

#makers
DO IT.

Inkl. Einsatz
von WiringPi,
ALSA & OpenCV



Raspberry Pi

Harald Schmidt

RASP- BERRY PI

programmieren mit
C/C++ und **BASH**

Mehr als 50 Programme rund um
Foto, Video & Audio

HANSER

Harald Schmidt

Raspberry Pi programmieren mit C/C++ und Bash



BLEIBEN SIE AUF DEM LAUFENDEN!

Hanser Newsletter informieren Sie regelmäßig über neue Bücher und Termine aus den verschiedenen Bereichen der Technik. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter

www.hanser-fachbuch.de/newsletter

Harald Schmidt

Raspberry Pi programmieren mit C/C++ und Bash

Mehr als 50 Programme rund um
Foto, Video & Audio

HANSER

Der Autor:

Harald Schmidt, München

Alle in diesem Buch enthaltenen Informationen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Weise aus der Benutzung dieser Informationen – oder Teilen davon – entsteht, auch nicht für die Verletzung von Patentrechten, die daraus resultieren können.

Ebenso wenig übernehmen Autor und Verlag die Gewähr dafür, dass die beschriebenen Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt also auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benützt werden dürften.

Bibliografische Information der deutschen Nationalbibliothek:

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

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

ISBN 978-3-446-45342-5

E-Book-ISBN 978-3-446-45349-4



© 2018 Carl Hanser Verlag, München

Lektorat: Julia Stepp

Herstellung: Isabell Eschenberg

Umschlagrealisation: Stephan Rönigk

Satz: Kösel Media GmbH, Krugzell

Druck und Bindung: Druckerei Hubert & Co. GmbH und Co. KG, Göttingen

Printed in Germany

www.hanser-fachbuch.de

Inhaltsverzeichnis

Vorwort	XVII
1 Die Welt des Raspberry Pi	1
1.1 Die Modelle des Raspberry Pi	1
1.1.1 Energieversorgung	5
1.1.2 Der Raspberry Pi Zero W	6
1.1.3 Raspberry Pi 2 und Pi 3	10
1.2 SD-Karten und microSD-Karten	14
1.3 Von Wheezy zu Stretch - Downloads für Raspbian	15
1.3.1 Die Release Notes als Chronik	16
1.4 Über NOOBS zur Raspbian SD-Karte	17
1.4.1 NOOBS aus dem Internet holen und entpacken	18
1.4.2 NOOBS auf die leere (micro)SD-Karte kopieren	20
1.5 NOOBS auf dem Raspberry Pi	21
1.5.1 Wohin mit dem Bild?	21
1.5.2 Energie für NOOBS	22
1.6 Der NOOBS Recovery-Modus	28
2 Der Raspberry Pi und Linux	31
2.1 Das Abbild von Raspbian für die SD-Karte	32
2.1.1 Die Abbilddatei mit einem Linux-PC übertragen	36
2.1.2 Der Win32 Disk Imager	36
2.2 Die Grundkonfiguration des Raspbian Desktops	38
2.2.1 Tastaturlayout und Spracheinstellung	39
2.2.2 Konfiguration von Autologin, Hostname und Bootmodus	47
2.2.3 Passwortänderung	48
2.2.4 Freigabe für SSH und Abfrage der IP-Adresse	49

2.3	Der Raspberry Pi Desktop als Bedienoberfläche	50
2.3.1	Die Lage der Panelleiste ändern	51
2.3.2	Der Aufbau der Panelleiste	51
2.3.3	Programmstart über Tastaturkürzel	52
2.3.4	Eine symbolische Desktop-Verknüpfung hinzufügen	53
2.3.5	Eine neue Desktop-Datei erzeugen	54
2.3.6	Ein Programm der Anwendungsstartleiste hinzufügen	56
2.3.7	Miniprogramme im Panel	57
2.3.8	Das Konsolenkommando lxpanelctl	59
2.4	Der Main Menu Editor Alacarte	59
2.5	Das Erscheinungsbild anpassen mit LxAppearance	60
2.6	Der Raspbian Desktop auf dem PC oder Mac	62
2.7	Nützliche Programme ohne und mit grafischer Oberfläche	64
2.8	Autostart mit und ohne GUI	71
2.8.1	Die versteckte Datei .bashrc	72
2.8.2	Die Datei rc.local	74
2.9	Die Verwendung von LXTerminal in Desktop-Dateien	76
2.10	Pakete - Installation und Verwaltung	78
2.10.1	Der vorinstallierte Paketmanager pi-packages	79
2.10.2	Der Paketmanager Synaptic	80
2.10.3	Paketverwaltung über Konsolenkommandos	81
2.11	Die Bildschirmauflösung mit und ohne den RPi-Desktop einstellen	89
2.11.1	Über HDMI zum Fernseher	93
2.11.2	Über den HDMI/VGA-Adapter zum Monitor	94
2.11.3	Einstellung der Auflösung mit raspi-config	96
2.12	Audiumschialtung und Bluetooth-Lautsprecher	99
2.13	Systembackup im laufenden Betrieb	101
3	Betriebssysteme, Partitionen, Backup und Restore	103
3.1	SD-Karte, USB-Stick, Festplatte - was erkennt der Raspberry Pi?	105
3.2	SD-Karten mit Betriebssystem	108
3.2.1	Hinein in die Box	108
3.2.2	Der passende Aufkleber	109
3.2.3	Binäre Einheiten und die Kapazität von SD-Karten	109
3.3	SD-Karten identifizieren mit sd-label.sh	110
3.4	Hinweise zu Backup und Restore	113
3.5	Formatierung von SD-Karten	115
3.6	Das USB Image Tool	117
3.7	Direktkopie der SD-Karte mit piclone	119

3.8	Der Partitionsmanager GParted	121
3.9	Eine bootfähige SD-Karte mit sd-restore.sh erzeugen	124
3.10	Backup von SD-Karten mit sd-back.sh	128
3.10.1	Nullen vor dem Backup	132
3.11	Ein Blick in das Abbild der SD-Karte	134
3.12	Festplatten am Raspberry Pi	137
3.12.1	Der Einhängepunkt und die Systemkonsole	140
3.12.2	Einhängen der Partition über /etc/fstab	141
4	Programmieren mit dem Pi	143
4.1	Dateien und Programme zum Buch	144
4.2	Paketinstallation über die Skriptdateien zum Buch	146
4.3	Die Skriptdateien zum Buch – was ist wo?	148
4.4	Die Shell als Kommandointerpreter	150
4.4.1	Umgebungsvariablen	153
4.4.2	vcgenclmd	154
4.5	Strings in Bash, C und C++	156
4.6	Programmieren mit Bash	157
4.6.1	Eingabe, Ausgabe und if else	157
4.6.2	Feldtrenner und float-Werte	160
4.6.3	Formatierte Ausgabe mit printfcalc.sh	162
4.6.4	Ausgabekommandos für ASCII-Dateien	162
4.6.5	Bash-Arrays mit numerischem Index	163
4.6.6	Assoziative Bash-Arrays mit Wortindex	165
4.6.7	Funktionen, Parameter und Exit-Codes	166
4.6.8	Ausgabeumleitung und Rückgabewerte	167
4.6.9	Ein Bash-Skript mit Funktionen und Indexsuche	170
4.6.10	Indezelemente sortieren und finden mit assosort.sh	173
4.6.11	Ausgabespalten neu anordnen mit dosdir.sh	174
4.6.12	Die Ausgabe Einlesen und die IFS-Trennzeichen	176
4.6.13	Filtern mit cut und grep	179
4.6.14	Farben in der Konsolenausgabe	186
4.6.15	Zeichen entfernen mit trim.sh	188
4.6.16	Dateinamen oder Pfade extrahieren	189
4.6.17	Die Position einer Zeichengruppe oder eines Strings finden	189
4.6.18	Ein Blick in raspi-config	191
4.7	Bash in Beispielen	193
4.7.1	Wie copydesk.sh funktioniert	193
4.7.2	Farbige Listings mit colordir.sh	194

4.7.3	Analyse von Partitionsgrößen mit sd-info.sh	196
4.7.4	Feldinhalte aus Audiodateien kompakt anzeigen	199
4.7.5	EXIF-Daten anzeigen mit jhead.sh	201
4.7.6	Metadaten ermitteln mit mediainfo.sh	204
4.7.7	Diagnose der Bildschirmauflösung mit scrreso.sh	208
4.7.8	Eine Bildschirmecke zur Videowiedergabe nutzen (OMXPlayer) ...	210
4.7.9	Prozesse anzeigen mit pgrep_htop.sh	212
4.8	Skriptdateien mit ASCII-GUI (whiptail/dialog)	217
4.8.1	Vom Skript zum Kommando	221
4.8.2	Eine allgemeingültige Dateiauswahl im Skript verwenden	223
4.8.3	Programmauswahl über Checkboxes	225
4.8.4	Bilder als ASCII-Art anzeigen	228
4.8.5	Programmstart mit fotomenu.sh	234
4.8.6	Programmstart über Desktop-Dateien und ASCII-Dialoge	237
4.8.7	Dateiauswahl bei sd-mount.sh	240
4.8.8	Der Programmstarter opencvgui.sh als ASCII-GUI	243
4.9	C-Programme mit Raspbian nutzen (hello_pi/GPU)	245
4.9.1	Videoprojektion auf eine rotierende Teekanne (hello_teapot.c) ...	247
4.9.2	Mathematische Kunst mit hello_triangle2.c	248
4.9.3	Das Skript helloGPU.sh auf einem „großen“ Linux-PC starten ...	249
4.9.4	Mit make zum eigenen C-Programm	250
4.9.5	Das Skript helloGPU2.sh für weitere GPU-Programme	253
4.9.6	Programme starten mit system und popen	254
4.9.7	Zeit ist relativ	257
4.9.8	Mausklicks, Bewegungen und Scrollevents auswerten	259
4.9.9	Von der Mausbewegung zur Audioausgabe	264
4.10	Der Qt Creator als Entwicklungsumgebung	268
4.11	Die Entwicklungsumgebung Code::Blocks	273
4.12	GTK+ Programmierung	276
4.13	Die Geany-Entwicklungsumgebung	279
4.14	Python und IDLE als Entwicklungsumgebung	282
4.15	Zenity-Dialoge	286
4.16	GitHub und Raspbian	287
5	Elektronik und Programme für den GPIO-Port	289
5.1	Schaltungen aufbauen	292
5.1.1	Drähte oder Drahtbrücken	295
5.1.2	SMD-Adapter und ICs	297
5.1.3	Widerstände, Widerstandsnetzwerke und Kondensatoren	298
5.1.4	Der LED-Test	300

5.2	Eine LED per Software schalten (Ein/Aus)	301
5.3	Eine LED per Software dimmen (Pulsweitenmodulation)	304
5.4	Eine LED am Optokoppler	306
5.5	GPIO-Pins im Read-Modus (Bash)	307
5.5.1	GPIO-Pins zyklisch abfragen (Polling)	308
5.5.2	GPIO-Pins im Interrupt-Modus (Flankenerkennung)	309
5.5.3	Flankenerkennung bei mehreren Schaltern	312
5.6	WiringPi-Beispiele mit C-Quelltext	315
5.6.1	WiringPi im Interrupt-Modus	316
5.7	Experimente zur Impulswahl	317
5.7.1	Audiodateien durch Impulswahl abspielen	321
5.7.2	Impulsdiagramme mit GnuPlot	322
5.8	Von der Helligkeit zur Frequenz	325
5.8.1	Frequenzen messen mit dem Oszilloskop und mit r2f.cpp	326
5.9	Stufenschalter, Tastenfelder, Encoder und Codierschalter	331
5.10	Analog/Digital-Wandler für die SPI-Schnittstelle	333
5.10.1	Analoge Messwerte mit spiloop.c erfassen	336
5.11	Die I ² C-Schnittstelle	338
5.12	Analog/Digital-Wandler für I ² C	339
5.13	Ein LCD-Display am I ² C-Bus	344
5.13.1	Ausgaben mit sysinfo.sh	349
5.13.2	PIDs auf dem LCD-Display mit pgrep_menuctrl.sh	351
5.14	Die C-Library bcm2835 für den GPIO-Port	353
5.15	Mehr Ampere mit externen Spannungsquellen	355
5.16	Vom Schaltplan zum Layout	357
5.16.1	EAGLE Light	358
6	Sensoren und Software für Infrarot und Ultraschall	363
6.1	LIRC als Empfänger	364
6.1.1	Kommandos per IR-Fernbedienung auslösen	370
6.1.2	Ein Auswahlmenü auf dem LCD-Display	373
6.2	Entfernungsmessung per Ultraschall	376
6.3	Ein Abstandswarner mit ultraloop.c im Qt Creator	381
6.4	Entfernungsmessung mit Infrarotlicht	386
6.4.1	Distanzmessung über den MCP3426 als A/D-Wandler	388

7	Netzwerke und der Fernzugriff auf den Raspberry Pi	389
7.1	Die automatische Vergabe von IP-Adressen	392
7.2	Netzwerkkabel, Übertragungsraten und ethtool	401
7.3	Eine Direktverbindung zum Raspberry Pi	402
7.3.1	Die Link-Local-IP	406
7.4	Netzwerkkameras	407
7.4.1	Eine Direktverbindung für Netzwerkkameras	409
7.5	Der SSH-Server und ein Linux-PC als Client	410
7.5.1	SSH und die Weiterleitung des X11-Fensters	415
7.5.2	Fingerabdrücke auf (micro)SD-Karten	417
7.6	Zwei Schlüssel zum Vergleich – SSH ohne Passwort	418
7.7	PuTTY als Windows-Client	420
7.8	Remote-Zugriff mit WinSCP und Krusader als Client	424
7.9	Der SSH-Fernzugriff per ASCII-GUI	428
7.9.1	Ein Desktop-Icon auf dem Client-PC	428
7.10	Ein Remote-Desktop mit TightVNC	429
7.11	Energie sparen mit TMux	432
7.12	Die ASCII-GUI pgrep_gui.sh	437
7.13	TMux und die ASCII-GUI pgrep_gui.sh	441
7.13.1	TMux-Skripting	443
7.14	WLAN-Verbindungen und WLAN-Sticks	445
7.14.1	WLAN-Verbindungen und wpa_gui	449
7.15	Die Netzwerk-Tools arp-scan und nmap	451
7.16	Netcat – vom Chatprogramm zum Videoempfänger	453
7.17	Fünf Wege zum Raspberry Pi	454
7.18	Über Android zum Raspberry Pi	456
7.18.1	Über F-Droid zu ConnectBot	457
7.19	Über USB OTG zum Raspberry Pi Zero	460
8	Audio und Composite Video am Pi	463
8.1	Buchsen, Kabel, Signale	463
8.2	Audiowiedergabe	467
8.3	Der ALSaMixer	474
8.4	Audioaufzeichnung und ALSA-Tools	476
8.4.1	SoX mit der Webcam als Audioquelle	479
8.4.2	SoX passt auf – Reaktion auf Geräusche	485
8.4.3	Informationen zur Audiodatei mit soxi	488
8.4.4	Audioaufzeichnung im Remote-Modus	489
8.4.5	Lautsprecherdurchsagen aus sicherem Abstand (remote)	492
8.4.6	Externe USB-Soundkarten und das Programm speaker-test	493

8.5	Sprachsynthese auf dem Linux-PC	496
8.5.1	Sprachausgabe mit eSpeak	497
8.5.2	Sprachausgabe über tts.sh mit eSpeak, SVOX-Pico und Festival ...	499
9	Videodaten – Aufzeichnung und Wiedergabe	503
9.1	Der OMXPlayer	504
9.1.1	Kommandos für den OMXPlayer (DBUS-Interface)	506
9.2	Videodateien aus dem Internet holen	508
9.3	Archivierte Videos abspielen mit guiPlay.sh	512
9.4	Videoplayer und multiplay.sh	520
9.5	Wiedergabe mit videoselect.sh und avplay	524
9.6	Wo ist das Videogerät?	525
9.7	Daten zum Videogerät mit dem V4L-Treiber	528
9.8	MPEG-Spezifikationen, Codecs und Containerdateien	531
9.9	Fotografieren und Filmen mit guvcview	532
9.10	Fotografieren mit fswebcam	536
9.11	Die libav-tools und avconv (bzw. ffmpeg)	542
9.11.1	Der Webcam-Zugriff mit avconv	543
9.11.2	Vertonung mit avconv	545
9.11.3	Bildschirmvideos mit dem Raspberry Pi	546
9.11.4	Videos konvertieren mit WinFF	548
9.12	Motion auf dem Pi	550
9.12.1	Der Programmstarter motiongui.sh	553
9.12.2	Der Blick auf die Uhr	558
9.12.3	Motion-Parameter geeignet festlegen	560
9.12.4	Namen für Bilder und Videodateien	566
9.12.5	Einblendung von Zeitstempeln und Texten	567
9.12.6	Markierte Bewegungen	568
9.12.7	Bewegungsalarm auf Teilbereichen des Bildes	569
9.12.8	Automatische E-Mails bei Ereignissen	572
9.12.9	Den Videostream anzeigen und weiterleiten	573
9.12.10	Parameteränderung per ASCII-GUI und setvalue.sh	576
10	Kameramodule zum Raspberry Pi	577
10.1	Das 8-Megapixel-Kameramodul mit Sony-IMX219-Sensor	581
10.2	Fotos mit raspistill	583
10.2.1	EXIF-Daten ermitteln	588
10.2.2	Fotografieren mit fotomenu.sh	589
10.2.3	Der manuelle Belichtungsmodus	591

10.2.4	Fotografieren mit Zeit-/ISO-Automatik	593
10.2.5	Farben und Spezialeffekte	596
10.2.6	Fotos bei sehr wenig Licht und im Signalmodus	600
10.3	Konverter für Einzelbilder	604
10.4	Bilder und das FBI	605
10.5	Videos aufzeichnen mit raspivid	608
10.6	Videodateien mit MP4Box und avconv verarbeiten	611
10.7	Der UV4L-Treiber zum Kameramodul des Raspberry Pi	613
10.7.1	Fotografieren mit multicam.sh	617
10.7.2	Filmen mit dd	621
10.8	Binning bei raspistill und raspivid	622
10.9	Motion und das Kameramodul	625
11	Automatischer Datentransfer	631
11.1	SFTP-Kommandos per Skript ausführen	632
11.2	Eine RAM-Disk als Zwischenpuffer	636
11.2.1	Bilder auf der RAM-Disk	637
11.2.2	Die RAM-Disk überwachen	639
11.3	Datentransfer mit FileZilla	641
12	Audiosignale analysieren	643
12.1	Audacity	645
12.2	Die Verwendung von libsndfile	647
12.2.1	Die Abtastwerte einer Audiodatei in Textform	649
12.2.2	Ein Blick auf die Dokumentation	650
12.2.3	Ausgabeumleitung für Audiodaten mit sndout.cpp	651
12.3	Pausenerkennung und Segmentierung	655
12.3.1	Audiosegmentierung und Sprachausgabe mit audiosegm.cpp	656
12.4	Die Library tinyalsa	661
12.4.1	Abtastraten mit tinypcm_info.c ermitteln	663
12.4.2	Audioaufnahmen mit capaudio.sh und tinycap.c	664
12.5	Der Pi als Papagei	668
12.6	Spektrogramme mit sndfile-tools und Audacity	670
12.7	Vokaltraining mit voxvokal.sh	679
12.8	Das Audioquiz	683

13	OpenCV und der Pi	687
13.1	Die Konfiguration von OpenCV	688
13.2	Mit dem Compiler zur ausführbaren Datei	691
13.3	OpenCV – Dokumentation und Beispielprogramme	695
13.4	OpenCV, CMakeGUI und Code::Blocks	698
13.5	Die Qt-Projektdateien zu den Beispielprogrammen	700
13.6	Das Programm fpsec.cpp auf dem Pi benutzen	703
	13.6.1 Fenster und Tastaturcodes von fpsec.cpp	704
	13.6.2 Konturbilder vom Webcambild	705
	13.6.3 Konturbilder vom Videobild	706
	13.6.4 FloodFill auf dem Videobild	707
13.7	Programmierung mit OpenCV	708
	13.7.1 Grundlegende Datentypen und OpenCV-Klassen	708
	13.7.2 Fenster, Steuerelemente und cv::waitKey	710
	13.7.3 Optimierung von Farb- und Grauwertbildern per Tastendruck ...	711
	13.7.4 Binarisierung von Bildern mit cv::threshold	714
	13.7.5 Regionen und Beschriftungen	716
13.8	Konfiguration für die WITH_QT-Oberfläche	717
14	Objektsegmentierung mit OpenCV	719
14.1	Segmentierung von Textzeilen	719
14.2	Symbolerkennung mit iconreader.cpp	722
14.3	Symbolerkennung auf Webcambildern	725
	14.3.1 Einsatzzweck, Bedienung und Tastaturcodes	727
	14.3.2 Vom Graustufenbild zum Erkennungsergebnis	732
	14.3.3 Der Alarm und die Statistik	738
14.4	Die sprechende Lichtschranke	740
15	Experimente	741
15.1	Booten mit Sprachausgabe	741
15.2	Ein Eingabemodul mit IR-Detektor und LCD-Display	744
	15.2.1 Den Schaltungsaufbau planen	747
	15.2.2 Lötarbeiten für die Elektronik zum DOGM204 LCD-Display	749
	15.2.3 Das LCD-Display testen	750
	15.2.4 Das Eingabemodul eichen	752
	15.2.5 Die Drehung des Codierschalters auswerten	755
15.3	Ein Client/Server-Konzept für das Eingabemodul	756
	15.3.1 Nur mit Maus und Display?	762

15.4	Ein Client/Server-Konzept mit menuctrl.sh	765
15.4.1	Fotos mit MenuCtrl	770
15.5	Über MenuCtrl fswebcam starten/beenden	773
15.6	Die hello_pi GPU-Programme mit MenuCtrl starten/beenden	774
15.7	MenuCtrl im Infrarotlicht	776
15.8	MenuCtrl schon beim Booten starten	781
16	Schlusswort	783
17	Übersicht der Programme und Skriptdateien zum Buch	785
	Stichwortverzeichnis	791

Meinem Patenonkel Josef Schmidt gewidmet

Vorwort

Der Raspberry Pi ist zum Computer mit Kultstatus geworden. Das mag einerseits an der umfangreichen Literatur liegen, die für dieses Gerät verfügbar ist. Ganz sicher ist dafür aber auch das von der Raspberry Pi Foundation getragene Konzept verantwortlich, auch die Hardware als Open Source zu betrachten. So konnte ein kleiner Computer entstehen, der sich als Steuerungscomputer für Elektronikbastler einsetzen lässt und dabei nur wenig Energie verbraucht.

Softwarebasis des Pi sind Betriebssysteme, die ihre Wurzeln in Linux haben. Am bekanntesten ist dabei vielleicht Raspbian, das aus einer Debian-Linux-Distribution entwickelt wurde. Des- sen Bedienoberfläche ist einfach aufgebaut und ähnelt der klassischen Oberfläche von Windows XP.

Pi ist aber auch ein Benutzername (**pi@raspi3**) – und mit dessen Augen werden Sie in diesem Buch spannende Blicke auf Ihren Mini-PC und seine Umgebung werfen. Dieses Buch führt Sie zugleich auch in die Welt zweier recht unterschiedlicher Programmiersprachen ein:

- Bash: für Ablaufsteuerungen und als Programmstarter mit Benutzeroberfläche
- C/C++: für alle Fälle, in denen man tiefer einsteigen will und in denen Geschwindigkeit wichtig ist

Sie werden noch sehen, dass sich diese Programmiersprachen nicht widersprechen, sondern wunderbar ergänzen, denn letztlich ist auch Linux eine Kombination daraus.

Da wir Menschen hauptsächlich über das Sehen, Hören und Sprechen kommunizieren, ist es angemessen, auch einen Computer in dieser Hinsicht zu erweitern. Der Schwerpunkt der für dieses Buch erstellten Programme liegt deshalb in diesen Bereichen.

Was erwartet Sie in diesem Buch?

Kapitel 1 erklärt, welche Modelle des Raspberry Pi existieren und wie Sie Raspbian Jessie/Stretch konfigurieren und erstmals starten. Zahlreiche Verweise auf andere Kapitel und Abschnitte erleichtern Ihnen den Einstieg.

Kapitel 2 bringt Ihnen Linux und Raspbian näher. Raspbian ist derzeit das beliebteste Betriebssystem auf dem Raspberry Pi. Sie lernen viele Details der grafischen Oberfläche kennen, aber auch die Verwendung der Konsole und vieler interessanter Programme werden erklärt.

Kapitel 3 erklärt den praktischen Umgang mit der (micro)SD-Karte als Boot-Medium des Raspberry Pi. Sie erfahren einiges über Backup und Restore und wie einfach es ist, in die „Computerproduktion“ einzusteigen, denn Raspbian Jessie hat mit PiClone ein hübsches Backupprogramm bekommen.

Kapitel 4 erklärt die Verwendung vieler Linux-Kommandos. Sie erhalten eine Einführung in Bash, welche Sie dazu befähigt, kleine effektive Werkzeuge bzw. Skriptdateien zu bauen. Selbst wenn Sie daran nur begrenzt Interesse haben sollten, erfahren Sie, wie Sie viele der Programme zum Buch durch Anklicken von Desktop-Icons ganz einfach starten können. Mit Bash können Sie sehr leicht ASCII-Dialoge aufbauen, die als einfacher und informativer Programmstarter verwendbar sind. Auch C/C++ Programme lassen sich so ganz ohne Vorkenntnisse über Compiler starten. In diesem Kapitel lernen Sie außerdem die GPU-Programme von Raspbian kennen – von der rotierenden Teekanne bis zum mathematischen Kunstwerk.

Kapitel 5 führt Sie in die Welt der Elektronik ein. Die Namen Faraday, Ohm und Tesla sollten Ihnen bekannt sein. Sie müssen aber kein Elektronikexperte sein, um funktionsfähige Schaltungen aufzubauen und anzusteuern. Lassen Sie eine LED blinken. Verwenden Sie einen lichtgesteuerten Widerstand, Optokoppler oder eine antike Wählscheibe. Nutzen Sie einen A/D-Wandler und die zugehörige Software am Raspberry Pi. Auch ein LCD-Display lässt sich verwenden.

Kapitel 6 erklärt die Verwendung einer IR-Fernbedienung am Raspberry Pi. Zusätzlich werden Entfernungen mit preisgünstigen Sensoren für Ultraschall und Infrarot gemessen.

Kapitel 7 bietet Ihnen Grundwissen zu Netzwerken. Wie werden IP-Adressen vergeben? Wie entsteht eine Direktverbindung zwischen dem Raspberry Pi und einem anderen PC? Wie gelingt ein Netzwerkzugriff? Nutzen Sie SSH, PuTTY, TMux und weitere Programme.

In **Kapitel 8** lernen sie ALSA kennen – das Linux-Soundsystem. Dabei geht es weniger darum, aus dem Pi eine Stereoanlage zu machen, als den Minicomputer zum Lauschen und Zuhören zu bringen.

In **Kapitel 9** erfahren Sie, wie Sie Videodateien abspielen und Ihre Webcam nutzen können: zum Fotografieren oder Filmen – auf Wunsch auch ereignisgesteuert.

Kapitel 10 zeigt die Verwendung der Kameramodule zum Raspberry Pi. Fotografieren und filmen Sie bei unterschiedlichen Lichtverhältnissen oder in Kombination mit einer Webcam.

Kapitel 11: Was geschieht mit den gewonnenen Daten? Auf der RAM-Disk ablegen, übertragen oder löschen? Cronjobs helfen bei der Organisation.

Kapitel 12 wendet sich nochmal dem Thema Audio zu, jetzt aber mit C/C++ Programmen, die es ermöglichen, Audiosignale zu analysieren. Lernen Sie Audiotools und Libraries kennen. Benutzen Sie den Raspberry Pi als Papagei, und lassen Sie sich von einem selbst gebauten Audioquiz überraschen.

In **Kapitel 13** lernen Sie die Bildverarbeitung mit OpenCV kennen. Viele Open-Source-Programme werden mit Tools wie CMakeGUI konfiguriert und direkt aus dem C++ Quelltext erzeugt. Bei mehr als 100 OpenCV-Beispielprogrammen ist garantiert etwas Passendes für Sie dabei! Ergänzend wird das ebenfalls von mir verfasste Buch „OpenCV Basiswissen“ empfohlen.

Kapitel 14 erläutert die Objekterkennung mit OpenCV – von der Kontrastspitzung und einfachen Segmentierung von Textzeilen bis hin zur anspruchsvolleren Symbolerkennung. Erleben Sie eine sprechende Lichtschranke!

In **Kapitel 15** wird MenuCtrl als Client-/Server-Software vorgestellt. Schrittweise wird eine Platine aufgebaut, die als Eingabegerät mit LCD-Display verwendet wird. So entsteht eine autarke Hardware und Software (gegebenenfalls für den Akkubetrieb), bei der auf Tastatur, Maus und Monitor verzichtet werden kann.

Nach dem Schlusswort in **Kapitel 16** folgt in **Kapitel 17** eine Übersicht der Programme und Skriptdateien zum Buch.

Hinweise zur Arbeit mit diesem Buch

Die Programme zum Buch, weitere Zusatzmaterialien sowie einen **elektronischen Anhang** mit ergänzende Informationen können Sie sich hier herunterladen: <http://downloads.hanser.de/978-3-446-45342-5>

Käufer der gedruckten Ausgabe dieses Buches erhalten das E-Book kostenfrei zum Buch. Die erste Seite des Buches enthält einen Zugangscode für das E-Book. Das E-Book ist komplett in Farbe und enthält in der EPUB-Version auch ein farblich hervorgehobenes Syntax-Highlighting.

Für den Raspberry Pi hat sich die Kurzbezeichnung Raspi eingebürgert. Wenn in diesem Buch vom Raspberry Pi oder Raspi gesprochen wird, sind damit alle bisher erschienenen Modelle gemeint. Wenn nur vom Pi gesprochen wird, ist sowohl der Raspberry Pi als auch der Banana Pi gemeint – auch wenn dieser Mini-PC nicht den Schwerpunkt des Buches bildet.

Im Buch sind viele Links zu Webseiten angegeben. Autor und Verlag können nicht für die dauerhafte Gültigkeit garantieren. Oft führt eine Suche nach Schlüsselwörtern zur genannten Webseite ebenfalls zu guten Treffern. Manchmal ist auch nur der Server vorübergehend nicht erreichbar.

Autor und Verlag haften nicht für eventuelle Schäden, die durch die Benutzung der im Buch beschriebenen Hardware oder Software entstehen könnten. Die Informationen wurden allerdings mit größtmöglicher Sorgfalt zusammengestellt, und es sind Warnhinweise zum Umgang mit Elektronik und elektrischen Spannungen enthalten.

Danksagungen

So wie das Raspbian-Betriebssystem eine bestimmte Zeit benötigt hat, um zum heutigen Entwicklungsstand zu gelangen, geht es einem manchmal auch als Autor. Deshalb möchte ich ganz besonders Herrn Frank Baur danken, der mich in zahlreichen Fachgesprächen immer wieder auf neue Ideen gebracht hat. Zur grafischen Umsetzung dieses Buches hat Frau Barbara Schmitz mit guten Tipps beigetragen. Meinem Bruder Erhard Eugen danke ich für zahlreiche Tests der Software, die er als völliger Laie auf dem Gebiet der Programmierung erfolgreich gemeistert hat. Ein ganz großer Dank gebührt auch Frau Julia Stepp vom Hanser Verlag, die die Entstehung dieses Buches über einen recht langen Zeitraum geduldig begleitet hat.

1

Die Welt des Raspberry Pi

Als der Bauingenieur Konrad Zuse 1941 den Z3 als ersten programmierbaren Computer mit binärer Gleitkommarechnung und insgesamt 2200 Relais erfand, waren Betriebssysteme noch nicht existent. Bei einem Hauptspeicher aus 1600 Relais dürfte das auch nicht weiter verwundern. Weniger bekannt ist, dass Zuse auch einen A/D-Wandler und erste Prozesssteuerungen erfand.

75 Jahre später arbeiten die meisten Computer fast unbemerkt und immer noch ohne Betriebssystem. Wenn Ihre Waschmaschine Temperatur, Drehzahl und Wasserstand ermittelt, dann leiten einige Sensoren die gemessenen Daten an einen Embedded Computer weiter, der einen vom Hersteller programmierten Mikrocontroller enthält und so eine Spezialaufgabe erfüllt.

Der Arduino ist ein sehr preisgünstiger Minicomputer, der ohne angeschlossene Peripherie allerdings absolut nutzlos ist. Mit angeschlossenen Sensoren und nach dem Transfer eines am PC entwickelten Programms können Sie damit hingegen Messwerte erfassen oder Relais ansteuern. Ein 8-Bit-Mikrocontroller führt das Programm aus, wobei der Arduino weder einen Monitoranschluss noch ein Betriebssystem hat.

Der Raspberry Pi ist im Gegensatz zum Arduino durch die Nutzung von Betriebssystemen viel universeller verwendbar und bindet Sie nicht an eine bestimmte Programmiersprache. Wie bei einem großen PC können Sie einen Bildschirm anschließen, eine Internetverbindung herstellen oder USB-Geräte anstecken. Zusätzlich kann dieser Mini-PC aber auch relativ leicht mit bestimmten Sensoren verbunden werden. Dort, wo ein Laptop zu groß, zu schwer und zu teuer ist, beginnt die Welt des Raspberry Pi.

■ 1.1 Die Modelle des Raspberry Pi

Als im Jahr 2012 der erste Raspberry Pi Computer auf den Markt kam, ahnte wohl noch niemand, dass sich daraus eine ganze Produktfamilie (siehe Bild 1.6) mit bislang mehr als 19 Millionen verkauften Exemplaren (März 2018) entwickeln würde. Alle Raspberry Pi Einplatinen-

computer haben einen recht geringen Energieverbrauch, der von der Prozessorleistung des jeweiligen Modells, aber auch von den angeschlossenen Geräten abhängt.

Kann man für Preise zwischen 11 und 45 € (je nach Modell und Bezugsquelle) auch wirklich einen vollwertigen Computer erwarten? Nicht ganz, denn Sie bekommen nur ein vollständig bestücktes Mainboard. Gehäuse, Netzteil, (micro)SD-Karte, HDMI-Kabel und ein USB-Hub sind nicht enthalten.



Bild 1.1 Ein Raspberry Pi 1 mit PS2/USB-Adapter, passivem USB-Hub, USB-Stick und WLAN-Stick

Was macht den Raspberry Pi so attraktiv? Es ist nicht nur der Preis, sondern vielleicht auch die Möglichkeit der Erweiterung und die Fähigkeit, externe Hardware vergleichsweise einfach anzusteuern. Wofür ist dieser kleine Computer zu empfehlen? Auf jeden Fall für den lautlosen Dauerbetrieb und für viele Arten von Spezialaufgaben, bei denen Tastatur und Monitor entfallen können – aber nicht müssen. Zusammen mit OpenCV können Sie z. B. eine Alarmanlage basteln, bei der verdächtige Bilder drahtlos zu einem anderen Rechner übertragen werden. Sehr oft wird der Raspberry Pi im eigenen Netzwerk eingesetzt und ist als einziger Computer permanent eingeschaltet.

A, A+, B, B+, Pi 2, Pi 3, Pi Zero W – welcher Raspberry Pi ist nun der richtige? Sämtliche lieferbaren Modelle werden auf der Website der Raspberry Pi Foundation beschrieben: <https://www.raspberrypi.org>

Der Raspberry Pi 3 und der ganz neue Pi Zero W sind mit einem WLAN-Chip ausgestattet. Beim Pi Zero W ist das besonders wichtig, denn dieses Minimodell hat keine Ethernet-Buchse. Letztlich hängt die Entscheidung davon ab, welche Rechenleistung und welche Schnittstellen Sie benötigen. Auf den Monitor kann man gegebenenfalls verzichten, wenn Sie den Raspberry Pi über SSH (SecureShell) fernsteuern. Ein WLAN-Stick kann bei allen Modellen über USB angeschlossen werden.

Im Jahr 2013 war der Raspberry Pi (Modell B) wahrscheinlich der weltweit bekannteste Einplatinencomputer mit folgenden Anschlüssen und Eigenschaften:

- Broadcom BMC2835-Prozessor, 700 MHz (ohne Overclock)
- Broadcom VideoCore-4-GPU mit OpenGL ES 2.0, h.264/MPEG-4
- 512 MB Hauptspeicher (DDR2-SDRAM)
- USB 2.0 (leider nur 2-fach)
- Micro-USB zur Spannungsversorgung (min. 700 mA)
- Netzwerkbuchse RJ 45 (Ethernet mit 10/100 MBit/s)
- HDMI-Ausgang (adaptierbar auf DVI)
- Cinchbuchse als analoger FBAS-Videoausgang
- 3,5 mm Klinensteckerbuchse für den Ton (Ausgang)
- CSI-Interface für das Kameramodul
- DSI-Interface für spezielle externe LCD-Displays
- GPIO-Port mit 26 Pins (z. B. für Sensoren, Relais etc.)
- Slot für eine SD-Karte mit Betriebssystem
- Energieverbrauch: 3,5 Watt bei 700 mA

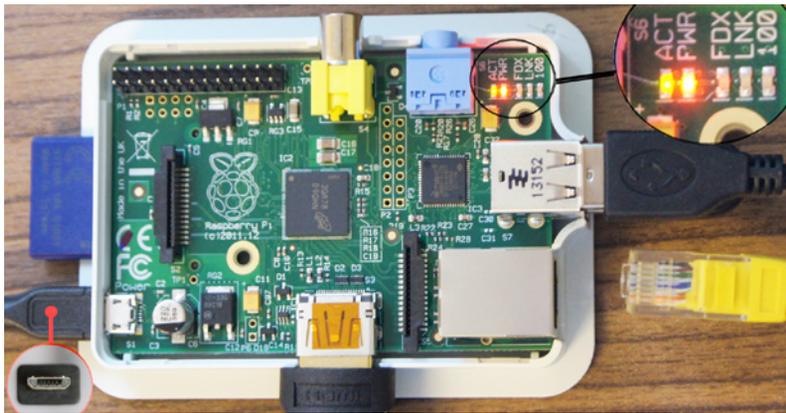


Bild 1.2 Raspberry Pi 1 (Modell B) mit aktiven LEDs für Power und Action

Inzwischen wird dieses weitverbreitete, aber ältere Modell mit SD-Karte als Raspberry Pi 1 (Modell B) bezeichnet (Bild 1.2). Die Modellreihe A ist preisgünstiger und etwas sparsamer ausgestattet. Zunächst geschah das durch Weglassen von Bauteilen auf der Platine. Die A+ Modellreihe (Bild 1.3) hat dagegen eine kleinere, fast quadratische Platine.

Die älteren Modelle A und B des Raspberry Pi (vor Juni 2014) werden nur noch selten als Neeware angeboten, haben aber immer noch einen teilweise erstaunlichen Sammlerwert. Bei beiden Modellen wird eine SD-Karte zum Booten des Betriebssystems benötigt. Die neueren Modelle (A+, B+, Zero, Pi 2, Pi 3) verwenden stattdessen eine microSD-Karte (Bild 1.4).

Beachten Sie bitte auch, dass es unterschiedliche Gehäuseformen gibt. Das Gehäuse aus Bild 1.2 ist für lediglich zwei USB-Buchsen vorgesehen, während die neueren Modelle der Reihe B über vier USB-Buchsen verfügen, wie in Bild 1.5 zu sehen. Nicht alle Gehäusehersteller haben Öffnungen für das Displaymodul und das Kameramodul vorgesehen. Die seriellen Anschlüsse dazu werden als DSI und CSI bezeichnet.

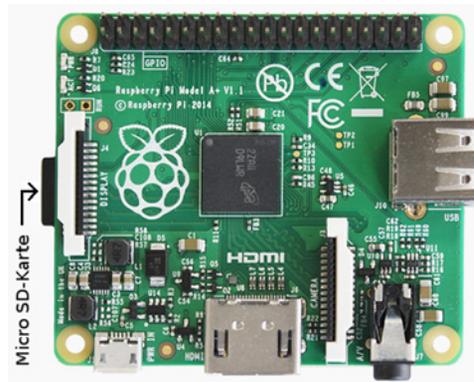


Bild 1.3 Das Modell A+ des Raspberry Pi 1 (© Bild: Raspberry Pi Foundation)

Ein signifikantes Merkmal aller Raspberry Pi Computer ist eine Stiftleiste, die als GPIO-Port bezeichnet wird. In Bild 1.2 sind das $2 \times 13 = 26$ Kontakte für den GPIO-Port. Es gibt jede Menge Hardware, die über General Purpose Input/Output (GPIO) anschließbar ist. Da die neueren Raspberry Pi Modelle 40 Kontakte haben, passt die eine oder andere Aufsteckplatine nur über einen zwischengeschalteten Adapter. Wie Sie elektronische Schaltungen über GPIO ansteuern, erfahren Sie in Kapitel 5.

	Raspberry Pi				
	B+	Zero W	2B	3B	3 B+
Prozessor	BMC2835		BMC2836	BMC2837	BMC2837B0
Architektur ^{Bit}	ARMv6 ³²	ARMv7 ³²	ARMv8 ⁶⁴		
CPU-Takt (MHz) ^{Co}	700 [•]	1000 [•]	900 ^{••}	1200 ^{•••}	1400 ^{•••}
GPU	VideoCore IV mit OpenGL ES 2.0, H264 / MPEG-4				
Hauptspeicher	512 MB		1024 MB		
USB-Ports (Anzahl)	4	1	4		
LAN (maximal)	100 MBit/s	—	100 MBit/s	300 MBit/s	
WLAN-Standard	—	2,4 GHz b/g/n	—	2,4 GHz b/g/n	2,4/5 GHz, ac
Audio-Out	Klinkenstecker	nur	Klinkenstecker		
Video-Out	4-polig	Kontakte	4-polig		
erhältlich seit	6/2014	2/2017	2/2015	2/2016	3/2018

Bild 1.4 Die neueren Modelle des Raspberry Pi (ab Juni 2014)

Der Raspberry Pi B+ ist ein Nachfolger des allerersten Raspberry Pi B Modells und kostet derzeit ca. 25 €. Der Hauptunterschied zum älteren Modell (Bild 1.2) liegt in einer 40-poligen Stiftleiste und in der nun 4-poligen Klinkensteckerbuchse. Es spricht aber aus meiner Sicht nur

noch wenig für dieses Modell, das von „unten“ und von „oben“ Konkurrenz aus dem eigenen Haus bekommen hat. Aber wer nicht viel Geld ausgeben möchte und eine Ethernet-Buchse braucht, ist mit diesem Modell gut bedient. Die Anschlüsse stimmen mit den wesentlich schnelleren Modellen Pi 2 und Pi 3 überein.

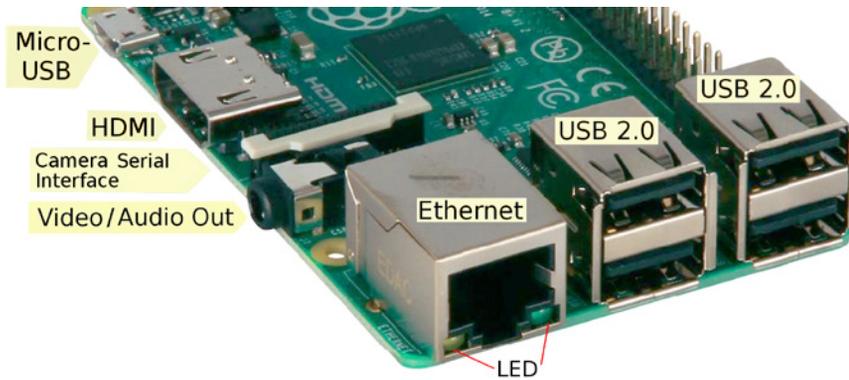


Bild 1.5 Anschlüsse am Modell B+, dem Raspberry Pi 2 und dem Pi 3

Obwohl Sie in Bild 1.5 vier USB-Buchsen sehen, hat sich in diesem Punkt nicht so viel geändert. Es gibt seit 2012 einen gemeinsamen Controller für LAN und USB, was dazu führen kann, dass sich die Geräte gegenseitig ausbremsen. Aber es bleibt die Hoffnung, dass ein noch nicht existenter Raspberry Pi 4 endlich auch USB 3.0 unterstützen könnte und auch getrennte Controller mitbringt.



Bild 1.6 Produkte der Raspberry Pi Foundation (Stand: Juni 2016)

1.1.1 Energieversorgung

Das am Raspberry Pi über die Micro-USB Buchse angeschlossene 5 Volt Netzteil sollte mindestens 1000 mA liefern können. Ein derartiges Netzteil reicht für die Raspberry Pi Modelle Zero, A, A+, B, B+ und knapp auch für den Pi 2 aus.

Dann kann es aber immer noch z.B. bei Festplatten kritisch werden, die keine eigene Spannungsversorgung haben und eine Energiezufuhr über USB erwarten. Es ist besser, wenn Sie von Anfang an ein Netzteil mit minimal 2000 mA und Micro-USB Kabel verwenden, das dann auch zum Banana Pi passen würde. Beachten Sie bitte, daß der Raspberry Pi 3 ein stärkeres Netzteil (2500 mA) als die Vorgängermodelle braucht. Nicht nur Festplatten konsumieren vergleichsweise viel Energie. Auch manche WLAN-Sticks und Soundkarten können nur über einen **aktiven** USB-Hub (siehe Bild 9.11) am Raspberry Pi betrieben werden.

In Bild 1.1 wird der Raspberry Pi 1 mit einem schwachen Netzteil und mit passivem USB-Hub betrieben. Würden Sie den WLAN-Stick erst im laufenden Betrieb anstecken, dann brähe die Spannung kurzzeitig zusammen, was zum erzwungenen Reboot führen würde. Es kommt also auch darauf an, welche Last ein Peripheriegerät beim Anstecken verursacht.

1.1.2 Der Raspberry Pi Zero W

Der Raspberry Pi Zero ist der kleinste und preisgünstigste Raspi (so der Spitzname) und wurde bislang in drei Varianten mit denselben Abmessungen (65 * 30 * 5 mm) gebaut:

Pi Zero (ohne Kameraanschluss)	ab 11/2015
Pi Zero (mit Kameraanschluss)	ab 5/2016
Pi Zero W (Kameraanschluss + WLAN)	ab 2/2017

Für die aktuelle Variante (ab 28.02.2017) wurde erstmals eine höhere Stückzahl eingeplant, und es gibt inzwischen pi3g als Distributor für den deutschsprachigen Raum. Der Pi Zero W kann unter www.buyzero.de (gegebenenfalls mit Zubehör) bezogen werden. Je nach mitgeliefertem Zubehör liegt der Preis zwischen 11 € und ca. 30 €.

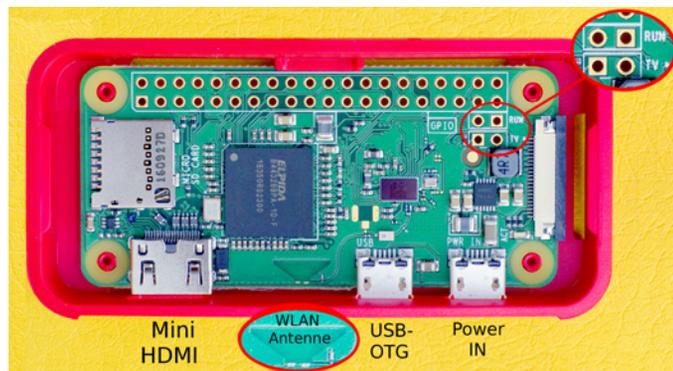


Bild 1.7 Der Raspberry Pi Zero W im Gehäuseunterteil

Der Preis hängt auch davon ab, welche Adapter(kabel) Sie benötigen. Der bei den größeren Modellen verwendete HDMI-Anschluss wurde aus Platzgründen durch Mini-HDMI ersetzt, ebenso USB durch Micro-USB. Kennzeichnungen am Gehäuseunterteil helfen, die Micro-USB-Buchsen voneinander zu unterscheiden (Bild 1.7). Auf die Verwendung von USB-OTG (On The Go) wird in Abschnitt 7.19 genauer eingegangen.



Bild 1.8 Adapter(kabel) für USB und HDMI

Die Raspberry Pi Foundation hat auch ein Gehäuseset entwickelt, das die in Bild 1.9 abgebildeten Teile (ohne den Pi Zero W) beinhaltet.

Der Pi Zero hält dabei ohne Schrauben im roten Gehäuseunterteil, muss aber (mit microSD-Karte) so eingesetzt werden, dass keine der Miniaturbuchsen in das Gehäuse hineinragt. Die gegenüberliegende Seite dient dabei quasi als Drehachse, wie in Bild 1.9 zu sehen.

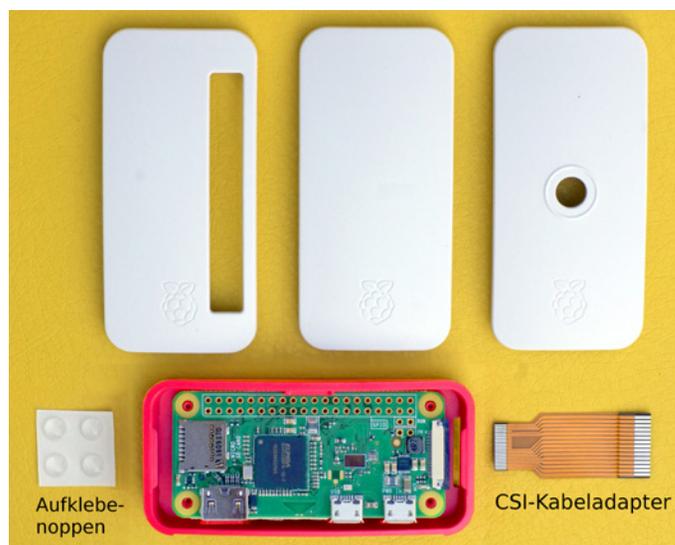


Bild 1.9 Raspberry Pi Zero W mit dem offiziellen Gehäuseset

Für den Kameraanschluss (siehe Bild 10.7) wird ein schmäleres Kabel als bei den großen Raspberry Pi Modellen benötigt. Das über pi3g erhältliche Gehäuseset enthält nicht nur diesen CSI-Kabel-Adapter, sondern neben dem roten Gehäuseunterteil auch drei Abdeckungen zur Auswahl. Die links in Bild 1.9 abgebildete Abdeckung enthält eine Öffnung für die GPIO-Stiftleiste. Die runde Öffnung im rechten Gehäuseoberteil ist für das Kameramodul vorgesehen.

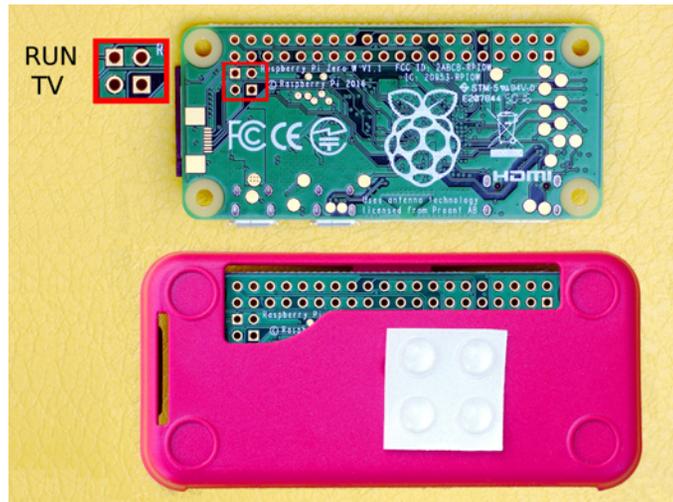


Bild 1.10 Platinenrückseite und Gehäuseboden mit zugänglichen Kontakten

Am auffälligsten sind am Raspberry Pi Zero vielleicht die Bohrungen für eine 40-polige Stiftleiste, deren Kontakte 2,54 mm Abstand zueinander haben. Die gesamte Platine ist nur 5 mm hoch, da die GPIO-Stiftleiste nicht eingelötet wurde und auch die Netzwerkbuchse fehlt. Der DSI-Anschluss für das Raspberry Pi Touchdisplay ist beim Pi Zero entfallen.

Der Zugang zur GPIO-Schnittstelle ist auch von der Gehäuseunterseite her möglich. So können Sie an den in den Bildern 1.7 und 1.10 vergrößert dargestellten Kontakten Drähte anlöten.

- Die Platinenbeschriftung „Run“ bezieht sich auf eine Reset-Möglichkeit durch Verbindung der beiden Kontakte.
- Die mit „TV“ beschrifteten durchkontaktierten Bohrungen liefern ein Composite Video-Signal, denn eine Klinkensteckerbuchse gibt es nicht.
- Eine bootfähige microSD-Karte muss vor dem Einbau in das Gehäuse eingesteckt werden. Es gibt keine Öffnung zum Wechsel der microSD-Karte.
- Die selbstklebenden Gumminoppen für den Gehäuseboden sind in Bild 1.10 noch nicht aufgeklebt.

Immerhin liegt die Prozessorleistung des Pi Zero etwa 40 % über der des Raspberry Pi 1 aus dem Jahr 2012. Dabei ist der Raspberry Pi Zero so klein, dass ein Einbau in nahezu jedes Gehäuse möglich erscheint.

Die im Vergleich zur A+ oder B+ Reihe weggefallenen Komponenten erleichtern den Akkubetrieb, wobei die Energieversorgung der HDMI-Buchse bei allen Modellen per Software abschaltbar ist (siehe Abschnitt 4.4.2).

Der Raspberry Pi Zero ist eher ein Bastelcomputer als ein klassischer Desktop-PC, für den die beiden Adapter(kabel) benötigt werden. Trotzdem sehen Sie in Bild 1.11 eine derartige Verkabelung, die Sie mit Bild 1.1 vergleichen können.

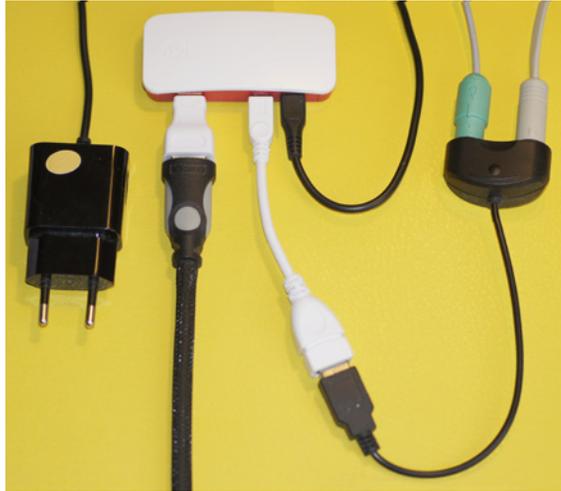


Bild 1.11 Verkabelung eines Pi Zero als Desktop-PC

Wer mehr als einen Pi-Zero-Computer hat, kann die Computer übereinander anordnen und z. B. über die serielle Schnittstelle miteinander vernetzen. Die nicht eingelötete Stiftleiste reduziert die Höhe eines solchen Arrays. Der Pi Zero ist als einzelne Platine viel zu leistungsschwach, um als dauerhafter Ersatz für einen älteren, vergleichsweise energiehungrigen Laptop zu dienen.

Für den Pi Zero W gelten folgende Daten:

- 1 GHz Single-Core-CPU
- 512 MB RAM
- Mini-HDMI-Port
- Micro-USB-On-the-Go-Port
- Micro-USB-Power
- HAT-kompatibler 40-Pin-Header
- Composite Video und Reset Header
- CSI Camera Connector
- 802.11n WLAN/Bluetooth v4.1/Bluetooth Low Energy
- Bluetooth 4.0

Man kann sich aber recht gut vorstellen, Tastatur und Maus über Bluetooth mit dem Pi Zero W zu verbinden. Am USB-Port kann dann z. B. ein USB-Stick als zusätzlicher Datenspeicher hängen. Bei bestimmten USB-Geräten (z. B. Soundkarten) werden Sie auch am Pi Zero W einen aktiven USB-Hub benötigen.

Benötigen Sie vielleicht auch noch einen Akku für mehr Mobilität? Alex Eames hat den Energiebedarf des Pi Zero W gemessen und die Daten auf folgender Webseite bereitgestellt: <http://raspi.tv/2017/how-much-power-does-pi-zero-w-use>