

Leseprobe

In diesem Buch werden alle Facetten des Raspberry Pi beschrieben. Diese Leseprobe macht Sie mit den ersten Schritten vertraut und zeigt Ihnen ein Bastelprojekt. Außerdem können Sie einen Blick in das vollständige Inhalts- und Stichwortverzeichnis des Buches werfen.

- »[Kauf und Inbetriebnahme](#)«
- »[Erste Schritte in Raspbian](#)«
- »[Arbeiten im Terminal](#)«
- »[Projekt: Der Raspberry Pi im Vogelhaus](#)«

- »[Inhalt](#)«
- »[Index](#)«
- »[Die Autoren](#)«
- »[Leseprobe weiterempfehlen](#)«

Michael Kofler, Charly Kühnast, Christoph Scherbeck

Raspberry Pi – Das umfassende Handbuch

1.064 Seiten, gebunden, in Farbe, September 2014
39,90 Euro, ISBN 978-3-8362-2933-3

www.galileo-press.de/3636

Kapitel 1

Kauf und Inbetriebnahme

Der Raspberry Pi ist ein winziger Computer. Seine Grundfläche ist etwas größer als eine Kreditkarte. In ein Gehäuse verpackt, hat der Computer das Volumen von zwei Smartphones. Das eigentliche Grundgerät kostet je nach Händler etwa 35 EUR. Zusätzlich brauchen Sie in der Regel ein Netzteil, ein Gehäuse, eine SD-Speicherkarte und eventuell ein paar Kabel. Die Gesamtinvestition liegt also deutlich unter 100 EUR.

Dafür erhalten Sie einen vollwertigen, Linux-basierten Computer mit einer ARM-CPU, den Sie zur Steuerung elektrischer Geräte, für Versuchsaufbauten, als Mini-Server oder als kleines Multimedia-Center in der Art des Apple TV einsetzen können. Prinzipiell kann der Raspberry Pi sogar als Ersatz für einen gewöhnlichen PC verwendet werden. Allerdings kann der Raspberry Pi hier, was die Geschwindigkeit betrifft, nicht mit modernen Rechnern mithalten. Viel Spaß macht es nicht, mit einem Webbrowser zu arbeiten, der bei jedem Scroll-Vorgang sekundenlange Pausen einlegt.

Dieses Kapitel gibt Tipps zum Kauf des Raspberry Pi samt des erforderlichen Zubehörs. Außerdem erfahren Sie, wie Sie auf Ihrem Notebook oder PC eine SD-Karte so einrichten, dass Sie diese als Betriebssystem für Ihren Raspberry Pi verwenden können. Sobald Sie diesen Schritt geschafft haben, können Sie Ihren Raspberry Pi erstmals starten und verwenden. Die ersten Schritte unter Raspbian, dem beliebtesten Betriebssystem für den Raspberry Pi, beschreibt dann das nächste Kapitel.

Gewissermaßen als Zuckerl für Linux-Experten enthält dieses Kapitel auch eine Anleitung, wie Sie Raspbian auf einen USB-Stick anstelle der SD-Karte installieren können. Der größte Vorteil dieser Vorgehensweise besteht darin, dass ein USB-Stick in der Regel zuverlässiger arbeitet als eine SD-Karte. Allerdings wird die Installation dadurch etwas komplizierter, weswegen Linux- bzw. Raspberry-Pi-Einsteiger zumindest vorerst von dieser Installationsvariante absehen sollten.

1.1 Kauf

Sofern Sie noch keinen Raspberry Pi besitzen, steht zuerst der Kauf an. Beachten Sie, dass Sie den Raspberry Pi ohne jedes Zubehör erhalten – es sei denn, Sie entscheiden sich für ein in der Regel überteuertes Komplettspaket! Zur Inbetriebnahme benötigen Sie deswegen auch ein Netzteil, eine SD-Karte, Tastatur und Maus mit USB-Anschluss, einen Monitor mit HDMI-Eingang sowie die dazugehörigen Kabel.

Bezugsquellen

Den Raspberry Pi sowie die gerade aufgezählten Zubehörteile können Sie unkompliziert im Internet erwerben. Neben Amazon und großen Elektronik-Händlern wie Conrad oder Pollin gibt es auch eine Menge kleinere Web-Shops, die sich auf Elektronikbastler und die sogenannte Maker-Szene spezialisiert haben. Beachten Sie beim Einkauf immer den jeweiligen Firmenstandort! Manche besonders günstige Angebote werden aus asiatischen Ländern versandt. Das kann nicht nur lange dauern, sondern auch zu Zollproblemen führen.

Raspberry-Pi-Modelle

Vom Raspberry Pi sind momentan (Stand: Sommer 2014) vier Modelle erhältlich:

- ▶ **Modell B:** Das Modell B ist der am weitesten verbreitete Raspberry Pi (siehe Abbildung 1.1). Bis zum Sommer 2014 wurden ca. 3 Millionen Stück davon verkauft. Das Modell hat zwei USB-2.0-Anschlüsse, einen 100-MBit-Netzwerkanschluss und verfügt über 512 MByte RAM. Die Rechenleistung stellt ein Broadcom BCM2835 System-on-a-Chip (SoC) zur Verfügung: Er besteht aus einem CPU-Core in ARMv6-Architektur mit 700 MHz sowie einem Broadcom Video-Core IV mit H.264 Encoder/Decoder. Die Leistungsaufnahme ohne Peripheriegeräte beträgt ca. 3,5 Watt.
- ▶ **Modell B+:** Dieses Modell wurde im Juli 2014 vorgestellt. Bei gleichbleibendem Preis unterscheidet es sich vom Modell B vordergründig dadurch, dass es vier statt zwei USB-2.0-Anschlüsse gibt und dass die GPIO-Leiste von 26 auf 40 Pins erweitert wurde (siehe Abbildung 1.2). Das bringt einige zusätzliche Steuerungsmöglichkeiten, wenn auch nicht ganz so viele, wie es den Anschein hat: Die Anschlüsse des so genannten P5-Headers auf der Rückseite des Modells A/B wurden nämlich in die GPIO-Leiste integriert und stehen nun einfach an einer anderen Stelle zur Verfügung.

Eine weitere wichtige Änderung betrifft die SD-Karte: Statt einer Speicherkarte in Standardgröße erwartet das Modell B+ nun eine Micro-SD-Karte. Da sich das Gesamtlayout der Platine und die Position der Stecker geändert hat, passt das Modell B+ nicht in Gehäuse für das Modell B.

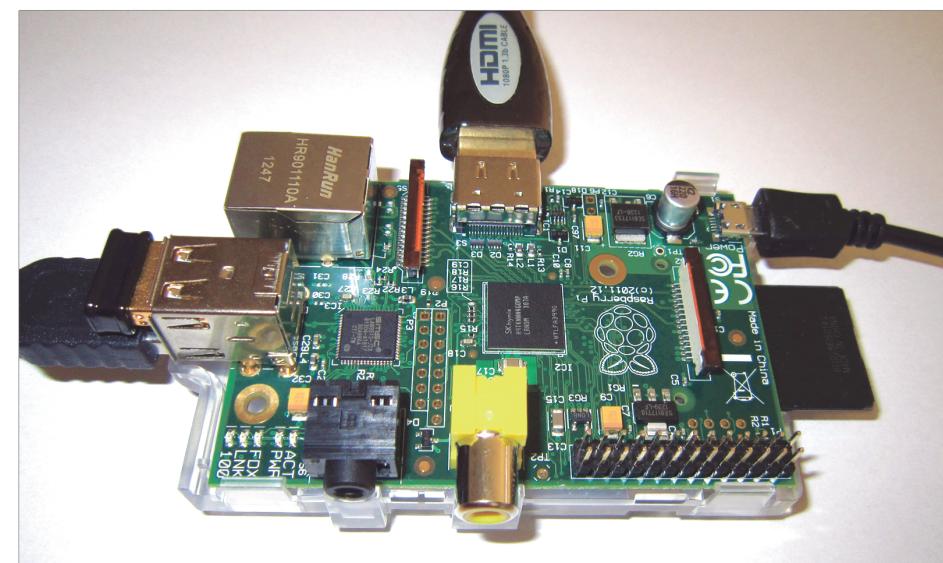


Abbildung 1.1 Ein Raspberry Pi Modell B mit einigen Kabeln und dem Unterteil eines Gehäuses

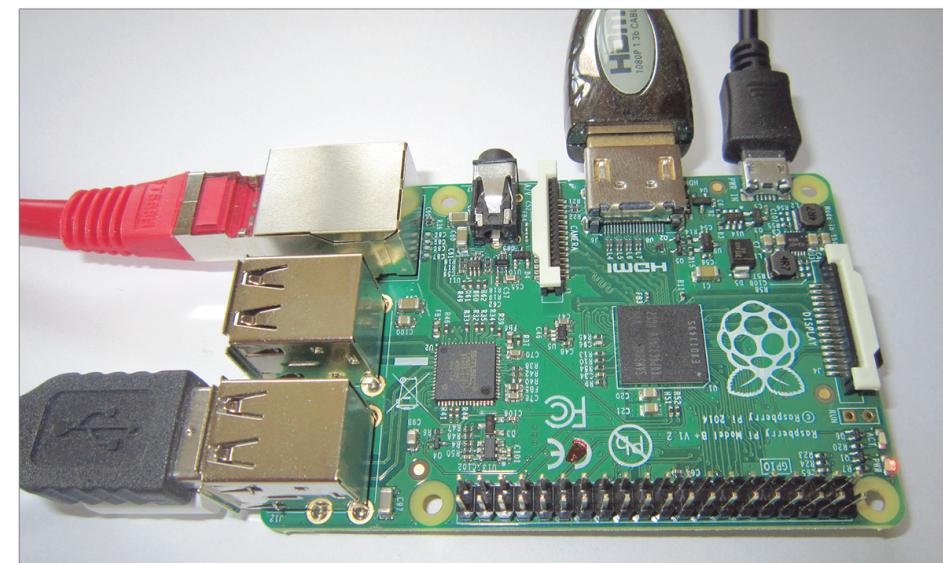


Abbildung 1.2 Zum Vergleich: ein Raspberry Pi Modell B+

Weniger offensichtlich sind die vielen Detailverbesserungen: So können die USB-Anschlüsse nun mit deutlich mehr Strom versorgt werden, sofern das Netzteil ausreichend dimensioniert ist. Gleichzeitig ist die Leistungsaufnahme des Grundgeräts ein wenig gesunken (3 Watt statt 3,5 Watt). Die Qualität des Analog-

Audio-Signals hat sich hörbar verbessert. Ganz verschwunden ist dafür der FBAS/Cinch-Stecker. Das Composite-Video-Signal wird nun über einen vierten Kontakt der Audio-Buchse geleitet. Seine Nutzung erfordert deswegen einen speziellen vierpoligen 3,5-mm-Adapterstecker.

- ▶ **Modell A:** Bei der abgespeckten Version des Raspberry Pi fehlt die Ethernet-Buchse. Außerdem hat der Minirechner nur einen USB-Anschluss und nur 256 MByte Arbeitsspeicher. Dafür ist dieses Modell ein wenig billiger. Außerdem ist die Leistungsaufnahme geringer (2,5 Watt).
- ▶ **Compute Module:** Bei dieser Raspberry-Pi-Variante wurde das gesamte Innenleben des Raspberry Pi auf einer deutlich kleineren Platine realisiert, die die Form eines DDR2-SODIMM-Speicherriegels hat und somit weniger als halb so groß wie der originale Raspberry Pi ist. Das Compute Module enthält standardmäßig einen 4 GByte großen Flash-Speicher und macht mehr Steuerungspins des BCM2835 zugänglich, bietet also mehr GPIOs (General Input Output). Wirklich genutzt werden kann dieser Raspberry Pi allerdings nur in Kombination mit einem I/O-Board, das die Anschlüsse nach außen führt. Das Compute Module ist vor allem für die industrielle Nutzung gedacht, z.B. wenn der Raspberry Pi zur Steuerung eines in hohen Stückzahlen produzierten Geräts verwendet werden soll.

In Planung ist darüber hinaus das Modell A+: Es soll wie Modell B+ durch mehr GPIO-Boards und eine etwas geringere Leistungsaufnahme punkten. Das Modell A+ soll aber weiterhin nur einen USB-Anschluss bieten und auf Ethernet ganz verzichten. Das Modell wird voraussichtlich Ende 2014 oder Anfang 2015 verfügbar sein.

Revision 1 versus Revision 2

Es gibt zwei Varianten (Revisionen) der Raspberry-Pi-Platine für die Modelle A und B: Revision 2 wird seit September 2012 ausgeliefert und hat eine viel größere Verbreitung. Soweit wir uns in diesem Buch auf das Modell B beziehen, ist immer die Revision 2 gemeint! Mit dieser Revision hat sich die Funktion von drei GPIO-Pins verändert. Für Elektronikprojekte ist es also wichtig zu wissen, welche Variante des Raspberry Pi Sie verwenden. Wie Sie die Revisionsnummer feststellen können, verrät Abschnitt 9.2.

Wir empfehlen Ihnen den Kauf des Modells B oder B+. Die Einschränkungen des Modells A rechtfertigen weder den geringeren Verkaufspreis noch den kleineren Stromverbrauch. Das Compute Module wiederum ist für Privatanwender und Bastler ungeeignet.

Die Entscheidung zwischen Modell B und B+ hängt primär davon ab, welche Peripheriegeräte Sie mit Ihrem Raspberry Pi verbinden wollen. Momentan (Stand Sommer 2014) gibt es einige Erweiterungs-Boards, die mechanisch nicht kompatibel zum

Modell B+ sind. Längerfristig wird sich aber vermutlich der gesamte Zubehörmarkt auf das Modell B+ umstellen – und dann gibt es eigentlich keinen Grund mehr, der für das ältere Modell B spricht.

USB-Mängel

Die vier USB-Anschlüsse des Modells B+ erfüllen sicher einen großen Wunsch vieler Raspberry-Pi-Anwender. Sie sollten sich aber darüber im Klaren sein, wie diese Anschlüsse technisch realisiert sind: Ein USB-Kanal, den der BCM2835 zur Verfügung stellt, führt zu einem internen Hub. Dieser ist dann mit den vier USB-Anschläßen und dem Ethernet-Anschluss verbunden. Mit anderen Worten: Alle USB-Geräte und der Ethernet-Anschluss teilen sich die Bandbreite eines USB-2.0-Kanals. Das gilt natürlich auch für das Modell B, nur stehen dort eben nur zwei USB-Anschlüsse zur Verfügung.

Von Äpfeln und Birnen, Himbeeren und Bananen

Man soll bekanntlich nicht Äpfel und Birnen vergleichen. An dieses Sprichwort dachten wir, also wir vom BananaPi hörten, der seit Frühjahr 2014 erhältlich ist. Das Gerät wird als kompatibel zum Raspberry Pi angepriesen, bietet aber gleichzeitig eine Menge Features, die sich viele auch für den Raspberry Pi wünschen: eine deutlich höhere CPU-Geschwindigkeit, 1 GByte RAM, einen echten GBit-Ethernet-Anschluss, einen SATA-Anschluss etc.

Erste Tests zeigten aber, dass das Gerät keineswegs kompatibel zum Raspberry Pi ist: Die Platine ist etwas größer, d.h., vorhandene Gehäuse passen nicht. Der Betrieb erfordert ein circa doppelt so leistungsfähiges Netzteil. Viele populäre Raspberry-Pi-Erweiterungen sind inkompatibel zum BananaPi: Das betrifft unter anderem die Kamera. Schließlich ist der BananaPi aufgrund einer anderen CPU auch nur eingeschränkt software-kompatibel. Beispielsweise laufen für den Raspberry Pi kompilierte Kernelmodule nicht auf dem BananaPi. Einen ausführlichen Vergleich können Sie hier nachlesen:

<http://blog.pi3g.com/2014/05/bananapi-erster-eindruck-und-vergleich-mit-dem-raspberry-pi>

Neben dem BananaPi gibt es mittlerweile unzählige Raspberry-Pi-Alternativen, jede mit individuellen Vor- und Nachteilen. Ungeschlagen ist der Raspberry Pi momentan in zwei Disziplinen: im Preis-Leistungs-Verhältnis und im Ökosystem. Aus unserer Sicht ist vor allem der zweite Punkt entscheidend: Für keinen anderen Minicomputer gibt es ein derart großes Angebot an Soft- und Hardware, an Online-Dokumentation und Foren, an Ideen und Bastelanleitungen!

Anschlüsse

Das Modell B bietet die folgenden Anschlussmöglichkeiten (siehe Abbildung 1.3):

- ▶ einen Micro-USB-Anschluss zur Stromversorgung (5 V, 700 mA, ergibt 3,5 Watt). Über diesen Anschluss können keine USB-Geräte betrieben werden, er dient ausschließlich zur Stromversorgung.
- ▶ zwei gewöhnliche USB-2.0-Anschlüsse für Tastatur, Maus und andere USB-Geräte mit einem maximalen Ausgangstrom von je 100 mA
- ▶ einen HDMI-Ausgang für Bild und Ton, Auflösung bis zu 1920×1200 Pixel
- ▶ einen Audio-Ausgang für einen 3,5-mm-Klinkenstecker
- ▶ einen Video-Ausgang (Composite Video, PAL oder NTSC)
- ▶ einen SD-Karten-Slot (SDHC)
- ▶ einen Ethernet-Anschluss (10/100 MBit)
- ▶ eine Steckerleiste mit 26 Pins (der sogenannte »P1-Header«) für allgemeine Zwecke (General Purpose Input/Output inklusive UART, I²C-Bus, SPI-Bus, I²S-Audio). Eine detaillierte technische Beschreibung der GPIO-Pins folgt in Kapitel 9.

Beim Modell A müssen Sie auf den Ethernet-Anschluss verzichten. Außerdem gibt es nur einen USB-Anschluss.

Beim Modell B+ ergeben sich im Vergleich zum Modell B die folgenden Änderungen (siehe Abbildung 1.4):

- ▶ vier statt zwei USB-2.0-Anschlüsse mit einem maximalen Ausgangstrom von insgesamt 600 mA, mit einer Zusatzoption sogar von 1200 mA (Hinweis: Der Betrieb energiehungriger USB-Geräte erfordert ein Netzteil mit 2 A bzw. 10 Watt sowie eine spezielle Option in /boot/config.txt, siehe auch Abschnitt 5.8.)
- ▶ ein Micro-SD-Slot (weiterhin SDHC)
- ▶ eine größere GPIO-Steckerleiste mit 40 Pins und dem neuen Namen »J8-Header«
- ▶ ein kombinierter Audio/Video-Ausgang für einen vierpoligen 3,5-mm-Klinkenstecker. Wenn das Video-Signal nicht genutzt werden soll, kann das Audio-Signal mit jedem handelsüblichen dreipoligen 3,5-mm-Klinkenstecker abgegriffen werden.
- ▶ geänderte Anordnung der USB-, HDMI- und Audio-Anschlüsse; erfordert daher ein speziell für das Modell B+ optimiertes Gehäuse

Allen Raspberry Pi-Modellen fehlt ein Ein/Aus-Schalter. Zum Einschalten stecken Sie das Micro-USB-Kabel zur Stromversorgung an. Um den Raspberry Pi auszuschalten, fahren Sie nach Möglichkeit zuerst das laufende Betriebssystem herunter, z. B. durch ABMELDEN im Startmenü oder mit dem Kommando halt. Anschließend lösen Sie das Micro-USB-Kabel vom Netzteil. Eine Anleitung, wie Sie Ihren Raspberry Pi über einen Taster ausschalten oder neu starten können, finden Sie in Abschnitt 38.3.

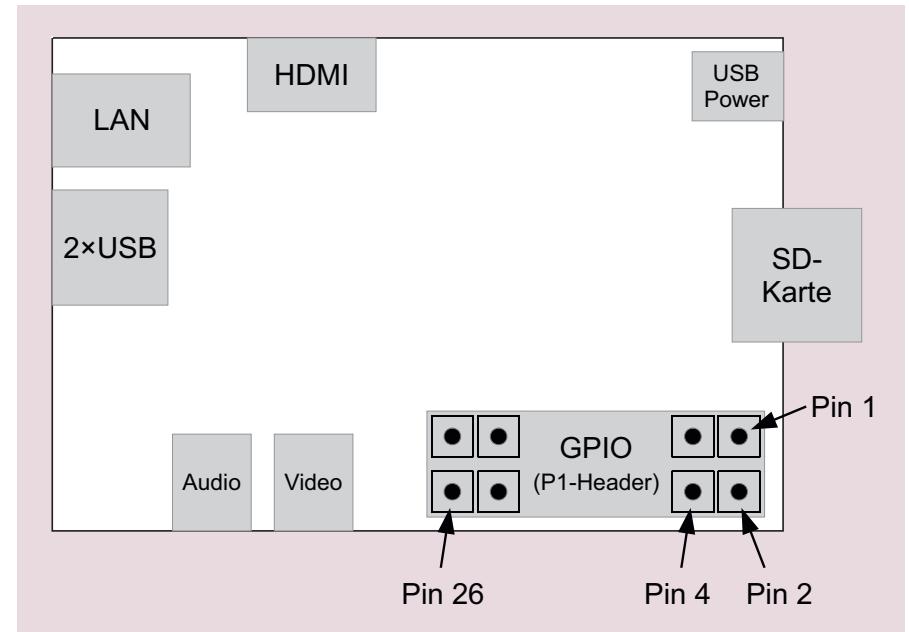


Abbildung 1.3 Schematische Darstellung der Raspberry-Pi-Anschlüsse (Modell B, Sicht von oben)

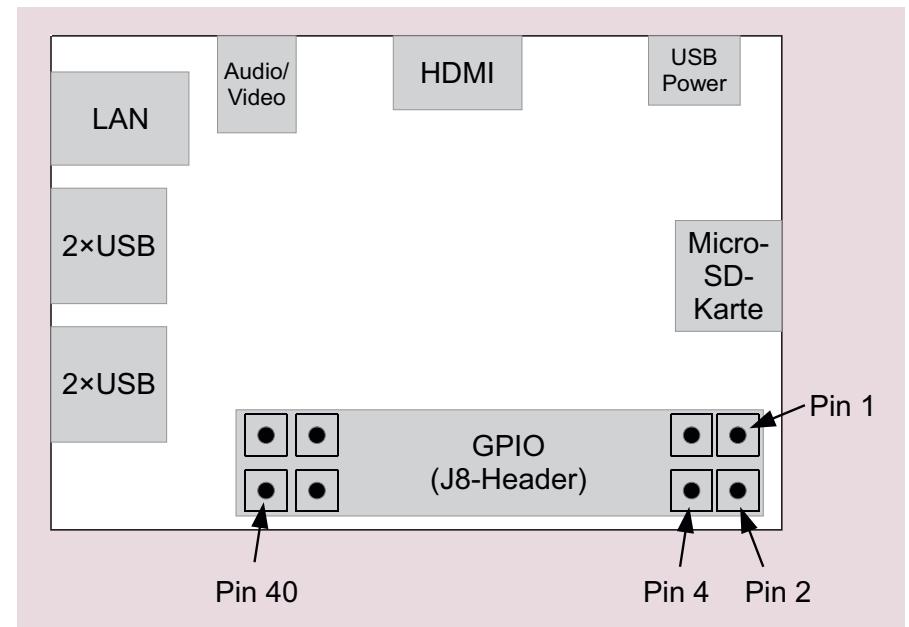


Abbildung 1.4 Schematische Darstellung der Raspberry-Pi-Anschlüsse (Modell B+, Sicht von oben)

Netzteil

Das Netzteil ist entscheidend dafür, dass der Raspberry Pi stabil und zuverlässig funktioniert. Achten Sie beim Kauf des Netzteils darauf, dass dieses ausreichend leistungsstark ist. Das Modell B des Raspberry Pi benötigt 700 mA Strom, das ergibt bei einer Spannung von 5 V eine Leistungsaufnahme von 3,5 W. Dazu kommt aber noch der Strombedarf von bis zu zwei USB-Geräten mit jeweils 100 mA sowie anderer Komponenten, die mit dem Raspberry Pi verbunden sind: So dürfen über GPIO-Pins bis zu 50 mA fließen. Wenn der Raspberry Pi über ein HDMI-Kabel mit einem Monitor verbunden ist, kostet das ca. 50 mA Strom. Das Kameramodul für den Raspberry Pi benötigt weitere 250 mA.

Beim Modell B+ sieht die Kalkulation an sich ähnlich aus, wenn auch mit einem etwas geringeren Grundstromverbrauch von ca. 600 mA. Ein wesentlicher Unterschied betrifft aber die vier USB-Anschlüsse: Über sie kann bei einer entsprechenden Konfiguration insgesamt bis zu 1200 mA Strom an energiehungreige Geräte weitergegeben werden, z. B. an externe Festplatten ohne eigenes Netzteil.

Sie benötigen also ein Netzteil, das zumindest 1000 mA (also 5 W) zur Verfügung stellen kann. Beim Modell B+ kann der Gesamtstrom sogar auf bis zu 2000 mA ansteigen (also 10 W), wenn Sie viele bzw. leistungsstarke USB-Geräte nutzen. Typische Handy-Netzteile sind ungeeignet, auch wenn diese vielleicht mit dem richtigen USB-Kabel ausgestattet sind und die Verlockung daher groß ist, das Netzteil einer neuen Verwendung zuzuführen!

Grundsätzlich ist der Raspberry Pi für den Dauerbetrieb ausgelegt. Viele Raspberry-Pi-Anwendungen setzen voraus, dass der Raspberry Pi Tag und Nacht läuft. Glücklicherweise verbraucht der Raspberry Pi dabei nur etwas mehr Strom als viele andere Geräte im Stand-by-Betrieb. Dennoch summiert sich der Strombedarf über ein Jahr gerechnet auf rund 35 bis 50 Kilowattstunden. Bei einem Strompreis von 10 Cent/kWh betragen die Stromkosten für den Raspberry Pi (ohne Zusatzgeräte) also rund 3 bis 5 Euro pro Jahr.

Akku- und Batteriebetrieb

Im Vergleich zu einem gewöhnlichen Computer verbraucht der Raspberry Pi zwar nur wenig Strom, für den Akku- oder Batteriebetrieb ist die Leistungsaufnahme aber dennoch recht hoch. Tipps, wie Sie Ihren Raspberry Pi zumindest etliche Stunden lang ohne Netzanschluss betreiben können, gibt Abschnitt 9.5.

SD-Karte

Der Raspberry Pi verfügt nicht wie ein gewöhnlicher Computer über eine Festplatte oder eine SSD. Stattdessen dient eine SD-Karte als Datenspeicher für das Betriebssystem sowie für Ihre Daten. Die Form der SD-Karte hängt vom Modell ab:

- ▶ **Modell A und B:** Die älteren Raspberry-Pi-Modelle erwarten eine SD-Karte im Standardformat. Mini- oder Micro-SD-Karten können nur mit einem Adapter verwendet werden.
- ▶ **Modell B+:** Das neuere Modell B+ erfordert eine Micro-SD-Karte. In der Regel ist es zweckmäßig, eine Micro-SD-Karte mit einem Adapter für das Standardformat zu erwerben. Den Adapter benötigen Sie, damit Sie die Micro-SD-Karte in den SD-Kartenslot Ihres gewöhnlichen Computers einführen und dort beschreiben können.

Unabhängig vom Format muss die SD-Karte dem SDHC-Standard entsprechen. Der neuere SDXC-Standard wird nicht unterstützt! Damit ist die maximale Größe auf 32 GByte limitiert.

Probleme mit SD-Karten

Wenn man den diversen Raspberry-Pi-Diskussionsforen trauen kann, sind defekte SD-Karten die häufigste Fehlerursache auf dem Raspberry Pi. Das hat sich auch bei unseren Tests bestätigt. Der Konkurrenzkampf der Anbieter hat SD-Karten in den letzten Jahren unglaublich billig gemacht. Darunter hat aber die Qualität gelitten. Leider gibt der Kauf einer etwas teureren SD-Karte keine Garantie dafür, ein hochwertigeres Produkt zu erhalten. Kurzum: Kümmern Sie sich regelmäßig um Backups Ihrer Daten, und halten Sie für den Notfall eine SD-Reservekarte bereit.

Auch wenn wir diesbezüglich keine negativen Erfahrungen gemacht haben, existieren offensichtlich auch vereinzelt SD-Karten, die inkompatibel zum Raspberry Pi sind und überhaupt nicht funktionieren. Diesbezügliche Informationen können Sie auf der folgenden Seite nachlesen. Dort finden Sie in einer Art Datenbank unzählige Erfahrungsberichte zu diversen SD-Karten:

http://elinux.org/RPi_SD_cards

SD-Karten gibt es in unterschiedlichen Geschwindigkeitsklassen – Class 4, 6 oder 10. Class 6 bedeutet beispielsweise, dass eine Schreibgeschwindigkeit von zumindest 6 MByte pro Sekunde garantiert wird. Das klingt gut, ist aber weniger als ein Zehntel dessen, was bei Festplatten üblich ist, von SSDs gar nicht zu sprechen! Wenn Sie also Wert auf einen zügigen Start des Raspberry Pi legen bzw. häufig größere Datenmengen lesen oder schreiben möchten, sollten Sie eine möglichst schnelle SD-Karte verwenden, also Class 10.

Bleibt noch die optimale Größe der SD-Karte zu klären: Wenn Sie Raspbian einsetzen möchten, also das gängigste Linux-System für den Raspberry Pi, dann beträgt das unterste Limit 2 GByte. Besser ist es, Sie entscheiden sich gleich für etwas mehr Speicherplatz, z. B. für 8 GByte. Dann haben Sie genug Platz für Zusatzpakete, eigene Daten etc.

Gehäuse

Für Versuchsaufbauten auf Ihrem Schreibtisch können Sie auf ein Gehäuse verzichten. Sollten Sie aber vorhaben, Ihren Raspberry Pi im Rahmen eines Projekts dauerhaft einzusetzen (beispielsweise als Multimedia-Center im Wohnzimmer), ist ein Gehäuse empfehlenswert.

Im Internet gibt es eine große Auswahl an Gehäusen für den Raspberry Pi. Beim Kauf müssen Sie unbedingt darauf Rücksicht nehmen, welches Raspberry-Pi-Modell Sie einsetzen. Gehäuse für die Modelle A und B sind nicht kompatibel zu Gehäusen für das Modell B+!

Achten Sie bei der Auswahl darauf, dass das Gehäuse Belüftungsschlitzte aufweist! Der Raspberry Pi läuft mangels Lüfter und anderer bewegter Teile vollkommen lautlos, produziert aber durchaus Abwärme. In einem Gehäuse ohne Luftzirkulation riskieren Sie ein vorzeitiges Ableben Ihres neuen Gadgets!

Sofern die Belüftung gewährleistet ist, benötigt der Raspberry Pi für den normalen Betrieb keine Kühlung. Es besteht allerdings die Möglichkeit, die CPU höher zu takten und damit die Geschwindigkeit des Raspberry Pi zu steigern (siehe Abschnitt 4.14). Sollten Sie sich dazu entschließen, ist es empfehlenswert, die CPU, den USB/LAN-Controller und den Spannungswandler mit passiven Kühlkörpern auszustatten. Diese leiten die Wärme besser ab.

Tastatur und Maus

Nahezu jede handelsübliche USB-Tastatur und Maus eignet sich als Eingabegerät für den Raspberry Pi. Bei den Raspberry-Pi-Modellen A und B besteht das einzige Kriterium darin, dass der Strombedarf jeweils nicht mehr als 100 mA betragen darf. Sollte das der Fall sein, müssen Sie die Geräte über einen aktiven USB-Hub mit dem Raspberry Pi verbinden (siehe den nächsten Abschnitt).

Möglicherweise fragen Sie sich, wie Sie herausfinden, wie groß die Leistungsaufnahme Ihrer Tastatur bzw. Maus ist. Letztlich können Sie das nur ausprobieren, ein entsprechend genaues Datenblatt steht leider selten zur Verfügung. Sollte Ihr Raspberry Pi nicht stabil laufen bzw. funktionieren Tastatur und Maus gar nicht, dann wird Ihre erste Zusatzinvestition ein aktiver USB-Hub sein.

Persönlich haben wir für unsere Experimente unter anderem eine Apple-Aluminium-Tastatur und eine preisgünstige Logitech-OEM-Maus verwendet. Die Apple-Tastatur hat den Vorteil, dass sie nicht nur wenig Strom benötigt, sondern gleichzeitig als passiver USB-Hub verwendet werden kann, an den Sie die Maus anschließen können. Das spart einen USB-Anschluss des Raspberry Pi.

Längerfristig können Sie den Raspberry Pi natürlich auch mit einer Bluetooth-Maus und -Tastatur steuern. Das erfordert aber nicht nur einen kleinen USB-Bluetooth-Adapter, sondern auch die vorherige Konfiguration der Bluetooth-Geräte. Tipps zur Bluetooth-Konfiguration folgen in Abschnitt 2.6.

USB-Hub

Wir haben es bereits erwähnt: Die älteren Modelle des Raspberry Pi verfügen über nur zwei USB-2.0-Anschlüsse und können USB-Geräten maximal 100 mA Strom zur Verfügung stellen – und auch das nur, wenn das Netzteil für den Raspberry Pi korrekt bemessen ist. Bei einer Spannung von 5 V ergibt sich daraus eine maximal zulässige Leistungsaufnahme von 0,5 W pro Gerät. Für viele USB-Geräte ist das zu wenig. Das gilt insbesondere für externe Festplatten, aber auch für manche Tastaturen und WLAN-Adapter.

Leider ist es nahezu unmöglich, den Strombedarf bzw. die Leistungsaufnahme eines USB-Geräts vor dem Kauf in Erfahrung zu bringen. Sie können den Erfahrungsberichten anderer Raspberry-Pi-Anwender vertrauen, eine wirklich zuverlässige Informationsquelle ist das aber nicht. Oder Sie probieren es einfach selbst aus: Wenn das USB-Gerät funktioniert und Ihr Raspberry Pi danach problemlos startet und über längere Zeit absturzfrei läuft, ist alles in Ordnung.

Beim Modell B+ ist die USB-Situation wesentlich entspannter: Sie können bis zu vier Geräte gleichzeitig anschließen und diese mit insgesamt bis zu 1200 mA Strom versorgen. Das setzt aber ein entsprechend großzügig dimensioniertes Netzteil voraus, denn der Raspberry Pi selbst benötigt ebenfalls mindestens 600 mA. Mit angeschlossenem HDMI-Monitor und einer Kamera braucht er sogar bis zu 900 mA.

Wenn Sie viele energiehungige USB-Geräte gleichzeitig verwenden möchten, führt an einem aktiven USB-Hub kein Weg vorbei. *Aktiv* bedeutet in diesem Zusammenhang, dass der USB-Hub über eine eigene Stromversorgung verfügt.

Stromversorgung per USB-Hub

Manche aktive USB-Hubs werden damit beworben, dass ihre Ausgänge ausreichend Leistung zur Verfügung stellen, um Smartphones oder Tablets aufzuladen – also mehr Leistung, als der USB-Standard eigentlich vorsieht. Derartige Geräte sind prinzipiell in der Lage, auch die Stromversorgung des Raspberry Pi zu übernehmen. Einzige Voraus-

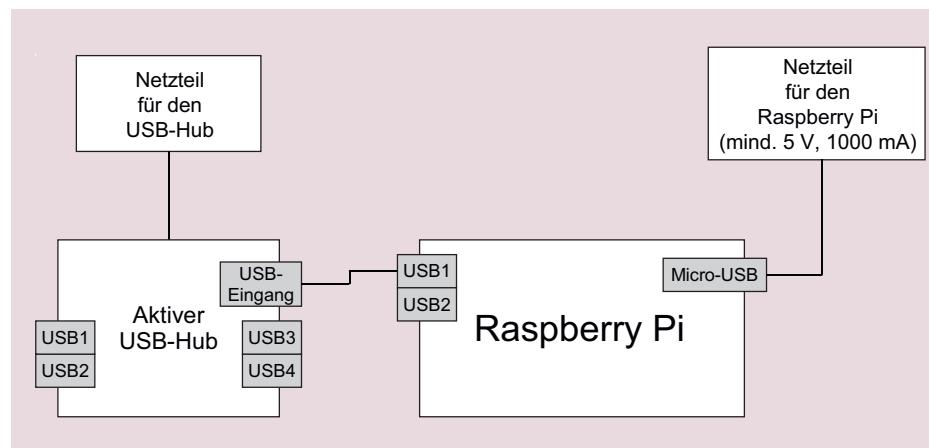


Abbildung 1.5 Raspberry Pi Modell B mit aktivem USB-Hub zur gleichzeitigen Verwendung von fünf USB-Geräten

setzung ist ein geeignetes Kabel, das auf der einen Seite einen Standard-USB-Stecker aufweist und auf der anderen einen Micro-USB-Stecker.

Soweit es Ihnen *nur* um die Stromversorgung des Raspberry Pi geht, ist dagegen nichts einzuwenden. Der betreffende USB-Ausgang des Hubs muss die erforderliche Leistung allerdings immer zur Verfügung stellen, d. h. entgegen dem USB-Standard ohne vorherige Kommunikation mit dem USB-Gerät.

Problematisch wird es aber, wenn Sie den USB-Hub gleichzeitig zur Stromversorgung *und* als Hub verwenden möchten. Dann benötigen Sie zwei Kabel zwischen dem Raspberry Pi und dem USB-Hub: eines für die Stromversorgung und ein zweites für die Datenübertragung. (Die Micro-USB-Buchse des Raspberry Pi ist ausschließlich zur Stromversorgung gedacht, die Datenleitungen bleiben dort ungenutzt!)

Diese doppelte Verkabelung sieht nicht nur unschön aus, sie birgt bei den Modellen A und B auch das Risiko, dass der Raspberry Pi nun über den gewöhnlichen USB-Stecker Strom bezieht. Genau dieser Fall, in der Fachsprache als Back-Powering bezeichnet, darf aber nicht eintreten – denn dann besteht die Gefahr von Datenverlusten auf der SD-Karte!

Abhilfe schaffen nur speziell für den Raspberry Pi konzipierte USB-Hubs, die dieses Back-Powering durch eine geeignete Schaltung verhindern. Wie der Schaltplan eines derartigen USB-Hubs aussehen muss, ist auf der folgenden Seite beschrieben:

http://elinux.org/RPi_Powered_USB_Hubs#USB_Hub_Power_Hookup

Keine Back-Powering-Gefahr besteht beim Modell B+: Dessen USB-Anschlüsse sind entsprechend abgesichert. Umgekehrt ist es beim Modell B+ unmöglich, einen normalen USB-Anschluss zu verwenden, um den Raspberry Pi mit Strom zu versorgen. Vorgesehen war diese Art der Stromversorgung zwar nie, bei den Modellen A und B funktioniert sie aber.

WLAN- und Bluetooth-Adapter

Wenn Sie Ihren Raspberry Pi per Funk in das lokale Netzwerk integrieren möchten oder Bluetooth-Geräte nutzen möchten, benötigen Sie einen USB-WLAN- bzw. USB-Bluetooth-Adapter. Diese winzigen USB-Stöpsel sind für wenige Euro erhältlich. Beachten Sie aber, dass nicht jedes Gerät kompatibel zum Raspberry Pi ist! Recherchieren Sie unbedingt vor dem Kauf, z. B. auf den folgenden Seiten:

http://elinux.org/RPi_USB_Wi-Fi_Adapters

http://elinux.org/RPi_USB_Bluetooth_adapters

Manche WLAN-Adapter, besonders solche mit einer größeren Antenne, benötigen mehr als 100 mA Strom und können daher nur über einen aktiven USB-Hub betrieben werden. Konfigurationsanleitungen finden Sie in den Abschnitten 5.5 und 5.6.

Was Sie sonst noch brauchen

Der Raspberry Pi ist zwar ein selbstständiger Computer, zur Inbetriebnahme benötigen Sie aber einen zweiten Computer: Dort laden Sie die Image-Datei mit dem Betriebssystem des Raspberry Pi herunter und übertragen das Image auf die SD-Karte. Dieser Vorgang wird im nächsten Abschnitt ausführlich beschrieben. Sollte Ihr Hauptcomputer über keinen SD-Slot verfügen, müssen Sie sich ein USB-SD-Kartenlesegerät besorgen, das Sie für wenige Euro in jedem Elektronik-Shop erhalten.

Auch für den weiteren Betrieb ist ein regulärer Computer hilfreich: Sobald auf Ihrem Raspberry Pi Linux läuft, können Sie die meisten Administrationsaufgaben auch über eine Netzwerkverbindung erledigen. Diese Vorgehensweise ist oft viel komfortabler als das direkte Arbeiten auf dem Raspberry Pi.

Für erste Experimente ist es ausgesprochen praktisch, wenn Sie Ihren Raspberry Pi mit einem Netzwerkkabel mit dem lokalen Netzwerk, also z. B. mit Ihrem ADSL-Router verbinden können. Damit hat Ihr Minicomputer ohne umständliche Konfigurationsarbeiten sofort Netzwerk- und Internetzugang. Das ist unter anderem auch deswegen zweckmäßig, weil der Raspberry Pi über keine batteriegepufferte Uhr verfügt und die aktuelle Uhrzeit aus dem Netzwerk bezieht. Mit der manchmal hakeligen WLAN-Konfiguration sollten Sie warten, bis Sie mit der Nutzung des Raspberry Pi vertraut sind.

Wenn Sie den Raspberry Pi für Elektronikprojekte einsetzen, benötigen Sie dazu natürlich die entsprechenden Bauteile, außerdem ein Multimeter, ein Steckboard für Versuchsaufbauten etc. Detaillierte Anleitungen für alle erdenklichen Einsatzzwecke folgen im dritten Teil dieses Buchs.

1.2 NOOBS-Installation

Die Installation eines Betriebssystems für den Raspberry Pi erfolgt anders als auf gewöhnlichen Computern: Der Raspberry Pi verfügt über kein CD/DVD-Laufwerk, das zur Installation verwendet werden könnte, und auch das Booten über einen USB-Stick samt Installationsprogramm ist nicht vorgesehen. Stattdessen müssen Sie die für den Raspberry Pi vorgesehene SD-Karte auf Ihrem regulären Notebook oder Desktop-Computer vorbereiten. Dazu gibt es zwei grundlegend unterschiedliche Vorgehensweisen: Entweder kopieren Sie die Dateien des in diesem Abschnitt beschriebenen NOOBS-Installationsprogramms direkt auf die SD-Karte, oder Sie laden sich eine sogenannte Image-Datei Ihrer Lieblingsdistribution herunter und schreiben diese auf die SD-Karte. Der Umgang mit Image-Dateien ist ein wenig komplizierter und wird im nächsten Abschnitt ausführlich erklärt.

Raspberry-Pi-Distributionen

Zuallererst müssen Sie sich für ein Betriebssystem entscheiden: Für den Raspberry Pi gibt es nämlich nicht nur eines, sondern es steht gleich eine ganze Menge von Betriebssystemen zur Auswahl. Nahezu alle Betriebssysteme basieren auf Linux. In der Linux-Welt ist es üblich, das eigentliche Betriebssystem sowie alle dafür verfügbaren Programme als *Distribution* zu bezeichnen. Die folgende Liste zählt die wichtigsten Distributionen auf, die für den Raspberry Pi geeignet sind:

- ▶ **Raspbian:** Raspbian ist die populärste Linux-Distribution für den Raspberry Pi. Das Wortgebilde *Raspbian* setzt sich aus Raspberry Pi und Debian zusammen. Fast alle Kapitel dieses Buchs beziehen sich auf Raspbian. Auch im Internet setzen fast alle Anleitungen und Tipps voraus, dass Sie Raspbian verwenden. Diverse Raspberry-Pi-Zusatzpakte stehen ausschließlich für Raspbian zur Verfügung (z.B. Mathematica) bzw. müssen beim Einsatz anderer Distributionen extra kompiliert werden. Für Raspberry-Pi-Einsteiger gibt es somit keinen plausiblen Grund, eine andere Distribution zu verwenden.
- ▶ **Pidora:** Pidora basiert auf der Fedora-Distribution. Fedora-Fans können so ihr Know-how unkompliziert auf den Raspberry Pi übertragen.

- ▶ **Arch Linux ARM:** Die Arch-Linux-Distribution richtet sich an fortgeschrittene Linux-Anwender. Die ARM-Version ist eine speziell für Minicomputer wie den Raspberry Pi und das BeagleBoard optimierte Arch-Variante.
- ▶ **RISC OS:** RISC OS ist ein Betriebssystem, das von der englischen Firma Acorn entwickelt und in seiner ersten Version 1987 freigegeben wurde. Das Betriebssystem ist vor allem für IT-Historiker interessant. Als einziges der hier genannten Betriebssysteme basiert es nicht auf Linux.
- ▶ **Volumio und Pi Musicbox:** Diese beiden Distributionen machen aus Ihrem Raspberry Pi einen Audio-Player für Ihre Stereoanlage. Beide Distributionen werden über einen Webbrowser bedient, z. B. auf dem Smartphone im WLAN zu Hause. Eine kurze Beschreibung finden Sie in Kapitel 6.
- ▶ **OpenELEC, Raspbmc, RasPlex und XBian:** Diese vier Distributionen sind speziell dazu gedacht, aus Ihrem Raspberry Pi ein Multimedia-Center zu machen. OpenELEC und RasPlex werden im Detail in den Kapiteln 7 und 8 beschrieben.

Eine eindrucksvolle Liste mit rund 40 für den Raspberry Pi geeigneten Distributionen finden Sie hier:

http://elinux.org/RPi_Distributions

Allerdings ist nicht jede der dort aufgeführten Distributionen so ausgereift wie Raspbian. Ein Teil der genannten Distributionen wird schon jetzt nicht mehr gewartet, und es ist zu befürchten, dass dies in Zukunft sogar für die Mehrheit der genannten Distributionen zutrifft. Wenn Sie sich für eine derartige Distribution entscheiden, werden Sie nicht mit Sicherheits-Updates und Bugfixes versorgt.

Warum gibt es kein Ubuntu für den Raspberry Pi?

Die populärste Linux-Distribution für Notebooks und PCs ist momentan Ubuntu. Warum gibt es also keine Ubuntu-Version für den Raspberry Pi? Das Problem besteht darin, dass Ubuntu momentan nur drei CPU-Architekturen unterstützt: 32- und 64-Bit-x86-Prozessoren sowie ARM-CPUs ab der Version 7. Die CPU des Raspberry Pi basiert aber auf der ARM-Version 6 und ist damit inkompatibel zu Ubuntu. Eine ausführliche Begründung, warum die Ubuntu-Entwickler nicht vorhaben, ARMv6 zu unterstützen, können Sie hier nachlesen:

<https://bugs.launchpad.net/ubuntu/+bug/848154>

SD-Karte formatieren

Bevor Sie auf Ihrem Notebook oder PC die Installationsdateien oder ein Image auf eine SD-Karte schreiben, müssen Sie die Karte formatieren. Das klingt nach einer trivialen Aufgabe, tatsächlich bereitet das Formatieren von SD-Karten aber überraschend viele Schwierigkeiten. Mit den Bordmitteln von Windows, OS X und Linux gehen Sie so vor:

- ▶ **Windows:** Unter Windows klicken Sie die SD-Karte (WECHSELDATENTRÄGER) im Windows Explorer mit der rechten Maustaste an und führen FORMATIEREN aus. Als Dateisystem verwenden Sie FAT32 (STANDARD).
- ▶ **OS X:** Unter OS X starten Sie das FESTPLATTENDIENSTPROGRAMM, wählen dort die SD-Karte aus, wechseln in das Dialogblatt LÖSCHEN und klicken dort auf den gleichnamigen Button. Als Dateisystem verwenden Sie MS-DOS-DATEISYSTEM (FAT).
- ▶ **Ubuntu Linux:** Unter Ubuntu Linux sowie bei Linux-Distributionen mit Gnome starten Sie das Programm LAUFWERKE (gnome-disks). Dort wählen Sie die SD-Karte aus, klicken dann auf den Zahnrad-Button und führen das Kommando FORMATIEREN aus. Das Listenfeld PARTITIONIERUNG belassen Sie in der Grundeinstellung KOMPATIBEL MIT ALLEN SYSTEMEN (MBR/DOS).
- ▶ **Andere Linux-Distributionen:** Natürlich können Sie die SD-Karte unter Linux auch in einem Terminalfenster neu formatieren. Dazu stellen Sie zuerst mit `mount` sicher, dass momentan keines der auf der SD-Karte befindlichen Dateisysteme verwendet wird. Gegebenenfalls lösen Sie diese Dateisysteme mit `umount` verzeichnis.

Außerdem müssen Sie den Device-Namen Ihrer SD-Karte feststellen. Dazu nehmen Sie das Kommando `lsblk` zu Hilfe. Es gibt einen Überblick über die Device-Namen aller Festplatten, SSDs, USB-Sticks und SD-Karten. Anhand der Größe lässt sich die SD-Karte in der Regel eindeutig ermitteln. Oft wird der Device-Name `sdb` oder `sdc` lauten, unter Umständen auch `mmcblk0`. Nach diesen Vorbereitungsarbeiten führen Sie drei Kommandos aus:

```
parted /dev/xxx mklabel msdos
parted /dev/xxx 'mkpart primary fat32 1MiB -1MiB'
mkfs.vfat -F 32 /dev/xxxxy
```

Mit dem ersten `parted`-Kommando erzeugen Sie eine neue Partitionstabelle auf der SD-Karte. Das zweite `parted`-Kommando legt eine Partition an. `mkfs.vfat` richtet darin ein FAT-Dateisystem ein. Bei den beiden `parted`-Kommandos geben Sie anstelle von `xxx` den Device-Namen der SD-Karte an. An `mkfs.vfat` übergeben Sie den Device-Namen der neuen Partition. Dieser lautet z.B. `/dev/sdc1` oder `/dev/mmcblk0p1`. Wenn Sie sich unsicher sind, rufen Sie vorher nochmals `lsblk` auf. Sollte

Linux das Kommando `parted` nicht kennen, installieren Sie vorher das gleichnamige Paket.

Mitunter treten beim Formatieren Probleme auf, insbesondere dann, wenn die SD-Karte bereits für eine Raspberry-Pi-Installation verwendet wurde und daher Linux-Partitionen enthält, die Windows- oder OS X-Rechner nicht erkennen. Viele Raspberry-Pi-Webseiten empfehlen deshalb, anstelle der Formatierwerkzeuge Ihres Betriebssystems das Formatierprogramm der *SD Association* (siehe Abbildung 1.6) einzusetzen. Dieses Programm steht für Windows und OS X auf den folgenden Seiten kostenlos zum Download zur Verfügung:

https://www.sdcard.org/downloads/formatter_4/eula_windows

https://www.sdcard.org/downloads/formatter_4/eula_mac

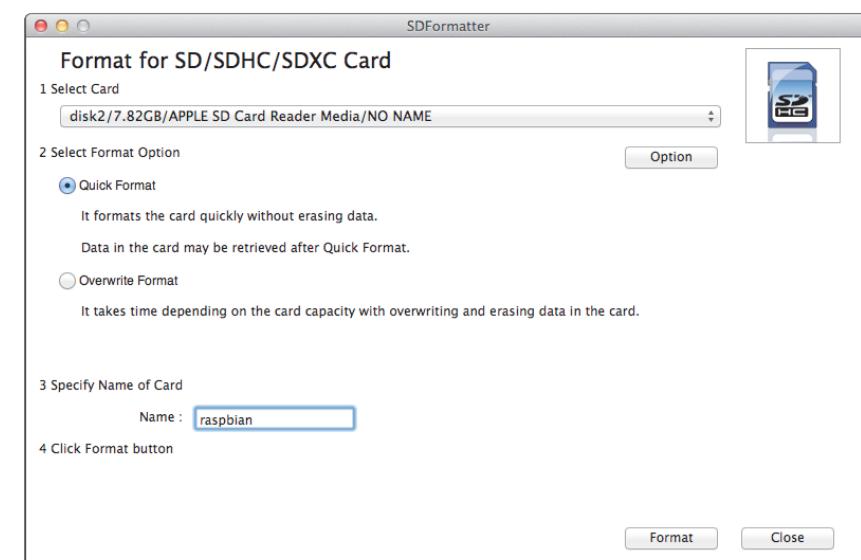


Abbildung 1.6 Das SDFormatter-Programm unter OS X

USB-SD-Card-Reader

Die meisten Notebooks besitzen einen Slot für SD-Karten in Standardgröße. Bei Micro-SD-Karten müssen Sie in der Regel einen SD-Kartenadapter verwenden, der bei vielen Micro-SD-Karten gleich mitgeliefert wird.

Sollten Sie Ihre SD-Karte auf einem Rechner formatieren bzw. beschreiben wollen, der über keinen Slot für eine SD-Karte verfügt, benötigen Sie einen SD-Karten-Reader. Mit diesen mitunter winzigen Geräten können Sie SD-Karten via USB ansteuern (siehe Abbildung 1.7).



Abbildung 1.7 Eine SD-Karte in Standardgröße, eine Micro-SD-Karte, ein SD-Karten-Adapter sowie ein winziger USB-Adapter für Micro-SD-Karten

Ärger mit dem Schreibschutz

Auf SD-Karten befindet sich am linken Rand ein winziger mechanischer Schalter, um den Schreibschutz zu aktivieren. Wenn der Schreibschutz aktiv ist und der SD-Slot diesen Zustand auch berücksichtigt, kann der Inhalt der Karte nur gelesen, aber nicht verändert werden. Naturgemäß kann die Karte dann auch nicht formatiert werden. Werfen Sie bei entsprechenden Fehlermeldungen also einen Blick auf diesen Schalter. Damit die SD-Karte verändert werden darf, muss sich der Schalter oben befinden, also in der Nähe der Kontaktleiste.

Einer unserer Testrechner, ein iMac von Apple, hatte offensichtlich Probleme mit der korrekten Erkennung dieses Read-only-Schalters. Immer wieder meldete das Festplattendienstprogramm, die SD-Karte könne nur beschrieben, aber nicht verändert werden, obwohl sich der Read-only-Schalter an der richtigen Position befand. Abhilfe: Nach mehrmaligem Ein- und Ausstecken der SD-Karte ist es uns letztlich immer wieder gelungen, die offensichtlich defekte Mechanik des Rechners zu überlisten.

NOOBS

NOOBS (*New Out Of Box Software*) ist keine Raspberry-Pi-Distribution, sondern vielmehr eine Sammlung von Installationsdateien, die auf eine leere, vorher formatierte SD-Karte kopiert werden. Beim ersten Start von der SD-Karte können Sie eine oder mehrere Distributionen auswählen und installieren. Momentan umfasst die Palette Raspbian, Arch, Pidora, RISC OS, OpenELEC und Raspbmc.

NOOBS richtet sich speziell an Raspberry-Pi-Einsteiger. Der größte Vorteil von NOOBS besteht darin, dass es sich nicht um eine Image-Datei handelt. Das vereinfacht das Einrichten der SD-Karte erheblich. NOOBS kann als ZIP-Datei von der folgenden Webseite heruntergeladen werden:

<http://www.raspberrypi.org/downloads>

Beim Download haben Sie die Wahl zwischen zwei Versionen:

- ▶ **Offline-Version:** Die rund 1,4 GByte große Offline-Version enthält alle Installationsdateien für sechs Raspberry-Pi-Distributionen. Die eigentliche Installation auf dem Raspberry Pi kann ohne Netzwerkverbindung durchgeführt werden. Da der Inhalt des Installationsprogramms in Form einer Recovery-Partition auf der SD-Karte verbleibt, sollten Sie die Offline-Version nur verwenden, wenn Ihre SD-Karte zumindest 8 GByte groß ist.
- ▶ **Netzwerk-Version:** Die Netzwerkversion umfasst nur bescheidene 20 MByte. Das reicht gerade aus, um den Raspberry Pi zu booten und ein Menü anzuzeigen, über das man die Installation einer Raspberry-Pi-Distribution starten kann. Die erforderlichen Installationsdateien werden danach aus dem Internet heruntergeladen. Das funktioniert allerdings nur dann, wenn Sie Ihren Raspberry Pi mit einem Netzwerkkabel an das lokale Netzwerk anschließen.

Egal für welche Variante Sie sich entscheiden, müssen Sie nun die SD-Karte formatieren und dann alle Dateien aus der heruntergeladenen ZIP-Datei auf die SD-Karte kopieren. Stellen Sie sicher, dass die Dateien `recovery.*` direkt auf der SD-Karte gespeichert werden, nicht in einem Unterverzeichnis! Denken Sie daran, im Dateimanager die SD-Karte per Kontextmenü auszuwerfen, bevor Sie die SD-Karte aus dem Slot entfernen.

Nach diesen Vorbereitungsarbeiten schließen Sie Ihren Raspberry Pi an einen Monitor an, verbinden Maus und Tastatur und stecken die SD-Karte mit den Kontakten nach oben in den SD-Slot. Wenn möglich, verbinden Sie den Raspberry Pi außerdem über ein Netzwerkkabel mit einem Switch/Hub im lokalen Netzwerk.

Erst nachdem Sie alle anderen Kabel verbunden haben, stecken Sie auch das Micro-USB-Kabel der Stromversorgung an. Auf dem Bildschirm sollte nun für circa zwei Sekunden ein buntes Quadrat angezeigt werden. Wenige Sekunden später erscheint das NOOBS-Fenster, in dem Sie die Sprache, das Tastaturlayout und das zu installierende Betriebssystem auswählen (siehe Abbildung 1.8).

In der Regel werden Sie im NOOBS-Menü nur den ersten Eintrag, RASPBIAN, auswählen. Fortgeschrittene Linux-Anwender werden vielleicht an der Zusatzoption DATA PARTITION Gefallen finden: Ist diese Option aktiv, dann richtet NOOBS während der Installation eine zweite, 512 MByte große Partition mit einem ext4-Dateisystem ein. Diese Dateisystem können Sie dann z. B. als Datenspeicher verwenden, der unabhängig von der Systempartition ist. Die Nutzung dieser Partition erfordert allerdings etwas Linux-Know-how. Außerdem verringert sich die Größe der Systempartition um ein halbes GByte.

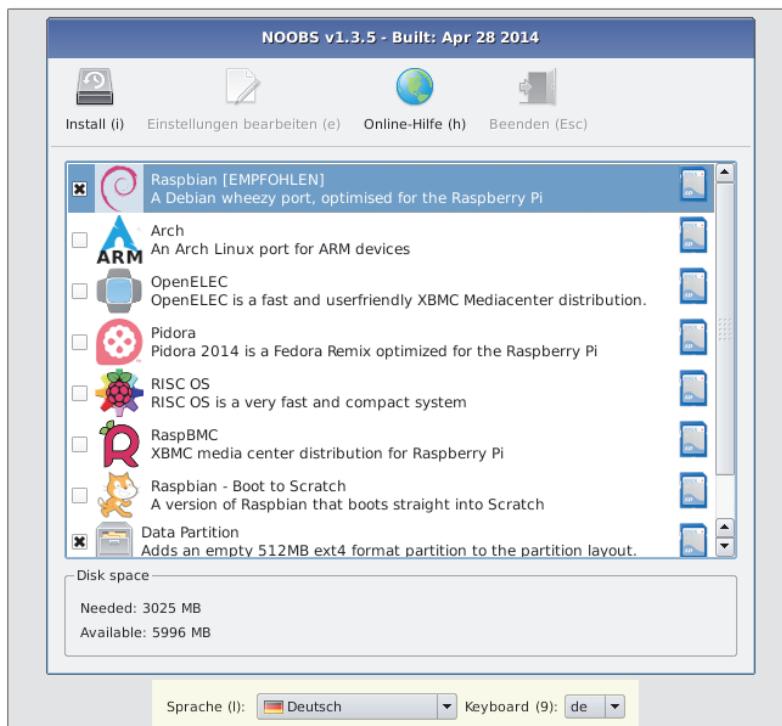


Abbildung 1.8 Das NOOBS-Installationsprogramm

Grundsätzlich ist es auch möglich, mehrere Distributionen auf einmal zu installieren. In diesem Fall erscheint jedes Mal beim Start des Raspberry Pi ein Boot-Menü, in dem Sie das zu startende Betriebssystem auswählen. Parallelinstallation haben allerdings den Nachteil, dass sich alle Betriebssysteme den Platz auf der Festplatte teilen. Es ist nachträglich nicht ohne Weiteres möglich, ein Betriebssystem zu entfernen und den freien Platz einem anderen Betriebssystem zuzuweisen. Deswegen raten wir Ihnen von Mehrfachinstallationen ab. Wenn Sie ein anderes Betriebssystem ausprobieren möchten, ist es besser, dafür eine zweite oder dritte SD-Karte zu verwenden.

Mit dem Button **INSTALL** starten Sie nun die Installation. Während der Installation, die für Raspbian circa eine viertel Stunde dauert, zeigt das Installationsprogramm eine Art Dia-Show mit Informationen und Tipps zum ausgewählten Betriebssystem aus (siehe Abbildung 1.9).

Nach Abschluss der Installation erscheint auf dem Bildschirm die Nachricht OS(es) INSTALLED SUCCESSFULLY. Sobald Sie diese Meldung mit OK bestätigen, wird der Raspberry Pi neu gestartet.



Abbildung 1.9 Statusanzeige während der Installation

Sofern Sie Raspbian als Betriebssystem ausgewählt haben, dauert der erstmalige Start relativ lange, weil dabei einige einmalige Initialisierungsarbeiten erledigt werden müssen. Anschließend erscheint das Programm `raspi-config`. Es hilft bei einigen grundlegenden Einstellungen. Sie können nun das Passwort für den Benutzer `pi` neu einstellen (standardmäßig gilt `raspberry`), die Sprache und das Tastaturlayout ändern etc. Bei Bedarf können Sie das Programm jederzeit mit dem Kommando `raspi-config` neuerlich ausführen, um Änderungen an der Konfiguration durchzuführen. Im Detail wird `raspi-config` in Kapitel 2 beschrieben. Dieses Kapitel stellt Ihnen außerdem Raspbian näher vor und hilft Ihnen bei den ersten Schritten.

Wenn etwas schiefgeht

Sollte die Installation aus irgendeinem Grund scheitern, können Sie jederzeit von vorne beginnen. Sie schalten also den Raspberry Pi aus, stecken die SD-Karte wieder in Ihr Notebook oder in Ihren PC ein, formatieren die Karte neu und kopieren dann nochmals den Inhalt der NOOBS-ZIP-Datei dorthin.

Schwieriger wird es, wenn es Hardware-Probleme gibt, d. h., wenn Sie z. B. kein stabiles Bild auf dem Monitor sehen oder Ihr Raspberry Pi während der Installation abstürzt. Für solche Fälle bietet Abschnitt 4.15 Hilfestellungen. Drei Tipps gleich vorweg: Stellen Sie sicher, dass die Stromversorgung ausreichend ist; probieren Sie es mit einer anderen SD-Karte; und verwenden Sie einen aktiven USB-Hub zum Anschluss von Tastatur und Maus.

NOOBS-VNC-Installation

Sollten Sie keinen Monitor für Ihren Raspberry Pi haben, erlaubt NOOBS sogar eine Installation, bei der Sie das Installationsprogramm via VNC bedienen. Virtual Network Computing (VNC) ist ein Programm, mit dem der Bildschirminhalt über eine

Netzwerkverbindung freigegeben wird. Damit können Sie ein Programm – in unserem Fall das NOOBS-Installationsprogramm – auf einem anderen Rechner steuern.

Um VNC zu nutzen, stecken Sie die SD-Karte zuerst in Ihr Notebook oder Ihren PC. Dort laden Sie die Datei `recovery cmdline` in einen Texteditor und fügen am Ende der einzigen Zeile dieser Datei die Option `vncinstall` ein. Nachdem Sie die Datei gespeichert und die SD-Karte ordnungsgemäß freigegeben haben, stecken Sie diese in den Raspberry Pi und starten ihn.

Auf Ihrem Arbeitsrechner starten Sie außerdem einen VNC-Client. Das Kunststück besteht nun darin, die IP-Adresse zu erraten, die das NOOBS-Installationsprogramm für seinen VNC-Server verwendet. Manche VNC-Clients bieten die Möglichkeit, das lokale Netzwerk nach VNC-Servern zu durchsuchen. Wenn der Raspberry Pi mit einem ADSL-Router verbunden ist, können Sie in der Regel auch auf dessen Administrationsseiten nachsehen, welche IP-Adresse dem Raspberry Pi zugewiesen wurde.

Auch die VNC-Installation ändert nichts daran, dass Sie für den weiteren Betrieb des Raspberry Pi – zumindest für die erstmalige Konfiguration – einen Monitor brauchen. Sobald Raspbian oder eine andere Raspberry-Pi-Distribution einmal zufriedenstellend läuft, können Sie einen SSH- oder VNC-Server einrichten und den Raspberry Pi in der Folge wieder über eine Netzwerkverbindung steuern. Im Detail sind die Vorgehensweisen in den Abschnitten 4.3 und 4.4 beschrieben.

1.3 Image-Datei auf eine SD-Karte schreiben

Das im vorigen Abschnitt beschriebene NOOBS-Konzept besteht darin, dass Sie zuerst einige Dateien auf eine SD-Karte schreiben. Der Raspberry Pi kann diese Dateien ausführen und dann im zweiten Schritt das Betriebssystem installieren. Diese Vorgehensweise ist einfach, hat aber zwei Nachteile: Zum einen dauert der Installationsprozess länger als notwendig und zum anderen ist ein Teil der SD-Karte auch nach der Installation blockiert, weil die Installationsdateien auf einer Recovery-Partition verbleiben.

Aus diesem Grund stellen viele Raspberry-Pi-Projekte ihre Distributionen in Form sogenannter Image-Dateien zur Verfügung. Eine Image-Datei ist eine blockweise Kopie der Daten, die sich auf einer SD-Karte befinden. Die Image-Datei enthält mehrere Partitionen sowie die darauf befindlichen Dateisysteme. Entscheidend ist nun, dass Sie nicht die Image-Datei als solche in das Dateisystem der SD-Karte kopieren. Vielmehr müssen Sie den *Inhalt* der Image-Datei blockweise auf die SD-Karte schreiben. Das können Sie nicht im Dateimanager Ihres Betriebssystems machen; vielmehr benötigen Sie dazu ein Spezialprogramm.

In diesem Abschnitt stellen wir Ihnen entsprechende Programme für Windows, OS X und Linux vor.

Image-Dateien herunterladen

Woher bekommen Sie die erforderliche Image-Datei? Für einige ausgewählte, besonders populäre Raspberry-Pi-Distributionen, unter anderem für Raspbian, Pidora, OpenElec und Raspbmc, finden Sie auf der folgenden Webseite Download-Links für Image-Dateien:

<http://www.raspberrypi.org/downloads>

Für alle anderen Distributionen müssen Sie auf der jeweiligen Projektseite nach der Image-Datei suchen. Die Image-Dateien sind häufig in eine ZIP-Datei verpackt. Sie müssen also das ZIP-Archiv entpacken. Üblicherweise erkennen Sie die Image-Datei an der Kennung `.img`.

Bei manchen Distributionen gibt es ähnlich wie bei NOOBS zwei Varianten der Image-Datei: Das oft deutlich größere Image enthält die komplette Distribution und ist für Offline-Installationen geeignet. Die kleinere Variante enthält hingegen nur das Grundgerüst der Distribution. Der verbleibende Rest wird beim ersten Start aus dem Internet heruntergeladen. Welche Variante für Sie besser ist, hängt davon ab, ob Ihr Raspberry Pi eine Netzwerkanbindung per Kabel hat. In diesem Fall können Sie dem kleineren Image den Vorzug geben.

Image-Datei unter Windows auf eine SD-Karte übertragen

Unabhängig davon, ob Sie unter Windows, OS X oder Linux arbeiten, sollten Sie auf jeden Fall zuerst die SD-Karte formatieren. Theoretisch wäre das gar nicht notwendig: beim Schreiben der Image-Datei werden ohnedies die Partitionstabelle und alle auf der SD-Karte befindlichen Dateisysteme überschrieben. In der Praxis hat sich aber gezeigt, dass Image-Writer viel seltener Probleme verursachen, wenn die SD-Karte leer und frisch formatiert ist.

Der populärste Image-Writer für Windows heißt *Win32 Disk Imager* (siehe Abbildung 1.10) und kann von der folgenden Webseite kostenlos heruntergeladen werden:

<http://sourceforge.net/projects/win32diskimager>

Das Programm wird nach den Windows-üblichen Rückfragen installiert. Der vom Installationsprogramm angebotene sofortige Start wird aber an mangelnden Rechten scheitern. Sie müssen das Programm nämlich mit Administratorrechten ausführen. Dazu suchen Sie im Startmenü bzw. in der Liste der Programme nach dem Eintrag *WIN32DISKIMAGER*, klicken diesen mit der rechten Maustaste an und wählen den Eintrag *ALS ADMINISTRATOR AUSFÜHREN*.

In dem kleinen Programm wählen Sie zuerst die Image-Datei aus und dann das Laufwerk, wohin das Image geschrieben werden soll. Aus Sicherheitsgründen stehen im Laufwerkslistenfeld nur SD-Karten und USB-Sticks zur Auswahl, aber keine

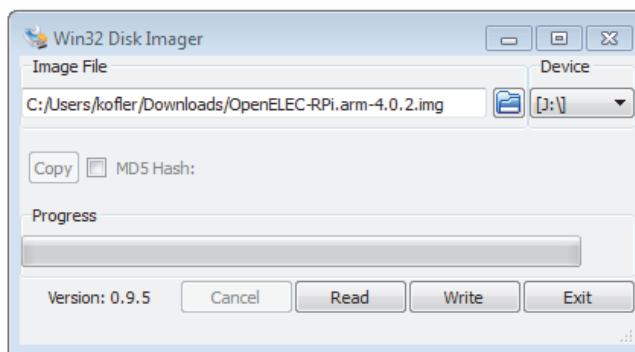


Abbildung 1.10 Win32 Disk Imager

Festplatten. Wenn Sie die Integrität Ihres Downloads überprüfen möchten, klicken Sie die Option MD5 HASH an. Das Programm errechnet dann eine Prüfsumme der Image-Datei, die Sie mit der oft auf der Download-Seite angegebenen Prüfsumme vergleichen können. WRITE schreibt die Image-Datei auf die SD-Karte. Das dauert wegen der zumeist bescheidenen Schreibgeschwindigkeit vieler SD-Karten mehrere Minuten.

Image-Datei unter OS X auf eine SD-Karte übertragen

Für OS X stehen diverse Programme zur Auswahl, um Image-Dateien auf eine SD-Karte zu übertragen. Die besten Erfahrungen haben wir mit dem *ApplePi-Baker* gemacht, das Sie von der folgenden Webseite kostenlos herunterladen können:

<http://www.tweaking4all.com/hardware/raspberry-pi/macosx-apple-pi-baker>

Nach dem Start des Programms wählen Sie im Listenfeld PI-CRUST die Device-Datei Ihrer SD-Karte aus (siehe Abbildung 1.11). Vorsicht, das Listenfeld enthält auch USB-Festplatten! Im Feld PI-INGREDIENTS klicken Sie rechts vom Textfeld IMG FILE auf den Button [...] und wählen dann die Image-Datei aus. Den eigentlichen Schreibprozess starten Sie schließlich mit dem Button IMG TO SD-CARD. Bevor der ApplePi-Baker mit seiner Arbeit beginnt, müssen Sie noch Ihr Passwort angeben. Der Schreibprozess erfordert Administratorrechte. Vergessen Sie nicht, die fertige SD-Karte zuerst im Finder auszuwerfen, bevor Sie die Karte aus dem Slot entfernen!

Das Programm bietet zwei Zusatzfunktionen: Mit dem Button PREP NOOBS CARD können Sie eine SD-Karte formatieren. BACKUP SD-CARD erstellt ein Backup von der eingelegten SD-Karte und speichert dieses als Image-Datei, die optional in einem komprimierten Archiv verpackt wird.

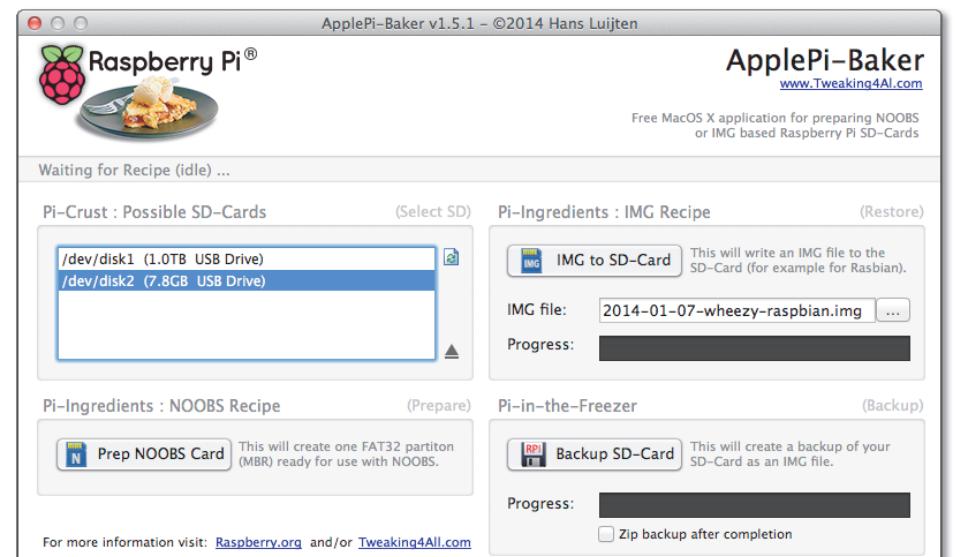


Abbildung 1.11 Der ApplePi-Baker

Roher Apfelkuchen

Bei unseren Tests ist es mehrfach vorgekommen, dass ApplePi-Baker bereits nach einer Sekunde behauptete, die SD-Karte sei fertig. Tatsächlich hatte der Schreibvorgang noch gar nicht begonnen. Abhilfe: Stellen Sie sicher, dass Sie die SD-Karte vorher formatieren, z.B. im Festplattendienstprogramm von OS X oder mit dem SD-Formatter – dann funktioniert es!

OS X-Experten können die SD-Karte natürlich auch im Terminal erstellen. Dazu ermitteln Sie zuerst mit diskutil list den Device-Namen des Datenträgers und lösen dann mit diskutil unmountDisk alle eventuell aktiven Partitionen des Datenträgers aus dem Verzeichnisbaum. Anschließend schreiben Sie mit sudo dd die Image-Datei direkt auf das Device der Festplatte. Anstelle von disk<n> geben Sie dabei aber rdisk<n> an. Damit sprechen Sie das *raw disk device* an, was erheblich schneller geht. Passen Sie aber auf, dass Sie sich beim if-Parameter nicht vertippen! Wenn Sie hier irrtümlich das falsche Device angeben, überschreiben Sie unrettbar Ihre Festplatte!

Das folgende Listing illustriert den Vorgang. Es gibt drei Datenträger: eine interne SSD (disk0), eine Backup-Festplatte (disk1) und die SD-Karte (disk2). Während der Ausführung von sudo dd gibt es leider keinerlei Feedback. Der Prozess dauert einige Minuten. Denken Sie daran, die SD-Karte anschließend im Finder auszuwerfen, bevor Sie sie aus dem SD-Slot entfernen.

```
diskutil list
/dev/disk0
#:          TYPE NAME      SIZE   IDENTIFIER
0: GUID_partition_scheme        512.1 GB   disk0
   EFI   EFI            209.7 MB   disk0s1
2:       Apple_HFS ssd        510.5 GB   disk0s2
3:       Apple_Boot Recovery    1.2 GB    disk0s3
/dev/disk1
#:          TYPE NAME      SIZE   IDENTIFIER
0: GUID_partition_scheme        1.0 TB    disk1
1:       EFI   EFI            209.7 MB   disk1s1
2:       Apple_HFS backup1     999.9 GB   disk1s2
/dev/disk2
#:          TYPE NAME      SIZE   IDENTIFIER
0: FDisk_partition_scheme        7.8 GB    disk2
1:       DOS_FAT_32 NO NAME    7.8 GB    disk2s1
diskutil unmountDisk /dev/disk2s1
Unmount of all volumes on disk2 was successful
sudo dd if=Downloads/raspbian.img of=/dev/rdisk2 bs=1M
2825+0 records in
2825+0 records out
2962227200 bytes transferred in 278.172140 secs
(10648900 bytes/sec)
```

Image-Datei unter Linux auf eine SD-Karte übertragen

Für Linux gibt es leider keine anerkannte Benutzeroberfläche, die beim Beschreiben von SD-Karten hilft. Das auf <http://elinux.org> erwähnte Programm ImageWriter wird nicht mehr gepflegt und steht nur auf alten Linux-Distributionen zur Verfügung. Sie müssen die SD-Karte daher in einem Terminal beschreiben. Das ist nicht schwierig: Wie unter OS X müssen Sie aber aufpassen, dass Ihnen keine Tippfehler unterlaufen!

Zuerst ermitteln Sie mit `lsblk` die Device-Namen aller Datenträger. Mit `umount` lösen Sie alle Dateisysteme der SD-Karte aus dem Verzeichnisbaum. Außer bei fabrikneuen SD-Karten sollten Sie auf der SD-Karte eine neue Partitionstabelle einrichten. Auf das eigentliche Formatieren können Sie diesmal aber verzichten. Die Vorgehensweise wurde ja schon in Abschnitt 1.2 beschrieben. Die folgenden Kommandos zeigen als Wiederholung nochmals die notwendigen Schritte. Beachten Sie, dass die Device-Namen auf Ihrem Linux-Rechner möglicherweise anders lauten! Im folgenden Beispiel ist `/dev/mmcblk0` der Device-Name der SD-Karte. Alle Kommandos müssen mit root-Rechten ausgeführt werden, unter Ubuntu also mit `sudo`.

```
lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0 119,2G  0 disk
      sda1   8:1    0 143,1M  0 part /boot/efi
      sda2   8:2    0   2,8G  0 part [SWAP]
...
mmcblk0 179:0  0   7,3G  0 disk
  mmcblk0p1 179:1  0   56M  0 part /media/kofler/boot
  mmcblk0p2 179:2  0   2,7G  0 part /media/kofler/fc254b57
```

```
umount /media/kofler/*
parted /dev/mmcblk0 mklabel msdos
```

Um die Image-Datei zu übertragen, verwenden Sie auch unter Linux das Kommando `dd`. Mit dem Parameter `if` geben Sie den Ort der Image-Datei an, mit `of` den Device-Namen der SD-Karte. Wie unter OS X gibt es auch unter Linux während der Ausführung von `dd` keinerlei Feedback. Sie müssen einfach einige Minuten abwarten, bis das Kommando abgeschlossen ist.

```
dd if=2014-01-07-wheezy-raspbian.img of=/dev/mmcblk0 bs=1M
2825+0 Datensätze ein
2825+0 Datensätze aus
2962227200 Bytes (3,0 GB) kopiert, 262,231 s, 11,3 MB/s
```

Distributionsspezifische Installationsprogramme

Da es immer wieder Raspberry-Pi-Einsteiger gibt, die das Übertragen einer Image-Datei auf die SD-Karte überfordert, sind manche Distributoren dazu übergegangen, eigene Installationsprogramme anzubieten, die direkt unter Windows, OS X und fallsweise auch unter Linux ausgeführt werden können. Beispielsweise gibt es derartige Installationshilfen für Raspbmc: Das ist eine Linux-Distribution, die speziell dafür gedacht ist, aus dem Raspberry Pi ein Multimedia-System zu machen (siehe auch Kapitel 7).

Leider sind uns beim Test dieser Installationsprogramme immer wieder Fehler aufgefallen. Insofern erscheint uns eine Image-Datei die einfachere Lösung, zumal das Beschreiben einer SD-Karte, wie Sie gerade gesehen haben, wirklich keine Hexerei ist.

1.4 Installation auf einen USB-Stick (für Fortgeschrittene)

Üblicherweise verwendet der Raspberry Pi eine SD-Karte als einzigen Datenspeicher: Die SD-Karte enthält gleichermaßen das Betriebssystem (oft Raspbian) als auch Ihre eigenen Daten, z. B. mit dem Raspberry Pi erstellte Fotos, Messdaten etc. Optional kann ein USB-Stick als zusätzlicher Datenspeicher verwendet werden.

Abweichend von diesem Standardszenario besteht auch die Möglichkeit, Linux direkt auf einen USB-Stick zu installieren. Die SD-Karte wird weiterhin benötigt, weil Raspbian von dort die für den Bootprozess erforderlichen Dateien liest. Aber alle weiteren Linux-Dateien und -Programme werden in der Folge direkt vom USB-Stick gelesen. Anstelle eines USB-Sticks können Sie auch eine USB-Festplatte mit eigener Stromversorgung verwenden.

Nur für Fortgeschrittene

Dieser Abschnitt richtet sich explizit an Leser bzw. Leserinnen, die bereits Linux- und Raspberry-Pi-Erfahrung haben. Viele Details, die in diesem Abschnitt vorkommen, werden erst in den weiteren Kapiteln dieses Buchs erklärt. Raspberry-Pi-Einsteiger sind gut beraten, vorerst eine normale Installation auf eine SD-Karte durchzuführen. Die Vorteile einer USB-Stick-Installation sind ohnedies nur in speziellen Anwendungsszenarien spürbar.

Vor- und Nachteile

Die Verwendung eines USB-Sticks anstelle einer SD-Karte hat einige offensichtliche Vorteile:

- ▶ USB-Sticks sind zuverlässiger als SD-Karten. Das gilt insbesondere dann, wenn Sie auf Ihrem Raspberry Pi Programme ausführen, die häufig große Datenmengen speichern bzw. ändern. Aus unserer Sicht ist das der entscheidende Punkt.
- ▶ SD-Karten dürfen maximal 32 GByte groß sein. Für USB-Sticks gilt diese Einschränkung nicht.
- ▶ Die Übertragungsgeschwindigkeit von bzw. zu USB-Sticks ist höher als bei SD-Karten. Ihr Raspberry Pi bootet schneller, Programme werden flotter gestartet. In der Praxis ist der Geschwindigkeitszuwachs freilich kleiner als erwartet. Limitierende Faktoren bleiben das USB-System und die CPU-Geschwindigkeit des Raspberry Pi. Die Bootgeschwindigkeit spielt zudem nur eine untergeordnete Rolle, weil der Raspberry Pi in den meisten Anwendungen ohnedies im Dauerbetrieb läuft und nicht ständig herunter- und wieder hochfahren wird.

Dem stehen die folgenden Nachteile gegenüber:

- ▶ Für die erste Phase des Bootprozesses wird weiterhin eine SD-Karte benötigt. Die erforderlichen Dateien beanspruchen weniger als 30 MByte. Den verbleibenden Platz können Sie immerhin als zusätzlichen Datenspeicher verwenden.
- ▶ Der USB-Stick blockiert einen der beiden wertvollen USB-Slots. Die Verwendung eines USB-Hubs ist damit in vielen Fällen unvermeidlich.
- ▶ Installation und Konfiguration sind etwas komplizierter.

USB-Stick vorbereiten

Wir gehen im Folgenden davon aus, dass Sie Raspbian installieren möchten. Prinzipiell ist eine USB-Stick-Installation natürlich auch für andere Distributionen möglich, allerdings sind dann unter Umständen kleine Abweichungen erforderlich.

Als Erstes müssen Sie Ihren USB-Stick so vorbereiten wie eine SD-Karte. Sofern es sich nicht um ein fabrikneues Modell handelt, formatieren Sie den USB-Stick. Anschließend übertragen Sie die Raspbian-Image-Datei auf den USB-Stick. Dazu können Sie dieselben Programme wie zum Beschreiben einer SD-Karte verwenden, also z. B. den Win32 Disk Imager, den ApplePi-Baker oder das Kommando dd.

SD-Karte vorbereiten

Wie bereits erwähnt wurde, benötigen Sie zusätzlich zum USB-Stick auch eine SD-Karte, wobei ein kleines Modell mit z. B. einem GByte Speicher vollkommen ausreichend ist. Nachdem Sie die SD-Karte formatiert haben, kopieren Sie *alle* Dateien aus der Boot-Partition des USB-Sticks dorthin. Es handelt sich dabei insbesondere um die Dateien `bootcode.bin`, `kernel.img`, `config.txt`, `cmdline.txt` und `start*.sh`. Diese Dateien müssen direkt auf der SD-Karte gespeichert werden, also nicht in einem Verzeichnis.

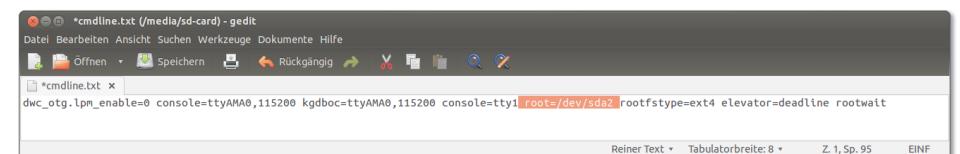


Abbildung 1.12 »cmdline.txt« in einem Editor verändern

Nun laden Sie die Datei `cmdline.txt` der SD-Karte in einen beliebigen Texteditor (siehe Abbildung 1.12). Diese Datei enthält in einer einzigen, sehr langen Zeile diverse Optionen, die beim Hochfahren des Raspberry Pi an den Kernel übergeben werden. Sie müssen im Editor nun *eine* Option verändern: Anstelle von `root=/dev/mmcblk0p2` muss es `root=/dev/sda2` heißen. Diese Änderung bewirkt, dass Linux die zweite Partition

des USB-Sticks als Systempartition verwendet, nicht wie sonst üblich die zweite Partition der SD-Karte.

Achten Sie darauf, dass die Datei weiterhin aus nur einer Zeile bestehen darf. Manche Editoren umbrechen den Text und machen aus der Optionszeile zwei oder drei kürzere Zeilen. In diesem Fall würde nur die erste Zeile berücksichtigt und der Bootvorgang würde scheitern! Nachdem Sie cmdline.txt gespeichert haben, stecken Sie die SD-Karte und den USB-Stick an Ihrem Raspberry Pi an und starten den Minicomputer.

Systempartition vergrößern

Sofern Ihnen bei den Vorbereitungsarbeiten keine Fehler unterlaufen sind, verläuft der Startprozess genauso wie bei der Verwendung einer normalen SD-Karte, nur ein klein wenig schneller. Beim ersten Start erscheint das Konfigurationsprogramm raspi-config, dessen Bedienung in Abschnitt 2.1 beschrieben ist.

Allerdings gibt es einen Menüpunkt in raspi-config, der für die USB-Stick-Installation nicht zutrifft: Bei einer SD-Karteninstallation können Sie mit EXPAND FILESYSTEM die Systempartition so weit vergrößern, dass sie den gesamten zur Verfügung stehenden Platz auf der SD-Karte füllt. Bei der USB-Stick-Installation funktioniert dieses Kommando leider nicht. Unabhängig davon, wie groß der USB-Stick ist, beträgt die Größe der Systempartition ca. 2,6 GByte, von denen noch ca. 450 MByte frei sind.

Damit Sie den ganzen Platz Ihres USB-Sticks verwenden können, müssen Sie daher selbst Hand anlegen. Die folgenden Kommandos zeigen, wie Sie mit dem Programm fdisk die Systempartition zuerst löschen und dann neu anlegen, wobei Sie unbedingt exakt dieselbe Startposition verwenden müssen. fdisk führen Sie wahlweise direkt in der Textkonsole oder in einem Terminalfenster aus.

```
sudo fdisk /dev/sda
Command (m for help): p
Disk /dev/sda: 31.4 GB, 31440961536 bytes
64 heads, 32 sectors/track, 29984 cylinders,
total 61408128 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: 0x000981cb
Device Boot Start End Blocks Id System
/dev/sda1 8192 122879 57344 c W95 FAT32 (LBA)
/dev/sda2 122880 61408127 30642624 83 Linux

Command (m for help): d
Partition number (1-4): 2
```

```
Command (m for help): n
Partition type:
 p primary (1 primary, 0 extended, 3 free)
 e extended
Select (default p): p
Partition number (1-4, default 2): 2
First sector (2048-61408127, default 2048): 122880
Last sector, +sectors or +size{K,M,G} (122880-61408127,
 default 61408127): <Return>
Using default value 61408127
```

```
Command (m for help): p
Disk /dev/sda: 31.4 GB, 31440961536 bytes
64 heads, 32 sectors/track, 29984 cylinders,
total 61408128 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: 0x000981cb
Device Boot Start End Blocks Id System
/dev/sda1 8192 122879 57344 c W95 FAT32 (LBA)
/dev/sda2 122880 61408127 30642624 83 Linux
```

```
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
WARNING: Re-reading the partition table failed with error 16:
Das Gerät oder die Ressource ist belegt. The kernel still uses
the old table. The new table will be used at the next reboot or
after you run partprobe(8) or kpartx(8)
```

Nun die ausführliche Erklärung, was hier vor sich geht: fdisk ist ein Kommando zur Partitionierung von Festplatten und anderen Datenträgern. Beim Start muss der sogenannte Device-Name des Geräts angegeben werden, also eine Spezialdatei, über die Linux auf die Festplatte zugreifen kann. Für den USB-Stick des Raspberry Pi lautet der Device-Name /dev/sda.

fdisk ist ein interaktives Programm, in dem Sie Kommandos ausführen. Die Eingabe von Kommandos erfolgt durch Buchstaben und [←]. Das erste Kommando [p] (print) listet die aktuelle Partitionstabelle auf. Es gibt also zwei Partitionen: eine kleine Partition mit einem Windows-Dateisystem, das in unserem Setup gar nicht verwendet wird, und eine größere Linux-Partition. Alle Start- und Endpositionen jeder Partition werden in Blöcken angegeben, wobei jeder Block 512 Byte groß ist. Davon abweichend wird die Größe der Partition in der Spalte Blocks hingegen in Vielfachen von 1024 Byte angegeben.

Die wichtigste Information für Sie ist die Startposition der zweiten Partition beim Block 122880. Sollte sich das Raspbian-Installations-Image nach Erscheinen dieses Buchs ändern, kann es sein, dass fdisk bei Ihnen eine andere Position anzeigt. Diese Position müssen Sie sich merken.

d (delete) löscht nun die zweite Partition. Keine Angst, sofern Sie die weitere Anleitung exakt befolgen, verlieren Sie dabei keine Daten! Denn bereits im nächsten Schritt wird die Partition mit **n** (new) wieder neu angelegt. fdisk fragt nun zuerst nach der Partitionsnummer (2), dann nach dem Partitionstyp (p für *primary*) und schließlich nach der Startposition der neuen Partition: Jetzt ist es entscheidend, dass Sie exakt dieselbe Startposition wie bisher angeben, in unserem Beispiel also 122880 Blöcke. Unkompliziert ist die Frage nach der Endposition: fdisk schlägt den letzten Block des USB-Sticks vor, und Sie bestätigen diese Position einfach mit **←**.

Bis jetzt haben Sie alle Änderungen nur im Speicher durchgeführt. Erst mit **w** (write) wird die neue Partitionstabelle tatsächlich auf dem USB-Stick gespeichert. Jetzt gibt es also kein Zurück mehr. Der Schreibvorgang endet mit einer Warnung: Da der USB-Stick momentan aktiv genutzt wird, kann der Linux-Kernel die geänderte Partitionstabelle vorerst noch nicht berücksichtigen. Sie müssen Ihren Raspberry Pi daher nun neu starten, am einfachsten mit sudo reboot.

Nach dem Neustart erkennt Linux die neue Partitionsgröße. Unverändert geblieben ist aber das Dateisystem, das weiterhin nur rund 2,6 GByte der Partition nutzt. Daher ist nun ein letzter Schritt erforderlich: Die Anpassung des Dateisystems an die neue Partitionsgröße. Dazu führen Sie das folgende Kommando in der Textkonsole oder in einem Terminal aus:

```
sudo resize2fs /dev/sda2
resize2fs 1.42.5 (29-Jul-2012)
Das Dateisystem auf /dev/sda2 ist auf / eingehängt;
Online-Größenveränderung nötig
old_desc_blocks = 1, new_desc_blocks = 2
Das Dateisystem auf /dev/sda2 ist nun 7660656 Blöcke groß.
```

Zuletzt sollten Sie sich vergewissern, ob alles funktioniert hat. Dazu führen Sie das Kommando df -h aus. Es gibt einen Überblick über alle aktiven Dateisysteme. Gleich die erste Zeile des Ergebnisses zeigt, dass das Dateisystem in der Systempartition auf unserem 32-GByte-USB-Stick nun rund 29 GByte beträgt. Vielleicht fragen Sie sich, wo die übrigen drei GByte geblieben sind. Der Grund für die Diskrepanz sind unterschiedliche Rechenweisen. Datenträgerhersteller rechnen immer dezimal. Ein USB-Stick mit 32 GByte umfasst demnach rund 32.000.000.000 Byte. df rechnet hingegen binär. Ein GByte entspricht dort 2^{30} Byte, also 1.073.741.824 Byte.

Die restlichen Zeilen des df-Ergebnisses betreffen größtenteils temporäre bzw. virtuelle Dateisysteme. Interessant wird es erst wieder bei den letzten beiden Zeilen. Die

/boot-Partition stammt von der SD-Karte und enthält die für den Bootprozess erforderlichen Daten. Über das Verzeichnis /media/boot ist außerdem die erste Partition des USB-Sticks zugänglich. Diese enthält ebenfalls Boot-Dateien, die aber ungenutzt sind.

	df -h
	Dateisystem Größe Benutzt Verf. Verw% Eingeht auf
rootfs	29G 2.0G 26G 8% /
/dev/root	29G 2.0G 26G 8% /
devtmpfs	211M 0 211M 0% /dev
tmpfs	44M 288K 44M 1% /run
tmpfs	5.0M 0 5.0M 0% /run/lock
tmpfs	88M 0 88M 0% /run/shm
/dev/mmcblk0p1	7.3G 22M 7.3G 1% /boot
/dev/sda1	56M 19M 38M 34% /media/boot

Kapitel 2

Erste Schritte in Raspbian

In diesem Kapitel zeigen wir Ihnen, wie Sie unter Raspbian elementare Arbeitsaufgaben erledigen. Wie wir im vorigen Kapitel ausgeführt haben, ist Raspbian die populärste Linux-Distribution für den Raspberry Pi. Sein Name setzt sich aus Raspberry Pi und Debian zusammen, d.h., Raspbian ist nichts anderes als eine für den Raspberry Pi optimierte Variante der bekannten Linux-Distribution Debian.

Wir setzen im Folgenden voraus, dass Sie Raspbian entweder via NOOBS installiert oder direkt als Image-Datei auf eine SD-Karte übertragen haben. Beim ersten Start von Raspbian erscheint automatisch das textbasierte Konfigurationsprogramm `raspi-config`, dessen Einstellmöglichkeiten wir Ihnen gleich erläutern.

Ist diese einmalige Konfiguration erledigt, startet Raspbian eine grafische Benutzeroberfläche. Dieses Kapitel gibt Ihnen einen Überblick über die wichtigsten Programme, die Ihnen dort zur Verfügung stehen. Außerdem finden Sie Tipps, wie Sie sich den Desktop Ihres Raspberry Pi nach eigenen Vorlieben einrichten, wie Sie Ihren Raspberry Pi mit dem WLAN verbinden, Bluetooth-Geräte einbinden und USB-Sticks verwenden.

2.1 Das Konfigurationsprogramm `raspi-config`

Der erste Start von Raspbian führt direkt in das Konfigurationsprogramm `raspi-config` (siehe Abbildung 2.1). Dieses Programm läuft im Textmodus und hilft dabei, einige grundlegende Einstellungen für Raspbian vorzunehmen.

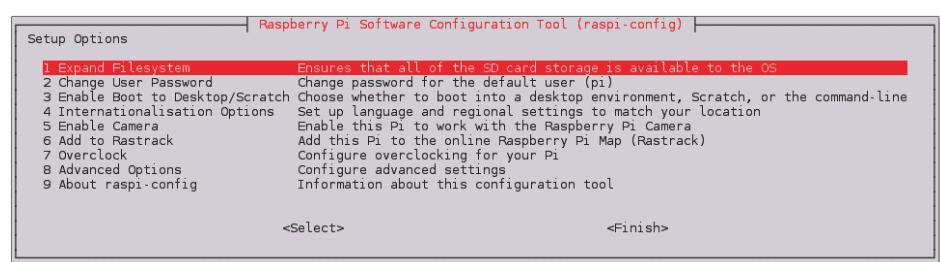


Abbildung 2.1 Das Raspbian-Konfigurationsprogramm

Sobald Sie eine erstmalige Konfiguration durchgeführt haben, wird das Programm nicht mehr automatisch gestartet. Sie brauchen aber keine Angst zu haben, dass Sie danach nichts mehr ändern können: Um das Konfigurationsprogramm später neuerlich auszuführen, starten Sie zuerst das Programm *LXTerminal* und führen darin das Kommando `sudo raspi-config` aus.

Der wichtigste Einstellungspunkt für Raspberry-Pi-Einsteiger ist im Menüeintrag **ENABLE BOOT TO DESKTOP/SCRATCH** versteckt. Dort wählen Sie die Option **DESKTOP**, damit Ihr Raspberry Pi beim nächsten Neustart die grafische Benutzeroberfläche startet. Die folgenden Abschnitte erklären die Konfigurationsoptionen von `raspi-config`.

Expand Filesystem

EXPAND FILESYSTEM vergrößert die Systempartition und das darauf befindliche Dateisystem, sodass dieses dann den gesamten zur Verfügung stehenden Platz der SD-Karte nutzen kann. Dieses Kommando ist erforderlich, wenn Sie Raspbian als Image-Datei auf eine SD-Karte mit mehr als 2 GByte Speicherkapazität geschrieben haben. Standardmäßig ist die Systempartition dann nur knapp 2 GByte groß, ganz egal, wie viel Speicherplatz die SD-Karte wirklich bietet. Die Größenanpassung wird beim nächsten Neustart des Raspberry Pi wirksam.

Bei einer NOOBS-Installation bringt **EXPAND FILESYSTEM** hingegen keinen Nutzen, weil sich NOOBS bereits um die optimale Einrichtung der Partition gekümmert hat.

Change User Password

CHANGE USER PASSWORD gibt Ihnen die Möglichkeit, das Passwort für den Defaultbenutzer `pi` zu ändern. Standardmäßig ist für diesen Benutzer das Passwort `raspberry` eingestellt. Dieses Passwort sollten Sie aus Sicherheitsgründen unbedingt ändern! Allerdings ist es zweckmäßig, vorher mit **INTERNATIONALISATION OPTIONS** das deutsche Tastaturlayout zu aktivieren (siehe unten).

Enable Boot to Desktop/Scratch

ENABLE BOOT TO DESKTOP/SCRATCH steuert, wie Raspbian gestartet werden soll. Dabei stehen drei Optionen zur Auswahl:

- ▶ **CONSOLE** bewirkt, dass Raspbian im Textmodus gestartet wird. Diese Einstellung gilt standardmäßig. Sie verkürzt den Startvorgang, ist aber nur zweckmäßig, wenn Ihr Raspberry Pi ohne Benutzerinteraktion automatisch irgendwelche Aufgaben erfüllen soll.

- ▶ Raspbian-Einsteiger sollten sich für die Variante **DESKTOP** entscheiden. Beim Start wird dann automatisch der Grafikmodus aktiviert und ein komfortables Desktop-System gestartet. Beachten Sie aber, dass dabei standardmäßig keine Login-Box erscheint. Jeder, der Zugang zum Minirechner hat, kann also damit arbeiten!
- ▶ Die Option **SCRATCH** führt dazu, dass beim nächsten Neustart ausschließlich das Programm *Scratch* gestartet wird. Dabei handelt es sich um eine einfache grafische Programmierumgebung, die Kindern bzw. Programmieranfängern dabei helfen soll, erste Konzepte des Programmierens zu verstehen. *Scratch* können Sie aber auch im Desktop-System starten. Insofern ist diese Option am ehesten geeignet, wenn Sie den Raspberry Pi im Unterricht einsetzen und vermeiden möchten, dass die Schüler mit anderen Programmen herumspielen.

Internationalisation Options

INTERNATIONALISATION OPTIONS führt in ein Untermenü, mit dessen Einträgen Sie die Sprache (**LOCALES**), die Zeitzone (**TIMEZONE**) und das Tastaturlayout (**KEYBOARD LAYOUT**) angeben können.

- ▶ Beim Punkt **LOCALES** sollten Sie außer der voreingestellten Option `EN_GB.UTF8` auch `DE_DE.UTF8` auswählen. Es ist nicht zweckmäßig, `EN_GB.UTF8` zu deaktivieren – viele Programme stehen nämlich nur mit englischsprachigen Dialogen zur Verfügung. Im nächsten Schritt können Sie unter allen Sprachen diejenige auswählen, die standardmäßig verwendet werden soll – in der Regel also nochmals `DE_DE.UTF8`.
- ▶ Als **TIMEZONE** wählen Sie **EUROPE • BERLIN** aus. Beachten Sie, dass der Raspberry Pi keine batteriegepufferte Uhrzeit hat und deswegen bei jedem Neustart mit derselben Zeit beginnt. Wenn das Sie stört, schließen Sie den Raspberry Pi an ein Netzwerk an, damit der Rechner die aktuelle Uhrzeit über das Network Time Protocol (NTP) beziehen kann.
- ▶ Die Einstellung des **KEYBOARD LAYOUTS** erfolgt in mehreren Schritten: Zuerst müssen Sie Ihr Tastatormodell angeben. Für deutsche PC-Tastaturen lautet der richtige Eintrag in der Regel **GENERIC 105-KEY (INTL) PC**.
Im zweiten Schritt geben Sie dann das eigentliche Tastaturlayout an: Zuerst führt **OTHER** in eine lange Liste aller Sprachen. Nach der Auswahl von **GERMAN** haben Sie noch die Wahl zwischen verschiedenen Untervarianten, die sich vor allem dadurch unterscheiden, wie Sonderzeichen einzugeben sind. Für Programmierer ist **GERMAN (ELIMINATE DEAD KEYS)** die beste Wahl: Damit können die Tilde sowie die Apostrophe Acute und Gravis direkt eingegeben werden.

Nun besteht die Möglichkeit, zusätzliche Optionen zu aktivieren. Damit können Sie z. B. steuern, welche Taste als [AltGr] zur Eingabe von Sonderzeichen verwendet werden soll. Normalerweise übernehmen Sie hier einfach die Defaulteinstellungen.

Zuletzt fragt das Konfigurationsprogramm, ob die Tastenkombination **[Strg]+[Alt]+[←]** dazu dienen soll, das Grafiksystem sofort zu beenden und neu zu starten. Diese Notfall-Tastenkombination war früher auf vielen Linux-Systemen üblich, ist heute aber nur selten zweckmäßig – also No.

Enable Camera

Zum Raspberry Pi gibt es als Erweiterungsmodul eine Kamera. Um diese zu verwenden, muss der Linux-Kernel zusätzliche Module laden. Die dafür erforderliche Konfiguration erledigt `raspi-config`, wenn Sie den Menüpunkt ENABLE CAMERA ausführen. Eine Menge weiterer Informationen und Tipps zum Einsatz der Raspberry-Pi-Kamera folgen in Kapitel 13.

Add to Rastrack

Der Menüpunkt ADD TO RASTRACK ermöglicht es Ihnen, Ihren Raspberry Pi in ein inoffizielles Verzeichnis aktiver Raspberry-Pi-Installationen einzutragen. Praktischer Nutzen ergibt sich daraus keiner, die resultierende Weltkarte ist aber hübsch anzusehen (siehe Abbildung 2.2). Absurderweise ist die grafische Gestaltung der Webseite derart aufwendig, dass eine Darstellung auf dem Webbrowser des Raspberry Pi unmöglich ist. Sie müssen sich die Seite also auf Ihrem Notebook oder PC ansehen.

<http://rastrack.co.uk>

Overclock

Die CPU des Raspberry Pi läuft standardmäßig mit 700 MHz. Mit dem Menüpunkt OVERCLOCK können Sie die Frequenz in vier Stufen auf bis zu 1 GHz erhöhen. Während manche Raspberry-Pi-Fans von den damit verbundenen Geschwindigkeitssteigerungen schwärmen, haben andere damit schlechte Erfahrungen gemacht: Abstürze, defekte Dateisysteme oder ein vorzeitiger Totalausfall des Raspberry Pi sind möglich. Ausführliche Informationen zu diesem Thema gibt Abschnitt 4.14. Unsere Empfehlung lautet, auf Overclocking zu verzichten.

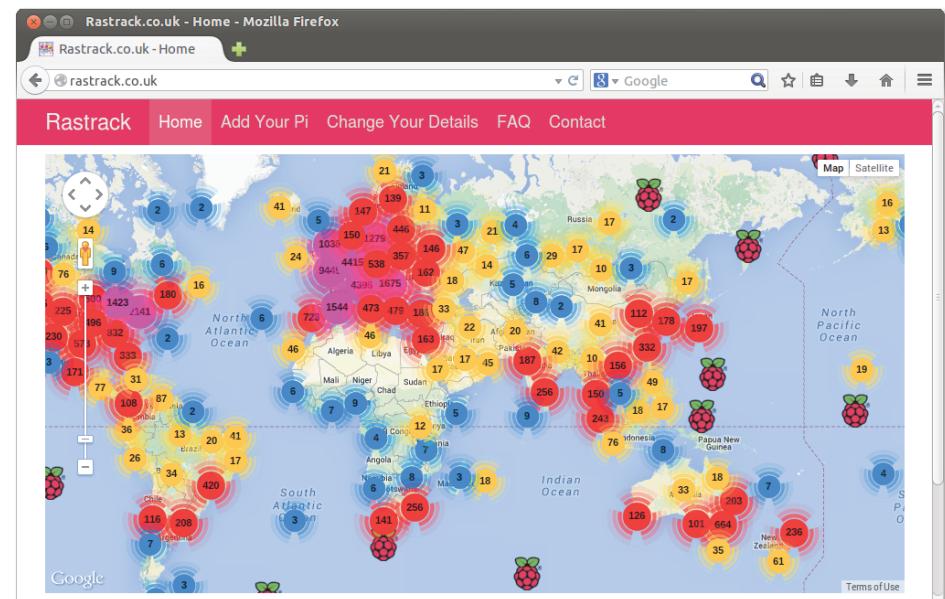


Abbildung 2.2 Die Rastrack-Weltkarte

Advanced Options

Der Menüeintrag ADVANCED OPTIONS führt in ein Untermenü mit weiteren Einstellungsmöglichkeiten:

- ▶ OVERSCAN: Dieser Punkt gibt Ihnen die Möglichkeit, im normalerweise schwarzen Randbereich eines analogen Monitors ebenfalls eine Farbe darzustellen. Bei einem HDMI-Monitor bzw. -Fernseher belassen Sie die Option auf disable.
- ▶ HOSTNAME: Hiermit können Sie den Rechnernamen Ihres Raspberry Pi einstellen. Standardmäßig lautet dieser `raspberrypi`.
- ▶ MEMORY SPLIT: Gibt an, wie viel Speicher für das Grafiksystem und wie viel für das Betriebssystem reserviert wird. Hintergrundinformationen zu dieser Einstellung finden Sie in Abschnitt 5.8.
- ▶ SSH: Standardmäßig wird auf Ihrem Raspberry Pi ein SSH-Server ausgeführt. Das ermöglicht es Ihnen, sich über das Netzwerk auf Ihrem Raspberry Pi einzuloggen. Im Menüpunkt SSH können Sie den SSH-Server deaktivieren bzw. wieder aktivieren. Wie Sie SSH nutzen und welche Sicherheitsmaßnahmen Sie beim Ausführen eines SSH-Servers beachten sollten, fasst Abschnitt 4.3 zusammen. So viel vorweg: Ändern Sie unbedingt vorher das Passwort des Benutzers `pi`, wenn Sie den SSH-Server ausführen!

- ▶ **SPI:** Die CPU des Raspberry Pi unterstützt das Serial Peripheral Interface, kurz SPI. Wenn Sie dieses Bus-System einsetzen möchten, beispielsweise in Kombination mit PiFace-Erweiterungsmodulen, können Sie mit dem Menüeintrag SPI das automatische Laden der erforderlichen Kernelmodule einrichten.
- ▶ **AUDIO:** Standardmäßig entscheidet der Raspberry Pi selbst, welchen Audio-Ausgang er verwendet: den HDMI-Ausgang, wenn ein Monitor angeschlossen ist, oder andernfalls den Analog-Ausgang für einen 3,5-mm-Klinkenstecker. Mit dem Menüpunkt AUDIO können Sie diesen Automatismus deaktivieren und die Audio-Ausgabe stattdessen zwingend auf einen der beiden Kanäle umleiten.
- ▶ **UPDATE:** Dieser Menüpunkt aktualisiert raspi-config auf die neueste Version. Das setzt eine funktionierende Internetverbindung voraus.

Neustart

Nach dem Abschluss der Konfigurationsarbeiten landen Sie in einer Textkonsole. Damit die neuen Einstellungen wirksam werden, müssen Sie den Raspberry Pi nun neu starten. Dazu tippen Sie das folgende Kommando ein:

```
sudo reboot
```

Je nach Geschwindigkeit der SD-Karte dauert ein kompletter Boot-Vorgang bis zur Anzeige der grafischen Benutzeroberfläche ungefähr eine halbe Minute.

2.2 Ein- und Ausschalten

Der Raspberry Pi verfügt über keinen Ein/Aus-Schalter. Um das Gerät zu starten, verbinden Sie den Minicomputer zuerst mit allen Peripheriegeräten (Monitor, Netzwerk, Tastatur, Maus etc.) und stecken die SD-Karte ein. Zuletzt stecken Sie das Micro-USB-Kabel für die Stromversorgung an. Daraufhin fährt der Raspberry Pi hoch.

Zum Ausschalten führen Sie in der grafischen Benutzeroberfläche das Menükommando **ABMELDEN • AUSSCHALTEN** aus. Alternativ können Sie im Textmodus, in einem Terminalfenster oder per SSH `sudo halt` oder `sudo reboot` ausführen. Das erste Kommando schaltet den Raspberry Pi aus, das zweite startet ihn neu. Erst wenn der Rechner vollständig heruntergefahren ist und nur noch die rote Diode des Raspberry Pi leuchtet, lösen Sie das Micro-USB-Kabel zur Stromversorgung oder stecken das Netzteil aus.

Wenn der Raspberry Pi abstürzt

Manchmal kommt es vor, dass der Raspberry Pi hängen bleibt und auf Tastatur- und Mauseingaben nicht mehr reagiert. Mögliche Ursachen können Soft- oder Hardware-Fehler sein, eine defekte SD-Karte, eine unzureichende Stromversorgung, ein Wackelkontakt etc.

Sie können nun noch versuchen, sich über einen anderen Rechner via SSH einzuloggen (siehe Abschnitt 4.3) und dann `sudo halt` auszuführen.

Ist auch das unmöglich, weil der Raspberry Pi vollkommen abgestürzt ist, hilft nur das Ein- und Ausstecken des Netzteils bzw. dessen Micro-USB-Kabels. Das ist eine Notlösung, die Sie im regulären Betrieb vermeiden sollten: Der Raspberry Pi hat so keine Möglichkeit, geöffnete Dateien zu schließen und das Dateisystem ordentlich auszuhängen.

Normalerweise reagiert der Raspberry Pi auf derartige Neustarts erstaunlich robust: Der Raspberry Pi startet einfach neu, als wäre nichts gewesen. Dennoch ist es nicht auszuschließen, dass das Dateisystem Schaden nimmt oder Sie einzelne Dateien verlieren.

Sollten sich die Abstürze häufen, müssen Sie auf Fehlersuche gehen: Probieren Sie ein leistungsstärkeres Netzteil, versuchen Sie es mit einer anderen SD-Karte, verbinden Sie Ihre USB-Geräte nicht direkt, sondern über einen aktiven USB-Hub etc. Weitere Notfalltipps sind in Abschnitt 4.15 zusammengefasst.

Erstellen Sie regelmäßige Backups!

Egal, auf welchem Computer Sie arbeiten: Im Katastrophenfall können Sie sich nur durch Backups vor einem Datenverlust schützen. Für den Raspberry Pi gilt diese Empfehlung umso mehr, weil SD-Karten erwiesenermaßen weniger zuverlässig sind als Festplatten oder SSDs. Wie Sie Backups erstellen, beschreibt Abschnitt 4.13.

2.3 Die Benutzeroberfläche LXDE

Sobald Sie in raspi-config die Option **ENABLE BOOT TO DESKTOP/SCRATCH** eingestellt haben, erscheint ab dem nächsten Neustart automatisch die grafische Benutzeroberfläche des Raspberry Pi (siehe Abbildung 2.3). Auch wenn Sie bisher primär unter Windows oder OS X gearbeitet haben, wird Ihnen die Bedienung nicht schwerfallen: Auf dem Desktop befinden sich standardmäßig einige Icons zum Start der wichtigsten Programme. Alle weiteren Programme sowie Konfigurations- und Neustart-Kommandos finden Sie im Startmenü, das Sie über das Icon am linken unteren Bildschirmrand öffnen.

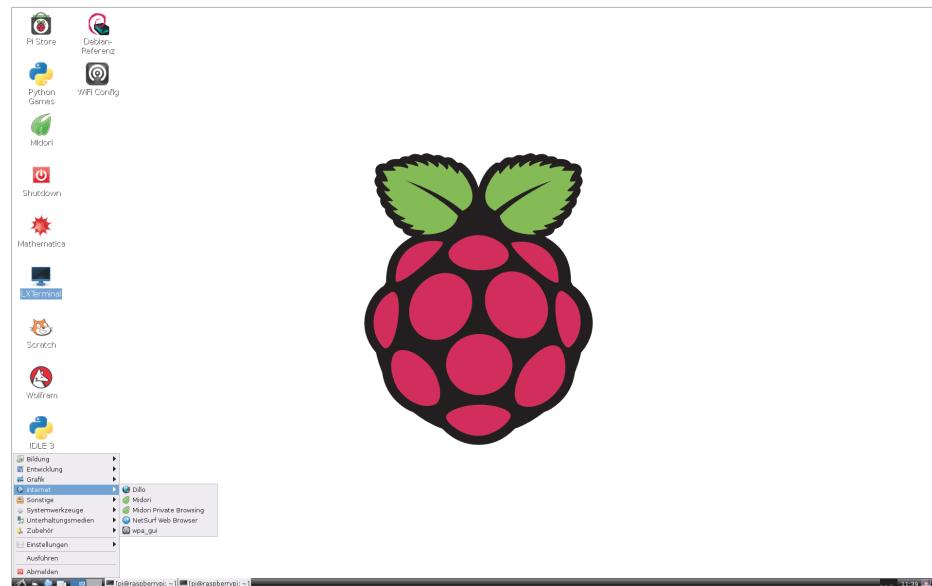


Abbildung 2.3 Der Raspbian-Desktop

Warum nicht KDE oder Gnome?

Wer sich schon vor dem Kauf des Raspberry Pi mit Linux auseinandergesetzt hat, ist vermutlich mit den Desktop-Systemen KDE und Gnome vertraut, die unter Debian, Fedora, openSUSE oder Ubuntu zum Einsatz kommen. Die Benutzeroberfläche von Raspbian sieht hingegen ganz anders aus. Der Grund dafür ist einfach: Die Desktop-Systeme KDE bzw. Gnome würden den Raspberry Pi mit seinen vergleichsweise geringen Hardware-Ressourcen überfordern. Deswegen haben sich die Raspbian-Entwickler dafür entschieden, standardmäßig die speziell für weniger leistungsstarke Computer konzipierte Benutzeroberfläche *Lightweight X11 Desktop Environment* zu installieren. Insofern beziehen sich viele Informationen dieses Kapitels also ganz allgemein auf das LXDE-Projekt und sind durchaus nicht Raspberry-Pi-spezifisch.

Icons

Die Icons auf dem Desktop sind frei platzierbar. Alle Icons, deren Position Sie *nicht* ändern, werden standardmäßig in der Reihenfolge sortiert, in der sie dem Desktop hinzugefügt wurden. Durch ein Kontextmenü des Desktops können Sie die Sortierreihenfolge ändern. Frei platzierte Icons behalten dabei ihre Position. Wollen Sie auch diese Icons wieder automatisch platzieren, klicken Sie diese mit der rechten Maustaste an und deaktivieren die Option AKTUELLE POSITION MERKEN. Über dieses Kontextmenü können Sie auch Icons vom Desktop entfernen.

Um ein Icon eines häufig benötigten Programms auf dem Desktop anzeigen zu lassen, klicken Sie das Programm im Startmenü mit der rechten Maustaste an und führen DEM DESKTOP HINZUFÜGEN aus. Desktop-Icons können Sie wiederum per Kontextmenü als STARTER ZUR LEISTE HINZUFÜGEN, also in das Panel am unteren Bildschirmrand integrieren.

Das Panel

Die Leiste am unteren Bildschirmrand wird als »Panel« bezeichnet. Es enthält standardmäßig neben dem Startmenü kleine Icons zum Start des Dateimangers und des Webbrowsers, zum Minimieren aller Fenster sowie zum Umschalten zwischen zwei virtuellen Arbeitsflächen. Danach folgt eine Task-Leiste im Stil von Windows 95. Der rechte Rand des Panels zeigt die aktuelle CPU-Auslastung und die Uhrzeit an.

Wenn Sie das Panel mit der rechten Maustaste anklicken, gelangen Sie in ein Kontextmenü zur Panelkonfiguration. Mit den dort zugänglichen Kommandos können Sie das Panel an einen anderen Bildschirmrand verschieben, zusätzliche Elemente in das Panel integrieren oder sogar mehrere Panel einrichten (siehe Abbildung 2.4).

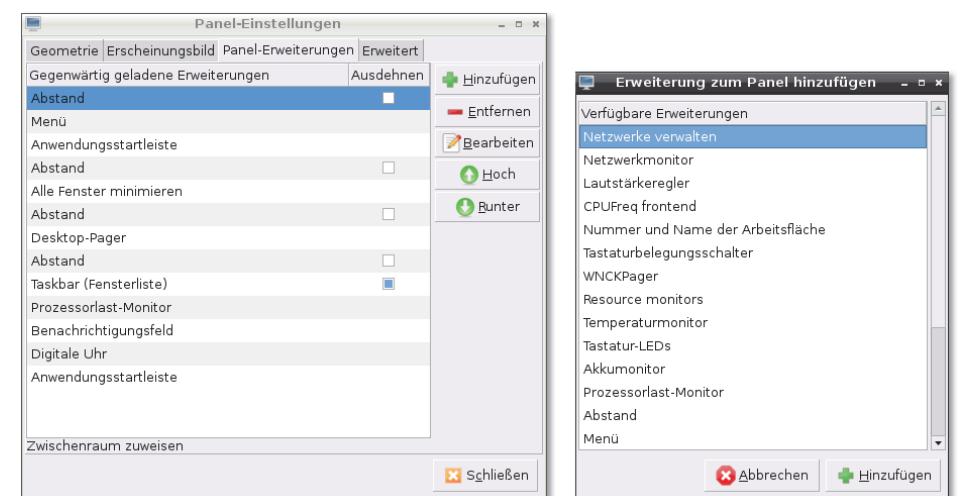


Abbildung 2.4 Panel-Einstellungen

Wenn Sie eigene Icons in den Schnellstartbereich neben dem Startmenü einfügen möchten, klicken Sie eines der bereits vorhandenen Icons mit der rechten Maustaste an und öffnen die ANWENDUNGSSTARTLEISTE-EINSTELLUNGEN. Sie können nun Einträge aus dem Startmenü in die Liste der Schnellstartprogramme aufnehmen (siehe Abbildung 2.5).

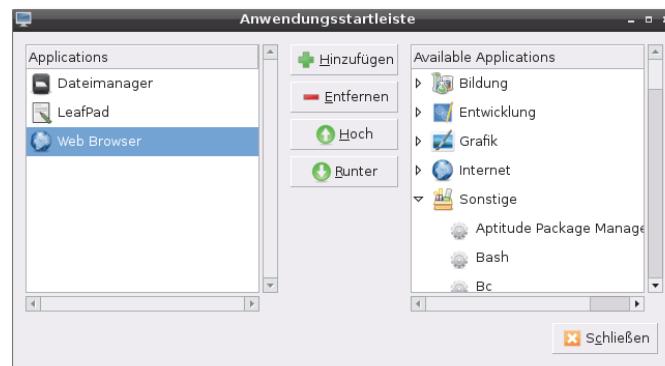


Abbildung 2.5 Konfiguration des Schnellstartbereichs

Programme starten

Sie wissen nun schon, dass Sie Programme über das Startmenü oder durch einen Doppelklick auf ein Icon starten können. Sofern Sie den internen Namen des Programms kennen (also den Namen der ausführbaren Datei des Programms), gibt es noch zwei Möglichkeiten:

- ▶ **[Alt]+[F2]** öffnet den winzigen AUSFÜHREN-Dialog. Dort geben Sie die ersten Buchstaben des Programmnamens an. Der Name wird automatisch vervollständigt. startet das Programm.
- ▶ Alternativ können Sie Programme auch aus einem Terminalfenster heraus starten. Dazu geben Sie den Programmnamen und anschließend das Zeichen & an, also z.B. leafpad &. Mit starten Sie das Programm als Hintergrundprozess. Damit können Sie im Terminal sofort weiterarbeiten.

Bei komplexeren Programmen kann der Programmstart etliche Sekunden dauern. Dass Raspbian Ihren Wunsch nicht einfach ignoriert, sondern hart an der Erfüllung arbeitet, erkennen Sie an der Darstellung der CPU-Auslastung rechts unten im Panel.

Konfiguration

Über das Startmenü EINSTELLUNGEN können Sie mehrere Programme zur Konfiguration des LXDE-Desktops starten. Dieser Abschnitt beschreibt die wichtigsten Konfigurationsmöglichkeiten.

Im Dialog EINSTELLUNGEN • BILDSCHIRMEINSTELLUNGEN können Sie ein Bild und eine Farbe einstellen, die zusammen den Bildschirmhintergrund ergeben. Wenn Sie einen einfarbigen Hintergrund ohne Bild vorziehen, wählen Sie im Listenfeld AUSRICHTUNG den Eintrag MIT HINTERGRUNDFARBE FÜLLEN.

In EINSTELLUNGEN • ERSCHEINUNGSBILD können Sie verschiedene Layout-Themen aktivieren. Diese Themen bestimmen die optische Gestaltung von Fensterrahmen und Dialogelementen.

Noch mehr Gestaltungsmöglichkeiten offenbaren sich im Programm EINSTELLUNGEN • OPENBOX EINSTELLUNGSMANAGER (siehe Abbildung 2.6). Hier können Sie zwischen verschiedenen Farbschemata auswählen, die Anordnung der Fensterbuttons verändern, die Schriftarten für den Fenstertitel und die Menüs einstellen, die Anzahl der Arbeitsflächen verändern etc.

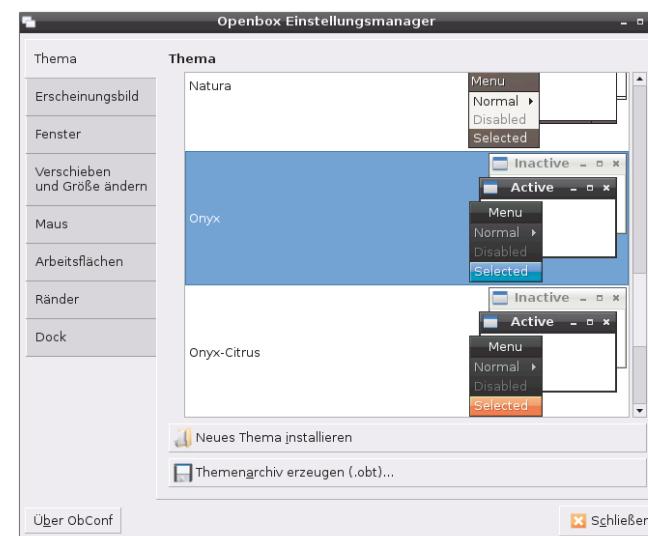


Abbildung 2.6 Gestaltungsoptionen für den Desktop

Der Dialog EINSTELLUNGEN • TASTATUR UND MAUS dient schließlich dazu, die Geschwindigkeit von Tastatur und Maus zu steuern.

Tastenkürzel

Zur Bedienung des Desktops gibt es eine Reihe von Tastenkürzeln, von denen Ihnen manche sicherlich aus anderen Betriebssystemen vertraut sind (siehe Tabelle 2.1). Die Tastenkürzel sind in den Dateien /etc/xdg/openbox/rc.xml und .config/openbox/lxde-rc.xml definiert. Vielleicht fragen Sie sich, was Openbox ist: Dabei handelt es sich um den von Raspbian eingesetzten Window Manager. Ein Window Manager ist wiederum das Programm, das für die Darstellung und Verwaltung von Fenstern auf dem Bildschirm verantwortlich ist.

Tastenkürzel	Bedeutung
[Alt] + [←]	wechselt das aktive Fenster.
[Alt] + [Esc]	platziert das aktuelle Fenster unter allen anderen.
[Alt] + Leertaste	zeigt das Fenstermenü an.
[Alt] + [F2]	startet ein Programm, dessen Name genannt werden muss.
[Alt] + [F4]	schließt das Fenster bzw. beendet das Programm.
[Alt] + [F11]	aktiviert bzw. deaktiviert den Vollbildmodus.
[Strg] + [Esc]	öffnet das Startmenü.
[Strg] + [Alt] + [Entf]	öffnet den Taskmanager.
[Windows] + [D]	minimiert alle Fenster bzw. zeigt diese wieder an.
[Windows] + [E]	startet den Dateimanager.
[Windows] + [R]	startet ein Programm, dessen Name genannt werden muss.
[Windows] + [F1]	aktiviert den virtuellen Desktop 1.
[Windows] + [F2]	aktiviert den virtuellen Desktop 2.

Tabelle 2.1 Tastenkürzel für den Raspbian-Desktop

Bei Bedarf können Sie mit einem Texteditor die Datei `.config/openbox/lxde-rc.xml` Ihren eigenen Wünschen entsprechend anpassen. Die leicht verständliche Syntax ist hier dokumentiert:

<http://openbox.org/wiki/Help:Bindings>

Beispielsweise bewirken die folgenden Zeilen, dass beim Drücken der Taste AUSWERFEN auf einer Apple-Tastatur ein Screenshot mit dem Programm xfce4-screenshooter erstellt wird. Dieses Programm muss allerdings bereits mit `sudo apt-get install xfce4-screenshooter` installiert worden sein.

```
# in der Datei .config/openbox/lxde-rc.xml
<keyboard>
  ...
  <keybind key="XF86Eject">
    <action name="Execute">
      <command>
        xfce4-screenshooter --fullscreen -s ~/Bilder
      </command>
    </action>
  </keybind>
</keyboard>
```

Änderungen in `lxde-rc.xml` werden normalerweise erst wirksam, wenn Sie sich ab- und neu anmelden. Sollte Ihnen das zu umständlich erscheinen, führen Sie einfach in einem Terminalfenster das Kommando `openbox --reconfigure` aus.

Bildschirmschoner

Nach zehn Minuten ohne Tastatur- und Mausaktivität aktiviert sich der Bildschirmschoner. Damit wird der Bildschirminhalt zwar schwarz, der Monitor läuft aber weiter. Der Raspberry Pi ist leider nicht in der Lage, das richtige DPMS-Signal (Display Power Management Signaling) an den Monitor zu senden, sodass sich dieser abschaltet:

<https://github.com/raspberrypi/linux/issues/487>

Da der Bildschirmschoner so wenig Sinn macht und im Gegenteil dazu führt, dass ein nur ein schwarzes Bild zeigender Monitor unbeabsichtigt länger läuft als notwendig, gibt es im Internet eine Menge Diskussionen darüber, wie der Bildschirmschoner deaktiviert werden kann. Das ist auch dann zweckmäßig, wenn der Raspberry Pi ein Display steuern soll, das ständig Informationen anzeigt – z. B. in einem Museum oder in einem Geschäft.

Die einfachste für Raspbian geeignete Lösung besteht darin, die Datei `/etc/lightdm/lightdm.conf` in einen Editor zu laden. Da es sich um eine Systemkonfigurationsdatei handelt, müssen Sie den Editor aus einem Terminal heraus mit `sudo` starten. Dazu führen Sie das folgende Kommando aus:

```
sudo leafpad /etc/lightdm/lightdm.conf
```

In der Konfigurationsdatei suchen Sie nach dem Abschnitt, der mit `[SeatDefaults]` eingeleitet wird, und fügen dort die Zeile `xserver-command=...` ein:

```
# in der Datei /etc/lightdm/lightdm.conf
...
[SeatDefaults]
xserver-command=X -s 0 -dpms
...
```

Die Einstellung bewirkt, dass das Programm X, das für die Darstellung der grafischen Benutzeroberfläche verantwortlich ist, in Zukunft gar keine Energiesparfunktionen verwendet.

2.4 Wichtige Programme

Dieser Abschnitt stellt Ihnen überblicksartig die wichtigsten Programme vor, die Ihnen unter Raspbian zur Verfügung stehen. Dabei berücksichtigen wir sowohl die vorinstallierten Programme als auch einige verborgene Perlen, die Sie erst installieren müssen.

Die Icon-Galerie des Raspbian-Desktops

Auf dem Raspbian-Desktop werden einige Programme prominent in den Vordergrund gestellt. Diese wollen wir Ihnen hier in alphabetischer Reihenfolge kurz vorstellen:

- ▶ **Debian Referenz** öffnet eine HTML-Version des englischsprachigen Buchs *Debian Reference*. Da Raspbian nichts anderes als eine für den Raspberry Pi optimierte Version von Debian ist, hilft dieses Buch – und natürlich auch jedes andere Debian-Buch! – beim tieferen Verständnis von Raspbian sowie bei der Lösung von Konfigurationsproblemen.
- ▶ **IDLE** und **IDLE3** sind einfache grafische Entwicklungsumgebungen für die Programmiersprache Python in den Versionen 2.6 bzw. 3.2. Dieses Buch enthält eine sehr ausführliche Beschreibung von Python, die in Kapitel 16 beginnt. Ganz egal, welche Funktion bzw. welche Hardware-Komponenten des Raspberry Pi Sie steuern möchten – der einfachste Weg führt nahezu immer über die Programmiersprache Python!
- ▶ **LXTerminal** startet einen sogenannten Terminal emulator. Das ist ein Fenster, in dem Sie Linux-Kommandos ausführen können. Je weiter Sie diesem Buch folgen, desto mehr werden Sie feststellen, wie dieses unscheinbare Programm zu Ihrem wichtigsten Werkzeug wird.
- ▶ **Mathematica** startet das gleichnamige Programm, mit dem Sie mathematische Probleme symbolisch lösen und grafisch darstellen können. Eine Einführung in dieses faszinierende Programm gibt Kapitel 23.
- ▶ **Midori** startet den gleichnamigen Webbrowser (siehe Abbildung 2.7). Obwohl dieses Programm sparsamer mit dem Speicher umgeht als beispielsweise Firefox, macht das Surfen auf dem Raspberry Pi wenig Spaß. Der Seitenaufbau erfolgt schleppend, und das Scrollen durch längere Seiten ist mit vielen Verzögerungen verbunden. Moderne Webseiten sowie Multimedia-Angebote überfordern Midori in der Regel. Anders formuliert: Zur Recherche eines Raspberry-Pi-Konfigurationsproblems reicht Midori aus, zum Ansehen von YouTube-Videos aber nicht. Ansonsten verhält sich Midori wie jeder andere Webbrowser.
- ▶ **OCR Ressources** öffnet die Webseite <http://www.ocr.org> mit Tutorials und Anleitungen rund um den Raspberry Pi.

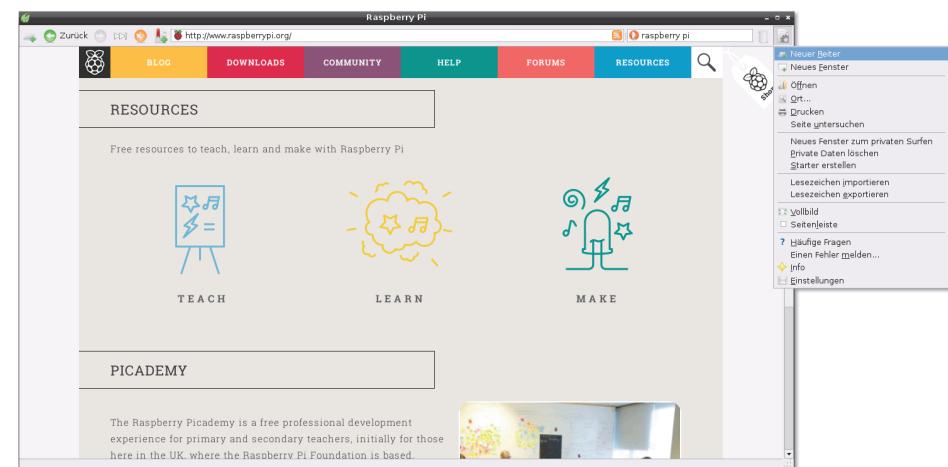


Abbildung 2.7 Der Webbrower Midori

- ▶ **Pi Store** führt in einen kleinen Webstore, in dem Sie Software und Literatur für den Raspberry Pi herunterladen bzw. kaufen können. Das Angebot ist aber recht unspektakulär.
- ▶ **Python Games** hilft dabei, diverse kleinere Spiele zu starten. Die Spiele an sich sind unspektakulär und werden Sie sicher nicht stundenlang fesseln. Interessant sind die Spiele aus einem anderen Aspekt: Ihr Quellcode ist auf der Webseite <http://inventwithpython.com> frei verfügbar und kann angehenden Python-Programmierern als Beispiel dienen.
- ▶ **Scratch** ist eine grafische Programmierumgebung, die Kindern bzw. Programmieranfängern die Konzepte der Programmierung nahezubringen versucht (siehe Abbildung 2.8).
- ▶ **Shutdown** fragt, ob Sie den Raspberry Pi herunterfahren möchten.
- ▶ **WiFiConfig** startet ein Programm zur WLAN-Konfiguration (siehe Abschnitt 2.5).
- ▶ **Wolfram** ist die Textoberfläche zum bereits erwähnten Programm Mathematica. Darin können Sie nicht nur Mathematica-Kommandos, sondern auch andere Konstrukte der Programmiersprache Wolfram ausführen. Das spröde Wolfram-Fenster strahlt zwar nicht dieselbe Faszination wie die Mathematica-Benutzeroberfläche aus, zeichnet sich dafür aber durch eine wesentlich höhere Verarbeitungsgeschwindigkeit aus.

Das Startmenü enthält eine Menge weiterer Programme. Erwähnenswert sind der Bildbetrachter, der Dateimanager (siehe den folgenden Abschnitt), der Editor *Leafpad* und das ZIP-Programm *Xarchiver*, die Sie beide im Untermenü ZUBEHÖR finden.

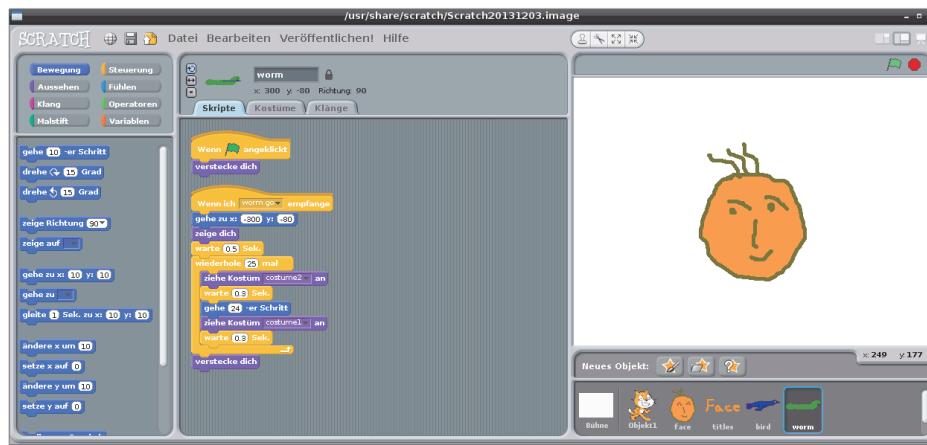


Abbildung 2.8 Programmieren lernen in Scratch

War das schon alles?

Erscheint Ihnen die Programmauswahl ein wenig mager? Keine Angst, unter Raspbian stehen Tausende weitere Programme in Form fertiger Pakete zur Verfügung: Um sie zu installieren, starten Sie das LXTerminal und führen dort das Kommando `sudo apt-get install name` aus. Wie die Paketverwaltung hinter den Kulissen funktioniert, erklärt Abschnitt 4.1.

Eine Einschränkung müssen wir aber noch erwähnen: Nicht jedes Programm läuft unter Raspbian zufriedenstellend. Schuld daran ist nicht Raspbian an sich, sondern die Hardware-Ausstattung des Raspberry Pi. Populäre Programme wie GIMP oder LibreOffice sind zwar rasch installiert, lassen sich auf dem Raspberry Pi aber nicht in angemessener Geschwindigkeit verwenden.

Der Dateimanager

Ihre eigenen Daten befinden sich unter Raspbian im Verzeichnis `/home/pi`. Dieses Verzeichnis wird auch Heimatverzeichnis genannt. Bei der Verwaltung der hier gespeicherten Dateien hilft Ihnen der Dateimanager *PCManFM* (siehe Abbildung 2.9), den Sie wahlweise mit **ZUBEHÖR** • **DATEIMANAGER** oder durch einen Klick auf das entsprechende Icon im Panel starten.

Die Grundfunktionen des Programms erschließen sich beim Ausprobieren in wenigen Minuten. Im Menü **ANSICHT** können Sie zwischen vier verschiedenen Darstellungsformen umschalten: Standardmäßig ist die Symbolansicht aktiv. Optisch weniger ansprechend, dafür aber informativer ist die Detailansicht, die auch die Dateigröße und den Zeitpunkt der letzten Änderung anzeigt.

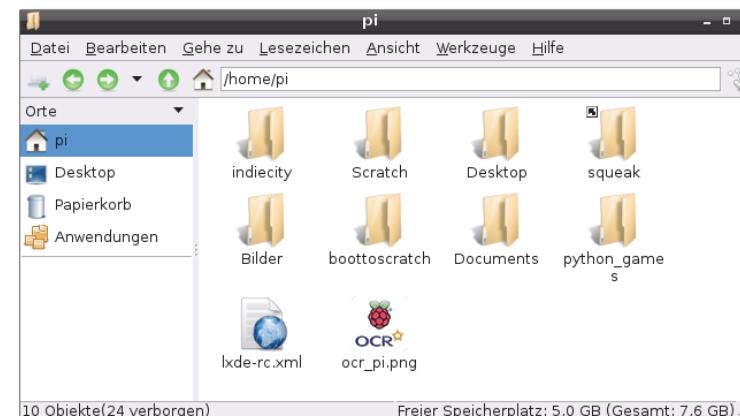


Abbildung 2.9 Der Dateimanager

In der Symbolansicht ist die Vorschau von Bildern standardmäßig auf Dateien limitiert, die kleiner als 2 MByte sind. Wenn Sie auch von größeren Bildern Miniaturen sehen möchten, müssen Sie den Grenzwert für die Vorschaufunktion im Dialogblatt **BEARBEITEN** • **EINSTELLUNGEN** • **ANSICHT** vergrößern. In diesem Dialogblatt können Sie auch die gewünschte Größe der Vorschaubilder einstellen. Standardmäßig verwendet der Dateimanager je nach Ansicht zwischen 24×24 und 128×128 Pixel.

Um Dateien zwischen zwei Verzeichnissen hin- und herzukopieren oder zu verschieben, öffnen Sie am einfachsten mit **Strg**+**T** ein neues Dialogblatt bzw. mit **Strg**+**N** ein neues Fenster.

Verborgene Dateien

Unter Linux gelten Dateien und Verzeichnisse, deren Namen mit einem Punkt beginnen, als verborgen. Diese Dateien und Verzeichnisse werden standardmäßig im Dateimanager nicht angezeigt. Um sie dennoch zu sehen und zu bearbeiten, führen Sie **ANSICHT** • **VERBORGENE DATEIEN ANZEIGEN** aus bzw. drücken **Strg**+**H**.

LXTerminal

Auch wenn Raspbian über eine grafische Benutzeroberfläche verfügt, ist es unter Linux üblich, viele administrative Aufgaben durch Kommandos im Textmodus zu erledigen. Umsteigern von Windows oder OS X mag das anfänglich sperrig und wenig intuitiv vorkommen, tatsächlich sparen Linux-Profis damit aber eine Menge Zeit. Das Programm, in dem Sie solche Kommandos ausführen, wird unter Linux *Terminal* genannt. Da überrascht es kaum, dass das Terminalprogramm für LXDE den Namen *LXTerminal* trägt.

Wenn Sie dieses Programm erstmals starten, erscheint ein fast vollständig schwarzes Fenster. Darin können Sie Kommandos eingeben und mit **Enter** ausführen. Auch die Ergebnisse und Fehlermeldungen erscheinen als Text. Wenn das düstere Erscheinungsbild des LXTerminals Sie irritiert, sollten Sie als Erstes eine hellere Hintergrundfarbe und dafür eine schwarze Textfarbe einstellen. Die entsprechenden Einstellungen finden Sie im Dialog BEARBEITEN • EINSTELLUNGEN. Etwas komplizierter ist es, auch die Sondertextfarben zu verändern, in denen die Eingabeaufforderung (der sogenannte Prompt) sowie bestimmte Dateien durch `ls` angezeigt werden. Diesbezügliche Anleitungen finden Sie in Kapitel 3.



Abbildung 2.10 Ein Terminalfenster

Text kopieren mit der Maus

Oft ist es notwendig, bei der Ausführung eines Kommandos die Ergebnisse eines anderen als Parameter anzugeben, z. B. einen Dateinamen. Wenn Sie den Text nicht abschreiben möchten, besteht der naheliegende Weg darin, den Text zuerst mit der Maus zu markieren, dann mit **Strg+Alt+C** zu kopieren und danach an der gerade aktuellen Cursorposition mit **Strg+Alt+V** wieder einzufügen. Beachten Sie, dass Sie nicht wie in anderen Programmen mit **Strg+C** und **Strg+V** arbeiten können, weil die **Strg**-Tastenkürzel im Terminal eine besondere Bedeutung haben. Insbesondere bricht **Strg+C** ein laufendes Programm vorzeitig ab.

Sofern Sie eine Maus mit drei Tasten besitzen oder ein Modell, bei dem das Mausrad gedrückt werden kann, dann können Sie sich die Fingerakrobatik für die Tastenkürzel **Strg+Alt+C** und **Strg+Alt+V** ersparen. Markieren Sie einfach den zu kopierenden Text mit der Maus, und drücken Sie dann die mittlere Maustaste. Fertig! Der zuvor markierte Text wird an der Cursorposition eingefügt. Dieses komfortable Textkopieren funktioniert übrigens nicht nur im Terminal, sondern auch in den meisten anderen Programmen.

Sollten Sie mehrere Terminal-Fenster brauchen, öffnen Sie einfach mit **Strg+N** ein weiteres. Alternativ können Sie auch mit **Strg+Tab** mehrere Tabs (Dialogblätter) öffnen und zwischen diesen wechseln.

Jetzt bleibt nur noch die Frage offen, welche Kommandos Ihnen im Terminal zur Verfügung stehen: Davon gibt es so viele, dass wir diesem Thema ein eigenes Kapitel gewidmet haben. Und selbst in Kapitel 3 können wir Ihnen nur die wichtigsten Kommandos vorstellen. Im Verlauf des Buches werden Sie noch viele weitere kennenlernen.

Leafpad und Gedit (Editoren)

Unter den vorinstallierten Texteditoren bietet das Programm *Leafpad* den meisten Komfort (siehe Abbildung 2.11). Sie starten das Programm mit SONSTIGE • LEAPAD. Der Editor ist ausreichend, um Konfigurationsdateien zu verändern, bietet aber kaum Zusatzfunktionen. Immerhin gibt es Kommandos zum Suchen und Ersetzen. Im Menü OPTIONEN können Sie die Anzeige von Zeilennummern aktivieren und die Schriftart einstellen.



Abbildung 2.11 Der minimalistische Editor Leafpad

Der Dateiauswahldialog von Leafpad zeigt standardmäßig keine vorborgenden Dateien an, also keine Dateien, deren Name mit einem Punkt beginnt. Wenn Sie eine derartige Datei öffnen möchten, klicken Sie die Dateiliste im Auswahldialog mit der rechten Maustaste an und aktivieren die Option VERBORGENE DATEIEN ANZEIGEN.

Leafpad kann immer nur *eine* Datei bearbeiten. Das Laden einer Datei ersetzt deshalb die zuletzt aktive Datei. Um dennoch mehrere Dateien gleichzeitig zu bearbeiten, öffnen Sie zuerst mit **Strg+N** ein neues, leeres Leafpad-Fenster.

Standardmäßig sind in Raspbian auch die Editoren nano und vi installiert. Diese Editoren werden direkt in einem Terminalfenster ausgeführt. Eine Zusammenfassung der wichtigsten Kürzel zur Bedienung dieser Programme finden Sie im nächsten Kapitel. Dort stellen wir Ihnen auch das Programm jmacs vor. Dieser Emacs-kompatible Editor läuft ebenfalls im Terminal.

Wenn Sie auf der Suche nach einem grafischen Texteditor sind, der mehr Funktionen als Leafpad bietet, sollten Sie sich das Programm *Gedit* ansehen (siehe Abbildung 2.12). Dazu installieren Sie das Programm zuerst in einem Terminalfenster mit `sudo apt-get install gedit` und führen es dann mit SONSTIGE • GEDIT aus. Leider setzt Gedit ziemlich viele Bibliotheken voraus. Das kostet nicht nur einige Platz auf der SD-Karte, sondern macht das Programm auch spürbar schwerfälliger als Leafpad. Dem steht als größter Vorteil das Syntax-Highlighting gegenüber: Gedit markiert also verschiedene Codeelemente in unterschiedlichen Farben.

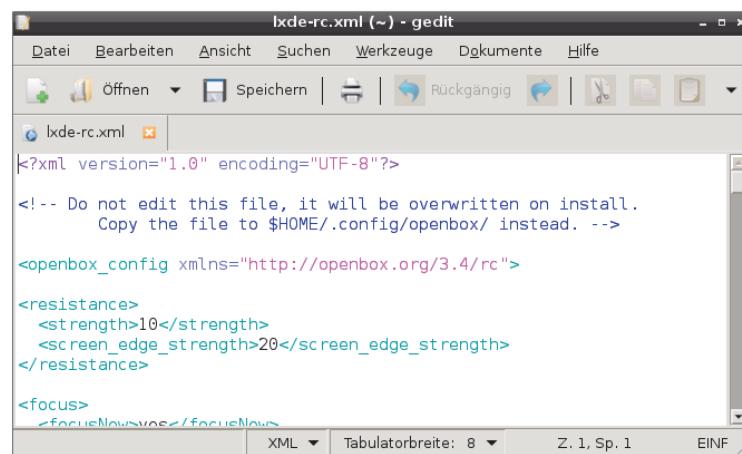


Abbildung 2.12 Der Texteditor Gedit

Firefox und Thunderbird, Chromium und Claws Mail

Wenn Sie anstelle von Midori den Webbrowser Firefox verwenden möchten, müssen Sie das Paket *iceweasel* installieren. Das klingt zugegebenermaßen absurd. Der Hintergrund ist, dass ein Programm nur Firefox bzw. Thunderbird heißen darf, solange es ausschließlich offiziellen Code der Firefox-Entwickler enthält. Um unkompliziert selbst Änderungen am Code vornehmen zu können, entschied sich Debian, den Webbrowser Firefox und den E-Mail-Client Thunderbird unter eigenen Namen in seine Distribution zu integrieren. Aus diesem Grund heißt Firefox bei Debian *Iceweasel* und Thunderbird *Icedove*. Somit lauten die Installationskommandos, die Sie in einem Terminal ausführen müssen, wie folgt:

```
sudo apt-get install iceweasel
sudo apt-get install icedove
```

Nach Abschluss der Installation starten Sie die Programme mit INTERNET • ICEWEASEL bzw. INTERNET • ICEDOVE. Sie werden feststellen, dass die Iceweasel-Version in der Regel um einige Versionen älter als die aktuelle Firefox-Version ist. Das liegt daran, dass Debiants *iceweasel*-Pakete auf der Enterprise-Version von Firefox basieren. Die Firefox-ESR-Versionen werden aber nur circa alle neun Monate aktualisiert.



Abbildung 2.13 Iceweasel alias Firefox unter Raspbian

Anstelle von Iceweasel können Sie auch den Webbrowser Chromium einsetzen. Dabei handelt es sich um einen von Google Chrome abgeleiteten Webbrowser, bei dem alle Komponenten entfernt wurden, die nicht als Open-Source-Code zur Verfügung stehen. Dazu zählt insbesondere das Flash-Plug-in:

```
sudo apt-get install chromium-browser
```

Microsoft-Fonts installieren

Viele Webseiten verwenden die von Microsoft vor vielen Jahren freigegebenen Fonts Arial, Verdana etc. Damit solche Webseiten auch auf Ihrem Raspberry Pi optimal dargestellt werden, sollten Sie den *ttf-mscorefonts-installer* installieren. Dabei handelt es sich um ein kleines Programm, das die Microsoft-Fonts herunterlädt und installiert.

Glücklich werden Sie mit Iceweasel, Icedove und Chromium voraussichtlich nicht werden. Diese sehr komplexen und funktionsreichen Programme beanspruchen eine Menge RAM und CPU-Zeit und laufen auf dem Raspberry Pi nicht flüssig.

Sollten Sie auf der Suche nach einem ressourcensparenden E-Mail-Client sein, werfen Sie am besten einen Blick auf das Programm *Claws Mail* (Paketname *claws-mail*). Das eher technisch orientierte Programm bietet viele Funktionen und unzählige Konfigurationsoptionen. Es unterstützt die gängigen Protokolle POP, IMAP und SMTP und ist damit zu den meisten Mail-Accounts kompatibel (siehe Abbildung 2.14).

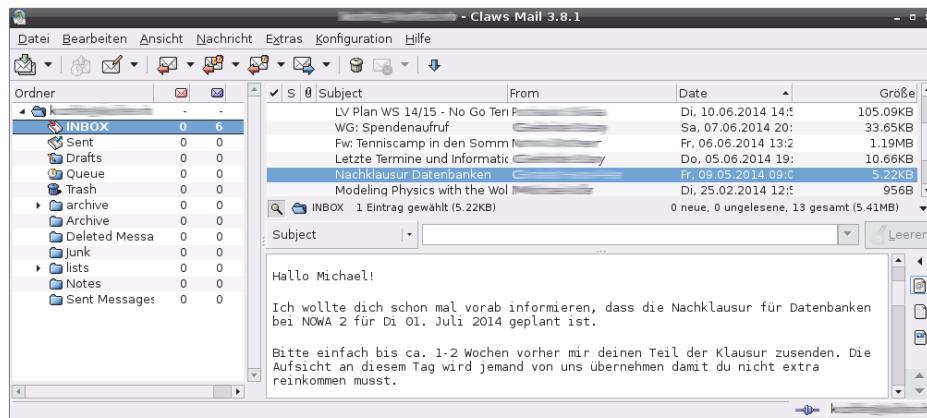


Abbildung 2.14 E-Mails auf dem Raspberry Pi lesen

Bild- und PDF-Viewer

Ein Doppelklick auf ein *.jpg- oder *.png-Bild im Dateimanager startet das Programm gpicview. Dabei handelt es sich um einen einfachen Bildbetrachter. Sie können damit durch alle weiteren Bilder des aktuellen Verzeichnisses blättern sowie Bilder drehen und spiegeln.

PDF-Dateien werden mit dem recht simplen Programm xpdf angezeigt (siehe Abbildung 2.15). Zum Blättern durch ein Datenblatt oder ein Handbuch reicht dieser Viewer aus. Mögliche Alternativen zu xpdf sind das minimalistische Programm gv oder der komfortable PDF-Viewer Evince aus den Gnome-Projekt.

```
sudo apt-get install gv
sudo apt-get install evince
```

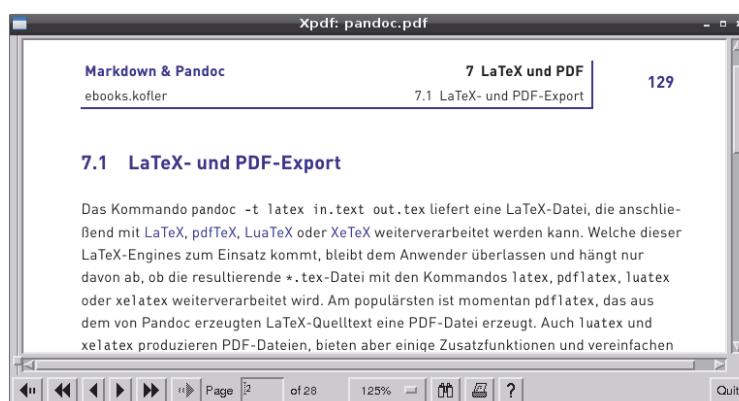


Abbildung 2.15 eBook-Lektüre mit xpdf

Audio- und Video-Player

Anfänglich ist unter Raspbian kein Audio-Player installiert, mit dem Sie MP3-Dateien abspielen können. Als kleiner, handlicher MP3-Player bietet sich das Programm LXMUSIC an (siehe Abbildung 2.16). Das Programm ist unkompliziert zu bedienen und spielt einfach die ausgewählten MP3-Dateien bzw. alle in einem Verzeichnis enthaltenen Musikdateien ab. LXMUSIC basiert intern auf xmms2, einem bewährten Linux-Audio-Player.

```
sudo apt-get install lxmusic
```

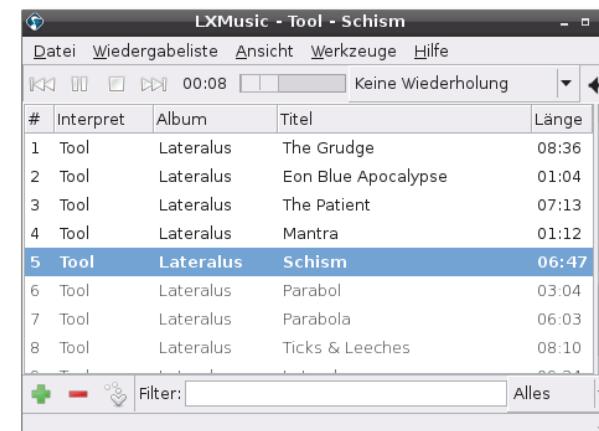


Abbildung 2.16 Audio-Dateien mit LXMUSIC abspielen

Kein Ton?

Wenn an den Raspberry Pi ein HDMI-Monitor oder -Fernseher angeschlossen ist, erfolgt die Tonwiedergabe standardmäßig via HDMI. Der Raspberry Pi besitzt aber auch einen Analog-Audio-Ausgang mit einer 3,5-mm-Klinkenbuchse. Falls Sie also zunächst nichts hören, schafft das Kommando amixer cset numid=3 1, das Sie in einem Terminalfenster ausführen müssen, Abhilfe. Es leitet die Tonausgabe auf den analogen Audio-Ausgang um.

Komfortabler als LXMUSIC, aber leider auch wesentlich schwerfälliger, ist das Programm Rhythmbox. Es greift auf unzählige andere Bibliotheken zurück, weswegen der Installationsumfang über 50 MByte beträgt. Dafür kann das Programm große MP3-Sammlungen verwalten, diese nach Genre und Interpreten filtern etc.

```
sudo apt-get install rhythmbox
```

Der Default-Video-Player von Raspbian ist das Programm omxplayer. Das schmucklose Programm kommt nur mit ausgewählten Video-Formaten bzw. -Codecs zurecht.

Dafür nutzt es aber die Hardware-Decodierfunktionen des Raspberry Pi und kann auch hochauflösende Videos ohne Ruckeln abspielen. Das setzt voraus, dass Sie die entsprechenden Lizenzschlüssel in `/boot/config.txt` eingetragen haben (siehe Abschnitt 7.2). `omxplayer` läuft grundsätzlich im Vollbildmodus. Zu seinen Vorteilen zählen eine Menge Tastenkürzel, mit denen das Programm sehr effizient gesteuert werden kann. Eine komplette Referenz aller Kürzel finden Sie auf der Projekt-Webseite:

<http://omxplayer.sconde.net>

Eine Alternative zu `omxplayer` ist das Programm *VLC*. Dieses Programm mit einer ansprechenden Benutzeroberfläche und unzähligen Spezialfunktionen beansprucht allerdings rund 100 MByte Speicherplatz auf der SD-Karte. Es kommt mit viel mehr Video-Formaten zurecht als `omxplayer`. Leider nutzt die *VLC*-Version aus den Raspbian-Paketquellen aber keine Hardware-Beschleunigung bei der Wiedergabe und ist deswegen nur für Videos in kleiner Auflösung geeignet.

```
sudo apt-get install vlc
```

Der Raspberry Pi als ferngesteuerter Audio- oder Video-Player

Zu den beliebtesten Anwendungen des Raspberry Pi zählt dessen Einsatz als Audio-Player für eine Stereo-Anlage oder als Video-Player für den Fernseher. Dabei kommen anstelle von Raspbian oft speziell für diesen Einsatzzweck optimierte Distributionen zum Einsatz. Gleich drei Kapitel dieses Buchs beschreiben verschiedene Anwendungsszenarien, die auf den Programmen MPD, XBMC und Plex aufbauen (siehe Kapitel 6, 7 und 8).

Textverarbeitung und Tabellenkalkulation

Das gängige Office-Paket unter Linux ist *LibreOffice*. Sein Einsatz auf dem Raspberry Pi ist aber nicht zu empfehlen – das Paket ist mit seinen zahllosen Funktionen einfach zu groß und zu langsam. Der Installationsumfang beträgt mehr als 200 MByte.

```
sudo apt-get install libreoffice
```

Deutlich flotter laufen das Textverarbeitungsprogramm *Abiword* und das Tabellenkalkulationsprogramm *Gnumeric* (siehe Abbildung 2.17). Wenn es also darum geht, rasch ein paar Notizen zu verfassen oder ein paar tabellarische Berechnungen durchzuführen, können wir Ihnen diese Programme empfehlen.

```
sudo apt-get install abiword
sudo apt-get install gnumeric
```

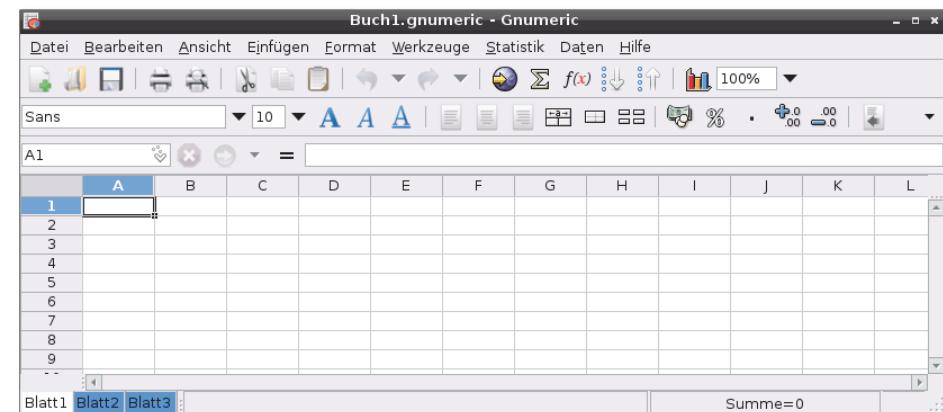


Abbildung 2.17 Das Tabellenkalkulationsprogramm Gnumeric

Screenshots

Unter Raspbian ist anfänglich kein Programm installiert, mit dem Sie Bildschirmabbildungen erstellen können. Geeignete Programme sind *xfce4-screenshooter* und *scrot*. Zur Installation führen Sie in einem Terminalfenster eines der beiden folgenden Kommandos aus:

```
sudo apt-get install scrot
sudo apt-get install xfce4-screenshooter
```

Beide Programme starten Sie aus einem Terminalfenster heraus. Wenn Sie *scrot* ohne weitere Parameter aufrufen, erstellt es eine Abbildung des gesamten Bildschirms und speichert diese im aktuellen Verzeichnis in einer PNG-Datei, deren Name sich aus dem aktuellen Datum und der Uhrzeit ergibt. Wenn Sie nur den Inhalt des gerade aktiven Fensters speichern möchten, geben Sie zusätzlich die Option `-u` ein. Um einen Screenshot des ganzen Bildschirms nach 10 Sekunden zu erstellen und die Abbildung in der Datei `bild.png` zu speichern, führen Sie dieses Kommando aus:

```
scrot -d 10 bild.png
```

xfce4-screenshooter hat sogar eine einfache Benutzeroberfläche, mit der Sie die Verzögerungszeit und den aufzunehmenden Bildbereich einstellen können (siehe Abbildung 2.18). Wir haben die meisten Abbildungen für dieses Buch mit diesem Programm erstellt. Das Hauptproblem von *scrot* besteht nämlich darin, dass es mitunter die Farben ändert. Aus einem reinen Weiß auf dem Bildschirm wird beispielsweise ein Fast-Weiß in der resultierenden Bilddatei.



Abbildung 2.18 Screenshots mit xfce4-screenshooter erstellen

2.5 WLAN-Konfiguration

Im vorigen Kapitel haben wir Ihnen geraten, Ihren Raspberry Pi für die ersten Schritte mit einem Ethernet-Kabel an das lokale Netzwerk anzuschließen. Damit ersparen Sie sich jede Konfiguration, und der Netzwerk- und Internetanschluss funktioniert auf Anhieb.

Wie Sie in diesem Abschnitt sehen werden, gelingt aber auch die WLAN-Konfiguration normalerweise rasch und unkompliziert. Dazu benötigen Sie zuallererst einen USB-WLAN-Adapter. Das sind vielfach winzige USB-Stöpsel, die nur wenige Millimeter über den eigentlichen USB-Stecker hinausragen. Entscheidend ist, dass Sie sich für ein Modell entscheiden, das von Raspbian unterstützt wird. Berücksichtigen Sie auch den Stromverbrauch: Manche USB-WLAN-Adapter erfordern in Kombination mit den Raspberry-Pi-Modellen A oder B einen aktiven USB-Hub. Gerade die winzigen Modelle ohne Antenne begnügen sich aber zumeist mit dem Strom, den die Raspberry-Pi-USB-Ports zur Verfügung stellen können. Kurzum: Recherchieren Sie vor dem Kauf und wählen Sie ein Modell, das von anderen Raspberry-Pi-Anwendern empfohlen wird.

Gleicher Modellname, unterschiedliches Innenleben

Mitunter verändern die Hersteller das Innenleben ihrer Geräte, lassen die Produktbezeichnung aber unverändert. Das kann dazu führen, dass ein Gerät, mit dem viele andere Raspberry-Pi-Anwender glücklich sind, plötzlich nicht mehr funktioniert.

Grundsätzlich ist es vorgesehen, USB-Geräte im laufenden Betrieb ein- und auszustecken. Gerade beim Arbeiten ohne ein stabiles Gehäuse ist uns dabei der Raspberry Pi mehrfach abgestürzt, vermutlich ganz einfach deswegen, weil die Stromversorgung kurzzeitig durch die mechanische Belastung unterbrochen wurde. Sicherer ist es daher, den Raspberry Pi zum Ein- und Ausstecken von USB-Geräten herunterzufahren.

Zur WLAN-Konfiguration starten Sie nun das Programm *WiFi Config*, das Sie als Desktop-Ikon sowie im Startmenü unter dem Eintrag INTERNET • WPA_GUI finden. Wenn Raspbian Ihren WLAN-Adapter erkannt hat, zeigt das Programm im Listenfeld ADAPTER (siehe Abbildung 2.19) den Namen der WLAN-Schnittstellen, in der Regel wlan0.



Abbildung 2.19 Die Benutzeroberfläche zur WLAN-Konfiguration

Nun klicken Sie auf den Button SCAN. In einem eigenen Fenster werden alle in Reichweite befindlichen Funknetze aufgelistet (siehe Abbildung 2.20). Per Doppelklick wählen Sie das gewünschte Netz aus und gelangen so in einen weiteren Dialog zur Eingabe der Verbindungsparameter (siehe Abbildung 2.21). Normalerweise müssen Sie hier nur einen einzigen Parameter angeben – das WLAN-Passwort. Die erforderliche Verschlüsselungsmethode erkennt das Programm in der Regel selbst; gegebenenfalls können Sie diese im Listenfeld AUTHENTICATION manuell einstellen.

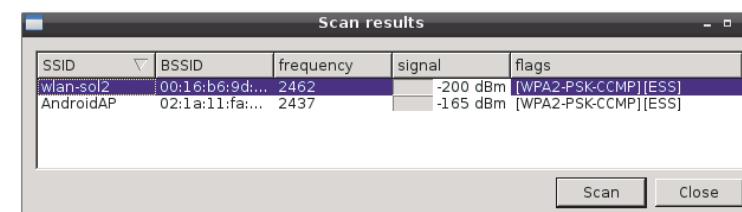


Abbildung 2.20 WLAN-Netz auswählen

Sobald die WLAN-Daten korrekt gespeichert sind, versucht Raspbian selbstständig eine Verbindung herzustellen. Den Zustand der Verbindung entnehmen Sie dem Dialogblatt CURRENT STATUS im Hauptfenster von wpa_gui. Bei Bedarf können Sie im Dialogblatt MANAGE NETWORKS die Parameter mehrerer Funknetze verwalten und ändern. Wenn Sie für mehrere WLANs Verbindungsparameter angeben, wählt

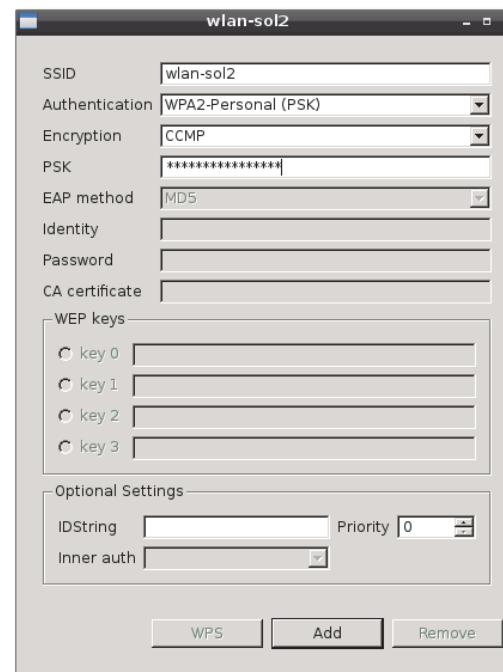


Abbildung 2.21 WLAN-Parameter eingeben

Raspbian automatisch das gerade am besten erreichbare Netz aus und verbindet sich dorthin.

Hinter den Kulissen ist für den WLAN-Verbindungsaufbau das Programm `wpa_supplicant` zuständig. Die Verbindungsparameter werden in der Datei `/etc/wpa_supplicant/wpa_supplicant.conf` gespeichert, inklusive der Passwörter im Klartext.

Noch viel mehr WLAN

In diesem Buch greifen wir das Thema WLAN noch öfter auf:

- ▶ In Kapitel 5 sind die technischen Details der WLAN- und WPA-Konfiguration beschrieben. Dort finden Sie auch Tipps, wie Sie bei Bedarf eine WLAN-Verbindung ohne die grafische Benutzeroberfläche `wpa_gui` einrichten.
- ▶ Kapitel 35 zeigt Ihnen, wie Sie Ihren Raspberry Pi als WLAN-Access-Point, -Router oder -Repeater konfigurieren – bei Bedarf sogar mit IPv6-Unterstützung.
- ▶ Wenn Sie das Internet so anonym wie möglich nutzen möchten, erklärt Kapitel 36, wie Sie aus Ihrem Raspberry Pi einen WLAN-Accesspoint mit Tor-Unterstützung machen.

2.6 Bluetooth-Konfiguration

Der Raspberry Pi enthält keinen Bluetooth-Adapter. Damit Sie Ihre Bluetooth-Maus oder -Tastatur mit dem Raspberry Pi verwenden können, benötigen Sie einen Bluetooth-USB-Adapter. Diese winzigen USB-Stöpsel können Sie bei jedem Elektronik- oder Computer-Händler für wenige Euro erwerben. Nehmen Sie sich vor dem Kauf aber ein wenig Zeit für die Recherche – der unter Raspbian laufende Linux-Kernel ist nicht zu jedem Adapter kompatibel!

Die Bluetooth-Konfiguration gelingt am einfachsten mit dem Bluetooth-Manager. Diese grafische Benutzeroberfläche ist allerdings nicht standardmäßig installiert. Das lässt sich aber leicht ändern, indem Sie ein Terminalfenster öffnen (Programm LXTerminal) und dort die beiden folgenden Kommandos ausführen:

```
sudo apt-get update
sudo apt-get install -y bluetooth bluez-utils blueman
```

`apt-get update` aktualisiert die Paketquellen. `apt-get install` installiert nicht nur die genannten Pakete, sondern darüber hinaus auch eine Menge abhängiger Pakete, von diversen Bluetooth-Bibliotheken bis hin zu Druckertreibern. Deswegen dauert die Installation mehrere Minuten. Das jeweils vorangestellte `sudo` bewirkt, dass die Programme mit Administratorrechten ausgeführt werden. Details zu den Kommandos `apt-get` und `sudo` folgen in den nächsten zwei Kapiteln.

Das frisch installierte Konfigurationsprogramm starten Sie mit **EINSTELLUNGEN • BLUETOOTH-MANAGER**. Der Button **SUCHE** bemüht sich nun, in Reichweite befindliche Bluetooth-Geräte aufzuspüren. Achten Sie darauf, dass die Geräte eingeschaltet sind, und drücken Sie, soweit vorhanden, die Verbindungstaste des Geräts.

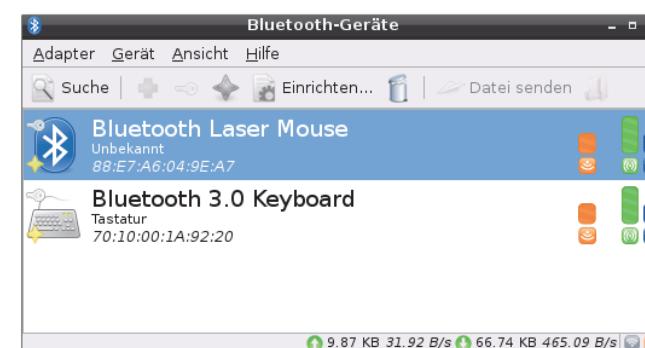


Abbildung 2.22 Bluetooth-Konfiguration

Bei den erkannten Geräten (siehe Abbildung 2.22) klicken Sie nun der Reihe nach auf die drei Buttons neben **SUCHE**. Diese haben die Funktionen **ZUR LISTE DER BEKANNTEN GERÄTE HINZUFÜGEN**, **DIESEM GERÄT VERTRAUEN** und **KOPPELUNG HERSTELLEN**.

Bei der Koppelung einer Tastatur müssen Sie auf dieser einen zufälligen Zahlencode eingeben, der im Koppelungsdialog am Bildschirm angezeigt wird. Bei einer Maus ist dies naturgemäß nicht möglich. Hier geben Sie im Koppelungsdialog an, dass zur Authentifizierung das Passwort 0000 verwendet werden soll. Bei manchen Geräten können Sie den Koppelungscode auch auf einem anderen Gerät eingeben.

Einmal gekoppelte Geräte berücksichtigt der Raspberry Pi auch nach dem nächsten Neustart korrekt. Bei unseren Tests dauerte es aber immer wieder mehrere Sekunden, bis diverse Bluetooth-Geräte auf neue Eingaben reagierten.

Bluetooth-Konfiguration im Textmodus

Wenn Sie eine Linux-Distribution ohne grafische Benutzeroberfläche verwenden, müssen Sie auf den Bluetooth-Manager verzichten und die Konfiguration im Textmodus durchführen. Wie das geht, beschreiben wir in Abschnitt 5.6.

2.7 USB-Sticks und -Festplatten

Raspbian kommt unkompliziert mit externen USB-Datenträgern zurecht, ganz egal, ob es sich um USB-Sticks oder Festplatten handelt. Beachten Sie aber, dass manche USB-Sticks und alle USB-Festplatten mehr Strom brauchen, als die USB-Schnittstellen des Raspberry Pi zur Verfügung stellen können! Verwenden Sie also gegebenenfalls einen aktiven USB-Hub oder eine USB-Festplatte mit einer eigenen Stromversorgung!



Abbildung 2.23 Raspbian erkennt einen gerade angesteckten USB-Stick

Beim Anstecken eines USB-Datenträgers erscheint auf dem Bildschirm ein Dialog, in dem Sie auswählen müssen, ob Sie den Inhalt des Datenträgers im Dateimanager öffnen möchten (siehe Abbildung 2.23). Der Datenträger wird im Verzeichnis /media/datenträgername in den Verzeichnisbaum integriert. Im Dateimanager können Sie nun Dateien kopieren und den Datenträger dann wieder lösen. Beim Lösen trat bei

unseren Tests vielfach die Fehlermeldung *failed to execute child process eject* auf, obwohl das Dateisystem fehlerfrei aus dem Verzeichnisbaum gelöst wurde. Sie können den USB-Stick anschließend also bedenkenlos abstecken.

USB-Stick formatieren

Raspbian stellt keine grafischen Werkzeuge zur Verfügung, die Ihnen dabei helfen, einen USB-Stick zu formatieren, darauf neue Partitionen einzurichten etc. Derartige Aufgaben müssen Sie durch den Aufruf der entsprechenden Kommandos im Terminal erledigen. Abschnitt 5.4 stellt Ihnen unter anderem die Kommandos `parted` und `mkfs` näher vor. Eine Anleitung, wie Sie Raspbian auf einen USB-Stick installieren, finden Sie in Abschnitt 1.4.

2.8 Drucker

Um Raspbian schlank zu halten und den Bootprozess nicht unnötig zu verlangsamen, ist standardmäßig kein Drucksystem installiert. Wenn Sie mit Ihrem Raspberry Pi einen per USB-Kabel angeschlossenen oder im lokalen Netzwerk erreichbaren Drucker nutzen möchten, müssen Sie zuerst die gesamte dafür erforderliche Infrastruktur installieren. Diese besteht aus einer Menge Paketen mit Druckertreibern, Konfigurationswerkzeugen und dem Hintergrundprogramm `cupsd`, das später die Druck-Jobs verarbeiten wird:

```
sudo apt-get install cups system-config-printer
```

Nach der Installation, die etliche Minuten dauert, müssen Sie den Benutzer pi zur Gruppe `lpadmin` hinzufügen. Das stellt sicher, dass der Benutzer pi die Druckfunktionen nutzen und administrieren kann. Damit die Einstellung wirksam wird, müssen Sie sich aus- und neu einloggen.

```
sudo usermod -a -G lpadmin pi
```

Die Druckerkonfiguration nehmen Sie nun mit **EINSTELLUNGEN • DRUCKEN** vor. Damit wird das Konfigurationsprogramm `system-config-printer` gestartet (siehe Abbildung 2.24). Mit dem Button **HINZUFÜGEN** starten Sie einen Assistenten. Dieser versucht, die angeschlossenen Drucker zu erkennen, und zeigt in einer Liste alle zur Auswahl stehenden Möglichkeiten. Nicht alle Listenpunkte spiegeln reale Drucker wider: Einige Einträge dienen als Ausgangspunkt für die Konfiguration von Druckern, die der Assistent selbst nicht erkannt hat.

Nachdem Sie sich für einen Drucker entschieden haben und auf **VOR** geklickt haben, ermittelt das Konfigurationsprogramm eine Liste mit allen passenden Treibern. Bei vielen Druckern stehen mehrere Treiber zur Auswahl. Beispielsweise können Sie

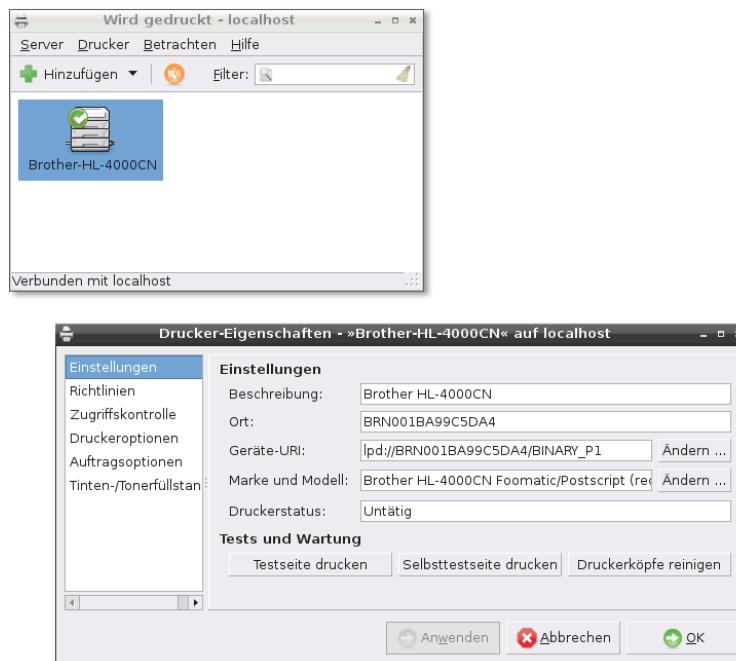


Abbildung 2.24 Druckereinstellungen

bei vielen Laser-Druckern zwischen PostScript- und HP-Laserjet-kompatiblen PCL-Treibern wählen.

Sobald die Grundkonfiguration einmal erledigt ist, können Sie alle erdenklichen Detaileinstellungen vornehmen und eine Testseite ausdrucken. Falls Sie mehrere Drucker eingerichtet haben, können Sie einen davon über den Kontextmenüeintrag **ALS STANDARD SETZEN** zum Standarddrucker machen.

Drucken im Schneckentempo

Je nachdem, in welchem Format die Druck-Jobs vorliegen und wie diese für den konfigurierten Drucker vorverarbeitet werden müssen, ist zum Drucken ein erheblicher Rechenaufwand erforderlich. Bei unseren Tests dauerte der Ausdruck der Testseite auf einem an sich schnellen Netzwerkdrucker mehrere Minuten, während denen die CPU des Raspberry Pi unter Volllast arbeitete. In Kapitel 37 gehen wir auf das Thema Drucken nochmals ausführlicher ein und zeigen Ihnen verschiedene Tricks, wie Sie die Druckgeschwindigkeit unter Umständen erheblich erhöhen können.

Kapitel 3

Arbeiten im Terminal

Erste Experimente mit dem Raspberry Pi haben Ihnen vermutlich klargemacht, dass der Raspberry Pi kein optimaler Ersatz für Ihr Notebook oder Ihren Desktop-PC ist. Die Stärken des Raspberry Pi liegen vielmehr dort, wo Sie das Gerät zur Programmierung, als *Embedded Device* (also z. B. als Internet-Radio, TV-Box oder Router) bzw. zur Steuerung elektronischer Geräte oder Komponenten einsetzen.

Diese Art der Nutzung erfordert immer Vorbereitungsarbeiten: Sie müssen Pakete installieren, Konfigurationsdateien ändern, kleine Scripts oder umfangreichere Programme erstellen, sich um automatische Backups kümmern etc. All diese Dinge erledigen Sie in der Regel im Programm *LXTerminal*, also in einem Fenster zur Ausführung von Linux-Kommandos.

Das Arbeiten im Terminal ist für erfahrene Linux-Anwender eine Selbstverständlichkeit. Wenn Sie aber bisher primär auf Windows- oder Apple-Rechnern gearbeitet haben, tauchen Sie jetzt in eine fremdartige Welt ein. Keine Angst! Sie werden in diesem Kapitel sehen, dass der Umgang mit Terminals nicht schwierig ist und dass sich mit einem Terminal viele Aufgaben sehr effizient erledigen lassen.

3.1 Erste Experimente

Bevor wir in diesem und den folgenden zwei Kapiteln systematisch viele Anwendungs- und Konfigurationsmöglichkeiten des Terminals beschreiben, gibt dieser kurze Abschnitt eine stark beispielorientierte Einführung. Sie starten also das Programm LXTerminal durch einen Doppelklick auf das Desktop-Icon oder mit ZUBEHÖR • LXTERMINAL. Per Tastatur geben Sie nun die Kommandos ein.

In diesem Buch sind die Ergebnisse der Kommandos eingerückt, um klarzumachen, was Eingabe und was Ausgabe ist. Im Terminal ist zudem jedem Kommando ein farbiger Prompt vorangestellt, also ein kurzer Text, der den eingeloggten Benutzer (zumeist pi), den Rechnernamen (standardmäßig raspberrypi) und das aktuelle Verzeichnis angibt. In diesem Buch geben wir den Prompt nicht an.

Dateien und Verzeichnisse

Nach dem Start des Terminals ist Ihr Arbeitsverzeichnis `/home/pi` das aktuelle Verzeichnis. Dieses Verzeichnis wird unter Linux auch Heimatverzeichnis genannt und mit dem Zeichen `~` abgekürzt. Mit dem Kommando `cd` wechseln Sie in andere Verzeichnisse. `ls` zeigt an, welche Dateien es im Verzeichnis gibt. `ls` mit der Option `-l` liefert ebenfalls die Liste der Dateien, allerdings ergänzt um weitere Informationen, wie die Uhrzeit und das Datum der letzten Änderung sowie die Größe in Byte.

```
cd Bilder
ls
raspbian-claws.png raspbian-firefox.png
ls -l
insgesamt 348
-rw-r--r-- 1 pi pi 168383 Jun 11 17:10 raspbian-claws.png
-rw-r--r-- 1 pi pi 103122 Jun 11 16:41 raspbian-firefox.png
-rw-r--r-- 1 pi pi 77186 Jun 13 08:56 terminal-intro.png
```

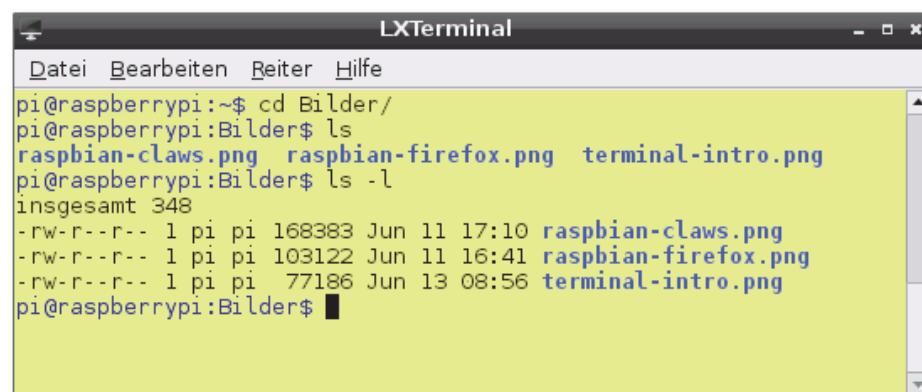


Abbildung 3.1 Erste Kommandos im Terminal

Zugriff auf Systemdateien

Die wichtigsten Konfigurationsdateien von Linux sind im Verzeichnis `/etc` gespeichert. Dort befindliche Dateien dürfen Sie als gewöhnlicher Benutzer aus Sicherheitsgründen nicht verändern und teilweise nicht einmal lesen. Um solche Dateien zu bearbeiten, müssen Sie dem betreffenden Kommando `sudo` voranstellen. `sudo` führt das Kommando dann mit Administratorrechten aus. Bei vielen Distributionen fordert `sudo` Sie zur Angabe Ihres Passworts auf. Bei Raspbian ist dies aber nicht der Fall.

Im folgenden Beispiel sucht `find` im Verzeichnis `/etc` nach Dateien, die sich während des letzten Tages geändert haben. Eine der drei gefundenen Dateien wird anschließend mit `cat` im Terminal angezeigt:

```
sudo find . -type f -mtime -1
./samba/dhcp.conf
./resolv.conf
./fake-hwclock.data
cat resolv.conf
domain lan
search lan
nameserver 10.0.0.138
```

Hardware-Tests

`lsusb` liefert eine Liste aller USB-Geräte, die der Raspberry Pi erkannt hat. `lsblk` gibt eine Liste aller Partitionen auf der SD-Karte und allen anderen Datenträgern aus. `lscpu` beschreibt die Architektur der CPU des Raspberry Pi.

```
lsusb
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 008: ID 05ac:021e Apple, Inc. ... Keyboard (ISO)
Bus 001 Device 009: ID 046d:c03e Logitech, Inc. Mouse (M-BT58)
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 0951:1665 Kingston Technology
Bus 001 Device 005: ID 2101:8500 ActionStar
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	1	7,5G	0	disk	
sda1	8:1	1	7,5G	0	part	/media/MUSIK
mmcblk0	179:0	0	7,4G	0	disk	
mmcblk0p1	179:1	0	56M	0	part	/boot
mmcblk0p2	179:2	0	7,4G	0	part	/

```
lscpu
Architecture: armv6l
Byte Order: Little Endian
CPU(s): 1
On-line CPU(s) list: 0
Thread(s) per core: 1
Core(s) per socket: 1
Socket(s): 1
```

vcgencmd

Mit dem Kommando `vcgencmd` können Sie diverse Statusinformationen der CPU auslesen. `vcgencmd commands` liefert eine Liste aller bekannten Kommandos. Die folgenden Beispiele zeigen einige Anwendungen:

```
vcgencmd measure_clock arm (CPU-Frequenz)
  frequency(45)=700072000
vcgencmd measure_clock core (Frequenz der Grafik-Cores)
  frequency(1)=250000000
vcgencmd measure_volts core (Spannung der Grafik-Cores)
  volt=1.20V
vcgencmd measure_temp (CPU-Temperatur)
  temp=47.1'C
vcgencmd codec_enabled H264 (steht Codec xy zur Verfügung?)
  H264=enabled
vcgencmd get_config int (Liste der aktiven Integer-Optionen)
  disable_overscan=1
  temp_limit=85
  force_pwm_open=1
vcgencmd get_config str (Liste der aktiven String-Optionen)
  ...
...
```

Noch mehr Anwendungsbeispiele finden Sie hier:

http://elinux.org/RPI_vcgencmd_usage

Software-Verwaltung

Sofern Ihr Raspberry Pi Internetzugang hat, können Sie mit dem Kommando `apt-get` weitere Programme installieren bzw. alle installierten Programme auf den aktuellen Stand bringen. Derartige Administratorarbeiten müssen ebenfalls mit `sudo` ausgeführt werden.

`apt-get` greift zur Software-Verwaltung auf im Internet befindliche Paketquellen zurück. Alle für den Raspberry Pi verfügbaren Programme und Erweiterungen sind dort in Form von »Paketen«, also speziellen Dateien gespeichert. `apt-get update` lädt von diesen Paketquellen die aktuelle Liste aller verfügbaren Pakete und Versionen herunter. `apt-get install` installiert das angegebene Paket – hier den Editor `joe`. Das Kommando `apt-get dist-upgrade` aktualisiert alle installierten Pakete. Dieser Vorgang kann durchaus eine Viertelstunde in Anspruch nehmen: Oft müssen viele Pakete heruntergeladen, dekomprimiert und schließlich installiert werden.

```
sudo apt-get update
Es wurden 6'926 kB in 1 min 13 s geholt (93.9 kB/s).
Paketlisten werden gelesen... Fertig
```

```
sudo apt-get install joe
```

Die folgenden NEUEN Pakete werden installiert: joe
Es müssen noch 0 B von 474 kB an Archiven heruntergeladen werden.

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

Die folgenden Pakete werden aktualisiert (Upgrade):
apt apt-utils base-files curl ...
Möchten Sie fortfahren [J/n]? j

Freien Speicherplatz ermitteln

Das Kommando `free` verrät, wie der Speicher (also das RAM) genutzt wird. Mit der Option `-m` erfolgen alle Ausgaben in MByte. Das Ergebnis ist leider nicht ganz einfach zu interpretieren. Insgesamt stehen Raspbian 437 MByte zur Verfügung. Der restliche Speicher ist für das Grafiksystem reserviert. Tatsächlich von Raspbian genutzt werden 318 MByte. Davon werden aber 21 + 224 MByte lediglich als Puffer- oder Cache-Speicher verwendet, um Dateioperationen zu beschleunigen. Dieser Speicher könnte bei Bedarf auch Programmen zugewiesen werden. Unter diesem Aspekt sind somit nur 72 MByte direkt genutzt, weitere 365 MByte sind noch verfügbar (siehe die zweite Ergebniszelle). Der Auslagerungsspeicher in der Swap-Datei ist ungenutzt.

```
free
```

	total	used	free	shared	buffers	cached
Mem:	437	318	118	0	21	224
-/+ buffers/cache:		72	365			
Swap:	99	0	99			

Wenn Sie wissen möchten, wie voll Ihre SD-Karte ist (oder, um es genauer zu formulieren, wie voll die aktiven Dateisysteme auf den Partitionen der SD-Karte sowie aller externen Datenträger sind), führen Sie `df -h` aus. Die Option `-h` bewirkt, dass die Ergebnisse nicht alle in kByte, sondern je nach Größe in MByte oder GByte präsentiert werden.

Die lange Ergebnisliste offenbart mehr Dateisysteme als erwartet. Der erste Eintrag und sein Doppelgänger in der zweiten Zeile beschreibt die Systempartition mit dem Einhängepunkt (*mount point*) /. Das darauf befindliche Dateisystem ist 7,2 GByte groß, 1,8 GByte sind noch frei. Die folgenden vier Einträge betreffen temporäre Dateisysteme, die diverse administrative Daten enthalten. Diese Daten befinden sich im RAM und stehen nur zur Verfügung, solange der Rechner läuft. Die vorletzte Zeile beschreibt die Bootpartition. Das ist ein 56 MByte kleiner Bereich der SD-Karte, der die für den Systemstart erforderlichen Dateien enthält. Die letzte Zeile zeigt schließlich, dass an den Raspberry Pi ein USB-Stick angesteckt wurde. Er enthält 2,3 GByte Daten, der restliche Speicher ist frei.

```
df -h
Dateisystem Größe Benutzt Verf. Verw% Eingeängt auf
rootfs 7,2G 5,2G 1,8G 75% /
/dev/root 7,2G 5,2G 1,8G 75% /
devtmpfs 183M 0 183M 0% /dev
tmpfs 38M 320K 38M 1% /run
tmpfs 5,0M 0 5,0M 0% /run/lock
tmpfs 75M 0 75M 0% /run/shm
/dev/mmcblk0p1 56M 26M 31M 45% /boot
/dev/sda1 7,6G 2,3G 5,3G 30% /media/MUSIK
```

Mitunter wäre es gut zu wissen, welche Verzeichnisse wie viel Speicherplatz beanspruchen. Diese Antwort gibt du. Das Kommando analysiert ausgehend vom aktuellen Verzeichnis alle Unterverzeichnisse und weist den Platzbedarf für jedes (Unter-)Verzeichnis aus. Abermals macht die Option `-h` das Ergebnis besser lesbar. Mit `--max 1` oder `--max 2` können Sie die Ausgabe auf eine oder zwei Verzeichnisebenen reduzieren, was zumeist ausreicht. Leider bietet du keine Möglichkeit, die Ergebnisse zu sortieren.

```
du -h --max 1
60K ./Desktop
5,3M ./Bilder
488K ./WolframEngine
20M ./rtl8188eu
8,0K ./Documents
...
1,1G .
```

Online-Hilfe

Kommmandos wie `ls`, `cp` oder `top`, die Sie üblicherweise in einem Terminalfenster ausführen, reagieren weder auf `F1` noch verfügen sie über ein Hilfe-Menü. Es gibt aber natürlich auch für diese Kommmandos Hilfetexte, die durch verschiedene Kommmandos gelesen werden können:

- ▶ kommando `--help` liefert bei sehr vielen Kommmandos eine Liste aller Optionen samt einer kurzen Erklärung zu ihrer Bedeutung.
- ▶ `man` kommando zeigt bei vielen Kommmandos den `man`-Hilfetext an (siehe Abbildung 3.2). `man ls` erklärt also, wie das Kommando `ls` zu nutzen ist und welche Optionen es dabei gibt. Durch den meist mehrseitigen Text können Sie mit den Cursortasten navigieren. Die Leertaste blättert eine ganze Seite nach unten. Mit `/` können Sie im Hilfetext einen Ausdruck suchen. `N` springt bei Bedarf zum nächsten Suchergebnis. `Q` beendet die Hilfe.

- ▶ `help` kommando funktioniert nur bei sogenannten Shell-Kommmandos, z.B. bei `cd` oder `alias`.
- ▶ `info` kommando ist eine Alternative zu `man`. Das `info`-System eignet sich vor allem für sehr umfangreiche Hilfetexte, allerdings ist die Navigation viel umständlicher. Ob der Hilfetext im `man`- oder `info`-System vorliegt, hängt davon ab, für welches Hilfesystem sich die Programmentwickler entschieden haben.

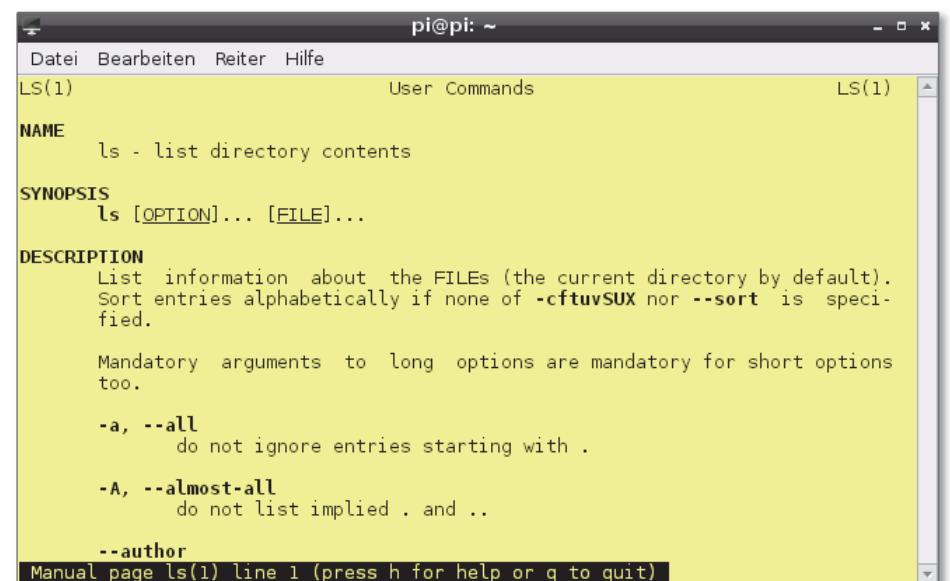


Abbildung 3.2 Der man-Hilfetext zum Kommando »ls«

3.2 Eingabeerleichterungen und Tastenkürzel

Es ist Ihnen sicher bei den ersten Experimenten schon aufgefallen, dass Sie den Cursor im Terminalfenster nicht frei bewegen können. Mit `←` und `→` bewegen Sie den Cursor innerhalb der gerade aktiven Eingabezeile. `↑` und `↓` dienen hingegen dazu, um durch früher ausgeführte Kommmandos zu blättern. Das erleichtert es, ein fehlerhaftes Kommando zu korrigieren und dann nochmals auszuführen.

Daneben gibt es etliche weitere Tastenkürzel, die Ihnen bei der Eingabe von Kommmandos behilflich sind (siehe Tabelle 3.1). Das Kürzel `Alt+B` funktioniert allerdings nur, wenn Sie mit **BEARBEITEN** • **EINSTELLUNGEN** • **VERSCHIEDENES** die Option **DISABLE ALT-N FOR MENU** aktivieren. Die meisten Tastenkürzel des Terminals sind in der Datei `/etc/inputrc` definiert.

Tastenkürzel	Bedeutung
⬅	vervollständigt ein Kommando oder einen Namen.
Strg + A	setzt den Cursor an den Beginn der Zeile.
Strg + C	bricht die Eingabe ab bzw. stoppt ein laufendes Kommando.
Strg + D	löscht das Zeichen an der Cursorposition.
Strg + E	setzt den Cursor an das Ende der Zeile.
Strg + K	löscht alles bis zum Ende der Zeile.
Strg + R abc	sucht nach einem früher ausgeführten Kommando.
Strg + Y	fügt den zuletzt gelöschten Text wieder ein.
⇪ + Strg + C	kopiert den markierten Text in die Zwischenablage.
⇪ + Strg + V	fügt die Zwischenablage an der Cursorposition ein.
Alt + ←	löscht ein Wort rückwärts.
Alt + B	bewegt den Cursor um ein Wort zurück.
Alt + F	bewegt den Cursor um ein Wort vorwärts.
Esc .	fügt den letzten Parameter des vorherigen Kommandos ein.

Tabelle 3.1 Tastenkürzel im Terminal

Das Tastenkürzel **⬅** erspart Ihnen eine Menge Tipparbeit. Sie müssen nur die ersten Buchstaben eines Kommandos, eines Verzeichnisses oder einer Datei angeben. Anschließend drücken Sie **⬅**. Wenn der Kommando-, Verzeichnis- oder Dateiname bereits eindeutig erkennbar ist, wird er vollständig ergänzt, sonst nur so weit, bis sich mehrere Möglichkeiten ergeben. Ein zweimaliges Drücken von **⬅** bewirkt, dass eine Liste aller Dateinamen angezeigt wird, die mit den bereits eingegebenen Anfangsbuchstaben beginnen.

Wenn Sie Text kopieren möchten, z.B. einen von `ls` gelieferten Dateinamen, den Sie im nächsten Kommando als Parameter verwenden möchten, gelingt dies am bequemsten mit einer Maus: Zuerst markieren Sie den Text, wobei Sie die linke Maustaste drücken. Alternativ markiert ein Doppelklick ein Wort oder ein Dreifachklick die ganze Zeile. Wenn Sie anschließend die mittlere Maustaste bzw. das Mausrad drücken, wird der markierte Text an der gerade aktuellen Cursorposition eingefügt.

Sollte Ihre Maus keine mittlere Taste haben, können Sie zuvor markierten Text auch mit **⇪ + Strg + C** kopieren und dann mit **⇪ + Strg + V** einfügen. Beachten Sie, dass diese Kürzel von den sonst üblichen **Strg + C** und **+V** abweichen, weil

Strg + C im Terminal traditionell eine andere Bedeutung hat: Das Kürzel bricht die Eingabe oder das laufende Programm ab.

Multitasking im Terminal

Wenn die Ausführung eines Kommandos länger dauert als erwartet, müssen Sie keineswegs warten und Däumchen drehen. Stattdessen öffnen Sie mit **Strg + N** ein neues Terminalfenster oder mit **Strg + T** ein neues Dialogblatt im aktuellen Fenster. Dort können Sie sofort weiterarbeiten. Wenn gerade Prozesse laufen, die viel CPU-Leistung erfordern oder eine Menge Daten auf die SD-Karte schreiben, dann wird Raspbian natürlich langsamer als sonst reagieren. Das hindert Sie aber nicht daran, in einem Editor Programmcode zu verfassen oder vergleichbare Arbeiten durchzuführen.

alias-Abkürzungen definieren

Mit der Zeit werden Sie feststellen, dass Sie manche Kommandos und Optionen häufiger benötigen als andere. Um sich ein wenig Tipparbeit zu ersparen, können Sie für solche Kommandos Abkürzungen definieren. Die Eingabe von `ll` bewirkt dann beispielsweise, dass das Kommando `ls -l` ausgeführt wird.

Abkürzungen werden ganz unkompliziert mit dem Kommando `alias` definiert:

```
alias ll='ls -l'
```

Allerdings werden mit `alias` definierte Abkürzungen nur so lange gespeichert, bis das Terminalfenster bzw. -dialogblatt geschlossen wird. Um Abkürzungen dauerhaft zu speichern, öffnen Sie die Datei `.bashrc` mit einem Editor (z. B. mit Leafpad) und fügen die gewünschten alias-Definitionen am Ende dieser Datei hinzu.

Textkonsolen

Normalerweise werden Sie Raspbian im Grafikmodus nutzen und Kommandos in Terminalfenstern ausführen. Alternativ dazu gibt es in Linux aber die sogenannten Textkonsolen. Die sind dann praktisch, wenn Sie Raspbian so konfiguriert haben, dass das Grafiksystem überhaupt nicht gestartet wird, oder wenn der Start des Grafiksystems aus irgendeinem Grund nicht funktioniert.

Unter Raspbian stehen sechs Textkonsolen zur Verfügung. Aus dem Grafikmodus führt **Strg + Alt + F1** in die erste Textkonsole und **Alt + F7** zurück in den Grafikmodus. Im Textmodus erfolgt der Wechsel zwischen den Textkonsolen mit **Alt + F1** für die erste Konsole, **Alt + F2** für die zweite etc.

Bevor Sie in einer Textkonsole arbeiten können, müssen Sie sich einloggen. Der Defaultbenutzername unter Raspbian lautet `pi`, das dazugehörige Passwort

raspberrypi, sofern Sie es noch nicht geändert haben. Wenn Sie mit der Arbeit fertig sind oder wenn Sie sich unter einem anderen Namen anmelden möchten, müssen Sie sich wieder ausloggen. Dazu drücken Sie einfach **[Strg]+D**.

Sie können in der einen Konsole ein Kommando starten, und während dieses Kommando läuft, können Sie in der zweiten Konsole etwas anderes erledigen. Jede Konsole läuft also vollkommen unabhängig von den anderen.

3.3 Farbspiele

Standardmäßig verwendet das LXTerminal eine schwarze Hintergrundfarbe. Mit BEARBEITEN • EINSTELLUNGEN können Sie unkompliziert eine hellere und augenfreundlichere Hintergrundfarbe und eine schwarze Vordergrundfarbe einstellen. Die Vordergrundfarbe gilt allerdings nur für gewöhnliche Textausgaben, nicht aber für die Eingabeaufforderung (den sogenannten Prompt) sowie für Kommandos wie ls, das Farben zur Kennzeichnung unterschiedlicher Dateitypen verwendet. In Kombination mit einem hellen Hintergrund ist der Kontrast dann leider recht schlecht. Dieser Abschnitt zeigt Ihnen, wie Sie die Farbe des Prompts und die Farben von ls so anpassen, dass die Ausgaben auch bei einem Terminal mit hellem Hintergrund gut lesbar sind.

Den Prompt individuell einrichten

Im Terminal werden am Beginn jeder Eingabezeile in grüner Schrift der Login-Name und der Hostname sowie in blauer Schrift das gerade aktuelle Verzeichnis angezeigt. Diesen Informationen folgt normalerweise das Zeichen \$; wenn Sie mit Administratortrechten arbeiten, wird hingegen das Zeichen # angezeigt.

Dieses Bündel an Informationen wird als *Prompt* bezeichnet. Für die Zusammensetzung des Prompts sowie für die Farben der verschiedenen Bestandteile ist die Umgebungsvariable PS1 verantwortlich. Unter Raspbian wird diese Variable in der Datei .bashrc eingestellt, und zwar standardmäßig wie folgt:

```
echo $PS1
\[ \e[0;\u@\h: \w\]${debian_chroot:+($debian_chroot)}\
[\033[01;32m]\u@\h[\033[00m] \[\033[01;34m\]\w \$\[\033[00m
\]
```

In dieser unleserlichen Zeichenkette ist \u ein Platzhalter für den Benutzernamen, \h für den Hostnamen, \w für das gesamte aktuelle Verzeichnis, \W für den letzten Teil des aktuellen Verzeichnisses und \\$ für den Promptabschluss (\$) oder #). Mit dem Code \[\e[0;nnm\] stellen Sie ein, in welcher Farbe (nn) die nachfolgenden Zeichen ausgege-

ben werden. Eine umfassende Anleitung zur Prompt-Konfiguration inklusive einer Auflistung aller ANSI-Farbcodes finden Sie im folgenden HOWTO-Dokument:

<http://tldp.org/HOWTO/Bash-Prompt-HOWTO>

Um den Prompt entsprechend Ihrer Vorstellungen zu gestalten, fügen Sie am Ende von .bashrc die folgenden Zeilen ein:

```
# am Ende der Datei .bashrc
...
# eigener Prompt
BLACK='\[ \e[0;30m\]'
BLUE='\[ \e[0;34m\]'
PS1="$BLUE\u@\h:\w\$ $BLACK"
unset BLACK BLUE
```

Mit dieser vergleichsweise einfachen Einstellung erreichen Sie, dass der Prompt in blauer Farbe angezeigt wird. Anders als in der Originaleinstellung wird anstelle des gesamten gerade aktuellen Verzeichnisses nur noch dessen letzter Teil angezeigt. Wenn gerade das Verzeichnis /var/www/html aktiv ist, zeigt das Terminal also nur html an.

Die neue Einstellung wird normalerweise erst gültig, wenn Sie das LXTerminal neu starten. Zum Testen ist es effizienter, die Datei .bashrc direkt im Terminal neu einzulesen. Dabei gilt der Punkt am Beginn des Kommandos als Anweisung, die folgende Datei zu verarbeiten:

```
. ~/ .bashrc
```

ls-Farben

Das Kommando ls verwendet standardmäßig Farben, um unterschiedliche Dateitypen zu kennzeichnen. Es stellt beispielsweise ausführbare Programme grün, Bilddateien rosarot, Links hellbau und Verzeichnisse dunkelblau dar. Die Farbdarstellung können Sie mit ls --color=none vermeiden. Welche Farben zum Einsatz kommen, bestimmt die Umgebungsvariable LS_COLORS. Die Defaulteinstellungen dieser Variablen werden in .bashrc mit dem Kommando dircolors erzeugt. Wenn Sie davon abweichende Einstellungen wünschen, erzeugen Sie zuerst mit dem folgenden Kommando in Ihrem Heimatverzeichnis die Datei .dircolors:

```
cd
dircolors -b > .dircolors
```

Anschließend laden Sie die Datei in einen Editor und bearbeiten sie. Die Syntax ist an sich einfach zu verstehen: Jede Zeile legt für einen bestimmten Dateityp die Darstellung fest. Dabei bestimmen mehrere, durch Strichpunkte getrennte Zahlencodes die Vorder- und Hintergrundfarbe sowie eventuell ein Zusatzattribut (fett, blinkend).

Beispielsweise bedeutet EXEC 01;32, dass ausführbare Dateien in fetter Schrift und in grüner Farbe angezeigt werden sollen. *.tar 01;31 bewirkt, dass tar-Archive ebenfalls fett, aber in roter Farbe angezeigt werden, etc.

```
# Aufbau der Datei .dircolors
# für welche Terminal-Arten gelten die Farbcodes
TERM Eterm
TERM ansi
...
# Attribut-Codes:
# 00=none 01=bold 04=underscore 05=blink 07=reverse 08=concealed
# Vordergrundfarben:
# 30=black 31=red    32=green 33=yellow
# 34=blue   35=magenta 36=cyan 37=white
# Hintergrundfarben:
# 40=black 41=red    42=green 43=yellow
# 44=blue   45=magenta 46=cyan 47=white
...
# Darstellungsregeln:
EXEC 01;32
.tar 01;31
.tgz 01;31
...
```

Das Neueinstellen aller Farbcodes ist natürlich ein mühsamer Prozess. Einfacher ist es, durch Suchen und Ersetzen bestimmte Farben durch andere zu ersetzen, also z. B. jedes Vorkommen von 32 (Grün) durch 34 (Blau). Die geänderten Einstellungen gelten, sobald Sie das Terminal neu starten.

3.4 Die Bourne Again Shell (bash)

Auf den ersten Blick erscheint es, als würde das Terminal selbst Kommandos entgegennehmen und ausführen. Das ist aber eine verkürzte Darstellung: In Wirklichkeit übergibt das Terminal Ihre Eingaben an eine sogenannte Shell. Das ist ein Kommandointerpreter, der die Eingabe auswertet und ausführt. Die Shell liefert dann die Ergebnisse zurück an das Terminal, das diese schließlich anzeigt.

Unter Linux stehen gleich mehrere Shells zur Auswahl. In Raspbian ist – wie in den meisten anderen Linux-Distributionen – standardmäßig die *Bourne Again Shell* (kurz bash) aktiv. Indirekt sind Sie auf die bash bereits gestoßen: Die vorherigen Abschnitte haben Ihnen gezeigt, wie Sie sowohl alias-Abkürzungen als auch die Konfiguration des Kommandoprompts in der Datei .bashrc durchführen können. .bashrc enthält Ihre persönlichen Konfigurationseinstellungen für die bash.

Vielleicht fragen Sie sich, warum wir uns hier mit derartigen technischen Details auseinander setzen. Der Grund dafür ist schnell erklärt: Die Shell hat großen Einfluss darauf, wie Kommandos ausgeführt werden. Sie ist dafür verantwortlich, dass *.png tatsächlich alle PNG-Dateien im aktuellen Verzeichnis verarbeitet und ermöglicht es beispielsweise, die Ergebnisse eines Kommandos mit dem nächsten weiterzuverarbeiten. Um das Terminal effizient zu nutzen, müssen Sie also die wichtigsten Mechanismen der bash kennen – und damit sind wir beim Thema für diesen Abschnitt.

Programmieren mit der bash

Die bash ist noch viel mehr als ein Komandointerpreter – sie ist eine vollständige Programmiersprache! Die bash eignet sich besonders gut dazu, Administrationsaufgaben durch Scripts zu automatisieren. Auf dem Raspberry Pi kann die bash z. B. auch verwendet werden, um verschiedene Hardware-Funktionen regelmäßig auszuführen. Eine Einführung in die Programmierung mit der bash finden Sie in Kapitel 19.

Jokerzeichen

Das Kommando rm *.bak löscht alle Dateien, deren Namen mit .bak enden. Für das Auffinden dieser Dateien ist aber nicht das rm-Kommando verantwortlich, sondern die bash. Sie durchsucht das aktuelle Verzeichnis nach passenden Dateien und ersetzt *.bak durch die entsprechenden Dateinamen. Diese Namen werden an das Kommando rm übergeben.

Zeichen	Bedeutung
?	ein beliebiges Zeichen
*	beliebig viele Zeichen (auch null)
[a,e,i,o,u]	eines der Zeichen a, e, i, o oder u
[a-f]	eines der Zeichen a bis f
[!a-f]	ein beliebiges Zeichen außer a bis f
[^a-f]	wie [!a-f]

Tabelle 3.2 Joker-Zeichen in der bash

Zur Formulierung des Suchausdrucks kennt die bash mehrere Jokerzeichen (siehe Tabelle 3.2). Wenn Sie ein Feedback haben möchten, wie die bash intern funktioniert, können Sie set -x ausführen. Die bash zeigt dann vor der Ausführung jedes weiteren Kommandos an, wie die Kommandozeile ausgewertet wird.

Ein- und Ausgabeumleitung

Bei der Ausführung von Kommandos in der bash existieren drei sogenannte Standarddateien. Der Begriff *Datei* ist dabei irreführend: Es handelt sich nicht um richtige Dateien, sondern um Dateideskriptoren, die wie Dateien behandelt werden.

- ▶ **Standardeingabe:** Das gerade ausgeführte Programm, z.B. die bash oder ein beliebiges von dort gestartetes Kommando, liest alle Eingaben von der Standardeingabe. Als Standardeingabequelle gilt normalerweise die Tastatur.
- ▶ **Standardausgabe:** Dorthin werden die Ausgaben des Programms geleitet – etwa die Dateiliste eines ls-Kommandos. Als Standardausgabe gilt das Terminal.
- ▶ **Standardfehler:** Auch Fehlermeldungen werden üblicherweise im aktuellen Terminal angezeigt.

An sich ist das alles selbstverständlich – woher sonst als von der Tastatur sollten die Eingaben kommen, wo sonst als auf dem Bildschirm sollten Ergebnisse oder Fehler angezeigt werden? Bemerkenswert ist aber die Möglichkeit, die Standardeingabe oder -ausgabe umzuleiten.

Nehmen wir an, Sie möchten die Liste aller Dateien eines Verzeichnisses nicht auf dem Bildschirm sehen, sondern diese in einer Textdatei speichern. Die Standardausgabe soll also in eine echte Datei umgeleitet werden. Das erfolgt in der bash durch das Zeichen >. Die Textdatei meine-bilder.txt enthält also nach der Ausführung des folgenden Kommandos zeilenweise die Dateinamen aller PNG-Dateien.

```
ls *.png > meine-bilder.txt
```

Die Ausgabeumleitung mit > datei ist die häufigste Anwendung. Daneben existieren viele Varianten: 2> fehler leitet die Fehlermeldungen in die angegebene Datei. >& alles bzw. &> alles leitet die Standardausgabe und die Fehlermeldungen in die angegebene Datei. Wenn Sie statt > die Verdoppelung >> verwenden, werden die jeweiligen Ausgaben an das Ende einer Datei angehängt, ohne diese zu überschreiben.

Eine Eingabeumleitung erfolgt mit < datei: Kommandos, die Eingaben von der Tastatur erwarten, lesen diese damit aus der angegebenen Datei.

Vorsicht

Sie können nicht gleichzeitig eine Datei zu bearbeiten und das Ergebnis dorthin schreiben! Die Kommandos sort datei > datei und sort < datei > datei führen beide dazu, dass die angegebene Datei gelöscht wird! Um die Datei zu sortieren, müssen Sie diese zuerst umbenennen, z.B. so:

```
mv datei tmp; sort tmp > datei; rm tmp
```

Kommandos im Hintergrund ausführen

Normalerweise wartet die bash bei jedem Kommando, bis dieses fertig ist. Es besteht aber auch die Möglichkeit, Kommandos als Hintergrundprozesse zu starten. Dazu stellen Sie dem Kommando einfach das Zeichen & hintan.

Das folgende Kommando durchsucht das aktuelle Verzeichnis und alle Unterverzeichnisse nach Dateien mit der Endung .c. Die resultierende Liste der Dateinamen wird in der Datei alle-c-dateien.txt gespeichert. Da die Suche nach den Dateien möglicherweise längere Zeit dauern wird, wollen Sie in der Zwischenzeit andere Dinge erledigen und starten das Programm mit & im Hintergrund. Sie können nun sofort weiterarbeiten. Die Information [1] 3537 bedeutet, dass es sich um den bash-Hintergrundprozess mit der Nummer 1 handelt. Die globale Prozessidentifikationsnummer lautet 3537. Sobald die Suche einige Zeit später abgeschlossen ist, werden Sie darüber informiert.

```
find . -name '*.c' > alle-c-dateien.txt &
[1] 3537
... (andere Kommandos)
[1]+ Fertig find . -name '*.py' > alle-python-dateien.txt
```

Der Programmstart im Hintergrund ist auch dann zweckmäßig, wenn das ausführende Programm eine grafische Benutzeroberfläche besitzt und daher in einem eigenen Fenster erscheint. Das trifft beispielsweise auf den Editor Leafpad oder auf den Webbrowser Midori zu. Ohne das &-Zeichen können Sie nach dem Start solcher Programme im Terminal erst weiterarbeiten, wenn das betreffende Programm beendet wird.

```
leafpad meine-datei.txt &
```

Ein Kommando nachträglich zum Hintergrundprozess machen

Nicht immer ist im Vorhinein erkennbar, dass die Ausführung eines Kommandos längere Zeit dauern wird. Mitunter werden Sie das Zeichen & auch ganz einfach vergessen. Das ist kein Problem! In solchen Fällen können Sie die Programmausführung mit **[Strg]+Z** unterbrechen und das betroffene Kommando anschließend einfach durch bg im Hintergrund fortsetzen.

Mehrere Kommandos ausführen

Um mehrere Kommandos hintereinander auszuführen, trennen Sie die Kommandos einfach durch Strichpunkte:

```
cmd1; cmd2; cmd3
```

Sie können die Kommandoabfolge auch logisch verknüpfen. `&&` bedeutet, dass das nachgestellte Kommando nur ausgeführt wird, wenn das erste erfolgreich war.

```
cmd1 && cmd2
```

Genau die umgekehrte Wirkung hat `||`: Das zweite Kommando wird nur dann ausgeführt, wenn das erste scheitert.

```
cmd1 || cmd2
```

Faszinierende Möglichkeiten ergeben sich daraus, dass ein Kommando die Ergebnisse des anderen verarbeitet. Dazu sieht die bash den Pipe-Operator `|` vor. `cmd1 | cmd2` bedeutet, dass das Ergebnis von `cmd1`, also seine Standardausgabe, an `cmd2` weitergegeben wird. `cmd2` erhält das Ergebnis als Standardeingabe.

Ein Beispiel macht das Konzept klarer: `ps ax` liefert die Liste aller laufenden Prozesse. `grep` filtert aus dieser Liste alle Prozesse heraus, die den Suchbegriff `sshd` enthalten. Das Endergebnis ist eine Liste mit den Prozessnummern aller `sshd`-Prozesse. Zusätzlich taucht auch das `grep`-Kommando selbst in der Ergebnisliste auf.

```
ps ax | grep sshd
2636 ? Ss 0:00 /usr/sbin/sshd
2758 ? Ss 0:00 sshd: pi [priv]
2776 ? S 0:00 sshd: pi@pts/0
3240 pts/0 S+ 0:00 grep --color=auto sshd
```

Eine Variante zu `cmd1 | cmd2` besteht darin, dass Sie das Ergebnis des ersten Kommandos nicht als Standardeingabe des zweiten Kommandos verwenden, sondern als Parameterliste. Die bash unterstützt diese Art der Datenweitergabe unter dem Begriff »Kommandosubstitution«, wobei es zwei gleichwertige Syntaxformen gibt:

```
cmd2 $(cmd1)
cmd2 `cmd1`
```

Auch dazu ein Beispiel: Nehmen Sie an, in einer Textdatei befinden sich zeilenweise Dateinamen. Sie möchten nun alle genannten Dateien in ein anderes Verzeichnis verschieben, kopieren, löschen etc. `cat dateien.txt` würde die Dateinamen einfach ausgeben. Mit `$(...)` wird nun diese Ausgabe als Parameterliste für ein weiteres Kommando verwendet:

```
ls -l $(cat dateien.txt)
cp $(cat dateien.txt) /in/ein/neues/verzeichnis
```

Die Grenzen der Kommandosubstitution

Meistens wollen Sie per Kommandosubstitution Dateinamen an `cmd2` weitergeben. Das funktioniert aber nur, solange die Dateinamen keine Leerzeichen enthalten. Ist dies doch der Fall, können Sie unter Umständen das Kommando `xargs --null` zuhilfnehmen.

Rechnen in der bash

Die bash ist nicht unmittelbar in der Lage, Berechnungen auszuführen. Wenn Sie `2+3` eingeben, weiß die Shell nicht, was sie mit diesem Ausdruck anfangen soll. Das ändert sich aber, sobald Sie den Ausdruck in eckige Klammern setzen und ein `$`-Zeichen voranstellen:

```
echo $[2+3]
5
```

Innerhalb der eckigen Klammern sind die meisten aus der Programmiersprache C bekannten Operatoren erlaubt: `+ - * /` für die vier Grundrechenarten, `\%` für Modulo-Berechnungen, `== != < <= > und >=` für Vergleiche, `<<` und `>>` für Bitverschiebungen, `! &&` und `||` für logische Operationen. Alle Berechnungen werden für 32-Bit-Integerzahlen ausgeführt.

Umgebungsverhalten

Manche Funktionen der bash werden durch sogenannte Umgebungsvariablen gesteuert. Umgebungsvariablen sind besondere bash-Variablen, die beim Aufruf neuer Prozesse erhalten bleiben und auch im Kontext (in der Umgebung) des neuen Prozesses gültig sind. Den Inhalt einer Umgebungsvariablen können Sie mit `echo $name` ansehen:

```
echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:\ 
/usr/local/games:/usr/games
echo $USER
pi
```

Eine Liste aller momentan definierter Umgebungsvariablen erhalten Sie, wenn Sie im Terminal `echo $` eingeben und dann zweimal `↹` drücken. Dabei leitet `$` den Namen einer Variablen ein. Mit `↹` versucht die bash, die Eingabe zu vervollständigen, und zeigt alle zur Wahl stehenden Variablen an.

In diesem Buch beschränken wir uns auf eine kurze Beschreibung der wichtigsten Umgebungsvariablen:

- ▶ **LANG:** Diese Variable enthält einen Code, der die Sprache des Benutzers und den Zeichensatz angibt. Eine übliche Einstellung im deutschen Sprachraum ist de_DE.UTF-8. Textausgaben sollen also, wenn möglich, in deutscher Sprache und im Zeichensatz UTF8 erfolgen.
- ▶ **HOME:** Diese Variable enthält das Home-Verzeichnis, z. B. /home/pi.
- ▶ **HOSTNAME:** Diese Variable enthält den Namen des Rechners, der standardmäßig raspberrypi lautet.
- ▶ **PATH:** Diese Variable enthält eine durch Doppelpunkte getrennte Liste aller Verzeichnisse, in denen die bash nach Kommandos sucht. Wenn Sie das Kommando ls ausführen, durchsucht die bash der Reihe nach die Verzeichnisse /usr/local /sbin, /usr/local/bin, /usr/sbin etc., bis sie das Kommando findet. Sollte das Kommando in keinem der in \$PATH genannten Verzeichnisse zu finden sein, liefert die bash eine Fehlermeldung.
- ▶ **PS1:** Diese Ihnen aus Abschnitt 3.3 bereits vertraute Variable definiert den Inhalt des Prompts.
- ▶ **USER:** Diese Variable enthält den aktiven Benutzer, unter Raspbian zumeist pi.

Wo sind Kommandos gespeichert?

Linux-Kommandos sind letzten Endes ganz gewöhnliche Dateien, die sich in den durch PATH aufgezählten Verzeichnissen befinden und die durch das execute-Zugriffsbit als ausführbar gekennzeichnet sind. Wenn Sie rasch wissen möchten, wo ein bestimmtes Kommando gespeichert ist, führen Sie which name aus. which liefert z. B. /bin/ls als Ergebnis. Bei manchen Kommandos funktioniert which allerdings nicht, z. B. bei cd. Das liegt daran, dass cd kein eigenständiges Kommando ist, sondern in die bash integriert ist.

3.5 Arbeiten mit Administratorrechten (sudo)

Normalerweise arbeiten Sie unter Raspbian als Benutzer pi. Damit haben Sie vollen Zugriff auf alle Dateien in Ihrem Arbeitsverzeichnis, also /home/pi. Sie können auch die meisten anderen Dateien von Raspbian lesen, aus Sicherheitsgründen aber viele nicht verändern. Administratorarbeiten darf unter Linux üblicherweise nur der Benutzer root durchführen. Auch der Zugriff auf manche Hardware-Funktionen des Raspberry Pi erfordert root-Rechte.

sudo-Anwendung

Um als root zu arbeiten, müssen Sie sich aber keineswegs aus- und neu einloggen. Stattdessen stellen Sie dem betreffenden Kommando einfach sudo voran:

```
sudo kommando
```

Anwenden von OS X und Ubuntu ist das Konzept bekannt: sudo führt das nachfolgende Kommando so aus, als hätte root das Kommando gestartet. Bei den meisten Distributionen muss bei der Ausführung von sudo das eigene Passwort angegeben werden. Unter Raspbian ist sudo aber sehr liberal konfiguriert: Wer als pi eingeloggt ist, darf sudo ohne weitere Rückfragen nutzen. Das ist zweifelsohne bequem, gleichzeitig aber auch ein erhebliches Sicherheitsrisiko.

Um Programme mit grafischer Benutzeroberfläche mit root-Rechten zu starten, also z. B. ein neues Terminalfenster, den Editor Leafpad oder den Dateimanager, führen Sie sudo mit der Option -b (*background*) aus:

```
sudo -b lxterminal
sudo -b leafpad
sudo -b pcmanfm
```

Bei umfangreicheren Administrationsaufgaben wird es zunehmend lästig, jedem Kommando sudo voranzustellen. Eleganter ist es, mit sudo -s in den root-Modus zu wechseln. Alle weiteren Kommandos werden wie von root ausgeführt. Sie beenden diesen Modus mit **[Strg]+[D]**.

```
sudo -s
kommando1
kommando2
kommando3
<Strg>+<D>
```

sudo-Konfiguration

Die Datei /etc/sudoers steuert, welcher Benutzer unter welchen Bedingungen welche Kommandos mit root-Rechten oder seltener auch als ein anderer Benutzer ausführen darf. Die Defaultkonfiguration unter Raspbian sieht so aus:

```
# Datei /etc/sudoers
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:\n/usr/sbin:/usr/bin:/sbin:/bin"
root     ALL=(ALL:ALL)  ALL
%sudo   ALL=(ALL:ALL)  ALL
pi      ALL=(ALL)      NOPASSWD: ALL
```

Kurz gefasst bedeutet das: Der Benutzer root sowie alle Mitglieder der Gruppe sudo dürfen sudo ohne Einschränkungen verwenden. Gruppen werden also durch ein vorangestelltes Prozentzeichen gekennzeichnet. Dieselben Rechte hat schließlich auch der Benutzer pi. Für diesen gilt zusätzlich die Sonderregel, dass zur sudo-Nutzung nicht einmal ein Passwort angegeben werden muss. Details zur Syntax der sudoers-Datei können Sie mit `man sudoers` nachlesen.

Wenn Sie möchten, dass auch der Benutzer pi zur sudo-Nutzung sein Passwort angeben muss, stellen Sie in `/etc/sudoers` der Zeile, die mit pi beginnt, einfach das Kommentarzeichen `#` voran. Da der Benutzer pi der Gruppe sudo angehört, wird nun die `\%sudo`-Regel wirksam. pi darf sudo also weiterhin ohne Einschränkungen benutzen, jetzt aber eben mit Passwort. Was Gruppen sind und wie diese administriert werden, erfahren Sie in Abschnitt 5.1.

Zur Veränderung von sudoers sollten Sie allerdings nicht einfach einen Editor starten, sondern vielmehr das Kommando `visudo` zuhilfe nehmen. Dieses Kommando müssen Sie selbst mit sudo starten, also so:

```
sudo visudo
```

`visudo` startet nun Ihren Defaulteditor. Beim Speichern überprüft `visudo` zuerst, ob die neue sudoers-Datei frei von Syntaxfehlern ist. Nur wenn das der Fall ist, kann der Prozess abgeschlossen werden. Diese Sicherheitsmaßnahme macht durchaus Sinn: Sollten Sie durch eine Ungeschicklichkeit sudoers so verändern, dass sudo nicht mehr funktioniert, dann könnten Sie diesen Fehler nicht mehr beheben – weil die sudoers-Datei ja nur mit root-Rechten verändert werden kann. Die Notlösung für solche Fälle ist ein Linux-Notebook: Dort stecken Sie die SD-Karte Ihres Raspberry Pi ein und korrigieren die Datei `/etc/sudoers`.

Hintergrundprozesse starten

Wenn Sie sudo durch ein Passwort absichern, funktioniert der Start von Hintergrundprozessen durch sudo kommando & nicht mehr. Das Problem ist, dass nicht nur das Kommando, sondern auch sudo selbst als Hintergrundprozess ausgeführt wird. Damit geht aber die Verbindung zur Standardeingabe und -ausgabe des Terminals verloren. Sie sehen daher keine Passwortanforderung und können das erforderliche Passwort nicht eingeben. Zum Start von Hintergrundprozessen müssen Sie vielmehr sudo -b Kommando verwenden. Um eine Systemkonfigurationsdatei mit Leafpad zu bearbeiten, führen Sie also z. B. sudo -b leafpad /etc/fstab aus.

3.6 Textdateien lesen und ändern

Die meisten Linux-Programme werden durch Konfigurationsdateien im Textformat gesteuert. Auch sonst sind Textdateien mit Programmcode oder Mess- und Logging-Daten allgegenwärtig. Dieser Abschnitt stellt Ihnen einige Kommandos vor, um Textdateien zu lesen, zu durchsuchen und zu verändern.

cat und less

Das einfachste Kommando zur Ausgabe einer Datei lautet `cat`. Es leitet einfach den gesamten Inhalt der Datei an die Standardausgabe. `cat textdatei` führt somit dazu, dass die Textdatei im Terminal zu lesen ist.

`cat` hat den Nachteil, dass es immer die *ganze* Datei ausgibt. Bei langen Texten sind dann zum Schluss nur die letzten Zeilen im Terminal zu lesen. Praktischer ist in solchen Fällen das Kommando `less`: Es ermöglicht ein zeilen- und seitenweises Blättern durch den Text (siehe Tabelle 3.3). Dieselben Tastenkürzel wie für `less` gelten übrigens auch für das Kommando `man` zur Anzeige von Hilfetexten.

Sie können `less` auch mit dem Zeichen `|` an ein anderes Kommando anhängen. Dann wird die Ausgabe des ersten Kommandos an `less` weitergeleitet, und Sie können mit `less` in Ruhe das Ergebnis lesen. Das ist vor allem bei solchen Kommandos zweckmäßig, die seitenlange Ergebnisse liefern. Im folgenden Beispiel liefert `ls -R` eine Liste aller Dateien in allen Unterverzeichnissen (die Option `-R` steht für rekursiv):

```
ls -R | less
```

Tastenkürzel	Funktion
Cursortasten	scrollen durch den Text.
<code>G</code>	springt an den Beginn des Texts.
<code>Shift</code> + <code>G</code>	springt an das Ende des Texts.
<code>/ abc</code> <code>←</code>	sucht vorwärts.
<code>? abc</code> <code>←</code>	sucht rückwärts.
<code>N</code>	wiederholt die letzte Suche (vorwärts).
<code>Shift</code> + <code>N</code>	wiederholt die letzte Suche (rückwärts).
<code>V</code>	öffnet die aktuelle Datei im Defaulteditor.
<code>Q</code>	beendet <code>less</code> .
<code>H</code>	zeigt die Online-Hilfe an.

Tabelle 3.3 less-Tastenkürzel

Wirre Zeichen im Terminal

Sollten Sie mit cat oder less versehentlich eine binäre Datei anzeigen, werden die in der Datei enthaltenen Bytecodes vom Terminal als Steuercodes interpretiert. Das kann dazu führen, dass das Terminal gewissermaßen durcheinanderkommt und weitere Ausgaben nicht mehr korrekt angezeigt. Abhilfe schafft das Kommando reset oder natürlich ein Neustart des Terminal-Programms.

tail

Bei langen Textdateien sind Sie womöglich nur an den letzten Zeilen interessiert. Das trifft besonders häufig auf Logging-Dateien oder Dateien mit Messdaten zu, in denen die gerade aktuellsten Daten in den letzten Zeilen stehen. tail datei zeigt die letzten 10 Zeilen an. Wenn Sie eine andere Zeilenanzahl wünschen, geben Sie diese mit -n <nn> an.

tail -f beobachtet die Datei und gibt automatisch neue Zeilen aus, sobald diese gespeichert werden. tail läuft in diesem Fall so lange, bis Sie die Ausführung mit **Strg+C** beenden.

```
tail -f /var/log/messages
```

Wie less kann natürlich auch tail mit | anderen Kommandos hintangestellt werden. Beispielsweise liefert dmesg alle Meldungen des Linux-Kernels seit dessen Start. Wenn Sie sich nur für die letzten fünf Zeilen interessieren, führen Sie das folgende Kommando aus:

```
dmesg | tail -n 5
```

grep

grep suchmuster dateien durchsucht die angegebenen Dateien nach dem Suchmuster und gibt alle passenden Zeilen aus. grep eignet sich dadurch auch dazu, um aus einer Datei nur die Zeilen herauszufiltern, die den Suchbegriff enthalten. Das Verhalten von grep kann durch Optionen gesteuert werden: -i bedeutet, dass die Groß- und Kleinschreibung im Suchmuster ignoriert wird. -v dreht die Wirkung von grep um: grep liefert jetzt als Ergebnis gerade die Zeilen, die den Suchbegriff *nicht* enthalten. -l bewirkt, dass grep nicht die gefundenen Textpassagen anzeigt, sondern nur die Dateinamen, in denen der Suchbegriff gefunden wurde.

grep wird besonders häufig mit dem Pipe-Operator verwendet, um die Ergebnisse anderer Kommandos auf die im jeweiligen Kontext gerade wesentlichen Informationen zu reduzieren.

Im folgenden Beispiel ermittelt dpkg -L die Liste aller installierten Pakete. grep filtert daraus Pakete heraus, die in irgendeiner Form mit Python zu tun haben.

```
dpkg -L | grep -i python
```

Texteditoren

Bisher haben Sie zwei Texteditoren mit grafischer Benutzeroberfläche kennengelernt: Das unter Raspbian standardmäßig installierte Programm Leafpad und die etwas komfortableren Alternativen Gedit. Beide Programme können Sie natürlich auch aus einem Terminalfenster heraus starten. Vergessen Sie dabei nicht das Zeichen &, um den Editor losgelöst vom Terminal in einem eigenen Hintergrundprozess zu starten:

```
leafpad textdatei &
```

Neben Editoren mit grafischer Benutzeroberfläche gibt es auch Programme, die direkt im Terminalfenster ausgeführt werden. Diese Programme haben weder eine Bildlaufleiste noch ein Menü und sind insofern schwerer zu erlernen. Dafür erfolgt ihr Start blitzschnell. Mit etwas Übung haben damit Sie kleine Änderungen bereits durchgeführt und gespeichert, bevor Leafpad auf dem Bildschirm erscheint. Das ist jetzt natürlich ein wenig übertrieben, aber wirklich nur ein wenig!

Editoren ohne Benutzeroberfläche haben noch einen wesentlichen Vorteil: Sie funktionieren auch in Textkonsolen sowie via SSH. Im Folgenden stellen wir Ihnen drei Programme näher vor:

- ▶ **nano:** Dieses Programm bietet zwar am wenigsten Funktionen, ist dafür aber besonders leicht zu erlernen. Das Programm ist standardmäßig installiert.
- ▶ **Vi:** Der Vi ist *der* Editor der Linux-Administratoren. Die Bedienung ist, gelinde gesagt, gewöhnungsbedürftig, aber wer sich einmal in Vi eingearbeitet hat, will die vielen Funktionen nicht mehr missen.
- ▶ **Emacs:** Was den Einsatz professioneller Editoren betrifft, ist die Linux-Welt gespalten. Dem Vi-Lager steht das Emacs-Lager gegenüber. Später betrachten den Emacs wegen seiner vielen Funktionen als eigenes Betriebssystem. Die Vollversion des Emacs ist auf dem Raspberry Pi aber nur in Ausnahmefällen zu empfehlen. Weit besser geeignet ist die abgespeckte Variante jemacs aus dem Paket joe.

Einige Programme starten zum Ansehen oder Editieren von Dateien selbstständig einen Editor, standardmäßig zumeist den Editor Vi bzw. den zuletzt installierten Editor. Um ein anderes Programm zum neuen Defaulteditor zu machen, führen Sie update-alternatives aus. Anschließend können Sie das gewünschte Programm festlegen:

```
sudo update-alternatives --config editor
Es gibt 8 Auswahlmöglichkeiten für die
Alternative editor (welche /usr/bin/editor bereitstellen).
```

Auswahl	Pfad	Priorität	Status
*	/usr/bin/joe	70	Auto-Modus
1	/bin/ed	-100	manueller Modus
2	/bin/nano	40	manueller Modus
3	/usr/bin/jmacs	50	manueller Modus
4	/usr/bin/joe	70	manueller Modus
5	/usr/bin/jpico	50	manueller Modus
6	/usr/bin/jstar	50	manueller Modus
7	/usr/bin/rjoe	25	manueller Modus
8	/usr/bin/vim.tiny	10	manueller Modus

Drücken Sie die Eingabetaste, um die aktuelle Wahl[*] beizubehalten,
oder geben Sie die Auswahlnummer ein: 3
update-alternatives: /usr/bin/jmacs wird verwendet, um
/usr/bin/editor (editor) im manuellen Modus bereitzustellen

nano

nano ist das optimale Programm für Linux-Einsteiger. Das Programm zeigt in den beiden unteren Terminalzeilen eine Übersicht der wichtigsten Tastenkürzel an (siehe Abbildung 3.3). Wenn der Editor den Typ der Datei erkennt – z.B., dass es sich um ein Python-Script handelt –, dann kennzeichnet er verschiedene Codekomponenten in unterschiedlichen Farben (Syntax-Highlighting).



Abbildung 3.3 Der Editor nano

Sie starten das Programm üblicherweise, indem Sie einfach den Dateinamen der zu bearbeitenden Datei übergeben:

`nano textdatei`

Systemkonfigurationsdateien können Sie nur ändern, wenn nano mit Administratorrechten läuft. Dazu stellen Sie dem Startkommando sudo voran:

`sudo nano textdatei`

Vi

Ein Urgestein der Unix-Geschichte ist der Editor Vi. Unter Raspbian ist standardmäßig das dazu kompatible Programm vim installiert, das aber nichtsdestotrotz mit dem Kommando vi gestartet wird. Der Original-Vi ist aus urheberrechtlichen Gründen nicht Teil von Linux.

Die Bedienung des Vi ist gewöhnungsbedürftig. Der wichtigste fundamentale Unterschied zu anderen Editoren besteht darin, dass der Vi zwischen verschiedenen Modi unterscheidet. Die Texteingabe ist nur im Insert-Modus möglich (siehe Tabelle 3.4). Die Eingabe der meisten Kommandos erfolgt dagegen im Complex-Command-Modus (siehe Tabelle 3.5), der mit [:] aktiviert wird. Vorher muss gegebenenfalls der Insert-Modus durch `Esc` verlassen werden.

Tastenkürzel	Funktion
<code>I</code>	wechselt in den Insert-Modus.
<code>Esc</code>	beendet den Insert-Modus.
<code>H</code>	bewegt den Cursor nach links.
<code>L</code>	bewegt den Cursor nach rechts.
<code>J</code>	bewegt den Cursor hinauf.
<code>K</code>	bewegt den Cursor hinunter.
<code>X</code>	löscht ein Zeichen.
<code>D</code> <code>D</code>	löscht die aktuelle Zeile.
<code>P</code>	fügt die gelöschte Zeile an der Cursorposition ein.
<code>U</code>	macht die letzte Änderung rückgängig.
<code>[:]</code>	wechselt in den Complex-Command-Modus.

Tabelle 3.4 Vi-Tastenkürzel

Kommando	Funktion
:w dateiname	speichert den Text unter einem neuen Namen.
:wq	speichert und beendet den Vi.
:q!	beendet den Vi, ohne zu speichern.
:help	startet die Online-Hilfe.

Tabelle 3.5 Vi-Kommandos im Complex-Command-Modus

jmacs

Der Editor Emacs steht unter Raspbian standardmäßig nicht zur Verfügung. Die Installation ist mit `apt-get install emacs` zwar problemlos möglich, aber selten zweckmäßig: Der Emacs ist ein derart komplexes und großes Programm (Installationsumfang fast 80 MByte), dass sein Einsatz auf dem Raspberry Pi aus Geschwindigkeitsgründen wenig Spaß macht.

Eine wunderbare Alternative zum originalen Emacs ist das Programm `jmacs`, das wiederum im Paket `joe` versteckt ist. `jmacs` ist in den Grundfunktionen kompatibel zum Emacs (siehe Tabelle 3.6).

```
sudo apt-get install joe
```

Tastenkürzel	Funktion
<code>[Strg]+[X], [Strg]+[F]</code>	lädt eine neue Datei.
<code>[Strg]+[X], [Strg]+[S]</code>	speichert die aktuelle Datei.
<code>[Strg]+[X], [Strg]+[W]</code>	speichert die Datei unter einem neuen Namen.
<code>[Strg]+[G]</code>	bricht die Eingabe eines Kommandos ab.
<code>[Strg]+[K]</code>	löscht eine Zeile.
<code>[Strg]+[X], [U]</code>	macht das Löschen rückgängig (Undo).
<code>[Strg]+[X], [Strg]+[C]</code>	beendet den Emacs (mit Rückfrage zum Speichern).

Tabelle 3.6 Emacs-Tastenkürzel

3.7 Verzeichnisse und Dateien

Ganz kurz die wichtigsten Fakten zu Datei- und Verzeichnisnamen:

- ▶ Unter Linux sind Dateinamen mit einer Länge bis zu 255 Zeichen zulässig.
- ▶ Es wird zwischen Groß- und Kleinschreibung unterschieden!
- ▶ Internationale Zeichen im Dateinamen sind zulässig. Raspbian verwendet wie alle anderen Linux-Distributionen Unicode in der UTF-8-Codierung als Standardzeichensatz. Aus der Sicht des Linux-Kernels ist der Dateiname einfach eine Bytefolge, in der lediglich das Zeichen / und der Code 0 nicht vorkommen dürfen.
- ▶ Dateinamen dürfen beliebig viele Punkte enthalten. `README.bootutils.gz` ist ein ganz normaler Dateiname, der andeutet, dass es sich um eine komprimierte README-Datei zum Thema Boot-Utilities handelt.
- ▶ Dateien, die mit einem Punkt beginnen, gelten als versteckte Dateien. Versteckte Dateien werden durch `ls` bzw. durch den Dateimanager normalerweise nicht angezeigt.
- ▶ Dateinamen, die bei der Eingabe von Kommandos nicht eindeutig als solche erkennbar sind, müssen in Hochkommata gestellt werden. Das gilt z. B. für Dateinamen mit Leerzeichen.
- ▶ Mehrere Verzeichnisse werden voneinander durch / getrennt, nicht wie unter Windows durch \.

Verzeichnisse

Im Terminal ist anfänglich automatisch das sogenannte Heimat- oder Home-Verzeichnis aktiv, also `/home/pi` bzw. `/home/loginname`. Eine Ausnahme gilt für den Systemadministrator root, dessen Heimatverzeichnis sich in `/root` befindet.

Mit dem Kommando `cd` wechseln Sie in ein anderes Verzeichnis. `cd` - wechselt zurück in das zuletzt aktive Verzeichnis, `cd ..` wechselt in das übergeordnete Verzeichnis, `cd` ohne weitere Parameter wechselt in das Heimatverzeichnis. Als Kurzschreibweise für das eigene Heimatverzeichnis gilt `~`. Die Heimatverzeichnisse anderer Benutzer können mit `~name` abgekürzt werden (siehe Tabelle 3.7).

```
cd                      # wechselt in das Heimatverzeichnis
cd Bilder               # wechselt in das Unterverzeichnis 'Bilder'
sudo cp *.png ~huber    # kopiert Bilder in das Heimat-
                        # verzeichnis des Benutzers huber
```

Verzeichnis	Bedeutung
.	das gerade aktuelle Verzeichnis
..	das übergeordnete Verzeichnis
~	das Heimatverzeichnis des aktuellen Benutzers
~name	das Heimatverzeichnis von name

Tabelle 3.7 Kurzschreibweise für Verzeichnisse

Neue Verzeichnisse richten Sie mit `mkdir` ein. `rmdir` löscht das Verzeichnis wieder – aber nur, wenn es leer ist. Um Verzeichnisse inklusive ihres Inhalts zu löschen, müssen Sie `rm -rf` ausführen. Passen Sie aber auf! Gelöschte Dateien können nicht wiederhergestellt werden:

```
mkdir neuesverzeichnis
rmdir neuesverzeichnis
rm -rf verzeichnis-mit-inhalt
```

Den Platzbedarf aller Dateien eines Verzeichnisses ermitteln Sie mit `du -h`. Standardmäßig listet das Kommando zuerst den Platzbedarf aller Unterverzeichnisse auf, bevor es schließlich die Endsumme angibt.

```
du -h
 166M ./Bilder
  51M ./Dokumente/gimp
 3,6M ./Dokumente/office
112M ./Dokumente
...
1,1G .
```

Wenn Sie nur am Endergebnis interessiert sind, geben Sie die zusätzliche Option `--max 0` an. Die Ausgabe des folgenden Beispiels bedeutet, dass alle Dateien im Heimatverzeichnis des aktuellen Benutzers rund 1,1 GByte beanspruchen.

```
cd
du -h --max 0
1,1G .
```

Linux-Verzeichnisbaum

Der Verzeichnisbaum von Linux beginnt im Wurzelverzeichnis `/`. Laufwerksangaben wie `C:` sind unter Linux nicht möglich. Linux-Einsteigern fällt die Orientierung im verzweigten Linux-Verzeichnisbaum oft schwer (siehe Tabelle 3.8). Im Regelfall ist es aber gar nicht erforderlich, den Verzeichnisbaum bis in die letzte Ecke zu kennen. Wichtig

sind vor allem zwei Verzeichnisse: `/home/<name>` mit Ihren persönlichen Daten sowie `/etc` mit den Dateien zur Systemkonfiguration.

Verzeichnis	Inhalt
/bin	besonders wichtige Kommandos
/dev	Device-Dateien zum Zugriff auf die Hardware
/boot	für den Startprozess erforderliche Dateien
/etc/	systemweite Konfigurationsdateien
/home/<name>	persönliche Daten des Benutzers name
/lib	besonders wichtige Bibliotheken
/media	externe Datenträger (USB-Sticks etc.)
/opt	nicht offizielle Ergänzungen zu /usr
/proc	virtuelle Dateien mit System- und Hardware-Informationen
/root	Heimatverzeichnis für den Benutzer root
/sbin	wichtige Kommandos zur Systemadministration
/sys	virtuelle Dateien mit System- und Hardware-Informationen
/usr/	Basisdaten der Linux-Distribution
/usr/bin	installierte Programme
/usr/lib	installierte Bibliotheken
/var/	veränderliche Dateien (Logging, E-Mail, Druck-Jobs)

Tabelle 3.8 Die wichtigsten Linux-Verzeichnisse

Dateien auflisten

`ls` liefert eine Liste aller Dateien eines Verzeichnisses. Mit der zusätzlichen Option `-l` zeigt `ls` außerdem zu jeder Datei an, wem sie gehört, wer sie lesen und verändern darf, wie groß sie ist etc. Wie bereits erwähnt, werden verschiedene Dateitypen durch unterschiedliche Farben hervorgehoben. Wenn Sie das nicht möchten, führen Sie `ls` mit der Option `--color=none` aus.

```
cd Bilder
ls -l *.png
-rw-r--r-- 1 pi pi 78275 Jun 13 14:01 terminal-man.png
-rw-r--r-- 1 pi pi 77909 Jun 14 08:13 terminal-nano.png
-rw-r--r-- 1 pi pi 96056 Jan 28 12:29 test.png
```

Kurz einige Anmerkungen zur Interpretation des ls-Ergebnisses: Die zehn Zeichen am Beginn der Zeile geben den Dateityp und die Zugriffsbits an. Als Dateityp kommen in Frage: der Bindestrich - für eine normale Datei, d für ein Verzeichnis (Directory), b oder c für eine Device-Datei (Block oder Char) oder l für einen symbolischen Link. Die nächsten drei Zeichen (rwx) geben an, ob der Besitzer die Datei lesen, schreiben und ausführen darf. Analoge Informationen folgen für die Mitglieder der Gruppe sowie für alle anderen Systembenutzer.

Die Zahl im Anschluss an die zehn Typ- und Zugriffszeichen gibt an, wie viele Hard-Links auf die Datei verweisen. Die weiteren Spalten geben den Besitzer und die Gruppe der Datei an (hier jeweils pi), die Größe der Datei, das Datum und die Uhrzeit der letzten Änderung und zuletzt den Dateinamen.

Wie funktionieren Zugriffsrechte?

Um zu verstehen, wer welche Dateien lesen, ändern und ausführen kann, müssen Sie die Konzepte der Benutzer- und Gruppenverwaltung sowie der Zugriffsrechte kennen. Ausführliche Informationen dazu folgen in Kapitel 5.

ls berücksichtigt normalerweise nur die Dateien des gerade aktuellen Verzeichnisses. Wenn Sie auch die Dateien aus Unterverzeichnissen einschließen möchten, verwenden Sie die Option -R. Diese Option steht übrigens auch bei vielen anderen Kommandos zur Verfügung. Das folgende Kommando listet sämtliche Dateien in allen Unterverzeichnissen auf. Diese Liste wird normalerweise recht lang. Daher leitet | less das Resultat von ls an less weiter, sodass Sie durch das Ergebnis blättern können.

```
ls -lR | less
```

Verbogene Dateien und Verzeichnisse sichtbar machen

Wir haben es ja bereits erwähnt: Dateien und Verzeichnisse, deren Namen mit einem Punkt beginnen, gelten in Linux als *verbogen*. Normalerweise gibt es in Ihrem Heimatverzeichnis eine Menge solcher Dateien und Verzeichnisse mit lokalen Einstellungen diverser Programme. ls zeigt verbogene Dateien standardmäßig nicht an. Abhilfe schafft ls -a.

Um *nur* verbogene Dateien und Verzeichnisse anzuzeigen, nicht aber gewöhnliche Dateien, führen Sie das Kommando ls -a .[^.]* aus. Warum nicht einfach ls -a .*, werden Sie sich vielleicht fragen. Weil Sie dann auch den Inhalt des aktuellen Verzeichnisses . und des übergeordneten Verzeichnisses .. zu sehen bekommen. Der Dateiname soll mit einem Punkt anfangen, das zweite Zeichen darf aber kein Punkt sein, und es darf auch nicht fehlen.

Dateien suchen

ls -R ist praktisch, um schnell rekursiv alle Dateien aufzulisten. Für eine gezielte Suche ist das Kommando aber ungeeignet. Dazu verwenden Sie besser find. Das ist ein ebenso leistungsfähiges wie komplexes Kommando zur Suche nach Dateien. Es kann verschiedene Suchkriterien berücksichtigen: ein Muster für den Dateinamen, die Dateigröße, das Datum der Erstellung oder des letzten Zugriffs etc. Eine vollständige Referenz aller Optionen gibt man find.

Ohne weitere Parameter liefert find eine Liste aller Dateien im aktuellen Verzeichnis und in allen Unterverzeichnissen. Anders als ls berücksichtigt find standardmäßig auch alle verborgenen Dateien und Verzeichnisse.

Mit der Option -name können Sie ein Suchmuster angeben, das die Dateinamen erfüllen müssen. Achten Sie darauf, das Muster in einfache Apostrophe zu stellen! Sie verhindern damit, dass die bash die im Muster enthaltenen Jokerzeichen sofort auswertet. Das folgende Kommando untersucht das Verzeichnis mycode und alle Unterverzeichnisse nach *.c-Dateien:

```
find mycode -name '*.c'
mycode/gpio-bcm2835/ledonoff.c
mycode/gpio-wiringpi/ledonoff.c
mycode/make-test/main.c
mycode/make-test/func1.c
...
```

Wenn Sie wissen möchten, welche Systemkonfigurationsdateien in /etc in der letzten Woche verändert wurden, verwenden Sie die Option -mtime zeitangabe. Die Zeitangabe erfolgt in Tagen, wobei Sie normalerweise + oder - voranstellen müssen. -mtime 7 bedeutet vor genau sieben Tagen. -mtime -7 bedeutet in den letzten sieben Tagen, -mtime +7 vor mehr als sieben Tagen.

find liefert standardmäßig sowohl Dateien als auch Verzeichnisse. Die Option -type f schränkt das Ergebnis auf Dateien ein. Die Option -ls bewirkt, dass find nicht einfach nur die Liste der Dateien liefert, sondern zu jeder Datei auch die Größe, die Zugriffsrechte etc., also ähnlich wie bei ls -l. Außerdem müssen Sie dem Kommando sudo voranstellen, weil manche Unterverzeichnisse von /etc nur für root lesbar sind.

```
sudo find /etc -type f -mtime -7
/etc/samba/dhcp.conf
/etc/resolv.conf
/etc/fake-hwclock.data
/etc/alternatives/editor
/etc/alternatives/editor.1.gz
/etc/alternatives/editorrc
...
```

Textdateien inhaltlich durchsuchen

`find` berücksichtigt nur formale Kriterien, nicht aber den Inhalt von Dateien. Wenn Sie Textdateien suchen, die ein bestimmtes Schlüsselwort enthalten, müssen Sie `find` und `grep` miteinander kombinieren. Das ist oft praktisch, die Syntax ist aber leider umständlich: In `find` können Sie mit `-exec` ein Kommando angeben, das `find` für jedes Ergebnis ausführen soll. Im Weiteren wird `{}` durch den gefundenen Dateinamen ersetzt. `\;` gibt das Ende des Kommandos an. Die nachfolgende Option `-print` ist erforderlich, weil diese sonst standardmäßig aktive Option durch `-exec` deaktiviert wurde. Bei `grep` ist wiederum die Option `-q` erforderlich, damit dieses keine Ausgaben durchführt und lediglich `find` meldet, ob die gewünschte Datei gefunden wurde oder nicht.

Das folgende Kommando durchsucht das aktuelle Verzeichnis und alle Unterverzeichnisse nach `*.c`-Dateien, in denen die Zeichenkette 'abc' vorkommt, und gibt die Dateinamen aus:

```
find . -type f -name '*.c' -exec grep -q abc {} \; -print
```

Dateien kopieren, verschieben und löschen

`cp` kopiert Dateien. Dabei gibt es zwei grundsätzliche Syntaxvarianten, die sich dadurch unterscheiden, ob der letzte Parameter ein Verzeichnis ist oder nicht:

- ▶ `cp datei1 datei2` erstellt eine Kopie von `datei1` und gibt ihr den neuen Namen `datei2`.
- ▶ `cp d1 d2 d3 v` kopiert die Dateien `d1` bis `d3` in das Verzeichnis `v`. Bei dieser Variante sind beliebig viele Parameter erlaubt. Der letzte Parameter muss ein Verzeichnis sein, alle anderen Dateien.

`cp` kann auch ganze Verzeichnisse inklusive ihres Inhalts kopieren. Dazu müssen Sie aber die Option `-r` (*recursive*) oder `-a` (*archive*) angeben. Der Unterschied zwischen den beiden Optionen besteht darin, dass sich bei der ersten Variante der Besitzer der Dateien, der Zeitpunkt der letzten Änderung etc. verändern, während diese Metadaten bei der zweiten Variante erhalten bleiben. Das folgende Kommando erstellt eine komplette Sicherheitskopie aller Dateien und Verzeichnisse in `/etc`:

```
sudo cp -a /etc/ /root/etc-bak
```

Um Dateien umzubenennen bzw. in ein anderes Verzeichnis zu verschieben, verwenden Sie das Kommando `mv`. Die Grundsyntax ist wie bei `cp`:

```
mv altername neuename
mv datei1 datei2 datei3 verzeichnis
```

Zum Löschen von Dateien verwenden Sie das Kommando `rm`. Standardmäßig kann `rm` nur Dateien löschen. Wenn Sie auch Verzeichnisse samt ihrem Inhalt löschen möchten, müssen Sie die Option `-r` angeben:

```
rm datei.bak
rm -r backup-verzeichnis
```

Mit `touch` erzeugen Sie eine neue, leere Datei. Sollte die Datei bereits existieren, dann lässt `touch` die Datei unverändert, aktualisiert aber die gespeicherte Zeit der letzten Änderung.

```
touch neue-leere-datei
```

Links

Links sind Querverweise auf Dateien oder Verzeichnisse. Unter Linux gibt es zwei Arten von Links: feste Links und symbolische Links, im Englischen also *hard links* und *soft/symbolic links*. Feste Links werden dateisystemintern effizienter gespeichert, sind aber schwerer zu handhaben. Symbolische Links sind nicht nur klarer nachzuvollziehen, sie haben zudem den Vorteil, dass sie von einem Dateisystem in ein anderes zeigen dürfen. Im manuellen Betrieb sind symbolische Links daher vorzuziehen. Backup-Systeme, die möglicherweise Tausende von Links erstellen, verwenden dagegen zumeist feste Links.

Im `/etc`-Verzeichnis gibt es unzählige symbolische Links, wovon Sie sich mit `find` rasch überzeugen können:

```
find /etc -type l
```

Um selbst symbolische Links einzurichten, verwenden Sie das Kommando `ln` quelle `ziel`. Es erstellt standardmäßig feste Links, mit der Option `-s` symbolische Links. Am einfachsten sind Links anhand eines Beispiels zu verstehen: Die folgenden Kommandos erzeugen zuerst das Verzeichnis `test` und dann darin mit `touch` die leere Datei `abc`. Mit `ln -s` wird nun ein symbolischer Link von `abc` auf `xyz` erzeugt. `ls -l` zeigt den Link anschaulich an:

```
mkdir test
cd test
touch abc
ln -s abc xyz
ls -l test
-rw-r--r-- 1 pi pi 0 Jun 14 13:51 abc
lrwxrwxrwx 1 pi pi 8 Jun 14 13:51 xyz -> abc
cd ..
rm -r test
```

Wenn Sie das Beispiel mit einem festen Link ausführen, sieht es so aus, als würden sich im Verzeichnis test zwei unabhängige Dateien befinden. In Wahrheit sind aber abc und xyz nur zwei Verweise auf ein und dieselbe Datei. ls -l gibt in der zweiten Spalte an, wie viele Links auf eine bestimmte Datei zeigen – im vorliegenden Beispiel also 2. Wenn Sie zusätzlich die Option -i verwenden, gibt ls auch den *Inode* der Datei an. Inodes sind interne Identifikationsnummern des Dateisystems. Die Übereinstimmung der Inodes beweist, dass abc und xyz durch einen festen Link verbunden sind:

```
mkdir test; cd test
touch abc
ln abc xyz
ls -l
-rw-r--r-- 2 pi pi 0 Jun 14 13:54 abc
-rw-r--r-- 2 pi pi 0 Jun 14 13:54 xyz
ls -li
131595 -rw-r--r-- 2 pi pi 0 Jun 14 13:54 abc
131595 -rw-r--r-- 2 pi pi 0 Jun 14 13:54 xyz
cd ..
rm -r test
```

3.8 Komprimierte Dateien und Archive

In Linux gibt es diverse Möglichkeiten, um Dateien zu komprimieren, um komprimierte Dateien wieder auszupacken sowie um mehrere Dateien zu einem Archiv zusammenzufassen (siehe Tabelle 3.9).

Kommando	Bedeutung
gzip	komprimiert eine Datei (Kennung .gz).
gunzip	dekompriamt die Datei wieder.
bzip2	komprimiert eine Datei (Kennung .bz2).
bunzip2	dekompriamt die Datei wieder.
xz	komprimiert eine Datei (Kennung .xz).
unxz	dekompriamt die Datei wieder.
tar	erstellt bzw. extrahiert ein Dateiarchiv.
zip	erzeugt ein Windows-kompatibles ZIP-Archiv.
unzip	extrahiert ein ZIP-Archiv.
zipinfo	zeigt Informationen über ein ZIP-Archiv an.

Tabelle 3.9 Kommandos zum Komprimieren und Archivieren von Dateien

Die wichtigsten Kommandos lernen Sie hier kennen. Im Raspberry-Pi-Alltag kommt vor allem das Dekomprimieren bzw. das Auspacken von Archiven häufig vor: Wenn Sie aus dem Internet Beispieldateien herunterladen, sind diese in der Regel komprimiert bzw. als Archiv verpackt. Die im Folgenden vorgestellten Kommandos sind aber natürlich auch für Backup-Skripts praktisch oder dann, wenn Sie mehrere Dateien per E-Mail weitergeben bzw. auf einen FTP-Server hochladen möchten.

Komprimieren und dekomprimieren

Das populärste Komprimierkommando unter Linux ist gzip. Es komprimiert die als Parameter angegebenen Dateien und benennt sie in name.gz um. gunzip funktioniert in die umgekehrte Richtung. Die beiden Kommandos verwenden den sogenannten LZ77-Lempel-Ziv-Algorithmus, der sich besonders gut für Textdateien eignet, nicht aber für Audio- oder Video-Dateien. Die Komprimierung ist verlustlos; nach dem Dekomprimieren steht die ursprüngliche Datei wieder unverändert zur Verfügung. Die folgenden Kommandos demonstrieren die Anwendung für eine CSV-Datei mit tabellarisch formatierten Messdaten, die sich besonders gut komprimieren lassen:

```
ls -l test.csv
-rw-r--r-- 1 pi pi 168325 Jun 14 16:47 test.csv
gzip test.csv
ls -l test.csv.gz
-rw-r--r-- 1 pi pi 24837 Jun 14 16:47 test.csv.gz
gunzip test.csv.gz
ls -l test.csv
-rw-r--r-- 1 pi pi 168325 Jun 14 16:47 test.csv
```

bzip2 und bunzip2 sind Alternativen zu gzip und gunzip. Der Vorteil dieser Kommandos besteht in der etwas besseren Komprimierung, der Nachteil in der etwas langsameren Ausführung. Die Dateiendung derart komprimierter Dateien ist .bz2. Anstelle von gzip und bzip2 können Sie zum Komprimieren auch xz verwenden. Damit erhalten Sie in den meisten Fällen noch kleinere Dateien. Das Komprimieren erfordert dafür noch mehr Zeit bzw. CPU-Ressourcen.

Noch mehr Komprimierkommandos

Wenn die bestmögliche Komprimierung das vorrangige Ziel ist, können Sie auch das Kommando 7zr aus dem Paket p7zip ausprobieren. Ganz anders ist die Zielsetzung des Kommandos lzop aus dem gleichnamigen Paket: Es arbeitet viel schneller als alle bisher genannten Kommandos. Dafür sind die resultierenden Dateien vergleichsweise groß (ca. 50 Prozent größer als bei gzip). Der Einsatz von lzop ist vor allem dann empfehlenswert, wenn Sie *on the fly* mit möglichst geringer CPU-Belastung komprimieren möchten, z. B. zur Übertragung einer großen Datei über eine Netzwerkverbindung.

TAR-Archive

Alle bisher genannten Kommandos komprimieren bzw. dekomprimieren *eine* Datei. Oft wollen Sie aber mehrere Dateien oder einen ganzen Verzeichnisbaum in einer möglichst kompakten Datei weitergeben oder archivieren. Unter Linux ist tar das populärste Kommando, um mehrere Dateien in einem Archiv zusammenzufassen.

Option	Bedeutung
-c	erzeugt ein Archiv (<i>create</i>).
-j	komprimiert/dekomprimiert mit bzip2/bunzip2.
-f	verwendet die angegebene Archivdatei (<i>file</i>).
-t	listet den Inhalt eines Archivs auf.
-v	zeigt an, was gerade passiert (<i>verbose</i>).
-x	packt ein Archiv aus (<i>extract</i>).
-j	komprimiert/dekomprimiert mit gzip/gunzip.

Tabelle 3.10 Die wichtigsten »tar«-Optionen

tar wird durch mehrere Optionen gesteuert (siehe Tabelle 3.10). Eine Besonderheit besteht darin, dass alle Optionen zusammen im ersten Parameter auch ohne vorangestelltes Minuszeichen angegeben werden können. tar tzf entspricht daher tar -t -z -f. Nach der Option -f bzw. als zweiter Parameter muss die Archivdatei angegeben werden. tar-Archive haben üblicherweise die Kennung .tar. Wenn das Archiv zusätzlich komprimiert ist, werden die Kennungen .tar.gz oder .tgz bzw. .tar.bz2 verwendet.

Die folgenden Beispiele verdeutlichen die Syntax von tar: tar tzf name.tgz liefert das Inhaltsverzeichnis eines komprimierten Archivs. Bei umfangreichen Archiven ist es zweckmäßig, die Ausgabe durch less zu leiten, um die Dateiliste in Ruhe lesen zu können.

```
tar tzf archiv.tar.gz
```

tar xzf packt das Archiv aus und extrahiert alle enthaltenen Dateien:

```
cd zielverzeichnis  
tar xzf archiv.tar.gz
```

Wenn Sie nur einzelne Dateien extrahieren möchten, müssen Sie diese explizit angeben, gegebenenfalls mit dem dazugehörigen Verzeichnis:

```
tar xzf archiv.tar.gz datei1 verz/datei2
```

Um ein neues Archiv zu erstellen, verwenden Sie die Optionsgruppe czf. Das folgende Kommando erstellt ein Backup aller Dateien in /etc und speichert das Archiv in der Datei /home/pi/backup-etc.tar.gz. Das Kommando muss mit sudo ausgeführt werden, weil nicht alle Dateien in /etc für gewöhnliche Benutzer lesbar sind:

```
cd /etc  
sudo tar czf /home/pi/backup-etc.tar.gz .
```

ZIP-Archive

Wenn Sie mit Windows-Anwendern kommunizieren, sind ZIP-Archive die bessere Wahl. Das folgende Kommando fügt alle als Parameter übergebenen HTML-Dateien in meinarchiv.zip ein:

```
zip meinarchiv.zip *.html
```

Wenn Sie den Inhalt ganzer Verzeichnisse archivieren möchten, geben Sie die Option -r an:

```
zip -r code-backup.zip mycode/
```

Den Inhalt einer ZIP-Datei sehen Sie sich mit zipinfo an:

```
zipinfo python-samples.zip  
Archive: python-samples.zip  
Zip file size: 12373 bytes, number of entries: 19  
-rwxr-xr-x 3.0 unx 572 tx defN 14-Mar-12 15:12 gpio-test.py  
-rwxr-xr-x 3.0 unx 1019 tx defN 14-Mar-23 21:21 grafik.py  
-rwxr-xr-x 3.0 unx 1849 tx defN 14-Mar-23 16:49 grafik-csv.py  
...
```

Zum Extrahieren des Archivs verwenden Sie unzip:

```
cd anderes-verzeichnis/  
unzip meinarchiv.zip
```

3.9 Prozessverwaltung

Zuerst ein paar Worte zur Nomenklatur: Ein Programm bzw. ein Kommando ist eigentlich nur eine ausführbare Datei. Eine Programmdatei unterscheidet sich von anderen Dateien also nur dadurch, dass das Zugriffsbit x gesetzt ist. Wie Linux Zugriffsrechte verwaltet, erfahren Sie in Abschnitt 5.2.

Linux-intern gibt es keine Unterscheidung zwischen einem Programm mit grafischer Benutzeroberfläche wie dem Dateimanager oder einem Kommando wie ls.

Umgangssprachlich werden textorientierte Programme wie `ls` aber oft als Kommandos bezeichnet.

Ein Prozess ist auf Betriebssystemebene für die Ausführung eines Programms oder Kommandos verantwortlich. Erst durch den Start eines Kommandos oder Programms wird dieses zu einem Prozess, der vom Linux-Kernel verwaltet wird.

In diesem Abschnitt lernen Sie, wie Sie feststellen, welche Prozesse gerade laufen und wie viel Rechenzeit diese beanspruchen, und wie Sie Systemprozesse starten und stoppen.

Liste aller laufenden Prozesse

Eine Liste der zurzeit laufenden Prozesse können Sie sehr einfach mit `ps` erzeugen. Ohne Optionen zeigt `ps` nur Ihre eigenen Prozesse an – und nur solche, die aus Textkonsolen bzw. Shell-Fenstern gestartet wurden. Wenn Sie `ps` die Optionenkombination `ax` hinzustellen, liefert das Kommando *alle* laufenden Prozesse, geordnet nach der Prozessidentifikationsnummer (PID). Prozesse, deren Namen in eckigen Klammern stehen, sind keine gewöhnlichen Programme, sondern Teilprozesse des Kernels.

```
ps ax
  PID TTY      STAT   TIME  COMMAND
    1 ?        Ss     0:05  init [2]
    2 ?        S      0:00  [kthreadd]
    3 ?        S      0:10  [ksoftirqd/0]
...
  6903 pts/1    S      0:03  leafpad
  7075 pts/0    R+    0:00  ps ax
```

`ps` kann durch zahllose Optionen gesteuert werden, wobei viele Optionen ohne das sonst übliche vorangestellte Minuszeichen angegeben werden. Im folgenden Beispiel wurde die Liste der Prozesse aus Platzgründen stark gekürzt. Unter Raspbian laufen normalerweise rund 100 Prozesse zugleich. Eine schnellere Möglichkeit, die Prozesse zu zählen, bietet das Kommando `wc` (*word count*). Es zählt je nach Option die Zeichen, Wörter oder Zeilen eines Texts.

```
ps ax | wc -l
  111
```

Praktischer als `ps` ist meist `top` (siehe Abbildung 3.4): Dieses Kommando ordnet die Prozesse danach, wie sehr sie die CPU belasten, und zeigt die gerade aktiven Prozesse zuerst an. Das Programm gibt auch einen Überblick über den aktuellen Speicherbedarf etc. Die Prozessliste wird alle paar Sekunden aktualisiert, bis das Programm mit `Q` beendet wird.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2383	root	20	0	56456	14m	4808	S	37,1	3,9	2:15.37	Xorg
2669	pi	20	0	16192	7056	4728	S	6,0	1,8	0:08.12	openbox
2846	pi	20	0	121m	9608	8	R	5,0	3,3	0:13.10	lxterminal
7135	pi	20	0	5220	1368	1028	R	5,0	0,4	0:00.19	top
2675	pi	20	0	135m	13m	10m	S	2,0	3,7	0:11.30	pcmanfm
1999	icecast2	20	0	11948	2192	1624	S	1,0	0,6	4:54.45	icecast2
2071	nobody	20	0	2020	632	520	S	1,0	0,2	0:03.97	thd
2673	pi	20	0	106m	9,8m	7552	S	1,0	2,6	4:06.61	lxpanel

Abbildung 3.4 »top« ordnet die Prozesse nach CPU-Last

Der Wert in der PID-Spalte gibt die Prozessnummer an. Wenn Sie diese Nummer kennen, können Sie außer Kontrolle geratene Programme oder Hintergrundprozesse mit dem Kommando `kill` gewaltsam stoppen.

Prozesse können verschiedene Zustände annehmen. Die zwei häufigsten Zustände sind *R* (*running*) und *S* (*sleeping*, das Programm hat also gerade nichts zu tun und wartet auf Eingaben). Programme können auch vorübergehend unterbrochen werden und weisen dann den Zustand *T* (*stopped*) auf.

`top` nimmt auch interaktiv Kommandos entgegen. Damit können Sie Prozesse stoppen (`K`, `kill`) oder ihre Priorität verändern (`R`, `renice`).

Eine grafische Variante zu `top` ist der Taskmanager, den Sie im Startmenü mit SYSTEMWERKZEUGE • TASKMANAGER starten (siehe Abbildung 3.5). Im Leerlaufbetrieb beansprucht der Taskmanager freilich selbst die meiste Rechenzeit.

Befehl	Benutzer	CPU%	RSS	Speicher	PID	Status	Prio	PPID
lxtask	pi	16%	9,5 MB	106,0 MB	7201	R	0	1
openbox	pi	12%	7,2 MB	16,6 MB	2669	R	0	2578
Xorg	root	5%	14,1 MB	54,9 MB	2383	S	0	2359
pcmanfm	pi	4%	13,9 MB	135,6 MB	2675	S	0	2578
thd	root	0%	632,0 KB	2,0 MB	2071	S	0	1
lxpanel	pi	0%	10,1 MB	106,1 MB	2673	S	0	2578
leafpad	root	0%	9,7 MB	107,0 MB	6903	S	0	6902
gvfsd-trash	pi	0%	3,1 MB	9,6 MB	6914	S	0	1
sudo	root	0%	1,7 MB	5,7 MB	6902	S	0	2848
bash	pi	0%	3,8 MB	7,1 MB	4406	S	0	4405
sshd	root	0%	1,5 MB	9,6 MB	4405	S	0	4400

Abbildung 3.5 Der Taskmanager

Prozesshierarchie

Intern wird mit jedem Prozess auch die PID-Nummer des Elternprozesses gespeichert. Diese Information ermöglicht die Darstellung eines Prozessbaums, an dessen Spitze unter Raspbian immer der Prozess init steht. init ist das erste Programm, das unmittelbar nach dem Laden des Kernels gestartet wird. Details dazu können Sie in Abschnitt 5.7 nachlesen.

Zur Darstellung der Prozesshierarchie rufen Sie das Kommando `pstree` auf. Mit der Option `-h` werden die Elternprozesse zum gerade laufenden Prozess fett hervorgehoben (siehe Abbildung 3.6).

```
pi@pi:~$ pstree -h
init─ apache2─5*[apache2]
      └─ avahi-daemon─ avahi-daemon
      └─ blueman-applet
      └─ bluetoothd
      └─ console-kit-dae─ 64*[console-kit-dae]
      └─ cron
      └─ cupsd
      └─ 2*[dbus-daemon]
      └─ dbus-launch
      └─ dhclient
      └─ dnsmasq
      └─ gconfd-2
      └─ 7*[getty]
      └─ gvfs-afc-volume─ {gvfs-afc-volume}
      └─ gvfs-gdu-volume
      └─ gvfs-gphoto2-vo
      └─ gvfsd
      └─ gvfsd-trash
      └─ icecast2─ 5*[icecast2]
      └─ 2*[ifplugd]
      └─ lightdm─ Xorg
          └─ lightdm─ lxsession─ lxpanel
              └─ openbox
              └─ pcmanfm
              └─ ssh-agent
              └─ {lxsession}
          └─ 2*[{lightdm}]
          └─ {lxpolkit}
          └─ lxterminal─ bash─ pstree
              └─ sudo─ leafpad
              └─ gnome-pty-help
              └─ {lxterminal}
          └─ menu-cached
          └─ mpd─ 2*[mpd]
          └─ notification-da─ {notification-da}
          └─ ntpd
          └─ obex-data-serve
          └─ polkitd─ {polkitd}
          └─ rsyslogd─ 3*[rsyslogd]
          └─ sshd─ sshd─ bash
          └─ sudo─ leafpad
          └─ thd
          └─ udevd─ 2*[udevd]
          └─ udisks-daemon─ udisks-daemon
              └─ 2*[{udisks-daemon}]
pi@pi:~$
```

Abbildung 3.6 »pstree« zeigt, welcher Prozess von welchem gestartet wurde.

Prozesse abbrechen

Normalerweise endet ein Prozess mit dem Programmende. Aber leider kommt es auch unter Linux vor, dass Programme Fehler enthalten, sich nicht mehr stoppen lassen und womöglich immer mehr Speicher und CPU-Kapazität beanspruchen. In solchen Fällen muss der Prozess gewaltsam beendet werden. Bei textorientierten Kommandos hilft in den meisten Fällen einfach `Strg+C`. Damit wird das Programm sofort beendet.

Das Kommando `kill` versendet Signale an einen laufenden Prozess, der durch die PID-Nummer spezifiziert wird. Diese Nummer können Sie mit `top` oder `ps` ermitteln. Um ein Programm *höflich* zu beenden, wird das Signal 15 verwendet. `kill` verwendet dieses Signal per Default. Hilft das nicht, muss das Signal 9 eingesetzt werden (hier für den Prozess 2725). `kill -9 2725` zwingt den Prozess, sich sofort zu beenden.

```
kill -9 2725
```

`kill` kann nur für eigene Prozesse verwendet werden. Zum Beenden fremder Prozesse müssen Sie `kill` mit Administratorrechten ausführen, also mit `sudo`.

Auch mit `top` können Sie Prozesse beenden: Geben Sie einfach `K` und anschließend die Prozessnummer und das gewünschte Signal ein!

`killall` ist eine Variante zu `kill`. Dabei wird nicht die Prozessnummer, sondern der Programmname angegeben. Es werden *alle* Prozesse dieses Namens beendet.

```
killall lxterminal
```

Prozesspriorität einstellen

Wenn mehrere Prozesse gleichzeitig laufen, entscheidet der Linux-Kernel, welcher Prozess wie viel CPU-Zeit erhält. In manchen Fällen ist es sinnvoll, einem Prozess bewusst mehr oder weniger Rechenzeit zuzuteilen. Dazu dient das Kommando `nice`, mit dem Programme mit reduzierter oder erhöhter Priorität gestartet werden können. Dazu wird an `nice` die gewünschte Priorität übergeben, die von 19 (ganz niedrig) bis -20 (ganz hoch) reicht. Per Default werden Prozesse mit der Priorität 0 gestartet. Im folgenden Beispiel wird ein Backup-Programm mit niedrigerer Priorität gestartet, damit es keine anderen Prozesse beeinträchtigt. Es spielt ja normalerweise keine Rolle, ob das Backup ein paar Sekunden länger dauert.

```
nice -n 10 ./mein-backup-script
```

Mit `renice` kann auch die Priorität von bereits laufenden Prozessen geändert werden. Als Parameter muss die Prozess-ID angegeben werden, die vorher mit `top` oder `ps` ermittelt wurde. Allerdings kann nur root Programme mit einer höheren Priorität als 0 starten bzw. die Priorität eines bereits laufenden Prozesses erhöhen.

Systemdienste steuern

Nach einem Blick auf die Ergebnisse von `ps ax` oder `pstree` fragen Sie sich vielleicht, woher die vielen Prozesse kommen. Nur wenige Prozessnamen lassen sich selbst gestarteten Kommandos oder Programmen zuordnen. Die restlichen Prozesse wurden automatisch gestartet, während der Rechner hochfuhr. Viele von ihnen sind der grafischen Benutzeroberfläche zugeordnet. Die restlichen Prozesse sind überwiegend Systemdienste und stellen beispielsweise Netzwerkfunktionen zur Verfügung.

Ein Beispiel dafür ist das Programm `sshd`, der Secure-Shell-Dämon. Ein »Dämon« ist die Unix-typische Bezeichnung für einen Systemdienst. `sshd` erlaubt es Ihnen, sich über das Netzwerk bei Ihrem Raspberry Pi anzumelden (siehe Abschnitt 4.3).

Systemdienste wie `sshd` werden normalerweise automatisch gestartet, und es gibt nur selten die Notwendigkeit, in diesen Automatismus einzugreifen. Wenn dies aber doch der Fall ist, hilft Ihnen das Kommando `service`, Systemdienste zu starten, zu stoppen, neu zu starten etc. Die Syntax für `service` sieht so aus:

```
sudo service dienstname aktion
```

Aktion	Bedeutung
start	startet den Systemdienst.
stop	stoppt den Systemdienst.
restart	startet den Systemdienst neu.
reload	fordert den Dienst auf, die Konfiguration neu einzulesen.
status	zeigt den Zustand des Dienstes an.

Tabelle 3.11 »service«-Aktionen

Wenn Sie also die Konfiguration von `sshd` geändert haben und möchten, dass das Programm die geänderten Einstellungen berücksichtigt, führen Sie das folgende Kommando aus:

```
service ssh reload
```

Vielleicht fragen Sie sich, warum es nicht `service sshd reload` heißt. Das ist zugegebenermaßen inkonsistent und liegt daran, welchen Namen die sogenannten Init-Scripts haben. Das Init-Script für `sshd` hat den Namen `ssh`. Wenn Sie nicht wissen, wie das Init-Script für ein bestimmtes Programm heißt, werfen Sie einen Blick in das Verzeichnis `/etc/init.d`. Was Init-Scripts überhaupt sind, ist in Abschnitt 5.7 beschrieben.

Manche Systemdienste unterstützen die Aktion `reload` nicht. In solchen Fällen müssen Sie `restart` ausführen. Bei Netzwerkdiensten hat `restart` aber den Nachteil, dass alle gerade aktiven Verbindungen unterbrochen werden.

Die mit `service` durchgeführte Aktion wird unmittelbar ausgeführt. Beim nächsten Neustart des Raspberry Pi gelten aber wieder die Defaulteinstellungen für den Systemdienst. Unter Debian und somit auch unter Raspbian ist es üblich, dass einmal installierte Systemdienste standardmäßig gestartet werden. Wollen Sie das nicht, besteht die einfachste Lösung darin, das betreffende Paket zu deinstallieren. Alternativ dazu können Sie mit `insserv -r` den automatischen Start unterbinden.

```
insserv -r ssh
```

Wenn Sie sich später dafür entscheiden, dass `sshd` wieder standardmäßig gestartet werden soll, aktivieren Sie den Systemdienst mit `insserv` erneut (diesmal also ohne die Option `-r`):

```
insserv ssh
```

3.10 Netzwerkkommandos

Netzwerkstatus ermitteln

Das Kommando `ip addr` liefert eine Liste aller bekannten Netzwerkschnittstellen und gibt an, welche IP-Adressen ihnen zugeordnet sind. Die Ausgaben im folgenden Beispiel sind so zu interpretieren: Die Loopback-Schnittstelle `lo` ist wie auf jedem Linux-System mit der IP-Adresse `127.0.0.1` verbunden. Diese Schnittstelle ermöglicht es, dass Programme des Raspberry Pi miteinander über Netzwerkprotokolle kommunizieren können. Dazu muss es keine Netzwerkverbindung nach außen geben.

Die Ethernet-Schnittstelle `eth0` hat die IP-Adresse `10.0.0.8`. Das ist vermutlich die wichtigste Information für Sie, weil Ihr Raspberry Pi unter dieser Adresse im lokalen Netzwerk erreichbar ist. `ip addr` zeigt schließlich noch die Schnittstelle `wlan0` an, allerdings ohne IP-Adresse. Das kann daran liegen, dass die WLAN-Verbindung noch nicht konfiguriert ist oder dass es momentan keine WLAN-Netze in Funkreichweite gibt.

Fortgeschrittenen Linux-Anwendern wird vielleicht auffallen, dass `ip addr` keine IPv6-Daten anzeigt, wie dies unter anderen Linux-Distributionen üblich ist. Das liegt daran, dass IPv6 auf dem Raspberry Pi standardmäßig deaktiviert ist. Wie Sie das bei Bedarf ändern können, erfahren Sie in Abschnitt 4.9.

```
ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc ...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ...
    link/ether b8:27:eb:c2:4d:4b brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.8/24 brd 10.0.0.255 scope global eth0
        valid_lft forever preferred_lft forever
3: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 ...
    link/ether 7c:dd:90:5a:2d:08 brd ff:ff:ff:ff:ff:ff
```

Was passiert mit Netzwerkpaketen, die nicht zur Verarbeitung durch den Raspberry Pi gedacht sind? Die Antwort auf diese Frage gibt `ip route`. Im Beispielnetzwerk werden alle IP-Pakete für den Adressbereich 10.0.0.*¹, also für das lokale Netzwerk, über die Schnittstelle `eth0` geleitet. Die default-Regel gibt außerdem an, dass IP-Pakete an alle anderen Adressen über das Gateway 10.0.0.138 geleitet werden sollen. Dabei kann es sich z. B. um einen ADSL-Router handeln.

```
ip route
default via 10.0.0.138 dev eth0
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.8
```

Die letzte Kerninformation einer Netzwerkkonfiguration unter Linux verrät die Datei `/etc/resolv.conf`. In dieser Datei ist eingetragen, welche Nameserver Ihr Raspberry Pi verwendet, um zu Hostnamen wie `yahoo.com` die entsprechende IP-Adresse zu ermitteln. Im folgenden Beispiel sind zwei Nameserver konfiguriert: einerseits 10.0.0.138 (der ADSL-Router) und andererseits 8.8.8.8 (der öffentliche Nameserver von Google). Die Schlüsselwörter `domain` und `search` geben an, dass das lokale Netzwerk den Domainnamen `lan` verwendet. Wenn Sie versuchen, eine Netzwerkverbindung zum Rechner `name` herzustellen, dann verwendet der Raspberry Pi automatisch `name.lan`.

```
cat /etc/resolv.conf
domain lan
search lan
nameserver 10.0.0.138
nameserver 8.8.8.8
```

Mit `ping` können Sie überprüfen, ob andere Rechner im lokalen Netzwerk bzw. im Internet erreichbar sind. `ping` sendet einmal pro Sekunde ein kleines Datenpaket an den angegebenen Rechner. Dieser antwortet in der Regel mit einem ebenso kleinen Paket. `ping` läuft normalerweise endlos, bis es mit `[Strg]+[C]` beendet wird. Alternativ können Sie an `ping` die Option `-c 3` übergeben – dann sendet `ping` nur drei Testpakete und endet dann.

```
ping -c 3 heise.de
PING heise.de (193.99.144.80) 56(84) bytes of data.
64 bytes from ... (...): icmp_req=1 ttl=247 time=43.6 ms
64 bytes from ... (...): icmp_req=2 ttl=247 time=41.4 ms
64 bytes from ... (...): icmp_req=3 ttl=247 time=42.1 ms

--- heise.de ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 41.498/42.423/43.665/0.942 ms
```

Dateien per FTP und HTTP übertragen

Der klassische Weg, um Dateien herunterzuladen, die auf einem FTP-Server zur Verfügung stehen, führt über das interaktive Kommando `ftp`. Beim Verbindungsauflauf müssen Sie sich anmelden – entweder mit einem Login-Namen und dem dazugehörigen Passwort oder mit dem Namen `anonymous` und Ihrer E-Mail-Adresse anstelle des Passworts. Die zweite Variante ist nur möglich, wenn der FTP-Server entsprechend konfiguriert ist. Die folgenden Zeilen geben ein kurzes Beispiel für seine Anwendung:

```
ftp ftp.servername.com
Name (ftp.servername.com:pi): anonymous
Password: name@mysite.de
ftp> cd verzeichnis
ftp> ls
...
ftp> get wichtige-datei.tar.gz
...
ftp> quit
```

Anstelle von `ftp` können Sie auch das Kommando `lftp` einsetzen, wobei Sie zuvor das gleichnamige Paket mit `apt-get install` installieren müssen. `lftp` stellt mehr Befehle als `ftp` zur Verfügung. Außerdem können Sie an `lftp` entweder mit `-c` mehrere durch Strichpunkte getrennte FTP-Kommandos übergeben oder mit `-f` eine Datei angeben, die diese Kommandos zeilenweise enthält.

Der interaktive Ansatz der Kommandos `ftp` und `lftp` ist zur Automatisierung von Downloads in einem Script ungeeignet. Abhilfe schafft das Kommando `wget`, das speziell zur Durchführung großer Downloads bzw. zur Übertragung ganzer Verzeichnisse konzipiert ist. `wget` unterstützt außer FTP auch die Protokolle HTTP und HTTPS. In der Grundform lädt `wget` die angegebene Datei einfach herunter, bei FTP-Servern mit einem `anonymous`-Login.

```
wget ftp://einftpserver.de/verzeichnis/name.tar.gz
wget http://einandererserver.de/verzeichnis/name.zip
```

Eine Alternative zu wget ist curl. Das Kommando hilft dabei, Dateien von oder zu FTP-, HTTP- oder sonstigen Servern zu übertragen. Die man-Seite listet eine beeindruckende Palette von Protokollen auf, die curl beherrscht. Hier beschränken wir uns allerdings auf FTP-Uploads. Für die Script-Programmierung besonders praktisch ist, dass curl auch Daten aus der Standardeingabe verarbeiten bzw. zur Standardausgabe schreiben kann. Sie müssen also nicht zuerst eine *.tar.gz-Datei erstellen und diese dann zum FTP-Server übertragen, sondern können beide Operationen mittels einer Pipe gleichzeitig ausführen. Das folgende Kommando überträgt die angegebene Datei zum FTP-Server backupserver und speichert sie im Verzeichnis verz:

```
curl -T datei -u username:password ftp://backupserver/verz
```

Um Daten aus dem Standardeingabekanal zu verarbeiten, geben Sie mit -T als Dateinamen einen Bindestrich an. Das folgende Kommando speichert das aus dem tar-Kommando resultierende Ergebnis direkt in der Datei name.tgz auf dem FTP-Server. Beachten Sie, dass auch bei tar als Dateiname ein Bindestrich angegeben ist, damit tar die Daten direkt in die Standardausgabe schreibt.

```
tar czf - verz/ | curl -T - -u usern:pw ftp://bserver/name.tgz
```

Kapitel 24

Der Raspberry Pi im Vogelhaus

Der Winter war schon fast vorbei, als die Idee aufkam, ein Vogelhäuschen im Garten aufzuhängen. Ein eher kleines sollte es sein, geeignet für Meisen. Ob es vielleicht möglich wäre, mithilfe eines Raspberry Pi und einer Kamera die Vögel beim Brüten zu beobachten, ohne sie zu stören? »Versuch macht kluch!« Dabei ist das Vogelhaus natürlich nur der Aufhänger, um Ihnen die Möglichkeiten des Raspberry-Pi-Kameramoduls näherzubringen: In diesem Kapitel lernen Sie, wie Sie Einzelbilder, Videos und Zeitrafferaufnahmen anfertigen, und realisieren zum Schluss eine rein software-gesteuerte Bewegungserkennung. Diese können Sie nicht nur zur Beobachtung von Tieren, sondern auch als Alarmanlage nutzen.

24.1 Einbau des Raspberry Pi samt Kameramodul in ein Vogelhaus

Als Grundlage für unser Projekt diente ein Vogelhaus aus dem Gartenmarkt (siehe Abbildung 24.1); handwerklich geschickte Menschen greifen sicher lieber selbst zur Säge. Das Vogelhaus, auf das die Wahl fiel, ist ein Mehrfamilienhaus: Es hat drei separate Brutkammern, jede mit einem eigenen Eingang.

Von den drei Kammern soll nur die mittlere für brütende Meisen (auf deren Größe sind die Einfluglöcher bemessen) zur Verfügung stehen. Die beiden äußeren Kammern sollen die Technik aufnehmen (siehe Abbildung 24.2). Zuerst wird der Raspberry Pi installiert und die Kamera ausgerichtet. Die Befestigung erfolgte mit kurzen Stahlnägeln, die durch die unteren Belüftungslöcher des Kunststoffgehäuses geführt wurden.

Vom Raspberry-Pi-Kameramodul gibt es zwei Varianten: eine »normale« und das sogenannte »NoIR«-Modul. Wir wählten hier die »NoIR«-Variante, weil sie auch bei dämmrigem Licht noch ansehnliche Bilder liefert. Der Name (*NoIR* = *No Infrared*, außerdem ist *noir* das französische Wort für *schwarz*) deutet darauf hin, dass diesem Kamera-Modul der sonst übliche Filter für infrarotes Licht fehlt. Das führt am Tag zu verfälschten Farben. Bei Dunkelheit oder bei Beleuchtung mit einem Infrarotscheinwerfer liefert die NoIR-Variante aber auch dann noch Bilder, wenn das normale Kameramodul schon lange aufgegeben hat.



Abbildung 24.1 Das Vogelhaus, frisch aus dem Gartenmarkt

Das Modul löst fünf Megapixel auf, liefert also Bilder mit 2592×1944 Pixeln. Kleinere Formate lassen sich per Software einstellen. Im Videobetrieb liefert das Modul maximal 1080p (Full HD) bei 30 Bildern pro Sekunde. Es misst $25 \times 20 \times 9$ Millimeter und wiegt 3 Gramm.

Im Vogelhaus reicht das durch die Einflugöffnung scheinende Mondlicht aus, um der Kamera brauchbare Aufnahmen zu ermöglichen (siehe Abbildung 24.3).

Flachbandkabel zu kurz?

Wenn das Flachbandkabel der Kamera zu kurz ist, können Sie inzwischen im Fachhandel eine längere Ausführung erwerben. Auch verschiedene Gehäuse für das Kameramodul (das in der Regel ohne Gehäuse verkauft wird) sind erhältlich.

Greifen Sie beim Stromanschluss unbedingt auf eine Steckdose zurück, die für den Betrieb im Außenbereich ausgelegt ist. In unser Vogelhaus hat es zwar nie hineingeregnet, aber Kondensfeuchte kann natürlich trotzdem entstehen. Achten Sie auch auf eine Zugentlastung der Zuleitung. Die Stromversorgung für den Raspberry Pi ist durch die Brutkammer geführt (siehe Abbildung 24.2). Bei Meisen funktioniert das, denn sie zerknabbern keine Kabel. Wenn Sie ein Vogelhaus wählen, dass auch größeren Vögeln Platz bietet, sollten Sie das Kabel entweder anders führen oder es durch



Abbildung 24.2 Eine Kammer für die Stromversorgung, eine für den Raspberry Pi, eine für die Mieter

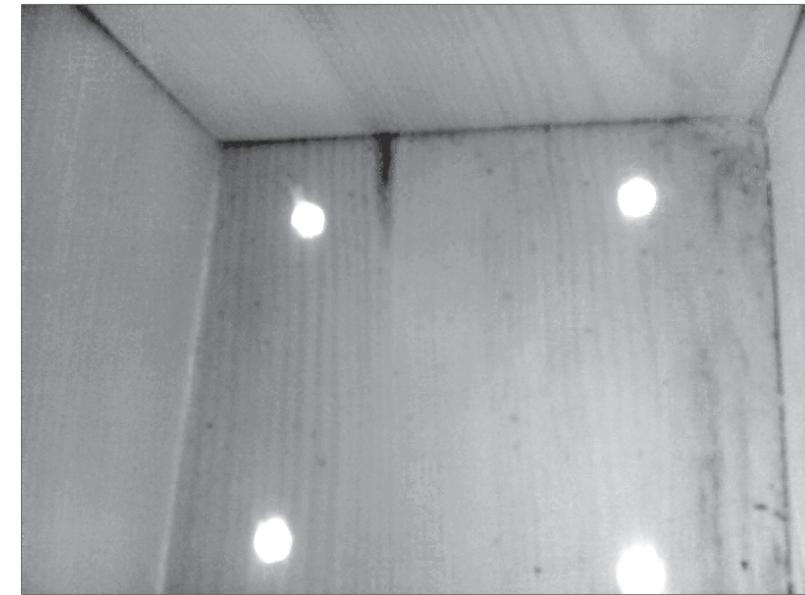


Abbildung 24.3 Die bezugsfertige Kammer, von innen fotografiert durch den Raspberry Pi bei Mondlicht ohne künstliche Beleuchtung

ein stabiles Rohr schützen. Auch für das Kameragehäuse wählen Sie in diesem Fall eine möglichst robuste Ausführung.

Zum Schluss schadet es nicht, das aufklappbare Dach der Vogelhauses zusätzlich mit etwas Dichtband abzudichten. Die Öffnung, aus der das Stromkabel austritt, kleben Sie mit witterfestem Band, etwa Gewebeband, ab.

24.2 Kamerapraxis

Wir haben das Kameramodul für den Raspberry Pi ja bereits in Abschnitt 13.7 näher vorgestellt. Dieser Abschnitt fasst nochmals zusammen, wie Sie das Modul in Betrieb nehmen und wie Sie Bild- und Filmaufnahmen erstellen. Außerdem erfahren Sie in diesem Abschnitt, wie Sie aus Einzelbildern einen Zeitrafferfilm machen, und lernen Möglichkeiten zur Bildoptimierung kennen.

Das Kameramodul betriebsbereit machen

Nachdem Sie das Kameramodul angeschlossen haben, rufen Sie einmal sudo raspi-config auf (siehe Abbildung 24.4). Wählen Sie den Menüpunkt Enable Camera und danach Finish. Sie werden nun aufgefordert, den Raspberry Pi einmal neu zu starten. Danach ist das Kameramodul betriebsbereit.

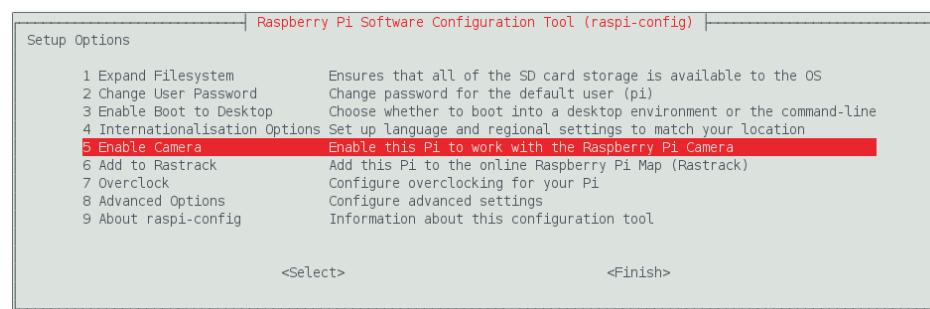


Abbildung 24.4 Aktivierung des Raspberry-Pi-Kameramoduls in raspi-config

Kamera-LED deaktivieren

Immer wenn die Kamera aktiv ist, also ein Bild oder Video aufzeichnet, leuchtet eine rote Leuchtdiode (LED) an der Vorderseite des Moduls. Das ist im Vogelhaus nicht erwünscht, denn auf so kleinem Raum ist die LED sehr hell und würde die Vögel mit Sicherheit verschrecken.

Sie können die Aktivierung der LED zum Glück unterdrücken. Editieren Sie dazu die Datei /boot/config.txt, und hängen Sie die folgende Zeile an das Ende der Datei an:

```
disable_camera_led=1
```

Auch diese Einstellung wird erst nach einem Neustart wirksam.

Standbilder mit raspistill aufnehmen

Bereits vorinstalliert sind die Programme raspistill und raspivid. Das eine ist für Bilder zuständig (engl. *still* = Standbild), das andere für Videos. Mit raspistill können Sie nicht nur einfach Bilder anfertigen, sondern diese auch in vielfältiger Weise manipulieren und nachbearbeiten. Es gibt Korrekturmöglichkeiten für die meisten gängigen Bildfehler, und Sie können auf Belichtung, Schärfe, Sättigung und vieles Weitere Einfluss nehmen. Sogar Reihen- und Zeitrafferaufnahmen sind möglich.

Im einfachsten Fall entlocken Sie Ihrer Kamera mit diesem Kommando ein Standbild:

```
raspistill -o bild.jpg
```

Das Ergebnis bild.jpg wird im aktuellen Verzeichnis abgelegt. Es wird in voller Auflösung erzeugt (2595 breit, 1944 Pixel hoch) und im JPEG-Format gespeichert. Obwohl das JPEG-Format eine Kompression beinhaltet, ist die Qualitätsstufe standardmäßig so gewählt, dass in der Regel keine Bildstörungen sichtbar sind. Falls Sie direkt auf dem Raspberry Pi arbeiten, also Monitor und Tastatur angeschlossen haben, ist Ihnen sicher aufgefallen, dass das aufgenommene Bild auf dem Monitor eingeblendet wird. Das ist die *Preview*-Funktion (*Preview* = Vorschau). Sie können diese Funktion mit dem Parameter -n abschalten:

```
raspistill -o bild.jpg -n
```

Auch die anderen Bildeigenschaften können Sie durch weitere Parameter beeinflussen. Wollen Sie etwa ein kleineres Bild haben, können Sie Höhe und Breite angeben:

```
raspistill -o bild.jpg -w 640 -h 480
```

Sind Ihnen die Bilddateien zu groß, können Sie die Qualität herunterschrauben und erhalten so kleinere Dateien. Hier wird die JPEG-Qualität auf 60 Prozent reduziert:

```
raspistill -o bild.jpg -q 60
```

Möchten Sie Ihre Bilder nicht im JPEG-Format bekommen, so stehen Ihnen auch noch die Ausgabeformate GIF, PNG und BMP zur Verfügung. Um die Ausgabe im PNG-Format zu wählen, geben Sie Folgendes ein:

```
raspistill -o bild.png -e png
```

Möglichkeiten zur Bildkorrektur

Wenn die Standardwerte des Kameramoduls keine zufriedenstellenden Ergebnisse liefern, gibt es einige Stellschrauben, an denen Sie drehen können. Ist Ihr Bild chromatisch unterbelichtet, ohne dass Sie das Problem durch eine Änderung der Beleuchtungssituation beheben können, kommt Ihnen vielleicht der Parameter `-ev` (Exposure Value, Belichtungskorrektur) zur Hilfe. Das Bild wird, wenn Sie hier einen positiven Wert angeben, künstlich aufgehellt. Dieser und weitere Parameter zu Bildkorrektur werden in Abschnitt 13.7 erläutert.

ISO-Einstellung

Wenn heute ein Kindergartenkind gebeten wird, Filmdöschen zum Basteln mitzubringen, guckt es in der Regel ein wenig verwirrt aus der Wäsche. Wir Älteren wissen aber noch, dass es Filmrollen in verschiedenen Empfindlichkeitsstufen gibt, die an der ISO-Zahl erkennbar sind und von Anhängern der analogen Fotografie noch heute gern benutzt werden. Je höher der ISO-Wert ist, umso empfindlicher ist der Film, also umso weniger Licht muss auf ihn fallen, um ein ansehnliches Bild zu produzieren.

Die Sensoren heutiger Kameras können ebenfalls in ihrer Empfindlichkeit eingestellt werden. Höhere ISO-Werte erhöhen auch hier die Empfindlichkeit und ermöglichen Aufnahmen bei schlechten Lichtverhältnissen. Diesen Vorteil erkauft man sich in der Regel mit zunehmendem Bildrauschen.

Auch den ISO-Wert der Raspberry-Pi-Kamera können Sie einstellen. Standardmäßig kommt eine Automatik zum Einsatz, die selbstständig versucht, die richtige ISO-Einstellung zu wählen. Bei schwierigen Lichtverhältnissen kann die Automatik aber versagen und über- oder unterbelichtete Bilder liefern. In diesem Fall stellen Sie den Wert manuell ein:

```
raspistill -o bild.jpg -ISO 800
```

Im Vogelhaus wurde die Kamera auf die höchste Empfindlichkeitsstufe eingestellt. Die Skala reicht bei der Raspberry-Kamera von 100 bis 800 in Hunderterschritten. Den ISO-Parameter können Sie auch für Videoaufnahmen mit raspivid nutzen.

Zeitverzögerung und Zeitrafferfilme

Sie können die Kamera eine Zeitlang warten lassen, bevor das Bild erzeugt wird. Die Länge der Pause geben Sie in Millisekunden an, für fünf Sekunden Verzögerung also 5000:

```
raspistill -o bild.jpg -t 5000
```

Eine Zeitrafferaufnahme erstellen Sie, indem Sie den zusätzlichen Parameter `-tl` (`tl = timelapse`, Zeitraffer) hinzunehmen. Das folgende Kommando erstellt alle fünf Sekunden ein Bild, insgesamt sechzig Sekunden lang. Das `\%03d` im Dateinamen führt dazu, dass raspistill die Bilder mit einer fortlaufenden dreistelligen Nummer versieht, also `bild-001.jpg`, `bild-002.jpg` und so weiter.

```
raspistill -o bild-\%03d.jpg -t 60000 -tl 5000
```

Jetzt haben Sie eine Reihe von Einzelaufnahmen, die Sie zu einem Zeitraffervideo zusammenbacken können. Das gelingt mit avconv (audio-video-converter). Sollte avconv auf Ihrem Raspberry Pi noch nicht installiert sein, können Sie das schnell mit `sudo apt-get -fym install libav-tools` nachholen. Das folgende Kommando erstellt aus Ihren Einzelbildern ein Video im MP4-Format mit fünf Bildern pro Sekunde:

```
avconv -r5 -f image2 -i bild-\%03d.jpg zeitraffer.mp4
```

Dieser Vorgang ist sehr rechenintensiv und dauert auf dem Raspberry Pi eine ganze Weile. Falls Sie noch einen weiteren, schnelleren Linux-Rechner zur Verfügung haben, ist es eine gute Idee, die Einzelbilder auf diesen zu kopieren und das Zeitraffervideo dort erstellen zu lassen.

Videos aufzeichnen mit raspivid

Das Aufzeichnen von Videos mit raspivid ist genau so einfach wie das Anfertigen von Standbildern, und viele Parameter sind ebenfalls gleich oder ähnlich. Das folgende Kommando nimmt ein Video von 10 Sekunden Länge auf (auch hier wieder in Millisekunden angegeben). Die Größe ist dabei auf 640 mal 480 Pixel reduziert.

```
raspivid -o video.h264 -w 640 -h 480 -t 10000
```

Das Videoformat H.264, das standardmäßig verwendet wird, können die meisten Abspieler problemlos verarbeiten. Sollten Sie doch einmal Probleme haben, können Sie das Video mit avconv, das Sie bei den Zeitrafferaufnahmen schon kennengelernt haben, konvertieren. Das folgende Kommando rechnet Ihr Video in das MP4-Format um:

```
avconv -i video.h264 -vcodec copy video.mp4
```

Hier gilt wie beim Zeitraffer: Es dauert auf dem Raspberry Pi recht lange. Sie können tricksen, indem Sie die Bildwiederholrate reduzieren, etwa auf 15 Bilder pro Sekunde:

```
avconv -i video.h264 -r 15 -vcodec copy video.mp4
```

Das geht natürlich zulasten der Bildqualität. Bei 15 Bildern pro Sekunde nimmt das menschliche Auge schon ein störendes Ruckeln wahr. Besser ist es, Sie nehmen die Konvertierung auf einem anderen, schnelleren Rechner vor.

24.3 Bewegungserkennung mit motion

Zeitrafferaufnahmen und Videos sind gut und schön, aber wenn sich vor der Linse nichts tut, sind sie genauso langweilig wie ein Standbild. Daher wäre es sinnvoll, eine Bewegungserkennung zu haben, die die Kamera nur dann zu einer Aufnahme veranlasst, wenn tatsächlich etwas passiert. Das ist nicht nur für unser Vogelhaus sinnvoll, sondern eignet sich auch gut als Alarmanlage während des Urlaubs oder zur Beobachtung schreckhafter Tiere.

Das Paket `motion` ermöglicht es Ihnen, diese Idee mit dem Raspberry-Kameramodul und natürlich auch mit anderen Webcams umzusetzen. Mit den folgenden Kommandos installieren Sie `motion` und das Paket `v4l-utils`. Dieses Paket enthält einen `v4l`-Treiber (`v4l = video for linux`), der das Raspberry-Kameramodul unter der Bezeichnung `/dev/video0` für `motion` sicht- und nutzbar macht.

```
apt-get update
apt-get upgrade
apt-get -fym install v4l-utils motion
```

Das folgende Kommando lädt das Treibermodul. Nachdem Sie es ausgeführt haben, existiert die Datei `/dev/video0`.

```
modprobe bcm2835-v4l2
```

Dass das Modul korrekt geladen und die Kamera erkannt wurde, sehen Sie auch im System-Logfile. Schauen Sie sich die letzten Zeilen mit `tail -n 20 /var/log/syslog` einmal an:

```
[ 864.023270] Linux video capture interface: v2.00
[ 864.068272] bcm2835-v4l2: scene mode selected 0, was 0
[ 864.074260] bcm2835-v4l2: V4L2 device registered as
                  video0 - stills mode > 1280x720
[ 864.079525] bcm2835-v4l2: Broadcom 2835 MMAL video capture
                  ver 0.0.2 loaded.
```

Das Treibermodul wird noch weiterentwickelt, deshalb können bei Ihnen andere Versionsnummern auftauchen. Wichtig ist `registered as video0`, denn das bedeutet, dass Ihre Kamera startklar ist.

Motion konfigurieren

Das Paket `motion` bringt bei der Installation eine Konfigurationsdatei mit, die `/etc/motion/motion.conf` heißt. Lassen Sie sich nicht von der Größe der Datei abschrecken! Man kann sehr viel einstellen, aber fast alle Werte haben sinnvolle Voreinstellungen und müssen nicht geändert werden. Um mit `motion` loslegen zu können, reichen ganz wenige Modifikationen, die wir nun Schritt für Schritt erläutern. Trotzdem ist

es immer eine gute Idee, die unveränderte Konfigurationsdatei unter einem anderen Namen zu sichern, etwa so:

```
sudo cp /etc/motion/motion.conf /etc/motion.motion.conf.sicher
```

Jetzt kann es losgehen. Öffnen Sie die `motion.conf`, und suchen Sie diese Zeilen:

```
# Datei /etc/motion/motion.conf
# Image width (pixels).
# Valid range: Camera dependent, default: 352
width 320

# Image height (pixels).
# Valid range: Camera dependent, default: 288
height 240
```

Hier können Sie die Bildgröße einstellen, die `motion` aufzeichnen wird. 320×240 Pixel ist arg klein, diese Werte können Sie getrost verdoppeln.

Danach stellen Sie die Empfindlichkeit ein, mit der `Motion` auf Änderungen im Bild reagiert. Die Bewegungserkennung funktioniert so, dass `motion` nacheinander aufgenommene Bilder miteinander vergleicht und prüft, wie viele Bildpunkte sich von einem zum anderen Bild geändert haben. Ist eine gewisse Schwelle überschritten, startet `motion` die Aufnahme und stoppt sie wieder, wenn das Bild sich beruhigt. Diese Schwelle stellen Sie an folgender Stelle der Konfigurationsdatei ein:

```
# Datei /etc/motion/motion.conf
# Threshold for number of changed pixels in an image that
# triggers motion detection (default: 1500)
threshold 1500
```

Standardmäßig müssen sich also 1500 Pixel zwischen zwei Bildern ändern, damit `motion` dies als Bewegung interpretiert und reagiert. Für Ihre ersten Experimente können Sie diesen Wert niedrig ansetzen. Später finden Sie den richtigen Wert eigentlich nur durch ein wenig Experimentieren heraus, denn er hängt natürlich auch ganz wesentlich davon ab, was Sie beobachten oder überwachen möchten. Eine Beobachtungskamera in einem Vogelhaus benötigt hier natürlich andere Werte, als wenn Sie `motion` nachts als Einbruchsüberwachung in einer Lagerhalle einsetzen.

Nun kommen wir zum Ausgabeformat. Per Default speichert `motion` alle Videos im Shockwave-Flash-Format mit der Dateiendung `.swf`:

```
# Datei /etc/motion/motion.conf
ffmpeg_video_codec swf
```

Dafür benötigen Sie jedoch eine proprietäre Abspielsoftware, was unschön ist. Um stattdessen Videos im MP4-Format zu erhalten, ändern Sie die Zeile so ab:

```
# Datei /etc/motion/motion.conf
ffmpeg_video_codec mpeg4
```

Nicht notwendig, aber eine nette Spielerei ist die `locate`-Funktion. Wenn `motion` irgendwo im Bild eine Bewegung erkannt hat, kann es diesen Bildbereich mit einem rechteckigen Rahmen kennzeichnen. Diese Funktion ist zunächst deaktiviert (`off`). Ändern Sie sie zu `locate on`, wenn Sie diese Funktion nutzen möchten.

```
# Datei /etc/motion/motion.conf
locate off
```

Auf Port 8081 stellt `motion` einen Mini-Webserver zur Verfügung, auf dem Sie das aktuelle Bild der Kamera live verfolgen können. Es gibt aber zwei Haken: Erstens ist so `motion` zunächst so konfiguriert, dass sich das Live-Bild nur bei einer erkannten Bewegung aktualisiert. Zweitens ist dieser Webserver nicht von anderen Rechnern im gleichen Netz erreichbar, denn er ist nur an das lokale Loopback-Interface gebunden. Glücklicherweise lässt sich beides leicht ändern. Finden Sie die folgenden Zeilen:

```
# Datei /etc/motion/motion.conf
# rate given by webcam_maxrate when motion is detected
# (default: off)
webcam_motion off
...
# Restrict webcam connections to localhost only
# (default: on)
webcam_localhost on
```

Ändern Sie nun `off` bzw. `on` in das jeweilige Gegenteil:

```
# Datei /etc/motion/motion.conf
webcam_motion on
...
webcam_localhost off
```

Jetzt stellt der Livestream ein Bild pro Sekunde dar, auch wenn nichts passiert (ein Video speichert `motion` natürlich trotzdem nur dann, wenn eine Bewegung erkannt wird). Außerdem ist der Livestream jetzt auch von den Netznachbarn Ihres Raspberry Pi zu bewundern. Das können Sie auch gleich einmal ausprobieren, denn die grundlegende Konfiguration ist damit abgeschlossen.

Starten Sie `motion`, indem Sie einfach `motion` auf der Kommandozeile eingeben. Es sucht automatisch nach der Konfigurationsdatei `/etc/motion/motion.conf` und liest sie ein. Jetzt können Sie auf einem Browser die Adresse Ihres Raspberry Pi, gefolgt von `:8081` eingeben. Hat der Raspi also zum Beispiel die IP-Adresse 192.168.2.10, so geben Sie in die Adresszeile des Browsers `192.168.2.10:8081` ein. Jetzt sehen Sie ein Bild pro Sekunde live aus der Raspberry-Pi-Kamera. Wenn eine Bewegung im Bild erkannt

wird, umrahmt `motion` den Bereich (siehe Abbildung 24.5) und startet gleichzeitig die Aufnahme.



Abbildung 24.5 `motion` erkennt eine Bewegung.

Der fehlende Infrarotfilter des NoIR-Kameramoduls verfälscht die Farben, wenn man es bei Tageslicht einsetzt. Die Aufnahmen speichert `motion` im Verzeichnis `/tmp/motion`. Es ist sinnvoll, alte Dateien regelmäßig aus diesem Verzeichnis zu löschen, denn je nach Aktivität kann es dort bald recht eng zugehen. Es empfiehlt sich, der in Linux eingebauten Zeitsteuerung `Cron` diese Aufgabe zu überlassen. `Cron` hat eine Art To-do-Liste, die `crontab`. Sie editieren sie mit dem Kommando `sudo crontab -e`. Fügen Sie diese Zeile hinzu:

```
0 0 * * * find /tmp/motion/ -iname "*" -mtime +7 -delete
```

Verlassen Sie nun den Editor. Jetzt werden täglich um Mitternacht alle Dateien aus `/tmp/motion` gelöscht, die älter als 7 Tage sind.

24.4 Das Vogelhaus im praktischen Einsatz

Nach so vielen Tipps zur optimalen Verwendung der Kamera sollen Sie zum Abschluss noch erfahren, welche der bisher dargestellten Möglichkeiten wir tatsächlich im Vogelhaus genutzt haben: Die Bewegungserkennung mit `motion`, die ein durchaus breites Einsatzspektrum hat, haben wir nicht genutzt. Der Grund: Wären tatsächlich Meisen in das Haus eingezogen, so wäre dort permanent Bewegung gewesen, und `motion` hätte praktisch unablässig gefilmt (jedenfalls, solange das Licht ausreicht).

Stattdessen wurde mit `raspistill` alle 60 Sekunden ein Einzelbild in der Auflösung 1024×768 Pixel geschossen. Es bietet sich an, das von `cron` erledigen zu lassen (siehe Abschnitt 4.12). Der folgende `crontab`-Eintrag ist hier nur aus Platzgründen über zwei Zeilen verteilt. Geben Sie das gesamte Kommando ohne \ in einer Zeile an!

```
* * * * * raspistill -o /var/www/html/birdpi.jpg -w 1024 -h 768 \
-ex night -ifx denoise -sh 50
```

Der Parameter `-ex night` schaltet die Kamera dabei in eine Art Nachtmodus. Dieser bewirkt hauptsächlich, dass die Kamera hohe ISO-Werte nutzt, die sonst nicht zum Einsatz kämen. Mit `-ifx denoise` wird eine Nachbearbeitung vorgenommen, die das Bildrauschen reduzieren soll, das durch die hohen ISO-Werte entsteht. Dadurch wird das Bild aber recht stark »gebügelt«, und es besteht die Gefahr, dass Details verloren gehen. Deshalb wird zum Schluss noch mit `-sh 50` ein wenig nachgeschärft.

Abgelegt wird das Bild unter `/var/www/html`. Das ist das Standardverzeichnis des Webservers Apache, der ebenfalls auf dem Raspberry Pi installiert ist. Im gleichen Verzeichnis liegt eine sehr einfach gestrickte HTML-Datei, die nichts weiter macht, als dieses Bild anzuzeigen:

```
<html>
  <head>
    <title>BirdPi</title>
  </head>
  <body>
    
  </body>
</html>
```

Durch Eingabe der IP-Adresse des Vogelhaus-Raspberry-Pi wird nun das Bild aus der Brutkammer angezeigt.

Lichtverhältnisse und Bildqualität

Wir waren uns nicht sicher, wie Meisen auf zusätzliches Infrarotlicht in der Brutkammer reagieren, und haben es daher nicht eingesetzt. Das hat zur Folge, dass für einige Stunden in der Mitte der Nacht ein rein schwarzes Bild entsteht. Allerdings

ist die NoIR-Variante der Raspberry-Kamera ausreichend lichtstark, um auch bei sehr geringem Umgebungslicht, etwa bei Mondschein oder in der Morgen- und Abenddämmerung, schon erkennbare Bilder zu liefern. Falls doch irgendwann ein Hilfslicht zum Einsatz kommen wird, werden wir zu einer einzelnen Infrarot-Diode greifen – alles andere wäre für den Einsatz auf so kleinem Raum völlig übertrieben.

Die Bildqualität haben wir getestet, indem wir kleine Gegenstände in das Vogelhaus gelegt haben, etwa ein Spielzeugauto oder eine Tomate. Die entstandenen Bilder waren brauchbar, wenn auch nicht hundertprozentig scharf. Das liegt daran, dass man bei dieser Entfernung an der Naheinstellgrenze der Kamera kratzt. Sie kennen das von Ihren eigenen Augen – was Sie sich direkt vor die Pupille halten, können Ihre Augen nicht scharf abbilden, ein gewisser Mindestabstand muss sein. Trotzdem war das Bild hinreichend gut, um von weiteren Modifikationen abzusehen.

Wo ist nun die brütende Meise?

Gern hätten wir Ihnen an dieser Stelle noch ein Foto von brütenden Meisen gezeigt, aber unser Vogelhaus wurde leider nicht bezogen. Ein Freund mit ornithologischem Sachverständig führte es darauf zurück, dass der Zeitpunkt (Ende Februar / Anfang März) schon zu spät gewählt war. Im nächsten Jahr versuchen wir unser Glück aufs Neue. Im Blog zu diesem Buch unter der Adresse <http://pi-buch.info> werden wir Sie darüber auf dem Laufenden halten!

Auf einen Blick

TEIL I

Den Raspberry Pi kennenlernen 39

TEIL II

Der Raspberry Pi als Media-Center 283

TEIL III

Hardware-Grundlagen 355

TEIL IV

Programmierung 613

TEIL V

Projekte 849

Inhaltsverzeichnis

Vorwort	35
---------------	----

TEIL I Den Raspberry Pi kennenlernen

1 Kauf und Inbetriebnahme	41
1.1 Kauf	42
Raspberry-Pi-Modelle	42
Von Äpfeln und Birnen, Himbeeren und Bananen	45
Anschlüsse	46
Netzteil	48
SD-Karte	49
Gehäuse	50
Tastatur und Maus	50
USB-Hub	51
Stromversorgung per USB-Hub	51
WLAN- und Bluetooth-Adapter	53
Was Sie sonst noch brauchen	53
1.2 NOOBS-Installation	54
Raspberry-Pi-Distributionen	54
SD-Karte formatieren	56
USB-SD-Card-Reader	57
NOOBS	58
NOOBS-VNC-Installation	61
1.3 Image-Datei auf eine SD-Karte schreiben	62
Image-Dateien herunterladen	63
Image-Datei unter Windows auf eine SD-Karte übertragen	63
Image-Datei unter OS X auf eine SD-Karte übertragen	64
Image-Datei unter Linux auf eine SD-Karte übertragen	66
Distributionsspezifische Installationsprogramme	67

Inhaltsverzeichnis

1.4	Installation auf einen USB-Stick (für Fortgeschrittene)	68
Vor- und Nachteile	68	
USB-Stick vorbereiten	69	
SD-Karte vorbereiten	69	
Systempartition vergrößern	70	
2	Erste Schritte in Raspbian	75
2.1	Das Konfigurationsprogramm raspi-config	75
Expand Filesystem	76	
Change User Password	76	
Enable Boot to Desktop/Scratch	76	
Internationalisation Options	77	
Enable Camera	78	
Add to Rastrack	78	
Overclock	78	
Advanced Options	79	
Neustart	80	
2.2	Ein- und Ausschalten	80
Wenn der Raspberry Pi abstürzt	81	
2.3	Die Benutzeroberfläche LXDE	81
Icons	82	
Das Panel	83	
Programme starten	84	
Konfiguration	84	
Tastenkürzel	85	
Bildschirmschoner	87	
2.4	Wichtige Programme	88
Die Icon-Galerie des Raspbian-Desktops	88	
Der Dateimanager	90	
LXTerminal	91	
Leafpad und Gedit (Editoren)	93	
Firefox und Thunderbird, Chromium und Claws Mail	94	
Bild- und PDF-Viewer	96	
Audio- und Video-Player	97	
Textverarbeitung und Tabellenkalkulation	98	
Screenshots	99	

Inhaltsverzeichnis

2.5	WLAN-Konfiguration	100
2.6	Bluetooth-Konfiguration	103
2.7	USB-Sticks und -Festplatten	104
2.8	Drucker	105
3	Arbeiten im Terminal	107
3.1	Erste Experimente	107
Dateien und Verzeichnisse	108	
Zugriff auf Systemdateien	108	
Hardware-Tests	109	
vcgencmd	110	
Software-Verwaltung	110	
Freien Speicherplatz ermitteln	111	
Online-Hilfe	112	
3.2	Eingabeerleichterungen und Tastenkürzel	113
alias-Abkürzungen definieren	115	
Textkonsole	115	
3.3	Farbspiele	116
Den Prompt individuell einrichten	116	
ls-Farben	117	
3.4	Die Bourne Again Shell (bash)	118
Jokerzeichen	119	
Ein- und Ausgabeumleitung	120	
Kommandos im Hintergrund ausführen	121	
Mehrere Kommandos ausführen	121	
Rechnen in der bash	123	
Umgebungsverhalten	123	
3.5	Arbeiten mit Administratorrechten (sudo)	124
sudo-Anwendung	125	
sudo-Konfiguration	125	
3.6	Textdateien lesen und ändern	127
cat und less	127	
tail	128	
grep	128	

Texteditoren	129
nano	130
Vi	131
jmacs	132
3.7 Verzeichnisse und Dateien	133
Verzeichnisse	133
Linux-Verzeichnisbaum	134
Dateien auflisten	135
Verbogene Dateien und Verzeichnisse sichtbar machen	136
Dateien suchen	137
Textdateien inhaltlich durchsuchen	138
Dateien kopieren, verschieben und löschen	138
Links	139
3.8 Komprimierte Dateien und Archive	140
Komprimieren und dekomprimieren	141
TAR-Archive	142
ZIP-Archive	143
3.9 Prozessverwaltung	143
Liste aller laufenden Prozesse	144
Prozesshierarchie	146
Prozesse abbrechen	147
Prozesspriorität einstellen	147
Systemdienste steuern	148
3.10 Netzwerkkommandos	149
Netzwerkstatus ermitteln	149
Dateien per FTP und HTTP übertragen	151
4 Arbeitstechniken	153
4.1 Programme installieren und verwalten	154
Programme bzw. Pakete installieren	154
Programme bzw. Pakete entfernen	155
Informationen über Pakete ermitteln	155
4.2 Updates	157
Raspbian aktualisieren	157
Firmware- und Kernel-Updates	158

4.3 SSH	161
Voraussetzungen auf dem Raspberry Pi	161
Voraussetzungen auf Ihrem Arbeitsrechner	162
SSH anwenden	162
Grafikprogramme ausführen	164
SSH-Login mit Schlüsseln statt Passwörtern	165
SSH-Login nach einer Raspberry-Pi-Neuinstallation	165
Dateien kopieren mit scp	167
Zugriff auf das Raspberry-Pi-Dateisystem im Dateimanager	168
Die Kommandos ssh und scp auf dem Raspberry Pi nutzen	168
4.4 VNC	169
Erste Tests	169
VNC-Viewer unter Windows	170
VNC-Viewer unter OS X	171
VNC-Viewer unter Linux	172
Start des VNC-Servers automatisieren	172
VNC über einen SSH-Tunnel	175
4.5 Netzwerkverzeichnisse nutzen	176
Grundlagen	177
Beispiele	179
Verzeichnisse dauerhaft einbinden (/etc/fstab)	181
4.6 Netzwerkverzeichnisse anbieten (Samba-Server)	182
Samba-Installation, Defaultkonfiguration	183
Ein Verzeichnis mit Passwortschutz freigeben	185
Freigabe testen	186
Weitere Verzeichnisse freigeben	187
Ein Verzeichnis ohne Passwortschutz freigeben	188
Absicherung und Grundeinstellungen	189
4.7 Internetzugriff auf den Raspberry Pi	189
Wo ist das Problem?	190
Lösungsvarianten	192
Alternativen	193
4.8 Dynamisches DNS	194
Wozu dient und wie funktioniert dynamisches DNS?	194
Dynamisches DNS mit noip.com einrichten	195
Port Forwarding einrichten	196
Test	197
Fehlersuche	198

4.9	IPv6	199
	IPv6 aktivieren	200
	gogoc-Installation	200
	Test	202
4.10	FTP-Server einrichten	203
	vsftpd installieren	204
	Defaultkonfiguration	204
	Eigene Konfiguration	205
	FTP ausprobieren	205
4.11	Programme beim Systemstart ausführen	206
	rc.local	206
	Autostart-Programme für den Desktop	207
	Autostart-Dateien für Openbox	207
4.12	Programme regelmäßig ausführen (Cron)	208
	/etc/cron.xxx-Verzeichnisse	208
	/etc/crontab-Datei	208
	/etc/cron.d-Verzeichnis	210
	Benutzerspezifische Crontab-Dateien	210
	Anacron	210
4.13	Backups erstellen	211
	Backups mit rsync	212
	Backups mit tar	213
	Backup einer SD-Karte unter Windows	213
	Backup einer SD-Karte unter OS X	214
	Backup einer SD-Karte unter Linux	214
4.14	Overclocking	215
	Overclocking-Parameter in config.txt	215
	Aktuelle Taktfrequenz und Temperatur ermitteln	217
4.15	Notfall-Tipps	217
	Stromversorgung	218
	SD-Karte	218
	Display-Probleme	219
	Status-Leuchtdioden (Modell A und B)	220
	Status-Leuchtdioden (Modell B+)	221

5	Linux-Grundlagen	223
5.1	Benutzer und Gruppen	224
	Benutzer und Passwörter	225
	Gruppen	227
	Kommandos zur Benutzer- und Gruppenverwaltung	228
	sudo	230
	Auto-Login abstellen	231
5.2	Zugriffsrechte	232
	Zugriffsrechte auf Dateien	232
	Zugriffsrechte auf Verzeichnisse	233
	Setuid-, Setgid- und Sticky-Bit	234
	Oktalschreibweise	235
	Zugriffsrechte einstellen	236
	Praxisbeispiele	238
5.3	Paketverwaltung	238
	Konfiguration	239
	Hauptpaketquelle	239
	Weitere Paketquellen	240
	Aus welcher Paketquelle stammt ein Paket?	241
5.4	Verwaltung des Dateisystems	241
	Dateisysteme	242
	Device-Dateien	242
	Das mount-Kommando	244
	Die Datei /etc/fstab	246
	Datenträger partitionieren und formatieren	247
5.5	Netzwerkkonfiguration	249
	Glossar	250
	Die Datei /etc/network/interfaces	252
	Statische LAN-Konfiguration	253
	Manuelle WLAN-Konfiguration	254
	Nameserver-Konfiguration	256
	Hostname ändern	257
5.6	Bluetooth	257
	Bluetooth-Geräte erkennen	258
	Pairing	258
	Verbindungsaufbau erzwingen	259

5.7	Systemstart	259
	1st-Stage-Bootloader	261
	2nd-Stage-Bootloader (bootcode.bin)	261
	Kernelparameter (cmdline.txt)	261
	Kernelstart	263
	Init-Systeme	263
	Die Runlevel des Init-V-Systems	264
	Systeminitialisierung	265
	Runlevel 2 aktivieren	266
	Runlevel wechseln	266
	Reboot und Halt	266
	Systemdienste manuell starten und beenden	267
	Den Start von Systemdiensten aktivieren und deaktivieren	267
	Eigene Initialisierungsarbeiten durchführen	268
	Eigene Systemdienste einrichten	269
5.8	Systemeinstellungen in config.txt	270
	HDMI-Einstellungen	271
	Overclocking	272
	Video-Decodierung	272
	Bootvorgang	273
	Hardware-Parameter	273
	Speicheraufteilung zwischen CPU und GPU	274
	Das Kommando vcgencmd	274
5.9	Grafiksystem	275
	Start des Grafiksystems	276
	Auto-Start des Grafiksystems ein-/ausschalten	276
	Auto-Login ein-/ausschalten	277
	Manueller Start des Grafiksystems	277
	Neue Grafiktreiber und Wayland	277
5.10	Kernel und Module	278
	Kernelmodule	278
	Module selbst kompilieren	280
	Den Kernel selbst kompilieren	280

TEIL II Der Raspberry Pi als Media-Center

6	Audio-Player mit Smartphone-Fernbedienung	285
6.1	MPD-Installation und -Konfiguration	285
	Kernel- und Firmware-Update	286
	Netzwerkkonfiguration	286
	MPD-Installation und -Konfiguration	287
	Erste Tests mit Xfmpc	288
	Tonausgabe zwischen HDMI und Analog-Audio umschalten	289
	MPDroid	290
6.2	MPD-Konfigurationsvarianten	291
	Audio-Dateien auf einem USB-Stick	291
	Audio-Dateien auf einem NAS	292
	PulseAudio	295
	Perfekter Klang für Audiophile	296
	CD-Cover-Abbildungen	296
	Weboberfläche zur Steuerung	298
	Musik als Audio-Stream verteilen (Icecast)	300
6.3	Volumio	302
	Installation und Konfiguration	303
	Interna und Absicherung	304
	Probleme	304
6.4	Pi MusicBox	305
	Installation	305
	Konfiguration	306
	Bedienung	307
7	Multimedia-Center mit XBMC	309
7.1	XBian versus Raspbmc versus OpenELEC	310
	XBian	310
	Raspbmc	311
	OpenELEC	311
	Die Qual der Wahl	312

7.2 OpenELEC-Installation und -Konfiguration	313
Standardinstallation auf eine SD-Karte	313
Installation auf einen USB-Stick (für Fortgeschrittene)	313
OpenELEC-Erstkonfiguration	315
Tastatur und Bluetooth	316
Zeitzone einstellen	316
Netzwerk- und WLAN-Konfiguration	316
SSH nutzen	317
SSH absichern	318
Samba absichern	319
Überblick über das OpenELEC-Dateisystem	319
Hardware-Decodierung für MPEG-2 und VC-1 aktivieren	320
Overclocking	321
Audio-Ausgang wählen	322
RSS-Newsfeed abstellen	322
XBMC-Layout (Skins)	322
Screenshots	323
Updates	323
7.3 Fernbedienung	324
CEC-Fernbedienung	324
Smartphone-Fernbedienung	324
Infrarot-Fernbedienung	326
7.4 XBMC-Betrieb	329
Das erste Video abspielen	329
Video-Dateien eines USB-Sticks abspielen	331
Zugriff auf Video-Dateien aus Netzwerkverzeichnissen	332
Videos aus dem Internet	334
Audio-Dateien abspielen	335
Internet-Radios und Spotify	336
AirPlay	336
8 Multimedia-System mit Plex	339
8.1 XBMC oder Plex?	339
Plex-Vorteile	340
Plex-Nachteile	340
Glossar	341
8.2 Plex-Server-Konfiguration	342

8.3 RasPlex-Installation und -Konfiguration	345
Installation	345
Konfigurationsprogramme	347
Sprache, Zeitzone und Tastaturlayout	347
WLAN-Konfiguration	349
Plex-Login	349
SSH und Samba	349
Hardware-Decodierung für MPEG-2 und VC-1 aktivieren	349
Audio-Ausgang	350
Fernbedienung	350
Bluetooth-Geräte	351
Bildschirmschoner	351
Screenshots	351
8.4 Plex-Anwendung	352

TEIL III Hardware-Grundlagen

9 Hardware-Einstieg	357
9.1 Platinenaufbau	357
9.2 Der BCM2835	359
9.3 GPIO-Kontakte	360
Der P1-Header	360
Revision 1 versus 2	362
Nummerierungssysteme bzw. Pin-Namen	362
Das 50-mA-Limit	363
Welcher Pin für welchen Zweck?	363
Die P2-, P3-, P5- und P6-Header	364
GPIO-Verbindungen herstellen	366
Vorsichtsmaßnahmen	367
9.4 GPIO-Änderungen beim Modell B+	368
9.5 Stromversorgung	369
Das Steckernetzteil	369
Betrieb mit einer USB-Powerbank	370
Betrieb mit AA-Batterien	372
Der Linearregler 7805	373

Der Schaltregler LM2596S	374
Versorgung über den USB-Port	376
9.6 Gehäuse	377
The Punnet – Das kostenlose Papiergehäuse	377
Das Legogehäuse	378
ModMyPi-Standardcase	378
Rail-Case – Der Raspberry Pi auf der Schiene	378
Multi-Pi – Das stapelbare Raspberry-Pi-Gehäuse	379
Designergehäuse aus edlem Holz	379
Durchsichtiges Gehäuse mit Kamerahalterung	380
10 Elektrotechnik-Crashkurs	383
10.1 Strom, Spannung und das Ohmsche Gesetz	383
10.2 Grundschaltungen	385
Darstellung der Schaltpläne	385
Reihenschaltung	386
Parallelschaltung	387
Kombinierte Schaltung	387
Spannungsteiler	389
10.3 Elektronische Grundbauteile	390
Diode	390
Leuchtdiode	390
Kondensator	392
Spule	392
Transistor	393
Relais	394
10.4 Das Multimeter	395
10.5 Breadboard oder Platine?	398
Breadboards	398
Streifen- oder Lochrasterplatten	399
10.6 Löten	400
10.7 Breadboardadapter	402
Breadboardadapter selbst erstellen	402
Der T-Cobbler	406

11 LEDs, Motoren und Relais	407
11.1 Leuchtdioden (LEDs)	407
LED im Dauerbetrieb	407
LED per Python ein- und ausschalten	408
LEDs über Transistoren schalten	410
LEDs mit Darlington-Transistoren ansteuern	412
LEDs mit einem Taster ein- und ausschalten	413
LEDs mit Software-PWM dimmen	419
LEDs mit Hardware-PWM dimmen	421
Hardware PWM beim Modell B+	422
RGB-LEDs	422
11.2 Optokoppler	426
11.3 Elektromotoren	427
Der geeignete Motor	428
Gleichstrom- und Wechselstrommotoren	429
Die H-Brücke	430
Der Motortreiber L298	432
Verdrahtung und Software	434
Motordrehzahl per Software ändern	438
11.4 Schrittmotoren	440
Bipolare Ansteuerung	441
Unipolare Ansteuerung	445
Intelligente Schrittmotortreiber	447
11.5 Relais	451
Relaistypen	451
12 Bussysteme	455
12.1 SPI	455
SPI per raspi-config einrichten	456
SPI manuell einrichten	457
Das digitale Potenziometer	458
Daten per SPI senden	459
Daten per SPI lesen	463
Der Analog-/Digitalwandler MCP3008	465
Der Digital-/Analogwandler MCP4811	470
SPI-Bauteile	475

12.2 I²C	476
Der Portexpander MCP23017	477
12.3 UART	484
UART aktivieren und säubern	484
Zwei Raspberry Pi via UART verbinden	485
Minicom	485
UART in Python	487
Wichtige pySerial-Funktionen	489
12.4 Der Audiobus I²S	489
I ² S beim Modell B+	491
12.5 1-Wire	491
13 Sensoren	493
13.1 PIR-Bewegungssensor	493
Hardware	493
Programmierung	495
13.2 Ultraschallsensor	497
Hardware	497
Programmierung	498
13.3 Wasserstandssensor	500
13.4 Temperatursensoren	501
Temperatursensor DS1820	501
Temperatur- und Luftfeuchtigkeitssensor DHT22/AM2302	503
13.5 Fotowiderstand	506
13.6 Multi-Analogsensor-Board PCF8591	508
13.7 Raspberry Pi Camera Board und PiNoIR	511
Anschluss	512
raspistill und raspivid	514
Kamerasteuerung durch Python	516
Videos und Bilder am Raspberry Pi anzeigen	517
Videostreams und Online-Zugriff	519
VLC-Livestream	520
FTP-Upload	521
PiNoIR	522
Infrarot-Scheinwerfer für die PiNoIR	522

13.8 Reed-Kontakt	524
13.9 Hardware Real Time Clock	526
13.10 IR-Empfänger	530
13.11 Leitfaden für fremde Sensoren	532
14 Erweiterungsboards	535
14.1 Das Gertboard	536
Inbetriebnahme	537
Die drei Taster	538
Digitale Ein- und Ausgänge und Leuchtdioden	541
Der Open-Collector-Treiber	543
Der Motortreiber	544
Der Analog/Digital-Wandler	547
Der Digital/Analog-Wandler	549
14.2 Der ATmega auf dem Gertboard	551
Hello World!	552
avrduude	553
Die Arduino-IDE	554
14.3 RasPiComm	557
Der Joystick und die 5-V-Ausgänge	558
Die Echtzeituhr	559
Die Schnittstellen	559
14.4 PiFace Digital	560
Der PiFace-Emulator	561
Die PiFace-Bibliothek	562
Interrupts und Events in der PiFace-Bibliothek	563
PiFace Rack und die Jumper	564
14.5 Quick2Wire Interface Board	565
Anschlüsse	567
Das Quick2Wire-Port-Expander-Board	568
14.6 Unterbrechungsfreie Stromversorgung mit PiUSV	569
Inbetriebnahme	569
Konfiguration und Steuerung	571

14.7	Andere Erweiterungen im Überblick	572
	Alamode	573
	BrickPi	574
	GrovePi	576
15	Displays	579
15.1	16×2-LC-Display	579
	Pin-Belegung	580
	Programmierung	581
15.2	RaspiLCD	584
	Der Anschluss an den Raspberry Pi	584
	Die Ansteuerung mit Python	586
	Funktionsübersicht	590
15.3	PiTFT – Der Touchscreen für den Raspberry Pi	591
	Der Zusammenbau	592
	Die Einrichtung unter Raspbian	593
	Schnellkalibrierung	596
	Genaue Kalibrierung des Touchscreens	597
	Die Konsole auf dem PiTFT	600
	Weiterführende Links	601
15.4	Kfz-Rückfahrmonitore	602
	Darstellung optimieren	604
15.5	LS-7T – Der Plug-and-Play-Touchscreen	606
	Der Anschluss	608
	Bildausschnitt anpassen	608
	Die Kalibrierung des Touchscreens	608
15.6	Weitere Display-Boards	610
	C-Berry	610
	RPi-Display von Watterott	610
	SainSmart Mini TFT Display	611
	HDMIPI	611
	Displays für den DSI-Anschluss	612

TEIL IV Programmierung

16	Python kennenlernen	615
16.1	Python ausprobieren	616
	Python in einem Terminalfenster ausführen	616
	IDLE	617
16.2	Python-Konzepte interaktiv kennenlernen	618
	Zahlen	619
	Zeichenketten	621
	Listen	623
	Tupel (Sequenzen)	625
	Sets (Mengen)	626
	Dictionaries (Assoziative Arrays)	627
16.3	Hello World!	628
	Texteditoren	629
	»Hello World« mit LeafPad verfassen und ausprobieren	629
	»Hello World« in IDLE3 verfassen und ausprobieren	632
	Noch ein Einführungsbeispiel	633
17	Python-Grundlagen	635
17.1	Elementare Syntaxregeln	635
	Anweisungen	635
	Blockelemente	636
	print	637
	print in Python 2	638
	input	638
	Module und »import«	639
	Kommentare	640
	Eingebaute Hilfe	640
17.2	Variablen und Objekte	641
	Variablen	642
	Mutable oder immutable	643
	Veränderliche Daten kopieren	644
	Datentypen	645

Typumwandlung	646
Variablen und Daten vergleichen	647
Gültigkeit von Variablen	648
Garbage Collection	649
Shared References	650
17.3 Operatoren	651
Zuweisungen	653
Extended Sequence Unpacking	654
17.4 Verzweigungen (if)	654
if-Verzweigungen	654
Bedingungen	655
if-Kurzschreibweise	655
17.5 Schleifen (for und while)	656
Schleifen über Zahlenbereiche (range)	657
Schleifen über die Zeichen einer Zeichenkette	658
Schleifen über Listen, Tupel und Sets	658
Schleifen über Dictionaries	659
Schleifen über alle Script-Parameter	660
Schleifen über die Zeilen einer Textdatei	661
Schleifen über alle Dateien eines Verzeichnisses	661
17.6 Zeichenketten	662
Unicode	664
raw-Zeichenketten	664
String-Konvertierung und -Formatierung	664
Reguläre Ausdrücke	667
17.7 Listen	668
map	669
reduce	670
filter	671
zip	671
17.8 Umgang mit Fehlern (Exceptions)	672
try/except	672
Fehler in Funktionen und Methoden	674
with/as	675
Selbst Exceptions auslösen	676
Programmunterbrechungen abfangen	677

17.9 Funktionen	678
Eigene Funktionen definieren	678
Lokale und globale Variablen	680
Funktionsparameter	682
Optionale Parameter	683
Variable Parameteranzahl	684
Lambda-Funktionen	685
Funktionales Programmieren	687
17.10 Objektorientiertes Programmieren	688
Eigene Klassen definieren	689
Methoden	690
Konstruktor	691
Klassenvariablen und Namensräume	692
Private Klassenvariablen	694
Getter- und Setter-Methoden	694
Operator Overloading	695
Beispiel: Rechteck-Klasse	695
Vererbung	697
Klassen, Objekte und Instanzen	698
17.11 Systemfunktionen	700
Zugriff auf die Programmparameter	700
Zugriff auf Standardeingabe und Standardausgabe	701
Modul-Metainformationen ermitteln	701
Programm beenden	701
Andere Programme bzw. Kommandos aufrufen	702
Warten (sleep)	703
18 Programmieren mit Python	705
18.1 Programmieren lernen	705
Beispiel 1: Die Summe der Zahlen von 1 bis 1000 ausrechnen	706
Beispiel 2: Die Fakultäten von 1! bis 40! ausrechnen	707
Beispiel 3: Palindromtest	709
18.2 GPIO-Zugriff	710
GPIO-Pins lesen und verändern	710
Taster und Schalter	713
Software-PWM	715

18.3 Kamera	716
Fotos aufnehmen	716
Videos aufnehmen	718
Kameraparameter steuern	719
18.4 Dateien bei Dropbox hochladen	720
18.5 E-Mails versenden	721
Eine E-Mail mit Bitmap versenden	722
Eine Aufnahme der Raspberry-Pi-Kamera versenden	723
18.6 Textdateien lesen und schreiben	724
CSV-Dateien	725
Messdaten verarbeiten	727
18.7 Grafikprogrammierung	729
pygame und Python 3	729
Hello World!	730
pygame-Zeichenmethoden	731
Bilder speichern	733
Grafikspielereien	734
Temperaturkurve zeichnen	738
18.8 Grafische Benutzeroberflächen mit tkinter	740
Steuerelemente anordnen	742
Ereignisverarbeitung	746
Aufräumarbeiten beim Programmende	747
LED ein- und ausschalten	748
Helligkeit einer Leuchtdiode mit PWM steuern	750
Tastenzustand anzeigen	751
19 bash-Programmierung	755
19.1 Einführung	755
Hello World!	755
Noch ein Beispiel	755
Elementare Syntaxregeln	756
Kommandos aufrufen	757
In andere Scripts verzweigen	758
bash oder Python?	758

19.2 Variablen	759
Variablen mit read einlesen	760
Umgebungsvariablen	761
Vordefinierte bash-Variablen	761
Felder	762
Parametersubstitution	762
Zeichenketten bzw. Aufzählungen bilden	764
19.3 Schleifen, Bedingungen und Funktionen	764
if-Verzweigungen	765
test	767
case	768
for	769
while und until	770
break und continue	772
function	772
Umgang mit Fehlern	773
exit	774
Reaktion auf Signale (trap)	774
19.4 WiringPi	775
gpio-Kommando	776
LED ein- und ausschalten	778
LED-Helligkeit steuern (PWM)	779
LED-Temperatur-Ampel	779
20 C-Programmierung	783
20.1 Hello World!	783
Der GNU-C-Compiler	783
make	784
20.2 GPIOs steuern	787
GPIO-Steuerung mit der WiringPi-Bibliothek	787
GPIO-Steuerung mit der bcm2835-Bibliothek	789
21 Java-Programmierung	791
21.1 Erste Schritte	791
Hello World!	791
Java 8	792

Entwicklungsumgebungen (IDEs)	793
Java versus Python	795
21.2 GPIO-Steuerung mit Java	795
Installation	796
Hello Pi4J!	796
Mehr DigitalOutput-Steuerungsmöglichkeiten	798
Pulse Width Modulation	798
Eingaben verarbeiten	799
22 PHP-Programmierung	801
22.1 Apache installieren und konfigurieren	802
Apache-Konfiguration	803
22.2 Webverzeichnisse einrichten und absichern	804
Eigene Webverzeichnisse definieren	806
Beispiel	807
Verzeichnisse absichern	808
Passwortschutz für Webverzeichnisse	808
.htaccess-Datei	810
22.3 HTTPS	811
22.4 PHP installieren und konfigurieren	812
PHP ausprobieren	813
22.5 MySQL installieren und administrieren	814
MySQL installieren	814
MySQL-Konfiguration	815
Das mysql-Kommando	815
Neue Datenbanken einrichten	816
Datenbanken sichern und wiederherstellen	817
Administration mit phpMyAdmin	817
22.6 Hello World! in PHP	819
Fehlersuche	820
22.7 GPIO-Programmierung mit PHP	821
GPIOs auslesen und verändern	821
LED ein- und ausschalten	822
Temperatur auslesen	823

22.8 Kamerafunktionen mit PHP nutzen	824
Apache den Kamerazugriff erlauben	824
Ein Echtzeitfoto erstellen und übertragen	825
Ein Foto mit frei wählbaren Aufnahmeparametern erstellen	825
23 Mathematica und die Wolfram Language	829
23.1 Installation und Start	830
23.2 Arbeiten mit Mathematica	830
Die grafische Benutzeroberfläche von Mathematica	830
Rechnen in Mathematica	832
Formeleingabe	833
Grafik	834
Mathematica im Textmodus	835
Grafiken im Textmodus anzeigen	836
Interna	837
23.3 Programmieren mit der Wolfram Language	837
Hello World!	838
Syntax	838
Funktionale Programmierung und symbolische Ausdrücke	840
GPIO-Zugriff	841
RaspiCam	843
Serial, UltimateGPS etc.	844
Beispiel – Grafische Darstellung von Temperaturdaten	844
Wolfram Language – Pro und Kontra	848
Weitere Informationen und Links	848

TEIL V Projekte

24 Der Raspberry Pi im Vogelhaus	851
24.1 Einbau des Raspberry Pi samt Kameramodul in ein Vogelhaus	851
24.2 Kamerapraxis	854
Das Kameramodul betriebsbereit machen	854
Kamera-LED deaktivieren	854
Standbilder mit raspistill aufnehmen	855
ISO-Einstellung	856

Zeitverzögerung und Zeitrafferfilme	856
Videos aufzeichnen mit raspivid	857
24.3 Bewegungserkennung mit motion	858
Motion konfigurieren	858
24.4 Das Vogelhaus im praktischen Einsatz	862
Lichtverhältnisse und Bildqualität	862
25 Zeitmessung mit Lichtschranken	865
25.1 Versuchsaufbau (Hardware)	865
Schaltungsaufbau mit IR-Lichtschranken	865
25.2 Software	868
pigpio-Bibliothek	868
Programmcode	869
26 Das autonome Auto	871
26.1 Hardware	871
Das Chassis	871
Der Liniensensor	872
Die Mechanik	873
Anschlusspläne	876
Die mobile Stromversorgung des Raspberry Pi	877
Stromversorgung der Motoren	878
Ausrichtung und erste Tests	878
26.2 Die Software	880
Die Jungfernfahrt	882
27 RFID-Türöffner mit Zutrittskontrolle	885
27.1 RFID-Hardware	886
Die RFID-Technologie	886
Reader und Transponder	886
Verdrahtung	888

27.2 Software	889
Die Zugangsdatenbank	889
Python-Code zum Auslesen der RFID-Transponder	890
Python-Code zum MySQL-Zugriff	891
Python-Code zum Hinzufügen der Transpondernummern	892
Python-Hauptprogramm	893
phpMyAdmin – der Hausmeister	895
27.3 Erweiterungsmöglichkeiten und Tipps	895
Den Schrank abschließen	895
LCD-Modul statt bash-Ausgabe	896
Sensor statt Zeitsteuerung	896
Alarm bei falschem Transponder	896
Die Empfangsantenne verstecken	896
28 Stromzähler auslesen	897
28.1 Stromzähler-Grundlagen	897
Lesekopf und Signalwandler	898
Die Smart Message Language	900
28.2 Einführung in RRDTool	901
Die Round-Robin-Datenbank erstellen	901
Werte in die Datenbank eintragen	903
Graphen erzeugen	903
28.3 Zählerdaten speichern und zu Graphen aufbereiten	906
29 Hausautomation mit Netzwerksteckdosen	909
29.1 Einführung	909
Universal-Schaltbox	910
29.2 Programmierung	911
Die Leistung der Photovoltaikanlage ermitteln	911
Das Web-Interface	914
Die Heizungssteuerung	917
Die Steuerung der schaltbaren Steckdose	921

30 Hausautomation mit 433-MHz-Funktechnologie	923
30.1 Hardware-Grundlagen	923
Funksteckdosen	923
Das Sendemodul	924
Anschluss des Sendemoduls	924
Vorbereitung der Steckdosen	925
30.2 Software und Steuerungsbeispiele	926
rcswitch-pi	926
Die klassische Zeitsteuerung	927
Automatische Anwesenheitserkennung	928
Überwachung von Analogsensoren	930
Fernbedienung per Weboberfläche	931
Noch mehr Ideen	933
31 Ort und Zeit per GPS empfangen	935
31.1 Haben Sie Zeit? Eine Uhr für den Raspberry Pi	935
Den NTP-Server installieren und konfigurieren	936
Das GPS-Modul in Betrieb nehmen	938
Die serielle Konsole deaktivieren	939
Den GPS-Dämon installieren und konfigurieren	940
Den NTP-Server mit dem GPS-Zeitsignal füttern	941
Das GPS-Zeitsignal mit PPS synchronisieren	942
31.2 Ortsbestimmung mit dem Raspberry Pi	944
Ortsangaben per Python aus dem Datenstrom filtern	945
Ortsangaben in ein Foto einbetten	947
32 Der Raspberry Pi lernt twittern	949
32.1 Voraussetzungen	949
Installation der benötigten Werkzeuge	949
Twitter-App registrieren	949
32.2 Programmierung	951
Die Twitter-Software in Python	951
Der erste Tweet	952
CPU-Temperatur twittern	953
Bilder twittern	955

33 Monitoring	957
33.1 Munin	957
Munin-Node: Installation und Konfiguration	958
Jäger und Sammler: Munin	960
33.2 SmokePing	962
Installation von SmokePing	964
Konfiguration	965
Mehr als Pings	967
34 Raspberry Pi-ratensender	969
34.1 FM-Transmitter einrichten	969
Technischer Hintergrund	969
Software-Installation	970
MP3-Dateien umwandeln und senden	970
Der DJ spricht	970
35 WLAN- und IPv6-Router	971
35.1 Einführung	971
Glossar	971
Voraussetzungen	972
Die schwierige Suche nach einem geeigneten WLAN-Adapter	973
Einschränkungen	976
35.2 WLAN-Access-Point	976
Netzwerk- und Brückenkonfiguration (/etc/network/interfaces)	978
WLAN-Authenticator (hostapd)	979
hostapd für Realtek-8118-Chipsets	981
35.3 WLAN-Router	982
Router-Funktionen	983
Konfigurationsüberblick	984
Netzwerkkonfiguration (/etc/network/interfaces)	985
Network Address Translation (NAT)	986
Forwarding	987
DHCP- und Nameserver (Dnsmasq)	987
Inbetriebnahme und Test	990

35.4 WLAN-Repeater	990
Unveränderliche Zuordnung der WLAN-Schnittstellen	991
Konfigurationsüberblick	992
Netzwerkkonfiguration (/etc/network/interfaces)	993
WLAN-Authenticator (hostapd) und DHCP-Server (Dnsmasq)	994
Inbetriebnahme	994
35.5 IPv6-Router	994
Grundlagen	995
Konfigurationsüberblick	996
IPv6 aktivieren	997
IPv6-Tunnel einrichten (SixXs)	997
Netzwerkkonfiguration (/etc/network/interfaces)	998
IPv6-Forwarding erlauben	999
radvd einrichten	999
Inbetriebnahme und Test	1000
36 Tor-Router	1003
36.1 Tor-Grundlagen	1003
Das Tor-Netzwerk	1004
Sicherheit und Recht	1005
36.2 WLAN-Router-Konfiguration	1006
Konfiguration der Netzwerkschnittstellen	1006
Ein schlanker DHCP-Server	1007
Konfiguration des HostAPD	1009
IP-Forwarding	1010
Masquerading	1010
Troubleshooting	1011
36.3 Anbindung des WLAN-Routers an das Tor-Netz	1012
37 AirPrint-Drucker selbst gemacht	1015
37.1 Voraussetzungen	1015
CUPS-Installation und -Konfiguration	1016
Druckerkonfiguration	1016
Druckgeschwindigkeit erhöhen	1018

37.2 AirPrint-Konfiguration	1020
Netzwerkdruck erlauben	1020
AirPrint-Beschreibung des Druckers für Avahi einrichten	1020
Praktische Erfahrungen	1021
38 Automatische Präsentationssysteme	1023
38.1 Einführung	1023
Tastensteuerung und Reset	1024
Datenquelle	1024
Fertige Lösungen	1025
38.2 Werkzeugkasten	1026
Bilder anzeigen mit pqiv	1026
Bilder anzeigen mit fbi	1027
Bilder optimal aufbereiten	1027
MP3-Dateien abspielen	1028
Videos abspielen	1028
Autostart	1029
Bilder von einem USB-Stick lesen	1030
Bildschirmschoner abstellen	1031
38.3 Reset/Shutdown-Taste	1032
Hard-Reset-Button	1032
Soft-Reset-Button	1033
38.4 Autonome Bilder-Projektion	1034
Ausgangspunkt	1035
Bildverzeichnis synchronisieren	1035
Bilder anzeigen	1036
Start der Diashow	1037
Verbesserungsideen und Varianten	1037
38.5 Video-Player mit Menü	1037
Ausgangspunkt	1038
python-pygame-menu-class	1039
uiinput	1040
Das Menüsystem	1040
Start des Videosystems	1044
Verbesserungsideen und Varianten	1045

Inhaltsverzeichnis

39 Luftraumüberwachung	1047
39.1 Technischer Hintergrund und Hardware	1047
39.2 Software	1048
Installation von RTL-SDR und dump1090	1049
Die Position der Flugzeuge im Webbrowser verfolgen	1049
Index	1051

Index

1-Wire-Sensor	491
<i>PHP</i>	823
<i>Python</i>	751
1st-Stage-Bootloader	261
7zr	141
16x2-Display	579
230-V-Verbraucher	909
433-MHz-Funktechnologie	923
7805-Linerarregler	373
8188eu-Kernelmodul	975
/boot-Verzeichnis	261
/boot/config.txt	270
/dev/vchiq-Datei	824
/etc/X11/xorg.conf	276
/etc/anacrontab	211
/etc/apt/sources.list	239
/etc/fstab	181, 246
/etc/group	227
/etc/hostname	257
/etc/inittab	264
<i>serielle Konsole deaktivieren</i>	940
/etc/lightdm/lightdm.conf	231, 277
/etc/login.defs	237
/etc/modules	279
/etc/motion/motion.conf	858
/etc/network/interfaces	252, 1006
<i>WLAN-Access-Point</i>	978
<i>WLAN-Router</i>	985
/etc/ntp.conf	936
/etc/passwd	225
/etc/rc.local	206
/etc/resolv.conf	150, 256
/etc/samba/smb.conf	183
/etc/shadow	226
/etc/sudoers	230
/etc/sysctl.conf	987, 1010
/etc/tor/torrc	1012
/etc/udhcpd.conf	1007
/lib/modules	278
/proc/cpuinfo	272

A

AA-Batterien	372
aaline (<i>pygame</i>)	732
Abiword	98
Abkürzungen (alias)	115
Abstürze	81
Access-Point	971, 976
add_event_detect	714
add_event_callback	714
adduser	229
ADS-B-System	1047
ADSL-Router	189
after (<i>tkinter</i>)	748
Agent (Plex)	342
AirPlay	305
<i>Volumio</i>	302
<i>XBMC</i>	336
AirPrint	1015
Akku-Stromversorgung	370
Alamode	573
alias (<i>bash</i>)	115
Alias (<i>Apache</i>)	806
ALSA	287
alsa-utils	970
AM2302	503
amixer	289
Anacron	210
Analog-Audio	97
<i>einschalten</i>	289
Analog/Digitalwandler	465
Anologsensoren	508
anonyme Funktionen (<i>Python</i>)	685
Anonymisierdienst (Tor)	1003
Anschlüsse	46
Anti Aliasing (<i>pygame</i>)	732
Apache	802
Apple	
<i>AirPlay</i>	305, 336
<i>AirPrint</i>	1015
<i>ApplePi-Baker</i>	64
<i>Fernbedienung</i>	327
<i>Image-Datei schreiben</i>	64

APT	238
apt-cache	155
apt-get	154
Archive	
tar	142
zip	143
Arduino	
Alamode	573
Gertboard	551
GrovePi	576
IDE	554
arecord	970
argv (Python)	700
arm_freq	215
ARMv6-Architektur	359
Arrays (bash)	762
asound.state-Datei	289
assoziative Arrays	
bash	762
Python	627
Attribute (Python)	688
Audio-Ausgang	97
Audio-Bus	489
Audio-Player	
LXMusic	97
MPD	285
XBMC	335
Audio-Streams (Icecast)	300
Ausgabeumleitung	120
Ausschalten	80
Auto-Login	231, 277
automones Auto	871
Autostart	206, 1029
autostart-Verzeichnis	207
Avahi	1021
avconv	857
avrsetup	553

B

Backpower	376
Backups	211
SD-Karte	213
bash	
im Terminal	118
Programmierung	755
Basiswiderstand	410
Batteriebetrieb	372
BCM-Nummerierung	362
BCM2835	359
bcm2835-Bibliothek (C)	789
Benutzerverwaltung	224

C

C (Programmiersprache)	783
C-Berry	610
call (Python)	702
Callback (Python)	417, 714
case (bash)	768
cat	127
cd	133
CEC-Fernbedienung	324
Chassis	871
check_output	702

chgpt	236
Chicken (VNC)	171
chmod	630
chown	236
CIFS	293
circle (pygame)	732
Class-4/6/10 (SD-Karten)	49
cleanup (Python)	713
Clock-Signal	969
cmdline.txt	261
serielle Konsole deaktivieren	939
Cobbler	366
collabora-Paketquelle	240
Compute Module	44
config.piusv-Datei	571
config.txt	261
Beschreibung der Optionen	270
Display-Optionen	219
OpenELEC	320
Overclocking	215
RasPlex	349
sicher ändern	220
console (Kerneloption)	262
continue	
bash	772
Python	656
convert (imagemagick)	947, 1028
Cover-Art (MPD)	296
cp	138
CPU-ID	272
cpuinfo-Datei	272
Cron	208
Temperatur protokollieren	845
CSV-Dateien (Python)	725
CUPS	105
AirPrint	1015
curl	152

D

Dämonen	148
Darlington-Transistor	412
Dateien	
auflisten	135
komprimieren	141
kopieren	138
löschen	138
suchen	137
Zugriff in Python	724
Dateimanager	90
mit root-Rechten	125
Dateisystem	242
DateListPlot (Mathematica)	846
Datentypen (Python)	645
Datum	935
Deadline-Scheduler	262
decode MPG2	272, 320
decode WVC1	272, 320
Decodierer (MPEG-2, VC-1)	320
Defaulteditor	130
Dekorator (Python)	691
Desktop	81
desktop-Datei	1029
Device-Dateien	242
DeviceOpen (Wolfram)	844
DeviceRead (Wolfram)	841, 844
DeviceWrite (Wolfram)	841
DHCP	249, 987
DHCP-Server	1007
udhcpd	987
DHT22	503
Dia-Show	1023
Dictionaries (Python)	627
Schleifen	659
zip	671
Digital/Analog-Wandler	296, 470
digitales Potentiometer	458
dimmen (LEDS)	419
Diode	390
Directory (Apache)	806
disable_camera_led	273
Display Manager	275
Displays	579
C-Berry	610
DSI-Display	612
HDMI Pi-Display	611
Kfz-Rückfahrronitor	602
PiTFT	591
Probleme beheben	219
RaspLCD	584
RCA-Monitor	602
RPi-Display	610
SainSmart-TFT-Display	611
Touchscreen	606
Zeilendisplay	579
Distribution	54
DLNA	
Audio-Dateien (MPD)	303
Video-Dateien (XBMC)	332
dmrc-Datei	276
DNS	256
dynamisches DNS	194
DocumentRoot-Verzeichnis	803
dpkg	156
DPMS	87

DrJava 794
 Dropbox (Python) 720
 Drucker 105
 AirPrint 1015
 Geschwindigkeit erhöhen 1018
 DS1307 526
 DS1820 501
 Temperaturkurve (pygame) 738
 Temperaturkurve (Wolfram Language) 844
 DSI-Anschluss 612
 du 134
 dump1090 1048
 Durchgangsprüfung 397
 Duty Cycle 419
 DVB-T-Empfänger 1047
 dynamisches DNS 194

E

E-Mail versenden (Python) 721
 Edimax-WLAN-Adapter 973, 981
 Editoren 93, 129
 Defaulteditor einstellen 130
 mit root-rechten 125
 EEPROM 368
 Ein-/Ausschalten 80
 Eingabe (Python) 638
 Eingabeumleitung 120
 Einschaltprozess (Bootvorgang) 259
 Elektromotor 427
 Elektronik-Grundkurs 383
 elevator (Kerneloption) 262
 else (Python) 654
 Entprellen von Tastern 713, 715, 1041
 Entwicklungsumgebung
 Java 793
 Python 617
 eq-Methode (Python) 696
 Ereignisse (tkinter) 746
 Error
 bash 773
 Python 672
 Erweiterungsboards 535
 ESD 367
 event_detected 714
 Events 417
 EW-7811UN (Edimax-WLAN-Adapter) 981
 Exceptions (Python) 672
 exec (PHP) 821
 exit
 bash 774
 Python 701

export (bash) 761
 Export (Mathematica) 836
 ext4-Dateisystem 242
 Extended Sequence Unpacking 654

F

fbi 517, 1027
 fdisk 70
 Fehler
 bash 773
 Python 672
 Felder (bash) 762
 Fernbedienung (XMBC) 324
 feste Links 139
 ffmpeg 970
 file (Python) 724
 file_get_contents (PHP) 823
 filter (Python) 671
 find 137
 Firefox 94
 Firmware 261
 Update 158
 Flugsicherung 1047
 FM-Transmitter 969
 Mikrofon 970
 pifm 970
 Sendefrequenz 970
 Fonts von Microsoft 95
 for
 bash 769
 Python 656
 force_turbo 216
 format (Zeichenketten) 664
 Formatierung von Datenträgern 247
 Forwarding
 IP-Forwarding 987, 1010
 IPv6-Forwarding 999
 Port Forwarding 196
 Fotos
 anzeigen 1026
 aufnehmen in Python 716
 Fototransistor 865
 Fotowiderstand 506
 Freigaben (Netzwerk-verzeichnisse) 176, 182
 Freischaltung I²C-Bus 478
 Frodo 310
 fstab-Datei 246
 FTP 151, 203
 Kamera-Upload 521
 Verzeichnis synchronisieren 1035

function (bash) 772
 Funksendemodul 924
 Funksteckdosen 925
 funktionales Programmieren
 Python 687
 Wolfram Language 840
 Funktionen (Python) 678
 future (Python) 729

G

Garbage Collection (Python) 649
 Gateway 251
 gcc 783
 Geany 793
 Gedit 94
 Gehäuse 50, 377
 geografische Breite/Länge 944
 Gertboard 536
 A/D-Wandler 547
 Arduino IDE 554
 ATmega 551
 D/A-Wandler 549
 digitale Ein-/Ausgänge 541
 Inbetriebnahme 537
 Motortreiber 544
 Open-Collector-Treiber 543
 Spezifikationen 536
 Taster 538
 Geschwindigkeitsmessung 865
 Getter-Methoden (Python) 694
 GhostScript 1018
 globale Variablen (Python) 681
 Gnumeric 98
 gogoc-Installation 200
 Gotham 310
 gpicview 96
 GPIO 357
 bash 775
 C 787
 C mit bcm2835-Bibliothek 789
 C mit WiringPi 787
 Java 795
 Nummerierungssysteme 362
 PHP 821
 Pinbelegung 360
 Python 710
 Wolfram Language 841
 gpio-Kommando 776
 GpioController 796
 GpioPinDigitalInput 799
 GpioPinDigitalOutput 796

GpioPinPwmOutput 798
 GpioSetStateTrigger 799
 GPS 935
 Modul 938
 Ortsbestimmung 944
 Shield 938
 gpsd 940
 GPT-Partitionen 248
 GPU-Firmware 261
 gpu_freq 215
 gpu_mem 274
 Grafikprogrammierung (Python) 729
 Grafiksystem 275
 Autostart 276
 HDMI 271
 Grafiktreiber 277
 grafische Benutzeroberflächen 740
 grep 128
 mit find kombiniert 138
 grid (tkinter) 745
 GrovePi 576
 Gruppen 224
 gunzip 141
 gzip 141

H

H-Brücke 430
 halt-Kommando 266, 1033
 Hard-Reset 1032
 Hardware-PWM 421
 Hausautomation 923
 HCSR04 497
 HDMI-Einstellungen 271
 HDMIPI-Display 611
 Hello World
 bash 755
 C 783
 Java 791
 PHP 819
 pygame 730
 Python 628
 tkinter 740
 Wolfram Language 838
 Hintergrundprozesse 121
 hostapd 979, 1009
 Hostname 250
 ändern 257
 HP-Toolbox 1017
 hplip-gui 1017
 htaccess-Datei 810

httpd.conf-Datei 803
 HTTPing 920
 HTTPS 811

I

¹C 364, 476
 i2c-bcm2708 478
 i2c-tools 478
 i2cdetect 479
 i2cget 482
 i2cset 480
 I²S 489
 DAC 296
 IC 412
 Icecast 300
 Icedove 94
 Iceweasel 94
 Icons (LXDE) 82
 IDE
 Java 793
 Python 617
 IDLE (Python) 617, 632
 if
 bash 765
 Python 654
 ifdown 254
 ifup 254
 Image-Datei 62
 Apple/OS X 64
 Linux 66
 Windows 63
 imagemagick 947, 1028
 Immutable Types (Python) 643
 import (Python) 639
 info-Kommando 112
 Infrarot-Empfänger 530
 Infrarot-Scheinwerfer 522
 init-Kommando 266
 Init-System 263
 init.m (Mathematica) 837
 init_uart_baud 273
 init_uart_clock 273
 inittab-Datei 264
 serielle Konsole deaktivieren 940
 input (Python) 638, 712
 insserv 149, 267
 Installation
 auf einen USB-Stick 68
 Image-Datei 62
 NOOBS 54
 Instanzen (Python) 688

intelligenter Schrittmotortreiber 447
 interfaces-Datei 252, 1006
 WLAN-Access-Point 978
 WLAN-Router 985
 Internet-Freigabe 982
 Internet-Radio 302, 336
 Internet-Zugriff 189
 IP-Adresse 250
 ermitteln 149
 IP-Forwarding 987, 1010
 IPv6 999
 ip-Kommando 149
 iptables 986
 IPv6 199
 aktivieren 200
 Router 994
 Tunnel einrichten 200
 IR-Fernbedienung 326
 items (Python) 660
 Iterator (Python) 669

J

J8-Header 46, 368
 Java 791
 jmacs (Editor) 132
 Jokerzeichen 119
 JTAG 364
 Jumperwire 366

K

Kamera 510
 bash 755
 ISO-Einstellung 856
 LED deaktivieren 854
 PHP 824
 Python 716
 Wolfram Language 843
 Kanal (Plex) 342
 Kernel 278
 Bootoptionen 261
 Mathematica 837
 Parameter 261
 Update 158
 kernel.img-Datei 158
 Key/Value-Paar (Python) 660
 KeyboardInterrupt (Python) 677
 kill 147
 killall 147

Klassen (Python) 688
 Kodi 310
 kombinierte Schaltung 387
 Kommandos
 ausführen (*bash-Scripts*) 757
 automatisch ausführen 206, 1029
 im Hintergrund ausführen 121
 mehrere Kommandos ausführen 121
 mit root-Rechten ausführen 124
 regelmäßig ausführen 208
 Substitution 122
 verknüpfen 122
 Komponentenübersicht 357
 Kondensator 392
 Konsole 115
 Konstruktor 691
 Kontext-Manager (Python) 675

L

L298 432
 Lambda-Funktion (Python) 669, 685
 LC-Display 579, 584, 591
 le-Methode (Python) 696
 LeafPad (Editor) 93, 629
 LEDs 390, 407
 dimmen 419
 Status-LEDs 220
 less 127
 Leuchtdioden 390, 407
 auf dem Raspberry Pi 220
 lftp 151, 1035
 LibreOffice 98
 librtlsdr-Fehlermeldung 1049
 Lichtschranke 865
 lightdm 275
 lightdm.conf 231, 277
 line (pygame) 732
 Line Follower 872
 Linearregler 7805 373
 Liniensor 872
 Link Power Management 262
 Links 139
 lirc_rpi-Modul 328
 lircd.conf 326
 List Comprehension (Python) 659
 listdir (Python) 661
 Listen (Python) 623, 668
 Livestream 520
 LM2596S-Schaltregler 374
 Lochrasterplatine 399
 Logger 914

lokale Variablen (Python) 680
 Loopback-Schnittstelle 250
 löten 400
 lp-Kommando 1017
 LPM 262
 ls-Kommando 135
 Farben 117
 LS-7T 606
 lsmod 278
 lt-Methode (Python) 696
 Luftdruck 506
 Luftfeuchtigkeitssensor 503
 Luftraumüberwachung 1047
 LXDE 81
 lxde-rc.xml 85
 LXMusic 97
 LXTerminal 91, 107
 Farben 116
 lzop 141

MAC-Adresse 250
 Magnetschalter 524
 mainloop (tkinter) 746
 make 784
 man 112
 map (Python) 669
 Map (Wolfram Language) 846
 mapfile (bash) 762
 Masquerading 1010
 Mathematica 829
 Maus 50
 Plex 341
 max_usb_current 274
 MBR-Partitionen 248
 MCP23017 477
 MCP3008 465
 Wassersensor 500
 MCP4132 458
 MCP4811 470
 Mengen (Python) 626
 Methoden (Python) 688, 690
 Microsoft-Fonts 95
 Midori 88
 MIMEText (Python) 721
 Mini-TFT 611
 Minicom 485
 Miracast 336
 mirror (lftp-Kommando) 1035
 mkdir 134
 mkfs 247

mmcblk-Devices 243
 mode2 328
 Modell A 44
 Modell A+ 44
 Modell B 42
 Modell B+ 42, 368
 Flachbandleitungen 403
 GPIO-Änderungen 368
 Hardware-PWM 422
 I²S 491
 Java/Pi4J 795
 kein Backpowering 376
 Kfz-Rückfahrmonitor 602
 Run-Header 1032
 Stacking Header 536
 Status-LEDs 221
 USB-Stromversorgung 273
 Wolfram Language 842
 zweiter I²C-Bus 368
 modinfo 279
 modprobe 279
 Module (Kernel) 278
 explizit laden 279
 nicht automatisch laden 279
 Module (Python) 639
 modules-Datei 279
 mogrify 1028
 Monitorprobleme 219
 Monitoring 957
 Munin 957
 SmokePing 962
 Visualisierung 961
 Mopidy 305
 motion 858
 motion.conf 858
 Motordrehzahl 438
 Motortreiber 432
 mount 244
 NAS-Verzeichnis 177
 MP3-Dateien
 lxmusic 97
 MPD 285
 mpg321 1028
 MPDroid 290
 MPEG-2-Decodierer 320
 Multimedia-Center
 MPD 285
 Plex 339
 XBMC 309
 Multimeter 395
 Munin 957
 Music Player Daemon (MPD) 285
 MusicBox 305

Mutable Types (Python) 643
 mv 138
 my.conf-Datei 815
 MySQL 814, 889
 Backup 817
 Datenbank einrichten 816
 Zugriff in Python 891
 mysql-Kommando 815
 mysqldb-Bibliothek 890
 mysqldump-Kommando 817

N

Nachtsichtgerät 522
 Namensraum (Python) 693
 Nameserver 251
 dynamisches DSN 194
 Konfiguration 256
 nano 130
 NAS-Gerät 292
 Audio-Dateien (MPD) 303
 Video-Dateien (XBMC) 332
 NAT (Network Address Translation) 986
 Neighbor Discovery Protocol 996
 Netzteil 48, 218
 Netzwerk
 dynamisches DNS 194
 Glossar 250
 Internetzugriff auf den Raspberry Pi 189
 IPv6 199
 Konfiguration 249
 manuelle WLAN-Konfiguration 254
 Netzwerkverzeichnisse freigeben 182
 Netzwerkverzeichnisse nutzen 176
 Router 971
 Samba 182
 Schnittstellen 250
 SSH 161
 VNC 169
 WLAN 100
 WLAN-Router 971
 Netzwerkbrücke 977
 Netzwerkkommandos 149
 Netzwerkmaske 251
 Netzwerkstatus 149
 Netzwerksteckdosen 909
 Alternative 433-MHz-technik 910
 Schalten per bash-Script 921
 Visualisierung 914
 Web-Steuerung 914
 Netzwerkverzeichnis 292
 Neustart-Button 1032

New-Style-Klassen (Python) 690
 Newsfeed abstellen (XBMC) 322
 nice 147
 NoIP 194
 NOOBS 58
 VNC 61
 Notebook-Oberfläche (Mathematica) 830
 Notfall 217
 NPN-Transistor 393
 NTP 526, 935
 ntp.conf 936
 ntpq 937

O

OBIS-Kennzahlen 900
 Objekte (Python) 641
 objektorientiertes Programmieren
 (iPython) 688
 Office-Software 98
 Ohmsches Gesetz 383
 Oktalcode (Zugriffsbits) 235
 Old-Style-Klassen (Python) 690
 omxplayer 97, 1028
 Sensoren 518
 open (Python) 724
 Openbox 85, 275
 Autostart 207
 OpenELEC 311
 OpenJDK 791
 Operatoren (Python) 651
 Operator-Overloading (Python) 695
 optionale Parameter (Python) 683
 Optokoppler 426
 Order (Wolfram) 847
 OS X
 Image-Datei schreiben 64
 VNC 171
 output (Python) 712
 over_voltage 215
 over_voltage_sdram 215
 Overclocking 272
 OpenELEC 321
 Raspbian 215, 312
 XBian 312
 Overloading (Python) 695

P

P1-Header 46, 360
 P2-Header 364
 P3-Header 364
 P5-Header 364
 P6-Header 364, 1032
 p7zip 141
 pack (tkinter) 743
 Pakete 238
 aktualisieren 157
 installieren 154
 suchen 155
 Paketverwaltung 238
 Panel (LXDE) 83
 Parallelschaltung 387
 Parametersubstitution (bash) 762
 parted 247
 Partitionierung 247
 pass (Python) 674
 passwd-Datei 225
 passwd-Kommando 230
 Passwort 230
 Apache 808
 Raspbian 230
 Samba 186
 path (Python) 701
 PATH-Variable 124, 757
 PCF8591 508
 PDF-Dateien anzeigen 96
 Photovoltaik-Wechselrichter 911
 PHP 801, 812
 Hello World 819
 php.ini-Datei 813
 phpinfo-Funktion 813
 phpMyAdmin 817
 Phtovoltaik 911
 Pi MusicBox 305
 Pi Presents 1025
 Pi-Baker 64
 Pi4J 795
 picamera (Python) 516, 716
 Pidora 54
 PiFace 560
 Emulator 561
 Jumper 564
 Python Bibliothek 562
 Rack 564
 Spezifikationen 560
 pifm 970
 pigpio 868
 Pin-Nummerierung 362
 Pinbelegung P1-Header 360

ping 150, 928
PiNoIR 511, 522
Pipe-Operator 122
PIR 493
PiTFT 591
PiUSV 569
Plex 339
Plot, Plot3D (Mathematica) 834
PNP-Transistor 393
Poly Fuse 369
polygon (pygame) 732
Port (TCP/IP) 250
Port Forwarding 196
Porterweiterung 477
ports.conf-Datei 199, 803
Powerbank 370
PPS-Signal 942
pqiv-Kommando 1026
Präsentationssysteme 1023
print (Python) 637
printenv (bash) 761
Programme 143
 automatisch ausführen 206, 1029
 installieren 154
 regelmäßig ausführen 208
Prompt 116
property-Dekorator (Python) 694
protocol (tkinter) 747
Prozesse 143
 abbrechen 147
ps 144
PS1 116
pscp.exe 167
pstree 146
Pulse Width Modulation 419, 715, 779
PulseAudio 295
PuTTY 162
PWM 419, 715, 779
 gpio-Kommando 779
 Java 798
 Modell B+ 422
 Python 715
 tkinter-Beispiel 750
pygame 729
 selbst kompilieren 729
pySerial 487, 945
Python 615
 Grundlagen 635
 interaktiv ausprobieren 616
 Syntax 635
 Tastatureingabe simulieren 1040
python_uinput 1040

Q

Quick2Wire-Interface-Board 565
Quick2Wire-Port-Expander-Board 568

R

Radio 336
Radiosender 969
radvd 999
raise (Python) 676
RAM 42
 Aufteilung GPU/CPU 274
range (Python) 657
Raspberry Pi 41
Raspbian 75
Raspbmc 311
Raspi Cam 511, 519
 LED deaktivieren (Sensoren) 523
 Zugriff in Python 516
raspi-config 75
raspi-Paketquelle 240
RaspiComm 557
RaspiLCD 584
raspistill 514, 855
raspivid 514, 857
RasPlex 345
RaspyFi 302
Raw-Drucker 1018
raw-Zeichenketten (Python) 664
rc.local-Datei 206
rc.xml-Datei 85
RCA-Monitor 602
rcswitch-pi 926
read (bash) 760
readline (Python) 725
Real Time Clock 526
Realtek-8188EU 974
Realtek-8188CUS 973
reboot-Kommando 266, 1033
Rechteck-Klasse (Python) 695
rect (pygame) 732
reduce (Python) 670
Reed-Kontakt 524
reguläre Ausdrücke (Python) 667
Reihenschaltung 386
Relais 394, 451
remmina (VNC) 172
render (pygame) 733
renice 147
Repeater 972, 990

Repositories 239

repr-Methode (Python) 696
Reset-Button 1032
resolv.conf 256
RFID 885
RGB-LED 422
RISC-OS 55
rm 134, 139
rmdir 134
root (Administrator) 124
root-Kerneloption 262
rootwait-Kerneloption 263
Round-Robin-Datenbank 911
Router 189, 971, 982
 Tor 1003
Router Advertisement 999
RPi-Display 610
RPi.GPIO-Modul 710
rpi_gpio_ntp 942
RRD-Datei 911
RRDTool 901
 Hausautomation 914
 Stromzähler 906
 Temperaturmessung 903
RS232 484
RS485 484
RSS-Newsfeed abstellen (XBMC) 322
rsync 212
RT5370-WLAN-Adapter 973
RTL-SDR 1048
RTL2832U-DVB-T-Chip 1047
Rückfahrmonitor 602
Run-Header 1032
Runlevel 264

S

Samba 182, 319
save (pygame) 733
save_mode_gpio 273
scaling-cur-freq-Datei 217
Scanner (Plex) 342
Schaltregler 374
Schlüssel/Wert-Paar (Python) 660
Schleifen (Python) 656
Schrittmotor 440
scp 167
Scratch (Programm) 77
Screenshots
 Raspbian 99
 RasPlex 351
 XBMC 323

Videos	
<i>anzeigen</i>	517
<i>aufnehmen</i>	514, 718
<i>konvertieren</i>	857
VLC	98
VNC	169
<i>mit SSH-Tunnel</i>	175
<i>NOOBS</i>	61
<i>vncserver</i>	169
Vogelhaus	851
<i>Bewegungserkennung</i>	858
<i>Kameramodul</i>	851
<i>NoIR-Kamera</i>	851
<i>Praxis</i>	862
Volumio	302
vsftp	204

W

wait_for_edge	714
wait_recording (Python)	718
Wassersensor	500
Wayland	277
wc	144
Webbrowser	88
Wetterstation	506
wget	151
which	124
while (Python)	656
Widerstand	383
Widerstandsmessung	396
Widgets (tkinter)	742
WiFi	971
Win32 Disk Imager	63
Window Manager	275
Windows	
<i>Image-Datei schreiben</i>	63
<i>SSH</i>	162
<i>VNC</i>	170
WiringPi	
<i>bash</i>	775
<i>C</i>	787
with/as (Python)	675

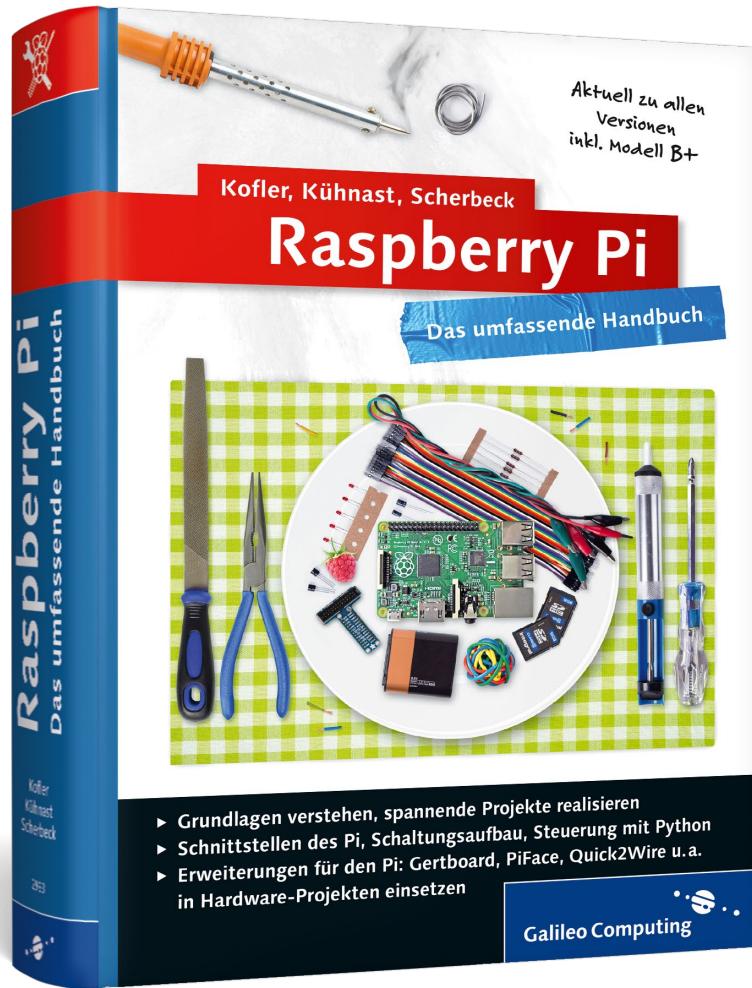
WLAN	971
<i>Access Point</i>	976
<i>Authenticator</i>	979, 1009
<i>Hotspot</i>	982
<i>Konfiguration</i>	100
<i>manuelle Konfiguration</i>	254
<i>OpenELEC</i>	316
<i>RasPlex</i>	349
<i>Repeater</i>	990
<i>Router</i>	971, 982
<i>Tor-Router</i>	1003
<i>Wolfram Language</i>	837
WPA	255, 980
wpa_gui-Programm	100
www-data-Account	803

X

X Window System	275
XBian	310
XBMC	309, 329
xsessions-Verzeichnis	1029
xfce4-screenshooter	99
Xfmpc	288
xorg.conf	276
xpdf	96
xz	141

Z

Zeichenketten	
<i>Zeichenketten (Python)</i>	621, 662
<i>Zeitrafferaufnahmen</i>	856
Zeroconf	1021
Zertifikate (HTTPS)	811
zip	143
zip (Python-Funktion)	671
Zuweisungen (Python)	653
zwei Events auf einem Eingang	417



Michael Kofler, Charly Kühnast, Christoph Scherbeck

Raspberry Pi – Das umfassende Handbuch

1.064 Seiten, gebunden, in Farbe, September 2014

39,90 Euro, ISBN 978-3-8362-2933-3

 www.galileo-press.de/3636



Michael Kofler studierte Telematik an der TU Graz. Er ist einer der erfolgreichsten und vielseitigsten Computerbuchautoren im deutschen Sprachraum. Zu seinen Themengebieten zählen neben Linux auch OS X, MySQL, KVM, Visual Basic und Excel-VBA. Viele seiner Bücher wurden übersetzt. Michael Kofler arbeitet auch als Software-Entwickler, Berater sowie als Lehrbeauftragter an zwei Fachhochschulen.



Charly Kühnast veröffentlicht bereits seit seinem 14. Lebensjahr Fachartikel, Kolumnen und Glossen zu IT-Themen. Auf einer Veranstaltung des Chaos Computer Clubs, dem er seit vielen Jahren angehört, lernte er das Betriebssystem Linux kennen. Er arbeitet beim Kommunalen Rechenzentrum Niederrhein in Kamp-Lintfort. Dort ist er für die Verfügbarkeit und Sicherheit der Internet-Infrastruktur verantwortlich. Daneben gibt er sein Wissen als Lehrbeauftragter an verschiedenen Hochschulen weiter.



Christoph Scherbeck ist begeisterter Raspberry-Pi-Bastler. Er ist 27 Jahre alt und gelernter Mechatroniker und Maschinenbautechniker. Privat betreibt er die Website www.elektronx.de, auf der er seine Bastelprojekte mit dem Raspberry Pi beschreibt.

Wir hoffen sehr, dass Ihnen diese Leseprobe gefallen hat. Gerne dürfen Sie diese Leseprobe empfehlen und weitergeben, allerdings nur vollständig mit allen Seiten. Die vorliegende Leseprobe ist in all ihren Teilen urheberrechtlich geschützt. Alle Nutzungs- und Verwertungsrechte liegen beim Autor und beim Verlag.

Teilen Sie Ihre Leseerfahrung mit uns!