombinieren Sie Leistung und Ressourcen mehrerer Raspberry Pis, indem Sie mithilfe der Docker-Software einen Verbund erstellen.

Docker ist ein System und Werkzeug, um unter Linux mehrere Container zu errichten, zu konfigurieren und einzusetzen. Container ermöglichen es, eine Anwendung und all ihre Abhängigkeiten zu einer Einheit zusammenzufassen. Somit lässt sie sich einfacher und platzsparender versenden.

Jede Anwendung läuft in ihrer eigenen Umgebung und teilt sich Kernel und Leistung mit dem Host – ganz im Gegensatz zu einer virtuellen Maschine, welche ihr eigenes Betriebssystem benötigt. Ein Docker-Container kann innerhalb einer Sekunde gestartet oder beendet werden, wobei er riesige Ausmaße annehmen kann, ohne den Host und seine Kapazitäten übermäßig zu belasten.

Die Docker-Community hat eine Clustering-Lösung namens „Swarm“ erfunden, welche seit der Version 1.0 als serienreif bezeichnet wird. Wenn ein einzelner Raspberry Pi 1 GB RAM und 4 Prozessorkerne besitzt, haben wir mit fünf Exemplaren bereits 5 GB RAM und 20 Kerne zur Verfügung. „Swarm“ hilft uns dabei, die Auslastung aufzuteilen.

Bereiten Sie Arch Linux zur Installation vor, stellen Sie Docker 1.9.1 zusammen, erstellen Sie einige Images, und starten Sie „Swarm“ zum ersten Mal.

01 Arch Linux auf SD-Karte installieren

Öffnen Sie die Arch-Linux-Startseite für den Pi 2 und klicken Sie auf den Installationslink (bit.ly/1SyrGqU). Sie müssen ein paar Schritte mit einem Linux-Rechner vollziehen. Folgen Sie den Anweisungen, um das tar.gz-Archiv für das Basissystem herunterzuladen. Partitionieren Sie danach die Karte und erstellen Sie vfat-(Boot-) und ext4-(Root-)Dateisysteme. Extrahieren Sie das Basissystem auf die Karte. Werfen Sie zuletzt die Partitionen aus und warten Sie, bis die Synchronisierung abgeschlossen ist.

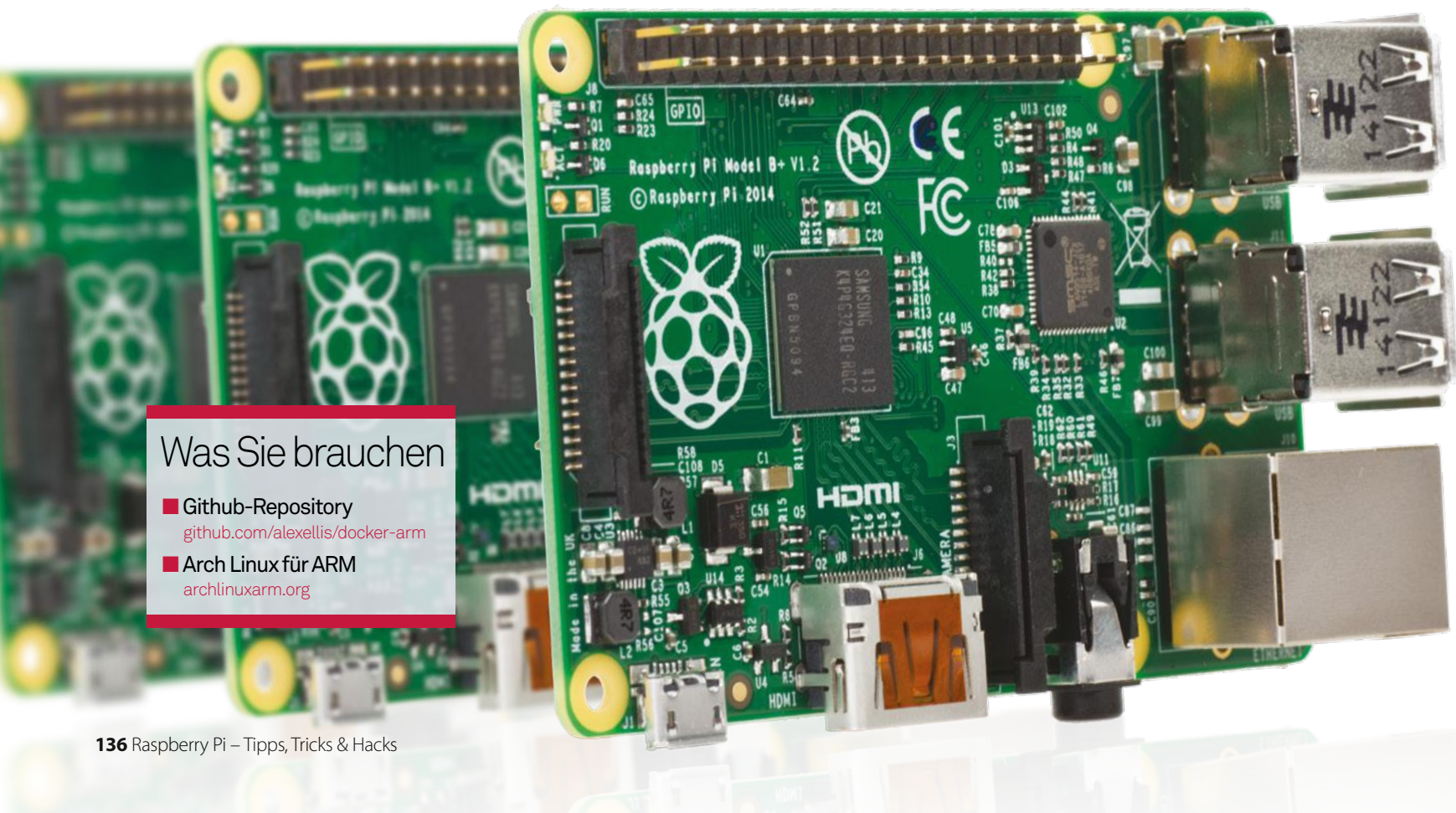
02 Benutzer konfigurieren

Wenn der Pi gestartet ist, können Sie sich mithilfe Ihrer Tastatur als root/root anmelden und anschließend Ihr Passwort festlegen. Auch das Standard-Benutzerkonto „alarm“ können Sie getrost löschen und ein eigenes anlegen. In unserem Fall haben wir ein Konto namens „lud“ eingerichtet:

```
# passwd root
# useradd lud -m -s /bin/bash -G wheel
# passwd lud
# userdel alarm
```

Was Sie brauchen

- Github-Repository
github.com/alexellis/docker-arm
- Arch Linux für ARM
archlinuxarm.org






[About](#) | [Platforms](#) | [Packages](#) | [Forum](#) | [Support](#) | [Developers](#) | [Donate](#)

Raspberry Pi 2

[Overview](#) | [Installation](#)

The Raspberry Pi 2 is the successor to the Raspberry Pi. It builds upon the original model B+ upgrading to 1 GB of RAM, and replacing the aged ARMv6l single-core with an ARMv7l Cortex-A7 quad-core.

The Raspberry Pi 2 measures 85.60mm x 53.98mm x 17mm, with a little overlap for the SD card and connectors which project over the edges. The SoC is a Broadcom BCM2836. This contains an quad-core Cortex-A7 running at 900Mhz, and a Videocore 4 GPU. The GPU is capable of BluRay quality playback, using H.264 at 40MBits/s.



Architecture

ARMv7l Cortex-A7

Processor

Broadcom BCM2836
900MHz

RAM

1024MB

SD

Micro SD

USB

4

Links Arch Linux ist eine gute Wahl für alle Projekte, die eine Softwarelösung auf dem neuesten Stand und ohne unnützen Ballast erfordern.

03 Feste IP-Adresse einrichten

Richten Sie sich nun eine feste IP-Adresse ein, damit Sie sich ohne Rätselraten mit jedem Pi verbinden können. Das Betriebssystem nutzt systemd zur Konfiguration. Bearbeiten Sie die Konfigurationsdatei `/etc/systemd/network/eth0.network` und starten Sie das Gerät neu.

```
[Match]
Name=eth0

[Network]
Address=192.168.0.200/24
Gateway=192.168.0.1
DNS=8.8.8.8
IPForward=ipv4
```

Wenn Sie lieber an einem Laptop oder PC arbeiten, können Sie sich über SSH mit 192.168.0.200 verbinden. Unser Verbund besteht aus fünf Knotenpunkten, deshalb sind alle Adressen zwischen 192.168.0.200 und 205 vergeben.

04 Tools und Dienstprogramme installieren

Arch Linux wird ständig weiterentwickelt, deshalb gibt es fortlaufend Updates. Wir werden nun mit dem Paketmanager Pacman gleichzeitig einige wichtige Bestandteile und Upgrades installieren:

```
# pacman -Syu --noconfirm base-devel wget git
sudo screen bridge-utils device-mapper apache
```

05 Sudo aktivieren

Erteilen Sie Ihrem neuen Benutzerkonto Zugang zu sudo, indem Sie in der `/etc/sudoers`-Liste folgende Zeile modifizieren:

```
## Same thing without a password
# %wheel ALL=(ALL) NOPASSWD: ALL
```

Dadurch erhalten alle Benutzer der „wheel“-Gruppe Zugang zu sudo. In Schritt 02 hatten wir „wheel“ als primäre Gruppe unseres neuen Nutzers festgelegt.

Arch Linux wird ständig weiterentwickelt, deshalb gibt es fortlaufend Updates.

06 Git-Repository kopieren

Wir haben ein paar wichtige Skripte, Konfigurationen und eine Version des Docker-Swarms für ARM in einem Git-Repository zusammengestellt. Melden Sie sich unter Ihrem Benutzerkonto an und kopieren Sie das Repository von Github in Ihr Verzeichnis:

```
# cd ~
# git clone http://github.com/alexellis/docker-arm/
```

07 Docker 1.7.1 installieren

Im Arch-Linux-Paketssystem befindet sich bereits die Version Docker 1.9.1, die jedoch noch nicht funktioniert. Deshalb greifen wir auf die zuletzt lauffähige Version zurück und fügen diese mithilfe der offiziellen Skripte zusammen:

```
# sudo pacman -U ~/docker-arm/pkg/docker-1:1.7.1-2-
armv7h.pkg.tar.xz --noconfirm
# sudo cp ~/docker-arm/pkg/docker.service /usr/
lib/systemd/system/docker.service
# sudo systemctl enable docker
# sudo systemctl start docker
# sudo usermod lud -aG docker
# sudo reboot
```

Melden Sie sich nun erneut an und prüfen Sie, ob es funktioniert hat:

```
# docker info
```

Nun werden wir in der `/etc/pacman.conf` veranlassen, dass unsere Änderungen nicht durch Systemupdates überschrieben werden:

```
# sudo ~/docker-arm/pkg/ignore_docker_package.sh
```

Arch Linux ARM

Einer der Vorteile von Arch Linux ist, dass das System sehr minimalistisch ausgestattet ist. Somit haben Sie die vollständige Kontrolle darüber, welche Bestandteile Sie benötigen. Ihr Gerät wird viel schneller hochfahren als mit Raspbian, welches eine breitere Zielgruppe hat. Das System basiert mittels Pacman auf einer kontinuierlichen Weiterentwicklung und bleibt dadurch auf dem aktuellen Stand der Entwickler-Community. Durch diese Weiterentwicklungen haben Sie allerdings immer nur die Möglichkeit, auf die aktuellste Package-Version zuzugreifen.

Rohmaterial

Wir konzentrieren uns hier auf die Aufteilung einer Internetanwendung innerhalb eines Verbunds. Der Raspberry Pi ist jedoch auch in der Lage, mit weiterer Hardware zu arbeiten und Messungen durchzuführen. Der Pi besitzt 4 USB-Anschlüsse, 42 GPIO-Pins, Audio-Ausgang und ein Kamera-Interface. Sie haben also alle Zutaten, um etwas Einzigartiges zu erschaffen. Könnte man das express-redis4.x-Image auch erweitern, sodass eine LED leuchtet, sobald eine Anfrage bearbeitet wird?

08 Docker auf der Basis von Docker erstellen!

Wir haben jetzt die funktionstüchtige Version, die wir kompilieren müssen:

```
# cd ~/docker-arm/images/docker-arm
# ./build.sh
```

In den nächsten 30–60 Minuten werden ein Docker-Entwickler-Image sowie ein Code erstellt und für ARM gepatcht. Das Ganze wird dann lokal auf Ihrem Pi gespeichert. Wenn das Skript fertig ist, bekommen Sie folgende Nachricht:

```
*Created binary: bundles/1.9.1/binary/docker-1.9.1*
```

Wenn alles geklappt hat, können Sie die Änderungen speichern:

```
# cd ~/docker-arm/images/docker-arm
# sudo ./install.sh
# sudo systemctl start docker
```

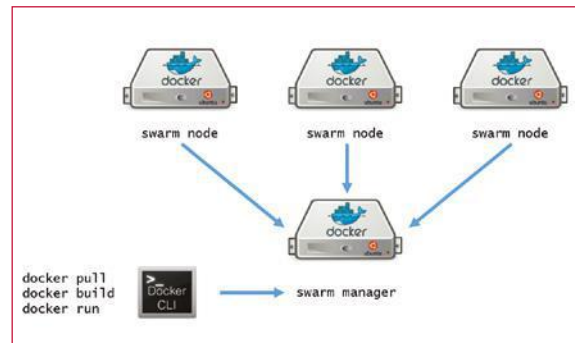
09 Docker-Swarm-Image erstellen

Das offizielle Swarm-Image ist für uns leider nicht nutzbar, da es für x86_64-Systeme, also normale PC-Systeme erstellt wurde. Deshalb konstruieren wir unser eigenes Image:

```
# cd ~/docker-arm/images/swarm-arm
# ./build.sh
# docker run alexellis2/swarm-arm --version
```

10 Zusätzliche Nodes

Sie können nun entweder Ihre SD-Karte duplizieren oder auf jedem Pi einzeln diese Routine durchlaufen. In beiden Fällen müssen Sie /etc/hostname und die IP-Adresse auf jedem Pi updaten. Wenn Sie sich für das Duplizieren entscheiden, müssen



Sie /etc/docker/key.json löschen, damit es innerhalb des Verbunds nicht zu Überschneidungen kommt. Es sind noch weitere Images verfügbar; Sie können sie je nach Bedarf bauen oder das build_all.sh-Skript verwenden (empfohlen).

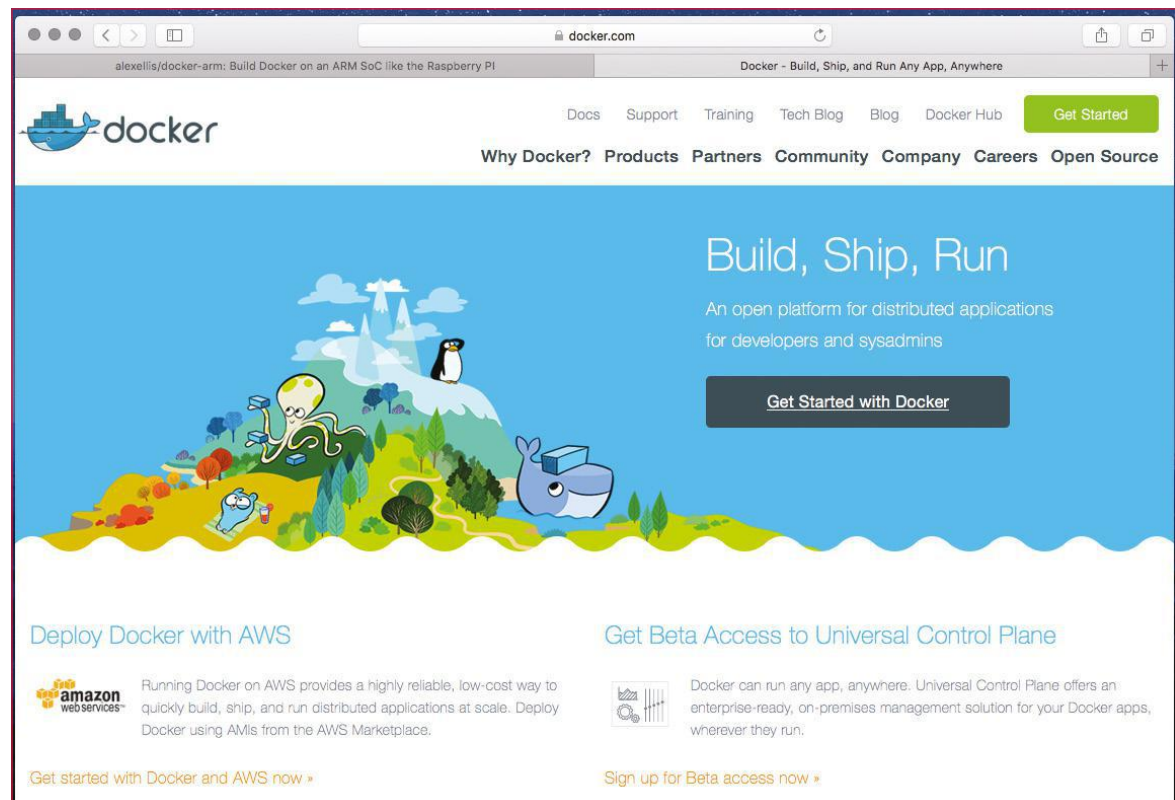
11 Mit dem ersten Node beginnen

Mit unserem ersten Node bestimmen wir, dass unser Verbund über Port 4000 organisiert wird und die Service Discovery per Consul über Port 8500 geleitet wird. Beide Dienste laufen lokal auf unserer Docker-Instanz.

```
# cd ~/docker-arm/images/consul-arm
# ./build.sh
```

```
# ~/docker-arm/script/start_consul.sh
# ~/docker-arm/script/manage_swarm.sh
```

Wenn Sie bereits einen Consul-Arm-Container erstellt haben, werden Sie feststellen, dass es jetzt viel schneller geht, denn Docker merkt sich die Schritte und nur die Veränderungen zwischen den Builds werden neu erstellt.



Rechts Docker hat sich schnell zum Industriestandard der Container-Technologie entwickelt.



12 Swarm beitreten

Verbinden Sie sich mit einem Node, etwa 192.168.0.201, und starten Sie das `auto_join_swarm.sh`-Skript. Dies fragt die IP von `eth0` ab, und das Ergebnis wird an Consul und den Swarm-Manager weitergeleitet.

```
# ~/docker-arm/script/auto_join_swarm.sh
```

Sie können dem Swarm-Agent jetzt unter `docker.ps` bei der Arbeit zusehen. Tippen Sie „`docker logs join`“, um die Ergebnisse zu sehen, und wiederholen Sie diesen Schritt für jeden einzelnen Node.

```
alex@docker1:~$ docker run alexellis2/swarm-arm list consul://192.168.0.200:8500/swarm
time="2016-01-25T21:19:22Z" level=info msg="initializing discovery without TLS"
192.168.0.201:2375
192.168.0.201:2375
192.168.0.202:2375
192.168.0.203:2375
192.168.0.204:2375
192.168.0.210:2375
192.168.0.211:2375
alex@docker1:~$ export DOCKER_HOST=tcp://192.168.0.200:4000
alex@docker1:~$ script docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
910e178023dd        alexellis2/swarm-arm:latest   "/swarm join --advert"   About a minute ago   Up About a minute
86d6402c250d        alexellis2/swarm-arm:latest   "/swarm join --advert"   About a minute ago   Up About a minute
80f08d77f4c         alexellis2/swarm-arm:latest   "/swarm join --advert"   About a minute ago   Up About a minute
81d09c22a538        alexellis2/swarm-arm:latest   "/swarm join --advert"   About a minute ago   Up About a minute
6847fcd9baa         alexellis2/swarm-arm:latest   "/swarm join --advert"   2 minutes ago        Up About a minute
7ba374d0bd1         alexellis2/swarm-arm:latest   "/swarm join --advert"   2 minutes ago        Up 2 minutes
alex@docker1:~$ script
```

13 Abfragen von Swarm

Loggen Sie sich im ersten Node ein und starten Sie das `Swarm-Arm-Image` an der Adresse des `Consul-Services`:

```
# docker run alexellis2/swarm-arm list
consul://192.168.0.200:8500/swarm
192.168.0.201:2375
192.168.0.202:2375
192.168.0.203:2375
192.168.0.204:2375
192.168.0.210:2375
```

Wenn Sie den Docker-Befehl mit dem Swarm nutzen, stellen Sie die `DOCKER_HOST`-Variable auf die Adresse des Swarm-Managers:

```
# export DOCKER_HOST=tcp://192.168.0.200:4000
```

Nun können Sie sehen, wie viele gebündelte Ressourcen wir haben:

```
# docker info
...
Nodes: 4
...
CPUs: 20
Total Memory: 3.785 GiB
```

14 Beispiel: Verteilte webbasierte Anwendung

Erstellen wir nun eine verteilte webbasierte Anwendung, welche in einer Redis-Datenbank jedes Mal einen Zähler erhöht, wenn er aktiviert wird. Wir werden hiervon mehrere Instanzen starten und eine Nginx-Load-Balance vorschalten. Wir können auch die Apache Bench nehmen, um Messgrößen zu generieren.

Diese Container müssen in der richtigen Reihenfolge gestartet werden, beginnend mit Redis, dann Node und letztlich Nginx.

15 Starten der Redis- und Node-Container

Rufen Sie zuerst die Redis-Container auf und benennen Sie sie mit `redis_1` bis `redis_5`:

```
# docker run -p 6379:6379 -d --name redis_1
alexellis2/redis-arm
```

Starten und verbinden Sie nun dieselbe Anzahl von `node.js`-Containern mit den Redis-Containern.

```
# docker run -p 3000:3000 -d \
--label='node_redis' \
--link redis_1:redis \
expressredis4.x
```

Zum Schluss starten Sie auf dem ersten Node den Load Balancer:

```
# DOCKER_HOST="" docker run -d --name=balancer
-p 80:80 nginx_dynamic
```

```
document Path: /
document Length: 31 bytes
Concurrency Level: 100
Time taken for tests: 13.252 seconds
Complete requests: 10000
Failed requests: 9819
  (Connect: 0, Receive: 0, Length: 9819, Exceptions: 0)
Total transferred: 2549819 bytes
HTML transferred: 319819 bytes
Requests per second: 748.94 [#/sec] (mean)
Time per request: 133.523 [ms] (mean)
Time per request: 1.335 [ms] (mean, across all concurrent requests)
Transfer rate: 186.49 [Kbytes/sec] received

Connection Times (ms)
  connect:    min     0.000  mean=0.000  max=0.000
processing:  min    132.161  mean=132.161 max=132.161
waiting:     min    132.161  mean=132.161 max=132.161
total:       min    132.161  mean=132.161 max=132.161

percentage of the requests served within a certain time (ms)
 50%    74
 66%   131
 75%   166
 80%   191
 90%   358
 95%   542
 98%   665
 99%   706
100%   820 (longest request)
alex@docker1:~$ ab -n 10000 -c 100 http://192.168.0.200/
```

16 Apache Bench starten

Nun lassen wir Apache Bench zeitgleich 10 Threads und 1.000 Anfragen ausführen. Wir haben unsere Anwendung auf sechs Swarm-Agenten nach der Aufstellung von zwei zusätzlichen Pis gestartet.

```
# ab -n 1000 -c 10 http://192.168.0.200/
...
Concurrency Level:      10
Time taken for tests:    2.593 seconds
Requests per second:    385.65 [#/sec] (mean)
...
```

Das gleiche Experiment mit einem einzelnen Pi ergab lediglich 88,06 Anfragen pro Sekunde und dauerte insgesamt 11,356 Sekunden. Man kann den Wert der gleichzeitigen Aufgaben (-c) auch auf 100 erhöhen.

17 Steuern Sie Ihren Swarm per PC

Wenn Sie aus der Liste der Programme des Docker-Clients die `DOCKER_HOST`-Variable nutzen und mit Ihrem Swarm-Manager verlinken, müssen Sie sich mit den Pis nicht per SSH verbinden. Diese Docker-Programme sind unter docs.docker.com/engine/installation/linux/docker-ce/binaries/ zu finden.

```
# wget https://get.docker.com/builds/Darwin/x86_64/
docker-1.9.1
# chmod +x docker-1.9.1
# export DOCKER_HOST=tcp://192.168.0.200:4000
# ./docker-1.9.1 info
```

18 Ausblick

Sie können die Schritte dieses Tutorials solange wiederholen, bis Sie genügend Swarm-Agenten in Ihrem Verbund haben. Einer der Vorzüge des Verbunds ist die Aufgabenverteilung zwischen den Nodes, aber Swarm erleichtert uns auch noch die Koordination innerhalb der Nodes. Um die Möglichkeiten noch weiter auszuschöpfen, könnte man auch einige Sensoren mittels GPIO-Pins verbinden und die Vorteile der Hardware-Fertigkeiten des Pi ausnutzen.

Docker Compose

Mit dem Tool Docker Compose können Sie eine YML-Datei auslesen und Container transparent verknüpfen. Dadurch erstellen Sie einen Webservice, der mehr als einen Container umfasst und weniger Tastenanschläge erfordert.

```
nodejs_1:
  image: node-
counter
ports:
  - "3000"
links:
  - redis_1
redis_1:
  image: redis
ports:
  - "6379"
nginx_1:
  image: nginx
links:
  - nodejs_1
ports:
  - "80:80"
```

VPN-Zugang per Raspberry Pi

Nutzen Sie den Pi als Wireless Access Point für permanente VPN-Verbindungen.

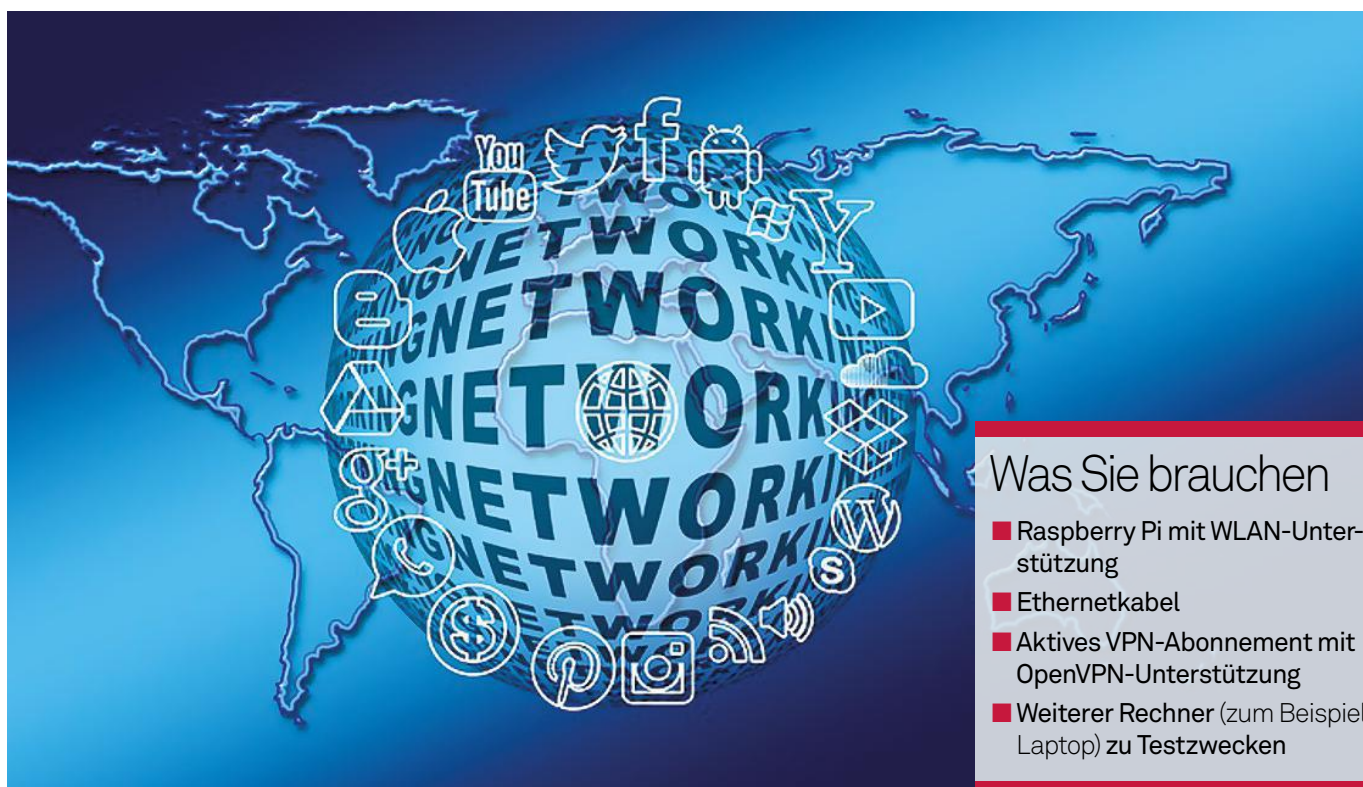
Wer schon mal im Ausland im Internet gesurft hat, kennt vermutlich das Problem, dass Content von Webseiten oft automatisch in der jeweiligen Landessprache angezeigt wird.

Und noch gravierender, in einigen Ländern sind bestimmte Angebote im Internet gar nicht zugänglich, etwa aufgrund von Zensur.

Es gibt jedoch Mittel und Wege, diese Beschränkungen zu umgehen, und dabei hilft Ihnen der Raspberry Pi. Sie können den Rechner so konfigurieren, dass er als Wireless Access Point (WAP), also WLAN-Anschlusspunkt fungiert, und ihn dann dauerhaft mit einem VPN-Dienst (Virtual Private Network) verbinden. Auf Reisen können Sie dann einfach den Pi mitnehmen und ihn an einen Router anschließen, und schon sind Sie mit Ihrem eigenen privaten Netzwerk verbunden (der ursprüngliche Zweck eines VPN war es, dass Mitarbeiter von Unternehmen sich von unterwegs aus über eine geschützte Verbindung in das interne Firmennetzwerk einwählen konnten). Für dieses Projekts brauchen Sie ein aktives VPN-Abonnement (kostet meist wenige Euro pro Monat) sowie eine Client-Konfigurationsdatei für die automatische Verbindung.

01 VPN-Anbieter auswählen

Um ein VPN nutzen zu können, benötigen Sie einen entsprechenden Anbieter. Dieser wird in der Regel einige Euro pro Monat für seine Dienste verlangen. Einige Provider akzeptieren Bitcoin als Zahlungsmethode, sodass Sie auch hier anonym bleiben können. Der VPN-Anbieter sollte das OpenVPN-Protokoll unterstützen, welches als besonders sicher gilt. Es gibt auch einige Gratis-Provider, die manchmal jedoch in puncto Geschwindigkeit oder Zuverlässigkeit zu wünschen übrig lassen.

Was Sie brauchen

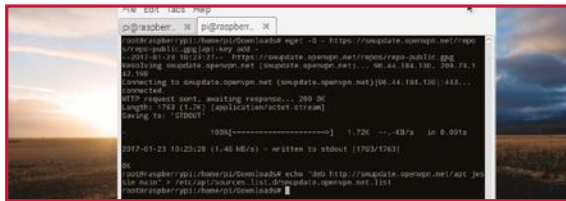
- Raspberry Pi mit WLAN-Unterstützung
- Ethernetkabel
- Aktives VPN-Abonnement mit OpenVPN-Unterstützung
- Weiterer Rechner (zum Beispiel Laptop) zu Testzwecken



02 VPN-Konfigurationsdatei herunterladen

Schließen Sie den Pi an den Router an. Öffnen Sie ein Terminal oder starten Sie eine SSH-Verbindung. Wenn Ihr Anbieter das OpenVPN-Protokoll unterstützt, wird er Ihnen eine Konfigurationsdatei (mit der Endung .conf oder .ovpn) zur Verfügung stellen. Übertragen Sie diese Datei auf den Pi, entweder vom entsprechenden Internetlink oder mittels wget-Kommando im Terminal, beim Anbieter VPNBook beispielsweise so:

```
wget http://www.vpnbook.com/free-openvpn-account/VPNBook.com-OpenVPN-Euro1.zip
```



03 OpenVPN installieren

Die Raspbian-Repositorys enthalten die OpenVPN-Software, allerdings nicht in der aktuellen Version. Wechseln Sie mit su zum Root-Nutzerkonto und tippen Sie dann:

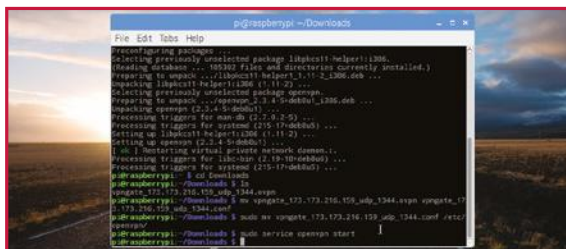
```
wget -O - https://swupdate.openvpn.net/repos/repo-public.gpg | apt-key add -
```

```
echo "deb http://swupdate.openvpn.net/apt jessie main" > /etc/apt/sources.list.d/swupdate.openvpn.net.list
```

```
apt-get update
```

```
apt-get install openvpn
```

Mit „openvpn -version“ können Sie kontrollieren, ob Sie nun die neueste Version der Software haben.



04 OpenVPN konfigurieren und ausführen

Verschieben Sie die Konfigurationsdatei mittels mv in den Ordner etc/openvpn, wobei Sie gegebenenfalls die Dateiendung korrigieren müssen:

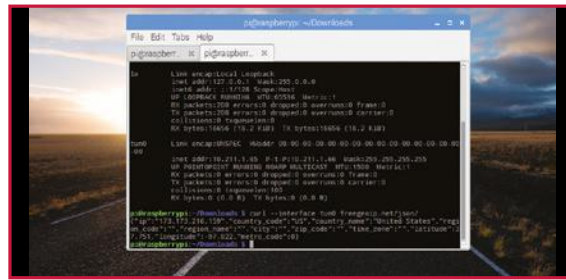
```
sudo mv vpngate_vpn15111650.opengw.net_udp_1344.ovpn /etc/openvpn/vpn1.conf
```

Mit dem folgenden Befehl können Sie die Konfigurationsdatei bearbeiten:

```
sudo openvpn --config /etc/openvpn/vpn1.conf
```

Starten Sie nun den OpenVPN-Dienst durch dieses Kommando:

```
sudo service openvpn start
```



05 OpenVPN-Verbindung testen

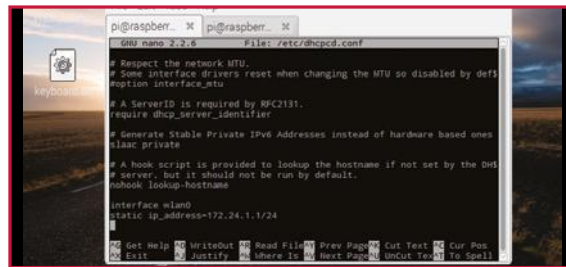
Sobald der OpenVPN-Dienst läuft, öffnen Sie einen neuen Tab im Terminal oder erzeugen Sie eine neue SSH-Verbindung und listen Sie per ifconfig-Befehl die Netzwerkschnittstellen auf. Die VPN-Verbindung wird in der Regel als tun0 aufgeführt. Checken Sie Ihren Standort mit:

```
curl --interface tun0 freegeoip.net/json/
```

Damit der OpenVPN-Dienst stets nach dem Einloggen gestartet wird, öffnen Sie zunächst diese Datei:

```
sudo nano /etc/default/openvpn
```

Entfernen Sie dann darin das # am Anfang der Zeile #AUTOSTART="all".



06 Vorbereitungen für WAP

Nun müssen wir noch die Software installieren, die für die Einrichtung des WLAN-Anschlusspunkts (WAP) notwendig ist. Tippen Sie im Terminal:

```
sudo apt-get install dnsmasq hostapd
```

Danach folgt dieses Kommando:

```
sudo nano /etc/dhcpd.conf
```

In der dadurch geöffneten Datei müssen Sie ganz unten die nachfolgenden Zeilen einfügen:

```
interface wlan0
static ip_address=172.24.1.1/24
```

Drücken Sie Strg+X, dann Strg+Y, dann Enter zum Speichern und Schließen.

Auto-Login

Falls Sie einen Benutzernamen und ein Passwort für Ihr VPN eingeben müssen, können Sie diese Angaben bei OpenVPN speichern, um ein automatisches Einloggen zu ermöglichen. Öffnen Sie die entsprechende Datei:

```
sudo nano auth.txt
```

Tragen Sie Ihren Benutzernamen in die erste Zeile und Ihr Passwort in die zweite Zeile ein, dann speichern und schließen Sie die Datei. Bearbeiten Sie anschließend die Konfigurationsdatei:

```
sudo nano /etc/openvpn/vpn2.conf
```

Scrollen Sie hinunter zur Zeile mit dem Text „auth-user-pass“. Setzen Sie ein Leerzeichen und tragen Sie dahinter den Pfad zur Datei auth.txt ein, nach folgendem Muster:

```
auth-user-pass /home/pi/auth.txt
```

Speichern und schließen Sie die Datei. Fertig!

07 Statische IP-Adresse festlegen

Öffnen Sie wie folgt die Konfiguration Ihrer Netzwerkschnittstellen:

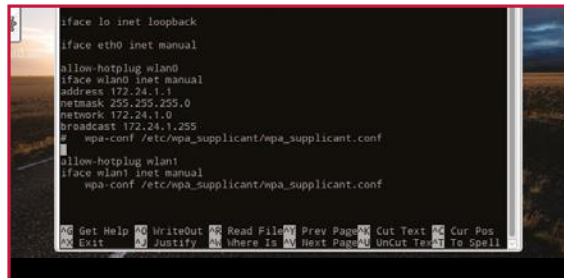
```
sudo nano /etc/network/interfaces//ENDCODE
```

Ändern Sie die Zeile „iface wlan0 inet static“ in „iface wlan0 inet manual“. Drücken Sie Enter, um eine neue Zeile zu beginnen, und fügen Sie dann diesen Block ein:

```
address 172.24.1.1
netmask 255.255.255.0
network 172.24.1.0
broadcast 172.24.1.255
```

Setzen Sie ein # an den Anfang der Zeile, die mit „wpa-conf“ beginnt, und speichern und schließen Sie die Datei wie zuvor. Starten Sie anschließend den dhcpd-Dienst neu:

```
sudo service dhcpd restart
```



```

#iface lo inet loopback
iface eth0 inet manual
allow-hotplug wlan0
iface wlan0 inet manual
address 172.24.1.1
netmask 255.255.255.0
network 172.24.1.0
broadcast 172.24.1.255
# wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
allow-hotplug wlan1
iface wlan1 inet manual
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

```

08 WAP konfigurieren

Öffnen Sie jetzt diese Datei:

```
sudo nano /etc/hostapd/hostapd.conf
```

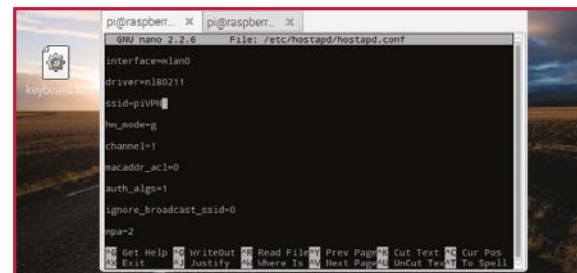
Tragen Sie dort Folgendes ein:

```
interface=wlan0
driver=nl80211
ssid=PiVPN
hw_mode=g
channel=1
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_passphrase=raspberry231
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Statt „PiVPN“ können Sie auch irgendeinen anderen Netzwerknamen vergeben. Außerdem sollten Sie ein sichereres Passwort verwenden als „raspberryp231“. Geben Sie dann ein:

```
sudo nano /etc/default/hostapd
```

Darin müssen wir die mit #DAEMON_CONF="" beginnende Zeile durch DAEMON_CONF="/etc/hostapd/hostapd.conf" ersetzen. Achten Sie auch auf das entfernte #.



```

interface=wlan0
driver=nl80211
ssid=PiVPN
hw_mode=g
channel=1
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2

```

09 dnsmasq konfigurieren

Benennen Sie die alte dnsmasq-Konfigurationsdatei um:

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```

Erstellen Sie dann eine neue mit dem ursprünglichen Dateinamen:

```
sudo nano /etc/dnsmasq.conf
```

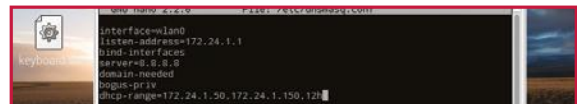
Fügen Sie Folgendes dort ein:

```
interface=wlan0
listen-address=172.24.1.1
bind-interfaces
server=8.8.8.8
domain-needed
bogus-priv
dhcp-range=172.24.1.50,172.24.1.150,12h
```

Hier verwenden wir einstweilen den DNS-Server von Google (8.8.8.8). Sie können diese Angabe gerne ändern. Speichern und schließen Sie die Datei und führen Sie anschließend das hier aus:

```
sudo nano /etc/sysctl.conf
```

Suchen Sie die Zeile beginnend mit „net.ipv4.ip_forward=1“ und entfernen Sie das # davor. Starten Sie den Pi neu.



```

#net.ipv4.ip_forward=1

```

10 IPV4-Forwarding einrichten

Die nächsten vier Befehle müssen alle nacheinander ausgeführt werden:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

```
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

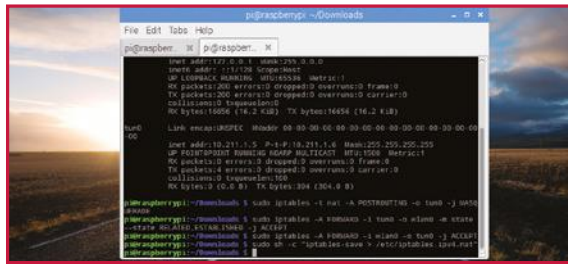
```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

Öffnen Sie danach diese Datei:

```
sudo nano /etc/rc.local
```

Fügen Sie das Folgende direkt über der Zeile „exit 0“ ein:

```
iptables-restore < /etc/iptables.ipv4.nat
/usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

11 WAP testen

Geben Sie dann im Terminal ein:

```
sudo update-rc.d hostapd enable
```

Danach dies:

```
sudo update-rc.d dnsmasq enable
```

Starten Sie den Pi neu. Sie benötigen an dieser Stelle einen zweiten Rechner, um zu überprüfen, ob Sie Zugriff auf den WAP bekommen. Suchen Sie im Netzwerkmenü danach und geben Sie das zuvor festgelegte Passwort ein. Falls Ihnen das Passwort nicht mehr einfällt, können Sie es mit folgendem Kommando betrachten:

```
sudo nano /etc/hostapd/hostapd.conf
```

Nach dem Verbinden können Sie mithilfe der Website whatismyipaddress.com feststellen, ob Sie tatsächlich im VPN sind.



12 DNS-Leaks schließen

Einige VPN-Anbieter verwenden eigene DNS-Server, doch andere sind weniger umsichtig. Besuchen Sie die Website dnsleak-test.com und führen Sie dort den „Extended test“ durch, um Ihren Sicherheitsstatus zu überprüfen. Falls mindestens einer der dort aufgeführten DNS-Server Ihrem tatsächlichen Internetprovider zugeordnet ist, dann ist Ihre VPN-Verbindung nicht vollständig anonym. Als Lösung können Sie Folgendes versuchen: Öffnen Sie diese Datei:

```
sudo nano /etc/openvpn/vpn2.conf
```

Fügen Sie direkt über „<ca>“ die folgenden Zeilen ein:

```
script-security 2
up /etc/openvpn/update-resolv-conf
down /etc/openvpn/update-resolv-conf
```

Speichern und schließen Sie die Datei, starten Sie dann den Pi neu. Führen Sie noch einmal den Test auf der genannten Website aus. Falls die Änderung nicht geholfen hat, sind Sie vermutlich bei einem anderen VPN-Anbieter besser aufgehoben.



13 Unerwünschte Verbindungen blockieren

Da der Raspberry Pi sich zwischen Ihrem zweiten Rechner und dem Internet befindet, kann auf ihn theoretisch von anderen Geräten aus zugegriffen werden. Um solche unerwünschten Zugriffe zu verhindern, setzen Sie die folgenden Befehle ein:

```
sudo iptables -A INPUT -i tun0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
```

```
sudo iptables -A INPUT -i tun0 -j DROP
```

Speichern Sie die Änderungen, sodass sie nach dem Neustart angewandt werden:

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

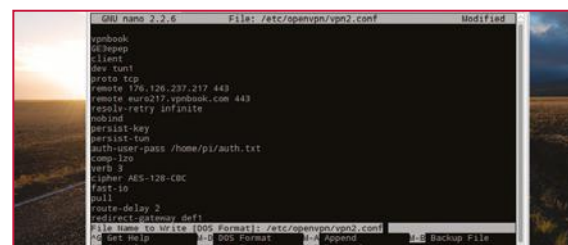


14 Gesamter Traffic durch OpenVPN

Wenn Sie den Pi hochfahren, könnten bestimmte Anwendungen versuchen, sich direkt mit dem Internet zu verbinden, was Ihre Anonymität gefährden würde. Um den gesamten Traffic über das VPN laufen zu lassen, müssen Sie noch eine Änderung an der Konfiguration vornehmen. Öffnen Sie diese Datei:

```
sudo nano /etc/openvpn/vpn2.conf
```

Ändern Sie die Zeile mit „redirect-gateway“ so, dass sie „redirect gateway def1“ lautet. DNS-Anfragen werden ebenfalls durch das VPN geleitet. Stellen Sie also sicher, dass Ihr Provider dies unterstützt.



15 Firewall einrichten

Um noch eine weitere Sicherheitsebene einzuziehen, können Sie das Programm Uncomplicated Firewall installieren. Tippen Sie:

```
sudo apt-get install ufw
```

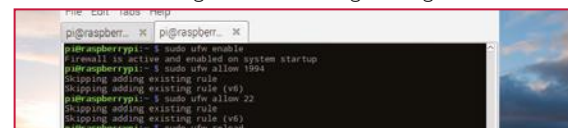
Aktivieren Sie das Programm mit dem folgenden Kommando:

```
sudo ufw enable
```

Öffnen Sie anschließend den Standardport von OpenVPN:

```
sudo ufw allow 1194
```

Für SSH-Verbindungen können Sie noch Port 22 öffnen. Denken Sie jedoch daran, dass dies keine Auswirkung darauf hat, welche Ports an dem Router geöffnet beziehungsweise geschlossen sind.



Beleuchtung mit Pi und Amazon Echo steuern

Steuern Sie das Home-Lighting-Kit von Pimoroni über den Alexa Skill mit dem Raspberry Pi

Mit dem Echo (€179) und dem kleineren Echo Dot (€59) hat Amazon mittlerweile einen eigenen Sprachassistenten für Ihr Zuhause im Angebot.

Der cloudbasierte Alexa-Dienst bietet viele integrierte Features wie Radio und Musikstreaming, das Erstellen und Bearbeiten von Einkaufslisten, Wetterinformationen und viele andere benutzerdefinierte Skills, die von Drittanbietern zur Verfügung gestellt werden. Die Hersteller dieser Skills, wie etwa Uber oder die Berliner Verkehrsbetriebe (BVG Berlin), veröffentlichen ihre Skills für die breite Öffentlichkeit, das heißt, sie durchlaufen einen Prüfungs- und Freigabeprozess ähnlich wie beim Apple App Store. In diesem Tutorial werden wir einen unveröffentlichten Skill für unseren eigenen Echo oder Echo Dot für die Steuerung des Home-Lighting-Kits „Mote“ von Pimoroni erstellen.

Unser Skill wird Sprachbefehle beinhalten, mit denen sich die Farbe der LEDs ändern lässt („Alexa, ändere Mote auf blau“) und lässt uns die Lampen mittels „Alexa, schalte Mote aus“ auch abschalten.

01 Hardware vorbereiten

Bereiten Sie Ihre Mote-Add-on-Platine vor, indem Sie deren 40-polige weibliche Buchsenleiste anbringen und verlöten; sie liegt dem Lieferumfang bei. Wenn Sie einen Pi Zero nutzen, müssen Sie auch noch die 40-polige männliche Buchsenleiste verlöten, bevor Sie weitermachen.

02 Basissystem aufbauen

Flashen Sie eine neue SD-Karte mit Raspbian Jessie Lite. Stellen Sie sicher, in der Bootpartition eine Datei namens „ssh“ zu erstellen. Darüber können wir uns mittels SSH-Remote verbinden und Befehle wie copy/paste erteilen, ohne UI-Pakete oder einen Bildschirm zu benötigen. Einmal eingesteckt, können Sie via `ssh pi@raspberrypi.local` dann auf Ihren Pi zugreifen.



Was Sie brauchen

- **GitHub-Repository:**
<https://github.com/alexellis/motephath-alexa>
- **Pimoroni mote-phat und Accessories**
(pimoroni.com)
- **Lötkolben, Flussmittel und etwas Lötzinn**



03 Reagieren auf Alexa – Lambda oder HTTPS

Alexa kann den Code entweder über einen HTTPS-Endpoint (Webdienst) oder eine Lambda-Funktion aufrufen. Letztere ist der Code, der auf Amazons AWS-Dienst hochgeladen und auf Anfrage ausgeführt wird. In diesem Tutorial richten wir mit dem Ngrok-Tool unseren eigenen HTTPS-Endpoint von unserem Pi zum öffentlichen Internet ein. Laden Sie Ngrok für Linux ARM von ngrok.com/download. Dann entpacken Sie die Datei nach `/usr/bin`.

04 Docker installieren

Docker ist ein Paket- und Runtime-System zur einfachen Erstellung, Verteilung und zum Start von Software. Führen Sie diese Befehle aus. Reconnecten Sie dann über SSH.

```
'''
# curl -sSL get.docker.com | sh
# sudo usermod pi -aG docker
'''
```

Klonen Sie nun die GitHub-Repository und erstellen Sie ein Docker-Image (das wird einige Zeit dauern):

```
'''
# apt-get update && apt-get -qy install git jq
# git clone https://github.com/alexellis/motephath-alexa
# docker build -t alexamote .
'''
```

Das so erstellte Docker-Image enthält alles, was wir für unsere Anwendung in einem isolierten Paket benötigen.

05 Beginn der Programmierung mit Docker

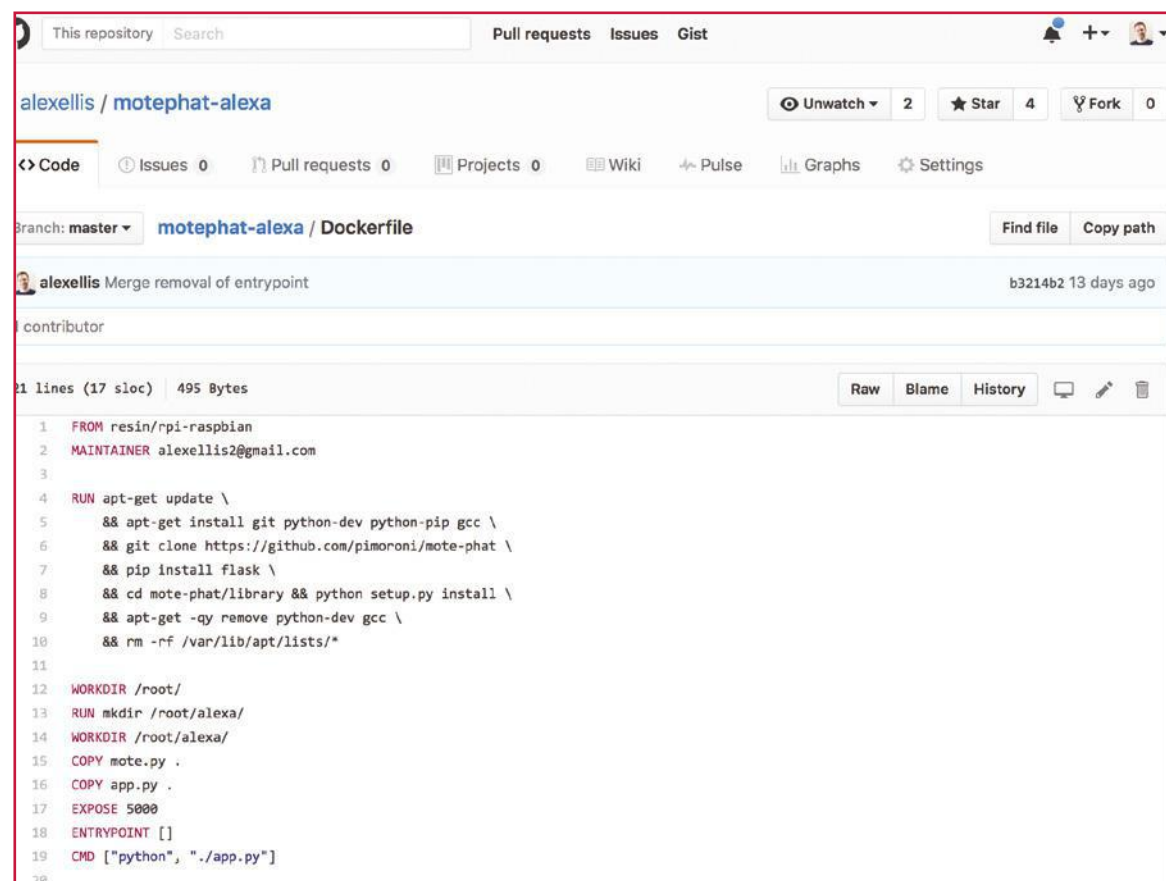
Der Code unseres Projekts ist mit all seinen Abhängigkeiten in einen einzigen Container verpackt. Wir können ihn nun im Hintergrund starten und den Ngrok-HTTPS-Tunnel zum Internet öffnen. Das Flag `-p` weist Docker an, den Port zu unserem Webservercode freizugeben, der mit Alexa kommuniziert. Das Flag `-d` bewirkt, dass der Dienst im Hintergrund läuft.

```
'''
# docker run --name mote --privileged -d -p 5000:5000 alexamote
# ngrok http 5000 > /dev/null &
# curl localhost:4040/api/tunnels | jq -r ".tunnels[1].public_url"
'''
```

Beachten Sie, dass Ihre „public_url“ mit `https` beginnt. Das ändert sich jedes Mal, wenn der Ngrok-Prozess startet. Zum späteren Stoppen des Mote-Phat-Prozesses geben Sie `docker rm -f mote` oder `docker ps` ein, um seinen Status anzuzeigen.

Natural Language Parsing (NLP)

Der Alexa-Dienst sendet Aufnahmen von Ihrem Gerät in die Amazon-Cloud, wo die NLP (Natural Language Parsing) sie in einzelne Wörter und Intentionen aufbricht. Eine Beispielfrase wie etwa „Alexa, wie wird morgen das Wetter in Berlin?“ würde als Absicht „FindeWetter“ mit den beiden Variablen: Datum=morgen und Ort=Berlin erkannt werden. Beispielausführungen helfen, die vielen verschiedenen Möglichkeiten, das Gleiche in der gleichen Sprache auszudrücken, abzudecken.



Docker

Docker ist eine bahnbrechende Technologie für das Packen, Verteilen und Starten von Software. Jedes Mal, wenn Sie Software programmieren, erstellt Docker ein Image mit eigenen Root-Dateisystem und Netzwerkadressen. Das Innere eines Docker-Containers fühlt sich genau wie eine komplette virtuelle Maschine an, nur schneller, denn ein Container ist ein regulärer Prozess mit einigen erweiterten Systemaufrufen für Sicherheit und Isolation. Das Docker-CLI ist mit Befehlen wie „docker ps“, „docker run“ und „docker kill“ für Linux-Benutzer intuitiv. Der „docker build“-Befehl nutzt ein Docker-File, ähnlich wie ein Make-File.

06 Endpunkt testen

Sobald Sie Ihre HTTPS-URL von Ngrok haben, können Sie alles ausprobieren, indem Sie eine Anfrage ähnlich denen vom Alexa-SDK, senden. Wir haben zwei Samples aufgenommen und in der Git-Repository gespeichert.

Test Lichtfarbe auf rot ändern:

```
'''
# curl -X POST -H "Content-type: application/json" -d @coloursample.json https://c00738f6.ngrok.io
'''
```

Test Licht ausschalten:

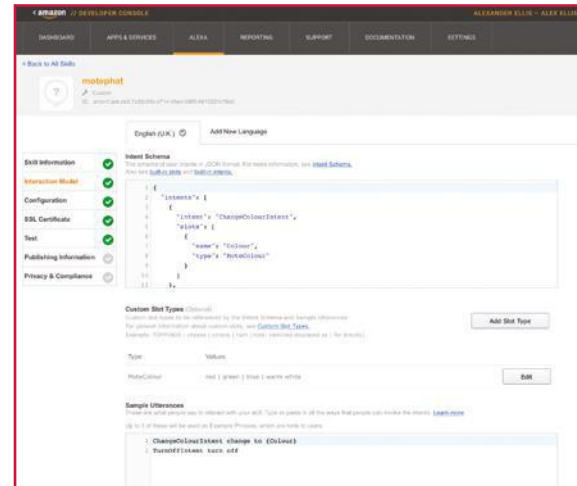
```
'''
# curl -X POST -H "Content-type: application/json" -d @coloursample.json https://c00738f6.ngrok.io
'''
```

07 Einen Alexa-Skill erstellen

Rufen Sie <https://developer.amazon.com/myapps.html> auf und klicken Sie auf Alexa > Alexa Skills Kit > Get Started. Möglicherweise müssen Sie sich für diesen Schritt anmelden und Abrechnungsinformationen für alle Einkäufe zur Verfügung stellen, die Sie tätigen möchten.

Klicken Sie auf Add a New Skill und geben Sie „mote“ als

Name und Aufrufname an. Als Intentions-Schema copy/pasten Sie „speechAssets/intentSchema.json“ und als Beispielläußerungen „speechAssets/sampleUtterances.txt“. Zudem müssen Sie noch den benutzerdefinierten Slot „Farbe“ mit den Werten rot/grün/blau in separaten Zeilen erstellen. Dieser Slot hilft Alexa, indem er eine Liste aller Dinge liefert, die man ihr sagen könnte – wie ein Parameter bei Codes.





amazon // DEVELOPER CONSOLE ALEXANDER ELLIS – ALEX ELLIS

DASHBOARD APPS & SERVICES **ALEXA** REPORTING SUPPORT DOCUMENTATION SETTINGS

< Back to All Skills

motephath
Custom
ID: amzn1.ask.skill.7a32c53b-d714-45ed-b955-0610221b70e0

English (U.K.) Add New Language

Skill Information ✓

Interaction Model ✓

Configuration ✓

SSL Certificate ✓

Test ✓

Publishing Information ✓

Privacy & Compliance ✓

Intent Schema
The schema of user intents in JSON format. For more information, see [Intent Schema](#). Also see [built-in slots](#) and [built-in intents](#).

```

1 {
2   "intents": [
3     {
4       "intent": "ChangeColourIntent",
5       "slots": [
6         {
7           "name": "Colour",
8           "type": "MoteColour"
9         }
10      ]
11    }
12  ]
13 }
```

Custom Slot Types (Optional)
Custom slot types to be referenced by the Intent Schema and Sample Utterances. For general information about custom slots, see [Custom Slot Types](#).
Example: TOPPINGS - cheese | onions | ham (note: newlines displayed as | for brevity)

Type	Values
MoteColour	red green blue warm white

Sample Utterances
These are what people say to interact with your skill. Type or paste in all the ways that people can invoke the intents. [Learn more](#)
Up to 3 of these will be used as Example Phrases, which are hints to users.

```

1 ChangeColourIntent change to {Colour}
2 TurnOffIntent turn off
```

08 Alexa den HTTPS-Endpoint zuweisen

Klicken Sie im Konfigurationstab Ihres Alexa Skills auf service endpoint type: HTTPS. Wählen Sie dann die nächstgelegene Region (Europa) und fügen Sie die Ngrok-URL, die sie bei Schritt 5 erhalten haben, ein. Klicken Sie nun auf „Mein Entwicklungsendpunkt hat ein Zertifikat von einer vertrauenswürdigen Zertifizierungsstelle“ im SSL-Zertifikats-Tab.

Im Test-Tab können Sie Beispieläußerungen wie etwa „wechseln auf blau“ oder „ausschalten“ eingeben. Wenn Sie auf „Frage Mote“ klicken, wird eine Nachricht von Ihrem Pi an Alexas Online-dienst übermittelt und läuft über den Echo/Dot.

English (U.K.) Add New Language

Global Fields
These fields apply to all languages supported by the skill.

Endpoint

Service Endpoint Type: ☐ AWS Lambda ARN (Amazon Resource Name) ☒ HTTPS

Recommended
AWS Lambda is a serverless compute service that runs your code in response to events and automatically manages the underlying compute resources for you.
[More info about AWS Lambda](#)
[How to integrate AWS Lambda with Alexa](#)

Pick a geographical region that is closest to your target customers:

☐ North America ☒ Europe

Europe

Account Linking
Do you allow users to create an account or link to an existing account with your? ☐ Yes ☒ No
[Learn more](#)

09 Mit Ihrem Echo reden

Wenn alles funktioniert, können Sie mit Ihrem Echo/Dot reden. Sagen Sie einfach „Alexa, Mote soll die Farbe auf rot ändern“ oder „Alexa, bitte Mote ausschalten“. Ideen für Codes zum Dimmen der Helligkeit finden Sie etwa auf Alex's Christmas Tree Hack's Source-Code unter: <http://blog.alexellis.io/christmas-iot-tree>

10 Nächste Schritte

Nun, da Sie Ihren ersten Skill erstellt haben, fallen Ihnen vielleicht Möglichkeiten zum Erweitern oder zum Umsetzen bei anderen Hardwareprojekten ein. Das Dimmen des Lichts könnte nützlich sein. Ebenso einfach ist das Hinzufügen weiterer Farben. Möchten Sie mehr über Docker wissen, lesen sie den Kasten auf S. 146, das Docker-File im GitHub-Artikel und Alex's Anfänger-Tutorials unter <http://blog.alexellis.io/tag/raspberry-pi/>

```

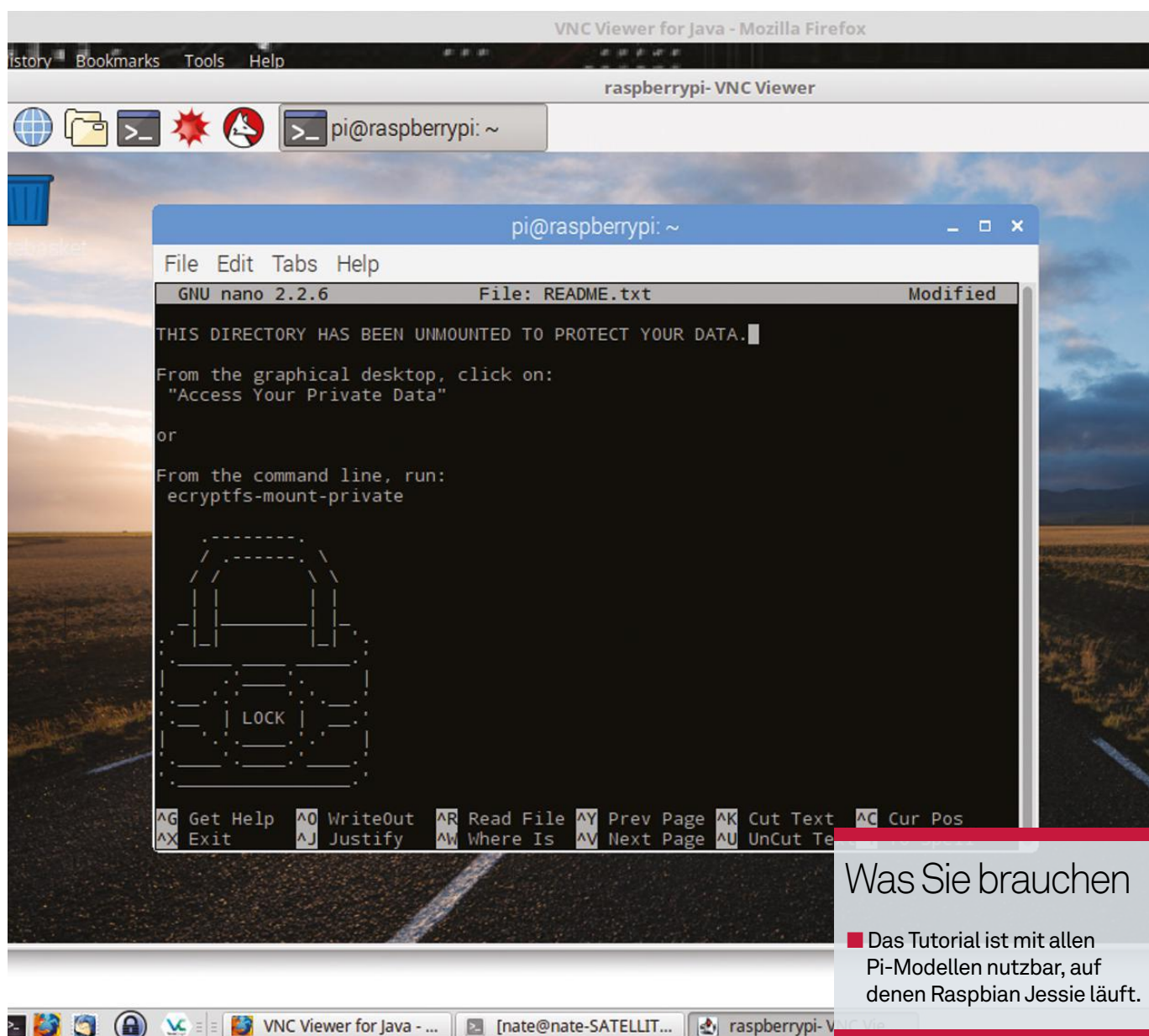
18 def post_data(request):
19     post_data = request.json if data is empty
20     response = None
21
22     if post_data["request"]["intent"] == "TurnOffIntent":
23         response = get_response("Turning off", "OK")
24     else:
25         response = get_response("OK setting desired colour", "OK")
26         slot_colour = post_data["request"]["intent"]["slots"]["colour"]["value"]
27         if not slot_colour in ["red", "green", "blue"]:
28             response = get_response("Can only set red, green or blue", "Error")
29         else:
30             red = 0
31             green = 0
32             blue = 0
33
34             if slot_colour == "red":
35                 red = 255
36             elif slot_colour == "green":
37                 green = 255
38             elif slot_colour == "blue":
39                 blue = 255
40
41             note.set_colour(red, green, blue)
42
43     return Response(json.dumps(response), mimetype="application/json")
```

Home-Verzeichnis des Pi verschlüsseln

Mit einer Verschlüsselung im Militärstandard sind Ihre Daten sicher.

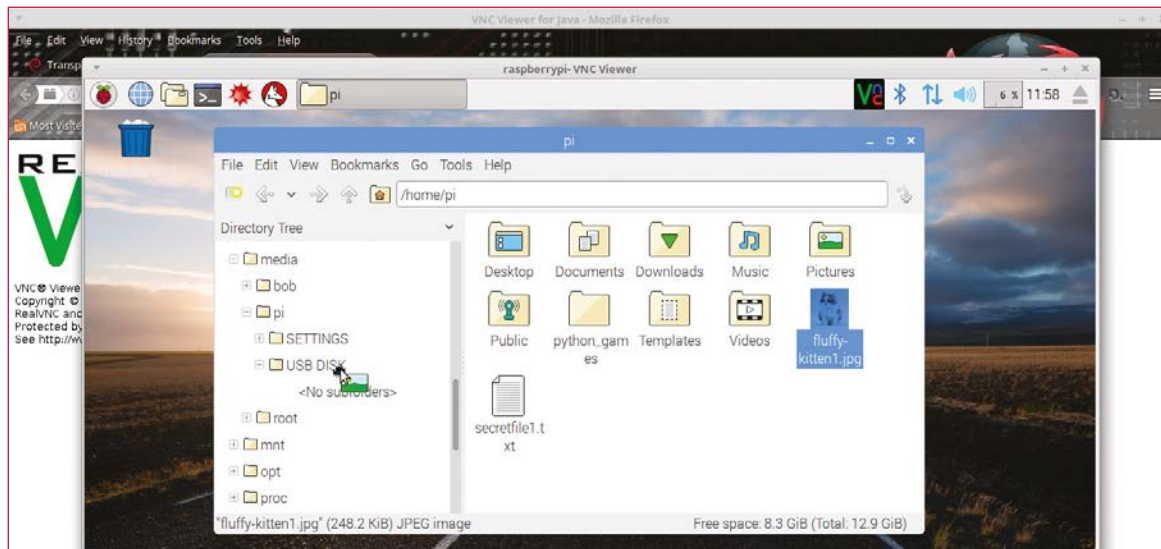
Seit der Einführung des prächtigen Pixel-Desktops auf dem Raspberry Pi, mit seinen auffälligen Icons und frischen Fenstern, möchte so mancher vielleicht den Pi als Heimcomputer nutzen. Allerdings loggt der Pi den Stan-

dardbenutzer automatisch ein, und selbst wenn ein Passwort eingestellt wird, kann jeder die SD-Karte mit Ihren Dokumenten, Bildern und Videos auslesen. Verschlüsselung ist also angesagt!



Was Sie brauchen

- Das Tutorial ist mit allen Pi-Modellen nutzbar, auf denen Raspbian Jessie läuft.



01 Backup der Daten

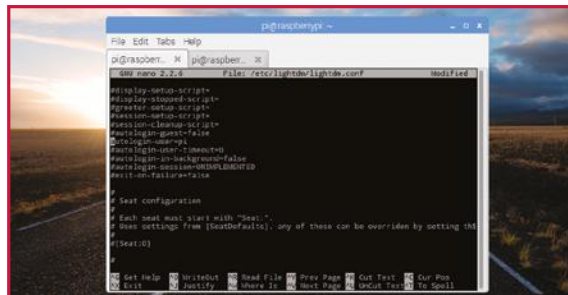
Falls nicht bereits geschehen, loggen Sie sich in Ihren Raspberry Pi ein und speichern Sie sämtliche Fotos, Dokumente und dergleichen auf einem USB-Stick. So können Sie Ihre Daten leichter in das verschlüsselte Home-Verzeichnis übertragen. Mit Strg+H können Sie versteckte Ordner anzeigen, etwa um ein Backup der Einstellungen Ihrer Anwendungen oder Ihrer Lesezeichen zu machen.

02 Login aktivieren

Öffnen Sie das Terminal und verbinden Sie sich über SSH mit dem Pi und geben Sie ein:

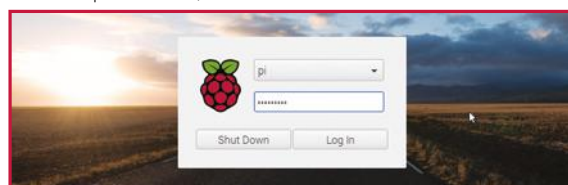
```
sudo nano /etc/lightdm/lightdm.conf
```

Damit rufen Sie die Login-Optionen auf. Darin vor `autologin-user=pi` eine Raute (#) setzen, um dies auszukommentieren. Drücken Sie Strg+X, dann Y und dann Enter, um zu speichern und zu schließen. Das voreingestellte Passwort ist „raspberrry“.



03 Neustart und einloggen

Den Pi neu starten und über den Login-Screen einloggen. Der Nutzername Pi sollte vorausgewählt sein und das Passwort „raspberrry“ lauten. Die nächsten Schritte sollten Sie direkt auf dem Pi-Desktop ausführen, da Sie zwischen Nutzern wechseln müssen



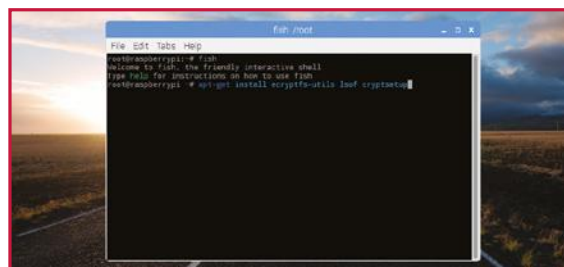
und dies über SSH schwierig ist. Haben Sie keinen Monitor für Ihren Pi, versuchen Sie eine Verbindung als Nutzer „Pi“ über VNC.

04 eCryptfs und verbundene Dateien installieren

Tippen Sie den folgenden Befehl ins Terminal:

```
sudo apt-get install ecryptfs-utils lsof cryptsetup
```

Bestätigen Sie mit „Y“, dass Sie die Software installieren möchten. Dies wird Ihnen ermöglichen, das Home-Verzeichnis zu verschlüsseln und aufzurufen, wenn Sie sich einloggen.

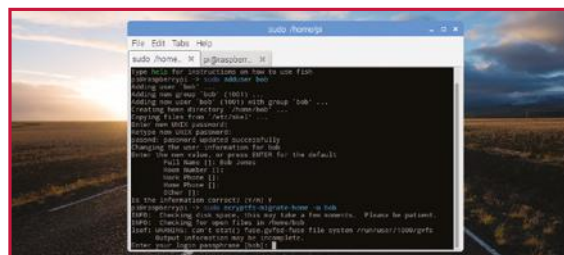


05 Neuen Nutzer erstellen

Möchten Sie Ihren Pi ernsthaft nutzen, werden Sie wohl einen eigenen Nutzeraccount anlegen wollen. Mit dem folgenden Befehl erstellen Sie einen neuen Nutzer:

```
sudo adduser name
```

Dabei ersetzen Sie „name“ durch Ihren gewünschten Nutzernamen. Tippen Sie zweimal das Passwort ein und drücken Sie die Enter-Taste, um die sonstigen voreingestellten Optionen zu akzeptieren.



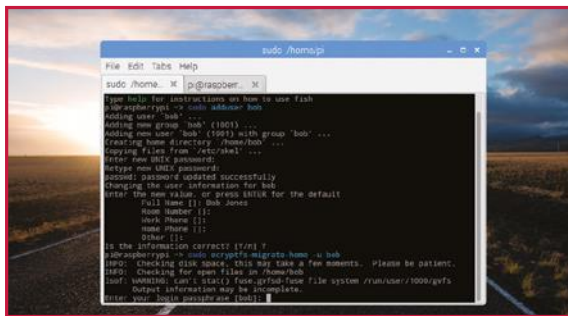
Gutes Passwort wählen

Auch bei verschlüsseltem Nutzerverzeichnis kann man von einem anderen Computer aus mit einem Brute-Force-Angriff Passwörter ausprobieren, bis Ihres gefunden wurde. Besuchen Sie die Seite howsecureismypassword.net, um zu prüfen, wie lange das dauern würde. Ist die Zeit kürzer als ein Jahrhundert, sollten Sie ein stärkeres Passwort wählen. Die Diceware-Website world.std.com/~reinhold/diceware.html bietet eine schnelle und einfache Möglichkeit, starke und einfach zu merkende Passwörter zu erstellen.

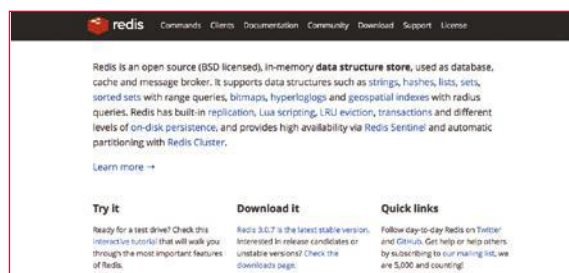
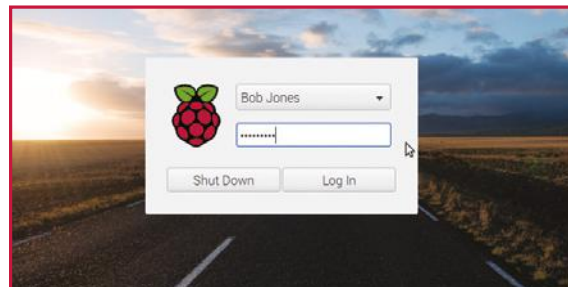
06 Neues Home-Verzeichnis verschlüsseln
Die Software eCryptfs enthält ein einfaches Hilfsmittel, um Home-Verzeichnisse zu verschlüsseln. Tippen Sie:

```
sudo ecryptfs-migrate-home -u name
```

Dabei steht „name“ wieder für den neuen Nutzernamen. Geben Sie auch hier zweimal das Passwort ein, es kann auch das vorher genutzte sein. Lesen Sie anschließend die Anmerkungen. An dieser Stelle noch nicht neu starten.



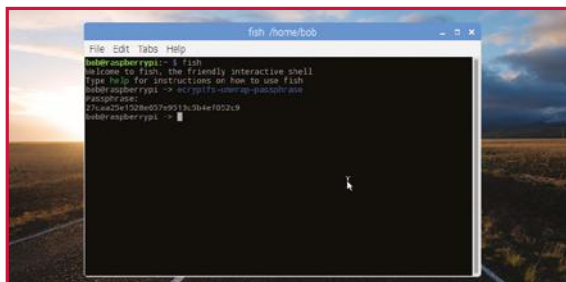
07 In den verschlüsselten Account einloggen
Mit Menu > Shutdown > Logout aus dem Nutzer Pi ausloggen. Im Drop-down-Menü den neuen Nutzer auswählen und einloggen. Bevor Sie neue Daten aufspielen, öffnen Sie ein Terminal und tippen einfach mount. Sehen Sie ein Durcheinander mit einem Hinweis auf „ecryptfs“ so war die Verschlüsselung erfolgreich.



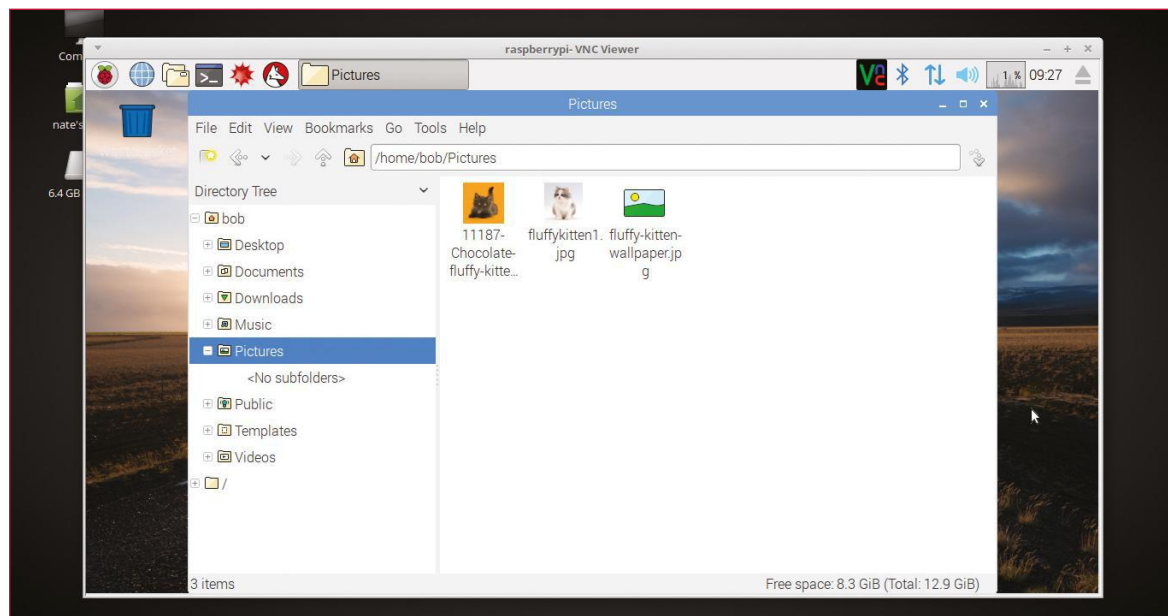
08 Backup der Passphrase
Können Sie wegen Systemproblemen nicht einloggen oder möchten Sie Daten an einen neuen Computer senden, werden diese standardmäßig nicht zugänglich sein. Glücklicherweise besitzt eCryptfs eine Funktion, die die Mount-Passphrase generiert, mit der ein Zugriff von einem anderen Computer aus möglich ist. Öffnen Sie ein Terminal und geben Sie ein:

```
ecryptfs-unwrap-passphrase
```

So bekommen Sie die Passphrase angezeigt. Notieren Sie sie und verwahren Sie sie an einem sicheren Ort.



09 Daten importieren
Mit Menu > Shutdown > Reboot starten Sie Ihren Pi neu. Loggen Sie sich mit dem neuen Nutzernamen ein. Verbinden Sie Ihr externes Laufwerk und übertragen Sie Ihre Daten zurück. Sie werden sehen, dass ein neues Unterverzeichnis in /home nach dem Muster /home/name erstellt wurde. Kein anderer Nutzer auf dem Pi wird auf die Daten in Ihrem neuen Nutzerverzeichnis zugreifen können.





10 Adminrechte erteilen

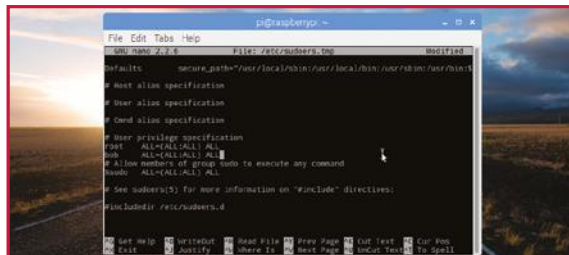
Aus dem Nutzeraccount aus- und in den Pi-Account einloggen. Im Terminal eingeben:

sudo visudo

Bis zur Zeile „root ALL=(ALL:ALL) ALL“ scrollen und in eine neue Zeile direkt im Anschluss schreiben:

name ALL=(ALL:ALL) ALL

Das „name“ wieder durch den tatsächlichen Nutzernamen ersetzen. Drücken Sie Strg+X, nun Y, dann Enter, um zu speichern und zu schließen. Den Pi neu starten, damit die Änderungen angewandt werden.



11 Backup des Home-Verzeichnisses löschen

Beim Verschlüsseln Ihres neuen Nutzerverzeichnisses hat eCryptfs ein Backup in Ihrem Nutzerverzeichnis erstellt, für den Fall, dass etwas schief läuft. Da dies ein neuer Account war, ist der Ordner leer. Um den genauen Namen des Ordners zu ermitteln, tippen Sie:

ls /home

Und dann:

sudo rm -r -f <directoryname>

So entfernen Sie den Ordner.



12 Daten aus dem Pi-Verzeichnis löschen

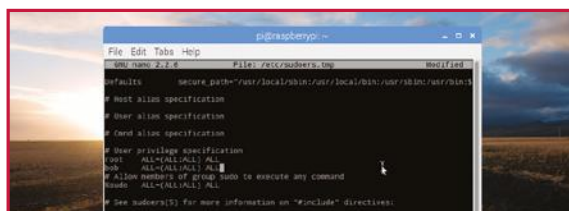
Persönliche Daten in Ihrem Nutzerverzeichnis im ursprünglichen Pi-Account sind nach wie vor unverschlüsselt und können von jedem angesehen werden, der an Ihre microSD-Karte gelangt. Eingeloggt in den Pi-Account, nutzen Sie den Befehl shred, um Dateien sicher zu löschen:

shred -zu beispieldatei.pdf

Mit diesem Befehl können Sie ganze Ordner löschen:

sudo rm -r -f <directoryname>

Achten Sie jedoch darauf, keine Hauptordner wie etwa „Desktop“ zu löschen.



13 Swap-Partition deaktivieren

Ein Teil der SD-Karte (Pfad in /var/swap) wird in ähnlicher Weise genutzt wie der RAM, wenn der Pi knapp an Ressourcen ist. Das kann ein Sicherheitsrisiko darstellen, da die Daten möglicherweise dort unverschlüsselt geschrieben werden. Eingeloggt in Ihren Nutzeraccount, öffnen Sie ein Terminal und geben Sie ein:

sudo swapoff -a -v

Sie sollten daraufhin eine Bestätigung über die Deaktivierung erhalten.

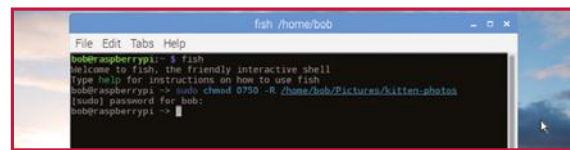


14 Zugriffsfehler beheben

Je nachdem, wie Sie das Backup und den Rücktransfer durchgeführt haben, kann es bei einzelnen Dateien und Ordnern zu Problemen bei den Zugriffsrechten kommen (in der Regel erkenntlich an einem Vorhängeschloss-Icon an der Datei). Mit dem folgenden Kommando stellen Sie sicher, dass der neue Nutzeraccount Besitzer aller Dateien dieses Ordners ist:

sudo chmod 0750 -R foldername

Wenn „bob“ Ihr Nutzernamen ist, dann wäre zum Beispiel „sudo chmod 0750 -R /home/bob/Fotos“ ein möglicher Befehl.



15 Testen Sie, ob Ihre Daten sicher sind

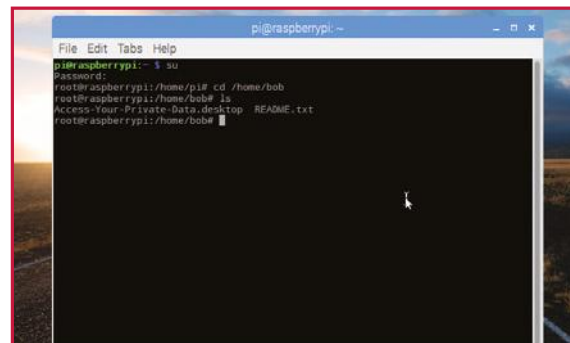
Dieser Schritt ist optional. Öffnen Sie ein Terminal und tippen Sie:

sudo passwd root

um ein Passwort für den Root-Account zu setzen. Geben Sie nun den Befehl su um in root zu wechseln. Tippen Sie (mit dem zutreffenden Namen statt „name“):

cd /home/name

Lassen Sie sich mit ls den Inhalt des Verzeichnisses auflisten. Nun sind da nur ein Shortcut und eine „ReadME“-Datei zu sehen, die besagt, dass Ihre Daten geschützt sind.





Was Sie brauchen

- NoIR-Kameramodul
- LISIPAROI LED-Licht
bit.ly/1meQGaR
- Infrarotsensor
- Batterie für den Pi

Nächtliche Fotos mit der Kamera Pi NoIR

Machen Sie mit dieser mobilen Kamera Nachtaufnahmen der in Ihrem Garten verborgenen Fauna.

Vielleicht sind Sie mit dem normalen Kameramodul des Raspberry Pi bereits vertraut. Es ist allerdings auch als NoIR-Edition erhältlich – NoIR steht hier für „No Infrared“, das heißt, es ist kein Infrarotfilter enthalten. Mit diesem Modul können Sie die Kamera, ähnlich einem Nachtsichtgerät, für Aufnahmen im Dunkeln verwenden. Viele von uns bekommen regelmäßig Tierbesuch im Garten. Tagsüber ist dieser natürlich viel sichtbarer, aber es gibt auch etliche Tiere, die auf nächtlichen Streifzügen unsere Gärten kreuzen. Diese Anleitung zeigt, wie man die Kamera aufbaut und ihren Auslöser mit Infrarotlicht und einem passiven Infrarot-Bewegungsmelder koppelt, um die nächtliche Tierwelt dokumentieren zu können. Jedes aufgenommene Bild wird auf dem Raspberry Pi gespeichert und mit der Aufnahmezeit versehen, damit genau ersichtlich wird, wann der Tierbesuch stattgefunden hat.

01 NoIR-Kamermodul anbringen

Zunächst muss die Kamera am Raspberry Pi angebracht werden. Leiten Sie vor dem Einbau sicherheitshalber die statische Aufladung Ihres Körpers ab, etwa indem Sie einen Heizkörper berühren; die Kamera könnte sonst Schaden nehmen. Das blaue Etikett muss von der HDMI-Schnittstelle weg zeigen. Ist die Kamera verbunden, starten Sie den Pi und aktivieren Sie sie per „sudo raspi-config“ in den Einstellungen. Starten Sie anschließend neu. Es empfiehlt sich außerdem, mit folgenden Befehlen etwaige Software-Updates auszuführen:

```
sudo apt-get update
sudo apt-get upgrade
```



Oben Das NoIR-Kameramodul fängt gerade genug Licht ein, um faszinierende nächtliche Außenaufnahmen zu ermöglichen.

02 Das erste Foto

Prüfen Sie nach dem Update, ob die Kamera korrekt funktioniert. Geben sie „`raspistill -v -o test.jpeg`“ ins Terminal ein; daraufhin sollte ein Vorschaubild angezeigt und ein Foto mit dem Dateinamen „test“ im Ordner `/home/pi` gespeichert werden. Der Parameter `-o` bedeutet Output und gibt den Dateinamen vor. Die Eingabe „`raspistill -o keyboard.jpeg`“ würde beispielsweise eine Datei namens „keyboard“ speichern.

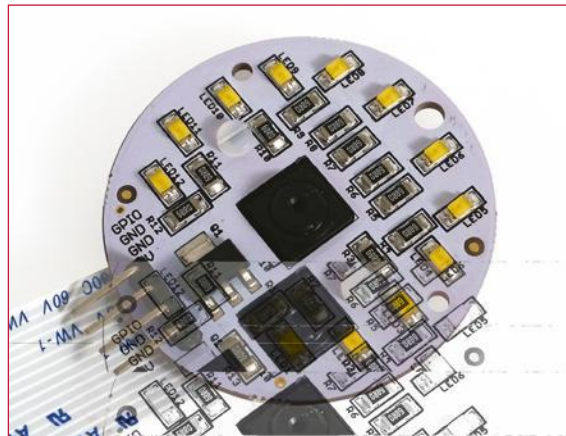
03 Fotos mit Python

Die Steuerung des NoIR-Moduls erfolgt mithilfe der Programmiersprache Python. Für Fotoaufnahmen sind nur wenige Zeilen Code nötig. Öffnen Sie Ihren Python-Editor und erstellen Sie die unten angegebene Funktion, dann speichern und starten Sie sie. Ein Vorschaubild kann mit dem Code „`camera.start_preview()`“ angezeigt werden. Den virtuellen Auslöser betätigt die Zeile „`camera.capture('nature.jpg')`“. Ersetzen Sie dabei „nature“ durch einen beliebigen Namen und testen Sie anschließend, ob die Kamera korrekt mit Python zusammenarbeitet.

```
import time
import picamera

def Nature_selfie()
    with picamera.PiCamera() as camera:
        camera.start_preview()
        time.sleep(2)
        camera.capture('nature.jpg')

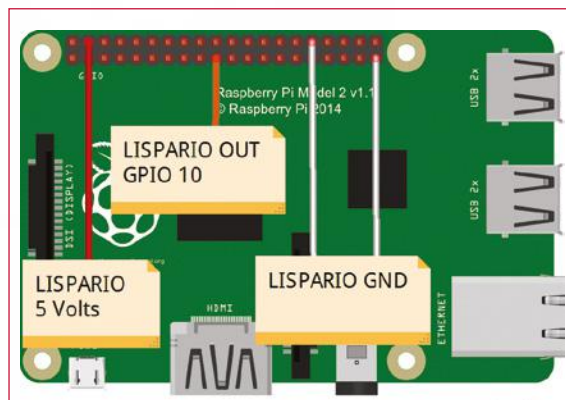
Nature_selfie()
```



04 Funktioniert das wirklich?

Bei der ersten Nachtaufnahme kann es den Anschein haben, dass das NoIR-Modul nicht funktioniert – es kommt kein Spionagefoto mit der Nachtsichtkamera heraus, sondern höchstwahrscheinlich ein völlig schwarzes Bild. Um das zu ändern, brauchen Sie eine Infrarot-Lichtquelle. Das LISIPAROI ist ein optionales Extra für das Kameramodul, das Stützbeleuchtung für Nachtaufnahmen bietet. Es hat eigene Befestigungspunkte, die sich gut für selbstgebaute Halterungen oder einen Schwanenhals eignen. Es gibt neben der Standardversion eine Infrarotversion für die NoIR-Kamera. Beim Infrarot-LISIPAROI sind 12 transparente Infrarot-LEDs um die Kamera herum angebracht, die eine starke Ausleuchtung in Situationen mit wenig oder gar keinem Licht bieten – perfekt für Aufnahmen der nächtlichen Tierwelt.

Für Fotoaufnahmen sind nur wenige Zeilen Python-Code nötig.



05 LISIPAROI anschließen

Um das Gerät anzuschließen, genügen vier Steckkabel, die von den Pins des LISIPAROI zum Raspberry Pi führen. Dort werden sie an 5V, zweimal Masse (GND) und GPIO 10 angesteckt. Der Code adressiert die BCM-Pins – zum Anschluss ist es sicher leichter, die physikalischen Pins anzugeben. 5V wird an den physikalischen Pin 4 angeschlossen, die Massekabel kommen an die Pins 32 und 39 (Sie können auch andere Masse-Pins nutzen). Das übrige Kabel wird an den Pin GPIO 10, der die Nummer 19 auf der Platine hat, gesteckt. Und schon ist das LISIPAROI angeschlossen und betriebsbereit.

Cron

Das Hintergrundprogramm Cron ist ein zeitbasierter Prozessmanager, der es ermöglicht, Prozesse (Eingaben oder Skripte) in bestimmten Zeitintervallen auszuführen. Cron wird oft verwendet, um Administrations- oder Festplatten-Wartungsaufgaben zu automatisieren.

BCM-Pins

GPIO-Pins sind eine physikalische Schnittstelle zwischen dem Raspberry Pi und der Welt. Ganz grundsätzlich sind sie fast wie Schalter, die Sie oder der Pi selbst ein- und ausschalten können. Die Option GPIO.BCM bewirkt, dass die Pins mit der BCM-Nummer (Broadcom SOC Channel) adressiert werden. Wenn Sie die Pins zählen, finden Sie hingegen die physikalische Pin-Nummer heraus. Die Option GPIO.BOARD gibt an, dass Pins mit dieser Nummer adressiert werden, also derjenigen, die auf der Platine abgedruckt ist.

06 Ein weiterer Test

Passen Sie jetzt den vorherigen Kamera-Code an, damit er das LISIPAROI für Aufnahmen in der Dunkelheit aktiviert. Importieren Sie das GPIO-Modul mit der Zeile „import RPi.GPIO as GPIO“ (siehe unten). Stellen Sie die Adressierung mit dem Code „GPIO.setmode(GPIO.BCM)“ auf BCM um. Das LISIPAROI liegt an Pin 10 an, darum wird „GPIO.setup(10, GPIO.OUT)“ festgelegt. Vor der Aufnahme muss die Ausgabe mit „GPIO.setup(10, GPIO.HIGH)“ auf HIGH gestellt werden, um die LEDs zu aktivieren. Das Bild wird aufgenommen und gespeichert, die LEDs werden wieder deaktiviert. Sehen Sie sich das Bild an – ein waschechtes Nachtsichtfoto.

```
import RPi.GPIO as GPIO
import time
import picamera
GPIO.setmode(GPIO.BCM)
GPIO.setup(10, GPIO.OUT)
GPIO.output(10, GPIO.HIGH)

with picamera.PiCamera() as camera:
    camera.start_preview()
    time.sleep(2)
    camera.capture('nature.jpg')
    camera.stop_preview()

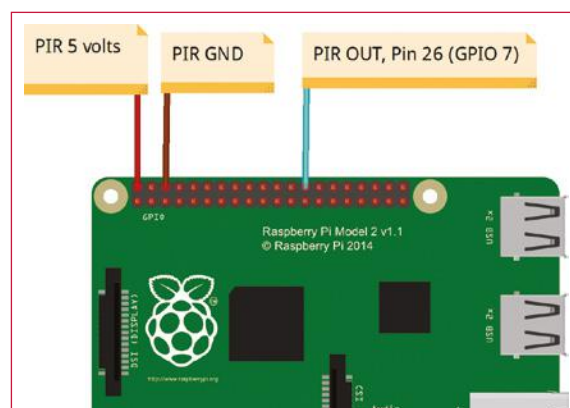
GPIO.output(10, GPIO.LOW)
```

07 Infrarotsensor

Ein PIR-Sensor (passiver Infrarotsensor) erkennt Objekte anhand der Wärmeenergie, die sie ausstrahlen. Diese Strahlung ist für das menschliche Auge unsichtbar, da sie sich im Infrarot-Wellenbereich bewegt. Elektronische Geräte wie ein PIR-Sensor können sie aber aufspüren, indem sie anzeigen, wenn sich eine Temperaturveränderung ereignet hat. Der Sensor hat zwei Einstellmöglichkeiten: Seine Wärmeempfindlichkeit kann reguliert werden, sodass er bei kleineren oder größeren Temperaturveränderungen anschlägt, und die Ruhezeit zwischen den Schaltintervallen kann angepasst werden. Sie liegt ursprünglich bei einigen Sekunden.

08 Sensor anschließen

Schließen Sie jetzt den Sensor an. Um seine Funktion zu testen, entfernen Sie zunächst die Kabel vom LISIPAROI. Er hat drei Kabel: 5V, Masse und Ausgabe. Denken Sie daran, dass Sie BCM-Pins adressieren müssen, die Nummer im Code ist also die GPIO-Nummer auf dem Raspberry Pi. Das +5V-Kabel kommt an den physikalischen Pin 2, das Ausgabekabel an Pin 26 (GPIO 7), und die Masse wird mit Pin 6 verbunden.



09 Sensor testen

Wenn der Sensor verbunden ist, prüfen Sie, ob er funktioniert, und stellen Sie ihn so ein, dass er korrekt auslöst. Öffnen Sie eine neue Datei in Ihrem Python-Editor und geben Sie den unten aufgeführten Testcode ein. Die wichtige Zeile ist:

```
GPIO.add_event_detect(PIR, GPIO.RISING,
    callback=Motion_Sensing)
```

Sie liest den Anstieg im GPIO aus, wenn beispielsweise ein Dachs in den Infrarotsensor gerät und so Pin 7 auslöst. Der Raspberry Pi erkennt, dass die Spannung an dem Pin steigt, und führt den Callback aus. In diesem Beispiel ist der Callback eine Funktion namens Motion_Sensing, die in der Python-Konsole den Text „We see you“ anzeigt, wenn sie ausgeführt wird. Wenn sich der Dachs bewegt, erkennt der PIR-Sensor eine weitere Temperaturveränderung.

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)

PIR = 7
GPIO.setup(PIR, GPIO.IN)

print "Ready to find you"
time.sleep(2)

def Motion_Sensing(PIR):
    print "We see you"

try:
    GPIO.add_event_detect(PIR, GPIO.RISING,
        callback=Motion_Sensing)
    while 1:
        time.sleep(100)
except KeyboardInterrupt:
    print "Quit"
GPIO.cleanup()
```

10 Neues Bild speichern

Aktuell überschreibt der Pi noch jedes Mal die vorhandene Datei mit dem gerade neu aufgenommenen Bild, da im Code aus Schritt 03 stets derselbe Dateiname spezifiziert ist. So kann man natürlich keine Bilderserie über die Nacht hinweg aufnehmen. Um ein Überschreiben zu vermeiden, erstellen Sie eine Variable namens File_Number. Immer wenn die Kamera auslöst, steigt der Wert der Variablen um 1. Im Beispiel weiter unten heißt die erste Datei also Nature1.jpg, die nächste Nature2.jpg, dann Nature3.jpg und so weiter. Jedes neue Bild erhält einen eigenen Dateinamen, nichts wird überschrieben.

```
camera.capture("Nature" + str(File_Number) + ".jpg")
File_Number = File_Number + 1
```

11 Aufnahmezeit hinzufügen

Sicher möchten Sie wissen, wann die Kamera denn tatsächlich ausgelöst und fotografiert hat – insbesondere, wenn Ihr Aufbau über Nacht gelaufen ist. Erstellen Sie dazu bei Programmstart eine Variable namens time_of_photo, die die aktuelle Uhrzeit speichert. Um die Aufnahmezeit abzurufen, verwenden Sie „time.asctime(time.localtime(time.time()))“. Der Code fragt die aktuelle Uhrzeit des Raspberry Pi ab und speichert sie in der Variablen time_of_photo. Prü-



fen Sie vor Programmstart noch, ob die Uhr Ihres Pi richtig eingestellt ist.

```
time_of_photo = time.asctime(time.localtime(time.time()))
```

12 Zeit im Bild

Diese Art der Zeitaufzeichnung ist nur relevant, wenn Sie live dabei sind und darauf warten, dass die Kamera auslöst und so den Wert sehen können. Eine bessere Lösung ist es, die Bilddateien direkt mit der Auslösezeit zu versehen, damit Sie beim späteren Betrachten der Bilder die Zeit oben eingeblendet sehen. Die Codezeile dafür lautet:

```
camera.annotate_text = "time_of_photo"
```

13 Funktion und Aufbau

Noch einmal eine kurze Wiederholung zu Funktion und Aufbau: Das NoIR-Kameramodul erlaubt es, mit dem Pi Fotos im Dunkeln aufzunehmen. Dazu braucht es noch eine Infrarot-Lichtquelle, die das LISIPAROI bietet. Ein PIR-Sensor wird verbunden, um Temperaturänderungen zu erkennen, die LEDs einzuschalten und die Kamera auszulösen. Die aktuelle Uhrzeit wird dem Bild hinzugefügt.

14 Videoaufnahmen

Vielleicht bevorzugen Sie es, dass der Infrarot-Sensor Videoaufnahmen startet; das Programm kann dahingehend angepasst werden. Mit Python sind auch Videoaufnahmen auf einfache Weise möglich, zum Beispiel mit dem Code „camera.start_recording('/home/pi/Desktop/evidence.h264')“, der eine Datei namens „evidence“ speichert. Um 20 Sekunden Video aufzunehmen, geben Sie `time.sleep(20)` ein und stoppen Sie danach die Aufnahme mittels „camera.stop_recording()“.

```
camera.start_preview()
camera.start_recording('/home/pi/Desktop/evidence.h264')
time.sleep(20)
camera.stop_recording()
```

15 Automatischer Programmstart

Wenn Sie Ihr Projekt außen aufbauen, haben Sie vermutlich keinen Monitor angeschlossen und können nicht sehen, was das Programm gerade macht. Lassen Sie es darum einfach automatisch starten, sobald die Stromzufuhr aktiv ist. Geben Sie den Pfad ein, unter dem Sie das Programm für Ihre „Night Box“ gespeichert haben – ist es etwa als `Night_Box.py` in `/home/pi` gespeichert, rufen Sie es mit `/home/pi/Night_Box.py` auf. Kontrollieren Sie noch einmal den Pfad durch folgende Eingabe im Terminal:

```
sudo cat /home/pi/name_of_your_script.py
```

Stimmt alles, wird Ihr Python-Code angezeigt. Legen Sie dann einen Cronjob an, indem Sie die Datei „crontab“ anpassen. Geben Sie „sudo crontab -e“ ein (dabei wird der Cron-Prozess für alle Nutzer gestartet). Scrollen Sie ans Ende des Fensters und fügen Sie folgende Zeile hinzu:

```
@reboot python /home/pi/name_of_your_program.py &
```

Das „&“ am Ende lässt den Code im Hintergrund laufen und stellt sicher, dass Ihr Raspberry Pi normal startet. Speichern Sie die Datei mit Strg+X, gefolgt von Y, starten Sie dann neu.

Vollständiger Code

```
import time
import os
import sys
import picamera
import subprocess
import RPi.GPIO as GPIO
```

```
GPIO.setmode(GPIO.BCM)
###Set up the PIR
PIR = 7
GPIO.setup(PIR, GPIO.IN)
###Set up the LISIPARIO
GPIO.setup(10, GPIO.OUT)
```

```
global File_Number ###number of photo
global file_name ###name of photo
File_Number = 1
```

```
def Nature_selfie(): ###Takes a picture of the wee beastie###
    global File_Number
    global file_name
    ###Grabs current time###
    time_of_photo = time.asctime( time.localtime(time.time()) )
    GPIO.output(10, GPIO.HIGH)
    with picamera.PiCamera() as camera:
        #camera.start_preview()
        time.sleep(0.5)
        camera.annotate_text = time_of_photo
        camera.capture("Nature" + str(File_Number) + ".jpg")
        file_name = "Nature" + str(File_Number) + ".jpg"
        print file_name
        File_Number = File_Number + 1
    GPIO.output(10, GPIO.LOW) ###Turn off LISIPARIO
```

```
def Motion_Sensing(PIR): ###Response to movement###
    print "We see you"
    Nature_selfie()
```

```
###Code to respond to a movement of the wee beastie###
print "Ready to find you"
time.sleep(2)
```

```
try:
    GPIO.add_event_detect(PIR, GPIO.RISING, callback=Motion_Sensing)
    while 1:
        time.sleep(100)
except KeyboardInterrupt:
    print "Quit"
    GPIO.cleanup()
```

16 Stellen Sie die „Night Box“ auf

Um Ihren Aufbau vernünftig verwenden zu können, empfiehlt sich ein Gehäuse, um die Gerätschaften vor Wind und Wetter zu schützen. Vielleicht möchten Sie eines mit dem 3D-Drucker anfertigen, oder Sie finden eine schöne alte Schachtel für die Komponenten. Der Cronjob ist konfiguriert, das Programm startet automatisch, sobald der Raspberry Pi hochgefahren ist – jetzt fehlt nur noch ein geeigneter Ort, um die Box aufzustellen. Schließen Sie den – bevorzugt mobil verfügbaren – Strom an, starten Sie den Pi und Ihr Programm, und bald können Sie betrachten, wer oder was Ihren Garten bei Nacht besucht hat.



Was Sie brauchen

- Raspberry Pi 3
- Bluetooth USB-Dongle (bei einem älteren Pi-Modell)

Den Raspberry Pi nutzen, um Ihr Telefon zu finden

Ein Programm, das Bluetooth-Geräte lokalisiert und auf sie reagiert

Seit dem Raspberry Pi Model 3 gibt es integriertes WLAN und Bluetooth-Funktionen. Damit ist es nun noch einfacher, mit bluetoothfähigen Geräten wie Mobiltelefonen, Tablets und Lautsprechern zu interagieren. Die Programmiersprache Python unterstützt viele Bibliotheken, mit denen Sie mit verschiedenen Elementen von Bluetooth-Geräten interagieren sowie diese überwachen und steuern können. Dieses Tutorial verbindet die Bluetooth-Hardware des Pi mit Python-Code, um drei einfache, aber nützliche Programme zu erstellen. Zunächst coden wir ein kurzes Programm für die Suche nach bluetoothfähigen Geräten, das die 12-teilige Adresse jedes Gerätes zurückliefert. Wenn Sie diese haben, können Sie nach Bluetooth-Diensten scannen, die auf dem jeweiligen Gerät vorhanden sind. Schließlich verwenden Sie die Bluetooth-Adresse und einige Bedingungen, um zu überprüfen, welche Geräte in einem Gebäude vorhanden sind und damit auch welche Personen in einem Gebäude an- bzw. abwesend sind. Unser System reagiert dann entsprechend, es arbeitet praktisch als eine Art automatisches Bluetooth-Check-in-System.

01 Bluetooth verwenden

Bluetooth ist ein drahtloser Standard für den Datenaustausch zwischen Geräten über kurze Distanzen zwischen

einem und zehn Metern. Die aktuelle Version Bluetooth v5 hat eine erhöhte Reichweite von mehr als 200 Metern, wurde 2017 verabschiedet und wird vermutlich die Grundlage für viele Geräte und Anwendungen des Internets der Dinge (Internet of things – IoT) sein. Bluetooth nutzt den Standard IEEE 802.11; das ist der gleiche Standard wie WLAN. Beide teilen Ähnlichkeiten wie das Einrichten einer Verbindung, das Übertragen und Erhalten von Dateien und das Streamen von Audio- und Media-Inhalten. Wenn Sie einen Pi 2 oder niedriger nutzen, können Sie diese Programme trotzdem erstellen und nutzen, indem Sie einen Bluetooth-USB-Dongle verwenden.

Funktioniert Ihr Dongle?

Um einen USB-Bluetooth-Dongle zu überprüfen, stecken Sie ihn ein und starten Sie den Pi durch Eintippen von „sudo reboot“ neu. Prüfen Sie, ob der Dongle erkannt wurde, indem Sie das LX-Terminal-Fenster laden und „lsusb“ eintippen. Das listet alle angeschlossenen USB-Geräte auf. Oder Sie identifizieren den Dongle mittels „hcitool dev“ und zeigen USB-Adresse an.



02 Benötigte Bibliotheken installieren

Obwohl das Image des OS-Image des Raspberry Pi mit einer Bluetooth-Bibliothek für die Anbindung an Geräten kommt, möchten wir für dieses Tutorial die Schnittstelle mit Python-Code steuern. Starten Sie Ihren Pi und öffnen Sie das LX-Terminal-Fenster. Prüfen Sie auf OS-Updates (Zeile 1 und 2). Installieren Sie dann die Python-Entwickertools (Zeile 3) und noch zwei weitere Bluetooth-Entwicklerbibliotheken. Starten Sie den Pi mittels **sudo halt** neu.

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install python-pip python-dev
ipython
sudo apt-get install bluetooth libbluetooth-dev
sudo pip install pybluez
```

03 Python laden

Laden Sie das LX-Terminal und geben Sie **sudo idle** ein. Es öffnet sich der Python-Editor mit Superuser-Privilegien, die Ihnen mittels Python-Code Zugriff auf die USB-Hardware geben. Öffnen Sie ein neues Fenster und importieren Sie das erste Bluetooth-Modul (Zeile 1). Fügen Sie dann eine kurze Nachricht hinzu, um dem Benutzer mitzuteilen, dass das Programm nach nahe gelegenen Geräten sucht.

```
import Bluetooth
print("Suche nach Geräten...")
```

04 Namen der Geräte suchen

Die nächste Codezeile sucht nach den Namen bluetoothfähiger Geräte. Bei jedem Gerät muss Bluetooth aktiviert und es muss für andere sichtbar sein, damit es gefunden werden kann. In der Zeile darunter erstellen wir eine Variable namens **nearby_devices**, um die Namen zu speichern, und nutzen dann den Code **bluetooth.discover_devices**, um die Namen der Geräte in Erfahrung zu bringen.

```
nearby_devices = bluetooth.discover_devices(lookup_names = True)
```

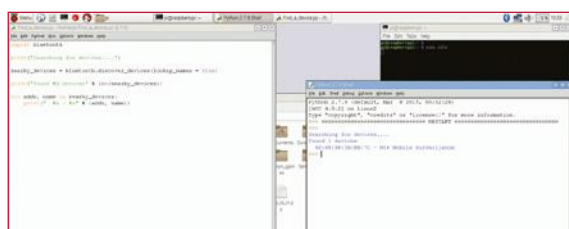
05 Gesamtzahl der gefundenen Geräte

Jeder Name der gefundenen Geräte wird nun in einer Variablen gespeichert. Verwenden Sie den Programmcode **len(nearby_devices)**, um die Anzahl der in der Variablen gespeicherten Geräte zurückzugeben – das ist die Zahl der vom Programm gefundenen Bluetooth-Geräte. Geben Sie dann die Gesamtzahl der Geräte aus. Fügen Sie Ihrem Programm nun noch den folgenden Code hinzu.

```
print("gefunden wurden %d devices" % len(nearby_devices))
```

06 Die Bluetooth-Adresse (BD_ADDR)

Jedes bluetoothfähige Gerät hat eine Bluetooth-



Vollständiger Code

```
import bluetooth

print("Searching for devices...")

nearby_devices = bluetooth.discover_
devices(lookup_names = True)

print("found %d devices" % len(nearby_
devices))

for addr, name in nearby_devices:
    print(" %s - %s" % (addr, name))

# [Find a list of services].py

#!/usr/bin/env python
import bluetooth
from bluetooth import *

device_address = "98:44:98:3A:BB:7C"

#find services on the phone
services = find_service(address=device_
address)
#print services

for i in services:
    print i, '\n'

# [Check to see who is in].py

#!/usr/bin/python
### add a def and then a while statement
import bluetooth
import time

print "Blue-Who Finder"

#find the devices and the name of the device
devices = bluetooth.discover_devices(lookup_
names = True)

#print how many devcies are found
print("Found %d devices" % len(devices))

#print the devices and the names
for addr, name in devices:
    print(" %s - %s" % (addr, name))

time.sleep(2)
print "Check to see who is in the building"
print "Checking " + time.strftime("%a, %d %b
%Y %H:%M:%S", time.gmtime())
time.sleep(1)
if len (devices) == 0:
    print "No one is currently in the
building"

#check the addresses against list to see who
is near
for person in devices:

    device = bluetooth.lookup_
name("68:88:98:3R:BB:7C", timeout=5)
    if (device != None):
        print "TeCoEd is in"
    else:
```

Adresse, die eine Kombination aus zwölf alphanumerischen Zeichen ist, etwa 69:58:78:3A:CB:7F. Die Adressen sind hexadezimal, beinhalten also Zahlen von 0 bis 9 und Buchstaben von A bis F. Die meisten Gerätehersteller drucken die Adresse auf einem Aufkleber an der Hardware oder auf dem Handbuch ab.

07 Name und Adresse des Geräts ausgeben

Für jedes der Geräte, die das Programm gefunden hat, wird dessen Namen und Adresse ausgegeben. Diese Information wird im nächsten Abschnitt benutzt, um herauszufinden, welche Dienste auf welchem Gerät verfügbar sind, und auch um eine Aktion für den Fall hinzuzufügen, wenn ein Gerät gefunden wird. Speichern Sie das Programm und starten Sie es dann. Stellen Sie sicher, dass die Geräte alle sichtbar für andere sind. Sie erhalten eine Liste der Geräte inklusive Namen und Adresse.

```
for addr, name in nearby_devices:
    print(" %s - %s" % (addr, name))
```

08 Verfügbare Dienste auf einem Bluetooth-Gerät herausfinden

Starten Sie das vorherige Programm und notieren Sie die Bluetooth-Adresse des Gerätes. Starten Sie ein neues Python-Programm und speichern Sie es. Öffnen Sie nun ein neues Python-Fenster und starten Sie ein neues Programm. Geben Sie die beiden benötigten Bibliotheken ein (Zeilen 1 & 2).

```
#!/usr/bin/env python
import bluetooth
from bluetooth import *
```

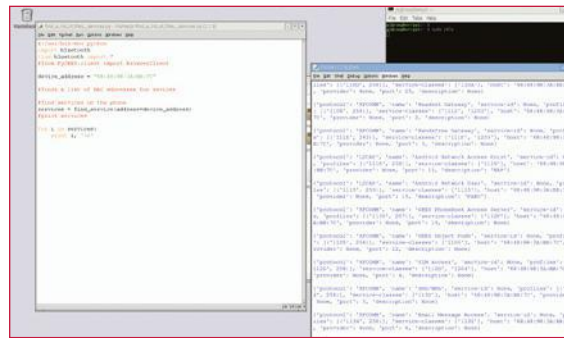
09 Zu findende Adresse festlegen

In der nächsten Zeile darunter geben Sie die Adresse des Gerätes ein, dessen Dienste Sie herausfinden möchten. Nutzen Sie das vorherige Programm oder schauen Sie auf den Aufkleber mit der Adresse. Als Nächstes erstellen Sie eine Variable namens `device_address`, um die Adresse zu speichern. Verwenden Sie den folgenden Code und ersetzen Sie die Beispielsadresse mit der Bluetooth-Adresse Ihres Gerätes.

```
device_address = "69:58:78:3A:CB:7F" # enter address of device
```

10 Einen Dienst finden

In der Zeile darunter fügen Sie den Code ein, um die Dienste zu finden, die Ihr Gerät unterstützt. Erstellen Sie eine neue Variable namens `services`, die eine Liste der Dienste



speichern wird. Nutzen Sie den Befehl `find_services`, gefolgt von der Bluetooth-Adresse Ihres freigegebenen Gerätes, um eine Liste der verfügbaren Dienste zu durchsuchen und jeden davon in der „services“-Variable abzuspeichern.

```
services = find_service(address=device_address)
```

11 Jeden Dienst ausgeben

Der letzte Schritt im Programm ist es, jeden der Dienste auszugeben. Diese werden in einer Liste gespeichert und müssen daher zeilenweise ausgegeben werden. Erstellen Sie eine Code-Schleife `for i in services` (Zeile 1). Diese Schleife prüft jedes einzelne Element in der Liste und gibt dann für jedes der Elemente in der Liste jeden Bluetooth-Dienst aus (Zeile 2). Nutzen Sie den Befehl „`\n`“, um jeden Dienst in einer neuen Zeile auszugeben.

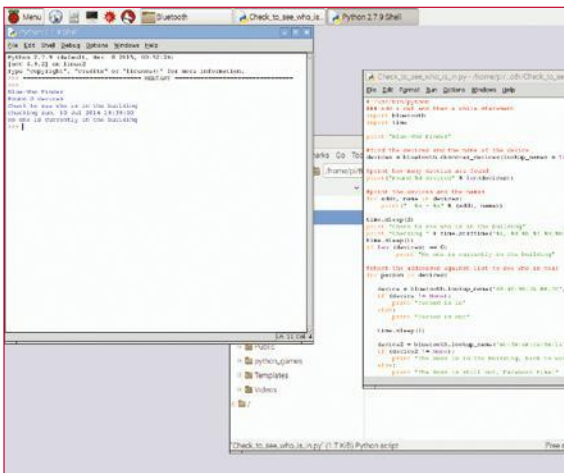
```
for i in services:
    print i, '\n'
```

12 Welche Dienste sind verfügbar?

Speichern und starten Sie Ihr Programm und Sie erhalten eine lange Liste mit Diensten, insbesondere wenn Sie ein modernes Gerät benutzen. Die Verwendung dieser Dienste mit Python ist komplex und erfordert mehrere andere Codezeilen. Allerdings können Sie potenziell Dateien übertragen und empfangen, Musik streamen und sogar das Gerät herunterfahren. Weitere Details und Kommentare finden Sie auf <https://github.com/blueman-project/blueman>. Denken Sie daran: diese Tools sind nur für den persönlichen Gebrauch.

13 Ein Gerät und eine Person finden

In Schritt 7 haben Sie ein kleines Programm genutzt, mit dem Sie die bluetoothfähigen Geräte entdeckt, geprüft



Ist Ihr Bluetooth-Dongle kompatibel?

Wenn Sie ein älteres Raspberry-Pi-Modell wie 1, 2 oder den Pi Zero besitzen, fehlt die Bluetooth-Fähigkeit. Sie können diese aber mit einem USB-Bluetooth-Dongle nachrüsten. Allerdings sind nicht alle Bluetooth-Dongles kompatibel. In der nachfolgend verlinkten Liste sind die darauf getesteten Dongles verzeichnet. Schauen Sie vor dem Kauf also am besten dort nach. http://elinux.org/RPi_USB_Bluetooth_adapters



Variablen

Eine Variable ist ein Ort im Speicher des Computers, in dem Sie Daten speichern können. Um diese wiederzufinden, geben Sie der Variable einen passenden Namen oder Label, etwa Tage = 5. Das bedeutet, dass die Variable „Tage“ aktuell die Zahl fünf enthält.

und deren Adresse zurückgegeben haben. Verwenden Sie nun die Codezeile `bluetooth.lookup_name`, um nach bestimmten Geräten zu suchen und mitzuteilen, ob sie gefunden wurden oder nicht. Wenn das Gerät gefunden wurde, ist es vorhanden – wenn nicht, dann nicht. Bedenken Sie aber, dass Bluetooth auch ein- oder ausgeschaltet sein kann. In Ihrem Python-Programm fügen Sie die Codezeile zum Lokalisieren des Gerätes hinzu. Ersetzen Sie dabei die Beispieladresse mit der des Gerätes, das Sie finden möchten.

```
device = bluetooth.lookup_name("33:38:33:6A:B-Q:7C", timeout=5)
```

14 Antwort, wenn das Gerät gefunden wurde

Sobald ein Gerät gesucht wurde, prüft der Code, ob es vorhanden ist, und gibt Antwort darauf (Zeile 1). Verwenden Sie eine IF-Anweisung, um zu sehen, ob das Gerät nicht gefunden wurde. Diese verwendet das Symbol „!=“, was „ist nicht“ bedeutet. Der Code prüft jedoch, ob es nicht „nicht“ ist – mit anderen Worten: Das Gerät wurde gefunden. Wenn der Code das gewünschte Gerät findet, gibt er eine Meldung aus (Zeile 2). Wird das Gerät nicht gefunden (Zeile 3), erscheint eine Meldung, um den Nutzer zu informieren (Zeile 4). Fügen Sie diese Codezeilen unter Ihren bisherigen hinzu. Stellen Sie sicher, dass Ihr Bluetooth aktiviert ist, speichern und starten Sie das Programm, um Ihr Gerät zu finden.

```
if (device != None):
    print "TeCoEd is in"
else:
    print "Tecoed is out"
```

15 Anderes Gerät finden

Um andere Geräte zu finden und mit einer Aktion zu antworten, nutzen Sie die gleiche Codefolge, erstellen aber erst eine andere Variable, um die Antwort darin zu speichern. Fügen Sie den Code unten ein und benennen Sie die Variable um. Nennen Sie sie etwa `device_one` und bearbeiten Sie die Adresse, damit sie mit dem zweiten Gerät übereinstimmt.

```
Device_one = bluetooth.lookup_name("44:67:73:6T:BR:7A", timeout=5)
```



```
print "Tecoed is out"

time.sleep(1)

device2 = bluetooth.lookup_name('CC:3B:4F:CA:5B:1A', timeout=5)
if (device2 != None):
    print "The Boss is in the building, back to work"
else:
    print "The Boss is still out, Facebook time!"

time.sleep(1)

device3 = bluetooth.lookup_name("00:26:DF:6F:D2:C8", timeout=5)
if (device3 != None):
    print "Wow Sherlock is here O wise one!"
else:
    print "Sherlock is still out on a case"

time.sleep(1)

device4 = bluetooth.lookup_name("28:18:78:47:0C:56", timeout=5)
if (device4 != None):
    print "Babbage is present in the building"
else:
    print "Babbage is not here"

device5 = bluetooth.lookup_name("E0:W8:47:77:6F:41", timeout=5)
if (device4 != None):
    print "We have a Bogie in the area!"
else:
    print "Airspace is clear"
```

16 Weitere Antwort: Nächstes Gerät gefunden

Auch hier prüfen und antworten wir in Zeile 1 mit einer IF-Anweisung, um zu sehen, ob das Gerät gefunden wurde. Denken Sie daran, den neuen Variablennamen zu verwenden – in diesem Beispiel `device_one`. Wird das benannte Gerät gefunden, wird eine Meldung ausgegeben (Zeile 2). Wird das Gerät nicht gefunden (Zeile 3), erfolgt die Ausgabe einer Meldung an den Nutzer (Zeile 4). Fügen Sie diese Codezeilen unter den vorherigen Programmzeilen hinzu. Speichern und starten Sie das Programm, um die beiden bestimmten Geräte innerhalb Ihres Standorts zu finden.

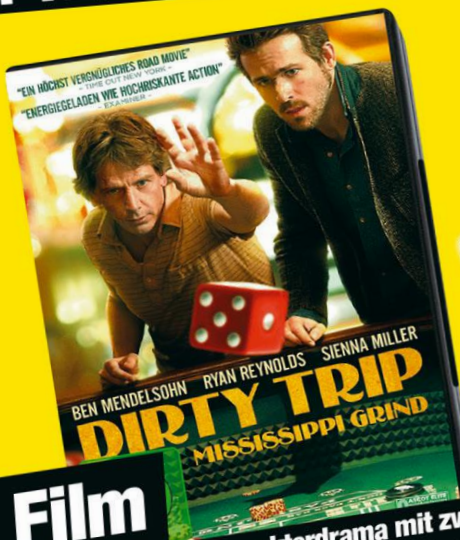
```
if (device_one != None):
    print "Linux Laptop is in"
else:
    print "Linux Laptop is out"
```

17 Alternative Aktion hinzufügen

Sie können natürlich weitere eigene Aktionen hinzufügen, falls Geräte gefunden werden. So könnte etwa jedes Mal, wenn ein Gerät den Standort betritt und erkannt wird, ein LED-Streifen aufleuchten. Oder Sie verwenden einzelne LEDs für jedes Gerät, um anzuzeigen, ob das Gerät an- oder abwesend ist. Oder Sie kombinieren den Code mit einem LCD-Bildschirm, der anzeigt, wer anwesend ist und wer nicht. Weitere Ideen: [youtube.com/watch?v=qUZQv87GVdQ](https://www.youtube.com/watch?v=qUZQv87GVdQ)

TOP-FILM AUF HEFT-DVD

SPIELE | FILME | TECHNIK



Film

„Einnehmendes Charakterdrama mit zwei herausragenden Hauptdarstellern“ (filmstarts.de)



Doppelfolge

„Genre-Fans schätzen die anspruchsvolle Story, die Platz für Tiefgang lässt.“ (kino.de)



159. AUSGABE
02/18 | Februar

€ 5,90

Erhältlich
auch ohne
DVD für € 4,90

Deutschland € 5,90;
Österreich € 6,50;
Schweiz sfr 9,90;
Holland, Belgien,
Luxemburg € 6,80;
Frankreich, Italien,
Spanien, Portugal,
Griechenland € 7,80

Kraft-
protz
für unter-
wegs

Microsofts erstes Surface Book stellte einen gelungenen Hybriden zwischen leistungsstarkem Tablet und vollwertigem Notebook dar. Die zweite Version soll noch besser werden. Ob das klappt, haben wir im Praxistest herausgefunden.



Top-Deals der Technikwelt

BEI DIESEN HIGHLIGHTS AUS 2017 KÖNNEN SIE BIS ZU 40 PROZENT SPAREN!

Kompaktes Klang-Upgrade

TEST Wenn der TV-Sound zu lasch tönt, muss eine Soundbar her! Wir nehmen fünf aktuelle Klangstangen unter die Lupe.



Taschenspieler

TEST Razer lädt mit seinem ersten Smartphone zum Spielen ein, während sich Huawei beim Mate 10 Pro mit KI durchsetzen will. Ob das gelingt?



TEST: FENSTER IN DIE VIRTUELLE WELT

Ob ultrahochauflösend, pfeilschnell oder extrem farbtreu: Wir testen sechs Monitore, die für Profizocker, Cineasten und Schreibtischhengste die perfekte Darstellung liefern wollen.



WEITERE HIGHLIGHTS

- Audio-Mythen im Check
- Samsung-TV 55MU9009
- NAS-Speicher
- TPCast für HTC Vive
- Shape of Water
- Blade Runner 2049
- Playerunknown's Battlegrounds
- Okami HD



DAS TEST-MAGAZIN FÜR DIGITALE UNTERHALTUNG

DAS BESTE AUS ALLEN TECHNIK-WELTEN:

Flat-TVs | Smartphones | Tablets | Notebooks | Digitalkameras | Video | HiFi
plus: **DIE COOLSTEN SPIELE** und **ALLE FILM-BLOCKBUSTER**

2 TOP-MOVIES
AUF HEFT-DVD



◀ Auch als
Magazin-Variante
ohne DVD für € 4,90



WWW.SPIELEFILMETECHNIK.DE

SFT bequem online bestellen:
shop.spielefilmeteknik.de

Oder einfach digital lesen:
epaper.spielefilmeteknik.de



**PC Games Hardware – Das IT-Magazin für
Gamer. Immer aktuell mit Kaufberatung,
Hintergrundartikeln und Praxistipps.**



HARDCORE FÜR SCHRAUBER

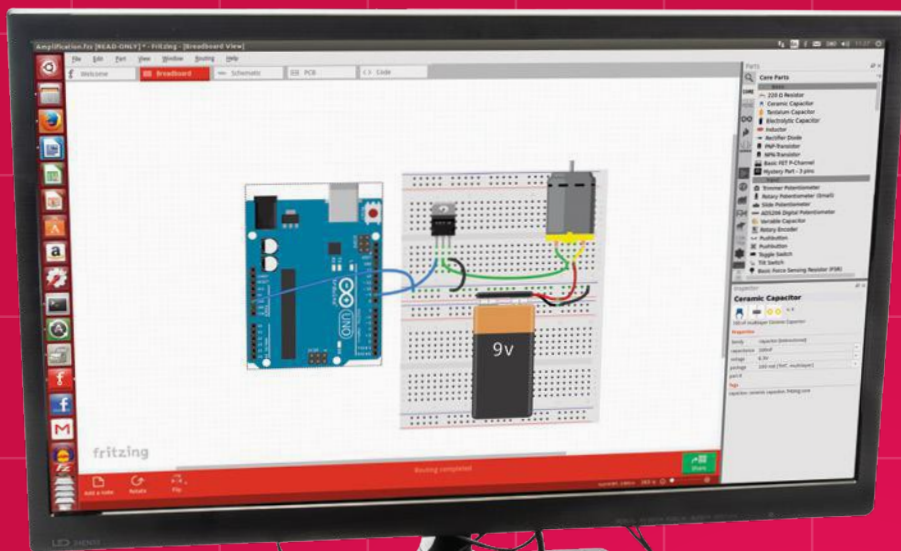


WWW.PCGAMESHARDWARE.DE

PC Games Hardware bequem online bestellen:
www.pcgh.de/shop



Oder einfach digital lesen:
epaper.pcgameshardware.de



Mehr als
200
Praxis-
Tipps

Neue Projekte für den Ein-Platinen-PC:

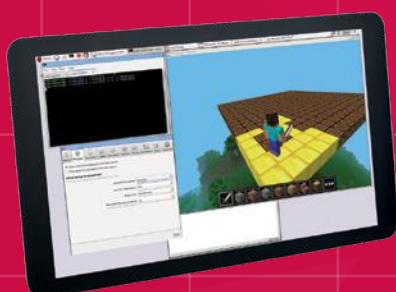
Raspberry Pi

Tipps, Tricks & Hacks



✓ Für Einsteiger

Es ist nicht schwer, die ersten Schritte mit dem Mini-Rechner zu machen.



✓ Für Fortgeschrittene

Viele spannende Projekte warten nur darauf, in die Tat umgesetzt zu werden.



✓ Für Profis

Bauen Sie auf Basis des Raspberry Pi starke Hardwarelösungen.

