

# Fahrzeugassistenzsystem in natürlichen Umgebungen mittels Bilderkennung

Studienarbeit

des Studiengangs

Informationstechnik

von

Sascha Moser

20. Dezember 2014

---

Dozent:	Hans-Jörg Haubner
E-Mail:	haubner@dh-karlsruhe.de
Bearbeitungszeitraum:	29.09.14 - 31.03.14
Klausurtermin:	19.12.2014
Autor:	Sascha Moser
Kurs:	TINF12B3
Ausbildungsfirma:	Harman/Becker Automotive Systems GmbH
Studiengangsleiter:	Jürgen Vollmer

---

---

# Inhaltsverzeichnis

---

<b>1</b>	<b>Einleitung</b>	<b>5</b>
1.1	Aufgabenstellung . . . . .	5
1.2	Projektplanung . . . . .	5
1.3	Verwendete Software . . . . .	6
<b>2</b>	<b>Grundlagen</b>	<b>9</b>
2.1	Bedeutung von Roboter und Assistenzsysteme . . . . .	9
2.2	Antriebsart des Versuchsroboters . . . . .	9
2.3	Assistenzsysteme heutiger Fahrzeuge . . . . .	10
2.3.1	ABS - Antiblockiersystem . . . . .	11
2.3.2	PDC - Park Distanz Kontrolle . . . . .	11
2.3.3	ESC-Electronic Stability Control . . . . .	12
2.3.4	Automatisches einparken . . . . .	12
2.3.5	Anwendung aktueller Assistenzsysteme . . . . .	13
2.4	Automatisch geführte Fahrzeuge . . . . .	14
<b>3</b>	<b>Erste Experimente mit dem EV3</b>	<b>15</b>
3.1	Aufbau des Roboters . . . . .	15
3.2	Aufbau und Test des Ultraschallsensors . . . . .	15
3.3	Test des mitgelieferten Farbsensors . . . . .	17
<b>4</b>	<b>Erste Experimente mit Lejos und der externen Kamera</b>	<b>20</b>
4.1	Installation von Lejos in Eclipse . . . . .	20
4.2	Beispielprogramm mit Lejos . . . . .	20
4.3	Installation Vision subsystem V4 . . . . .	20
4.4	Vision subsystem V4 — Kammeraprogramm . . . . .	21
<b>5</b>	<b>Umsetzung des Projekts</b>	<b>22</b>
5.1	Festlegung der Werte . . . . .	22
5.2	Einsatz Gyrosensor . . . . .	22
5.3	Einsatz eigener Blöcke . . . . .	23

5.4 Das Programm . . . . .	23
<b>Abbildungsverzeichnis</b>	<b>24</b>
<b>Tabellenverzeichnis</b>	<b>25</b>
<b>Listings</b>	<b>26</b>

# KAPITEL 1

---

## Einleitung

---

### 1.1 Aufgabenstellung

Aufgabe dieser Studienarbeit ist es, mit Hilfe von Lego Mindstorms, einen Roboter aufzubauen. Dieser Roboter soll sich in einer ihm unbekannten Umgebung fortbewegen. Durch farbliche Objekte und Markierungen innerhalb dieser Umgebung soll er sich entsprechend derer Bedeutung verhalten.

Des weiteren soll der Roboter soll der Roboter Hindernissen, wie Wänden, ausweichen können.

Die Erkennung der farblichen Marikierungen soll mit Hilfe des mitgelieferten Farbsensors erfolgen. Die Wanderkennung soll mit Hilfe des Ultraschallsensors realisiert werden. Ein entsprechender Parkur, mit Markierungen und Hindernissen, ist der Abbildung [1.1 auf der nächsten Seite](#) zu entnehmen. In der Abbildung [1.2 auf Seite 7](#) ist die entsprechende Legende, in der Symbole und Flächen näher erleutert.

### 1.2 Projektplanung

Dauer	Art der Tätigkeit	Meilenstein
1Woche	Einarbeitung	1. Meilenstein

**Tabelle 1.1:** Übersicht der Projektplanung

**Dies ist die Tabelle der Projektplanung**

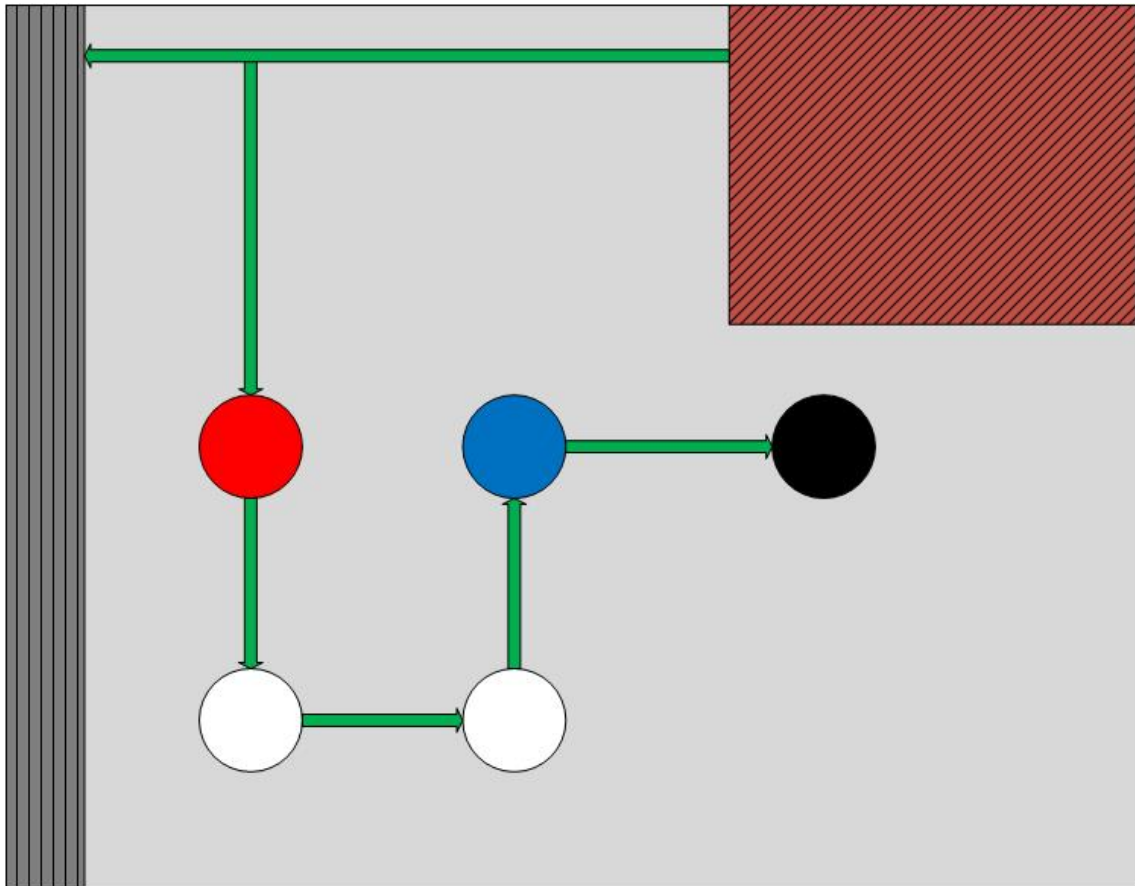


Abbildung 1.1: Übersicht des zu befahrenden Parkurs

### 1.3 Verwendete Software

In diesem Projekt werden verschiedene Programmierumgebungen benutzt. Eine Programmierumgebung ist die Lego-eigene grafische Programmierumgebung. Diese Programmierumgebung gibt es in verschiedenen Ausführungen, genutzt wird die "LEGO MINDSTORMS Education EV3" Version. Diese IDE zeigt dem Nutzer unter anderem die aktuellen Werte der genutzten Sensoren an so wie deren Steckplatz am "Brick" **Glossareintrag**. Diese Programmierumgebung ist für den Einstieg und erste kleinere Programme geeignet, jedoch geht die Übersicht bei komplizierteren Programmen verloren. Aus diesem Grund wird in dieser Studienarbeit diese Umgebung dazu genutzt den Brick und die Funktionsweise der Sensoren näher kennen zu lernen. Dieses Wissen wird später dann auf die Java Programmierung transferiert.

Des Weiteren wird die Java IDE "Eclipse" mit der Erweiterung "Lejos" genutzt. Auf die Installation der Erweiterung wird im Anhang näher eingegangen. Diese Programmierumgebung ...

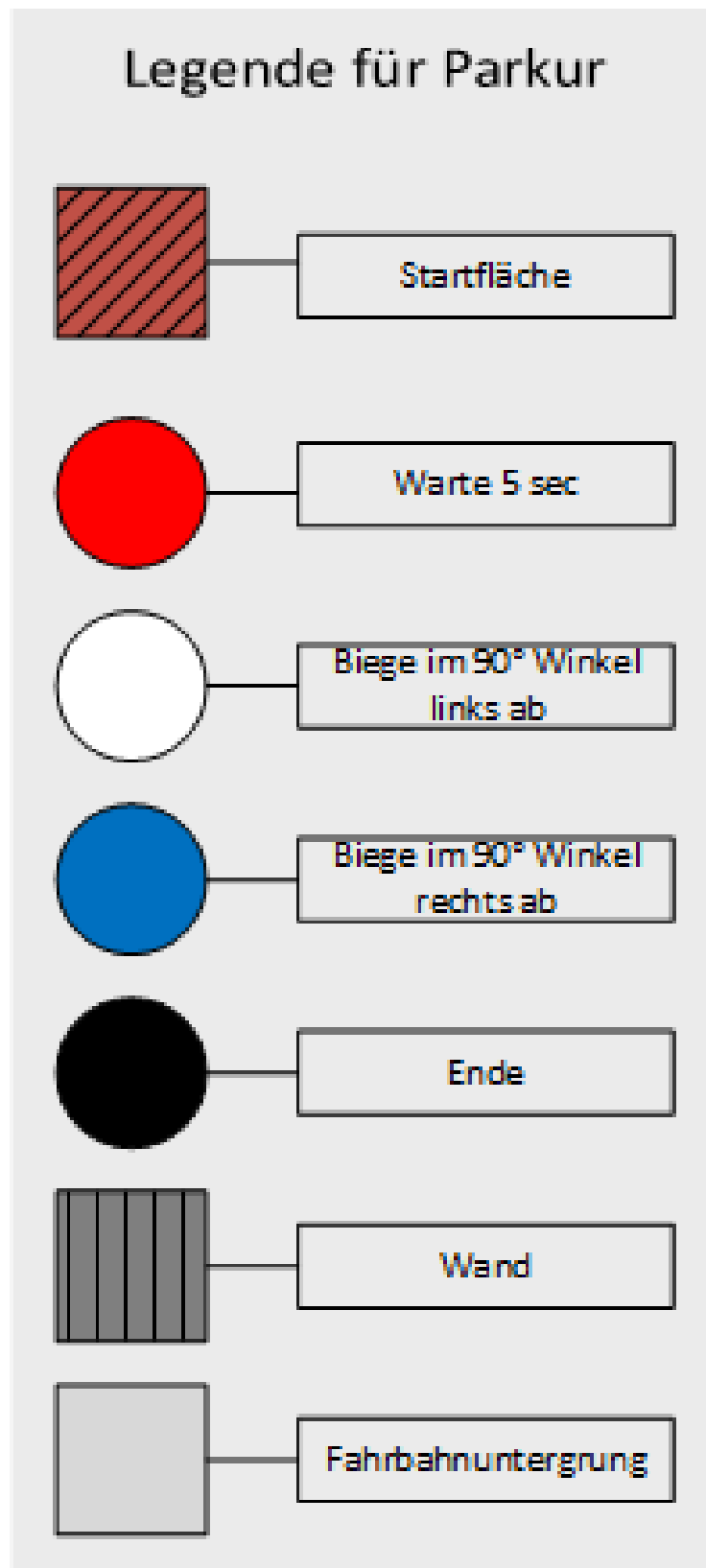


Abbildung 1.2: Legende zu Bild 1.1 auf der vorherigen Seite

Damit bei einem etwaigen Datenverlust der Verlust gering gehalten wird, wird die Versionsverwaltung GITHUB benutzt. Darin werden die Programme und auch dieses Dokument verwaltet. Bei einem Datenverlust kann somit auf eine vorhergehende Version des Programms oder dieses Dokuments zurückgegriffen werden. Das verwendete Repository ist öffentlich und kann unter <https://github.com/saschlick/Studienarbeit/> eingesehen werden.

Ebenso werden die gesamten Projektdaten in meiner DropBox gespeichert. Dies hat mehrere Vorteile. Zum einen kann ich an von jedem Computer mit Internetzugang darauf zugreifen und zum anderen ist es ebenfalls eine Absicherung gegen Datenverlust.

# KAPITEL 2

---

## Grundlagen

---

In diesem Kapitel werden die Grundlagen von Robotern behandelt. Es wird der zu Grunde liegende Aufbau des Roboters, im Bezug auf dessen Fortbewegung, vorgestellt. Ebenso werden heutige Fahrassistenzsysteme wie beispielsweise ABS vorgestellt. Zu diesem Kapitel gehört auch eine Übersicht über die Verwendung aktueller Assistenzsysteme.

### 2.1 Bedeutung von Roboter und Assistenzsysteme

- Anwendungsbeispiele

Wie wichtig sind sie wirklich ?

Weshalb eigener Industriezweig?

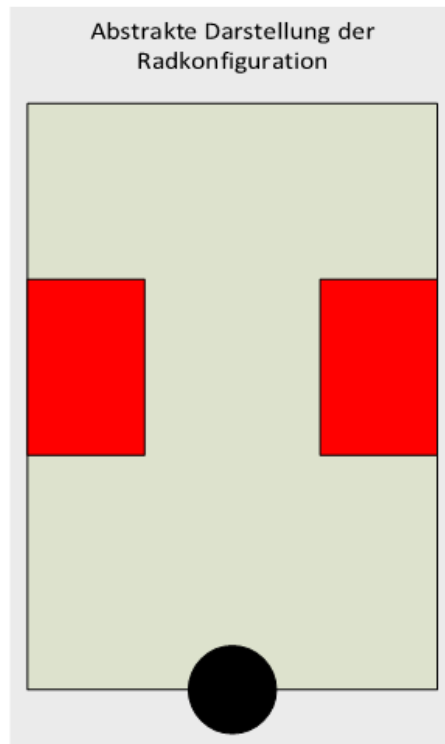
Kurze nennung anderer Roboter (Robotino, Toyota Roboter etc.

### 2.2 Antriebsart des Versuchsroboters

Der Versuchsroboter wird mit zwei unabhängigen Motoren ausgestattet. Diese Motoren treiben jeweils eins der beiden Haupträder an. Die Haupträder sind in der Abbildung [2.1 auf der nächsten Seite](#) rot dargestellt. Zusätzlich erhält der Roboter ein Stützrad, welches durch eine freilaufende Kugel realisiert wurde. In der Abbildung [2.1](#) wird das Stützrad schwarz dargestellt. Es stützt den Roboter soweit ab, dass dieser nicht nach hinten kippen kann und somit Bauteile auf dem Untergrund schleifen.

Der Antrieb in diesem Aufbau ist ein Differential Antrieb wie in Abbildung [2.1 auf der nächsten Seite](#) dargestellt. Mit diesem Antrieb sind sowohl Geradeausfahrten als auch Kurvenfahrten sowie das Drehen auf der Stelle möglich. Sein Hauptvorteil besteht in der einfachen Mechanik die Verbaut





**Abbildung 2.1:** Abstrakte Darstellung der Antriebsräder des Versuchsroboters

wird. Aufgrund seiner zwei unabhängig von einander steuerbaren Räder ist sein Einsatz für dieses Projekt optimal geeignet und sein eigentlicher Nachteil fällt somit nicht weiter auf.

Sein Nachteil ist, dass die Räder in Echtzeit geregelt werden müssen.

**Quelle : Strand, Marcus Vorlesungsfolien**

## 2.3 Assistenzsysteme heutiger Fahrzeuge

In der heutigen Zeit unterstützen die Fahrer einige Helfer, auch Assistenzsysteme genannt. Diese Systeme sollen dazu beitragen, den Straßenverkehr sicherer zu machen. Sie entlasten und unterstützen den Fahrer in jeglicher Fahrsituation. So helfen sie beim Einparken des Fahrzeugs oder verhindern in Gefahrensituationen den Verlust über die Kontrolle.

Im Laufe der Automobilära wurden Assistenzsysteme erfunden und Bestehende weiter entwickelt.

### 2.3.1 ABS - Antiblockiersystem

Ein Assistenzsystem, welches sich in der Automobilbranche flächendeckend durchgesetzt hat, ist das Antiblockiersystem. Es hilft dem Fahrer, das Fahrzeug sicher zum stehen zu bringen. Wie der Name schon sagt, versucht es durch gezieltes vermindern des Bremsdrucks ein blockieren der Räder zu verhindern. Aufgrund diesen Eingriffs wird der Bremsweg verringert und der Verschleiß an den Lauflächen wird vermindert. Durch dieses System wird die Lenkbarkeit und Spurtreue erhöht.

Dieses System ist inzwischen bei jedem großen Automobilhersteller in den Ausstattungslisten der Fahrzeuge. Meist ist es serienmäßig in die Fahrzeuge integriert oder kann für einen geringen Aufpreis nachgeordert werden.

Antiblockiersysteme werden nicht nur bei Autos eingesetzt. Es wird ebenfalls in Flugzeugen Zügen und Motorrädern eingesetzt. Den ersten Einsatz eines solchen Systems fand 1920 statt. Jedoch war dies kein Auto sondern ein Flugzeug.

### 2.3.2 PDC - Park Distanz Kontrolle

Dieses System unterstützt den Fahrer, mit Hilfe akustischen und visuellen Warnhinweisen, beim Einparken seines Fahrzeugs.

**Funktion** Mit Hilfe von Ultraschallsensoren, welche meist in den Heck-/Frontschürzen untergebracht sind, wird der Abstand zu Gegenständen in einer gewissen Distanz, meist ab 1-2 Meter, gewarnt.

Diese Warnungen erfolgten bei den ersten Systemen meist akustisch über verschiedene Warntöne. Danach wurden zusätzlich kleine LED-Lampen in verschiedenen Farben eingeführt. Diese leuchten je nach Distanz zum Gegenstand oder Fahrzeug in Grün- Orange/Gelb- Rot. Die nächste Stufe ist eine optische Darstellung des Fahrzeugs auf dem Bildschirm des Infotainmentsystems. Im Bildschirm wird dann der Abstand visuell dargestellt. Zusätzlich dazu, erhält der Fahrer akustische Hinweise.

Die neusten Systeme verfügen inzwischen noch über eine Kamera. Mit Hilfe der Kamera erhält der Fahrer beim einparken ein Livebild auf den Bildschirm seines Infotainmentsystems. In der Abbildung 2.2 auf der nächsten Seite wird die Darstellung eines solchen Systems im aktuellen 1er BMW gezeigt. Es zeigt ein Livebild der Rückfahrkamera so wie die aktuellen Abstände in der Headunit an. Bei anderen Herstellern ist es möglich, dass zusätzlich noch der bestmögliche Fahrweg visuell im Livebild dargestellt wird.

Dieses System und seine Weiterentwicklungen ist die Grundlage für automatische Einparksysteme.

**Quelle:** [http://www.bmw.com/\\_common/shared/newvehicles/1series/3door/2012/showroom/comfort/distance\\_control/distance\\_control\\_02.jpg](http://www.bmw.com/_common/shared/newvehicles/1series/3door/2012/showroom/comfort/distance_control/distance_control_02.jpg), Vorlesung Strand, Marcus



Abbildung 2.2: Darstellung des Abstandes in der Headuni am Beispiel BMW 1er

### 2.3.3 ESC-Electronic Stability Control

**Geschichtliches** ESC-Electronic Stability Control ist noch ein recht junges Assistenzsystem. Im Vergleich zum Antiblockiersystem, welches schon in den 20er Jahren des 20. Jahrhunderts entwickelt und bis heute immer weiter entwickelt wurde, führte Mercedes-Benz, mit Hilfe der Bosch AG, dieses Assistenzsystem erst 1995 in der damals neu aufgelegten S-Klasse in die Serienproduktion ein.

Der Name Elektronisches Stabilitätsprogramm (ESP) ist Eigentum der Firma Bosch. Bei anderen Herstellern gibt es ähnliche Systeme mit anderen Bezeichnungen. Bei BMW wird das System DSC-Dynamic Stability Control genannt. Bei den Herstellern von Jaguar und Mazda ist das entsprechende System ebenfalls das DSC.

Da diese unterschiedlichen Bezeichnungen zu Verwirrung führen kann, wird in Fachkreisen und im Anbieter-neutralen Markt meist der Begriff ESC-Electronic Stability Control oder Fahrdynamikregelung verwendet.

**Funktion** Dieses Assistenzsystem hilft dem Fahrer im Grenzbereich die Kontrolle seines Fahrzeuges zu behalten.

**Beschlüsse und Gesetze** Seit dem Beschluss des EU-Parlaments am 10. März 2009 müssen seit November 2011 alle in der EU zugelassenen Neuwagen und LKW ein entsprechendes System verwenden. Es gab eine Übergangsfrist bis Oktober 2014 für bereits zugelassene Fahrzeugtypen. Jedoch wurden Mitte 2014 immer noch einige Kleinwagen ohne ESC angeboten.

### 2.3.4 Automatisches einparken

### 2.3.5 Anwendung aktueller Assistenzsysteme

In den Fahrzeugen, die heute die Fabrikhallen verlassen finden sich inzwischen immer mehr Assistenzsysteme. Diese reichen vom ESP bis hin zum Spurhalteassistent oder dem Automatischen Einparksystemen. Jedoch ruhen sich die Hersteller auf diesen Systemen nicht aus und entwickeln immer umfangreichere Systeme für folgende Fahrzeuggenerationen.

**Probleme aktueller Systeme** Bei allen Erleichterungen und aller Unterstützung für den Fahrer sollte nicht vergessen werden, dass der Fahrer immer noch selbst am Steuer sitzt und für sein Fahrzeug verantwortlich ist. Bei Testfahrten eines Erbkönigs der Marke BMW kam es am 28. März diesen Jahres zu einem Unfall. Während der Testfahrt, auf der der Fahrer laut dem Hersteller „kamerabasierte“ Fahrerassistenzsysteme bei Dunkelheit testen sollte kam es in Stuttgart zu einem Unfall. In diesen Unfall war ein, mit Sondersignal fahrendes, Polizeiauto verwickelt, welches im Kreuzungsbereich durch den Zusammenstoß umkippte. Dieser Unfall zeigt, dass die heutigen Assistenzsysteme schon sehr weit entwickelt sind. Jedoch bedarf es immer noch einem Fahrer, der in Ausnahmesituationen eingreifen kann und einen solchen Unfall verhindert. Weshalb der Fahrer in dieser Situation nicht eingegriffen hat, oder ob es ein Problem mit der Steuerung gab ist nicht bekannt.

Wie dieser Unfall zeigt sind die Assistenzsysteme noch nicht soweit, dass sie alleine, ohne Fahrer, im Straßenverkehr bestehen können.

**Zukunftsweisende Projekte** Jedoch sind die Hersteller inzwischen soweit, dass sie Fahrzeuge im Testbetrieb haben, die ohne Fahrer ihre Runden auf den Rennstrecken dieser Welt drehen können. So stellte AUDI, im Rahmen des letzten DTM-Rennens der Saison 2014, ein Auto (Abbildung 2.3 auf der nächsten Seite) vor, welches ohne Fahrer eine Runde im Renntempo über den Hockenheimring fuhr. Die Daten für den Fahrtweg ermittelt es mithilfe von Kameras, die in und um das Fahrzeug verteilt sind. Unterstützt wird es durch weitere Sensoren wie GPS, Raddrehzahlsensor, Gierratensensor und andern Sensoren. Aus diesen Daten errechnet der Computer die Optimale Geschwindigkeit, Bremspunkte und Position auf der Strecke. Bei den Zuschauern kam diese Vorführung gut an. Das Fahrzeug wurde in der Zwischenzeit in mehreren Fernsehsendungen, unter anderem bei GRIP auf RTL II, vorgestellt. **Quelle Bild:** [http://audi-encounter.com/de/img/article-image-main/2015\\_1\\_Bobby\\_Car\\_Gallery\\_09.jpg\\_2130.jpg](http://audi-encounter.com/de/img/article-image-main/2015_1_Bobby_Car_Gallery_09.jpg_2130.jpg)

Ein ähnliches Auto hat auch BMW im Fuhrpark. Dieses Fahrzeug wurde in der Sendung TopGear mit Jeremy Clarkson vorgestellt und getestet, jedoch ist dieses Fahrzeug weit weniger bekannt als das von AUDI.

Trotz dem Erfolg dieser Autos auf den Rennstrecken, darf nicht vergessen werden, dass diese Systeme in dieser Konstellation nicht auf öffentlichen Straßen verwendet werden dürfen. Jedoch arbeiten die Hersteller intensiv daran, den Fahrer immer mehr zu entlasten und ihm bei kritischen Situationen zu unterstützen. So werden auch Systeme für den Stop-and-Go Verkehr getestet. Ein



**Abbildung 2.3:** Selbst fahrender AUDI RS7 auf dem Hockenheim 2014

solches System soll dem Fahrer entlasten. Es übernimmt innerhalb dieser Fahrsituation gewisse Aufgaben des Fahrers.

**Fazit** Heutige Fahrassistenzsysteme unterstützen den Fahrer in Gefahrensituationen wie beispielsweise mit ESP für schleudernde Fahrzeuge oder einem Notbremssystem für Fußgänger. Ebenso helfen sie dem Fahrer durch vollautomatisches oder teil-automatisiertes einparken.

Trotz der Fülle verschiedener Assistenzsysteme ist immer noch ein Fahrer nötig, der das Fahrzeug steuert.

## 2.4 Automatisch geführte Fahrzeuge

# KAPITEL 3

---

## Erste Experimente mit dem EV3

---

In diesem Abschnitt werden verschiedene Experimente mit dem EV3 vorgestellt. Dazu zählen das kennenlernen der Sensoren so wie deren Zusammenspiel.

### 3.1 Aufbau des Roboters

Anhand einer beiliegenden Anleitung wurde der Roboter zusammengebaut. Wie schon im Abschnitt 2.2 auf Seite 9 beschrieben, erhält der Roboter zwei unabhängige Motoren für seine Räder sowie ein voll bewegliches Stützrad, realisiert durch eine Kugel.

Die Steuereinheit, im Folgenden auch *Brick* genannt, ist oberhalb der Motoren so wie dem Stützrad angebracht. Die Motoren und Sensoren werden über Kabel mit dem Brick verbunden.

Für die folgenden Experimente wurden Halterungen für die benötigten Sensoren bereitgehalten.

### 3.2 Aufbau und Test des Ultraschallsensors

In diesem Abschnitt wird über den Test des Ultraschallsensors geschrieben. Das dargestellte Programm dient nur zu testzwecken und fand im späteren Verlauf des Projekts in dieser Version keine Verwendung mehr.

**Aufbau und Programmierung des Roboters** Der Roboter wurde mit Hilfe der beiliegenden Anleitung zusammengebaut. Er besitzt zwei Motoren, die jeweils eines der Räder antreiben. Damit das Heck nicht auf dem Untergrund aufsetzt, wurde eine freilaufende Kugel installiert. Als Sensor wurde bei diesem Aufbau der Ultraschallsensor verwendet. Er dient zu Hinderniserkennung und hat einen Arbeitsbereich von 3-250cm.

**Ziel** Der Roboter soll nach dem einprogrammierten Muster in einer ihm unbekannten Umgebung selbstständig Hindernissen ausweichen. Der Ultraschallsensor soll die Hindernisse erkennen. Dieser Versuch soll helfen, die Grenzen des Sensors kennenzulernen. Des weiteren dient dieser Versuchsaufbau dazu weitere Eigenheiten des Roboters kennenzulernen. Auch soll der Aufbau als Grundgerüst für weitere Sensoren und Versuche dienen.

Der Roboter wird mit einer von Lego Entworfenen Programmierungsumgebung programmiert. Diese Umgebung ist eine grafische Programmierung des Bricks. Diese Programmierungsumgebung soll im laufe des Projekts benutzt und getestet werden.

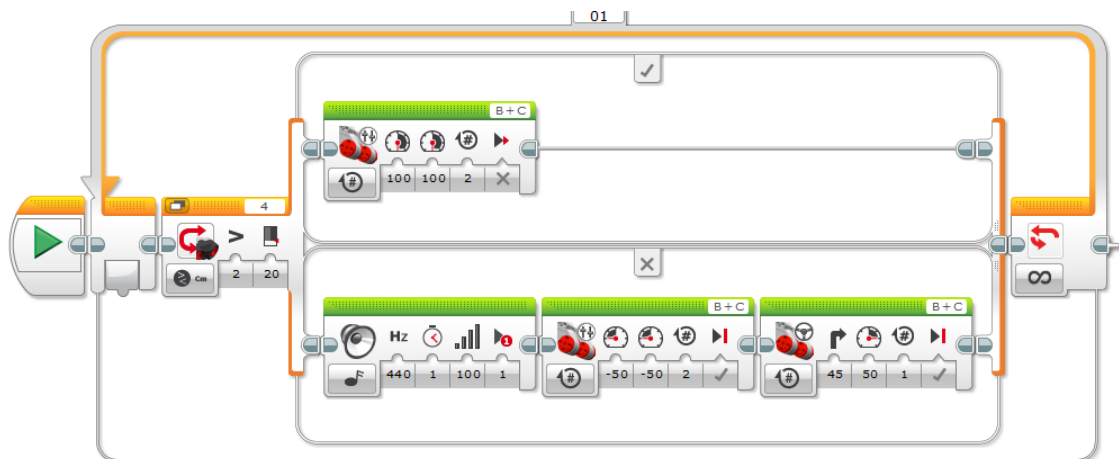


Abbildung 3.1: Lego-Programm zum Test und Kennenlernen des Ultraschallsensor

**Testprogramm des Ultraschallsensors** Im der Abbildung 3.1 wird das, mit der Lego- Programmierungsumgebung erstellte, Programm dargestellt.

Das Programm besteht aus zwei ineinander geschachtelte Schleifen. Die äußere Schleife besitzt keine Abbruchbedingung. Die innere der beiden Schleifen überwacht den Ultraschallsensor und entscheidet je nach Ergebnis des Sensors, was der Roboter machen soll.

Trifft der Vergleich **Vergleich erläutern** zu so fährt der Roboter mit Höchstgeschwindigkeit nach vorne, solange bis sich jedes der Räder genau zweimal gedreht hat. Danach wird auf Grund der Endlosschleife wieder der Ultraschallsensor abgefragt. Diesen Zweig erkennt man an dem Haken über den Anweisungen.

Trifft der Vergleich nicht zu, wird zuerst ein Ton über den integrierten Lautsprecher ausgegeben. Dieser Ton ist ein 440Hz Ton und wird mit voller Lautstärke eine Sekunde lang ausgegeben. Anschließend fährt der Roboter mit halber Kraft zwei Radumdrehung zurück. Ist dies erledigt, soll sich der Roboter um 45 Grad drehen. Aufgrund der Endlosschleife wird wieder von Vorne begonnen und der Ultraschallsensor wird abgefragt. Dieser Zweig ist in der Darstellung 3.1 an dem darüber stehenden x zuerkennen.



**Beobachtungen** Bei den Versuchen wurde festgestellt, dass die Hindernisse eine gewisse Breite und auch Höhe haben müssen, damit der Sensor sie wahrnimmt. Damit die Sensoren zeitlich relativ gute Ergebnisse liefern, dürfen die Schleifen der Sensoren nicht zu viele andere Programmbausteine besitzen. Dies kann dazu führen, dass der Sensor seine Abfrage zu spät ausführt und dadurch schon auf ein Hindernis gestoßen ist.

Des Weiteren wurde bei den Tests festgestellt, dass der Roboter im jetzigen Aufbau auf diversen Oberflächen, wie Teppich, Fliesen oder Parkett, verschieden große Kurvenradien fährt. Dieses Verhalten liegt an den unterschiedlichen Reibungswerten des jeweiligen Untergrunds.

Innerhalb der Programmierumgebung wurde festgestellt, dass die Übersichtlichkeit mit Zunahme der Komplexität abnimmt. Zum Testen der Sensoren und Aktoren des Roboters ist die Programmierumgebung geeignet.

Als positiv zu werten ist die Darstellung der Sensoren. Bei bestehender Verbindung mit dem EV3 kann der Anwender die Sensorwerte direkt in der Programmierumgebung nachvollziehen.

**Auswertung** Das Programm zum Ausweichen von Gegenständen ist in [Abbildung 3.1 auf der vorherigen Seite](#) dargestellt und näher erläutert. Die Programmierumgebung ist zum Einarbeiten und Kennenlernen der Sensoren geeignet. Jedoch wird das Programm je komplexer es wird auch unübersichtlicher. Die Umgebung zeigt bei bestehender Verbindung zwischen PC und Brick die Echtzeitdaten der Motoren und Sensoren an. Dies unterstützt beim Kennenlernen der Sensoren und hilft bei Problemen. So wurde festgestellt, dass der Ultraschallsensor Gegenstände, die eine gewisse Breite (Breite unter 2-5 cm) nicht erfüllen, nur fehlerhaft oder gar nicht erkannt werden.

### 3.3 Test des mitgelieferten Farbsensors

**Aufbau** Der Lego eigene Farbsensor wird zuerst nur über ein Kabel mit dem Brick verbunden.

Getestet wurde der Sensor wieder mit der Software von Lego. Dort wurde für die Erkennung von Farben die entsprechenden Parameter eingestellt, damit der Sensor verschiedene Farben erkennen kann. Dies sind acht verschiedene Farben, die von farblos über Gelb und Grün bis hin zu Rot und Schwarz reichen.

Im ersten Test zum Kennenlernen wurde auf ein gleichzeitiger Einsatz von Ultraschallsensor und Farbsensor verzichtet.

Im zweiten Test wird der Sensor fest am Roboter verbaut und auf den Boden gerichtet. Aufgrund des vorigen Tests wird der Sensor nicht zu weit über dem Boden montiert. Mit entsprechender Programmierung soll der Roboter solange fahren, bis sich auf dem Boden eine Markierung mit einer anderen Farbe befindet. Wird dies festgestellt, soll der Roboter anhalten.



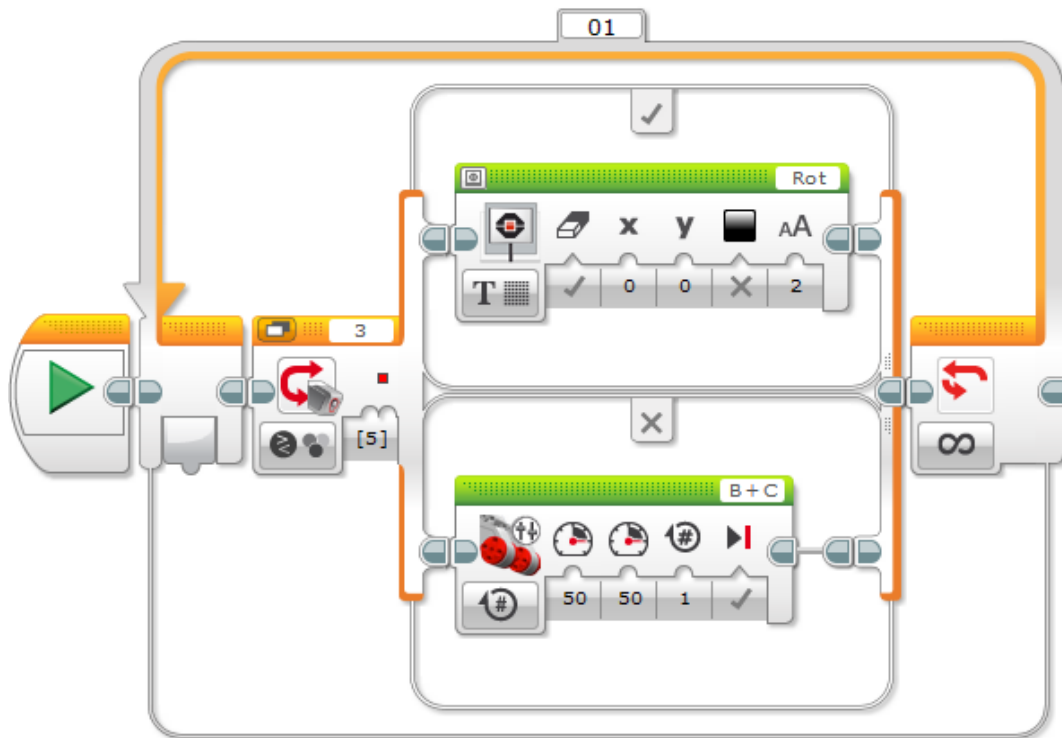


Abbildung 3.2: Lego-Programm zum Test und Kennenlernen des Farbsensors von Lego

**Testprogramm des Lego-Farbsensor** Das unter Abbildung 3.2 dargestellte Programm, ist ein Beispielprogramm für den Lego eigenen Farbsensor. Mit Hilfe dieses Programms wurde festgestellt, dass der Sensor manche Farben fehlerhaft erkennt. Dies ist zurückzuführen auf seine niedrige Farbauflösung. Diese Detektiert lediglich acht verschiedene Farben.

**Ziel** Ziel dieser Tests ist es den Sensor kennenzulernen. Hierzu zählen die notwendigen Abstände des Sensors, so wie die möglichen Farben die der Sensor erkennen könnte. Ebenfalls soll mit dem Test festgestellt werden, ob der Sensor für das Projekt genutzt werden kann. Dazu wird ein Programm geschrieben, in dem sich der Roboter bewegt und sobald er eine andere, noch nicht festgelegte, Farbe auf dem Boden erkennt, anhält. - Sensor kennenlernen

- Sensor einsetzen können
- Feststellung ob der Sensor für das Projekt geeignet ist

**Beobachtungen** Während dem ersten Teil des Tests, als der Sensor noch nicht fest am Roboter gefestigt war, wurde festgestellt, dass der Sensor einen gewissen Bereich hat in dem es zu Fehlmessungen kommen kann. Zusätzlich dazu war festzustellen, dass der Sensor bei Mischfarben falsche Farbwerte liefert. Er zeigte beispielsweise bei einer Mischung aus grüner und brauner Farbe, dass er die Farbe schwarz erkannt hätte.

**Auswertung** -Bedingt einsetzbar für das Projekt.

Grund  $\Rightarrow$  Farben müssen eindeutig sein und dürfen keine Mischfarben sein.

# KAPITEL 4

---

## Erste Experimente mit Lejos und der externen Kamera

---

### 4.1 Installation von Lejos in Eclipse

In diesem Abschnitt soll beschrieben werden wie die Erweiterung **Lejos** in die Java IDE Eclipse installiert wird.

- Eingehen auf die Reihenfolge der Installationen
- Alternativer Weg zu meinem Installationsweg

### 4.2 Beispielprogramm mit Lejos

Screenshot/ Listing  
Erklärung

### 4.3 Installation Vision subsystem V4

Aufgrund der schlechten Farbauflösung des Lego eigenen Sensors, wurde ein weiterer Sensor angeschafft. Für diesen Sensor mussten weiter Vorarbeiten und Änderungen vorgenommen werden.

**Besonderheiten dieses Sensors** -Reaktion auf der Kamera auf fluoreszierendes Licht!!!

**Installation der Kamera-Gerätetreiber unter Windows7** Die Kamera wurde mittels USB-A auf Micro-USB Kabel mit dem PC verbunden. Die Treiber wurden von der Seite [http://mindsensors.com/index.php?module=pagemaster&PAGE\\_user\\_op=view\\_page&PAGE\\_id=78](http://mindsensors.com/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=78) geladen.

Mit diesen Treibern kann anschließend das Programm NTXCamView ausgeführt werden. Dieses Programm kann unter <http://nxtcamview.sourceforge.net/> geladen werden.

Mit Hilfe dieses Programms kann die Kamera Bilder erstellen und die Farbe für die Detektion festgelegt werden.

Besonderheit dieser Treiberinstallation ist, dass die Treiber per Hand geladen werden müssen und nicht automatisch geladen werden.

**Deutsche Anleitung zum Installieren hinzufügen!!!** Für die Installation ist eine Internetverbindung notwendig. Es wird explizit angegeben an welchen Stellen diese benötigt wird. Für eine spätere Nutzung ist keine Internetverbindung vorgesehen.

1. Download der Geräte Treiber **Internetverbindung notwendig**
2. Download des Programms NTXCamView **Internetverbindung notwendig**
3. bla bla

## 4.4 Vision subsystem V4 — Kammeraprogramm

-Screenshot

-Was kann die Kamera

-Für aufgabe nutzbar? Ja / Nein -¿Wieso??

# KAPITEL 5

---

## Umsetzung des Projekts

---

### 5.1 Festlegung der Werte

- Geschwindigkeit
- Drehgeschwindigkeit
- Wartedauer

### 5.2 Einsatz Gyrosensor

In diesem Abschnitt wird näher erläutert weshalb für die Gesamtlösung des Projekts ein Gyrosensor genutzt wird. Dieser wäre zwar für die finale Umsetzung zwar nicht zwingend notwendig, jedoch wird die Handhabung und die Lesbarkeit des Programms deutlich verbessert durch den Einsatz eines Gyrosensors.

**Idee** Mit fortlaufender Dauer des Projektes stellte sich immer mehr heraus, dass es zu umständlich ist, für jeden Boden immer die Parameter für eine Drehung anzupassen. Nach kurzer Recherche entschied ich mich dazu, zusätzlich zum Ultraschall- und Farbsensor noch den Gyrosensor zu nutzen.

**Problem** Bei den ersten Versuchen mit dem Gyrosensor wurde recht schnell festgestellt, dass er doch eine Erhebliche Fehlertoleranz aufweist. Da diese nicht eingestellt werden kann muss diese mit Hilfe einer Berechnung gelöst werden.

**Problemlösung** Um diese Fehlertoleranzen auszugleichen wurde extra ein Programm entwickelt, welches unter einem eigenen Block zusammen gefasst wurde. Mit Hilfe des Programms ist es, bis auf kleinere Abweichungen von 1-3 Grad, möglich den Roboter um bis 180 Grad in jede Richtung zu drehen.

### 5.3 Einsatz eigener Blöcke

- Wieso weshalb warum
- was tun die Blöcke

### 5.4 Das Programm

---

## Abbildungsverzeichnis

---

1.1	Übersicht des zu befahrenden Parkurs . . . . .	6
1.2	Legende zu Bild <a href="#">1.1 auf Seite 6</a> . . . . .	7
2.1	Abstrakte Darstellung der Antriebsräder des Versuchsroboters . . . . .	10
2.2	Darstellung des Abstandes in der Headuni am beispiel BMW 1er . . . . .	12
2.3	Selbst fahrender AUDI RS7 auf dem Hockenheim 2014 . . . . .	14
3.1	Lego-Programm zum Test und Kennenlernen des Ultraschallsensor . . . . .	16
3.2	Lego-Programm zum Test und Kennenlernen des Farbsensors von Lego . . . . .	18

---

## Tabellenverzeichnis

---

1.1 Übersicht der Projektplanung . . . . .	5
--	---



---

## Listings

---