



---

# Robotik und Künstliche Intelligenz



Dr. Günter Spanner



elektor

LEARN > DESIGN > SHARE

● © 2019: Elektor Verlag GmbH, Aachen.

1. Auflage 2019

● Alle Rechte vorbehalten.

Die in diesem Buch veröffentlichten Beiträge, insbesondere alle Aufsätze und Artikel sowie alle Entwürfe, Pläne, Zeichnungen und Illustrationen sind urheberrechtlich geschützt. Ihre auch auszugsweise Vervielfältigung und Verbreitung ist grundsätzlich nur mit vorheriger schriftlicher Zustimmung des Herausgebers gestattet.

Die Informationen im vorliegenden Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Die in diesem Buch erwähnten Soft- und Hardwarebezeichnungen können auch dann eingetragene Warenzeichen sein, wenn darauf nicht besonders hingewiesen wird. Sie gehören dem jeweiligen Warenzeicheninhaber und unterliegen gesetzlichen Bestimmungen.

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autor können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Für die Mitteilung eventueller Fehler sind Verlag und Autor dankbar.

● Erklärung

Der Autor und der Herausgeber dieses Buches haben alle Anstrengungen unternommen, um die Richtigkeit der in diesem Buch enthaltenen Informationen sicherzustellen. Sie übernehmen keine Haftung für Verluste oder Schäden, die durch Fehler oder Auslassungen in diesem Buch verursacht werden, unabhängig davon, ob diese Fehler oder Auslassungen auf Fahrlässigkeit, Unfall oder andere Ursachen zurückzuführen sind.

Umschlaggestaltung: Elektor, Aachen

Satz und Aufmachung: D-Vision, Julian van den Berg | Oss (NL)

Druck: WILCO, Amersfoort, Niederlande

Printed in the Netherlands

● **ISBN 978-3-89576-345-8**  
**EISBN 978-3-89576-374-8**  
**EPUB 978-3-89576-375-5**

Elektor-Verlag GmbH, Aachen

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



Elektor ist Teil der Unternehmensgruppe Elektor International Media (EIM), der weltweit wichtigsten Quelle für technische Informationen und Elektronik-Produkte für Ingenieure und Elektronik-Entwickler und für Firmen, die diese Fachleute beschäftigen. Das internationale Team von Elektor entwickelt Tag für Tag hochwertige Inhalte für Entwickler und DIY-Elektroniker, die über verschiedene Medien (Magazine, Videos, digitale Medien sowie Social Media) in zahlreichen Sprachen verbreitet werden. [www.elektor.de](http://www.elektor.de)

**LEARN > DESIGN > SHARE**

---

<b>Warnhinweise</b> . . . . .	<b>10</b>
<b>Programmdownload</b> . . . . .	<b>11</b>
<b>Kapitel 1 • Einführung</b> . . . . .	<b>13</b>
1.1 Voraussetzungen . . . . .	14
1.2 Die Programmier- und Entwicklungsumgebungen . . . . .	15
<b>Kapitel 2 • Grundelemente der Robotik</b> . . . . .	<b>17</b>
<b>Kapitel 3 • Controller und Prozessoren: Die "Gehirne" der Roboter</b> . . . . .	<b>18</b>
3.1 Arduino als programmierbare Steuerzentrale . . . . .	18
3.2 Arduino-IDE . . . . .	20
3.3 Praxisanwendung: Roboter-Alarm . . . . .	22
3.4 C und "Processing" – Ein Vergleich. . . . .	23
3.5 Raspberry Pi . . . . .	24
3.6 Installation des Betriebssystems mit NOOBS. . . . .	27
3.7 Raspi-Config . . . . .	28
3.8 Die Kommandozeile . . . . .	28
3.9 Das Linux-Dateisystem. . . . .	29
3.10 Die wichtigsten Linux-Befehle . . . . .	30
3.11 Der Desktop des Pi. . . . .	30
<b>Kapitel 4 • Mechanische Bauelemente</b> . . . . .	<b>32</b>
4.1 Schrauben, Muttern und Bolzen. . . . .	32
4.2 Metalle, Holz und Kunststoffe . . . . .	34
4.3 Mechanik-Baukästen und Bausätze . . . . .	36
4.4 Räder. . . . .	37
<b>Kapitel 5 • Elektronische Komponenten und Module</b> . . . . .	<b>38</b>
5.1 Für Experimente und Prototypen: Breadboards . . . . .	38
5.2 Lochrasterplatten für einfache Aufbauten . . . . .	39
5.3 Widerstände . . . . .	40
5.4 Kondensatoren . . . . .	41
5.5 Potentiometer . . . . .	41
5.6 Leuchtdioden (LEDs) . . . . .	42
5.7 Dioden und Transistoren. . . . .	43
5.8 Motortreiber, Freilaufdioden und Abblockkondensatoren . . . . .	43

5.9 Für kleinere Motoren: der L293D . . . . .	44
5.10 Das Kraftpaket: L298 . . . . .	45
5.11 Motortreiber ULN2003 . . . . .	49
5.12 Motortreiber-Modul TBB6612. . . . .	50
5.13 PWM und Servo-Modul . . . . .	51
5.14 Das Robot-HATs-Modul für den Raspberry Pi . . . . .	52
<b>Kapitel 6 • Elektrische Einheiten und Aktoren . . . . .</b>	<b>55</b>
6.1 Gleichstrom- und Getriebemotoren . . . . .	55
6.2 PWM-Steuerung . . . . .	56
6.3 Stepper . . . . .	57
6.4 Einzelschrittmodus. . . . .	60
6.5 Präzise gesteuerte Motorkraft: Der Servo . . . . .	61
6.6 Servomotoren im Griff: Die Servo-Bibliothek. . . . .	64
6.7 Servos am Raspberry Pi . . . . .	66
<b>Kapitel 7 • Die Sinne der Roboter: Sensoren . . . . .</b>	<b>70</b>
7.1 Erfassung von Messwerten, Auflösung und Präzision . . . . .	71
7.2 Der ADC im Einsatz: Erfassung einer Potentiometerspannung . . . . .	72
7.3 Interne und externe Referenzspannungen . . . . .	74
7.4 Spannungsteiler für höhere Eingangsspannungen . . . . .	75
7.5 Präzise Messwertaufnahme mit Hilfe von Sensoren . . . . .	78
7.6 Temperaturmessung . . . . .	79
7.7 Präzise Temperaturwerte mit vor-kalibrierten Sensoren . . . . .	80
7.8 Digitale Sensoren . . . . .	82
7.9 Optische Sensoren. . . . .	83
7.10 Elektronisches Luxmeter. . . . .	85
7.11 Graphische Messwertaufnahme . . . . .	86
7.12 Entfernungsmessung und Kollisionsvermeidung. . . . .	88
7.13 Ultraschallmessung . . . . .	89
7.14 Optische Abstandsmessung. . . . .	94
7.15 Der Geruchssinn für Roboter: Gassensoren. . . . .	98
7.16 Lagebestimmung . . . . .	100

---

<b>Kapitel 8 • Kameras und Displays</b> .....	<b>105</b>
8.1 Die RasPi-Cam .....	106
8.2 WebCams mit USB-Anschluss .....	108
8.3 Einfach und universell: Standarddisplays .....	110
8.4 LCD-Display via I <sup>2</sup> C-Bus .....	115
8.5 Unübersehbar: blinkendes Display-Modul als Warnsignal .....	117
8.6 OLED-Module: Displays im Kleinformat .....	118
8.7 Schnelle Grafikanwendungen .....	120
8.8 OLED-Display am Raspberry Pi .....	124
<b>Kapitel 9 • Energieriegel für Roboter: Die Stromversorgung</b> .....	<b>129</b>
9.1 Safety first: Sicherungen .....	129
9.2 NiMH-Akkkus .....	133
9.3 Li-Ionen-Akkus .....	134
9.4 Umgang mit Li-Ionen-Technologie .....	137
9.5 Noch genug Leistungsreserven? – Akkuspannungs-Überwachung .....	139
<b>Kapitel 10 • Fahrgestelle, Rahmen und Chassis</b> .....	<b>142</b>
10.1 Eigenbauten und Kits .....	142
10.2 JOY-iT Robot Car Kit .....	144
10.3 PiCar-V .....	144
10.4 Raupenantriebe .....	145
<b>Kapitel 11 • Roboter an der unsichtbaren Leine:     Drahtlose Fernsteuerung</b> .....	<b>147</b>
11.1 Funkfernsteuerung .....	147
11.2 Steuerung eines Roboterfahrzeugs .....	149
11.3 Steuern mit dem Smartphone: Bluetooth .....	154
11.4 Bluetooth-Kommunikation .....	156
11.5 WLAN .....	159
11.6 Heimnetzwerke .....	160
11.7 Raspberry Pi im WLAN .....	162
<b>Kapitel 12 • Roboterfahrzeuge und autonomes Fahren</b> .....	<b>165</b>
12.1 Für den Einstieg: Auf zwei Rädern durch die Welt .....	166
12.2 Autonomes Fahren durch Linienverfolgung .....	170

12.3	Musterbeispiel der Bionik: Lichtsuchende Roboter . . . . .	177
12.4	Lichtverfolger mit zwei Sensoren . . . . .	182
12.5	Fledermaustechnik: Kollisionsvermeidung mit Ultraschall . . . . .	184
12.6	Bluetooth-gesteuerter Zwei-Rad-Robot. . . . .	191
12.7	Vier Räder und Servolenkung . . . . .	198
12.8	Pi-Car mit WLAN-Anbindung . . . . .	201
12.9	Geländegängiger Raupen-Robot. . . . .	207
12.10	StepperBot und Odometrie . . . . .	213
12.11	Umweltdaten sammeln: EnviRoBot. . . . .	217
<b>Kapitel 13 • Meisterwerke der Regelungstechnik: Selbstbalancierende Robots . . . . .</b>		<b>223</b>
13.1	Zwei- und Dreipunktregelung . . . . .	224
13.2	Grundlagen der digitalen Regelungstechnik. . . . .	225
13.3	Balance auf einer Achse: BalBot . . . . .	227
13.4	Selbstbalance-Algorithmen . . . . .	229
<b>Kapitel 14 • Unentbehrlich in der Industrie: Roboterarme und Manipulatoren . . . . .</b>		<b>236</b>
14.1	Roboterarm mit Servosteuerung . . . . .	237
14.2	Armsteuerung mit Gleichstrommotoren . . . . .	241
14.3	Roboterfahrzeug mit Greifarm . . . . .	247
<b>Kapitel 15 • Zwei, vier oder sechs Beine: Laufroboter in allen Varianten . . . . .</b>		<b>248</b>
15.1	Von Krabben und Spinnen: Quadrupeds und Hexapods . . . . .	249
15.2	Vierbeiner in der Praxis: Quadruped. . . . .	251
15.3	Laufen auf zwei Beinen. . . . .	254
15.4	Bipeds in der Praxis . . . . .	255
15.5	GoGoBot als zweibeinige Experimentierplattform. . . . .	256
<b>Kapitel 16 • Künstliche Intelligenz . . . . .</b>		<b>262</b>
16.1	Ein kurzer Blick in die Geschichte. . . . .	262
16.2	Anwendungen und Errungenschaften der KI . . . . .	264
16.3	KI-Praxis mit Python . . . . .	266
16.4	Objekterkennung. . . . .	267

---

16.5 Gesichtserkennung . . . . .	272
<b>Kapitel 17 • Intelligente Roboter . . . . .</b>	<b>275</b>
17.1 Der elektronische Jagdhund: Verfolgung eines rollenden Balls. . . . .	275
17.2 Autonomes Fahren mit Kamera-Unterstützung . . . . .	282
<b>Kapitel 18 • Humanoide Roboter . . . . .</b>	<b>292</b>
18.1 Roboter, Androiden und Cyborgs erobern die Welt . . . . .	294
18.2 Schöne neue Welt? – Die Zukunft der Robotik . . . . .	296
18.3 Wer bin ich? – Maschinenbewusstsein . . . . .	299
<b>Kapitel 19 • Ausblick: Neuromorphe Chipstechnologie . . . . .</b>	<b>302</b>
<b>Anhang: Bussysteme und Mikropowertechniken . . . . .</b>	<b>307</b>
Schnell und Einfach: der I <sup>2</sup> C-Bus . . . . .	307
Mikropowertechniken und Sleep-Modi . . . . .	309
<b>Bezugsquellen . . . . .</b>	<b>311</b>
<b>Literatur . . . . .</b>	<b>312</b>
<b>Abbildungsverzeichnis . . . . .</b>	<b>313</b>
<b>Index . . . . .</b>	<b>317</b>

## Warnhinweise

1. Für den Betrieb der in diesem Buch vorgestellten Schaltungen dürfen nur geprüfte doppelt-isolierte Sicherheitsnetzgeräte verwendet werden, da bei einem Isolationsfehler eines einfachen Netzteils lebensgefährliche Spannungen an nicht-isolierten Bauelementen anliegen können.
2. Autor und Verlag übernehmen keinerlei Haftung für Schäden, die durch den Aufbau der beschriebenen Projekte entstehen.
3. Elektronische Schaltungen können elektromagnetische Störstrahlung aussenden. Verlag und Autor haben keinen Einfluss auf die technische Ausführung der in diesem Buch vorgestellten Schaltungen. Der Anwender der Schaltung ist daher selbst für die Einhaltung der relevanten Emissionsgrenzwerte verantwortlich.
4. Durch Roboter können mechanische Beschädigungen entstehen und sogar Menschen verletzt werden. Beim Betrieb ist daher stets auf ausreichende Sicherheitsmaßnahmen zu achten. Roboterarme können z. T. erhebliche Kräfte entwickeln. Es ist daher auf ausreichende Sicherheitsabstände zu achten.
5. Bei der Arbeit mit höheren elektrischen Strömen besteht Brandgefahr. Insbesondere beim Einsatz von Akkumulatoren sind daher geeignete Sicherungen vorzusehen.

## **Programmdownload**

Alle Programme aus diesem Buch können unter

[www.elektor.de/robotik-und-ki](http://www.elektor.de/robotik-und-ki)

herunter geladen werden. Falls ein Programm nicht identisch mit dem im Buch beschriebenen sein sollte, dann sollte die Version aus dem Download verwendet werden, da diese aktueller ist.



## Kapitel 1 • Einführung

Beim Begriff "Roboter" denken viele Menschen noch immer an Science-Fiction und utopische Romane. Seit vielen Jahren sind sogenannte "Androiden" in der Literatur und in Filmen zu finden. Die Geschichte der fantasievollen Roboterwesen reicht weit zurück. Bereits seit der Antike wurde versucht, verschiedenste Tätigkeiten auf rein mechanischen Wegen zu erledigen. So entstanden erste automatische Theater und Maschinen, die in der Lage waren, einfache Musik zu erzeugen. Als die antiken Hochkulturen verschwanden, ging mit ihnen auch das wissenschaftlich-technische Wissen dazu verloren.

Heute führen die meisten Roboter Arbeiten aus, die für Menschen zu mühsam oder sogar zu gefährlich wären. Seit mehreren Jahren sind Robots in nahezu allen Industriebereichen zu finden. Weder in der Chemieproduktion noch in der Elektro- oder Automobilfertigung kommt man ohne sie aus. Kaum ein technische Gerät entsteht heute ohne ihre Mitwirkung. Smartphones, Autos oder Laptops wären unbezahlbar, wenn sie ohne Verwendung von Montagerobotern produziert werden müssten. Von der Raumfahrt bis hin zur Medizintechnik, in allen Bereichen sind die Universalmaschinen im Vormarsch. Sie eroberten in den letzten Jahren zudem auch ganz eigene Aufgabenbereiche. So erforschen bionische Unterwasserroboter die Ozeane, während der Mars Rover den roten Planeten unter die Lupe nimmt. Damit werden auch Gebiete der Forschung zugänglich, die für Menschen nur sehr schwer oder gar nicht erreichbar wären.

Seit der tschechische Schriftsteller Karel Čapek den Begriff "Roboter" geprägt hat, steht er für universell einsetzbare Automaten. Das Wort "Robota" stammt aus dem slawischen und steht für Fronarbeit bzw. ursprünglich Arbeit allgemein. Es wird erstmals im Jahr 1920 im Drama "R.U.R." verwendet. Der Name dieses Stücks steht für Rossum's Universal Robots, einer Firma, die künstliche Menschen erzeugt. Der Name "Rossum" ist zudem eine ironische Anspielung des Autors: Das tschechische Wort rozum bedeutet Vernunft, Verstand. Der Begriff Roboter wurde nach dem Erscheinen des Theaterstückes in zahlreiche Sprachen übernommen.

Im vorliegenden Buch soll eine Einführung in das hochaktuelle Gebiet der Robotik gegeben werden. Dabei soll es nicht nur bei theoretischen Betrachtungen bleiben. Vielmehr werden auch praktische Anwendungsbeispiele im Vordergrund stehen. Neben den technischen und mechanischen Grundlagen werden auch die elektronischen Komponenten und Module erläutert. Eine zentrale Rolle spielt dabei der Mikrocontroller.

Die Mikrocontrollertechnologie ist nach wie vor eines der faszinierendsten Gebiete der modernen Elektronik. Mikrocontroller haben in sich den letzten Jahren in praktisch allen Bereiche der modernen Technik etabliert. In zunehmendem Maße dringen sie auch in die Gebiete der Künstlichen Intelligenz und der Robotertechnik vor. Für die Programmierung kommen die Sprachen "C" und Python zum Einsatz. Diese nehmen auf dem Gebiet der Firmwareentwicklung eine dominierende Stellung ein. Die bei weitem überwiegende Mehrheit der professionellen Entwicklungsarbeit wird in den beiden Hochsprachen ausgeführt. Neben dem Arduino wird auch das Raspberry Pi-System eingesetzt. Bei Arduino-Anwendungen kommen stets C-Programme bzw. das speziellere Processing zum Einsatz. Bei der

Programmierung des Raspberry Pi wären prinzipiell sowohl C- als Python-Programme einsetzbar. Allerdings wird hier, dem allgemeinen Trend folgend, Python der Vorzug gegeben. Die Forschungsgebiete "Robotik" und "Künstliche Intelligenz" (KI oder AI für engl. Artificial Intelligence) waren ursprünglich zwei vollständig getrennte Fachbereiche. Allerdings zeigt sich seit einigen Jahren eine immer enger werdende Verflechtung der beiden Gebiete. Prinzipiell kann KI zwar vollkommen ohne Robotik-Elemente auskommen. Beispiele hierfür sind Spracherkennungssysteme oder die Gesichtserkennung. Universell einsetzbare Roboter sind dagegen kaum ohne KI vorstellbar. Mit lediglich regelbasierten Algorithmen werden Roboter niemals in der Lage sein, wirklich anspruchsvolle Aufgaben zu übernehmen. Dies wird bereits bei den aktuellen Forschungen zum autonom fahrenden Auto immer deutlicher. Immer wieder hat sich gezeigt, dass die reale Welt so komplex ist, dass klassische Automaten ohne Lernfähigkeit schnell an ihre Grenzen stoßen.

Im vorliegenden Buch soll die Philosophie des "Learning by doing" im Vordergrund stehen. Damit eignet es sich hervorragend als praktische Ergänzung für Unterricht, Seminare und Vorlesungen in

- FabLabs und Maker-Clubs
- Weiterführenden Schulen
- Technischen Berufsschulen und Fachakademien
- Hochschulen und Universitäten

Auch ambitionierte, nichtprofessionelle Anwender können sich mit dem Lernmaterial einen Überblick über den neuesten Stand der Robotertechnik und KI verschaffen.

Für praktische Anwendungen können sowohl komplette Bausätze als auch einzelne Komponenten verwendet werden. Dabei wurde stets darauf geachtet, dass die Hardware möglichst universell einsetzbar ist, so dass sowohl Bausätze als auch einzelne Komponenten in verschiedenen Projekten immer wieder verwendet werden können. Das Bezugsquellenverzeichnis am Ende des Buches erleichtert die Beschaffung der benötigten Bauteile und Komponenten.

### **1.1 Voraussetzungen**

Um dieses Buch erfolgreich durcharbeiten zu können, sind folgende Voraussetzungen erforderlich:

1. Sicherer Umgang mit dem Betriebssystem Windows oder Linux wird vorausgesetzt. Grundkenntnisse einer beliebigen Programmiersprache sind natürlich von Nutzen, aber nicht unbedingt erforderlich.
2. Spezielle Erfahrungen im Bereich Elektronik und Elektrotechnik sind nicht notwendig. Zum Verständnis der Übungen werden aber grundlegende Kenntnisse zu den Themen "Strom – Spannung – Widerstand" vorausgesetzt. Vorteilhaft sind auch elementares Wissen im Bereich der elektronischen Messtechnik. Im Bedarfsfall sind die entsprechenden Informationen dazu in der entsprechenden Fachliteratur nachzulesen (s. Literaturverzeichnis).

3. Grundkenntnisse in C und Python sollten vorhanden sein, wenn man die Programmbeispiele im Detail analysieren möchte. Hierzu steht umfangreiche Fachliteratur zur Verfügung (s. Literaturverzeichnis).

Um die Praxisbeispiele in diesem Buch durcharbeiten zu können, ist das folgende Grundmaterial erforderlich:

- PC oder Laptop mit USB-Schnittstelle
- Internetzugang

Die einzelnen Hardware- und Roboterkomponenten werden in den jeweiligen Projekten ausführlich beschrieben. Häufig wird auch auf komplette Material- oder Bausätze verwiesen, da der Kauf eines kompletten Bausatzes häufig deutlich günstiger ist als die Bestellung einzelner Komponenten.

Um den maximalen Nutzen aus den Bauteilen ziehen zu können, wurde stets darauf geachtet, dass diese möglichst universell einsetzbar sind. So können die meisten beschriebenen Sensoren und Motortreiber usw. sowohl an einem Arduino als auch am Raspberry Pi betrieben werden. Neben der Verwendung von Einzelkomponenten kommen auch immer wieder komplette Module zum Einsatz. Diese finden sich sowohl in den angegebenen Bausätzen als auch in den diversen Online-Shops wie Amazon oder Alibaba, so dass deren Beschaffung keine Probleme bereiten sollte.

## 1.2 Die Programmier- und Entwicklungsumgebungen

Für Robotik-Anwendungen haben sich zwei Controller-Boards besonders etabliert:

- Arduino, insbesondere Arduino UNO, NANO und Leonardo
- Raspberry Pi

Beim den ersteren handelt es sich um ein reine Controller-Boards. Diese benötigen zur Programmierung stets einen Laptop oder PC. Über eine USB-Verbindung werden die fertig kompilierten Programm dann zum Board übertragen. Die Arduino-Boards werden über eine spezielle, anfängerfreundliche Benutzeroberfläche (IDE = Integrated Design Environment = Integrierte Entwicklungsoberfläche) programmiert. Der große Vorteil im Vergleich zu einer klassischen "Tool-Chain" liegt darin, dass sie sehr intuitiv bedient werden kann. Neben dem Arduino-Board selbst ist diese spezielle Entwicklungsumgebung sicher einer der Hauptfaktoren für den großen Erfolg des Arduino-Konzeptes.

Der Raspberry Pi dagegen ist ein vollständiges kleines Motherboard. Zusammen mit

- SD-Karte als Massenspeicher
- USB-Tastatur
- USB-Maus
- HDMI-Monitor
- Netzteil

wird daraus ein kompletter und leistungsfähiger Linux-Rechner. Ein zusätzlicher PC oder Laptop ist dann nicht mehr erforderlich.

## **Kapitel 2 • Grundelemente der Robotik**

Der besondere Reiz, aber auch die Problematik bei der Beschäftigung mit der Robotik liegt darin, dass eine Vielzahl von technischen Disziplinen erforderlich sind, um einen vollständigen Roboter aufzubauen. So sind mehr oder weniger detaillierte Kenntnisse in den folgenden Technologiebereichen notwendig:

- Mikrocontrollertechnik
- Programmierung
- Elektrotechnik und Elektronik
- Sensortechnik
- Mechanik

Die folgenden Kapitel geben jeweils Einführungen in diese Gebiete. Natürlich können die einzelnen Bereiche nicht immer in ihrem vollen Umfang dargelegt werden. Insbesondere für Einsteiger in ein bestimmtes Gebiet kann es daher sinnvoll sein, weitere Literatur zu Rate zu ziehen. Das Literaturverzeichnis am Ende des Buches kann dabei Hilfestellung leisten.

## Kapitel 3 • Controller und Prozessoren: Die "Gehirne" der Roboter

Moderne Roboter kommen praktisch nicht mehr ohne einen eigenen Controller oder Prozessor aus. Diese Komponenten ist daher eine der zentralen Einheiten in der Robotik geworden. Der Controller/Prozessor stellt sozusagen das "Gehirn" der Maschine dar. Hier erfolgt die Erfassung von Sensorwerten oder die Ausgabe von Steuerbefehlen für die mechanischen Einheiten des Robots.

Prinzipiell könnten Roboter zwar auch ohne programmierbare Komponenten auskommen. So existieren etwa Konzepte und Bausätze, die Konstruktionen ermöglichen, welche auf rein analoger Basis arbeiten. Eine Hinderniserkennung und -umfahrung kann beispielsweise allein mit optischen Sensoren und Operationsverstärkern umgesetzt werden. Allerdings findet diese Technik auch rasch ihre Grenzen. Komplexere Algorithmen wie etwa kontrolliertes Zurücksetzen oder aber die Erkennung verschiedener Hindernisse lassen sich kaum mehr mit rein analog-elektronischen Schaltungen realisieren.

Andere Aufgaben wie etwa Bild- oder Mustererkennung sind mit rein analogen Mitteln ohnehin kaum lösbar. Spätestens hier muss man auf digitale Komponenten zurückgreifen. Neben der Digitalisierung spielt auch die Programmierbarkeit eine wesentliche Rolle. Fest verdrahtete Systeme können nur noch mit erheblichem Aufwand verändert und verbessert werden. Ein mit Software umgesetzter Algorithmus kann dagegen problemlos kontinuierlich angepasst, modifiziert und optimiert werden. Auf diese Art und Weise können einem Roboter immer wieder neue Lösungswege "beigebracht" werden. Im Rahmen dieses Buches werden vor allem die beiden bereits in der Einleitung angesprochenen Systeme "Arduino" und "Raspberry Pi" eingesetzt, um den Robotern maximale Flexibilität zu verleihen. Die nächsten Kapitel beschäftigen sich daher eingehender mit den beiden beliebten Boards.

### 3.1 Arduino als programmierbare Steuerzentrale

Das System "Arduino" stellt eine Open-Source-Plattform für den Bau von Mikrocontroller- und Elektronikprojekten dar. Es besteht aus einer Platine mit einem programmierbaren ATmega328 Mikrocontroller und einer Software, der sogenannten IDE (**I**ntegrated **D**evelopment **E**nvironment), die auf einem PC oder Laptop ausgeführt wird. Mit Hilfe der IDE werden Programme, auch als "Sketches" bezeichnet, entwickelt und anschließend auf den Controller hochgeladen.

Die Arduino-Plattform erfreut sich bei Elektroneinsteigern seit über zehn Jahren größter Beliebtheit. Neben einer Vielzahl von Sensoren können auch Aktoren wie Servos oder Motoren mit dem Arduino problemlos angesteuert werden. Dies macht ihn auch für Robotik-Anwendungen zum Mittel der Wahl.

Im Gegensatz zu früheren Mikrocontroller-Systemen benötigt der Arduino keine separate Hardware wie etwa Programmiergeräte oder EEPROM-Brenner, um neuen Code auf den Controller zu laden. Ein einfaches USB-Kabel genügt, um den Arduino zu programmieren. Darüber hinaus verwendet die Arduino IDE eine vereinfachte Version von C++, um das Programmieren zu erleichtern.

Arduinos sind inzwischen in einer nahezu unüberschaubaren Anzahl von Varianten verfügbar. Neben dem klassischen Arduino sind die verschiedensten Größen und Formen erhältlich. Genannt sei hier der Arduino MEGA, der mit einem ATmega1280 oder ATmega2560 bestückt ist. Diese Prozessoren weisen einen erheblich erweiterten Funktionsumfang auf und auch die Anzahl der verfügbaren Pins ist deutlich größer. Wegen der großen Anzahl von I/O Pins haben die Boards auch etwa die doppelte Größe eines klassischen Arduinos.

Wenn man dagegen möglichst kleine und kompakte Geräte aufbauen möchte, kann man auf die Mikro- oder Nano-Versionen zurückgreifen. Diese Boards haben nur noch die Größe einer Briefmarke. Anstelle der Kontaktbuchsen weisen sie lediglich Lötunkte auf. Diese können direkt mit Kabeln verlötet werden. Alternativ sind hier auch Stiftleisten einlötlbar, sodass diese kompakten Boards direkt in ein sogenanntes Breadboard oder aber in eine IC-Fassung eingesetzt werden können.

Als Goldstandard hat sich allerdings der Arduino UNO herauskristallisiert. Wird gemeinhin von einem Arduino gesprochen, so ist meist der "UNO" gemeint (s. Abb.).



Abbildung 3.1: Arduino Uno

Die folgende Tabelle fasst die Technischen Daten des UNOs zusammen:

Mikrocontroller	ATmega328P
Betriebsspannung	5 V
Digitale I/O Pins	14
PWM Kanäle	6
Analoge Eingänge	6
Serielle Ports (USART)	1
I <sup>2</sup> C-Ports	1
SPI-Ports	1
Timer	3

Der Vollständigkeit halber hier noch auf die unterschiedlichen Pin-Abstände der Arduino-Buchsen hingewiesen. Bei der Entwicklung der klassischen Arduino-Variante wurden nicht alle Buchsen im gängigen Rastermaß von 1/10 Zoll bzw. 2,54 mm angeordnet. So weisen die beiden oberen Buchsenleisten einen Abstand von nur 1/20 Zoll auf. Damit lassen sich Lochrasterplatinen mit Standardrastermaß nicht direkt über Stiftleisten mit allen Buchsen verbinden. Hier muss man also immer auf Sonderlösungen zurückgreifen.

### 3.2 Arduino-IDE

Das für die Programmierung des Arduinos erforderliche Programmpaket kann unter

[www.arduino.cc/en/Main/Software](http://www.arduino.cc/en/Main/Software)

kostenlos aus dem Internet geladen werden. In dieser "Arduino-Programmierungsumgebung" werden die Programme erstellt, die der Mikrocontroller des Arduinos später ausführt. Im Arduino-Umfeld sind diese Programme auch unter dem Namen "Sketch" bekannt. Für den Download der Software stehen zwei Optionen zur Verfügung:

1. ein Installationspaket
2. eine ZIP-Datei

Im ersten Fall wird die Programmieroberfläche durch den Aufruf einer Installationsdatei auf dem Rechner installiert. Im zweiten Fall muss man die gepackte Datei herunterladen und in ein eigenes Verzeichnis entpacken.

Ist die Installation abgeschlossen, öffnet man den Softwareordner und startet das Programm mit der Datei `arduino.exe`. Zunächst sind nun zwei wichtige Einstellungen vorzunehmen:

Als erstes muss das richtige Board ausgewählt werden. Im Fall des Arduino UNO wird der Board-Typ "Arduino UNO" verwendet. Die entsprechende Auswahl kann unter

Werkzeuge → Board

getroffen werden.

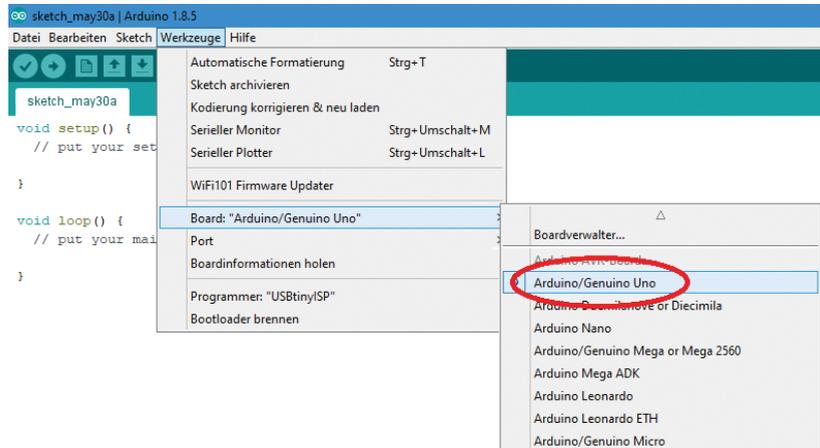


Abbildung 3.2: Auswahl des richtigen Boardtyps

Als Nächstes ist die korrekte Schnittstelle auszuwählen. Auf diese Weise wird dem PC mitgeteilt, an welchem Port-Anschluss das aktuelle Arduino-Board angeschlossen ist. Ist der Arduino mit dem Rechner verbunden, erscheint in der Port-Auswahlliste ein neuer Menüpunkt, z. B.

COM4

Diese Schnittstelle muss nun aktiviert werden. Damit ist die Verbindung fertiggestellt.

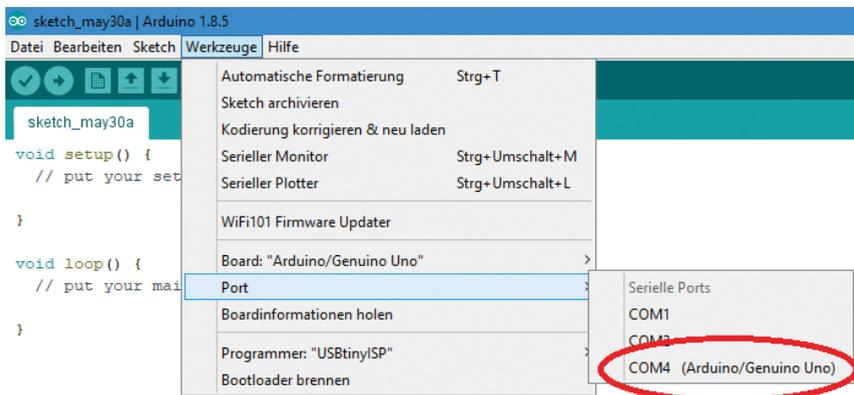


Abbildung 3.3: Auswahl des COM-Ports

Nun ist das Board mit dem Rechner verbunden und das erste Programm kann auf den Controller geladen werden. Unter

Datei → Beispiel → 01.Basics → Blink

kann ein einfacher Sketch geöffnet werden, der eine LED blinken lässt. Mit

Sketch → Hochladen

wird dieses erste Programm auf das Controller-Board geschrieben.

Wenn alles korrekt ausgeführt wurde, sollte nun die mit "13" oder "L" bezeichnete LED blinken. Damit wurde bereits eine erste programmgesteuerte Anwendung realisiert!

Die Abbildungen und Tests beziehen sich auf die Arduino-Programm-Version 1.8.5 unter Windows 10. Bei Verwendung anderer Versionen oder anderer Betriebssystem-Varianten können sich leichte Abweichungen ergeben.

Alle Programme in diesem Buch wurden mit dieser Konfiguration erstellt und geprüft. Üblicherweise sind Arduino-Sketches aufwärtskompatibel, d. h. Sketches, die auf älteren Programmierumgebungen erstellt wurden, laufen auch auf den neuen Versionen. Allerdings gibt es auch immer wieder Ausnahmen von dieser Regel. Falls es also zu Problemen kommen sollte, ist es empfehlenswert, die betreffenden Sketches mit der Version 1.8.5 zu verarbeiten.

In der hier vorgestellten Konfiguration ist keine aktive Installation von USB-Treibern mehr erforderlich. Bei älteren Betriebssystem-Versionen kann es allerdings erforderlich sein, solche Treiber manuell zu installieren. In diesem Fall meldet der PC, dass zwar das angeschlossene Board erkannt wurde, aber dennoch eine Treiberinstallation erforderlich ist. Häufig wird der erforderliche Treiber nicht automatisch geladen. Man muss diesen dann im Verlauf der Installation selber auswählen. Er befindet sich in dem Arduino-Programmordner und dort Unterordner "Drivers".

Abschließend sollte man eine einfache Kontrolle durchführen: In der Systemsteuerung des Computers findet man u. A. den "Gerätemanager". Nach einer erfolgreichen Installation ist das Arduino-Board hier aufgelistet. Andernfalls erscheint ein unbekanntes USB-Gerät, gekennzeichnet mit einem gelben Ausrufezeichen. In diesen Fall ist das unbekannte Gerät anzuklicken und auf die Auswahl "Treiber aktualisieren" zu aktivieren. Nun kann der Ablauf der manuellen Installation erneut durchgeführt werden.

### **3.3 Praxisanwendung: Roboter-Alarm**

Als erste Praxisanwendung für den Arduino kann ein Alarmsignal aufgebaut werden. Derartige Geräte werden durchaus auch in verschiedenen Variationen eingesetzt. Sie bestehen aus einer LED, die in regelmäßigen Abständen kurz aufblitzt, und werden an geeigneter Stelle, beispielsweise in einem Fahrzeug, eingebaut. Sie sind von einer scharf geschalteten Alarmanlage nicht zu unterscheiden und zumindest Gelegenheitsdiebe lassen sich damit von einem Einbruch abschrecken. Im Robotik-Bereich finden entsprechende Blinklichter als Warn- oder Positionsleuchten ihren Einsatz. In industriellen Anwendungen kann damit beispielsweise die Betriebsbereitschaft eines Roboterarms signalisiert werden. Das Wartungspersonal sollte sich dann aus Sicherheitsgründen nicht mehr im Zugriffsbereich des Roboterarms aufhalten. Das zugehörige Programm sieht so aus und findet sich auch im Downloadpaket zu diesem Buch:

```
// Alarm_simulator.ino
// IDE 1.8.5

int led = 13;

void setup()
{ pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(100);
  digitalWrite(led, LOW);
  delay(3000);
}
```

Die Unterschiede zum Blink-Sketch bestehen in den folgenden Änderungen:

- Die Kommentarzeilen, welche mit "//" beginnen, wurden angepasst
- Die Werte in der Funktion "delay" wurden von jeweils 1000 auf 100 bzw. 3000 geändert. Damit wird aus dem regelmäßigen Blinken ein kurzes Aufblitzen.

Wenn der neue Sketch geladen wird, blitzt die angeschlossene LED nun im Abstand von jeweils 3 Sekunden kurz auf. Damit ist der Warnblitzer einsatzbereit.

Für eine reale Anwendung könnte man nun eine externe LED über einen Vorwiderstand und etwas längere Drähte mit dem Board verbinden und diese an geeigneter Stelle an einem Roboter anbringen. Damit wäre dieser auch bei ungünstigen Lichtverhältnissen nicht mehr zu übersehen.

### 3.4 C und "Processing" – Ein Vergleich

Die Processing-Programme der Arduino-IDE sind auch für Einsteiger leicht zu lesen. Dennoch basiert Processing auf der klassischen Programmiersprache "C". Die in C so beliebten Abkürzungen werden hier jedoch vermieden und durch wesentlich leichter verständliche Anweisungen ersetzt. Man kann ohne große Vorkenntnisse schon vermuten, dass eine Leuchtdiode (LED) in einem bestimmten Rhythmus geschaltet wird.

Ein entsprechendes Standard-C-Programm ist wesentlich schwieriger zu durchschauen:

```
// Blink_C.ino
// UNO @ IDE 1.8.5

#define F_CPU 16e6
#include <avr/io.h>
#include <util/delay.h>
```

```

int main(void)
{ DDRB = 0b00100000;
  while (1)
  { PORTB|=(1<<5);
    _delay_ms(1000);
    PORTB&=~(1<<5);
    _delay_ms(1000);
  }
  return 0;
}

```

Bitmanipulationen und logische Operatoren (|, &, ~, << etc.) lassen den embedded C-Code schwer lesbar erscheinen. Dennoch ist hier auch die enge Verwandtschaft zwischen Arduino-Processing und C erkennbar.

Bei den einfachen Anweisungen wie "digitalWrite()" handelt es sich lediglich um vorgefertigte Funktionen und Unterprogramme, die den Einstieg in die Programmierung erleichtern. Deutliche Unterschiede zeigen sich auch in der Programmgröße im .hex-Format, also den vom Prozessor lesbaren Code:

Blink_1s.ino:	940 Bytes
Blink_C.ino:	178 Bytes

Der auf C basierende Code benötigt also weniger als 20% des Speicherplatzes im Vergleich zum von Processing erzeugten Programm. Auch in der Ablaufgeschwindigkeit ist der reine C-Code der Processing-Version deutlich überlegen. Dies zeigt sich in den maximal erreichbaren Blinkfrequenzen (ohne delay):

C:	2.667 MHz = 16 MHz / 6 Taktzyklen
Processing:	0.145 MHz = 16 MHz / 110 Taktzyklen

Bei geschwindigkeitskritischen Roboter-Steuerungen kann es also durchaus vorteilhaft sein, wenn man die Arduino-Anweisungen durch klassische C-Befehle ersetzt.

### 3.5 Raspberry Pi

Der Raspberry Pi ist ein Mini-Computer-Motherboard mit ARM-Prozessor. Ursprünglich war der "RasPi" oder "Pi" dafür gedacht, Kindern und Jugendlichen die moderne Computertechnik und Programmierung wieder näher zu bringen. Allerdings hat sich die Platine schnell auch zum Liebling erwachsener Elektronik-Enthusiasten entwickelt. Darüber hinaus eignet sich die Platine auch bestens für Robotik-Anwendungen, da sie sowohl über ein Kamera-Interface als auch über eine Reihe von frei programmierbaren Eingabe/Ausgabe-Pins verfügt. Das Herz der Raspberry Pi ist ein Ein-Chip-System von Broadcom. Die Abmessungen des RasPi entsprechen etwa denen einer Kreditkarte. Der Raspberry Pi kam Anfang 2012 auf den Markt und bis Ende 2018 wurden weit über 20 Millionen Einheiten verkauft. Das meistverkaufte Modell ist aktuell der Pi 3B(+), der rund 50 Prozent der Verkaufszahlen ausmacht.

Inzwischen existiert ein umfangreiches Zubehör- und Softwareangebot für zahlreiche Anwendungsbereiche. Als Betriebssystem kommen vor allem speziell adaptierte Linux-Distributionen mit grafischer Benutzeroberfläche zum Einsatz.

Als Massenspeicher dient eine auswechselbare SD-Speicherkarte. Zusätzliche Speichermedien können per USB-Schnittstelle angeschlossen werden. So arbeitet der Pi auch problemlos mit externen Festplatten oder USB-Speichersticks zusammen. Neuere Modelle (ab Version 3) können zudem drahtlos via Bluetooth oder WLAN kommunizieren.

Die wichtigsten Leistungsmerkmale des RasPi sind:

- die geringen Abmessungen (85,6 mm x 56,0 mm)
- der günstige Preis (unter 40 Euro)
- die einfache Erweiterbarkeit

Zum Anschluss von Peripheriegeräten, Sensoren und elektronischen Bauelementen stehen 21 GPIO-Pins zur Verfügung. Für die Verbindung mit einem Monitor ist eine HDMI-Schnittstelle vorgesehen. Über diese lassen sich Bildinhalte mit Full-HD-Auflösung von bis zu 1080p wiedergeben.

Das Betriebssystem startet der Rechner von einer Speicherkarte, die in den vorhandenen SD-/MMC-Karten-Slot eingesteckt wird. Das System kann u. A. mit verschiedenen Linux-Versionen betrieben werden.

Typische Anwendungen für den Raspberry Pi Minicomputer sind Mediaplayer, Spielekonsolen, NAS- und Cloud-Server, VPN-Server oder intelligente Steuerungen für das Internet of Things (IoT) und Smart Home. Darüber hinaus hat sich der Pi aber auch als zentrale Steuereinheit für Robotik-Systeme etabliert. Insbesondere die Möglichkeit, über die GPIO-Schnittstelle mit elektronischen Komponenten aller Art zu kommunizieren, macht den Pi für Robotikanwendungen so attraktiv. Damit ist der Pi bestens für die Erfassung, Auswertung von Sensorwerten oder zur Ansteuerung von Motortreibern etc. geeignet.

Zusammen mit der Programmiersprache Python bietet der Raspberry eine optimale Umgebung für die Umsetzung verschiedenster Robotik-Projekte. Vor allem im Bereich der Sensortechnik verfügt der Pi über Möglichkeiten, die früher nur teuren und aufwändigen Spezialsystemen vorbehalten waren.

Von besonderem Vorteil ist die Option, den Pi mit Kameras auszurüsten. Damit werden in der Robotik auch hochinteressante Anwendungen im Bereich der Maschinellen Sehens möglich. Zusammen mit der CV (**C**omputer**V**ision)-Bibliothek für Python können damit Projekte wie die Erkennung oder sogar Verfolgung von Objekten umgesetzt werden.

Für den Anschluss von Kameras stehen zwei Optionen zur Verfügung:

- Raspberry-Pi-Cam
- WebCams via USB

Beide Varianten werden in späteren Kapiteln vorgestellt. Im Rahmen praktischer Anwendungsbeispiele wird auf die Vor- und Nachteile der beiden Möglichkeiten eingegangen.

Allerdings weist die Platine auch einige Nachteile auf:

- Die I/O-Ports sind direkt mit dem Prozessoreingängen verbunden. Dies bedeutet, dass sie vollkommen ungeschützt sind. Zu hohe Eingangsspannungen können daher leicht den Prozessor und damit den gesamten Raspberry Pi zerstören.
- Die Pins sind nicht 5-Volt-tolerant
- Es stehen keine ADCs / DACs zur Verfügung. Für die Auswertung analoger Spannungen ist daher immer ein zusätzlicher Baustein erforderlich.

Die folgende Abbildung zeigt den Raspberry Pi mit seinen verschiedenen Funktionseinheiten und Schnittstellen. Die GPIO-Pinleiste ist in diesem Bild links oben zu erkennen.



*Abbildung 3.4: Das Raspberry Pi Board*

Der Raspberry Pi und die SD-Karte mit dem Betriebssystem werden meist nicht als fertiges Komplettsystem geliefert. Um mit dem Pi arbeiten zu können, muss das aktuelle Betriebssystem auf die die SD-Karte geladen werden. Damit lassen sich dann alle Funktionen der Platine einschließlich der Pin-Leiste des WLAN-Interfaces und der Bluetooth-Schnittstelle etc. ansteuern. Die folgende Abbildung liefert einen Überblick über diese einzelnen Funktionseinheiten eines Raspberry Pi:

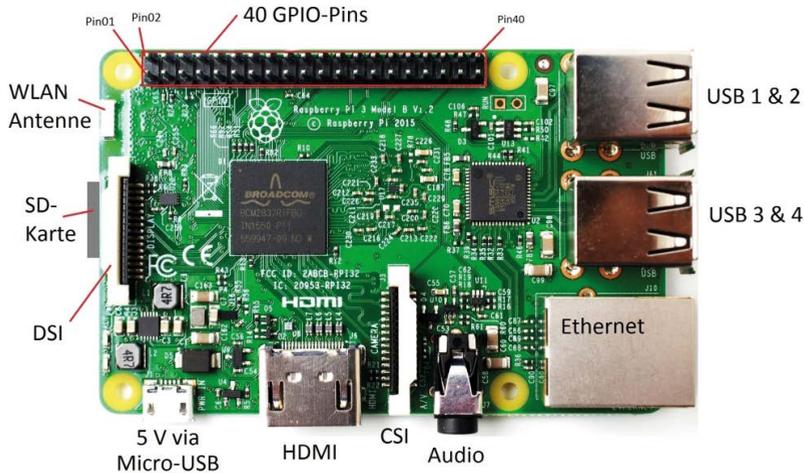


Abbildung 3.5: Funktionseinheiten auf dem Raspberry Pi

Das folgende Kapitel beschreibt, wie das Raspbian-System installiert wird. Alternativ können auch komplette Systeme mit vorinstalliertem Betriebssystem erworben werden. Diese liegen allerdings im Preis meist etwas höher.

### 3.6 Installation des Betriebssystems mit NOOBS

Für die Installation des Betriebssystems steht ein NOOBS-Lite-Programmpaket als ZIP-Archiv unter

<https://www.raspberrypi.org/downloads/noobs/>

im Internet zur Verfügung. Die Bezeichnung "NOOBS" steht dabei für **New Out of the Box Software**-Paket. Nach dem Download ist das Archiv zu entpacken. Nun kann man die Dateien auf eine neue oder neu formatierte SD-Karte kopieren. Anschließend wird nur noch die Karte in den Pi gesteckt. Nach dem Anschließen der Versorgungsspannung bootet das System automatisch. Dann wird das neue Betriebssystem vollständig installiert. Voraussetzung für diese weitgehend automatische Installation ist, dass der Pi über ein LAN-Kabel mit dem Internet verbunden ist. Diese Art der Installation kann eine Stunde oder auch länger in Anspruch nehmen.

Alternativ kann auch das vollständige NOOBS-Programmpaket von der gleichen Internetseite geladen werden. Diese muss ebenfalls entpackt werden und kann dann mittels eines Kopierprogramms wie etwa dem Diskmanager direkt auf eine SD-Karte geschrieben werden.

### 3.7 Raspi-Config

Über die Anweisung

```
sudo raspi-config
```

kann im LXTerminal (s. Abb. 8.1) das sogenannte Konfigurationsmenü aufgerufen werden.

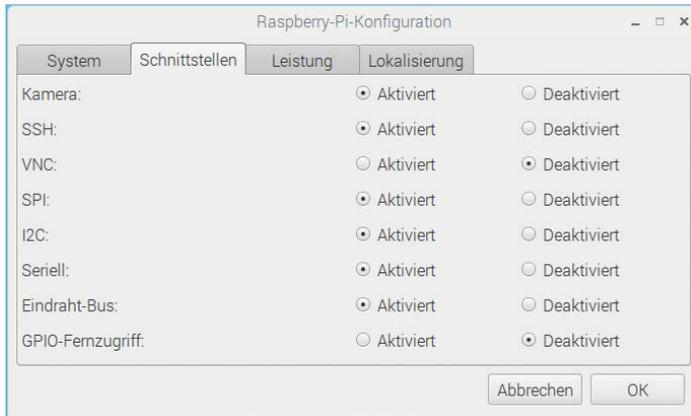


Abbildung 3.6: Der Konfigurationsbildschirm des Pi

In diesem Konfigurationstool (raspi-config) kann man verschiedene Grundeinstellungen vornehmen. Unter anderem kann man die folgenden Optionen einstellen:

- Deutsche Tastatur
- Zeitzone: Europe - Berlin
- Freigabe von Schnittstellen
- Freigabe der Kameraschnittstelle

Abschließend wird mit "OK " der Konfigurationsvorgang beendet.

Das Konfigurationsmenü kann jederzeit über das Startmenü oder durch Eingabe von sudo raspi-config in einer Kommandozeile aufgerufen werden.

### 3.8 Die Kommandozeile

Im Allgemeinen bootet der Raspberry direkt in eine graphische Oberfläche. Nur in Ausnahmefällen bleibt er mit einer Eingabeaufforderung (englisch: Prompt) stehen, die wie folgt aussieht:

```
pi@raspberrypi ~ $
```

Dabei ist pi der User und raspberrypi der Name des Raspberry Pi. Die Eingabeaufforderung besagt, dass man nun als User "pi" auf dem Raspberry mit dem Namen "raspberrypi" eingeloggt ist.

Es folgt die Anzeige des aktuellen Verzeichnisses. Die Tilde (~) steht für das User-Verzeichnis des aktuellen Users (hier: /home/pi ).

Das Dollarzeichen zeigt an, dass es sich um einen Prompt handelt, hinter dem nun Linux-Befehle eingegeben werden können. Dabei ist immer auf Groß-/Kleinschreibung zu achten

Als Beispiel könnte der Befehl `date` eingegeben werden, welcher das Betriebssystem dazu bringt, das aktuelle Systemdatum und die Uhrzeit anzuzeigen.

Dieses Terminal kann jederzeit auch über die grafische Benutzeroberfläche gestartet werden (Terminal ">\_" in der Taskleiste).

Wie in jedem Linux-System können sich auch beim RasPi verschiedene User anmelden. Der Standarduser ist allerdings:

```
pi
```

Das zugehörige Standardpasswort lautet:

```
raspberry
```

Sollte man sich einmal versehentlich ausgeloggt haben, kann man damit wieder die gewohnte Umgebung zurückholen.

### 3.9 Das Linux-Dateisystem

Das für den Raspberry maßgeschneiderte Betriebssystem Raspbian ist eine spezielle Linux-Version. Bei Linux wird fast alles als Datei dargestellt. Dateien werden in Ordnern in einer baumartigen Struktur gespeichert:

```

/
bin
boot
dev
etc
home
    - pi
        - desktop
        - python_games
        - sonstiges_Verzeichnis
usr
var
```

Diese Struktur ist zunächst nicht von großer Bedeutung. Später wird sie allerdings wichtig, insbesondere wenn verschiedene Dateien gesucht oder abgelegt werden müssen. In diesem Fall kann man sich am oben angegebenen Dateibaum orientieren.

### 3.10 Die wichtigsten Linux-Befehle

Bei der Arbeit mit dem Pi ist zu beachten, dass bei Linux zwischen Groß- und Kleinschreibung unterschieden wird. Die folgende Tabelle zeigt eine Übersicht zu den wichtigsten Linux-Befehlen. Diese werden benötigt, wenn man mit dem sogenannten Terminal arbeitet:

<code>ls</code>	Dateien auflisten
<code>ls-l</code> bzw. <code>ls-lh</code>	Dateien übersichtlich inkl. Zugriffsrechten auflisten
<code>startx</code>	grafische Oberfläche starten
<code>shutdown -h now</code>	Raspberry Pi herunterfahren (-h = halt)
<code>shutdown -r now</code>	Raspberry Pi neu starten (-r = restart)
<code>cd xyz</code>	in Verzeichnis xyz wechseln
<code>cd ..</code>	ein Verzeichnis höher gehen
<code>mkdir xyz</code>	Verzeichnis erstellen (make directory)
<code>rmdir xyz</code>	Verzeichnis löschen (remove directory)
<code>nano datei</code>	Datei editieren (im Editor: Strg-o für Speichern Strg-x für Exit)
<code>rm datei</code>	Datei löschen
<code>pwd</code>	aktuelles Verzeichnis (print current workingdir), z. B. <code>/home/pi</code>
<code>whoami</code>	aktuellen Benutzernamen anzeigen, z. B. <code>pi</code>
<code>date</code>	Datum und Uhrzeit anzeigen
<code>ifconfig</code>	IP-Adresse usw. anzeigen
<code>whatis befehl</code>	Grobbeschreibung von befehl, z. B. <code>whatis ls</code>
<code>man befehl</code>	ausführliche Beschreibung von befehl, z. B. <code>man ls</code>
<code>df / -h</code>	Speicherkartengröße anzeigen
<code>lsusb</code>	angeschlossene USB-Geräte anzeigen
<code>mv datA datB</code>	Datei verschieben oder umbenennen

### 3.11 Der Desktop des Pi

Will man mit dem Raspberry Pi so arbeiten, wie man es von einem PC her gewohnt ist, muss man die graphische Benutzeroberfläche starten. Damit ist der RasPi einem Windows-PC von der Bedienung her praktisch gleichwertig. Standardmäßig wird diese Oberfläche bereits beim Booten aktiviert. Sollte dies einmal nicht der Fall sein oder wurde die grafische Oberfläche beendet, sein kann sie mit dem Befehl

```
sudo startx
```

in der Kommandozeile neu aktiviert werden.

Praktisch alle Arbeiten können auch über die grafische Oberfläche erledigt werden. Auch der Internetzugriff über LAN oder WLAN ist problemlos möglich. Dies hat den Vorteil, dass man sich so alle nötigen Informationen, Treiber oder Programme etc. direkt aus dem Internet auf den Pi laden kann.