



Leseprobe

Andreas Stadler

Mein LEGO®-EV3-Buch

Eigene Roboter bauen und programmieren mit LEGO® MINDSTORMS®

ISBN (Buch): 978-3-446-44737-0

ISBN (E-Book): 978-3-446-44900-8

Weitere Informationen oder Bestellungen unter

<http://www.hanser-fachbuch.de/978-3-446-44737-0>

sowie im Buchhandel.

# 2

## Aufgaben zur Rubrik Aktion – die Aktoren im Einsatz

In der Informatik und der Robotik ist das EVA-Prinzip von grundlegender Bedeutung. Informationen werden über die Tastatur oder die Maus in den Computer eingegeben, im Prozessor des Rechners verarbeitet und über den Bildschirm und den Lautsprecher ausgegeben.

Auch der EV3-Stein ist ein kombiniertes Eingabe-, Verarbeitungs-, Ausgabe-Gerät. So können beispielsweise über die Stein-Tasten Eingaben getätigt werden, die im EV3-Stein verarbeitet und über das Display (die Anzeige), den Lautsprecher (den Klang) oder die Stein-Statusleuchte wieder ausgegeben werden. Zudem stehen Eingabe- und Ausgabe-Ports zur Verfügung, an welche externe Ein- und Ausgabegeräte angeschlossen werden können. Bei den Eingabe-Ports handelt es sich um die Ports 1 bis 4, bei den Ausgabe-Ports um die Ports A bis D. Auch über einen USB-Port und einen SD-Kartenleser können Ein- und Ausgaben getätigt werden.

An die Ausgabe-Ports des EV3-Steins wird üblicherweise die im Standardumfang des EV3-Kastens enthaltene Hardware, der mittlere und der große Motor, angeschlossen.

In der LEGO® MINDSTORMS® EV3-Software findest du die Arbeitsoberfläche für das grafische Programmieren unter *Hauptmenü/Datei/Neues Projekt (/Programm)*. In der LEGO® MINDSTORMS® Education EV3-Software kannst du ein neues Projekt zudem über die *Lobby* öffnen, wenn du dem Pfad *Lobby/Datei/Neues Projekt/Programm* folgst und abschließend *Öffnen* klickst.

Auf der Arbeitsoberfläche findest du nun in der Rubrik **Aktion** die Programmierblöcke mit Modi und Parametern für die Ausgaben (grüner Reiter im Menü). Um die Rubrik **Aktion** soll es in diesem Kapitel gehen.

Die Rubrik **Aktion** besteht aus folgenden Programmierblöcken (von links nach rechts in der folgenden Abbildung dargestellt):

- Mittlerer Motor
- Großer Motor
- Standardsteuerung (Bewegungslenkung)
- Hebellenkung

- Anzeige
- Klang
- Stein-Statusleuchte



### Hinweis

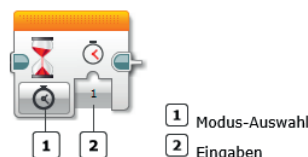
Um ein Programm starten und die zeitliche Abfolge des Informationsflusses in einem Programm steuern zu können, sollen auch Eingaben über die Blöcke *Start* und *Warten* aus der Rubrik **Ablauf-Regelung** (oranger Reiter im Menü) verwendet werden. Dazu musst du diese Blöcke, wie auch alle anderen Blöcke, mit der linken Maustaste aus dem Menü auf die Arbeitsoberfläche ziehen.

Durch Klicken auf den grünen Pfeil des *Start*-Blocks wird dann die ihm folgende Programmierblock-Sequenz abgearbeitet und ein Roboter, der über USB-Kabel (oder drahtlos über Bluetooth bzw. WLAN) am Computer angeschlossen ist, wird in Aktion versetzt. Hierfür wird das Programm zuerst kompiliert, dann auf den EV3-Stein heruntergeladen und letztendlich ausgeführt.



Enthält ein Programm mehr als eine Programmierblock-Sequenz, dann sind die entsprechenden *Start*-Blöcke separat zu starten. Alternativ kann man Programme nach einem einzigen *Start*-Block auch in Parallelreihen verzweigen lassen. Nähere Informationen hierzu findest du in der LEGO® MINDSTORMS® EV3-Software unter *Hauptmenü/Hilfe/EV3-Hilfe einblenden/Allgemein/Programme* oder in Abschnitt 3.10.

Wird der *Warten*-Block in eine Programmierblock-Sequenz eingebunden, dann ist im Modus *Zeit* der Eingabe-Parameter *Sekunden* auf den entsprechenden Zahlenwert zu setzen. Ist der *Warten*-Block während der Abarbeitung der Programmierblock-Sequenz aktiv, blinkt er und setzt das Programm erst dann wieder fort, wenn die zu wartenden Sekunden verstrichen sind.



## ■ 2.1 Mittlerer Motor – die Steuerung über einen oder mehrere Programmblöcke

### Erfassen der Aufgabenstellung

Ist es möglich, die Aktion eines mittleren Motors (Anschalten, Drehen, Ausschalten) sowohl mit einem Programm, bestehend aus drei Blöcken, als auch mit nur einem einzigen Programmblock vollständig zu beschreiben?

Wie können Umfang und Geschwindigkeit einer Rotation beeinflusst werden?

Lassen sich ganz allgemein mehrere Programmblöcke zu einem einzigen Programmblock – einem Unterprogramm, einer Prozedur oder hier einem **Eigenen Block** – zusammenstellen?

### Erschließen des Themas

Baue einen einfachen Roboter, indem du den mittleren Motor über ein Verbindungskabel am EV3-Stein montierst. Stelle die USB-Verbindung zwischen EV3-Stein und Rechner sowie die Stromversorgung des EV3-Steins über das Netzteil sicher.

- a) Schreibe ein Programm, mit dem der mittlere Motor einerseits unter Verwendung von drei Blöcken (zwei Blöcke aus der Rubrik **Aktion** und ein Block aus der Rubrik **Ablauf-Regelung**) mit 50% der Maximalgeschwindigkeit für 2 s läuft und dann gestoppt wird und andererseits mit nur einem Block (aus der Rubrik **Aktion**) dieselbe Funktion erbringt.
- b) Wie kann man den mittleren Motor mit langsamer Geschwindigkeit (25) eine 180°-Drehung und dann mit schneller Geschwindigkeit (75) eine halbe Umdrehung in entgegengesetzter Richtung durchführen lassen?
- c) Wie könnte man nun das erstellte Programm, welches aus drei Blöcken besteht, auf ganz allgemeine Weise zu einem einzigen **Eigenen Block** (= einem Unterprogramm) zusammenfassen, der analog zu dem aus drei Blöcken bestehenden Programm verwendet werden kann?

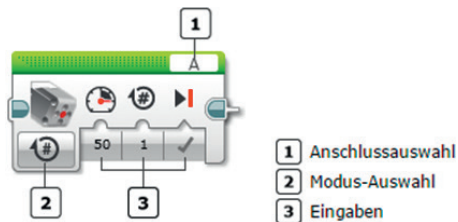
### Stoffsammlung

**Hardware:** Verbinde den mittleren Motor über eine Verbindungsleitung mit dem Port A des EV3-Steins, da im entsprechenden Software-Block standardgemäß dieser Port voreingestellt ist. Nun musst du den EV3-Stein noch über eine USB-Leitung mit dem Computer verbinden.

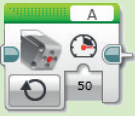
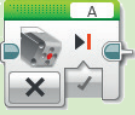
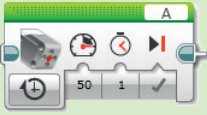




Photo courtesy of the LEGO Group.  
© 2015 The LEGO Group

**Software:** Wähle *Hauptmenü/Datei/Neues Projekt* in der LEGO® MINDSTORMS® EV3-Software aus, um ein neues Programm zu öffnen. In der Rubrik **Aktion** findest du den Block für den mittleren Motor.



Mit der *Anschlussauswahl* kann der Port gewählt werden, über welchen der mittlere Motor mit der Software kommunizieren soll; üblicherweise ist dies Port A (siehe auch *Hauptmenü/Hilfe/EV3-Hilfe einblenden/Allgemein/Anschlussauswahl*). Über die *Modus-Auswahl* kann der Modus festgelegt werden, in welchem der mittlere Motor betrieben werden soll. Entsprechend des gewählten Modus kann ein Satz an *Eingaben* getätigt werden.

Modus	Funktion
<b>An</b> 	Im Modus <i>An</i> wird der Motor angeschaltet. Über den einzigen Eingabe-Parameter <i>Leistung</i> (-100, ..., 50, ..., 100) können die Drehgeschwindigkeit und die Drehrichtung (+/-) eingestellt werden. Anschließend wird sofort der nächste Block im Programm abgearbeitet. Der Motor läuft, bis er von einem späteren Block gestoppt wird, bis er anderweitig angesteuert wird oder bis das Programm beendet wird.
<b>Aus</b> 	Im Modus <i>Aus</i> wird ein bereits angeschalteter Motor wieder ausgeschaltet. Wenn der einzige Parameter <i>Am Ende bremsen</i> auf <i>Wahr</i> gesetzt ist, wird der Motor sofort gestoppt. Ist er auf <i>Falsch</i> gesetzt, dann wird lediglich die Stromversorgung des Motors abgeschaltet und der Motor läuft aus.
<b>An für n Sekunden</b> 	Im Modus <i>An für n Sekunden</i> wird der Motor für die im Parameter <i>Sekunden</i> eingestellte Dauer in Sekunden (Kommazahlen für Sekundenbruchteile) angeschaltet. Die Parameter <i>Leistung</i> und <i>Am Ende bremsen</i> werden wie bereits beschrieben verwendet.
<b>An für n Umdrehungen</b> 	Im Modus <i>An für n Umdrehungen</i> wird der Motor für die im Parameter <i>Umdrehungen</i> eingestellte Anzahl an Umdrehungen (Kommazahlen für Bruchteile davon) eingeschaltet. Die Parameter <i>Leistung</i> und <i>Am Ende bremsen</i> werden wie bereits beschrieben verwendet.
<b>An für n Grad</b> 	Im Modus <i>An für n Grad</i> wird der Motor für die im Parameter <i>Gradzahl</i> eingestellte Gradzahl (Kommazahlen für Bruchteile von Winkelgraden) angeschaltet. Die Parameter <i>Leistung</i> und <i>Am Ende bremsen</i> werden wie bereits beschrieben verwendet.

Rotationsbewegungen lassen sich über die Modi *An für n Sekunden*, *An für n Umdrehungen* und *An für n Grad* einstellen, wobei die Rotationsbewegung im Modus *An für n Grad* am exaktesten definiert werden kann.

Das mit einer Motorumdrehung zurückgelegte Bogenmaß ist in den Modi *An für n Umdrehungen* und *An für n Grad* über die einzugebenden Parameter bereits vollständig festgelegt (1 Umdrehung =  $360^\circ = 2\pi$ ). Im Modus *An für n Sekunden* ist hierfür noch die Geschwindigkeit (Parameter *Leistung*) zu berücksichtigen.

In Abhängigkeit vom Vorzeichen der eingestellten Leistung ist die Drehrichtung des mittleren Motors wie in der folgenden Abbildung festgelegt.



## Lösungshinweis

Eine Lösung wäre, nach dem *Start-Block* (Rubrik **Ablauf-Regelung**) den Block *Mittlerer Motor* (Rubrik **Aktion**) über das Modul *An* einzuschalten und seine Geschwindigkeit über den Parameter *Leistung* auf 50 zu setzen. Um die Dauer dieser Motorbewegung auf 2 s zu begrenzen, ist im Block *Warten* (Rubrik **Ablauf-Regelung**) das Modul auf *Zeit* und der Parameter *Sekunden* auf 2 zu setzen. Abschließend ist der Motor über den Block *Mittlerer Motor* (Rubrik **Aktion**) mit dem Modul *Aus* wieder auszuschalten. Hierbei kann für den Parameter *Am Ende bremsen* einer der beiden logischen Werte *Wahr* oder *Falsch* gewählt werden. Im ersten Fall bremst der Roboter abrupt ab, im zweiten etwas weniger schnell.

Eine andere Lösung wäre, nach dem *Start-Block* (Rubrik **Ablauf-Regelung**) den Block *Mittlerer Motor* (Rubrik **Aktion**) über das Modul *An für n Sekunden* einzuschalten. Damit stehen nun die Parameter *Leistung*, *Sekunden* und *Am Ende bremsen* zur Auswahl, welche auf 50, 2 und beispielsweise *Wahr* zu setzen sind. So werden in dieser zweiten Lösungsmöglichkeit auf effiziente Art und Weise alle drei Parameter des Programms innerhalb eines Blocks gesetzt.

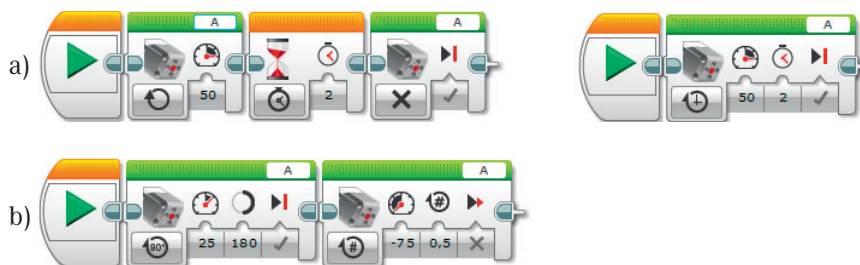
Ganz ähnlich ist der zweite Aufgabenteil zu lösen. Hier verwendest du nach dem *Start-Block* (Rubrik **Ablauf-Regelung**) zuerst den Block *Mittlerer Motor* (Rubrik **Aktion**) mit dem Modul *An für n Grad* und dann einen entsprechenden Block mit *An für n Umdrehungen*. Parameter sind dann (wieder) die *Leistung*, die *Gradzahl* oder die Anzahl der *Umdrehungen* sowie *Am Ende bremsen*, welche entsprechend der Aufgabenstellung zu setzen sind.

Um ein Unterprogramm aus einem bereits geschriebenen Programm zu erstellen, wählst du *Hauptmenü/Werkzeuge/Eigene Blöcke erstellen* aus und folgst der Menüführung.

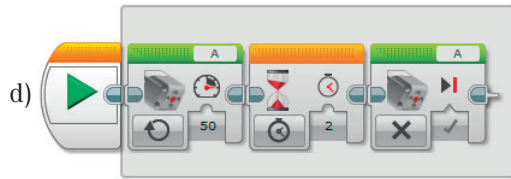
## Ausarbeitung

Sowohl die Blöcke selbst als auch deren Tasten und Felder für die Modi, Eingabe- und Ausgabe-Parameter besitzen Symbole, welche einen Block und dessen Einstellungen eindeutig identifizieren. So kannst du ein Programm, welches mit einer grafischen Programmiersprache erstellt worden ist, wie ein Bilderbuch lesen, das eine ihm eigene Grammatik besitzt.

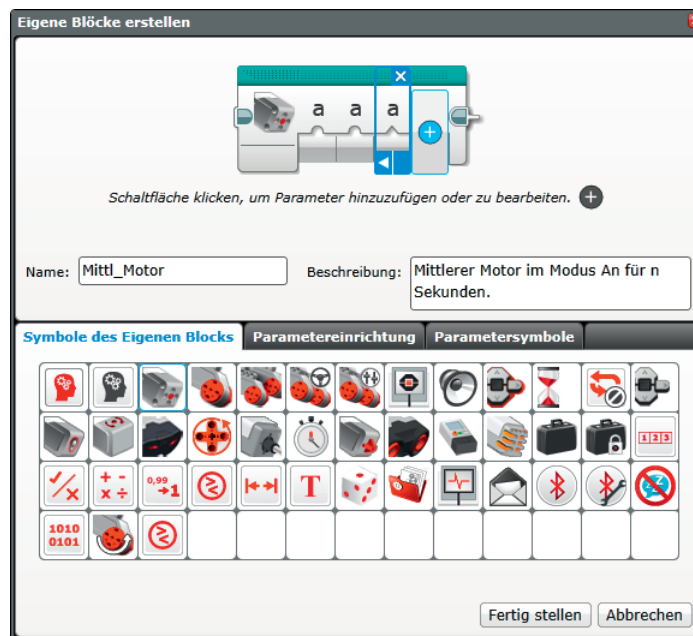
Im Folgenden findest du die ausgearbeiteten Lösungen dieser Aufgabenstellung.



- c) Zur Erstellung von Unterprogrammen markierst du die entsprechenden Blöcke (Auswahl-Werkzeug).

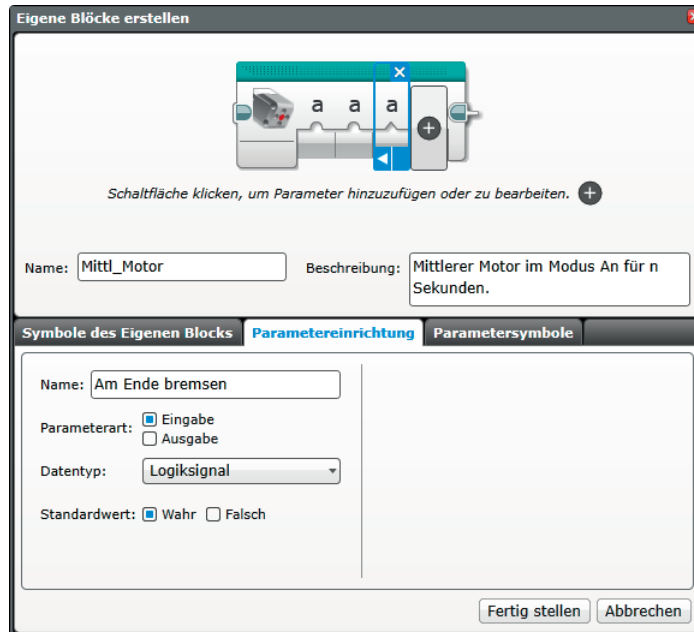


Anschließend wählst du *Hauptmenü/Werkzeuge/Eigene Blöcke erstellen* aus und gibst im sich öffnenden Fenster den *Namen* des eigenen Blocks (Unterprogramms), eine kurze *Beschreibung* und vielsagende *Symbole des Eigenen Blocks* an.

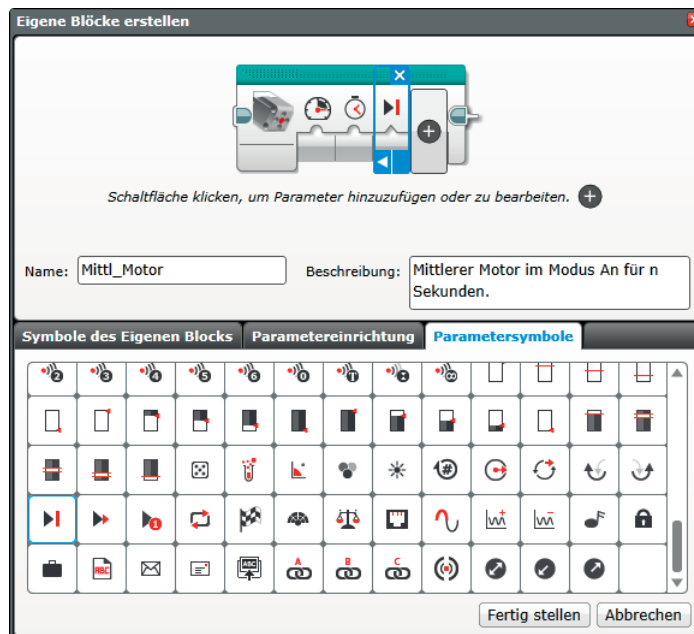


Unter dem Reiter *Parametereinstellung* können *Name*, *Parameterart*, *Datentyp* und *Standardwert* festgelegt werden.



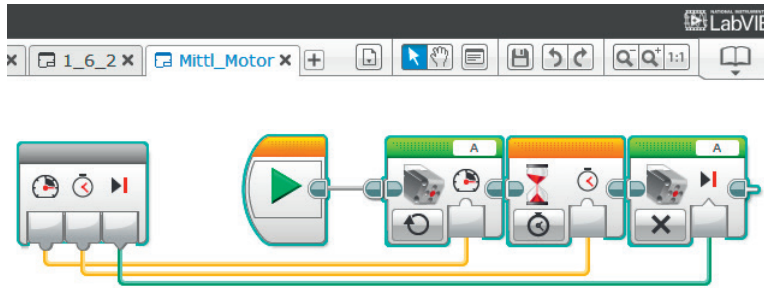


Unter dem Reiter *Parametersymbole* können diesen Parametern nun auch die entsprechenden Symbole zugewiesen werden.



Klicke abschließend auf *Fertig stellen*.

Für dieses Unterprogramm wird nun ein eigenes Programmblatt erstellt, in welchem die globalen Parameter des neuen eigenen Blocks (graue Parameterbox) über Ziehen von Datenleitungen noch den lokalen Parametern der einzelnen Teilblöcke zuzuordnen sind. Damit können die Parameter am eigenen Block dann beliebig eingestellt werden (siehe Abbildung).



Auch in der Rubrik **Eigene Blöcke** wird automatisch der neue Block *Mittl\_Motor* abgelegt.



Nun kann dieser Block auch im Programm verwendet werden, wobei beide der in der folgenden Abbildung dargestellten Programme (Blöcke) den gleichen Funktionsumfang haben.



#### Wusstest du, dass ...?

Blöcke können nicht nur selbst erzeugt werden, sondern auch über das Internet importiert werden. Nähere Informationen hierzu findest du unter *Hauptmenü / Hilfe / EV3-Hilfe einblenden / Werkzeuge / Block-Import*.

## ■ 2.2 Großer Motor – der Einsatz verschiedener Modi für eine Umdrehung

### Erfassen der Aufgabenstellung

Ist es möglich, mit lediglich einem Typ von Block auf drei unterschiedlichen Weisen exakt die gleiche Aktion in einem Programm zu motivieren?

### Erschließen des Themas

Schließe an den Port D des EV3-Steins über ein Verbindungskabel einen großen Motor an. Verlege das USB-Kabel zwischen EV3-Stein und Rechner und stelle die Stromversorgung für den EV3-Stein sicher.

Schreibe ein Programm, mit dem der große Motor dreimal gleich schnell exakt eine Umdrehung macht, getrennt von jeweils einer Sekunde Pause. Wähle für jede Umdrehung einen anderen Modus und passe die entsprechenden Parameter an.

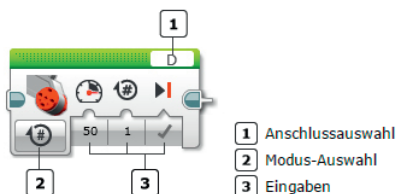
### Stoffsammlung

**Hardware:** Der große Motor ist etwas leistungsstärker als der mittlere Motor. Schließe ihn über ein Verbindungskabel an den Port D des EV3-Steins an und verbinde den EV3-Stein mit dem Rechner.

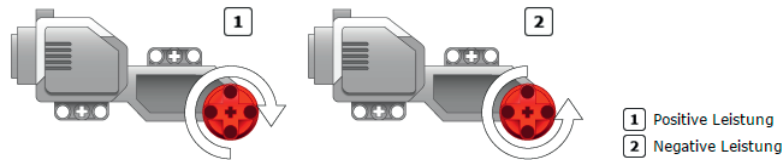


Photo courtesy of the LEGO Group.  
© 2015 The LEGO Group

**Software:** In der EV3-Software findest du unter *Hauptmenü/Hilfe/EV3-Hilfe einblenden/Programmierblöcke/Aktions-Blöcke/Großer Motor* Informationen zu den Software-Optionen für den großen Motor.



Verschafe dir einen Überblick über die Möglichkeiten, den großen Motor anzusteuern. Vergleiche diese Möglichkeiten mit den Ausführungen zum mittleren Motor. Beachte hierzu, dass die Drehrichtung des großen Motors, in Abhängigkeit vom Vorzeichen der eingestellten Leistung, wie in der folgenden Abbildung festgelegt ist.

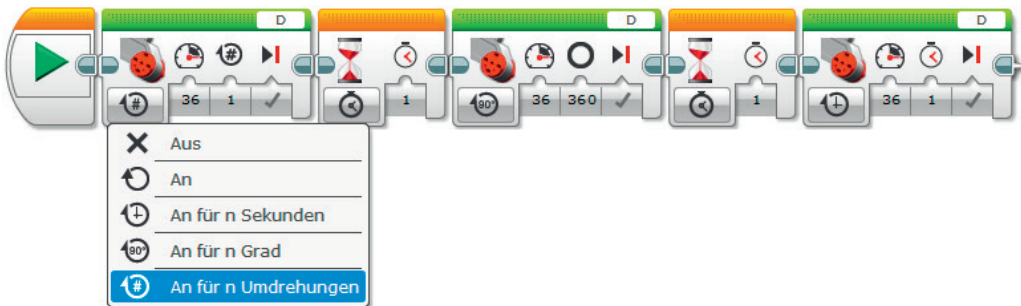


### Lösungshinweis

Auf den *Start*-Block (Rubrik **Ablauf-Regelung**) folgen die drei Blöcke des Typs *Großer Motor* (Rubrik **Aktion**), die über je einen *Warten*-Block (Rubrik **Ablauf-Regelung**) voneinander getrennt sind.

Die Blöcke des Typs *Großer Motor* weisen die Modi *An für n Umdrehungen*, *An für n Grad* bzw. *An für n Sekunden* auf. In diesen Blöcken sind nun die Parameter entsprechend der Aufgabenstellung so zu setzen, dass sich der Motor gleich schnell eine Umdrehung (d. h. 360°) lang dreht. Die im Modus *An für n Sekunden* einzustellende Zeit ist natürlich von der gewählten Geschwindigkeit (Leistung) abhängig.

### Ausarbeitung



### Verständnisfragen

Welchen der hier bereits behandelten drei Modi würdest Du für die Programmierung eines Uhrwerks verwenden? Und welchen für die Programmierung eines Ventilators? Wie begründest du deine Wahl?

Welche Möglichkeiten gibt es, die Geschwindigkeit eines Motors mit Hard- und Softwarelösungen herab- oder heraufzusetzen?

## Antworten auf die Verständnisfragen

Uhrwerke laufen sehr langsam. Der Sekundenzeiger bewegt sich  $6^\circ$  in einer Sekunde. Deshalb wählt man hier den Modus *An für n Grad*. Ventilatoren hingegen laufen sehr schnell und nahezu ununterbrochen, deshalb würde man hier vorzugsweise große Zeitspannen einstellen, d. h., den Modus *An für n Sekunden* wählen. Anwendungen, die bezüglich Bogenmaß oder Dauer zwischen den genannten Beispielen liegen, werden häufig im Modus *An für n Umdrehungen* realisiert.

Die feinste Einstellungsmöglichkeit für einen Drehwinkel bietet die Grad-Angabe. Pro Umdrehung werden bereits  $360^\circ$  überschritten, sodass diese Einstellung schon vergleichsweise grob ist. Mit steigender Geschwindigkeit (Leistung) nimmt während einer Zeitspanne die Anzahl an Umdrehungen zu, sodass ein gewünschter Winkel nur sehr grob eingestellt werden könnte.

Eine Hardwarelösung zum Verringern der Geschwindigkeit ist die Verwendung eines Getriebes mit Untersetzung; zum Erhöhen der Geschwindigkeit benötigt man eine Übersetzung (begrenzt durch die Motorleistung). Eine Softwarelösung zur Verringerung der Geschwindigkeit wäre das Zwischenschalten von entsprechend langen *Warten*-Blöcken (Rubrik **Ablauf-Regelung**) zwischen zwei kleinen Bewegungsintervallen. Softwarelösungen ermöglichen keine Erhöhung der Geschwindigkeit über die Maximalgeschwindigkeit (Leistung = 100%) hinaus.



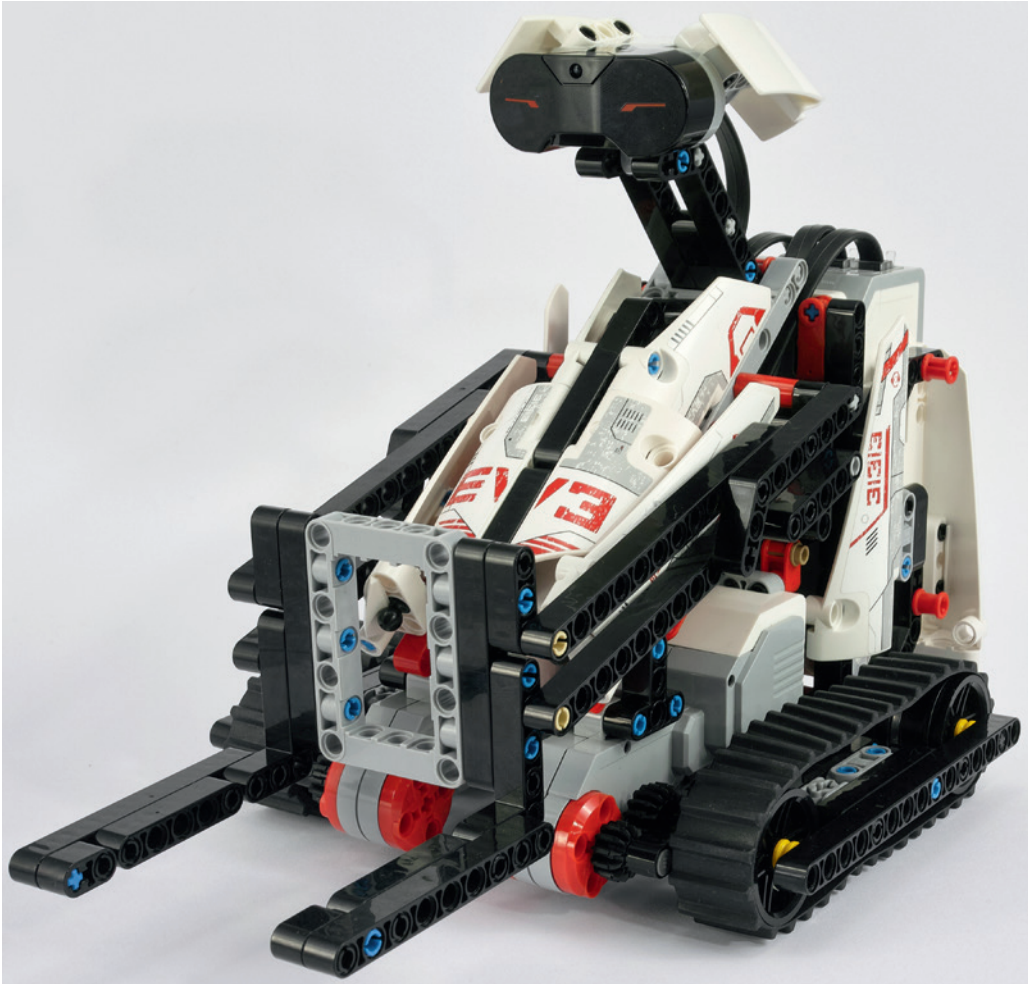
### Motivation

BOBB3E ist etwas stärker als andere Roboter – und das nicht nur, weil er ein Gabelstapler ist und große Lasten heben kann. Seine Antriebsketten werden auch über eine Untersetzung von den Motoren angetrieben (siehe vorderes Ende des Kettenantriebs). Damit wird er zwar langsamer, aber auch stärker.

Wie könntest du ihn schneller (und damit gleichzeitig auch schwächer) machen? Das funktioniert, indem du die beiden Zahnräder tauschst und so eine Übersetzung baust. Überdies könntest du auch ein ganzes Getriebe mit mehreren Gängen bauen (Hardware-Lösung), welches dann jedoch auch etwas mehr Platz benötigt als die beiden oben gezeigten Zahnräder.

Eine Bauanleitung für BOBB3E findest du unter <http://www.lego.com/de-de/mindstorms/build-a-robot/bobb3e>.

Programmiere nun seine beiden Laufketten so, dass er nicht nur mit einem Motor im Kreis fährt, sondern auch mit beiden Motoren geradeaus fahren kann – und dies mit unterschiedlichen Geschwindigkeiten (Software-Lösung). Bei der Lösung dieser Aufgabe hilft dir auch Abschnitt 2.3.



## ■ 2.3 Standardsteuerung (Bewegungslenkung) – einen Meter geradeaus fahren

### Erfassen der Aufgabenstellung

Baue einen Roboter, der programmgesteuert exakt einen Meter weit fährt.

### Erschließen des Themas

Baue entsprechend der Bauanleitung in Abschnitt 1.1.2 bzw. im *Robot Educator (Robot Educator/Bauanleitungen/Fahrgestell)* das Fahrgestell für einen Roboter. Stelle die Kommunikation zwischen Rechner und Roboter über die USB-Leitung sowie die Stromversorgung des EV3-Steins über das Netzteil sicher.

Programmiere nun das Roboter-Fahrgestell so, dass es mit der Standardsteuerung genau einen Meter weit fährt (der Reifenumfang beträgt 17,6 cm). Verwende hierzu alle drei geeigneten Modi.

Wie sähe eine Lösung mit der Hebelsteuerung aus?

### Stoffsammlung

**Hardware:** Schließe beim Fahrgestell die beiden großen Motoren über Verbindungsleitungen an die Ports B und C des EV3-Steins an, da im entsprechenden Software-Block (*Standardsteuerung*) diese beiden Ports standardgemäß gesetzt sind. Nun musst du den EV3-Stein noch über eine USB-Leitung mit dem Rechner verbinden.



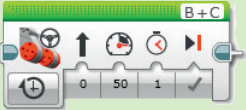
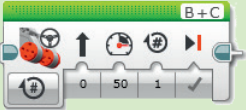



Photo courtesy of the LEGO Group.  
© 2015 The LEGO Group

**Software:** Wähle *Hauptmenü/Datei/Neues Projekt* in der LEGO® MINDSTORMS® EV3-Software aus, um ein neues *Programm* zu öffnen (in der Education-Version der Software kann auch *Lobby/Datei/Neues Projekt/Programm* und abschließend *Öffnen* gewählt werden). In der Rubrik **Aktion** findest du den Block für die Standardsteuerung.



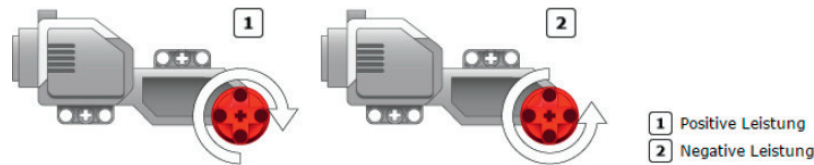


Modus	Funktion
<b>An</b> 	<p>Im Modus <i>An</i> wird die Standardsteuerung angeschaltet, welche aus den beiden großen Motoren besteht, die an Port B und C angeschlossen sind. Mit dem Eingabe-Parameter <i>Lenkung</i> kann über einen Schieberegler eine Links- oder Rechtskurve eingestellt werden (-100, ..., 0, ..., 100). Mit dem Eingabe-Parameter <i>Leistung</i> (-100, ..., 50, ..., 100) können die Drehgeschwindigkeit und die Drehrichtung (+/-) eingestellt werden. Anschließend wird sofort der nächste Block im Programm abgearbeitet. Der Motor läuft, bis er von einem späteren Block gestoppt wird, bis er anderweitig angesteuert wird oder bis das Programm beendet wird.</p>
<b>Aus</b> 	<p>Im Modus <i>Aus</i> wird eine bereits angeschaltete Standardsteuerung wieder ausgeschaltet. Wenn der einzige Parameter <i>Am Ende bremsen</i> auf <i>Wahr</i> gesetzt ist, wird der Motor sofort gestoppt. Ist er auf <i>Falsch</i> gesetzt, dann wird lediglich die Stromversorgung des Motors abgeschaltet und der Motor läuft aus.</p>
<b>An für n Sekunden</b> 	<p>Im Modus <i>An für n Sekunden</i> wird die Standardsteuerung für die im Parameter <i>Sekunden</i> eingestellte Dauer in Sekunden (Kommazahlen für Sekundenbruchteile) angeschaltet. Die Parameter <i>Lenkung</i>, <i>Leistung</i> und <i>Am Ende bremsen</i> werden wie bereits beschrieben verwendet.</p>
<b>An für n Umdrehungen</b> 	<p>Im Modus <i>An für n Umdrehungen</i> wird die Standardsteuerung für die im Parameter <i>Umdrehungen</i> eingestellte Anzahl an Umdrehungen (Kommazahlen für Bruchteile davon) eingeschaltet. Die Parameter <i>Lenkung</i>, <i>Leistung</i> und <i>Am Ende bremsen</i> werden wie bereits beschrieben verwendet.</p>
<b>An für n Grad</b> 	<p>Im Modus <i>An für n Grad</i> wird die Standardsteuerung für die im Parameter <i>Gradzahl</i> eingestellte Gradzahl (Kommazahlen für Bruchteile von Winkelgraden) angeschaltet. Die Parameter <i>Lenkung</i>, <i>Leistung</i> und <i>Am Ende bremsen</i> werden wie bereits beschrieben verwendet.</p>

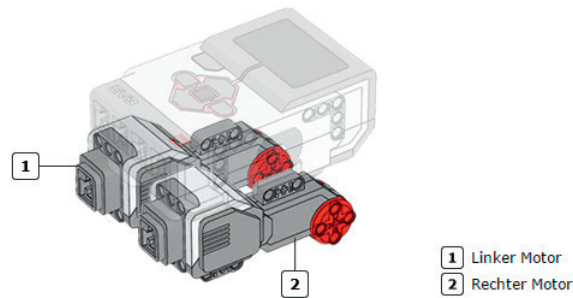
Rotationsbewegungen lassen sich über die Modi *An für n Sekunden*, *An für n Umdrehungen* und *An für n Grad* einstellen, wobei die Rotationsbewegung im Modus *An für n Grad* am exaktesten definiert werden kann.

Das mit einer Motorumdrehung zurückgelegte Bogenmaß ist in den Modi *An für n Umdrehungen* und *An für n Grad* über die einzugebenden Parameter bereits vollständig festgelegt (1 Umdrehung =  $360^\circ = 2\pi$ ). Im Modus *An für n Sekunden* ist hierfür noch die Geschwindigkeit (Parameter *Leistung*) zu berücksichtigen.

Die Drehrichtung der beiden großen Motoren ist, in Abhängigkeit vom Vorzeichen der eingestellten Leistung, wie in der folgenden Abbildung festgelegt.



Die Zuordnung, welcher Motor links und welcher rechts ist, wird wie in der folgenden Abbildung vorgenommen.



### Lösungshinweis

Sinnvollerweise berechnet man aus der zurückzulegenden Strecke und dem Reifenumfang die Anzahl der Umdrehungen, die ein Rad zu bewältigen hat.

Nach dem *Start-Block* (Rubrik **Ablauf-Regelung**) platziert man nun den Block *Standardsteuerung* (Rubrik **Aktion**) im Modul *An für n Umdrehungen* und gibt neben den laut Aufgabenstellung frei wählbaren Parametern auch den Parameter *Umdrehungen* ein.

Alternativ könnte man auch das Modul *An für n Grad* verwenden. Hierfür müsste man aus der Anzahl der Umdrehungen den entsprechenden Winkel berechnen und diesen im Block *Standardsteuerung* verwenden.

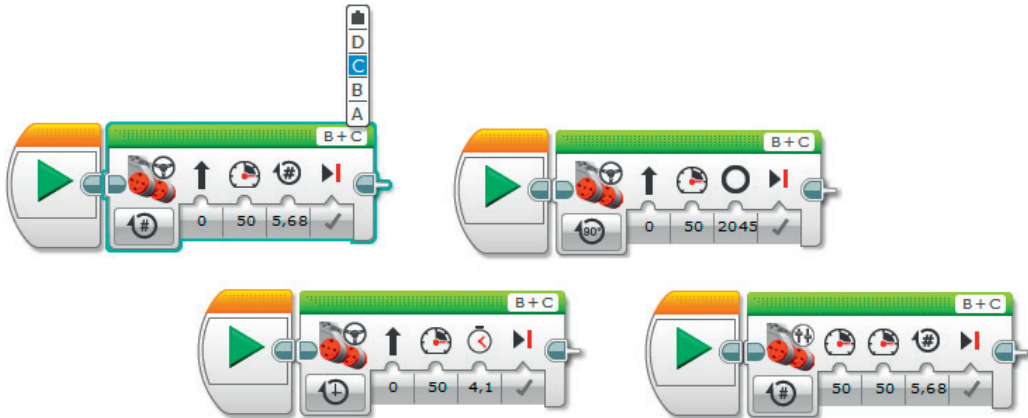
Eine weitere Möglichkeit wäre die Verwendung des Moduls *An für n Sekunden*. Die Zeit für die zurückzulegende Strecke kann jedoch nicht berechnet werden, denn dazu fehlt die Geschwindigkeit des fahrenden Roboters. Hier hilft nun nur die Trial-and-Error-Methode.

Natürlich kann man es auch bei den beiden vorangegangenen Möglichkeiten auf einen Versuch ankommen lassen.

Lösungen für die Hebelsteuerung lassen sich analog durchführen.

### Ausarbeitung

Für die Anzahl der Umdrehungen gilt:  $Strecke/Radumfang = 100\text{ cm}/17,6\text{ cm} = 5,68$ . Die entsprechende Grad-Zahl erhält man mit  $Umdrehungen \times (Winkel/Umdrehung) = 5,68 \times 360^\circ = 2045^\circ$ . Die angemessene *Zeit* muss man sich über die Trial-and-Error-Methode erarbeiten. Nach einer Fahrtzeit von einer Sekunde haben wir knapp 25 cm zurückgelegt. Dementsprechend führen 4,1 s zum Ziel.



### Verständnisfrage

Mit welcher mittleren Geschwindigkeit fährt der Roboter, wenn man die Leistung auf dem Standardwert von 50% belässt? Die Geschwindigkeit kann man berechnen, indem man die zurückgelegte Strecke durch die dafür benötigte Zeit teilt.

### Antwort auf die Verständnisfrage

Geschwindigkeit = Strecke/Zeit = 1 m/4,1 s = 24,4 cm/s = 0,036 km/h.

## ■ 2.4 Hebelsteuerung – eine halbe Umdrehung um die eigene Achse

### Erfassen der Aufgabenstellung

Baue einen Roboter, der sich programmgesteuert genau eine halbe Umdrehung um seine vertikale Achse dreht.

### Erschließen des Themas

Baue entsprechend der Bauanleitung in Abschnitt 1.1.2 bzw. im *Robot Educator (Robot Educator/Bauanleitungen/Fahrgestell)* das Fahrgestell für einen Roboter oder entwerfe ein eigenes Fahrgestell. Achte auf eine ausreichende Stromversorgung und einen Anschluss der USB-Verbindungsleitung.