

## Block 3: Aufgabenideen

Die folgende Sammlung von Aufgabenideen basiert auf der Einführung in die Programmierung (Block 2 dieser Unterrichtseinheit). Ideen können auch im Education EV3 Classroom bei den Lerneinheiten und den 'Modelle des EV3 Sets' gefunden werden. Es wird empfohlen, dass die SuS zur Lösung einer Aufgabe nach Bedarf die dazu nötigen Lösungen studieren. Es macht keinen Sinn diese alle oder teilweise auf Vorrat im Voraus zu erlernen. Viel zielgerichteter und näher an der Praxis ist es, die Lösungsprinzipien nach Bedarf zu studieren. In der folgenden Tabelle wird dokumentiert, welche Programmfunktionen für die jeweilige Beispielaufgabe gebraucht werden kann.

Aufgabe	Konstruktion	Programm	Schlaufe	Einfachschalter	Parallelität	Rechenoperationen	Variablen	Signallinien	eigener Baustein	Sensorkalibrierung
Ballwurfgerät	mittel	einfach								
Fahrroboter A	einfach	einfach-mittel								
Fahrroboter B	einfach	mittel	x	x						
Fahrroboter C	einfach	mittel-schwierig	x	x		x				
Zeichnungsroboter	mittel	einfach-mittel	x							
einfaches Sortieren	mittel	mittel	x	x						
Musikerroboter	einfach	einfach-schwierig	x		x					
Lichtsucher	einfach	einfach	x							
Plotter	mittel	einfach-schwierig	x			(x)				
Fernsteuerung mit Licht	einfach	einfach-mittel	x	x						
Fahren im Kreis	einfach	einfach-mittel	x			(x)				
Linienfolger mit Hindernis	einfach	schwierig	x	x						
Sortieren	mittel	schwierig	x			x	x	x	x	
Roboterrennen	einfach	mittel-schwierig	x	x		(x)		(x)	(x)	(x)
Suchroboter	mittel	schwierig	x	x			x	x	x	
Rohrroboter	einfach	mittel	x			x		x		
Schnitzeljagd	einfach	mittel	x							
Tanzroboter	einfach bis ..	einfach-schwierig	x		x	x	x	x	x	
Roboterkampf	einfach bis ..	einfach-schwierig	x	x						

Die Schwierigkeit einer Aufgabe kann je nach Ausführung sehr unterschiedlich sein. ‚Einfach-schwierig‘ bedeutet, dass man sowohl eine sehr einfache Lösung als auch eine schwierige realisieren kann. Man kann auch einfach beginnen und die Lösung fortlaufend ausbauen. Die Aufgabensammlung beinhaltet auch herausfordernde Aufgaben für SuS, die bereits über gute Programmierkenntnisse verfügen.

## Grundausrüstung

Folgende Hilfsmittel können für verschiedene Aufgaben eingesetzt werden:

- Brett 1.0 m x 1.5 m weiss beschichtet,
- schwarzes Klebeband
- verschiedene Objekte in der Grösse eines Trinkbechers

## Mögliche Lernziele

Die Schülerinnen und Schüler ...

1. üben die Handhabung der Software und die Übertragung von Programmen auf den Roboter.
2. können ihre Grundfähigkeiten des Programmierens aus Block 2 anwenden und vertiefen.
3. entwickeln ihre Fähigkeiten zur Programmierung des Roboters weiter.
4. analysieren bestehende Programmvorschläge (im Menu 'komplexere Programme') hinsichtlich Funktion.
5. entwickeln Strategien des Trouble-Shootings (wie gehe ich vor, wenn der Roboter nicht das macht, was ich erwarte).
6. können den Basisroboter zielgerichtet umbauen.
7. erkennen welche baulichen Funktionalitäten ein Roboter zur Lösung einer Problemstellung braucht.
8. vertiefen ihre Kenntnisse zur Funktionsweise und zu den Einsatzmöglichkeiten der Sensoren.
9. können im Team eine (oder mehrere) Aufgabenstellung angemessener (oder zunehmender) Schwierigkeit lösen.
10. können ihre Lösung einer Aufgabe anderen SuS demonstrieren und erklären.
11. entwickeln Freude im Umgang mit den Robotern.
12. entwickeln ein Interesse an Möglichkeiten mobiler Robotik.
13. arbeiten möglichst eigenständig.

## Didaktische Hinweise

Die obige Auflistung zeigt die Vielfältigkeit der Fördermöglichkeiten in diesem Unterrichtsblock. Daraus wird auch ersichtlich, dass sich hier viele Möglichkeiten der **Differenzierung** zwischen unterschiedlichen Schülerinnen und Schülern bzw. unterschiedlichen Leistungszügen ergeben.

Während im Leistungszug A eher die Lernziele 1,2,6,7,9 und 11 im Zentrum stehen werden, darf im Leistungszug P sicher auch eine Entwicklung der Lernziele 3,4,5,8 und 13 erwartet werden.

Empfehlen Sie den Jugendlichen **mit einer einfachen Problemstellung zu beginnen** (z.B. indem Sie ihnen 2 oder 3 Beispiele dieser Sammlung kurz vorstellen). Der Erfolg darin wird sie motivieren, dann mit der nötigen Hartnäckigkeit auch an einem schwierigeren Problem zu arbeiten. Je nach Unterrichtszeit, die Sie für diesen Block einsetzen, können alle Gruppen 2-3 Probleme bearbeiten.

Natürlich können die Schülerinnen und Schüler auch eigene Problemstellungen identifizieren und bearbeiten. Achten Sie darauf, dass sich die Gruppen dabei realistische Aufgaben vornehmen. Zum Teil können einfach erscheinende Problemstellungen ziemlich schwierig zu bearbeiten sein. Das **Arbeitsblatt 11\_Aufgabenanalyse.docx** dient den Schülerinnen und Schülern, sowie Ihnen als Lehrperson, der **richtigen Einordnung**. Jede Gruppe füllt jedes Mal vor der Bearbeitung einer Aufgabe dieses Blatt aus und bespricht es mit Ihnen.

Vermutlich sind Sie bei der erstmaligen Durchführung dieses Moduls eher noch Anfänger/Anfängerin als Experte/Expertin in Lego-Robotik. Haben Sie keine Angst davor und machen Sie auch kein Geheimnis daraus. **Sie müssen auch nicht Experte/Expertin sein, Sie sind Lerncoach** der Jugendlichen. Sie brauchen selber nicht alle Antworten zu kennen, Sie können den Schülerinnen und Schülern aber dabei helfen, selber die richtigen Antworten zu finden, indem Sie

- kritisch hinterfragen, was sie genau machen,
- die SuS darauf hinweisen ganz genau hinzuschauen und genau zu beobachten,
- die SuS darauf hinweisen systematisch vorzugehen,
- die Planung der SuS kritisch begutachten,
- auf Möglichkeiten hinweisen, das Problem anders anzugehen oder zu vereinfachen,
- etc.

Darin sind sie Experte/Expertin, auch wenn Sie kein Lego-Robotik Experte/Expertin sind. Und nachdem Sie das Modul ein, zweimal mit einer Klasse durchgeführt haben, werden Sie allmählich auch zum Experten/zur Expertin werden.

Je nach Vorlieben, Rahmenbedingungen und Bedürfnissen von Ihnen und Ihrer Klasse können Sie diesen Block ausdehnen oder verkürzen. Es ist insbesondere auch **möglich, diesen Block parallel zum Block 4 durchzuführen**. Dies kann je nach Situation eine interessante Option sein.

Dieser dritte (wie auch der vierte) Block bietet eine gute Möglichkeit, die Arbeit der Schülerinnen und Schülern zu **beurteilen** und zu **bewerten**. In der Folge wird ein mögliches **Beurteilungsraster** vorgeschlagen. Je nach Leistungszug und Schwerpunktsetzung können Sie auch einzelne Kriterien weglassen oder verstärken (z.B. durch doppelte Gewichtung) bzw. weitere Kriterien ergänzen.

Kriterium	Prädikat			
	ungenügend (0)	genügend (1)	gut (2)	sehr gut (3)
<b>Herangehensweise:</b> Wurde die Aufgabe vor Beginn angemessen analysiert? War eine klare Vorstellung erkennbar, wie die Aufgabe gelöst werden kann? Wurde während der Bearbeitung immer wieder geklärt, ob man auf dem richtigen Weg ist, oder ob eine Plananpassung angezeigt ist? Kann das Team rückblickend beschreiben, wie sie vorgegangen sind? Etc.				
<b>Konstruktion:</b> Ist der Aufbau und die Konstruktionsweise des Roboters für die Lösung des Problems angemessen? (also funktional, stabil bzw. flexibel genug, nicht unnötig kompliziert oder aufwendig, etc.)				
<b>Programmstruktur:</b> Ist die Programmstruktur durchdacht? Macht die Sequenz der Programmbausteine und ihre Kombination Sinn? Gäbe es Möglichkeiten das Programm zu vereinfachen? Sind sinnvolle Programmelemente vorhanden (auch wenn die Aufgabe ev. nicht abschliessend gelöst werden konnte)? Etc.				
<b>Verständnistiefe:</b> Hat das Team verstanden warum genau der Roboter bzw. das Programm genau das ausführt, was beobachtet werden kann? Kann das Team den Programmablauf in Worten erklären? Kann das Team begründen, weshalb eine bestimmte Aufbauweise des Roboters oder ein Programmablauf anderen Varianten vorzuziehen ist?				
<b>Eigenständigkeit:</b> Hat das Team im Rahmen seiner Möglichkeiten selbständig gearbeitet? Hatten sie eine Vorstellung davon, welche Hilfe sie genau benötigten, wenn sie Hilfe in Anspruch genommen haben? Konnte das Team selbständig Entscheidungen treffen und dann umsetzen? Etc.				
<b>Funktionieren der Gruppe:</b> Hat die Gruppe als Team funktioniert? Haben sie sich gegenseitig unterstützt? Wurden Probleme gemeinsam gelöst? Wurden				

unterschiedliche Stärken der Teammitglieder berücksichtigt? Etc.				
<b>Lösung der Aufgabe:</b> Hat das Team die Aufgabe gelöst? Sind kaum Lösungsansätze zu erkennen? Sind Teilaspekte der Aufgabe gelöst? Ist die Aufgabe vollständig gelöst? Ist die Aufgabe sinnvoll adaptiert worden? Etc.				
<b>Bonuspunkte:</b> z.B. 0 Bonuspunkte für einfache Aufgabe, 2 Bonuspunkte für eine mittelschwere Aufgabe, 4 Bonuspunkte für eine schwierige Aufgabe, sofern dies zumindest teilweise gelöst wurden.				

## Aufgaben

In der Folge werden die einzelnen Aufgabenideen kurz umschrieben und es wird auch eine mögliche Lösung vorgestellt. In den meisten Fällen enthält diese Lösung noch ein paar Besonderheiten, die bereits weiterführen als die Basisaufgabe. Falls eine Aufgabe nicht richtig gelöst werden kann, können Sie diesen Lösungsvorschlag somit analysieren ggf. vereinfachen und damit das zur Lösung notwendige Verständnis aufbauen.

### 1. Ballwurfgerät

**Beschreibung:** Es wird ein Fahrroboter gebaut, der einen Ball werfen kann. Der Roboter fährt von einem Startpunkt bis zu einer schwarzen Linie, stoppt dort und versucht den Ball in einen Behälter (ca. 20 - 40 cm entfernt) zu werfen. Natürlich kann zuerst auch nur das Werfen alleine umgesetzt werden.

**Funktionsablauf:** Das Wurfobjekt wird in die Abschussvorrichtung des Roboters gelegt. Nach dem Drücken der Starttaste oder Berührungsschalters fährt der Roboter bis zur schwarzen Linie und stoppt. Mit einem weiteren Motor wird die Abwurfvorrichtung so gedreht, dass der Ball möglichst im Behälter landet. Über Geschwindigkeit und Weg wird der Ballwurf gesteuert. Durch Ausprobieren soll eine funktionierende Lösung gefunden werden. Nach dem Abwurf kann der Roboter wieder eine gewünschte Strecke zurück fahren.

**Material:** EV3 Basic-Fahrroboter, 1 Motor für Wurfvorrichtung,

**Zusatzmaterial:** Wurfball (Pingpong-Ball ist gut geeignet), kleine Gummiringe

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Mittel	einfach	mittel

Beispiellösung:



Abbildung 1: einfacher Ballwurfroboter

Beispiel-Programm:



Abbildung 2: Programm Ballwurfmaschine

Weiterentwicklungsmöglichkeiten:

- Der Roboter fährt entlang Linie zur Wurfposition (2. Lichtsensor nötig)
- Der Roboter richtet sich an der Linie aus durch Vorwärts und Rückwärtsfahren bis beide Sensoren den gleichen Grauwert anzeigen.
- Der Abstand des Auffangbehälters kann mit Ultraschallsensor gemessen werden und die Wurfedistanz im Programm entsprechend angepasst werden (für sehr Begabte/Motivierte)
- Optimierung der Konstruktion, z.B. Spannen eines Gummikatapults

## 2. Fahrroboter

**Beschreibung:** Der Roboter fährt einen geometrisch vordefinierten Fahrweg ab. Der Fahrweg kann auch über eine Zeichnung mit Hindernissen und verbotenen Gebieten und einem Fahrziel vorgegeben werden. Alternativ können auch Objekte als Hindernisse am Boden ausgelegt werden und ein Start und Zielpunkt wird vorgegeben. Je nach Variante ergeben sich daraus deutlich unterschiedliche Schwierigkeitsgrade. Ohne grosse Spielfeldkonstruktionen kann auch einfach eine geometrische Figur, z.B. eine Spirale oder ein Stern abgefahren werden.

### Funktionsablauf

**Variante A:** Der Roboter wird sehr genau reproduzierbar an eine Startposition gestellt. Nach dem Drücken der Starttaste fährt der Roboter den Weg zum Ziel. Das Programm kann ein einfacher, sequentieller Ablauf von Fahrbefehlen sein. Diese werden so optimiert, dass die Fahraufgabe erfüllt werden kann. Das Wissen aus dem Block 2 dieses Moduls ist ausreichend zur Bewältigung dieser Aufgabe.

**Variante B:** der Roboter benutzt zusätzlich seinen Lichtsensor, um schwarzen Linien auszuweichen.

**Variante C:** Der Roboter benutzt einen Ultraschall-Distanzsensoren, um Objekte zu erkennen und auszuweichen.

**Material:** EV3, 1 Farbsensor, 1 Distanzsensoren

**Zusatzmaterial:** Spielfeld auf weissem Papier oder Brett, Schachteln als Hindernisse

Anforderungen:

Variante	Konstruktion	Programmierung	Inbetriebnahme
A	Einfach	Einfach bis mittel	Mittel
B	Einfach	Mittel	Mittel
C	Einfach	Mittel bis schwierig	Mittel

Weiterführendes Beispiel-Programm: Fahren einer Suchspirale

Ruckartiges Fahren



Kontinuierliches Fahren



Abbildung 3: Fahren einer Suchspirale

Erklärung zum Programm: Eine Spirale kann erzeugt werden, indem man Kurven mit immer geringerem Lenkwinkel fahren kann. Dazu wird die Zählervariable 'Zaehler' verwendet. Die Variable wird auf 50 initialisiert. Dadurch dreht nur das äussere Rad. Durch Subtraktion von 1 wird der Wert der Bewegungsrichtung immer kleiner und damit der Kurvenradius immer grösser.

Der Bewegungsblock '1 Umdrehung in folgende Richtung bewegen ...' umfasst das Beschleunigen bis zur maximalen Geschwindigkeit, vor Erreichen der gewünschten Endposition ein Abbremsen mit Stillstand am Ende der Bewegung. Die Geschwindigkeit ist weggesteuert. Werden solche weggesteuerten Bewegungsblöcke aneinandergereiht oder durch eine Schleife wiederholt durchgeführt, so erfolgt eine sehr ruckartige Bewegung. In der Lösung rechts wird die Geschwindigkeit eingestellt und die Richtung kontinuierlich geändert. Die Bewegung wird über die Zeit gesteuert. Folglich werden die Motoren nicht immer wieder angehalten.

Weiterentwicklungsmöglichkeiten:

- Roboterrennen über Parcours



### 3. Zeichnungsroboter

**Beschreibung:** Am Roboter wird ein Schreiber befestigt (Filzschreiber). Anhand des Fahrwegs sollen Buchstaben oder Figuren auf ein Blatt gezeichnet werden.

**Funktionsablauf:** Der Roboter wird präzise bereitgestellt. Nach dem Drücken der Startaste fährt der Roboter die gewünschte Figur.

**Material:** EV3-Basic-Roboter

**Zusatzmaterial:** Schreiber und Klebband zur Befestigung des Schreibers

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Falls für Stift ein Greifer mit Gummibänder oder Motor gebaut wird: mittel, sonst einfach	Einfach bis mittel	Einfach bis mittel

Weiterentwicklungsmöglichkeiten:

- Der Roboter kann mit einem zusätzlichem Motor und Greifer den Stift auf und ab bewegen

### 4. Einfaches Sortieren von farbigen Objekten

**Beschreibung:** Ein Objekt wird auf eine Unterlage unter einen Farbsensor gelegt. Die Unterlage kann nach links oder rechts gekippt werden. Links und rechts werden je kleine Behälter aufgestellt, welche die Objekte sammeln.

**Funktionsablauf:** Das zu prüfende Objekt wird unter den Farbsensor gelegt. Mit einem Berührungssensor wird die Farbmessung gestartet. Je nach erkannter Farbe wird die Unterlage mit einem Motor nach links oder rechts gekippt. Die Unterlage wird danach wieder in die waagrechte Position zurückgefahren.

**Material:** EV3, 1 Motor, 1 Farbsensor, 1 Berührungssensor und verschiedene Bauteile

**Zusatzmaterial:** farbige Objekte, z.B. Smarties oder m&m oder farbige Würfel, kleine Boxen (können z.B. aus Lego konstruiert werden oder mit Papier gefaltet werden)

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Mittel	einfach	einfach

Beispiellösung:

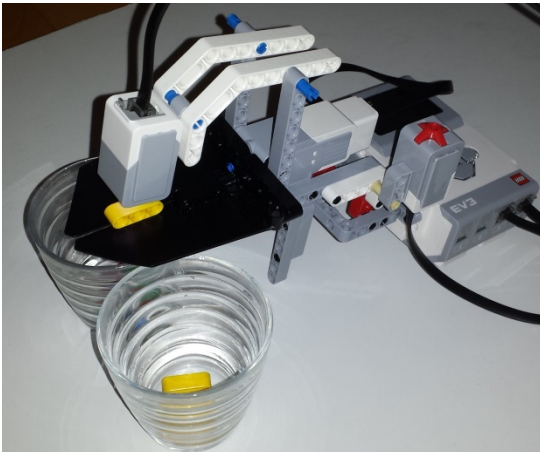


Abbildung 4: einfache Sortiermaschine

Beispiel-Programm:

Abbildung 5: Programm Sortiermaschine

Beachten Sie, dass der übliche Block für Programmstart gelöscht wurde und nur das Ereignis 'Wenn 1 gedrückt' als Start für den 'Stapel' verwendet wird. Die 'Falls <.>, dann ... sonst'-Verzweigung wurde verwendet, um die gelben Objekte auf eine Seite, die andersfarbigen auf die andere Seite zu kippen. Damit die Objekt aufgrund einer schnellen Bewegung nicht weggeschleudert werden, wird die Geschwindigkeit auf 25% festgelegt. Der Warteblock ist nötig, damit das Objekt abrutschen kann.

Weiterentwicklungsmöglichkeiten:

- Die Objekterkennung kann mit Ton angezeigt werden (Programm)
- Das Objekt kann weggespickt werden (Konstruktion und Programm)
- Das Objekt kann über eine Rutschbahn zugeführt werden (Konstruktion)
- Die Anwesenheit von Objekt wird selber erkannt (Programm)

## 5. Musiker-Roboter

**Beschreibung:** Der Roboter fährt eine Figur am Boden (z.B. zwei Kreise einer Zahl Acht) und spielt dazu ein Lied ab. Alternativ kann er auch einen Ton ausgeben, der abhängig von der Lichtstärke ist. Es gibt beliebig viele Alternativmöglichkeiten unterschiedlicher Schwierigkeit. Neu kommt hier das Element der Parallelität dazu.

**Funktionsablauf:** Der Roboter wird bereitgestellt. Nach dem Drücken der Starttaste fährt der Roboter die gewünschte Figur und spielt parallel dazu eine Klangdatei ab oder spielt ein nach Noten programmiertes Lied parallel dazu ab.

**Material:** EV3-Basic-Roboter, Lichtsensor

**Zusatzmaterial:** keines, evtl. Taschenlampe

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Keine zusätzliche	Einfach bei Abspielen Klangdatei Mittel-schwer falls durch Sensor bestimmt	einfach

**Beispiel-Programm** (weiterführende Lösung helligkeitsabhängiger Tonmeldodie):

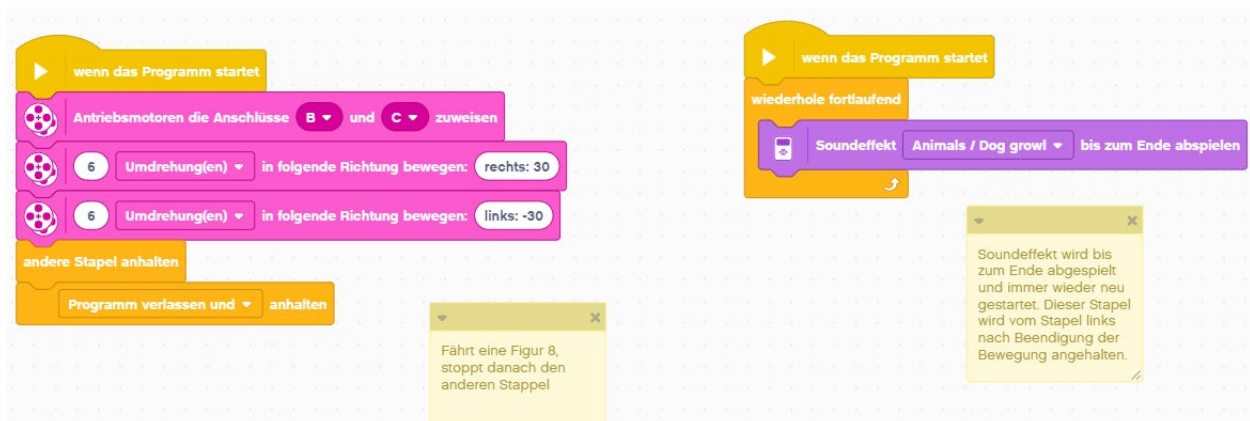


Abbildung 6: Programm mit parallelen Pfaden

Im Programm werden im Programmpfad links die beiden Kreise gesteuert. Parallel dazu wird rechts ein Geräusch erzeugt. Im Programm rechts wird das Knurren eines Hundes abgespielt.

Weiterentwicklungsmöglichkeiten:

- kann dazu noch eine Fahne schwenken (Fan-Roboter)
- unterschiedliche Töne für Rechtskurve oder Linkskurve:



Abbildung 7: Variation I Musikerroboter

Ein Piepton wird abgespielt bis ein neuer Piepton befohlen wird. Am Ende werden die Soundeffekte ausgeschaltet.



Abbildung 8: Variation I Musikerroboter parallel

Ereignisgesteuerte Lösung: Vor dem ersten Kreis wird die Meldung ‚rechts‘ an alle gesendet. Dies startet den ersten Stapel rechts oben. Der Soundeffekt wird abgespielt bis zum Ende. Sobald der erste Kreis abgefahren ist, wird die Meldung ‚links‘ gesendet und der rechts unten dargestellte Soundeffekt wird abgespielt. Ist der erste noch nicht fertig, so wird er überschrieben.

## 6. Lichtsucher

**Beschreibung:** Mittels einer Glühbirne (Licht soll in alle Richtungen ausgestrahlt werden) wird eine Licht-Boje simuliert. Der Roboter versucht, zur Lichtquelle zu gelangen.

**Funktionsablauf:** Der Roboter wird am Start bereitgestellt. Nach dem Drücken der Starttaste dreht er sich an Ort und misst mit dem Lichtsensor die Helligkeit. Hat die Helligkeit einen bestimmten Grenzwert überschritten, so ist dies die Lichtquelle, die Drehung wird gestoppt und der Roboter fährt ein Stück. Danach wird das Prozedere wiederholt. Ist die Lichtquelle genügend stark, so stoppt der Roboter. Dies kann natürlich auch über die Zeit oder über einen Weg vorgegeben werden.

**Material:** EV3-Basic-Roboter, 1 Lichtsensor

**Zusatzmaterial:** Leuchte. Die Lichtquelle ist idealerweise eine Lampe, die in alle Richtungen abstrahlt

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Keine zusätzliche	Mittel	einfach

Beispiellösung:

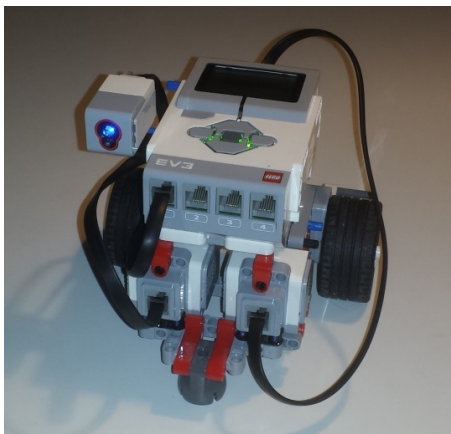


Abbildung 9: Roboter mit Lichtsensor

Beispiel-Programm:

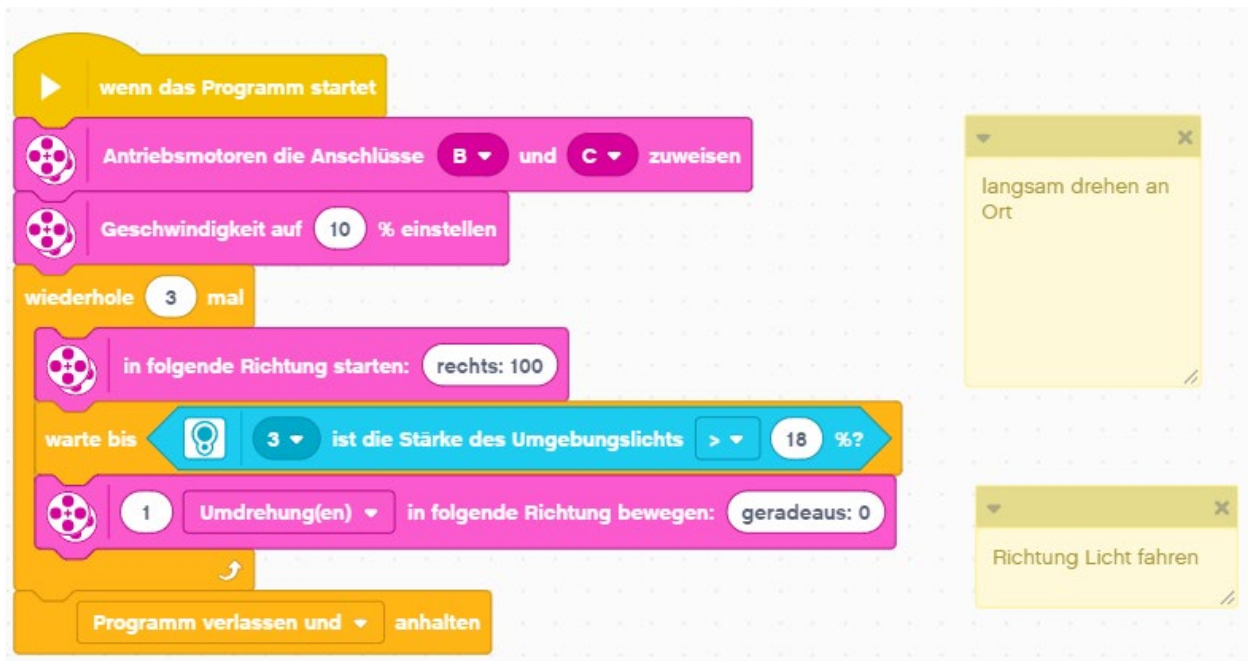


Abbildung 10: Programm Lichtsucher

Weiterentwicklungsmöglichkeiten:

- Die Ausgangslage kann schwieriger gemacht werden. Erkennt der Roboter kein Licht, so kann er in einer Suchspirale herumfahren (siehe Fahrroboter).
- Es gibt verschiedene Programmvariationen. So kann der Roboter z.B. eine 360 Grad Drehung machen und sich den Winkelwert des Gyrosensors mit der grössten Lichtintensität merken. Nach der 360 Grad Drehung führt er eine Drehung bis zum gemessenen Winkel aus und fährt in diese Richtung (schwierige Programmieraufgabe).
- Die Lichtquelle kann sich langsam bewegen und der Roboter folgt der Quelle.
- Der Lichtsensor kann auch durch den Distanzsensoren ersetzt werden. Es können dann Objekte gesucht werden.

## 7. Plotter

**Beschreibung:** mittels 2 Motoren und einem kreuzförmigen Gestänge werden Figuren auf ein Blatt gezeichnet.

**Funktionsablauf:** Der Roboter wird bereitgestellt. Nach dem Drücken der Starttaste drehen die Motoren in die gewünschte Positionen oder drehen sich mit unterschiedlichen Geschwindigkeiten. Ein Stift am Ende des Kreuzgestänges zeichnet.

**Material:** EV3, 2 Motoren

**Zusatzmaterial:** Filzschreiber, evtl. Klebeband

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Einfach	Von sehr einfach bis schwierig, je nach gewünschtem Bild	einfach

Beispiellösung:

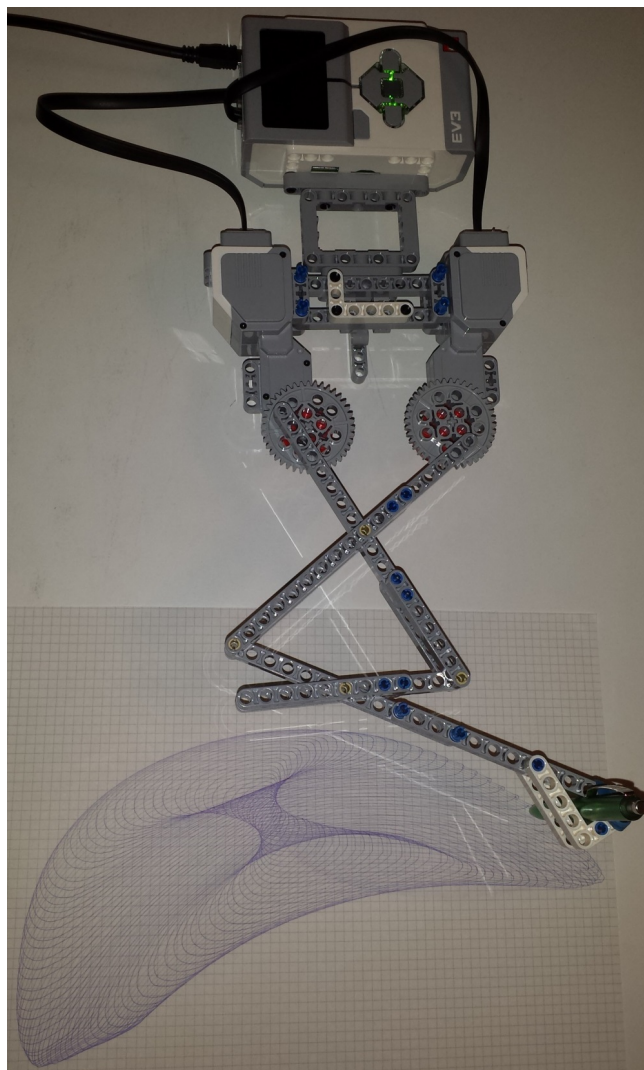


Abbildung 11: Plotter

Beispiel-Programm:



Abbildung 12: Plotterprogramm

Weiterentwicklungsmöglichkeiten:

- komplexere Programme
- Modifikation der Gestänge
- Zusätzliche Bewegung durch Erweiterung mit zusätzlichem Motor, der einen oder beide anderen Motoren bewegt.

## 8. Fernsteuerung mit Licht

**Beschreibung:** Mittels einer Taschenlampe wird ein Roboter gesteuert. Der Fahrroboter wird mit einem oder zwei Lichtsensoren (von einem anderen Kasten ausleihen) ausgerüstet. Diese sollen möglichst weit voneinander am Roboter montiert werden. Es wird ein Programm erstellt, damit der Roboter eine Linkskurve fährt, wenn der eine Lichtsensor mit der Taschenlampe beleuchtet wird, aber eine Rechtskurve macht, wenn der andere Lichtsensor beleuchtet wird.

**Funktionsablauf:** Der Roboter wird am Start bereitgestellt. Nach dem Drücken der Starttaste fährt der Roboter gerade (nicht zu schnell). Mit Hilfe der Taschenlampe soll der Roboter an ein Ziel gesteuert werden.

**Material:** EV3-Basic-Roboter, 1-2 Lichtsensoren

Zusatzmaterial: Taschenlampe

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Keine zusätzliche	einfach	einfach



Beispiellösung:

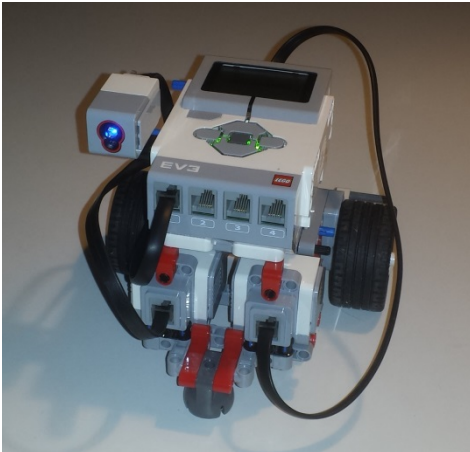


Abbildung 13: Roboter mit Lichtsensor

Beispiel-Programm:

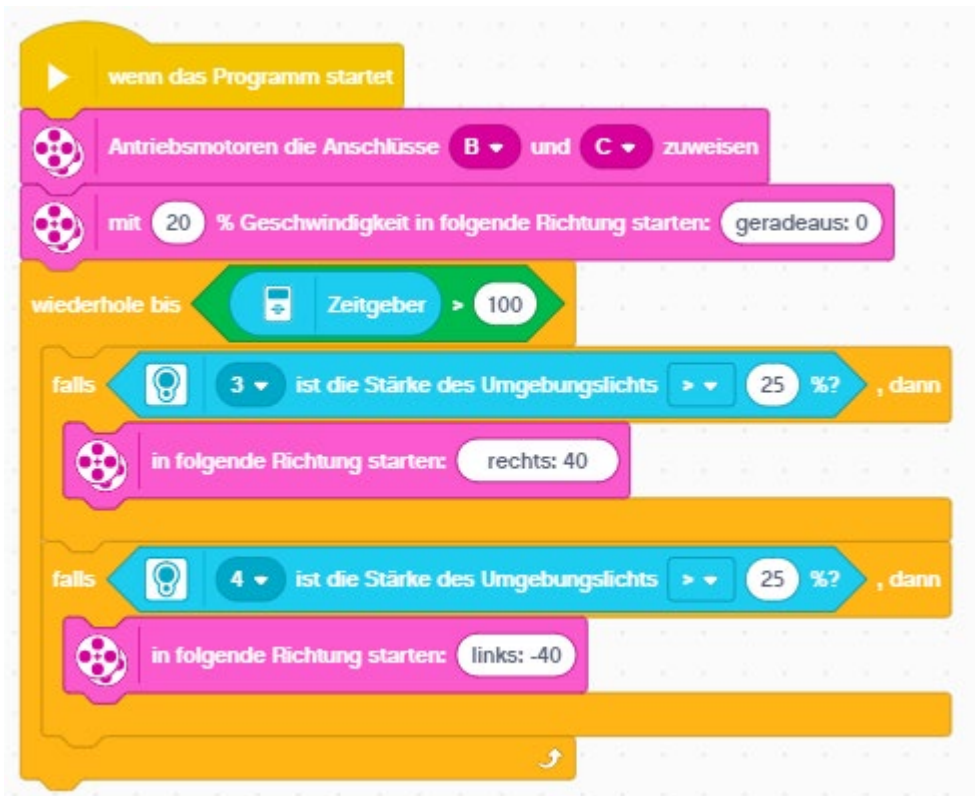


Abbildung 14: Programm Lichtsteuerung mit 2 Farbsensoren

Das Programm benutzt eine Schleife, d.h. nach z. B. 100 Sekunden ist die Fahrübung vorbei. Die Schleife wird sehr oft pro Sekunde durchgerechnet, d.h. es wird fast permanent überprüft, was an den Sensoren gemessen wird. Die beiden Schalter/Verzweigungen erlauben zuerst die Prüfung des 1. Sensors am Anschluss 3, danach des zweiten am Anschluss 4.

Weiterentwicklungsmöglichkeiten:

- Figuren durch Lichtstrahl einer Lampe auslösen.

## 9. Fahren im Kreis

**Beschreibung:** Auf einem weissen Brett wird mit schwarzem Klebeband ein Kreis oder Ähnliches aufgeklebt. Aussen kann mit einem zweiten Band ein Toleranzband aufgeklebt werden, das ca. 5 cm vom ersten entfernt ist. Es gilt nun einen Roboter mit einem Lichtsensor so zu steuern, dass nie ein Teil des Roboters über das Toleranzband hinaus ragt. Ziel ist, so lange wie möglich bei vorgegebener Geschwindigkeit innerhalb des Kreises zu bleiben.

**Funktionsablauf:** Der Roboter wird innerhalb des Kreises zum Start bereitgestellt. Nach dem Drücken der Starttaste fährt der Roboter geradeaus. Erreicht er die Linie, so wird gestoppt, etwas zurück gefahren und dann den Roboter um 180 Grad gedreht. Danach fährt er wieder vorwärts bis er wieder eine schwarze Linie trifft. Wie lange funktioniert dies? Wie kann man diese Zeit maximieren?

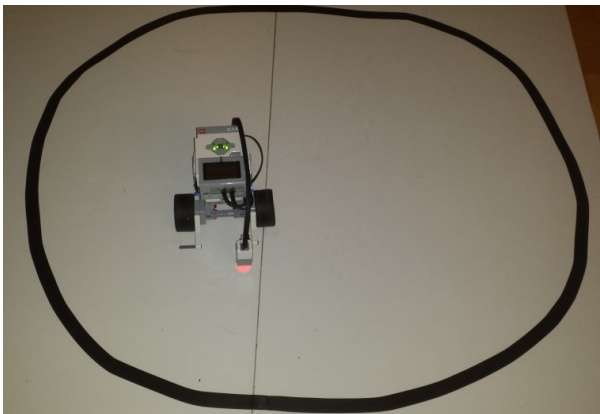
**Material:** EV3-Basic-Roboter, 1 Lichtsensor

**Zusatzmaterial:** Brett mit Klebband

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Keine zusätzliche	Einfach - mittel	einfach

Beispiellösung:



Das Brett besteht aus zwei Teilen. Dies macht den Umgang und das Transportieren viel einfacher und stört nur in wenigen Fällen.

Abbildung 15: Roboter mit Lichtsensor im Kreis

Beispiel-Programm:

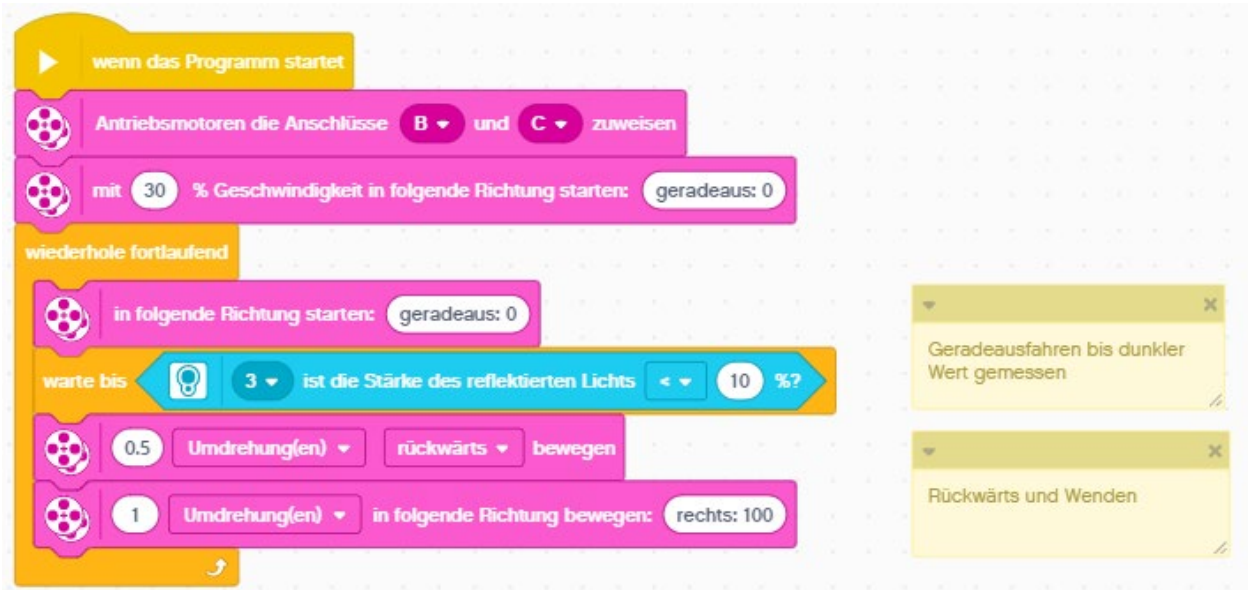


Abbildung 16: Programm 'Fahren im Kreis'

Für das Wenden des Roboters wird die Drehrichtung mit einer Zufallszahl bestimmt. Dies reduziert die Wahrscheinlichkeit, dass der Roboter aufgrund eines nicht perfekten Drehens immer näher zum Rand kommt und nur noch kurze Bewegungen weit weg vom Mittelpunkt durchführt.

Weiterentwicklungsmöglichkeiten:

- Erweiterung auf 2 Lichtsensoren. Dadurch kann Roboter nach dem Drehen senkrecht zur Tangente ausgerichtet werden und wird so immer möglichst über den Kreismittelpunkt zur nächsten Linie fahren

## 10. Linienfolger mit Hindernis

**Beschreibung:** Der Roboter folgt auf einem weissen Brett einer schwarzen Linie (Beispiel für Linienfolgeprogramm in EV3 Classroom vorhanden). Mit dem Distanzsensoren (Ultraschallsensoren) bemerkt er ein Objekt (z.B. kleine Schachtel, mittig auf der Linie, rechtwinklig zur Linie aufstellen, Grösse der Schachtel muss etwa bekannt sein). Der Roboter muss nun um die Schachtel herum fahren und danach wieder die Linie suchen und dieser folgen.

**Funktionsablauf:** Der Roboter wird am Start bereitgestellt. Nach dem Drücken der Starttaste fährt der Roboter der Linie nach. Detektiert er ein Hindernis (While-Schleife-Abbruchbedingung), so stoppt der Roboter. Danach dreht er an Ort um 90 Grad (Qualität der Linienverfolgung spielt hier eine wichtige Rolle!) fährt einen etwa halben Bogen (oder drei gerade Wegstücke), bis er wieder auf die schwarze Linie trifft. Danach dreht er um einen zu optimierenden Winkel und beginnt wieder von vorne mit dem Folgen einer Linie.

**Material:** EV3-Basic-Roboter, 1 Lichtsensor

**Zusatzmaterial:** weisses Brett mit Kreis aus schwarzem Klebband

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Keine zusätzliche	mittel	mittel

Beispiellösung:

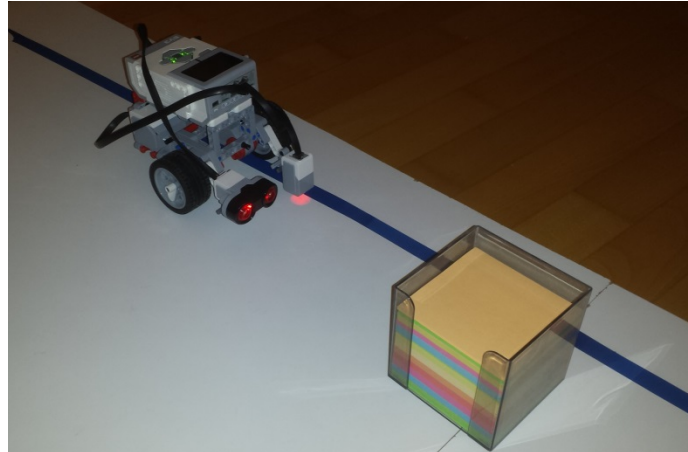
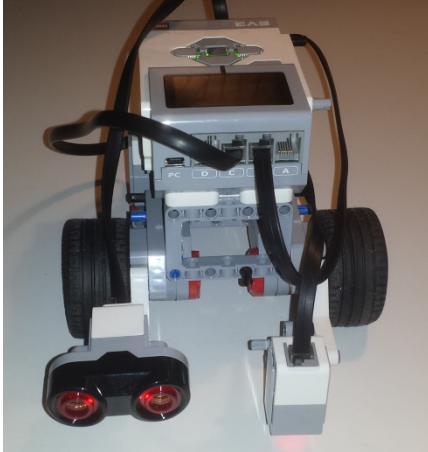


Abbildung 17: Linienfolger mit Hindernis umgehen

Beispiel-Programm:

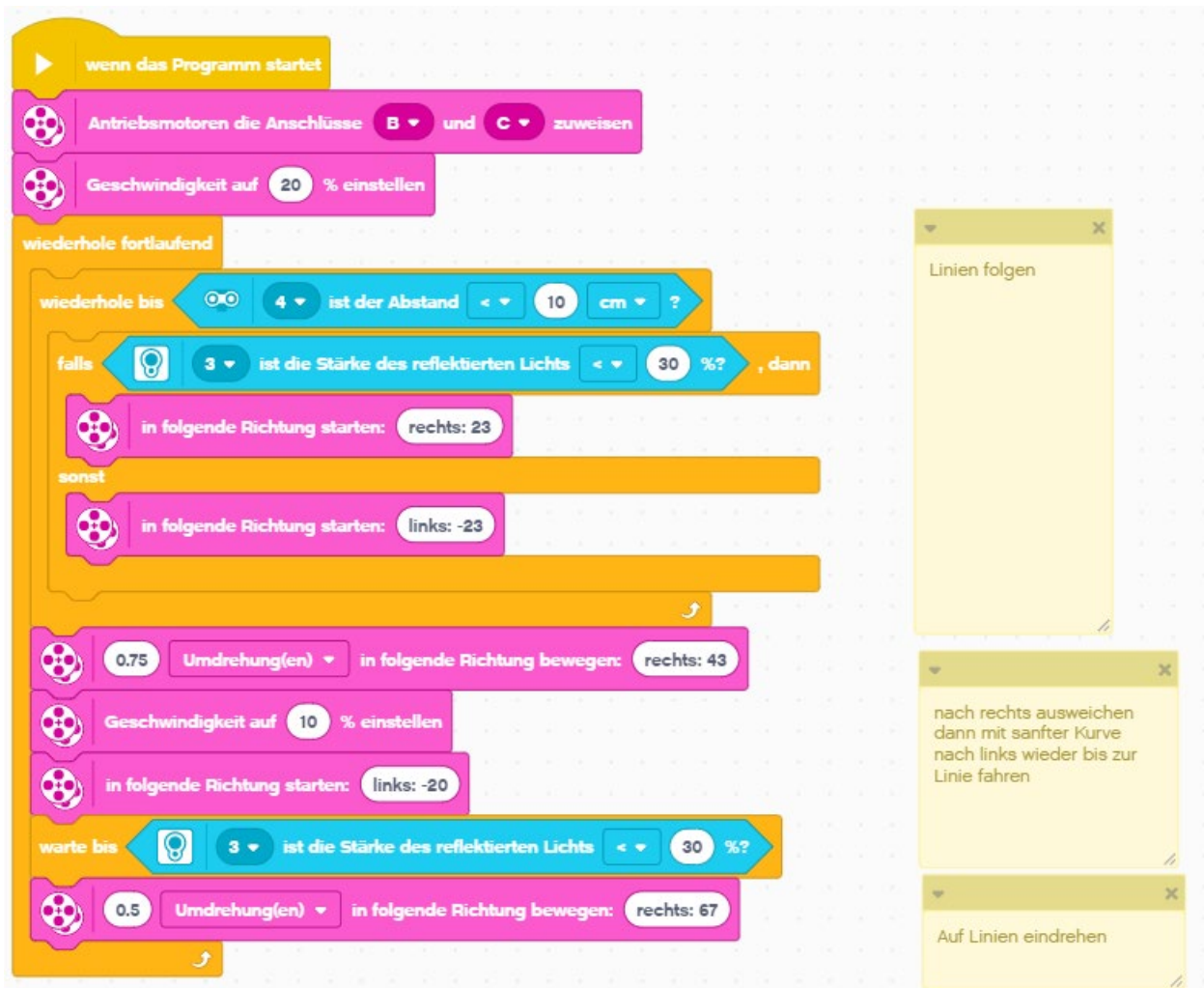


Abbildung 18: Linienfolger mit Ausweichmanöver

Weiterentwicklungsmöglichkeiten:

- Verwendung des Gyrosensors

## 11. Sortieren von farbigen Objekten

**Beschreibung:** Ein Objekt wird auf eine Unterlage unter einen Farbsensor gelegt. Die Unterlage kann gedreht werden. Links und rechts werden je kleine Behälter aufgestellt, welche die Objekte sammeln.

**Funktionsablauf:** Das zu prüfende Objekt wird auf eine Unterlage (schwarz) unter den Farbsensor gelegt. Die Farbe des Objekts wird auf dem Bildschirm angezeigt. Wird diese richtig erkannt, so wird mit einem Taster auf dem EV3 die Sortierung gestartet. Je nach erkannter Farbe wird die Unterlage mit einem Motor nach links oder rechts geschwenkt. Ist die gewünschte Position erreicht, so wird das Objekt in ein Gefäss gekippt. Die Unterlage wird danach wieder in die waagrechte Position zurückgefahren.

**Material:** EV3, 1 Motor gross, 1 Motor mittel, 1 Farbsensor, verschiedene Bauteile

**Zusatzmaterial:** farbige Objekte, z.B. Smarties oder m&m oder farbige Würfel, kleine Boxen (können z.B. aus Lego konstruiert werden, Trinkgläser sein oder mit Papier gefaltet werden).

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Mittel	schwer	mittel

Beispiellösung:



Abbildung 19: Mehrfachsortierer

Beispiel-Programm:

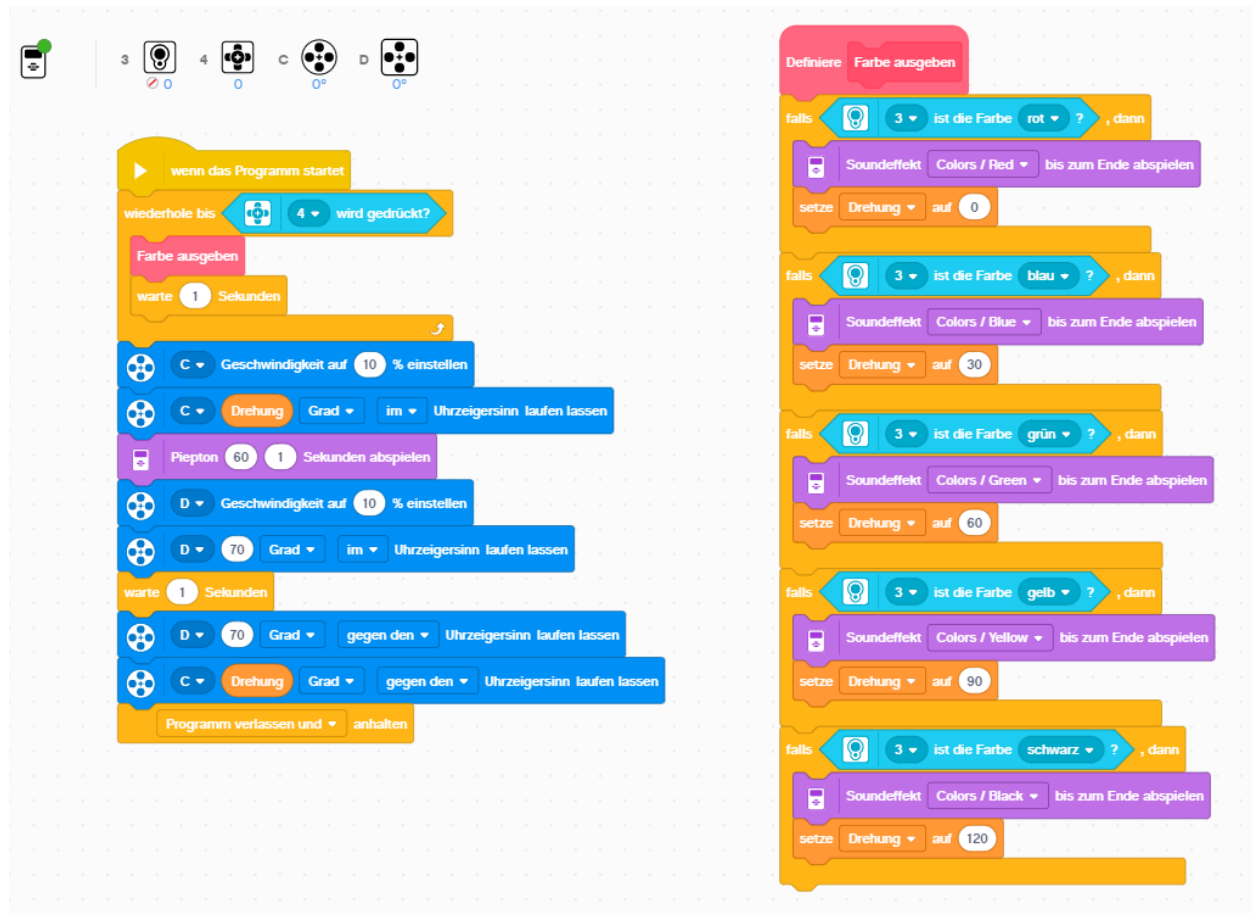


Abbildung 20: Mehrfachsorrierprogramm

Bei diesem Programm ist man mit der Aufgabe konfrontiert, dass nach dem Drehen zum gewünschten Farbbehälter und dem Ausleeren, die Ausgangslage wieder erreicht werden muss. Nur so ist ein mehrmaliges Funktionieren möglich. Nach dem Ausleeren des farbigen Objekts kann das Programm nicht mehr mit Sensoren bestimmen, wo die Ausgangslage ist. Es muss also irgendwie abgespeichert werden, welche Bewegung gemacht wurde, damit diese wieder rückgängig gemacht werden kann. Dazu gibt es verschiedene Lösungsmöglichkeiten. In diesem Programm wurde eine Lösung mit Werten in der Variablen 'Drehung' gewählt.

Die Ansage des gemessenen Farbwerts wird **mit einem eigenen Baustein gelöst. Ein solcher kann einfach selber erstellt werden.**

## Eigene Blöcke



Durch eigene Blöcke kann ein Programm viel übersichtlicher programmiert werden. Leider können eigene Blöcke nicht in andere Programme übernommen werden.

Weiterentwicklungsmöglichkeiten:

- Holen von Objekten mittels Greifer
- Objekt kann über eine Rutschbahn zugeführt werden (Konstruktion)
- Anwesenheit von Objekt wird selber erkannt (Programm)

## 12. Roboterrennen

**Beschreibung:** Auf einem weissen Brett wird mit schwarzem Klebeband ein Kreis oder Ähnliches aufgeklebt. Es gilt nun einen Roboter mit einem Lichtsensor (Farbsensor als Helligkeitsmesser gebraucht) so zu steuern, dass er im Kreis herum fährt und die Fahrbahn nie verlässt. Das Rennen gewinnt derjenige Roboter, der 3 Runden in der kleinsten Zeit gemacht hat. Damit der Roboter auch bei unterschiedlicher Beleuchtung immer gleich funktioniert, soll der Lichtsensor kalibriert werden.

**Funktionsablauf:** Auf dem Roboter wird im Display angezeigt, dass der Roboter zur Kalibrierung so hingestellt wird, dass der Lichtsensor auf das schwarze Klebband leuchtet. Mit der programmierten Taste wird nun der zweite Kalibrierungsschritt gestartet. Auf dem Display wird angezeigt, dass der Roboter nun so hingestellt wird, dass der Lichtsensor auf die weisse Fläche leuchtet. Ist dies erfolgt, so wird dies wiederum mit der programmierten Taste auf dem EV3 bestätigt. Ist die Kalibrierung abgeschlossen, so wird auf dem EV3 angezeigt, dass der Roboter zum Start bereit ist. Dieser wird nun so aufgestellt, dass er in der Nähe der schwarzen Linie ist und auf diese zufährt. Mit der programmierten Starttaste startet die Fahrt.

**Zusatzmaterial:** Brett mit Klebband

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Keine zusätzliche	mittel	Mittel



Beispiellösung:



Abbildung: EV3 mit Licht- und Distanzsensor

Beispielprogramm:

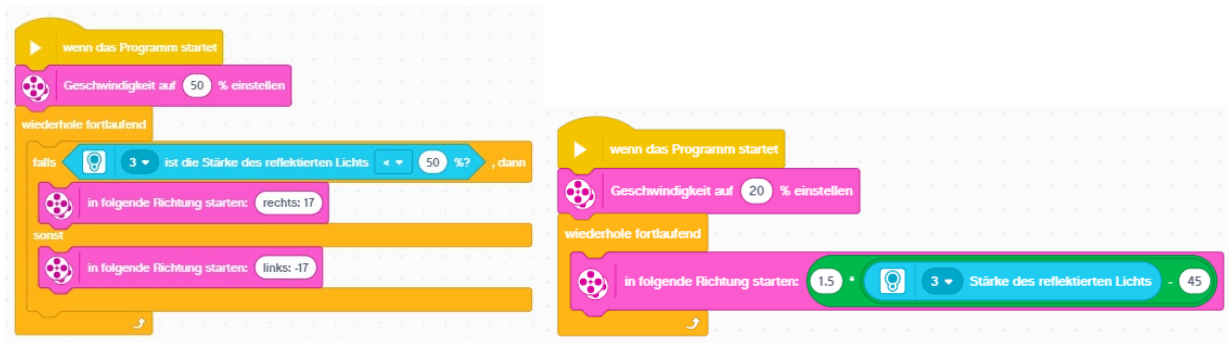


Abbildung: Linienfolger geschaltet (links) und geregelt (rechts)

Wichtig im Programm links ist der Grenzwert für das Schalten (hier nun 50) und die gewählte Fahrrichtung. Der Grenzwert wird durch Messen der Helligkeit so bestimmt, dass der Sensor auf einer Kante des schwarzen Bandes steht und dieses zur Hälfte mit dem roten Lichtkegel ausleuchtet. Je nach Parcours müssen andere Richtungen und Fahrgeschwindigkeiten vorgegeben werden, damit der Roboter der Linie folgen kann. Dies kann Inhalt eines kleinen Wettbewerbs sein. Rechts ist eine Lösung mit einer Regelung dargestellt. Abhängig vom Messwert des Sensors wird der Lenkwinkel berechnet:

$$\text{Lenkrichtung} = 1.5 * (\text{Sensorwert} - 45)$$

Ist der Sensorwert 45, so ist die Lenkrichtung gleich 0. Misst der Sensor 35, so ist nach der vorangehenden Formel die Lenkrichtung gleich -15. Je niedriger der Sensorwert desto stärker wird von der Linie weggesteuert, aber nur wenn der Roboter auf der richtigen Seite der Linie steht.

Weiterentwicklungsmöglichkeiten:

- Roboterkolonne: Mehrere Roboter sind gleichzeitig in der Rennbahn. Jeder ist aber mit einem zusätzlichen Distanzsensor ausgerüstet und versucht so, nicht in einen anderen hinein zu fahren.

### 13. Suchroboter

**Beschreibung:** Auf einer Fläche wird ein Objekt (z.B. Trinkbecher) ausgelegt. Der Roboter soll dieses finden.

**Funktionsablauf:** Der Roboter wird in der Nähe des Suchobjekts aufgestellt. Das Objekt soll mit dem Distanzsensor (Ultraschallsensor) erkennbar sein. Der Roboter dreht nun langsam um die eigene Achse und misst fortlaufend den Abstand. Dabei speichert er den Wert immer ab, wenn er kleiner als der vorangehend gemessene Wert war. Mit dem Drehsensor/Gyrosensor misst er den Drehwinkel. Nach einer ganzen Drehung um die eigene Achse kennt er die kürzeste, gemessene Distanz. Nun muss er sich wieder um die eigene Achse drehen, bis er einen Distanzwert misst, der kleiner oder nur wenig grösser als der vorher gemessene kleinste Wert ist. Hat er diese Ausrichtung gefunden, stoppt er die Drehung und fährt die gemessene Distanz nach vorne. Ist er näher als 5 cm am Objekt, soll ein Freudensound abgegeben werden.

**Zusatzmaterial:** Objekt, z.B. Trinkbecher, Krug

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Bau des Scanners: mittel	schwer	mittel

Beispiel-Programm:

Abbildung 21: Suchprogramm mit den beiden Unterprogrammen

Weiterentwicklungsmöglichkeiten:

- Suchgebiet mit schwarzer Linie (für Lichtsensor) oder mit Bande (Berührungssensor) begrenzen
- Objekte mittels Pflug wegstossen

Hinweise: Der Ultraschallsensor wird öfters durch rauen Boden gestört. Der Gyrosensor kann ‚davonlaufen‘. Abhilfe verschafft eine Initialisierung bei absolut ruhendem Roboter.

## 14. Rohr- oder Badewannenroboter

**Beschreibung:** Der Roboter soll mit Hilfe des Gyrosensors durch ein Rohr oder in der Badewanne fahren können.

**Funktionsablauf:** Der Roboter wird im Rohr gestartet. Fährt der Roboter gegen eine Seite, so soll dies durch den Gyrosensor erfasst werden und der Roboter wieder in die Rohrmitte gelenkt werden.

**Zusatzmaterial:** Rohr, ca 40cm Durchmesser oder Badewanne oder andere Begrenzungen, die den Roboter neigen.

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
einfach	Einfach	einfach

Beispiellösung:

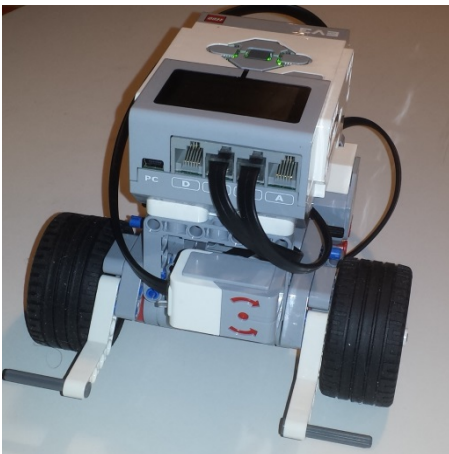


Abbildung 22: Roboter mit Gyrosensor um Längsachse

Beispiel-Programm:

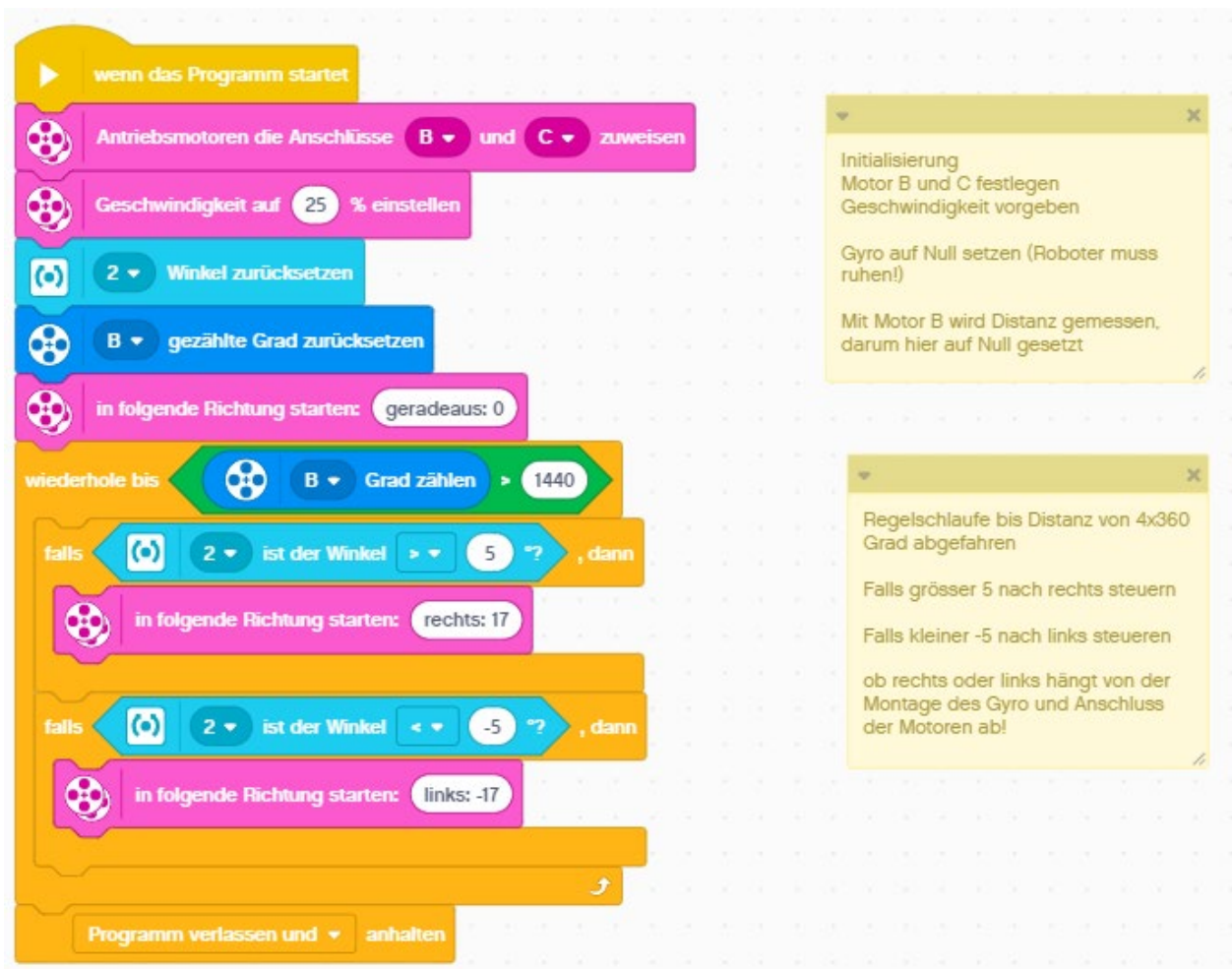


Abbildung 23: Gyrosensor-geregelter Fahrroboter mit Schaltregler

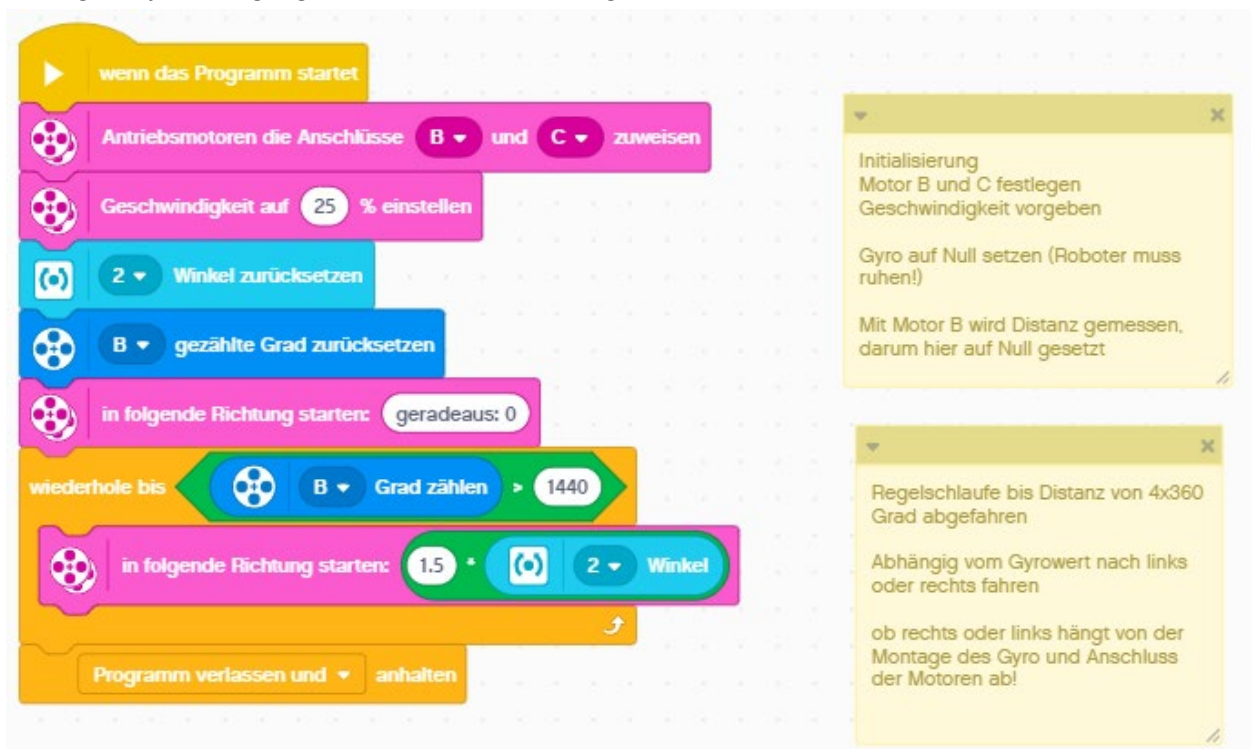


Abbildung 24: Gyrosensor-geregelter Fahrroboter mit kontinuierlichem Regler

Weiterentwicklungsmöglichkeiten:

- Erweiterung der Funktionalität durch Wenden im Rohr

## 15. Schnitzeljagd

**Beschreibung:** Der Roboter soll auf einem weissen Brett fahren. Trifft er auf einen blauen Farbfleck, soll er eine 90 Grad-Kurve nach rechts machen. Trifft er auf einen grünen Farbfleck, soll er eine 90 Grad-Kurve nach links machen. Trifft er auf Rot, so soll er stoppen.

**Funktionsablauf:** Der Roboter wird an die Startposition gestellt und soll den roten Farbfleck erreichen, auch wenn die Farbflecken umgestellt werden.

**Zusatzmaterial:** Brett mit ca. 10x10cm grossen Farbstücken (2 grün, 2 blau, 1 rot, Farben können auch variiert werden, es soll aber vorher geprüft werden, ob sie durch den Farbsensor richtig erkannt werden, Cremefarben nicht geeignet).

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Standardroboter Farbsensor	mittel	Mittel-schwer

Beispiellösung:

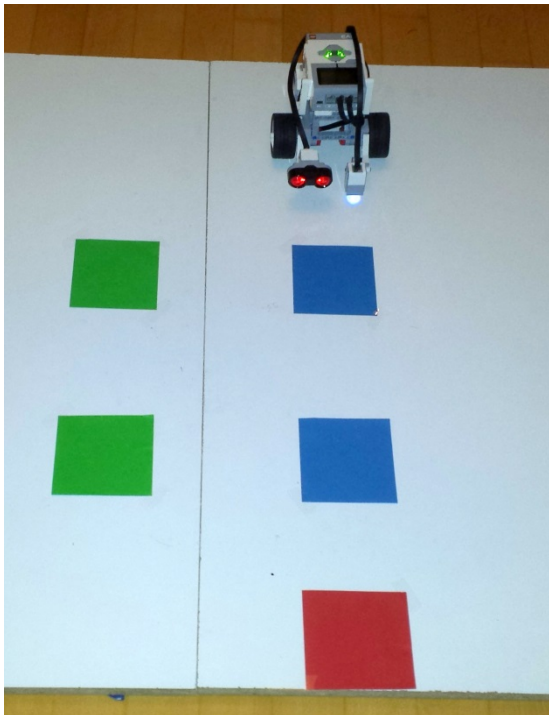


Abbildung 25: Beispiel Farblabyrinth

Beispiel-Programm:



Abbildung 26: Programm Schnitzeljagd

Das Programm umfasst eine Schleife, die bei Farbe = rot abgebrochen wird. Wird weder blau noch grün erkannt, so fährt der Roboter gerade. Wird blau erkannt, so fährt er eine Kurve nach rechts, bei grün eine nach links. Aufgrund unterschiedlicher Reibung und der Realisierung der Kurvenfahrt sind für Linkskurven nicht die gleichen Einstellungen wie für Rechtskurven nötig. Hier ist bei der Inbetriebnahme einiges an Probieren nötig.

Weiterentwicklungsmöglichkeiten:

- Unterschiedliche Abzweigwinkel

## 16. Tanzroboter

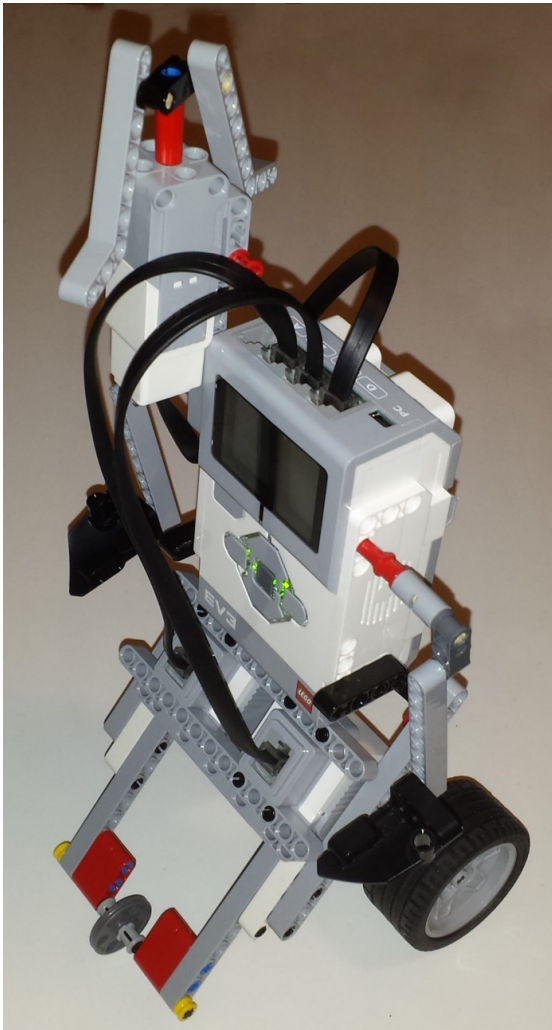


Abbildung 27: Tanzroboter

**Beschreibung:** Die Bewegungen eines Roboters sollen mit Tönen synchron erfolgen. Es soll ein Roboter gebaut werden, der entweder durch Fahren oder Bewegung von Armen, Musik und Bewegung zeitgleich und möglichst effektiv kombiniert.

**Zusatzmaterial:** ev. Roboterverzierungen

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Bau eines Tanzroboters: einfach bis zeitaufwändig	einfach-schwer	einfach

Beispiel-Programm:

Das Programm für einen Tanzroboter kann von unterschiedlicher Komplexität sein. Die Lösung kann sehr einfach sein, aber auch eine fast beliebige Komplexität aufweisen. Motivierte SuS werden mit einer einfachen Lösung beginnen und diese immer weiter ausbauen, bis das Programm sehr gross wird. Spätestens dann stellt sich die Frage nach einer besseren Strukturierung des Programms. **Ein wichtiges Element der Strukturierung ist die Programmierung von Stapeln oder von eigenen Blöcken.** Beide Strukturen können im 'Hauptprogramm' mehrfach aufgerufen werden und müssen darum nicht für jede Anwendung neu programmiert werden. Der Unterschied zwischen einzelnen Blöcken und dem Aufrufen von



Stapeln über Ereignisse liegt darin, dass Blöcke im Ablauf fertig sein müssen, bevor der nächste Block ausgeführt wird. Bei Ereignis-gesteuerten Stapel laufen diese parallel zu andere Programmen. Dies wird im Beispiel für den Walzer-Tanz-Roboter deutlich. Da sich die Bewegungen immer wiederholen, werden diese in eigenen Stapeln realisiert. Natürlich muss der Stapel für die Vorwärtsbewegung fertig sein, bevor der Stapel für die Rückwärtsbewegung gestartet werden sollte. Dies kann durch Geschwindigkeitsvorgaben oder Winkelvorgaben eingestellt werden.

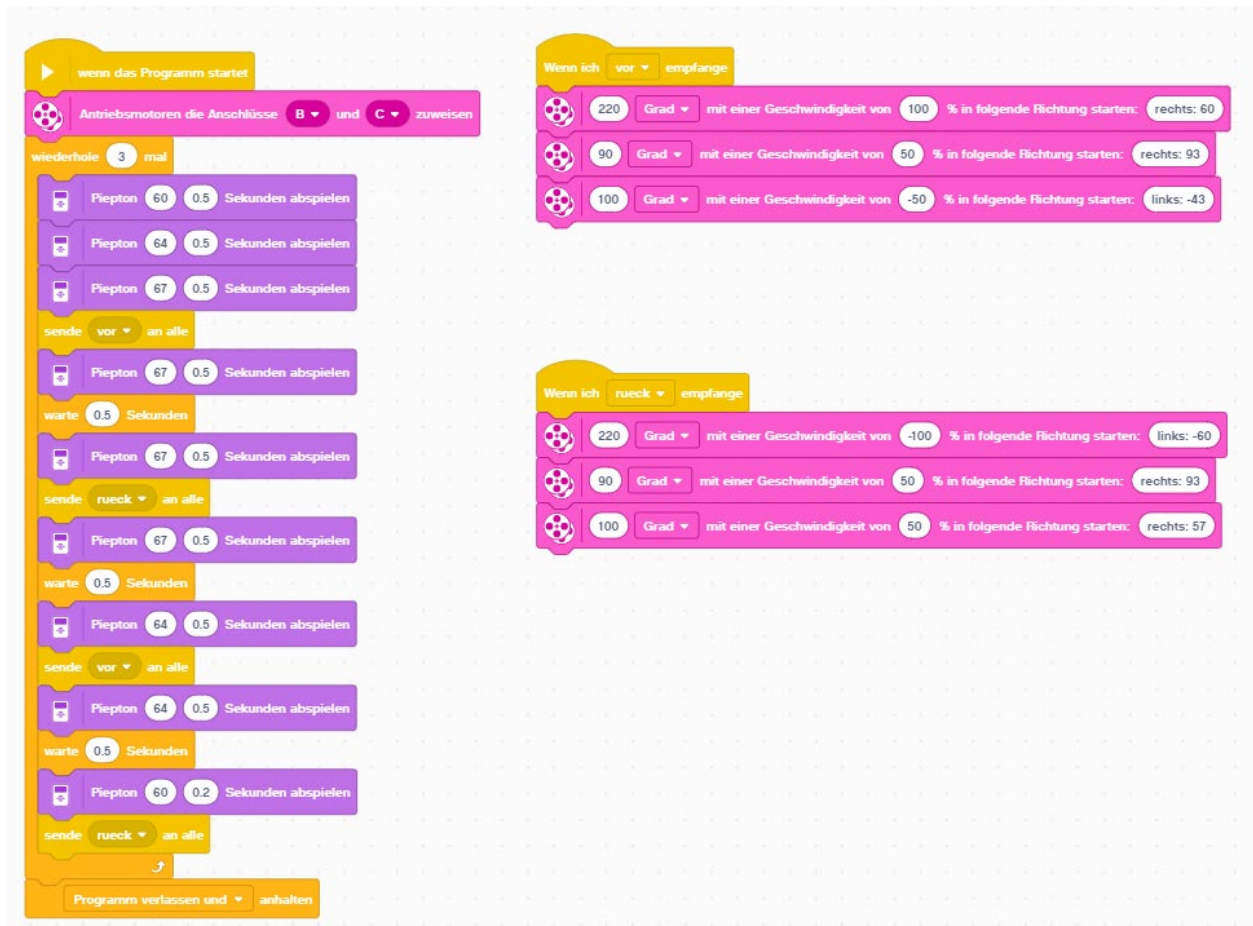


Abbildung 28: Walzertanz

In einer Schleife, die 3-mal wiederholt wird, bewegt sich der Roboter vorwärts und danach macht er eine Rückwärtsdrehung.

Damit die Musik nicht zu eintönig wird, kann die Tonart über eine Variable festgelegt werden und so die Tonfolge bei jeder Wiederholung moduliert werden.

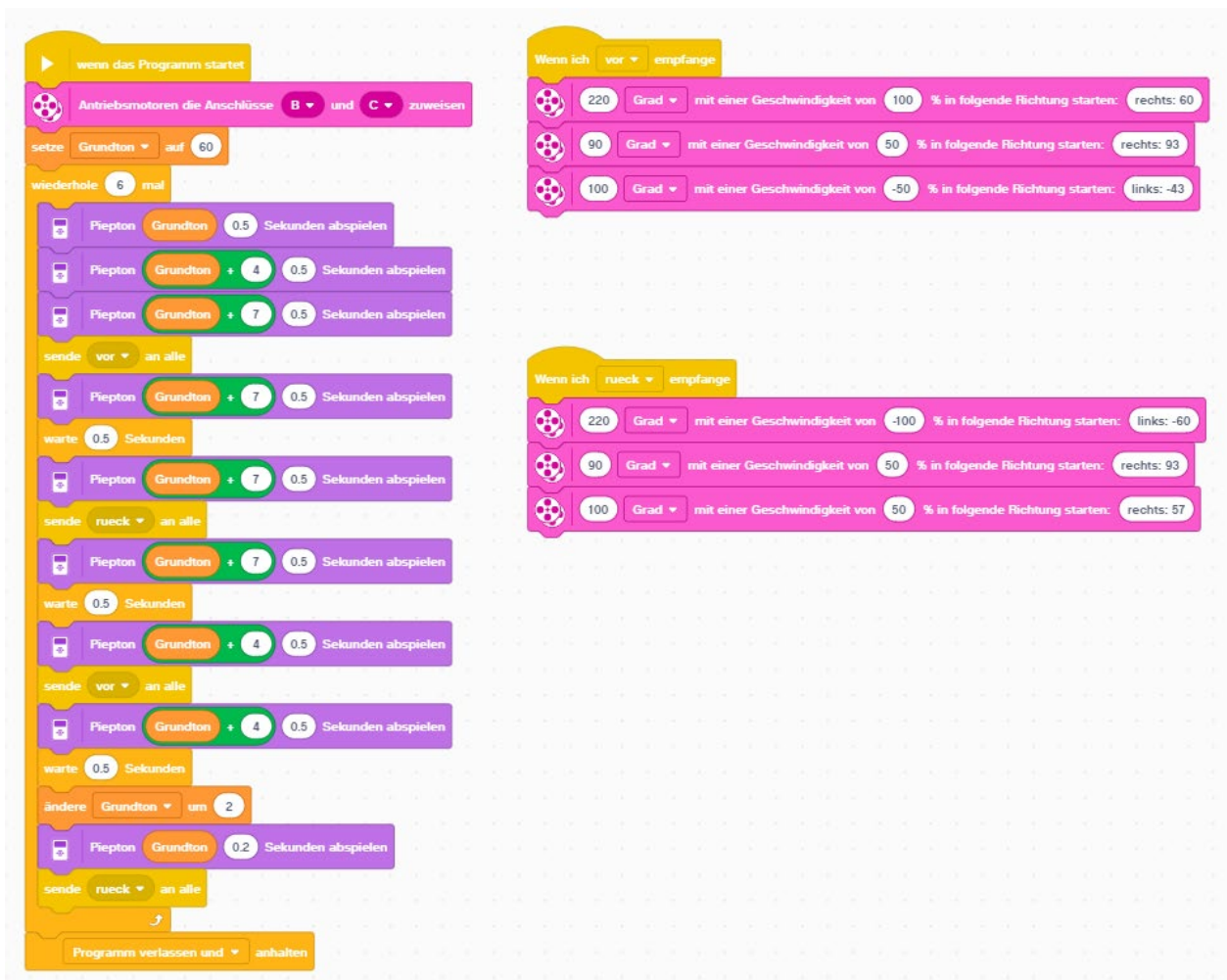


Abbildung 29: Walzertanz moduliert

## 17. Roboterkampf

**Beschreibung:** Zwei Roboter sind innerhalb eines kreisförmigen Rings auf einem weissen Brett. Ziel des Roboterwettkampfs ist es, innerhalb des Kreises solange wie möglich in Bewegung zu bleiben. Verloren hat, wer den Kreis verlassen hat oder wenn der Roboter sich nicht mehr fortbewegen kann.

**Funktionsablauf:** Die Roboter werden innerhalb des Kreises zum Start bereitgestellt. Die Startpositionen sind vorher bekannt. Nach dem Drücken der Starttaste sollen die Roboter losfahren. Erreichen sie die Linie, so wird gestoppt, etwas zurück gefahren und dann den Roboter um 180 Grad gedreht. Danach fahren sie wieder vorwärts bis sie wieder eine schwarze Linie treffen. Mit welcher Strategie bleibt man am längsten im Rennen?

**Zusatzmaterial:** weisses Brett mit schwarzem Klebband

Anforderungen:

Konstruktion	Programmierung	Inbetriebnahme
Einfach-schwierig	Einfach-schwierig	einfach

**Beispiel-Programm:** siehe 'Fahren im Kreis'

Hinweis: Es ist lohnenswert, den Roboterkampf in mehreren Runden zu machen. Zwischen den Runden sollen die SuS Zeit haben, ihre Konstruktion oder das Programm zu verbessern. Erst nach der ersten Runden sieht man kreative Lösungen.

## Inhalt

Block 3: Aufgabenideen.....	1
Grundausrüstung .....	2
Mögliche Lernziele .....	2
Didaktische Hinweise.....	2
Aufgaben.....	6
1. Ballwurfgerät.....	6
2. Fahrroboter .....	7
3. Zeichnungsroboter .....	9
4. Einfaches Sortieren von farbigen Objekten .....	9
5. Musiker-Roboter .....	11
6. Lichtsucher .....	13
7. Plotter .....	15
8. Fernsteuerung mit Licht .....	16
9. Fahren im Kreis .....	18
10. Linienfolger mit Hindernis.....	19
11. Sortieren von farbigen Objekten.....	21
12. Roboterrennen.....	24
13. Suchroboter .....	26
14. Rohr- oder Badewannenroboter.....	28
15. Schnitzeljagd .....	30
16. Tanzroboter .....	32
17. Roboterkampf .....	34
Inhalt .....	36
<i>Abbildungsverzeichnis .....</i>	<i>37</i>

## Abbildungsverzeichnis

Die Bilder und Screenshot der Programme stammen von J. Keller

Abbildung 1: einfacher Ballwurfroboter .....	6
Abbildung 2: Programm Ballwurfmaschine .....	7
Abbildung 3: Fahren einer Suchspirale.....	8
Abbildung 4: einfache Sortiermaschine .....	10
Abbildung 5: Programm Sortiermaschine .....	10
Abbildung 6: Programm mit parallelen Pfaden.....	11
Abbildung 7: Variation I Musikerroboter .....	12
Abbildung 8: Variation I Musikerroboter parallel.....	12
Abbildung 9: Roboter mit Lichtsensor .....	13
Abbildung 10: Programm Lichtsucher .....	14
Abbildung 11: Plotter .....	15
Abbildung 12: Plotterprogramm .....	16
Abbildung 13: Roboter mit Lichtsensor .....	17
Abbildung 14: Programm Lichtsteuerung mit 2 Farbsensoren .....	17
Abbildung 15: Roboter mit Lichtsensor im Kreis .....	18
Abbildung 16: Programm 'Fahren im Kreis'.....	19
Abbildung 17: Linienfolger mit Hindernis umgehen.....	20
Abbildung 18: Linienfolger mit Ausweichmanöver.....	21
Abbildung 19: Mehrfachsortierer.....	22
Abbildung 20: Mehrfachsortierprogramm.....	23
Abbildung 21: Suchprogramm mit den beiden Unterprogrammen .....	27
Abbildung 22: Roboter mit Gyrosensor um Längsachse .....	28
Abbildung 23: Gyrosensor-geregelter Fahrroboter mit Schaltregler .....	29
Abbildung 24: Gyrosensor-geregelter Fahrroboter mit kontinuierlichem Regler .....	29
Abbildung 25: Beispiel Farblabyrinth .....	30
Abbildung 26: Programm Schnitzeljagd.....	31
Abbildung 27: Tanzroboter .....	32
Abbildung 28: Walzertanz.....	33
Abbildung 29: Walzertanz moduliert .....	34