

Diplomarbeit

Höhere Technische Bundes- Lehr- und Versuchsanstalt Salzburg
Abteilung für Elektrotechnik

Entwicklung eines emissionsfreien Sportmotorrades

Entwicklung der Zentralsteuerung / Projektleitung

Martin Kronberger 5AHET Betreuer: Dipl.-Ing. (FH) Johannes Ferner

Entwicklung des Antriebssystems

Jakob Lackner 5AHET Betreuer: Prof. Dipl.-Ing. MBA Adolf Reinhart

Entwicklung des Akkusystems

Simon Kern 5AHET Betreuer: Prof. Dipl.-Ing. Reinhold Benedikter

Entwicklung der mechanischen Komponenten

Tobias Schmeisser 5AHET Betreuer: Prof. Dipl.-Ing. Peter Lindmoser

Höhere Technische Bundeslehr-
und Versuchsanstalt Salzburg

Itzlinger Hauptstraße 30

A-5022 Salzburg

www.htl-salzburg.ac.at



Eidesstaatliche Erklärung

Wir erklären an Eides statt, dass wir die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht haben. Wir versichern, dass wir dieses Diplomarbeitsthema bisher weder im In- noch im Ausland (einer Beurteilerin oder einem Beurteiler) in irgendeiner Form als Prüfungsarbeit vorgelegt haben.

Gendererklärung

Aus Gründen der besseren Lesbarkeit wird in dieser Diplomarbeit die Sprachform des generischen Maskulinums angewendet. Es wird an dieser Stelle darauf hingewiesen, dass die ausschließliche Verwendung der männlichen Form geschlechtsunabhängig verstanden werden soll.

Martin Kronberger

Ort, Datum

Jakob Lackner

Ort, Datum

Simon Kern

Ort, Datum

Tobias Schmeisser

Ort, Datum

Vorwort

In immer mehr Großstätten werden Fahrzeuge mit Verbrennungsmotoren verboten. Viele Motorräder und Autos können nicht mehr produziert werden, da sie die immer strenger werdenden Abgasnormen nicht mehr einhalten können und das Thema der Klimaerwärmung wird immer präsenter und immer mehr Menschen versuchen ihren „carbon footprint“ zu verkleinern.

Doch leider gibt es für Motorradfahrer zumeist keine wirklichen alternativen, um für ihr Hobby auf eine emissionsfreie Alternative umzusteigen. Denn zumeist ist das Preis-Leistungsverhältnis, oder auch das Produkt selbst, nicht sehr einladend. Daher ist unser Ziel die Entwicklung in diesem Bereich voranzutreiben und dadurch den Markt zu vergrößern, wodurch immer mehr und bessere Produkte angeboten werden können.

Danksagung

TEXT DANKSAGUNG

DIPLOMARBEIT

DOKUMENTATION

DIPLOMA THESIS

DOCUMENTATION

Erklärung

Die unterfertigten Kandidaten haben gemäß §34 (3) SchUG in Verbindung mit §22 (1) Zi. 3 lit. b der Verordnung über die abschließenden Prüfungen in den berufsbildenden mittleren und höheren Schulen, BGBl. II Nr. 70 vom 24.02.2000 (Prüfungsordnung BMHS), die Ausarbeitung einer Diplomarbeit mit der umseitig angeführten Aufgabenstellung gewählt. Die Kandidaten nehmen zur Kenntnis, dass die Diplomarbeit in eigenständiger Weise und außerhalb des Unterrichtes zu bearbeiten und anzufertigen ist, wobei Ergebnisse des Unterrichtes mit einbezogen werden können. Die Abgabe der vollständigen Diplomarbeit hat bis spätestens

03.04.2020

beim zuständigen Betreuer zu erfolgen. Die Kandidaten nehmen weiters zur Kenntnis, dass gemäß §9 (6) der Prüfungsordnung BMHS nur der Schulleiter bis spätestens Ende des vorletzten Semesters den Abbruch einer Diplomarbeit anordnen kann, wenn diese aus nicht beim Prüfungskandidaten / bei den Prüfungskandidaten gelegenen Gründen nicht fertiggestellt werden kann.

Kandidaten / Kandidatinnen	Unterschrift
Martin Kronberger	
Jakob Lackner	
Simon Kern	
Tobias Schmeisser	

Prof. Dipl.-Ing. Reinhold Benedikter
Prüfer

Dipl.-Ing. (FH) Johannes Ferner
Prüfer

Prof. Dipl.-Ing. MBA Adolf Reinhart
Prüfer

Lindmoser, Prof. Dipl.-Ing. Peter
Prüfer

Prof. Dipl.-Ing. (FH) Roland Holzer
Abteilungsvorstand

Dipl.-Ing. Dr.techn. Franz Landertshamer
Direktor

Inhaltsverzeichnis

I Einführung	2
1 Projektteam	2
2 Projektbetreuer	3
3 Aufgabeneinteilung	3
II Einleitung	5
1 Motivation	5
2 Zielsetzung	5
3 Topologie des Gesamtsystems	5
4 Leitfaden	5
III Stand der Technik	6
1 Steuereinheiten	6
1.1 Battery Management System	6
1.2 Raspberry PI	6
2 Bussysteme	6
2.1 SPI Bus	6
2.2 CAN Bus	6
IV Mechanische Umsetzung	7
1 Section	7
V Human-Computer Interaction System	8
1 Übersicht	8
1.1 Grundfunktionen des Systems	8
1.2 Grundaufbau des Systems	9
2 Versorgung	10
2.1 Aufbau des Versorgungssystems	10
2.2 Spannungswandler	10
2.2.1 5V Versorgungssystem	10
2.2.2 12V Versorgungssystem	10
3 Steuerung der Peripherie	11
3.1 Hardware	11
3.1.1 Input	11
3.1.2 Output	11
3.2 Software	11
3.2.1 GPIO Zero	11
3.2.2 Threading	11
4 Benutzeroberfläche	12
4.1 Program Fenster	12
4.1.1 Login	12
4.1.2 Fahrdaten	12
4.1.3 Akku- und Ladedaten	12
4.1.4 Fahrdaten Diagnose	12
4.1.5 Errors	12
4.2 Realisierung der Benutzeroberfläche	12

4.2.1	QML	12
4.2.2	Qt-Quick	12
4.2.3	Slots und Signals	13
4.2.4	Bridge	13
5	Kommunikation	14
5.1	Hardware	14
5.2	Listener	14
5.2.1	Receive Data	14
6	Fahrdatenspeicher	15
6.1	Datenbankstruktur	15
6.1.1	Login System	15
6.1.2	Motor Daten	15
6.1.3	Akku Daten	15
6.2	Handler	15
6.2.1	SELECT Befehl	15
6.2.2	INSERT Befehl	15
VI	Antriebsstrang	16
1	Übersicht	16
1.1	Grundfunktionen des Systems	16
2	Hardwareaufbau des Antriebssystems	17
2.1	Mechanische Umsetzung	18
2.2	Der Laststromkreis	18
2.3	Verbindung zum Motor	18
2.3.1	Verbindung zum Akumulator	18
2.3.2	Sonstige Komponenten	18
2.4	Der Steuerstromkreis	18
2.4.1	Inputs	18
2.4.2	Outputs	18
3	Softwareaufbau des Antriebssystems	18
4	Steuerung der In- und Outputs (I/O Assingment)	18
4.1	Funktionen	18
4.2	Zuweisung	18
5	Drehmomentsteuerung (Torquecontrol)	18
5.1	Grundfunktion	18
5.2	Parameter	18
6	Kommunikation (CAN-Bus)	18
6.1	Grundfunktion	18
6.1.1	Parameter	18
VII	Akku und Ladekonzept	19
1	Section	19
VIII	Ergebnis	20
1	Section	20
A	Arbeitsnachweis	21
1	Zeitplan	21
2	Kosten	21
B	Programmier-Code	22
C	CAD-Zeichnungen	23
D	Schaltpläne	24
	Literaturverzeichnis	24
	Abbildungsverzeichnis	24

Tabellenverzeichnis	25
Codeverzeichnis	26

Kapitel I

Einführung

1 Projektteam



Martin Kronberger



Jakob Lackner



Simon Kern



Schmeisser Tobias

2 Projektbetreuer

Prof. Dipl.-Ing. Reinhold Benedikter

unterstützte Jakob Lackner bei der Entwicklung des Akku- und Ladesystems

Dipl.-Ing. (FH) Johannes Ferner

unterstützte Martin Kronberger bei der Entwicklung des Human-Computer Interaction Systems

Prof. Dipl.-Ing. MBA Adolf Reinhart

unterstützte Jakob Lackner bei der Entwicklung des Antriebssystems

Lindmoser, Prof. Dipl.-Ing. Peter

unterstützte Tobias Schmeisser bei der Entwicklung der mechanischen Komponenten

3 Aufgabeneinteilung

Martin Kronberger

- Projektleitung
- Projektfindung und Projektplanung
- Projektaufteilung
- Erstellen der Einreichdokumente
- Entwickeln der Hardware des Human-Computer Interaction Systems
- Entwickeln der Software des Human-Computer Interaction Systems
- Planung und Umsetzung der elektrischen Installation
- Verfassen der Dokumentation

Jakob Lackner

- Projektleitung
- Projektfindung und Projektplanung
- Projektaufteilung
- Entwicklung des Antriebssystems
- Entwicklung der Software des Motorsteuergerätes
- Erstellen der Einreichdokumente
- Verfassen der Dokumentation

Simon Kern

- Projektleitung
- Projektfindung und Projektplanung
- Projektaufteilung
- Entwicklung des Akkusystems
- Erstellen der Einreichdokumente
- Verfassen der Dokumentation

Tobias Schmeisser

- Projektleitung
- Projektfindung und Projektplanung
- Projektaufteilung
- Entwicklung der mechanischen Komponenten
- Entwicklung der Getriebemechanik
- Erstellen der Einreichdokumente
- Verfassen der Dokumentation

Kapitel II

Einleitung

- 1 Motivation
- 2 Zielsetzung
- 3 Topologie des Gesamtsystems
- 4 Leitfaden

Kapitel III

Stand der Technik

1 Steuereinheiten

1.1 Battery Management System

1.2 Raspberry PI

2 Bussysteme

2.1 SPI Bus

2.2 CAN Bus

Kapitel IV

Mechanische Umsetzung

1 Section

Kapitel V

Human-Computer Interaction System

1 Übersicht

Das Human-Computer Interaction System ist, wie der Name schon verrät, die Komponente, welche als Schnittstelle zwischen dem Nutzer und dem gesamten elektrischen System dient. Durch es sollte die fehlerfreie Nutzung der Funktionen des Motorrades gewährleistet sein, ebenso sollte es wichtige Fahrdaten und andere Informationen speichern und dem User anzeigen können. Wichtig ist das System, trotz der großen Komplexität, so intuitiv und nutzerfreundlich wie möglich zu gestalten.

1.1 Grundfunktionen des Systems

Die geplanten Funktionen des HCIS lassen sich grob in vier Grundfunktionen einteilen.

- **Steuerung der Peripherie**

Die Schalter und Buttons am Lenker, welche zuvor über den Kabelbaum die Leuchten, Blinker oder der Hupe gesteuert haben. Werden nun über die General-purpose input/output (GPIO) anschlüsse des Raspberry Pi Micro Computers gesteuert.

- **Graphische Benutzeroberfläche**

Dient der Anzeige wichtiger Fahr- und Ladedaten welche entweder in echtzeit oder über die Datenbankschnittstelle abgerufen und graphisch angezeigt werden können.

- **Kommunikation mit den Steuereinheiten des Motorrades**

Über CAN-Bus werden Daten von dem Batterie Management Systems (BMS) und der Curtis Motorsteuerung empfangen und an die Benutzeroberfläche zur echtzeit verwertung und an die Datenbankschnittstelle zur Langzeitsicherung der Fahrdaten weiter gegeben.

- **Speichern der relevanten Fahrdaten über die Datenbankschnittstelle**

Die über den CAN-Bus empfangenen Daten werden sofort an die Datenbankschnittstelle (Handler) weitergegeben um für Datenauswertung und Testberichte die Daten zu speichern. Ebenso bezieht das Diagnosesystem der Benutzeroberfläche die Daten über diese Schnittstelle.

1.2 Grundaufbau des Systems

In der Abbildung wird der Grundaufbau des Systems und die Datenverbindungen der folgenden Komponenten veranschaulicht.

- Raspberry Pi - Die Steuereinheit des Systems.
Kommuniziert über CAN-Bus mit den anderen Steuerkomponenten des Motorrads.
- User Input - Die vorhandenen Buttons am Lenker des Motorrads werden über pull down Widerstände mit den Inputs des Raspberry Pi verbunden.
- Peripherie - Die Grundkomponenten des Motorrads wie Scheinwerfer oder Hupe.
- Dashboard - Der Bildschirm zur Anzeige der Verarbeiteten Informationen.

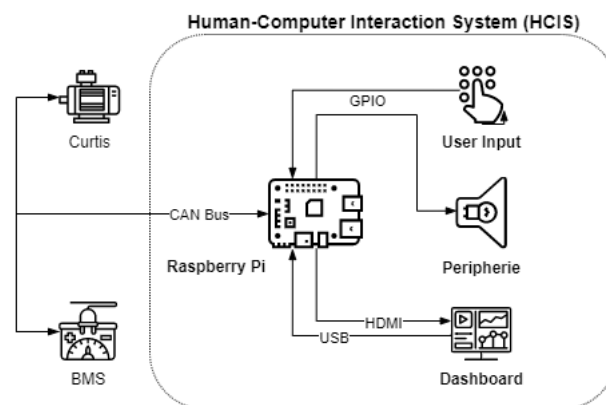


Abbildung V.1: Grundaufbau des Human-Computer Interaction Systems

Nicht in der Abbildung dargestellt ist die Versorgung der einzelnen Komponenten, welche in dem folgenden Abschnitt noch genauer erläutert wird.

2 Versorgung

2.1 Aufbau des Versorgungssystems

2.2 Spannungswandler

2.2.1 5V Versorgungssystem

2.2.2 12V Versorgungssystem

3 Steuerung der Peripherie

3.1 Hardware

3.1.1 Input

3.1.2 Output

3.2 Software

3.2.1 GPIO Zero

3.2.2 Threading

4 Benutzeroberfläche

4.1 Program Fenster

4.1.1 Login

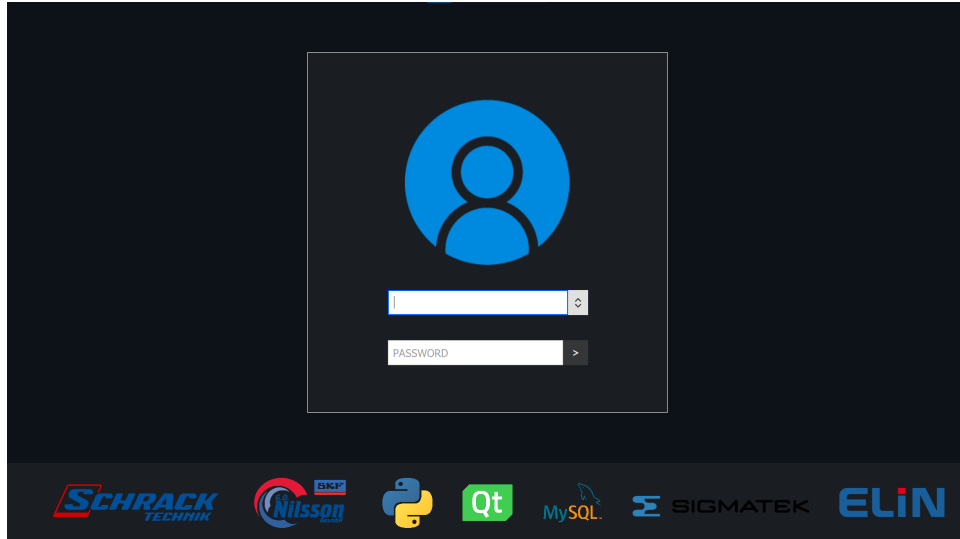


Abbildung V.2: Login Fenster der Benutzeroberfläche

4.1.2 Fahrdaten

4.1.3 Akku- und Ladedaten

4.1.4 Fahrdaten Diagnose

4.1.5 Errors

4.2 Realisierung der Benutzeroberfläche

4.2.1 QML

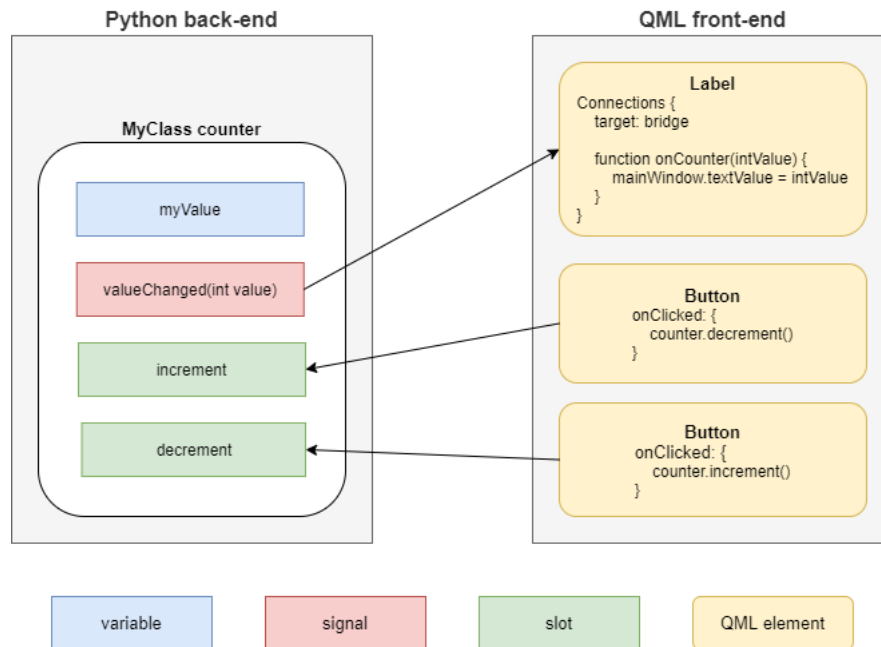
QML ist eine deklarative Sprache, mit der Benutzeroberflächen anhand ihrer visuellen Komponenten und ihrer Interaktion und Beziehung zueinander beschrieben werden können. Es ist eine gut lesbare Sprache, die entwickelt wurde, um die dynamische Verbindung von Komponenten zu ermöglichen, und die die einfache Wiederverwendung und Anpassung von Komponenten innerhalb einer Benutzeroberfläche ermöglicht. Es bietet eine gut lesbare, deklarative, Syntax mit Unterstützung für JavaScript-Ausdrücke in Kombination mit dynamischen Eigenschaftsverbindungen.

4.2.2 Qt-Quick

Das Qt-Quick-Modul ist die Standardbibliothek zum schreiben von QML-Anwendungen. Während das QML-Modul die Engine und die Sprachinfrastruktur bereitstellt, bietet das Qt Quick-Modul alle grundlegenden Typen, die zum Erstellen von Benutzeroberflächen mit QML erforderlich sind. Es bietet eine visuelle Zeichenfläche und Typen zum Erstellen und Animieren visueller Komponenten, zum Empfangen von Benutzereingaben, zum Erstellen von Datenmodellen und Ansichten sowie zum verzögerten Objektinstanzieren. Es können problemlos flüssige, animierte Benutzeroberflächen in QML erstellt werden. Diese Benutzeroberflächen können mit beliebigen Back-End Bibliotheken verbunden werden.

4.2.3 Slots und Signals

Slots und Signals werden in Qml zur ereignisgesteuerte Kommunikation zwischen front-end und back-end verwendet. In der folgenden Illustration wird diese anhand eines einfachen Beispiels erklärt.



Signale

Diese können als Mitteilungen angesehen werden welche über das Aufrufen der `signal.emit` funktion vom back-end an das front-end geschendet wird. Im front-end wird wiederum eine eigens definierte Funktion benötigt um dem Wert einem Property eines QML Elements zuzuweisen.

Slots

Slots sind Call-Back Funktionen, welche im Back-End definiert werden und sind über die Bridge Class mit dem Front-End verknüpft. Dadurch können diese Funktionen im Front-End aufgerufen und mit Signalen verbunden werden. Sie stellen daher die wichtigste Verbindung zwischen dem Programm und der Benutzeroberfläche dar.

4.2.4 Bridge

5 Kommunikation

5.1 Hardware

5.2 Listener

5.2.1 Receive Data

6 Fahrdatenspeicher

6.1 Datenbankstruktur

6.1.1 Login System

6.1.2 Motor Daten

6.1.3 Akku Daten

6.2 Handler

6.2.1 SELECT Befehl

6.2.2 INSERT Befehl

Kapitel VI

Antriebsstrang

1 Übersicht

Die Hauptaufgabe des Antriebssystems ist die Umwandlung der von dem Akkumulator zur Verfügung gestellten elektrischen Energie in die kinetische Energie. Diese tritt zuerst kreisförmig am Motor auf und wird zunächst über das direkt Getriebe umgeformt und auf die passende Drehzahl gebracht, anschließend wird die kreisförmige kinetische Energie mithilfe des Hinterrades auf die Straße übertragen und das ganze Motorrad beschleunigt. Neben dem Antrieb des Motorrades hat die Motorsteuerung noch weitere Bedeutung als Steuereinheit, diese fungiert als Bindemittel zwischen dem Human-Computer Interacting System und den elektrischen Anforderungen an das Gesamtsystem.

1.1 Grundfunktionen des Systems

Die geplanten Funktionen des Antriebssystems lassen sich grob in zwei Grundfunktionen einteilen.

- Der Antrieb - Translation ist eine Grundfunktion eines jeden Verkehrsmittels
Durch die Umwandlung der elektrischen in kinetische Energie erfährt das gesamte System eine Translation.
- Die Steuereinheit - Steuerung und Kommunikation mit anderen Betriebsmitteln
Realisiert durch In- und Outputs, Datenübertragung mithilfe des CAN-Buses

Nun unterscheiden wir zwischen dem Hardware- und dem Softwareaufbau des Antriebssystems.

2 Hardwareaufbau des Antriebssystems

Der grundsätzliche Hardwareaufbau des Antriebssystems lässt sich in zwei galvanisch getrennte Stromkreise und der mechanischen Umsetzung unterscheiden:

- Die mechanische Umsetzung (Kraftübertragung und Montage)
Umfasst das Getriebe und die Befestigung aller Komponenten am Rahmen
- Der Laststromkreis
Beinhaltet die Verbindung des Motorcontrollers mit dem Motor und dem Akkumulator.
- Der Steuerstromkreis
Beinhaltet alle elektrischen Verbindungen, welche mithilfe des 35-poligen Niederleistungs-Steckers mit dem Motorcontroller verbunden sind.

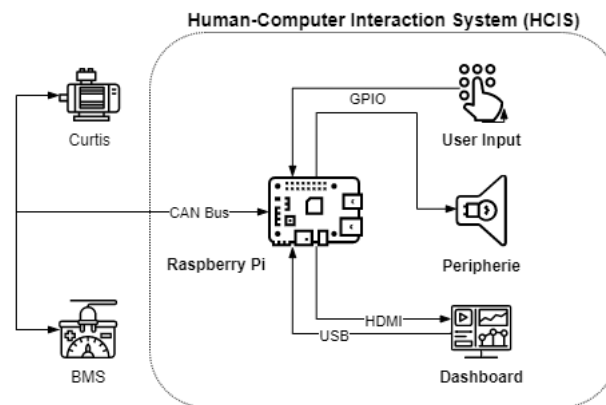


Abbildung VI.1: Grundaufbau des Human-Computer Interaction Systems

Nicht in der Abbildung dargestellt ist die Versorgung der einzelnen Komponenten, welche in dem folgenden Abschnitt noch genauer erläutert wird.

/

2.1 Mechanische Umsetzung

Dieser Teil des Antriebssystems wurde vollständig von Tobias Schmeisser übernommen. -> link

2.2 Der Laststromkreis

2.3 Verbindung zum Motor

2.3.1 Verbindung zum Akumulator

2.3.2 Sonstige Komponenten

2.4 Der Steuerstromkreis

2.4.1 Inputs

Encoder Throttle Switches

2.4.2 Outputs

Spannungswandler Driver

3 Softwareaufbau des Antriebssystems

4 Steuerung der In- und Outputs (I/O Assingment)

4.1 Funktionen

4.2 Zuweisung

5 Drehmomentsteuerung (Torquecontrol)

5.1 Grundfunktion

5.2 Parameter

6 Kommunikation (CAN-Bus)

6.1 Grundfunktion

6.1.1 Parameter

Kapitel VII

Akku und Ladekonzept

1 Section

Kapitel VIII

Endergebnis

1 Section

Anhang A

Arbeitsnachweis

- 1 Zeitplan
- 2 Kosten

Anhang B

Programmier-Code

Anhang C

CAD-Zeichnungen

Anhang D

Schaltpläne

Abbildungsverzeichnis

V.1 Grundaufbau des Human-Computer Interaction Systems	9
V.2 Login Fenster der Benutzeroberfläche	12
VI.1 Grundaufbau des Human-Computer Interaction Systems	17

Tabellenverzeichnis

Listings