

# 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 . . . . .	5
<b>2</b>	<b>Grundlagen</b>	<b>7</b>
2.1	Bedeutung des Roboters . . . . .	7
2.2	Antriebsart des Roboters . . . . .	7
2.3	Assistenzsysteme heutiger Fahrzeuge . . . . .	7
2.3.1	ABS - Antiblockiersystem . . . . .	7
2.3.2	PDC - Park Distanz Kontrolle . . . . .	8
2.3.3	ESP - Elektronische Stabilitätscontrolle . . . . .	8
2.3.4	Automatisches einparken . . . . .	8
2.4	Spurgeführte Fahrzeuge . . . . .	8
<b>3</b>	<b>Erste Experimente mit dem EV3</b>	<b>9</b>
3.1	Aufbau des Roboters . . . . .	9
3.2	Aufbau und Test des Ultraschallsensors . . . . .	9
3.3	Test des mitgelieferten Farbsensors . . . . .	10
<b>4</b>	<b>Anhang</b>	<b>12</b>
4.1	Installation von Lejos in Eclipse . . . . .	12
4.2	Testprogramm des Ultraschallsensors . . . . .	12
4.3	Testprogramm des Lego-Farbsensor . . . . .	13
4.4	Installation Vision subsystem V4 . . . . .	14
	<b>Abbildungsverzeichnis</b>	<b>15</b>
	<b>Tabellenverzeichnis</b>	<b>16</b>
	<b>Listings</b>	<b>17</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 ausweichen können.

### 1.2 Projektplanung

Dauer	Art der Tätigkeit	Meilenstein
1Woche	Einarbeitung	1. Meilenstein

**Tabelle 1.1:** Übersicht der Projektplanung

**Dies ist die Tabelle der Projektplanung**

### 1.3 Verwendetet 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 ...

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.

### 2.1 Bedeutung des Roboter

### 2.2 Antriebsart des Roboters

Zwei Haupträder werden mit Getrieb- oder Schrittmotoren unabhängig angetrieben. Ein rein passives Stützrad, ähnlich wie bei den Einkaufswagen im Supermarkt (da sind es aber zwei), stützt das Fahrzeug an einem dritten Punkt.

### 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.

#### 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 des Fahrzeugs 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.

### 2.3.2 PDC - Park Distanz Kontrolle

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. Zusätzlich dazu wird der Abstand und der bestmögliche Fahrweg visuell im Livebild dargestellt.

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

### 2.3.3 ESP - Elektronische Stabilitätskontrolle

### 2.3.4 Automatisches einparken

## 2.4 Spurgeführte 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.

### 3.2 Aufbau und Test des Ultraschallsensors

**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 zur 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.

**Beobachtungen** Bei den Versuchen wurde festgestellt, dass die Hindernisse eine gewisse Breite und auch Höhe haben damit der Sensor sie richtig 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. **Problematik schildern**

Des Weiteren wurde bei den Tests festgestellt, dass der Roboter im jetzigen Aufbau auf diversen Oberflächen, wie Teppich, Fliesen oder Parkett, verschieden große Kurvenradien fahren. Dies 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 ?? auf 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 **Breite unter 2-5 cm** die eine gewisse Breite nicht erfüllen nur fehlerhaft oder gar nicht erkannt werden.

### 3.3 Test des mitgelieferten Farbsensors

**Aufbau** Der Legoeigene 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.



**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.



Im der Abbildung 4.1 auf der vorherigen Seite 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. Anschliesend 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 4.1 auf der vorherigen Seite an dem darüber stehenden x zuerkennen.

### 4.3 Testprogramm des Lego-Farbsensor

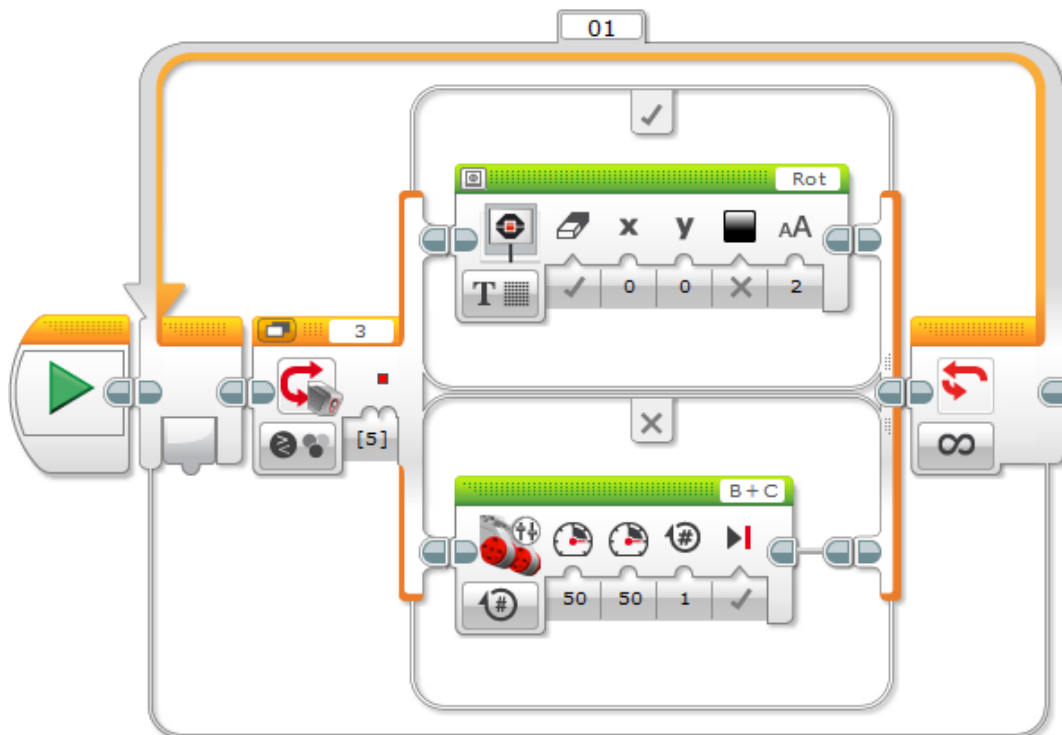


Abbildung 4.2: Lego-Programm zum Test und Kennenlernen des Farbsensors von Lego

Das unter Abbildung 4.2 auf der vorherigen Seite 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.

## 4.4 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

---

## Abbildungsverzeichnis

---

4.1	Lego-Programm zum Test und Kennenlernen des Ultraschallsensor . . . . .	12
4.2	Lego-Programm zum Test und Kennenlernen des Farbsensors von Lego . . . . .	13

---

## Tabellenverzeichnis

---

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

---

## Listings

---