Autonome Abstandskontrolle

-Dokumentation-

HS-Augsburg - SS2010 - IAM6 - Studienarbeit Interaktion - Prof. Dr.-Ing Thomas Rist

Autoren

Florian Dootz
Am Hasenbichl 13
86678 Ehingen
+491781417802
florian.dootz@chilloutsofa.com

Geronimo Wüst
Stettenstraße 32
86150 Augsburg
+491707460983
schlonz2k@abwesend.de

Abstract

Diese Studienarbeit beschäftigt sich damit, aus relativ Preisgünstigen Bauteilen eine eigene Fahrzeugsteuerung zu löten und zu programmieren. Es wurde ein Prototyp erstellt, welcher mittels Infrarot-Sensoren den Abstand zum Objekt vor ihm, immer gleich hält.

Categories and Subject Descriptors

B1.1 (Control Design Styles)
» Hardwired control
» Microprogrammed logic arrays
» Writable control store

General Terms

» Experimentation
» Security
» Measurement

Keywords

» Roboter
» Steuerung
» Infrarot
» H-Brücke
» Arduino
» Lego Motoren

Inhaltsverzeichnis

1. Motivation
2. Verwandte Arbeiten
3. Andwendungsszenario & Use Cases
    3.1 Erweiterung von Staubsaugerbots als Szenario
    3.2 Definition der Use Cases
4. Entwurf eines Prototyps
    4.1 Verwendete Technologie
    4.2 Konzeptueller Aufbau des Prototypen
        4.2.1 Arduino Board (Microcontroller)
        4.2.2 Infrarot Sensor
        4.2.3 H-Brücke
        4.2.4 Motoren
5. Implementierung
    5.1 Hard- und Softwareumgebung
        5.1.1 Hardware
        5.1.2 Software
    5.2 Kern der Implementierung
    5.3 Umfang des Prototypen
6 Evaluation
    6.1 Ziele der Evaluation und Methodik
    6.2 Durchführung der Evaluation
        6.2.1 Verlässlichkeit
        6.2.2 Kostengünstige Umsetzung
    6.3 Interpretation der Evaluationsergebnisse
        6.2.1 Verlässlichkeit
        6.2.2 Kostengünstige Umsetzung
7 Zusammenfassung
    7.1 Erzielte Ergebnisse
    7.2 Erweiterungsmöglichkeiten / Ausblick
8 Danksagungen
9 Referenzen

1. Motivation

Das Thema der Fahrzeugsteuerung mittels Umgebungsinformationen ist ein sehr aktuelles Thema. Sei es beim Militär, wo schon sehr ausgefeilte Systeme vorhanden sind, oder im zivilen Bereich, in welchem einem das Auto immer mehr mittels Sensoren unterstützt.

Hier haben wir angesetzt. In unserer Studienarbeit haben wir uns generell in das Gebiet der Elektrotechnik eingearbeitet und konnten damit ein kleines Fahrzeug konstruieren, welches automatisch einen gewissen Abstand zu Objekten hält, welche vor ihm liegen.

2. Verwandte Arbeiten

Seit längerem schon geben Autos akkustische Signale beim annähern an Hindernisse. Aktuelle Modelle unterstützen den Fahrer auch beim Verhindern von Auffahrunfällen.

Das große Pioniergebiet, in welchem noch viel zu erreichen ist sind Drohnen, also jegliche Art von unbemannten Fahrzeugen. Unbemannte Fahrzeuge kann man in zwei Kategorien aufteilen. Einmal die, welche ferngesteuert handeln, und diese welche autonom handeln. Die zweite Gruppe ist die wesentlich interessantere. Gerade beim Militär gibt es hier große Entwicklungen was den Luftraum betrifft. (UAVs, Unmanned Air Vehicle)

Es ist wesentlich leichter Autonome Fahrzeuge für den Luftraum zu entwerfen als für den Bodeneinsatz. Im Luftraum kann man dem UAV leicht GPS Koordinaten übergeben und das Fluggerät wird es anfliegen. Ein Bodenfahrzeug muss hier sehr viele Hindernisse überwinden. Es gibt sogar einen weltweiten Wettbewerb, in welchem sich autonome Fahrzeuge von Wissenschaftlern gegenseitig in verschiedenen Aufgaben messen (DARPA Grand Challenge).

Beim Militär findet man nicht ohne Grund die am weitesten fortgeschrittensten Systeme. Zum einen natürlich wegen den Gefahren, welchen man die eigenen Männer nicht aussetzen möchte, zum anderen jedoch auch aus Kostengründen. Forschungen beim Militär werden erstmal nicht für den Straßenverkauf getätigt. Darum kann man den Kostenfaktor und die Produktionskosten etwas in den Hintergrund stellen.

Genau aus diesem Grund haben wir uns dieses Gebiet vorgenommen. Während es für den Luftraum schon mehrere Drohnen privat gebaut werden gibt es für den Bodeneinsatz noch sehr wenig zu finden. In unserer Arbeit hatten wir nun sowohl die Möglichkeit uns mit der Technik und der Programmierung vertraut zu machen als auch ein sehr kostengünstiges Modell herzustellen.

3. Andwendungsszenario & Use Cases

3.1 Erweiterung von Staubsaugerbots als Szenario

Dadurch, dass wir uns im allgemeinen damit befasst haben, eine Steuerung für Fahrzeuge zu entwickeln, sind eine Vielzahl von Szenarien denkbar. Im folgenden wird ein sehr Praxisnahes Beispiel beschrieben.

Es gibt bereits autonome Staubsaugerbots. Diese finden Ihren Weg jedoch nicht mittels Berührungslosen Sensoren. Mit leistungsstarken Infrarot Sensoren könnte so ein Staubsaugerbot nicht nur verhindern gegen sensible Objekte zu fahren, er hätte auch sofort einen Überblick über den gesamten Sichtbereich des Raumes und könnte seine Route energie und zeiteffizient planen.

3.2 Definition der Use Cases

Akteure: Prototyp, Hindernis

Use Case: Vorwärts
- Scenario: Prototyp fährt vorwärts
- Vorbedingung: Kein Hinderniss im Ruhebereich oder näher
- Nachbedingung: Kein Hindernis im Ruhebereich oder Hindernis im Ruhebereich

Use Case: Rückwärts
- Scenario: Prototyp fährt rückwärts
- Vorbedingung: Hindernis näher als der Ruhebereich, Beliebiger Fahrzeugzustand
- Nachbedingung: Hindernis im Ruhebereich, Fahrzeug im Stillstand

Use Case: Anhalten
- Scenario: Prototyp hält an
- Vorbedingung: Hinderniss im Ruhebereich, Fahrzeug in Bewegung
- Nachbedingung: Hinderniss im Ruhebereich, Fahrzeug im Stillstand

4. Entwurf eines Prototyps

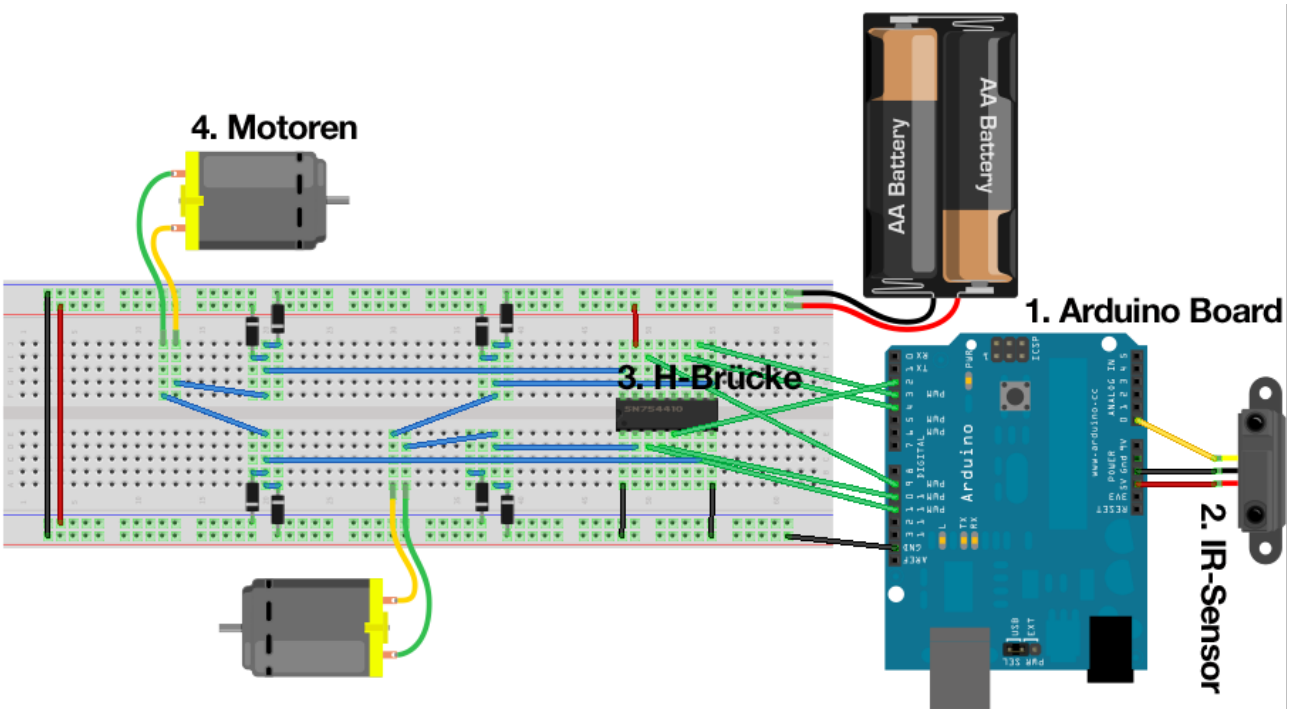
4.1 Verwendete Technologie

Als Sensor, um Informationen über die Umgebung zu erhalten, verwendeten wir die Infrarot Technologie. Dies ist eine kostengünstige Möglichkeit um Abstände zu ermitteln. Der Infrarot Sensor sendet hierbei Infrarot Strahlen aus, treffen diese nun auf ein Objekt werden sie reflektiert und ein zweites Element auf dem Sensor erkentn diese Strahlung.

Um die Signale zu verarbeiten wird ein Arduino Board verwendet welchem mittels einer H-Brücke die Möglichkeit gegeben wird zwei LEGO-Motoren zu steuern.

4.2 Konzeptueller Aufbau des Prototypen

Der Prototyp besteht im Wesentlichen aus 4 Elementen, diese sind in der nachfolgenden Grafik dargestellt und werden beschrieben.



4.2.1 Arduino Board (Microcontroller)

Das Arduino Board ist ein Board mit Microcontroller und mehreren ein- und Ausgängen. Es lässt sich über USB sehr leicht Programmieren. Als 5V Stromquelle nutzt es entweder den USB Port oder eine andere Stromquelle, wie eine Batterie.

Dieses Board liest die Daten des Infrarot Sensors über einen Analogen Eingang aus. Außerdem übernimmt es auch die korrekte Ansteuerung der Motoren. Somit befindet sich die komplette Logik auf dem MC des Arduino Boards.

4.2.2 Infrarot Sensor

Der Infrarot Sensor nutzt 5V als Stromquelle und gibt ein analoges Signal an das Arduino Board weiter. Der verwendete Sensor erfasst Objekte auf eine maximale Distanz von etwa 50cm zuverlässig. Um einen schöneren und genaueren Datenstrom zu erhalten, wurde vor den Infrarot-Sensor noch ein Kondensator Modul geschaltet.

4.2.3 H-Brücke

Die H-Brücke bietet dem Arduino Board insgesamt sechs Eingänge um zwei gleichstrom Motoren beliebig steuern zu können. Als zusätzliche Stromquelle wird

eine 9V Batterie verwendet.

Mittels den logischen Signalen des Arduino Boards lassen sich dise 9V nun auf beliebige Ausgänge der H-Brücke umleiten.

4.2.4 Motoren

Bei den Motoren handelt es sich um 9V Lego Motoren welche jeweils 3,5mA Strom verbrauchen. Motoren sind eine induktive Last, also träge Verbraucher, welche selbst Strom erzeugen können, etwa durch auslaufen. Um in diesem Fall die restliche Schaltung zu schützen sind pro Motor vier Freilaufdioden verbaut.

5. Implementierung

5.1 Hard- und Softwareumgebung

5.1.1 Hardware

- Desktop Computer
- Arduino Board (Duemilanove)
- Infrarot Sensor
- Kondensator
- H-Brücke (L298)
- Zwei Motoren (LEGO 9V)
- Acht Dioden
- 9V Batterie

Mittels eines Computers mit USB-Ausgang wurde das Arduino Board programmiert und erhielt seine 5V Stromzufuhr. Dieses Board steuert die Motoren an. Um dies zu erreichen wurde zunächst mit Transistoren experimentiert, man stellte aber schnell fest, dass dies nicht die nötige Flexibilität mit sich bringt. So ist es bei dieser Methode zum Beispiel nicht möglich die Drehrichtung des Gleichstrommotors zu beeinflussen.

Als wir uns anschließend die Bauteile für eine logische Schaltung auswählten entschieden wir uns dafür ein IC als H-Brücke zu verwenden um die Motoren flexibel ansteuern zu können. Dabei griffen wir zu dem IC (L298). Um den IC vor der induktiven Last der Motoren zu schützen wurden pro Motor noch vier Freilaufdioden verbaut.

Um den Abstand zum Objekt vor dem Fahrzeug zu ermitteln, verwendeten wir einen Infrarot Sensor. Im ersten Anlauf hat der Sensor die Daten direkt an das Arduino Board weitergegeben. Es stellte sich jedoch schnell heraus, dass die Daten etwas ungenau und unruhig waren. Um einen saubereren Datenstrom zu erhalten, welcher von Unregelmäßigkeiten befreit ist, wurde dann ein Kondensator zwischen den IR-Sensor und das Arduino Board geschaltet.

5.1.2 Software

- » Entwicklungs-Betriebssystem: Windows 7
- » Entwicklungsumgebung: Arduino IDE
- » Programmiersprache: Processing
- » Verwendete Bibliotheken: Keine
- » Plattform: Arduino (Atmel Microcontroller)

Die Software wurde in der Arduino eigenen Entwickungsumgebung auf Windows 7 Systemen geschrieben und vom Arduino Board ausgeführt. Die Software verarbeitet die Sensor Daten des Infrarot-Sensors und steuert die Motoren an. Mit etwas Experimentieren wurden die richtigen Werte für die Drehrichtung der Motoren und für die Bewertung der IR Daten schnell ausfindig gemacht.

5.2 Kern der Implementierung

Es möchte kurz erwähnt sein, dass der Hauptfokus der Studienarbeit auf der Hardwareseite lag, undzwar bei dem Sensor und dem Motortreiber. Hier nun Auszüge aus dem Code, welcher die Signaldaten verarbeitet und den Motor ansteuert.

Über die Klassenvariablen hat man im Code Zugriff auf alle wichtigen Pins.

```
int frontSensor = 0;          // infrared input pin
int frontSensorValue = 0;     // stores infrared informations

int motorA =  3;              // pin to activate motor A
int motorAPin1 = 4;           // first control pin for motor A
int motorAPin2 =  5;          // second control pin for motor A

int motorB =  10;             // pin to activate motor B
int motorBPin1 =  11;         // first control pin for motor B
int motorBPin2 =  12;         // second control pin for motor B
```

Um die Motoren korrekt anzusteuern muss man die richtigen logischen Signale auf die jeweiligen Pins legen. Die verschiedenen Kombinationen sind hier für Motor A als Tabelle dargestellt:

Eingaben		Motorfunktion
motorA = 1	motorAPin1 = 1 motorAPin2 = 0	Vorwärts
	motorAPin1 = 0 motorAPin2 = 1	Rückwärts
	motorAPin1 = 0 motorAPin2 = 0	Vollbremsung
	motorAPin1 = 1 motorAPin2 = 1	Vollbremsung
motorA = 0	motorAPin1 = egal motorAPin2 = egal	Motor läuft 'normal' aus.

Hier nun ein Codeauszug, welcher den Prototypen nach vorne fahren lässt, wobei die Pins 'MotorA' und 'MotorB' schon zu Beginn der Applikation mit einem logischen '1' belegt sind. Man kann jeden Pin entweder mit HIGH oder LOW ansteuern. Dies entspricht einem Bit, also einer logischen 1 oder 0.

```
void forwards()
{
  digitalWrite(motorAPin1, HIGH);      // digital signal '1'
  digitalWrite(motorAPin2, LOW);       // digital signal '0'

  digitalWrite(motorBPin1, HIGH);      // digital signal '1'
  digitalWrite(motorBPin2, LOW);       // digital signal '0'
}
```

Hier nun noch der Loop, welcher immer wieder ausgeführt wird und überprüft ob sich etwas in Reichweite befindet.

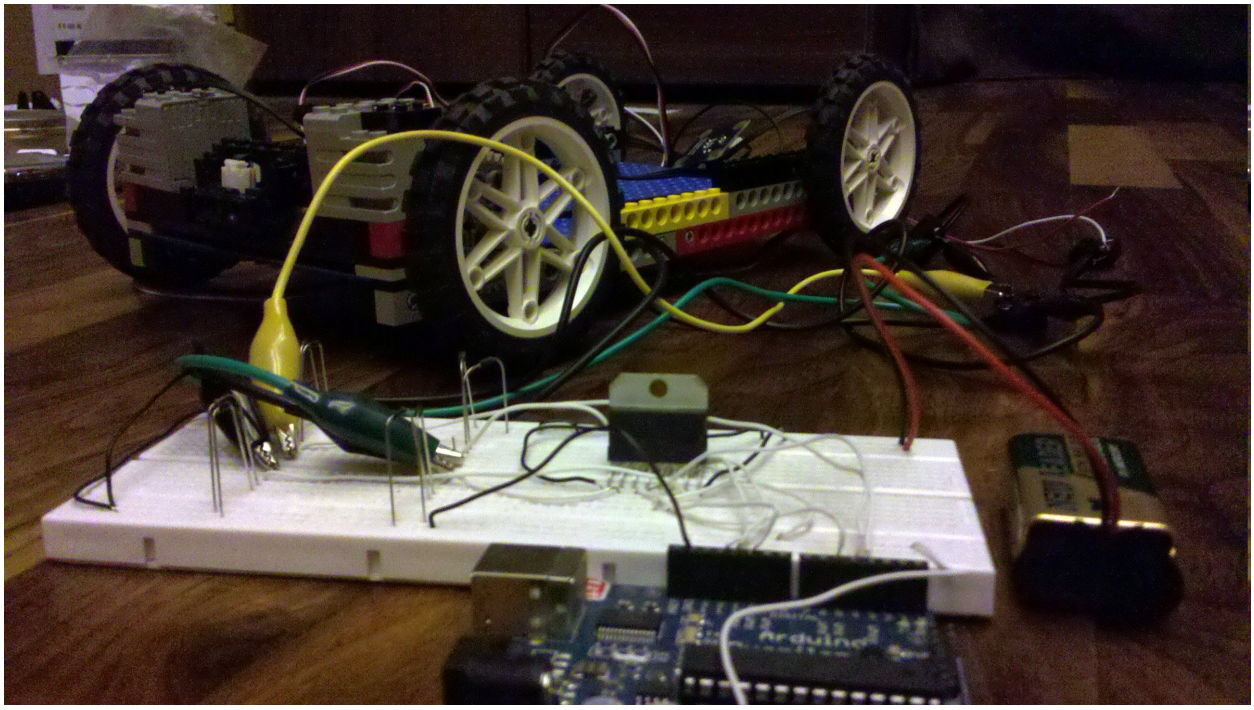
```
void loop()
{
  frontSensorValue = analogRead(frontSensor);   // sensor read
  Serial.println(frontSensorValue);    // IR-value test output

  digitalWrite(motorA, HIGH);          // enables motor A
  digitalWrite(motorB, HIGH);          // enables motor B

  if (frontSensorValue < 120)          // object too far away
  {
    forwards();                        // -> go forwards
  }
  else if (frontSensorValue > 170)     // object too close
  {
    backwards();                       // -> go backwards
  }
  else                                 // optimal distance
  {
    brake();                           // -> stop motors
  }
}
```

5.3 Umfang des Prototypen

Das beschriebene Konzept der automatischen Abstandseinhaltung wurde im Prototypen funktionsfähig umgesetzt. Allerdings wurde das Prinzip der Autonomie nicht komplett umgesetzt.

Um die Stromversorgung der Motoren sicherzustellen besitzt der Prototyp eine 9V Batterie. Die 5V-Stromversorgung des Microcontrollers erfolgt jedoch momentan noch über einen USB-Port. Bei diesem Aufbau bleibt das Arduino Board mitsamt Motor-Treiber neben dem Computer stehen und die Fahrzeugmotoren werden über lange Kabel angetrieben.

Um den Prototypen autonom losschicken zu können müsste man also die 5V Stromversorgung sicherstellen und den Motortreiber fest auf ein Board löten. Diese Schritte wären für eine praktische Verwendung, abseits des Testumfeldes, nötig.



6 Evaluation

6.1 Ziele der Evaluation und Methodik

Mittels dieser Evaluation soll die Praxis- und Alltagstauglichkeit der im Prototypen umgesetzten Abstandskontrolle ermittelt werden. Dabei spielen vor allem drei Faktoren des Systems eine wichtige Rolle:

» Verlässlichkeit
» Kostengünstige Umsetzung

Um die Verlässlichkeit überprüfen zu können, wurden einige Indoor-Testfahrten abgehalten und dokumentiert. Natürlich wären Testläufe im Außenareal mit windigen oder staubigen Verhältnissen sehr interessant gewesen, dafür ist der Prototyp aber keinesfalls ausgelegt.

Um einen Überblick über die Kosten zu haben werden die Anschaffungskosten

zusammengerechnet und dem erzielten Ergebnis gegenübergestellt.

6.2 Durchführung der Evaluation

6.2.1 Verlässlichkeit

Es wurden 10 Testläufe mit dem Roboter durchgeführt. Dabei wurde ein Hindernis vor den Sensor gestellt und verschoben um festzustellen ob dieser die geforderte Aufgabe mehrmals nacheinander fehlerfrei bewältigen konnte.

Wir konnten zwei Probleme feststellen. Beim ersten Durchlauf fuhr der Prototyp in die falsche Richtung und rammte das Hindernis. Ein einziges mal konnte man Schwankungen in der Messung registrieren was zu unnötigen Bewegungen führte. Ansonsten verhielt sich der Roboter genauso wie erwartet.

6.2.2 Kostengünstige Umsetzung

Im folgende eine grobe Preisliste der einzelnen Bauteile:

» Arduino Board (Duemilanove)	25,00€
» Infrarot Sensor	15,00€
» Kondensator	00,20€
» H-Brücke (L298)	02,50€
» Zwei Motoren	06,00€
» Acht Dioden	01,00€
» 9V Batterie	02,00€
» Gesamt	50,70€

Einen funktionierenden und ausbaufähigen Prototypen kann man für etwa 50€ erstehen. Selbstverständlich kommt es vor allem bei den Sensoren und den Motoren darauf an, in welcher Qualität man diese Bauteile erstehen möchte.

6.3 Interpretation der Evaluationsergebnisse

6.3.1 Verlässlichkeit

Insgesamt waren die Testläufe sehr zufriedenstellend. Beim ersten Testdurchlauf als der grobe Fehler auftrat und der Prototyp in die falsche Richtung fuhr hatten wir schlicht die Motoren falsch herum angeschlossen. Die Schwankungen in der Messung wurden immer dann zum Problem, wenn das Hindernis genau an die Grenze des Ruhebereiches des Roboters gestellt wurde. Normalerweise fuhr der Prototyp in den Ruhebereich hinein und das Problem trat nicht auf. Zudem können bei Infrarotsensoren Probleme auftreten, denn Staub oder Verschmutzungen können die Werte verfälschen.

Wir können also abschließend sagen, dass unser Prototyp in geschlossenen Räumen zuverlässig funktioniert.

6.3.2 Kostengünstige Umsetzung

Wir haben unter anderem versucht eine möglichst Preiswerte Lösung zu Entwickeln. Mit 50€ ist der Prototyp zwar etwas teurer als erwünscht, dies liegt jedoch stark am Arduino Board, welches die hälfte dieses Betrages ausmacht. Dabei sollte man bedenken, dass man zum einen auch einen wesentlich billigeren Microcontroller verwenden könnte und zum anderen, dass das Arduino Board für eine vielzahl an Projekten verwendet werden kann.

Das größte Problem beim preiswerten Basteln und Entwickeln sind die Infrarot Sensoren. Für eine ausgefeiltere Fahrzeugsteuerung wären vier oder mehr IR-Sensoren notwendig, was die Kosten leider explosionsartig in die Höhe treiben würde. Möchte man dann noch Sensoren verwenden, welche eine Entfernung von einem Meter oder mehr messen können, werden die Experimente schnell sehr teuer.

7 Zusammenfassung

7.1 Erzielte Ergebnisse

In der hier vorliegenden Studienarbeit wurde durch Implementierung eines Prototyps überprüft, ob eine Robotersteuerung kostengünstig hergestellt und in relativ kuzer Zeit zum Laufen gebracht werden kann.

Anfangs wurden bereits existierende, technisch ähnliche Projekte, wie zum Beispiel der Roboterstaubsauger recherchiert, beschrieben und in Bezug zu dem hier angewandten Anwendungsszenario gebracht. Die H-Brücke gab uns die Möglichkeit die Drehrichtung der Gleichstrommotoren zu wechseln und somit den Prototyp vorwärts als auch rückwärts fahren lassen.

Wir haben erfolgreich eine simple Fahrzeugsteuerung implementiert und bewiesen, dass dies ohne allzu teuere Bauteile möglich ist. Daraus ziehen wir den Schluss, dass man neben bereits weit entwickelten autonomen Luftfahrzeugen auch Bodenfahrzeuge für den Privatgebrauch nutzen kann.

7.2 Erweiterungsmöglichkeiten / Ausblick

Natürlich müsste der Prototyp mit weiteren Sensoren bestückt werden um richtig effektiv eingesetzt werden zu können. Für den Einsatz im Freien bräuchte man wesentlich zuverlässigere und robustere Sensoren sowie ein stabileres Chassis.

Mit vier Sensoren, einen nach vorne, einen nach hinten, und jeweils einen an der Seite, oder einen rotierenden Sensor auf dem Roboter liese sich der Roboter noch um einiges erweitern. Mittels dieser Hardwareerweiterung könnte man die Software so erweitern, dass sich das Fahrzeug autonom durch einen Raum bewegen kann. Außerdem wären damit dann vielfältige Szenarien denkbar. Beispiele sind eine Parkhilfe, Roboterstaubsauger, oder Rettungs- und Krisensituationen.

8 Danksagungen

Danke an Prof. Dr.-Ing Thomas Rist, Christian Glas und Michael Ilg für die Unterstützung.

9 Referenzen

Datenblätter für Transistoren, ICs und Dioden
http://www.philohome.com/motors/motorcomp.htm
http://www.kpsec.freeuk.com/trancirc.htm
http://www.mikrocontroller.net/topic/72766
http://www.strippenstrolch.de/1-2-9-motortreiber-298.html
http://en.wikipedia.org/wiki/Integrated_circuit
http://en.wikipedia.org/wiki/H-bridge
http://de.wikipedia.org/wiki/Transistor
http://www.youtube.com/user/makemagazine
http://www.afrotechmods.com/

Tools:
http://fritzing.org/
http://jumk.de/formeln/widerstand.shtml
http://www.sengpielaudio.com/Farbcodewiderstaende02.htm