



**Hochschule Merseburg**

Fachbereich Ingenieur und Naturwissenschaften

Studiengang Mechatronik, Industrie- u. Physiktechnik

---

# Bachelorarbeit

im Zeitraum vom 04.02.2016 bis 14.04.2016

betreut durch

Prof. Dr.-Ing. Heike. Mrech (Erstgutachter)

Dipl.-Phys. Egon. Hanauska (Zweitgutachter)

## **Thema:**

**Entwicklung eines multimedialen Tutorials zum  
Bau und zur Programmierung eines modularen  
Mini-Roboters für verschiedene Anwendungsfälle**

Vorgelegt von

Tao Zou

Platz der Bausoldaten 2

Zimmer 207

06217 Merseburg

[zoutao321@hotmail.com](mailto:zoutao321@hotmail.com)

## Inhaltverzeichnis

1. Einleitung.....	1
1.1 Aufgabenstellung .....	1
1.2 Motivation der Arbeit .....	2
2. Grundlagen.....	4
2.1 Definition der Zielgruppe.....	4
2.1.1 Informationsrecherche über den Schulstoff für die Zielgruppe .4	
2.1.2 Die Anforderungen der Zielgruppe.....	5
2.2 LEGO MINDSTORMS Education EV3-Grundlagen .....	6
2.2.1 Das Grundmodul.....	6
2.2.2 Sensorik.....	12
2.2.3 EV3 Programmierungstechnik .....	16
2.2.4 EV3 Konstruktionen Technik .....	17
2.3 Übersicht bestehenden EV3-Lernmaterialien.....	19
3. Konzept für das Tutorial.....	22
3.1 Bau des Roboters .....	22
3.1.1 Wichtige Bauprinzipen .....	22
3.1.2 Problemstellungen und Lösungsansätze .....	26
3.2 Programmierung des Roboters .....	30
3.2.1 Allgemeiner Aufbau von Blöcken .....	30
3.2.2 Programmierung verschiedener Sensoren .....	37
3.2.3 Problemstellungen und Lösungsansätzen .....	42
4. Anwendungsfälle .....	46
4.1 Positionierung von Huhn und Methangas .....	46
4.1.1 Aufgabenbeschreibung.....	46
4.1.2 Erarbeitung einer Lösungsvariante .....	47
4.1.3 Beispiele für Lösungsvarianten.....	47
4.2 Positionierung des Methangases und Transport.....	48
4.2.1 Aufgabenbeschreibung.....	48
4.2.2 Erarbeitung einer Lösungsvariante .....	50

4.2.3 Beispiele für Lösungsvarianten .....	50
4.3 Kraftübertragung und Sortierung .....	51
4.3.1 Aufgabenbeschreibung .....	51
4.3.2 Erarbeitung einer Lösungsvariante .....	52
4.3.3 Beispiele für Lösungsvarianten .....	52
5. Zusammenfassung .....	53
6. Danksagung .....	54
7. Selbständigkeitserklärung .....	55
8. Literaturverzeichnis .....	56
9. Abbildungsverzeichnis .....	57
10. Anlagenverzeichnis .....	59

# 1. Einleitung

## 1.1 Aufgabenstellung

Der Einsatz modularer Robotersysteme findet in der Praxis immer mehr Anwendungsfelder. Für ein modulares Mini-Robotersystem ist ein multimediales Tutorial für drei unterschiedliche Anwendungsfälle zu entwickeln. Dabei sind mindestens drei verschiedene Sensortypen einzusetzen und verschiedene mechanische Problemstellungen zu lösen. Das Tutorial ist für eine spezifische Zielgruppe auszurichten und multimedial umzusetzen.

Im Rahmen der vorliegenden Arbeit ergeben sich dementsprechend folgende Schwerpunkte:

1. Die Recherche über vorhandene Dokumentationen und Tutorials bei LEGO MINDSTORMS Education EV3
2. Die Darstellung des aktuellen Technikstandes sowie die Empfehlung der besten Quellen
3. Die Definition bzw. die Bestimmung der Zielgruppe und deren Anforderungen
4. Die Wahl der drei unterschiedlichen Anwendungsfälle
5. Die Erarbeitung der Lösungen
6. Die Verfassung der Dokumentation mit Lösungen in einem zielgruppenspezifischen, multimedialen Tutorial

## 1.2 Motivation der Arbeit

Ziel der vorliegenden Arbeit ist es, ein multimediales Tutorial zu entwickeln, das Schülerinnen und Schüler für die sogenannten MINT-Fächer (Mathematik, Informatik, Naturwissenschaft, Technik) begeistert. Im Bereich der MINT-Fächer gibt es heute eine Vielzahl interessanter Berufsfelder. Schülerinnen und Schüler haben jedoch zu wenige Möglichkeiten ihre Talente in diesem Bereich zu entdecken und wählen daher viel zu selten diesen Berufsweg.

LEGO MINDSTORMS Education EV3 hilft Lehrern und Lehrerinnen, Themen aus Mathematik, Informatik, Technik, Robotik und Naturwissenschaften erfolgreicher zu vermitteln. Die Kombination aus Hardware und Software zielt auf einen handlungsorientierten Unterricht, in dem Schüler selbst Modelle konstruieren, programmieren und Inhalte praxisorientiert erarbeiten. [1]

Diese Plattform bietet auch großes Potential für die Gestaltung motivierender Arbeitsgemeinschaften und Wettbewerbe. Damit können Schülerinnen und Schüler frühzeitig für MINT-Themen motiviert werden und das Verständnis und Selbstbewusstsein erhöht werden.

Ein modularer Mini-Roboter enthält viele unterschiedliche Bauelemente, die bei geeigneter Form und Funktion zusammengefügt werden. Deshalb gibt es unbegrenzte Möglichkeiten, wie man gewünschte Roboter bauen kann. Dadurch können alle Schüler und Schülerinnen mehrere Baukombinationen testen, um ihre praktischen Fähigkeiten zu verbessern.

Erfahrungen der Hochschule Merseburg zeigen, dass vorhandene Dokumentationen und Tutorials über modularen Mini-Roboter selten Informationen zu detaillierten Lösungen, zum Bau und zur Programmierung eines modularen Mini-Roboters erklären. Deswegen ist es Aufgabe dieser Bachelorarbeit ein multimediales Tutorial für drei unterschiedlicher Anwendungsfälle zu entwickeln. Dabei sind vier verschiedene Sensortypen einzusetzen und verschiedene mechanische Problemstellungen zu lösen. Das Tutorial soll Schülerinnen und Schüler sowohl zur Arbeit mit LEGO MINDSTORMS Education EV3 motivieren als auch technische Kenntnisse vermitteln, um Schülerinnen und Schüler für die Teilnahme zu Roboter-Wettbewerben vorzubereiten. Dabei sind Schritt für

Schritt Problemstellungen, die Analyse von Aufgaben, die Arbeitsweise beim Bau und der Programmierung sowie die physikalischen Grundlagen zu erläutern. Die Schülerinnen und Schüler sollen mit Hilfe des Tutorials spielerisch ihre Kompetenzen erweitern und ihr Selbstbewusstsein für die Teilnahme an Wettbewerben stärken.

## 2. Grundlagen

### 2.1 Definition der Zielgruppe

#### 2.1.1 Informationsrecherche über den Schulstoff für die Zielgruppe

Das Tutorial soll Schülerinnen und Schüler im Alter von 10 bis 16 Jahren ansprechen. Dabei sind die Grundkenntnisse aus dem Schulunterricht zu berücksichtigen.

Aufgrund des Fachlehrplans in den einzelnen Schuljahren gibt es folgende Übersicht über die Kompetenzschwerpunkte.

Alter der Zielgruppe	Schuljahrgänge (Klasse)	Kompetenzschwerpunkte
10/11	5/6	Den Computer als Werkzeug nutzen
		Mehrteilige Gebrauchsgegenstände aus Holz und Kunststoff fertigen
		Funktionsmodelle entwickeln, bauen und untersuchen
12/13	7/8	Lösungen für technische Probleme untersuchen, vergleichen und bewerten
		Funktion und Einsatz von Maschinen analysieren
14/15	9/10	Technische System beschreiben und analysieren
		Technische Prozesse steuern

[2]

Aus der Übersicht ist ersichtlich, dass die Schüler und Schülerinnen im Alter von 10 bis 15 Jahren schon mit technischen Themen vertraut sind. Deshalb sollte das multimediale Tutorial für die Zielgruppe verständlich sein.

## 2.1.2 Die Anforderungen der Zielgruppe

Um die besten Darstellungsformen und Anwendungsbeispiele dieses multimedialen Tutorials zu entwickeln, wurde eine Umfrage unter den Schülern und Schülerinnen aus der Sekundarschule Bad Dürrenberg und Mitgliedern einer Arbeitsgemeinschaft der Hochschule Merseburg durchgeführt. Dadurch wurden folgende Problemstellungen und Anforderungen gesammelt.

### **Problemstellungen in Kürze:**

- Funktionsweise und Programmiermethode des Kreiselensors
- Das Durchdrehen der Zahnräder
- Lenksystem des Roboters
- Zeitsparender Umbau des Roboters
- Kalibrierung des Farbsensors
- Geradeausfahrt des Roboters
- Lösungsvarianten der Linie-Verfolgung

### **Anforderungen des multimedialen Tutorials**

- Videolänge soll 5 Minuten nicht überschreiten
- Entspannende Hintergrundmusik ist erforderlich
- Klare Video Erläuterungen



## 2.2 LEGO MINDSTORMS Education EV3-Grundlagen

### 2.2.1 Das Grundmodul

In der folgenden Abbildung sind die elektronischen Komponenten von LEGO EV3-Set dargestellt. Jedes einzelne Element hat seine eigene Funktion. Der Roboter kann mit Hilfe dieser Funktionen verschiedene praktische Aufgaben erfüllen.

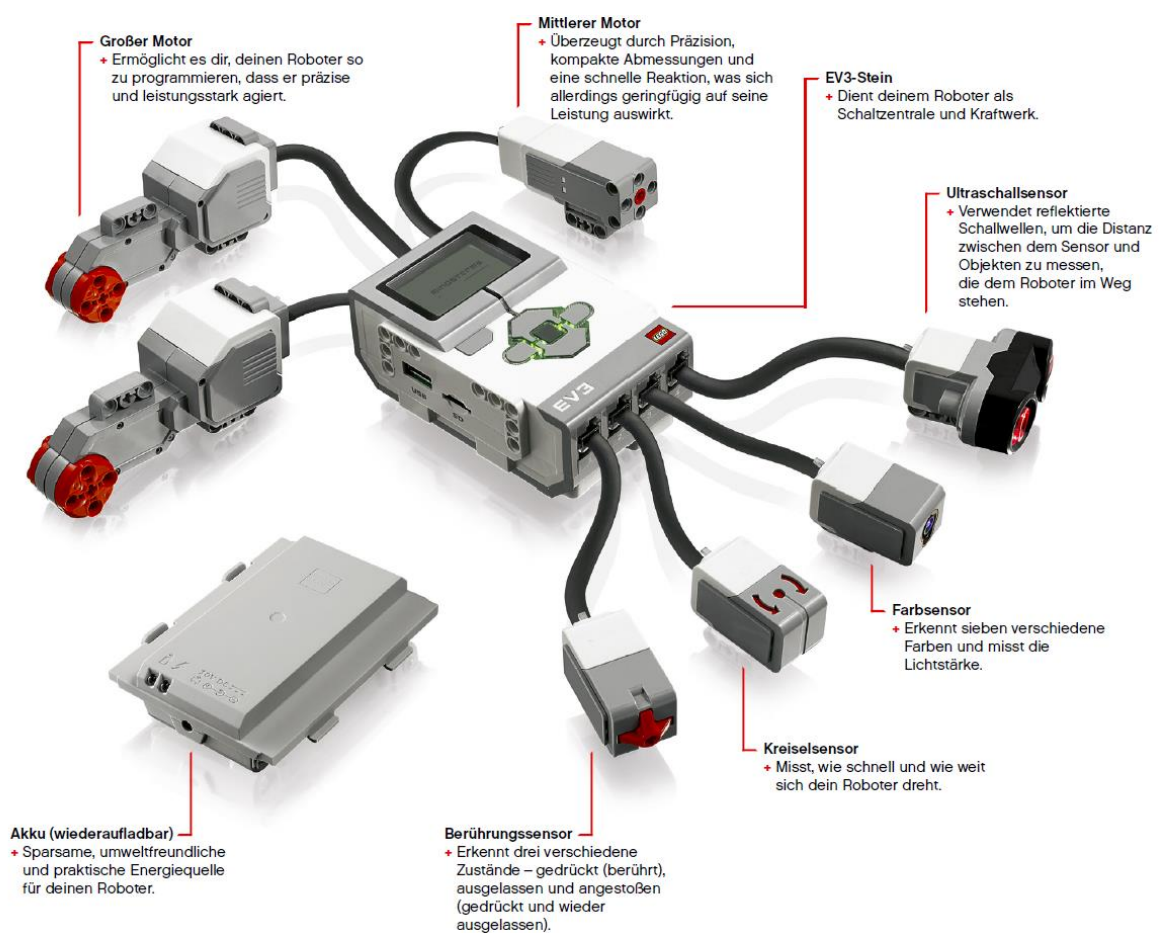


Abb. 1: LEGO MINDSTORMS EV3 Education-Set [3]

## EV3-Stein

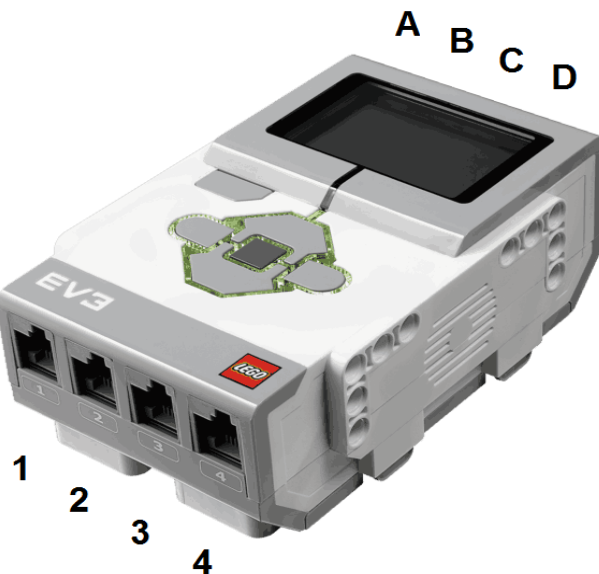


Abb. 2: EV3-Stein [3]

### INHALT DES SETS

Der EV3-Stein dient der Roboter als Schaltzentrale und Kraftwerk.

4 Eingabe-Anschlüsse: 1, 2, 3, 4 (zum Anschließen von Sensoren an den EV3-Stein)

4 Ausgabe-Anschlüsse: A, B, C, D (zum Anschließen von Motoren an den EV3-Stein)

1 Mini-USB-PC-Buchse (zum Anschließen des EV3-Steins an einen Computer)

USB-Host-Port (zum Anschließen eines WiFi-Dongles sowie für Reihenschaltungen)

Micro-SD-Karten-Buchse (zur Erweiterung des auf dem EV3-Stein verfügbaren Speicherplatzes)

Eingebauter Lautsprecher [4]

## Motoren

### Großer Motor



Großer Motor

Abb. 3: EV3, großer Motor [3]

Der große Motor ist ein leistungsstarker Motor. Er ist optimal mit der Achse und Rädern darauf ausgelegt, als Fahrgestell der Roboter zu dienen. [3]

### Mittler Motor







Mittlerer Motor

Abb. 4: EV3, mittlerer Motor [3]

Der mittlere Motor ist jedoch kleiner und leichter als der große Motor. Folglich ist er reaktionsfreudiger als der große Motor, das heißt er spricht schneller auf Befehle an. [3]

## Sensoren

Sensortypen	Verwendungsart	Modus	Anwendungsfälle
<p><b>Berührungssensor</b></p>  <p>Abb. 5: Berührungssensor [3]</p>	<p>Der Sensor erkennt, wenn die Rote Taste des Sensors gedrückt und wieder ausgelassen wird.</p>	<ol style="list-style-type: none"> <li>1. gedrückt</li> <li>2. ausgelassen</li> <li>3. angestoßen</li> </ol>	<p>Mit dem Sensor kann unser Roboter beispielsweise erkennen, ob er gegen eine Wand bzw. einen Gegenstand fährt.</p>
<p><b>Kreiselsensor</b></p>  <p>Abb. 6: Kreiselsensor [3]</p>	<p>Der Sensor erkennt, wenn man den Sensor in Richtung der Pfeile dreht, kann der Sensor die Drehrate von 440 Grad/s messen.</p>	<ol style="list-style-type: none"> <li>1. Winkel</li> <li>2. Änderungsrate</li> </ol>	<p>Genaue Drehbewegungen</p>
<p><b>Farbsensor</b></p>  <p>Abb. 7: Farbsensor [3]</p>	<p>Der Sensor erkennt, wenn die Farbe oder die Stärke des Lichts durch das kleine Fenster an der Vorderseite des Sensors eindringt.</p>	<ol style="list-style-type: none"> <li>1. Farbmodus</li> <li>2. Stärke des reflektierten Lichts</li> <li>3. Stärke des Umgebungslichts</li> </ol>	<p>Der Roboter vorgezeichneten Linien folgen; Farbe unterscheiden</p>
<p><b>Ultraschallsensor</b></p>  <p>Abb. 8: Ultraschallsensor [3]</p>	<p>Der Sensor misst, die Distanz zu einem Objekt, das sich vor dem Sensor befindet.</p>	<ol style="list-style-type: none"> <li>1. Messen</li> <li>2. Vorhanden (Test auf vorhandenes Signal)</li> </ol>	<p>Der Roboter vor Hindernis stoppen; An der Wand entlang fahren.</p>

## Grundlegende Bausteine

### a. Balken

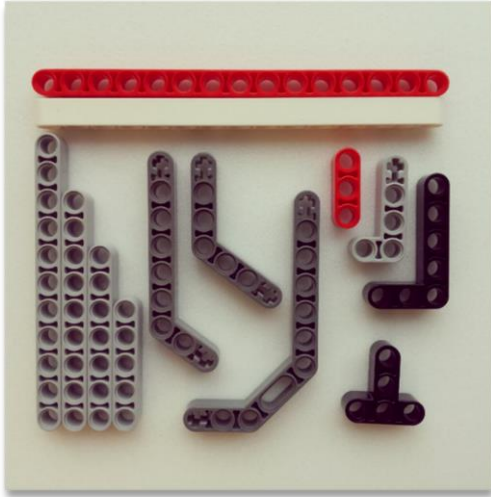


Abb. 9: EV3-Sets, Balken

Balken sind die wichtigsten Konstruktionsteile. Balken gibt es in einer Vielzahl von Formen, Längen und Farben. Üblicherweise wird zwischen geraden, winkligen und rechtwinkligen Balken unterschieden. Gemessen wird die Länge der Balken in Modulen. Ein Modul ist der Abstand zwischen den Mittelpunkten zweier Löcher. [5]

### Verbindungsstücke



Abb. 10: EV3-Sets, Verbindungsstücke

Verbindungsstücke dienen, weil der Name bereits sagt, dazu, Verbindungen zwischen Teilen zu ermöglichen. Aufgrund der ausschließlich noppenlosen Teile im LEGO MINDSTORMS EV3-Sets ist die Wahl der geeigneten Verbindungsstücke für den Bau eines stabilen Roboters sehr wichtig. [5]

## Zahnräder



Abb. 11: EV3-Sets, Zahnräder

Zahnräder bieten die Möglichkeit, den Roboter um viele interessante Konstruktionen zu erweitern. Neben der eigentlichen Aufgabe, die Bewegung von Motoren weiterzugeben, dienen Zahnräder auch dazu, Kraft und Geschwindigkeit einer Bewegung bzw. eines Motors zu verstärken bzw. zu reduzieren. Ebenso können Zahnräder dazu genutzt werden, die Richtung umzukehren. [5]

## Achsen

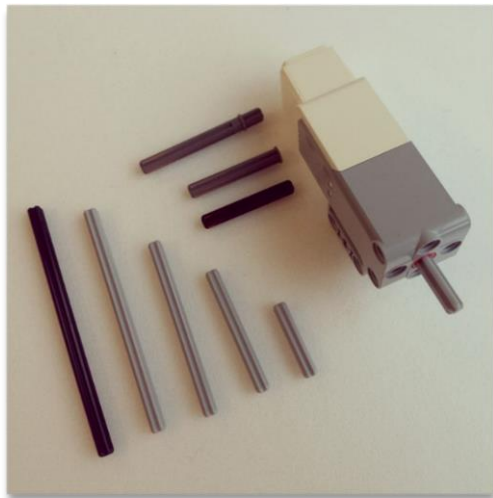


Abb. 12: EV3-Sets, Achsen

Achsen können ebenfalls zum Verbinden von Bauteilen genutzt werden. Häufiger werden Achsen aber dazu verwendet, Umdrehungen eines Motors auf Zahnräder oder Räder zu übertragen. Mittels Achsen können also sich bewegende/drehende Konstruktionen in einen Roboter eingebaut werden. Die Länge einer Achse kann am leichtesten anhand der Lochbalken bestimmt werden. Die Länge der Achse entspricht dabei der Anzahl der Löcher. [5]

## 2.2.2 Sensorik

### Technologie des Berührungssensors

Der Berührungssensor ist ein sehr einfacher, aber auch robuster Sensor. Sein roter Taster, in Abbildung 13 in der Mitte zu erkennen, wird bei Druck in den Sensor hineingeschoben. Dort schließt er einen Stromkreis, der Strom kann also fließen. Wenn der Taster nicht gedrückt wird, wird er durch eine Feder wieder nach außen geschoben und der Stromkreis ist unterbrochen. Der Berührungssensor hat also drei verschiedene Zustände: Ausgelassen, Gedrückt oder Angestoßen bzw. 0, 1 oder 2. Diese Zustände werden vom EV3-Stein erkannt und können dort noch weiter unterschieden werden.



Abb. 13: EV3-Berührungssensor [3]

Die Achsaufnahme im Taster bietet dem Konstrukteur eines Roboters vielfältige Möglichkeiten, diesen Sensor zu verwenden. [5]

### Erkennung der Hindernisse mit einem Ultraschallsensor

Der Ultraschallsensor gehört zu den aktiven Sensoren, denn er sendet aktiv Ultraschallwellen aus, um sie anschließend mit dem Empfänger wieder wahrzunehmen. Ultraschallwellen sind Schallwellen, deren Frequenzen oberhalb des Hörbereichs von ca. 16Hz bis 20kHz liegt, also  $>20\text{kHz}$  und ist daher vom Menschen nicht wahrnehmbar. Hz (Hertz) ist die Einheit für die Frequenz: Anzahl der Schwingungen pro Sekunde. Einige Tiere, wie z.B. der Hund oder die Fledermaus, können Ultraschallwellen hören. Fledermäuse senden sogar selber Ultraschallwellen aus, um sich damit zu orientieren, ganz ähnlich wie beim Ultraschallsensor.

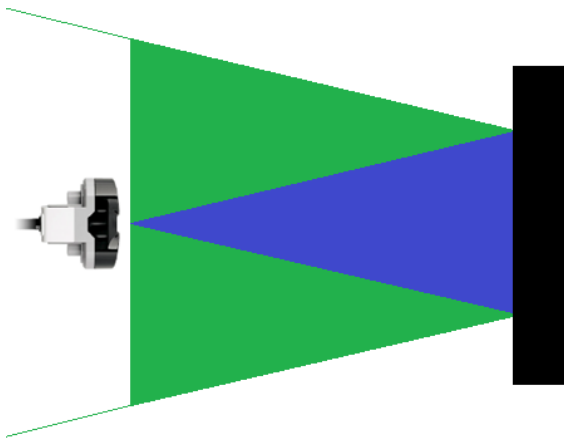


Abb. 14: Die Reflexion an einem Hindernis [5]

Links im Bild ist der Ultraschallsender zu sehen. Er sendet Ultraschallwellen aus und nimmt den Schall wahr, der von einem Hindernis reflektiert wird. Der Ultraschallsensor misst die Zeit, die der Schall vom Aussenden bis zum Empfangen benötigt. Die Geschwindigkeit, mit der Schallwellen sich fortbewegen, ist in der Regel konstant, sodass die Entfernung zum Hindernis aus der gemessenen Zeit und der Geschwindigkeit berechnet werden kann.

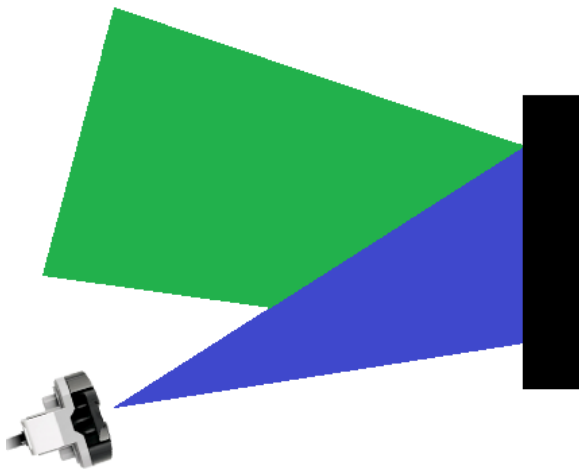


Abb. 15: Die Reflexion an einem Hindernis [5]

Verschiedene Formen der Hindernisse können das Messergebnis verfälschen. Ist der Winkel zu steil, wird der Schall so abgelenkt, dass er den Empfänger nicht mehr erreichen kann.

Das Material der Oberfläche des Hindernisses kann die Reflexion auch verhindern. Ein Plüschtier z.B. wird nicht oder nur schlecht erkannt. [5]



## Unterscheiden von Objekten mit dem Farbsensor

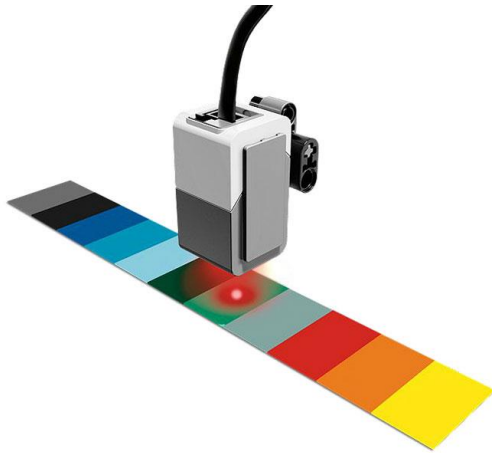


Abb. 16: Farbmodus [3]

Der Farbsensor besteht aus einer Fotodiode und drei LEDs. Die Fotodiode befindet sich hinter dem kleinen unteren Fenster des Farbsensors, siehe Abbildung 4. Sie misst das Licht, das auf ihre Oberfläche fällt, und gibt es als Zahl an die Elektronik des Sensors weiter. Hinter dem oberen größeren Fenster sind drei LEDs angebracht. Sie erzeugen farbiges Licht, rot, grün oder blau, das dann von einem Gegenstand reflektiert wird. Je nachdem, ob und welche LEDs gerade von der Elektronik des Farbsensors eingeschaltet sind, kann der Farbsensor verschiedene Messungen vornehmen.



Abb. 18: Farbsensor [5]



Abb. 17: RGB-Verfahren

Ein Dreibereichs-Farbsensor arbeitet nach dem additiven RGB-Verfahren (Rot-Grün-Blau). Alle drei Farben zusammen ergeben Weiß.

Die drei LEDs werden abwechselnd angeschaltet und das reflektierte Licht gemessen. Je nach Anteil und Stärke von Rot, Grün und Blau kann der Farbsensor die Farbe eines Gegenstandes erkennen. Dabei unterscheidet er zwischen sieben Farben. [5]

## Exaktes Steuern mit dem Kreisel sensor(Gyrosensor)

Ein Gyrosensor ist ein Beschleunigungs- oder Lagesensor, der auf kleinste Beschleunigungen, Drehbewegungen oder Lageänderungen reagiert. Er erfasst die Richtungsänderungen des Roboters. Sobald man den Roboter während der Fahrt nach links oder nach rechts lenkt, erfasst der Gyrosensor diese Bewegungen und liefert ein Signal (in einer Spannungsänderung bezogen auf die Drehgeschwindigkeit angegeben) an den EV3-Stein. [6]

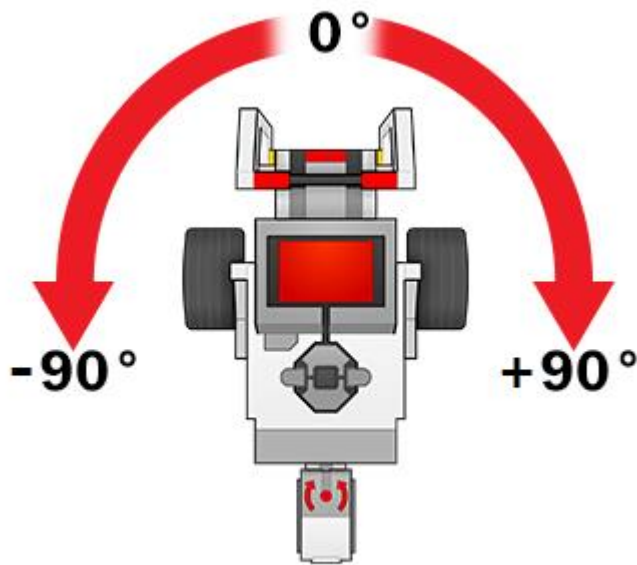


Abb. 19: Roboter mit Kreisel sensor [7]

### 2.2.3 EV3 Programmierungstechnik

Wie bereits erwähnt, besitzt die EV3-Software eine grafische Darstellungsform. Dies bedeutet: Anders als bei anderen Programmiersprachen wie zum Beispiel JAVA, C, Python erfolgt die Programmierung hier nicht durch das Schreiben von Text, sondern durch die Verknüpfung grafischer Blöcke und Elemente.

Bei der EV3-Software handelt es sich um eine datenflussorientierte Programmiersprache, deren Elemente in EV3 hauptsächlich Blöcke sind. Diese Blöcke stellen Operationen dar, die Daten entweder erzeugen (z.B. Sensor-Block) oder manipulieren (z.B. Daten-Block)

Es gibt eine große Anzahl verschiedener Blöcke, mit deren Hilfe der EV3-Stein programmiert werden kann. Grob lassen sich die Blöcke in drei Hauptkategorien einordnen.

1. Blöcke zur Steuerung der elektronischen Komponenten (Motoren, Sensoren, EV3-Stein)
2. Blöcke zur Ablaufsteuerung des Programms
3. Blöcke zur Manipulation von Daten [5]

## 2.2.4 EV3 Konstruktionen Technik

### Starre noppenlose Verbindungen

Die weitaus meisten der nicht elektronischen Teile im EV3-Set sind nicht speziell für die LEGO-Mindstorms-Produktreihe produziert worden, sondern gehören zur LEGO-Technik-Serie, die hauptsächlich technische Konstruktionen/Bauanleitungen wie Autos, Baumaschinen oder Pneumatik beinhaltet. Die Technik-Serie enthält hauptsächlich Teile, die nicht auf dem ursprünglichen Prinzip der Noppen basieren. Stattdessen baut es auf das sogenannte »noppense« Bauen auf. Hierzu werden Balken hauptsächlich über Stecker miteinander verbunden. Damit lassen sich wesentlich stabilere Modelle konstruieren, als dies mit dem Noppen-Prinzip möglich ist. [5]

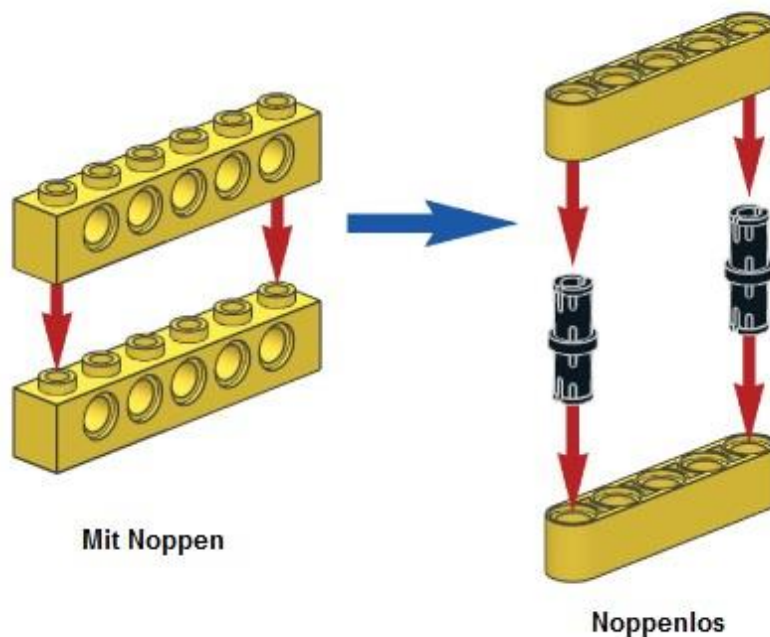


Abb. 20: Die noppense Balken [8]

### Vorteile von noppense Konstruktionen

Im Allgemeinen weisen Verbindungen ohne Noppen folgende Vorteile auf:

- Die Teile lassen sich auf einfache Weise in jeder Richtung kombinieren, was dreidimensionales Bauen ermöglicht.
- Die Teile lassen sich auf einfache Weise mit den meisten modernen Technik-Elementen wie Motoren und Aktoren kombinieren.

- Die Teile haben mehr Stiftlöcher, was kompakte Konstruktionen ermöglicht.
- Es sind nur selten zusätzliche Sicherungen erforderlich (wenn man starre Verbindungen verwendet).
- Die Teile sind kleiner und leichter als solche mit Noppen.

### **Nachteile von noppenlosen Konstruktionen**

Im Allgemeinen weisen Verbindungen ohne Noppen folgende Nachteile auf:

- Sie sind weniger starr; größere noppenlose Strukturen müssen sehr komplex gebaut (oder gesichert) werden, um ihre Form zu fixieren.
- Die Teile lassen sich schwerer mit Nicht-Technik-Elementen kombinieren.
- Für eine starre Verbindung sind mindestens zwei Stifte erforderlich; zum Aussteifen werden häufig zusätzliche Teile mit komplizierten Formen benötigt.
- Die Konstruktionen sehen nicht immer so gut aus wie diejenigen aus Steinen. [8]

## 2.3 Übersicht bestehenden EV3-Lernmaterialien

Es gibt derzeit schon eine Reihe von Dokumentationen und Tutorials, die sich auf den EV3-Roboter und den Wettbewerb beziehen. Hiermit wurden solche Informationen als drei unterschiedliche Quellen sortiert:

- Dokumente von Lego
- Dokumente von Lehrerinnen / Lehrern
- Dokumente von Schülerinnen / Schülern

Hierfür können folgende Beispiele für oben genannte Dokumente als drei Kategorien angeführt werden.

### 1. Basiswissen

#### von LEGO:

- EV3-Bedienungsanleitung;
- <http://www.lego.com/de-de/mindstorms/products/mindstorms-ev3-31313>;

#### von Lehrerinnen / Lehrern:

- <http://ev3-wissen.engeln.info/doku.php?id=start>;
- <https://www.youtube.com/watch?v=b0ogu5r9EWw>;
- Das EV3 Roboter Universum, M.P. Scholz

#### von Schülerinnen / Schülern:

- <https://www.youtube.com/watch?v=N5fzeplrFw8&nohtml5=False>;

### 2. Verhalten des Roboters

#### von LEGO:

- <http://www.lego.com/de-de/mindstorms/videos>;
- <http://www.lego.com/de-de/mindstorms/learn-to-program>;

#### von Lehrerinnen / Lehrern:

- <https://dual.tuhh.de/informationen-zu-dual-tuhh1/events-messen/53-robotik-kurse/schueler/robotikkurse/robotertechnik>;
- <https://www.youtube.com/watch?v=eRK3ZR0eq6E>;
- <https://www.youtube.com/watch?v=ppNE-jYWYBg&nohtml5=False>;
- Das LEGO-Mindstorms-EV3-Labor, Daniele. Benedetteilli
- Programmieren Lernen mit EV3, Terry. Griffin

#### von Schülerinnen / Schülern:

- <https://www.youtube.com/watch?v=mZaeaBtd4-0>;
- [https://www.youtube.com/watch?v=tT\\_bqRjt2bM](https://www.youtube.com/watch?v=tT_bqRjt2bM);
- <https://www.youtube.com/watch?v=gh-HWmNgXWE&nohtml5=False>;

### 3. Ideen für den FLL-Wettbewerb

#### von LEGO:

- [http://www.first-lego-league.org/de/;](http://www.first-lego-league.org/de/)
- [http://www.first-tech-challenge.org/de/;](http://www.first-tech-challenge.org/de/)
- <https://www.youtube.com/user/FLLGlobal;>

#### von Lehrerinnen / Lehrern:

- <https://www.youtube.com/watch?v=7kBBKMW9uR0&nohtml5=False;>
- <https://www.youtube.com/watch?v=ZZCi6o9OJA8&nohtml5=False;>

#### von Schülerinnen / Schülern:

- [https://www.youtube.com/watch?v=95qiH6\\_6OBq&feature=youtu.be;](https://www.youtube.com/watch?v=95qiH6_6OBq&feature=youtu.be;)
- <https://www.youtube.com/watch?v=10U9NbF3rz4&nohtml5=False;>
- <https://www.youtube.com/watch?v=Pcl-rUcHeyw&nohtml5=False;>
- [https://www.youtube.com/watch?v=G-1eS\\_v8IJY&nohtml5=False;](https://www.youtube.com/watch?v=G-1eS_v8IJY&nohtml5=False;)

Jede Quelle wurde im folgenden Bewertungsschema nach wichtigen Kriterien bewertet.

Ermittlung der Wichtigkeit aller Kriterien:

2-wichtiger 1-gleich wichtig 0- unwichtiger	1	2	3	4	5	6	7	$\Sigma$	Wichtigkeit in %
1. informativ	1	0	0	2	2	1	0	6	13
2. motivierend	2	1	0	2	2	1	0	8	16
3. verständlich	2	2	1	2	2	1	1	11	22
4. Kosten	0	0	0	1	0	1	0	2	4
5. zugkräftiges Video	0	0	0	2	1	1	0	4	8
6. ausführlich	1	1	1	1	1	1	0	6	13
7. zuverlässig	2	2	1	2	2	2	1	12	24

In der Tabelle sind 7 Kriterien aufgeführt, die nun miteinander verglichen werden.

Eine 1 bedeutet, dass die Kriterien gleich wichtig sind, eine 2, dass das Kriterium wichtiger als das andere ist und 0 eben umgekehrt.

Die Einzelpunkte werden nun zeilenweise zusammengezählt, als Summe aufgeschrieben und letztendlich auch prozentual bewertet. 100% sind dabei die höchste erreichbare Punktzahl.

## Bewertungsschema:

Kriterienkatalog	Wichtigkeit in %	Lego	Lehrerinnen / Lehrern	Schülerinnen / Schülern
		Punkte	Punkte	Punkte
informativ	13	7	7	6
motivierend	16	8	7	8
verständlich	22	8	8	7
Kosten	4	6	7	9
zugkräftiges Video	8	9	6	6
ausführlich	13	7	8	6
zuverlässig	24	9	9	5
Gesamte Punktzahl		7,98	7,75	6,42

Den drei unterschiedlichen Quellen wurden jeweils Punkte von 1 bis 10 zugeordnet. 10 ist dabei die höchst mögliche Bewertung. Die gesamten Punkte aller drei Quellen wurden durch die Wichtigkeit aller Kriterien und die Summe aller Punkte der Kriterien errechnet.

Dieses Bewertungsschema zeigt die Bewertungen der Dokumente von LEGO, Lehrerinnen / Lehrern und Schülerinnen / Schülern. Die haben eigene Vorteile und Nachteile. Durch die zusammengefassten Punkte werden die offizielle LEGO Dokumente empfohlen.



## 3. Konzept für das Tutorial

### 3.1 Bau des Roboters

#### 3.1.1 Wichtige Bauprinzipien

##### Dreieckig Struktur

Es gibt einen erheblichen Unterschied zwischen dreieckigen und viereckigen Strukturen. Eine dreieckige Struktur kann Kräften besser widerstehen.

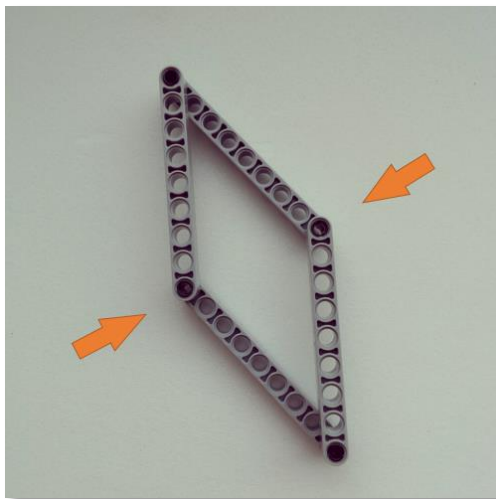


Abb. 22: die parallele Struktur

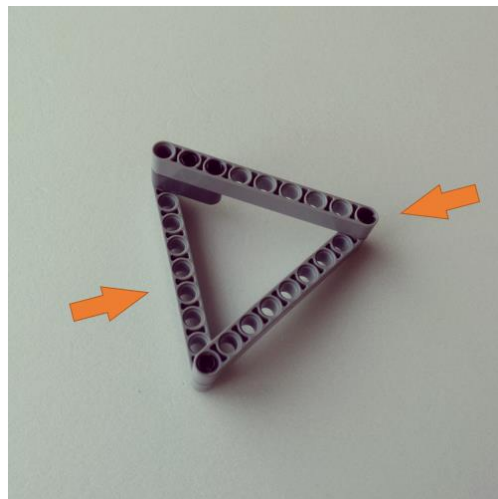


Abb. 21: die dreieckige Struktur

Wenn Kraft auf ein Parallelogramm ausübt wird, kann es ganz leicht zusammengedrückt werden. Eine dreieckige Struktur kann Kräften ohne Verformung widerstehen.

##### Balkenverlängerung

Wenn Balken verlängert werden müssen, können zwei oder mehr Balken über Pins mit Reibung verbunden werden.

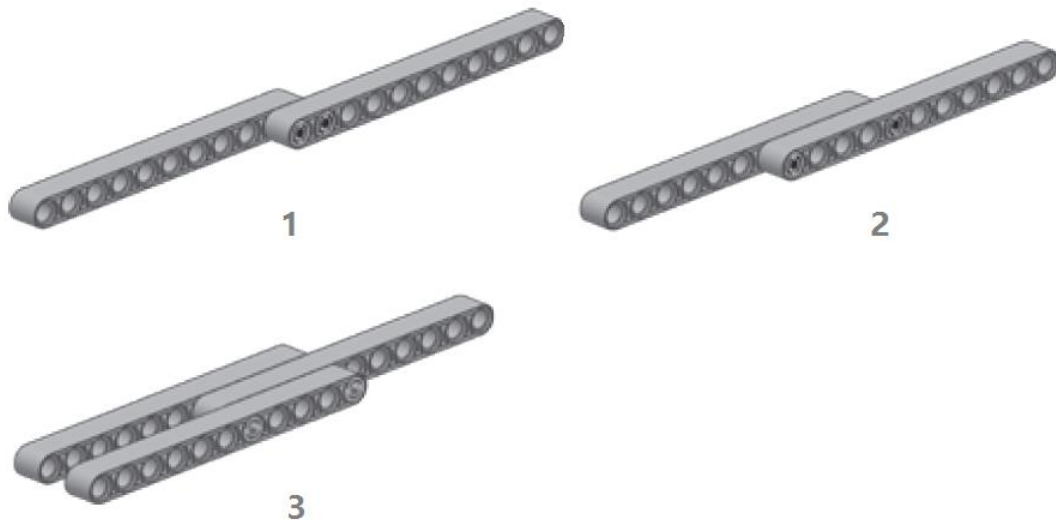


Abb. 23: Balken Verlängerung [9]

1. Die Balken 1 sind mit schwarzen Pins verbunden und überlappen sich nur über zwei Löcher. Diese Konstruktion ist zwar gerade, aber immer noch biegsam.
2. Es ist zu sehen. Eine Verlängerung der Überlappung auf fünf Löcher macht die Konstruktion fester.
3. Man kann auch die langen blauen Pins verwenden, um die Konstruktion dicker zu machen und dadurch zu verstärken.

## Montieren der Zahnräder

### Grundlegendes Wissen in EV3-Zahnräder

- Das Maß für Zahnräder ist die Anzahl der Zähne.
  - Ob zwei Zahnräder miteinander kombiniert können oder nicht, hängt von ihrem Radius ab.
  - Zahnräder weisen ein kreuzförmiges Mittelloch auf, sodass sie auf Achsen gesteckt werden können. Damit kann eine Drehbewegung von einer Achse auf eine andere übertragen werden. [9]
1. Zahnräder ändern die Richtung, die Geschwindigkeit und das Drehmoment einer Drehbewegung

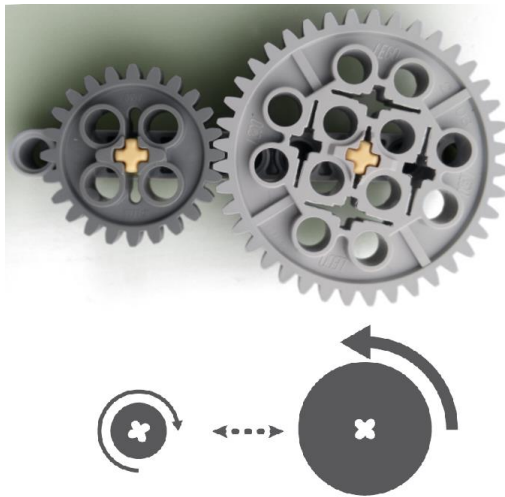


Abb. 24: Zahnräder

Wenn das kleinere Zahnrad als Antriebsrad fungiert, verringern sich sowohl die Anzahl der Umdrehungen, als auch die Rotationsgeschwindigkeit des angetriebenen Zahnrads. Aber das Drehmoment des angetriebenen Zahnrades steigt.

Die Abbildung 24 zeigt ein Zahnrad mit 24 Zähnen, das ein Zahnrad mit 40 Zähnen antreibt. Die Übersetzung wird durch das Verhältnis der Anzahl der Zähne bestimmt. In diesem Beispiel beträgt dieses Verhältnis der Zähne  $40:24 = 5:3$ . Das bedeutet, dass sich das 24z-Rad fast zweimal so schnell dreht wie das 40z-Rad.

## 2. Rechtwinklige Verzahnung

Neben der eigentlichen Aufgabe, die Bewegung von Motoren weiterzugeben, wird die Änderung der Richtung der Achse benötigt (siehe Abb. 25). Im LEGO-EV3-Set sind einige Kegelräder von unterschiedlicher Größe enthalten, die miteinander verbunden werden können. Damit wird die rechtwinklige Verzahnung erreicht.

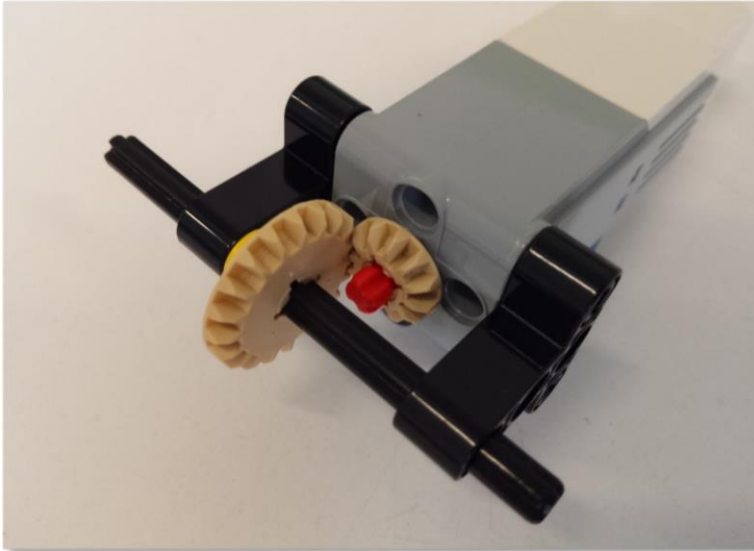


Abb. 25: Rechtwinklige Verzahnung

### 3. Schneckengetriebe

Schneckengetriebe sind eine Kategorie der Schraubwalzgetriebe und bestehen aus einer schraubenformigen sogenannten Schnecke, die bei Drehbewegung ein in diese greifendes Zahnrad (Schneckenrad) dreht.

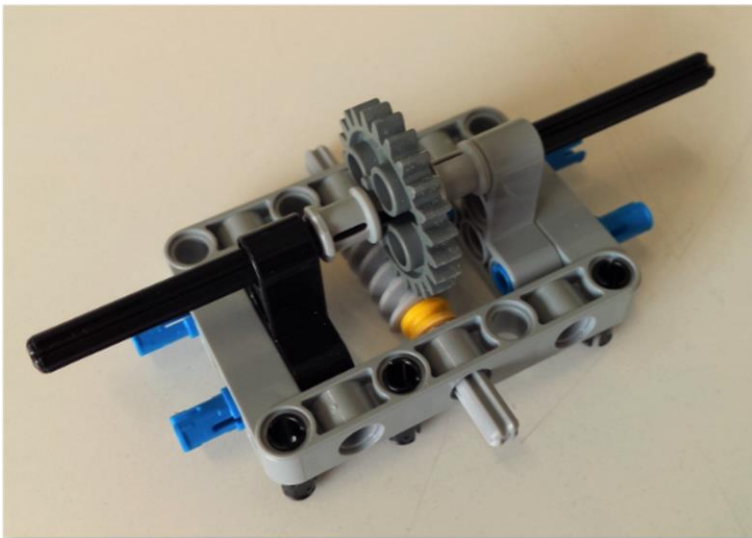


Abb. 26: Schneckengetriebe

Das Schneckengetriebe ist imstande, Kraft zu ubertragen und die Umdrehungsgeschwindigkeit zu reduzieren. Aber das Schneckengetriebe ist selbstsperrend. Das heist, wenn man die Schnecke dreht, kann man damit ein anderes Zahnrad antreiben, aber umgekehrt ist es nicht moglich, die Schnecke durch Drehen des anderen Rads zu bewegen.

### 3.1.2 Problemstellungen und Lösungsansätze

#### Allgemeine Lösung zum Durchdrehen der Zahnräder

Es kann immer passieren, dass zwei Zahnräder nicht so gut verzahnt sind, dass sie den Kontakt verlieren und Zähne durchrutschen. Für Lego-Anfänger ist das ein kniffliges Problem. Folgende Abbildung zeigt drei typische Arten von Zahnradkombinationen:



1 einwandfreier

2 mangelhafter

3 kein Kontakt

Abb. 27: Drei typische Arten von Zahnradkombinationen

Die Lösung wurde in folgender Tabelle dargestellt. Alle Zahnräder aus dem EV3-Set haben einen bestimmten Durchmesser.

Bezeichnung	Radius (in Lego-Einheiten)
8z-Zahnrad	0,5
12z-Doppelkegelrad	0,75
16z-Zahnrad	1
4z-Knebelrad	1
20z-Doppelkegelrad	1,25
24z-Zahnrad	1,5
Kleine Drehscheibe (28z)	1,75
36z-Doppelkegelrad	2,25
40z-Zahnrad	2,5
Große Drehscheibe (56z)	3,5

Abb. 28: Radien der verschiedenen Zahnräder [9]

Eine einwandfreie Zahnradkombination liegt vor, wenn die Summe der Radien ganzzahlig ist. Das ist beispielsweise bei der Kombination 8z mit 24z ( $0,5+1,5=2$ ) der Fall. [9]

## Empfehlenswertes Lenksystem für den Roboter

Ein flexibles Lenksystem für die Bewegung eines Mini-Roboters ist sehr sinnvoll. Durch die exakte Drehbewegung wird der Roboter bei einem FLL-Wettbewerb die unterschiedlichen Aufgaben besser erledigen.

Hier werden zwei typische Lösungsvarianten dargestellt:

### Ein Kugelrad als Schwenkrad

Mit dieser Bauart kann der Roboter viel exakter gesteuert werden. Diese Lenkung erfordert keinen zusätzlichen Antrieb. Der Bau dieser Struktur ist sehr einfach. Das Kugelrad ist leichter zu de-/montieren als eine Lenkung mit normalen Räder. Aber mit diesem Kugelrad kann der Roboter nur auf dem ebenen Boden fahren.

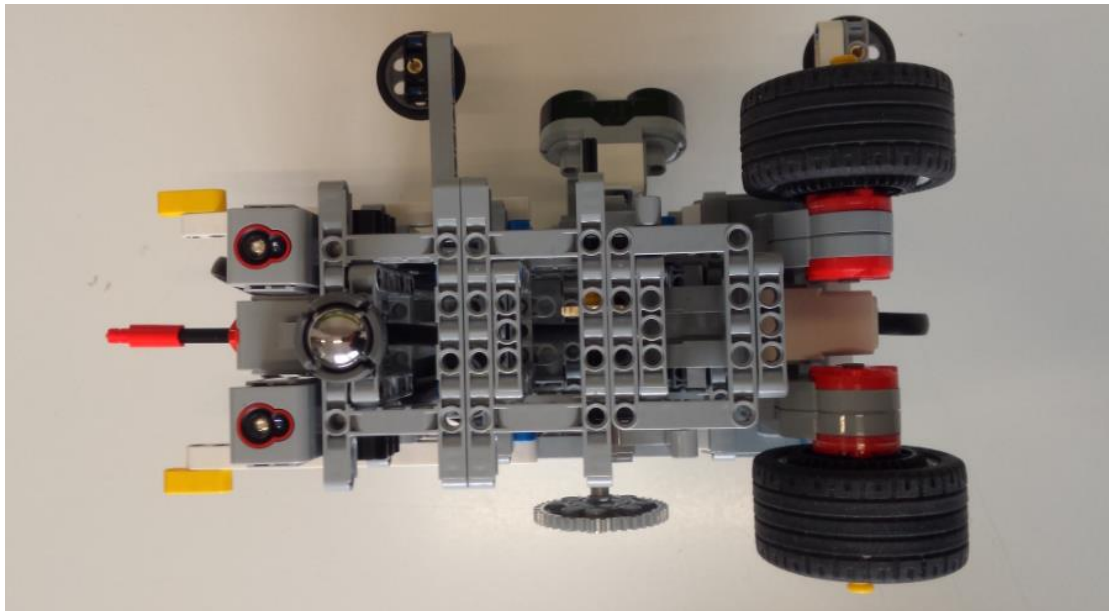


Abb. 29: Ein Kugelrad als Schwenkrad

### Ein ähnliches Lenksystem wie am Auto

Mit dieser Bauart kann sich der Roboter wie ein allgemeines Auto bewegen. Die beiden Schwenkräder wurden von einem mittleren Motor angetrieben. Dadurch schwenkt der Roboter nach links oder rechts. Ein Nachteil dieses Lenksystems ist, dass sich der Roboter nur in einem relativ großen Kreis drehen kann. Bei einem FLL-Wettbewerb auf dem kleinen Spielfeld ist dieser Roboter nämlich eingeschränkt.

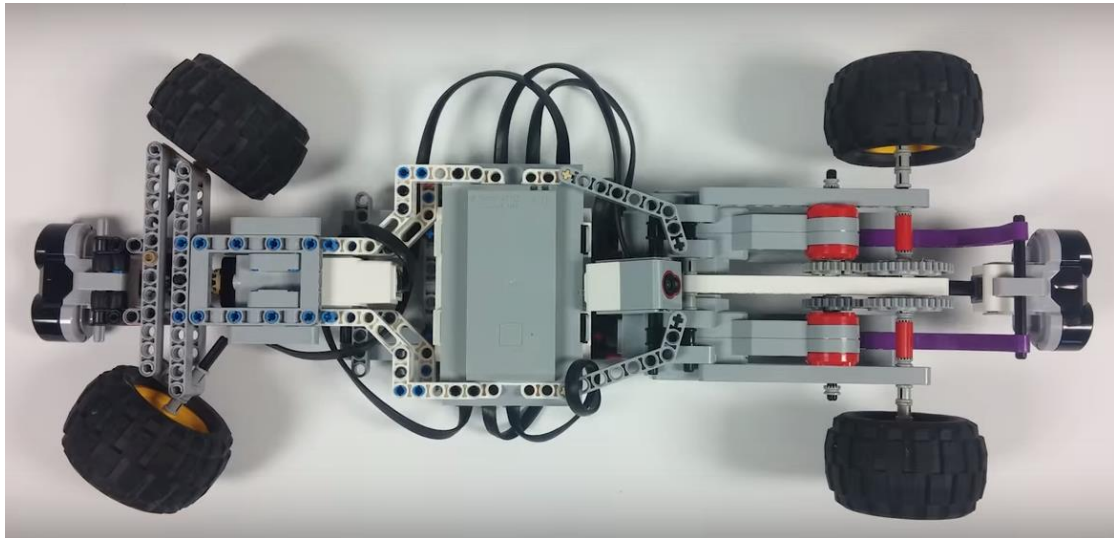


Abb. 30: Ein ähnliches Lenksystem wie am Auto

## **Effizienter Umbau des Roboters**

Für ein FLL Wettbewerb hat jedes Team pro Match 2:30 Minuten Zeit, um die Aufgaben zu lösen. Während des Matches dürfen zwei Teilnehmer jedes Teams den Roboter für unterschiedliche Aufgaben umbauen. Unter dem großen zeitlichen Druck ist ein einwandfreier effizienter Umbau erforderlich.

Mit allen LEGO-EV3 Teilen werden Balken hauptsächlich über Stecker (Pins) miteinander verbunden. Bei einem FLL-Wettbewerb gibt es jährlich ungefähr 12 unterschiedliche Aufgaben zu einem Thema. Die Möglichkeit liegt eine Haaresbreite über null, dass ein Roboter ohne Umbau alle Aufgaben erledigt. Deshalb fordert jedes Team einen basierten Unterbau, der die planare Bewegung ausführt und den Antrieb liefert, der die wechselbaren Ersatzteile z.B. Auslegerarm, Transportband oder Sammlungskiste ermöglicht.

Es werden formschlüssige Verbindungen für Umbau als Referenz angegeben:

Unter formschlüssigen Verbindungen versteht man einfache Verbindungen von Bauteilen durch Verbindungselemente, wie beispielsweise Stifte oder Bolzen. Für den Umbau des EV3-Roboters können diese verwendet werden. Die langen Balken könnten als Stifte benutzt werden. Die Steck-Balken des Ersatzteils kann man direkt im Unterbau stecken. Und das Getriebe wird durch die Verbindung der Zahnräder erreicht. Diese Verbindungsart ist sehr einfach sowie leicht ein- und abzubauen.



## 3.2 Programmierung des Roboters

### 3.2.1 Allgemeiner Aufbau von Blöcken

Alle Programmierblöcke zur Steuerung des Roboters befinden sich in den Programmierpaletten am unteren Rand der Programmierumgebung unter dem Programmierbereich. Die Programmierblöcke sind nach Typ und Einsatzzweck in Kategorien unterteilt, um die Suche nach dem gewünschten Block zu erleichtern.

In der folgenden Abbildung werden drei grundlegende Kategorien vorgestellt:

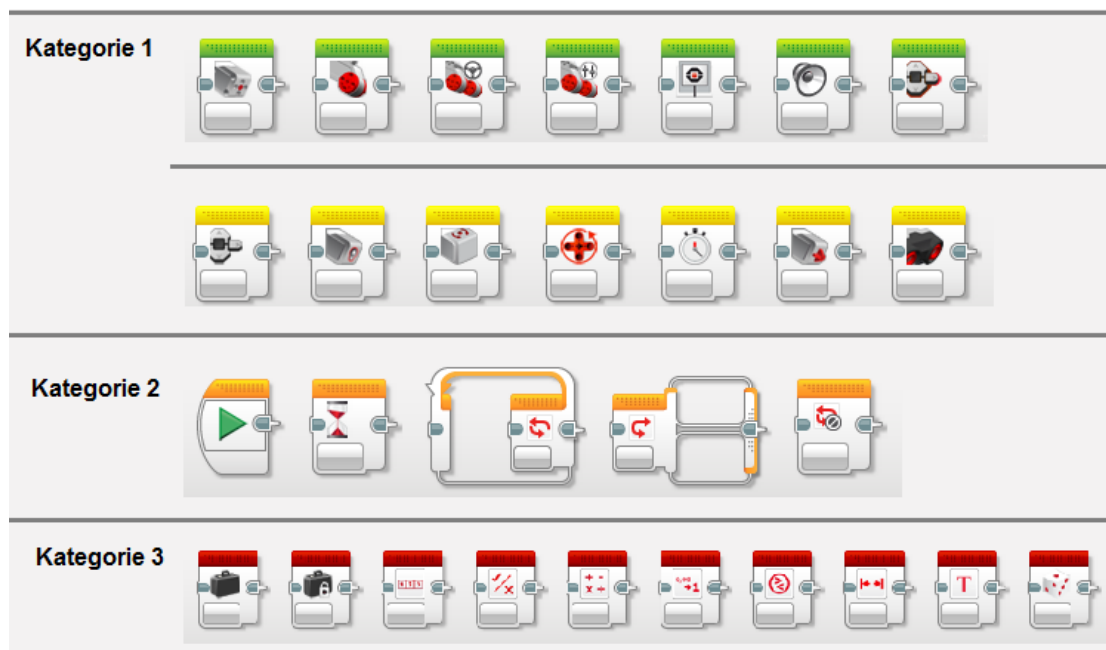


Abb. 31: Zuordnung der EV3-Programm-Blöcke [10]

- In die erste Kategorie fallen beispielsweise Blöcke, die zur Programmierung der EV3-Hardware, also der Sensoren, Motoren und der Funktionalität des EV3-Steins (Display, Lautsprecher, Status-Licht) benötigt werden.
- Mit Blöcken der zweiten Kategorie wird der generelle Programmablauf strukturiert bzw. geregelt. Beispielsweise kann mit einem Schleifen-Block ein bestimmter Programmteil mehrmals durchlaufen werden.
- Die dritte Kategorie beinhaltet Blöcke, mit deren Hilfe Operationen auf Daten ausgeführt werden. Soll zum Beispiel eine Variable gespeichert oder eine Rechenoperation ausgeführt werden, nutzt man üblicherweise Blöcke aus der dritten Kategorie. [5]

In den folgenden Abschnitt werden einige ausgewählte sehr nützliche und schwerverständliche Blöcke erklärt. Das ist eine den Anfängern in der EV3-Programmierung leicht verständliche Einführung.

## Warte-Block

Einer der wichtigsten Blöcke bei der Programmierung eines EV3-Roboters ist der Warte-Block. Abbildung.32 zeigt einen Warte-Block mit der Einstellung WARTE EINE SEKUNDE. Das Programm stoppt an dieser Stelle für eine Sekunde, bis es dann mit der Ausführung nachfolgender Blöcke weitermacht.



Abb. 32: Warte-Block [10]

Interessant wird die Verwendung des Warte-Blocks, wenn auf einen bestimmten Sensorwert gewartet werden soll. Wie in Abbildung gezeigt, können alle für den EV3 zur Verfügung stehenden Sensoren als Parameter im Warte-Block ausgewählt werden.

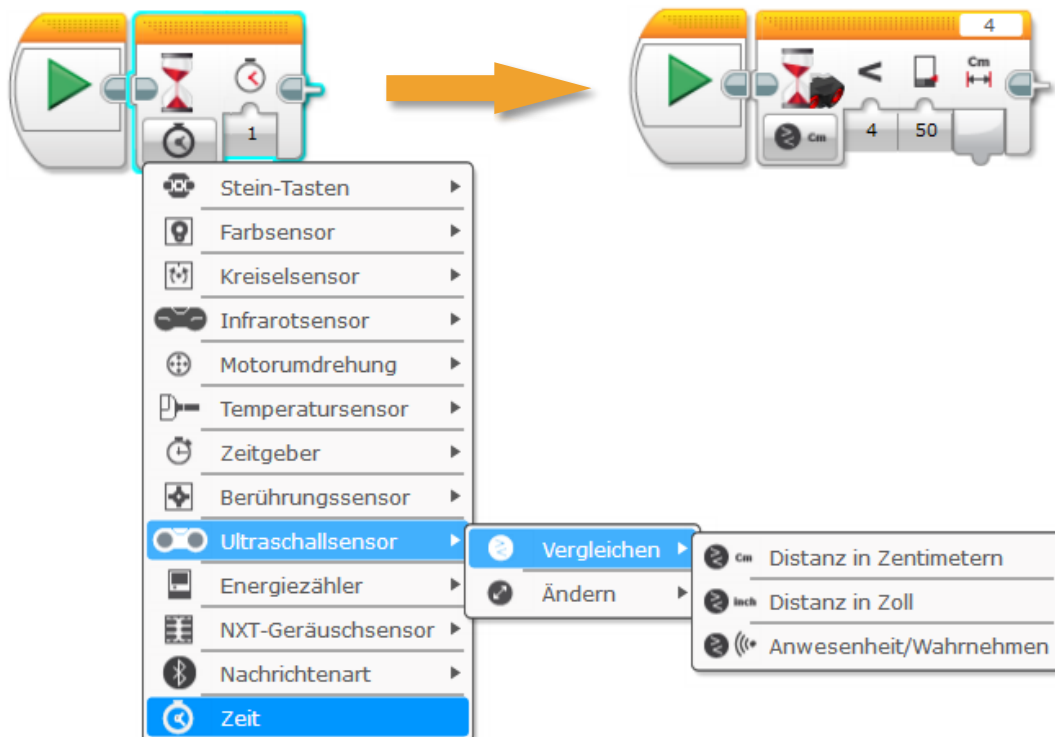


Abb. 33: Warte-Block im Modus Ultraschallsensor [10]

Mit der Auswahl eines Sensors können zusätzlich noch die Optionen **Vergleichen** und **Ändern** gewählt werden. Wird beispielsweise der Ultraschallsensor mit den Parametereinstellungen wie in oben dargestellte Abbildung.33 gewählt, wartet das Programm darauf, dass der vom Ultraschallsensor gemessene Wert kleiner als 50 cm ist. Erst danach werden die weiteren Programmblöcke abgearbeitet. [5]

## Schleifen-Block

Ebenso wie der Warte-Block ist auch der Block SCHLEIFE für die Ablaufsteuerung eines Programms von großer Bedeutung.

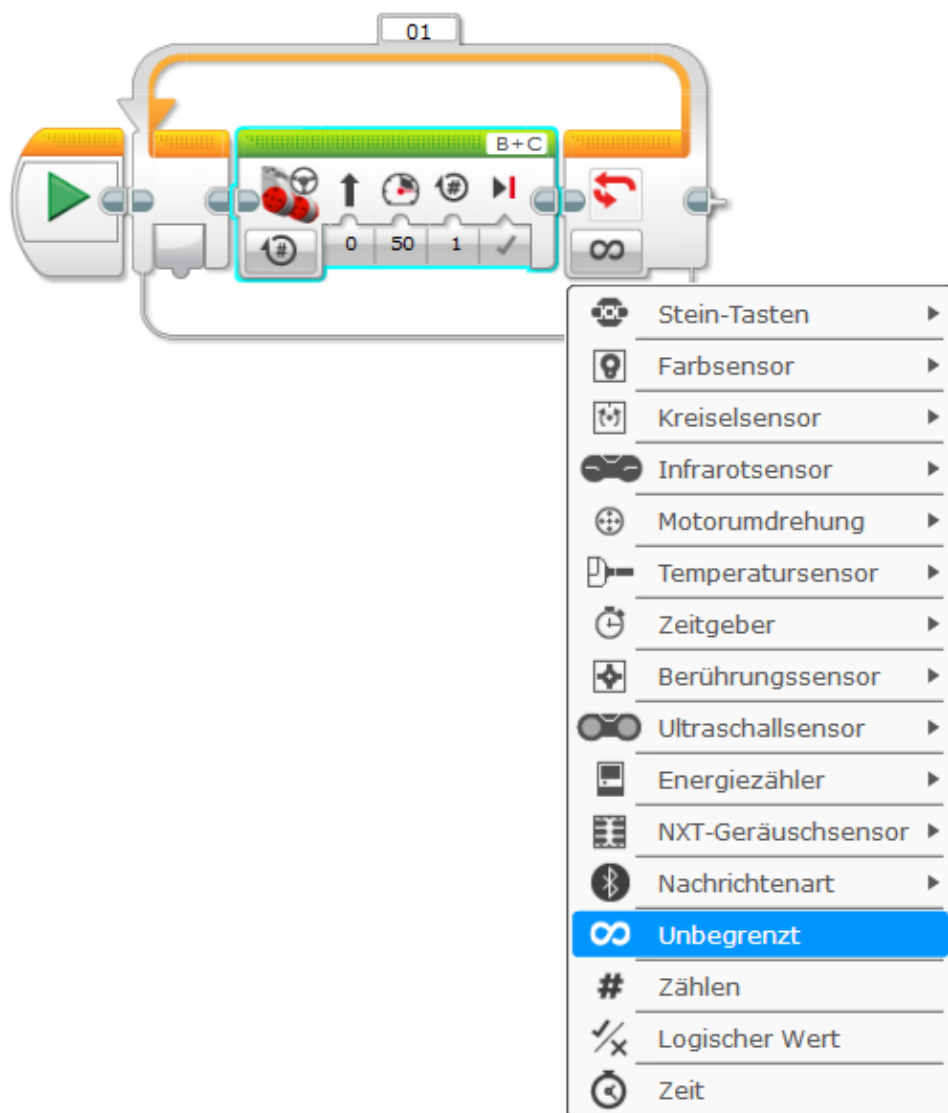


Abb. 34: Schleifen-Block im Modus Unbegrenzt [10]

Mit einer Schleife können ein oder mehrere EV3-Blöcke wiederholt werden. Hierfür wird der Block einfach in die Schleife gezogen. Anschließend kann die sogenannte Abbruchbedingung wie in Abbildung.34 dargestellt, der Schleife gewählt werden. Abbruchbedingung bedeutet: Unter welcher Voraussetzung soll der Schleife-Block enden, damit das Programm die nachfolgenden Programmblöcke abarbeiten kann. [5]

## Schalter-Block

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.

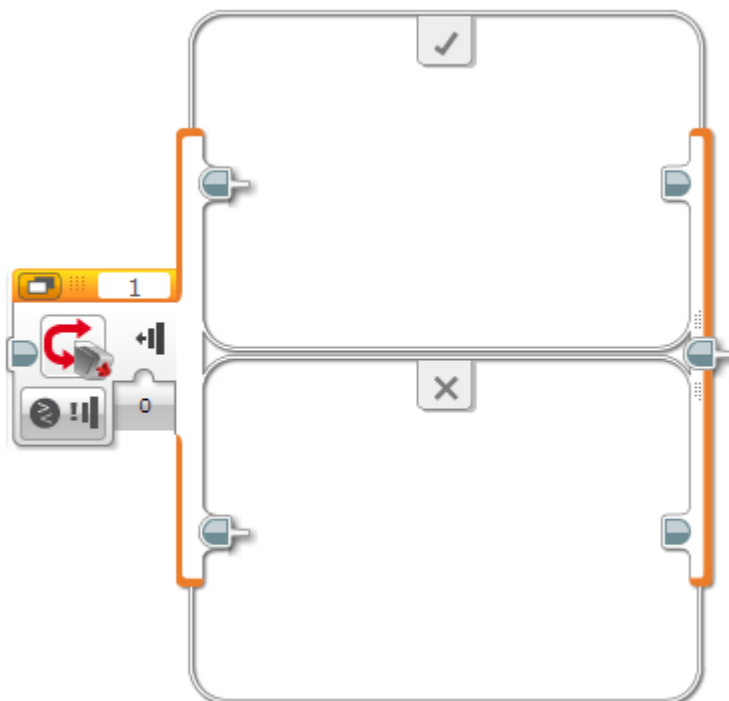


Abb. 35: Schalter-Block [10]

Der oben 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. [11]

## Variablen-Block

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

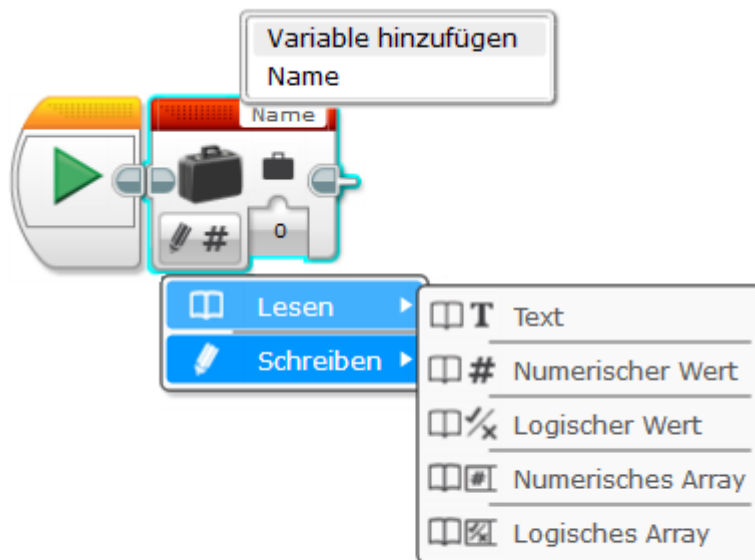


Abb. 36: Variablen-Block [10]

Jede Variable verfügt über einen Typ und einen Namen. Es gibt folgende Typen: Numerische Variable, Logische Variable, Text-Variablen, Numerisches Array und Logisches Array. Man kann den Namen der Variablen wählen, der verwendet wird, um die Variable zu identifizieren. [12]

## Mathe-Block

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

Wähle die zu verwendende mathematische Operation, indem man mithilfe der Modus-Auswahl einen Modus auswählt. Nach der Auswahl des Modus kann man die Werte für die Eingaben festlegen. Die verfügbaren Eingaben sind von Modus zu Modus verschieden. [12]

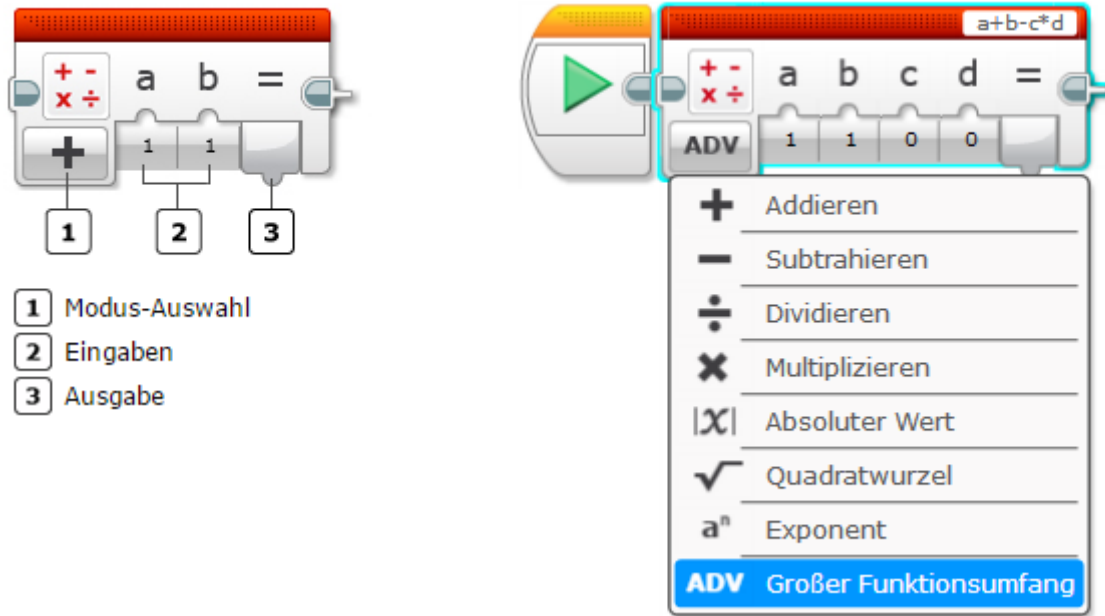


Abb. 37: Mathe-Block im unterschiedlichen Modus [10]

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. [12]

	Modus	Verwendete Eingaben	Ausgegebenes Ergebnis
Addieren		A, B	$A + B$
Subtrahieren		A, B	$A - B$
Multiplizieren		A, B	$A \times B$
Dividieren		A, B	$A \div B$
Absoluter Wert		A	A, wenn $A = 0$ ; $-A$ , wenn $A < 0$ Das Ergebnis ist immer $\geq 0$ .
Quadratwurzel		A	$\sqrt{A}$
Exponent		A (Basis), N (Exponent)	$A^N$
Erweiterte Operationen		A, B, C, D	$A + B - C * D$

Abb. 38: Der mathematische Operationsmodi [12]

Beim Modus ADV ist die Eingabe eigener Gleichungen möglich.

## Eigene-Blöcke

Mit der Option Eigene Blöcke können Programmblöcke zu einem einzigen Block zusammengefasst werden. Dadurch kann das eigene Programm übersichtlicher gestaltet und die im eigenen Block ausgewählten Programmblöcke in anderen Programm oder Projekten ebenfalls genutzt werden.

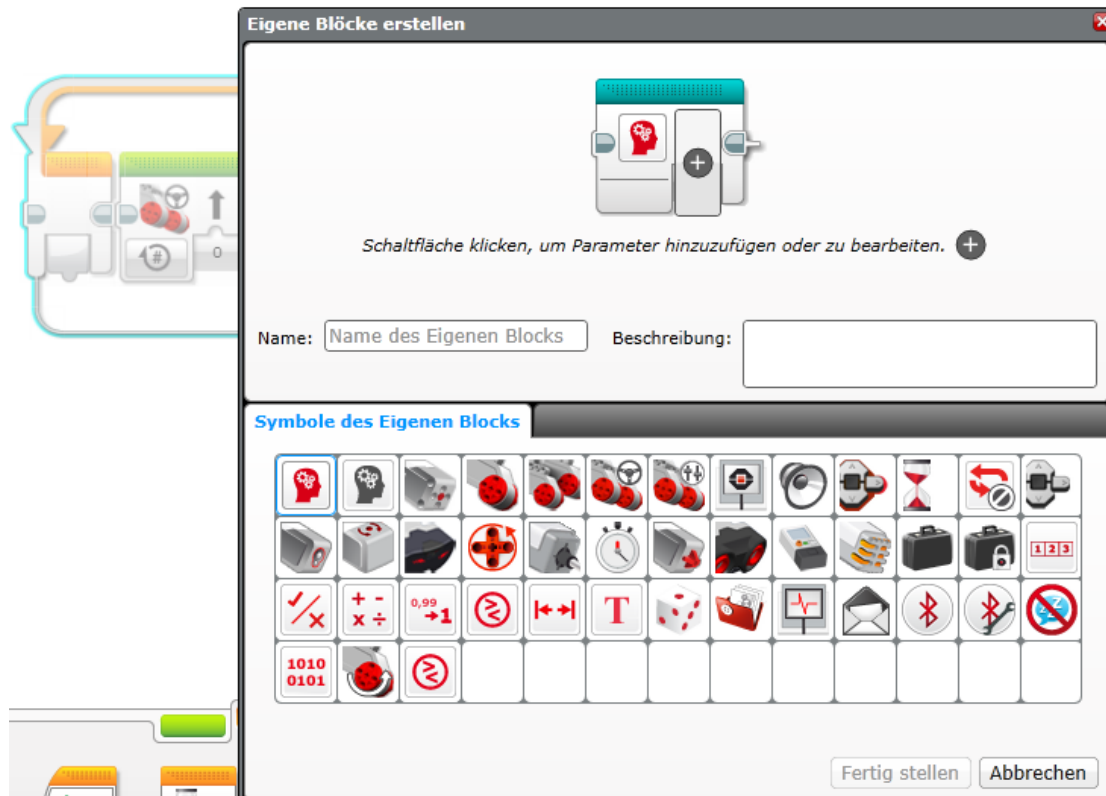


Abb. 39: Eigene-Block [10]

Zuerst muss das Programm natürlich programmiert werden. Anschließend markiert man die für den eigenen Block benötigten Blöcke. Im Menü kann man »eigenen Block erstellen« wählen. Es sollte sich wie oben in Abbildung gezeigt, ein Fenster öffnen, in dem der Name des eigenen Blocks, eine Beschreibung und ein Blocksymbol festgelegt werden können. [5]

### 3.2.2 Programmierung verschiedener Sensoren

#### Der Berührungssensor

Häufig wird der Berührungssensor dazu verwendet, um ein Programm durch Drücken der Taste zu steuern oder um zu erkennen, ob der Roboter gegen irgendein Objekt gestoßen ist.

Der Warte-, Schleifen-, und der Schalterblock können anhand der Eingaben von diesem Sensor bestimmen, ob die Taste gedrückt, ausgelassen oder angestoßen wurde. z.B. im Warte-Block gibt es zwei Möglichkeiten, um den Berührungssensor einzusetzen.

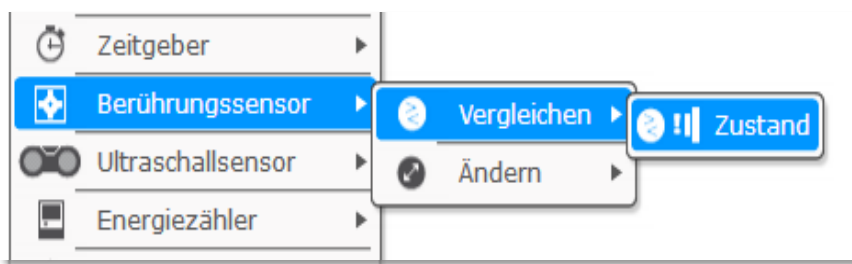


Abb. 40: Modus-Berührungssensor [10]

- Im Modus Vergleichen gibt man einen Zustand vor (Gedrückt, Ausgelassen oder Angestoßen.) Der Warte-Block hält das Programm dann an, bis der Berührungssensor den betreffenden Zustand einnimmt.
- Im Modus Ändern prüft der Warte-Block den Zustand des Sensors und wartet darauf, dass er sich von Gedrückt in Ausgelassen ändert oder umgekehrt. [11]

#### Der Farbsensor

Zur Programmierung des Farbsensors kann man in drei verschiedenen Modi einsetzen.

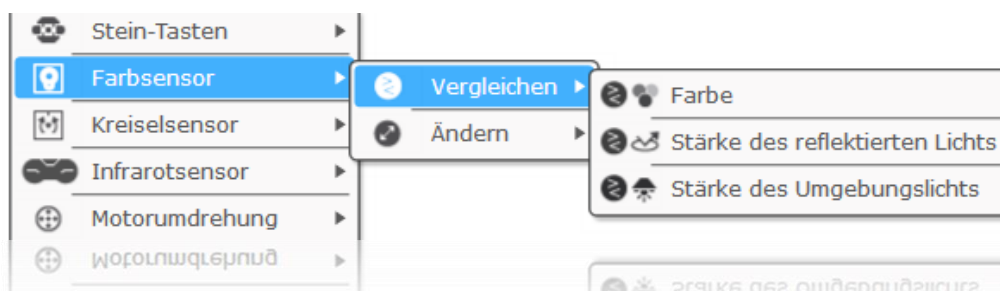
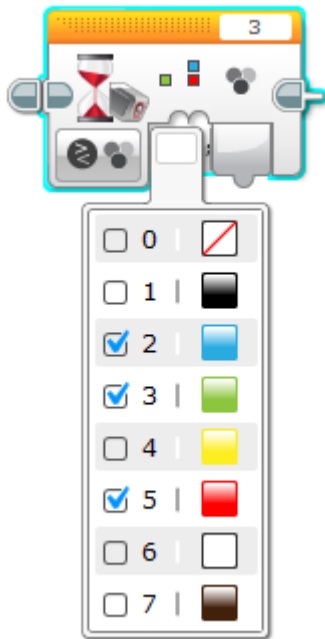


Abb. 41: Modus-Farbsensor [10]



- Der Farbmodus

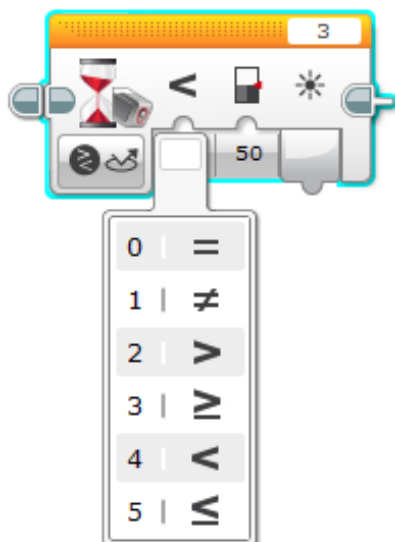


Im Farbmodus kann der Sensor die Farbe des Objekts erkennen, das vor ihm liegt, und dabei zwischen Schwarz, Blau, Grün, Gelb, Rot, Weiß und Braun unterscheiden.

Im Vergleichsmodus wartet der Block, bis der Sensor die in der Liste ausgewählte Farbe erkennt. Dieser Block wartet auf Grün, Blau oder Rot.

Abb. 42: Auswahl der Farben [10]

- Stärke des reflektierten Lichts



Im Modus Stärke des reflektierten Lichts schaltet der Sensor seine eigene rote LED ein und misst das von einem Objekt zurückgeworfene Licht. Die Messwerte reichen dabei von 0 bis 100, wobei 0 »sehr dunkel« und 100 »sehr hell«. Dieser Modus eignet sich dazu, den Roboter einer Linie folgen zu lassen. [11]

Abb. 43: Modus Stärke des reflektierten Lichts [10]

- Stärke des Umgebungslichts

Die dritte Funktionsweise des Farbsensors ist die Messung des Umgebungslichts, also des Lichts, das aus der Umgebung des Roboters einfällt. Die Abbildung zeigt den Warteblock im Modus **Farbsensor » Vergleichen » Stärke des Umgebungslichts**. Bei der Messung des Umgebungslichts erlaubt der Sensor sowohl einen Vergleichs- als auch einen Änderungsmodus mit den gleichen Konfigurationsoptionen wie bei der Messung des reflektierten Lichts. [11]



Abb. 44: Modus Stärke des Umgebungslichts [10]

## Der Kreisel sensor

Der Kreisel sensor teilt dem Programm mit, wie schnell er sich in Richtung einer der beiden Pfeile dreht, die auf seinem Gehäuse abgebildet sind. Da der Sensor weiß, wie schnell er sich bewegt- also wie viele Grad er pro Sekunde zurücklegt-, kann er auch bestimmen, um welchen Winkel er sich gedreht hat. [11] Für die Programmierung gibt es drei verschiedene Modi:

- Der Ratenmodus

Im Modus Rate misst der Kreisel sensor die Drehgeschwindigkeit in Grad pro Sekunde. Mit dem Sensor kann man Änderungen bei Drehbewegungen erkennen.

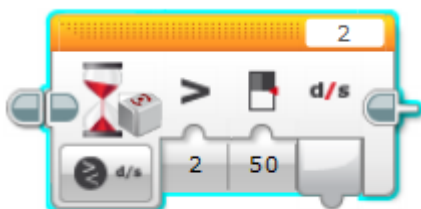


Abb. 45: Der Ratenmodus [10]

In Abbildung sieht man den Warteblock im Modus Kreisel sensor » Vergleichen » Rate. Das heißt, dass die vom Kreisel sensor gemessene Drehgeschwindigkeit größer als 50d/s ist. Erst danach werden die weiteren Programmblöcke abgearbeitet. [11]

- Der Winkelmodus

Im Modus Winkel misst der Sensor, um wie viel Grad sich der Roboter gedreht hat, seit der Sensor zurückgesetzt worden ist.



Abb. 46: Der Winkelmodus [10]

In Abbildung sieht man den Warte-Block im Modus Kreiselsensor » vergleichen » Winkel. Das heißt, dass der Roboter sich um +/- 10 Grad drehen wird.

- Den Winkel zurücksetzen

Zu Beginn des Programms setzt der Kreiselsensor den Winkelwert auf 0. Um ihn selbst zurücksetzen, während das Programm läuft, kann man einen Kreiselsensorblock im Modus Zurücksetzen verwenden (siehe Abbildung). Diesen Block findet man in der Palette Sensoren. [11]



Abb. 47: Den Winkel zurücksetzen [10]

## Der Ultraschallsensor



Abb. 48: Modus-Ultraschallsensor [10]

- Der Modus Distanz in Zentimetern / Zoll

Der Ultraschallsensor wird gewöhnlich im Modus Distanz in Zentimetern verwendet. Der Modus Distanz in Zoll funktioniert auf die gleiche Weise, benutzt aber eine andere Maßeinheit.



Abb. 49: Der Modus Distanz [10]

In Abbildung 49 sieht man den Warte-Block im Modus Ultraschallsensor » Vergleichen » Distanz in Zentimetern. Das heißt, dass der vom Ultraschallsensor gemessene Abstand zu einem Objekt kleiner als 5cm ist. Erst danach werden die weiteren Programmblöcke abgearbeitet. [11]

- Der Modus Anwesenheit / Wahrnehmen

Im Modus Anwesenheit / Wahrnehmen kann der Ultraschallsensor die Gegenwart eines anderen Ultraschallsensors erkennen, was praktisch ist für Spiele oder Wettkämpfe mit mehreren Robotern. Die einzige Einstellung, die man in diesem Modus vornehmen kann, ist der Anschluss, mit dem der Sensor verbunden ist (siehe Abbildung). Der Warte-Block wartet, bis ein anderer Ultraschallsensor erkannt wird. [11]



Die einzige Einstellung, die man in diesem Modus vornehmen kann, ist der Anschluss, mit dem der Sensor verbunden ist (siehe Abbildung). Der Warte-Block wartet, bis ein anderer Ultraschallsensor erkannt wird. [11]

Abb. 50: Der Modus Anwesenheit/ Wahrnehmen [10]

### 3.2.3 Problemstellungen und Lösungsansätzen

#### Kalibrierung des Farbsensors

Im oben stehenden Abschnitt der Programmierung des Farbsensors wird erläutert, dass die Messwerte des reflektierten Lichts des Farbsensors zwischen 0 und 100 liegen. Aber das heißt nicht, dass ein Farbsensor im schwarzen Bereich einen Messwert von 0 oder auf weißen Bereich ein Messwert von 100 bekommt. Um exakte Messwerte zu erhalten, muss der Farbsensor zuerst kalibriert werden.

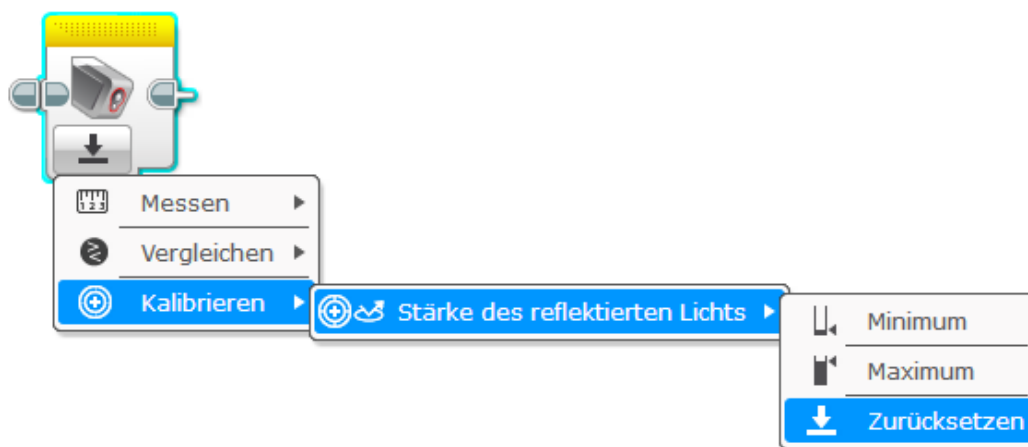


Abb. 51: Farbsensor-Block im Modus Kalibrieren [10]

Die Abbildung zeigt den Farbsensor-Block im Modus **Kalibrieren » Stärke des reflektierten Lichts » Zurücksetzen**.

Der detaillierte Programmablauf der Kalibrierung ist dem beigefügten Video zu entnehmen.

## Geradeausfahrt des Roboters

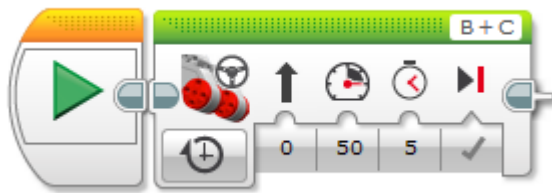


Abb. 52: Standardsteuerung-Block [10]

Wenn der Roboter wie in der oben gezeigten Abbildung programmiert wird, fährt er theoretisch 5 Sekunden mit der Leistung 50 ohne zu lenken. Aber im praktischen Fall fährt der Roboter meistens nicht geradeaus, sondern weicht davon unabsehbar ab. Mit dem Kreiselsensor lässt sich dieses Problem lösen.

Mit dem Kreiselsensor kann der Roboter die Drehbewegungen erkennen. Wenn er nicht geradeaus fährt, misst der Sensor die Änderung des Winkels. Diese Messwerte werden mit Hilfe des Mathe-Blocks umgewandelt und danach im Variablen-Block gespeichert. Die Übertragung der Messwerte erfolgt über die jeweilige Datenleitung. Die Leistung des Motors ändert sich durch die Änderung des Messwerts. Deshalb wird die Richtung des Roboters automatisch korrigiert.

Für den detaillierten Programmablauf dieser Funktion kann das beigefügte Video angesehen werden.

## Linie Folgen

In den FLL-Wettbewerb gibt es jedes Jahr verschiedene Themen und ein entsprechendes Spielfeld, das durch farblich gekennzeichnete Linien in viele Arbeitsbereiche eingeteilt wurde. Aufgrund dessen ist es sehr wichtig, dass ein Roboter den richtigen Linien folgen kann.

In diesem Zusammenhang werden zwei Lösungsvarianten vorgestellt:

### 1. Gesteuert von einem Farbsensor

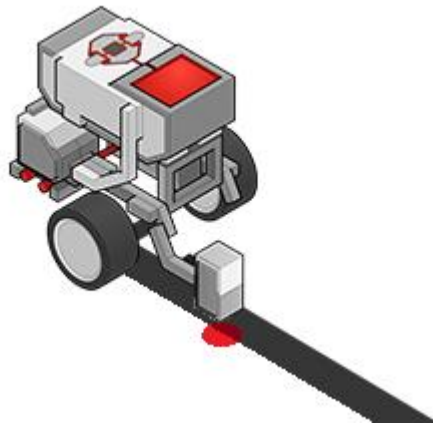


Abb. 53: Linie Folgen mit einem Farbsensor [7]

Im Modus Stärke des reflektierten Lichts gibt der Messwert des Farbsensors an, wie viel Licht von einem kleinen kreisförmigen Bereich unterhalb des Sensors zurückgeworfen wird. Befindet sich der Sensor über dem weißen Untergrund, ergibt sich ein hoher Messwert. Wenn der Sensor dagegen komplett über der dunklen Linie steht, wird nur wenig Licht reflektiert, sodass ein kleiner Wert gemessen wird. Während der Roboter vorwärts fährt, liest er anhand der Messwerte des Farbsensors ab, wo er sich relativ am Rand der Linie befindet.

In der folgenden Abbildung zeigen die Farbsensor-Messwerte in verschiedenen Stellungen.

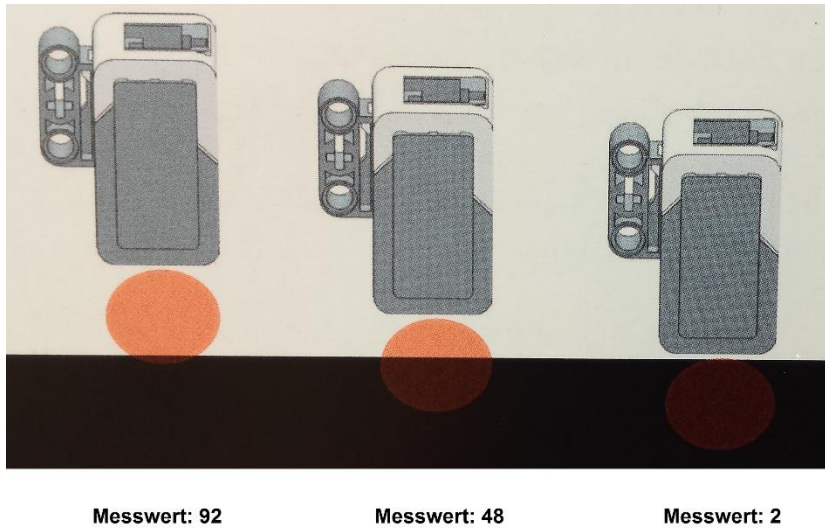


Abb. 54: Farbsensor-Messwerte in verschiedenen Stellungen [12]

## 2. Gesteuert von zwei Farbsensoren

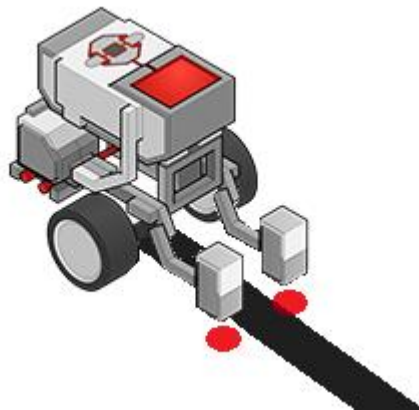


Abb. 55: Linie Folgen mit zwei Farbsensoren [7]

Wenn der Roboter zu schnell fährt und die Linie scharfe Ecken aufweist, können Probleme entstehen. Deswegen wurde die Variante, die der Roboter von zwei Farbsensoren gesteuert wird, entwickelt.

Der detaillierte Programmablauf der beiden Varianten ist dem beigefügten Video zu entnehmen.



## 4. Anwendungsfälle

### 4.1 Positionierung von Huhn und Methangas

#### 4.1.1 Aufgabenbeschreibung



Abb. 56: Die zwei gelben Schlaufen bzw. die Methanbehälter [13]

Sammelt Methan von der Deponie



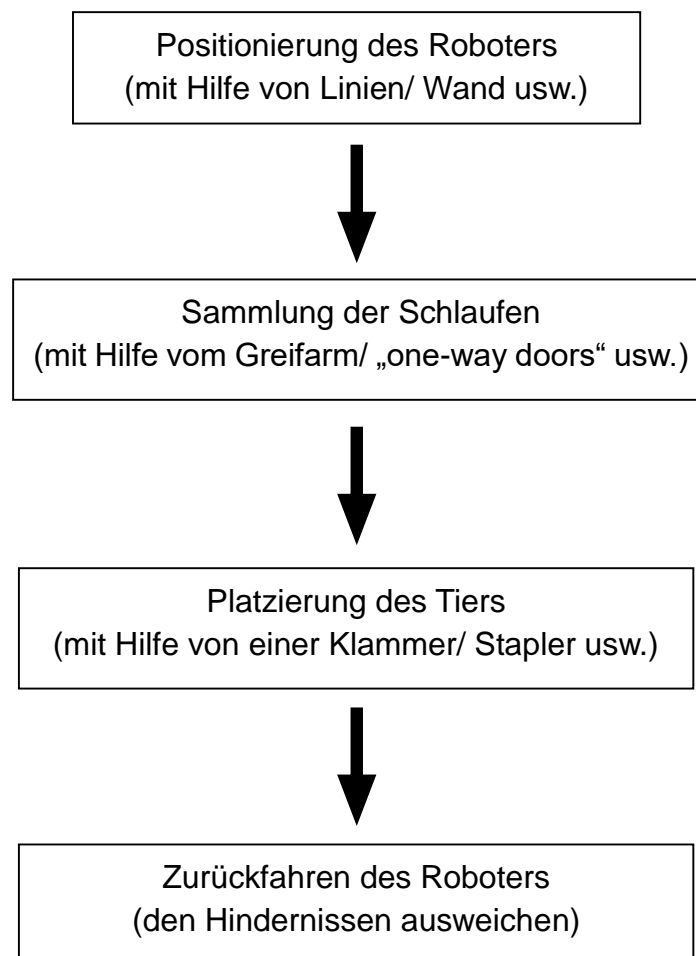
Abb. 57: Das Huhn im kreisförmigen Bereich [13]

Transport des Tieres an seinen Lieblingsplatz, das Huhn ist komplett im Kreis zu positionieren.

### 4.1.2 Erarbeitung einer Lösungsvariante

Entwickle eine Lösungsvariante für die Aufgabe „Positionierung des Huhns und Sammlung des Methangases“ mit ihren aus LEGO Teilen gebauten und programmierten Robotern!

Analyse der Aufgabe/ Lösungsstrategie



### 4.1.3 Beispiele für Lösungsvarianten

In den Anlagen sind die Tutorial-Videos für die Lösung der Aufgabe zu finden.

## 4.2 Positionierung des Methangases und Transport

### 4.2.1 Aufgabenbeschreibung



Abb. 58: Kraftwerk der Fabrik & Motorraum des LKWs

Sammelt Methan von der Deponie und nutzt es, um den LKW und die Fabrik zu betreiben.

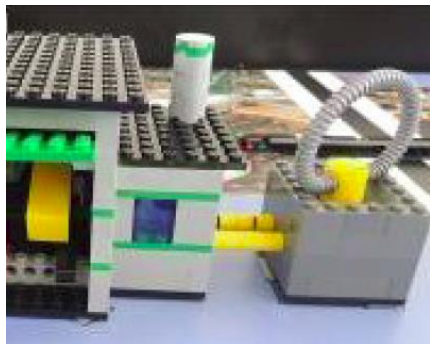


Abb. 60: Methan im Kraftwerk der Fabrik [13]



Abb. 59: Methan im Motorraum [13]

Endzustand:

Methan ist im Motorraum des LKWs und im Kraftwerk der Fabrik.

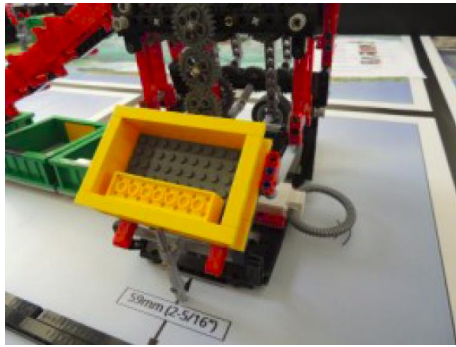


Abb. 62: Die gelbe Kiste in die Halterung auf dem Sortierer [13]



Abb. 61: Die gelbe Kiste auf den LKW [13]

Lade die gelbe Kiste auf den LKW, transportiere diese ostwärts!



Abb. 63: Abladung der gelben Kiste [13]

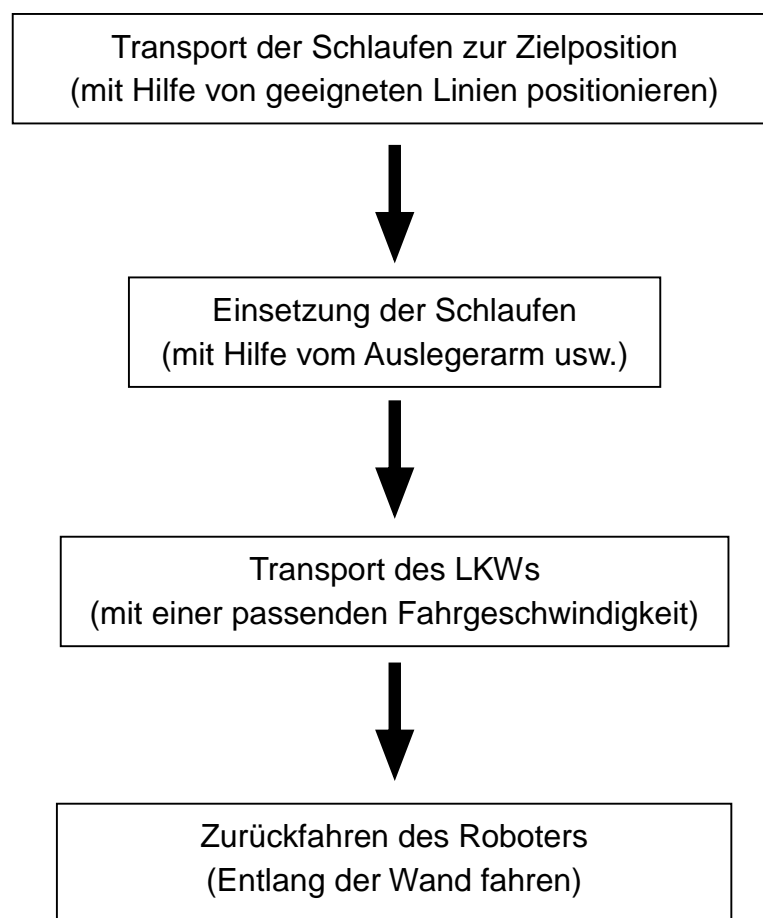
Endzustand:

Die Kiste befindet sich komplett östlich der LKW-Führungsschiene.

## 4.2.2 Erarbeitung einer Lösungsvariante

Entwickle eine Lösungsvariante für die Aufgabe „Positionierung des Methangas und Transport“ mit ihren aus LEGO Teilen gebauten und programmierten Robotern!

Analyse der Aufgabe/ Lösungsstrategie



## 4.2.3 Beispiele für Lösungsvarianten

In den Anlagen sind die Tutorial-Videos für die Lösung der Aufgabe zu finden.

## 4.3 Kraftübertragung und Sortierung

### 4.3.1 Aufgabenbeschreibung

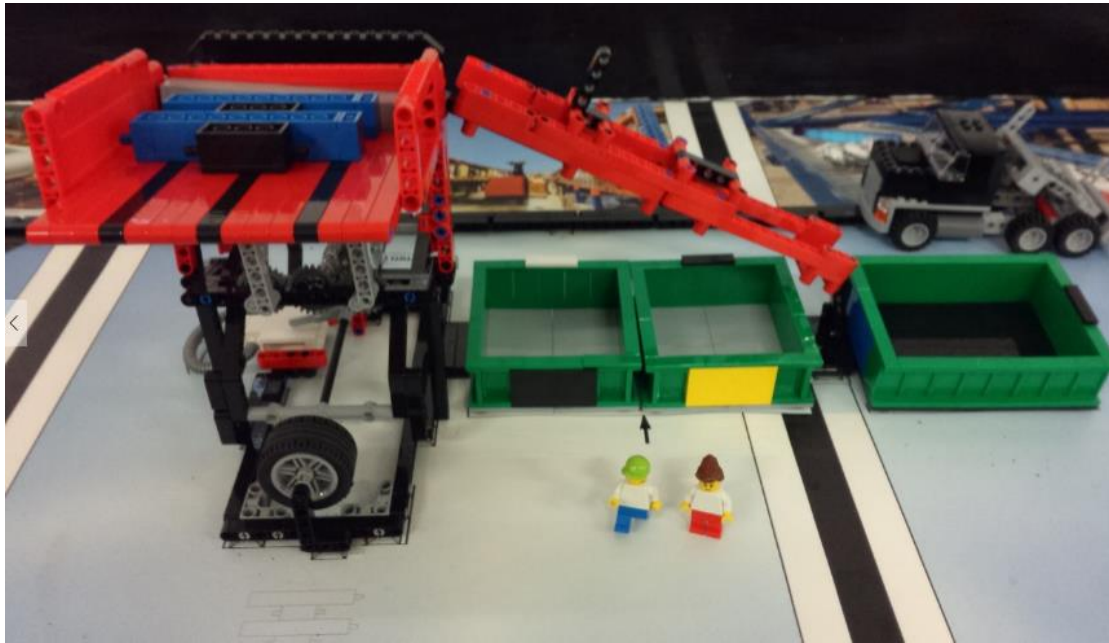


Abb. 64: Sortieranlage

Der Roboter trägt die schwarzen, gelben und blauen Balken zu der Sortieranlage.



Abb. 65: Sortier-Bereich

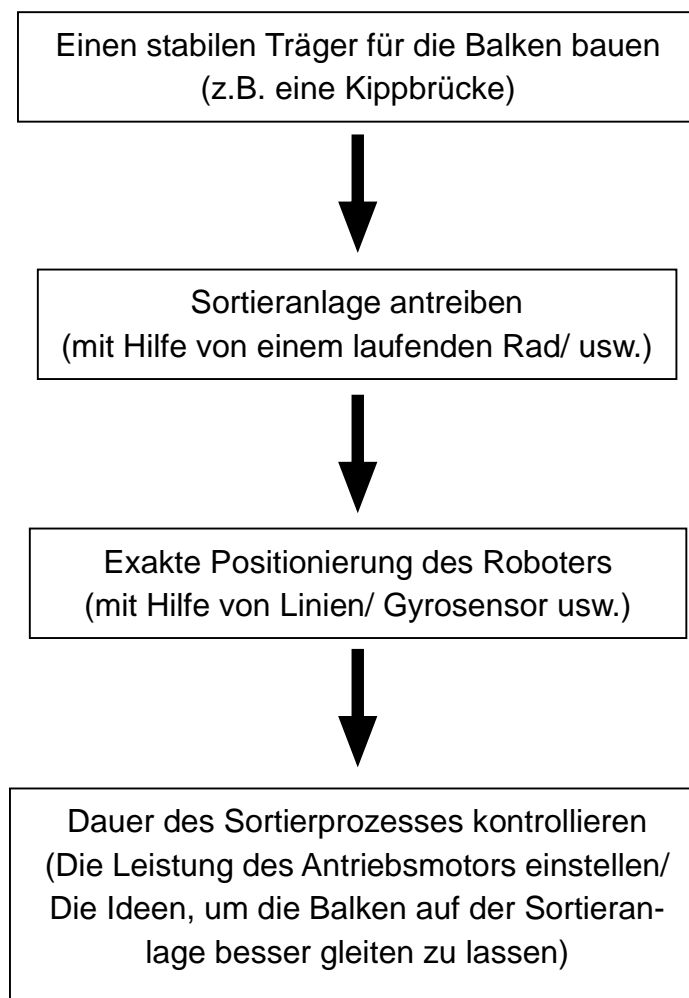
Endzustand:

Schwarze, gelbe und blaue Balken sind in den passenden Kisten.

### 4.3.2 Erarbeitung einer Lösungsvariante

Entwickle eine Lösungsvariante für die Aufgabe „Kraftübertragung und Sortierung“ mit ihren aus LEGO Teilen gebauten und programmierten Robotern!

Analyse der Aufgabe/ Lösungsstrategie



### 4.3.3 Beispiele für Lösungsvarianten

In den Anlagen sind die Tutorial-Videos für die Lösung der Aufgabe zu finden.

## 5. Zusammenfassung

Die vorliegende Arbeit beschreibt ein multimediales Tutorial zum Bau und zur Programmierung eines modularen Mini-Roboters für drei verschiedene Anwendungsfälle.

Im ersten Teil der Arbeit wurden die LEGO MINDSTORMS Education EV3-Grundlagen vorgestellt. Hauptsächlich wurden die Funktionen der Bauelemente im EV3-Set beschrieben. Außerdem wurden die unterscheidenden Merkmale der EV3-Programmierungstechnik und der Konstruktionen Technik dargestellt.

Ein weiterer Teil der Arbeit beschäftigt sich mit einem Konzept für das Tutorial, das wichtige Bauprinzipien und die Programmierung des Roboters präsentiert. Speziell wurden die Lösungsansätze für die Problemstellungen, mit denen die Schüler und Schülerinnen konfrontiert wurden, entwickelt. Die Problemstellungen umfassen das Verhalten des Roboters, z.B. Linie-Verfolgung, Geradeausfahrt des Roboters und die Ideen für den FLL-Wettbewerb, z.B. ein passendes Lenksystem und effizienter Umbau des Roboters.

Anschließend wurden drei Anwendungsfälle im Rahmen des FLL-Wettbewerbs entwickelt. Damit wurden drei Aufgaben gewählt. Die Schüler und Schülerinnen können mit ihren aus LEGO Teilen gebauten und programmierten Robotern die Aufgaben lösen. Laut Analyse der Aufgaben werden die Schüler und Schülerinnen zur systematischen und analytischen Arbeit motiviert.

Zum Schluss wurden 9 Tutorial-Videos eingefügt, die detaillierte Lösungsansätze für alle Problemstellungen beinhalten. In den Anlagen sind die Tutorial-Videos für die Lösung der Aufgabe zu finden.



## 6. Danksagung

Hiermit möchte ich mich bei allen herzlich bedanken, die mich während der Anfertigung meiner Bachelorarbeit unterstützt haben.

Speziell gilt mein Dank...

- Prof. Dr. -Ing. Heike. Mrech für die Bereitstellung des Themas sowie für die angenehme Betreuung
- Dipl.-Phys. Egon. Hanauska für die ständige Unterstützung, die vielen hilfreichen Tipps und die geduldige Beantwortung meiner Fragen
- Herr Aleksandar Turuntas für die Bearbeitung des Tutorial-Videos
- meinen Eltern und meiner Deutschlehrerin für die immerwährende Unterstützung in jeglicher Form während meines Studiums

Merseburg, den 14. April. 2016

Tao Zou

## **7. Selbständigkeitserklärung**

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Merseburg, den 14. April. 2016

Unterschrift: \_\_\_\_\_

## 8. Literaturverzeichnis

- [1] „LEGO education,“ [Online]. Available: <https://education.lego.com/de-de/lesi/middle-school/mindstorms-education-ev3>. [Zugriff am 3 2016].
- [2] „Fachlehrplan Sekundarschule,“ Kultusministerium, Bad Dürrenberg, SACHSEN-ANHALT, 2016.
- [3] „LEGO-EV3 Bedienungsanleitung“.
- [4] „LEGO mindstorms EV3,“ [Online]. Available: <http://www.lego.com/de-de/mindstorms/products/mindstorms-ev3-31313>. [Zugriff am 3 2016].
- [5] M. P. Scholz, Das EV3 Roboter Universum, Medienhaus Plump GmbH, Rheinbreitbach.
- [6] „EV3,NXT&RCX Mindstorms Wissen,“ [Online]. Available: <http://ev3-wissen.engeln.info/doku.php?id=ev3:gyrosensor>. [Zugriff am 3 2016].
- [7] „Introduction to programming,“ [Online]. Available: <http://curriculum.cs2n.org/ev3/lesson/5-1Gyro4.html>. [Zugriff am 3 2016].
- [8] „Mit oder ohne Noppen,“ [Online]. Available: [https://www.dpunkt.de/leseproben/4315/4\\_Mit%20oder%20ohne%20Noppen.pdf](https://www.dpunkt.de/leseproben/4315/4_Mit%20oder%20ohne%20Noppen.pdf). [Zugriff am 2 2016].
- [9] D. Benedettelli, Das LEGO-Mindstorms-EV3-Labor, Heidelberg: dpunkt.verlag GmbH, 2014.
- [10] „LEGO Mindstroms Education EV3 Programm Software“.
- [11] T. Griffin, Programmieren Lernen mit EV3, Heidelberg: dpunkt.verlag GmbH, 2015.
- [12] „Hilfe-Bereich von LEGO® MINDSTORMS® EV3.,“ [Online]. Available: [http://localhost:58401/localizedMapping\\_B90BDB05-F70E-4B0B-8CEA-031DCF197215/de/editor/](http://localhost:58401/localizedMapping_B90BDB05-F70E-4B0B-8CEA-031DCF197215/de/editor/). [Zugriff am 3 2016].
- [13] „FLL"TRASH TREK"-REGELN & ROBOT GAME,“ Hands on Technologz, 2016.

## 9. Abbildungsverzeichnis

Abb. 1: LEGO MINDSTORMS EV3 Education-Set [3] .....	6
Abb. 2: EV3-Stein [3] .....	7
Abb. 3: EV3, großer Motor [3] .....	8
Abb. 4: EV3, mittlerer Motor [3].....	8
Abb. 5: Berührungssensor [3] .....	9
Abb. 6: Kreiselsensor [3].....	9
Abb. 7: Farbsensor [3] .....	9
Abb. 8: Ultraschallsensor [3].....	9
Abb. 9: EV3-Sets, Balken .....	10
Abb. 10: EV3-Sets, Verbindungsstücke .....	10
Abb. 11: EV3-Sets, Zahnräder .....	11
Abb. 12: EV3-Sets, Achsen.....	11
Abb. 13: EV3-Berührungssensor [3] .....	12
Abb. 14: Die Reflexion an einem Hindernis [5] .....	13
Abb. 15: Die Reflexion an einem Hindernis [5] .....	13
Abb. 16: Farbmodus [3] .....	14
Abb. 17: RGB-Verfahren .....	14
Abb. 18: Farbsensor [5] .....	14
Abb. 19: Roboter mit Kreiselsensor [7] .....	15
Abb. 20: Die noppenlose Balken [8].....	17
Abb. 21: die dreieckige Struktur .....	22
Abb. 22: die parallele Struktur .....	22
Abb. 23: Balken Verlängerung [9] .....	23
Abb. 24: Zahnräder .....	24
Abb. 25: Rechtwinklige Verzahnung .....	25
Abb. 26: Schneckengetriebe .....	25
Abb. 27: Drei typische Arten von Zahnradkombinationen .....	26
Abb. 28: Radien der verschiedenen Zahnräder [9] .....	26
Abb. 29: Ein Kugelrad als Schwenkrad.....	27
Abb. 30: Ein ähnliches Lenksystem wie am Auto.....	28
Abb. 31: Zuordnung der EV3-Programm-Blöcke [10].....	30
Abb. 32: Warte-Block [10] .....	31
Abb. 33: Warte-Block im Modus Ultraschallsensor [10] .....	31
Abb. 34: Schleifen-Block im Modus Unbegrenzt [10] .....	32
Abb. 35: Schalter-Block [10] .....	33
Abb. 36: Variablen-Block [10].....	34
Abb. 37: Mathe-Block im unterschiedlichen Modus [10] .....	35
Abb. 38: Der mathematische Operationsmodi [12] .....	35
Abb. 39: Eigene-Block [10] .....	36

Abb. 40: Modus-Berührungssensor [10] .....	37
Abb. 41: Modus-Farbsensor [10].....	37
Abb. 42: Auswahl der Farben [10].....	38
Abb. 43: Modus Stärke des reflektierten Lichts [10].....	38
Abb. 44: Modus Stärke des Umgebungslichts [10] .....	39
Abb. 45: Der Ratenmodus [10] .....	39
Abb. 46: Der Winkelmodus [10] .....	40
Abb. 47: Den Winkel zurücksetzen [10] .....	40
Abb. 48: Modus-Ultraschallsensor [10] .....	41
Abb. 49: Der Modus Distanz [10] .....	41
Abb. 50: Der Modus Anwesenheit/ Wahrnehmen [10] .....	41
Abb. 51: Farbsensor-Block im Modus Kalibrieren [10].....	42
Abb. 52: Standardsteuerung-Block [10] .....	43
Abb. 53: Linie Folgen mit einem Farbsensor [7] .....	44
Abb. 54: Farbsensor-Messwerte in verschiedenen Stellungen [12] .....	45
Abb. 55: Linie Folgen mit zwei Farbsensoren [7] .....	45
Abb. 56: Die zwei gelben Schlaufen bzw. die Methanbehälter [13].....	46
Abb. 57: Das Huhn im kreisförmigen Bereich [13] .....	46
Abb. 58: Kraftwerk der Fabrik & Motorraum des LKWs .....	48
Abb. 59: Methan im Motorraum [13].....	48
Abb. 60: Methan im Kraftwerk der Fabrik [13].....	48
Abb. 61: Die gelbe Kiste auf den LKW [13].....	49
Abb. 62: Die gelbe Kiste in die Halterung auf dem Sortierer [13] .....	49
Abb. 63: Abladung der gelben Kiste [13].....	49
Abb. 64: Sortieranlage .....	51
Abb. 65: Sortier-Bereich.....	51

# 10. Anlagenverzeichnis

Tutorial-Videos:

01: Kalibrierung des Farbsensors

02: Linie Folgen

03: Geradeausfahrt des Roboters

04: Positionierung von Huhn und Methangas

05: Das Programm für Positionierung von Huhn und Methangas

06: Positionierung des Methangases und Transport

07: Das Programm für Positionierung des Methangases und Transport

08: Kraftübertragung und Sortierung

09: Das Programm für Kraftübertragung und Sortierung