

LEGO MINDSTORMS: AUSFÜHRLICHE ANLEITUNG EV3

1	INHALT	
2	Verwendung der Ausführlichen Anleitung.....	6
3	Sich als Lehrperson vorbereiten.....	6
4	Grundsätzliches.....	6
5	Erste Schritte.....	7
6	Programme	8
6.1.1	Programmierblöcke bearbeiten	8
6.1.2	Reihenleitungen	9
6.1.3	Code-Segmente	9
6.1.4	Die Grösse von Ablauf Programmierblöcken ändern	9
6.1.5	Parallellreihen	10
7	Anschlussauswahl	11
7.1.1	Motor-Anschlüsse	11
7.1.2	Sensor-Anschlüsse.....	11
7.1.3	Anschlussfehler	12
8	Datenleitungen.....	12
8.1.1	Typen von Datenleitungen	13
8.1.2	Datenleitung Erstellen.....	13
8.1.3	Datenleitung löschen	14
9	Verwendung von Sensoren	14
9.1	Infrarot (IR) – Näherungsmodus.....	14
9.1.1	Daten des Infrarotsensors im Näherungsmodus.....	14
9.2	Ultraschall.....	15
9.2.1	Vom Ultraschall gelieferte Daten	15
9.2.2	Beispiele für die Verwendung des Ultraschallsensors	16
9.3	Farbsensor	16
9.3.1	Farbmodus	17
9.3.2	Modus „ Stärke des reflektierenden Lichts“	17
9.3.3	Modus „Stärke des Umgebungslichts“	18
9.3.4	Daten des Farbsensors	18
10	Berührungssensor	20
10.1.1	Vom Berührungssensor gelieferte Daten	20
10.1.2	Beispiele für die Verwendung des Zustandes „gedrückt“	21
10.1.3	Beispiele für die Verwendung des Zustandes „Ausgelassen“	22
10.1.4	Den Zustand „Angestossen“ Verstehen.....	23
10.1.5	Beispiele für die Verwendung des Zustandes „Angestossen	23
11	Eigene Blöcke erstellen	23

11.1.1	Einen eigenen Block erstellen	24
12	Aktions-Blöcke	26
12.1	Mittlerer Motor	26
12.1.1	Wähle den Anschluss und den Steuermodus für deinen Motor.....	26
12.1.2	Modi.....	26
12.1.3	Leistung und Drehrichtung deines Motors	28
12.1.4	Eingaben	29
12.2	Grosser Motor	30
12.2.1	Modi.....	30
12.2.2	Leistung und Drehrichtung des Motors	32
12.2.3	Eingaben	33
12.3	Bewegungslenkung.....	33
12.3.1	Wähle die Anschlüsse und den Steuermodus für deine Motoren	34
12.3.2	Modi.....	34
12.3.3	Leistung und Drehrichtung des Motors	37
12.3.4	Lenkung und Motordrehzahl	37
12.3.5	Eingaben	37
12.4	Hebellenkungs-Block	38
12.4.1	Wähle deine Motor-Anschlüsse und den Steuermodus	39
12.4.2	Modi.....	39
12.4.3	Leistung und Drehrichtung des Motors	42
12.4.4	Eingaben	43
12.5	Anzeige	44
12.5.1	Wähle den Modus „Anzeige“	44
12.5.2	Anzeigekoordinaten.....	44
12.5.3	Modi.....	45
12.5.4	Angezeigte Objekte sichtbar lassen	49
12.5.5	Anzeige mehrerer Objekte.....	50
12.5.6	Zahlen anzeigen	50
12.5.7	Eingaben	50
12.6	Klang	52
12.6.1	Wähle den Klangmodus	52
12.6.2	Modi.....	53
12.6.3	Eingaben	55
13	Programmieraufbau-Blöcke	56
13.1	Start	56
13.2	Warten.....	57

13.2.1	Wähle den Wartemodus aus	57
13.2.2	Sensor-Vergleichsmodi	58
13.2.3	Auf einen Sensor-Schwellenwert warten	58
13.2.4	Auf bestimmte Sensorwerte warten.....	59
13.2.5	Sensor-Änderungsmodi	60
13.2.6	Warten, dass sich ein Sensorwert um einen bestimmten Betrag ändert	60
13.2.7	Darauf warten, dass sich ein Sensorwert in einen anderen Wert ändert.....	61
13.2.8	Eingaben und Ausgaben	62
13.3	Schleife	63
13.3.1	Wähle den Schleifenmodus	64
13.3.2	Modi.....	64
13.3.3	Sensor Modi	67
13.3.4	Sensordaten mit einem Schwellenwert vergleichen	67
13.3.5	Sensordaten mit bestimmten Werten vergleichen	69
13.3.6	Verwendung der Ausgabe „Zählen“	69
13.3.7	Eingaben und Ausgaben	70
13.4	Schalter	71
13.4.1	Ansicht im Registerformat	72
13.4.2	Eine Sensorschwelle Testen	74
13.4.3	Auf bestimmten Sensorwerte Testen	75
13.4.4	Auf mehrere Werte testen.....	76
13.4.5	Einen Wert aus einer Datenleitung testen.....	78
13.4.6	Modi.....	78
13.4.7	Eingaben	80
13.5	Schleifen-Interrupt	80
13.5.1	Wähle den Schleifennamen	81
13.5.2	Modus	81
14	Sensor-Blöcke.....	82
14.1	Ultraschall.....	82
14.1.1	Wähle den Anschluss und den Modus des Sensors aus	82
14.1.2	Modi.....	82
14.1.3	Eingaben und Ausgaben	83
14.2	Infrarot.....	84
14.2.1	Wähle den Anschluss und den Modus des Sensors aus	84
14.2.2	Modi.....	85
14.2.3	Eingaben und Ausgaben	85
14.3	Farbe.....	86

14.3.1	Wähle den Anschluss und den Modus des Sensors	86
14.3.2	Modi.....	87
14.3.3	Kalibrierungsmodi.....	88
14.3.4	Eingaben und Ausgaben	89
14.4	Berührung.....	90
14.4.1	Wähle den Anschluss und den Modus des Sensors	91
14.4.2	Modi.....	91
14.4.3	Eingaben und Ausgaben	92
15	Daten-Blöcke.....	93
15.1	Variable.....	93
15.1.1	Neue Variable hinzufügen.....	93
15.1.2	In eine Variable schreiben	94
15.1.3	Aus einer Variablen lesen	95
15.1.4	Beispiele für die Verwendung von Variablen.....	95
15.2	Logische Werte	96
15.2.1	Logische Verknüpfungen.....	96
15.3	Mathe	98
15.3.1	Wähle die mathematische Operation aus.....	98
15.3.2	Einfache mathematische Operationsmodi	98
15.4	Vergleichen.....	99
15.4.1	Vergleichsmodi	100
15.5	Zufall	101
15.5.1	Wähle den Ausgaben-Typ aus.....	101
15.5.2	Modi.....	102
16	Literatur	103

2 VERWENDUNG DER AUSFÜHRLICHEN ANLEITUNG

Die hier abgedruckte Anleitung brauchen die Schülerinnen und Schüler im Allgemeinen nicht zu lesen. Vielmehr versteht sie sich als Nachschlagewerk, wenn Probleme auftreten oder die Lehrperson und die Schülerinnen und Schüler etwas genauer wissen möchten.

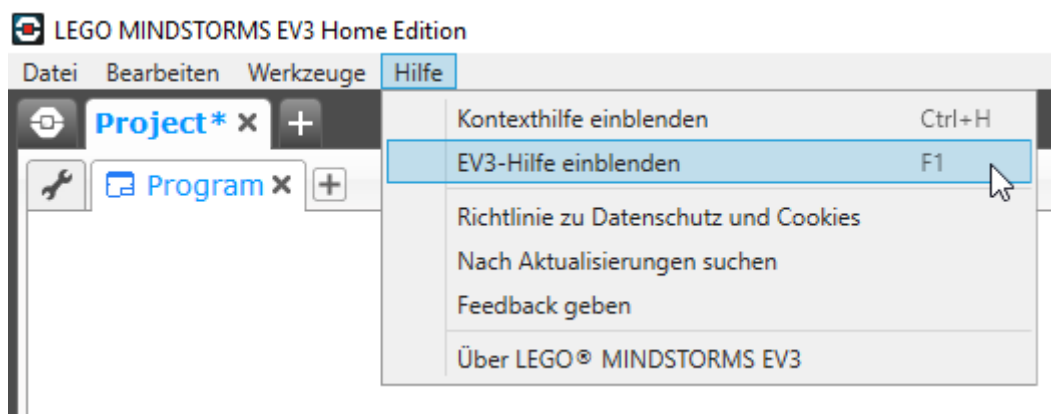
3 SICH ALS LEHRPERSON VORBEREITEN

Grundsätzlich gilt folgendes: Wissen durch praktisches Arbeiten. Es lohnt sich deshalb sehr, wenn die Lehrperson die Aufgaben der Schülerinnen und Schüler zumindest einmal durchlöst. So sammelt man schnell Erfahrung mit dem Themengebiet und gewinnt Sicherheit.

Tipp: Für wen die englische Sprache kein Problem darstellt, der findet auf der Videoplattform YouTube viele weitere Anleitungen.

4 GRUNDSÄTZLICHES

Bei der ausführlichen Anleitung zum Roboter EV3 handelt es sich im Wesentlichen um die in der LEGO MINDSTORMS EV3 Software eingebaute Hilfe (*LEGO MINDSTORMS EV3 Home Edition*, 2013). Die Hilfe kann entweder über den Reiter Hilfe → EV3-Hilfe einblenden oder direkt über die Taste „F1“ aufgerufen werden. Je nach Klasse bietet es sich an, dass Schülerinnen und Schüler hier zuerst selbständig nach Antworten für ihre entsprechenden Fragen suchen.



Es erscheint folgendes Fenster, welches den Nutzer durch die verschiedenen Themen navigieren lässt.

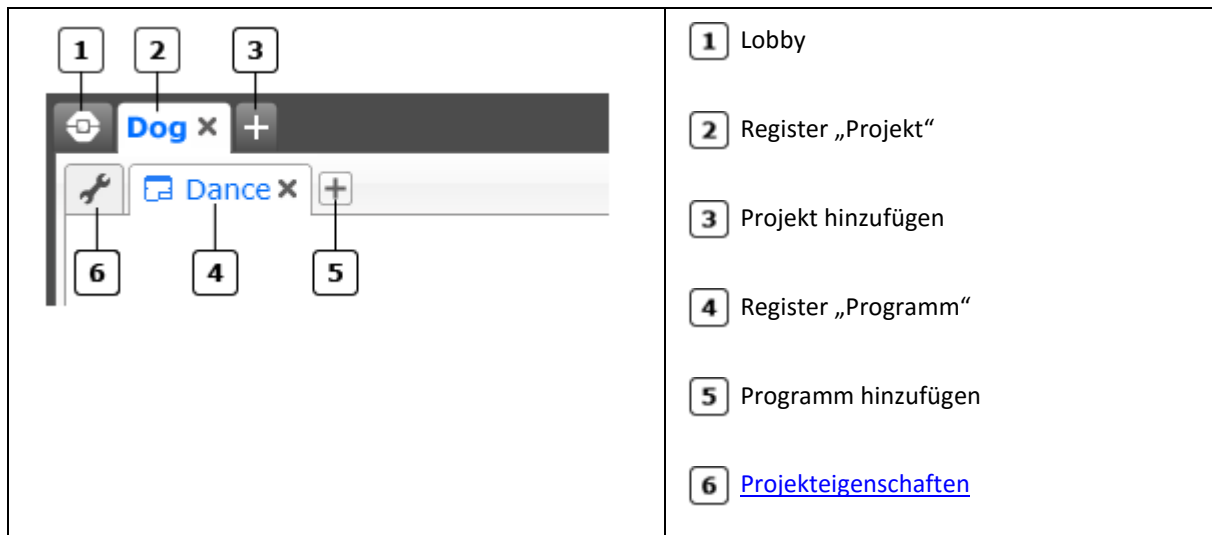


Im offiziellen Hilfe-Bereich findet man eine komplette Anleitung inklusive Bildern, Beispielen und Tipps. Für den interessierten Nutzer ist der Hilfs-Bereich einen Blick Wert. Hier in der ausführlichen Anleitung findet man eine gekürzte Version des Hilfs-Bereich. Die Anleitungen beschränken sich dabei auf die im Unterricht explizit verwendeten Blöcke und Modi. Die einzelnen Einträge sind gegenüber des Hilfs-Bereiches daher teilweise kürzer gehalten. Inhaltlich unterscheiden sich die einzelnen Artikel aber nicht.

5 ERSTE SCHRITTE

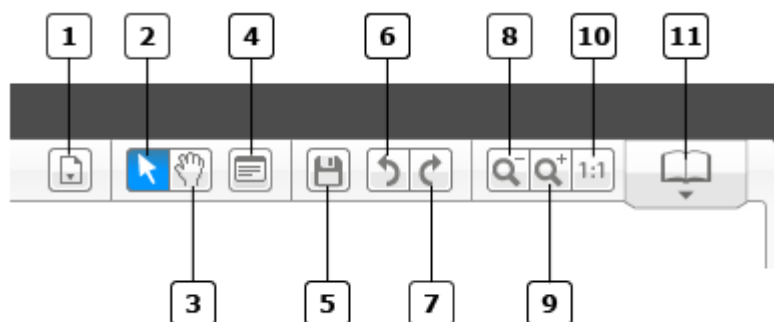
Die EV3-Software wird in der Lobby (Startseite) geöffnet. Zum Aufrufen der Programmierumgebung klickst du auf „Projekt öffnen“, um ein bestehendes Projekt zu öffnen. Du kannst auf „Neues Projekt“ klicken, um ein neues Projekt zu erstellen.

DATEIORGANISATION



Wenn du z.B. einen Roboterhund gebaut hast, kannst du ein Projekt „Hund“ anlegen und innerhalb dieses Projekts zahlreiche Programme, Bilder und Klang-Dateien speichern, mit denen das Verhalten dieses Roboterhundes bestimmt wird. Wenn du dann einen Farbsortierer baust, würdest du ein Projekt „Farbsortierer“ anlegen, in dem alle Programme, Bilder und Klang-Dateien für diesen Roboter enthalten sind.

SYMBOLLEISTE DES PROGRAMMIERBEREICHS



- 1 **Programmliste:** Wenn du auf dieses Werkzeug klickst, wird das Dropdownmenü „Programmliste“ eingeblendet. Dort sind alle derzeit innerhalb des Projekts geöffneten Programme aufgelistet.
- 2 **Auswählen:** Um dein Programm bearbeiten zu können, musst du dich in diesem Modus befinden. Klicke auf „Auswählen“ und benutze dann deine Maus, um einen einzelnen zu bearbeitenden Programmierblock auszuwählen. Alternativ hierzu kannst du einen Rahmen um mehrere Programmierblöcke ziehen, um sie auszuwählen.

- 3** Bildschirmausschnitt ändern
- 4** Kommentar
- 5** Projekt speichern: Wenn du das Projekt speicherst, werden auch alle mit dem Projekt verknüpften Programme gespeichert.
- 6** Rückgängig: Macht die letzte Aktion im Programmierbereich rückgängig.
- 7** Wiederherstellen
- 8** Verkleinern
- 9** Vergrößern
- 10** Zoom zurücksetzen
- 11** Inhalts-Editor öffnen

6 PROGRAMME

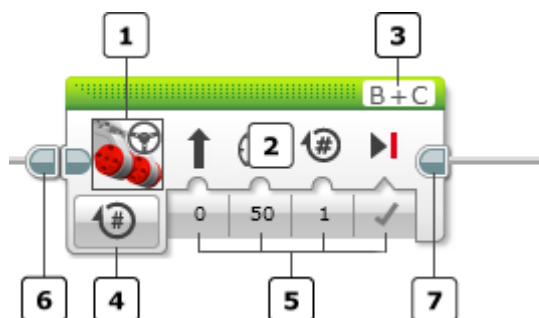
Du kannst Programm erstellen, indem du Programmierblöcke (aus den Programmierpaletten am unteren Rand des Bildschirms) in den Programmierbereich ziehst.

Wenn zwischen den Programmierblöcken nur ein geringer Abstand herrscht, docken sie automatisch aneinander an.



Wenn du dein Programm ausführst, werden die Programmierblöcke in der Reihenfolge ausgeführt, in der sie auf dem Bildschirm angezeigt werden (von links nach rechts).

6.1.1 PROGRAMMIERBLÖCKE BEARBEITEN

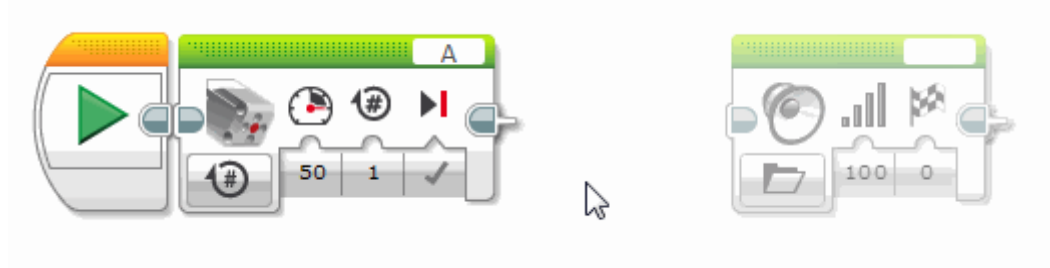


- 1** Block-Typ: Das Symbol zeigt den Block-Typ an.

- 2** Ziehpunkt des Blocks Klicke auf diesen Ziehpunkt, um den Block auszuwählen oder zu ziehen.
- 3** Anschlussauswahl
- 4** Modus-Auswahl Klicke hier, um die Modus-Auswahlliste zu öffnen und den Modus auszuwählen.
- 5** Eingabeparameter-Werte: Hier können Eingaben ausgewählt oder eingetragen werden.
- 6** Eingang des Reihensteckers
- 7** Ausgang des Reihensteckers

6.1.2 REIHENLEITUNGEN

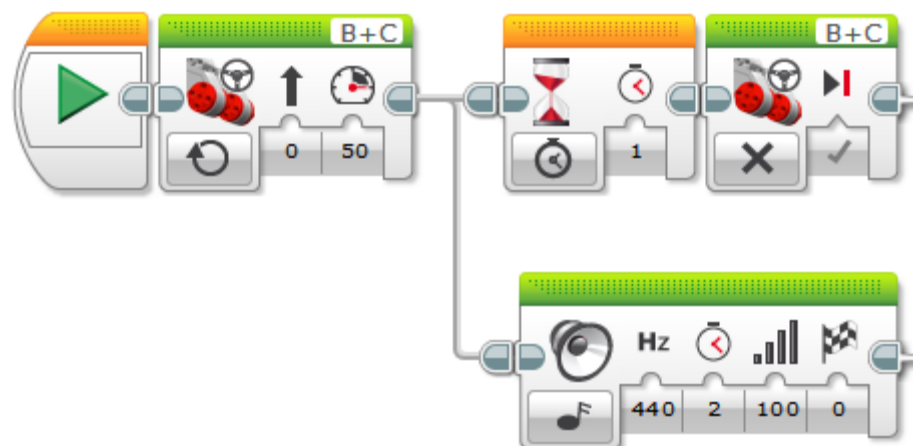
Programmierblöcke, die sich nicht direkt nebeneinander befinden, kannst du trotzdem miteinander verbinden. Zieh einfach eine Reihenleitung vom ersten zum zweiten Programmierblock.



Eine Reihenleitung kann entfernt werden, indem du auf den Eingang des Reihensteckers am zweiten Programmierblock klickst.

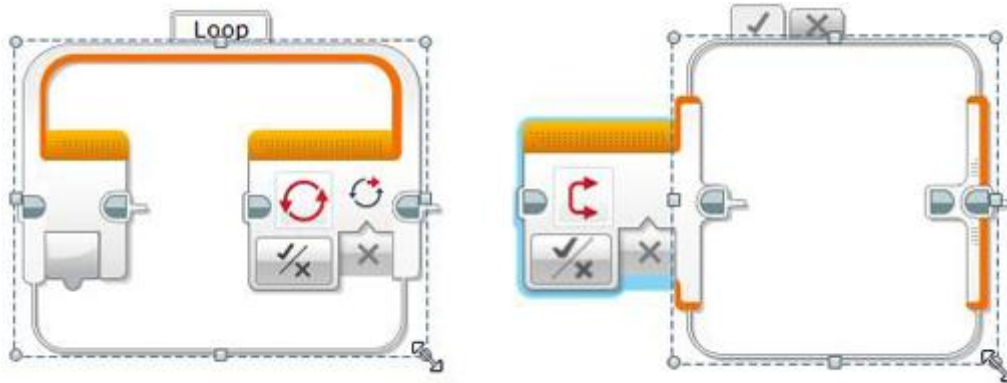
6.1.3 CODE-SEGMENTE

Bei längeren Programmen bietet es sich an, das Programm in kleinere Segmente aus mehreren Programmierblöcken aufzuteilen, zwischen denen ein Leerfeld gelassen wird. Auf diese Weise lässt sich das Programm leichter verstehen.

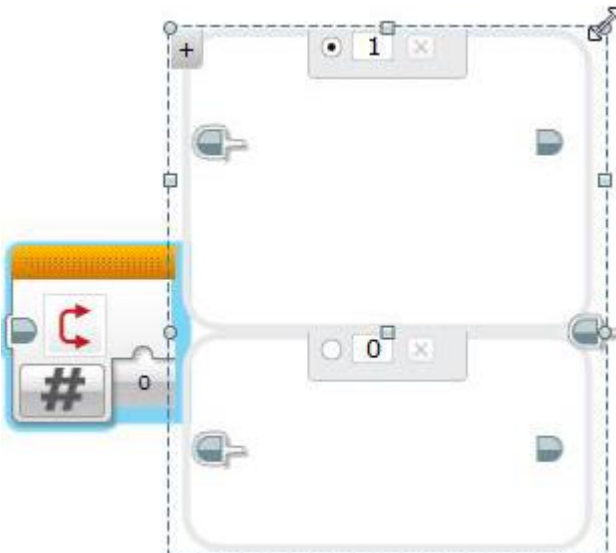


6.1.4 DIE GRÖSSE VON ABLAUF PROGRAMMIERBLÖCKEN ÄNDERN

Die Größe der Schleifen- und Schalter-Blöcke lässt sich ändern. Du kannst Schleifen und Schalter vergrößern, um dort weitere Programmierblöcke unterbringen zu können. Du vergrößerst diese Programmierblöcke, indem du an den Ziehpunkten zur Größenänderung ziehst.

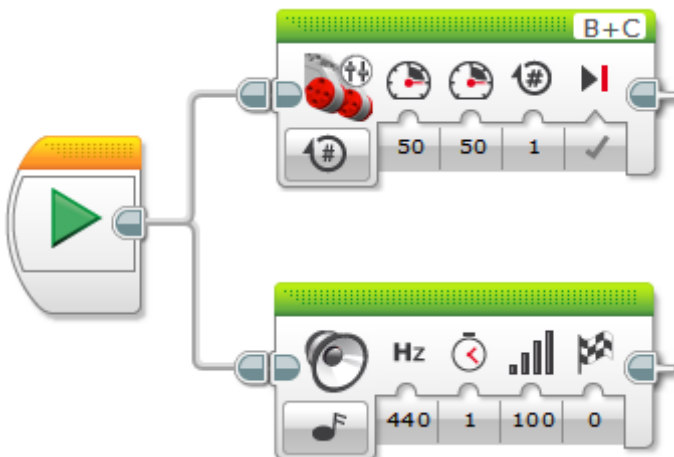


In einem Schalter lässt sich die Größe jedes Falls separat ändern.



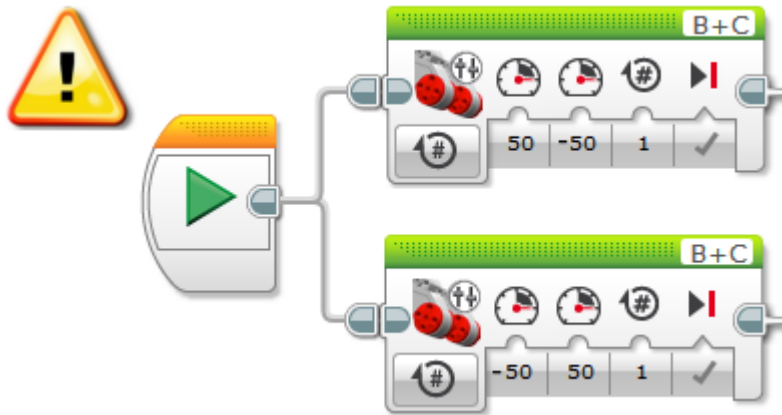
6.1.5 PARALLELLREIHEN

Du kannst gleichzeitig mehrere Aufgabenreihen ausführen. Du könntest bspw. eine Reihe von Programmierblöcken die Vorwärtsbewegung des Roboters und ein zweite Reihe von Programmierblöcken den Roboterarm oben am Roboter steuern lassen.



Tipps und Tricks

Achte auf Ressourcenkonflikte (siehe unten)! Zu einem Ressourcenkonflikt kann es bei der gleichzeitigen Ausführung mehrerer Aufgaben kommen, wenn zum Beispiel eine Reihe von Programmierblöcken versucht, den Roboter nach links zu lenken, während eine andere Reihe gleichzeitig versucht, den Roboter nach rechts zu lenken. Das Verhalten des EV3-Steins bei einem solchen Ressourcenkonflikt lässt sich nicht vorhersagen.

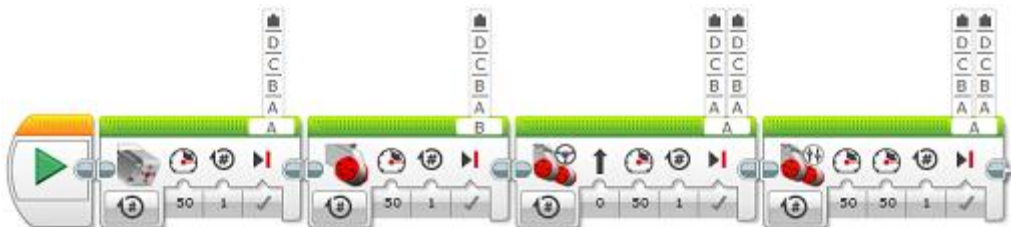


7 ANSCHLUSSAUSWAHL

Viele Programmierblöcke erfordern die Auswahl der von diesen Blöcken verwendeten Anschlüsse am EV3-Stein (A, B, C, D, 1, 2, 3 und 4). Die Anschlussauswahl befindet sich jeweils in der oberen rechten Ecke dieser Blöcke.

7.1.1 MOTOR-ANSCHLÜSSE

Die Abbildung zeigt die Anschlussauswahl an sämtlichen Motor-Blöcken.

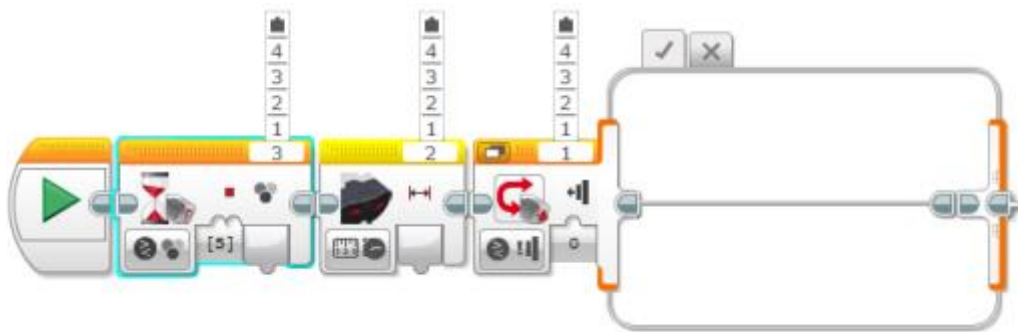


Klicke auf die Anschlussauswahl, um die Anschlussliste anzuzeigen und einen Anschluss auszuwählen. Motoren müssen an die Anschlüsse A, B, C oder D am EV3-Stein angeschlossen werden.

Da die Bewegungslenkungs- und Hebellenkungs-Blöcke zwei verschiedene Motoren steuern, verfügen diese beiden Blöcke jeweils über zwei Anschlussauswahlen. Die erste Anschlussauswahl ist für den linken Motor vorgesehen, und die zweite für den rechten Motor.

7.1.2 SENSOR-ANSCHLÜSSE

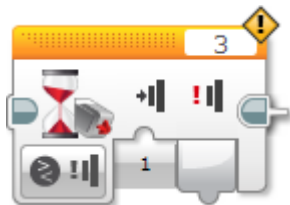
Die nachstehende Abbildung zeigt die Anschlussauswahlen an einigen der Blöcke, die eine Sensor-Eingabe verwenden.



Klicke auf die Anschlussauswahl, um die Anschlussliste anzuzeigen und einen Anschluss auszuwählen. Die Sensoren müssen an die Anschlüsse 1, 2, 3 oder 4 am Intelligenten EV3-Stein angeschlossen werden.

7.1.3 ANSCHLUSSFEHLER

Je nach ausgewähltem Anschluss kann die Anschlussauswahl einen Anschlussfehler anzeigen, wie unten dargestellt.

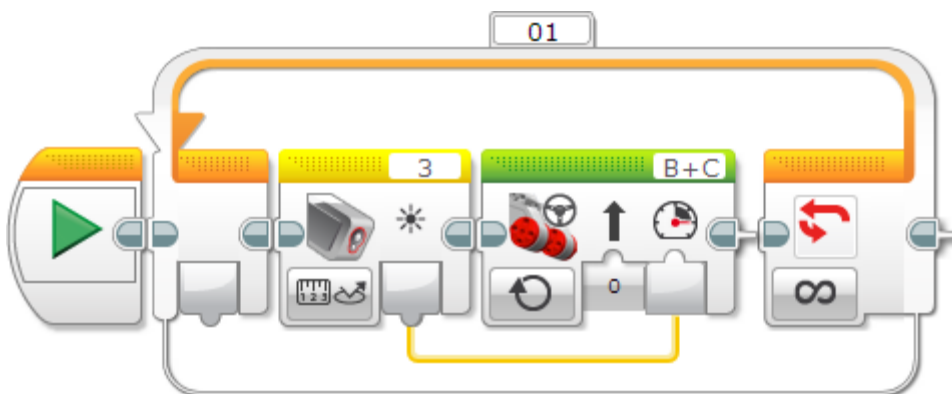


Hierbei handelt es sich um eine Warnung, dass der an den ausgewählten Anschluss am EV3-Stein angeschlossene Motor oder Sensor nicht mit dem Typ übereinstimmt, den der Programmierblock erfordert. Wenn du bspw. den Anschluss 3 für einen Berührungssensor-Block auswählst, der EV3-Stein jedoch am Anschluss 3 keinen Sensor bzw. einen anderen Sensortyp erkennt, dann wird ein Anschlussfehler angezeigt.

8 DATENLEITUNGEN

Eine Datenleitung ermöglicht es dir, einen Ausgabewert von einem Programmierblock in deinem Programm als Eingangswert für einen anderen Programmierblock in demselben Programm zu verwenden. Auf diese Weise kannst du Blöcke miteinander interagieren lassen und deinem Roboter komplexere Verhaltensweisen „beibringen“.

Beispiel



In diesem Programm wird eine Datenleitung verwendet, um die Lichtausgabe des Farbsensor-Blocks an die Leistungseingabe des Bewegungslenkungs-Blocks anzuschließen. Auf diese Weise wird die

Fahrgeschwindigkeit des Roboters durch die vom Farbsensor erkannte Lichtstärke bestimmt. Der Prozess wird in einer Schleife wiederholt, sodass die Leistung ständig aktualisiert wird – und zwar auf Grundlage der jeweils neuesten Sensormesswerte.

8.1.1 TYPEN VON DATENLEITUNGEN

Datenleitungen übertragen Werte von einem Block zu einem anderen. Der Typ der jeweiligen Datenleitung wird vom Ausgabebetyp des Blocks am Ausgangspunkt der Leitung bestimmt. Dieser Ausgabebetyp gibt außerdem an, welcher Wert von der Datenleitung übertragen wird.

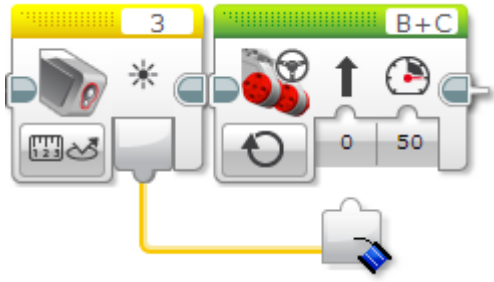
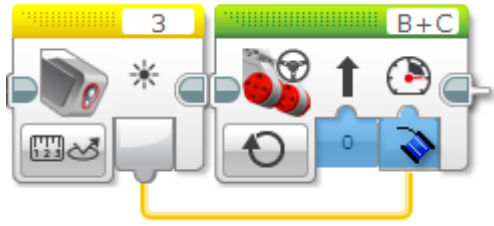
Datenleitungen, Block-Eingaben und Block-Ausgaben besitzen je nach Typ ein anderes Erscheinungsbild.

Typ	Block-Eingabe	Block-Ausgabe	Block-Ausgabe Datenleitung
Logische Werte			
Numerische Werte			
Text			
Numerisches Array			
Logisches Array			

8.1.2 DATENLEITUNG ERSTELLEN

Eine Datenleitung wird erstellt, indem sie von der Ausgabe eines Programmierblocks zur Eingabe eines anderen Programmierblocks gezogen wird (siehe die nachstehenden Schritte).

Schritt	Aktion	Beispiel
1.	Beginne an einer Block-Ausgabe.	

2.	Ziehe von links nach rechts.	
3.	Beende das Ziehen an einer Block-Eingabe.	

Eine Datenleitung ersetzt den Eingabewert, der direkt in den Block eingegeben wurde. Im Beispiel oben wird der Wert 50 in der Eingabe „Leistung“ durch eine Datenleitung ersetzt. Jetzt bezieht der Bewegungslenkung-Block den Wert für die Eingabe „Leistung“ aus der Datenleitung.

8.1.3 DATENLEITUNG LÖSCHEN

Klicke auf die Block-Eingabe am Ende der Datenleitung, um die Datenleitung zu löschen. Die Datenleitung wird gelöscht, und der Eingabewert, der direkt in den Block eingegeben wurde, wird wiederhergestellt. Klicke erneut auf die Block-Eingabe, um einen neuen Eingabewert einzugeben.

9 VERWENDUNG VON SENSOREN

9.1 INFRAROT (IR) – NÄHERUNGSMODUS



Im Nahemodus strahlt der Infrarotsensor ein IR-Signal aus, um dann zu erkennen, wie dieses Signal von einem Objekt vor dem Sensor reflektiert wird. Die Stärke des reflektierten Signals kann verwendet werden, um die Nähe (Distanz) zu dem Objekt abzuschätzen. Du könntest den Nahemodus bspw. verwenden, um zu erkennen, dass sich dein Roboter einer Wand nähert.

9.1.1 DATEN DES INFRAROTSENSORS IM NÄHERUNGSMODUS

Im Nahemodus gibt der Infrarotsensor die folgenden Daten aus:

Daten	Numerische Werte	Werte	Hinweise
Nähe	Numerische Werte	0 bis 100	Die relative Distanz zu einem Objekt vor dem Sensor. 0 bedeutet sehr nah, und 100 bedeutet sehr weit entfernt.

Tipps und Tricks

Der Wert „Nähe“ entspricht nicht direkt einer bestimmten Distanz. Der Wert ist vielmehr von der Farbe und dem Material des Objekts vor dem Sensor sowie von anderen Faktoren abhängig.

Der Infrarotsensor kann die Nähe zu einem Objekt, das sich sehr nah am Sensor befindet (näher als ca. 1 cm oder einen halben Zoll), nicht erkennen.

Beispiel 1: Halte an, bevor du gegen eine Wand fährst.



Dieses Programm lässt einen Roboter vorwärts fahren, bis der Infrarotsensor erkennt, dass er sich kurz vor einer Wand oder einem Objekt befindet. Nachdem der Roboter losgefahren ist, wird im Programm ein Warte(n)-Block im Modus „Infrarotsensor – Vergleichen – Nähe“ verwendet, um darauf zu warten, dass der Parameter „Nähe“ einen Wert von weniger als 35 erreicht, bevor es den Roboter anhält.

9.2 ULTRASCHALL



Der Ultraschallsensor kann die Distanz zu einem Objekt messen, das sich vor dem Ultraschallsensor befindet. Diese Messung wird vorgenommen, indem Schallwellen ausgestrahlt werden und dann die Zeit gemessen wird, bis die reflektierte Schallwelle wieder beim Sensor eintrifft.

Du kannst die Distanz zu einem Objekt in Zoll oder Zentimetern messen.

9.2.1 VOM ULTRASCHALL GELIEFERTE DATEN

Der Ultraschallsensor kann die folgenden Daten ausgeben:

Daten	Typ	Bereich	Hinweise
Distanz in Zentimetern	Numerische Werte	0 bis 255	Distanz zum Objekt in Zentimetern.

Tipps und Tricks

Der Ultraschallsensor erweist die besten Dienste, wenn er Objekte mit harter Oberfläche erkennen soll, die Schall gut reflektieren. Weiche Objekte wie Tücher könnten die Schallwellen absorbieren und

würden dann unter Umständen nicht erkannt werden. Objekte mit abgerundeten oder abgeschrägten Oberflächen sind ebenfalls schwerer zu erkennen.

Der Ultraschallsensor kann keine Objekte erkennen, die sich sehr nah am Sensor befinden (näher als ca. 3 cm oder 1,5 Zoll).

Der Ultraschallsensor besitzt ein breites „Sehfeld“ und könnte anstelle eines weiter entfernten Objekt direkt vor dem Sensor ein näheres Objekt erkennen, das ein wenig seitlich versetzt ist.

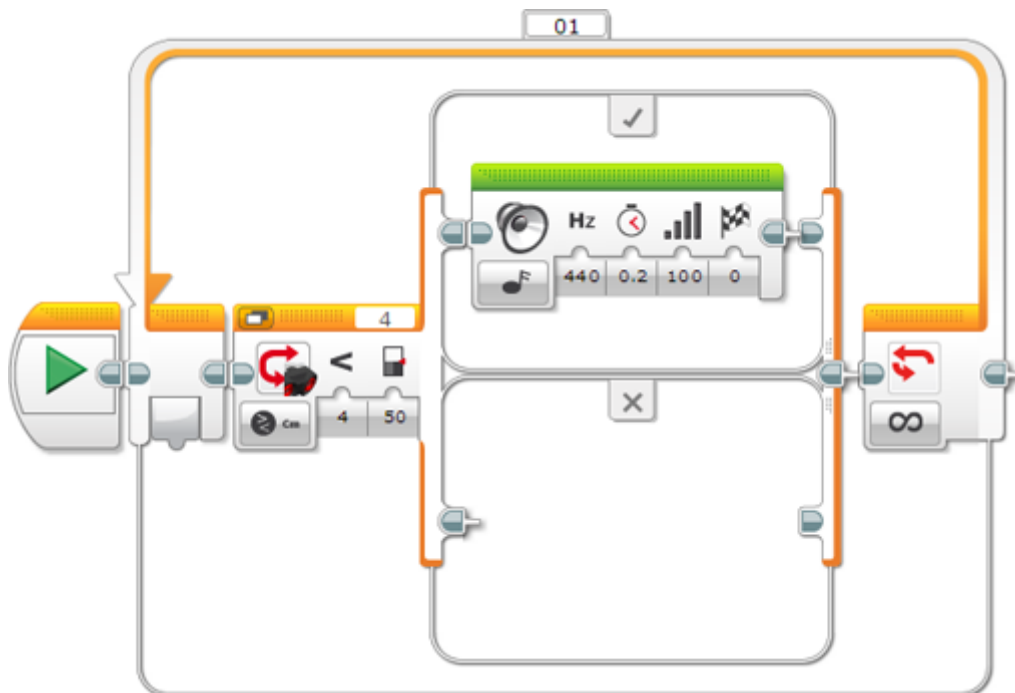
9.2.2 BEISPIELE FÜR DIE VERWENDUNG DES ULTRASCHALLSENSORS

Beispiel 1: In einer bestimmten Distanz zu einer Wand stoppen



Dieses Programm lässt einen Roboter vorwärts fahren, bis der Ultraschallsensor etwas erkennt, was weniger als 10 Zoll entfernt ist. Daraufhin wird der Roboter gestoppt. In diesem Programm wird der Warte(n)-Block im Modus „Ultraschallsensor – Vergleichen – Abstand in Zoll“ verwendet, um darauf zu warten, dass die erkannte Distanz weniger als 10 Zoll beträgt. Wenn der Ultraschallsensor nach vorn gerichtet ist, wird der Roboter ca. 10 Zoll vor einer Wand gestoppt.

Beispiel 2: Einen Alarm ertönen lassen, wenn ein Objekt in der Nähe erkannt wird.



Dieses Programm lässt einen Roboter jedes Mal einen Alarmton abgeben, wenn der Ultraschallsensor ein Objekt erkennt, das weniger als 50 Zentimeter entfernt ist. In diesem Programm wird ein Schalter im Modus „Ultraschallsensor – Vergleichen – Abstand in Zentimetern“ verwendet, um zu testen, ob die erkannte Distanz weniger als 50 cm beträgt. Ist dies der Fall, dann spielt der Schalter einen Ton ab. Der Schalter wird in einer Schleife wiederholt, damit der Test ständig wiederholt wird.

9.3 FARBESENSOR

Der Farbsensor kann die Farbe oder die Stärke des Lichts erkennen, das durch das kleine Fenster in der Vorderseite des Sensors eindringt. Der Farbsensor kann in drei verschiedenen Modi verwendet werden: Farbmodus, Modus „Stärke des reflektierten Lichts“ und Modus „Stärke des Umgebungslichts“.

9.3.1 FARBMODUS

Im Farbmodus kann der Farbsensor die Farbe eines nahen Objekts bzw. die Farbe einer Oberfläche in der Nähe des Sensors erkennen.



Tipps und Tricks

Das Objekt oder die Oberfläche sollte sich sehr nah am Sensor befinden (ohne ihn zu berühren), damit die Farbe richtig erkannt wird. Doch auch dann erkennt der Sensor die Farbe nicht richtig. Bevor du also dein ganzes Programm veränderst, kontrollierst du am besten zuerst ob der Sensor die Farbe überhaupt erkennt.

9.3.2 MODUS „ STÄRKE DES REFLEKTIERENDEN LICHTS“

Im Modus „Stärke des reflektierten Lichts“ erkennt der Farbsensor die Stärke des Lichts, das in den Sensor eindringt. Die Lichtstärke wird als Prozentwert zwischen 0 und 100 gemessen, wobei 0 für „sehr dunkel“ und 100 für „sehr hell“ steht.

Wenn sich der Farbsensor im Modus „Stärke des reflektierten Lichts“ befindet, dann wird die rote LED-Lampe vorne am Sensor angeschaltet. Wenn sich der Sensor in der Nähe eines Objekts oder einer Oberfläche befindet, dann wird das rote Licht von dem Objekt bzw. der Oberfläche reflektiert und dringt in den Sensor ein, um dort erkannt zu werden. Das kannst du nutzen, um Farbtöne auf einer Oberfläche oder einem Objekt zu messen, weil dunklere Farbtöne weniger rotes Licht zum Sensor reflektieren.

Du kannst diesen Modus verwenden, um bspw. deinen Roboter einer schwarzen Linie auf einer weißen Oberfläche folgen zu lassen.



9.3.3 MODUS „STÄRKE DES UMGEBUNGSLICHTS“

Im Modus „Stärke des Umgebungslichts“ erkennt der Farbsensor die Stärke des Lichts, das in den Sensor eindringt – genau wie im Modus „Stärke des reflektierten Lichts“. Die Lichtstärke wird als Prozentwert zwischen 0 und 100 gemessen, wobei 0 für „sehr dunkel“ und 100 für „sehr hell“ steht.

Wenn sich der Farbsensor im Modus „Stärke des Umgebungslichts“ befindet, dann beginnt eine blaue LED-Lampe vorne am Sensor schwach zu leuchten. An dem blauen Licht erkennst du, dass sich der Sensor im Modus „Stärke des Umgebungslichts“ befindet.

Diesen Modus kannst du verwenden, um die Helligkeit der Zimmerbeleuchtung zu erkennen oder um zu erkennen, ob andere Lichtquellen auf den Sensor leuchten.



9.3.4 DATEN DES FARBSENSORS

Der Farbsensor kann die folgenden Daten ausgeben:

Daten	Typ	Be-reich	Hinweise
Farbe	Numeri-sche Werte	0 bis 7	Wird im Farbmodus verwendet. 0 = Keine Farbe 1 = Schwarz 2 = Blau 3 = Grün

			4 = Gelb 5 = Rot 6 = Weiß 7 = Braun
Licht	Numerische Werte	0 bis 100	Wird in den Modi „Stärke des reflektierten Lichts“ und „Stärke des Umgebungslichts“ verwendet. Misst die Lichtstärke als Prozentwert: 0 = sehr dunkel, 100 = sehr hell

Beispiel 1: Fahre, bis eine schwarze Linie erreicht wird (Methode 1)



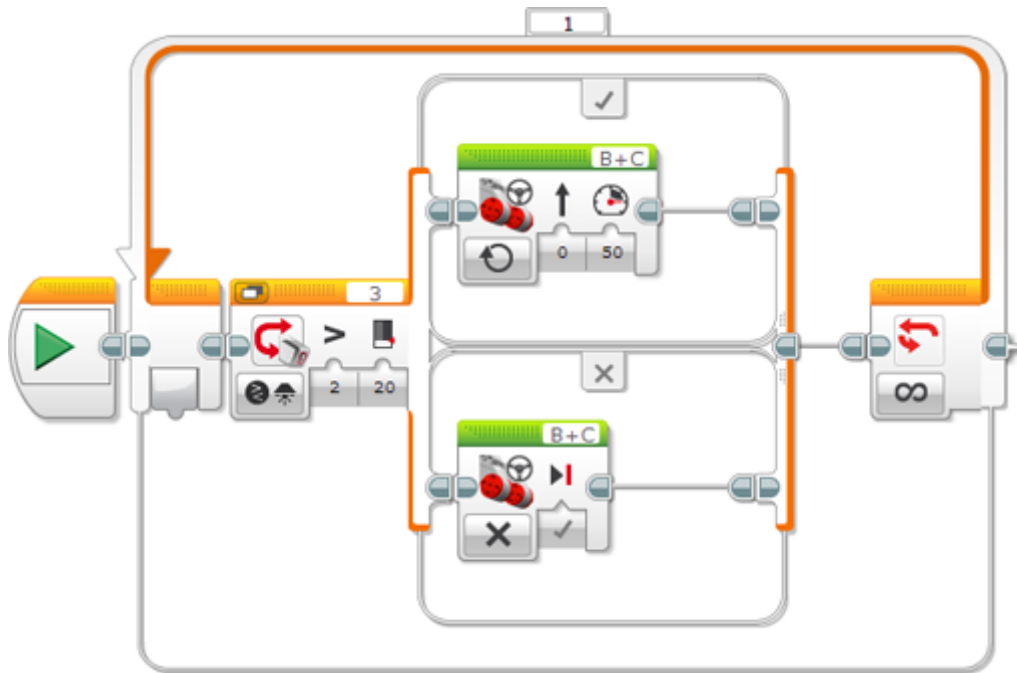
Dieses Programm lässt einen Roboter fahren, bis der Farbsensor eine schwarze Farbe erkennt. Daraufhin stoppt das Programm den Roboter. Bei diesem Programm wird ein Warte-Block im Modus „Farbsensor – Vergleichen – Farbe“ verwendet, um den Roboter nach schwarzer Farbe suchen zu lassen.

Beispiel 2: Fahre, bis eine schwarze Linie erreicht wird (Methode 2)



Dieses Programm lässt einen Roboter fahren, bis der Farbsensor eine dunkle Farbe erkennt. Daraufhin stoppt das Programm den Roboter. In diesem Programm wird der Warte-Block im Modus „Farbsensor – Vergleichen – Stärke des reflektierten Lichts“ verwendet, um darauf zu warten, dass eine Lichtstärke von weniger als 50 % erkannt wird.

Beispiel 3: Fahre nur bei eingeschalteter Zimmerbeleuchtung



Dieses Programm lässt einen Roboter fahren, wenn die Zimmerbeleuchtung eingeschaltet ist, und stoppen, wenn du die Lampen ausschaltetest. Bei diesem Programm wird ein Schalter im Modus „Farbsensor – Vergleichen – Stärke des Umgebungslichts“ verwendet, um zu testen, ob die Lichtstärke weniger als 20 % beträgt. Der Schalter entscheidet, ob die Motoren an- oder ausgeschaltet werden. Der Schalter wird in einer Schleife wiederholt, damit der Roboter immer wieder auf veränderte Lichtverhältnisse reagiert.

10 BERÜHRUNGSSENSOR



Der Berührungssensor erkennt, ob die Taste an der Vorderseite des Sensors gedrückt ist. Du kannst den Berührungssensor bspw. verwenden, um zu erkennen, ob der Roboter gegen etwas gefahren ist. Du könntest aber auch mit einem Finger auf den Berührungssensor drücken, um eine Aktion auszulösen.

Der Berührungssensor kann anzeigen, ob er gerade gedrückt wird (oder nicht). Er kann jedoch nicht messen wie weit bzw. wie kräftig er gedrückt wird. Der Berührungssensor gibt Logische Werte („Wahr“ oder „Falsch“) aus. Die Stellung der Berührungssensor-Taste wird als ihr Zustand bezeichnet. Der Zustand lautet „Wahr“, wenn die Taste gerade gedrückt wird, und „Falsch“, wenn sie gerade nicht gedrückt wird (ausgelassen wurde).

Der Berührungssensor kann auch verfolgen, ob die Taste in der Vergangenheit gedrückt und dann wieder ausgelassen wurde. Dieser Zustand wird als „Angestoßen“ bezeichnet und eignet sich bspw., um die Betätigung der Taste mit einem Finger zu erkennen.

10.1.1 VOM BERÜHRUNGSSSENROS GELIEFERTE DATEN

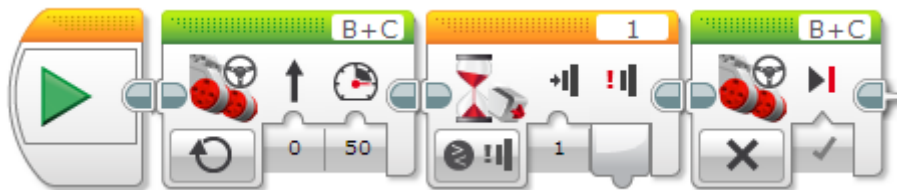
Der Berührungssensor kann die folgenden Daten ausgeben:

Daten	Typ	Hinweise
-------	-----	----------

Zustand	Logische Werte	„Wahr“, wenn die Taste gedrückt wird, „Falsch“, wenn die Taste nicht gedrückt wird.
Ge- drückt	Logische Werte	„Wahr“, wenn die Taste gedrückt wird, „Falsch“, wenn die Taste nicht gedrückt wird (mit dem Zustand identisch).
Ausge- lassen	Logische Werte	„Falsch“, wenn die Taste gedrückt wird, „Wahr“, wenn die Taste nicht gedrückt wird (das Gegenteil des Zustands).
Ange- stoßen	Logische Werte	„Wahr“, wenn die Taste in der Vergangenheit gedrückt und dann wieder ausge- lassen wurde. Das nächste Eintreten des Zustands „Angestoßen“ erfordert dann ein erneutes Drücken und Auslassen der Taste.

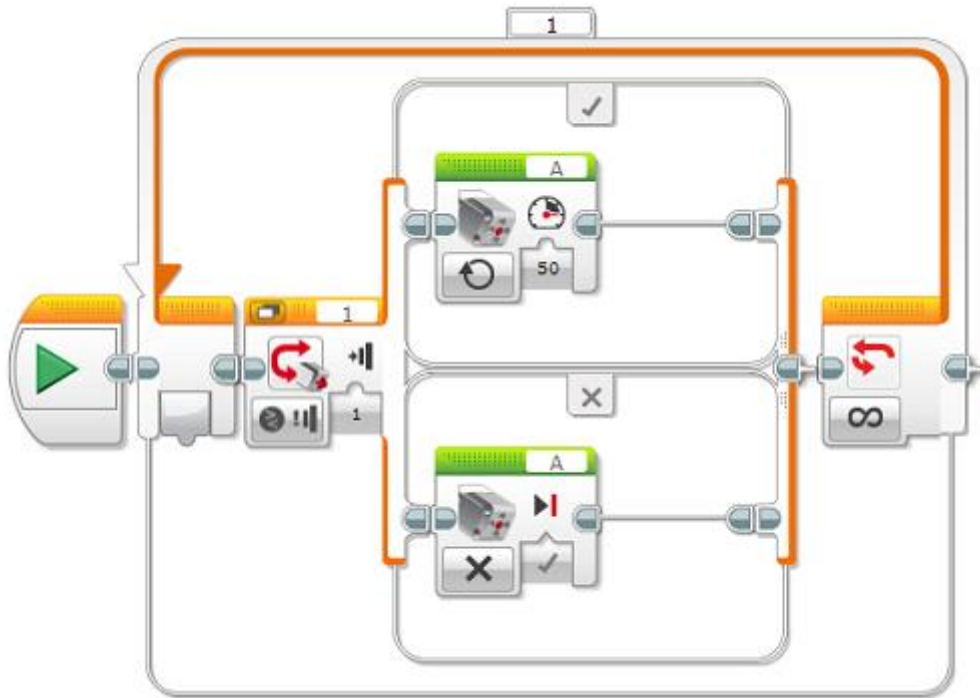
10.1.2 BEISPIELE FÜR DIE VERWENDUNG DES ZUSTANDES „GEDRÜCKT“

Beispiel 1: Fahre, bis ein Berührungssensor gedrückt wird.



Dieses Programm lässt einen Roboter in gerader Linie vorwärts fahren, bis ein Berührungssensor gedrückt wird, woraufhin der Roboter gestoppt wird. In diesem Programm wird ein Warte(n)-Block im Modus „Berührungssensor – Vergleichen – Berührung“ verwendet, um zu testen, ob die Taste gerade „Gedrückt“ wird.

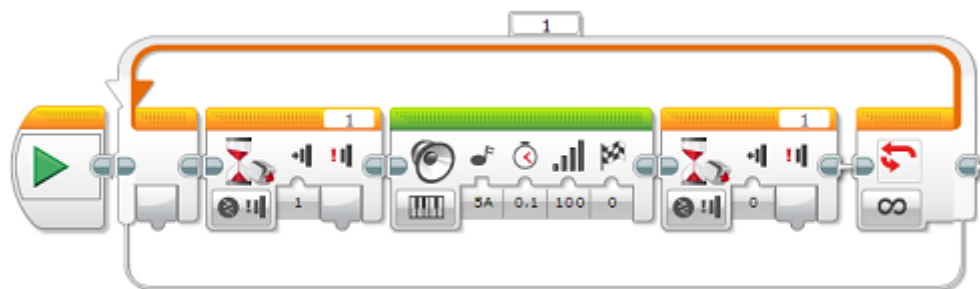
Beispiel 2: Einen Motor jedes Mal laufen lassen, wenn ein Berührungssensor gedrückt gehalten wird (Methode 1)



Dieses Programm lässt einen Motor jedes Mal laufen, wenn der Berührungssensor gedrückt gehalten wird. Der Motor wird jedes Mal gestoppt, wenn der Berührungssensor wieder ausgelassen wird. In diesem Programm wird ein Schalter-Block im Modus „Berührungssensor – Vergleichen – Berührung“ verwendet, um zu testen, ob die Taste gerade „Gedrückt“ wird. Das Ergebnis dieses Tests wird benutzt, um zu entscheiden, ob der Motor an- oder ausgeschaltet wird. Der Test wird in einer Schleife endlos wiederholt.

10.1.3 BEISPIELE FÜR DIE VERWENDUNG DES ZUSTANDES „AUSGELASSEN“

Beispiel 4: Bei jedem Drücken des Berührungssensors einen Piepton abgeben

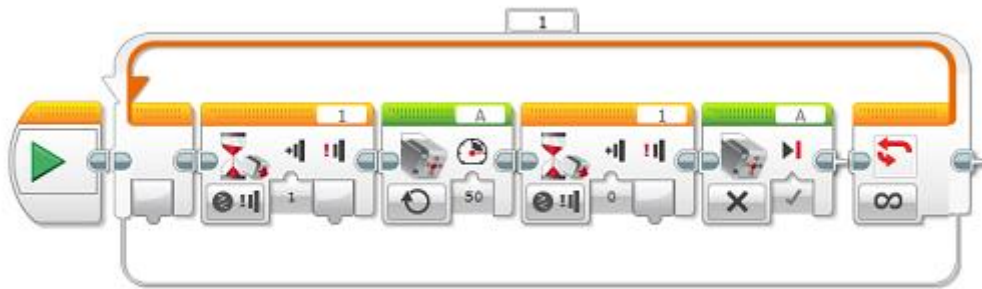


Dieses Programm lässt jedes Mal einen kurzen Piepton erklingen, wenn der Berührungssensor gedrückt wird. Bei jedem Drücken erklingt nur ein Piepton. In diesem Programm wird ein Warte(n)-Block im Modus „Berührungssensor – Vergleichen – Berührung“ verwendet, um zu testen, ob die Taste „Gedrückt“ wird, und dann ein weiterer Warte(n)-Block, um darauf zu warten, dass die Taste „Ausgelassen“ wird, bevor die Schleife fortgesetzt wird.

Tipps und Tricks

Wenn du den Warte-Block, mit dem das Auslassen der Taste abgewartet wird, aus diesem Programm löschst, wirst du feststellen, dass der Piepton wiederholt wird, solange der Berührungssensor gedrückt gehalten wird. Probiere es einfach mal aus!

Beispiel 5: Einen Motor jedes Mal laufen lassen, wenn ein Berührungssensor gedrückt gehalten wird (Methode 2)



Dieses Programm lässt einen Motor jedes Mal laufen, wenn der Berührungssensor gedrückt gehalten wird. Der Motor wird jedes Mal gestoppt, wenn der Berührungssensor wieder ausgelassen wird. In diesem Programm wird ein Warte(n)-Block im Modus „Berührungssensor – Vergleichen – Berührung“ verwendet, um darauf zu warten, dass die Taste „Gedrückt“ wird, um den Motor zu starten. Dann wird ein weiterer Warte(n)-Block verwendet, um darauf zu warten, dass die Taste „Ausgelassen“ wird, um den Motor zu stoppen. Dieser Prozess wird in einer Schleife wiederholt.

10.1.4 DEN ZUSTAND „ANGESTOSSEN“ VERSTEHEN

Der Berührungssensor meldet dir jedoch nicht nur, ob seine Taste derzeit gedrückt oder ausgelassen ist, sondern verfolgt auch, ob sie in der Vergangenheit gedrückt und wieder ausgelassen – also „Angestoßen“ – wurde.

Nachdem ein Berührungssensor angezeigt hat, dass er „Angestoßen“ wurde, zeigt er diesen Zustand erst wieder an, wenn der Berührungssensor erneut gedrückt und wieder ausgelassen wurde. Auf diese Weise lässt sich leicht sicherstellen, dass bspw. jedes Drücken der Taste einer nur einmal ausgeführten Aktion entspricht.

10.1.5 BEISPIELE FÜR DIE VERWENDUNG DES ZUSTANDES „ANGESTOSSEN“

Beispiel 6: Ändere die Anzeige, wenn der Berührungssensor gedrückt wird.



Dieses Programm lässt den EV3-Stein auf dem Display die Werte „Zero“ (null), dann „One“ (Eins) und dann „Two“ (Zwei) anzeigen. Die Anzeige ändert sich jedes Mal, wenn der Berührungssensor angestoßen (gedrückt und wieder ausgelassen) wird.

Tipps und Tricks

Wenn du die Warte-Blocks in diesem Programm so änderst, dass sie auf den Zustand „Gedrückt“ anstatt auf den Zustand „Angestoßen“ testen, wirst du feststellen, dass die Anzeige von „Zero“ (Null) direkt zu „Two“ (Zwei) wechselt und „One“ (Eins) einfach überspringt. Das liegt daran, dass die Anzeige-Blöcke so schnell ausgeführt werden, dass beim zweiten Warten auf den Zustand „Gedrückt“ die Taste noch immer gedrückt ist. Deshalb wird das zweite Warten (der zweite Warte-Block) sofort beendet. Beim Testen auf den Zustand „Angestoßen“ ist nach jedem Drücken jeweils nur ein Test erfolgreich.

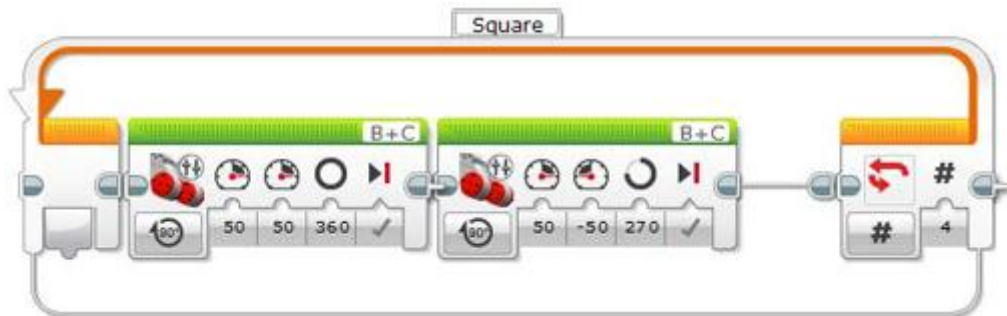
11 EIGENE BLÖCKE ERSTELLEN

Manchmal erstellst du ein richtig tolles Miniprogramm, das du vielleicht in einem anderen Projekt oder Programm erneut verwenden möchtest.

Das Miniprogramm besteht aus einer Reihe von Programmierblöcken, die du im Arbeitsbereich auswählst. Die Funktion „Eigene Blöcke erstellen“ erleichtert es dir, die Blöcke aus diesem Miniprogramm zu gruppieren, um einen Eigenen Block zu erstellen.

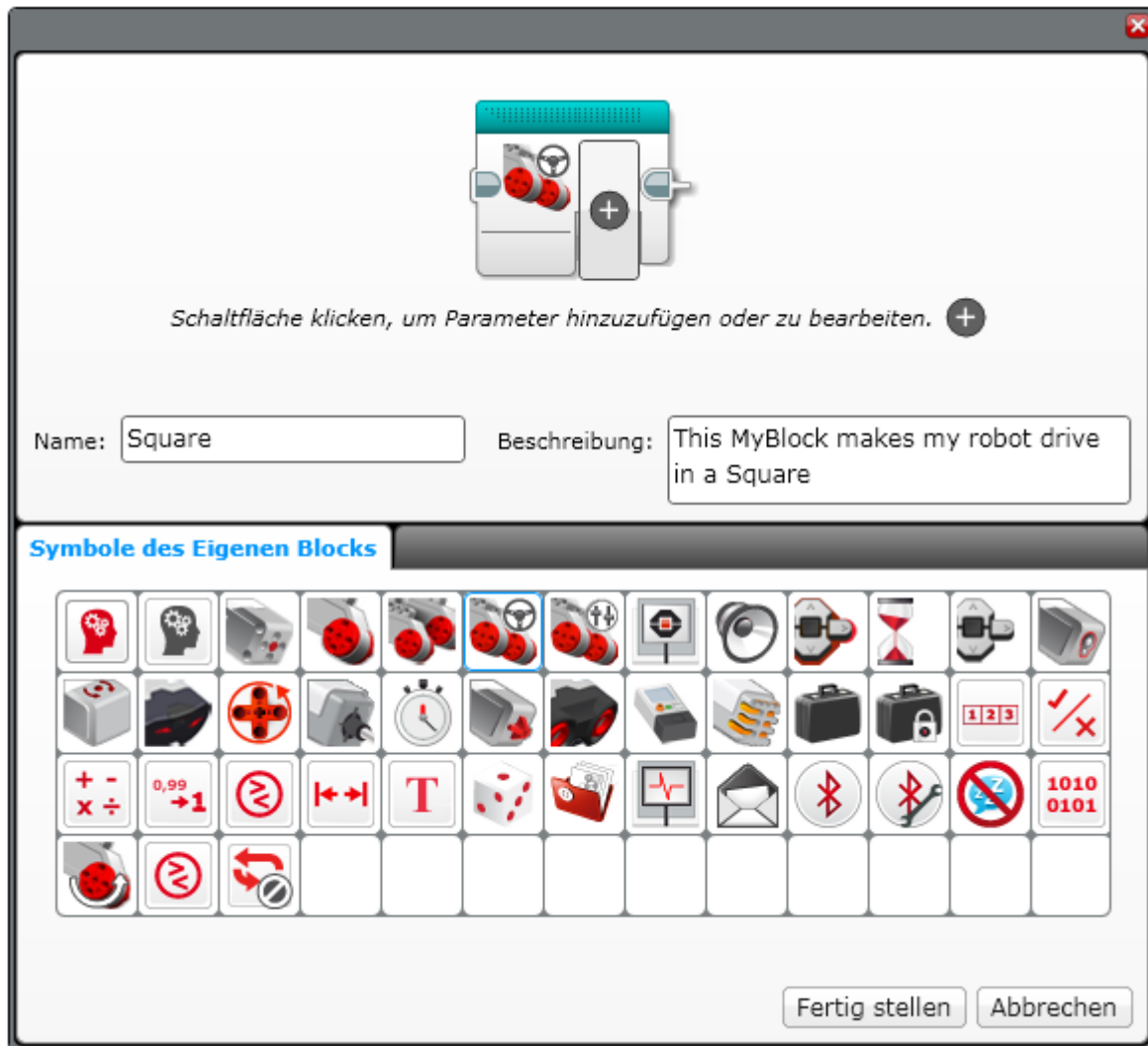
11.1.1 EINEN EIGENEN BLOCK ERSTELLEN

Mal angenommen, du hättest ein Miniprogramm erstellt, dass deinen Roboter ein Quadrat fahren lässt:



Du verwandelst dieses Miniprogramm in einen Block, indem du das Auswahl-Werkzeug verwendest: wähle alle gewünschte Blöcke aus, indem du einen Rahmen um sie ziehst.

Wähle im Werkzeug-Menü die Option „Eigene Blöcke erstellen“, um das Werkzeug „Eigene Blöcke erstellen“ zu starten.

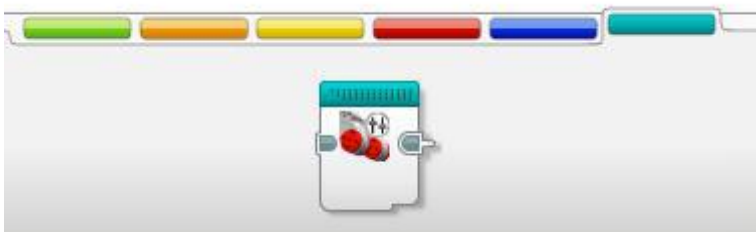


Gib deinem Block einen Namen und versieh ihn mit einer kurzen Beschreibung

Wähle eines der Symbole des Eigenen Blocks, um deinen Block leichter identifizieren zu können.

Klicke auf „Fertig stellen“.

Dein Miniprogramm wird jetzt in einen ganz individuellen Block umgewandelt! Alle Eigenen Blöcke, die du für ein Projekt erstellst, werden in der Palette „Eigene Blöcke“ angezeigt.



Du kannst den neuen Eigenen Block in den Programmierbereich ziehen, um ihn in deinem Programm zu verwenden.



Durch Doppelklicken auf den Eigenen Block lassen sich beliebige Teile dieses Blocks bearbeiten.

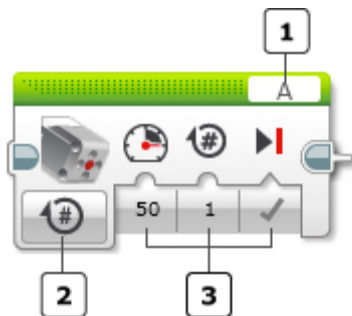
12 AKTIONS-BLÖCKE

12.1 MITTLERER MOTOR



Der Block „Mittlerer Motor“ steuert den Mittleren Motor. Du kannst den Motor ein- und ausschalten, die Leistungsstufe einstellen oder den Motor für eine bestimmte Anzahl an Sekunden oder Umdrehungen anschalten.

12.1.1 WÄHLE DEN ANSCHLUSS UND DEN STEUERMODUS FÜR DEINEN MOTOR



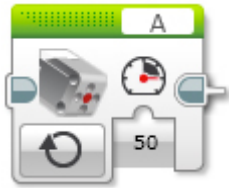
- 1** Anschlussauswahl
- 2** Modus-Auswahl
- 3** Eingaben

Wähle den Motor (A, B, C oder D), der vom Block „Mittlerer Motor“ gesteuert werden soll, indem du die Anschlussauswahl oben im Block verwendest.

Benutze die Modus-Auswahl, um festzulegen, wie der Motor gesteuert werden soll. Nach der Auswahl des Modus kannst du die Werte für die Eingaben festlegen.

12.1.2 MODI

An



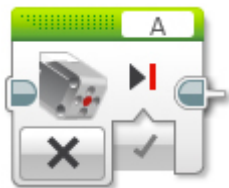
Im Modus „An“ wird der Motor angeschaltet und sofort im Anschluss wird mit dem nächsten Block im Programm weitergemacht. Drehzahl und Drehrichtung des Motors lassen sich mithilfe der Eingabe Leistung steuern. Der Motor läuft, bis er von einem späteren Block im Programm gestoppt wird oder einen anderen Befehl erhält bzw. bis das Programm endet.

Beispiel



Benutze den Modus „An“, wenn andere Blöcke in deinem Programm steuern sollen, wie lange der Motor angeschaltet bleibt. In diesem Programm läuft der Motor, bis ein Berührungssensor gedrückt wird. Erst dann wird der Motor gestoppt.

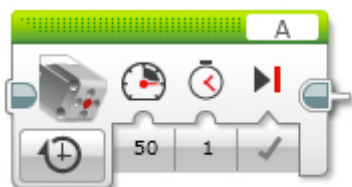
Aus



Mit dem Modus „Aus“ wird der Motor ausgeschaltet. Für gewöhnlich wird dieser Modus verwendet, um einen Motor zu stoppen, der mit dem Modus An angeschaltet wurde. Siehe Beispiel oben.

Wenn für die Eingabe Am Ende bremsen der Wert „Wahr“ ausgewählt wurde, wird der Motor sofort gestoppt. Der Motor wird in der gestoppten Stellung gehalten, bis ein anderer Motor-Block diesen Motor startet oder das Programm endet. Wenn für die Eingabe Am Ende bremsen der Wert „Falsch“ ausgewählt wird, dann wird einfach die Stromversorgung des Motors abgeschaltet. Der Motor läuft aus und nutzt dabei jeglichen verbliebenen Schwung, bis er stoppt oder bis ein anderer Motor-Block startet.

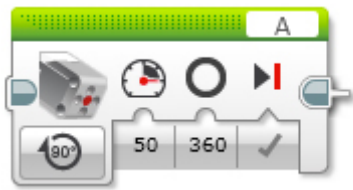
An für n Sekunden



Im Modus „An für n Sekunden“ wird der Motor genau für die Anzahl an Sekunden angeschaltet, die in der Eingabe Sekunden festgelegt wurde, und dann wird er wieder ausgeschaltet.

Drehzahl und Drehrichtung des Motors lassen sich mithilfe der Eingabe Leistung steuern. Verwende für die Eingabe Am Ende bremsen den Wert „Wahr“, um den Motor exakt nach der festgelegten Anzahl an Sekunden zu stoppen.

An für n Grad



Im Modus „An für n Grad“ wird der Motor genau für die Gradzahl angeschaltet, die in der Eingabe Grad festgelegt wurde, und dann wird er wieder ausgeschaltet. 360 Grad bewirken eine volle Umdrehung oder Rotation des Motors.

Drehzahl und Drehrichtung des Motors lassen sich mithilfe der Eingabe Leistung steuern. Verwende für die Eingabe Am Ende bremsen den Wert „Wahr“, um den Motor exakt nach der festgelegten Gradzahl zu stoppen.

Beispiel

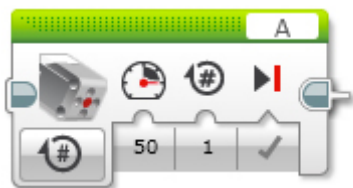


Dieses Programm lässt den Motor an Anschluss A mit 50 % Leistung genau 900 Grad (zweieinhalb Umdrehungen) laufen.

Tipps und Tricks

Bei Nutzung des Modus „An für n Grad“ wartet der Block, bis sich der Motor genau um die angegebene Gradzahl weitergedreht hat, bevor das Programm mit dem nächsten Block weitermacht. Falls der Motor auf Widerstand oder ein physisches Hindernis stößt und sich deshalb nicht um die angegebene Gradzahl drehen kann, nimmt der Block einen Wartezustand ein. Erst wenn der Widerstand beseitigt wurde, werden weitere Blöcke im Programm ausgeführt.

An für n Umdrehungen

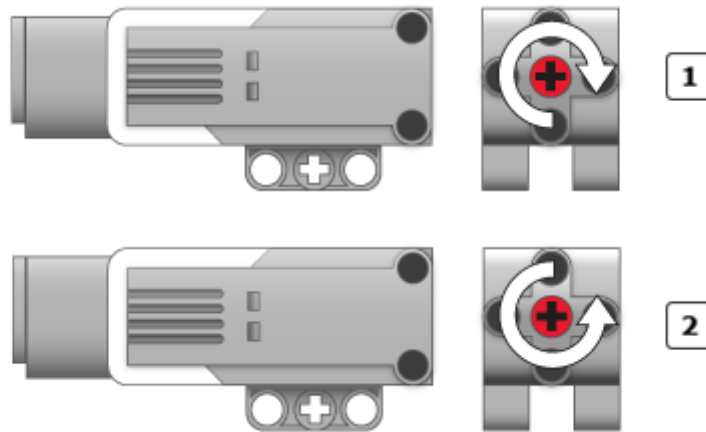


Im Modus „An für n Umdrehungen“ wird der Motor genau für die Anzahl an Umdrehungen angeschaltet, die in der Eingabe Umdrehungen festgelegt wurde, und dann wird er wieder ausgeschaltet.

Drehzahl und Drehrichtung des Motors lassen sich mithilfe der Eingabe Leistung steuern. Verwende für die Eingabe Am Ende bremsen den Wert „Wahr“, um den Motor exakt nach der festgelegten Anzahl an Umdrehungen zu stoppen.

12.1.3 LEISTUNG UND DREHRICHTUNG DEINES MOTORS

Für die Eingabe Leistung ist eine Zahl zwischen -100 und 100 zulässig. Bei Eingabe einer positiven Zahl dreht sich der Motor im Uhrzeigersinn, bei Eingabe einer negativen Zahl gegen den Uhrzeigersinn, wie der Grafik zu entnehmen ist.



1 Positive Leistung

2 Negative Leistung

12.1.4 EINGABEN

Mit den Eingaben des Blocks „Mittlerer Motor“ lässt sich im Detail steuern, wie sich der Motor verhalten soll. Du kannst die Eingabewerte direkt in den Block eingeben. Alternativ hierzu können als Eingabewerte auch die Ausgaben anderer Programmierblöcke dienen, die über Datenleitungen übertragen werden.

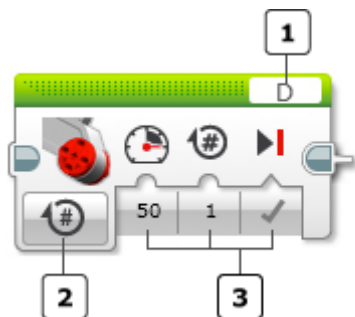
Eingabe	Typ	Zulässige Werte	Hinweise
Leistung	Numerische Werte	-100 bis 100	Motor-Leistungsstufe
Am Ende bremsen	Logische Werte	Wahr/Falsch	Kommt am Ende des Blocks zum Tragen. Wenn „Wahr“, wird der Motor sofort gestoppt und in seiner Stellung gehalten. Wenn „Falsch“, wird die Stromversorgung des Motors ausgeschaltet und der Motor darf noch auslaufen.
Sekunden	Numerische Werte	≥ 0	Bewegungszeit in Sekunden.
Grad	Numerische Werte	Beliebige Zahl	Betrag der Bewegung in Grad. 360 Grad bedeuten eine volle Umdrehung.
Umdrehungen	Numerische Werte	Beliebige Zahl	Betrag der Bewegung in Umdrehungen.

12.2 GROSSER MOTOR



Der Block „Großer Motor“ steuert den Großen Motor. Du kannst den Motor ein- und ausschalten, die Leistungsstufe einstellen oder den Motor für eine bestimmte Anzahl an Sekunden oder Umdrehungen anschalten.

WÄHLE DEN ANSCHLUSS UND DEN STEUERMODUS FÜR DEINEN MOTOR.



- 1** Anschlussauswahl
- 2** Modus-Auswahl
- 3** Eingaben

Wähle den Motor (A, B, C oder D), der vom Block „Großer Motor“ gesteuert werden soll, indem du die Anschlussauswahl oben im Block verwendest.

Benutze die Modus-Auswahl, um festzulegen, wie der Motor gesteuert werden soll. Nach der Auswahl des Modus kannst du die Werte für die Eingaben festlegen.

12.2.1 MODI

An



Im Modus „An“ wird der Motor angeschaltet und sofort im Anschluss wird mit dem nächsten Block im Programm weitergemacht. Drehzahl und Drehrichtung des Motors lassen sich mithilfe der Eingabe Leistung steuern. Der Motor läuft, bis er von einem späteren Block im Programm gestoppt wird oder einen anderen Befehl erhält bzw. bis das Programm endet.

Beispiel



Benutze den Modus „An“, wenn andere Blöcke in deinem Programm steuern sollen, wie lange der Motor angeschaltet bleibt. In diesem Programm läuft der Motor, bis ein Berührungssensor gedrückt wird. Erst dann wird der Motor gestoppt.

Aus



Mit dem Modus „Aus“ wird der Motor ausgeschaltet. Für gewöhnlich wird dieser Modus verwendet, um einen Motor zu stoppen, der mit dem Modus An angeschaltet wurde. Siehe Beispiel oben.

Wenn für die Eingabe Am Ende bremsen der Wert „Wahr“ ausgewählt wurde, wird der Motor sofort gestoppt. Der Motor wird in der gestoppten Stellung gehalten, bis ein anderer Motor-Block diesen Motor startet oder das Programm endet. Wenn für die Eingabe Am Ende bremsen der Wert „Falsch“ ausgewählt wird, dann wird einfach die Stromversorgung des Motors abgeschaltet. Der Motor läuft aus und nutzt dabei jeglichen verbliebenen Schwung, bis er stoppt oder bis ein anderer Motor-Block startet.

An für n Sekunden



Im Modus „An für n Sekunden“ wird der Motor genau für die Anzahl an Sekunden angeschaltet, die in der Eingabe Sekunden festgelegt wurde, und dann wird er wieder ausgeschaltet. Der Block wartet, bis diese eingestellte Zeit verstrichen ist, bevor das Programm mit dem nächsten Block weitermacht.

Drehzahl und Drehrichtung des Motors lassen sich mithilfe der Eingabe Leistung steuern. Verwende für die Eingabe Am Ende bremsen den Wert „Wahr“, um den Motor exakt nach der festgelegten Anzahl an Sekunden zu stoppen.

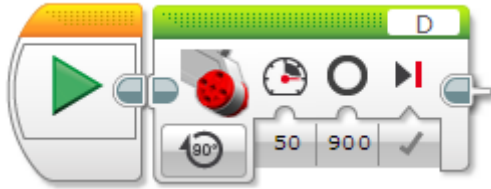
An für n Grad



Im Modus „An für n Grad“ wird der Motor genau für die Gradzahl angeschaltet, die in der Eingabe Grad für die Umdrehung festgelegt wurde, und dann wird er wieder ausgeschaltet. Der Wert 360 Grad bewirkt eine volle Umdrehung des Motors.

Drehzahl und Drehrichtung des Motors lassen sich mithilfe der Eingabe Leistung steuern. Verwende für die Eingabe Am Ende bremsen den Wert „Wahr“, um den Motor exakt nach der festgelegten Gradzahl zu stoppen.

Beispiel



Dieses Programm lässt den Motor an Anschluss D mit 50 % Leistung genau 900 Grad (zweieinhalb Umdrehungen) laufen und stoppt ihn dann.

Tipps und Tricks

Bei Nutzung des Modus „An für n Grad“ wartet der Block, bis sich der Motor genau um die angegebene Gradzahl weitergedreht hat, bevor das Programm mit dem nächsten Block weitermacht. Falls der Motor auf Widerstand oder ein physisches Hindernis stößt und sich deshalb nicht um die angegebene Gradzahl drehen kann, nimmt der Block einen Wartezustand ein. Erst wenn der Widerstand beseitigt wurde, werden weitere Blöcke im Programm ausgeführt.

An für n Umdrehungen

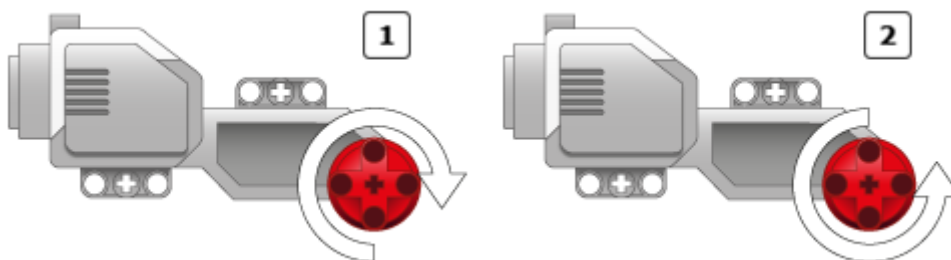


Im Modus „An für n Umdrehungen“ wird der Motor genau für die Anzahl an Umdrehungen angeschaltet, die in der Eingabe Umdrehungen festgelegt wurde, und dann wird er wieder ausgeschaltet.

Drehzahl und Drehrichtung des Motors lassen sich mithilfe der Eingabe Leistung steuern. Verwende für die Eingabe Am Ende bremsen den Wert „Wahr“, um den Motor exakt nach der festgelegten Anzahl an Umdrehungen zu stoppen.

12.2.2 LEISTUNG UND DREHRICHTUNG DES MOTORS

Für die Eingabe Leistung ist eine Zahl zwischen -100 und 100 zulässig. Durch Eingabe positiver und negativer Zahlen lässt sich die Drehrichtung des Großen Motors festlegen (siehe nachstehende Grafik).



1 Positive Leistung

2 Negative Leistung

12.2.3 EINGABEN

Mit den Eingaben des Blocks „Großer Motor“ lässt sich im Detail steuern, wie sich der Motor verhalten soll. Du kannst die Eingabewerte direkt in den Block eingeben. Alternativ hierzu können als Eingabewerte auch die Ausgaben anderer Programmierblöcke dienen, die über Datenleitungen übertragen werden. Die verfügbaren Eingaben und deren Funktionen sind vom ausgewählten Steuermodus abhängig.

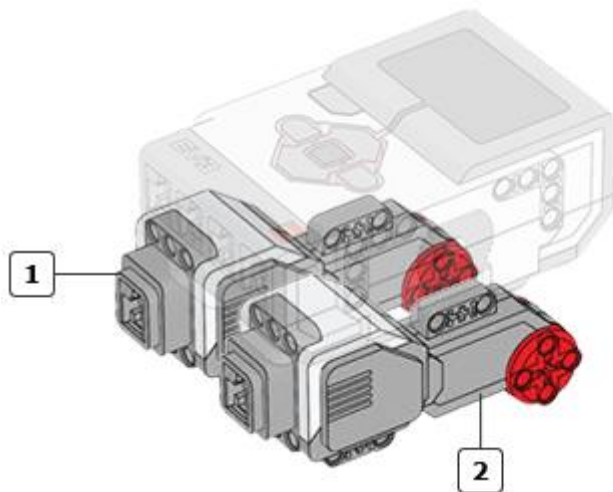
Eingabe	Typ	Zulässige Werte	Hinweise
Leistung	Numerische Werte	-100 bis 100	Motor-Leistungsstufe
Am Ende bremsen	Logische Werte	Wahr/Falsch	Kommt am Ende des Blocks zum Tragen. Wenn „Wahr“, wird der Motor sofort gestoppt und in seiner Stellung gehalten. Wenn „Falsch“, wird die Stromversorgung des Motors ausgeschaltet und der Motor darf noch auslaufen.
Sekunden	Numerische Werte	≥ 0	Bewegungszeit in Sekunden.
Grad	Numerische Werte	Beliebige Zahl	Betrag der Bewegung in Grad. 360 Grad bedeuten eine volle Umdrehung.
Umdrehungen	Numerische Werte	Beliebige Zahl	Betrag der Bewegung in Umdrehungen.

12.3 BEWEGUNGSLENKUNG



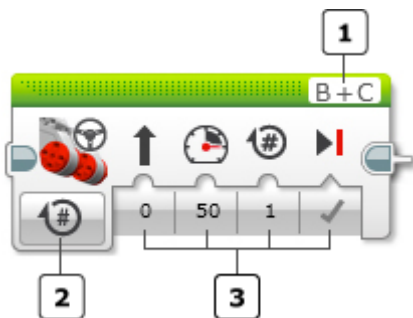
Mit dem Bewegungslenkungs-Block kannst du deinen Roboter vorwärts und rückwärts fahren, wenden oder anhalten lassen. Du kannst die Lenkung so anpassen, dass dein Roboter geradeaus, weite Bögen oder enge Kurven fährt.

Benutze den Bewegungslenkungs-Block für Roboterfahrzeuge mit zwei Großen Motoren, bei denen einer die linke Seite und der andere die rechte Seite des Fahrzeugs antreibt.



- 1 Linker Motor
- 2 Rechter Motor

12.3.1 WÄHLE DIE ANSCHLÜSSE UND DEN STEUERMODUS FÜR DEINE MOTOREN



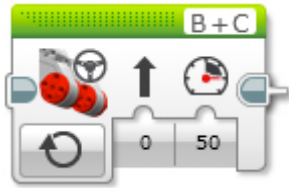
- 1 Anschlussauswahl
- 2 Modus-Auswahl
- 3 Eingaben

Wähle die beiden Motoren (A, B, C oder D), die vom Bewegungslenkungs-Block gesteuert werden sollen, indem du die Anschlussauswahl oben im Block verwendest. Klicke auf den jeweiligen Buchstaben des Motors, um den entsprechenden Anschluss für diesen Motor auszuwählen. An den ersten Anschluss sollte Motor auf der linken Seite des Fahrzeugs angeschlossen werden und an den zweiten Anschluss sollte der Motor auf der rechten Seite des Fahrzeugs angeschlossen werden.

Benutze die Modus-Auswahl, um festzulegen, wie die Motoren gesteuert werden sollen. Nach der Auswahl des Modus kannst du die Werte für die Eingaben festlegen.

12.3.2 MODI

An



Im Modus „An“ werden beide Motoren angeschaltet und sofort im Anschluss wird mit dem nächsten Block im Programm weitergemacht. Drehzahl und Drehrichtung der Motoren lassen sich mithilfe der Eingaben Leistung und Lenkung steuern. Die Motoren laufen, bis sie von einem späteren Block im Programm gestoppt werden oder einen anderen Befehl erhalten bzw. bis das Programm endet.

Wenn der Wert in der Eingabe Lenkung ungleich null ist, wird einer der Motoren abgebremst oder umgekehrt, um den Roboter drehen/wenden zu lassen. Bei diesem Beispiel wird der Wert der Eingabe Grad an dem Motor gemessen, der sich schneller dreht.

Beispiel



Benutze den Modus „An“, wenn andere Blöcke in deinem Programm steuern sollen, wie lange die Motoren angeschaltet bleiben. In diesem Programm fährt der Roboter vorwärts. Wenn der Berührungssensor gedrückt wird, stoppt der Roboter.

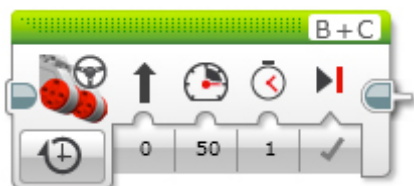
Aus



Mit dem Modus „Aus“ werden beide Motoren ausgeschaltet. Benutze den Modus „Aus“, um einen Roboter zu stoppen, der an früherer Stelle im Programm durch den Modus An gestartet wurde. Siehe Beispiel oben.

Wenn für die Eingabe Am Ende bremsen der Wert „Wahr“ ausgewählt wurde, werden die Motoren sofort gestoppt. Die Motoren werden in der gestoppten Stellung gehalten, bis ein anderer Bewegungs- oder Motor-Block sie wieder startet oder das Programm endet. Wenn für die Eingabe Am Ende bremsen der Wert „Falsch“ ausgewählt wird, dann wird einfach die Stromversorgung der Motoren abgeschaltet. Die Motoren laufen aus und nutzen dabei jeglichen verbliebenen Schwung, bis sie stoppen oder bis ein anderer Bewegungs- oder Motor-Block startet.

An für n Sekunden



Im Modus „An für n Sekunden“ werden beide Motoren genau für die Anzahl an Sekunden angeschaltet, die in der Eingabe Sekunden festgelegt wurde, und dann werden sie wieder ausgeschaltet. Der Block wartet, bis diese eingestellte Zeit verstrichen ist, bevor das Programm mit dem nächsten Block weitermacht.

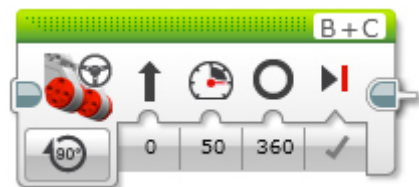
Geschwindigkeit und Fahrtrichtung deines Roboters lassen sich mithilfe der Eingaben Leistung und Lenkung steuern. Verwende die Eingabe Am Ende bremsen, um deinen Roboter exakt nach der festgelegten Anzahl an Sekunden zu stoppen.

Beispiel



Dieses Programm lässt deinen Roboter mit voller Leistung 2 Sekunden in gerader Linie vorwärts fahren und dann anhalten.

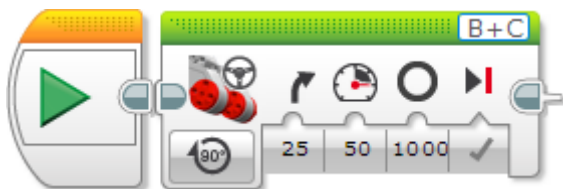
An für n Grad



Im Modus „An für n Grad“ werden beide Motoren angeschaltet. Dann wird gewartet, bis einer der Motoren die in der Eingabe Grad über die eingestellte Gradzahl definierte Umdrehung vollführt hat, und anschließend werden beide Motoren ausgeschaltet. Dieser Modus kann benutzt werden, um deinen Roboter eine bestimmte Distanz zurücklegen oder um einen bestimmten Betrag (eine bestimmte Umdrehungszahl) drehen zu lassen. 360 Grad entsprechen einer vollen Umdrehung des Motors.

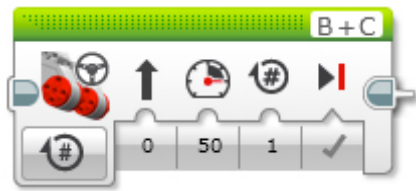
Geschwindigkeit und Fahrtrichtung deines Roboters lassen sich mithilfe der Eingaben Leistung und Lenkung steuern. Verwende die Eingabe Am Ende bremsen, um deinen Roboter exakt nach der festgelegten Gradzahl zu stoppen.

Beispiel



Dieses Programm lässt einen Roboter mithilfe eines Werts von 25 in der Eingabe Lenkung und mit 50 % Leistung einen Bogen nach rechts vollführen. Diese Kurvenfahrt wird fortgesetzt, bis sich der linke (äußere und schnellere) Motor genau um 1000 Grad gedreht hat, und dann stoppt der Roboter.

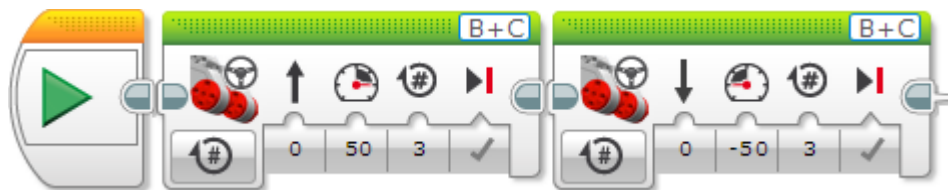
An für n Umdrehungen



Im Modus „An für n Umdrehungen“ werden beide Motoren angeschaltet. Dann wird gewartet, bis einer der Motoren die in der Eingabe Umdrehungen anhand der eingestellten Umdrehungszahl definierte Umdrehung vollführt hat, und anschließend werden beide Motoren ausgeschaltet.

Geschwindigkeit und Fahrtrichtung deines Roboters lassen sich mithilfe der Eingaben Leistung und Lenkung steuern. Verwende die Eingabe Am Ende bremsen, um deinen Roboter exakt nach der festgelegten Anzahl an Motorumdrehungen zu stoppen.

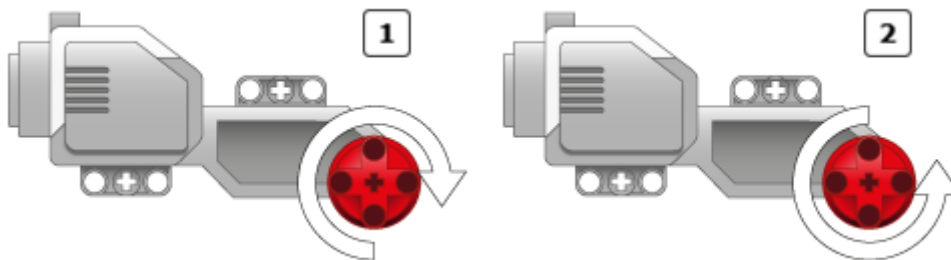
Beispiel



Dieses Programm lässt deinen Roboter mit 50 % Leistung in gerader Linie vorwärts fahren, bis die Motoren 3 volle Umdrehungen vollführt haben. Dann wird er in gerader Linie 3 Umdrehungen rückwärtsfahren (mithilfe einer negativen Zahl in der Eingabe Leistung) und dort stoppen, wo er ursprünglich losgefahren war.

12.3.3 LEISTUNG UND DREHRICHTUNG DES MOTORS

Für die Eingabe Leistung ist eine Zahl zwischen -100 und 100 zulässig. Mit positiven und negativen Zahlen in der Eingabe Leistung lässt sich die Drehrichtung des Großen Motors festlegen (siehe nachstehende Grafik).



1 Positive Leistung

2 Negative Leistung

12.3.4 LENKUNG UND MOTORDREHZAHL

Für die Eingabe Lenkung ist eine Zahl zwischen -100 und 100 zulässig. Ein Wert von 0 (null) lässt deinen Roboter geradeaus fahren. Eine positive Zahl (größer als null) lässt deinen Roboter nach rechts drehen, eine negative Zahl (kleiner als null) lässt deinen Roboter dagegen nach links drehen. Je stärker der Lenkungswert von null abweicht, desto enger wird die gefahrene Kurve.

12.3.5 EINGABEN

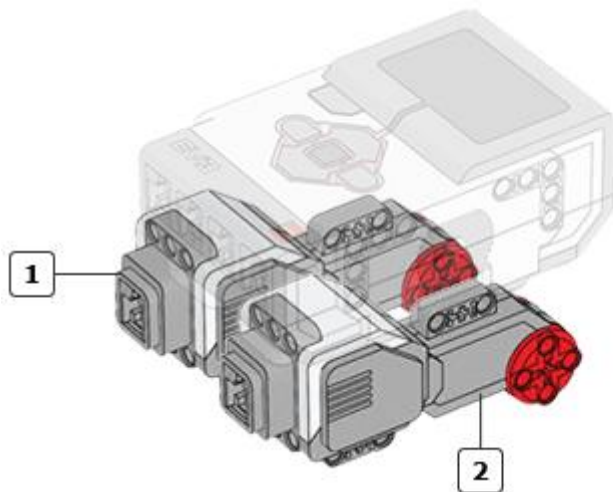
Mit den Eingaben des Bewegungslenkungs-Block lässt sich im Detail steuern, wie sich die Motoren verhalten sollen. Du kannst die Eingabewerte direkt in den Block eingeben. Alternativ hierzu können als Eingabewerte auch die Ausgaben anderer Programmierblöcke dienen, die über Datenleitungen übertragen werden.

Eingabe	Typ	Zulässige Werte	Hinweise
Leistung	Numerische Werte	-100 bis 100	Motor-Leistungsstufe
Lenkung	Numerische Werte	-100 bis 100	Lenkrichtung.
Am Ende bremsen	Logische Werte	Wahr/Falsch	Kommt am Ende des Blocks zum Tragen. Wenn „Wahr“, werden die Motoren sofort gestoppt und in ihrer Stellung gehalten. Wenn „Falsch“, wird die Stromversorgung der Motoren ausgeschaltet und die Motoren dürfen noch auslaufen.
Sekunden	Numerische Werte	≥ 0	Bewegungszeit in Sekunden.
Grad	Numerische Werte	Beliebige Zahl	Betrag der Bewegung in Grad. 360 Grad bedeuten eine volle Umdrehung.
Umdrehungen	Numerische Werte	Beliebige Zahl	Betrag der Bewegung in Umdrehungen.

12.4 HEBELLENKUNGS-BLOCK

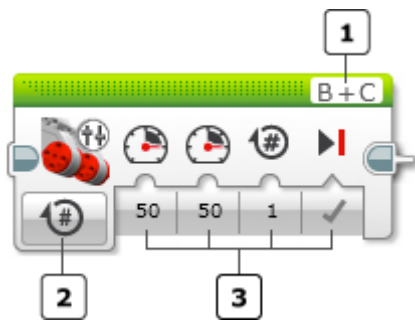


Mit dem Hebelblock kannst du einen Roboter vorwärts und rückwärts fahren, wenden oder anhalten lassen. Benutze den Hebelblock für Roboterfahrzeuge mit zwei Großen Motoren, bei denen einer die linke Seite und der andere die rechte Seite des Fahrzeugs antreibt. Du kannst die beiden Motoren mit unterschiedlichen Geschwindigkeiten fahren oder in unterschiedliche Richtungen drehen lassen, um deinen Roboter wenden zu lassen.



- 1** Linker Motor
- 2** Rechter Motor

12.4.1 WÄHLE DEINE MOTOR-ANSCHLÜSSE UND DEN STEUERMODUS



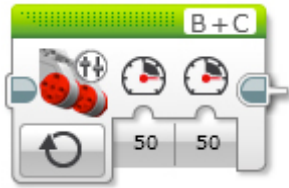
- 1** Anschlussauswahl
- 2** Modus-Auswahl
- 3** Eingaben

Wähle die beiden Motoren (A, B, C oder D), die vom Hebellenkungs-Block gesteuert werden sollen, indem du die Anschlussauswahl oben im Block verwendest. Klicke auf den jeweiligen Buchstaben des Motors, um den entsprechenden Anschluss für diesen Motor auszuwählen. An den ersten Anschluss sollte Motor auf der linken Seite des Fahrzeugs angeschlossen werden und an den zweiten Anschluss sollte der Motor auf der rechten Seite des Fahrzeugs angeschlossen werden.

Benutze die Modus-Auswahl, um festzulegen, wie die Motoren gesteuert werden sollen. Nach der Auswahl des Modus kannst du die Werte für die Eingaben festlegen.

12.4.2 MODI

An



Im Modus „An“ werden beide Motoren angeschaltet und sofort im Anschluss wird mit dem nächsten Block im Programm weitergemacht. Drehzahl und Drehrichtung der Motoren lassen sich mithilfe der Eingaben Leistung links und Leistung rechts steuern. Die Motoren laufen, bis sie von einem späteren Block im Programm gestoppt werden oder einen anderen Befehl erhalten bzw. bis das Programm endet.

Beispiel



Benutze den Modus „An“, wenn andere Blöcke in deinem Programm steuern sollen, wie lange die Motoren angeschaltet bleiben. In diesem Programm fährt der Roboter in gerader Linie vorwärts. Wenn der Berührungssensor gedrückt wird, stoppt der Roboter.

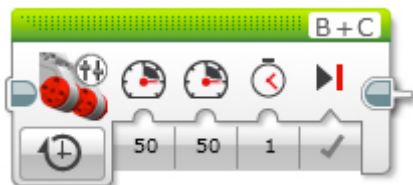
Aus



Mit dem Modus „Aus“ werden beide Motoren ausgeschaltet. Benutze den Modus „Aus“, um einen Roboter zu stoppen, der an früherer Stelle im Programm durch den Modus An gestartet wurde. Siehe Beispiel oben.

Wenn für die Eingabe Am Ende bremsen der Wert „Wahr“ ausgewählt wurde, werden die Motoren sofort gestoppt. Die Motoren werden in der gestoppten Stellung gehalten, bis ein anderer Bewegungs- oder Motor-Block sie wieder startet oder das Programm endet. Wenn für die Eingabe Am Ende bremsen der Wert „Falsch“ ausgewählt wird, dann wird einfach die Stromversorgung der Motoren abgeschaltet. Die Motoren laufen aus und nutzen dabei jeglichen verbliebenen Schwung, bis sie stoppen oder bis ein anderer Bewegungs- oder Motor-Block startet.

An für n Sekunden



Im Modus „An für n Sekunden“ werden beide Motoren genau für die Anzahl an Sekunden angeschaltet, die in der Eingabe Sekunden festgelegt wurde, und dann werden sie wieder ausgeschaltet. Der Block wartet, bis diese eingestellte Zeit verstrichen ist, bevor das Programm mit dem nächsten Block weitermacht.

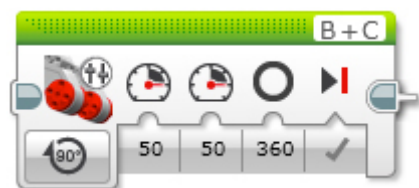
Die Fahrgeschwindigkeit und die Fahrtrichtung deines Roboters lassen sich mithilfe der Eingaben Leistung links und Leistung rechts steuern. Verwende die Eingabe Am Ende bremsen, um deinen Roboter exakt nach der festgelegten Anzahl an Sekunden zu stoppen.

Beispiel



Dieses Programm lässt deinen Roboter mit voller Leistung 2 Sekunden in gerader Linie vorwärts fahren und dann anhalten.

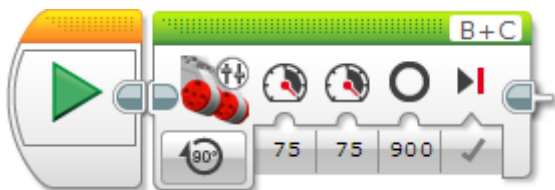
An für n Grad



Im Modus „An für n Grad“ werden beide Motoren angeschaltet. Dann wird gewartet, bis einer der Motoren die in der Eingabe Grad über die eingestellte Gradzahl definierte Umdrehung vollführt hat, und anschließend werden beide Motoren ausgeschaltet. Dieser Modus kann benutzt werden, um deinen Roboter eine bestimmte Distanz zurücklegen oder um einen bestimmten Betrag (eine bestimmte Umdrehungszahl) drehen zu lassen. 360 Grad entsprechen einer vollen Umdrehung des Motors.

Die Fahrgeschwindigkeit und die Fahrtrichtung deines Roboters lassen sich mithilfe der Eingaben Leistung links und Leistung rechts steuern. Verwende die Eingabe Am Ende bremsen, um deinen Roboter exakt nach der festgelegten Anzahl an Sekunden zu stoppen.

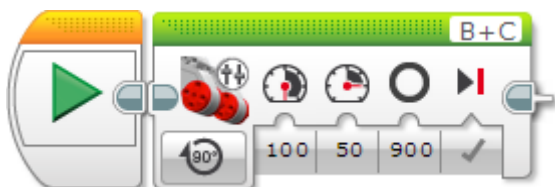
Beispiel 1



Dieses Programm lässt deinen Roboter mit 75 % Leistung in gerader Linie vorwärts fahren, bis sich die Motoren um 900 Grad gedreht (zweieinhalb Umdrehungen vollführt) haben, und lässt ihn dann stoppen.

Beispiel 2

Das nachstehende Programm lässt einen Roboter in einem Bogen nach rechts fahren, weil die Leistung links größer als die Leistung rechts ist. Diese Kurvenfahrt wird fortgesetzt, bis sich der linke (äußere und schnellere) Motor genau um 900 Grad gedreht hat, und dann stoppt der Roboter.

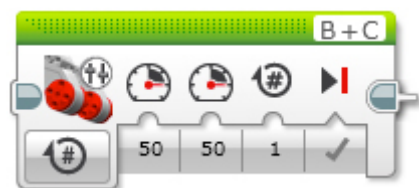


Beispiel 3



Dieses Programm lässt einen Roboter auf der Stelle wenden, indem die beiden Motoren in unterschiedlichen Drehrichtungen laufen gelassen werden. Durch Verwendung einer negativen Zahl in der Eingabe Leistung rechts dreht sich der rechte Motor rückwärts. Der Roboter dreht sich, bis sich der linke Motor um 250 Grad vorwärts gedreht hat, dann stoppt der Roboter. Der rechte Motor dreht sich 250 Grad rückwärts.

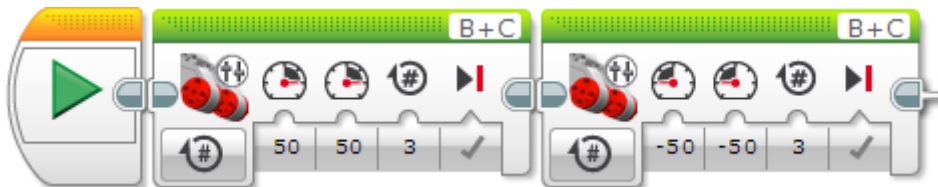
An für n Umdrehungen



Im Modus „An für n Umdrehungen“ werden beide Motoren angeschaltet. Dann wird gewartet, bis einer der Motoren die in der Eingabe Umdrehungen anhand der eingestellten Umdrehungszahl definierte Umdrehung vollführt hat, und anschließend werden beide Motoren ausgeschaltet.

Die Fahrgeschwindigkeit und die Fahrtrichtung deines Roboters lassen sich mithilfe der Eingaben Leistung links und Leistung rechts steuern. Verwende die Eingabe Am Ende bremsen, um deinen Roboter exakt nach der festgelegten Anzahl an Sekunden zu stoppen.

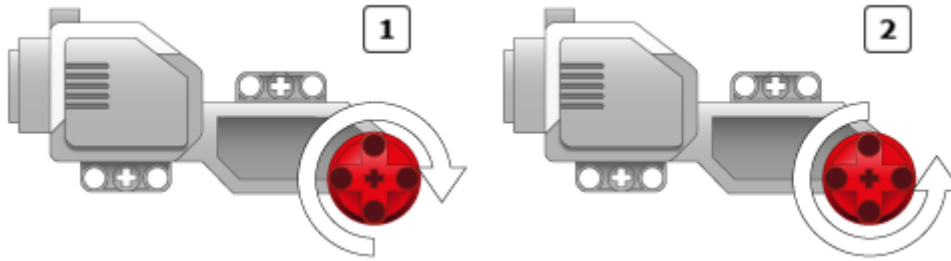
Beispiel



Dieses Programm lässt deinen Roboter mit 50 % Leistung in gerader Linie vorwärts fahren, bis die Motoren 3 volle Umdrehungen vollführt haben. Dann wird er in gerader Linie 3 Umdrehungen rückwärtsfahren (mithilfe eines negativen Wertes für die Leistung) und dort stoppen, wo er ursprünglich losgefahren war.

12.4.3 LEISTUNG UND DREHRICHTUNG DES MOTORS

Für die Eingaben Leistung links und Leistung rechts ist eine Zahl zwischen -100 und 100 zulässig. Durch Eingabe positiver und negativer Zahlen lässt sich die Drehrichtung des Großen Motors festlegen (siehe nachstehende Grafik).



1 Positive Leistung

2 Negative Leistung

12.4.4 EINGABEN

Mit den Eingaben des Hebellenkungs-Blocks lässt sich im Detail steuern, wie sich die Motoren verhalten sollen. Du kannst die Eingabewerte direkt in den Block eingeben. Alternativ hierzu können als Eingabewerte auch die Ausgaben anderer Programmierblöcke dienen, die über Datenleitungen übertragen werden. Die verfügbaren Eingaben und deren Funktionen sind vom ausgewählten Steuermodus abhängig.

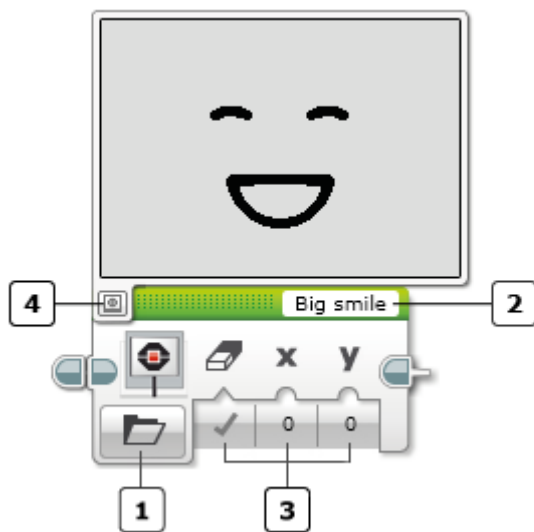
Eingabe	Typ	Zulässige Werte	Hinweise
Leistung links	Numerische Werte	-100 bis 100	Motorleistungsstufe für den linken Motor.
Leistung rechts	Numerische Werte	-100 bis 100	Motorleistungsstufe für den rechten Motor.
Am Ende bremsen	Logische Werte	Wahr/Falsch	Kommt am Ende des Blocks zum Tragen. Wenn „Wahr“, werden die Motoren sofort gestoppt und in ihrer Stellung gehalten. Wenn „Falsch“, wird die Stromversorgung der Motoren ausgeschaltet und die Motoren dürfen noch auslaufen.
Sekunden	Numerische Werte	≥ 0	Bewegungszeit in Sekunden.
Grad	Numerische Werte	Beliebige Zahl	Betrag der Bewegung in Grad. 360 Grad bedeuten eine volle Umdrehung.
Umdrehungen	Numerische Werte	Beliebige Zahl	Betrag der Bewegung in Umdrehungen.

12.5 ANZEIGE



Mit dem Anzeige-Block lassen sich auf dem Display des EV3-Steins Texte oder Grafiken anzeigen.

12.5.1 WÄHLE DEN MODUS „ANZEIGE“



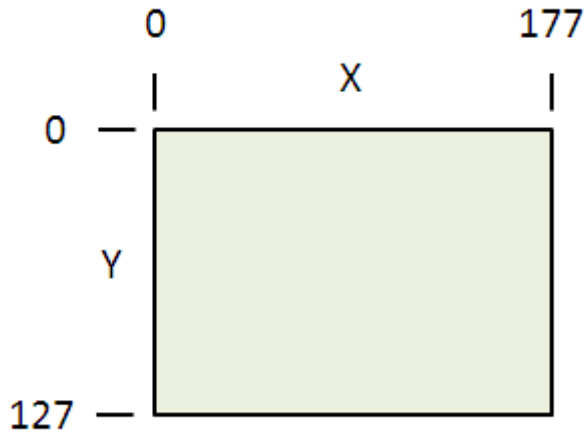
- 1 Modus-Auswahl
- 2 Block-Textfeld
- 3 Eingaben
- 4 Schaltfläche „Vorschau anzeigen“

Wähle die Text- oder Grafikarten aus, die mithilfe der Modus-Auswahl angezeigt werden sollen. Nach der Auswahl des Modus kannst du die Werte für die Eingaben festlegen.

Du kannst auf die Schaltfläche „Vorschau anzeigen“ klicken, um eine Vorschau dessen einzublenden, was der Anzeige-Block auf dem Display des EV3-Steins anzeigen wird. Während du die Eingabewerte für den Block auswählst, lässt du die Vorschau geöffnet.

12.5.2 ANZEIGEKOORDINATEN

In vielen Modi des Anzeige-Blocks werden x- und y-Koordinaten verwendet, um anzugeben, wo ein Objekt gezeichnet werden soll. Die Koordinaten geben die Pixelposition auf dem Display des EV3-Steins an. Die Position (0/0) befindet sich in der oberen linken Ecke des Displays (siehe nachstehendes Bild).



Das Display ist 178 Pixel breit und 128 Pixel hoch. Die x-Koordinatenwerte reichen von 0 (ganz links außen auf dem Display) bis 177 (ganz rechts außen). Die y-Koordinatenwerte reichen von 0 (ganz oben auf dem Display) bis 127 (ganz unten).

Tipps und Tricks

Du kannst die Schaltfläche „Vorschau anzeigen“ in der oberen linken Ecke des Anzeige-Blocks verwenden, um dir die Suche der richtigen Anzeigekoordinaten zu erleichtern.

12.5.3 MODI

Text – Pixel



Im Modus „Text – Pixel“ kann Text an jeder beliebigen Position auf dem Display des EV3-Steins eingeblendet werden.

Wenn für Bildschirm löschen der Wert „Wahr“ gewählt wurde, dann wird die Anzeige vor dem Eintragen des Textes gelöscht.

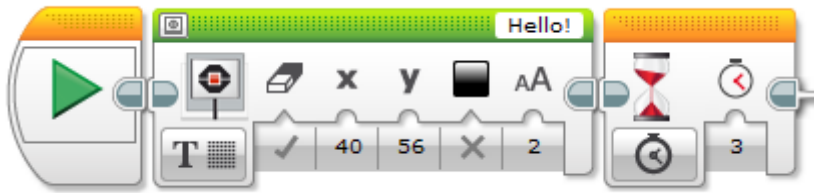
Die Eingabe Text befindet sich oben im Block. Sie enthält die anzuzeigende Textzeichenfolge. Wähle die Option „Leitung“, um eine Eingabe für Text einzublenken, in die Daten per Datenleitung übertragen werden können. Der Text kann jegliche

Die Eingaben X und Y geben die Anzeigekoordinaten des Textanfangs an, der sich in der oberen linken Ecke des angezeigten Textes befindet.

Verwende die Eingabe Farbe, um die Textfarbe zu wählen. Wenn du Schwarz auswählst, wird der Text schwarz auf weißem Hintergrund angezeigt. Wenn du Weiß auswählst, wird der Text weiß auf schwarzem Hintergrund angezeigt.

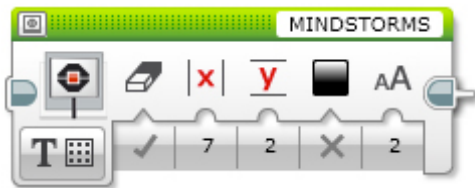
Du kannst die Eingabe Schriftart verwenden, um eines von drei verschiedenen Schriftbildern auszuwählen: Normal, Fett und Groß.

Beispiel



Mit diesem Programm wird der Schriftzug „Hello!“ in der Schriftart „Groß“ in der Mitte des Displays angezeigt.

Text – Raster



Im Modus „Text – Raster“ wird der Text an einem Raster aus Zeilen und Spalten ausgerichtet und angezeigt. Das erleichtert die Anzeige und Ausrichtung mehrerer Textzeilen. Die Höhe einer Zeile reicht, um eine Textzeile in den Schriftarten „Normal“ oder „Groß“ anzuzeigen. Und jedes Zeichen in diesen Schriftarten ist genau eine Spalte breit.

Wenn für Bildschirm löschen der Wert „Wahr“ gewählt wurde, dann wird die Anzeige vor dem Eintragen des Textes gelöscht.

Die Eingabe Text befindet sich oben im Block. Sie enthält die anzuzeigende Textzeichenfolge. Wähle die Option „Leitung“, um eine Eingabe für Text einzublenden, in die Daten per Datenleitung übertragen werden können.

Mit der Eingabe Spalte wird die Anfangsspalte (horizontale Position) des Textes festgelegt. Jede Spalte ist 8 Pixel breit. Die Spalten sind von links nach rechts von 0 bis 21 durchnummeriert.

Mit der Eingabe Zeile wird die Zeile oder Zeilennummer (vertikale Position) des Textes festgelegt. Jede Zeile ist 10 Pixel hoch. Die Zeile 0 befindet sich ganz oben auf dem Display, die Zeile 11 dagegen befindet sich am unteren Rand des Displays.

Verwende die Eingabe Farbe, um die Textfarbe zu wählen. Wenn du Schwarz auswählst, wird der Text schwarz auf weißem Hintergrund angezeigt. Wenn du Weiß auswählst, wird der Text weiß auf schwarzem Hintergrund angezeigt.

Du kannst die Eingabe Schriftart verwenden, um eines von drei verschiedenen Schriftbildern auszuwählen: Normal, Fett und Groß.

Beispiel



Dieses Programm zeigt zwei Textzeilen an. Der zweite Anzeige-Block verwendet für die Eingabe Bildschirm löschen den Wert „Falsch“, um eine zweite Zeile hinzuzufügen, ohne die erste Zeile zu löschen.

Formen – Linie

Im Modus „Formen – Linie“ wird zwischen zwei beliebigen Punkten auf dem Display eine gerade Linie gezeichnet.

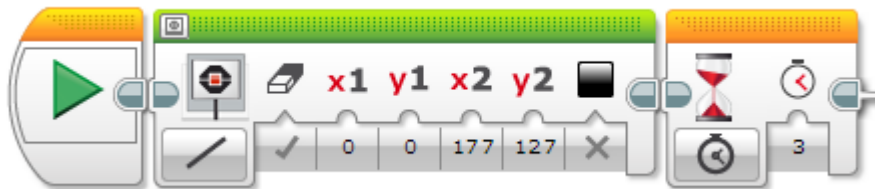
Wenn für Bildschirm löschen der Wert „Wahr“ gewählt wurde, dann wird die Anzeige vor dem Einzeichnen der Linie gelöscht.

Die Eingaben X1 und Y1 legen die Anzeigekoordinaten des Ausgangspunktes der Linie fest.

Die Eingaben X2 und Y2 legen die Koordinaten des Endpunktes fest.

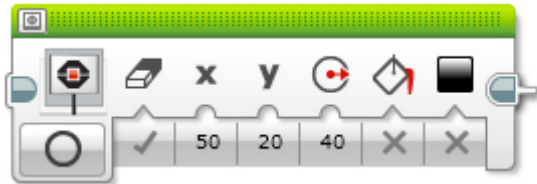
Du kannst die Eingabe Farbe verwenden, um eine schwarze oder weiße Linie zu wählen. Eine weiße Linie ist nur sichtbar, wenn für Bildschirm löschen der Wert „Falsch“ gewählt wurde und die Linie über einen Bereich verläuft, in dem ein vorangegangener Anzeige-Block etwas Schwarzes angezeigt hat.

Beispiel



Dieses Programm zieht eine diagonale Linie quer über das Display des EV3-Steins.

Formen – Kreis



Im Modus „Formen – Kreis“ wird ein Kreis auf dem Display eingezeichnet.

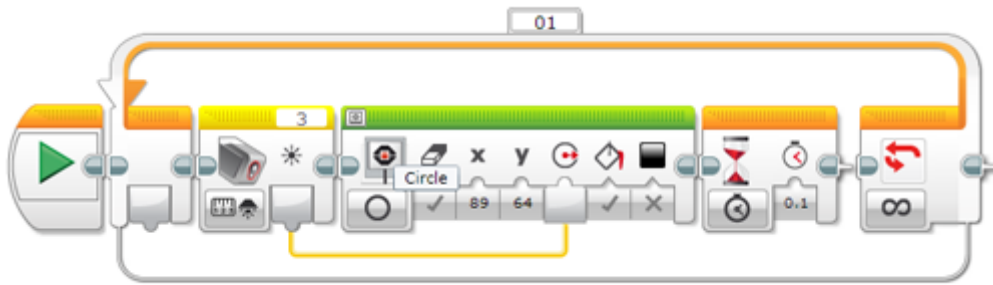
Wenn für Bildschirm löschen der Wert „Wahr“ gewählt wurde, dann wird die Anzeige vor dem Einzeichnen des Kreises gelöscht.

Mit den Eingaben X und Y werden die Anzeigekoordinaten des Kreismittelpunktes festgelegt. Mit der Eingabe Radius wird der Radius des Kreises in Pixeln angegeben.

Wenn für Ausfüllen der Wert „Wahr“ gewählt wurde, dann wird die Innenfläche des Kreises ausgefüllt. Wenn für Ausfüllen der Wert „Falsch“ gewählt wurde, dann wird nur der Umriss des Kreises eingezeichnet.

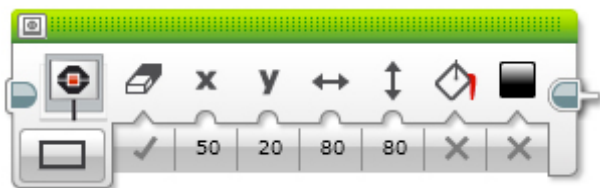
Du kannst die Eingabe Farbe verwenden, um einen schwarzen oder weißen Kreis zu wählen. Ein weißer Kreis ist nur sichtbar, wenn für Bildschirm löschen der Wert „Falsch“ gewählt wurde und der Kreis in einem Bereich liegt, in dem ein vorangegangener Anzeige-Block etwas Schwarzes angezeigt hat.

Beispiel



Dieses Programm zeigt ständig einen ausgefüllten Kreis an, dessen Radius dem vom Farbsensor gemessenen Wert „Stärke des Umgebungslichts“ entspricht. Diese Anzeige wird im Zehntelsekundentakt aktualisiert.

Formen – Rechteck



Im Modus „Formen – Rechteck“ wird ein Rechteck auf dem Display eingezeichnet.

Wenn für Bildschirm löschen der Wert „Wahr“ gewählt wurde, dann wird die Anzeige vor dem Einzeichnen des Rechtecks gelöscht.

Mit den Eingaben X und Y werden die Anzeigekoordinaten der oberen linken Ecke des Rechtecks festgelegt. Mit den Eingaben Breite und Höhe wird die Größe des Rechtecks in Pixeln festgelegt.

Wenn für Ausfüllen der Wert „Wahr“ gewählt wurde, dann wird die Innenfläche des Rechtecks ausgefüllt. Wenn für Ausfüllen der Wert „Falsch“ gewählt wurde, dann wird nur der Umriss des Rechtecks eingezeichnet.

Du kannst die Eingabe Farbe verwenden, um ein schwarzes oder weißes Rechteck zu wählen. Ein weißes Rechteck ist nur dann sichtbar, wenn für Bildschirm löschen der Wert „Falsch“ gewählt wurde und das Rechteck in einem Bereich liegt, in dem ein vorangegangener Anzeige-Block etwas Schwarzes angezeigt hat.

Beispiel

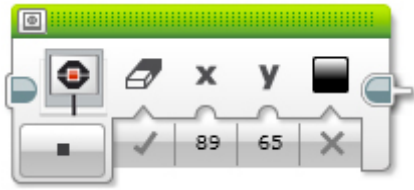


Dieses Programm füllt das ganze Display schwarz aus, indem es ein großes ausgefülltes Rechteck anzeigt. Dann zeichnet es darüber den Umriss eines kleineren weißen Rechtecks ein.

Tipps und Tricks

Du kannst ein ausgefülltes weißes Rechteck verwenden, um einen Teil des Displays zu löschen, ohne die anderen Bereiche des Displays zu beeinträchtigen.

Formen – Punkt



Im Modus „Formen – Punkt“ wird ein einziges Pixel auf dem Display eingezeichnet.

Wenn für Bildschirm löschen der Wert „Wahr“ gewählt wurde, dann wird die Anzeige vor dem Einzeichnen des Pixels gelöscht.

Mit den Eingaben X und Y werden die Anzeigekoordinaten des Pixels festgelegt.

Du kannst die Eingabe Farbe verwenden, um das Pixel in Schwarz oder Weiß einzuzichnen. Durch Einzeichnen weißer Pixel lassen sich schwarze Pixel löschen, die von vorangegangenen Anzeige-Blöcken hinterlassen wurden.

Bild



Im Modus „Bild“ wird eine grafische Bild-Datei gezeichnet.

Die Eingabe Datei-Name befindet sich oben im Block. Dort kannst du deine Auswahl aus einer Liste mit Bild-Dateien treffen. Bei den Bildern in der Kategorie „LEGO Bilder“ handelt es sich um die im Lieferumfang der EV3-Software enthaltenen Bild-Dateien. Die Kategorie „Projekt-Bilder“ enthält die Bilder, die du bereits in deinem Projekt verwendet hast.

Wenn für Bildschirm löschen der Wert „Wahr“ gewählt wurde, dann wird die Anzeige vor dem Einzeichnen des Bildes gelöscht.

Mit den Eingaben X und Y werden die Anzeigekoordinaten der oberen linken Ecke des Bildes festgelegt. Das Bild wird von diesem Punkt aus nach unten und rechts eingezeichnet. Verwende X = 0 und Y = 0 für Bild-Dateien, die das ganze Display ausfüllen sollen.

Bildschirm zurücksetzen



Mit dem Modus „Bildschirm zurücksetzen“ wird das Display des EV3-Steins wieder in den normalen Info-Bildschirm verwandelt, der angezeigt wird, wenn ein Programm ausgeführt wird.

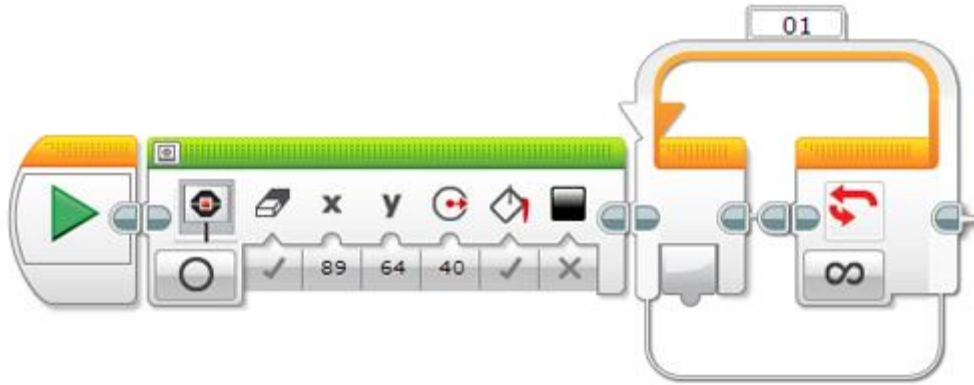
12.5.4 ANGEZEIGTE OBJEKTE SICHTBAR LASSEN

Nach Abschluss eines EV3-Programms wird die Anzeige auf dem EV3-Stein gelöscht. Stattdessen wird wieder die Menü-Anzeige des EV3-Steins eingeblendet. Somit werden jegliche Texte oder Grafiken gelöscht, die vom abgeschlossenen Programm angezeigt wurden. Wenn dein Programm bspw.

einen einzigen Anzeige-Block enthält, dem kein weiterer Block folgt, dann wird die Anzeige bei Abschluss des Programms so schnell gelöscht, dass du keine Gelegenheit hast, dir die Ergebnisse deines Anzeige-Blocks anzusehen.

Wenn du möchtest, dass am Ende des Programms noch eine Anzeige eingeblendet bleibt, musst du einen Block am Ende des Programms hinzufügen, damit das Programm nicht sofort beendet wird

Beispiel



Bei diesem Programm wird ein Schleifen-Block am Ende des Programms verwendet, um das Programm endlos laufen zu lassen. Durch diese Schleife bleibt der Kreis eingblendet, bis du das Programm stoppst, indem du die Taste „Zurück“ auf dem EV3-Stein drückst.

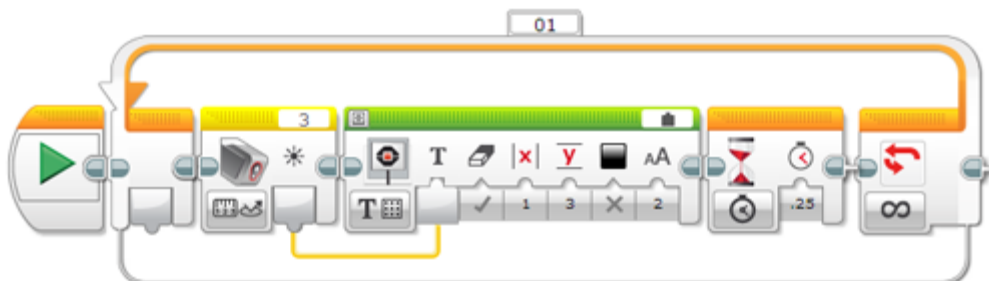
12.5.5 ANZEIGE MEHRERER OBJEKTE

Wenn du mehrere Text- oder Grafikobjekte gleichzeitig auf dem Display anzeigen möchtest, darf die Anzeige des EV3-Steins zwischen den Objekten auf keinen Fall gelöscht werden. Um mehrere Objekte auf dem Display anzeigen zu können, musst du bei jedem Anzeige-Block – mit Ausnahme des ersten Blocks – für die Eingabe Bildschirm löschen den Wert „Falsch“ wählen.

12.5.6 ZAHLEN ANZEIGEN

Du kannst den Wert einer Zahl in deinem Programm anzeigen, indem du eine Datenleitung an die Eingabe Text eines Textanzeige-Blocks anschließt. Eine numerische Datenleitung wird durch Konvertierung des Datenleitungs-Typs automatisch in Text umgewandelt.

Beispiel



Mit diesem Programm wird ständig der vom Farbsensor gemessene Wert „Stärke des reflektierten Lichts“ auf dem Display angezeigt, wobei die Anzeige im Viertelsekundentakt aktualisiert wird.

12.5.7 EINGABEN

Mit den Eingaben im Anzeige-Block werden die anzuzeigenden Texte oder Grafiken sowie deren Darstellungsweise festgelegt. Du kannst die Eingabewerte direkt in den Block eingeben. Alternativ hierzu können als Eingabewerte auch die Ausgaben anderer Programmierblöcke dienen, die über Datenleitungen übertragen werden.

Eingabe	Typ	Zulässige Werte	Hinweise
Bildschirm löschen	Logische Werte	Wahr/Falsch	Wenn „Wahr“, wird die Anzeige gelöscht, bevor das Objekt eingezeichnet wird.
Datei-Name	Text	Name einer bestehenden Bild-Datei	Wähle in der Eingabe „Datei-Name“ oben im Block.
Text	Text	Zeichen aus dem Zeichensatz	Anzuzeigender Text.
Spalte	Numerische Werte	0 bis 21	Nummer der Startspalte (horizontal von links nach rechts). Die Spalten sind 8 Pixel breit. Die Spalte 0 befindet sich ganz links auf dem Display.
Zeile	Numerische Werte	0 bis 11	Nummer der Textzeile (vertikal von oben nach unten) Die Zeilen sind 10 Pixel hoch. Die Zeile 0 befindet sich ganz oben auf dem Display.
X	Numerische Werte	0 bis 177	Die X-Koordinate 0 befindet sich ganz links auf dem Display.
Y	Numerische Werte	0 bis 127	Die Y-Koordinate 0 befindet sich ganz unten auf dem Display.
X1	Numerische Werte	0 bis 177	Die X-Koordinate des Ausgangspunktes einer Linie.
Y1	Numerische Werte	0 bis 127	Die Y-Koordinate des Ausgangspunktes einer Linie.
X2	Numerische Werte	0 bis 177	Die X-Koordinate des Endpunktes einer Linie.

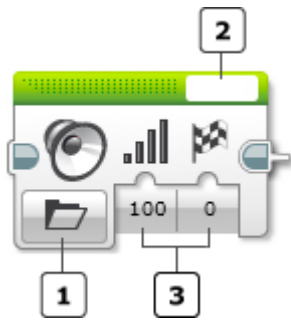
Y2	Numerische Werte	0 bis 127	Die Y-Koordinate des Endpunktes einer Linie.
Radius	Numerische Werte	≥ 0	Radius eines Kreises in Pixeln
Breite	Numerische Werte	≥ 0	Breite eines Rechtecks in Pixeln
Höhe	Numerische Werte	≥ 0	Höhe eines Rechtecks in Pixeln
Ausfüllen	Logische Werte	Wahr/Falsch	Wenn „Wahr“, wird die Innenfläche der Form ausgefüllt.
Farbe	Logische Werte	Falsch = Schwarz Wahr = Weiß	Wird Text in weißer Schrift eingetragen, dann werden die Zeichen mit einem schwarzen Hintergrund unterlegt. Durch Zeichnen von Formen in weißer Farbe werden schwarze Pixel vom Display gelöscht.
Schriftart	Numerische Werte	0 = Normal 1 = Fett 2 = Groß	In der Schriftart „Normal“ sind die Zeichen 9 Pixel hoch und 8 Pixel breit. In der Schriftart „Fett“ sind die Zeichen 8 Pixel hoch und 8 Pixel breit. In der Schriftart „Groß“ sind die Zeichen 16 Pixel hoch und 16 Pixel breit.

12.6 KLANG



Der Klang-Block erzeugt mithilfe des in den EV3-Steins integrierten Lautsprechers Geräusche. Du kannst vorab aufgezeichnete Klang-Dateien abspielen oder eine musikalische Note bzw. einen Ton festlegen.

12.6.1 WÄHLE DEN KLANGMODUS



- 1 Modus-Auswahl
- 2 Eingabe „Datei-Name“
- 3 Eingaben

Wähle die zu erzeugende Klangart mithilfe der Modus-Auswahl aus. Nach der Auswahl des Modus kannst du die Werte für die Eingaben festlegen.

12.6.2 MODI

Datei abspielen

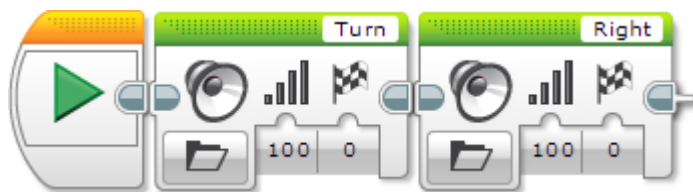
Im Modus „Datei abspielen“ wird eine vorab aufgezeichnete Klang-Datei wiedergegeben.

Die Eingabe Datei-Name befindet sich oben im Block. Dort kannst du aus einer Liste mit Klang-Effekten sowie mit gesprochenen Wörtern und Sätzen deine Auswahl treffen. Bei den Klängen in der Kategorie „LEGO Geräusche“ handelt es sich um die im Lieferumfang der EV3-Software enthaltenen Klang-Dateien. Die Kategorie „Projekt-Klänge“ enthält die Klänge, die du bereits in deinem Projekt verwendet hast.

Die Eingabe Lautstärke regelt, wie laut der Klang abgespielt wird.

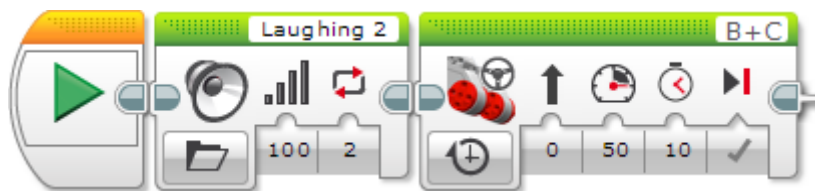
Die Eingabe Wiedergabeart regelt, ob der Block wartet, bis der Klang vollständig abgespielt wurde, bevor das Programm mit dem nächsten Block weitermacht, und ob der Klang wiederholt wird.

Beispiel 1



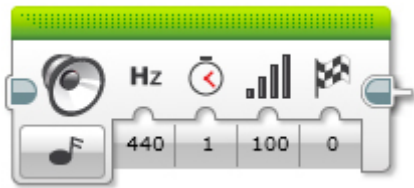
Dieses Programm lässt den EV3-Stein „Turn Right“ (Rechts abbiegen) sagen, indem es zwei verschiedene Klang-Dateien abspielt und wartet, bis beide vollständig wiedergegeben wurden.

Beispiel 2



Dieses Programm erzeugt ein ständiges Lachen, während der Roboter 10 Sekunden vorwärts fährt. Dies wird durch die Option „Wiederholen“ in der Eingabe Wiedergabeart bewerkstelligt.

Ton abspielen



Im Modus „Ton abspielen“ wird der Ton einer bestimmten Frequenz abgespielt. Die Frequenz des Tons regelt die Tonlage, d. h. sie bestimmt, wie hoch oder tief der Ton ist.

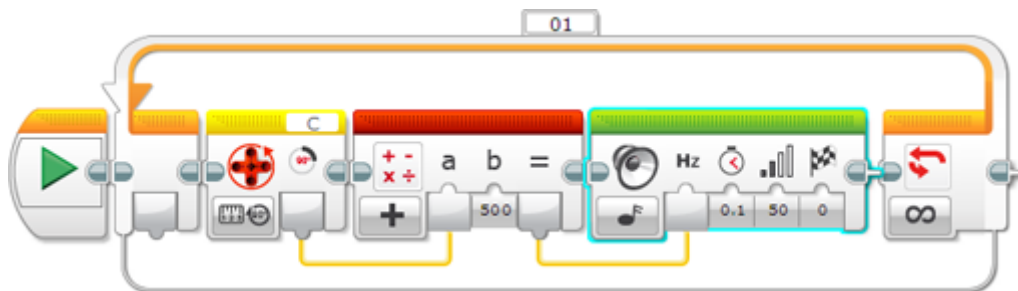
Die Eingabe Frequenz bestimmt die Frequenz des Tons in Hz (Schwingungen pro Sekunde). Du kannst eine Zahl als Frequenz eingeben oder eine Frequenz aus der Liste der Standard-Musiknoten auswählen.

Die Eingabe Dauer regelt die Abspieldauer des Tons (in Sekunden).

Die Eingabe [Lautstärke](#) regelt, wie laut der Klang abgespielt wird.

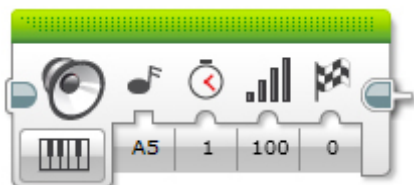
Die Eingabe [Wiedergabeart](#) regelt, ob der Block wartet, bis der Klang vollständig abgespielt wurde, bevor das Programm mit dem nächsten Block weitermacht, und ob der Klang wiederholt wird.

Beispiel



Dieses Programm lässt einen Ton in seiner Frequenz variieren – und zwar in Abhängigkeit von der Stellung des Motorumdrehungssensors. Wenn du den Motor mit der Hand drehst, ändert sich der Ton.

Note abspielen



Im Modus „Note abspielen“ wird eine Musiknote wiedergegeben.

Die Eingabe Note ermöglicht dir die Auswahl der Note auf einer Klaviertastatur.

Die Eingabe Dauer regelt die Abspieldauer der Note (in Sekunden).

Die Eingabe Lautstärke regelt, wie laut der Klang abgespielt wird.

Die Eingabe Wiedergabeart regelt, ob der Block wartet, bis der Klang vollständig abgespielt wurde, bevor das Programm mit dem nächsten Block weitermacht, und ob der Klang wiederholt wird.

Beispiel



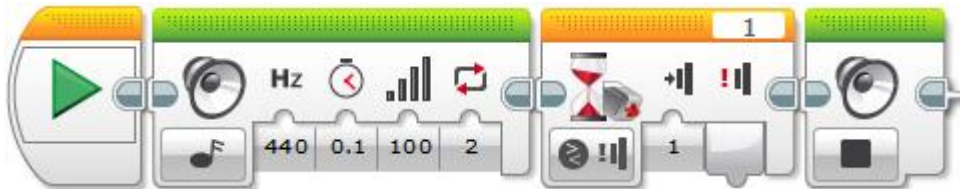
Dieses Programm spielt eine kurze Melodie ab, indem es drei verschiedene Noten wiedergibt.

Stopp



Mit dem Stoppmodus wird jeder aktuell vom EV3-Stein abgespielte Klang gestoppt.

Beispiel



Dieses Programm spielt einen Ton ab, bis ein Berührungssensor gedrückt wird, und stoppt daraufhin den Ton.

12.6.3 EINGABEN

Die Eingaben des Klang-Blocks regeln die Details beim Abspielen des Klangs. Du kannst die Eingabewerte direkt in den Block eingeben. Alternativ hierzu können als Eingabewerte auch die Ausgaben anderer Programmierblöcke dienen, die über Datenleitungen übertragen werden.

Eingabe	Typ	Zulässige Werte	Hinweise
Datei-Name	Text	Name einer bestehenden Klang-Datei	z. B. „Gelächter 2“
Hinweis	Text	C bis B, gefolgt von „#“ (optional), gefolgt von 4 bis 6.	A bis G sind Notennamen. 4 bis 6 bezeichnen Oktaven. „#“ steht für die Erhöhung um einen Halbton („-is“) Beispiele: „C4“ ist auf einem Standardklavier das „mittlere C“ und „C#4“ liegt einen Halbton höher.
Frequenz	Zahl	300 bis 10000	Tonfrequenz in Hz

Dauer	Zahl	≥ 0	Dauer einer Note oder eines Tons in Sekunden
Lautstärke	Zahl	0 bis 100	Ein Prozentwert der vollen Lautstärke
Wiedergabeart	Zahl	0, 1 oder 2	<p>0 = Warten auf Abschluss: Der Klang wird einmal abgespielt und das Programm wartet, bis der Klang vollständig abgespielt wurde, bevor es fortgesetzt wird.</p> <p>1 = Einmal abspielen: Der Klang wird einmal abgespielt und das Programm wird sofort fortgesetzt.</p> <p>2 = Wiederholen: Der Klang wird immer wieder abgespielt, bis ein anderer Klang-Block ausgeführt wird, und das Programm wird sofort fortgesetzt.</p>

13 PROGRAMMIERABLAUF-BLÖCKE

13.1 START



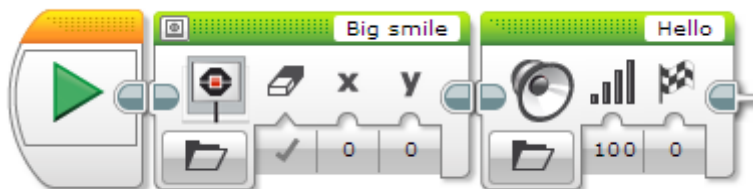
Der Start-Block markiert den Anfang einer Programmierblock-Sequenz in deinem Programm. Dein Programm kann mehr als eine Programmierblock-Sequenz enthalten. Alle Programmierblock-Sequenzen mit einem Start-Block werden bei der Ausführung eines Programms automatisch gestartet und gleichzeitig ausgeführt.

Wenn eine Programmierblock-Sequenz nicht mit einem Start-Block beginnt, werden die Blöcke nicht ausgeführt.

Tipps und Tricks

Durch Klicken auf den grünen Pfeil auf dem Start-Block wird das gesamte Programm kompiliert und auf deinen EV3-Stein heruntergeladen, allerdings wird nur die ausgewählte Sequenz ausgeführt.

Beispiel 1: Starten einer einzelnen Sequenz

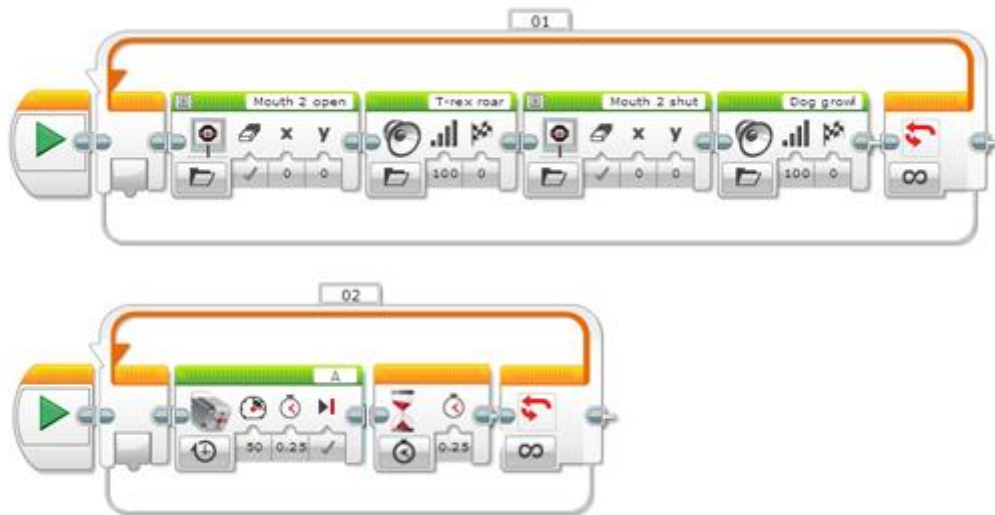


Dieses Programm enthält eine einzelne Block-Sequenz, die mit einem Start-Block beginnt.

Es gibt drei verschiedene Möglichkeiten, dieses Programm auf deinem EV3-Stein auszuführen:

1. Klicke auf die Schaltfläche „Herunterladen“ in der Software und wähle dann das Programm auf dem EV3-Stein aus, um es auszuführen.
2. Klicke in der Software auf die Schaltfläche „Herunterladen und ausführen“. Das Programm wird sofort heruntergeladen und ausgeführt.
3. Klicke auf den grünen Pfeil auf dem Start-Block. Das Programm wird heruntergeladen und die Sequenz wird sofort ausgeführt.

Beispiel 2: Zwei Block-Sequenzen



In diesem Programm werden zwei verschiedene Blocksequenzen verwendet, um zwei verschiedenartige Aktionen gleichzeitig ablaufen zu lassen. In der ersten Sequenz lässt eine Schleife den EV3-Stein zwischen zwei Klängen und zwei Bildern wechseln. Die zweite Sequenz lässt einen Mittleren Motor abwechselnd ¼ Sekunde laufen und dann eine ¼ Sekunde stoppen.

Wenn du im oben beschriebenen Programm auf den grünen Pfeil auf einem der beiden Start-Blöcke klickst, kannst du sehen, was die jeweilige Sequenz des Programms bewirkt. Wenn du das ganze Programm auf dem EV3-Stein ausführst, kannst du sehen, wie beide Sequenzen gleichzeitig ablaufen.

13.2 WARTEN

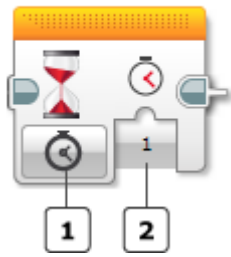


Der Warte-Block lässt dein Programm auf etwas warten, bevor es mit dem nächsten Block in der Sequenz weitermacht. Du kannst darauf warten, dass eine bestimmte Zeit (ein bestimmter Zeitbetrag) verstreicht, dass ein Sensor einen bestimmten Wert erreicht oder dass sich ein Sensorwert ändert.

Tipps und Tricks

Der Warte-Block lässt deinen Roboter nicht anhalten. Falls bei Beginn des Blocks irgendwelche Motoren angeschaltet sind, laufen sie auch während des Wartens weiter.

13.2.1 WÄHLE DEN WARTEMODUS AUS



1 Modus-Auswahl

2 Eingaben

Verwende die Modus-Auswahl, um den Wartemodus auszuwählen. Wähle den Zeitmodus, um einen bestimmten Zeitbetrag in Sekunden zu warten. Wähle einen Sensortyp und einen Vergleichsmodus aus, um darauf zu warten, dass der Sensor einen bestimmten Wert erreicht. Wähle einen Sensortyp und einen Änderungsmodus aus, um darauf zu warten, dass der Sensor einen neuen Messwert ausgibt oder sich der Messwert um einen bestimmten Betrag ändert.

Zeit



Im Zeitmodus wartet der Warte-Block genau den Zeitbetrag, den du in der Eingabe [Sekunden](#) festlegst. Die Zeit wird ab dem Start des Warte-Blocks gemessen.

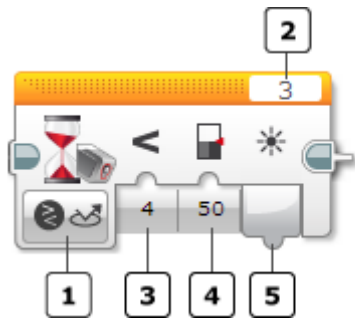
13.2.2 SENSOR-VERGLEICHSMODI

Jeder der im Warte-Block aufgelisteten Sensortypen verfügt über einen oder mehrere Vergleichsmodi. In einem Vergleichsmodus werden ständig Daten vom Sensor gelesen und es wird darauf gewartet, dass der Messwert einen von dir festgelegten Wert erreicht.

Manche Sensordatentypen lassen sich mit einem [Schwellenwert](#) vergleichen, andere dagegen mit bestimmten festgelegten Werten.

13.2.3 AUF EINEN SENSOR-SCHWELLENWERT WARTEN

In einem Sensor-Vergleichsmodus mit einer Eingabe Schwellenwert liest der Warte-Block ständig Numerische Daten vom Sensor und vergleicht diese Messwerte mit einem von dir festgelegten Schwellenwert. Der Warte-Block beendet das Warten, wenn der Vergleich mit dem Schwellenwert erfüllt und somit „Wahr“ ist. Du könntest bspw. warten, bis der Farbsensor einen Wert für die Stärke des reflektierten Lichts erkennt, der kleiner als 50 ist.



- 1 Modus-Auswahl
- 2 Anschlussauswahl
- 3 Eingabe „Vergleichsart“
- 4 Eingabe „Schwellenwert“
- 5 Ausgabe „Messwert“

Verwende die Modus-Auswahl, um einen Sensor-Vergleichsmodus mit einer Eingabe Schwellenwert auszuwählen. Falls in dem Modus ein Anschluss benutzt wird, verwendest du die Anschlussauswahl, um sicherzustellen, dass dieser Anschluss dem Anschluss am EV3-Stein entspricht, an den der Sensor oder Motor angeschlossen ist.

Wähle die zu verwendende Vergleichsart aus der Liste Vergleichsart und gib den Schwellenwert ein. Der Warte-Block wartet, bis der Vergleich mit dem Schwellenwert erfüllt und somit „Wahr“ ist.

13.2.4 AUF BESTIMMTE SENSORWERTE WARTEN

Die folgenden Vergleichsmodi des Warte-Blocks ermöglichen es dir, auf bestimmte Sensorwerte zu warten:

Stein-Tasten – Vergleichen – Stein-Tasten



Du kannst darauf warten, dass eine oder mehrere Stein-Tasten „Gedrückt“, „Ausgelassen“ oder „Angestoßen“ werden. Z.B. könntest du darauf warten, dass die Mittlere Taste „Gedrückt“ wird.

Wähle eine oder mehrere der Stein-Tasten aus der Menge an Tasten-Kennungen und wähle dann „Ausgelassen“, „Gedrückt“ oder „Angestoßen“ als „Aktion“. Der Warte-Block wartet, bis an einer der ausgewählten Stein-Tasten die ausgewählte Aktion stattgefunden hat. Am Ende kann der Ausgabe Messwert entnommen werden, welche Taste gedrückt wurde – sofern überhaupt eine Taste gedrückt wurde.

Beispiel



Dieses Programm lässt einen Roboter vorwärts fahren, bis die Mittlere Taste gedrückt wird. Daraufhin stoppt das Programm den Roboter und lässt einen Klang ertönen.

Farbsensor – Vergleichen – Farbe



Du kannst darauf warten, dass der Farbsensor eine oder mehrere bestimmte Farben erkennt. Z.B. könntest du darauf warten, dass der Farbsensor die Farbe Schwarz erkennt.

Wähle eine oder mehrere Farben aus der Eingabe Menge an Farben. Der Block wartet, bis eine der ausgewählten Farben erkannt wird. Die Ausgabe der erkannten Farbe erfolgt unter Messwert.

Berührungssensor – Vergleichen – Zustand



Du kannst darauf warten, dass ein Berührungssensor „Gedrückt“, „Ausgelassen“ oder „Angestoßen“ wird.

Wähle in der Eingabe Zustand eine der Optionen „Ausgelassen“, „Gedrückt“ oder „Angestoßen“. Der Block wartet darauf, dass der Berührungssensor den ausgewählten Zustand einnimmt.

13.2.5 SENSOR-ÄNDERUNGSMODI

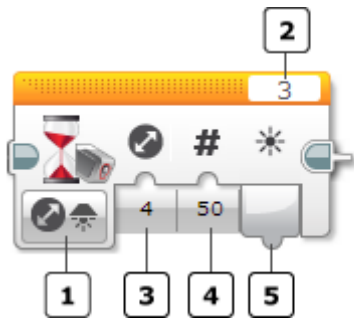
Ändern

Jeder der im Werte-Block aufgelisteten Sensortypen verfügt über einen oder mehrere Änderungsmodi. In einem Änderungsmodus werden ständig Daten vom Sensor gelesen und es wird darauf gewartet, dass sich diese Daten in einen anderen Wert bzw. um einen von dir festgelegten Betrag ändern.

Bei einigen Typen von Sensordaten kannst du darauf warten, dass sich der Wert um einen bestimmten Betrag ändert, bei anderen Datentypen dagegen auf eine Änderung des erkannten Wertes.

13.2.6 WARTEN, DASS SICH EIN SENSORWERT UM EINEN BESTIMMTEN BETRAG ÄNDERT

In einem Sensor-Änderungsmodus mit einer Eingabe Betrag liest der Werte-Block bei Beginn des Blocks zunächst einen Ausgangswert und anschließend ständig weitere Messwerte von diesem Sensor, um darauf zu warten, dass sich der Sensorwert – verglichen mit dem Ausgangswert – um den festgelegten Betrag ändert. Du kannst die erforderliche Richtung der Änderung („Erhöhen“, „Verringern“ oder „Beliebig“) auswählen. Du könntest bspw. darauf warten, dass sich der vom Farbsensor gemessene Wert „Stärke des Umgebungslichts“ um 10 erhöht.



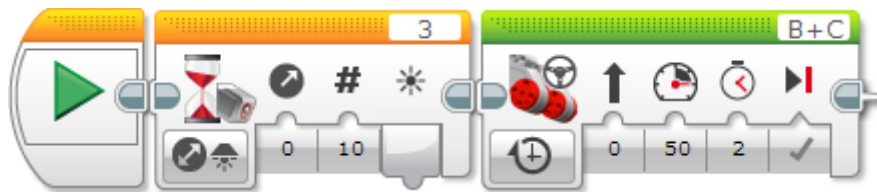
- 1** Modus-Auswahl
- 2** Anschlussauswahl
- 3** Eingabe „Richtung“
- 4** Eingabe „Betrag“
- 5** Ausgabe „Messwert“

Verwende die Modus-Auswahl, um einen Sensor-Änderungsmodus mit einer Eingabe Betrag auszuwählen. Falls in dem Modus ein Anschluss benutzt wird, verwendest du die Anschlussauswahl, um sicherzustellen, dass dieser Anschluss dem Anschluss am EV3-Stein entspricht, an den der Sensor oder Motor angeschlossen ist.

Wähle in der Eingabe Richtung die gewünschte Richtung der Änderung („Erhöhen“, „Verringern“ oder „Beliebig“) und gib den Betrag der Änderung ein. Indem du in der Eingabe Richtung die Option „Beliebig“ wählst, lässt du den Block darauf warten, dass sich der Sensorwert um den festgelegten Betrag erhöht oder verringert.

Der endgültige Sensorwert wird unter Messwert ausgegeben.

Beispiel



Dieses Programm wartet darauf, dass sich der vom Farbsensor gemessene Wert „Stärke des Umgebungslichts“ um 10 erhöht, bevor es den Roboter losfahren lässt. Das könntest du dazu verwenden, deinen Roboter losfahren zu lassen, wenn du die Zimmerbeleuchtung anschaltest oder mit einer Taschenlampe auf den Sensor leuchtest.

13.2.7 DARAUF WARTEN, DASS SICH EIN SENSORWERT IN EINEN ANDEREN WERT ÄNDERT

Die folgenden Änderungsmodi des Warte-Blocks ermöglichen es dir, darauf zu warten, dass ein Sensor irgendeinen anderen Wert als bei Beginn des Blocks erkennt.

Stein-Tasten – Ändern – Stein-Tasten



Im Modus „Stein-Tasten – Ändern – Stein-Tasten“ wird darauf gewartet, dass irgendeine der Stein-Tasten gedrückt oder ausgelassen wird. Die Tasten-Kennung der Taste, die gedrückt oder ausgelassen wurde, wird unter Geänderte Taste ausgegeben.

Farbsensor – Farbe



Im Modus „Farbsensor – Ändern – Farbe“ wird darauf gewartet, dass sich der vom Farbsensor im Farbmodus erkannte Farbstatus (0 bis 7) ändert. Die Ausgabe der zuletzt erkannten Farbe erfolgt unter [Farbe](#).

Berührungssensor – Ändern – Zustand



Im Modus „Berührungssensor – Ändern – Zustand“ wird darauf gewartet, dass sich der Zustand des Berührungssensors ändert. Wenn sich der Berührungssensor zu Beginn des Blocks im Zustand „Ausgelassen“ befindet (er also nicht gedrückt ist), wartet der Block darauf, dass der Berührungssensor gedrückt wird. Wenn sich der Berührungssensor zu Beginn des Blocks im Zustand „Gedrückt“ befindet, wartet der Block darauf, dass der Berührungssensor „Ausgelassen“ wird. Die Ausgabe des Endzustands erfolgt unter Messwert.

13.2.8 EINGABEN UND AUSGABEN

Die für den Warte-Block verfügbaren Eingaben sind vom jeweils ausgewählten Modus abhängig. Du kannst die Eingabewerte direkt in den Block eingeben. Alternativ hierzu können als Eingabewerte auch die Ausgaben anderer Programmierblöcke dienen, die über Datenleitungen übertragen werden.

Eingabe	Typ	Hinweise
Sekunden	Numerische Werte	Die Anzahl der Sekunden, die im Zeitmodus gewartet werden soll.
Vergleichsart	Numerische Werte	Vergleichsart für einen Sensor-Vergleichsmodus mit einem Schwellenwert 0: = (Gleich)

		1: \neq (Ungleich) 2: $>$ (Größer als) 3: \geq (Größer gleich) 4: $<$ (Kleiner als) 5: \leq (Kleiner gleich)
Schwellenwert	Numerische Werte	Mit den Sensordaten zu vergleichender Wert für einen Sensor-Vergleichsmodus mit einem Schwellenwert
Richtung	Numerische Werte	Richtung, in die sich ein Numerischer Sensorwert ändern soll. Wird in den Sensor-Vergleichsmodi verwendet, die über eine Eingabe „Betrag“ verfügen. 0 = Erhöhen 1 = Verringern 2 = Beliebig
Betrag	Numerische Werte	Betrag, um den sich ein Sensorwert in einem Sensor-Vergleichsmodus ändern soll.
(Eingaben für einzelne Sensortypen)		Weitere Informationen über die Sensordaten entnimmst du den Hilfe-Beiträgen zu den einzelnen Sensortypen.

Die Ausgabe des Warte-Blocks enthält den am Ende des Warte-Vorgangs in einem Sensor-Vergleichsmodus oder Sensor-Änderungsmodus gemessenen Sensorwert. Verwende eine Datenleitung, um die Ausgabe mit einem anderen Programmierblock zu verbinden.

Ausgabe	Typ	Hinweise
(Die Ausgaben variieren je nach Sensortyp.)		Weitere Informationen über die Sensordaten entnimmst du den Hilfe-Beiträgen zu den einzelnen Sensortypen.
Messwert	Numerische Werte	Der Sensorwert nach dem Warten

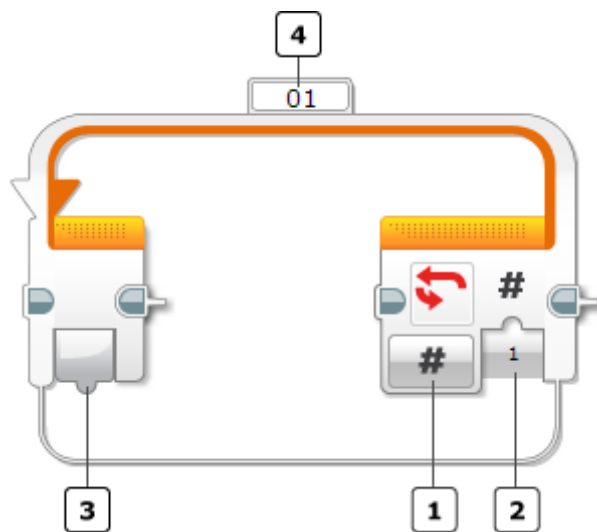
13.3 SCHLEIFE



Beim Schleifen-Block handelt es sich um einen Container, der eine Sequenz von Programmierblöcken enthalten kann. Der Schleifen-Block sorgt dafür, dass die enthaltene Blocksequenz wiederholt wird. Du kannst entscheiden, wie oft die Blocksequenz wiederholt werden soll. Du kannst sie aber auch unendlich wiederholen lassen – oder bis ein Sensortest oder eine andere Bedingung erfüllt ist (den Wert „Wahr“ ausgegeben hat).

Nur die Blöcke in der Schleife werden wiederholt. Nach Abschluss der Schleife wird das Programm mit den Blöcken nach der Schleife fortgesetzt.

13.3.1 WÄHLE DEN SCHLEIFENMODUS



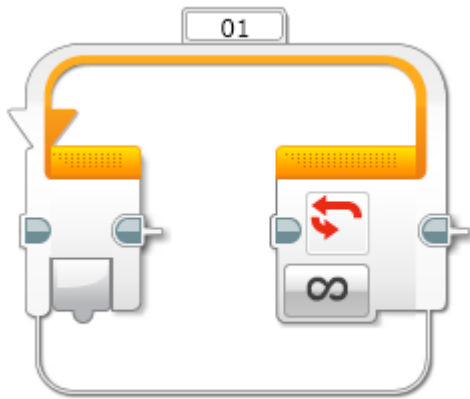
- 1** Modus-Auswahl
- 2** Eingaben
- 3** Ausgabe „Zählen“
- 4** Schleifen-Name

Verwende die Modus-Auswahl, um festzulegen, wie oft die Schleife wiederholt wird. Mit den unterschiedlichen Modi wird bestimmt, welche Bedingung die Schleife enden lässt.

In das Feld „Schleifen-Name“ oben im Schleifen-Block kannst du einen Namen für die Schleife eingeben. Dieser Name wird vom Schleifen-Interrupt-Block verwendet, um dir noch eine weitere Möglichkeit zum Beenden der Schleife zu eröffnen.

13.3.2 MODI

Unbegrenzt



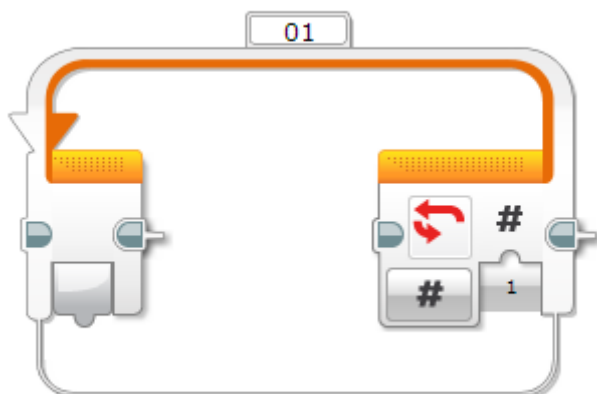
Im Endlosmodus werden die Blöcke innerhalb der Schleife endlos wiederholt. Jegliche hinter der Schleife platzierten Blöcke werden niemals erreicht.

Beispiel 1



Dieses Programm lässt einen Roboter abwechselnd geradeaus und Kurven fahren – und zwar immer und immer wieder –, bis das Programm stoppt.

Zählen



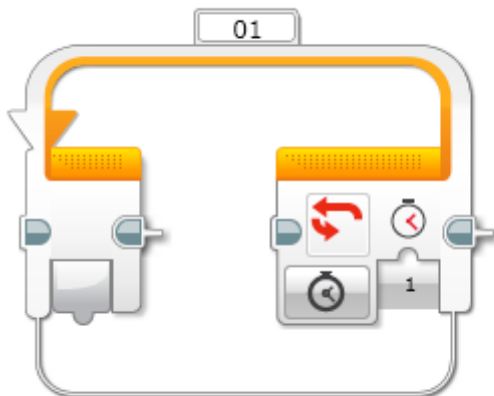
Im Zählmodus gibt die Eingabe Zählen an, wie oft die in der Schleife enthaltenen Blöcke wiederholt werden sollen.

Beispiel



Dieses Programm lässt den Roboter einmal „Hello“, dann sechsmal „Go“ und anschließend einmal „Bravo“ sagen. Der Schleifen-Block enthält einen Klang-Block und ist so konfiguriert, dass er sechsmal wiederholt wird.

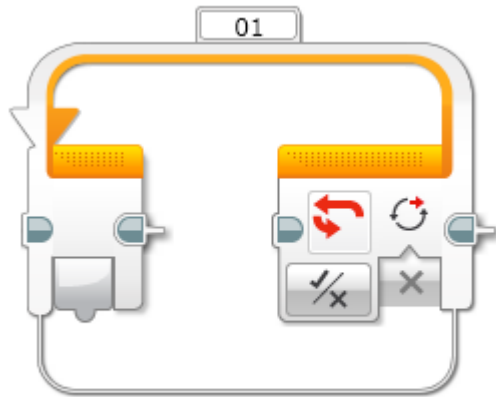
Zeit



Im Zeitmodus kannst du die Dauer, in der die Schleife wiederholt werden soll, in der Eingabe [Sekunden](#) festlegen. Die Zeit wird ab dem Anfang der Schleife gemessen.

Die zeitliche Beschränkung wird erst am Ende der Schleifensequenz getestet. Die Schleifensequenz wird immer mindestens einmal ausgeführt und die Schleife beginnt nur dann wieder am Anfang, wenn die in diesem Moment verstrichene Zeit unter dem Betrag liegt, der unter [Sekunden](#) festgelegt ist.

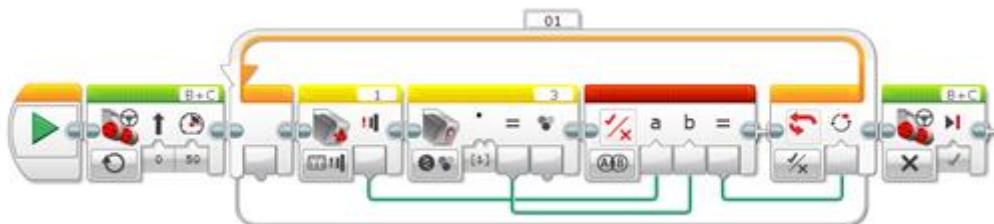
Logische Werte



Im Modus „Logischer Wert“ wird die Schleife wiederholt, bis am Ende der Schleifensequenz für die Eingabe der Wert „Wahr“ gilt. Die Schleifensequenz wird immer mindestens einmal ausgeführt und die Eingabe wird am Ende jeder Schleifenwiederholung überprüft.

Verwende eine Datenleitung, um die Eingabe mit einer Ausgabe „Logischer Wert“ in einem Programmierblock innerhalb der Schleife zu verbinden. Diese Ausgabe sollte den Wert „Wahr“ enthalten, um die Schleife zu beenden.

Beispiel



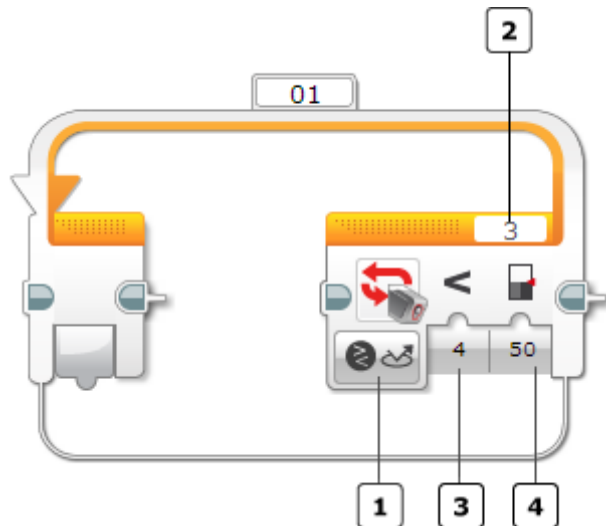
Dieses Programm lässt einen Roboter vorwärts fahren, bis der Berührungssensor gedrückt wird oder der Farbsensor die Farbe Schwarz erkennt – je nachdem, was zuerst der Fall ist. Die Ausgabe eines Blocks Logische Verknüpfungen wird für die Eingabe der Schleife verwendet. Am Ende der Schleife stoppt der Roboter.

13.3.3 SENSOR MODI

Der Schleifen-Block enthält mehrere Modi, die Sensordaten lesen und mit einem Eingabewert vergleichen. Für jeden Sensortyp gibt es unterschiedliche Modi. Manche Sensordatentypen lassen sich mit einem [Schwellenwert](#) vergleichen, andere dagegen mit bestimmten festgelegten Werten.

13.3.4 SENSORDATEN MIT EINEM SCHWELLENWERT VERGLEICHEN

In einem Modus mit einer Eingabe [Schwellenwert](#) wiederholt der Schleifen-Block die in der Schleife enthaltene Blocksequenz, bis für den Schwellenvergleich der Wert „Wahr“ gilt. Du könntest bspw. eine Schleife wiederholen lassen, bis der Farbsensor einen Wert für die Stärke des reflektierten Lichts erkennt, der kleiner als der von dir festgelegte Schwellenwert ist.



- 1 Modus-Auswahl
- 2 Anschlussauswahl
- 3 Eingabe „Vergleichsart“
- 4 Eingabe „Schwellenwert“

Verwende die Modus-Auswahl, um einen Sensor-Modus mit einer Eingabe Schwellenwert auszuwählen. Falls in dem Modus ein Anschluss benutzt wird, verwendest du die Anschlussauswahl, um sicherzustellen, dass dieser Anschluss dem Anschluss am EV3-Stein entspricht, an den der Sensor oder Motor angeschlossen ist.

Wähle die zu verwendende Vergleichsart aus der Liste Vergleichsart und gib den Schwellenwert ein.

Jedes Mal, wenn die in einer Schleife enthaltene Blocksequenz endet, liest der Schleifen-Block die Sensordaten und vergleicht sie mit der Schwelle. Liefert dieser Vergleich den Wert „Falsch“, dann wird die Schleife wiederholt. Liefert dieser Vergleich den Wert „Wahr“, dann macht das Programm mit dem nächsten Block nach der Schleife weiter.

Beispiel



Dieses Programm lässt einen Roboter vorwärts fahren und gleichzeitig drei verschiedene Klänge wiederholen, bis der Farbsensor eine Stärke des reflektierten Lichts von weniger als 50 erkennt, woraufhin der Roboter stoppt.

Tipps und Tricks

Erst am Ende der Schleifensequenz wird überprüft, ob die Sensorschwelle überschritten wurde. Wenn der Farbsensor im oben beschriebenen Beispiel also bspw. in der Mitte der Schleife kurzzeitig eine

dunkle Farbe überquert, „sieht“ der Roboter das unter Umständen nicht und dann hält er auch nicht an.

13.3.5 SENSORDATEN MIT BESTIMMTEN WERTEN VERGLEICHEN

Die folgenden Sensor-Modi gestatten es einer Schleife, die Sensordaten mit bestimmten Eingabewerten zu vergleichen. Die Schleife wird wiederholt, bis die Sensordaten einem bestimmten Eingabewert entsprechen.

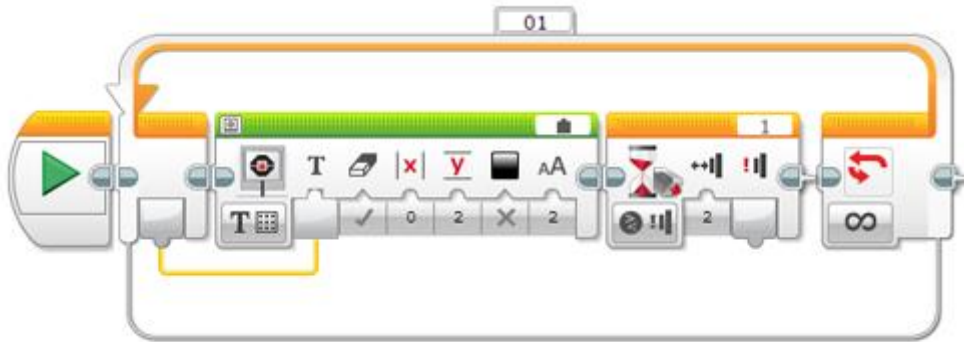
Modus	Hinweise
Stein-Tasten	Die Schleife wird wiederholt, bis eine der ausgewählten Tasten die ausgewählte Aktion auslöst.
Farbsensor – Farbe	Die Schleife wird wiederholt, bis eine der ausgewählten Farben erkannt wird.
Infrarotsensor – Fernsteuerung	Die Schleife wird wiederholt, bis eine der ausgewählten Tasten auf der IR-Fernsteuerung gedrückt wird.
Berührungssensor	Die Schleife wird wiederholt, bis sich der Berührungssensor im ausgewählten Zustand befindet.
Ultraschallsensor – Vorhanden	Die Schleife wird wiederholt, bis ein Ultraschallsignal erkannt wird.
Nachrichtenart	Die Schleife wird wiederholt, bis die festgelegte Nachrichtenart empfangen wird.

13.3.6 VERWENDUNG DER AUSGABE „ZÄHLEN“

Die Ausgabe [Zählen](#) am Anfang einer Schleife informiert dich darüber, wie oft die Schleifensequenz bereits ausgeführt wurde. Die kannst die Ausgabe „Zählen“ innerhalb der Schleife dazu verwenden, um die Schleife bei jeder Wiederholung etwas anderes machen zu lassen.

Während des ersten Durchlaufs der Schleife besitzt die Ausgabe „Zählen“ den Wert „0“. Mit jeder Wiederholung der Schleife erhöht sich dieser Wert um 1.

Beispiel



Dieses Programm zeigt eine Zählung an, wie oft der Berührungssensor angestoßen (gedrückt und wieder ausgelassen) wurde. Ein Warte(n)-Block innerhalb der Schleife lässt die Schleife warten, bis der Berührungssensor erneut angestoßen wird, bevor sie wiederholt wird und die Ausgabe Zählen wird als Texteingabe im Anzeige-Block verwendet.

13.3.7 EINGABEN UND AUSGABEN

Die für den Schleifen-Block verfügbaren Eingaben sind vom jeweils ausgewählten Modus abhängig. Du kannst die Eingabewerte direkt in den Block eingeben. Alternativ hierzu können als Eingabewerte auch die Ausgaben anderer Programmierblöcke dienen, die über Datenleitungen übertragen werden.

Eingabe	Typ	Hinweise
Zählen	Numerische Werte	Wie oft die Schleife im Zählmodus zu wiederholen ist.
Sekunden	Numerische Werte	Die Anzahl der Sekunden, in denen die Schleife im Zeitmodus wiederholt werden soll.
Bis	Logische Werte	Im Modus „Logischer Wert“ endet die Schleife, wenn diese Eingabe den Wert „Wahr“ enthält.
Vergleichsart	Numerische Werte	Vergleichsart für einen Sensor-Modus mit einem Schwellenwert 0: = (Gleich) 1: ≠ (Ungleich) 2: > (Größer als) 3: ≥ (Größer gleich) 4: < (Kleiner als) 5: ≤ (Kleiner gleich)
Schwellenwert	Numerische Werte	Mit den Sensordaten zu vergleichender Wert für einen Sensor-Modus mit einem Schwellenwert

(Eingaben für einzelne Sensortypen)		Weitere Informationen über die Sensordaten entnimmst du den Hilfe-Beiträgen zu den einzelnen Sensortypen.
-------------------------------------	--	---

Die Ausgabe des Schleifen-Blocks ist die Anzahl der Wiederholungen dieser Schleife. Verwende eine Datenleitung, um die Ausgabe mit einem anderen Programmierblock zu verbinden.

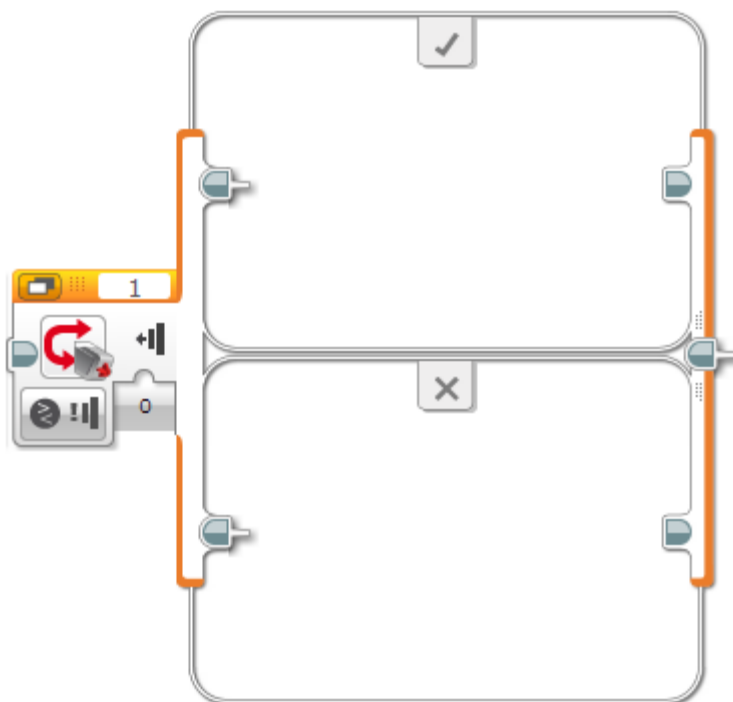
Ausgabe	Typ	Hinweise
Zählen	Numerische Werte	Die Anzahl der Wiederholungen dieser Schleife. Während des ersten Durchlaufs der Schleife besitzt die Ausgabe „Zählen“ den Wert „0“.

13.4 SCHALTER



Beim Schalter-Block handelt es sich um einen Behälter, der zwei oder mehr Sequenzen von Programmierblöcken enthalten kann. Jede Sequenz wird als ein Fall bezeichnet. Ein Test zu Beginn des Schalter-Blocks bestimmt, welcher Fall ausgeführt wird. Bei jeder Ausführung des Schalter-Blocks wird jeweils nur ein Fall ausgeführt.

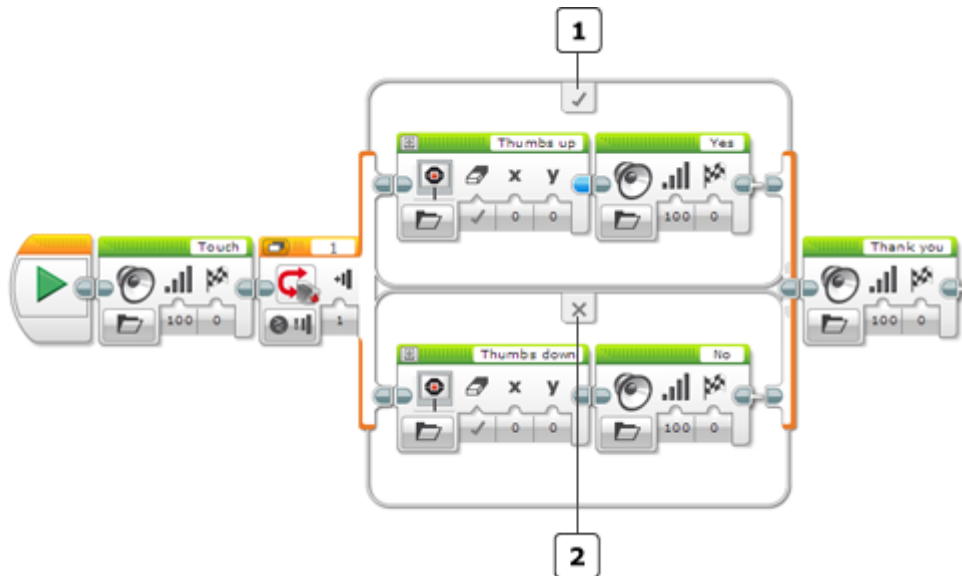
Der hier dargestellte Schalter-Test kann auf Grundlage eines Sensordatenwertes bzw. eines Wertes aus einer [Datenleitung](#) entscheiden, welcher Fall auszuführen ist. Nach der Auswahl und Ausführung eines Falls wird das Programm mit den Blöcken nach dem Schalter fortgesetzt.



BEISPIEL

Das Programm unten lässt einen Roboter das Wort „Touch“ (Berührung) sagen und testet dann, ob der Berührungssensor gedrückt wird. Wenn dies so ist, wird der (obere) Wahre Fall ausgeführt und der Roboter zeigt „Thumbs up“ (Daumen nach oben) und sagt „Yes“ (Ja). Wird der Berührungssensor nicht gedrückt, dann wird der (untere) Falsche Fall ausgeführt und der Roboter zeigt „Thumbs down“ (Daumen nach unten) und sagt „No“ (Nein). Nach dem Schalter sagt der Roboter „Thank you“ (Danke).

Beispiel



1 Wahrer Fall

2 Falscher Fall

Der Berührungssensor wird getestet, sobald der Roboter das Wort „Touch“ (Berührung) vollständig ausgesprochen hat. Wenn der Sensor in diesem Moment nach unten gehalten wird, führt der Schalter den „Wahren Fall“ aus, anderenfalls führt er den „Falschen Fall“ aus.

Tipps und Tricks

Ein Schalter wartet nicht darauf, dass ein Sensordatenwert oder eine Datenleitung einen bestimmten Wert erreicht. Der Test wird ausgeführt, sobald der Schalter-Block startet, und einer der Fälle wird sofort nach dem Test ausgewählt und ausgeführt.

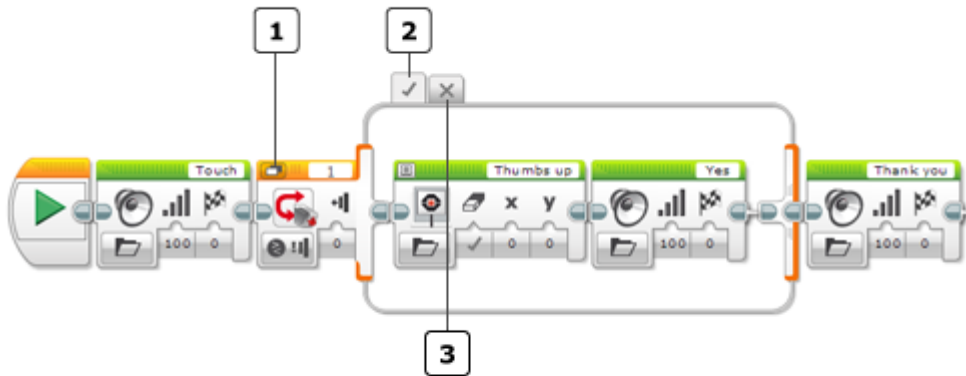
In dem Programm oben kannst du den Berührungssensor vor dem Test drücken (oder sogar bevor das Programm startet) und während des Tests gedrückt halten, um sicherzustellen, dass der Schalter den Wahren Fall ausführt. Probiere einfach mal aus, wie lange du mit dem Drücken des Sensors warten und den Roboter immer noch „Yes“ (Ja) sagen lassen kannst.

Du kannst jeden Fall in einem Schalter leer lassen, wenn der Roboter in dieser Situation nichts machen soll. Probiere aus, zwei Blöcke aus dem Falschen Fall im Programm oben zu entfernen, und beobachte, was dann passiert.

13.4.1 ANSICHT IM REGISTERFORMAT

Ein Schalter kann in der „Flachen Ansicht“ (siehe Beispiel oben) oder in der „Ansicht im Registerformat“ (siehe Beispiel unten) dargestellt werden. In der Flachen Ansicht werden alle Fälle in verschiedenen Zeilen dargestellt. In der Ansicht im Registerformat ist jeweils immer nur einer der Fälle sichtbar.

Du kannst die Ansicht im Registerformat verwenden, damit dein Programm weniger Platz auf dem Bildschirm belegt. Die Ansicht hat keine Auswirkung auf die Art und Weise, wie der Schalter ausgeführt wird.

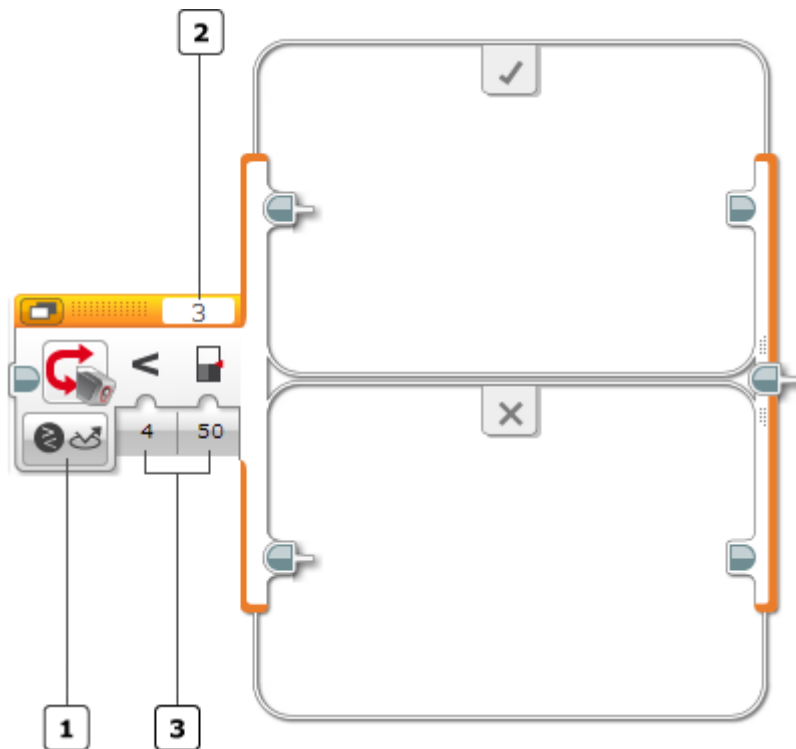


- 1 Ansichtsauswahl (Flach-/Registerformat)
- 2 Wahrer Fall
- 3 Falscher Fall

Um bei der Darstellung eines Schalters zwischen der Flachen Ansicht und der Ansicht im Registerformat zu wechseln, klickst du auf die Ansichtsauswahl (Flach-/Registerformat)

Um bei einem im Registerformat dargestellten Schalter einen anderen Fall anzuzeigen, klickst du auf die verschiedenen Register am oberen Rand des Schalters.

Wähle den Schalter-Test



- 1 Modus-Auswahl
- 2 Anschlussauswahl
- 3 Eingaben

Verwende die Modus-Auswahl, um die Testart auszuwählen, die der Schalter verwenden wird, um zu entscheiden, welcher Fall ausgeführt werden soll. Du kannst einen Sensorwert oder den Wert aus einer Datenleitung benutzen.

Falls du einen Modus auswählst, in dem ein Anschluss benutzt wird, verwendest du die Anschlussauswahl, um sicherzustellen, dass dieser Anschluss dem Anschluss am EV3-Stein entspricht, an den der Sensor oder Motor angeschlossen ist.

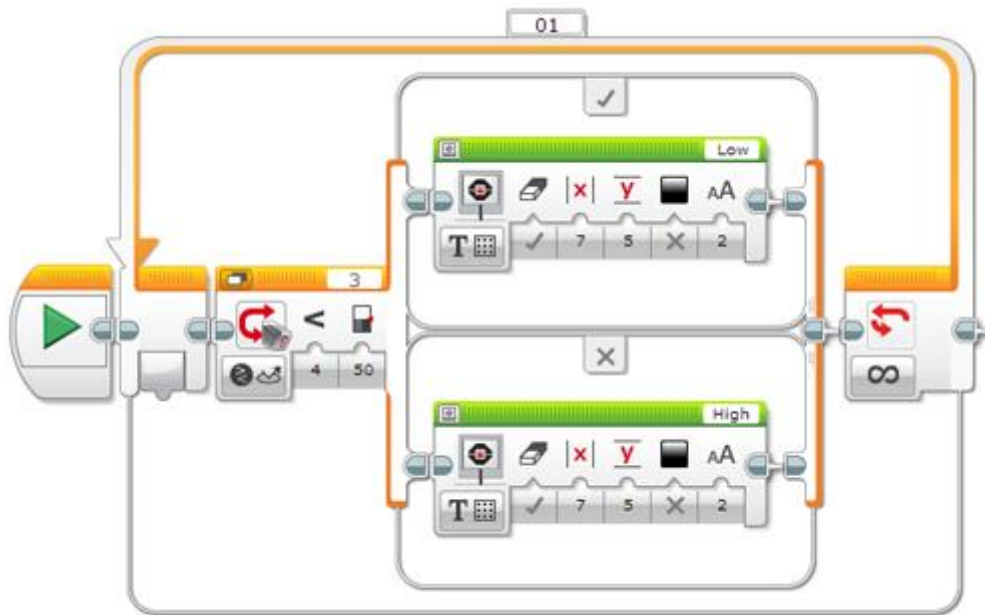
13.4.2 EINE SENSORSCHWELLE TESTEN

Der Schalter-Block enthält mehrere Modi, die einen numerischen Sensordatenwert lesen und mit einem Schwellenwert vergleichen, um als Ergebnis den Wert „Wahr“ oder „Falsch“ zu erhalten. Im Modus „Farbsensor – Vergleichen – Stärke des reflektierten Lichts“ könntest du bspw. testen, ob die vom Farbsensor gemessene Stärke des reflektierten Lichts unter dem Wert 50 liegt.

In diesen Modi enthält der Schalter-Block zwei Fälle. Wenn als Ergebnis des Tests der Wert „Wahr“ ausgegeben wird, dann wird der Wahre Fall ausgeführt – und ansonsten der Falsche Fall.

Um einen Modus mit einer Sensorschwelle zu verwenden, wählst du die Vergleichsart (z.B. Kleiner als) und gibst den mit den Sensordaten zu vergleichenden Schwellenwert (z.B. 50) ein. Der Schalter-Block liest einen einzigen Sensordatenwert und vergleicht diesen mit einem Schwellenwert, um als Ergebnis den Wert „Wahr“ oder „Falsch“ zu erhalten.

Beispiel



In diesem Programm wird der Schalter-Block im Modus „Farbsensor – Vergleichen – Stärke des reflektierten Lichts“ verwendet, um zu testen, ob die Stärke des reflektierten Lichts weniger als 50 % beträgt. Wenn dies so ist, wird „Low“ (Niedrig) angezeigt, anderenfalls „High“ (Hoch). Der Schalter wird in einer Schleife wiederholt, sodass die Anzeige ständig aktualisiert wird – und zwar auf Grundlage neuer Sensortests.

Tipps und Tricks

Sensortests gehen sehr schnell vonstatten. Ohne die [Schleife](#) im Beispiel oben würde das Programm den Sensor nur einmal testen und den Test so schnell abschließen, dass du ihn nicht einmal bemerkst würdest.

13.4.3 AUF BESTIMMTEN SENSORWERTE TESTEN

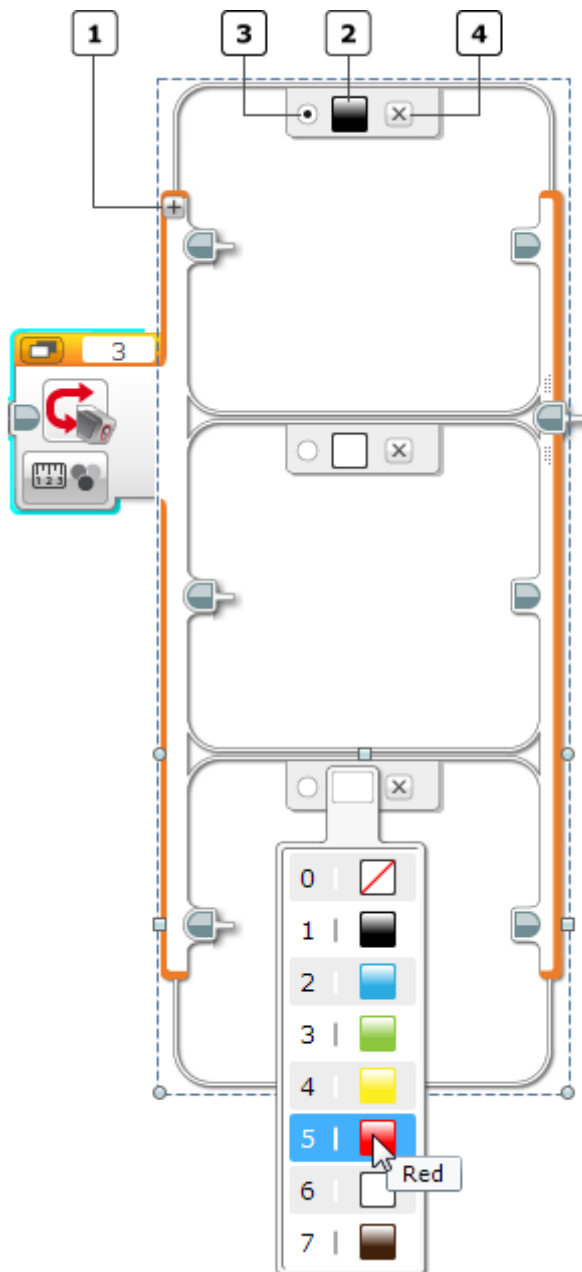
Die folgenden Modi gestatten es einem Schalter, auf bestimmte Sensorwerte zu testen. Du kannst zwei oder mehr Fälle in dem Schalter erstellen, die den unterschiedlichen Werten entsprechen, auf die getestet werden soll, und der Schalter wählt dann den passenden Fall.

Modus	Verwendungszweck
Stein-Tasten – Messen	Je nach gedrückter Stein-Taste zwischen zwei oder mehr Fällen auswählen.
Stein-Tasten – Vergleichen	Je nachdem, ob eine der ausgewählten Stein-Tasten „Gedrückt“, „Ausgelassen“ oder „Angestoßen“ wird, zwischen zwei Fällen entscheiden.
Farbsensor – Messen – Farbe	Je nach erkannter Farbe zwischen zwei oder mehr Fällen auswählen.
Farbsensor – Vergleichen – Farbe	Je nachdem, ob eine der ausgewählten Farben erkannt wurde, zwischen zwei Fällen auswählen.

Berührungssensor	Je nachdem, ob der Berührungssensor „Gedrückt“ wird (oder nicht), „Ausgelassen“ wird (oder nicht) bzw. „Angestoßen“ wird (oder nicht), einen von zwei Fällen auswählen.
------------------	---

13.4.4 AUF MEHRERE WERTE TESTEN

Die Sensoren-Messmodi des Schalter-Blocks gestatten dir die Angabe von zwei oder mehr verschiedenen Sensorwerten, auf die getestet werden soll. Du kannst jedem Wert einen anderen Fall im Schalter zuordnen. Im Modus „Farbsensor – Messen – Farbe“ könntest du beispielsweise auf die Farben Schwarz, Weiß und Rot testen lassen und für jede Farbe einen anderen Fall erstellen.



1 Fall hinzufügen

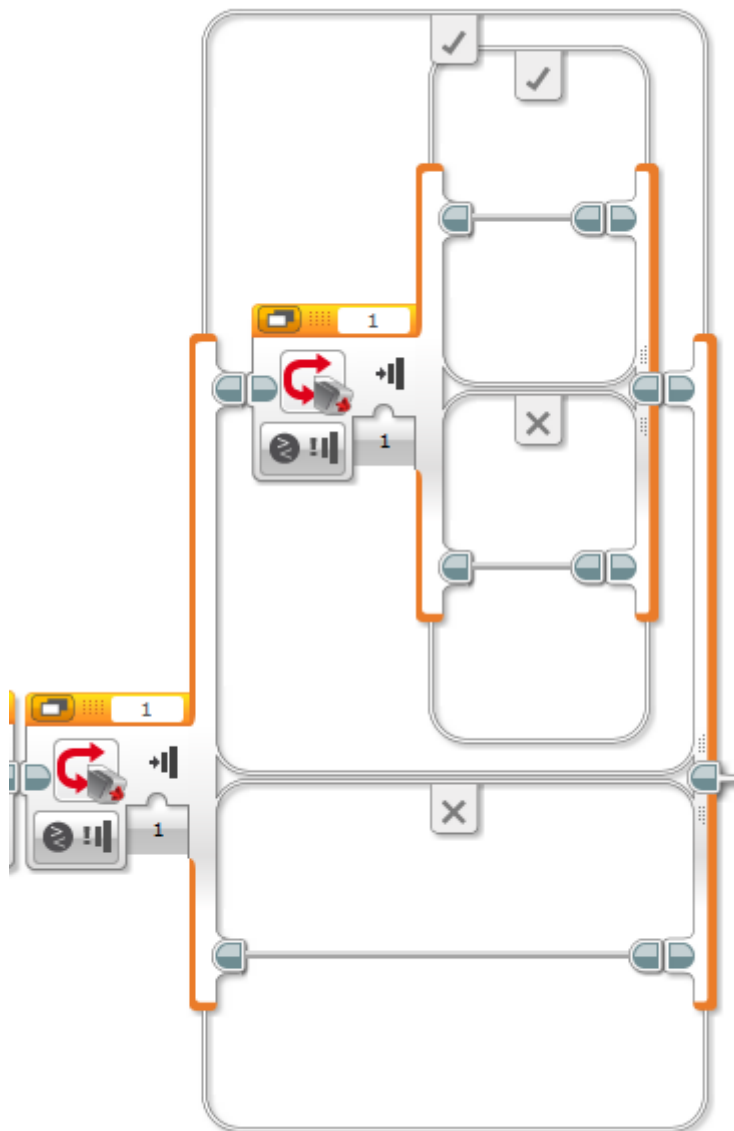
- 2 Fallwert
- 3 Standardfall
- 4 Fall entfernen

Um in einem Sensoren-Messmodus mehrere Fälle zu verwenden, klickst du auf die Schaltfläche „Fall hinzufügen“, um die gewünschte Anzahl an Fällen zu erstellen. Klicke für jeden Fall auf die Schaltfläche „Fallwert“, um aus der Liste einen Wert für den Sensor auszuwählen. Du kannst auf die Schaltfläche „Fall entfernen“ klicken, um einen Fall zu entfernen.

Klicke auf die Schaltfläche „Standardfall“, um einen Fall als Standardfall zu markieren. Der Standardfall wird ausgeführt, wenn der Sensor einen Wert erkennt, der keinem der Fälle im Schalter entspricht.

Tipps und Tricks

Eine andere Variante mehrere Werte zu testen, ist mit verschachtelten Schaltern zu arbeiten. Dazu wird einfach ein weiterer Schalterblock in einen bereits vorhandenen Schalterblock eingeführt.



13.4.5 EINEN WERT AUS EINER DATENLEITUNG TESTEN

In den Modi „Logischer Wert“, „Text“ und „Numerischer Wert“ kann der Schalter den auszuführenden Fall auf der Grundlage eines Eingabewerts aus einer Datenleitung auswählen.

13.4.6 MODI

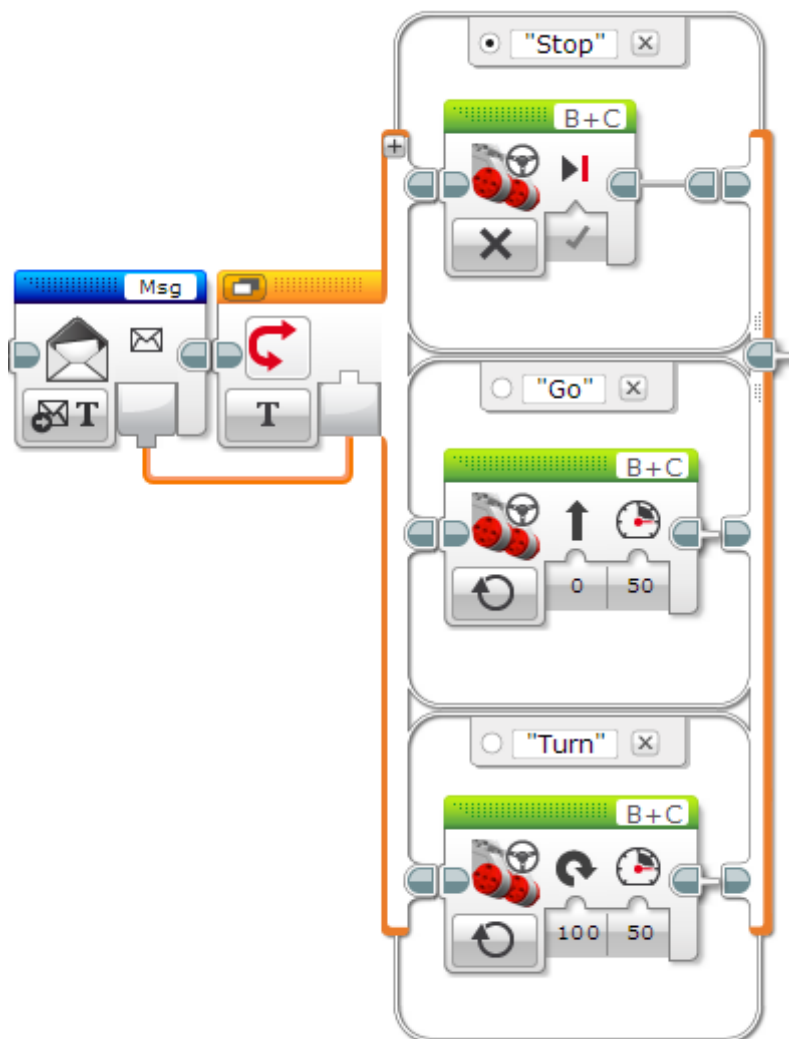
Modus „Logischer Wert“

Im Modus „Logischer Wert“ entscheidet ein Schalter auf Grundlage des Wertes in der Eingabe Logischer Wert zwischen einem Wahren Fall und einem Falschen Fall. Mithilfe einer Datenleitung kannst du die Ausgabe „Logischer Wert“ eines beliebigen Programmierblocks mit der Eingabe „Logischer Wert“ eines anderen Programmierblocks verbinden.

Modus „Text“

Im Modus „Text“ vergleicht der Schalter den Wert in der Eingabe Text mit zwei oder mehr von dir angegebenen Textwerten und führt dann für jeden Wert einen anderen Fall aus. Es wird der Fall ausgeführt, dessen Wert genau der Texteingabe entspricht. Wenn keine Werte übereinstimmen, wird der Standardfall ausgeführt.

Beispiel

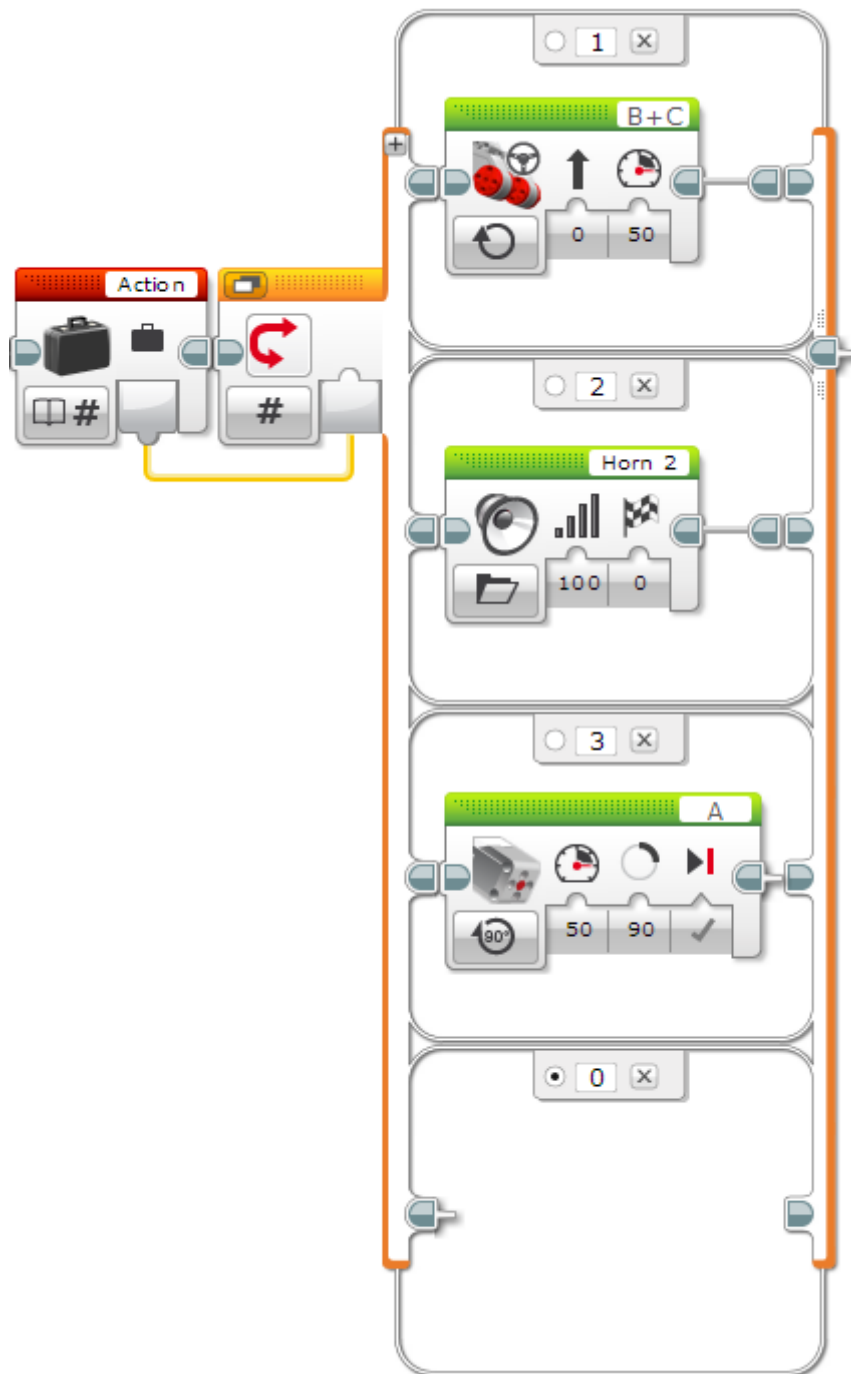


Bei diesem Beispiel verwendet ein Schalter im Modus „Text“ die Ausgabe eines Nachrichten-Blocks, um je nach empfangener Nachrichtenart, einen von drei verschiedenen Fällen auszuwählen.

Modus „Numerischer Wert“

Im Modus „Numerischer Wert“ vergleicht der Schalter den Wert in der Eingabe Numerischer Wert mit zwei oder mehr von dir angegebenen Numerischen Werten und führt dann für jeden Wert einen anderen Fall aus. Es wird der Fall ausgeführt, dessen Wert genau der Eingabe Numerischer Wert entspricht. Wenn keine Werte übereinstimmen, wird der Standardfall ausgeführt.

Beispiel



Bei diesem Beispiel verwendet ein Schalter im Modus „Numerischer Wert“ die Ausgabe eines Variablen-Blocks, um einen von drei verschiedenen Fällen auszuwählen. Der Standardfall des Schalters ist

leer. Wenn also die Variable „Aktion“ irgendeinen anderen Wert als 1, 2 oder 3 besitzt, macht der Schalter nichts.

13.4.7 EINGABEN

Die für den Schalter-Block verfügbaren Eingaben sind vom jeweils ausgewählten Modus abhängig. Du kannst die Eingabewerte direkt in den Block eingeben. Alternativ hierzu können als Eingabewerte auch die Ausgaben anderer Programmierblöcke dienen, die über Datenleitungen übertragen werden.

Eingabe	Typ	Hinweise
Logische Werte	Logische Werte	Werden verwendet, um einen Fall im Modus „Logischer Wert“ auszuwählen.
Zahl	Numerische Werte	Werden verwendet, um einen Fall im Modus „Numerischer Wert“ auszuwählen.
Text	Text	Wird verwendet, um einen Fall im Modus „Text“ auszuwählen.
Vergleichsart	Numerische Werte	<p>Vergleichsart für einen Modus mit einer Eingabe „Schwellenwert“</p> <p>0: = (Gleich)</p> <p>1: ≠ (Ungleich)</p> <p>2: > (Größer als)</p> <p>3: = (Größer gleich)</p> <p>4: < (Kleiner als)</p> <p>5: = (Kleiner gleich)</p>
Schwellenwert	Numerische Werte	Der mit den Sensordaten zu vergleichende Wert, um einen Wahren Fall oder Falschen Fall auf der Grundlage eines Sensorwertes „Numerischer Wert“ auszuwählen.

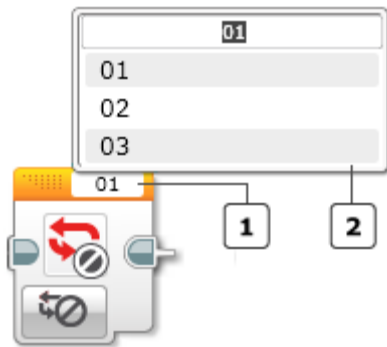
13.5 SCHLEIFEN-INTERRUPT



Der Schleifen-Interrupt-Block lässt einen [Schleifen](#)-Block enden. Danach werden keine weiteren Blöcke aus der Schleifensequenz mehr ausgeführt und das Programm wird mit den Blöcken fortgesetzt, die nach der Schleife kommen (sofern vorhanden). Du kannst den Schleifen-Namen verwenden, um den zu unterbrechenden Schleifen-Block anzugeben.

Darüber hinaus kannst du den Schleifen-Interrupt-Block verwenden, um die Schleife vorzeitig zu beenden oder um auf eine veränderte Bedingung zu reagieren. Du kannst eine Schleife im Schleifen-Block selbst unterbrechen oder von jeder Blocksequenz aus, die gleichzeitig ausgeführt wird.

13.5.1 WÄHLE DEN SCHLEIFENNAMEN



1 Block-Textfeld (Schleifen-Name)

2 Namensauswahl

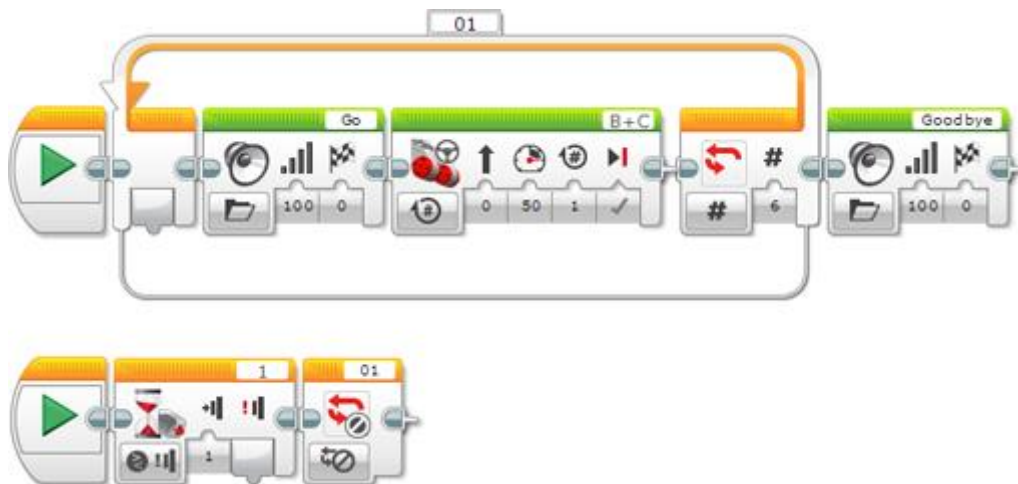
13.5.2 MODUS

Interrupt

Der Schleifen-Interrupt-Block verfügt über einen einzigen Modus. Das Block-Textfeld oben im Block zeigt den Schleifen-Namen der zu unterbrechenden Schleife an. Klicke auf das Block-Textfeld, um das Pop-upmenü einzublenden. Im Pop-upmenü sind die Namen der Schleifen-Blöcke in deinem Programm aufgelistet. Wähle den Namen der zu unterbrechenden Schleife aus der Liste aus.

Wenn der Schleifen-Interrupt-Block ausgeführt wird, dann wird die angegebene Schleife sofort beendet und das Programm springt zu den Blöcken nach der Schleife (sofern vorhanden).

Beispiel



Dieses Programm enthält eine Schleife mit dem Schleifen-Namen „01“, die so konfiguriert ist, dass sie sechsmal wiederholt wird. Eine zweite ausgeführte Blocksequenz wartet jedoch darauf, dass ein Berührungssensor gedrückt wird, und verwendet den Schleifen-Interrupt-Block, um die Schleife vorzeitig zu beenden, wenn der Berührungssensor gedrückt wird.

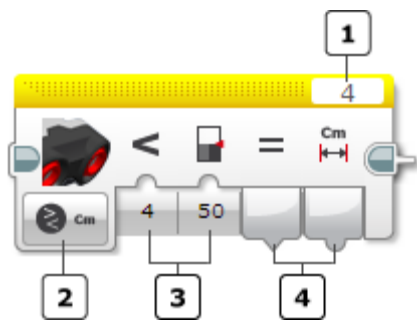
14 SENSOR-BLÖCKE

14.1 ULTRASCHALL



Der Ultraschallsensor-Block ruft Daten vom Ultraschallsensor ab. Du kannst die Distanz in Zentimetern oder Zoll messen und eine Numerische Ausgabe abrufen. Außerdem kannst du die Distanz mit einem Schwellenwert vergleichen, um einen Logischen Wert (Wahr oder Falsch) als Ausgabe zu erhalten.

14.1.1 WÄHLE DEN ANSCHLUSS UND DEN MODUS DES SENSORS AUS



- 1** Anschlussiauswahl
- 2** Modus-Auswahl
- 3** Eingaben
- 4** Ausgaben

Benutze die Anschlussiauswahl oben im Block, um sicherzustellen, dass die Sensor-Anschlussnummer (1, 2, 3 oder 4) dem Anschluss am EV3-Stein entspricht, an den der Ultraschallsensor angeschlossen ist.

Benutze die Modus-Auswahl, um den Modus für den Block auszuwählen.

14.1.2 MODI

Messen – Distanz – Zentimeter



Im Modus „Messen – Distanz – Zentimeter“ wird das Messergebnis unter Distanz in Zentimetern ausgegeben.

Vergleichen – Distanz – Zentimeter



Im Modus „Vergleichen – Distanz – Zentimeter“ wird die Distanz in Zentimetern mit dem Schwellenwert verglichen – und zwar unter Verwendung der ausgewählten Vergleichsart. Die Ausgabe des Ergebnisses – „Wahr“ oder „Falsch“ – erfolgt als Ergebnis des Vergleichs und die Ausgabe der gemessenen Distanz erfolgt unter Distanz in Zentimetern.

Erweitert – Zentimeter



Der Modus „Erweitert – Zentimeter“ ist mit dem Modus „Messen – Distanz – Zentimeter“ vergleichbar, nur dass du hier in der Eingabe Mess-Modus wählen kannst, ob nur ein einziges kurzes Ultraschallsignal (Ping) gesendet werden soll oder ob dauerhaft Ultraschallwellen ausgestrahlt werden sollen. Die Distanz in Zentimetern wird unter Distanz ausgegeben.

14.1.3 EINGABEN UND AUSGABEN

Die für den Ultraschallsensor-Block verfügbaren Eingaben sind vom jeweils ausgewählten Modus abhängig. Du kannst die Eingabewerte direkt in den Block eingeben. Alternativ hierzu können als Eingabewerte auch die Ausgaben anderer Programmierblöcke dienen, die über Datenleitungen übertragen werden.

Eingabe	Typ	Zulässige Werte	Hinweise
Vergleichsart	Numerische Werte	0 bis 5	0: = (Gleich) 1: ≠ (Ungleich) 2: > (Größer als) 3: = (Größer gleich) 4: < (Kleiner) 5: = (Kleiner gleich)
Schwellenwert	Numerische Werte	Beliebige Zahl	Mit den Sensordaten zu vergleichender Wert
Mess-Modus	Numerische Werte	0 oder 1	Modus des Ultraschallsignals in den Erweiterten Modi 0 = Ping

			1 = Ständiges Signal
--	--	--	----------------------

Die verfügbaren Ausgaben sind vom jeweils ausgewählten Modus abhängig. Verwende eine Datenleitung, um die Ausgabe mit einem anderen Programmierblock zu verbinden.

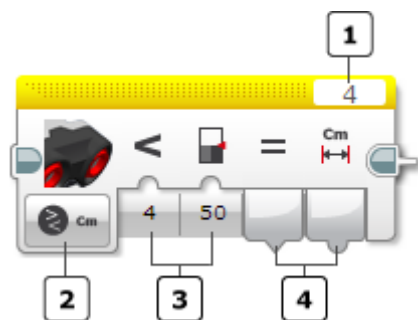
Ausgabe	Typ	Hinweise
Distanz in Zentimetern	Numerische Werte	Distanz in Zentimetern (0 bis 255 cm)
Ergebnis des Vergleichs	Logische Werte	Das von einem Vergleichsmodus gelieferte Ergebnis „Wahr“ bzw. „Falsch“.

14.2 INFRAROT



Der Ultraschallsensor-Block ruft Daten vom Ultraschallsensor ab. Du kannst die Distanz in Zentimetern oder Zoll messen und eine Numerische Ausgabe abrufen. Außerdem kannst du die Distanz mit einem Schwellenwert vergleichen, um einen Logischen Wert (Wahr oder Falsch) als Ausgabe zu erhalten.

14.2.1 WÄHLE DEN ANSCHLUSS UND DEN MODUS DES SENSORS AUS



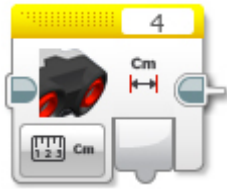
- 1** Anschlussauswahl
- 2** Modus-Auswahl
- 3** Eingaben
- 4** Ausgaben

Benutze die Anschlussauswahl oben im Block, um sicherzustellen, dass die Sensor-Anschlussnummer (1, 2, 3 oder 4) dem Anschluss am EV3-Stein entspricht, an den der Ultraschallsensor angeschlossen ist.

Benutze die Modus-Auswahl, um den Modus für den Block auszuwählen.

14.2.2 MODI

Messen – Distanz – Zentimeter



Im Modus „Messen – Distanz – Zentimeter“ wird das Messergebnis unter Distanz in Zentimetern .

Vergleichen – Distanz – Zentimeter



Im Modus „Vergleichen – Distanz – Zentimeter“ wird die Distanz in Zentimetern mit dem Schwellenwert verglichen – und zwar unter Verwendung der ausgewählten Vergleichsart. Die Ausgabe des Ergebnisses – „Wahr“ oder „Falsch“ – erfolgt als Ergebnis des Vergleichs und die Ausgabe der gemessenen Distanz erfolgt unter Distanz in Zentimetern.

Erweitert – Zentimeter



Der Modus „Erweitert – Zentimeter“ ist mit dem Modus „Messen – Distanz – Zentimeter“ vergleichbar, nur dass du hier in der Eingabe Mess-Modus wählen kannst, ob nur ein einziges kurzes Ultraschallsignal (Ping) gesendet werden soll oder ob dauerhaft Ultraschallwellen ausgestrahlt werden sollen. Die Distanz in Zentimetern wird unter Distanz ausgegeben.

14.2.3 EINGABEN UND AUSGABEN

Die für den Ultraschallsensor-Block verfügbaren Eingaben sind vom jeweils ausgewählten Modus abhängig. Du kannst die Eingabewerte direkt in den Block eingeben. Alternativ hierzu können als Eingabewerte auch die Ausgaben anderer Programmierblöcke dienen, die über Datenleitungen übertragen werden.

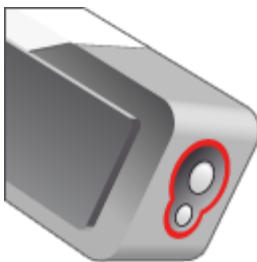
Eingabe	Typ	Zulässige Werte	Hinweise
Vergleichsart	Numerische Werte	0 bis 5	0: = (Gleich) 1: ≠ (Ungleich)

			2: > (Größer als) 3: = (Größer gleich) 4: < (Kleiner) 5: = (Kleiner gleich)
Schwellenwert	Numerische Werte	Beliebige Zahl	Mit den Sensordaten zu vergleichender Wert
Mess-Modus	Numerische Werte	0 oder 1	Modus des Ultraschallsignals in den Erweiterten Modi 0 = Ping 1 = Ständiges Signal

Die verfügbaren Ausgaben sind vom jeweils ausgewählten Modus abhängig. Verwende eine Datenleitung, um die Ausgabe mit einem anderen Programmierblock zu verbinden.

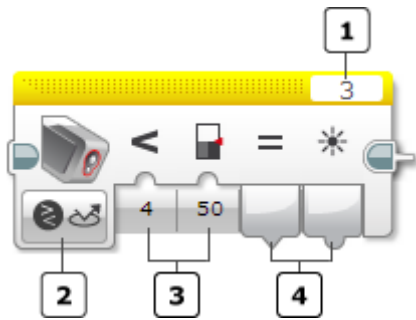
Ausgabe	Typ	Hinweise
Distanz in Zentimetern	Numerische Werte	Distanz in Zentimetern (0 bis 255 cm)
Ergebnis des Vergleichs	Logische Werte	Das von einem Vergleichsmodus gelieferte Ergebnis „Wahr“ bzw. „Falsch“.

14.3 FARBE



Der Farbsensor-Block ruft Daten vom Farbsensor ab. Du kannst die Farbe oder Lichtstärke einer Lichtquelle messen und eine numerische Ausgabe abrufen. Außerdem kannst du Sensordaten mit einem Eingabewert vergleichen und eine logische Ausgabe (Wahr oder Falsch) abrufen.

14.3.1 WÄHLE DEN ANSCHLUSS UND DEN MODUS DES SENSORS



1 Anschlussauswahl

2 Modus-Auswahl

3 Eingaben

4 Ausgaben

Benutze die Anschlussauswahl oben im Block, um sicherzustellen, dass die Sensor-Anschlussnummer (1, 2, 3 oder 4) dem Anschluss am EV3-Stein entspricht, an den der Farbsensor angeschlossen ist.

Benutze die Modus-Auswahl, um den Modus für den Block auszuwählen.

14.3.2 MODI

Messen – Farbe



Im Modus „Messen – Farbe“ wird der erkannte Farbstatus unter Farbe ausgegeben.

Messen – Stärke des reflektierten Lichts



Im Modus „Messen – Stärke des reflektierten Lichts“ wird die Lichtstärke als Messwert ausgegeben.

Messen – Stärke des Umgebungslichts



Im Modus „Messen – Stärke des Umgebungslichts“ wird die Lichtstärke als Messwert ausgegeben.

Vergleichen – Farbe



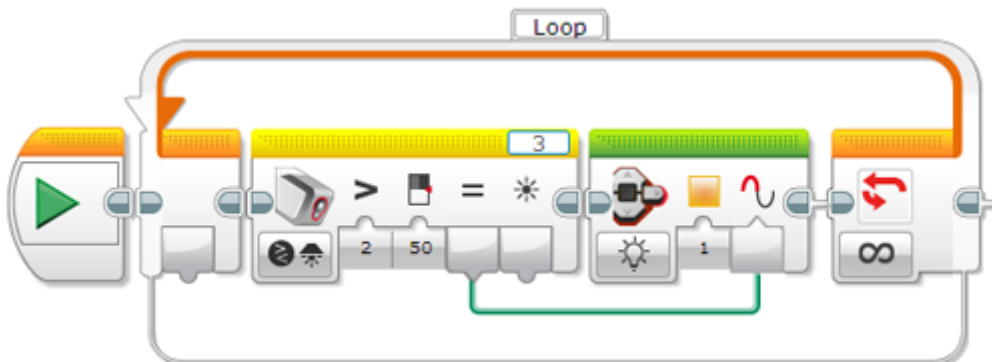
Im Modus „Vergleichen – Farbe“ kannst du eine oder mehrere Farben aus der Menge an Farben auswählen. Als Ergebnis des Vergleichs wird der logische Wert „Wahr“ ausgegeben, wenn die erkannte Farbe mit einer beliebigen ausgewählten Farbe übereinstimmt. Die Ausgabe der erkannten Farbe erfolgt unter Messwert.

Vergleichen – Lichtstärke



In den Modi „Vergleichen – Stärke des reflektierten Lichts“ und „Vergleichen – Stärke des Umgebungslichts“ wird die erkannte Lichtstärke mit dem Schwellenwert verglichen – und zwar anhand der ausgewählten Vergleichsart. Die Ausgabe des Ergebnisses – „Wahr“ oder „Falsch“ – erfolgt als Ergebnis des Vergleichs und die Ausgabe der erkannten Lichtstärke erfolgt unter Messwert.

Beispiel

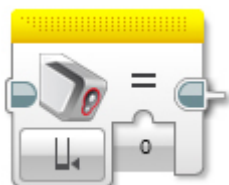


Dieses Programm lässt die Stein-Statusleuchte dauerhaft orange leuchten – und jedes Mal, wenn der Farbsensor eine Umgebungslichtstärke von mehr als 50 erkennt, lässt es die Leuchte blinken.

14.3.3 KALIBRIERUNGSMODI

Die Kalibrierungsmodi ermöglichen dir die Kalibrierung des Farbsensors innerhalb eines Programms. Die vom Sensor zu verwendenden Höchst- und Mindestwerte können auch manuell eingegeben werden.

Kalibrieren – Minimum



Im Modus „Kalibrieren – Minimum“ kannst du die minimale Lichtstärke in der Eingabe Wert festlegen. Nach der Kalibrierung meldet der Farbsensor diese Lichtstärke als 0 bzw. als den festgelegten Wert.

Kalibrieren – Maximum



Im Modus „Kalibrieren – Maximum“ kannst du die maximale Lichtstärke in der Eingabe Wert festlegen. Nach der Kalibrierung meldet der Farbsensor diese Lichtstärke als 100 bzw. als den festgelegten Wert.

Kalibrieren – Zurücksetzen



Im Modus „Kalibrieren – Zurücksetzen“ wird der Farbsensor wieder in den Standardzustand zurückgesetzt, d.h. die Kalibrierung entspricht dann wieder der Standardvorgabe.

14.3.4 EINGABEN UND AUSGABEN

Die für den Farbsensor-Block verfügbaren Eingaben sind vom jeweils ausgewählten Modus abhängig. Du kannst die Eingabewerte direkt in den Block eingeben. Alternativ hierzu können als Eingabewerte auch die Ausgaben anderer Programmierblöcke dienen, die über [Datenleitungen](#) übertragen werden.

Eingabe	Typ	Zulässige Werte	Hinweise
Menge an Farben	Numerisches Array	Jedes Element: 0 bis 7	<p>Zur Erkennung im Modus „Vergleichen – Farbe“ ausgewählte Farbe(n):</p> <p>0 = Keine Farbe</p> <p>1 = Schwarz</p> <p>2 = Blau</p> <p>3 = Grün</p> <p>4 = Gelb</p> <p>5 = Rot</p> <p>6 = Weiß</p> <p>7 = Braun</p>
Vergleichsart	Numerische Werte	0 bis 5	0: = (Gleich)

			1: ≠ (Ungleich) 2: > (Größer als) 3: = (Größer gleich) 4: < (Kleiner) 5: = (Kleiner gleich)
Schwellenwert	Numerische Werte	Beliebige Zahl	Mit den Sensordaten zu vergleichender Wert
Wert	Numerische Werte	0 bis 100	Lichtstärke für Kalibrierungsmodi

Die verfügbaren Ausgaben sind vom jeweils ausgewählten Modus abhängig. Verwende eine Datenleitung, um die Ausgabe mit einem anderen Programmierblock zu verbinden.

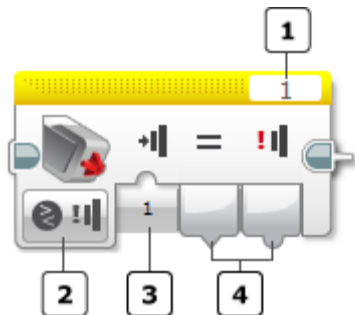
Ausgabe	Typ	Hinweise
Farbe	Numerische Werte	Erkannter Farbstatus: 0 = Keine Farbe 1 = Schwarz 2 = Blau 3 = Grün 4 = Gelb 5 = Rot 6 = Weiß 7 = Braun
Ergebnis des Vergleichs	Logische Werte	Das von einem Vergleichsmodus gelieferte Ergebnis „Wahr“ bzw. „Falsch“.
Messwert	Numerische Werte	Der in einem Vergleichsmodus verwendete Sensordatenwert.

14.4 BERÜHRUNG



Der Berührungssensor-Block ruft Daten vom Berührungssensor ab. Du kannst testen, ob der Berührungssensor „gedrückt“, „ausgelassen“ oder „angestoßen“ wird. Zudem kannst du einen logischen Wert (Wahr oder Falsch) als Ausgabe abrufen.

14.4.1 WÄHLE DEN ANSCHLUSS UND DEN MODUS DES SENSORS



- 1** Anschlussiauswahl
- 2** Modus-Auswahl
- 3** Eingaben
- 4** Ausgaben

Benutze die Anschlussiauswahl oben im Block, um sicherzustellen, dass die Sensor-Anschlussnummer (1, 2, 3 oder 4) dem Anschluss am EV3-Stein entspricht, an den der Berührungssensor angeschlossen ist.

Benutze die Modus-Auswahl, um den Modus für den Block auszuwählen.

14.4.2 MODI

Messen – Zustand



Im Modus „Messen – Zustand“ wird mit der Ausgabe Zustand der Zustand des Berührungssensors angezeigt. Der Zustand lautet „Wahr“, wenn der Berührungssensor derzeit gedrückt wird, und „Falsch“, wenn dies nicht der Fall ist.

Vergleichen – Zustand



Im Modus „Vergleichen – Zustand“ kannst du in der Eingabe Zustand auswählen, welcher Zustand des Berührungssensors getestet werden soll („gedrückt“, „ausgelassen“ oder „angestoßen“). Der logische Wert „Wahr“/„Falsch“ für den ausgewählten Zustand wird unter Ergebnis des Vergleichs ausgegeben. Der aktuelle Zustand des Sensors wird unter Messwert ausgegeben.

14.4.3 EINGABEN UND AUSGABEN

Die für den Block „Berührungssensor“ verfügbaren Eingaben sind vom jeweils ausgewählten Modus abhängig. Du kannst die Eingabewerte direkt in den Block eingeben. Alternativ hierzu können als Eingabewerte auch die Ausgaben anderer Programmierblöcke dienen, die über Datenleitungen übertragen werden.

Eingabe	Typ	Zulässige Werte	Hinweise
Zustand	Numerische Werte	0 bis 2	Im Vergleichsmodus zu erkennender Zustand 0 = Ausgelassen 1 = Gedrückt 2 = Angestoßen

Die verfügbaren Ausgaben sind vom jeweils ausgewählten Modus abhängig. Verwende eine Datenleitung, um die Ausgabe mit einem anderen Programmierblock zu verbinden.

Ausgabe	Typ	Hinweise
Zustand	Logische Werte	Wird im Messmodus verwendet. „Wahr“, wenn der Berührungssensor gedrückt wird, „Falsch“ wenn der Berührungssensor nicht gedrückt wird.
Ergebnis des Vergleichs	Logische Werte	Der Wert des im Vergleichsmodus ausgewählten Sensorzustands
Messwert	Numerische Werte	Der aktuelle Sensorzustand im Modus „Vergleichen“ 0 = Ausgelassen 1 = Gedrückt 2 = Angestoßen

15 DATEN-BLÖCKE

15.1 VARIABLE



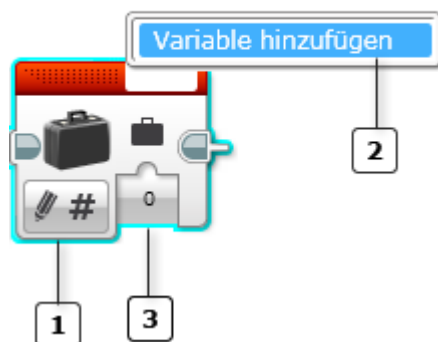
Der Variablen-Block gestattet es dir, eine Variable in deinem Programm zu lesen oder zu schreiben. Du kannst aber auch eine neue Variable erstellen und benennen.

Bei einer Variablen handelt es sich um einen Speicherort im Speicher des EV3-Steins, an dem ein Datenwert aufbewahrt werden kann. Du kannst in eine Variable schreiben, um einen Datenwert zu speichern. An späterer Stelle im Programm kannst du aus der Variablen lesen, um auf den gespeicherten Wert zuzugreifen.

Jede Variable verfügt über einen Typ und einen Namen. Es gibt folgende Typen: Numerische Variable, Logische Variable, Text-Variable, Numerisches Array und Logisches Array. Du kannst den Namen der Variablen wählen, der verwendet wird, um die Variable zu identifizieren.

Der Wert einer Variablen kann geändert werden, während ein Programm ausgeführt wird. Jedes Mal, wenn du in eine Variable schreibst, wird jeglicher vorheriger Wert gelöscht und durch den neuen Wert ersetzt. Du könntest bspw. eine Variable namens „Max. Licht“ verwenden, um zu verfolgen, welche maximale Lichtstärke dein Roboter bisher mithilfe des Farbsensor gemessen hat. Jedes Mal, wenn der Roboter eine größere Lichtstärke erkennt, könnte er den neuen Wert in die Variable „Max. Licht“ schreiben.

15.1.1 NEUE VARIABLE HINZUFÜGEN

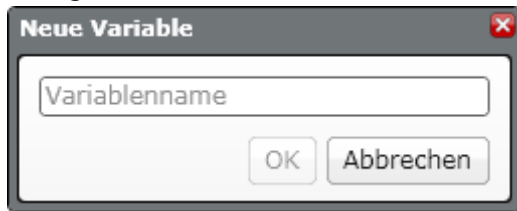


- 1 Modus-Auswahl
- 2 Variablenauswahl
- 3 Wert-Eingabe

So fügst du eine neue Variable zu deinem Projekt hinzu:

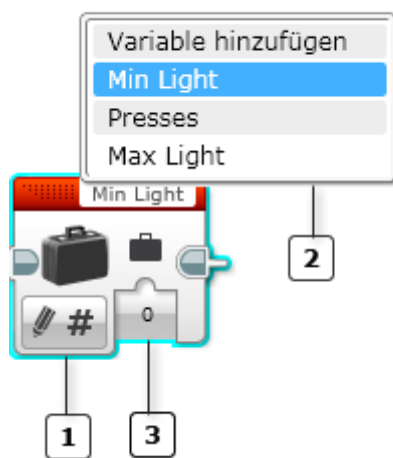
1. Variablen-Block einfügen oder auswählen.
2. Verwende die Modus-Auswahl, um den Schreibmodus auszuwählen.
3. Wähle den Variablentyp (Numerische Variable, Logische Variable, Text-Variable, Numerisches Array oder Logisches Array) aus.

- Klicke auf das Block-Textfeld oben im Block und wähle die Option „Variable hinzufügen“. Das Dialogfenster „Neue Variable“ wird eingeblendet (siehe unten).



- Gib in das Dialogfenster „Neue Variable“ einen Namen für deine Variable ein und klicke auf „OK“. Ein Variablenname kann aus einem einzigen Buchstaben, aus einem Wort, aus mehreren Wörtern oder aus einer beliebigen Zeichenfolge aus Buchstaben und Zahlen bestehen.

15.1.2 IN EINE VARIABLE SCHREIBEN



- Modus-Auswahl
- Variablenname
- Eingabe

Schreiben

Im Schreibmodus kannst du eine Variable auswählen, die du bereits zu deinem Programm hinzugefügt hast und einen Wert für diese Variable speichern.

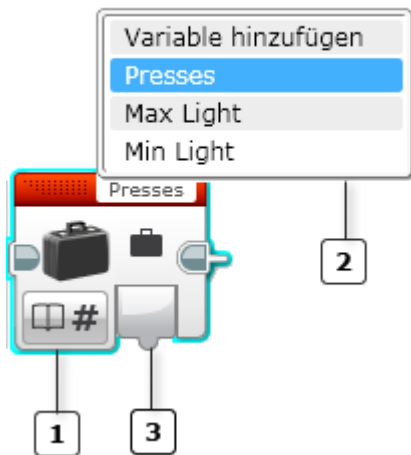
So wird in eine Variable geschrieben:

- Verwende die Modus-Auswahl, um den Schreibmodus auszuwählen
- Wähle den Variablentyp aus.
- Klicke auf das Block-Textfeld, um das Popupmenü einzublenden.
- Wähle den Namen der zu verwendenden Variablen.
- Jetzt kannst du einen Wert in der Variablen speichern – und zwar unter Verwendung der Eingabe Wert. Du kannst den Wert entweder direkt in die Eingabe Wert eingeben oder den Wert aus einer Datenleitung übernehmen.

Tipps und Tricks

Du kannst beliebig oft in eine Variable schreiben. Allerdings wird jeweils nur der letzte Wert dort gespeichert. Wenn du in eine Variable schreibst, wird der vorherige Wert der Variablen gelöscht und durch den neuen Wert ersetzt.

15.1.3 AUS EINER VARIABLEN LESEN



- 1 Modus-Auswahl
- 2 Variablenname
- 3 Ausgabe

Lesen

Im Lesemodus kannst du eine Variable auswählen, die du bereits zu deinem Programm hinzugefügt hast und den Wert abrufen, den du in der Variablen gespeichert hast.

So wird aus einer Variablen gelesen:

1. Verwende die Modus-Auswahl, um den Lesemodus und den Variablentyp auszuwählen.
2. Klicke auf das Textfeld oben im Block, um das Popupmenü einzublenden, und wähle den Namen der Variablen, die du verwenden möchtest.
3. Jetzt kannst du den Wert der Variablen aus der Ausgabe Wert per Datenleitung abrufen und in deinem Programm verwenden.

Tipps und Tricks

Du musst erst den Variablentyp aus der Modus-Auswahl auswählen, bevor du den Variablennamen aus dem Popupmenü auswählst. Im Popupmenü sind nur die Variablen aufgelistet, die dem ausgewählten Variablentyp entsprechen.

Wenn du aus einer Variablen liest, in die noch nie ein Wert geschrieben wurde, dann ist das Ergebnis für eine Numerische Variable „null“, für eine Logische Variable „Falsch“, für eine Text-Variable leerer Text und für die Variablen „Numerisches Array“ und „Logisches Array“ ein leeres Array.

15.1.4 BEISPIELE FÜR DIE VERWENDUNG VON VARIABLEN

Beispiel 1: Einen Wert speichern, um ihn später zu verwenden



In diesem Programm wird ein Variablen-Block im Schreibmodus verwendet, um einen Datenwert vom Farbsensor in der Variablen namens „Light“ (Licht) zu speichern. Dann wird ein Variablen-Block im Lesemodus verwendet, um den Wert abzurufen und später im Programm zu benutzen.

Das Programm ist so konzipiert, dass es einen Roboter herausfahren und wieder zurückfahren lässt. Auf dem Rückweg wird der Roboter von einer Farbe gestoppt, die mindestens genauso dunkel wie die Farbe ist, die ihn hat losfahren lassen.

Tipps und Tricks

Das oben beschriebene Programm könnte auch mit einer langen Datenleitung anstelle einer Variablen ausgeführt werden, aber bei langen Programmen bietet es sich an, Werte in Variablen zu speichern, weil das die Arbeit mit dem Programm erleichtert.

Beispiel 2: Zählen, wie oft der Berührungssensor gedrückt wird



Dieses Programm zählt, wie oft ein Berührungssensor gedrückt wird, und zeigt dann den Wert der Zählung auf dem EV3-Display an. Es verwendet eine Variable namens „Presses“ (Druck-Vorgänge), um zu verfolgen, wie oft der Berührungssensor gedrückt wurde.

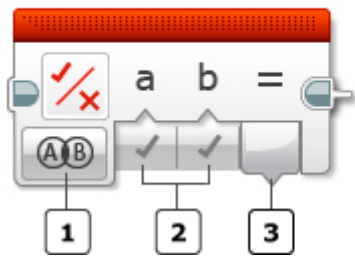
Ein Variablen-Block im Schreibmodus setzt den Anfangswert der Variablen „Presses“ zu Beginn des Programms auf 0. Jedes Mal, wenn der Berührungssensor angestoßen wird, addiert das Programm den Wert 1 zu dem in der Variablen gespeicherten Wert – und zwar innerhalb der Schleife. Dies wird folgendermaßen bewerkstelligt: Der aktuelle Wert wird aus einem Variablen-Block im Lesemodus abgerufen, zu diesem Wert wird mithilfe eines Mathe-Blocks und anhand von Datenleitungen der Wert 1 addiert und schließlich wird dieses Ergebnis unter Verwendung eines Variablen-Blocks im Schreibmodus wieder in die Variable geschrieben. Zu guter Letzt wird eine weitere Variablen-Block im Lesemodus benutzt, um den aktuellen Wert abzurufen und anzuzeigen.

15.2 LOGISCHE WERTE



Der Block „Logische Verknüpfungen“ unterzieht seine Eingaben einer Logischen Operation und gibt dann das Ergebnis aus. Eingaben, die „Wahr“ oder „Falsch“ sind, werden einer Logischen Operation unterzogen, die als Ergebnis die Ausgabe „Wahr“ oder „Falsch“ liefert. Die verfügbaren Logischen Verknüpfungen sind UND, ODER, EXKLUSIVES ODER und NICHT.

15.2.1 LOGISCHE VERKNÜPFUNGEN







1 Modus-Auswahl

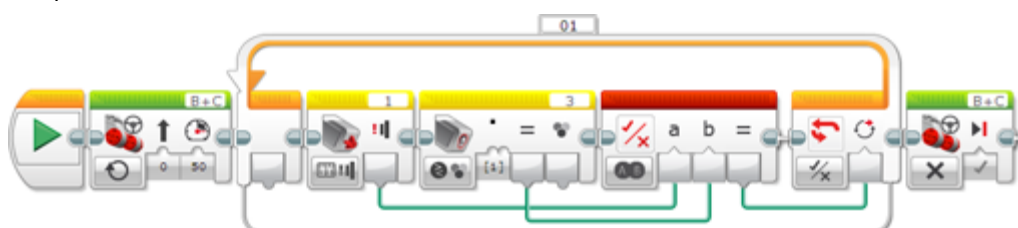
2 Eingaben

3 Ausgabe

Wähle die zu verwendende Logische Verknüpfung, indem du mithilfe der Modus-Auswahl einen Modus auswählst. Der Block berechnet das Ergebnis auf Grundlage der Eingaben – wie in dieser Tabelle dargestellt.

Modi	Verwendete Eingaben	Ergebnis
 UND	A, B	„Wahr“, wenn sowohl A als auch B „Wahr“ sind, anderenfalls „Falsch“
 ODER	A, B	„Wahr“, wenn entweder A oder B (bzw. A und B) „Wahr“ ist, „Falsch“, wenn sowohl A als auch B „Falsch“ sind.
 EXKLUSIVES ODER	A, B	„Wahr“, wenn ausschließlich A oder B „Wahr“ ist, „Falsch“, wenn sowohl A als auch B „Wahr“ sind, „Falsch“, wenn sowohl A als auch B „Falsch“ sind.
 NICHT	A	„Wahr“, wenn A „Falsch“ ist, „Falsch“, wenn A „Wahr“ ist.

Beispiel



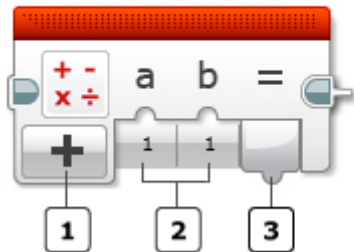
Dieses Programm lässt einen Roboter vorwärts fahren, bis entweder der Berührungssensor gedrückt wird oder der Farbsensor die Farbe Schwarz erkennt. Die Logische Verknüpfung „ODER“ wird verwendet, um aus den Ausgaben der beiden Sensorblöcke ein einziges Ergebnis – „Wahr“ oder „Falsch“ – zu gewinnen. Das Ergebnis „Wahr“ beendet die Schleife, woraufhin der Roboter gestoppt wird.

15.3 MATHE



Der Mathe-Block nimmt eine mathematische Berechnung seiner Eingaben vor und gibt dann das Ergebnis dieser Berechnung aus. Du kannst eine einfache mathematische Operation mit ein oder zwei Eingaben ausführen oder eine Formel mit bis zu vier Eingaben verwenden.

15.3.1 WÄHLE DIE MATHEMATISCHE OPERATION AUS










- 1** Modus-Auswahl
- 2** Eingaben
- 3** Ausgabe

Wähle die zu verwendende mathematische Operation, indem du mithilfe der Modus-Auswahl einen Modus auswählst. Nach der Auswahl des Modus kannst du die Werte für die Eingaben festlegen. Die verfügbaren Eingaben sind von Modus zu Modus verschieden.

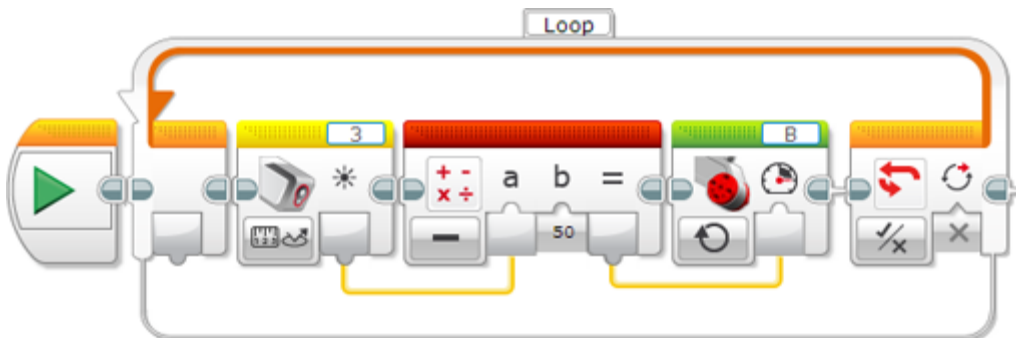
15.3.2 EINFACHE MATHEMATISCHE OPERATIONSMODI

Die einfachen mathematischen Operationsmodi berechnen das Ergebnis anhand einer einzigen mathematischen Berechnung und unter Verwendung von ein oder zwei Eingaben. Diese Modi sind in der nachstehenden Tabelle dargestellt.

Modus	Verwendete Eingaben	Ausgegebenes Ergebnis
 Addieren	A, B	$A + B$
 Subtrahieren	A, B	$A - B$

Subtrahieren		
 Multiplizieren	A, B	$A \times B$
 Dividieren	A, B	$A \div B$
 Absoluter Wert	A	A, wenn $A \geq 0$; -A, wenn $A < 0$ Das Ergebnis ist immer ≥ 0 .
 Quadratwurzel	A	\sqrt{A}
 Exponent	A (Basis), N (Exponent)	A^N

Beispiel



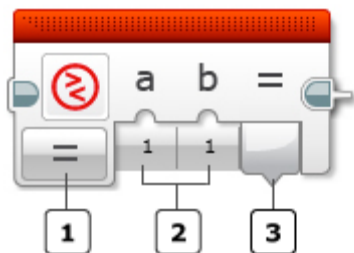
Dieses Programm subtrahiert die Zahl 50 von der Ausgabe „Stärke des reflektierten Lichts“ des Farbsensor-Blocks und verwendet das Ergebnis als Eingabe „Leistung“ für einen Motor. Auf diese Weise lässt das Programm den Motor rückwärts drehen, wenn der Farbsensor etwas Dunkles erkennt, und vorwärts drehen, wenn der Sensor etwas Helles wahrnimmt.

15.4 VERGLEICHEN



Der Vergleichs-Block vergleicht zwei Zahlen, um festzustellen, ob sie gleich sind bzw. welche Zahl größer ist. Du kannst zwischen sechs verschiedenen Vergleichen auswählen. Das ausgegebene Ergebnis lautet „Wahr“ oder „Falsch“.

15.4.1 VERGLEICHSMODI









1 Modus-Auswahl

2 Eingaben

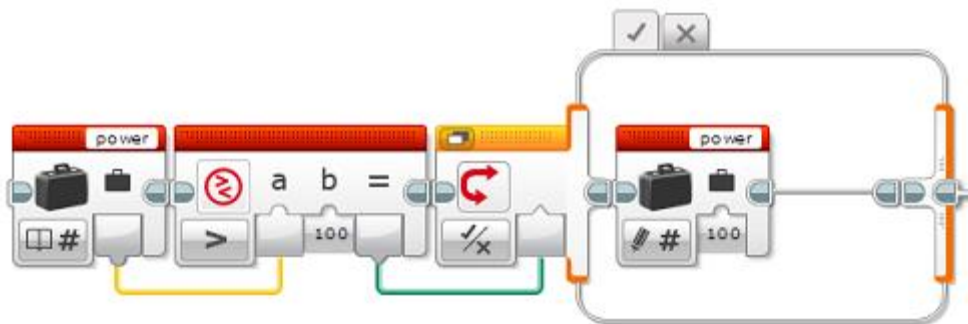
3 Ausgabe

Wähle die zu verwendende Vergleichsart, indem du mithilfe der Modus-Auswahl einen Modus auswählst. Der Block berechnet dann die Ausgabe Ergebnis, indem er die beiden Eingaben A und B miteinander vergleicht, wie es in der nachstehenden Tabelle dargestellt ist.

Modus	Verwendete Eingaben	Ausgegebenes Ergebnis
 Gleich	A, B	Wahr, wenn $A = B$, ansonsten Falsch
 Ungleich	A, B	Wahr, wenn $A \neq B$, andernfalls Falsch
 Größer als	A, B	Wahr, wenn $A > B$, andernfalls Falsch
 Kleiner als	A, B	Wahr, wenn $A < B$, andernfalls Falsch

 Größer gleich	A, B	Wahr, wenn $A \geq B$, andernfalls Falsch
 Kleiner gleich	A, B	Wahr, wenn $A \leq B$, andernfalls Falsch

Beispiel



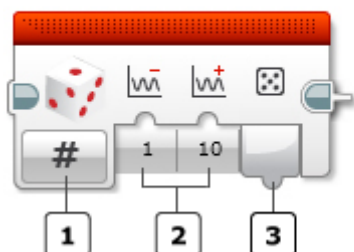
Diese Blocksequenz testet, ob der Wert der Variablen „Leistung“ größer als 100 ist. Ist dies der Fall, setzt sie den Wert auf 100. Der Vergleichs-Block vergleicht den Wert der Variablen mit dem Wert 100, und der Logische Wert (das Ergebnis dieses Vergleichs) wird vom Schalter verwendet, um zu entscheiden, ob der Wert der Variablen geändert werden muss.

15.5 ZUFALL



Der Zufalls-Block kann einen Numerischen oder Logischen Zufallswert ausgeben. Du kannst das Ergebnis des Zufalls-Blocks verwenden, um deinen Roboter eine Zufallsauswahl aus verschiedenen Aktionen treffen zu lassen.

15.5.1 WÄHLE DEN AUSGABEN-TYP AUS



1 Modus-Auswahl

2 Eingaben

3 Ausgabe

Benutze die Modus-Auswahl, um festzulegen, ob ein Numerischer oder Logischer Zufallswert ausgegeben werden soll. Nach der Auswahl des Modus kannst du die Eingaben festlegen. Mit den Eingaben werden der Bereich und die Wahrscheinlichkeit der Ausgabe Wert bestimmt.

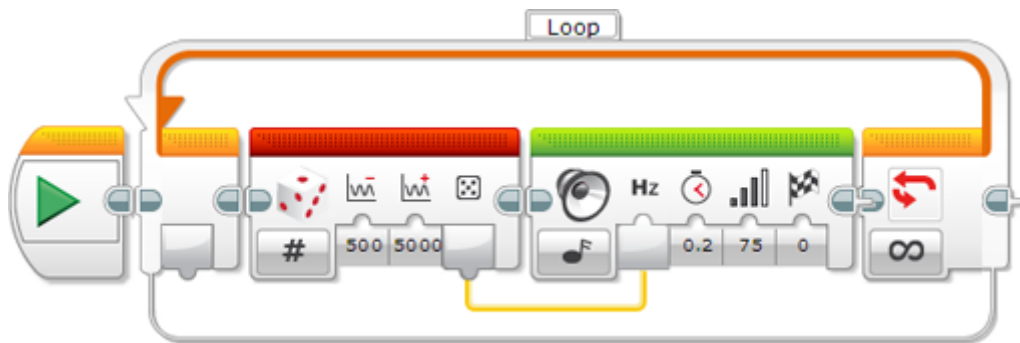
15.5.2 MODI

Numerische Werte



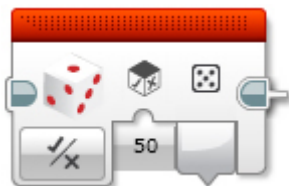
Im Numerischen Modus wird ein ganzzahliger Zufallswert ausgegeben. Als Wert wird eine zufällige Ganzzahl eingestellt, die innerhalb des Bereichs liegt, der durch die Untere Begrenzung und die Obere Begrenzung definiert ist. Die Auswahl kann auf jeden ganzzahligen Wert innerhalb dieses Bereichs fallen – und zwar mit derselben Wahrscheinlichkeit.

Beispiel



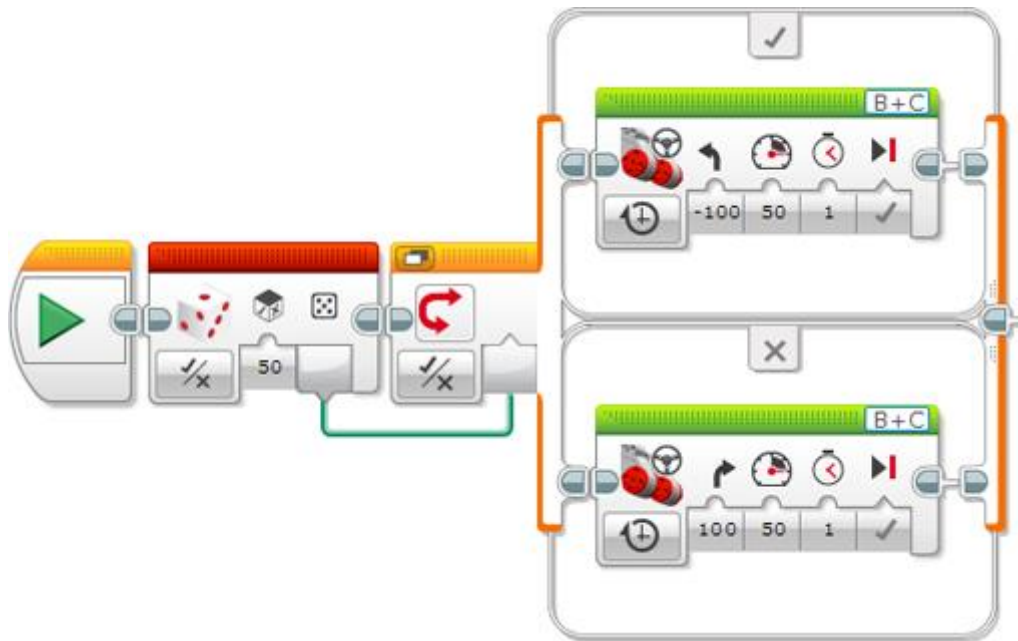
Dieses Programm erzeugt eine endlose Serie von Zufallstönen aus dem Frequenzbereich von 500 bis 5000 Hz.

Logische Werte



Im Logischen Modus wird der Wert „Wahr“ oder „Falsch“ ausgegeben. Die Wahrscheinlichkeit, dass der Wert „Wahr“ ausgegeben wird, ist durch die Eingabe Wahrscheinlichkeit für „Wahr“ festgelegt. Bei dieser Eingabe handelt es sich um einen Prozentwert von 0 bis 100. Eine Wahrscheinlichkeit von 25 würde bspw. bedeuten, dass eine Chance von 25 % besteht, dass die Ausgabe den Wert „Wahr“ liefert, und dass eine Chance von 75 % besteht, dass die Ausgabe den Wert „Falsch“ enthält.

Beispiel



Dieses Programm lässt den Roboter eine Zufallsauswahl treffen, ob er nach links oder rechts fährt. Für beide Optionen besteht eine 50-Prozent-Chance.

Tipps und Tricks

Um deinen Roboter eine Zufallsauswahl aus mehr als zwei Aktionen treffen zu lassen, kannst du den Zufalls-Block im Numerischen Modus sowie einen Schalter-Block im Numerischen Modus verwenden. Du könntest bspw. eine Zufallszahl zwischen 1 und 5 ausgeben lassen und diesen Wert verwenden, um einen von fünf verschiedenen Fällen im Schalter auszuwählen.

16 LITERATUR

LEGO MINDSTORMS EV3 Home Edition. (2013). (Version 1.1.1). DK-7190 Billund, Dänemark: LEGO Group.