

RoboKing 2006

Wettbewerbsdokumentation
erste korrigierte Fassung



TECHNISCHE UNIVERSITÄT
CHEMNITZ

Niko Sünderhauf

Chemnitz, am 25. September 2005

Inhaltsverzeichnis

I	Allgemeine Informationen & Regelwerk	8
1	Allgemeines	9
1.1	Veranstalter	9
2	Wettbewerbsbestimmungen	10
2.1	Teilnahmebestimmungen	10
2.2	Gutschrift	10
2.2.1	Einlösen der Gutschrift	11
2.3	Dokumentationspflicht	11
2.4	Roboter	11
2.4.1	Einschränkungen	12
2.4.2	Hinweise zur Wahl der Mikrocontroller	13
2.5	Aufgabenbeschreibung	13
2.5.1	Das Spielfeld	13
2.5.2	Zielstellung	14
2.6	Zwischenwettbewerb	17
2.7	Hinweise für die Teamleiter	18
2.8	Termine	19
II	Technische Informationen	21
3	Mechanik	23
3.1	Antriebskonzepte	23
3.1.1	Differentialantrieb	23
3.1.2	Kettenantrieb	24
3.1.3	Dreiradantrieb	24
3.1.4	Ackermannlenkung	25
3.1.5	Omni-Wheel Antrieb	25
3.2	Motoren	27
3.2.1	Dimensionierung	27
3.2.2	Ansteuerung	30
3.2.3	Servomotoren	31

3.2.4	Servomotoren als Antriebsmotor	32
4	Mikrocontroller	33
4.1	Das Handyboard	33
4.2	Atmel AVR	35
4.2.1	Programmierung	35
4.2.2	Mega32 Board von Krause-Robotik	36
4.2.3	USC008 Board von Embedit	37
4.2.4	ATmega128 Platine	38
4.2.5	ATmega64 Platine von Embedit	39
4.3	weitere Controller Familien	40
5	Stromversorgung - Akkus	41
5.1	Hinweise zu Auswahl und Anschluß der Akkus	42
6	Sensoren	43
6.1	Reflexkoppler	43
6.1.1	Funktionsweise	44
6.1.2	Reflexkoppler - CNY-70	44
6.1.3	Farbsensoren	45
6.1.4	Farbsensoren im Eigenbau	45
6.1.5	Probleme	46
6.2	Entfernungsmessung	47
6.2.1	Objekterkennung auf IR-Basis - Sharp IS471	48
6.2.2	Entfernungsmessung auf IR-Basis - GP2D1xx	48
6.2.3	Entfernungsmessung mit Ultraschall - SRF04	50
6.2.4	Entfernungsmessung mit Ultraschall - SRF08	52
6.3	Radencoder	54
6.3.1	mechanische Radencoder	55
6.3.2	optische Radencoder	55
6.3.3	induktive Drehzahlmessung	57
6.4	Kompass	59
7	Kommunikation	60
7.1	Der I^2C -Bus	60
7.1.1	Hardware und Elektrische Aspekte	61
7.1.2	Ablauf der Übertragung	61
7.1.3	Programmierung	62
7.1.4	Anwendungen	62
7.2	RS-232	63
7.2.1	Hardware und Elektrische Aspekte	63
7.2.2	Ablauf der Übertragung	63
7.2.3	RS-232 mit Mikrocontrollern	64
7.2.4	Anwendung	65

8 Weiterführende Informationen	67
A Bezugsquellen und Preise	69

Änderungen gegenüber der ersten korrigierten Fassung

Dies ist die zweite korrigierte Version dieses Dokuments vom 25. September 2005. Nachfolgende Liste enthält alle Änderungen zur vorhergehenden Fassung. Änderungen sind im Text **rot** gekennzeichnet.

- Termin der Hannover Messe korrigiert
- Maßfeil in Abbildung 2.2 ergänzt
- Bild 2.6 getauscht

Änderungen der ersten korrigierten Fassung vom 25. Juli 2005 gegenüber der Originalfassung

Nachfolgende Liste enthält alle Änderungen zur vorhergehenden Fassung. Änderungen sind im Text **rot** gekennzeichnet.

- Maße der Steine geändert
- Maße in Abbildungen 2.3, 2.2 geändert

Teil I

**Allgemeine Informationen &
Regelwerk**

Kapitel 1

Allgemeines

Der RoboKing ist ein bundesweiter Roboterwettbewerb für Schüler. Ziel des Wettbewerbes ist die Förderung des technisch und wissenschaftlich interessierten Nachwuchses. Der RoboKing wird auf der Hannover Messe 2006 öffentlich ausgetragen.

1.1 Veranstalter

Veranstalter des RoboKings ist die Arbeitsgemeinschaft Robotik der TU Chemnitz und die Professur für Prozessautomatisierung unter Leitung von Prof. Dr.-Ing. Peter Protzel

Der Veranstalter ist erreichbar unter:

RoboKing
z.H. Frau Herfter
Professur für Prozessautomatisierung
Technische Universität Chemnitz
09107 Chemnitz

eMail: info@roboking.de
Telefon: 0371 531 3442
Fax: 0371 531 3352
www: <http://www.roboking.de>

Kapitel 2

Wettbewerbsbestimmungen

2.1 Teilnahmebestimmungen

Teilnahmeberechtigt sind Teams aus 3 bis maximal 5 Mitgliedern plus einem Teamleiter.

Die Teammitglieder müssen am Tag der Wettbewerbsendrunde im April 2006 Schüler, Zivildienstleistende oder Wehrdienstleistende sein.

Personen mit begonnener oder abgeschlossener Berufsausbildung und/oder begunenem Studium sind von der Teilnahme ausgeschlossen.

Der Teamleiter muss volljährig sein. Ansonsten gelten für den Teamleiter keinerlei Einschränkungen.

2.2 Gutschrift

Für zum Bau des Roboters benötigte Bauteile erhalten die teilnehmenden Teams eine Gutschrift in Höhe von 350 Euro. Alle ausgehändigten oder von der Gutschrift gekauften Materialien verbleiben im Besitz des Veranstalters.

Erst wenn das Team zur Wettbewerbsendrunde mit einem der Aufgabenstellung entsprechenden Roboter antritt, gehen die genannten Materialien in den Besitz des Teamleiters über. Dazu bedarf es einer formalen Bestätigung durch den Veranstalter.

Tritt ein Team nicht mit einem der Aufgabe entsprechenden Roboter zum Wettbewerb an, oder gibt ein Team vor oder während des Wettbewerbes auf, oder wird ein Team vor oder während des Wettbewerbes disqualifiziert, sind sämtliche ausgehändigten und bisher von der Gutschrift gekauften Materialien an den Veranstalter zurückzugeben.

Verantwortlich dafür ist der Teamleiter.

2.2.1 Einlösen der Gutschrift

Die Gutschrift in Höhe von 350 Euro kann bei beliebigen Händlern eingelöst werden.

Dazu werden die Bestellwünsche über ein Formular auf der RoboKing-Homepage dem Veranstalter mitgeteilt. Der Veranstalter kann eine Bestellung nach Überprüfung akzeptieren oder zurückweisen. Das Team wird per eMail über die Entscheidung benachrichtigt. Danach können die Materialien vom Team gekauft werden. Nach Einschicken der Original-Rechnung an den Veranstalter wird der Kaufpreis erstattet. Das Team muß dem Veranstalter dazu zusammen mit der Rechnung eine Bankverbindung mitteilen und die per Mail erhaltene Bestätigung ausdrucken und mitschicken. Außerdem ist eine Kopie des entsprechenden Kontoauszugs beizulegen.

Der Kaufpreis wird nur bei akzeptierten Bestellungen erstattet.

2.3 Dokumentationspflicht

Jede Gruppe hat eine Dokumentation zu ihrem Roboter zu erstellen.

Die Dokumentation enthält eine ausführliche Erläuterung der Lösungsstrategie sowie Erläuterungen und Schaltpläne zu allen verwendeten Sensoren und Schaltungen und den kompletten und ausreichend kommentierten Programmquelltext.

Die vollständige Dokumentation ist den Veranstaltern spätestens am Wettbewerbstag zu übergeben, die Abgabe ist Voraussetzung zur Teilnahme an der Endrunde.

Die Dokumentation ist nach Möglichkeit in elektronischer Form (als .pdf, .doc oder sonstigen Standard-Formaten auf CD oder USB-Stick) vorzulegen. Die Dokumentation wird nach dem Wettbewerb auf der RoboKing-Homepage veröffentlicht.

Bis zum 30. November 2005 ist beim Veranstalter eine Zwischendokumentation abzugeben, die den aktuellen Stand der Arbeiten am Roboter belegt. Sie besteht aus Fotos sowie einer Konzeptbeschreibung, welche auf den mechanischen Aufbau, Sensorik und Aktorik sowie die geplante Strategie eingeht. Natürlich kann der Roboter danach noch umgebaut werden oder andere Sensoren etc. benutzt werden. Die Zwischendokumentation ist Teilnahmevoraussetzung zum Zwischenwettbewerb. Sie wird erst nach dem Zwischenwettbewerb im Januar den anderen Teilnehmern auf der Homepage zugänglich gemacht.

2.4 Roboter

Bevor ein Team zum Wettbewerb antreten kann, muß der Roboter vom Veranstalter formal genehmigt werden.

Der Roboter wird dazu einer Prüfung unterzogen um festzustellen, ob er der Aufgabenstellung entspricht. Insbesondere wird geprüft, ob der Roboter in der Lage ist, bevorstehende oder bereits erfolgte Kollisionen mit dem gegnerischen Roboter zu erkennen und zu verarbeiten. Grundsätzlich hat sich der Roboter bei bevorste-

henden Kollisionen defensiv zu verhalten (Ausweichen, Zurückfahren), um eine Beschädigung des Gegners oder des Spielfeldes zu vermeiden. Roboter, die dieses defensive Verhalten nicht zeigen, können von den Schiedsrichtern mit Strafpunkten belegt werden.

Die Roboter dürfen nicht so konstruiert oder programmiert sein, dass sie den Gegner absichtlich beschädigen oder übermäßig behindern.

Es dürfen keine Gegenstände, Flüssigkeiten usw. ausgebracht werden. Manipulationen am Spielfeld sind untersagt, insbesondere das Entfernen oder Anbringen von Markierungen. Die Verwendung von Baken oder ähnlichen Systemen zur Markierung der Heimatbasis etc. ist nicht gestattet. Das Antriebssystem muß so ausgelegt sein, dass die Spielfeldoberfläche auf keinen Fall beschädigt wird. (Vorsicht bei Kettenantrieb!) Der Gegner darf nicht absichtlich in seiner Umgebungswahrnehmung eingeschränkt werden, zum Beispiel durch Blenden oder Täuschen seiner Sensoren.

Jeder Roboter ist mit einem Not-Aus-Schalter (oder -Taster) zu versehen, der stets leicht erreicht werden kann. Dieser Schalter ist so anzubringen und kenntlich zu machen, dass er auch von den Veranstaltern jeder Zeit erkannt und bedient werden kann. Nach einer Betätigung des Not-Aus-Schalters hat der Roboter sofort seine Aktorik (insbesondere Antriebsmotoren) zu deaktivieren.

Die Roboter müssen für 20 Minuten Dauerbetrieb ausgelegt sein.

Die Roboter müssen völlig autonom sein. Jegliche Verbindung zu externen Geräten ist untersagt. Dies gilt insbesondere für Verbindungen zur Datenübertragung oder zur Stromversorgung. Jeder Roboter muß aufgrund seiner Sensorik und Aktorik in der Lage sein, die Aufgabenstellung selbständig zu lösen. Keinesfalls dürfen Handeingriffe ein Bestandteil der Lösungsstrategie sein.

Die Roboter müssen so konstruiert sein, dass ein unbeabsichtigtes Betätigen von Schaltern, Tastern und sonstigen Bedienelementen verhindert wird. Ebenso sind Kabel, Steckverbinder usw. so anzubringen, dass sie sich während eines Spiels nicht lösen können oder über den Umfang des Roboters hinausragen oder den Gegner behindern.

Verstöße gegen die hier aufgeführten Punkte können zu einer sofortigen Disqualifikation führen.

2.4.1 Einschränkungen

Es darf nur *ein* nutzerprogrammierter 8-Bit Controller oder Prozessor verwendet werden. Herstellerprogrammierte Controller, wie sie auf verschiedenen Sensoren (SRF08, CMU-Cam, SD20 etc.) zu finden sind, unterliegen keiner Einschränkung, solange sie das vom Hersteller eingebrachte Programm enthalten.

Die Verwendung von Baken oder ähnlichen Systemen zur Markierung der Heimatbasis etc. ist nicht gestattet.

Weitere Einschränkungen werden nicht gemacht. Es liegt in der Eigenverantwortung der Teams, ihren Roboter so zu konstruieren, dass er die gestellte Aufgabe erfüllen kann.

2.4.2 Hinweise zur Wahl der Mikrocontroller

Teams, die noch keine Erfahrung mit Mikrocontrollern haben, empfehlen wir die Verwendung des Handyboards. Eine genaue Beschreibung seiner Features, sowie eine Gegenüberstellung mit anderen Controller-Typen gibt es im Kapitel 4.

2.5 Aufgabenbeschreibung

Jedes Team soll einen autonomen Roboter so konstruieren und programmieren, dass er in der Lage ist, die nachfolgend beschriebene Aufgabe zu lösen.

2.5.1 Das Spielfeld

Das Spielfeld ist 300x250 cm groß und durch einen 30 cm breiten Graben in zwei Hälften geteilt. Begrenzt wird das Spielfeld durchgängig von einer 14,7 cm hohen weißen, 2 cm breiten Bande. Das Spielfeld selbst ist gelbgrün (RAL 6018) gestrichen.

An der längeren Seite des Feldes befinden sich nebeneinander, durch den Graben getrennt, die beiden Heimatbasen. Hierbei unterscheiden wir zwischen der unteren Basis und der oberen Basis. Letztere liegt um 7 cm erhöht über der Spielfeldgrundfläche und ist über eine Schräge zu erreichen. Die Basen sind hellgrau (RAL 9018) und durch einen weißen, 3 cm breiten Streifen vom Rest des Feldes getrennt. Auf der den Basen gegenüberliegenden Seite des Spielfeldes befinden sich in den äußeren Ecken je eine Abholstation für Holzklötze (der "Steinbruch"). Die Klötze sind 70 mm lang, 40 mm tief und 15 mm hoch. Zur besseren Unterscheidung gibt es graue und sandsteinfarbene Steine. Die Abholstation in der linken unteren Ecke gehört zur Basis auf der rechten Hälfte. Aufgabe ist also, die Klötzchen von der linken Station in die rechte Basis zu befördern und umgekehrt. Dazu muß zunächst die Brücke über den Graben überwunden werden. Diese ist 55 cm breit. Zur Orientierung sind zwei schwarze 19 mm breite Linien angebracht, wie in Abbildung 2.2 zu sehen. Ebenfalls zur Orientierung dienen die beiden 3 cm breiten farbigen Streifen, die die Brücke begrenzen. Einer der Streifen ist in der Farbe Fernblau (RAL 5023) gestrichen, der andere in Rubinrot (RAL 3003).

Wie den Bildern zu entnehmen, ist der Graben, die obere Basis und die Rampe durch eine immer 2 cm hohe weiße Bande gesichert.

Der zu überwindende Hindernisparcours besteht neben der Brücke auch aus einer *Schlucht* und einem *Wald*. Die Schlucht besteht aus zwei parallel zueinander aufgestellten weißen Holzbrettern. Der Abstand zwischen ihnen beträgt 30 cm, ihre Höhe 14,7 cm. Am Ausgang der Schlucht beginnt eine schwarze Linie, die durch den Wald führt. Die Linie verläuft dabei immer in der Mitte zwischen zwei Bäumen. Die stilisierten Bäume sind rund und haben einen Durchmesser von 8 cm

RAL	Bezeichnung	Wofür?
9010	Reinweiß	Geländer, Unterbau, Brücke, Linie, Basis, Banden
9018	Papyrusweiß	Basis-Feld, Basis-Podest
6018	Gelbgrün	Feld, Auffahrt, Brücke
1002	Sandgelb	Steine
7012	Basaltgrau	Steine
8011	Nussbraun	Bäume
3003	Rubinrot	Linie Brücke
5023	Fernblau	Graben, Linie Brücke

Tabelle 2.1: Farbübersicht

(Mineralwasserflasche). Sie sind 11 cm hoch und nussbraun (RAL 8011). Darüber befinden sich noch grüne Blätter, die jedoch nicht über den Umfang des Stammes herausragen. Die Bäume werden auf dem Spielfeld fixiert und gegen Verschieben gesichert.

Der Steinbruch Die Steine werden nicht durch die Teammitglieder auf den Roboter gelegt, sondern durch eine automatische Abholstation. In der Ecke des Spielfeldes befindet sich auf die Bande der längeren Seite (parallel zu den Basen) gesteckt eine Platte, auf der zwei Servos befestigt sind, wie in Abbildung 2.5 gezeigt. Die Platte ragt etwa 5 cm in das Spielfeld hinein und ist 30 cm breit. Der Stein wird von einem Teammitglied an dem vorgesehenen Platz an der Vorderkante der Platte platziert. Der Roboter kann jetzt durch Auslösen eines Schalters unterhalb der Platte den Steinbruch aktivieren. Dabei drehen sich die beiden Servos wie durch die Pfeile verdeutlicht und schieben den Stein nach vorn von der Platte. Er fällt also aus etwa 16 cm Höhe 15 cm von der Seitenwand und 5 cm von der unteren Wand entfernt ins Spielfeld. Der Schalter befindet sich dabei mittig genau unterhalb des Klotzes und ist in seinem Aufbau gleich mit den Schaltern am Räuberversteck vom RoboKing 2005 bzw. den Bojen vom RoboKing 2004. Abbildung 2.6 zeigt den Aufbau. Im Gegensatz zu den letzten beiden Jahren hat die Infrarot-Diode diesmal keinerlei Funktion. Es wird weder ein Signal zum Orten des Schalters ausgestrahlt, noch eine Bestätigung nach Drücken des Schalters.

2.5.2 Zielstellung

Es treten pro Spielrunde immer zwei Roboter gegeneinander an. Die Spielzeit beträgt 7 Minuten. Ziel des Spiels ist, innerhalb der Spielzeit so viele eigene Klötchen wie möglich durch den Hindernisparcour in seine Heimatbasis zu befördern.

Punkteverteilung Es wird zwischen vier möglichen Zuständen unterschieden, wo sich ein Klotz nach Spielende befinden kann. Dementsprechend werden nach



Abbildung 2.1: Das Spielfeld im Überblick.

Ablauf der Spielzeit Punkte verteilt:

innerhalb der oberen Basis	7 Punkte
innerhalb der unteren Basis	5 Punkte
in oder auf dem Roboter	2 Punkte
außerhalb des Roboters und der Basen	0 Punkte

Ein Klotz gilt als innerhalb einer Basis befindlich, wenn sich seine Auflagefläche vollständig innerhalb der Basis befindet. Der weiße Rand gilt jeweils als zur Basis gehörig. Klötze, die sich nach Ablauf der Spielzeit innerhalb des Umfangs des Roboters oder auf dem Roboter befinden, zählen nicht als innerhalb der Basis befindlich, selbst wenn der Roboter in der Basis steht.

Es ist nicht gestattet, die Klötze ohne vorheriges Durchqueren des Hindernisparcours in einer der Basen abzulegen.

Handeingriffe Jeder Handeingriff wird mit drei Minuspunkten bestraft. Handeingriffe dienen dazu, in Ausnahmefällen korrigierend in das ansonsten autonome Verhalten des Roboters einzugreifen. Sie dürfen nicht Bestandteil der Lösungsstrategie sein. Bevorstehende Handeingriffe sind den Schiedsrichtern deutlich anzuzeigen. Der Roboter darf bei einem Handeingriff lediglich gedreht werden. Er darf nur soweit verschoben werden, wie es für die Ausführung der Drehung

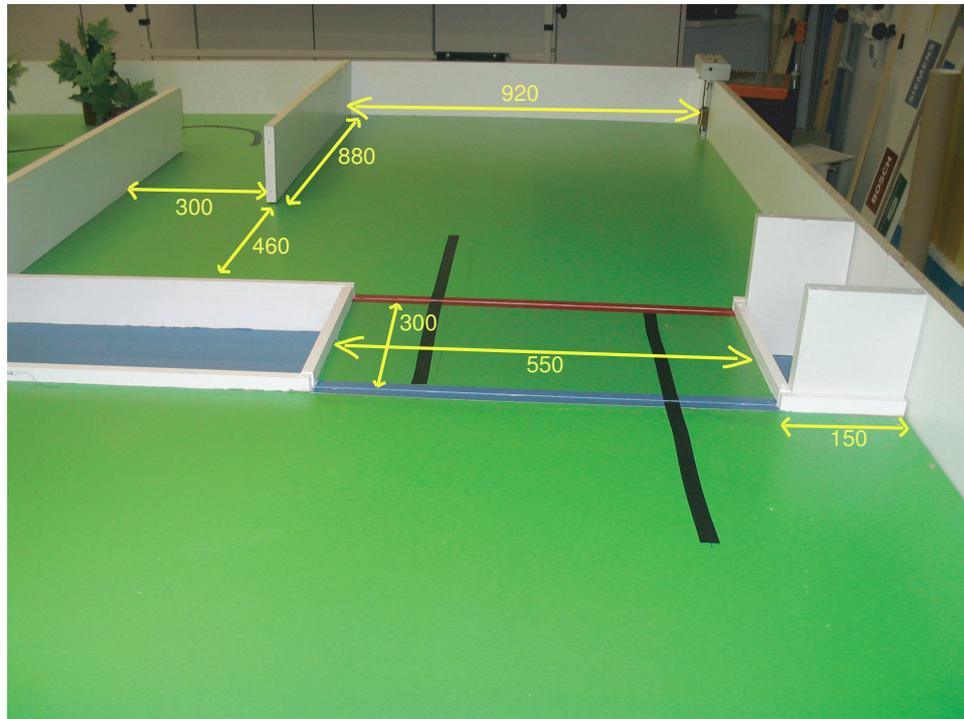


Abbildung 2.2: Die Brücke. Maße in mm

unbedingt notwendig ist. Befindet sich ein Stein im Roboter, so hat dieser bei einem Handeingriff im Roboter zu verbleiben. In keinem Fall dürfen bei einem Eingriff Steine außerhalb des Roboters bewegt werden oder sonstige Manipulationen am Spielfeld vorgenommen werden. Durch einen Handeingriff darf der Gegner nicht behindert werden. Gegebenenfalls ist abzuwarten, bis sich der Gegner entfernt hat.

Weitere Bestimmungen Vor dem Spiel wird die Spielfarbe (grau oder gelb) ausgelost. Diese Farbe entspricht der Farbe der zu sammelnden Steine (Granit oder Sandstein). Die Roboter können von den Teams vor dem Start innerhalb der ihrer Basis gegenüberliegenden Spielhälfte platziert werden, müssen aber unmittelbaren Kontakt zu der ihrer Klötzchenstation zugewandten Seite der äußeren Wand der Schlucht haben. Das Team mit der gelben Farbe muß seinen Roboter dabei zuerst platzieren.

Gegnerische Klötze dürfen nicht aus der gegnerischen Basis entfernt werden. Geschieht dies doch, werden die Klötze von den Schiedsrichtern zurück gelegt.

Es darf immer nur ein Bauklotz transportiert werden. Solange sich noch ein Klotz auf oder im Roboter befindet, ist es nicht gestattet, einen weiteren Klotz aufzuladen.

Fälschlicherweise in der Basis des Gegners abgelegte Klötze zählen als zwei Punk-

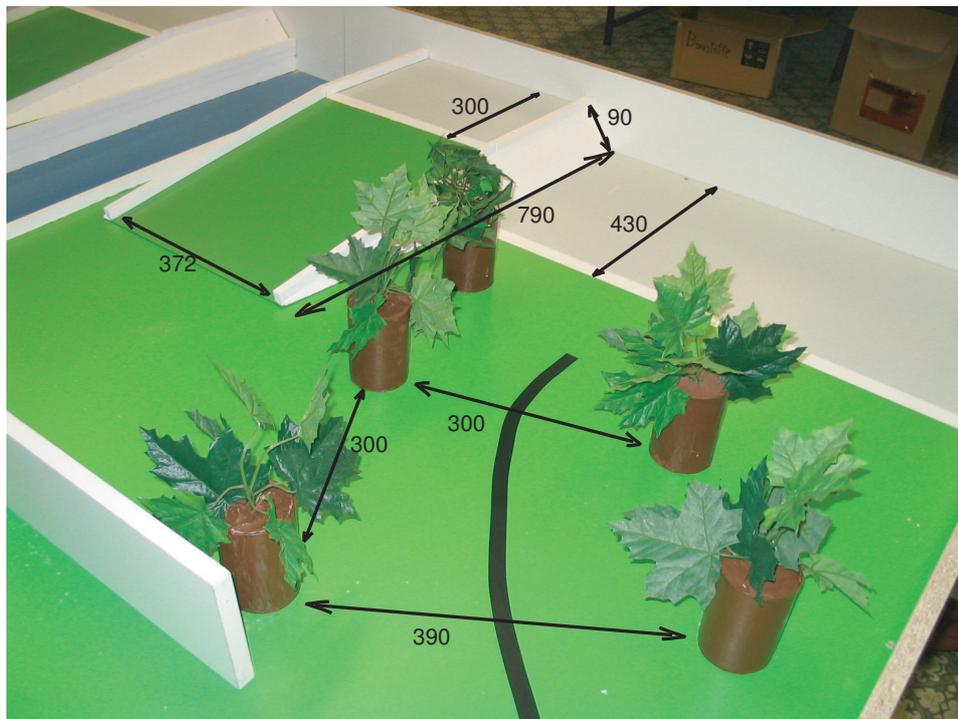


Abbildung 2.3: Der Wald. Maße in mm

te für den Gegner, unabhängig ob der Stein in der oberen oder unteren Basis liegt. Verliert ein Roboter seinen Stein während der Fahrt oder lädt ihn ausserhalb seiner Basis ab, darf er durch einen Handeingriff *nicht* wieder aufgeladen werden. Die Schiedsrichter können falsch abgelegte Steine vom Spielfeld entfernen. Die Brücke stellt eine Schlüsselstelle im Spiel dar. Besonders im Bereich der Brücke ist ein defensives Verhalten der Roboter absolut nötig, um Kollisionen mit dem Gegner zu vermeiden. Aggressive Roboter werden mit Punktabzügen bestraft. Über Strafpunkte entscheiden die Schiedsrichter direkt während oder unmittelbar nach dem Spiel. Sieger einer Spielrunde ist das Team mit der höheren Gesamtpunktzahl nach Abzug der Strafpunkte.

2.6 Zwischenwettbewerb

An einem Wochenende im Januar 2006 wird (voraussichtlich in Erfurt) ein Zwischenwettbewerb durchgeführt. Der genaue Termin wird noch bekannt gegeben. Aufgabe beim Zwischenwettbewerb ist, innerhalb von sieben Minuten soviel Steine wie möglich in beide Basen zu befördern. Die Steine dürfen also in beiden Heimatbasen abgelegt werden. Allerdings müssen die grauen Steine in der "grauen"

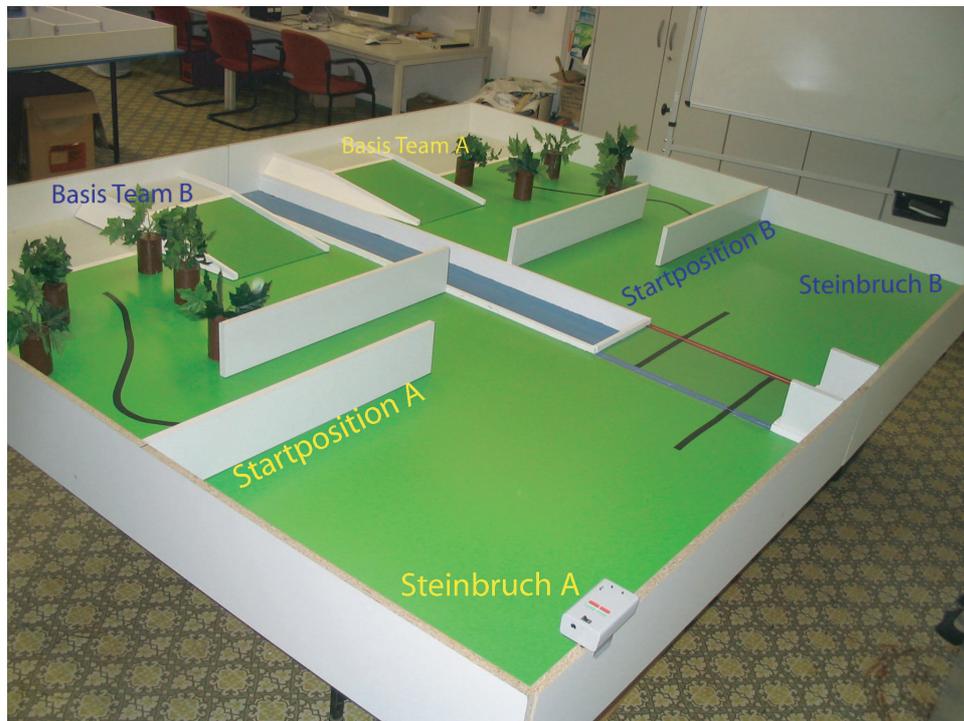


Abbildung 2.4: Startpositionen und Zuordnung Basis/Steinbruch

Basis abgelegt werden, die sandsteinfarbenen entsprechend in der anderen Basis. Die Roboter sind also gezwungen, die Brücke zu überqueren. Jeder Roboter tritt allein, ohne Gegner an. Die den Steinbrüchen zugewandten Wände der Schluchten werden beim Zwischenwettbewerb entfernt.

Ziel des Zwischenwettbewerbs ist es, die Teilnehmer zum zielstrebigem Arbeiten zu animieren. Deshalb werden sich nur die besten 16 aus 25 Teams für die Teilnahme an der Endrunde auf der Hannover Messe im April 2006 qualifizieren. Die restlichen Teams scheidern aus dem Wettbewerb aus.

Zusätzlich besteht die Möglichkeit, die Roboter ausführlich unter Wettbewerbsbedingungen zu testen. Teilnahmevoraussetzung zum Zwischenwettbewerb ist die Abgabe einer Dokumentation bis 30. November (siehe Dokumentationspflicht).

2.7 Hinweise für die Teamleiter

Aufgrund der in den letzten Jahren gemachten Erfahrungen möchten wir Sie auf einige organisatorische Fallstricke hinweisen.

In vielen Bundesländern gibt es dienstrechtliche Hürden, wenn ein Lehrer seine Schüler im privaten PKW zu Veranstaltungen befördern will. Bitte überlegen Sie rechtzeitig, wie Sie mit Ihrem Team nach Erfurt und nach Hannover anreisen können und wer die Kosten dafür übernimmt. Grundsätzlich liegt die Anreise im Ver-

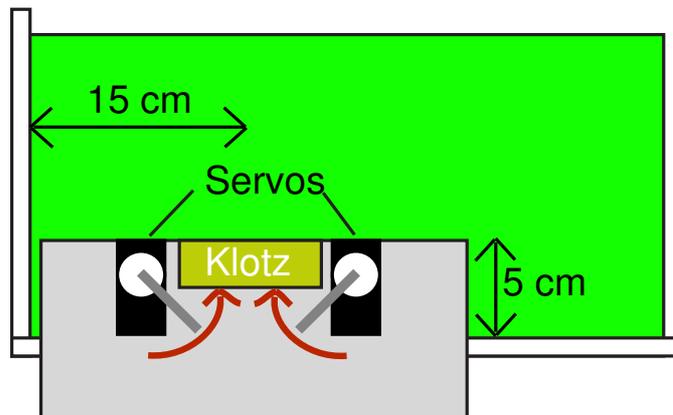


Abbildung 2.5: Spielfeldecke mit Steinbruch in der Draufsicht

antwortungsbereich der einzelnen Teamleiter.

Wir werden sowohl in Erfurt als auch in Hannover die Unterbringung in einer Jugendherberge organisieren. Sie brauchen diesbezüglich also nicht selbst aktiv zu werden. Die Übernachtungskosten müssen die Teilnehmer selbst tragen.

Die Endrunde des Wettbewerbs auf der Hannover Messe wird an zwei aufeinanderfolgenden Tagen (wahrscheinlich Donnerstag und Freitag der Messewoche) ausgetragen werden. Die Teilnehmer müssen dann bereits am Mittwoch anreisen und können an diesem Trainingstag ihre Roboter nochmals auf dem Spielfeld unter Wettkampfbedingungen testen. Es ist also erforderlich, die Schüler für zwei bis drei Tage vom Unterricht zu befreien. Bitte beachten Sie die jeweiligen Regelungen zur Freistellung an Ihrer Schule. Der Zwischenwettbewerb wird Samstag und Sonntag stattfinden, der genaue Termin wird noch festgelegt. Dazu werden wir über eine Umfrage im Forum den Teilnehmern Gelegenheit geben, ihre Terminwünsche einfließen zu lassen (Klausuren etc.).

Es ist empfehlenswert, sich zur Abdeckung der weiteren entstehenden Kosten um lokale Sponsoren zu bemühen.

Bitte lesen Sie regelmäßig zumindest den Bereich News & FAQ im Forum. Neuigkeiten, Hinweise, sowie Änderungen oder Ergänzungen der Regeln werden dort bekannt gemacht.

2.8 Termine

- 30. November 2005: Abgabe der Zwischendokumentation
- Januar 2006: Zwischenwettbewerb
- 24. bis 28. April 2006: Hannover Messe

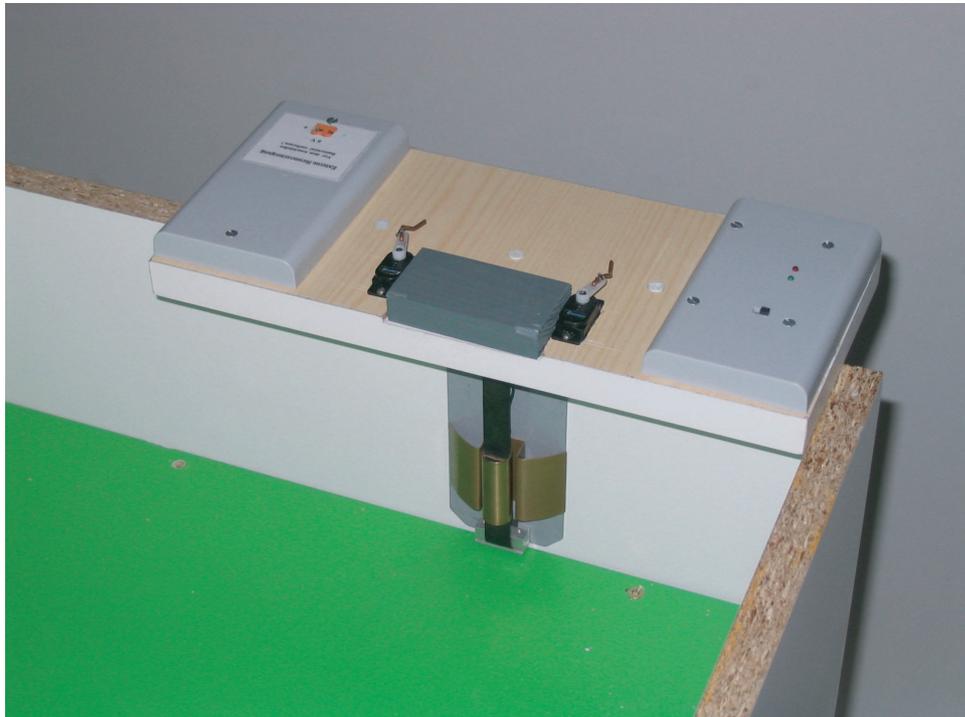


Abbildung 2.6: Der Steinbruch.

Teil II

Technische Informationen

Kapitel 3

Mechanik

3.1 Antriebskonzepte

Beim Entwurf eines Roboters kann man aus vielen verschiedenen Antriebskonzepten wählen, von denen jedes seine eigenen Vor- und Nachteile besitzt. Wir wollen hier die wichtigsten Antriebsformen vorstellen.

3.1.1 Differentialantrieb

Diese manchmal auch als "Panzerantrieb" bezeichnete Form des Antriebs besteht aus zwei separat angetriebenen Rädern. Der Roboter hat damit die Möglichkeit geradeaus zu fahren, auf der Stelle zu drehen und sich auf einer Kreisbahn zu bewegen.

Ein Nachteil des Differentialantriebes ist die Balance. Neben den beiden Antriebsrädern müssen zusätzlich ein oder zwei Freilauf- oder Stützräder angebracht werden. Ob ein oder zwei Stützräder nötig sind, hängt von der Gewichtsverteilung des Roboters ab.

Das Freilaufrad muß frei schwenkbar sein und einen möglichst großen Nachlauf haben, damit es leicht zur Seite wegklappen kann, wenn sich der Roboter dreht.

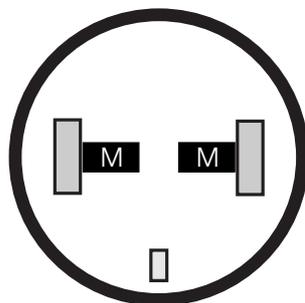


Abbildung 3.1: Schema Differentialantrieb

Anstelle eines Freilaufrades kann man auch eine Kugel o.ä. als dritten Auflage-

punkt benutzen. Die Kugel sollte auf einer Achse montiert sein, so dass sie sich bei Vorwärtsfahrt drehen kann. Bei seitlichen Bewegungen des Roboters rutscht die Kugel dann über den Untergrund.

Man sollte generell darauf achten, dass der Großteil des Gewichtes auf den Antriebsrädern lastet.

Ein weiteres Problem beim Differentialantrieb ist die Geradeausfahrt. Die Motoren drehen sich manchmal mit unterschiedlichen Geschwindigkeiten, obwohl an beiden die gleiche Spannung anliegt. Der Roboter driftet dann nach einer Seite ab. Diese Drehzahlabweichungen resultieren aus bauartbedingten Toleranzen der Motoren, einer ungleichmäßigen Gewichtsverteilung des Roboters oder unterschiedlichem Untergrund unter dem einem und dem anderen Antriebsrad.

Die Motorgeschwindigkeit muß dynamisch angepaßt werden. Dies geschieht durch eine Regelung, die die Motorgeschwindigkeit während der Fahrt überwacht und gegebenenfalls ändert. Eine Solche Regelung kann in Software oder Hardware realisiert werden.

3.1.2 Kettenantrieb

Hier wird der Roboter nicht von Rädern, sondern, wie ein Panzer, von zwei Ketten angetrieben. Diese Antriebsform entspricht im Wesentlichen dem Differentialantrieb, nachteilig ist der größere Realisierungsaufwand und das höhere Gewicht. Zu beachten ist auch der zumeist größere Schlupf der Ketten gegenüber Rädern.

3.1.3 Dreiradantrieb

Ein Motor treibt über ein Differentialgetriebe die beiden Hinterräder an, ein zweiter (Servo-) Motor steuert das Vorderrad.

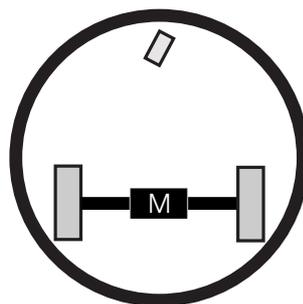


Abbildung 3.2: Schema Dreiradantrieb

Bei dieser Art des Antriebs muß die Geschwindigkeit der Antriebsräder nicht kontrolliert werden, um eine Geradeausfahrt zu erreichen. Es reicht aus, das Vorderrad geradeauszustellen.

Nachteilig ist hier, das der Roboter nicht auf der Stelle drehen kann.

3.1.4 Ackermannlenkung

Die von Autos bekannte Lenkung ist analog zum Dreiradantrieb, nur dass hier eben zwei synchron lenkbare Vorderräder vorhanden sind.

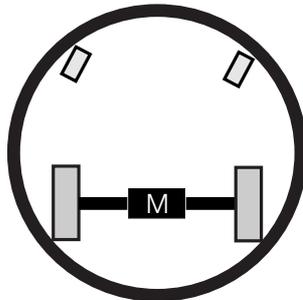


Abbildung 3.3: Schema Ackermannantrieb

3.1.5 Omni-Wheel Antrieb

Eine sehr interessante und enorm leistungsfähige Antriebsvariante ist der Omni-Direktional Antrieb. Unter Verwendung von drei (manchmal auch 4) angetriebenen sogenannten Allseitenrädern (auch Omni-Wheels, Mecanum-Räder) ist der Roboter in der Lage, sich in jede beliebige Richtung zu bewegen. Es sind also nicht nur die bekannten Bewegungsmuster Vorwärts-Drehen-Vorwärts usw. möglich, der Roboter kann auch einfach seitwärts fahren oder schräg nach hinten etc. Die exakte Regelung ist natürlich schwieriger als beim normalen Differentialantrieb. Wer bereit ist, diesen Mehraufwand in Kauf zu nehmen, wird mit ganz neuen Bewegungsmöglichkeiten für seinen Roboter belohnt.

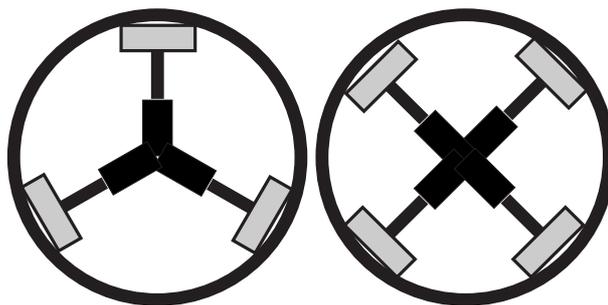


Abbildung 3.4: Schema Omniweheel-Antrieb mit drei und vier Rädern

Nachfolgend die Formeln zur Ansteuerung der Motoren für die Variante mit 3 Rädern.



Abbildung 3.5: Allseitenräder verschiedener Größe

Dabei bezeichnen V_x und V_y jeweils die x und y-Komponente des gewünschten Geschwindigkeitsvektors (im lokalen Koordinatensystem des Roboters), R ist der Abstand der Räder vom Mittelpunkt des Roboters. ω bezeichnet die Drehgeschwindigkeit, wobei man im Normalfall während des Fahrens keine zusätzliche Drehbewegung ausführen sollte, so dass $\omega = 0$ gesetzt werden kann.

Der Term $\frac{\sqrt{3}}{2}$ kommt von $\sin(120^\circ) = \frac{\sqrt{3}}{2}$ (der Winkel zwischen den Rädern beträgt gerade 120°). Das $-\frac{1}{2}$ ergibt sich dementsprechend aus $\cos(120^\circ)$

$$\begin{aligned} V_0 &= V_x - \omega R \\ V_1 &= -\frac{1}{2}V_x + \frac{\sqrt{3}}{2}V_y - \omega R \\ V_2 &= -\frac{1}{2}V_x - \frac{\sqrt{3}}{2}V_y - \omega R \end{aligned}$$

Die großen Allseitenräder (Durchmesser 6 cm) sind für 10 Euro bei krause-robotik.de erhältlich. Bei www.tec-shop.de gibt es die kleineren Räder (Durchmesser 4 cm).

3.2 Motoren

Nachdem wir uns im vorhergehenden Kapitel mit den verschiedenen Antriebskonzepten beschäftigt haben, wollen wir uns nun der Wahl eines passenden Motors zuwenden.

3.2.1 Dimensionierung

Im Katalog (Conrad,...) findet man verschiedene Angaben zu den Motoren, wie zum Beispiel Drehmoment, Drehzahl, Leistung usw.

Um aus diesen auf den ersten Blick verwirrenden Werten den richtigen Motor auswählen zu können, müssen wir ein wenig rechnen. Zunächst muß man sich über bestimmte Eckdaten des Roboters im Klaren sein. Dazu gehören:

- Geschwindigkeit v
- Gewicht m
- Raddurchmesser d

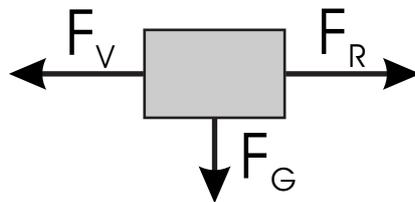


Abbildung 3.6: Kräfte am Roboter bei konstanter Geschwindigkeit

Betrachten wir zunächst, welche Kräfte wirken, wenn der Roboter mit konstanter Geschwindigkeit fährt. F_V ist die Kraft, die den Roboter vorwärts treibt, die sogenannte Vortriebskraft. F_R bezeichnet die Reibungskraft, die der Vortriebskraft entgegengesetzt wirkt. Allgemein gilt:

$$F_R = \mu \cdot F_N \quad (3.1)$$

μ ist die Reibungszahl. Welche Werte man hier einsetzen muß, kann man in einem Tafelwerk nachschlagen. F_N ist die sogenannte Normalkraft und berechnet sich aus $F_N = F_G \cdot \cos\alpha$. Da der Roboter keine geneigte Ebene herauf fahren soll, sondern sich stets in der Waagerechten bewegt, ist $\alpha = 0^\circ$. Damit gilt also

$$\begin{aligned} F_R &= \mu \cdot F_G \cdot \cos 0^\circ \\ &= \mu \cdot F_G \\ &= \mu \cdot m \cdot g \end{aligned} \quad (3.2)$$

Da sich der Roboter mit gleichbleibender Geschwindigkeit bewegt, heben sich F_V und F_R gerade gegenseitig auf.

$$F_V = F_G = \mu \cdot m \cdot g \quad (3.3)$$

Wir wählen für die Reibungszahl μ die Gleitreibung von Reifen auf Asphalt, also $\mu = 0,5$. Damit ist die im Antriebssystem auftretende Reibung sehr großzügig dimensioniert.

Nun können wir uns an die Berechnung der Leistung des Motors machen. Es gilt:

$$P = F_V \cdot v \quad (3.4)$$

v steht, wie oben erwähnt, für die zu erreichende Höchstgeschwindigkeit. Setzen wir nun die Werte ein, so erhalten wir:

$$P = \mu \cdot m \cdot g \cdot v \quad (3.5)$$

Wir schätzen das Gesamtgewicht m des Roboters auf $2kg$. Als Geschwindigkeit geben wir $20 \frac{cm}{s}$ vor.

$$\begin{aligned} P &= 0,5 \cdot 2kg \cdot 9,81 \frac{m}{s^2} \cdot 0,2 \frac{m}{s} \\ &= 1,96W \end{aligned} \quad (3.6)$$

Da wir zwei Motoren haben, entfällt auf jeden Motor eine Leistung von

$$P \approx 1W \quad (3.7)$$

Nun müssen wir noch berechnen, wieviele Umdrehungen pro Sekunde die Antriebswelle machen muß, damit wir mit unserem Raddurchmesser auf die gewünschte Geschwindigkeit kommen.

Angenommen, unsere Räder haben einen Durchmesser von $d = 5cm$. Dann gilt

$$\begin{aligned} n &= \frac{v}{\pi \cdot d} \\ &= \frac{0,2 \frac{m}{s}}{\pi \cdot 0,05m} \\ &= 1,3 \frac{1}{s} \end{aligned} \quad (3.8)$$

n gibt also die erforderliche Drehzahl in Umdrehungen pro Sekunde an.

Mit diesen errechneten Werten, kann ich nun einen geeigneten Motor auswählen. Ich schaue zum Beispiel im Conrad Modellbau-Katalog auf Seite 263. Dort gibt es einen Getriebemotor mit der Bezeichnung 33G-50 mit $n = 1,4 \frac{1}{s}$. Weiterhin sind im Katalog noch mehrere Werte angegeben, so zum Beispiel verschiedene

Drehmomente. Es findet sich jedoch keine Angabe zur Leistung. Doch aus den gegebenen Werten kann man die Leistung des Motors einfach berechnen.

Es gilt:

$$P = M \cdot \omega \quad (3.9)$$

M ist dabei das Drehmoment bei der Winkelgeschwindigkeit ω . Die Winkelgeschwindigkeit ist einfach ein anderer Begriff für die Drehzahl und berechnet sich aus

$$\omega = 2\pi \cdot n \quad (3.10)$$

Rechnen wir also:

$$\begin{aligned} P &= M \cdot \omega & (3.11) \\ &= 0,2Nm \cdot 2\pi \cdot 1,4 \frac{1}{s} \\ &\approx 1,8W \end{aligned}$$

Der Motor bringt also viel mehr Leistung, als wir oben berechnet hatten. Laut Angaben im Katalog liegt das Spitzendrehmoment, was kurzzeitig bei großer Belastung geleistet werden kann, 3 mal so hoch, wie das Dauerdrehmoment, mit dem wir gerechnet haben. Daher ist sichergestellt, daß der Roboter auch sehr schnell anfahren kann und schnell beschleunigt.

Man kann nun noch nach weiteren Motoren suchen, die ebenfalls die errechneten Leistungswerte erfüllen. Ein weiterer Kandidat könnte zum Beispiel einer der Motoren aus der Reihe RB-35 sein. Insgesamt sollte man die Motoren lieber etwas über- als unterdimensionieren.

Einen wichtigen Punkt haben wir in unseren bisherigen Betrachtungen noch nicht beachtet: Strom- und Spannungsbedarf der Motoren. Die meisten Motoren arbeiten mit Spannungen von 6 bis 12 Volt, sind also für unsere Zwecke gut geeignet. Die Standardmotortreiber L293D, die auch auf dem Handyboard verbaut sind, dürfen nur mit maximal $600mA$ belastet werden, ansonsten werden sie zerstört! Darum sollte man darauf achten, daß der Dauerstrom unterhalb dieses Wertes liegt. Kurzfristige Stromspitzen verkraften die Motortreiber, man sollte sie aber nicht ständig am Limit betreiben.

Der im Beispiel oben ausgewählte Motor 33G-50 hat einen Dauerstrom von $380mA$, liegt also innerhalb des tolerablen Bereichs.

Es gibt durchaus noch andere Motortreiber, die einen höheren Dauerstrom zulassen. Siehe dazu Kapitel 3.2.2.

Interessant ist vielleicht auch die Betrachtung des Wirkungsgrades der Motoren. Aus Spannung und Strom lässt sich ja die aufgenommene elektrische Leistung berechnen.

$$P = U \cdot I \quad (3.12)$$

In unserem Beispiel also

$$\begin{aligned} P &= 12V \cdot 0,38A & (3.13) \\ &= 4,56W \end{aligned}$$

Nun hatten wir doch berechnet, dass der Motor nur $1,8W$ mechanische Leistung abgibt. Die Differenz zwischen aufgenommener elektrischer Leistung und abgegebener mechanischer Leistung nennt man Verlustleistung. Sie geht dem System in Form von Abwärme durch Reibung "verloren". Den Quotienten aus abgegebener und aufgenommener Leistung nennt man Wirkungsgrad. Es ist:

$$\begin{aligned}\eta &= \frac{P_{ab}}{P_{auf}} & (3.14) \\ &= \frac{1,8W}{4,56W} \\ &\approx 40\%\end{aligned}$$

Nur etwa 40% der aufgenommenen elektrischen Energie wird also in nutzbare mechanische Energie umgewandelt. Der Wirkungsgrad bei Elektromotoren ist sehr unterschiedlich. Billigere Motoren haben einen niedrigen Wirkungsgrad und teurere einen höheren. Neben den üblichen Gleichstrommotoren gibt es noch Bürstenlose Motoren, die ähnlich den Schrittmotoren aufgebaut sind, sich aber nicht in Schritten steuern lassen. Diese Motoren erreichen mitunter einen Wirkungsgrad deutlich über 90%. Sie sind jedoch nicht ohne Weiteres an das Handyboard anschließbar. Schaltet man aber entsprechende Fahrtregler für bürstenlose Motoren dazwischen (Servoanschluß), können auch diese Motoren angesteuert werden.

3.2.2 Ansteuerung

An das Handyboard können direkt bis zu vier Motoren angeschlossen werden. Die Motoren werden über spezielle Treiberstufen, sog. Motortreiber vom Typ L293D direkt mit der Akkuspannung versorgt. Bei der Auswahl der Motoren ist auch auf die maximale Stromaufnahme zu achten. Das Maximum liegt bei 600 mA pro Motor. Fließen höhere Ströme, werden die Motortreiber oder sogar das gesamte Handyboard zerstört. Ein zum L293D pinkompatibler Baustein ist der SN754410 NE, der Ströme von bis zu 1A pro Kanal verkraftet. Ein weiterer oft benutzter Motortreiber ist der L298, der für 2A Dauerbelastung ausgelegt ist. Beide sind unter anderem bei embedit.de erhältlich.

Um ein sauberes Fahren zu ermöglichen, können die Motoren beim Handyboard nicht nur ein- und ausgeschaltet werden, sondern auch in ihrer Drehgeschwindigkeit geregelt werden. Das dabei zum Einsatz kommende Verfahren nennt sich "Pulsweitenmodulation", oder kurz PWM.

Dabei wird der Motor nicht kontinuierlich mit Strom versorgt, sondern in kurzen Impulsen. Je nach Verhältnis zwischen Impuls und Pause dreht sich der Motor schneller oder langsamer. Abbildung 3.7 verdeutlicht die Arbeitsweise.

Hierbei handelt es sich nicht etwa um Wechselstrom, sondern vielmehr um einen gepulsten Gleichstrom.

Unter Interactive C ist die Ansteuerung der Motoren einfach. Der Befehl `motor(1, 70)` lässt den Motor 1 mit 70% Geschwindigkeit vorwärts drehen. `motor(1, -100)`

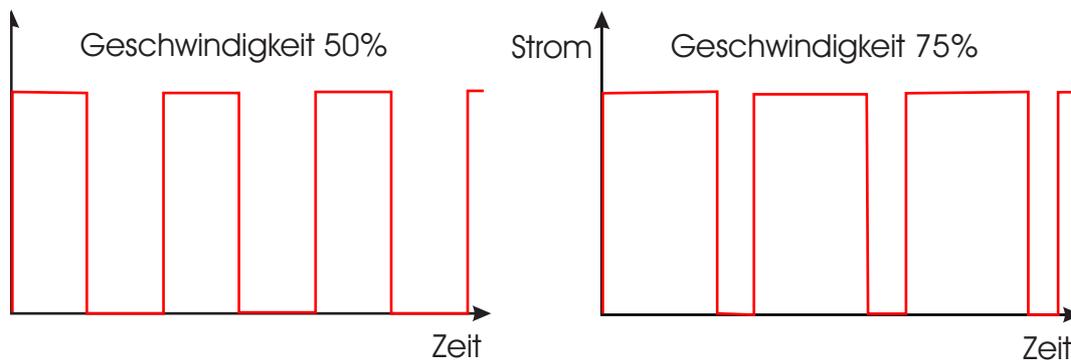


Abbildung 3.7: Pulsweitenmodulation

dreht den Motor mit voller Geschwindigkeit zurück. Diese Betrachtungsweise ist jedoch noch etwas ungenau. Die Motoren haben eine nichtlineare Kennlinie, das bedeutet, bei einer PWM-Rate von 70% erbringt der Motor nicht 70% Leistung, sondern einen davon abweichenden Wert. Wie genau sich die Motoren verhalten, kann man natürlich ausmessen. In den meisten Fällen ist eine solche Linearisierung der Motorkennlinie aber nicht notwendig. Grob abgeschätzt quadriert man wegen $P = U \cdot I$ einfach die PWM-Rate um auf die erbrachte Leistung zu kommen. Bei einer Rate von 70% werden also $0,7 \cdot 0,7 = 0,49$, also 49% der Maximalleistung erbracht.

Bei vielen AVR oder PIC-Controllern finden sich PWM-Ausgänge, die bereits hardwareseitig angesprochen werden. Man programmiert lediglich einige Steuerregister, die Signalerzeugung übernimmt die Hardware unabhängig vom weiter laufenden Programm.

3.2.3 Servomotoren

Neben den soeben betrachteten Gleichstrommotoren, die vornehmlich zum Antrieb des Roboters geeignet sind, gibt es noch eine andere sehr nützliche Art von Motoren.

Servomotoren kommen sehr oft in ferngesteuerten Modellautos als Stellmotor für die Lenkung zum Einsatz. Ihre Fähigkeit, eine vorgegebene Position sehr schnell mit hoher Genauigkeit anzufahren und auch gegen Belastung zu halten, machen Servomotoren auch für viele Anwendungen bei mobilen Robotern interessant.

Servomotoren bestehen aus einem kleinem Gleichstrommotor, einem Getriebe, Endanschlügen, über die die Welle nicht hinweg drehen kann, einem Potentiometer um die aktuelle Position festzustellen und einem IC zur Positionssteuerung.

Servos besitzen neben den obligatorischen Leitungen für Masse und Phase eine dritte Leitung, die sogenannte Steuer- oder Signalleitung. Über diese Leitung wird dem Servo die anzufahrende Position mitgeteilt. Dies geschieht ähnlich der oben angesprochenen PWM-Methode: Dabei wird alle 20ms ein Impuls von 1 bis 2 ms Länge gesendet, je nach gewünschter Position.

Das Handyboard bietet Anschlußmöglichkeiten für 5 Servomotoren und stellt komfortable Befehle zum Ansprechen der Servos zur Verfügung: `servo2=2200` etwa stellt den zweiten Servo in die Mittelposition. Die beiden Endanschläge sollten vorsichtig ausgelotet werden, meist liegen sie bei etwa 800 und 3600. Der Servo kann beschädigt werden, wenn er ständig überdreht wird.

Die Ansteuerung von Servos ist natürlich auch mit anderen Controllern möglich. Hier ist das Datenblatt zu Rate zu ziehen.

Ein interessanter Baustein zur Servoansteuerung ist der SD20, ein vorprogrammierter PIC 16F872. Dieser kann bis zu 20 Servos unabhängig von einander ansteuern. Über einen I^2C -Bus werden einfache Befehle an den SD20 geschickt. Erhältlich ist der Baustein bei www.roboter-teile.de für etwa 11 Euro.



Abbildung 3.8: typischer Standard-Servo

3.2.4 Servomotoren als Antriebsmotor

Neben dem Einsatz als Stellmotoren lassen sich Servos nach einigen Eingriffen auch gut als Antriebsmotoren einsetzen. Sie sind kleiner und leichter als "normale" Getriebemotoren und lassen sich aufgrund ihrer kompakten, rechteckigen Form oftmals leichter verbauen.

Um handelsübliche Servo zu einem Antriebsmotor umzubauen, ist ein nicht mehr rückgängig zu machender Eingriff ins Innenleben des Servos nötig. Unter anderem muß der Endanschlag (meist eine kleine Plattenase) entfernt werden, sowie das Potentiometer von der Endachse entkoppelt werden.

Die Seite www.electronicplanet.ch/Schaltun/Servo/Servo.html gibt ein kleines Tutorial zum Thema.

Einfacher ist der Einsatz von Servos, die von vornherein ohne Elektronik verkauft werden. Bei diesen Typen braucht nur der Endanschlag entfernt zu werden. Conrad verkauft solche Servos unter der Bestellnummer 225137-68 im Modellbaukatalog Seite 206. Wer sich viel Mühe gibt, kann sich ins Servogehäuse noch einen Radencoder hineinbasteln.

Kapitel 4

Mikrocontroller

Die Wahl eines geeigneten Mikrocontrollers bzw. eines Mikrocontrollerboards ist sicher eine der ersten und grundlegendsten Entscheidungen, die die Teilnehmer zu treffen haben. Um die Wahl gerade für Einsteiger in die Materie zu vereinfachen, wollen wir an dieser Stelle einige auf dem Markt verfügbare Systeme vorstellen.

4.1 Das Handyboard

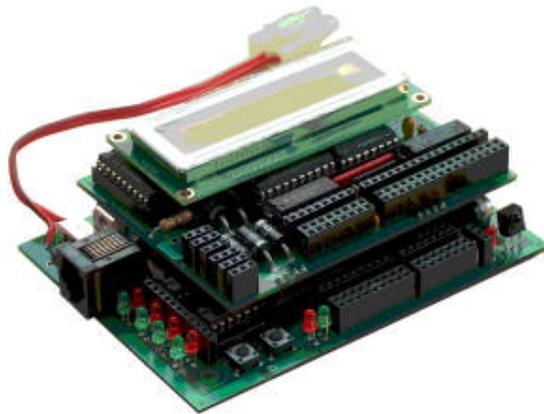


Abbildung 4.1: Handyboard

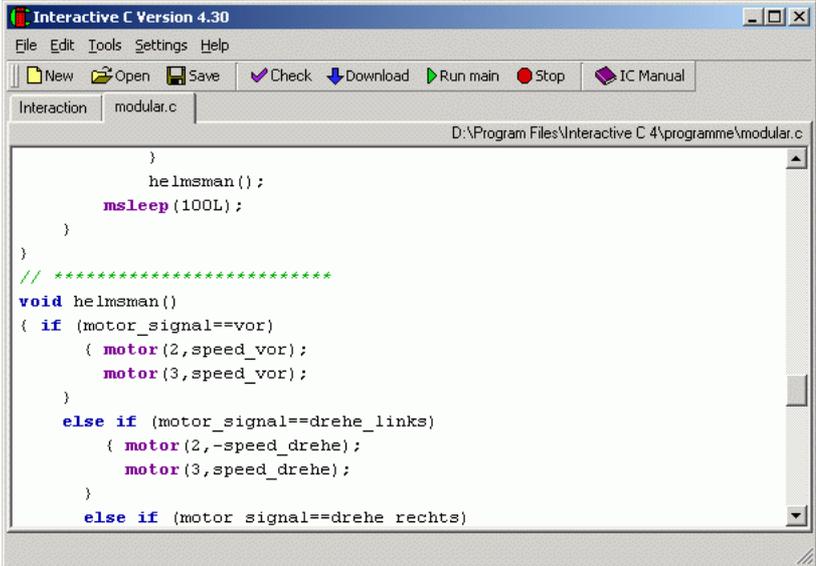
Das Handyboard ist ein am MIT entwickeltes Mikrocontrollerboard auf Basis eines Motorola 68HC11. Wer den RoboKing-Wettbewerb als Einstieg in die Welt der Mikrocontroller nutzen will, findet mit dem Handyboard eine "Rundum-Sorglos-Lösung".

4 Motoren und 6 Servos lassen sich einfach direkt an das Board anschließen. Zur Interaktion mit dem Benutzer (Debugging) sind 2 Taster und ein LCD vorhanden. Die digitalen und analogen Ein- bzw. Ausgänge sind übersichtlich angeordnet. Neben jedem Port liegen Spannungs- und Masse-Buchsen, wodurch eine einfache und

übersichtliche Verkabelung möglich ist.

Programmiert wird das Handyboard in Interactive-C, einem speziell angepassten C-Dialekt. Interactive-C besteht neben dem eigentlichen Compiler auch aus einer komfortablen Entwicklungsumgebung, aus der heraus man seine Programme compilieren und auf das Board laden kann. Der sogenannte Interaction-Modus ermöglicht das Ausführen von einzelnen Befehlen, die über eine Kommandozeile eingegeben werden können, oder das Abfragen von Variablen. IC stellt alle benötigten Funktionen zum Ansprechen der Ports und der Motoren, des LCDs usw. zur Verfügung. Sogar Multitasking, also das parallele Abarbeiten mehrerer Prozesse ist möglich.

Das Handyboard in der RoboKing-Spezial Ausstattung enthält zusätzlich ein I^2C -Interface und eine Speicherhalteschaltung, die den Inhalt des Programmspeichers auch nach Abziehen des Akkus noch etwa 48 Stunden hält.



```

Interactive C Version 4.30
File Edit Tools Settings Help
New Open Save Check Download Run main Stop IC Manual
Interaction modular.c
D:\Program Files\Interactive C 4\programme\modular.c

    }
    helmsman();
    msleep(100L);
}
// *****
void helmsman()
{ if (motor_signal==vor)
  { motor(2,speed_vor);
    motor(3,speed_vor);
  }
  else if (motor_signal==drehe_links)
  { motor(2,-speed_drehe);
    motor(3,speed_drehe);
  }
  else if (motor signal==drehe rechts)

```

Abbildung 4.2: Interactive C

Features:

- Motorola 68HC11 mit 4 MHz, 32kByte Programmspeicher (16kB frei für Nutzercode und -daten)
- 16 analoge Eingänge, die auch als digitale Ausgänge genutzt werden können
- 9 digitale Eingänge, 8 digitale Ausgänge
- 4 PWM-Ausgänge zur Ansteuerung von DC-Motoren

- 6 Ausgänge zum Ansteuern von Servos
- 4 Eingänge für LEGO-Mindstorms-Sensoren
- 1 Infrarotempfänger, 1 Infrarotsender
- IC-Bus, serielle Schnittstelle
- Speicherhalteelko
- LCD-Display
- Versorgungsspannung: 7,2 - 15,0 Volt

Das Handyboard gibt es bei www.krause-robotik.de zum Preis von 250 Euro. Zum Handyboard wird neben dem Bord inkl. Erweiterungsplatine das Programmierkabel, Software (Interactive C, Assembler, die Schaltpläne und Handbücher auf Englisch mitgeliefert). Am Bord befindet sich ein Tomya-Stecker für den direkten Anschluss von Modellbauakkus.

4.2 Atmel AVR

Die Firma Atmel hat mit den AVR's eine ganze Familie von 8-Bit RISC Mikrocontrollern auf den Markt gebracht, die sich nicht nur unter Roboterbastlern größter Beliebtheit erfreuen. Die AVR sind in einer Vielzahl von Typen erhältlich, jeder mit unterschiedlichen Features und Einschränkungen. Eine Übersicht aller erhältlicher Typen findet man auf www.atmel.com/dyn/products/param_table.asp?family_id=607.

4.2.1 Programmierung

Bevor wir uns einige Controllerboards im Detail anschauen, wollen wir einen Blick auf die Programmiermöglichkeiten werfen.

Neben der Wahl der Programmiersprache (siehe unten) stellt sich eine dringende Frage: Wie bekommt man das kompilierte Programm auf den Controller? Glücklicherweise sind alle AVR's mit einer ISP-Schnittstelle (In Socket Programming) ausgerüstet, die das Aufspielen des Codes denkbar einfach macht. Man benötigt trotzdem ein Programmierdongle (quasi einen Adapter zwischen PC und Controller) und natürlich ein entsprechendes Kabel.

Einen solchen Programmieradapter mit Kabel zum Anschluß an die parallele Schnittstelle gibt es zum Beispiel bei embedit.de für knapp 12 Euro. Wer sich den Adapter lieber selbst bauen möchte, findet auf rumil.de/hardware/avrISP.html eine entsprechende Anleitung. Die beiden dargestellten Adapter sind bis auf die Stecker (6 oder 10 polig) identisch.

Neben dem Programmieradapter ist noch eine passende Software nötig. Weit verbreitet ist Ponyprog, das es unter www.lancos.com/prog.html zum Download gibt.

Programmierung in C Für die AVR's sind Compiler für verschiedene Hochsprachen erhältlich. Ein wie gewohnt freier C-Compiler ist der gcc. Die aktuelle Version für Windows-Umgebungen ist unter winavr.sourceforge.net oder in unserem Downloadbereich erhältlich. Neben dem eigentlichen Compiler sind natürlich noch Bibliotheksfunktionen zum Ansprechen der einzelnen Features (IO-Ports, AD-Wandler, I^2C , UART, PWM usw.) erforderlich. Eine umfangreiche Bibliothek gibt es unter <http://procyonengineering.com/avr/avr-lib/>. www.kreatives-chaos.com/index.php?seite=avrgcc erklärt ausführlich die ersten Schritte bei Installation und Einrichtung des AVR-gcc.

Programmierung mit Pascal Die Firma E-Lab (elab.de) bietet mit AVRco einen Pascal Compiler für fast alle AVR-Typen an. AVRco umfasst neben dem eigentlichen Compiler auch eine grafische Nutzeroberfläche und umfangreiche Funktionsbibliotheken. Die frei erhältliche Demo-Version ist allerdings auf 2kb Code-Größe beschränkt.

Programmierung mit Basic Ein kostenloser Compiler für Basic ist ebenfalls verfügbar, allerdings ähnlich wie beim AVRco nur als Demo mit eingeschränktem Code-Umfang. Unter www.mcselec.com/bascom-avr.htm gibt es weitere Informationen.

4.2.2 Mega32 Board von Krause-Robotik

Betrachten wir nun einige Controllerboards auf AVR-Basis:

Das AtmelBoard ist eine Neuentwicklung mit einem ATmega32 als Controller. Das Board enthält alles, was man für die Steuerung von mobilen Robotern, vielen Versuchsaufbauten und kleinen Anwendungen braucht. So lassen sich zwei Motoren an den vorhandenen Motortreiber anschließen. 2 Servos lassen sich ebenfalls direkt am Board betreiben (incl. Spannungsversorgung). Ein Display (LCD, 2x16) rundet das Board ab. Als besonderes Feature ist im Gegensatz zu den anderen Boards kein externer Programmieradapter notwendig, man braucht nur ein 9-poliges serielles Kabel. Das Board wird mit auf der Procyon-Lib aufbauenden Bibliotheksfunktionen geliefert, die das Ansprechen der verschiedenen Funktionen auch für Programmieranfänger einfach machen.

Features:

- ATmega32 DIP mit 16 MHz, 32kByte Programmspeicher, 2kB SRAM, 1kB EEPROM
- 8 analoge Eingänge, 24 digitale Eingänge, 20 digitale Ausgänge
- 2 Motorausgänge für Gleichstrommotoren je 1A

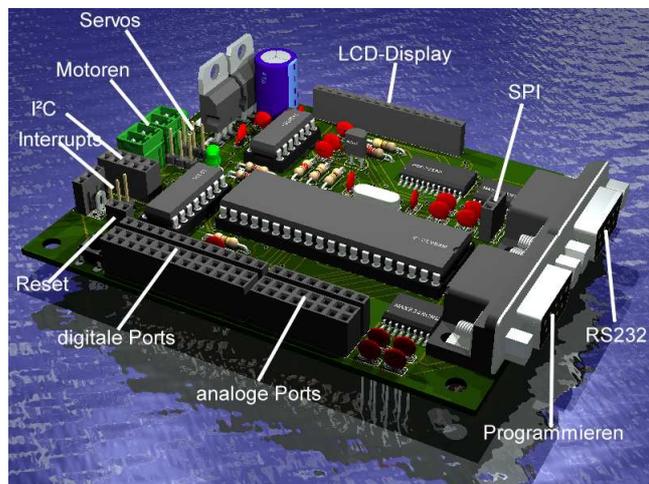


Abbildung 4.3: das Mega32 Board

- 2 Servoausgänge zur Ansteuerung handelsüblicher Servos aus dem Modellbaubereich
- 1 I^2C Interface
- LCD-Display 2x16 Zeichen, direkt auf das Board gesteckt
- 1 serielle Schnittstelle (D-SUB-9 Stecker)
- 1 serielle Schnittstelle mit TTL-Pegel
- 3 Interruptports (bereits nach außen geführt)
- getrennte Spannungsversorgungen für Mikrocontroller, Servos und Motoren
- jeder Kreis mit Sicherungen abgesichert

Erhältlich ist das Board zum Preis von 155 Euro bei www.krause-robotik.de.

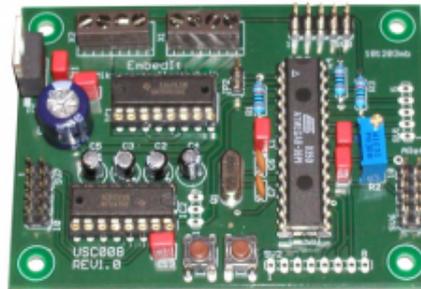
4.2.3 USC008 Board von Embedit

Das USC008 Board ist eine kleine Steuerungsplatine, basierend auf einem ATmega8-16PI von Atmel. Ein bestückter SN754410 erlaubt den Betrieb von zwei Motoren (max. 1A). Zur Kommunikation mit dem PC ist ein RS232 Treiber bestückt, für Erweiterungen ist ein I2C Port vorhanden.

Die Funktionen im Überblick:

- ATmega8 DIP mit 16 Mhz
- SN754410 Motortreiber (2 Motoren, je 1A maximal)

- MAX232 Treiberbaustein
- 5V Spannungsregler
- 10 poliger ISP Anschluß zum einfachen Programmieren
- I2C Port
- 2 Taster
- 6 digitale I/O und 4 AD-Wandler



[C] embedit.de

Abbildung 4.4: das USC008 Board mit ATmega8

Erhältlich ist das Board bei www.embedit.de für etwa 30 Euro.

4.2.4 ATmega128 Platine

Ebenfalls bei embedit.de gibt es eine Platine mit aufgelötetem ATmega128 Controller. Der Mega128 ist der leistungsfähigste und größte Controller aus der AVR-Reihe.

Alle 56 Ports sind über Stiftleisten im Rastermaß 2,54 mm nach außen geführt. Davon können 8 als Analog-Eingänge verwendet werden. Für die Programmierung ist eine 10 polige Stiftleiste vorhanden. Sie ist kompatibel zu der STK200 Programmierschnittstelle, wodurch eine Vielzahl von Programmiergeräten zur Verfügung steht. Das Board kann wahlweise mit 5V stabilisiert oder mit 8-12V unstabilisiert versorgt werden. Ein 16Mhz Quarz ist ebenfalls aufgelötet.

Features:

- ATmega128-16AI Mikrocontroller (16 MHz, 128kB Programmspeicher, 4 kB RAM, 4 kB EEPROM)
- 53 I/O-Ports



Abbildung 4.5: ATmega128 Platine

- 8 Kanal 10 Bit AD-Wandler
- 6 PWM Kanäle (für Motoren und Servos, aber zusätzliche Treiber erforderlich)
- 2x Full Duplex USART
- I2C Bus
- 10 polige Stiftleiste für In System Programming, kompatibel zu STK200
- Spannungsversorgung mit 5V stabilisiert oder 8-12V ungestabilisiert durch Spannungsregler

Die Platine enthält keine Motortreiber, Taster oder Anzeigeelemente. Um den Mega128 sinnvoll einsetzen zu können, muß also noch eine Trägerplatine mit der benötigten Peripherie angefertigt werden.

Der Kaufpreis bei embedit.de beträgt 35 Euro.

4.2.5 ATmega64 Platine von Embedit

Als RoboKing-Special gibt es von www.embedit.de eine Mega64-Platine zum Preis von 35 Euro. Die bei uns kostenlos erhältliche AVRco Version arbeitet ausschließlich mit dem ATmega64 zusammen. Wer sich diese Platine oder einen einzelnen ATmega64 bestellt, bekommt die CD mit dem Compiler zugeschickt.

Wie bei der Mega128 Platine sind auch hier alle Ports nach außen geführt und liegen im Rastermaß 2,54 mm vor. Zusätzlich enthält die Platine zwei serielle Schnittstellen (RS232). Die entsprechenden Treiber befinden sich ebenfalls auf dem Board. Features:

- ATmega64-16AI Mikrocontroller (16 MHz, 64kB Programmspeicher, 4 kB RAM, 2 kB EEPROM)
- zwei serielle Schnittstellen (RS232) incl. bestückten Treibern

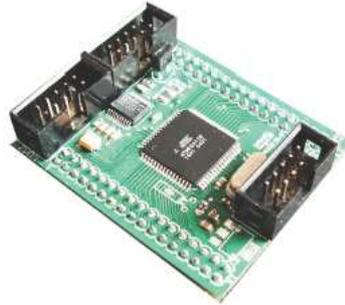


Abbildung 4.6: ATMega64 Platine mit zwei seriellen Ports

- sonst gelten die gleichen Daten wie bei der Mega128 Platine

Die Platine enthält keine Motortreiber, Taster oder Anzeigeelemente. Um den Mega64 sinnvoll einsetzen zu können, muß also noch eine Trägerplatine mit der benötigten Peripherie angefertigt werden.

Da die Platine ein Spezial-Angebot ist, befindet es sich nicht im Webshop, sondern muß direkt per Mail an info@embedit bestellt werden.

4.3 weitere Controller Familien

Neben den AVR's sind die PIC-Controller der Firma Microchip sehr verbreitet. Eine gute Informationsquelle zum Einstieg in die PIC-Welt ist www.sprut.de

Die bei Einsteigern recht beliebte C-Control von Conrad ist für die RoboKing Aufgabe aufgrund des geringen Leistungsumfanges nicht geeignet. Die C-Control II verwendet einen 16bit Controller von Infineon und ist daher laut Reglement verboten.

Kapitel 5

Stromversorgung - Akkus

Das Handyboard sowie nahezu alle auf dem Markt verfügbaren Controller, Sensoren und ICs benötigen eine stabilisierte Betriebsspannung von 5 Volt. Zum Betreiben von Motoren und Servomotoren ist aber eine höhere Spannung notwendig.

Auf dem Handyboard ist ein Spannungswandler vom Typ L78S05 integriert, der die angelegte Akkuspannung auf die benötigten 5 Volt herunterregelt. Die Motoren werden direkt mit der anliegenden Akkuspannung betrieben. Servomotoren arbeiten nur mit Spannungen von 4,8 bis 6 Volt. Auf dem Handyboard befinden sich 5 Dioden, die zusammen die Akkuspannung um ca. 3,5 Volt verringern. An jeder Diode fallen etwa 0,7 Volt ab. Damit beträgt die maximal verwendbare Akkuspannung 9,6 Volt, da $9,6V - 3,5V = 6,1V$, was für die Servos gerade noch zulässig ist. Man sollte die Spannung auch nicht geringer wählen, um die Leistungsfähigkeit der Motoren voll auszunutzen.

9,6 Volt erreicht man zum Beispiel mit 8 Mignon-Zellen zu je 1,2 Volt.

Der Spannungsregler 78S05 ist natürlich für wenige Cent auch einzeln erhältlich, zum Beispiel bei Reichelt. Von der 7805-Serie gibt es drei Typen (78L05, 7805, 78S05), die sich jeweils in ihrer maximalen Strombelastbarkeit unterscheiden. (100mA, 1A, 2A). Spannungsregler für andere oft benötigte Spannungen sind ebenfalls erhältlich, zum Beispiel der 7806 für 6 Volt usw. Beim Einsatz der Spannungsregler sollten auf keinen Fall die Stützkondensatoren zum Glätten der Spannung vergessen werden. Welche Kondensatoren sinnvoll einzusetzen sind, kann den entsprechenden Datenblättern entnommen werden.

Allen diesen Reglerbausteinen gemeinsam ist, dass sie sehr schnell heiß werden. Das sorgt oftmals für Irritationen. Folgende Überlegung erklärt die hohen Temperaturen. Angenommen, wir haben eine Eingangsspannung von 10V. Der 7805 regelt diese auf 5V herunter. Die nachfolgende Schaltung (zum Beispiel ein Mikrocontroller mit einigen Sensoren) benötigt vielleicht einen Strom von 1A. Dieser Strom wird ebenfalls aus dem Akku gezogen, fließt also in den 7805 hinein. Wir haben also eine aufgenommene Leistung von $10V \cdot 1A = 10Watt$. Die abgegebene beträgt aber nur $5V \cdot 1A = 5Watt$. Die Differenz, also 5 Watt, wird im 7805 in Wärme umgesetzt. Man braucht sich also nicht zu wundern, warum man sich an

diesen Linearreglern schonmal die Finger verbrennt. Ein Kühlkörper (passend für TO-220 Gehäuse) schafft hier etwas Abhilfe.

5.1 Hinweise zu Auswahl und Anschluß der Akkus

Es gibt prinzipiell drei Arten von Akkus: NiCd (Nickel-Cadmium), NiMH (Nickel-Metallhydrid) und Blei-, bzw. Bleigel-Akkus. Blei- bzw. Bleigel-Akkus sind nur für Spannungen von 6 bzw. 12 Volt erhältlich, und relativ schwer, sind aber sehr hochstromfähig. NiCd-Akkus haben einen ausgeprägten Memory-Effekt, d.h., sie müssen immer vollständig entladen werden, bevor man sie ohne die Lebensdauer zu reduzieren wieder aufladen kann.

NiMH-Akkus haben nahezu keinen Memory-Effekt und zusätzlich eine höhere Energiedichte. Sie werden meist als AA-Zellen mit jeweils 1,2V angeboten und sind aus vielen elektronischen Geräten, wie zum Beispiel Digitalkameras bekannt. Grundsätzlich empfiehlt sich also die Verwendung von NiMH-Akkus.

Entscheidend für die Laufzeit des Roboters ist die Kapazität der Akkus. 1600 mAh stellen hier das Minimum dar.

Nach Möglichkeit sollte man auf die einfachen "Batteriehalterungen" (siehe zum Beispiel Conrad-Katalog Seite 761) verzichten. An den Kontaktstellen fällt bei großer Belastung (hohe Ströme) zu viel Spannung ab. Besser sind Akkus mit Löt-fähigen, die man dann miteinander verlötet, oder fertige Modellbau-Akkupacks mit bereits angelöteten Tomya-Steckern (haben die meisten). Diese können auch direkt an das Handyboard angeschlossen werden.

Kapitel 6

Sensoren

In diesem Kapitel wollen wir verschiedene Sensortypen allgemein vorstellen und Einsatzgebiete sowie Vor- und Nachteile beleuchten. Wir werden außerdem Beispielsensoren zu jedem Typ sowie Bezugsquellen und Preise angeben.

Die dargestellten Sensoren sind dabei lediglich als unverbindliche Vorschläge zu betrachten. Sie sollen weniger erfahrenen Teilnehmern die Möglichkeit geben, eine sinnvolle Auswahl für den Kauf zu treffen. Wir ermutigen die Teilnehmer ausdrücklich, sich auch nach Alternativen umzusehen.

6.1 Reflexkoppler

Reflexkoppler ermöglichen die Unterscheidung von verschiedenfarbigen Oberflächen. So ist zum Beispiel das Erkennen einer weißen Fläche auf schwarzem Untergrund problemlos möglich. Auch die Unterscheidung verschiedener anderer Farbkombinationen (rot auf weiß, grün auf rot, usw) kann mit geeigneten Reflexkopplern realisiert werden.

Reflexkoppler erkennen, wie der Name vermuten lässt, Unterschiede im Reflexionsverhalten verschiedenfarbiger Oberflächen. Wird eine farbige Fläche mit Licht bestrahlt, so wird nur der Teil des einfallenden Lichtes reflektiert, dessen Wellenlänge der Farbe der Oberfläche entspricht. Eine grüne Oberfläche reflektiert also nur 'grünes' Licht mit Wellenlängen von etwa 550 nm. Fällt Licht anderer Wellenlängen (zum Beispiel rotes Licht mit 700 nm) auf eine grüne Fläche, so wird nichts reflektiert. Das rote Licht wird absorbiert.

Normales Umgebungslicht, sei es Sonnenlicht oder Licht von Lampen, ist aus Licht vieler verschiedener Wellenlängen zusammen gesetzt. Durch die Überlagerung der vielen einzelnen Spektralfarben erscheint es uns als weiß. Entsprechend reflektieren weiße Oberflächen alles einfallende Licht, egal welcher Wellenlänge. Oberflächen, die uns dunkel grau oder schwarz erscheinen, reflektieren nur wenig oder gar kein Licht, sondern absorbieren nahezu die gesamte einfallende Strahlung.

6.1.1 Funktionsweise

Betrachten wir zunächst den prinzipiellen Aufbau eines Reflexkopplers.

Wie in Abbildung 6.1 abgebildet, besteht ein Reflexkoppler aus einer Fotodiode und einem Fototransistor. Die durch die Diode ausgesandte Lichtstrahlung wird vom zu untersuchenden Objekt (hier als rote Fläche) teilweise reflektiert und gelangt anschließend in den Fototransistor. Je nach Stärke des reflektierten Lichts steuert dieser Fototransistor durch, so dass ein direkter (wenn auch nichtlinearer) Zusammenhang zwischen Transistorstrom und reflektierter Lichtmenge besteht. Statt des Foto-Transistors kann auch ein Spannungsteiler mit einem Fotowiderstand benutzt werden.

Die Beschaffenheit des Untergrundes beeinflusst die Wahl der Fotodiode und auch des einzusetzenden Transistors erheblich.

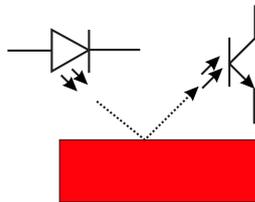


Abbildung 6.1: Funktionsweise eines Reflexkopplers

6.1.2 Reflexkoppler - CNY-70

Besteht die Aufgabe im Unterscheiden von einzelnen Graustufen, empfehlen wir die bereits fertig aufgebauten CNY-70 Reflexkoppler zu verwenden. Allerdings sollte man mit den im Internet stehenden Datenblättern vorsichtig sein, einige zeigen den Anschlussplan genau verkehrt herum. Die CNY-70 haben vier Anschlüsse und arbeiten mit einer IR-Diode. Diese sehr kompakten Sensoren gibt es

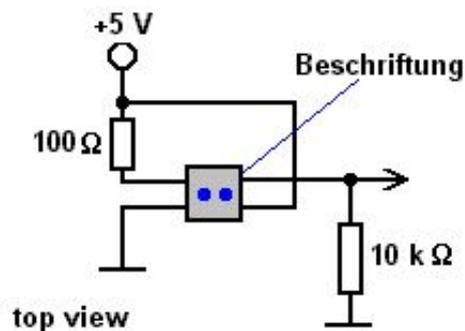


Abbildung 6.2: Anschluss des CNY70

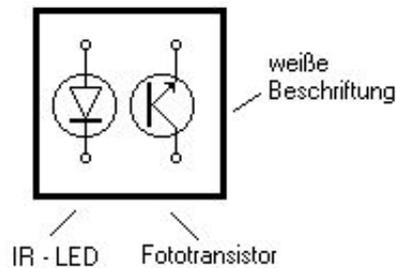


Abbildung 6.3: Aufbau des CNY70

unter anderem bei Reichelt für 0,58 Euro (CNY 70) zu bestellen.

6.1.3 Farbsensoren

Zum Erkennen von Farben hat www.micromaus.de verschiedene Sensoren im Angebot. So sind zum Beispiel die Sensoren vom Typ TSL257 für Rot, Grün und Blau erhältlich (je 4 Euro).

6.1.4 Farbsensoren im Eigenbau

Die in Abbildung 6.4 vorgestellte Schaltung ermöglicht das sichere Erkennen einer roten Fläche auf weißem Grund. Der oben vorgestellte CNY-70 kann diese Aufgabe nicht zufriedenstellend meistern, da die sich die Menge an reflektiertem IR-Licht bei hellgrauen bzw. weißen und roten Flächen nicht stark genug unterscheidet.

Zunächst fällt auf, dass der Reflexkoppler aus zwei LEDs besteht, eine rote und eine grüne (Komplementärfarben). Je nach Zustand des Input-Signals leuchtet eine der beiden LEDs, die jeweils andere ist im Sperrzustand und dunkel. Der rechte Zweig der Schaltung besteht aus einem einfachen Spannungsteiler, wobei einer der Widerstände ein Fotowiderstand ist. Zur Funktionsweise: Man schaltet abwechselnd die rote und die grüne LED ein. Dies geschieht einfach, indem man den Input-Pin abwechselnd auf Plus und Masse legt. Zur Erkennung der Farben vergleicht man den Spannungsabfall über dem Fotowiderstand. (Der Output-Pin wird sinvollerweise an einen Analog/Digital Eingang des Controllers angeschlossen. Mit einer nachgeschalteten Transistorstufe ist auch ein Schwellwertbetrieb möglich.)

Weißer Untergrund reflektiert rotes und grünes Licht gleich stark. Die über dem Fotowiderstand abfallende Spannung ist also in beiden Fällen gleich groß. Roter Untergrund hingegen reflektiert kaum oder gar kein grünes Licht, aber sehr viel rotes. Es sollte sich also ein großer Spannungsunterschied am Output-Pin messen lassen. Auf die gleiche Weise lassen sich also auch grüne Flächen erkennen. Die Schwarz/Weiß-Erkennung funktioniert mit diesem Reflexkoppler ebenfalls ohne Probleme.

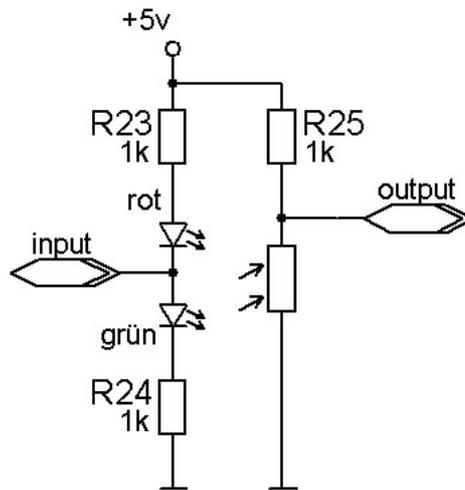


Abbildung 6.4: Farberkennung mit Reflexkoppler

6.1.5 Probleme

Wie alle optischen Sensoren sind auch Reflexkoppler höchst empfindlich gegen direkte Sonneneinstrahlung oder sonstige starke Licht- und IR-Quellen. Je nach Umgebungsbedingungen können sich Meßwerte auf ein und derselben Fläche relativ stark ändern.

Man muß ebenfalls beachten, dass nicht nur die für den Menschen auf den ersten Blick sichtbare Farbe für das Meßverhalten der Sensoren verantwortlich ist. So kann raues, matt weiß gestrichenes Holz für einen Reflexkoppler durchaus wie eine relativ dunkle Fläche erscheinen und ein glänzend schwarzer Gegenstand kann für einen IR-Reflexkoppler heller aussehen, als er in Wirklichkeit ist.

6.2 Entfernungsmessung

Das Messen von Entfernungen zu bestimmten Objekten ist ein entscheidender Punkt in nahezu allen Anwendungsbereichen mobiler Robotik.

In den meisten Fällen wird eine Abstandsmessung zur Vermeidung von Kollisionen mit Hindernissen eingesetzt. Darüberhinaus gibt es aber noch vielfältige weitere Anwendungsmöglichkeiten, wie Suchen und Finden von Objekten im Raum, Orientierung und Positionsbestimmung in der Umgebung, Kartierung der Umgebung und darauf aufbauendes planendes Verhalten.

Die einfachste Methode, ein vor dem Roboter liegendes Hindernis zu erkennen, ist der Einsatz eines Berührungssensors. Damit können aber nur wenige Informationen gewonnen werden. Spricht der Berührungssensor an, ist der Roboter bereits mit dem Hindernis in Kontakt gekommen. Ziel sollte es aber sein, Hindernisobjekte zu erkennen, bevor der Roboter gegen sie stößt und sich unter Umständen an ihnen verkeilt. Deshalb kommen Berührungssensoren in der Regel nur als 'Not-ausschalter' zum Einsatz. Ihre Reichweite kann mit entsprechenden mechanischen Konstruktionen wie Bügel etc. vergrößert werden, liegt aber im Allgemeinen im Bereich von wenigen Zentimetern um den Roboter.

Die zweite Klasse von Abstandsmessern besteht aus Sensoren, die Objekte innerhalb eines bestimmten Entfernungsbereiches erkennen können und ein Signal abgeben, wenn die Entfernung des Objekts einen bestimmten Schwellwert unterschreitet. Die Sensoren liefern also nur ein binäres Ausgangssignal: Entweder 'Objekt erkannt' oder 'kein Objekt'. Dahingehend sind sie den Berührungssensoren ähnlich, ihre Reichweite kann aber 50 cm und mehr betragen.

Die dritte Klasse schließlich liefert nun auch Informationen über die tatsächliche Entfernung eines Objektes. Sensoren dieser Kategorie kommen in den meisten Anwendungen zum Einsatz, denn sie sind flexibel einsetzbar und können mitunter die Aufgaben der Sensoren der ersten beiden Kategorien übernehmen. Je nach Sensortyp gibt es aber in der Regel einen Totbereich in unmittelbarer Nähe des Sensors, in dem die Daten unbrauchbar sind. Sind Messungen in diesem Bereich notwendig, muß doch auf die erste oder zweite Kategorie zurückgegriffen werden. Die maximale Reichweite variiert beträchtlich und beträgt bei Infrarotsensoren etwa 80 cm, bei Ultraschallsensoren mehrere Meter und kann bei professionellen Lasersensoren mehrere 100 Meter betragen.

Nachfolgend wollen wir einige Sensoren auf Infrarot- und Ultraschall-Basis genauer betrachten.

6.2.1 Objekterkennung auf IR-Basis - Sharp IS471

Der IS471 liefert ein binäres Ausgangssignal - je nachdem, ob ein Objekt erkannt wurde, oder nicht. Über die Entfernung des Hindernisses wird keine Angabe gemacht. Dieser Sensor gehört also in die zweite der obigen Kategorien.

Der IS471 sendet über eine anzuschließende IR-LED moduliertes Infrarotlicht aus. Befindet sich vor dem Sensor ein Objekt, wird das Licht in den IS471 zurück reflektiert. Durch das modulierte Licht ist der IS471 recht unempfindlich gegen Fremdeinstrahlung. Die Reichweite lässt sich über die Abstrahlleistung der IR-LED verändern und beträgt ohne zusätzliche Transistorstufe etwa 10 cm. Der unter anderem von Sharp produzierte Sensor ist bei www.micromaus.de für 2,18 Euro erhältlich.

6.2.2 Entfernungsmessung auf IR-Basis - GP2D1xx

Der GP2D12 von der Firma Sharp arbeitet mit IR-Strahlung und gibt einen zur gemessenen Entfernung proportionalen Spannungspegel aus, wie nebenstehendes Diagramm verdeutlicht. Deutlich sichtbar ist der Totbereich bei Entfernungen unter

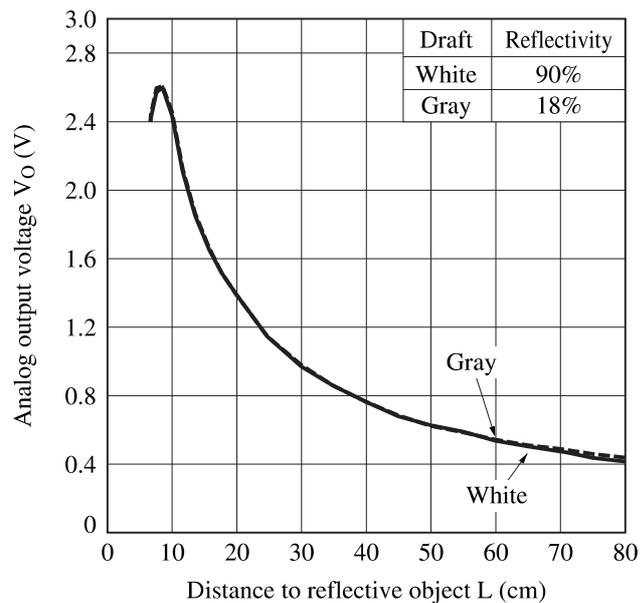


Abbildung 6.5: Kennlinie eines GP2D12

8 cm. Die maximale auswertbare Entfernung beträgt etwa 50 bis 60 cm. Der Sensor wird mit einem Kabel mit 3 Adern (Betriebsspannung, Masse und Signal) geliefert, der Anschluss ist also problemlos. Der Ausgangspin wird mit einem Analog-Eingang des Controllers verbunden.

Zur Funktionsweise: Der Sharp-Sensor besteht im Wesentlichen aus einer IR-LED

und einem sogenannten PSD-Chip (Position Sensitive Detector). Dazu kommt noch ein wenig Auswerte-Elektronik, die aber nicht weiter interessant ist. Abbildung 6.6 verdeutlicht die Arbeitsweise des Sensors. Das von der LED emittierte IR-Licht wird von Objekten in unterschiedlicher Entfernung zu unterschiedlichen Punkten auf dem PSD reflektiert. Je nach Auftreffpunkt gibt der PSD über die Elektronik ein anderes Spannungssignal aus.

Der GP2D12 arbeitet zuverlässig und lässt sich durch Materialeigenschaften (Far-

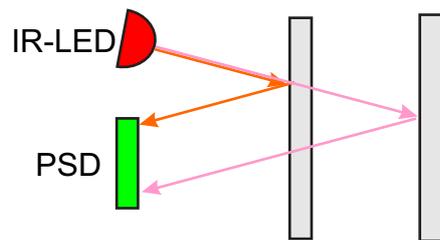


Abbildung 6.6: Funktionsweise eines PSD

be, Reflexionsverhalten etc.) nur wenig beeinflussen. Da der Sensor aber mit Infrarot-Strahlung arbeitet, wird er durch andere IR-Quellen wie Sonneneinstrahlung oder starke Scheinwerfer enorm gestört, die Messungen werden äußerst unzuverlässig. Die fremden IR-Strahlen müssen dabei nicht einmal direkt in den Sensor gestrahlt werden, es reicht schon eine diffuse Reflexion. Der Sensor reagiert auf diese Fremd-Strahlung mit einem sehr niedrigen Ausgangspegel, so als würde er kein Hindernis vor sich sehen.

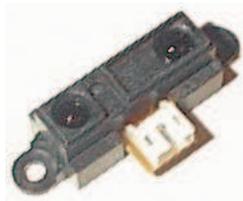


Abbildung 6.7: GP2D12

Es gibt noch weitere baugleiche Sensoren, die sich nur in ihrem Meßbereich unterscheiden:

Der GP2D120 entspricht dem GP2D12, nur liegt der Meßbereich bei etwa 4 bis 30 cm. Weiterhin gibt es den GP2D150, der einen Meßbereich von 20 bis 150 cm besitzt.

Bezugsquellen

Die Sensoren von Sharp sind bei www.micromaus.de für etwa 12 bis 13 Euro erhältlich.

6.2.3 Entfernungsmessung mit Ultraschall - SRF04

Ultraschall-Sensoren (auch Sonarsensoren genannt) basieren auf dem selben Prinzip, dass auch von Fledermäusen verwendet wird. Ein kurzer hochfrequenter Schallimpuls wird ausgestoßen und dessen Reflexion an Objekten in der Umgebung über einen Empfänger aufgenommen. Die Entfernung kann man einfach aus der Zeit zwischen Aussenden und Empfangen des Impulses errechnen, da die Schallgeschwindigkeit bekannt und konstant ist.

Es gilt: $s = v \cdot \frac{t}{2}$

v , die Schallgeschwindigkeit beträgt $343 \frac{m}{s}$ bei $20^\circ C$ Lufttemperatur. Sie steigt mit zunehmender Temperatur an. (bei $25^\circ C$ ist $v = 346 \frac{m}{s}$, bei $0^\circ C$ nur noch $331 \frac{m}{s}$) Die Zeit ist durch zwei zu dividieren, da die gesuchte Strecke ja zweimal vom Schall zurück gelegt wird (hin und zurück).

Sonarsensoren haben eine Reihe von Vor- und Nachteilen.

Vorteilhaft ist, dass sie natürlich unabhängig vom Umgebungslicht sind und so nicht wie die IR-Sensoren durch Sonneneinstrahlung etc. gestört werden können. Die gemessene Entfernung ist prinzipiell auch extrem genau, je nach verwendeter Auswerteelektronik liegen die Standardabweichungen im oder unter dem Millimeterbereich.

Nachteilig ist, dass sich der Schall nicht strahlförmig ausbreitet (wie IR-Licht) sondern kegelförmig. Je nach Ausbreitungswinkel des Sensors werden also auch Objekte erfasst, die relativ weit außerhalb der Mittelrichtung des Schallkegels liegen. Die präzise Position eines Hindernisses ist also nicht bekannt, es kann sich überall innerhalb des Sonarkegels auf einem Kreisbogen mit Radius s befinden.

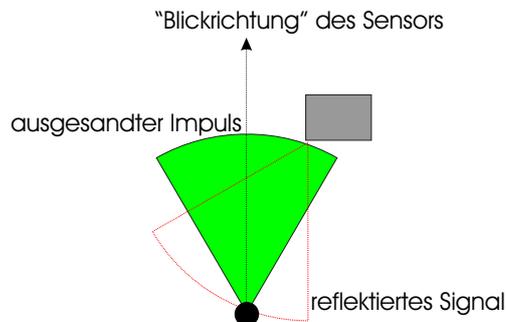


Abbildung 6.8: kegelförmige Reflexion bei Sonarsensoren

Ein weiteres Problem sind sogenannte Spiegelreflexionen. Dabei trifft der Ultraschallimpuls in einem zu flachen Winkel auf das Hindernis, das Signal wird gar nicht mehr zum Sensor zurück geworfen, oder erst über andere Hindernisse. In beiden Fällen ergibt sich ein ungültiger oder falscher Meßwert. Bei der Verwendung mehrerer Sonarsensoren ist auch das Problem des Übersprechens zu beachten. Dabei nimmt ein Sensor das reflektierte Signal eines anderen Sensors auf und erhält falsche Meßergebnisse.

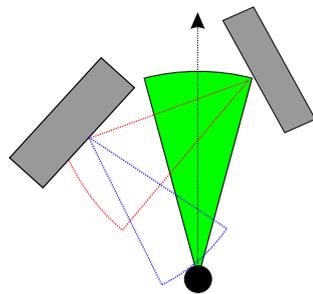


Abbildung 6.9: Spiegelreflexion

Nun wollen wir den SRF04 der Firma Devantech betrachten.



Abbildung 6.10: Frontansicht des SRF04

Der Sensor besteht aus Sender und Empfänger sowie einer Auswerteelektronik. Neben Betriebsspannung (5V) und Masse gibt es nur 2 weitere Anschlüsse. Am Trigger-Pulse Input ist ein mindestens $10 \mu\text{s}$ langer Impuls anzulegen, um eine Messung zu starten. Daraufhin wird ein Ultraschall-Impuls abgefeuert und der Echo-Pulse Ausgang auf Betriebsspannung gelegt. Sobald das erste reflektierte Signal wieder beim Sensor eintrifft, wird dieser Ausgang wieder auf Masse gezogen. Die Länge dieses High-Impulses ist also direkt proportional zur gemessenen Entfernung.

Der Anschluss des SRF04 ans Handyboard ist problemlos. Es steht der Befehl `sonar()` zur Verfügung, der die Messung und Auswertung durchführt und die ermittelte Entfernung in Millimeter zurückliefert.

Der Echo-Pulse Ausgang des Sensors wird mit dem Digital-Eingang 7 des Handyboards verbunden. Der Trigger-Pulse Eingang kommt an den Digitalausgang 0 auf dem Expansionboard. Zwischen den Aufrufen von `sonar()` sollte man mindestens 30 ms warten, damit die Sonarechos der letzten Messung verklungen sind. Dies kann einfach mit `sleep(0.03)` geschehen.

Bei den anderen Mikrocontrollern (AVR, PICs, ...) muß man entsprechende Funktionen zum Messen der Pulslänge am Echo-Pulse Ausgang nutzen bzw. selbst

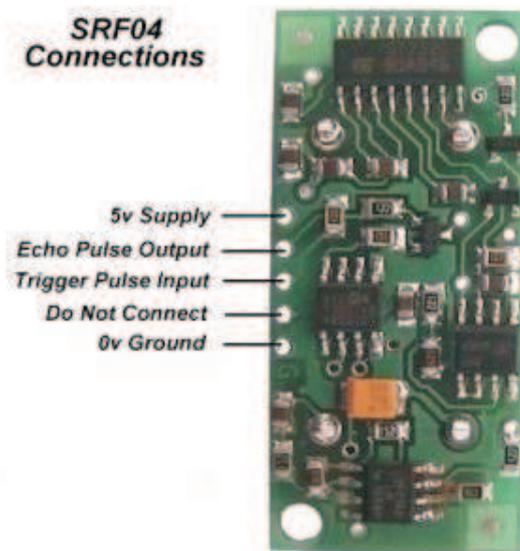


Abbildung 6.11: Anschlüsse des SRF04

schreiben.

Der SRF04 erkennt Objekte in 3 cm bis zu 3 Metern Entfernung. Die absolute Genauigkeit unter Verwendung des `sonar()`-Befehls ist bemerkenswert, die Standardabweichung vom statistischen Mittelwert beträgt nur wenige Millimeter. Der Effekt der Spiegelreflexion tritt bei Winkeln unter 60° ein.

In Deutschland wird der SRF04 auf www.roboter-teile.de für 25,52 Euro vertrieben. Dort gibt es auch ein Datenblatt. Der Sensor ist auch bei Acroname (www.acroname.com) erhältlich. Auf deren Homepage findet man weitere Informationen und Anwendungsbeispiele. Hergestellt wird der SRF04 bei Devantech.

6.2.4 Entfernungsmessung mit Ultraschall - SRF08

Der SRF08 ist gewissermaßen der große Bruder des SRF04. Seine Vorteile liegen in der größeren maximalen Reichweite (bis 6 Meter statt 3m beim SRF04) und der integrierten I^2C -Schnittstelle. Über den I^2C -Bus lässt sich der SRF08 sehr komfortabel konfigurieren und auslesen. Unter anderem können die maximale Meßentfernung und die Signalvorverstärkung eingestellt werden. Letztere beeinflusst die Empfindlichkeit des Sensors. Die Ergebnisse der Messungen gibt der SRF08 nach Anfrage über den I^2C -Bus zurück. Dabei wird die Entfernung bereits in Zentimetern angegeben. Bis zu 16 SRF08 lassen sich an einem I^2C -Bus betreiben, die Adressen können ebenfalls über den Bus eingestellt werden. Als kleines Extra hat der Hersteller an jedem SRF08 noch einen Fotowiderstand als Helligkeitssensor verbaut, der natürlich ebenfalls über den Bus abgefragt werden kann.

Alles in Allem ist der SRF08 dem SRF04 aufgrund der einfachen Handhabung klar vorzuziehen. Einziger Nachteil ist der hohe Preis. Erhältlich sind die Sensoren bei

www.roboter-teile.de für 44 Euro pro Stück. Deutlich kleiner aber funktionsgleich ist der SRF10, ebenfalls bei roboter-teile.de erhältlich.



Abbildung 6.12: SRF08

6.3 Radencoder

Viele Anwendungen erfordern die genaue Messung des vom Roboter zurückgelegten Weges. Wir wollen hier die verschiedenen Möglichkeiten der Wegmessung vorstellen und auch ihre Anwendungsgebiete erläutern.

Schon das einfache Geradeausfahren kann unter Umständen große Schwierigkeiten bereiten, wenn man keine Sensoren zur Bestimmung des zurückgelegten Weges einsetzt. Betrachten wir einen Roboter mit zwei unabhängigen Antriebsrädern (Differentialantrieb oder sog. "Panzerlenkung"). Obwohl beide Motoren mit der gleichen Spannung angetrieben werden, drehen sie sich nicht immer gleich schnell. Einmal unterliegen die Motoren bauartbedingt bestimmten Toleranzen, so daß zwei Motoren nie exakt gleich sein werden. Zum Anderen können ungleichmäßige Gewichtsverteilung, unterschiedlicher Untergrund (glatt / rauh) oder Unterschiede im Getriebe zu unterschiedlichen Drehgeschwindigkeiten führen. Der Roboter fährt also eine Kurve.

Dieses Fehlverhalten kann aber ausgeglichen werden. Man muß die tatsächliche Drehgeschwindigkeit bzw. den tatsächlich zurückgelegten Weg für jedes der beiden Antriebsräder bestimmen. Treten Differenzen auf, fährt der Roboter offensichtlich nicht geradeaus. Dann kann mittels Software nachgeregelt werden, indem man den einen Motor mit etwas höherer Spannung versorgt.

Soll sich der Roboter um einen bestimmten Winkel drehen, ist es ebenfalls notwendig, den gefahrenen Weg für jedes Rad exakt zu kennen. Anhand des Radabstandes kann man die zu fahrende Strecke für jedes Rad bestimmen, wenn sich der Roboter um einen bestimmten Winkel drehen soll. Kennt man zu jeder Zeit den gefahrenen Weg und den aktuellen Drehwinkel des Roboters, ist auch eine Positionsbestimmung und Navigation möglich. Aus gefahrenem Weg und Winkel läßt sich eindeutig eine (x,y)-Koordinate errechnen an der sich der Roboter im Moment befindet. Diese Navigationsmethode bezeichnet man als Koppelnavigation.

Sensoren zur Wegmessung werden als Radencoder bezeichnet.

Sie werden normalerweise auf der Antriebswelle eines Motors oder einer Achse befestigt. (Natürlich kann auch ein Getriebe zwischengeschaltet sein.) Man unterscheidet grob zwei Arten: absolute Meßwertgeber und inkrementelle Meßwertgeber.

Absolute Meßwertgeber liefern ein Signal zurück, das eindeutig die aktuelle Position der Welle bestimmt. Solche Sensoren werden oft zur Bestimmung der Position von Roboterarmen verwendet. Zur Bewegungsanalyse sind sie weniger geeignet. Inkrementelle Meßwertgeber liefern eine Folge von Impulsen. Jedesmal, wenn sich die Welle oder das Rad ein wenig dreht, ändert sich der Ausgangszustand von High auf Low oder umgekehrt. Die Anzahl der Impulse ("Inkmente") pro Zeiteinheit entspricht also der Drehgeschwindigkeit des Rades. Die nachfolgend betrachteten Sensoren zählen alle zu dieser Kategorie

6.3.1 mechanische Radencoder

Die inkrementellen Radencoder lassen sich nocheinmal in mechanische und optische Sensoren unterteilen.

Mechanische Radencoder besitzen eine rotierende Scheibe, in die Schlitze gestanzt sind. Diese Schlitze dienen der Herstellung eines elektrischen Kontaktes. Dreht sich die Scheibe, entsteht also eine Folge von Low-High-Low Impulsen. Meist besitzen mechanische Encoder zwei solcher Scheiben, deren Schlitze etwas versetzt voneinander angebracht sind. Dadurch ist eine Unterscheidung der Drehrichtung (vor- oder rückwärts) möglich. Bild 6.13 verdeutlicht die Funktionsweise.

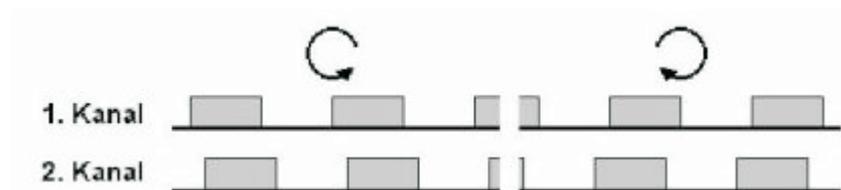


Abbildung 6.13: Impulsfolge beim mechanischen Radencoder

Ein solcher Sensor besitzt also 2 Signalausgänge.

Mechanische Sensoren sind relativ groß und teuer. Außerdem neigen sie zum Prellen, das heißt anstatt einer klaren Impulsflanke werden mehrere schnell aufeinanderfolgende Impulse erzeugt, bevor sich das Signal stabilisiert. Das Ausgangssignal muß also in der Regel noch geglättet bzw. entprellt werden. Dies kann in Hard- oder Software geschehen. Wird keine Entprellung vorgenommen, sind die gemessenen Wegstrecken zu ungenau.

Ein Beispiel für einen mechanischen Radencoder ist der Drehimpulsgeber von Conrad (705586-77), der aber mit 7,13 Euro pro Stück recht teuer ist.

6.3.2 optische Radencoder

Optische Encoder stellen eine sinnvolle Alternative zu den mechanischen Sensoren dar. Es gibt zwei Arten optischer Radencoder: Photounterbrecher und Photoreflektoren.

Photounterbrecher

Photounterbrecher arbeiten wie eine Lichtschranke. Ein Lichtsignal wird ausgesandt und empfangen. Zwischen Sender und Empfänger befindet sich eine rotierende Scheibe die den Lichtstrahl abwechselnd unterbricht und hindurchlässt. Dadurch entstehen wiederum Impulse, die gezählt werden können.

Die Scheibe kann eine Metall- oder Plastescheibe sein, in die Löcher oder Schlitze gefräst sind. Man kann auch eine durchsichtige Kunststoffscheibe verwenden,

bei der bestimmte Bereiche geschwärzt, also lichtundurchlässig sind. (siehe Bild 6.14) Sinnvollerweise verwendet man statt mit sichtbarem Licht mit Infrarotlicht, um weniger anfällig gegen Umgebungslicht zu sein. Trotzdem sollte man auf eine Abschirmung des Empfängers achten, so dass möglichst keine Fremdstrahlung empfangen wird. Man kann natürlich auch eine normale im sichtbaren Bereich arbeitende LED verwenden. Fotowiderstände sind aber als Empfänger meist ungeeignet, da sie zu träge reagieren. Hier sollte also ein Fototransistor oder eine Fotodiode verwendet werden. Zur Erkennung der Drehrichtung muß eine zweite, etwas versetzt angebrachte Lichtschranke verwendet werden.

Wer die Lichtschranken nicht selbst aufbauen möchte, für den bietet Conrad fertig aufgebaute Gabellichtschranken CNY37 für 1,51 Euro an (184250-77). Für 17 Euro erhält man eine Lichtschranke mit Taktscheibe (GP1A70R, Bestellnummer 185019-77). Dieser Sensor ermöglicht eine Erkennung der Drehrichtung, nach dem in Bild 6.13 verdeutlichtem Prinzip.

Günstiger gibt es den selben Sensor (GP1A70R) mit Taktscheibe bei www.micromaus.de für 14,89 Euro.

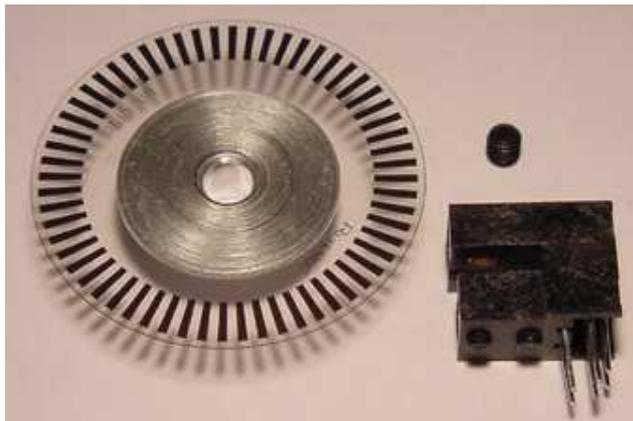


Abbildung 6.14: GP1A70R mit Taktscheibe

Photoreflektoren

Photoreflektoren arbeiten nach einem ähnlichem Prinzip, nur wird hier der Strahl nicht unterbrochen, sondern es wird das reflektierte Signal ausgewertet. Als Sensor verwendet man also einen Reflexkoppler, zum Beispiel den CNY70. (Zu Reflexkopplern gibt es auf unserer Homepage hier einen extra Artikel.) Dieser Reflexkoppler erkennt Hell/Dunkelunterschiede. Lässt man also eine Scheibe mit Schwarz-Weiß-Muster vor dem Reflexkoppler rotieren, erhält man die gewünschte Impulsfolge. Man kann das Hell-Dunkel Muster auch auf der Radinnenseite anbringen, wie in Abbildung 6.15 gezeigt. Der Reflexkoppler "schaut" von unten an die Radinnenseite. Man kann auch eine Lochscheibe verwenden, die unmittelbar vor dem Sensor rotiert.



Abbildung 6.15: optischer Radencoder (Photoreflektor)

6.3.3 induktive Drehzahlmessung

Das von krause-robotik.de angebotene Board misst nicht wie die anderen Sensoren die Anzahl der Radumdrehungen, sondern die tatsächliche Geschwindigkeit der Räder.

Es funktioniert bei allen mit einem PWM-Signal angesteuerten Gleichstrommoto-

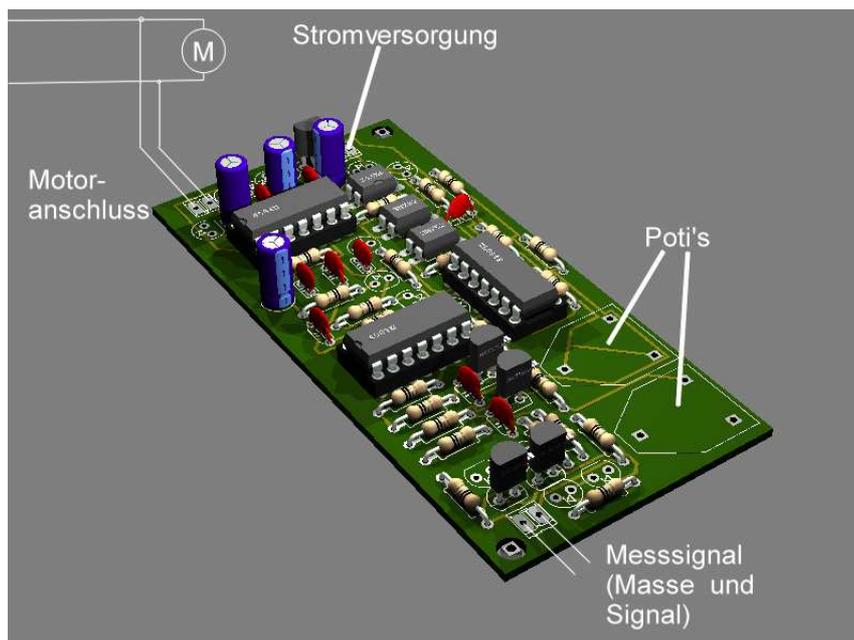


Abbildung 6.16: Schaltung zur induktiven Drehzahlmessung

ren. Das Board wird einfach zwischen Controller (bzw. den Ausgang des Motortreibers) und den Motor selbst geschaltet. Auf dem Board befindet sich ein analoger Ausgang, an dem je nach gemessener Geschwindigkeit eine Spannung zwischen 0 und 5V anliegt und einfach an einen analogen Eingang des verwendeten Mikrocontrollers angeschlossen werden kann. Die Nullstellung kann über ein Poti eingestellt

werden und liegt üblicherweise bei 2,5 Volt. Dreht der Motor vorwärts, wird eine höhere Spannung ausgegeben, dreht er rückwärts, eine kleinere Spannung. Die Empfindlichkeit kann über ein zweites Poti eingestellt werden. Damit wird festgelegt, bei welcher Drehzahl der Maximalwert erreicht werden soll.

Um die Funktionsweise zu verstehen, rufen wir uns noch einmal Abbildung 3.7 in Erinnerung. Während der Pausen zwischen den Impulsen wird die Spannung nicht auf Masse gezogen, wie man vielleicht meinen könnte. Vielmehr schalten die Motortreiber ihre Ausgänge während dieser Zeit in einen hochohmigen Zustand. Das heißt, die anliegende Spannung ist undefiniert, es kann aber kein Strom vom Treiber in den Motor hineinfließen. Der Motor dreht sich natürlich während dieser kurzen Unterbrechungen in der Stromversorgung aufgrund seiner Trägheit weiter. Er wirkt aufgrund seines Aufbaus jedoch als Generator, induziert also eine Spannung in den PWM-Kanal. Diese Spannung ist abhängig von der aktuellen Drehgeschwindigkeit. Das Board wartet also gewissermaßen auf die fallende Flanke von PWM-Signal und mißt kurz danach die vom Motor zurückinduzierte Spannung. Das Board ist bei www.krause-robotik.de für 62 Euro erhältlich.

6.4 Kompass

Ein weiterer sinnvoller Sensor ist das Kompassmodul CMPS03. Mit ihm kann der Roboter mit bestimmten Einschränkungen seine Ausrichtung in der Umwelt bestimmen. Das Kompassmodul misst die Ausrichtung mit einer Genauigkeit von etwa 4° . Allerdings ist dazu eine exakt waagerechte Montage des Sensors erforderlich. Er ist sehr empfindlich gegenüber Störungen wie den elektromagnetischen Feldern von Motoren, Akkus usw. Mit einer genauen Kalibrierung lässt sich der CMPS03 allerdings recht gut an seine Einsatzbedingungen anpassen. Diese muß allerdings bei jeder Inbetriebnahme neu ausgeführt werden.

Ansprechen lässt sich das Kompassmodul komfortabel per I^2C . Es stehen aber auch mehrere digitale Ports zur Kommunikation zur Verfügung.

In Deutschland ist der CMPS03 bei roboter-teile.de für einen Preis von 40 Euro erhältlich.



Abbildung 6.17: Kompassmodul CMPS03

Kapitel 7

Kommunikation

7.1 Der I^2C -Bus

Der I^2C -Bus ist ein bidirektionaler serieller Bus, der mehrere adressierbare Bausteine miteinander verbindet. Eingeführt und zum Standard entwickelt wurde der Bus von Phillips. Er zeichnet sich durch Datenraten von bis zu 100kBit/s (in der neuen Revision bis 400kBit/s), und einem 7 Bit breitem Adressfeld (10 Bit neu) aus.

Man unterscheidet zwei logische Betriebsmodi: Single-Master und Multi-Master-



Abbildung 7.1: das I^2C -Logo

Modus. Beim Single-Master gibt es, wie der Name schon sagt, nur einen Master am Bus. Alle anderen Teilnehmer an der Kommunikation sind Slaves, d.h. sie senden erst nach expliziter Aufforderung durch den Master und nie von alleine. Damit ist sichergestellt, dass nie zwei Bausteine gleichzeitig auf den Bus zugreifen wollen. Beim Multi-Master-Modus gibt es entsprechend mehrere Master. Hier muß mit einem speziellen Protokoll das Übersprechen der Teilnehmer verhindert oder erkannt werden. Meist kommt jedoch ohnehin der Single-Master-Modus zum Einsatz. Von den 7 Adressbits sind meist die oberen 3 oder 4 für eine bestimmte Bausteinfamilie fest eingestellt. Die unteren Bits sind entweder als Pins nach außen geführt oder können (wie bei den SRF08) über den Bus eingestellt werden. Somit kann man immer mehrere Bausteine der selben Familie an einem Bus betreiben und trotzdem hat jeder Baustein seine eigene Adresse. Die genauen Spezifikationen stehen als PDF im Downloadbereich unserer Homepage zur Verfügung.

7.1.1 Hardware und Elektrische Aspekte

Hardwareseitig besteht der Bus aus zwei Leitungen. Mitunter wird er daher auch als TWI-Bus (Two Wire Interface) bezeichnet. Eine der beiden Leitungen ist dabei die Taktleitung (SCK, Serial Clock), die andere logischerweise die Datenleitung (SDA, Serial Data).

Im Gegensatz zum RS-232 werden hier keine exotischen Spannungen benötigt, beide Leitungen führen im Normalfall Standard-TTL Pegel (0V bzw. 5V). (Es sind auch andere Spannungen zulässig, zum Beispiel 3,3V für den High-Pegel, genauere Informationen stehen in den Spezifikationen). Im Ruhezustand führen beide Leitungen High-Pegel (üblicherweise 5V). Dies muß durch je einen Pullup-Widerstand (zwischen 10k und 4k7) gegen 5V sichergestellt werden.

7.1.2 Ablauf der Übertragung

Im Ruhezustand führen beide Leitungen 5V. Anfang und Ende einer Datenübertragung werden von einem Start- bzw. Stop-Signal eingerahmt. Eine fallende Flanke auf SDA, während SCL noch High ist, codiert dabei das Start-Signal. Eine steigende Flanke auf SDA stellt den Stop-Zustand dar. Abbildung 7.2 verdeutlicht den Zusammenhang.

Wurde das Start-Signal gesendet, kann der Master (oder der Slave) mit der eigent-

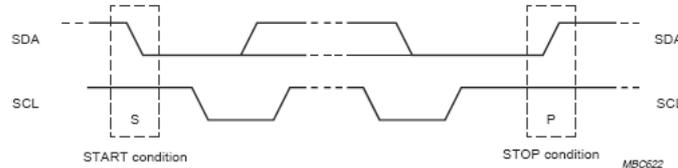


Abbildung 7.2: Start- und Stop-Zustände auf dem Bus

lichen Datenübertragung beginnen. Dabei wird die Taktleitung SCL vom Master gepulst. Auf der SDA-Leitung wird das zu sendende Bitmuster angelegt. Die Daten werden von den Empfängern dabei während der High-Phasen der SCL-Leitung übernommen. SDA darf sich also nur während der Low-Phasen des Taktes ändern. Siehe Abbildung 7.3.

Die ersten Daten, die auf dem Bus übertragen werden, sind die 8 Bit des Adressfel-

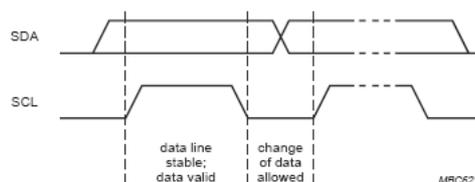


Abbildung 7.3: Datentransfer

des. Es besteht aus den 7 Adressbits, die den anzusprechenden Baustein adressieren, sowie einem R/W Bit. Dieses 8. Bit im Adressfeld gibt an, ob lesend oder schreibend auf den Empfänger zugegriffen werden soll.

Fühlt sich einer der am Bus lauschenden Bausteine durch die übertragene Adresse angesprochen, so wird er dem Sender eine Bestätigung (Acknowledge, ACK) schicken. Dies geschieht einfach während des 9. Taktes. Zieht der angesprochene Baustein die SDA-Leitung auf Low, entspricht das einem ACK an den Sender. Grundsätzlich wird jedes übertragene Byte durch den Empfänger bestätigt, d.h. nach jeweils 8 übertragenen Bits sendet der Empfänger ein ACK an den Sender zurück.

Abbildung 7.4 zeigt den Ablauf eines kompletten Datentransfers. Weitere Informa-

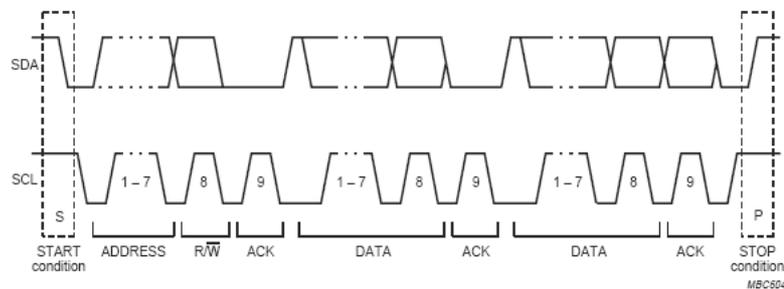


Abbildung 7.4: eine vollständige Übertragung

tionen sind in der I^2C -Spezifikation sowie in den Datenblättern der Bausteine mit I^2C -Interface zu finden.

7.1.3 Programmierung

Solange man nicht in Assembler programmiert, bieten alle im Kapitel 4 vorgestellten Programmierumgebungen fertige Bibliotheksfunktionen zum Senden und Empfangen an (zumindest für den Single-Master-Mode).

7.1.4 Anwendungen

Von den vielen Bausteine mit I^2C Interface wollen wir hier einige kurz vorstellen. Im Kapitel 6 wurden bereits der Ultraschallsensoren SRF08 und ein Kompassmodul vorgestellt, die über den I^2C -Bus ansprechbar sind. Der Servo-Controller SD20 wurde schon im Kapitel 3.2 angesprochen.

Portexpander Hat man nicht genügend digitale I/O Ports an seinem Controller zur Verfügung, kann man sich mit Portexpandern behelfen. Der PCF8574 stellt zum Beispiel 8 zusätzliche digitale Ports zur Verfügung, die über I^2C gesetzt oder gelesen werden können. Er ist bei Reichelt für 1,70 Euro im Angebot. Zwei weitere

Portexpander sind der PCA9555D bzw. PCA9554AD mit 16 bzw. 8 zusätzlichen Ports. Sie sind leider nur im SMD-Gehäuse bei embedit.de für etwa 3 Euro erhältlich.

D/A Wandler Den meisten Mikrocontrollern fehlt ein D/A Wandler. Will man einen echten analogen Ausgang, kann man einen externen DA-Wandler benutzen. Der TDA8444 ist ein 8-Kanal 6 Bit DA-Wandler mit I2C Interface. Bei Reichelt ist er für 1,15 Euro im Programm.

Speicher Die größte Bausteinfamilie mit I2C-Interface sind die zahlreichen EEPROMs, die bei Reichelt in allen möglichen Größen vorhanden sind.

7.2 RS-232

Jeder kennt wohl die RS-232 Schnittstelle von seinem PC. Jahrelang kommunizierten die PCs über diese serielle Schnittstelle mit Mäusen, Modems und anderen Geräten. Durch den Siegeszug von USB droht RS-232 völlig vom PC-Markt zu verschwinden, moderne Notebooks werden schon seit längerer Zeit ohne RS-232 Schnittstelle ausgeliefert. Trotzdem ist RS-232 nach wie vor sehr beliebt, wenn es darum geht einen Mikrocontroller mit dem Rechner kommunizieren zu lassen. Per Definition handelt es sich bei RS-232 nicht um einen Bus, sondern wirklich nur um eine Schnittstelle, da genau zwei Geräte miteinander verbunden werden.

7.2.1 Hardware und Elektrische Aspekte

Im Gegensatz zu I^2C definiert RS-232 ein asynchrones Übertragungsverfahren. Asynchron heißt, dass es keine zusätzliche Taktleitung gibt, sondern nur eine Datenleitung. Meist wird jedoch im bidirektionalen Betrieb gearbeitet, d.h. beide Geräte sollen sowohl senden als auch empfangen können. Dementsprechend sind dann zwei Datenleitungen nötig (und eine gemeinsame Masseleitung).

Beide Geräte (PC und Mikrocontroller) verfügen über einen RXD und TXD Pin bzw. Anschluß. RX steht dabei für Receive Data, TXD für Transmit Data. RXD vom PC wird mit dem TXD des Controllers verbunden und umgekehrt. Abbildung 7.5 zeigt die Pinbelegung eines D-SUB-9 Steckers. Pin 2 ist RXD, Pin 3 ist TXD. Die Masse wird über Pin 5 übertragen.

Die beiden Datenleitungen arbeiten nicht mit den gewohnten TTL-Pegeln. Zusätzlich wird noch eine negative Logik verwendet. Der Low-Zustand (logische 1) wird durch -12V codiert, der High-Zustand (logische 0) durch +12V. Wir werden gleich noch sehen, wie wir diese Spannungen erzeugen können.

7.2.2 Ablauf der Übertragung

Zu allererst müssen beide Teilnehmer der Kommunikation auf die selbe Übertragungsgeschwindigkeit eingestellt sein. Diese wird in Baud (in dem Fall Bit pro Se-

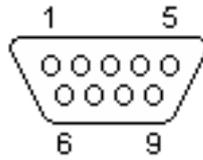


Abbildung 7.5: Pinbelegung bei D-SUB-9 Stecker

kunde) angegeben. Maximal sind meist 115kBaude möglich. Bei 115kBaude ist ein Bit 8, $68\mu s$ lang. Weiterhin können noch andere Vereinbarungen getroffen werden, wie zum Beispiel die Anzahl der Datenbits pro Paket (normal 8), die Anzahl der Stopbits usw.

Im Ruhezustand liegt die Übertragungsleitung auf 1 (entspricht Low-Pegel, also $-12V$). Der Sender sendet als erstes ein Startbit aus, um den Empfänger auf die beginnende Übertragung aufmerksam zu machen. Das Startbit ist einfach ein 0-Bit, die Leitung wird also auf $+12V$ gehoben. Der Empfänger erkennt die steigende Flanke und kann sich synchronisieren. Sind 115kBaude vereinbart, wird er das Startbit und die Hälfte des ersten Datenbits abwarten und danach alle 8, $68\mu s$ die Leitung abtasten. Je nach Pegel speichert er eine 1 oder 0 in seinem Empfangspuffer. Je nach Vereinbarung wird nach 8 gesendeten Datenbits ein Stopbit (logische 1) gesendet. Damit ist die Übertragung abgeschlossen.

7.2.3 RS-232 mit Mikrocontrollern

RS-232 arbeitet, wie wir gesehen haben, mit Spannungen von plus und minus $12V$. Diese Spannungen kann man natürlich nicht einfach an die I/O-Ports eines Mikrocontrollers anschließen, geschweige denn, aus den Ports abgreifen. Wir brauchen also eine Möglichkeit, diese Spannungen zu erzeugen. Genau dafür gibt es Pegelwandler wie dem MAX232.

Der MAX232 erzeugt aus den normal vorhandenen $+5V$ die beiden benötigten Spannungen. Wie man in Abbildung 7.6 erkennen kann, stellt er zwei komplette RS-232 Schnittstellen zur Verfügung. Wir verbinden TXD vom Controller mit einem der Inputs des MAX232, zum Beispiel mit Pin 11. RXD vom PC kommt an den entsprechenden Ausgang vom MAX232, also Pin 14. Umgekehrt wird noch TXD vom PC mit Pin 13 und RXD vom Controller mit Pin 12 verbunden.

Der MAX232 muß mit 5 Kondensatoren, wie in der Abbildung dargestellt, beschalten werden. Alle Kondensatoren haben eine Kapazität von $1\mu F$

Weiterhin erkennt man im Inneren des MAX232 einige Inverter. Der Baustein invertiert also die durchlaufenden Signale. Damit entspricht auf der Controller-Seite der High-Pegel wieder einer logischen 1. Schauen wir also noch einmal auf den Ablauf einer Übertragung.

Wir wollen mit unserem Mikrocontroller ein 1-Bit senden. Wir legen also unseren TXD-Pin auf High-Pegel ($5V$). Der MAX232 wandelt diese $5V$ in $-12V$, gibt also $-12V$ auf die RXD-Leitung des PCs. Wie wir oben beschrieben haben, ent-

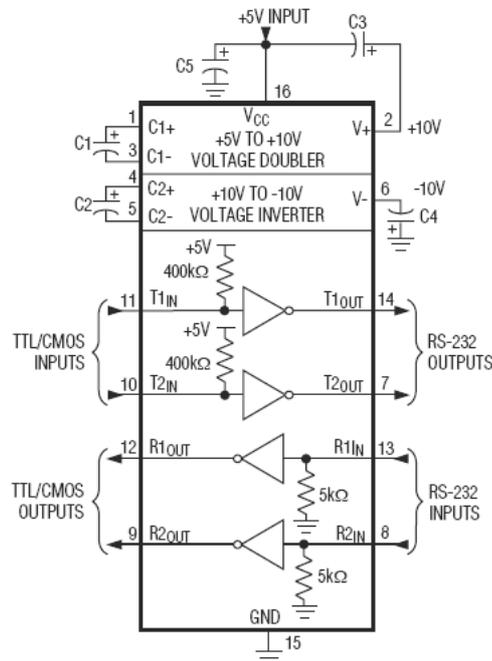


Abbildung 7.6: MAX232 Pinbelegung und externe Beschaltung

spricht -12V gemäß RS-232 Spezifikation der logischen 1, also wird der PC ein 1-Bit empfangen, genau wie wir wollten. Das gleiche klappt auch mit dem 0-Bit. Der Controller legt seinen TXD-Ausgang auf Masse, der MAX232 legt +12V an den RXD des PCs.

Noch ein Hinweis zur Beschaltung des MAX232. Es ist unbedingt darauf zu achten, die Elkos genauso wie im Datenblatt gezeigt anzuschließen, auch wenn sie scheinbar verpolt werden. Das ist richtig so und muß auch so sein, sonst funktioniert die Spannungswandlung nicht.

Neben dem MAX232 gibt es von Maxim noch zahlreiche andere Typen von Pegelwandlern. So sind sie als SMD Bausteine erhältlich, manche Typen brauchen nur kleine oder gar keine Kondensatoren usw. Für weitere Informationen ist das gemeinsame Datenblatt zu Rate zu ziehen. Fast alle Typen sind bei Reichelt erhältlich.

7.2.4 Anwendung

RS-232 eignet sich gut, um einen Controller mit dem PC kommunizieren zu lassen. So kann man Statusmeldungen oder Sensorwerte übertragen, auf dem PC auswerten und Befehle zurück schicken. Zum Experimentieren und Debuggen ist dies hilfreich, besonders wenn keine andere Anzeigemöglichkeit (LCD) vorhanden ist.

Kapitel 8

Weiterführende Informationen

Die Teilnehmer sind angehalten, sich weiterführende Informationen durch selbständige Recherchen zu beschaffen. Speziell im Internet ist eine Fülle von Informationen zu elektronischen Schaltungen und Robotik zu finden.

Besonders empfehlenswert sind diverse Foren zum Thema Roboter-Bau, wie zum Beispiel www.elektronik-projekt.de, www.roboterwelt.de/forum oder auch www.roboternetz.de. Dort wird einem sehr kompetent geholfen. Grundsätzlich sollte man aber wie überall im Usenet erst selbst recherchieren, bevor man dort eine Frage stellt. Natürlich sollte auch das RoboKing-Forum eine Anlaufstelle bei Problemen und zum Erfahrungsaustausch sein. Wer einmal mit anderen Teilnehmern chatten möchte, kann das im euIRC-Netz im Channel #roboking tun. (Server: irc.ffm.de.euirc.net Port: 6667).

Weiterhin steht euch das RoboKing-Team unter info@roboking.de mit Rat und Tat zur Seite.

Anhang A

Bezugsquellen und Preise

Angaben ohne Gewähr, Stand Juli 2005.

Bezeichnung	Beschreibung	Händler	Preis
Sharp GP2D120	IR-Entfernungsmesser 30cm	www.micromaus.de	12,50
Sharp GP2D12	IR-Entfernungsmesser 80cm	www.micromaus.de	12,00
Sharp GP2Y0A02YK	IR-Entfernungsmesser 150cm	www.micromaus.de	20,50
Sharp IS471F	IR-Sensor moduliert	www.micromaus.de	2,18
SRF04	Ultraschall-Entfernungssensor 3m	www.roboter-teile.de	25,52
SRF08	Ultraschall-Entfernungssensor 6m <i>I²C</i>	www.roboter-teile.de	44,02
Sharp GP1A70R	Radencoder vorwärts/rückwärts	www.micromaus.de	14,89
Drezahlboard	induktive Drehzahlmessung	www.krause-robotik.de	62,00
L293D	Motortreiber 600mA	www.reichelt.de	2,00
SN754410 NE	Motortreiber 1A pinkompatibel zu L293D	www.embedit.de	2,50
L298	Motortreiber 2A	www.reichelt.de	2,70
Handyboard	Handyboard RoboKing-Special	www.krause-robotik.de	250,00
ATmega32 Board	incl. Motortreiber, Servoausgänge, Display, RS232	www.krause-robotik.de	150,00
USC008	Controllerboard mit ATmega8, Motortreiber	www.embedit.de	29,90
ATmega128 Platine	Platine mit ATmega128 ohne Peripherie	www.embedit.de	32,90
ATmega64 Platine	Platine mit ATmega64, 2 RS232, ohne Peripherie	www.embedit.de	35,00

RoboKing Spezial-Angebote

- www.krause-robotik.de bietet ein Handyboard mit I^2C und Speicherhaltung für 250 Euro an.
- www.e-lab.de stellt eine kostenlose Version des AVRco Compilers für den ATmega64 zur Verfügung. Die CD ist auf Nachfrage bei uns erhältlich.
- www.embedit.de produziert ein zum AVRco passendes Board mit ATmega64 und zwei seriellen Schnittstellen. Preis 35 Euro.