#makers
DO IT.

PLUS.HANSER-FACHBUCH.DE

Jörn Donges

Mach was mit Python & Raspberry Pi!

Spielerisch programmieren lernen
und mit DIY-Projekten durchstarten

E-Book
EXTRA
E-Book inside

HANSER

HANSER

Jörn Donges

Mach was mit Python & Raspberry Pi!

Spielerisch programmieren lernen und mit DIY-Projekten durchstarten

Ihr Plus – digitale Zusatzinhalte!

Auf unserem Download-Portal finden Sie zu diesem Titel kostenloses Zusatzmaterial.

Geben Sie auf **plus.hanser-fachbuch.de** einfach diesen Code ein:

plus-me459-11ca6

Der Autor:
Jörn Donges, Hamburg

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.

www.hanser-fachbuch.de
Lektorat: Julia Stepp
Herstellung: Björn Gallinge
Coverkonzept: Marc Müller-Bremer, www.rebranding.de, München
Titelgrafik: © shutterstock.com/goodcat, Teguh Mujiono und AlesiaKan
Coverrealisation: Max Kostopoulos

Print-ISBN: 978-3-446-46150-5
E-Book-ISBN: 978-3-446-46600-5

E-Pub-ISBN: 978-3-446-46601-2

Inhalt

1 Einführung

Mit dem Raspberry Pi (auch Pi oder RasPi genannt) bekommt man für wenig Geld ein vollwertiges Computersystem mitsamt reichhaltiger Softwarekollektion, die das Arbeiten vom spielerischen Einstieg bis hin zur Softwareentwicklung auf hohem Niveau ermöglicht.

Aufgrund der stetig zunehmenden Digitalisierung aller Lebensbereiche wird es immer wichtiger, die Konzepte der digitalen Informationsverarbeitung zu verstehen. Nur wer die Arbeitsweise von programmierbaren Systemen und Algorithmen versteht, kann auch realistisch einschätzen, wo deren Grenzen liegen. Darüber hinaus macht es einfach Spaß, eigene Ideen umzusetzen und zum Mitgestalter der digitalen Welt zu werden, anstatt sie nur zu konsumieren.

Dafür stellt der Raspberry Pi den idealen Einstieg dar. Der kleine Einplatinenrechner hat seit seiner Einführung im Jahr 2012 eine riesige Verbreitung gefunden. Über 20 Millionen Exemplare wurden bereits verkauft. Zu den Nutzern zählen nicht nur Bastler und Maker, sondern insbesondere auch Softwaretüftler und angehende Entwickler, denen der Raspberry Pi eine preiswerte Möglichkeit bietet, ein Linux-System einzurichten und damit zu experimentieren oder ein System lautlos und stromsparend im

Dauerbetrieb laufen zu lassen, etwa als Homeserver oder als Medienzentrale.

Auch die Programmiersprache Python erfreut sich einer immer größer werdenden Community von Entwicklern und Anwendern, da die Sprache leicht erlernbar und verständlich ist, andererseits aber umfangreiche Bibliotheken für nahezu jeden erdenklichen Anwendungszweck bietet und daher universell einsetzbar ist. Was liegt also näher, als die beiden Welten miteinander zu verbinden?

Die Mission dieses Buches

Dieses Buch bietet eine Einführung in den Umgang mit dem Raspberry Pi und seinem bevorzugten Betriebssystem Linux (in der Raspbian-Distribution) sowie in die Programmiersprache Python 3.

Du brauchst keine Vorkenntnisse. Es wird alles von Grund auf erklärt. Dennoch habe ich versucht, einen breiten Bereich typischer Anwendungen abzudecken, wie etwa Spieleprogrammierung, GUI-Entwicklung oder Web Apps.

Bei der Auswahl der Beispielprojekte habe ich darauf geachtet, dass es sich um brauchbare kleine Anwendungen handelt, die du auch im Alltag benutzen kannst. Gleichzeitig ist der Code aber möglichst kurz gehalten, um für die Veröffentlichung in einem Buch geeignet zu sein. Daher handelt es sich immer um Basisversionen der entsprechenden Anwendungen, die jederzeit mit weiteren Features ausgebaut und erweitert werden können. Ich möchte dich genau dazu in die Lage versetzen. Die Beispielprojekte sind also nicht zum sturen Abtippen gedacht,

sondern eher als Ideengeber, die dich dazu bringen sollen, selbst aktiv zu werden und an den Programmen weiterzuarbeiten.

Die Projekte sollen nicht in Stein gemeißelt sein, sondern zum Experimentieren und Probieren einladen. Die in diesem Buch angeschnittenen Themenbereiche sind in sich jeweils so komplex und umfangreich, dass sie nicht erschöpfend dargestellt werden können. Nimm also nicht nur dieses Buch als Anleitung zur Hand, sondern nutze bei Veränderungsideen und Verständnisschwierigkeiten auch Google oder die Coder-Hilfsplattform *https://stackoverflow.com*, um nach entsprechenden Begriffen zu suchen, denn du wirst schnell feststellen, dass eine Anwendung nie wirklich fertig ist. Es gibt immer Möglichkeiten der Weiterentwicklung. Vielleicht tauchen auch nach langem Einsatz noch Fehler auf, die behoben werden müssen.

An einigen Stellen im Buch setze ich voraus, dass du englischsprachige Texte verstehen kannst. Viele Begriffe in der Programmierung sind auf Englisch und bei fortgeschrittenen Themen wie API-Programmierung stehen oft auch keine deutschsprachigen Schnittstellen und Dokumentationen zur Verfügung. Daher war das nicht immer zu umgehen, doch wenn du tiefer in die IT-Welt einsteigen willst, wirst du wahrscheinlich schon wissen, dass man nicht darum herumkommt, Englisch zu lernen.

So ist dieses Buch aufgebaut

In Kapitel 2 zeige ich dir, wie du deinen Raspberry Pi einrichtest und welche Tools für die Python-Programmierung benötigt

werden. Du lernst auch die Grundlagen der Linux-Distribution Raspbian kennen, die standardmäßig auf dem RasPi installiert wird.

Mit einer Raspbian-Installation steht dir ein riesiges Softwarepaket kostenlos zur Verfügung, und du kannst gleich mit der Python-Programmierung loslegen. Für fortgeschrittene Projekte musst du manchmal ein externes Modul oder ein neues Programm installieren. Auch das erkläre ich dir alles genau. Du erfährst, wie du über das Paketverwaltungssystem apt ganze Softwarepakete nachinstallieren kannst, aber auch, wie du mit dem Python Package Index gezielt eines der unzähligen Python-Module finden, herunterladen und einbinden kannst.

Es gibt verschiedene Möglichkeiten, auf dem kleinen Rechner zu arbeiten. Du kannst entweder klassisch Tastatur, Monitor und Maus anschließen und wie an einem herkömmlichen Desktop-PC arbeiten. Es ist aber auch möglich, über eine Netzverbindung auf den RasPi zuzugreifen und die Benutzeroberfläche auf einen anderen Computer zu holen. Auch das wird ausführlich erklärt.

In Kapitel 3 führe ich dich in die Grundlagen der Python-Programmierung ein. Anhand eines kleinen Zahlenratespiels stelle ich dir die grundlegenden Programmierkonzepte wie Abfragen, Schleifen, Funktionen und Variablen vor. Außerdem werfen wir einen Blick auf den Debugger, ein wichtiges Werkzeug, um Programmierfehlern auf die Spur zu kommen. Die nächste Stufe ist dann die objektorientierte Programmierung. Diese ist für die Entwicklung moderner Apps notwendig. Das erste Projekt wird ein Vokabeltrainer sein, der dich in einer beliebigen Sprache abfragen kann, für die du eine Textdatei mit den Vokabeln hast.

In Kapitel 4 werfen wir einen Blick auf das weite Feld der Spieleprogrammierung. Mit Pygame ist im Standardumfang von

Python auf dem RasPi eine Multimedia-Bibliothek vorhanden, die dir ermöglicht, Grafik und Sound in deine Projekte einzubinden. Das ist ein Thema, das ganze Bücher füllen könnte, deshalb wird im Rahmen dieses Buches nur ein kurzer Einblick möglich sein. Ähnliches gilt für alle weiteren Projekte. Die Projekte können immer nur eine erste Annäherung ans Thema sein. Dennoch wirst du in diesem Kapitel lernen, einfache Grafiken und Animationen zu programmieren und Audiodateien abzuspielen. Am Ende wirst du einen Klassiker unter den Videospielen nachprogrammiert haben, wobei ich die Gelegenheit nutze und die Einführung in die objektorientierte Programmierung nochmals vertiefe.

Zentrales Thema von Kapitel 5 ist die Entwicklung grafischer Benutzeroberflächen. Mit Tkinter stellt Python eine Bibliothek zur Verfügung, die das Erstellen grafischer Benutzerschnittstellen mit Textfeldern, Menüs und Schaltflächen stark vereinfacht. In Kapitel 5 wirst du dem Vokabeltrainer aus Kapitel 3 eine grafische Oberfläche verpassen, um die Anwendung komfortabler zu gestalten und eine Taschenrechner-App für den Desktop entwickeln. Nach dem Durcharbeiten des Kapitels solltest du in der Lage sein, eigene grafische Benutzeroberflächen zu gestalten und umzusetzen und damit deinen eigenen Projektideen eine Oberfläche zu geben.

Neben grafischen Oberflächen sind Web Apps heutzutage eine wichtige Schnittstelle, um Anwendungen an den Benutzer zu bringen. Der Vorteil: Jedes internetfähige Gerät mit einem Webbrowser wird zum Client. Mit Python und dem Modul Flask wirst du in Kapitel 6 lernen, Web Apps zu erstellen. Als praktisches Beispiel entwickelst du eine To-do-App für deinen Raspberry Pi, die auch zur Einkaufsliste oder App für kurze Notizen erweitert werden kann.

Eine Anwendung davon bietet dann Kapitel 7: Hier bauen wir ein Webradio. Dieses wird sowohl als Web App als auch als GUI umgesetzt, sodass du den Radiostream nicht nur vom Raspberry Pi aus steuern kannst, sondern auch per Browser von anderen Geräten in deinem Heimnetzwerk. Dann musst du nur noch einen Lautsprecher anschließen und die Reise in die Welt der weltweiten Audiostreams kann losgehen.

Kapitel 8 greift schließlich ein Thema auf, das du in wenigen anderen Python-Büchern finden wirst. Doch gerade für den Raspberry Pi ist es relevant, weil dieser kleine Rechner gerne im Dauerbetrieb benutzt wird, um Ereignisse im Internet zu protokollieren und Daten zu sammeln, wie z. B. Sportergebnisse, Börsenkurse, Wetter usw. Du wirst lernen, wie man mittels Web Scraping interessante Daten automatisch aus Webseiten extrahiert und wie man mithilfe einer API einen Server kontaktiert, um bestimmte Daten herunterzuladen und sie weiter auszuwerten.

Die Beispielprojekte aus dem Buch in der Übersicht	
Kapitel 3	▪ Zahlenratespiel ▪ Vokabeltrainer (Konsole)
Kapitel 4	▪ Analoguhr ▪ Pong
Kapitel 5	▪ Vokabeltrainer (GUI) ▪ Taschenrechner

Kapitel 6	▪ To-do-App
Kapitel 7	▪ Webradio (Web) ▪ Webradio (GUI)
Kapitel 8	▪ Wetter-App (Konsole) ▪ Wetter-App (Web)

Sämtliche Quellcodes zu den Beispielprojekten stehen in ungekürzter Form auf *plus.hanser-fachbuch.de* zur Verwendung bereit.

Geben Sie dazu einfach diesen Code ein:
plus-me459-11ca6

Ansonsten bleibt mir an dieser Stelle nur, dir viel Spaß auf der spannenden Reise durch die Möglichkeiten der Sprache Python und des kleinen Raspberry Pi zu wünschen.

Hamburg, August 2020

Jörn Donges

2 Den Raspberry Pi und Python zum Laufen bringen

In diesem Kapitel bereiten wir alles Nötige vor, um mit dem Programmieren zu beginnen. Wir werden uns erst einmal einen Überblick verschaffen, was du an Zubehör brauchst, welche Modelle des Raspberry Pi es gibt und worin sie sich unterscheiden. Danach zeige ich dir, wie du deinem RasPi das richtige Betriebssystem verpasst und ihn damit zum Leben erweckst. Normalerweise sind keine zusätzlichen Installationen notwendig, da alle Python-Werkzeuge bei der Standardinstallation bereits an Bord sind, doch für den Fall, dass du Python in einer anderen Linux-Umgebung einrichten möchtest, gehe ich in Abschnitt 2.5 auch darauf ein.

2.1 Was brauchen wir alles?

Der Raspberry Pi ist als leiser, stromsparender Rechner gut dafür geeignet, Langzeitaufgaben zu übernehmen. Er fällt dabei kaum auf, wenn er seinen Dienst verrichtet, weil er selbst im Gehäuse nicht größer ist als eine Schachtel Zigaretten. Dennoch kommt er natürlich ohne gewisse Verbindungen zur Außenwelt nicht aus, und sei es nur, um die Programme darauf einzurichten und ihn dann später alleine laufen zu lassen. Du brauchst also erst einmal eine Tastatur und einen Bildschirm. Es ist auch möglich, den RasPi ohne Tastatur und Bildschirm über eine Netzwerkverbindung von einem anderen Rechner aus zu bedienen. Dieser andere Rechner kann sich entweder in deinem Heimnetzwerk befinden oder irgendwo auf der Welt, solange Rechner und RasPi beide über eine Internetverbindung verfügen. Vielleicht kennst du das Remotedesktop-Programm unter Windows. Es beruht auf demselben Prinzip. Wir greifen dann über einen sogenannten SSH-Server (SSH für Secure Shell) auf den Raspberry Pi zu, der einen sicheren und verschlüsselten Datenübertragungskanal zur Verfügung stellt.

Für die ersten Versuche empfiehlt es sich aber, die Peripheriegeräte direkt anzuschließen. Um loszulegen, brauchst du Folgendes:

- ein beliebiges Modell des Raspberry Pi
- ein USB-Netzteil oder eine Powerbank mit passendem Micro-USB-Stecker (Typ C ab Modell 4, Typ B für ältere Modelle)
- eine SD-Karte mit vorinstalliertem Betriebssystem
- eine Tastatur mit USB-Anschluss (Typ A)
- einen Bildschirm mit HDMI-Anschluss
- einen WLAN- oder Ethernet-Anschluss (bei Modellen kleiner 3B, nur Ethernet)

Bei all diesen Dingen handelt es sich – bis auf den RasPi selbst – um Geräte, die du wahrscheinlich schon zu Hause hast. Du kannst z. B. das Ladegerät eines Smartphones nutzen oder auch den RasPi über den USB-Anschluss eines PCs oder Laptops mit Strom versorgen.

2.2 Ein Blick auf die Hardware

Der Raspberry Pi ist ein Einplatinencomputer, der auf der sogenannten ARM-Prozessorarchitektur basiert. Er enthält alle notwendigen Komponenten für ein vollständiges Computersystem, untergebracht auf einer einzelnen Platine von der ungefähren Fläche einer Kreditkarte. Die britische Raspberry Pi Foundation brachte das erste Modell im Jahr 2012 auf den Markt, um experimentierfreudigen Menschen einen preiswerten Zugang zum Erlernen von Hardware- und Programmierfähigkeiten zu bieten. Für einen Preis von etwa 40 € war es vorher nicht möglich, einen universell einsetzbaren Computer zu bekommen, der einerseits dazu einlädt, mit Elektronik und Hardware zu experimentieren, andererseits für einfachere Heimanwendungen einen brauchbaren Ersatz für einen vollwertigen Personal Computer bietet. Gerade die neuen Raspberry-Pi-4-Modelle eignen sich durchaus, um damit für den Hausgebrauch Office-Anwendungen laufen zu lassen.

Bild 2.1 Der Raspberry Pi 3 (Quelle: Pixabay, User: tw19831113, *https://pixabay.com/de/photos/ger%C3%A4t-der-raspberry-pi-pc-3438525*)

Einplatinenrechner versus Mikrocontroller

Es gab vor dem ersten Release des Raspberry Pi zwar bereits Mikrocontroller, die als eine programmierbare Steuerzentrale für Schaltungen und Hardwareprojekte eingesetzt werden konnten, die aber nicht dieselbe Flexibilität und universelle Einsetzbarkeit hatten. Dazu zählt zum Beispiel der Arduino als leicht zugänglicher Mikrocontroller, der einen spielenden Einstieg in die Welt der Elektronik und Steuerung ermöglicht (siehe hierzu auch das von Robert Jänisch und mir veröffentlichte Buch *Mach was mit Arduino!,* ISBN 978-3-446-45128-5).

Der wichtigste Unterschied zwischen den beiden ist, dass der Raspberry Pi über ein Betriebssystem verfügt. Ganz wie du es von anderen Rechnern her gewohnt bist, kannst du mit dem RasPi

verschiedene Anwendungen (Apps) starten, Dateien bearbeiten, Peripheriegeräte wie Tastatur und Monitor anschließen usw. Es ist dadurch ein viel größerer Anwendungsbereich für das Gerät möglich. Der Arduino als Mikrocontroller ist dagegen eher ein Spezialist. Er wird genau für die Aufgabe programmiert, die er in einem Projekt verrichten soll. Inzwischen hast du die Auswahl zwischen vier verschiedenen Modellen des Raspberry Pi.

Der Raspberry Pi Zero

Der Raspberry Pi Zero ist das kleinste Modell der Raspberry-Pi-Reihe und macht daher am ehesten den klassischen Mikrocontrollern Konkurrenz. Auf einer Platine, die etwa halb so groß wie eine Scheckkarte ist, findest du Anschlussmöglichkeiten für Tastatur und Monitor sowie eine 1 GHz CPU und 512 MB Arbeitsspeicher. Der Zero ist bereits ab 10 € erhältlich. Dadurch ist er gut geeignet, um seinen Dienst als Steuerzentrale für Roboterfahrzeuge oder andere elektronische Gadgets zu leisten. Mit dem Raspberry Pi Zero W ist auch ein Modell mit WLAN-Chip und Bluetooth-Konnektivität erhältlich.

Der Raspberry Pi 1

Der Raspberry Pi 1 ist die erste Generation. Dier ersten Modelle, die 2012 auf den Markt kamen, trugen die Bezeichnungen B und C (nachdem A für einen internen Prototyp verwendet worden war). Im Jahr 2014 änderte die Raspberry Pi Foundation diese alten Modellbezeichnungen und fasste sie unter der Reihe Raspberry Pi 1 zusammen. Sie sind noch im Handel erhältlich,

kosten aber in etwa genauso viel wie die aktuellen 2er und 3er Modelle, sodass für Einsteiger kein Grund besteht, heute noch darauf zurückzugreifen.

Der Raspberry Pi 3

Der Raspberry Pi 3 und auch die erweiterte Version, der 3+, sind Allrounder mit einem günstigen Preis-Leistungs-Verhältnis. Sie bieten einen eingebauten WLAN-Chip, sodass du kein Ethernet-Kabel mehr brauchst, um den RasPi mit dem Netz zu verbinden. Außerdem ist dieses Modell auch das erste, das neben Linux auch mit der Windows IoT-Edition betrieben kann. Darauf werde ich in diesem Buch aber nicht weiter eingehen.

Der Raspberry Pi 4

Zum Entstehungszeitpunkt dieses Buches ist der Raspberry Pi 4 das neueste und leistungsfähigste Modell der Reihe. Mit seiner Rechenleistung eignet sich das Modell durchaus schon als Desktop-Computer, den du zum Surfen, für Office-Arbeiten und zum Programmieren benutzen kannst.

Welcher RasPi für dieses Buch?

Alle vorgestellten Modelle kannst du mit der gleichen Installation des Betriebssystems betreiben. Um die Beispielcodes aus diesem Buch auszuführen, sind also alle Modelle gleich gut geeignet. Der

einzige Unterschied ergibt sich in der Geschwindigkeit und der Laufzeit, gerade bei komplexeren Programmen. Wenn du vorhast, anspruchsvollere Anwendungen zu programmieren und den Pi vielleicht auch als Medienzentrale für Audio/Video nutzen möchtest, dann solltest du das leistungsfähigere Modell nehmen. Im Moment wäre das also der Raspberry Pi 4, jedoch hast du auch mit dem Pi 3+ einen guten Allrounder zur Hand. Und wer sagt denn, dass du nur einen einzigen Raspberry haben wirst? Es ist ja gerade Teil des Konzepts, dass die Modelle günstig zu haben sind und man daher nicht so vorsichtig sein muss.

2.3 Raspbian – das Mini- Linux für deinen Raspberry Pi

Wenn du einen PC oder Laptop einschaltest, dann folgt erst einmal ein mehr oder weniger langer Startvorgang, das so genannte „Booten“. Das heißt, der Rechner lädt zunächst das Betriebssystem von der Festplatte in den Arbeitsspeicher. Das Betriebssystem kümmert sich um alle anfallenden Aufgaben, die bei der Arbeit mit dem Rechner anfallen. Es liest Tastatureingaben, regelt die Bildschirmausgabe, verwaltet Dateien und Programme. Du kennst wahrscheinlich Windows als das am weitesten verbreitete Betriebssystem für PCs, aber auch Android und iOS für mobile Geräte sind dir sicher bekannt. Vielleicht hast du auch schon von Linux gehört, einem freien und offenen Betriebssystem für PCs, das in vielen Varianten erhältlich ist.

Daran siehst du schon, dass ein Betriebssystem eine recht komplexe Angelegenheit ist. Der Hauptunterschied zwischen

einem einfachen Mikrocontroller und einem Rechner wie dem Raspberry Pi ist ja, dass der Controller nicht über ein Betriebssystem verfügt und deshalb nur ein einziges, speziell für ihn entwickeltes Programm ausführt. Die Flexibilität eines universellen Computersystems kommt erst durch das Betriebssystem zustande.

Wenn du deinen neuen RasPi aus dem Karton auspackst, dann kann er zunächst mal noch nichts, weil genau diese Komponente noch fehlt. Im Unterschied zum PC bootet der Raspberry nämlich das Betriebssystem nicht bei jedem Einschalten, sondern es muss einmalig von einem Datenspeicher geladen werden. Wenn dies geschehen ist, dann kann der Raspberry direkt nach dem Einschalten loslegen, muss also nicht zuerst einen Bootvorgang abwarten.

Allerdings besitzt dein RasPi keine Festplatte. Stattdessen findet der Datenaustausch über eine SD-Karte statt. Um die SD-Karte vorzubereiten, benötigst du einen Rechner mit SD-Karten-Leser. Heute ist praktisch jeder Laptop damit ausgerüstet, sodass dies kein Problem sein sollte. Falls du keinen hast, kannst du im Elektronikfachhandel einen SD-Leser für den USB-Port kaufen. Die SD-Karte ersetzt beim Raspberry Pi die Festplatte, was für die kompakte Bauweise des Raspberry einige Vorteile hat, denn die Karte ist leicht, ein Zugriff verbraucht sehr wenig Strom und außerdem kommt sie komplett ohne bewegliche Teile aus.

Bevor du also mit deinem neuen RasPi loslegen kannst, musst du zuerst das Betriebssystem installieren. Normalerweise wird der RasPi mit einem System namens Raspbian betrieben. Dies ist ein Ableger des bekannten Linux-Betriebssystems. Falls du bisher ausschließlich mit Mac oder Windows-Computern gearbeitet hast, dann wird dir auffallen, dass bei Linux einige Dinge anders

sind, doch keine Sorge, wir werden die Installation Schritt für Schritt gemeinsam durchführen. Außerdem kann nie schaden, ein paar Linux-Kenntnisse zu haben. Linux ist immer noch eines der stabilsten Betriebssysteme und wird daher auch im Profibereich gerne für leistungsfähige Server in Rechenzentren eingesetzt.

Ein wesentlicher Unterschied zwischen Linux und anderen Betriebssystemen ist der, dass hier nicht eine einzige Firma dahintersteckt wie Apple bei MacOS und Microsoft bei Windows. Linux ist ein komplett freier Betriebssystemkern, der von Linus Torvalds bereits 1990 entwickelt wurde, um das von Großrechnern aus dem akademischen Bereich bekannte UNIX-Betriebssystem auch auf kleinen Desktop-PCs verfügbar zu machen. Um diesen offiziellen Kern haben sich viele kommerzielle und freie Projekte gebildet, die alle das System ihren Bedürfnissen und Wünschen angepasst haben. Es gibt Desktop-Systeme, solche für den Serverbetrieb und Spezialeditionen für exotische Hardware oder ganz spezielle Anwendungsbereiche. Man spricht hier von sogenannten Linux-Distributionen. Es würde den Rahmen dieses Buches sprengen, auf all dies näher einzugehen. Es soll an dieser Stelle genügen zu sagen, dass dein Raspberry Pi mit einem echten modernen Linux-Kernel läuft, der auf einer offenen Distribution namens Debian basiert. Debian ist auch der Urvater der heute recht verbreiteten Desktop-Distributionen Ubuntu und Linux Mint.

2.4 Die SD- Karte vorbereiten

Nach der langen Vorrede wird es nun Zeit, endlich in die Praxis einzusteigen. Zunächst einmal musst du eine SD-Karte mit mindestens 32 GB Speicherkapazität besorgen. Sie darf natürlich gerne größer sein, abhängig davon, welchen Einsatzzweck du mit deinem RasPi verfolgst. Für den Einstieg in Raspbian und die Python-Programmierung bist du mit einer 64-GB-Karte auf der sicheren Seite und hast genug Platz, um die Beispiele aus diesem Buch und noch viel mehr zu speichern.

Doch wie kommen jetzt die Inhalte auf die leere SD-Karte?

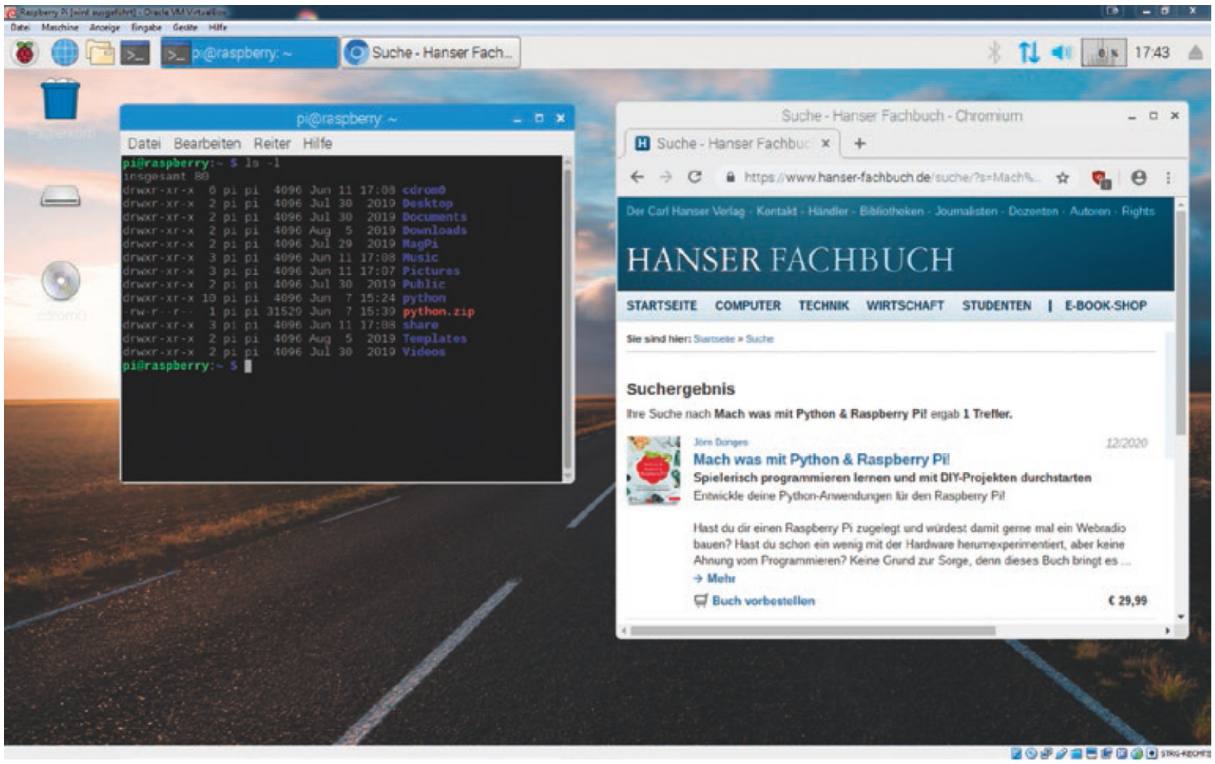

Bild 2.2 Der Raspbian-Standard-Desktop mit Terminalfenster und Webbrowser

2.4.1 NOOBS oder Raspbian?