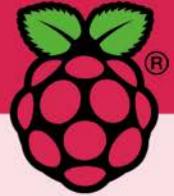




Eben  
Upton

Gareth  
Halfacree



# Raspberry Pi<sup>®</sup>

## Einstieg und User Guide



**HJR**

Verlagsgruppe  
Hühlig Jehle Rehm

## **Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)**

Der Verlag räumt Ihnen mit dem Kauf des ebooks das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und Einspeicherung und Verarbeitung in elektronischen Systemen.

Der Verlag schützt seine ebooks vor Missbrauch des Urheberrechts durch ein digitales Rechtemanagement. Bei Kauf im Webshop des Verlages werden die ebooks mit einem nicht sichtbaren digitalen Wasserzeichen individuell pro Nutzer signiert.

Bei Kauf in anderen ebook-Webshops erfolgt die Signatur durch die Shopbetreiber. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Eben Upton, Gareth Halfacree

# Raspberry Pi

## Einstieg und User Guide

Übersetzung aus dem Englischen  
von Maren Feilen



**mitp**

**Bibliografische Information der Deutschen Nationalbibliothek**

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <<http://dnb.d-nb.de>> abrufbar.

ISBN 978-3-8266-9562-9

1. Auflage 2013

E-Mail: [kundenbetreuung@hjr-verlag.de](mailto:kundenbetreuung@hjr-verlag.de)

Telefon: +49 6221/489-555

Telefax: +49 6221/489-410

[www.mitp.de](http://www.mitp.de)

Übersetzung der englischen Originalausgabe:

Eben Upton, Gareth Halfacree: Raspberry Pi User Guide, ISBN: 978-1118464465

Original English Edition Copyright © 2012 Eben Upton and Gareth Halfacree.

Published by John Wiley & Sons Ltd., The Atrium, Southern Gate, Chichester, West Sussex, PO19 8 SQ, United Kingdom

All Rights Reserved. Authorised translation from the English language edition published by John Wiley & Sons Limited. Responsibility for the accuracy of the translation rests solely with Verlagsgruppe Hüthig Jehle Rehm GmbH and is not the responsibility of John Wiley & Sons Limited. No part of this book may be reproduced in any form without the written permission of the original copyright holder, John Wiley & Sons Limited.

© 2013 mitp, eine Marke der Verlagsgruppe Hüthig Jehle Rehm GmbH  
Heidelberg, München, Landsberg, Frechen, Hamburg

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Lektorat: Sabine Schulz

Sprachkorrektorat: Petra Heubach-Erdmann

Satz: III-satz, Husby, [www.drei-satz.de](http://www.drei-satz.de)

*Für Liz, die dies alles ermöglichte.*  
– Eben

*Für meinen Vater, die passionierte Vergangenheit,  
und meine Tochter, die spannende Zukunft.*  
– Gareth



# Inhaltsverzeichnis

	Die Autoren .....	11
	Einleitung .....	13
<b>Teil I</b>	<b>Inbetriebnahme des Boards.</b> .....	<b>25</b>
<b>1</b>	<b>Gestatten: Der Raspberry Pi</b> .....	<b>25</b>
1.1	ARM vs. X86 .....	26
1.2	Windows/Apple OS X vs. Linux .....	27
1.3	Erste Schritte mit dem Raspberry Pi .....	29
1.3.1	Bildausgabegerät anschließen .....	29
1.3.2	Audioausgabe .....	31
1.3.3	Tastatur und Maus anschließen .....	32
1.3.4	SD-Karte flashen .....	34
1.3.5	Externen Datenträger anschließen .....	39
1.3.6	Netzwerkverbindung herstellen .....	40
1.3.7	Stromversorgung anschließen .....	43
<b>2</b>	<b>Linux-Systemadministration.</b> .....	<b>45</b>
2.1	Linux: Übersicht .....	45
2.2	Linux-Grundlagen .....	48
2.3	Einführung in Debian .....	49
2.4	Externe Datenträger verwenden .....	54
2.5	Neues Benutzerkonto erstellen .....	55
2.6	Dateisystemlayout .....	57
2.6.1	Logisches Layout .....	57
2.6.2	Physisches Layout .....	59
2.7	Software installieren und deinstallieren .....	60
2.7.1	Software finden .....	61
2.7.2	Software installieren .....	62
2.7.3	Software deinstallieren .....	64
2.7.4	Software upgraden .....	64

<b>3</b>	<b>Fehlerdiagnose und -behebung</b> .....	<b>67</b>
3.1	Tastatur und Maus .....	67
3.2	Stromstabilitätsprüfung .....	69
3.3	Bildausgabe .....	71
3.4	Systemstart .....	72
3.5	Netzwerk .....	72
3.6	Der Notfall-Kernel .....	75
<b>4</b>	<b>Netzwerkkonfiguration</b> .....	<b>79</b>
4.1	Kabelgebundenes Netzwerk .....	79
4.2	WLAN-Netzwerk .....	83
<b>5</b>	<b>Partitionsmanagement</b> .....	<b>95</b>
5.1	Neue Partition anlegen .....	95
5.2	Größe vorhandener Partitionen anpassen .....	100
5.2.1	Automatische Größenanpassung .....	100
5.2.2	Manuelle Größenanpassung .....	102
5.3	Auf eine größere SD-Karte wechseln .....	107
5.3.1	Imagedatei unter Linux erstellen .....	107
5.3.2	Imagedatei unter OS X erstellen .....	108
5.3.3	Imagedatei unter Windows erstellen .....	109
<b>6</b>	<b>Den Raspberry Pi konfigurieren</b> .....	<b>113</b>
6.1	Hardwareeinstellungen – config.txt .....	113
6.1.1	Bildausgabe konfigurieren .....	115
6.1.2	Booteinstellungen .....	119
6.1.3	Den Raspberry Pi übertakten .....	119
6.1.4	L2-Cache deaktivieren .....	124
6.1.5	Testmodus aktivieren .....	125
6.2	Speicherpartitionierung – start.elf .....	126
6.3	Softwareeinstellungen – cmdline.txt .....	128

---

**Teil II Der Raspberry Pi als Mediacenter, Arbeitsgerät und Webserver...** 131

<b>7</b>	<b>Der Raspberry Pi als Mediacenter</b> .....	<b>131</b>
7.1	Musikwiedergabe im Terminal .....	132
7.2	Dedizierter HTPC mit Rasbmc .....	134
7.2.1	Mediastreaming per Internet .....	136

7.2.2	Mediastreaming im lokalen Netzwerk . . . . .	139
7.2.3	Rasbmc konfigurieren . . . . .	141
<b>8</b>	<b>Der Raspberry Pi als Produktivitätstool . . . . .</b>	<b>143</b>
8.1	Cloudbasierte Anwendungen nutzen . . . . .	144
8.2	OpenOffice.org nutzen . . . . .	147
8.3	Bildbearbeitung mit The Gimp . . . . .	149
<b>9</b>	<b>Der Raspberry Pi als Webserver . . . . .</b>	<b>153</b>
9.1	Einen LAMP Stack installieren . . . . .	153
9.2	WordPress installieren . . . . .	158
<b>Teil III Programmierung und Hardware-Hacking . . . . .</b>		<b>163</b>
<b>10</b>	<b>Einführung in Scratch . . . . .</b>	<b>163</b>
10.1	Was ist Scratch? . . . . .	164
10.2	Beispiel 1: Hallo Welt . . . . .	165
10.3	Beispiel 2: Animation und Sound . . . . .	169
10.4	Beispiel 3: Ein einfaches Spiel . . . . .	171
10.5	Robotik und Sensorik . . . . .	179
10.5.1	Sensorik mit dem PicoBoard . . . . .	179
10.5.2	Robotik mit LEGO . . . . .	180
10.6	Referenzmaterial . . . . .	180
<b>11</b>	<b>Einführung in Python . . . . .</b>	<b>183</b>
11.1	Was ist Python? . . . . .	183
11.2	Beispiel 1: Hello World . . . . .	184
11.3	Beispiel 2: Kommentare, Eingaben, Variablen und Schleifen . . . . .	190
11.4	Beispiel 3: Spielen mit Pygame . . . . .	196
11.5	Beispiel 4: Python und der Netzwerkbetrieb . . . . .	207
11.6	Weitere Referenzquellen . . . . .	213
<b>12</b>	<b>Hardware-Hacking . . . . .</b>	<b>217</b>
12.1	Elektronische Bauteile und Werkzeuge . . . . .	217
12.2	Farbkodierung von Widerständen . . . . .	220
12.3	Bezugsquellen für Bauteile . . . . .	222
12.3.1	Lizenzierte Händler . . . . .	222
12.3.2	Online-/Versandhandel . . . . .	223
12.3.3	Fachhandel . . . . .	223

12.4	Die GPIO-Schnittstelle .....	224
12.4.1	Serieller UART-Bus .....	226
12.4.2	I <sup>2</sup> C-Bus .....	227
12.4.3	SPI-Bus .....	227
12.5	Den GPIO-Port in Python nutzen .....	227
12.5.1	Die GPIO-Python-Bibliothek installieren .....	228
12.5.2	GPIO-Ausgabe: Blinkende LED .....	230
12.5.3	GPIO-Eingabe: Einen Schalter auslesen .....	235
12.6	Von der Steckplatine zur Lochrasterplatine .....	240
12.7	Kurze Lötanleitung .....	243
13	<b>Erweiterungsboards</b> .....	249
13.1	Ciseco Slice of Pi .....	249
13.2	Adafruit Prototyping Pi Plate .....	253
13.3	Fen Logic Gertboard .....	256
<b>Teil IV Anhänge</b> .....		261
A	<b>Python-Programmlistings</b> .....	261
B	<b>HDMI-Anzeigemodi</b> .....	267
	<b>Stichwortverzeichnis</b> .....	273

# Die Autoren

**Eben Upton** ist Mitbegründer, Kurator und Geschäftsführer der *Raspberry Pi Foundation*. Er zeichnet für die allgemeine Hard- und Softwarearchitektur des Raspberry Pi verantwortlich und ist außerdem für das Key-Account-Management der Stiftung zuständig. Vor seiner Zeit bei der Foundation gründete er bereits zwei erfolgreiche Software-Start-ups für Mobile Games und Middleware – Ideaworks 3d Ltd. und Podfun Ltd. –, war als Direktor des Fachbereichs Informatik am St. John's College in Cambridge tätig und erwarb an der University of Cambridge die akademischen Grade Bachelor of Arts (BA), PhD und Master of Business Administration (MBA).

Hauptberuflich arbeitet Eben Upton als Entwickler anwendungsspezifischer integrierter Schaltungen und »Troubleshooter« für den US-amerikanischen Halbleiterhersteller Broadcom.

**Gareth Halfacree** ist freier Wissenschaftsjournalist und neben dem Projektmitbegründer Eben Upton Co-Autor des *Raspberry Pi User Guide*. Als ehemaliger Systemadministrator im Bildungssektor konnte sich Gareth seine Leidenschaft für die Open-Source-Welt im Laufe seiner beruflichen Karriere stets bewahren und so unterstützt er auch heute noch Projekte wie GNU/Linux, LibreOffice, Fritzing und Arduino durch Reviews, Dokumentationen und manchmal sogar aktive Mitarbeit. Außerdem begründete er selbst die Open-Hardware-Projekte »Sleepduino« und »Burnduino«, die die Funktionsfähigkeit der Physical-Computing-Plattform »Arduino« erweitern und ergänzen. Ein Portfolio seiner aktuellen Projekte findet sich unter <http://freelance.halfacree.co.uk>.



# Einleitung

»Die Jugendlichen von heute sind doch alle »Digital Natives««, meinte ein Mann, mit dem ich mich letztes Jahr auf einer abendlichen Feuerwerksveranstaltung unterhielt. »Ich verstehe nicht so ganz, was Sie mit diesem Ding, das Sie da bauen, eigentlich bezwecken wollen. Meine Kinder kennen sich mit PCs ohnehin schon besser aus als ich.«

Auf meine Frage, ob sie denn auch programmieren könnten, erwiderte er: »Warum sollten sie? Die Rechner leisten doch bereits alles, was die Kids von ihnen erwarten. Das ist ja schließlich auch Sinn und Zweck der Sache, oder?«

Tatsächlich sind aber viele junge Leute heutzutage eben keine »Digital Natives«. Die Vorstellung von jugendlichen Digitalfreaks, die sich unter wildem Gebrüll in feinstem Python zügellos von einem Twisted-Pair-Kabel zum nächsten schwingen, existiert zurzeit lediglich in unserer Fantasie. Vielmehr stellen wir im Rahmen der von der Raspberry Pi Foundation durchgeführten Bildungsprojekte immer wieder fest, dass sich die gesamte Interaktion der meisten jungen Leute mit digitaler Technik auf geschlossene Plattformen mit grafischen Benutzeroberflächen (GUIs) beschränkt, die sie nutzen, um Filme anzuschauen, Hausaufgaben mit der Textverarbeitung zu erledigen und Spiele zu spielen. Oder im Internet zu surfen, Bilder und Videos hochzuladen und Websites zu gestalten. (Meist sind sie zudem auch eher als ihre Eltern in der Lage, den heimischen Fernseher an die Satellitenbox anzuschließen.) Das ist zwar schon ein guter Anfang, es reicht aber bei Weitem nicht aus – vor allem dann nicht, wenn man bedenkt, dass in einem Land wie Großbritannien, in dem es in 20% der Haushalte immer noch keinen Computer gibt, längst nicht alle Kinder und Jugendlichen Zugang zu dieser Technik haben.

Anders als in der blühenden Fantasie meiner Zufallsbekanntschaft von der eingangs erwähnten abendlichen Feuerwerksveranstaltung programmieren sich Computer nicht von allein. Fakt ist, dass wir eine ganze Industrie mit zahllosen fähigen Informatikern und IT-Fachleuten brauchen, um technische Entwicklungen voranzutreiben – und natürlich Nachwuchskräfte, die in diese Positionen nachrücken, damit durch das Ausscheiden älterer Mitarbeiter keine Engpässe in der Branche entstehen. Das Schulen einer Fähig-

keit wie dem programmatischen, zielgerichteten Denken bewirkt aber noch weit mehr, als nur eine neue Generation von Programmierern und Hardware-Hackern hervorzubringen. Kreative Ideen und Aufgaben auf komplexe, nicht lineare Art strukturieren zu können, ist ein erlerntes Talent, und zwar eins, das jedem, der es sich aneignet – von Historikern über Designer und Anwälte bis hin zu Chemikern –, erheblichen Nutzen bringt.

## Programmieren macht Spaß!

Und zwar enormen, nutzbringenden und kreativen Spaß. Es ermöglicht die Realisierung großartiger, komplexer Dinge und das Austüfteln von (meiner Meinung nach noch großartigeren) cleveren, blitzschnellen und trügerisch einfach scheinenden Lösungen, um Unwägbarkeiten aller Art erfolgreich zu meistern. Man kann etwas erschaffen, das andere vor Neid erblassen lässt und einem selbst für den Rest des Tages ein wunderbares Gefühl der Selbstzufriedenheit beschert. In meinem Hauptberuf – ich entwickle die Art von Siliziumchips, die wir im Raspberry Pi als Prozessor verwenden, und kümmere mich auch um die Low-Level-Software, die darauf läuft – werde ich im Prinzip dafür bezahlt, den ganzen Tag einfach nur herumzusitzen und zu spielen. Und was könnte es Besseres geben, als auch anderen Menschen ein solches Dasein zu ermöglichen?

Vor allem ist es ja nicht so, als würden sich junge Leute nicht für die Computerbranche interessieren. Dass in diesem Zusammenhang Handlungsbedarf besteht, wurde mir vor ein paar Jahren bewusst, als wir das Raspberry-Pi-Projekt eher nebenbei betrieben. Die gesamte Entwicklungsarbeit an dem Gerät fand vorwiegend an den freien Abenden und Wochenenden der Verantwortlichen und freiwilligen Mitarbeiter der Stiftung statt, denn wir alle mussten noch einem Vollzeitjob nachgehen, um unsere Rechnungen bezahlen zu können. Dadurch geriet die Motivation natürlich hin und wieder ein wenig ins Stocken – in meinem Fall beispielsweise, wenn ich einfach nur mal wieder mit meinen Arrested-Development-DVDs (Anm. d. Ü.: US-amerikanische Comedy-Serie) und einem Glas Wein vor der Glotze herumgammeln wollte. Doch dann unterhielt ich mich eines Abends, an dem ich nicht bloß herumgammelte, mit dem Neffen eines Nachbarn über die Schulfächer, die er für sein General Certificate of Secondary Education (GCSE entspricht in etwa dem deutschen Schulabschluss der mittleren Reife) belegt hatte, und fragte ihn, welchen Beruf er denn später ergreifen wolle.

»Ich will Computerspiele schreiben«, antwortete er.

»Das ist toll! Was für einen Computer hast du denn zu Hause? Ich habe bestimmt ein paar Programmierbücher, die dich interessieren könnten.«

»Eine Wii und eine Xbox.«

Im weiteren Verlauf unserer Unterhaltung stellte sich letztendlich heraus, dass dieser wirklich intelligente junge Mann noch nie mit echter Programmierung zu tun gehabt hatte: Einerseits stand ihm zu Hause kein Computer zur Verfügung, auf dem er hätte programmieren können. Und andererseits wurde ihm in seinen IKT-Kursen (Informations- und Kommunikationstechnologie) – in denen er sich einen Rechner mit anderen Schülern teilen musste und lernte, wie Webseitendesign, Tabellenkalkulation und Textverarbeitung funktionierten – nicht auch nur das Geringste über die Nutzungsmöglichkeiten eines Computers im eigentlichen Sinne beigebracht. Er spielte jedoch leidenschaftlich gern Computerspiele. Und da der Wunsch, ein lieb gewonnenes Hobby zum Beruf zu machen, ja nun keineswegs allzu ungewöhnlich ist, hoffte er, die von ihm gewählten Schulabschlussfächer könnten ihm dies ermöglichen. Er besaß sicherlich die künstlerischen Fertigkeiten, die in der Computerspielbranche gefragt sind, und seine Noten in Mathematik und Naturwissenschaften waren auch nicht schlecht. Doch seine schulische Ausbildung ließ jeden Bezug zur Programmierung vermissen – der Lehrplan sah keinerlei Optionen zum Thema Datenverarbeitung vor, stattdessen wurden nur weitere IKT-Kurse angeboten, die sich schwerpunktmäßig auf die Belange der Endnutzer konzentrierten, nicht aber auf die Programmierung. Und angesichts der Möglichkeiten, die sich ihm im Hinblick auf den Umgang mit dem Computer zu Hause boten, waren seine Chancen, sich das Wissen anzueignen, das er für die Realisierung seines Berufsziels brauchte, verschwindend gering.

Genau diese Situation möchte ich jedoch nicht mehr erleben müssen: die sinnlose Vergeudung von Potenzial und Enthusiasmus. Natürlich bin ich nicht so verrückt zu glauben, der Raspberry Pi allein würde schon reichen, um die nötigen Veränderungen herbeizuführen. Ich bin allerdings fest davon überzeugt, dass er als Katalysator dienen kann. Die Bildungsinhalte an den britischen Schulen erfahren inzwischen eine bedeutende Umgestaltung, das Computerwesen hält zunehmend Einzug in den Lehrplan und das Unterrichtsfach IKT wird reformiert. Ebenso erleben wir seit der Einführung des Raspberry Pi einen massiven Bewusstseinswandel hinsichtlich der Missstände in Bezug auf die für Kids bereitgestellten bildungsbezogenen und kulturellen Entfaltungsmöglichkeiten.

Bei zu vielen der datenverarbeitenden Geräte, mit denen Kinder tagtäglich interagieren, handelt es sich um derartig geschlossene Systeme, dass sie sich

nicht kreativ als »Werkzeuge« einsetzen lassen – obwohl die Datenverarbeitung an sich eine kreative Angelegenheit ist. Versuchen Sie doch mal, Ihr iPhone als Robotersteuerung einzusetzen oder Ihre PS3 dazu zu bringen, ein von Ihnen selbst geschriebenes Spiel zu starten. Natürlich kann man auf dem PC programmieren, aber auch hier tun sich bedeutende Hürden auf, die viele Kinder einfach nicht überwinden können: Zum einen besteht die Notwendigkeit, spezielle Software herunterzuladen, und zum anderen sollte man am besten auch die Sorte Eltern haben, die nicht ständig besorgt sind, dass ihre Sprösslinge irgendetwas durcheinanderbringen, von dem sie selbst keine Ahnung haben, wie es wieder in Ordnung zu bringen ist. Viele Kinder wissen nicht einmal, dass das Programmieren auf dem heimischen Rechner überhaupt möglich ist. Sie betrachten den PC als eine Maschine mit netten anklickbaren Symbolen, die es ihnen ermöglichen, auf einfache Art das zu erledigen, was zu erledigen ist, ohne allzu viel darüber nachdenken zu müssen. Außerdem benutzen Mama und Papa diesen geschlossenen Kasten, um ihre Bankgeschäfte abzuwickeln – und sollte er mal nicht mehr funktionieren, kostet es eine Menge Geld, einen neuen anzuschaffen!

Der Anschaffungspreis des Raspberry Pi ist so gering, dass er mit dem Taschengeld weniger Wochen bezahlt werden kann. Und das übrige Equipment, das man braucht, um ihn funktionsfähig zu machen, besitzt fast jeder ohnehin bereits: einen Fernseher, eine SD-Karte, die aus einer alten Kamera stammen kann, ein Handy-Ladegerät, eine Tastatur und eine Maus. Der Raspberry Pi wird nicht mit der ganzen Familie geteilt, sondern gehört dem Kind oder Jugendlichen allein und ist klein genug, um ihn in die Tasche stecken und zu Freunden mitnehmen zu können. Sollte mal etwas schiefgehen, ist das auch keine große Sache – einfach eine neue SD-Karte rein und schon ist das Gerät wieder wie fabrikneu. Sogar sämtliche Tools, Arbeitsumgebungen und Lernmaterialien, die man braucht, um die langgezogene, flache Lernkurve zur Programmierung des Raspberry Pi erfolgreich nehmen zu können, stehen unmittelbar nach dem Einschalten bereit.

## Die Entstehungsgeschichte

Vor etwa sechs Jahren, damals war ich noch als Direktor des Fachbereichs Informatik an der Universität Cambridge tätig, begann ich mit der Entwicklungsarbeit an einem winzigen, erschwinglichen, rudimentären Computer. Ich hatte während meiner Lehrtätigkeit dort meinen Abschluss in Informatik gemacht und bastelte inzwischen an meiner Doktorarbeit, als mir auffiel, dass bei den jungen Leuten, die sich für den Studiengang Informatik einschrieben, ein klarer Kompetenzeinbruch festzustellen war. Mitte der

1990er Jahre hatten sich noch 17-Jährige an der Uni eingeschrieben, die zumindest Grundkenntnisse in verschiedenen Computersprachen mitbrachten, ein wenig über Hardware-Hacking Bescheid wussten und häufig sogar in Assembler arbeiteten. 2005 war dagegen ein Punkt erreicht, an dem die Studienanfänger höchstens gerade mal ein bisschen mit HTML experimentiert hatten – und bestenfalls den Hauch einer Ahnung von PHP und Cascading Style Sheets hatten. Es handelte sich immer noch um beeindruckend intelligente Leute mit jeder Menge Potenzial, aber ihre Erfahrungen im Umgang mit Computern waren dem Kenntnisstand, den wir früher erlebt hatten, weit unterlegen.

Das dreijährige Informatikstudium in Cambridge umfasst etwa 60 Vorlesungs- und Seminarwochen. Wenn allerdings schon allein das gesamte erste Jahr darauf verwendet werden muss, die Studenten auf den aktuellen Stand der Dinge zu bringen, ist es natürlich schwieriger, ihnen innerhalb der folgenden beiden Jahre auch noch all das Wissen zu vermitteln, das sie brauchen, um eine Doktorarbeit anzufangen oder in die freie Wirtschaft zu gehen. Die erfolgreichsten Absolventen des Grundstudiums – also diejenigen, die am Ende ihres dreijährigen Studiengangs die besten Leistungen erbracht hatten – waren diejenigen, die nicht nur dann programmierten, wenn sie dazu aufgefordert wurden, um etwa ihre wöchentlichen Aufgaben oder ein Studienprojekt zu realisieren. Es waren diejenigen, die auch in ihrer Freizeit programmierten. Die Grundidee hinter dem Raspberry Pi verfolgte also eine sehr einfache, klare (und recht schlichte) Zielsetzung: Ich wollte ein Tool entwickeln, das der geringen Anzahl an Bewerbern für diesen kleinen Studiengang einen schnellen Einstieg in die Materie ermöglichte. Meine Kollegen und ich hatten vor, die Minicomputer an den Tagen der offenen Tür kostenlos an die Schulabgänger zu verteilen und sie ein paar Monate später, wenn sie zu ihren Bewerbungsgesprächen nach Cambridge zurückkämen, zu befragen, was sie damit angefangen hatten. Wer dann etwas Interessantes vorzuweisen hätte, wäre definitiv für das Studium geeignet. Zu diesem Zeitpunkt gingen wir noch davon aus, dass wir vielleicht ein paar Hundert dieser Minirechner bauen würden, bestenfalls langfristig ein paar Tausend.

Als wir das Projekt dann aber ernsthaft in Angriff nahmen, dämmerte uns langsam, dass wir mit einem preiswerten kleinen Computer wie diesem noch viel mehr erreichen konnten. Allerdings war das Gerät, das wir damals zusammenbauten, noch meilenweit von dem heutigen Raspberry Pi entfernt. Ich lötete auf dem Küchentisch jeweils einen Atmel-Chip auf die längste Steckplatine, die man bei Maplin Electronics (eine britische Elektronik-Handelskette) kriegen konnte. Unsere ersten groben Prototypen waren

mit preiswerten Mikrocontrollern ausgestattet, mit denen sich ein Fernsehgerät mit Standardauflösung ansteuern ließ. Mit nur 512 KB RAM und einigen MIPS Rechenleistung wiesen diese Rechner eine ähnliche Performance auf wie die alten 8-Bit-Mikrocomputer – kaum vorstellbar, dass solche Geräte die Fantasie von Kindern und Jugendlichen anregen könnten, die an moderne Spielkonsolen und iPads gewöhnt waren.

Im Computerlabor der Universität hatte das allgemein vorherrschende IT-bezogene Bildungsniveau der Studenten schon länger für Diskussionen gesorgt – und als ich später einen nicht-akademischen Job in der freien Wirtschaft annahm, stellte ich auch hier bei jüngeren Stellenbewerbern genau dieselben Defizite fest. Also traf ich mich mit Dr. Rob Mullins und Professor Alan Mycroft (zwei Kollegen aus dem Computerlabor), Jack Lang (Dozent für den Studiengang Unternehmertum), Pete Lomas (Hardware-Guru) sowie David Braben (einer der führenden Köpfe der Computerspielbranche in Cambridge mit unbezahlbaren Kontakten) auf ein paar Bierchen (na ja, in Jacks Fall war es Käse und Wein) und wir gründeten die Raspberry Pi Foundation – eine kleine Wohltätigkeitsorganisation mit großen Ideen.

### Warum »Raspberry Pi«?

Wir werden häufig gefragt, wo der Name »Raspberry Pi« eigentlich herkommt. Nun, zunächst einmal wurde er von den diversen Treuhändern gemeinsam geprägt – und zählt damit zweifellos zu den sehr seltenen durch ein Gremium erzielten »Designerfolgen«, die mir bekannt sind. Um ehrlich zu sein, hielt sich meine Begeisterung anfangs allerdings sehr in Grenzen. (Inzwischen finde ich den Namen richtig gut, weil er richtig gut funktioniert – ich musste mich aber erst mal daran gewöhnen, denn ich selbst hatte das Projekt insgeheim jahrelang »ABC Micro« getauft.) Der Begriff »Raspberry« (Himbeere) ist auf die lange Tradition der Nennung von Früchten in den Namen von Computerfirmen zurückzuführen (neben dem allseits bekannten Unternehmen gab es ja in früheren Zeiten schon Tangerine (Mandarine) und Apricot (Aprikose) Computers – und auch Acorn (Eichel) bezeichnet eine Art Frucht). »Pi« ist eine Kurzform für »Python« – die Programmiersprache, die wir ursprünglich als die einzig mögliche Option für eine sehr viel weniger leistungsfähige Plattform ansahen, als der Raspberry Pi inzwischen bietet. Tatsächlich empfehlen wir Python nach wie vor als zu bevorzugende Programmiersprache für den Einstieg in und die Entwicklungsarbeit mit dem Raspberry Pi, darüber hinaus steht Ihnen jedoch eine ganze Welt voller weiterer Sprachoptionen zur Verfügung.

In meinem neuen Job als Chipentwickler bei Broadcom, einem großen Halbleiterhersteller, hatte ich Zugang zu preiswerter, aber zugleich äußerst leistungsstarker Hardware, die das Unternehmen für High-End-Handys produzierte – die Art, wie sie auch in HD-Video- und 14-Megapixel-Kameras verwendet wird. Der Qualitätsunterschied zwischen den Chips, die man als kleiner Entwickler für 10 Dollar kaufen konnte, und jenen, die man als Handyhersteller für etwa denselben Preis bekam, war erstaunlich: allgemeines Processing, 3D-Grafiken, Video und Speicher gebündelt in einer einzigen BGA (Anm. d. Ü.: Schaltungsgehäuse mit Kugelgitteranordnung) von der Größe eines Fingernagels. Diese Mikrochips verbrauchen nur sehr wenig Energie, bieten aber weitreichende Möglichkeiten. Da sie sich insbesondere für Multimedia eignen, wurden sie von Set-Top-Box-Unternehmen auch schon für High-Definition-Video verwendet. Ein solcher Chip schien der nächste logische Schritt für die Ausgestaltung des Raspberry Pi zu sein – also machte ich mich daran, eine preisgünstige Variante auszutüfteln, die einen ARM-Mikroprozessor an Bord hatte und die Verarbeitungsleistung aufbringen konnte, die wir brauchten.

Uns war sehr daran gelegen, einen Weg zu finden, Kinder und Jugendliche für den Raspberry Pi zu begeistern, auch wenn sie kein besonderes Interesse am Programmieren hatten. In den 1980ern musste man, wenn man ein Computerspiel spielen wollte, einen Kasten einschalten, der dann »bing« machte und ein Eingabefenster öffnete. Hier musste man dann ein bisschen Code eintippen und konnte loslegen. Viele Benutzer beließen es auch dabei, andere aber nicht – denn schon diese minimale Interaktion hatte sie dazu inspiriert, unbedingt lernen zu wollen, wie man programmierte. Da wir inzwischen erkannt hatten, dass sich der Raspberry Pi durchaus als winziges, äußerst leistungsfähiges und zugleich extrem preiswertes modernes Mediacenter einsetzen ließ, konzentrierten wir uns besonders auf diese Qualitäten, um auch die scheinbar Desinteressierten zu ködern – und sie so vielleicht ein wenig für die Programmierkunst begeistern zu können.

Nach etwa fünf Jahren Schufferei hatten wir schließlich einen ansehnlichen Board-Prototyp etwa von der Größe eines USB-Sticks mit einem fest implementierten Kameramodul vorliegen, das die problemlose Ergänzung von Peripheriegeräten demonstrierte. Dieses Gerät nahmen wir nun zu einer Reihe von Meetings mit den Leuten von der Entwicklungsabteilung der BBC mit. Diejenigen von uns, die in den 80er-Jahren in Großbritannien aufgewachsen waren, hatten durch den BBC-Mikrocomputer und das ganze Ökosystem, das sich dazu entwickelt hatte – mit von der BBC produzierten Büchern, Magazinen und Fernsehsendungen –, eine Menge über die 8-Bit-Datenverarbeitung gelernt. Deshalb hoffte ich, das Unternehmen

könnte möglicherweise auch an der Weiterentwicklung des Raspberry Pi Interesse zeigen. Wie sich jedoch herausstellte, hat sich seit unserer Jugend einiges verändert: Aufgrund diverser Wettbewerbsgesetze in Großbritannien und der EU konnte sich das Unternehmen nicht in der Form mit einbringen, wie wir es uns gewünscht hatten. In einem allerletzten verzweifelten Versuch, doch noch *irgendetwas* mit ihnen auf die Beine zu stellen, ließen wir die Idee mit der Entwicklungsabteilung fallen und David (der mit den vielen Kontakten) organisierte im Mai 2011 ein Meeting mit Rory Cellan-Jones, einem erfahrenen Wissenschaftsjournalisten. Rory machte uns nicht viel Hoffnung auf eine Kooperation mit der BBC, fragte aber, ob er mit seinem Handy ein Video von dem kleinen Board-Prototyp machen und in seinen Blog stellen dürfe.

Schon am nächsten Morgen hatte Rorys Video wie ein Computervirus die Runde gemacht – und mir wurde klar, dass wir der Welt mehr oder weniger unbeabsichtigt versprochen hatten, einen 25-Dollar-Computer für jedermann zu bauen.

Während Rory von dannen zog, um einen Blogpost zu den Hintergründen der viralen Verbreitung eines Videos zu schreiben, war es für uns an der Zeit, ein paar wichtige Überlegungen anzustellen. Der ursprüngliche Prototyp von der Größe eines USB-Sticks wurde den Anforderungen nicht gerecht: Eine standardmäßig mit einer Kamera ausgestattete Ausführung des Rechners war viel zu teuer, um dem von uns vorgesehenen Kostenmodell entsprechen zu können. (Die Preishöhe von 25 Dollar ergab sich übrigens aus meiner Aussage gegenüber der BBC, dass der Raspberry Pi in etwa so viel kosten sollte wie ein gedrucktes Buch – ein grandioser Beleg dafür, dass ich absolut keine Ahnung hatte, wie viel Bücher heutzutage kosten.) Außerdem bot das winzige Prototypmodell nicht genug Platz für all die Ports, die wir brauchten, um das Gerät in der Form funktionsfähig zu machen, wie wir uns das vorstellten. Also brüteten wir ein Jahr lang über einer technischen Konzeptionierung des Boards, die es uns ermöglichte, einerseits die Kosten, so weit es ging, zu senken (das Erarbeiten von Kostenreduzierungen ist viel anstrengender, als man glauben mag), andererseits aber auch sämtliche Funktionen beizubehalten, die wir anbieten wollten. Und darüber hinaus sollte der Raspberry Pi für Käufer, die sich keine Peripheriegeräte leisten konnten, natürlich ebenfalls noch möglichst attraktiv sein.

Der Raspberry Pi sollte wie der ZX Spectrum der Achtzigerjahre am heimischen Fernsehgerät betrieben werden können, damit sich der Benutzer kein separates Display anschaffen musste. Aber nicht jeder besitzt einen HDMI-Fernseher, deshalb ergänzten wir einen Composite-Ausgang, so dass der Raspberry Pi auch an ältere Röhrenfernseher angeschlossen wer-

den konnte. Weil SD-Karten im Vergleich zu MicroSD-Karten preiswerter, aber genauso problemlos erhältlich sind, entschieden wir uns gegen MicroSD als Speichermedium, zumal diese nur fingernagelgroßen Karten für Kinderhände zu zerbrechlich sind und leicht verloren gehen können. Außerdem stellten wir diverse Überlegungen zur Stromversorgung an, die uns letztendlich zum Mikro-USB-Kabel führten, das sich in jüngster Vergangenheit EU-weit (und bald auch weltweit) zum Standard-Ladekabel für Handys gemausert hat, so dass es zunehmend allgegenwärtig und somit häufig bereits in den Haushalten vorhanden ist.

Obwohl der Verkaufsbeginn erst für Februar 2012 geplant war, zeichnete sich schon Ende 2011 ab, dass es nun Schlag auf Schlag ging und die Nachfrage größer war, als wir erwartet hatten. Die erste Markteinführung sollte zunächst einmal auf Entwickler abzielen, der Produktstart für den Bildungssektor sollte dagegen erst später im Jahr 2012 erfolgen. Wir konnten zwar auf einige sehr eifrige freiwillige Helfer zählen, waren darüber hinaus aber auch auf die Unterstützung der großen Linux-Community angewiesen, um einen Softwarebestand vorhalten zu können und die Kinderkrankheiten des Boards auszubügeln, bevor wir es im Bildungssegment anboten. Die Stiftung verfügte über genügend Kapital, um die Bauteile für 10.000 Raspberry Pis zu beschaffen, die dann innerhalb etwa eines Monats gefertigt werden konnten – und wir gingen davon aus, dass die Leute aus der Community in etwa in dieser Größenordnung an der ersten Board-Version interessiert wären. Allerdings war der Aufbau unserer Online-Community – Glück und Pech zugleich – so erfolgreich verlaufen, dass sich das Interesse an dem Gerät keineswegs nur auf Großbritannien oder den Bildungssektor beschränkte. Eine Stückzahl von »nur« 10.000 schien daher immer unrealistischer.

### Unsere Community

Die Raspberry-Pi-Community gehört zu den Errungenschaften, auf die wir besonders stolz sind. Der Startschuss hierfür fiel kurz nach der Veröffentlichung von Rorys Video im Mai 2011 mit einem sehr rudimentären Blog auf [www.raspberrypi.org](http://www.raspberrypi.org), wo wir bald darauf auch ein Forum bereitstellten. Dieses Forum hat inzwischen mehr als 20.000 Mitglieder – die mehr als 100.000 geistreiche und ausgefuchste Posts zum Thema Raspberry Pi beigetragen haben. Hier weiß immer irgendwer eine Antwort auf so ziemlich jede Frage, wie abstrus sie auch scheinen mag, zum Raspberry Pi oder zur Programmierung im Allgemeinen (wenn sie sich nicht in diesem Buch findet, dann in den Foren).

Eine meiner Aufgaben bei Raspberry Pi besteht unter anderem auch darin, Vorträge vor Hackergruppen, bei Computertagungen, vor Lehrerkollegien, Programmierer-Kollektiven und Ähnlichem zu halten, und es ist immer jemand im Publikum, der bereits über die Raspberry-Pi-Website mit mir oder meiner Frau Liz (die sich um die Community kümmert) Kontakt hatte – mit manchen von ihnen sind wir inzwischen richtig gut befreundet. Im Durchschnitt wird pro Tag jede Sekunde eine Anfrage auf der Raspberry-Pi-Website gestellt.

Inzwischen gibt es auch Hunderte von Fansites im Internet sowie ein von Community-Mitgliedern monatlich veröffentlichtes Fan-Magazin namens »The MagPi« (kostenloser Download unter [www.themagpi.com](http://www.themagpi.com)) mit Listings zum Abtippen, zahlreichen Artikeln, Projektanleitungen, Tutorials und vielem mehr. Durch solche in Magazinen und Büchern veröffentlichten Spiele-Listings zum Abtippen fand ich selbst seinerzeit zur Programmierung – meine früheste Programmiererfahrung mit dem BBC Micro war die Änderung des Listings für ein Hubschrauberspiel, das ich um Gegner und Pick-ups ergänzte.

Mindestens einmal am Tag bloggen wir auf [www.raspberrypi.org](http://www.raspberrypi.org) interessante Neuigkeiten zu dem Gerät. Schauen Sie doch einfach mal vorbei und machen Sie mit!

Auf unserer Mailingliste standen 100.000 Leute, die einen Raspberry Pi haben wollten – und sie alle bestellten ihn dann auch gleich am ersten Tag! Das stellte uns natürlich vor ein paar Probleme.

Zunächst einmal waren da die unvermeidlichen Papierschnittwunden, die man sich beim Verpacken und Versenden von 100.000 Minicomputern unweigerlich zuzieht – Tatsache war jedenfalls, dass wir nun mal absolut kein Geld hatten, um Leute anzuheuern, die uns diese Arbeit hätten abnehmen können. Ein Lager hatten wir auch nicht – nur Jacks Garage. Und wir konnten unmöglich das Geld auftreiben, um 100.000 Einheiten auf einmal herzustellen – eigentlich hatten wir gedacht, wir könnten alle paar Wochen Chargen von je 2.000 Stück bauen, was bei einer derartig hohen Nachfrage allerdings so lange gedauert hätte, dass das Gerät schon veraltet gewesen wäre, noch bevor wir alle Bestellungen abgearbeitet hätten. Die Herstellung und der Vertrieb waren also eindeutig Bereiche, die wir auslagern und jemandem überlassen mussten, der bereits über eine entsprechende Infrastruktur und das notwendige Kapital verfügte. Also wandten wir uns an element14 und RS Components, zwei weltweit operierende Vertriebe für elektronische Bauteile, und vereinbarten, dass sie die Herstellung und den

weltweiten Vertrieb übernehmen, damit wir uns auf die Entwicklung und die gemeinnützigen Ziele der Raspberry Pi Foundation konzentrieren konnten.

Die Nachfrage war am ersten Tag immer noch so groß, dass sowohl die Website von RS als auch die von element14 für mehrere Stunden zusammenbrachen – bei element14 gingen zeitweise sieben Bestellungen pro Sekunde ein. Und bei Google wurden am 29. Februar einige Stunden lang weltweit mehr Suchanfragen nach »Raspberry Pi« gestellt als nach »Lady Gaga«. Dieser User Guide wurde Anfang Juni 2012 geschrieben. Allein in den drei Monaten, seit wir unsere Geschäftstätigkeit aufgenommen haben, sind die Bestellungen auf über eine halbe Million Einheiten angestiegen, obwohl wir immer noch an einem Punkt sind, an dem beide Unternehmen keine über einen einzigen Raspberry Pi hinausgehende Abgabemenge zulassen (weil sie zunächst einmal versuchen müssen, ihre Auftragsüberhänge abzarbeiten, bevor Mehrfachbestellungen angenommen werden können). Wäre es nach unserer ursprünglichen Planung gegangen, hätten wir zu diesem Zeitpunkt gerade mal vielleicht 100 Geräte oder so für die Tage der offenen Tür der Universität fertig gehabt – und das wär's gewesen.

Nichts treibt den Blutdruck schneller in die Höhe als mehr oder weniger zufällig der Betreiber einer großen Computerfirma zu werden!

## Was kann man denn nun mit dem Raspberry Pi anfangen?

In diesem Buch werden zahlreiche Anwendungsmöglichkeiten Ihres Raspberry Pi beschrieben – von der Hardwaresteuerung mittels Python über die Nutzung als Mediacenter bis hin zur Spieleprogrammierung in Scratch. Das Schöne am Raspberry Pi ist, dass es sich um einen winzigen Allzweck-Computer handelt (der vielleicht ein bisschen langsamer als manche Ihrer gewohnten Desktop-Anwendungen ist, dafür aber andere Dinge viel besser beherrscht als ein herkömmlicher PC), so dass Sie alles damit machen können, was auch mit einem normalen Computer möglich ist. Außerdem bietet der Raspberry Pi auch leistungsstarke Multimedia- und 3D-Grafikfunktionen und hat damit das Potenzial, als Spieleplattform genutzt zu werden. Insofern sind wir sehr zuversichtlich, dass die Benutzer bald Spiele dafür schreiben werden.

Unserer Meinung nach wird das »Physical Computing« – also das Bauen von Systemen unter Einbeziehung von Sensoren, Motoren, Lämpchen und Mikrocontrollern – häufig zugunsten reiner Softwareprojekte vernachlässigt.

sigt. Und das ist sehr schade, denn »Physical Computing« macht *unheimlichen Spaß*. Was den gegenwärtigen Trend der Computeraktivitäten von Jugendlichen angeht, bewegt sich alles in Richtung »Physical Computing«. Die LOGO-Turtles, die das Physical Computing repräsentierten, als wir Kinder waren, kämpfen heute gegen Roboter, Quadrocopter oder dienen als Eltern-Detektoren für die Schlafzimmertür – und wir lieben es. Aber die fehlende Allzweckeingabe/-ausgabe (GPIO) an herkömmlichen PCs ist für viele, die sich an Robotik-Projekten versuchen wollen, ein echtes Handicap. Der Raspberry Pi dagegen stellt einen GPIO-Port zur Verfügung, so dass man sofort mit der Arbeit loslegen kann.

Ich bin immer wieder überrascht, welche Ideen die Community zutage fördert, an die ich in tausend Jahren nicht gedacht hätte: das australische Meteorverfolgungs-Schulprojekt, die Boreatton Scouts in Großbritannien und ihr Roboter, der über ein elektro-enzephalografisches Headseat gesteuert wird (der weltweit erste Roboter, der durch Scouting-Hirnströme gesteuert wird), die Familie, die einen roboterartigen Staubsauger baut. Und ich bin ein echter Traumtänzer, deshalb kriege ich eine Gänsehaut, wenn ich von Leuten lese, die ihre Raspberry Pis mittels Raketenantrieben und Ballons in die Atmosphäre katapultieren.

Ein Erfolg wäre für uns, wenn weitere 1.000 Leute pro Jahr an den britischen Universitäten Informatik studieren würden. Das wäre nicht nur für unser Land, die Hard- und Softwareindustrie und die Wirtschaft von Vorteil, sondern vor allem für jeden einzelnen dieser 1.000 Leute, die – so hoffe ich – eine ganze Welt voller Möglichkeiten und jeder Menge Spaß entdecken würden. Als Kind oder Jugendlicher einen Roboter zu bauen, kann die Fantasie jenseits aller Vorstellungskraft beflügeln – und ich weiß, wovon ich rede, denn mir ist es so ergangen!

– Eben Upton

# Teil I

## Inbetriebnahme des Boards

### Kapitel 1

## Gestatten: Der Raspberry Pi

Ihr Raspberry-Pi-Board ist ein kleines Wunderwerk, das auf einer Gehäusegeometrie von der Größe einer Kreditkarte eine enorme Rechenleistung bietet. Mit diesem Gerät lassen sich erstaunliche Dinge vollbringen – allerdings sollten Sie zunächst über einige Besonderheiten Bescheid wissen, bevor Sie sich ans Werk begeben.

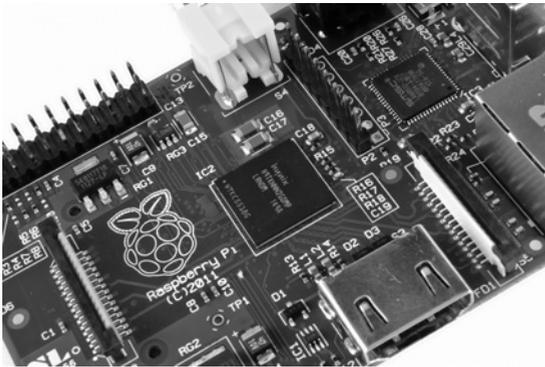
#### Tipp

Wenn Sie dennoch sofort loslegen möchten, können Sie die nächsten Seiten auch überspringen und direkt bei Abschnitt 1.3, »Erste Schritte mit dem Raspberry Pi«, weiterlesen, wo Sie erfahren, wie Sie ein Display, eine Tastatur und eine Maus an Ihren Raspberry Pi anschließen können.

## 1.1 ARM vs. X86

Das Herzstück des Raspberry-Pi-Systems bildet ein Broadcom-BCM2835-SoC-(System-on-Chip)-Multimediaprozessor. In diesem Prozessor, der sich unterhalb des 256-MB-Speichermoduls in der Mitte des Boards befindet (siehe Abbildung 1.1), ist der überwiegende Teil der Systemkomponenten bereits integriert, einschließlich CPU, Grafikprozessor sowie Audio- und Kommunikationshardware.

Der BCM2835 unterscheidet sich aber nicht nur durch sein SoC-Ein-Chip-Design von den in herkömmlichen Desktop-PCs oder Notebooks verbauten Prozessoren. Er bedient sich auch einer anderen Befehlssatzarchitektur, der so genannten ARM-Architektur.



**Abb. 1.1:** Der unterhalb eines Hynix-Speichermoduls angebrachte BCM2835 SoC

Die ARM-Architektur wurde in den späten 1980er-Jahren von dem britischen Unternehmen Acorn Computers entwickelt und ist in der Desktop-Welt nur relativ selten anzutreffen. Bei Mobilgeräten sieht das allerdings ganz anders aus: Im Inneren Ihres Handys befindet sich mit an Sicherheit grenzender Wahrscheinlichkeit wenigstens ein ARM-basierter Prozessorkern. Wegen seines einfachen *RISC (Reduced Instruction Set Computer)*-Designs und seines geringen Stromverbrauchs ist er im Vergleich zu den geradezu stromfressenden, auf *CISC (Complex Instruction Set Computer)*-Befehlssätzen aufbauenden Desktop-Prozessoren erste Wahl.

Dank des ARM-basierten BCM2835 kommt der Raspberry Pi mit einer Stromstärke von 5 Volt und 1 Ampere aus, die über den Onboard-Mikro-USB-Port gewährleistet wird. Aus diesem Grund finden sich auch keine Kühlkörper in dem Gerät: Der minimale Stromverbrauch des Prozessors

sorgt für eine verschwindend geringe Hitzeentwicklung, selbst während der Ausführung komplizierter Rechenprozesse.

Andererseits heißt das aber auch, dass der Raspberry Pi nicht mit traditioneller PC-Software kompatibel ist. Die meiste Software für PCs und Notebooks wird mit Blick auf die x86-Mikroprozessor-Architektur entwickelt, wie sie in den Prozessoren der Hersteller AMD, Intel und VIA verwendet wird, und ist daher auf dem ARM-basierten Raspberry Pi nicht lauffähig.

Der BCM2835 gehört der Prozessorgeneration ARM11 an, die wiederum auf der Befehlssatzarchitektur ARMv6 basiert. Das Interessante daran ist: ARMv6 ist eine kompakte und leistungsfähige Architektur, die jedoch in der fortschrittlicheren ARMv7-Architektur, wie sie in der ARM-Cortex-Prozessorfamilie zum Einsatz kommt, einen Rivalen hat. Ebenso wie die für x86 geschriebenen Programme ist auch die für ARMv7 entwickelte Software leider nicht mit dem BCM2835 des Raspberry Pi kompatibel – wenngleich die Entwickler sie normalerweise passend umschreiben können.

Das bedeutet aber noch lange nicht, dass Sie in Ihrer Auswahl eingeschränkt wären. Wie Sie im weiteren Verlauf dieses Buches feststellen werden, gibt es jede Menge für den ARMv6-Befehlssatz geeignete Software – und mit zunehmender Popularität des Raspberry Pi wird auch die Vielfalt der verfügbaren Software ansteigen. Außerdem werden Sie im Rahmen dieses User Guides erfahren, wie Sie selbst ohne Programmierkenntnisse eigene Software für den Raspberry Pi schreiben können.

## 1.2 Windows/Apple OS X vs. Linux

Ein weiterer wichtiger Unterschied zwischen dem Raspberry Pi und Ihrem PC beziehungsweise Notebook ist, abgesehen von der Größe und dem Kostenfaktor, das verwendete Betriebssystem – also die Software, die es Ihnen gestattet, Ihren Computer überhaupt in Betrieb zu nehmen.

Auf den meisten modernen PCs und Notebooks findet sich eins der zwei marktführenden Betriebssysteme Microsoft Windows oder Apple OS X. Beide Plattformen sind *Closed Source*, das heißt, sie werden unter Anwendung unternehmenseigener Verfahren und Techniken entwickelt, die einer strengen Geheimhaltung unterliegen.

Bei diesen Betriebssystemen handelt es sich um unfreie, proprietäre Software, deren *Quelltext* vom Hersteller unter Verschluss gehalten beziehungsweise nicht öffentlich zugänglich gemacht wird. Dieser in einer Programmiersprache geschriebene Programmcode sagt dem System, was es zu tun hat, und wird daher sorgsam gehütet. Anwender können die fer-

tige Software zwar erwerben und nutzen, bekommen aber niemals zu sehen, woraus sie tatsächlich besteht beziehungsweise wie sie im Einzelnen aufgebaut ist.

Im Falle des Raspberry Pi kommt dagegen ein Betriebssystem namens GNU/Linux – im Folgenden der Einfachheit halber nur Linux genannt – zum Einsatz. Anders als Windows und OS X ist Linux *Open-Source-Software*, das heißt, man kann sich den Quelltext des gesamten Betriebssystems herunterladen und ihn beliebig ändern und weiterverbreiten. Der Code steht dem Anwender also vollumfänglich zur Verfügung und sämtliche daran vorgenommenen Modifikationen sind öffentlich zugänglich. Dieser Grundsatz der Open-Source-Entwicklung ermöglichte eine schnelle Anpassung der Linux-Software für den Raspberry Pi – ein Vorgang, der als *Portierung* bezeichnet wird. Zum Zeitpunkt der Drucklegung dieses User Guides wurden bereits mehrere Linux-Versionen – so genannte *Distributionen* – auf den BCM2835-Prozessor des Raspberry Pi portiert, darunter auch Debian, Fedora Remix und Arch Linux.

Auch wenn die diversen Distributionen üblicherweise auf unterschiedliche Belange abzielen, so weisen sie doch eine Gemeinsamkeit auf: Sie alle sind Open Source. Außerdem sind sie im Großen und Ganzen allesamt untereinander kompatibel, so dass auf einem Debian-System geschriebene Software problemlos auch unter Arch Linux läuft und umgekehrt.

Natürlich ist Linux nicht exklusiv nur auf dem Raspberry Pi einsetzbar. Es gibt Hunderte verschiedene Distributionen für PCs, Notebooks und sogar Mobilgeräte – und selbst Googles populäre Android-Plattform wurde auf der Grundlage eines Linux-Kerns entwickelt. Wenn Ihnen die Arbeit mit Linux auf dem Raspberry Pi gefällt, sollten Sie vielleicht in Erwägung ziehen, es auch auf anderen Datenverarbeitungsgeräten einzusetzen, die Sie sonst noch benutzen. Es lässt sich ohne Probleme parallel zu Ihrem aktuellen Betriebssystem verwenden, so dass Sie in den Genuss der Vorzüge beider Umgebungen kommen und darüber hinaus auch dann mit einem vertrauten System arbeiten können, wenn Sie Ihren Raspberry Pi gerade einmal nicht zur Hand haben.

Ähnlich wie schon bei den ARM- und x86-Architekturen beschrieben, besteht allerdings auch hier in der Praxis ein wesentlicher Unterschied zwischen Windows beziehungsweise OS X und Linux: Software, die für Windows oder OS X geschrieben wurde, ist unter Linux nicht lauffähig. Glücklicherweise ist für den überwiegenden Teil der gängigen Softwareprodukte jedoch eine Fülle von kompatiblen Alternativen erhältlich – und was noch besser ist: Die meisten davon sind frei verfügbar und genauso Open Source wie das Betriebssystem selbst.

## 1.3 Erste Schritte mit dem Raspberry Pi

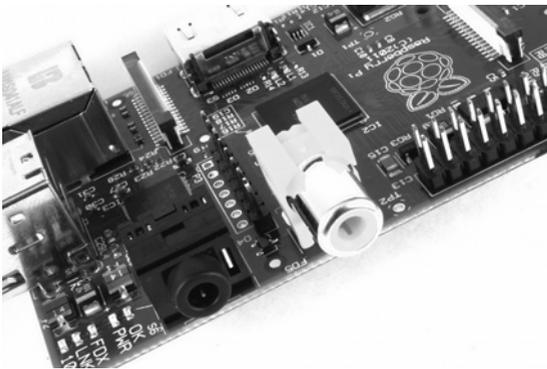
Nachdem Sie nun einen grundlegenden Überblick darüber gewonnen haben, inwiefern sich der Raspberry Pi von anderen Datenverarbeitungsgeräten unterscheidet, ist es an der Zeit, richtig loszulegen. Sollten Sie Ihren Raspberry Pi gerade erst bekommen haben, nehmen Sie ihn zunächst einmal aus der Antistatik-Schutzhülle heraus und platzieren Sie ihn auf einem ebenen, nicht leitenden Untergrund, bevor Sie den Ausführungen in diesem Kapitel folgen.

### 1.3.1 Bildausgabegerät anschließen

Um Ihren Raspberry Pi benutzen zu können, müssen Sie ihn als Erstes an ein Anzeigegerät anschließen. Zu diesem Zweck finden Sie drei verschiedene Videoausgänge an dem Gerät vor: *Composite Video*, *HDMI (High Definition Multimedia Interface)* und *DSI (Display Serial Interface)*. Die Composite- und HDMI-Anschlüsse sind, wie Sie im Folgenden noch sehen werden, unmittelbar einsatzbereit, der DSI-Anschluss erfordert dagegen spezielle zusätzliche Hardware.

#### Composite Video

Über den Composite-Video-Port, die gelb-silberfarbene *RCA-Cinchbuchse* (siehe Abbildung 1.2) an der Oberseite des Raspberry Pi, können auch ältere, inzwischen aus der Mode gekommene Anzeigegeräte, wie z. B. CRT-Röhrenmonitore, angeschlossen werden. Dabei werden die Farbsignale – Rot, Grün und Blau – des Bildes gemischt und über eine einzelne Leitung an das Bildausgabegerät gesendet.



**Abb. 1.2:** Die gelb-silberfarbene Cinchbuchse für den Composite-Video-Ausgang