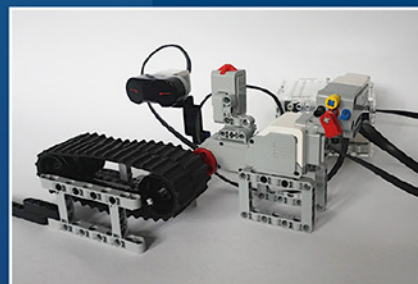




# LEGO<sup>®</sup>-Roboter

bauen, steuern und programmieren  
mit Raspberry Pi und Python



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

Liebe Leserinnen und Leser,

dieses E-Book, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Mit dem Kauf räumen wir Ihnen das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Jede Verwertung außerhalb dieser Grenzen ist ohne unsere Zustimmung unzulässig und strafbar. Das gilt besonders für Vervielfältigungen, Übersetzungen sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Je nachdem wo Sie Ihr E-Book gekauft haben, kann dieser Shop das E-Book vor Missbrauch durch ein digitales Rechtemanagement schützen. Häufig erfolgt dies in Form eines nicht sichtbaren digitalen Wasserzeichens, das dann individuell pro Nutzer signiert ist. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Beim Kauf des E-Books in unserem Verlagsshop ist Ihr E-Book DRM-frei.

Viele Grüße und viel Spaß beim Lesen,

*Ihr mitp-Verlagsteam*



Neuerscheinungen, Praxistipps, Gratiskapitel,  
Einblicke in den Verlagsalltag –  
gibt es alles bei uns auf Instagram und Facebook



[instagram.com/mitp\\_verlag](https://www.instagram.com/mitp_verlag)



[facebook.com/mitp.verlag](https://www.facebook.com/mitp.verlag)

*für Viorica*

Thomas Kaffka

**LEGO®-Roboter**  
**bauen, steuern und programmieren**  
mit Raspberry Pi und Python



## **Bibliografische Information der Deutschen Nationalbibliothek**

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

ISBN 978-3-7475-0311-9

1. Auflage 2021

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

E-Mail: [mitp-verlag@sigloch.de](mailto:mitp-verlag@sigloch.de)

Telefon: +49 7953 / 7189 - 079

Telefax: +49 7953 / 7189 - 082

© 2021 mitp Verlags GmbH & Co. KG, Frechen

LEGO®, das LEGO Logo, die Konfiguration der Noppen und die Minifigur sind Marken der LEGO Gruppe, durch die das vorliegende Buch weder gesponsert noch autorisiert oder unterstützt wird.

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

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt 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 benutzt werden dürften.

Lektorat: Janina Bahlmann, Rebecca Saalfeld

Sprachkorrektorat: Petra Heubach-Erdmann

Coverbild: Thomas Kaffka

Covergestaltung: Christian Kalkert

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

# Inhaltsverzeichnis

<b>Einleitung</b> . . . . .	11
Mit dem Buch arbeiten . . . . .	13
Der LEGO Digital Designer 4.3 . . . . .	15
<b>Teil I Die Hardware für die Roboter</b>	17
<hr/>	
<b>1 LEGO als Grundlage für unsere Roboter</b> . . . . .	19
1.1 Roboter als Bausatz . . . . .	19
1.2 LEGO mit Elektronikkomponenten versehen . . . . .	22
1.2.1 Folgende LEGO-Elektronikkomponenten werden verwendet . . . . .	22
1.2.2 Folgende Fremdkomponenten werden verwendet . . . . .	23
<b>2 Der Raspberry Pi stellt sich vor</b> . . . . .	25
2.1 Der Einplatinencomputer . . . . .	25
2.2 Die GPIO-Schnittstelle . . . . .	27
2.3 Den Raspberry Pi konfigurieren . . . . .	30
2.3.1 Den RPi mit weiterer Hardware versehen . . . . .	31
2.3.2 Die SD-Karte vorbereiten . . . . .	31
2.4 Den BrickPi3 anschließen . . . . .	32
<b>3 Die elektronischen Komponenten</b> . . . . .	35
3.1 Verwendung der LEGO-Elektronik-Komponenten . . . . .	35
3.1.1 Der LEGO-Colorsensor . . . . .	35
3.1.2 Der LEGO-Touchsensor . . . . .	36
3.1.3 Der LEGO-Infrarotsensor . . . . .	36
3.1.4 Der Hitechnic-Kompass-Sensor . . . . .	37
3.1.5 Der Hitechnic-Gyrosensor . . . . .	37
3.1.6 Der mittlere LEGO-Motor . . . . .	37
3.1.7 Der LEGO-Motor . . . . .	38
3.2 Verwendung von Fremdkomponenten . . . . .	40
3.2.1 Der Fototransistor . . . . .	41
3.2.2 Die LED . . . . .	41

3.2.3	Der Touchsensor . . . . .	41
3.2.4	Der Schallgeber . . . . .	42
3.2.5	Die Motoren und der Motortreiber . . . . .	42
3.2.6	Die Kamera . . . . .	50
3.2.7	Der BrickPi3 . . . . .	51

**Teil II Programmieren lernen** 53

---

<b>4</b>	<b>Die Programmiersprache Python</b> . . . . .	<b>55</b>
4.1	Die Entwicklungsumgebung . . . . .	55
4.2	Die ersten Schritte . . . . .	56
4.3	Hallo, ich bin ein Roboter . . . . .	58
4.4	Editieren und ausführen . . . . .	60
<b>5</b>	<b>Variablen</b> . . . . .	<b>63</b>
5.1	Datentypen . . . . .	63
5.1.1	Zahlen . . . . .	63
5.1.2	Strings (Zeichenketten) . . . . .	67
5.1.3	Wahrheitswerte . . . . .	69
5.2	Datenstrukturen . . . . .	70
5.2.1	Listen . . . . .	70
5.2.2	Tupel . . . . .	72
5.3	Konstanten . . . . .	72
<b>6</b>	<b>Verzweigungen</b> . . . . .	<b>73</b>
6.1	Bedingungen . . . . .	73
6.2	Das if-Statement . . . . .	74
6.3	Das else-Statement . . . . .	75
6.4	else-if-Kaskaden . . . . .	79
6.5	Modulbibliotheken . . . . .	80
6.6	Experiment: LED schalten . . . . .	80
<b>7</b>	<b>Schleifen</b> . . . . .	<b>85</b>
7.1	Das while-Statement . . . . .	85
7.2	Das for-Statement . . . . .	88
7.3	Das break-Statement . . . . .	91
7.4	Das continue-Statement . . . . .	92
7.5	Experiment: Blinklicht . . . . .	92
7.6	Experiment: LED dimmen . . . . .	94



<b>8</b>	<b>Funktionen</b> . . . . .	97
8.1	Deklaration . . . . .	97
8.2	Parameter . . . . .	98
8.3	Rückgabewert . . . . .	99
8.4	Experiment: Licht erkennen . . . . .	100
<b>9</b>	<b>Klassen und Objekte</b> . . . . .	107
9.1	Definition einer Klasse . . . . .	107
9.2	Methoden einer Klasse . . . . .	109
9.3	Vererbung . . . . .	111
9.4	Experiment: Töne erzeugen . . . . .	112
<b>Teil III Projekte</b>		119
<b>Projekte mit dem BrickPi3 und LEGO-Komponenten</b>		120
<b>10</b>	<b>Wänden und Gegenständen ausweichen</b> . . . . .	121
10.1	Das LEGO-Modell . . . . .	122
10.2	Die LED . . . . .	123
10.3	Der Infrarotsensor . . . . .	125
10.4	Die Motoren . . . . .	127
10.5	Wänden und Gegenständen ausweichen . . . . .	136
<b>11</b>	<b>Himmelsrichtungen erkennen</b> . . . . .	141
11.1	Das LEGO-Modell . . . . .	142
11.2	Der Kompass-Sensor . . . . .	142
11.3	Der Touchsensor . . . . .	147
11.4	Die übrige Hardware . . . . .	148
11.5	Himmelsrichtungen erkennen . . . . .	148
<b>12</b>	<b>Auf dem Tisch bleiben</b> . . . . .	153
12.1	Das LEGO-Modell . . . . .	154
12.2	Der Gyrosensor . . . . .	154
12.3	Die übrige Hardware . . . . .	157
12.4	Auf dem Tisch bleiben . . . . .	157
<b>13</b>	<b>Ein Labyrinth lösen mit einem Expertensystem</b> . . . . .	161
13.1	Das LEGO-Modell . . . . .	162
13.2	Der Colorsensor . . . . .	163
13.3	Die übrige Hardware . . . . .	165

13.4	Das Expertensystem. . . . .	165
13.4.1	Die Regelbasis. . . . .	166
13.4.2	Der Stapelspeicher (Stack) . . . . .	168
13.4.3	Der Regelinterpreter. . . . .	170
13.4.4	Die Move Engine . . . . .	172
13.4.5	Die Funktion move_since_wall . . . . .	173
13.4.6	Die Funktion move_back . . . . .	174
13.4.7	Die Funktion rotate. . . . .	175
13.5	Labyrinth. . . . .	176
<b>14</b>	<b>Linienverfolgung mit einem neuronalen Netz . . . . .</b>	<b>185</b>
14.1	Das LEGO-Modell . . . . .	187
14.2	Einführung in neuronale Netze . . . . .	187
14.3	Der Colorsensor als Lichtsensor. . . . .	191
14.4	Die übrige Hardware . . . . .	192
14.5	Das neuronale Netz . . . . .	192
14.6	Linienverfolgung klassisch . . . . .	197
14.7	Linienverfolgung trainieren . . . . .	202
14.8	Linienverfolgung mit neuronalem Netz. . . . .	210
<b>15</b>	<b>Objekte klassifizieren mit einem neuronalen Netz. . . . .</b>	<b>215</b>
15.1	Das LEGO-Modell . . . . .	218
15.2	Benötigte Hardware. . . . .	219
15.3	Kategorisieren lernen . . . . .	219
15.4	Trainingsdaten erzeugen. . . . .	221
15.5	Das Training. . . . .	226
15.6	Objekte klassifizieren . . . . .	231
<b>16</b>	<b>Pappkarten abschießen per Bilderkennung . . . . .</b>	<b>237</b>
16.1	Das LEGO-Modell . . . . .	238
16.2	Die RPi-Kamera . . . . .	240
16.3	Die übrige Hardware . . . . .	243
16.4	Bilderkennung . . . . .	243
16.4.1	Farben identifizieren und definieren . . . . .	244
16.4.2	Objekt im Bild erkennen . . . . .	252
16.4.3	Die eigentliche Bilderkennung . . . . .	256
16.5	Pappkarten abschießen . . . . .	262
<b>17</b>	<b>Joghurtbecher sammeln per Bilderkennung . . . . .</b>	<b>267</b>
17.1	Das LEGO-Modell . . . . .	269

17.2	Benötigte Hardware . . . . .	270
17.3	Bildererkennung . . . . .	271
17.4	Joghurtbecher sammeln . . . . .	271

---

	<b>Projekte mit elektronischen Fremdkomponenten</b>	287
<b>18</b>	<b>Texte morsen</b> . . . . .	289
18.1	Das LEGO-Modell . . . . .	291
18.2	Eine Hilfsplatine basteln . . . . .	292
	18.2.1 Die Spannungsversorgung . . . . .	293
	18.2.2 Die Spannungsteiler . . . . .	293
	18.2.3 Das I <sup>2</sup> C-Interface . . . . .	294
	18.2.4 Der Taster . . . . .	295
	18.2.5 Die LED . . . . .	295
	18.2.6 Der Piezo-Schallgeber . . . . .	297
	18.2.7 Der IC MCP3008 . . . . .	300
18.3	Touchsensoren basteln . . . . .	302
18.4	Morsecode übersetzen und eingeben . . . . .	304
	18.4.1 Text in Morsecode übersetzen . . . . .	304
	18.4.2 Morsecode in Text überführen . . . . .	307
<b>19</b>	<b>Abfahren der »platonischen Flächen«</b> . . . . .	313
19.1	Das LEGO-Modell . . . . .	314
19.2	Getriebemotor mit Encoder und Motortreiber . . . . .	316
19.3	Die »platonischen Flächen« . . . . .	321
<b>20</b>	<b>Suche des hellsten Orts im Raum</b> . . . . .	325
20.1	Das LEGO-Modell . . . . .	326
20.2	Der Fototransistor . . . . .	326
20.3	Die hellste Lichtquelle des Raumes finden . . . . .	329
<b>21</b>	<b>Ausblick</b> . . . . .	333
<b>A</b>	<b>Anhang</b> . . . . .	335
A.1	Download . . . . .	335
A.2	Bezugsquellen . . . . .	336
	<b>Stichwortverzeichnis</b> . . . . .	338



# Einleitung

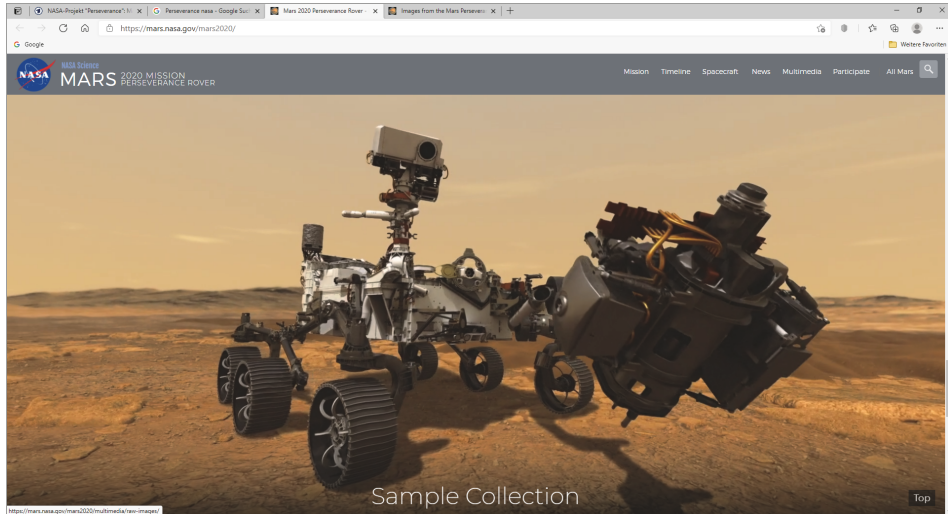
Die Robotertechnologie wird in unserem alltäglichen Leben immer präsenter. Es gibt Roboter, die unseren Boden kehren, die unseren Rasen mähen oder uns in einem gewissen Umfang bedienen können. In der Industrie werden schon seit längerem Roboter zur Fertigung eingesetzt. Es werden von ihnen dort aber auch Überwachungs- und Serviceleistungen erbracht. Es spannt sich schon jetzt ein weites Feld auf, in dem Roboter eingesetzt werden können.



**Abb. 1:** Roboter, die beispielsweise in der industriellen Fertigung eingesetzt werden können (mit freundlicher Genehmigung der Firma KUKA AG, Augsburg)

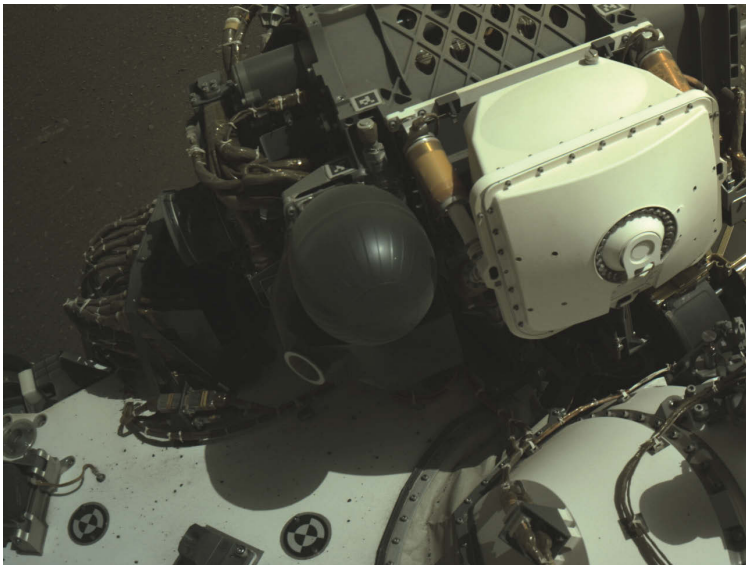
Und die Einsatzmöglichkeiten werden immer ausgefeilter. Es befinden sich Roboter in der Entwicklung, die kranke und alte Menschen versorgen können, oder Roboter, die uns eine echte Küchenhilfe sein können.

Außerdem wäre die Erforschung anderer Planeten, wie beispielsweise des Mars, ohne Roboter nicht denkbar. Aktuell befindet sich der NASA-Roboter »Perseverance« (Ausdauer) auf dem Mars, um diesen nach Lebensspuren zu erkunden.



**Abb. 2:** Der Roboter Perseverance (Quelle: NASA-Webseite)

Wegen der großen Entfernung zwischen Mars und Erde sind die Signale, die der Roboter und das Kontrollzentrum der NASA austauschen, etwa drei Minuten unterwegs (der Mars ist zurzeit etwa 56 Mio. km von der Erde entfernt und die Lichtgeschwindigkeit beträgt etwa 300.000 km/sec). Diese Zeit ist zu lang, als dass die NASA den Roboter in Echtzeit steuern könnte. Es ist nur möglich, dem Roboter generelle Befehle zu senden, die er dann autark ausführt.



**Abb. 3:** Perseverance fotografiert sich selbst. (Quelle: NASA: NASA/JPL-Caltech)

Der Roboter muss, um seine Aufgaben zu erfüllen, eigenständig Entscheidungen treffen können. Dazu ist eine gewisse Intelligenz notwendig.



**Abb. 4:** Perseverance fotografiert die Marslandschaft. (Quelle: NASA: NASA/JPL-Caltech/ASU)

Wie Roboter so agieren können, erfahren Sie in diesem Buch. Sie lernen Grundfertigkeiten von Robotern kennen, die für verschiedene Aufgaben angewendet werden können. Vielleicht verfügt ja ein NASA-Roboter auch über eine von mir vorgestellte Grundfertigkeit.

Sie können mit diesem Buch alles praktisch erkunden, indem Sie meine Roboter mit LEGO-Teilen nachbauen, um so einen persönlichen Eindruck von der Leistungsfähigkeit auch schon kleinerer Roboter zu erlangen. LEGO-Teile haben dabei den Vorteil, dass Sie einen aufgebauten Roboter später wieder zerlegen und dann einen neuen bauen können.

Wir beschäftigen uns zunächst mit der generellen Steuerung von Robotern, später aber auch mit Themen der »Künstliche Intelligenz Forschung«. Sie werden neuronale Netze und ein Expertensystem einsetzen, um Roboter zu steuern. Solche Roboter besitzen schon eine gewisse Grundintelligenz und treffen intelligente Entscheidungen auf der Basis eigener Steuerungen.

## Mit dem Buch arbeiten

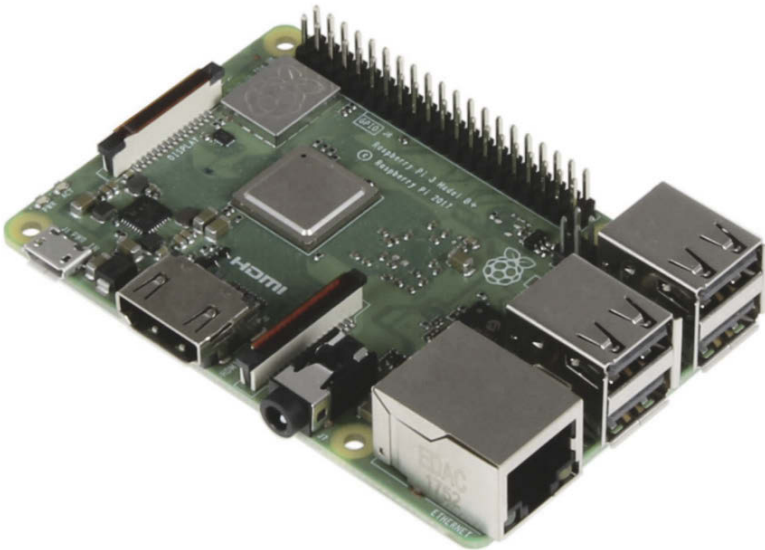
Dieses Buch ist in drei Teile geteilt. Als Erstes stelle ich die Hardware für die Roboter vor. Danach gebe ich Ihnen eine Einführung in Python, die von mir zur Robotersteuerung verwendete Programmiersprache. Im dritten Teil stelle ich Ihnen elf Projekte vor, bei denen Sie selbst Roboter bauen und programmieren. Sie lernen dabei sukzessive die Roboter-Programmierung mit der Programmiersprache

Python, aber auch Konzepte kennen, die zur Steuerung von Robotern herangezogen werden.

Ich verwende zum Bau des Chassis der Roboter Bauteile aus dem LEGO-EV3-Baukasten. Sie können natürlich auch die Bauteile aus eigenen Beständen verwenden oder sich gewisse Teile, die Ihnen fehlen, von LEGO bestellen (Einzelteillisten befinden sich jeweils im Download zum Buch auf der Webseite [www.miptp.de/0310](http://www.miptp.de/0310)). Ein Teil der Roboter wird mit den zum EV3-Baukasten gehörenden Motoren und Sensoren ausgestattet. Ich verwende aber auch LEGO-Sensoren, die dazugekauft werden können. Allgemein können Sie sich sämtliche LEGO-Komponenten separat beschaffen (richten Sie sich dazu an den oben genannten Teilelisten aus). Ein EV3-Baukasten ist natürlich nicht unbedingt erforderlich, Sie hätten aber so die Bauteile, bis auf die Hitechnic-Sensoren, komplett vorliegen.

Drei Projekte (Kapitel 18 bis 20) erstelle ich mit Fremdkomponenten im Sinne von allgemeinen Technikkomponenten, die nicht von LEGO sind. Es handelt sich dabei um Getriebemotoren und Sensoren wie einen Fototransistor oder Kommunikationsmittel wie eine LED. Wenn Sie Interesse haben, diese Projekte mit Fremdkomponenten zu erarbeiten, werden Sie in diesem Rahmen mit dem Lötcolben basteln.

Als Computer, der die Roboter steuert, verwende ich den *Raspberry Pi*-Computer. Dabei handelt es sich um einen etwa scheckkartengroßen Einplatinencomputer, der durch das Betriebssystem Linux gesteuert wird.



**Abb. 5:** Der Raspberry Pi




Er verfügt, wie Sie noch sehen werden, über eine Hardwareschnittstelle in Form von 40 kleinen Pins, an die die Roboterhardware zur Steuerung angeschlossen werden kann.

Zusätzlich verwende ich den *BrickPi3*. Das ist eine Hardwareplatine, die auf den Raspberry Pi gesteckt wird, um die LEGO-Hardware zu steuern. Der Verbund von Raspberry Pi und BrickPi3 erledigt dabei die Aufgaben, die der EV3-Baustein im LEGO-Baukasten erledigen würde. Der Vorteil bei der Verwendung des Raspberry Pi besteht darin, dass Sie zum einen eine vollwertige, momentan sehr aktuelle Programmiersprache (Python) einsetzen können, um Ihre Roboter zu steuern, der Einsatz neuronaler Netze wäre anders gar nicht denkbar. Zum anderen lernen Sie einen sehr interessanten und preiswerten Computer kennen, den Sie auch für andere Aufgaben verwenden können. Er verfügt über ein Betriebssystem, das *Raspberry Pi OS*, das eine grafische Benutzeroberfläche hat, sodass sich auch Apple- und Windows-Benutzer sehr schnell heimisch fühlen werden.

## Der LEGO Digital Designer 4.3

Ich habe die Bauanleitungen zu den einzelnen Robotern mit dem Tool »LEGO Digital Designer 4.3« (<https://www.lego.com/de-de/lld>) erstellt und diese in entsprechenden Dateien gespeichert, die dem Download zum Buch unter [www.mitp.de/0310](http://www.mitp.de/0310) beiliegen (die Dateien haben die Endung `.lxf`). Auf diese Weise können Sie alle Roboter-Modelle im Buch ganz einfach nachbauen.

Wenn Sie, nachdem Sie den LEGO Digital Designer installiert haben, auf die Datei doppelklicken, wird das Programm gestartet und das Modell angezeigt. Da LEGO dieses Tool per Internetserver nicht mehr unterstützt, wird eine Fehlermeldung angezeigt, die Sie mit einem Klick auf den OK-Button quittieren.

In der oberen rechten Ecke der Buttonleiste befindet sich der Button , mit dem Sie die Bauanleitung generieren können. Danach können Sie mit den Pfeilbuttons zwischen den einzelnen Steinen hin- und herschalten und das Modell erstellen. Da es für den BrickPi3 natürlich kein Symbol im Programm gibt, habe ich den Platz, den er einnimmt, im Modell immer frei gelassen.

Nun sind Sie sicherlich schon ganz gespannt auf die Themen, die ich Ihnen in meinem Buch zeigen werde, daher sollten wir sofort in die Behandlung der Technologie einsteigen. Ich wünsche Ihnen viel Vergnügen bei der Lektüre meines Buches und beim Bau und der Programmierung der vorgestellten Roboter.

Thomas Kaffka, im Juli 2021



# Teil I

## Die Hardware für die Roboter

### In diesem Teil:

- **Kapitel 1**  
LEGO als Grundlage für unsere Roboter . . . . . 19
- **Kapitel 2**  
Der Raspberry Pi stellt sich vor . . . . . 25
- **Kapitel 3**  
Die elektronischen Komponenten . . . . . 35



# LEGO als Grundlage für unsere Roboter

Es ist eine gute Idee, LEGO-Komponenten zum Roboterbau zu verwenden. Ein aufgebauter Roboter kann immer wieder zerlegt werden und ein neuer erfunden und gebaut werden. Der eigenen Fantasie sind dabei grundsätzlich keine Grenzen gesetzt. Daher verwende ich in meinem Buch LEGO-Komponenten, um ein Roboterchassis zu erstellen. Ich lege dabei die Bauteile des LEGO-EV3-Baukastens zugrunde.

Leser, die diesen Baukasten nicht besitzen, aber über LEGO-Bausteine verfügen oder gewillt sind, sich diese anzuschaffen, können auch ganz entspannt sein. Denn ich gebe bei jedem Roboter eine Teileliste über die LEGO- oder Fremdkomponenten an, die ich für den jeweiligen Roboter verwende. Diese Teilelisten sind im Download zum Buch ([www.mipt.de/0310](http://www.mipt.de/0310)) sowie in den einzelnen Kapiteln zu finden.

## 1.1 Roboter als Bausatz

Sie könnten sich natürlich, wenn Sie sich mit der Robotik beschäftigen möchten, auch einen Bausatz zu einem fertigen Roboter anschaffen, statt eigene Roboter aus LEGO zu bauen. Diese Roboter haben aber den Nachteil, dass es sich um fertige Maschinen handelt, die in ihren Freiheitsgraden eben deshalb beschränkt sind. Wenn man ihre Motoren und sonstigen Aktoren programmiert hat, ist es nicht mehr interessant, sich mit ihnen weiter zu beschäftigen.

Dazu hier einige Beispiele von Robotern, die ich mir unter anderem angeschafft habe (Abbildung 1.1 bis Abbildung 1.3).

Der YETI (Abbildung 1.1) ist ein einfacher Roboter, der über zwei Beine verfügt. Er ist mechanisch so aufgebaut, dass er mit zwei Servomotoren gesteuert werden kann. Um zu gehen, verlagert er sein Gewicht auf eines der beiden Beine und bewegt das andere vor. Danach wird das gegenüberliegende Bein belastet. Ich habe für den Roboter zusätzlich ein Vier-Segment-Display angeschafft und eingebaut. Damit kann er Meldungen oder seinen Status mitteilen.



Abb. 1.1: Der YETI von AREXX Engineering

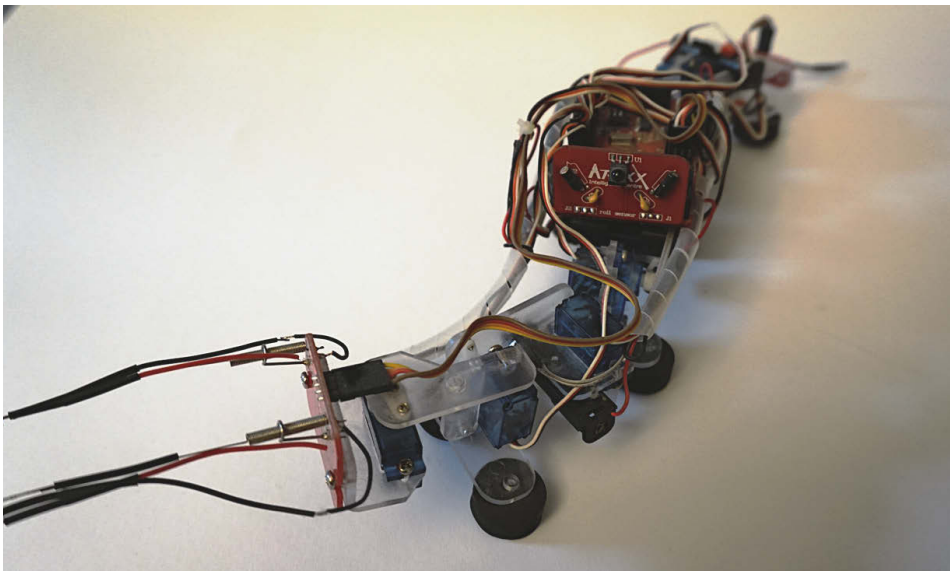
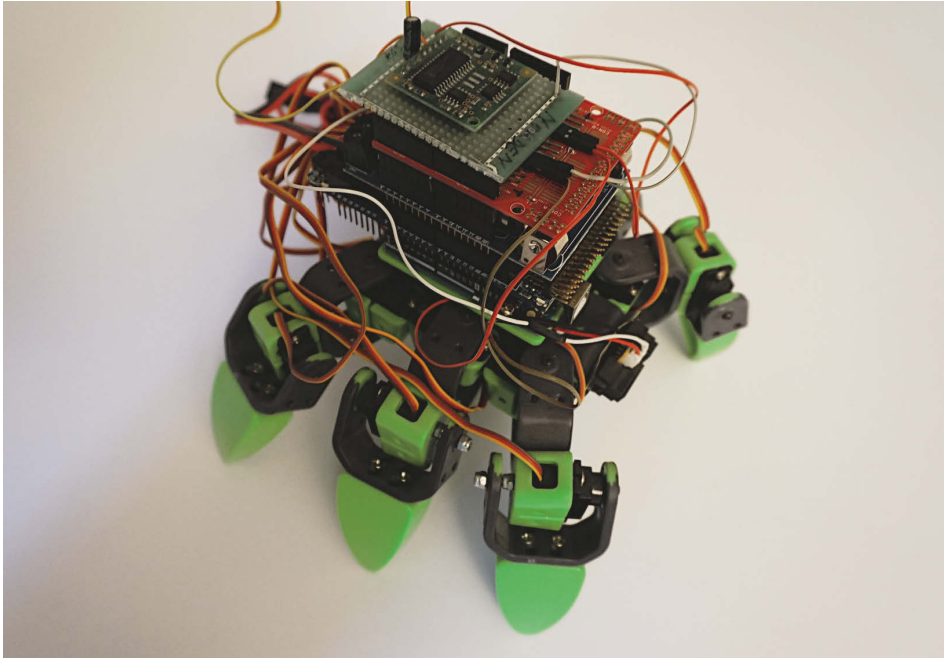


Abb. 1.2: Die Caterpillar von AREXX Engineering

Die Caterpillar ist ein wurmförmiger Roboter. Sein Körper ist in verschiedene Segmente geteilt, die sich durch Servomotoren jeweils auf und ab oder links und rechts bewegen lassen. Durch eine koordinierte Ansteuerung der Motoren ist der

Roboter in der Lage, sich fortzubewegen. Seine Sensoren sind zwei Antennen vorne und eine Antenne hinten sowie ein Rollsensor.



**Abb. 1.3:** Der ALLBOT von Velleman

Der ALLBOT ist ein sechsbeiniger Roboter, der entsprechend anspruchsvoll programmtechnisch gesteuert wird. Die sechs Beine müssen sich ja koordiniert bewegen, damit er sich vorwärtsbewegt. Dem Roboter habe ich einen Infrarotsensor (vorne am Chassis) spendiert, damit er Abstände zu einem Hindernis messen kann, sowie einen Kompass-Sensor (oberste Platine), der ihn befähigt, die Richtung zu bestimmen, in der er sich bewegt.

Natürlich kann man mit den vorgestellten Robotern vortrefflich experimentieren und diese auch mit zusätzlichen elektronischen Komponenten erweitern. Aber irgendwann hat man sie ausgereizt und dann stehen sie nur noch herum und sehen schön aus. Das ist schade.

Daher gehe ich in diesem Buch einen anderen Weg und verwende LEGO-Komponenten sowie elektronische Fremdkomponenten, um Roboter zu bauen. LEGO-Roboter haben den Vorteil, dass man sie, nachdem man sich ausführlich mit ihnen beschäftigt hat, zerlegen und dann einen neuen Roboter bauen kann. Wenn man LEGO für seine Roboter verwendet, wird dieses interessante Hobby niemals langweilig.

## 1.2 LEGO mit Elektronikkomponenten versehen

Das LEGO-Mindstorms-System bietet Computer, Motoren und Sensoren für den Bau von Robotern aus LEGO-Technikteilen an. Diese originalen LEGO-Bauteile sind im Vergleich zu Fremdkomponenten sehr teuer. Daher können sowohl die Originalteile als auch – als Alternative – elektronische Fremdkomponenten verwendet werden, wie Sie in den letzten drei Kapiteln zu den Roboterprojekten sehen können.

### 1.2.1 Folgende LEGO-Elektronikkomponenten werden verwendet

Für die Modelle im Buch verwende ich die folgenden LEGO-Komponenten.



Abb. 1.4: Der LEGO-Colorsensor



Abb. 1.8: Der mittlere Motor von LEGO EV3



Abb. 1.5: Der LEGO-Infrarotsensor



Abb. 1.9: Der LEGO-EV3-Motor



Abb. 1.6: Der LEGO-Touchsensor



Abb. 1.10: Der Hitechnic-Gyrosensor



Abb. 1.7: Der Hitechnic-Kompass-Sensor



## 1.2.2 Folgende Fremdkomponenten werden verwendet

Ein echter Elektronik-Bastler hat keine Angst vor Komponenten von Drittanbietern. Diese Bauteile haben den enormen Vorteil, dass sie teilweise nur einen Bruchteil der originalen LEGO-Komponenten kosten. Man muss allerdings bereit sein, auch mal etwas zu löten, da beispielsweise der Lichtsensor zusammengebaut werden muss. Aber keine Bange, das kriegen Sie hin.

Ich verwende beim elektronischen Basteln einen 16-Watt-LötKolben. Seine Leistung darf nicht zu hoch sein, damit die elektronischen Bauteile keinen Schaden nehmen. Dazu benötigt man noch eine Spule Lötzinn und wer es ganz komfortabel haben möchte, der legt sich noch eine Abisolierzange zu, fertig.

Als Fremdkomponenten setze ich die folgenden Bauteile ein.



Abb. 1.11: Der Lichtsensor (Fototransistor)



Abb. 1.14: Getriebemotor plus Reifen und Encoder



Abb. 1.12: Der Touchsensor (Mikroschalter)



Abb. 1.15: Die LED



Abb. 1.13: Die Raspberry-Pi-Kamera



Abb. 1.16: Der Piezo-Schallgeber



# Der Raspberry Pi stellt sich vor

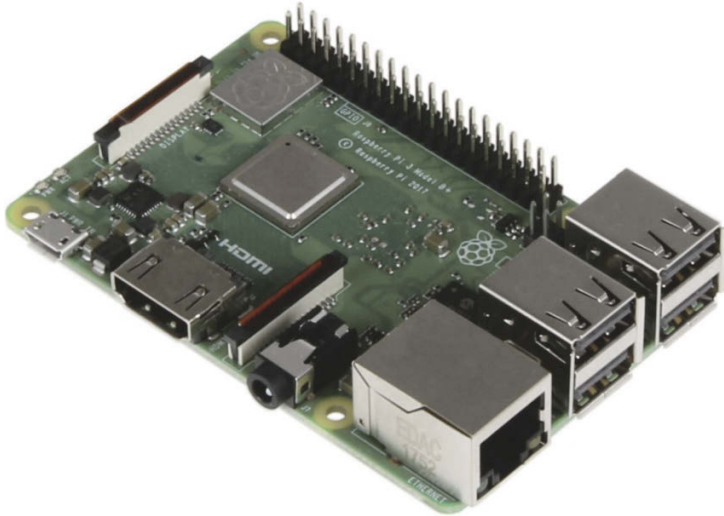


Abb. 2.1: Der Raspberry Pi 3 B+

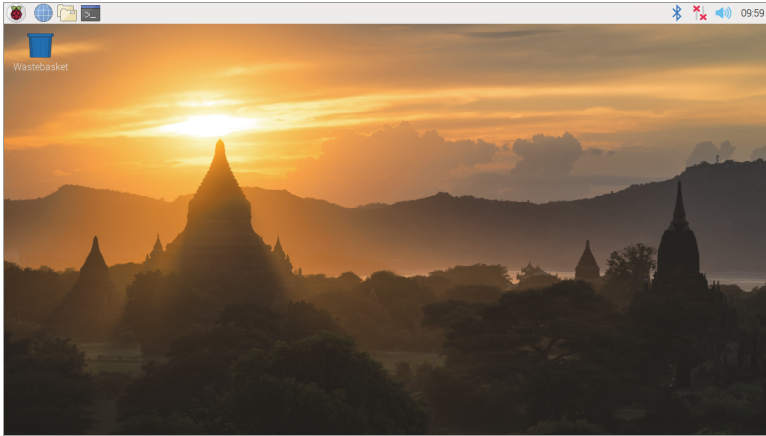
## 2.1 Der Einplatinencomputer

Der *Raspberry Pi 3 B+* hat das Format einer Scheckkarte, ist also sehr klein. Auch wenn es nicht so aussieht, er ist ein vollständiger Computer mit Betriebssystem. Es ist zumeist das *Raspberry Pi OS*, bei dem es sich um eine Linux-Distribution handelt. An den Raspberry Pi können eine USB-Tastatur, eine USB-Maus, ein HDMI-Monitor, ein LAN-Kabel sowie ein USB-Netzteil angeschlossen werden. Er verfügt aber auch über WLAN.

Das Betriebssystem wird auf einer Mikro-SD-Karte installiert. Das hat einen entscheidenden Vorteil. Wir verwenden den Raspberry Pi in diesem Buch sowohl zur Ansteuerung von LEGO-Komponenten als auch von sogenannten »Fremdkomponenten« (das sind elektronische Komponenten, die auf dem Elektromarkt verfügbar sind). Dazu werden zwei verschiedene Betriebssystemumgebungen benötigt. Insbesondere verwenden wir bei der Steuerung der LEGO-Komponenten eine spezielle Distribution von *Raspbian* (auch ein Betriebssystem für den Raspberry Pi), die von Dexter Industries, dem Lieferanten der Schnittstelle (BrickPi3) zu den LEGO-Komponenten, zur Verfügung gestellt wird. Beide Umgebungen halten wir

jeweils auf einer eigenen Mikro-SD-Karte vor. Wenn wir zwischen der Ansteuerung von LEGO- und Fremdkomponenten wechseln, brauchen wir so nur die Mikro-SD-Karte auszutauschen, wirklich praktisch.

Das Betriebssystem *Raspberry Pi OS* sowie das *Raspbian* (der Vorgänger des erstgenannten Betriebssystems) bieten eine grafische Oberfläche (GUI, graphical user interface), ähnlich wie der von Windows, sodass man mit dem Computer komfortabel arbeiten kann.



**Abb. 2.2:** Die GUI des Raspberry Pi OS

Das Betriebssystem bringt bereits eine Installation von Python mit, sodass Sie mit dem Programmieren Ihrer Roboter sofort loslegen können. Dabei verwenden wir Python 3.

Python ist eine moderne Programmiersprache, die von Guido van Rossum 1989 erfunden wurde. Es gibt eine ganze Menge Erleichterungen, die das Programmieren sehr komfortabel gestalten. Python verfügt unter anderem auch über die Möglichkeit, Programmabschnitte parallel zum Hauptprogramm verarbeiten zu lassen. Solche Programmabschnitte werden »Threads« genannt. Diese Technologie machen wir uns etwa bei der Verwaltung der Zähler zunutze, die von den Encodern der Motoren gesteuert werden. Eine kurze Einführung in Python erfolgt in **Teil II** meines Buches.

Den Raspberry Pi gibt es mittlerweile in verschiedenen Ausführungen, wie beispielsweise:

- Raspberry Pi 4, Modell B 4
- Raspberry Pi 4, Modell B 2
- Raspberry Pi 3, Modell B+
- Raspberry Pi 3
- Raspberry Pi 2, Modell B
- Raspberry Pi 1, Modell B+
- Raspberry Pi, Modell A+ etc.

Wir verwenden in diesem Buch den **Raspberry Pi 3 B+**, bei dem ich aber zumeist die »Drei« und das »B+« unterschlage. Leider können wir mit der neuesten Version des Raspberry Pi 4 nicht arbeiten, da er zur Zeit der Drucklegung des Buches nicht mit dem BrickPi3 (dem Modul zur Ansteuerung der LEGO-Komponenten) kompatibel ist.

Besonders schön ist es, dass es für den Raspberry Pi auch eine spezielle, preiswerte Kamera gibt, die direkt auf dem Board angeschlossen werden kann. Wir werden bei unseren Robotern auch diese Kamera zur Bilderkennung verwenden.

Für die Stromversorgung des Raspberry Pi bieten sich verschiedene Alternativen an. Wenn der Roboter programmiert wird, verwenden wir ein USB-Netzteil, das mindestens 2,5 Ampere liefern sollte. Damit ist es möglich, bei der Programmierung auch weitere elektronische Komponenten mit Strom zu versorgen. Die Motorspannung von +6 Volt (+9 Volt beim BrickPi3) entnehmen wir einem Batteriepack mit AA-Batterien. Wenn ein Roboter autonom laufen soll, verwenden wir zur Versorgung des Raspberry Pi einen USB-Akku.

## 2.2 Die GPIO-Schnittstelle

Die GPIO-Schnittstelle (general purpose input/output) ist eine 40-polige Stiftleiste am Rande der Raspberry-Pi-Platine, mit der Sie den Raspberry Pi mit verschiedenen Hardware-Komponenten verbinden können (siehe Abbildung 2.3).

GPIO - Schnittstelle

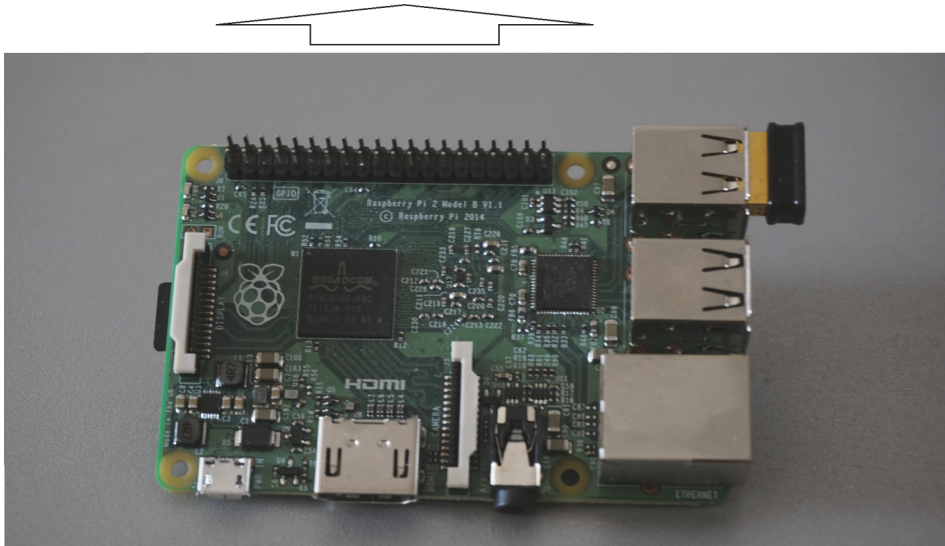


Abb. 2.3: Die GPIO-Schnittstelle des Raspberry Pi

Diese hat die Pinbelegung, die aus Abbildung 2.4 hervorgeht.

## Raspberry Pi

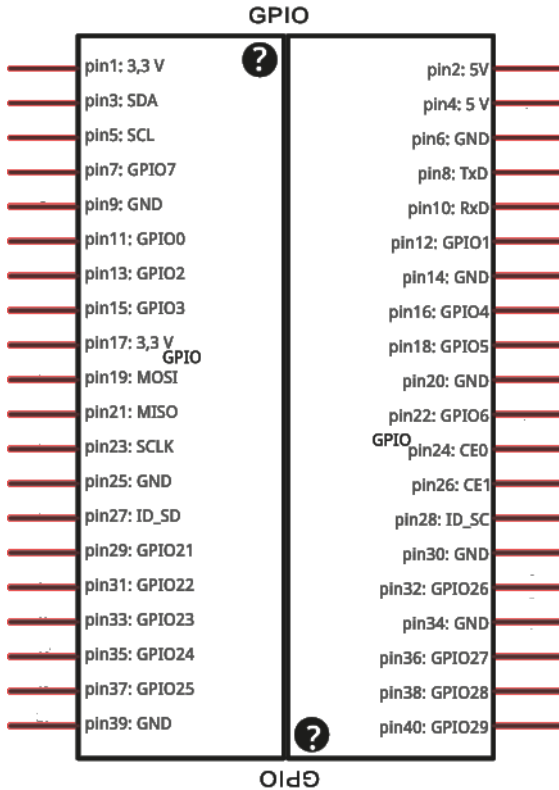


Abb. 2.4: Pinbelegung der GPIO-Schnittstelle

Die Pins 3 und 5 beziehen sich auch auf die I<sup>2</sup>C-Schnittstelle. Dabei handelt es sich um eine Schnittstelle, die zur Kommunikation zwischen dem Raspberry Pi und ICs verwendet wird.

Die Pins 11, 12, 13 können zum Anschluss von SPI-Bausteinen verwendet werden (Kanal 1). Auch die SPI-Schnittstelle dient zur Kommunikation mit elektronischen Komponenten.

### Hinweis

ICs sind elektronische Bauteile, die sich in kleinen schwarzen Plastikboxen befinden und über verschiedene Kontaktbeinchen verfügen. Das IC MCP3008