

Einsatz von humanoiden Robotern (Robosapiens V1) in der Lehre

FORSCHUNGSBERICHT

2005/2006

von Frank Röber
(f.roeber@googlemail.com)



Inhaltsverzeichnis

1	EINLEITUNG.....	5
1.1	AUSGANGSSITUATION	5
1.2	AUFGABENSTELLUNG.....	5
2	FRAGESTELLUNGEN UND ZIELE.....	7
2.1	FRAGESTELLUNGEN.....	7
2.2	ZIELE	7
3	GRUNDLAGEN.....	8
3.1	KOMPONENTEN	8
3.2	AUFBAU	9
3.2.1	<i>Mechanischer und statischer Aufbau</i>	9
3.2.2	<i>Elektrotechnischer Aufbau</i>	10
3.2.3	<i>Aktoren und Sensoren</i>	12
3.3	BEDIENKONZEPT UND BEWEGUNGSABLÄUFE.....	13
3.3.1	<i>Motorik der Beine</i>	14
3.3.2	<i>Motorik der Arme</i>	16
4	MODIFIKATIONSMÖGLICHKEITEN.....	17
4.1	NACHBILDUNG DER FERNBEDIENUNGSSIGNALE ÜBER EINEN PC.....	17
4.1.1	<i>Aufbereitung der IR-Signale</i>	17
4.1.2	<i>Beurteilung des Verfahrens</i>	26
4.2	EINBRINGEN EINER BLUETOOTH FUNKVERBINDUNG.....	27
4.2.1	<i>Auswahl geeigneter Bluetooth-Module</i>	27
4.2.2	<i>Kommunikation zwischen PC und Robosapien</i>	28
4.2.3	<i>Benötigte Modifikation</i>	29
4.2.4	<i>Konfiguration der BlueRS+E Module</i>	31
4.2.5	<i>Konfiguration der WinLIRC Software</i>	31
4.2.6	<i>Beurteilung des Verfahrens</i>	32
4.3	ERWEITERUNG DURCH ZWEITEN MIKROCONTROLLER	33
4.3.1	<i>Vorbemerkung</i>	33
4.3.2	<i>Benötigte Komponenten</i>	33
4.3.3	<i>Modifikation</i>	36
4.3.4	<i>Beurteilung des Verfahrens</i>	40
5	SOFTWARE ZUR STEUERUNG DES ROBOSAPIENS	41
5.1	ENTWICKLUNG DER SOFTWARE	41
5.2	IMPLEMENTIERUNG DER SOFTWARE	42
6	ERGEBNISSE.....	44
6.1	ROBOSAPIENS IN DER LEHRE	45
6.1.1	<i>Möglichkeiten der Anwendung</i>	45
6.1.2	<i>Aufwand der Modifikationen</i>	46
6.1.3	<i>Nutzen für die Studierenden</i>	46
6.2	FAZIT UND EMPFEHLUNGEN	47
	QUELLENVERZEICHNIS	48
	ANHANG.....	50

Abbildungsverzeichnis

Abbildung 1: Lieferumfang des Robosapiens.....	8
Abbildung 2: Anordnung der Motoren und Batterien innerhalb des Robosapiens	10
Abbildung 3: Rückseite der Hauptplatine des Robosapiens mit vergossenem Mikrocontroller [CRA04]	11
Abbildung 4: Vorderseite der Hauptplatine des Robosapiens mit Vergrößerung des Motortreibers	11
Abbildung 5: Schematischer Signalablauf bei drücken des Knopfes „Aufheben“ (Pick Up) auf der Fernbedienung	13
Abbildung 6: Prinzip eines Schrittes mit dem Rechten Fuß durch Gewichtsverlagerung auf den Linken	15
Abbildung 7: Ablauf des Schließens einer Hand mit Verlagerung der Handposition	16
Abbildung 8: Schaltbild eines Infrarot Empfängers	18
Abbildung 9: Verwendeter IR-Empfänger.....	19
Abbildung 10: Ausschnitt aus dem automatisierten Lernmodus der Software WinLIRC	20
Abbildung 11: Lernmodus der WinLIRC Software zur Ermittlung von IR-Rohsignalen	21
Abbildung 12: Schaltbild des Hochleistungs-IR-Transmitters [LIT03].....	22
Abbildung 13: Modifiziertes Referenzschaltbild mit zusätzlicher Betriebsspannung von 9V	23
Abbildung 14: Modifizierter Hochleistungstransmitter	24
Abbildung 15: Vorbereitung der WinLIRC Software für den Gebrauch der PC-emulierten Robosapiens- Fernbedienung	25
Abbildung 16: Die sendebereite und fertig konfigurierte WinLIRC Software	25
Abbildung 17: Stollmann BlueRS+E Bluetooth Modul.....	27
Abbildung 18: Schematischer Ablauf einer Bluetooth-Kommunikation zwischen PC und Robosapien	28
Abbildung 19: Schaltbild eines einfachen IR-Senders mit einer Sende-LED [LST03].....	29
Abbildung 20: Schaltbild eines aktiven IR-Senders mit einer Sende-LED [LST03].....	29
Abbildung 21: IR-Sender mit zusätzlicher Spannungsversorgung und SUB-D Anschluss.....	30
Abbildung 22: Autonomes Spannungsversorgungspack für Stollman BlueRS+E-Modul	30
Abbildung 23: Konfigurationsparameter der WinLIRC Software	31
Abbildung 24: AVR Programmierboard mit parallelem Anschluss für den PC.....	34
Abbildung 25: 20-Pin Sockel des AVR Programmierboards mit aufgestecktem AT90S2313 Mikrocontroller	34
Abbildung 26: Screenshot der AVR Programmierumgebung (Version <i>w95s2313v5.exe</i>).....	37
Abbildung 27: Konfiguration des Programms UserPort.exe.....	38
Abbildung 28: Mit AVR Prozessor ausgestatteter Robosapien.....	39
Abbildung 29: Funktion zum ausführen des externen Programms " <i>transmit.exe</i> "	42
Abbildung 30: Grafische Benutzeroberfläche der <i>RoboControl</i> Software	43
Abbildung 31: Beispielfunktion zum Ausführen eines spezifischen Kommandos	43
Abbildung 32: Schaltdiagramm Mikrocontroller.....	50
Abbildung 33: Schaltdiagramm Motortreiber	50

Abbildung 34: Schaltdiagramm Infrarot- und Schallsensor	51
Abbildung 35: Schaltdiagramm Taktgeber und Geräuschverstärker	51
Abbildung 36: Leuchtzustände der LEDs im Kopf des Robosapiens [BRO05].....	52
Abbildung 37: Verbindungsschema des AVR Mikrocontrollers mit dem Robosapien [AIB05].....	54
Abbildung 38: Verbindungsschema der Debug-Konsol mit dem Robosapien [AIB05]	54

1 Einleitung

Die Verfügbarkeit von (humanoiden) Robotern ist dank der rasanten technischen Entwicklung der vergangenen Jahre nicht mehr nur reine Fiktion. Heute werden Roboter in Spielzeuggeschäften für Kinder für einen äußerst günstigen Preis angeboten. Im Preissegment unter 100 €¹ liegt der Spielzeugroboter „Robosapien“, welcher auf Grund seines Einstandpreises und seinen beworbenen Fähigkeiten für die Lehre und Forschung an Schulen und Hochschulen sehr interessant erscheint.

1.1 Ausgangssituation

Im Wintersemester 2005/2006 wird neben dem bereits bestehenden Diplomstudengang an der Fachhochschule Fulda auch ein Bachelorstudiengang angeboten. Im Rahmen dieser Neuausrichtung werden neue Veranstaltungen, wie das Fach „*Robotik*“ aus dem Studienschwerpunkt „*Embedded Systems*“, angeboten. Das praxisorientierte Bachelorstudium sieht hierbei neben den theoretischen Vorlesungen auch zahlreiche Laborübungen, Praktika und Projektarbeiten vor. Über das praxisnahe Arbeiten soll das Verständnis der Studierenden gefördert und gefordert werden.

Als Untersuchungsgegenstand wurde der Spielzeugroboter „Robosapien² V1“ der taiwanesischen Firma WooWee gewählt. Diesen Roboter gibt es günstig in vielen Spielwarengeschäften zu erwerben.

Für diese Arbeit wurden zwei identische Roboter erworben.

1.2 Aufgabenstellung

Gegensand dieser Arbeit ist es, die Verwendbarkeit des Roboters „Robosapien“ für das Fach „*Robotik*“ an der Fachhochschule Fulda zu untersuchen. Da dieser Roboter normalerweise über einer Fernbedienung gesteuert wird, ist zu ermitteln, in wie weit sich die Steuerung über einen PC realisieren lässt. Dabei stehen insbesondere die Durchführbarkeit und die spätere Handhabung für die Studenten im Vordergrund.

¹ Stand: September 2005

² <http://www.robosapienonline.com/>

Der Bericht ist dabei in vier Teile untergliedert. Der erste Teil beschäftigt sich mit den Grundlagen des „Robosapiens“. Der zweite Teil behandelt die verschiedenen Möglichkeiten, den Roboter PC-gestützt zu Steuerung und der dritte Teil befasst sich mit der konkreten Umsetzung. Im vierten und letzten Teil werden die Ergebnisse nochmals zusammengefasst.

2 Fragestellungen und Ziele

2.1 Fragestellungen

Ausgehend von der Aufgabenstellung werden drei Schwerpunktthemen entwickelt und beantwortet:

- Welche Möglichkeiten bestehen in Bezug auf ein Informatikstudium, den Roboter in der Lehre einzusetzen?
- Welche technischen Modifikationen müssen hierfür an dem Roboter vorgenommen werden?
- Welcher Nutzen wird dabei für die Studierenden erzeugt?

2.2 Ziele

Die Fragestellung wird anhand von praktischen Umsetzungen versucht zu beantworten. Hieraus lassen sich wiederum Ziele verfolgen, die zur Beantwortung der Fragestellungen geeignet erscheinen:

- Modifikation eines Roboters vom Typ „Robosapiens“, um diesen über einen externen PC zu steuern.
- Erweiterung der Modifikation, dass die Kontrolle des Roboters kabellos von-statten gehen kann.
- Entwicklung und Implementierung eines einfachen Interfaces, um den Roboter über den Computer mittels einer grafischen Oberfläche programmieren und steuern zu können.

3 Grundlagen

3.1 Komponenten

In der Verpackung des Robosapiens befinden sich neben dem eigentlichen Roboter, ein Plastikbecher (Demonstrationszubehör), eine Infrarot-Fernbedienung sowie eine ausführliche Betriebs- und Bedienungsanleitung.

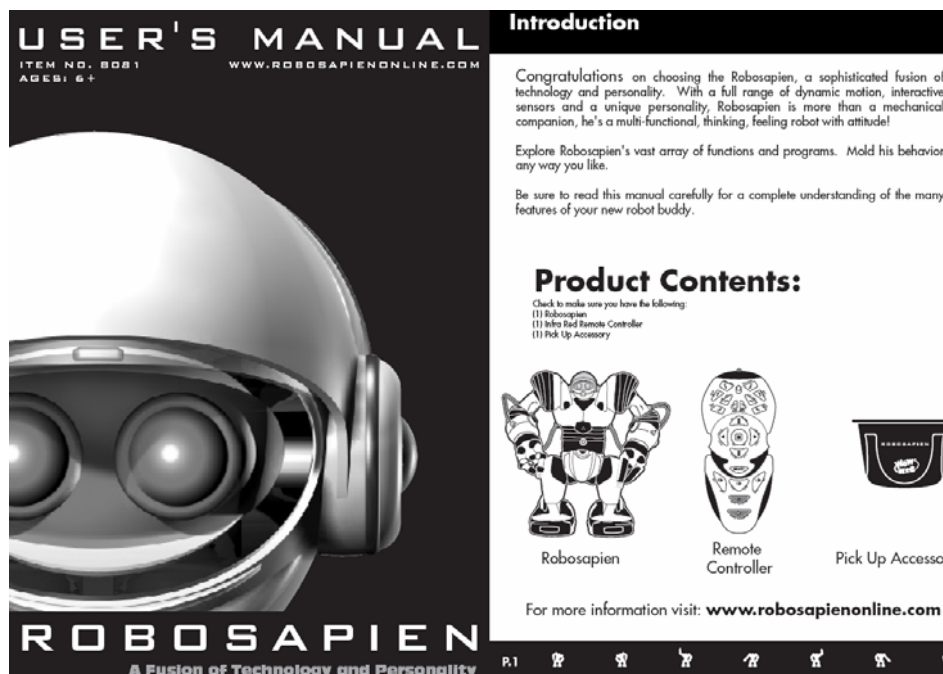


Abbildung 1: Lieferumfang des Robosapiens

3.2 Aufbau

Die Umsetzung humanoider Roboter erfolgt i.d.R. über die Nachbildung des biologischen Vorbilds. Eine einfache Nachahmung der menschlichen Muskeln und Gelenke ist zurzeit noch nicht möglich, so dass diese stark vereinfacht mit Hilfe von Motoren oder hydraulisch/pneumatischen Systemen nachgebildet werden müssen.

3.2.1 Mechanischer und statischer Aufbau

Die äußere Erscheinung des Robosapiens lässt kaum Zweifel aufkommen, dass es sich bei dem Produkt um einen menschenähnlichen Roboter handelt. Die Plastikverkleidung verdeckt ein Metallgerüst, an dessen Bewegungspunkten kleine Elektromotoren mit einem Übersetzungsgetriebe befestigt sind. Für die Bewegungen stehen dem Roboter hierbei insgesamt sieben solcher Antriebspunkte zur Verfügung.

- Jeweils ein Motor für die Bewegung der Beine (Oberschenkel)
- Jeweils ein Motor für die Bewegung des Unterarms (Ellenbogen)
- Jeweils ein Motor für die Bewegung des Oberarms (Schulter)
- Und ein Motor für seitliche Bewegungen des Oberkörpers (Hüfte)

Die Bewegungen der Hände erfolgt über eine mechanische Vorrichtung im Unterarm, die je nach Beugungsgrad dessen die Hand öffnet oder schließt. Ähnlich verhält sich der Kopf, der sich über einen Federmechanismus stets in die Richtung eines hochgehaltenen Arms dreht. Befinden sich beide Oberarme in gleicher vertikaler Ausrichtung, ist der Kopf gerade nach Vorne gerichtet.

Im Fuß des Roboters finden vier Batterien vom Typ Mono D (LR20) platz, die durch ihr Gesamtgewicht von knapp 400g den Schwerpunkt des Roboter weit nach unten legen. Die Füße liegen dabei nicht plan auf dem Boden auf, sondern sind nach innen hin leicht angestellt.

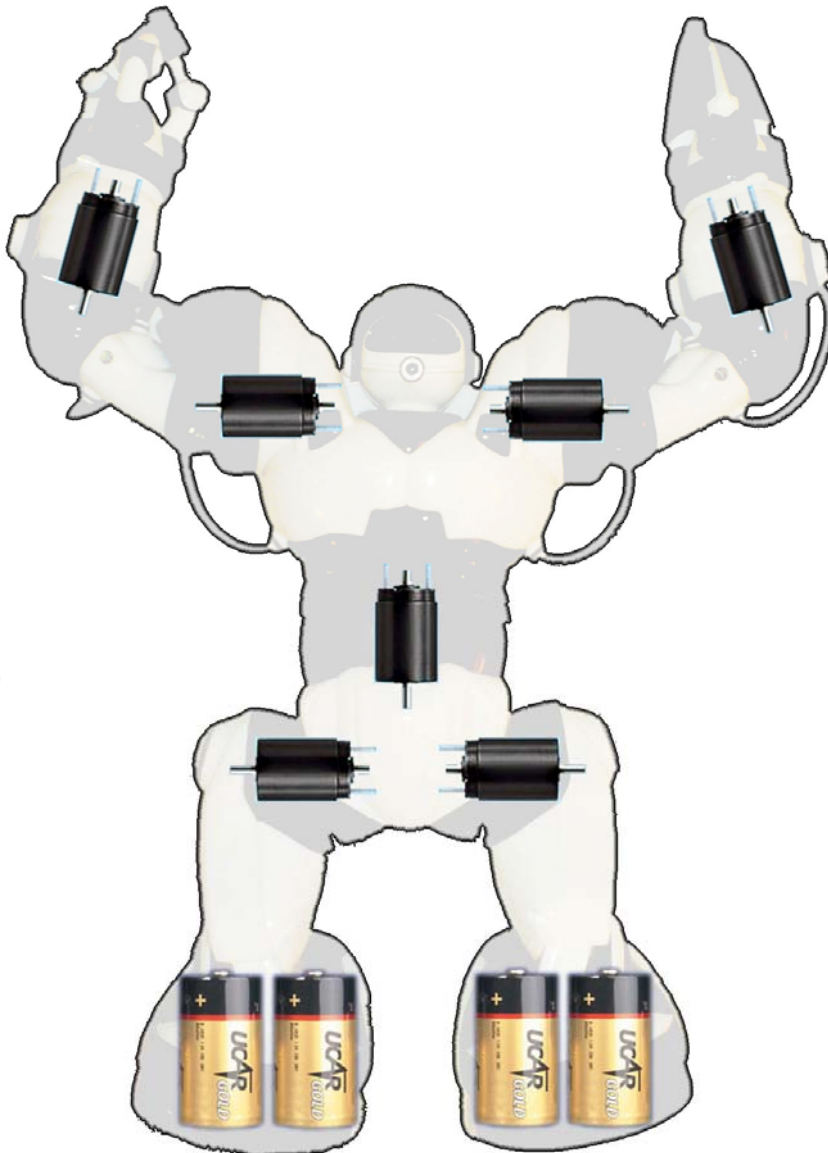


Abbildung 2: Anordnung der Motoren und Batterien innerhalb des Robosapiens

3.2.2 Elektrotechnischer Aufbau

Die zentrale Steuereinheit befindet sich auf der einzigen Platine die der Roboter in sich verbirgt. Der Mikrocontroller, der Steuerbefehle von der Infrarot-Empfangsdiode oder den Sensoren verarbeitet und diese an seine Aktoren weiter leitet, ist nicht näher spezifiziert und befindet sich auf der Rückseite der eingebauten Platine. Die Motoren werden über einen Motortreiber, der von dem Mikrocontroller angesteuert wird, mit Strom versorgt.

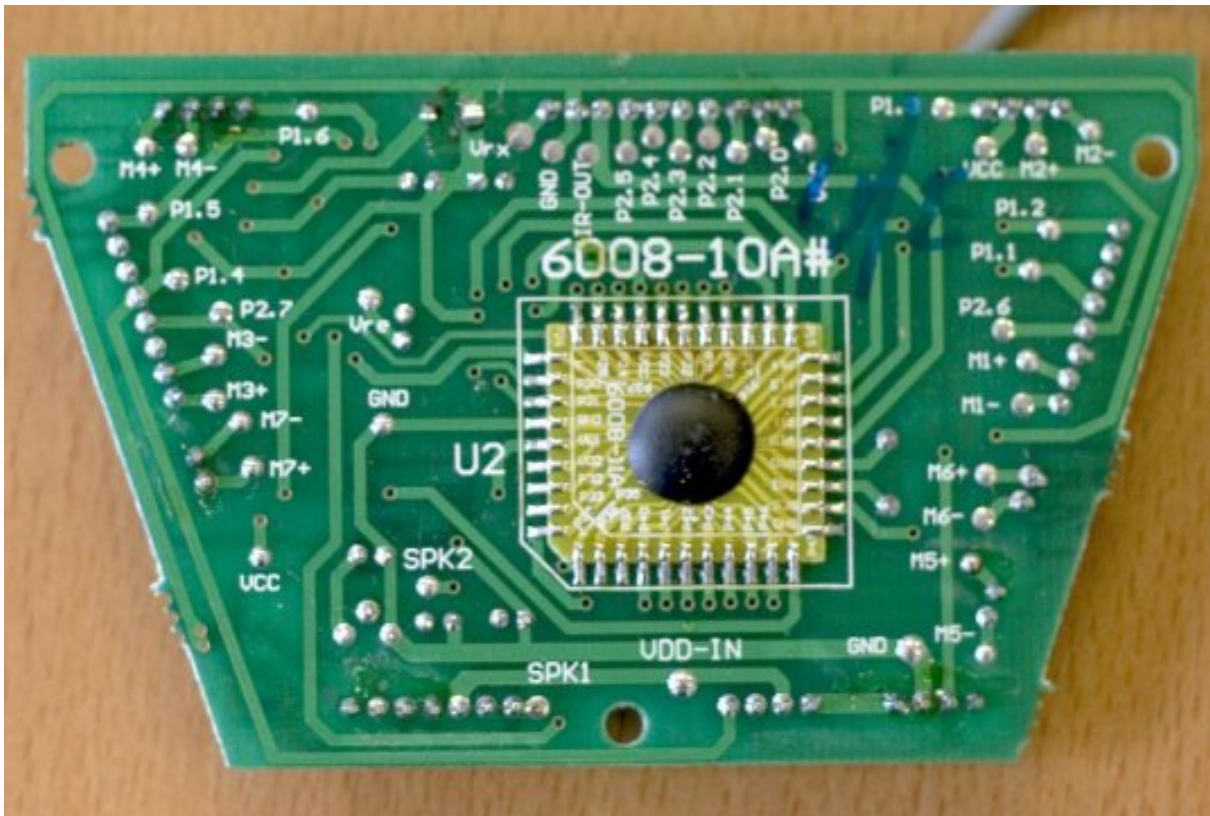


Abbildung 3: Rückseite der Hauptplatine des Robosapiens mit vergossenem Mikrocontroller [CRA04]

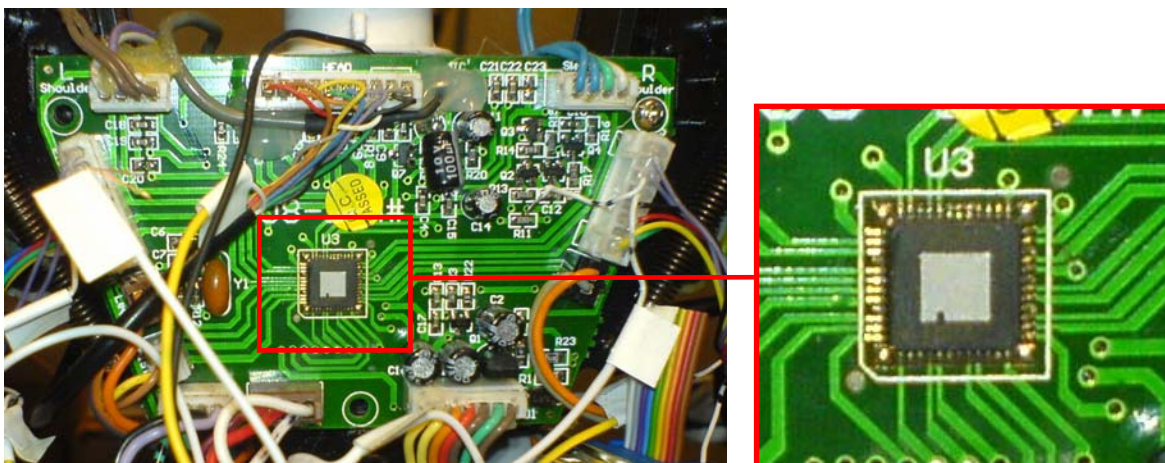


Abbildung 4: Vorderseite der Hauptplatine des Robosapiens mit Vergrößerung des Motortreibers

Im Anhang können die einzelnen Schaltprogramme der aufgezeigten Bauelemente eingesehen werden.

3.2.3 Aktoren und Sensoren

Der Robosapiens verfügt insgesamt über drei unterschiedliche Typen von Aktoren

1. Sieben Motoren für die Motorik
2. Sechs LEDs im Kopf für die Statusanzeige (siehe auch Anhang)
3. Einen Lautsprecher, der gleichzeitig als Akustiksensordient

und zwei unterschiedliche Sensortypen

1. Jeweils zwei Berührungssensoren an jedem Fuß, jeweils ein Berührungssensordient an jeder Hand
2. Ein Akustiksensordient (Lautsprecher)

Über die Infrarot-Fernbedienung werden dem Roboter die gewünschten Befehle zugesendet. Der Mikrocontroller steuert dabei seine einzelnen Aktoren über ein, im Chip vorgegebenes, Programm an. Die Berührungssensoren können dabei den Roboter veranlassen seine programmierten Bewegungen zu unterbrechen. Der Akustiksensordient startet bei Aktivierung (Händeklatschen) ein zuvor ausgewähltes Programm.

3.3 Bedienkonzept und Bewegungsabläufe

Der Roboter wird über die mitgelieferte Infrarot-Fernbedienung mit Befehlen gespeist. Hierbei stehen 67 unterschiedliche Funktionen zur Verfügung, sowie 4 Programmiermodi. Die Signale der Fernbedienung werden von einem Infrarot-Empfänger an den Mikrocontroller auf der Hauptplatine des Roboters weitergeleitet. Dieser gibt die im Mikrochip fest einprogrammierten Steuersignale für die Bewegung an den Motortreiber weiter, welcher wiederum die benötigten Elektromotoren mit Strom versorgt. Eine einzelne und unabhängige Ansteuerung der einzelnen Motoren ist nicht möglich oder vorgesehen.

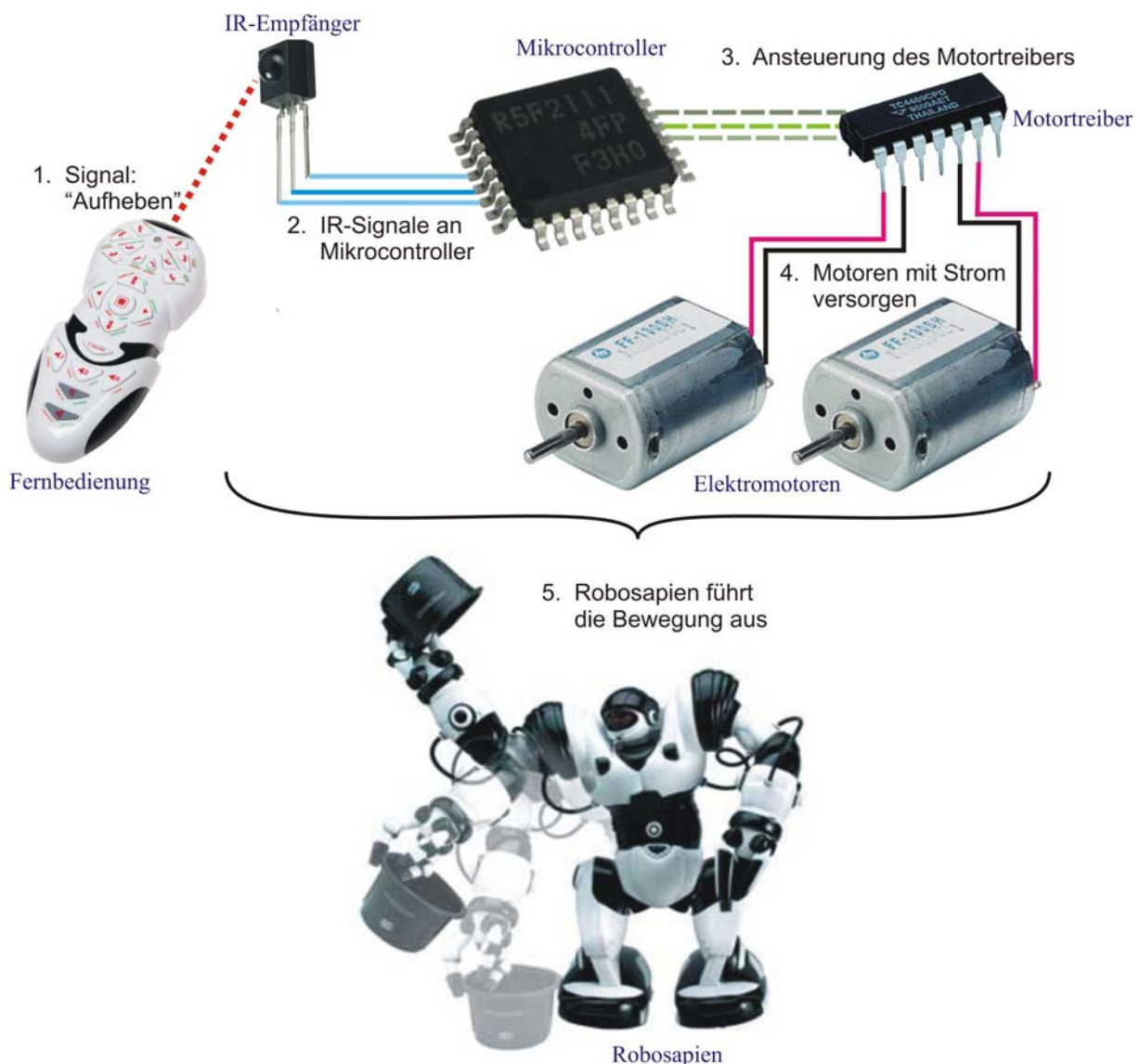


Abbildung 5: Schematischer Signalablauf bei drücken des Knopfes „Aufheben“ (Pick Up) auf der Fernbedienung

3.3.1 Motorik der Beine

Das Laufen auf zwei Beinen ist für einen Roboter, ebenso wie für den Menschen ein Prozess des kontrollierten Fallens. Um einen Schritt zu vollziehen, muss stets dafür gesorgt werden, dass das Gewicht ausbalanciert auf ein Bein verlagert wird. Das andere Bein wird hierbei entlasten, so dass dieses einen Schritt in die gewünschte Richtung durchführen kann. Der Mensch, sowie weit entwickelte humanoide Roboter verfügen über zahlreiche Gleichgewichts- und Bewegungssensoren, so dass die Steuereinheit (Gehirn/Mikrocontroller) den Bewegungsablauf ständig kontrollieren und in ihn ggf. eingreifen kann. Stellt eine Unebenheit der Oberfläche für einen Menschen i.A. kein großes Hindernis dar, sind unwegsame Gelände für einen Roboter auch bis auf weiteres ein Kompromiss aus verfügbaren Sensoren, Rechenleistung und Präzision der Aktoren.

Wie bereits erwähnt, verfügt der Robosapien über keinerlei der soeben angesprochenen Merkmale. Der Roboter muss dabei davon ausgehen, dass die im Mikrocontroller vorprogrammierten Schrittabfolgen ungehindert durchgeführt werden können. Lediglich die Berührungssensoren an den Füßen werden als Interruptsignal an den Mikrocontroller weitergeleitet und unterbrechen den Bewegungsablauf der Beine.

Neben diesen Restriktionen wird ein natürlicher Bewegungsablauf der Beine noch weitere eingeschränkt. Durch den Einsatz nur eines einzigen Motors an der *Hüfte* fehlen dem Roboter zwei charakteristische Gelenke

- das Kniegelenk
- sowie das Fußgelenk.

Ein Schritt erfolgt dabei über die Verlagerung des Oberkörpers (Hüftmotor) auf das Standbein mit anschließender Vorwärtsbewegung des frei schwingenden Beines. Bei der Gewichtsverlagerung kommt die bereits angesprochene Anschrägung der Laufflächen zum Einsatz.

Ein Laufen wie nach biologischen Ursprungs ist bei diesem Bewegungsablauf nicht möglich, da ein Schritt eher einen schiebenden, als einen Anhebenden Charakter besitzt.



Abbildung 6: Prinzip eines Schrittes mit dem Rechten Fuß durch Gewichtsverlagerung auf den Linken

3.3.2 Motorik der Arme

Die Bewegung der Arme erfolgt im Gegensatz zu den Beinen über zwei Motoren. Dem Roboter werden somit ein *Schultergelenk* sowie ein *Ellenbogen* zur Verfügung gestellt. Wie bereits erwähnt entfällt ein Aktor für die Hände, sodass ein Greifen nur über das Ein- und Ausbeugen des *Unterarms* über den Ellenbogenmotor realisierbar bleibt. Die Nutzbarkeit der Hände ist dadurch stark eingeschränkt. Eine präzise Ansteuerung des Greifmechanismus ist kaum möglich, da die Position einer geöffneten Hand durch das schließbedingte Abwinkeln des Unterarms stets von der Position der geschlossenen Hand abweicht.



Abbildung 7: Ablauf des Schließens einer Hand mit Verlagerung der Handposition

4 Modifikationsmöglichkeiten

Eingangs wurde bereits erwähnt, dass die Verwendung des Roboters in der Lehre nur dann Sinn macht, wenn die Studenten ihn über einen PC steuern können. Hierzu gibt es Verschiedene Ansätze, die im Folgenden erläutert werden.

4.1 *Nachbildung der Fernbedienungssignale über einen PC*

Der einfachste Weg den Robosapien über den PC zu steuern scheint in der Nachbildung der Infrarotsignale der Fernbedienung. Hierzu muss eine Möglichkeit bestehen, die Signale der Fernbedienung zu erfassen, um sie später über eine geeignete Vorrichtung über den PC programmgesteuert wiederzugeben. Nachteil dieser Lösung ist die Einschränkung auf die bereits vorprogrammierten Bewegungsabläufe. Eine Möglichkeit zur Ansteuerung der einzelnen Motoren ist nicht zu erwarten.

4.1.1 **Aufbereitung der IR-Signale**

Zur Aufnahme und Wiedergabe von Infrarotsignalen hat sich das Projekt *LIRC* [LIR06] als günstiger Quasistandard herauskristallisiert, das neben freien Quelltexten auch ein einfaches Konzept zur Umsetzung eines Infrarotsenders und -empfängers am seriellen Port eines PC bereitstellt. Beide Ein- und Ausgabemöglichkeiten können Verwendung für das hier vorgestellte Projekt finden.

Anmerkung: *Die Aufnahme der IR-Signale wurde während des Projektes nicht über die nachfolgende Technik realisiert, sollen aber der Vollständigkeit halber erwähnt werden. Während der Forschungsarbeit wurde eine Datei im Internet veröffentlicht, die die nächsten zwei Schritte hinfällig machten.*

LIRC – Der Infrarot-Empfänger

Damit die Infrarot-Signale der Fernbedienung auf den PC gelangen können, steht zunächst die Konstruktion eines geeigneten Empfängers an. Als Schnittstelle dient das etwas betagte RS232 Interface.

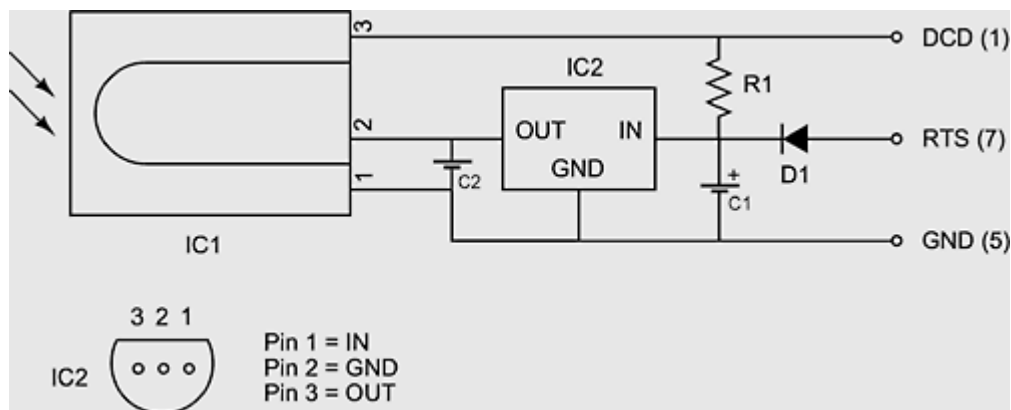


Abbildung 8: Schaltbild eines Infrarot Empfängers

Die Schaltung ist wegen ihrer sechs Bauteile sehr übersichtlich und sollte auch von lötlernerfahrenen Menschen problemlos zusammengesetzt werden können. Eine ausführliche Beschreibung ist unter [BAR00] zu finden.

Stückliste des IR-Receiver

Neben einer 9-Pin SUB-D Buchsenleiste für den Lötanschluss benötigt man folgende Bauelemente:

Diode (D1)	1N4148
Widerstand (R1)	4,7 kOhm
IR-Empfangsmodul (IC1)	TSOP1738 oder SFH506-38
Spannungswandler (IC2)	78L05, 100 mA
Kondensator C1	10 μ F, 16 V
Kondensator C2	100 nF

Tabelle 1: Stückliste für den IR-Empfänger [BAR00]

Pin-Belegung der seriellen Schnittstelle

Name 25-Pin 9-Pin

RTS	4	7	Request To Send (Spannungsversorgung)
GND	7	5	Ground
DCD	8	1	Data Carrier Detect (IR-Signal)

Tabelle 2: Pin-Belegung der Seriellen Schnittstelle für den Anschluss des IR-Empfängers [BAR00]

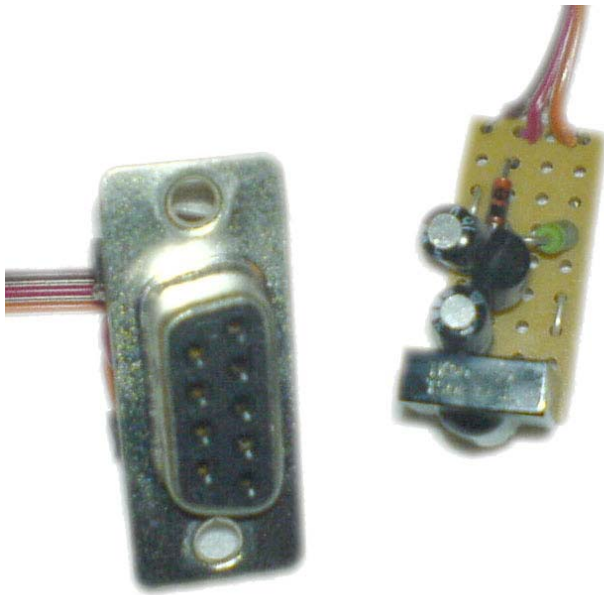


Abbildung 9: Verwendeter IR-Empfänger

LIRC – Die Software zur Aufnahme von IR Signalen

Das *LIRC* Projekt stellt sowohl für linuxbasierte Rechner [LIR06], als auch für Windows Rechner [WLI01] eine fertig kompilierte Software bereit. Für dieses Projekt ist der Einsatz eines Windows-PC vorgesehen. Es wurde die Version WinLirc 0.6.5 verwendet.

Zunächst gilt es, sämtliche Infrarot-Signale der Robosapien-Fernbedienung über die RS232 Schnittstelle mit Hilfe des IR-Empfängers zu ermitteln. Hierzu verfügt die Software über einen automatisierten Lernmodus, indem die am Empfänger ankommenden Signale in einer Konfigurationsdatei gespeichert und somit später wieder abgerufen werden können.

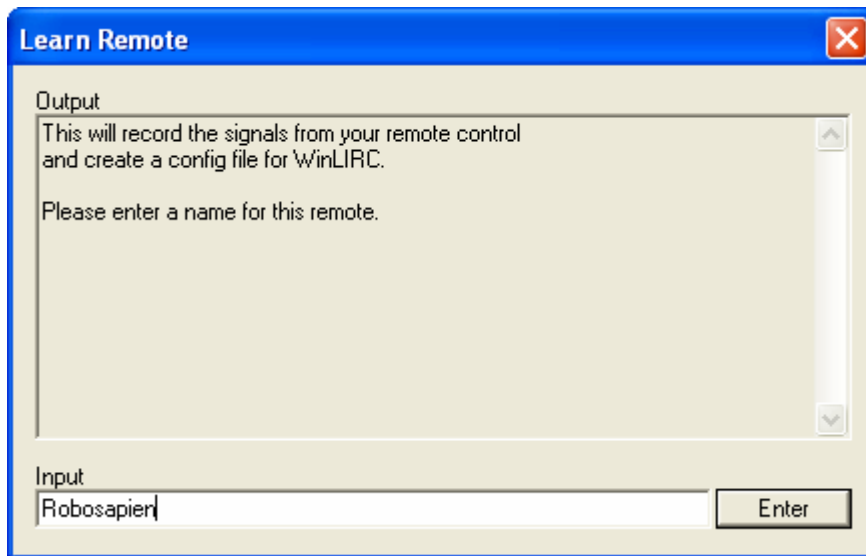


Abbildung 10: Ausschnitt aus dem automatisierten Lernmodus der Software WinLIRC

Problem hierbei ist jedoch die Tatsache, dass die Software zur Analyse jeweils eine Reihe von selben Signalen einer Taste benötigt. Wird z.B. bei einer Fernseher-Fernbedienung durch permanentes Drücken der Lautstärketaste ein fortlaufender Strom an selben Signalen für die Lautstärkeregelung ausgesendet, sendet die Robosapien-Fernbedienung nur ein einziges Mal das Signal aus. Jedoch wird genau diese Funktion von der Software vorausgesetzt, um das Signal sicher analysieren zu können. Etliche Aufwendungen zur Überwindung dieses Problems scheiterten, sodass dieser Lösungsansatz nicht weiter verfolgt werden konnte.

Auch die Funktion zum einlesen eines rohen IR-Datenstroms über die Software wurde aus zeittechnischen Gründen nicht weiter verfolgt.

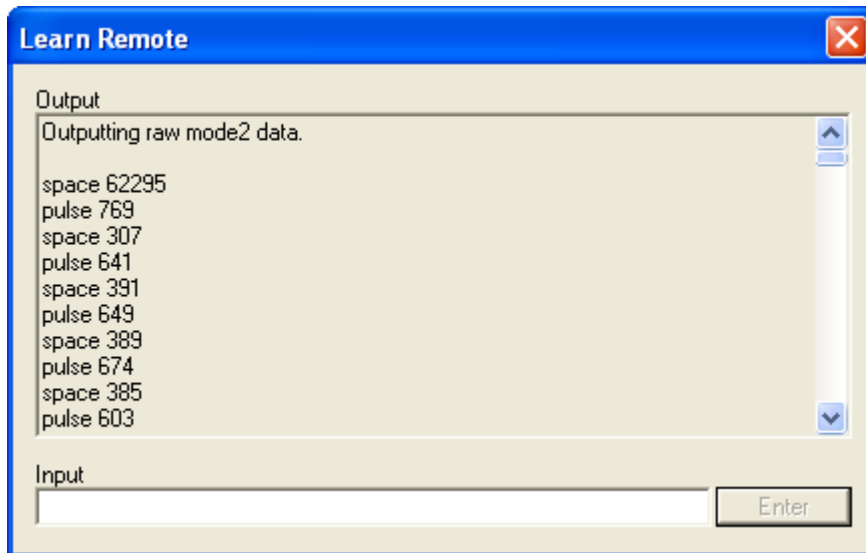


Abbildung 11: Lernmodus der WinLIRC Software zur Ermittlung von IR-Rohsignalen

Zeitgleich zu dieser Arbeit wurde unter [BUE05] eine fertige Datei von Eric Buehl veröffentlicht, die sich ohne weiteres als Konfigurationsdatei in die WinLIRC Software integrieren ließ (siehe Anhang).

LIRC – Der Infrarot-Sender

Damit der PC die gewünschten Infrarot-Signale nach Außen emittieren kann, wird eine IR-Sendeeinheit benötigt. Eine Konstruktionsvorlage wird ebenfalls vom LIRC Projekt bereitgestellt [LIT03].

Zum Einsatz kommt hierbei ein leistungsfähiger IR-Transmitter, der durch die Bestückung von mehr als einer IR-Sendediode eine höhere Reichweite haben soll.

Das unten stehende Schaltbild erläutert den Aufbau dieser Schaltung.

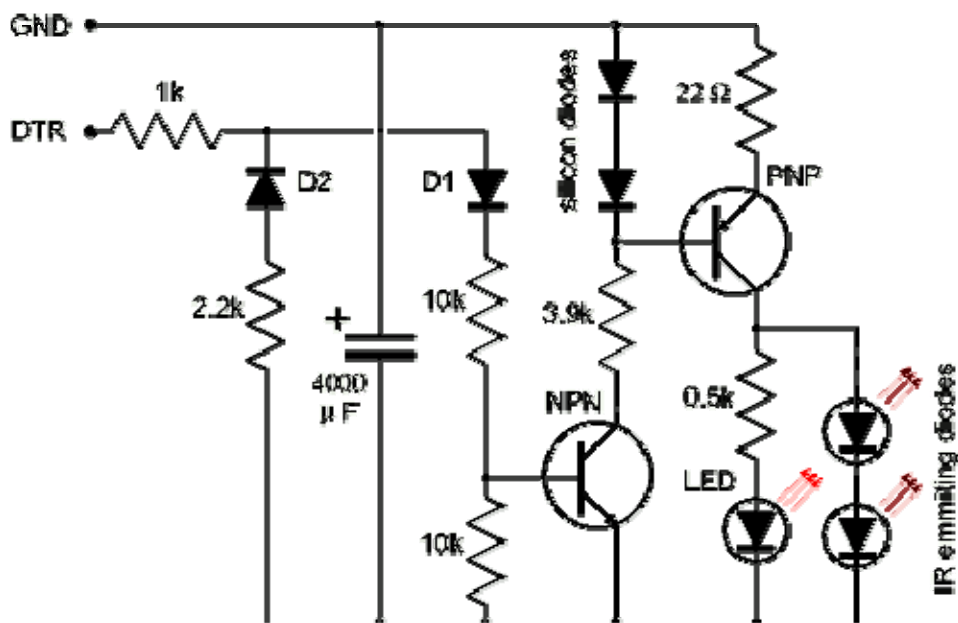


Abbildung 12: Schaltbild des Hochleistungs-IR-Transmitters [LIT03]

Angesteuert wird der Sender wiederum über die RS232 Schnittstelle des PCs. Dabei werden die Sendedioden über die DTR Leitung des COM-Ports, verstärkt über die Dargestellte Transistorschaltung, mit Strom versorgt. Der 4000µF Elektrolytkondensator speichert dabei Energie aus der permanent anliegenden Spannung zwischen GND und DTR. Schaltet der PNP Transistor durch, wird die gespeicherte Energie des Elkos über die IR-Sendedioden abgegeben. Diese Leuchten dadurch deutlich heller als mit der eigentlichen Spannung DTR-GND. Eine normale LED macht das Signal auch für uns Menschen sichtbar, da das menschliche Auge den Infrarot-Wellenbereich nicht wahrnehmen kann.

Damit mehr als nur zwei Sendedioden in die Schaltung integriert werden können, wurde zudem eine Modifikation der Referenzschaltung vorgenommen.

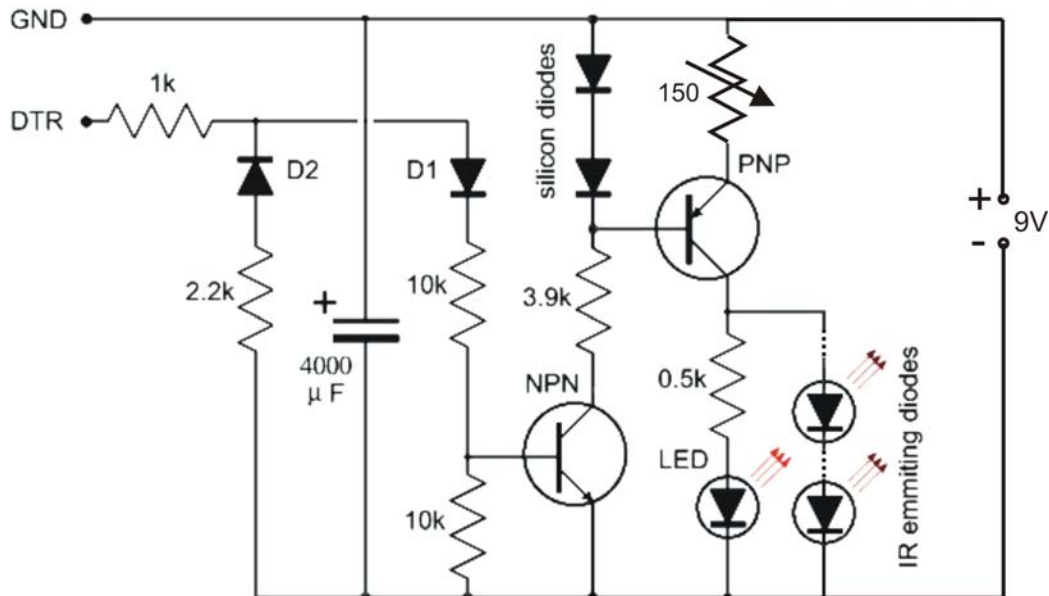


Abbildung 13: Modifiziertes Referenzschaltbild mit zusätzlicher Betriebsspannung von 9V

Die Zusätzliche Betriebsspannung von 9V wird über ein Drehpotentiometer an die Konfiguration der IR-Sendedioden angepasst. Je höher die Anzahl an Sendedioden, desto geringer muss der Widerstand sein.

Hinweis: Da diese Schaltung keine Begrenzungswiderstand für die Zusätzliche Spannungsversorgung besitzt, muss das Poti mit größter Sorgfalt gehandhabt werden, da ein zu niedriger Widerstand die IR-Dioden zerstören würde.

Die Nachstehende Abbildung zeigt den modifizierten Hochleistungstransmitter. Deutlich zu sehen ist hier die extern gehaltene Platine, die mehrere, in Reihe geschaltete Sendedioden aufnehmen kann.

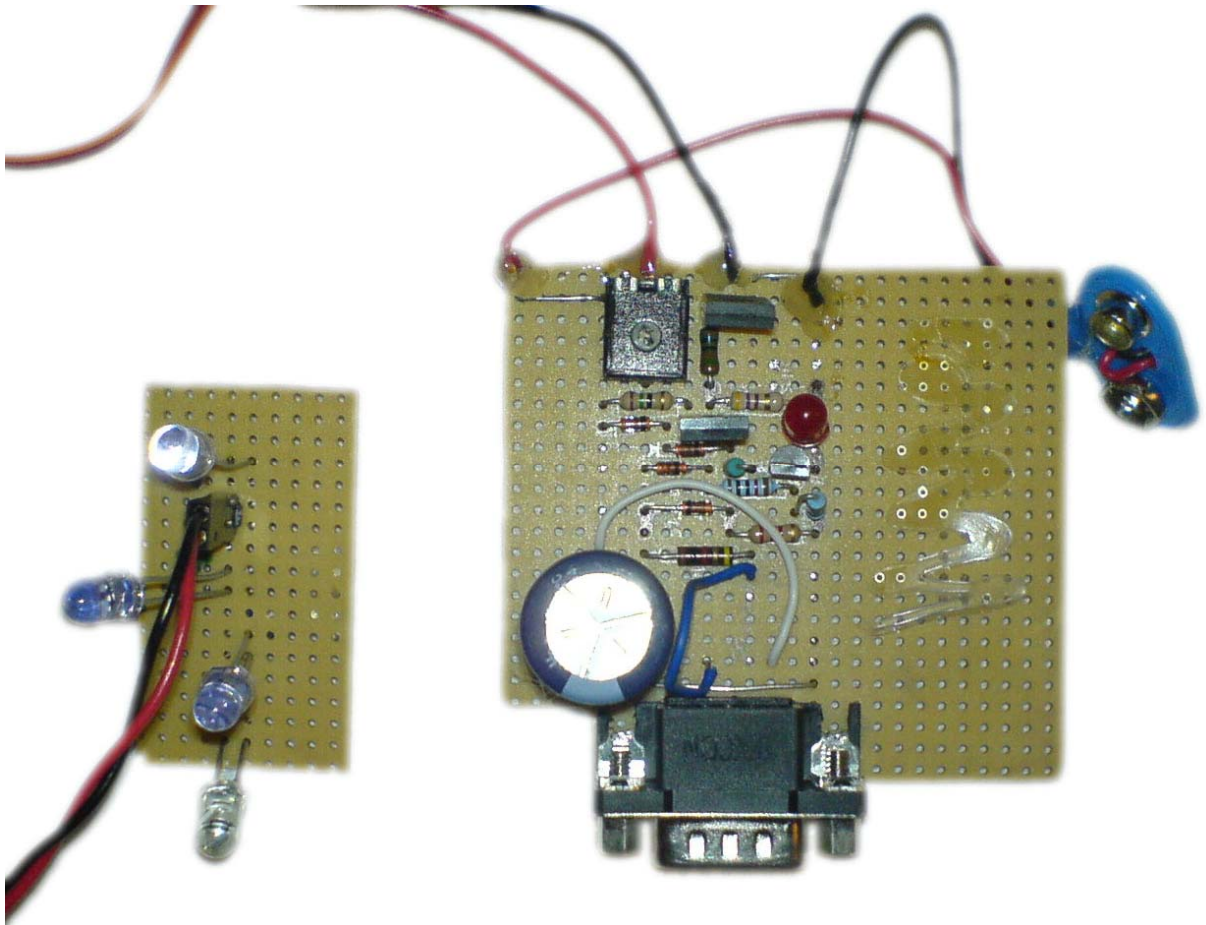


Abbildung 14: Modifizierter Hochleistungstransmitter

LIRC – Die Software zur Wiedergabe von IR Signalen

Mit der fertigen Konfigurationsdatei kann die WinLIRC Software sämtliche Signale der originalen Robosapiens-Fernbedienung emulieren. Dabei wird über die Schaltelemente „*Reconfigure/Browse...*“ die **Robosapiens.conf**-Datei geladen. Das bestätigen mit „OK“ aktiviert die Änderungen.

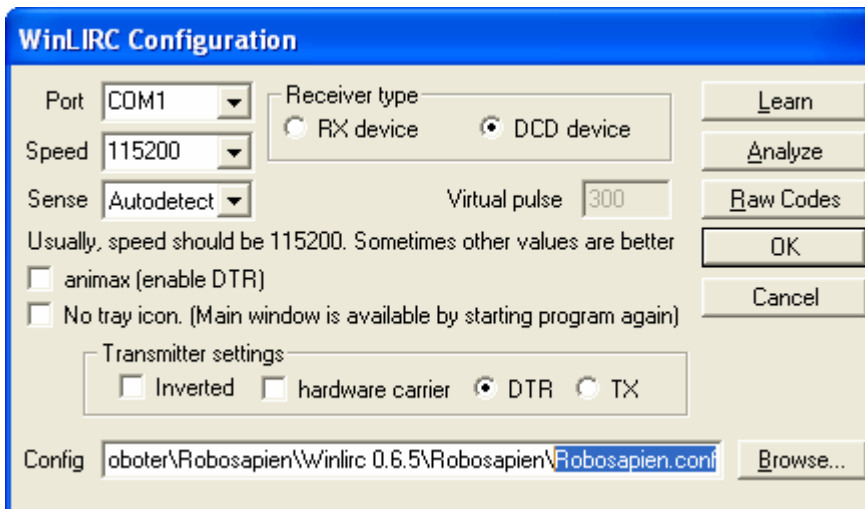


Abbildung 15: Vorbereitung der WinLIRC Software für den Gebrauch der PC-emulierten Robosapiens-Fernbedienung

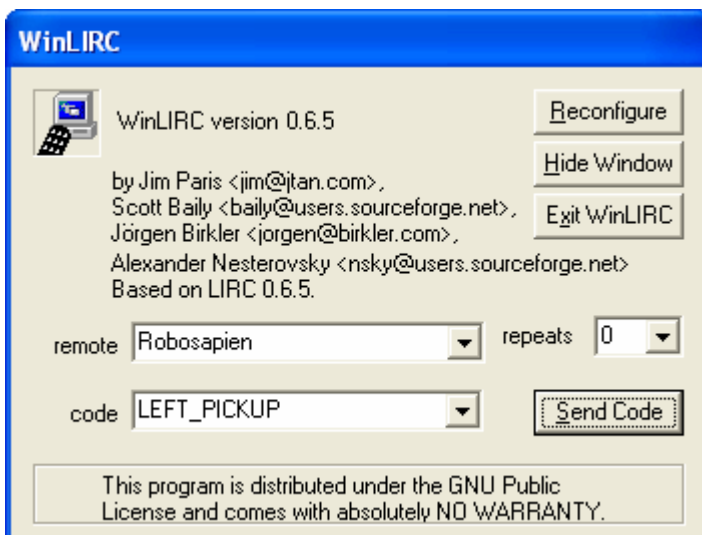


Abbildung 16: Die sendebereite und fertig konfigurierte WinLIRC Software

Ist der IR-Transmitter an die seriellen Schnittstelle angeschlossen und korrekt konfiguriert, kann über das Auswahlménü (wie in **Abbildung 16** gezeigt) ein Befehl ausgewählt, und durch anschließendem Drücken der „*Send Code*“ über die IR-Sendeeinheit ausgestrahlt werden. Das Programm greift dabei auf die externe Datei

„**transmit.exe**“ zu, die ebenso innerhalb einer Konsole verwendet werden kann. Die WinLIRC Software muss jedoch stets im Hintergrund laufen. Die Datei befindet sich im Hauptverzeichnis der WinLIRC-Installation und wird wie folgt aufgerufen:

```
transmit.exe remotename codename reps
```

So wird z.B. bei der konfigurierten Fernbedienung „*Robosapien*“ der Befehl zum Anheben des rechten Arms (*LEFT_ARM_UP*) mit *dreimaliger Wiederholung* wie folgt über die Konsole aufgerufen:

```
transmit.exe Robosapien LEFT_ARM_UP 3
```

4.1.2 Beurteilung des Verfahrens

Für erste Testversuche wurde die separate Diodenplatine über ein langes Kabel an einem hohen Punkt im Raum aufgehängt. Dadurch entstand jedoch wiedererwartend nicht dieselbe Sendeleistung, wie sie die originale Fernbedienung liefert. Mögliche Ursachen könnten hier falsch dimensionierte Bauteile auf der Schaltung sein. Eine Reduzierung des Potentiometers verstärkt zwar die Leuchtkraft der Dioden, jedoch scheinen die Transistoren die gepulsten Signale nicht ausreichend zu schalten, so dass die Reichweite stark eingeschränkt bleibt.

Ein weiteres Defizit erwies sich in der Tatsache, dass durch die vorgestellte Methode keine einzelnen Robosapiens angesteuert werden können. Durch den „globalen“ Transmitter können zwar mehrere Roboter mit Befehlen versorgt werden, jedoch empfangen alle samt dasselbe Signal, was einen individuellen Betrieb einzelner Roboter praktisch unmöglich macht.

Weiterhin konnten erwartungsgemäß keine einzelnen Motoren, sondern nur vorprogrammierte Bewegungen realisiert werden. Jedoch konnte gezeigt werden, dass eine Ansteuerung mit technisch einfach umzusetzenden Mitteln möglich ist.

4.2 Einbringen einer Bluetooth Funkverbindung

Wie bereits im vorangegangenen Abschnitt erläutert, ist die Ansteuerung einzelner Robosapiens mit der vorgestellten Infrarot-Technik nicht möglich. Abhilfe könnte hier eine direkte Funkverbindung schaffen, die die Signale zielgerichtet an die einzelnen Roboter weitergibt.

Im hier vorgestellten Versuchsaufbau sollen die Signale des PCs über eine *Bluetooth*-Funkverbindung direkt an eine IR-Sendeeinheit, die auf dem Roboter montiert ist, weitergegeben werden. Die Sende-Diode soll dabei direkt auf die Empfangsdiode des Roboters gerichtet sein und nach Außen hin abgeschottet werden, um Störsignale zu anderen Robotern zu unterbinden.

4.2.1 Auswahl geeigneter *Bluetooth*-Module

Es liegt nahe, die bereits entwickelte Lösung zu modifizieren, sodass bereits gewonnenen Kenntnisse weiter verwendet werden können. Das unter 4.1 **Nachbildung der Fernbedienungssignale über einen PC** erarbeitete Ergebnis basiert auf einer Kabelverbindung zwischen der RS232 Schnittstelle des PCs und der IR-Sende-Vorrichtung. Die Firma Stollmann produziert kleine und leistungsstarke *Bluetooth*-Module, die ein Kabel ersetzen sollen. Da diese Module bereits für ein anderes Projekt an der FH Fulda besorgt wurden, fiel die Auswahl zur Simulation des seriellen Kabels auf das Produkt BlueRS+E.



Abbildung 17: Stollmann BlueRS+E Bluetooth Modul

4.2.2 Kommunikation zwischen PC und Robosapien

Um eine serielle Verbindung aufzubauen, werden zwei identische Module benötigt. Eines direkt am PC, das andere direkt am Roboter.

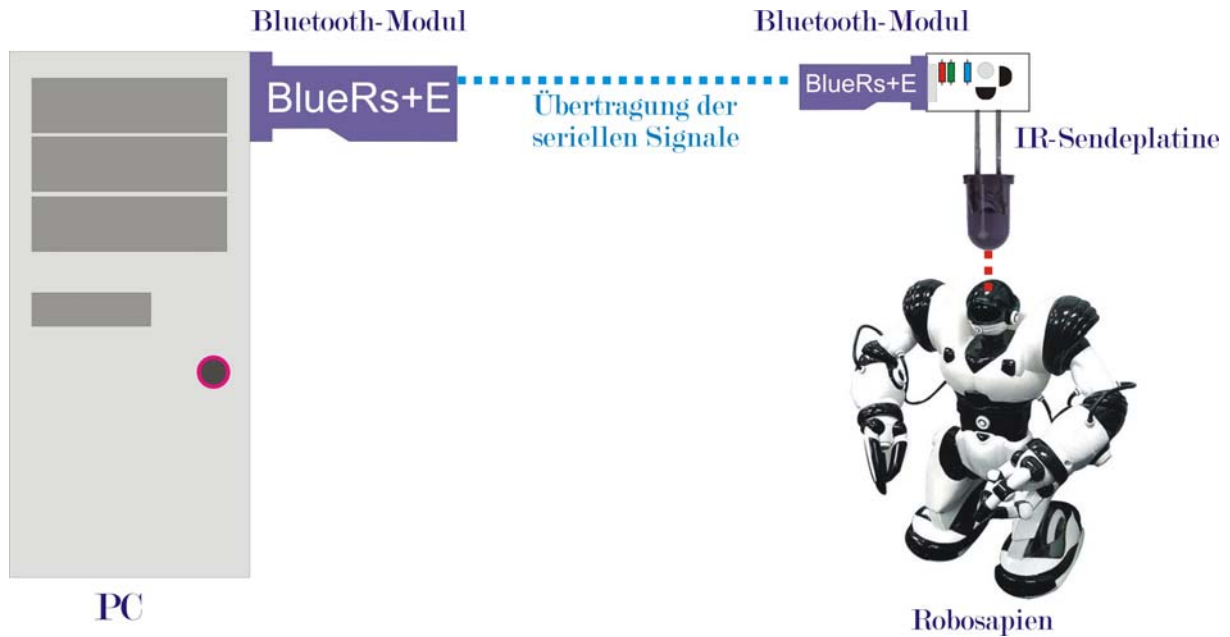


Abbildung 18: Schematischer Ablauf einer Bluetooth-Kommunikation zwischen PC und Robosapien

Werden sämtliche Signalleitungen 1:1 von den beteiligten Modulen durchgereicht, so sollte kein Unterschied zur Verwendung eines Kabels bestehen. Der Aufbau wäre demnach identische mit dem zuvor Vorgestellten und sollte reibungslos funktionieren.

4.2.3 Benötigte Modifikation

Die Infrarotsignale des Senders müssen bei dem beschriebenen Aufbau nicht über einen Hochleistungssender an den Roboter weitergeleitet werden, da der Abstand zwischen IR-Sender und dem im Kopf des Robosapiens befindlichen Empfänger auf wenige Zentimeter reduzieren lässt. Aus diesem Grund wird auf eine vereinfachte Form des oben beschriebenen IR-Transmitters zurückgegriffen. Das LIRC Projekt bietet hierzu unter [LST03] folgende Schaltung.



Abbildung 19: Schaltbild eines einfachen IR-Senders mit einer Sende-LED [LST03]

Für erste Testversuche wurde jedoch eine Schaltung aus gleicher Quelle gewählt, die wiederum über eine zusätzliche 9V Spannungsversorgung verfügt.

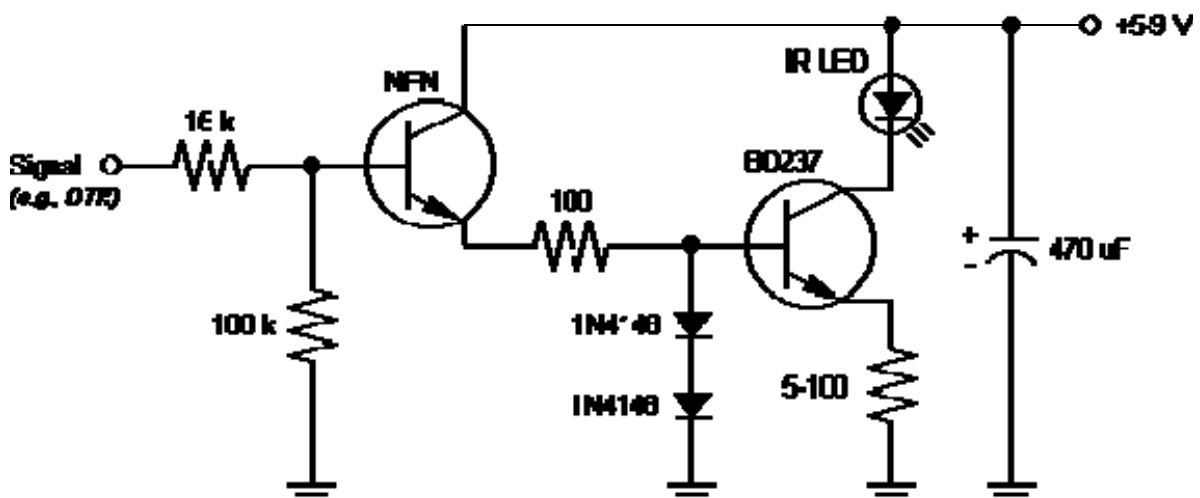


Abbildung 20: Schaltbild eines aktiven IR-Senders mit einer Sende-LED [LST03]

Die Schaltung in **Abbildung 20** wurde um eine 9-Polige SUB-D Steckerbuchse erweitert, um die Anbindung des Bluetooth-Moduls zu vereinfachen.

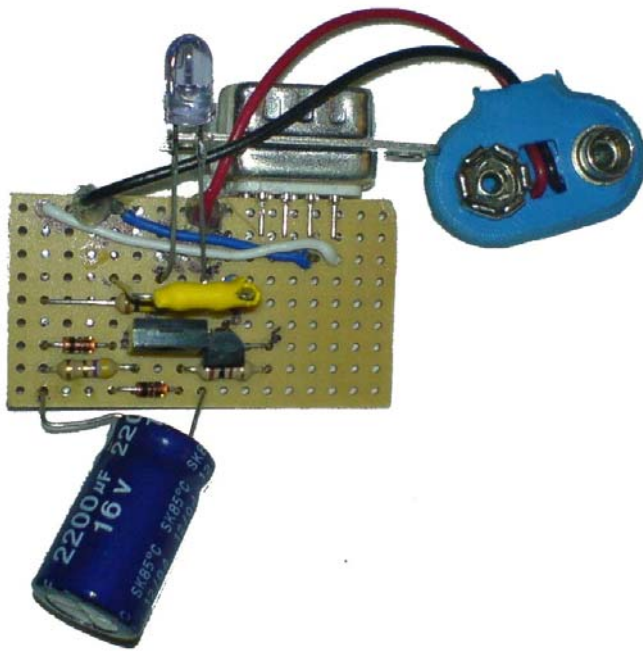


Abbildung 21: IR-Sender mit zusätzlicher Spannungsversorgung und SUB-D Anschluss

Das Bluetooth-Modul benötigt eine Betriebsspannung von 5-6V. Obwohl der Roboter genau diese Spannung aus seinen vier Mono-Zellen liefert, wurde ein kleines Batteriepack mit passender Steckverbindung für das Bluetooth-Modul konstruiert. Dies ermöglicht einen flexibleren Einsatz der gesamten Empfangs- und Sendeeinheit, bestehend aus IR-Sender, Bluetooth-Modul und Batteriepack. Die Gesamte Konstruktion kann somit leicht und ohne zusätzliche Kabelverbindungen von einem Roboter zum nächsten portiert werden.

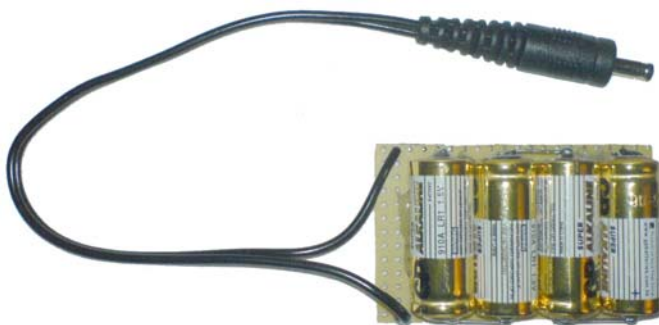


Abbildung 22: Autonomes Spannungsversorgungspack für Stollman BlueRS+E-Modul

Das ganze Paket wird auf den Rücken des Robosapiens mittels Gummibänder fixiert und die Diode der IR-Sendeeinheit auf den Kopf des Roboters ausgerichtet.

4.2.4 Konfiguration der BlueRS+E Module

Bevor eine Verbindung mit den beiden Bluetooth-Adaptoren hergestellt werden kann, muss sichergestellt werden, dass beide dieselben Einstellungen besitzen, sodass die Funkübertragung einwandfrei funktioniert. Das bei den Bluetooth-Adaptoren mitgelieferte Konfigurationstool bietet hierzu die geeigneten Einstellmöglichkeiten. Es ist dabei darauf zu achten, dass sowohl die Übertragungsrate, als auch das Übertragungsprotokoll einheitlich bei beiden Modulen gesetzt wird.

4.2.5 Konfiguration der WinLIRC Software

Generell sind bei der Software keine großen Änderungen durchzuführen. Lediglich die Übertragungsrate (*Speed*), sowie das Einschalten des DTR Signals ist von essentieller Bedeutung. **Abbildung 23** gibt ein Beispiel für eine sinnvolle Konfiguration.

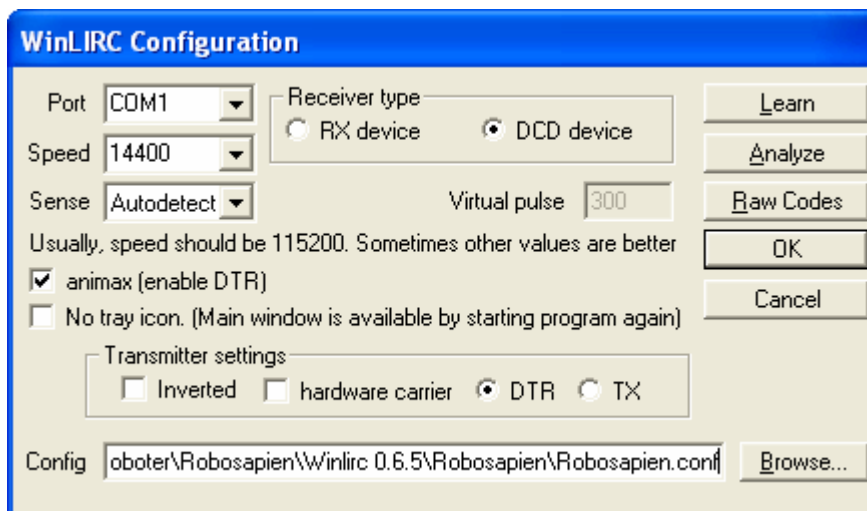


Abbildung 23: Konfigurationsparameter der WinLIRC Software

4.2.6 Beurteilung des Verfahrens

Eine Übertragung über Bluetooth konnte mit der aufgezeigten Methode nicht erfolgreich durchgeführt werden. Aufschluss auf den Fehler gab die Messung mit einem Oszilloskop. Ergaben sich bei der Übertragung mit dem Kabel saubere und beständige Signalfanken, konnte bei der Bluetooth-Übertragung zwar das selbe Signalmuster erkannt werden, jedoch brachen die seriellen *Active-High* Signale kurz nach der Flankenerhebung wieder zusammen. Abhilfe konnte innerhalb des Forschungszeitraumes weder durch die Manipulation verschiedener Parameter an den Bluetooth-Adaptoren, noch an der Übertragungssoftware geschaffen werden.

Ein möglicher Grund könnte in dem fehlenden Rückkanal, des auf dem Robosapien installierten Adapters, liegen. Möglicherweise könnten die Probleme mit der Neuerscheinung einer überarbeiteten Firmware der Module behoben werden.

Ein Mitarbeiter von Stollmann analysierte das Problem, konnte jedoch bis zum Ende dieser Arbeit keine abhilfeschaftenden Maßnahmen einbringen.

4.3 Erweiterung durch zweiten Mikrocontroller

Die bereits angesprochenen Modifikationsvorschläge basieren auf der Emulation der IR-Fernbedienung. Somit ist es nicht möglich, die Funktionen des Robosapiens zu erweitern. Sollte es möglich sein, den vorhandenen Mikrocontroller durch einen zweiten zu ersetzen bzw. zu erweitern, könnte der Roboter um zusätzliche Funktionen erweitert werden.

Die Idee zu dieser Modifikation stammt von [AIB05]. Das dort aufgezeigte Projekt verwendet eine kleine CPU mit AVR RISC Architektur. Der hier verwendete *Atmel*[®] Chip „**AT90S2313**“ basiert auf einem 8-bit CMOS Mikrocontroller und bietet 2K Byte an Flash-Speicher für Programme, sowie 128 Byte Arbeitsspeicher. Durch sein integriertes serielles Interface kann er Befehle von einem PC über die RS232 Schnittstelle annehmen und verarbeiten.

4.3.1 Vorbemerkung

Bevor die Umrüstung erfolgt, sollten einige Merkmale beachtet werden.

Anders als bei den vorangegangenen Methoden sind hier mechanische und elektronische Modifikationen an dem Roboter vorzunehmen. Aus diesem Grund ist es ratsam, stets einen funktionierenden Roboter als Backup-Lösung zur Verfügung zu haben.

***Hinweis:** Fehler bei der Umsetzung können u.U. zum Totalausfall des Robosapiens oder zu defekten des PCs führen!*

4.3.2 Benötigte Komponenten

AVR Chip

Zunächst wird ein Chip vom Typ *AT90S2313-10PI* benötigt. Dieser muss über eine geeignete Schaltung programmiert werden. Dabei wird über einen PC eine Firmware in den Flashspeicher des Prozessors geschrieben.

AVR Programmierhardware

Zum Programmieren wurde das *AT90S2313 Development Board* der Firma *Futurelec* [FUT05] verwendet.

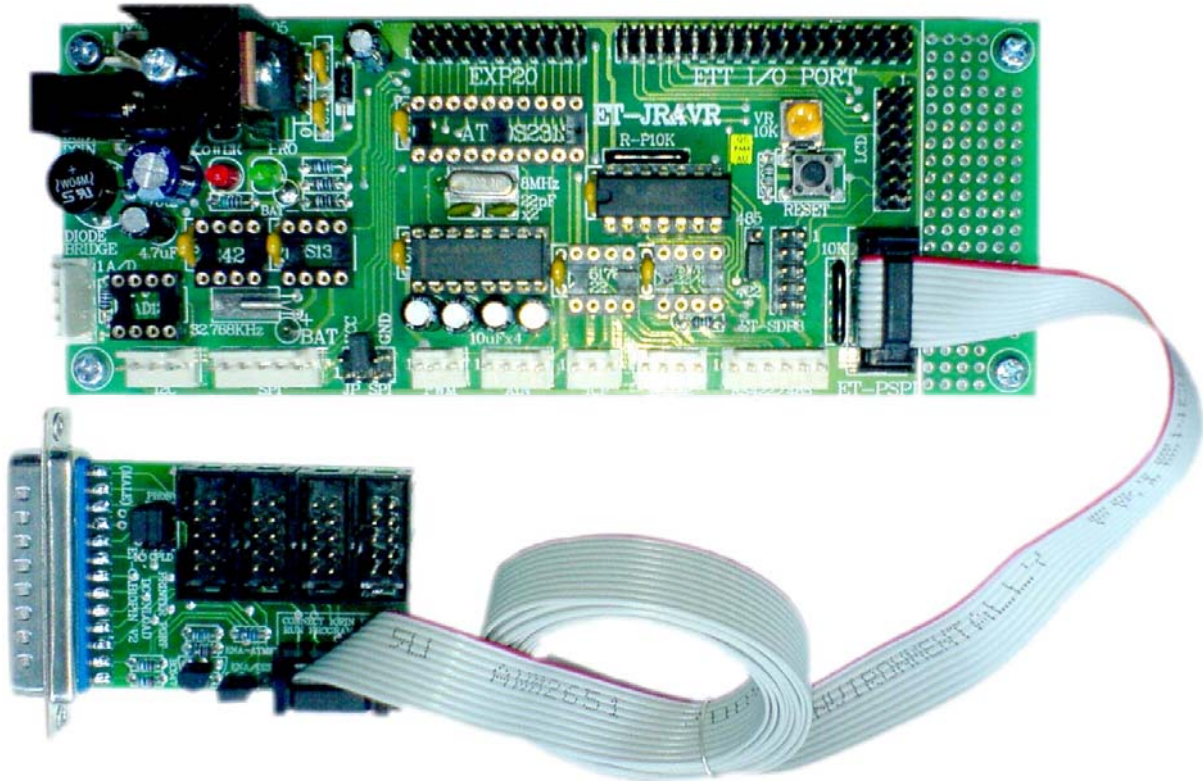


Abbildung 24: AVR Programmierboard mit parallelem Anschluss für den PC

Der zu programmierende *Atmel*[®] Chip wird auf den beschrifteten Sockel gesteckt.



Abbildung 25: 20-Pin Sockel des AVR Programmierboards mit aufgestecktem AT90S2313 Mikrocontroller

Software – Firmware für ATMEL AT90S2313 Mikrocontroller

Der *Atmel*[®] Prozessor benötigt für den Betrieb eine Firmware, die zum einen eingespeicherte Befehle an den Robosapiens weitergeben kann und zum anderen die Be-

fehle der RS232 Schnittstelle umleitet. Als Grundlage für die hier vorgestellte Modifikation dient die noch in der Entwicklungsphase befindliche Firmware von [AIB05].

Software – AVR Compiler

Eine binäre Datei kann von der zuvor erwähnten Quelle (noch) nicht bezogen werden. Deshalb wird ein Compiler benötigt, der die Quelldateien [ROE05] der Firmware zu einer binären .hex Datei übersetzt. Der AVR Compiler ist unter [WAV06] zu finden.

Software – AVR Programmiersoftware

Ist die Firmware vorhanden, muss diese schließlich mit einer geeigneten Software über den Parallelport des PCs auf den Mikrocontroller geladen werden. Hierzu wird die Software *ws2313.exe* aus [ROE05] verwendet. Weiterhin wird aus selber Quelle u.U. das Programm *UserPort.exe* benötigt, welches unter modernen 32-Bit Betriebssystemen den direkten Zugriff auf den Parallelport zulässt.

4.3.3 Modifikation

Bevor eine Modifikation am Roboter vorgenommen wird, sollte der nicht triviale Umgang mit der AVR Programmierung durchgeführt werden. Als Anhaltspunkt soll folgende Beschreibung Hilfestellung leisten:

Programmieren des AVR Chips

1. Installation des AVR Compilers (<http://winavr.sourceforge.net>) [WAV06]
2. Entpacken von http://www.aibohack.com/robosap/sapavr_fw01src.zip [AIB05]
 - a. In das Verzeichnis der entpackten Dateien wechseln und folgende Kommandos eingeben

```
make clean
make
```
 - b. als Resultat wird die Datei namens „*sapavr.hex*“ erzeugt, welche in den AT90S2313-10PI geladen werden kann.
3. Das Programmieren des AT90S2313 ATMEL-ICs ist u. U. ein bisschen trickreich.
 - a. Zunächst muss das Experimentierboard (ET-PSPI Pfostensteckerbuchse) mit dem mitgelieferten Parallelport-Adapter (ATMEL-SPI Pfostensteckerbuchse) über ein 10-poliges Flachbandkabel verbunden werden. Nachdem der Parallelport-Adapter an den Computer angeschlossen wurde, kann nun die (nicht mitgelieferte) Stromversorgung angeschlossen werden. Hierbei ist darauf zu achten, dass die Spannung zwischen 9-12V liegt und das Netzgerät eine Mindestleistung von 800 mA bietet. Die Polung des Steckers ist wie folgt.
 - i. Innen PLUS, Außen MINUS
 - ii. + → O) ← -
 - b. Das mitgelieferte Programm **w95s2313v5.exe** [ROE05] des Experimentierboards ET-JRAVR funktionierte beim Autor dieser Arbeit nicht einwandfrei. Deshalb wurde eine ältere Version (**ws2313.exe** [ROE05]) des Programms verwendet.

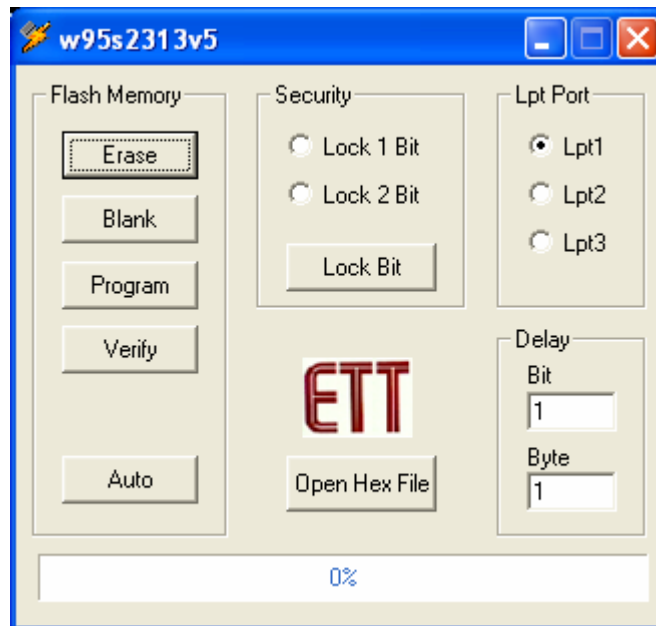


Abbildung 26: Screenshot der AVR Programmierumgebung (Version *w95s2313v5.exe*)

- c. Als erste Hürde musste ein direkter Zugriff auf den Parallelport des Rechners gewährleistet sein, welcher unter den Betriebssystemen Windows NT/2000/XP/... standardmäßig nicht gestattet ist. Da jedoch auch die ältere Version unter Windows XP nicht zum laufen gebracht werden konnte (*siehe auch HINWEISE FÜR DIE VERWEUNDUNG UNTER WinNT/2000/XP/...*), wurde Windows 98 unter VMWare verwendet (Parallelport muss hier von Hand in die Konfiguration der VMWare (über *Optionen*) hinzugefügt werden).
- d. Nach Aufruf des Programms muss zunächst eine hex-Datei ausgewählt werden. Die ältere Version **ws2313.exe** zeigt im DateiDialog-Browser die benötigte Datei *sapavr.hex* nicht an, so dass der Pfad ausgewählt und der Name der Datei manuell eingetragen werden muss.
- e. Nach Betätigung der Schaltfläche „*Program*“ wird die Datei in den IC geladen (grüne LED auf dem Experimentierboard blinkt)

HINWEIS FÜR DIE VERWENDUNG UNTER WinNT/2000/XP/...

- f. Diese Konfiguration wurde vom Autor dieser Arbeit nicht getestet, dennoch sollen hier ein paar hilfreiche Tipps zum ausprobieren aufgestellt werden.

Da wie bereits erwähnt kein direkter Zugriff auf den Parallelport unter diesen Betriebssystemen gestattet ist, muss dieser zunächst gewährt werden.

Hierzu wird ein weiteres Programm namens **UserPort.exe** [ROE05] mitgeliefert. Dieses Programm erlaubt den direkten Zugriff auf den Parallelport.

- g. Nach dem Starten des Programms muss evtl. noch die Adresse des aktuellen Parallelports (z.B. LPT1) eingetragen werden (hier: 0x037b-0x037F), der Liste hinzugefügt werden (Schaltfläche „Add“)

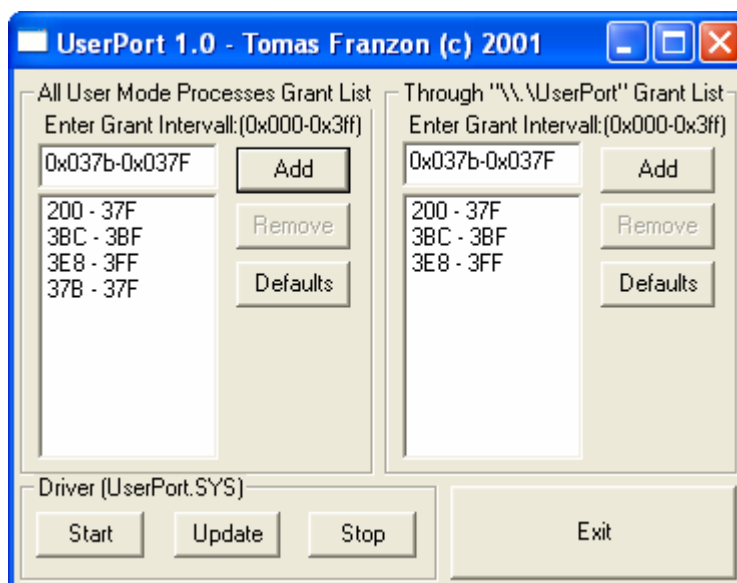


Abbildung 27: Konfiguration des Programms UserPort.exe

- h. und anschließend der UserPort aktiviert werden (Schaltfläche: „Start“)
- i. Hiernach kann normal fortgefahren werden.

Änderung an der Hardware des Robosapiens

Ist die Firmware ohne Fehler auf den AVR Mikrocontroller übertragen worden, kann nun die Modifikation der Hardware erfolgen. Hierzu soll auf die Internetadresse des Urhebers verwiesen werden [AIB05]. In dieser detaillierten und bebilderten Darstellung werden einige Signalkabel des Roboters an den neuen AVR Chip umgeleitet.

Die daraus resultierende Konfiguration sollte nun in der Lage sein, Signale über eine serielle Verbindung vom PC aus zu erhalten und diese wiederum an den Robotereigenen Mikrocontroller weiterzuleiten. Das Programm der Firmware kann dabei bestimmte Vorgaben des ursprünglichen Prozessors überschreiben (Reaktion beim Berühren der Sensoren, Bewegungsabläufe, etc.)

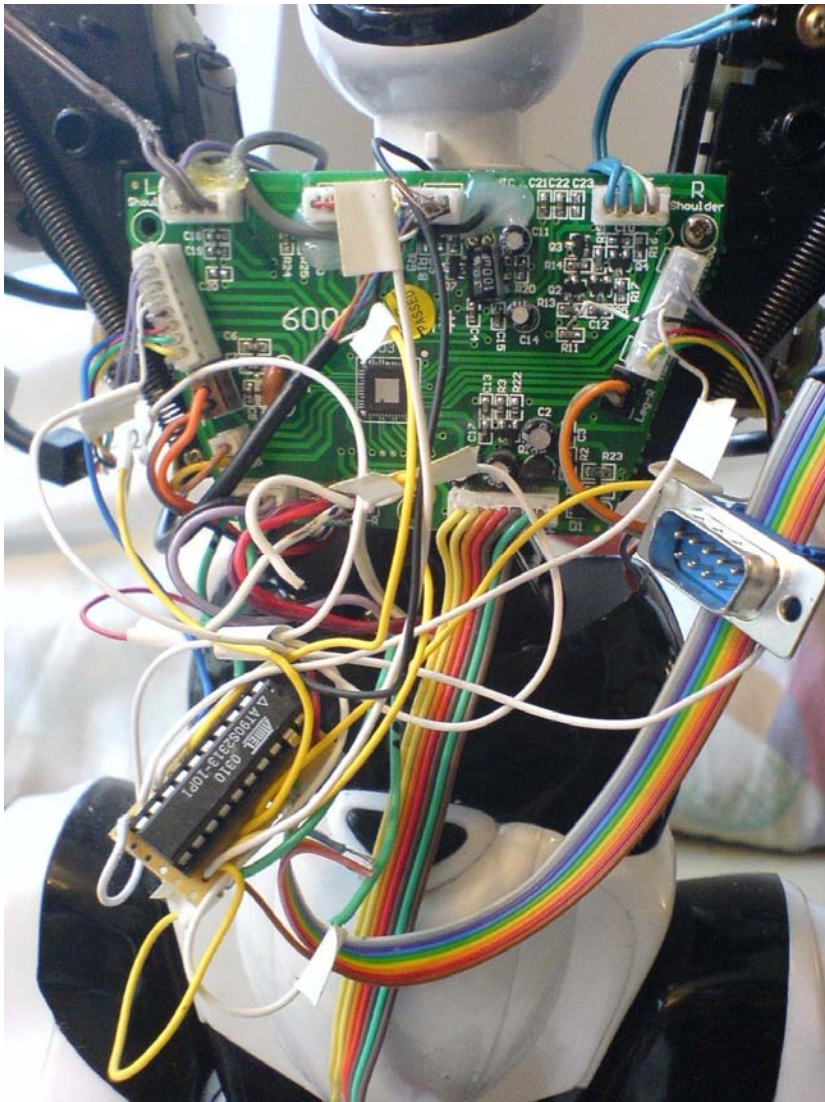


Abbildung 28: Mit AVR Prozessor ausgestatteter Robosapien

4.3.4 Beurteilung des Verfahrens

Allein die Programmierung des AVR Prozessors erfordert mit den hier aufgezeigten Mitteln eine Menge Geduld und Motivation. Nach der Modifikation konnte der Roboter allerdings über das serielle Interface vom PC aus gesteuert werden. Die Verwendung der zuvor vorgestellten Bluetooth-Module führte ebenso zum Erfolg.

Während einer erneuten Programmierung des AVR Prozessors brach jedoch wiederholt die Spannungsversorgung zusammen, sodass der Chip hiernach nicht mehr ansprechbar war. Während des Bearbeitungszeitraumes dieser Arbeit konnten keine neuen *Atmel*[®] Mikrocontroller beschafft werden, so dass eine weitere Verfolgung dieses Verfahrens nicht möglich war.

In Kombination mit den Bluetooth-Modulen und einer neuen Programmiersoftware sollte es jedoch möglich sein, die Modifikation nach den eingangs erwähnten Vorstellungen zu realisieren. Aus diesem Grund scheint das aufgezeigte Verfahren ein kompliziertes, jedoch auch ein Erfolg versprechendes zu sein.

5 Software zur Steuerung des Robosapiens

Das vorangegangene Kapitel befasste sich mit der Entwicklung von Techniken, die es erlauben sollten, den Robosapien über einen PC zu steuern. Wie dort bereits erwähnt, bringt die Software WinLIRC eine kleine Oberfläche mit sich, über die der Anwender konkrete Befehle an den Roboter vom PC aus senden kann.

Um jedoch einen Roboter auch in der Lehre zielführend einsetzen zu können, wird eine weitaus komplexere Art der Ansteuerung benötigt. Einzelne Befehle sollen zu Befehlsgruppen verschmelzen und dem Roboter somit flüssige Bewegung ermöglichen.

5.1 Entwicklung der Software

Die Software basiert auf dem bereits vorgestellten Programm „*transmit.exe*“, welches im Rahmen des LIRC Projektes mitgeliefert wird. Da eine praktische Anwendung nur für den ersten Versuchsaufbau realisiert werden kann, beschränkt sich das Programm auf die serielle Kommunikation über das Programm „*transmit.exe*“.

Hinweis: Um den dritten Versuchsaufbaus (AVR Chip) ansteuern zu können, müsste eine Modifikation der später vorgestellten Funktion `Execute()` erfolgen.

Als Entwicklungsumgebung wurde zunächst das Visual Studio 6 von Microsoft gewählt, deren Programmiersprache auf Visual Basic festgelegt wurde. Diese einfach zu erlernende Sprache bietet neben einem übersichtlichen GUI-Builders auch einfach zu handhabende Schnittstellen zu benötigten Funktionen und Systemkomponenten.

Eine grafisch ansprechende Oberfläche soll es dem Benutzer ermöglichen, den Roboter mit der Mouse zu steuern und Bewegungsabläufe einzuprogrammieren.

Anmerkung: Die Software wurde 2006 auf die Entwicklungsumgebung Visual Studio .NET von Microsoft portiert. Der Quellcode liegt diesem Bericht ebenfalls bei.

Um ein externes Programm ausführen zu können, werden bestimmte Betriebssystemfunktionen benötigt. Die *Dynamic Link Library* shell32.dll bietet auf 32-Bit Systemen die passende Schnittstelle „ShellExecuteA“ und muss in die Software integriert werden.

Unter Verwendung dieser Systemfunktion kann eine ausführbare Datei mit ihren benötigten Parametern ausgeführt werden.

5.2 Implementierung der Software

Visual Basic bietet die Möglichkeit wichtige Funktionen in sog. Modulen einzubetten. Die Datei *modRoboControl.vb* implementiert dabei die Hauptfunktion **Execute()**, sowie die Zusatzfunktionen **ExecuteCmd()** und **ExecuteRecord()**.

```
// Modul modRoboControl.vb

Private Function Execute(ByRef cmd As String) As Object
    ' Führt den übergebenen Befehl aus. Dabei wird das Programm
    ' transmit.exe, welches im Projektordner liegen muss, mit
    ' den übergebenen Parametern ausgeführt

   RetVal = ShellExecute(0, "Open", "transmit.exe",
        REMOTE_CONTROL & " " & cmd & " 0", VB6.GetPath, 1)
End Function
```

Abbildung 29: Funktion zum ausführen des externen Programms "transmit.exe"

Der Aufruf von **ExecuteCmd()** unterscheidet sich dabei nur dann von der Funktion **Execute()**, sobald der Benutzer eine Befehlsfolge einprogrammiert. Ein Befehl wird hierbei jedoch stets unverändert an die Funktion **Execute()** weitergeleitet.

Die zentrale Steuereinheit des Roboters bildet eine GUI, die den Roboter mit all seinen Bewegungsmöglichkeiten präsentiert. Schaltelemente können mit der Mouse betätigt werden, woraufhin das passende IR-codierte Signal über die serielle Schnittstelle an den Roboter weitergeleitet werden. Ein Knopf im unteren linken Eck dient zur Aufnahme von bis zu 1000 Bewegungsabläufen. Diese müssen vom Benutzer in Echtzeit eingegeben werden. Ein erneutes Drücken des Aufnahmeknopfes stoppt den *Record-Modus* und die Bewegungen können über das darunter liegende Schaltelement wiedergegeben werden.

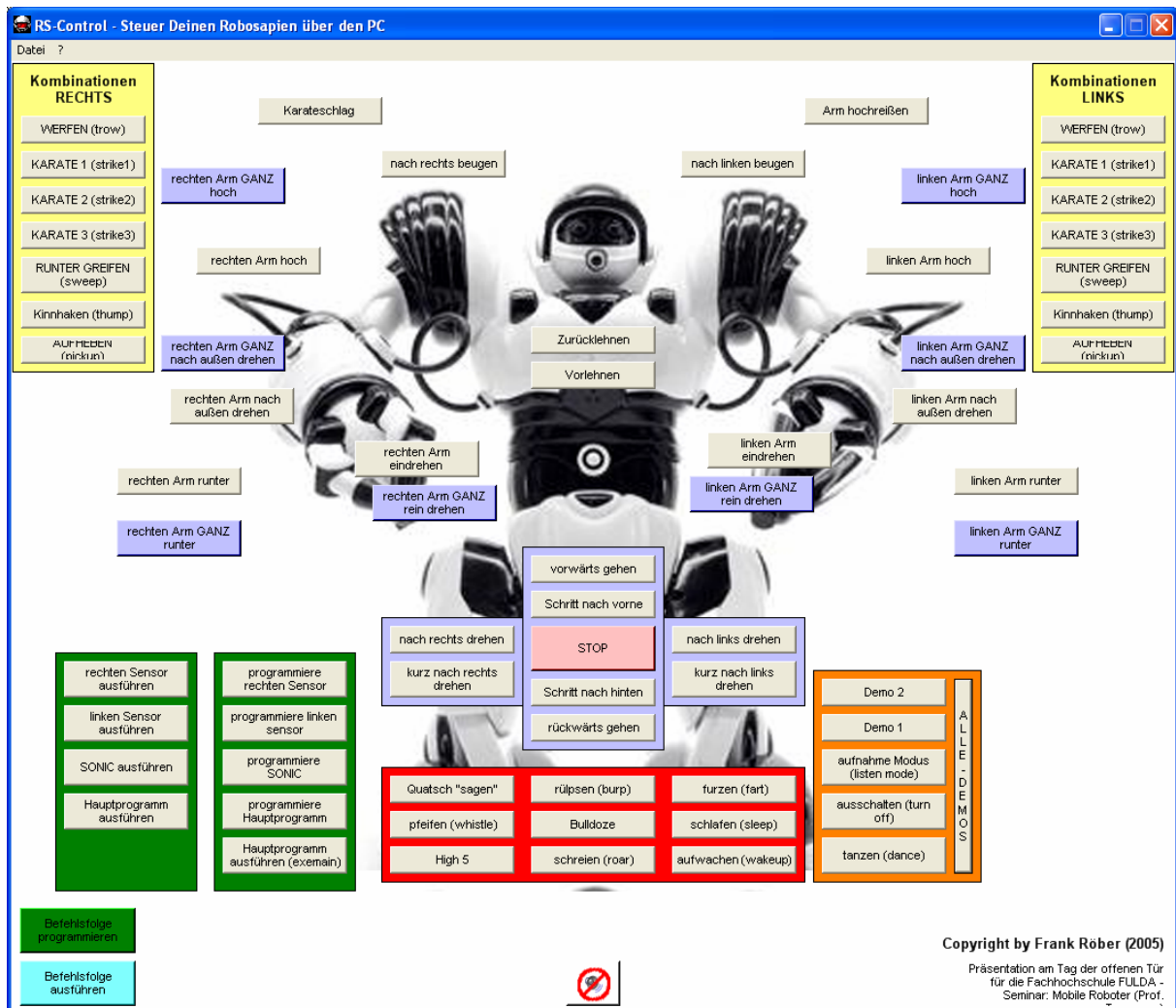


Abbildung 30: Grafische Benutzeroberfläche der RoboControl Software

Die Ausführung eines Befehls erfolgt über Funktionen, die jedem Knopfdruck zugeordnet sind.

```
// Projektdatei RoboControl.vb

Private Sub karateChop_Click(ByVal eventSender As System.Object,
    ByVal eventArgs As System.EventArgs) Handles karateChop.Click

   RetVal = modRS_Control.ExecuteCmd("KARATE_CHOP")
End Sub
```

Abbildung 31: Beispielfunktion zum Ausführen eines spezifischen Kommandos

6 Ergebnisse

Die gesammelten Erkenntnisse ergeben ein geteiltes Bild. Nicht alle Ideen konnten wie gewünscht umgesetzt werden. Zudem traten bei der intensiven Auseinandersetzung mit dem Roboter einige Schwachstellen hervor, die im Folgenden aufgeführt werden:

- ➖ Über keine der aufgeführten Methoden können die Motoren einzeln angesteuert werden.
- ➖ Die Verarbeitung des Roboters ist nur mäßig - so versagte an beiden Testexemplaren nach kurzer Zeit der linke Unterarmmotor.
- ➖ Die Bewegungsfreiheiten des Roboters sind stark eingeschränkt und verbieten den Vergleich mit einer menschenähnlichen Maschine.
- ➖ Die Programmierung über die entwickelte Software ist aufgrund von fehlenden Rückkanälen (Sensoren) am Roboter stark eingeschränkt.

Auf der anderen Seite bringt der Robosapien jedoch auch einige Vorteile mit sich:

- ➕ Er ist kostengünstig zu beziehen.
- ➕ Benötigte Modifikationen zur Steuerung vom PC aus sind mit wenig Aufwand zu bewältigen.
- ➕ Durch seine bloße Erscheinung zieht er die Blicke und das Interesse schnell an sich.
- ➕ Die „Programmierung“ über die entwickelte Software führt schnell zum Erfolg.

6.1 Robosapiens in der Lehre

Der Einsatz der Roboter in der Lehre ist nach Meinung des Autors von eingeschränktem Interesse. Als Instrument zum Erlernen von Bewegungsabläufen eines Roboters eignet sich der hier vorgestellte Robosapiens nicht. Zu unpräzise sind die Motorik der Hände und der Beine. Zudem existieren auf dem Markt in ähnlicher Preiskategorie bereits Systeme, die eine explizite Ansteuerung sämtlicher Aktoren ermöglichen.

6.1.1 Möglichkeiten der Anwendung

Der Robosapiens sorgt sicherlich bei jeder Art von Präsentation, z.B. an Veranstaltungen wie dem „*Tag der offenen Tür*“, für großes Aufsehen. Gerade junge Menschen, darunter auch Mädchen, sind von dem possierlich anmutenden Roboter begeistert und wollen ihn näher kennen lernen. Die vorhandene Verkleidung verdeckt dabei gekonnt das Innenleben des Roboters, sodass Berührungssängste mit solch einer Art von technischem Gerät fast völlig verschwinden.

Um den Roboter auch in der Lehre sinnvoll einsetzen zu können, müssten jedoch noch einige zusätzliche Modifikationen an den hier vorgestellten Lösungen vorgenommen werden. So könnte z.B. das Ersetzen der an den Füßen befindlichen Berührungssensoren durch Ultra-Schallsensoren in Verbindung mit einem zweiten „Gehirn“ dem Roboter eine gewisse Autonomie verleihen. Hierzu müsste der auf dem Roboter befindliche Mikrocontroller durch den *Atmel*[®] AT90S2313 Prozessor erweitert werden. Vorschläge hierzu sind wie beschrieben unter [AIB05] zu finden. Durch diese Modifikation wird es dem Entwickler z.B. erlaubt, die standardmäßig vorgegebenen Aktionen des Roboters bei Aktivierung einer der Berührungssensoren selbst zu bestimmen. Ein auf eine Wand zulaufender Roboter könnte somit noch vor der Kollision mit der selben seinen Körper drehen und seine Bewegung in eine andere Richtung fortsetzen.

6.1.2 Aufwand der Modifikationen

Für einen Studenten der Informatik sollte der Aufwand für die benötigten Modifikationen, um den Roboter von einem PC aus zu steuern, leicht zu bewältigen sein. Die hier aufgezeigten, zusätzlich benötigten Komponenten sind selbst ohne Erfahrung mit dem Lötkolben mit vertretbarem Aufwand zu realisieren. Einzig die sekundären Modifikationen, wie z.B. das Programmieren des *Atmel*[®] Mikrocontrollers können u.U. sehr Zeit zehrend sein.

6.1.3 Nutzen für die Studierenden

Durch die Auseinandersetzung mit sämtlichen Hardwarekomponenten wird dem/der Studierenden auf einfache Weise ein gewisses technisches und mechanisches Verständnis abverlangt. Der Umgang mit elektronischen Bauteilen ergänzt die standardmäßig vermittelten Lehrinhalte, sodass die Studenten Ihre Fähigkeiten auch in diesem Bereich verstärkt unter Beweis stellen können. Das mit der Informatik untrennbar verbundene Wissen über die Elektrotechnik richtet sich dabei jedoch in erster Linie an diejenigen Hochschulüler, die nach ihrem Studium in solchen Bereichen weiter arbeiten wollen.

6.2 Fazit und Empfehlungen

Ohne eine weiterführende Arbeit wird der Einsatz des Robosapiens in der Lehre nicht durchgeführt werden können. Die Probleme der beiden ersten aufgezeigten Modifikationsmöglichkeiten sind nur mit erheblichem Aufwand zu beseitigen.

Zielführend erscheint hier nur die Erweiterung durch einen zweiten Mikrocontroller (AVR Prozessor). Sollte es gelingen eine stabile Konfiguration zu erzeugen, steht dem Einsatz von Funkmodulen, wie einer Bluetooth-Verbindung, nichts im Wege. Hierdurch können einzelne Robosapiens getrennt voneinander programmiert und angesteuert werden. Die vom Autor dieser Arbeit entwickelte Software „*RoboControl*“ kommt dabei prinzipiell mit jeder Art von Modifikation zurecht. Um jedoch interessante Funktionen wie das freie Programmieren der Sensoren verwenden zu können, ist die Änderung der auf dem AVR-Chip befindlichen Firmware von Nöten. Die Software „*RoboControl*“ ist hierzu nicht in der Lage, da sie ausschließlich die Ansteuerung der Aktoren übernimmt. Eine dynamische Programmierung der Sensoren ist somit nicht möglich.

Jedoch steht diesen Möglichkeiten nach wie vor die eingeschränkte Motorik der Roboter gegenüber. Deshalb sollten der Einsatz der hier vorgestellten Robosapiens in der Lehre höchstens für die Steigerung des technischen Verständnisses in Betracht gezogen werden. Für einen umfassenden Einsatz in einem Seminar ist die Hardware zu fehleranfällig und die Modifikationen noch zu fehlerträchtig.

Das Verständnis von Hard- und Software sollte jedoch in einem Informatikstudium anhand von Projekten dieser Art weiter gefördert werden.

Quellenverzeichnis

- [AIB05] RoboSapienPet**
AiboHack Project – Robosapien AVR modification
<http://www.aibohack.com/robosap/index.html>
- [BAR00] Bartelmus, Christoph**
Ein Pinguin sieht (infra)rot
In: Zeitschrift für Computertechnik (BAR), Sept. 18/2000
<http://www.heise.de/ct/00/18/208/>
- [BRO05] Brown, Chance**
Robosapien – IR Control
Toronto, Ontario Canada 2003
<http://www.geocities.com/chancebrown2003/ircontrol.html>
- [BUE05] Buehl, Eric**
Robosapien-LIRC Interface
<http://www.andrew.cmu.edu/user/ebuehl/robosapien-lirc/>
- [CRA04] Craig, Mark**
Robosapien Project
Schottland, 2004
<http://personal.strath.ac.uk/mark.craig/robot/robos.shtml>
- [FUT05] Futurlec**
Futurelec Homepage –AVR Development Boards
<http://www.futurlec.com/ATDevBoard.shtml>
- [LIR06] Linux Infrared Remote Control Project**
Hauptseite
<http://www.lirc.org/>
- [LIT03] Linux Infrared Remote Control Project**
Leistungsstarker IR-Transmitter
http://www.lirc.org/improved_transmitter.html
- [LST03] Linux Infrared Remote Control Project**
Einfacher IR-Transmitter
<http://www.lirc.org/transmitters.html>
- [MEI06] Meier, Jan Hendrik**
Einfache AVR Programmierschaltung für den parallelen Port
http://www.meierspage.net/how2s/elektronik/at90s2313_mikrocontroller_programmieren/at90s2313_mikrocontroller_programmieren_01.php
- [ROB05] Robosapiens**
Offizielle Homepage des Herstellers WooWee
<http://www.robosapienonline.com/>

- [ROE05] Röber, Frank**
Autor dieses Berichtes
Ravensburg 2005/2006
Bitte durchsuchen Sie die, in Zusammenhang dieser Arbeit beigefügten Dokumente/CD
- [WAV06] WinAVR**
Homepage der AVR Entwicklungsumgebung und des Compilers
<http://winavr.sourceforge.net/>
- [WLI01] Windows (Linux) Infrared Remote Control Project**
Hauptseite
<http://winlirc.sourceforge.net/de/>

Anhang

Schaltpläne der elektronischen Bauteile des Robosapiens [CRA04]

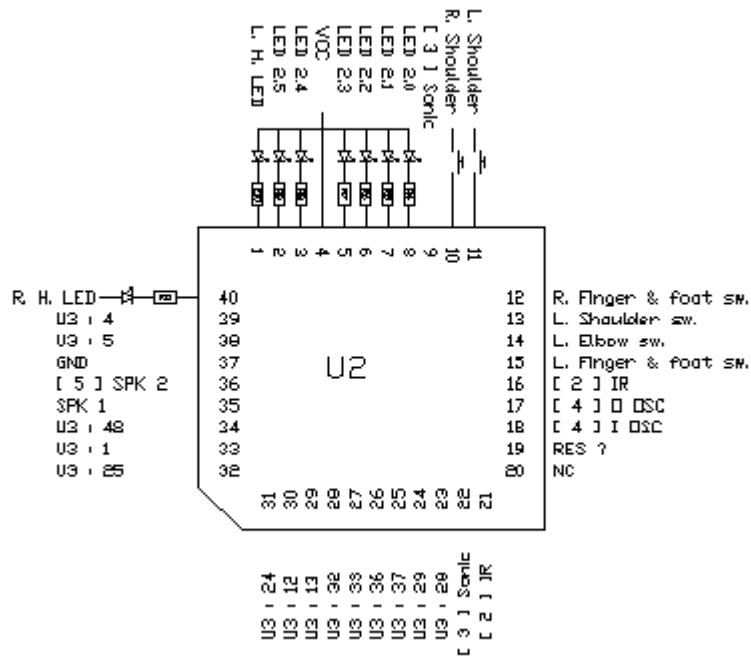


Abbildung 32: Schaltpläne Mikrocontroller

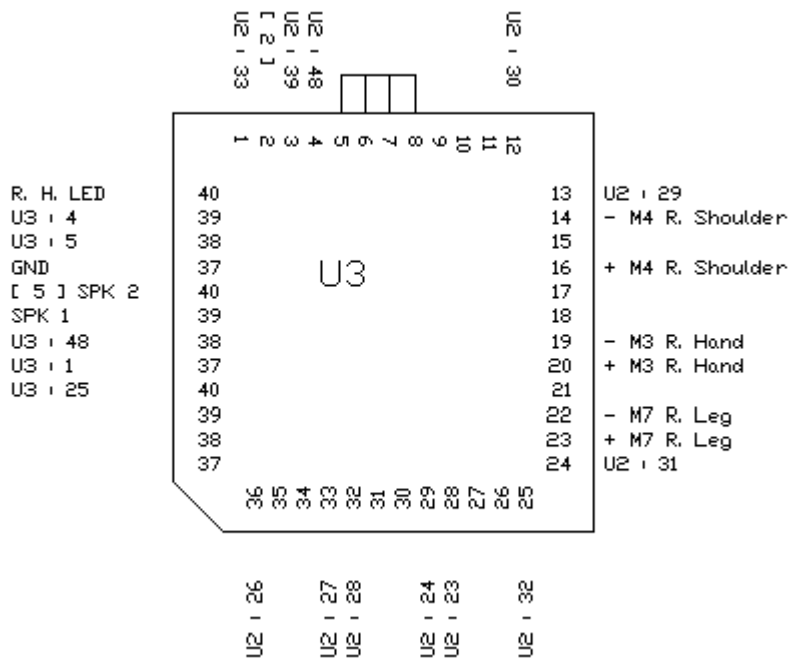


Abbildung 33: Schaltpläne Motortreiber

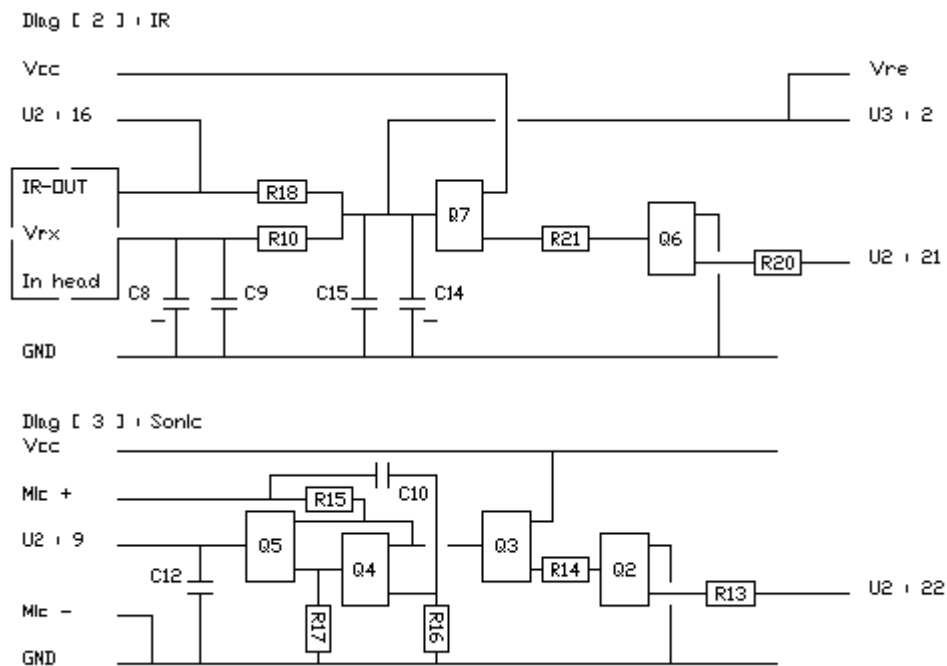


Abbildung 34: Schaltdiagramm Infrarot- und Schallsensor

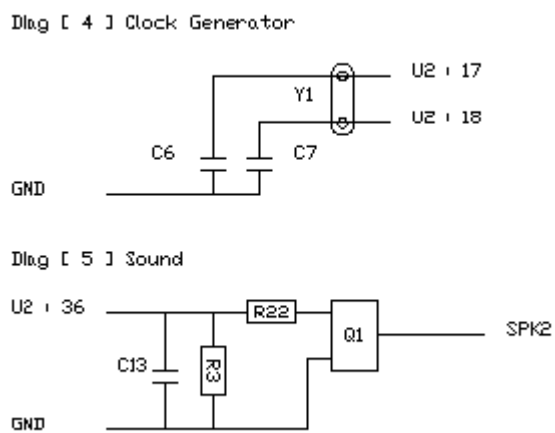


Abbildung 35: Schaltdiagramm Taktgeber und Geräuschverstärker

Optische Signaldeutung des Robosapiens

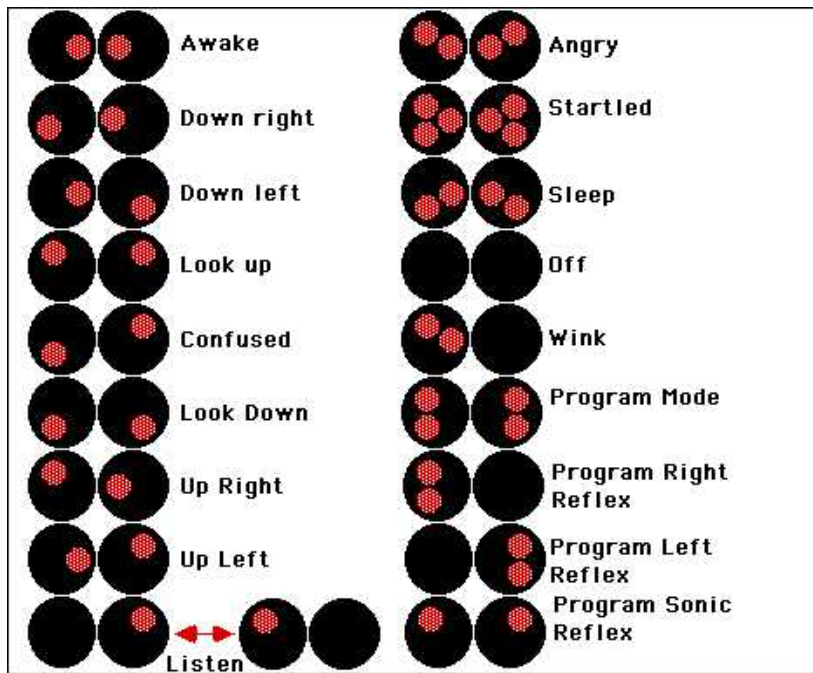


Abbildung 36: Leuchtzustände der LEDs im Kopf des Robosapiens [BRO05]

Bauteile (Standard)

Bauteilbezeichnung	Markings	Schalt
Q1	TD	[5]
Q2, Q6	S 1A 35	[2] and [3] respectively
Q3, Q4, Q5, Q7	S 2A 30	[3], [3], [3], [2]
R1	510	[6]
R2	4R7	[6]
R3	195	[5]
R4-R11	621	[1a]
R12	473	[1a] U2:19 -R12- Vcc
R13	472	[3]
R14	103	[3]
R15	472	[3]
R16	475	[3]
R17	224	[3]
R18	223	[2]
R19	470	[2]
R20	472	[2]
R21	103	[2]
R22	0	[5]
R23	510	[6]
R24	103	[1a] U2:13 -R24- Gnd
R25	103	[1a] U2:12 -R25- Gnd
C1	100uF electrolytic	[6]
C2	100uF electrolytic	[6]
C8	100uF electrolytic	[2]
C11	47uF electrolytic	[3]
C14	10uF electrolytic	[2]
C16	100uF electrolytic	[6]
Y1	Z 4.0 M	[4]

Anschlussbelegung des AVR Mikrocontrollers

RoboSapienPet #1 – SapAVR Mod

© 2004 RoboSapienPet

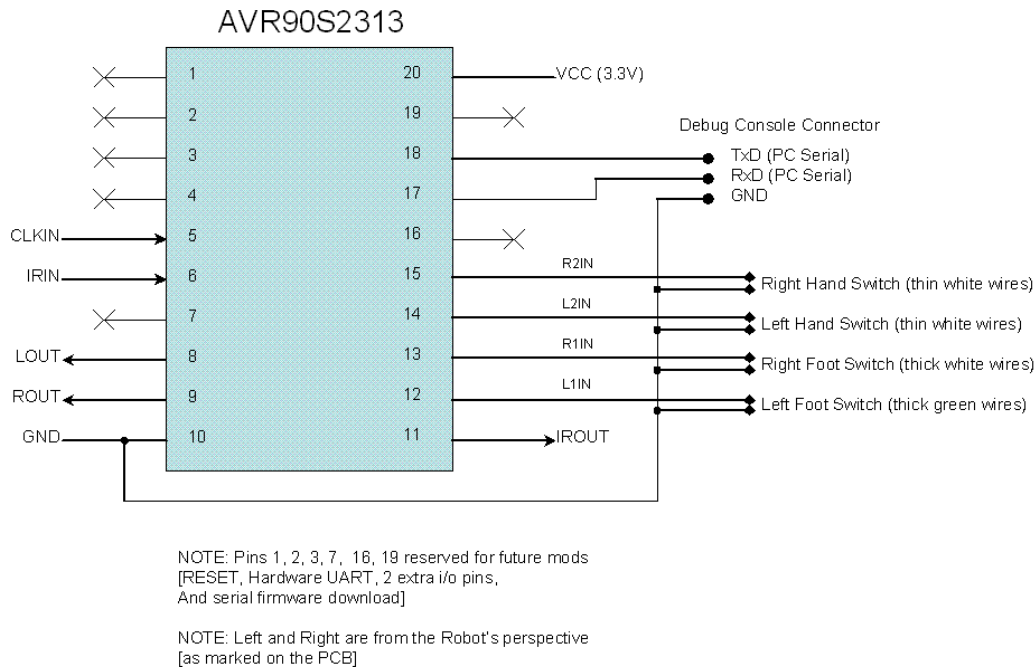


Abbildung 37: Verbindungsschema des AVR Mikrocontrollers mit dem Robosapien [AIB05]

RS-232 level shifter (3 wire version)

© 2004 RoboSapienPet

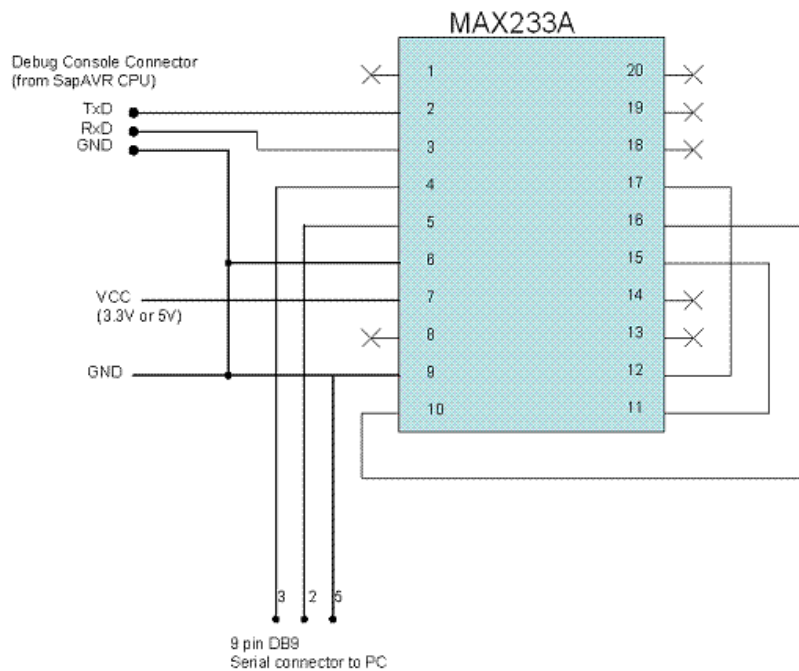


Abbildung 38: Verbindungsschema der Debug-Konsol mit dem Robosapien [AIB05]

LIRC Konfigurationsdatei

```

#
# contributed by Eric Buehl (2005)
# modified          by Frank Roeber (2005)
#
# brand:            Robosapien
# model:
# supported devices:
#

begin remote

    name    Robosapien
    bits    8
    eps     500
    aeps    400

    header  6666 3333
    one     833 3333
    zero    833 833
    gap     0
    toggle_bit 0

    frequency 39200

# transmitter 0x05

    begin codes
        TURN_RIGHT          0x0000000000000000
        RIGHT_ARM_UP        0x0000000000000002
        RIGHT_ARM_OUT        0x0000000000000004
        TILT_BODY_RIGHT      0x0000000000000006
        RIGHT_ARM_DOWN       0x0000000000000008
        RIGHT_ARM_IN         0x000000000000000A
        WALK_FORWARD         0x000000000000000C
        WALK_BACKWARD        0x000000000000000E
        TURN_LEFT            0x0000000000000010
        LEFT_ARM_UP          0x0000000000000012
        LEFT_ARM_OUT         0x0000000000000014
        TILT_BODY_LEFT       0x0000000000000016
        LEFT_ARM_DOWN        0x0000000000000018
        LEFT_ARM_IN          0x000000000000001A
        STOP                  0x000000000000001C
        MASTER_COMMAND        0x0000000000000020
        PROGRAM_PLAY          0x0000000000000022
        RIGHT_PROGRAM         0x0000000000000024
        LEFT_PROGRAM          0x0000000000000026
        SONIC_PROGRAM         0x0000000000000028
        RIGHT_STEP            0x0000000000000040
        RIGHT_THUMP           0x0000000000000042

```

```
RIGHT_THROW          0x0000000000000044
SLEEP                0x0000000000000046
RIGHT_PICKUP         0x0000000000000048
LEAN_BACKWARD       0x000000000000004A
FORWARD_STEP        0x000000000000004C
BACKWARD_STEP       0x000000000000004E
LEFT_STEP           0x0000000000000050
LEFT_THUMP          0x0000000000000052
LEFT_THROW          0x0000000000000054
LISTEN              0x0000000000000056
LEFT_PICKUP         0x0000000000000058
LEAN_FORWARD        0x000000000000005A
RESET               0x000000000000005C
MASTER_EXECUTE     0x0000000000000060
WAKEUP              0x0000000000000062
RIGHT_EXECUTE       0x0000000000000064
LEFT_EXECUTE        0x0000000000000066
SONIC_EXECUTE       0x0000000000000068
RIGHT_STRIKE3       0x0000000000000080
RIGHT_SWEEP         0x0000000000000082
BURP                0x0000000000000084
RIGHT_STRIKE2       0x0000000000000086
HIGH5               0x0000000000000088
RIGHT_STRIKE1       0x000000000000008A
BULLDOZER          0x000000000000008C
FART                0x000000000000008E
LEFT_STRIKE3        0x0000000000000090
LEFT_SWEEP          0x0000000000000092
WHISTLE            0x0000000000000094
LEFT_STRIKE2        0x0000000000000096
TALKBACK           0x0000000000000098
LEFT_STRIKE1        0x000000000000009A
ROAR                0x000000000000009C
DEMO_ALL            0x00000000000000A0
POWER_OFF           0x00000000000000A2
DEMO1               0x00000000000000A4
DEMO2               0x00000000000000A6
DANCE               0x00000000000000A8
QUIET_EXECUTE       0x0000000000000030
FEET_SHUFFLE        0x00000000000000EC
NOTHING             0x00000000000000F6
RAISE_ARM           0x00000000000000F8
KARATE_CHOP         0x00000000000000AC
end codes

end remote
```

Sämtliche Quellcodes und verwendete Programme

Siehe CD