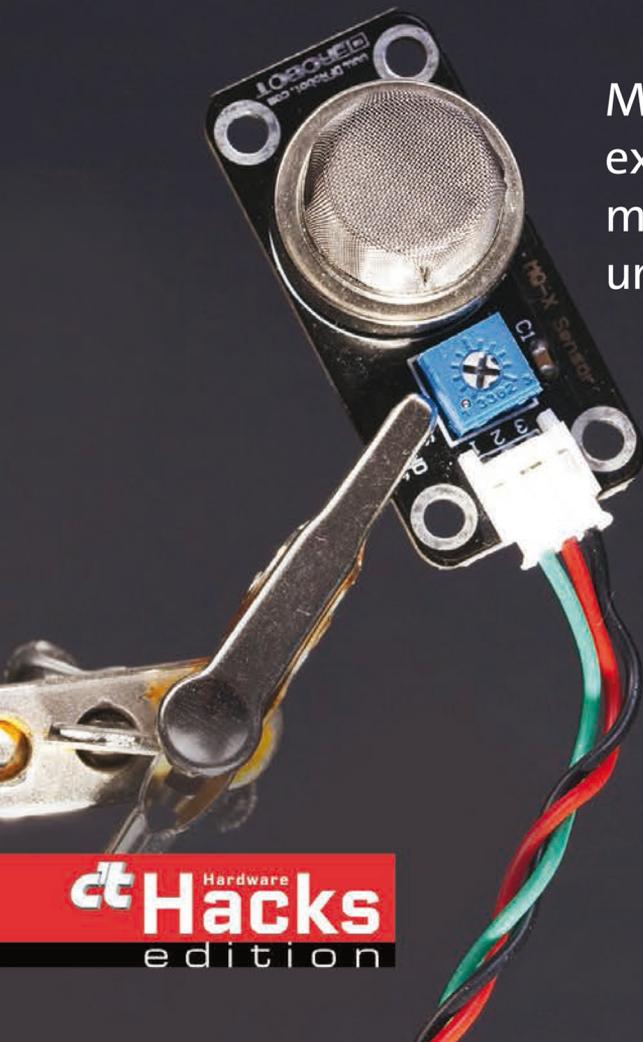


Kimmo Karvinen · Tero Karvinen · Ville Valtokari

Sensoren

Messen und
experimentieren
mit **Arduino**
und **Raspberry Pi**



Hardware
ct Hacks
edition



dpunkt.verlag

Tero Karvinen lehrt Linux und Embedded-Systeme in Haaga – Helia University of Applied Sciences, wo seine Arbeit auch Lehrplanentwicklung und Forschung im Bereich der drahtlosen Vernetzung umfasst.

Kimmo Karvinen arbeitet als CEO in einem führenden Unternehmen in der AV- Automatisierung in Finnland. Davor war er u.a. CTO bei einem Hardwarehersteller, der sich auf intelligente Gebäudetechnik spezialisiert hatte.

Ville Valtokari arbeitet als Chefprogrammierer bei einem Hersteller von Automatisierungs-Hardware. Zuvor gestaltete und programmierte er modernste AV-Systeme.

Tero Karvinen · Kimmo Karvinen · Ville Valtokari

Sensoren – messen und experimentieren mit Arduino und Raspberry Pi



dpunkt.verlag

Tero Karvinen
Kimmo Karvinen
Ville Valtokari

Lektorat: Dr. Michael Barabas
Copy-Editing: Ursula Zimpfer, Herrenberg
Herstellung: Frank Heidt
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Umschlagfoto: Kimmo Karvinen
Fotografien: Kimmo Karvinen
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

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
Buch 978-3-86490-160-7
PDF 978-3-86491-601-4
ePub 978-3-86491-602-1

1. Auflage 2015
Copyright © 2015 dpunkt.verlag GmbH
Wiebinger Weg 17
69123 Heidelberg

Authorized German translation of the English edition of Make: Sensors, ISBN/EAN:978-1-4493-6810-4
© Tero Karvinen, Kimmo Karvinen, Ville Valtokari, published by Maker Media, Inc. This translation is
published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to sell the same.

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung
der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags
urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung
oder die Verwendung in elektronischen Systemen.

Make., Maker Shed und Maker Fair sind eingetragene Warenzeichen der Maker Media, Inc.
Das Maker-Media-Logo ist ein Warenzeichen der Maker Media, Inc.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie
Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-,
marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor
noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der
Verwendung dieses Buches stehen.

5 4 3 2 1 0

Inhaltsverzeichnis

	Vorwort	1
1	Raspberry Pi	11
1.1	Raspberry Pi: Von null zum ersten Start	12
1.1.1	NOOBS*.zip entpacken	13
1.1.2	Kabel anschließen	13
1.1.3	Raspbian hochfahren und installieren	14
1.1.4	Fehlersuche bei der Raspberry Pi-Installation	16
1.2	Willkommen bei Linux	18
1.2.1	Die allgegenwärtige Kommandozeile	18
1.2.2	Schauen Sie sich um	19
1.2.3	Textdateien für die Konfiguration	20
1.2.4	sudo mach mir ein Butterbrot!	20
1.3	Elektronische Bauteile an die Pins des Raspberry Pi anschließen	22
1.3.1	Hallo GPIO, lass eine LED blinken	23
1.3.2	Die Schaltung aufbauen	24
1.3.3	Zwei Nummerierungssysteme: Zweck und Ort	25
1.3.4	GPIO-Pins über die Kommandozeile steuern	27
1.3.5	Dateien ohne Editor bearbeiten	27
1.3.6	Die LED aufleuchten lassen	28
1.3.7	Fehlerbehebung	29
1.4	GPIO-Steuerung ohne Root-Berechtigungen	31
1.4.1	Fehlersuche bei der GPIO-Steuerung	33
1.5	GPIO in Python	34
1.5.1	Hello Python	34
1.6	Wie geht es weiter?	37

2	Arduino	39
2.1	Grundinstallation des Arduino	40
2.1.1	Ubuntu Linux	40
2.1.2	Windows 7 und Windows 8	41
2.1.3	OS X	42
2.1.4	Hello World	42
2.1.5	Der Aufbau eines Arduino-Programms	43
2.1.6	Einfach und vielseitig dank Shields	44
3	Entfernung	47
3.1	Experiment: Abstände mit Ultraschall messen (Ping)	48
3.1.1	Code und Schaltung für den Ping am Arduino	49
3.1.2	Code und Schaltung für den Ping am Raspberry Pi	51
3.2	Ultraschallsensor HC-SR04	54
3.2.1	Code und Schaltung für den HC-SR04 am Arduino	54
3.2.2	Code und Schaltung für den HC-SR04 am Raspberry Pi	56
3.2.3	Echoberechnungen	57
3.2.4	Praxisexperiment: Unsichtbare Objekte	59
3.3	Experiment: Hindernisse mit Infrarot erkennen (IR-Abstandssensor)	60
3.3.1	Code und Schaltung für den IR-Sensor am Arduino	61
3.3.2	Code und Schaltung für den IR-Sensor am Raspberry Pi	63
3.4	Praxisexperiment: Infrarotlicht sichtbar machen	64
3.5	Experiment: Bewegungen mit Infrarot verfolgen (IR-Facettenauge)	65
3.5.1	Code und Schaltung für das Facettenauge am Arduino	67
3.5.2	Code und Schaltung für das Facettenauge am Raspberry Pi	69
3.5.3	Bibliothek spidev installieren	72
3.5.4	Alternative Schaltungen für den Raspberry Pi	73
3.6	Testprojekt: Haltungswarner (Arduino)	74
3.6.1	Lernziele	74
3.6.2	Piezo-Summer	75
3.6.3	Alarm!	77
3.6.4	Piezo-Summer und IR-Sensor kombinieren	78
3.6.5	Eine elegante Verpackung für das Projekt	80

4	Rauch und Gas	83
4.1	Experiment: Rauchmelder (analoger Gassensor)	84
4.1.1	Code und Schaltung für den MQ-2 am Arduino	85
4.1.2	Code und Schaltung für den MQ-2 am Raspberry Pi	86
4.1.3	Praxisexperiment: Rauch steigt nach oben	88
4.1.4	Experiment: Alkotest (Alkoholsensor MQ-303A)	89
4.1.5	Praxisexperiment: Nüchtern bleiben beim Experimentieren	93
4.2	Testprojekt: Rauchalarm per E-Mail senden	93
4.2.1	Lernziele	94
4.2.2	Python für E-Mails und Social Media	94
4.2.3	Das Projekt bauen	94
4.2.4	Wie funktioniert E-Mail?	95
4.2.5	Kann der Arduino E-Mails senden? – Nicht ohne Weiteres!	96
4.2.6	Code für den Raspberry Pi	96
4.2.7	Gehäuse	99
5	Berührung	105
5.1	Experiment: Drucktasten	106
5.1.1	Pullup-Widerstand	107
5.1.2	Code und Schaltung am Arduino	107
5.1.3	Code und Schaltung am Raspberry Pi	109
5.2	Experiment: Mikroschalter	110
5.2.1	Code und Schaltung für den Mikroschalter am Arduino .	111
5.2.2	Code und Schaltung für den Mikroschalter am Raspberry Pi	112
5.3	Experiment: Potenziometer (regelbarer Widerstand, Poti)	114
5.3.1	Code und Schaltung für das Potenziometer am Arduino .	115
5.3.2	Code und Schaltung für das Potenziometer am Raspberry Pi	116
5.4	Experiment: Berührungsfreier Berührungssensor (kapazitiver Berührungssensor QT113)	119
5.4.1	Code und Schaltung für den QT113 am Arduino	120
5.4.2	Code und Schaltung für den QT113 am Raspberry Pi . . .	121
5.5	Praxisexperiment: Berührungen durch eine hölzerne Oberfläche hindurch erkennen	122
5.6	Experiment: Druck messen (FlexiForce)	123
5.6.1	Code und Schaltung für den FlexiForce am Arduino	123
5.6.2	Code und Schaltung für den FlexiForce am Raspberry Pi .	124

5.7	Experiment: Berührungssensor im Eigenbau	126
5.7.1	Code und Schaltung für den kapazitiven Sensor am Arduino	127
5.7.2	Code und Schaltung für den kapazitiven Sensor am Raspberry Pi	128
5.8	Testprojekt: Geisterglocke	129
5.8.1	Lernziele	130
5.8.2	Servomotoren	130
5.8.3	Code und Schaltung für die Geisterglocke am Arduino . . .	134
5.8.4	Servo an der Glocke montieren	137
6	Bewegung	139
6.1	Experiment: Wo ist oben? (Tilt-Schalter)	139
6.1.1	Code und Schaltung für den Kippsensor am Arduino	140
6.1.2	Code und Schaltung für den Kippsensor am Raspberry Pi .	141
6.2	Experiment: Good Vibrations mit Interrupts (digitaler Vibrationssensor)	142
6.2.1	Code und Schaltung für den Vibrationssensor am Arduino	142
6.2.2	Code und Schaltung für den Vibrationssensor am Raspberry Pi	144
6.3	Experiment: Am Regler drehen	145
6.3.1	Code und Schaltung für den Drehgeber am Arduino	146
6.3.2	Code und Schaltung für den Drehgeber am Raspberry Pi .	148
6.4	Experiment: Analoger Zweiachs-Daumen-Joystick	149
6.4.1	Code und Schaltung für den Joystick am Arduino	150
6.4.2	Code und Schaltung für den Joystick am Raspberry Pi . . .	152
6.5	Praxisexperiment: Teile eines Xbox-Controllers wiederverwenden .	154
6.6	Experiment: Alarmanlage (passiver IR-Sensor)	155
6.6.1	Code und Schaltung für die Alarmanlage am Arduino . . .	156
6.6.2	Code und Schaltung für die Alarmanlage am Raspberry Pi	158
6.6.3	Praxisexperiment: Eine Alarmanlage überlisten	159
6.7	Testprojekt: Pong	162
6.7.1	Lernziele	163
6.7.2	Tipps für ein ansprechendes Gehäuse	167
6.7.3	Das Spiel beim Hochfahren des Raspberry Pi automatisch starten	170

7	Licht	175
7.1	Experiment: Feuer erkennen (Flammensensor)	175
7.1.1	Code und Schaltung für den Flammensensor am Arduino	176
7.1.2	Code und Schaltung für den Flammensensor am Raspberry Pi	177
7.2	Praxisexperiment: Genauigkeit bei der Erkennung von Flammen . .	179
7.3	Experiment: Siehst du dieses Licht? (Fotowiderstand, LDR)	180
7.3.1	Code und Schaltung für den Fotowiderstand am Arduino	181
7.3.2	Code und Schaltung für den Fotowiderstand am Raspberry Pi	182
7.4	Praxisexperiment: Die Richtung erkennen	184
7.5	Experiment: Linien verfolgen	185
7.5.1	Code und Schaltung für den Liniensensor am Arduino . . .	186
7.5.2	Code und Schaltung für den Liniensensor am Raspberry Pi	187
7.6	Praxisexperiment: Schwarz ist weiß	189
7.7	Experiment: Alle Farben des Regenbogens	191
7.7.1	Code und Schaltung für den Farbsensor am Arduino	191
7.7.2	Code und Schaltung für den Farbsensor am Raspberry Pi	194
7.8	Testprojekt: Chamäleonkuppel	197
7.8.1	Lernziele	197
7.8.2	RGB-LEDs	197
7.8.3	Code und Schaltung für RGB-LEDs am Arduino	199
7.8.4	Gleitender Mittelwert	201
7.8.5	Beliebige Farben mit RGB-LEDs erzeugen	202
7.8.6	Eingänge in Ausgänge umwandeln	203
7.8.7	Code für eine Kombination aus RGB-LED und Farbsensor	205
7.8.8	Tipps zum Bau der Kuppel	209
8	Beschleunigung	213
8.1	Beschleunigung und Winkelgeschwindigkeit	214
8.2	Experiment: Beschleunigung mit dem MX2125 messen	214
8.2.1	Impulslängen des MX2125 entschlüsseln	215
8.2.2	Code und Schaltung für den Beschleunigungsmesser am Arduino	217
8.2.3	Code und Schaltung für den Beschleunigungsmesser am Raspberry Pi	219
8.3	Experiment: Beschleunigungsmesser und Gyroskop	220
8.3.1	Code und Schaltung für den MPU 6050 am Arduino	222
8.3.2	Code und Schaltung für den MPU 6050 am Raspberry Pi	227

8.3.3	Hexadezimale, binäre und andere Zahlen	232
8.3.4	Bitweise Operationen	235
8.4	Experiment: Den Wii Nunchuk zweckentfremden (über I2C)	239
8.4.1	Code und Schaltung für den Nunchuk am Arduino	240
8.4.2	Code und Schaltung für den Nunchuk am Raspberry Pi	244
8.5	Testprojekt: Steuerung einer Roboterhand mit dem Wii Nunchuk	246
8.5.1	Lernziele	247
8.5.2	Die Mechanik der Roboterhand	251
9	Identität	253
9.1	Tastenfelder	254
9.1.1	Code und Schaltung für das Tastenfeld am Arduino	255
9.1.2	Code und Schaltung für das Tastenfeld am Raspberry Pi	257
9.2	Praxisexperiment: Fingerabdrücke erkennen	260
9.3	Fingerabdruckscanner GT-511C3	261
9.3.1	Code und Schaltung für den Fingerabdrucksensor am Arduino Mega	263
9.3.2	Code und Schaltung für den Fingerabdrucksensor am Raspberry Pi	270
9.4	RFID mit dem ELB149C5M	276
9.4.1	Code und Schaltung für das RFID-Lesegerät am Arduino Mega	278
9.4.2	Code und Schaltung für das RFID-Lesegerät am Raspberry Pi	280
9.5	Testprojekt: Futuristische Schatztruhe	284
9.5.1	Lernziele	284
9.5.2	Die Funktionsweise der Truhe	284
9.5.3	Die Truhe	285
9.5.4	Code und Schaltung für die futuristische Schatztruhe am Arduino	287
9.6	Menschen und Objekte erkennen	293
10	Elektrizität und Magnetismus	295
10.1	Experiment: Spannung und Stromstärke	295
10.1.1	Code und Schaltung für den AttoPilot am Arduino	297
10.1.2	Code und Schaltung für den AttoPilot am Raspberry Pi	299

10.2	Experiment: Ist es magnetisch?	300
10.2.1	Code und Schaltung für den Hall-Effekt-Sensor am Arduino	301
10.2.2	Code und Schaltung für den Hall-Effekt-Sensor am Raspberry Pi	302
10.3	Experiment: Magnetischer Norden mit dem Kompass/ Beschleunigungsmesser LSM303 bestimmen	304
10.3.1	Den Sensor kalibrieren	305
10.3.2	Code und Schaltung für den LSM303 am Arduino	306
10.3.3	Code und Schaltung für den LSM303 am Raspberry Pi	311
10.3.4	Das Protokoll für den LSM303	315
10.3.5	Berechnungen für die Kompassrichtung	316
10.4	Experiment: Hall-Schalter	318
10.4.1	Code und Schaltung für den Hall-Schalter am Arduino	319
10.4.2	Code und Schaltung für den Hall-Schalter am Raspberry Pi	320
10.5	Testprojekt: Webüberwachung von Solarzellen	321
10.5.1	Lernziele	321
10.5.2	Die Solarzellen anschließen	322
10.5.3	Der Raspberry Pi als Webserver	325
10.5.4	Die IP-Adresse ermitteln	326
10.5.5	Eine Homepage auf dem Raspberry Pi einrichten	326
10.5.6	Code und Schaltung für die Solarzellenüberwachung am Raspberry Pi	327
10.5.7	Zeitliche Planung von Aufgaben mit cron	329
10.6	Wie geht es weiter?	330
11	Schall	333
11.1	Experiment: Stimmen hören/Lautstärke	333
11.1.1	Code und Schaltung für das Mikrofon am Arduino	334
11.1.2	Code und Schaltung für das Mikrofon am Raspberry Pi	335
11.2	Praxisexperiment: Eine Stecknadel fallen hören	336
11.3	Testprojekt: Töne über HDMI sichtbar machen	337
11.3.1	Lernziele	337
11.3.2	Den seriellen Port des Raspberry Pi aktivieren	338
11.3.3	Code und Schaltung für die Visualisierung am Raspberry Pi	338
11.3.4	Schnelle Fourier-Transformation	341
11.4	Wie geht es weiter?	343

12	Wetter und Klima	345
12.1	Experiment: Heiß hier drin, oder?	345
12.1.1	Code und Schaltung für den LM35 am Arduino	346
12.1.2	Code und Schaltung für den LM35 am Raspberry Pi	347
12.2	Praxisexperiment: Temperaturwechsel	348
12.3	Experiment: Wie feucht ist es?	349
12.3.1	Wie feucht ist Ihr Atem?	350
12.3.2	Code und Schaltung für den DHT11 am Arduino	351
12.3.3	Code und Schaltung für den DHT11 am Raspberry Pi	353
12.3.4	Kommunikation vom Raspberry Pi zum Arduino	355
12.4	Luftdrucksensor GY-65	357
12.4.1	Code und Schaltung für den GY-65 am Arduino	358
12.4.2	Arduino-Bibliotheken verwenden	359
12.4.3	Die Arduino-Bibliothek für den GY-65	360
12.4.4	Code und Schaltung für den GY-65 am Raspberry Pi	365
12.5	Experiment: Brauchen Ihre Pflanzen Wasser? (Sensor für Bodenfeuchtigkeit)	368
12.5.1	Code und Schaltung für den Bodenfeuchtigkeitssensor am Arduino	369
12.5.2	Code und Schaltung für den Bodenfeuchtigkeitssensor am Raspberry Pi	370
12.6	Testprojekt: Wettervorhersage auf E-Paper	371
12.6.1	Lernziele	372
12.6.2	Code und Schaltung für die Wetterstation am Arduino	372
12.7	Praxisexperiment: Schau mal, ganz ohne Stromanschluss!	380
12.8	Bilder in Headerdateien speichern	380
12.8.1	Programm zur Umwandlung von BMP-Bildern in C-Dateien	382
12.8.2	Tipps für das Gehäuse	383
A	Linux-Befehle für den Raspberry Pi	387
	Index	391

Vorwort

Willkommen bei »Sensoren – messen und experimentieren mit Arduino und Raspberry Pi«. Sie werden schon bald mit dem Bau von Geräten beginnen können, die in der Lage sind, alle möglichen Dinge wahrzunehmen – von gefährlichen Gasen bis zu Beschleunigungen. In diesem Buch erfahren Sie, wie Sie Sensoren einsetzen, um die Phänomene der physischen Welt zu messen, und wie Sie die Ergebnisse als Zahlenwerte darstellen und aufgrund solcher Werte automatisch Aktionen auslösen lassen können.

Sensoren können beispielsweise Wärme, Druck, Beschleunigung und Licht messen und Werte wie 22 °C, 1015 mbar, eine Beschleunigung von 2,3 g oder das Vorhandensein von Licht melden. (Beim Licht haben wir hier keine numerische Quantität angegeben, sondern einen booleschen Wert, also ja oder nein. Beispiele dafür werden Sie später noch kennenlernen.)

Eine Mikrocontroller-Platine stellt das Gehirn des Roboters, des Systems oder Geräts dar, das Sie bauen. Die Software, die auf dem Mikrocontroller ausgeführt werden soll, schreiben Sie selbst. In diesem Buch arbeiten wir mit den beiden beliebtesten Platinen, dem Arduino und dem Raspberry Pi. Bei beiden ist es einfach, Software zur Steuerung der Elektronik zu schreiben.

Es geht um Ihre eigenen Ideen

Wenn Sie schnell einige Grundlagen der Elektronik lernen möchten, um dann Ihre eigenen Roboter, Geräte oder Projekte zu gestalten, dann sind Sie hier richtig. In diesem Buch lernen Sie, wie Sie Ihre Ideen rasch umsetzen können.

Theorien, Fertigkeiten und Grundlagenwissen sind sehr nützlich – solange sie Ihre Kreativität unterstützen. Spielen Sie mit Ihren eigenen Ideen herum und trauen Sie sich, die Ergebnisse im Web zu veröffentlichen.

In jedem Kapitel finden Sie ein Miniprojekt, das Ihnen zeigt, wie Sie verschiedene Technologien kombinieren können. Beispielsweise werden Sie im Rahmen dieser Projekte einen Holzkasten mit einem Schloss versehen, das Sie mit Ihrem Fingerabdruck öffnen können, und eine Chamäleonkuppel bauen, die ihre Farbe

wechselt. Diese Projekte machen Spaß, bilden aber auch gute Ausgangspunkte für spätere eigene Erfindungen.

Die Fähigkeiten, die Sie sich mithilfe des Arduino aneignen, lassen sich leicht auf ernsthafte Projekte übertragen. Beispielsweise haben wir den Prototyp des Sonnensensors für Finnlands ersten Satelliten mithilfe des Arduino entwickelt (siehe [Abb. 1](#)).



Abb. 1 *Der erste finnische Satellit startet 2014. Den Prototyp für den Sonnensensor haben wir mithilfe des Arduino entworfen und gebaut.*

Wie Sie dieses Buch lesen sollten

Wenn Sie eine Idee haben, können Sie mithilfe dieses Buches schnell einen ersten Prototyp bauen. Anstatt Stunden damit zuzubringen, die Datenblätter von Komponenten zu studieren, können Sie einfach einen Sensor und einen fertigen Schaltplan sowie den zugehörigen Code herausgreifen. Die Sensoren können Sie als Bausteine für Ihre Projekte verwenden, aber anders als bei anderen Bausystemen, wie Stabilbaukästen und Lego, haben Sie beim Arduino und beim Raspberry Pi wirklich fast unbegrenzte Möglichkeiten.

Wenn Sie wissen, was Sie messen wollen, können Sie schnell den passenden Sensor dafür finden. Dieses Buch ist nach den Erscheinungen der physischen Welt geordnet, die sich messen lassen:

- Entfernung ([Kapitel 3](#))
- Rauch und Gas ([Kapitel 4](#))
- Berührung ([Kapitel 5](#))

- Bewegung (Kapitel 6)
- Licht (Kapitel 7)
- Beschleunigung und Winkelgeschwindigkeit (Kapitel 8)
- Identität (Kapitel 9)
- Elektrizität (Kapitel 10)
- Schall (Kapitel 11)
- Wetter und Klima (Kapitel 12)

Sie können dieses Buch als Inspirationsquelle verwenden: Stöbern Sie darin, um eine Vorstellung davon zu bekommen, was für Sensoren verfügbar sind, und lassen Sie sich davon zu neuen Projekten anregen.

Wenn Sie wissen wollen, wie die Sensoren mit dem Arduino bzw. dem Raspberry Pi zusammenarbeiten, dann werden Ihnen die ausführlichen Erläuterungen gefallen. Alle Codebeispiele sind vollständig und zeigen die gesamte Interaktion mit dem Sensor. Wenn Sie genau verstehen, wie die in diesem Buch behandelten Sensoren funktionieren, können Sie Ihre neu erworbenen Fähigkeiten auch auf andere Sensoren übertragen, sogar auf solche, die es heutzutage noch gar nicht gibt.

Bei der Auswahl der Sensoren haben wir uns für besonders nützliche und interessante entschieden, ohne darauf zu achten, ob sie besonders leicht oder besonders schwer zu verwenden sind. Daher finden Sie in diesem Buch Experimente mit unterschiedlichem Schwierigkeitsgrad.

In jedem Kapitel werden Experimente, Praxisexperimente und ein Testprojekt vorgestellt:

1. Die *Experimente* geben Ihnen eine schnelle Anleitung dafür, wie Sie einen bestimmten Sensor am Arduino oder am Raspberry Pi einsetzen können. Sie können Sie als Bausteine für Ihre eigenen Projekte verwenden, aber auch einfach nur zum Ausprobieren, wie die einzelnen Sensoren funktionieren.
2. In *Praxisexperimenten* spielen Sie mit den Sensoren herum und beobachten damit Änderungen in Ihrer Umgebung. Dadurch können Sie sich ein Bild davon machen, wie der Sensor die Welt sieht und wie er wirklich funktioniert.
3. Die Arbeit mit Sensoren macht mehr Spaß, wenn Sie auch irgendetwas mit den Messwerten tun. In den *Testprojekten* bauen Sie jeweils ein Gerät, das den Sensor nutzt. Dabei lernen Sie verschiedene Ausgabemöglichkeiten wie RGB-LEDs, E-Paper und Servomotoren kennen. Testprojekte können Sie auch als Ausgangspunkte für Ihre eigenen Konstruktionen nutzen.

Eingabe, Verarbeitung, Ausgabe

Alle Roboter oder Geräte, die Sie bauen, brauchen eine *Eingabe*, eine *Verarbeitung* der Daten und eine *Ausgabe*.

1. Da die meisten der von Ihnen gebauten Geräte weder über eine Tastatur noch über eine Maus verfügen, dienen die Sensoren als Eingabequelle. Werfen Sie einen Blick in das Inhaltsverzeichnis, aber denken Sie daran, dass hier nur ein Bruchteil der tatsächlich existierenden Sensoren behandelt wird. Es gibt unzählige weitere Sensoren, um alles zu messen, was Sie sich vorstellen können.
2. Die Verarbeitung geschieht in Ihrem Programm, das auf dem Arduino oder Raspberry Pi ausgeführt wird. In diesem Programm geben Sie an, was als Nächstes geschieht.
3. Die Ausgabe hat einen Einfluss auf die Welt außerhalb des Geräts. Sie können eine LED aufleuchten lassen, einen Servomotor einschalten oder einen Klang abspielen. Das sind die drei häufigsten Arten von Ausgaben, aber es gibt noch weitere (z. B. haptische Rückmeldung durch Vibration, die Anzeige auf einem E-Paper-Bildschirm oder das Einschalten von Haushaltsgeräten).

Protokolle

Ein *Protokoll* bestimmt, wie ein Sensor mit einer Mikrocontroller-Platine, wie dem Arduino oder dem Raspberry Pi, kommuniziert. Es legt fest, wie die Kabel angeschlossen werden und wie Ihr Code die Messwerte abfragen muss.

Es gibt zwar unzählig viele Sensoren, aber nur eine eingeschränkte Anzahl von häufig verwendeten Protokollen. Sie werden diese Protokolle im Einzelnen kennenlernen, während Sie an den Experimenten und Projekten arbeiten. In diesem Abschnitt geben wir Ihnen aber bereits einen Überblick über das, was Sie erwartet.

Eine Übersicht über die gebräuchlichen Sensorprotokolle finden Sie in [Tabelle I](#).

■ *Digitale Widerstandssensoren*

Einige Sensoren funktionieren wie Drucktasten mit zwei Zuständen, »ein« und »aus«. Sie lassen sich besonders einfach ablesen. Der Ein-Status wird dadurch dargestellt, dass an den Eingangspin des Mikrocontrollers der Spannungspegel **HIGH** angelegt wird. Je nach verwendeter Platine ist das gewöhnlich 3,3 V oder 5 V.

■ *Analoger Widerstandssensor*

Analoge Widerstandssensoren ändern ihren Widerstand als Reaktion auf eine physische Änderung (z. B. die Drehung eines Reglers). Der Arduino und der

Raspberry Pi messen die Widerstandsänderung anhand der am Sensor anliegenden Spannung. Wenn Sie beispielsweise an einem Potenziometer drehen, wird der Widerstand größer oder kleiner. Solche analogen Widerstandssensoren lassen sich am Arduino problemlos einsetzen. Beim Raspberry Pi dagegen brauchen Sie einen zusätzlichen Chip, um die Analogwerte zu messen. In [Abschnitt 3.5](#) lernen Sie, wie Sie den Analog-Digital-Wandler MCP3002 zur Widerstandsmessung mit dem Raspberry Pi verwenden. Die meisten analogen Eingangssensoren geben den Messwert in Form des Widerstands an, sind also analoge Widerstandssensoren.

■ *Pulsbreite*

Einige Sensoren geben den Messwert über die Pulsbreite an, also die Länge der Zeit, in der der Pin auf dem Pegel `HIGH` gehalten wird. Um die Impulslänge zu lesen, verwenden Sie Funktionen wie `pulseIn()` und `gpio.pulseInHigh()`. Bei der Nutzung solcher Funktionen müssen Sie sich nicht um die maschinen-nahen Mikrocontroller-Operationen wie *Interrupts* kümmern, da Ihnen dies von einer Bibliothek abgenommen wird.

■ *Serieller Port*

Ein *serieller Port* überträgt Textzeichen zwischen zwei Geräten. Das ist dieselbe Technik, die auch Ihr Computer zur Kommunikation mit dem Arduino über USB einsetzt. Mit dem seriellen Monitor werden Sie gut vertraut werden, da Sie in mehreren Projekten Meldungen auf dem seriellen Monitor der Arduino-IDE ausgeben.

■ *I2C*

I2C ist ein häufig verwendetes Industriestandardprotokoll. Es wird auf vielen Computern eingesetzt und ist vor allem dank der Wii-Nunchuk-Joysticks bekannt geworden. I2C macht es möglich, 128 Geräte an einem Bus anzuschließen. In diesem Buch finden Sie fertigen Code und Schaltpläne für zwei Sensoren, die I2C nutzen.

■ *SPI*

SPI ist ein weiteres Industriestandardprotokoll. Der in diesem Buch vorgestellte Code für den Einsatz eines Analog-Digital-Wandlers am Raspberry Pi lässt sich sehr einfach verwenden. Von Grund auf neuen Code für andere Geräte zu schreiben, die SPI nutzen, erfordert jedoch etwas mehr Anstrengung.

■ *Bitoperationen*

Es gibt auch ungewöhnliche Sensoren, die keines der Standardprotokolle verwenden. In solchen Fällen müssen Sie selbst für Code sorgen, der mit dem Sensor kommunizieren kann. Dabei ist es oft erforderlich, das vom Sensor kommende Signal auf Bitebene zu bearbeiten. Ein Beispiel dafür finden Sie in [Abschnitt 12.3](#).

Beim Experimentieren mit den Sensoren lernen Sie diese Protokolle immer besser kennen. Wenn Sie es eilig haben, neue Sensoren in Ihre Roboter und innovativen Geräte einzubauen, können Sie auch einfach den Code aus diesem Buch verwenden und sich später um die Einzelheiten kümmern.

Protokoll	Beispielwerte	Arduino	Raspberry Pi	Beispielsensoren
Digitaler Widerstandssensor	1 oder 0	<code>digitalRead()</code>	<code>botbook_gpio.read()</code>	Drucktaste, IR-Sensor, Neigungssensor, passiver IR-Bewegungssensor
Analoger Widerstandssensor	5 %, 10 %, 23 °C	<code>analogRead()</code>	<code>botbook_mcp3002.readAnalog()</code> , Chip	Potenzimeter, lichtabhängiger Widerstand, Alkoholsensor MQ-3, MQ-X-Gassensoren (Rauch, Kohlenwasserstoffe, CO usw.), Drucksensor Flexi-Force, Flammensensor KY-026, Farbsensor HDJD-S822-QR999, Temperatursensor LM35, Bodenfeuchtigkeitssensor
Pulsbreite	20 ms	<code>pulseIn()</code>	<code>gpio.pulseInHigh()</code>	Ultraschall-Entfernungssensoren Ping und HC-SR04, Beschleunigungssensor MX2125
Serieller Port	A9B3C5B3C5	<code>Serial.read()</code>	<code>pySerial.read()</code>	Fingerabdruckscanner GT-511C3, RFID-Scanner ELB149C5M
I2C	(2,11 g, 0,0 g, 0,1g), sehr genaue Werte	<code>Wire.h</code>	<code>smbus</code>	Wii Nunchuk, Beschleunigungsmesser/Gyroskop-Kombimodul MPU 6050, Luftdrucksensor GY-65
SPI	57°, sehr genaue Werte	Bitoperationen	<code>spidev</code>	Analog-Digital-Wandler MCP3002
In sehr kurzen Impulsen codierte Bits	53 %	Bitoperationen	Bitoperationen	Feuchtigkeitssensor DHT11

Tab. 1 Sensorprotokolle mit ansteigender Kompliziertheit

Folgen Sie Ihrem eigenen Stil

Nackte Platinen und Bauteile wirken für die meisten Benutzer nicht gerade ansprechend. Wenn Sie Ihr Gerät oder Ihren Roboter mit einem attraktiven Gehäuse versehen, kann das schon einen großen Unterschied ausmachen.

Zu allen Projekten in diesem Buch machen wir Ihnen einen Vorschlag für die Verkleidung, aber das heißt nicht, dass Sie unseren Anleitungen stur folgen müs-

sen. Probieren Sie auch andere Materialien und andere Herstellungsmöglichkeiten aus.

Wie wäre es mit Pappe (siehe [Abb. II](#)), Stoff ([Abb. III](#)) oder einem Gehäuse aus einem 3D-Drucker ([Abb. IV](#))?



Abb. II Pappmodell. Foto von der Ars Electronica in Linz (nicht von uns gemacht).



Abb. III Pelzroboter. Foto von der Ars Electronica in Linz (nicht von uns gemacht).



Abb. IV *Bender in 3D. Foto von der Ars Electronica in Linz (nicht von uns gemacht).*

Erlernen Sie neue Techniken und probieren Sie sie aus. Es macht die Arbeit gleich viel interessanter, wenn Sie zwischen der ständigen Löterei auch mal etwas schmieden oder töpfern.



Abb. V *Grundmodell für einen animatronischen Gorillakopf und die damit hergestellte Latexhaut*

In unseren eigenen Projekten verwenden wir auch viel recyceltes Material. Es ist nicht nur billig (kostenlos!), sondern kann einem Projekt auch ein ganz besonderes Erscheinungsbild verleihen.

Bauteile kaufen

Wenn Sie qualitativ hochwertige Bauteile brauchen, mit denen Sie problemlos arbeiten können, suchen Sie sich einen renommierten Anbieter aus, vorzugsweise aus der westlichen Welt. Wenn Sie billige Bauteile haben wollen, sehen Sie sich in Asien um.

Zu den Qualitätsanbietern für Bastler zählen Maker Shed (<http://www.makershed.com>), SparkFun (<https://www.sparkfun.com>), Parallax (<http://www.parallax.com>) und Adafruit (<http://www.adafruit.com/>). Maker Shed gehört dem Herausgeber der Originalversion dieses Buches. SparkFun vertreibt viele Breakout-Platinen, die Sie an die Steckerleisten anlöten müssen. Parallax ist der Erfinder von Basic Stamp, der vorherigen Generation von Mikrocontroller-Platinen für Bastler. Adafruit bietet eine Menge Bauteile an, viele davon hat Adafruit selbst entwickelt. Auf den Websites von SparkFun und Adafruit finden Sie viele Informationen über die dort erhältlichen Komponenten, unter anderem auch Bauanleitungen.

Weitere gut sortierte Anbieter (im deutschsprachigen Bereich) sind Watterott Electronics (<http://www.watterott.com>), Digi-Key (<http://www.digikey.de>) und Mouser Electronics (<http://www.mouser.de>).

Heutzutage sind sogar bekannte Handelsunternehmen wie Element14 (<http://www.element14.com>) und RS electronics (<http://de.rs-online.com>) in den Bastlermarkt vorgedrungen. Da sie in ihren umfangreichen Katalogen inzwischen eigene Bereiche für den Arduino und den Raspberry Pi eingerichtet haben, lassen sich die entsprechenden Bauteile darin jetzt auch einfacher finden.

Für einige ausgefallene Bauteile oder um wirklich äußerst niedrige Preise zu erhalten, müssen Sie sich in Asien umsehen. DealExtreme (<http://dx.com>) ist zurzeit der Renner. Die Lieferung dauert lange und die Qualität schwankt, aber die Preise sind niedrig und das Sortiment ist sehr breit gefächert. Ein weiterer asiatischer Anbieter, der einen Versuch lohnt, ist AliExpress (<http://www.aliexpress.com>).

Schreibweisen in diesem Buch

In diesem Buch haben wir die folgenden Schreibweisen verwendet:

Kursivschrift

Weist auf einen neuen Begriff hin und kennzeichnet URLs, E-Mail-Adressen, Dateinamen und -erweiterungen.

Nichtproportionalschrift

Wird für Programmlistings und im Fließtext für Programmelemente wie Variablen und Funktionsnamen, Datenbanken, Datentypen, Umgebungsvariablen, Anweisungen und Schlüsselwörter verwendet.

In diesen Kästen finden Sie Tipps, Warnungen oder allgemeine Hinweise

Die Codebeispiele verwenden

Den gesamten Quellcode zu diesem Buch können Sie auf der Seite www.dpunkt.delsensoren herunterladen (Schauen Sie auch auf die englische Website des Originalbuchs: <http://makesensors.botbook.com>).

Um das ZIP-Archiv zu entpacken, doppelklicken Sie darauf. Sie können auch rechtsklicken und *Extrahieren* aus dem Kontextmenü wählen.

Dieses Buch soll Ihnen dabei helfen, Aufgaben zu erledigen. Den Code aus diesem Buch können Sie in Ihren Programmen und der zugehörigen Dokumentation benutzen. Sie müssen uns nicht um Erlaubnis fragen, es sei denn, dass Sie einen erheblichen Teil des Codes reproduzieren. Wenn Sie beispielsweise ein Programm schreiben, das mehrere Teile des Codes aus diesem Buch nutzt, brauchen Sie dazu keine ausdrückliche Erlaubnis von uns. Wollen Sie dagegen eine CD-ROM mit Beispielen aus diesem Buch verkaufen oder verteilen, ist eine Genehmigung erforderlich. Wenn Sie eine Frage beantworten, indem Sie aus diesem Buch und dem Beispielcode zitieren, brauchen Sie dazu keine ausdrückliche Erlaubnis. Wollen Sie dagegen eine erhebliche Menge des Beispielcodes aus diesem Buch in Ihre Produktdokumentation aufnehmen, ist eine Erlaubnis notwendig.

Eine Quellenangabe ist nicht erforderlich, aber wir würden uns darüber freuen. Gewöhnlich umfasst eine solche Angabe den Titel, den Autor, den Herausgeber und die ISBN, z. B.

»*Sensoren - messen und experimentieren mit Arduino und Raspberry Pi* von Tero Karvinen, Kimmo Karvinen und Ville Valtokari. Copyright 2015 Tero Karvinen, Kimmo Karvinen und Ville Valtokari, 978-3-86490-160-7«.

Wenn Sie unsere Codebeispiele auf eine Weise verwenden möchten, die den Rahmen der »fairen Verwendung« sprengt oder über die hier gegebene Erlaubnis hinausgeht, wenden Sie sich bitte über bookpermissions@makermedia.com an uns.

Danksagungen

Die Autoren möchten Hipsu, Marianna, Nina, Paavo Leinonen und Valteri danken.

1 Raspberry Pi

Wir empfehlen Ihnen, mit dem Modell B des Raspberry Pi anzufangen, das einen kabelgebundenen Ethernet-Anschluss sowie ausreichend USB-Ports für eine Maus und eine Tastatur aufweist. Das macht den Einstieg viel einfacher.

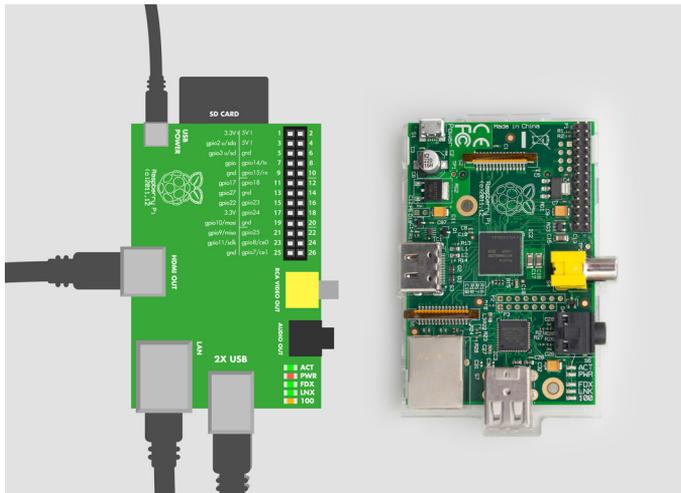


Abb. 1–1 Peripherieanschlüsse am Raspberry Pi

Sofern Sie den Raspberry Pi nicht als Teil eines Komplettpakets gekauft haben, ist kein Gehäuse dabei. Es macht aber überhaupt nichts aus, einfach die nackte Platine auf den Tisch zu legen; dadurch wirken Sie noch mehr wie ein richtiger Elektronikfreak. Wenn Sie Zugang zu einem 3D-Drucker, CNC-Maschinen oder Laserschneidergeräten haben, können Sie auch selbst ein Gehäuse bauen. Viele Vorlagen dafür finden Sie auf <http://www.thingiverse.com>.

Eine SD-Speicherkarte mit 4 GB ist groß genug für das Betriebssystem. Bei einer größeren Karte kann die Abnutzung im Laufe der Zeit jedoch geringer sein (der Verschleiß kann über einen größeren Speicherplatz verteilt werden). Wenn Sie also über eine Speicherkarte von 8 GB oder mehr verfügen, ist das noch besser.

Der Raspberry Pi kann ein Full-HD-Display steuern, und über einen HDMI-Anschluss können Sie sogar Tonsignale senden. Daher eignet sich ein HD-Fernseher gut als Anzeige für einen Pi.

Eine Tastatur und eine Maus erleichtern den Einstieg. Das Modell B des Raspberry Pi verfügt über zwei USB-Anschlüsse, was gerade für Maus und Tastatur ausreicht.

Wenn Sie zusätzlich einen USB-WLAN-Adapter verwenden möchten, brauchen Sie einen USB-Hub mit eigener Stromversorgung. Eine Liste von WLAN-Adaptoren, die mit dem Raspberry Pi kompatibel sind, finden Sie auf http://elinux.org/RPi_USB_Wi-Fi_Adapters. Um das WLAN auf dem Pi einzurichten, doppelklicken Sie nach der Installation des Betriebssystems und dem Start der grafischen Benutzeroberfläche auf das Symbol **WiFi CONFIG** auf dem Desktop.

Der teuerste 30-€-Computer?

Kabel, Tastatur, Maus und Anzeige kosten zusammen mehr als zwei Raspberry Pi. Wenn Sie all diese Teile nicht noch irgendwo als Staubfänger gelagert haben, kann das eine ziemlich große Ausgabe für einen so kleinen Computer sein. Trotzdem spart es Zeit (= Geld), wenn Sie sich eine komfortable Entwicklungsumgebung einrichten. Wenn Ihr Projekt funktioniert, können Sie das System später auf die absolut notwendigen Teile abspecken. Wie heißt es so schön? Der Raspberry Pi ist der einzige 30-€-Computer, der 100 € kostet.

Wenn Sie mit SSH oder VNC über das Netzwerk mit dem Raspberry Pi arbeiten wollen, müssen Sie nur für einen Netzwerk- und einen Stromanschluss sorgen. Dann brauchen Sie – außer für die Ersteinrichtung – keine Tastatur, keine Maus und keinen Monitor.

1.1 Raspberry Pi: Von null zum ersten Start

In diesem Kapitel erfahren Sie, wie Sie Ihren Raspberry Pi schnell arbeitsbereit machen. Als Erstes müssen Sie Linux auf dem Pi installieren. Dazu sind folgende Schritte erforderlich:

- Laden Sie den Installer herunter und entpacken Sie ihn auf einer formatierten SD-Karte.
- Legen Sie die Karte in den Raspberry Pi ein und schließen Sie eine Tastatur, eine Maus und einen Monitor an.
- Schalten Sie den Pi ein, wählen Sie aus, was installiert werden soll, und warten Sie.

Sobald alles fertig ist, müssen Sie den Pi mit der grafischen Linux-Oberfläche neu starten.

Sie benötigen dazu die folgenden Teile:

- Raspberry Pi Modell B
- Micro-USB-Kabel und USB-Ladegerät (oder Computer)
- 4-GB-SD-Karte
- Display mit HDMI-Anschluss
- HDMI-Kabel
- USB-Maus
- USB-Tastatur

1.1.1 NOOBS*.zip entpacken

Laden Sie *NOOBS_vX_Y_Z.zip* von <http://raspberrypi.org/downloads> herunter (während wir diese Zeilen schreiben, ist die Version *NOOBS_v1_3_4.zip* aktuell, aber wenn Sie dies lesen, kann der Dateiname durchaus schon wieder anders lauten).

Alle in diesem Buch erwähnten wichtigen Links sowie Kopien einiger Dateien finden Sie auf <http://www.dpunkt.de/sensoren> (oder <http://botbook.com>).

Legen Sie die SD-Karte in Ihren Computer ein. Die meisten SD-Karten sind bereits werkseitig mit FAT32 formatiert. Sofern Sie also keine SD-Karte verwenden, die Sie selbst formatiert haben, reicht es aus, die NOOBS-Zip-Datei auf der SD-Karte zu entpacken. Vergewissern Sie sich anschließend, dass sich die Datei *bootcode.bin* im Wurzelverzeichnis (oberste Ebene) der SD-Karte befindet.

Wenn Sie die SD-Karte formatieren müssen, verwenden Sie dazu das Formatierungsprogramm der SD Card Association.

In modernen Versionen von Linux, Windows und OS X können Sie einfach auf die NOOBS-Zip-Datei doppelklicken oder rechtsklicken, um sie zu entpacken. Bei älteren Versionen von Windows können Sie *7zip* installieren und damit Zip-Dateien entpacken.

1.1.2 Kabel anschließen

Der Anschluss der Kabel ist einfach, da jedes Kabel nur in die dafür vorgesehene Buchse passt. Schließen Sie die Maus und die Tastatur an den USB-Ports des Raspberry Pi an. Wenn Sie einen HDMI-Monitor verwenden, verbinden Sie ihn und den Pi mit einem HDMI-Kabel. Bei einem PAL- oder NTSC-Bildschirm

schließen Sie ein Composite-Video-Kabel an der gelben Buchse des Pi und am Monitor an.

Als Nächstes verbinden Sie den Raspberry Pi über das Micro-USB-Kabel mit der Stromversorgung. Dabei kann es sich entweder um den USB-Anschluss eines Computers oder um ein USB-Ladegerät mit mindestens 700 mA handeln.

1.1.3 Raspbian hochfahren und installieren

Sobald Sie den Stromanschluss hergestellt haben, fährt der Raspberry Pi hoch. Sie müssen ihn nicht eigens einschalten.

Wenn sich auf dem Bildschirm nichts tut, müssen Sie möglicherweise noch den richtigen Ausgabemodus für den Raspberry Pi wählen. Der standardmäßige Ausgabemodus ist HDMI, aber wenn Sie über HDMI verbunden sind und trotzdem nichts zu sehen ist, versuchen Sie, auf der an den Pi angeschlossenen Tastatur (2) zu drücken, um den sicheren HDMI-Modus auszuwählen. Erfolgt die Verbindung über den Composite-Video-Anschluss (gelb), drücken Sie (3) für einen PAL-Monitor oder -Fernseher bzw. (4) für ein NTSC-Gerät.

Zur Begrüßung sehen Sie ein Menü, das verschiedene Betriebssysteme, Sprachen und Tastaturtypen anbietet. Wählen Sie *Raspbian [RECOMMENDED]* (siehe Abb. 1–2), Ihre Sprache und den Typ der verwendeten Tastatur aus.

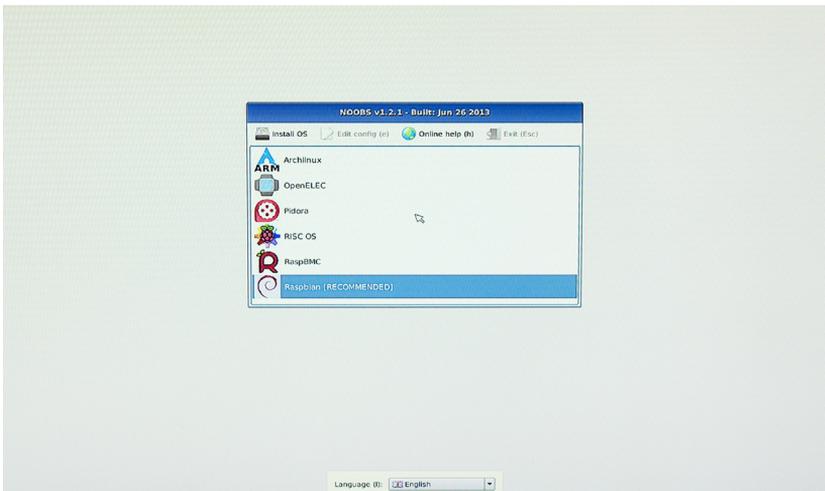


Abb. 1–2 Das Betriebssystem auswählen

Wenn Sie Debian, Mint oder Ubuntu kennen, werden Sie sich bei dieser Auswahl schnell zurechtfinden. Wenn nicht, lesen Sie einfach weiter – dann werden Sie auch bald damit vertraut sein! Es braucht einige Minuten, bis die Installation von

Raspbian abgeschlossen ist (siehe [Abb. 1-3](#)). Anschließend erhalten Sie die Meldung, dass das Betriebssystem erfolgreich installiert wurde. Drücken Sie (**Enter**) oder klicken Sie auf **OK**, um den Pi neu zu starten.

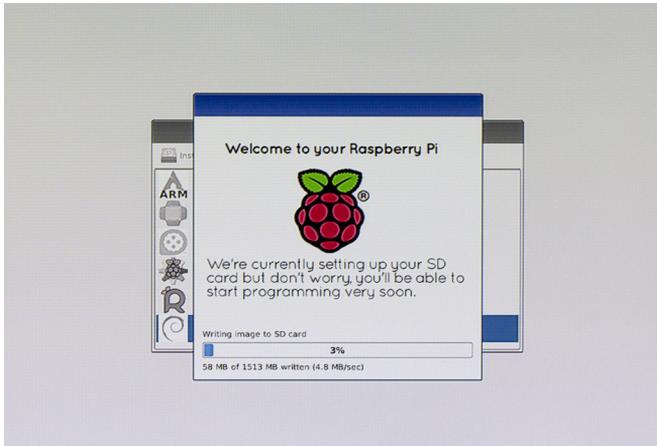


Abb. 1-3 Raspbian wird installiert.

Das Dienstprogramm zur Konfiguration des Raspberry Pi wird geöffnet. Zur Navigation verwenden Sie die Pfeiltasten und (**Tab**) und zur Auswahl einer Option (**Enter**) bzw. (**Return**) (siehe [Abb. 1-4](#)).



Abb. 1-4 Das Passwort ändern

Sie sollten die Option *Boot to Desktop* aktivieren. Wenn Sie mit der Änderung der Einstellungen fertig sind, drücken Sie (**Tab**), um *Finish* auszuwählen und den Pi neu zu starten, wenn Sie dazu aufgefordert werden.

Nach dem Neustart zeigt der Raspberry Pi die grafische Benutzeroberfläche und meldet Sie automatisch an.

Wenn Sie die Option *Boot to Desktop* nicht aktiviert haben, wird beim Start immer die Kommandozeile angezeigt. Melden Sie sich dort als *raspberrypi* mit dem Passwort *pi* an (sofern Sie das Passwort nicht geändert haben) und geben Sie nach der Anmeldung *startx* ein, um das X-Window-System zu starten, also die grafische Benutzeroberfläche.

Willkommen bei Linux! Sie haben jetzt Raspbian auf dem Raspberry Pi installiert (siehe Abb. 1–5).

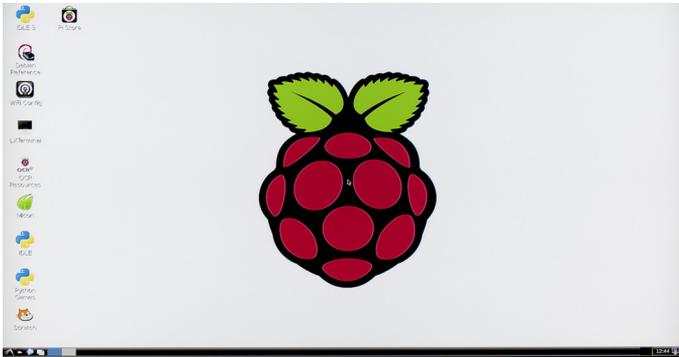


Abb. 1–5 Willkommen bei Linux

Um den Raspberry Pi abzuschalten, doppelklicken Sie auf das Ausschaltssymbol auf dem Desktop. Nachdem der Abschaltvorgang beendet ist, sollten Sie den Pi von der Stromversorgung trennen.

1.1.4 Fehlersuche bei der Raspberry Pi-Installation

Im Folgenden finden Sie Lösungen für häufig auftretende Probleme.

- Die Karte ist nicht richtig mit FAT32 formatiert.
Wenn Sie Probleme beim Starten von der SD-Karte haben, kann es sein, dass sie nicht korrekt formatiert ist. Unter Linux verwenden Sie den mitgelieferten grafischen Partitionierungseditor. (Geben Sie `sudo gparted` ein, um ihn zu starten.) Formatieren Sie die gesamte Karte mit FAT. Mit dem Befehl `sudo palimpsest` (oder `sudo gnome-disks`) können Sie noch ein anderes Werkzeug dieser Art aufrufen. Für erfahrene Benutzer steht mit `sudo parted` ein klassisches Kommandozeilen-Partitionierungsprogramm zur Verfügung. Unter Windows und OS X verwenden Sie das Formatierungsprogramm der SD Association (https://www.sdcard.org/downloads/formatter_4/). Unter Windows wählen Sie in diesem Programm die Option *Format size adjustment*, auf dem Mac *Overwrite format*.
- Die rote Stromanzeige-LED (PWR) leuchtet nicht.
Die Stromanzeige-LED bleibt dunkel oder blinkt? Sie geht nur kurz an und erlischt dann wieder? Dann hat der Raspberry Pi nicht genügend Strom. Schließen Sie ihn an eine USB-Stromversorgung an, die 5 V und mindestens 1 A liefert. Wenn der Pi an der USB-Buchse eines Laptops hängt, dann wechseln Sie zu einem Desktop-Computer, zu einem leistungstärkeren Mobiltelefon- oder einem Tablet-Ladegerät.

- Der Bildschirm bleibt schwarz (obwohl die rote PWR-LED leuchtet).
Möglicherweise kann der Raspberry Pi den Bootloader auf der SD-Karte nicht lesen. Schalten Sie ihn ab, entfernen Sie die SD-Karte und legen Sie sie wieder fest ein. Stellen Sie sicher, dass sich die erste Datei in der Startsequenz, *bootcode.bin*, auf der obersten Ebene der SD-Karte befindet. Wenn das Problem weiterhin besteht, formatieren Sie die SD-Karte und entpacken Sie die NOOBS-Zip-Datei erneut. Sollte das immer noch keine Abhilfe schaffen, probieren Sie es mit einer anderen SD-Karte.
- Auf dem Bildschirm sind nur vier farbige Kästen zu sehen.
Der Bootloader konnte zwar von der SD-Karte gelesen werden, aber es war nicht möglich, die Datei *kernel.img* des Betriebssystems hochzufahren. Formatieren Sie die SD-Karte und entpacken Sie die NOOBS-Zip-Datei erneut oder probieren Sie es mit einer anderen SD-Karte.
- Der Startvorgang schlägt fehl und es werden Fehlermeldungen angezeigt.
Trennen Sie alle USB-Geräte wie Tastatur, Maus und ggf. WLAN-Adapter ab. Lassen Sie nur SD-Karte, Anzeige und Stromversorgung angeschlossen. Nehmen Sie die SD-Karte heraus und legen Sie sie erneut ein, um sicherzustellen, dass der Kontakt ordnungsgemäß hergestellt ist. Wenn das Problem weiterhin besteht, formatieren Sie die SD-Karte und entpacken Sie die NOOBS-Zip-Datei erneut.
- Das Betriebssystem arbeitet fehlerhaft.
Wenn normale Befehle nicht funktionieren, auf dem Bildschirm nur wirres Zeug angezeigt wird oder der Raspberry Pi plötzlich aufhört zu arbeiten, ist das kein Grund zur Panik. Halten Sie beim Starten die (Umschalt)-Taste gedrückt und wählen Sie die Option zur Neuinstallation von Raspbian. Das geht schnell und einfach, allerdings werden dabei alle Daten auf der SD-Karte gelöscht. Sollte das keine Abhilfe schaffen, formatieren Sie die Karte neu, entpacken darauf die NOOBS-Zip-Datei und führen die Installation erneut durch.
- Es besteht keine Verbindung zum Internet.
Wenn Sie vor dem Starten ein Ethernet-Kabel angeschlossen haben, sollte es in einem normalen Netzwerk eigentlich ganz problemlos funktionieren. Schauen Sie nach, ob die Verbindungsanzeige (LNK) auf dem Raspberry Pi leuchtet. Wenn nicht, glaubt der Pi, dass ein Ende des Kabels nicht richtig angeschlossen ist. Normalerweise sollten auch 100 (zur Anzeige einer Verbindung mit 100 Mbit/s oder mehr) und FDX (Vollduplex) leuchten. Wenn die LNK-Anzeige leuchtet, es aber trotzdem Probleme gibt, können Sie die folgenden anspruchsvolleren Befehle zur Fehlersuche ausprobieren: `ipconfig` (zeigt die Konfiguration der Netzwerkkarte an), `route -n` (zeigt die Routingtabelle für die Netzwerkverbindung an), `cat /etc/resolv.conf` (zeigt den verwendeten Namensserver an) und `ping -c 1 google.com` (teilt Ihnen mit, ob Sie über das Netzwerk Google erreichen können).

Wenn Sie irgendwelche Fehlermeldungen erhalten, die hier nicht erwähnt werden, geben Sie den genauen Wortlaut der betreffenden Meldung in das Suchfeld von Google ein. Achten Sie genau auf die Schreibweise! Wenn die Meldung beim Hochfahren des Systems erscheint, nehmen Sie mit einer Kamera oder einem Handy ein Foto davon auf und schreiben Sie sie ab.

1.2 Willkommen bei Linux

Raspbian *ist* Linux. Nun, zumindest baut es auf Linux auf. Der Name »Linux« bezeichnet den Betriebssystemkernel und das Betriebssystem selbst. Das Betriebssystem Linux besteht aus dem Kernel und Tausenden von Dienstprogrammen und Anwendungen aus verschiedenen Quellen.

Der Raspberry Pi ist keine Workstation. Was die Rechenleistung angeht, ist er eher mit dem Einsteigermodell eines Tablet-Computers oder mit einem Mobiltelefon vergleichbar. Selbst wenn Sie also die grafische Benutzeroberfläche des Pi verwenden, ist das noch lange kein Ersatz für Ihren Laptop oder Desktop-Computer. Wegen der äußerst geringen Rechenleistung und des kleinen Arbeitsspeichers können Anwendungen wie LibreOffice und Mozilla Firefox nicht darauf laufen.

1.2.1 Die allgegenwärtige Kommandozeile

Sind Sie bereit, sich mit den Möglichkeiten der Eingabeaufforderung `$` vertraut zu machen?

Die Kommandozeile hat sich bewährt, und es kann durchaus sein, dass Sie ihre Verwendung noch Ihren Enkelkindern beibringen werden. Die ständig benutzten Befehle wie `pwd`, `ls` und `cat` gab es schon lange, bevor Linux erfunden wurde. (Linux wurde von einem finnischen Studenten in Helsinki entwickelt, und zwar gerade mal fünf Kilometer von dem botanischen Garten entfernt, in dem ich diesen Text verfasse!) Sowohl unter OS X als auch unter Windows tauchen »Power-User« in die Kommandozeile ein, wenn sie etwas tun müssen, was mit der Maus allein nicht möglich ist.

Die meisten Befehle, die Sie auf dem Raspberry Pi verwenden werden, sind die gleichen, die Sie auch auf einem Mac- oder Linux-Computer einsetzen, und viele ähneln sogar den Kommandozeilenwerkzeugen von Windows.

Wie Sie vielleicht wissen, laufen weltweit die meisten Server mit Linux, ebenso Google, Facebook, Amazon und die meisten Supercomputer. Webserver haben keine grafische Benutzeroberfläche, weshalb die meisten Programmierer und Systemadministratoren den Umgang mit der Kommandozeile (Command Line Interface, CLI) kennen müssen. Die gleichen Befehle können Sie auch auf einem Desktop- oder Laptop-Computer mit Linux einsetzen.