

**VORLÄUFIGE ARBEITSKOPIE!**

**ÜBERSETZEN VON**

**SCHRITTMOTORPROTOKOLLEN**

**Entwurf eines Hardwareübersetzers**

**Praxisbericht**

im Fachgebiet Mess- und Sensortechnik



vorgelegt von: Johannes Dielmann  
Studienbereich: Technik  
Matrikelnummer: 515956  
Erstgutachter: Prof. Dr. Carstens-Behrens

© 2012

Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>III</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Ziel der Arbeit . . . . .	1
1.3. Aufbau der Arbeit . . . . .	2
<b>2. Hardware</b>	<b>3</b>
2.1. Lasererfassungssystem VI-900 . . . . .	3
2.2. Ansteuerung für den Drehtisch . . . . .	4
2.2.1. Drehtisch . . . . .	4
2.2.2. Spannungsversorgung . . . . .	4
2.2.3. Schrittmotoren . . . . .	4
2.2.4. Schrittmotorkarten . . . . .	4
2.2.5. Motorverkabelung . . . . .	5
2.2.6. Endschalter . . . . .	5
2.3. Mikrocontroller . . . . .	6
2.3.1. Entwicklerboard STK500 . . . . .	6
2.3.2. AVRISP mkII . . . . .	7
2.3.3. MAX232 . . . . .	7
2.4. Platinenlayout . . . . .	8
<b>3. Software</b>	<b>9</b>
3.1. Entwicklungsumgebung . . . . .	9
3.2. RapidForm2004 . . . . .	9
3.3. Watchdog . . . . .	10
3.4. Einfaches Beispiel für LEDs und Taster . . . . .	10
3.5. LCD Bibliothek . . . . .	10
3.6. RS-232 . . . . .	10
3.7. Menü Bibliothek . . . . .	10

3.8. Interrupts . . . . .	10
3.8.1. Endschalter . . . . .	10
3.8.2. Watchdog . . . . .	10
3.9. Protokoll der Stepperkarte . . . . .	10
3.10. Manueller Betrieb . . . . .	10
3.11. Protokolle aus RapidForm . . . . .	10
3.12. Übersetungs Logik . . . . .	10
3.13. Automatische Protokollwahl . . . . .	10
<b>4. Fazit und Zukunft</b>	<b>11</b>
4.1. Fazit . . . . .	11
<b>Eidesstattliche Erklärung</b>	<b>13</b>
<b>A. Anhang</b>	<b>i</b>
A.1. Schritt für Schritt Anleitung . . . . .	ii
A.2. ASCII Befehlssatz R+S Schrittmotorsteuerung . . . . .	iii
A.3. Vom Autor verwendete Software . . . . .	iv

## Abkürzungsverzeichnis

ADC .....	analog digital convertor
ASCII .....	American Standard Code for Information Interchange
AVRISP .....	AVR in system programmer
CAD .....	(engl. computer-aided design) Computer gestützter Entwurf
CF .....	Compact Flash
CPU .....	central processing unit
DAC .....	digital analog convertor
DIL .....	dual in line package
IC .....	integrated cuircuit
LCD .....	(engl. liquid crystal display) Flüssigkristall Display
MHz .....	megahertz
RS .....	Radio Sector
SCSI .....	Small Computer System Interface
USB .....	universal serial bus
V .....	Volt

# Abbildungsverzeichnis

2.1. Überblick des Arbeitsplatz . . . . .	3
2.2. Block Diagram eines Mikrocontroller . . . . .	7

# Tabellenverzeichnis

A.1. ASCII Befehlssatz R+S Schrittmotorsteuerung . . . . .	iii
--	-----

# Codeverzeichnis

3.1. Funktion - Laufflicht . . . . .	9
--------------------------------------	---

# 1. Einleitung

**(TODO: DIE EINLEITUNG IST SEHR WICHTIG!!!) (TODO: AUSBAUEN? NEU SCHREIBEN!)** Die 3D-Lasererfassung bietet zahlreiche Anwendungsgebiete. Von der Erfassung kleiner Objekte über die Erkennung von

**(TODO: KLARSTELLEN DER BEGRIFFLICHKEITEN)** In der CAD-Entwicklung kommt es vor das für ein real existierendes Objekt eine Erweiterung konstruiert werden muss. Um die Erweiterung sinnvoll konstruieren zu können müssen dazu die Abmessungen des Objektes möglichst genau bekannt sein. Das übertragen der Abmessungen, kann insbesondere für komplexe Objekte, sehr aufwendig sein. Abhilfe soll ein Laserscanner schaffen der das Objekt aus mehreren Richtungen vermisst und aus diesen Informationen ein genaues 3D-Modell davon generiert.

## 1.1. Motivation

Im Projekt soll nun mit einer Kombination aus einem Lasererfassungssystem, einem Drehtisch und der dazugehörigen Software auf einfachem Wege ein 3D-Modell erfasst werden. Dieses soll dann in einer CAD-Software wie *Solidworks* nutzbar sein.

## 1.2. Ziel der Arbeit

Die Kommunikation zwischen Software und Drehtisch soll ermöglicht werden. Dazu werden die Befehle der Software mit einem Mikrocontroller ausgewertet und in für den Drehtisch verständliche Befehle übersetzt.

Um den Drehtisch manuell bedienen zu können und den aktuellen Status des Drehtisch anzuzeigen sind noch ein LC-Display und mehrere Taster vorgesehen.

Die Ansteuerung des Drehtisches ist als Einschub für ein 19Rack realisiert. Daher wird die Platine für den Mikrocontroller auch als 19Einschub realisiert.



### 1.3. Aufbau der Arbeit

Der Aufbau der Arbeit gliedert sich im Wesentlichen in die Entwicklung neuer und die Nutzung vorhandener Hardware, sowie in die Entwicklung der Software für den Mikrocontroller.

Zur Hardware gehören die Auswahl des Mikrocontroller, die Endschalter, die Schrittmotorkarten, die Schrittmotoren, sowie die verwendeten PCs.

Zur Software gehören die Entwicklungsumgebungen und die 3D-Erfassungssoftware.

**(TODO: ÜBERBLICK VERSCHAFFEN. KOMMUNIKATION MIT SCHRITTMOTOR-KARTE VON PC AUS. AUFBAUEN DES STK500. EINARBEITEN IN UC ENTWICKLUNG. STEUERN DER SCHRITTMOTOR KARTE VOM UC AUS. ERARBEITEN DER PROTOKOLLE (REVERSE ENGINEERING). C-PROGRAMM ZUM VERSTEHEN EINGEHENDER BEFEHLE. ÜBERSETZEN DER BEFEHLE. NEUER MIKROCONTROLLER. UMGEBUNGSWECHSEL. LC-DISPLAY. ENDSCHALTER. MAX232. PLATINENLAYOUT.)**

## 2. Hardware

(TODO: BESSERES BILD! SCHEMA? BESSERE BESCHRIFTUNG!)



Abbildung 2.1.: Überblick des Arbeitsplatz

(TODO: FUSSNOTEN FÜR KOMPONENTEN, HERST. U. WEBSITES)  
(TODO: KOMPONENTEN AUF ABILDUNG ERWÄHNEN!)

### 2.1. Lasererfassungssystem VI-900

(TODO: BILD) Das Lasererfassungssystem *VI-900* der Firma *Minolta* besteht aus einem *Lasertriangulator* und einer Kamera. Das System lässt sich über eine *SCSI*-Schnittstelle ansprechen und konfigurieren. Zur mobilen Nutzung kann das Gerät

auch auf der Rückseite bedient werden. Aufgenommene Daten können auf einer *CF-Karte* gespeichert werden. Im Projekt wurde jedoch lediglich die Ansteuerung via SCSI genutzt.

## 2.2. Ansteuerung für den Drehtisch

### 2.2.1. Drehtisch

Der Drehtisch ist eine Eigenkonstruktion der Werkstatt des RheinAhrCampus. Er besteht aus einer massiven Edelstahl Arbeitsplatte, welche auf 4 Füßen ruht. Aus dieser ist ein Rechteck mit aufgesetztem Halbkreis ausgeschnitten. In diesem Ausschnitt befindet sich der Drehtisch. Er ist auf einem Schienensystem gelagert. Mit dem Schienensystem lässt der Drehtisch sich in der Vertikalen positionieren. Mit einem Schrittmotor lässt sich der Drehtisch sich zusätzlich in der Höhe verstellen. Die Höhenverstellung wird mit einem Schneckengetriebe realisiert. Ein weiterer Schrittmotor ist für die Drehung des Tisches zuständig. Der Tisch ist über ein *Harmonic-Drive-Getriebe* mit dem Schrittmotor verbunden. Das Übersetzungsverhältnis beträgt 1:50.

### 2.2.2. Spannungsversorgung

**(TODO: VERKABELUNG STECKBAR UND UNIVERSELL GEMACHT)** Die Schrittmotorkarten werden von einem PC-Netzteil gespeist. Die Kabel waren direkt an die Verbindungsleisten gelötet. Um den Aufbau modular und erweiterbar zu machen, ersetzte ich die feste Lötverbindung durch eine Standard PC-Netzteil Verbindung. Dadurch kann das Netzteil einfach ausgebaut werden, bzw. das System leicht mit neuen Einschubkarten erweitert werden.

### 2.2.3. Schrittmotoren

**(TODO: MOTOREN BESCHREIBEN! TECHNISCHE DATEN! SCHRITTE, SPANNUNGEN. VERDRAHTUNG.)**

### 2.2.4. Schrittmotorkarten

Die Ansteuerung für die Schrittmotoren sind als 19Einschübe realisiert. Für jeden Schrittmotor wird ein Einschub benötigt. Die Einschübe sind Produkte der Firma R+S. Mittels RS-232 Schnittstelle lassen sich die Karten konfigurieren und ansteuern.

Die Konfiguration und Ansteuerung erfolgt über einen vorgegeben *ASCII* Befehlssatz. Der Befehlssatz befindet sich im Kapitel A.2. Es können 2 oder mehr Karten als *Daisy-Chain*<sup>1</sup> in Reihe geschaltet werden.

### 2.2.5. Motorverkabelung

Die Schrittmotoren benötigen ein mindestens 4-adriges Kabel. Das Kabel für den Schrittmotor der für die Rotation zuständig ist war bereits gefertigt. Das Kabel für den Schrittmotor der für die Höhenverstellung zuständig ist habe ich selbst gefertigt. Hier wurden 3 weitere Adern für die beiden Endschalter benötigt. **(TODO: SCHEMAZEICHNUNG KABEL!)**

### 2.2.6. Endschalter

Die Schrittmotorkarten unterstützen das Abschalten der Motoren wenn ein sogenannter Endschalter ausgelöst wird. Dies sind im allgemeinen mechanische Schalter die ausgelöst werden wenn der Tisch sich dem Ende des Arbeitsbereiches nähert. Dies verhindert eine Beschädigung des Aufbaus.

Im Aufbau waren bereits induktive Endschalter der Firma Pepperl+Fuchs verbaut. Normalerweise unterstützt die Schrittmotorkarte nur mechanische Endschalter. Durch geschickte Verdrahtung ließen sich die induktiven Endschalter verwenden. Hierzu musste über einen Spannungsteiler die Spannung herabgesetzt werden und konnte somit direkt an den Optokoppler der Schrittmotorkarte angeschlossen werden. **(TODO: SCHEMAZEICHNUNG DER VERDRAHTUNG)**

Am Drehtisch war ein Metallstutzen angebracht der den Endschalter auslösen sollte. Dieser war jedoch ungeeignet da er nicht dicht genug an den Induktiven Schalter heran kam, obwohl der Tisch schon in der Endposition war.

Abhilfe schaffte ein längerer Metallstutzen der von der Werkstatt gefertigt wurde. Wenn der Tisch sich in der Endposition befindet soll dies auch auf dem Mikrocontroller angezeigt werden. Die Signale der Endschalter liegen auf der Rückseite **(TODO: ZEICHNUNG DER ANSCHLÜSSE REFERENZIEREN.)** am Verbindungsstecker an. Es muss also nur eine Brücke zu den entsprechenden Pins des Verbindungsstecker des Mikrocontroller gelötet werden.

Auf der Mikrocontroller Platine sind diese Pins mit 2 Pins des Mikrocontroller verbunden. Die beiden Pins werden im Mikrocontroller als Interrupts definiert. Die

---

<sup>1</sup>Als Daisy Chain (englisch, wörtlich „Gänseblümchenkette“) bezeichnet man eine Anzahl von Hardware-Komponenten, welche in Serie miteinander verbunden sind (meist in sogenannten Bussystemen in der Automatisierungstechnik). [Wikipedia \[2012a\]](#)

Interrupt-Service-Routine ist im Kapitel **(TODO: SOFTWARE KAPITEL REFERENZIEREN)** beschrieben.

## 2.3. Mikrocontroller

Ein Mikrocontroller vereint, in einem IC, die wichtigsten Komponenten um komplexe technische Probleme leicht lösen zu können. Dazu gehören z.B. CPU, Flash-Speicher, Arbeitsspeicher, Register, Ports, ADC, DAC und mehr. Einen schematischen Überblick über die Komponenten eines Mikrocontroller bietet das Blockdiagramm in Abbildung 2.2.

In einer Programmierungsumgebung lässt sich dann für den Mikrocontroller ein Programm schreiben. Diese Programme können Signale an Pins des Mikrocontroller auswerten und Signale über andere Pins ausgeben. Eingehende Signale können binär ausgewertet werden oder mit einem ADC die Spannungshöhe bestimmt werden. Ausgehende Signale können auch binär oder mit einem DAC analog ausgegeben werden. Binäre Signale können zur Steuerung von LEDs oder Peripherie Geräten genutzt werden. Auch LC-Displays und Serielle Schnittstellen können so angesteuert werden. Für unterschiedliche Aufgaben sind verschiedene Mikrocontroller geeignet. Zu Beginn stand ein ATmega 8515 [Atmel \[2012b\]](#) im DIL-Gehäuse zur Verfügung. Dieser hatte 8 Kbyte Flash, 3 externe Interrupts, 1 Serielle Schnittstelle und konnte mit bis zu 16 MHz betrieben werden. Dieser war geeignet sich in die Programmierung mit C ein zu finden und eine Serielle Schnittstelle an zu steuern.

Für dieses Projekt sind jedoch 2 externe Schnittstellen nötig. Der ATmega 324A erfüllt diese Voraussetzung. [Atmel \[2012a\]](#) Er ist dem ATmega 8515 recht ähnlich, bietet jedoch die benötigten 2 seriellen Schnittstellen. Des weiteren hat er 32 Kbyte Flash. **(TODO: MEHR SCHREIBEN??)**

### 2.3.1. Entwicklerboard STK500

Um Mikrocontroller zu programmieren und die Programmierung zu überprüfen kann das *Entwicklerboard* STK500 der Firma ATMEL verwendet werden. Das Board enthält mehrere Mikrocontroller Steckplätze, 2 Serielle Schnittstellen, 8 Taster, 8 LEDs, 2 Erweiterungsports, ein *ISP* **(TODO: BESSERER NAME!)** und mehrere Jumper zum konfigurieren des Boards.

Von den beiden seriellen Schnittstellen kann die eine zur Programmierung des Mikrocontroller verwendet werden. Die andere kann zur Kommunikation mit dem Mikrocontroller genutzt werden.

## 2. Hardware

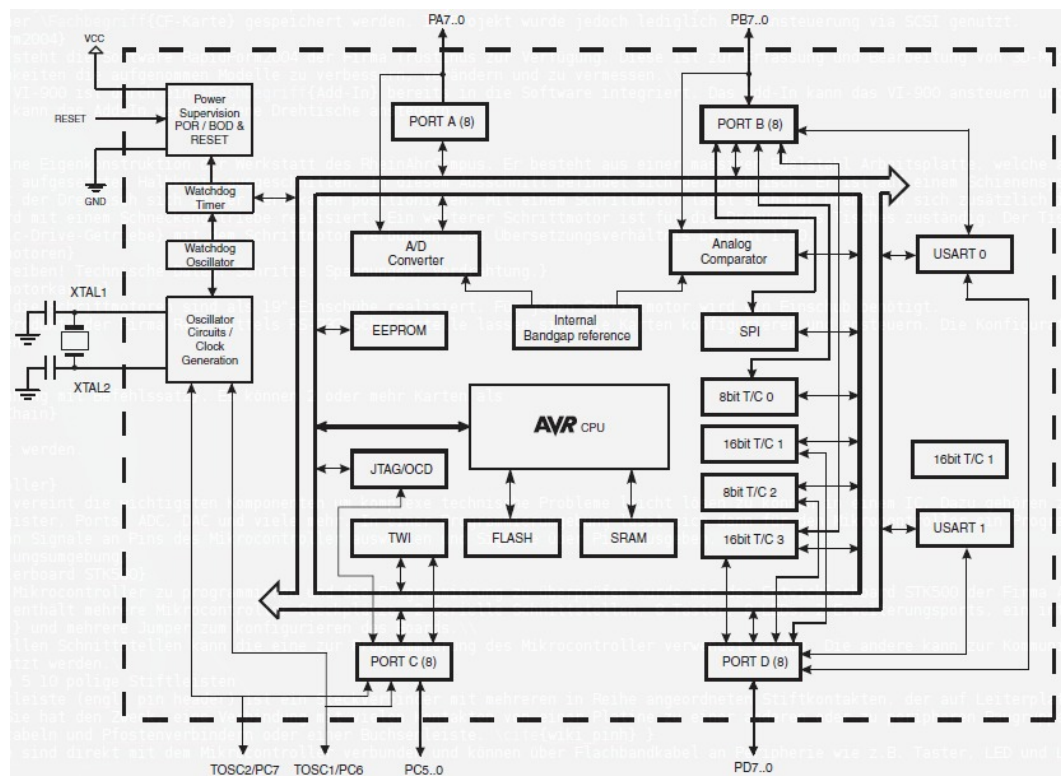


Abbildung 2.2.: Block Diagram eines Mikrocontroller  
[Atm 2011]

Auf dem Board stehen 5 10 polige Stiftleisten zur Verfügung. Diese sind direkt mit dem Mikrocontroller verbunden und können über Flachbandkabel an Peripherie wie z.B. Taster, LED und LC-Displays angeschlossen werden.

### 2.3.2. AVRISP mkII

Das AVRISP mkII ist ein USB-basiertes In-System-Programmiersystem. Dieses kann anstelle des RS-232 basierten Programmiersystem des STK500 verwendet werden. Die Übertragungsgeschwindigkeit des AVRISP mkII ist wesentlich höher als die über die Serielle Verbindung. Desweiteren wurde der ATmega324A nicht mehr vom STK500 internen ISP unterstützt.

Der AVRISP mkII lässt sich einfach an den Programmierport, eine 6-Polige Stiftleiste, des STK500 anschließen.

### 2.3.3. MAX232

Die Spannungspegel des Mikrocontroller (typ. 0-5 V) sind nicht kompatibel zu den Spannungspegeln des RS-232 Standards (typ. -12-+12 V). Daher wird der *Pege-lumsetzer* MAX232 genutzt. Dieser wandelt mit internen Operationsverstärkern die Spannungspegel auf den richtigen Wert. **(TODO: BESCHALTUNG?)**

## 2.4. Platinenlayout

Für den Mikrocontroller und seine Peripherie wurde ein Platinenlayout entwickelt. Dieses wurde in der Opensource Software KiCad entwickelt.

Dazu wurden die Schaltungen wie auf dem STK500 in den Schaltplan übernommen und dort das Layout entwickelt. **(TODO: SCHALTPLAN UND LAYOUT BILD EINBINDEN.)**

## 3. Software

(TODO: EINFÜHRUNG SCHREIBEN) (TODO: WEITERE SOFTWARE IN BEGRIFFEN ERKLÄREN. MINOLTA)

### 3.1. Entwicklungsumgebung

Als Entwicklungsumgebung wird eine Software bezeichnet die es dem Anwender erleichtert Programme für den Mikrocontroller zu schreiben. Im allgemeinen bestehen Entwicklungsumgebungen aus einem Editor, dem Compiler und einer Programmiersoftware. Der Editor bietet dabei meist Komfortfunktionen wie Syntaxhighlighting, Autovervollständigung und Projektmanagement. (TODO: BESSER SCHREIBEN!) (TODO: AVR STUDIO ECLIPSE BUG DEFECTS BIBLIO?)

Listing 3.1: Funktion - Lauflicht

```
1
2 void led_laufflicht (void) {
3     uint8_t i = LED_PORT;
4     i = (i & 0x07) | ((i << 1) & 0xF0);
5     if (i < 0xF0)
6         i |= 0x08;
7     LED_PORT = i;
8 }
```

### 3.2. RapidForm2004

Zur Erfassung am PC steht die Software RapidForm2004 der Firma TrustInus zur Verfügung. Diese ist zur Erfassung und Bearbeitung von 3D-Modellen gedacht. Sie bietet umfangreiche Möglichkeiten die aufgenommen Modelle zu verbessern, verändern und zu vermessen.

Die Ansteuerung des VI-900 ist durch ein *Add-In* bereits in die Software integriert. Das Add-In kann das VI-900 ansteuern und die Aufgenommenen Daten auslesen. Weiterhin kann das Add-In verschiedene Drehtische ansteuern.



### 3.3. Watchdog

(TODO: FUSES ERKLÄREN!)

### 3.4. Einfaches Beispiel für LEDs und Taster

### 3.5. LCD Bibliothek

### 3.6. RS-232

### 3.7. Menü Bibliothek

### 3.8. Interrupts

(TODO: IN DEN BEGRIFFEN ERKLÄREN! VERLINKEN!)

#### 3.8.1. Endschalter

#### 3.8.2. Watchdog

### 3.9. Protokoll der Stepperkarte

### 3.10. Manueller Betrieb

### 3.11. Protokolle aus RapidForm

### 3.12. Übersetungs Logik

### 3.13. Automatische Protokollwahl

## 4. Fazit und Zukunft

### 4.1. Fazit

**(TODO: FAZIT SCHREIBEN!)**

## Literaturverzeichnis

### Atm 2011

ATMEL (Hrsg.): *ATmega164A/PA/324A/PA/644A/PA/1284/P Complete*. San Jose, CA 95131, USA: Atmel, 06 2011 **2.2**

### Atmel 2012a

ATMEL: *ATmega324A- Atmel Corporation*. <http://www.atmel.com/devices/ATMEGA324A.aspx>. Version: 2012. – [Online; Stand 11. Februar 2012] **2.3**

### Atmel 2012b

ATMEL: *ATmega8515- Atmel Corporation*. <http://www.atmel.com/devices/ATMEGA8515.aspx>. Version: 2012. – [Online; Stand 11. Februar 2012] **2.3**

### V9141 2001

RS (Hrsg.): *Schrittmotor-Platine mit integriertem Treiber*. Mörfelden-Walldorf: RS, 03 2001 **A.1**

### Wikipedia 2012a

WIKIPEDIA: *Daisy chain* — *Wikipedia, Die freie Enzyklopädie*. [http://de.wikipedia.org/w/index.php?title=Daisy\\_chain&oldid=98475104](http://de.wikipedia.org/w/index.php?title=Daisy_chain&oldid=98475104). Version: 2012. – [Online; Stand 11. Februar 2012] **1**

### Wikipedia 2012b

WIKIPEDIA: *Stiftleiste* — *Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Stiftleiste&oldid=99052435>. Version: 2012. – [Online; Stand 11. Februar 2012]

## Eidesstattliche Erklärung

Hiermit versichere ich, dass ich den vorliegenden Bericht:

*Übersetzen von Schrittmotorprotokollen*  
*Entwurf eines Hardwareübersetzers*

selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel verfasst habe.

Remagen, den 12. Februar 2012



JOHANNES DIELMANN

## A. Anhang

## **A.1. Schritt für Schritt Anleitung**

Eine Schritt für Schritt Anleitung zum vollständigen Scannen und exportieren eines 3D-Objektes.

## A.2. ASCII Befehlssatz R+S Schrittmotorsteuerung

_A	Motorstatus liefern
_C n	konstante Geschwindigkeit einstellen
_D n	Bezugswert definieren
_E n	Motorstrom einstellen
_F	Standardeinstellungen aktivieren
_H	Sanfter stop
_I	4-Bit-Eingang lesen
_J jdss	Joystickparameter einstellen
_L n	lokalen Modus aktivieren/beenden
_M n	n Schritte ausführen
_MA n	zu n bewegen
_MC n	mit konstanter Geschwindigkeit bewegen
_MCA n	MA mit konstanter Geschwindigkeit
_MCL n	MC zu Endschalterposition
_ML n	zur Endschalterposition bewegen
_N n	Zeilenvorschub (LF, hex. 0A) einfügen/löschen
_O n	n an 4-Bit-Ausgang senden
_P nnnn	Motorparameter einstellen
_Q	Parameter in EEROM speichern
_R n	Mikroschritteilung einstellen
_RL	Endschalterwerte lesen
_RS	verbleibende Schritte lesen
_S	Nothalt
_T n	Eingang n auslösen
_W	Position anfordern

Tabelle A.1.: ASCII Befehlssatz R+S Schrittmotorsteuerung  
[V9141 \[2001\]](#)

### A.3. Vom Autor verwendete Software

Hier ist die verwendete Software aufgelistet. Soweit es möglich war, wurden Open-Source-Programme eingesetzt. **(TODO: ÜBERARBEITEN!!!)**

- **RapidForm2004**

asdf

- **AVRStudio 5**

Atmel.

Website: <http://www.atmel.com/>

- **Eclipse**

Eclipse mit CDT und AVRPlugin

Website: <http://www.eclipse.org/>

- **AVRDude**

Prorammer

**(TODO: WEITERE?!)**