

FLORIAN SCHÄFFER

ROBOTER BAUEN & PROGRAMMIEREN FÜR KIDS

EINFACHER EINSTIEG IN ELEKTRONIK,
ROBOTIK UND MECHANIK

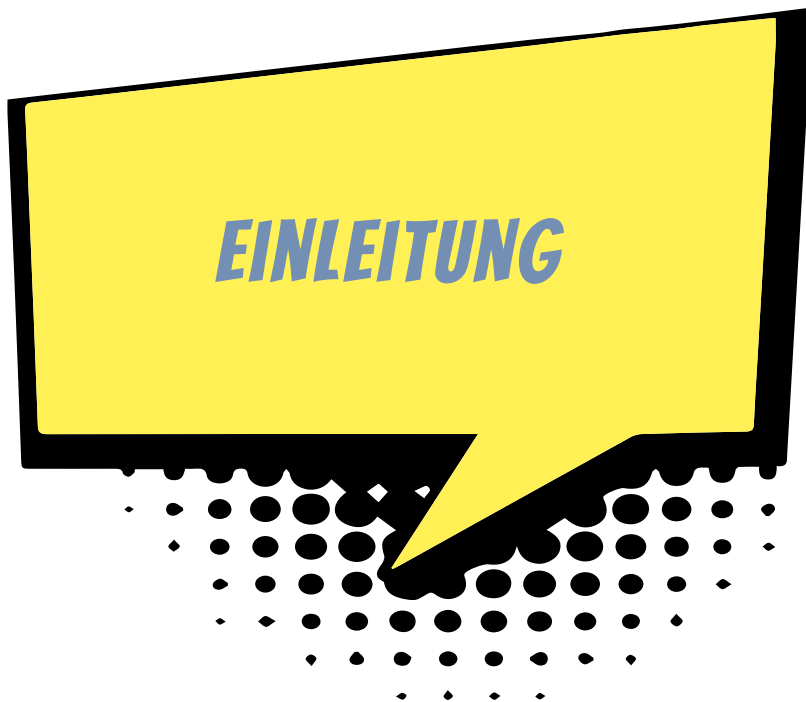


INHALT

EINLEITUNG	9	
So arbeitest du mit diesem Buch	10	
Was du für dieses Buch schon können solltest	11	
Dateien zum Download	11	
Fragen und Aufgaben am Ende der Kapitel	12	
Was dich in diesem Buch erwartet	12	
SCHRUBBI, DER BÜRSTEN-KÄFER	15	1
Was ist überhaupt ein Roboter?	17	
Noch mehr Input	25	
Ein paar Fragen	26	
... und ein paar Aufgaben	26	
HALLO ARDUINO	27	2
Wir lassen eine LED leuchten	28	
Erste Schritte mit Arduino und NEPO	32	
Arduino? Was ist das überhaupt?	54	
micro:Bit, Raspberry Pi, Calliope und mehr	61	
Was ist wo beim Arduino Uno?	64	
Mit Software wird Hardware intelligent	67	
Ein paar Fragen	72	
... und ein paar Aufgaben	72	
PROGRAMMIEREN MIT GRAFIK-BLÖCKEN	73	3
Die Oberfläche von Open Roberta Lab erkunden	74	
Aktionen	82	
Sensoren	86	
Kontrolle	87	
Logik	90	

	Mathematik	91
	Text	93
	Variablen	93
	Funktionen	95
	Ein paar Fragen	98
	... und ein paar Aufgaben	98
4	DAS ROBOTER-FAHRGESTELL ENTSTEHT	99
	Ein Blick in den Motor und das Getriebe	100
	Mit Vollgas gegen die Wand	106
	Der Arduino gibt Gas und bremst	116
	Geschwindigkeitskontrolle	132
	Kurven fahren	135
	Gefahrenre Strecke messen	138
	Ein paar Fragen	145
	... und ein paar Aufgaben	145
5	DER ROBOTER WIRD EIGENSTÄNDIG	147
	Achtung, ich komme! Hupe und Beleuchtung	148
	Gegen die Wand und dann immer weiter	152
	Schildkröti: Linienfolger mit Intelligenz	157
	Pfiffi folgt dir überall hin	163
	Eine bessere Hinderniserkennung	166
	Mit Ultraschall Entfernung genauer messen	169
	Ein paar Fragen	177
	... und ein paar Aufgaben	178
6	SERVO-MOTOREN FÜR PRÄZISE POSITIONIERUNG	179
	Katapult mit Zielerkennung	179
	Industrieroboter-Arm	189
	Auf zwei Beinen laufen	199
	Sechsfüßiger Krabbelkäfer: Hexapoden-Bot	209
	Aus einem Irrgarten entkommen	214
	Ein paar Fragen	233
	... und ein paar Aufgaben	234

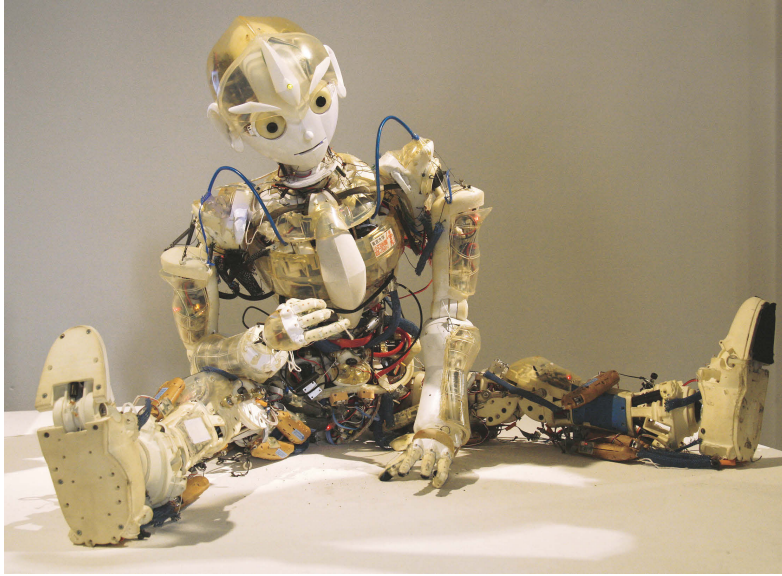
EIN ABSTECHER IN ANDERE WELTEN	235	7
Spiel mit Scratch programmieren	236	
LEGO Mindstorms EV3 simulieren	244	
Schaltungen und Programm online ausprobieren	249	
Ein paar Fragen	261	
... und ein paar Aufgaben	262	
MATERIAL- UND EINKAUFLISTE	263	A
ARBEITSTECHNIKEN	273	B
LÖSUNGEN ZU DEN FRAGEN UND AUFGABEN	285	C
STICHWORTVERZEICHNIS	297	



Willkommen bei deinem neuen Abenteuer als Roboterentwickler! Du bist schon immer von Robotern fasziniert gewesen und willst selber welche bauen? Dann bist du hier genau richtig: Erlebe, wie einfach es ist, eigene Roboter zu schaffen und zu kontrollieren.

Dies ist ein Buch mit viel Praxis: Es gibt viel auszuprobieren und zu basteln. Dich aufs Sofa zu legen und alles durchzulesen, reicht nicht aus und macht viel weniger Spaß. Wann immer Handanlegen gefragt ist, wird dir schrittweise gezeigt, was zu machen ist. Es lohnt sich, auch einfache Aufgaben (die du vielleicht sogar schon kennst) auszuprobieren, um dadurch Erfahrung und Routine zu sammeln.

Beim Selbermachen kann immer irgendetwas schief laufen und nicht klappen. Vielleicht ist uns auch mal ein Fehler bei der Beschreibung unterlaufen oder die Anleitung war nicht verständlich genug. Das ist schade, manchmal ärgerlich, aber eigentlich sogar gut. Wenn etwas nicht auf Anhieb so funktioniert wie gewollt, dann kannst du alles hinwerfen und aufgeben oder du suchst den Fehler. Das ist manchmal gar nicht einfach und du wirst Hilfe (zum Beispiel von deinen Eltern oder einem Lehrer) benötigen – frage sie! Hast du den Fehler gefunden, wirst du feststellen, dass du stolz auf dich sein wirst, und vor allem hast du viel neue Erfahrung gesammelt. Der Fehler wird dir hoffentlich nie wieder unterlaufen oder du wirst ihn schnell beheben.



Kotarō ist ein in Japan entwickelter humanoider (menschlich aussehender) Roboter. Er ist 1,33 m hoch und wiegt 20 kg.

Bild: Manfred Werner, Wikimedia Commons, CC BY-SA 3.0

SO ARBEITEST DU MIT DIESEM BUCH

Im Buch benutzen wir verschiedene Kästen, um dich auf Besonderheiten hinzuweisen:



Hinweise sehen so aus. Dabei handelt es sich um zusätzliche Informationen, die dir einen tieferen Einblick in ein Thema geben. Manchmal könnte es auch »Besserwisser-Wissen« sein: Mit dem, was im Kasten steht, weißt du dann mehr als andere.



Tipps sind dazu da, dich auf etwas Kniffliges hinzuweisen, oder ich habe einen echten Profitrick für dich. Es kann auch sein, dass ich aus meiner Erfahrung heraus weiß, dass etwas oft falsch gemacht wird und ich dich davor bewahren will.



So sieht es aus, wenn etwas ganz besonders wichtig ist und unbedingt beachtet werden muss. Andernfalls wird etwa beschädigt oder es wird gefährlich (dazu lassen wir es aber nicht kommen).

WAS DU FÜR DIESES BUCH SCHON KÖNNEN SOLLTEST

Damit wir uns im Buch auf das Thema Roboter, Arduino und Programmierung konzentrieren können, sind ein paar Grundlagen wichtig, die wir voraussetzen:

- ◇ Du hast einen Computer mit Microsoft Windows 10.
- ◇ Du kennst dich mit Windows und dem Dateisystem aus. Ordner anlegen und im (Datei-)Explorer Dateien verschieben sowie der Umgang mit Komprimierungsprogrammen wie Zip oder Rar beherrscht du.
- ◇ Du kennst dich im Internet (im Besonderen im WWW, dem World Wide Web) aus und kannst einen Webbrowser wie Firefox, Chrome oder Edge bedienen.
- ◇ Wenn du ein paar Elektronik-Grundkenntnisse besitzt, ist das hilfreich, aber nicht zwingend erforderlich.

Internetadressen (URLs) sind oft lang, bestehen aus merkwürdigen Zeichen und sind schwer fehlerfrei im Browser einzugeben. Aus diesem Grund steht nach der Angabe der tatsächlichen Adresse in Klammern immer noch eine Kurz-URL. Beispielsweise: <https://de.wikipedia.org/wiki/Wikipedia:Hauptseite> (*t1p.de/qwje*). Es reicht, wenn du die Kurz-URL im Browser eingibst – du wirst dann zu der gewünschten Webseite mit der langen Adresse geleitet.

Uns (und deinen Robotern) ist es egal, ob du ein Mädchen/eine Frau oder ein Junge/ein Mann bist oder dich als etwas ganz anderes siehst. Wichtig ist nur, dass dir dieses Buch und das Thema Spaß machen. Seit einigen Jahren gibt es eine Diskussion darüber, ob man in Texten geschlechtsneutral schreiben muss. Also beispielsweise nicht mehr »du bist ein Programmierer«, sondern »du bist ein Programmierer oder eine ProgrammiererIn«, manchmal wird auch »du bist ein/e Programmierer*In« oder Ähnliches benutzt. Wir wollen diese Diskussion gar nicht aufgreifen oder bewerten, finden aber, dass es einen Text schwerer zu lesen macht, weil es sperrig und gekünstelt wirkt. Aus diesem Grund verzichten wir auf diesen Schreibstil, der »gendern« (Geschlechtsidentität des Menschen) genannt wird.



DATEIEN ZUM DOWNLOAD

Zu diesem Buch gibt es ein Download, in dem ergänzende Dateien zu finden sind und vor allem die Beispielprogramme. Die Programme wurden mit NEPO erstellt

und aus dem Open Roberta Lab heraus als XML-Datei gespeichert, sodass sie dort auch wieder importiert werden können. Bei den jeweiligen Aufgaben wird der Dateiname mit einem Symbol gekennzeichnet:

 beispiel-datei.xml

FRAGEN UND AUFGABEN AM ENDE DER KAPITEL

Am Ende eines Kapitels findest du ein paar Fragen und ein paar Aufgaben. Die Fragen helfen dir, zu prüfen, ob du wichtige Inhalte verstanden hast. Im Anhang C am Ende des Buchs stehen die Antworten – manchmal etwas ausführlicher, um dir zu helfen. Die Aufgaben bauen auf dem auf, was du im jeweiligen Kapitel ausprobiert hast, und geben dir Anregungen, weiter zu experimentieren und dein Wissen für neue Ideen einzusetzen. Auch hierzu findest du im Anhang Hinweise, wie die Aufgabe zu lösen sein kann. Du kannst aber auch einen anderen Weg gegangen sein – Hauptsache, du hattest Erfolg.

WAS DICH IN DIESEM BUCH ERWARTET

Zuerst einmal soll dich hier vor allem Spaß erwarten und deine Entdeckerfreude gestillt werden. Der Bau von Robotern teilt sich in zwei Aufgabengebiete:

- ◇ Aufbau der Elektronik und Mechanik
- ◇ Programmierung des Mikrocontrollers

Jeder Roboter besteht aus verschiedenen elektronischen und mechanischen Bauteilen, die seine Fähigkeiten ausmachen, aber auch seine Beschränkungen.

WAS DU FÜR DEINE ROBOTER BENÖTIGST

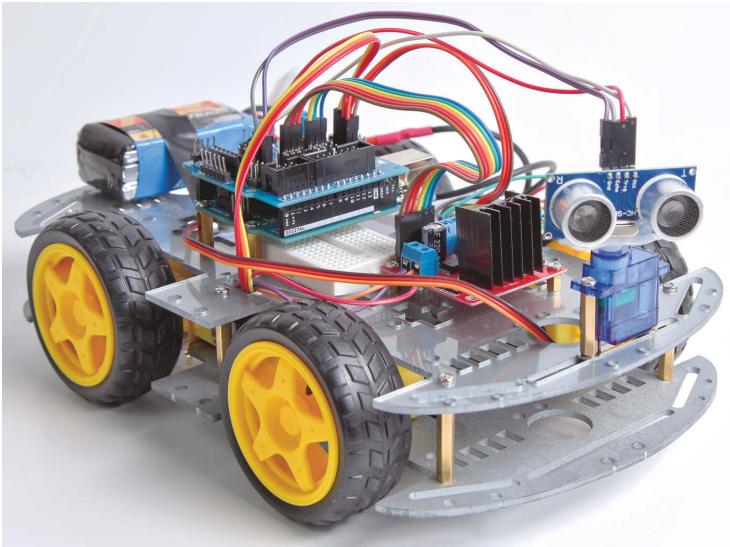
Im Anhang A findest du eine umfangreiche Bauteil- und Materialliste mit den in diesem Buch benötigten Teilen. Du kannst dir gleich auf einmal alle Bauteile kaufen und dann in Ruhe jedes Modell nachbauen. Oder du schaust dir erst einmal die Bauvorschläge an und besorgst dir dann nur das notwendige Material für einzelne Projekte.



Das Herz aller Roboter wird ein Arduino Uno sein, der diese mit verschiedenen Programmen so steuert, wie du es willst. Ab Seite 54 erfährst du, was ein Arduino ist, falls du noch gar keine Vorstellung hast

DAS WIRST DU ALLES SELBER BAUEN

Der aufwendigste Roboter wird ein Fahrzeug sein, das mit vier Rädern angetrieben wird und verschiedene Aufgaben bewerkstelligen kann, wie sie auch die großen Brüder aus der Industrie lösen. Dazu gehören verschiedene Fahrmanöver, das Erkennen von Hindernissen, Verfolgen von Linien und auch die selbstständige Wegfindung.



Ein möglicher Ausbau deines Roboterfahrzeugs

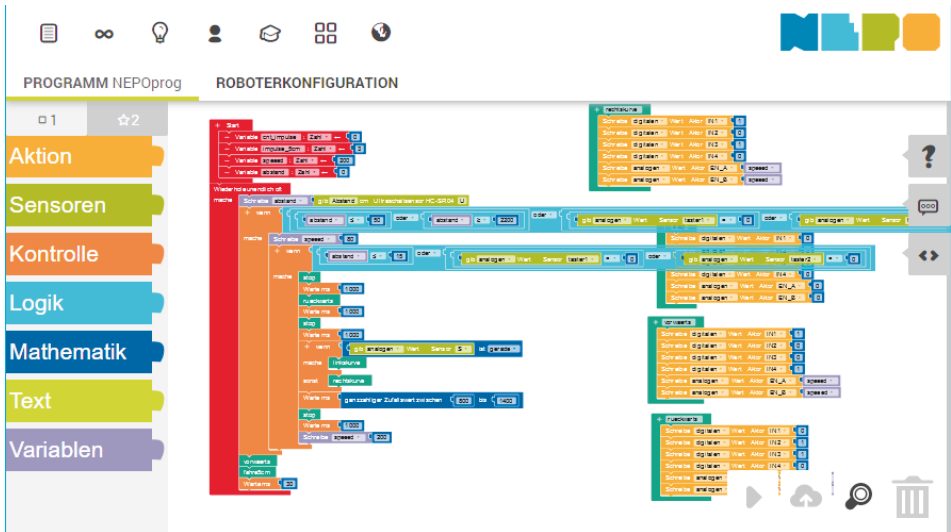
Oft wirst du Angebote finden, bei denen ähnliche Roboterfahrzeuge aufgebaut werden, die dann mit einem Empfänger ausgestattet werden, damit du ihn mit einem Smartphone oder einer kleinen Fernbedienung steuern kannst. Das sind dann allerdings keine Roboter, sondern einfach nur ferngesteuerte Modellautos.



Wir bauen aber auch kleinere Roboter, die selbstständig und unabhängig vom Fahrzeug sind. Los geht es gleich im nächsten Kapitel mit einem witzigen kleinen Bürsten-Roboter. Für dich mag es leicht sein, auf zwei Beinen zu laufen, aber für Roboter ist es extrem schwer – also testen wir zwei verschiedene Lösungen und probieren aus, ob zwei oder sechs Beine besser geeignet sind. Ein echter Industrieroboter-Arm darf natürlich nicht fehlen, damit du Objekte greifen und bewegen kannst – vielleicht eine gute Ergänzung für deine Bauten aus Legosteinen oder Ähnlichem.

Neben der greifbaren Technik ist das zweite Standbein die Programmierung eines Mikrocontrollers, der die Elektronik kontrolliert: Sensoren erfassen die Umgebung und melden dies an eine Art kleinen Computer, der daraufhin Entscheidungen trifft und die Aktoren (beispielsweise Motoren) ansteuert.

Wir bedienen uns der Roboter, um zu verstehen, wie die Elektronik funktioniert und wozu sie genutzt werden kann. Der Mikrocontroller auf einem Arduino Uno wird von dir selbst programmiert. Das Open Roberta Lab bietet dazu eine kostenlose Entwicklungsumgebung für die grafische Programmiersprache NEPO, in der du Befehle in Form farbiger Blöcke zusammenstellst.



Im Open Roberta Lab nutzt du NEPO, um den Arduino zu programmieren.

Schrittweise wirst du erleben, wie du die einfach zu verstehenden Befehlsblöcke so kombinierst, dass deine Roboter genau das machen, was du willst. Am Ende wirst du dann nicht nur verschiedene Roboter gebaut haben, sondern nebenbei auch viel über Programmierung gelernt haben, ohne dass es langweilig wurde. So bist du gut gerüstet, um eigene Ideen umzusetzen oder Projekte anderer zu verstehen, nachzubauen oder weiterzuentwickeln.

Im letzten Kapitel zeige ich dir, was es darüber hinaus noch zu entdecken gibt: Mit Scratch kannst du ähnlich wie mit NEPO programmieren und deine eigenen Spiele entwickeln – und das, ohne viel Neues lernen zu müssen. Eine andere Webseite bietet dir die Möglichkeit, elektronische Schaltungen in einer virtuellen Umgebung aufzubauen und zu simulieren. Willst du noch einen Schritt weitergehen, dann kannst du dort sogar einen Arduino in C/C++ programmieren und ausprobieren – ganz ohne selber einen besitzen zu müssen. Auch andere Roboter lassen sich auf den entsprechenden Webseiten in einer simulierten Welt ausprobieren und programmieren. Solange du beispielsweise auf die bestellten Teile für deine eigenen Roboter wartest, kannst du dir da prima die Zeit vertreiben.

Ein Blick in den Anhang B lohnt sich ebenfalls, denn dort zeige ich dir verschiedene Arbeitstechniken, wie du Verbindungen mit Kabeln herstellen kannst und wie mit einer Crimpzange Kontakte für die im Buch verwendeten Jumper-Kabel gefertigt werden können.

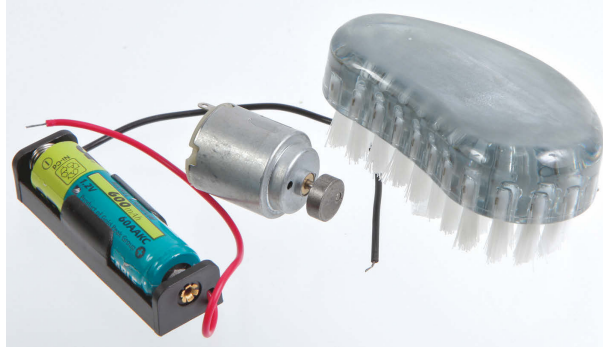


1 *SCHRUBBI, DER BÜRSTEN-KÄFER*

In diesem Abschnitt lernst du:

- ⊙ einen einfachen Roboter zu bauen
- ⊙ was ein Roboter überhaupt ist und welche Vorbilder es in der Pop-Kultur gibt

Kleine wuselige Käfer aus Zahn-, Hand- oder Spülbürsten lassen sich mit wenigen Handgriffen bauen und sorgen immer wieder für Begeisterung. Du benötigst nur einen kleinen Vibrationsmotor, einen Batteriehalter und eine (ausgediente) Bürste. Eine große Auswahl an Vibrationsmotoren gibt es in Internetauktionshäusern – es spielt keine Rolle, was für einen du nimmst. Du musst nur einen ungefähr zur Betriebsspannung passenden Batteriehalter und Batterien dazu wählen.



Batterie mit -halter, Vibrationsmotor und Handbürste – mehr braucht es nicht.



Ein Vibrationsmotor wird in jedem Smartphone für den Vibrationsalarm benutzt. Es ist ein ganz gewöhnlicher Motor, auf den eine Unwucht montiert ist. Das ist ein kleines Gewicht, das nicht mittig auf der Achse sitzt und dafür sorgt, dass der Motor wackelt, sobald er sich dreht. Du kannst auch selber aus einem Motor einen Vibrationsmotor basteln, indem du einfach einen Flaschenkorken auf die Achse steckst.

1. Verbinde die Anschlüsse des Batteriehalters mit dem Motor. Die Polung (Plus und Minus) ist nicht wichtig. Mehr zu Möglichkeiten, eine Verbindung herzustellen, findet du im Anhang ab Seite 273.
2. Lege probeweise die Batterie ein und teste, ob sich der Motor dreht.
3. Klebe mit Heiß- oder Kraftkleber die Batterie und den Motor auf deine Bürste. Der Bürsten-Käfer ist nun eigentlich fertig. Aber erst der nächste Schritt macht aus ihm etwas Besonderes.



- Schmücke den Käfer mit lustigem Dekokram wie Klebeaugen, Pfeifenreinigern, Glitzerstaub, Aufkleber und was du sonst noch so vorrätig hast.

Falls du etwas Inspiration beim Dekorieren brauchst, schau dir doch einfach die Ergebnisse einer Bildersuche im Web nach dem Stichwort »bristlebot« (englisch: Bürsten-Roboter) an.



Schrubbi ist mein ganz persönlicher Käfer.

Setzt du deinen Bürsten-Bot auf den Boden, wird er sich durch die Vibration bewegen und umherwandern – Haustiere werden ihn ebenso lieben.

WAS IST ÜBERHAUPT EIN ROBOTER?

Du hast jetzt schon einen Roboter bauen können und es wird Zeit, sich zu fragen, was überhaupt ein Roboter ist. Wenn wir es nämlich genau nehmen, dann war das bisher gar kein Roboter – Spaß macht er aber dennoch.

Ein Roboter ist eine technische Apparatur, die üblicherweise dazu dient, dem Menschen häufig wiederkehrende mechanische Arbeit abzunehmen.



Vor sich hinzuwackeln oder immer im Kreis einer Linie zu folgen, ist eigentlich keine Arbeit, die ein Mensch verrichtet und die ihm eine Apparatur abnehmen muss. **Spielzeugroboter** besitzen einen gewissen automatisierten Bewegungsablauf: Sie wiederholen eine Handlung ständig oder können sogar auf bestimmte Ereignisse reagieren, aber ohne dass ihr Funktionsumfang einen arbeits- oder forschungstechnischen Nutzen hat.



Fußballspielende Spielzeugroboter beim RoboCup

Bild: Peter Schulz, Wikimedia Commons, CC BY-SA 4.0



Als *Bionik* bezeichnet man die Übertragung von Phänomenen der Natur auf die Technik. Dabei schaut man, wie die Natur ein Problem gelöst hat, und baut das Prinzip für vom Menschen hergestellte Produkte nach. Woher Klettverschlüsse stammen, wirst du dir sicher denken können, wenn du schon mal durch eine Wiese gekrabbelt bist. Durch den Lotuseffekt perlt Schmutz an einer Oberfläche ab: Toilettenschüssel und Hauswände bleiben sauber dank der speziellen Oberfläche, die man sich von der Lotuspflanze abgeschaut hat. Die Gelenke in Roboterarmen sind nichts anderes als Kopien der Gelenke an deinem Arm. Robotik und Bionik haben viele Berührungspunkte.

Wir sehen das hier nicht so eng und bezeichnen einfach alles als Roboter, was sich bewegt und ein Bewegungsmuster wiederholt ausführen kann. Bestimmt hast du schon viele verschiedene Roboter gesehen, die dich fasziniert haben – hier ein kleiner Überblick:

Industrieroboter fallen oft auf, weil sie groß sind und meistens nur aus einem Arm bestehen, der immer wieder die gleiche Aufgabe wiederholt, wie beispielsweise ein Werkstück bewegen, schweißen, lackieren oder fräsen. Sie sind an einem Ort fest aufgebaut und können sich nicht fortbewegen. Ihr Vorteil ist, dass sie die immer gleichen Abläufe fehlerfrei und schnell wiederholen können, ohne eine Pause

machen zu müssen. Sie haben viele Jobs, die früher von Menschen ausgeführt wurden, übernommen.

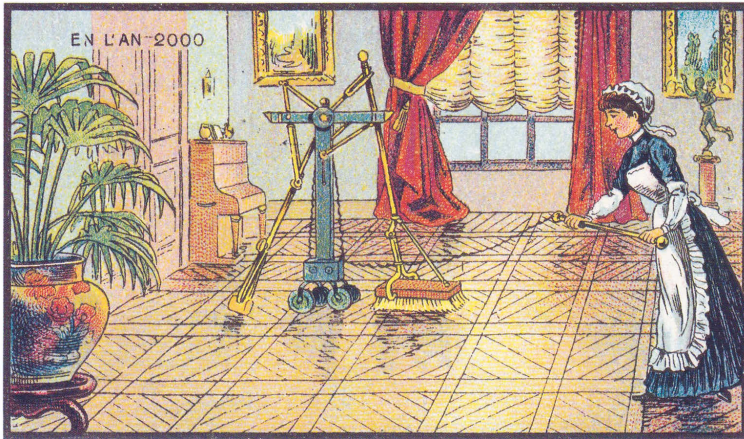


Industrieroboter schweißen eine Autokarosserie zusammen.

Bild: Zen wave, Wikimedia Commons, CC BY-SA 4.0

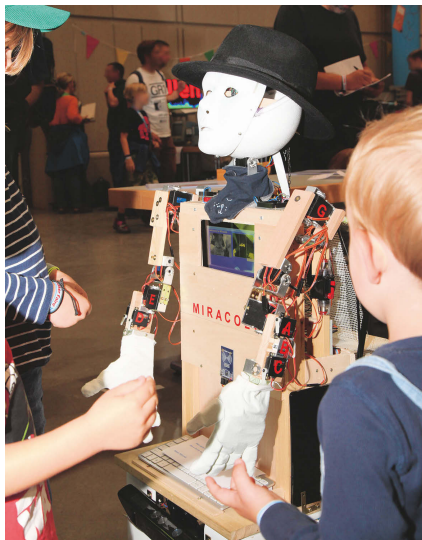
Damit der Industrieroboter weiß, wie er sich bewegen muss, kann ein Programmierer die Abläufe mit Befehlen vorgeben oder sie dem Roboter zeigen: Beim Playback-Verfahren entspannt der Roboter gewissermaßen seine Muskeln (Motoren) und kann von Hand bewegt werden. Ein erfahrener Arbeiter bewegt den Roboter dann genau so, wie er sich später automatisch bewegen soll. Diese Bewegungen speichert der Roboter ab und kann sie später wiederholen.

Das Teach-In-Verfahren ist ähnlich: Hier wird der Roboter mit einer Steuerkonsole (oder auch einem Datenhandschuh) an eine Position gefahren und der Punkt wird gespeichert. Anschließend wird dem Roboter der nächste Punkt gezeigt und so weiter. Später berechnet der Roboter dann selber, wie er sich von einem Punkt zum nächsten bewegen kann.



1899 hat man sich vorgestellt, dass so im Jahr 2000 der Hausputz mit einem Roboter aussieht.

Die gut 100 Jahre alte Idee, wie in der Zukunft Roboter den Boden putzen, mag etwas komisch aussehen – völlig falsch ist sie nicht. **Serviceroboter** übernehmen immer mehr Aufgaben: Anfangs dachten viele Entwickler, der Gipfel des nützlichen Roboterdaseins sei, Cocktails auf Partys zu servieren und den Geschirrspüler (übrigens im Grunde auch schon ein Roboter) auszuräumen. Inzwischen hat man eingesehen, dass das nicht sinnvoll ist, und beschränkt sich auf einfache Aufgaben wie Staubsaugen oder Rasenmähen. Die kleinen Roboter wimmeln inzwischen durch viele Wohnungen und Gärten und suchen sich selbstständig ihren Weg – wie das genau funktioniert, erfährst du später, wenn wir selber einen solchen Roboter bauen.



Einen humanoiden Roboter kannst auch du aus Holz und den Teilen bauen, die du kennenlernen wirst.

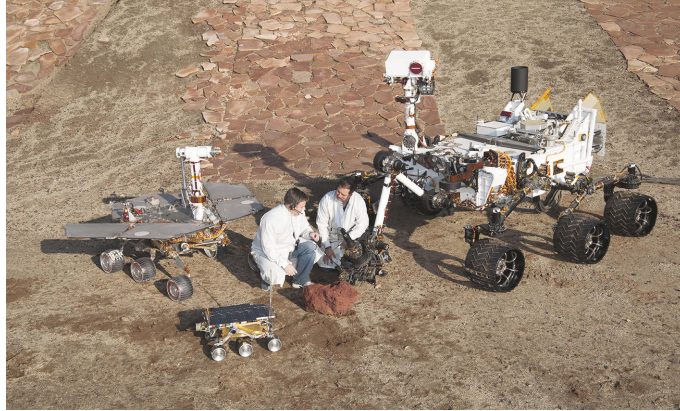
Ein besonderer Reiz geht von **humanoiden Robotern** aus, die der menschlichen Gestalt nachempfunden sind und einen Kopf und Hände, vielleicht sogar Beine haben. Durch die menschliche Form soll der Blechkamerad freundlicher wirken und Ängste vor ihm abgebaut werden. Automaten, die wie Menschen aussehen, sind schon vor vielen Hundert Jahren gebaut worden, als es noch gar keine Elektronik gab. Oft steuerte aber ein Mensch die Apparatur – entweder direkt durch seine Hände oder über eine kabelgebundene Fernbedienung. Eine besondere Herausforderung für Entwickler ist es, einen Roboter auf zwei Beinen laufen zu lassen. Bis heute kann das kein Roboter so gut wie ein Mensch.



Ein Feuerwehrmann steuert einen Löschroboter zur Brandbekämpfung.

Bild: Superbass, Wikimedia Commons, CC BY-SA 3.0

Ein recht neues Betätigungsfeld für Roboter sind **Aufklärungs- und Rettungsmissionen** (SAR: Search and Rescue, englisch: suchen und retten). Bei Katastrophen oder in bedrohlichen Situationen kann es für Menschen zu gefährlich sein – dann ist es gut, wenn ein Roboter die Aufgabe übernehmen kann, dessen Beschädigung oder Verlust vielleicht teuer ist, aber so wird kein Menschenleben riskiert. Das Militär hat an diesen Robotern ebenfalls ein riesiges Interesse, da man so bequem »von zu Hause aus« weit entfernte Kriege führen kann. Drohnen (selbstfliegende Fluggeräte) dienen beispielsweise der Aufklärung: Sie starten automatisch, fliegen ohne Steuerung eines Menschen über einem Gebiet und landen dann wieder. Viele Rettungsroboter sind allerdings im Grunde nur aufgemotzte ferngesteuerte Fahrzeuge und arbeiten nicht selbstständig. Sobald ein Erkundungsroboter auf einem anderen Himmelskörper unterwegs ist, spricht man von einem *Rover*.



Die drei Rover (baugleiche Kopien), die im Rahmen des Pathfinder-Projekts den Mars erkunden, auf einem irdischen Testgelände. Vorne: Sojourner (seit 1997 auf dem Mars), links: Spirit/Opportunity (2004) und rechts: Curiosity (2012).

Bild: NASA/JPL-Caltech

Bei Robotern denken wir fast immer sofort an größere Maschinen – es gibt aber auch Forscher, die sich um die extreme Verkleinerung von Robotern bemühen und Roboter in molekularer Größe (so klein wie eine Blutzelle, die du nur mit einem sehr guten Mikroskop sehen kannst) erschaffen. Diese winzig kleinen **Nanobots** sollen dann im Körper Krankheiten bekämpfen. Bisher gibt es noch keine so kleinen Roboter, die wirklich Aufgaben übernehmen können. Aber das wird sicher nicht mehr lange dauern und auf der Webseite <https://drive.google.com/file/d/1TLFyleGzs66nl-R9tfSM5iMn1MfWS3WHP/t1p.de/k1c1> kannst du ein Video von einem einfachen krabbelnden Roboter sehen, der nicht größer ist, als ein Haar dick ist, und durch Sonnenlicht über eine winzige Solarzelle angetrieben wird.

ROBOTERGESETZE



Isaac Asimov 1965

Isaac Asimov war einer der bekanntesten russisch-amerikanischen Science-Fiction-Schriftsteller im 20. Jahrhundert und auch wenn dir der Name vielleicht nichts sagt, hast du bestimmt schon einen Film gesehen, der Ideen aus einem seiner Bücher aufgreift. In vielen Filmen werden vor allem immer wieder die drei Asimov'schen Robotergesetze genannt:

1. Ein Roboter darf kein menschliches Wesen verletzen oder durch Untätigkeit zulassen, dass einem menschlichen Wesen Schaden zugefügt wird.
2. Ein Roboter muss den ihm von einem Menschen gegebenen Befehlen gehorchen – es sei denn, ein solcher Befehl würde mit Regel eins kollidieren.

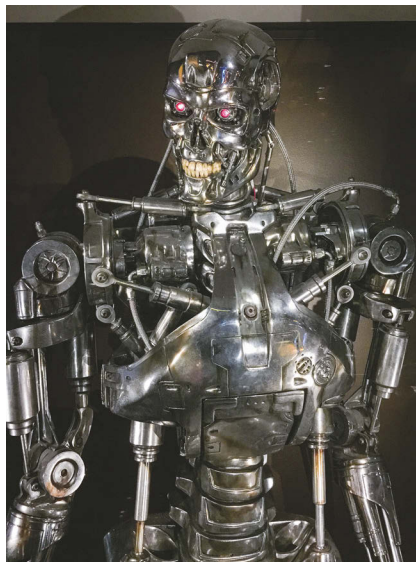
- Ein Roboter muss seine Existenz beschützen, solange dieser Schutz nicht mit Regel eins oder zwei kollidiert.

Diese Gesetze gelten nicht wirklich für Roboter (Kampfroboter sind beispielsweise nur dazu gebaut worden, um Menschen zu schaden), sie enthalten sogar Schlupflöcher und wurden im Rahmen der Pop-Kultur bereits erweitert, zeigen aber, wie sich ein Roboter verhalten sollte. Letztendlich ist immer der Programmierer und Erbauer (also auch du) dafür verantwortlich, wie ein Roboter handelt.

ROBOTER IM FILM

Am häufigsten werden dir wohl bisher Roboter in Filmen begegnet sein. Mal sehen, was wir von diesen lernen können und inwieweit sie uns für eigene Entwürfe inspirieren oder unser Bild von Robotern geprägt haben.

Die erste Ernüchterung ist, dass es sich dabei meistens um Tricktechnik handelt: Die Roboter können sich oft gar nicht wirklich bewegen, nichts selbstständig erledigen oder gar denken und reden.



Der T-800 ist ein **Androide**: eine Maschine umhüllt von lebendem menschlichen Gewebe (gespielt von Arnold Schwarzenegger).

Bild: Dale Cruse, Wikimedia Commons, CC BY 2.0

Oft werden die Asimov'schen Robotergesetze thematisiert, weil sich Roboter gegen Menschen wenden, obwohl sie es gar nicht dürften, oder sie Hilfeleistungen unterlassen und so gegen ihre inneren Protokolle verstoßen. Deine Eltern erinnern sich sicher noch an den ersten »Terminator«-Film, in dem der T-800 aus der Zukunft zurückgeschickt wird und sehr brutal gegenüber Menschen vorgeht.



Die meisten Roboter im Film oder in Science-Fiction-Romanen besitzen eine künstliche Intelligenz (KI, englisch: artificial intelligence, AI). Die Programmierer versuchen dabei, Entscheidungsstrukturen des Menschen nachzubilden, sodass sich der Roboter menschlich verhält. Dafür sind enorm viel Rechenleistung und eine große Datenbank mit Beispielen notwendig, aus der das Computerprogramm lernen kann. Du nutzt vermutlich bereits Computersysteme, die eine sehr schwache künstliche Intelligenz besitzen und dir vorgaukeln, du redest mit einem Menschen: Siri, Amazon Echo oder Google Now sind Sprachassistenten, die sich bemühen, deine Sprache zu verstehen und eine Antwort zu kennen. Um zu testen, ob eine Maschine eine dem Menschen gleichwertige Intelligenz simuliert, gibt es den Turing-Test, den bisher keine Maschine bestanden hat. Du bist also intelligenter als jeder Computer dieser Welt!



Fernsteuerbare Nachbauten aus Holz und Kunststoff der Roboter R2-D2 (vorne) und BB-8

trächlich neben die Schauspieler in den Film eingefügt wird, auch wenn das Bewegungsprinzip mit der sich endlos drehenden Kugel als Körper und dem sich stets oben befindenden Kopf technisch umsetzbar ist. In reinen Trickfilmen ist es natürlich noch viel leichter, fantasievolle Roboter zu zeigen, wie beispielsweise WALL-E und Eve oder Bender (»Futurama«).

Bis auf wenige Ausnahmen sehen Roboter in Filmen immer faszinierend aus (oder wie Topmodels) und beherrschen die tollsten Techniken. Wie die »Transformers« können sich die Roboter verändern, werden größer oder kleiner und bewegen sich

Ein einfacher Trick, um Roboter zum »Leben« zu erwecken, ist es, einen Menschen in ein Roboter-Kostüm zu stecken. R2-D2 und C-3PO aus »Star Wars« kennst du bestimmt – sie sind bekannte Beispiele für Roboter, in denen Schauspieler als sogenannte Puppenspieler steckten. Und auch im 2019er Horror-Film »I Am Mother« steckt ein Mensch im Roboter-Anzug.

Wenn kein Platz im Roboter für einen (kleinen) Menschen ist, dann wird oft zu ferngesteuerten Modellen gegriffen, die mit Seilzügen oder wie ein funkgesteuertes Modellauto bewegt werden. Sehr oft gibt es auch nur einen bestimmten Körperteil, der wirklich bewegt werden kann – beispielsweise der Kopf oder eine Hand, der dann in Nahaufnahme gezeigt wird, sodass man im Film gar nicht sehen kann, dass der restliche Körper fehlt.

Heute nutzt man computeranimierte Figuren: Der quirliche Roboter BB-8 (ebenfalls aus »Star Wars«) ist in fast allen Szenen eine Animation, die nach-

flüssig durch ihre Umwelt. Sie erkennen alles um sich herum, reagieren auf Sprache und reparieren sich notfalls sogar selbst.

In der Realität ist nichts davon möglich. Die meisten Roboter sehen klobig aus, sind strohdumm und können nur das wenige, das ihnen von jemandem vorher mühsam eingetrichtert wurde. Behalte das im Hinterkopf, wenn du deinen eigenen Roboter baust: Auch wenn er nicht elegant ist oder die Welt beherrschen kann, so ist es ein ganz besonderer Roboter – deiner! Und je mehr Zeit du in seine Weiterentwicklung steckst, desto besser wird er. Oder du baust einen weiteren, bei dem du das zuvor Ausprobierte nutzt und verbesserst.



Nachbauten der Roboter Eve und WALL-E (rechts) in einem Freizeitpark

Bild: Helfmann, Wikipedia (DE), CC-BY-SA 3.0/de

NOCH MEHR INPUT

Während du deine Roboter baust oder auch wenn du am Ende des Buchs angelangt bist, juckt es dich vielleicht in den Fingern und du willst mehr wissen oder dich mit anderen Gleichgesinnten treffen. Im Internet findest du sehr viele Anlaufstellen. Wenn du eine Frage hast oder ein Praxisbeispiel brauchst, sind Suchmaschinen natürlich die erste Wahl. Gib deine Frage ein und du wirst sicher viele Antworten finden. Vor allem Videos sind eine gute Ergänzung. Es gibt aber auch ein paar Webseiten mit Diskussionsforen, in denen du Fragen stellen kannst und die Leute sich freuen, wenn sie dir helfen können. Dazu gehören diese Seiten:

- ◇ <https://www.mikrocontroller.net> ([t1p.de/820g](https://www.mikrocontroller.net))
Ein Forum für jede Form von Mikrocontroller, aber auch allgemein rund um Elektronik.
- ◇ <https://forum.arduino.cc> ([t1p.de/tbfl](https://forum.arduino.cc))
Das Forum der Arduino-Entwickler. Viele Themengebiete, in denen Englisch benutzt wird, aber auch ein deutschsprachiger Teil.
- ◇ <https://www.elektronik-kompodium.de> ([t1p.de/xqw5](https://www.elektronik-kompodium.de))
Einführung, Fachartikel und ein Forum zum Thema Elektronik.

Eine wirklich beeindruckende Möglichkeit, gleichgesinnte Bastler und Entwickler zu treffen, sind die weltweit stattfindenden Treffen. Zu den bekanntesten gehören die Maker-Faires (<https://maker-faire.de>, [t1p.de/npur](https://maker-faire.de)) oder die Veranstaltungen, die »make« gefolgt von einem Stadtnamen (meistens in Englisch) heißen – beispielsweise »make munich« in München. Auch du kannst dort deine Projekte dem interessierten Publikum vorführen, um Gleichgesinnte zu treffen oder neue Ideen zu erhalten – meistens bekommst du sogar kostenlos einen Ausstellerplatz.

In vielen Städten gibt es mittlerweile sogenannte Makerspaces, FabLabs, Hackerspaces oder offene Werkstätten, in denen sich Menschen jeden Alters mit den unterschiedlichsten Interessen treffen und austauschen. Dort findest du nicht nur Hilfe bei deinem Projekt, sondern du kannst bei anderen mithelfen oder die dort vorhandenen Geräte wie 3D-Drucker oder Lasercutter nutzen.

EIN PAAR FRAGEN ...

1. Gibt es die Robotergesetze wirklich und für wen gelten sie?
2. Wenn du Drähte und Bauteile miteinander verbinden sollst: Wo findest du dazu Anleitungen?
3. Was ist ein Rover?
4. Handelt es sich bei dem Bürsten-Roboter um einen echten Roboter?

... UND EIN PAAR AUFGABEN

1. Besorge dir die Teile aus den Einkaufslisten, die du am Ende des Buchs findest.
2. Frage deine Eltern, ob sie an ihrem Arbeitsplatz mit Robotern zu tun haben und wie sich dadurch ihre Arbeit verändert hat.

STICHWORTVERZEICHNIS

.ino 82
.xml 80
// 259
133
~ 67
3,7 V 272
3D 250

A

Abisolieren 273
Abisolierzange 274
Ablaufsteuerung 87
ABS 138
Abstand
 messen 166
Acrylglas 119
AGV 157
AI 24
Akku 271
Aktor 84
Alkali-Mangan-Batterie 132
Alkaline 132
Ampel 72
Analoge Eingänge 66
Android 23
Anfänger 75
Anker 100
Anmeldung 77
Anode 31
Anschlüsse (COM & LPT) 34
Anschlussübersicht 64
Antiblockiersystem 138
App Inventor 68
Arbeitsbereich 75
Arbeitstechnik 273
Arduino 27, 34, 42, 54, 58, 64, 111
 Clone/Klon 59
 Fälschung 59

Logo 60
Mega 2560 59
Mini Pro 59
Original 59
Arduino Sensor Shield 111
Arduino-IDE 70, 81
Ariadne 215
Arithmetischer Mittelwert 222
ARM Cortex 64
Artificial intelligence 24
ASIMO 199
Asimov, Isaac 22
Asimov'sche Robotergesetze 22
ATmega16U2 35, 56
ATmega328P 55
Atmel 55
Aufgabe 12
Aufklärung 21
Ausgang 50, 84
Automated Guided Vehicle 157
Autonomes Fahrzeug 148
AVR 55

B

Balancing 272
Baseball
 Scratch 239
Basic 68
Batterie 271
Batteriebelastung 131
Batteriehalter 114
BB-8 24
BBC micro:Bit 61
Bender 24
Benutzerkonto 77
Bildererkennung 56, 63
Bionik 18
BoB the BiPed 199
Bohren 119
Bremslicht 150

Brick 245
 Bristlebot 17
 Browser 41, 235
 Buchsenleiste 58
 Bug 83
 Bühne
 Scratch 237
 Bürste 15
 Bürstenmotor 100
 Buzzer 84, 148
 BYOB 68

C

C/C++ 68, 81, 249, 260
 C-3PO 24
 Calliope 62
 CH340 35
 Chassis 106
 Circuits 250
 Clone 59
 COM 34
 Compiler 69
 Continuous rotation 187
 Creative-Commons-Lizenz 59
 Crimpkontakt 276
 Crimpzange 277
 Curiosity 22

D

DC 100
 DC-Motor 100
 Debugging 83
 Delfin 169
 Dezimaltrennzeichen 31
 Dezimalzahl 31
 einbauen 92
 Didaktik 32
 DIL-Gehäuse 55
 Direct current 100
 Download 11
 Draht 274
 Drehmoment 104
 Drehrichtung 102, 119
 Dual in-line 56
 Dupont-Kabel 276

Durchmesser 142
 Dynamo 105

E

Echtzeitfähigkeit 63
 Editor 70
 Eigenschaft
 Scratch 239
 Ein-Chip-Computersystem 54
 Eingang 50
 Einplatinencomputer 61
 Elektromotor 100, 179
 Elektronik 249
 Enable 123
 Entfernung
 messen 158, 167, 169
 Entfernungsmessung 86
 Entwicklerboard 58
 Entwicklungsboard 61, 63
 Entwicklungsumgebung 68, 69, 235
 Ereignisorientiert 237
 ESP32 64
 ESP8622 63
 EV3 244
 Eve 24
 Experte 75
 Export 80

F

FabLab 26
 Fahrerloses Transportfahrzeug 157
 Fahrgestell 106
 Fahrzeug
 autonomes 148
 Fälschung 59
 Farbsensor 246
 Fehlersuche 9, 50
 Figur
 Scratch 237
 Firmware 68, 69
 First LEGO League 244
 Flachbandkabel 125
 Flattern 185
 Fledermaus 169
 Forum 26

Fotowiderstand 164
Frage 12
Frequenz
 Ton 150
Fritzing 252
FT232 35
FTF 157
Funktion 95
 entfalten 96
 zusammenfalten 96
Funktionsaufruf 224, 226
Fußball 18

G

Gabellichtschränke 140
Gangschaltung 104
Genderneutral 11
Genuino 58
Geräte-Manager 33
Geschwindigkeitssensor 138
Getriebe 186
Getriebemotor 101
Gleichstrom 100
GND 30, 31, 66
Greifer
 Roboterarm 195
Grip 116
Groß-/Kleinschreibung 85, 260
Ground 31

H

Hackerspace 26
Halbleiterchip 54
Hall, Edwin 157
Hall-Sensor 157
Handyladegerät 29
Hardware 67
Hash-Symbol 133
Hausputz 20
Haustier 169
H-Brücke 117
HC-020K 138
HC-SR04 87, 169
Hexapoden-Bot 209
High-Pegel 51
High-Signal 67

Hilfe
 zu Open Roberta Lab 76
Hindernis
 erkennen 152
Hohlbuchse 64, 122
Honda 199
HSS-Bohrer 119
Humanoid 21
Hupe
 einbauen 148

I

I/O 67
I/O-Pin 50
IC 117
IC-Sockel 55
IDE 68
Import 80
Industrieroboter 18
Industrieroboter-Arm 189
Infrarot
 Entfernung messen 169
 Gabellichtschränke 140
Infrarotlicht 158
INO 82
Input/Output 50
Integrated Development Environment 68
Integrierte Entwicklungsumgebung 68
Internetadresse 11, 79
INx 126
Irrgarten 215
IR-Sensor 158

J

JavaScript 68
Jumper 125
Jumperkabel 113, 276
Jumpersteckplatz 113

K

Kabel
 verbinden 273
Katapult 179
Kathode 31

Katze
 Scratch 239
 KI 24
 Klemmblock 113
 Klon 59
 Kohlebürste 100
 Kommunikationsanschluss 34
 Kompilieren 69
 Koppelnavigation 157
 Kotarō 10
 Kraft 104
 Krankheiten 22
 Kreiszahl 141
 Krokodilklemme 62
 Künstliche Intelligenz 24
 Kurve
 fahren 135
 Kurz-URL 11
 KY-006 148
 KY-033 158

L

L298N 117
 Labyrinth 214
 Ladegerät 272
 Laufroboter 199
 LCD 261
 LDR 164
 LED 28, 39, 105
 Anode 31
 Kathode 31
 L 30, 67
 Matrix 62
 On 30, 67
 Polung 30
 RX 49, 67
 TX 49, 67
 Legierung 281
 Lego Mindstorms 244
 Lego RCX-Code 68
 LEGO-Technik-System 244
 Leitlinie 157
 Leuchtdiode 28
 Lichtschranke 140
 Light Dependent Resistor 164
 Linienfolger 163
 Link 79

Linux 63
 LiPo-Akku 271
 Lithium-Polymer-Akku 271
 Litze 274
 Logik 90
 Logikpegel 51
 Löschroboter 21
 Löten 281
 LötKolben 281
 Lotuseffekt 18
 Lötzinn 281
 Low-Pegel 51
 Low-Signal 67

M

M3-Schraube 121
 Maker-Bewegung 58
 Maker-Faire 26
 Makerspace 26
 Mars 22
 Maschinensprache 69
 Masse 31, 66, 112
 Mathematik 91
 Mathematische Operation 92
 MCU 54
 Menschliche Gestalt 21
 Metallbohrer 120
 Microchip Technology 55
 μ C 54
 Mikrocontroller 54, 105
 μ Controller 54
 Mikroschalter 153
 Militär 21
 Millisekunde 52
 MINT 62
 Minuspol 30
 Mittelwert
 arithmetischer 222
 Modulo 92
 Montagematerial 121
 Motor 100
 Motordrehzahl 104
 Motorhalterung 107
 Motortreiber 117
 Multimeter 254

N

- Nachbau 59
- Nanobot 22
- Nanosekunde 56
- Neopixel 261
- NEPO 43, 69
 - Absolutwert 194
 - Addition 92
 - Aktor digital/analog 85
 - Berechne 92
 - Division 92
 - falsch 90
 - Funktion 95
 - gib Abstand cm Ultraschallsensor HC-SR04 86
 - gib digitalen/analogen Wert 86
 - gleich 90
 - größer als 90
 - größer als oder gleich 90
 - kleiner als 90
 - kleiner als oder gleich 90
 - Multiplikation 92
 - nicht 91
 - Potenzieren 92
 - Quadratwurzel 194
 - Rest von 92
 - Schalte LED 84
 - Schleife abbrechen 88
 - Schreibe 94
 - Schreibe digitalen/analogen Wert 84
 - Sensor digital/analog 86
 - Servomotor SG90 82
 - Spiele 84
 - Start 87
 - Subtraktion 92
 - Summer HYT120 84
 - Text 93
 - Ultraschallsensor HC-SR04 87
 - und/oder 91
 - ungleich 90
 - Variable 95
 - Vergleich 90
 - wahr 90
 - wahr/falsch 91
 - Warte bis 89
 - Warte ms 90
 - wenn ... mache ... sonst 87
 - Wert (Zahl) 92
 - Wiederhole n-Mal 88
 - Wiederhole unendlich oft 87, 88
 - Zähle ... von ... solange ... 88
 - Zeige auf seriellem Monitor 82
- Nepo4Arduino 41
- Netzteil 31, 64
- Nucleo 64
- NXT 244

O

- Oberfläche 74
- Objekt
 - Scratch 239
- Objektorientiert 237
- Offene Werkstatt 26
- Open Roberta Lab 14, 41, 74, 245
 - Fehler 80, 135
- Open-Roberta-USB-Software 37, 39
- Opportunity 22

P

- Pathfinder-Projekt 22
- Pause
 - einlegen 52
- PC 34
- Peripherie 55
- π 141
- Pi 141
- PIC32 64
- PICAXE 64
- Pinout 64
- Platine 67
 - aufstecken 111
- Playback-Verfahren 19
- Plexiglas 119
- Pluspol 30
- Polung 30, 101
- Polyfuse 106
- Potenzimeter 254
- Poti 158, 252
- Programm
 - Start 48
- Programmbibliothek 188
- Programmiersprache 68
 - NEPO 43
 - visuelle 69

Programmiersprache NEPO 14
 Programmierung 13
 Prozessor 55
 Pulsweitenmodulation 133, 152, 187
 Puzzleteile 69
 PWM 67, 119, 133, 187
 PWM-Enable 134
 Python 68

Q

Quellcode 69, 81, 258
 anzeigen 71
 Quetschen 280

R

R2-D2 24
 Radsensor 138
 Rasenmähen 20
 Raspberry Pi 62
 Raspi 62
 Rauschen
 elektronisches 174
 RCX 244
 Rechte-Hand-Methode 216
 Reflexionssensor 158
 Reifen 101
 Rescue Robot League 163
 Reset 66, 113
 Rest
 Division 92
 Rettung 21
 RGB-LED 28
 Ritzel 104
 RoboCup 18
 Roboter
 Definition 17
 humanoid 10
 im Film 23
 mit Taschenlampe lenken 165
 Roboterarm 195
 Robotergesetze 22
 Roboterkonfiguration 44, 80
 Rover 21

S

SAR 21
 Saugroboter 156
 Schalter 156
 Schaltplan 250
 Schaltungssimulation 249
 Scheinwerfer 150
 Schleife 88
 Schlitzscheibe 138
 Schrägstrich 259
 Schrittweite 89
 Schrumpfschlauch 283
 Schutzfolie 107
 Science-Fiction 22
 Scratch 68, 236
 Scratch4Arduino 236
 Search and Rescue 21
 Sechsfüßer 209
 Seitenschneider 274
 SEL 113
 Sensor 86
 Sensor-Shield 111
 Serial Monitor 83
 Serial Port 34
 Serviceroboter 20
 Servohorn 180
 Servomotor 218
 SG90 180
 Shield 111
 Sicherung 106
 Simulation
 Hardware 235
 starten 256
 Slash 259
 SMD 55
 Sn60Pb40 281
 Snap! 68
 Snap4Arduino 236
 SoC 54
 Software 67
 Sojourner 22
 Sonar
 Entfernung messen 170
 Sonnenlicht 168
 Sourcecode 69, 81
 Spannung 31, 66

Spannungsregler 113
Speed Encoder 138
Speicher 55
Speichern
 Open Roberta Lab 79
Spielzeugroboter 17
Spirit 22
Spline 180
Sprachassistent 24
Spülbürste 15
Spule 100
strand () 173
Star Wars 24
Start
 Programm 48
Startwert 89
Staubsaugen 20
Steckbrücke 113
Steckdose 29
Stellwinkel 187
STM32 64
Stromversorgung 64, 188
Summer 84, 148
Surface-mounted device 56
System-on-a-Chip 54

T

T-800 23
Taster 152
TCRT5000 158
Teach-In 19
Teensy 64
Textdisplay 261
Tilde 67
Tinkercad 249
Token 40, 42
TowerPro SG90 180
Traktion 116
Transformers 24
Treiber 30
Tricktechnik 23
Trimpoti 158
Turing-Test 24

U

Übertragen 69
U-Boot
 Entfernung messen 170
Ultraschall
 Entfernung messen 169
Ultraschallsensor 86, 246
Umlaute
 für Variablen- und Funktionsnamen
 143
Unix 63
Uno R3 58
Unterricht 235
Updatemöglichkeit 55
URL 11, 79
USB 29, 33, 60
 Open-Roberta-USB-Software 37, 39
 verbinden 40
USB-Controller 33

V

Variable 89, 93
VCC 112
Verbrennungsmotor 100
Verdrillen 275
Vergleich 90
Verkehrsampel 72
Vibrationsmotor 15
VIN 66
Visuelle Programmiersprache 69
Vorwiderstand 28

W

Wackelkontakt 113
WALL-E 24
Wartezeit 52
Wettbewerb 163, 244
Widerstand 28, 156, 254
World Robot Olympiad 244

X

XML 80
XML-Datei 12

Z

Zahl

einbauen 92

gerade 93

ungerade 93

Zählschleife 89

Zahnbürste 15

Zahnrad 104

Zink-Kohle-Batterie 132

Zufallszahl 172

Zugentlastung 280

Zweibeiner 199

Zwischenablage 41