DHBW
Duale Hochschule
Baden-Württemberg

HARMAN

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

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 unteranderem 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 Ëclipse"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 gesammten 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

Erste Experimente mit dem EV3

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

2.1 Aufbau des Roboters

Anhand einer beiliegenden Anleitung wurde der Roboter zusammengebaut.

2.2 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 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 Programmierumgebung programmiert. Diese Umgebung ist eine grafische Programmierung des Bricks. Diese Programmierumgebung 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 dazuführen, 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.

2.3 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 :⇒ Farben müssen eindeutig sein und dürfen keine Mischfarben sein.

KAPITEL 3

Anhang

3.1 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

3.2 Testprogramm des Ultraschallsensors

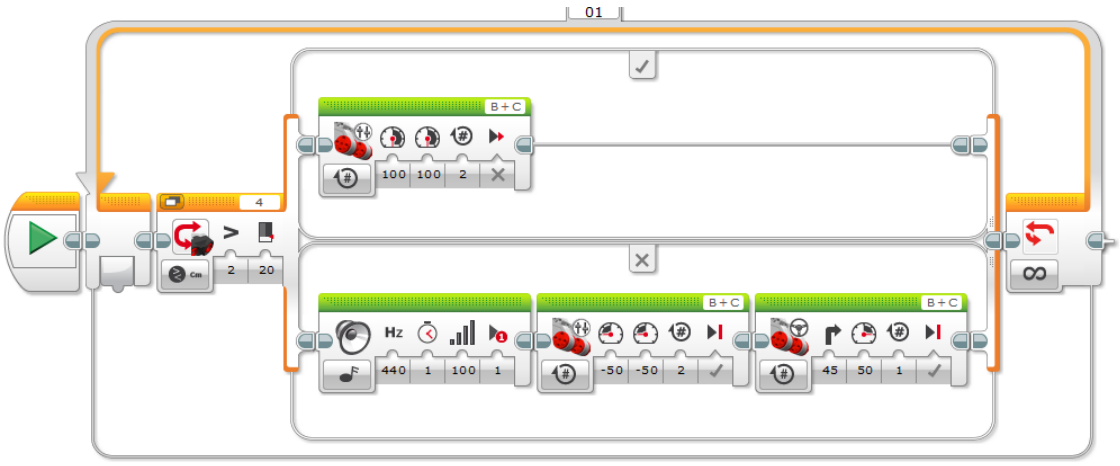

Abbildung 3.1: Lego-Programm zum Test und Kennenlernen des Ultraschallsensor

Im der Abbildung 3.1 auf der vorherigen Seite wird das, mit der Lego-Programmierumgebung 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 3.1 auf der vorherigen Seite an dem darüber stehenden x zuerkennen.

3.3 Testprogramm des Farbsensor

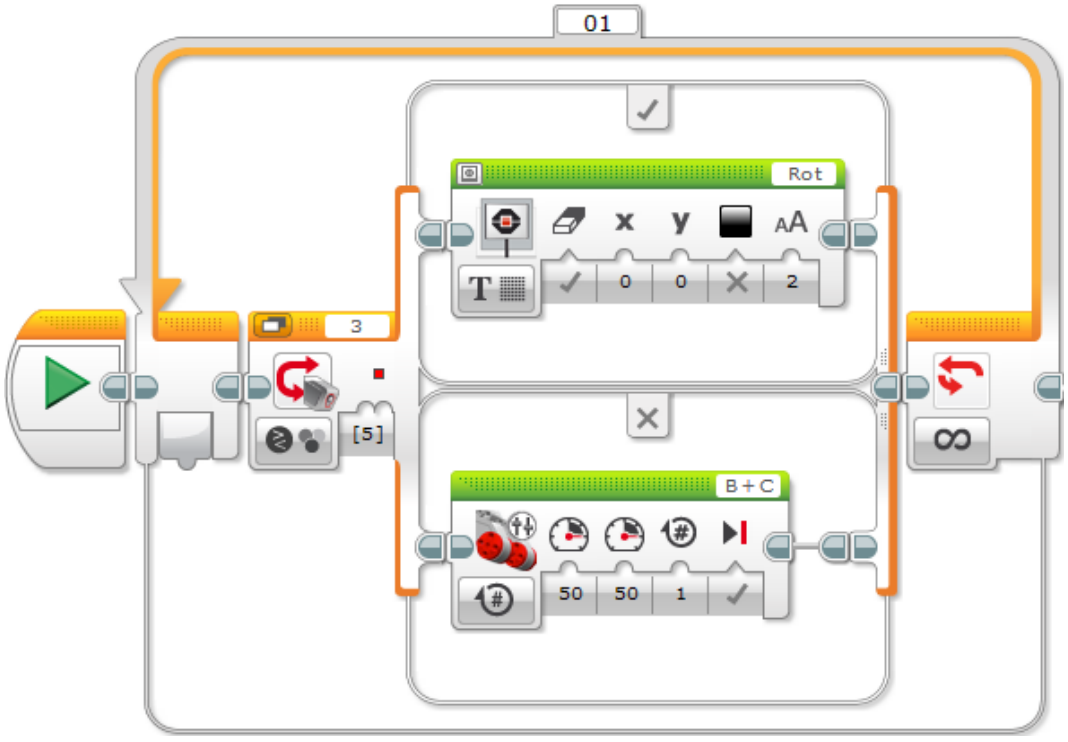

Abbildung 3.2: Lego-Programm zum Test und Kennenlernen des Lichtsensors von Lego

Abbildungsverzeichnis

Tabellenverzeichnis

Listings