



Komplett
in Farbe!



Terry Griffin

PROGRAMMIEREN LERNEN MIT EV3

Vom Einsteiger zum Meisterprogrammierer
mit LEGO® MINDSTORMS® EV3

PROGRAMMIEREN LERNEN MIT EV3

**Vom Einsteiger zum Meisterprogrammierer
mit Lego® Mindstorms®**

Terry Griffin

Lektorat: Dr. Michael Barabas
Übersetzung: G&U Language & Publishing Services GmbH, www.gundu.com
Satz: G&U Language & Publishing Services GmbH, www.gundu.com
Copy-Editing: Ursula Zimpfer, Herrenberg
Herstellung: Susanne Bröckelmann
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

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:

Buch 978-3-86490-275-8

PDF 978-3-86491-714-1

ePub 978-3-86491-715-8

Mobi 987-3-86491-716-5

1. Auflage 2015

Copyright der deutschen Übersetzung © 2015 dpunkt.verlag GmbH

Wieblinger Weg 17 · 69123 Heidelberg

Copyright der amerikanischen Originalausgabe: © 2014 by Terry Griffin

Titel der Originalausgabe: The Art of LEGO® MINDSTORMS® EV3 Programming

No Starch Press, Inc. · 245 8th Street, San Francisco, CA 94103 | www.nostarch.com

ISBN: 978-1-59327-568-6

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten.

Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware- Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

LEGO, LEGO-Figuren und LEGO-Bausteine sowie MINDSTORMS sind Warenzeichen der LEGO-Gruppe.

Dieses Buch ist von der LEGO-Gruppe weder unterstützt noch autorisiert worden.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert.

Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die im Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Papier
plus⁺
PDF.

Zu diesem Buch – sowie zu vielen weiteren dpunkt.büchern –
können Sie auch das entsprechende E-Book im PDF-Format
herunterladen. Werden Sie dazu einfach Mitglied bei dpunkt.plus⁺:

www.dpunkt.de/plus

Für meine Familie, die all die Arbeit lohnenswert macht.
Und für Bella, die mir morgens einen Grund zum Aufstehen gibt.

Der Autor

Terry Griffin arbeitet seit über 20 Jahren als Softwareingenieur und hat den größten Teil dieser Zeit damit zugebracht, Software zur Steuerung der unterschiedlichsten Arten von Maschinen zu entwickeln. Er hat an der Universität von Massachusetts seinen Master in Informatik gemacht und am College sowie in der Erwachsenenbildung Programmierkurse gegeben. Als lebenslanger Lego-Fan hat er *The Art of LEGO MINDSTORMS NXT Programming* (No Starch Press) geschrieben, um seiner Frau, einer begeisterten Lehrerin für Naturwissenschaften und Mathematik für Sekundarstufe II, dabei zu helfen, die faszinierenden NXT-Roboter auch im Unterricht einzusetzen. Heute arbeitet er für das Ion Microscopy Innovation Center von Carl Zeiss und entwickelt dort Software zur Steuerung von Elektronen- und Ionenmikroskopen.

Die Fachgutachter

Daniele Benedettelli ist weltweit für seine originellen Lego-Roboter bekannt, vor allem für seine Lösungssysteme für den Zauberwürfel und seine menschenähnlichen Roboter. Als LEGO MINDSTORMS Community Partner (MCP) hilft er bei der Entwicklung und dem Test neuer Mindstorms-Produkte. Er hat einen Master in Robotik und Automatisierungstechnik von der Universität in Siena. Überall auf der Welt hält er Vorträge und Seminare über Robotik und Informations- und Kommunikationstechnologie. Außerdem unterrichtet er Robotik in der Oberstufe und entwirft als freier Mitarbeiter Lego-Modelle für die Lego-Unterrichtsprogramme. Aus seiner Feder stammt das Buch *Das LEGO-MINDSTORMS-EV3-Labor* (dpunkt).

Rob Torok arbeitet als Lehrer in Tasmanien und verwendet schon seit 2001 Lego-Robotik im Unterricht. Er hat Teams beim RoboCup Junior und dem FIRST Robotics Competition betreut und gibt den Online-Robotikkurs SmartBots. 2010 hat er sechs Monate im Tufts Center for Engineering Education and Outreach (CEEEO) in Boston verbracht, mit dem er nach wie vor eng zusammenarbeitet. Zurzeit fungiert er als Redakteur sowohl für <http://LEGOengineering.com/> als auch für <http://LEGOeducation.com.au/>.

Danksagung

Ich möchte meiner Familie für ihre Geduld während der Zeit danken, in der ich dieses Buch schrieb. Mein besonderer Dank gilt meiner Frau Liz, die zahllose Stunden mit der Durchsicht des Textes verbrachte und es gestattete, dass die Roboter den Esszimmertisch eroberten.

Ohne die Hilfe und die Unterstützung von Bill Pollock und allen Mitarbeitern bei No Starch Press wäre dieses Buch nicht möglich gewesen. Es war mir ein Vergnügen, mit Seph Kramer, Laurel Chun und Jennifer Griffith-Delgado zusammenzuarbeiten. Ihr Wissen und ihre Fachkenntnisse waren für die Fertigstellung dieses Projekts unverzichtbar.

Ich möchte auch meinen Fachgutachtern Daniele Benedettelli und Rob Torok danken. Ihre Kenntnisse über EV3 und Robotik im Allgemeinen waren eine große Hilfe, um den wirklich wichtigen Stoff auszuwählen und ihn technisch korrekt darzustellen.

Übersicht

Einleitung	xxi
Kapitel 1 Lego und Roboter: eine großartige Kombination	1
Kapitel 2 Die EV3-Programmierungsumgebung	7
Kapitel 3 TriBot: der Testroboter	17
Kapitel 4 Bewegung	43
Kapitel 5 Sensoren	57
Kapitel 6 Programmablauf	77
Kapitel 7 Das Programm WallFollower: Orientierung im Labyrinth	91
Kapitel 8 Datenleitungen	105
Kapitel 9 Datenleitungen am Schalterblock	117
Kapitel 10 Datenleitungen am Schleifenblock	127
Kapitel 11 Variablen	135
Kapitel 12 Eigene Blöcke	149
Kapitel 13 Der Mathe- und der Logikblock	161
Kapitel 14 Tasten, Statusleuchte und Anzeige des EV3-Steins	177
Kapitel 15 Arrays	187
Kapitel 16 Dateien	205
Kapitel 17 Datenprotokollierung	221
Kapitel 18 Multitasking	231
Kapitel 19 Eine PID-gesteuerte Version des Programms LineFollower	243
Anhang A Kompatibilität von NXT und EV3	257
Anhang B Websites rund um EV3	259
Index	261

Inhaltsverzeichnis

Einleitung	xxi
Zielpublikum	xxi
Voraussetzungen	xxi
Der Inhalt dieses Buches	xxi
Wie du dieses Buch lesen solltest	xxiii
1	
Lego und Roboter: eine großartige Kombination	1
Lego Mindstorms EV3	1
Der Lego-Mindstorms-EV3-Kasten	2
Die Lego-Mindstorms-EV3-Software	3
Software, Firmware und Hardware	3
Kunst und Wissenschaft	3
Merkmale von guten Programmen	4
Was du in diesem Buch lernst	4
Die Online-Community zu Lego Mindstorms	5
Wie geht es weiter?	5
2	
Die EV3-Programmierungsumgebung	7
Die Mindstorms-Software im Überblick	7
A: Der Programmierbereich	8
B: Der Inhalts-Editor	8
C: Die Programmierpaletten	8
D: Die Hardwareseite	8
E: Schaltflächen zum Herunterladen und Ausführen	8
EV3-Programme schreiben	9
Allgemeiner Aufbau von Blöcken	9
Dein erstes Programm	10
Deine Änderungen speichern	11
Das Programm ausführen	11
Sicherungskopien anlegen	11
Projekteigenschaften	12
Dein zweites Programm	12
Kommentare	14
Kommentare hinzufügen	14
Tipps für die Arbeit mit Kommentaren	15
Kontexthilfe	15
Zusammenfassung	15

3

TriBot: der Testroboter	17
Bestandteile des TriBots	17
Motor und Räder	20
Die Laufrolle	24
Die Laufrolle aus Teilen der Home Edition bauen	24
Die Laufrolle aus Teilen der Education Edition bauen	26
Den EV3-Stein hinzufügen	27
Den Infrarot- oder den Ultraschallsensor montieren	27
Den Farbsensor anschließen	28
Den Kreisel sensor anbauen (Education Edition)	30
Die Stoßstange mit Berührungssensor bauen	30
Die Kabel anschließen	33
Den Berührungssensor verkabeln	33
Den Infrarot- oder Ultraschallsensor verkabeln	33
Den Farbsensor verkabeln	34
Den Kreisel sensor verkabeln (Education Edition)	34
Die Motoren verkabeln	34
Alternative Platzierung des Farbsensors	34
Alternative Platzierung des Infrarot- oder Ultraschallsensors	35
Den Hubarm bauen	36
Zusammenfassung	41

4

Bewegung	43
Die EV3-Motoren	43
Der Bewegungslenkungsblock	43
Modus	44
Lenkung	44
Leistung	45
Dauer	45
Am Ende bremsen	46
Anschlüsse	46
Die Anschlussansicht	46
Die Anschlussansicht auf dem EV3-Stein	47
Das Programm ThereAndBack	47
Vorwärtsbewegung	47
Wenden	48
Einen einzelnen Block testen	48
Zurück zur Ausgangsposition	48
Das Programm AroundTheBlock	49
Die erste Seite und die erste Ecke	49
Die drei anderen Seiten und Ecken	49
Das Programm testen	50
Der Hebel lenkungsblock	51

Die Blöcke für große und mittlere Motoren	51
Der Hubarm	51
Der Block »Motor umkehren«	52
Ein Problem bei der Einstellung »Auslaufen«	53
Noch mehr zum Ausprobieren	55
Zusammenfassung	55
5	
Sensoren	57
Sensoren verwenden	57
Der Berührungssensor	58
Das Programm BumperBot	59
Vorwärtsbewegung	59
Hindernisse erkennen	59
Zurücksetzen und wenden	59
Aufgabe 5-1	60
Aufgabe 5-2	60
Testen	60
Der Farbsensor	60
Der Farbmodus	60
Stärke des reflektierten Lichts	61
Stärke des Umgebungslichts	62
Die Anschlussansicht	62
Aufgabe 5-3	62
Das Programm IstBlue	62
Der Schalterblock	62
Das Programm verbessern	63
Das Programm LineFinder	65
Den Schwellenwert mithilfe der Anschlussansicht finden	65
Aufgabe 5-4	66
Aufgabe 5-5	66
Der Infrarotsensor und die Fernsteuerung	66
Der Nähemodus	67
Die Modi Signal-Richtung und Signal-Nähe	67
Der Fernsteuerungsmodus	67
Aufgabe 5-6	68
Das Programm BumperBotWithButtons	68
Der Ultraschallsensor	68
Die Modi »Distanz in Zentimetern« und »Distanz in Zoll«	69
Der Modus Anwesenheit/Wahrnehmen	69
Das Programm DoorChime	69
Eine Person wahrnehmen	69
Den Gong abspielen	70
Die Wiedergabe des Gongs beenden	70

Der Kreisel­sen­sor	71
Der Ra­ten­mo­dus	71
Der Win­kel­mo­dus	71
Den Win­kel zu­rück­set­zen	72
Das Pro­gramm GyroTurn	72
Aufgabe 5-7	72
Der Mo­to­rum­drehungs­sen­sor	73
Das Pro­gramm BumperBot2	73
Noch mehr zum Ausprobie­ren	74
Zusammenfassung	75

6

Pro­grammab­lauf	77
Der Schal­ter­block	77
Die Be­dingung fest­le­gen	77
Die Grö­ße eines Blocks än­dern	78
Das Pro­gramm LineFol­lo­wer	78
Das Grund­pro­gramm	79
Einen Schwellenwert für den Farbsen­sor wäh­len	79
Die Be­we­gungs­len­kungs­blö­cke ein­rich­ten	80
Das Pro­gramm testen	80
Mehr als zwei Wahl­mög­lich­kei­ten	80
Das Pro­gramm testen	82
Die Re­gis­ter­ansicht	82
Aufgabe 6-1	83
Das Pro­gramm RedOrBlue	83
Rote Ob­jek­te er­ken­nen	84
Einen neuen Fall hinzu­fü­gen	84
Der Standardfall	85
Aufgabe 6-2	85
Der Schleifenblock	85
Der Schleifen-Interrupt-Block	86
Das Pro­gramm BumperBot3	86
Noch mehr zum Ausprobie­ren	89
Zusammenfassung	89

7

Das Pro­gramm WallFol­lo­wer: Ori­en­tie­rung im La­by­rinth	91
Pseudocode	91
Aus einem Labyrinth herausfinden	92
Anforderungen an das Programm	93
Annahmen	94
Erster Entwurf	95
Verwendung der Education-Ausgabe	96

Einer geraden Wand folgen	96
Den Code schreiben	96
Testen	97
Um die Ecke biegen	98
Den Code schreiben	99
Testen	99
Durch eine Öffnung fahren	100
Den Code schreiben	101
Testen	101
Klangblöcke für das Debugging	102
Abschlusstest	103
Noch mehr zum Ausprobieren	103
Zusammenfassung	104

8

Datenleitungen **105**

Was sind Datenleitungen?	105
Das Programm GentleStop	105
Das Programm schreiben	106
Tipps für die Verwendung von Datenleitungen	107
Aufgabe 8-1	108
Das Programm SoundMachine	108
Die Lautstärke regeln	109
Den Matheblock verwenden	109
Die Tonhöhensteuerung hinzufügen	110
Datentypen	111
Die Frequenz- und Lautstärkewerte anzeigen	111
Den Textblock verwenden	112
Beschriftungen für angezeigte Werte hinzufügen	112
Die Lautstärke anzeigen	113
Aufgabe 8-2	113
Aufgabe 8-3	113
Aufgabe 8-4	115
Noch mehr zum Ausprobieren	115
Zusammenfassung	115

9

Datenleitungen am Schalterblock **117**

Die Wertmodi des Schalterblocks	117
Das Programm GentleStop umschreiben	118
Daten in einen Schalterblock übergeben	119
Vorteile bei der Verwendung eines Sensorblocks	119
Daten aus einem Schalterblock heraus übergeben	120
Aufgabe 9-1	123

Das Programm LineFollower vereinfachen	123
Aufgabe 9-2	125
Noch mehr zum Ausprobieren	125
Zusammenfassung	125
10	
Datenleitungen am Schleifenblock	127
Der Logikmodus	127
Aufgabe 10-1	128
Der Schleifenindex	128
Das Programm LoopIndexTest	128
Eine Schleife neu starten	128
Der letzte Wert des Schleifenindex	129
Das Programm SpirallineFinder	130
Einen Spiralkurs fahren	130
Linien während der Spiralbewegung erkennen	131
Aufgabe 10-2	132
Bessere Drehungen mit dem Kreiselsensor	132
Noch mehr zum Ausprobieren	133
Zusammenfassung	133
11	
Variablen	135
Der Variablenblock	135
Das Programm RedOrBlueCount	136
Variablen erstellen und initialisieren	137
Sinnvolle Variablenamen	137
Die Anfangswerte anzeigen	138
Die roten Objekte zählen	138
Die blauen Objekte zählen	139
Variablen auf der Projekteigenschaftenseite verwalten	140
Der Vergleichsblock	141
Das Programm LightPointer	141
Die Variablen definieren	142
Die Lichtquelle finden	142
Variablen initialisieren	144
Das Programm LightPointer schreiben	144
Der Konstantenblock	146
Aufgabe 11-1	146
Aufgabe 11-2	146
Noch mehr zum Ausprobieren	147
Zusammenfassung	147

12

Eigene Blöcke	149
Eigene Blöcke erstellen	149
Die Palette Eigene Blöcke	151
Eigene Blöcke bearbeiten	151
Aufgabe 12-1	151
Der Eigene Block LogicToText	152
Parameter hinzufügen, entfernen und verschieben	156
Das Register Parametereinrichtung	156
Der Eigene Block DisplayNumber	156
Eigene Blöcke und das Debugging	158
Parameter von Eigenen Blöcken ändern	158
Variablen und Eigene Blöcke	158
Aufgabe 12-2	160
Noch mehr zum Ausprobieren	160
Zusammenfassung	160

13

Der Mathe- und der Logikblock	161
Der Modus »Großer Funktionsumfang« des Matheblocks	161
Unterstützte Operatoren und Funktionen	161
Der Modulo-Operator	162
Fehler im Matheblock	162
Ein proportionales Spurfolgeprogramm	164
EV3-Zeitgeber	165
Aufgabe 13-1	166
Das Programm DisplayTimer	166
Den Zeitgeber-Messwert in Minuten und Sekunden aufteilen	166
Den Text für die Anzeige zusammenstellen	167
Der Rundungsblock	168
Aufgabe 13-2	169
Aufgabe 13-3	169
Der Zufallsblock	169
Das Programm BumperBot mit einer zufälligen Drehung	170
Der Logikblock	171
Logische Operationen im Programm BumperBot	171
Der Bereichsblock	172
Das Programm agAlong	173
Das Programm GyroPointer	174
Aufgabe 13-4	175
Noch mehr zum Ausprobieren	175
Zusammenfassung	175

14

Tasten, Statusleuchte und Anzeige des EV3-Steins 177

Die Tasten des EV3-Steins	177
Das Programm PowerSetting	178
Der Anfangswert und die Schleife	178
Den aktuellen Wert anzeigen	179
Den Leistungswert anpassen	179
Das Programm testen	179
Schnelle Änderung des Werts	179
Aufgabe 14-1	181
Die Stein-Statusleuchte	181
Das Programm ColorCopy	181
Aufgabe 14-2	182
Der Anzeigeblock	182
Ein Bild anzeigen	182
Das Programm Eyes	183
Auf dem EV3-Bildschirm zeichnen	184
Das Programm EV3Sketch	184
Noch mehr zum Ausprobieren	186
Zusammenfassung	186

15

Arrays 187

Überblick und Terminologie	187
Arrays erstellen	187
Der Block Array-Operationen	188
Der Längemodus	188
Der Modus »Lesen aus Index«	188
Der Modus »Schreiben in Index«	189
Der Modus »Anfügen«	189
Das Programm ArrayTest	189
Aufgabe 15-1	191
Das Programm ButtonCommand	191
Das Befehlsarray erstellen	191
Die Befehle anzeigen	192
Die Befehle ausführen	193
Aufgabe 15-2	194
Das Programm ColorCount	195
Der Eigene Block ColorToText	195
Der Eigene Block AddColorCount	197
Eine Klangdatei über eine Datenleitung auswählen	197
Initialisierung	198
Farben zählen	200
Das Programm MemoryGame	200
Der Anfang der Schleife	200
Die Farbfolge zusammenstellen	200

Der Eigene Block WaitForButton	201
Die Antwort des Benutzers überprüfen	202
Noch mehr zum Ausprobieren	203
Zusammenfassung	203

16

Dateien	205
Der Dateizugriffsblock	205
Den Dateinamen festlegen	205
In eine Datei schreiben	205
Probleme mit Dateinamen verhindern	206
Aus einer Datei lesen	206
Den Highscore in MemoryGame speichern	207
Das Programm FileReader	209
Aufgabe 16-1	210
Das Dateiende finden	210
Dem Programm ColorCount ein Menü hinzufügen	211
Der Eigene Block CreateMenu_CC	211
Der Eigene Block SelectOption	212
Die neue Struktur des Programms ColorCount	215
Objekte zählen	217
Die Zähldaten speichern und laden	217
Testen	217
Aufgabe 16-2	218
Speicherverwaltung	218
Noch mehr zum Ausprobieren	219
EV3-Textdateien unter Windows	220
Zusammenfassung	220

17

Datenprotokollierung	221
Datenerfassung mit dem EV3-System	221
Der Messwert »Stromstärke«	221
Das Programm CurrentPowerTest	221
Der Eigene Block LogData	224
Das Programm CurrentPowerTest2	225
Aufgabe 17-1	226
Die aktuelle Leistung mit dem Bewegungslenkungsblock testen	226
Das Programm SteeringTest	227
Das Programm VerifyLightPointer	228
Aufgabe 17-2	229
Die Menge der Daten beeinflussen	229
Noch mehr zum Ausprobieren	230
Zusammenfassung	230

18	
Multitasking	231
Mehrere Startblöcke	231
Der Block »Programm beenden«	232
Aufgabe 18-1	232
Blockierende Schleifen vermeiden	232
Ein Blinklicht zum Programm DoorChime hinzufügen	233
Programmablaufregeln	236
Startblöcke und Datenleitungen	236
Werte aus einem Schleifen- oder Schalterblock verwenden	238
Eigene Blöcke verwenden	238
Zwei Sequenzen zeitlich abstimmen	238
Schwierigkeiten vermeiden	240
Noch mehr zum Ausprobieren	241
Zusammenfassung	241
19	
Eine PID-gesteuerte Version des Programms LineFollower	243
Der PID-Regler	243
Proportionalregelung	244
Die Rohdaten	245
Gute und schlechte Zonen	245
Den niedrigsten und den höchsten Messwert des Sensors bestimmen	247
Die Sensormesswerte und den Sollwert normalisieren	249
Die Proportionalregelung des Programms LineFollower verbessern	249
Die Kalibrierungsmodi des Farbsensors	250
Die PID-Regelung hinzufügen	252
Den Differenzialterm hinzufügen	252
Den Integralterm hinzufügen	253
Den Regler einstellen	256
Noch mehr zum Ausprobieren	256
Zusammenfassung	256
A	
Kompatibilität von NXT und EV3	257
Motoren	257
Sensoren	257
Software	258
B	
Websites rund um EV3	259
Index	261

Einleitung

In diesem Buch geht es darum, Programme für Lego-Mindstorms-EV3-Roboter zu schreiben. Die EV3-Software ist sehr leistungsfähig, und hier lernst du, wie du sie am besten nutzt, und erwirbst die Programmierkenntnisse, die du zum Schreiben deiner eigenen Programme benötigst.

Zielpublikum

Dieses Buch ist für alle geeignet, die lernen wollen, wie sie Programme zur Steuerung ihrer EV3-Roboter schreiben können. Es spielt keine Rolle, ob du ein junger Roboterfan bist, ein Erwachsener, der Robotik für Kinder unterrichtet, ein Elternteil, ein FIRST-LEGO-League-Teambetreuer oder ein Lehrer, der EV3 im Unterricht einsetzt. Beim Schreiben dieses Buches habe ich einerseits darauf geachtet, den Stoff auch für junge Leser aufzubereiten, und andererseits genügend in die Tiefe zu gehen, um Schülern und Lehrern das Wie und Warum der EV3-Programmierung zu erschließen.

Voraussetzungen

Zum Testen sämtlicher Programme wird ein einziger Allzweckroboter verwendet, den du sowohl mit der normalen Einzelhandels- als auch der Education-Ausgabe von EV3 bauen kannst. Es gibt bei den Programmen nur ein paar wenige wichtige Unterschiede zwischen den beiden Ausgaben, auf die ich jeweils hinweisen werde. Fast der gesamte hier dargestellte Stoff kann mit beiden Ausgaben nachvollzogen werden.

Vorkenntnisse in Programmierung sind nicht erforderlich. Die EV3-Software ist leistungsfähig, aber einfach anzuwenden, und bildet ein hervorragendes Werkzeug für die erste Einführung in die Programmierung.

Der Inhalt dieses Buches

In diesem Buch geht es um die Programmierung von EV3-Robotern und nicht um den mechanischen Teil beim Bauen. Alle Programme in diesem Buch sind entweder für einen Allzweckroboter gedacht oder auch nur für den EV3-Stein. Du lernst hier, wie du mit den wichtigsten Elementen der EV3-Software arbeitest, also z. B. mit Blöcken, Datenleitungen, Dateien und Variablen, und wie diese Elemente zusammenwirken. Außerdem erfährst du etwas über gute Programmier Techniken, über schlechte Angewohnheiten, die du vermeiden solltest, und über Debugging-Verfahren, mit denen du frustrierende Erfahrungen in Grenzen halten kannst, um mehr Spaß beim Programmieren zu haben.

In diesem Buch findest du Schritt-für-Schritt-Anleitungen und Erklärungen für viele EV3-Programme. Das sind einerseits kleine Beispiele, die dazu dienen, dir die Funktionsweise von EV3-Programmen genau deutlich zu machen, andererseits aber auch anspruchsvolle Programme, die vielschichtige Verhaltensweisen zeigen. Zwischendurch wirst du auch immer wieder Programmieraufgaben finden, in denen du dazu aufgefordert wirst, dich selbstständig mit Fragen der EV3-Programmierung zu befassen, um das Gelernte zu vertiefen.

Zu Anfang des Buches steht eine Einführung in den EV3-Kasten und die Software, die du zum Schreiben der Programme verwendest. Darauf folgt die Bauanleitung für den Testroboter. Die nächsten Kapitel behandeln die Grundlagen der EV3-Software, was in Kapitel 7 mit einem Programm seinen Höhepunkt findet, mit dem der Roboter aus einem Labyrinth herausfinden kann. Darauf folgen mehrere Kapitel, die fortgeschrittenere Elemente der Programmierumgebung behandeln. Am Schluss des Buches steht ein anspruchsvolles Spurfolgeprogramm mit PID-Regler. Der folgende Überblick zeigt, was du in den einzelnen Kapiteln lernen wirst.

Kapitel 1: Lego und Roboter: eine großartige Kombination

Das erste Kapitel gibt eine kurze Einführung in die Software Lego Mindstorms EV3. Außerdem stellt es die Unterschiede zwischen der Einzelhandels- und der Education-Ausgabe und deren Auswirkungen auf die Experimente in diesem Buch vor.

Kapitel 2: Die EV3-Programmierungsumgebung

In diesem Kapitel werden die einzelnen Merkmale der EV3-Software vorgestellt. Anhand von zwei einfachen Beispielen erfährst du, wie du Programme erstellst und auf dem EV3-Stein ausführst. Hier lernst du auch, wie du Blockparameter änderst, Kommentare hinzufügst und die Anschlussansicht nutzt.

Kapitel 3: TriBot: der Testroboter

In diesem Kapitel baust du den Testroboter TriBot. Diesen Allzweckroboter wirst du zum Prüfen sämtlicher Programme in diesem Buch verwenden.

Kapitel 4: Bewegung

In diesem Kapitel geht es um die EV3-Motoren und die Blöcke zu deren Steuerung. Hier erstellst du mehrere Programme, die die typische Verwendung dieser Blöcke veranschaulichen. Außerdem wirst du auf einige Fallstricke hingewiesen, über die Lego-Bastler häufig stolpern.

Kapitel 5: Sensoren

Dieses Kapitel behandelt die EV3-Sensoren: den Berührungs-, den Farb-, den Ultraschall-, den Infrarot-, den Kreisel- und den Motorumdrehungssensor. Für jeden dieser Sensoren schreibst du ein Beispielprogramm. Du erfährst hier auch, wie du dir die Sensorwerte in der Anschlussansicht anschauen kannst, während du ein Programm entwickelst oder ausführst.

Kapitel 6: Programmablauf

Schwerpunkt dieses Kapitels sind der Schalterblock (mit dem du in deinen Programmen Entscheidungen treffen kannst) und der Schleifenblock (mit dem sich Aktionen in einem Programm wiederholen lassen). Mit diesen Programmablaufblöcken schreibst du ein einfaches Spurfolgeprogramm.

Kapitel 7: Das Programm WallFollower: Orientierung im Labyrinth

Nachdem du nun die Grundlagen der EV3-Programmierung beherrschst, bist du jetzt in der Lage, kompliziertere Aufgabenstellungen anzupacken. In diesem Kapitel lernst du, wie du ein Wandfolgeprogramm, mit dem dein Roboter aus einem Labyrinth herausfinden kann, gestaltest, schreibst und darin Fehler behebst.

Kapitel 8: Datenleitungen

Datenleitungen gehören zu den vielseitigsten Elementen für die EV3-Programmierung. In diesem Kapitel erfährst du, was Datenleitungen sind und wie du sie wirkungsvoll einsetzt. Die Beispielprogramme zeigen dir, wie du mithilfe von Datenleitungen Informationen von Sensoren abrufst und wie du anhand dieser Sensordaten einen Motor steuerst.

Kapitel 9: Datenleitungen am Schalterblock

In diesem Kapitel geht es um die anspruchsvolleren Merkmale des Schalterblocks, die bei der Verwendung von Datenleitungen zur Verfügung stehen. Außerdem lernst du hier, wie du Daten über Datenleitungen in einen Schalterblock hinein- und aus ihm herausführst.

Kapitel 10: Datenleitungen am Schleifenblock

In diesem Kapitel lernst du den Umgang mit Datenleitungen an Schleifenblöcken kennen. Du erstellst ein Programm, das den Roboter in einem rechteckigen Spiralkurs bewegt, und verwendest dabei neue Techniken wie Schleifenzähler und die Beendigungsbedingung der Schleife.

Kapitel 11: Variablen

Thema dieses Kapitels sind der Variablen- und der Konstantenblock. Du lernst hier, wie du Variablen hinzufügst und verwaltest, um Werte zu speichern und zu aktualisieren.

Kapitel 12: Eigene Blöcke

Ein Eigener Block ist ein neuer Block, den du erstellst, indem du mehrere andere Blöcke gruppierst. In diesem Kapitel erfährst du, wie du Eigene Blöcke anlegst, in deinen Programmen verwendest und sie von einem Projekt zu einem anderen überträgst.

Kapitel 13: Der Mathe- und der Logikblock

Dieses Kapitel beschreibt die Blöcke für mathematische und logische Operationen: den Mathe-, den Logik-, den Bereichs-, den Rundungs- und den Zufallsblock. Du lernst die erweiterte Verwendung dieser Blöcke kennen, indem du einige der in früheren Kapiteln geschriebenen Programme ausbaust.

Kapitel 14: Tasten, Statusleuchte und Anzeige des EV3-Steins

In diesem Kapitel lernst du, wie du ein Programm über die Stein-Tasten steuerst, wie du mithilfe des Stein-Statusleuchten-Blocks die farbigen Lichter auf dem EV3-Stein verwendest und wie du auf die Anzeige auf dem Stein zugreifst. Als Beispiel wirst du dazu ein einfaches Zeichenprogramm schreiben.

Kapitel 15: Arrays

Thema dieses Kapitels sind Arrays und ihre Verwendung in der EV3-Programmierung. Hier entwickelst du ein Programm, mit dem du dem TriBot eine Liste von Befehlen vorgeben kannst, die er ausführen soll.

Kapitel 16: Dateien

In diesem Kapitel erfährst du, wie du mithilfe von Dateien Informationen auf dem EV3-Stein speicherst, wie du den Speicher des EV3-Steins verwaltest und wie du Dateien vom EV3-Stein auf den Computer überträgst und umgekehrt. Hier entwickelst du auch ein Programm, dessen Einstellungen du in einer Datei speicherst und später wieder von dort abrufen kannst.

Kapitel 17: Datenprotokollierung

Die Programme in diesem Kapitel zeigen dir, wie du den EV3-Stein zur Datenprotokollierung verwenden kannst. Hier zeige ich

dir die Grundlagen der Datenerfassung und -analyse. Außerdem nutzen wir die Datenprotokollierung, um ein tieferes Verständnis der Funktionsweise des Bewegungslenkungsblocks zu gewinnen.

Kapitel 18: Multitasking

Der EV3-Stein kann mehrere Gruppen von Blöcken parallel ausführen. Hier lernst du, wie du mehrere Sequenzen gleichzeitig ausführst und wie du Probleme vermeidest, die dabei häufig auftreten.

Kapitel 19: Eine PID-gesteuerte Version des Programms

LineFollower

Im letzten Kapitel verwenden wir anspruchsvolle EV3-Programmierfunktionen, um ein hoch entwickeltes Spurfolgeprogramm zu schreiben. Du lernst hier, wie du einen PID-Regler (Proportional-Integral-Differenzial) einsetzt, um den Roboter so zu steuern, dass er der Linie schnell und genau folgt.

Anhang A: Kompatibilität von NXT und EV3

In diesem Anhang geht es um die gemeinsame Verwendung von älteren NXT-Mindstorms- und EV3-Produkten.

Anhang B: Websites rund um EV3

Dieser Anhang stellt Websites mit Informationen über EV3-Programmierung vor.

Wie du dieses Buch lesen solltest

Um den größten Nutzen aus diesem Buch zu ziehen, solltest du beim Lesen die Schritt-für-Schritt-Anleitungen zum Erstellen der Beispielprogramme auf deinem Computer nachvollziehen. Programmieren gehört zu den Tätigkeiten, die man am besten durch Ausprobieren lernt. Du kannst auf jeden Fall viel mehr lernen, wenn du Programme selbst schreibst und mit ihnen herumexperimentierst, als wenn du nur über sie liest.

Zum besseren Verständnis der Programme und der zugehörigen Erklärungen ist es am besten, die Kapitel in der vorgegebenen Reihenfolge zu lesen. In den ersten Kapiteln werden mehrere Beispielprogramme eingeführt, die du dann in späteren Kapiteln, wenn du mehr über die EV3-Programmierung weißt, erweiterst. Wenn du am Ende des Buches angelangt bist, verfügst du über die Kenntnisse und Fähigkeiten eines erfahrenen EV3-Programmierers!

Lego und Roboter: eine großartige Kombination

Willkommen in der Welt der Robotik! Es ist noch gar nicht so lange her, da konnte man Roboter nur in Science-Fiction-Geschichten finden. Heutzutage gehören Roboter unzweifelhaft zur Realität und führen eine breite Palette wichtiger Aufgaben durch, von der Erforschung fremder Planeten und Unterwasservulkane über die Automontage bis zur Chirurgie. Abbildung 1-1 zeigt den Mars-Forschungsrover *Curiosity*. Du kannst im Elektrohandel inzwischen sogar Roboter kaufen, die den Boden fegen, während du dich ausruhst!

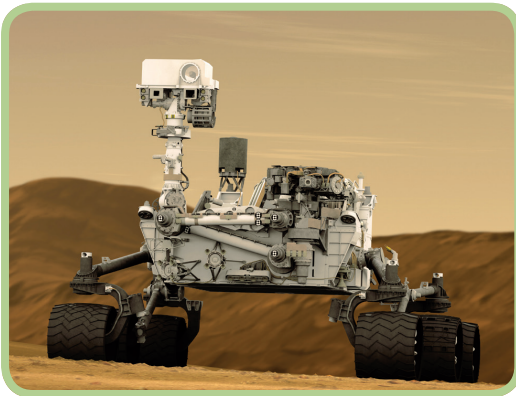


Abbildung 1-1: Mars-Forschungsrover
(mit freundlicher Genehmigung von NASA/JPL-Caltech)

Lego Mindstorms EV3

Mit dem Bausystem Lego Mindstorms EV3 kannst du eigene Roboter bauen. Abbildung 1-2 zeigt eines der vielen möglichen Modelle – einen einfachen Roboter, der dein Wohnzimmer erkundet.



Abbildung 1-2: Der Wohnzimmer-Rover

Es macht sehr viel Spaß, mit dem EV3-Kasten herumzuspielen, aber er ist mehr als nur ein Spielzeug. Lehrer in der Mittel- und Oberstufe setzen dieses System ein, um natur- und ingenieurwissenschaftliche Kenntnisse zu vermitteln. Die Lego-Gruppe hat mit *LEGO Education* sogar eine pädagogische Abteilung, die Lernmittel für den Einsatz von Lego-Produkten im Unterricht bereitstellt.

In Bildungswettbewerben wie der FIRST LEGO League (FLL), der World Robot Olympiad und RoboCup Junior nutzen Schüler aus aller Welt Mindstorms-Kästen, um Roboter zu bauen und mit ihnen die vorgegebenen Aufgaben zu lösen.

Den EV3-Kasten gibt es in zwei Versionen: Die Einzelhandelsversion (Home Edition) mit der Bestellnummer 31313 wird in normalen Geschäften verkauft und ist für das allgemeine Publikum gedacht, die Education-Version mit der Nummer 45544 wird von LEGO Education an Schulen, Erzieher und FLL-Teams verkauft. Zwischen den beiden Ausgaben gibt es kleine Unterschiede bei der Zusammenstellung der Lego-Bauteile und Sensoren. Außerdem verfügt die Education-Ausgabe der Software über einige zusätzliche Funktionen, die es erlauben, den EV3-Stein für wissenschaftliche Experimente einzusetzen. Für unsere Zwecke spielen diese Unterschiede jedoch keine Rolle, denn unseren Roboter kannst du sowohl mit der Einzelhandels- als auch mit der Education-Ausgabe bauen. Es gibt zwar kleine Unterschiede zwischen den Robotern der beiden Ausgaben (z. B. unterschiedliche Reifengrößen), die aber nicht ins Gewicht fallen.

Der EV3-Kasten ist die dritte Generation von Lego Mindstorms. Viele der Teile aus dem Vorgängersystem NXT können auch zusammen mit dem EV3-Stein eingesetzt werden. Einzelheiten darüber findest du in Anhang A.

Der Lego-Mindstorms-EV3-Kasten

Der EV3-Kasten enthält den intelligenten EV3-Stein, drei Motoren, mehrere Sensoren, eine Anleitung zum Herunterladen der Mindstorms-EV3-Software sowie Lego-Elemente zum Bauen von Robotern. Wie schon im vorherigen Abschnitt erwähnt, hängt der genaue Inhalt des Kastens von der Version ab.

Zu den Bauelementen gehören Zahnräder, Achsen, Pins und Balken aus der Produktreihe Lego Technic (siehe Abbildung 1-3). Diese Bauteile sind sowohl stabil als auch leicht und lassen sich zu anspruchsvollen beweglichen Teilen zusammensetzen. Dadurch sind sie ideal für den Roboterbau geeignet. Zur Erweiterung deiner Roboter kannst du auch andere Teile aus Technic-, Bionicle- und sogar herkömmlichen Lego-Kästen verwenden.

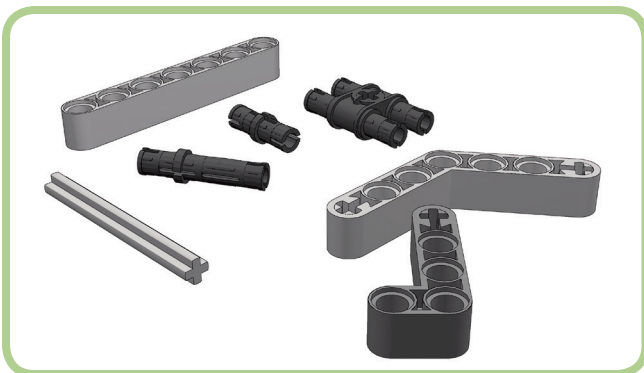


Abbildung 1-3: Balken und Pins

Der intelligente EV3-Stein bildet das Gehirn deines Roboters. Es handelt sich dabei praktisch um einen kleinen Computer, den du so programmieren kannst, dass sich deine Modelle bewegen. Statt eines ausgewachsenen Monitors und einer Tastatur bringt der Stein einen kleinen LCD-Bildschirm, einige Tasten und Anschlüsse für Motoren und Sensoren mit. Du kannst die Programme direkt auf dem EV3-Stein erstellen («On-Brick-Programmierung»), sie aber auch in der EV3-Software für Windows oder OS X schreiben und dann auf den EV3-Stein herunterladen. Bei der Ausführung eines solchen Programms ruft der EV3-Stein Daten von den Sensoren ab und bewegt die Motoren gemäß den Anweisungen, die du in dem Programm gegeben hast.

Mit den EV3-Motoren kannst du aus einem Lego-Modell einen beweglichen Roboter machen. Die beiden großen Motoren ermöglichen es, mobile Roboter zu bauen, die sich auf Rädern oder Raupenketten bewegen. Außerdem kannst du mithilfe dieser Motoren und des dritten, kleineren Motors auch Roboterhände, Kräne, Katalpulte und andere Vorrichtungen konstruieren. Bei vielen Robotern dienen zwei Motoren zur Fortbewegung, während der dritte eine andere Funktion ausführt. Es gibt aber auch Modelle, die sich gar nicht bewegen, sondern alle drei Motoren für andere Aufgaben einsetzen.

Die EV3-Sensoren ermöglichen es dem Roboter, entsprechend deinen Befehlen auf seine Umgebung zu reagieren. Im EV3-System gibt es Ultraschall-, Infrarot-, Berührungs-, Farb-, Kreis- und Motorumdrehungssensoren. Die Umdrehungssensoren sind in die EV3-Motoren eingebaut, alle anderen sind eigene Bauteile. Die einzelnen Sensoren weisen folgende Funktionen auf:

- **Ultraschallsensor:** Misst den Abstand zu einem Objekt oder einem Hindernis. Er kann auch die Anwesenheit eines anderen Ultraschallsensors erkennen.
- **Infrarotsensor:** Misst den Abstand zu einem Objekt oder einem Hindernis. Er kann auch den Abstand und die Richtung zur Infrarotfernsteuerung erkennen und reagiert auf Betätigung der Tasten auf der Fernsteuerung.
- **Berührungssensor:** Erkennt die Betätigung des Druckschalters an der Vorderseite des Sensors. Damit kann der Roboter feststellen, wenn er gegen ein Hindernis läuft oder wenn ein Objekt ihn berührt.
- **Farbsensor:** Erkennt die Farbe eines Objekts oder die Helligkeit des Lichts, das vorn auf den Sensor trifft. Dieser Sensor verfügt an der Vorderseite über eine kleine Leuchte, sodass er sowohl das Umgebungs- als auch das reflektierte Licht messen kann.
- **Kreiselsensor (Gyrosensor):** Misst Drehbewegungen. Dabei kann er sowohl die Drehzahl als auch den Winkel ermitteln, um den sich der Roboter dreht.
- **Motorumdrehungssensor:** Misst die Umdrehungen des Motors. Jeder EV3-Motor verfügt über einen eingebauten Umdrehungssensor.

Die Einzelhandelsversion enthält einen Berührungs-, einen Farb- und einen Infrarotsensor sowie die Infrarotfernsteuerung. In der Education-Version findest du zwei Berührungssensoren, einen Farb-, einen Ultraschall- und einen Kreiselsensor. Je nach Version hast du also entweder einen Ultraschall- oder einen Infrarotsensor zur Verfügung. Mit beiden ist es möglich, den Abstand des Roboters von einem Objekt zu messen. Bei den meisten Beispielprogrammen in diesem Buch kannst du sowohl den Ultraschall- als auch den Infrarotsensor verwenden.

Die Lego-Gruppe bietet zusätzlich (getrennt erhältliche) Temperatursensoren an. Auch andere Firmen stellen weitere Sensoren für das EV3-System her, z. B. HiTechnic, Vernier, Dexter Industries und Mindsensors. Erhältlich sind von ihnen u. a. ein Kompass, ein Beschleunigungs- und ein Luftdrucksensor.

Die Lego-Mindstorms-EV3-Software

Die EV3-Software ist eine grafische Programmierumgebung mit sämtlichen Werkzeugen, die du brauchst, um Programme für EV3-Roboter zu erstellen. Diese Art von Anwendung wird oft als *integrierte Entwicklungsumgebung* bezeichnet (Integrated Development Environment, daher kurz IDE). Bei der EV3-IDE handelt es sich um eine *grafische* Programmierumgebung, da du das Programm aus farbigen Symbolen, sogenannten *Blöcken*, zusammenstellst. Es gibt Blöcke zur Steuerung der Motoren, zur Verwendung der Sensoren und für viele andere Aufgaben. Um ein Programm zu erstellen, verschiebst du Blöcke auf dem Bildschirm, verbindest sie und änderst ihre Einstellungen.

Die EV3-Software bietet einen bemerkenswerten Kompromiss zwischen einfacher Bedienung und Programmierleistung. Es ist damit sehr leicht, einfache Programme zu schreiben, und doch kannst du damit auch sehr komplizierte Programme erstellen. Einige der erweiterten Funktionen mögen zu Anfang etwas schwer verständlich sein, aber mit ein wenig Übung werden sie dir schon einleuchten.

Software, Firmware und Hardware

Dein Programm ist eines der drei Elemente, die zusammen dafür sorgen, dass du deinen Roboter steuern kannst: Es stellt die sogenannte *Software* dar, also eine Zusammenstellung von Anweisungen, die ein Computer ausführen kann. Der Computer ist in diesem Fall der EV3-Stein. Das Wörtchen »soft«, also »weich«, in der Bezeichnung »Software« rührt daher, dass sich Änderungen leicht durchführen lassen. Diese Formbarkeit macht es möglich, nur mit dem EV3-Stein, drei Motoren und einigen wenigen Sensoren eine unendliche Vielzahl von Programmen zu erstellen.

Blöcke in einem grafischen Programmierbereich anzuordnen, ist für uns Menschen eine bequeme Möglichkeit, um Programme zu erstellen, aber um das Programm ausführen zu können, braucht der EV3-Stein etwas anderes. Dein Programm ist der sogenannte *Quellcode*, der erst in einen Satz von Anweisungen übersetzt werden muss, die der EV3-Stein verstehen kann. Anschließend müssen diese Anweisungen von deinem Computer auf den Stein kopiert werden. Nachdem das Programm übersetzt und heruntergeladen ist, kannst du es ausführen.

Auf dem Stein selbst läuft außerdem ein Programm, das als *Firmware* bezeichnet wird und sich nur selten ändert. Im Grunde genommen gehört es mit zu dem Gerät. Die EV3-Firmware ist etwa mit dem Betriebssystem eines Computers oder eines Smartphones wie Windows, iOS, Linux oder Android zu vergleichen. Sie ist das Programm, das einen Klang abspielt, wenn du den Stein einschaltest, das die Anzeige steuert und auf die Betätigung der Tasten auf dem Stein reagiert. Wenn du den EV3-Stein mit einem Computer verbindest, kommuniziert die Mindstorms-Umgebung mit der Firmware des Steins.

HINWEIS **Hin und wieder aktualisiert Lego die Firmware, um neue Funktionen hinzuzufügen oder Probleme zu beheben. Wenn dein Computer mit dem Internet verbunden ist, sucht die Mindstorms-Anwendung nach Aktualisierungen und fordert dich bei Bedarf auf, sie herunterzuladen.**

Der EV3-Stein ist die *Hardware*, auf der deine Programme laufen. Der Begriff »Hardware« bezeichnet die physischen Teile eines Computers. Dazu gehören der EV3-Stein, die Motoren, die Sensoren und die Lego-Bauelemente. Die Hardware ändert sich nicht. Du kannst sie umbauen und sogar auf verschiedene Weise einsetzen, aber die Fähigkeiten der einzelnen Teile bleiben gleich.

Kunst und Wissenschaft

Der faszinierendste Aspekt beim Bau eines Roboters besteht für mich darin, das Programm zu schreiben, das ihn zum Leben erweckt. Computerprogrammierung ist eine Kunst und eine Wissenschaft zugleich. Wir wenden *wissenschaftliche* Prinzipien an, wenn wir einer Reihe logischer Schritte folgen, um ein praktisches Problem zu lösen. Im weiteren Verlauf dieses Buches – und vor allem bei den längeren Programmen am Ende – lernst du wissenschaftliche Prinzipien und Programmier Techniken kennen, die dir helfen, bessere Programme zu schreiben (und einige geläufige schlechte Gewohnheiten abzulegen). Der grundlegende Vorgang beim Schreiben eines Programms, das ein bestimmtes Problem lösen soll, ist jedoch eher eine Kunst als eine Wissenschaft. Beim Programmieren gehst du meistens nicht Schritt für Schritt nach einem festen Schema vor, sondern brauchst eine Menge Kreativität und Einfallsreichtum. Dieses schöpferische Nachdenken ist es, was mir beim Programmieren so viel Spaß macht.

Programmieren kann jedoch auch enttäuschend und entmutigend sein, wenn etwas nicht so läuft, wie du es dir vorstellst. Die Gründe dafür, dass ein Programm nicht funktioniert, können manchmal ziemlich rätselhaft sein. In diesem Buch zeige ich dir daher immer wieder, wie du Problemen mit deinen Programmen auf den Grund gehst und sie behebst. Denke dabei aber immer daran, dass auch die Lösung eines Rätsels Spaß machen soll!

Merkmale von guten Programmen

Viele der Entscheidungen, die du beim Schreiben deiner Programme triffst, sind Geschmackssache. Im Laufe der Zeit wirst du deinen persönlichen Programmierstil entwickeln. Es bestehen fast immer mehrere korrekte Möglichkeiten, um ein Problem zu lösen. Allerdings gibt es auch drei Regeln, mit denen du die Qualität eines Programms bestimmen kannst. Gute Programme weisen folgende Eigenschaften auf:

1. Sie erfüllen die vorgesehene Funktion.
2. Sie lassen sich leicht ändern.
3. Sie sind für Personen, die mit der zum Schreiben des Programms benutzten Programmiersprache vertraut sind, gut verständlich.

Die erste Regel ist einleuchtend, aber es ist nicht ganz so einfach, wie es scheint. Bevor du sicher sein kannst, dass ein Programm funktioniert, musst du erst die *Anforderungen* dafür kennen – also die vollständige Beschreibung dessen, was es leisten soll. Wenn du ein Programm für ein Schulprojekt oder eine FLL-Aufgabe erstellst, werden dir die Anforderungen meistens vor Beginn ausgehändigt. Wenn du einen Roboter nur zum eigenen Vergnügen baust, kannst du die Anforderungen im Verlauf des Baufortschritts aufstellen. In jedem Fall musst du genau wissen, was der betreffende Roboter tun soll, bevor du beurteilen kannst, ob du Erfolg gehabt hast.

Die zweite Regel gibt es, da sich Anforderungen nach Beginn der Programmierung oftmals ändern. Das kann daran liegen, dass sich ein Problem nicht auf die Weise lösen lässt, die du zuerst im Sinn gehabt hast, oder dass du die Anforderungen erweiterst, um ein schwierigeres Problem zu lösen. Daher ist es gut, wenn sich das Programm leicht ändern lässt, um es an die neuen Anforderungen anzupassen. Programme, die sich leicht abwandeln lassen, können auch einfacher wiederverwendet werden, um ähnliche Probleme zu lösen. Es kann viel Zeit sparen, vorhandene Programme wiederzuverwenden, anstatt ein von Grund auf neues Programm zu schreiben.

Bei der dritten Regel geht es darum, deine Programme so einfach und verständlich zu halten wie möglich. Programme, die komplizierter sind als nötig, weisen meistens mehr Fehler auf und lassen sich nicht so gut wiederverwenden. Um dein Programm leicht verständlich zu machen, kannst du Kommentare einfügen, die erklären, wie es funktioniert. Sinnvolle Kommentare bieten eine einfache Möglichkeit, um deine Programme auch für andere Programmierer nutzbar zu machen.

Was du in diesem Buch lernst

Um ein guter Programmierer zu werden, brauchst du Wissen und Übung. In diesem Buch konzentriere ich mich auf drei Wissensgebiete, die sehr wichtig sind, um als EV3-Programmierer erfolgreich zu sein:

- **Das Verhalten der einzelnen Blöcke:** Um einen Block in einem Programm verwenden zu können, musst du zunächst einmal wissen, wie er funktioniert. Es gibt zwar viele Blöcke, die jeweils über verschiedene Optionen verfügen, doch ist es nicht weiter schwierig, die einzelnen Blöcke kennenzulernen. Die EV3-Hilfdatei gibt eine umfassende Beschreibung der verschiedenen Blöcke, und es ist auch sehr einfach (und macht Spaß!), kleine Testprogramme zu schreiben, mit denen du herausfinden kannst, was die einzelnen Blöcke jeweils tun können.
- **Mehrere Blöcke zu einem funktionierenden Programm zusammenschließen:** Dazu musst du etwas über den Programmablauf, über Datenleitungen und Variablen lernen. Hier wird es etwas komplizierter, aber Einzelheiten über die Funktionsweise von EV3-Programmen zu lernen, wird dir helfen, Irritationen zu vermeiden, die sich bei vielen Benutzern einstellen, wenn sie über einfache Programme hinausgehen.
- **Allgemeine Programmiertechniken:** Die ersten Beispiele hierfür sind die zuvor angegebenen drei Regeln. Im weiteren Verlauf werde ich dir noch weitere Techniken vorstellen, die unabhängig von der verwendeten Programmiersprache und der Art des Programms sehr nützlich sind.

Programmierung gehört zu den Tätigkeiten, die man am besten dadurch lernt, dass man sie tut. Damit sind wir auch schon beim Thema »Übung« angelangt. Viele der Programmiertechniken, die du lernen musst, werden dir erst dann richtig klar, wenn du sie anwendest. Je mehr Programme du schreibst, umso sicherer wirst du dabei.

Die Online-Community zu Lego Mindstorms

Es gibt eine blühende Online-Community zum Thema Lego-Robotik, darunter auch Websites, die Hunderte von innovativen Roboterkonstruktionen vorstellen. Insbesondere die Website mindBOARDS (<https://www.mindboards.net/>) ist für ihr Forum berühmt, in dem die Benutzer Ideen austauschen und Antworten auf ihre Fragen finden können. Das sind hervorragende Informationsquellen, wenn du nicht herausfinden kannst, warum dein Roboter nicht so funktioniert, wie er es deiner Meinung nach tun sollte. Eine Suche im Forum fördert oft schnell die Antwort zutage, die du brauchst. Wenn noch niemand eine Lösung zu deinem Problem eingestellt hat, kannst du selbst eine Frage stellen, in der du deine Schwierigkeiten beschreibst. Eine Liste nützlicher Websites rund um Mindstorms findest du in Anhang B.

Wie geht es weiter?

Im nächsten Kapitel stelle ich die EV3-Programmierungsumgebung vor und zeige dir anhand einiger einfacher Programmier- und Beispielprogramme, wie du sie verwendest. In den darauf folgenden Kapiteln führe ich nach und nach weitere Blöcke und Programmier- und Programmier-Techniken ein, mit denen du immer anspruchsvollere Programme schreiben kannst.

Den Quellcode für alle Programme in diesem Buch kannst du von <http://www.nostarch.com/ev3programming.htm> herunterladen.