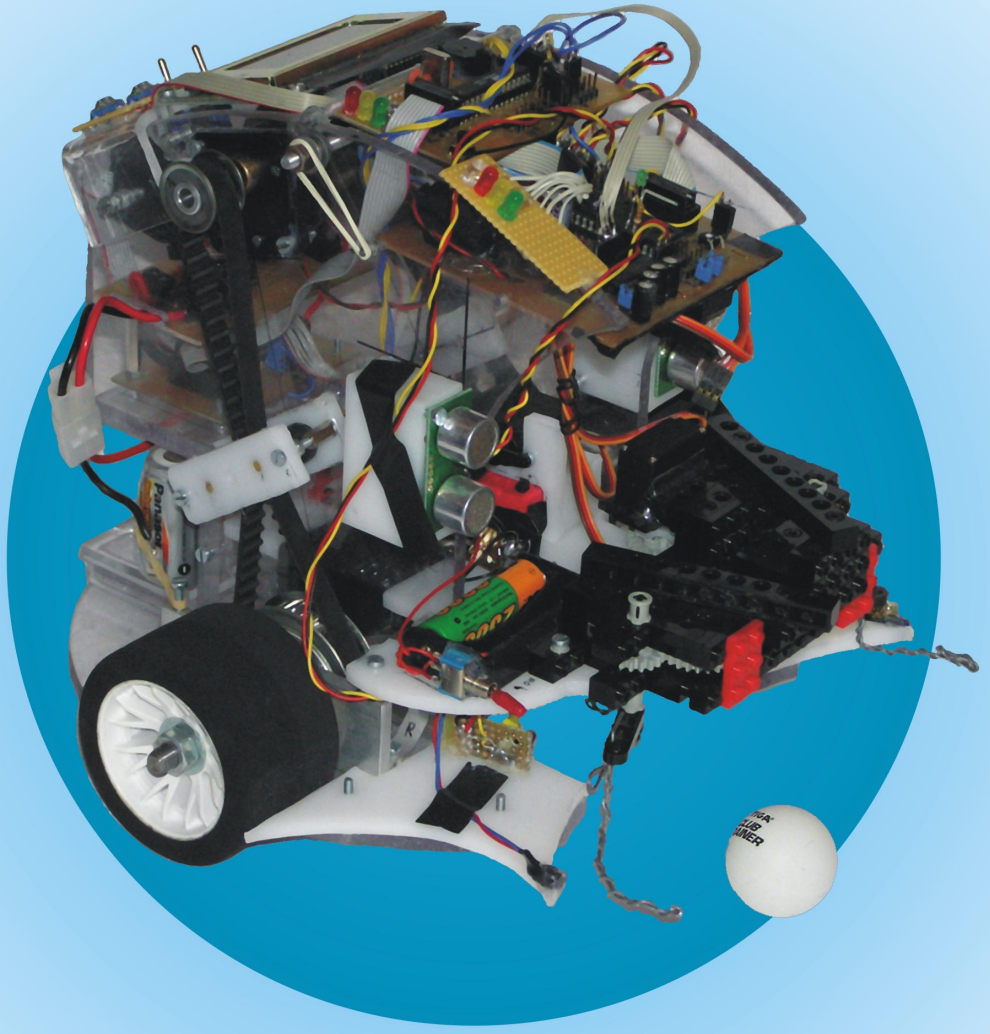

Aufbau und Programmierung eines Wettbewerbroboters

Wernher-von-Braun-Gymnasium Kollegstufenjahrgang 2006/2007

F A C H A R B E I T

aus dem Fach

P H Y S I K

Thema: Aufbau und Programmierung eines Wettbewerbroboters

Verfasser: Julian Straub

Leistungskurs: Physik

Kursleiter: Herr Matthias Turba

Abgabetermin: 26.01.2007

Erzielte Note: _____ In Worten: _____

Erzielte Punkte: _____ In Worten: _____

(einfache Wertung)

_____

(Unterschrift des Kursleiters)

Inhaltsverzeichnis

1. EINLEITUNG

Den Roboter, mit dem sich die Facharbeit beschäftigt, habe ich in einem halben Jahr Arbeit für den Roboking Wettbewerb 2007 konstruiert. Zum dritten Mal nahm ich an diesem Wettbewerb, der von der TU Chemnitz veranstaltet wird, teil. Jedes Jahr gilt es, eine andere Aufgabe zu lösen. Bei meinem ersten Roboking, den ich unter der Leitung von Herrn Turba mit einem Team aus Mitschülern im Jahr 2005 bestritten hatte, ging es darum, Tennisbälle aufzusammeln und in eine Basis zu bringen. Spätestens ab diesem Wettbewerb war meine Begeisterung für Roboter geweckt und wieder in einem Team aus Mitschülern bestritt ich den Roboking 2006, bei dem Steine durch eine Hindernisstrecke zu einer Abladestation transportiert werden mussten. Alle Erfahrungen, die ich bei diesen Wettbewerben gemacht habe, stecken in dem Roboter für den Roboking 2007, von dem auch diese Facharbeit handelt. Die Aufgabe dieses Robokings, dessen Vorrunde schon im November 2006 stattfand, war es, die eigene Spielfeldhälfte von Tischtennisbällen zu leeren. Dazu konnte man die Tischtennisbälle entweder in einen Korb in der eigenen Hälfte abladen oder sie durch zwei Tunnel auf die gegnerische Seite befördern.
Im Folgenden werde ich zuerst eine genauere Aufgabenbeschreibung geben und meine Lösungsstrategie beschreiben. Danach gehe ich näher auf den Aufbau und die Programmierung des Roboters ein.

2. WETTBEWERBSREGELN UND AUFGABENBESCHREIBUNG

Das Spielfeld[1] ist in zwei gespiegelt gleiche Hälften geteilt, die durch zwei niedrige Tunnel miteinander verbunden sind, so dass Tischtennisbälle zwischen den Spielfeldhälften ausgetauscht werden können. Die Startposition des Roboters befindet sich in der oberen Spielfeldhälfte („Anhöhe“) am Rand der „Schlucht“. Von dort aus muss der Roboter eine Rampe hinunterfahren um in den Bereich zu kommen in dem sich die Tischtennisbälle befinden. Vor jedem Lauf wird eins von sechs verschiedenen Szenarien zufällig bestimmt, welches die Positionen der Tischtennisbälle festlegt. Der Roboter kann die Tischtennisbälle entweder in der eigenen „Schlucht“ abladen, wozu er aber den gesamten Weg über die Rampe auf die Anhöhe fahren muss, oder aber er befördert die Tischtennisbälle durch einen der zwei Tunnel auf die gegnerische Seite. Insgesamt darf der Roboter aber nur maximal fünf Tischtennisbälle auf einmal befördern. Am Ende des zweiminütigen Laufes erhält das Team pro Tischtennisball auf der gegnerischen Seite oder in der eigenen „Schlucht“ zwei Punkte. Die offizielle Wettbewerbsdokumentation des Roboking 2007 kann unter [[2]] gefunden werden.

---

[1] Organisationsteam Roboking, *Roboking 2007 Spielfeld*. 20.06.2006, CD\Roboking\RoboKing2007_Spielfeld.pdf <http://www.tu-chemnitz.de/etit/proaut/rk/fileadmin/user_upload/downloads/RoboKing2007_Spielfeld.pdf>

[2] Organisationsteam Roboking, *Roboking 2007 Wettbewerbsdokumentation*. 20.06.2006, CD\Roboking\RoboKing2007_Wettbewerbsdoku.pdf <http://www.tu-chemnitz.de/etit/proaut/rk/fileadmin/user_upload/downloads/RoboKing2007_Wettbewerbsdoku.pdf>

3. LÖSUNGSSTRATEGIE

Bevor mit dem Bau des Roboters begonnen werden kann, wird eine Strategie zur Lösung der Aufgabe benötigt. Aus dieser Strategie ergeben sich unmittelbar konstruktionstechnische Besonderheiten die beim Bau des Roboters berücksichtigt werden müssen.
Die Lösungsstrategie sieht vor, dass der Roboter einen festen Kurs, der ihm über ein einprogrammiertes Koordinatensystem vorgegeben ist, abfährt und gefundene Tischtennisbälle sofort mittels eines Kickersystems auf die gegnerische Spielfeldhälfte schießt. Dadurch wird verhindert, dass es der Roboter vor Ablauf der zwei Minuten nicht mehr schafft die schon gesammelten Bälle auf die gegnerische Spielfeldhälfte zu schießen. Der Roboter soll keine Tischtennisbälle in die „Schlucht“ befördern, da der Weg, der jedes Mal zurückgelegt werden muss, zu viel Zeit benötigen würde.
Der Kurs des Roboters soll so vorgegeben werden, dass er möglichst die gesamte Fläche abdeckt, vor allem aber so, dass der Kurs alle Plätze abdeckt, auf denen sich mit hoher Wahrscheinlichkeit ein Tischtennisball befindet. Um diese Wahrscheinlichkeiten zu bestimmen, wurden am Tag des Wettbewerbs für alle möglichen Positionen eines Tischtennisballs die Anzahl von Szenarien gezählt, in denen ein Tischtennisball an dieser Position liegen würde. Je mehr Szenarien für eine bestimmte Lage gezählt wurden, desto wahrscheinlicher ist es an dieser Position einen Tischtennisball zu finden.

4. MECHANISCHER AUFBAU DES ROBOTERS

Die Grundplatte des Roboters ist aus Plexiglas und weist an der Frontseite für den Kicker eine neun Zentimeter breite und fünfzehn Zentimeter tiefe Einbuchtung auf. An dieser sind beidseitig der Einbuchtung zwei Plexiglasplatten senkrecht befestigt, zwischen denen die Controllerboards, das LCD, die Spannungsversorgung, die Motorenkontrollplatine und die Motoren angebracht sind. Die Kraft der Motoren wird jeweils über deren Stirnradgetriebe, einen Zahnriemen und eine doppelt kugelgelagerte Welle auf die zwei Moosgummireifen übertragen. Die Gesamtuntersetzung des so entstandenen Getriebes beträgt 30:1. Außerdem dient ein antriebsloses Kugelrad als Stütze. Aufgrund der Kombination von zwei angetriebenen Rädern und einem Stützrad besitzt der Roboter drei Freiheitsgrade: Die x-und y-Richtung sowie die Orientierung θ, welche als Winkel zur x-Richtung angegeben wird. Außerdem wurden an zwei Plastikblöcken, die beidseitig außen an den senkrechten Plexiglasplatten angeschraubt sind, zwei Ultraschall-Abstandssensoren und ein Infrarot-Abstandssensor befestigt.

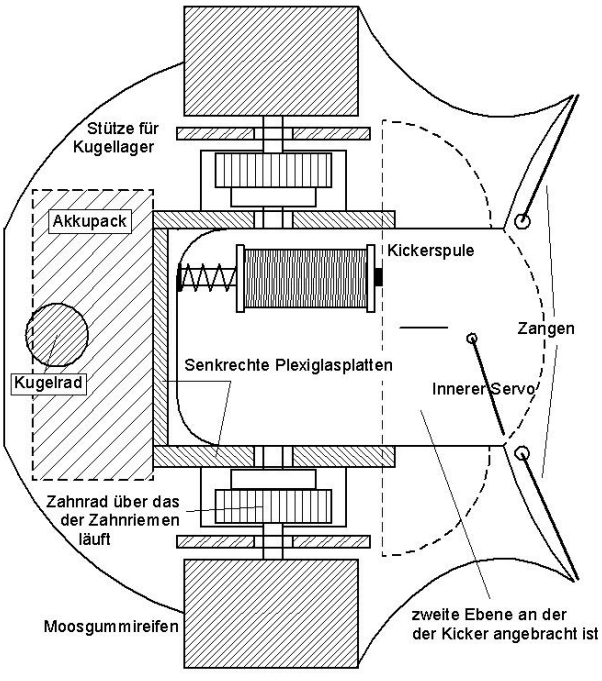

Abbildung 1

4.1. Die Antriebsmotoren

Die Antriebsmotoren sind nach hinten oben zwischen die senkrechten Plexiglasplatten versetzt worden, um zwischen den Rädern Platz für den Kicker zu schaffen. Die Verbindung der Motoren mit den Rädern wird über Zahnriemen hergestellt.
Wegen ihres hohen Wirkungsgrades und der integrierten Radencoder (siehe dazu 6.4 „Die Radencoder“) wurden Motoren der Firma Maxon gewählt. Durch den hohen Wirkungsgrad war es möglich die Motoren mit 6V zu betreiben. Das bedeutet Gewichtsersparnis, da so nur ein 6V Akkupack zur Spannungsversorgung benötig wird.

4.2. Die Servos (Stellmotoren)

Bei diesem Roboter werden Servos dazu verwendet die Zangen zu öffnen und zu schließen, einen Tischtennisball in die Abschussvorrichtung hineinzudrücken und den Abschussschalter zu betätigen. Servos sind vielfältig einsetzbare Stellmotoren. Sie zeichnen sich durch ihr hohes Drehmoment bei kompakter Bauform aus. Ein Servo besitzt eine innere Regelung, die über eine pulsweiten modulierte Rechteckspannung (PWM) angesteuert wird. Es wird also bei gleich bleibender Frequenz das Verhältnis von T_{on} zu T_{off} verändert. Über dieses Verhältnis wird die Stellung des Servos vorgegeben.

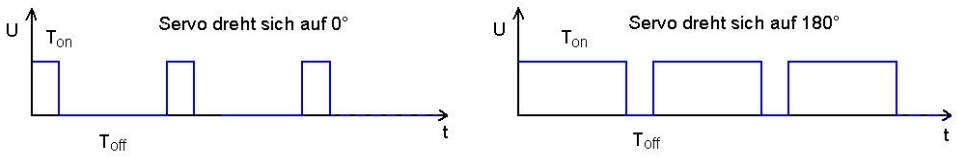

Abbildung 2

4.3. Der Greifmechanismus

Der Greifmechanismus sorgt dafür, dass ein Tischtennisball, der durch die äußere Lichtschranke erkannt wurde, sicher in die Abschussposition vor dem Kicker gelangt. Der Mechanismus besteht aus den Greifzangen (zwei Zangen in Höhe der Tischtennisbälle) die über ein Getriebe von einem Servo gesteuert werden und dem inneren Servo, an dem ein Drahtbügel zum Schieben der Tischtennisbälle angebracht ist. Außerdem dienen zwei Lichtschranken zur Lokalisation des Tischtennisballs während des Greifvorgangs.
In der ersten Phase sind die äußeren Zangen offen und der innere Servo gibt den Weg zum Kicker frei. Der Mechanismus wartet darauf, dass ein Tischtennisball die äußere Lichtschranke unterbricht. Sobald diese Lichtschranke unterbrochen wurde, werden die Zangen geschlossen und der Tischtennisball gesichert, sodass er nicht mehr wegrollen kann. Als nächstes schiebt der innere Servo den Tischtennisball vor den Kicker. Dort wird der Tischtennisball von einem gebogenen Drahtbügel seitlich fixiert, damit er nicht mehr davon rollen kann. Ist nun die innere Lichtschranke nicht unterbrochen, so ist ein Fehler unterlaufen und es wurde kein Ball gesammelt. In diesem Fall werden die Greifzangen und der innere Servo wieder geöffnet und der Mechanismus wartet wieder auf Tischtennisbälle. Falls aber die innere Lichtschranke unterbrochen wird, befindet sich ein Tischtennisball vor dem Kicker und kann abgeschossen werden. Deshalb wird die Fahrt an dieser Stelle unterbrochen und der Roboter dreht sich in die Abschussrich-

tung. Währenddessen werden die Zangen geöffnet und der innere Servo dreht sich wieder in die Anfangsposition zurück. Sobald der Roboter seine Drehung beendet hat erfolgt der Abschuss des Tischtennisballs.

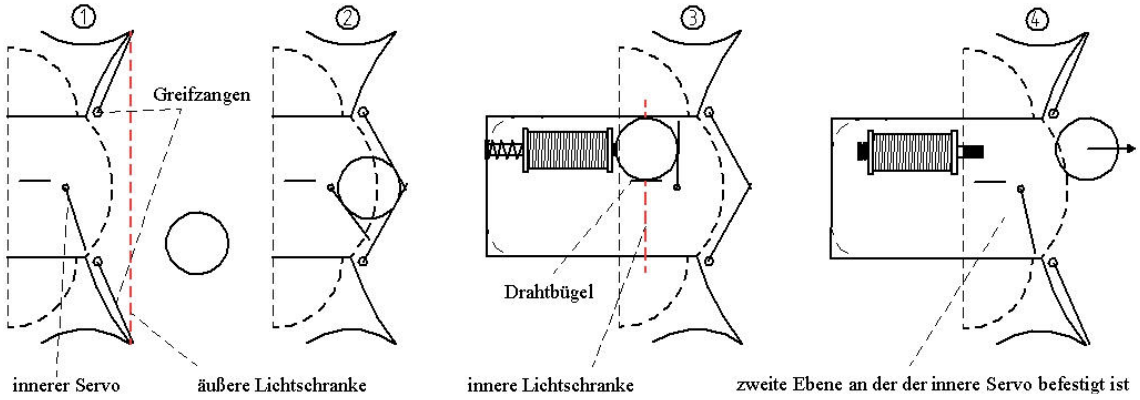

Abbildung 3

Auf der beigelegten Film-DVD ist der Greifmechanismus und der Kicker in Aktion zu sehen.

5. ELEKTRONISCHE KOMPONENTEN DES ROBOTERS

Alle Platinen wurden mit dem Schaltplan und Layout Programm Target 3001 entworfen und dann eigens geätzt, gebohrt, bestückt und gelötet. Die Schaltpläne und Layouts sind als pdf-Dateien auf der CD im Ordner „Platinen“ zu finden.

Eine Übersicht über die Verknüpfung der elektronischen Komponenten und der Sensoren des Roboters ist in Abbildung 4 zu sehen.

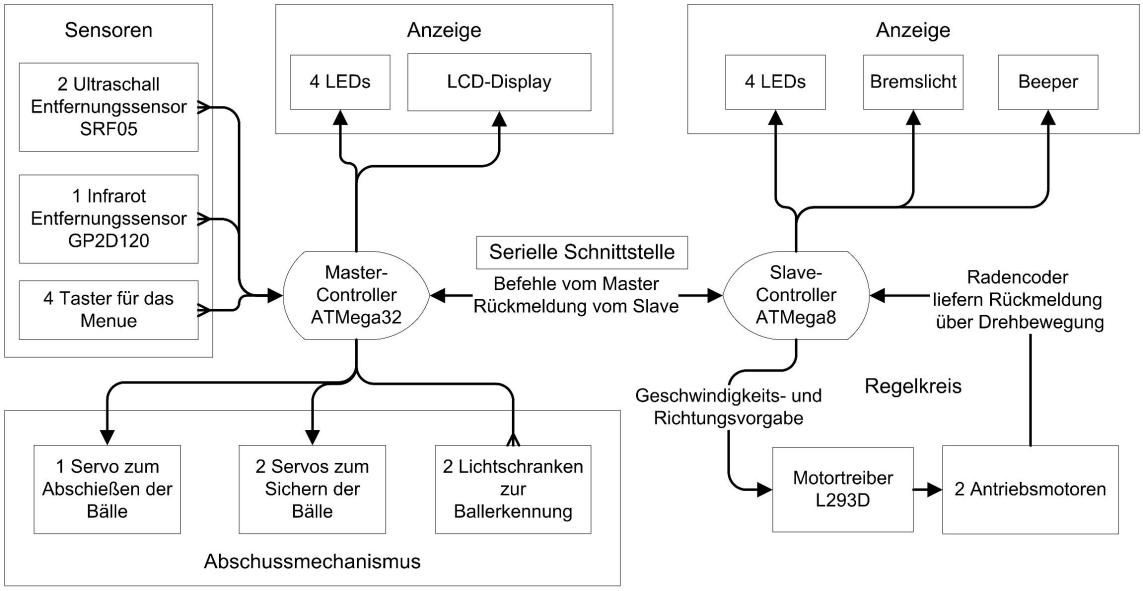

Abbildung 4

Die Stromversorgung

Primär wird der Roboter durch einen aus fünf Nickel Metallhydrid Zellen bestehenden 6V 3500mAh Akkupack mit Strom versorgt. Die Wahl fiel auf diesen Akkupack, da Nickel Metallhydrid Zellen keinen Memory-Effekt aufweisen und schnellladefähig sind.
Die Spannung wurde mit 6V minimal gewählt, um die Verlustleistung des Festspannungsreglers für die 5V der Controllerboards so gering wie möglich zu halten. Der verwendete Festspannungsregler L78S05[3] regelt die Spannung auf konstante 5V herunter. Die Differenzspannung von ca. 1V fällt über dem Regler ab und sorgt bei höheren Strömen (>1A) für ein starkes Erhitzen des ICs. Deshalb wird der IC durch einen Kühlkörper passiv gekühlt.
Für die Spannungsstabilisierung sorgen 100nF Kondensatoren in der Nähe der Mikrocontroller.

5.1. Der Entwurf der Controllerboards

Das Zentrum dieses Roboters bilden zwei über die serielle Schnittstelle verbundene Mikrocontroller der Firma Atmel, auf die die Rechenarbeit aufgeteilt wird. Die beiden ausgewählten 8-bit Controller ATmega8[4] und ATmega32[5] unterscheiden sich hauptsächlich in der Anzahl von Input/Output Pins und der Größe der programmierbaren Flash-Speicher. Der innere Aufbau beider Controller ist identisch. Die Controller-Architektur ist in den Datenblättern genauer beschrieben. Mit dem ATmega32, dem Master-Controller, wird die Logik, die Pfadplanung und das Auswerten der Sensoren realisiert, da dieser Controller ausreichend I/O Pins für den Anschluss von allen Sensoren, dem LCD, den LEDs und der Servos besitzt. Wegen der zeitkritischen Motorenregelung war ein zweiter Controller notwendig. Hier kommt der ATmega8 zum Einsatz, da für diese Aufgabe neben vier LEDs nur die Motorenkontrollplatine und die Radencoder an den Controller angeschlossen werden müssen und dementsprechend weniger Pins benötigt werden. Aus diesen Vorgaben entstanden die Schaltpläne beider Controllerbords

5.1.1. Das Master-Controllerboard

Da an das Master-Controllerboard[6] alle Sensoren angeschlossen werden sollen, wurden alle acht Analogeingänge, die auch als digitale Ein- und Ausgänge verwendbar sind, über universelle dreipolige Anschlusssteckerleisten herausgeführt. Diese bestehen jeweils aus dem herausgeführten Controllerpin und einer 5V Spannungsversorgung für den Sensor. An diese Anschlüsse sind sowohl die digitalen Ausgänge der Lichtschranken als auch der analoge Ausgang des Infrarotsensors angeschlossen. Für die Servos wurden drei weitere gleichartig beschaltete Anschlussstecker vorgesehen. Außerdem existiert auf dem Board ein Anschluss für das LCD, ein Anschluss für die Serielle Schnittstelle, drei Anschlüsse für den I2C Bus, ein Anschluss für die Motorenkontrollplatine, und zwei Anschlüsse für Ultraschallsensoren, die jeweils aus zwei her-

[3] SGS-Thomson microelektronics, *L78S00 Series*. 29.12.2006, CD\datasheets\L78S05.pdf<http://www.datasheetarchive.com/datasheet.php?article=1969221>
[4] Atmel, *Datenblatt des ATmega8*. 16.12.2006, CD\Datenblätter\ATmega8.pdf <http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf>
[5] Atmel, *Datenblatt des ATmega32*. 16.12.2006, CD\Datenblätter\ATmega32.pdf <http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf>
[6] Schaltplan und Layout des Master-Controllerboards: CD\Platinen\Mastercontrollerboard_Schaltplan.pdf CD\Platinen\Mastercontrollerboard_Layout.pdf

ausgeführten I/O Pins und einer 5V Spannungsversorgung bestehen. Die vier LEDs sind an die Pins für die Motorenkontrollplatine angeschlossen, da diese nicht verwendet werden.

5.1.2. Das Slave-Controllerboard

Das Slave-Controllerboard[7] wurde drehbar gelagert anmontiert, damit es leicht hochgeklappt werden kann. Dadurch bleiben die darunter befindlichen Anschlüsse des Master-Controllerboards zugänglich.
Da der Slave-Controller für die Motorregelung zuständig ist besitzt das Board einen Anschluss für die Motorenplatine und vier Pins für die Radencoder. Außerdem sind Anschlüsse für die Serielle Schnittstelle und den I2C Bus herausgeführt. Als Anzeigeelemente stellt das Slave-Controllerboard vier verschiedenfarbige LEDs zur Verfügung. Über diese lässt sich der aktuelle Zustand des Slaves ablesen. Damit der Roboter auch akustisch auf sich aufmerksam machen kann, besitzt das Board einen Piezo-Beeper. Die übrigen drei I/O Pins wurden mit denselben universalen Anschlusssteckern wie beim Master-Controllerboard herausgeführt.

5.2. Die Platine zur Ansteuerung der Motoren

Die Elektronik für die Ansteuerung der Motoren wurde aus Platzgründen auf eine eigene Platine[8] ausgelagert, die in der Nähe der Motoren montiert ist.
Für die Bewegungen des Roboters, wie Drehungen, Vor- und Rückwärtsfahren, müssen die Antriebsmotoren sowohl in Drehrichtung als auch in der Rotationsgeschwindigkeit kontrollierbar sein. Der IC (integrated circuit) L293D[9] stellt diese Steuermöglichkeiten für zwei Motoren zur Verfügung. Er ist für 600mA Dauerstrom pro Motor ausgelegt und besitzt integrierte Freilaufdioden. Diese leiten die beim Leerlauf der Motoren induzierten Ströme ab, damit die Elektronik nicht zerstört wird.
Die Drehrichtungen der Motoren werden über die Zustände von je zwei Eingängen C und D (siehe Abbildung 5) festgelegt. Durch pulsweitenmodulierte Rechteckspannungen an den Enable-Eingängen (in Abbildung 5 als VINH bezeichnet) des ICs werden die Geschwindigkeiten separat gesteuert. Im Prinzip wird dem jeweiligen Motor dabei immer nur kurz, während der Enable-Eingang auf „high“ liegt, Strom zugeführt. Liegt der Enable-Eingang auf „low“ fliest kein Strom durch den Motor, er dreht sich frei weiter.

INPUTS		FUNCTION
VINH = H	C = H; D = L	Turn Right
	C = L; D = H	Turn Left
	C = D	Fast Motor Stop
VINH = L	C = X; D = X	Free Running Motor Stop

L = Low H = High X = Don't Care

Abbildung 5

[7] Schaltplan und Layout des Slave-Controllerboards: CD\Platinen\Slavecontrollerboard_Schaltplan.pdf
CD\Platinen\Slavecontrollerboard_Layout.pdf
[8] Schaltplan und Layout der Motorenkontrollplatine: CD\Platinen\Motorenplatine_Schaltplan.pdf
CD\Platinen\Motorenplatine_Layout.pdf
[9] Unitrode, *Push-Pull Four Channel Driver – L293D.* 29.12.2006, CD\Datenblätter\L293D.pdf
<http://www.datasheetarchive.com/datasheet.php?article=3898324>

5.3. Das Entstören der Motoren

Durch plötzliches Blockieren oder schnelle Richtungsänderungen der Motoren können starke Spannungsschwankungen in der Versorgungsspannung entstehen, die im Extremfall zum Reset eines Controllers führen können.
Das Eliminieren solcher Spannungsschwankungen, auch Entstören genannt, erfolgte dadurch, dass die Zuleitungen beider Motoren mehrmals durch einen Ferritringkern gewickelt wurden. Die so entstandenen Spulen verzögern ein starkes Ansteigen des Motorstroms und dämpfen so die Spannungsschwankungen.
Durch diese einfache Methode konnten Controller-Resets erfolgreich vermieden werden.

5.4. Der Kicker

Der Kicker ist ein Schussmechanismus für Tischtennisbälle, der zwischen den Radachsen des Roboters angebracht ist. Er besteht aus einer Spule mit beweglichem Eisenkern und einer Ladeschaltung. Bei einem Schuss wird ein Kondensator über die Spule kurzgeschlossen. Wegen dem hohen Strom durch die Spule baut sich in ihr ein starkes Magnetfeld auf, das den beweglich im Inneren der Spule gelagerten Eisenbolzen beschleunigt.
Experimente haben gezeigt, dass sich der Eisenbolzen vor dem Schuss nur halb in der Spule befinden darf, um einen harten Schuss zu erhalten. Deshalb wird der Eisenkern durch eine schwache Feder nach dem Abschuss wieder aus der Spule gezogen. Außerdem zeigte sich, dass der Eisenbolzen nur in die Spule hinein, aber nicht aus der Spule heraus beschleunigt, sondern vielmehr abgebremst wird. Aus diesem Grund war es nötig am vorderen Ende des Eisenbolzens einen Plastikzylinder aufzukleben, um auf den Tischtennisball die maximale Beschleunigung zu übertragen.

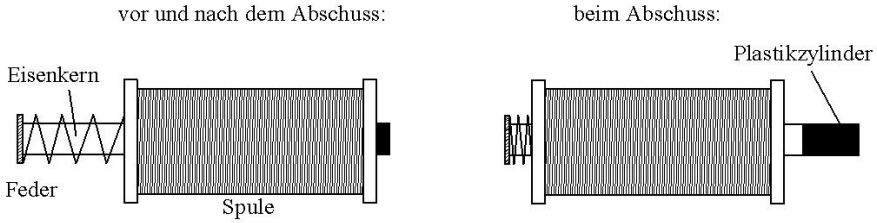

Abbildung 6

5.4.1. Die Ladeschaltung

Die Ladeschaltung, die mit einer 1,5V Batterie betrieben wird, lädt in ca. 18 Sekunden, einen Kondensator auf 330V auf, der die benötigten hohen Ströme für ein starkes Magnetfeld in der Spule bereitstellt. Die hohe Spannung ist wegen des ohmschen Widerstandes der Spule, von ungefähr 50Ω, nötig, der den Strom sonst auf zu geringe Werte begrenzen würde.

Bei 330V fliesen so kurzzeitig immerhin $I = \frac{U}{R} = \frac{330V}{50\Omega} = 6{,}6A$.

Die Ladeschaltung wurde aus einer Wegwerfkamera ausgebaut, in der sie ein Kondensator auf 330V auflädt, welcher dann für den Blitz über eine Blitzlichtbirne kurzgeschlossen wird. Statt den Kondensator über die Blitzlichtbirne zu entladen wird er, mittels eines Schalters, über die Abschussspule kurzgeschlossen. Um eine galvanische Trennung der Ladeschaltung von den

Controllerboards zu erreichen, wird der Schalter von einem Servo betätigt. So wird vermieden, dass sich der Kondensator durch einen Schaltungsfehler oder einen Unfall über die Controllerboards entlädt und diese damit zerstört.

6. SENSORIK DES ROBOTERS

Sensoren sind die wichtigsten Komponenten eines autonomen Roboters. Sie geben dem Roboter laufend aktuelle Informationen über seine Umwelt, auf die er entsprechend seiner Programmierung reagieren kann. Bei diesem Roboter kommen zwei Ultraschallsensoren, ein Infrarotabstandssensor, zwei Lichtschranken sowie zwei Radencoder zum Einsatz.
Die Funktionsweisen der verschiedenen Sensoren werden im Folgenden näher beschrieben.

6.1. Die Ultraschall-Abstandssensoren

Am Roboter sind frontal, im Abstand von 13,5cm, zwei Ultraschallsensoren, des Typs SRF05, angebracht. Sie sind beide in Fahrtrichtung gerichtet und dienen der Erkennung von Hindernissen und der Bestimmung der Entfernung zu diesen. Durch den für Ultraschallsensoren charakteristischen, kegelförmigen Erfassungsbereich decken die beiden SRF05 den gesamten Bereich vor dem Roboter ab, sodass kein Hindernis übersehen werden kann. Des Weiteren kann sich der Roboter mit ihrer Hilfe senkrecht zu einer Bande ausrichten und sogar anhand einer Ecke seine Position bestimmen (siehe 7.4.1.1 und 7.4.1.2). Mit den SRF05[10] Ultraschall-Abstandssensoren lässt sich der Abstand zu einem Hindernis in bis zu vier Metern Entfernung durch die Laufzeitmessung eines Ultraschallimpulses sehr präzise messen. Im Datenblatt unter „Mode 2 - Single pin for both Trigger and Echo“ kann Genaueres zur Laufzeitmessung mit dem SRF05 nachgelesen werden. Aus der gemessenen Laufzeit lässt sich über die Schallgeschwindigkeit in Luft c=331m/s [11] der Abstand zu dem Hindernis berechnen: $x = \frac{331\frac{m}{s} \cdot t}{2}$

Um verfälschte Messungen zu verhindern, wird nach jeder Messung fünfzig Millisekunden gewartet, damit keine Ultraschallimpulse von der vorigen Messung detektiert werden.

6.1.1. Die Justierung der Ultraschallsensoren

Berechnet man die Entfernung zum Hindernis aus der gemessenen Laufzeit des Ultraschallimpulses und der Schallgeschwindigkeit in Luft erhält man leider noch keine exakte Entfernung. Dies erklärt sich durch Bauteileigenschaften, ungenaue Taktung des Mikrocontrollers und Verzögerungen beim Starten und Stoppen der Zeitmessung.
Um diese Abweichungen korrigieren zu können wurde eine Kennlinie des Sensors aufgenommen. Auf der x-Achse ist die vom Controller aus der Laufzeit des Ultraschallimpulses berechnete Entfernung angetragen und die y-Achse ist mit dem tatsächlichen mit einem Meterstab gemessenen Abstand zum Hindernis beschriftet.

---

[10] Jörg, Pohl. *SRF05 - Ultra-Sonic Ranger*. 03.01.2007, CD\Datenblätter\SRF05.pdf <http://www.roboter-teile.de/datasheets/srf05.pdf>
[11] Hammer/Hammer: *Pysikalische Formeln und Tabellen*. München: J. Lindauer Verlag (Schaefer), 2002. S. 87

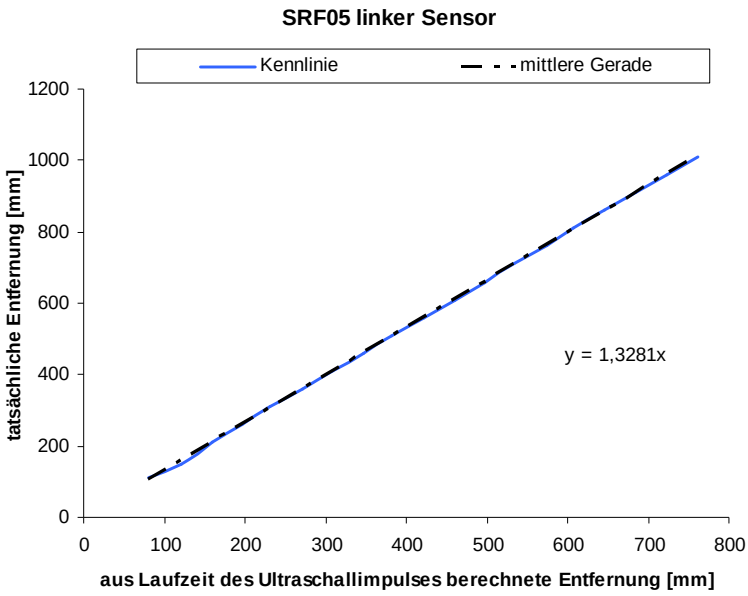

Abbildung 7

Anhand der Kennlinie aus Abbildung 7 sieht man sehr schön, dass eine direkte Proportionalität zwischen tatsächlicher Entfernung und berechneter Entfernung besteht.

Durch Einsetzen von $x = \frac{331\frac{m}{s} \cdot t}{2}$ in die Gleichung der mittleren Geraden $y = 1{,}3281 \cdot x$ erhält man für den SRF05 eine spezifische Formel zur Berechnung der tatsächlichen Entfernung y zu einem Hindernis aus der Laufzeit t des Ultraschallimpulses: $y = \frac{331\frac{m}{s} \cdot t}{2} \cdot 1{,}3281$

Durch diesen Korrekturfaktor erhält man bei der Entfernungsmessung Genauigkeiten von ±4mm

6.2. Der Infrarot-Abstandssensor

Ein GP2D120[12] Infrarot-Abstandssensor von Sharp ist in Fahrtrichtung links am Roboter angebracht. Mit ihm wird ständig der seitliche Abstand gemessen, um verhindern zu können, dass der Roboter seitlich mit einem Hindernis kollidiert. Der Sensor misst im Nahbereich von 4 bis 30cm die Entfernung mittels Triangulation[13]. Bei einer Abstandsmessung wird innerhalb von ca. 39 Millisekunden 32 mal ein Lichtimpuls ausgesendet und jedes mal gemessen, wo das reflektierte, durch eine Linse gebündelte, Licht auf dem Position Sensitive Device (PSD) auftrifft. Die Funktionsweise eines PSD ist in [14] beschrieben.

Aus den 32 Messungen wird auf die Entfernung zum Hindernis

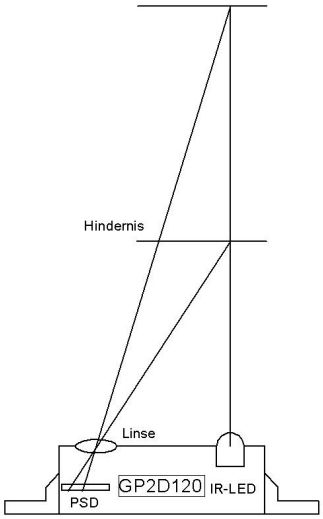

Abbildung 8

[12] Sharp, *GP2D120 Optoelectronic Device*. 03.01.2007, CD\Datenblätter\GP2D120.pdf <http://document.sharpsma.com/files/GP2D120-DATA-SHEET.PDF>

[13] Sharp, *chapter 12 - Use of Optical Sensor Units*: *"12.2.1 Principle of operation of distance measuring sensors",* 04.01.2007, CD\Datenblätter\Triangulation.pdf <http://document.sharpsma.com/files/Optical_Sensors_AN.pdf>

[14] Sharp, *chapter 5 – Use of Light Detecting Devices: "5.5.1 Principle of position sensors",* 04.01.2007, CD\Datenblätter\PSD.pdf <http://document.sharpsma.com/files/Light_Detecting_Device_AN.pdf>

geschlossen und vom GP2D120 eine analoge Ausgangsspannung erzeugt. Diese Ausgangsspannung ist leider nicht proportional zur Entfernung zum Hindernis.

6.2.1. Die Linearisierung des Infrarotsensors

Um die genaue Abhängigkeit der Ausgabespannung des GP2D120 von der Entfernung zum Hindernis zu bestimmen wurde eine Kennlinie des Sensors aufgenommen. Dabei ist auf der y-Achse der Ausgabewert des Analogeingangs und auf der x-Achse die Entfernung zum Hindernis angetragen. Der 10bit Ausgabewert des Analogeingangs wird durch Vergleichen der Ausgangsspannung des GP2D120 mit einer Referenzspannung von 2,5V bestimmt und berechnet sich zu: $y = \frac{U_{GP2D120}}{2{,}5V} \cdot 1024$

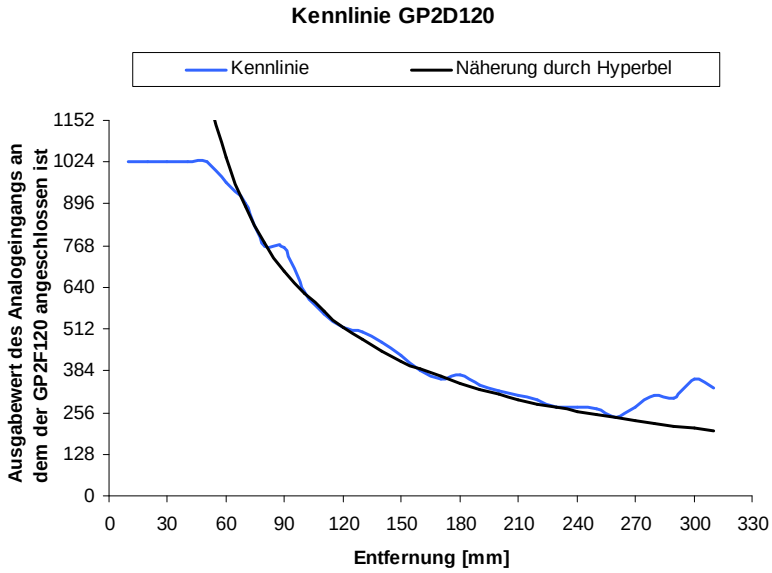

Abbildung 9

Anhand der blauen Kennlinie aus Abbildung 9 kann man eindeutig erkennen, dass die Entfernung in keinster Weise proportional zur ausgegebenen Spannung ist. Um von der Spannung auf Entfernung schließen zu können, muss die Kennlinie zuerst durch eine Funktion angenähert werden. Eine Näherung der Kurve durch Excel ist nicht hilfreich, da Excel in diesem Fall Funktionen mit reellen Exponenten ausgibt, die sich in dem Controller nicht so einfach berechnen lassen. Deshalb wurde die Kennlinie durch eine Hyperbel mit der allgemeinen Funktionsgleichung $y = \frac{a}{x-b}$ angenähert. Die Unbekannten a und b erhält man aus zwei Messungen:

$$\left.\begin{array}{ll}(I) & a = y_1 \cdot (x_1 - b) \\ (II) & a = y_2 \cdot (x_2 - b)\end{array}\right\} \Rightarrow [15]\ a = \frac{y_1 y_2 (x_1 - x_2)}{y_2 - y_1} \text{ und } b = \frac{y_2 x_2 - y_1 x_1}{y_2 - y_1}$$

Die Funktion des in Abbildung 9 abgebildeten Näherungsgraphen wurde über die Messwerte

$$\left.\begin{array}{l}x_1 = 100; y_1 = 627 \\ x_2 = 260; y_2 = 240\end{array}\right\} \Rightarrow a \approx 62210;\ b \approx 0 \text{ zu } y = \frac{62210}{x} \text{ bestimmt.}$$

[15] Siehe 11.1 „Herleitungen“ im Anhang

Um nun für jeden Ausgabewert des Analogeingangs, an dem der Infrarotsensor angeschlossen ist, die Entfernung zu erhalten, wird die Umkehrfunktion gebildet: $x = \frac{a}{y} + b$ bzw. $x = \frac{62210}{y}$

Problematisch sind bei der Infrarotabstandsmessung die Analog-Eingänge des Controllers. Deren theoretische Auflösung ist zwar mit 10bit sehr hoch aber die Ausgabewerte schwanken von Messung zu Messung erheblich, was faktisch zu einer geringeren Auflösung führt. Da die Steigung der Kennlinie bei großen Entfernungen zu gering wird, können die Analogpins diese unterschiedlichen Abstände nicht mehr auflösen. Eine Entfernungsmessung ist daher nur im Bereich von 4 bis 24cm sinnvoll. Um aussagekräftige Messungen zu erhalten, wird außerdem immer der Mittelwert der Entfernungen aus vier Messungen verwendet.

6.3. Die Lichtschranken

Der Roboter besitzt zwei Lichtschranken. Die äußere Lichtschranke ist so angebracht, dass sie unterbrochen wird, wenn ein Ball in Reichweite der Greifzangen ist.
Ein Tischtennisball, der sich in Abschussposition befindet, unterbricht die innere Lichtschranke.
Die Lichtschranken bestehen jeweils aus einer Sende IR-LED und dem IS471[16]. Der IS471 moduliert die IR-LED mit ca. 8kHz. Dadurch wird die Lichtschranke unempfindlich gegen das Umgebungslicht, da der Empfänger des IS471 nicht ein „AN“ oder „AUS“ der Sendediode detektiert, sondern ein „gepulstes Licht AN“ oder „gepulstes Licht AUS“.

6.4. Die Radencoder

An beiden Antriebsmotoren sind Radencoder angebracht, über die sowohl die Geschwindigkeit der Motoren als auch die Anzahl von Umdrehungen gemessen werden kann. Diese beiden Messungen sind für die Wegstreckenmessung und somit für den Aufbau eines Koordinatensystems unerlässlich. (Siehe dazu 7.2.1 und 7.2.2) Die verwendeten Radencoder bestehen aus einer Lichtschranke und einer durchsichtigen Scheibe, auf der in kleinen Abständen, schwarze (undurchsichtige) radiale Striche aufgebracht sind.
Dreht sich die Scheibe, die an der Motorwelle angebracht ist, durch die Lichtschranke so wird diese periodisch unterbrochen. Das daraus resultierende elektrische Rechtecksignal steht am Ausgang des Radencoders zur Verfügung.
Die verwendeten Radencoder „Enc 22“[17] geben pro Motorumdrehung 100 Impulse aus. Wegen der Untersetzung von 30:1 werden also pro Radumdrehung 3000 Impulse gezählt.
Die Radencoder stellen zwei Ausgänge A und B, die in der Phase um 90° verschoben sind, zur Verfügung. Aus diesen zwei Signalen lässt sich die Drehrichtung des Motors bestimmen: Je nachdem ob sich der Motor links oder rechts herum dreht, eilt Signal A oder Signal B um 90° voraus. Betrachtet man Signal B bei steigender Flanke von Signal A, so ist Signal B, je nach Drehrichtung, gerade entweder „high“ oder „low“. (siehe Abbildung 10 auf der nächsten Seite)

---

[16] Sharp, *IS471F*. 03.01.2006, CD\Datenblätter\IS471F.pdf
<http://www.roboter-teile.de/datasheets/IS471F.pdf>
[17] Maxon motor, *Encoder Enc 22, 100 Impulse, 2 Kanal*. 09.01.2007, CD\Datenblätter\Enc 22.pdf
<http://test.maxonmotor.com/docsx/Download/catalog_2006/Pdf/06_248-249_d.pdf>

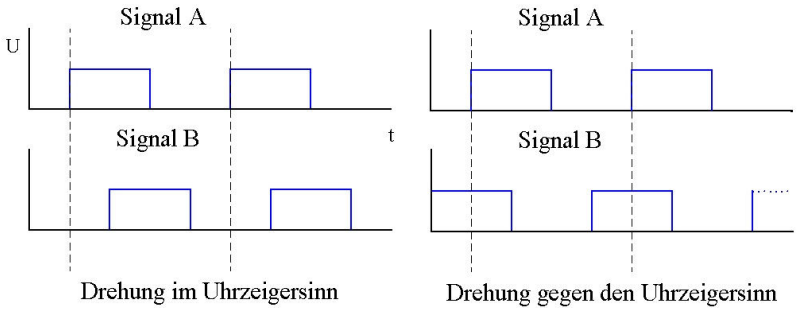

Abbildung 10

Während die Ausgänge A der zwei Radencoder jeweils an einen Interrupt Pin des Slave-Controllers angeschlossen sind, genügen für die Ausgänge B der Radencoder zwei normale Input-Pins. In der Software wird jedes Mal bei steigender Flanke von Signal A ein Interrupt ausgelöst. In der jeweiligen Interruptroutine wird die Drehrichtung aus dem Zustand des entsprechenden Ausgangs B des Radencoders ermittelt und je nach Drehrichtung die dazugehörige Streckenvariable in- bzw. decrementiert.

7. SOFTWARE DES ROBOTERS

Die Programmierung erfolgt am PC über den Opensource C Compiler Winavr, der den C-Code in Maschinencode umwandelt. Dieser wird dann über einen Adapter mit dem Programm PonyProg2000 auf die Controller des Roboters geladen.
Ein Programm besteht aus einer Hauptroutine, die immer nach einem Reset ausgeführt wird, und mehreren Unterprogrammen und Funktionen. Außerdem werden noch Interruptroutinen definiert, die bei einem bestimmten Ereignis das Hauptprogramm unterbrechen, um ihren Code auszuführen. Im Folgenden werden alle wichtigen Funktionen und Unterprogramme der beiden Controller beschrieben. Der Quellcode für Master- und Slave-Controller ist auf der CD im Ordner „Quellcode“ zu finden.

7.1. Die Schnittstelle zwischen den Controllern

Master- und Slave-Controller sind über die serielle RS232 Schnittstelle und über eine extra Signalleitung verbunden. Über die Signalleitung kann der Master jederzeit ablesen, ob der Slave gerade einen Befehl ausführt oder nicht.
Der Datenaustausch über die Serielle Schnittstelle erfolgt in einem vorgegebenen Datenpaket mit einer Größe von vier Bytes, welche sowohl Vektordaten als auch andere Befehle enthalten können. Mögliche Befehle sind das Abbrechen der Bewegung, die der Slave gerade ausführt, das Abfragen des zu letzt abgefahrenen Vektors, sowie das Geradeausfahren ohne Regler.
Bei einem Vektordatenpaket enthält das erste Byte die Drehrichtung, das zweite den Winkel in Grad, um den gedreht werden soll, und das dritte und vierte die Länge in Millimetern, um die dann geradeaus gefahren werden soll. Die Länge wird vom Slave aus den beiden letzten Bytes zusammengesetzt und kann somit maximal $2^8 \cdot 2^8 = 2^{16}$ mm betragen. Außerdem existiert ein Befehl, der den Slave um eine bestimmte Länge rückwärts fahren lässt.
Das Senden und Empfangen von Bytes wird auf beiden Controllern von je zwei Interrupthandler übernommen. Der Vorteil der Interupthandler ist, dass empfangene Bytes sofort in einen Puffer

geschrieben werden und somit keine Bytes verloren gehen können. Auch das Senden von Bytes ist sehr einfach, da zu sendende Bytes nur in einen Puffer geschrieben werden müssen, aus dem sie die Interruptroutine verschickt.

7.2. Der Slave-Controller

Die Hauptroutine des Slave-Controllers überprüft in einer Endlosschleife ständig, ob Zeichen über die serielle Schnittstelle empfangen wurden. Währenddessen werden die LEDs wie ein Lauflicht angesteuert. Sobald ein Datenpaket empfangen wurde, wird der entsprechende Befehl ausgeführt. Danach wartet der Slave wieder auf Befehle vom Master.

7.2.1. Das Abfahren von Vektoren

Jeder Bewegungsvektor wird vom Master-Controller in die Richtung θ, in die sich der Roboter drehen muss, und die Länge l der geraden Strecke, die er in diese Richtung fahren muss, umgewandelt. Da der Roboter sich nur in zwei Dimensionen bewegt, ist die Umformung eines Vektors $\vec{u}$ sehr einfach: $\vec{u} = \begin{pmatrix} x \\ y \end{pmatrix} \Rightarrow \theta = \arctan\left(\frac{y}{x}\right)$ und $l = \left|\vec{u}\right| = \sqrt{x^2 + y^2}$

Der Winkel δ um den sich der Roboter ausgehend von seiner aktuellen Orientierung $\theta_{Aktuell}$ drehen muss, um die berechnete Richtung θ zu erreichen, ergibt sich aus $\delta = \theta - \theta_{Aktuell}$.

Diesen Winkel δ und die Länge l des Vektors sendet der Master dann dem Slave in einem Vektordatenpaket. Der Slave rechnet nun die Länge l und den Winkel δ in die Anzahl von Radencoderimpulsen um, die die Regelschleife der Motoren zählen muss. Da pro Radumdrehung 3000 Impulse von den Radencodern ausgegeben werden und sich der Roboter dabei um den Radumfang $U = d\pi$ vorwärts bewegt, berechnen sich die nötigen Impulse x für das Abfahren einer Strecke der Länge l zu: $x = \frac{3000}{d\pi} \cdot l$ mit $d = 68mm$ folgt: $x = 14{,}04 \cdot l$.

Bei einer Drehung des Roboters bewegen sich die Räder auf einem Kreis dessen Durchmesser der Abstand der Räder ist. Man kann also die Strecke und mit Hilfe obiger Formel die Anzahl von Impulsen berechnen, die die Räder zurücklegen müssen, damit sich der Roboter um einen bestimmten Winkel dreht: $x = 14{,}04 \cdot \frac{D\pi}{360°} \cdot \delta$ mit $D = 220mm$ folgt $x = 26{,}96 \cdot \delta$.

Um also einen Vektor abzufahren dreht sich der Roboter zuerst um den Winkel δ auf die berechnete Richtung θ und fährt anschließend eine Strecke der Länge l geradeaus. Dazu wird jeweils die Motorregelschleife, mit den erforderlichen Motorendrehrichtungen und der nötigen Anzahl von Impulsen, gestartet.

7.2.2. Die Regelung der Motoren

Auf Grund von Ungleichheiten beider Motoren, sowie unterschiedlich starker Reibung an den beiden Antrieben, ist eine Regelung nötig, damit bei Drehungen oder beim Geradeausfahren beide Motoren präzise mit der gleichen Geschwindigkeit drehen. Im Slave-Controller wurde deshalb eine dreifache Regelung realisiert. Je ein Regler sorgt dafür, dass sich der jeweilige Motor in einer vorgegebenen Geschwindigkeit unabhängig von der Last (im Rahmen der Kraft des Motors) dreht. Ein dritter Regler regelt die Differenz der Geschwindigkeiten beider Motoren auf Null. Dieser dritte Regler ist zum Beispiel beim Anfahren nötig, da dieses, aufgrund von unterschiedlichen Beschleunigungen der Motoren, sonst nicht geradlinig erfolgen würde.
Das Schema der dreifachen Regelung ist in Abbildung 11 graphisch dargestellt:

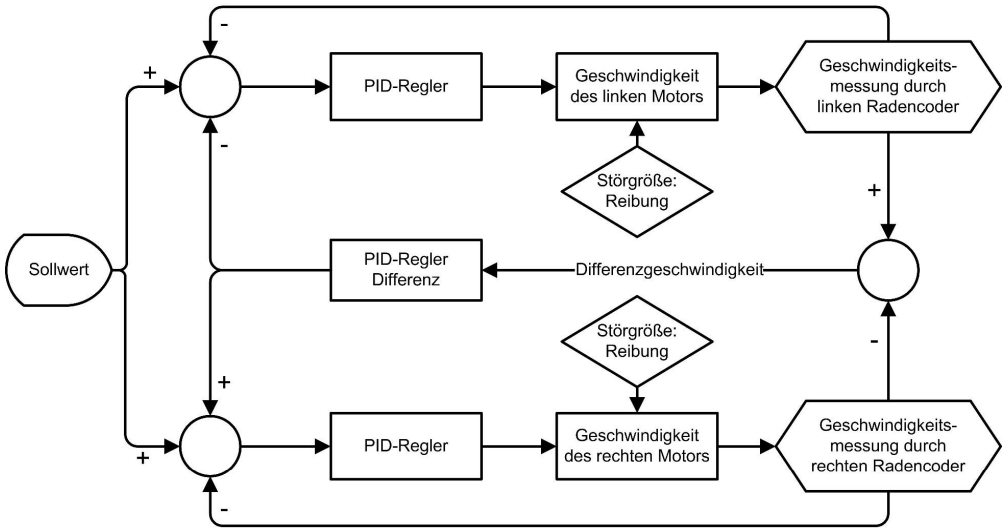

Abbildung 11

Damit die Regler präzise arbeiten, muss die Regelung möglichst oft pro Zeiteinheit und immer in gleichen Zeitabständen erfolgen. Währenddessen wäre es nicht möglich Sensoren abzufragen, da dies meist mehrere hundertstel Sekunden pro Messung benötigt. Deshalb wurde die Regelung der Motoren auf einen eigenen Controller, den Slave, verlagert, wo einmal in der Millisekunde die Stellwerte für die Motoren neu berechnet werden.
Die implementierten PID-Regler berechnen den jeweiligen Stellwert R nach der in [18] beschriebenen Formel $R(t) = K_P \cdot e(t) + K_I \cdot \int e(\tau) d\tau + K_D \cdot \frac{de(t)}{dt}$. Dabei ist $e(t)$ die Abweichung des Istwerts vom Sollwert, also zum Beispiel die Abweichung der tatsächlichen Geschwindigkeit von der Vorgegebenen. Bei einem PID-Regler wird die Abweichung des Istwerts vom Sollwert proportional, integral und differenziell zur Berechnung des Stellwerts herangezogen.
Diese drei Anteile finden sich in obiger Formel als Summanden, die durch die drei Konstanten K_P, K_I, K_D unterschiedlich gewichtet werden können. Genaueres zu der Funktion der einzelnen Summanden kann in [18] nachgelesen werden. Durch Experimentieren mit den Konstanten kann der Regler auf die Regelstrecke (die Motoren) eingestellt werden.

---

[18] Daniel, Bachfeld. „*Steuermann*“. c't Magazin für Computer Technik: Ausgabe 9/2006, S. 222 CD\Internetseiten\c't 9_2006, S. 222 c't-Bot.htm <http://www.heise.de/ct/06/09/222/>

Für die Umsetzung der Formel in der Software müssen die beiden mathematischen Operationen Integrieren und Differenzieren nachgebildet werden. So wird das Integrieren von e(t) durch das Aufsummieren von e(t) bei jedem Durchgang gelöst und das Differenzieren durch den Quotienten $\frac{AlterIstwert - Istwert}{T}$ umschrieben, der die Steigung der Zu- bzw. Abnahme des Istwerts berechnet. Dabei ist T die Zeit zwischen zwei Stellwertberechnungen. Im Weiteren wird T dem Faktor K_D zugerechnet und taucht deshalb nicht mehr auf. Durch diese Modifikationen ergibt sich folgende Schleife für einen Regler:

$$Do\{$$

$$Abweichungssumme = Abweichungssumme + Sollwert - Istwert$$

$$Stellwert = \underbrace{K_P(Sollwert - Istwert)}_{proportional} + \underbrace{K_I(Abweichungssumme)}_{\text{int}egral} + \underbrace{K_D(AlterIstwert - Istwert)}_{differentiell}$$

$$AlterIstwert = Istwert$$

$$\}Loop\ Until\ (abgefahrene\ Strecke \geq abzufahrende\ Strecke)$$

Nach einer vorgegebenen Anzahl von Impulsen der Radencoder wird die Regelschleife verlassen. So wird die Länge der abzufahrende Strecke oder der Winkel der Drehung festgelegt.
Für die Regelschleife ist es unerheblich in welche Richtung die Räder drehen, da der Betrag der Geschwindigkeit für den Regler als Istwert verwendet wird. Deshalb können vor Beginn der Regelschleife die Drehrichtungen der Motoren gestellt werden. Dadurch kann sich der Roboter geregelt um eine vorgegebene Anzahl von Impulsen sowohl in beide Richtungen drehen als auch vor- oder rückwärts fahren.

7.3. Der Master-Controller

Über ein Menü können am Master-Controller verschiedene Programme, wie das Anzeigen von Sensorwerten, das Demonstrieren ausgewählter Funktionen oder das Abfahren verschiedener Ballsammelkurse, gewählt werden. Diese Kurse bestehen, ausgehend von einer einprogrammierten Startposition, aus einer Abfolge von Koordinaten, die der Roboter der Reihe nach anfährt. Dazu berechnet der Master-Controller die einzelnen Bewegungsvektoren, die die jeweilige Position des Roboters mit der nächsten Position im Kurs verbinden und übermittelt sie dem Slave. Dieses Verfahren hat unter anderem den Vorteil, dass der Roboter sich im Falle einer Positionskorrektur, selbstständig einen Vektor zur nächsten Kursposition berechnen kann.

7.3.1. Das Koordinatensystem

Das Koordinatensystem ist so auf das Spielfeld gelegt, dass die Koordinaten des Kurses im Sammelbereich positiv und damit einfach zu berechnen sind. Die Position des Roboters wird durch den Ortsvektor des Mittelpunkts des Roboters repräsentiert. Durch Aufsummieren der einzelnen Bewegungsvektoren ist dem Master-Controller die Position des Roboters immer bekannt.

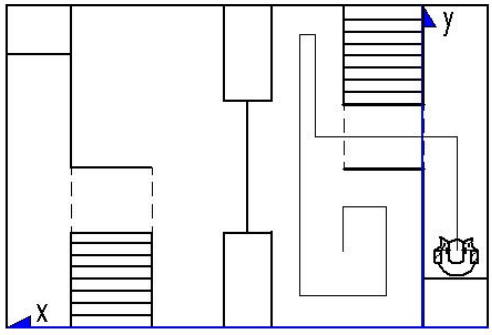

Abbildung 12

7.3.2. Das gezielte Abschießen der Tischtennisbälle zur gegnerischen Seite

Nachdem ein Tischtennisball in die Abschussposition gebracht wurde, wird die Fahrt durch den entsprechenden Befehl an den Slave abgebrochen. Durch die vom Slave angeforderten Daten des schon abgefahrenen Vektors kann der Master die aktuelle Position des Roboters berechnen. Über eine Fallunterscheidung bestimmt der Master dann in welchem der vier Gebiete, die in Abbildung 13 zu sehen sind, er sich befindet. Zu jedem der Gebiete existieren

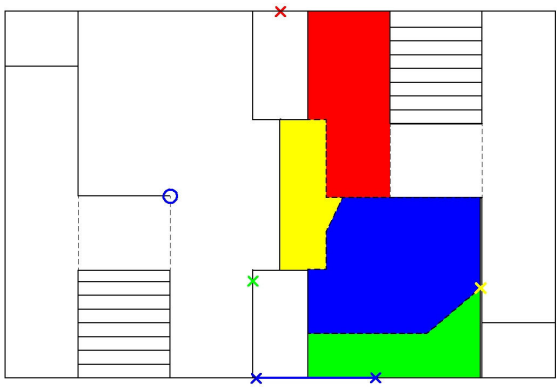

Abbildung 13

Koordinaten (im Bild in der Farbe des dazugehörigen Gebietes gehaltene Kreuze) die so liegen, dass ein auf sie abgeschossener Tischtennisball entweder durch Stoßen mit einer Bande oder direkt auf die gegnerische Spielfeldhälfte gelangt und dort zum Liegen kommt. Während im roten, gelben und grünen Bereich jeweils nur ein Zielpunkt existiert, werden im blauen Bereich für jede Position eigene Zielkoordinaten berechnet, sodass die Tischtennisbälle an der Bande zu dem blauen Kreis reflektiert werden. Im Folgenden wird gezeigt, wie der Master die Orientierung berechnet, auf die er sich drehen muss, um die Zielkoordinaten zu treffen.

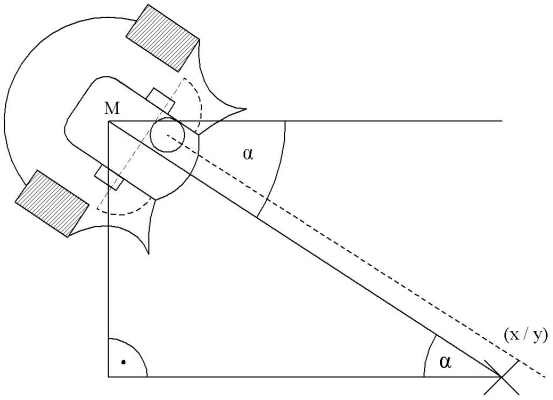

Abbildung 14

Zunächst wird angenommen, dass der Ball aus der Mitte $M(x_R / y_R)$ des Roboters abgeschossen wird. Die Richtung α, auf die sich der Roboter drehen müsste, um das Ziel zu treffen, berechnet sich mit Hilfe des Tangens zu:

$$\alpha = \arctan\left(\frac{y - y_R}{x - x_R}\right).$$

Da der Ball aber nicht mittig abgeschossen wird, sondern leicht versetzt, muss sich der Roboter noch ein Stück weiter drehen. Diesen Winkel β erhält man nach Abbildung 15 über den Sinus:

$\beta = \arcsin\left(\frac{d}{D}\right)$ wobei d die Länge des Lotes vom Mittelpunkt M des Roboters auf die verlängerte Ballabschussgerade im Punkt B und D der Abstand des Robotermittelpunkts von den Zielkoordinaten ist.

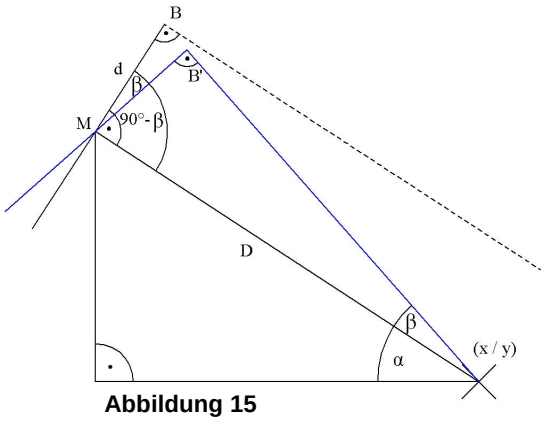

Abbildung 15

Mit $D = \sqrt{(x - x_R)^2 + (y - y_R)^2}$ folgt $\beta = \arcsin\left(\frac{d}{\sqrt{(x - x_R)^2 + (y - y_R)^2}}\right)$.

Addiert man α und β erhält man die Orientierung θ, auf die sich der Roboter drehen muss um das Ziel zu treffen.

7.4. Die Positionsbestimmung

Der Roboter hat über das einprogrammierte Koordinatensystem immer eine genaue Position in seiner virtuellen Realität, die aber nie exakt mit seiner wirklichen Position übereinstimmt. Vielmehr hält sich der Roboter irgendwo in der Nähe der berechneten Position auf. Je mehr Bewegungen der Roboter in dem Koordinatensystem ausführt, desto mehr weichen die berechneten Koordinaten von der wirklichen Position ab. Dazu führen Ungenauigkeiten bei den Drehungen oder beim Geradeausfahren, die durch unterschiedliche Reibung innerhalb der beiden Antriebe zustande kommen, leichtes Rutschen auf dem Untergrund beim Bremsen oder Anfahren und die Tatsache, dass sich der Roboter, wegen Reibung des Stützrades mit dem Untergrund, nie um seinen theoretischen Mittelpunkt dreht, wodurch sich auch bei einer Drehung sein Standort geringfügig ändert. Um diesen Faktoren entgegen zu wirken wurden breite Moosgummi-Reifen gewählt, die für eine gute Haftung mit dem Boden sorgen. Das Stützrad besteht aus einer, in alle Richtungen drehbaren, Metallkugel, welche auf glattem Untergrund gut rutscht, um die Bewegungen des Roboter möglichst wenig zu beeinflussen. Des Weiteren wurde in der Software des Slaves ein sanftes Anfahren und Abbremsen realisiert, um ein Rutschen der Räder zu verhindern. Außerdem sorgt eine Routine nach dem Verlassen der Regelschleife durch eine Korrekturbewegung dafür, dass exakt die vorgegebene Strecke abgefahren wird.

7.4.1. Möglichkeiten der Positionskorrektur

Trotz all dieser Maßnahmen kommt es zu Fehlern, die sich aufsummieren. Es muss also versucht werden die Position durch andere, von den Radencodern unabhängige Messergebnisse zu bestimmen, um die Position von Zeit zu Zeit korrigieren zu können. Im Zusammenhang mit dem vorgegebenen Szenario, bei dem die Lagen der Banden bekannt sind, bieten sich Abstandssensoren an um eine unabhängige Positionsbestimmung zu realisieren.

7.4.1.1. Das Ausrichten an einer Bande

Eine Möglichkeit ist das senkrechte Ausrichten zu einer Bande. Wenn der Roboter nach seiner virtuellen Realität senkrecht zu einer Bande stehen müsste kann dies leicht über die frontal angebrachten Ultraschallsensoren nachgemessen und gegebenenfalls die Ausrichtung korrigiert werden. Dazu werden die beiden gemessenen Abstände verglichen und über den Tangens ein Korrekturwinkel berechnet:

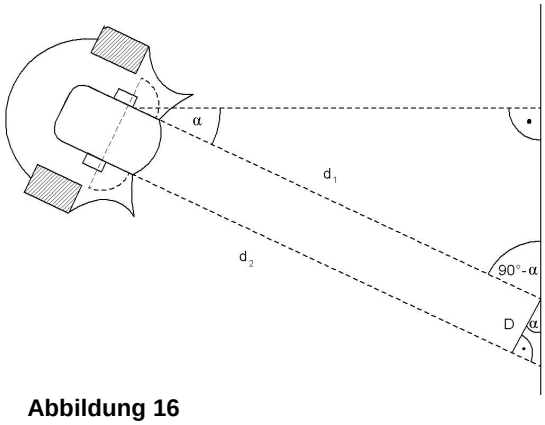

Abbildung 16

$$\tan\alpha = \frac{d_2 - d_1}{D} \quad \Rightarrow \quad \alpha = \arctan\left(\frac{d_2 - d_1}{D}\right)$$

Der Roboter dreht sich nun um den berechneten Winkel, wobei die Drehrichtung dem Vorzeichen des berechneten Winkels entnommen wird. Es wird so oft nachkorrigiert, bis die Ultraschallsensoren den gleichen Abstand messen.
Da durch aufsummierte falsche Drehungen große Positionsfehler entstehen, wird das Ausrichten möglichst oft wiederholt, um Orientierungsfehler und damit Positionsfehler zu minimieren.

7.4.1.2. Die Positionsbestimmung an einer Ecke

Eine vollständige Positionsbestimmung ist möglich, wenn der Roboter sich auf eine Ecke zu bewegt, von der ihm die Koordinaten einprogrammiert wurden.
Zunächst richtet sich der Roboter, mit der in 7.4.1.1 genannten Methode, senkrecht zu der vor ihm liegenden Bande aus und speichert den Abstand A zu dieser.
Im zweiten Schritt dreht er sich um 90°, so dass er nun zu der anderen Bande schaut. Aus Zeitgründen wird nun angenommen, dass ein einmaliges Drehen um 90° vernachlässigbar geringe Fehler beinhaltet und deshalb kein nochmaliges Ausrichten an dieser Bande nötig ist. Deshalb wird einfach die Entfernung B zur Bande gemessen und wiederum gespeichert.
Die Abstände A und B sind vom Mittelpunkt des Roboters aus gerechnet. Man erhält sie dadurch, dass zu dem vom Ultraschallsensor gemessenen Abstand immer noch der Abstand Ultraschallsensor $\leftrightarrow$ Mittelpunkt des Roboters hinzuaddiert wird.

Die genauen Koordinaten des Roboters (x_R / y_R) lassen sich aus den zwei Abständen A und B, unter Verwendung der einprogrammierten Koordinaten (x / y) der Ecke, bestimmen.

Über eine Fallunterscheidung muss außerdem noch einbezogen werden, welche Orientierung der Roboter nach dem Ausrichten hat und in welche Richtung er sich dann um 90° dreht. Nimmt man zum Beispiel an, der Roboter würde nach dem Ausrichten an der Bande die Orientierung $\theta = 180°$ haben, A messen und sich dann um 90° im Uhrzeigersinn drehen um B zu messen, so ergeben sich die Koordinaten $(x + A / y + B)$ für den Roboter.

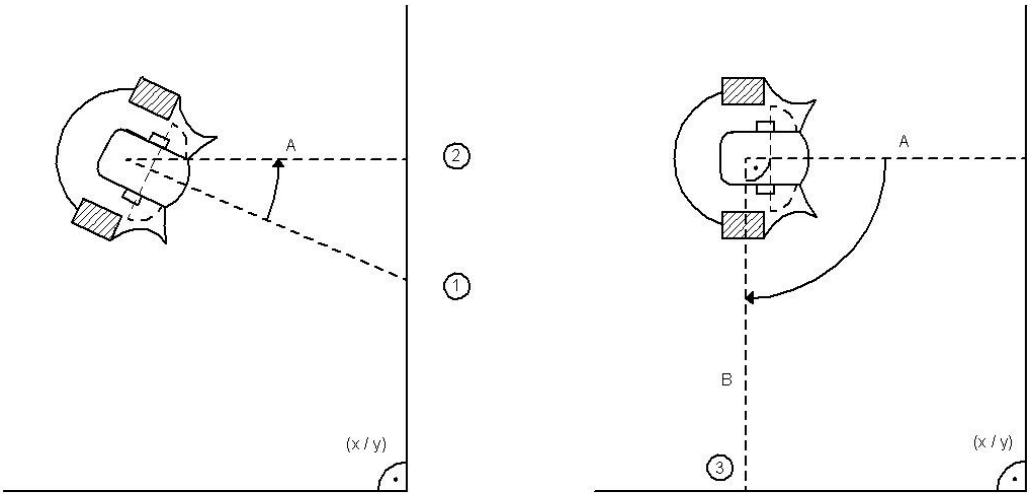

Abbildung 17

Die Position des Roboters wird mit den auf diese Weise erhaltenen Koordinaten überschrieben.

8. DANKSAGUNG

An dieser Stelle möchte ich mich ganz besonders bei meinen Eltern bedanken ohne deren finanzielle Unterstützung der Roboter nicht realisierbar gewesen wäre.
Außerdem gilt mein Dank Ludwig Nägele für das professionelle Filmen bei Roboking 2007. Und natürlich möchte ich mich auch bei meinem altbewährten Roboking Team, bestehend aus Martin Sevenich, Ferdinand Sedlmair und Matthias Weber, für die seelische und moralische Unterstützung bei den Wettbewerben bedanken.

9. SCHLUSSWORT

Rückblickend habe ich festgestellt, wie wichtig ein strukturiertes Vorgehen bei dem Bau eines Roboters ist. Dies zeigt sich insbesondere an der Programmierung, bei der ein gut strukturierter Quellcode schnelle und gezielte Änderungen und Fehlerbehebung auch unter dem enormen Zeitdruck eines Wettbewerbs ermöglicht. Ein ähnlich wichtiger Punkt ist die flexible Gestaltung des Roboters, zum Beispiel durch standartisierte Stecksysteme, die ein schnelles Anpassen an veränderte Situationen gestattet. Denn vor dem eigentlichen Wettbewerb bleiben nur wenige Stunden, um den Roboter unter realen Einsatzbedingungen zu testen. Aber am Wichtigsten ist ein genauer Vorgehensplan, der feste Teilziele setzt. Ohne diese Etappenziele hätte ich in den über acht Monaten Bau- und Programmierungszeit den Überblick verloren und wäre sicher in den Wochen vor dem Wettbewerb in Zeitnot geraten. Der Roboter hat mich in diesen acht Monaten durchschnittlich geschätzte drei Stunden am Tag beschäftigt. In dieser Zeit habe ich in vielen Bereichen, wie der Konstruktion, der Elektronik und der Programmierung, dazugelernt. Dieses große Spektrum an verschiedenen Wissensgebieten, die alle zum Bau eines Roboters notwendig sind, macht den Reiz und die Faszination der Robotik aus. Mir persönlich macht das ständige Weiterentwickeln meiner Techniken und Fertigkeiten und die Erweiterung meines Wissenshorizontes so sehr Spaß, dass ich, trotzdem ich den Einzug ins Finale des Roboking 2007 knapp verpasst habe, mit dem Bau von Robotern weitermachen werde. Ganz nach dem Motto:

„Der Weg ist das Ziel.“

10. LITERATURVERZEICHNIS

1. Datenblätter und Internetquellen

Atmel, *Datenblatt des ATmega8*. 16.12.2006, CD\Datenblätter\ATmega8.pdf
<http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf>

Atmel, *Datenblatt des ATmega32*. 16.12.2006, CD\Datenblätter\ATmega32.pdf
<http://www.atmel.com/dyn/resources/prod_documents/doc2503.pdf>

Daniel, Bachfeld. „*Steuermann“*. c't Magazin für Computer Technik: Ausgabe 9/2006, S. 222
CD\Internetseiten\c't 9_2006, S. 222 c't-Bot.htm <http://www.heise.de/ct/06/09/222/>

Hammer/Hammer: *Pysikalische Formeln und Tabellen*. München: J. Lindauer Verlag (Schaefer), 2002. S. 87

Jörg, Pohl. *SRF05 - Ultra-Sonic Ranger*. 03.01.2007, CD\Datenblätter\SRF05.pdf
<http://www.roboter-teile.de/datasheets/srf05.pdf>

Maxon motors, *Encoder Enc 22, 100 Impulse, 2 Kanal*. 09.01.2007, CD\Datenblätter\Enc 22.pdf
<http://test.maxonmotor.com/docsx/Download/catalog_2006/Pdf/06_248-249_d.pdf>

Organisationsteam Roboking, *Roboking 2007 Wettbewerbsdokumentation*. 20.06.2006,
CD\Roboking\RoboKing2007_Wettbewerbsdoku.pdf
<http://www.tu-chemnitz.de/etit/proaut/rk/fileadmin/user_upload/downloads/RoboKing2007_Wettbewerbsdoku.pdf>

Organisationsteam Roboking, *Roboking 2007 Spielfeld*. 20.06.2006,
CD\Roboking\RoboKing2007_Spielfeld.pdf
<http://www.tu-chemnitz.de/etit/proaut/rk/fileadmin/user_upload/downloads/RoboKing2007_Spielfeld.pdf>

SGS-Thomson microelektronics, *L78S00 Series*. 29.12.2006, CD\Datenblätter\L78S05.pdf
<http://www.datasheetarchive.com/datasheet.php?article=1969221>

Sharp, *GP2D120 Optoelectronic Device*. 03.01.2007, CD\Datenblätter\GP2D120.pdf
<http://document.sharpsma.com/files/GP2D120-DATA-SHEET.PDF>

Sharp, *chapter 12 - Use of Optical Sensor Units*. 04.01.2007, CD\Datenblätter\Triangulation.pdf
<http://document.sharpsma.com/files/Optical_Sensors_AN.pdf>

Sharp, *chapter 5 – Use of Light Detecting Devices*. 04.01.2007, CD\Datenblätter\PSD.pdf
<http://document.sharpsma.com/files/Light_Detecting_Device_AN.pdf>

Sharp, *IS471F*. 03.01.2006, CD\Datenblätter\IS471F.pdf
<http://www.roboter-teile.de/datasheets/IS471F.pdf>

Unitrode, *Push-Pull Four Channel Driver – L293D.* 29.12.2006, CD\Datenblätter\L293D.pdf
<http://www.datasheetarchive.com/datasheet.php?article=3898324>

2. Abbildungen

Abbildung 5: CD\Datenblätter\L293D.pdf
<http://www.datasheetarchive.com/datasheet.php?article=3898324>

Abbildung 1-3, 6, 8, 10, 12-17: entworfen mit Solide Edge V16 Layout

Abbildung 4, 11: entworfen mit Microsoft Office Visio 2003

Abbildung 7, 9: erstellt mit Microsoft Office Excel 2003

11. ANHANG

11.1. Herleitungen

Herleitung von b:

$$\left.\begin{array}{l}(I)\ \ a = y_1 \cdot (x_1 - b) \\ (II)\ a = y_2 \cdot (x_2 - b)\end{array}\right\} \Rightarrow y_2 x_2 - y_2 b = y_1 x_1 - y_1 b$$

$$\Rightarrow y_2 b - y_1 b = y_2 x_2 - y_1 x_1 \quad | : (y_2 - y_1)$$

$$\Rightarrow \underline{\underline{b = \frac{y_2 x_2 - y_1 x_1}{y_2 - y_1}}}$$

Herleitung von a:

$$\left.\begin{array}{l}(I)\ \ a = y_1 \cdot (x_1 - b) \ \Rightarrow \frac{a}{y_1} = x_1 - b \ \Rightarrow b = x_1 - \frac{a}{y_1} \\ (II)\ a = y_2 \cdot (x_2 - b) \ \Rightarrow \frac{a}{y_2} = x_2 - b \ \Rightarrow b = x_2 - \frac{a}{y_2}\end{array}\right\} \Rightarrow x_1 - \frac{a}{y_1} = x_2 - \frac{a}{y_2}$$

$$\Rightarrow x_1 - x_2 = \frac{a}{y_1} - \frac{a}{y_2}$$

$$\Rightarrow x_1 - x_2 = \frac{a y_2 - a y_1}{y_1 y_2}$$

$$\Rightarrow y_1 y_2 (x_1 - x_2) = a \cdot (y_2 - y_1)$$

$$\Rightarrow \underline{\underline{a = \frac{y_1 y_2 (x_1 - x_2)}{y_2 - y_1}}}$$

11.2. Bildergalerie

Die Bilder finden sich in der Bildergalerie meiner Homepage:
www.javik-robotik.de.vu

11.3. Bauteilliste

Bauteil	Händler	Anzahl	Gesamtpreis
Zahnriemenscheibe 30Z	Conrad	2	23,90 €
Zahnriemenscheibe 10Z	Conrad	2	13,90 €
Zahnriemen 85Z	Conrad	2	6,30 €
C-Kugellager 8/16	Conrad	4	10,00 €
2 Moosgummireifen	Conrad	1	14,95 €
Akkupack Panasonic 5 Zellen 6V 3500mAh	Conrad	1	30,72 €
RS-101MGB Servo	Conrad	3	7,50 €
Maxon motoren 10:1 mit Enc 22 Radencodern	eBay	2	25,00 €
LCD 4*16 Zeichen	eBay	1	12,00 €
Kugelrad	Baumarkt	1	4,99 €
Kunststoff- und Plexiglasplatten	Siegele	1	13,00 €
2 Wegwerfkameras	Müller	1	9,95 €
SRF05 Ultraschall-Entfernungssensor	Roboter-teile	2	44,08 €
GP2D120 Infrarot-Entfernungssensor	Roboter-teile	1	12,40 €
Set Sharp Entfernungssensor IS471 und IR-LED	Roboter-teile	2	5,26 €
Diverse Kondensatoren und Widerstände	Ausgelötet		Kostenlos
ATmega8 Controller	Reichelt	1	2,75 €
IC-Sockel, 28-polig	Reichelt	1	0,33 €
ATmega32 Controller	Reichelt	1	3,50 €
IC-Sockel, 40-polig	Reichelt	1	0,45 €
Standardquarz, 14,7456MHz	Reichelt	1	0,44 €
Standardquarz, Grundton, 16,0 MHz	Reichelt	1	0,23 €
Glimmer-Kondensator, 22pF	Reichelt	4	1,96 €
100nF Kondensator	Reichelt	5	0,75 €
MAX232 RS232-Driver, DIL-16	Reichelt	1	0,41 €
L293D 4-Ch-Driver mit Diode, DIL-16	Reichelt	1	2,00 €
IC-Sockel, 16-polig	Reichelt	2	0,36 €
L78S05 Spannungsregler 2A positiv, TO-220	Reichelt	2	0,70 €
Piezosummer	Reichelt	1	2,15 €
LED 5mm, Low-Current, 2 mA, gelb	Reichelt	2	0,18 €
LED 5mm, Low-Current, 2 mA, grün	Reichelt	3	0,27 €
LED 5mm, Low-Current, 2 mA, rot	Reichelt	2	0,18 €
Kurzhubtaster 6x6mm, Höhe: 4,3mm,	Reichelt	7	0,57 €
Zylinderkopfschrauben M3* 16 100 Stk.	Reichelt	1	1,40 €
Zylinderkopfschrauben M3* 20 100 Stk.	Reichelt	1	1,51 €
6-kant Muttern, M3, 100 Stk.	Reichelt	1	0,75 €
Federringe, 3mm, 100 Stk.	Reichelt	1	0,77 €
Unterlegscheiben, 3,2mm, 100 Stk.	Reichelt	1	0,74 €
Distanzhülsen, Metall, 6-Kant, 10mm	Reichelt	4	0,36 €
Platine, Epoxyd, einseitig Cu, 160*100	Reichelt	1	2,40 €
VDE-Isolierband, 10M, schwarz	Reichelt	2	0,92 €
40pol. Stiftleiste, gerade, RM 2,54	Reichelt	2	0,37 €
2x40pol.-Stiftleiste, gerade, RM 2,45	Reichelt	1	0,28 €
20pol. Buchsenleiste, gerade	Reichelt	4	1,36 €
Tamiya-Stecker	Reichelt	2	1,90 €
Tamiya-Kupplung	Reichelt	2	1,50 €
Wannenstecker, 10-polig,	Reichelt	3	0,21 €
Wannenstecker, 10-polig, Schneidklemmtechnik	Reichelt	1	0,54 €
Pfostenbuchse, 10-polig	Reichelt	6	0,54 €
Flachbandkabel, 10-polig, 3m	Reichelt	1	1,85 €
Anschlussklemme 2-polig, RM 5,08	Reichelt	4	0,52 €
Versandkosten	Reichelt/Conrad	5	25,00 €
	Gesamtkosten:		294,10 €

12. ERKLÄRUNG

Ich erkläre hiermit, dass ich die Facharbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benützt habe.

_____, den _____ _____

(Ort, Datum) (Unterschrift des Schülers)