

Facharbeit im Fach Physik

Aufbau und Funktion eines digitalen Interface für eine Antennenrotorsteuerung

Verfasser: Björn Biesenbach

Grundkurs: Physik

Kursleiter: Klaus Paschenda

Abgabetermin: 27. Februar 2004

Bearbeitungszeitraum: 19.12.03 - 27.2.04

Inhaltsverzeichnis

1. Einleitung	3
1.1. Aufgabenstellung	3
1.2. Gegebenes	3
1.2.1. Antennenrotor	3
1.2.2. Steuergerät	4
1.2.3. Physikalische Eigenschaften des belasteten Rotors	5
2. Das digitale Steuerungsmodul	6
2.1. Anforderungen	6
2.1.1. Das Konzept	6
2.2. Hardware	7
2.2.1. Mikrocontroller	7
2.2.2. AD - Wandler	7
2.2.3. Eingabe	8
2.2.4. Spannungsversorgung	8
2.3. Software	9
2.3.1. Grundsätzliche Programmstruktur	9
2.3.2. Grundsätzlicher Funktionsablauf	10
2.3.3. Umsetzung	10
2.4. Entwicklungswerkzeuge	12
3. Nutzung und Technische Daten	13
4. Schlusswort	14
Literatur	15
A. Schaltplan	16
B. Programmcode	18

1. Einleitung

Die Idee zu diesem Projekt entstand bereits im Sommer 2003. Ich hatte angefangen mich in die Materie der Mikrocontroller einzuarbeiten, als ich auf der Suche nach etwas war, worauf ich hinarbeiten konnte. Die Idee des Aufbaus eines digitalen Interface für eine Antennenrotorsteuerung war das, wonach ich lange gesucht hatte. Viele Themengebiete konnten vereint werden und als Ergebniss sollte etwas entstehen das ich mit meinem Hobby, dem Amateurfunk ¹, auch alltäglich verwenden konnte.

1.1. Aufgabenstellung

Eine Antennenanlage bestehend aus einer oder mehreren Richtantennen soll mit einer Genauigkeit von etwa 5 Grad bequem ausrichtbar sein. In einigen Fällen muss die Antenne in bestimmte Richtungen gedreht werden, zum Beispiel bei Ausrichtung auf eine bekannte Gegenstation oder einen Satelliten. Hierfür soll ein Gerät aufgebaut werden, das es dem Benutzer ermöglicht, durch direkte Eingabe der gewünschten Richtung, den Rotor in die entsprechende Position zu fahren.

Vorhanden ist bereits der Antennenrotor samt Steuergerät. Die Position soll in dezimaler Form eingegeben werden. Es soll ein digitales Steuerungsmodul aufgebaut werden.

1.2. Gegebenes

1.2.1. Antennenrotor

Der Antennenrotor ist ein Elektromotor der speziell zum Drehen von Antennenanlagen konstruiert ist. Richtantennen müssen in vielen Fällen in alle Richtungen drehbar sein, deshalb besitzen die meisten Rotoren einen Drehbereich zwischen 0° - 360° bzw. neuere von 0° - 480° . Der hier verwendete Rotor (Abbildung 1) hat einen drehbaren Bereich von 360° .

¹DO2BBA, seit dem 5.2.03 lizenziert

1. Einleitung



Abbildung 1: Der Rotor

1.2.2. Steuergerät



Abbildung 2: Das Steuergeraet

Das vorhandene Steuergerät für den hier verwendeten Rotor besitzt drei Taster zum Drehen des Rotors, sowie ein Messinstrument zur Anzeige der Ausrichtung des Rotors. Ein Taster ist zum Lösen der Bremse, die anderen beiden zum Drehen nach rechts, bzw. nach links (Abbildung 2).

1.2.3. Physikalische Eigenschaften des belasteten Rotors

Die bei diesem Rotor gemessene Umlaufdauer T beträgt ungefähr 60 Sekunden. Somit ist die Winkelgeschwindigkeit $\omega = \frac{2\pi}{T} = 0.105 \frac{1}{s}$. Die Nachlaufzeit dieses Rotors beträgt etwa eine halbe Sekunde, was somit einem Bremsweg von 3° entspricht. Physikalisch wird hier der Bremswinkel angegeben, weil dieser sich leichter in Relation zur gewünschten Genauigkeit setzen lässt.

2. Das digitale Steuerungsmodul

2.1. Anforderungen

Das digitale Steuerungsmodul hat die Aufgabe, mit Hilfe des vorhandenen Steuergerätes des Antennenrotors, den Rotor an eine bestimmte, vom Benutzer eingegebene, Position zu fahren. Der gewünschte Winkel soll digital eingegeben werden. Dazu ist es nötig, die am Steuergerät vorhandenen Tasten durch Relais zu ersetzen, die dann entsprechend geschaltet werden, sowie die Spannung die am Messinstrument anliegt, abzugreifen. Mit Hilfe der Relais wird die Bremse gelöst, und der Rotor entweder links oder rechts gedreht. Die am Messinstrument anliegende Spannung dient zur Auswertung der momentanen Position des Rotors. Als Eingabegerät soll ein Tastenfeld dienen. Dieses muss vom digitalen Teil ausgewertet werden.(Abbildung 3)

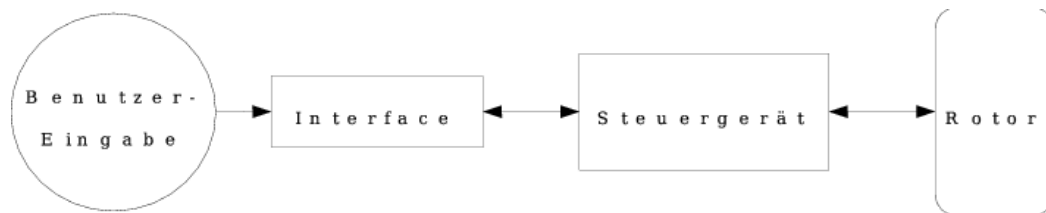


Abbildung 3: Allgemeine Darstellung

2.1.1. Das Konzept

Das Konzept (Abbildung 4) lässt sich im Wesentlichen mit dem Prinzip EVA² beschreiben. Dem Mikrocontroller³ wird über das Tastenfeld ein Wert in digitaler Form zugeführt, während die vom Steuergerät kommende Spannung analog ist (Eingabe). Da Berechnungen innerhalb des Mikrocontrollers jedoch nur digital durchgeführt werden können, muss die analoge Spannung vorher über einen AD - Wandler⁴ digitalisiert werden.

²Eingabe, Verarbeitung, Ausgabe

³Ein Prozessor der Peripherie wie RAM, Ports oder AD-Wandler integriert hat.

⁴Analog-Digital-Wandler; Umrechnung einer analogen Spannung in einen digitalen Wert

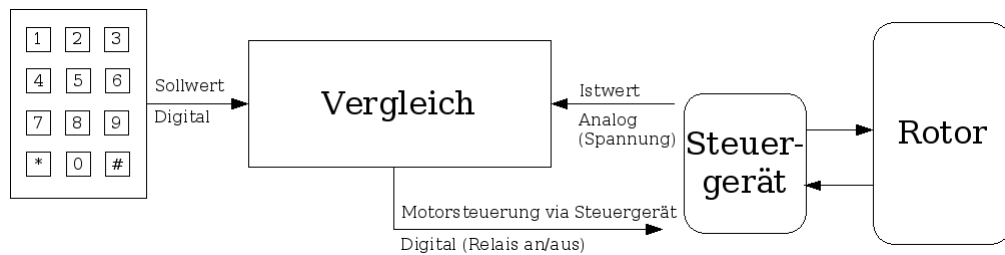


Abbildung 4: Das Konzept

Diese Werte werden nun verglichen (Verarbeitung) und über die Relaisschaltung wird der Rotor an die entsprechende Position gefahren (Ausgabe).

2.2. Hardware

2.2.1. Mikrocontroller

Der Mikrocontroller ist das Herzstück der gesamten Schaltung. Er übernimmt die Auswertung des Tastenfeldes, die Berechnung der aktuellen Position mit Hilfe seines AD-Wandlers und die Steuerung der Relais, welche mit den Tastern des Steuergerätes verbunden sind.

Die Wahl des Mikrocontrollers fällt auf den AT90S4433 der Atmel AVR Serie. Dieser bietet sich aufgrund seiner 20 I/O Ports, seines integrierten AD-Wandlers und seiner ISP⁵ Fähigkeit an. [Atm03]

2.2.2. AD - Wandler

Der im Mikrocontroller integrierte AD - Wandler rechnet die am Messinstrument anliegende Spannung in einen digitalen Wert um. Hierzu wird die Spannung mit einem Referenzwert verglichen, in diesem Fall mit 5V, und anteilig dargestellt. Der hier gewählte AD - Wandler besitzt eine Auflösung von 10-Bit, das heisst er kann das Verhältniss zwischen zu messender Spannung und Referenzspannung in $2^{10} = 1024$ Schritten darstellen.

⁵'In System Programmable', der Mikrocontroller kann ohne grossen Aufwand in seiner Schaltung programmiert werden

Da diese hohe Genauigkeit, aufgrund der Ungenauigkeit des Rotors, allerdings keinen Sinn machen würde, werden die 10-Bit auf 8-Bit heruntergerechnet. Somit stehen noch $2^8 = 256$ Schritte zur Verfügung. Dies verringert die Genauigkeit der Messung auf $\frac{360^\circ}{256} = 1.4^\circ$, was mehr als ausreichend ist.

Die vom Steuergerät abgegriffene Spannung hat einen gemessenen Minimalwert von etwa 0V bei 0° , und einen Maximalwert von etwa 12V bei 360° . Die Spannung verhält sich im Zwischenbereich *proportional zum Drehwinkel*. Diese Spannung wird mit Hilfe eines einfachen Spannungsteilers auf 5V gebracht, da die Referenzspannung am AD-Wandler auch 5V beträgt. Zusätzlich muss die abgegriffene Spannung noch durch einen Tiefpass geglättet werden.

2.2.3. Eingabe

Die Eingabe der Werte erfolgt über ein Tastenfeld mit 12 Tasten (vgl. Abbildung 4). Das hier verwendete Tastenfeld besitzt eine gemeinsame Masse, das heißt es wird pro Taster nur ein Pin nach aussen geführt, der zweite ist mit dem aller anderen Taster verbunden. Aus diesem Grund werden die Taster direkt an die IO-Pins des Mikrocontrollers angeschlossen.

2.2.4. Spannungsversorgung

Das Interface soll mit einer Spannung von etwa 12V versorgt werden können. Da das Herzstück der Schaltung, der Mikrocontroller, eine Spannung von 5V benötigt, bietet es sich an, die gesamte Schaltung für 5V Betriebsspannung auszulegen. Der einfachste Weg, um diese stabile Spannung zu bekommen, ist der Einsatz eines Festspannungsreglers. Um eine passende Wahl zu treffen, ist die Berechnung der ungefähren, maximalen Stromaufnahme der Schaltung nötig. Die Verlustleistung der Transistoren zur Ansteuerung der Relais ist zu vernachlässigen, da diese kaum messbar klein ist.

Die Stromaufnahme der Einzelkomponenten beträgt in etwa: 9mA Mikrocontroller; 20mA Quarzoszillator; 160mA Relais; 90mA Leuchtdioden. Die maximale Stromaufnahme liegt somit in der Größenordnung von etwa 0.3A.

Da wir nun eine Eingangsspannung von 12V haben, muss der Spannungsregler die restliche Leistung abführen.

$$U_{\text{Differenz}} = U_{\text{Eingang}} - U_{\text{Ausgang}}$$

$$U_{\text{Differenz}} = 12\text{V} - 5\text{V} = 7\text{V}$$

$$P_{\text{Verlust}} = U_{\text{Differenz}} * I = 7\text{V} * 0.28\text{A} = 1.96\text{W}$$

Die Wahl fällt auf einen Spannungsregler der 7805 Serie. Bei der errechneten Verlustleistung von fast 2W ist eine Kühlung des Spannungsreglers sinnvoll [Mic98].

2.3. Software

2.3.1. Grundsätzliche Programmstruktur

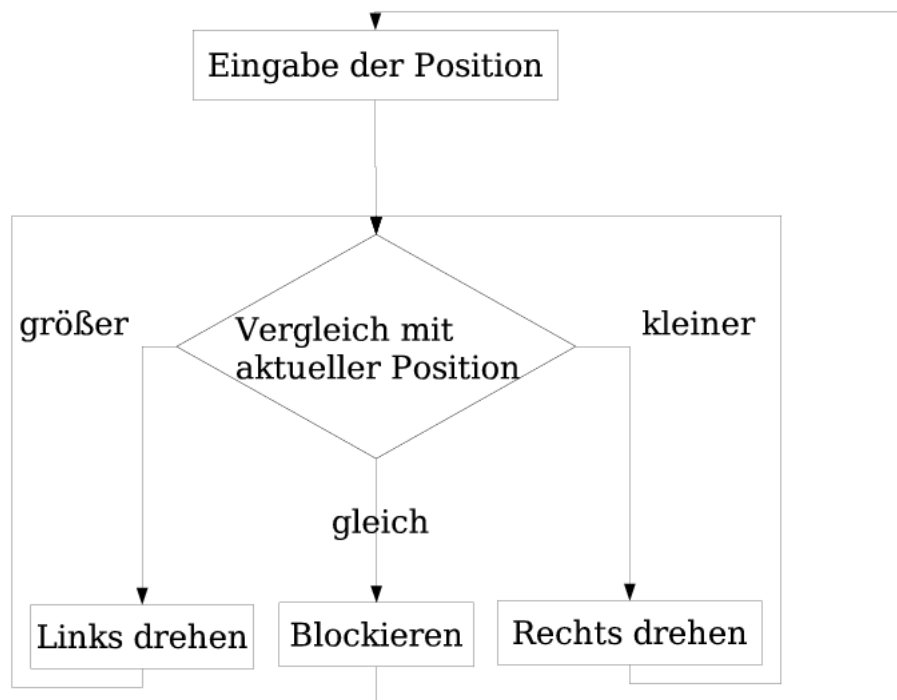


Abbildung 5: Allgemeiner Programmlaufplan

Wie in Abbildung 5 zu sehen ist, soll das Programm nach Eingabe der gewünschten Position mit dem Drehen des Rotors beginnen und den eingegebenen Wert fortlaufen mit dem vom AD-Wandler zurückgegebenen Wert vergleichen. Sind beide Werte identisch, wird das Drehen abgebrochen, die Bremse nach 3 Sekunden wieder geschlossen, und das digitale Steuerungsmodul wartet auf eine neue Positionseingabe.

Zur Schonung des Rotors ist im Programm noch eine Verzögerung zwischen Brems- und Drehvorgang implementiert worden. Die Bremse wird bereits drei Sekunden vor Beginn des Drehvorgangs gelöst. Ist der Drehvorgang beendet, vergehen wiederum drei Sekunden, bevor die Bremse wieder geschlossen wird.

2.3.2. Grundsätzlicher Funktionsablauf

Nach Einschalten des Steuerungsmoduls wird der Mikrocontroller zurückgesetzt und beginnt mit der Abarbeitung seines Programmes. Er beginnt mit der Initialisierung der Ports, Timer und des AD-Wandlers. Hiernach ist er bereit, die Benutzereingabe entgegenzunehmen. Die momentane Position des Rotor spielt bis hierhin keine Rolle.

Gibt der Benutzer nun die Position in Form von drei Ziffern ein, vergleicht der Mikrocontroller den eingegebenen Wert mit dem der aktuellen Position, welchen er über den AD-Wandler vom Steuergerät erhält. Sind diese Werte ungleich, so löst der Mikrocontroller die Bremse, wartet drei Sekunden, und beginnt dann mit dem Drehvorgang. Sobald der vom AD-Wandler errechnete Wert mit der eingegebenen Position übereinstimmt, wird der Motor abgeschaltet und die Bremse nach drei Sekunden wieder geschlossen. Der Benutzer kann am Messinstrument des Steuergerätes erkennen, ob der Rotor an der gewünschten Endposition angekommen ist.

2.3.3. Umsetzung

Die Umsetzung gestaltet sich als relativ komplex. Eine ausführliche Beschreibung würden den Rahmen dieser Facharbeit sprengen. Wie in Abbildung 6 zu sehen ist, werden zunächst nach dem Reset die Timer, Ports und der AD - Wandler initialisiert. Danach

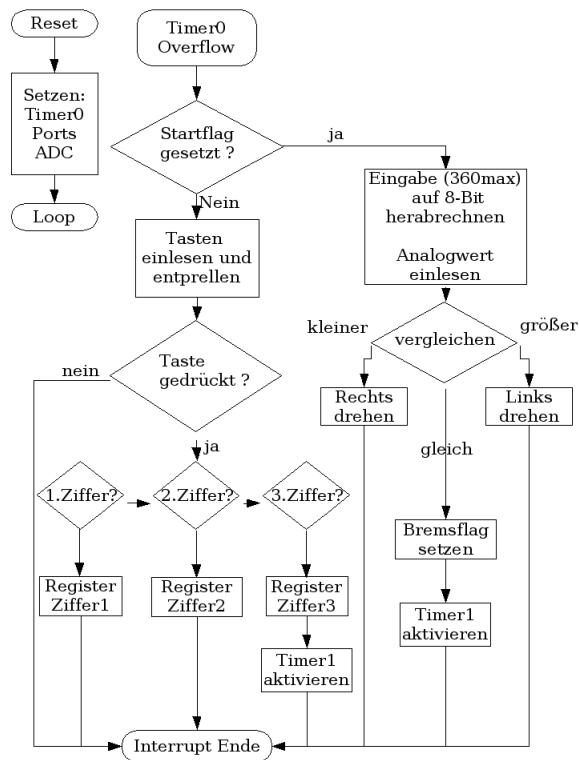


Abbildung 6: Komplexer Programmlaufplan Teil 1

begibt sich das Programm in eine leere Endlosschleife, welche nur durch die Interrupts ⁶ unterbrochen wird.

Der wichtigste Teil des Programms wird bei der Auslösung des Timer0 Overflow Interrupts abgearbeitet. Wurde das 'startflag' gesetzt, so wird bei jedem neuen Aufrufen dieses Interrupts der Teil zum Drehen des Rotors aufgerufen. Ist das 'startflag' nicht gesetzt, wird die Eingabe des Tastenfeldes entprellt [Dan03] und überprüft, ob eine Taste gedrückt wurde. Ist dies nicht der Fall, wird der Interrupt beendet. Wurde jedoch eine Taste gedrückt, wird überprüft die, welche Ziffer von drei möglichen eingegeben wurde, und die Ziffer wird in das entsprechende Register einsortiert. Nach Eingabe der dritten Ziffer wird der Timer1 aktiviert, welcher so konfiguriert ist, dass er genau einmal pro

⁶Unterbrechung im Programm bei bestimmten Ereignissen

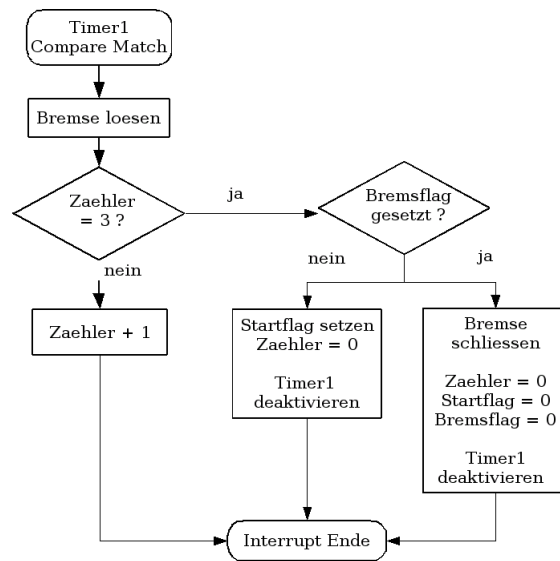


Abbildung 7: Komplexer Programmlaufplan Teil 2

Sekunde einen Interrupt auslöst (Abbildung 7)

Dieser Timer1 Interrupt leitet das Drehen des Rotors ein. Die Bremse wird gelöst, und drei Sekunden später wird das 'startflag' gesetzt. Danach deaktiviert sich der Timer1 von selbst.

2.4. Entwicklungswerkzeuge

Tavrasm [Mor04] ist das benutzte Programm zum assemblieren des Programmcodes. Es erstellt eine Binärdatei im Intel-Hex-Format.

Zur Übertragung dieser Binärdatei in den Flash ⁷ Speicher des Mikrocontrollers wird das Programm uisp [MM98] verwendet.

Der Schaltplan wurde mit dem Programm Eagle [Gmb03] erstellt.

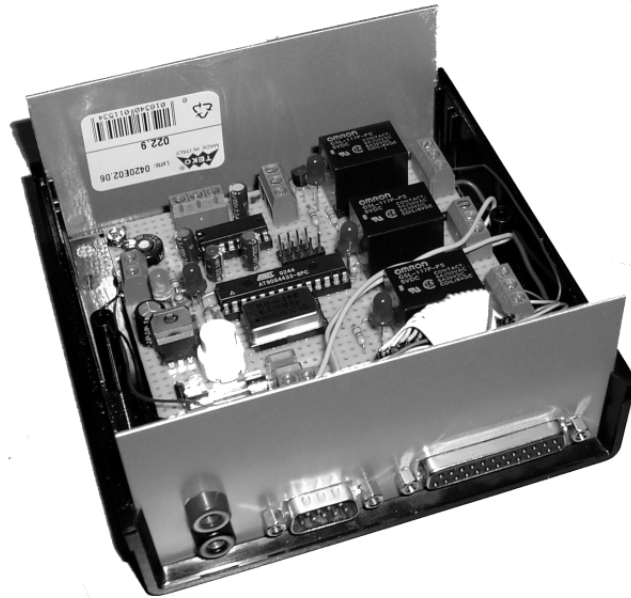


Abbildung 8: Die aufgebaute Platine im Gehäuse

3. Nutzung und Technische Daten

Die praktische Bedienung des Gerätes gestaltet sich sehr einfach. Über das Tastenfeld wird die gewünschte, *drei-stellige*, Position eingegeben. Der Rotor beginnt nun, sich an die gewünschte Position zu drehen. Dieser Vorgang kann mit der *Sterntaste* abgebrochen werden. In einer späteren Version des Steuerungsmoduls soll ein Bestätigungston das Erreichen der Position deutlich machen.

Die gewünschte Genauigkeit von etwa 5° wurde erreicht. Deshalb macht es Sinn, die Eingabe in Fünfer-Schritten vorzunehmen.

Betriebsspannung	7 - 14V
Stromaufnahme Standby (12V)	40mA
Stromaufnahme Betrieb (12V)	280mA

⁷nicht flüchtiger Speicher, hier lagert das Programm



Abbildung 9: Das abgesetzte Tastenfeld

4. Schlusswort

Es zeigte sich, dass der Aufbau des Interface sich schwieriger als erwartet gestaltete. Vor allem das Erlernen der richtigen, strukturierten Programmierung in Assembler auf dem Mikrocontroller hat viel Zeit gekostet. Dafür war die Freude, als sich der Rotor zum ersten Mal drehte um so größer. Das Gerät ist inzwischen voll funktionsfähig. Das Ziel des Projektes wurde in allen Punkten erfüllt.

Ein besonderer Dank geht an Andreas Schwarz, der mich bei der Erstellung des LaTeX Codes unterstützte, sowie an Hartmut Päsler, der mir bei technischen Fragen zur Seite stand.

Anhang

Literatur

- [Atm03] ATMEL, CORPORATION: *AT90S4433 Datasheet*. http://www.atmel.com/dyn/resources/prod_documents/doc1042.pdf, 2003.
- [BH01] BEIERLEIN, THOMAS und OLAF HAGENBRUCH: *Taschenbuch Mikroprozessortechnik*. Fachbuchverlag Leipzig, 2001.
- [Dan03] DANNEGGER, PETER: *Tasten entprellen*. Forum auf mikrocontroller.net: <http://www.mikrocontroller.net/forum/read-4-20549.html>, 2003.
- [Gmb03] GMBH, CADSOFT COMPUTER: *Eagle*. <http://www.cadsoft.de>, 2003.
- [Mic98] MICROELECTRONICS, SGS-THOMSON: *7805 Datasheet*. http://www.produktinfo.conrad.com/datenblaetter/175000-199999/179205-da-01-en-festspannungsregler_78xx.pdf, 1998.
- [MM98] MAREK MICHALKIEWICZ, THEODORE A. ROTH: *uisp*. <http://savannah.nongnu.org/projects/uisp/>, 1998.
- [Mor04] MORTENSEN, TOM: *tavrasm*. <http://www.tavrasm.org>, 2004.

A. Schaltplan

A. Schaltplan

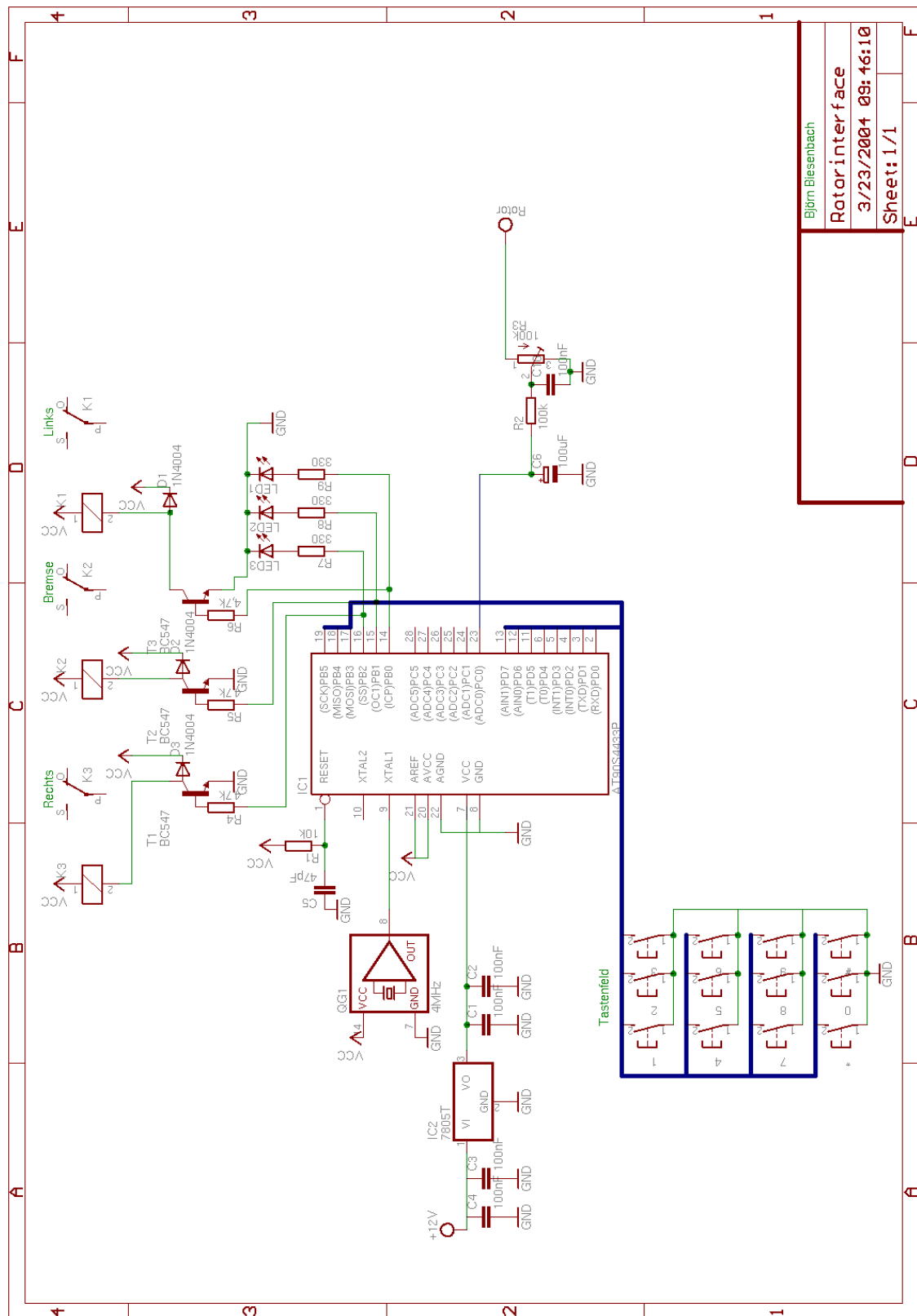


Abbildung 10: Schaltplan des Interface

B. Programmcode

B. Programmcode

Hiermit erkläre ich, dass ich die Facharbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benutzt habe.

Oer-Erkenschwick, den 27. Februar 2004

.....