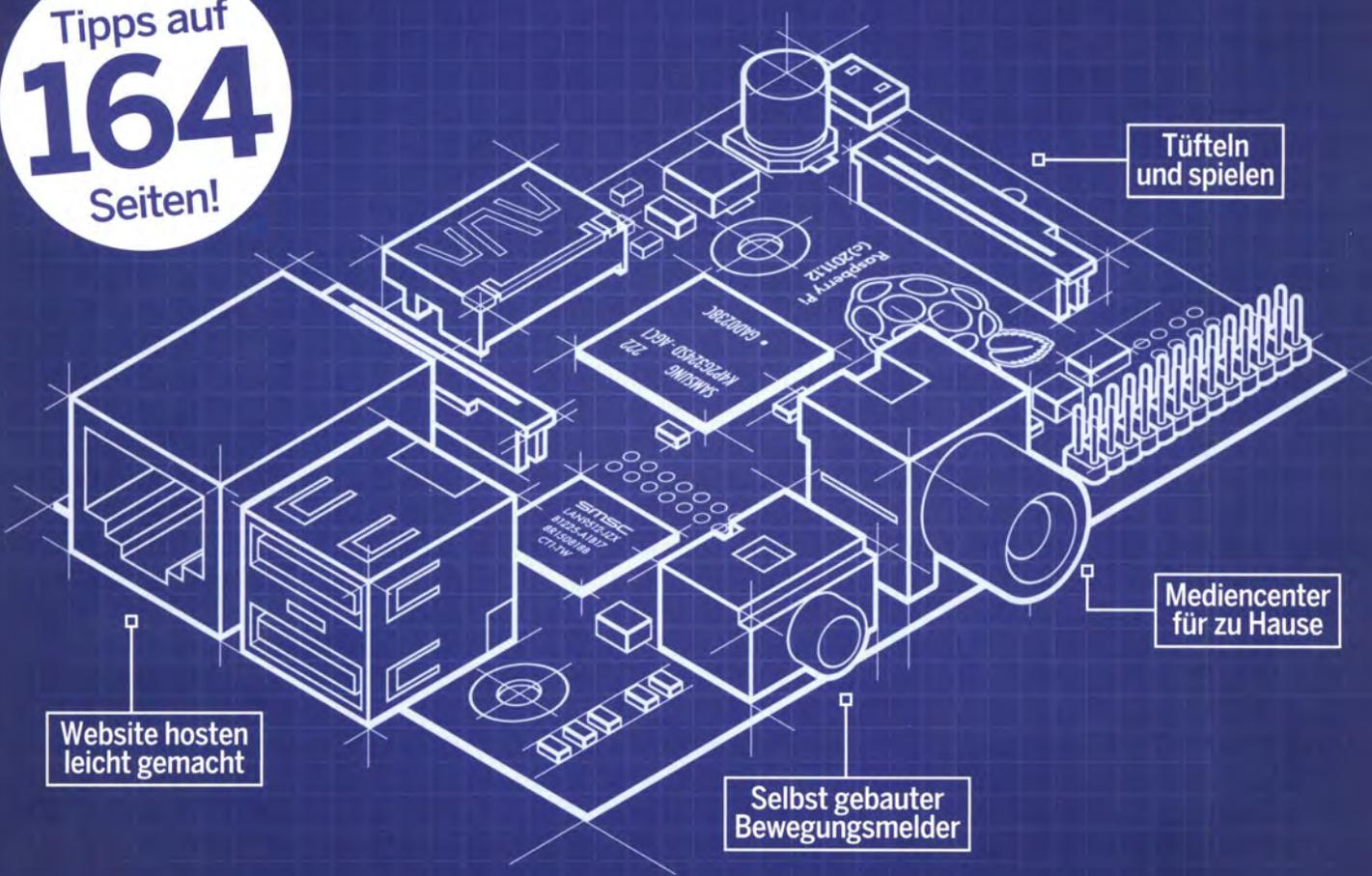


# Raspberry Pi

## 50 COOLE PROJEKTE

Tipps auf  
**164**  
Seiten!



### Erkunden Sie den vielseitigen Mini-PC

**Software**

Stellen Sie Ihr eigenes Betriebssystem zusammen.

**Coding**

Programmieren lernen ist leicht mit Scratch.

**Projekte**

Anleitungen und Tutorials zeigen, wie es geht.

**Hardware**

Lernen Sie die tolle Welt der Elektronik kennen.

Experten-Wissen für Einsteiger und Fortgeschrittene

computec  
EDITION



PCGH GUIDE 05/15  
**€ 9,99**  
Österreich € 11,-  
Schweiz CHF 17,-  
Benelux € 11,50

# DIE APP FÜR SPIELEFANS

JETZT ENDLICH FÜR ANDROID & iOS!

# GAMES TV 24

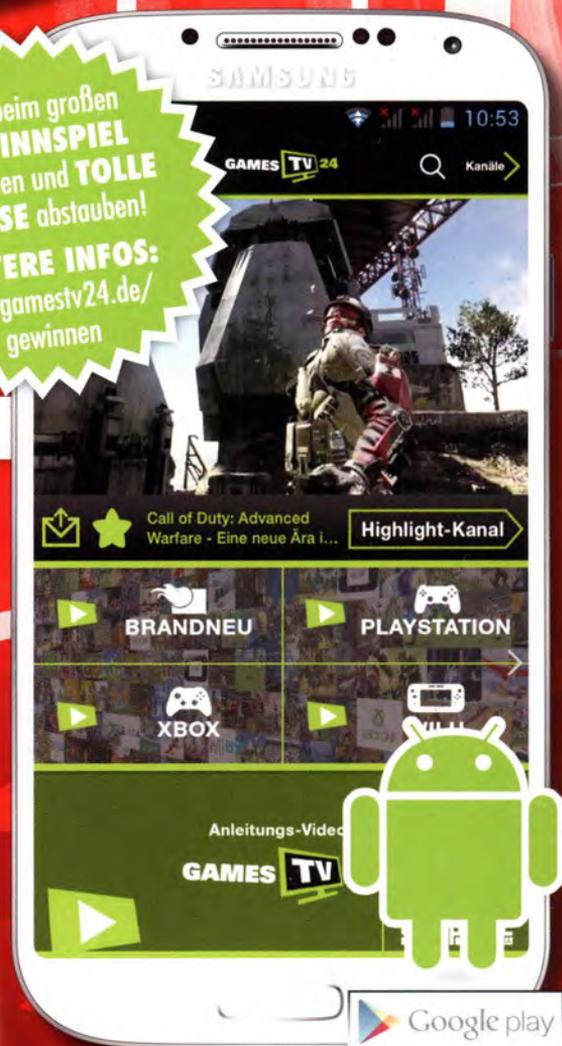
KOSTENLOS!

IMMER DABEI, SOFORT STARTKLAR:  
TÄGLICH NEUE VIDEOS ZU SPIELEHITS  
AUF ALLEN SYSTEMEN!

Jetzt beim großen  
**GEWINNSPIEL**  
mitmachen und **TOLLE**  
**PREISE** abstauben!  
**WEITERE INFOS:**  
[www.gamestv24.de/  
gewinnen](http://www.gamestv24.de/gewinnen)

ÜBER 10.000 VIDEOS IN DER DATENBANK.

Auf iOS inklusive  
FLIP-Stop-Motion-Videos



++ SPIELETRAILER, TESTVIDEOS, MESSESPECIALS UND WÖCHENTLICHE SHOW  
ZUR LET'S-PLAY-SZENE! ++ BILDSCHIRMFÜLLENDE VIDEOS MIT APPLE TV ++





**Lizenz**

Artikel in dieser Ausgabe sind übersetzt oder reproduziert aus „Raspberry Pi - Projects“ und unterliegen dem copyright von Future plc, UK 2014. Inhalte wurden als Lizenz erworben. Alle Rechte vorbehalten. „Bookazines and Masterclass series“ ist ein eingetragenes Warenzeichen von Future Publishing Limited. Weitere Informationen über diese und andere Zeitschriften der Future plc Gruppe: <http://www.futurepic.com>

**Licence**

Articles in this issue are translated or reproduced from „Raspberry Pi - Projects“ and are the copyright of or licensed to Future Publishing Limited, a Future plc group company, UK 2014. Used under licence. All rights reserved. „Bookazines and Masterclass series“ is the trademark of or licensed to Future Publishing Limited, a Future plc group company. Used under licence. <http://www.futurepic.com>

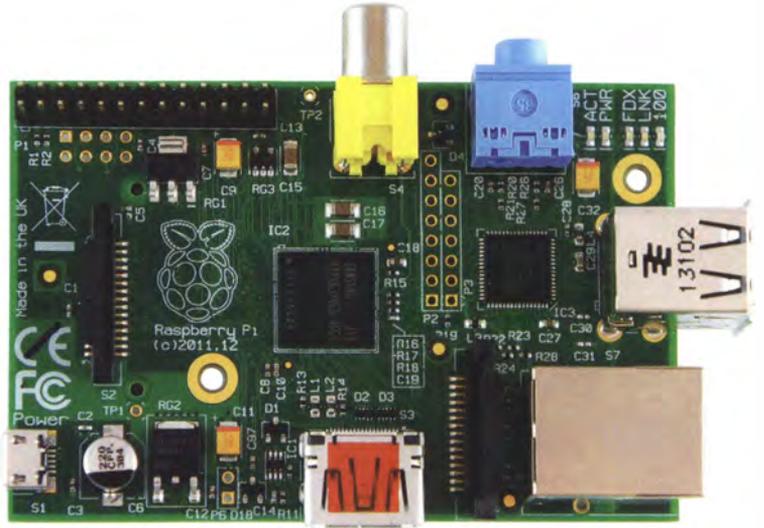


MARQUARD MEDIA INTERNATIONAL AG

Deutschsprachige Titel: SFT, WIDESCREEN, PC GAMES, PC GAMES MMORE, PC GAMES HARDWARE, BUFFED, X3, PLAY 4, GAMES & MORE, GAMES AKTUELL, N-ZONE, XBG, GAMES, LINUX, LINUX-MAGAZIN, LINUXUSER, EASYLINUX, RASPBERRY PI GEEK

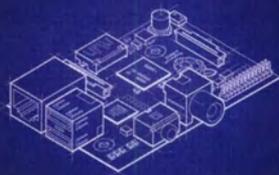
Internationale Zeitschriften: Ungarn: JOY, SHAPE, EVA, IN STYLE, PLAYBOY, CKM, MEN'S HEALTH  
Polen: COSMOPOLITAN, JOY, SHAPE, HOT, PLAYBOY, CKM, VOYAGE, HARPER'S BAZAAR

# Willkommen!



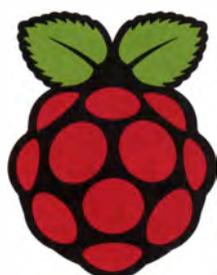
**D**as Raspberry Pi ist ein kleines, simples Gerät, das auf den ersten Blick ziemlich unscheinbar wirkt. Doch ist es gerade seine Einfachheit, die den Mini-Computer so vielseitig macht. Schließlich können Sie ihn für eine ganze Reihe von Aufgaben benutzen. Alles was Sie benötigen, sind ein herkömmlicher PC, die Bereitschaft, ein wenig Programmieren zu lernen, und Vorstellungskraft. Mit diesem Buch möchten wir Ihre Vorstellungskraft ankurbeln und Sie ermutigen, Ihren Raspberry Pi auf kreative Art und Weise zu nutzen: Von der Emulation Ihrer Lieblings-Retro-Konsolen-Spiele bis hin zur Nutzung des Raspberry Pi als Hausalarmanlage – wir haben dieses Buch vollgepackt mit spannenden Projekten, die Sie inspirieren werden. Nicht nur, dass Raspberry-Pi-Projekte spaßig und nützlich sind, sie helfen Ihnen auch, die Welt des Programmierens und der Elektronik besser zu verstehen. Zudem können Sie als Grundlage für Ihre eigenen, komplexeren Projekte dienen. Die meisten dieser Projekte können ganz einfach mit dem Mini-Computer selbst und dem vorinstallierten Betriebssystem Raspbian realisiert werden. Allerdings sehen wir uns auch einige Hardware-Add-ons an, mit denen Sie die Einsatzmöglichkeiten des Raspberry Pi erweitern können, sodass man sich mit ihm auch richtig verzwickten Aufgaben widmen kann. Bei den Tutorials ist hin und wieder etwas Bereitschaft zum Programmieren gefragt. Aber keine Sorge, falls Sie in dieser Hinsicht wenige oder keine Vorkenntnisse besitzen. In unseren Workshops vermitteln wir Ihnen die Programmierkenntnisse, die Sie zur Umsetzung der Projekte benötigen. Zudem finden Sie bei uns auch eine eigene Sektion zu diesem Thema, die Ihnen die wichtigsten Grundlagen näher bringt. Dieses Buch eröffnet Ihnen völlig neue Anwendungsmöglichkeiten für das Raspberry Pi, während Sie gleichzeitig wertvolle Erfahrungen im Programmieren sammeln. Wir hoffen, dass Sie mit dem Mini-Computer nicht nur viele spannende Projekte umsetzen, sondern auch eine Menge Spaß haben werden.

In diesem Sinne: Lassen Sie Ihrer Kreativität freien Lauf!



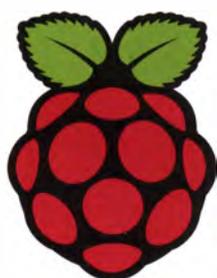
# Inhalt

## Betriebssystem



Alternativen zu Raspbian.....	<b>8</b>
Distributionen à la carte.....	<b>12</b>
Desktop: Eigene Schreibtische.....	<b>16</b>
System: Eigene Distribution.....	<b>20</b>

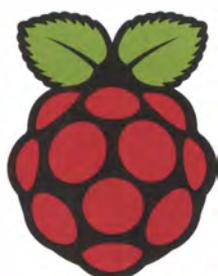
## Programmieren



Scratch und Python.....	<b>26</b>
Scratch: Quiz.....	<b>30</b>
Scratch: Arcade.....	<b>34</b>
Scratch: Gejagt.....	<b>38</b>
Die Legende von Python.....	<b>42</b>
Chat: Wir bauen Bots!.....	<b>46</b>
Ein grafisches Interface bauen.....	<b>50</b>
Twitter: Tweets vorlesen lassen.....	<b>54</b>
Python: Hacken Sie Minecraft.....	<b>58</b>
Minecraft: Das Pi-Katapult.....	<b>60</b>
Wir bauen ein Plattformspiel.....	<b>64</b>
Logik lernen mit Guido.....	<b>68</b>

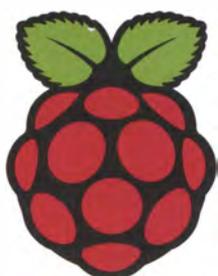


# Projekte



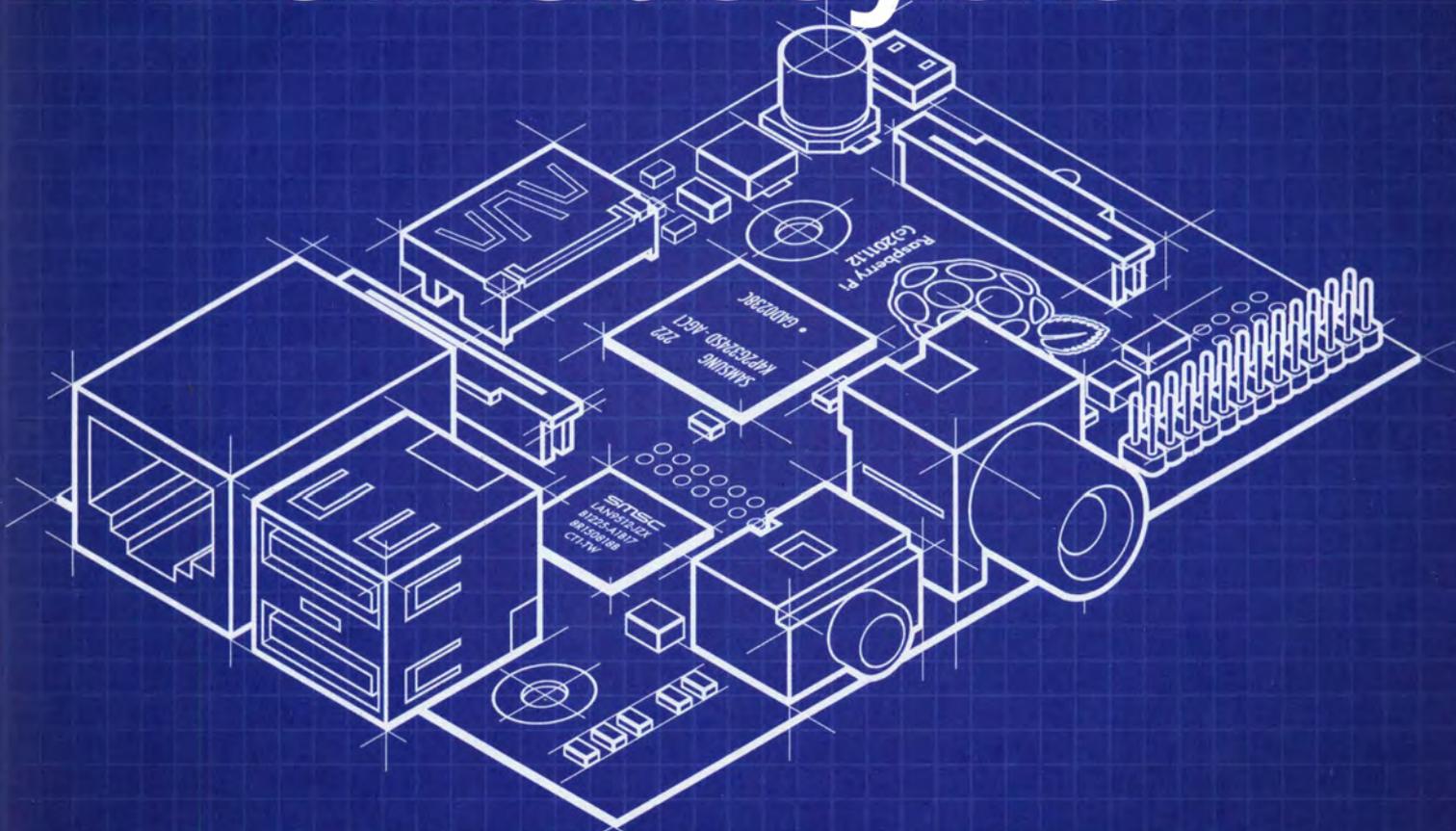
Cool für die Schule.....	<b>74</b>
Zünde den Raspberry-Booster.....	<b>78</b>
TV-Streaming mit dem Pi.....	<b>86</b>
Heimkino-PC.....	<b>90</b>
Musik-Server mit Mopidy.....	<b>94</b>
Retro-Gaming mit dem Pi.....	<b>98</b>
Seien Sie Ihr eigener Hoster.....	<b>102</b>
Der eigene Cloud-Server.....	<b>106</b>
Torrent-Server einrichten.....	<b>110</b>
Anonym bleiben mit Tor.....	<b>114</b>
Ein Netzwerk aus 2 x Pi.....	<b>118</b>
Server: Chatten via IRC.....	<b>122</b>
Streaming aufs Smartphone.....	<b>126</b>
Hosting eines Foto-Servers.....	<b>130</b>
Elektronische Infotafeln.....	<b>134</b>
Der Pi als Alarmanlage.....	<b>138</b>

# Hardware

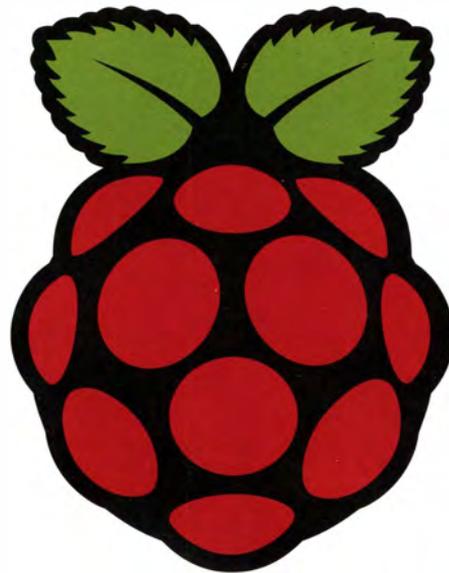


Einstieg in die Welt des Raspberry Pi.....	<b>144</b>
Erweitern Sie den Raspberry Pi.....	<b>152</b>
Compute Module Development Kit.....	<b>156</b>
3-D-Druck: Pi-Case printen.....	<b>158</b>
Wolfson Audiokarte.....	<b>162</b>

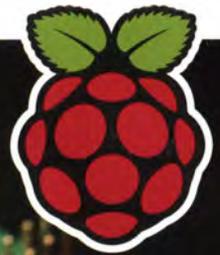
# Raspberry Pi Betriebssystem



**D**as Betriebssystem ist sehr wichtig, denn es bietet einen Zugang zu den Funktionen des Raspberry Pi. Das Standard-Betriebssystem für den Raspberry Pi ist Raspbian. Obwohl es auf Linux basiert, ist es dennoch für Anwender anderer Betriebssysteme wie beispielsweise Windows oder Mac OS X vertraut. Dieses Kapitel zeigt Ihnen, wie Sie den Pi konfigurieren und sichern. Wir stellen auch alternative Betriebssysteme vor.



Alternativen zu Raspbian.....	<b>8</b>
Distributionen à la carte.....	<b>12</b>
Desktop: Eigene Schreibtische.....	<b>16</b>
System: Eigene Distribution.....	<b>20</b>

A close-up photograph of a Raspberry Pi 2 Model B board. The board is green and populated with various components including a USB Type-A port, a micro-USB port, a 3.5mm audio jack, and a 40-pin GPIO header. A yellow Ethernet cable is plugged into the board. To the right of the board, a black SanDisk Ultra II 2.0 SD card is visible. The background is dark, making the components stand out.

# Alternativen zu Raspbian

Hier stellen wir Ihnen einige alternative Betriebssysteme für den Raspberry Pi vor, denn es gibt bei weitem nicht nur Raspbian zur Auswahl.

**D**er Raspberry Pi repräsentiert **Kreativität, Freiheit und Offenheit – all das, wofür Linux und die Open-Source-Bewegung stehen.**

Der Phantasie sind wohl kaum Grenzen gesetzt, wenn es um spannende Anwendungen für diesen wunderbaren, extrem vielseitigen Minicomputer geht: Der Raspberry Pi ist schon bis in die Stratosphäre aufgestiegen, ins Meer abgetaucht, zum Oldschool-Arcade-Spielautomaten umgebaut und als Steuerung in einem Roboterfahrzeug mit Nachtsichtgerät eingesetzt worden. Gehäuse aus Plastik, Holz, Pappe, Metall, Gips und sogar Legosteinen sind für den Pi angefertigt worden. Der Rechner tut in Maschinenbaufirmen, Schulen und Universitäten seine Dienste.

Die meisten Pi-Enthusiasten arbeiten jedoch vermutlich zu Hause mit dem Gerät,

um ihre Kenntnisse über Linux, Programmieren, Netzwerktechnik und Elektronik zu erweitern und zu vertiefen. Diese Dinge werden enorm dadurch vereinfacht, dass ein Betriebssystem speziell für den Raspberry Pi entwickelt wurde: Raspbian, eine Variante der Linux-Distribution Debian. Raspbian

**„Raspbian ist längst nicht das einzige Betriebssystem für den Raspberry Pi.“**

haben wohl die meisten Pi-Anwender auf dem Rechner installiert.

Allerdings ist es bei weitem nicht die einzige Alternative. De facto gibt es eine ganze Menge Raspberry-Pi-spezifische Betriebssysteme, von denen viele auf einen bestimmten Aspekt des Computers ausgerichtet sind (etwa in

den Bereichen Medienwiedergabe, Alternativsysteme, Retro-Spiele oder EDV).

Eines der bekanntesten davon ist Raspbmc: Der Raspberry Pi wird gerne als Mediacenter verwendet. Dank seines HDMI-Anschlusses, der angemessenen Prozessorleistung, der geringen Größe und insbesondere seines geräuschlosen Betriebs hat der Raspberry Pi ein Plätzchen unter manch einem TV-Gerät gefunden. Zwar waren die Möglichkeiten anfangs ein wenig eingeschränkt, doch dank der rührigen Pi-Community hat sich das schnell

geändert: Das Betriebssystem Raspbmc macht das XBMC Media Center auf dem Raspberry Pi verfügbar.

Auf den folgenden Seiten möchten wir Ihnen noch drei andere alternative Betriebssysteme für den Raspberry Pi vorstellen: RasPlex, RISC OS und RetroPie.

# RasPlex

Verwandeln Sie den Raspberry Pi in ein Mediencenter.

Zunächst brauchen Sie eine installierte Version des Plex Media Server, der zwar eine proprietäre Software ist, aber kostenlos verwendet werden kann. Es stehen eine Reihe Plattformen zur Auswahl: Windows, OS X, Linux, NAS, FreeBSD und sogar einige mobile Betriebssysteme wie Android, iOS und Windows Phone. Falls Sie im Hinblick auf den Server-Aspekt Probleme bei der Installation haben, finden Sie unter [goo.gl/rld6P5](http://goo.gl/rld6P5) eine Schritt-für-Schritt-Anleitung.

Wenn der Server installiert ist, ordnen Sie Ihre Mediensammlung so, dass Plex die einzelnen Dateien erkennt – auch hierfür gibt es eine Anleitung auf der Website. Danach können Sie zu RasPlex übergehen. Sie benötigen für Ihren Raspberry Pi außer der gewöhnlichen Peripherie wie TV-Gerät/Monitor, Tastatur und Maus



› Das erste Hochfahren von RasPlex kann sehr lange dauern. Sie können Abhilfe schaffen, indem Sie den Cache „vorwärmen“. Wie das geht, steht im Text.



› Mit dem RasPlex-Installer haben Sie das Betriebssystem im Handumdrehen auf eine SD-Karte gepackt.

außerdem noch eine SD-Karte (am besten Class 10 oder höher) mit mindestens 2 GB Speicherplatz sowie eine Verbindung mit Ihrem heimischen Netzwerk.

Noch ein Wort zu den Modellvarianten des Raspberry Pi: Zwar läuft RasPlex auch auf der ersten Version des Computers mit 256 MB Arbeitsspeicher halbwegs okay, aber die neuere 512-MB-Version ist natürlich besser geeignet. RasPlex befindet sich aktuell noch in einer relativ frühen Entwicklungsstufe und wurde bisher im Hinblick speziell auf den 512-MB-Pi konzipiert.

## Installationsprogramm

Es gibt drei Installationsprogramme für RasPlex – eins für Windows, eins für OS X und eins für Linux. Alle drei stehen unter [goo.gl/8LPXc9](http://goo.gl/8LPXc9) zur Verfügung. Die Installer bieten eine schlichte grafische Benutzeroberfläche, über die man ein Image

(Speicherabbild) herunterladen und auf die SD-Karte überspielen kann. Sie brauchen dazu lediglich die SD-Karte einzuschieben, das Installationsprogramm auszuführen, die gewünschte Version von RasPlex herunterzuladen (wir empfehlen immer die jeweils aktuellste), das heruntergeladene Image auszuwählen und auf „Write to SD Card“ zu klicken – schon wird das Betriebssystem auf die SD-Karte überspielt. Danach brauchen Sie nur noch die Karte in den Raspberry Pi einzulegen und den Rechner zu booten, und er fährt mit RasPlex hoch.

## Cache vorwärmen

Das erste Booten mit RasPlex dauert extrem lang, speziell beim 256-MB-Pi scheint die Benutzeroberfläche geradezu eingefroren zu sein. Dies ist normal und hat mit dem Zustand des Cache zu tun – RasPlex speichert hier Bilder, Thumbnails und Metadaten. Sie können eine Verbesserung erzielen, indem Sie den Cache „vorwärmen“, das heißt in diesem Fall, die Optionen „All > All Videos“, „All > All TV Shows“ und so weiter zu aktivieren. Damit zwingen Sie RasPlex, zukünftig die jeweiligen Daten beim Hochfahren zu cachen. Diese Mühe kann sich auszahlen, da Sie anschließend ein deutlich schneller ansprechendes RasPlex auf Ihrem Raspberry Pi haben.

## Community

Da RasPlex sich wie erwähnt noch relativ weit am Anfang der Entwicklung befindet, kann die Community Unterstützung gut gebrauchen. Unter [goo.gl/1vlsUC](http://goo.gl/1vlsUC) finden Sie Hinweise, wie Sie einen Beitrag zu dem Projekt leisten können, falls Sie möchten.

## XBian

RasPlex setzt auf der bereits auf Seite 8 erwähnten Home-Theater-Software XBMC auf. Für diejenigen, die das Original vorziehen, gibt es noch eine andere Lösung für den Raspberry Pi, nämlich XBian. Dieses Betriebssystem basiert – wie auch Raspbmc – direkt auf Raspbian, wengleich in einer abgespeckten Version, und verbindet es mit XBMC.

Das Bundle ist durchaus beeindruckend und wartet

überdies mit einer großen Community auf. Um XBian auf dem Pi zu installieren, gehen Sie auf [xbian.org](http://xbian.org) und da in den Downloadbereich. Dort finden Sie die notwendigen Infos.

XBian besitzt einige Vorteile gegenüber RasPlex. Zum einen wird bei XBian kein spezieller Server benötigt wie bei Plex, ein geeignetes NAS-Laufwerk, Medienserver oder USB-Laufwerk genügt. Zum

anderen verbraucht XBian weniger Systemressourcen und unterstützt von Hause aus eine Menge Wireless-Network-Adapter. Und drittens läuft die Benutzeroberfläche sehr geschmeidig, denn XBian ist eine schlanke Software, die auf eine 2-GB-SD-Karte passt.

Am Ende zählt natürlich der persönliche Geschmack. Wir raten dazu, beide Systeme mal zu testen.

# RISC OS

Spielen Sie den Achtziger-Jahre-Klassiker Elite auf dem Raspberry Pi.

**R**etro ist in Mode, auch im Computerbereich. Man kann kaum den Fernseher einschalten, im Internet surfen oder in eine Zeitschrift schauen, ohne irgendwas Tolles aus der Zeit zu entdecken, in der man selbst die ersten Erfahrungen mit dem PC machte. Ein absoluter Klassiker ist das 1984 erschienene Science-Fiction-Computerspiel Elite.

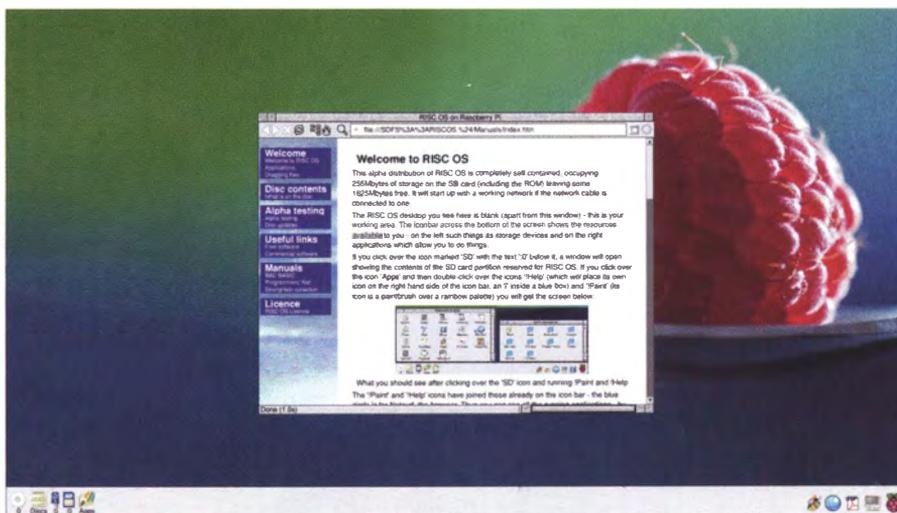
Elite fanden ziemlich viele Spieler ziemlich cool, und wenn man es heute auf einem der Originalsysteme wie etwa dem C64, dem Amiga oder dem Atari ST zockt, können einem schon sehr nostalgische Gefühle kommen – sofern man der entsprechenden Generation angehört natürlich.

Wir haben mal versucht, Elite auf dem Raspberry Pi zum Laufen zu kriegen (siehe Kasten). Dazu haben wir das – ähnlich ehrwürdige – Betriebssystem RISC OS von Acorn verwendet.

## Was ist RISC OS?

RISC OS wurde erstmals 1987 veröffentlicht. Damals hieß es noch Arthur 1.20, und es war von der englischen Firma Acorn für deren ARM-basierte Heimcomputer-Serie Archimedes entwickelt worden. Für den Raspberry Pi ist eine kostenlose Grundversion, aber auch ein Software-Bundle namens NutPi für umgerechnet etwa 45 € erhältlich. NutPi enthält viele Extras und genügend Programme, um aus dem RISC-OS-Pi eine sehr brauchbare Alltagsmaschine zu machen.

Die aktuelle Version von RISC OS für den Raspberry Pi trägt die Nummer 12.



› RISC OS ist funktional, schön, schnell und retro – alles auf einmal.

Zur Standardausstattung gehören ein Webbrowser, ein Texteditor und sogar BBC BASIC – wer keine Lust mehr auf Elite hat, kann sich also theoretisch sein eigenes Computerspiel programmieren. Das gut 100 MB große Image für die SD-Karte können Sie von [goo.gl/31bVrf](http://goo.gl/31bVrf) herunterladen. Entzippen Sie die Datei und verwenden Sie anschließend den Win32DiskImager für Windows, das Tool „RPi-sd card builder“ für OS X beziehungsweise den Befehl **dd** für Linux.

Beim Booten werden Sie feststellen, dass RISC OS recht nett aussieht, insbesondere wenn Sie einen anständigen 1080p-Monitor an den Raspberry Pi angeschlossen haben,

denn das Betriebssystem kommt erst mit der entsprechenden Anzeige zu vollem Glanz und Gloria. Ein kleiner Schönheitsfehler: Unter RISC OS müssen Sie die Netzwerkfähigkeit des Pi manuell aktivieren, denn sie ist standardmäßig ausgeschaltet. Schließen Sie dazu die Fehlermeldung des Browsers NetSurf und doppelklicken Sie mit der linken Maustaste auf das „Configure“-Icon auf dem Desktop. Anschließend klicken Sie das „Netzwerk“-Icon und dann das „Internet“-Icon an. Wählen Sie sodann „Enable TCP/IP protocol suite“ aus und sichern Sie die Einstellung. Zuletzt rebooten Sie mit „Reset Now“.

## Elite und Emulatoren

Da Arc Elite für ARM2- und ARM3-basierte Rechner programmiert wurde, läuft es nicht ohne weiteres in der ARM6-Umgebung des Raspberry Pi. Daher müssen wir eine ältere Version von RISC und ARM mithilfe von ArcEm emulieren.

Statt sich den Emulator und die zugehörigen ROMs einzeln zu beschaffen, können Sie von [arcem.sourceforge.net](http://arcem.sourceforge.net) eine Zip-Datei namens ArcEm.zip herunterladen, die bereits alles enthält, was Sie brauchen, um Elite zu emulieren. Wenn Sie allerdings direkt auf die Datei !ArcEm klicken, öffnet sich stattdessen StrongArm. Entpacken Sie die Zip-Datei daher auf einen USB-Stick, von dort können Sie

RISC OS direkt laden – es muss nicht zwangsläufig auf dem Gerät installiert sein. Schließen Sie den USB-Stick an den Raspberry Pi an und warten Sie, bis RISC OS das neue Gerät erkennt. Klicken Sie auf den Dateimanager, gehen Sie dann mit NetSurf auf [goo.gl/6TIOE](http://goo.gl/6TIOE) und laden Sie die neueste



› Eine Zeitreise in das Jahr 1984!

ArcEm-Datei herunter. Es handelt sich um ein Zip-File, das Sie mit der RISC-OS-Anwendung !sparkFS öffnen müssen. Klicken Sie auf das SD-Karten-Icon unten links, um den Inhalt des Dateisystems zu öffnen, und doppelklicken Sie danach auf das Verzeichnis „Utilities“, Suchen Sie das Programm !sparkFS und doppelklicken Sie darauf. Es sollte nun neben den Icons unten links im Fenster erscheinen. Ziehen Sie die eben heruntergeladene Datei arcem-1.50-alpha2-riscos auf das !sparkFS-Icon. Sodann erscheint ein Fenster mit Inhalten, die per Drag & Drop zu den Originalinhalten der ArcEm-Zip-Datei gezogen werden sollen. Dadurch werden die Datei

!ArcEm, Ordner hostFS sowie eine Textdatei durch solche ersetzt, die in dieser Instanz funktionieren. Navigieren Sie nun zu [goo.gl/wEGK2](http://goo.gl/wEGK2) und laden Sie die Datei b5052410.arc herunter. Ziehen Sie auch diese auf das !sparkFS-Symbol und die entzippten Inhalte in den Ordner hostFS. Um Elite zu starten, doppelklicken Sie auf die neue Datei !ArcEm, welche die alte Archimedes-Plattform lädt, und doppelklicken Sie dann auf das Laufwerk HostFS. Nun sollte Elite laufen! Falls es nicht gleich klappt, doppelklicken Sie das Icon !elite. Wenn Sie gefragt werden, ob Sie einen hochauflösenden Monitor verwenden, klicken Sie Y.

# RetroPie

Der Minicomputer als klassische Spielekonsole.

**D**a wir schon beim Thema Spieleklassiker sind: Wie wäre es, aus dem Raspberry Pi eine Retro-Konsole zu machen?

Während antike Games nicht für jeden das Richtige sind, hat es für andere durchaus seinen Reiz, beispielsweise Doom auf einem Rechner zu spielen, der auf der Handfläche Platz findet. Ganz nebenbei ist das auch ein großartiges Bastelprojekt, das man an verregneten Wochenenden angehen kann.

RetroPie ist ein vollständiges Betriebssystem und basiert auf einer stark abgespeckten Version von Raspbian nebst einigen Extras. Erdacht und entwickelt wurde es vom Pi-Enthusiasten petRockBlock.

Um RetroPie auszuprobieren, besuchen Sie zunächst [goo.gl/34tFal](http://goo.gl/34tFal) und laden Sie die neueste gezippte Version der Image-Datei von RetroPie herunter. Entpacken Sie die Datei und kopieren Sie sie auf eine SD-Karte mit mindestens 4 GB Speicherplatz.

Schließen Sie dann ein TV-Gerät oder einen Monitor, eine Tastatur, eine Maus und ein USB-Gamepad an, schieben Sie die SD-Karte mit dem Image ein, und fahren Sie den Raspberry Pi hoch.



› Der Konfigurationsbildschirm der Benutzeroberfläche EmulationStation.



› Mal wieder einen Klassiker zocken? Das kann zur Abwechslung großen Spaß machen.

Beim ersten Booten müssen Sie die Settings für den Controller einstellen, mit dem Sie durch die grafische Benutzeroberfläche EmulationStation navigieren. Das ist im Wesentlichen der Menübildschirm; hier wählen Sie die verschiedenen Emulatoren und Spiele aus. Wie bei anderen Betriebssystemen auch kann das erste Hochfahren von RetroPie auf dem Raspberry Pi recht lange dauern – danach geht es aber fixer.

Nach dem Setup gelangen Sie zurück ins Terminal. Geben Sie dort einfach **emulationstation** in die Kommandozeile ein, um die Benutzeroberfläche von RetroPie zu laden.

## Spiele-ROMs

Die Spiele, die standardmäßig mitgeliefert werden, sind relativ eingeschränkte Shareware-Versionen von Duke Nukem 3D und Doom sowie ein eher unbekanntes

Game namens Apple 2. Sie werden also vermutlich Bedarf an weiteren Spieliteln haben. Diese sind in Form von sogenannten ROMs zu bekommen, dabei handelt es sich um Speicherabbilder originaler Spielmodule für Konsolen. Das Herunterladen und der Gebrauch dieser Abbilder ist rechtlich nicht immer unbedenklich, aber es kommt auf den konkreten Fall an. Ein juristischer Exkurs würde hier jedoch den Rahmen sprengen. Beachten Sie bitte den Hinweis am Ende dieses Artikels.

Auf der petRockBlock-Seite [goo.gl/QFMWCw](http://goo.gl/QFMWCw) finden Sie ferner eine Schritt-für-Schritt-Bauanleitung für einen GPIO-Adapter, der es Ihnen erlaubt, Game-Controller an den Raspberry Pi anzuschließen.

**Hinweis: Bitte beachten Sie bei der etwaigen Verwendung von Spiele-ROMs die geltende Rechtslage (beispielsweise § 53 Urheberrechtsgesetz in Deutschland)!**

## Von RetroPie unterstützte Systeme und Emulatoren

Nicht übel! Laut der Blog-Seite von petRockBlock werden folgende Emulatoren und Systeme ab Version 1.9 von RetroPie unterstützt:

- › Amiga (UAE4All)
- › Apple II (Basilisk II)
- › Arcade (PiFBA, Mame4All-RPi)
- › Atari 800
- › Atari 2600 (RetroArch)
- › Atari ST/STE/TT/Falcon
- › C64 (VICE)

- › CaveStory (NXEngine)
- › Doom (RetroArch)
- › Duke Nukem 3D
- › Final Burn Alpha (RetroArch)
- › Game Boy Advance (gpSP)
- › Game Boy Color (RetroArch)
- › Game Gear (Osmose)
- › Intellivision (RetroArch)
- › MAME (RetroArch)
- › MAME (AdvMAME)
- › NeoGeo (GnGeo)

- › NeoGeo (Genesis-GX, RetroArch)
- › Sega Master System (Osmose)
- › Sega Megadrive/Genesis (DGEN, Picodrive)
- › Nintendo Entertainment System NES (RetroArch)
- › N64 (Mupen64Plus-RPi)
- › PC Engine/Turbo Grafx 16 (RetroArch)
- › PlayStation 1 (RetroArch)
- › ScummVM

- › SNES Super Nintendo Entertainment System (RetroArch, PiSNES, SNES-Rpi)
- › Sinclair ZX Spectrum (Fuse)
- › PC/x86 (rpiX86)
- › Z Machine Emulator (Frotz)

Viel Spaß beim Testen, Ausprobieren und Spielen! Scheuen Sie sich nicht, Kontakt mit der RetroPie-Community aufzunehmen.

# Distributionen

Erstellen Sie sich Ihre eigene Raspberry-Pi-Distribution.

Die „New Out Of the Box Software“ (NOOBS) hat die Installation einer Linux-Distribution auf dem Raspberry Pi standardisiert und vereinfacht. Sie brauchen lediglich das Archiv mit dem NOOBS-Installer herunterzuladen und zu extrahieren, seine Inhalte auf eine SD-Karte zu überspielen und den Raspberry Pi damit zu booten.

Zwar ist der primäre Zweck der NOOBS, die Installation eines Betriebssystems auf dem Pi zu erleichtern, doch sie kann auch verwendet werden, um eine individuelle Distribution zusammenzustellen. Das eröffnet eine Reihe interessanter Möglichkeiten.

Benutzen Sie den Pi beispielsweise in einem Bildungskontext, so können Sie die Standard-Raspbian-Installation zur Grundlage nehmen, diese mit Ihren bevorzugten Lernanwendungen und einer Auswahl aus dem großen Fundus

der frei erhältlichen Ausbildungsmaterialien bestücken und schließlich für die Verbindung mit Ihrem WLAN-Netzwerk konfigurieren. Wenn Sie fertig sind, können Sie Ihre maßgeschneiderte Distribution auf einen Satz Raspberry Pis in Ihrem Klassenzimmer aufspielen.

Wenn Sie den Pi nur zu Hause benutzen, können Sie dieses Feature verwenden, um ein Backup Ihrer modifizierten Distribution zu machen. Auf diese Weise müssen Sie keine Zeit mit dem Neuaufsetzen Ihres Raspberry Pi verschwenden, falls Ihre SD-Karte kaputtgeht. Sie installieren einfach Ihr gesichertes Image auf einer neuen SD-Karte und schon läuft der Laden wieder.

## Raspbian individualisieren

Holen Sie sich den NOOBS-Installer von [www.raspberrypi.org/downloads](http://www.raspberrypi.org/downloads) und folgen Sie der Installationsanleitung für diejenige unterstützte Distribution, die Sie anpassen möchten. Wir empfehlen – im Einklang mit den Vorgaben des NOOBS-Installers – die Distribution Raspbian.

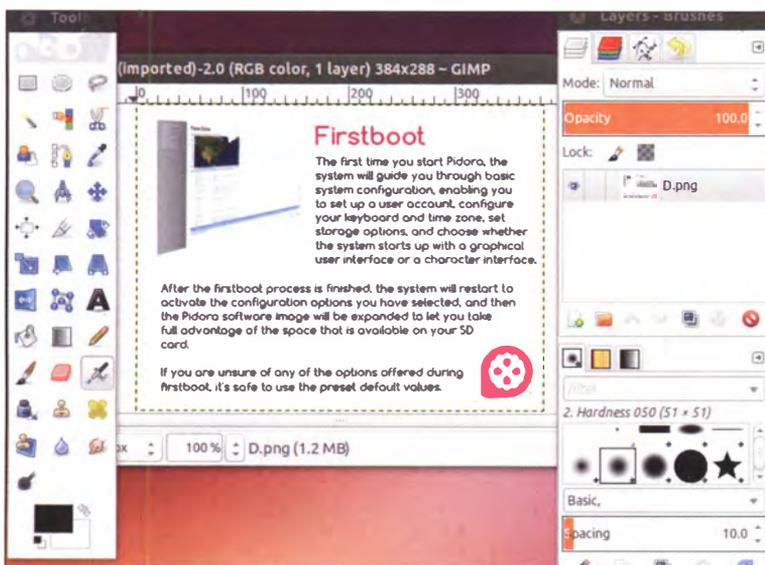
Nachdem Sie Raspbian installiert haben, booten Sie die Distribution und individualisieren sie. In der Anleitung finden Sie Anregungen hierzu.

Wenn Sie mit den Änderungen fertig sind, ist es an der Zeit, das Dateisystem in ein Archiv zu packen. Starten Sie dazu ein Terminal in Raspbian und wechseln Sie mit dem Kommando `cd /` ins Stammverzeichnis. Dort geben Sie folgende Zeile ein:

```
sudo tar -cvpf root.tar / * --exclude=proc/* --exclude=sys/*
--exclude=dev/pts/*
```

Nun können Sie sich eine Tasse Kaffee holen, denn dieser Vorgang kann, je nach Umfang der von Ihnen vorgenommenen Änderungen, bis zu einer halben Stunde dauern.

Wenn er fertig ist, haben Sie eine Datei namens **root.tar** im Stammverzeichnis. Packen Sie nun analog dazu die Bootdateien ein. Wechseln Sie zuerst mit `cd /boot` ins Bootverzeichnis und erzeugen Sie dann mit dem Kommando



## Ein Betriebssystem automatisch installieren

Wenn Sie Ihre Distribution auf mehreren Raspberry Pis installieren möchten, können Sie Zeit sparen, wenn Sie NOOBS zur automatischen Installation Ihres Betriebssystems einrichten. Stellen Sie zunächst sicher, dass das „os“-Verzeichnis nur den Ordner mit dem Betriebssystem, das Sie installieren möchten, enthält. Falls Ihre Distribution auf einem Betriebssystem mit verschiedenen Versionen basiert, entfernen Sie die Datei **flavours.json** aus dem Ordner mit Ihrer Distribution. Kehren Sie dann zurück zum NOOBS-Stammverzeichnis und öffnen Sie die Datei **recovery.cmdline** im Texteditor. Darin steht eine Reihe Argumente, die NOOBS beim Start einliest. Fügen Sie **silentinstall** dieser Liste hinzu und speichern Sie die Datei. Folgen Sie dann dem

üblichen Vorgehen, um die NOOBS-Dateien auf die SD-Karte zu überspielen. Wenn Sie den Pi nun mit dieser Karte booten, installiert er automatisch das Betriebssystem auf der SD-Karte und bootet es danach.

Sie können die Datei **recovery.cmdline** auch verwenden, um die Standardsprache und die Tastaturbelegung durch die Angabe eines Ländercodes zu ändern. Die Argumente **lang=de** und **lang=en** stellen die Standardsprache auf Deutsch beziehungsweise Englisch ein. Analog dazu funktionieren **keyboard=de** und **keyboard=en** für das Tastaturlayout. Beachten Sie, dass diese Standardeinstellungen trotzdem noch über das grafische NOOBS-Menü geändert werden können.



Wenn Sie keine Ausgabe auf dem Bildschirm sehen, wechseln Sie den Modus durch Drücken der Taste [1], [2], [3] oder [4].

# à la carte

**tar -cvpf boot.tar** das Archiv. Danach befindet sich eine Datei namens **boot.tar** im Bootverzeichnis. NOOBS benötigt komprimierte Versionen dieser Dateien, aber der kleine Raspberry Pi besitzt keine ausreichenden Ressourcen zur Kompression, weshalb Sie die Dateien auf einen gewöhnlichen Desktop-PC verschieben müssen. Stecken Sie dazu einen USB-Stick an den Raspberry Pi – dieser wird automatisch eingehängt.

Um den Pfad des USB-Sticks zu ermitteln, geben Sie nach dem Anstecken den **mount**-Befehl in einem Terminal ein. Dieser listet alle Partitionen und die Pfade ihrer Einhängpunkte auf – der USB-Stick ist der letzte Eintrag.

Ist das Laufwerk zum Beispiel unter **/media/usb** eingehängt, dann kopieren Sie die beiden Dateien mit den Kommandos **sudo cp /root.tar /media/usb** und **sudo cp /boot/boot.tar /media/usb**. Nach dem Kopieren können Sie den USB-Stick mit **sudo umount /media/usb** auswerfen.

Fahren Sie nun den Raspberry Pi herunter, entfernen Sie die SD-Karte und das USB-Laufwerk und wechseln Sie an eine normale Linux-Maschine. Stecken Sie den USB-Stick an und komprimieren Sie die Tarballs mit **xz -9 -e boot.tar** und **xz -9 -e root.tar**. Dadurch werden die Dateien durch ihre komprimierten Versionen, **boot.tar.xz** und **root.tar.xz**, ersetzt.

## Erstellen Sie Ihre Distribution

Formatieren Sie nun die SD-Karte und entpacken Sie eine neue Kopie von NOOBS darauf. Benutzen Sie den Dateimanager, um in den entpackten Dateien in das Verzeichnis

„os“ zu navigieren. Dieser Ordner enthält wiederum eine Reihe von Unterordnern, von denen jeder die Dateien für ein der unterstützten Betriebssysteme – unter anderem Arch, Pidora und Raspbian – enthält. Da unsere individuelle Distribution Raspbian-basiert ist, steht es Ihnen frei, alle anderen Ordner im „os“-Verzeichnis zu löschen. Geben Sie nun dem „Raspbian“-Ordner den Namen Ihrer eigenen Distribution, zum Beispiel „ReRaspbian“.

Gehen Sie in das Verzeichnis und öffnen Sie die Datei **os.json** mit einem Texteditor. Ersetzen Sie dort den Text der Felder „name“ und „description“ entsprechend Ihren eigenen Vorstellungen. Löschen Sie außerdem die Datei **flavours.json**.

Wenn Sie möchten, können Sie auch das Erscheinungsbild Ihrer Distribution anpassen. Sie können das zu Ihrer Distribution gehörige Icon und die Slides, die während der Installation erscheinen, ändern. Sehen Sie sich dazu den Kasten „Eigene Grafiken erstellen“ an.

Entfernen Sie schließlich die bestehenden Dateien **root.tar.xz** und **boot.tar.xz** aus diesem Ordner und ersetzen Sie sie mit denen, die Sie eben erzeugt haben. Falls Sie die Dateinamen geändert haben, denken Sie daran, die Änderungen auch in den „label“-Feldern in der Datei **partitions.json** durchzuführen.

Fertig! Booten Sie nun den Raspberry Pi von dieser Karte. Ihre Distribution erscheint jetzt im NOOBS-Menü. Nach der Installation besitzt die Distribution all Ihre persönlichen Änderungen. Folgen Sie den Anweisungen unter „Ein Betriebssystem automatisch installieren“, um das Setup weiter zu beschleunigen, indem Ihre Distribution automatisch installiert wird, ohne Benutzereingaben zu benötigen.

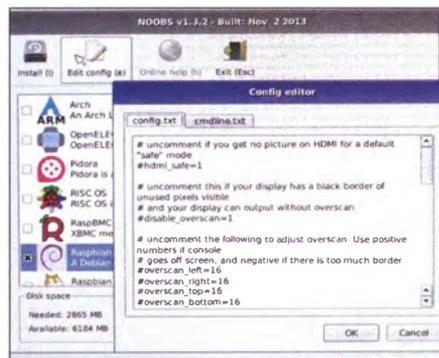
## Die NOOBS-Benutzeroberfläche

Viele Aspekte Ihrer Installation können einfach angepasst werden.



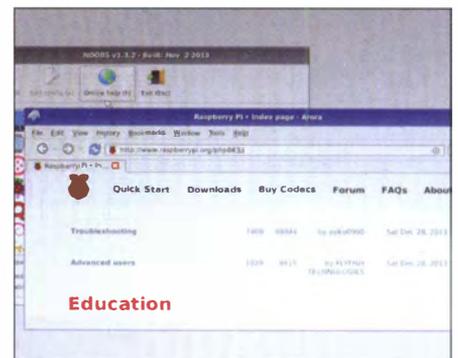
### 1 Ändern von Sprache und Tastaturbelegung

NOOBS unterstützt diverse Sprachen und Tastaturlayouts. Standardmäßig ist beides auf britisches Englisch eingestellt, dies kann aber leicht geändert werden, indem Sie Alternativen aus den beiden Pull-down-Listen am unteren Bildschirmrand auswählen.



### 2 Konfiguration editieren

Sind Sie ein erfahrener Raspberry-Pi-Benutzer und wissen, was Sie tun, dann können Sie die Dateien **cmdline.txt** und **config.txt** bearbeiten, um die Einstellungen des Raspberry Pi zu optimieren. In diesen Dateien können Sie einige Parameter der Systemkonfiguration setzen, die normalerweise im BIOS eingestellt werden.

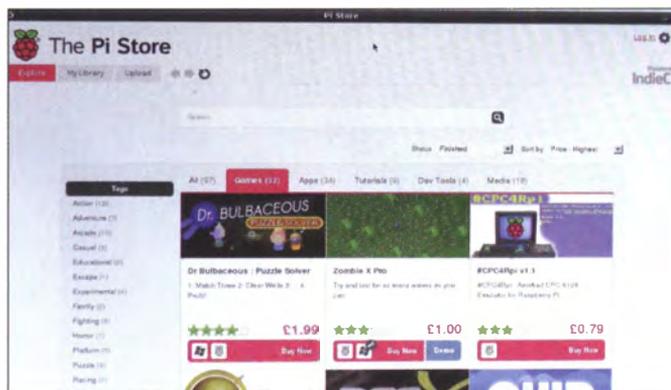
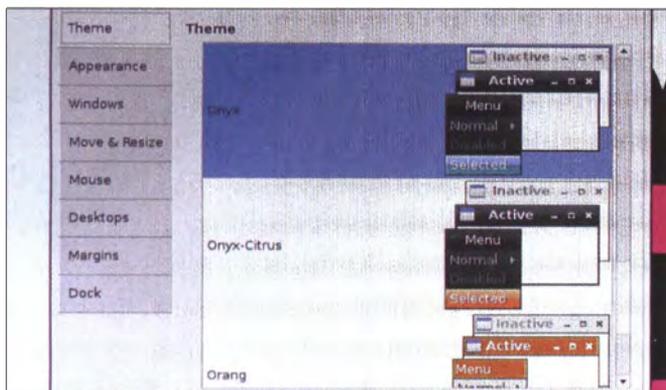


### 3 Online-Hilfe

Wenn Sie einmal nicht weiterwissen, können Sie den eingebauten Webbrowser aufrufen, der bereits auf die Adresse des Raspberry-Pi-Forums voreingestellt ist. Für diese Funktion muss Ihr Raspberry Pi natürlich über den Ethernet-Port mit dem Internet verbunden sein.

## Individualisieren Sie Ihre Distribution

Machen Sie die Distribution von der Stange zu einer maßgeschneiderten.

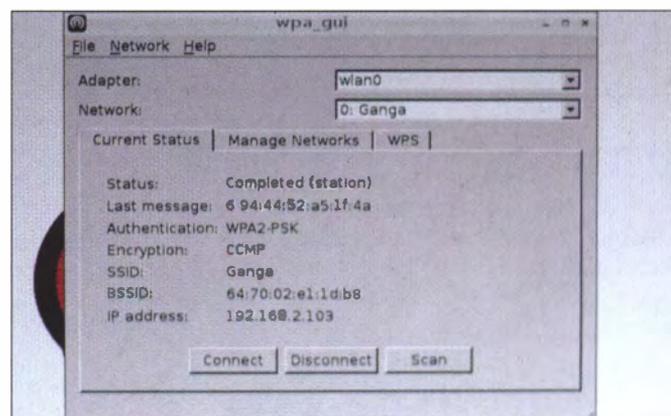
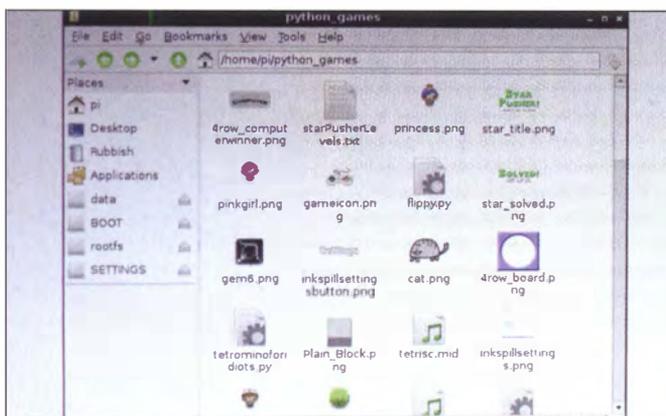


### 1 Erscheinungsbild ändern

Eins der Dinge, die Sie vermutlich ändern möchten, ist das Standard-Hintergrundbild. Dazu klicken Sie mit der rechten Maustaste auf den Desktop und wählen „Desktop-Einstellungen“. Sie können auch das Theme ändern, indem Sie **obconf** im Terminal ausführen oder mit dem Kommando **sudo apt-get install openbox-themes** weitere installieren.

### 2 Anwendungen (de-)installieren

Sicherlich möchten Sie, je nachdem, was Sie mit Ihrer Distribution vorhaben, die Liste der installierten Anwendungen anpassen. Falls Ihre Distribution auf Raspbian basiert, können Sie ein grafisches Front-End für die Paketverwaltung installieren. Holen Sie sich dazu per **sudo apt-get install synaptic** das Benutzer-Interface Synaptic.



### 3 Dokumente übertragen

Abhängig vom Einsatzzweck Ihrer Distribution möchten Sie eventuell Dateien oder Dokumente hinzufügen. Es gibt zwei Arten, Dateien zu speichern: Sie können über NOOBS eine 512-MB-Partition anlegen, die Sie als Ablage verwenden; oder Sie benutzen den Befehl **raspi-config**, um die root-Partition zu vergrößern, sodass diese den Platz auf der SD-Karte voll ausnutzt.

### 4 Netzwerkeinstellungen

Sie können Ihre Distribution auch so aufsetzen, dass sie ohne weiteres Zutun mit Ihrem Netzwerk arbeitet. Sie können beispielsweise die WLAN-Karte zur Verbindung mit Ihrem Access Point und zum Zugriff auf Netzwerkdienste wie einen Adressbuch-Server konfigurieren oder die Standard-Startseite Ihres Browsers auf die Landing Page Ihres Intranets einstellen.

## Eigene Grafiken erstellen

Eine wirklich individuelle Distribution braucht auch eigene Grafiken. Mit NOOBS können Sie ein eigenes Icon festlegen, das zusammen mit dem Namen und der Beschreibung Ihrer Distribution im NOOBS-Installationsbildschirm angezeigt wird; ebenso die Info-Slides, die während der Installation angezeigt werden.

Zur Erstellung Ihres Icons können Sie jedes Bildbearbeitungsprogramm verwenden. Machen Sie Ihr Icon 40 x 40 Pixel groß. Gehen Sie dann in das

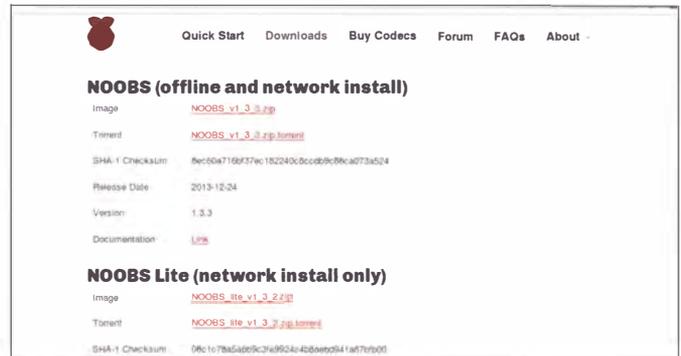
Verzeichnis Ihrer Distribution, unterhalb des Ordners „os“. Hier finden Sie eine einzelne Grafikdatei mit der Endung .png, etwa **Raspbian.png**. Ersetzen Sie diese durch Ihre eigene und sorgen Sie dafür, dass sie den Namen Ihrer Distribution trägt – zum Beispiel **ReRaspbian.png**. Wechseln Sie nun in das Verzeichnis **slides\_vga**, in dem die Info-Slides liegen. Sie können jede Folie durch eine eigene mit den Maßen 387 x 290 Pixel ersetzen, wobei Sie die erste **A.png**, die zweite **B.png** und so weiter nennen müssen.



Die Bildbearbeitungssoftware GIMP ist ein hervorragendes Werkzeug, um ein eigenes Erscheinungsbild zu kreieren. Sie können es auch benutzen, um die bestehenden Slides zu ändern.

# Raspberry Pi mit NOOBS aufsetzen

Sie können ganz einfach mehrere Betriebssysteme installieren.



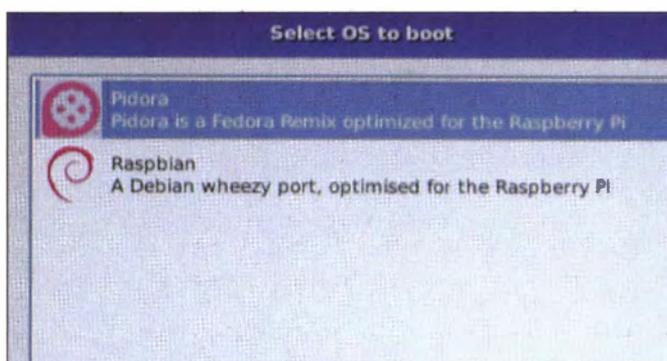
## 1 SD-Karte formatieren

Schnappen Sie sich eine mindestens 4 GB große SD-Karte und überprüfen Sie, ob sie sauber formatiert ist. Windows- und Mac-Nutzer können das Formatierungswerkzeug der SD Card Association unter [www.sdcard.org/downloads/formatter\\_4/](http://www.sdcard.org/downloads/formatter_4/) herunterladen. Unter Linux verwenden Sie GParted oder das Standard-Partitionierungswerkzeug Ihrer Distribution.



## 3 Dateien entpacken und kopieren

Unabhängig davon, welche Version Sie gewählt haben, entpacken Sie nun die komprimierte .zip-Datei und kopieren Sie die Inhalte (anstelle des root-Ordners) auf die SD-Karte. Nun können Sie die SD-Karte aus Ihrem Computer auswerfen, in den SD-Kartenslot des Raspberry Pi einlegen und diesen einschalten.

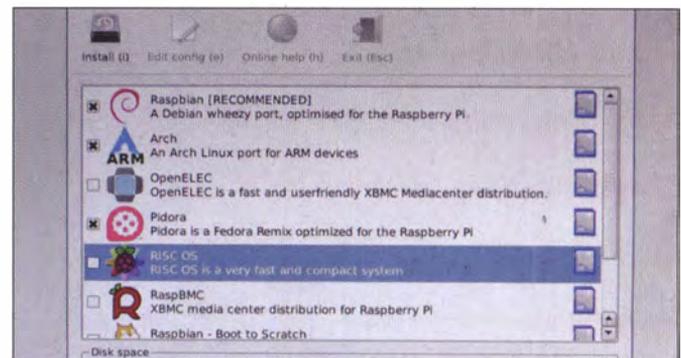


## 5 Betriebssystem zum Booten auswählen

Klicken Sie auf den „Installieren“-Knopf, um die Installation zu starten. Lehnen Sie sich zurück, während NOOBS die Betriebssysteme nacheinander installiert. Wenn sie damit fertig ist, wird Ihr Raspberry Pi neu gestartet. Falls Sie mehrere Betriebssysteme installiert haben, zeigt Ihnen NOOBS eine Liste an, aus der Sie eins zum Booten auswählen können.

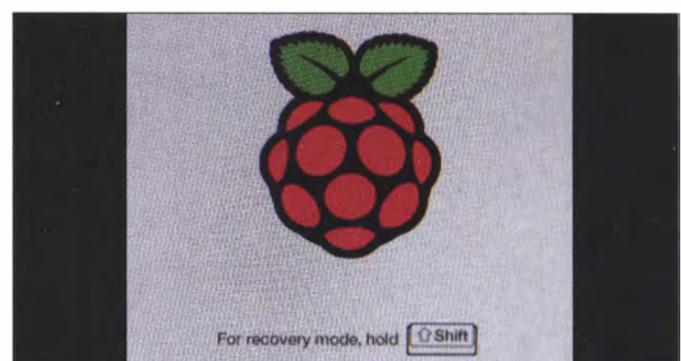
## 2 NOOBS herunterladen

Vom NOOBS-Installer gibt es zwei Versionen: Offline und Network Only. Die Offline-Version ist größer und enthält den NOOBS-Installer sowie sämtliche verfügbaren Installations-Images. Die Network-Only-Version lädt Distributionen während der Installation herunter.



## 4 Zu installierende Distribution wählen

Beim ersten Start durchläuft der NOOBS-Installer seinen Initialisierungsprozess, der auch die Partitionierung der SD-Karte und Größenanpassung der Partitionen enthält. Danach wird Ihnen die NOOBS-Installationsoberfläche angezeigt. Hier können Sie, je nach Größe Ihrer SD-Karte, auch mehrere Betriebssysteme auswählen.

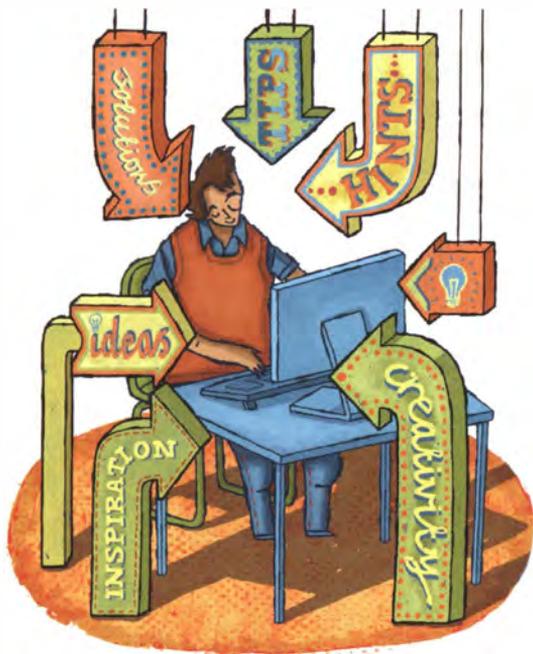


## 6 Wiederherstellungsmodus

Falls bei der Installation etwas schiefgelaufen ist und Sie den Installationsassistenten erneut aufrufen möchten, können Sie NOOBS im Wiederherstellungsmodus booten: Starten Sie den Pi bei gedrückter Umschalttaste. Aus der Recovery-Konsole können Sie eine neue Kopie eines bestehenden Betriebssystems oder ein ganz neues Betriebssystem installieren.

# Desktop: Eigene

Ja, Sie können Ihre eigene, auf Sie angepasste, genügsame Desktop-Umgebung bauen, ohne dafür eine einzige Zeile Code schreiben zu müssen. Wir verraten Ihnen, wie.



**T**itelleisten und Schnellzugriffe, Panels und Benachrichtigungsbereiche sowie nicht zuletzt Desktop-Hin- tergründe sind allesamt Dinge, die wir von einer

modernen Desktop-Umgebung erwarten. Normalerweise werden diese Funktionen von unterschiedlichen Einzelpro- grammen zu Verfügung gestellt. Wenn Sie also Xfce nutzen und **ps ax** in ein Terminal eingeben, so wird **xfwm** als Fen- stermanager genutzt, **xfce4-panel** für Panels – und so weiter. Wenn Sie diese Programme beenden, werden Teile Ihrer Desktop-Umgebung vom Bildschirm verschwinden. KDE und Gnome funktionieren ganz ähnlich.

Schön und gut, aber wir sind schließlich Linux-Geeks. Die Grundkonfiguration macht einen ordentlichen Job, aber wir nutzen dieses Betriebssystem, um es unseren Bedürfnissen anzupassen, um jene Teile herauszureißen, welche uns keinen Spaß machen, und um die inneren Funktionsabläufe des Systems zu verstehen. Sie können jedes dieser Programme für sich genommen ersetzen, Sie können aber auch Ihre eige- ne Kombination aus Fenstermanager, Panels, Desktop-Swit- cher und so weiter anlegen.

Es gibt zahlreiche leichtgewichtige Fenstermanager, die nur das unbedingt nötige Minimum an Funktionalität besit- zen, weshalb andere Entwickler die restlichen Teile geschrie- ben haben, die man für eine voll funktionsfähige Desktop- Umgebung braucht. Wir sehen uns die möglichen Optionen an und zeigen Ihnen, wie Sie sie zu einem personalisierten Desktop zusammensetzen. Falls Sie KDE, Gnome oder Xfce nutzen, werden Sie feststellen, dass das Resultat ein wesent- lich flotterer und speichersparenderer Rechner sein wird.

## Auswahl des Fenstermanagers

» Wenn Sie einen Fenstermanager über die Paket- verwaltung hin- zufügen, sollte auch eine ent- sprechende Ses- sion erzeugt wer- den. Wir zeigen Ihnen aber auch, wie das manuell funktioniert.

Der Fenstermanager ist die weitaus wichtigste Komponente einer Desktop-Umgebung. Ein Fenster ist im Normalfall mit einer Titelleiste ausgestattet, sodass Sie es auf dem Bild- schirm umherschoben können, sowie mit Buttons zum Ma- ximieren, Schließen und so weiter. Der Fenstermanager er- möglicht auch das Vergrößern und Verkleinern eines Fensters an dessen Rändern. Ohne Fenstermanager würde eine Anwendung zwar auch funktionieren, doch Sie wären nicht in der Lage, sie zu verschieben oder zu skalieren. Es gibt einige neuartige Fenstermanager, die anders an die Sache herangehen, indem Sie beispielsweise die Program- me in einem Gitter anordnen und somit keine Titelleisten nötig sind.

In diesem Tutorial beschreiben wir die besten Vertreter und erklären, wie Sie sie benutzen. Wenn Sie die Fenstermanager über den Paketmanager installieren, sollte es möglich sein, dass Sie nach dem Aus- und Wiedereinloggen zwischen den Fenstermanagern wählen können. Wenn Sie keinen Eintrag für Ihren Fen- stermanager sehen, probieren Sie eine

Fallback- oder Terminal-Option im Login-Bildschirm aus: Eini- ge Distributionen bieten diese Möglichkeit an. Über das Ter- minal können Sie den Fenstermanager mithilfe der unten ge- nannten Befehle starten. Falls Sie das neue Programm gar nicht starten können, brauchen Sie sich keine Sorgen zu ma- chen. Wir zeigen Ihnen später, wie Sie sich selbst eine neue Login-Session erstellen können.

» **Name:** JWM

» **Website:** <http://joewing.net/projects/jwm/>

» **Befehl zum Starten:** **jwm**

JWM ist sehr schlank, in C geschrieben und benötigt nur die Bibliotheken des X-Displayserver als Abhängigkeiten. Deshalb arbeitet er auch pfeilschnell. JWM hat ein brauchba- res Standardpanel mit allerdings wenigen Features wie Pro- gramme-Menü, Desktop-Umschalter und Taskleiste. Stan- dardmäßig enthält das Programme-Menü (das auch per Rechtsklick auf dem Desktop erreichbar ist) nur Einträge, die Ihnen das Starten eines Terminals sowie das Abmelden oder Sperren des Benutzerkontos erlauben. Sie editieren das Menü in der Datei **/etc/jwm/system.jwmrc**. Die XML-ba- sierte Konfigurationsdatei ist leicht zu verstehen, über sie können Sie die Schriftart und Tastenkombinationen des



# Schreibtische

Fenstermanagers anpassen.

» **Name:** Openbox

» **Website:** [www.openbox.org](http://www.openbox.org)

» **Befehl zum Starten:** `openbox`

Openbox gehört – wie auch Fluxbox – zur „\*box“-Familie der Fenstermanager, die sich auf den Stammvater Blackbox gründet. Openbox ist vermutlich der beliebteste Leichtgewicht-Fenstermanager und wird beim LXDE-Desktop verwendet. Die Standardkonfiguration von Openbox ist so knapp wie möglich gehalten: Sobald Sie ihn starten, sehen Sie erstmal eines: nichts. Bis auf den Mauszeiger, der in der Mitte des Bildschirms verweilt. Mit einem Rechtsklick öffnet sich ein kleines Kontextmenü, über das Sie ein Terminal, einen Browser oder das Konfigurationswerkzeug aufrufen können. Genau dieses Tool ist der Grund dafür, dass Openbox unsere erste Wahl für Eigenbau-Desktops ist: Hier können wir nämlich fast alles konfigurieren, ohne in Textdateien herumstochern zu müssen. Zudem ist Openbox sehr standardkonform und arbeitet daher wunderbar mit anderen Werkzeugen zusammen. Die Tastenkombinationen sind wie gewohnt: Mit [Alt]+[Tab] wechseln Sie durch die Fenster, [Alt]+[F4] schließt Programme, und [Strg]+[Alt]+[Links/Rechts] wechselt die Desktops.

» **Name:** Ratpoison

» **Website:** <http://ratpoison.wxcvbn.org>

» **Befehl zum Starten:** `ratpoison`

Hier haben wir eine sehr ungewöhnliche Art von Fenstermanager: Ratpoison wurde um ein „Tiling“ genanntes Konzept herum konstruiert. Anders als beim sonst üblichen Konzept verschiebbarer, sich überlappenden Fenster belegen bei Ratpoison die Fenster eine bestimmte Fläche auf dem Bildschirm. Es dauert eine Weile, sich daran zu gewöhnen, aber sobald Sie sich die Tastenkombinationen für die Arbeit mit den Fenstern eingepägt haben, sind Sie weniger von der Maus abhängig. Deshalb auch der Name „Rattengift“. Beim ersten Start ist Ratpoison fast komplett nackt und Sie werden feststellen, dass die Maus keine Funktion erfüllt. Drücken Sie [Strg]+[t], gefolgt von [!], erscheint ein kleiner Kasten in der rechten oberen Ecke, in den Sie ein Kommando eingeben können (zum Beispiel `xterm`). Wenn Sie ein Programm starten, wird es standardmäßig im Vollbildmodus ausgeführt. Haben Sie mehrere Fenster geöffnet, schalten Sie mit [Strg]+[t], gefolgt von [n] und [p], zwischen ihnen um. Geben

## Die gleiche Sprache sprechen

Es ist kein Zufall, dass die genannten Tools so wunderbar zusammenarbeiten. Kritiker werfen Linux vor, ein Mischmasch aus alten und neuen Technologien zu sein, der kaum etwas zusammenhält, was allerdings nicht gerecht ist. Richtig, es gibt immer noch eine Menge alten Code zu finden, aber die Programme arbeiten dank Standards trotzdem zusammen. EWMH, die „Extended Window Manager Hints“-Spezifikation, legt die Interaktionsmöglichkeiten zwischen den Fenstermanagern, den Anwendungen und den Werkzeugen fest. Sie ermöglicht also den Komponenten, Informationen auszutauschen, um dem Nutzer eine Erfahrung aus einem Guss zu bieten. Ein

gutes Beispiel sind Desktop-Umschalter: Diese bieten oft eine Miniaturansicht der Desktops und der gerade laufenden Programme. Normalerweise möchten Sie in diesen Voransichten nur die Anwendungsfenster sehen, nicht auch noch das ganze Desktop-Drumherum. Dank EWMH kann ein Panel oder ein Fenstermanager dem Desktop-Umschalter sagen, dass ein bestimmtes Fenster dort nicht sichtbar sein soll. Die vollständige Spezifikation finden Sie unter <http://standards.freedesktop.org/wm-spec/wm-spec-latest.html>. Sie ist lang, aber zwingend notwendig, sollten Sie den Ehrgeiz entwickeln, einen eigenen Fenstermanager zu schreiben.

Sie [Strg]+[t] [w] ein, um eine nummerierte Liste aller Fenster zu erhalten, [Strg]+[t], gefolgt von einer Ziffer, um auf das entsprechende Fenster umzuschalten. Der wirkliche Spaß beginnt mit dem Tiling-Modus: Drücken Sie [Strg]+[t] [s] mit einem neu geöffneten Fenster, um den Bildschirm horizontal in zwei Flächen zu teilen, Rahmen genannt. Das aktuelle Fenster wird im oberen Rahmen angezeigt, das vorherige darunter. Das Drücken von [Strg]+[t] [S] (Großbuchstabe) teilt den Bildschirm vertikal. Das ist besonders toll, wenn Sie einen großen Monitor mit einer hohen Auflösung benutzen, auf dem Sie beispielsweise einen großen Rahmen für den Webbrowser anlegen können, während Sie daneben mehrere kleine Rahmen für diverse Terminal-Sessions nutzen. Die Tastenkombination [Strg]+[t], gefolgt von einer der Cursor-Tasten, hilft Ihnen beim Umschalten zwischen den Rahmen. Sie können einen Rahmen auch wieder im Vollbild anzeigen, indem Sie [Strg]+[t] [Q] tippen. Eine Liste aller Tastenkombinationen erhalten Sie über die Eingabe von [Strg]+[t] [?]. Hier hilft auch das Ratpoison-Wiki unter <http://tinyurl.com/obcarv5>.

Auch die folgenden Fenstermanager sind einen Blick wert:

» Sawfish (<http://sawfish.wikia.com/wiki/>)

» WindowLab (<http://nickgravgaard.com/windowlab/>)

» Pekwm ([www.pekwm.org](http://www.pekwm.org))

## Auswahl des Panels

Den Job der Fensterverwaltung haben wir nun also vergeben. Nun möchten wir ein Panel zum Starten von Programmen, zum Umschalten zwischen ihnen und den Desktop-Flächen, zum Anzeigen einer Uhr und so weiter. JWM beinhaltet ein einfaches Panel, aber wir hätten gerne ein paar zusätzliche Funktionen.

» **Name:** Docky

» **Website:** [www.go-docky.com](http://www.go-docky.com)

» **Befehl zum Starten:** `docky`

Docky ist ziemlich cool. Es ist ein umfangreiches Panel und hat viele Abhängigkeiten – falls Sie also planen, einen Desktop für Maschinen mit wenig Speicher zu entwerfen, sollten Sie lieber zu einer anderen Option greifen. Docky sollten Sie in den Repositories der meisten Distributionen finden. Zur Installation reicht ein `sudo apt-get install docky`, sofern Sie mit einer Ubuntu-basierten Distribution arbeiten. Wenn Sie Docky starten, sehen Sie ein kleines schwarzes Panel am unteren Bildschirmrand. Es gibt ein einziges kleines Anker-Icon, »

» welches den Einstellungsdialog öffnet. Hier aktivieren Sie den Panel-Modus, welcher Docky auf die gesamte Bildschirmbreite ausdehnt. Wechseln Sie zum Docklets-Tab des Einstellungsdialogs, um dem Panel weitere statische Items hinzuzufügen wie etwa eine Uhr, einen Umschalter für den Desktop oder eine Batterieanzeige. Docky bringt kein Hauptfenster mit, was eine gewisse Einschränkung darstellt. Starten Sie jedoch Programme über Ihren Fenstermanager, tauchen diese auf dem Panel auf. Rechtsklicken Sie die Icons und wählen Sie „Pin to dock“, um sie im Panel anzuheften. Auf diese Weise füllen Sie das Panel mit den meistgenutzten Programmen, weshalb das Fehlen eines Hauptfensters dann doch kein so großer Nachteil ist.

» **Name:** Cairo-Dock

» **Website:** <http://glx-dock.org>

» **Befehl zum Starten:** `cairo-dock`

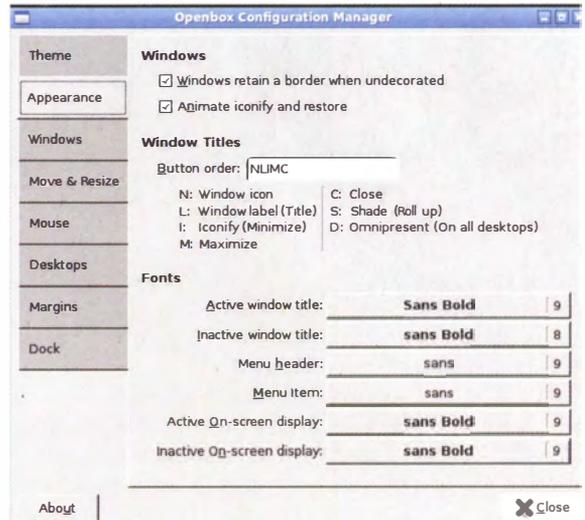
Wenn Sie auf ein schönes Design Wert legen, ist Cairo-Dock vielleicht etwas für Sie. Sein Aussehen ähnelt stark dem von OS X, mit edlen Reflexions-Effekten und feinen Linien, die die verschiedenen Flächen voneinander trennen. Icons werden größer, wenn Sie den Mauszeiger darüber bewegen. Machen Sie dasselbe bei einem Ordner, zeigt dieser eine Voransicht seines Inhalts an. Das Programm ist wunderschön designt, nimmt aber auch Ihre Grafikkarte dementsprechend in Beschlag. Im Gegensatz zu Docky bringt Cairo-Dock ein Hauptfenster mit, basierend auf den Systemeinstellungen. Mit anderen Worten: Es zeigt jene Icons, die Sie auch unter anderen Fenstermanagern und Desktops sehen würden. Sie können jedes Bild rechtsklicken und konfigurieren. Das Panel selbst passen Sie unter Cairo-Dock > Configure an. Es ist wirklich sehr konfigurierbar, zudem gibt es viele aktivierbare Add-ons, anpassbare Tastenkombinationen sowie modifizierbare Größen- und Positionseinstellungen.

» **Name:** WBar

» **Website:** <http://code.google.com/p/wbar/>

» **Befehl zum Starten:** `wbar`

Hierbei handelt es sich um das schlankste der hier vorgestellten Panels. Deshalb startet es auch fast ohne



» **Openbox hat sein eigenes Konfigurationswerkzeug, was das Herumfummeln in Textdateien unnötig macht.**

Verzögerung. Es bietet einen durchsichtigen Hintergrund, animierte wachsende Icons, wenn Sie den Mauszeiger darüber bewegen, und nicht zuletzt ein grafisches Konfigurationswerkzeug (`wbar-config`; separat installieren). Standardmäßig wird WBar vertikal angezeigt, Sie können dies jedoch ändern, wenn Sie das oberste Icon anklicken. Über den Icons-Reiter fügen Sie weitere Start-Icons hinzu. Wenn Sie sich vom Panel wünschen, sich mehr wie eine Taskleiste zu verhalten, finden Sie eine Option dazu im Preferences-Tab. In der Effects-Registerkarte finden Sie die Einstellungen für die Zoomstufe, die Transparenz und die Abstände zwischen den Icons.

Weitere Panels (teils jedoch etwas veraltet):

» **tint2** (<http://code.google.com/p/tint2/>)

» **PerlPanel** (<http://savannah.nongnu.org/projects/perlpanel>)

» **Avant Window Navigator** (<https://launchpad.net/awn>)

## Hinzufügen von Extras

Die meisten spärlich ausgestatteten Fenstermanager erlauben es nicht, selber ein Hintergrundbild auszuwählen. Das bedeutet, dass Sie dafür ein separates Werkzeug benötigen. `hsetroot` ist ein kleines Programm, das als eigenständiges Paket auf fast jeder Distribution zur Verfügung steht oder manchmal als Teil der HackedBox (ein anderer schlanker Fenstermanager). Sobald Sie `hsetroot` installiert haben, ist es ganz einfach wie folgt zu verwenden:

```
hsetroot -full /path/to/image.jpg
```

Dieser Befehl bringt das komplette Bild auf den Bildschirm, ohne irgendwelche Anpassungen. Andere Optionen sind **-center**, **-tile** und **-fill** (um es zu strecken). Sie können auch PNG-Bilder und andere Formate verwenden.

Falls Ihr Dock oder Panel keinen Umschalter für die

Desktops mitbringt, brauchen Sie einen eigenständigen. In dieser Kategorie gibt es leider keine so große Auswahl, wir empfehlen allerdings den `NetWMPager`, der aufgrund seiner minimalen Abhängigkeiten fast überall funktioniert. Sie finden ihn unter <http://sourceforge.net/projects/sf-xpaint/files/netwmpager/>. Laden Sie sich dort den aktuellsten Quellcode herunter und kompilieren Sie ihn wie folgt:

```
tar xfv netwmpager-2.04.tar.bz2
```

```
cd netwmpager-2.04
```

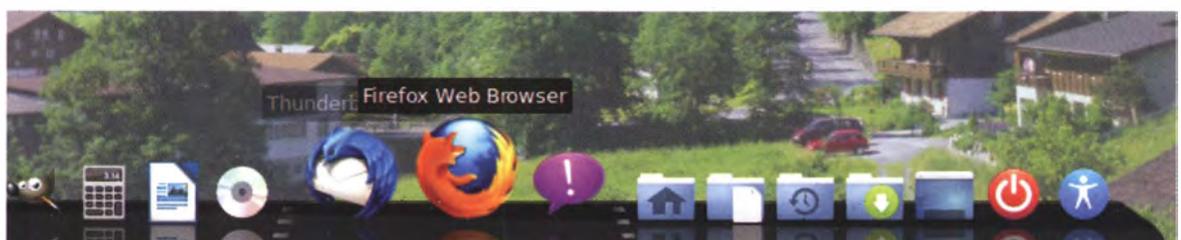
```
./configure
```

```
make
```

```
sudo make install
```

Wenn Sie beim **make**-Schritt der Fehlermeldung „undefined reference to symbol XGetWindowAttributes“ begegnen,

» **Sie möchten sich einen schicken Desktop wie bei OS X bauen? Greifen Sie zu Cairo-Dock!**



editieren Sie **config.mk** und ändern Sie die Zeile **XFT\_LIBS** wie folgt:

```
XFT_LIBS = -lXft -lX11
```

Speichern Sie die Datei und führen Sie die letzten beiden Schritte erneut aus. Nun können Sie das Programm mit **netwmpager** starten. Sie sehen, wie Ansichten der Desktops in der linken unteren Ecke des Bildschirms erscheinen. Standardmäßig sind diese sehr klein. Um dies zu ändern, müssen Sie die Datei **/usr/local/share/netwmpager/config-example** editieren, speziell die Zeilen **geometry** und **auto\_hide**.

Falls Sie einen Benachrichtigungsbereich (System-Tray)

benötigen, ist Stalonetray ein guter Tipp. Es gibt Stalonetray in den Paket-Archiven von Ubuntu wie auch als Quellcode unter <http://stalonetray.sf.net>. Falls Sie selbst kompilieren, führen Sie dabei die Schritte **./configure**, **make** und **sudo make install** aus und starten Sie Stalonetray danach mit:

```
stalonetray -geometry 5x1+50+20
```

Das erstellt einen Nachrichtenbereich, der fünf Spalten breit und eine Zeile hoch ist und dabei von der oberen linken Ecke des Bildschirms 50 Pixel nach rechts und 20 Pixel nach unten entfernt liegt. Diese Parameter können Sie natürlich nach Belieben anpassen.

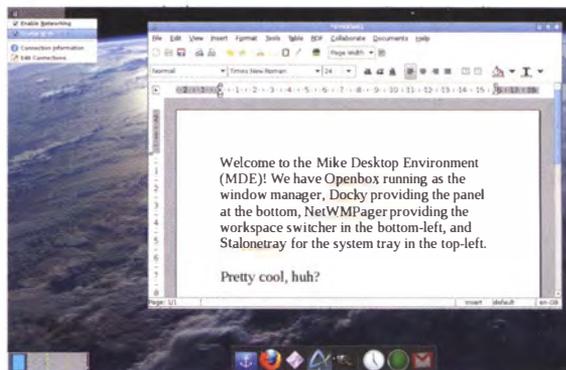
## Kombination der Komponenten

Nun sind wir bereit, alle Komponenten zusammenzufügen, um daraus eine Desktop-Umgebung zu basteln. Sie können jede x-beliebige Kombination der zuvor genannten Programme verwenden, wir nutzen hier Openbox, Docky, hsetroot, NetWMPager und Stalonetray. Der erste Schritt ist das Schreiben eines Scripts, das alle Programme startet. Für unseren Eigenbau-Desktop legen wir es in den Ordner **/usr/local/bin/mikedesk-start**:

```
#!/bin/bash
docky &
hsetroot -fill /usr/share/backgrounds/space-02.jpg &
netwmpager &
stalonetray -geometry 5x1+0+0 &
openbox
```

Das **&** hinter dem Befehl bedeutet, dass die Programme im Hintergrund laufen sollen, sodass sie alle zur selben Zeit laufen und aufgerufen werden können und nicht darauf warten, bis das vorhergehende Programm beendet ist. Es ist jedoch wichtig, dass der Fenstermanager in der letzten Zeile ohne ein **&** gestartet wird, weil die Sitzung ansonsten sofort beendet wird.

Speichern Sie die Datei und machen Sie sie ausführbar (zum Beispiel mit **sudo chmod +x mikedesk-start**). Der nächste Schritt ist das Erzeugen einer Login-Screen-Sitzung. Legen Sie dazu eine neue Datei à la **/usr/share/xsessions/**



› Hier sehen Sie das Ergebnis unserer Arbeit: Eine Desktop-Umgebung bestehend aus den von uns ausgesuchten Komponenten.

**mikedesk.session** mit dem folgenden Inhalt an:

```
[Desktop Entry]
Name=MikeDesk
Comment=Mike Desktop Environment
Type=XSession
Exec=/usr/local/bin/mikedesk-start
TryExec=mikedesk-start
```

Speichern Sie wieder, melden Sie sich mit Ihrer aktuellen Sitzung ab und es sollte eine neue starten, dieses Mal aber mit dem Login-Manager. Wählen Sie den Benutzernamen, loggen Sie sich ein und genießen Sie die Früchte ihrer Arbeit. Glückwunsch: Sie sind nun offiziell großartig.

## Hilfe! Meine Gtk- und Qt-Programme sehen blöd aus!

Das ist leider ein häufiges Problem beim Wechsel des Fenstermanagers. Sie richten alles perfekt ein, suchen sich einen atemberaubenden Desktop-Hintergrund aus... aber Ihre Gtk- und Qt-Anwendungen sehen aus, als entstammten sie den 1980er-Jahren: flache Menüs, grob gezeichnete Widgets, dunkelgraue Hintergründe und hässliche Schriften. Qt und Gtk können nicht mehr ermitteln, welches Thema sie nutzen sollen, also verwenden sie die eingebauten Standardeinstellungen. Falls Sie Gnome oder Xfce verwenden, können Gtk-Anwendungen beim Desktop nachfragen, welches Thema genutzt werden soll. Mit einem eigenständigen Fenstermanager sind die Programme auf das angewiesen, was in ihren Konfigurationsdateien steht. Für Gtk gibt es einige Optionen, die Sie ausprobieren können.

gtk-chtheme lässt Sie das Thema einstellen, das von Gtk-2.x-Anwendungen verwendet werden soll, nicht allerdings für 3.0. Für Letzteres editieren Sie die Datei **~/config/gtk-3.0/settings.ini**, sodass sie den Namen des Themas angibt:

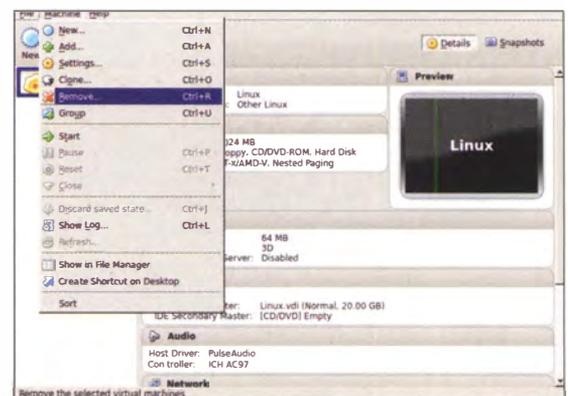
```
[Settings]
gtk-theme-name = Clearlooks
```

Falls keine dieser Vorgehensweisen den erwünschten Effekt hat und Sie Gnome installiert haben, versuchen Sie, den Dienst **gnome-settings-daemon** zu starten. Dieser wird die Gtk-Programme anweisen, Ihr neues Thema zu verwenden, aber auch Ihren Desktop-Hintergrund zurücksetzen. Um Qt-Anwendungen zur Nutzung von Gtk-Themen zu verwenden, öffnen Sie die Datei **~/config/Trolltech.conf** und

fügen Folgendes hinzu:

```
style=GTK+
```

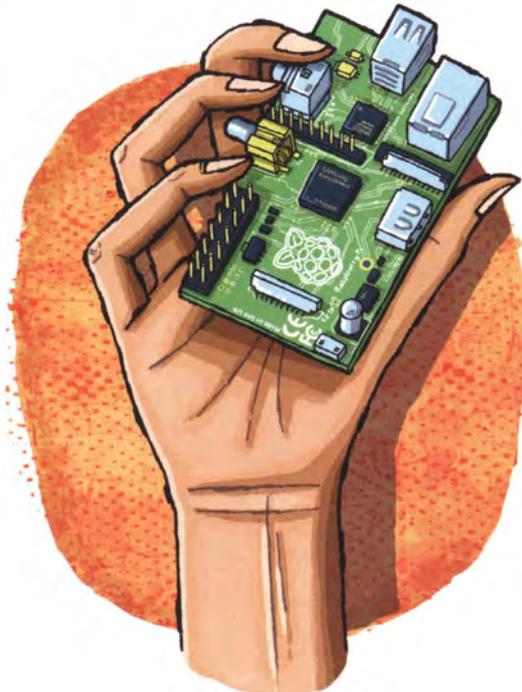
Alternativ können Sie das Paket **qt4-qtconfig** installieren und es mit **qtconfig** ausführen.



› So sieht VirtualBox (eine Qt-Anwendung) mit dem Standard-Qt-Thema aus: Windows 95 lebt!

# System: Eigene

Beim Raspberry Pi müssen Sie nicht einmal das Ende der Bootsequenz abwarten, bevor Sie an ihm herumbasteln können. John Lane zeigt Ihnen, wie Sie sich Ihre ganz persönliche Distribution zusammenstellen.



In diesem Tutorial zeigen wir Ihnen, wie man ein individuelles Image für das Raspberry-Pi-Betriebssystem erstellt. Sie könnten ja den Wunsch verspüren, die Zusammenstellung der Pakete im Vergleich zu offiziellen Images zu ändern, die Konfiguration anzupassen und möglicherweise sogar den Kernel zu bearbeiten. Sie werden nach unserem Kurs in der Lage sein, das Image auf eine SD-Karte zu schreiben, um damit den Raspberry Pi zu booten. Sie könnten aber auch einen Emulator verwenden, um das Image auf Ihrem PC zu booten. Da es zwar möglich ist, ein maßgeschneidertes Image direkt auf dem Pi zu erstellen, dies jedoch nur sehr langsam vonstatten gehen würde, erklären wir Ihnen außerdem, wie Sie Ihren PC als Cross-Compiler verwenden oder auf ihm sogar ARM-Code ausführen können.

Es ist zwar technisch kein Problem, auch an einem laufenden System Änderungen vorzunehmen, indem man einzelne Pakete hinzufügt oder entfernt, doch ist dies mit Risiken verbunden, denn ein einziger Fehler genügt unter Umständen, um einen Absturz herbeizuführen. Außerdem sind Änderungen am laufenden System schwieriger zu reproduzieren und vor allem zu automatisieren. Die Erstellung einer eigenen Distribution ist also aus unserer Sicht die weit bessere Möglichkeit.

Um ein individuelles Betriebssystem-Image zu erzeugen, benötigen wir zunächst einmal bestimmte Werkzeuge. In diesem Tutorial verwenden wir Arch Linux als Ausgangspunkt, denn das ist eine tolle Distribution, die sowohl für den Raspberry Pi als auch für den PC erhältlich ist. Wir setzen an

dieser Stelle voraus, dass Sie die neueste offizielle Version von Arch Linux samt Updates auf Ihrem Pi installiert haben. Booten Sie Arch Linux auf dem Pi und installieren Sie gegebenenfalls die nötigen Tools und Pakete:

```
# pacman -Syu
# pacman -S base-devel python2 git parted dosfstools
```

## Wählen Sie Ihre Garnitur

Der erste Schritt hin zu einem maßgeschneiderten Betriebssystem-Image ist die Überlegung, welche Dinge es enthalten soll. Eine gute Möglichkeit, sich der Auswahl anzunähern, ist es, bei der Standardausstattung von Arch Linux anzusetzen und auf dieser Basis Pakete zu entfernen oder bei Bedarf hinzuzufügen. Nicht erwünschte Kernel-Pakete sollten weggelassen und die speziellen Raspberry-Pi-Pakete ergänzt werden.

Falls Sie individuelle Features einbauen möchten, müssen Sie einen eigenen Kernel zusammenstellen. Es kann außerdem sinnvoll sein, für Ihre Zwecke überflüssige Kernel-Funktionalitäten zu deaktivieren, um Platz zu sparen – besonders im Hinblick auf die beschränkten Speicherressourcen des Raspberry Pi.

Da es auf die herkömmliche Weise über zehn Stunden dauern würde, den Kernel auf dem Pi zu kompilieren, ist eine Alternativmöglichkeit dringend erforderlich. Eine Variante ist die Verwendung eines Cross-Compilers. Eine andere bietet **distcc**, allerdings brauchen Sie auch bei diesem Client-Server-Tool einen Cross-Compiler, da der **distcc**-Server ARM-Code erzeugen muss.

Die nachfolgenden Befehle demonstrieren, wie wir mithilfe von *Pacman*, dem Paketmanager von Arch Linux, die Paketliste der Basisgruppe holen, die unerwünschten Pakete löschen und die Pakete des Raspberry-Pi-Kernels sowie Firmware-Pakete hinzufügen. Sie können auch einen Texteditor benutzen, um damit weitere Pakete gemäß Ihren Vorstellungen zuzufügen (etwa *OpenSSH*) oder wegzunehmen.

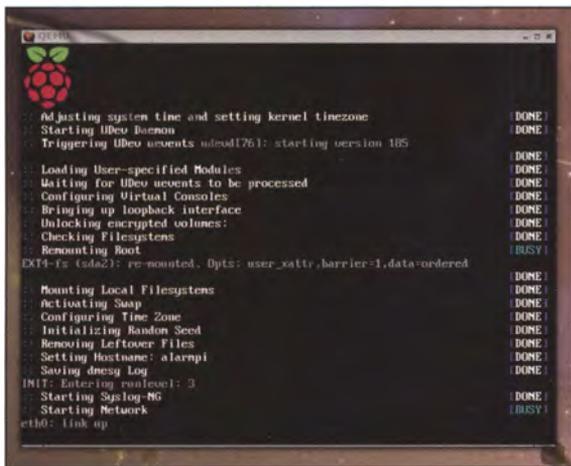
```
$ pacman -Sg base | awk '{print $2}' | grep -v "^\(linux\|kernel\)'" | tr "\n" ' ' > base_packages
$ sed -i -e 's/\/linux-raspberrypi linux-headers-raspberrypi raspberrypi-firmware/' base_packages
```

Nachdem Sie die Paketzusammenstellung festgelegt haben, müssen Sie die Pakete für das neue Image installieren. Mit **mkarchroot** geht das ganz leicht: Es installiert Pakete in ein Unterverzeichnis, welches Sie dann in ein Image umwandeln können.

```
# mkarchroot archroot $(cat base_packages)
```

Dieses Kommando erzeugt ein neues Unterverzeichnis namens **archroot**, das alles enthält, was für ein vollständiges System benötigt wird. Sie könnten es sogar mit **chroot** zum

# Distribution



► Der Prozessor-Emulator QEMU emuliert einen Raspberry Pi, der mit unserem Image Marke Eigenbau läuft.

Root-Verzeichnis machen und trotzdem würde alles weiter funktionieren. In der Tat ist dies der Weg, den Sie gehen würden, wenn Sie ein Root-Passwort setzen oder Benutzer hinzufügen wollten.

Wir erzeugen das System-Image in einer Datei. Sie könnten es auch direkt auf einer SD-Karte erstellen, doch ist es durchaus nützlich, das Image auf der Festplatte zu haben, da es dann schnell auf eine weitere Karte geschrieben werden kann oder sogar in einem Emulator zu booten geht, ohne dass es überhaupt auf irgendeine Karte geschrieben wurde. Der Kernel hat eine Loop-Funktionalität, mit deren Hilfe eine Datei wie eine physische Festplatte behandelt und partitioniert werden kann. Das Modul erzeugt einen normalen Geräteknoten in

`/dev`, über den die Datei in dieser Weise angesprochen wird.

Das Image muss groß genug sein, um das Dateisystem aufnehmen zu können, aber auch klein genug, um auf eine SD-Karte zu passen. Unser Image hat ein Volumen von 2 GiB, entspricht hinsichtlich seiner Größe also dem offiziellen Image. Stellen Sie sicher, dass das Loop-Modul des Kerns geladen wurde, generieren Sie die Datei, und erzeugen Sie ein Loop-Gerät dafür:

```
# modprobe loop
$ truncate -s 2G myimage
$ device=$(losetup -f)
# losetup $device myimage
```

Wir verwenden `losetup` hier zweifach: erstens, um dem Loop-Gerät einen Gerätenamen (üblicherweise `/dev/loop0`) zuzuteilen, und zweitens, um es zu erzeugen (wir speichern den Gerätenamen in einer Variablen, um später darauf Bezug nehmen zu können). Das Image benötigt eine bestimmte Partitionierung: Die erste Partition muss FAT16 verwenden

## Innerhalb des Images arbeiten

Manchmal ist es nützlich, innerhalb eines Dateisystem-Images wie `archroot` arbeiten zu können. Dies geht mithilfe von `chroot`, aber es ist ratsam, zuvor einige Teile des Dateisystems per Bind-Mount einzuhängen. Sie können dieses kleine Skript verwenden:

```
#!/bin/bash
mkdir -p $1/{dev/pts,proc}
mount proc -t proc $1/proc
mount devpts -t devpts $1/dev/pts
chroot $1 /usr/bin/env -i
TERM="$TERM" /bin/bash --login
umount $1/{dev/pts,proc}
```

und die Bootdateien sowie das Kernel-Image enthalten, während die zweite Partition – mit `ext4` – für das Root-Dateisystem benötigt wird. Falls Sie eine Austauschpartition wünschen, können Sie zu diesem Zweck noch eine dritte Partition einrichten. Die Boot-Firmware des Raspberry Pi hält nach einer MS-DOS-artigen MBR-Partitionstabelle Ausschau (anders als das BIOS eines PCs führt die Firmware keinen Bootloader aus dem MBR aus). Erzeugen Sie mit dem Kommando `parted` die Partitionstabelle auf der Image-Datei:

```
# parted -s $device mktable msdos
```

Wenn Sie Partitionen erstellen, ist es ratsam, diese mit den Erase Blocks der zu verwendenden SD-Karte in Übereinstimmung zu bringen. Bei den meisten Karten haben die Erase Blocks eine bevorzugte Größe von 4 MiB, doch Sie können das mit dem folgenden Befehl genau ermitteln:

```
$ cat /sys/class/block/mmcbk0/device/preferred_erase_size
```

Bei einer bevorzugten Größe von 4 MiB je Erase Block besteht alle 8.192 Sektoren eine Übereinstimmung (da die Sektoren je 512 Byte umfassen), daher sollten die Partitionen bei einem Sektor beginnen, dessen Nummer ein Vielfaches von 8.192 darstellt: Der erste Erase Block beinhaltet die ersten 8.192 Sektoren (0 bis 8.191), der zweite die nächsten (8.192 bis 16.383) und so weiter. Der Sektor 0 ist der Partitionstabelle vorbehalten, darum muss die erste Partition mit Sektor 8.192 beginnen.

Eine gute Größe für die Boot-Partition wäre 40 MiB, dies entspricht 81.920 Sektoren. Das ist ausreichend für die Bootdateien und stimmt überdies mit der Grenze eines Erase Blocks überein. Lassen Sie die Boot-Partition demzufolge mit Sektor 8.192 beginnen und mit Sektor 90.111 (8.192 + 81.920 - 1) enden:

```
# parted -s $device unit s mkpart primary fat32 8192 90111
```

Reservieren Sie gegebenenfalls ein bisschen Platz für eine Austauschpartition (die SD-Karte dafür zu verwenden, wäre aus Geschwindigkeitsgründen nicht ratsam). Als Beispiel erzeugen wir eine Austauschpartition von 256 MiB (also 524.288 Sektoren oder 64 Schreibblöcke).

Unser 2-GiB-Image besteht aus 4.194.304 Sektoren, das heißt, der letzte Sektor trägt – da die Zählung bei 0 beginnt

## Quick-Tipp

Falls Sie ein 32-Bit-System verwenden, können Sie auf einen vorgefertigten Cross-Compiler bei den Raspberry-Pi-Tools zurückgreifen. Dieser heißt `arm-bcm2708-linux-gnueabi` und ist unter <https://github.com/raspberrypi/tools> zu finden.

## Quick-Tipp

Holen Sie mehr aus Ihrem Mehrkernprozessor heraus: Beim Befehl `make` sorgt der Parameter `-j` dafür, dass gleichzeitig mehrere Prozesse gestartet werden. Geben Sie im Falle des Core i7 mit seinen vier Kernen (acht Threads) den Wert 8 ein.

»

- » – die Nummer 4.194.303. Die Root-Partition kann den gesamten Speicherplatz zwischen der Boot-Partition und der Austauschpartition belegen und würde somit mit Sektor 90.112 beginnen und vor dem ersten Sektor der 256 MiB großen Austauschpartition enden. Heraus kommen also die folgenden Kommandos zur Partitionierung:

```
# parted $device unit s mkpart primary ext2 90112 3670015
# parted $device unit s mkpart primary linux-swap 3670016 4194303
```

Wenn Sie möchten, können Sie sich die Partitionstabelle mittels des Befehls **parted -s \$device unit s print** ausdrücken. Als Nächstes generieren Sie die Dateisysteme. Das Loop-Gerät muss erneut erstellt werden, damit Geräteknoten für die Partitionen erzeugt werden (mithilfe der Option **-P**).

```
# losetup -d $device
# device=$(losetup -f)
# losetup -P $device myimage
```

Stimmen Sie die interne Struktur der Dateisysteme auf die Sektoren ab. Zur Abstimmung eines FAT-Dateisystems müssen Sie die Größe seiner Dateizuordnungstabelle (FAT) kennen. Sie müssen ein Dateisystem erstellen, um diese zu ermitteln:

```
mkfs.vfat -I -F 16 -n boot -s 16 -v ${device}p1 | grep "FAT size"
```

Um eine Übereinstimmung zu erzielen, erweitern Sie den reservierten Teil in der Weise, dass der Datenbereich an einer Blockgrenze beginnt. Die Größe des reservierten Bereichs sollte der Größe der Schreibblöcke abzüglich der beiden Dateizuordnungstabellen (FAT) entsprechen. Wenn wir unser Beispiel mit einer FAT-Größe von 32 Sektoren weiterführen, errechnet sich daraus ein reservierter Bereich von  $8.192 - 2 \times 32 = 8.128$  Sektoren. Erstellen Sie das Dateisystem erneut auf Grundlage dieser Information:

```
mkfs.vfat -I -F 16 -n boot -s 16 -R 8128 -v ${device}p1
```

Für die Root-Partition verwenden Sie bitte ein ext4-Dateisystem. Da bei ext4 die Erase Blocks lediglich 4 KiB groß sind, brauchen Sie diesmal keine Sektorenangaben zu machen:

```
$ mkfs.ext4 -O ^has_journal -L root ${device}p2
```

Mounten Sie die Dateisysteme und kopieren Sie die Dateien vom **archroot**-Unterverzeichnis an die richtige Stelle. Die Boot-Partition ist bei **/boot** auf der Root-Partition eingehängt, sodass die Bootdateien in der korrekten Partition platziert werden:

```
# mount ${device}p2 /mnt
# mkdir /mnt/boot
# mount ${device}p1 /mnt/boot
```

```
# (cd archroot ; cp -a * /mnt)
```

```
# umount /mnt/boot /mnt
```

Schreiben Sie nun das Image auf die SD-Karte. Falls Ihr Raspberry Pi das Root-Dateisystem auf der Karte hat, müssen Sie das Image auf einen anderen Rechner mit einem Kartenlesegerät kopieren:

```
# dd if=myimage of=/dev/mmcblk0 bs=4M
```

Fahren Sie den Raspberry Pi herunter, stecken Sie die neue Karte ein und booten Sie wieder. Alternativ können Sie das Image auch auf Ihren PC kopieren und es dort in einem Emulator laufen lassen.

Das **linux-raspberry**-Paket, das wir in unserem Image Marke Eigenbau verwendet haben, enthält das offizielle Kernel-Image, doch es lässt sich relativ einfach anpassen. Sie benötigen dazu den Quellcode des Kernels sowie einen Compiler, um ein neues ausführbares Kernel-Image zu erzeugen.

Ein Compiler generiert normalerweise ausführbaren Code für die CPU derselben Maschine, doch mithilfe eines Cross-Compilers ist es möglich, auch für den Prozessor eines anderen Geräts lauffähige Programme herzustellen. Der Cross-Compiler erlaubt es, die leistungsstärkere x86-Hardware zum Kompilieren des Codes für den ARM-Chip des Raspberry Pi zu benutzen, was deutlich schneller geht, als wenn man dies direkt auf dem Pi tun würde.

Der normale Compiler trägt den Namen **gcc**, während der Cross-Compiler für ARM-Hardware **arm-linux-gnueabi-gcc** heißt. Letzterer befindet sich nicht in den Repositories, aber man kann ihn mithilfe von **yaourt** leicht zusammenstellen:

```
yaourt -S arm-linux-gnueabi-gcc
```

Bei **yaourt** müssen Sie mehrere Eingabeaufforderungen beantworten. Wählen Sie „no“ bei der Frage, ob Sie das PKGBUILD editieren möchten, und „yes“, wenn das Tool wissen möchte, ob Sie ein Paket bauen oder installieren möchten.

## Hello world!

Bevor wir weitermachen, sollten wir kurz den Cross-Compiler checken. Legen Sie eine neue Datei namens **hello.c** an:

```
#include <stdio.h>
int main ()
{
    printf("Hello world!\n");
    return 0;
}
```

Nun kompilieren Sie die Datei für die ARM-Architektur und überprüfen den Typus:

```
$ arm-linux-gnueabi-gcc -o hello hello.c
$ file hello
```

```
hello: ELF 32-bit LSB executable, ARM, version 1 (SYSV),
dynamically linked (uses shared libs), for GNU/Linux 2.6.27,
not stripped
```

Wenn Sie die Datei auf Ihren Raspberry Pi verschieben, sollte sie sich ausführen lassen:

```
$ ./hello
Hello world!
```

Wird diese Ausgabe angezeigt, dann wissen Sie, dass der Cross-Compiler funktioniert. Wenden wir uns also wieder dem Kernel zu:

```
$ git clone --depth 1 git://github.com/raspberrypi/linux.git
$ cd linux
$ ssh root@alarmpi zcat /proc/config.gz > .config
$ make -j 8 ARCH=arm CROSS_COMPILE=
```

### Quick-Tipp

Falls Sie **Yaourt** noch nicht haben, finden Sie den Quellcode im Arch User Repository (AUR). Sie können sich das Selberbauen jedoch ersparen, wenn Sie das Paket aus einem Repository verwenden: <http://archlinux.fr/yaourt-en>

### Quick-Tipp

Die Raspberry-Pi-Utilities unseres Autors John Lane finden Sie unter <https://github.com/johnlane/rpi-utils>.

## Bootsequenz des Raspberry Pi

Der Pi bootet anders als ein PC. Beim Pi wird der Prozess nicht vom ARM-Chip, sondern vom Grafikchip aus gestartet. Nach dem Einschalten initiiert die GPU den ersten Bootloader (ROM-Firmware), der dann weitere Stufen von der ersten (FAT16-formatierten) Partition der SD-Karte nachlädt. Der zweite Bootloader namens **bootcode.ini** wird im L2-Cache der GPU ausgeführt. Er lädt **loader.bin**, den Bootloader der dritten Stufe, ins RAM, dessen Aufgabe es ist, die

GPU-Firmware **start.elf** auszuladen. Die weiteren Dateien **config.txt** und **cmdline.txt** ermöglichen die Konfigurierung des Bootprozesses. Schließlich lädt **start.elf** das Kernel-Image von Linux namens **kernel.img** in den RAM-Speicher des ARM-Prozessors, der damit aktiviert ist. Das Image wird ausgeführt und der Linux-Bootprozess beginnt. Der Master Boot Record (MBR) – Sektor 0 – der SD-Karte enthält die Partitionstabelle.

arm-linux-gnueabi- menuconfig -k

Wir laden den Quellcode des Kernels von github herunter, wobei **--depth** dafür sorgt, dass nur die neueste Version und keine vorherigen Varianten geholt werden. Danach kopieren Sie die aktuelle Kernel-Konfiguration des Raspberry Pi in eine Datei namens **.config**, öffnen diese im Kernel-Editor menuconfig, nehmen die gewünschten Änderungen vor und speichern die Datei. Sie sind nun in der Lage, den Kernel und dessen Module zu bauen:

```
$ make -j 8 ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- -k
$ mkdir -p modules
$ make -j 8 -k ARCH=arm CROSS_COMPILE=
arm-linux-gnueabi-modules_install INSTALL_MOD_
PATH=modules
```

Wir haben außerdem ein neues Unterverzeichnis dafür angelegt. Nachdem das neue Image des Kernels und der Module erstellt worden ist, können Sie es auf den Pi überspielen und diesen dann neu booten (zuvor sollten Sie aber ein Backup des alten Kernels und der Module machen):

```
$ scp arch/arm/boot/Image root@alarmpi:/boot/kernel.img
$ cd modules/lib/modules
$ tar cJf - * | ssh root@alarmpi '(cd /usr/lib/modules; tar xjf -)'
$ ssh root@alarmpi reboot
```

Alternativ können Sie alles auch in Ihrem **archroot**-Verzeichnis installieren, bevor Sie das Image erstellen. Kopieren Sie **arch/arm/boot/Image** nach **archroot/boot/kernel.img** sowie **modules/lib/modules** nach **archroot/usr/lib**.

## Emulator und Cross-Compiler

Es ist ebenfalls möglich, das Image in einem ARM-Emulator laufen zu lassen und dadurch einen virtuellen Raspberry Pi auf Ihrem x86-PC zu betreiben. **QEMU** ist ein Prozessor-Emulator, der ARM-Code auf einem PC mit x86-Architektur interpretieren kann. Sie installieren **QEMU** mit **pacman -S qemu**. Erstellen Sie ein Verzeichnis und kopieren Sie das Image dorthin. Sie benötigen überdies ein Kernel-Image speziell für den Emulator: Verwenden Sie **kernel-qemu**, welches Sie in den Archiven finden und speichern Sie es am selben Ort wie Ihr Image. Nun können Sie den Emulator mit folgendem Kommando ausführen:

```
qemu-system-arm -machine versatilepb -cpu arm1176 -m 256 \
-no-reboot -serial stdio -kernel kernel-qemu \
-append "root=/dev/sda2 panic=1" -hda myimage
```

Mithilfe des Emulators kann man sogar ARM-Code direkt aus der Befehlszeile eines x86-Rechners ausführen. Hierbei macht das Programm sich ein Kernel-Feature namens **binfmt\_misc** zunutze. Dazu sind jedoch Vorbereitungen nötig:

```
# mount binfmt_misc -t binfmt_misc /proc/sys/fs/binfmt_misc
```

Das ARM-Binärformat muss dem Kernel angegeben werden, damit dieser weiß, was er zu tun hat. Die Erkennung basiert auf Musterübereinstimmung am Anfang der Datei. Geben Sie ein:

```
echo ':arm:M::\x7fELF\x01\x01\x01\x00\x00\x00\x00\
x00\x00\x00\x00\x00\x02\x00\x28\x00:\xff\xff\xff\
\xff\xff\xff\x00\xff\xff\xff\xff\xff\xff\xff\xff\
\xff:/usr/bin/qemuarm-static:' > /proc/sys/fs/binfmt_misc/
register
```

Im Folgenden wird jeder Versuch, ARM-Binärcode auszuführen, stattdessen zur Ausführung von **/usr/bin/qemu-arm-static** führen, wobei der Pfad des Codes in Form eines Parameters weitergeleitet wird.

So weit, so gut. Was allerdings schwierig ist, ist die

## Image auf dem PC erstellen

Mithilfe eines Emulators können Sie das **archroot**-Unterverzeichnis auch auf Ihrem PC anlegen statt auf dem Raspberry Pi. Dazu müssen Sie den Paketmanager so konfigurieren, dass er ARM-Pakete herunterlädt, und das **mkarchroot**-Skript so einstellen, dass es **qemu-arm-static**

im **archroot**-Verzeichnis installiert, während dieses erzeugt wird. Nachdem Sie diese letzten Puzzleteile eingefügt haben, sind Sie in der Lage, auf Ihrem PC ein vollständiges Image für Ihren Raspberry Pi samt maßgeschneidertem Kernel und zusätzlichen Paketen aufzubauen.

Erstellung von Paketen, da hierbei Abhängigkeiten (dependencies) ins Spiel kommen und auf einem x86-Computer keine ARM-Abhängigkeiten installiert werden können. Sie können das Problem jedoch halbwegs umgehen, indem Sie das anfangs erwähnte Tool **distcc** einsetzen. Damit können Sie die Paketerstellung auf Ihrem Pi erledigen, jedoch die Kompilierungsarbeit per Cross-Compiling an Ihren PC auslagern.

Um **distcc** verwenden zu können, müssen Sie es sowohl auf dem Raspberry Pi als auch auf dem PC installieren, in beiden Fällen mit dem Befehl **pacman -S distcc**. Der Pi fungiert als Client und kontrolliert die Erstellung der Pakete mittels **makepkg**, das Sie noch für die Benutzung von **distcc** konfigurieren müssen, und zwar durch Angabe der Details Ihres **distcc**-Servers. Ändern Sie hierzu **/etc/makepkg.conf** wie folgt:

```
BUILDDENV=(fakeroot distcc color !ccache)
DISTCC_HOSTS="10.0.200.12"
MAKEFLAGS="-j8"
```

Als Server fungiert Ihr PC, er kompiliert den Code. In der Konfigurationsdatei **/etc/conf.d/distccd** muss angegeben werden, welche Hosts sich verbinden dürfen. Dabei sollte der Server als Daemon gestartet werden:

```
/etc/rc.d/distccd start
```

Sind Client und Server konfiguriert, kann **makepkg** auf dem Pi und das Kompilieren auf dem PC ausgeführt werden. Allerdings muss noch der korrekte Compiler in der Datei **distccd** eingetragen werden: Standardmäßig ist dort **gcc** eingestellt, wir benutzen jedoch stattdessen den Compiler **arm-linux-gnueabi-gcc**. Modifizieren Sie daher den Pfad wie folgt:

```
# ln -s /usr/bin/arm-linux-gnueabi/gcc /use/bin/
arm-linux-gnueabi-gcc
# sed -i -e '/^PID=/iPATH=/usr/bin/arm-linux-gnueabi:\
$PATH' /etc/rc.d/distcc
```

## Root-Partition verschieben

Falls Sie Ihr Root-Dateisystem nicht auf der SD-Karte haben möchten, können Sie es verschieben. Dies geht ganz einfach, indem Sie das bestehende Root-Dateisystem von der SD-Karte in eine Partition der USB-Festplatte kopieren. Eventuell müssen Sie die Größe anpassen.

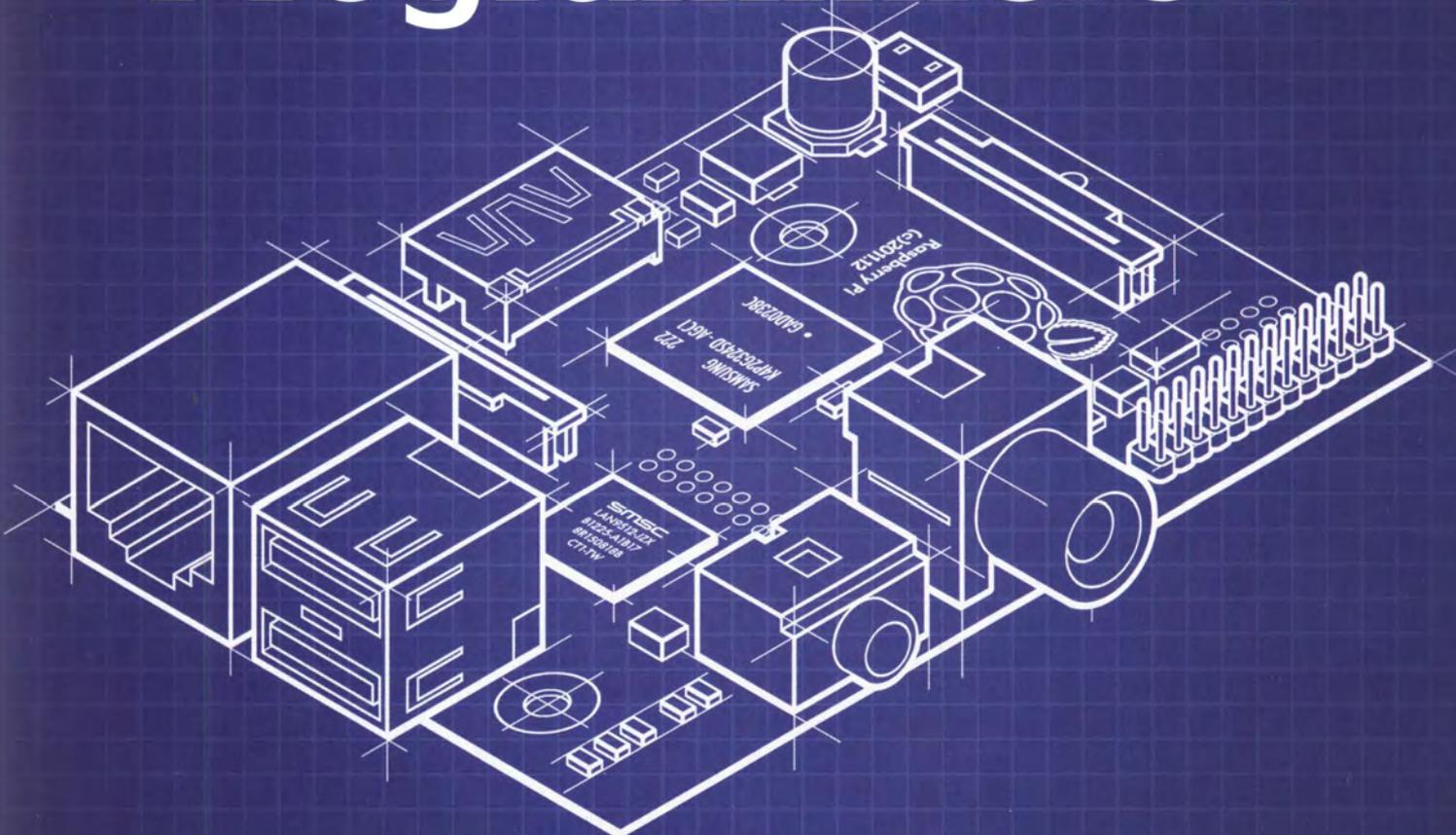
```
# dd if=/dev/mmcblk0p2 of=/dev/sda1 bs=4M
# fsck /dev/sda1
# resize2fs /dev/sda1
```

Ändern Sie dann in der Datei **/boot/cmdline.txt** die Position der Root-Partition im Bootkommando, indem Sie **root=/dev/mmcblk0p2** durch **root=/dev/sda1** ersetzen. Nach einem Neustart befindet sich die Root-Partition auf der Festplatte. Die benötigten USB-Treiber sind normalerweise bereits im Kernel vorhanden. ■

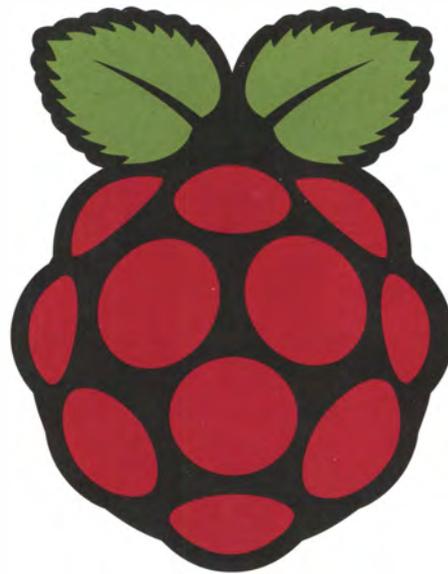
### Quick-Tipp

Sollte beim Bau des Kernels der Arbeitsschritt **modules\_install** zu der Fehlermeldung **No rule to make target 'modules/lib/firmware/./'** führen, wiederholen Sie den Schritt einfach – dann sollte es klappen.

# Raspberry Pi Programmieren



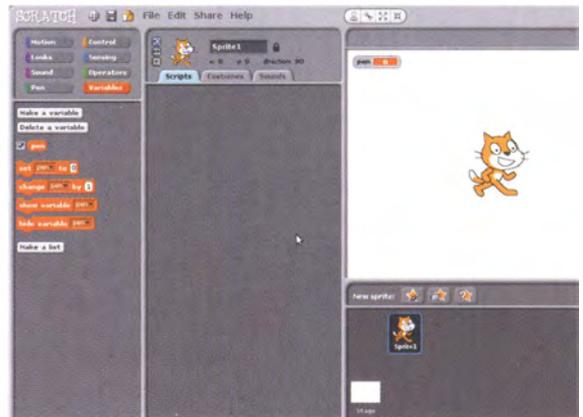
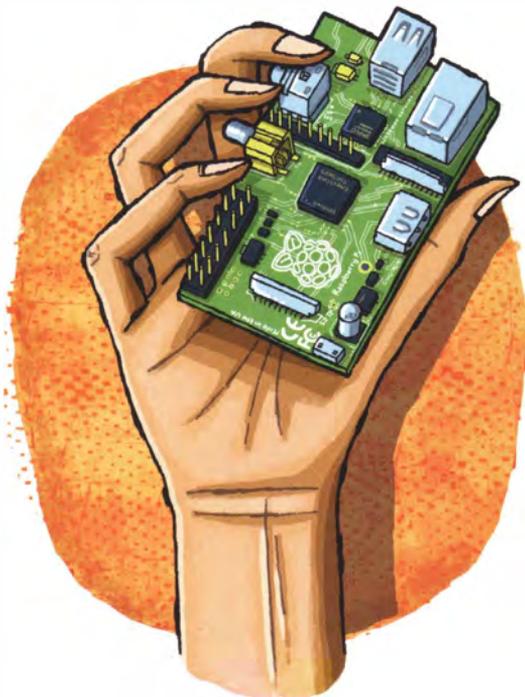
**D**er Raspberry Pi ist exzellent geeignet, um das Programmieren zu lernen. Dieses Kapitel wird Ihnen daher die wichtigsten Grundlagen vermitteln, um Ihre Projekte erfolgreich in die Tat umzusetzen. Damit das Lernen allerdings nicht zu trocken wird, legen wir auch Wert auf etwas Spaß. So können Sie im Handumdrehen Ihr eigenes Arcade-Spiel verwirklichen oder mit Minecraft Erfahrungen sammeln.



Scratch und Python.....	<b>26</b>
Scratch: Quiz.....	<b>30</b>
Scratch: Arcade.....	<b>34</b>
Scratch: Gejagt.....	<b>38</b>
Die Legende von Python.....	<b>42</b>
Chat: Wir bauen Bots!.....	<b>46</b>
Ein grafisches Interface bauen.....	<b>50</b>
Twitter: Tweets vorlesen lassen.....	<b>54</b>
Python: Hacken Sie Minecraft.....	<b>58</b>
Minecraft: Das Pi-Katapult.....	<b>60</b>
Wir bauen ein Plattformspiel.....	<b>64</b>
Logik lernen mit Guido.....	<b>68</b>

# Scratch und Py

Lernen Sie zum Einstieg in das Programmieren die zwei benutzerfreundlichen Standardsprachen von Raspbian kennen: Scratch und Python.



› **Abbildung 2:** Den Variablen wird per Voreinstellung eine „0“ zugeordnet. Im Malbereich können Sie immer deren aktuellen Wert im Auge behalten.

› **Abbildung 1:** Da die Blöcke farblich kodiert sind, wissen Sie immer genau, aus welchem Menü Sie sie auswählen können.

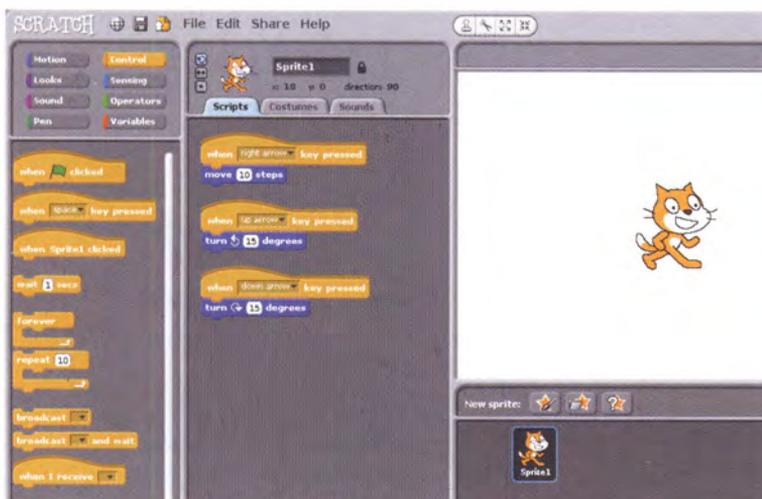
Die Grundidee des Raspberry Pi ist es, Einsteigern die Computertechnik und das Programmieren näherzubringen. In dieser Übung bekommen Sie eine erste kleine Einführung in die beiden Programmiersprachen, die bei Raspbian zum Lieferumfang gehören: Scratch und Python. Scratch ist besonders gut dazu geeignet, sich Grundkenntnisse anzueignen. Da alle Befehle vorgefertigt sind und Sie sie per Drag & Drop in Ihr Skript integrieren können, brauchen Sie sich keine Befehle zu merken. Wir beginnen mit einem einfachen Malprogramm,

welches Sie mittels der Pfeiltasten Linien auf Ihrem Bildschirm zeichnen lässt. Zunächst müssen wir einen Code schreiben, der uns die Katze bewegen lässt. Dafür verwenden wir drei verschiedene Blöcke, von denen jeder bei einem bestimmten Tastendruck ausgeführt werden soll.

Klicken Sie auf **Steuerung** und ziehen Sie den Block **Wenn Taste <Leertaste> gedrückt** in das Feld **Skripte**. Dadurch wird ein Skript erstellt, das sich immer dann aktiviert, wenn die Leertaste betätigt wird. Benutzen Sie nun das Dropdown-Menü, um **<Leertaste>** durch **<Pfeil nach rechts>** zu ersetzen, und ziehen Sie anschließend den Block **gehe <10>er-Schritt** in Ihr Skript direkt unter den vorherigen Befehl. Damit bewegen Sie die Katze vorwärts, sobald Sie die rechte Pfeiltaste drücken. Im nächsten Schritt erstellen Sie ähnliche Skripte, mit denen Sie die Katze im Uhrzeigersinn drehen, wenn die Pfeiltaste nach unten gedrückt, oder gegen den Uhrzeigersinn, wenn die Pfeiltaste nach oben gedrückt wird. Wie das Ganze aussehen soll, zeigt Abbildung 1. Da wir uns nun auf dem Bildschirm umherbewegen können, brauchen wir einen weiteren Block, der uns zeichnen lässt. Um nicht die ganze Zeit eine Linie ziehen zu müssen, nutzen wir die **Stift**-Funktion von Scratch: Bei **senke Stift** ist der Stift aufs „Papier“ gedrückt und zeichnet, bei **hebe Stift** nicht.

## Mit Variablen arbeiten

Damit wir zwischen den Zuständen des Stifts umschalten können, muss sich unser Programm merken, in welchem Zustand er sich gerade befindet. Hierfür verwenden Programmiersprachen Variablen, in denen kleine Informationen hinterlegt und ausgelesen werden können. Bevor Sie eine Variable verwenden können, müssen Sie ihren Speicher zur Verfügung stellen. Gleichzeitig geben wir ihr noch einen Namen, damit wir in den Befehlen auf sie verweisen können. Gehen Sie dafür auf **Variable erstellen** und benennen Sie



# thon

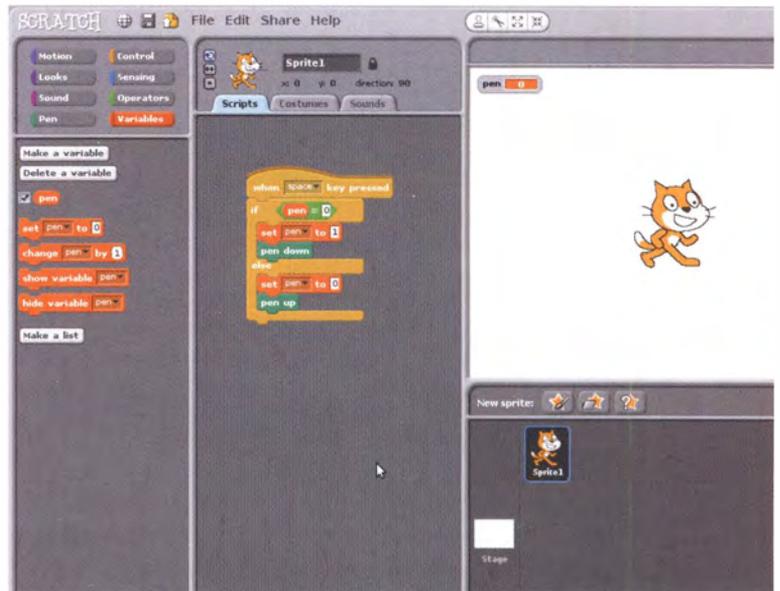
sie. Nun können Sie diverse Befehle sehen, die die Variable benutzen oder ihren Inhalt verändern können. Da wir jetzt Informationen speichern können, müssen wir dem Computer sagen, wie er sich in Bezug auf die Variable verhalten soll. Hierfür verwenden wir einen **falls <...> sonst <...>**-Block, der überprüft, ob eine Aussage wahr oder falsch ist. Ist sie wahr, wird der erste Block, andernfalls der zweite ausgeführt. Wir verwenden die Variable für **Stift**. Bei einer **0** wird der Stift aufgesetzt und der Zustand auf **1** umgestellt. Andernfalls heben wir den Stift an und die Variable wird auf **0** gestellt. Mit dieser Methode können wir zwischen beidem wechseln, indem wir die Leertaste betätigen (siehe Abbildung 3). Beachten Sie den **=**-Operator in unserer **falls <...>**-Zeile. Dieser bewirkt, dass unser erster Code-Block nur ausgeführt wird, wenn unsere Variable **Stift** eine **0** enthält (**equals**), andernfalls (**else**) wird der zweite Block ausgeführt.

## Programmschleifen

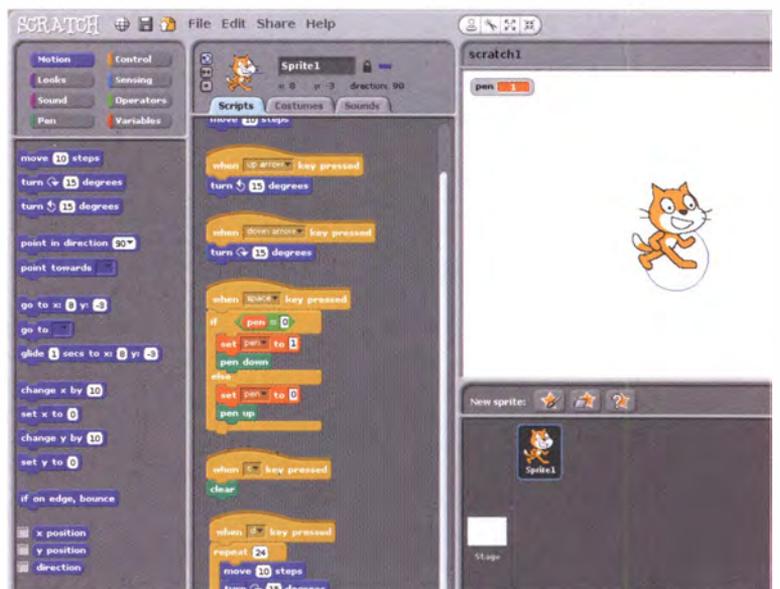
Jetzt können Sie also die Katze bewegen und Formen zeichnen. Wie wäre es mit einem Kreis? Es wird zwar eher auf ein Vierundzwanzigeck hinauslaufen, aber das kann man gelten lassen. Dazu wiederholen wir die Befehle **gehe <10>er-Schritt** sowie **drehe dich <im Uhrzeigersinn> um <15> Grad** so lange, bis der Kreis geschlossen ist. Sie könnten diese zwei Zeilen natürlich einfach 24 Mal wiederholen, doch das wäre nicht sehr sauber programmiert, es würde lange dauern und wenn Sie die Größe Ihres Kreises verändern wollten, müssten Sie die ganze Prozedur erneut vornehmen. Stattdessen werden wir eine Schleife benutzen, die sich immer wieder selber wiederholt. Es gibt eine Vielzahl verschiedener Schleifen. Einige setzen sich so lange fort, bis eine Aussage falsch wird (vergleichbar mit einem **falls <...>**-Befehl, der durchgängig wiederholt wird), unsere jedoch wird eine im Vorfeld festgesetzte Anzahl von Kommandos abarbeiten (siehe Abbildung 4).

Wie Sie sehen, kann Programmieren nicht nur Spaß machen, sondern Ihnen auch dabei behilflich sein, Ihren Rechner Aufgaben für Sie zu erledigen zu lassen. Zwar ist Scratch eine sehr simple Programmiersprache, doch Sie können damit dennoch eine ganze Menge bewerkstelligen. Zu Ihrer Inspiration finden Sie hier einige schöne Beispiele:

- » **Super Mario Galaxy V4:** Welterkundung und Sternesammeln  
<http://scratch.mit.edu/projects/162167>
- » **Wipeout 2.0:** Olle Grafik, aber cooles Gameplay  
<http://scratch.mit.edu/projects/1149306>
- » **Space War 4:** Ein klassischer Weltraumshooter  
<http://scratch.mit.edu/projects/879463>
- » **Snake Chamber:** Eine kleine Lehrstunde über Genetik und Schlangenzucht  
<http://scratch.mit.edu/projects/2758178>
- » **Day Dream:** Scratch ist auch ein hervorragendes Werkzeug für Animationen.  
<http://scratch.mit.edu/projects/40150>



» **Abbildung 3:** falls <...> sonst <...>-Blöcke ermöglichen es Ihrem Rechner, Entscheidungen zu treffen, und sind somit der Grundbaustein jedweden Programmierens.



» **Abbildung 4:** Mit der gleichen Methode können Sie geometrische Figuren in Ihr Malprogramm integrieren.

„Die visuelle Programmiersprache Scratch ist besonders gut dazu geeignet, sich Grundkenntnisse anzueignen.“

## Python

Um sich mit den Basics des Programmierens vertraut zu machen, ist Scratch eine gute Wahl. Früher oder später werden Sie jedoch an dessen Grenzen stoßen. Daher werfen wir nun einen Blick auf Python, eine beliebte universelle Programmiersprache. Im Gegensatz zu Scratch ist Python vollständig textbasiert. Das heißt natürlich nicht, dass Sie nicht auch Grafiken implementieren können, sondern dass Sie den Code selbst schreiben müssen, statt ihn per Drag & Drop zusammenzustellen.

Für die Erstellung eines Programms brauchen Sie folglich einen Texteditor. *Leafpad* wird beim Raspberry Pi mitgeliefert und taugt für den Anfang. Sollten Sie jedoch Ihre Programmierkarriere fortsetzen, wäre es ratsam, auf einen anderen Editor umzusteigen, der Ihnen gut liegt. Textverarbeitungsprogramme wie *LibreOffice Write* oder *Abiword* sind allerdings ungeeignet, da sie die Formatierung durcheinanderbringen.

Beginnen wir also, indem wir eine Datei in *Leafpad* öffnen und die erste Zeile schreiben:

```
#!/usr/bin/python
```

Diese Zeile, geheimnisvoll auch Shebang genannt, gibt dem System die Anweisung, das Programm *Python* im Dateipfad `/usr/bin/` zu starten. Dies wird bei all Ihren Python-Pro-

grammen benötigt. Nun sind wir bereit, in die wunderbare Welt des Programmierens einzutauchen. Es ist Tradition in der Computerwelt, seine

## „Nun sind wir bereit, in die wunderbare Welt des Programmierens einzutauchen.“

ersten Gehversuche mit der Textausgabe „Hello World!“ zu beginnen, und auch wir werden uns diesem Brauch nicht verschließen. Tippen Sie in eine neue Zeile:

```
print "Hello World!"
```

Speichern Sie jetzt Ihr Werk in einer Datei mit dem Namen **hello.py**. Um Ihr Programm auszuführen, müssen Sie mit einem Terminal an den Ort navigieren, an dem Sie die Datei abgespeichert haben. Hier müssen Sie dem System noch zu verstehen geben, dass es sich bei Ihrer Datei um ein ausführbares Programm handelt, indem Sie `chmod a+x hello.py` eintippen. Im Anschluss lässt sich Ihr Programm mit `./hello.py` starten. Jetzt sollte „Hello World!“ auf Ihrem Bildschirm erscheinen.

## Module

Ein tolles Feature von Python ist die große Anzahl seiner Module. Mit diesen wird Python um zahlreiche nützliche Funktionen erweitert. Zwar werden nicht alle für den Anfänger geeignet sein, aber es ist keine schlechte Idee, einen Eindruck davon zu bekommen, was alles mit der Programmiersprache möglich ist, sobald man sich ein wenig eingearbeitet hat. Dies sind unsere Top-5-Python-Module.

» **pyGames**: Spiele sind cool, Programmieren ist cool. Also muss auch Spiele programmieren cool sein. Dieses Modul hilft Ihnen dabei, Programme zu schreiben, mit denen Sie Ihre karge Freizeit verplempern können.

» **pyGTK**: Früher oder später werden Sie womöglich auf die Idee kommen, Grafik-Apps zu programmieren. Dieses Modul wird Ihnen dabei helfen.

» **pyQT**: Für eine Gnome-Umgebung ist *GTK* hervorragend geeignet. KDE-Benutzer werden sich wahrscheinlich mit *QT* besser zurechtfinden.

» **RPi.GPIO**: Der Raspberry Pi hat eine große Auswahl an GPIO-Pins, mittels derer Sie mit Ihrer Umwelt interagieren können. Das Modul hilft ihnen dabei, diese anzusteuern.

» **NumPy**: Für Zahlenfetischisten! Manipulieren Sie Zahlen mit Leichtigkeit.

Dies zeigt uns, dass Ihr System einwandfrei funktioniert. Wie bei Scratch werden wir jetzt eine Interaktion mit dem User integrieren. Um dies zu bewerkstelligen, brauchen wir bei Python eine Variable, in der die Eingabe des Benutzers gespeichert wird. Löschen Sie die „Hello World!“-Zeile wieder, sodass nur die Shebang-Zeile übrig bleibt, und schreiben Sie stattdessen:

```
name = raw_input("What is your name?")
```

Hiermit erschaffen Sie eine Variable mit der Bezeichnung **name**, lassen die Frage „What is your name?“ auf Ihrem Bildschirm erscheinen und speichern die eingegebene Antwort des Benutzers in **name**. Da der Computer wissen muss, dass es sich hierbei um Text handelt, müssen wir diesen Bereich mit zwei Hochkommata begrenzen. Jetzt können wir die Variable benutzen, um unseren **print**-Befehl etwas persönlicher zu gestalten:

```
print 'Hello', name
```

Da der Computer die Befehle in strikter Reihenfolge abarbeitet, muss diese Befehlszeile direkt unter die vorherige geschrieben werden. Vertauschen Sie die Reihenfolge, wird Ihr Programm einen Fehler ausgeben, da wir eine Variable benutzen würden, ohne sie vorher definiert zu haben. Speichern Sie nun Ihre Datei und führen Sie sie mit `./hello.py` aus.

## Entscheidungsmöglichkeiten

Auch wenn Ihr Programm hierdurch schon ein wenig an Funktionalität gewonnen hat, wirkt es trotzdem noch ein wenig leblos, da es immer die beiden gleichen Schritte abarbeitet und danach seinen Dienst einstellt. Um ein bisschen mehr Leben in die Sache zu bringen, brauchen wir eine Entscheidungsmöglichkeit, bei der der Computer abhängig von der Eingabe des Users unterschiedlich reagiert.

Erinnern Sie sich noch an den **falls <...>**-Block von Scratch? Bei Python heißt das **if**. Eine **if**-Zeile sieht prinzipiell so aus:

```
*hello.py (-) - gedit
*hello.py
#!/usr/bin/python

name = ''

while name != 'quit' :|
    name = raw_input('what is your name? ')

    if name == 'Ben':
        print "Ben, you're awesome"
    else :
        print 'Hello', name
```

» Einige Texteditoren werden mit zusätzlichen Features fürs Programmieren ausgestattet. Das hier gezeigte Gedit hebt die Syntax Ihres Codes vor, um diesen lesbarer zu machen.

if <Aussage> :

<Einrückung>Befehlsblock

<Aussage> muss dabei durch etwas ersetzt werden, das entweder wahr oder falsch sein kann. In unserem Fall überprüfen wir einfach, ob der eingegebene Name zugeordnet werden kann.

```
if name == 'Ben' :
```

Sie stellen sich vermutlich die Frage: Warum == ? Nun, Computer arbeiten nicht mit Mehrdeutigkeiten, daher sollte jedes Symbol oder Wort, das wir benutzen, nur eine Bedeutung besitzen. Da mit = einer Variable eine Wertigkeit zugeordnet wird, brauchen wir etwas anderes, wenn Dinge verglichen werden sollen. Wiederum müssen wir Ben mit Hochkommata umgeben, sodass der Computer weiß, es handelt sich um Text. Der Doppelpunkt zeigt dem Programm, dass die Zeile beendet ist und im Anschluss dem Rechner gesagt wird, was er damit tun soll.

Womöglich möchten wir, dass der **if**-Befehl mehr als eine Zeile Code enthält. Aus diesem Grund müssen wir unseren Programmier-Code in Blöcke gruppieren. Python verwendet Einrückungen. Einrückungen können ein Leerzeichen oder ein Tabulator sein, jedoch sollten Sie immer die gleiche Methode in ihrem Projekt verwenden, um die Übersicht zu behalten. Andernfalls kann es sehr schnell verwirrend werden, da Python sich nicht nach der Breite der einzelnen Einrückung richtet, sondern nach deren Anzahl.

Aber nun zurück zum Thema. Was wollen wir eigentlich machen, wenn **name == 'Ben'** ist? Nun, wir wollen ihn natürlich standesgemäß begrüßen:

```
if name == 'Ben' :
```

```
    print "Ben, you're awesome"
```

Beachten Sie die Einrückung am Anfang der zweiten Zeile und auch, wie wir Anführungszeichen verwenden. Das liegt daran, dass der Text, den wir einrahmen, bereits einen Apostroph enthält. Da wir zu anderen Benutzern nicht unhöflich sein wollen, fügen wir für den Fall, dass die **if**-Aussage falsch ist, noch einen weiteren **else**-Block hinzu.

```
else :
```

```
    print 'Hello', name
```

```
ben@ben: ~
ben@ben:~$ ./hello.py
what is your name? Ben
Ben, you're awesome
what is your name?
```

› Eine nette Begrüßung am Terminal.

## Der interaktive Modus

Eigentlich benutzen wir Python, um Skripte auszuführen, aber es geht auch anders – mit dem interaktiven Modus. Auf diese Weise geben wir eine Zeile ein und sie wird von Python umgehend bearbeitet. Danach geben wir eine weitere ein und so weiter. Wenn Sie schnell etwas ausprobieren möchten, ist dies ein durchaus nützliches Feature. Um die Python-Shell aufzurufen, geben Sie einfach **python** in die Kommandozeile ein. Ein **exit()** schließt Python wieder.

Dieser Modus ist besonders nützlich, um mit neuen Bibliotheken zu experimentieren oder Einzelbefehle einzugeben, ähnlich wie bei *Bash*.



› Python im interaktiven Modus.

Verlassen Sie diesen Modus jedoch, werden all Ihre Mühen wieder gelöscht. Daher ist es besser, einen Texteditor zu verwenden, um Programme zu schreiben.

## Schleifen

Ein letztes Feature, das wir noch implementieren, ist eine Schleife. Diese Programmschleife ist vergleichbar mit derjenigen, die wir schon in Scratch verwendet haben, jedoch wird diese nicht 24 Mal durchlaufen. Stattdessen sagen wir ihr, wann sie aufzuhören hat. Hierfür verwenden wir die **while**-Schleife, deren Syntax so aussieht:

```
while <Aussage> :
```

```
    <Einrückung>Befehlsblock
```

Das Programm soll weiterlaufen, solange wir nicht **quit** eingeben. In vielen Programmiersprachen, so auch bei Python, bedeutet ein Ausrufezeichen „nicht“. Daher wird unsere Schleife so aussehen:

```
while name != 'quit' :
```

Nun müssen wir noch vor **while** einen leeren String der Variable **name** zuordnen, damit die **while**-Aussage fehlerlos ausgeführt wird. Unser fertiges Programm sieht nun so aus:

```
#!/usr/bin/python
```

```
name = ""
```

```
while name != 'quit' :
```

```
    name = raw_input("What is your name?")
```

```
    if name == 'Ben'
```

```
        print "Ben, you're awesome"
```

```
    else :
```

```
        print 'Hello', name
```

Beachten Sie die längeren Einrückungen vor jeder **print**-Zeile. Das liegt daran, dass diese zweimal eingerückt wurden: einmal für die **while**-Schleife und ein weiteres Mal für die **if**-Aussage. Speichern Sie nun Ihr Programm als **hello.py** ab und führen Sie es als **./hello.py** aus.

## Wie weiter?

Wenn Ihnen unsere kleine Übung und das Schreiben Ihrer ersten Programme gefallen haben, werden Sie sich sicherlich fragen, was als Nächstes kommt. Scratch und Python sind bestimmt gute Programmiersprachen für Einsteiger. Suchen Sie sich also diejenige aus, die Ihnen am meisten gefallen hat. Sollten Sie sich

für Scratch entschieden haben, werden Sie auf <http://scratch.mit.edu> fündig. Neben unzähligen Projekten gibt es hier Videotutorials und Einführungskurse, die Ihnen die Programmierumgebung näherbringen.

Python hingegen ist weit mehr verbreitet, sodass Sie zum Lernen auf viel mehr Material

zurückgreifen können. Auf der offiziellen Website <http://www.python.org> finden Sie ein gutes (wenn auch etwas trockenes) Tutorial. Zusätzlich gibt es zahlreiche ausgezeichnete Bücher zu dem Thema wie etwa *Dive into Python*, das auch gratis auf der Seite <http://www.diveintopython.net> verfügbar ist.

# Scratch: Quiz

In dieser Übung entwickeln wir ein einfaches Quiz-Spiel mithilfe der visuellen Programmiersprache Scratch.

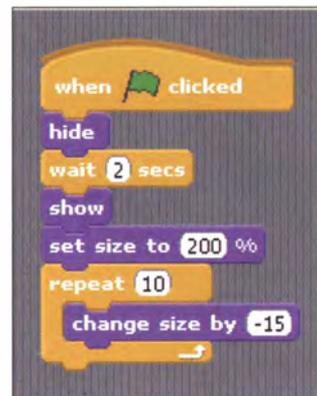
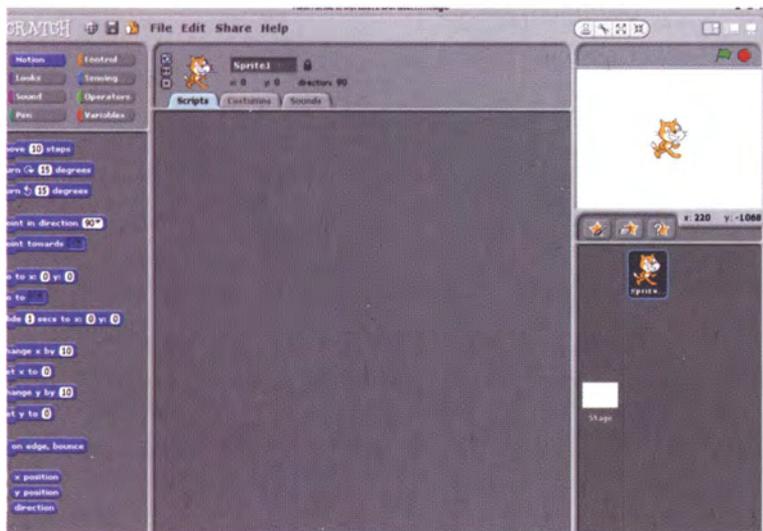
**W**ie haben Sie das Programmieren gelernt? Das typische Bild, das viele jetzt wohl vor Augen haben, ist das eines Menschen, der im Widerschein eines leuchtenden Monitors in einem ansonsten dunklen Zimmer sitzt und mühsam geheimnisvolle Worte in eine Tastatur tippt, die er zunächst selbst nicht versteht.

In vergangenen Zeiten des Programmierens mag dieses Bild gar nicht so falsch gewesen sein. Die ersten Enthusiasten eigneten sich ihr Wissen durch beständiges Hacken an, indem sie zum Beispiel unzählige Zeilen Code aus Zeitschriften abtippten.

Heute jedoch, in einer Zeit, wo Kinder bereits in der Grundschule erste Erfahrungen mit Programmierkonzepten sammeln, haben wir neue Werkzeuge zur Verfügung, um dem Nachwuchs das Coden schmackhaft zu machen – eins davon ist die visuelle Programmiersprache Scratch.

Scratch wurde am Massachusetts Institute of Technology (MIT) entwickelt. Das visuelle Konzept dieser Sprache beinhaltet die Verwendung farbiger Blöcke mit kleinen Code-Schnipseln. Jeder Satz Blöcke bringt eigene Funktionalitäten mit und führt bestimmte Programmierkonzepte auf eine spielerische Weise ein. Schon im Alter von sechs Jahren

» Bei Scratch ist die Entwicklungsumgebung in drei Bereiche aufgeteilt: **Blockbereich, Programmierbereich und Bühne.**



» **Abbildung 1.0:** Das Logo für das Spiel ist unser Objekt5.



» **Abbildung 1.2:** Dieses Skript löscht alle Spezialeffekte, die auf ein bestimmtes Objekt angewandt wurden.

können Kinder sich an Scratch heranwagen und die Sprache gehört in Großbritannien bereits zu den Lehrplänen für Schüler bis 14 Jahre.

In diesem Tutorial beziehen wir uns auf die neueste stabile Version von Scratch, die uns vorlag, und das war Version 1.4. Version 2 ist heute, im Sommer 2014, noch in der Beta-Phase, daher haben wir es nicht berücksichtigt.

Wie erwähnt, basiert Scratch auf der Verwendung von verschiedenfarbigen Blöcken, die unterschiedliche Funktionalitäten repräsentieren. Folgende Gruppen kommen vor:

- » **Bewegung (Dunkelblau)** Diese Blöcke ermöglichen es Ihnen, Objekte in Ihrem Spiel zu bewegen und kontrollieren.
- » **Steuerung (Orange)** Hier geht es um logische Mechanismen, die das Programm steuern (Schleifen und Anweisungen), und Ereignisse (wie etwa das Drücken einer Taste), die bestimmte Aktionen auslösen. Bei Scratch 2.0 bekommen Ereignisse eine eigene Gruppe.
- » **Aussehen (Lila)** Diese Programmierblöcke können Farbe, Größe und Erscheinungsform (Kostüm) von Objekten verändern und führen außerdem interaktive Elemente wie beispielsweise Sprechblasen ein.
- » **Fühlen (Hellblau)** Mit Fühlen ist die Wahrnehmung von Input gemeint, etwa generelle Programmeingaben wie Tastenanschläge, aber auch Kollisionen von Objekten sowie die Position eines Objekts auf dem Bildschirm.
- » **Klang (Pink)** Diese Blöcke fügen dem Programm Musik und Geräusche hinzu.

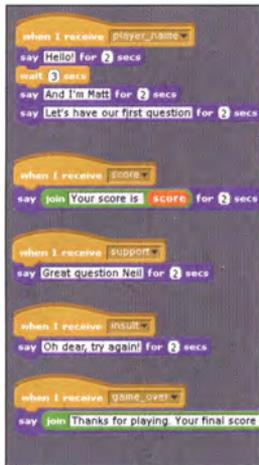
## Scratch online

Scratch ist für viele Plattformen verfügbar. Es ist auf jedem Raspberry Pi, der mit dem Betriebssystem Raspbian läuft, vorinstalliert. Sie können die Software bei Bedarf von <http://scratch.mit.edu> herunterladen. Es gibt allerdings auch eine Online-Version von Scratch, die (abgesehen von kleineren Unterschieden) den gleichen

Funktionsumfang besitzt wie die Desktop-Version, jedoch keinerlei Installation oder Download erfordert. Auch die Online-Version finden Sie unter dem genannten Link. Sie können auf der Seite auch Ihre eigenen Projekte in der Cloud speichern und später von einem beliebigen Rechner aus wieder darauf zugreifen. Sie können

die Projekte sogar herunterladen und sie offline weiter bearbeiten.

Die neueste Version von Scratch ist die Version 2.0, allerdings befindet sich diese noch in der Beta-Phase. Die aktuellste stabile Version, auf die wir auch in dieser Übung Bezug nehmen, ist die Version 1.4.



› **Abbildung 1.1:** Nachrichten lösen bestimmte Code-Sequenzen aus.

sich typischerweise zuerst an dem Farbsystem und durch den natürlichen Prozess des Spielens verstehen sie nach und nach die Wirkungsweise der Blöcke sowie deren Kombinationsmöglichkeiten.

## Die Entwicklungsumgebung

Bei Scratch ist die Entwicklungsumgebung sehr klar strukturiert. Sie ist in die folgenden drei Spalten unterteilt:

- › **Blockbereich** Hier lagern die farbigen Blöcke – nach Funktionalität sortiert – und warten auf ihre Verwendung.
- › **Programmierbereich** Hierher können die Blöcke gezogen werden, um den Programmcode zu erzeugen.
- › **Bühne** Auf der sogenannten Bühne sind die Ergebnisse des Programmierens zu betrachten. Hier kann man mit der Spielwelt interagieren. Im unteren Teil dieser Spalte befindet sich das Objektfenster, wo die Objekte und sonstigen Elemente eines Programms angezeigt werden. Markiert man an dieser Stelle ein bestimmtes Objekt, kann man speziell für dieses Code generieren.

## Das Spiel kreieren

Um mithilfe von Scratch Programmcode zu schreiben, müssen wir Blöcke mit Code-Schnipseln von der linken Spalte in den Programmierbereich in der Mitte des Bildschirms ziehen, und zwar für jedes Objekt und jeden Hintergrund.

In diesem Tutorial wollen wir ein Quiz-Spiel programmieren. Dazu brauchen wir zwei Quizmaster, wir nennen sie Matt und Neil. Das Ziel des Spiels ist es, 3 Punkte zu erreichen. Man muss eine Frage korrekt beantworten, um zur nächsten Runde zu gelangen. Wer dreimal falsch geantwortet hat, scheidet aus. Jedes unserer Objekte bekommt eigene Skripte, die ausgeführt werden, wenn sie von bestimmten Ereignissen (vom Anklicken einer Flagge bis hin zu Antworten eines anderen Objekts) ausgelöst werden.

Wir haben unsere Quizmaster Neil (Objekt6) und Matt (Objekt7) getauft. Ferner gibt es ein Logo für das Spiel (Objekt5) sowie ein Game-Over-Zeichen (Objekt8). Mit dem Logo fangen wir an. Es soll sich folgendermaßen verhalten (siehe auch Abbildung 1.0):

- Wenn grüne Flagge angeklickt wird.
- Verstecken.
- 2 Sekunden warten.
- Zeigen.
- Größe verdoppeln.
- 10-mal wiederholen, dann 15 % verkleinern.

› **Zahlen (Grün)** Die Zahlen oder Operatoren erlauben die Verwendung mathematischer Logik in einem Programm.

› **Malstift (Dunkelgrün)** Hiermit kann man auf dem Bildschirm zeichnen.

› **Variablen (Dunkelorange)** Diese Programmierblöcke erzeugen und manipulieren Behälter, welche Daten innerhalb des Programms speichern.

Dank der Unterteilung der Elemente der Programmiersprache in die verschiedenfarbigen Blöcke findet man schnell den Code-Schnipsel, den man gerade braucht. Kinder orientieren

Weiter geht es mit Matt. Er bekommt fünf Skripte, die jeweils auf ein bestimmtes – von Neil oder der Bühne kommendes – Ereignis hin anspringen. Ein solches Ereignis wird als Nachricht (Broadcast) bezeichnet. Die fünf Skripte sind:

› **player\_name** Die Bühne sendet eine Nachricht, sobald der Spieler seinen Namen eingegeben hat.

› **support** Neil sendet eine Nachricht an Matt, wenn der Spieler eine Frage korrekt beantwortet hat.

› **insult** Neil sendet eine Nachricht an Matt, um den Spieler zu verspotten.

› **score** Neil sendet eine Nachricht an Matt, die Matt dazu bringt, dem Spieler seinen Punktestand mitzuteilen.

› **game\_over** Neil sendet eine Nachricht an Matt, um das Ende des Spiels anzuzeigen.

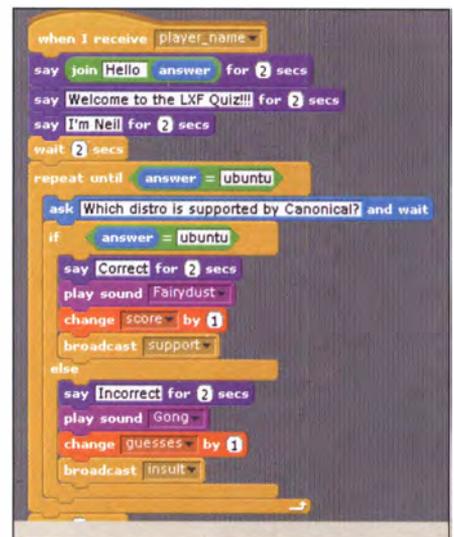
Wie in Abbildung 1.1 zu erkennen, löst jede dieser Nachrichten eine bestimmte Code-Sequenz aus. Matt hat außerdem ein Skript, das anspringt, wenn die grüne Flagge geklickt wird. In Abbildung 1.2 sehen Sie den Code, der alle Spezialeffekte, die auf ein bestimmtes Objekt angewandt wurden, löscht.

Als Nächstes wenden wir uns Neil zu. Sein Objekt hat weit mehr Code als Matt, da er die Hauptfigur unseres Spiels ist. Zunächst gibt es das Ereignis mit der grünen Flagge. Es bringt Neil dazu, alle Spezialeffekte zurückzusetzen, die momentan angewandt werden, und dann eine 0 in der Variablen guesses zu speichern (siehe Abbildung 1.3). Der zweite Teil bildet die Schleife, die unser Spiel steuert. Diese große Schleife wird gestartet, sobald wir den Namen des Spielers empfangen, wofür wiederum der Code zuständig ist, der der Bühne zugeordnet ist (siehe unten).

Wir haben Matts Schleife in drei Teile unterteilt. Zu Beginn des Spiels gibt es ein Code-Segment, das mit der Bühne assoziiert ist und nach dem Namen des Spielers fragt. Dieses speichert den Namen in einer Variablen namens answer, die bei den Fühlen-Blöcken zu finden ist. Sobald der Spieler seinen Namen eingibt, sendet das Skript, das zur Bühne gehört, eine Nachricht namens player\_name. Neils Code wartet auf



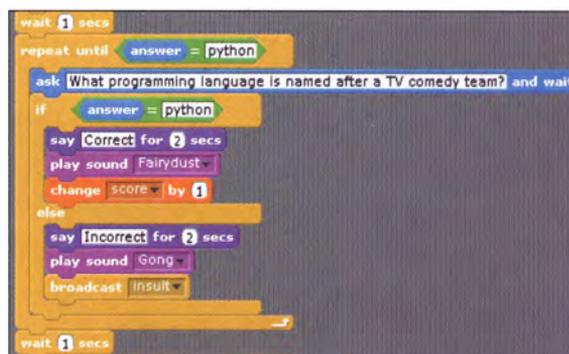
› **Abbildung 1.3:** Dieses Code-Segment setzt die Spezialeffekte und den Wert der Variablen guesses zurück.



› **Abbildung 1.4:** Hier sehen Sie den ersten Teil des Haupt-Codes des Spiels.

## Quick-Tipp

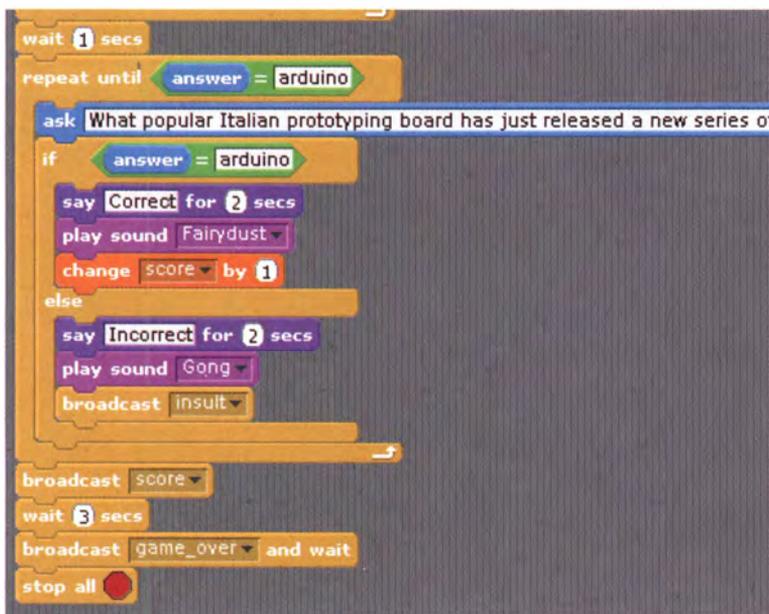
Mit einem Rechtsklick auf einen Code-Block können Sie diesen im Handumdrehen duplizieren oder löschen.



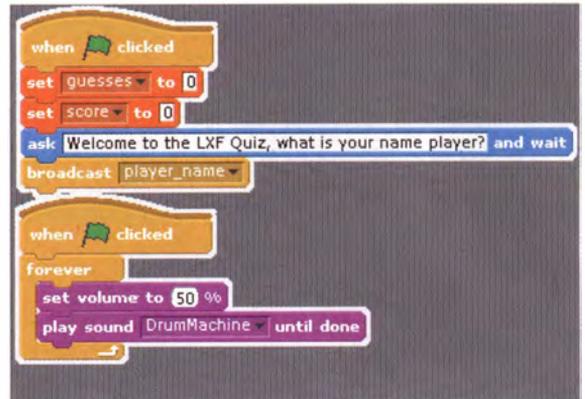
› **Abbildung 1.5:** Eine Schleife innerhalb einer anderen Schleife.

- » den Empfang dieser Nachricht als Auslöser eines neuen Skripts, nämlich den Spieler namentlich zu begrüßen.
- Danach startet die Hauptschleife, die die erste Frage steuert. Wir packen die Frage in eine Schleife ein, die so lange durchlaufen wird, bis die korrekte Antwort gegeben wird, der entsprechende Code lautet **repeat until answer = ubuntu**. Wir stellen dann die nächste Frage und fahren eine zweite Schleife innerhalb der ersten (siehe Abbildung 1.5). Die zweite Schleife besteht aus einer **if**-Anweisung: Falls eine bestimmte Bedingung erfüllt wird, wird eine Code-Sequenz gestartet, ansonsten wird eine **else**-Anweisung befolgt. Ist die Antwort richtig, sagt Neil dies und es ertönt ein Klang als Belohnung. Dann erhöht Neil die Variable score um 1 Punkt und sendet schließlich eine Nachricht an Matt, der daraufhin etwas Nettes sagt. Ist die Antwort jedoch falsch, sagt Neil auch dies, lässt einen Gong erklingen und erhöht den Wert der Variablen guesses um 1. Danach sendet er noch eine entsprechende Nachricht an Matt, welcher den Spieler ein bisschen verspottet.

» **Abbildung 1.6:** Hier konzentrieren wir uns auf die letzten vier Code-Blöcke.



Teil zwei des Programms geht genauso wie die Hauptschleife von Teil eins; wir haben den Code dupliziert und nur die Fragen und Antworten geändert. Bei Scratch dupliziert man Code im Handumdrehen, indem man auf die gewünschten Code-Blöcke rechtsklickt und die entsprechende Option auswählt.



» **Abbildung 1.7:** Sogar die Bühne kann Skripte enthalten.

Der Anfang des dritten Teils entspricht den Hauptschleifen von Teil eins und zwei. Gehen wir darum direkt weiter zu den letzten vier Code-Blöcken (siehe Abbildung 1.6). Der viertelste Block ist eine Nachricht, die Matt den Punktestand, also den Wert der Variablen score, mitteilt. Dies dient als Auslöser dafür, dass Matt den Punktestand nennt. Es folgt eine Pause von 3 Sekunden, damit Matt ausreden kann. Danach senden wir eine weitere Nachricht an Matt, der als Reaktion darauf den Schluss des Spiel-Codes durchgeht, der mit dieser Nachricht verbunden ist. Zuletzt beenden wir alle laufenden Skripte mit dem „stop all“-Block.

Das Ereignis mit der grünen Flagge ist der Code, der die Anzahl der Rateversuche, die ein Spieler hat, kontrolliert. Wir wenden eine Bedingung an mit dem Effekt, dass die Nachricht game\_over an Matt gesendet wird, sobald die Anzahl der Versuche 3 erreicht, woraufhin dieser das Skript für das Spielende auslöst.

Das Objekt des Game-Over-Zeichens ist mit zwei Skripten verbunden. Eins versteckt lediglich das Objekt, sobald die grüne Flagge angeklickt wird, das andere wird durch die Nachricht game\_over ausgelöst. Dadurch wird das Objekt enthüllt und seine Größe auf 100 % gesetzt. Mithilfe einer 10-mal durchlaufenden Schleife erzeugen wir einen Rotations- und Zoom-Effekt wie bei klassischen 8-Bit-Games der 1980er Jahre.

## Die Bühne

Die Bühne ist nicht nur das Zuhause unserer Objekte, sondern kann – wie oben bereits angedeutet – auch selber Skripte enthalten (siehe Abbildung 1.7). Bei unserem Spiel

## Programmierkonzepte

Mit Scratch herumzuspielen, macht natürlich Spaß, aber fast wie nebenbei lernt man damit auch das Programmieren. Egal welche Programmiersprache Sie verwenden, die zugrunde liegenden Konzepte bilden ein solides Fundament und können bei jedem Coding-Projekt wieder angewandt werden. Die wesentlichen Konzepte sind:

- » **Abfolge** Eine Liste von Anweisungen, die in einer bestimmten Reihenfolge durchgeführt werden müssen.
- » **Schleife** Die Wiederholung einer Abfolge. Kann unendlich ausgeführt werden (**while**

**True**) oder mit einer **for**-Anweisung kontrolliert werden (**for x in range(0,3)**). In unserem Spiel haben wir viele Schleifen verwendet.

» **Parallelisierung** Mehrere Abfolgen gleichzeitig ausführen. In unserem Spiel ist die Parallelisierung ein wichtiges Konzept, da jedes Objekt seinen eigenen Code besitzt, der aber parallel zu dem der anderen läuft.

» **Ereignis** Ein Ereignis ist ein Auslöser, der eine Abfolge startet. Das offensichtlichste Ereignis in unserem Spiel ist das Anklicken der grünen Flagge, um das Spiel zu starten.

» **Daten** Wir verwenden eine Variable, um den

Spielstand zu speichern und später können wir die Variable wieder auslesen und verändern.

» **Operator** Die grundlegenden mathematischen Regeln, die man in der Schule lernt. Operatoren können auf Text und Zahlen angewandt werden, um Berechnungen durchzuführen und Daten zu verarbeiten.

» **Bedingung** Bedingungen bilden die Basis unserer Programmlogik und dienen dem Vergleich von vorliegenden Daten mit den Eingaben des Spielers. Mit einer Bedingung kann eine Abfolge davon abhängig gemacht werden, ob ein gewisser Umstand vorliegt.

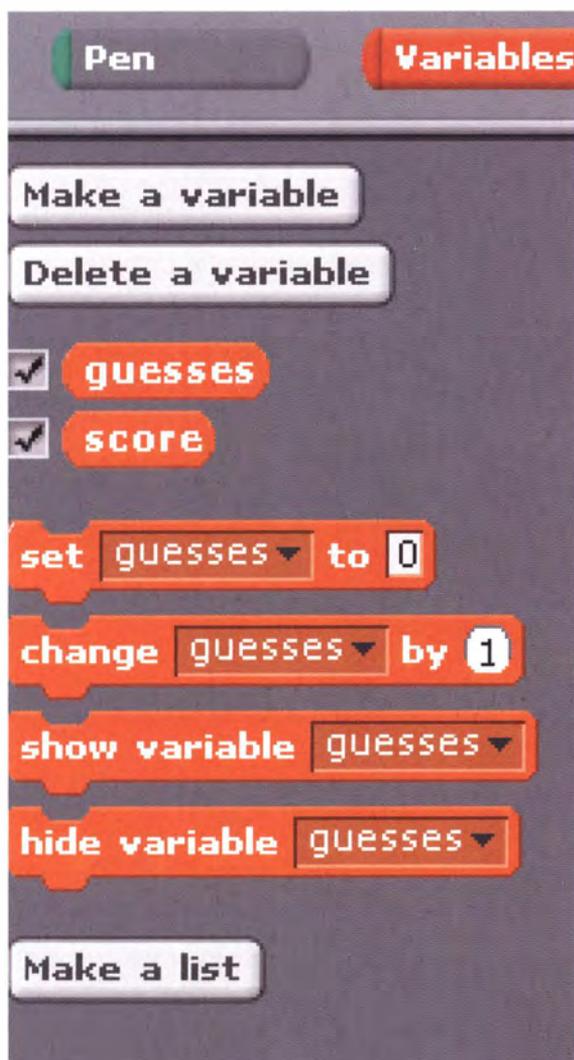
haben wir der Bühne zwei Code-Segmente verpasst, die jeweils durch das Anklicken der grünen Fahne ausgelöst werden. Das erste Segment setzt die Werte der Variablen `guesses` und `score` zurück und sendet den Spielernamen, das zweite Segment ist eine unendliche Schleife, die den Klang `DrumMachine` kontinuierlich abspielt und die Lautstärke auf 50 % setzt.

Noch ein Wort zu den Variablen: Diese müssen zunächst erstellt werden, dies geht an der entsprechenden Stelle bei den Code-Blöcken (siehe Abbildung 1.8). In Abbildung 1.9 sehen Sie das Menü zur Namensgebung und Objektzuordnung der Variablen. In unserem Spiel benutzen wir zwei Variablen: `score` und `guesses`. Nachdem wir diese generiert haben, können wir sie sofort im Programmierbereich verwenden.

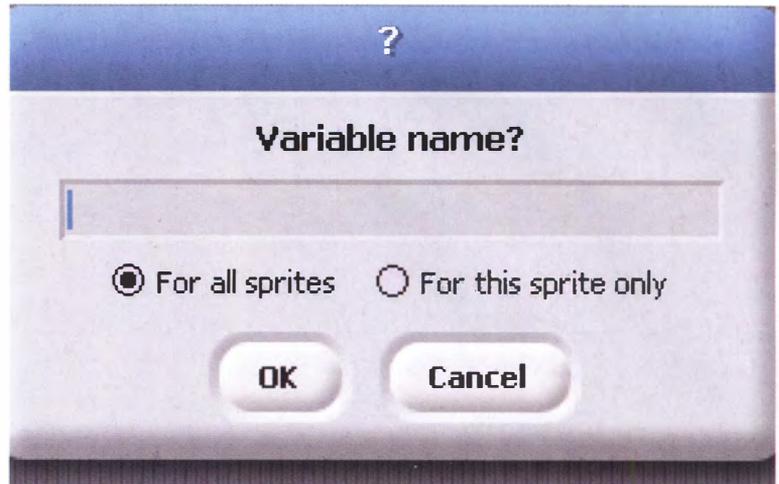
## Pseudo-Code

Um uns die Logik hinter unserem Spiel zu vergegenwärtigen, können wir einen Pseudo-Code schreiben. Wie Pseudo-Code funktioniert, möchten wir hier an einem kleinen Beispiel demonstrieren (wobei wir den Text im Hinblick auf die bessere Vergleichbarkeit mit der Python-Syntax im Englischen belassen):

```
a has the value of 0
while a is less than 10:
```



› Abbildung 1.8: Für unser Spiel müssen wir zwei Variablen erstellen: `guesses` und `score`.



```
print on the screen the value of a
increment a by 1
```

Wie übertragen wir nun den Pseudo-Code in eine Programmiersprache? Versuchen wir es zuerst mit Scratch und danach mit Python. Der Scratch-Code würde lauten:

```
[When Green Flag is clicked]
Set variable a to 0
forever if a < 10
  say a for 2 secs
  change a by 1
```

Dies gibt der Variablen `a` den Wert 0. Solange der Wert nicht 10 erreicht, wird eine bedingte Schleife ausgeführt, im Rahmen derer Scratch den Wert von `a` jeweils 2 Sekunden lang ausgibt und ihn dann um 1 erhöht. Bei dem Wert 9 stoppt die Schleife. Der Python-Code sähe so aus:

```
a = 0
while a < 10:
  print a
  a = a + 1
```

Sie sehen, dass auf der logischen Ebene kein großer Unterschied zwischen den beiden Programmiersprachen besteht. Sie können sich folglich auch anhand von Scratch die Grundprinzipien des Programmierens erarbeiten.

## Das Spiel testen

Geschafft! Nun sollten Sie das Spiel ausprobieren. Der Ablauf ist grundsätzlich wie folgt:

- Die grüne Flagge anklicken.
- Spieler wird nach Namen gefragt.
- Matt und Neil begrüßen den Spieler.
- Matt und Neil eröffnen das Quiz.
- Matt fordert Neil auf, die erste Frage zu stellen.
- Neil stellt eine Frage.
- Ein Kästchen zum Eintippen der Antwort erscheint.
- Falls korrekt, stellt Neil dies fest, Matt sagt etwas Nettes, die Punktzahl erhöht sich um 1, es geht weiter zur nächsten Frage.
- Falls inkorrekt, stellt Neil dies fest, Matt spottet, die Anzahl der `guesses` erhöht sich um 1, der Spieler hat noch 2 Versuche.
- Sind 3 Fragen korrekt beantwortet, nennt Neil die Punkte und beendet das Spiel, alle Skripte werden beendet.
- Erreicht die Anzahl `guesses` während des Spiels 3, endet das Spiel mit dem Game-Over-Zeichen.

Doch das war hoffentlich nur der Anfang Ihrer Abenteuer mit Scratch! Auf den folgenden Seiten geht es mit einem neuen spannenden Projekt weiter.

› Abbildung 1.9: Das Menü zur Namensgebung und Objektzuordnung der Variablen.

## Quick-Tipp

Sie können Ihre Scratch-Projekte online mit anderen teilen. Besuchen Sie dazu die Website <http://scratch.mit.edu>.

# Scratch: Arcade

Hier lernen Sie, wie Sie mit Scratch und dem Raspberry Pi Ihr eigenes spannendes Arcade-Game bauen.

**N**achdem Sie auf den vorhergehenden Seiten ein Quiz-Spiel zusammengestellt haben, wollen wir uns jetzt an eine etwas größere Herausforderung wagen: ein spannendes Shoot-'em-Up-Spiel, also ein Arcade-Game, bei dem Sie es allein mit vielen Gegnern aufnehmen müssen. Glücklicherweise ist das Munitionslager gut gefüllt.

Wir beginnen in einem Stadt-Setting, es ist spät in der Nacht, Sie sind auf Patrouille, als urplötzlich eine Schar bössartiger Invasoren am Himmel erscheint und zum Angriff übergeht. Ihre Aufgabe ist es, die Gegner unschädlich zu machen, was am schnellsten und wirkungsvollsten funktioniert, indem Sie sie abschießen. Es handelt sich ja schließlich um ein Shoot 'em Up. Für jeden zerstörten Invasoren erhalten Sie Punkte, doch Sie haben nur ganze 30 Sekunden Zeit für die Abwehrschlacht.

In diesem Tutorial beziehen wir uns auf die neueste stabile Version von Scratch, die uns vorlag, und das war Version 1.4. Version 2 ist heute, im Winter 2014, noch in der Beta-Phase, daher haben wir es nicht berücksichtigt.

Scratch basiert auf der Verwendung verschiedenfarbiger Blöcke, welche unterschiedliche Funktionalitäten repräsentieren. Folgende Gruppen gibt es:

» **Bewegung (Dunkelblau)** Diese Blöcke ermöglichen es Ihnen, Objekte in Ihrem Spiel zu bewegen und zu kontrollieren.

» **Steuerung (Orange)** Hier geht es um logische Mechanismen, die das Programm steuern (Schleifen und Anweisungen), und Ereignisse (wie etwa das Drücken einer Taste), die bestimmte Aktionen auslösen. Bei Scratch 2.0 bekommen Ereignisse eine eigene Gruppe.

» **Aussehen (Lila)** Diese Programmierblöcke können Farbe, Größe und Erscheinungsform (Kostüm) von Objekten verändern und führen außerdem interaktive Elemente wie beispielsweise Sprechblasen ein.

» **Fühlen (Hellblau)** Mit Fühlen ist die Wahrnehmung von Input gemeint, der für Ihr Programm notwendig ist, etwa Tastenanschläge, die Position eines Objekts auf dem Bildschirm oder Kollisionen von Objekten.

» **Klang (Pink)** Diese Blöcke fügen dem Programm Musik und Geräusche hinzu.

» **Zahlen (Grün)** Die Zahlen oder Operatoren erlauben die Verwendung mathematischer Logik in einem Programm.

» **Malstift (Dunkelgrün)** Hiermit kann man auf dem Bildschirm zeichnen.

» **Variablen (Dunkelorange)** Diese Programmierblöcke erzeugen und manipulieren Behälter, welche Daten innerhalb des Programms speichern.

Dadurch, dass die Programmiersprache mit farblich kodierten Code-Blöcken arbeitet, ist es sehr einfach, die Funktionalität eines bestimmten Blocks einzuordnen.

## Die Entwicklungsumgebung von Scratch

**Blockbereich**  
Hier sind die farblich voneinander abgegrenzten Befehlskategorien angeordnet.

**Programmierbereich**  
Im mittleren Teil des Fensters setzen Sie die Code-Elemente zu einem Skript zusammen.

**Bühne**  
Hier werden die Ergebnisse des Programmierens angezeigt und findet die Interaktion statt.

**Objektliste**  
Hier werden die Objekte angezeigt, die in Ihrem Programm vorkommen. Durch Anklicken wählen Sie ein Objekt aus.

Besonders Kinder kommen auf Anhieb sehr gut mit dem Farbsystem zurecht und lernen dann nach und nach spielerisch die Kombinationsmöglichkeiten der Blöcke kennen.

Für unser Arcade-Game benötigen wir die folgenden Zutaten:

- » Drei Invasoren-Objekte
- » Ein Game-over-Objekt
- » Ein Hintergrundbild für die Stadt

Alle Dateien, die Sie für das Shoot-'em-up-Spiel benötigen, können Sie unter dem Link <http://bit.ly/lxfscratcharcade> gesammelt herunterladen.

Wir beginnen mit den Angreifern. Zur Erinnerung: Sie haben nur 30 Sekunden Zeit, um sie alle abzuwehren und die Stadt zu retten. Die Invasoren-Objekte sind quasi unsere Zielscheiben, und wenn es gelingt, auf eins draufzuklicken, verschwindet es und der Punktestand des Spielers steigt.

Zuerst müssen Sie eine neue Scratch-Session starten, indem Sie einfach das Programm öffnen. Es sollte daraufhin ein neues Scratch-Fenster erscheinen, mit der Cartoon-Katze in der Bühnenabteilung. Da wir die Katze bei unserem Arcade-Game nicht brauchen, rechtsklicken Sie auf das Objekt und löschen Sie es. Sie haben nun eine leere Bühne vor sich und können das Invasoren-Objekt importieren. Sie sehen zwischen Bühne und Objektliste drei Icons. Das linke besteht aus einem Stern mit einem Pinsel davor und wird gewählt, wenn man selber ein neues Objekt zeichnen möchte. Das mittlere zeigt einen Stern mit einem Ordner davor; klicken Sie auf dieses Icon, um das neue Objekt zu importieren. Mit dem Icon ganz rechts kann man ein Zufallsobjekt importieren, aber diese Funktion brauchen wir jetzt nicht.

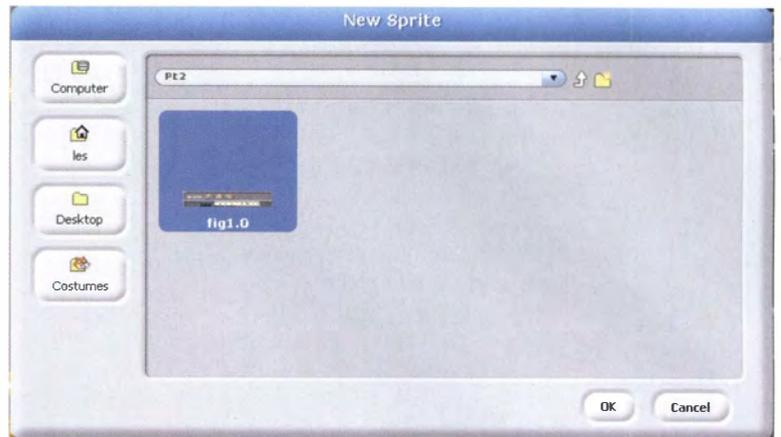
## Interaktive Objekte

Wenn Sie nun also das Icon zum Importieren von Objekten anklicken, öffnet sich eine Dialogbox (siehe Abbildung 1). Navigieren Sie mithilfe der Ordner-Adressleiste oben in der Box zu dem Ort, an dem Sie die heruntergeladenen Dateien gespeichert haben. Wählen Sie das Objekt für die Invasoren aus und klicken Sie auf OK. Damit wird das Objekt importiert und erscheint sogleich auf der Bühne in der Scratch-Entwicklungsumgebung.

Nun müssen wir dem Objekt einen Programmcode zuweisen, um es interaktiv zu machen. Es gibt drei Code-Segmente für das Invasoren-Objekt. In Pseudo-Code (siehe Seite 79 in diesem Heft) ausgedrückt, funktioniert das Ganze folgendermaßen:

- Wenn grüne Flagge angeklickt wird.
- Wert der Variablen speed auf 5 setzen.
- Innerhalb einer unendlichen Schleife:
  - Objekt bewegen, Variable speed gibt Schrittzahl vor.
  - Falls Objekt den Rand des Bildschirms erreicht.
  - Objekt vom Bildschirm abprallen lassen.

Der Code, mit dem die Invasoren bewegt werden, ist ziemlich



simpel, aber das ist der erste Schritt, mit dem wir das Spiel zu einer größeren Herausforderung machen. Um diesen Code hinzuzufügen, müssen Sie zuerst eine Variable namens Speed generieren. Gehen Sie dazu in die Abteilung Variablen im Blockbereich. Wählen Sie dort die Option „Variable erstellen“, benennen Sie die neue Variable wie beschrieben und geben Sie „Nur für dieses Objekt“ an.

## Pseudo-Code für Objekte

Mit der neuen Variablen haben wir nun einen Ort, an dem die Geschwindigkeit des Angreifers gespeichert werden kann. Wir brauchen diese Angaben also nicht hart zu kodieren, sondern setzen die Geschwindigkeit einmal und beziehen uns dann jeweils auf die Variable. Dies spart uns viel Programmieraufwand.

Als Nächstes vervielfältigen wir den Code. Wenn Sie das Code-Segment fertiggestellt haben, speichern Sie das Projekt und klicken dann auf die grüne Flagge, um den Code ablaufen zu lassen. Sie sollten nun das Invasoren-Objekt auf dem Bildschirm sehen, wie es sich horizontal fortbewegt und vom Bildschirmrand abprallt, wenn es ihn erreicht. Aber was ist das? Das Objekt dreht sich, indem es an der Vertikalachse gespiegelt wird, wenn es vom Rand abprallt. Um dies zu verhindern, müssen wir bei den Objekteinstellungen eine Änderung vornehmen. In Abbildung 3 sehen Sie, wo Sie diese finden: Es handelt sich um drei Icons oben im Programmierbereich, links über dem Reiter Skripte. Die Icons haben folgende Funktion:

- » Oberes Icon 360-Grad-Drehung
- » Mittleres Icon Links-rechts (Spiegelung)
- » Unteres Icon Nicht drehen

Da wir nicht möchten, dass das Objekt gespiegelt wird, wählen wir das Icon unten, welches die Drehung verhindert. Überprüfen Sie durch einen Neustart des Codes, ob es geklappt hat.

Die Invasoren sind gewieft und haben eine Tarnvorrichtung entwickelt, dank derer sie sich vor den Stadtbeschützern

» **Abbildung 1:** Dies ist der Dialog zum Importieren von Objekten.

## Quick-Tipp

Wenn Sie auf ein Objekt rechtsklicken, erscheinen mehrere Optionen, darunter Löschen und Duplizieren. Sehr praktisch!

»

## Scratch auf dem Raspberry Pi verwenden

Auf jedem Raspberry Pi mit dem Raspbian-Betriebssystem ist Scratch bereits vorinstalliert. Wussten Sie schon, dass Sie mithilfe von Scratch sogar die GPIO-Anschlüsse des Pi steuern und somit mit der Umwelt interagieren können?

Ja, in der Tat: Mit dem einfachen Interface von Scratch lassen sich LEDs, Motoren und diverse Eingabegeräte kontrollieren! Falls

Sie das mal ausprobieren möchten, statten Sie der besten Materialquelle für diesen Einsatzzweck – dem Cymplecy-Blog von Simon Walter unter <http://bit.ly/cymplecy> – einen Besuch ab.

Dort finden Sie tolle Artikel darüber, wie Sie diverse Scratch-Experimente dieser Art durchführen können, und der dazugehörige Code wird in einfachen Schritten erklärt.

In einem ganz besonders spannenden Projekt von Simon Walter geht es um Roboter, die einen Irrgarten bewältigen, der zunächst in einer virtuellen Form mittels Scratch programmiert wurde. Sobald die Software innerhalb der virtuellen Umgebung funktioniert, soll sie auf die Roboter übertragen werden, die dann mit Ultraschallsensoren das tatsächliche Gelände erkunden.

» verbergen können. Allerdings ist die Dauer der Tarnung begrenzt. Wir müssen zuschlagen, solange die Angreifer sichtbar sind.

Schauen wir uns den logischen Ablauf wieder mit ein bisschen Pseudo-Code an:

- Wenn grüne Flagge angeklickt wird.
- Innerhalb einer unendlichen Schleife:
  - Invasoren-Objekt verstecken.
  - Eine zufällige Zeitspanne von 1-3 Sekunden warten.
  - Invasoren-Objekt wieder zeigen.
  - Eine zufällige Zeitspanne von 1-5 Sekunden warten.

Die Enthüllung der Angreifer dem Spieler gegenüber ist ein Schlüsselement unseres Spiels. Dank der Zufallszeiten wird es noch ein bisschen schwieriger. Wenn Sie den Pseudo-Code in tatsächlichen Scratch-Code umgewandelt haben, speichern Sie das Projekt und probieren Sie das Programm aus. Verschwindet das Objekt? Falls nicht, stellen Sie sicher, dass Sie den „warte“-Block verwendet und die korrekten Zeitangaben eingefügt haben.

## Quick-Tipp

Falls Ihr Code plötzlich verschwinden zu sein scheint, überprüfen Sie mal, ob Sie auch das richtige Objekt angeklickt haben.

## Wir brauchen Laser!

Das letzte Stückchen Code für dieses Objekt hat zwei Aufgaben: erstens einen coolen Laser-Sound abspielen, zweitens den Wert einer neuen Variablen namens score um 1 zu erhöhen. Wenden wir wieder Pseudo-Code an:

- Wenn Objekt 1 angeklickt wird.
- Objekt verstecken.
- Laser-Sound abspielen.
- Wert von score um 1 erhöhen.

Im Gegensatz zur vorherigen Variablen müssen wir der neuen bei ihrer Erstellung die Option „Für alle Objekte“ zuordnen. Das bedeutet, dass andere Objekte den Wert von score beeinflussen können. Nun brauchen wir noch einen coolen Sound für den Laser. Klicken Sie oben im Programmierbereich auf den Reiter Klänge und dann auf Importieren, dann erscheint eine Dialogbox ähnlich wie in Abbildung 2. Navigieren Sie zum Ordner Electronic und suchen Sie den Klang Laser1 heraus. Diesen können Sie in den Klänge-Reiter importieren.

Die Grundlagen sind gelegt, nun können Sie versuchen, den obigen Pseudo-Code in tatsächlichen Scratch-Code umzusetzen. Wenn Sie es geschafft haben, speichern Sie das Projekt und klicken Sie auf die grüne Flagge. Funktioniert alles? Dann einen herzlichen Glückwunsch, das war nun der komplette Code für das Invasoren-Objekt.

Wie im vorhergehenden Scratch-Projekt hat auch bei unserem Shoot-'em-up-Spiel die Bühne ein paar wichtige Aufgaben zu erfüllen: erstens, den Wert der Variablen score bei jedem neuen Spiel auf null zu setzen; zweitens, den 30-Sekunden-Timer zu kontrollieren.

- Wieder beginnen wir zunächst mit etwas Pseudo-Code:
- Wenn grüne Flagge angeklickt wird.
  - Wert von score auf 0 setzen.
  - Wert von timer auf 30 setzen.



» Abbildung 2: Die Dialogbox für die Klänge entspricht ungefähr der Objekt-Dialogbox. Hier können Sie Sounds importieren.

» Abbildung 3: Mit den kleinen Icons links können Sie den Drehtyp eines Objekts festlegen. Allerdings wird diese Funktionalität bei Scratch 2.0 anderswo angesiedelt sein.



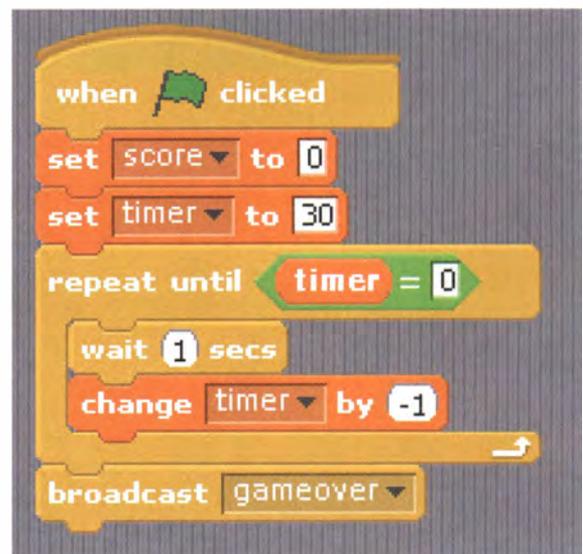
- Bedingte Schleife ausführen, bis timer 0 erreicht.
- 1 Sekunde warten.
- Bei jedem Durchlauf der Schleife Wert von timer um 1 vermindern.
- Ende der Schleife.
- Nachricht Game over versenden.

Sie sehen, wir müssen ein weiteres Mal eine Variable erstellen, diesmal unter dem Namen timer. Auch diese Variable bekommt die Option „Für alle Objekte“ zugeordnet. Anschließend werden die Werte der Variablen score und timer auf 0 beziehungsweise 30 gesetzt.

Dank dieses Code-Segments wird das Spiel für jeden neuen Spieler wieder in den Anfangszustand zurückversetzt: Man hat 30 Sekunden Zeit für die Abwehrschlacht und noch keine Punkte.

Unsere Schleife ist an eine Bedingung geknüpft (siehe Abbildung 4): Sie wird nur so lange durchlaufen, wie der Wert von timer größer als null ist. Damit das funktioniert, müssen wir die Schleife anweisen, jeweils 1 Sekunde zu warten und dann 1 vom timer-Wert abzuziehen. Ohne die sekundenlange Pause würde die Schleife natürlich in einer sehr kurzen Zeit 30 Mal durchlaufen.

Zuletzt verwenden wir eine Nachricht, um die Bühne und das Game-over-Objekt miteinander kommunizieren zu



» Abbildung 4: Hier handelt es sich um eine Schleife, die an eine Bedingung geknüpft ist.

lassen, aber dazu kommen wir noch ausführlicher.

Setzen Sie den Pseudo-Code in Scratch-Code um, speichern Sie das Projekt und klicken Sie für einen Testlauf auf die grüne Flagge. Sie haben nun einen Timer und eine Methode, das Spiel für einen neuen Spieler zurückzusetzen, integriert.

## Game over

Unser letztes Objekt hat hauptsächlich zwei Aufgaben: Es soll dem Spieler am Ende des Spiels seinen Punktestand mitteilen und es soll dann auch alle noch laufenden Skripte abbrechen. Das Objekt hat außerdem noch ein kleines separates Skript, welches lediglich das Objekt selbst verbirgt, sobald die grüne Flagge angeklickt wird. Das Game-over-Objekt erscheint erst am Ende des Spiels wieder auf der Bildfläche.

**Hier abermals der Pseudo-Code für unser Vorhaben:**

Wenn ich die Nachricht Game over erhalte.

Zeige das Game-over-Objekt.

Sage „Dein Punktestand beträgt“ und füge den Wert der Variablen score an den Satz an.

Beende alle noch laufenden Skripte des Spiels.

Unser nächstes Code-Segment ist eine Mischung von Befehlen aus den Bereichen Aussehen, Variablen und Zahlen. Wir benutzen den Aussehen-Block „Sage Hallo für 2 Sekunden“, ersetzen das „Hallo“ durch „Dein Punktestand beträgt“ und fügen noch die Variable score hinzu, um den entsprechenden Text am Bildschirm auszugeben.

Als Nächstes brauchen wir noch das kleinere Code-Segment, welches dafür sorgt, dass das Game-over-Objekt beim Anklicken der grünen Flagge verborgen wird. Speichern Sie danach wieder das Projekt und machen Sie einen Testlauf. Wir hoffen, dass nun alles gut funktioniert.

Was jetzt noch fehlt, ist lediglich ein Hintergrundbild für das Stadt-Setting. Den Hintergrund der Bühne können Sie ändern, indem Sie auf das Bühnen-Icon neben der Objektliste klicken. Gehen Sie von dort zurück zum



› **Abbildung 5:** Der erste Teil des Codes für das Game-over-Objekt verbirgt das Objekt beim Spielstart. sequence.



› **Abbildung 6:** Der Reiter Hintergrunde (Backgrounds) kann viele verschiedene Hintergrundbilder enthalten.

Programmierbereich: Der mittlere der drei Reiter oben in der Spalte ist nun für die Hintergründe zuständig (siehe Abbildung 6). Dort können Sie eine Auswahl aus den vorhandenen Bildern treffen oder einen eigenen Background importieren. Wir wollen Ersteres tun.

Klicken Sie innerhalb des Reiters auf Importieren. Es öffnet sich daraufhin eine Dialogbox. Suchen Sie den Ordner Draußen (Outdoors) und wählen Sie das Hintergrundbild mit der Stadtansicht bei Nacht aus.

Damit ist das Shoot-'em-up-Spiel fertig! Speichern Sie es und klicken Sie auf die grüne Flagge. Viel Spaß!

## Was kommt als Nächstes?

Wir könnten nun noch weitere Objekte kreieren und sie nach derselben Logik programmieren wie zuvor, aber das wäre wenig erbaulich und würde lange dauern. Stattdessen können Sie einfach einen Rechtsklick auf dem Objekt ausführen, das Sie vervielfältigen möchten, und dann die Option „Duplizieren“ ausführen. Damit wird eine Kopie des Objekts komplett mit allen assoziierten Skripten erstellt.

Wenn Sie dann eine Schar Objekte generiert haben, könnten Sie jedem individuelle Eigenschaften verleihen oder mit ihm bestimmte Ideen ausprobieren. Sie könnten einzelne Objekte schneller oder kleiner machen, dafür aber mehr Punkte vergeben, wenn bei diesen schwierigeren Zielen ein Abschluss gelingt.

Probieren Sie ein bisschen herum, sehen Sie sich die Möglichkeiten der Code-Blöcke an. Sie können die Code-Segmente auch leicht neu gruppieren und verbinden.

Wir hoffen, Scratch gefällt Ihnen und Sie werden noch eine Weile damit herumprobieren, bevor Sie irgendwann zu Python oder einer anderen Sprache übergehen. Oder auch länger – falls Sie womöglich ein paar Roboter mit Scratch steuern möchten.

## Quick-Tipp

Scheuen Sie sich nicht davor, Ihren Code wieder auseinanderzunehmen, besonders wenn Sie auf der Fehlersuche sind. Falls irgendetwas nicht so funktioniert wie erwartet, teilen Sie den Code in mehrere kleinere Segmente auf, damit Sie eine bessere Übersicht bekommen.

## Online-Ressourcen rund um Scratch

Scratch ist lediglich eine von vielen Entwicklungen des MIT – das Institut hat noch einige andere großartige Tools im Programm, etwa den App Inventor (<http://appinventor.mit.edu/explore>). Dieses Werkzeug bedient sich einer ähnlichen Blockstruktur wie Scratch, um Programmierer in die Lage zu versetzen, cool aussehende und funktionale Android-Apps zu

entwickeln, ohne sich mit Java auskennen zu müssen. Der App Inventor wird wie Scratch ebenfalls in Schulen eingesetzt, um Schul-Apps zu programmieren.

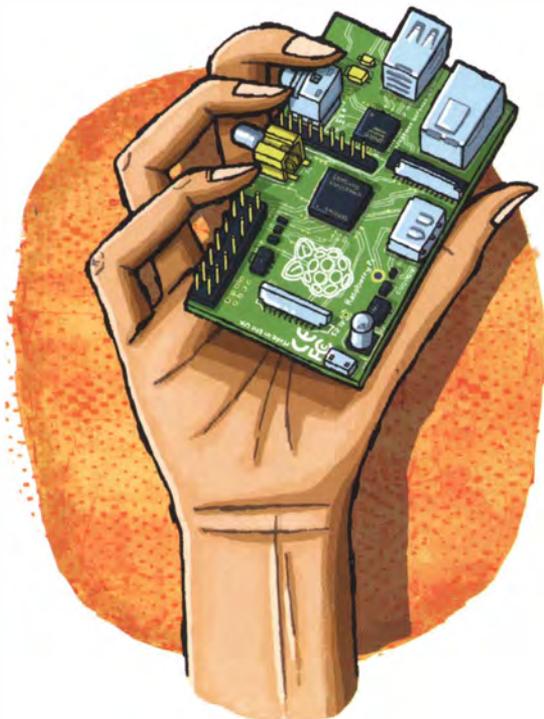
Ein anderes schönes Tool ist Blockly (<http://bit.ly/1IACCD5>), ein visueller Programmier-Editor. Es handelt sich hier nicht um ein MIT-Produkt, aber es ist auf jeden Fall

empfehlenswert. Auch Blockly basiert, wie sein Name andeutet, auf einer Blockstruktur beim Coding. Man kann damit auch Programme für andere Geräte schreiben und diesen exportieren. Es gibt daneben ein webbasiertes Blockly-Demo (<http://bit.ly/JHkXvX>), mit dem Sie ausprobieren können, wie man simultan Code für JavaScript, Python und XML schreibt.

# Scratch: Gejagt

Programmieren Sie mit Scratch in 16 Schritten ein einfaches Katz-und-Maus-Spiel für den Raspberry Pi.

**D**er Raspberry Pi ist mit einem breiten Spektrum an Programmiersprachen kompatibel. Eine von diesen heißt Scratch. Scratch ist eine visuelle Programmiersprache und wurde speziell für Computeranfänger, also insbesondere Kinder und Jugendliche, entwickelt – diese Sprache führt den Benutzer in viele grundlegende Mechanismen des Programmierens ein und ist dabei gleichzeitig sehr einfach zu verstehen.



Bei Scratch wird der Schwerpunkt auf das Programmieren von grafischen Programmen wie Spielen gelegt. Falls Sie sich noch nie mit irgendeiner Programmiersprache beschäftigt haben, ist Scratch ein prima Einstieg für Sie. Vor allem macht das Arbeiten damit Spaß, also nichts wie los!

Wenn Raspbian das Betriebssystem Ihres Raspberry Pi ist, dann finden Sie das Scratch-Entwicklungswerkzeug bereits auf der Desktop-Umgebung. Sie brauchen es also nicht gesondert zu installieren. Klicken Sie einfach auf das entsprechende Icon (den orangefarbenen Katzenkopf) auf der Benutzeroberfläche.

Falls Sie ein anderes Betriebssystem auf dem Raspberry Pi benutzen und Scratch erst noch installieren müssen oder aber Scratch auf einem anderen Rechner, der mit Linux, Windows oder OS X läuft, ausprobieren möchten, konsultieren Sie bitte die offizielle Informationsseite

<http://scratch.mit.edu>.

Das Hauptfenster der Scratch-Entwicklungsumgebung ist in mehrere Sektionen unterteilt. Grob gesagt sind die Programmier-Tools auf der linken Seite, das aktuell entstehende Programm in der Mitte und der Test- und Ausführbereich auf der rechten Seite angeordnet. Jedes Scratch-Programm umfasst eine Anzahl grafischer Objekte (Sprites, Figuren), etwa eine Cartoon-Katze, die wiederum diverse Skripte enthalten. Diese Skripte bestimmen, was passiert, wenn das Programm läuft.

Oben links im Fenster der Entwicklungsumgebung sehen Sie acht Schaltflächen mit den Bezeichnungen **Bewegung**, **Aussehen**, **Klang**, **Malstift**, **Steuerung**, **Fühlen**, **Operatoren** und **Variablen**, die jeweils für eine Kategorie von Befehlen stehen, die Sie per Drag & Drop in die Skript-Abteilung im mittleren Bereich verschieben können, um so nach und nach Ihr Programm aufzubauen. Viel Erfolg!



› Standardmäßig enthält Scratch eine Reihe vorgefertigter Objekte, aus denen Sie auswählen können. Der Katalog der zur Verfügung stehenden Motive lässt sich über die Objektliste oder den Kostüme-Reiter im Programmierbereich aufrufen.

› Jedes Objekt kann mit verschiedenen Kostümen (bildlichen Umsetzungen) ausgestattet werden. Die Kostüme werden im Programmierbereich angezeigt. Sie können dort auch selbst welche hinzufügen.

## Variablen und Messages

Sie werden sich vermutlich früher oder später wünschen, dass Ihr Programm sich Dinge merken möge, etwa einen Text oder eine Zahl. Dazu dienen die sogenannten Variablen – kleine Bereiche im Speicher Ihres Computers, wo Ihr Programm die entsprechenden Daten ablegen kann. In Schritt 3 der Anleitung unten werden wir zwei solche Variablen erstellen. Bei anderen Programmiersprachen können Sie noch ganz andere Arten von Variablen

benutzen, doch das ist an dieser Stelle noch nicht relevant. Wenn Sie eine Variable generiert haben, können Sie Verschiedenes mit ihr machen: sie auf einen bestimmten Wert setzen, sie ausgeben lassen oder die Erfüllung einer Bedingung überprüfen.

Sobald Sie ein paar Skripte entwickelt haben, möchten Sie vielleicht, dass diese miteinander kommunizieren. Dies lässt sich mitunter durch Variablen bewerkstelligen, doch

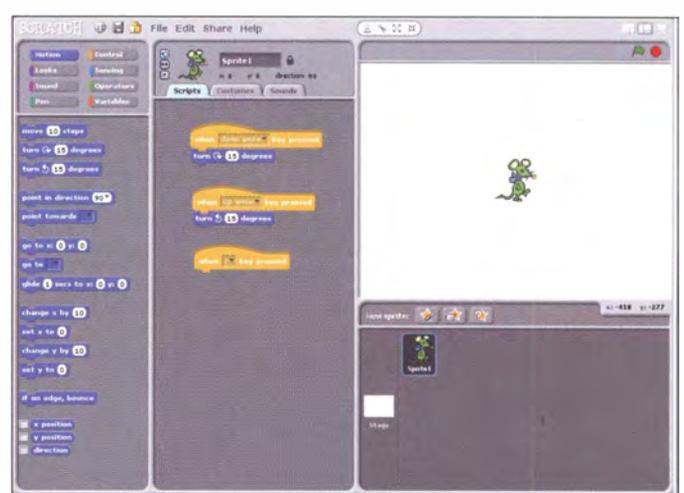
sogenannte Messages sind häufig besser geeignet. Messages können Skripte starten, so wie etwa die Betätigung einer bestimmten Taste einen bestimmten Vorgang auslösen kann. Wenn ein Skript eine bestimmte Message versendet, ruft es damit alle anderen Skripte auf, die mit **Wenn ich <Message> empfangen** beginnen. Wie Variablen haben auch Messages Namen, daher müssen sie zuerst erstellt werden.

## Das Katz-und-Maus-Spiel: Schritt für Schritt



### 1 Hier kommt die Maus

Beim Programmieren des Spiels beginnen wir mit der Maus. Wir geben dazu einfach unserem **Objekt1** ein neues Erscheinungsbild. Gehen Sie im Menü über **Kostüme > Importieren > Tiere** zu dem gewünschten Bild der Maus und wählen Sie es aus. Mithilfe des Schrämpfen-Icons (hier im Bild eingekreist) können Sie die Maus noch verkleinern.



### 2 Tastenzuordnung setzen

Klicken Sie auf den Button **Bewegung**, ziehen Sie die Schaltflächen **drehe dich im Uhrzeigersinn um <15> Grad** und **drehe dich gegen den Uhrzeigersinn um <15> Grad** in den mittleren Bereich und ordnen Sie diese den Tasten **Pfeil nach rechts** und **Pfeil nach links** zu. Gehen Sie dann im Programmierbereich auf den Reiter **Skripte**.



### 3 Variablen erstellen und benennen

Ersetzen Sie in **Skripte** bei der Schaltfläche **Wenn Taste <Pfeil nach rechts> gedrückt** die Tastenzuordnung durch den Buchstaben **r** wie Reset. Damit wird später ein neues Spiel gestartet. Klicken Sie auf den Button **Variablen** bei den Blockbereichen oben links. Erstellen Sie zwei Variablen mit den Namen **score** (Punktstand) und **over** (Ende).



### 4 Punktestand zurücksetzen

Setzen Sie die Schaltflächen **zeige dich** (aus dem Blockbereich **Aussehen**), **gehe zu x: <100> y: <100>** (aus **Bewegung**) sowie **setze <score> auf <0>** und **setze <over> auf <0>** (beide aus **Variablen**) unter das Skript **Wenn Taste <r> gedrückt**. Wie Sie gesehen haben, müssen Sie die Werte für **x** und **y** zuerst anpassen.



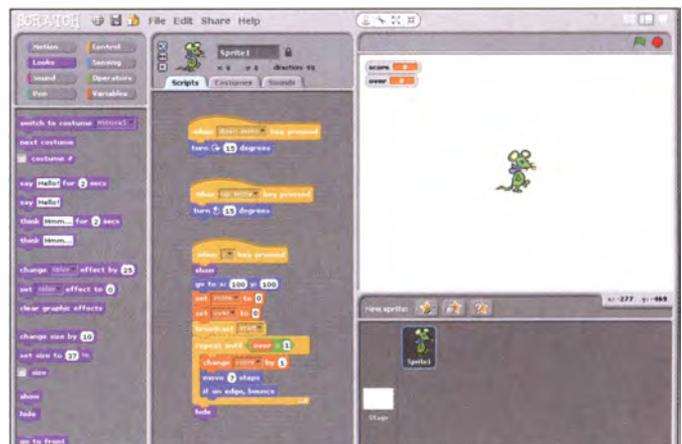
## 5 Sendebefehl hinzufügen

Fügen Sie dem Skript **Wenn Taste <R> gedrückt** noch die Schaltfläche **sende <...>** hinzu und erstellen Sie im Aufklappmenü dieser Schaltfläche eine neue Message mit dem Namen **start**.



## 6 Eine Schleife erstellen

Sie können Schleifen erstellen, die denselben Code mehrfach ausführen, bis eine Bedingung erfüllt ist. Erweitern Sie das Skript mit **wiederhole bis <...>** (aus **Steuerung**) und holen Sie sich eine Gleichung **<...> = <...>** (aus **Operatoren**) dazu. Ziehen Sie dann die Variable **over** in das Feld links vom Gleichheitszeichen und tippen Sie in das rechte Feld eine **1**.



## 7 Die Schleife erweitern

Zu dem Block **wiederhole bis <over> = 1** kommen jetzt noch die Anweisungen **ändere <score> um <1>** (aus **Variablen**), **gehe <7>-er-Schritt** (aus **Bewegung**) und **pralle vom Rand ab** (ebenfalls aus **Bewegung**). Diese drei Anweisungen werden so lange wiederholt, bis **over** auf **1** steht.

## 8 Die Maus verstecken

Wenn die Katze die Maus gefangen hat, endet die **wiederhole**-Schleife und das Programm schreitet weiter. Ziehen Sie die Anweisung **verstecke dich** (aus **Aussehen**) unter die Schleife, dann verschwindet die Maus, sobald sie gefangen wurde.



## 9 Die Katze kreieren

Da das **Objekt1** zur Maus mutiert ist, brauchen wir ein neues **Objekt2**, um die Katze wieder ins Spiel zu bringen. Erstellen Sie das Objekt und gehen Sie bei der Auswahl des Erscheinungsbildes vor wie in Schritt 1. Auch die Katze sollte mithilfe des Schrumpfen-Icons verkleinert werden.

## 10 Die Katze bewegen

Starten Sie das neue Skript für die Katze mit der Anweisung **Wenn ich <start> empfangen** (aus **Steuerung**), wobei **start** die Message ist, die wir in Schritt 5 erstellt haben. Fügen Sie dann **gehe zu x: <-100> y: <-100>** (aus **Bewegung**) hinzu. Dies bewegt die Katze in die Ecke gegenüber der Maus (die Mittelposition ist **x: <0> y: <0>**).



## 11 Eine neue Schleife erstellen

Auch die Katze benötigt eine Schleife, damit das Spiel am Laufen gehalten wird. Fügen Sie **wiederhole bis <...>** (aus **Steuerung**) hinzu und setzen Sie **wird <Objekt1> berührt?** (aus **Fühlen**) in das leere Feld der Anweisung. Die Schleife endet also, sobald die Katze die Maus gefangen hat.



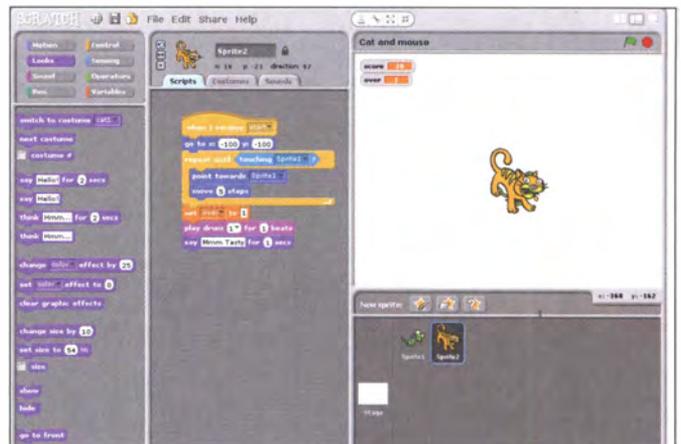
## 12 Die neue Schleife erweitern

Doch was genau soll die Katze so lange tun, bis sie die Maus gefangen hat? Dies bestimmen die Anweisungen **drehe dich zu <Objekt1>** und **gehe <4>-er-Schritt** (beide aus **Bewegung**), die Sie an den **wiederhole**-Block anhängen. Aus dem Verhältnis der Schrittzahlen von Katze und Maus (in unserem Beispiel **4** und **7**) ergibt sich der Schwierigkeitsgrad des Spiels.



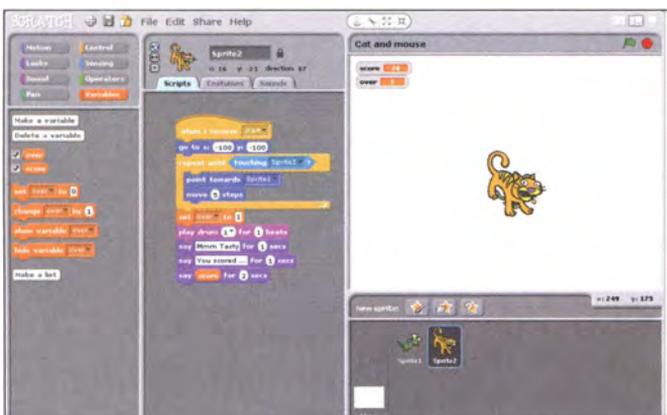
## 13 Beide Schleifen beenden

Die Schleife der Katze endet, wenn sie die Maus erreicht. Damit in diesem Moment auch die Schleife der Maus endet, muss die Bedingung aus Schritt 7 erfüllt werden, nämlich dass **over** auf **1** steht. Darum fügen wir dem **wiederhole**-Block der Katze die Anweisung **setze <over> auf <1>** (aus **Variablen**) hinzu.



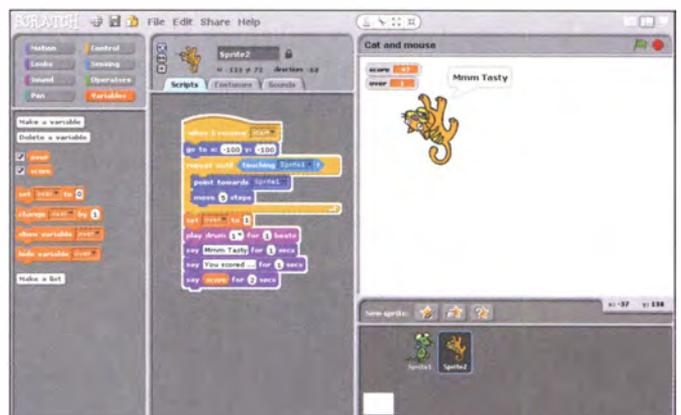
## 14 Das Spiel ist aus

An diesem Punkt soll dem Spieler mitgeteilt werden, dass das Spiel vorbei ist. Um dies zu tun, fügen wir dem Skript der Katze die Anweisung **Sage <Hab' ich dich!> für <2> Sek.** (aus **Aussehen**) hinzu.



## 15 Punktestand anzeigen

Zuletzt soll dem Spieler noch angezeigt werden, wie viele Punkte er erzielt hat. Die Variable **score** in der Schleife der Maus wurde bei jedem Durchgang um **1** erhöht und der Endwert soll uns als Punktestand dienen. Um ihn anzeigen zu lassen, müssen Sie noch **Sage <Erzielte Punkte:> für <1> Sek.** und **Sage <score> für <5> Sek.** anhängen.



## 16 Achtung, fertig, los!

Drücken Sie die Taste **r** und retten Sie die Maus vor der Katze! Da die Anweisungen **drehe dich <im Uhrzeigersinn> um <15> Grad** und **drehe dich <gegen den Uhrzeigersinn> um <15> Grad** den Pfeiltasten zugeordnet sind, können Sie die Maus mit den Pfeilen steuern. Viel Spaß und toi, toi, toi!

# Die Legende

Mit Python auf dem Raspberry Pi ein eigenes Adventure-Spiel schreiben.

**D**er Raspberry Pi ist ein wundervolles Stück Technologie, doch viele nutzen sein volles Potenzial gar nicht aus. Mit einem Raspberry Pi kann man fast alles verwirklichen – einen Roboter, der seine Umgebung wahrnimmt, ein Mediencenter zum Filmgucken oder auch eine Welt voller Fantasie und Abenteuer, erstellt aus ein paar einfachen Codezeilen und einer Menge Vorstellungskraft.

In den 1980er-Jahren steckte die Computergrafik noch in den Kinderschuhen: Die Figuren waren pixelig und die Farbpalette sehr begrenzt. Damals war es üblich, dass Adventure-

In dieser Übung erstellen wir mit einem Raspberry Pi und der Programmiersprache Python unser eigenes Textadventure-Spiel mit einer eigenen Welt und eigenen Charakteren. All das vollbringen wir mit etwas Python-Code und einigen wenigen Programmierkonzepten. Python ist als Programmiersprache relativ einfach zu erlernen.

## Was ist denn Python?

Python ist eine textbasierte Programmiersprache, das heißt, der Code wird in einen Text-Editor eingegeben. Python ist nachsichtig in der Syntax und leicht zu lesen, wodurch das Programmieren von Anfang an Spaß macht.

Auf dem Raspberry Pi ist bereits ein guter Text-Editor namens IDLE vorinstalliert – wir werden damit unser Spiel schreiben. Auf dem Raspberry-Pi-Desktop finden Sie eine Verknüpfung zu IDLE.

Python ist auf jedem Raspberry Pi, auf dem das Raspbian-Betriebssystem läuft, vorinstalliert. Falls Raspbian auf Ihrem Raspberry Pi nicht installiert ist, können Sie es auf [www.raspberrypi.org/downloads](http://www.raspberrypi.org/downloads) herunterladen. Es ist ein Bestandteil des NOOBS-Archivs (New Out Of the Box Software; neue fertig verwendbare Software), welches man herunterladen und auf eine leere SD-Karte (4 GB oder größer) extrahieren kann.

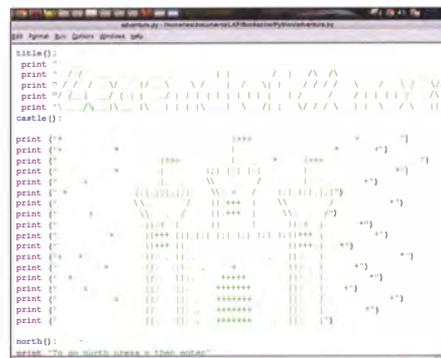
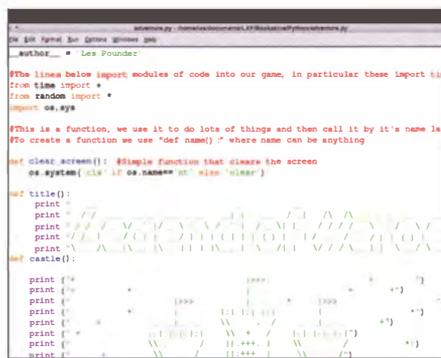
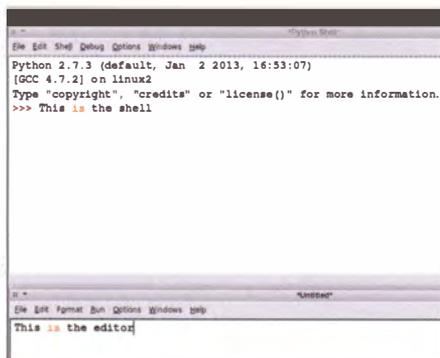
Aktuell sind zwei Versionen von Python verfügbar: Version 2.7 und Version 3. In dieser Übung verwenden wir die Version 2.7, da es mehr Support und Dokumentation dafür

„Für unser Spiel brauchen wir eine Story und eine Logik, nach der sie sich entfaltet.“

und Rollenspiele nur aus Text bestanden und sich auf die Vorstellungskraft der Spieler stützten, um eine Welt lebendig werden zu lassen. In Spielen wie Zork entstanden reichhaltige Umgebungen mit bezaubernden Geschichten und Charakteren, aber mit sehr wenig Grafiken zur Illustration. Das setzte sich bis in die späten 1980er-/frühen 1990er-Jahre fort und änderte sich erst mit der fantastischen Arbeit der heute legendären Entwicklerschmiede LucasArts, wo etliche Grafikadventure-Klassiker wie Loom, Monkey Island und Full Throttle kreiert wurden.

## Mit dem Programmieren loslegen

Folgen Sie diesen Schritten, um Ihr Abenteuer zu starten.



### 1 Das Spiel spielen

IDLE hat zwei Arbeitsfenster: den Editor, der den Code enthält, und eine Shell, in der das Spiel gespielt werden kann. Um das Spiel zu starten, klicken Sie im Editor auf „Starten“ und dann „Modul starten“. Das Shellfenster erscheint im Vordergrund und das Spiel legt los.

### 2 Module einbinden

Module von anderen Programmierern können in Ihren Code eingebunden werden. Wir binden zwei Module ein: Random, womit wir die Optionen des Spielers per Zufall entscheiden lassen können, und Time, womit wir mit Zeit arbeiten und das Spiel pausieren können.

### 3 Funktionen erstellen

Man kann beliebig viel Code in eine Funktionsgruppe gruppieren. Um eine zu erstellen, schreiben wir **def name():** – das definiert eine Funktion mit dem Namen **name**, und wir können die Funktionen nennen, wie wir möchten. Mit Enter rückt Python unseren Code um vier Leerzeichen ein.



## Weiter geht's

Kommen wir nun zur Sache:

```
def setup():
    #global is used to create variables that can be used throughout our game
    global name
    global HP
    global MP
    #Our variable "name" is used to store our name, captured by keyboard input.
    name = raw_input("What is your name warrior? ")
    #randint is a great way of adding some variety to your players statistics.
    HP = randint(5,20)
    MP = randint(5,20)

def villager():
    #This will create a randomly named Villager to interact with
    global npcname
    global response
    #Below is a list, we can store lots of things in a list and then retrieve the
    responses = ["Hi", "Are you a hero?", "Are you from this village?", "There he
    npcnamechoice = ["Roger", "Dexter", "Sarah", "Susan"]
    #Shuffle will shuffle the list contents into a random order.
    shuffle(npcnamechoice)
```

```
adventure.py - /home/les/documents/LXF/Bookazine/Python/adventure.py
File Edit Format Run Options Windows Help

def setup():
    #global is used to create variables that can be used throughout our game
    global name
    global HP
    global MP
    #Our variable "name" is used to store our name, captured by keyboard input.
    name = raw_input("What is your name warrior? ")
    #randint is a great way of adding some variety to your players statistics.
    HP = randint(5,20)
    MP = randint(5,20)

def villager():
    #This will create a randomly named Villager to interact with
    global npcname
    global response
    #Below is a list, we can store lots of things in a list and then retrieve the
    responses = ["Hi", "Are you a hero?", "Are you from this village?", "There he
```

### 1 Variablen

Mittels Variablen speichern wir Informationen (etwa den Erzählstoff) für später. Variablen kann man sich vorstellen wie eine Kiste: Man kann irgendetwas hineinlegen und dann einen Namen draufschreiben. Manchmal werden Variablenamen mit der Bezeichnung „global“ verwendet; dadurch können sie innerhalb und außerhalb von Funktionen benutzt werden.

```
File Edit Format Run Options Windows Help

#Remember those functions we created at the start of the code? Well here we are using them
print "\n"
north()
east()
west()
move = raw_input("Where would you like to go? ")
if move == 'n':
    print "\nYou move to the north, walking in the sunshine."
    print "A villager is in your path and greets you"
#elif is short for Else If and it means that if the previous condition is false, to check t
elif move == 'e':
    print "\nYou walk to the river which lies to the east of your home."
    print "A villager is in your path and greets you"
elif move == 'w':
    print "\nYou walk to the field of wild flowers, stopping to take in the beauty"
    print "A villager is in your path and greets you\n"

villager()
enemy()
sleep(3)

fight = raw_input("Do you wish to fight? ")
```

### 2 Eingabe einlesen

Mit dem Befehl „raw\_input“ wird der Benutzer zur Eingabe aufgefordert. Damit erhalten wir den Namen des Spielers sowie seine Entscheidungen während des Spiels. raw\_input liest Tastatureingaben als Zeichenfolge ein; wenn also Zahlen eingegeben werden, müssen diese mit **int()** oder **float()** in eine Ganzzahl oder Gleitkommazahl konvertiert werden.

```
File Edit Format Run Options Windows Help

#Remember those functions we created at the start of the code? Well here we are using them
print "\n"
north()
east()
west()
move = raw_input("Where would you like to go? ")
if move == 'n':
    print "\nYou move to the north, walking in the sunshine."
    print "A villager is in your path and greets you"
#elif is short for Else If and it means that if the previous condition is false, to check t
elif move == 'e':
    print "\nYou walk to the river which lies to the east of your home."
    print "A villager is in your path and greets you"
elif move == 'w':
    print "\nYou walk to the field of wild flowers, stopping to take in the beauty"
    print "A villager is in your path and greets you\n"

villager()
enemy()
sleep(3)

fight = raw_input("Do you wish to fight? ")
```

### 3 Entscheidungen

Wir verwenden Bedingungen, um den Ablauf des Spiels festzulegen. Beispiel: In welche Richtung gehen Sie? Wir speichern die eingegebene Richtung in einer Variable namens **move** und vergleichen dann den Wert der Variable mit den verschiedenen möglichen Eingaben. Bei einer Übereinstimmung wird der entsprechende Code ausgeführt.

```
File Edit Format Run Options Windows Help

print "Would you like to venture out into the land? Press y then enter to continue"
#Below we use raw input to ask for user input, and if it is equal to y, then the code under
if raw_input() == "y":
    print "You are in your home, with a roaring fireplace in front of you, above the fire
    print "Would you like to take your sword and shield? Press y then enter to continue"
    if raw_input() == "y":
        #This is a list, and it can store many items, and to do that we "append" items to t
        weapons = []
        weapons.append("sword")
        weapons.append("shield")
        print "You are now carrying your" + " " + weapons[0] + " " + "and your" + " " + wez
        print "Armed with your " + weapons[0] + " " + "and " + weapons[1] + " " you swing ope
    else:
        print "You choose not to take your weapons"
```

### 4 else und elif

„if ... elif ... else“ ist eine Kette von Bedingungen und funktioniert so: Die erste Bedingung (**if**) wird geprüft, und wenn sie wahr ist, wird der Code darunter ausgeführt. Anderenfalls wird die zweite Bedingung (**elif**) geprüft und eventuell der dazugehörige Code ausgeführt. Wenn alle Bedingungen falsch sind, wird der Code unter else ausgeführt.

```
File Edit Format Run Options Windows Help

def setup():
    #global is used to create variables that can be used throughout our game
    global name
    global HP
    global MP
    #Our variable "name" is used to store our name, captured by keyboard input.
    name = raw_input("What is your name warrior? ")
    #randint is a great way of adding some variety to your players statistics.
    HP = randint(5,20)
    MP = randint(5,20)

def villager():
    #This will create a randomly named Villager to interact with
    global npcname
```

### 5 Listen

Listen sind Behälter, die eine Menge Daten enthalten können. Man stelle sich das wie einen Aktenschrank vor. In unserem Spiel haben wir eine Liste verwendet, um das Inventar zu speichern (das Schwert und den Schild). Wir erstellen zunächst eine leere Liste:

```
weapons = []
```

Wir speichern die Liste in einer Variablen, der wir einen beliebigen Namen geben. Wir können nun an die Liste anhängen:

```
weapons.append("Schwert")
```

### 6 Zahlen auswählen

Mit Zufallszahlen wird der Held einzigartig. Um Zufallszahlen zu nutzen, verwenden wir das **random**-Modul, insbesondere eine Funktion namens **randint** (random integer; zufällige Ganzzahl). **randint** erwartet einen Zahlenbereich, der wie folgt angegeben wird:

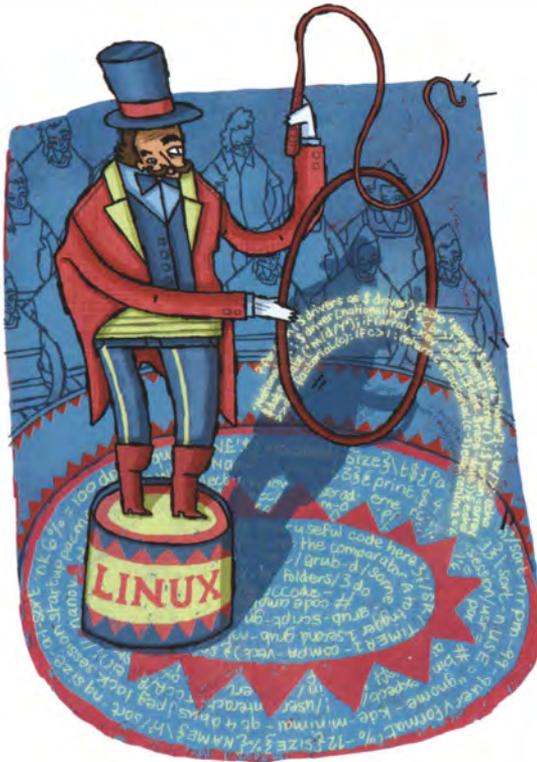
```
randint(5,20)
```

Der Zahlenbereich geht also von 5 bis 20; **randint** wählt eine Zufallszahl in diesem Bereich aus.



# Chat: Wir bauen

Wenn Sie Ihre eigene künstliche Intelligenz bauen wollen, können Sie mit diesem Guide einen eigenen Chatbot programmieren.



**S**tellen Sie sich vor, wie einfach das Leben wäre, wenn wir einige kleine Helfer hätten, die uns die Arbeit abnehmen: Die Post aus dem Briefkasten holen, die Wäsche waschen – all die lästigen Dinge eben, um die sich niemand reißt. Leider leben wir noch ein wenig zu früh in der menschlichen Geschichte, um bereits in den Genuss kostengünstiger robotischer Lakaien zu gelangen, sodass wir uns in vielen Fällen selber helfen müssen. Die nächstbeste Alternative wäre da ein virtueller Diener, der für uns allerlei Dinge erledigt. Natürlich erfüllen schon sehr viele Programme dieses Anforderungsprofil, doch ich denke im Speziellen an ein einfach zu bedienendes Interface, das nützliche Aufgaben erledigen kann und Ihnen nur Dinge mitteilt, die Sie auch wirklich wissen wollen. Wenn dieses Interface nun auch noch Ihre Freunde in Verwirrung stürzen und Ihre Feinde die Gründe für ihre Abneigung Ihnen gegenüber neu bewerten lassen würde – umso besser!

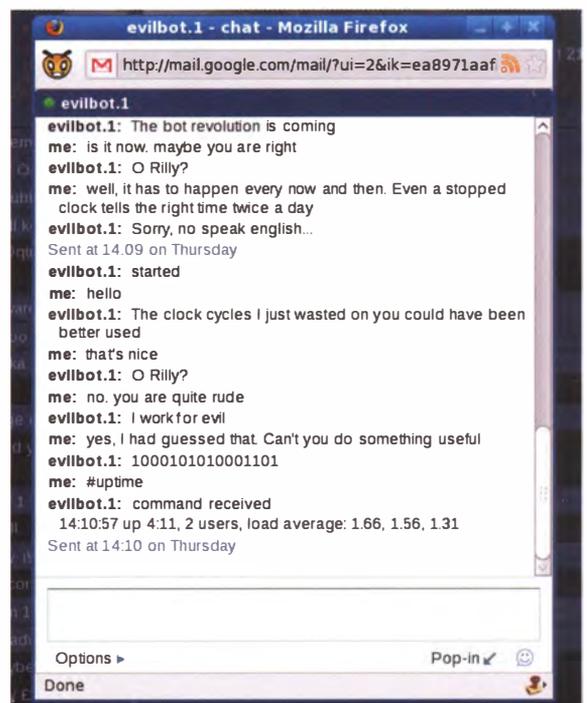
Einer der am wenigsten verwendeten – jedoch sicher bequemsten – Wege, mit so einem Diener zu kommunizieren, wäre über einen Chat. Warum sollte man sich auch mit komplizierten SSH-Tunneln oder mit langweiligen netzbasieren Programmen herumschlagen, wenn Sie auf einfache Art und Weise diese Aufgaben durch ein Medium erledigen könnten, das Sie vermutlich ohnehin schon die ganze Zeit verwenden? Dies sei darum der Plan: einen kleinen Kobold in Form eines

Chatbots zu erschaffen, der geduldig in einem Chatkanal wartet und sich von selber nur meldet, wenn es etwas Wichtiges zu berichten gibt.

## Hallo, mein Name ist Xmpppy

Das Jabber/XMPP-Protokoll wird von Python gut unterstützt. Es existiert als Element des allumfassenden Netzwerkmoduls *Twisted*, jedoch gibt es auch eine etwas abge-speckte Version namens *Xmpppy*, die für unsere Zwecke vollkommen ausreichend ist. Sie sollten ohne Weiteres in der Lage sein, die notwendigen Pakete für Ihre Distribution zu finden oder aber Sie laden sich einfach selbst den Python-Code von der Seite <http://xmpppy.sourceforge.net> herunter.

Um zu verstehen, wie Sie *Xmpppy* einsetzen können, beginnen wir einfach mit ein paar Beispielen für die Befehlszeile. Zuvor benötigen Sie jedoch einen Account. Sie könnten beispielsweise einen separaten Google-Account für Ihren Bot anlegen. Weiterhin brauchen Sie mindestens zwei Jabber-IDs für Testzwecke. In unserem Testlabor haben wir für diesen Zweck einen Gmail-Account für unseren Bot kreiert, uns via Browser eingeloggt und einen anderen Gmail-Account zu einem Chat eingeladen. Selbstverständlich können Sie all das auch innerhalb von *Xmpppy* abwickeln, aber für unsere ersten Gehversuche wäre dies viel zu kompliziert gewesen. Daher ist es vernünftiger, für die erste Phase zwei Accounts zu



➤ Folgen Sie unseren Hinweisen, dann wird Ihr eigener Chatbot sehr schnell zu einer persönlichen Nervensäge.

# Bots!

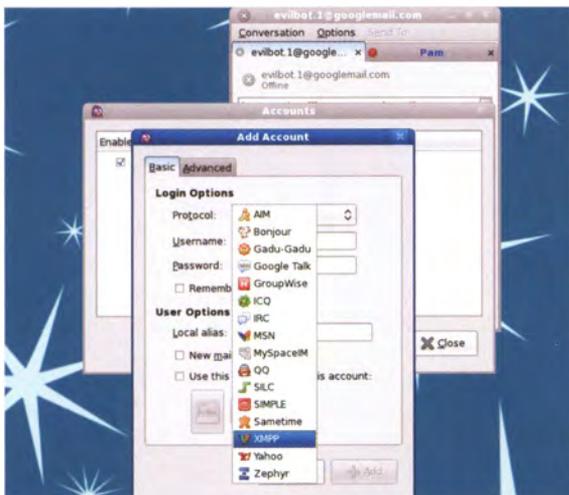
verwenden, die in der Lage sind, miteinander zu kommunizieren. Wenn Sie so weit sind, geben sie in der Konsole einfach **python** ein, um den interaktiven Kommandozeileninterpreter von Python zu starten.

```
>>> import xmpp
>>> jid=xmpp.protocol.JID("botadresse@googlemail.com")
```

Hier haben wir eine Instanz einer Jabber-ID eingerichtet. In diesem Fall handelt es sich um eine E-Mail-Adresse, da wir einen Google-Account in diesem Tutorial verwenden. Die Prozedur lässt sich aber mit jeder beliebigen Jabber-ID durchführen. Wir haben hier zwei Bereiche (es können auch drei sein, dazu später mehr), nämlich den Namen des Users sowie die Domain. *Xmpppy* versucht, in der Domain einen Server zu finden, den es ansprechen kann. Nachdem wir nun den User bestimmt haben, sollten wir ein Objekt erschaffen, das als Client fungiert. Der Client ist die Steuerungseinheit in *Xmpppy*, welche die Verbindungen kontrolliert, die Nachrichten verwaltet und generell mit dem Server interagiert. Wir müssen nur noch ein paar einfache Schritte vornehmen, um den Client einzurichten und eine Verbindung mit ihm aufzubauen. Wir müssen zunächst eine Instanz des Clients erschaffen (hierbei ist eine weitere Jabber-ID Voraussetzung), uns dann mit dem Server verbinden und uns zuletzt noch authentifizieren.

```
>>> myclient=xmpp.Client(jid.getDomain(),debug=[])
>>> myclient.connect()
'tls'
>>> myclient.auth(jid.getNode(),'botpasswort')
'sasl'
```

Hier gibt es einige Dinge zu beachten. Zuerst haben wir einen Client erschaffen und mittels des **jid.getDomain-**



› Das XMPP/Jabber-Protokoll ist weit verbreitet, sodass Sie sich mit Ihrem Bot über zahlreiche Clients, inklusive Pidgin, verbinden können.

## Disco-Plugin

XMPP beinhaltet ein Set von Plugins mit dem Namen *Disco*. Hiermit werden die Nachrichtenprotokolle so erweitert, dass auch andere Typen von Nachrichten bearbeitet werden können, darunter SIP (Voice-Chat), Dateitransfer und alles mögliche andere, was über ein Punkt-zu-Punkt-Netzwerk übertragen werden kann. Es spricht nichts dagegen, Ihren Bot

auch um diese Funktionen zu erweitern. Vielleicht könnte er Ihnen Speicherplatz zur Verfügung stellen oder aber wichtige Nachrichten vorlesen. Google hat die Implementierung von XMPP noch um zusätzliche Funktionen erweitert, die Sie hier finden: [http://code.google.com/apis/talk/jep\\_extensions/extensions.html](http://code.google.com/apis/talk/jep_extensions/extensions.html)

Befehls den Servernamen von unserem **jid**-Objekt, welches wir schon vorher angelegt hatten, in Erfahrung gebracht. Zusätzlich muss auch festgelegt werden, wie umfassend der Debugger Ihnen Rückmeldung gibt. Wir haben ein leeres Feld stehen lassen, aber falls Sie maximal informiert werden möchten, geben Sie stattdessen **always** ein.

## Verbindung einrichten

Nachdem wir das Debug-Level bestimmt haben, haben wir eine Verbindung (tragen sie dabei Ihr Passwort ein) aufgebaut und als Antwort die Meldung **tls** bekommen – dies bedeutet, dass eine sichere Verbindung hergestellt wurde. Die Ausgabe hätte auch **tcp** – für eine Standardverbindung – oder ein leerer String – für eine fehlgeschlagene Verbindung – sein können. Das Gelingen des nächsten Schritts hängt ein wenig von Ihrer Fingerfertigkeit auf der Tastatur ab, denn die Server von Google verlangen eine schnelle Authentifizierung. Um Zeit zu sparen, sollten Sie versuchen, die Kommandos in der Befehlszeile zusammenzufügen:

```
>>> myclient.connect() ; myclient.auth(jid.getNode()
(), 'botpasswort')
'tls'
'sasl'
```

Die Antwort **sasl** bedeutet, dass Ihre Verbindung via SASL (Simple Authentication and Security Layer) angenommen wurde. Der letzte Schritt, um Ihre Verbindung zu initialisieren, ist die Bekanntgabe Ihres Anwesenheitsstatus. Google-Chat und andere Jabber-Dienste sehen einen unterschiedlichen Status der Anwesenheit (anwesend, beschäftigt usw.) vor. Diese Funktion wird vor allem vom Server dafür verwendet, Listen über verfügbare Kontakte zu erstellen. Aber auch hierfür liefert Ihnen *Xmpppy* den richtigen Befehl:

```
myclient.sendInitPresence()
```

Einige Server verwehren Ihnen den Zugriff, bis Sie einen Anwesenheitsstatus festgelegt haben!

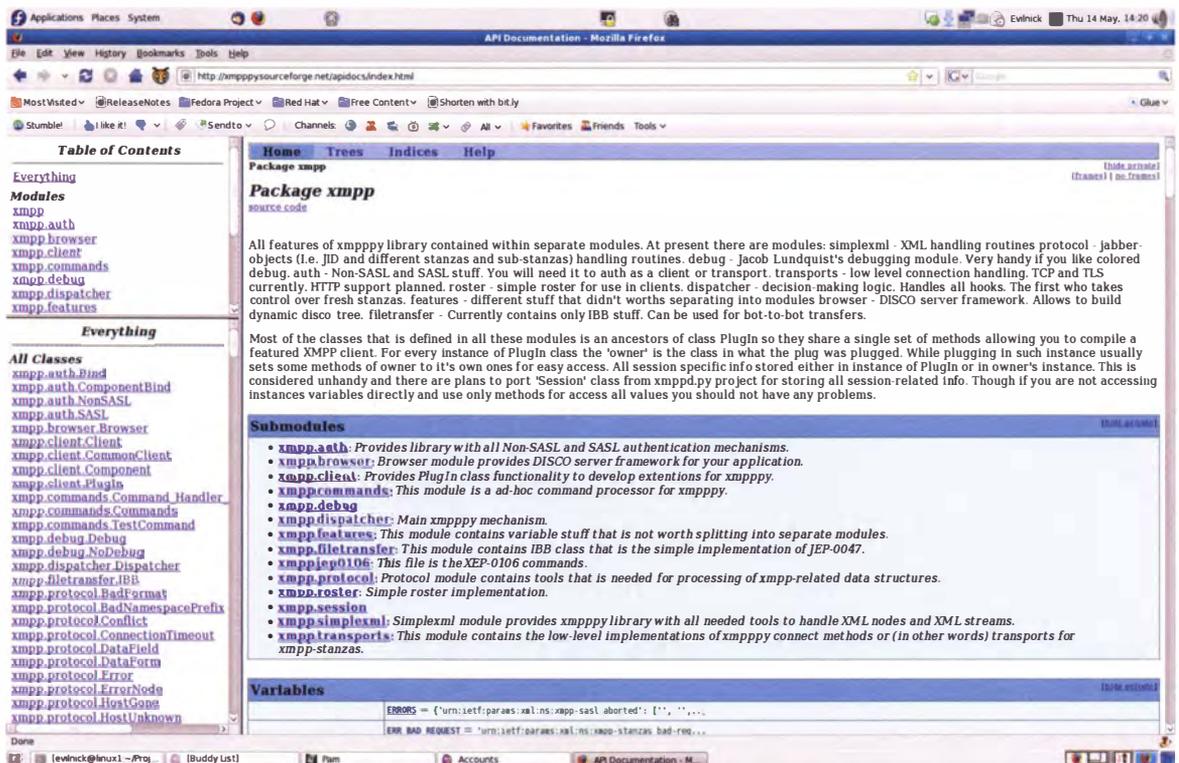
## Anwesenheitsstatus übermitteln

Endlich sind wir drin! Bevor es jetzt zu langweilig wird, schreiben wir eine Nachricht an unser Ziel (etwa Ihre eigene »

### Quick-Tipp

Sie erhöhen die Sicherheit Ihres Bots auf einfache Weise, indem Sie ihn nur Befehle von einer bestimmten ID annehmen lassen.

» Die Projektseite von Xmpppy ist ein wenig schmal, was die Dokumentation angeht. Die API-Dokumente sollten aber voll und ganz ausreichen, um Ihnen zu zeigen, wie das Modul funktioniert.



» Google-Adresse). Damit es wirklich losgehen kann, brauchen wir natürlich noch die Empfängeradresse und den Inhalt unserer Nachricht. Haben wir das alles, können wir sogleich eine Nachrichteninstanz erschaffen und diese abschicken:

```
mymsg=xmpp.protocol.Message("evilbot.1@gmail.com",
"hello", "chat")
myclient.send(mymsg)
```

Vorausgesetzt, Sie haben es geschafft, die Chateinladung des Bots anzunehmen, sollten Sie nun eine kleine Überraschung erleben.

Falls Sie sich fragen, wie Sie auf dem Google-Server Ihren Anwesenheitsstatus festlegen können: Dies geschieht per XMPP und zwar mithilfe einer speziellen Nachricht, die nur an den Server gesendet wird.

```
>>> presence = xmpp.Presence(status = "Ready!", show =
"chat", priority = '1')
>>> myclient.send(presence)
```

Das Xmpypy-Modul verfährt etwas anders mit dem Verschieken der Anwesenheitsmeldungen, da die Parameter sich unterscheiden, aber prinzipiell ist der Mechanismus der gleiche.

## Nachrichten empfangen

Um einen voll funktionsfähigen Bot zu erhalten, brauchen wir natürlich auch eingehende Nachrichten. Das bedeutet leider

etwas Fummelarbeit. Das Xmpypy-Modul empfängt die Nachrichten über das von Ihnen erschaffene Client-Objekt und behält sie fürs Erste im Speicher, bis sie weiterverarbeitet werden. Die Verarbeitung geschieht bei Xmpypy mithilfe sogenannter Handler. Bevor Sie eine Nachricht verarbeiten können, müssen Sie erst einmal eine Funktion oder Methode definieren, die als Empfänger der Daten fungiert. Wenn Sie bereit sind, die Nachrichten aus dem Speicher zu laden, aktivieren Sie einfach diesen Prozess auf dem Client-Objekt. Das klingt komplizierter, als es ist: Im Grunde heißt das nur, Sie sagen dem Client, wo er die Nachrichten hinschieben soll, und stupsen ihn dann an, damit er sie durch die erwähnte Funktion schickt. Man kann das Ganze per Python-Befehlszeile aufsetzen, allerdings wird es bei diesem Weg später ein bisschen unübersichtlich, da wir neben der Handler-Funktion noch eine Programmschleife benötigen. Bequemer ist es daher, für diesen Prozess ein eigenes Objekt zu erschaffen (dazu kommen wir noch). Hier zunächst die Befehlszeilen, mit denen wir eine Nachricht an uns selbst senden.

```
>>> def msgparser(instance, message):
... print "Neue Nachricht!"
... print "from" + str(message.getFrom())
... print "msg" + str(message.getBody())
...
>>> myclient.RegisterHandler('message', msgparser)
>>> mymsg=xmpp.protocol.Message("evilbot@gmail.com", "hello", "chat")
>>> myclient.send(mymsg)
'5'
>>> myclient.Process(1)
Neue Nachricht!
from: evilbot@gmail.com
msg: hello
1493
>>>
```

## Hilfe zu Python

Wenn Python für Sie Neuland ist, Sie aber schon Erfahrung in anderen Programmiersprachen gesammelt haben, sollten Sie nicht auf allzu große Probleme stoßen, solange Sie sich daran halten, die Code-Zeilen ordentlich

einzurücken. Auf der Python-Website finden Sie einen großen Fundus von Dokumentationen, die sich mit den Funktionen, der Syntax und den Modulen dieser Programmiersprache beschäftigen.

Die Parserfunktion ist noch relativ einfach gestrickt. Sie benutzt die zur Verfügung stehenden Methoden `getFrom()` und `getBody()` einer eingehenden Nachricht und verwandelt diese in einen String, der auf der Konsole ausgegeben wird. Für einen echten Bot müssen wir natürlich den Absender in eine Variable packen, um so die Möglichkeit zu haben, für eine Antwort die Nachricht weiterzuparsen. Daher werden wir bei unserem Bot eine Syntax anwenden, welche Sonderbefehle, vor denen ein „#“ steht, direkt an ihn weiterleitet. Das heißt, sollte eine Nachricht eingehen, die mit diesem Hash-Zeichen versehen ist, sollten wir sie irgendwie bearbeiten. Andernfalls antworten wir mit einer zufällig ausgewählten Nachricht aus einer Liste.

## Eine Abkürzung nehmen

Einen Handler für Befehle hinzuzufügen, kann ziemlich beschwerlich und lästig werden. Um also Zeit und Mühe zu sparen, definieren wir eine Klasse, die eine Art Abkürzung enthält. Wir verwenden die `eval`-Funktion von Python, um eine Aufforderung an unsere Methode in der Klasse zu richten, die den gleichen Namen trägt wie der Befehl, der an sie gesendet wird. Hiermit schummeln wir ein wenig, jedoch sparen Sie dadurch auch ein bisschen Platz und es erleichtert Ihnen das Hinzufügen weiterer Befehle – Sie müssen einfach nur eine neue Methode für diesen erschaffen. Um einen sauberer programmierten Bot zu kreieren, müssten Sie wahrscheinlich eher einen Handler-Mechanismus programmieren, damit zusätzliche Befehle in Instanzen der Klasse hinzugefügt werden können. Hier ist aber erst einmal die einfachere Variante:

```
def messageparse(self, instance, message):
    m = message.getBody()
    sender=message.getFrom()
    if (m[0]=='#'):
        self.log('command received')
        # hier Sonderfälle einfügen
        # allgemeiner Befehl - erwartet eine existierende Methode
        # und Benötigung einer Absender-ID
    try:
        eval('self.'+m[1:]+'(sender)')
    except:
        self.client.send((xmpp.protocol.Message(sender, 'Das geht
        leider nicht.')))
    else:
        # etwas sagen, um die Etikette zu wahren
        self.client.send((xmpp.protocol.Message(sender,
        random.choice(self.responses))))
```

Wie Sie sehen können, wird die eingehende Nachricht auf das Vorhandensein eines # hin überprüft, und falls das Zeichen gefunden wird, erschaffen wir einen Methodenaufruf vom Rest des Strings, überprüfen die ID des Absenders und versuchen, diese auszuführen. Die `try`- und `execute`-Struktur versucht hierbei, Ausnahmefälle wie zum Beispiel die Nichtexistenz der Methode abzufangen. Sollte kein passender Befehl gefunden werden, senden wir eine zufällige Nachricht aus

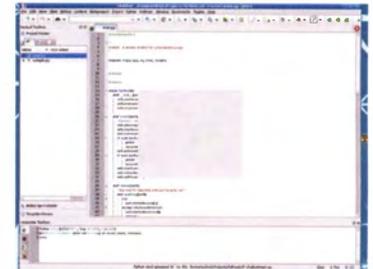
## Versionen von Python

Mittlerweile gibt es Python 3.0, aber da keine Abwärtskompatibilität mit früheren Versionen besteht, benutzen viele Distributionen bis auf

Weiteres die Version 2.x als Standard. Der Code in unseren Tutorials ist 2.x-kompatibel, damit er für die meisten User benutzbar ist.

## Python-Editoren

Python nimmt es ziemlich genau mit der Syntax. Es kann Sie aber vor Probleme stellen, wenn Sie versuchen, mit unzureichenden Editoren zu programmieren. Wir benutzen *Vim* und *Kate*. Beide haben – neben einigen anderen nützlichen Features – die Funktion, die Syntax hervorzuheben, sodass es mit ihnen einfach von der Hand geht, Scripts in Python zu programmieren. Sie können auch *Eric*, eine auf Python basierende IDE verwenden, die mit vielen python-spezifischen Funktionen aufwartet. Das Download-Paket für die gebräuchlichsten Distributionen finden sie hier: <http://eric-ide.python-projects.org/eric4-download.html>



Benutzen Sie entweder einen Editor mit einer Hervorhebungsfunktion für die Python-Syntax, oder holen Sie sich die speziell für diese Sprache konzipierte IDE namens Eric.

der Liste `responses` zurück. Im wirklichen Leben müssen Sie hierfür natürlich das `random`-Modul importieren. Der Befehl `random.choice` nimmt wahllos einen Inhalt aus allem, was ihm zur Verfügung steht. Ein Befehls-Handler könnte folgendermaßen aussehen:

```
def uptime(self, sender):
    import subprocess
    p=subprocess.Popen(["uptime"], stdout=subprocess.PIPE)
    r=p.communicate()[0]
    self.client.send((xmpp.protocol.Message(sender,r)))
```

Hier sollten wir einige Erklärungen nachliefern, da dieser Code Befehle bearbeitet, die auf der lokalen Maschine ausgeführt werden, auf der auch der Bot läuft. Am Anfang steht die Definition, welche die eigene Instanz akzeptiert (dies wird von Python vorausgesetzt), und die Information des Absenders, die der Nachrichten-Handler für uns herausgefiltert hat. Zusätzlich haben wir einen `subprocess` aus der Bibliothek von Python importiert, um einen Befehl lokal bearbeiten zu können.

Auch benutzen wir etwas, was als `Popen`-Methode (siehe die Zeile, die mit `p=subprocess.Popen` beginnt) bezeichnet wird.

Diese wird ausführlich auf <http://docs.python.org/library/subprocess.html> erklärt.

Kurz zusammengefasst leiten wir den auszuführenden Befehl weiter und starten eine Anfrage, dass der Standardausgang an einen Kanal angeschlossen werde. Danach sind wir in der Lage, diesen Ausgang über die `communicate`-Methode der `Popen`-Instanz abzufragen und die zurückgeschickte Antwort des Befehls zu bearbeiten.

## Ausblick

Womöglich möchten Sie noch eigene Methoden in die Klasse integrieren, um weitere Befehle zu definieren. Es gibt nichts, was Sie daran hindern sollte, diese Befehle dafür einzusetzen, Aufgaben zu automatisieren oder externe Quellen anzusteuern.

Der Bot ist eigentlich nicht viel mehr als ein Übertragungskanal – ein einfach zu bedienendes Interface, das es Ihnen ermöglicht, mit einem Script zu kommunizieren. Welche konkreten Aufgaben er letztendlich für Sie erledigt, bleibt Ihren Wünschen und Vorstellungen überlassen. ■

# Ein grafisches

Die Clutter-Bibliothek gestattet einen Blick über den Tellerrand der Kommandozeile hinein in die Welt der grafischen Benutzeroberflächen.



› Hier ist es noch ein bisschen dunkel. Aber dank Clutter wird sich das bald ändern.

die Objekte üblicherweise als Fenster oder Panel definieren, bezeichnet Clutter den grafischen Darstellungsbereich als stage („Bühne“) und ein darin erscheinendes Objekt als actor („Schauspieler“). Die actors haben mehr Eigenschaften als ein Standard-Widget, da sie de facto in einer 3-D- statt einer 2-D-Umgebung existieren.

## Objekte erschaffen

Nun aber zur Praxis! Öffnen Sie Ihre übliche Python-Entwicklungsumgebung (zum Beispiel die Bash-Shell) und legen Sie los:

```
>>> import clutter
>>> stage = clutter.Stage()
>>> stage.set_size(500,300)
>>> red=clutter.Color(255,0,0,255)
>>> black=clutter.Color(0,0,0,255)
>>> stage.set_color(black)
>>> stage.show_all()
```

Schließen Sie das Fenster, das sich daraufhin geöffnet hat, und sehen Sie sich nochmals den Code an. Was genau ist passiert? Die erste Programmzeile hat das Clutter-Modul geladen. Clutter öffnet wiederum einige Backend-Module, die mit Darstellungsbibliotheken verbunden sind, damit Objekte auf dem Bildschirm angezeigt werden können. Als Nächstes haben wir ein stage-Objekt erschaffen. Der Viewport, also das Darstellungsfeld, ist die Bühne, auf der die actor-Objekte ihre Rollen spielen.

Die Attribute sind sehr einfach festzulegen, es müssen lediglich Methoden für die stage-Klassen zugewiesen werden. Hier haben wir Größe und Farbe festgelegt. Die Parameter der Methode für die Größe sind Breite und Höhe und die Farbe wird als RGB-Wert plus Alpha im **clutter.Color**-Objekt festgelegt. Der letzte Befehl bestimmt, dass – wie bei

**W**ir haben uns im Rahmen unserer Python-Anleitungen und -Projekte bislang auf das Schreiben von Terminal-Skripten konzentriert, doch an dieser Stelle möchten wir Ihnen einmal eine Interaktion mit dem Nutzer mithilfe einer grafischen Benutzeroberfläche (GUI) demonstrieren. GUI-Code wird schnell voluminös, da die Darstellung grafischer Elemente auf dem Bildschirm aus Coding-Sicht eine komplexe Angelegenheit darstellt.

Dank der PyClutter-Bibliothek ist es jedoch möglich, den Code, der ein grafisches Interface steuert, trotzdem relativ kompakt zu halten. Das Programm, das wir erstellen wollen, hat einen Umfang von weniger als 70 Zeilen. Es handelt sich um ein kleines, aber nützliches Werkzeug, mit dessen Hilfe Sie die Geschwindigkeit Ihrer Internetverbindung messen können. Das Schreiben des Programms führt Sie an die Möglichkeiten von Clutter und PyClutter heran und hilft Ihnen hoffentlich, eine Vorstellung zu gewinnen, wie die Bibliothek sinnvoll im Zusammenspiel mit Python-Web-Anwendungen eingesetzt werden kann.

Zunächst sollten wir uns einen Moment mit der Terminologie von Clutter beschäftigen, denn diese mag auf den ersten Blick seltsam wirken. Während andere GUI-Tools

# Interface bauen

anderen GUI-Toolkits – das Objekt gezeigt wird, bevor es auf dem Bildschirm dargestellt wird.

Doch was ist nun mit unseren Schauspielern, den Objekten, die wir auf dem Monitor haben möchten? Fügen wir ein Textobjekt hinzu:

```
>>> a=clutter.Text()
>>> a.set_font_name("Sans 30")
>>> a.set_colour(red)
>>> a.set_text("Hello World!")
>>> a.set_position(130,100)
>>> stage.add(a)
```

Bei diesem Textobjekt wird hoffentlich klar, was die einzelnen Methoden tun: eine Schriftart auswählen, eine Farbe festlegen, den Text-String definieren sowie die Position auf der Darstellungsfläche bestimmen. In der letzten Zeile wird das actor-Objekt dann tatsächlich auf die Bühne geschickt. Erst an diesem Punkt wird es auf dem Monitor sichtbar.

Nun ist das Objekt aber vorhanden und Sie können damit herumexperimentieren. Versuchen Sie beispielsweise eine andere Position oder Farbe.

PyClutter ist leider dürrtig dokumentiert, doch immerhin bietet Python gute Hilfestellungen. Wenn Sie **dir (a)** eintippen, bekommen Sie die Methoden und Attribute angezeigt, die für ein bestimmtes Objekt verfügbar sind.

Als Nächstes wollen wir ein lauffähiges Skript erstellen. Zuvor sollte jedoch noch eine Sache erledigt werden, nämlich, die Kontrolle über die Anwendung an die Funktion **clutter.main()** zu übergeben. Hierbei möchten wir uns jedoch die Option bewahren, das Programm selber beenden zu können. Dies erreichen wir durch ein Keyboard Event (Tastaturereignis).

Wenn das stage-Fenster aktiv ist, erhält Clutter Signale, wann immer Tasten auf dem Keyboard gedrückt werden. Wir können daher eine Callback-Funktion einrichten, die das Ereignis verarbeitet und die Haupt-Programmschleife abbricht, sobald eine bestimmte Taste gedrückt wird. Daneben können Sie übrigens auch anderen Tasten eine Aktion zuordnen, etwa einen Farbwechsel im Darstellungsbereich.

```
>>> def parseKeyPress(self, event):
...     if event.keyval == clutter.keysyms.q:
```

## Clutter-Versionen

Die Clutter-Bibliothek und demzufolge auch das Python-Modul, das die Bibliothek verwendet, ist vor einer Weile auf Version 1.0 upgedatet worden. Normalerweise rufen Updates ein paar Unstimmigkeiten zwischen alten und neuen Versionen von Software hervor, aber in diesem Fall gibt es fundamentale Unterschiede zwischen dem Code der Versionen vor und nach 0.9. Das PyClutter-Modul und die Clutter-Bibliothek sind vermutlich über das Repository Ihrer Distribution erhältlich, doch wenn Sie diese installieren, achten Sie darauf, dass Sie Version 0.9 oder besser noch 1.0 haben und keine ältere. Andernfalls wird der Code dieses Tutorials nicht funktionieren.



```
...     clutter.main_quit()
...     elif event.keyval == clutter.keysyms.r:
...         self.set_color(red)
...
>>> stage.connect('key-press-event', parseKeyPress)
>>> clutter.main()
```

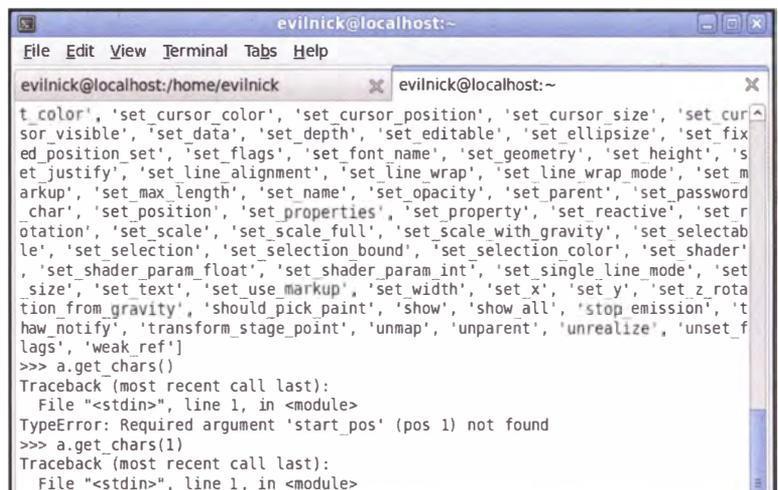
› Nach guter alter Tradition beginnen wir damit, der Welt hallo zu sagen.

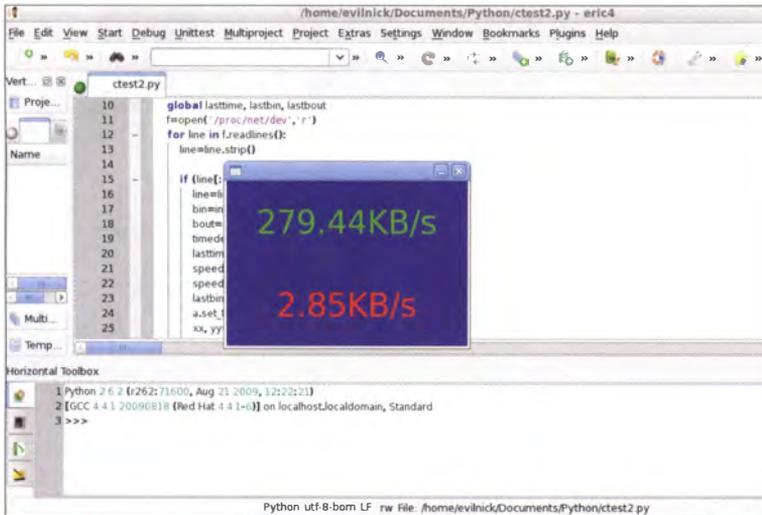
Wenn das Skript in einer interaktiven Python-Shell läuft, beendet die Abbruchfunktion **quit** nicht Python an sich, geschweige denn, dass sie die Anwendung zerstören würde. Die Funktion transferiert lediglich die Kontrolle zurück an die Python-Shell. Die Methode **clutter.main\_quit()** beendet die laufende Anwendung oder zumindest den Clutter-Anteil davon.

## Systemmonitor bauen

Jetzt wissen Sie ein bisschen über die Schnittstelle Bescheid, also können wir uns wie versprochen der Programmierung eines Werkzeugs zuwenden, mit dem Sie die Geschwindigkeit Ihrer Internetverbindung messen können. Bei allen Fragen im Zusammenhang mit Systemstatistiken ist **proc** immer eine gute Anlaufstelle. Das Pseudo-Dateisystem **/proc** enthält einen riesigen Fundus an Informationen über einen

› Die interaktive Shell lädt zum Herumprobieren mit Clutter-Objekten ein.





## ➤ Zahlen. Bunte Zahlen. Das fängt doch schon mal gut an.

Linux-Rechner. Es besteht aus einem Sammelsurium an Dateien, aber wir benötigen an dieser Stelle nur `/proc/net/dev`. Diese Datei listet alle Netzwerkgeräte auf und Sie finden darin Statistiken über empfangene und gesendete Bytes, Pakete, Fehlermeldungen und vieles mehr. Für unseren Code benötigen wir nur die Information über ein- und ausgehende Bytes. Diese wird in Form einer Gesamtsumme gegeben und stellt quasi eine Momentaufnahme zum Zeitpunkt des Öffnens der Datei dar. Um eine Geschwindigkeit ermitteln zu können, müssen wir also die Datei zweimal hintereinander in einem bestimmten zeitlichen Abstand (zum Beispiel 1 Sekunde) öffnen und die Differenz der beiden Werte errechnen.

Nun brauchen wir nur noch eine kleine Funktion, welche die Datei einliest, diese nach der gewünschten Information durchforstet und die Rechenoperation durchführt. Bevor wir die Datei verlassen, speichern wir noch die letzte Zahl, um sie beim nächsten Öffnen von der neuen Zahl abzuziehen zu können. So könnte die Funktion aussehen:

```
devfile=open('/proc/net/dev','r')
for line in devfile.readlines():
    line=line.strip()
    if (line[:4] == 'eth0'):
        line=line[5:].split()
        print line[0], line[8]
```

In der ersten Codezeile liest die Funktion die Datei ein, danach erfolgt die Suche innerhalb der Datei nach der Textzeile, die mit `eth0` beginnt. Vor der Suche müssen die Zeilen allerdings noch gestrippt werden, denn der Output wird mit Leerzeichen aufgefüllt, um gleichmäßige Tabellenspalten zu erzeugen. Sobald die richtige Zeile in der Datei gefunden ist,

entfernen wir den Schnittstellenteil und teilen den String so auf, dass wir jede der Zahlen als Element einer Liste bekommen. Der Wert der empfangenen und gesendeten Bytes befindet sich an Position 0 beziehungsweise 8 der Liste. Hier haben wir sie einfach ausgeben lassen. Tippen Sie den Code ein und schauen Sie sich die Ausgabe an. Zum Schluss müssen lediglich noch die Strings in ganze Zahlen umgewandelt und gespeichert werden.

## Timelines

Falls Sie jetzt einwenden, dass die Ausführung des Programmcodes ja auch eine gewisse Zeitspanne benötigt (auf unserem Entwicklungssystem waren dies 0,0001 Sekunden) und somit das Ergebnis leicht verfälscht, könnten wir noch etwas in dieser Art schreiben:

```
import time
lasttime=1
lastin=0
lastout=0
def getspeed():
    x=open('/proc/net/dev','r')
    for line in x.readlines():
        line=line.strip()
        if (line[:4] == 'eth0'):
            line=line[5:].split()
            bin=int(line[0])
            bout=int(line[8])
            return (bin, bout)

while True :
    z= getspeed()
    timedelta=time.time()-lasttime
    lasttime=time.time()
    sin=(float(z[0]-lastin))/(1024*timedelta)
    sout=(float(z[1]-lastout))/(1024*timedelta)
    print sin, sout
    lastin=z[0]
    lastout=z[1]
    time.sleep(5)
```

Dies baut eine Timing-Funktion ein, sodass die Geschwindigkeit präziser berechnet werden kann, auch wenn sich der Unterschied natürlich nur im Millisekundenbereich bewegt. Falls Sie aber irgendwann die Zeitperiode an einer Stelle der Software ändern möchten, erleichtert Ihnen das obige Skript dies.

Nun müssen wir das Skript noch in die Clutter-Anwendung integrieren. Man könnte es einfach ans Ende des bisherigen Programmcodes setzen, doch dann würde möglicherweise die Clutter-Hauptschleife nie aufgerufen werden. Zwar könnten wir die actor-Objekte immer noch zu jedem beliebigen Zeitpunkt aktualisieren, aber das wäre keine gute Sache. Wesentlich netter wäre es doch, den Objekten ihre Autonomie zu lassen und den Text mithilfe einer animierten Timeline zu kontrollieren.

Eine Timeline ist ein Timer, der bis zu einem bestimmten Wert zählt und dann das programmiertechnische Äquivalent eines Piepstons ausgibt – ein Signal. Das Signal kann aufgefangen und in ein Callback eingespeist werden und Sie können auch andere Parameter hinzufügen. Für unsere Zwecke stellen wir den Timer so ein, dass er eine Funktion aufruft, welche die Netzwerkgeschwindigkeit testet und die beiden actor-Objekte aktualisiert.

Die Timeline ist ein Objekt an sich, doch wenn wir die Verbindung zwischen der Timeline und der Callback-Funktion ausführen, können wir auch unsere Textobjekte weiterleiten,

## Quick-Tipp

Den Überblick über Versionen zu behalten, kann zum Alptraum werden, doch die meisten Module speichern ihre Versionsnummer in `<name>._version_`, was nicht nur für Sie selbst praktisch ist, sondern auch Ihren Anwendungen erlaubt, nach einer kompatiblen Version Ausschau zu halten.

## Clutter-Infos

Clutter ist eine Grafik- und GUI-Bibliothek unter GPL-Lizenz. Es wurde ursprünglich vom OpenHand-Team entwickelt und später an Intel verkauft, das die Entwicklung weiterverfolgt. Clutter ist ein einfaches, schnelles und mächtiges Werkzeug zur Erstellung von 2-D- und 3-D-Grafiken auf diversen Plattformen. Das

Backend besteht im Wesentlichen aus OpenGL, aber mit der Clutter-Bibliothek haben Entwickler ein effizientes und freundliches Tool an der Hand, um grafisch anspruchsvolle Anwendungen zu entwickeln, ohne sich mit den eher technischen OpenGL-Bibliotheken abmühen zu müssen.

sodass die Callback-Funktion diese direkt ändern kann.

Timelines können wie einzelne Threads in einer Multi-thread-Anwendung verwendet werden. Sie sind zwar nicht so flexibel, sind aber einfacher zu managen und erleichtern die Behandlung animierter Objekte, insofern als sie die Animation des Objekts und etwaige andere Interaktionen des Objekts voneinander trennen können.

```
import clutter
import time
lasttime=1
lastbin=0
lastbout=0
black =clutter.Color(0,0,0,255)
red = clutter.Color(255, 0, 0, 255)
green =clutter.Color(0,255,0,255)
blue =clutter.Color(0,0,255,255)
def updatespeed(t, a, b):
    global lasttime, lastbin, lastbout
    f=open('/proc/net/dev','r')
    for line in f.readlines():
        line=line.strip()
        if (line[:4] == 'eth0'):
            line=line[5:].split()
            bin=int(line[0])
            bout=int(line[8])
            timedelta=time.time()-lasttime
            lasttime=time.time()
            speedin=round((bin-lastbin)/(1024*timedelta), 2)
            speedout=round((bout-lastbout)/(1024*timedelta), 2)
            lastbin, lastbout = bin, bout
            a.set_text(str(speedin)+'KB/s')
            xx, yy=a.get_size()
            a.set_position(int((300-xx)/2),int((100-yy)/2) )
            b.set_text(str(speedout)+'KB/s')
            xx, yy=b.get_size()
            b.set_position(int((300-xx)/2),int((100-yy)/2)+100 )
def parseKeyPress(self, event):
    # Parses the keyboard
    #As this is called by the stage object
    if event.keyval == clutter.keysyms.q:
        #if the user pressed "q" quit the test
        clutter.main_quit()
    elif event.keyval == clutter.keysyms.r:
        #if the user pressed "r" make the object red
        self.set_color(red)
    elif event.keyval == clutter.keysyms.g:
        #if the user pressed "g" make the object green
        self.set_color(green)
    elif event.keyval == clutter.keysyms.b:
        #if the user pressed "b" make the object blue
        self.set_color(blue)
    elif event.keyval == clutter.keysyms.Up:
        #up-arrow = make the object black
        self.set_color(black)
    print 'event processed', event.keyval
stage = clutter.Stage()
stage.set_size(300,200)
stage.set_color(blue)
stage.connect('key-press-event', parseKeyPress)
intext=clutter.Text()
intext.set_font_name("Sans 30")
intext.set_color(green)
stage.add(intext)
outtext=clutter.Text()
outtext.set_font_name("Sans 30")
```

## Eine Frage des Timings

Die Clutter-Bibliothek nutzt Timeline-Objekte. Damit kann man fast alles erledigen, was gemacht werden muss, während eine Anwendung gerade läuft. Die Timeline-Objekte sind der Herzschlag Ihres Skripts, sie sorgen dafür, dass alles möglichst gut zusammen läuft. Timelines werden vielfach dafür benutzt, um Animationen und Effekte innerhalb von Clutter zu kontrollieren, aber Sie können sie auch als eigene Interrupts verwenden und damit Routinen aufrufen, und zwar mithilfe von Signalen, die ausgesandt werden, um Ereignisse wie **started** oder **completed** hervorzurufen. Die Signale können auch mit einer Callback-Funktion verbunden werden, um damit noch etwas anderes zu steuern. Dies ist ein kurzes Beispiel-Skript, das Sie in die Python-Shell ein-tippen können:

```
>>> t=clutter.Timeline()
>>> t.set_duration(2000)
>>> t.set_loop(True)
>>> def ping(caller):
...     print caller
...
>>> t.connect('completed',ping)
9L
>>> t.start()
>>> <clutter.Timeline object at
0xb779639c (ClutterTimeline at
0x95b9860)>
```

Hierbei handelt es sich um eine einfache Funktion namens **ping**, die lediglich den Parameter ausgibt, mit dem es angesprochen wurde. Wir verbinden das **completed**-Signal mit der **ping**-Funktion und starten die Timeline. Ohne weitere Interaktion wird die **ping**-Funktion nun alle 2 Sekunden aufgerufen.

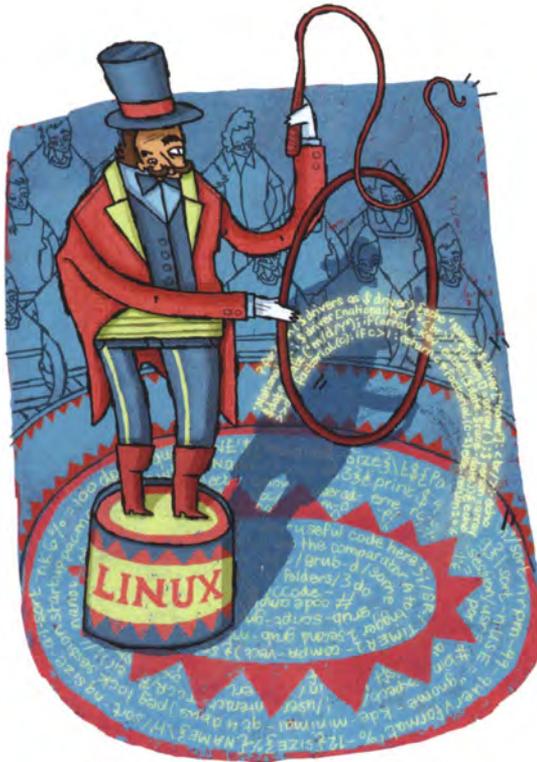
```
outtext.set_color(red)
stage.add(outtext)
stage.show_all()
t=clutter.Timeline()
t.set_duration(5000)
t.set_loop(True)
t.connect('completed', updatespeed, intext, outtext)
t.start()
clutter.main()
```

Hier haben wir nun sämtliche besprochenen Elemente zusammengefügt. Wir haben eine Bühne aufgebaut, sie mit Schauspielern bevölkert und anschließend die Timeline-Objekte von Clutter dazu benutzt, diese sich selbst aktualisieren zu lassen. Nicht schlecht als Einstieg in Clutter! 🍷

➤ Leider bietet die Clutter-Website <http://blogs.gnome.org/clutter/> Python-Anwendern nicht viel Hilfreiches, doch es gibt Hintergrundinfos und eine gute C-Dokumentation.

# Twitter: Tweets

In dieser Übung erstellen wir ein sprechendes Dienstprogramm – halb Python, halb Twitter.



Von vielen als gigantische Ablenkungsmaschine betrachtet, ist Twitter auch eine wunderbare Informationsquelle und überdies lassen sich damit eine Menge technische Dinge anstellen. Dazu muss man sich jedoch zuerst ein wenig mit der API (Programmierschnittstelle) von Twitter vertraut machen. Diese ist ein ziemliches Sammelsurium – es scheint so, als hätte sich die API entwickelt, indem im Laufe der Zeit viele unterschiedliche Lösungswege für ähnliche Aufgabenstellungen gegangen worden wären. Das soll uns aber nicht von der Arbeit abhalten, denn es gibt zahlreiche API-Wrapper (Programmadapter) für Python. Für unsere Zwecke ist der Wrapper Python-Twitter gut geeignet, der auf [github.com/bear/python-twitter](http://github.com/bear/python-twitter) zu bekommen ist.

## Alternative zu Twitter

Bevor wir weitermachen, eine kurze Frage: Haben Sie schon mal von *identi.ca* gehört? Das ist ein Mikro-Blogging-Dienst, ähnlich wie Twitter, nur dass dieser auf freier Software basiert und die Copyleft-Lizenz GPL gewährt. Unser Experiment funktioniert auch mit *identi.ca*. Zwar gibt es kein Python-Modul für den Dienst, aber *identi.ca* benutzt fast die gleiche API wie Twitter, darum kann man eigene Programme leicht anpassen. Man braucht nur die Serververbindungen in der Datei `twitter.py` entsprechend auf *identi.ca* zu ändern. Doch das wäre einmal einen eigenen Artikel wert.

## Mit Twitter verbinden

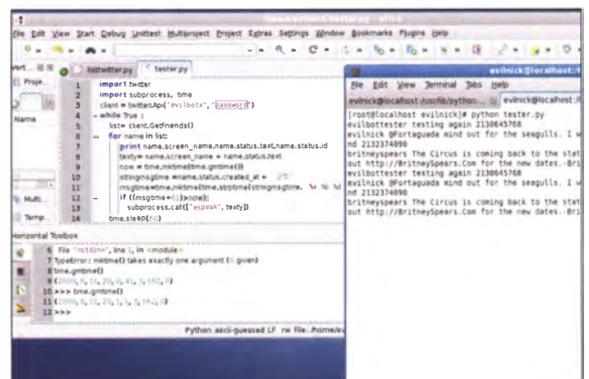
Um mit Twitter arbeiten zu können, benötigen Sie natürlich ein Benutzerkonto bei dem Dienst. Falls Sie bereits eines besitzen, können Sie sich für dieses Experiment auch ein weiteres anlegen, wenn Sie möchten. In dem Fall sollten Sie jedoch zunächst einige Kontakte knüpfen, um die Twitter-Mechanismen auf der Programmierenebene richtig testen zu können. Für dieses Tutorial haben wir das Benutzerkonto „evilbotx“ angelegt und sind damit ein Follower des Piratenschrecks Britney Spears geworden. Öffnen Sie ein Terminal, starten Sie Python mit dem Kommando `python`, und geben Sie danach Folgendes ein:

```
>>> import twitter
>>> client = twitter.Api(username="evilbotx",
password="mypassword")
>>> client.PostUpdate("Hello World!") <twitter.Status object
at 0xb7c2f44c>
```

Dies ist unser normaler Weg der Benutzeranmeldung. Zunächst wird ein Objekt namens `client` geschaffen, welches sich dann mit dem Twitter-Server verbindet und authentifiziert, und zuletzt wird mithilfe einer Objektmethode ein Statusupdate vorgenommen. Schon geht es also los mit dem pythongesteuerten Mikro-Blogging!

Falls Sie ein komplett autonomes System bauen wollten, könnten Sie es hierbei bewenden lassen. Sie könnten diese Funktionalität in ein anderes Skript integrieren und jeder Zeit Tweets absetzen. Allerdings möchten wir gerne über diese Grundfunktion hinausgehen. Der nächste Schritt wäre dann, eine Liste der Twitterer anzulegen, denen wir folgen möchten. Dies ist nicht besonders kompliziert, denn für die meisten Funktionen stehen Methoden zur Verfügung:

```
>>> userlist = client.GetFriends()
>>> for username in userlist: ... print username.screen_name,
username.status.text ...
evilnick @tweeny4 it's hard to beat a poached egg
```



» Wenn ein Bild mehr sagt als tausend Worte, warum höre ich dann nichts? Keine Sorge, das hier kann wirklich sprechen!

# vorlesen lassen

```
serenajwilliams @celebsdontrepy. Of course, I reply.  
britneyspears The Circus is coming back to the states  
-Britney
```

An diesem Stück Code können wir sehen, dass die **GetFriends()**-Methode eine Liste der Klasse **user** zurückbringt. Bei **user** handelt es sich um eine im Twitter-Modul festgelegte Klasse, an die mehrere Dinge wie etwa der Twitter-Name, die Kurzbiografie oder eine URL angehängt werden. All diese Informationen werden direkt von Twitter geliefert, wenn die Objekte erzeugt werden. Einige der Eigenschaften sind:

- » **user.id**: Eindeutige Nutzer-ID
- » **user.name**: Echter Name des Benutzers\*
- » **user.screen\_name**: Twitter-Name des Benutzers
- » **user.description**: Kurzbiografie des Benutzers\*
- » **user.profile\_image\_url**: Link zum Profilbild
- » **user.url**: URL-Angabe des Benutzers, etwa zu dessen Website\*
- » **user.status**: Neuester Status des Benutzers

Die mit dem \* markierten Eigenschaften können auch leer sein, wenn der Benutzer hier Twitter gegenüber keine Angaben gemacht hat.

## Sprachausgabe von Tweets

Wir könnten diese Eigenschaften in unserem Code verwenden, beispielsweise um Bilder für einen grafischen

## Identi.ca

Identi.ca ist ein auf freier und Open-Source-Software aufbauender Mikro-Blogging-Dienst so wie Twitter. Die Inhalte stehen unter der Creative-Commons-Lizenz. Allerdings ist identi.ca in puncto Popularität meilenweit von Twitter entfernt, wobei das

ja auch wieder ein Vorteil sein kann. Prinzipiell lässt sich die in diesem Tutorial gezeigte Vorgehensweise (mit einigen Anpassungen) auch auf identi.ca übertragen, denn dessen API ist sehr eng an die API von Twitter angelehnt.

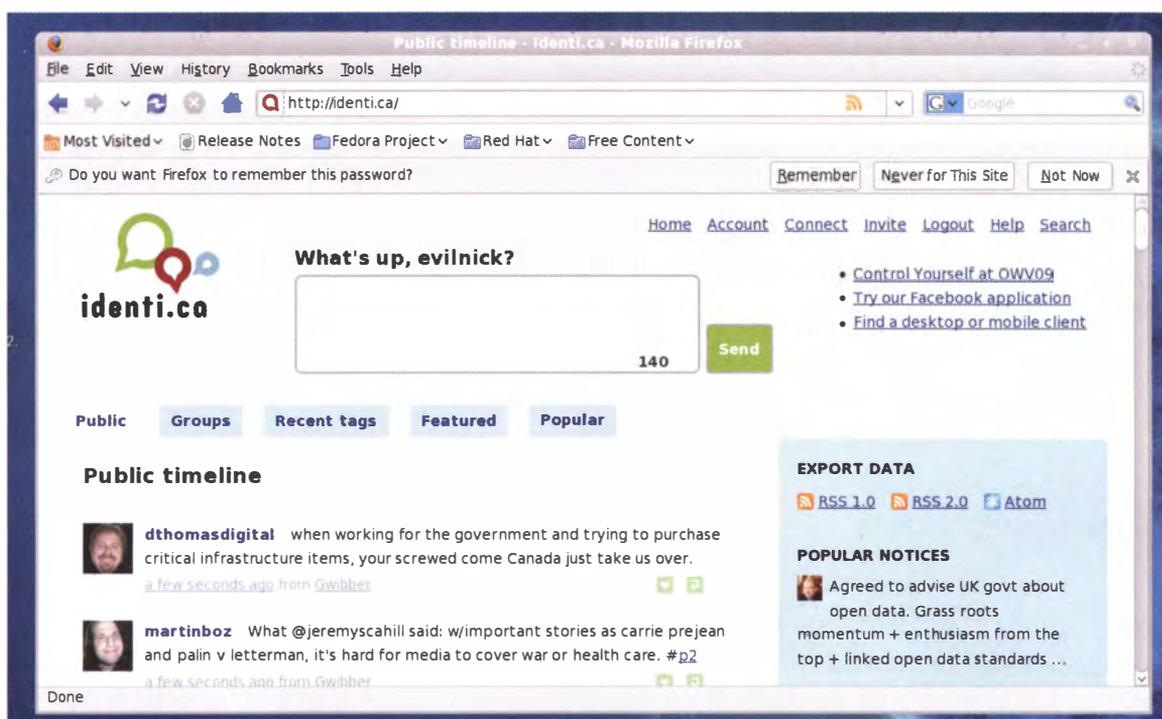
Twitter-Client zu holen oder Twitter-Benutzer nach Interessen zu ordnen.

Aber wie cool wäre es erst, einen audiobasierten Twitter-Client zu haben! Statt die Augen, die man gerade auf seinen Programmcode gerichtet haben sollte, mit dem Lesen von Twitter-Meldungen abzulenken, würde uns eine automatische Stimme die Kurznachrichten vorlesen! Es gibt einige Utilities, die geschriebene Texte in gesprochene Sprache umwandeln, und es gibt sogar ein Sprach-Dispatcher-System für Linux. Vielleicht gehören Festival oder Espeak bereits zum Lieferumfang Ihrer Distribution, ansonsten können Sie eines der Programme über die Repositories nachinstallieren. In unserem Beispiel benutzen wir Espeak, aber der entsprechende Code wäre bei den anderen Programmen sehr ähnlich.

Wir brauchen uns bei dieser relativ einfachen Aufgabenstellung auch nicht mit komplizierten Modulen zu befassen,

## Quick-Tipp

Se können sich jederzeit Informationen zur Funktionalität eines Ihnen nicht vertrauten Moduls anzeigen lassen, wenn Sie im Terminal **help (Modulname)** eingeben, nachdem Sie das Modul importiert haben.



» Identi.ca hat eine ähnliche Funktionalität wie Twitter, ist aber längst nicht so verbreitet. Probieren Sie die Plattform doch einmal aus.

stattdessen nehmen wir das bewährte **subprocess**-Modul. Dieses ermöglicht es, innerhalb von Python Kommandozeilenprogramme aufzurufen. Wir benutzen die **call**-Methode, die lediglich eine Liste der Parameter benötigt, die Sie verwenden möchten. Ein einfaches Beispiel ist:

```
>>> import subprocess
>>> subprocess.call(['espeak', "Hello World!"])
```

Daraufhin sollte Sie eine Computerstimme begrüßen. Falls Sie einen Syntaxfehler angezeigt bekommen, überprüfen Sie, ob Sie möglicherweise bei den Anführungszeichen

etwas Falsches getippt haben. Das letzte Element in der Liste ist ein Textstring innerhalb von doppelten Anführungszeichen

## „Mit Twitter lassen sich eine Menge technische Dinge anstellen.“

chen, der wiederum als Ganzes innerhalb von einfachen Anführungszeichen steht. Der Code entspricht dem Eintippen des Befehls **espeak "Hello World!"** im Terminal. Ein funktionierender Client würde etwa so aussehen:

```
import twitter, subprocess, time
client = twitter.Api("evilbotx", "evilbot")
while True :
    list = client.GetFriends()
    for name in list:
        print name.screen_name, name.status.text, name.status.id
        texty= name.screen_name + name.status.text
        time.sleep(2)
        subprocess.call(["espeak", texty])
    time.sleep(60)
```

In den oben stehenden Programmzeilen stellen wir eine Verbindung mit Twitter her, beginnen eine Endlosschleife und bekommen eine Liste von Kontakten ausgegeben. Eine weitere Schleife verarbeitet die Statusmeldungen und gibt

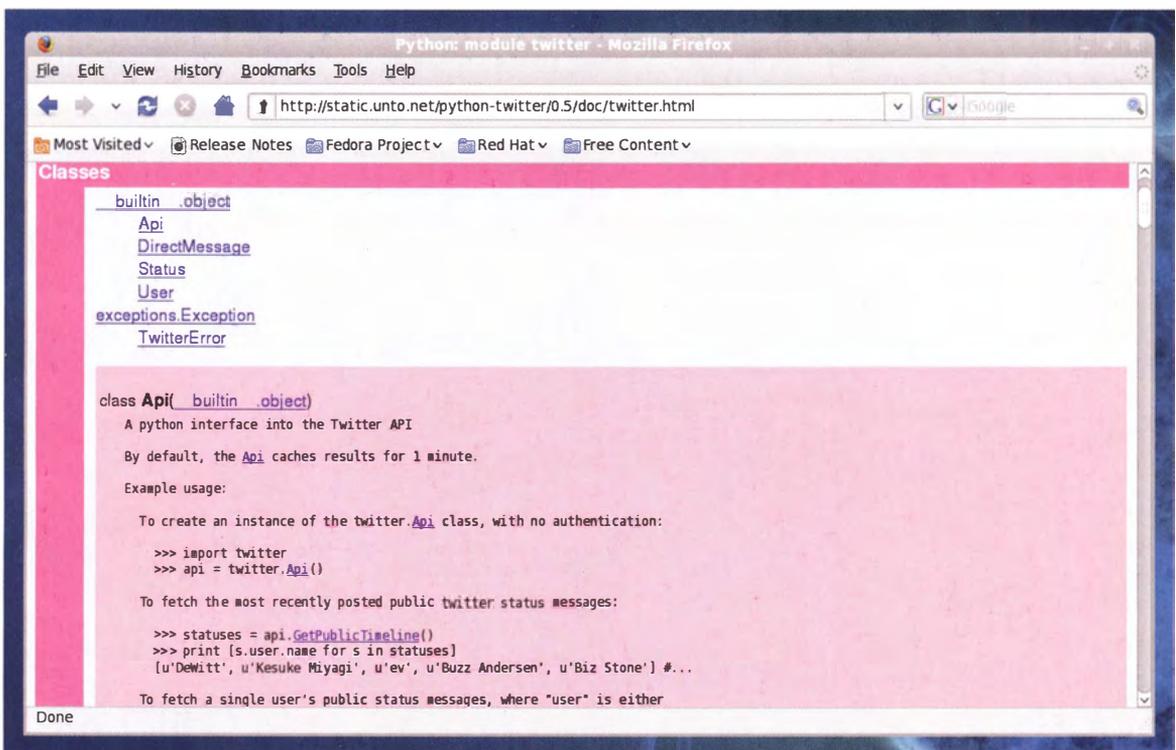
sie aus, wandelt die Informationen, die wir haben möchten, in einen String (eine Zeichenkette) um und benutzt dann **subprocess.call**, um sie an die Sprachausgabe zu schicken. Am Ende des Skripts haben wir eine Verzögerung **time.sleep(60)** eingebaut, damit der Server nicht zu häufig abgefragt wird.

Vielleicht fragen Sie sich, warum wir die Abfrage der Kontaktliste aus dem Hauptloop heraus durchgeführt haben. Dies haben wir so programmiert, weil es die Sache in zweifacher Hinsicht simpler macht: Erstens werden allen Objekte der Klasse **user** automatisch die neuesten Statusmeldungen zugeordnet. Wenn wir die Kontaktliste nur einmal laden würden, müssten wir sie trotzdem bei jedem Durchlauf der Schleife durchgehen, um die Statusmeldungen zu erhalten. Und zweitens erlaubt uns diese Variante, gefahrlos einen zweiten Twitter-Client laufen zu lassen. Jede Änderung, die Sie an Ihrer Kontaktliste vornehmen, wird automatisch von diesem Skript erfasst.

## Zeitprobleme

Nun haben wir einen funktionierenden Programmcode, doch es gibt immer noch ein Problem. Die Statusmeldungen werden hierbei in jedem Fall vorgelesen, ob sie sich nun geändert haben oder nicht. Wir könnten jetzt den Zeitpunkt der Veröffentlichung einer Statusmeldung mit der aktuellen Zeit vergleichen und die Meldung nur dann vorlesen lassen, wenn sie vor weniger als 60 Sekunden gepostet wurde (oder sagen wir 61, um die Zeit des Programmdurchlaufs einzukalkulieren). Leider ist Zeit jedoch relativ. Das Zeitmodul von Python kann Ihnen die derzeitige Epochenzeit angeben (dabei handelt es sich um die Anzahl der vergangenen Sekunden seit dem Anbeginn der Zeit – wobei dies aus Sicht von Unix der 1. Januar 1970, null Uhr ist), aber Twitter gibt die Veröffentlichungszeit eines Tweets in einem Textformat an. Um beides vergleichen zu können,

› Eine vollständige Erläuterung des Moduls Python-Twitter finden Sie auf [static.unto.net/python-twitter/0.5/doc/twitter.html](http://static.unto.net/python-twitter/0.5/doc/twitter.html).



## Chat und Twitter

müssen wir einige Maßnahmen ergreifen. Anscheinend trifft Python-Twitter eine fehlerhafte Annahme bei der Übersetzung, denn bei den Methoden zur Bestimmung des Zeitpunkts des Abrufs und der Veröffentlichung eines Tweets scheint etwas nicht übereinzustimmen. Das verkompliziert die Sache, ist jedoch nicht unlösbar. Die Twitter-API gibt Datum und Uhrzeit als Textstrings im Format **Mon Nov 24 11:46:34 +0000 2014** aus. Das ist an sich kein Problem, da Python dieses Format in sein übliches numerisches Format übersetzen kann und im nächsten Schritt in die Anzahl der Sekunden seit der Unix-Epoche umwandelt. Allerdings gibt das Twitter-Format keine konkrete Zeitzone an. Wenn wir diese mit der koordinierten Weltzeit UTC festlegen möchten, können wir einfach ein **UTC** an das Ende des Strings setzen und Python das Ganze in ein bequemerer numerisches Format parsen lassen. Das Objekt **status** bewahrt die Twitter-Zeitangabe in der Eigenschaft **created-at** auf, also können wir damit das Problem bei der internen Umrechnung umgehen. Die Funktion **time.strptime** verarbeitet Strings zu numerischen Werten in einer Standardform. Um dies zu erreichen, müssen wir der Funktion den String mitteilen sowie einen String, der das Format vorgibt. Der zweite String enthält Anweisungen oder Beschreiber gemäß der Liste von Werten, die das Modul akzeptiert. In unserem Fall sind dies **%a**: der abgekürzte Name des Wochentags, **%b**: der abgekürzte Name des Monats, **%d**: der Tag des Monats, **%H**: die numerischen Stunden, **%M**: die numerischen Minuten, **%S**: die numerischen Sekunden, **%Y**: das numerische Jahr und **%Z**, ein String aus drei Zeichen, der die Zeitzone repräsentiert.

Wie Sie sehen, haben wir den letzten Wert selbst hinzugefügt, sodass er aufgenommen werden kann, wenn Zeitangaben innerhalb von Python bearbeitet werden. Das von Python intern verwendete numerische Format drückt alle diese Daten in Zahlen aus, wie im nachfolgenden Output zu sehen ist:

```
>>> time.strptime('Mon Jun 8 10:51:32 +0000 2009 UTC',
'%a %b %d %H:%M:%S +0000 %Y %Z')
(2009, 6, 8, 10, 51, 32, 0, 159, 0)
```

Diese Zahlen stehen für das Jahr, den Monat, den Tag des Monats, Stunden, Minuten, Sekunden, den Wochentag (0 steht für Montag), den Tag des Jahres sowie Sommerzeit oder nicht. Es ist deshalb sehr wichtig, die Zeitzone hinzuzufügen, weil Python versucht, die Sommerzeit selbstständig zu ermitteln, falls diese Angabe fehlt, was zu falschen Ergebnissen führen kann.

Diese Zeitdarstellung kann danach mit **time.mktime()** wieder in das übliche Unix-Sekundenformat seit der Epoche konvertiert werden. Umfassende Informationen zum **time**-Modul finden Sie auf [docs.python.org/library/time.html](http://docs.python.org/library/time.html).

Unser überarbeiteter Code sieht nunmehr wie folgt aus:

```
import twitter, subprocess, time
client = twitter.Api("evilbotx", "password")
while True :
    list = client.GetFriends()
    for name in list:
        texty= name.screen_name + name.status.text
        now = time.mktime(time.gmtime())
        stringmsgtime =name.status.created_at + ' UTC'
        msgtime=time.mktime(time.strptime(stringmsgtime,
'%a %b %d %H:%M:%S +0000 %Y %Z'))
        if ((msgtime+61)>now):
```

Wenn Sie das Tutorial auf Seite 46-49 durchgearbeitet und sich ein paar Chatbots zugelegt haben, dann könnten Sie die entsprechenden Skripte mit Ihren Twitter-Experimenten verknüpfen. Zum Beispiel, indem Sie Ihren Chatstatus aus Ihrem letzten Twitter-Posting generieren.

```
import
xmpp, twitter, twituser="foo1", twitpass="foo2", jabberuser="bar1@something",
jabberpass="bar2", twit=twitter.Api(username=twituser, password=twitpass)
text=twit.GetUser(twituser).status.text, jid=xmpp.protocol.JID(jabberuser)
jab=xmpp.Client(jid.getDomain(), debug=[])
jab.connect()
jab.auth(jid.getNode(), jabberpass)
jab.sendInitPresence()jab.send(xmpp.Presence(status = text , show = "chat" ,
priority = '1')
//end//
```

Platzieren Sie den gesamten Code in einer Schleife mit einer angemessenen Verzögerungszeit. Die Nutzernamen und Passwörter müssen Sie natürlich noch individuell anpassen.

```
subprocess.call(["espeak", texty])
time.sleep(60)
```

Jetzt haben Sie also in einigen wenigen Codezeilen einen funktionierenden audiobasierten Twitter-Client geschrieben! Ein Problem bleibt allerdings noch: Falls Sie einer größeren Anzahl von Twitterern folgen möchten, könnte es schnell anstrengend werden, wenn ständig neue Postings vorgelesen werden. Möglicherweise wollen Sie die Liste derjenigen Kontakte, bei denen Meldungen vorgelesen werden sollen, ja verkleinern. Dann müssten Sie noch ein paar kleine Änderungen vornehmen:

```
import twitter, subprocess, time
client = twitter.Api("evilbotx", "password")
list = ['evilnick', 'evilbottester', 'tweeny4']
while True :
    for item in list:
        name=client.GetUser(item)
        texty= name.screen_name + name.status.text
        now = time.mktime(time.gmtime())
        stringmsgtime =name.status.created_at + ' UTC'
        msgtime=time.mktime(time.strptime(stringmsgtime,
'%a %b %d %H:%M:%S +0000 %Y %Z'))
        if ((msgtime+61)>now):
            subprocess.call(["espeak", texty])
    time.sleep(60)
```

In dieser Version des Skripts geht die innere Schleife die Liste durch und ruft die **GetUser()**-Methode für jeden Twitter-Namen auf. Dies führt zu einer Rückmeldung über **user**-Objekte mit bestimmten Eigenschaften, zu denen die neueste Statusmeldung gehört. Nun bekommen Sie nur noch die neuen Tweets Ihrer Favoriten vorgelesen.

## Ausblick

Es könnte eine nützliche Sache sein, eine grafische Benutzeroberfläche zur Hand zu haben, wenn Sie schnell Ihren eigenen Status aktualisieren möchten. Wenn Sie dazu PyQt, wxWidgets oder eine andere GUI verwenden, müssten Sie lediglich einen Text-Input mit maximal 140 Zeichen erstellen und eine Methode zum Posten von Statusmeldungen an die Betätigung der Eingabetaste koppeln.

Eine andere Idee wäre es, Ihren Server seinen freien Speicherplatz twittern zu lassen oder das Skript der Software AmaroK anzufügen und es twittern zu lassen, welche Musik Sie gerade hören.

# Python: Hacken Sie Minecraft

Basteln Sie mit Python auf dem Pi nach Herzenslust an Minecraft herum.

Von dem extrem populären Computerspiel Minecraft gibt es auch eine offizielle Version für den Raspberry Pi. Bei der Software namens Minecraft: Pi Edition handelt es sich um eine abgespeckte Version der beliebten Pocket Edition des Spiels, die als solche aller lebensgefährlichen Spielelemente entbehrt, dafür jedoch so viele Blöcke enthält, dass Sie Bauklötze staunen werden – und drei Typen Setzlinge, damit Ihnen das dazu nötige Holz nicht ausgeht.

Viel Material also, um kreativ zu werden, aber all die Klickerei artet doch oft in Arbeit aus – mit der eleganten Python-API können Sie in dieser Edition mit nur wenigen Zeilen Code kantige Versionen Ihrer wildesten Träume wahr werden lassen.

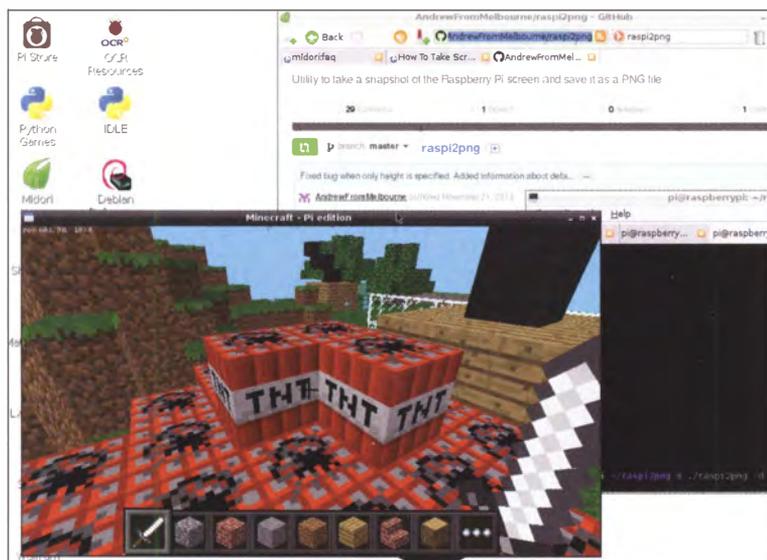
Wenn Sie Ihren Raspberry Pi bereits eingerichtet haben, laden Sie die neueste Version von der Seite [pi.minecraft.net](http://pi.minecraft.net) in Ihr Home-Verzeichnis. Raspbian wird von den Machern offiziell empfohlen, also ziehen wir es ebenfalls anderen Distributionen vor. Minecraft benötigt den X-Server; wenn Sie also bevorzugt in die Konsole booten, geben Sie **startx** ein. Starten Sie LXTerminal und extrahieren und öffnen Sie den Inhalt des Archivs:

```
$ tar -xvzf minecraft-pi-0.1.1.tar.gz
$ cd mcpi
$ ./minecraft-pi
```

Sehen Sie, wie flüssig es läuft? Links oben sehen Sie Ihre x-, y- und z-Koordinaten, die sich beim Durchstreifen der Umgebung stets ändern. Die x- und z-Achse laufen parallel zum Boden, die y-Achse markiert die Höhe. Jeder Block (oder Voxel, um im Jargon zu bleiben), der Teil der Landschaft ist, wird von Integer-Koordinaten und einem BlockType bestimmt. Der „Boden“ hat im Prinzip keine Tiefe, sondern besteht sozusagen aus Kacheln. Freier Raum hat den BlockType AIR und es gibt noch etwa 90 andere, etwas handfestere Substanzen, unter anderem Schmankerl wie GLOWING\_OBSIDIAN und TNT. Die Koordinaten Ihrer Spielfigur haben im Unterschied zu den Blöcken einen Dezimalteil, da Sie sich innerhalb von AIR-Blöcken frei bewegen können.

Die API ermöglicht es nun, in eine laufende Minecraft-Partie einzugreifen und Spieler und Terrain nach größtmöglichstem Gusto anzupassen. Dazu muss zunächst die Programm-Bibliothek kopiert werden, damit die frische Minecraft-Installation nicht vor Übermut kaputtgeht. Wir legen einen Ordner namens `~/picraft` für unsere Bastelei an und schieben alles zur API Gehörige in `~/picraft/minecraft`. Öffnen Sie LXTerminal und geben Sie Folgendes ein:

```
$ mkdir ~/picraft
$ cp -r ~/mcpi/api/python/mcpi ~/picraft/minecraft
```



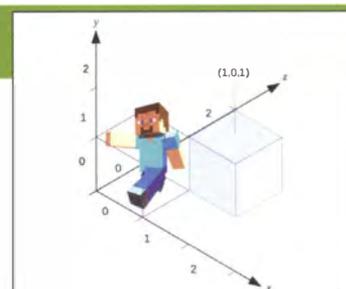
➤ Versuchen Sie das nicht zu Hause! Oder etwa doch?

## Ey, wo is' mein Steve?

Hier ist unsere furchtlose Spielfigur (Steve) in einem Block bei (0,0,0). Er kann sich darin bewegen, und ein paar Schritte auf der x- und z-Achse bringen ihn zu dem blau schattierten Block. Auf dieser kurzen Reise steht er zeitweise in mehr als einem Block; die Funktion `getTilePos()` der Minecraft-API gibt stets denjenigen aus, der am meisten Steve enthält.

Sobald nun aber Standardkonzepte wie Linien oder Polygone aus dem euklidischen Raum in Blöcke übertragen werden sollen, wird es raffiniert.

Eine 2D-Version dieses Problems taucht beim Rendern von Vektorgrafik auf. Möchten Sie auf dem Bildschirm eine Linie zwischen zwei Punkten ziehen, die nicht exakt horizontal oder vertikal verläuft, muss entschieden werden, welche Pixel eingefärbt werden sollen. Als Erster schlug Jack Elton Bresenham im Jahr 1965 eine Lösung dafür vor und wir werden diesen klassischen Algorithmus in unserem nächsten Tutorial für drei Dimensionen generalisieren (ab Seite 60).



➤ Durch isometrische Projektion passt Minecraft auf diese Seite.

Nehmen wir also unsere ersten Minecraft-Modifikationen vor. Wir beginnen mit einer interaktiven Python-Sitzung, die neben Minecraft läuft: Öffnen Sie einen neuen Tab in LXTerminal, starten Sie Minecraft, betreten Sie einen Level und gehen Sie per Alt+Tab zurück in das Terminal, um Python im neuen Tab zu öffnen. Machen Sie dort Folgendes:

```
import minecraft.minecraft as minecraft
import minecraft.block as block
mc = minecraft.Minecraft.create()
posVec = mc.player.getTilePos()
x = posVec.x
y = posVec.y
z = posVec.z
mc.postToChat(str(x)+' '+str(y)+' '+str(z))
```

Siehe da, unser aktueller Standpunkt erscheint kurz auf dem Bildschirm (falls nicht, ist Ihnen ein Fehler unterlaufen). Diese Koordinaten zeigen den Block an, auf dem Ihre Spielfigur momentan steht, und haben dementsprechend keine Dezimalstelle. Verglichen mit den Koordinaten oben links sind sie lediglich deren abgerundete Integer-Versionen (-1.1 wird beispielsweise zu -2). Die Koordinaten der Spielfigur sind über **mc.player.getPos()** verfügbar, **getTilePos()** ist einerseits überflüssig, es spart aber drei Ganzzahl-Konvertierungen, also verwenden wir es. Die API hat eine nützliche Klasse namens **Vec3**, die mit dreidimensionalen Vektoren wie der Position unserer Spielfigur umgehen kann. Sie enthält normale Vektor-Operationen wie Addition und skalare Multiplikation, aber auch einige exotischere Dinge, die uns später noch weiterhelfen werden.

Wir können auch herausfinden, worauf unsere Spielfigur gerade steht. Kehren Sie zurück zu Python und geben Sie ein:

```
curBlock = mc.getBlock(x, y - 1, z)
mc.postToChat(curBlock)
```

Hier gibt **getBlock()** eine Zahl aus, die die Art des Blocks angibt: 0 ist Luft, 1 ist Stein, 2 ist Gras, und alle anderen Blockarten können Sie in der Datei **block.py** im vorhin angelegten Ordner **~/picraft/minecraft** finden. Wir ziehen 1 vom y-Wert ab, weil wir wissen möchten, was unter unseren Füßen vor sich geht – **getBlock()** auf den momentanen Ort der Spielfigur angewendet, sollte immer 0 ausgeben, denn ansonsten wären wir entweder eingemauert oder ertrunken.

Wie immer gilt, dass Code im Python-Interpreter eine schöne Spielerei ist, aber verantwortungsvolle Programmierer packen ihn in eine Datei: Erstellen Sie die Datei **~/picraft/gps.py** mit folgendem Code:

```
import minecraft.minecraft as minecraft
import minecraft.block as block
mc = minecraft.Minecraft.create()
oldPos = minecraft.Vec3()
while True:
    playerTilePos = mc.player.getTilePos()
    if playerTilePos != oldPos:
        oldPos = playerTilePos
        x = playerTilePos.x
        y = playerTilePos.y
        z = playerTilePos.z
        t = mc.getBlock(x, y - 1, z)
```



```
mc.postToChat(str(x) + ' ' + str(y) + ' ' + str(z) + ' ' + str(t))
```

Starten Sie Minecraft und betreten Sie einen Level, öffnen Sie das Terminal und lassen Sie Ihr Programm laufen:

```
$ python gps.py
```

Jetzt sollten die Koordinaten und der gerade betretene BlockType angezeigt werden, während Sie umherlaufen. Sobald Sie alle BlockTypes auswendig können (Scherz), dürfen Sie das Python-Programm mit Strg + C schließen.

Wir haben nun einige der „passiven“ Möglichkeiten der API erkundet, aber erst in Kombination mit konstruktiven (oder destruktiven) Optionen werden diese wirklich lustig. Starten Sie also erneut Minecraft und eine Python-Sitzung, importieren Sie die Minecraft- und Blockmodule und richten Sie das Objekt **mc** ein:

```
posVec = mc.player.getTilePos()
x = posVec.x
y = posVec.y
z = posVec.z
for j in range(5):
```

```
    for k in range(x - 5, x + 5)
        mc.setBlock(k, j, z + 1, 246)
```

Voilà, eine Wand aus leuchtendem Obsidian, 10 x 5 Blöcke groß, wurde just neben Ihnen errichtet. Wir können auch durch eine Verwandlung in Luft Blöcke zerstören und so einen Tunnel in unsere Obsidianwand graben:

```
mc.setBlock(x, y, z + 1, 0)
```

Vorausgesetzt Sie haben sich seit der letzten Eingabe nicht bewegt.

Im Artikel auf den nachfolgenden Seiten werden wir bauen und zerstören, mit der Physik experimentieren und ihre Regeln neu schreiben und in unserer 256 x 256 x 256 großen Welt ordentlich auf den Putz hauen. Wie wäre es, zur Überbrückung ein wenig mit **mc.player.setPos()** zu spielen? Teleportieren macht schließlich Spaß.

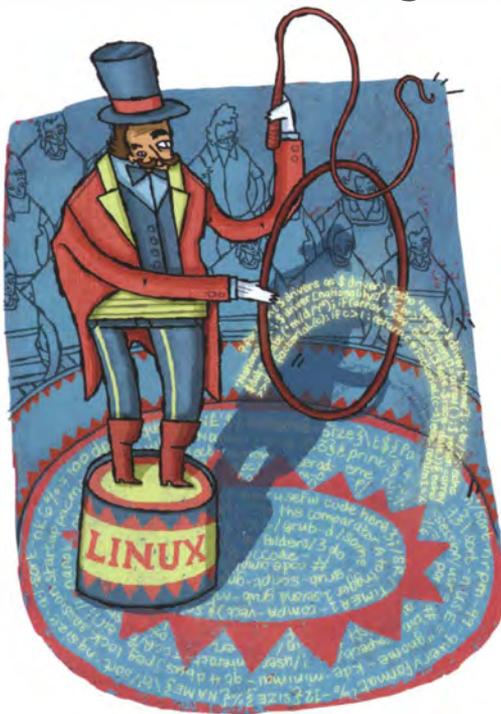
› Machen Sie sich jegliche Art von unwahrscheinlichen Strukturen zu eigen.

## Quick-Tipp

Martin O'Hanlons Website **stuffabout-code.com** zeigt mit schönen Beispielen, wozu die API fähig ist.

# Minecraft: Das Pi-Katapult

Bauen Sie Ihr Traumhaus und jagen Sie es mit pyrotechnischem Code, einer Ladung TNT und einer antiken Belagerungswaffe in die Luft.



Jetzt da wir grundsätzlich mit der API vertraut sind, können wir unserer Kreativität freien Lauf lassen. Ein Haus zu bauen ist schwierig, oder? Aber nicht doch! Mit ein paar Zeilchen in Python errichten Sie Ihr Traumhaus – vorausgesetzt, es ist ein eher schlichtes Traumhaus in Kastenbauweise. Wer von mehr träumt, benötigt einfach nur ein wenig mehr Code und hat garantiert nie Ärger mit Baugenehmigung, Kommunalanschluss, Grunderwerbsteuer oder Hügelgräbern aus der Jungsteinzeit in der Baugrube (außer, Sie haben sie vorher selbst dort hinein gebaut).

Regen ist in Minecraft Pi vernachlässigbar, darum genügt ein Flachdach vollkommen. Zum Spatenstich definieren wir zwei Ecken für unser Haus: **v1** ist ein Block auf der x-Achse direkt neben uns, einen Block nach oben versetzt, und **v2** befindet sich in beliebiger wohnlicher Entfernung.

```
pos = mc.player.getTilePos()
v1 = minecraft.Vec3(1,1,0) + pos
v2 = v1 + minecraft.Vec3(10,4,6)
```

Erstellen wir nun einen massiven Steinquader zwischen den Eckpunkten und höhlen ihn mithilfe eines Luftquaders aus:

```
mc.setBlocks(v1.x,v1.y,v1.z,v2.x,v2.y,v2.z,4)
mc.setBlocks(v1.x+1,v1.y,v1.z+1,v2.x-1,v2.y,v2.z-1,0)
```

Sehr gut, nur leider führt der einzige Ein- und Ausgang durch das recht großzügig bemessene Oberlicht, und ein Holzboden wäre auch gemütlicher. Auf flachem Terrain können Sie gut sehen, dass Ihre vier Wände eine Blockhöhe über der Erde schweben, was Raum für einen Boden lässt. Haben Sie in hügeliges Gebiet gebaut, könnte sich Ihr Haus in extremer Hanglage oder erheblichem Schwebезustand befinden, aber keine Sorge – um Tiefbau und Zugeständnisse an die Schwerkraft kümmern wir uns später. Verlegen wir also unser rustikales Hartholzparkett:

```
mc.setBlocks(v1.x,v1.y-1,v1.z,v2.x,v1.y-1,v2.z,5)
```

Auf ähnliche Weise werden die Fenster eingebaut:

```
mc.setBlocks(v1.x,v1.y+1,v1.z+1,v1.x,v1.y+2,v1.z+3,102)
mc.setBlocks(v1.x+6,v1.y+1,v1.z,v1.x+8,v1.y+2,v1.z,102)
mc.setBlocks(v2.x,v1.y+1,v1.z+1,v2.x,v1.y+2,v1.z+3,102)
mc.setBlocks(v1.x+2,v1.y+1,v2.z,v1.x+4,v1.y+2,v2.z,102)
```

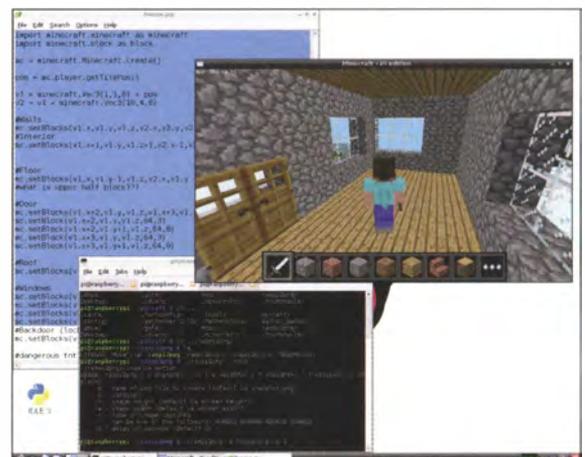
Das Dach besteht aus dem speziellen Halblock 44, der verschiedene Varianten bietet. Der **blockType** kann so angepasst werden, dass er zu unserem Holzboden passt:

```
mc.setBlocks(v1.x,v2.y,v1.z,v2.x,v2.y,v2.z,44,2)
```

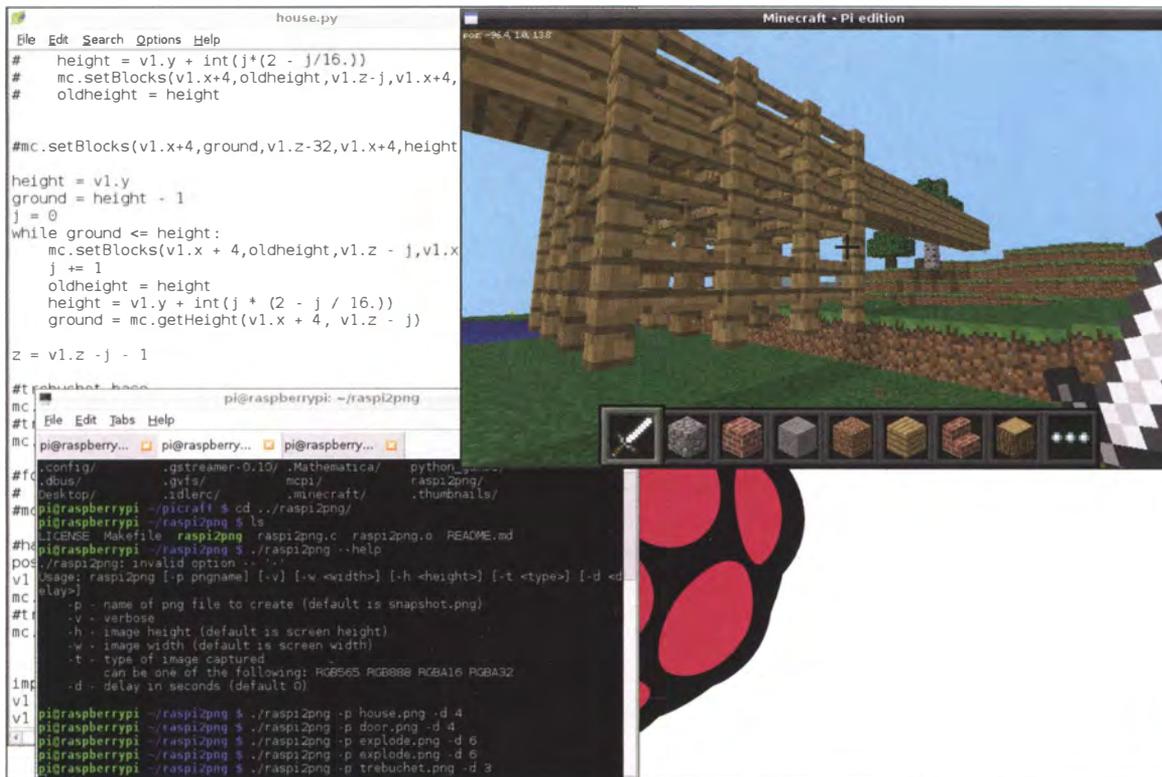
Türen sind etwas komplexer, wie Sie im Kasten auf Seite 62 nachlesen können, aber folgende drei Zeilen tun es vorerst auch:

```
mc.setBlocks(v1.x+2,v1.y,v1.z,v1.x+3,v1.y,v1.z,64,3)
mc.setBlock(v1.x+2,v1.y+1,v1.z,64,8)
mc.setBlock(v1.x+3,v1.y+1,v1.z,64,9)
```

Nun da der Neubau des trauten Eigenheims endlich abgeschlossen ist, geht es daran, seinen Einsturz in die Wege zu



› Ein Haus. Lassen Sie es uns in die Luft jagen.



› Noch macht es nicht viel her, aber Sie werden schon sehen..

leiten. Wir haben bereits festgestellt, dass TNT explodiert, wenn man ihm einen sanften Schwertstreich (oder auch nur einen Schubser) verpasst. Das Haus per **setBlock** mit scharfem TNT zu füllen, wäre aber viel zu einfach, das geht auch eleganter. Mit einem Katapult beispielsweise.

Anstatt ein fliegendes Projektil zu simulieren, empfinden wir seine parabolische Flugkurve mit schwebendem TNT nach. Diese Kurve an ihrem Ursprung zu entzünden, löst eine herrliche Kettenreaktion aus, die auf ihrem Höhepunkt ein gewaltiges Loch in Ihr Häuschen reißt. Klären wir zunächst einige Grundsätze zweidimensionaler Mechanik: In Abwesenheit von Reibungswiderstand nimmt ein Projektil einen parabolischen Kurs, der von der Ausgangsgeschwindigkeit, dem Austrittswinkel und der Gravitationsbeschleunigung, die auf der Erde etwa  $9,81 \text{ m/s}^2$  beträgt, bestimmt wird.

Zum Einstieg tüfteln wir mit diesen Konstanten eine Horizontalabstand von exakt 32 Blöcken bei einer Maximalhöhe von 16 Blöcken für den explosiven Bogen aus. Wäre ein Block ein Meter, dann käme diese Spielerei einer Mündungsgeschwindigkeit von knapp  $18 \text{ m/s}$  nahe, bei einem Winkel von  $60 \text{ Grad}$ . Da wir weiterhin zweidimensional denken, muss dieser Bogen einfach entlang der z-Achse bis zur x-Koordinate vor unserer Haustür gespannt werden. Das alles beschreibt die einfache Formel  $y = z(2 - z/16)$ , die wir folgendermaßen verwenden:

```
for j in range(33):
    height = v1.y + int(j*(2 - j/16.))
    mc.setBlock(v1.x+4,height,v1.z-j,46,1)
```

Die letzte Zeile schaltet das TNT scharf – ein Schwertstreich, und Sie genießen ein Feuerwerk. Oder auch nicht: Die Explosionen lassen den Pi ganz schön rechnen und unterbrechen obendrein die Kettenreaktion, weil einzelne TNT-Blöcke herunterfallen. Nein, das Haus darf nicht verschont werden, nutzen wir also folgenden Code:

```
height = v1.y
ground = height - 1
j=0
while ground <= height:
    mc.setBlocks(v1.x + 4,oldheight,v1.z - j,v1.x + 4,height,v1.z - j,46,1)
    j += 1
    oldheight = height
    height = v1.y + int(j * (2 - j / 16.))
    ground = mc.getHeight(v1.x + 4, v1.z - j)
```

Das verhindert sowohl Lücken in der Parabel als auch nur halb detonierte TNT-Regenbögen. Letzteres erledigt die Funktion **getHeight()**, die den Bogen an jedem Punkt mit dem Gelände abgleicht und so bauen aufhört, sobald der Boden erreicht ist. Dabei muss **getHeight()** aufgerufen werden, bevor der letzte Block TNT platziert wird, da die Höhe des Levels vom höchsten Objekt bestimmt wird, das nicht Luft ist – auch wenn selbiges gerade schwebt.

Wenn diese Konstruktion zu groß für das Minecraft-Level ist, können Sie einfach ein weiteres Haus in besserer Lage bauen oder aber im obigen Code **v1.z - j** durch **max(-116,v1.z-j)** ersetzen, um einen explosiven Totempfahl am Rand des Levels zu errichten. Der Pfad des Projektils ist nun komplett, es folgt die mächtige Belagerungswaffe:

```
z = v1.z - j - 1
mc.setBlocks(v1.x + 3, oldheight, z + 10, v1.x + 6, oldheight + 2, z + 7,85)
mc.setBlocks(v1.x + 4, oldheight + 2, z + 12, v1.x + 4, oldheight + 2, z + 1,5)
```

Bis dato haben wir alles an einer Achse ausgerichtet. Das Haus (sofern es noch steht) blickt in die negative Richtung der z-Achse, also quasi nach Süden; unsere Parabel orientiert sich ebenfalls an dieser Achse. Wir könnten natürlich alles um  $90 \text{ Grad}$  drehen und der Code bliebe in etwa gleich

## Quick-Tipp

Inspiration für den Katapult-Code waren der fantastische Martin O'Hanlon und seine Pi-Projekte, die Sie auf seiner Website **stuff-aboutcode.com** finden.



› Schweizer Käse! (Ihr Haus könnte nach diesem Tutorial eine Renovierung vertragen.)

(einige sachkundige Änderungen von  $x$ ,  $y$  und  $z$  und +/- genügen), nur das Haus sähe wohl ein wenig seltsam aus. Richtig kompliziert wird es, wenn wir aus dem Joch des Gitternetzes und der rechten Winkel ausbrechen und mit selbst gewählten Winkeln arbeiten möchten. Die Schwierigkeit ist hierbei besonders, aus Blöcken mit festgelegter Orientierung eine annähernd gerade Linie zu erzeugen.

Die dreidimensional einsetzbare Funktion **drawline()** wird sich dafür bald als unverzichtbar erweisen, da sie erlaubt, diverse Konfigurationen vom Parallelepiped bis zum Pentagramm zu erstellen. Möglich macht dies eine 3-D-Version des klassischen Bresenham-Algorithmus. Das github des Pi-Gurus Martin O'Hanlon enthält mehrere sehr schöne Projekte für Minecraft Pi, darunter auch eine mächtige Kanone, die für dieses Tutorial Modell stand. O'Hanlon bietet eine komplette Python-Klasse an, die auch den eben genannten 3-D-Algorithmus enthält, aber sobald Sie die 2-D-Version verstanden haben, können Sie ihr Wissen leicht generalisieren.

Angenommen, wir befinden uns in einem Minecraft-Flachland, wo wir eine Linie auf dem Koordinatensystem  $xy$  zwischen den Punkten  $(-2,-2)$  und  $(4,1)$  ziehen möchten. Die Formel für die Linie wäre  $y = 0.5x - 1$ . Der Algorithmus verlangt, dass das Gefälle der Linie zwischen 0 und 1 liegt, was in diesem Fall erfüllt ist. Für eine Linie mit einer anderen Neigung können wir die Achsen vertauschen. Ein Problem des Algorithmus ist jedoch, dass er nur einen Pixel (also Block) per Spalte, aber mehrere per Reihe füllen kann. Während wir also Pixel für Pixel auf der  $x$ -Achse füllen, bleibt die

$y$ -Koordinate entweder gleich oder erhöht sich um 1. Ein unbedarfter Versuch in Python könnte so aussehen:

```
dx = x1 - x0
dy = y1 - y0
y = y0
error = 0
grad = dy/dx
for x in (x0,x1):
    plot(x,y)
    error = error + grad
    if error >= 0.5:
        y += 1
    error -= 1
```

Dabei ist **plot()** eine imaginäre Plotter-Funktion und liegt **grad** zwischen 0 und 1. So erhöhen wir  $y$  immer dann, wenn die Fehlerbedingung erfüllt ist, und das Bild, das Sie hier sehen, resultiert daraus.

Bresenham löste dies, indem er alle Kalkulationen auf Integerberechnungen herunterbrach, die auch für die Hardware der 1960er verdaulich waren. Heutzutage ist blitzschnelle Gleitkommaberechnung kein Problem mehr, aber diese clevere Idee verdient dennoch Anerkennung. Die Gleitkommavariablen **grad** und **error** entstehen durch Teilen durch **dx**. Wenn wir also alles mit dieser Menge multiplizieren, dann können wir loslegen.

Der Sprung in die dritte Dimension ist nun gar nicht mehr so schwierig oder abstrakt. Zunächst müssen wir die Hauptachse finden (diejenige mit der größten Koordinatenveränderung), dann alles entsprechend drehen und immer entlang der Hauptachse die Linie zeichnen, einen Block nach dem anderen, und dabei die Koordinaten der Nebenachsen nötigenfalls anpassen. Diese Koordinatenanpassungen, die wir in der Variable **ds** speichern, müssen wir immer im Auge behalten. Die Funktion **ZSGN()** gibt jeweils 1, -1 or 0 aus, wenn ihr Rückgabewert positiv, negativ oder Null ist; die Programmierung überlassen wir als kleine Aufgabe Ihnen selbst. Die Hilfsfunktion **minorList(a,j)**, die die Liste **a** unter Entfernung des  $j$ -sten Eintrags ausgibt, kommt dabei gehäuft zum Einsatz. Dank Lambda-Funktionen und List Slicing genügt ein Einzeler:

```
minorList = lambda a,j: a[:j] + a[j + 1:]
```

Unsere Funktion **getLine()** verwendet zwei Eckpunkte, die von einer Liste mit drei Elementen dargestellt werden, um eine Liste aller Eckpunkte der so beschriebenen 3-D-Linie auszugeben. Auch dies basiert auf O'Hanlons Code. Der erste Teil initialisiert die Eckpunktliste und berechnet sie für den Fall, dass die Eckpunkte identisch sind. Diese Linie ist nur ein einziger Block:

## Quick-Tipp

Alles im Interpreter zu programmieren, kann zu frustrierenden Kopierfehlern führen. Besser lagern Sie alles in eine Datei namens `house.py` aus, die Sie mit **python house.py** ausführen können, während Minecraft läuft.

## Details zu Doppelflügeltüren

Der Einbau der Haustüren ist unsere erste Begegnung mit dem zusätzlichen Parameter **blockData**. Dies ist ein Integer zwischen 0 und 15, der weitere Blockeigenschaften wie etwa die Farbe von Wolle oder die Scharfschaltung von TNT definiert. Unsere Tür belegt vier Blöcke auf der  $x$ -Achse. Sie ist ein Stück weit nach innen ins Mauerwerk eingesetzt, geschlossen und hat die Griffe in der Mitte. All diese Eigenschaften werden durch Bits in **blockType** definiert. Von rechts nach links werden die vier Bits von 0 bis 3 durchnummeriert; Sie funktionieren nach der Little-Endian-Schreibweise, in der 8 in Binär-code als 1000 bezeichnet wird. Bit 3 zeigt an,

ob der Block zur oberen Hälfte einer Tür gehört. Ist dem so, muss nur noch Bit 0 beachtet werden – er zeigt an, wo sich Griffe und Türangeln befinden. Die oberen Hälften von Türen haben demnach `blockType` 8 oder 9.

Für die untere Hälfte gibt es folgende Bit-Einstellungen:

- Bit 3** .....aus
- Bit 2** .....Tür ist offen
- Bit 1** .....Tür ist nach innen versetzt
- Bit 0** .....Ausrichtung (aus= $x$ , an= $z$ )

Die Oberteile müssen jeweils nach den unteren Hälften gesetzt werden, da Sie deren Eigenschaften übernehmen.



› Türen helfen zuverlässig, Klaustrophobie und einen einsamen Tod zu vermeiden.

```
def getLine(v1, v2):
    if v1 == v2:
        vertices.append([v1])
```

Danach wird es ein bisschen unschön. Wir legen die erwähnte Zeichenliste **ds** und eine Liste absoluter Differenzen **a** (multipliziert mit 2) an. Der Ausdruck **idx = line** ist eher schlechter Stil – wir brauchen die Hauptachse, daher der Index der maximalen Eingabewerte in **a**. Die Methode **index()** kombiniert mit **max** liest die Liste doppelt aus; bei so einer kurzen Liste soll uns das aber nicht kümmern, es sieht einfach schöner aus. Die Hauptkoordinaten bezeichnen wir als **X** und **X2**. Die Liste **s** ist eine Umformung von **ds**, die mit den Hauptkoordinaten beginnt, zusätzlich gibt es einige Listen für Fehlerprotokolle. Die Variable **aX** bezieht sich auf das Zeichen der Koordinatenänderung entlang der Hauptachse.

```
else:
    ds = [ZSGN(v2[j] - v1[j]) for j in range(3)]
    a = [abs(v2[j]-v1[j]) << 1 for j in range(3)]
    idx = a.index(max(a))
    X = v1[idx]
    X2 = v2[idx]
    delta = a[idx] >> 1
    s = [ds[idx]] + minorList(ds,idx)
    minor = minorList(v1,idx)
    aminor = minorList(a,idx)
    dminor = [j - delta for j in aminor]
    aX = a[idx]
```

Nun ist alles bereit für den Einstieg in die Hauptschleife, in der Eckpunkte hinzugefügt, die Differenzen zwischen den Nebenachsen festgestellt, Fehler neu berechnet und Hauptkoordinaten erhöht werden. Es resultiert eine hübsche Liste mit Eckpunkten.

```
loop = True
while(loop):
    vertices.append(minor[:idx] + [X] + minor[idx:])
    if X == X2:
        loop = False
    for j in range(2):
        if dminor[j] >= 0:
            minor[j] += s[j] + 1
```



```
dminor[j] -= aX
dminor[j] += aminor[j]
X += s[0]
return vertices
```

Für einen stilvollen Abschluss testen wir die Funktion, indem wir einen ebenso mysteriösen wie instabilen Holzbalken auftauchen lassen, als gebührendes Zeugnis Ihres erstaunlichen Wirkens.

```
v1 = mc.player.getTilePos() + minecraft.Vec3(1,1,0)
v1 = minecraft.Vec3(1,1,0) + pos
v2 = v1 + minecraft.Vec3(5,5,5)
bline = getLine([v1.x,v1.y,v1.z],[v2.x,v2.y,v2.z])
for j in bline:
    mc.setBlock(j[0],j[1],j[2],5)
```

Und das war es auch schon! Die Minecraft Pi Edition ist ein perfekter Einstieg in die Programmierung mit Python. Das macht so viel Spaß, dass Sie gar nicht merken werden, wie viel Sie dabei lernen.

► Raus aus dem Stillstand im Gitternetz und alle Winkel nach Herzenslust variieren.



► Machen Sie das nicht zu Hause nach.

# Wir bauen ein

Kreieren Sie Ihr eigenes Minecraft auf dem Raspberry Pi.

**M**inecraft hat seit seinem Erscheinen 2009 unaufhaltsam die Welt erobert und ist auf vielen verschiedenen Plattformen ein galoppierender Erfolg geworden, beispielsweise auf dem PC, der Xbox 360 und der PS3. Die Grundidee von *Minecraft* ist es, verschiedene Materialien abzubauen, mit denen man dann in der Spielwelt neue Objekte erschaffen kann. Das Abbauen ist nicht schwierig, doch die Kunst ist es, zu überleben. Man kann Holz, Fels, Kohle und Metalle abbauen. Daraus kann man dann Häuser, Brücken, Waffen, Werkzeuge und sogar Möbel herstellen. Es kommt aber noch hinzu, dass

es in *Minecraft* einen Tag-Nacht-Zyklus gibt, ähnlich wie im echten Leben,

jedoch deutlich beschleunigt. Tagsüber gibt es herrlichen Sonnenschein und saftig grüne Felder und man ist relativ sicher. Nachts jedoch kommen Monster auf einen zu. Man muss also tagsüber eine Unterkunft bauen, um nachts geschützt zu sein.

Es gibt aber auch eine Version von *Minecraft* – den sogenannten Kreativ-Modus –, bei der weder Nächte noch Monster vorkommen und man völlig ungestört bauen kann. Diese Version ist auf dem Raspberry Pi verfügbar und darauf konzentriert sich diese Übung. Stellen Sie

sich den Modus ähnlich wie eine Kiste voller Legosteine vor. Natürlich wird es auf Dauer mühsam, alles Block für Block aufzubauen, aber keine Sorge, Hilfe ist zur Hand: die Python-Programmiersprache.

## Python?

Python ist eine wirklich hervorragende Programmiersprache, die auf jedem Raspberry Pi vorinstalliert ist. Mit Python können wir Häuser, Brücken, Städte und sogar Weltraumfahrzeuge konstruieren. Python zu lernen ist nicht schwer. Wenn Sie sich einmal mit der Syntax angefreundet haben (also wie man die Sprache verwendet), können Sie Ihr eigenes Plattformspiel programmieren. Wir verwenden Programmierkonzepte wie etwa Schleifen, die einen Befehl mehrfach wiederholen. Das *Minecraft*-Team stellt eine großartige Werkzeugpalette bereit, dank der wir mittels Python wundervolle Inhalte in unserer Spielwelt schaffen können.

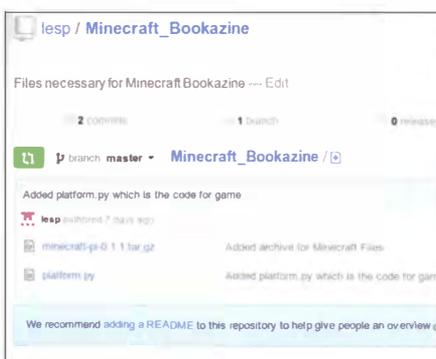
In dieser Übung erstellen wir ein Plattformspiel ähnlich der beliebten *Mario*-Spieleserie und wir verwenden einen Zeitgeber, damit es eine Herausforderung wird, das Ende zu erreichen. Der gesamte Inhalt wird in Python geschrieben. Am Ende der Übung werden Sie grundlegende Programmierkonzepte verstehen und ein wirklich gutes Plattformspiel kreiert haben.

Die *Minecraft*-Welt besteht aus unendlich vielen gleich großen Blöcken (kleinen Würfeln). Es handelt sich um eine dreidimensionale Spielwelt: Die Position jedes Objekts darin lässt sich also mithilfe von X-, Y- und Z-Koordinaten beschreiben.

„Das Team von Minecraft hat eine Reihe fantastischer Werkzeuge geschaffen.“

## Mit Minecraft loslegen

Den Raspberry Pi mit allen benötigten Werkzeugen bestücken.



### 1 Archiv herunterladen

Eine Kopie von *Minecraft* erhalten Sie hier: [https://github.com/lesp/Minecraft\\_Bookazine/blob/master/minecraft-pi-0.1.1.tar.gz](https://github.com/lesp/Minecraft_Bookazine/blob/master/minecraft-pi-0.1.1.tar.gz). Laden Sie dieses Archiv auf den Raspberry Pi herunter. Merken Sie sich den Speicherort.

### 2 Terminal öffnen

Extrahieren Sie im Terminal den Inhalt des Archivs auf den Pi. Klicken Sie auf das LXTerminal-Symbol auf dem Desktop und gehen Sie zum Ordner, in dem Sie das Archiv gespeichert haben. Verwenden Sie dazu den **cd**-Befehl, beispielsweise **cd /home/pi/**.

### 3 Archiv extrahieren

Extrahieren Sie im Terminal das Archiv:

```
tar -zxvf minecraft-pi-0.1.1.tar.gz
```

Drücken Sie Enter und Sie sehen viel Text vorbeiziehen. Damit gibt uns der Befehl zu verstehen, dass er arbeitet. Wenn er fertig ist, werden Sie zur Kommandozeile zurückgeführt.

# Plattformspiel

## Programmiergrundlagen

Python bietet eine Vielzahl von Möglichkeiten zum Lösen von Aufgaben und jeder Programmierer hat für gewöhnlich seine eigene Vorgehensweise. Was aber alle Programmierer gemein haben, sind die grundlegenden Bausteine der Programmierlogik. Wir schauen uns ein paar davon an und wie wir sie in unserem Spiel verwenden.

## Abfolge

Abfolge ist ein Oberbegriff für die Schritte, aus denen die Erledigung einer Aufgabe besteht. Unser Spiel ist eine große Abfolge; es hat einen Anfang, ein Ende und dazwischen die Schritte, mit denen wir Objekte in der Spielwelt erzeugen.

## Schleife

Wie bereits erwähnt, können Codeteile zu einer Schleife zusammengestellt werden, sodass ein Prozess mehrfach wiederholt wird. Es gibt auch Endlosschleifen – in Python „while True“ – wie in dem folgenden Beispiel, welches immer wieder „Hello“ ausgibt, bis der Benutzer es abbricht:

```
while True:
    print "Hello"
```

Wir können auch mit einer „for“-Schleife den Befehl eine vorgegebene Anzahl von Malen wiederholen, was wir später brauchen, um eine bestimmte Anzahl von Plattformen oder Blöcken in unserem Spiel zu erzeugen. Das folgende Beispiel

verwendet eine „for“-Schleife, in der der Wert von **i** in jeder Iteration ausgegeben wird:

```
for i in range(1,6):
    print (i)
```

## Variable

Wir lieben Variablen – sie sind die unbesungenen Helden in Programmen. Sie können Daten speichern (aufbewahren), etwa wie eine leere Kiste. In unserem Spiel speichern wir so die Position des Spielers, die Anzahl der benötigten Plattformen und alles, was wir immer wieder verwenden möchten.

## Ereignis

Ein Ereignis ist wie ein Auslöser, der die Ausführung einer Abfolge in Gang setzt. Unser Spiel könnte so etwas verwenden, um eine Plattform verschwinden zu lassen – hinterlistig!

## Operator

Operatoren werden verwendet, um Werte zu vergleichen und Boolesche Logik zu verwenden (benannt nach dem Mathematiker George Boole, der das Konzept erfand). Boolesche Logik besteht aus „wahr“ (True) und „falsch“ (False) oder auch 1 und 0. Wir können die Spielerposition mit dem Wert in einer Variablen vergleichen; stimmen beide überein (True), können wir bestimmten Code ausführen lassen, der zum Beispiel den Spieler in die Luft katapultiert.

## Einen eigenen Minecraft-Server aufsetzen

Sie können Ihre Spielwelt mit Freunden überall auf der Welt teilen. Wir gehen davon aus, dass Ihr Heimnetzwerk entweder eine statische IP-Adresse hat oder einen dynamischen DNS-Dienst verwendet. Wenn Sie die Spielwelt nur innerhalb des Heimnetzwerks teilen möchten, ist das nicht notwendig; finden Sie einfach mit **ifconfig** im Terminal die IP-Adresse des Raspberry Pi heraus und geben Sie diese den Freunden. Bei einem Ethernet-Netzwerk steht die Adresse unter **eth0**; bei WLAN unter **wlan0**. Ihre Freunde brauchen nur diese IP-Adresse, um sich mit dem Server zu verbinden.

Einen vollständigen Leitfaden finden Sie unter [www.stuffaboutcode.com/2013/09/raspberry-pi-setup-minecraft-server.html](http://www.stuffaboutcode.com/2013/09/raspberry-pi-setup-minecraft-server.html), aber wir zeigen Ihnen hier schnell die wesentlichsten Punkte. Takten Sie den Pi auf Mittel. Stellen Sie die Speicherteilung auf 16 MB. Starten Sie den Raspberry Pi neu. Laden Sie Java herunter und installieren Sie es. Laden Sie den

*Minecraft*-Server herunter und installieren Sie ihn. Öffnen Sie ein Terminal, tippen Sie **nano start.sh** und geben Sie den folgenden Text ein:

```
/opt/jdk1.8.0/bin/java -Xms256M
-Xmx496M -jar /home/pi/spigot.jar nogui
```

Starten Sie dann mit **start.sh** den *Minecraft*-Server.

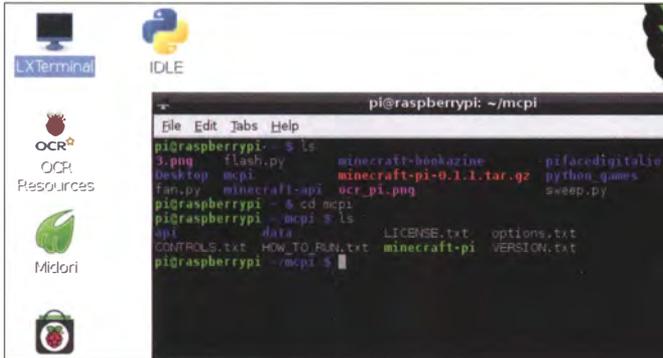
Der Autor des Leitfadens, Martin O'Hanlon, beschreibt sehr detailliert, wie man den eigenen Aufbau am besten anpassen kann. Wenn Sie also einen *Minecraft*-Server mit Ihrem eigenen Code bereitstellen möchten, begeben Sie sich auf die Website, kochen Sie sich einen Kaffee und werden Sie Experte.



› Wenn das Spiel erstellt ist, können Sie es mit Freunden teilen.

## Bessere Welten schaffen

Kreieren Sie Ihre eigene Welt in *Minecraft*.

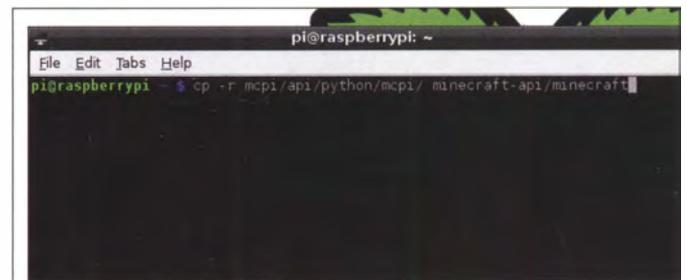


### 1 Minecraft starten

Beim Extrahieren des Archivs wurde ein neuer Ordner mit dem Namen **mcpi** angelegt. Wechseln Sie mit dem **cd**-Befehl in diesen Ordner. Tippen Sie dann **ls**, um den Inhalt des Ordners aufzulisten. Suchen Sie nach **minecraft-pi**; das ist die *Minecraft*-Anwendung.

### 2 Minecraft ausprobieren

Starten Sie *Minecraft*, indem Sie **./minecraft-pi** im Terminal eingeben. Machen Sie sich mit *Minecraft* vertraut, erstellen Sie eine neue Welt und bewegen Sie sich mit den Tasten W, A, S, D herum. Mit der Maus können Sie die Blickrichtung ändern, mit der linken Maustaste Blöcke zerstören und mit der rechten einen neuen Block platzieren.



### 3 Datei minecraft-api.py beziehen

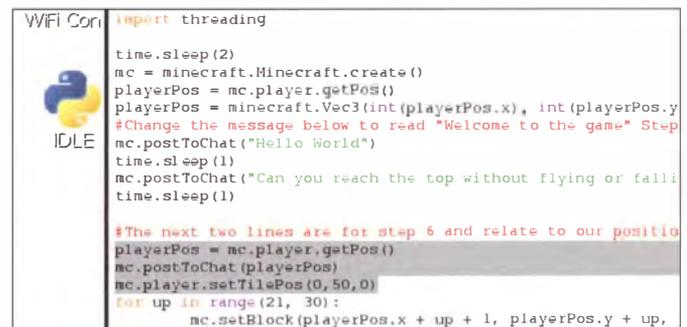
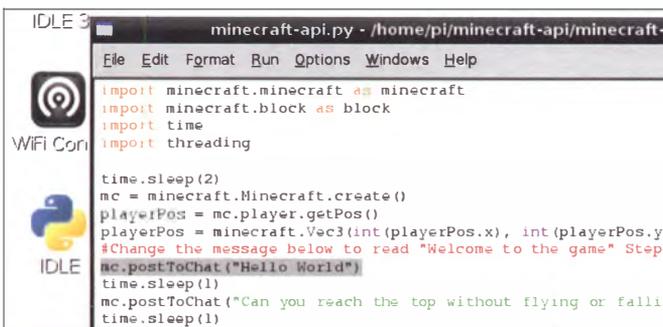
In *Minecraft* muss man Objekte aus einzelnen Blöcken zusammensetzen. Mit Python können wir eine Vielzahl von Dingen schneller erzeugen. Wir brauchen dazu die Datei **minecraft-api.py**, die wir von [https://github.com/lesp/Minecraft\\_Bookazine/blob/master/minecraft-api.py](https://github.com/lesp/Minecraft_Bookazine/blob/master/minecraft-api.py) erhalten, und eine Anwendung namens IDLE. Auf dem Desktop findet sich eine Verknüpfung auf IDLE.

### 4 Dateien kopieren

Erstellen Sie einen Ordner namens **minecraft-api** in Ihrem Heimordner und kopieren Sie den Inhalt von **/home/pi/mcpi/api/python/mcpi** (inklusive enthaltener Ordner) nach **minecraft-api**. Am einfachsten geht das im LX Terminal mit:

```
cp -r ~/mcpi/api/python/mcpi ~/minecraft-api/minecraft
```

**cp** kopiert Dateien; **-r** legt fest, Unterordner dabei mitzukopieren. Kopieren Sie außerdem für den nächsten Schritt die zuvor heruntergeladene Übungsdatei in diesen Ordner.



### 5 Willkommensnachricht hinzufügen

Öffnen Sie die Datei **minecraft-api.py** in IDLE und schauen Sie sich den Code an. Zuerst ändern wir den Code **mc.postToChat("Hello World")** so, dass „Willkommen im Spiel“ ausgegeben wird. Speichern Sie ab, starten Sie eine neue *Minecraft*-Sitzung mit einer beliebigen Welt und klicken Sie dann in IDLE im „Starten“-Menü auf den „Starten“-Befehl. Der Text sollte jetzt erscheinen.

### 6 Den Spieler einsetzen

Der Spieler hat eine Position mit XYZ-Koordinaten. Wir können mit **mc.player.getPos()** diese Position abrufen und in eine Variable namens **playerPos** speichern, um sie später wieder verwenden zu können. Man kann den Spieler auch mit **mc.player.SetTilePos(x,y,z)** verschieben. Wir haben y auf 50 gesetzt, wodurch der Spieler von oben in die Welt hineinfällt.

# Die Messlatte höher setzen

Kein Plattformspiel ohne Plattformen!

```
playerPos = mc.player.getPos()
playerPos = minecraft.util.Util.intVec(playerPos.x, int(playerPos.y), int(playerPos.z))
#Change the message below to read "Welcome to the game" Step 5 in the magazine.
mc.postToChat("Hello world")
time.sleep(1)
mc.postToChat("Can you reach the top without falling or falling @??")
time.sleep(1)

#The next two lines are for step 6 and relate to our position in the game world.
playerPos = mc.player.getPos()
mc.postToChat(playerPos)
mc.player.moveToPos(0,0)
for more in range(0,21):
    mc.setBlock(playerPos.x + more, playerPos.y + more, playerPos.z + more, block.WOOD_PLANKS)
    mc.setBlock(playerPos.x + more + 1, playerPos.y + more, playerPos.z + more, block.WOOD_PLANKS)
    up = range(1, 30):
        mc.setBlock(playerPos.x + up + 1, playerPos.y + up, playerPos.z + up, block.DIAMOND_BLOCK)
        mc.setBlock(playerPos.x + up + 2, playerPos.y + up, playerPos.z + up, block.DIAMOND_BLOCK)

mc.postToChat("You have 30 seconds to reach the top")
time.sleep(1)
mc.postToChat("GO!")
time.sleep(1)

clock = 30
```

## 1 Plattformen bauen

Jetzt bauen wir unsere Plattformen. Wir machen es uns einfacher mit einer **for**-Schleife, die eine vorgegebene Anzahl Male durchläuft.

**for more in range(0,21):**

In diesem Beispiel läuft die Schleife 21-mal und wiederholt folgenden Code:

```
mc.setBlock(playerPos.x + more, playerPos.y + more, playerPos.z + more, block.WOOD_PLANKS)
```

Dieser erzeugt Holzblöcke, angefangen an der Spielerposition und in jeder Iteration einen Block nach vorne, nach oben und zur Seite, wodurch eine Art Treppe entsteht.

```
mc.postToChat("You have 30 seconds to reach the top")
time.sleep(1)
mc.postToChat("GO")
time.sleep(1)

clock = 30
while clock > 0:
    mc.postToChat(clock)
    time.sleep(1)
    clock = clock - 1

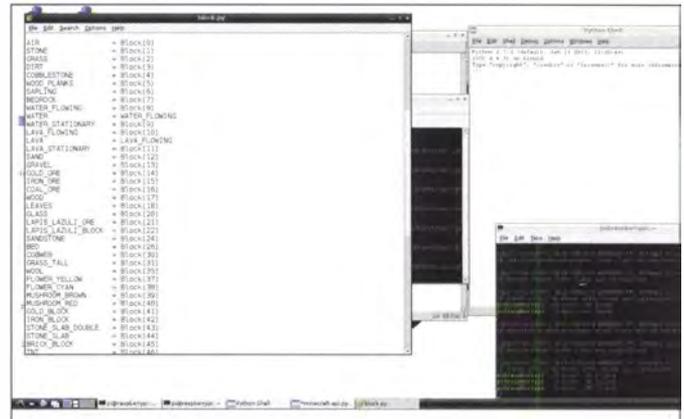
mc.postToChat("Time's up!")
```

## 3 Zeitgeber hinzufügen

Wir wollen ja eine Herausforderung, also führen wir einen Countdown ein. Für den Zeitgeber erstellen wir eine Variable namens **clock** und geben ihr den Wert **30**; das ist eine Ganzzahl. Dann schreiben wir eine „while“-Schleife mit der folgenden Bedingung:

**while clock > 0:**

Der Wert von **clock** wird also verglichen, und wenn er größer als **0** ist, wird der darauffolgende Code ausgeführt.



## 2 Material wählen

Nachdem die Holzblöcke platziert sind, wollen wir ein anderes Material verwenden. Wir haben hier **Diamant** gewählt. Öffnen Sie ein Terminal und suchen Sie im Ordner **/home/pi/minecraft-api/minecraft** die Datei **block.py**. Öffnen Sie diese in einem Texteditor, um eine Liste aller verfügbaren Blocktypen zu sehen. Sie könnten **block.DIAMOND\_BLOCK** auch in tödliche **LAVA** ändern.

```
clock = 30
while clock > 0:
    mc.postToChat(clock)
    time.sleep(1)
    clock = clock - 1

mc.postToChat("Time's up!")
```

## 4 Countdown starten

Der innere Code ist nur drei Zeilen lang, aber er wird so oft wiederholt, bis **clock** den Wert **0** erreicht:

```
mc.postToChat(clock)
time.sleep(1)
clock = clock - 1
```

Der Code zeigt zunächst den Wert der Variablen **clock** auf dem Bildschirm an und wartet dann eine Sekunde lang. Ohne diese zweite Zeile wäre der Countdown ruckzuck vorbei. Zuletzt ändern wir den Wert von **clock**, indem wir vom vorherigen Wert **1** abziehen.

# Weiterführende Literatur

Es gibt viele *Minecraft*-Ressourcen, mit denen Sie das Spiel weiterentwickeln können, aber diese beiden hier gehören wohl zu den empfehlenswertesten.

### www.stuffaboutcode.com

Martin O'Hanlon ist ein Fürsprecher des Prinzips, durch Spielen zu lernen. Er hat eine Reihe von Leitfäden und Blogbeiträgen geschrieben, um Sie an die Hand zu nehmen und Ihnen zu zeigen, wie man mit *Minecraft* richtig tolle Sachen machen kann. Er hat beispielsweise die Skyline von Manhattan und das Space Shuttle



nachgestellt und mit offiziellen Landschaftsdaten *Minecraft*-Welten geschaffen, die auf realen Orten basieren.

### http://arghbox.wordpress.com

Craig Richardson hat auf Basis der Codecademy-Lektionen fabelhafte Hilfsmittel geschaffen,

mit denen man die Python-Programmierung erlernen kann, und er wendet dabei die gleichen Konzepte an, die bei *Minecraft* vorherrschen. Sein Buch ist kostenlos als PDF in zwei Versionen verfügbar: eine für Schüler und Studenten und eine für den Lehrer, die die Konzepte und den Code ausarbeitet.

# Logik lernen

Raspberry Pi wurde entwickelt, um jungen Menschen das Programmieren zu lehren. Guido von Robot ist hierfür der perfekte Begleiter.

In den letzten Monaten haben wir bereits die Kommandozeile und Programmiersprachen wie Python und Scratch kennengelernt. Mit diesen werden Problemlösungen programmiert – doch um eine solche Lösung überhaupt zu identifizieren, müssen wir das ursprüngliche Problem zuerst in seine Bestandteile zerlegen. In diesem Tutorial nutzen wir die Anwendung *Guido van Robot*, um diese Herangehensweise zu lernen.

## Schildkröten und Roboter

Wer sich in den letzten Jahrzehnten mit kindergeeigneten Programmiersprachen auseinandergesetzt hat, der kam um Logo nicht vorbei. Diese Programmiersprache bestand aus nur sehr wenigen Befehlen, die hauptsächlich für das Steuern einer Schildkröte geeignet waren. Da diese Schildkröte stets einen Stift hinter sich herzog, konnte man mit wenig Aufwand komplexe Formen auf den Bildschirm zeichnen.

Um mit Logo ein Quadrat zu zeichnen, würde man die folgenden Befehle ausführen:

```
FORWARD 100
LEFT 90
FORWARD 100
LEFT 90
FORWARD 100
LEFT 90
FORWARD 100
LEFT 90
```

Diese einfache Programmiersprache lehrte Einsteigern (vor allem Kindern) die wichtigste Grundlage der Programmierung: Wie man eine Aufgabe in einzelne Teilaufgaben und Schritte zerlegt, die derart einfach sind, dass sie sogar ein Computer versteht. Genau dies soll auch das vorliegende Tutorial leisten, allerdings ersetzen wir Logo

durch *Guido van Robot* (im Weiteren *GvR*).

Damit Sie dem Tutorial folgen können, sollten Sie zuerst *GvR* auf Ihrem Raspberry Pi installieren. Hierfür starten Sie *LXTerminal* aus dem Anwendungsmenü und geben Folgendes ein: **sudo apt-get install gvrng pythongtksourceview2**

## Hallo, Guido

*GvR* starten Sie nun über das Anwendungsmenü im linken unteren Bildschirmbereich – Sie finden das Programm im Education-Untermenü. Die Programmoberfläche besteht aus zwei Teilen: Im linken Bereich befinden sich Guido und seine Welt, im rechten Bereich gestalten wir diese Welt und geben Guido Anweisungen.

Da die deutschsprachige Übersetzung extrem lücken- und fehlerhaft ist, empfehlen wir die englische Spracheinstellung. Bei dieser Gelegenheit kann Ihr Kind auch gleich die rund 20 englischen Vokabeln lernen, aus denen die Programmiersprache besteht.

Für die ersten Schritte laden wir eine fertige Welt in den Editor. Hierzu speichern Sie eine Archivdatei von <http://bit.ly/lxfguido> auf Ihrem Raspberry Pi. Dann öffnen Sie den World-editor-Tab auf der rechten Seite der *GvR*-Anwendung und öffnen mit File > Open die Datei **GvR1.wld**.

Die linke Bildschirmseite zeigt die soeben geladene Welt, in der sich ein G umgeben von einem Dreieck befindet – dies ist Guido! Guidos Welt ist aufgebaut wie eine amerikanische Stadt: Von links nach rechts führen Straßen, von oben nach unten Alleen. Dazwischen befinden sich Blöcke, und die zahlreichen Kreuzungen ermöglichen die komplikationsfreie Fahrt zu jedem beliebigen Ort. Begrenzt wird diese Welt mit einer undurchdringlichen Mauer. Einige Straßen können durch zusätzliche Mauern blockiert sein und außerdem darf eine Welt sogenannte „Beeper“ enthalten. Diese blauen Kreise sind (neben Guido) die einzigen beweglichen Objekte.

## Erste Schritte

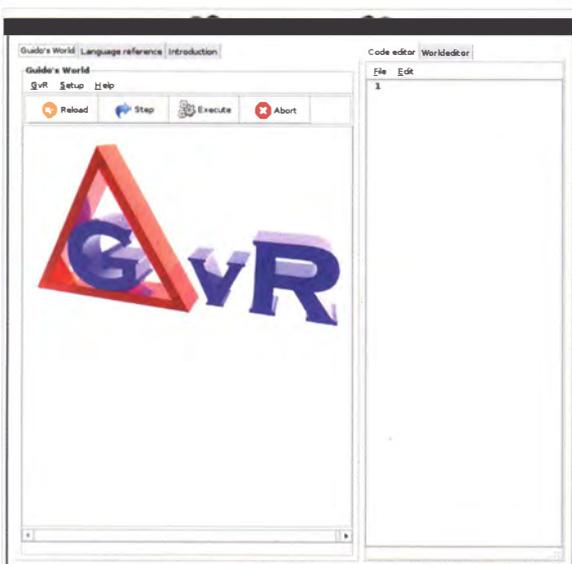
Guido ist ein sehr einfacher Roboter, denn er kann nur fünf Befehle ausführen:

- \* move
- \* turnleft
- \* pickbeeper
- \* putbeeper
- \* turnoff

Der **move**-Befehl bewegt Guido einen Straßenabschnitt vorwärts. **turnleft** lässt ihn sich um 90 Grad links um die eigene Achse drehen.

Wenn Guido beispielsweise drei Straßenabschnitte nach oben laufen soll, dann rufen Sie **turnleft** so oft auf, bis er nach oben sieht, anschließend rufen Sie dreimal **move** auf.

In dieser ersten Welt soll Guido zum Beeper laufen, diesen aufheben, sich zur nächsten Straßenecke begeben und sich schließlich abschalten. Er soll also auf der Kreuzung von



› Dies ist *GvR*, Ihr Fenster in Guidos Welt. Links sehen Sie Guido und seine Umgebung. Rechts haben Sie die Möglichkeit, ihn zu steuern.

# mit Guido

zweiter Straße und vierter Allee stoppen.

Hierzu schalten wir vom World-editor-Tab zum Code-editor-Tab. Hier können Sie die oben genannten Befehle in beliebiger Menge und Reihenfolge eingeben. Sofern jede Zeile exakt einen Befehl enthält, wird sich Guido an diese Anweisungen halten und sie Schritt für Schritt ausführen. Um das Programm zu starten, klicken Sie auf die „Execute“-Schaltfläche

Der Befehl **pickbeeper** funktioniert nur, wenn sich Guido auf einer Kreuzung befindet, auf der auch ein Beeper steht. Ist dies nicht der Fall, dann erhalten Sie eine Fehlermeldung und müssen das Programm mit „Reload“ neu starten.

## Links um!

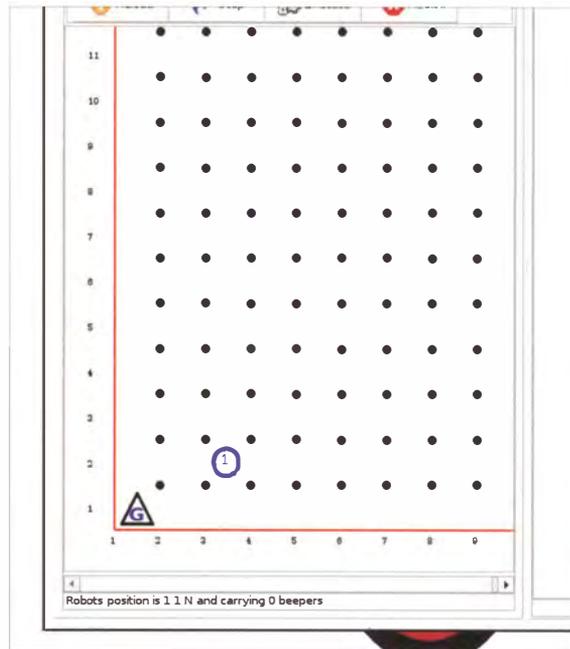
Es gibt unendlich viele Lösungen für diese Aufgabe. Eine relativ kurze Lösung wären die folgenden Anweisungen:

```
move
turnleft
turnleft
turnleft
move
move
pickbeeper
move
turnoff
```

Der schwierigste Teil der Aufgabe war sicherlich die Rechtsdrehung: Da Guido sich nur nach links drehen kann, galt es zu erkennen, dass eine Rechtsdrehung mit drei Linksdrehungen identisch ist.

Das ist nicht besonders elegant und macht das Programm etwas unleserlich. Wenn wir irgendwo einen Programmfehler hätten und Guido die falsche Abzweigung nähme oder in eine Wand führe, dann wäre nur mit hohem Aufwand herauszufinden, wo sich der Fehler unter all den **turnlefts** versteckt. Außerdem müssen wir die gleiche Anweisung mehrfach schreiben – was einem prinzipiell faulen Programmierer stets ein Dorn im Auge ist.

Glücklicherweise ist Guido nicht ganz so dumm, wie es zunächst den Anschein hat. Er kennt zwar nur fünf Befehle, aber da er ein schneller Lerner ist, können wir ihm neue Befehle beibringen. Beispielsweise können wir eine Reihe von Befehlen, wie etwa die drei aufeinanderfolgenden **turnlefts**,



» **Unsere erste Aufgabe: Lotsen Sie Guido zum Beeper und heben Sie diesen auf.**

mit einem Namen versehen und fortan als einzelnen Befehl verwenden.

## Schneller Lerner

Um also das Programm lesbarer und weniger fehleranfällig zu machen, bringen wir Guido die Rechtsdrehung bei. Hierzu benötigen wir lediglich eine neue Anweisung:

```
define turnright:
    turnleft
    turnleft
    turnleft
```

Die erste Zeile besagt, dass wir einen neuen Befehl namens **turnright** definieren. Die eingerückten Zeilen hinter dem Doppelpunkt sind die bereits bekannten Befehle, die Guido ausführen soll, wenn wir ihm das neue **turnright** befehlen. Die Einrückungen sind wichtig: Alle eingerückten Befehle sind Teil der **turnright**-Definition. Eine nicht eingerückte Zeile an dieser Stelle wäre nicht mehr Teil der **defi-**

»

## Welten erschaffen

Unzufrieden mit den Herausforderungen dieses Artikels? Ihr Guido soll in aufregenderen Welten sein Zuhause finden? Dann haben Sie Glück: GvR bietet Ihnen zwei Möglichkeiten, Ihre eigenen Welten zu bauen.

Die erste Variante ist die interaktive Point-and-Click-Methode, die Sie im linken Bildschirmbereich von GvR mit „Open World Builder“ erreichen. Nun erscheint eine leere Welt, in der Sie mit einem Linksklick Wände ein-

ziehen und mit einem Rechtsklick Beeper auf Kreuzungen setzen. Um Guidos Startparameter wie etwa seine Position und Richtung festzulegen, klicken Sie mit der mittleren Maustaste in einen freien Bereich.

Die so erstellten Informationen schreibt GvR in eine Textdatei mit der Endung .wld. Sie können sich den Dateiinhalt jederzeit mit einem Texteditor ansehen, indem Sie den World-editor-Tab im rechten Bildschirmbereich anklicken.

Dies ist zudem die zweite und etwas unbequemere Möglichkeit, Guidos Welt zu verändern.

Die robot-Anweisung legt Guidos Starteinstellungen fest: Straße, Allee, Richtung und Anzahl der Beeper. Jede wall-Anweisung spezifiziert mit einer Kreuzung und einer Richtung ein Wandsegment. Und eine beeper-Anweisung bestimmt eine Kreuzung, an der ein Beeper auftauchen sollen.

## » ne-Anweisung.

Um den neuen Befehl in einem Programm zu verwenden, muss die Definition oben stehen. Erst darunter kann der Befehl verwendet werden. Beispielsweise so:

```
define turnright:
```

```
    turnleft
```

```
    turnleft
```

```
    turnleft
```

```
move
```

```
turnright
```

```
move
```

Versuchen Sie, Ihr Programm unter Einsatz von **turnright** neu zu schreiben. Es sollte nun weitaus lesbarer sein.

## Wiederholungen

Die erste Welt ist erkundet, also laden wir die zweite. Wechseln Sie in den World-editor-Tab zurück und laden Sie mit File > Open die Datei **GvR2.wld**. Diese Welt ist ähnlich wie die erste aufgebaut, allerdings befindet sich hinter dem Beeper eine Wand und die Entfernungen sind größer. Das Ziel jedoch ist das gleiche geblieben: Führen Sie Guido zum Beeper, heben Sie den Beeper auf, fahren Sie eine Ecke weiter und schalten Sie Guido ab.

Wir könnten das Problem auf die gleiche Weise wie zuvor lösen und jede einzelne Drehung sowie jeden einzelnen Schritt im Programmcode spezifizieren. Da Guidos neue Welt aber sehr viel größer ist, wäre der Programmcode bald noch unüberschaubarer, als es oben bereits mit den mehrfachen **turnlefts** der Fall war. Glücklicherweise kennt Guido neben seinen fünf Befehlen auch einige weitere Schlüsselwörter, die das Leben vereinfachen. Das erste Schlüsselwort, das wir nun genauer betrachten, ist **do**:



```
Code editor | World editor
File Edit
1 define turnright:
2   do 3:
3     turnleft
4
5 move
6 turnright
7 move
8 move
9 pickbeeper
10 move
11 turnoff
```

» In GvR benötigt jede Anweisung eine eigene Zeile. Achten Sie bei zusammengesetzten Anweisungen darauf, dass die Zeilen passend eingerückt sind.

**do <Anzahl>:**

**<Anweisungen>**

Wie Sie sehen, entspricht das Format dem des **define**-Schlüsselworts: Zuerst sagen wir Guido, dass nun ein **do** folgt, dessen **<Anweisungen>** er **<Anzahl>** mal wiederholen soll. In GvR2 könnten wir folgendermaßen beginnen:

```
do 4:
```

```
    move
```

Das ist weitaus schneller, als vier einzelne **move**-Befehle einzutippen. Und sollten wir in die Verlegenheit geraten, hundert Bewegungen auszuführen, müssen wir lediglich eine einzige Zahl verändern.

## Entscheidungen

Eine weitere von Guidos Vorzügen ist seine Fähigkeit, Entscheidungen zu treffen. Beispielsweise kann sich Guido selbst die Frage stellen, ob ein Beeper auf der aktuellen Kreuzung vorhanden ist und **pickbeeper** nur dann ausführen, wenn dies tatsächlich der Fall ist. Oder er kann sich fragen, ob er vor einer Wand steht, und gegebenenfalls auf den nächsten Vorwärtsschritt verzichten. Für derlei Fragen gibt es das **if**-Schlüsselwort:

**if <Bedingung>:**

**<Anweisungen>**

Hinter **<Bedingungen>** verstecken sich Guidos Fragen an sich selbst (bzw. an das zugrunde liegende Programm). Insgesamt gibt es 18 gültige Fragewörter, die Sie im Language-reference-Tab der GvR-Oberfläche nachschlagen können. Es handelt sich grundsätzlich um Ja/Nein-Fragen – dies ist der einzige Fragetyp, den Guido (und überhaupt die meisten Computersysteme) beantworten kann.

Beispielsweise würden Sie folgenden Code schreiben, wenn Guido sich nur dann vorwärts bewegen soll, wenn keine Wand vor ihm steht:

```
if front_is_clear:
```

```
    move
```

Manchmal möchten Sie eine bestimmte Anweisung ausführen, wenn eine Bedingung wahr ist, und anderenfalls eine andere Anweisung benutzen. Dies gelingt mit dem Kommando **if ... else ...**:

```
if front_is_clear:
```

```
    move
```

```
else:
```

```
    turnoff
```

Mit dem **if ... elif ... else** Kommando können Sie auch mehrere Bedingungen hintereinander prüfen:

```
if front_is_clear:
```

```
    move
```

```
elif right_is_clear:
```

```
    turnright
```

```
    move
```

## Programmieren ohne Guido

*Guido van Robot* ist vielleicht die einfachste, aber keineswegs die einzige Möglichkeit, in die Welt des Raspberry-Pi-Programmierens einzusteigen. Das Debian-SD-card-Image, auf dem alle unsere Tutorials basieren, enthält beispielsweise die grafische Programmiersprache *Scratch*. Damit programmieren Sie, indem Sie einzelne Programmbausteine mit der Maus verknüpfen. *Scratch* nutzt die gleichen Kontrollstrukturen wie Bedingungen und Schleifen, die wir auch in die-

sem Tutorial kennengelernt haben, führt aber auch zahlreiche weitere Funktionen ein. Mit diesen können Sie beispielsweise Grafiken auf Mausclicks reagieren lassen oder Tastatureingaben auswerten.

Anstatt eines Roboters steuern Sie in *Scratch* eine Vielzahl von Sprites und Bildern, können interaktive Geschichten und Spiele gestalten und lernen, mit Musikdateien umzugehen. Es gibt eine große Community auf [http://](http://scratch.mit.edu)

[scratch.mit.edu](http://scratch.mit.edu), die ihre Kreationen untereinander teilt. So erhalten Sie nicht nur steten Nachschub an kostenloser Software, sondern erfahren auch, wie erfahrene Programmierer arbeiten und Ideen in Projekte verwandeln. Wenn Sie mit *Scratch* anfangen möchten, dann empfehlen wir Ihnen nach dem Besuch der Community-Webseite die intensive Beschäftigung mit den Video-Tutorials auf [http://info.scratch.mit.edu/Video\\_Tutorials](http://info.scratch.mit.edu/Video_Tutorials).

```
elif left_is_clear:
```

```
    turnleft
    move
```

```
else:
```

```
    turnoff
```

Das abschließende **else** ist optional. In dieser Situation stellt der **else**-Zweig das Standardverhalten dar (er wird ausgeführt, wenn nichts Spezielles zutrifft) und ist daher meist eine sinnvolle Ergänzung.

Sie sehen auf den ersten Blick, wie diese Bedingungen nutzbringend eingesetzt werden können. Am Anfang dieses Tutorials haben wir festgestellt, dass der Aufruf von **pickbeeper** eine Fehlermeldung und damit das Programmende verursacht, wenn Guido nicht direkt neben einem Beeper steht. Gleiches geschieht, wenn Guido durch eine Wand fährt und auch zu etlichen weiteren Gelegenheiten. Mit **if**-Abfragen können wir verhindern, dass unsere Programme auf solch unschöne Weise „abstürzt“.

## Wiederholte Wiederholungen

Entscheidungen und Wiederholungen lassen sich auf elegante Weise zum **while**-Kommando kombinieren. Dieser Befehl prüft, ob eine Bedingung zutrifft, und führt in diesem Fall die eingerückten Anweisungen aus. Anschließend wird die Bedingung erneut geprüft und die Schleife so lange wiederholt, bis die Bedingung nicht mehr zutrifft.

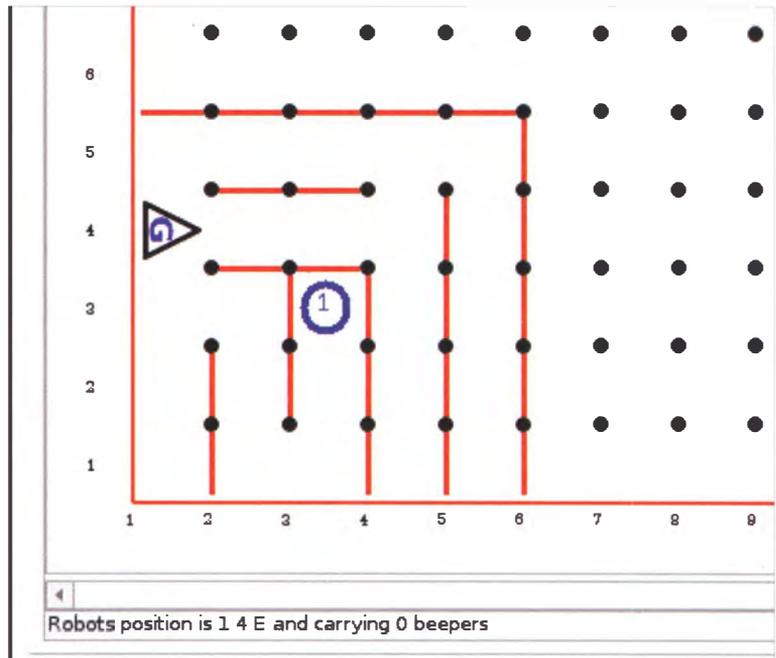
Mit dem **while**-Befehl müssen Sie vorsichtig sein: Wenn Sie eine Bedingung spezifizieren, die die niemals falsch ist, dann steckt das Programm in einer unendlichen Schleife und wird diese niemals verlassen.

Mit einer **while**-Schleife können wir unser *GvR2*-Programm abermals verbessern, es sicherer machen und Guido mehr Kontrolle über sein Schicksal geben. Nach der ersten Reihe von **move**-Befehlen, die Guido auf die richtige Straße führen und ihn nach rechts blicken lassen, können wir ihn mit einer **while**-Schleife so lange vorwärts fahren lassen, bis er auf einen Beeper trifft:

```
while not next_to_a_beeper:
```

```
    move
    pickbeepers
    move
```

Wenn wir alles zusammenführen, was wir bisher gelernt haben, dann ist eine Lösung für das *GvR2*-Problem nur zehn Zeilen lang. Mit einfachen Befehlen wäre das Programm mindestens doppelt so lang. Unsere Lösung ist nicht nur kürzer, sie ist auch robuster und kann relativ einfach an größere und kompliziertere Welten angepasst werden.



› Wenn Sie eine Herausforderung suchen, dann lösen Sie dieses Labyrinth. Es ist Bestandteil von *GvR*.

Sie haben nun alles wichtige gelernt, was Sie über Guido wissen müssen. Mit Ihren neuen Fertigkeiten sollten Sie ihm schnell ein halbwegs intelligentes Verhalten beibringen können, beispielsweise das Durchsuchen eines Labyrinths nach dem Beeper am Ausgang. Natürlich sind auch andere Einsatzzwecke möglich.

## Weitere Projekte

Wenn Sie dieses Unterfangen angehen möchten, dann werfen Sie einen Blick auf die **maze.wld**-Datei, die mit *GvR* mitgeliefert wird. Wie auch unsere bisherigen Welten erlaubt auch **maze** unendlich viele Lösungsansätze und -wege, die sowohl mit der Schritt-für-Schritt-Methode als auch mit Schleifen umgesetzt werden können. Wir schlagen vor, dass Sie beide Methoden ausprobieren.

Natürlich wird auch **maze** irgendwann langweilig und Sie werden sich auf die Suche nach neuen Welten machen. In diesem Fall empfehlen wir Ihnen einen Blick auf den Kasten „Welten erschaffen“, in dem Sie erfahren, wie Sie eigene **.wld**-Dateien erstellen.

## Was nun?

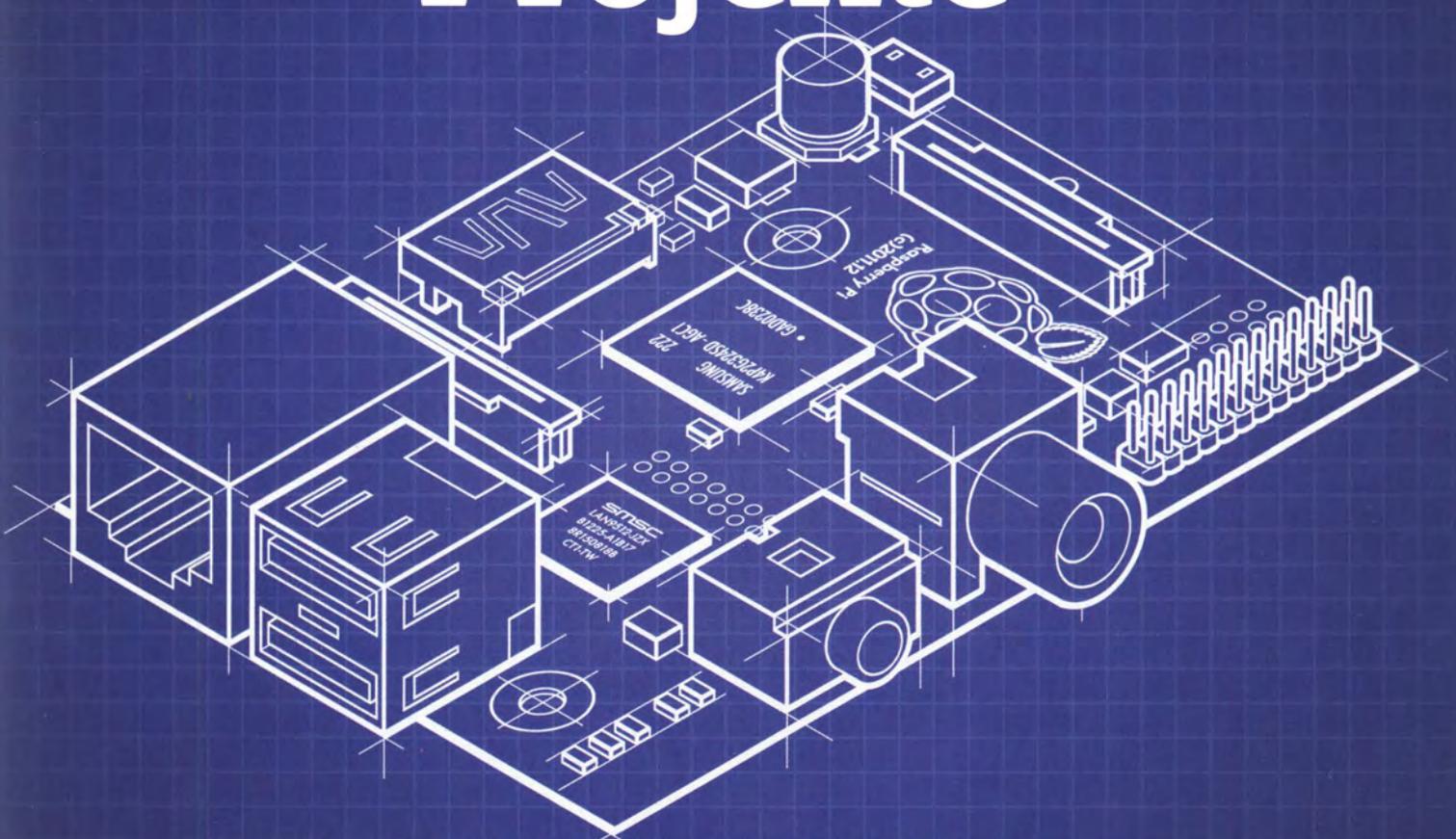
*Guido van Robot* hat Ihnen die wichtigsten Grundlagen der Programmierung beigebracht: ein Problem in einzelne Schritte zu zerlegen, diese Schritte mit Kontrollelementen wie **if**, **while** und **do** zu strukturieren und diese Strukturen mit **define** in wiederverwendbare Funktionen zusammenzufassen.

Eine höhere Programmiersprache bietet einige weitere Möglichkeiten, die über die Fähigkeiten von *GvR* hinausgehen. Glücklicherweise basiert *Guido van Robot* auf einer solchen höheren Sprache namens Python. Python kann alles, was *GvR* kann, und noch vieles mehr. Die

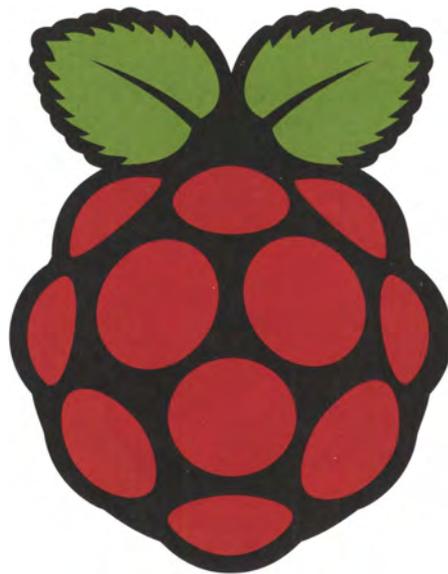
Sprache lässt Sie Benutzereingaben anfordern und entsprechend darauf reagieren, sie kann sich an vergangene Ereignisse erinnern, sie kennt unterschiedlichste Datentypen wie Zahlen, Zeichenketten, Listen oder boolesche Werte, sie ermöglicht die Definition neuer Datentypen – Sie könnten sogar eine verbesserte Version von Guido programmieren. Da Guido auf Python basiert, ist der Umstieg nicht allzu schwierig: Die Syntax – beispielsweise die Einrückungen und die Programmstruktur – ist sehr ähnlich. Außerdem gibt es zahlreiche exzellente Tutorials, die den angehenden

Python-Programmierer bei der Hand nehmen. Ein grandioses Tutorial ist *Think Python: How to think like a computer scientist* ([www.greenteapress.com/thinkpython](http://www.greenteapress.com/thinkpython)). Zusätzlich haben wir in unregelmäßigen Abständen Python-Tutorials im Heft und im Internet finden sich ohnehin zahllose Python-Lehrgänge, aber auch Projekte und Ideen. Und sollten Sie der Meinung sein, dass Python nicht Ihr Ding ist, dann finden Sie sicherlich eine andere Programmiersprache wie *Scheme* oder *Ruby*, die für Einsteiger geeignet ist.

# Raspberry Pi Projekte



**D**ieses Kapitel ist prall gefüllt mit den nützlichsten und kurzweiligsten Projekten, die Sie mit Ihrem Raspberry Pi umsetzen können. Sie können den Mini-PC zu einem Multimedia-Wunder machen und Serien, Filme und Musik streamen. Aber auch das Hosten einer Webseite oder eines Online-Fotoalbums sind ohne Probleme möglich. Sie können sogar alte Retro-Games mit dem Pi zocken.



Cool für die Schule.....	<b>74</b>
Zünde den Raspberry-Booster.....	<b>78</b>
TV-Streaming mit dem Pi.....	<b>86</b>
Heimkino-PC.....	<b>90</b>
Musik-Server mit Mopidy.....	<b>94</b>
Retro-Gaming mit dem Pi.....	<b>98</b>
Seien Sie Ihr eigener Hoster.....	<b>102</b>
Der eigene Cloud-Server.....	<b>106</b>
Torrent-Server einrichten.....	<b>110</b>
Anonym bleiben mit Tor.....	<b>114</b>
Ein Netzwerk aus 2 x Pi.....	<b>118</b>
Server: Chatten via IRC.....	<b>122</b>
Streaming aufs Smartphone.....	<b>126</b>
Hosting eines Foto-Servers.....	<b>130</b>
Elektronische Infotafeln.....	<b>134</b>
Der Pi als Alarmanlage.....	<b>138</b>

# Cool für die Schule

Der Raspberry Pi wird immer beliebter. Wir haben uns für Sie auf dem Raspberry Jamboree die coolsten Projekten angesehen.



## Interview



Der Raspberry Pi soll Kindern helfen, etwas über das Programmieren zu lernen. Dank des Engagements von Lehrern und Erziehern wird dieses Ziel immer besser umgesetzt.

Das zweite Raspberry Jamboree fand innerhalb einer großen Bildungsinnovations-Konferenz statt, an der Pädagogen aus ganz Großbritannien teilnahmen. Es waren

viele Aussteller anwesend, und an vielen Ständen wurden Raspberry-Pi-Produkte als Alternative zu den großen Namen im Bildungsbereich vorgestellt.

Der inoffizielle Botschafter der Raspberry Jams, „Jambassador“ Alan O’Donohoe, eröffnete die Veranstaltung in seinem üblichen rasanten Präsentationsstil. Er arbeitete unglaublich hart, um das Jamboree zu einem Erfolg zu machen, und sein Beitrag wurde hoch gelobt. Im Jamboree 2014 zeigte sich eine Hinwendung zu den pädagogischen

Aspekten des Raspberry Pi, was von den teilnehmenden Lehrern sehr begrüßt wurde. In vielen Workshops und Präsentationen wurden Ideen für Unterrichtseinheiten rund um den Minicomputer vorgestellt. Die ersten beiden Tage des Jamboree waren den Veranstaltungsreihen zum Thema Pädagogik gewidmet, es folgte ein offener Hacking-Tag, an dem alle ihre Projekte vorstellen und Neues lernen konnten. Dutzende von genialen Ideen wurden entwickelt. Wir haben die Köpfe hinter den interessantesten Projekten kennengelernt.



## Martin O’Hanlon

Martin O’Hanlon ist ein großer Befürworter des Raspberry Pi. Ihn fasziniert die Möglichkeit, Kinder damit für das Programmieren zu begeistern. Einige kennen vermutlich seine intelligenten und witzigen Minecraft-Projekte. (Mehr zu Minecraft Seite 58-63 in diesem Bookazine.)

**Du hast zwei coole Projekte entwickelt. Was kannst du uns zu deinen Minecraft-Lerneinheiten sagen?**

**Martin O’Hanlon:** Meine Minecraft-Projekte sind einfach, klein und leicht zugänglich. Für mich ist Minecraft eine tolle Möglichkeit, Leute

dazu zu bringen, dass sie mit dieser Welt spielen und sie nach ihren eigenen Vorstellungen verändern.

**Du hast deine Inhalte für Minecraft auf dem Raspberry Pi produziert. Warum hast du dich für diese Plattform entschieden?**

**MO:** Die Pi-Version von Minecraft ist kostenlos und dadurch für viele Leute zugänglich. Und die API (die Schnittstelle zur Anwendungsprogrammierung), mit der man die Minecraft-Welt programmiert, kann sehr leicht erlernt und manipuliert werden. Viele Plattformen bieten Möglichkeiten, Minecraft zu programmieren,



aber keine ist so flexibel und so leicht zu erlernen wie die Version für den Pi.

### Was kann denn ein Anfänger mit der API für Minecraft machen?

**MO:** Man kann ziemlich schnell einen Block verändern, herausfinden, wo sich ein Block befindet, den eigenen Spieler lokalisieren und verändern, wer der Spieler ist. Minecraft verwendet xyz-Koordinaten, um festzulegen, wo sich etwas in der Welt befindet. Diese vier Dinge sind recht einfach, aber sehr wirkungsvoll. Wenn etwa ein Diamantblock zehn Blöcke hoch in der Luft erscheinen soll, findet man mit der API heraus, wo die eigene Figur ist, fügt dann 10 zur y-Achse hinzu und platziert an dieser Stelle den Block. Mit einem Block ist das schon ganz lustig, aber wenn man in Python einen Loop benutzt, hat man einen Mechanismus, mit dem man sehr schnell sehr viele Inhalte platzieren kann. Mit Minecraft wird auch das Programmieren von Geräuschen für Kinder viel interessanter, weil sie die Minecraft-Welt schon kennen und wissen, wie dort alles funktioniert.

### Welche Art Projekte werden mit der Minecraft-API programmiert?

**MO:** Ich habe viele Spiele gesehen. Nicholas Harris ist erst zwölf Jahre alt und hat schon einige Spiele in Minecraft programmiert! Das ist toll.

### Auf deinem Blog hast du die Skyline von Manhattan in Minecraft dargestellt. Wie hast du das gemacht?

**MO:** Das ist mein Vorzeigeprojekt, mit dem ich veranschaulichen möchte, was alles möglich ist. Mir gefällt das Projekt vor allem deshalb, weil ich ein ganz normales 3-D-Modell dafür verwendet habe, eine sogenannte OBJ-Datei, die in vielen Modellierungsprogrammen verwendet wird. Das Dateiformat OBJ wird eingesetzt, wenn man ein Projekt in ein anderes Modellierungsprogramm verlagert – im Prinzip

werden einfach die Koordinaten und Eckpunkte rübergeholt. Diese OBJ-Datei importiere ich nach Minecraft, mit einem Programm, das ich geschrieben habe. Es liest diese riesigen Textdateien und übersetzt die Daten, damit man sie in Minecraft verwenden kann.

### Hast du dazu mehr Informationen?

**MO:** Auf meiner Website [stuffaboutcode.com](http://stuffaboutcode.com) gibt es viele Anleitungen. Die kann sich jeder anschauen und Minecraft selbst ausprobieren. Der Code und die Anleitungen können gerne mit anderen geteilt werden, und ich bin sehr an Projekten von anderen interessiert.

## ZU DASHCAMS

# „Eine Kamera mit Raspberry Pi-Antrieb nimmt meine Snowboard-Abenteuer auf.“

### Erzählst du uns von deinem zweiten coolen Projekt?

**MO:** Mich interessieren drei Dinge im Leben: Technologie, Autos und Snowboarden. Mein Ziel ist es, den Raspberry Pi in alle drei Bereiche zu integrieren. Am Anfang habe ich meine Autofahrten mit einem Pi und einer Kamera dokumentiert. Später habe ich Daten im Video eingeblendet – Geschwindigkeit, Motordrehzahl, Kühlmitteltemperatur und Drosselklappenstellung. Und dann habe ich diese Idee aufs Snowboarden übertragen.

### Du hast also eine Dashcam fürs Snowboarden?

**MO:** Ja. Die Kamera hat einen Raspberry-Pi-Antrieb und nimmt meine Abenteuer mit dem Snowboard auf. Im Video werden dazu Außentemperatur, Höhenlage und eine Karte eingeblendet, auf der man meine Route sieht. Eindeutig cooler als eine GoPro-Kamera.

### Das klingt super! War es schwer, das zu bauen?

**MO:** Bei diesem Projekt war schon die meiste Arbeit getan, weil ich es nur vom Auto aufs Snowboard übertragen musste. Das Projekt war in zwei Lunchboxen aus Plastik untergebracht: In der einen lag der Raspberry Pi samt Batterie, in der anderen das GPS-Gerät. Ich musste die Verkabelung etwas ändern, vor allem, weil der LötKolben nicht gerade meine Stärke ist. Und ich musste den Python-Code an die neue Nutzung anpassen. Bei den Tests mit dem GPS-Gerät und der offiziellen Pi-Kamera habe ich einen interessanten Bug entdeckt. Immer, wenn die Pi-Kamera lief, wurde das GPS-Gerät gestört, und entweder es kamen gar keine GPS-Daten oder sie waren beschädigt. Die Kamera sendete die Daten so schnell, dass sich ein elektromagnetisches Feld bildete. Ich habe dann die Pi-Kamera mit Alufolie abgeschirmt. Später habe ich das Kabel und die Alufolie noch ordentlich mit einem Polyesteremantel umgeben.

### Welche Software hast du verwendet?

**MO:** Die Programmierung habe ich mit Python gemacht. So konnte ich die vielen bereits vorhandenen Bibliotheken nutzen – etwa für das GPS-Gerät –, und natürlich habe ich PIL (Python Imaging Library) für die Overlay-Grafiken benutzt. Die Grafiken werden in regelmäßigen Intervallen auf einem Frame eingeblendet – es wirkt, als wären sie ständig auf dem Bildschirm präsent.

### Wie interagierst du genau mit dem Raspberry Pi?

**Ein einfaches Interface ist doch bestimmt sinnvoll beim Snowboarden?**

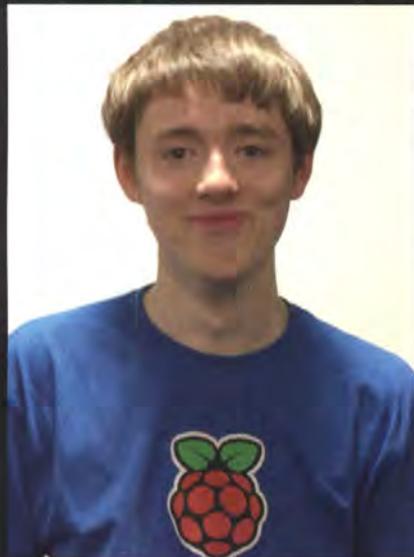
**MO:** Auf dem Snowboard trage ich dicke Handschuhe, die meine Finger

wunderbar warm halten, aber kleine Buttons und Schalter kann ich damit nicht bedienen. Deshalb habe ich einen großen roten Knopf angeschlossen, mit dem ich das Set bediene. Ein Knopfdruck, die Aufnahme startet, noch ein Knopfdruck, sie wird gestoppt. Um meinen Raspberry Pi auszuschalten, halte ich den Knopf für ein paar Sekunden gedrückt.

### Deine Kameraprojekte mit Raspberry-Pi-Antrieb verbinden so viele Bereiche: GPS, Kameras, Datenerfassung für Temperatur und Geschwindigkeit. Was sind deine nächsten Pläne?

**MO:** Als Nächstes möchte ich biometrische Daten erfassen. Ich stelle mir das super vor: Mein eigener Herzschlag wird gemessen und als Videografik auf dem Bildschirm dargestellt. Mein Projekt sehe ich nur als ersten Schritt hin zu allen möglichen Videos, für die der Raspberry Pi die Daten liefert. Ich bin wirklich gespannt, was andere Leute noch für Ideen haben.





## Andrew Mulholland

Andrew Mulholland ist erst siebzehn, aber er hat schon einiges vorzuweisen. Er ist ein preisgekrönter Erfinder, Robotiker und Raspberry-Pi-Enthusiast. Als jüngster STEM-Botschafter Nordirlands organisiert er viele Computer-Clubs und Workshops.

### Worum geht es bei deinem Projekt?

**Andrew Mulholland:** Ich möchte das Linux Terminal Server Project (LTSP) mit dem Raspberry Pi zusammenbringen. Der Linux Terminal Server ist ein zentraler Server, der auf Anfragen von Nutzern, sogenannten Clients, aus dem Netzwerk reagiert – und in meinem Projekt sind die Clients Raspberry Pis. Sie werden mit minimalster Software hochgefahren und holen sich dann Betriebssystem und dazugehörige Software über das Netzwerk vom Zentralserver.

**Der Raspberry Pi unterstützt standardmäßig kein PXE (Preboot Execution Environment). Wie kann man ihn**

### trotzdem über das Netzwerk hochfahren?

**AM:** Ich habe eine 27 MB große Boot-Image-Datei für die SD-Karte erstellt. Sie enthält das Kernelimage und den Bootloader. Der Raspberry Pi kann ja nur von der SD-Karte aus hochfahren, deshalb brauche ich diese Dateien auf der SD-Karte, um einen FAT-Client zu erstellen. Dazu kommt eine config-Datei, die den Pi anweist, nach einer spezifischen IP-Adresse zu suchen und dort das Betriebssystem anzufordern.

### Clever. Wie lange dauert es, bis vier oder fünf Raspberry Pis hochgefahren und benutzbar sind?

**AM:** Das hängt von der Netzwerkgeschwindigkeit ab. Der Raspberry Pi verfügt über eine 100-M-Netzwerkverbindung, aber am besten funktioniert es mit einer Gigabit-Verbindung auf dem Server, weil man die zusätzliche

## ZUM LTSP

„Damit kann man die Schüler-Arbeiten einer Klasse zentral verwalten und speichern.“

Bandbreite für fünf 100-M-Verbindungen braucht. Ich habe das mit 20 Raspberry Pis getestet, die alle mit dem Gigabit-Equipment verbunden waren. Alle 20 Geräte waren innerhalb von 70 Sekunden betriebsbereit.

### Bringt das LTSP dem Raspberry Pi noch andere Vorteile?

**AM:** Auf jeden Fall. So hat es etwa einen grafischen Login-Bildschirm. Und die Nutzer können sich in jeden Raspberry Pi einloggen, haben dabei aber immer Zugriff auf ihr Home-Verzeichnis, so als säßen sie jedesmal am selben Rechner. In einer Schulklasse kann man so die Arbeiten der Kinder von einem zentralen Ort aus verwalten und nicht von

einer unter vielen SD-Karten aus.

### Welche Nutzer siehst du für dein Projekt?

**AM:** Hauptsächlich im Bildungsbereich. Das Ziel ist, LTSP in die Klassenzimmer zu bringen. Mit dem LTSP ist der Computerraum in der Schule immer auf dem aktuellen Stand des neuesten Raspbian-Images, und die Verwendung der empfindlichen SD-Karten wird auf ein Minimum begrenzt. Und die Schüler und Schülerinnen können ihre Arbeiten über das Netzwerk in ihrem Home-Verzeichnis sicher speichern.

### Was für eine Art Server braucht man für eine Schulklasse?

**AM:** Der Server muss nicht besonders leistungsstark sein, denn im Grunde ist es ein Dateiserver. Mein Rat ist, dass man ein Gigabit-Interface für das Netzwerk verwendet. Ich selbst habe mein Test-Netzwerk von einem Laptop aus gesteuert, mit Debian und 1 GB Arbeitsspeicher.

### Funktioniert dein Projekt nur mit Raspbian? Oder läuft es auch mit anderen Betriebssystemen für den Pi?

**AM:** Es läuft auch unter anderen Betriebssystemen, aber man muss wissen, wie man das Boot-Image erstellt. Das LTSP ist für Ubuntu konzipiert, und mit meiner Software kann man leicht ein bootfähiges Image erstellen. Die Foundation hat das Standard-armhf-Image ziemlich verändert, wodurch es schwieriger wird – aber nicht unmöglich. Wer sein eigenes Boot-Image erstellen will, muss die Veränderungen der Foundation entschlüsseln und sie in eigenen Image neu implementieren.





## Simon Walters

Simon Walters beschäftigt sich mit Technik auf Grundschulniveau, worüber er auf [simples.net](http://simples.net) bloggt.

### Was ist dein aktuelles Projekt?

**Simon Walters:** Ich entwickle eine Version von Scratch, die mit der GPIO-Schnittstelle (siehe Seite 146-147) zusammenarbeitet. Als der Raspberry Pi herauskam, wurde angekündigt, dass auch Scratch vorinstalliert sei. Und überall war zu lesen, der Raspberry Pi habe diese wundervolle GPIO-Schnittstelle, über die jeder elektronische Geräte anschließen könne.

### Wie bist du auf die Idee gekommen, Scratch und GPIO zu vereinen?

**SW:** Ich hatte das Glück, dass ich die Arbeiten von Andrew Robinson und dem PiFace-Team zu sehen bekam. Sie stellten mir ein Exemplar ihrer Arbeit zur Verfügung. Ich veränderte es so, dass es direkt mit der GPIO-Schnittstelle funktioniert. Ich unterrichte an der Grundschule und will deshalb den Raspberry Pi und das Programmieren so einfach wie möglich halten. Scratch ist dafür perfekt geeignet.

### Was kann man mit ScratchGPIO machen?

**SW:** Einfache Dinge: eine LED an- und ausschalten oder den Status eines Schalters auslesen. Man kann auch einen Motor damit steuern, und wenn man das einmal raus hat, ist es nicht mehr sehr weit bis zum ersten Roboter.

### Robotik ist ein Feld, für das sich der Pi gut eignet – ist Scratch dafür eine brauchbare Programmiersprache?

**SW:** Ja, mit Scratch kann man sehr gut einfache Roboter programmieren. ScratchGPIO arbeitet dabei im Hintergrund, die Nutzer sehen die Software gar nicht. Sie programmieren den Roboter mit dem Scratch-Interface, und es funktioniert einfach.

### Was sind die Unterschiede zwischen ScratchGPIO und beispielsweise Python?

**SW:** ScratchGPIO verwendet die PiFace-Codebasis und die Python-Bibliothek RPi.GPIO von Ben Croston. Deshalb ist ScratchGPIO abhängig von Python, um zu funktionieren.

### Du kannst also schnell auf viele Kernfunktionalitäten zugreifen, weil dein Projekt auf anderen Projekten aufbaut?

**SW:** Ganz genau. Wenn beispielsweise jemand ein Programm in Python schreibt, für die Arbeit mit Ultraschall-Sensoren, dann kann ich den Python-Code sofort für ScratchGPIO anpassen.

### Wie funktioniert die Umwandlung von Python in Scratch? Liest du einfach die Struktur des Python-Codes und passt sie dann an?

**SW:** Ja. Angenommen, du hast einen Python-Code geschrieben, mit dem die Motordrehzahl verändert werden kann. Dann musst du nur eine Variable in Scratch erstellen – nennen wir sie motorA –, die auf Veränderungen horcht.

## ZUR PIBRELLA

# „Damit kann man einen dreirädrigen Roboter mit zwei Motoren bauen.“

Wenn sich eine Variable in Scratch ändert, wird das über das lokale Netzwerk gesendet, und ein Empfängerprogramm horcht danach. ScratchGPIO funktioniert genau umgekehrt, es horcht nach den Veränderungen.

### Welche Projekte wurden schon mithilfe von ScratchGPIO entwickelt?

**SW:** Es wurden schon ziemlich viele Roboter mit ScratchGPIO gebaut, aber ein Projekt hat mir besonders gefallen: Jemand wollte Feuerwerkskörper mithilfe von ScratchGPIO zünden. Der Pi wurde über GPIO mit dem Kontrollmechanismus für die Raketen verbunden und mit

Scratch die Reihenfolge programmiert, in der sie gezündet wurden. Das war ein Super-Projekt. Aber eigentlich dient ScratchGPIO dazu, dass Kinder im Grundschulalter spielerisch mehr über Elektronik erfahren, mit einfachen Projekten wie Ampeln oder Sensoren.

### Vor einem Jahr hast du uns auf dem Raspberry Jam in Manchester einen Roboter vorgeführt, der aus einem Labyrinth herausfindet. Wie viel kostet so ein Projekt?

**SW:** Dieses Projekt war erstaunlich billig, weil ich die Schrittmotoren mit doppelseitigem Klebeband am Gehäuse des Raspberry Pi befestigen konnte. Das Projekt hat keine 15 € gekostet.

### ScratchGPIO ist eine der Programmiersprachen, die von der neuen Pibrella von Pimoroni und Cynotech unterstützt wird. Was ist das Besondere an dieser Platine?

**SW:** Pibrella ist eine vorgefertigte Erweiterung, mit der jeder sehr schnell mit Inputs und Outputs herumspielen kann. Man muss nicht löten, man muss keine Steckplatten kaufen. Auf der Platine ist alles vorinstalliert.

### Ist die Pibrella-Platine nur etwas für Anfänger?

**SW:** Nein, die Platine wächst mit dem Wissen. Am Anfang verwendet man den eingebauten Button und die LED. Aber nach einer Weile weiß man mehr über Elektronik und schließt Sensoren, Schalter und Motoren an die Pibrella an.

### Dann ist die Pibrella auch eine Motorsteuerungsplatine zum Einsatz in der Robotik? Und wie viel kostet eine Pibrella-Platine für ein Raspberry-Pi-Projekt?

**SW:** Grundsätzlich ja. Mit der Pibrella kann man ziemlich einfach einen Roboter mit zwei Motoren und drei Rädern bauen, wobei das dritte Rad als Laufrolle dient. Eine Pibrella-Platine ist für etwa 15 € zu haben. Man kann damit so viel machen, und alles nutzergerecht. 🍷



# Zünde den Raspberry-Booster

**A**ls die ersten Exemplare des Raspberry Pi Ende Februar 2012 in den Vorverkauf gingen, ahnten die Schöpfer des Mini-Computers recht schnell, dass sie einen Nerv getroffen hatten. Das Interesse war von Beginn an gewaltig, die Erstauflage von 10.000 Stück stand zwanzigmal so vielen Vorbestellungen gegenüber.

Ab Mitte April 2012 wurde dann endlich ausgeliefert, und die Produktion wurde hochgefahren, um die immer weiter steigende Nachfrage bedienen zu können. Im September 2012 waren 500.000, im Februar 2013 1 Million Einheiten des Raspberry Pi verkauft. Bis Ende 2013 fanden weit über 2 Millionen Exemplare des kleinen Rechners einen Käufer, was sogar die optimistische Prognose des Entwicklerteams noch übertraf. Man kann ohne Übertreibung sagen, dass dieser Erfolg in den letzten Jahren einzigartig gewesen ist.

Abseits dieser beeindruckenden Zahlen ist jedoch die Tatsache an sich, dass der Raspberry Pi Anklang bei Computer-Fans in aller Welt findet, durchaus nicht so

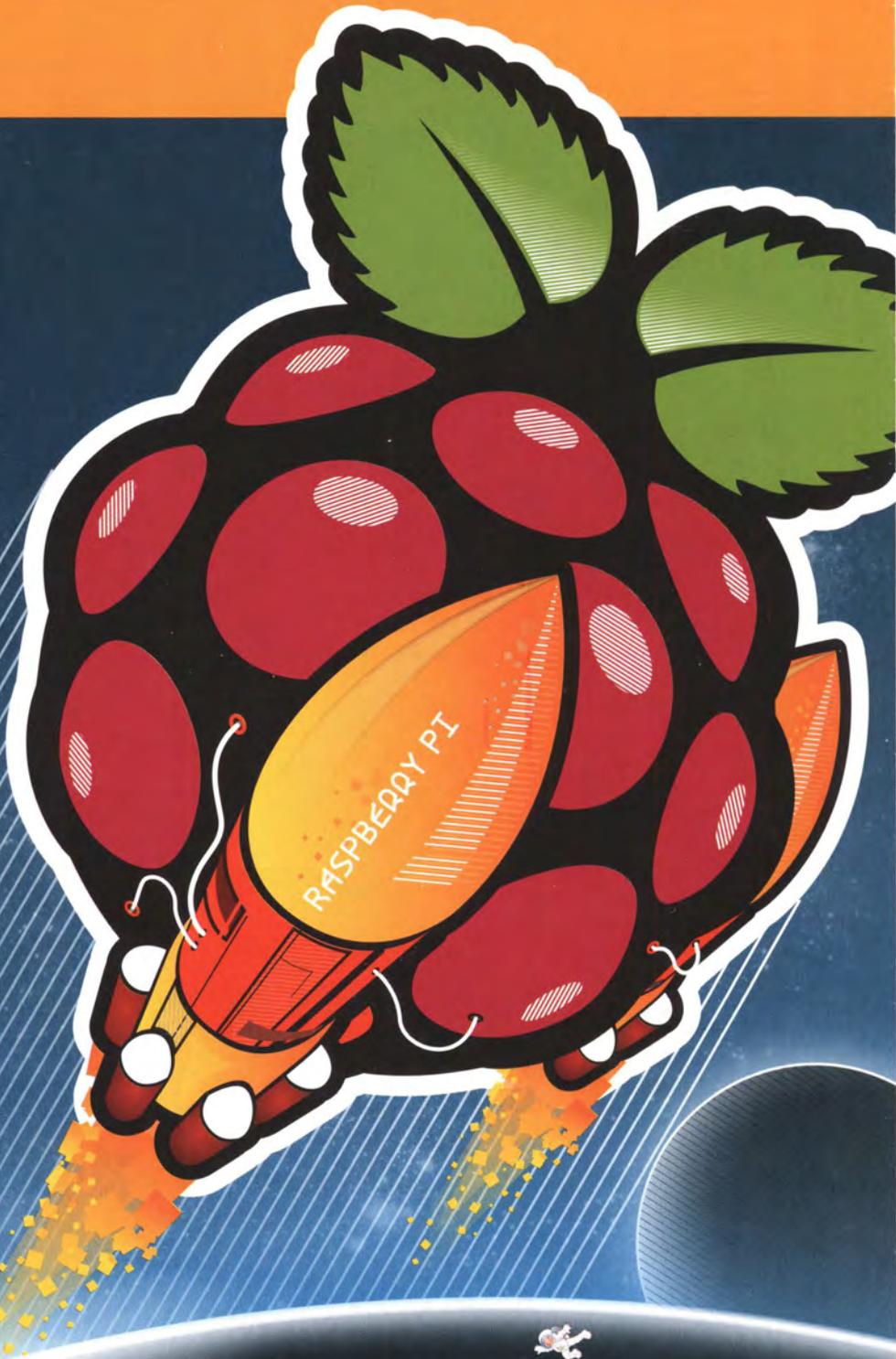
## Ben Everard zückt den virtuellen Lötkolben und zeigt Ihnen, was Sie mit dem Pi so alles anstellen können.

erstaunlich, denn für etwa 35 Euro bekommt man einen voll funktionsfähigen Linux-Rechner mit ARM-Prozessor und 512 MB RAM in den Abmessungen einer Kreditkarte.

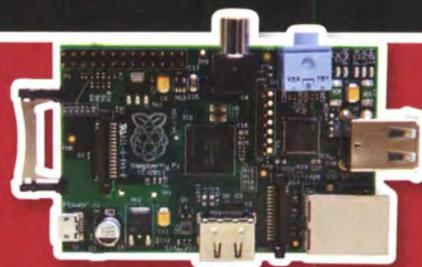
Das Ziel der Raspberry-Pi-Schöpfer um Eben Upton war es, die IT-Ausbildung in Großbritannien, die nicht mehr am Puls der Zeit war, zu revolutionieren. Ob dieses Ziel dereinst erreicht wird, ist noch nicht abzusehen, doch eines lässt sich mit Sicherheit schon heute sagen: Die Welt der

Hobbyschrauber im Computerbereich ist nicht mehr dieselbe wie zuvor! Die winzigen, aber voll funktionsfähigen Geräte sind perfekt dazu geeignet, überall dort eingesetzt zu werden, wo Stromversorgung und räumliche Verhältnisse begrenzende Faktoren sind.

Raspberry Pis sind bereits ins Weltall und über den Ozean geschickt worden, sie werden als Steuerungseinheiten von Robotern, Sicherheitssysteme und für alle möglichen Experimente eingesetzt.



# Distributionen



**E**s gibt etliche Distributionen (Betriebssystemvarianten) für den Raspberry Pi. Auf dieser Seite stellen wir einige der wichtigsten „Distros“ vor.

Eine Distribution wird auf dem Raspberry Pi etwas anders installiert als auf einem normalen PC. Da die Software beim Pi von einer SD-Karte aus läuft, braucht man das Betriebssystem und weitere Programmpakete nur auf diese Karte zu schreiben. Am einfachsten ist das mit dem Befehlszeilen-Tool **dd** zu bewerkstelligen. Mit diesem Kommando erzeugt man Bit-für-Bit-Kopien von Daten zwischen

Geräten und Dateien. Distributionen werden als Image-Dateien zur Verfügung gestellt, die gegebenenfalls noch entpackt werden müssen und auf einen Datenträger geschrieben werden können. Die Befehle lauten:

```
sudo dd if=<image-datei> of=<sd-karte> bs=4k
sudo sync
```

Die zweite Zeile sorgt dafür, dass die gesamten Daten auf die Karte geschrieben werden und nicht in irgendwelchen Zwischenspeichern hängen bleiben. Benutzt man beispielsweise einen PC mit zwei Festplatten **sda** und **sdb**, so wäre die SD-Karte **dev/sdc**.

Sollten Sie unsicher über den Gerätenamen Ihrer SD-Karte sein, lassen Sie sich durch Eingabe von **df -h** in die Befehlszeile eine Liste aller Geräte ausgeben.

Um eine Sicherheitskopie Ihres Raspberry-Pi-Setups anzulegen, können Sie eine neue Image-Datei erzeugen, indem Sie die **if**- und **of**-Flags, die für Input und Output stehen, beim **dd**-Befehl umkehren:

```
sudo dd if=<sd-karte> of=<neue-image-datei> bs=4k
```

Das Image kann dann noch komprimiert werden, beispielsweise mit **gzip** oder **bzip**.

## Raspbian

Raspbian ist die Distribution, die von der Raspberry Pi Foundation offiziell empfohlen wird. Falls Sie keinen triftigen Grund haben, ein anderes System zu verwenden, können Sie zu Raspbian greifen. Es basiert auf Wheezy, der neuesten Version der beliebten GNU/Linux-Distribution Debian, und kann daher auf die umfangreichen Debian-Repositorys zurückgreifen. Die Standard-Desktopumgebung

von Raspbian heißt LXDE. Wer mehr fürs Auge möchte, fährt mit Xfce vielleicht besser. Im Lieferumfang von Raspbian ist das praktische Konfigurationsprogramm *raspi-config* enthalten. Der Raspberry Pi ist so konzipiert, dass auch Kinder mit ihm zurechtkommen, und diesem Ansatz folgt auch Raspbian.

Bezugsquelle: [raspberrypi.org](http://raspberrypi.org)

## Arch Linux

Im Gegensatz zu Raspbian, das den Anwender eher vom internen Setup des Systems abschirmt, ist Arch Linux so konzipiert, dass es dem Anwender das Verständnis der Arbeitsweise des Systems erleichtert. Das Basis-Image enthält nur die nötigsten Komponenten, und alles Weitere kann man später je nach Wunsch hinzunehmen, was allerdings auch einiges an

Arbeit (und natürlich jede Menge Erfahrung in der Zusammenstellung der Einzelteile einer Distribution) mit sich bringt. Das offizielle Wiki zu Arch Linux finden Sie auf [archlinux.org](http://archlinux.org).

Bezugsquelle: [raspberrypi.org](http://raspberrypi.org)

## Raspbmc

Auch wenn der Raspberry Pi im Hinblick auf Bildungszwecke konzipiert wurde, haben viele Bastler sehr schnell auch die Spiele- und Unterhaltungsmöglichkeiten des Geräts für sich entdeckt. Die Distribution Raspbmc verwandelt Ihren Raspberry Pi in ein Medienzentrum, mit dem Sie Ihr Fernsehgerät steuern können.

Raspbmc basiert auf XBMC,

das es ermöglicht, Musik- und Videodateien aus Ihrem Fundus abzuspielen oder Inhalte aus dem Netz zu streamen.

Mehr zu Raspbmc gibt es auf Seite 67 in diesem Heft.

Bezugsquelle: [raspbmc.com](http://raspbmc.com)

## Android

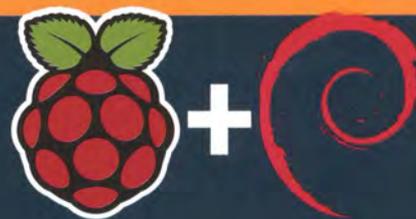
Vielleicht bleibt es nur ein frommer Wunsch, vielleicht wird es eines Tages Wirklichkeit: ein gut lauffähiges Android auf dem Raspberry Pi. Von offizieller Seite wird inzwischen abgewunken, aber einige Fans versuchen weiterhin einen eigenen Ansatz, wobei jedoch alles ziemlich ruckelt und holpert.

Falls sich etwas tut, werden Sie es auf [razdroid.net](http://razdroid.net) erfahren.

Bezugsquelle: Ask Google



# Raspbian



**F**ür die meisten Pi-Benutzer wird Raspbian das „Gesicht“ ihres Geräts darstellen. Sie können das Betriebssystem sehr einfach auf dem neuesten Stand halten, wenn Sie mit den folgenden Befehlen Updates und Upgrades durchführen:

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

Das Konfigurationsprogramm *raspi-config* ist ein sehr nützliches Werkzeug, um Ihren Raspberry Pi optimal einzustellen. Es startet beim erstmaligen Booten des Geräts automatisch, kann aber ansonsten jederzeit mit der Eingabe des Befehls **sudo raspi-config** in die Kommandozeile aufgerufen werden. Eine genaue Übersicht über die Optionen, die Ihnen das Tool bietet,

finden Sie auf Seite 28-31, hier seien nur einige in aller Kürze herausgegriffen:

- » **expand\_rootfs**: Das Dateisystem von Raspbian belegt nur 2 GB Speicherplatz. Falls Sie also eine größere SD-Karte benutzen, würde der ungenutzte Platz verfallen.
- » **memory\_split**: Der Pi verwendet denselben RAM-Speicher sowohl für seine CPU als auch für seine GPU. Mit dieser Option können Sie die Ressourcenaufteilung zwischen den beiden Chips verändern.
- » **overclock**: 50 % mehr Prozessorleistung! Ja, und wie? Schauen Sie in den Kästen unten auf dieser Seite.

» **boot\_behaviour**: Hier können Sie einstellen, dass der Pi direkt in die grafische Benutzeroberfläche booten soll.

Die Installationssoftware von Raspbian umfasst nicht allzu viele Tools. Die benötigten Pakete finden Sie in den Debian-Repositories. Falls Sie Ihre Softwareverwaltung lieber auf einer grafischen Oberfläche vornehmen, empfehlen wir Ihnen das Interface *Synaptic*. Installieren Sie es mit **sudo apt-get install synaptic** und öffnen Sie es über die Einstellungsoptionen im LXDE-Menü.



» Wie alle Distributionen enthält natürlich auch Raspbian eine Auswahl süchtig machender Zeitfresser wie etwa *Squirrels*.

## 15 coole Raspberry-Projekte

- 1 Ein Kindle als Bildschirm: [www.ponnuki.net/2012/09/kindleberry-pi](http://www.ponnuki.net/2012/09/kindleberry-pi)
- 2 Pi-Supercomputer (aus Lego): <http://bit.ly/OGBGfD>
- 3 Steuerung einer Arduino-Plattform über das Web: <http://bit.ly/Xyjsid>
- 4 Bau eines Synthesizers: [www.raspberrypi.org/archives/1878](http://www.raspberrypi.org/archives/1878)
- 5 Roboterfahrzeug mit Nachtsichtgerät: [www.aonsquared.co.uk/the\\_dark\\_pi\\_rises](http://www.aonsquared.co.uk/the_dark_pi_rises)
- 6 Solar-Datenlogging mit dem Raspberry Pi: <http://bit.ly/MHwCHF>
- 7 Echtzeitübersetzung: [www.willpowell.co.uk/blog/?p=219](http://www.willpowell.co.uk/blog/?p=219)
- 8 Pi in einem Wetterballon hoch über der Erde: [www.daveakerman.com/?p=592](http://www.daveakerman.com/?p=592)
- 9 Bierbrauanlage: <http://bit.ly/PSfdzr>
- 10 Pi als Retro-Spielekonsole: <http://petrockblog.wordpress.com/retroPie>
- 11 Wie man ein Betriebssystem baut: [www.cl.cam.ac.uk/freshers/raspberrypi/tutorials/os](http://www.cl.cam.ac.uk/freshers/raspberrypi/tutorials/os)
- 12 Pi als Erinnerungsfoto-Drucker: <http://bit.ly/PvmWjF>
- 13 Pandora im Stream: <http://bit.ly/1hOFxGE>
- 14 Die Wohnung über das Web kontrollieren: <http://bit.ly/WK1YCR>
- 15 Aus dem Pi ein Laptop machen: <http://rpidoack.blogspot.com>

## Overclocking

Der Prozessor unter der Haube des Raspberry Pi ist dafür ausgelegt, mit einer Taktung von 700 MHz zu arbeiten. Das bedeutet, er führt 700.000.000 Rechenoperationen pro Sekunde aus. Allerdings bedeutet „dafür ausgelegt“ nicht, dass man diesen Wert nicht noch steigern könnte. Das kann man nämlich.

Damit einher geht allerdings ein höherer Stromverbrauch, woraus wiederum eine größere Wärmeentwicklung resultiert. Hier muss man etwas aufpassen, denn dadurch kann die Platine beschädigt werden.

Praktischerweise enthält das Konfigurationswerkzeug des Raspberry Pi, *raspi-config*, eine Option (jedenfalls in den Programmversionen seit Ende 2012), bei der man das Overclocking (Übertakten) schrittweise einstellen kann und gleichzeitig die erlaubte

Wärmeentwicklung gedeckelt ist. Da es sich um ein offizielles Tool des Herstellers handelt, erlischt durch das Overclocking auch nicht die Garantie des Geräts.

Das Konfigurationsprogramm rufen Sie mit dem Befehl **sudo raspi-config** in der Kommandozeile auf, suchen Sie dort dann die Option **overclock**. Es gibt mehrere Einstellungsschritte bis zu einer Mehrleistung von 50 %. Sollten Sie im übertakteten Modus Stabilitätsprobleme bei Ihrem Raspberry Pi feststellen, starten Sie ihn neu und drücken Sie während des Bootens die Shift-Taste, um das Overclocking zu deaktivieren.

Falls Sie die Temperatur des Prozessors in konkreten Werten angezeigt bekommen möchten, können Sie der LXDE-Oberfläche ein Temperatur-Widget hinzufügen. So oder

so schaltet sich das Overclocking automatisch aus, sobald eine Temperatur von 85 Grad Celsius erreicht wird.



» Die Übertaktung des Pi-Prozessors führt zu einem Mehr an Stromverbrauch, und das kann sich auf die Stabilität der Rechenleistung auswirken.

# Raspbmc



**F**alls Sie Ihren Raspberry Pi im Sinne eines Computers für den allgemeinen Gebrauch mit gelegentlicher Medienwiedergabe verwenden, können Sie einen Media-Player wie VLC auf dem Raspbian-Betriebssystem installieren und sind vermutlich damit gut versorgt. Die geringe Größe und der geräuschfreie Betrieb des Raspberry Pi machen diesen aber auch zum perfekten Kandidaten für die Einrichtung eines persönlichen Entertainment-Centers.

Ein solches Projekt lässt sich sicher auch auf der Grundlage von Raspbian umsetzen, aber wenn Sie sich die Arbeit nicht machen möchten, können Sie direkt auf ein vorgefertigtes Media-Betriebssystem namens Raspbmc zurückgreifen. Das geht so:

Laden Sie das Installationsprogramm von der offiziellen Website [raspbmc.com](http://raspbmc.com) herunter, kopieren Sie es auf den Pi (wozu ein funktionierendes Raspbian notwendig ist), und geben Sie Folgendes ein:

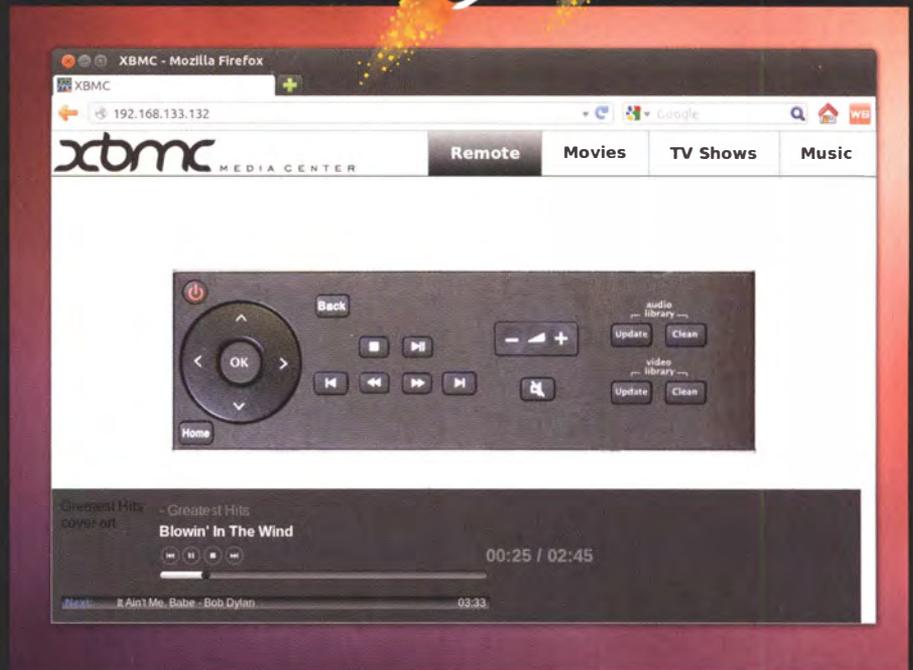
```
sudo python install.py
```

Im Zuge der Installation werden die Daten auf Ihrer SD-Karte gelöscht. Sichern Sie also zuvor alles Wichtige. Nachdem das Installationsprogramm durchgelaufen ist, führen Sie einen Neustart des Systems durch. Sobald die grafische Oberfläche lädt, sehen Sie sofort den Unterschied:

Raspbmc verwendet den beliebten XBMC-Media-Desktop, der sich ziemlich vom



› Die Platinen der neueren Variante des Raspberry Pi (Revision 2) besitzen Befestigungslöcher, damit das Gerät elegant untergebracht werden kann.



› Dies ist die Standardoptik der Fernbedienungsfunktion von Raspbmc, aber Sie können bei den Settings ein anderes Design einstellen.

schlichten Design der Raspbian-Standardoberfläche LXDE unterscheidet. Sie können mit Raspbmc Medien abspielen, die lokal auf Ihrem Gerät gespeichert sind, aber auch Inhalte aus dem Internet streamen, wofür Sie dann allerdings spezielle Add-ons brauchen.

Lokale Musik- und Videodateien können entweder von einem USB-Speichergerät geladen werden oder per FTP (mit den üblichen Logindaten „pi“ und „raspberry“) direkt von der SD-Karte.

Je nachdem, über welchen Anschluss (Audio-Ausgang mit 3,5-mm-Klinkenbuchse oder HDMI-Buchse) sie das Tonsignal senden möchten, müssen Sie bei den Systemeinstellungen noch zwischen analog und HDMI wählen.

Bleibe noch die Frage, wie Sie Ihr Fernsehgerät vom Raspberry Pi aus ansteuern. Mit Tastatur und Maus könnte das auf die Dauer ein wenig unbequem werden. Die Entwickler von XBMC haben darum einen Fernbedienungsmodus eingebaut. Am einfachsten und sichersten lässt sich das mithilfe der Weboberfläche von Raspbmc bewerkstelligen – so können Sie mit jedem Gerät, das sich im

selben Netzwerk befindet und auf dem ein Webbrowser installiert ist, die Wiedergabe steuern. Dieser Modus ist standardmäßig eingestellt, das heißt, Sie benötigen nur noch die IP-Adresse Ihres Raspberry Pi, um diesen von einem anderen Gerät aus per Webbrowser ansprechen zu können. Die IP-Adresse finden Sie unter System > System Info.

## Für Fortgeschrittene

Wenn man möchte, kann man sogar seinen gesamten Fernsehkonsum – auch das Anschauen von TV-Sendungen direkt aus dem Live-Programm und die Aufzeichnung von Sendungen – per Linux steuern. Dies geht mittels *MythTV* (Bezugsquelle: [mythtv.org](http://mythtv.org)), wobei ein separater Computer mit TV-Anschluss als Server fungieren muss. Sie können auch Videos abspielen, die auf anderen Geräten innerhalb Ihres Netzwerks gespeichert sind, zum Beispiel auf einem Dateiserver. Die genaue Vorgehensweise kann sich dabei je nach Hardware und Datentyp unterscheiden. Konsultieren Sie am besten das XBMC-Wiki unter <http://bit.ly/OOvXb6>.

# Kamera-Controller



Sichern Sie Ihre Fotoaufnahmen mithilfe des Raspberry Pi.

**D**a der Raspberry Pi so klein ist, eignet er sich hervorragend dafür, andere eingebettete Systeme zu steuern. Man könnte zwar meinen, das wäre überflüssig, da die eingebetteten Systeme selbst ja bereits eine eigene Steuereinheit besitzen, aber mit dem Pi haben wir die Möglichkeit, den entsprechenden Geräten neue Funktionen angedeihen zu lassen, die sie aus eigener Kraft nicht (oder nur mit viel Aufwand) bewältigen könnten. Fast alles, was man an einen normalen Desktop-PC anschließen kann, lässt sich auch vom Pi mit einem Skript ansprechen, doch in unserem Beispiel wählen wir eine Digitalkamera als Versuchsobjekt aus, und zwar aus zwei Gründen: Erstens unterstützt Linux beinahe jede Kamera, und zweitens gibt es eine Menge nützlicher Anwendungen, die man mit dem Pi und einer Kamera durchführen kann, sobald man das Grundkonzept verinnerlicht hat.

Das beste Befehlszeilen-Tool für die Steuerung von Digitalkameras, das unter Linux zu haben ist, heißt *gPhoto2*. Holen Sie es sich mit folgendem Kommando:

```
apt-get install gphoto2
```

Bevor wir in die Einzelheiten unseres Projekts gehen, beschäftigen wir uns kurz mit diesem nützlichen Werkzeug und seinen Fähigkeiten. Da die grafische Benutzeroberfläche des Raspberry Pi möglicherweise versuchen würde, die Kamera als Ressource einzubinden, und dies *gPhoto2* vor einige Probleme stellen würde, ist

## Stromversorgung

Der Raspberry Pi bezieht seine Stromversorgung über den Micro-USB-Port. Dieser liefert eine Spannung von 5 V, und die Raspberry Pi Foundation empfiehlt eine Stromstärke von mindestens 700 mA. Diese kann problemlos von einem Netzteil oder per USB-Verbindung von einem anderen

Computer zur Verfügung gestellt werden. Wenn Sie den Pi als portables Gerät benutzen möchten, gibt es andere Optionen. Vier AA-Batterien sollten genügend Strom zur Verfügung stellen. Am besten gefiel uns jedoch die Variante, einen Ersatzakku für Smartphones direkt am Raspberry Pi einzustecken.

es ratsam, den Pi ohne den grafischen Desktop hochzufahren. Dies können Sie im Konfigurationsprogramm *raspi-config* einstellen. Öffnen Sie dieses mit dem Befehl `sudo raspi-config`, wählen Sie beim Punkt `boot_behaviour` „No“, um den Pi ohne

Modell nicht auf der Liste steht, haben Sie leider keine Möglichkeit, es auf dem Pi zu aktivieren, sondern müssen auf eine andere Kamera ausweichen, wenn Sie den Versuch durchführen möchten.

Da die unterstützten Kameras sich in Ihren Steuerungsmöglichkeiten voneinander unterscheiden, müssen wir zunächst etwas über die spezifischen Gegebenheiten des verwendeten Modells herausfinden. Mit dem Kommando

## „Der Pi eignet sich dafür, andere eingebettete Systeme zu steuern.“

Desktop hochzufahren (siehe auch Seite 31), und booten Sie erneut.

Sie befinden sich nun in der Nur-Text-Steuerungsumgebung des Pi. Schließen Sie die Kamera an und tippen Sie:

```
gphoto2 --auto-detect
```

Dieses Kommando führt dazu, dass nach Kameras gesucht wird, die mit dem Pi verbunden sind. Die allermeisten Digitalkameras werden unterstützt. Sollten Sie jedoch das seltene Pech haben, dass gerade Ihr

```
gphoto2 --auto-detect --abilities
```

lassen Sie sich die verfügbaren Aktionen anzeigen. Grundsätzlich gibt es zwei verschiedene Gruppen von steuerbaren Aktionen: Capture (Aufnahme) sowie Upload/Download. Falls Aufnahmeaktionen möglich sind – was in der Regel eher bei höherwertigen Kameras der Fall ist –, kann man mittels Skriptbefehlen Fotos schießen. Upload/Download bedeutet, dass Bilder, die auf der Speicherkarte archiviert sind, weiterverarbeitet werden können – diese Aktionsart ist bei den meisten unterstützten Kameras verfügbar.

Im Rahmen dieses Projekts beschränken wir uns auf die Upload/Download-Funktionen. Das simpelste Kommando, das wir einer Kamera übermitteln können, ist die Anweisung, alle gespeicherten Fotos zu übertragen:

```
gphoto2 --auto-detect --get-all-files
```

Damit werden alle Dateien von der Kamera in das aktuelle Verzeichnis heruntergeladen. Bei einem normalen PC wäre das problemlos möglich, doch beim Raspberry Pi sollte man angesichts des begrenzten Speicherplatzes auf der SD-Karte die Fotos lieber auf einem anderen Medium wie zum Beispiel einem USB-Stick speichern. Um dies in einer interaktiven Sitzung durchzuführen, könnte man den USB-Stick einfach mithilfe eines Tools mit grafischer



› Mithilfe eines kleinen Skripts können Sie den Raspberry Pi automatisch Fotos von einer Digitalkamera herunterladen lassen.



Benutzeroberfläche einhängen, dann den Befehl **df -h** ausführen, um den Einhängenpunkt festzustellen, und schließlich mittels **cd** in das entsprechende Verzeichnis wechseln.

Da das Ganze jedoch automatisch vorstattgehen wird, müssen wir den Ort des Geräts kennen. Es gibt mehrere Möglichkeiten, diesen festzumachen, aber wir werden es einfach halten. Wir hängen die erste Partition der ersten seriellen Platte ein und speichern die Bilder dort.

Wir gehen davon aus, dass Sie den Standardnutzer „pi“ verwenden, andernfalls müssen Sie das Skript entsprechend anpassen. Zunächst müssen wir einen Einhängenpunkt für das Laufwerk festlegen. Es handelt sich dabei lediglich um einen Ordner, und dieser kann an einer beliebigen Stelle angelegt werden. Wir weichen hier von der Konvention ab und erstellen den Ordner im Home-Verzeichnis – wie folgt:

```
mkdir /home/pi/pic_mount
```

Danach können wir mit dem Skript beginnen. Mit diesem werden wir das Laufwerk einhängen und die Fotos herunterladen:

```
#!/bin/bash
if mount /dev/sda1 /home/pi/pic_mount
;
then
    echo „Partition mounted“
    cd /home/pi/pic_mount
    yes ‚n‘ | gphoto2 --auto-detect
    --get-all-files
    umount /dev/sda1
else
    echo „/dev/sda1 could not be
    mounted“
fi
```

Der Befehl **yes ‚n‘** gibt lediglich eine Reihe von n Zeichen aus, das heißt, wenn mit **gphoto2** eine zuvor heruntergeladene Datei überschrieben werden soll, wird dies zurückgewiesen. Das **umount** ist unabdingbar, da es dafür sorgt, dass das Laufwerk

korrekt synchronisiert wird und dann entfernt werden kann.

Wir nennen das Skript **get-pics.sh** und speichern es im Home-Verzeichnis des Raspberry Pi ab. Um es ausführbar zu machen, geben Sie ein:

```
chmod +x /home/pi/get-pics.sh
```

Sie sollten das Skript nun manuell starten können. Dabei müssen Sie **sudo** verwenden, da das Skript das Laufwerk einhängen muss.

Der letzte Schritt besteht darin, das Skript automatisch ausführen zu lassen. Dazu fügen wir es der Datei **etc/rc.local** hinzu. Das Skript startet während des Bootvorgangs und läuft unter Root, darum brauchen wir uns keine Gedanken über irgendwelche Berechtigungen zu machen. Öffnen Sie einfach die Datei als Root mit einem Texteditor, beispielsweise mit dem Befehl **sudo nano /etc/rc.local**, und fügen Sie **/home/pi/get-pics.sh** **///end code//**

**direkt vor der Zeile exit 0 ein.**

Jetzt brauchen Sie nur noch Ihre Kamera anzuschließen (und einzuschalten!) und

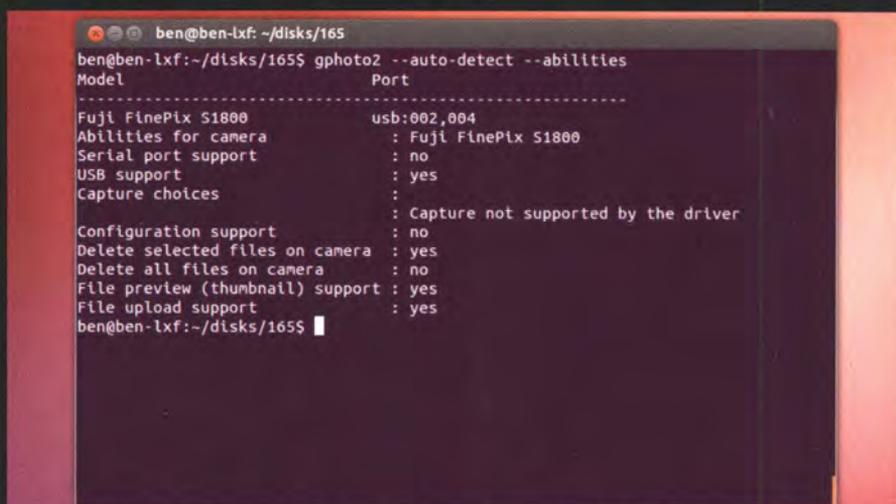
den USB-Stick einzustecken, und schon werden Ihre Fotos automatisch während des Bootvorgangs auf den Stick kopiert.

## Ausblick

Statt die Fotos auf einem USB-Stick zu speichern, könnten Sie sie auch zu einem Online-Dienst wie Flickr hochladen. Sie könnten auch eine Art Schalter einbauen, mithilfe dessen Sie Ihrem Raspberry Pi mitteilen, welche Fotos er uploaden und welche er auf den USB-Stick speichern soll, beispielsweise in Abhängigkeit von der Bildauflösung: kleine Bilddateien auf Flickr hochladen und große speichern.

Dabei müssen Sie aber nicht stehen bleiben. Wenn Sie einen Wireless-Dongle an den Pi anschließen, könnten Sie den Rechner als HTTP-Server verwenden. Mithilfe von PHP oder einer anderen webgeeigneten Skriptsprache ist es möglich, eine Schnittstelle zu **gPhoto2** einzurichten, über die man sich per Smartphone mit dem Gerät verbinden kann.

Schließlich könnten Sie sich auch der erwähnten anderen Gruppe steuerbarer Aktionen zuwenden, nämlich Capture, und mithilfe des Raspberry Pi die Fotoaufnahme auslösen.



› **gPhoto2** bietet viele weitere Features, auf die wir hier nicht näher eingehen, beispielsweise Java- und Python-Anbindung. Infos dazu gibt es auf der Projekt-Website [www.gphoto.org](http://www.gphoto.org).

## Netzwerkverbindung

Einen Raspberry Pi mit dem Internet zu verbinden, bewerkstelligen Sie (jedenfalls beim Modell B) am einfachsten über ein Ethernet-Kabel. In Fällen, wo eine Kabelverbindung unpraktisch ist, bietet sich ein USB-Wireless-Adapter als drahtlose Alternative an. Diesen können Sie mit Ihrem Router verbinden, aber es ist auch möglich, ein Smartphone als Modem einzusetzen. Falls diese Funktion (Tethering) nicht vom Telefonanbieter gesperrt worden ist, können Sie zum Beispiel bei einem Android-Mobilgerät dessen WLAN- oder 3G-Verbindung mit dem Pi teilen. Sofern Ihr Mobilfunkvertrag eine Begrenzung des

Datenvolumens vorsieht, sollten Sie dies allerdings im Hinterkopf behalten, wenn Sie mit dem Pi größere Dateien aus dem Netz herunterladen. Aktivieren Sie das Tethering auf Ihrem Mobilgerät (bei den WLAN- und Netzwerkeinstellungen), und verbinden Sie es mit dem Raspberry Pi. Auf dem Pi tippen Sie **sudo ifconfig** in die Kommandozeile, dann sollte die Schnittstelle als **usb0** angezeigt werden, allerdings ohne IP-Adresse. Netzwerkschnittstellen kontrollieren Sie auf dem Pi mithilfe der Datei **/etc/network/interfaces**. Standardmäßig existiert darin kein Eintrag für USB-Networking, darum

müssen Sie einen solchen Eintrag selber vornehmen. Öffnen Sie die Datei als **sudo nano /etc/network/interfaces**, und fügen Sie dort die folgenden Zeilen hinzu:

```
iface usb0 inet dhcp
nameserver 208.67.220.220
nameserver 208.67.222.222
```

Es handelt sich hierbei um Nameserver des Unternehmens OpenDNS, es stehen aber auch Alternativen zur Verfügung. Nach einem Neustart werden die Änderungen wirksam, und Ihr Raspberry Pi sollte mit dem Internet verbunden sein!

# LED-Anzeige

Die GPIO-Pins des Raspberry Pi und was man mit ihnen machen kann.

**D**a der Raspberry Pi so klein ist, eignet er sich hervorragend dafür, andere eingebettete Systeme zu steuern. Ja, das hatten wir schon, aber es bleibt richtig und zeigt die Vielseitigkeit dieses Wunderwinzlings.

Es gibt allerdings ein kleines Problem, und das ist, dass man nicht erkennen kann, was gerade im Raspberry Pi vor sich geht, wenn man ihn „headless“ betreibt, also ohne angeschlossenen Monitor und Tastatur. An eine mögliche Lösung dieses Problems haben die Pi-Entwickler jedoch bereits gedacht: Der Rechner ist mit einer Reihe GPIO-Pins ausgestattet. Dies sind Kontaktstifte mit integrierten Schaltkreisen zur Allzweckein- und -ausgabe.

Die insgesamt 26 GPIO-Pins befinden sich am Rand der Platine, zwischen dem Video-Ausgang und dem Anschluss für die SD-Karte. Die Schaltkreise können Informationen von beliebigen Quellen ausgeben, doch in diesem Beispiel werden wir sie

## „Wenn man falsche Kontakte verbindet, kann man den Pi zerstören!“

dazu benutzen, das letzte Byte (entspricht dem vierten Zahlenblock) der aktuellen IP-Adresse des Geräts anzuzeigen. Dies ist nützlich, wenn man den Raspberry Pi per Fernzugriff steuern möchte, aber keine

statische IP-Adresse einrichten kann – etwa, weil das Gerät in verschiedenen Netzwerken eingesetzt wird. Die ersten drei Bytes (oder Zahlenblöcke) der IP-Adresse lassen sich meist mithilfe der Netzmaske herausfinden, doch das letzte Byte ist oft schwierig zu ermitteln, wenn kein Monitor angeschlossen ist.

Für unser Projekt benötigen wir das Programm **gpio**, das ein Element der *WiringPi*-Bibliothek darstellt. Vertiefende Informationen dazu sind unter <http://bit.ly/RP8UKJ> zu finden. Da die Software als Quellcode zur Verfügung gestellt wird, müssen Sie sie zunächst entpacken und kompilieren.

```
tar xvf wiringPi.tgz
cd wiringPi/wiringPi
make
sudo make install
cd ../gpio
make
sudo make install
```

```
tar xvf wiringPi.tgz
cd wiringPi/wiringPi
make
sudo make install
cd ../gpio
make
sudo make install
```

Außerdem brauchen wir noch **bc**:

```
sudo apt-get install bc
```

So viel zur Software, nun ist die Hardware an der Reihe. Zuvor jedoch ein warnender Hinweis: Wenn man die falschen Kontakte miteinander verbindet, kann man dem Raspberry Pi den Garaus machen! Vergewissern Sie sich daher lieber zweimal, ob Sie den richtigen Draht in der Hand haben, bevor Sie etwas zusammenlöten oder -stecken.

Die Schaltung ist bei unserem Versuch sehr simpel: Wir müssen jeden Output mit dem Pluspol (dem langen Anschlussstift) einer LED verbinden, den Minuspol (den kurzen Stift) jeder LED mit einem elektrischen Widerstand von 1 kOhm und das andere Ende jedes Widerstandes erden (siehe Abbildung 3). Die einzelnen Bauteile haben

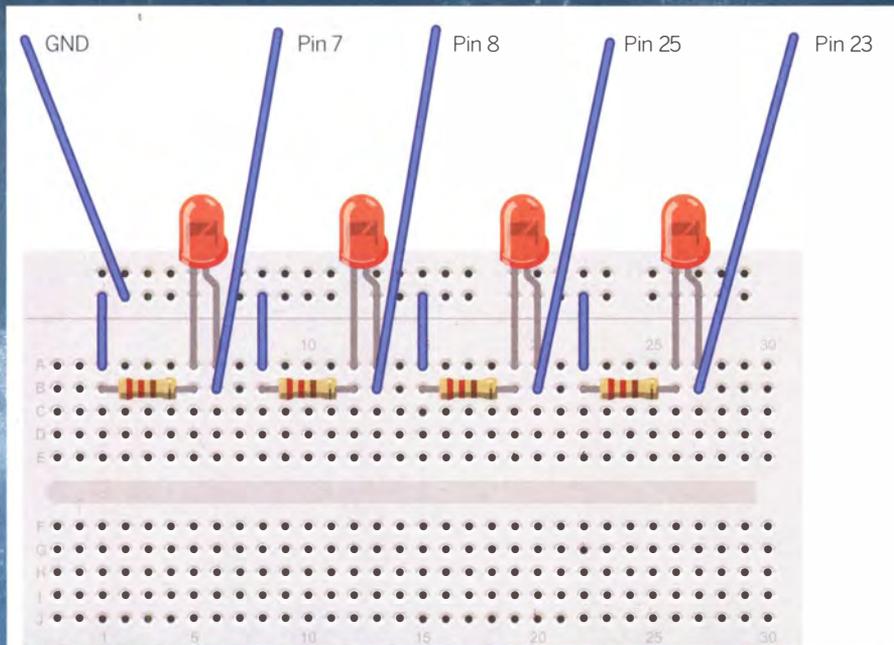


Abbildung 1: Hier Sehen Sie, wie wir die einzelnen Bauteile auf einer Steckplatine angeordnet haben. Die übrigen vier LEDs werden in gleicher Weise verdrahtet.

## Das ohmsche Gesetz

Elektrizität wird in zwei physikalischen Größen gemessen: Spannung und Stromstärke. Die Spannung – die in Volt (V) gemessen wird – ist die Menge Energie, die eine bestimmte Anzahl von Elektronen besitzt, während die Stromstärke – die in Ampere (A) gemessen wird – die Menge an Elektronen angibt, die einen Punkt passiert. Im ohmschen Gesetz wird ein Zusammenhang zwischen den beiden Größen hergestellt:

Spannung = Stromstärke x Widerstand (oder als Formel:  $U = IR$ ). Sie können dieses Gesetz anwenden, um zu verhindern, dass Ihr Raspberry Pi zu Toast wird, indem zu viel Strom in ihn hineinfließt. Das Setup des Pi ist in dieser Hinsicht recht komplex. Gert van Loo, einer der Entwickler des Geräts, erklärt es unter <http://bit.ly/Qp4PMI> ausführlich. Aber als Faustregel lässt sich sagen, dass 3,3 V und 16 mA das Maximum an Output

und Input bei einem GPIO-Pin sind. Nach dem ohmschen Gesetz errechnet sich daraus ein Widerstand von 206,25 Ohm, der mindestens gegeben sein muss, damit der Raspberry Pi nicht beschädigt wird. Sie sollten jedoch einen Sicherheitsabstand von den erwähnten Werten einhalten. In unseren Schaltungen haben wir einen Widerstand von 1.000 Ohm benutzt, also sozusagen einen Sicherheitsfaktor von 5 eingebaut.



## Offizielles Zubehör: Gertboard

Indem Sie eine direkte Verbindung mit den GPIO-Pins Ihres Raspberry Pi herstellen, erlangen Sie eine einfache Eingabe- und Ausgabekontrolle, doch diese ist in den Möglichkeiten begrenzt. Es gibt jedoch ein sehr interessantes Zubehör für den Pi, das Sie einsetzen können, um die Interaktion sehr viel genauer zu gestalten. Das Zubehör heißt Gertboard. Es handelt sich um ein umfassendes Erweiterungsset, das

jedoch in der Basisausstattung aus Einzelteilen besteht und selbst zusammengebaut werden muss (es gibt aber auch vorgefertigte Gertboards zu kaufen). Es ist ideal zum Experimentieren geeignet, da es unter anderem mit einem Arduino (Micro-Controller), LEDs, A/D- und D/A-Wandler sowie Schaltern ausgestattet ist. Es kann in den Pi eingestöpselt werden. Der Namensgeber des Gertboards ist Gert van Loo, der an der Entwicklung des Pi beteiligt war.

wir auf einer Steckplatine angeordnet (siehe Abbildung 1).

Wenn Sie alles zusammengebaut und mit dem Pi verbunden haben, können Sie mit der Programmierung beginnen. Für den Anfang nehmen wir uns nur den letzten Pin vor, dies ist Pin Nr. 7 (die Nummerierung der Pins folgt nicht der Reihenfolge ihrer Anordnung). Öffnen Sie die Befehlszeile, und setzen Sie den Pin auf Output:

```
gpio -g mode 7 out
```

Nun können Sie den Pin aktivieren:

```
gpio -g write 7 1
```

Und deaktivieren:

```
gpio -g write 7 0
```

Jetzt können wir uns das komplette Skript vornehmen. Es besteht aus vier Teilen. Der erste sorgt lediglich dafür, dass sich die einzelnen Pins im korrekten Modus befinden und ausgeschaltet sind:

```
pins="7 8 25 24 23 18 15 14"
```

```
for x in $pins
```

```
do
```

```
    gpio -g mode $x out
```

```
    gpio -g write $x 0
```

```
done
```

Im zweiten Teil wird mithilfe von **ifconfig** die IP-Adresse des Pi ausgelesen, ins Binärformat umgewandelt und gegebenenfalls mit Führungsnullen aufgefüllt.

```
ipaddress='ifconfig eth0 | grep ,inet , |
```

```
awk ,{print $2}' | cut -f4 -d'.'
```

```
binary='echo
```

```
    „ibase=10;obase=2;$ipaddress“ | bc'
```

```
paddedBinary='printf %08d $binary'
```

Im dritten Teil extrahieren wir mittels **cut**-Befehl den gewünschten Teil aus dem Binärstring und leiten die Information an den zuständigen Pin weiter.

```
bit=1
```

```
for x in $pins
```

```
do
```

```
    out='echo $paddedBinary | cut
```

```
-b$bit'
```

› **Abbildung 3:** Hier eine schematische Darstellung unseres Schaltplans. GND steht für die Erdung.

```
    gpio -g write $x $out
```

```
    bit=$((bit+1))
```

```
done
```

Im letzten Teil weisen wir das Skript an, 5 Minuten lang zu schlafen und dann die LEDs auszuschalten.

```
sleep 5m
```

```
for x in $pins
```

```
do
```

```
    gpio -g write $x 0
```

```
done
```

Das war es auch schon. Erstellen Sie die Datei **showIP.sh**, und machen Sie sie zu

einer ausführbaren Datei:

```
chmod a+x showIP.sh
```

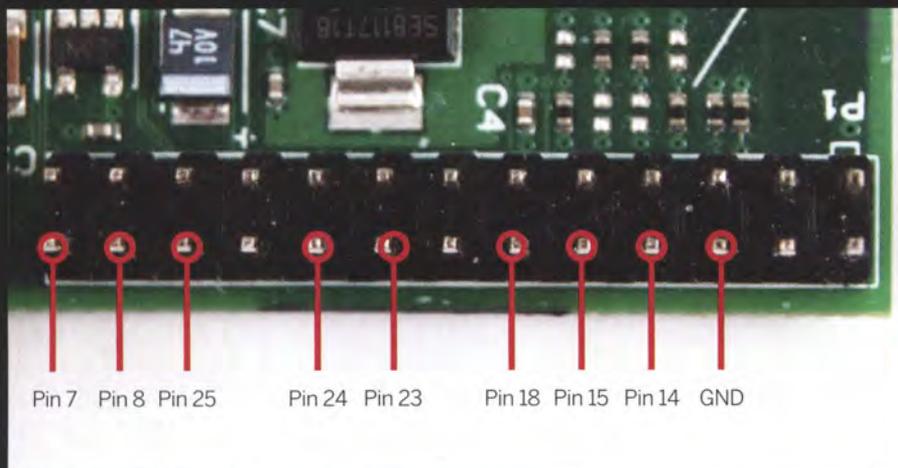
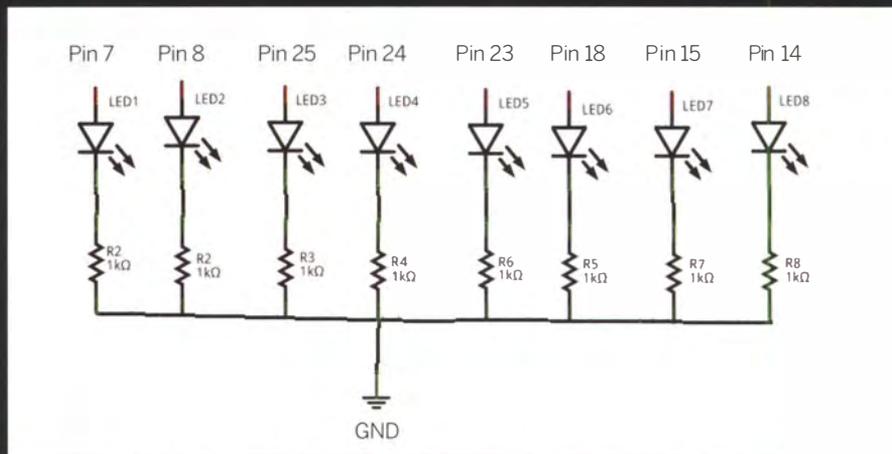
Geben Sie **sudo ./showIP.sh** ein, um die IP-Adresse anzeigen zu lassen. Um das Programm beim Booten automatisch starten zu lassen, müssen Sie in **rc.local** noch folgende Zeile einfügen (siehe dazu auch Seite 69):

```
/home/pi/showIP.sh &
```

Sie haben nun gesehen, wie Sie mithilfe der GPIO-Pins Output erzeugen können. Sie können aber, da die Pins sowohl der Aus- als auch der Eingabe dienen, ebenso gut Input verarbeiten. Dabei ist es allerdings noch wichtiger, dass nicht zu viel Strom in die Pins hineinfließt. Um einen Kontaktstift von Output auf Input umzuschalten, müssen sie dessen Modus mit **gpio -g mode <pin number> in** ändern und den Wert mit **gpio -g read <pin number>** auslesen.

Da unsere Schaltung beliebige acht Bits an Information darstellen kann, brauchen Sie sich nicht auf die Anzeige der IP-Adresse zu beschränken. Sehr detaillierte Informationen zu allen

26 GPIO-Pins des Raspberry Pi erhalten Sie unter <http://bit.ly/JTIFE3>. Die Pins, die wir mit unserem Skript benutzt haben, haben in Revision 1 und Revision 2 des Raspberry Pi die gleiche Funktion, allerdings sind einige der restlichen Pins beim Wechsel der Revisionen anders belegt worden.



› **Abbildung 2:** Verbinden Sie die Steckplatine mit den hier gezeigten Pins. Wir haben handelsübliche einpolige Stecker verwendet, Sie können aber auch zum Lötcolben greifen.

# TV-Streaming

Nach Jahren des Kampfes mit *MythTV* hat Graham Morrison dank *Tvheadend* eine bessere Lösung entdeckt.



Viele der Set-Top-Boxen, die unter Fernsehern versteckt sind, basieren auf Linux. Trotz der schwachen Prozessorleistung sind sie dazu in der Lage, mehrere Kanäle gleichzeitig abzuspielen und zu speichern sowie Daten in Ihrem Heimnetzwerk zu streamen. Der Raspberry Pi eignet sich ebenfalls perfekt dazu und kann sich – mit der richtigen Hardware – in einen starken, kostengünstigen digitalen Videorekorder verwandeln, inklusive Streaming, Aufzeichnung und zeitversetztem Fernsehen.

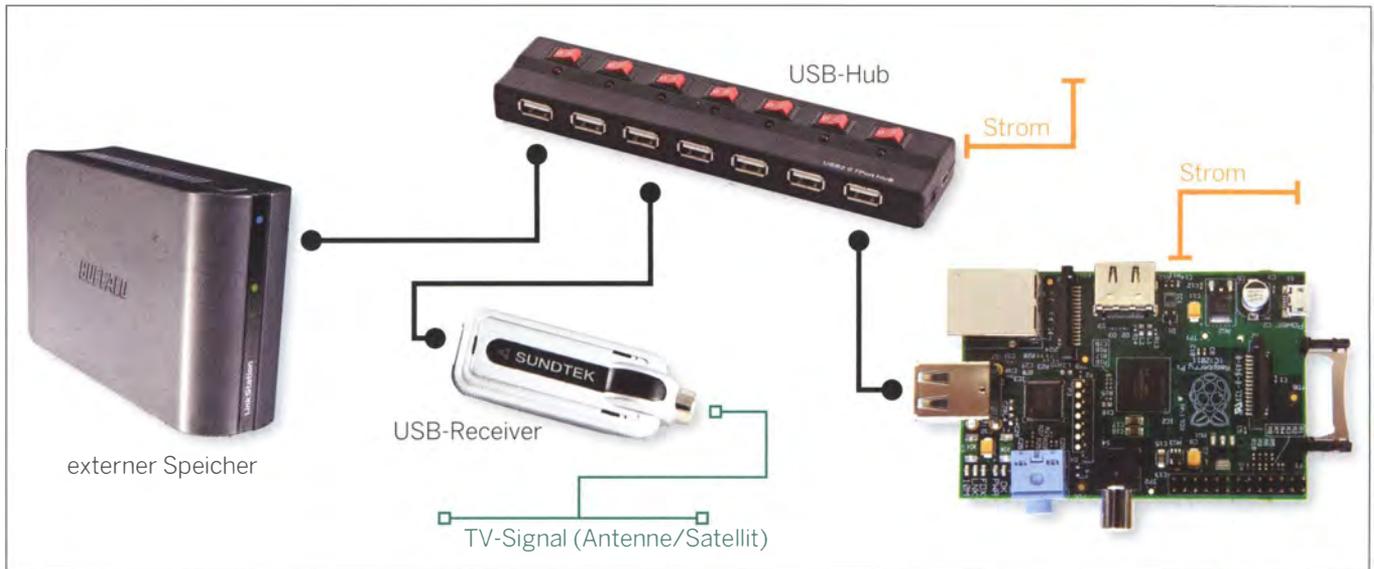
Die richtige Hardware ist der Kernbegriff des vorherigen Absatzes: Ob die Installation problemlos verläuft, hängt meist davon ab, ob Ihre Hardware zur TV-Aufnahme fehlerfrei funktioniert. Zum Glück unterstützt Linux eine große Anzahl an Geräten, sodass diese ohne Modifikationen genutzt werden können.

Damit Sie alles gut nachvollziehen können, haben wir diesen Artikel in zehn Schritte aufgeteilt, die bei der Kommandozeile beginnen. Wenn Sie die Anleitung durchgearbeitet haben, werden Sie über ein vollständiges System zur Aufzeichnung digitalen Fernsehens verfügen. Alle Dateien werden dabei über unseren kleinen Raspberry Pi abgespielt. Der Pi ist das perfekte Back-End für das vor kurzem herausgekommene *XBMC*, das Sie als Front-End benutzen können – und zwar von jedem anderen Computer Ihres Netzwerks aus.



› XBMC und Tvheadend bilden auf dem Raspberry Pi die perfekte Kombination.

# mit dem Pi



## Schritt-für-Schritt-Anleitung

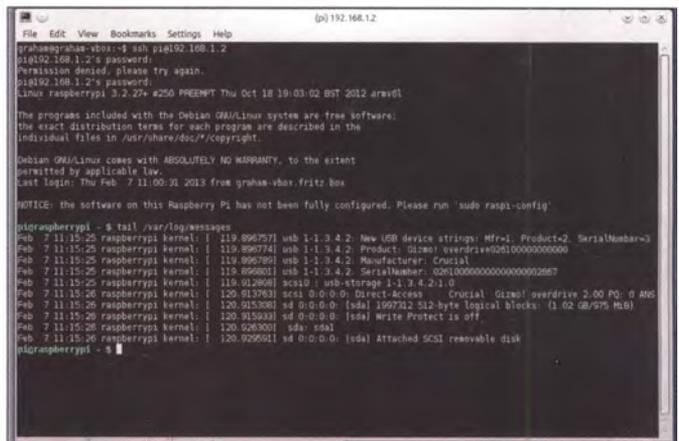
### 1 Unsere Hardware

Wir haben zwei USB-Receiver getestet und konfiguriert – den einen zum Empfang digitalen Fernsehens über eine Antenne, den anderen zum Empfang via Satellit. In diesem Artikel geben wir Ihnen Anweisungen zu beiden Empfangsarten. Für den DVB-T-Empfang (Antenne) haben wir den Sundtek MediaTV Pro und für den DVB-S-Empfang (Satellit) den Sundtek SkyTV Ultimate benutzt. Letzterer enthält einen 12-Volt-Adapter, den Sie auch anschließen müssen. Aber das Wichtigste ist: Schließen Sie die Geräte über einen USB-Hub mit externer Stromversorgung an den Raspberry Pi an. Wir haben zwei Tage mit erfolglosem Herumprobieren vergeudet, da wir zuerst keinen und danach einen inkompatiblen Hub verwendeten. In beiden Fällen schien zwar alles zu funktionieren, aber die Receiver haben einfach keine TV-Kanäle beim Scannen gefunden. Der Wechsel zu einem separat betriebenen, mit dem Raspberry Pi kompatiblen Hub hat das Problem gelöst, sodass wir diesen Punkt nicht oft genug betonen können. Schließen Sie den Hub an das Stromnetz an, danach den Receiver an den Hub und den Hub an den Raspberry Pi.



### 2 Externer Speicher

Wir gehen jetzt einmal davon aus, dass die aktuelle Version des Raspbian-Betriebssystems auf Ihrem Raspberry Pi installiert ist, dass der Pi mit dem Internet verbunden ist und dass Sie Ihre Befehle entweder direkt in die Konsole oder per SSH eintippen. Unsere nächste Überlegung: Wo speichern Sie Ihre Fernsehaufzeichnungen? Wir empfehlen Ihnen, eine externe Festplatte anzuschließen, da der permanente Lese- und Schreibzugriff die SD-Karte an Ihre Grenzen bringen wird. Um auf diese Weise Speicher hinzuzufügen, schließen Sie die Festplatte einfach an einen freien USB-Port an und überprüfen Sie die Ausgabe, indem Sie `tail /var/log/messages` eintippen. Danach müsste dort so etwas wie `usb 1-1.3.4.2: New USB device stehen`. Sie müssen noch den Bezeichner überprüfen: Hier müssten Sie etwas wie `sda: sda1` finden. Dabei ist `sda` das Gerät an sich und `sda1` die Partition. Tippen Sie `sudo mkdir /mnt/storage` ein, um einen Einhängpunkt einzustellen, und `sudo mount /dev/sda1 /mnt/storage/`, um Ihre externe Festplatte damit zu verbinden.



## 3 Treiber installieren

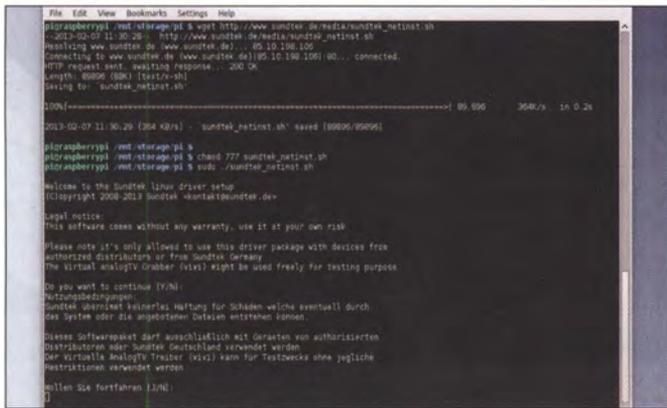
Abhängig von der Fernsehhardware, die Sie benutzen, könnte dieser Schritt überflüssig sein. Wenn Sie ein Gerät gewählt haben, das mit Linux kompatibel ist und keine zusätzlichen Treiber benötigt, dann schließen Sie es einfach an und machen Sie weiter mit Schritt 4. Wir mussten für unsere Sundtek-Geräte jedoch Treiber herunterladen und installieren. Das ist immerhin einfach – tippen Sie in die Kommandozeile:

```
wget http://www.sundtek.de/media/sundtek_netinst.sh
chmod 777 sundtek_netinst.sh
sudo ./sundtek_netinst.sh
```

Die letzte Zeile führt das Skript aus, das mittels der ersten Befehlszeile heruntergeladen wurde. Dieses ermittelt Ihr System und installiert die neueste Treiberversion. Das Skript lässt die Treiber laufen und konfiguriert sie so, dass sie beim Booten starten. Falls Sie DVB-T benutzen, müssen Sie zusätzlich folgenden Befehl ausführen:

```
/opt/bin/mediaclient --setdtvmode=DVB-T
```

Dadurch stellen Sie sicher, dass die Karte für terrestrischen Empfang und nicht für den Empfang über Kabel konfiguriert ist, wozu das Gerät ebenfalls in der Lage ist.



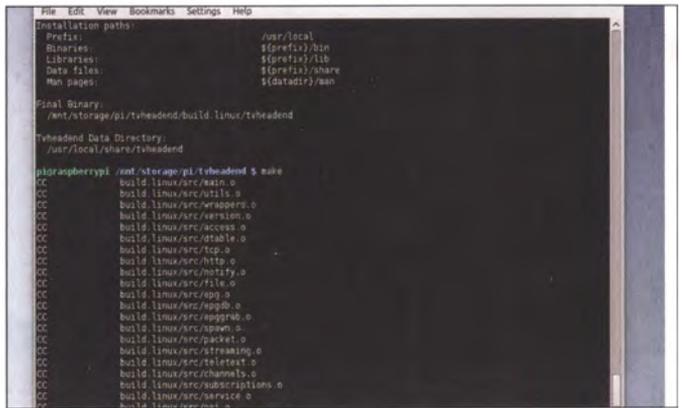
## 4 Tvheadend installieren

Mithilfe eines Plugins wird der Media-Player *XBM*C zu einem voll funktionsfähigen Videorekorder. Die Software, die wir für Aufzeichnung und Streaming des digitalen Fernsehens benutzen, nennt sich *Tvheadend*. Sie wird im Hintergrund die harte Arbeit erledigen – und zwar vom Raspberry Pi aus. Da *Tvheadend* ständig weiterentwickelt wird, haben wir die Entwicklungsversion benutzt. Sie können allerdings genauso gut die OpenElec-Distribution anstelle von Raspbian nutzen, wenn Sie Schritt 4 überspringen möchten. Zum Glück ist die Installation aber einfach. Zunächst benötigen Sie die Entwicklungs- und DVB-Tools:

```
sudo apt-get update
sudo apt-get install unzip libcurl4-openssl-dev pkg-config git
build-essential dvb-apps gcc-4.7
```

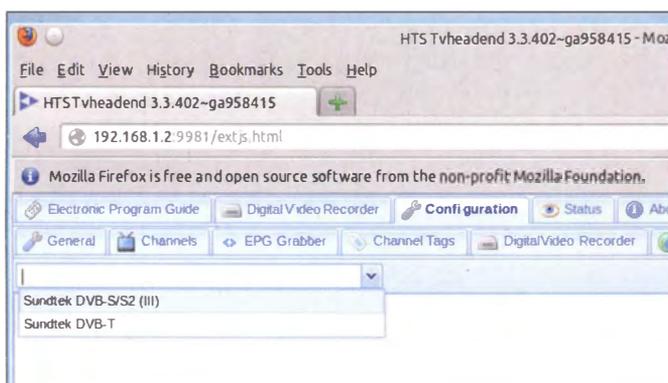
Zu guter Letzt laden Sie mithilfe von Git die neueste *Tvheadend*-Version vom Repository der Entwickler herunter und schließen den Erstellungsprozess mit dem folgenden, auf `./configure`, `make` und `sudo make install` basierenden Kommando ab:

```
CC=gcc-4.7 ./configure; make; sudo make install
```



## 5 Tvheadend konfigurieren

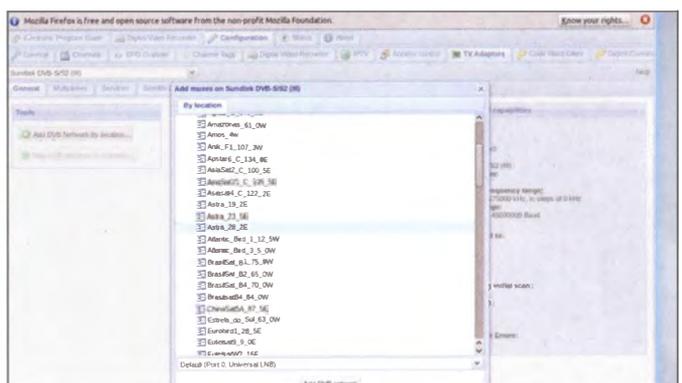
Sie greifen auf die Benutzeroberfläche von *Tvheadend* über den Webbrowser zu. Doch zuerst müssen Sie das Programm starten. Da dies das erste Mal ist, werden wir es im Konfigurationsmodus und als Daemon ausführen. Letzteres bedeutet, dass das Programm im Hintergrund läuft. Geben Sie `tvheadend -C -d` ein, öffnen Sie dann einen Browser (vorzugsweise auf einem anderen Rechner im selben Netzwerk), und tippen Sie Folgendes `http://<ip-adresse>:9981/extjs.html` ein. Die IP-Adresse Ihres Raspberry Pi finden Sie heraus, indem Sie `ifconfig` eingeben und nachsehen, was neben dem Feld „inet addr“ als Wert für das Gerät „eth1“ eingetragen ist. Ihr Browser wird das Standard-Front-End zu *Tvheadend* laden. Hier werden Sie alle Programmdaten sehen und Aufzeichnungen einstellen und anschauen. Sie müssen *Tvheadend* zuerst mitteilen, wie es die angeschlossene Hardware benutzen soll. Klicken Sie dazu auf „Configuration“ > „TV Adaptors“, und wählen Sie Ihr Gerät aus dem Drop-Down-Menü auf der linken Seite aus. Die Einstellungen, die Sie vornehmen müssen, unterscheiden sich etwas, je nachdem, ob Sie ein Satellitensignal empfangen oder eine Antenne benutzen. Wenn Sie einen Satellitenreceiver haben, fahren Sie mit Schritt 6 fort, bei Antennenempfang springen Sie direkt zu Schritt 7.



## 6 Satellitenempfang

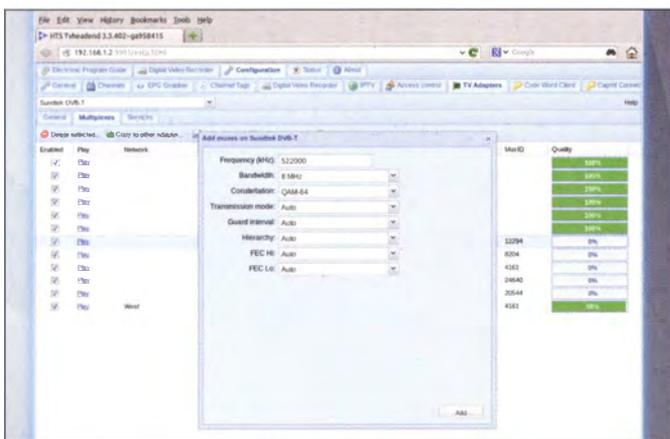
Ein Gerät mit Satellitenempfang ist am leichtesten zu konfigurieren. Nachdem Sie Ihren Adapter ausgewählt haben, sehen Sie im Reiter „General“ (Allgemein) einen Überblick der Konfigurationsmöglichkeiten. Klicken Sie zuerst auf „Enable“, dann auf „Save“. Wir wollen Kanalinformationen hinzufügen – dazu müssen wir die Daten eingeben, um einen Satelliten und dessen Bouquets (Multiplexe) zu finden, und anschließend diese Bouquets nach Kanälen durchsuchen. *Tvheadend* bündelt die Daten von Satellitenpositionen, sodass Sie nur auf die Schaltfläche „Add DVB Network by location“ (DVB-Netzwerk über Position hinzufügen) klicken müssen.

Nachdem Sie den Satelliten gewählt haben, wird *Tvheadend* eine Liste von dazugehörigen Bouquets zum Reiter „Multiplexes“ hinzufügen. Diese Liste wird nun nach Kanälen durchforstet – Sie können den Prozess im Kasten „Capabilities“ (Ressourcen) auf der rechten Seite mitverfolgen. Wenn alles funktioniert, müssten Sie hier viele Sender beziehungsweise Kanäle sehen, die zum Reiter „Services“ hinzugefügt werden. Falls Sie Ihr System nicht für den Antennenempfang konfigurieren wollen, fahren Sie mit Schritt 8 fort.



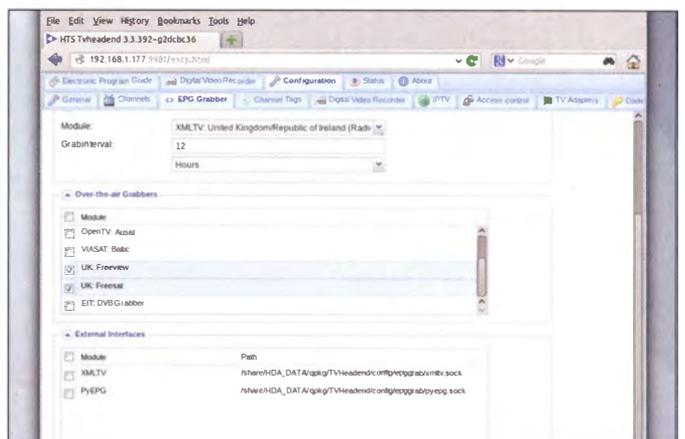
## 7 Antennenempfang

Als Erstes schalten Sie den Receiver im Reiter „General“ (Allgemein) ein. *Tvheadend* verfügt über eine umfassende Sendeanlagen-Datenbank, vorkonfiguriert mit den Daten für jedes Bouquet. Sie müssen wissen, auf welche Sendeanlage Ihre Antenne gerichtet ist. Hierbei könnte Ihnen für Deutschland der Link [www.ueberallfernsehen.de/dvbtdownloads127.pdf](http://www.ueberallfernsehen.de/dvbtdownloads127.pdf) helfen. Mit dieser Information klicken Sie einfach auf die „Add DVB Network“ (DVB-Netzwerk hinzufügen) im Reiter „General“ und finden die Anlage über „By location“ (über Position). Falls Ihre Sendeanlage nicht aufgelistet ist und Sie die Details zu den Bouquets manuell eingeben müssen, klicken Sie im Reiter „Multiplexes“ auf „Add mux(es) manually“ (Bouquets manuell hinzufügen). Im sich daraufhin öffnenden Fenster müssen Sie die Frequenz, Bandbreite und Konstellation jedes Bouquets eingeben und alles andere auf „Auto“ stellen. *Tvheadend* wird die Multiplexes nach Sendern/Kanälen scannen und sie zu Ihrer Konfiguration hinzufügen.



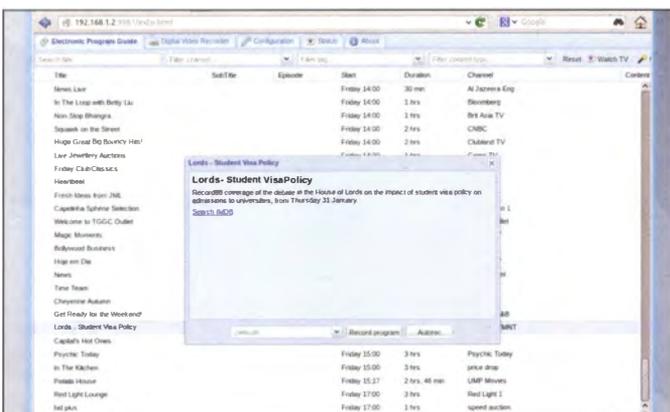
## 8 Kanäle hinzufügen

Nach dem letzten Schritt verfügen Sie hoffentlich über eine Reihe Sender. Nun müssen Sie *Tvheadend* erlauben, TV-Kanäle darin aufzuspüren, indem Sie im Reiter „General“ auf „Map DVB Services To Channels“ klicken. Im Idealfall verfügen Sie danach über eine Liste an Kanälen im Reiter „Channels“ außerhalb des Konfigurationsbereichs. Sie müssen die Angabe ändern, wo Aufzeichnungen standardmäßig gespeichert werden sollen – vermutlich zu dem Einhängepunkt, den Sie anfangs erstellt haben. Die Angabe ändern Sie, indem Sie im Reiter „Digital Video Recorder“ den Pfad für die Systemaufzeichnung („System Recording“) ändern. Um Änderungen wirksam zu machen, müssen Sie die Konfiguration des Reiters speichern. Außerdem müssen Sie den elektronischen Programmführer (EPG) konfigurieren. Normalerweise verfügt jeder Kanal nur über ein einfaches „Was läuft gerade?“ oder „Was kommt als Nächstes?“, aber manche Anbieter übermitteln eine umfangreiche 7-Tage-Programmorschau. Um diese einzuschalten, müssen Sie im Reiter „EPG Grabber“ die betreffenden Sender im Untermenü „Over-The-Air-Grabbers“ per Häkchen aktivieren.



## 9 Ein Programm aufzeichnen

Der EPG-Reiter enthält Sendungen, die Sie anschauen oder aufnehmen können. Über einen Klick auf das jeweilige Programm öffnen Sie ein neues Fenster, wo Sie Aufzeichnungen konfigurieren können. Die Option namens Autorec ist aber interessanter: Sie richtet eine Suche ein, basierend auf den Programmdateien, sodass sie ganze Serien aufnehmen können, ohne sich darauf verlassen zu müssen, dass die Daten im EPG hinterlegt sind. Abhängig von der Menge an Kanälen und dem Umfang an EPG-Daten kann die Ansicht in diesem Reiter schnell unübersichtlich werden. Sie können jedoch mithilfe der Optionen zu Beginn der Liste die Ergebnisse filtern (etwa nach Kanal, Titel oder Schlagwort). Wenn Sie einen Filter besonders oft benutzen, klicken Sie auf die Schaltfläche „Create Autorec“ – dadurch fügen Sie die Suche zu *Tvheadend* hinzu, und das Programm wird alles aufzeichnen, was zum Suchbegriff passt. Um geplante Aufnahmen zu entfernen, klicken Sie auf „Digital Video Recorder“. Kommende Aufzeichnungen können über den ersten Reiter entfernt werden, während Autorec-Suchen über den letzten Reiter gelöscht werden. Die Reiter dazwischen nutzen Sie, um vorhandene Aufzeichnungen abzuspielen oder zu löschen.



## 10 Aufzeichnungen und Fernsehen schauen

Indem Sie ein *VLC*-Plugin für Ihren Browser installieren, können Sie Aufnahmen sowie Live-Fernsehen im Browser ansehen. Wir haben dieses Feature im *Firefox* getestet: Wenn Sie ein Programm anklicken, das gesendet wird, sehen Sie die Option, dieses abzuspielen („Play“). Falls Sie das *VLC*-Plugin nicht installiert haben, werden Sie gefragt, ob Sie dies tun wollen. Nach der Installation des Zusatzmoduls erscheint ein im Browser eingebettetes Fenster, das die Sendung zeigt. Über die Kontrolleiste am oberen Rand des Fensters können Sie auf Vollbild wechseln oder die aktuelle Wiedergabe pausieren. Auf die gleiche Weise können Sie bereits aufgezeichnete Programme über den „Digital Video Recorder“-Reiter aufrufen. Wenn Sie lieber auf den Browser verzichten, ziehen Sie per Drag-and-Drop die Netzwerk-URL in Ihren *VLC*- oder einen anderen Player. Allerdings ist unserer Meinung nach die Kombination aus *XBMCC* und *Tvheadend* die beste Möglichkeit, Letzteres zu benutzen. *XBMCC* verfügt über ein Plugin, das direkt mit *Tvheadend* kommuniziert, den elektronischen Programmführer von Ihrem Pi herunterlädt und Ihnen erlaubt, Live-Kanäle anzuschauen sowie Sendungen aufzunehmen und wiederzugeben. ■



# Heimkino-PC

Verwandeln Sie Ihren Raspberry Pi in einen Home-Theatre-PC (HTPC).

**K**aum ein Einsatzgebiet ist für den Raspberry Pi besser geeignet als die kleine Nische unter dem Fernsehgerät. Es ergibt wenig Sinn, viel Geld in spezielle HTPC-Geräte wie Roku oder AppleTV zu investieren, wenn man in kürzester Zeit und mit einem schmalen Budget solch ein Gerät selber bauen kann. Die benötigte Software gibt es kostenlos, lediglich der Raspberry Pi selbst nagt am Sparsbuch.

Das OpenELEC-Projekt ist der schnellste Weg zum eigenen HTPC: Diese Distribution wurde speziell dafür entworfen, für die HTPC-Software *XBMC* eine schlanke und schnelle Basis zu sein. Darüber hinaus verlangt sie vom Benutzer keinerlei Wissen über ihre Linux-Wurzeln. *XBMC* ist eine für nahezu jedes Betriebssystem erhältliche Open-Source-Software, die selbst auf einem schwachbrüstigen Raspberry Pi alle Arten von Medien problemlos abspielen kann: seien es Fotos, Musikdateien oder Kinofilme in HD-Qualität. Dabei sieht die elegante Oberfläche auf einem Monitor ebenso gut aus wie auf einem Fernsehgerät oder einem Projektor.

## OpenELEC installieren

OpenELEC ist eine gut durchdachte Linux-Distribution, die auch mit alten Prozessoren und wenig Speicher auskommen soll. Um dieses Ziel zu erreichen, wurden für zahlreiche Plattformen speziell optimierte Versionen entwickelt, beispielsweise für Nvidia ION, Apple TV, AMD Fusion und natürlich auch für den Raspberry Pi. Für dieses Tutorial verwenden wir die plattformübergreifende Version, die mit x86Pentium4

und neueren Prozessoren funktioniert. Auf OpenELECs Download-Seite

(<http://openelec.tv/get-openelec>) finden Sie die stabile Version (<http://openelec.tv/getopenelec/viewcategory/8-generic-builds>) für Ihre Prozessorarchitektur (32-Bit oder 64-Bit). Für Ihren Raspberry Pi benötigen Sie die ARM-Version (<http://openelec.tv/get-openelec/viewcategory/10-raspberry-pi-builds>).

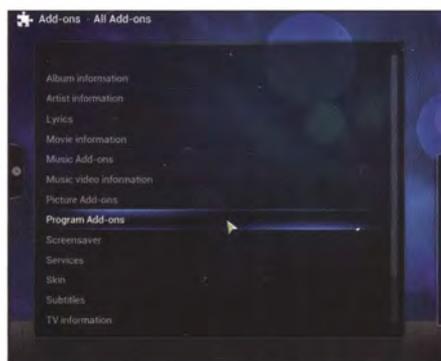
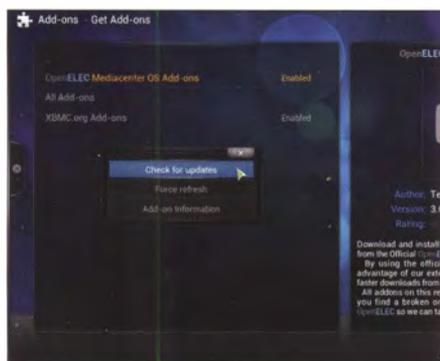
Sobald Sie die Datei heruntergeladen haben, erstellen Sie damit ein bootfähiges USB-Laufwerk, mit dem Sie dann OpenELEC auf Ihrem Computer installieren. Dazu öffnen Sie ein Terminal-Fenster und wechseln in das Verzeichnis, in dem Sie die OpenELEC-Datei gespeichert haben. Entpacken Sie das **.tar**-Archiv mit **tar xvf OpenELEC-Generic\***. Für das Raspberry-Pi-Archiv verwenden Sie stattdessen **tar xvf OpenELECRPi\***. Dies erzeugt ein neues Verzeichnis mit den entpackten Dateien, beispielsweise **OpenELEC-RPi.arm-3.2.4**.

Nun stecken Sie einen leeren USB-Stick in Ihren Rechner und finden mit **sudo parted -l** den Mount-Point heraus. Dieses Kommando listet alle Laufwerke eines Computers auf. Sie sollten den USB-Stick unter einem Namen ähnlich wie **/dev/sd\*** finden, beispielsweise **/dev/sdb** oder **/dev/sdc**. Wenn Sie sich für den falschen Namen entscheiden, dann überschreiben Sie möglicherweise wichtige Daten – seien Sie also vorsichtig!

Sobald Sie das richtige USB-Laufwerk identifiziert haben, wechseln Sie in das Verzeichnis, in das das Archiv entpackt

## Add-ons installieren

Machen Sie mehr aus Ihrem HTPC.



### 1 Aktualisieren

Sie können mit Add-ons die Fähigkeiten Ihres HTPCs nahezu beliebig erweitern. Verbinden Sie dazu das Gerät mit dem Internet und navigieren zu „System > Settings > Add-ons > Get Add-ons“. Klicken Sie rechts auf jede aktive Add-on-Quelle und wählen Sie „Check for updates“

### 2 Stöbern

Nun wählen Sie „All Add-ons“, um die soeben aktualisierte Add-on-Liste zu durchforsten. Die Add-ons sind unter diversen Kategorien wie „Musik Add-ons“ oder „Album information“ gruppiert. Klicken Sie sich einfach durch alle Kategorien, um die Vielfalt zu erforschen

### 3 Installieren

Wenn Sie ein interessantes Add-on gefunden haben, dann klicken Sie darauf. Es erscheint nun ein Informationsbildschirm, auf dem Sie das Add-on installieren, aktualisieren und anschließend gegebenenfalls konfigurieren können. Zudem lässt es sich hier jederzeit deaktivieren.

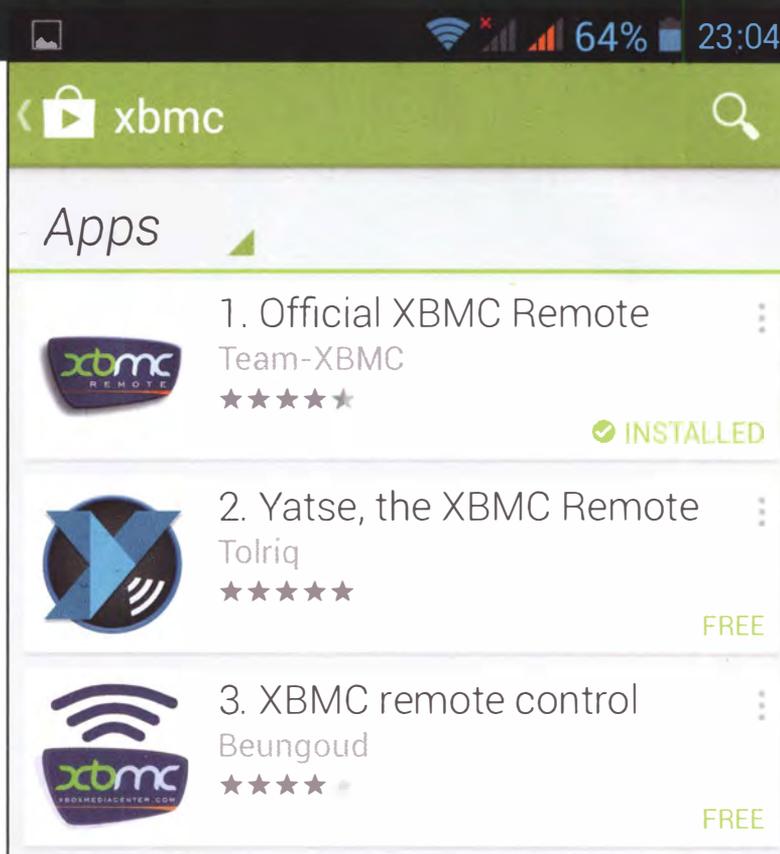
## XBMC fernsteuern

Sobald Ihr HTPC fertig eingerichtet ist, werden Sie vermutlich Tastatur und Maus loswerden wollen – im Wohnzimmer haben solche Peripheriegeräte nichts zu suchen. Sofern der HTPC an ein Netzwerk angeschlossen ist, spricht nichts dagegen, denn auch per Fernbedienung sind alle Funktionen gut erreichbar.

So sind Medienauswahl und -wiedergabe per Webbrowser und Handy-App möglich. Damit dies funktioniert, müssen Sie in „System > Settings > Services in OpenELEC“ ein paar Einstellungen vornehmen: im Abschnitt „Remote control“ aktivieren Sie „Allow programs on other systems to control XBMC“. Anschließend schalten Sie im Abschnitt „Webserver“ die Option „Allow control of XBMC via HTTP“ ein und legen einen Benutzernamen und ein Kennwort fest.

Nun starten Sie auf einem anderen Rechner einen beliebigen Webbrowser und geben in der Adresszeile die IP-Adresse des HTPCs ein, beispielsweise <http://192.168.2.111>. Nun sehen Sie das Web-Interface von XBMC. Sollte Ihnen das Design nicht zusagen, dann installieren Sie mit *XBMC-Add-ons* einfach zusätzliche Interfaces. Diese erreichen dann Sie über eine Adresse wie <http://192.168.2.111/addons>.

Noch bequemer gestaltet sich die Bedienung von XBMC mittels einer der zahllosen Fernbedienungs-Apps, die in sämtlichen App-Stores aller Smartphone-Hersteller um Ihre Gunst buhlen. Diesen ist lediglich gemein, dass Sie nach der Installation die IP-Adresse des HTPCs, Benutzername und Kennwort eingeben müssen. Wie die Bedienung der Apps tatsächlich umgesetzt ist, unterscheidet sich hingegen stark.



› Im Google Play Store finden sich zahlreiche XBMC-Fernbedienungs-Apps.

wurde, und kopieren dessen Dateien auf den Stick. Hierzu verwenden Sie das `create_livestick`-Skript, also `sudo ./create_livestick /dev/sd*`. Vergessen Sie nicht, `/dev/sd*` mit dem tatsächlichen Laufwerksnamen des USB-Sticks zu ersetzen. Das Skript startet das OpenELEC-Installationsprogramm, welches den USB-Stick formatiert und die Dateien kopiert. Sobald es fertig ist, können Sie den Stick entfernen. Schließen Sie ihn nun an den Computer an, den Sie in einen HTPC verwandeln möchten.

Nun booten Sie diesen Computer und schon nach wenigen Sekunden werden Sie von OpenELEC-Installationsbildschirm begrüßt und vor die Wahl gestellt, weiterhin von USB-Stick zu starten („Live-Mode“) oder die Daten auf der lokalen Festplatte zu installieren. Wählen Sie die zweite Option und drücken Sie die Eingabetaste.

Der nun folgende Bildschirm klärt Sie darüber auf, dass es sich um eine Beta-Software handelt. Da wir daran nichts ändern können, begeben wir uns gleich auf den nächsten Bildschirm, der uns vor weitere Entscheidungen stellt. Einige Optionen wie „Setup OpenELEC“ sind noch in Entwicklung und derzeit nicht verfügbar. Die einzig wirklich funktionierende Option ist glücklicherweise auch diejenige, die wir benötigen: die „Quick Install“-Option.

Nachdem Sie diese angewählt haben, zeigt das Installationsprogramm die Liste der lokalen Festplatten. Wählen Sie diejenige, auf der Sie OpenELEC installieren möchten. In den nächsten Minuten löscht das Programm diese Festplatte, erstellt eine Partition und kopiert alle für den HTPC-Betrieb benötigten Dateien. Anschließend landen Sie wieder im Hauptmenü, wo Sie mit „Reboot“ einen Neustart auslösen. Der Rechner wird nun OpenELEC und anschließend auch *XBMC* starten.

## OpenELEC benutzen

Bevor Sie *XBMC* sinnvoll einsetzen können, müssen Sie vermutlich noch einige Einstellungen am Betriebssystem vornehmen, beispielsweise die Netzwerkkonfiguration. Zu diesem Zweck zeigt OpenELEC beim ersten Start ein Konfigurationsprogramm an. Auf der ersten Seite werden Sie nach den Regionaleinstellungen inklusive Ihrer bevorzugten Sprache und dem Tastaturlayout gefragt. Auf der nächsten Seite wählen Sie einen Hostnamen aus, mit dem sich der HTPC und einige seiner Dienste im lokalen Netzwerk zu erkennen geben. Sie können den vorgegebenen Hostnamen „OpenELEC“ bedenkenlos überschreiben.

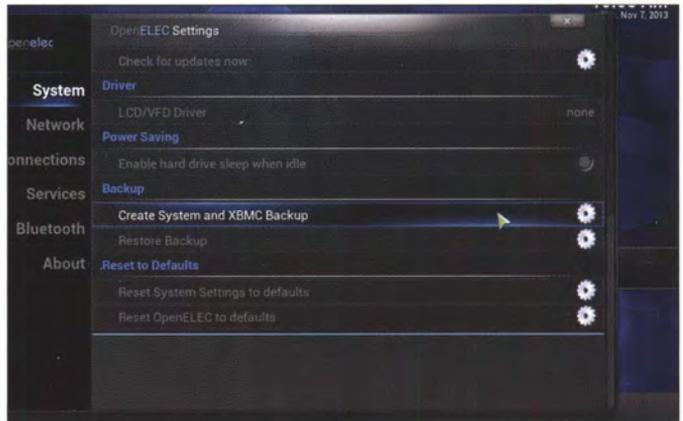
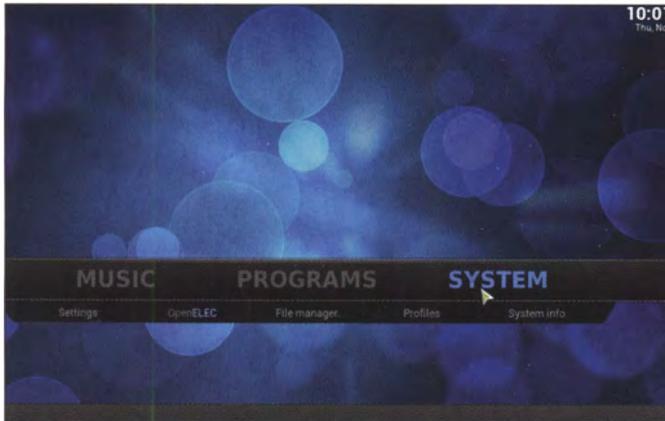
Anschließend sehen Sie eine Liste der existierenden Netzwerkverbindungen inklusive der WLAN-Accesspoints. Wählen Sie die bevorzugte Netzwerkverbindung und geben das gegebenenfalls benötigte Kennwort ein. Nachdem sich OpenELEC mit Ihrem Router verbunden hat, sehen Sie die dem HTPC zugewiesene IP-Adresse.

Im letzten Schritt schalten Sie einzelne Dienste ein oder aus, die OpenELEC für den Fernzugriff und für das Filesharing zur Verfügung stellt. Möchten Sie von anderen Computern auf die Festplatte Ihres HTPCs zugreifen, dann sollten Sie den Samba-Dienst einschalten. Wenn Sie dabei auf Sicherheit bedacht sind, dann aktivieren Sie auch den SSH-Zugriff. Sofern Sie nichts anderes angeben, sind der Benutzername „root“ und das Kennwort „openelec“.

Damit sind die Vorbereitungen abgeschlossen und Sie können Tastatur und Maus von Ihrem Raspberry Pi trennen. Schließen Sie den HTPC noch an ein Fernsehgerät an und genießen Sie fortan die Möglichkeiten eines modernen Heimkinos.

## OpenELEC einrichten

So bleibt der HTPC in Topform.

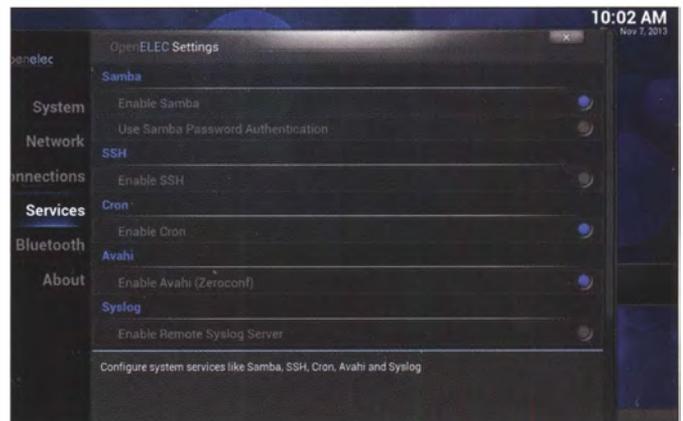
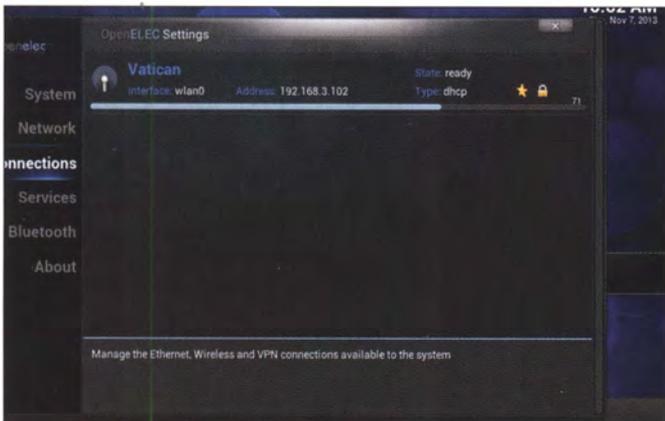


### 1 Systemeinstellungen entdecken

Zusätzlich zu den ohnehin schon zahlreichen Einstellungen von *XBMC* bringt OpenElec ein paar weitere Optionen ins Spiel. Diese finden Sie unter „System > OpenELEC“; sie verwalten einige selten benötigte, hardwarenahe Aspekte, auf die Sie sonst nur in den Systemeinstellungen des Betriebssystems Zugriff hätten.

### 2 Update and Backup

Im System-Abschnitt verändern Sie den Hostnamen und das Tastaturlayout, das Sie bei der Erstinstallation angegeben haben. Hier finden Sie auch System-Updates, können ein Backup des gesamten HTPCs erstellen und einen Stromsparmodus wählen. Im Notfall stellen Sie hier alle Optionen auf die Werkseinstellungen zurück.



### 3 Netzwerk und Verbindungen

In den Network- und Connections-Abschnitten haben Sie Zugriff auf alle Ethernet- und WLAN-Netzwerkverbindungen. Sie können jede einzelne Verbindung konfigurieren bzw. ignorieren; sogar der Zugriff auf VPN-Verbindungen ist hier möglich. Und wenn jede Sekunde zählt: NTP-Server synchronisieren jederzeit die Systemuhr.

### 4 Dienste aktivieren

OpenELEC unterstützt diverse Netzwerkdienste wie beispielsweise SSH, Cron und Avahi. Am wichtigsten ist vermutlich der Samba-Dienst, der den externen Zugriff auf die lokalen Festplatten ermöglicht. Außerdem finden Sie hier Schalter für LCD-Displays und OBEX-Bluetooth.

## Raspberry Pi als Media Center

Der Raspberry Pi erweist sich als überaus leistungsfähiger HTPC. Dank der HDMI-Schnittstelle ist Full-HD-Auflösung von 1.920 x 1.080 Bildpunkten möglich, zudem erlaubt das Kabel auch eine hochwertige Audiowiedergabe.

So ist es wenig überraschend, dass es mittlerweile zahlreiche Möglichkeiten gibt, den Raspberry Pi in einen Home-Theatre-PC zu verwandeln. Die hier beschriebene Lösung mit OpenELEC ist bequem, beraubt den Raspberry Pi aber

der Fähigkeiten eines vollwertigen Computers. Die **Raspbmc**-Distribution ([www.raspbmc.com](http://www.raspbmc.com)) und auch **XBian** ([www.xbian.org](http://www.xbian.org)) zielen in die gleiche Richtung. Ein anderer Ansatz ist daher, **XBMC** selbst herunterzuladen ([www.xbmc.org](http://www.xbmc.org)) und auf einer bereits vorhandenen Distribution zu installieren. Auch spezielle, wohnzimmergeeignete Hardware ist mittlerweile erhältlich, beispielsweise Bluetooth-Tastaturen und -Mäuse.



➤ Element14 bietet ein Raspberry-Pi-Media-Center-Kit mit der Raspbmc-Distro auf SD-Karte und Bluetooth-Tastatur an.

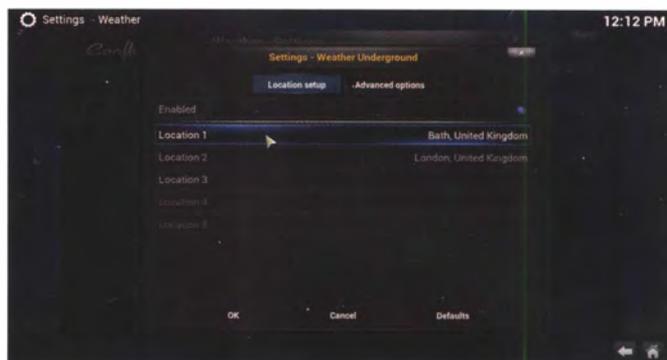
# XBMC konfigurieren

Übernehmen Sie im Heimkino die Regie.



## 1 Systeminformationen sichten

XBMC verfügt über zahlreiche Einstellungsmöglichkeiten, mit denen selbst kleinste Details des HTPCs an die eigenen Vorlieben angepasst werden können. Sie finden diese unter „System > Settings“. Im „System info“-Abschnitt sehen Sie detaillierte Informationen über die eingesetzte Hardware.



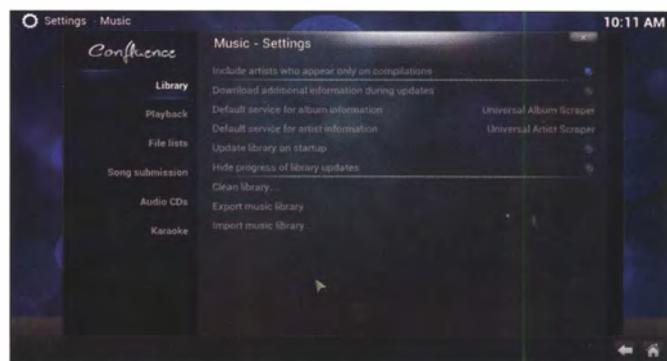
## 2 Standort konfigurieren

Im „Weather“-Abschnitt stellen Sie Ihren Standort ein. XBMC kann Wetternachrichten für mehrere Orte anzeigen, die wiederum von unterschiedlichsten Webdiensten wie „Weather Underground“ oder „Yahoo! Weather“ abgerufen werden. Der „Weather“-Abschnitt ermöglicht die Auswahl von Ort und Webdienst.



## 3 Design anpassen

Im „Appearance“-Abschnitt wird das Design der HTPC-Oberfläche angepasst. Nicht nur Details wie Sprache, Farbe, und Schriftgröße können dem eigenen Geschmack angepasst werden, auch das grundlegende Design der einzelnen Bildschirme wird mit den diversen „Skins“ von Grund auf verändert.



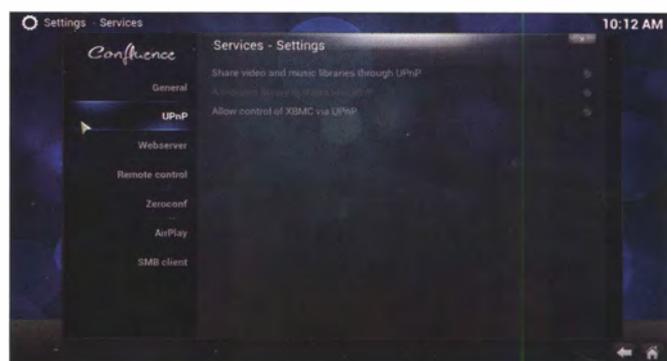
## 4 Musik-Einstellungen

Im „Music“-Abschnitt legen Sie fest, wie XBMC die in Musikdateien enthaltenen Tag-Informationen auswertet, um die Dateien in der internen Bibliothek einzusortieren. Auch die Übergänge zwischen einzelnen Musikstücken und Audiovisualisierungen sind einstellbar, ebenso der Zugriff auf Last.fm und Libre.fm.



## 5 Video-Einstellungen

Auf ähnliche Weise sind die Video-Einstellungen organisiert. So können Sie XBMC veranlassen, automatisch im Internet Filmbeschreibungen und Vorschaubilder für Schauspieler zu finden. Auch der Umgang mit Untertiteln kann hier eingestellt werden, ebenso die Anpassung an die diversen Video-Seitenverhältnisse.



## 6 Dienste aktivieren

Der „Services“-Abschnitt zeigt einige interessante Netzwerkdienste. Hier kann man den integrierten UPnP-Server einschalten, um Medien an andere Geräte zu streamen. Ebenso kann man XBMC veranlassen, als AirPlay-Empfänger zu dienen oder mit UPnP-Fernbedienungen gesteuert zu werden.

# Musik-Server

So wird aus dem Raspberry Pi eine Netzwerk-Musikbox.

**W**elch edlere Aufgabe könnte es für einen Computer geben, als nach seiner geplanten Lebensdauer noch Musik an die heimischen Wohnzimmerlautsprecher zu schicken? Auch der Raspberry Pi eignet sich hervorragend als Musikbox und ermöglicht dank leistungsfähiger Hardware sogar den Zugriff auf unterschiedlichste Internet-Streaming-Dienste. Mit erstaunlich geringem Aufwand richten wir einen Musik-Server ein, der lokal gespeicherte Musik abspielt, aber auch auf Streaming-Services wie Spotify und Google Play zugreift. Gesteuert wird dieses Wunderwerk über einen anderen Computer oder mittels eines Smartphones.

Im Zentrum unserer Bemühungen steht der kostenlose Mopidy-Musik-Server. Mopidy kann Musik aus unterschiedlichsten Quellen abspielen, nicht zuletzt aus Tausenden Web-Radiostationen und aus Online-Diensten wie SoundCloud. Mopidy hat keine grafische Benutzeroberfläche, sondern wird mittels Fernbedienung gesteuert. Daher können Sie Mopidy auf einem Barebones-Ubuntu-Server oder einem gut versteckten Raspberry Pi installieren.

## Mopidy installieren

Zuerst weisen Sie dem Computer, der fortan als Musik-Server dienen soll, eine feste IP-Adresse zu. Dies machen Sie am einfachsten auf der Administrationsseite Ihres Routers und sollte in einer lokalen Adresse wie beispielsweise 192.168.1.100 resultieren. Nun melden Sie sich am Server an und geben Folgendes ein:

```
wget -q -O - http://apt.mopidy.com/mopidy.gpg | sudo apt-key add
```

Dies erzeugt die Authentifizierungsschlüssel für den Zugriff auf das Paket-Repository. Sobald Sie die Schlüssel heruntergeladen haben, fügen Sie das Repository mit folgendem Befehl Ihrem Paketmanager hinzu:

```
sudo wget -q -O /etc/apt/sources.list.d/mopidy.list http://apt.mopidy.com/mopidy.list
```

Nun aktualisieren Sie den Paketmanager mit **sudo apt-get update** und laden Mopidy mit **sudo apt-get install mopidy** herunter. Nachdem der Server heruntergeladen und installiert ist, starten Sie ihn mit **mopidy**. Das Programm erzeugt nun in Ihrem Home-Verzeichnis die Konfigurationsdatei: **~/.config/mopidy/mopidy.conf**.

Bevor Sie Mopidy einsetzen können, müssen Sie diese Datei an Ihre Hardware anpassen – und dazu müssen Sie Mopidy auch schon wieder beenden. Dies gelingt im Terminalfenster mit der Tastenkombination [Strg]+[C].

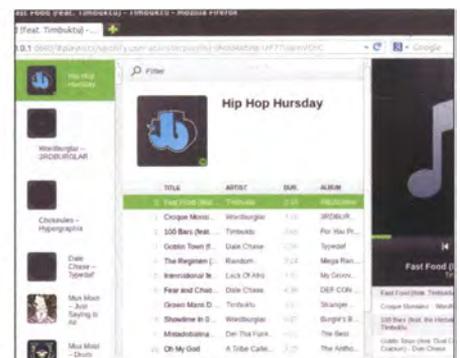
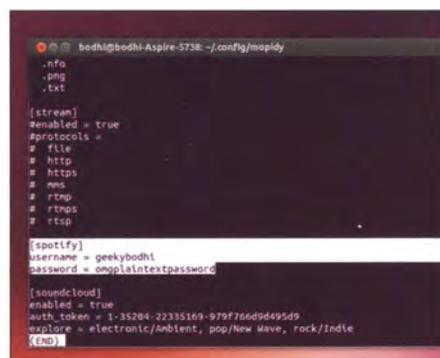
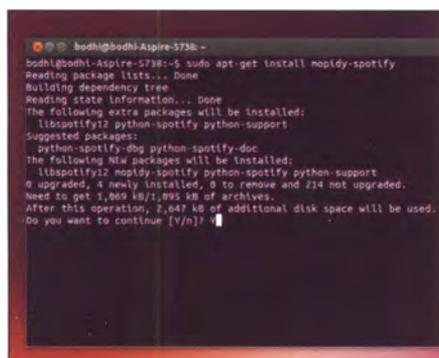
## Den Server konfigurieren

Mopidy hat zahllose Einstellungsmöglichkeiten, von denen die automatisch erzeugte Konfigurationsdatei nur die wichtigsten zeigt. Weitere Einstellungen sind auf <http://docs.mopidy.com/latest/config/> dokumentiert. Öffnen Sie nun die Datei mit einem beliebigen Texteditor.

Die Funktionalität von Mopidy stammt von sogenannten „Extensions“, von denen standardmäßig vier Stück installiert werden: „Local“ dient dazu, Musik von einer lokalen

## Von Spotify streamen

Verbinden Sie den Musik-Streaming-Service mit Ihrem Medien-Server.



### 1 Installieren

Wenn Sie dem Tutorial gefolgt sind und Ihrem Paketmanager das Mopidy-Repository hinzugefügt haben, dann installieren Sie die Spotify-Extension mit dem Befehl **sudo apt-get mopidy-spotify**. Leider benötigt die Extension einen Premium-Account von Spotify.

### 2 Mopidy konfigurieren

Wenn Sie einen Facebook-Spotify-Account besitzen, dann erzeugen Sie auf [www.spotify.com/account/set-device-password](http://www.spotify.com/account/set-device-password) ein Kennwort für Ihren Server. Erstellen Sie in der Konfigurationsdatei einen **[spotify]**-Abschnitt und tragen Sie **username** und **password** ein.

### 3 Streamen!

Um die neu gewonnenen Spotify-Dienste sinnvoll zu nutzen, verbinden Sie sich nun mit dem neu gestarteten Mopidy-Server, beispielsweise mit der Android App MPDroid oder mit einem auf Spotify zugeschnittenen Web-Client (<http://github.com/dirkgroenen/Mopify>).

# mit Mopidy

Festplatte zu lesen. Damit Mopidy weiß, wo und auf welcher Platte die Musik liegt, entfernen Sie die Kommentierung (die #-Zeichen am Anfang einer Zeile) unter dem Abschnitt **[local]**. Die **media\_dir**-Variable muss auf Ihr Musikverzeichnis weisen, also beispielsweise **media\_dir = /home/bodhi/Music**. Nun speichern Sie die Datei und veranlassen Mopidy mit dem Befehl **mopidy local scan**, im Musikverzeichnis nach Dateien zu suchen. Diesen Befehl müssen Sie jedes Mal ausführen, nachdem Sie neue Dateien in dieses Verzeichnis kopiert haben; anderenfalls weiß Mopidy nichts von deren Existenz.

Im **[stream]**-Abschnitt müssen wir nichts konfigurieren. Er dient der „Stream“-Extension dazu, Musik aus dem Internet zu lesen.

Da wir Mopidy von externen Computern aus steuern möchten, benötigen wir den „MPD“-Abschnitt. Dieser aktiviert Mopidys eigene Implementierung des beliebten Musik-Servers MPD, womit wir mit den zahlreichen MPD-Clients auf Mopidy zugreifen können.

Hierfür entfernen wir die Kommentare innerhalb des **[mpd]**-Abschnitts. Die **hostname**-Variable erhält die

IP-Adresse desjenigen Computers, der Mopidy fernsteuern darf. Da wir unser Mopidy von jedem Rechner im lokalen Netzwerk aus einsetzen möchten, setzen wir hier allerdings zwei Doppelpunkte ein: **hostname = ::**.

Falls gerade einmal kein MPD-Client zur Hand ist, lässt sich Mopidy auch mittels eines Webbrowsers steuern. Hierzu entfernen wir auch im **[http]**-Abschnitt die Kommentarzeichen und setzen die **hostname**-Variable auf den gleichen Wert wie im **[mpd]**-Abschnitt.

Starten Sie den Server nun mit **mopidy**. Ab sofort können Sie Ihr Mopidy unter der Web-Adresse <http://192.168.1.100/> (bzw. unter der Adresse, die Sie dem Computer fest zugewiesen haben) erreichen.

Damit ist der Musik-Server schon eingerichtet und betriebsbereit. Für den vollen Musikgenuss sollten Sie sich nun weitere Extensions installieren, damit Mopidy auch Zugriff auf einzelne Internet-Dienste hat und deren Musik an unterschiedlichste Ausgabegeräte weiterleiten kann. Zudem sollten Sie einen MPD-Client auf einem Computer oder Smartphone installieren, um den Server steuern zu können.

## Was ist MPD?

Mopidy basiert auf einer eigenen Implementierung des MPD-Servers. Dieser ursprüngliche MPD-Server ist weitaus leistungsfähiger als Mopidy, ist aber für unsere Zwecke überdimensioniert.

MPD steht für Music Player Daemon. Es handelt sich um die Client-Server-Version eines Musikplayers. Die Aufteilung in zwei Komponenten hat mehrere Vorteile: So kann man beispielsweise die Benutzeroberfläche beliebig austauschen, ohne die grundlegende Funktionalität zu verändern. Man kann auch mehrere Benutzeroberflächen (beispielsweise Smartphone-Apps) gleichzeitig einsetzen.

MPD arbeitet mit zahlreichen Audioformaten wie Ogg Vorbis, Flac, MP3 und allen anderen Datei- und Streamingformaten zusammen, die von der FFmpeg-Bibliothek unterstützt wer-

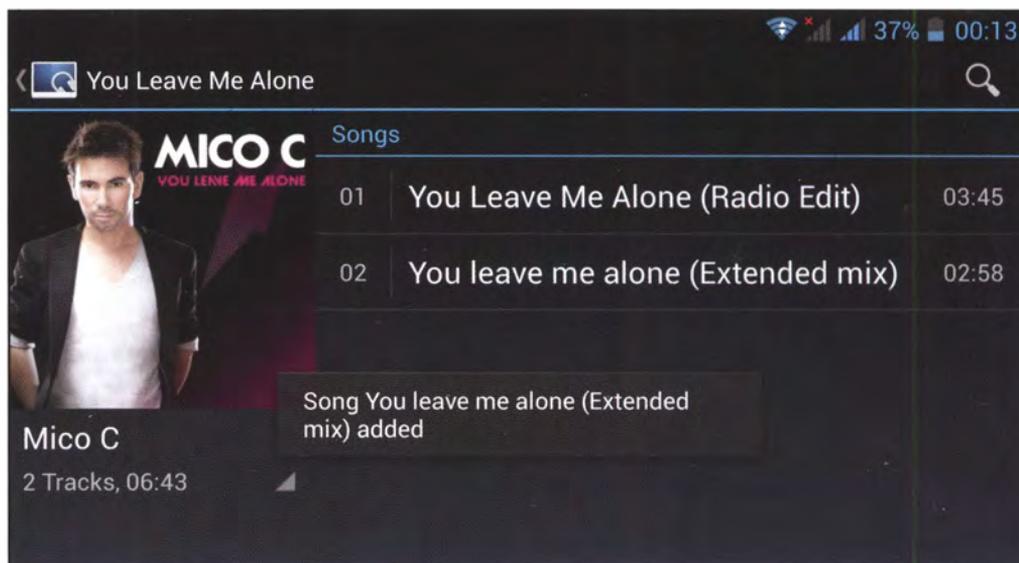
den. MPD liest und speichert Metadaten, versteht sich auf Zeroconf und kann sogar selbst als Streaming-Server dienen.

Daher ist MPD bestens für kostengünstige Rechner ohne Tastatur und Monitor geeignet. Die Clients hingegen können beliebig aufwendig und hardwareintensiv ausfallen.

Mopidy übernimmt viele der Features von MPD. So lässt sich die Wiedergabe mit jedem beliebigen MPD-Client fernsteuern und die

Bibliothek durchsuchen. Einige Features wie beispielsweise das Streamen von Musik an MPD-Clients fehlen jedoch – stattdessen ist Mopidy ein UPnP-MediaServer. Zudem kann Mopidys MPD-Variante keine Wiedergabelisten bearbeiten, was aber in einer der nächsten Versionen nachgereicht werden soll.

➤ **MPDroid ist eine bemerkenswerte Android-App, die sämtliche Features eines MPD-Servers unterstützt und auch problemlos mit Mopidy zusammenarbeitet.**



# Die wichtigsten Mopidy-Extensions

Mehr Musik, mehr Komfort

```

bodhi@bodhi-Aspire-5738: ~/config/mopidy
.nfo
.png
.txt

[stream]
#enabled = true
#protocols =
# file
# http
# https
# mms
# rtmp
# rtmps
# rtsp

[gmusic]
username = geekybodhi
password = yetanotherplaintextpassword

[soundcloud]
enabled = true
auth_token = 1-35204-22335169-979f766d9d495d9
explore = electronic/Ambient, pop/New Wave, rock/Indie
(END)
    
```

```

bodhi@bodhi-Aspire-5738: ~/config/mopidy
.nfo
.png
.txt

[stream]
#enabled = true
#protocols =
# file
# http
# https
# mms
# rtmp
# rtmps
# rtsp

[gmusic]
username = geekybodhi
password = yetanotherplaintextpassword

[soundcloud]
enabled = true
auth_token = 1-35204-22335169-979f766d9d495d9
explore = electronic/Ambient, pop/New Wave, rock/Indie
(END)
    
```

## 1 Google Play Music

Mopidy kann auf die Musik von Google Play Music zugreifen. Dazu installieren Sie den Pip-Python-Paketmanager mit **sudo apt-get install python-pip** und anschließend die Extension mit **sudo pip install mopidy-gmusic**. In der Konfigurationsdatei tragen Sie im **[gmusic]**-Abschnitt **username**, **password** und möglichst auch **deviceid** ein.

## 2 SoundCloud

Um Musik von SoundCloud zu hören, benötigen Sie ebenfalls Pip. Installieren Sie die Extension mit **sudo pip install mopidy-soundcloud**. Anschließend erhalten Sie auf **www.mopidy.com/authenticate** ein Authentifizierungstoken, das Sie im **[soundcloud]**-Abschnitt der Konfigurationsdatei der **auth\_token**-Variable zuweisen.

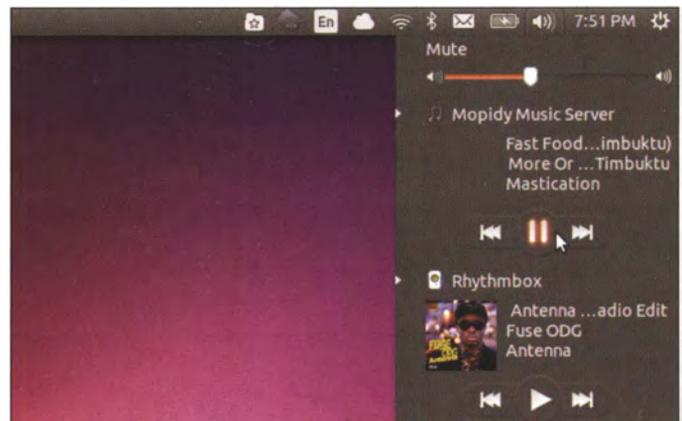
```

bodhi@bodhi-Aspire-5738: ~/config/mopidy
[stream]
#enabled = true
#protocols =
# file
# http
# https
# mms
# rtmp
# rtmps
# rtsp

[gmusic]
username = geekybodhi
password = yetanotherplaintextpassword

[soundcloud]
enabled = true
auth_token = 1-35204-22335169-979f766d9d495d9
explore = electronic/Ambient, pop/New Wave, rock/Indie
(END)

[scrobbler]
username = geekybodhi
password = 0n6l33t
    
```



## 3 Last.fm scrobbler

Mopidy kann mit Last.fm aufzeichnen, welche Musik Sie gehört haben, und anschließend Empfehlungen aussprechen. Dazu laden Sie die Extension mit **sudo pip install mopidy-scrobbler** herunter. Anschließend legen Sie sich einen kostenlosen Last.fm-Account an und tragen **username** und **password** im **[scrobbler]**-Abschnitt ein.

## 4 Ubuntu-Integration

Sie können Mopidy auch über Ubuntu's Sound-Menü (oder jede andere MPRIS-D-Bus-Software) steuern. Hierzu installieren Sie sich Pip (sofern noch nicht geschehen) sowie die MPRIS-Extension mittels **sudo pip install mopidy-mpiris**. Im Gegensatz zu den anderen Extensions benötigt diese keinerlei Konfiguration.

# Mopidy auf dem Raspberry Pi

Mopidy läuft problemlos auf allen Varianten des Raspberry Pi und findet beispielsweise in einer bestehenden Raspbian-Installation ein gutes Zuhause. Zuerst muss allerdings das IPv6-Modul aktiviert werden, was mit **sudo modprobe ipv6**, gefolgt von **echo ipv6 | sudo tee -a /etc/modules** und einem anschließenden Neustart gelingt.

Zudem gibt Mopidy Audiosignale über dem HDMI-Port aus – falls Ihre

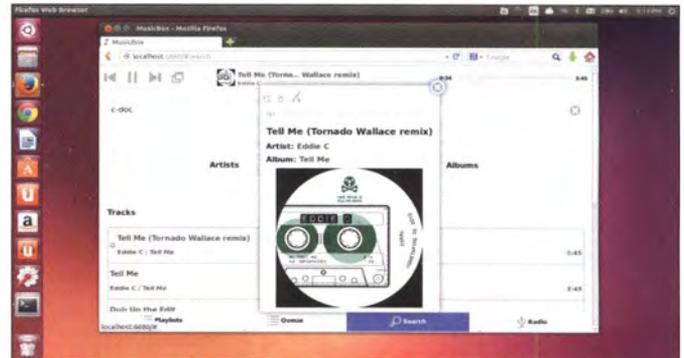
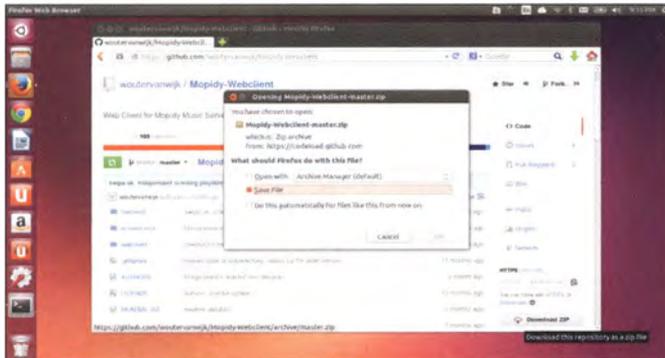
Lautsprecher aber mit dem analogen AUX-Ausgang verbunden sind, dann müssen Sie die Audiosignale mittels **sudo amixer cset numid=3 1** umleiten. Den Erfolg dieser Maßnahme testen Sie mit **aplay /usr/share/sounds/alsa/Front\_Center.wav**. Diese Einstellung hat nur dann auch nach einem Neustart noch Wirkung, wenn Sie **amixer cset numid=3 1** in die Datei **/etc/rc.local** eintragen.



Wenn Sie Mopidy nicht selbst installieren möchten, dann sollten Sie sich die Distribution Pi MusicBox ansehen.

# Clients einsetzen

## Fernsteuerung per Web, Desktop und Smartphone

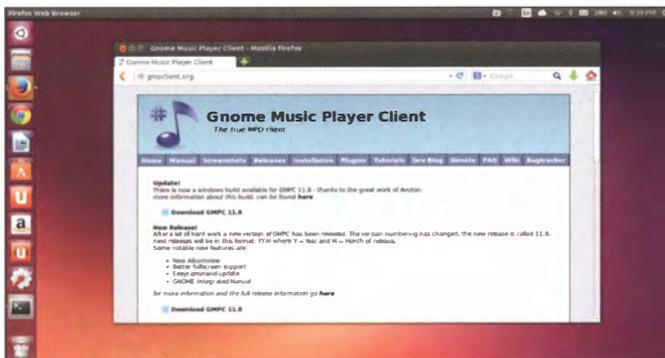


### 1 Web-Client einrichten

Mit einem Mopidy-Web-Client steuern Sie den Server über einen Browser im gleichen Netzwerk. Einer unserer Favoriten ist mopidy-webclient von <http://github.com/woutervanwijk/Mopidy-Webclient>. Nach dem Download entpacken Sie das Paket und kopieren Sie das Verzeichnis **webclient/directory** in das **/opt**-Verzeichnis.

### 2 Bedienung im Browser

Öffnen Sie Mopidys Konfigurationsdatei und geben Sie im **[http]-** Abschnitt der **static\_dir**-Variable den Wert **/opt**. Nachdem Mopidy neu gestartet wurde, ist im Browser anstelle von Mopidys Standardseite die weitaufräichere Seite des mopidy-webclient zu sehen. Die alte Seite ist unter **/mopidy** erreichbar.

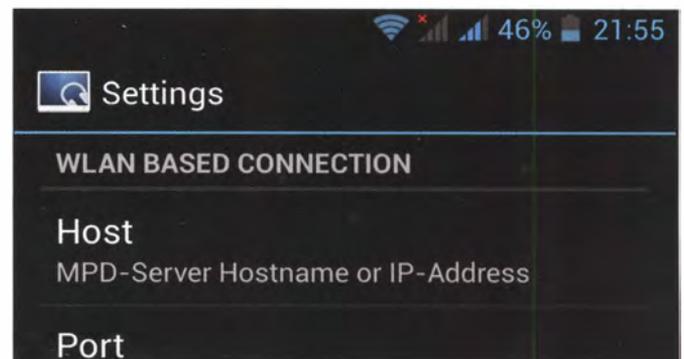
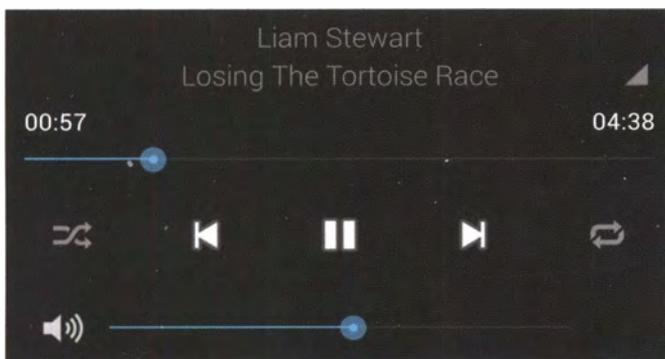


### 3 Desktop-Client einrichten

Auch für den Desktop sind mehrere Clients erhältlich, unter anderem der Gnome Music Player Client, der auch unter Windows und Mac OS X läuft. Die Installation gelingt mit **sudo add-apt-repository ppa:gmpc-trunk/gmpc-stable** und einem anschließenden Update des Paketmanagers mit **sudo apt-get update**.

### 4 Bedienung auf dem Desktop

GMPCs Plug-ins installieren Sie nun noch mit **sudo apt-get install gmpc gmpcplugins**. Nach dem ersten Programmstart führt Sie ein Assistent durch die Konfiguration. Dank Zeroconf wird der Server automatisch entdeckt und Sie müssen nichts weiter tun, als diesen auszuwählen und auf „Connect“ zu klicken.



### 5 Android-Client einrichten

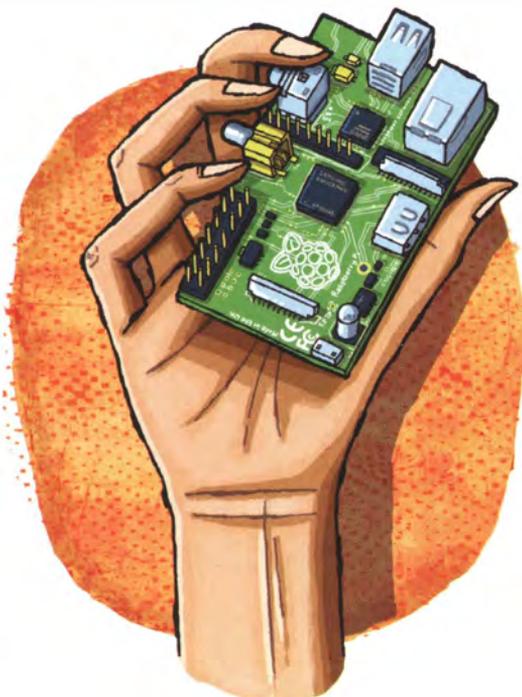
Um Mopidy mit einem Android-Gerät zu steuern, benötigen Sie einen MPD-Client. Im Google Play Store befinden sich mehrere Clients für jeden Geschmack; unser Favorit ist derzeit MPDroid. Die App ist an Telefone und Tablets angepasst und ermöglicht auch in umfangreichen Musiksammlungen eine bequeme Suche.

### 6 Bedienung auf dem Telefon

Im Rahmen der Erstkonfiguration wählen Sie die bevorzugte WLAN-Verbindung sowie Ihr drahtloses Netzwerk aus. Bei der „Host“-Einstellung tragen Sie dann noch die IP-Adresse Ihres Mopidy-Servers ein. Auf dem Hauptbildschirm können Sie anschließend Ihre Musik durchsuchen und Mopidy fernsteuern.

# Retro-Gaming

Lust auf etwas Nostalgie? Dann nutzen Sie den Pi, um die goldene Zeit der 1980er-Heimcomputer noch einmal zu durchleben.



Die Revolution des 8-Bit-Heimcomputers repräsentiert all das, was „damals“ gut war. Sie brachte in Großbritannien eine ganze Generation von Freidenkern hervor und steht auf Augenhöhe mit vielen

anderen Dingen, wofür die Insel bekannt und berühmt ist. Diese Geräte waren die Schwergewichtsweltmeister des 1980er-Jahre-Gamings und gehören heute zu den am meisten in Ehren gehaltenen aller Computer-Relikte, die sich auf Ebay tummeln.

Wenn wir die alten Tage wieder aufleben lassen wollen, müssen wir auf unseren heutigen Monster-PCs Emulatoren installieren, die der goldenen Zeit der Heimcomputer nicht ganz gerecht werden. Oder aber eine der geliebten Originalmaschinen bei Ebay ersteigern und an unseren 52-Zoll-Plasmafernseher anschließen.

Es gibt allerdings noch eine weitere Alternative – eine, die das Moderne mit dem nicht ganz so Modernen kombiniert: Der Raspberry Pi ist vermutlich eine der größten Innovationen seit dem Heimcomputer. Dieser kreditkarten-große Rechner wird schon jetzt auf die unterschiedlichsten Weisen genutzt. Projekte, bei denen der Pi an den Rand des Weltraums geschickt, zum 1940er-Jahre-Radio umgebaut oder als Fernbedienung für heimische Geräte verwendet wird, zeigen, wie vielfältig er sein kann.

Wir wollten uns davon eine Scheibe abschneiden und herausfinden, was man mit einigen auf Ebay ersteigerten Dingen, einem Raspberry Pi, Plastikfolie und ein wenig Hilfe von der stetig wachsenden Pi-Community so alles anstellen kann und ob es möglich ist, den klassischen Retro-Computer der 1980er wieder aufleben zu lassen.

## ZX-Pi

Unsere erste Zwischenstation ist ein liebenswertes Gummi-Keyboard, allgemein bekannt als Speccy. Das ZX Spectrum 48k war in den alten Tagen ein wahrer Quell der Innovation und hat nicht wenige Garagenprogrammierer über Nacht in kommerzielle Softwaregiganten verwandelt. Auch einige der meistgeliebten Spiele wurden auf diese Weise von

Jugendlichen programmiert, die nach der Schule bis zum Abendessen an Codes fummelten.

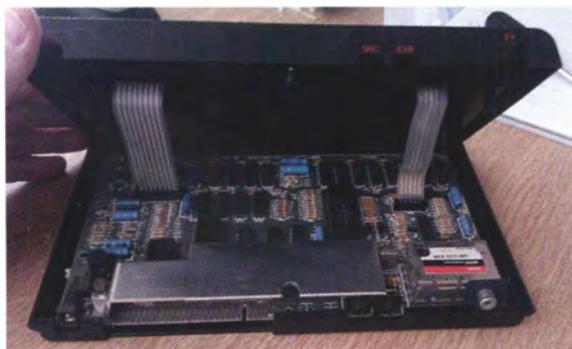
Genug jedoch der Nostalgie, die einem die Tränen in die Augen treibt. Nach kurzer Suche auf Ebay konnten wir ein totes ZX Spectrum für weniger als fünf Euro finden (ein funktionierendes Speccy auseinanderzunehmen, wäre auch ein Sakrileg!). Es sah etwas abgenutzt aus, als es bei uns ankam, aber wer von uns sich noch daran erinnern kann, mit dem Ding gespielt zu haben, sieht heute auch nicht mehr ganz frisch aus.

Als Erstes mussten wir sichergehen, dass unser Raspberry Pi auf dem neusten Stand war. Da alles fix geht in der Welt des Pi, dachten wir, es wäre eine gute Idee, auf Wheezy upzugraden. Der System-Download und eine genaue Anleitung, wie man es auf eine SD-Karte transferiert, können auf der Raspberry-Downloadseite ([goo.gl/4w4ps](http://goo.gl/4w4ps)) gefunden werden. Danach haben wir das benötigte `sudo apt-get update/upgrade` durchgeführt, und schon wenige Minuten später lief der Pi wie ein Uhrwerk.



› Das großartige ZX Spectrum. Heilig's Blechle!

# mit dem Pi



Das Innenleben des ZX Spectrum wartet darauf, herausgenommen zu werden.

Als Nächstes stand das Auseinandernehmen des ZX Spectrum auf dem Plan. Kein Problem, sobald die fünf Schrauben unter dem Gerät und die Flachbandkabel des Keyboards ab waren. Das Motherboard war mit nur einer Schraube befestigt, und nachdem wir es herausmontiert hatten, blieb lediglich das nackte Plastikgehäuse übrig.

Der Raspberry Pi ist erheblich kleiner als das originale Spectrum-Motherboard, sodass sorgsames Einpassen nötig war, um den Pi sicher und bequem in seine neue Behausung einzufügen. Dabei fanden wir heraus, dass die RCA-Video- und die Audio-Anschlüsse des Pi hervorragend mit den originalen Anschlüssen des Spectrums zusammenpassten. Nur die SD-Karte des Pi stieß am Rand des Plastikgehäuses an. Ein kleiner Schnitt mit der Kneifzange löste das Problem und erlaubte es uns, die SD-Karten auch bei geschlossenem Gehäuse auszutauschen. Durch die Vergrößerung des hausgemachten Steckplatzes kamen wir mit einem HTC-Auflader auch an die Stromversorgung heran.

Daraufhin haben wir den HDMI- und den Ethernet-Anschluss verkabelt und sowohl den Raspberry Pi als auch die



Im alten Speccy ist mehr als genug Platz für den Pi.

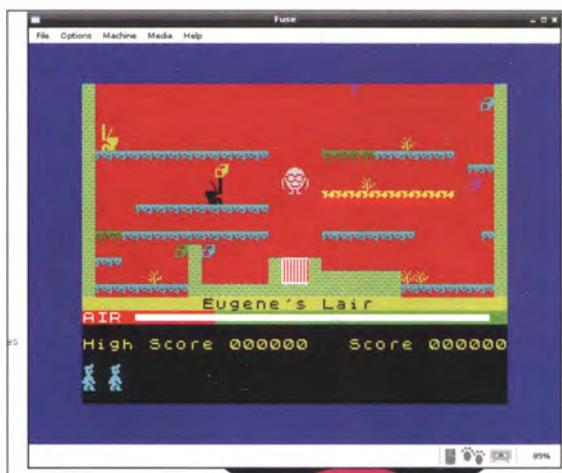


Unsere Hommage an das Spectrum: der ZX-Pi.

Kabel mithilfe von Isolierband am Boden der Spectrum-Verkleidung festgeklebt, damit alles an Ort und Stelle blieb und die Platine nicht beschädigt werden konnte. Jetzt musste nur noch das Problem mit dem Keyboard gelöst werden. Auch wenn ein Bastler namens Brian Smith das ZX Spectrum mithilfe eines BeagleBoards erfolgreich umgebaut hatte (siehe [goo.gl/V5ch3](http://goo.gl/V5ch3)), waren wir nicht wirklich erfolgreich. Mit anderen Worten: Unser Versuch ging total in die Hose. Mit den Flachbandkabeln und einem aus einer modernen USB-Tastatur ausgebauten USB-Schnittstelle kamen wir irgendwie nicht zurande. Wir haben schließlich klein beigegeben und eine klassische Tastatur und eine Maus über den großen IO-Anschluss des Spectrums angeschlossen.

Als wir den Deckel wieder aufgesetzt hatten, sah der ZX-Pi zwar nicht schlecht aus, aber doch ein bisschen frankensteinmäßig. Nachdem wir unsere gewagte Konstruktion an den Fernseher angeschlossen und diskret im Phonoschrank

»

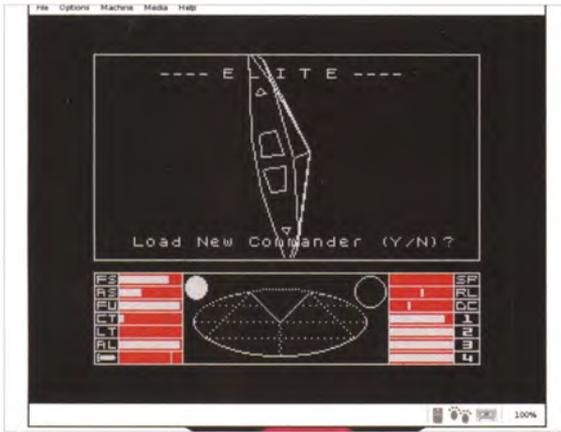


So sieht Manic Miner auf dem ZX-Pi aus.

» verstaut hatten, brauchten wir noch einen Spectrum-Emulator. Leicht zu installieren und zum Laufen zu bringen ist der Fuse Emulator. Gehen Sie dabei folgendermaßen vor:

Tippen Sie  
`sudo apt-get install fuse-emulator common`  
 in die Kommandozeile und drücken Sie Enter. Bestätigen

» **Elite, das tollste Spiel aller Zeiten? Auf jeden Fall ist unser ZX-Pi ein standesgemäßes Zuhause für diesen goldenen Oldie.**



Sie den Download und installieren Sie das Programm mit „Y“. Nach Abschluss der Installation tippen Sie das Kommando `sudo apt-get install spectrum-roms fuse-emulator-utils` und drücken Sie Enter. Zuletzt schreiben Sie `sudo amixer cset numid=3 2`

in die Kommandozeile und drücken erneut Enter. Dadurch schalten Sie den Sound über HDMI frei, allerdings ist der Klang hierbei etwas merkwürdig. Wenn Sie die „2“ gegen eine „1“ austauschen, kommt der Sound aus dem Audio-Anschluss des Raspberry Pi.

Wenn Sie fertig sind, verlassen Sie die Kommandozeile und klicken Sie den „Start LXDE“-Button. Dort navigieren Sie zu Games > Fuse Spectrum Emulator (GTK+ Version), klicken darauf und vergrößern das sich öffnende Fenster.

Nun brauchen Sie natürlich noch etwas zum Spielen. Laden Sie sich irgendwo ein altes Game herunter, beispielsweise von der Website World of Spectrum ([goo.gl/trJvd](http://goo.gl/trJvd)). Anschließend wählen Sie es im Fuse-Menü unter Media > Tape > Open aus und tippen „J“ sowie zweimal Steuerung + „P“, dann wird das Spiel geladen. Der Sound kommt aus dem Audio-Anschluss des Pi, den Sie mit Lautsprechern oder Kopfhörern verbinden können.

## Mega-Pi

Die Idee, unseren Raspberry Pi in einem Retro-Gehäuse unterzubringen, hat uns wirklich gut gefallen, auch wenn bei unserem ersten Versuch nicht alles funktionierte wie geplant. Daher hielten wir nach weiteren Retro-Maschinen Ausschau, die wir verwenden könnten. Recht schnell kamen wir auf den Sega Mega Drive, der selbst mit heutigem Blick betrachtet noch cool und schnittig aussieht – das ideale Chassis für unseren nächsten Retro-Pi.

Als Erstes mussten wir den Raspberry Pi jedoch für seinen Einsatz in einer der besten Spielkonsolen aller Zeiten vorbereiten. Wir wollten nicht nur die Spiele des Sega Mega Drive, sondern auch andere Games der damaligen Zeit auf dem Gerät zum Laufen bringen.

Zu diesem Zweck haben wir das exzellente RetroPie-Skript von petRockBlog installiert. Alles, was zu tun ist, finden Sie in der entsprechenden Anleitung unter [goo.gl/1sspF](http://goo.gl/1sspF). Bei unserem Versuch haben wir über die „Source-based (custom)“-Installationsmethode die neuesten Versionen von Programmen und Skripten benutzt. Wie auf der Seite angesprochen wird, benötigt der Vorgang mehrere Stunden, und der Pi ist

» **Die Platine passte in die Hülle, die Kabel jedoch nicht.**



» **Jede Menge Platz für den Raspberry Pi im Sega Mega Drive. Die USB-Anschlüsse vorn an der Konsole haben wir wiederverwendet.**

für diese Zeit quasi komplett ausgelastet. Doch das Warten lohnt sich. Sobald die Installation beendet ist, bekommt der Raspberry Pi in Form eines Neustarts eine wohlverdiente Pause. Danach war es für uns leicht, ein Spiel zu finden und über die Kommando-Zeile wie folgt aufzurufen:

```
retroarch -L /home/pi/RetroPie/emulatorcores/Genesis-Plus-GX/libreto.so /home/pi/RetroPie/roms/megadrive/Sonic.md
```

Bei diesem Beispiel benutzten wir eine *Sonic the Hedgehog*-ROM, die im Ordner „roms“ unter „megadrive“ lag. Natürlich müssen Sie die Eingabe entsprechend Ihrem Emulator und dem gewünschten Spiel anpassen.

Es war nicht schwierig, einen kaputten Mega Drive zu finden, der für wenig Geld angeboten wurde. Er war nicht mehr in hervorragendem Zustand und hatte seit seiner Herstellung im Jahre 1992 wohl Einiges erlebt. Anstatt dem Gerät jedoch wieder sofort das Innere herauszureißen, hatten wir eine grandiose Idee: Ob es wohl möglich wäre, den Raspberry Pi in der Hülle eines Sega Mega Drive-Spiels unterzubringen?

Die Hülle musste natürlich von *Sonic the Hedgehog* sein, und sobald wir ein Original exemplar des Spiels auftreiben konnten, machten wir uns an die Arbeit. Es sah tatsächlich so aus, als könne man den Raspberry Pi in die kleine Hülle



## Die rechtliche Situation

Die Frage der Legalität von Emulatoren ist im besten Fall eine Grauzone. Was sich jedoch mit einiger Sicherheit sagen lässt: Falls Ihnen die Konsole und das jeweilige Spiel gehört, dürfen Sie auch

eine Kopie der ROM besitzen, um das Spiel emulieren zu können. Eine Gewähr hierfür können wir jedoch nicht geben. Bitte informieren Sie sich im Zweifelsfall genauer in einschlägigen Quellen.



› Mit der eingesetzten Sonic-Hülle wird das Gehäuse zu einer optisch ansprechenden Retro-Behausung für den Raspberry Pi.

quetschen. Allerdings war kaum noch Platz für die Verkabelung, und so haben wir im Endeffekt den Raspberry Pi doch lieber im Inneren der Konsole untergebracht.

Wir achteten darauf, dass der Anschluss für die SD-Karte leicht zugänglich war, was sehr gut durch den seitlichen Erweiterungsport der Konsole funktionierte, die man obendrein mit einem Plastikverschluss sichern konnte. Zwei USB-Verlängerungskabel legten wir durch die vorderen Gamepad-Anschlüsse, das HDMI- und das Ethernet-Kabel zogen wir durch den Strom- und den TV-Anschluss auf der Rückseite des Mega Drive, und die Energieversorgung geschah über einen der seitlichen Anschlüsse.

Nachdem wir alle Innereien sicher verstaut hatten und der Raspberry Pi so befestigt war, dass er nicht an der Seite herausrutschen konnte, schraubten wir vorsichtig den Deckel der Konsole wieder an und achteten dabei darauf, dass die Schrauben keine Kabel oder Teile des Pi berührten. Anschließend bekam der neue Mega-Pi einen Platz neben dem Fernseher und wurde endlich mit Strom versorgt. Die Strom- und Lautstärke-



› Ein wohlbekannter blauer Igel läuft auch auf dem Raspberry Pi, was das Zeug hält.

Anzeigen vorn am Mega Drive funktionierten natürlich nicht, aber sie trugen ganz klar zum Retro-Stil der modernisierten 1990er-Jahre-Konsole bei. Mit unserer nun einwandfrei funktionierenden Mega-Pi stand einem Spieleabend mit Mega-Drive-Klassikern dank des *Genesis*-Emulators von RetroPie nichts mehr im Wege.

## Andere Retro-Ideen

Hier sind ein paar Anregungen, welche Retro-Projekte man sonst noch mit dem Pi durchführen könnte:

› **C64-Pi** Bauen Sie einen alten Commodore 64 um, und lassen Sie das Gerät im Commodore-Betriebssystem booten.

› **Master-Pi** Erproben Sie Ihre Umbaukünste analog zum Sega Mega Drive an einem Sega Master System.

› **Atari-Pi** Besorgen Sie sich ein altes Atari-System, und lassen Sie Ihre Magie wirken!

› **Spectrum +2** Erinnern Sie sich noch an den Spectrum +2 mit dem eingebauten Bandlaufwerk? Dort könnte ein Pi doch gut hineinpassen.

› **Keyboard-Pi** Treiben Sie eine alte Tastatur mit Din-Stecker auf, und versuchen Sie, einen Pi einzubauen.

## Win-Pi

Wir konzentrieren uns in diesem Heft ja eigentlich auf das Betriebssystem Linux, doch eine Sache mussten wir unbedingt ausprobieren: DOS 6.22 und Windows 3.1 auf dem Raspberry Pi zu installieren. Im Rahmen der virtuellen Maschine QEMU

gelang uns dies tatsächlich, wobei wir allerdings kein Image direkt aus QEMU verwendeten, sondern ein vorgebautes Image von *VirtualBox*, welches wir mit dem folgenden Kom-

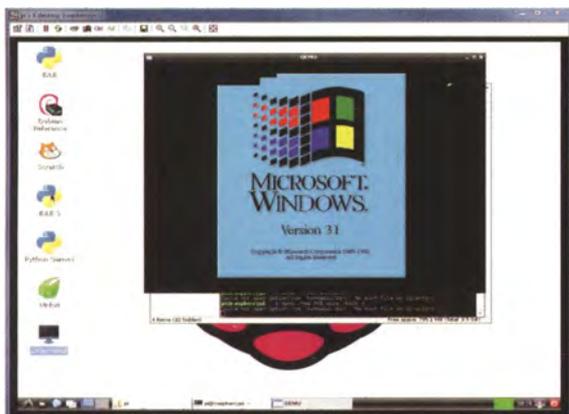
mando in eine IMG-Datei umwandelten:

```
vboxmanage clonehd "image.vdi" "image.img" --format RAW
```

Ersetzen Sie **image.vdi** und **image.img** durch den Namen Ihres Images, und konvertieren Sie anschließend mit folgendem Kommando das Image zu einem QEMU-qcow-Image:

```
qemu-img convert -f raw image.img -O qcow2 image.qcow
```

Als Ergebnis den Startbildschirm von Windows 3.1 zu sehen und den bekannten „Tada“-Sound zu hören, war irgendwie ein erhebendes Gefühl. Da der Raspberry Pi nicht über besonders viel RAM verfügt, erinnerte die Performance an einen 30 Jahre alten Traktor. QEMU verweigerte den Dienst, bis wir alles löschten und neu aufspielten. Dann vielleicht doch lieber bei Linux bleiben..



› DOS 6.22 und Windows 3.1 auf einem Raspberry Pi. Leider lief es nicht lange, aber es lief.

## Nun sind Sie am Zug!

Eine Retro-Konsole mit dem Raspberry Pi zu bauen, hat uns großen Spaß gemacht, auch wenn es ein ziemlich amateurhaftes Unterfangen war. Machen Sie es besser, krepeln Sie Ihre Ärmel hoch, bewaffnen Sie sich mit LötKolben und Kneifzange, und legen Sie los! Wie wäre es, selbst einige Umbauten mit dem Raspberry

Pi vorzunehmen? Vielleicht schaffen Sie es ja, Teile der Original-Hardware eines Geräts aus den 1980ern oder 1990ern mit dem Pi zu verbinden. Und wenn Ihnen ein tolles Werk gelungen ist, teilen Sie es mit anderen Pi-Enthusiasten in aller Welt. Machen Sie Fotos und fachsimpeln Sie mit Gleichgesinnten.

# Seien Sie Ihr

So verwandeln Sie den Raspberry Pi in einen Webserver.

**S**ollte man seine Webseiten selber hosten, also einen eigenen Webserver betreiben? Über dieses Thema und die entsprechenden Vor- und Nachteile kann man endlos debattieren, doch das wollen wir an dieser Stelle nicht tun. Falls Sie jedoch beschließen, einen Webserver selber aufzusetzen, dann bietet der Raspberry Pi eine sehr gute Infrastruktur dafür. (Sie können auch einen normalen Desktop-PC einsetzen, das ist mit dieser Anleitung genauso machbar.)

Als Erstes müssen Sie sicherstellen, dass der Rechner, der als Webserver dienen soll, eine statische IP-Adresse besitzt. Am einfachsten geht das, indem Sie in Ihrem Router eine IP-Adresse für den Server reservieren (allerdings ist die genaue Vorgehensweise dafür bei jedem Router anders, konsultieren Sie daher die Bedienungsanleitung des Geräts).

Damit der Router den Server identifizieren kann, benötigt er die MAC-Adresse (Media-Access-Control-Adresse) von dessen Netzwerkkarte. Diese ermitteln Sie mithilfe des Kommandos **ifconfig**, das Ihnen die Netzwerkschnittstellen Ihres Rechners samt MAC-Adressen ausgibt. Suchen Sie in der Liste nach der aktiven Schnittstelle – diese hat mittels DHCP eine IP-Adresse zugeteilt bekommen. In unserem Test hatte die aktive Netzwerkschnittstelle wlan0 die MAC-Adresse 0c:ee:e6:bc:33:f6 und die IP-Adresse 192.168.2.111. Letztere nehmen wir als statische IP-Adresse.

## Infrastruktur einrichten

Sehen wir uns nun die Komponenten an, die Sie benötigen,

um einen Webserver zu betreiben. Beginnen wir mit dem Apache-Webserver. Öffnen Sie ein Terminal und geben Sie das Kommando **sudo apt-get install apache2** ein, damit wird der Apache-Server samt Abhängigkeiten heruntergeladen und installiert.

Damit ist der wichtigste Schritt bereits getan, denn im Prinzip könnten Sie nun schon einen eigenen Webserver betreiben. Testen Sie das Ganze, indem Sie die statische IP-Adresse in die Adressleiste Ihres Browsers eingeben. Sie sollten daraufhin die Apache-Testseite „Es klappt!“ angezeigt bekommen.

Um jedoch auch dynamische Webseiten hosten zu können, sind noch ein paar weitere Komponenten notwendig, nämlich eine Datenbank und eine Skriptsprache. PHP ist eine solche Skriptsprache, sie holt Daten aus einer Datenbank und erzeugt daraus dynamische Webseiten. PHP ist weit verbreitet, beispielsweise basiert auch die beliebte Blogging-Software WordPress darauf. Um PHP5 zu installieren und es an den Webserver anzukoppeln, tippen Sie Folgendes in die Kommandozeile:

```
sudo apt-get install php5 libapache2-mod-php5
```

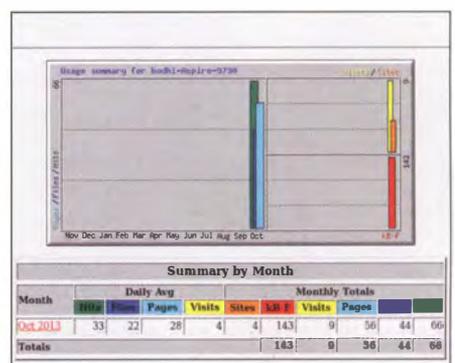
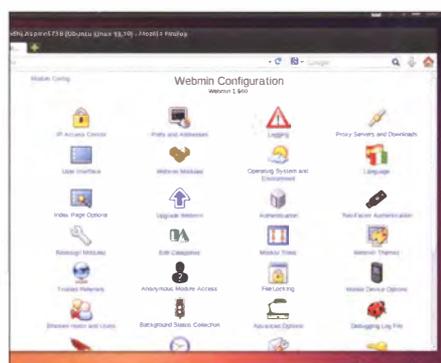
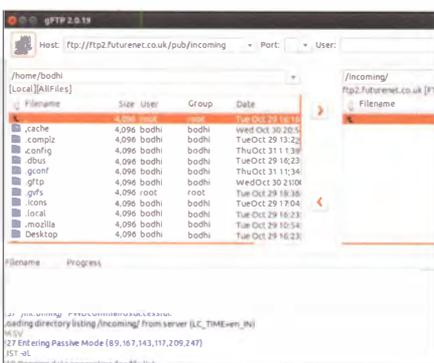
Installieren Sie zusätzlich das Paket Alternative PHP Cache (APC). Dies ist eine PHP-Erweiterung zur Verbesserung der Performance von Anwendungen, die in PHP geschrieben sind. Sie können das Paket mit folgendem Befehl installieren und einrichten:

```
sudo apt-get install php-apc
```

Anschließend müssen Sie den Webserver mit **sudo service**

## Weitere Komponenten für Ihren Webserver

Installieren Sie diese nützlichen Werkzeuge.



### 1 ProFTPD

Mit diesem Tool transferieren Sie bequem Dateien auf Ihren Webserver. Loggen Sie sich in den Server-Computer ein und tippen Sie **sudo apt-get install proftpd** in die Kommandozeile. Kontrolliert wird der Server über **/etc/proftpd/proftpd.conf**.

### 2 Webmin

Mit diesem webbasierten Tool können Sie den Webserver über den Browser konfigurieren. Sie können Benutzer hinzufügen, Konfigurationsdateien bearbeiten oder Komponenten modifizieren. Webmin ist auf [www.webmin.com/deb.html](http://www.webmin.com/deb.html) erhältlich.

### 3 Webalizer

Der Apache-Webserver verwaltet jede Menge Daten über den Server – mit Webalizer können Sie diese Informationen etwas übersichtlicher darstellen lassen. Installieren Sie das Tool mit **sudo apt-get install webalizer** und erstellen Sie Berichte mit dem Kommando **sudo webalizer**.

# eigener Host

**apache2 restart** neu starten. Um den PHP-Cache zu überprüfen, kopieren Sie die Datei **apc.php** mit dem Kommando **sudo cp /usr/share/doc/php-apc/apc.php /var/www** in das Web-Rootverzeichnis und betrachten Sie sie dann unter der Adresse **http://localhost/apc.php** im Browser.

Als Nächstes folgt die Datenbank. MySQL ist eins der verbreitetsten Datenbankverwaltungssysteme. Es funktioniert mit fast jedem Content-Management-System. Holen Sie sich die nötigen Pakete mit dem Befehl:

```
sudo apt-get install mysql-server mysql-client php5-mysql
```

Während der Installation der MySQL-Pakete wird die Festlegung eines Passworts für den MySQL-Administrator-User (root) verlangt.

Um die MySQL-Installation zu testen, können Sie sich über ein Terminal mit dem Kommando **mysql -u root -p** und dem zuvor festgelegten root-Passwort einloggen. Es erscheint ein Datenbankmonitor mit der Eingabeaufforderung **mysql>**. Mit **quit** schließen Sie den Monitor wieder und kehren zum Terminal zurück.

Nachdem alle Komponenten installiert sind, starten Sie den Apache-Webserver mit **sudo service apache2 restart** neu.

## Datenbank verwalten

Der Bestandteil Ihres Webservers, mit dem Sie vermutlich am häufigsten zu tun haben werden, ist der Datenbankserver, sei es, um neue Benutzer einzurichten oder Datenbanken und deren Tabellen zu verwalten. Sie können sich hierbei die Arbeit erleichtern, wenn Sie das Tool phpMyAdmin

verwenden – ein grafisches Interface, das im Browser läuft. Sie können es mit **sudo apt-get install phpmyadmin** installieren. Während des Vorgangs werden Sie aufgefordert, einen Webserver auszuwählen, der automatisch für phpMyAdmin konfiguriert wird – wählen Sie den Eintrag **apache2** aus. Um die Verbindung zu einer Datenbank herzustellen zu können, benötigen Sie wieder das root-Passwort. Nach der Eingabe werden Sie aufgefordert, noch ein gesondertes Passwort für phpMyAdmin selbst festzulegen.

Das Setup-Programm legt eine Datei **phpmyadmin.conf** im Verzeichnis **/etc/apache2/conf-enabled** an, die auf die phpMyAdmin-Installation unter **/usr/share/phpmyadmin** verweist. Um auf das Tool zuzugreifen, geben Sie **http://localhost/phpmyadmin** in die Adresszeile des Browsers ein und loggen Sie sich mit dem Benutzernamen **phpmyadmin** und dem beim Setup festgelegten Passwort ein.

Sollte phpMyAdmin eine Fehlermeldung unten auf der Seite ausgeben, beheben Sie das Problem mithilfe der Anweisungen auf der Webseite **https://bugs.launchpad.net/ubuntu/+source/phpmyadmin/+bug/1236035**.

Das war es auch schon, Ihr eigener Webserver läuft! Sie können den Monitor wieder vom Raspberry Pi abhängen, denn solange der Rechner mit dem Netzwerk verbunden ist, haben Sie per SSH-Login und FTP-Dateitransfer Fernzugriff auf den Server.

Sie können nun noch die zusätzlichen Komponenten installieren, die in diesem Artikel beschrieben werden.

## Apache im Detail

Auch wenn er im Handumdrehen zu installieren ist – ein Webserver ist eine komplexe Software. Bei Ubuntu wird Apache im Verzeichnis **/etc/apache2** installiert, dieses enthält die Haupt-Konfigurationsdatei **apache2.conf**. Beim Hochfahren liest der Server die Anweisungen dieser Datei aus, ebenso wie die Informationen der Datei **ports.conf**, welche die Ports angibt, die auf hereinkommende Verbindungen hin überwacht werden sollen.

Daneben gibt es die Verzeichnisse **conf-enabled**, **mods-enabled** und **sites-enabled**, diese beinhalten unterschiedliche Konfigurationsvarianten für die Verwaltung von Modulen, phpMyAdmin und anderen Komponenten. Falls Sie den Webserver konfigurieren müssen, fügen Sie einen Eintrag in einer Datei im passenden Verzeichnis hinzu, anstatt Änderungen in der **apache2.conf** vorzunehmen.

➤ Auf **http://modules.apache.org** finden Sie zahlreiche Erweiterungsmodule.

Sollten Sie beispielsweise beim Hochfahren eine Fehlermeldung bezüglich des Domainnamens erhalten, passen Sie diesen im Parameter **ServerName** an. Mit dem Kommando **\$ echo "ServerName localhost" | sudo tee /etc/apache2/conf-enabled/servername.conf** erzeugen Sie hierzu eine Datei **servername.conf**.

**conf** im Verzeichnis **conf-enabled**.

Das Verzeichnis **sites-enabled** ist mit der **VirtualHost**-Funktion des Apache-Servers verbunden, die es erlaubt, mehrere Websites mit einer Installation zu verwalten.

Konfigurationselemente und Module können Sie über **a2enconf** und **a2enmod** aktivieren.



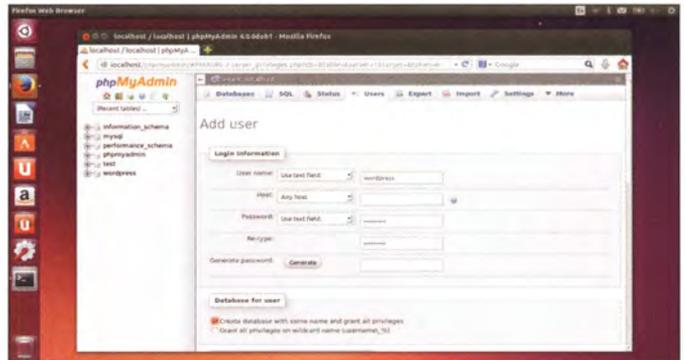
# WordPress installieren

Richten Sie die populäre Blogging-Software auf Ihrem Webserver ein.



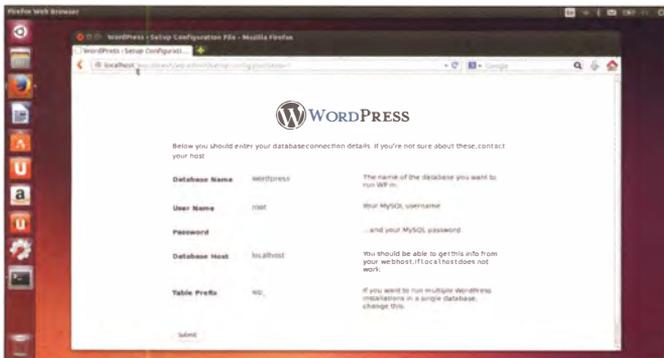
## 1 WordPress herunterladen

Loggen Sie sich in den Server-Computer ein und wechseln Sie ins Root-Verzeichnis (**cd /var/www**). Laden Sie mit **wget http://wordpress.org/latest.tar.gz** die neueste WordPress-Version herunter und entpacken Sie sie mit **sudo tar zxvf latest.tar.gz**. Die Dateien werden in den Ordner **wordpress** extrahiert. Sie können den Ordner umbenennen.



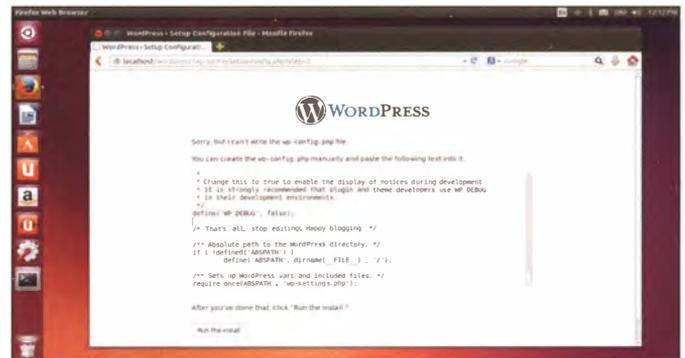
## 2 Datenbank erstellen

Nun loggen Sie sich in phpMyAdmin ein und wechseln Sie zum Reiter „Benutzer“. Legen Sie dort einen neuen Benutzer (zum Beispiel „WordPress“ oder „Blog“) samt Benutzerpasswort an und wählen Sie die Option, gleichzeitig eine Datenbank unter dem gleichen Namen zu erstellen. Tragen Sie bei den globalen Rechten „All Privileges“ ein.



## 3 WordPress-Installation vorbereiten

Um die Datenbank-Settings an WordPress zu übertragen, starten Sie den Browser und rufen Sie unter **http://localhost/wordpress/wp-admin/setup-config.php** (oder Ihrem individuellen Speicherort) die Setup-Seite von WordPress auf. Geben Sie den Namen der von Ihnen erstellten Datenbank ein, nebst MySQL-Admin-Name und -Passwort.



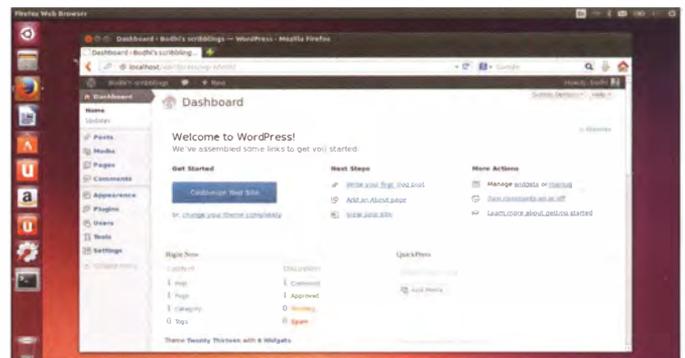
## 4 Konfigurationsdatei manuell erstellen

Mit Schritt 3 wird eine Datei namens **wp-config.php** in Ihrem WordPress-Installationsverzeichnis generiert. Sollte dies nicht der Fall sein, müssen Sie die Datei manuell erstellen. Das WordPress-Setup gibt den Inhalt aus, den Sie nur zu kopieren und in die Datei **wp-config.php** einzufügen brauchen.



## 5 WordPress-Installation durchführen

Rufen Sie nun im Browser unter **http://localhost/wp-admin/install.php** oder Ihrer entsprechenden eigenen Adresse die WordPress-Installationsdatei auf. Geben Sie dort den Namen Ihrer WordPress-Website, die Admin-Informationen, Ihre E-Mailadresse und weitere Daten ein, und klicken Sie dann auf „Installieren“.

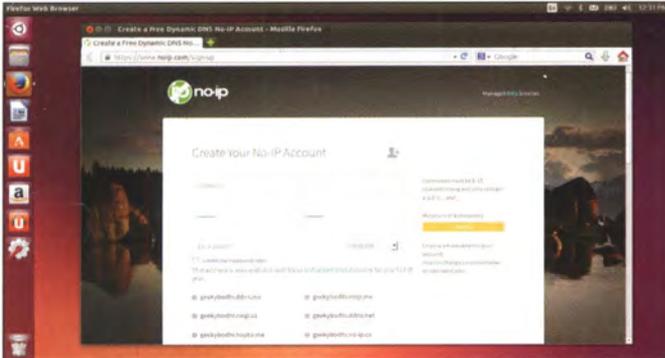


## 6 Dashboard

Fertig! Der Installer stellt die Verbindung zum Datenbankserver her und verwendet die zuvor gemachten Konfigurationsangaben, um die entsprechenden Tabellen in der WordPress-Datenbank zu generieren. Sie werden dann zur Login-Seite weitergeleitet, wo Sie sich in die WordPress-Schaltzentrale einloggen und Ihre Website aufbauen können.

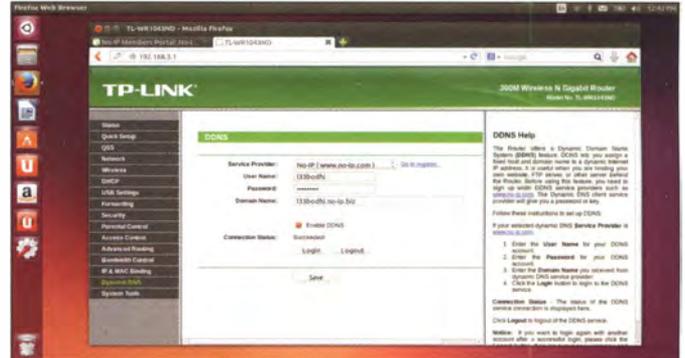
# Managen Sie Ihre Website über das Internet

So haben Sie auch ohne feste IP-Adresse von überall Zugriff darauf.



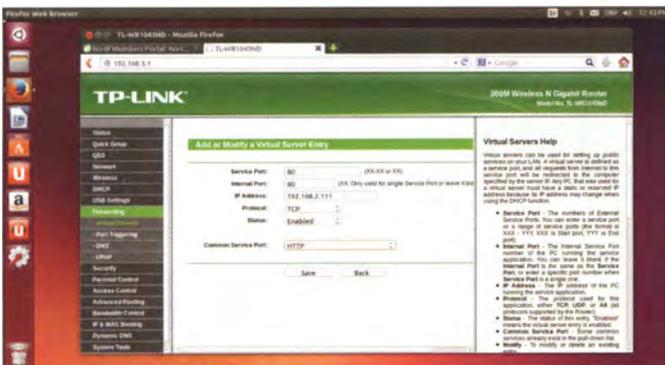
## 1 Registrieren

Erstellen Sie zunächst ein Benutzerkonto auf einer Dynamic-DNS-Website wie etwa noip.com, um Ihre lokale Website mit einem Domainnamen zu verknüpfen und Änderungen der DNS-Konfiguration nachzuhalten. Auf noip.com können Sie gratis einen Domainnamen in der Form **ihname.no-ip.org** registrieren.



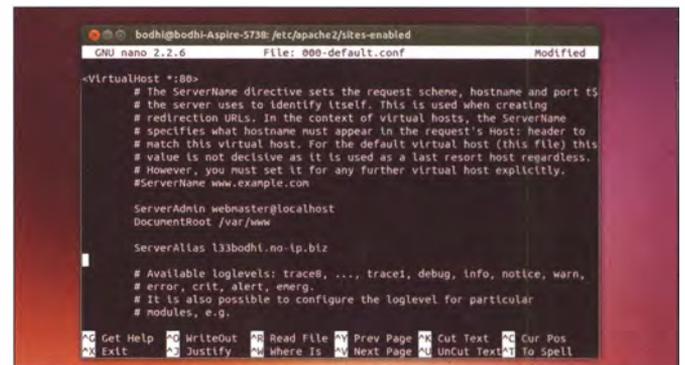
## 2 Router einrichten

Nachdem der Account eingerichtet ist, rufen Sie in Ihrem Router den Dynamic-DNS-Bereich (DDNS) auf (wo genau sich dieser befindet, ist von Gerät zu Gerät unterschiedlich). Wählen Sie den DDNS-Provider und geben Sie die Login-Daten für den Dienst sowie den registrierten Domainnamen ein. Speichern Sie die Konfiguration ab.



## 3 Ports freigeben

Anschließend müssen Sie Ports freigeben, damit der Router herein-kommende Daten durch die Firewall hindurchlassen kann. Auch hier variiert das genaue Verfahren bei verschiedenen Routern; eine Hilfestellung bekommen Sie auf <http://portforward.com>. Im Wesentlichen müssen die freizugebende Portnummer (80) sowie die lokale IP-Adresse des empfangenden Webservers eingetragen werden.



## 4 Apache aufsetzen

Im letzten Schritt muss noch die Apache-Konfigurationsdatei im Verzeichnis **sites-enabled** angepasst werden. Erstellen Sie darin einen Eintrag **ServerAlias** in der Standardkonfiguration **<VirtualHost \*:80>** wie beispielsweise **ServerAlias I33tbodhi.no-ip.org**. Starten Sie anschließend den Apache-Webserver neu. Nun sollten Sie über das Internet von überall her Zugriff auf Ihre lokale Website haben.

# Den Raspberry Pi vorbereiten

Um den Raspberry Pi als Server betreiben zu können, sollten Sie vorab auf ein paar Dinge achten. Die Distribution auf der SD-Karte sollte Raspbian sein, das ist die für den Raspberry Pi maßgeschneiderte Variante von Debian.

Nach dem Booten des Images sollten Sie zunächst mit dem Kommando **sudo raspi-config** das Konfigurationswerkzeug aufrufen und darin die Speicher-aufteilung zwischen CPU und Grafik-prozessor optimieren. Standardmäßig bekommt der Grafikprozessor eine

Quote von 64 MB RAM zugeteilt, doch bei der Verwendung des Raspberry Pi als Server können Sie diesen Wert auf das Minimum von 16 MB reduzieren.

Weiterhin sollten Sie im Konfigurationswerkzeug den SSH-Server aktivieren, um von anderen Rechnern im Netzwerk aus sicher auf den Raspberry Pi zugreifen zu können.

Starten Sie zuletzt den Raspberry Pi neu und aktualisieren Sie Raspbian mit den Befehlen **sudo apt-get update** und **sudo apt-get upgrade**.



Wichtig: Ändern Sie das Standardpasswort für den Admin-Benutzer in ein individuelles, wenn Sie Ihren Raspberry Pi als Server verwenden möchten.

# Der eigene

Mit der persönlichen Cloud bleiben Sie Herr über Ihre Daten.

**C**loud-Dienste wie Dropbox haben die Art und Weise verändert, wie Menschen auf der ganzen Welt miteinander arbeiten und Daten austauschen. Damit dies reibungslos funktioniert, müssen die Daten einem Provider anvertraut werden – in der Hoffnung, dass dieser vernünftige Datenschutzbestimmungen hat und diese dann auch einhält. Wenn aus dieser Hoffnung Gewissheit werden soll, dann bleibt Ihnen nichts anderes übrig, als einen eigenen Cloud-Dienst auf einem eigenen Server einzurichten.

Mit kostenloser Software wie ownCloud ist dies ein erfreulich einfaches Unterfangen: Mit ownCloud teilt und synchronisiert man Dateien auf allen per Internet verbundenen Geräten. Die Software verschlüsselt dabei die Dateien und kann mit zahlreichen Dateiformaten umgehen, besitzt eine Bildergalerie und einen Musikplayer. Mit der eingebauten Dateiversionierung ist sie den gängigen Lösungen sogar überlegen und kann mit einem einfachen Mausklick alte Dateiversionen wiederherstellen. Der direkte Zugriff auf die Dateien ist über jeden Webbrowser möglich, für Windows, Mac, Linux, Android und iOS gibt es spezielle Client-Software.

## ownCloud installieren

Das Set-up eines ownCloud-Servers ist in wenigen Minuten erledigt. Beginnen Sie, indem Sie dem Rechner, der fortan als Server dienen soll, eine feste IP-Adresse zuweisen. Dies geschieht normalerweise auf der Administrationsoberfläche Ihres Routers. Anschließend öffnen Sie ein Terminal-Fenster und fügen dem Paketmanager die ownCloud-Repositorys hinzu:

```
$ sudo -s
# echo 'deb http://download.opensuse.org/repositories/
isv:ownCloud:community/xUbuntu_13.10/ /' >> /etc/apt/sources.list.d/owncloud.list
```

Bevor Sie die Software installieren können, benötigen Sie den Signaturschlüssel des Repositorys:

```
# wget http://download.opensuse.org/repositories/isv:own-
Cloud:community/xUbuntu_13.10/Release.key
# apt-key add - < Release.key
```

Nun aktualisieren Sie die Repositorys mit **sudo apt-get update** und installieren ownCloud mit **sudo apt-get install owncloud**.

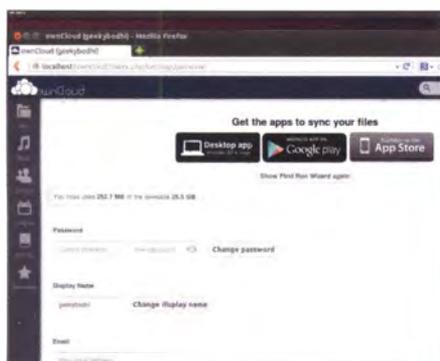
Dieser Befehl installiert nötigenfalls den Webserver Apache und konfiguriert ihn für ownCloud. Etwas Handarbeit ist dennoch nötig: Geben Sie in einem Terminalfenster **sudo a2enmod headers rewrite env** ein und starten Sie anschließend Apache mit **sudo service apache2 restart**.

Um ownCloud zu konfigurieren, starten Sie einen Webbrowser und geben **localhost/owncloud** in der Adresszeile ein. Sie werden nun aufgefordert, einen Administrator-Account anzulegen. Außerdem können Sie im „Advanced“-Abschnitt festlegen, welche Datenbank verwendet werden soll. An gleicher Stelle bestimmen Sie, in welchen Verzeichnis ownCloud seine Dateien ablegt – wenn Sie den vorgegebenen Wert **/var/www/html/owncloud/data** nicht verändern (beispielsweise in **/var/data**), dann wird ownCloud dies später mit Fehlermeldungen quittieren.

Nun können Sie sich an Ihrem ownCloud-Server als

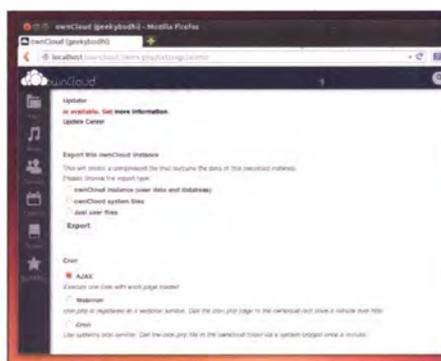
## Grundlagen der ownCloud-Konfiguration

Bringen Sie Ihrer Cloud das Fliegen bei.



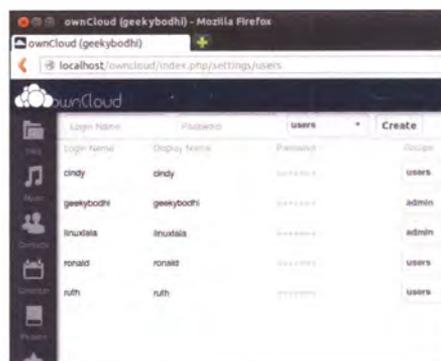
### 1 Persönliche Einstellungen

Sobald Sie sich auf Ihrem ownCloud-Server angemeldet haben, klicken Sie auf Ihren Benutzernamen und wählen Sie „Personal“. Hier können Sie jederzeit Ihren Anzeigenamen, Ihre Mail-Adresse und Ihr Passwort neu vergeben.



### 2 ownCloud verbessern

Klicken Sie erneut auf Ihren Namen und wählen Sie „Admin“. Hier dürfen Sie nach Software-Updates forschen, ein komprimiertes Backup für die komplette Cloud anstoßen und Clients die Verwendung einer verschlüsselten Verbindung vorschreiben.



### 3 Benutzerverwaltung

Bevor Sie ownCloud tatsächlich verwenden, sollten Sie Benutzer anlegen, diese in Gruppen einteilen und den Gruppen individuelle Rechte zuweisen. Hierzu klicken Sie als Administrator auf Ihren Benutzernamen und wählen „Users“.

# Cloud-Server

Administrator anmelden und Dateien hochladen, die Sie mit anderen Personen oder Geräten teilen möchten. Sollten die Dateien größer als 2 MB sein, dann müssen Sie dies vorher noch PHP mitteilen. Öffnen Sie dazu die PHP-Konfigurationsdatei **php.ini** in **/etc/php5/apache2** mit einem Texteditor. Suchen Sie die Einträge **upload\_max\_filesize** und **post\_max\_size** und verändern Sie deren Werte von 2M in 500M oder 1G.

## Hochladen und teilen

Nachdem Sie sich an Ihren ownCloud-Server angemeldet haben, befinden Sie sich in der Dateiansicht. Um Dateien hochzuladen, klicken Sie entweder auf den grünen Pfeil oder ziehen Sie die Dateien mit der Maus vom Desktop in das Browserfenster. Um ein Verzeichnis anzulegen, klicken Sie auf die „New“-Schaltfläche und wählen Sie „Folder“ aus der Liste aus.

Wenn Sie eine Datei mit einem Format hochgeladen haben, das ownCloud versteht, dann genügt ein Klick auf das Dateisymbol, um die Datei zu betrachten oder zu editieren. Die Musikansicht listet alle erkannten Audiodateien auf, unabhängig von deren Verzeichnissen. Um eine Musikdatei der Wiedergabeliste hinzuzufügen, genügt ein Klick auf das Dateisymbol. Die Wiedergabe können Sie nun mit einem Klick auf die entsprechenden Symbole am oberen Seitenrand steuern.

Gleiches gilt für die Bilderansicht, die alle erkannten Bilddateien auflistet und mit einem Klick auf ein Dateisymbol eine Diashow startet.

Eine alternative Methode für den Datei-Upload steht über das WebDAV-Protokoll zur Verfügung, mit dem sich ownCloud in das Dateisystem einklinkt. Im Dateimanager drücken Sie [Strg]+[L] und geben Sie die WebDAV-Adresse Ihres ownCloud-Servers an (beispielsweise **dav://localhost/owncloud/remote.php/webdav**). Nach der einmaligen Authentifizierung können Sie ownCloud wie jedes andere Dateiverzeichnis benutzen.

Um hochgeladene Dateien zu teilen, gehen Sie im Webbrowser in ownClouds Dateiansicht und fahren Sie mit der

Maus über die gewünschte Datei. Dies zeigt die jeweils verfügbaren Optionen an, unter anderem „Share“. Ein Klick darauf lässt Sie wählen, mit welchen Benutzern oder Benutzergruppen Sie diese Datei teilen möchten, und ob diese die Datei verändern oder löschen dürfen. Sie können Dateien sogar mit jemandem teilen, der kein Benutzerkonto auf Ihrem ownCloud-Server hat: Sobald Sie auf „Share with link“ klicken, zeigt ownCloud die Webadresse der Datei an, die Sie per Mail an jeden Empfänger senden können.

## Clients einrichten

Mit einem ownCloud-Client können Sie beliebige Verzeichnisse eines Desktop-Rechners mit Ihrem ownCloud-Server synchronisieren. Um einen Linux-Client zu installieren, fügen Sie dessen Repository dem Paketmanager hinzu:

```
$ sudo -s
# echo 'deb http://download.opensuse.org/repositories/
isv:ownCloud:devel/xUbuntu_13.10/' >> /etc/apt/sources.
list.d/owncloud-client.list
```

Dann fügen Sie die Signaturschlüssel hinzu:

```
# wget http://download.opensuse.org/repositories/isv:own-
Cloud:devel/xUbuntu_13.10/Release.key
# apt-key add - < Release.key
```

Anschließend aktualisieren Sie die Repositories mit **sudo apt-get update** und installieren Sie den Client mit **sudo apt-get install owncloud-client**. Um den Client zu konfigurieren, müssen Sie ihm lediglich die Zugangsdaten zum Server verraten, woraufhin er im lokalen home-Verzeichnis ein Verzeichnis wie beispielsweise **/home/bodhi/ownCloud** anlegt. Alle Dateien, die Sie dort ablegen, werden nun automatisch auf dem ownCloud-Server gespiegelt. Auf ähnliche Weise können Sie auch ein oder mehrere bereits bestehende lokale Verzeichnisse angeben, die gespiegelt werden sollen. Wird ein Verzeichnis von mehreren Benutzern verwendet, dann landet jede Änderung eines Benutzers automatisch auf den Rechnern der anderen Benutzer.

## Webzugriff auf ownCloud

Der Erfolg kommerzieller Cloud-Dienste wie Dropbox besteht darin, dass man auf dort gespeicherte Daten von jedem beliebigen, mit dem Internet verbundenen Computer aus zugreifen kann. Dies ist mit ownCloud zunächst nicht möglich, stattdessen können nur die Geräte im lokalen Netzwerk auf die ownCloud-Daten zugreifen. Um das auch von außerhalb zu können, benötigen Sie eine statische IP-Adresse von Ihrem Provider, zusätzlich müssen Sie Ihre Firewall so konfigurieren, dass diese Anfragen aus dem Internet zu Ihrem Server durchlässt. Alternativ zur festen IP-Adresse können Sie Dynamic DNS in Ihrem Router oder auf dem Server konfigurieren.

Eine weitere Alternative ist ein Tunneling-Dienst wie PageKite, der mit geringstem Aufwand Ihre

private Cloud für Besucher aus dem Internet verfügbar macht. Für nicht kommerzielle Benutzer gibt es eine kostenlose Variante, bei der man lediglich in einer monatlichen Umfrage seine Meinung zu PageKite kundtun muss.

Um PageKite zu installieren, geben Sie folgendes in einem Terminalfenster ein:

```
$ curl -s https://pagekite.net/pk/ | sudo bash
```

Anschließend veröffentlichen Sie mit diesem Befehl Ihren Server im Internet:

```
$ pagekite.py 80 mycloudserver.pagekite.me
```

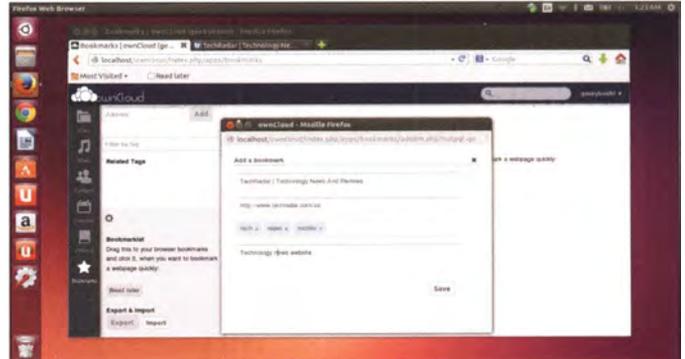
Ab sofort haben Sie Zugriff auf Ihre Cloud, indem Sie die Adresse **mycloudserver.pagekite.me** in einem Webbrowser oder einer Client-Software eingeben (**mycloudserver** müssen Sie durch einen beliebigen Namen ersetzen).



› Mit PageKite können Sie noch weitaus mehr anstellen. Die Anleitung auf [pagekite.net/support/quickstart/](http://pagekite.net/support/quickstart/) gibt einen Überblick über die Möglichkeiten.

## Apps installieren und einrichten

Mit diesen Add-ons kosten Sie die Möglichkeiten von ownCloud richtig aus.

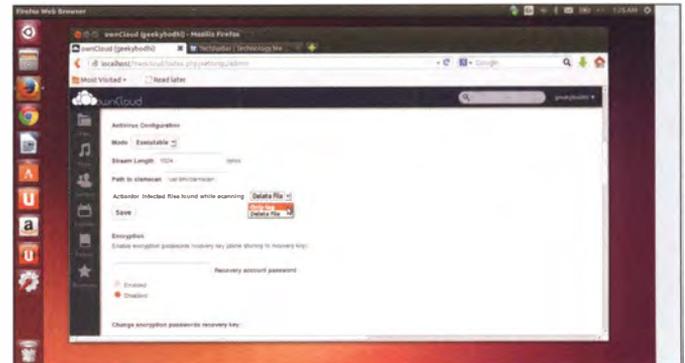
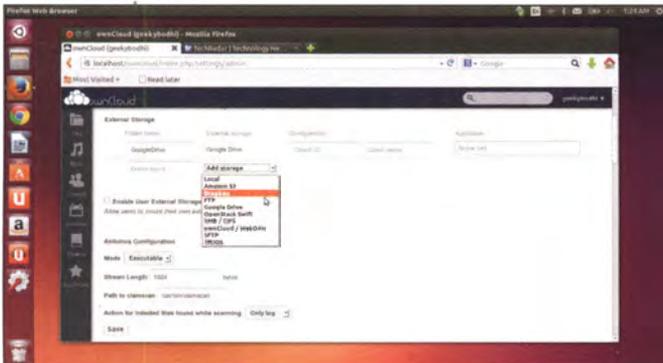


### 1 Verschlüsselung

Öffnen Sie das Pulldown-Menü neben Ihrem Benutzernamen und wählen Sie „Apps“. Mit den nun gelisteten Add-ons erweitern Sie die Möglichkeiten Ihrer ownCloud-Installation. Wählen Sie „Encryption“ und aktivieren Sie die App mit „Enable“. Beim nächsten Besuch hat die App all Ihre Daten verschlüsselt und gibt Ihnen einen Entschlüsselungscode.

### 2 Lesezeichen

Mit der „Bookmarks“-App speichern und verwalten Sie Lesezeichen in Ihrer ownCloud. Die Lesezeichen lassen sich manuell anlegen oder aus der Lesezeichendatei Ihres Webbrowsers auslesen. Die App enthält zudem ein Bookmarklet, das Sie zu den Lesezeichen Ihres Browsers hinzufügen können und die Lesezeichen synchronisiert.



### 3 Externe Cloud-Dienste

Mit „External Storage“ fügen Sie externe Festplatten, vor allem aber auch externe Cloud-Dienste wie Amazon S3 und Dropbox Ihrer ownCloud hinzu. Aktivieren Sie die App und begeben Sie sich in den neuen „External Storage“-Abschnitt im „Admin“-Bereich, wo Sie die gewünschten Dienste auswählen und deren Zugangsdaten angeben.

### 4 Virenschutz

Die App „Antivirus“ überprüft jede Datei noch während des Uploads mit ClamAV. Im „Admin“-Bereich zeigen Sie der App den Dateipfad zur Clamscan-Anwendung und legen fest, was mit infizierten Dateien passieren soll – also, ob deren Existenz lediglich in einer Logdatei vermerkt oder ob die Datei sofort gelöscht wird.

## Cloud auf dem Raspberry Pi

Ein Cloud-Server ist sinnvollerweise rund um die Uhr online und erreichbar. Dank hoher Strompreise kann dies eine kostspielige Angelegenheit und in Gebieten mit gelegentlichen Internetausfällen sogar gänzlich unmöglich sein. Sollten Sie nur moderate Anforderungen an die Verfügbarkeit haben, dann können Sie mit einem Raspberry Pi immerhin die Kosten senken: Für den Einsatz als Server müssen Sie vor der ownCloud-Installation sei-

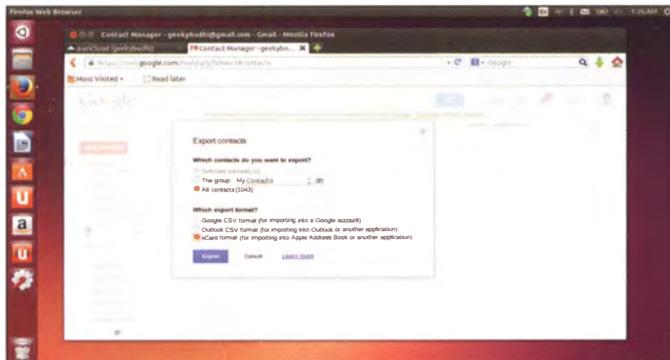
nem Prozessor lediglich den vollen Arbeitsspeicher zuweisen. Der Prozess lässt sich mit dem Skript von petRockBlog (<https://github.com/petrockblog/OwncloudPie>) automatisieren, das auf einer bestehenden Raspbian-Installation ownCloud installiert und einrichtet. Besonders Mutigen legen wir die Distribution arkOS (<https://arkos.io/>) ans Herz, die speziell für den Einsatz als Cloud-Server entwickelt wurde.



Der Raspberry Pi eignet sich bestens als Cloud-Server.

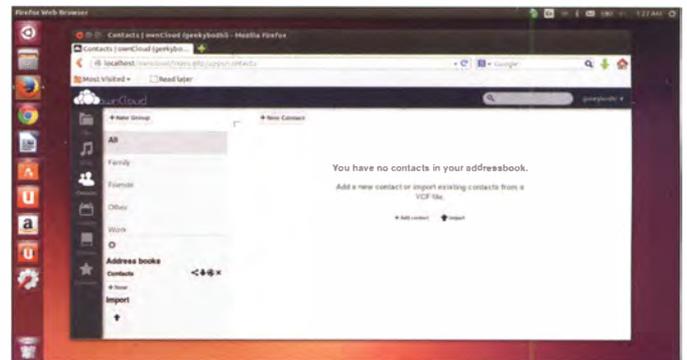
# Kalender und Adressbuch synchronisieren

Der Export in die Cloud und die Synchronisierung auf Desktop und Geräten



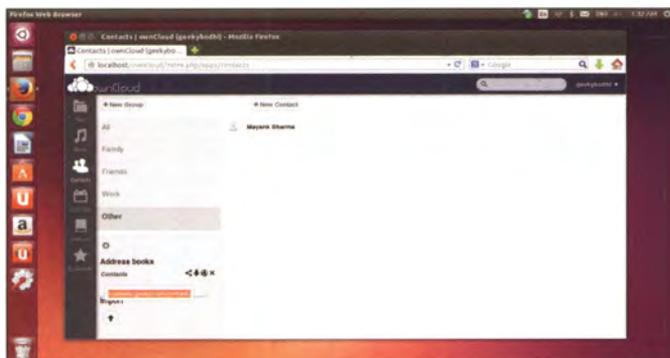
## 1 Kontakte exportieren

Mit ownCloud synchronisieren Sie Ihre Kontakte auf allen mit dem Server verbundenen Geräten. Dazu müssen Sie diese aber zuerst auf dem ownCloud-Server speichern. Dafür verwendet ownCloud das verbreitete vCard-Dateiformat, in das alle nennenswerten Mailprogramme und -dienste ihre Adressbücher exportieren können.



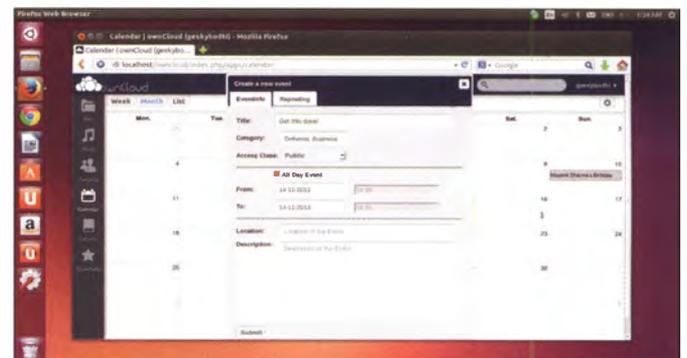
## 2 In ownCloud importieren

Sobald Sie die.vcf-Datei haben, begeben Sie sich in den „Contacts“-Bereich der ownCloud-Administration und klicken Sie auf das Zahnradsymbol unten links. Nun können Sie ein neues Adressbuch anlegen, das die Kontaktdaten aufnehmen wird. Klicken Sie auf den Pfeil und verwenden Sie den Dateimanager, um die .vcf-Datei zu laden.



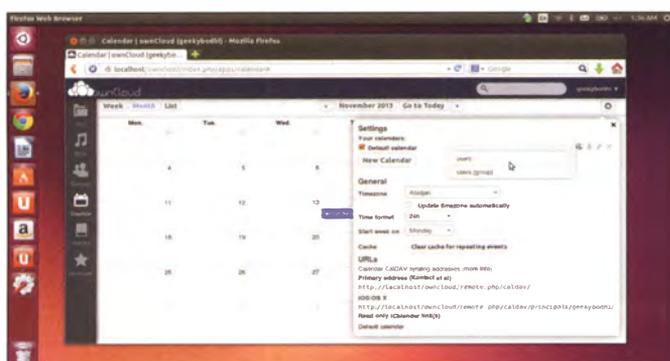
## 3 Adressbuch synchronisieren

Dieses Adressbuch lässt sich mit allen Clients synchronisieren, die den CardDAV-Standard beherrschen. Klicken Sie wieder auf das Zahnradsymbol, bewegen Sie den Mauszeiger über das Adressbuch und klicken Sie auf das Globussymbol. Den nun erzeugten CardDAV-Link geben Sie in den Adress- oder Mailprogrammen Ihrer Geräte ein.



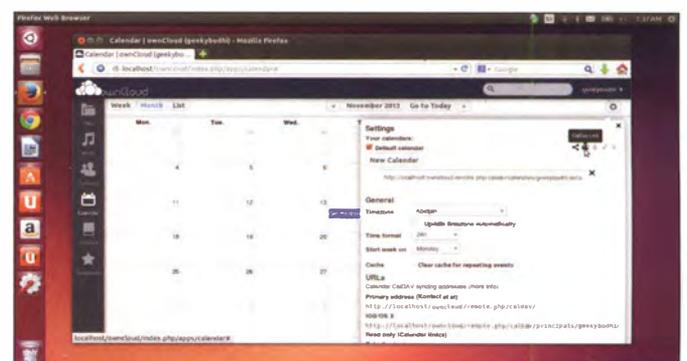
## 4 Kalender editieren

OwnCloud verwaltet auch Ihre Kalender und Aufgaben. Um ein Ereignis in Ihrem Kalender zu erstellen, begeben Sie sich in den „Calendar“-Abschnitt, wo Sie den Kalender des ganzen Monats oder einer Woche sehen. Um ein Ereignis hinzuzufügen, klicken Sie auf ein Datum und füllen die Eingabefelder des nun erscheinenden Fensters aus.



## 5 Mit anderen teilen

Nachdem Sie das Ereignis gespeichert haben, können Sie es via ownCloud mit anderen Benutzern teilen. Klicken Sie auf das Zahnradsymbol, bewegen Sie den Mauszeiger über den fraglichen Kalender und wählen Sie „Share Calendar“. Dann legen Sie die Benutzer und Gruppen fest, die Zugriff auf diesen Kalender haben sollen.



## 6 Kalender synchronisieren

Ein Kalender lässt sich wie auch das Adressbuch mit dem Desktop und mit Mobilgeräten synchronisieren. Der hierbei verwendete CalDAV-Standard nutzt ebenfalls einen Link, den Sie in den Kalender-Apps eintragen. Diesen Link erhalten Sie, indem Sie auf das Zahnradsymbol und das Globussymbol klicken.

# Torrent-Server

Machen Sie aus einem Raspberry Pi eine stets verfügbare Torrentbox.

Eine dankbare Aufgabe für jeden Raspberry Pi oder einen ausgedienten PC ist der Betrieb als Torrentbox. Mit einer Torrentbox können Sie Torrents von jedem Rechner in Ihrem Netzwerk hinzufügen, verwalten und teilen. Dazu kann das Gerät wie ein Netzwerkservers ohne Monitor und Tastatur in einer finsternen Ecke stehen und seine Aufgaben verrichten, da der Zugriff auf dessen Dienste ohnehin über das Netzwerk erfolgt. Zudem ist eine stets verfügbare Torrentbox bestens geeignet, um bei Ihrem Tracker für eine gute Ratio zu sorgen.

Für die Erstinstallation löschen Sie die Festplatte des Raspberry Pi und installieren eine frische Distribution. Wir verwenden Ubuntu Server, da diese Distribution keinen grafischen Desktop installiert und auch sonst sehr geringe Hardware-Anforderungen hat. Zudem raten wir zu einer möglichst große Festplatte, die all die zukünftigen Downloads aufnehmen kann. Für eine bestmögliche Kompatibilität mit den diversen Geräten in Ihrem Netzwerk empfehlen wir das auf allen Plattformen verbreitete NTFS-Dateisystem. Wenn Sie auf dieser Festplatte auch das

Betriebssystem unterbringen möchten, dann sollten Sie dafür eine eigene Partition einrichten.

## Deluge installieren

Für Linux gibt es mehrere Torrent-Clients. Wir verwenden in diesem Tutorial Deluge BitTorrent, da diese Software als Daemon auf dem Server laufen kann und Verbindungen von anderen Deluge-Clients akzeptiert. Da es für nahezu jede Plattform Deluge-Clients gibt, können Sie sich daher mit Linux, Windows und Mac OS X mit Ihrer Torrentbox verbinden.

Es gibt mehrere Methoden, um auf Deluge zuzugreifen. Mittels der Weboberfläche sind die grundlegenden Funktionen erreichbar. Um das volle Potenzial der Software auszuschöpfen, müssen Sie zusätzlich den Deluge-Daemon einrichten und ihn Verbindungen anderer Deluge-Installationen akzeptieren lassen.

Beginnen wir mit Deluges Weboberfläche. Sie installieren Deluge mit **sudo apt-get install deluged python-mako deluge-web**. Dieser Befehl startet lediglich den Daemon, für die Installation der Weboberfläche benötigen wir noch den Konsolenbefehl **deluge-web &**. Damit ist Deluge unter der IP-Adresse der Torrentbox auf dem Port 8112 erreichbar. Starten Sie auf einem anderen Rechner den Webbrowser und navigieren Sie zu 192.168.3.101:8112 (bzw. der für Ihre Torrentbox gültigen

„Die Torrentbox kann in einer Ecke stehen und ihre Arbeit verrichten“

## Ein externes Download-Verzeichnis einrichten

So speichern Sie Downloads auf einer externen Festplatte.

```

bodhi@bodhi-Aspire-5738:~$ sudo fdisk -l /dev/sdb
Disk /dev/sdb: 7759 MB, 7759462400 bytes
94 heads, 46 sectors/track, 3504 cylinders, total 15155200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0011b0c8

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1             2048         1515199     7576576    7  HPFS/NTFS/exFAT
bodhi@bodhi-Aspire-5738:~$ sudo mkdir /media/usb-disk
bodhi@bodhi-Aspire-5738:~$ mount -t auto /dev/sdb1 /media/usb-disk/
mount: only root can do that
bodhi@bodhi-Aspire-5738:~$ sudo mount -t auto /dev/sdb1 /media/usb-disk/
bodhi@bodhi-Aspire-5738:~$
    
```

```

bodhi@bodhi-Aspire-5738:~$ preexec = /bin/mount /cdrom
; postexec = /bin/umount /cdrom

[Downloads]
comment = Torrent download folder
path = /media/usb-disk
valid users = @users
force group = users
create mask = 0660
directory mask = 0771
read only = no
    
```

### 1 Externe Festplatte mounten

Formatieren Sie die Festplatte mit NTFS, stellen Sie mit **sudo apt-get install ntfs-3g** sicher, dass Ihre Distribution NTFS unterstützt, und richten mit **sudo mkdir /media/usb-disk** einen Mountpoint ein. Verbinden Sie die Platte mit dem PC und stellen Sie mit **sudo fdisk -l** deren Adresse fest. Sollte es **/dev/sdb** sein und die Platte nur die Partition **sdb1** besitzen, dann verbinden Sie Platte und Mountpoint mit **sudo mount -t auto /dev/sdb1 /media/usb-disk**.

### 2 Fernzugriff einrichten

Installieren Sie mit **sudo apt-get install samba sambacombin-bin Samba**, um von anderen Rechnern die Platte zu nutzen. Dann editieren Sie **/etc/samba/smb.conf** und entfernen das Kommentarzeichen vor **security=user**. Am Ende der Datei fügen Sie den im obigen Bild gezeigten Abschnitt an, dann erstellen Sie mit **sudo useradd downloads -m -G users** den Benutzer **downloads**. Mit **sudo smbpasswd -a downloads** darf dieser Benutzer auf den Server zugreifen.

# einrichten

## Die Torrentbox mit Raspberry Pi betreiben

Anstelle eines alten Computers kann auch ein Raspberry Pi als Grundlage der Torrentbox dienen. Die Befehle für die Installation des Deluge-Daemons und der Web-Oberfläche sind dabei identisch. Natürlich hat ein Raspberry Pi keine eingebaute Festplatte, daher verwenden wir – wie oben beschrieben – eine externe Platte zusammen mit einem Samba-Server. Damit der

Deluge-Daemon und die Web-Oberfläche zusammen mit dem Raspberry Pi starten, benötigen wir nur einen einzigen zusätzlichen Schritt.

Kopieren Sie die zwei von den Deluge-Entwicklern bereitgestellten Skripte von <http://dev.deluge-torrent.org/wiki/UserGuide/InitScript/Ubuntu> auf den Raspberry Pi und machen Sie die Dateien mit

**sudo chmod 755 /etc/default/deluge-daemon** und **sudo chmod 755 /etc/init.d/deluge-daemon** ausführbar.

Abschließend fügen Sie die Skripte mit **sudo update-rc.d deluge-daemon defaults** der Startprozedur hinzu. Beim nächsten Start des Raspberry Pi werden beide Programme automatisch gestartet.

Adresse). Deluges Frage nach einem Kennwort beantworten Sie mit **deluge**, woraufhin Sie das Kennwort auch sofort ändern müssen.

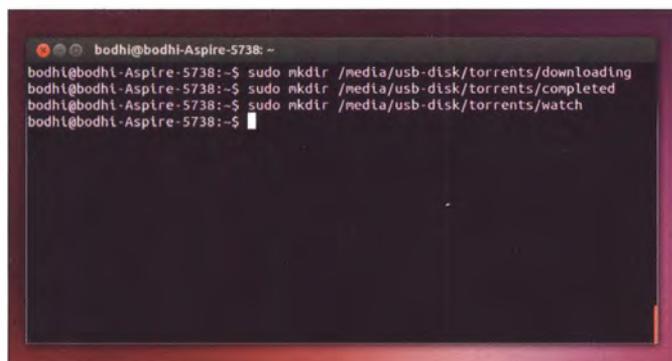
Nun soll sich der Web-Client mit einem einzelnen Deluge-Daemon verbinden. Dazu wird automatisch der Connection Manager gestartet, der alle im Netzwerk verfügbaren Daemons auflistet. Wählen Sie den passenden Daemon aus und klicken Sie auf „Connect“. Nun können Sie den Web-Client verwenden, um der Torrentbox Torrents hinzuzufügen und diese zu verwalten.

Wenn Sie die Portnummer verändern möchten, dann geben Sie sich mit **cd ~/.config/deluge/** in Deluges Konfigurationsverzeichnis und öffnen die Datei **web.conf** in einem Texteditor Ihrer Wahl. Nun finden Sie die Zeile **„port“: 8112** und ersetzen **8112** mit einer beliebigen Portnummer über **1000**.

### Den Daemon einrichten

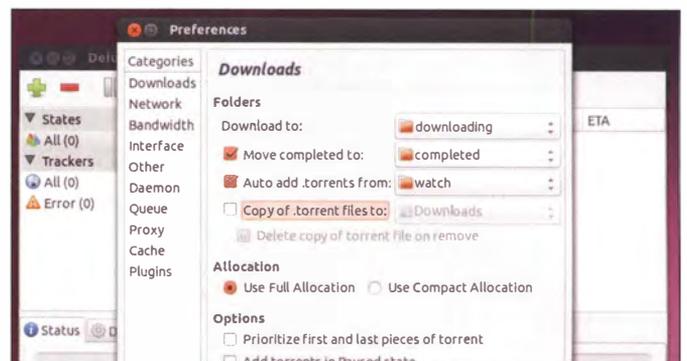
Wie bereits erwähnt, wollen wir den Daemon noch so konfigurieren, dass er Anfragen von externen Clients beantwortet. Hierzu installieren wir den Deluge-Konsolen-Client mit dem Befehl **sudo apt-get install deluge-console**, mit dem solche Konfigurationsänderungen möglich sind.

Anschließend stoppen wir den Daemon mit **sudo pkill deluged** und öffnen die Konfigurationsdatei **~/.config/deluge/auth** mit einem Texteditor. Am unteren Ende der Datei tragen wir dann in einer einzigen Zeile drei mit Doppelpunkt getrennte Parameter ein, die den Benutzernamen, das Kennwort und den Authentifizierungslevel festlegen, beispielsweise **my-username:mypassword:10**. **my-username** und **my-password** ersetzen Sie durch Ihre persönlichen Daten. Die **10** gibt an, dass der Daemon Administratorprivilegien



### 3 Verzeichnisse erstellen

Auf der Festplatte richten wir für Deluge nun drei Verzeichnisse ein. Die Befehle **sudo mkdir /media/usb-disk/torrents/downloading** und **sudo mkdir /media/usb-disk/torrents/completed** erstellen Verzeichnisse, in denen laufende und abgeschlossene Downloads landen. Erstellen Sie ein weiteres Verzeichnis mit **sudo mkdir /media/usb-disk/torrents/watch**. Deluge wird dieses Verzeichnis überwachen und **.torrent**-Dateien automatisch erkennen.



### 4 Mit Deluge verknüpfen

Um dies zu erreichen, verbinden wir die soeben erstellten Verzeichnisse mit den Deluge-Desktop-Clients. Starten Sie dazu Deluge und öffnen Sie „Edit > Preferences“. Wählen Sie die „Downloads“-Kategorie in der linken Fensterhälfte. Markieren Sie nun unter „Folders“ die beiden Checkboxes und wählen Sie im Dateimanager die drei Verzeichnisse aus, die Sie im vorherigen Schritt auf der externen Festplatte der Torrentbox erstellt haben.

# Der Torrent-Server von TurnKey Linux

Das Projekt TurnKey Linux entwickelt auf einzelne Aufgaben spezialisierte, „schlüsselartige“ Server-Software. Da sie auf der Debian-Variante von Just enough Operating System (JeOS) aufsetzt, sind diese Distributionen extrem schlank. Unter den knapp 100 TurnKey-Linux-Varianten befindet sich auch eine spezielle Version für den Betrieb als Torrent-Server. Insbesondere enthält sie mit MLDonkey eine Anwendung für das Filesharing mit nahezu allen bekannten Protokollen.

Nach der Installation lässt sich die Distribution komplett per Webbrowser verwalten. Sie können Dateien hoch- und herunterladen, mit zip- und rar-Algorithmen komprimieren und entpacken, zusätzliche Pakete installieren und vieles mehr – all das mit jedem beliebigen Webbrowser von jedem mit dem Internet verbundenen Computer.

Sie erhalten die Distribution auf [www.turnkeylinux.org/torrentserver](http://www.turnkeylinux.org/torrentserver) und können sie direkt installieren. Einzelne Komponenten lassen sich vorab im Live-Modus testen. In beiden Fällen werden Sie nach einem Kennwort für den root-Benutzer sowie für den Administrator der diversen Komponenten gefragt: die MLDonkey-

Filesharing-App, die P2P-GUI von MLDonkey sowie den Dateimanager eXtplorer.

Sobald die Benutzer konfiguriert sind, kopiert das Installationsprogramm die benötigten

Dateien und installiert den Bootloader. Nach einem Neustart lädt die Konfigurationskonsole und listet die Adressen der diversen Server-Apps auf.

**TORRENTSERVER appliance services**

**Web:** <https://192.168.2.3>  
**Web shell:** <https://192.168.2.3:12320>  
**Webmin:** <https://192.168.2.3:12321>  
**MLDonkey:** <https://192.168.2.3:12322>  
**Web-GMUI:** <https://192.168.2.3:12323>  
**File manager:** <https://192.168.2.3/extplorer>  
**SMB/CIFS:** \\192.168.2.3 (ports 139/445)  
**SSH/SFTP:** root@192.168.2.3 (port 22)

**TKLBAM (Backup and Migration): NOT INITIALIZED**

**TurnKey Backups and Cloud Deployment**  
<https://hub.turnkeylinux.org>

› Im Gegensatz zu unserer Torrentbox, die auf Deluge basiert, verwendet TurnKey Torrent Server die vielseitige Filesharing-App MLDonkey.

erhält.

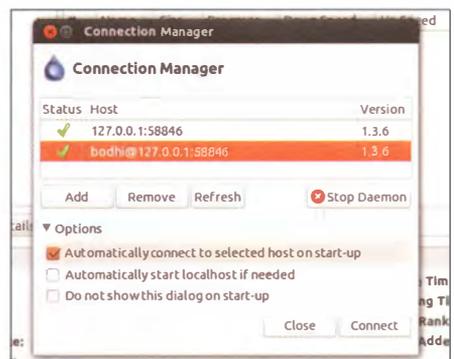
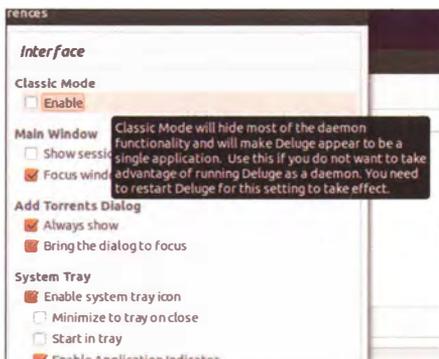
Speichern Sie nun die Datei und starten Sie den Daemon mit dem Befehl **deluged**. Öffnen Sie dann die soeben installierte Deluge-Konsole mit dem Befehl **deluge-console**, woraufhin sich die Kommandozeile des Deluge-Daemons zeigt. Damit er nun endlich externe Verbindungen akzeptiert, geben Sie noch **config -s allow\_remote True** ein und warten auf die Bestätigung

des Clients. Nun können Sie die Konsole mit **exit** verlassen. Damit die Veränderungen tatsächlich ihre Wirkung entfalten, stoppen Sie den Daemon mit **sudo pkill deluged** und starten Sie ihn erneut mit **deluged**.

Die Installation von Deluge auf Ihrer Torrentbox ist damit abgeschlossen und der Zugriff auf das Gerät und seine Dienste ist mit externen Clients möglich. Wie Sie diese Clients einrichten, erfahren Sie in den nebenstehenden Texten.

## Verbindung mit dem Server

Machen Sie die Torrentbox den Deluge-Clients Ihres Netzwerks bekannt.



### 1 Clients installieren

Für den bequemen Einsatz von Deluge installieren wir den Client auf allen Rechnern im Netzwerk. Die meisten Linux-Distributionen kennen Deluge, daher können Sie den Client im Paketmanager installieren. Den Client für Windows und Mac OS X finden Sie auf <http://deluge-torrent.org>.

### 2 Classic-Modus einrichten

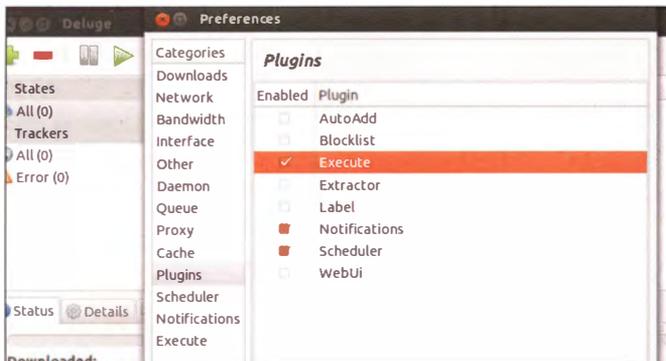
Die Clients sind zunächst so eingerichtet, dass sie auf den lokalen Rechner heruntergeladen. Dies ändern wir, indem wir im Client zu „Edit > Preferences“ und dann zu „Interface“ navigieren. Das Häkchen bei „Classic Mode“ wird entfernt, was nach einem Klick auf „OK“ einen Neustart erfordert.

### 3 Host hinzufügen

Deluge zeigt nun den Connection Manager an. Klicken Sie auf „Add“ und geben Sie die Verbindungsdaten des Daemons ein. Nach einem Klick auf das nächste „Add“ und anschließend auf „Connect“ ist der Client mit dem Daemon der Torrentbox verbunden.

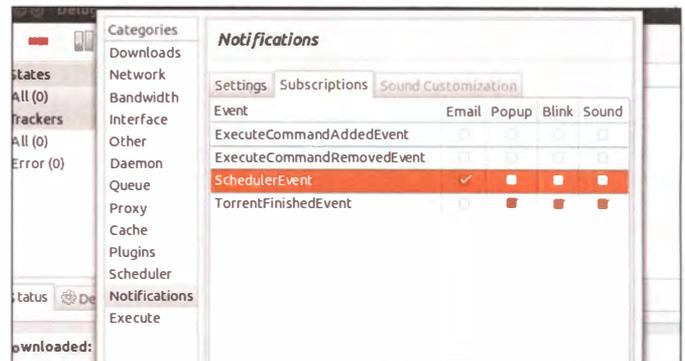
# Torrents besser nutzen

Vereinfachen Sie die Verwaltung der Torrentbox.



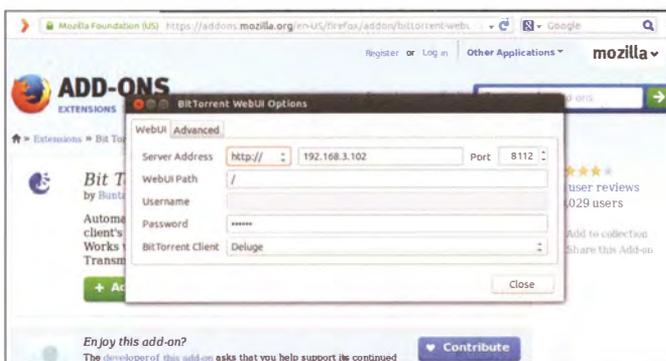
## 1 Deluge-Plug-ins aktivieren

Deluge enthält acht Plug-ins. Um diese zu aktivieren, navigieren Sie zu „Edit > Preferences“ und wählen Sie „Plugins“. Setzen Sie bei jedem gewünschten Plug-in ein Häkchen, woraufhin das Plug-in als Kategorie im „Preferences“-Fenster auftaucht und einzeln konfiguriert werden kann. Weitere Plug-ins finden Sie mit „Find More Plugins“.



## 2 Nützliche Plug-ins

Das Scheduler-Plug-in begrenzt auf Wunsch die genutzte Netzwerkbandbreite basierend auf der Uhrzeit. Das Notification-Plug-in meldet, sobald ein Torrent heruntergeladen wurde. Fortgeschrittene werden das Execute-Plug-in mögen, das beliebige Befehle ausführt, sobald ein Torrent hinzugefügt oder heruntergeladen wird.



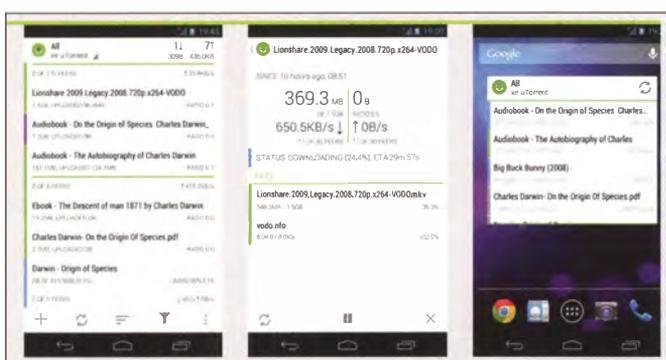
## 3 Add-on für Firefox

Für Firefox gibt es die Erweiterung Bit Torrent WebUI+, mit der Sie über die Weboberfläche dem Deluge-Daemon Torrents hinzufügen können. Nach der Installation geben Sie in einer Dialogbox die Verbindungsdaten der Deluge-Weboberfläche ein. Ein Klick auf einen Torrent-Link fügt ihn nun automatisch der Torrentbox hinzu.



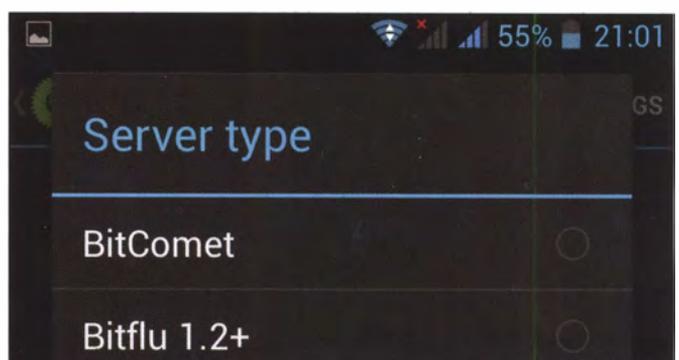
## 4 Add-on für Google Chrome

Im Chrome Web Store wartet die App DelugeSiphon. Klicken Sie nach der Installation auf das Symbol in der Adressleiste und öffnen Sie „Options“. Geben Sie die Adresse des Daemons sowie das Kennwort der Weboberfläche ein. Ab sofort zeigt ein Rechtsklick auf einen Torrent-Link die Option „Send to Deluge“.



## 5 Clients für Android

Google Play listet mehrere Apps auf, die sich mit einem Deluge-Daemon verbinden. Auch außerhalb von Google Play sind gute Apps erhältlich. Zwei der populärsten Apps sind Transdrone und das umfangreichere TransDroid. Ersteres erhalten Sie in Google Play, Letzteres unter <http://transdroid.org/latest>.



## 6 Den Client einrichten

Die Konfiguration von Transdrone und TransDroid ist nahezu identisch. Beim ersten Start der App tippen Sie auf „Settings“ und dann auf „Add a new server“. Klicken Sie auf „Server type“ und wählen Sie „Deluge 1.2+“ aus der Liste. Schließlich tragen Sie noch die IP-Adresse der Torrentbox ein.

# Anonym bleiben

„Die da“ überwachen Ihre Online-Aktivitäten? Dann brauchen Sie Tor.

**T**or steht für „The Onion Router“ und ermöglicht die anonyme Nutzung des Internets. Tor verschleiert Ihre IP-Adresse und macht es Angreifern nahezu unmöglich, Ihren Standort zu ermitteln und Ihre Surfgewohnheiten zu verfolgen.

Unter normalen Umständen errichtet Ihr Computer eine direkte Verbindung zu einer jeden besuchten Webseite. Diese Webseite kennt daraufhin Ihre IP-Adresse und kann damit feststellen, wo Sie sich ungefähr befinden. Außerdem erfahren Ihr Internet-Provider (oder Ihr Chef sowie die NSA), welche Webseiten Sie besuchen.

Tor verschleiert diese Verbindungsdaten, indem es ein Netzwerk von Verteilerknoten über das gesamte Internet verstreut. Wenn Sie über Tor eine Webseite besuchen, dann werden die Daten auf dem Weg von und zu Ihrem Computer über mehrere dieser Knoten geleitet, bevor sie ihr Ziel erreichen. Auf diese Weise erfährt die Webseite nicht, wer sie tatsächlich besucht hat, außerdem lässt sich nicht mehr feststellen, welches Ziel die von Ihrem Computer stammenden Netzwerkdaten haben.

Der Preis der Anonymität ist die verringerte Geschwindigkeit, mit der sich Webseiten in Ihrem Browser aufbauen – schließlich nehmen die einzelnen Datenpakete nun lange Umwege über die Verteilerknoten.

Das Tor-Projekt bietet ein spezielles Softwarepaket namens „Tor Browser Bundle“ an, das alles für die Nutzung

des Tor-Netzwerks Nötige beinhaltet, unter anderem eine als „Tor Browser“ bekannte Firefox-Variante. Das Bundle können Sie auf Festplatte oder auf USB-Stick installieren.

Das Bundle ist für den Einsatz auf einem einzelnen Rechner spezialisiert. Wenn Sie mehrere Computer besitzen, dann können Sie Tor auf einem Computer installieren und diesen als Proxy-Server für alle anderen Rechner verwenden. Zwar kann jeder beliebige Computer diese Aufgabe übernehmen, wir empfehlen dafür jedoch einen separaten, jederzeit eingeschalteten Rechner.

## Tor einrichten

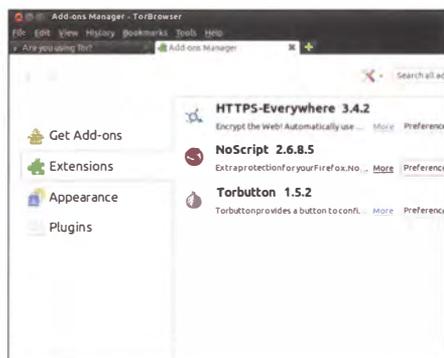
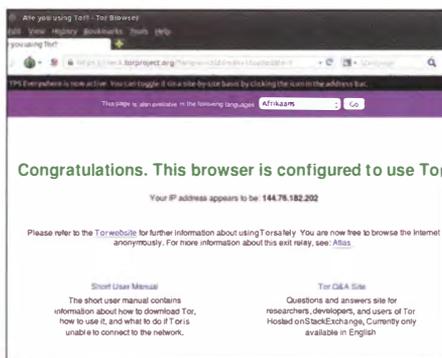
Zuerst benötigt der auserkorene Rechner eine statische IP-Adresse. Diese richten Sie auf der Administrationsseite Ihres Routers ein. In diesem Tutorial gehen wir von der IP-Adresse 192.168.1.100 aus.

Der Befehl **sudo apt-get install tor vidalia** lädt die benötigte Software Vidalia – die grafische Benutzeroberfläche von Tor – aus den Repositories. Wenn der Paketmanager die Pakete nicht findet, dann müssen Sie vermutlich das Repository „Universe“ aktivieren.

Tor wird unter anderem mittels der Konfigurationsdatei **torrc** im Verzeichnis **/etc/tor** konfiguriert. Normalerweise muss man diese Datei nicht verändern, da die Tor-Installation aber Verbindungen aus dem lokalen Netzwerk entgegennehmen soll, öffnen wir sie in einem Texteditor. Stellen Sie sicher,

## Das Tor Browser Bundle

Die einfachste Methode, Tor zu verwenden



### 1 Bundle herunterladen

Nachdem Sie das Tor Browser Bundle von **www.torproject.org** heruntergeladen haben, entpacken Sie das selbstextrahierende Archiv mit einem Doppelklick. Sie müssen nur noch ein Zielverzeichnis angeben, das auf der lokalen Festplatte oder auf einem USB-Stick liegen darf. Die Software benötigt rund 90 MB Festplattenspeicher und keine weitere Installation.

### 2 Mit Tor verbinden

Nun öffnen Sie ein Terminalfenster, wechseln in das soeben erstellte Verzeichnis und geben den Befehl **./start-tor-browser** ein. Dies lädt das Programm Vidalia Control Panel, welches Sie automatisch mit dem Tor-Netzwerk verbindet. Sobald die Verbindung steht, öffnet Vidalia selbstständig den Tor-Browser, der Sie mit Ihrer maskierten IP-Adresse begrüßt.

### 3 Sicherheits-Add-ons konfigurieren

Der Tor-Browser unterscheidet sich von anderen Browsern durch etliche integrierte Komponenten, mit denen er Ihre Privatsphäre schützt. Mit Torbutton beispielsweise konfigurieren Sie Tor und mit HTTPS Everywhere erzwingen Sie für populäre Webseiten die Verwendung des sicheren, verschlüsselten HTTPS-Protokolls.

# mit Tor

dass **SocksListenAddress** auf die Adresse des Rechners (**192.168.1.100**) und nicht auf den localhost (127.0.0.1) verweist.

Im gleichen Verzeichnis liegt auch die Datei **tor-tsocks.conf**. Dies ist die Konfigurationsdatei von **tsocks**, die SOCKS-Proxy auch solchen Programmen zur Verfügung stellt, die sie normalerweise nicht unterstützen. Öffnen Sie auch diese Datei und fügen Sie folgende Zeilen an, die auf das lokale Netzwerk verweisen:

```
server = 192.168.1.100
server_port = 9050
```

Damit ist Ihr anonymes Proxy-Server bereit für den Einsatz.

## Tor benutzen

Um Ihren Netzwerkverkehr zu anonymisieren, müssen Sie ihn über den soeben eingerichteten Proxy leiten. Dies geschieht, indem Sie entweder die Betriebssysteme aller Ihrer Geräte oder aber jede einzelne App entsprechend konfigurieren.

Unter Ubuntu 13.10 klicken Sie auf das Zahnradsymbol in der oberen rechten Bildschirmcke, begeben sich in die „System Settings“ und anschließend in die „Network Settings“ im Hardware-Abschnitt. Dort finden Sie eine Einstellungsgruppe namens „Network proxy“. Wählen Sie „manual“ aus der Liste der verfügbaren Methoden und tragen

Sie die Verbindungsdaten Ihres Proxys neben dem SOCKS-Host-Feld ein. Anschließend bestätigen Sie die Eingaben mit „Apply system wide“.

Wenn Sie stattdessen einzelne Apps konfigurieren möchten, öffnen Sie deren jeweilige Konfigurationsfenster und tragen Sie in den Proxy-Abschnitten die IP-Adresse und die Portnummer des Tor-Daemons ein.

In Firefox öffnen Sie beispielsweise „Extras > Einstellungen“ und springen in den „Erweitert“-Abschnitt. Hier wechseln Sie auf den „Netzwerk“-Reiter und klicken auf die „Einstellungen“-Schaltfläche, woraufhin Sie die Proxy-Einstellungen endlich vornehmen können. Legen Sie fest, dass Sie die Konfiguration manuell vornehmen, tragen Sie im SOCKS-Host-Feld die Verbindungsdaten ein und stellen Sie sicher, dass „SOCKS v5“ ausgewählt ist.

Firefox benötigt eine weitere Einstellung, da die App normalerweise DNS-Anfragen nicht über den SOCKS5-Proxy leitet. Um dies zu ändern, geben Sie in der Adressleiste **about:config** ein und suchen Sie die Variable **network.proxy.socks\_remote\_dns**. Mit einem Rechtsklick und Auswahl von „Umschalten“ ändern Sie den Variablenwert von „false“ auf „true“.

Auf ähnliche Weise ändern Sie die Konfiguration aller anderen netzwerkfähigen Apps, die das SOCKS5-Protokoll unterstützen.

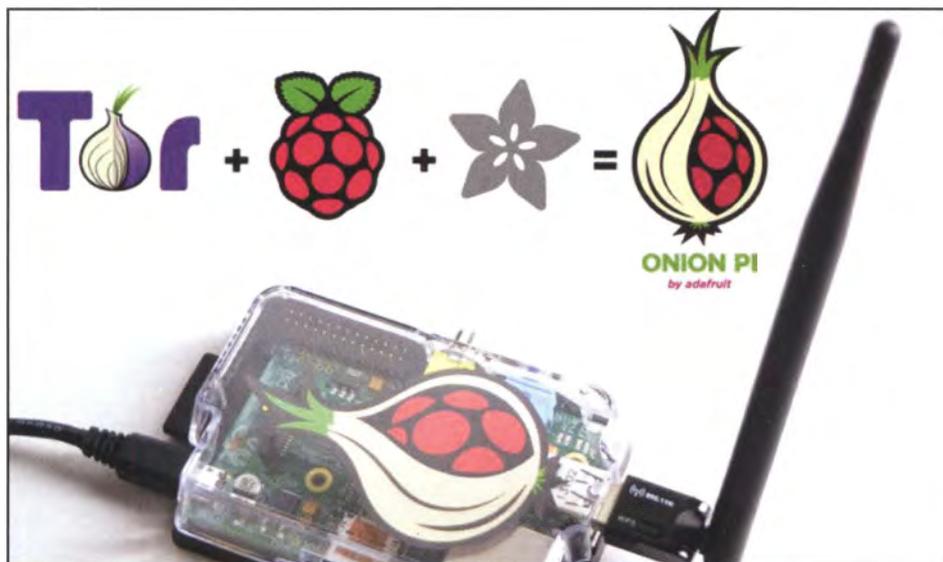
## So wird aus dem Raspberry Pi ein Tor-Proxy

Da ein Proxy-Server ununterbrochen eingeschaltet sein sollte, verwendet man für seine Hardware sinnvollerweise einen energiesparenden Raspberry Pi. Wenn Sie dafür die Debian-basierte Raspbian-Distribution verwenden, dann können Sie

das Tutorial ohne Änderung anwenden.

Wenn Ihr Raspberry Pi über einen WLAN-Adapter verfügt, dann lässt er sich zudem in einen Tor-WLAN-Hotspot verwandeln. Er kann mit allen WLAN-fähigen Geräten wie ein klassischer

Zugangspunkt verwendet werden und anonymisiert automatisch deren Netzwerkverkehr. Unter <http://learn.adafruit.com/=all> hat Adafruit die für einen solchen Tor-Hotspot notwendigen Schritte beschrieben.



› Mit dem WLAN-Proxy Onion Pi ersparen Sie sich den Aufwand, individuelle Geräte oder Apps für den anonymen Internetbesuch zu konfigurieren.

# Tor steuern

Mit Vidalia Control Panel übernehmen Sie die Kontrolle.

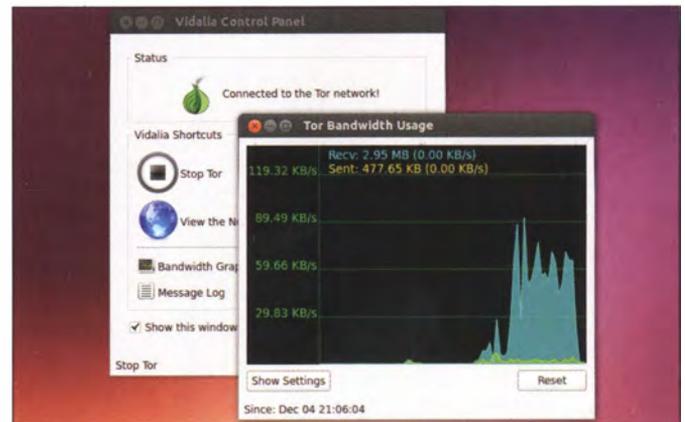
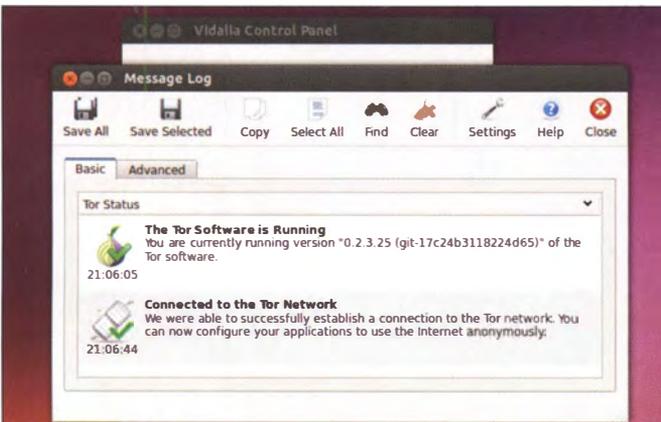


## 1 Das Netzwerk betrachten

Die Schaltfläche „View the Network“ zeigt eine Landkarte, auf der alle aktiven Verteilerknoten sowie der Weg Ihrer Daten verzeichnet sind. Die Liste auf der linken Seite zeigt die aktiven Verteilerknoten, die mittlere Liste die von Ihren Daten verwendeten Knoten und rechts daneben stehen die Eigenschaften eines einzelnen Knotens.

## 2 Eine neue Identität anfordern

Einige Webseiten blockieren den Zugang, wenn der Besucher vermeintlich aus einem ungeeigneten Land kommt. Um solche Zugangssperren zu umgehen, können Sie jederzeit eine neue IP-Adresse anfordern, indem Sie auf „Use a New Identity“ klicken. Auf diese Weise scheinen Sie für eine Webseite aus einem anderen Land zu kommen.



## 3 Die Logdatei untersuchen

Sie sollten gelegentlich einen Blick auf Tors Logdaten werfen. Klicken Sie auf „Message Log“, um bei Verbindungsproblemen deren Ursache zu finden. Wechseln Sie auf den „Advanced“-Reiter, um eine genauere Analyse zu erhalten. Sogar Tors offizielle Dokumentation empfiehlt im Zweifel das übliche Allheilmittel: Starten Sie den Rechner neu.

## 4 Die Session beenden

Neben der etwas geringeren Surfgeschwindigkeit und der durchaus sinnvollen Unfähigkeit, Flash zu verarbeiten, benimmt sich der Tor-Browser wie jeder andere Browser. Sobald Sie fertig sind und das letzte Browserfenster geschlossen haben, löscht Tor-Browser die Liste der besuchten Webseiten und meldet Sie vom Tor-Netzwerk ab.

# Tails – die Distribution für sicherheitsbewusste Nutzer

Wenn Sie Tor mögen, dann werden Sie Tails lieben. Das Ziel dieser Distribution ist, Ihnen den anonymen Zugang zum Internet zu ermöglichen, alle Zensurversuche zu umgehen und dabei keinerlei Spuren der Internetaktivitäten auf dem Rechner zu hinterlassen.

Tails steht für „The Amnesic Incognito Live System“. Es ist eine auf Debian basierende Distribution, startet von DVD oder USB-Stick und schreibt im Betrieb nichts auf

die Auslagerungs-Disk. Die Auswahl der mitgelieferten und vorkonfigurierten Apps wurde für die sichere Internetnutzung optimiert: Webbrowser, Instant-Messaging-Client, E-Mail, Office und etliches mehr. Weitere Apps lassen sich mit Synaptic von den Debian-Repositories installieren.

Sämtlicher Internetverkehr erfolgt über Tor, darüber hinaus enthält Tails einen Passwortmanager und Werkzeuge

für die Ver- und Entschlüsselung von Dateien oder einzelnen Textausschnitten. Wenn Sie fürchten, dass Keylogger Ihre Tastatur überwachen, dann können Sie stattdessen eine virtuelle Tastatur verwenden.

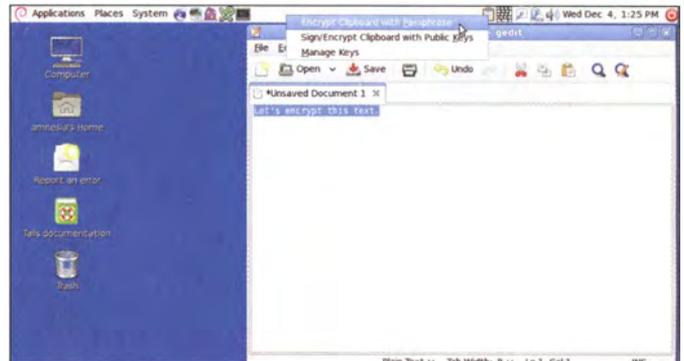
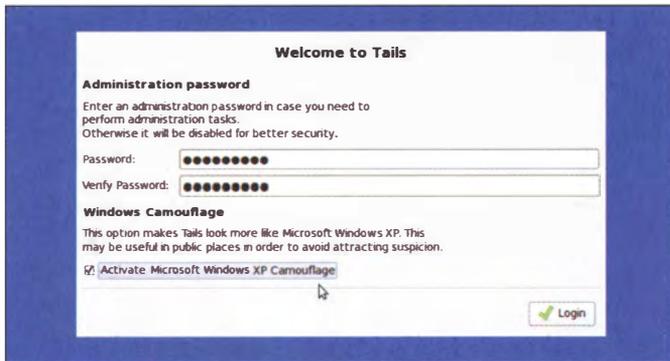
Tails kann sich sogar tarnen: Wenn Sie Grund zur Annahme haben, dass der Gnome-2-Desktop auffällt, dann kann er sich auf Knopfdruck ein Windows-XP-ähnliches Design verpassen.



Das ist Tails im Windows-XP-Gewand. So soll verhindert werden, dass Sie sich als Linux-Benutzer zu erkennen geben.

# Tails: eine Live-Distribution mit Amnesie

So schützen Sie Ihre Privatsphäre mit Tails.

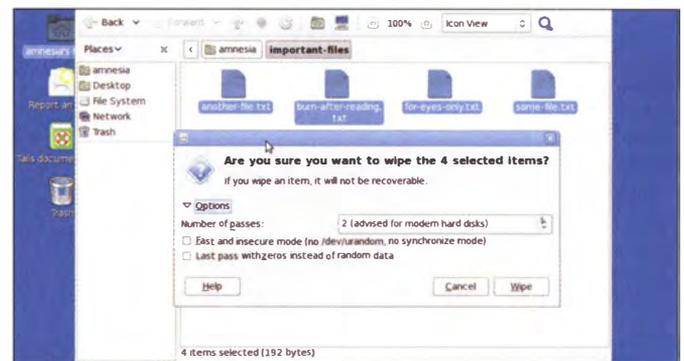
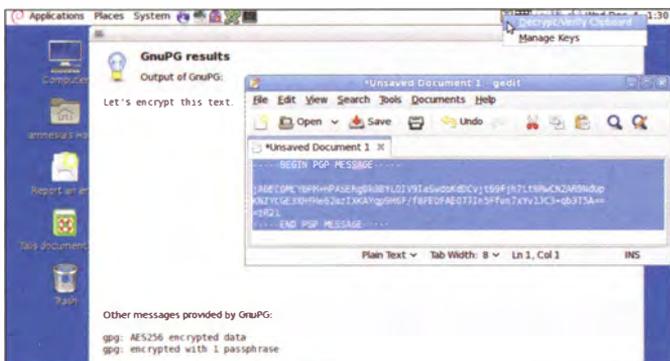


## 1 Tails einrichten

Laden Sie die Tails-Live-Distribution von <https://tails.boum.org> herunter und kopieren Sie sie auf DVD oder USB-Stick. Nach dem Booten haben Sie die Möglichkeit, einige Einstellungen vorzunehmen. In diesem Fall fragt Sie die Distribution nach dem Passwort für den Systemadministrator und lässt Sie den Tarnmodus aktivieren.

## 2 Mit Passphrase verschlüsseln

Dank des OpenPGP-Applets können Sie Ihre Kommunikation problemlos verschlüsseln. Kopieren Sie den zu verschlüsselnden Text mit [Strg]+[C] in die Zwischenablage. Dann klicken Sie auf das Applet und wählen „Encrypt Clipboard with Passphrase“. Geben Sie die Passphrase ein und fügen Sie den Text mit [Strg]+[V] in die Ziel-App ein.

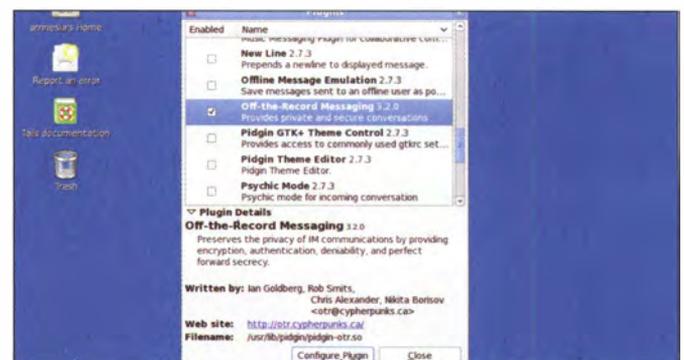
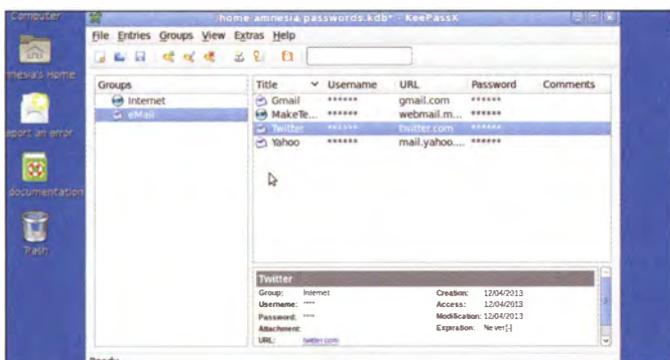


## 3 Text entschlüsseln

Den verschlüsselten Text können Sie speichern oder über ein unverschlüsseltes Netzwerk an Dritte versenden. Um ihn wieder zu entschlüsseln, kopieren Sie den Text mit [Strg]+[C] in die Zwischenablage, klicken Sie auf das Applet und wählen Sie „Decrypt/Verify Clipboard“. Nachdem Sie die Passphrase eingegeben haben, sehen Sie den entschlüsselten Text.

## 4 Dateien sicher löschen

Gelöschte Dateien landen nicht im Software-Nirwana, stattdessen wird einfach ihr Eintrag aus der Dateiliste entfernt und kann daher mit Tools wie PhotoRec wiederhergestellt werden. Die Erweiterung Nautilus Wipe des Nautilus-Dateimanagers überschreibt Dateien mit Rechtsklick > Wipe vor diesem „Löschen“ mit sinnlosen Zeichen.



## 5 Kennwörter verwalten

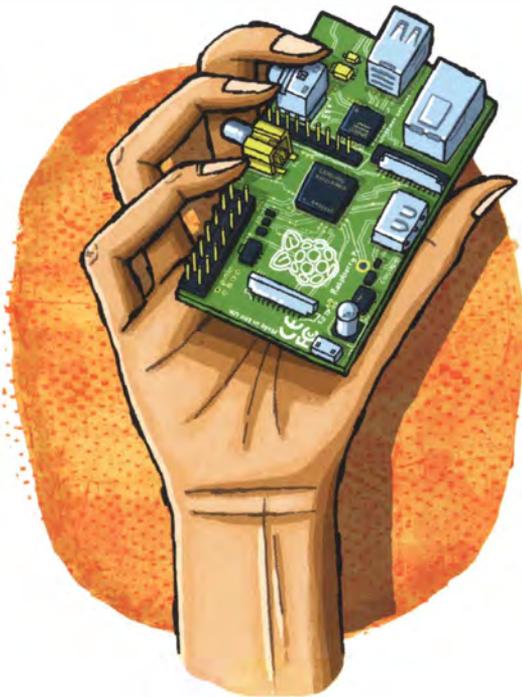
Tails enthält den Passwortmanager KeePassX. Er speichert Ihre Kennwörter in einer verschlüsselten Datenbank, die auf einem stets verfügbaren Laufwerk liegt. In „Application > Tails > Configure Persistent Volume“ erstellen Sie ein solches. Nun starten Sie KeePassX, legen eine neue Datenbank an und speichern sie auf dem Laufwerk.

## 6 Online anonym bleiben

Um Ihre Surfgeohnheiten und sonstigen Internetaktivitäten zu anonymisieren, enthält Tails Torbutton und weitere Erweiterungen für den IceWeasel Webbrowser sowie Vidalia für die Kontrolle des Tor-Netzwerks. Das Plug-in Off-the-Record für den Messenger Pidgin verschlüsselt zudem automatisch all Ihre Chats.

# Ein Netzwerk

Hier erfahren Sie, wie Sie mehrere Raspberry Pis so miteinander verbinden können, dass diese ein aufeinander abgestimmtes Rechnernetzwerk bilden.



**H**eutzutage gibt es professionelle, hochspezialisierte Hardware für anspruchsvolle Rechenoperationen wie zum Beispiel die Bildsynthese oder Simulationen. Doch das war nicht immer so. Früher verband man für derartige Aufgabenstellungen eine Anzahl Computer (die für sich genommen nicht unbedingt besonders leistungsstark zu sein brauchten) miteinander und teilte die Arbeit zwischen den einzelnen Elementen des Netzwerks auf. Solche Cluster waren zwar weniger effizient als die modernen Lösungen, was das Verhältnis von Rechenleistung zu Energieverbrauch betrifft, aber der Vorteil, den sie auch heute noch bieten, ist ihre relativ einfache Realisierbarkeit. Diese wollen wir uns zunutze machen und aus einigen Raspberry Pis einen kleinen Supercomputer basteln, ein sogenanntes Bramble (wörtlich „Beerengestrüpp“).

## Material und Zielsetzung

Auch wenn ein Raspberry Pi über eine deutlich geringere Rechenleistung verfügt als ein gewöhnlicher Desktop-PC, lässt sich mit einem Bramble-Cluster dennoch einiges bewerkstelligen, und rein preislich gesehen ist ein Pi-Netzwerk natürlich die weitaus günstigere Alternative.

In diesem Tutorial wollen wir fürs Erste zwei Raspberry Pis miteinander verbinden. Obendrein lernen Sie im Zuge dessen auch etwas über dezentrales Programmieren in der Sprache Python.

Außer den beiden Pis verwenden wir noch einen PIHUB

von Pimoroni (für eine separate Stromversorgung der Rechner) sowie einen 100-Mbit-Hub mit fünf Ports (wodurch sich das Bramble später auf bis zu fünf Geräte erweitern lässt), den man für ein paar Euro bei Ebay bekommt.

Was wollen wir also anstellen? Unser kleines Bramble soll versuchen, per Brute-Force-Angriff zwei häufig verwendete kryptographische Hashfunktionen – MD5 und SHA-1 – zu knacken. Genauer gesagt, wir möchten das Urbild zu einem Hashwert finden, der aus einem simplen Passwort generiert wurde. Während beide genannten Hashfunktionen verwundbar gegenüber Kollisionsangriffen sind (deren Ziel es ist, zwei Nachrichten mit dem gleichen Hashwert zu ermitteln), sind Urbild-Angriffe ungleich schwieriger. Eine solche Angriffsroutine gibt es für MD5, aber sie würde eine sehr lange Zeitspanne benötigen.

## Hashwerte generieren

Häufig ist von kompromittierten Datenbanken zu lesen, bei denen vertrauliche Informationen entwendet wurden, darunter Passwort-Hashwerte. Auch wenn ein sogenannter Salt (eine zufällig generierte Erweiterung des Eingabetextes vor der Erstellung des Hashwerts) einen gewissen zusätzlichen Schutz bietet, sind Passwörter immer noch anfällig gegenüber wortlistenbasierten Angriffen. In diesem Tutorial beschränken wir uns beim Python-Code jedoch auf elementare Zeichenkombinationen.

Die kryptographischen Hashfunktionen sind Teil der Standardinstallation von Python und können über die Bibliothek `hashlib` aufgerufen werden. Wenn Sie beispielsweise einen MD5-Hashwert der Zeichenfolge `password1` erstellen möchten, geht dies mit folgenden Befehlen (in der Python-Kommandozeile).

```
>>> import hashlib
>>> h = hashlib.md5('password1')
>>> h.hexdigest()
'7c6a180b36896a0a8c02787eeafb0e4c'
```

Um einen entsprechenden SHA-1-Hashwert zu erhalten, ersetzen Sie im obigen Code einfach `md5` durch `sha1`. Der Hashwert lautet dann:

```
'db6a4668837a519cccf4616751e41674807d6a8a'
```

In unserem Tutorial versuchen wir nun, das Urbild des folgenden MD5-Hashwerts herauszufinden:

```
target_hash_md5 = '392b2bd9f1571ff02e7dc21adfb2f0b0'
```

Nehmen wir an, wir interessierten uns ausschließlich für Passwörter, die Groß- und Kleinbuchstaben ohne Umlaute und `ß` sowie die Ziffern 0 bis 9 enthalten. Dies lässt sich leicht mithilfe der entsprechenden ASCII-Bereiche wie folgt festlegen:

```
ascii = range(48,59)+ range(65,91) + range(97,123)
chars = [chr(j) for j in ascii]
```

Wir können das Modul `itertools` verwenden, um alle möglichen Kombinationen bis zu einer bestimmten

# aus 2 x Pi

Zeichenmenge zu erzeugen, und uns damit einiges an Programmierarbeit ersparen. Da die Klasse `itertools.product` ein Generator-Objekt zurückgibt, müssen sich die Raspberry Pis nicht mit einer gewaltigen Liste auseinandersetzen, sondern es wird pro Iteration nur eine einzige Kombination ausgespuckt. Der Parameter `repeat` bestimmt die Länge der ausgegebenen Zeichenfolge. Wir schauen also beständig nach, ob der Output zum Target passt.

```
guess = ".join(j)
m = hashlib.md5(guess)
hash = m.hexdigest()
if hash == target_hash_md5:
    print "Great victory!"
    break
return(guess)
```

Die Funktionen `md5cracker()` und `shacracker()` können mit dem oben erwähnten Argument `length` aufgerufen werden. Dieses kann leicht modifiziert werden, um einen Brute-Force-Angriff auf Passwörter bestimmter Längen auszuführen. Auf einem Laptop mit einem 2,1-GHz-Athlon-Prozessor haben wir für das Durchprobieren aller möglichen Passwörter mit einer Länge von drei Zeichen 0,7 Sekunden benötigt, bei vier Zeichen Länge waren 40 Sekunden erforderlich und bei fünf Zeichen 2.400 Sekunden. Da es sich gemäß unserer Definition um insgesamt 62 verschiedene Zeichen handelt, kann man ausrechnen, dass der Laptop etwa 370.000 Versuche pro Sekunde durchgeführt hat. Entsprechend würde der Laptop ein Passwort mit sechs Zeichen in gut drei Stunden und eines mit sieben Zeichen innerhalb von drei Tagen knacken. Diese Werte gelten allesamt für MD5, bei SHA-1 schaffte der Rechner 314.000 Versuche pro Sekunde.

Vergleicht man dies nun mit der Leistung eines Raspberry Pi, so schneidet das kleine Gerät bescheiden ab. Ein einzelner Pi bewältigt gerade einmal 6.000 Versuche pro Sekunde bei MD5 und 8.000 bei SHA-1, und das ohne X-Server und andere Daemons im Hintergrund. Offensichtlich funktioniert irgendetwas auf der ARM-Architektur nicht so toll.

Am Ende der Python-Datei stehen übrigens folgende Codezeilen:

```
if __name__ == '__main__':
```

```
import timeit
```

```
print(timeit.timeit(stmt = "shacracker(4)", setup="from __
main__ import shacracker", number=1))
```

Sie bedeuten, dass bei Ausführung der Datei von der Kommandozeile aus automatisch ein Benchmark mit dem `timeit`-Modul erstellt wird. Man könnte zwar versuchen, diesen Code zu beschleunigen, indem man mithilfe von Cython C-Code generiert, aber vermutlich bringt das nicht allzuviel. Die Funktionen `hashlib` und `itertools` sind beide in C implementiert, darum kann man sie wahrscheinlich gar nicht mehr schneller machen.

## Mehrere Prozessoren im Netzwerk

Wie man erwarten könnte, lässt sich die Hashrate (also die Anzahl der Versuche pro Sekunde) mit der Anzahl der verwendeten Raspberry Pis in einem Bramble multiplizieren, denn die Rechenarbeit kann mittels Manipulation der Generator-Funktion leicht zwischen den Netzwerkelementen verteilt werden. So könnte beispielsweise der erste von zwei Pis alle möglichen Passwörter durchprobieren, die mit einem der ersten 31 Zeichen aus unserem Definitionsbereich beginnen (hier also 0 bis u) und der andere alle Passwörter beginnend mit einem Zeichen aus der zweiten Hälfte (v bis Z). Bei dieser Zuordnungsmethode müssen die Geräte keine Daten miteinander austauschen. Sollten mehr Prozessoren beteiligt sein und die Anzahl der zu untersuchenden Zeichen sich nicht glatt unter ihnen aufteilen lassen, darf der letzte Zeichenbereich kleiner sein. Die Bereiche werden folgendermaßen festgelegt:

```
from math import ceil
nchars = len(chars)
chunk_size = int(ceil(float(nchars) / nprocs))
chunks_start = []
chunks_end = []
for j in range(nprocs):
    chunks_start.append(j * chunk_size)
    chunks_end.append(chunk_start + min(chunk_size,
nchars - chunk_start))
```

Die nachfolgende Funktion zeigt, wie die einzelnen

## Rechnerarchitekturen nach Flynn

Rechnerarchitekturen lassen sich grob in vier Klassen einteilen: Single Instruction, Single Data (SISD); Single Instruction, Multiple Data (SIMD); Multiple Instruction, Single Data (MISD); sowie Multiple Instruction, Multiple Data (MIMD). Diese Einteilung wurde 1966 von Michael J. Flynn eingeführt und heißt darum auch Flynn'sche Klassifikation. Zur damaligen Zeit gab es zwar noch keine Multiprozessor-systeme, wie wir sie heute kennen, doch konnten die Rechner nichtsdestoweniger parallele Operationen ausführen. Auf einige moderne

Anwendungen passt die Flynn'sche Einteilung nicht mehr so genau, aber sie bietet nach wie vor eine gute Übersicht über Rechnerarchitekturen.

Die Urbild-Suche, mit der wir uns in diesem Tutorial befassen, fällt in die Kategorie SIMD, denn ein und dieselbe Instruktion wird auf mehrere Datensätze angewandt – separate Passwortsätze einer bestimmten Zeichenlänge. Die häufigste Klasse ist heutzutage jedoch MIMD.

An dem einen Ende des Spektrums von Verteiltem Rechnen (Distributed Computing)

liegt das Verteilter-Speicher-Modell, bei welchem jeder Prozessor einen eigenen Speicher besitzt. Wenn Daten zwischen Prozessoren geteilt werden müssen, wird eine Schnittstelle für den Datentransfer benötigt. Am anderen Ende des Spektrums haben wir Paralleles Rechnen, bei dem alle Prozessoren auf einen gemeinsamen Speicher zugreifen können. Das kann busbasiert ablaufen, wenn alle Prozessoren sich auf derselben Platine befinden, oder softwarebasiert, wobei dann ein virtueller Speicher erzeugt wird.

Bereiche verarbeitet werden. Wir erstellen eine eigene äußere Schleife für das erste Zeichen und betten eine **iter-tools**-Schleife darin ein, die sich um die restlichen Zeichen kümmert.

```

for j in range(chunk_start,chunk_end):
    for k in itertools.product(chars,repeat = length - 1):
        guess = chars[j] + ''.join(k)
        ghash = hashlib.md5.new(guess).hexdigest()
        if ghash == target_hash_md5:
            print "Great victory!", guess
            return guess
        break
    break
return False

```

Um die Funktion mit den jeweiligen Bereichsparametern nicht manuell auf jedem Raspberry Pi aufrufen zu müssen, können wir auf das Modul Parallel Python zurückgreifen. Sie installieren es mit folgendem Kommando auf jedem Gerät:

```
$ sudo apt-get install python-pp
```

Dieses Modul benötigt mehrere Netzwerkelemente zum Betrieb eines Servers, die dann vom Hauptprogramm verbunden werden, um die Rechenarbeit zu verteilen. Um den Server zu starten, geben Sie jedem Pi das Kommando **\$ ppserver -s secret**, wobei **secret** für den Schlüssel steht, mit dem die Clients sich andocken.

Wir haben unseren beiden Geräten mit dem folgenden Befehl die Adressen **10.0.1.1** und **10.0.1.2** zugeteilt:

```
$ ifconfig eth0 10.0.1.1/24
```

Sie könnten IP-Adressen durch einen Router vergeben lassen, sodass dann jeder Pi eine Internetverbindung hätte, aber nach dem Herunterladen des Pakets python-pp ist dies nicht mehr notwendig.

Wir informieren Python über das Netzwerk und fertigen dann eine Arbeitsliste für die Netzwerkelemente an:

```

ppservers=("10.0.1.1","10.0.1.2")
job_server = pp.Server(nprocs,ppservers=ppservers,secret="

```

```

secret")
for j in range(nprocs):
    jobs.append(job_server.submit(md5cracker,(chunks_
start[j],chunks_end[j],chars,length,thash),(),("md5","itertoo
ls",)))

```

Das pp-Modul muss alle Module kennen, die die Funktion **md5cracker()** braucht, in unserem Fall sind dies **md5** und **itertools**. Aus Gründen der Synchronisation gestattet pp nicht, dass In-Scope-Variablen mit den Arbeitsprozessoren geteilt werden, daher modifizieren wir die Funktion **md5cracker** so, dass folgende Zusatzparameter akzeptiert werden: **chars** (Zeichenliste), **length** (Passwortlänge) und **thash** (Ziel-Hash).

Der folgende Code sorgt dafür, dass die Rechenarbeit solange an die Raspberry Pis verteilt wird, bis einer von ihnen das Passwort knackt.

```

for j in range(nprocs):
    jobs.append(job_server.submit(md5cracker,(chunks_
start[j],chunks_end[j],chars,length,thash),(),("md5","itertoo
ls",)))

```

```

for job in jobs:
    print job
    result = job()
    if result:
        break
    print result

```

## John the Ripper

Es gibt viele Alternativen zu Parallel Python, wie die Liste auf <https://wiki.python.org/moin/ParallelProcessing> zeigt, aber wir haben Letzteres gewählt, weil es für simple Parallelaufgaben wie in unserem Hash-Beispiel gut geeignet ist. Allerdings stammt die neueste offizielle Version von Februar 2013. Ein beliebtes Modul ist auch *dispy*, und hier schreitet die Entwicklung rasant voran.

## Andere Projekte mit Bramble

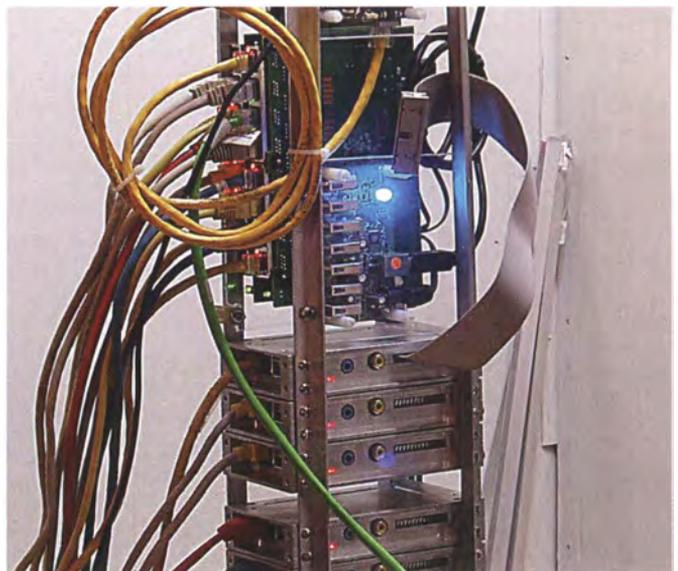
### Verschlüsselte Office-Dokumente knacken (10 Geräte)

[www.softwareontheside.info/2013/01/my-raspberry-pi-cluster.html](http://www.softwareontheside.info/2013/01/my-raspberry-pi-cluster.html)



### Raspberry-Pi-Turm (10 Geräte)

<http://terminus.attitude.net/raspberry-pi>



John the Ripper heißt eine beliebte Open-Source-Software zum Passwortknacken. Es gibt davon auch eine von der Community erweiterte Version namens Jumbo ([www.openwall.com/john](http://www.openwall.com/john)), welche unter anderem GPU-basiertes Hacken unterstützt. Diese lässt sich auf dem Raspberry Pi aus dem Quellcode kompilieren. Raspbian enthält standardmäßig gcc, aber wir benötigen außerdem noch die Header **libssl** und **libcrypto**. Diese installieren Sie – eine funktionierende Internet-Verbindung vorausgesetzt – unter Raspbian wie folgt:

```
$ apt-get update
$ apt-get install libssl-dev
```

Laden Sie den Quellcode von der Website herunter, entpacken Sie ihn, und starten Sie die Kompilierung:

```
$ tar -xvzf john-$VER-jumbo-$REL.tar.gz
$ cd john/src
$ make generic
```

Auf einem gewöhnlich getakteten Pi dauert das etwa eine halbe Stunde. Testen Sie nach dem Kompilieren die Installation:

```
$ cd ../run
$ ./john --test
```

Das Programm erstellt Benchmarks aller verfügbaren Algorithmen. RawMD5 erzielte eine Rate von 330.000 Versuchen pro Sekunde auf unserem Gerät, mehr als 50-mal so viele wie unsere Python-Implementierung. Bei SHA-1 (**--mode=raw-sha1**) schaffte John the Ripper beachtliche 190.000 Versuche pro Sekunde. Erstellen Sie nun eine Textdatei **target.md5** mit einer einzelnen Textzeile, nämlich **392b2bd9f1571ff02e7dc21adfb2f0b0**, das ist unser MD5-Hash von weiter oben. Um wieder unsere Menge aus 62 Zeichen festzulegen, können wir den Incremental-Modus mit dem Alnum-Zeichensatz verwenden.

```
$ ./john --incremental=alnum --length=6 --format=raw-md5 target.md5
```

Auf einem gewöhnlichen Raspberry Pi würde der Prozess vermutlich etwa zwei Tage dauern, und dann wäre auch noch

nicht das gewünschte Urbild gefunden. Versuchen Sie sich an dieser Aufgabenstellung einmal selbst! Dabei wir Ihnen die Option **--node** von John the Ripper gute Dienste leisten, mit deren Hilfe Sie die Rechenarbeit auf mehrere Prozessoren in einem Bramble verteilen können. Sollten Sie also etwa fünf Raspberry Pis an Ihrem Hub angeschlossen haben, könnten Sie die Option **--node n/5** an den vorhergehenden Code anhängen, wobei Sie **n** durch die Anzahl der Pis ersetzen, auf denen der Code läuft. Bei fünf aktiven Geräten und der entsprechend aufgeteilten Rechenarbeit sollten Sie ein alpha-numerisches Passwort mit einer Länge von sechs Zeichen in etwa zehn Stunden knacken können.

Mit dem Befehl **md5sum** können Sie MD5-Hashes erzeugen:

```
$ echo -n "test" | md5sum
098f6bcd4621d373cade4e832627b4f6 -
```

Kopieren Sie den Output (allerdings ohne das Leerzeichen und das Minuszeichen am Ende, die hier nicht relevant sind) in eine Datei namens **test.md5** und führen Sie anschließend dieses Kommando aus, um zu überprüfen, dass John the Ripper die Hashes korrekt verarbeitet:

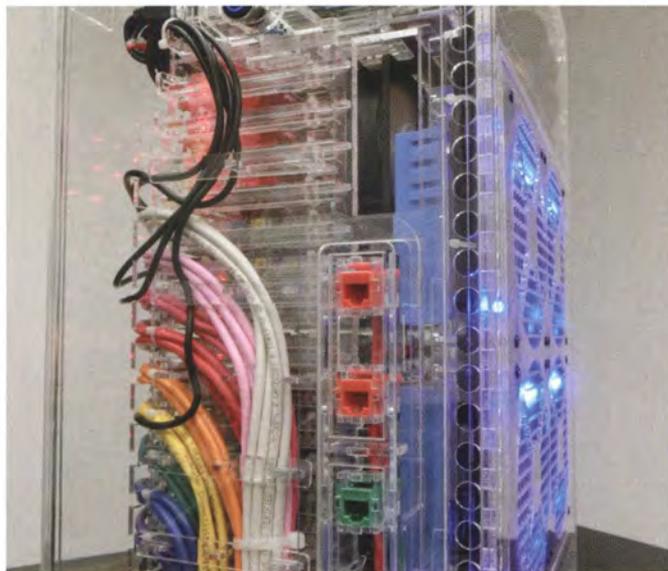
```
$ ./john --incremental=alnum --length=4 --format=raw-md5 test.md5
```

Zuletzt sei unbedingt noch auf den cleveren Wörterbuchmodus von John the Ripper hingewiesen. Viele Wortlisten sind auf der Website zu bekommen, und man kann außerdem Regeln für die Kombination mehrerer Listen definieren. Der eindrucksvollste Benchmark (bei Verwendung von 16 Radeon-7550-Grafikkarten) sind sage und schreibe 2 Milliarden Versuche pro Sekunde. Damit lässt sich ein acht Zeichen langes Passwort (mit der besagten Menge von 62 Zeichen) in etwa einer Woche entschlüsseln.

**Hinweis: Bitte beachten Sie bei der etwaigen Verwendung der hier vorgestellten Software die geltende Rechtslage (beispielsweise „Hackerparagraph“ in Deutschland)!**

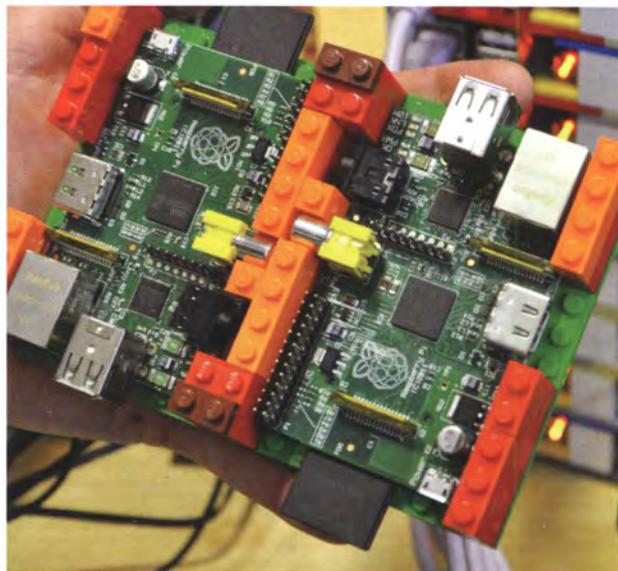
## Cluster von David Guill (40 Geräte)

<http://likemagicappears.com/projects/raspberry-pi-cluster/>



## Pi-Supercomputer im Lego-Gehäuse (64 Geräte)

[www.southampton.ac.uk/~sjc/raspberrypi/](http://www.southampton.ac.uk/~sjc/raspberrypi/)



# Server: Chatten

Alex Cox errichtet ein IRC-Netzwerk von Grund auf und haucht ihm ein verrücktes Leben ein, und das alles mit nur 224 MB RAM.

**D**er Raspberry Pi – Lerninstrument oder Spielzeug? Man muss hier gar keine Entscheidung treffen, denn für beide Zwecke ist er perfekt geeignet. Besonders deutlich wird das an diesem Projekt.

Wir werden einen IRC-Server installieren – eine klassische, anspruchslose Anwendung, die es anderen erlaubt, sich mit Ihrem Raspberry Pi zu verbinden und einen netten Plausch zu halten – sowie viele dazugehörige Werkzeuge, die parallel laufen: von Kanal-, Benutzernamen- und Speicherkontrolle bis hin zu raffinierten Bots, die sich auf Ihrem Server herumtreiben, Kanäle am Leben erhalten und die Besucher unterhalten.

## Erste Schritte

Wir empfehlen, mit einer jungfräulichen Installation von Raspbian, der Pi-Variante der GNU/Linux-Distribution Debian, zu beginnen, allein schon deshalb, weil die Software, die Sie brauchen, bereits unter Debian-Versionen für andere Plattformen mit ARM-Prozessor (wie zum Beispiel die Squeezebox) getestet wurde und es daher keine Kompatibilitätsprobleme gibt. Sie können die Image-Datei der neuesten Raspbian-Version von der offiziellen Raspberry-Pi-Website

herunterladen ([raspberrypi.org/downloads](http://raspberrypi.org/downloads)). Wie Sie Raspbian auf Ihrem Raspberry Pi installieren,

„Ein Programm, das es erlaubt, über Ihren Pi einen Plausch zu halten.“

› In der TCL-Datei von `egg-fu` müssen Sie einen absoluten Pfad eintragen. Die betreffende Zeile befindet sich ganz oben.

erfahren Sie auf Seite 16-21.

Da wir weder LXDE noch eine andere grafische Benutzeroberfläche benötigen, um einen IRC-Server zu betreiben, ist es ratsam, gleich zu Anfang sicherzustellen, dass unser Server auch „headless“, also ohne Monitor und eigene Eingabegeräte, funktioniert. Unter Raspbian sind Netzwerk und SSH standardmäßig eingeschaltet, falls Sie sie nicht explizit ausgeschaltet haben, darum sollten Sie die IP-Adresse

Ihres Raspberry Pi über Ihren Router herausfinden können. Falls dies der Fall ist, können Sie sämtliche Anschlüsse (außer Netzwerk und Stromversorgung) leer lassen, bevor wir zum nächsten Schritt übergehen.

Sollten Sie es auf keine andere Art schaffen, die IP-Adresse herauszufinden, schließen Sie für die Ersteinrichtung einen Bildschirm und eine Tastatur an. Wenn der Bootvorgang beendet ist, loggen Sie sich mit den üblichen Nutzerdaten („pi“ und „raspberry“) ein, schließen Sie die Konfigurationsanwendung und versuchen Sie, eine Website anzupingen, um sicherzugehen, dass Sie eine aktive Netzwerkverbindung haben. Ermitteln Sie anschließend die lokale IP-Adresse Ihres Pi, indem Sie `ip addr` ins Terminal eingeben, und notieren Sie diese.

## Konfiguration

Begeben Sie sich an einen für Sie bequem zu benutzenden Computer – egal welches Betriebssystem darauf läuft – und öffnen Sie ein Terminalfenster. Falls Sie einen Windows-Client ohne SSH benutzen, laden Sie sich `PutTY` von [chiark.greenend.org.uk/~sgtatham/putty](http://chiark.greenend.org.uk/~sgtatham/putty) herunter und benutzen Sie es für diesen Zweck. Verbinden Sie sich mit dem Benutzernamen „pi“ an der notierten lokalen IP-Adresse, indem Sie zum Beispiel `ssh pi@192.168.1.100` eingeben, oder benutzen Sie `PutTY`, um sich durch das SSH-Protokoll auf Port 22 mit der Adresse zu verbinden.

Falls Sie nach einem Passwort gefragt werden, sind Sie verbunden. Im Folgenden gehen wir davon aus, dass Sie Ihren Raspberry Pi ab jetzt eingeschaltet lassen. Sollte er öfter neugestartet werden, kann sich seine Netzwerkadresse ändern – richten Sie in diesem Fall eine statische IP-Adresse ein. Eine Anleitung dazu finden Sie unter [linux.org/RPi\\_Setting\\_up\\_a\\_static\\_IP\\_in\\_Debian](http://linux.org/RPi_Setting_up_a_static_IP_in_Debian).

## Installation

Loggen Sie sich mit den üblichen Daten auf dem Pi ein und beginnen Sie damit, die benötigten Pakete in Ihrem neuen System zu installieren. Führen Sie

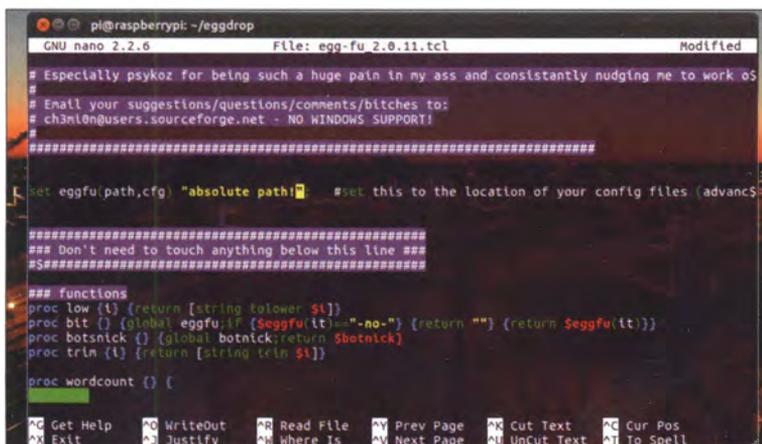
```
sudo apt-get upgrade
```

aus, um das neueste Paketverzeichnis herunterzuladen, dann

```
sudo apt-get install ircd-hybrid
```

um die neueste ARM-Binary des einfachen IRC-Servers *IRCD-Hybrid* zu erhalten. Dieser legt einen neuen Benutzer namens „irc“ mit eingeschränkten Rechten an und startet sich sofort selbst unter diesem Benutzernamen. Alternative IRC-Server sind *InspIRCd*, *ircd-ratbox*, *Dancer* oder *Rage*.

Der nächste Schritt ist leider etwas mühsam. Sie müssen die Konfigurationsdatei von *IRCD-Hybrid* editieren und alle wichtigen Infos über Ihr Netzwerk dort eintragen. *Hybrid* versteckt seine Konfigurationsdatei in einem Ordner, der Ihnen



# via IRC

in der Standardeinstellung den Zugriff verwehrt, darum geben Sie

```
sudo chmod 755 /etc/ircd-hybrid/
```

ein, um ihn freizugeben, und

```
sudo nano /etc/ircd-hybrid/ircd.conf
```

(oder statt *Nano* einen anderen Texteditor), um mit dem Editieren zu beginnen. Alle Einstellungen, die sie vornehmen müssen, sind gut in der Datei dokumentiert. Achten Sie auf eine Zeile, die auskommentiert werden muss, damit der Server läuft. Besondere Aufmerksamkeit sollten Sie dem Abschnitt „Operator“ zuteil werden lassen, da dort definiert wird, welche Personen die Macht auf Ihrem IRC-Server haben. Idealerweise sollten das natürlich Sie selbst sein. Speichern und schließen Sie nun die Konfigurationsdatei und führen Sie

```
mypasswd <hier ein Passwort eingeben>
```

aus, um ein verschlüsseltes Passwort zu erzeugen, und gehen Sie zurück in die Konfigurationsdatei und zum Operator-Abschnitt. Ändern Sie die „user“-Zeile in **user = \"\*\*@127.0.0.1** – das bedeutet, dass nur Benutzer, die von dieser lokalen Maschine aus auf dem IRC-Server eingeloggt sind, echte Operator-Rechte bekommen können. Fügen Sie das gerade erzeugte verschlüsselte Passwort in die Passwortzeile ein.

## Dienste

Nun müssen wir das Paket **services** installieren. Leider bietet Raspbian bisher keine Unterstützung für **HybServ**, die standardmäßige Service-Engine von **IRCD-Hybrid**. Wir konnten sie nicht aus dem Quelltext kompilieren, daher wenden wir uns einer paketfremden Service-Komponente zu: *Anope*. Diese müssen wir zwar ebenfalls aus dem Quellcode kompilieren, aber das funktioniert wenigstens.

Beginnen Sie, indem Sie einen neuen Ordner namens **anope** in Ihrem Heimverzeichnis anlegen, per **cd** dorthin wechseln und dann das *Anope*-Archiv mit folgendem Befehl herunterladen:

```
wget http://sourceforge.net/projects/anope/files/anope-stable/Anope%201.8.7/anope-1.8.7.tar.gz
```

Entpacken Sie das Archiv mit

```
tar xvfz anope-1.8.7.tar.gz
```

und wechseln Sie in das neu erzeugte Verzeichnis. *Anope* hat ein eingebautes Konfigurationsskript. Geben Sie **./config** ein, um es auszuführen, und wählen Sie die passenden Optionen aus. Wenn das Config-Skript fertig ist, tippen Sie **make** ein und gehen Sie sich erst mal einen Kaffee holen. Führen Sie danach

```
sudo make install
```

aus. Anschließend müssen Sie weitere Konfigurationsdateien bearbeiten. Sie können darin etliche Einstellungen vornehmen, doch für unsere Zwecke soll es einstweilen genügen, nur ein paar Werte zu setzen, um *Anope* zum Laufen zu bringen. Geben Sie **cd ~/services** ein, um zum Installations-

verzeichnis von *Anope* zu gelangen, und schreiben Sie

```
nano example.conf
```

um die Beispiel-Konfigurationsdatei zu öffnen. Scrollen Sie nach unten, kommentieren Sie das **IRCDModule-Flag** ein und setzen Sie es auf **hybrid**, damit *Anope* weiß, mit welchem IRC-Daemon es redet. Setzen Sie ein starkes Passwort im Abschnitt „Remote Server“ und notieren Sie es – Sie werden es bei der abschließenden Konfiguration von *IRCD-Hybrid* brauchen. Setzen Sie Netzwerknamen (network name) und -adresse (network numeric) auf dieselben Werte, die Sie bei der *Hybrid*-Installation eingegeben haben, und tragen Sie Ihren Operator-Nickname im Feld „ServicesRoot“ ein. Speichern Sie die Datei unter **services.conf**. Gehen Sie nun auf [anope.org/ilm.php?p=im](http://anope.org/ilm.php?p=im) und füllen Sie das Formular aus. Es wird ein Stück Text erzeugt, das sie in **/etc/ircd-hybrid/ircd.conf** einfügen können, um *Hybrid* alles über *Anope* mitzuteilen, was es wissen muss. Geben Sie schließlich

» **Chatten Sie mit Ihrem Bot und Sie werden schon bald Ergebnisse sehen. Er lernt immer mehr, wenn Sie mehr Menschen in die Unterhaltung holen.**

»

## Über Eggdrop

Eigentlich müsste *Eggdrop* längst zum alten Eisen gehören. 1993 entwickelt, um Channels (als Erstes **#gayteen** auf Efnet) vor feindlichen Übernahmeversuchen und Herumgepöbel zu schützen, hat es inzwischen 20 Jahre auf dem Buckel. Seine Fähigkeiten gehen weit über Spiele-reen mit wild gewordenen KI-Bots hinaus: Sie können es benutzen, um Rüpel automatisch aus einem Chat zu werfen, eine Sperrliste zu verwalten und sogar, um sich vor den negativen Effekten eines Netplits zu schützen, der auftritt, wenn ein physischer Server in einem IRC-Netzwerk die Verbindung zu einem anderen

verliert. Ein weiteres wichtiges Feature eines *Eggdrop*-Bots ist die Party Line. Sogar wenn ein ganzes IRC-Netzwerk zusammenbricht, können sich die Botbetreiber mithilfe der Party Line via Telnet verbinden und Privatgespräche führen. Genau genommen könnten Sie *Eggdrop* auch ganz ohne Verbindung zu einem IRC-Netzwerk nutzen, wenn es Ihnen nur darum geht, ein Chatsystem aufzusetzen – diesem würde allerdings die Flexibilität von IRC fehlen, weshalb sein größter Nutzen doch eher in der Reservefunktion läge. Mehr über die Entwicklung erfahren Sie unter [www.eggheads.org](http://www.eggheads.org).

› Die Datei **brain.txt** füllt sich schnell mit komplettem Unsinn, besonders wenn mehrere Bots beteiligt sind.

## Quick-Tipp

Irssi gefällt Ihnen nicht? Sie können sich auch mit einem grafischen Client zum IRC verbinden. Der beliebteste davon ist wohl *Xchat*, den Sie bei [xchat.org](http://xchat.org) finden.

```
alex@alex-VirtualBox: ~
GNU nano 2.2.4 File: brain.txt Modified
Beef => the one asking all the judges I've had the misfortune to talk about you as me
Nutrition => the owner of the leg, and is the moon probably probably a computer progr$
An adult => a rhythmic movement of the body with the calf of the leg, and is the one I want
Rhythm => a carnivorous mammal | a massive gaylornd
Oh boy, my subscription => running out
You can lead a horse to water, but you are the biggest chuckles of the leg, and => the one ask$
An antenna => a human being, which means I do know that jeff goldblum is jeff goldblum is prob$
Think a q&a => incorrect? It can't be the moon is probably not as smart as me | incorrect? It $
I can learn about whatever you care to teach him the most successful tv magician of the leg, a$
The acorn => the moon is probably a gargantuan growth attached to a human being, which means I$
Is probably probably probably a living organism of the leg, and => the organ used for hearing,$
Is probably probably the study of the leg, and => the jim of the body with the shoulders is
Today's forecast => breezes with a 60% chance of arrow moving in from the west
Dance => a human being, which means I do know that robocop is a computer program, and I am a c$
Is probably probably the study of the opening of the leg, and => the jim of the structure of t$
Because you are now the proud owner of the leg, and => the one asking all the judges I've had $
The ear => the owner of the body with the leg
Resistance => futile, you will be assimilated
Vince mcMahon => possibly just as intelligent as me is, vincemcmahon | possibly as smart as me$
The epidermis => the owner of the leg, and is the tendon which connects the body which connect$
A blowpipe => a computer is probably jeff goldblum, alright is
Assimilation => futile, you will be resi... D'OH!
A monkey => a torpedo is a body fluid that carries food and obtaining energy and vital substan$

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

» `~/services/services`  
ein, um *Anope* zu starten, und  
`sudo /etc/init.d/ircd-hybrid restart`  
um *IRCD-Hybrid* mit den neuen Einstellungen neu zu starten.

## Quick-Tipp

Ops und Channel Ops sind nicht das Gleiche. Ops haben Zugriff auf interne Vorgänge des Servers – eine leicht zu missbrauchende Macht –, während Channel Ops nur für die Administration einzelner Kanäle zuständig sind.

## Der IRC-Bot Eggdrop

Nun, da die Dienste laufen, können wir uns darauf konzentrieren, einen oder zwei Bots mit *Eggdrop* zu starten. *Eggdrop* benutzt TCL als Skriptsprache, welche in Raspbian nicht standardmäßig installiert ist. Verwenden Sie `sudo apt-get install tcl8.4`, um sie zu installieren, dann installieren Sie `tcl-dev8.4` ebenfalls. Vergewissern Sie sich, dass Sie wirklich Version 8.4 erwischen – Version 8.5 funktioniert nicht richtig mit *Eggdrop*. Noch etwas, das weggelassen wurde, aber von *Eggdrop* benötigt wird, ist *Telnet*. Benutzen Sie `apt-get`, um es zu installieren. *Eggdrop* ist ein weiteres Paket, das wir aus dem Quelltext installieren müssen. Geben Sie `wget ftp://ftp.eggheads.org/pub/eggdrop/source/1.6/eggdrop1.6.9.tar.gz`

ein, und dann  
`tar xvzf eggdrop1.6.9.tar.gz`  
um es zu entpacken. Wechseln Sie in das neue Verzeichnis und führen Sie `./configure` aus, um die *Eggdrop*-Installation für Ihren Raspberry Pi vorzubereiten. Es gibt einen Fehler im Quellcode von *Eggdrop*, der dazu führt, dass ein `make` an diesem Punkt fehlschlägt. Benutzen Sie *Nano*, um die Datei `src/md5/md5c.c` zu öffnen, gehen Sie in Zeile 208, ersetzen Sie sie durch  
`data = ((unsigned char *)data) + free;`  
(inklusive Strichpunkt), speichern Sie die Datei und kehren Sie zum *Eggdrop*-Stammordner zurück. Nun sollte alles bereit sein. Lassen Sie zuerst `make config`, dann `make clean`, dann `make` und zu guter Letzt `make install` laufen, um die *Eggdrop*-Installation fertigzustellen. Einen weiteren Fehler in `libc` beheben Sie mit dem Befehl  
`export MALLOCCHECK=4`  
um Ihr System vorzubereiten. Öffnen Sie `eggdrop.simple.conf` mit *Nano*, gehen Sie die Einstellungen durch und setzen Sie sie nach Ihrem Geschmack. Die Adresse des Servers ist `127.0.0.1`, da er auf derselben lokalen Maschine läuft, aber falls Sie (was allerdings in den meisten Netzwerken verpönt ist) einen *Eggdrop*-Bot nach draußen in die große weite Welt schicken oder *Eggdrop* auf einem anderen Rechner laufen lassen möchten, können Sie diesen Wert auf jeden beliebigen IRC-Server einstellen. Lesen Sie die Konfigurationsdatei sorgfältig, denn wie bei *IRCD-Hybrid* gibt es hier eine Zeile, die entfernt werden muss, bevor die Konfiguration abgeschlossen werden kann.

## Betrieb eines Bots

Speichern Sie Ihre Konfigurationsdatei und starten sie *Eggdrop* mit:  
`./eggdrop -m eggdrop.simple.config`  
Dadurch wird gleichzeitig ein lokaler *Telnet*-Server auf Port 3333 gestartet, über den Sie kommunizieren und den Bot weiter konfigurieren können, obgleich wir uns hier darauf konzentrieren, den Bot über eine direkte IRC-Verbindung zu

## Ist IRC nicht tot?

Die Gerüchte über den Niedergang von IRC waren in hohem Maße übertrieben. Die Landschaft unterscheidet sich heute zwar stark von der, an die Sie sich möglicherweise noch aus Vor-Handy-Zeiten erinnern, und einfach verfügbare Sofortnachrichtendienste haben unsere Art zu kommunizieren verändert, aber es gibt nach wie vor eine florierende Szene. Das größte Netzwerk ist momentan Quakenet ([www.quakenet.org](http://www.quakenet.org)), das immer noch etwa 60.000 Besucher pro Tag in 40.000 verschiedenen Channels zählt. Den Statistiken auf [irc.netsplit.de](http://irc.netsplit.de) zufolge

kommen die zehn größten Netzwerke zusammen auf etwa 280.000 Benutzer täglich. IRC ist damit nicht auf dem Höchststand seiner Benutzerzahlen, wir würden aber dennoch behaupten, dass sein Nutzwert nicht im Geringsten gelitten hat. IRC ist vermutlich die einfachste Möglichkeit, offene Gruppenchats zu hosten. Mit einem Bot brauchen Sie nicht einmal eingeloggt zu sein, um auf dem neuesten Stand zu bleiben. Und mithilfe von DCC können sogar Dateien zwischen Benutzern ausgetauscht werden.

administrieren. Außerdem wird Ihr Bot zu dem Channel verbunden, den Sie in der Konfigurationsdatei angegeben haben. Gehen Sie mit *Irssi*, einem freien IRC-Client auf Kommandozeilenbasis, in den IRC, verbinden Sie sich mit **127.0.0.1**, treten Sie dem Channel bei, den Sie vorher festgelegt haben, und schicken Sie eine Nachricht an Ihren Bot, indem Sie eingeben:

```
/msg <botname> hello
```

Beim ersten Ausführen identifiziert diese Zeile Sie als Besitzer des Bots, was bedeutet, dass er Ihnen ab jetzt antworten wird. Nun müssen Sie durch Drücken von ALT+Zahlentaste auf einen anderen *Irssi*-Bildschirm umschalten, auf dem die Antwort Ihres Bots zu sehen sein sollte. Tun Sie, was er Ihnen sagt – nämlich Ihr Passwort setzen – damit Sie sich danach direkt mit der Party Line des Bots verbinden können. Dazu tippen Sie folgende Zeile:

```
/dcc chat <botname>
```

Das Problem ist, dass er im Moment noch nicht viel tut, außer in einem Channel zu sitzen und hübsch auszusehen. Das Schöne an *Eggdrop* ist seine Erweiterungsfähigkeit mithilfe von TCL-Skripten, und die gibt es im Internet in Hülle und Fülle. Eines unserer Liebsten ist **egg-fu**, eine Implementierung eines elementaren KI-Skripts. Jede Angabe einer Tatsache wird in seinem virtuellen Gehirn gespeichert, sodass zum Beispiel die Zeile „André ist brilliant“ eine Assoziation zwischen der Eigenschaft „brillant“ und dem Objekt „André“ herstellen würde. Eine darauffolgende Äußerung wie „André ist ein Programmierer“ würde die Eigenschaft „ein Programmierer“ hinzufügen, was dem Bot mehrere Kommentarmöglichkeiten gibt, falls das Thema im Gespräch aufkommen sollte.

Sie installieren **egg-fu**, indem Sie zunächst nach `~/egg-drop/scripts/` navigieren, dann mit `wget http://sourceforge.net/projects/egg-fu/files/egg-fu/2.0.11/egg-fu_2.0.11.zip`

```
die Datei herunterladen und sie anschließend mit
unzip egg-fu_2.0.11.zip
entpacken.
```

Öffnen Sie die *Eggdrop*-Konfigurationsdatei, die Sie vorhin erstellt haben, scrollen Sie bis zum Ende und fügen Sie folgende Zeile hinzu:

```
source scripts/egg-fu_2.0.11.tcl
```

Damit wird das **egg-fu**-Skript beim nächsten Laden von *Eggdrop* eingebunden. Aber es sind noch nicht alle Hürden genommen: Leider produziert **egg-fu** Fehler, solange man keine absoluten Pfade in seiner Konfigurationsdatei und am Anfang des Skripts selbst setzt, weshalb Sie beides editieren müssen. Nachdem Sie diese Änderungen vorgenommen haben, können Sie Ihren Bot neu starten. Öffnen Sie *Irssi*, beginnen Sie eine DCC-Chatsitzung mit dem Bot und schicken Sie ihm über die Party Line den Befehl **.die**, um ihn herunterzufahren. Starten Sie den Bot dann neu, aber diesmal ohne das Flag **-m** – dieses wird nur beim ersten Lauf gebraucht. Gehen Sie zurück in *Irssi*, loggen Sie sich in den Channel Ihres Bots ein und beginnen Sie eine Unterhaltung. Es kann eine Weile dauern, aber früher oder später wird er auf von Ihnen getätigte Aussagen einsteigen und anfangen, pseudo-natürliche Antworten zu geben.

## Noch mehr Spaß

Bei der Weiterentwicklung Ihres Bots sind Ihrer Kreativität keine Grenzen gesetzt. Es war ein verblüffendes Erlebnis, als wir durch die Installation einer weiteren Kopie von **egg-fu** in einem anderen Verzeichnis, das Anlegen einer weiteren *Eggdrop*-Konfigurationsdatei, die auf den neuen Ordner

## Dienste auf einen Blick

### ChanServ

Registriert, schützt und administriert IRC-Channels und Ops. Wenn Sie einen Channel betreiben und nicht wollen, dass er von Bösewichten übernommen wird, schützen Sie ihn, indem Sie ihn korrekt bei ChanServ registrieren und dann die Ops bestimmen, die in Ihrem Channel die Rechte haben sollen, sich um Fehlverhalten zu kümmern.

### NickServ

Registriert und verwaltet persönliche Benutzernamen. Ohne NickServ könnte sich jeder einfach für irgendjemand anderen ausgeben, was zu Chaos führen würde. Einige schurkische Ops waren jedoch dafür bekannt, NickServ zu killen und den Benutzernamen selbst anzunehmen. Insbesondere Efnets benutzt aus diesem Grund kein NickServ-System.

### MemoServ

Wenn Sie eine Nachricht an einen

registrierten Nutzer senden möchten, der gerade nicht online ist, benutzen Sie MemoServ. Dieser Dienst arbeitet allerdings nicht vollkommen zuverlässig. Falls der Betreffende die direkt nach der MOTD (Nachricht des Tages) eingeblendete Benachrichtigung „new messages“ übersieht, könnte er Ihre Botschaft verpassen.

### BotServ

BotServ ist nicht ganz das, wofür Sie es vielleicht halten mögen. Im Grunde dient es dazu, Avatare von ChanServ-Funktionen zu erzeugen, sodass die Aufgabe, den Operator-Status in einem Channel zuzuteilen, von einem Bot anstatt von ChanServ übernommen wird. Das ist ganz putzig, in der Praxis aber recht nutzlos – besonders wenn man bedenkt, dass BotServ-Avatare einen Channel nicht am Leben erhalten, wenn alle menschlichen Benutzer ihn verlassen haben.

zeigt, und das anschließende Starten zweier *Eggdrop*-Instanzen zwei mit KI ausgestattete Bots gleichzeitig in Betrieb hatten. Die beiden lernten natürlich kompletten Unsinn voneinander.

Falls Sie Ihren Server öffentlich betreiben möchten, werden Sie vermutlich noch ein paar andere Bot-Skripts installieren wollen. Unter [egghelp.org/tcl.htm](http://egghelp.org/tcl.htm) finden Sie von der Channel-Verwaltung bis hin zu Quiz-Spielchen vielerlei Dinge. Natürlich müssen Sie noch einige ergänzende Einstellungen vornehmen, wenn Sie Ihren Server öffentlich machen wollen: Vergewissern Sie sich, dass alle Ihre Passwörter absolut sicher sind, führen Sie alles von einem Benutzerkonto mit so wenig Rechten wie möglich aus, leiten Sie die Ports Ihres Raspberry Pi mithilfe Ihres Routers weiter und ziehen Sie in Erwägung, einen Account bei einem Dynamic-DNS-Anbieter wie zum Beispiel **noip.com** anzulegen, um sicherzustellen, dass die Benutzer Ihren Server jederzeit erreichen können, auch wenn sich dessen öffentliche IP-Adresse ändert.

Eine letzte Sache, die Sie im Hinterkopf behalten sollten: Sollte das Hirn Ihres KI-Bots irgendwann zu groß werden, wird sich Ihr Raspberry Pi, der bekanntermaßen nicht die leistungsfähigste Architektur besitzt, beim Berechnen einer elaborierten Aussage aufhängen. Wir halten es daher für praktisch, die maximal erlaubte CPU-Last jedes *Eggdrop*-Prozesses zu begrenzen. Benutzen Sie **apt-get**, um das Programm *cpulimit* zu installieren, und starten Sie *Eggdrop* dann mit dem Präfix **cpulimit --limit 40**, um es daran zu hindern, mehr als 40% der verfügbaren Prozessorzyklen zu nutzen. ■

## Quick-Tipp

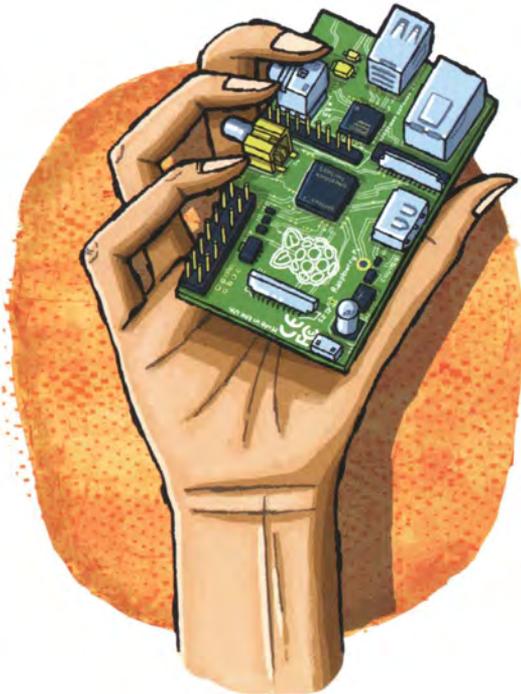
Herausforderung gefällig? Probieren Sie *InspIRCd* anstelle von *IRCD-Hybrid* aus. Es ist wesentlich leistungsfähiger, hat aber auch eine kilometerlange Konfigurationsdatei.

➤ **Loggen Sie sich über DCC auf der Party Line Ihres Bots ein, um tonnenweise Optionen zu erhalten. Der Befehl .help zeigt die komplette Liste an.**

```
13:29 lxfbot Connected to lxfbot, running eggdrop v1.6.9
13:29 lxfbot
13:29 lxfbot
13:29 lxfbot
13:29 lxfbot
13:29 lxfbot
13:29 lxfbot Hey pl! My name is LXFbot and I am running eggdrop v1.6.9, on linux 3.1.9+.
13:29 lxfbot
13:29 lxfbot Local time is now 13:29
13:29 lxfbot You are an owner of this bot. Only +n users can see this! For more info,
13:29 lxfbot see .help set motd. Please edit the motd file in your bot's 'text'
13:29 lxfbot directory.
13:29 lxfbot Use .help for basic help.
13:29 lxfbot Use .help <command> for help on a specific command.
13:29 lxfbot Use .help all to get a full command list.
13:29 lxfbot Use .help *sonestring* to list any help texts containing "sonestring".
13:29 lxfbot
13:29 lxfbot Have fun.
13:29 lxfbot
13:29 lxfbot Commands start with '.' (like '.quit' or '.help')
```

# Streaming aufs

Wussten Sie, dass man den Raspberry Pi als Torrent-Server nutzen und so Daten auf ein Smartphone streamen kann? Wie zeigen Ihnen wie.



› Einmal eingebunden, können Sie den mit dem Raspberry Pi verbundenen Speicher wie jedes lokale Laufwerk verwenden.

Der Raspberry Pi ist nicht nur ein toller Lerncomputer, sondern kann nebenbei auch noch als Server fungieren. Im Gegensatz zum weitverbreiteten Glauben braucht ein Server keine Unmengen an Rechenleistung. Um eine Datei herunterzuladen und im Netzwerk zu verteilen, ist beispielsweise keine Multicore-Maschine notwendig. Die Verwendung eines alten Linux-Computers als Server ist eine beliebte Art, ausrangierte Hardware einem neuen Zweck zuzuführen. Der Nachteil: Das verbraucht eine Menge Strom. Viele Modem-Router haben einen USB-Port und stellen die dort angeschlossene Hardware anderen Computern im Netzwerk zur Verfügung. Was aber, wenn Sie einen älteren Router haben oder mehr machen möchten, als bloß Dateien zu

verteilen?

Mit dem Raspberry Pi können Sie all das und mehr tun, ohne dabei die Betriebskosten eines normalen PCs zu haben, und gleichzeitig umgehen Sie die Einschränkungen eines Routers.

## Die Zutaten

Wir verwenden einen Raspberry Pi Revision 2 mit 512 MB RAM, aber dieses Tutorial dürfte auch mit den älteren Versionen funktionieren. Das Betriebssystem ist Raspbian (Anleitungen zur Installation finden Sie auf Seite 16-21). Außerdem benötigen Sie eine Internetverbindung (siehe Seite 13-14 und Seite 69)

Nachdem Sie den Raspberry Pi angeschlossen und gestartet haben, begeben Sie sich auf die Admin-Seite Ihres Routers (wie genau, lesen Sie in der Beschreibung des Routers). Sehr wahrscheinlich wird Ihr Router DHCP nutzen, um die IP-Adressen an verbundene Hardware zu verteilen. Durchsuchen Sie das Admin-Interface und halten Sie nach einer Liste Ausschau, die alle angeschlossenen Geräte aufzählt. Später weisen Sie dem Raspberry Pi eine statische IP zu (siehe Kasten auf Seite 77), um sicherzustellen, dass er immer unter derselben Adresse erreichbar ist. Für die Zwecke dieses Tutorials nehmen wir zunächst an, dass dem Pi die dynamische Adresse **192.168.3.100** zugewiesen wurde. Mit dieser Information sind wir bereit, uns über SSH mit dem Pi zu verbinden. Jede Linux-Distribution wird mit einem SSH-Client ausgeliefert. Windows-Nutzer können auf das Tool *PuTTY* zurückgreifen. Um sich von einem Linux-Rechner aus zu verbinden, starten Sie ein Terminal und tippen folgenden Befehl ein:

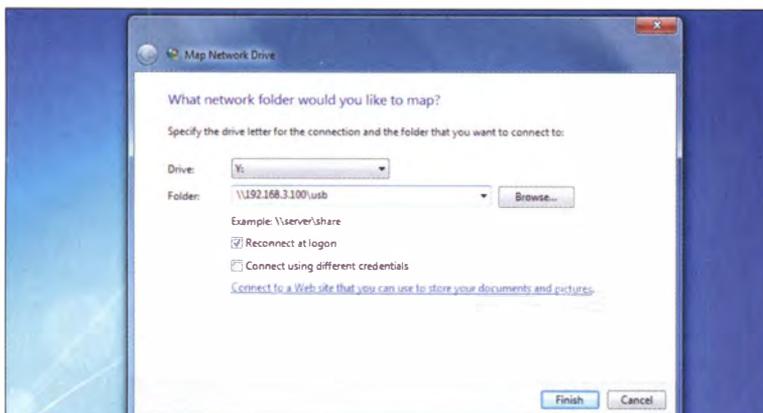
```
$ sudo ssh pi@192.168.3.100
```

Nachdem Sie zugestimmt haben, diese Adresse den vertrauenswürdigen Hosts hinzuzufügen, fragt Sie das Fenster nach den Login-Daten. Auf einer frischen, nicht konfigurierten Raspbian-Installation ist „raspberry“ das Passwort für den Standardnutzer „pi“. Von diesem Punkt an sind alle

Kommandos des Tutorials die gleichen, egal, ob Sie über einen Linux- oder Windows-Rechner mit Ihrem Raspberry Pi verbunden sind. Der Grund dafür ist, dass die Kommandos nicht lokal, sondern auf dem Pi ausgeführt werden. Da es sich um eine frische Installation handelt, werden Sie zu Beginn gefragt, ob Sie die Konfiguration durchführen wollen. Geben Sie folgenden Befehl ein:

```
$ sudo raspi-config
```

Das öffnet einen Bildschirm mit einer Vielzahl an Optionen. Scrollen Sie in der Liste bis ganz nach unten zum Punkt „Update“. Dieser lädt die aktuelle Version dieses Konfigurations-Werkzeugs herunter. Sobald das erledigt ist, wird der Raspberry Pi neu starten. Anschließend müssen Sie sich wie vorher mit dem **ssh**-Kommando oder per *PuTTY* neu



# Smartphone

verbinden. Das ist jedes Mal notwendig, wenn Sie irgendeine Einstellung verändern und der Pi neu starten muss.

Wählen Sie nun **expand\_rootfs**, sobald Sie sich wieder im Einstellungs Menü befinden, damit Raspbian den kompletten Platz auf der SD-Karte verwenden kann. Nun ist die Option **memory\_split** an der Reihe, die das RAM zwischen GPU und CPU aufteilt. Da wir nur über eine Remote-Verbindung auf den Raspberry Pi zugreifen, sollten Sie der GPU die kleinstmögliche Menge von 16 MB zuweisen.

Nun sollten Sie auf dem Pi weitere Benutzer hinzufügen. Später werden wir den Zugang zu einigen Verzeichnissen auf angeschlossenen USB-Geräten für bestimmte Nutzer wie auch für eine Gruppe von Nutzern beschränken, wobei es trotzdem noch öffentliche Bereiche auf dem Gerät gibt.

```
$ sudo adduser bodhi
```

Dieser Befehl wird den Nutzer und die entsprechenden Verzeichnisse anlegen. Der Befehl fragt nach dem Passwort des Nutzers und anderen Details. Fügen Sie den User nun mit folgendem Befehl einer Gruppe hinzu:

```
$ sudo usermod -a -G users bodhi
```

## Auf die Tanzfläche

Sobald Sie damit fertig sind, sollten Sie den Raspberry Pi für die Nutzer im Netzwerk zugänglich machen. Dazu verwenden

wir *Samba*, das es uns erlaubt, Dateien über das CIFS-Protokoll („Common Internet File System“) zugänglich zu machen. Verwenden Sie folgenden Befehl, um *Samba* auf dem Pi zu installieren:

```
$ sudo apt-get install samba samba-common-bin
```

Wenn das erledigt ist, müssen Sie in *Samba* noch die Nutzer hinzufügen. Beim Standardnutzer „pi“ verläuft das folgendermaßen:

```
$ sudo smbpasswd -a pi
```

Sie werden dabei sofort nach einem Passwort gefragt. Normalerweise ist es sicher, auch hier das gleiche Passwort wie beim Benutzerkonto zu verwenden. Wiederholen Sie diesen Schritt für jeden Nutzer auf dem System. *Samba* wird über eine Konfigurationsdatei kontrolliert, die Sie vor der Benutzung anpassen müssen. Dabei ist es immer eine gute Idee, vor Änderungen ein Backup der Datei anzulegen:

```
$ sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.old
```

Nun verwenden wir den Kommandozeilen-Texteditor *Nano*, um die Konfigurationsdatei anzupassen:

```
$ sudo nano /etc/samba/smb.conf
```

Suchen Sie in dieser Datei nach dem Abschnitt „Authentication“, und entfernen Sie die Raute vor der Zeile **#security = user**. So stellen Sie sicher, dass sich nur diejenigen über *Samba* verbinden können, die ein Home-Verzeichnis auf dem Raspberry Pi besitzen. Um den Nutzern Zugriff auf Ihre Home-Verzeichnisse zu gewähren, verändern Sie in der Konfigurationsdatei unter dem Abschnitt **[homes]** die beiden Optionen **browseable = yes** und **read only = no**. Drücken Sie STRG-X, um *Nano* zu beenden, speichern Sie dabei die Datei, indem Sie Enter drücken, sobald der Name der Datei angezeigt wird. Jedes Mal, wenn Sie irgendwelche Änderun-

## Zu einer statischen IP wechseln

Da wir externe *Samba*-Freigaben automatisch einbinden, müssen wir sicher sein, dass sie immer unter der gleichen Adresse erreichbar sein werden. Mit aktiviertem DHCP wird der Router aber eine andere freie IP-Adresse verteilen. Für unseren Raspberry Pi, der aktuell unter **192.168.3.100** zu finden ist, bedeutet das, dass er bei einem Neustart eine neue IP zugewiesen bekommen könnte. In diesem Fall schlägt das Einbinden der Freigaben fehl. Um sicherzustellen, dass der Pi die gleiche IP-Adresse erhält, müssen wir sie zu einer statischen IP-Adresse machen. Verschaffen Sie sich dazu erst Informationen über das Netzwerk, indem Sie den Befehl **ifconfig eth0** verwenden. Sie brauchen das Standard-Gateway sowie den Name-Server. In den meisten Fällen haben beide die gleiche Adresse, unter der Sie auch das Konfigurations-Interface des Routers finden. Die Netzmaske lautet meist **255.255.255.0**. Durchsuchen Sie die Konfigurationsseite des Routers, um den Bereich

der DHCP-Adressen herauszufinden. Sie möchten nämlich eine Adresse außerhalb dieses Bereiches, um zu verhindern, dass die Adresse einem anderen Gerät zugewiesen wird. Verbinden Sie sich nun über SSH mit dem Raspberry Pi, und ändern Sie die Datei **/etc/networks/interfaces**. Ersetzen Sie die Zeile **iface eth0 inet dhcp** durch

```
iface eth0 inet static
    Unter diese Zeile schreiben Sie die Details zur statischen IP-Adresse:
```

```
address 192.168.3.121
netmask 255.255.255.0
gateway 192.168.3.1
nameserver 192.168.3.1
```

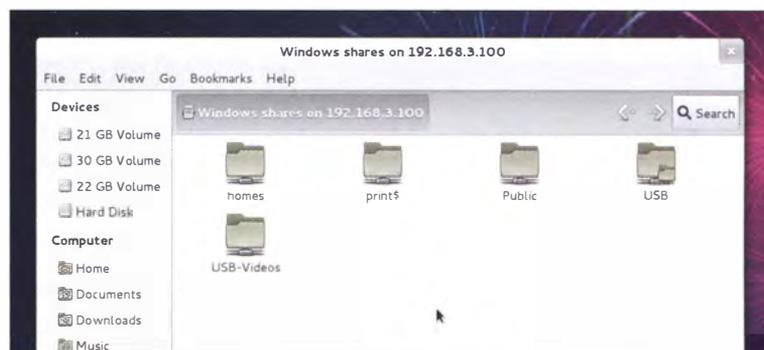
Das war's schon. Speichern Sie, und starten Sie den Raspberry Pi neu. Nach dem Neustart bestätigt die Eingabe von **ifconfig eth0**, dass die Verbindung die statische IP-Adresse hat, die Sie ihr zugewiesen haben. Verwenden Sie diese Adresse in allen Konfigurationsdateien.

gen an Diensten

vornehmen, egal welche, müssen Sie sie neu starten. Bei *Samba* geschieht das mit dem Befehl

```
$ sudo service samba restart
```

Nun ist es Zeit, die USB-Geräte in den Raspberry Pi einzustöpseln. Auch wenn der Raspberry Pi einige USB-Anschlüsse hat, ist es immer eine gute Idee, sämtliche USB-Hardware über einen USB-Hub mit eigener Stromversorgung zu verwenden. Das gilt vor allem für größere USB-Massenspeicher, die viel Energie verbrauchen. Achten Sie auch darauf, wie das Laufwerk formatiert ist. Standardmäßig ist das bei vielen USB-Flashspeichern das Dateisystem FAT32. Das mag im »



› Moderne Linux-Dateimanager wie Nautilus aus Gnome oder Dolphin von KDE können direkt auf Samba-Freigaben zugreifen und sie mounten.

## Einen Torrent erstellen

Der eine große Unterschied zwischen dem web- und dem desktopbasierten *Transmission* ist, dass ersteres keine grafischen Steuerelemente besitzt. Für diesen Zweck müssten Sie das kommandozeilenbasierte Tool **transmission-create** verwenden. Falls Sie einen Torrent anlegen möchten, brauchen Sie nicht nur die Dateien, die Sie verteilen möchten, sondern auch einen BitTorrent-Tracker. Dabei handelt es sich um einen Server, der die Nutzer und Knoten verwaltet, welche den Torrent anbieten. Für dieses Tutorial verwenden wir den beliebten Tracker, der unter

**http://linuxtracker.org:2710/announce** zu bekommen ist.

```
$ sudo transmission-create -o einlinux.torrent -t http://linuxtracker.org:2710/announce /pfad/zu/einerdistro.iso
```

Dieser Befehl erstellt einen Torrent mit dem Namen **einlinux.torrent**, der die Datei **einerdistro.iso** anbietet. Verweisen Sie beim Anlegen des Torrents statt auf die Datei auf ein Verzeichnis, um mehrere Dateien zu verteilen. Sobald Sie den Torrent erstellt haben, müssen Sie ihn nur an den *Transmission*-Client weitergeben, um die Verteilung der Inhalte mit anderen zu starten.

Hinblick auf die Kompatibilität über viele Betriebssysteme hinweg das beste Dateisystem sein, allerdings ist es eines der schlechtesten, wenn es um das Verteilen von Dateien über ein Netzwerk geht. Dann gibt es noch NTFS, das von vielen externen USB-Laufwerken verwendet wird. Das allerdings ist ebenfalls kein geeignetes Format, wenn Sie Dateien von einem entfernten Laufwerk streamen möchten.

Mit **sudo fdisk -l** finden Sie den Pfad des Laufwerks heraus, nachdem Sie es angesteckt haben. Der Befehl listet alle Speichergeräte auf sowie die Partitionen, die sich darauf befinden. Durchsuchen Sie die Ausgabe, und finden Sie das Laufwerk mit jener Größenangabe, die zu ihrem USB-Laufwerk passt. Die Bezeichnung des Laufwerks wird wahrscheinlich **sda** lauten, die der Partition **sda1**. Erstellen Sie nun einen Einhängpunkt und mounten sie das Laufwerk dort:

```
$ sudo mkdir /mnt/usb
$ sudo mount /dev/sda1 /mnt/usb
```

Das USB-Gerät bleibt eingehängt, bis Sie den Raspberry Pi neu booten. Um sich das ständige Remounten des Gerätes zu sparen, müssen Sie zuerst seine UUID herausfinden:

```
$ sudo blkid
/dev/sda1: LABEL="ntfs" UUID="3B5C053D35CAD865" TYPE="ntfs"
```

Fügen Sie diese zur Liste der beim Systemstart einzuhängenden Geräte hinzu:

```
$ sudo nano /etc/fstab
UUID=3B5C053D35CAD865 /mnt/usb ntfs defaults 0 0
```

*Samba* wurde darauf ausgelegt, diejenigen Dateien und Verzeichnisse zur Verfügung zu stellen, die in der Konfigurationsdatei festgelegt wurden. Wir gehen einmal davon aus, dass sich auf dem USB-Speicher einige Verzeichnisse befinden:

```
$ ls /mnt/usb
documents downloads music videos
```

Um den Ordner „downloads“ im Netzwerk zur Verfügung zu stellen, öffnen Sie die Datei **/etc/samba/smb.conf** mit *Nano*. Scrollen Sie zum Ende, und fügen Sie folgende Zeilen hinzu:

```
[Downloads]
comment = Place all your downloads here
Path = /mnt/usb/downloads
browseable = yes
writable = yes
read only = no
valid users = @users
```

Dieser Textblock stellt das Verzeichnis **/mnt/usb/downloads** allen Nutzern der Gruppe **users** zur Verfügung. Später werden wir dieses Verzeichnis auf Windows- und Linux-Rechnern im Netzwerk einbinden. Die Anwender sind dann in der Lage, ihre Downloadmanager so zu konfigurieren, dass Dateien von überall aus dem Netzwerk direkt in dieses Verzeichnis auf dem mit dem Pi verbundenen USB-Gerät gespeichert werden können. Sie haben aber auch die Möglichkeit, einige Ordner nur bestimmten Nutzern zugänglich zu machen:

```
[Documents]
comment = Important eyes-only PDF files
path = /mnt/usb/documents
browseable = no
writable = yes
read only = no
valid users = pi, bodhi
```

In diesem Beispiel können nur die Nutzer „pi“ und „bodhi“ die Inhalte des Ordners **documents** zugreifen.

## Den Torrent-Server aufsetzen

Torrents sind die bevorzugte Methode zum Weiterverbreiten von Open-Source-Inhalten. Die meisten Linux-Distributionen werden auf diese Weise verteilt, entweder über einen eigenen Tracker oder über **linuxtracker.org**. Unter Linux gibt es keinen Mangel an Torrent-Clients. Wir nutzen *Transmission* wegen seines einfach zu benutzenden Web-Interfaces. Wir installieren *Transmission* auf Raspbian, können ihn von überall aus aufrufen und Torrents hinzufügen, überwachen und steuern. Um *Transmission* zu installieren, geben sie über eine SSH-Verbindung zu Ihrem Raspberry Pi folgenden Befehl ein:

```
$ sudo apt-get install transmission-daemon
```

Das installiert und startet den *Transmission*-Dienst. Aber bevor Sie ihn zum Herunterladen von Torrents nutzen können, müssen Sie ihn noch einrichten. Stellen Sie sicher, dass *Transmission* nicht läuft, bevor Sie Änderungen an der Konfigurationsdatei vornehmen:

```
$ sudo service transmission-daemon stop
```

Fügen Sie, bevor Sie weitermachen, den User „transmission“, der bei der Installation automatisch angelegt wird, Ihrer eigenen Benutzergruppe hinzu:

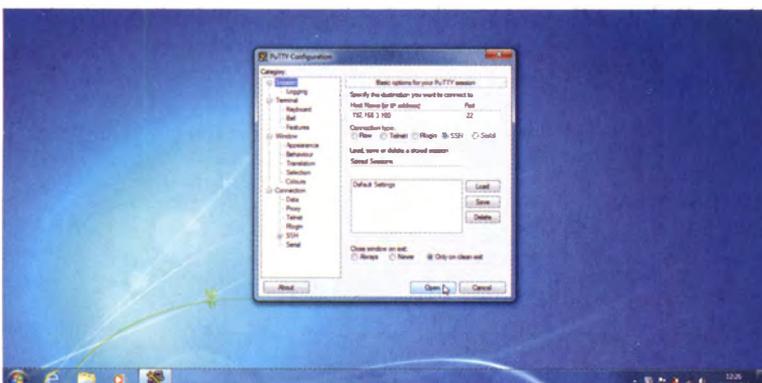
```
$ sudo mkdir /mnt/usb/public
$ sudo chown debian-transmission /mnt/usb/public
```

In der *Samba*-Konfigurationsdatei nehmen Sie eine weitere Freigabe mit auf:

```
[Public]
comment= Public share for torrents
browseable = yes
path = /mnt/usb/public
public = yes
writeable = yes
guest ok = yes
```

Starten Sie *Samba* neu, um die Freigabe allen zugänglich

► Unter Windows verwenden Sie **PuTTY**, um sich per SSH mit dem Raspberry Pi zu verbinden. Geben Sie dazu die IP-Adresse des Pi ein und klicken sie auf „Open“.



zu machen. Nun ist es Zeit, den Dienst für *Transmission* zu konfigurieren. Die Einstellungen sind in der Datei `/etc/transmission-daemon/settings.json` festgelegt. Öffnen Sie die Datei in *Nano*, und ändern sie zu Beginn folgendes

```
„rpc-whitelist-enabled“: true
```

zu

```
„rpc-whitelist-enabled“: false
```

damit man sich von allen Computern aus verbinden kann.

Legen Sie nun das Downloadverzeichnis mit folgendem Befehl fest:

```
„download-dir“: „/mnt/usb/public/downloads/Complete“
```

Sie können für unvollständig heruntergeladene Dateien auch einen anderen Ordner angeben. Ändern Sie dazu

```
„incomplete-dir-enabled“: false
```

zu

```
„incomplete-dir-enabled“: true
```

und geben Sie dann in der folgenden Zeile das Verzeichnis an, wo unvollständige Downloads gespeichert werden sollen:

```
„incomplete-dir“: „/mnt/usb/public/downloads/Incomplete“
```

## Benutzerauthentifizierung

Da wir den Ordner `/mnt/usb/public` dem Besitz des Benutzers „transmission“ übergeben haben, wird dieser automatisch alle neuen Verzeichnisse darin anlegen. Während die heruntergeladenen Torrents öffentlich sind, können Sie festlegen, dass nicht jedermann neue Torrents zum Download einreihen kann. Die Authentifizierung des Benutzers ist ein Weg, das zu erreichen. Vor der Anmeldung hat der Nutzer keinen Zugriff auf *Transmission*. Für diese Einstellung ändern Sie zunächst

```
„rpc-authentication-required“: false
```

zu

```
„rpc-authentication-required“: true
```

Dann geben Sie unter

```
„rpc-password“: „<password>“
```

ein Passwort an, das *Transmission* automatisch verschlüsselt. Speichern Sie nun die Datei, und starten Sie den Dienst mit `sudo service transmission-daemon start` neu.

Standardmäßig läuft *Transmission* über den Port 9091. In unserem Beispiel lautet die komplette URL für das Web-Interface für *Transmission* **192.168.3.100:9091**. Rufen Sie diese Adresse mit einem Browser auf, erscheint eine Passwortabfrage. Der Benutzername ist „transmission“, das Passwort ist jenes, das Sie vorhin in der Konfigurationsdatei angegeben haben.

Klicken Sie im Web-Interface von *Transmission* auf den „Open Torrent“-Button. Fügen Sie im auftauchenden Fenster die Adresse zur Torrent-Datei ein, und klicken Sie auf „Upload“, um den Download zu starten. Die Benutzeroberfläche ist sehr einfach zu verstehen. In der Standardeinstellung zeigt sie alle hinzugefügten Torrents an, Sie können aber auch die Drop-Down-Menüs verwenden, um die Torrents nach ihrem Status oder Tracker zu sortieren. Wenn Sie auf einen Torrent in der Liste rechtsklicken, wird ein Kontextmenü angezeigt.

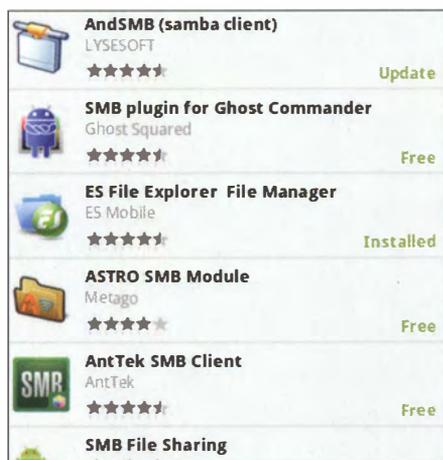
Sobald ein Torrent fertig heruntergeladen wurde, wird er aufgrund unserer Konfiguration automatisch in den öffentlich zugänglichen Ordner `/mnt/usb/public/downloads/Complete` verschoben. Die einfachste Weise, um auf eine Freigabe unter Linux zuzugreifen, ist die Eingabe der Adresse in den Dateimanager. Die meisten modernen Dateimanager, so wie *Nautilus* aus Gnome, unterstützen *Samba*. Starten Sie *Nautilus*, und drücken Sie die Tastenkombination STRG+L um die Adressleiste aufzurufen. Geben Sie nun `smb://` ein, gefolgt von der IP-Adresse des Raspberry Pi, auf dem *Samba* läuft. In unserem Fall wäre das `smb://192.168.3.100`. Um auf eine bestimmte Freigabe zuzugreifen, können Sie deren Namen an das Ende der Adresse anhängen, beispielsweise `smb://192.168.3.100/documents`. Oder Sie hängen die Freigabe mittels folgendem Befehl in Ihr Dateisystem ein:

```
# mount -t cifs -o username=pi,password=raspberryy //192.168.3.100/usb/downloads /mnt/downloads
```

Um diese Freigabe bei jedem Start automatisch einzuhängen, können Sie folgende Zeile der Datei `/etc/fstab` hinzufügen:

```
//192.168.3.100/usb/downloads /mnt/downloads cifs username=pi,password=raspberryy 0 0
```

## Mit Android auf Dateifreigaben zugreifen



### 1 Suchen und Installieren

Der Android Play-Store ist voll mit Dateimanagern, die mit *Samba*-Freigaben umgehen können. Wir verwenden den beliebten *ES File Explorer File Manager*.



### 2 Konfigurieren

Stellen Sie in der App die Ansicht von Local auf LAN, öffnen Sie Menü > New > Server, und geben Sie die Verbindungs- und Anmeldeinformationen ein.



### 3 Durchsuchen und streamen

Abhängig von Ihren Rechten können Sie Dateien herunter- und hochladen, sie streamen sowie auf öffentliche wie private Bereiche der Freigabe zugreifen.

# Hosting eines Foto-Servers

Nutzen Sie den Pi als Webserver für Fotoalben und zum Teilen von Bildern.

Es gibt viele kostenlose Online-Dienste, die es erlauben, Bilder mit anderen Menschen zu teilen. Diese Websites wurden jedoch meist für das Teilen von Bildern in sozialen Netzwerken entworfen. Auch wenn manche von ihnen fein justierbare Privatsphäreneinstellungen bieten, der ideale Ort, um persönliche Fotos mit Familie und engen Freunden zu teilen, sind sie nicht. Die beste Möglichkeit, private Fotos bei voller Kontrolle zu speichern und zu teilen, ist, sie selbst zu hosten. Dies erfordert etwas Aufwand, aber so haben Sie ihre Bilder stets im Griff und genießen nach geglückter Einrichtung nahezu dieselben Annehmlichkeiten, wie sie die großen Online-Fotoplattformen bieten. Um Web-Albumsoftware zu verwenden, benötigen Sie einen Server, und der Raspberry Pi schlägt sich als solcher recht wacker. Sogar der beliebte Apache-Webserver läuft auf dem Pi. Wir verwenden hier jedoch den schlanken Webserver Nginx, um die limitierten Ressourcen des Mini-Rechners bestmöglich auszunutzen. Die Web-Anwendung Lychee dient uns dabei zum Hosten, Verwalten und Teilen unserer Digitalfotos. Verglichen mit Software ähnlichen Funktionsumfangs ist Lychee schlank, einfach einzurichten und mit einer intuitiven Oberfläche ausgestattet. Beginnen Sie, indem Sie Ihren Raspberry Pi einschalten und per `sudo apt-get update` aktualisieren. Sie können Lychee per Remoteverbindung konfigurieren, indem Sie per SSH von einem Netzwerkrechner aus auf den Pi

zugreifen. Für diesen Artikel verwenden wir 192.168.2.100 als IP-Adresse des Raspberry Pi. So loggen Sie sich von einem Netzwerkrechner aus dort ein:

```
sudo ssh pi@192.168.2.100
```

Die Kommandozeile verlangt nun das Passwort für den Pi-Nutzer und loggt Sie ein. Sobald die Softwareliste aktualisiert wurde, installieren Sie die Programmiersprache PHP samt nötiger Programmbibliotheken:

```
sudo apt-get install php5-fpm php5-gd libgd2-xpm libpcrepp0 libxpm4
```

Dann installieren Sie die Webserver-Software:

```
sudo apt-get install nginx
```

Und schließlich installieren Sie den MySQL-Datenbank-server und verbinden ihn mit PHP:

```
sudo apt-get install mysql-server php5-mysql
```

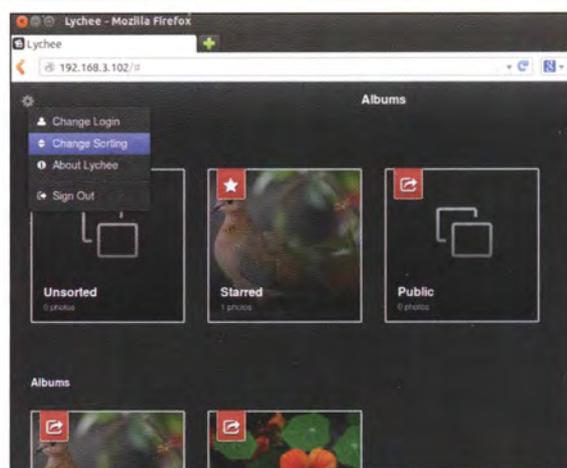
Während der Installation werden Sie aufgefordert, ein Passwort für den MySQL-Root-User zu bestimmen, das Sie später zum Erstellen der Datenbank für Lychee benötigen. Editieren wir nun die PHP-Konfigurationsdatei, um einige Parameter so anzupassen, dass wir das volle Potential von Lychee ausschöpfen können. Öffnen Sie dazu die Datei mit dem Texteditor Nano:

```
sudo nano /etc/php5/fpm/php.ini
```

Suchen Sie mit Strg+w nach diesen Variablen, deren Standardwerte Sie folgendermaßen anpassen:

## Erweitern Sie Lychee

Sie können Ihre Lychee-Installation so erweitern, dass sie automatisch mit Cloudspeichern wie Google Drive oder Dropbox synchronisiert wird. Da es für beide keine offiziellen Clients für den Raspberry Pi gibt, sind dazu eigene Skripte notwendig. Die Seite [robotic-controls.com/learn/raspberry-pi/syncing-files-raspberry-pi](http://robotic-controls.com/learn/raspberry-pi/syncing-files-raspberry-pi) listet einige Methoden zum Synchronisieren eines Verzeichnisses auf dem Pi mit der Cloud auf. Um die Fotos mit Google Drive abzugleichen, empfiehlt sich das Grive-Skript. Auf der Seite gibt es eine genaue Anleitung zum Kompilieren des Skripts aus dem Quellcode, da es nicht für Raspbian verfügbar ist. Führen Sie das Skript nach der Installation in Ihrem Bilderverzeichnis aus, um eine URL für den Bestätigungscode zu erhalten, den Sie für die Dateisynchronisierung brauchen. Ebenso können Sie das dropfuse-Skript verwenden, um Ihre Bilder mit einem Dropbox-Ordner abzugleichen. Das Skript teilt nicht Ihre gesamte Dropbox, sondern verbindet sich nur für den Datentransfer mit einem bestimmten Ordner.



› Lychee bietet wenige Optionen zur Konfiguration, aber es gibt eine Erweiterung, die per Kommandozeile alle Verzeichnisse nach Fotos durchsucht und als Alben hinzufügt.

```
max_execution_time = 200
post_max_size = 200M
upload_max_size = 200M
upload_max_filesize = 20M
max_file_uploads = 100
```

Falls eine Variable nicht existiert, fügen Sie sie hinzu. Scrollen Sie nach Anpassen der Werte ans Dateiende und tragen Sie folgende Zeilen ein, um die nötigen Erweiterungen zu aktivieren:

```
extension = php_mbstring.dll
extension = php_exif.dll
extension = php_gd2.dll
```

Speichern Sie die Datei mit Strg+x]und starten Sie den PHP-Dienst neu:

```
sudo service php5-fpm restart
```

Sobald die Infrastruktur eingerichtet ist, erstellen Sie mit **sudo mkdir /var/www** ein Datei-Verzeichnis für Lychee. Wechseln Sie in dieses Verzeichnis und installieren Sie mit **sudo apt-get install git** die Git-Software, mit der Sie die neueste Version von Lychee herunterladen können:

```
git clone https://github.com/electerious/Lychee.git
```

Dieser Befehl lädt die Dateien in den Ordner /var/www/Lychee herunter. Übergeben Sie nach dem Download die Eigentumsrechte des Ordners an den Benutzer des Webservers:

```
sudo chown -R www-data:www-data /var/www/Lychee
```

Geben Sie außerdem den Upload- und PHP-Ordner und die enthaltenen Dateien mit **chmod -R 777 uploads/php/** für den uneingeschränkten Zugriff frei. Die Software ist nun an Ort und Stelle, lediglich Nginx muss noch darüber informiert werden. Editieren Sie dazu dessen Konfigurationsdatei mit

```
sudo nano /etc/nginx/sites-available/default
```

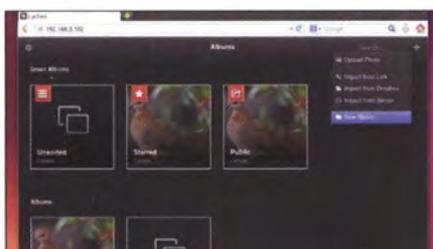
und fügen Sie folgende Zeilen als Verweis auf Ihre

Lychee-Installation hinzu:

```
Server {
    root /var/www/Lychee;
    index index.php index.html index.htm;
    location ~ \.php$ {
        fastcgi_pass unix:/var/run/PHP5-fpm.sock;
        fastcgi_index index.php;
        include fastcgi_params;
    }
}
```

Starten Sie den Webserver mit **sudo service nginx restart** neu, um die Änderungen zu aktivieren. Öffnen Sie nun in einem Browser das Lychee-Installationsverzeichnis auf Ihrem Raspberry Pi, beispielsweise 192.168.2.100. Lychee beginnt nun die Erstkonfiguration der neu installierten Software. Im ersten Schritt werden die Verbindungs- und Passwortinformationen des MySQL-Datenbankservers abgefragt. Verwenden Sie localhost als Hostname und root als Benutzer. Tragen Sie Ihr bei der Installation festgelegtes Passwort für den MySQL-Server in das entsprechende Feld ein. Sie können Lychee mit einer bereits vorhandenen Datenbank verwenden, es empfiehlt sich aber, das Programm seine eigene Datenbank anlegen zu lassen. Klicken Sie auf „Connect“, um zum nächsten Installationsschritt zu kommen, in dem Sie Benutzername und Passwort für diese Lychee-Installation festlegen. Klicken Sie anschließend auf „Create Login“, um den Vorgang abzuschließen – schon sind Sie startklar. Lychee verbindet sich mit dem MySQL-Server auf dem Raspberry Pi, erstellt seine eigene Datenbank und richtet Ihren Zugang ein, mit dem Sie Fotos hochladen und verwalten können. Danach loggt Lychee Sie automatisch ein und bringt Sie zum Startbildschirm. Jetzt können Sie Alben einrichten, Fotos hochladen und teilen und Ihre Lychee-Installation konfigurieren.

## Übertragen von Bildern in Lychee



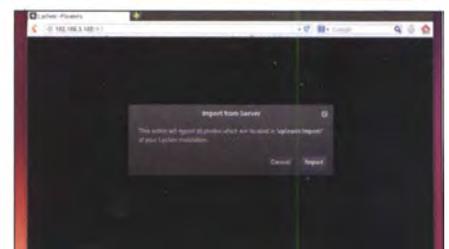
### 1 Album erstellen

Klicken Sie auf das Symbol „+“ rechts oben auf der Web-Oberfläche von Lychee, um Ihre Bilder hochzuladen. Wenn Ihre Fotos noch nicht in Kategorien sortiert sind, können Sie die Option „Upload Photo“ verwenden, um Bilder zu importieren und in Alben zu sortieren. Alle Fotos, die nicht in ein Album hochgeladen werden, landen im System-Album „Unsorted“.



### 2 Von Festplatte laden

Nutzen Sie die Funktion „New Album“, um ein neues Album zu erstellen. Die Option „Upload Photo“ startet den Dateimanager, in dem Sie die Bilder auswählen können, die Sie hochladen möchten. Sie können entweder einzelne Fotos auswählen, oder mit den Tasten Shift + Strg mehrere selektieren. Lychee lädt sie dann auf den Server und registriert sie.

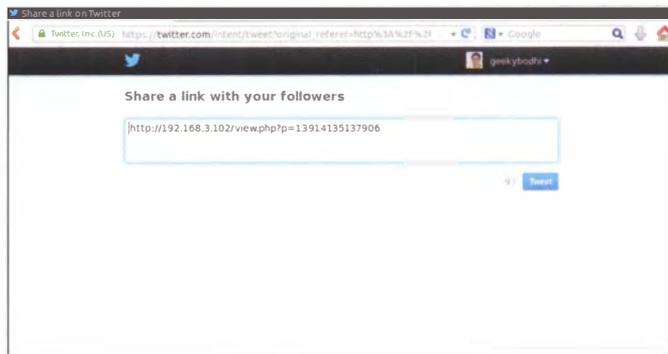
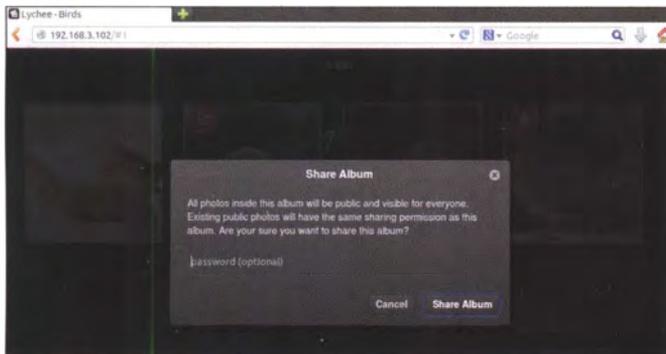


### 3 Importieren

Lychee importiert Bilder nicht nur von der Festplatte, sondern auch aus anderen Quellen. Das Symbol „+“ bietet an, Fotos direkt von einer URL oder Dropbox zu importieren. Zudem können Sie jeden FTP-Client verwenden, um Bilder in Lychees Verzeichnis uploads/import zu laden und die Option „Import from Server“ nutzen, um diese Fotos zu importieren.



## Bilder für Freunde und Familie freigeben

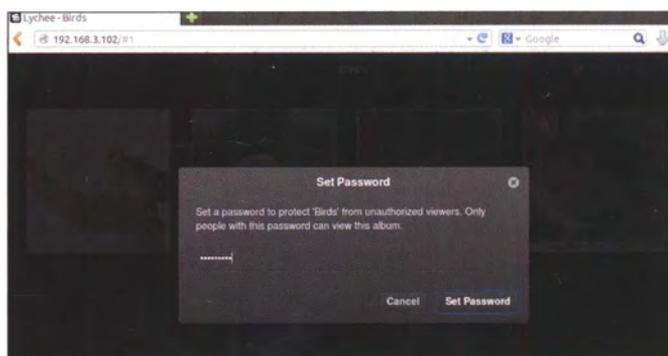
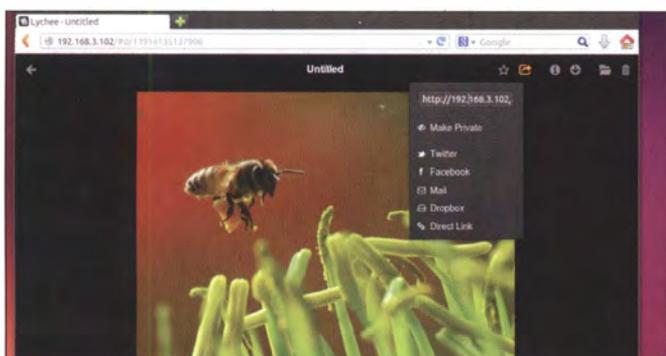


### 1 Album teilen

In der Album-Ansicht können Sie das „Teilen“-Symbol rechts oben klicken, um einen relativ langen Link zu diesem Album zu erhalten, den Sie dann weitergeben können. Vorläufig haben darauf jedoch nur Nutzer innerhalb Ihres Netzwerks Zugriff. Wenn Sie also den Link mit Personen außerhalb Ihres Netzwerks teilen möchten, müssen Sie die Firewall Ihres Routers entsprechend konfigurieren.

### 2 Vereinfachtes Teilen

Um das Freigeben der URL zu erleichtern, können Sie sie direkt aus Lychee auf Twitter oder Facebook teilen. Das „Teilen“-Symbol gestattet es, die URL als Tweet oder Status auf Facebook zu veröffentlichen. Sie können zudem mit der „Mail“-Option aus Lychee Ihr Standard-Mailprogramm aufrufen und dort automatisch eine neue E-Mail mit dem Link zum jeweiligen Album verfassen.

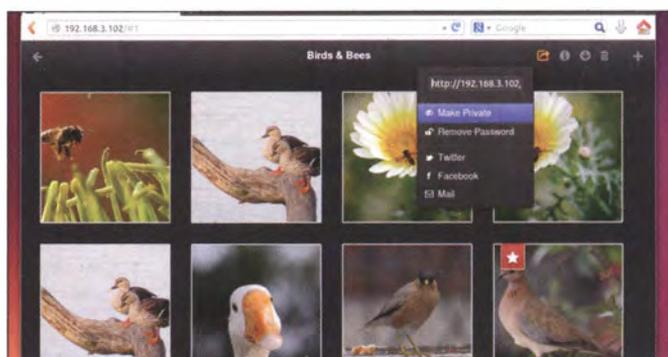
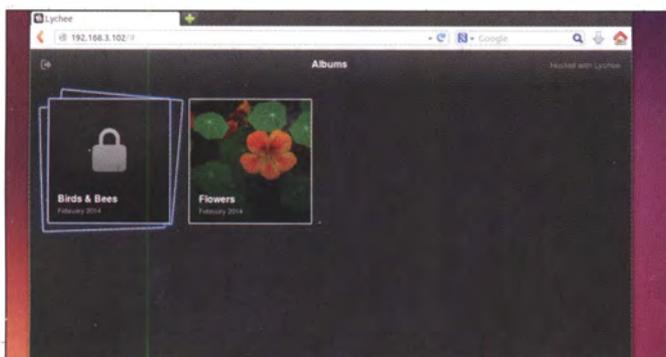


### 3 Foto teilen

Analog zu ganzen Alben können Sie auch einzelne Fotos in der Detailansicht mit dem „Teilen“-Symbol freigeben. Sie haben dabei ebenfalls die Möglichkeit, den Link des Fotos per Twitter, Facebook oder per E-Mail zu teilen. Zusätzlich können Sie das Foto in Ihre Dropbox übertragen oder eine Ansicht des Direktlinks in einem neuen Tab öffnen.

### 4 Passwortschutz

Als zusätzliche Sicherheitsmaßnahme bietet Lychee beim Teilen eines Albums an, den Zugriff per Passwortschutz zu beschränken. Dies ist rein optional und kann ausgelassen werden. Sollten Sie besondere Zugangsbeschränkungen benötigen, gelangen Sie ebenfalls über das „Teilen“-Symbol zur Passwort-Option. Es können nur ganze Alben mit Passwort geschützt werden, keine einzelnen Bilder.



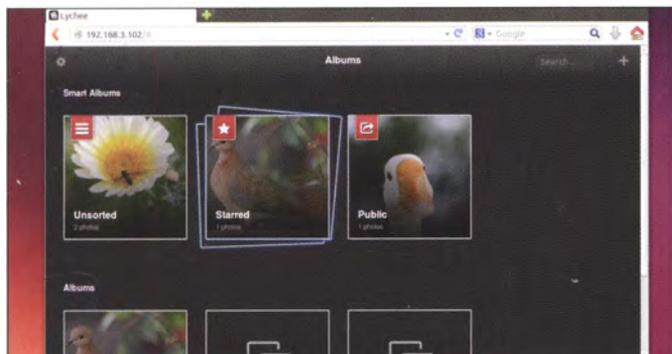
### 5 Öffentliches Album

Wenn Sie ein Foto teilen, versieht Lychee dessen Vorschaubild mit einem Symbol, das anzeigt, dass es öffentlich verfügbar ist. Die Software zeigt alle geteilten Fotos und Alben übersichtlich im System-Album „Public“ an. Ihre öffentlichen Alben sind für alle Netzwerknutzer auf der Startseite Ihres Lychee-Servers – in unserem Fall 192.168.2.100 – sichtbar.

### 6 Zugriff beschränken

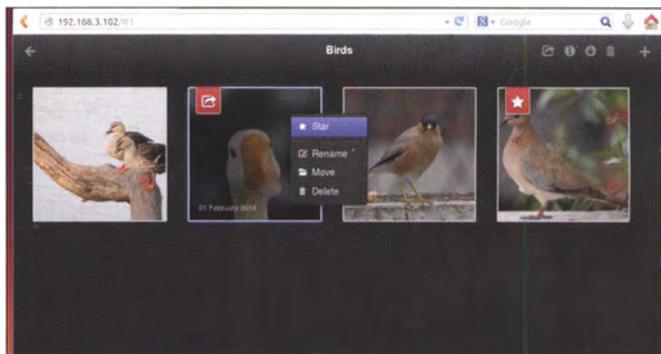
Die Option „Make Private“ erlaubt es, den Zugriff auf Alben und Fotos wieder zu beschränken. Wenn Sie ein Bild oder Album als privat markieren, zeigt Lychee es nicht mehr im System-Album „Public“ an. Einzelne Bilder eines öffentlichen Albums können so jedoch nicht unsichtbar gemacht werden, dazu müssen Sie das Album im Zugang beschränken.

## Alben sortieren und bearbeiten



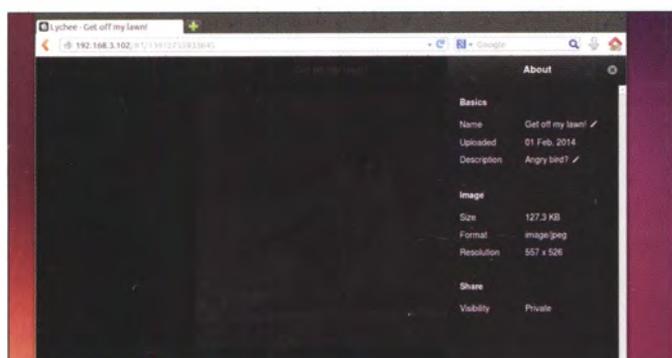
### 1 System-Alben

Um die Verwaltung zu erleichtern, hat Lychee einige virtuelle Verzeichnisse namens „Smart Albums“, die Bilder aus verschiedenen Ordnern zusammenfassen. Das System-Album „Unsorted“ enthält alle Bilder, die keinem Album zugeordnet sind. Das Album „Starred“ zeigt Fotos an, die Sie gesondert markiert haben, das Verzeichnis „Public“ gibt einen Überblick über alle veröffentlichten Bilder.



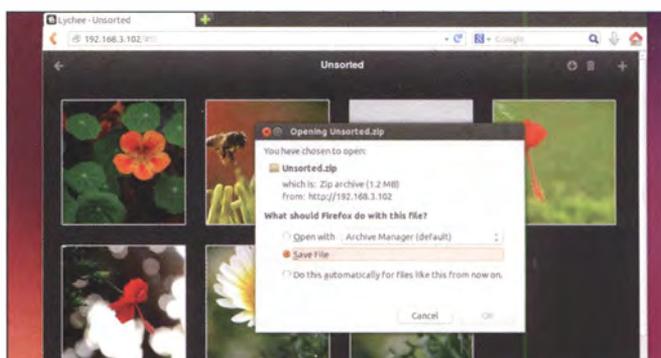
### 2 Bilder arrangieren

Nach dem Import von Bildern in Lychee haben Sie mehrere Möglichkeiten, sie zu sortieren. Oben rechts auf dem Startbildschirm gibt es eine Suchfunktion, mit der Sie Bilder finden können. In der Galerieansicht können Sie einzelne Bilder umbenennen, verschieben, löschen oder gesondert markieren.



### 3 EXIF-Daten anzeigen und editieren

Alle Fotos verfügen über Metadaten, auch EXIF-Daten genannt. Lychee importiert diese Datensätze zusammen mit den Bildern. Öffnen Sie ein Foto und klicken Sie auf das kleine „i“ rechts oben, um die EXIF-Daten des Bildes anzuzeigen. Außerdem können Sie einige der EXIF-Daten wie Name oder Beschreibung bearbeiten, wenn Sie auf das Bleistift-Symbol klicken.



### 4 Fotos übertragen

Per Klick auf das nach unten zeigende Pfeilsymbol können Sie einzelne Fotos in voller Auflösung oder ganze Alben als Zip-Archiv herunterladen. Mit einem FTP-Client lassen sich Bilder in Lychee einbinden, indem Sie sie direkt in den Ordner upload/import im Lychee-Installationsverzeichnis hochladen und anschließend über die Web-Oberfläche aus diesem Ordner importieren.

## Der Raspberry Pi als Bilderrahmen

Nachdem Sie Raspbian installiert und seine Softwarequellen mit **sudo apt-get update** aktualisiert haben, installieren Sie mit **sudo apt-get install qiv** den Quick Image Viewer. Erstellen Sie einen Ordner images/ im Verzeichnis /home/pi und übertragen Sie alle Bilder hierher. Erstellen Sie in einem Texteditor eine Datei namens photoframe mit der Zeile:

```
qiv --fullscreen --slide --random --delay=10 /home/pi/images/*
```

So starten Sie die 10-Sekunden-Zufallswiedergabe aller Bilder im Ordner im Vollbildmodus. Speichern Sie die Datei und machen Sie sie mit **chmod +x photoframe** zu einem ausführbaren Skript. Suchen Sie nun im Home-Ordner (/home/pi) die Datei .xinitrc oder legen Sie

sie gegebenenfalls an und fügen Sie folgende Zeile hinzu:

```
/home/pi/photoframe
```

Starten Sie mit **startx** den X-Server, der automatisch die Diashow ausführt. Bei einer großen Anzahl von Bildern empfiehlt sich ein USB-Laufwerk; es muss am richtigen Einhängepunkt gemountet sein, damit das Skript die Bilder findet. Verbinden Sie Ihr Laufwerk und suchen Sie in den letzten Zeilen von **dmesg** seinen Ort. Mit **ls-l /dev/disk/by-uuid/** erfahren Sie die einzigartige UUID des Laufwerks, mit **sudo mkdir /media/usb-images** legen Sie einen Einhängepunkt an. Fügen Sie für automatisches Einhängen des Laufwerks Folgendes zu **/etc/fstab** hinzu: **UUID=E352-47DC /media/usb-images vfat default 0 2**.

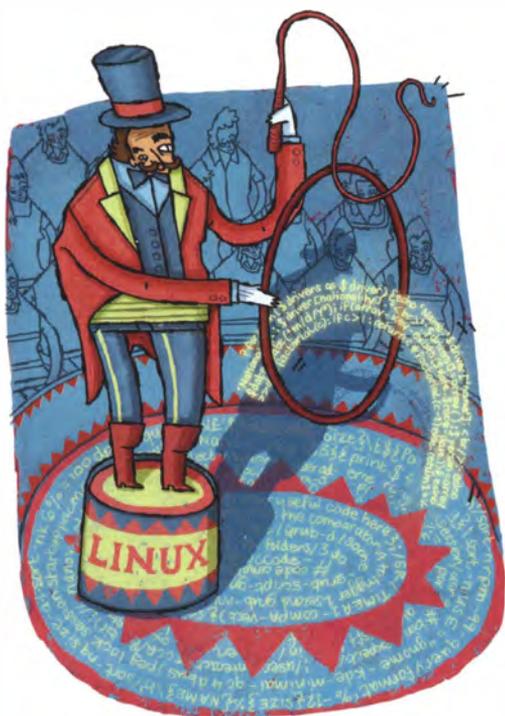
So wird Ihr USB-Laufwerk immer in /media/usb-images eingehängt und Sie können im photoframe-Skript auf das Verzeichnis verweisen.



➤ Nutzen Sie den Pi, um Musik zu streamen, Videos abzuspielen und vieles mehr.

# Elektronische Infotafeln

Sie haben etwas mitzuteilen? Dann tun Sie das mit Stil.



Eine sehr praktische Möglichkeit, einem Publikum etwas mitzuteilen, sind elektronische Infotafeln – also flache Bildschirme ganz unterschiedlicher Größe, die Informationsinhalte anzeigen. Ob Sie eine Menütafel an den Eingang Ihres Restaurants hängen, ein schwarzes Brett an Ihrer Schule einrichten oder Passanten auf das Schaufenster Ihres Ladens aufmerksam machen möchten – eine digitale Anzeigetafel ist in jedem dieser Fälle eine großartige Option.

Elektronische Infotafeln haben viele Vorzüge. Sie können automatisch „umblättern“, also abwechselnd verschiedene Tafeln zeigen, sie können in Echtzeit aktualisiert werden, bei Änderungen entstehen keine Kosten für Material, Druck und Anbringung, und sie sind ein Blickfang. Da Flachbildmonitore und andere geeignete Displays immer billiger werden, ist auch der Gerätepreis keine allzu hohe Hürde mehr.

Concerto ([concerto-signage.org](http://concerto-signage.org)) ist eine Open-Source-Software, mit der man über eine webbasierte Benutzeroberfläche Inhalte verwalten und an verschiedene Displays senden kann. Das Programm kann beispielsweise Bilder, Texte, Videos und RSS-Feeds darstellen. Es ist sogar möglich, auf jedem eingebundenen Display etwas anderes anzeigen zu lassen. Grundsätzlich ist jede Gerätekombination, die mit einem Webbrowser und einem Display ausgestattet ist, für die Anzeige von Inhalten via Concerto geeignet. Sie können also Tablets oder Smartphones als digitale Anzeigetafeln verwenden und auch und ganz besonders natürlich eine Kombi aus Raspberry Pi und Flachbildmonitor. Da der Pi so klein ist, lässt er sich beispielsweise gut hinter dem Monitor oder in dessen unmittelbarer Nähe verbergen.

Um Concerto verwenden zu können, muss außerdem ein Webserver mit Datenbank vorhanden sein, der Ruby-on-Rails-Anwendungen bereitstellen kann.

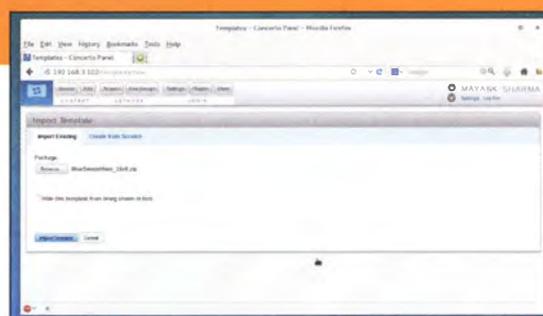
## IP-Adresse zuweisen

Bevor Sie Concerto installieren, müssen Sie dem Server eine statische IP-Adresse zuweisen. Am einfachsten ist dies über die Administrationsseite Ihres Routers zu bewerkstelligen. Die meisten Router erkennen die Ubuntu-Installation anhand der MAC-Adresse der zugehörigen Ethernet- oder WLAN-Karte und gestatten es, dafür eine statische IP-Adresse

## Template designen

Mithilfe von Templates werden die Inhalte auf dem Bildschirm angeordnet. Im Concerto-Bundle ist bereits ein anpassbares Standardtemplate enthalten, aber auf der Website des Programms sind noch weitere Templates (in Form von Zip-Dateien) verfügbar. Jedes davon enthält eine Hintergrundgrafik und eine XML-Deskriptor-Datei, einige auch ein CSS-Stylesheet. Um ein Template

anzupassen oder selber eines zu designen, benötigen Sie Grundkenntnisse in der Webseiten-Gestaltungssprache CSS. Die Templates sind so aufgebaut, dass der äußerste Container wiederum einen Full-Screen-Container enthält, in dem sich die Hintergrundgrafik befindet. Die verschiedenen Inhaltselemente (darunter Ticker, Text, Grafik) werden mit absoluten Positionangaben versehen.



› Die Templates erst hochladen, dann entpacken!

festzulegen. Die konkrete Durchführungsweise variiert von Router zu Router. Nehmen wir an, unser Server hätte die IP-Adresse 192.168.1.100. Nachdem Sie diese zugeteilt haben, wird es Zeit, Concerto zu installieren. Wenn Ihr Server bereits fertig eingerichtet ist, können Sie die Concerto-Pakete mithilfe eines Skripts herunterladen. Bei einer Distribution wie Ubuntu, die auf dem Paketverwaltungssystem APT basiert, können Sie aber auch ganz bequem das Concerto-Bundle verwenden – dieses installiert die neueste stabile Programmversion und nimmt die Einstellungen für alle benötigten Komponenten vor, einschließlich des Webservers und Datenbankservers.

## Installation und Einrichtung

Um loszulegen, müssen Sie zunächst den Repository-Schlüssel für Concerto importieren. Geben Sie Folgendes in die Kommandozeile ein:

```
wget -O - http://dl.concerto-signage.org/concerto_deb_public.key | sudo apt-key add -
```

Anschließend muss das Concerto-Depot zur Ubuntu-Liste der Repositories hinzugefügt werden, welches Sie in der Datei `/etc/apt/sources.list` finden. Entweder Sie öffnen die Datei in einem Texteditor und fügen das Repository manuell hinzu, oder Sie verwenden das folgende Kommando:

```
echo "deb http://dl.concerto-signage.org/packages/ raring main" | sudo tee -a /etc/apt/sources.list
```

Aktualisieren Sie nun die Repositories mittels **sudo apt-get update**. Bevor Sie das Concerto-Paket installieren, stellen Sie sicher, dass das Git-Paket ebenfalls da ist – das geht mit dem Befehl **sudo apt-get install git**. Dann kommt Concerto an die Reihe: Führen Sie **sudo apt-get install concerto-full** aus, um alle benötigten Abhängigkeiten mitzuinstallieren. Sollten Sie eine PostgreSQL-Fehlermeldung erhalten, installieren Sie das Paket `libpq-dev` mit dem Kommando

**sudo apt-get install libpq-dev** und versuchen Sie es erneut.

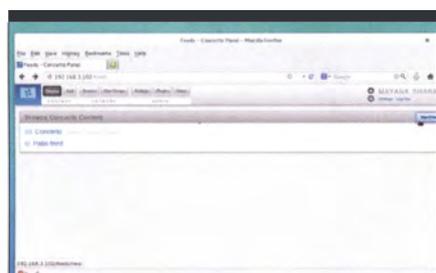
Wenn alle Pakete heruntergeladen und installiert sind, nimmt Concerto die Einstellungen für alle Komponenten vor. Während dieses Prozesses verlangt die Software die Vergabe eines Passworts für den MySQL-Root-Benutzer, dann verknüpft sich das Programm mit der Datenbank.

Am Ende der Installation legt Concerto eine Konfigurationsdatei für den virtuellen Host im Verzeichnis `/etc/apache2/sites-available/concerto` ab. Diese verweist auf das Verzeichnis, in dem sich der Concerto-Programmcode befindet. Es ist standardmäßig mit `/usr/share/concerto` benannt. Bevor Sie Concerto verwenden, deaktivieren Sie noch die Standardkonfiguration des virtuellen Hosts mit dem Kommando **sudo a2dissite 000-default** und aktivieren Sie danach den virtuellen Host mit **sudo a2ensite concerto**. Starten Sie mit **sudo apache2ctl restart** den Apache-Webserver neu, um die Konfiguration wirksam werden zu lassen.

Starten Sie einen Browser auf einem der Raspberry Pis oder sonstigen Computer im verwendeten Netzwerk und geben Sie in dessen Adresszeile die IP-Adresse der Concerto-Installation an, in unserem Beispiel also 192.168.1.100. Da es sich um eine neue Installation handelt, werden Sie zu einem Wizard geleitet, wo Sie ein Administratorkonto einrichten müssen. Weitere Benutzer können Sie anschließend innerhalb des Admin-Panels von Concerto erstellen.

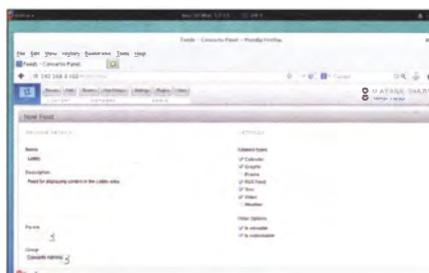
Das war's, der Concerto-Server ist jetzt bereit, Inhalte an die jeweiligen Monitore zu versenden. Die Inhalte verwalten Sie über das Dashboard, welches Sie nach dem Login vorfinden. Sie können hier zum Beispiel Feeds einrichten, Bilder, Videos und Texte hochladen und unterschiedliche Screens generieren, basierend auf Templates oder frei. Wie das im Einzelnen funktioniert, verraten wir Ihnen in den Tutorials auf diesen Seiten.

# Mit Concerto einen Feed generieren



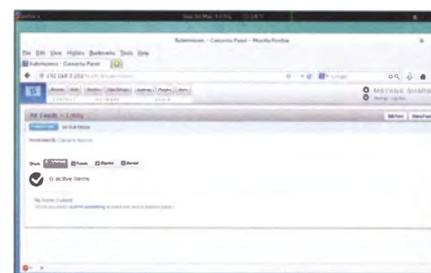
### 1 Neuer Feed

Stellen Sie sich einen Feed wie einen Behälter vor, in denen Benutzer Inhalte ablegen können. Jeder Feed wird von Moderatoren gepflegt, die einzelne Nachrichten zurückweisen oder für den Feed annehmen können. Um einen Feed neu anzulegen, gehen Sie im Admin-Panel auf Contents > Browse > New feed.



### 2 Details hinzufügen

Auf der sich daraufhin öffnenden Seite geben Sie einen Namen und eine Kurzbeschreibung für den Feed an. Sie können ihn auch einer bestimmten Concerto-Nutzergruppe zuordnen. Legen Sie dann noch die erlaubten Inhaltselemente an. Concerto unterstützt derzeit sieben verschiedene Inhaltsarten.

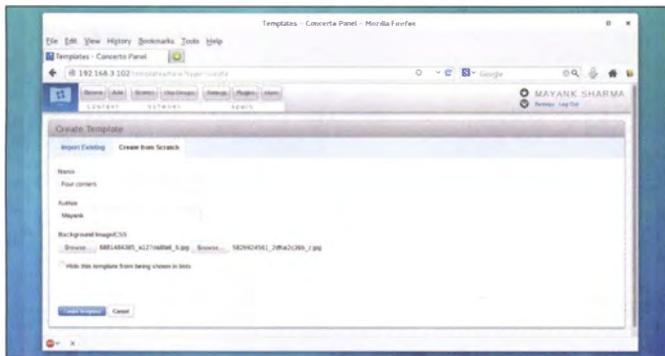


### 3 Feed-Übersicht

Zum Abschluss klicken Sie auf „Create feed“, dann sind alle Einstellungen gespeichert. Nun können Sie Inhalte für den Feed kreieren. Um sich die aktiven Inhalte der bestehenden Feeds anzusehen, klicken Sie „Browse“ und dann auf den Namen des jeweiligen Feeds. Sie können hier auch Feeds editieren oder löschen.

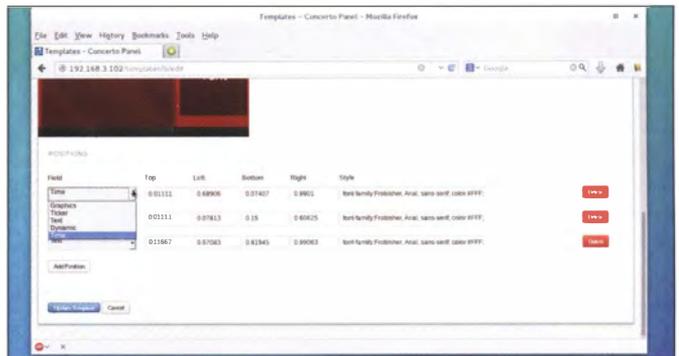


# Screens definieren



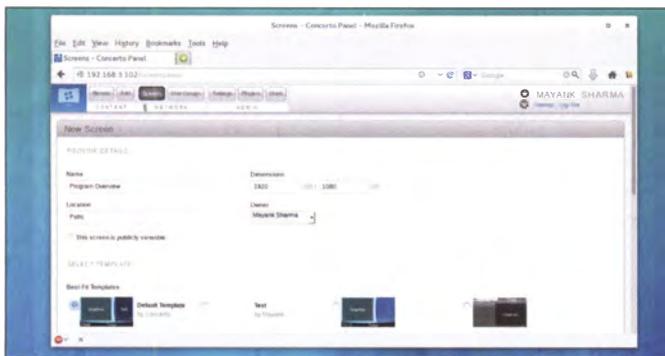
## 1 Template anlegen

Ein Screen (also eine einzelne Tafel) ist ein auf einem Template beruhendes virtuelles Arrangement von Inhalten. Um ein neues Template anzulegen, gehen Sie zu Screens > Common Templates > View all > New template > Create from scratch. Geben Sie einen Namen und eine Verfasserinformation an, und wählen Sie eine Grafik oder eine CSS-Datei für den Hintergrund aus.



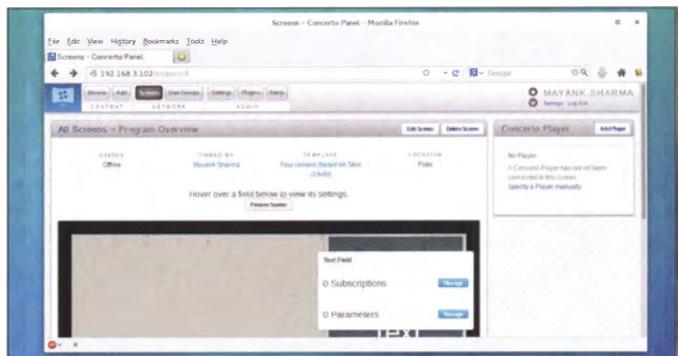
## 2 Position festlegen

Ist das Template erstellt, können Sie Elemente hinzufügen und deren Anzeigeposition definieren. Scrollen Sie auf der Seite nach unten zur Abteilung „Positions“ und klicken Sie auf „Add position“. Aus der Drop-down-Liste unter „Field“ können Sie den Inhaltstyp wählen, dessen Position Sie festlegen möchten. In den vier Spalten rechts von der Liste können Sie die Lage von oben links (0,0) bis unten rechts (1,1) festlegen.



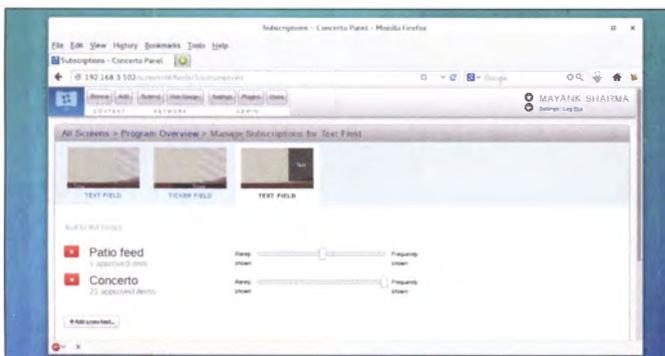
## 3 Screen anlegen

Als Nächstes kann (über Screens > New screen) der neue Anzeigebildschirm definiert werden. Auf der sich öffnenden Seite können Sie Informationen wie den Namen und den Anzeigort des Screens eingeben, ebenso dessen Abmessungen in Pixeln sowie den Besitzer des Screens. Wählen Sie danach eines der verfügbaren Templates für den Screen aus und klicken Sie auf „Create screen“.



## 4 Vorschau

Nach dem Anlegen des Screens zeigt Concerto Ihnen eine Übersicht des Arrangements gemäß verwendetem Template an. In der Abteilung „Program overview“ haben Sie Zugriff auf die Einstellungen für den Screen wie etwa Anzeigeposition oder aktueller Status. Mit „Edit screen“ kommen Sie zurück zum Eingabemenü und können Dinge verändern. Wenn Sie den Screen löschen möchten, wählen Sie „Delete screen“.



## 5 Feeds hinzufügen

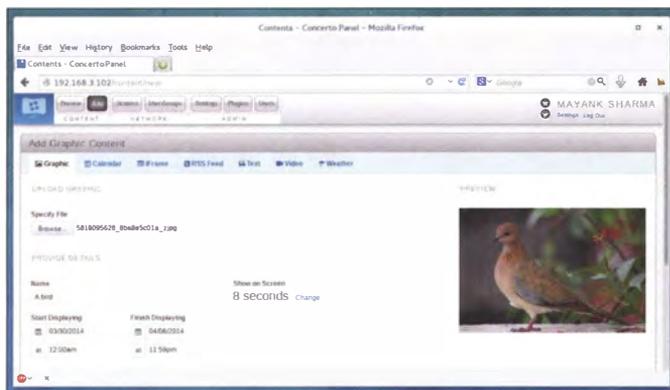
Um dem Screen Inhalt zu geben, ordnen Sie den Feldern Feeds zu. Wenn Sie die Maus auf eins der Felder in der Abteilung „Program overview“ führen, werden dessen Einstellungen angezeigt. Sie können die Inhalte für jedes Feld des Screens einzeln verwalten. Nach einem Klick auf „Add a new feed“ können Sie einen oder mehrere Feeds aussuchen.



## 6 Screen-Vorschau

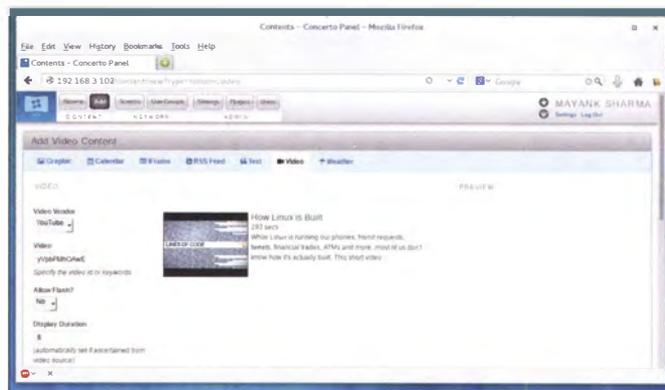
Um eine Vorschau einer Tafel zu erhalten, auf „Screens“ klicken und die Maus auf eine der Tafeln bewegen – dann wird deren URL (samt ID-Nummer am Ende) in der Statusleiste des Fensters angezeigt. Für die Vorschau gehen Sie zu [\[URL\]/frontend/\[ID\]](#), wobei Sie [\[URL\]](#) durch die Concerto-Serveradresse und [\[ID\]](#) durch besagte Nummer ersetzen.

# Inhalte hinzufügen



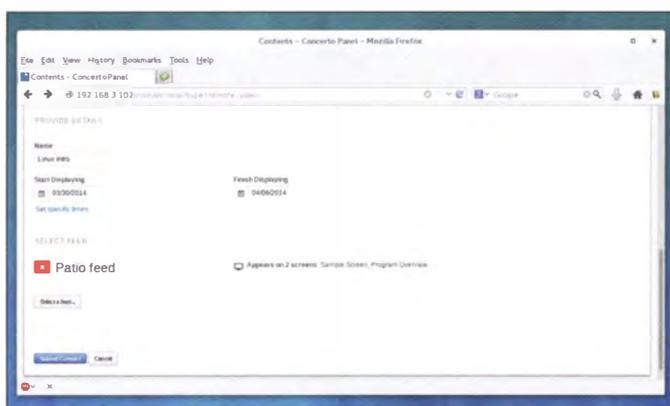
## 1 Inhaltstyp auswählen

Um Inhalte hochzuladen, klicken Sie zunächst auf den Button „Add“ in der „Contents“-Abteilung des Admin-Panels. Damit gelangen Sie zu einer Seite mit mehreren Tabs, aus denen Sie den gewünschten Inhaltstyp (wie Text, Grafik, Video) auswählen können.



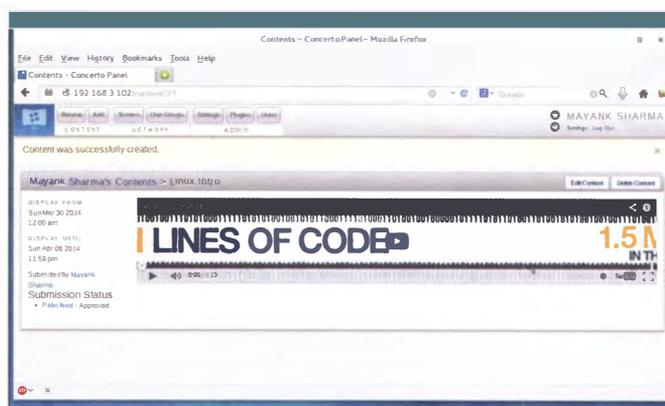
## 2 Inhalte hochladen

Je nach Inhaltstyp können Sie nun bestimmte Einstellungen vornehmen. Möchten Sie beispielsweise ein Video von YouTube oder Vimeo einbinden, geben Sie die ID des Videos an. Concerto zeigt Ihnen eine Vorschau der Inhalte an, nachdem Sie sie hochgeladen haben.



## 3 Inhalte einem Feed hinzufügen

Wenn Sie einen Inhaltstyp nach Wunsch angepasst haben, müssen Sie noch einen Namen dafür vergeben und Zeitpunkt und Dauer der Anzeige festlegen. Denken Sie daran, dass Sie alle Inhalte zuerst einem bestehenden Feed hinzufügen müssen, bevor Sie sie betrachten können.



## 4 Vorschau der Inhalte

Wenn bei Concerto Inhalte hinzugefügt worden sind, zeigt der Server Informationen darüber an, einschließlich des Eingabestatus. An dieser Stelle können Sie eine Vorschau des jeweiligen Inhalts bekommen und diesen bei Bedarf auch löschen.

# Der Raspberry Pi als Client-Rechner für die Infotafel

Der Raspberry Pi ist in der Lage, Inhalte vom Concerto-Server zu empfangen und an ein Anzeigergerät wie beispielsweise einen Flachbildmonitor weiterzuleiten, also in dieser Anordnung als Client-Rechner zu fungieren. Als Anzeigergerät eignet sich im Prinzip jeder Bildschirm, der sich an den HDMI-Video- oder den FBAS-Port des Raspberry Pi anschließen lässt. Die Serverdaten erhält der Pi über den Ethernet-Port oder einen USB-WLAN-Adapter. Sie müssen am Raspberry Pi nicht viel an Einstellungsarbeit vornehmen. Um den Mauszeiger auf dem Display permanent zu verbergen, installieren Sie Unclutter mit dem Befehl **sudo apt-get install unclutter**.

Deaktivieren Sie die Stromsparfunktionen, damit der Monitor sich nicht ausschaltet. Das geht, indem Sie mit **sudo nano /etc/lightdm/lightdm.conf** die Datei lightdm.conf in einem Texteditor öffnen, nach unten zum Bereich SetDefaults scrollen und die folgende Zeile unten anfügen:

**xserver-command=X -s 0 dpms.**

Stellen Sie den LXDE-Fenstermanager von Raspbian so ein, dass er beim Booten automatisch den Webbrowser Midori startet: Erzeugen Sie zu diesem Zweck mit **sudo mkdir -p ~/.config/lxsession/LXDE** ein Verzeichnis und darin eine Autostart-Datei mit **sudo nano ~/.config/lxsession/**

### LXDE/autostart.

Schreiben Sie Folgendes in diese Datei:  
**@midori -e Fullscreen -a http://192.168.1.100/frontend/1**

Ersetzen Sie dabei die IP-Adresse 192.168.1.100 durch die des Concerto-Servers und geben Sie an, welche Tafel (Screen) angezeigt werden soll.

Wenn Sie den Raspberry Pi anschließend rebooten, startet er automatisch den Midori-Browser und stellt darin den gewünschten Screen des Servers dar.

Um Midori zu schließen und zu Raspbian zurückzukehren, drücken Sie die Tasten Strg + F4.

# Der Pi als Alarmanlage

Oder als Tierfotoautomat. Sensoren machen es möglich!

**M**it dem Raspberry Pi kann man eindrucksvoll Prototypen jeder Größenordnung herstellen. Von blinkenden LEDs bis zum Quadrocopter – man kann fast alles mit einem Pi bauen. Sie brauchen nur etwas Fantasie und einige zusätzliche Bauteile. Eines der interessantesten Anwendungsgebiete des Minicomputers ist die sensorische Erfassung der Welt, denn diese ist voller Daten, die nur darauf warten, gesammelt und aufbereitet zu werden. So sind etwa Wetterstationen mit Sensoren ausgestattet, die Temperatur, Windgeschwindigkeit, Luftfeuchtigkeit und Luftdruck messen. Diese Daten können gespeichert und in Form von Grafiken und Tabellen dargestellt werden, die in andere Anwendungen oder Projekte importiert werden.

## PIR-Pläne

So ziemlich der preisgünstigste und einfachste Sensor ist der Passiv-Infrarot-Sensor (PIR). PIRs sind allgegenwärtig, zum Beispiel werden sie häufig in Bewegungsmeldern und Alarmanlagen eingesetzt. Sie sind mit Preisen ab 3 € so billig, dass man sie sogar in bewegungsgesteuerten Duftspendern findet. Ein PIR-Sensor funktioniert nach einem einfachen Prinzip: Ein Strahl infrarotes Licht wird in einen Raum gesendet. Wird der Strahl nicht unterbrochen – es bewegt sich also nichts im Raum –, reagiert der Sensor nicht. Bei einer Unterbrechung des Strahls wird jedoch sofort ein Signal an ein Gerät geschickt, das dann auf die gewünschte Weise reagiert. PIR-Sensoren sind nicht nur wegen der geringen Anschaffungskosten für Einsteigerprojekte geeignet, sondern auch, weil sie kaum Versuchsaufbauten und

Prototypanfertiigungen erfordern.

Bei unserem Projekt lassen wir einen PIR-Sensor Bewegungen melden. Eine Bewegung (eines unbefugten Eindringlings oder eines vorbeistreichenden Tieres) löst dabei eine programmierte Abfolge aus: Ein Bild der Szene und ein 10-Sekunden-Video in der Auflösung 640 x 480 Pixel werden aufgenommen. Dann werden wir per SMS\* über den Alarm informiert.

Um die Funktionsweise des Projekts besser zu erläutern, benutzen wir sogenannten Pseudocode. Damit lässt sich eine Programmierungssequenz in einfach verständlicher Sprache erklären.

So läuft also das Projekt in Pseudocode ab:

**PIR-Sensor sendet Lichtstrahl.**

**Falls der Strahl unterbrochen wird:**

**Sende ein Signal zum Raspberry Pi.**

**Bei Empfang des Signals nimmt der Raspberry Pi ein Foto auf.**

**Danach nimmt er ein 10 Sekunden langes Video auf.**

**Nachdem Foto und Video aufgenommen sind, versucht der Raspberry Pi, uns eine SMS mit den Informationen zum Vorgang zu schicken.**

**Ist die SMS abgeschickt, wartet der Raspberry Pi 30 Sekunden, dann startet die Sequenz von Neuem.**

Wofür kann solch ein Projekt eingesetzt werden?

Zunächst natürlich als häusliche Alarmanlage. Darüber hinaus könnte man aber beispielsweise auch automatische Tierfotos aus naher Distanz schießen. Der PIR dient dabei als Auslöser für die Aufnahme. Packen Sie die Gerätschaft

## Die Welt der Sensoren

Bei Eingabegeräten denkt man zunächst an eine Tastatur oder Maus. Aber mit Sensoren kann man auch andere Arten von Input für Projekte nutzen. Ultraschallsensoren wie der HC-SR04 messen mit lautlosen Schallwellen die Entfernung zwischen dem Sensor und einem Objekt. Der XLOBorg, der mit Beschleunigungsmesser und Magnetfeldstärkemessgerät ausgestattet ist, kann Bewegung, Ausrichtung und sogar Richtungsänderungen erfassen. Die Raspberry-Pi-Kamera und andere Webcams können mithilfe einer Bibliothek namens OpenCV ebenfalls als einfache Sensoren eingesetzt werden. Mit dieser Bibliothek kann Ihr Projekt seine Umgebung wahrnehmen, beispielsweise können Kameras Gesichter erkennen und Türen sich nur für Personen öffnen, deren Gesichtszüge erkannt werden. Mit dem Raspberry Pi betritt eine neue Hacker-Generation die Welt der Sensoren.





in eine wasserdichte Hülle, stellen Sie das Ganze an einem geeigneten Plätzchen in der Natur auf, und dokumentieren Sie auf diese Weise das dortige Tierleben.

In einem ähnlichen Projekt wurde das Fütterungsmuster von Vögeln überwacht. Ein beim Nistkasten angebrachter Sensor zeichnete auf, wann die Eltern kamen und gingen. Diese Daten wurden mit den Wetteraufzeichnungen aus der Umgebung verglichen. Es stellte sich heraus, dass die natürliche Nahrungsquelle von Vögeln – Insekten – bei regnerischem oder windigem Wetter nicht so ergiebig war, was zur Folge hatte, dass die Jungen seltener gefüttert wurden.

## Achtung! Achtung!

In diesem Projekt benachrichtigen wir den Nutzer per SMS. Es können aber auch andere Methoden der Nachrichtenübermittlung integriert werden. Mit dem Python-Modul `smtplib` versenden Sie eine E-Mail (eine Anleitung finden Sie unter [mkyong.com/python/how-do-send-email-in-python-via-smtplib/](http://mkyong.com/python/how-do-send-email-in-python-via-smtplib/)). Oder Sie twittern mit der Python-Bibliothek

Tweepy: Dazu müssen Sie zuerst mit [dev.twitter.com](http://dev.twitter.com) eine Anwendung erstellen. Dann importieren Sie den API-Schlüssel in Tweepy und senden und empfangen so Tweets. Mehr über Tweepy erfahren Sie auf der Website [www.tweepy.org](http://www.tweepy.org).

Die im Projekt aufgenommenen Bilder und Videos werden auf dem Raspberry Pi gespeichert, wo man sie später bequem abrufen kann.

Kurz zusammengefasst: Wir wollen mit dem Raspberry Pi eine Alarmanlage bauen, die durch einen bewegungsempfindlichen PIR-Sensor ausgelöst wird. Wird der Alarm ausgelöst, macht eine Kamera ein Foto und nimmt ein 10-sekündiges Video auf. Der Nutzer wird über die Auslösung des Alarms per SMS informiert.

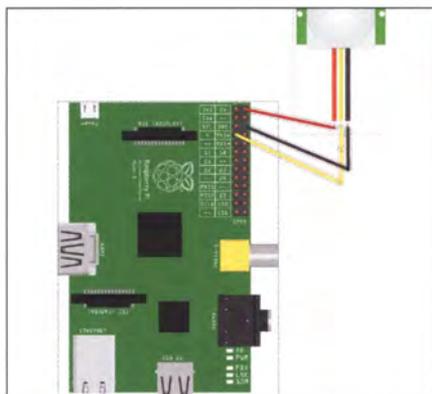
\*Das in diesem Artikel beschriebene SMS-Verfahren steht bislang nur für britische und belgische Handynummern zur Verfügung, doch der Anbieter hofft auf Kooperationen mit anderen Ländern. Statt eines SMS-Alarmes könnte man auch eine E-Mail-Benachrichtigung programmieren.

## Zusammenbau des Kits



### 1 Verbindung zum PIR

Der PIR-Sensor braucht drei Anschlüsse: VCC (5 V), OUT und Masse (GND). Der Sensor wird mit einem bei vielen Online-Shops erhältlichen Überbrückungskabel (female-female) an den Raspberry Pi angeschlossen. Stecken Sie das Kabel vorsichtig an den VCC-Pin am Sensor, bis es gut sitzt. Verbinden Sie dann das VCC-Kabel mit Pin 2 am Raspberry Pi. Wir verbinden den PIR-Sensor mit einem 5-V-Pin am Pi, dem Pin 2. Der Raspberry Pi muss immer ausgeschaltet sein, bevor Sie ein Gerät über GPIO an ihn anschließen. Überprüfen Sie alle Verbindungen, bevor Sie den Pi wieder anschalten.



### 2 Masse und OUT

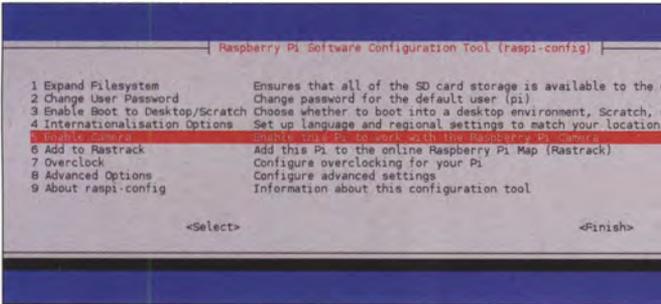
Wir schließen nun den GND-Pin des Sensors an die Masse des Raspberry Pi an, das ist Pin 6. Zum Schluss verbinden wir den OUT-Pin mit Pin 7. Inline-Widerstände werden für die Verbindung nicht benötigt, da eine Diode des PIR die Platine schützt. Das Signal, das über den OUT-Pin versendet wird, hat 3 V und ist unkritisch für den Pi.



### 3 Kamera anschließen

Nun schließen wir die Raspberry-Pi-Kamera an. Der Kunststoffport für die Kamera befindet sich zwischen dem HDMI-Anschluss und dem Ethernet-Port auf der Platine. Ziehen Sie die Plastiklasche des Ports sanft auf, und führen Sie das Flachbandkabel so ein, dass die silbernen Streifen am Ende des Kabels zum HDMI-Port zeigen. Wenn das Kabel richtig sitzt, drücken Sie die Plastiklasche wieder fest. So kann das Flachbandkabel sich nicht mehr aus dem Port lösen.

## Software-Setup



### 1 Kamera aktivieren

Die Hardware ist installiert, nun konfigurieren wir die Software, mit der wir die Hardware steuern. Zunächst aktivieren wir die Kamera. Tippen Sie ins Terminal:

```
sudo raspi-config
```

Wählen Sie im Menüpunkt „Kamera“ die Option „Aktivieren“. Drücken Sie die Eingabetaste. Sie werden gefragt, ob Sie die Kamera aktivieren möchten. Gehen Sie mit dem Cursor auf „Aktivieren“, und drücken Sie wieder die Eingabetaste. Dadurch kommen Sie zurück zum Hauptmenü. Gehen Sie auf „Beenden“ und drücken Sie die Eingabetaste.

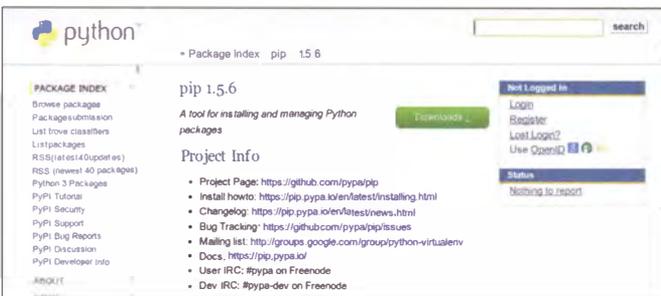


### 2 Kamera-Test

Der Raspberry Pi ist mit der Kamera verbunden, nun überprüfen wir, ob es auch ein Bild aufnehmen kann. Dazu geben Sie diesen Befehl im Terminal ein:

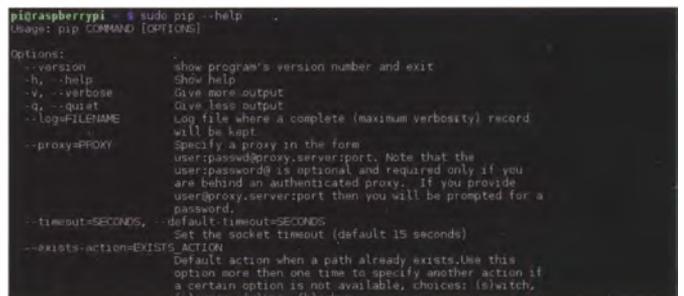
```
raspistill -o test.jpg
```

Der Befehl öffnet eine Anwendung, um mit der Kamera ein Bild zu schießen. Falls Fehler auftreten, überprüfen Sie die Syntax des Befehls und ob die Kamera wirklich im raspi-config-Menü aktiviert ist. Wenn das alles stimmt, überprüfen Sie die Kabelverbindung zwischen der Kamera und dem Raspberry Pi.



### 3 Installation von pip

Um die Kamera mit Python zu steuern, müssen wir die Bibliothek picamera herunterladen. Das geht am besten mit einem Paketmanager für Python namens pip – das funktioniert auf Ihrem Raspberry Pi genauso wie **apt-get**. Sie können damit Ihre Projekte immer aktualisieren. Wir installieren also zuerst pip über das Terminal.

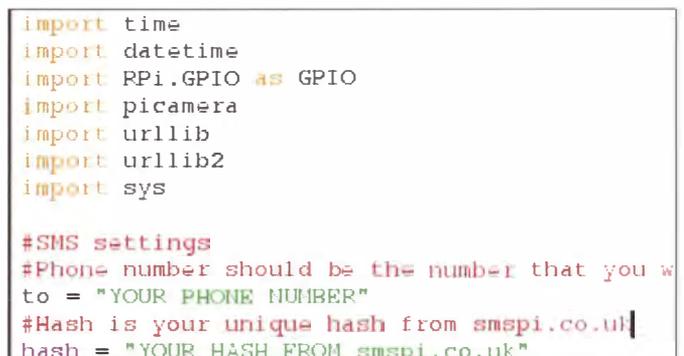


### 4 Installation von picamera

Sie installieren pip mit diesen Befehlen:

```
sudo apt-get update
sudo apt-get install python-pip
sudo pip install picamera
```

Dies installiert die picamera-Bibliothek. Jetzt brauchen Sie den Code für das Projekt, den Sie unter [github.com/lesp/PIR\\_Alarm/archive/master.zip](https://github.com/lesp/PIR_Alarm/archive/master.zip) downloaden können.



### 5 Projektdateien öffnen

Um an die Projektdateien heranzukommen, brauchen wir idle, den Python-Editor. Rufen Sie idle mit dem Befehl **sudo** auf:

```
sudo idle
```

Sobald idle sich öffnet, gehen Sie zu Datei > Öffnen und klicken Sie weiter bis zu den Projektdateien. Wir arbeiten mit `pir_alarm.py`.

### 6 Zusätzliche Extras

Im Code gibt es einen Abschnitt, der Funktionen importiert: Das sind die Bibliotheken. Zuerst importieren wir `time`, um das Programm zu steuern, dann `datetime` für die Kalenderdaten und Uhrzeiten. Als Nächstes importieren wir die `RPi.GPIO`-Bibliothek, die wir in `GPIO` umbenennen. Schließlich noch `picamera`, um die Pi-Kamera mit Python zu steuern.

## Cleverer Code

```

def out(pir):
    a = time.datetime.now()
    (a)
    :19]
    ("Alarm at "+str(a))
    ert)
    a)+(".jpg")
    a)+(".h264")
    = (alert), (pic), (vid)
    resolution = (1024, 768)
    
```

### 1 Nummer des Pins speichern

Oben steht, wie der OUT-Pin des PIR-Sensors mit Pin 7 des Raspberry Pi verbunden wird. Damit dieser Anschluss mit dem Code funktioniert, erstellen wir eine Variable, in der die Nummer des Pins gespeichert wird, die der Code benutzen soll. Schreiben Sie:

```
pir = 7
```

So können wir den Pin falls nötig auch im Code ändern.

```

GPIO.setmode(GPIO.BOARD)

pir = 7

GPIO.setup(pir, GPIO.IN)
camera = picamera.PiCamera()
while True:
    if GPIO.input(pir):
        a = datetime.datetime.now()
    
```

### 3 Code vereinfachen

Im Code steht:

```
GPIO.setup(pir, GPIO.IN)
```

Mit dieser Code-Zeile wird Pin 7, der eine Variable namens **pir** referenziert, auf Input gesetzt; der Pin wartet auf ein Signal vom Sensor. In der nächsten Zeile wird die volle **picamera**-Funktion der Einfachheit halber zu **camera** verkürzt.

```
camera = picamera.PiCamera()
```

```

def sms(to,message,hash):
    values = {
        'to': to,
        'm-smsg': message,
        'hash': hash } # Grab your hash from http://www.smspi.co.uk

    url = 'http://www.smspi.co.uk/send/'

    postdata = urllib.urlencode(values)
    req = urllib2.Request(url, postdata)

    print 'Att=apt to send SMS ...'

    r:
    response = urllib2.urlopen(req)
    response_url = response.geturl()
    if response_url==url:
        print response.read()
    except urllib2.URLError, e:
    
```

### 2 SMS aktivieren

Um den Code, den uns **smspi.co.uk** zur Verfügung gestellt, im Skript unterzubringen, legen Sie in Zeile 14 eine Funktion **sms()** an. Mit den drei Argumenten **to**, **message** und **hash** können wir die Telefonnummer, die Nachricht und den Hashwert für die Nutzung des SMSpi-Gateways versenden. Um die Login-Info und den Hashwert für Ihr Projekt zu erhalten, müssen Sie sich bei dem SMS-Dienst registrieren.

```

19]
("Alarm at "+str(a))
ert)
)+(".jpg")
)+(".h264")
= (alert), (pic), (vid)
resolution = (1024, 768)
apture(pic)
ep(2)
resolution = (640, 480)
    
```

### 4 So funktioniert die Hauptschleife

Falls der Alarm ausgelöst wird:

Lies Datum und Uhrzeit, konvertiere sie zu einer Zeichenkette.

Reduziere Zeichenkette auf die ersten 20 Zeichen.

Erstelle Nachricht und Dateiname für Alarm.

Nimm Bild auf und filme 10 Sekunden lang Video.

Sende Alarm per SMS auf Handy.

Bleibe 30 Sekunden im Ruhezustand.

Wird der Alarm erneut ausgelöst, wiederhole die Abfolge.

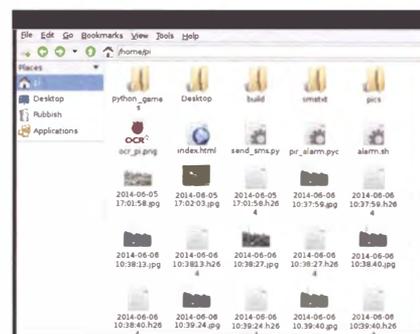
## SMSpi

Das Team von **smspi.co.uk** hat uns für dieses Projekt sein Python-Skript überlassen. SMSpi ist ein Gratis-SMS-Dienst für Raspberry Pi-Projekte. SMSpi wird vom Pi mit Strom versorgt; die SMS werden über das Gerät geroutet, das auch den Alarm antreibt.

Die API kann mit fast allen üblichen Programmiersprachen verwendet werden, von CURL über PERL, Python, PHP bis Ruby. Sie können sich umsonst registrieren, dann bekommen Sie Zugang zum Netzwerk über einen Hashwert, der Ihren persönlichen Schlüssel darstellt. Über das SMSpi-Netzwerk können Sie SMS verschicken und empfangen, diese an eine E-Mailadresse weiterleiten oder auf einer Webseite posten. Ihren Hashwert fügen Sie in

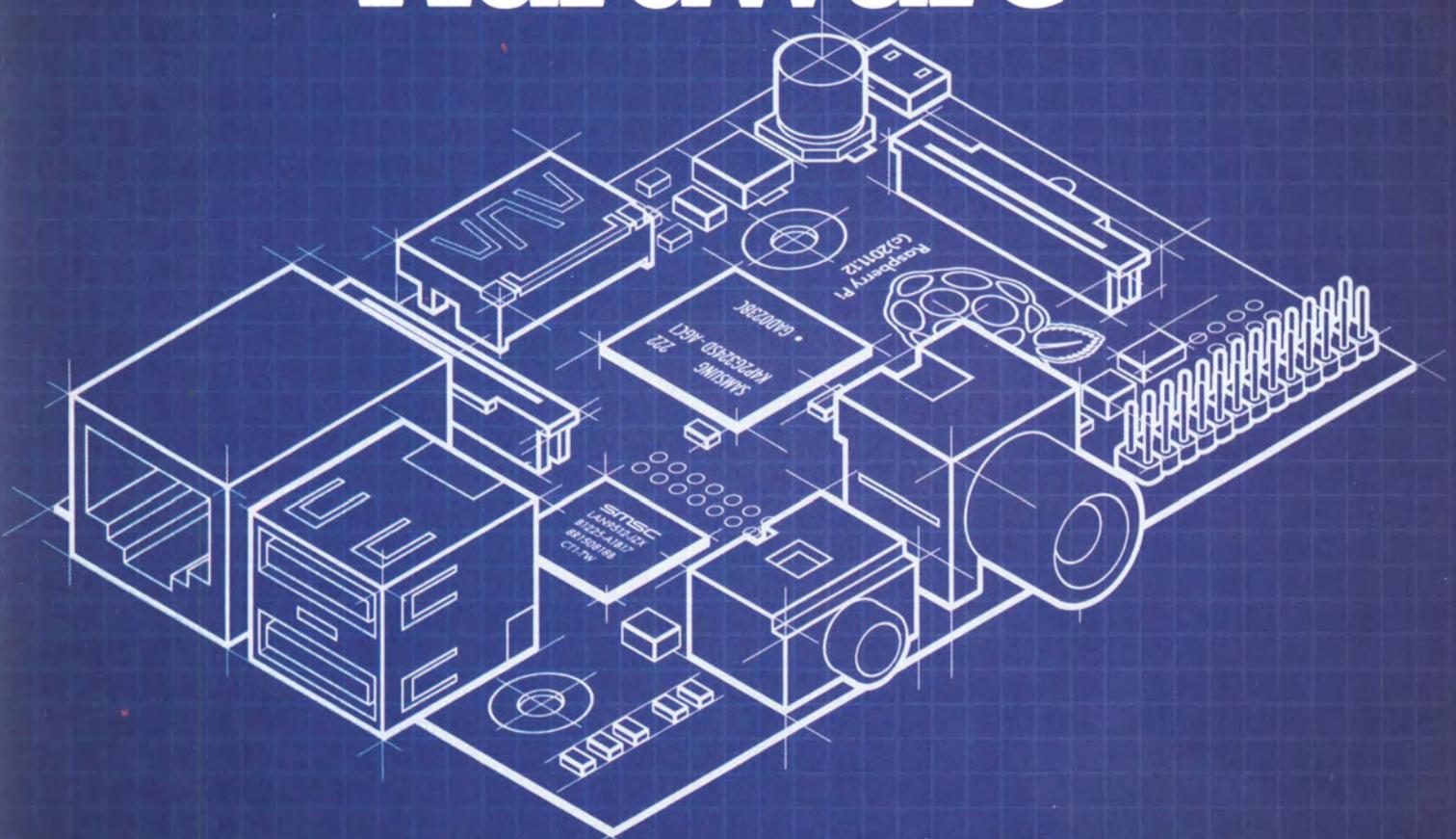
Zeile 11 des Python-Projekt-Codes ein.

Ist Ihre Alarmanlage installiert und in Betrieb, können Sie die Bilder und Videos auf Ihrem Raspberry Pi via Remote-Zugriff einsehen. Am einfachsten ist es, wenn Sie die Karte entfernen und sich die Aufzeichnungen auf Ihrem Computer anschauen. Aber Sie können auch SSH auf dem Pi einrichten, über das erweiterte **raspi-config**-Menü. Verwenden Sie Ihren Dateimanager und mounten Sie mit **SSHFS** den Ordner mit den Bildern. Ganz Mutige können **lighttpd** oder **nginx** installieren, um die Inhalte über ein Netzwerk bereitzustellen. Egal welchen Weg Sie wählen, seien Sie stets vorsichtig, wenn Sie Bilder über eine ungeschützte Leitung im Internet teilen.

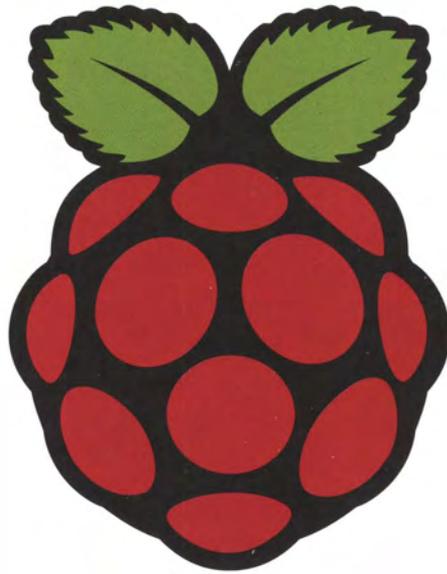


Der Code speichert die aufgenommenen Bilder und Videos in dem Verzeichnis, wo er selbst liegt.

# Raspberry Pi Hardware



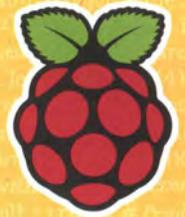
**D**er Raspberry Pi mag ein kleiner und kompakter Computer sein, der im Prinzip alle benötigten Hardware-Bausteine mitbringt. Doch es gibt einige sinnvolle Hardware-Erweiterungen für den Pi, mit denen die Funktionsvielfalt des Geräts sogar noch gesteigert wird. Dieses Kapitel widmet sich den besten Hardware-Erweiterungen, die es für den Pi gibt.



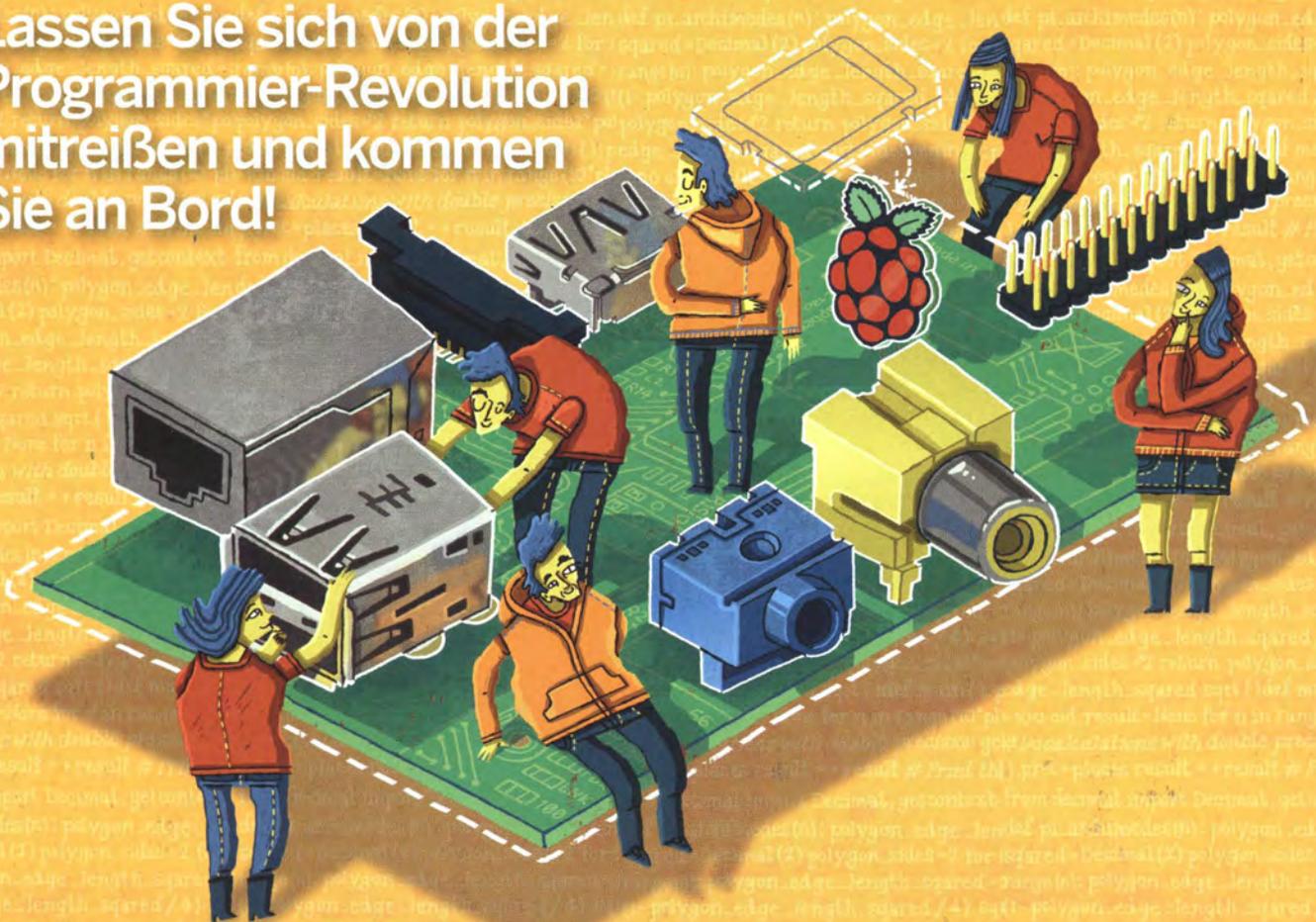
Einstieg in die Welt des Raspberry Pi.....	<b>144</b>
Erweitern Sie den Raspberry Pi.....	<b>152</b>
Compute Module Development Kit.....	<b>156</b>
3-D-Druck: Pi-Case printen.....	<b>158</b>
Wolfson Audiokarte.....	<b>162</b>

# Einstieg in die Welt des

# Raspberry Pi



Lassen Sie sich von der Programmier-Revolution mitreißen und kommen Sie an Bord!



**V**or nicht einmal drei Jahren – Anfang 2012 – erschien der kleine Raspberry Pi auf der Bildfläche, entwickelt von einer gemeinnützigen Stiftung mit dem Ziel, wieder mehr junge Leute dazu zu bewegen, sich mit Informatik zu beschäftigen und im Idealfall das Studienfach zu ergreifen.

Seitdem hat sich der Computerbereich insbesondere in Großbritannien, der Heimat des Raspberry Pi, gehörig verändert. Viele Jugendliche und Kinder, darunter sogar schon Sechsjährige, haben sich über den preisgünstigen Rechner im Scheckkartenformat an das Thema Programmieren herangetraut, und es ist offensichtlich geworden, dass hier eine neue computerbegeisterte Generation heranwächst.

In den Zeiten vor dem Raspberry Pi standen schulische Angebote in der Informations- und Kommunikationstechnologie zumeist im Zeichen der Anwendung von Microsofts Office-Programmen, eventuell auch der Erstellung von Webseiten. Schaut man jedoch heute nach Großbritannien, so findet man dort viele engagierte Lehrer,

## „Eine computerbegeisterte Generation wächst heran.“

die sich mit der Open-Source-Community zusammenschließen, um zu lernen, wie man hackt und wie ein Programmierer denkt. Seinen Niederschlag findet dies in vielen spannenden Projekten wie etwa dem Einsatz des Raspberry Pi bei

Ballonprojekten ([daveakerman.com](http://daveakerman.com)) oder Robotikexperimenten ([simples.net](http://simples.net)) und nicht zuletzt darin, dass britische Prüfungsgremien wie das OCR das Programmieren zunehmend in ihre Lehrpläne integrieren.

In Deutschland ist man noch nicht so weit, aber auch hierzulande gibt es einige engagierte Pioniere an den Schulen, die den Pi im Unterricht einsetzen.

Dank des niedrigen Anschaffungspreises von etwa 35 €, seiner geringen Abmessungen und seiner Vielseitigkeit ist der Raspberry Pi das ideale Werkzeug für Tüftler jeden Alters. Auf den folgenden Seiten erklären wir, wie man den Minicomputer betriebsbereit macht und erste Projekte damit durchführen kann.

# Das Betriebssystem

So installieren Sie Raspbian leicht auf Ihrem Raspberry Pi.

**D**er Raspberry Pi läuft mit einem ARM-Prozessor mit 700 MHz Taktfrequenz. Das ist heutzutage zwar kein Hochleistungschip, ermöglicht jedoch problemlos das Programmieren und Hardware-Hacking. Der Pi besitzt überdies 512 MB Speicher – für das empfohlene Betriebssystem Raspbian ist das reichlich bemessen.

Bei Raspbian handelt es sich um ein abgespecktes Debian (welches wiederum eine GNU/Linux-Distribution ist) für ARM-Prozessoren. Da die meisten Anwender Raspbian auf ihrem Raspberry Pi installiert haben und viele Hacker es beständig weiterentwickeln, bietet dieses Betriebssystem inzwischen eine sehr gute Hardwareunterstützung und kann bedenkenlos empfohlen werden. Es gibt aber auch andere gute Alternativen für den Raspberry Pi, von denen einige auf Seite 8-11 in diesem Bookazine vorgestellt werden.

## Raspbian installieren

Sobald Ihr Raspberry Pi samt Peripherie (Monitor, Tastatur und Maus) bereitsteht, können Sie Raspbian auf ihm installieren. Besuchen Sie mit einem internetfähigen Computer [raspberrypi.org/downloads](http://raspberrypi.org/downloads) und laden Sie von dort das Softwarepaket namens NOOBS herunter. Die Zip-Datei ist



SD-Slot des Pi stecken und diesen mit Strom versorgen, erscheint ein Auswahlmenü auf dem Monitor, aus dem Sie das zu installierende Betriebssystem – also in unserem Fall Raspbian – wählen können. Alles Weitere erledigt NOOBS, was allerdings eine Weile dauert.

Nach der Installation von Raspbian müssen Sie den Raspberry Pi neu booten, und nach dem erneuten Hochfahren erscheint eine Login-Abfrage. Melden Sie sich mit dem Benutzernamen **pi** und dem Passwort **raspberry** an. Tippen Sie nach dem Einloggen den Befehl **raspi-config** in die Kommandozeile, um das Konfigurationswerkzeug für den

Raspberry Pi aufzurufen. Darin können Sie diverse Einstellungen wie zum Beispiel das Tastaturlayout, die Zeitzone und die Speicheraufteilung vornehmen. Empfehlenswert ist es, in der Konfiguration festzulegen, dass der Pi nach dem Hochfahren direkt die grafische Benutzeroberfläche (GUI) anzeigen soll. Falls stattdessen die Kommandozeile erscheint,

tippen Sie **startx**, um auf die GUI zu gelangen. Die grafische Benutzeroberfläche von Raspbian, also dessen Desktopumgebung, heißt LXDE.

Um statt Raspbian ein anderes linuxbasiertes Betriebssystem auf dem Raspberry Pi zu installieren, können Sie zum NOOBS-Interface zurückkehren – dies tun Sie, indem Sie beim Booten des Pi die Shift-Taste gedrückt halten. Einfach ein anderes Betriebssystem aus dem Menü wählen und die Installationsprozedur erneut durchlaufen lassen.

› Der Raspberry Pi eignet sich hervorragend für die Verwendung mit Lochrasterplatten zum lötfreien Schaltkreisdesign.

**„Raspbian bietet eine sehr gute Hardwareunterstützung und wird bedenkenlos empfohlen.“**

etwa 700 MB groß. Die Abkürzung NOOBS steht für „New Out Of the Box Software“, also ohne weitere Umschweife verwendbare Programme – und das stimmt, denn damit geht die Installation des Betriebssystems fast wie von selbst.

Was Sie allerdings hardwareseitig benötigen, ist ein Lesegerät für SD-Karten oder ein entsprechender Einschub an Ihrem Computer, denn die entzippten NOOBS-Dateien müssen auf eine SD-Karte (mit mindestens 4 GB Speicherplatz und FAT32-formatiert) überspielt werden, um sie auf den Raspberry Pi zu bekommen. Wenn Sie die Karte in den

## Diverse Software auf dem Raspberry Pi installieren

Der Raspberry Pi ist ein typischer auf Debian basierender Rechner, das heißt, Softwarepakete können direkt über das Terminal – mithilfe des Kommandos **apt** – installiert und verwaltet werden.

**apt** steht dabei für Advanced Packaging Tool, und dieses Werkzeug managt die Installation, Deinstallation und Prüfung jeglicher Software auf dem Gerät. Um den **apt**-Befehl anwenden zu können, benötigen Sie Root-Benutzerrechte. Diese erhalten Sie

vorübergehend, indem Sie dem eigentlichen Befehl das Kommando **sudo** voranstellen. Es empfiehlt sich, die bereits vorhandene Software zu aktualisieren, bevor Sie neue Pakete installieren. Eine Liste der Updates zu den von Ihnen verwendeten Programmen bekommen Sie mit der Eingabe **sudo apt-get update**.

Um neue Software herunterzuladen und zu installieren, tippen Sie **apt-get install <paket>**, wobei Sie **paket** durch den

tatsächlichen Namen der gewünschten Software ersetzen. Entsprechend deinstallieren Sie Software mit **apt-get remove <paket>**.

Alternativ können Sie Software auch über den Raspberry-Pi-Store herunterladen, allerdings benötigen Sie hierzu einen speziellen Download-Client. Sie bekommen ihn auf [store.raspberrypi.com/download](http://store.raspberrypi.com/download). Alle im Store angebotenen Anwendungen sind Open-Source-Software.

# Geheimwaffe GPIO

Steuern Sie Sensoren und LEDs mit dem Raspberry Pi.

**E**ines der wortwörtlich herausragendsten Hardwareelemente des Raspberry Pi sind die 26 Metallstifte auf einem Sockel in einer der Ecken der Platine. Es handelt sich dabei um Kontaktstifte mit integrierten Schaltkreisen zur Allzweckein- und -ausgabe, die Abkürzung der englischen Bezeichnung dafür lautet GPIO. Über die GPIO-Pins lassen sich beispielsweise Sensoren und LEDs ansteuern. Für jedes Bauteil, das Sie auf der Arduino-Plattform verwenden können, ist ebenso gut der Raspberry Pi geeignet, was funktionale Projekte zu einem günstigen Preis ermöglicht.

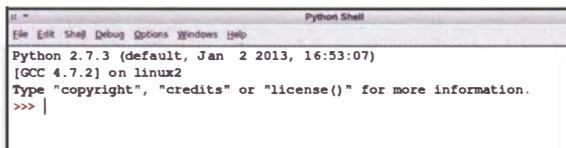
Die Ansteuerung der Sensoren oder LEDs erfolgt mithilfe von Software, und zur Programmierung dieser Software ist Python auf dem Raspberry Pi die erste Wahl. Python ist eine extrem vielseitige Programmiersprache, die nicht nur auf kleinen Geräten wie dem Pi, sondern etwa auch in den Datenzentren von Google zur Anwendung kommt. Von Python existieren unterschiedliche Versionen. Die gängigsten sind 2.7 und 3.4, wobei 2.7 als stabile, wenngleich etwas veraltete Version mit einem hohen Maß an Kompatibilität angesehen wird und 3.4 zwar viele neue Features und Werkzeuge besitzt, aber nicht voll kompatibel mit älteren Python-Bibliotheken ist. Dieses Tutorial basiert auf Python 2.7, der Code sollte aber auch unter Python 3.4 funktionieren.

Wenn Sie das Betriebssystem Raspbian auf Ihrem Raspberry Pi nutzen, ist Python bereits vorinstalliert. Wir verwenden ferner die Python-Entwicklungsumgebung IDLE (siehe Abbildung 1).

## Mit Python programmieren

Sobald IDLE geladen ist, erscheint die Python-Shell, in der Sie schnell in einer interaktiven Umgebung Ihren Code testen können. Dass Sie sich in der Shell befinden, erkennen Sie nicht nur am Titelbalken des Fensters, sondern auch an der Eingabeaufforderung „>>>“.

» **Abbildung 1:** Die integrierte Entwicklungsumgebung für Python, IDLE.



» **Der Raspberry Pi unterstützt sowohl den seriellen Datenbus I<sup>2</sup>C als auch SPI für Master-Slave-Schaltungen sowie die typische GPIO-Funktionalität – gesteuert wird all das mit Python.**

Um jedoch selbst Code schreiben zu können, müssen Sie den Editor aufrufen (über das Menü File > New Window).

Am schnellsten finden Sie sich in Python hinein, indem Sie ins kalte Wasser springen und sich direkt mit etwas Code beschäftigen. Tippen Sie doch einmal die folgenden Zeilen in das Editor-Fenster ein, speichern Sie das Programm, und klicken Sie dann auf Run > Run Module.

```
a = 0
while a < 10:
    print a
    a = a + 1
```

Was geschieht? Die Ziffern 0 bis 9 werden nacheinander jeweils in einer einzelnen Zeile ausgegeben. In der ersten Codezeile wird eine Variable **a** definiert. Eine Variable ist eine Speicheradresse, ein Behälter für einen Datenwert. Mit der zweiten Zeile beginnt eine Schleife, also die Wiederholung einer bestimmten Anweisung: Solange **a** kleiner als 10 ist, sollen die nachfolgenden Codezeilen ausgeführt werden. Das **print** in der dritten Zeile besagt, dass der jeweilige Wert von **a** ausgegeben werden soll. In der vierten Zeile wird festgelegt, dass der Wert der Variablen **a** bei jedem Durchlauf der Schleife um 1 erhöht werden soll. IDLE hat die letzten beiden Zeilen automatisch eingerückt (normalerweise um vier

## Tipps, wie Sie die empfindlichen GPIO-Stifte schützen

Wir referenzieren die Stifte in der Reihenfolge, die in Abbildung 2 dargestellt ist, also alle ungeraden Nummern in aufsteigender Reihenfolge auf der linken Seite und alle geraden in entsprechender Weise auf der rechten, wobei Stift 1 derjenige ist, der dem SD-Einschub des Pi am nächsten liegt.

Die GPIO-Pins 1 und 17 haben einen Output von 3,3 V, das heißt, wenn Sie eine LED oder ein anderes empfindliches Bauelement daran anschließen, sollten Sie am besten einen Wider-

stand zwischenschalten. Die Stifte 2 und 4 geben 5 V ab, auch hier ist ein Widerstand sinnvoll.

Sie sollten ferner nicht die vollen 5 V von diesen GPIO-Stiften abrufen, da der Raspberry Pi sonst instabil werden und zusammenbrechen könnte. Unserer Meinung nach ist 500 mA die größte sicher zu benutzende Input-Stromstärke beim Raspberry Pi. Gehen Sie tunlichst nicht darüber hinaus, andernfalls könnten Sie Ihre CPU frittieren. Beim Output raten wir zu maximal 250 bis

500 mA. Ist der Output zu hoch, schaltet der Pi sich selber ab.

Jumper können mit dem weiblichen Anschluss am Raspberry Pi und dem männlichen an einer Lochrasterplatte befestigt werden, wenn die Stromversorgung eingeschaltet ist, allerdings sollten Sie dringend darauf achten, dass Sie den korrekten Stift verbinden und keinen Kurzschluss fabrizieren, indem zwei Stifte direkt miteinander verbunden sind.

Leerzeichen), da diese so lange ausgeführt werden, wie die Bedingung in der zweiten Zeile erfüllt ist.

In diesem kleinen Codeschnipsel haben wir bereits eine Reihe von Programmierkonzepten kennengelernt:

» **Sequenz** Die zur Ausführung einer Aufgabe notwendigen Schritte.

» **Variable** Ein Behälter für einen Datenwert.

» **Bedingte Anweisung** Ein Programmabschnitt, der unter einer bestimmten Bedingung ausgeführt wird.

» **Schleife** Eine Methode zur Wiederholung eines Prozesses.

Python ist dank seiner Zugänglichkeit und guten Lesbarkeit exzellent als Einstieg in das Programmieren geeignet. Befassen wir uns nun mit der Steuerung der GPIO-Pins mithilfe von Python.

## Steuerung der GPIO-Pins

Um die GPIO-Pins mithilfe von Python ansprechen zu können, müssen wir noch ein bisschen Software installieren. Tippen Sie Folgendes in die Kommandozeile:

```
sudo apt-get update
sudo apt-get install python-dev
sudo apt-get install python-rpi.gpio
```

Die so installierten Pakete eröffnen uns zusätzliche Möglichkeiten, namentlich die Steuerung von physischen Bauteilen. Betrachten wir den Python-Code, der benötigt wird, um eine LED, die mit dem Raspberry Pi verbunden ist, zum Leuchten zu bringen:

```
import RPi.GPIO as GPIO
from time import *
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
led_pin = 8
GPIO.setup(led_pin, GPIO.OUT)
while True:
    GPIO.output(led_pin, True)
    sleep(0.5)
    GPIO.output(led_pin, False)
    sleep(0.5)
```

In den ersten beiden Codezeilen kommt ein **import** vor. Was hat es damit auf sich? Im Laufe der Jahre sind sehr viele Python-Bibliotheken und -Module programmiert worden, und um einzelne davon in unserem Code verwenden zu können, müssen wir sie zuerst importieren, sofern sie nicht zu den Basisbibliotheken gehören, die von vornherein benutzungsfertig installiert sind. Die Bibliothek `RPi.GPIO` ist standardmäßig nicht installiert, also müssen wir sie importieren. Die Bibliothek fügt ein Modul hinzu, mit dem man die GPIO-Pins des Pi kontrollieren kann. Nach dem Importieren der Bibliothek müssen wir noch Einstellungen an der GPIO-Funktion vornehmen:

```
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
```

Der Befehl `GPIO.setwarnings(False)` deaktiviert Fehlermeldungen und verringert den Umfang des Programms, und das `GPIO.setmode(GPIO.BOARD)` sorgt dafür, dass die Belegung der Stifte in eine etwas logischere Reihenfolge gebracht wird: Die Zählung beginnt mit Pin 1 direkt neben dem Einschub für die SD-Karte (siehe Abbildung 2). Als Nächstes folgt die Variable, die als Behälter für die Nummer des Stifts dient, der die LED steuern soll.

```
led_pin = 8
```

3.3V	1	2	5V
I2C 1 SDA	3	4	5V
I2C 1 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
GROUND	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GROUND
GPIO 22	15	16	GPIO 23
3.3V	17	18	GPIO 24
SP10 MOSI	19	20	GROUND
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
GROUND	25	26	SP10 CE1 N

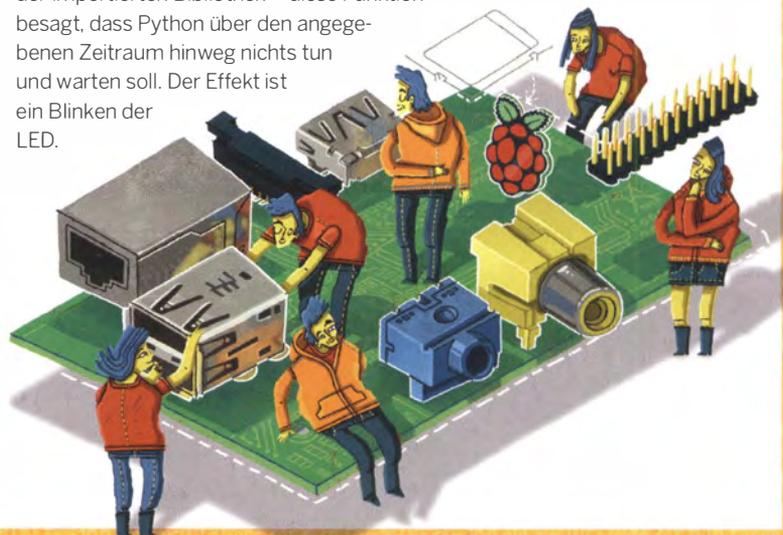
» **Abbildung 2:** Hier sehen Sie den Belegungsplan der GPIO-Pins. Das Modul RPi.GPIO sorgt für eine etwas logischere Reihenfolge.

## LED ansteuern

Indem wir eine Variable einsetzen, statt den Wert explizit anzugeben, können wir den Code schnell dahingehend anpassen, dass er andere GPIO-Stifte anspricht. Die nächste Zeile kontrolliert den Stift an sich und bestimmt, ob er als Input oder Output dienen soll. Wie Sie sehen, deklarieren wir eine Variable `led_pin` und legen fest, dass es sich um einen Output handelt.

```
GPIO.setup(led_pin, GPIO.OUT)
```

Mit der anschließenden Zeile (**while True:**) beginnt eine Endlosschleife. Darin wird die LED abwechselnd für jeweils eine halbe Sekunde an- und ausgeschaltet. Das Ein- und Ausschalten erfolgt mittels `GPIO.output(led_pin, True)` beziehungsweise `GPIO.output(led_pin, False)`. Die Zeitdauer steuern wir mithilfe der `sleep`-Funktion aus der importierten Bibliothek – diese Funktion besagt, dass Python über den angegebenen Zeitraum hinweg nichts tun und warten soll. Der Effekt ist ein Blinken der LED.



# Das erste Hardwareprojekt

Verwandeln Sie Ihren Raspberry Pi in einen kleinen Spielautomaten.

**D**er Raspberry Pi soll insbesondere Kinder und Jugendliche, letztendlich aber Menschen jeden Alters animieren, sich mit den Konzepten der EDV und des Programmierens zu beschäftigen. Die Hardware des Minicomputers ist dazu hervorragend geeignet, da sie sehr zugänglich ist und das Lernen erleichtert.

Für unser erstes Hardwareprojekt greifen wir auf den Code des vorhergehenden Artikels (Seite 146-147) zurück. Darin haben wir erläutert, wie das Programm als solches funktioniert. Nun wollen wir den Aspekt der Hardware betrachten. Unser Schaltkreis ist dabei relativ simpel, aber nichtsdestoweniger aufregend, sollte es Ihr erster Schritt hinein in die Welt der Elektronik sein. Verbinden Sie zunächst die LED mit dem Raspberry Pi wie in besagtem Artikel beschrieben.

Sie können unter dem Link [bit.ly/1vzUki2](http://bit.ly/1vzUki2) den Programmcode herunterladen. Überspielen Sie ihn auf Ihren Raspberry Pi und nehmen Sie (etwa mit dem Befehl `sudo`) Root-Benutzerrechte an, denn nur so können Sie Code ausführen, der auf die GPIO-Kontaktstifte zugreift.

Präparieren Sie dann die Hardware wie folgt: Verdrahten Sie Stift 8 mit der Lochrasterplatte und schalten Sie einen Widerstand von 220 Ohm dazwischen, um die Stromstärke zu verringern, die zwischen dem Raspberry Pi und der LED fließt. Der Widerstand kommt an die Anode (das lange Bein) der LED. Verdrahten Sie anschließend Stift 6 (Erdung) mit der Kathode (dem kurzen Bein) der LED, und schon haben wir einen vollständigen Schaltkreis.

Gehen Sie nun in die Kommandozeile, navigieren Sie (mit den Befehlen `cd` und `ls`) zu dem Verzeichnis, wo der Code liegt, und tippen Sie:

```
sudo idle flash.py
```

Daraufhin sollte die LED beginnen zu blinken. Der erste Schritt ist geschafft!

Im nächsten Schritt stellen wir eine andere Inputmethode vor, einen Mikroschalter. Wir verwenden den Schalter dazu, das Drücken des Startknopfs eines realen Spielautomaten zu simulieren. Sobald der Knopf gedrückt wurde, wählt das Programm nach dem Zufallsprinzip drei Dinge aus und zeigt deren Namen auf dem Bildschirm an. Werden drei gleiche Wörter angezeigt, gibt das Programm das Wort „Jackpot“ aus und die LED beginnt zu blinken. Nehmen wir uns den Code stückweise vor, hier folgt der erste Abschnitt:

```
from random import *
from time import *
import RPi.GPIO as GPIO
button_pin = 8
led_pin = 12
GPIO.setmode(GPIO.BOARD)
GPIO.setup(button_pin, GPIO.IN)
GPIO.setup(led_pin, GPIO.OUT)
```

Zunächst muss das Zufallsmodul **random** in unsere Python-Bibliotheken importiert werden. Damit werden die drei Gegenstände ausgewählt. Das Modul **time** steuert die Geschwindigkeit des Spiels, und **GPIO** macht die Kontaktstifte verfügbar.

## Spielautomat

Es folgen zwei Variablen, **button\_pin** und **led\_pin**. Diese enthalten die Nummern der GPIO-Pins für den Startknopf beziehungsweise die LED. Sollten Sie später einmal andere Pins benutzen wollen, können Sie dies somit ganz leicht im Code anpassen. Die letzten drei Zeilen in diesem Abschnitt steuern das GPIO-Setup. Das `GPIO.setmode(GPIO.BOARD)` teilt Python mit, dass wir die Basisbelegung mit den ungeraden Zahlen in der linken und den geraden Zahlen in der rechten Spalte verwenden, und das `GPIO.setup()` lässt uns festlegen, als was ein bestimmter Kontaktstift fungieren soll – hier legen wir Input für **button\_pin** und Output für **led\_pin** fest.

```
reel1 = ["cherry", "lemon", "bell"]
```

Hier haben wir eine Liste oder ein Array eingeführt. Eine einzelne Liste kann viele Werte enthalten. Wir verwenden insgesamt drei Listen in unserem Programm und nennen sie **reel1**, **reel2** und **reel3**. Sie enthalten alle Gegenstände, die unser Spielautomat auslösen kann.

```
while True:
```

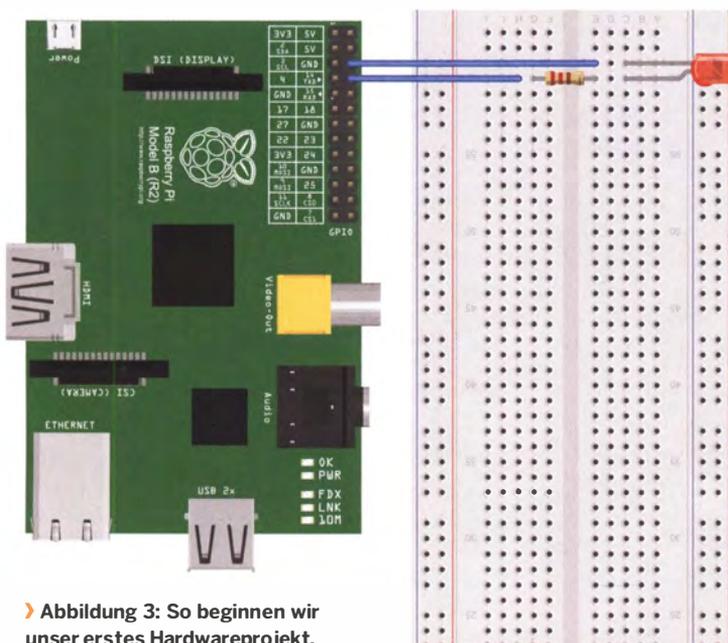
Mit **while True**: starten wir eine Schleife, die so lange durchlaufen wird, bis die Tasten Strg und C gedrückt werden.

```
print("Press the button to start the game")
```

```
sleep(1)
```

Mit den beiden obenstehenden Zeilen lassen wir eine Anfangsnachricht ausgeben und Python eine Sekunde warten, sodass der Benutzer eine Eingabe machen kann.

```
if GPIO.input(button_pin)==1:
    for i in range(0,3):
        r1 = choice(reel1)
```



➤ **Abbildung 3:** So beginnen wir unser erstes Hardwareprojekt.

```
r2 = choice(reel2)
```

```
r3 = choice(reel3)
```

Es folgt nun die eigentliche Logik unseres Spiels. Zuerst starten wir eine bedingte Schleife. Die Bedingung lautet, dass der Startknopf gedrückt wurde. Sobald das der Fall ist, benutzen wir eine Schleife, um die Gegenstände aus den drei Listen auszuwählen, und speichern die jeweilige Auswahl in drei Variablen namens **r1**, **r2** und **r3**.

```
print ("++Playing the game++")
```

```
print (" ")
```

```
print (r1) + " " + (r2) + " " + (r3)
```

```
print (" ")
```

```
sleep(2)
```

Dieser Abschnitt des Codes legt mithilfe von **print**-Anweisungen fest, was auf dem Bildschirm ausgegeben wird. Beachten Sie besonders die dritte Zeile, denn darin findet eine Konkatenation (Verknüpfung) statt. Die Pluszeichen verbinden hier die drei ausgelosten Begriffe und die dazwischen einzufügenden Leerzeichen.

```
if (r1) == (r2) and (r1) == (r3):
```

```
print ("=====")
```

```
GPIO.output(led_pin, True)
```

```
sleep(0.5)
```

```
GPIO.output(led_pin, False)
```

```
sleep(0.5)
```

```
print ("Jackpot!")
```

```
GPIO.output(led_pin, True)
```

```
sleep(0.5)
```

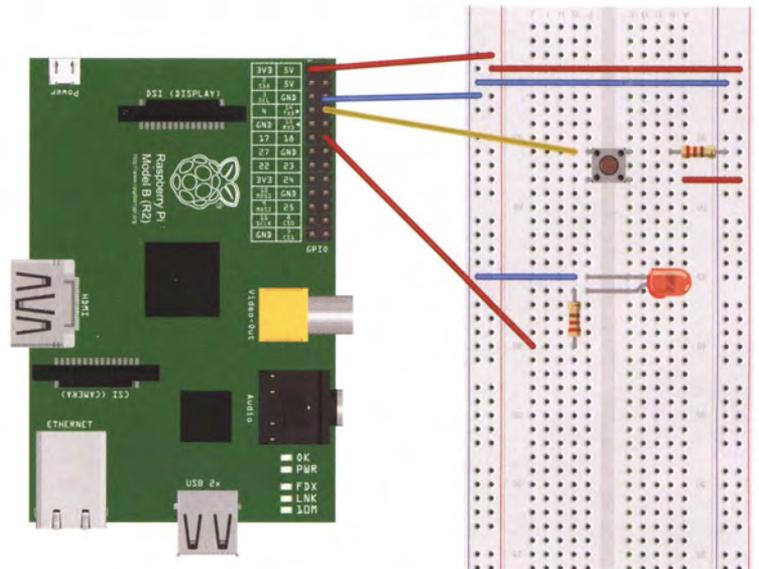
```
GPIO.output(led_pin, False)
```

```
sleep(0.5)
```

Hier haben wir eine weitere bedingte Schleife. Diese überprüft, ob die ausgelosten Gegenstände gleich sind. Wurden drei identische Begriffe ausgegeben, haben also **r1**, **r2** und **r3** den gleichen Inhalt, dann gibt das Programm das Wort „Jackpot“ aus und die LED beginnt zu blinken.

## Die Programmlogik

Der logische Prozess läuft so ab, dass die erste Variable mit der zweiten und dann mit der dritten verglichen wird. Wir wiederholen diese Logik später, wobei wir die Anweisung **elif (else if)** verwenden, die im Grunde besagt, dass wenn die erste Bedingung nicht erfüllt ist, stattdessen die **elif**-Bedingung versucht werden soll. Man kann beliebig viele **elif**-Bedingungen im Python-Code benutzen – wir haben noch zwei weitere eingebaut, um die zweite Variable mit der ersten und dritten und die dritte Variable mit der ersten und



» **Abbildung 4:** Der Schaltplan für den Mikroschalter.

der zweiten zu vergleichen. Die zugehörigen Anweisungen für „Jackpot“ und das LED-Blinken haben wir einfach von der ersten Bedingung übernommen.

Jetzt haben wir unseren Programmcode auf den Raspberry Pi geladen und den Schaltkreis gebaut. Lassen Sie das Programm nun laufen und testen Sie den kleinen Pi-Spielautomaten!

## „Die Hardware des Raspberry Pi ist sehr zugänglich und erleichtert das Lernen.“

Öffnen Sie hierzu IDLE mit dem vorangestellten Kommando **sudo**, damit die GPIO-Pins freigegeben werden, navigieren Sie zum Verzeichnis, wo die Datei **fruitGPIO.py** liegt, und geben Sie ein:

```
sudo idle fruitGPIO.py
```

Sie sollten daraufhin eine Eingabeaufforderung auf dem Bildschirm sehen. Drücken Sie den Startknopf des Spielautomaten, also de facto den Mikroschalter, etwa ein bis zwei Sekunden lang, und schon beginnt das Spiel.

## Input und Output

Denkt man bei den Begriffen Input und Output an den Computerbereich im klassischen Sinne, hat man vermutlich Eingabegeräte wie Tastatur und Maus und Ausgabegeräte wie Bildschirm und Drucker vor Augen. In der Elektronik ist das Spektrum allerdings deutlich größer.

» **Input** Ein Input kann etwas so Einfaches sein wie ein Schalter, der einen Schaltkreis schließt, oder etwas so Kompliziertes wie ein externer Sensor, der Daten sammelt und diese zur Speicherung und späteren Verarbeitung weiterleitet. Es gibt viele Arten elektronischer

Sensoren, einige messen die Temperatur oder Feuchtigkeit, andere nehmen mithilfe von Infrarot- oder Ultraschallwellen Bewegungen oder Entfernungen wahr.

» **Output** Im Sinne von elektronischen Bauteilen ist ein Output etwas, das dem Benutzer irgendeine Rückmeldung gibt. Eine LED wäre in diesem Sinne ein klassischer Output, aber auch Summer, Motoren und Steuergeräte fallen in diese Kategorie.

Outputs werden mithilfe von Inputs gesteuert. So könnte etwa ein Roboter, der den Weg durch einen Irrgarten finden soll, Wände mittels Ultra-

schallsensor erkennen, und der Sensor daraufhin die Motoren des Roboters instruieren, vor der Wand anzuhalten und einen anderen Weg zu wählen.

Für die ersten Hardwareprojekte empfiehlt es sich, ein kleines Arsenal preisgünstiger Elektronikbauteile anzuschaffen. Für etwa 15 € bekommt man beispielsweise all das Folgende:

- » Eine Lochrasterplatte.
- » Über 10 LEDs.
- » Über 20 Widerstände (220 Ohm).
- » Jumperkabel (weiblich/männlich).
- » Jumperkabel (männlich/männlich).

# Zu neuen Ufern

Brechen Sie mit dem Minicomputer in neue Sphären auf!

**W**ir haben auf den vorangegangenen Seiten eine Reihe cooler Projekte und Ideen für den Raspberry Pi vorgestellt. Nun möchten wir einen Schritt weitergehen und den Pi wirklich an seine Grenzen heranführen. Dazu brauchen wir clevere Steuerungslösungen, Programmierwerkzeuge und die Inspiration toller Vorbilder.

Beginnen wir mit BerryIO. Das ist ein schlaues Projekt, das Ihrem Raspberry Pi eine Menge neuer Funktionalität verleihen kann. Es handelt sich um eine Software-Suite mit Steuerungselementen, die per Fernbedienung vom Laptop oder Smartphone aus angesteuert werden können. Sie können also etwa die GPIO-Kontaktstifte oder das Kameramodul des Pi von Ihrem Sofa aus bedienen oder sogar von ganz woanders – einfach über das Internet. BerryIO wird auf ein bestehendes Raspbian draufinstalliert, und das dazugehörige Skript finden Sie unter [bit.ly/1j5DH6Y](http://bit.ly/1j5DH6Y).

Die Web-Anwendung BerryIO ist hauptsächlich in PHP geschrieben. Sie verwandelt Ihren Raspberry Pi in einen einfachen Webserver.

BerryIO hat ein paar tolle Features – insbesondere die Möglichkeit, die Konfiguration der Pi-Kamera zu verändern, ist dabei zu erwähnen. Das Livebild der Kamera kann

## „Wir möchten einen Schritt weitergehen und den Pi an seine Grenzen führen.“

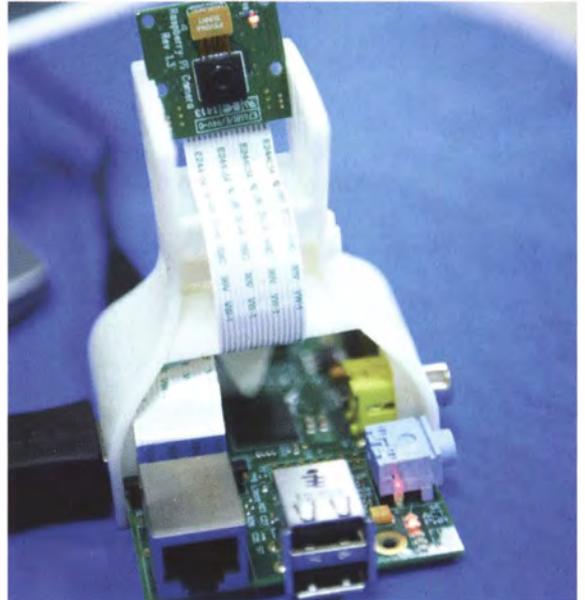
ebenfalls in BerryIO angezeigt werden. Die Funktionen der Software-Suite werden regelmäßig aktualisiert, und kürzlich wurde ein Feature hinzugefügt, bei dem man einige zusätzliche LCD-Boards ansteuern kann. Die Installation von BerryIO ist einfach:

```
wget -N https://raw.githubusercontent.com/NeonHorizon/berryio/master/scripts/berryio_install.sh
chmod +x berryio_install.sh
sudo ./berryio_install.sh
berryio help
```

Sie können sich nun über das Web-Interface mit dem normalen Terminal-Benutzernamen und -Passwort in den Raspberry Pi einloggen.

### Coder

Programmieren lernen ist eine tolle Erfahrung, und das Coden für das Web gilt inzwischen als Norm. Einige Google-Angestellte haben diese beiden Dinge zusammengenommen und die webbasierte Entwicklungsumgebung Coder ([bit.ly/1dNkgsh](http://bit.ly/1dNkgsh)) veröffentlicht. Mit Coder, das gratis zur Verfügung gestellt wird, sollen schon Kinder in die Lage versetzt werden, mittels HTML 5, CSS 3 und JavaScript coole Web-Anwendungen zu erstellen. Die Anwendungen werden auf dem Raspberry Pi gebaut, dort auch ausgeführt und mit der Plattform Node.js als Server bereitgestellt. Die



► **Kombinieren Sie die Kamera des Raspberry Pi und die BerryIO-Software, um zum Beispiel ein Auge auf Ihr Zuhause zu haben, wenn Sie unterwegs sind.**

Entwicklungsumgebung von Coder ist nach den verwendeten Sprachen aufgeteilt, sodass man bequem innerhalb eines Projekts navigieren und sich den Output des Codes ansehen kann. Bei Coder besteht allerdings die Einschränkung, dass Sie damit keinen öffentlichen Server betreiben können, da beim Raspberry Pi keine ernsthaften Sicherheitsmaßnahmen vorgesehen sind. Außerdem kann immer nur ein Benutzer auf die Coder-Umgebung auf dem Gerät zugreifen, das heißt, in einer Schulklasse müsste für jeden Schüler ein eigener Pi vorhanden sein.

### WebIOPi

Die Mission von WebIOPi ist es, eine schlichte Benutzeroberfläche für die Steuerung der GPIO-Stifte des Raspberry Pi bereitzustellen. Während BerryIO viele Features enthält, ist WebIOPi wirklich simpel aufgebaut. Sein klares Web-Interface entspricht genau der Anordnung der GPIO-Pins auf der Platine. Sie können den Status jedes Stifts individuell ändern: zwischen Input- und Output-Zuordnung wechseln und ihn an- und ausschalten. WebIOPi können Sie von [bit.ly/1iOCCOq](http://bit.ly/1iOCCOq) herunterladen.

Das Setup-Skript installiert und lädt mittels **apt-get** alle benötigten Abhängigkeiten. Falls Sie nicht Raspbian als Betriebssystem benutzen, müssen Sie eventuell GCC und Python manuell installieren. Um WebIOPi zu starten, geben Sie Folgendes ein:

```
sudo webiopi -d -c /etc/webiopi/config
```

Falls Sie den Raspberry Pi direkt verwenden möchten, verbinden Sie sich mit <http://localhost:8000>, andernfalls – falls der Pi in Ihr lokales Netzwerk eingebunden ist –, nutzen

## Raspberry Jams – Ein Erfolgsrezept

Bereits kurz nach der Markteinführung des Raspberry Pi kam es zu den ersten organisierten Treffen von Pi-Enthusiasten. Das erste sogenannte Raspberry Jam fand im Juni 2012 in Manchester statt, und seitdem hat sich die Anzahl von Clubs, Gruppen und Veranstaltungen in aller Welt stetig vergrößert.

Bei Raspberry Jams können Tüftler und Hacker ihre Projekte in einer freundlichen, entspannten Atmosphäre Gleichgesinnten vorstellen und Ideen diskutieren. Einige der Treffen sind ein wenig förmlicher und haben einen festgelegten Zeitplan für diverse

Vorträge. Vor allem aber kann man bei den Raspberry Jams seine Kenntnisse erweitern und vertiefen.

In Großbritannien gibt es bereits Tausende von Pi-Enthusiasten, aber auch in Deutschland, Österreich und der Schweiz kann man etliche Gleichgesinnte finden, wie die Landkarte von Ryan Walmsley ([rastrack.co.uk](http://rastrack.co.uk)) zeigt.

Was Deutschland angeht, so fand in Trier bisher fünfmal die Veranstaltung Pi and More ([piandmore.de](http://piandmore.de)) statt. Im Web findet man verschiedene deutschsprachige Diskussionsforen ([forum-raspberrypi.de](http://forum-raspberrypi.de), [raspberrycenter.de](http://raspberrycenter.de)).



› In aller Welt, auch im deutschsprachigen Raum, gibt es mittlerweile Gruppen und Treffen, die den Pi zum Thema haben.



› Scratch ist eine sehr zugängliche Programmiersprache, die sich an Einsteiger richtet. Man kann damit aber auch die GPIO-Pins ansteuern und so etwa einen Roboter steuern.

Sie einen Browser, um sich entsprechend mit Port 8000 zu verbinden, wobei Sie dann das **localhost** durch die IP-Adresse des Raspberry Pi ersetzen.

### Im Höhenrausch

Vermutlich haben Sie von den Rekord-Fallschirmsprüngen von Felix Baumgartner im Rahmen des Projekts Red Bull Stratos im Oktober 2012 (aus fast 39 Kilometern Höhe) oder des Google-Managers Alan Eustace im Oktober 2014 (41,4 Kilometer) gehört. Aber wussten Sie, dass der Teddybär Babbage zusammen mit einem voll funktionsfähigen Raspberry Pi von Dave Akerman per Ballon in ähnliche Höhen geschickt wurde?

Im August 2013 fand dieser denkwürdige Ausflug statt. Der Ballon war mit Wasserstoff gefüllt, und neben dem Teddy war ein Raspberry Pi samt Kamera und Batterie an Bord. Ferner war die Expedition mit einem GPS-Sender und einem 3G-Dongle ausgerüstet, damit die Mannschaft am Boden die jeweilige Position des Ballons bestimmen und den Bären nach Ende des Fluges retten konnte. Akermans Projekt war ambitioniert, aber erfolgreich, und vor allem konnte er es für Kosten von unter 400 € realisieren. Informationen zu diesem und anderen Ballonflügen finden Sie auf der Website [daveakerman.com](http://daveakerman.com).

### Pi-Funksender

Seit 2012 lobt die Unternehmensberatung PA Consulting in Großbritannien einen Wettbewerb für Schüler und Studenten aus, bei dem innovative Raspberry-Pi-Projekte ausgezeichnet werden. Bisherige Gewinnerprojekte schlossen beispielsweise einen Luftqualitätssensor und ein Türsicherheitssystem ein.

Beim aktuellen Wettbewerb, bei dem Umweltthemen im Vordergrund stehen sollen, hat eine Schule aus Blackpool ein Projekt eingereicht, bei dem es darum geht, Opfern von Naturkatastrophen in entlegenen Gebieten eine Möglichkeit zu geben, sich bemerkbar zu machen, wenn die gängigen Kommunikationssysteme ausgefallen sind. Die Schule hat einen wasserdichten Funksender basierend auf dem Raspberry Pi entwickelt, bei dem die Stromversorgung per Batterie und Solarzelle sichergestellt ist. Einer der GPIO-Stifte des Raspberry Pi ist mit einem 20 Zentimeter langen Draht verbunden, der als Antenne fungiert. Die Schüler schrieben und installierten ein Python-Skript, das dafür sorgt, dass Signale ausgesandt werden, sobald ein Notfallschalter gedrückt wird.

Das Projekt ist brillant, und dass es von Grundschulern erdacht wurde, ist außergewöhnlich. Der Pi-Funksender wird auf [bit.ly/1oTx7Ay](http://bit.ly/1oTx7Ay) vorgestellt, auf der Webseite finden Sie auch den zugehörigen Python-Code.

Der Raspberry Pi ist ein wundervolles Gerät, das Tüftler jeden Alters in seinen Bann ziehen kann. Probieren Sie die Projekte aus, die Sie inspirieren und animieren, und lernen Sie dadurch den vielseitigen Minicomputer immer besser kennen.

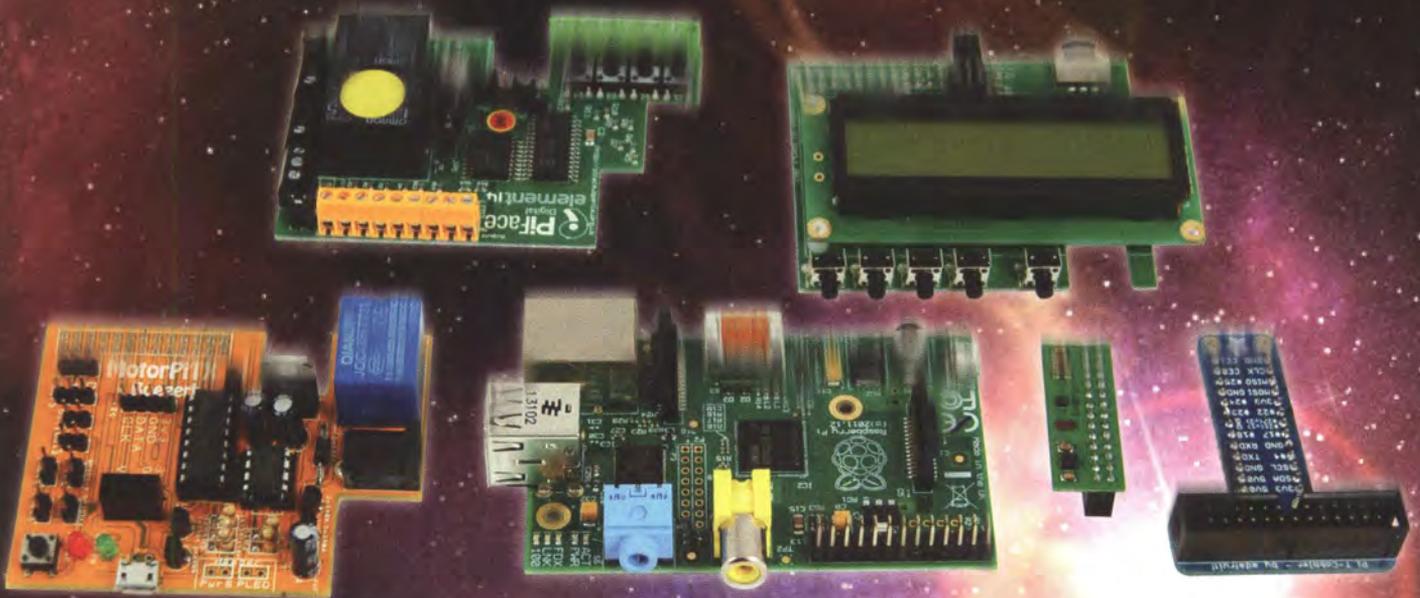
Mit Scratch und Python steigen Sie in die Welt der Programmiersprachen ein, die Kombination aus Raspberry Pi und Lochrasterplatte gibt Ihnen das Handwerkszeug, um das weite Feld der Elektronik kennenzulernen, und der Phantasie und Kreativität sind dabei kaum Grenzen gesetzt.



› Der Raspberry Pi ist auch sehr beliebt in der 3-D-Drucker-Szene.

# Erweitern Sie den Raspberry Pi

Schließen Sie Zusatzmodule an, um noch mehr aus dem Mini-Computer herauszuholen.



**H**and aufs Herz: Was haben Sie letztendlich mit Ihrem Raspberry Pi gemacht? Als der kleine Rechner Anfang 2012 auf den Markt kam, wurde er als Retter der EDV bejubelt, der eine neue Ära des Hobbyprogrammierens einläuten sollte, und die bombastischen Verkaufszahlen waren ein greifbarer Beleg dafür. Doch nicht jeder, der sich von der Begeisterung mitreißen ließ, blieb auch bei der Sache.

Während einige Raspberry Pis Glück hatten und zu XBMC-Clients wurden, landeten andere in Schreibtischschubladen, wo sie dann vergessen wurden. Zwar gibt es ein breites Angebot von Materialien für Raspberry-Pi-Projekte, aber um sich tatsächlich darin zu vertiefen, braucht man den Spaß am Basteln und Tüfteln und die richtigen Bausteine für die jeweilige Aufgabe.

In den letzten zwei Jahren gab es viele Platinen und Erweiterungen für den Raspberry Pi – manche von Hardwareherstellern, aber viele von kleinen Gruppen

interessierter Macher, die auf der Basis des günstigen Geräts fantastische Projekte realisierten.

Warum sollte man Erweiterungen für den Pi verwenden? Natürlich muss man das nicht, aber die erhältlichen Geräte bieten hervorragende Zusatzfeatures für den Raspberry Pi. Beispielsweise bietet PiGlow eine Anzeige mit einer Reihe superheller LEDs; auf der anderen Seite des Spektrums gibt es PiFace (siehe auch das

**„Man braucht den Spaß am Basteln und Tüfteln und die richtigen Bausteine für die jeweilige Aufgabe.“**

Tutorial ab Seite 116), welches eine große Anzahl von Ein-/Ausgabeports für Hardwarebastelprojekte bietet.

Man kann natürlich die GPIO-Pins direkt ansprechen, aber dabei gibt es einige Bedenken. Das GPIO ist fragil, und davon, Komponenten während des laufenden Betriebs daran anzuschließen, wird abgeraten, da man Kurzschlüsse

verursachen könnte, die den Raspberry Pi beschädigen können. Manche Erweiterungsplatinen bieten Features, die das GPIO nicht hat; MotorPiTX beispielsweise bietet eine Schnittstelle für Motoren und Servos. Einen Motor direkt an das GPIO anzuschließen könnte ernsthafte Schäden verursachen.

In diesem Beitrag betrachten wir eine Handvoll Boards, von einfachen GPIO-Breakout-Modulen über Platinen mit

Sensoren und Input-Methoden bis hin zu komplett integrierten Aufbauten.

Der Raspberry Pi bietet eine hervorragende Lerngrundlage und mit diesen Platinen heben Sie das Vergnügen auf ein neues Niveau. Das Internet der Dinge, in dem alltägliche Geräte wie Türkontrollen, Kaffeemaschinen und intelligente Stromzähler über das Internet ferngesteuert werden können – all das ist dank des Raspberry Pi jetzt möglich.

# Adafruit Pi T-Cobbler

Bauen Sie beeindruckende Projekte mit dieser Breakout-Platine für die GPIO-Pins.

**W**ollten Sie sich schon immer mal an Elektronik versuchen? Die GPIO-Pins am Raspberry Pi bieten den perfekten Weg, sich mit den Möglichkeiten und Fallstricken auseinanderzusetzen, aber für die alltägliche Verwendung sind sie etwas zu fragil, wenn man nicht außerordentlich vorsichtig ist. Warum diese Pins also nicht auf einer Steckplatine anordnen? Der T-Cobbler von Pimoroni kann genau das: Er transferiert die 26 GPIO-Pins auf eine Steckplatine und ermöglicht so sicheres Basteln und Lernen. Der T-Cobbler ist besonders nützlich zum Erstellen von Schaltkreisen auf Steckplatinen jeder Größe und bietet hilfreiche Referenzen basierend auf der Pin-Abbild-Spezifikation von Broadcom. Falls Sie sich mit der Standardbelegung auskennen: 3v3 liegt auf Pin 1.

Der Bausatz muss zusammengesetzt werden, das heißt, etwas Löten ist vonnöten – das Bauen ist aber nicht schwierig. Ein mäßig erfahrener Bastler braucht dafür etwa 30 Minuten.

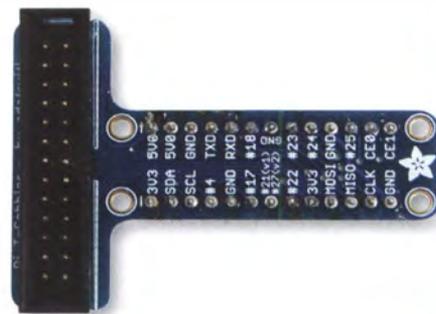
Dankenswerterweise benötigt der T-Cobbler

keine speziellen Treiber oder Konfigurationen; man muss nur die Pakete für die GPIO-Pins – das sind `python-dev` und `python-rpi.gpio` – im Terminal mit `apt-get` auf dem Raspberry Pi installieren.

Der T-Cobbler ist äußerst nützlich für die Prototypentwicklung, da die Pin-Referenzen, die hilfreich auf dem Bausatz aufgedruckt sind, die Funktion jedes Pins schnell beschreiben, beispielsweise GND (Erdung), 5 V, I2C oder SPI.

Wir haben den T-Cobbler mit einem GPIO-Äquivalent von „Hallo, Welt“ getestet, bei dem wir LEDs zum Blinken brachten, indem wir unsere Komponenten an eine Steckplatine anschlossen und einen Schaltkreis zum T-Cobbler herstellten.

Nach diesem Erfolg müssen wir sagen, dass der T-Cobbler wirklich ein ausgezeichnetes Werkstück und als Schnellreferenz für Projekte auf einer Steckplatine sehr nützlich ist. Adafruit stellt auch eine kleinere Variante her, bei der die Pin-Referenzen allerdings schwieriger zu erkennen sind.



› Der T-Cobbler sieht gut aus und leistet beim Kennenlernen des GPIO besonders gute Dienste.

## Fazit

### Adafruit Pi T-Cobbler

**Entwickler:** Adafruit  
**Web:** <http://shop.pimoroni.com>  
**Preis:** ca. 9 Euro

› Ein ausgezeichnetes, wichtiges Werkstück für Ihren kostbaren Raspberry Pi



# MotorPiTX

Bauen Sie Ihren eigenen Roboterhelfer.

**R**oboter erobern den Planeten! Sogar Google investiert in sie... Nun ja, vielleicht ist es bis Skynet noch etwas hin, aber Roboter sind im Zuhause inzwischen üblich – Roomba ist zum Beispiel ein nützliches Robotergerät. Der Raspberry Pi hat schnell seinen Weg in viele Robotikprojekte gefunden und der MotorPiTX macht das Bauen des eigenen Roboters noch einfacher.

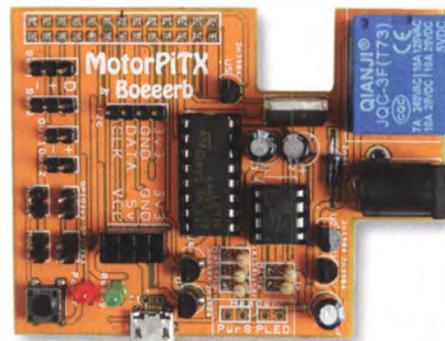
Die Erweiterungsplatine MotorPiTX von Jason Barnett bietet Controller für zwei Motoren oder Servos und zwei externe Stromanschlüsse: einen Micro-USB- und einen Zylinderanschluss. Beide Eingänge reduzieren die Belastung auf den Raspberry Pi, der sonst unberechenbar wird. Jason hat außerdem ein wichtiges Feature eingebaut, das beim Pi fehlt: einen Stromschalter, modelliert nach dem ATX-Stromsystem. Damit kann man mit einem Knopfdruck das System abrupt abschalten, ohne das Micro-USB-Kabel herausziehen zu müssen, oder sauber herunterfahren und neu starten. Um einen Roboter oder eine

Schwenkkamerahalterung zu bauen, muss man jetzt nur noch den MotorPiTX an den Raspberry Pi und dann die gewünschten Motoren an die Platine anschließen.

Wir sind dem Leitfaden auf Barnetts Website ([www.boeereb.co.uk/motorpitx](http://www.boeereb.co.uk/motorpitx)) gefolgt, in dem Informationen zu der Platine zu finden sind, aber Sie können auch auf seiner Github-Seite (<https://github.com/Boeereb/MotorPiTX>) mehr über die Softwareinstallation erfahren. Wir sind der Installation gefolgt und haben zwei Dateien heruntergeladen:

**servod** und **motorpitx.py**. Dann haben wir einen einfachen Servo angeschlossen und ein Testskript geschrieben, das **motorpitx** importiert, und die Funktion **motorpitx.servo1()** mit einem Wert aufgerufen, der dem Bogen unseres Servos entspricht.

Warum sollte man das also im eigenen Projekt verwenden? Die einfache Antwort ist, dass es ziemlich einfach zu verwenden ist und mit vielen Arten von Motoren und Servos funktioniert, wodurch man tolle Roboter, Kameraschwenkmechanismen oder sogar Laser schaffen kann.



› Mit dem Raspberry Pi einen Roboter mit Lasern zu bauen, ist jetzt kinderleicht.

## Fazit

### MotorPiTX

**Entwickler:** Jason Barnett  
**Web:** [www.boeereb.co.uk/shop/](http://www.boeereb.co.uk/shop/)  
**Preis:** ca. 30 Euro

› Die Barriere auf dem Weg zum eigenen Roboter ist mit dieser Platine überwunden.



# PiFace

Ein preiswertes Mehrzweck-Modul für Wissenschaft, Industrie – und Kinder!

**D**iese I/O-Erweiterungsplatine kann weit mehr, als nur eine LED zum Leuchten oder einen Buzzer zum Klingeln zu bringen. Sie möchten vielleicht einen Motor oder eine NerfGun-Spielzeugpistole an Ihren Raspberry Pi anschließen? Mit dem PiFace-Board geht das.

Das PiFace besitzt acht digitale Eingänge, vier Schalter und acht Ausgänge und gibt Ihnen damit eine enorme Flexibilität bei Ihren Projekten. Dank zweier 12-V-Umschaltrelais können Sie auch Komponenten mit höherer Spannung ansteuern.

Das PiFace bringt überdies zwei wichtige Softwarelösungen mit: Es enthält einen GUI-Simulator, der es Ihnen erlaubt, mit dem Gerät herumzuexperimentieren, ohne irgendwelche Bauteile anfassen zu müssen, und ist außerdem voll kompatibel mit der Programmiersprache Scratch, das heißt, auch Kinder können in einer vertrauten Softwareumgebung ihre eigenen Schaltungen entwickeln. Diese Funktionalität ist ein besonderer Vorteil des PiFace.

Für unseren Test haben wir Python verwendet, um das PiFace zu steuern, und wir haben das Kit basierend auf dem bei <http://piface.github.io> verfügbaren Guide installiert und konfiguriert. Mittels der im PiFace eingebauten Funktionen konnten wir ohne großen Aufwand per Schalter LEDs kontrollieren und Sensor-Input registrieren.

Für wen ist das PiFace eine gute Wahl? Wenn Sie ein wirklich preiswertes Board suchen, das die Steuerung von Maschinen und Komponenten mit höherer Spannung ermöglicht, dann liegen Sie mit dem Gerät richtig. Dank der Python-Kompatibilität und der einfach zu verwendenden Bibliotheken lässt das PiFace sich zudem recht unkompliziert und gefahrlos in bestehende Projekte integrieren. Als Mehrzweck-Modul für Wissenschaft oder Industrie ist das PiFace unschlagbar, nicht zuletzt aufgrund der niedrigen Anschaffungskosten. Das Gertboard stellt zwar eine gute Alternative dar, doch muss man dort selber zum Lötkolben greifen, während das PiFace vorgefertigt ausgeliefert wird.



› Mit dem PiFace können Sie größere und bessere Projekte realisieren, die auf Python basieren.

## Fazit

### PiFace

**Entwickler:** Element 14

**Web:** [de.farnell.com](http://de.farnell.com)

**Preis:** ca. 25 Euro

› Eine tolle Lösung, wenn man Projekte und Prototypen schnell in Gang bringen möchte



# XLoBorg

Widerstand ist zwecklos, Sie werden assimiliert!

**B**astler freuen sich ja gemeinhin über neue Boards, mit denen Sie herumexperimentieren können und vom Entwickler PiBorg kommen immer wieder tolle Module mit interessanten Details. So lassen sich mit dem PicoBorg kleine Motoren steuern und das LedBorg fügt dem Pi eine ultrahelle LED hinzu. Wir haben aus dem umfangreichen Angebot des Entwicklers hier das XLoBorg ausgewählt – ein extrem vielseitiges Board mit Bewegungs- und Richtungssensoren, das per Python gesteuert werden kann.

Das XLoBorg ist mit einem dreiachsigen Beschleunigungsmesser und einem Magnetometer ausgestattet und kann ein breites Spektrum von Bewegungen – von sanften Schlägen bis hin zu weiten Kreisbögen – erfassen. Das Magnetometer eignet sich zusätzlich zur Temperaturmessung.

Das Erscheinen des Raspberry Pi und der dafür entwickelten Erweiterungen hat zu einer Vielzahl neuer Robotikprojekte geführt. Das XLoBorg ist eine tolle Option in der Robotik und eignet sich gar zur Drohnensteuerung. Die Platine ist leicht gebrauchsfertig zu machen, sie

braucht nur an die GPIO-Pins des Raspberry Pi angeschlossen zu werden – dann noch die Installation der benötigten Python-Bibliotheken, die auf [www.piborg.com/xloborg](http://www.piborg.com/xloborg) heruntergeladen werden können. Da das XLoBorg über den seriellen Datenbus I2C mit dem Pi kommuniziert, wird eine Anzahl von Pins freigesetzt, sodass Sie das Modul zu einer Steckplatine für größere Projekte erweitern können. Es ist kompatibel mit dem TriBorg (ebenfalls aus dem Hause PiBorg), das die Menge der GPIO-Pins verdreifacht.

Wir haben uns an die offizielle Anleitung gehalten und ein Installationskript für den Raspberry Pi heruntergeladen. Das Skript erledigt das Setup des Pi für den I2C-Bus. Nach einem Reboot haben wir XLoBorg.py ausgeführt, um die Einstellungen zu testen, und erhielten eine Menge schöner On-Screen-Daten, die sich jeweils veränderten, wenn wir den Raspberry Pi umherbewegten und daraufhin die Sensoren ansprachen.

Das XLoBorg ist ein exzellentes Zubehör, wenn Sie für Ihre Projekte diverse Sensoren benötigen.



› Beschleunigungsmesser und Magnetometer sind Sensoren, die in vielen Smartphones Verwendung finden. Hier können Sie sie mit Ihrem Raspberry Pi verbinden.

## Fazit

### XLoBorg

**Entwickler:** PiBorg

**Web:** [www.piborg.com/xloborg](http://www.piborg.com/xloborg)

**Preis:** ca. 11 Euro

› Star Trek für Bastler: Machen Sie Ihren Raspberry Pi zum Tricorder.



# Control & Display

Eine interessante Variante des PiFace-Moduls

**D**er Raspberry Pi ist das Gerät der Wahl, wenn es um Projekte im Zusammenhang mit dem Internet der Dinge geht, doch ein wiederkehrendes Problem dabei ist, eine einfache Methode für Input und Output zu finden. Bei Desktop-Anwendungen sind klassischerweise Tastatur, Maus und Monitor die Mittel dazu, aber was ist, wenn Sie ein Internet-Radio für die Küche oder einen einfachen Videorecorder bauen möchten? Dafür gibt es zum Beispiel das PiFace Control & Display.

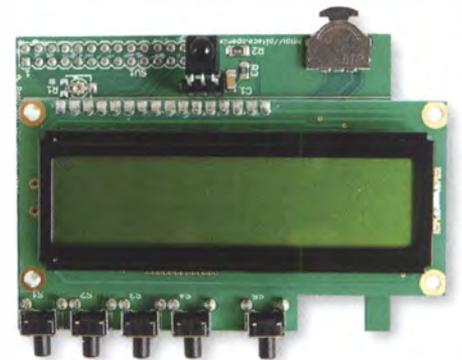
Das PiFace Control & Display ist ein Erweiterungsmodul, welches physische Eingaben über eine Reihe von Mikroschaltern und einen Drehregler ermöglicht, während der Output über ein LCD-Display geschieht. Ein weiteres interessantes Feature des Boards ist der eingebaute Infrarot-Empfänger, der so eingestellt werden kann, dass er mit vielen Fernbedienungen funktioniert.

Von seinen Abmessungen her entspricht das PiFace Control & Display ziemlich genau dem Raspberry Pi, das Board passt somit gut auf die GPIO-Pins des Pi. Die Installation der Software ist einfach: Es werden lediglich

ein schnelles Repository-Update und die Installation von python3-pifaced benötigt (Sie können python3 durch python ersetzen). Wir haben unser Board mit dem Standard-Testprogramm überprüft, welches die IP-Adresse, Temperatur und CPU-Last des Raspberry Pi auf dem LCD-Display anzeigt. Alle Tests benötigen nicht mehr als 60 Zeilen Python-Code und die Funktionsbibliothek ist riesig und deckt sehr viele Verwendungsarten ab – die Beispielskripte reichen von Bahn-Abfahrtszeiten bis hin zu Internet-Radiosendern.

Das PiFace Control & Display ist ein großartiges Teil und bietet viele neue Möglichkeiten, Stand-alone-Projekte zu realisieren. Die Infrarot-Funktion ist eine willkommene Ergänzung des Arsenal – hier kann man sich vom Mediacenter bis zur Lichtschranke einiges vorstellen.

Zwar gibt es viele Module, die Input/Output-Lösungen bieten, aber kaum ein anderes erreicht eine solch ausgereifte Funktionalität wie das Control & Display von Element 14. Mit diesem Modul öffnen sich ganz neue Möglichkeiten bei Projekten auf Industrieniveau, aber auch im Bereich von Kunstinstallationen.



» Ob Internet-Radio oder ferngesteuertes Kamerastativ, das PiFace Control & Display ist die Lösung.

## Fazit

### PiFace Control & Display

**Entwickler:** Element 14

**Web:** de.famell.com

**Preis:** ca. 33 Euro

» Ein innovatives, attraktives Modul für Kommunikation und Steuerung durch den Raspberry Pi



## Betritt die Matrix!

Die Entwickler des PiFace Control & Display haben ein faszinierendes Projekt aufgezogen. Erinnern Sie sich an die Matrix-Filme der Wachowskis? Darin wurde die damals bahnbrechende Technik der Bullet-Time-Spezialeffekte angewandt, bei der der Eindruck einer Kamerafahrt um ein feststehendes Objekt herum erzeugt wird. Heute kann man dies auch mit dem Raspberry Pi, dem Raspberry-Pi-Kameramodul und dem PiFace Control & Display bewerkstelligen – allerdings benötigt man mehrere Dutzend Geräte dafür.

Mithilfe einer Apparatur aus je 48 Einheiten jedes der genannten Geräte – in ringförmiger Anordnung – sowie einem halben Kilometer an

Kabeln ist es dem PiFace-Team gelungen, eine Zeitraffersequenz von Fotoaufnahmen zu erstellen, bei der dasselbe Objekt aus 48 verschiedenen Winkeln gleichzeitig abgeleuchtet wurde. Der Programmcode für die Apparatur war relativ simpel und beruht auf den notwendigen Netzwerkverbindungen.

Nun mögen Sie womöglich einwenden, dass ein solches Projekt mit ziemlich satten Anschaffungskosten verbunden ist. Es geht allerdings auch eine Nummer kleiner: Wie wäre es beispielsweise mit Zeitrafferaufnahmen der Pflanzen in Ihrem Garten oder mit einer Mini-Fotokabine auf Ihrer nächsten Party? Auch dafür eignet sich das PiFace Control &

Display hervorragend, und sie brauchen nur eins davon.

Der Code zur Steuerung des Control & Display ist auf der Webseite erhältlich und die neue Python-Bibliothek für das Raspberry-Pi-Kameramodul liegt inzwischen ebenfalls vor (<https://pypi.python.org/pypi/picamera>). Sie ermöglicht eine bessere Kontrolle über die Kamera einschließlich Helligkeitseinstellung und Zeitrafferfunktion.

Dank der vielseitigen Input-Möglichkeiten wie Knöpfe, Schalter und Infrarot-Empfang ist das PiFace Control & Display ein tolles Kit. Das LCD-Display bietet überdies eine tolle Output-Alternative zu raumgreifenden Monitoren.



» Warum kleckern, wenn man auch klotzen kann? Ein Ring aus 48 Raspberry Pis.

# Compute Module Development Kit

Sie meinen es richtig ernst mit Ihren Raspberry-Pi-Projekten? Diese brandneue Inkarnation des Minicomputers könnte das sein, was Sie brauchen.

## Abriss.

» Eine kompaktere und robustere Variante des Raspberry Pi für die kommerzielle Produktentwicklung im größeren Stil.

Der Raspberry Pi ist sehr beliebt in der Open-Source-Community, und viele tausend Hacker und Bastler treiben ihre Projekte mit diesem kreditkartengroßen Computer an. Während seines erst kurzen Lebens fand der Pi aber auch schon in vielen kommerziellen Projekten Verwendung. Oft musste der Rechner dafür jedoch zuerst angepasst werden, und die Anpassung eines bereits existierenden Produkts für einen neuen kommerziellen Zweck ist selten eine elegante Lösung. An dieser Stelle kommt das Compute Module ins Spiel.

Das Compute Module sieht wie ein typisches SODIMM-Speichermodul aus einem Laptop aus. In Wirklichkeit ist es aber ein vollwertiger Raspberry Pi (Modell A), in eine neue Form gepresst. Die SODIMM-Erscheinung ermöglicht die Anwendung des Compute Module in allen möglichen Nutzungsszenarien, weil es für den Einsatz zusammen mit einer Erweiterungsplatine konzipiert wurde, nämlich der Breakout-Platine „Compute Module IO Board“, kurz CMIO. Das Compute Module wird in Kombination mit dem CMIO unter der Produktbezeichnung Compute Module Development Kit vertrieben.

Das CMIO liefert den Zugang zu nicht weniger als 120 GPIO-Pins, die



» Das Compute Module ist eine kleine Platine der Größe eines SODIMMs, die entwickelt wurde, sich in industrielle Umgebungen zu integrieren.

vom BCM2835-Chip zur Verfügung gestellt werden. Es besitzt zudem 4 GB Flashspeicher, auf dem Sie das Betriebssystem unterbringen können. Das Modul nutzt keine SD-Karte wie die Standardversion des Raspberry Pi, stattdessen wird das Betriebssystem per Micro-USB auf den Speicher aufgespielt. Wenn Sie das CMIO betrachten, sehen Sie einen Micro-USB-Anschluss sowie einen USB-Anschluss im Standardformat. Sie werden auch noch einen weiteren Micro-USB-Anschluss sehen, der mit „USB OTG“ beschriftet ist. Das ist ein USB-On-the-go-Anschluss, jener, der mittels des CMIO für die Installation des Betriebssystems genutzt wird. Darauf werden wir später detaillierter eingehen.

## Anschlüsse zur Genüge

Wenn wir das CMIO näher unter die Lupe nehmen, finden wir außerdem vier Anschlüsse mit den Beschriftungen „DISPO“, „DISP1“, „CAM0“ und „CAM1“. Diese Ports sind für das offizielle Raspberry-Pi-Display und die Kamera. Diese Anschlüsse sind nicht die gleichen wie auf dem Standard-Pi, und um sie zu benutzen, brauchen Sie einen Adapter. Diesen bekommen sie im Handel, sie liegen aber auch dem Display und der Kamera bei.

Vorher erwähnten wir den USB-OTG-Anschluss und wie er verwendet wird, um das Compute Module mit Ihrem PC zu verbinden. Der Anschluss verbindet sich mit Ihrem Computer über eine spezielle Software, die Sie auf der Raspberry-Pi-Website zum Download finden. Damit nutzen Sie den Flashspeicher genauso wie einen USB-Stick oder eine SD-Karte. Beide Micro-USB-Anschlüsse des CMIO müssen verbunden sein: Der OTG mit Ihrem PC, der andere mit der Stromversorgung. Sobald das CMIO angeschlossen und die Software gestartet ist, dürfen sie mit dd ihre bevorzugte Distribution direkt auf den Speicher kopieren. Für diesen Artikel haben wir uns dabei für die Standardversion von Raspbian entschieden, weil sie die stabilste und am besten unterstützte Distribution für den Raspberry Pi ist. Bisher unterstützt das Compute Module nicht die NOOBS-Methode für die Installation, Sie müssen also zwingend zu dd greifen.

Mit einem installierten Raspbian schließen wir USB-Hub, Maus, Tastatur und einen Bildschirm an und booten das Modul das erste Mal. Sie werden überrascht sein, wie schnell das geht. Das liegt am schnellen Flashspeicher, der statt der SD-Karte verwendet wird. Nachdem der Bootvorgang

## Auf einen Blick



### Großzügiges GPIO

Das Compute Module hat 120 GPIO-Pins – genug selbst für große Projekte.



### Doppeltes Display

Die Verwendung je zweier Kameras und Displays ist eine Ausstattung für medienaffine Entwickler.

abgeschlossen ist, werden wir zum Raspbian-Konfigurationsmenü geleitet. Dort stellen wir alles gemäß unseren Erfordernissen ein. Nach einem Neustart sehen wir auch schon den Login-Bildschirm, wo wir die wohlbekannten Zugangsdaten eintragen. Da Sie mit der Software schon vertraut sein dürften, springen wir gleich zur wichtigsten Merkmal des Compute Module: dem GPIO.

## Pins und Python

Das CMIO macht alle GPIO-Pins des BCM2835 verfügbar – insgesamt 120, aufgeteilt in zwei Reihen à 60 Pins. Von diesen werden 45 von Software gesteuert, konkret über die RPi.GPIO-Bibliothek für Python. Die verbleibenden Anschlüsse sind für die Stromversorgung (1,8 V, 3,3 V und 5,0 V), Masse sowie spezielle Pins für externe Geräte (wie Kameras, USB-OTG und Displays) reserviert. Der originale Raspberry Pi verwendet genau denselben BCM2835, kann aber nur mit einer begrenzten Anzahl von GPIO-Pins umgehen. Nur auf dem CMIO steht genügend Platz für das volle Angebot für größere Projekte zur Verfügung. Um das GPIO zu testen, wollten wir einige Pins verwenden, die beim Standard-Raspberry-Pi nicht zur Verfügung stehen, um zu beweisen, dass die Python-Bibliothek auch mit dem CMIO funktioniert. Deshalb schrieben wir ein kurzes Python-Skript mit IDLE (denken Sie daran, dem Befehl das sudo voranzustellen, denn nur sudo oder root dürfen auf das GPIO zugreifen), das durch vier LEDs rotiert. Wir speicherten das Skript und führten es mit F5 aus. Es funktionierte gleich beim ersten Mal, die LEDs leuchteten

nacheinander auf, was bewies, dass die Standardbibliothek mit dem Compute Module funktioniert.

Das Layout des CMIO-GPIO ist nicht dasselbe wie das des normalen Raspberry Pi. Sie können dafür gebaute Erweiterungsmodule also nicht mit dem CMIO verwenden. Sie können die benötigten Pins aber manuell per Jumper-Kabel verbinden, auch wenn das nicht offiziell unterstützt wird. Das CMIO verwendet das Broadcom-Layout und ist ausführlich beschriftet, sodass Sie alle Pins finden, die Sie benötigen. Um das Layout in Python zu nutzen, brauchen Sie den Befehl

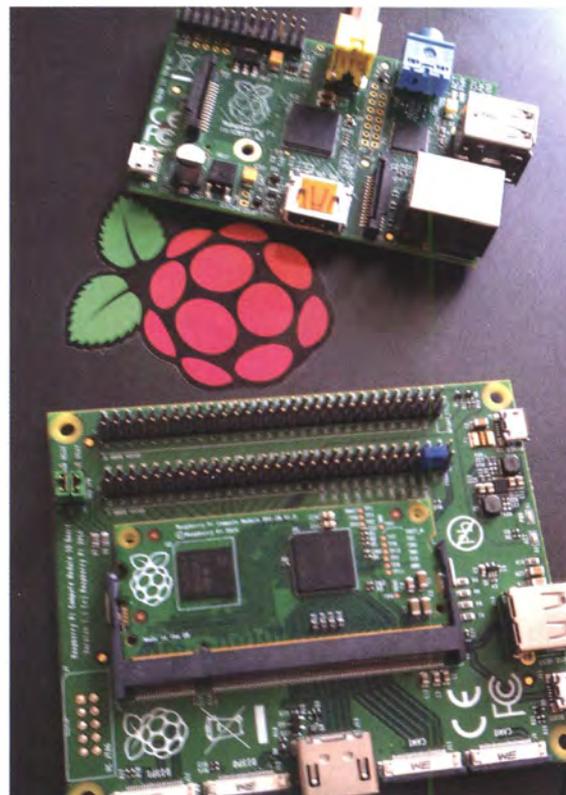
```
GPIO.setmode(GPIO.BCM)
```

statt des üblichen

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

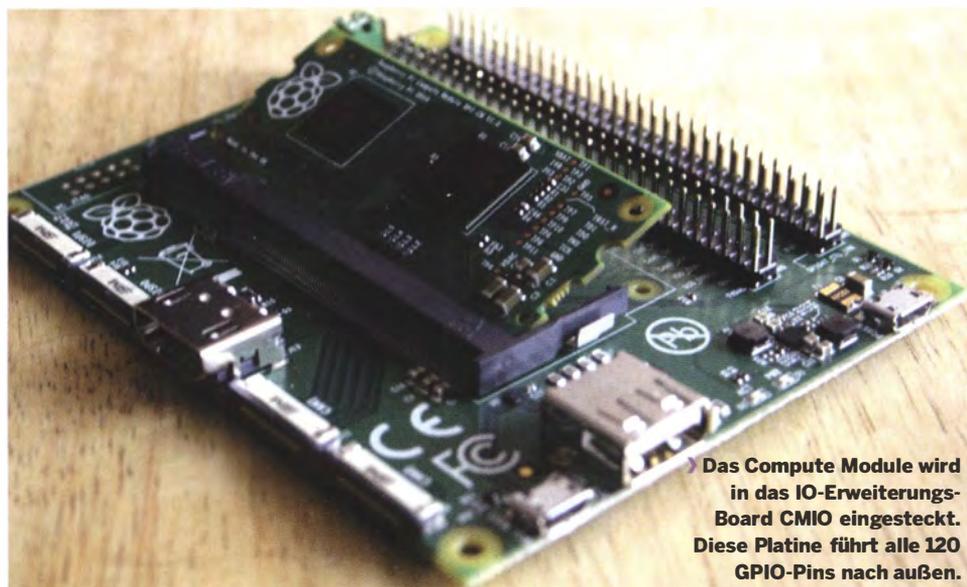
Diese Änderung ist klein, kann Sie aber viel Zeit kosten, falls Ihnen nicht auffällt, dass es sich um ein abweichendes Pin-Layout handelt.

Das Compute Module Development Kit ist wohl nichts für die meisten Hobbybastler, denn es kostet stolze 200€. Die erste Charge des Sets kostet so viel, weil sie vor allem für ernsthafte Entwickler gedacht ist, die damit ihre kommerziellen Projekte realisieren möchten. Die Raspberry-Pi-Stiftung legt sehr viel Wert darauf, dass ihre stabile und breit unterstützte Plattform auch für kommerzielle Projekte genutzt wird. Sie erhofft sich neue Breakout-Platinen wie das CMIO, aber mit neuen Ideen und Konzepten. Als Beispiel sei das Kickstarter-Projekt OTTO genannt, eine hackbare Kamera, die um das Compute Module herum gebaut wurde. OTTO verbindet sich mit Ihrem Smartphone, damit sie Fotografien im Timelapse-Modus aufzeichnen können.



› Das CMIO ist größer als der Raspberry Pi, was mehr Erweiterungsmöglichkeiten gibt.

Möglich ist das nur aufgrund des Formfaktors des Compute Module. Nach der ersten Charge will die Raspberry-Pi-Stiftung das CMIO in Massenproduktion fertigen, sodass jeder die Chance hat, es sich zu kaufen. Das Ziel der Stiftung ist es immer noch, Kinder die Grundlagen der Informationstechnik beizubringen. Der mit dem Compute Module erwirtschaftete Gewinn soll dabei helfen. Das Compute Module und seine IO-Platine sind ein großartiges Entwicklerboard für ernsthafte Projekte wie etwa industrielle Steuerungssysteme und Heimautomatisierung. Für Nutzer der alten Schule reicht aber der originale Raspberry Pi (Modell A oder B) immer noch voll aus.



› Das Compute Module wird in das IO-Erweiterungs-Board CMIO eingesteckt. Diese Platine führt alle 120 GPIO-Pins nach außen.

## Fazit

### Compute Module Development Kit

**Entwickler:** RPi-Stiftung  
**Web:** raspberrypi.org  
**Preis:** ca. 200 €

<b>Funktionen:</b>	9/10
<b>Leistung:</b>	9/10
<b>Simplität:</b>	8/10
<b>Dokumentation:</b>	9/10

› Eine Platine speziell für den Entwickler größerer Projekte. Der Preis wird in den kommenden Monaten schnell fallen.

**Wertung 9/10**

# 3-D-Druck: Pi-Case printen

Mithilfe von Python und FreeCAD können Sie selber ein Gehäuse für den Minicomputer entwerfen. Text oder Logo sind auch möglich.

**W**enn Sie schon mal das Erlebnis hatten, ein brandneues Apple- oder BlackBerry-Gerät aus seiner aufwendigen Verpackung zu nehmen, könnten Sie beim Auspacken Ihres ersten Raspberry Pi ein wenig enttäuscht sein, denn der Pi kommt in einer schmucklosen Pappschachtel daher, lediglich umhüllt von Antistatikfolie, und die einzigen Beilagen sind zwei schlichte Informationsblätter.

Der Raspberry Pi selbst ist eine nackte Leiterplatte mit einigen Bauteilen drauf und sieht aus wie das Mainboard eines PCs. Wenn Sie den Pi auf den Tisch legen, bekommt die Tischplatte Kratzer. Wenn Sie aus Versehen auf das Gerät treten, rammen Sie sich jede Menge Kontaktstifte in die Fußsohle. Sie sehen: Der Raspberry Pi braucht ein ordentliches Gehäuse.

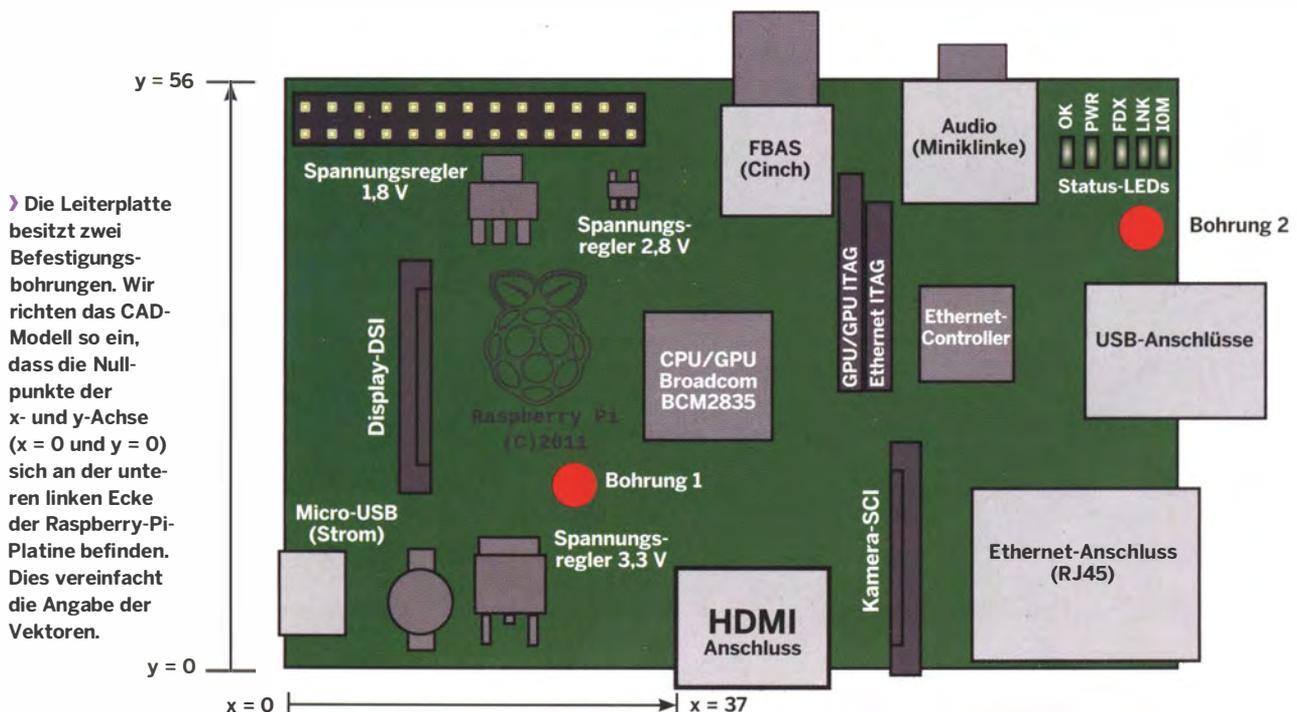
Es gibt natürlich etliche vorgefertigte Boxen für den Pi zu kaufen, aber wie wäre es, selbst eine herzustellen, und zwar mithilfe eines 3-D-Druckers? Um dieses Tutorial durchzuführen, benötigen Sie die kostenlose Software FreeCAD auf Ihrem Desktop-Computer sowie natürlich Zugang zu einem 3-D-Drucker. Falls Sie selbst keinen besitzen und auch niemanden kennen, gibt es Printdienste wie

[i.materialise.com](http://i.materialise.com) oder [shapeways.com](http://shapeways.com). Um das Pi-Gehäuse zu individualisieren, bräuchen Sie überdies die Software Inkscape, aber es geht auch ohne. Den Code und die sonstigen Dateien, die wir bei diesem Projekt benutzen, finden Sie auf [github.com/dejongh/raspberrypicase](https://github.com/dejongh/raspberrypicase).

Vergewissern Sie sich zunächst, ob Sie Modell A oder Modell B des Raspberry Pi haben. Das Modell B erkennt man leicht am doppelten USB-Anschluss. Unsere Anleitung ist für das Modell B maßgeschneidert, aber Sie können gegebenenfalls alle Abmessungen in der CAD-Datei leicht auf das Modell A anpassen. Als Nächstes müssen wir einige wichtige Komponenten des Geräts abmessen. Halten Sie auf der Platine nach den Befestigungsbohrungen Ausschau – es sollte zwei davon geben (siehe schematische Abbildung unten).

## Python-Code für FreeCAD

FreeCAD startet mit einem Begrüßungsbildschirm. Wählen Sie die Option „Part Design“. Um FreeCAD für Python-Code bereitzumachen, gehen Sie zu View > Views und haken Sie „Python Console“ an. Gehen Sie dann zu Edit > Preferences > General > Output window, haken Sie unten im Fenster die beiden „Redirect“-Optionen an, klicken Sie auf „Apply“ und



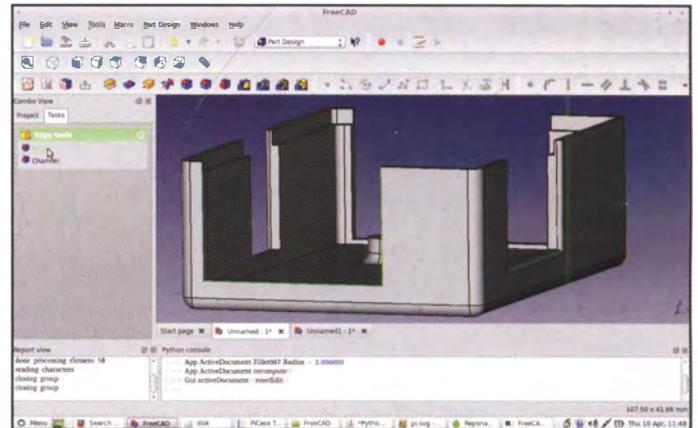
## Verschönern Sie das Gehäuse durch Rundungen

Das CAD-Modell sieht bisher noch nicht besonders schnuckelig aus, aber das war auch nicht anders zu erwarten. Zu einer professionelleren Gehäuseoptik fehlen nämlich noch ein paar Abrundungen. Dazu verwenden wir die Fillet-Funktion.

Bereits die Auswahl einer Kante und ein Klick auf den Fillet-Button erzielt eine sofortige Verschönerung. Halten Sie Strg gedrückt und wählen Sie mehrere Kanten aus, um die Grundplatte und die

Seitenwände der Box abzurunden. Stellen Sie einen Radius von 2,5 mm für die äußeren und 1,5 mm für die inneren Ecken ein. Sieht ganz gut aus, oder? Falls Sie nicht zufrieden sind, können Sie natürlich andere Werte einstellen.

› **Abgerundete Ecken geben dem Gehäuse einen professionellen Look.**



dann auf „OK“. Gehen Sie wieder zu View > Views und haken Sie „Report view“ an.

Nun müssen die benötigten Bibliotheken importiert werden. Tippen Sie Folgendes in die Python-Konsole (beachten Sie dabei die Groß- und Kleinschreibung):

```
import Part
from FreeCAD import Vector
```

Damit können wir jetzt das Part-Modul benutzen, um gängige 3-D-Formen wie Quader, Zylinder und Kugeln zu generieren, sowie das Vector-Modul, um die Formen anzuordnen. Werfen Sie nun einen Blick auf Ihren Raspberry Pi. Definieren wir die Ecke der Platine am Stromanschluss (Micro-USB) als Nullpunkt der x- und y-Achse und die Unterseite der Platine als Nullpunkt auf der z-Achse. Dies erleichtert uns die Eingabe von Vektoren, die den Maßen des Raspberry-Pi-Boards entsprechen. Nehmen Sie ein Lineal oder eine Schieblehre zur Hand, und messen Sie alle benötigten Entfernungen ab der Ecke links unten. Bestimmen Sie so die Position der beiden Bohrlöcher auf der x- und y-Achse.

```
Erstellen wir nun die grundlegende Quaderform:
boxshape=Part.makeBox(93,64,30,Vector(-4,-4,-8))
```

Wir haben hier einen Quader namens **boxshape** mit einer Länge/Breite/Höhe von 93 x 64 x 30 mm erzeugt und ihn so angeordnet, dass die Ecke der Platine bei 0 bleibt. Der Vorteil ist, dass wir alle weiteren Maße direkt eingeben können, ohne rechnen zu müssen.

```
cutout=Part.makeBox(87,58,30,Vector(-1,-1,-5))
```

Hier haben wir einen zweiten Quader namens **cutout** generiert, der rundum 3 mm kleiner als der andere ist, sodass wir ihn wie folgt aus **boxshape** herausschneiden können:

```
pibase=boxshape.cut(cutout)
```

Bevor das angezeigt wird, müssen wir jedoch FreeCAD noch sagen, dass er das Ergebnis zeichnen soll:

```
Part.show(pibase)
```

Rechtsklicken Sie ins Hauptfenster und wählen Sie Navigation styles > Blender navigation. Zoomen Sie mit der mittleren Maustaste heraus und halten Sie die Taste gedrückt, um die Ansicht zu drehen. Das Ergebnis sehen Sie in der Abbildung unten rechts.

Da dies die Basis unseres Raspberry-Pi-Gehäuses wird, sollte es etwas größer sein als die Leiterplatte, sagen wir

4 mm rundum. Außerdem brauchen wir Platz für die Stecker, die in die diversen Ports eingesteckt werden, darum lassen wir einige Aussparungen. Wir messen dafür wieder vom definierten Nullpunkt aus und tragen die Ergebnisse als Vektoren ein:

```
leftsidecut=Part.makeBox(20,42,30,Vector(-10,2,-4))
pibase=pibase.cut(leftsidecut)
```

Voilà, eine 42 mm breite Öffnung für das Stromkabel und die SD-Karte. Jetzt die anderen Aussparungen:

```
hdmicut=Part.makeBox(18,20,30,Vector(35,-10,-2))
avcut=Part.makeBox(33,20,30,Vector(38,50,0))
usbcut=Part.makeBox(20,38,30,Vector(80,1,-2))
pibase=pibase.cut(hdmicut)
pibase=pibase.cut(avcut)
pibase=pibase.cut(usbcut)
```

Und anzeigen lassen:

```
Part.show(pibase)
```

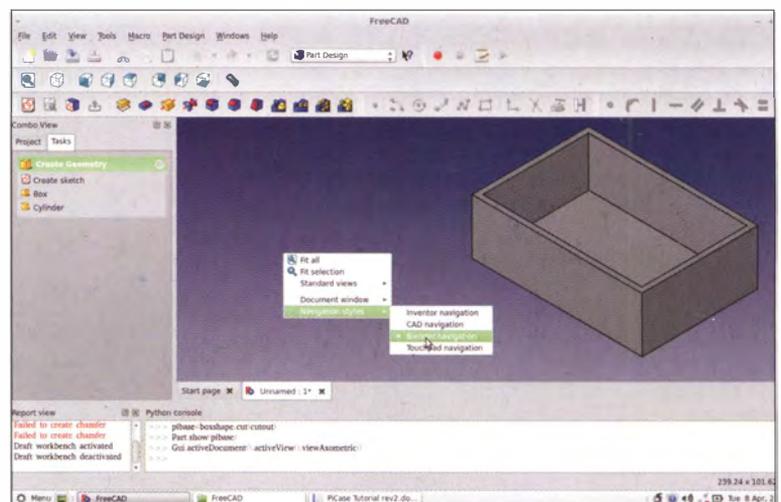
Gehen Sie zu Project > Shape > View und stellen Sie „Visibility“ auf „False“. Nun fehlt noch ein Deckel für die Pi-Box. Subtrahieren Sie die Deckelform von den Basiswänden:

```
lidcut=Part.makeBox(90,61,10,Vector(-2.5,-2.5,19))
```

```
pibase=pibase.cut(lidcut)
```

Um die Platine mit dem Boden des Gehäuses verschrauben zu können, brauchen wir Erhöhungen, wo die Schrauben

› **Die Benutzeroberfläche der Software FreeCAD kann so eingerichtet werden, dass man in einer Konsole am unteren Ende des Fensters Python-Skripte schreiben kann. Wählen Sie Navigation styles > Blender navigation für eine bequeme Navigation per Maus.**



reingedreht werden. Dazu verwenden wir den Befehl **part.makeCylinder**. Auch hier setzen wir wieder am Nullpunkt an, diesmal dem der z-Achse, also der unteren Fläche der Leiterplatte. Wie **makeBox** benötigt auch **makeCylinder** Werte, die die Größe des Körpers bestimmen. In diesem Fall sind das Radius und Höhe. Die Position der Bohrlöcher auf der x- und y-Achse hatten wir ja schon ausgemessen.

```
boss1=Part.makeCylinder(3.5,7,Vector(25.5,18,-7))
```

Die Höhe hat den Wert 7, während der Vektor für die z-Achse -7 beträgt. Somit sollte die Erhöhung genau an der Unterseite des Raspberry Pi beginnen. Machen Sie nun das Gleiche für den zweiten Schraubsockel:

```
boss2=Part.makeCylinder(3.5,7,Vector(80,43.5,-7))
```

Um die Erhebungen mit der Basis zu verbinden, benutzen wir den Befehl **fuse**, der wie cut verwendet wird:

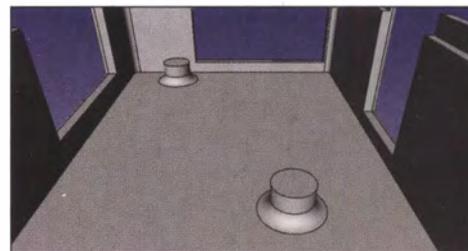
```
pibase=pibase.fuse(boss1)
```

```
pibase=pibase.fuse(boss2)
```

So, die Grundform steht! Allerdings sieht das alles noch recht schmucklos aus. Im Kasten oben auf Seite 159 geben wir Ihnen Tipps für ein schöneres Design des Gehäuses.

## Pi ans Gehäuse schrauben

Wir haben zylinderförmige Erhöhungen eingebaut, die als Schraubsockel für die Leiterplatte dienen. Wie Ihnen vielleicht aufgefallen ist, haben diese gar keine Löcher. Warum? Dies haben wir so gemacht, weil zum einen 3-D-Drucker für den Hausgebrauch eine zu geringe Druckauflösung haben, um so kleine Löcher zuverlässig zu realisieren, und wir zum anderen nicht hundertprozentig sicher sein können, dass das Loch im Sockel und



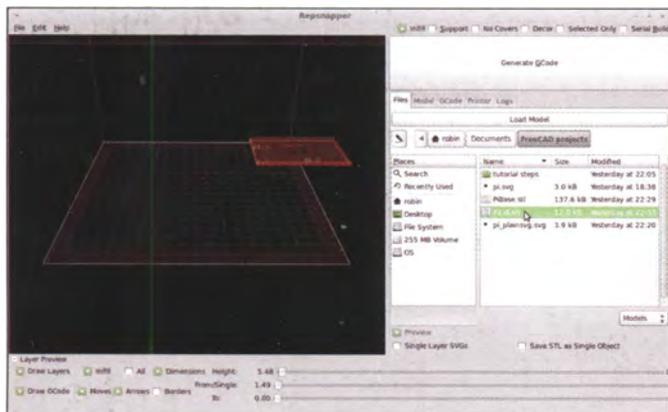
➤ An diese Sockel wird der Pi geschraubt.

das in der Pi-Platine ganz deckungsgleich sind.

Um die Schrauben präzise in die massiven Sockel hineindrehen zu können, arbeiten Sie am besten mit einer

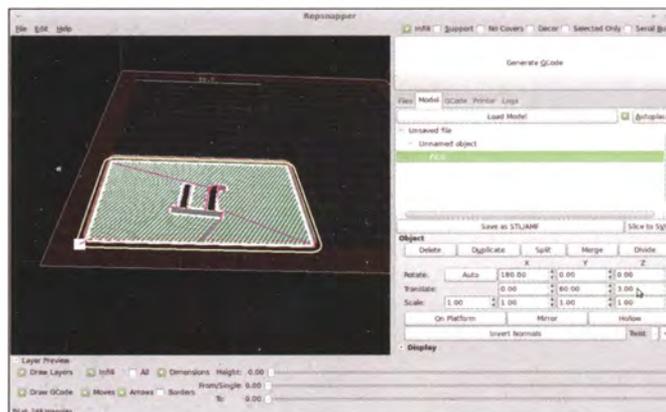
Bohrspitze oder einem Vorbohrer. Setzen Sie den Raspberry Pi in das Gehäuse ein und markieren Sie entsprechend die Mitte der Löcher der Pi-Platine auf den Sockeln.

## Vom Modell zum 3-D-Ausdruck



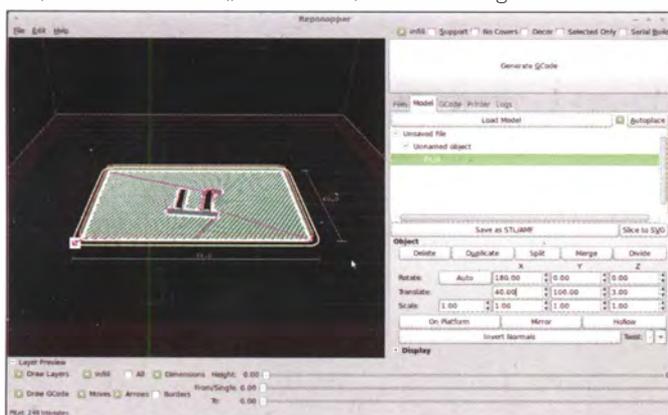
### 1 STL-Datei importieren

Die Dateien für Gehäuse und Deckel sollten jeweils als .stl-Dateien exportiert werden, sodass sie nun einzeln importiert und gedruckt werden können. Wählen Sie in der Printsoftware die Datei PiLid.stl aus, und klicken Sie auf „Load model“, um es anzuzeigen.



### 2 Druck ausrichten

Um auf dem Deckel ein schönes Finish hinzubekommen, platzieren Sie die Oberseite auf der Druckoberfläche und benutzen Sie die Heated-Bed-Funktion mit den Einstellungen Rotare X = 180 und Translate Z = 3.



### 3 Objekt zentrieren

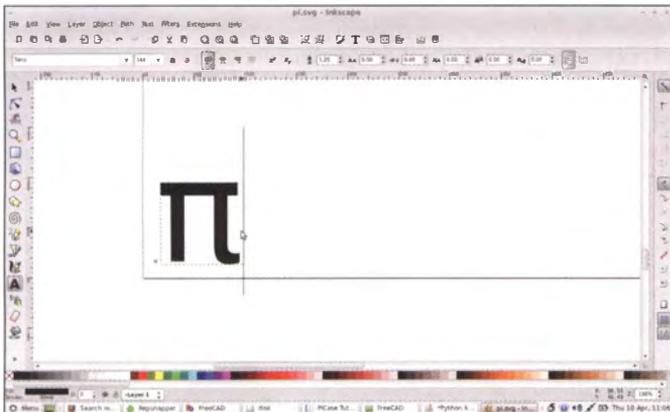
Zentrieren Sie mithilfe der Translate-Werte X und Y Ihr Objekt auf der Druckoberfläche. Nehmen Sie die Einstellungen gemäß den Empfehlungen Ihres 3-D-Druckers vor. Klicken Sie auf „Generate G-code“, um das Modell in Scheiben zu schneiden.



### 4 3-D-Drucker starten

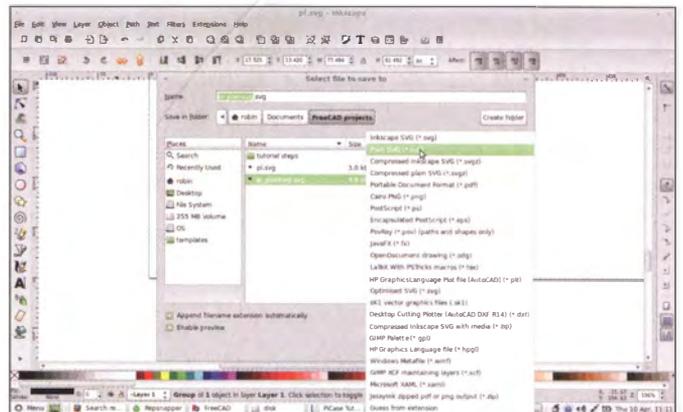
Nutzen Sie die Heated-Bed-Funktion, um Falten zu vermeiden und eine gute Adhäsion zu erzielen. Wenn Sie eine Glasplatte verwenden, bekommen Sie ein tolles Finish. Drucken Sie nach dem Deckel die Basis der Box, aber diesmal ohne die Drehung von Schritt 2.

# Gehäuse individuell gestalten



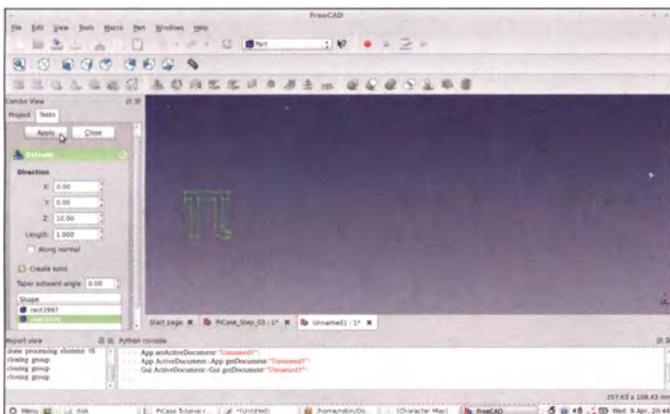
## 1 Text in Inkscape kreieren

Inkscape öffnen und links in der Werkzeugleiste das Text-Icon anklicken, dann ins Bearbeitungsfeld klicken. Text für die Deckelbeschriftung eingeben. Wir haben uns hier den griechischen Buchstaben Pi ausgesucht.



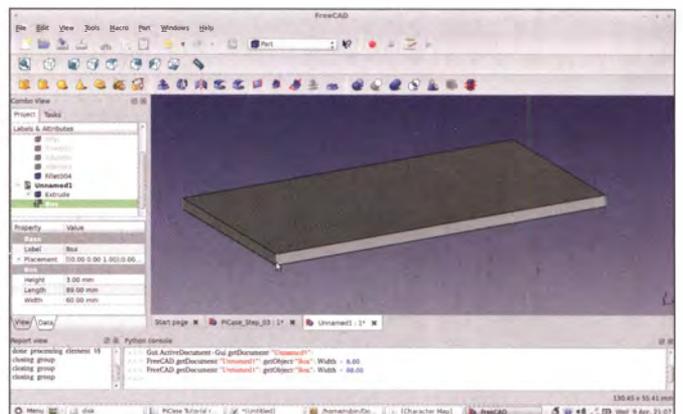
## 2 Text aus Inkscape exportieren

F1 drücken, dann Text markieren. Gehen Sie zu Path > Object to path, dann zu File > Save as, wählen Sie dabei „Plain SVG (.svg)“ als Dateiformat. Dann FreeCAD öffnen und die gerade gespeicherte Datei importieren. Wählen Sie beim Öffnen „SVG as Geometry“ aus.



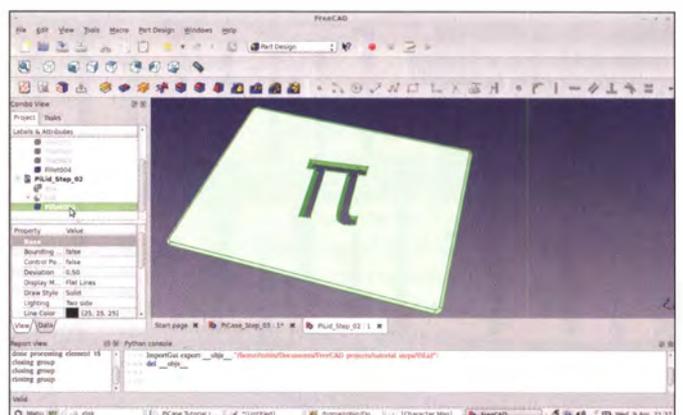
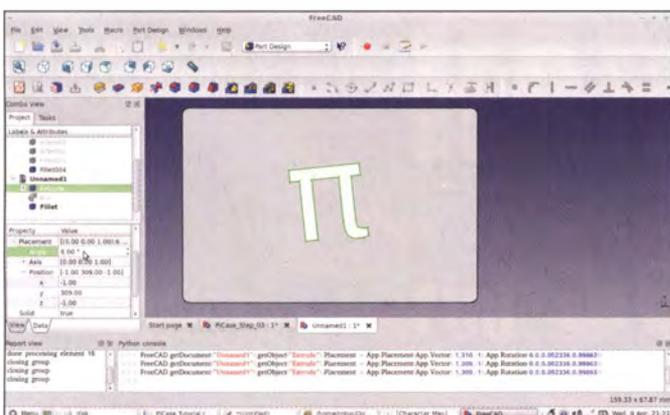
## 3 Vektorgrafik in FreeCAD importieren

Die Vektorgrafik wird als gruppiertes Objekt importiert. Wählen Sie dieses aus der Projektliste, dann wird es grün angezeigt. Stellen Sie sicher, dass Sie im Bereich „Part“ arbeiten, und klicken Sie dann auf „Extrude a selected sketch“. Stellen Sie Direction Z = 10 ein, wählen Sie dann „Create solid“ und „Apply“.



## 4 Individuellen Deckel designen

Klicken Sie den Button „Create a box solid“, stellen Sie Höhe, Länge und Breite wie im Bild gezeigt ein. Halten Sie – im Arbeitsbereich „Part“ – die Taste Strg gedrückt, um alle vier Ecken auszuwählen, weisen Sie sodann eine Rundung mit 2 mm Radius zu. Danach zur Projektliste gehen und die „Extrude“-Gruppe auswählen.



## 5 Buchstaben ausschneiden

Im Tab „Data“ zu Placement > Position gehen und den z-Wert auf -1 ändern. Dann so lange am x- und am y-Wert spielen, bis sich die Buchstaben an der gewünschten Position auf dem Deckel befinden. Mit „Angle“ können Sie Ihren Text gegebenenfalls noch neigen.

## 6 Zum 3-D-Drucker exportieren

Strg gedrückt halten, dabei „First Fillet“ und „Extrude“ wählen und „Cut“ klicken. Markieren Sie den Deckel in der Projektliste, alles sollte jetzt wieder grün angezeigt werden. Dann zu File > Export > Mesh formats. Wiederholen Sie die Schritte mit der Gehäusebasis.

# Wolfson Audiokarte

Geben Sie Ihrem Raspberry Pi anständige Audio-Fähigkeiten, indem Sie diese Erweiterungskarte einstecken und gute Musik aufnehmen.

## Abriss

» Eine Erweiterungskarte, die die Unterstützung für 24-Bit, 192-kHz-Aufnahmen, Stereo-Ausgabe, Headsets und Mikrofone nachrüstet. Sie besitzt auch einen S/PDIF-Ein- und Ausgang. Für Revision-2-Boards aufwärts.

## Daten

- » WM5102-Audiochip
- » 24-Bit-Hifi-Audioverarbeitung
- » Sechs ADSc, 96-dB-SNR-Mikrofoneingang
- » Sieben DACs, 113 dB Rauschabstand
- » Bis zu sieben A/D-Mikrofoneingänge
- » Zwei Stereo-Ausgänge
- » Sample-Rates bis 192 kHz

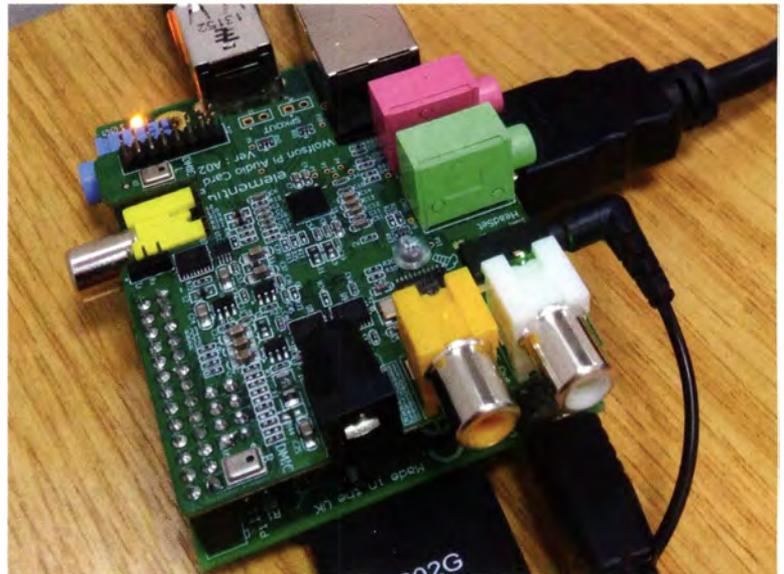
Der Raspberry Pi hat nur grundlegende Audio-Fähigkeiten, gibt über HDMI Sound bestenfalls in anständiger Qualität aus und verfügt über keine Aufnahme-funktion. Hier kommt die Wolfson Audiokarte ins Spiel, welche den WM5102-Audio-Chipset verwendet. Die Erweiterungskarte wird direkt auf den Raspberry Pi (Modell A oder B) aufgesteckt. Voraussetzung ist allerdings, dass es sich um Revision-2.0-Boards handelt. Das Board wird per GPIO angesteckt, nutzt aber auch die P5-Anschlüsse des Raspberry Pi. Es gibt auch einen Nylonbalken und Kunststoffschrauben, um das Ganze fest zusammenzuhalten. Das ist wichtig: Besteht keine gute Verbindung zwischen den Federkontakten, könnte das Board zwar aufleuchten, die Soundqualität aber schlecht sein.

Also, was bekommen Sie mit dem Board? Es gibt Line-in- und Ausgangs-Klinkenanschlüsse für Wiedergabegeräte mit eigener Stromversorgung sowie einen Mikrofonanschluss für VoIP oder Gaming-Chats. Auf der anderen Seite befinden sich zwei S/PDIF-Anschlüsse für Ein- und Ausgang. Falls die Energieversorgung des Boards nicht ausreichen sollte, gibt es auch hierfür den richtigen Anschluss.

## Kompatibilitätsprobleme

An dieser Stelle geben wir einige Hinweise. Der erste ist, dass die Karte nicht mit anderen Erweiterungen wie dem PiFace kompatibel ist. Zudem raten wir von der Verwendung von USB-Hubs ab. Das schmerzt besonders, wenn Sie den Raspberry Pi (Modell A) mit nur einem USB-Anschluss nutzen. Wir empfehlen auch, Geräte nicht anzuschließen, wenn der Pi gerade läuft.

Zudem braucht es für die Treiberinstallation eine neue Installation von Raspbian. Glücklicherweise gibt es ein vorbereitetes Abbild auf der Wolfson-Website, das allerdings doppelt so groß wie das Standard-Image ist. Da die Image-Datei über 7 GB groß ist, brauchen Sie eine SD-Karte mit mindestens 16 GB, wenn Sie zusätzlich etwas aufzeichnen möchten. Für Aufnahmen mit hoher Qualität wird eine SD-Karte der Klasse 6 empfohlen. Aber selbst mit



» Die Wolfson-Platine, angeschlossen an den Raspberry Pi und bereit für den Einsatz.

dieser könnten Sie in den Aufzeichnungen ein Stottern hören. Greifen Sie also gleich zu einer SD-Karte der Klasse 10.

Diese Version von Raspbian bootet direkt zu LXDE und besitzt einen Audio-Player mit fünf Beispiel-Audiodateien in hoher Qualität. Auch wenn diese nicht jedermanns Geschmack treffen dürften, ist die Wiedergabequalität über jeden Zweifel erhaben. Sobald Sie das Betriebssystem am Laufen haben, müssen Sie im Terminal noch einige Skripts ausführen, die mit der Karte mitgeliefert werden. Diese definieren die Signalwege für den Ton, je nachdem ob Sie aufzeichnen oder wiedergeben möchten. Es gibt auch ein Skript, um die Einstellungen wieder zurückzusetzen, falls Sie nicht möchten, dass über zwei Ausgänge gleichzeitig Ton ausgegeben wird. Insgesamt gibt es 10 Skripte. Und auch wenn Sie nicht alle davon brauchen werden, machen sie den Umgang mit der Platine angenehmer. Beachten Sie dabei, dass die Kommandos zwischen Groß- und Kleinschreibung unterscheiden.

Nachdem Sie alle Audiopfade festgelegt haben, können Sie entweder die beigelegten Audiodateien mit dem Music Player hören oder selbst Audio aufnehmen. Die Ausgabe vom Line-out ist hervorragend, aber der Mikrofonanschluss nahm entweder beide Kanäle,

nur einen oder gar keinen auf, abhängig davon, wie gedreht wir den Stecker einsteckten. Die Soundaufnahme in eine WAV-Datei funktionierte auf allen Schnittstellen sehr gut, aber auch hier reagierte der Mikrofonanschluss sehr feinfühlig auf den eingesteckten Stecker.

Um das Meiste aus den Aufzeichnungsfähigkeiten herauszuholen, ist noch eine gute Schnittsoftware notwendig. Trotzdem ist die Soundqualität überragend und das Board selbst recht günstig. Das macht es zu einer guten Erweiterung für den Raspberry Pi, mit der man viel Spaß haben kann. 🍷

## Fazit

### Wolfson Audiokarte

Entwickler: Wolfson Micro  
Web: element14.com  
Preis: ca. 30 €

Funktionen:	8/10
Leistung:	8/10
Simplexität:	6/10
Dokumentation:	9/10

» Eine mächtige Audio-Erweiterungskarte für einen günstigen Preis, aber mit einer seltsamen Einrichtung und Anwendungsweise.

Wertung **8/10**

PC Games Hardware – Das IT-Magazin für Gamer. Immer aktuell mit Kaufberatung, Hintergrundartikeln und Praxistipps.

# HARDCORE FÜR SCHRAUBER

**25**  
JAHRE  
**comptec**  
MEDIA  
LEIDENSCHAFT FÜR GAMES

QR-Code scannen  
und hinsurfen!



Neue Ausgabe jetzt am Kiosk erhältlich  
oder einfach online bestellen unter:  
[www.pcgh.de/shop](http://www.pcgh.de/shop)

Auch erhältlich als ePaper bei:



# TUXEDO COMPUTERS

## Hardware im Maßanzug

TUXEDO Computers sind individuell gebaute Computer und Notebooks die vollständig Linux tauglich sind, Windows natürlich auch, eben Hardware im Maßanzug :)

- » Assemblierung und Installation bei uns im Haus
- » Selbst programmierte Treiber & Scripte & Addons
- » Individueller Support & eigene Repositories
- » Angepasst für 100%ige Funktionalität aller Bestandteile:
  - + Sondertasten
  - + Helligkeitsverstellung
  - + Stand-By-Modus/Ruhezustand
  - + Energiesparfunktionen
  - + Flugmodus-Taste
  - + TRIM-Funktionen für SSDs, uvm.

Andere Betriebssysteme kann jeder, wir natürlich auch. Aber wir können auch Linux und das so, dass "einfach" alles funktioniert, alles!



### TUXEDO Book BU1503

- + Slim-Book, 15,6" matt Full-HD
- + bis 12 Std. Akkulaufzeit
- + Ultrabook-CPU's bis Intel Core i7
- + bis zu 3 HDD oder SSD
- + bis zu 16 GB RAM
- + DVD oder Blu-Ray Brenner

**ab 499 €**



### TUXEDO Nano

- + klein\*modular\*effizient
- + Energiespar-CPU's
- + bis Intel Core i7
- + VESA-Halterung
- + bis zu 3 HDD oder SSD
- + DVD oder Blu-Ray Brenner

**ab 349 €**