

Programmieren lernen mit EV3

Vom Einsteiger zum Meisterprogrammierer mit LEGO® Mindstorms® EV3

Bearbeitet von
Terry Griffin

1. Auflage 2015. Taschenbuch. XXIV, 264 S. Paperback

ISBN 978 3 86490 275 8

Format (B x L): 20,3 x 25,4 cm

[Weitere Fachgebiete > Sport, Tourismus, Freizeit > Freizeit & Lifestyle: Allgemeines > Hobbies & Spiele](#)

Zu [Inhaltsverzeichnis](#)

schnell und portofrei erhältlich bei

The logo for beck-shop.de features the text 'beck-shop.de' in a bold, red, sans-serif font. Above the 'i' in 'shop' are three red dots of increasing size. Below the main text, the words 'DIE FACHBUCHHANDLUNG' are written in a smaller, red, all-caps, sans-serif font.

beck-shop.de
DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung beck-shop.de ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

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!

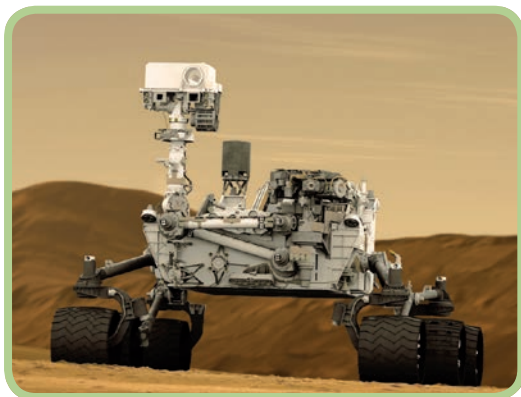


Abbildung 1-1: Mars-Forschungsrover
(mit freundlicher Genehmigung von NASA/JPL-Caltech)

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.

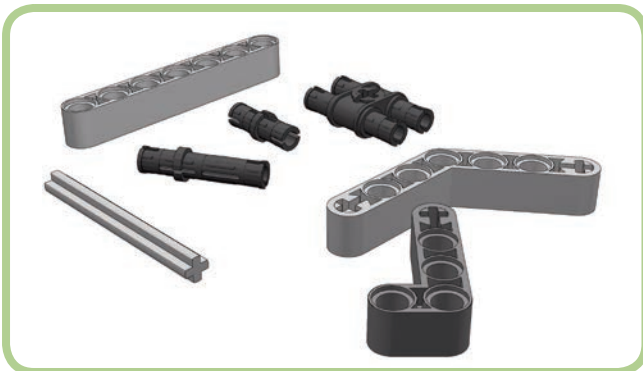


Abbildung 1-3: Balken und Pins

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-, Kreisel- 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 Kreisel sensor. 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 Programmiertechniken und Beispielprogramme, wie du sie verwendest. In den darauf folgenden Kapiteln führe ich nach und nach weitere Blöcke und Programmiertechniken 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.