

KONSTRUKTION EINES BIPEDALEN ROBOTERS

NUMMER 3

Besondere Lernleistung

| | | |
|-----------------|--------------------------------|---------------------------|
| Betreut von | Herrn Prof. Pretschner | HTWK Leipzig |
| | Herrn Scheuermann | Wilhelm-Ostwald-Gymnasium |
| Eingereicht von | Kreiß, Sven | Wilhelm-Ostwald-Gymnasium |
| Bearbeitet von | September 2002 bis Januar 2004 | |

BIBLIOGRAFISCHE BESCHREIBUNG

Kreiß, Sven

Konstruktion eines bipedalen Roboters – Nummer 3

31 Seiten 1 Anlage

Referat

Es gibt eine Vielzahl von Roboterkonstruktionen. Roboter z. B. mit Rädern, mit Flügeln und mit sehr vielen Beinen sind speziell an bestimmte Umweltbedingungen angepasst. Keines dieser Systeme ist in der Lage, dem Menschen im Alltag überall zu folgen.

Ein relativ unerforschtes Gebiet sind bipedale Roboter. Sie besitzen zur Fortbewegung zwei Beine. Sie haben prinzipiell die Möglichkeit, sich in der Umgebung des Menschen zu bewegen, da sie sich genauso wie ein Mensch bewegen. Der Bewegungsablauf des Gehens oder Laufens ist jedoch im Verhältnis zum Rollen mit Rädern sehr komplex. Die Bewegung besteht aus mehreren Phasen, welche von äußeren Gegebenheiten wie der Oberfläche des Bodens abhängen. Somit muss auch eine Möglichkeit geschaffen werden, die Oberfläche zu erkennen.

Abstract

There are a lot of different robot constructions. Robots e. g. with wheels, with wings and with many legs are adapted to certain environmental conditions. Not a single one of these systems is able to follow a human being in everyday life.

A relatively unexplored field are bipedal robots. They have two legs for their movement. So they have the mechanical ability to move like humans. Movements like going or walking are more complex than rolling on wheels. The movement consists of several parts and coordinative events. In addition it depends on outside conditions like the structure and consistency of the ground. Therefore a method has to be developed to scan and recognize the surface.

VORWORT

Anliegen und Motivation

Roboter bewegen sich heute auf sehr vielfältige Art und Weise durch unsere Welt. Sie können durch Kanalisationen klettern, durch Rohre rollen, tiefer als ein Mensch tauchen und in der Luft fliegen. Es wird versucht, durch neue Fortbewegungsmechanismen dem Menschen Zugang zu sonst unzugänglichen Stellen auf diesem Planeten zu verschaffen. Doch man muss schon jetzt sehr intensiv nach solchen Stellen suchen.

Es gibt auf unserer Erde kaum noch Landschaften, welche vom Menschen unberührt sind. Fast der gesamte Planet ist eine Kulturlandschaft, eine Landschaft, welche vom Menschen an die Eigenschaften des menschlichen Körpers angepasst wurde. Es liegt daher nahe, dass ein zweibeiniges Laufsystem technisch die beste Fortbewegungsmethode für diese Umgebung sein sollte.

Und selbst in der wilden Natur hat sich in 65 Mio. Jahren Evolution nicht das Rad und nicht ein Sechs- oder Vierbeiner sondern ein Zweibeiner durchgesetzt.

Mittels eines eigens entworfenen humanoiden Laufsystems sollen vor- und nachteilige Konstruktionen sowie wichtige und unwichtige mechanische Komponenten einer solchen Bewegung ermittelt werden.

Danksagung

Bedanken möchte ich mich bei meinen beiden Betreuern, Herrn Prof. Pretschner und Herrn Scheuermann, die mich im Rahmen ihrer Möglichkeiten unterstützt und immer wieder mit neuen zusätzlichen Ideen angeregt haben und somit die Erstellung dieser Arbeit ermöglichten. Des Weiteren möchte ich mich beim Förderverein „Wilhelm-Ostwald-Gymnasium“ Leipzig e.V. und bei der Olaf Kreiß Film- und Fernsehproduktion bedanken, die diese Arbeit finanziell unterstützten.

Persönliche Stellungnahme zur Besonderen Lernleistung

Für eine Besondere Lernleistung beschäftigt man sich ein Jahr lang mit einem ganz bestimmten Thema aus der Wissenschaft. Dabei lernt man ein Sachgebiet umfangreich kennen. Zur Konstruktion dieses komplexen Systems sind nicht nur die Fähigkeiten zur Fertigung und Steuerung, sondern auch Kenntnisse über das Bewegungsverhalten beim Gehen und Laufen eines Menschen sowie entsprechende Bewegungsstudien notwendig. Somit erfordert diese Arbeit Kenntnisse in verschiedenen Wissenschaftsbereichen sowie die Bereitschaft und das Vermögen diese anzuwenden und neu zu kombinieren.

Dieses Thema wird nicht nur umfassend, sondern auch sehr detailliert untersucht. Gerade bei der Ansteuerung treten unerwartete Probleme auf, welche unsere eigentliche Unwissenheit und den Bedarf der näheren Untersuchung erst deutlich machen.

INHALTSVERZEICHNIS

| | | |
|-----------|--|-----------|
| 1. | Grundlagen | 6 |
| 1.1. | Bewegungsarten..... | 6 |
| 1.1.1. | Statisches Gehen | 7 |
| 1.1.2. | Dynamisches Gehen | 7 |
| 1.1.3. | Laufen..... | 8 |
| 1.2. | Bipedale Roboter..... | 8 |
| 1.2.1. | Asimo (Advanced Step in Innovative MObility) | 10 |
| 1.2.2. | Pino | 10 |
| 2. | Analyse der Einzelsysteme | 11 |
| 2.1. | Kommunikation mit dem PC..... | 11 |
| 2.1.1. | Infrared Data Association Standard (IrDA Standard) | 12 |
| 2.1.2. | RS232..... | 12 |
| 2.1.3. | Fehlerkorrektur | 13 |
| 2.2. | Kommunikation mit den Servos..... | 14 |
| 2.2.1. | I ² C Protokoll..... | 14 |
| 2.2.2. | Bussystem | 16 |
| 2.2.3. | Pulsbreiten- und Pulsfolgemodulation..... | 16 |
| 2.3. | Steuerung | 16 |
| 2.3.1. | Mikrocontroller | 17 |
| 2.3.2. | Software | 19 |
| 2.4. | Servosysteme | 19 |
| 2.5. | Sensorik..... | 21 |
| 3. | Zielstellung..... | 22 |
| 3.1. | Mechanik | 22 |
| 3.2. | Steuerung | 22 |
| 3.3. | Sensorik..... | 22 |
| 3.4. | Visualisierung | 22 |
| 3.5. | Lerntechniken | 23 |
| 3.6. | Bewegungstechniken | 23 |
| 4. | Resultat – Nummer 3..... | 24 |
| 4.1. | Mechanik | 24 |
| 4.1.1. | Oberschenkel | 25 |
| 4.1.2. | Schienbein..... | 26 |
| 4.1.3. | Knöchel..... | 26 |
| 4.2. | Steuerung | 26 |
| 4.2.1. | AT90S8535..... | 27 |

| | | |
|--------|-----------------------------------|----|
| 4.2.2. | Programmierung..... | 27 |
| 4.2.3. | Testplatine | 28 |
| 4.2.4. | Platine..... | 29 |
| 4.2.5. | Funkstrecke | 30 |
| 4.2.6. | Algorithmen | 32 |
| 4.3. | Sensorik..... | 33 |
| 4.4. | Visualisierung | 33 |
| 4.5. | Lerntechniken | 34 |
| 4.6. | Bewegungstechniken | 35 |
| | Literaturverzeichnis | 36 |
| | Abkürzungen | 38 |
| | Sachwortverzeichnis | 39 |
| | Abbildungsverzeichnis..... | 40 |
| | Anlage | 41 |
| | Selbstständigkeitserklärung | 42 |

1. GRUNDLAGEN

Für den Einstieg in die Thematik werden hier bekannte Sachverhalte und Techniken beschrieben. Dabei sollen die technische Grundlage für die Realisierung des Gehens herausgefiltert sowie derzeit bestehende bipedale Roboter analysiert werden.

1.1. BEWEGUNGSARTEN

Auf dem Land bewegen sich die meisten Tiere, indem sie gehen. Es gibt jedoch im Tierreich verschiedene Techniken des Gehens.

Katalogisieren könnte man diese Arten nach der Anzahl der beteiligten Beine. So gibt es Tiere, welche mit mehr als vier Beinen gehen, andere bewegen sich mit vier Beinen und es gibt Tiere, die auf zwei Beinen gehen.

Beim Menschen setzte sich in 65 Mio. Jahren Evolution das Gehen auf zwei Beinen durch. Zum evolutionären Erfolg des *Homo sapiens* hat auch die Art der Fortbewegung einen Beitrag geleistet. Somit bleiben die Fragen: Warum ist das Gehen so effektiv? Welcher Faktor ist für die darwinistische Auslese bei der Technik der Fortbewegung bestimmend?

Die Technik desjenigen, welcher überlebt und sich fortpflanzt, ist die Technik mit Erfolg. Eine ausschlaggebende Rolle spielte sicherlich das Freiwerden der vorderen Gliedmaßen zum Greifen. Wichtig, um zu überleben, ist auch der Faktor Geschwindigkeit bei der Fortbewegung. Nur, schnell ist ein Mensch im Vergleich zu vielen anderen Tieren nicht. Auch in Bezug auf Sprunghöhe und -weite sind uns viele Tiere um Längen voraus. Bleibt die Betrachtung nach der verbrauchten Energie, um somit die Frage nach der Effizienz zu beantworten.

In der Forschung war das Thema der bipedalen Fortbewegung lange Zeit tabu, da das Gehen auf zwei Beinen als energetisch aufwendig galt. Betrachtet man die bisherigen Versuche, diese Bewegung technisch mit einem Roboter zu simulieren und die eingesetzten Hochleistungsmotoren, so ist die Skepsis durchaus zu verstehen.

Die Diplomarbeit des Münchners Amos Albert beschäftigt sich sehr ausführlich mit der Energie beim Gehen. Grundlage ist die Betrachtung eines Beines, welches sich bei ihm aus zwei gekoppelten Schwingern zusammensetzt. Mathematisch ist diese Betrachtung äußerst schwierig, da sich eine solche Konstruktion mit beliebigen Ausgangswerten chaotisch bewegt. Trotzdem versuchte er einen mathematischen Ansatz, um die Bewegung mit der niedrigsten Energie zu ermitteln. Er baute auch einen Roboter („*BART-UH*“), um sein Ergebnis praktisch zu testen. Sein Prototyp hat sechs Freiheitsgrade (drei pro Bein), die durch Inkrementalgeber mit 40 400 Inkrementen pro 360° überwacht werden. Jedes dieser Gelenke wird mit einem 90W-Gleichstrommotor gesteuert.¹ Von energetisch günstig kann daher keine Rede sein.

¹ Albert: Intelligente Bahnplanung und Regelung für zweibeinige Roboter, Seite 19

Die unterschiedlichen Techniken systematisiert man auch nach der Anzahl der stehenden Beine (*supporting legs*). Ein stehendes Bein ist das Standbein in der jeweiligen Phase der Bewegung. Eine Phase mit einem Standbein ist eine *single support phase* und eine Phase mit beiden Beinen auf dem Boden heißt *double support phase*. So ergeben sich die drei Techniken Statisches Gehen, Dynamisches Gehen und Laufen.

1.1.1. Statisches Gehen

Viele Roboter, die mit zwei Beinen gehen, stehen zu jedem Zeitpunkt stabil. Sie können in jeder beliebigen Position anhalten und sie können ihre Bewegungen beliebig langsam ausführen. Die Projektion des Körperschwerpunktes auf den Boden (*GCoM – Ground Projection of the Center of Mass*) befindet sich immer über oder zwischen den stehenden Füßen. Diese Zustände sind stabil.

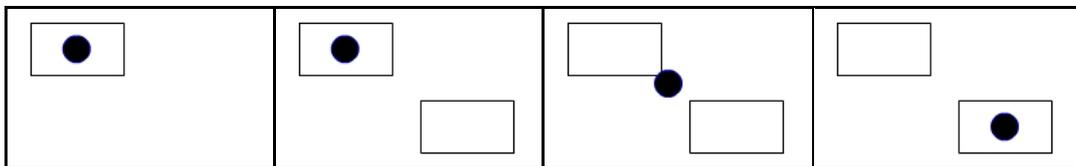


Abbildung 1: *GCoM und Füße beim statischen Gehen (Vogelperspektive)*

Aber gerade das Aufsetzen eines Fußes ohne den Schwerpunkt zu verschieben, erfordert extreme Kräfte. Erst wenn der zweite Fuß fest auf dem Boden steht, wird die Projektion des Körperschwerpunktes von dem ersten Fuß auf den zweiten verschoben. Für das menschliche Auge sieht diese Bewegung abartig aus. Man könnte es mit der Bewegung eines Menschen vergleichen, der über dünnes Eis läuft und sich nicht sicher ist, ob das Eis ihn tragen kann.

Das Gehen mit ausschließlich stabilen Gleichgewichtszuständen setzt sich aus *single* und *double support phases* zusammen. Hier hat man jederzeit vollständige Kontrolle über jede Bewegung.

1.1.2. Dynamisches Gehen

Der Mensch hingegen kann nicht in jeder Phase des Gehens anhalten. Es gibt Momente, in denen er labil steht. Der Körperschwerpunkt befindet sich dann nicht über oder zwischen den stehenden Füßen. Hier kippt der Körper von einer stabilen Position ins Leere und wird dann durch das Aufsetzen des anderen Fußes abgefangen. Während der Kippbewegung kann der Vorgang nicht unterbrochen werden.

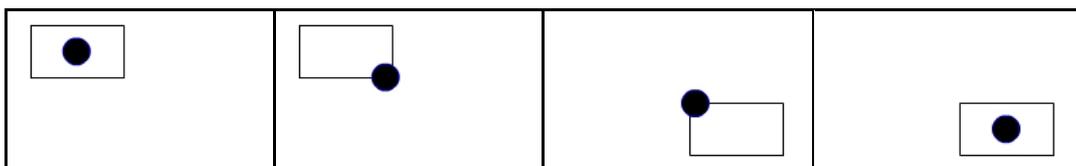


Abbildung 2: *GCoM und Füße beim dynamischen Gehen (Vogelperspektive)*

Die Geschwindigkeit des Kippens hängt von der trägen Masse und den Proportionen des Körpers ab. Die Kippbewegung wird so gewählt, dass beim Entlasten des einen stehenden

Beines sofort das andere belastet wird. Hierbei steht immer genau ein Bein. Sehr deutlich wird diese Phase bei dickeren Menschen sowie bei Leuten, welche gerade einen schweren Rucksack tragen.

Durch das dynamische Laufen wird die Kontrolle geringer. Die Schrittfrequenz ist jedoch größer. Honda - entwarf Asimo, den derzeit am weitesten entwickelten Humanoiden - analysierte die Dauer der Schritte beim statischen und dynamischen Gehen. Bei Asimo konnte die Zeit von über 5,0 auf 1,2 Sekunden gesenkt werden. Betrachtet man die dynamische Fortbewegung, dann ähnelt sie sehr stark dem menschlichen Laufen und sieht nicht mehr entartet aus. Die Pendelbewegung des Oberkörpers spielt bei dieser Bewegung zur Verlagerung des Schwerpunktes eine wesentliche Rolle und wäre ohne sie nicht möglich.

Man spart viel Energie durch den Einsatz dieser Technik. Pino (ein ebenfalls gut entwickelter Humanoid) benötigt Servos mit einem Drehmoment von über 210 Ncm. Er besitzt eine große Masse durch seine vielen Freiheitsgrade, die auch in Kopf und Armen eine enorme Flexibilität ermöglichen. Versucht man nicht das stabile Laufen als Vorstufe zum dynamischen Laufen in die Forschung einzubauen und geht direkt zum dynamischen Stil über, so ist es möglich, Servos mit einem Drehmoment von 30 statt 210 Ncm einzusetzen.

1.1.3. Laufen

Das Laufen ist eine Weiterführung des dynamischen Gehens. Beim Laufen wird die Kippbewegung so stark überhöht und eine zusätzliche Beschleunigung nach oben erzeugt, dass es zeitweise kein Standbein gibt. Durch die zwangsläufige höhere Geschwindigkeit und die Trägheit des Körpers verringert sich die Pendelbewegung der GCoM. Sie nähert sich mit steigender Schrittfrequenz einer Geraden, die in Bewegungsrichtung zwischen den beiden Füßen verläuft.

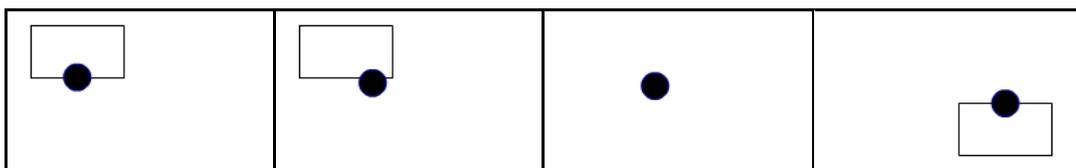


Abbildung 3: GCoM und Füße beim Laufen (Vogelperspektive)

Von diesen drei hier erläuterten Formen hat man demnach beim Laufen die geringste Kontrolle. Derzeit ist nur ein Roboter von Sony (im Dezember 2003 erstmals präsentiert) in der Lage, beide Beine für 0,04 Sekunden in der Luft zu halten.

1.2. BIPEDALE ROBOTER

Es gibt auf der Welt eine überschaubare Menge von Projekten mit humanoiden Robotern. Motivation zur Beschäftigung mit diesem Feld liefert die Roboterfußballweltmeisterschaft – kurz RoboCup. Das ist ein Wettbewerb mit einem großen Ziel:

Bis 2050 soll die Entwicklung so weit fortgeschritten sein, dass eine Mannschaft aus humanoiden Robotern gegen den amtierenden menschlichen Fußball-Weltmeister antreten kann – und natürlich gewinnt.

Quelle: Pressemitteilung der RoboCup German Open 2001

Der Wettbewerb ist derzeit in Ligen unterteilt. Das Spektrum reicht von *Simulation*, *Small Size*, *Middle Size* über *Legged* bis zu *Humanoid*. Die *Humanoid League* ist die einzige Klasse, in der ein Roboter auf zwei Beinen läuft. Diese Liga existiert erst seit dem Jahr 2002. Die Teilnehmer spielen noch nicht Fußball und flitzen auch nicht über das Feld, sondern messen sich in Einzeldisziplinen. Diese umfassen „auf einem Bein stehen“, „laufen“, fünf „Strafstöße“ und eine „freie Vorführung“.

Zur Weltmeisterschaft 2003 traten in Padua (Italien) acht Mannschaften an. Jedoch zeigte sich schnell, dass nur ein Roboter überhaupt die Chance hatte, die Disziplinen zu meistern. Bis auf den Nachfolger von Hondas Asimo war kein Humanoid wirklich in der Lage, eine Minute auf einem Bein zu stehen (siehe Anlage: News vom 03.06.03).

Diese Schwierigkeiten waren sicher ein Grund für den Rückgang der Teilnehmerzahl zum vergangenen Jahr. Zur Weltmeisterschaft 2002 in Fukuoka traten noch 13 Teams aus fünf Nationen an.



Abbildung 4: Alle Humanoiden des RoboCups 2002 in Fukuoka

1.2.1. Asimo (Advanced Step in Innovative MObility)

Asimo bzw. dessen Nachfolger *Hits Dream* ist der Humanoid mit dem mit Abstand am besten entwickelten Gesamtkonzept. Kein anderer Roboter kann auf so vielen Gebieten so große Erfolge vorweisen. Für einen humanoiden Roboter hat er eine ausgezeichnete KI (Künstliche Intelligenz), ist in der Lage auf über 140 japanische Sprachbefehle zu reagieren und ist unter anderem in der Lage, Gesichter und Körper zu erkennen und wieder zu erkennen (Video-Pattern-Recognition).



Abbildung 5: Asimo beim "Penalty Shot"

Es gibt Einzelsysteme, die nur einen Teil des menschlichen Körpers simulieren und Asimo weit voraus sind. *Kissmeth* ist ein Kopf mit Augen und einem Mund und der am besten entwickelten KI.

Am M.I.T. (Massachusetts Institute of Technology) wurde dieses System basierend auf einer KI entwickelt. *Kissmeth* ist in der Lage, sein Gesicht bestimmten Gefühlszuständen anzupassen. Er erkennt den Gemütszustand seines Gegenübers mit Hilfe einer Stereokamera und passt sein äußeres Erscheinungsbild der Situation an. Solche Systeme überragen auch das System von Asimo.

1.2.2. Pino

Pino ist ein Open-Source Projekt der „Japan Science and Technology Corporation“. Alle technischen Details sind im Internet verfügbar. Er hat 26 Freiheitsgrade und erreicht somit eine ähnliche Flexibilität wie Asimo. Diese Gelenke werden durch Servos mit drei unterschiedlichen Größen kontrolliert. Die Beine, die Hüfte und die Schultern werden von 14 Servos mit einem Drehmoment von 210 Ncm kontrolliert. Das sind die derzeit leistungsstärksten Servos dieser Bauart. Arm- und Nackenbewegungen werden durch acht Servos (80 Ncm) gesteuert und die Handbewegungen durch vier Servos (41 Ncm). Durch die Verwendung der fertigen Motor-Sensor-Einheiten, den Servos, ist so ein Roboter relativ einfach zu konstruieren und soll als Vorbild für dieses Projekt dienen.

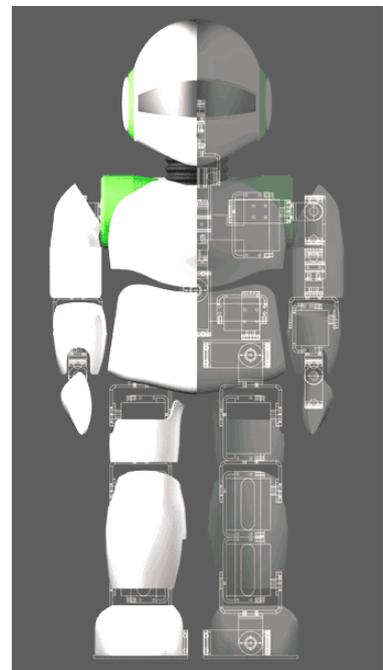


Abbildung 6: Pino

Die Stromversorgung und die Steuerung erfolgen über externe Kabel, welche an der Hüfte nach hinten heraushängen. Als mobile Steuerkonsole dient ein *Game pad*.

2. ANALYSE DER EINZELSYSTEME

Im folgenden Teil sollen die beteiligten Systeme zur Steuerung und zur Kontrolle des Humanoiden erläutert werden. Die Reihenfolge der folgenden Gebiete entspricht dem Signalweg des Positionssignals für einen Servo und wird am Ende durch den Komplex Sensorik ergänzt.

2.1. KOMMUNIKATION MIT DEM PC

Die Kommunikation des Controllers mit einer Steuereinheit ist besonders wichtig. Der Controller führt keine autonomen Routinen aus. Er kann nur die Befehle der seriellen Schnittstelle umsetzen. Dies erfordert eine Datenverbindung mit hoher Sicherheit und Bandbreite.

Die parallele Datenübertragung erfolgt über mehrere Datenleitungen gleichzeitig. Dadurch werden hohe Datenraten erzielt. Um die vielen Datenleitungen zur Verfügung zu stellen, ist aber auch eine komplexere Verdrahtung bzw. ein komplexeres Layout eines Chips nötig. Deswegen sieht man beim Anschluss von peripheren Komponenten von dieser Methode der Datenübermittlung ab. Beim internen Transfer in einem Chip oder auf einer Platine nimmt man die komplexere Architektur allerdings meist in Kauf, da Transferraten erreicht werden können, die mit einer seriellen Verbindung nicht möglich sind.

Eine der wichtigsten Möglichkeiten des Informationsaustausches mit anderen Systemen (z. B. mit dem Computer) bildet die serielle Kommunikation. Die Daten gelangen nacheinander Bit für Bit (daher seriell) vom Sender zum Empfänger. Meist wird eine bidirektionale Verbindung (full-duplex) aufgebaut. Dabei werden auf zwei Leitungen Daten gleichzeitig gesendet und empfangen. Alternativ gibt es die Betriebsmodi half-duplex und simplex. Half-duplex unterscheidet sich nur in der Eigenschaft, Daten nicht gleichzeitig in beide Richtungen verarbeiten zu können. Ein Sende- und Empfangsbaustein kann zu einer bestimmten Zeit nur senden oder empfangen. Bei einer simplex-Verbindung werden die Daten nur in eine Richtung gesendet. Es wird also nur eine monodirektionale Verbindung aufgebaut. Bei einer Kommunikation, bei der die Daten nacheinander übertragen werden, müssen Sender und Empfänger auch den gleichen Bittakt, die so genannte *Baudrate* (siehe Sachwortverzeichnis), besitzen.

Die serielle Verbindung benötigt nur wenige Datenleitungen. Das ist bei Verbindungen zu externen Geräten besonders wichtig. Die Sendeleitung nennt man TxD (Transmit Data), die Empfangsleitung RxD (Receive Data) und die Signalmasse SGND (Signal Ground). Je nach Protokoll kann es passieren, dass weitere Leitungen hinzukommen.

Bei der synchronen Datenübertragung, wie sie in einem SPI (Serial Peripheral Interface) verwendet wird, wird z.B. noch der Sendetakt übermittelt. Dabei können die einzelnen Bits exakter abgetastet werden und es wird eine höhere Datenrate erreicht.²

² Bermbach: Embedded Controller, 2001; Seite 127

2.1.1. Infrared Data Association Standard (IrDA Standard)

Das IrDA Protokoll wird für drahtlose Verbindungen mit einer Reichweite von bis zu zwei Metern unter Verwendung von infrarotem Licht als Datenträger verwendet. Es beinhaltet den Aufbau einer bidirektionalen Verbindung mit Geschwindigkeiten von 9600 b/s bis zu 4 Mb/s. Bis auf kleine Unterschiede im Timing der Bits gleicht diese Methode dem RS232 Standard, der im folgenden Abschnitt beschrieben wird.

2.1.2. RS232

Bei der asynchronen Datenübertragung nach RS-232 liegt im Ruhezustand an den Datenleitungen eine logische Null an. Den Beginn einer Übertragung kennzeichnet ein Startbit. Dies ist ein Low-Pegel für genau einen Bittakt (siehe *Abbildung 7: RS232 Eingangsdaten*). Darauf folgen die einzelnen Datenbits, beginnend mit dem LSB. Die Anzahl der Datenbits ist unterschiedlich, sie hängt vom System und der Anwendung ab. Normalerweise werden fünf bis neun Datenbits verwendet, am häufigsten sind acht Bit. Auf die Datenbits folgt ein Paritätsbit. Im Modus Even Parity (gerade Parität) ist dieses Bit Null (Parity=0), wenn die Anzahl der Einsen im Datenwort gerade ist. Bei Odd Parity ist es umgekehrt. Anders betrachtet kann man auch sagen, dass bei Odd Parity die Anzahl der Einsen mit dem Parity-Bit ungerade sein muss.³ Die Übertragung dieses Datenwortes wird durch ein oder mehrere Stoppbits beendet, welche High-Pegel besitzen. Anschließend ist die Leitung wieder inaktiv (High). Der High-Pegel beträgt im RS232C Standard -3V bis -15V und der Low-Pegel +3V bis +15V.⁴

Um in der Praxis bei einer solchen asynchronen Übertragung eine höhere Datensicherheit zu erreichen, wird jedes Bit in der Mitte mehrmals (in der SCU, Serial Control Unit, eines 80C186EC dreimal) abgetastet und dann eine Mehrheitsentscheidung getroffen. Deswegen arbeiten die meisten seriellen Bausteine mit einem vielfachen Takt der Baudrate (die SCU des 80C186EC arbeitet mit der achtfachen Baudrate).⁵

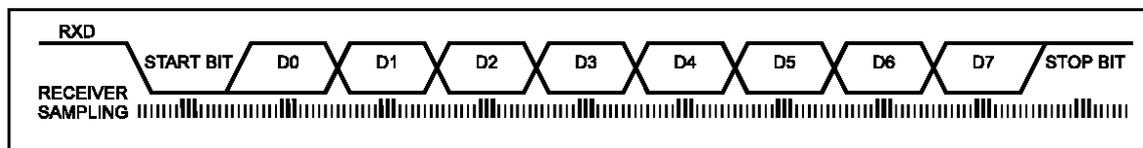


Abbildung 7: RS232 Eingangsdaten⁶

In diesem Projekt wird die Kommunikation mit dem Steuerrechner über dieses Protokoll gelöst, bei einer Verbindung über ein serielles Cross-Over-Kabel im Multiplex-Modus und bei einer Geschwindigkeit von 38400 Baud sowie acht Datenbits und einem Stoppbit.

³ Arizona State University: Serial Communication; Seite III-4
⁴ Arizona State University: Serial Communication; Seite III-6
⁵ Bermbach: Embedded Controller, 2001; Seite 127-128
⁶ Atmel Corporation: Datenblatt AT90S8535, 2001

2.1.3. Fehlerkorrektur

Ein weiteres Mittel der Fehlererkennung ist der CRC (*Cyclic Redundancy Check*). Dieser wird u. a. in RS232 und IrDA Verbindungen eingesetzt. Eine IrDA Verbindung wird immer mit CRC gesichert (mit CRC-16 bei Geschwindigkeiten bis 1 Mb/s und CRC-32 bei bis zu 4 Mb/s).⁷

Die Cyclic Redundancy Checksum wird beim Senden in Schicht 2 (Data Link Layer) des OSI-Schichtenmodells berechnet. Beim Empfangen wird die Checksumme in der Schicht 2 des Empfängers erneut gebildet und in Schicht 4 (Transport Layer) mit der übermittelten Checksumme verglichen. Sollten diese Werte nicht übereinstimmen, wird von dem Transport Layer eine Aufforderung zur erneuten Übertragung gesendet.

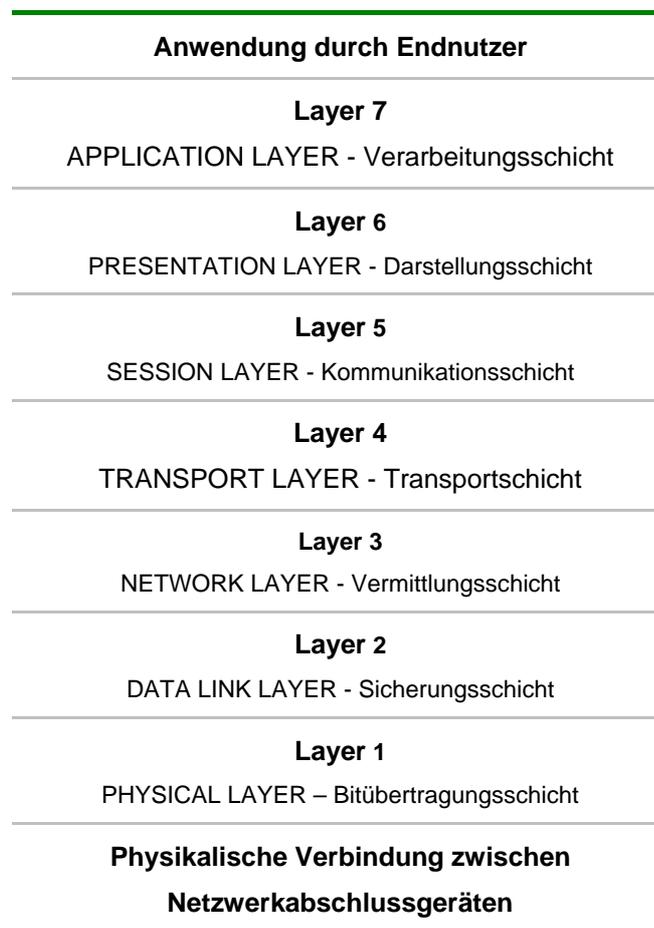


Abbildung 8: OSI - Deutsche Bezeichnungen nach DIN ISO 7498⁸

Ein Schichtenmodell versucht die verschiedenen Problembereiche der computervermittelten Kommunikation auf Schichten klar zu verteilen. Das einfachste Schichtenmodell bestünde demnach aus einer anwendungsorientierten Schicht und einer Übertragungs- oder Endsystem-zu-Endsystem orientierten Schicht.

⁷ The INFRARED DATA ASSOCIATION: Technical Summary of "IrDA DATA" and "IrDA CONTROL"

⁸ Payer: Computervermittelte Kommunikation, 2001

Durch die Verteilung bestimmter Probleme auf verschiedene Schichten können sich Entwickler spezialisieren und Fehler innerhalb einer Schicht gelöst werden, ohne das ganze System umzuschreiben.

OSI selbst definiert die Dienste und Funktionen, die auf den einzelnen Schichten erfüllt werden sollen. OSI definiert aber nicht die Standards, die diese Dienste und Funktionen verwirklichen. So können die Funktionen einer Schicht durch unterschiedliche Protokolle erfüllt werden.⁹

Im Signalweg der Servoposition folgt nun die Steuereinheit. Da die Funktionsweise der Steuereinheit von der Kommunikation mit den Servos abhängt, wird das Thema über die Kommunikation mit den Servos an dieser Stelle vorgezogen.

2.2. KOMMUNIKATION MIT DEN SERVOS

Derzeit sind zwei Protokolle zur Ansteuerung von Servos üblich. Im industriellen Bereich nutzt man zumeist das I²C (auch IIC) Protokoll. Das ist ein Zweileiterbus, welcher aus der Taktleitung SCL (Serial Clock) und der Datenleitung SDA (Serial Data) besteht. Es ist ein bidirektionaler Datenverkehr mit Geschwindigkeiten von bis zu 100 KBit/s mit bis zu 127 Teilnehmern möglich.

Im Amateurbereich nutzt man die einfachere Pulsbreitenmodulation. Hier wird die Position des Servos in der Pulsdauer einer Rechteckspannung kodiert. Weitere Informationen kann man nicht übertragen und man kann nur einen Servo über ein Kabel ansprechen.

2.2.1. I²C Protokoll

„Der I²C-Bus wurde 1979 von der Firma Philips definiert und sollte in erster Linie ICs in einem Gerät miteinander verbinden. Als Buslänge wurden maximal 4 m angesehen, was sich durch entsprechende Treiber jedoch um ein Vielfaches verlängern lässt. In einem Fernsehgerät, als typisches Beispiel für ein Gerät der Unterhaltungselektronik, empfängt der eingebaute Mikroprozessor über seinen Infrarot-Empfangsbaustein die Signale von der Fernbedienung und wertet sie durch sein Programm aus. Angenommen, er hat den Befehl bekommen, die Lautstärke zu verringern, so adressiert er über den I²C-Bus den angeschlossenen Lautstärkesteller und schickt ihm den Befehl, seinen Ausgangspegel um einige dB zu verringern. Der Lautstärkesteller quittiert den Empfang und verringert sofort die Lautstärke. Aber nicht nur der Lautstärkesteller ist über den I²C-Bus mit dem Mikroprozessor verbunden, auch die Helligkeit, der Kontrast und die Senderwahl werden über ihn gesteuert.“¹⁰

Jeder Teilnehmer bekommt eine 7-Bit-Adresse zugewiesen. Somit sind maximal 127 Bausteine ansprechbar. Alle Teilnehmer können als Sender (Transmitter) und Empfänger (Receiver) fungieren. Da kein Takt für den I²C-Bus festgelegt wurde, muss ein Teilnehmer einer Übertragung, der Master, den Takt generieren. Die Gegenstelle, der Slave, muss sich diesem Takt anpassen. Welcher Baustein Master/Slave ist, ist unabhängig von Sender und Empfänger.

⁹ Glossar.de: OSI, 1997-2003

¹⁰ Kamm, W.: Der I²C-Bus

Aus diesen variablen Eigenschaften ergeben sich vier Betriebsarten.

- | | |
|-----------------------|---|
| 1. Master Transmitter | <i>Master sendet Daten</i> |
| 2. Master Receiver | <i>Master empfängt Daten</i> |
| 3. Slave Transmitter | <i>Slave sendet Daten</i> |
| 4. Slave Receiver | <i>Slave empfängt Daten</i> ¹¹ |

„Es ist prinzipiell möglich, mehrere Master-fähige Bausteine in einem System zu betreiben. Jeder Master überwacht die Aktivitäten auf dem Bus und kann nur dann aktiv werden, wenn kein anderer Master den Bus belegt. Wenn 2 Master gleichzeitig auf den Bus zugreifen, wird durch einen Arbitrationsvorgang entschieden, welcher Master zuerst die Kontrolle über den Bus bekommt (Arbitration = Entscheidung).“¹² Nach dem Prinzip der bitweisen Arbitrierung wird die Entscheidung immer zugunsten des Teilnehmers mit der höheren Adresse gefällt (wie in *Abbildung 9: Prinzip der bitweisen Arbitrierung* beschrieben).

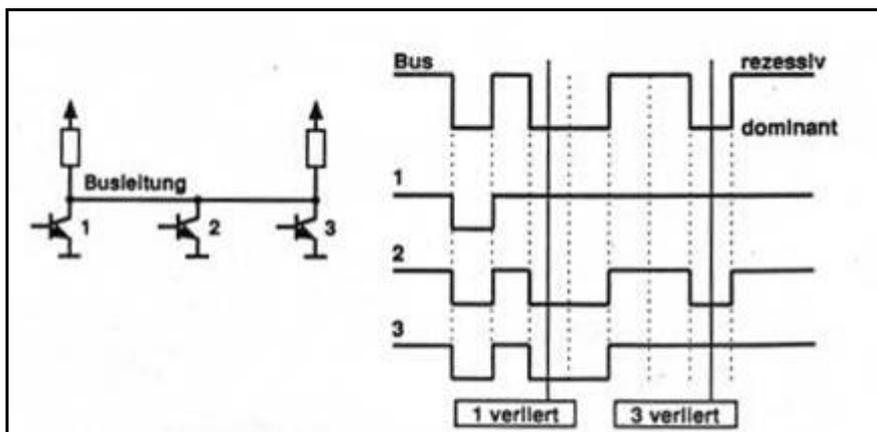


Abbildung 9: Prinzip der bitweisen Arbitrierung¹³

Die Signalpegel zur Kommunikation über diesen Bus sind nicht festgelegt. Somit ist es möglich, unterschiedliche Technologien (CMOS, Bipolar) zu vernetzen. Allerdings sollten alle Bausteine eines Systems die gleiche Versorgungsspannung haben, um Potentialunterschiede zu vermeiden.

Laut Protokoll liegt im Ruhezustand auf beiden Leitungen ein High-Pegel an. Die Leitungen sind über Pull-up-Widerstände mit der Versorgungsspannung verbunden. Alle Ausgangsstufen der ICs sind mit einer Open-Collector-Schaltung an den Bus angeschlossen. Sie macht es möglich,

¹¹ Kamm, W.: Der I²C-Bus

¹² Kamm, W.: Der I²C-Bus

¹³ Lichtenegger, CAN-Bus

Bausteine in einer Wired-AND-Verknüpfung an den Bus zu schalten.¹⁴ (Schematische Darstellung einer Wired-AND-Verknüpfung von drei Teilnehmern auf der linken Seite von *Abbildung 9: Prinzip der bitweisen Arbitrierung*)

2.2.2. Bussystem

Sobald an einer Verbindung mehr als zwei Teilnehmer partizipieren, nennt man diese Art der Datenübertragung Bussystem. Es gibt Bussysteme mit serieller und mit paralleler Datenübermittlung. Der Aufbau eines Protokolls für ein Bussystem ist ungleich komplizierter, da vor einer Übertragung genau festgestellt werden muss, ob der Bus belegt ist. Anzusprechende Teilnehmer müssen genau adressiert werden.

I²C ist ein Protokoll für ein serielles Bussystem.

2.2.3. Pulsbreiten- und Pulsfolgemodulation

Die Pulsbreitenmodulation (PWM) ist kein Protokoll und besitzt daher genau einen Empfänger und genau einen Sender. Diese Modulation ist eine Rechteckspannung, dessen Pulsdauer die Position des Servos widerspiegelt. Die Pulsbreitenmodulation wird vor allem bei Servos im Modellbaubereich eingesetzt. Heute hat sich hier der Positivimpuls durchgesetzt. Allerdings unterstützen Impulsgeneratoren häufig auch die Ausgabe von Negativimpulsen, die von älteren Servos genutzt werden.

Die Angaben in den Datenblättern über die maximalen bzw. minimalen Pulslängen schwanken stark. Die minimalen Pulslängen werden zwischen 0,7 und 0,9 ms und die maximalen zwischen 2,1 und 2,2 ms angegeben. Für die neutrale Mittelstellung des Servos werden überall 1,5 ms angegeben.

Heute nicht mehr üblich ist die Pulsfolgemodulation (PFM). Hier handelt es sich ebenfalls um eine Modulation einer Rechteckspannung. Träger der Positionsinformation ist die Frequenz. Da die Frequenz allerdings eine absolute Größe und nicht wie die Pulsbreite eine relative innerhalb des Signals ist, würde man eine Referenzfrequenz und somit einen größeren elektronischen Aufwand zur genauen Auswertung benötigen.

Mit Hilfe des PWM-Signals wird die Positionsinformationen vom Controller an den Servo übertragen.

2.3. STEUERUNG

Zur Erzeugung des Signals gibt es verschiedene Möglichkeiten. Ein PWM-Signal kann man mit wenigen Bauteilen und analoger Technik, aber auch mit digitalen Mikrocontrollern erzeugen. Zur Ansteuerung eines I²C basierten Servosystems ist in jedem Fall ein Controller nötig. PWM-Signale kann man analog mit einem astabilen Multivibrator erzeugen. Diese Schaltung ist sehr

¹⁴ Lichtenegger, CAN-Bus

einfach und stabil. So kann man in defekten Anlagen die eigentliche digitale Steuereinheit mit einer solchen Platine überbrücken, um Fehler zu finden.

Wie bereits erwähnt, bestehen die heutigen Steuereinheiten aus digitalen Mikrocontrollern. Bei einer ansonsten komplett digitalen Datenverarbeitung werden eine direkte Signalerzeugung und eine höhere Flexibilität ermöglicht.

2.3.1. Mikrocontroller

Mikrocontroller umgeben uns heutzutage überall. Am häufigsten werden sie in so genannten Embedded Applications (wörtliche Übersetzung: „eingebettete Anwendungen“) eingesetzt. Das sind Systeme, welche ihre logischen Operationen bzw. „Entscheidungen“ im Gerät selber treffen. Der Controller ist in der Anwendung „eingebettet“.

Daher findet man heute Controller in Konsumgütern wie Spül- und Waschmaschinen, in Geräten zur Kommunikation wie Telefonen, Faxgeräten und Modems, in Computer- und Bürogütern wie Keyboards, Mäusen und Monitoren, aber auch in Airbags, Getriebesteuerungen und Navigationssystemen von Kraftfahrzeugen. Dies ist natürlich nur ein winzig kleiner Ausschnitt aus den vielfältigen Anwendungen von Mikrocontrollern.¹⁵

Prinzipieller Aufbau

Ein Mikrocontroller benötigt bestimmte essentielle Einheiten. Diese gewährleisten die Funktionsfähigkeit des Bausteins. Das sind u. a. der Prozessor, der interne Bus und der Speicher. Die essentiellen Einheiten werden durch zusätzliche Einheiten ergänzt. Sie bestimmen die besonderen Leistungsmerkmale eines Controllers.

Der Prozessor ist für die sequentielle Instruktionausführung zuständig, wobei eine Instruktion einer Zustandsänderung im Speicher oder in einem Register entspricht. Die Ausführung der Befehle erfolgt synchron zu einem meist internen Taktgenerator, welcher durch einen externen Quarz-Oszillator gesteuert wird. Da sich ein Befehl aus mehreren Aktionen zusammensetzt, beträgt die Verarbeitungszeit mehr als einen Taktzyklus.

Der interne Bus verbindet zwei Einheiten miteinander. Ganz besonders wichtig ist hier die Verbindung zwischen CPU und Speicher. Bei den meisten Controllern arbeitet der interne Bus im Multiplexbetrieb. Das heißt, dass der Bus durch eine Steuerleitung kontrolliert wird und als Daten- und Adressbus fungiert.

Die Speichermöglichkeiten sind sehr vielfältig. Zum einen gibt es Speichertypen, welche sehr schnell, aber flüchtig sind und somit nur während der Laufzeit genutzt werden. Zum anderen gibt es Speicherarten, welche als nichtflüchtige Massenspeicher dienen.

Neben diesen Einheiten gibt es weitere Komplexe in einem Mikrocontroller, welche nötig sind, um den allgemeinen Betrieb aufrecht zu erhalten. Komplexe, welche die Funktion des

¹⁵ Bermbach: Embedded Controller, 2001; Seite 21

Controllers erweitern und ihn im Speziellen ausmachen, werden individuell hinzugefügt. Da das nicht Gegenstand der Arbeit ist, möchte ich diese Komplexe nur nennen und auf die Sekundärliteratur im Literaturverzeichnis verweisen. In dem Buch *Embedded Controller* von *Rainer Bermbach* sind diese Einheiten noch mal detailliert aufgeführt und erläutert:

- Bus Interface Unit
- Central processing Unit (CPU)
- Clock Generator
- Power Management Unit
- Interrupt Control Units
- Watchdog Timer Unit
- Portmultiplexer
- Timer; Timer Control
- Internal DMA Request Multiplexer; DMA Channel
- Serial Channel; SCU Control
- Chip Select Unit

Typen

Man kann Mikrocontroller auf zwei verschiedenen Wegen in jeweils zwei verschiedene Arten einteilen.

Zum einen kann man sie nach ihrem Gehäuse unterscheiden. Es gibt Controller, welche alle nötigen Komplexe in einem IC (*Integrated Circuit*) vereinen, so genannte *Single-Chip-Mikrocontroller*. Durch ihre einfache und flexible Einsetzbarkeit sind sie vor allem bei Hobbybastlern sehr beliebt.

Das Gegenstück zu dieser Gruppe sind Controller, welche durch externe Komponenten ergänzt werden müssen. Dabei wird der interne Bus an Portpins ausgegeben, um zum Beispiel einen externen RAM anzuschließen. Diese Controller werden meist in industriellen Massenproduktionen eingesetzt, da sie genauer auf die Anwendung zugeschnitten werden und somit billiger sind. Außerdem ist die Leistung der verwendeten Einzelkomponenten höher.

Mikrocontroller lassen sich auch nach ihrer Position im Gesamtsystem nach autonomen und semiautonomen Controllern unterscheiden. Autonome Mikrocontroller treffen auf Grund von Eingangswerten nach einem vorgegebenen Muster Entscheidungen und geben das Ergebnis über ihre Ports aus.

Semiautonome Controller werden von im Gesamtsystem höher liegenden Einheiten dirigiert und führen deren Entscheidungen nur aus.

UART im Mikrocontroller

Um semiautonome Controller ansprechen zu können, benötigen sie Schnittstellen. Damit ein Mikrocontroller von einem PC gesteuert werden kann, besitzt er ein UART.

Ein UART (Universal Asynchronous Receiver/Transmitter) ist ein IC, welcher die gesamte Software zum Kontrollieren einer seriellen Schnittstelle des PCs besitzt. Heute ist der UART auch direkt als Einheit in einen Mikrocontroller eingebunden. Dieser Baustein übersetzt Bytes in bitserielle Signale und umgekehrt. Bei der Generierung der seriellen Daten erzeugt er auch die für das Protokoll nötigen zusätzlichen Bits.

Um fehlerhafte Daten abzufangen, unterstützen UARTs den Parity-Check auch auf Hardwareebene. Nähere Informationen finden Sie unter 2.1.2 RS232. Gleichzeitig ist es auch möglich, Checksummen auf Softwareebene zu implementieren.¹⁶

2.3.2. Software

Eines der wichtigsten Elemente dieser Steuerung ist die Software, die die Bewegungsabläufe berechnet. Die Berechnungen können je nach Ausführung und Anspruch sehr aufwendig sein und werden von sehr unterschiedlich dimensionierten Computern ausgeführt. Bei Pino werden die Positionen von einem relativ einfachen Controllerboard berechnet. Bei Asimo hingegen wird jede Bewegung von einem System aus vier Prozessoren simuliert. Über Sensorinformationen aus den Füßen kann er die aktuelle Position der Projektion des Körperschwerpunktes auf dem Boden (GCoM) messen. Mit Hilfe dieser Information ist er in der Lage, jede Bewegung im Voraus zu simulieren. Jedoch ist selbst bei Asimo die Benutzeroberfläche eine primitive Anordnung von Buttons, ohne den Roboter als Animation auf dem Bildschirm zu sehen.

Die errechneten Gelenkstellungen werden an die ausführenden Stellglieder in der Mechanik des Humanoiden, an die Servosysteme, übermittelt.

2.4. SERVOSYSTEME

„Servoeinrichtung, in der Technik allgemein ein Gerät oder eine Kombination von Geräten zur automatischen Steuerung eines Mechanismus oder einer Kraft- bzw. Energiequelle. Das typische Merkmal einer Servoeinrichtung besteht darin, dass der geregelte Ausgang eines Mechanismus automatisch mit dem steuernden Eingang verglichen wird. Die Differenz zwischen den Einstellungen oder Positionen von Ausgang und Eingang bezeichnet man als Fehlersignal, das den Ausgang wieder auf seinen gewünschten Wert bringt. Es gibt mechanisch, elektrisch, hydraulisch oder optisch funktionierende Servoeinrichtungen. Den Vorgang der Rückübertragung des Fehlersignals zum Vergleich mit dem Eingang nennt man Rückführung. Den gesamten Prozess von Eingang, Ausgang, Fehlersignal und Rückführung bezeichnet man als Kreis mit Rückführung.“¹⁷

Die wohl bekanntesten Anwendungen von Servosystemen, Servoeinrichtungen oder nur kurz Servos in der Industrie sind die Servolenkung in Personen- und Lastkraftwagen oder der direkte Einsatz in Industrierobotern. Auch sehr bekannt ist der Einsatz in Bandmaschinen

¹⁶ U.S. Robotics: All about UARTs

¹⁷ Microsoft: Encarta 2003

(Videorecorder, Camera, ...) zur Geschwindigkeitsregulierung des Bandes sowie in Festplatten z. B. zur Positionierung des Schreib- und Lesekopfes.

Im Modell- bzw. Amateurbereich benutzen Hobbybastler *Single Case Servos*. Das sind Servoeinrichtungen, welche alle erforderlichen Einheiten eines Servos in einem Gehäuse vereinen. Eingesetzt werden diese Systeme unter anderem in Modellflugzeugen und in modernen Spielzeugen (z.B. Sonys Roboterhund *Aibo*).

Die Arbeit beschränkt sich auf *Single Case Servos*. Gründe dafür sind der günstige Preis und somit die Verfügbarkeit für Versuche in der Praxis. Sie sind für die Erfordernisse der Beispielanwendung ausreichend und unterscheiden sich in ihrer prozesstechnischen Funktionsweise nicht von industriellen Servosystemen.

Aktuelle Literatur ist über *Single Case Servos* kaum vorhanden. Selbst die Datenblätter zu den Servos geben nur Aufschluss über die genauen Maße der Gehäuse und nicht über elektronisch und mechanisch interessante Details.

Aufbau / Funktionsweise

Jeder Servo bekommt über ein Steuersignal ein Lagesoll übermittelt. Dieses Lagesoll gibt den Winkel einer Achse, die Position auf einer Geraden oder ähnliche Werte einer regelbaren Eigenschaft eines Systems an. Ein Sensor, auch Lagegeber, misst den Istwert. Aus dem Soll- und dem Istwert wird ein Fehlersignal errechnet, welches an den Lageregler übergeben wird. Der Lageregler errechnet daraus die Befehle für den Antrieb.

Bei einer Änderung des Lagesolls resultiert aus der Trägheit dieses Gesamtsystems aus elektronischen und mechanischen Teilen folgendes Bild: Der Istwert nähert sich dem Sollwert wie eine gedämpfte mechanische Schwingung der Nullstellung. Er schwingt einige Male um den Sollwert, bevor er ihn annimmt. In der Praxis versucht man daher die Elektronik so anzupassen, dass die Schwingung nicht spürbar ist. Im Amateur- und Modellbereich kann man solche Fehler allerdings mit bloßem Auge sehen.

Wenn ein Servo eine Position eingenommen hat, sollte er diese halten können. In der Realität stellt man allerdings schnell fest, dass dies nicht so ist. Ein sehr starker Servo gibt unter relativ geringem Druck nach. Theoretisch lässt sich diese Eigenschaft mit dem Prinzip von Ursache und Wirkung erklären. Es ist wie bei einer Wirbelstrombremse, die durch die Drehung einer Metallscheibe durch ein statisches magnetisches Feld die Metallscheibe bremst. Die Stärke der Verzögerung wird durch die Stärke der Drehung hervorgerufen. Die Ursache ist die Drehung und die Wirkung die Verzögerung. Dreht sich die Scheibe langsamer, wird sie auch nicht mehr so stark gebremst, da die Ursache der Verzögerung kleiner geworden ist. Somit kann man mit einer solchen Bremse theoretisch nicht anhalten.

Im Servo wird der Antrieb abgeschaltet, sobald der Istwert des Lagegebers mit dem Sollwert übereinstimmt. Lässt man nun eine äußere Kraft einwirken, verändert sich die Position und somit der Istwert. Auf diese Veränderung reagiert der Lageregler und wirkt mit Hilfe des

Antriebes der äußeren Kraft entgegen. Die Ursache ist der Unterschied des Ist- und Sollwertes und die Wirkung ist das Angleichen des Istwertes an den Sollwert. Somit hebt die Wirkung die Ursache auf und der Antrieb wird wieder deaktiviert. Da aber die äußere Kraft nicht aufgehoben wurde, hat die Deaktivierung des Antriebes wieder eine Änderung der Position und somit des Istwertes zur Folge. Der Kreislauf beginnt von vorn. In der Praxis ist das akustisch als ein leichtes „rattern“ wahrnehmbar.

2.5. SENSORIK

Wie die meisten Roboter verfügt auch Pino nicht über einen einzigen Sensor. Asimo bildet auch in dieser Kategorie wieder die Ausnahme. Er hat hochempfindliche und präzise Drucksensoren an seinen Füßen. An jeder Fußsohle befinden sich vier dieser Sensoren, aus denen sich die aktuelle Position der Projektion des Körperschwerpunktes auf den Boden (GCoM) errechnen lässt. Diese Sensoren ermöglichen, auf schiefen Ebenen zu stehen.

3. ZIELSTELLUNG

Nun sollen die Ziele für den neuen Humanoiden innerhalb dieser Arbeit definiert werden.

3.1. MECHANIK

Die Grundlage bildet eine stabile und kompakte Mechanik.

Aufgrund der wenigen Hilfsmittel zur Konstruktion ist ein einfaches Konzept erforderlich. Aus den Erfahrungen von Nummer 1 und 2 weiß man, dass sie sehr stabil sein muss und sehr klein, um die Auswirkungen der von den Servos verursachten Ungenauigkeiten zu minimieren.

3.2. STEUERUNG

Die Mechanik benötigt eine robuste und kleine Steuereinheit, die auf die Arbeit zugeschnitten sein muss.

Die Steuerung soll sehr klein und einfach sein, damit sie auch auf den Roboter verlagert werden kann. Des Weiteren muss sie stabile PWM-Signale erzeugen, damit sich der Roboter durch unkoordinierte Bewegungen nicht selbst beschädigt. Die Schaltungen müssen an die Erfordernisse dieser Arbeit zielgerichtet angepasst werden. Eine Steuerung der Superlative, wie sie Asimo besitzt, kann in diesem Rahmen und mit den finanziellen Mitteln nicht angestrebt werden.

3.3. SENSORIK

Die Steuerung sollte die Möglichkeit haben, die Bewegungen der Servos aufgrund von Sensorinformationen über die Oberfläche des Bodens anzupassen.

Sensoren an den Füßen, wie sie bei Nummer 1 und 2 bereits vorhanden waren, sollen das Laufverhalten weiter verbessern. Durch einfache Taster anstelle von hochempfindlichen Drucksensoren war es bereits bei Nummer 1 und 2 möglich, den Erdboden schnell zu registrieren und in der Abwärtsbewegung des Beines die Lage des Fußes an die Lage des Bodens anzupassen. Somit können Unebenheiten ohne Kamera und komplizierte Videoauswertung detektiert werden.

3.4. VISUALISIERUNG

Die Visualisierung muss die Bewegung des Roboters vor der realen Erprobung zeigen können. Man muss einen Eindruck über den Bewegungsablauf erhalten.

Besonders in der Entwicklungsphase ist es sehr wichtig, die Bewegung vor der Umsetzung zu sehen. Programmiert man eine falsche Bewegung, könnte man die Mechanik zerstören. Die Visualisierung soll direkt am steuernden PC erfolgen.

Auch hier müssen reale Ziele gesetzt werden. Eine Bewegungssimulation ist in diesem Rahmen nicht möglich. Dafür müssten Kräfte an Körpern berechnet und eine Kollisionserkennung geschrieben werden. So eine komplexe Berechnung ist erst in den neusten 3D-Animationsprogrammen möglich.

Daher wird eine relativ einfache Animation der Bewegungen angestrebt.

3.5. LERnteCHNIKEN

In Abhängigkeit zur Lauftechnik sind verschiedene Lerntechniken zu realisieren.

Für die Realisierung ist die Anwendung von zwei Lerntechniken erforderlich. Für das statische Gehen programmiert man die Bewegungsabläufe in zeitlicher Abfolge. Beim dynamischen Gehen wird in Stufen der gesamte Bewegungsablauf angepasst.

3.6. BEWEGUNGsteCHNIKEN

Ein realistisches Ziel in dieser Arbeit ist das dynamische Gehen.

Aufgrund der vorangegangenen Analysen der Bewegungstechniken und derzeit bestehender Laufsysteme sollte das dynamische Gehen am einfachsten zu realisieren sein. Die Umsetzung des statischen Gehens ist ebenfalls anstrebenswert. Sollte das statische Gehen nicht realisiert werden können, jedoch das dynamische Gehen, so ist ein weiteres Indiz für die höhere Effizienz des dynamischen Stils gefunden.

Die Umsetzung des Laufens wird nicht angestrebt. Die Bewegung ist sehr kompliziert und mit dieser Mechanik nicht umsetzbar.

4. RESULTAT – NUMMER 3

Im Rahmen dieser Arbeit ist es gelungen, das bipedale Laufsystem eines Menschen auf einen Roboter zu adaptieren. Die Details werden in diesem Abschnitt dargestellt.

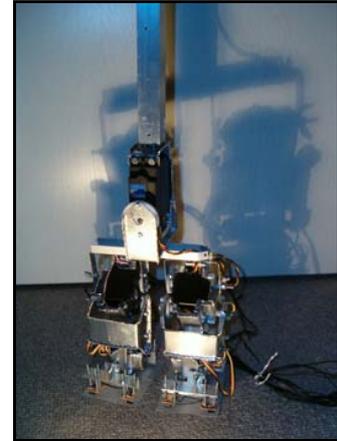


Abbildung 10: Foto - Nummer 3,
Totale

4.1. MECHANIK

Dieser Roboter besitzt neun Freiheitsgrade. An den Beinen steuern jeweils vier Servos die Position der Gelenke. Zusätzlich wird die Lage des Oberkörpers durch einen starken Servo kontrolliert.

Um zu beweisen, dass die Technik des dynamischen Laufens weniger Energie erfordert als das stabile Laufen wurden besonders leistungsschwache Servos verwendet. Diese sind nicht in der Lage, die Positionen beim stabilen Laufen zu fixieren.

Zur mechanischen Konstruktion wurden bei Nummer 1 noch Alu-Schienen, Hartpapierplatten, Drähte zur Kraftübertragung und Schrauben als Gelenke verwendet. Nummer 2 ist aus acht fast baugleichen Bausteinen zusammengesetzt. Sie enthalten einen Servo und eine Metallkonstruktion zur Verbindung der Baugruppen. Die Metallkonstruktion besteht aus sehr schwerem verzinktem Stahl.

Nachdem auch die Version aus verzinktem Stahl zu instabil war, wurde die Technik verändert. Zur Konstruktion von Nummer 3 werden zum ersten Mal in diesem Projekt CAD Technologien zur Erstellung der neuen Komponenten verwendet. Die Bauteile für Nummer 3 sollen stabiler und leichter zugleich sein.



Abbildung 11:
Standardbaustein
(Nummer 2), verzinkter
Stahl

Im Grunde ähneln die neuen Bausteine der Version von Nummer 2. Jedoch sind sie aus Aluminium. Dies hat in der Regel einen Verlust an Stabilität zur Folge. Durch das Abkanten an den Seitenrändern ist jedoch ein Verbiegen oder Knicken nicht mehr möglich. Zusätzlich werden die neuen Teile auch an der Unterseite des Servos eingeschraubt. Dies hat einen enormen Stabilitätsgewinn zur Folge. Da man die Servos im Inneren nicht verändern kann und somit die verzögerte Reaktion beim Halten einer Position nicht verhindert, wird die Länge sämtlicher Köperteile auf ein Minimum reduziert. Somit hat eine kleine Veränderung des Stellrades des Servos nicht so eine große Auswirkung. Es entsteht eine sehr kompakte Bauform der Beine.



Abbildung 12:
Aluminiumteil
(Nummer 3),
Abkantung und
Bohrung für
Verschraubung

4.1.1. Oberschenkel

Besonders kompakt sind die Oberschenkel. Hier sind die Servos für die Bewegung des Beines an der Hüfte und am Knie untergebracht. Durch das Verbinden der Servos an der langen Seite und die Schrägstellung der Servos im Ruhezustand ist dieser Teil nur 3,253 cm lang. Dies ist aufgrund der Stellräder an den Servos der nahezu kleinstmögliche Oberschenkel. Durch die vier Schrauben pro Servo an der Unterseite ist diese Konstruktion extrem stabil.

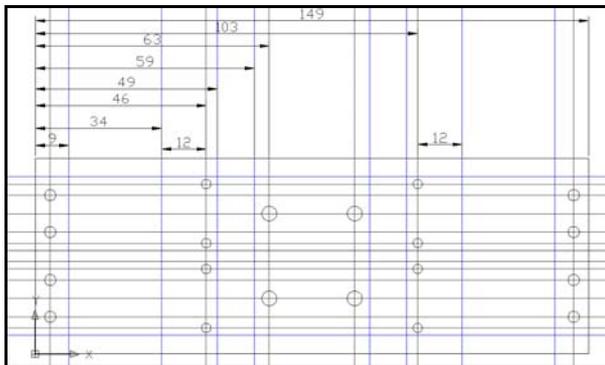


Abbildung 13: CAD-Modell eines Oberschenkels

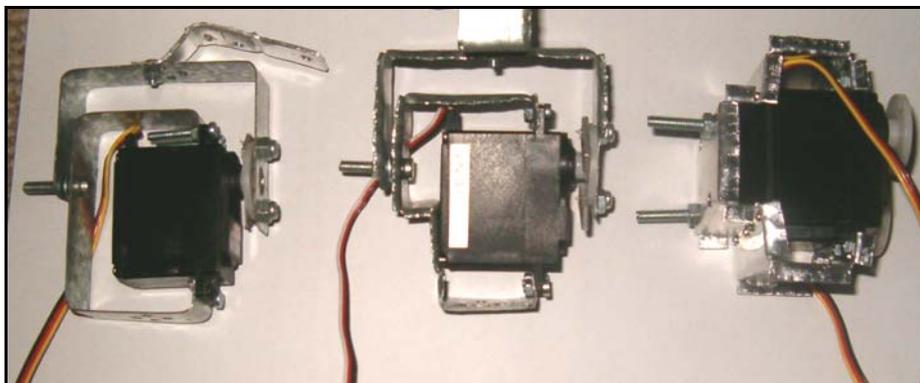


Abbildung 14: Entwicklung des Oberschenkels

Links: verzinkter Stahl; Mitte: Abkantung und Einschraubung; Rechts: Aluminium-Oberschenkel

4.1.2. Schienbein

Der Unterschenkel verbindet den oberen Servoblock, den Oberschenkel, mit dem unteren Servoblock, dem Knöchel. Jeder Servo benötigt eine bestimmte Größe des Haltebügels, um Platz für die Drehung zu schaffen. Im Schienbein wird bei Nummer 3 der Haltebügel des unteren und oberen Servos verknüpft und die Verbindung zwischen linker und rechter Seite nicht zwischen, sondern vor den Servos realisiert. Somit wird der freie Platz nicht von einem, sondern von zwei Servos verwendet und ist daher insgesamt kleiner.

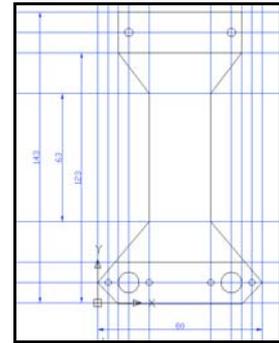


Abbildung 15: CAD-Modell eines Schienbeins

4.1.3. Knöchel

Die Konstruktion des Knöchels orientiert sich am stabilen und kompakten Oberschenkel. Die Schwierigkeit liegt hier in der Drehung des unteren Servos um 90°. Es gibt nur eine Anordnungsmöglichkeit, um die Servos an den langen Seiten zu verbinden und trotzdem den Bewegungsspielraum beider Servos zu erhalten. Zur Realisierung werden pro Knöchel zwei Aluminiumteile erstellt. Die erreichte Stabilität und die geringen Abmessungen sind gegenüber den Vorgängerversionen ein großer Fortschritt.

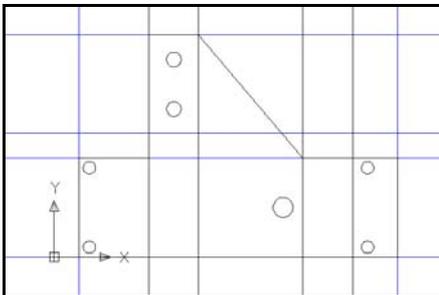


Abbildung 16: CAD-Modell eines Knöchelteils

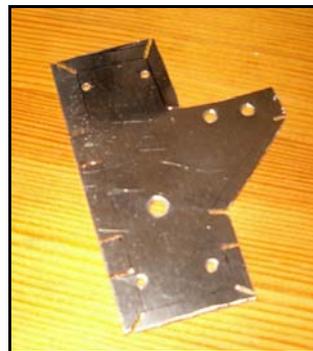


Abbildung 17: Foto des Alu-Stücks für den Knöchel

4.2. STEUERUNG

Zur Steuerung von Servos gibt es viele Möglichkeiten. Die wohl einfachste Variante ist die Erzeugung eines PWM-Signals an einer parallelen Schnittstelle eines PC. Unter DOS muss hierzu lediglich die Portadresse 378h verändert werden. Jedes Bit in diesem Register repräsentiert eine Datenleitung an der Schnittstelle. Effektiv lässt sich mit dieser Methode nur ein Servo steuern.

Um mehrere Servos anzusteuern, kann man astabile Multivibratoren einsetzen. Sie erzeugen eine Rechteckspannung, deren Pulsbreite von der Stellung eines Drehwiderstandes abhängt. Ersetzt man diesen Drehwiderstand durch einen digitalen Widerstand, so kann man diese Schaltung über die parallele Schnittstelle ansteuern. Da die digitalen Widerstände nur über 100 Schritte verfügen, verdoppelt man über einen zusätzlichen Optokoppler und einen Widerstand

die Gesamtzahl der Schritte auf 200. Somit benötigt ein Servo vier Steuer- bzw. Datenleitungen einer parallelen Schnittstelle, eine für den Reset des digitalen Widerstandes, zwei für die Einstellung des Widerstandes (der digitale Widerstand wird durch Up- und Down- Impulse gesteuert) und eine Leitung für den Optokoppler mit Zusatzwiderstand. Um diese Steuerung für Nummer 1 zu realisieren, werden drei Rechner über ein eigens entwickeltes Netzwerk mit seriellen Datenkabeln vernetzt. Vier parallele Schnittstellen steuern acht astabile Multivibratoren mit digitalen Widerständen und zusätzlichen Optokopplern.

Die Multivibratoren sind sehr empfindlich gegenüber Spannungsschwankungen. Die Versorgung erfolgt über die 5V-Leitung eines Computernetzteiles (mit 5A abgesichert) und galvanisch getrennt von der Stromversorgung der Servos. Ein weiteres Argument gegen beide bisher genannten Methoden (Erzeugung am parallelen Port; über astabile Multivibratoren) ist, dass man unter moderneren Betriebssystemen aus Sicherheitsgründen keinen direkten Portzugriff hat. Daher kann man die parallele Schnittstelle nicht mit der oben erläuterten Methode ansteuern.

Um die Größe und Komplexität der Steuerung zu reduzieren, wird für Nummer 3 eine Steuerung mit einem Mikrocontroller eingesetzt. Dadurch verringert sich die Zahl der Steuerrechner auf eins. Die Ansteuerung des Controllers ist nun über die serielle Schnittstelle möglich und die Software kann auch unter Windows oder Linux programmiert werden. Neben vielen anderen Vorteilen, die in den nächsten Abschnitten noch angeführt werden, wird die Größe der Steuerplatine wieder übersichtlich.

4.2.1. AT90S8535

Dieser Mikrocontroller hat viele Vorteile. Er unterstützt die Generierung von PWM-Signalen, bietet Schnittstellen für die Kontrolle über einen PC und ist billig.

Es ist möglich, 16 PWM-Signale simultan zu erzeugen. Man kann daher 16 Servos gleichzeitig kontrollieren. Die Positionswerte für die Servos werden über das integrierte RS232 Interface übermittelt. Des Weiteren dient die serielle Schnittstelle zur Übertragung von Debug Informationen sowie von Sensorinformationen, soweit vorhanden. Der interne Analog-Digital-Konverter (ADC) ermöglicht die einfache Einbindung von Sensoren.

Besonders vorteilhaft ist die *ICP-Fähigkeit (In-Circuit-Programming)* des Controllers. Sie ermöglicht das Programmieren des Schaltkreises, ohne ihn aus der Schaltung zu entfernen und in ein externes Programmiergerät zu stecken. Dadurch kann Zeit bei der Entwicklung gespart werden.

4.2.2. Programmierung

Zum Erstellen der Programme, die auf dem Controller laufen sollen, wird die kostenlose Entwicklungsumgebung BASCOM verwendet. Das ist eine graphische Umgebung speziell für Mikrocontroller von ATMEL® und basiert auf BASIC. Besonders vorteilhaft zum Erstellen erster Programme sind der integrierte Compiler, der Brenner und die Konsole. Bestimmte Funktionen,

wie das Interfaceprotokoll für die serielle Schnittstelle, den 1Wire- und I²C-Bus oder Protokolle zur Ansteuerung verschiedener LC-Displays, sind hier schon in der Umgebung verankert. Zum Testen kann man mit der Konsole einfache Signale über die serielle Schnittstelle senden und empfangen.

Die Routinen zur Erzeugung eines Servoarray, das durch Interruptroutinen gesteuert wird, sind ebenfalls schon verankert.

4.2.3. Testplatine

Um mit der Materie auch in der Praxis vertraut zu werden, wird eine Testplatine angefertigt. Sie soll alle gewünschten Ziele, welche hier gesetzt wurden, erfüllen. Ein Mikrocontroller, der PWM-Signale für Servos generieren kann, eine geeignete Schnittstelle zu einem handelsüblichen PC besitzt und möglichst flexibel zu programmieren ist.

Diese Anforderungen werden von dem „AVR Controllerboard“ von Kai Sachsenheimer ideal erfüllt. Wie der Name erahnen lässt, handelt es sich um eine Experimentierplatine mit einem Mikrocontroller von Atmel, dem AT90S8535. Dieses Bord wurde ursprünglich als Prototypplatine für einfache zweimotorige fahrende Roboter entwickelt und besitzt folgende Eigenschaften:

Technische Daten¹⁸

- Betriebsspannung 8 bis 16 Volt DC
- Stromaufnahme in Ruhe: 90mA
- Platinengröße: 120mm x 100mm

Features¹⁹

- AT90S8535 Controller von Atmel (8-Bit-RISC-Architektur)
- 8 KB Programmspeicher (Flash)
- 512 Bytes EEPROM
- Taktfrequenz: 4 bzw. 8 MHz (mit Jumper wählbar)
- 8 Analoge Eingänge bzw. 8 Digitale Ein/Ausgänge (Port A)
- 8 Digitale Ein/Ausgänge (Port C)
- 4 Lastausgänge für Motoren (max. Strom: 600mA)
- 8 LEDs
- 2 Ausgänge für die Ansteuerung von Servos
- 3 Taster und zwei Schalter zur Eingabe
- Resettaster
- Serielles Interface zur Kommunikation mit einem Terminalprogramm
- Anschluss des Parallelports des PCs zum programmieren des Controllers
- Einstellung der Referenzspannung des A/D-Wandlers
- Kontrasteinstellung für ein LC-Display
- Einfacher Anschluss eines LC-Displays

Eigenschaften des Controllers²⁰

- AVR® - Low-power und High-performance RISC-Architektur
- Memories
 - o 8 KB Flash (1000 Schreib-/Lesezyklen)
 - o 512 B EEPROM (100.000 Schreib-/Lesezyklen)
 - o 512 B SRAM
- Peripherie
 - o 8-Kanal 10-Bit ADC
 - o programmierbare UART
 - o Master/Slave SPI Serial Interface
 - o Zwei 8-Bit Timer/Counter mit separatem Prescaler und Compare-Mode

¹⁸ Sachsenheimer: Bau und Bedienungsanleitung

¹⁹ Sachsenheimer: Bau und Bedienungsanleitung

²⁰ Atmel: Datenblatt AT90S8535

Interface). Dabei entfällt auch das Programmieren des Mikrocontrollers, was je nach Implementation desselben einige Zeit in Anspruch nehmen kann. Sollte der Mikrocontroller nicht ICP-fähig (In Circuit Programming) sein, muss man ihn sogar aus der Schaltung entfernen und in einen Brenner einsetzen, um ihn programmieren zu können.

Um die Entwicklung weiter zu beschleunigen, werden die Anschlüsse für die Servos sowie Komponenten für eine Funkverbindung und der Slot für den Funkempfänger auf der Platine untergebracht. Eine getrennte

Stromversorgung von Servos und elektronischer Schaltung ist ebenfalls integriert. Die Stromversorgung für die Schaltung ist immer bei genau 5V geglättet und für die Servos kann sie über Jumper wahlweise stabilisiert werden. Da die stabilisierte Spannung nur einen Strom von 500 mA liefert, ist die Stabilisierung zur Versorgung der Servos nur für Testzwecke empfehlenswert.

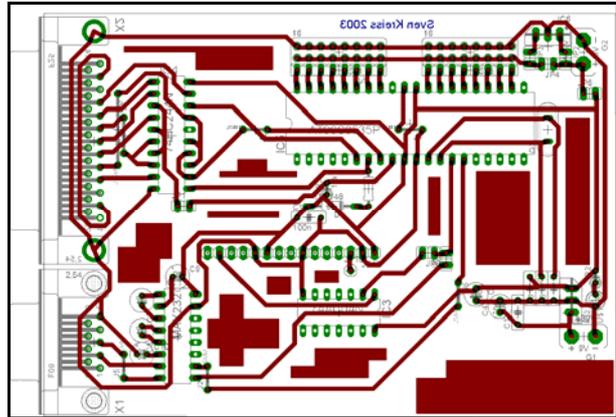


Abbildung 19: Layout des AVR-Boards

PC Software

Durch den heutigen hohen und vielfältigen Standard an Objektorientierten und Webbasierten Programmiersprachen ist es auf sehr einfache Art und Weise möglich, die serielle Schnittstelle komfortabel anzusprechen. Unter Linux ist das selbst direkt aus der Konsole möglich. Durch diesen weit verbreiteten Standard wird man in die Lage versetzt, seine Anwendung nicht nur lokal, sondern auch über das Internet überwachen und steuern zu können.

4.2.5. Funkstrecke

Eine Funkverbindung scheint für diese Art von Roboter erstrebenswert. Jedoch stellt man schnell fest, dass man eine Datenverbindung mit hoher Datenrate und Stabilität benötigt. Außerdem darf sie nicht zu teuer sein.

Für diese Anwendung wurden einfache Funkmodule ohne interne Kodierung verwendet. Die angewendete Hybridtechnik, eine Kombination aus Amplituden- und Frequenzmodulation, verspricht hohe Stabilität. Erste Versuche scheiterten aufgrund von Störungen am Anfang der Datenpakete.

Testprotokoll

Datum: 28.12.2002

Testdauer: 65 min

analysiertes Problem:

Störungen am Anfang von Datenpaketen

Analyse mit einem Oszilloskop



Abbildung 20: Anzeige während einer Datenverbindung mit Sende- und Empfangsdaten



Abbildung 21: Anzeige, wenn keine Daten übertragen werden

Während einer Datenübertragung ist alles in Ordnung. Beim Beenden des Datenstromes wird nach dem RS232 Protokoll ein High-Level als inaktiver Zustand definiert. Aber der Funkempfänger gibt ein Low-Level Signal bei Frequenzen unter 50 Hz heraus. Diese Potentialunterschiede sind auf dem rechten Foto zu erkennen.

Lösung

Der Schaltung wurden zwei Inverter hinzugefügt, einer vor der Übertragung und einer nach dem Empfang. Eine doppelte Inversion des Signals verändert es nicht. Jedoch der inaktive Level zwischen den beiden Invertern ist nun Low und wird durch die Funkübertragung nicht verändert.

Um Funkstrecken zu stabilisieren, werden verschiedene Techniken eingesetzt. Eine der effektivsten ist die *Manchester-Kodierung*. Eine logische Eins wird durch 01 (Null – Eins) und eine logische Null durch 10 (Eins – Null) repräsentiert. In einem Datenstrom können so maximal zwei Einsen und zwei Nullen nebeneinander stehen. Somit ist es nicht möglich, dass die Frequenz der übertragenen Daten auf unter 50 Hz sinken kann, was einen Zusammenbruch zur Folge hätte. Für diese Technik ist allerdings ein elektronisch komplizierter Umbau der Schaltung nötig.

Stattdessen wird in diesem Projekt versucht, eine Häufung von Nullen und Einsen durch *Balanced-Data-Packages* zu verhindern. Ein spezieller Programmcode und nicht eine elektronische Schaltung (*Manchester-Kodierung*) übernimmt die Überwachung. Es wird von der Software garantiert, dass die Datenpakete eine ausgewogene Zahl und keine Häufung von Nullen und Einsen enthalten. Mit dieser Methode gelingt es auch, die Datenrate der bis 2000 Hz getesteten Module auf 9600 Baud zu steigern.

Aufgrund der gegenüber einem Kabel verhältnismäßigen Instabilität und geringen Datenrate wird diese Funkstrecke im Prototyp jedoch nicht eingesetzt.

4.2.6. Algorithmen

An dieser Stelle möchte ich nur auf den Algorithmus zur Errechnung der Servoposition in Bewegungen eingehen.

Bei der Berechnung der aktuellen Position eines Servos in einer Bewegung sind fünf Größen bekannt. Das sind Anfangszeit, Endzeit, Anfangsposition und Endposition der Bewegung sowie die aktuelle Zeit.

Am einfachsten errechnet man die Position mit einer Verhältnisgleichung:

| | |
|--------------------------|-----------------------|
| Pos – aktuelle Position | Zeit – aktuelle Zeit |
| anfPos – Anfangsposition | anfZeit – Anfangszeit |
| endPos – Endposition | endZeit – Endzeit |

$$\frac{Pos - anfPos}{endPos - anfPos} = \frac{Zeit - anfZeit}{endZeit - anfZeit}$$

Diese Gleichung liefert eine lineare Abhängigkeit von Position und Zeit. Stellt man diese Gleichung nach *Pos* um und leitet diese Gleichung zweimal ab (um die Beschleunigung zu erhalten), so wird man bei *Zeit = anfZeit* und *Zeit = endZeit* unendlich starke Beschleunigungen bzw. Verzögerungen feststellen. In der Physik wird dies als „Ruck“ bezeichnet und ist aufgrund der trägen Masse nicht möglich. Daher muss man eine Zuordnung von Zeit und Position ohne unendliche Beschleunigungen finden.

Der zweite Ansatz ist über die Umkehrfunktion von $f(x)=x^3$. Dafür muss der Graph der Funktion durch den Koordinatenursprung und den Punkt (1;1) verlaufen, und sieht daher so aus:

$$f(x) = \frac{1}{2} \sqrt[3]{2x-1} + \frac{1}{2}$$

Diese Funktion ist im Bereich der reellen Zahlen nicht für Argumente kleiner als 0,5 lösbar. Daher wird für den niedrigeren Teil eine andere Funktion benutzt, welche zu der oben genannten Funktion punktsymmetrisch ist:

$$f(x) = -\frac{1}{2} \sqrt[3]{-(2x-1)} + \frac{1}{2}$$

Durch diese Unterscheidung kann man nun auch beliebige Wurzelexponenten einsetzen und bestimmt damit die Charakteristik der Kurve. Leider haben alle diese Funktionen für $x=0$ einen unendlichen Anstieg. Die Geschwindigkeit ist daher an dieser Stelle unendlich hoch. Man

braucht also eine weitere Lösung ohne unendliche Beschleunigungen und ohne unendliche Geschwindigkeiten.

Der dritte Ansatz ist über die trigonometrischen Funktionen. Hier bietet sich der Kosinus an:

$$f(x) = -\frac{1}{2} \cos(\pi x) + \frac{1}{2}$$

Auch hier muss die Funktion wieder geeignet gestaucht und verschoben werden, damit der Graph durch den Koordinatenursprung und den Punkt (1;1) verläuft. Diese Funktion hat weder in der ersten noch in der zweiten Ableitung Polstellen und ist daher geeignet.

Mit den entsprechenden Variablen und der C Syntax entsteht folgende Zuordnung:

```
tmp=(-0.5)*cos(3.141*(t-anfZeit[zae])/(endZeit[zae]-anfZeit[zae]))+0.5)*(endPos[zae]-anfPos[zae])+anfPos[zae];
```

4.3. SENSORIK

Die Sensorik zeichnet diesen Roboter besonders aus. Sie orientiert sich eher an den Härchen einer Haut und nicht am Tastsinn.

Der Tastsinn kann einen auf die Haut ausgeübten Druck in Stärke und Region genau analysieren. Für diese Analyse wird jedoch Zeit beansprucht. Ein Härchen auf der Haut kann den äußeren Druck nicht analysieren. Es gibt lediglich die Auskunft, ob eine Berührung stattfindet. Diese Information kann auch sofort in einen Reflex umgesetzt werden.



Abbildung 22: Foto der Sensoren an den Füßen

Nummer 3 besitzt vier hochempfindliche Taster an jedem Fuß. Die Abtastrate eines Tasters ist ungleich höher als die eines Drucksensors. Die Lage des Fußes wird bei dem kleinsten Kontakt mit dem Boden schnell an die Unebenheiten angepasst. Die Unebenheiten können ohne Kamera und aufwendige Auswertung eines Videosignals (Video-Pattern-Recognition) detektiert werden.

Eine Messung des GCoM ist mit Tastern im Gegensatz zur Messung mit Drucksensoren nicht möglich.

4.4. VISUALISIERUNG

Die Steuersoftware wird unter Linux ausgeführt. Basis für die Entwicklung ist die integrierte Umgebung KDevelop in der Version 2.0.2. Das eigentliche Programm ist eine Konsolenanwendung in der Programmiersprache C.

Ausgangsposition der Bewegungsschleife von der parallelen Beinsetzung auf den Schritt angepasst werden.

4.6. BEWEGUNGSTECHNIKEN

Im Rahmen dieser Arbeit konnte nur die Technik des dynamischen Gehens realisiert werden. Doch, dass die Technik des statischen Gehens trotz etlicher Versuche nicht realisiert werden konnte, kann auch als Erfolg gewertet werden. Damit wird gezeigt, dass das dynamische Gehen viel effizienter und energetisch günstiger ist. Dies war eines der angestrebten Ziele dieser Arbeit.

Bei jedem Schritt wird derzeit noch ein Drehimpuls erzeugt. Dieser entsteht durch die Bewegung des gegenüber dem Oberkörper massereichen Beines nach vorn. Diese Drehung kann man durch eine gezielte Armbewegung in eine Vorwärtsbewegung umwandeln.

Literaturverzeichnis

Primärliteratur:

| | |
|-------------------|---|
| Atmel Corporation | Datenblatt, AT90S8535, 2001 |
| Hitec | Datenblatt, General Servo Information, Korea, English, 2002, Ver. 2.0 |
| Quasar | Datenblatt, QM Transmitter Module, QFMT1-XXX, 2000, Rev. 1.3 |
| Quasar | Datenblatt, QM Receiver Module, QMR1-XXX, 2000, Rev. 1.3 |
| R.F. Solutions | Datenblatt, AM RF Receiver, AM-HRRn-XXX, 1999, Rev. 3.0 |
| R.F. Solutions | Datenblatt, AM Super Regenerative Receivers, AM-HRRn-XXX, 2003 |
| R.F. Solutions | Datenblatt, AM Hybrid Transmitter, AM-RTn-XXX, 1999, Ver. 1.6 |

Sekundärliteratur:

| | |
|---|--|
| Arizona State University; Electrical Engineering Department | Serial Communication, www.eas.asu.edu/~sserc/226/pptfiles/s01p3.PPT |
| Albert, Amos | Intelligente Bahnplanung und Regelung für zweibeinige Roboter, Fortschritt-Berichte VDI – Reihe 8, Nummer 927, Düsseldorf VDI Verlag, 2002 |
| Ben M. Chen Tong H. Lee Venkatakrisnan Venkataramanan | Hard disk drive servo systems, Advances in industrial control London; Berlin; Heidelberg; New York; Barcelona; Hong Kong; Milan; Paris; Singapore; Tokyo Springer, 2002 |
| Bermbach, Rainer | Embedded Controller, Eine Einführung in Hard- und Software, Hanser, München, 2001 |
| Dipl.-Ing. Mustafa Kemal Cevik | Ein Beitrag zur vollständigen digitalen Lageregelung von Servoantrieben ohne Tachogenerator am Beispiel eines Industrieroboters mit verteilter Intelligenz Berlin, 1987, Doktorarbeit |
| Dipl.-Ing. Stefan Faulhaber | Lageregelung für hydraulische Servoantriebe, Fortschritt-Berichte VDI Reihe 8: Mess-, Steuerungs- und Regelungstechnik Nr. 84 VDI Verlag, Aachen, 1985 |
| Dünhölder, Kuno | ISO/OSI, http://members.aol.com/duenhoel/netze501.htm |
| Fraunhofer, Institut Mikroelektronische Schaltungen und Systeme | I ² C-Buscontrollerkern IMS_IIC, http://www.ims.fhg.de/datenblaetter/intellectual_properties/iic/iic-d.pdf |
| Glossar.de | OSI; Neustadt; Deutschland; 1997-2003, http://www.glossar.de |
| Kamm, W. | Der I ² C-Bus, http://sol.ea.rbs.schule.ulm.de/sol/datatech/intfhard/iic/i2c_316.htm |

| | |
|---|--|
| Lichtenegger, Christian | CAN-Bus, http://www.htlmo-klu.ac.at/Projekte_Nachrichtentechnik/1999-2000/5AHN/pr5ahn06/lichte.htm |
| Meik Felser, Robot Building Lab 99 | Überblick: der Mikrocontroller 68HC11, http://www8v1.informatik.uni-erlangen.de/rbl/teilnehmer/Gruppe3/meik.html |
| Microsoft | Servoeinrichtungen, Encarta 2003; Redmond; Washington; USA; 1992 – 2002 |
| Payer, Margarete | Computer vermittelte Kommunikation, Kapitel 3 – Open Systems Interconnection Model, 2001, http://www.payer.de/cmcs/cmcs03.htm |
| Pfaffeneder, Dipl.-Ing. Bernd | Netzwerke und Architekturen, Vom seriellen Interface, http://dbindustrie.svfi.securitas.net/Al/resources/e6c9d46b175.pdf |
| Robotbooks.com | Rug Warrior, Mobile Robots: Inspiration to Implementation, http://www.robotbooks.com/rug_warrior.htm |
| Sachsenheimer, Kai | Bau und Bedienungsanleitung |
| Schossig, Dieter | Aufbau, Anwendung und Programmierung, Mikrocontroller, Seite, Göttingen |
| TALtech | Introduction to Serial Communications, http://www.taltech.com/TALtech_web/resources/intro-sc.html |
| The INFRARED DATA ASSOCIATION | Technical Summary of "IrDA DATA" and "IrDA CONTROL", http://www.irda.org/standards/standards.asp |
| U.S. Robotics | All about UARTs, http://www.usr.com/support/s-cour/s-cour-docs/10589.htm |
| Universität Bonn, Institut für Informatik | Praktikum Mobile Roboter, Roboter „Bender“, http://web.informatik.uni-bonn.de/I/Lehre/Praktika/Roboter/bender.html |
| Universität Tübingen | Outdoor-Roboter, http://www-ra.informatik.uni-tuebingen.de/forschung/outdoor/welcome.html |
| Wolf, Tobias; Fraunhofer Institut | Serviceroboter im häuslichen Bereich, 2002, http://ls1-www.cs.uni-dortmund.de/~asg/asg/Paper/SeminarRIA02/wolf.ausarbeitung.pdf |

Abkürzungen

| | |
|-----------------------|--|
| CPU | Central processing Unit |
| CRC | Cyclic Redundancy Check |
| CTS | Clear to Send, Sendefreigabe |
| IC | Integrated Circuit Integrierter Schaltkreis |
| IIC, I ² C | Inter Integrated Circuit |
| ms | Millisekunde |
| PC | Personal Computer |
| PFM | pulse-frequency-modulation Pulsfolgemodulation |
| PWM | pulse-width-modulation Pulsbreitenmodulation |
| RAM | Read Access Memory |
| ROM | Read Only Memory |
| SCL | Serial Clock |
| SCU | Serial Control Unit |
| SDA | Serial Data |
| SPI | Serial Peripheral Interface |
| UART | Universal Asynchronous Receiver/Transmitter Universeller asynchroner Empfänger/Sender |
| UI | User Interface, Bedienoberfläche |
| WpA | Wissenschaftlich-praktische Arbeit |
| GCoM | Ground Projection of the Center of Mass |
| ZMP | Zero Moment Point |
| CoP | Center of Pressure |

Sachwortverzeichnis

| | |
|----------------------|--|
| Arbitration | Entscheidung, schiedsrichterlicher Vorgang |
| Baud, Baudrate | Die Baudrate wird in Baud nach dem französischen Techniker Émile Baudot angegeben. |
| Bit | Grundeinheit der digitalen Speicherung in dualen Speichersystemen; ein Bit kann nur Null oder Eins annehmen |
| Byte | eine Gruppe von Bits (heute meist acht) ergeben ein Byte; durch die unterschiedlichen Kombinationen der Bits kann ein Byte mit 8 Bit $2^8=256$ Zustände annehmen |
| Double support phase | Phase des Gehens, in der es genau zwei Standbeine gibt |
| duplex, full- | Modus der seriellen Datenübertragung; hier bidirektional |
| duplex, half- | Modus der seriellen Datenübertragung; nur in eine Richtung zu einer Zeit |
| Embedded Application | wörtliche Übersetzung: „eingebettete Anwendung“ |
| simplex | Modus der seriellen Datenübertragung; Datenfluss generell nur in eine Richtung möglich |
| Single support phase | Phase des Gehens, in der es genau ein Standbein gibt |

Abbildungsverzeichnis

| | |
|--|----|
| GCoM und Füße beim statischen Gehen (Vogelperspektive) | 7 |
| GCoM und Füße beim dynamischen Gehen (Vogelperspektive) | 7 |
| GCoM und Füße beim Laufen (Vogelperspektive) | 8 |
| Alle Humanoiden des RoboCups 2002 in Fukuoka | 9 |
| Asimo beim "Penalty Shot" | 10 |
| Pino | 10 |
| RS232 Eingangsdaten | 12 |
| OSI - Deutsche Bezeichnungen nach DIN ISO 7498 | 13 |
| Prinzip der bitweisen Arbitrierung | 15 |
| Foto - Nummer 3, Totale | 24 |
| Standardbaustein (Nummer 2), verzinkter Stahl | 24 |
| Aluminiumteil (Nummer 3), Abkantung und Bohrung für Verschraubung..... | 25 |
| CAD-Modell eines Oberschenkels | 25 |
| Entwicklung des Oberschenkels | 25 |
| CAD-Modell eines Schienbeins..... | 26 |
| CAD-Modell eines Knöchelteils..... | 26 |
| Foto des Alu-Stücks für den Knöchel..... | 26 |
| Schematische Darstellung des AVR-Boards | 29 |
| Layout des AVR-Boards..... | 30 |
| Anzeige während einer Datenverbindung mit Send- und Empfangsdaten..... | 31 |
| Anzeige, wenn keine Daten übertragen werden | 31 |
| Foto der Sensoren an den Füßen | 33 |
| Screenshot (Bildschirmfoto) der Steuerungssoftware (invertierte Farben)..... | 34 |

Anlage

News vom 06. Juli 2003

(<http://www.heise.de/newsticker/data/ad-06.07.03-001/>)

Robocup-WM: Auftritt der Humanoiden

Sein Name ist Murphy und so sieht er auch aus. Der von Studenten der [Universität Uppsala](#) konstruierte humanoide Roboter ist nach dem Polizisten benannt, der im Film "Robocop" zum Cyborg umgebaut wird, und macht ihm mit seinen etwa zwei Metern Körpergröße alle Ehre. Mit seinen Fähigkeiten dagegen schon weniger: Wie alle anderen Teilnehmer im [Wettbewerb](#) der [Humanoid League](#) musste er heute auf die Bühne getragen werden.

Es ist das erste Mal, dass in Europa humanoide Roboter beim RoboCup in Wettbewerb treten. Insgesamt acht Teams haben sich registriert, einer der Roboter hat allerdings den Weg nach Padua nicht gefunden: Arne, der Anthropomorphic Robot von New Era, befindet sich aufgrund von Transportproblemen immer noch in seiner Heimat St. Petersburg. Fotos des Roboters zeigen, dass er vom Design her [Hondas Asimo](#) ähnelt.

Dessen Nachfolgemodell Hits Dream hatte heute Vormittag am Wettbewerb teilgenommen und die erste Aufgabe, das einminütige Stehen auf einem Bein, souverän gemeistert. Der Roboter konnte dabei sogar noch mit den Armen winken und zeigte im Gesicht, das auf einem auf den Schultern montierten Monitor animiert wird, eine beachtliche Mimik. "Wir haben uns beim Design dieses Roboters einen zeh- bis elfjährigen Jungen vorgestellt, der sich für Fußball begeistert", erklärte Teamleiter Shuji Imura. Mit 120 cm Körpergröße und einem Gewicht von 45 kg kommt Hits Dream diesem Ideal tatsächlich schon recht nahe.

Die übrigen Roboter hatten mit dem einbeinigen Stehen größere Schwierigkeiten. Robo-Erectus von der [Singapore Polytechnic](#) brauchte mehrere Anläufe, ehe er den richtigen Schwerpunkt gefunden hatte und stabil stehen konnte. Thomas Christaller vom Fraunhofer-Institut für Autonome Intelligente Systeme, der den Wettbewerb moderierte, forderte die Zuschauer auf, es selbst einmal zu versuchen. Als viele noch größere Probleme hatten, eine Minute lang ruhig auf einem Bein zu stehen, erklärte er warum: "Sie haben Muskeln, der Roboter dagegen hat Motoren, die er blockieren kann."

Aber das hilft eben nur, wenn es dem Roboter gelingt, einen stabilen Schwerpunkt zu finden. [Isaac](#) vom italienischen Politecnico di Torino schaffte es nicht und fiel schon beim Versuch, das Bein zu heben, immer wieder hin.

Auch Tao-Pie-Pie von der kanadischen University of Manitoba schwankte stark und schien mit dem frei schwebenden Bein immer wieder den Boden zu berühren. Der Schiedsrichter, der direkt daneben stand, wertete den Versuch aber als gültig. Tao-Pie-Pie ist mit einer Größe von lediglich 33 Zentimetern der kleinste Teilnehmer im Wettbewerb der Humanoiden und wird von seinen menschlichen Teammitgliedern geradezu liebevoll behandelt: Stets halten sie bei den Gehversuchen schützend die Hände um das Maschinchen, das ein wenig an das Helferlein von Daniel Düsentrieb erinnert, damit es sich bei einem Sturz ja nicht verletzt. "Seit der letzten Weltmeisterschaft ist Tao-Pie-Pie durch eine Kamera erweitert worden", sagte Teamleiter Jacky Baltés. "Er kann durch einen PC oder auch durch einen PDA gesteuert werden."

Doch wie erging es Murphy? Der versuchte gar nicht erst, auf einem Bein zu stehen. Mit seinem Gesamtgewicht von 180 Kilogramm wäre er wahrscheinlich sofort eingeknickt oder umgekippt. Die schwedischen Teammitglieder wollten aber zeigen, dass er auf zwei Beinen stehen kann und lockerten dafür vorsichtig die Seile, mit denen er am Gerüst aufgehängt war. Tatsächlich konnte er zumindest leichte Stöße ausbalancieren.

"Wir sind froh, dass wir ihn immerhin so weit gebracht haben", sagte Kristin Johannsson, die ihn hinterher mit einem Papiertuch vorsichtig abtupfte, als würde sie einen verletzten Fußballspieler behandeln. Es ging aber nicht um die Beseitigung von Blutspuren, sondern von Öl.

Murphy ist ein studentisches Projekt, das bereits seit zwei Jahren läuft, jedoch nur langsam vorankommt, da sich jede neue Studentengeneration erst neu einarbeiten muss. In einem eigenen Projekt wird Murphys Kopf unter dem Namen Murray entwickelt, benannt nach dem sprechenden Schädel im Computerspiel "Curse of Monkey Island". Er ermöglichte Murphy heute immerhin, sich selbst dem Publikum vorzustellen. Eines Tages sollen Kopf und Körper zu einem einzigen Roboter vereinigt werden. Wie aber wird der dann heißen, Murray oder Murphy? Johannsson zögert mit ihrer Antwort keine Sekunde: "Murphy!" Das passt: Genau die gleiche Antwort gibt der Robocop, als er am Ende des Films gefragt wird: "Guter Schuss, Officer. Wie ist ihr Name?"

(Hans-Arthur Marsiske) / (ad/c't)

Selbstständigkeitserklärung

Hiermit erkläre ich, Sven Kreiß, diese Arbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis aufgeführten Quellen genutzt zu haben.

Ort, Datum

Sven Kreiß