

Diplomarbeit

Höhere Technische Bundeslehr- 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

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 Großstädten 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 Möglichkeiten, um für ihr Hobby auf eine emissionsfreie Alternativen 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

Besonders für die informelle Unterstützung im Bezug auf die technische Auslegung, Konfiguration und Programmierung des Curtis Controllers möchten wir uns auch bei den Technikern (Bereich E-Mobilität) Daniel Kramser, Sebastian Bayer und Wolfgang Strasser bedanken.

DIPLOMARBEIT

DOKUMENTATION

Namen der Verfasser	Martin Kronberger, Jakob Lackner, Simon Kern, Tobias Schmeisser	
Jahrgang / Schuljahr	5AHET, 2020/2021	
Thema der Diplomarbeit	Entwicklung eines emissionsfreien Sportmotorrades	
Kooperationspartner	Schrack for Students, Sigmatek , EBG GmbH,	
Aufgabenstellung	<p>Es soll ein vollständig elektrifiziertes E-Motorrad aus einem alten Model mit Verbrennungsmotor entwickelt werden. Der vorherige Verbrennungsmotor wird durch eine elektrische Steuer- und Motoreinheit ersetzt. Ebenfalls soll das Zweirad über ein eigenständiges Akkusystem, mit einem für das Motorrad individuellem Ladesystem verfügen. Über eine zentrale Steuereinheit wird die Kommunikation zwischen den Systemen und dem Menschen gewährleistet.</p>	
Realisierung	<p>Als Chassis wird eine ausgeschlachtete Duacati Monster S4 2001 verwendet und mit einem eigens entwickelten Rahmen versehen. Angetrieben wird das Motorrad über eine bürstenlose Synchronmaschine, welche über ein BMS gesteuertes 50,4 Volt Lithiumionen Akkupack versorgt wird. Das Moment wird vom Motor über ein Kettengetriebe mit einer Übersetzung von ungefähr 1/9 auf die Straße übertragen. Die Peripherie wird über einen Raspberry Pi Minicomputer und Taster am Lenker gesteuert. Ebenso wird über ihn die Benutzeroberfläche gesteuert, welche über ein 11,6 Zoll Touch Panel angezeigt und gesteuert werden kann.</p>	
Ergebnisse	<p>Zum Abgabezeitpunkt befindet sich die Komponenten des Motorrads im Entwicklungszustand. Der Rahmen ist noch in der Endphase der Fertigung. Der Motor und dessen Software sind einsatzbereit und müsste nur mehr eingebaut werden. Die Versorgung ist fertig entwickelt und muss ebenso nur mehr gefertigt werden. Die Steuerung der Benutzeroberfläche befindet sich im Prototypen-Status, beinhaltet jedoch noch einige experimentelle Funktionen, welche noch etwas Programmier- und Testzeit benötigen. Das gesamte Konzept ist vollständig und kann mit etwas mehr Zeit und Sponsoren Gelder fertiggestellt werden.</p>	
Möglichkeit der Einsichtnahme in die Arbeit	<p>Die Diplomarbeit ist in gebundener Form sowohl in der Schulbibliothek als auch bei AV Prof. Dipl-Ing. (FH) Roland Holzer einzusehen. Darüber hinaus besitzt jedes Mitglied des Projektteams eine vollständige Version in gebundener und digitaler Form.</p>	
Approbation (Datum / Unterschrift)	Prüfer:	Abteilungsvorstand:

DIPLOMA THESIS

DOCUMENTATION

Authors	Martin Kronberger, Jakob Lackner, Simon Kern, Tobias Schmeisser	
Form / Academic year	5AHET, 2020/2021	
Topic	Development of an emission-free sports motorcycle	
Co-operation partners	Schrack for Students, Sigmatek , EBG GmbH,	
Assignment of Tasks	The plan is to develop a fully electrified e-motorcycle. The previous internal combustion engine is replaced by an control unit and electric motor unit. The two-wheeler should also have an independent battery system with a charging system that is individual for the motorcycle. Communication between the systems and humans is ensured via a central control unit.	
Realisation	A cannibalized Duacati Monster S4 2001 is used as the chassis and provided with a specially developed frame. The motorcycle is driven by a brushless synchronous machine, which is supplied by a BMS controlled 50.4 volt lithium ion battery pack. The torque is transmitted from the engine to the road via a chain gear with a ratio of about 1/9. The periphery is controlled via a Raspberry Pi minicomputer and button on the handlebar. It also controls the user interface, which can be displayed and controlled via an 11.6 inch touch panel.	
Results	The components of the motorcycle are at the time of delivery in the state of development. The framework is still in its final stages Production. The engine and its software are ready for use and would just have to be built in. The supply is over developed and just needs to be manufactured. The Control of the user interface is in the prototype status, however still contains some experimental functions which are still unavailable	
Accessibility of Diploma Thesis	The diploma thesis is available in the school library and at Prof. Dipl.-Ing (FH) Roland Holzer's office. Furthermore, each member of the project team has a complete version.	
Approval (Date/Signature)	Examiner:	Head of Department:

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

06.04.2020

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

Prof. Dipl.-Ing. (FH) Johannes Ferner
Prüfer

Prof. Dipl.-Ing. Adolf Reinhart, MBA
Prüfer

Prof. Dipl.-Ing. Peter Lindmoser
Prüfer

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

Dipl.-Ing. Dr.tech. 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	6
4 Leitfaden	8
III Stand der Technik	9
1 Kühlung	9
1.1 Thermodynamische Grundlagen	9
1.2 Technische Anwendung	10
1.3 Funktionsweise	10
1.4 Technische Kühlungsbeispiele	10
1.5 Wasserkühlung	11
1.5.1 Wasserkühlung bei elektronischen Geräten	11
1.5.2 Wasserkühlung in Personal Computern	12
1.6 Luftkühlung	13
1.6.1 Luftkühlung bei Personalcomputern	13
1.7 Aktive Kühlung	14
1.7.1 Aufbau	14
1.8 Passive Kühlung	14
2 Schutzarten Elektrischer Betriebsmittel	15
3 Solid Edge	16
3.1 Erklärung	16
4 Akkusysteme	18
5 Batteriearten	19
5.1 Bleiakkumulator	19
5.2 Nickel-Metallhybrid Akkumulatoren	19
5.3 Nickel-Cadmium Akkumulatoren	20
5.4 Lithium-Ionen Batterie	21
5.4.1 Geschichte	21
5.4.2 Allgemeines	21
5.4.3 Prinzip der Lithium-Ionen Batterie	22
5.4.4 Lagerung und Sicherheitshinweise	23
5.4.5 Anwendungsbereiche von Lithium-Ionen Akkumulatoren	24
5.5 Batteriemanagementsystem	25
5.5.1 Komponenten eines BMS	26
5.5.2 Battery-Balancing	27
6 Synchronmaschine mit Dauermagneterregung	28
6.1 Allgemeines	28
6.2 Aufbau	28
6.3 Funktionsweise	28

	6.4	Auswertung der Antriebswelle (Encoder)	29
7	Curtis Controller		30
	7.1	Allgemeines	30
	7.2	Feldorientierte Regelung	31
8	Der Regler		32
	8.1	Einleitung	32
	8.2	Der Regelkreis	32
	8.3	Der PID-Regler	34
9	Bussysteme		36
	9.1	CAN-Bus	36
	9.2	SPI-Bus	36
10	Programmiersprachen		37
	10.1	Python	37
	10.2	JavaScript	37
	IV Mechanische Umsetzung		38
1	Gehäuse		38
	1.1	Anforderung an das Gehäuse	38
	1.2	Dimensionierung	39
	1.3	Gewicht	39
	1.4	Gesamtgewicht des Zero-Emission-Power-Bikes:	39
	1.5	Festlegung der Maße	39
	1.6	Material	39
2	Getriebe		41
	2.1	Aufgabe des Getriebes	41
	2.2	Das Getriebe	41
3	Akkupacks		52
	3.1	Akkupack Vorderseite	52
	3.2	Akkupack Motorblock	53
	3.3	Akkupack Mitte	53
4	Akkukühlung		54
	4.1	Warum eine Kühlung notwendig ist	54
5	Zusammenbau		55
	V Akku und Ladekonzept		57
1	Übersicht		57
	1.1	Aufgaben der Energieversorgung	57
	1.2	Aufgaben des Batteriemanagement	57
2	Energieversorgung		58
	2.1	Dimensionierung der Akkuzellen	58
	2.2	Zusammenstellung der Batteriezellen	60
	2.3	Verschaltung der Batteriezellen	63
	2.3.1	Serienschaltung	63
	2.3.2	Parallelschaltung	64
	2.3.3	Kombination aus Serien- und Parallelschaltung	65
	2.3.4	Verschaltung der Zellen zu einem Akkupack	65
	2.3.5	Verschaltung der Akkupacks	66
	2.3.6	Geschätzt Betriebszeit des Akkumulators	67
3	Batteriemanagementsystem		68
	3.1	Funktionen des Batteriemanagementsystems	70
	3.1.1	Balancing	71
	3.1.2	Spannungsmessung	74
	3.1.3	Strommessung	74
	3.1.4	Temperaturüberwachung	74
	3.2	Verschaltung des Batteriemanagementsystems	75
	3.2.1	Anschließen der Akkupacks an das BMS	75
	3.2.2	Temperaturmessung mithilfe des BMS	76
	3.2.3	Anschlussplan der Last	77
	3.2.4	Kommunikation mittels CAN-Bus	78

3.3	Applikation des BMS Mini	79
3.3.1	Konfiguration der EMUS EVGUI App	81
3.4	Laderegelung	85
3.4.1	Verschaltung	86
VI Antriebsstrang		87
1	Übersicht	87
1.1	Grundfunktionen des Systems	87
2	Hardwareaufbau des Antriebssystems	88
2.1	Mechanische Umsetzung	88
2.2	Der Laststromkreis	89
2.2.1	Elektrische Energieübertragung	90
2.2.2	Leitungsschutzorgane	92
2.2.3	Motorbeschreibung	93
2.3	Der Steuerstromkreis	94
2.3.1	Übersicht Ein- und Ausgänge	94
2.3.2	Digitale Eingänge (Digital Inputs)	96
2.3.3	Analoge Eingänge (Analog Inputs)	96
2.3.4	Gas- und Bremseingänge (Throttle and Brake Inputs)	97
2.3.5	Positionsrückmeldung vom Encoder (Position-Feedback Input)	97
2.3.6	Prozessorversorgung und Spulenrücklauf (KSI and Coil Return)	98
2.3.7	Analoge Ausgänge (Analog Outputs)	98
2.3.8	Digitale und Pulsweitenmodulierbare Ausgänge (Digital and PWM Outputs)	99
2.3.9	Spannungsversorgungs-Ausgänge (Power Supply Outputs)	99
2.3.10	Kommunikations-Ports	100
3	Softwareaufbau des Antriebssystems	101
3.1	Parameterbasierte Programmierung (Programmer)	102
3.1.1	Allgemeines	102
3.1.2	Funktionen	102
3.2	Drehmomentsteuerung (Torquecontrol)	104
3.2.1	Parameter	104
3.2.2	ECO- und Sportmodus (Speed-Mode-Select)	106
3.3	Vehicle-Control-Language (VCL) Programmierung	107
3.3.1	Grundfunktion	107
3.3.2	Kommunikation (CAN-Bus)	108
3.3.3	Speed-Mode-Select	109
4	Inbetriebnahme	110
4.1	Leonard-Versuchsaufbau	110
4.2	Bleiakkumulator-Versuchsaufbau	112
VII Human-Computer Interaction System		113
1	Übersicht	113
1.1	Grundfunktionen des Systems	113
1.2	Steuereinheit	114
1.3	Grundaufbau des Systems	114
2	Spannungsversorgung	115
2.1	Aufbau des Versorgungssystems	115
2.1.1	12V Versorgungssystem	115
2.1.2	5V Versorgungssystem	116
2.1.3	Abschalten der Spannungswandler	116
3	Steuerung der Peripherie	117
3.1	Hardware	117
3.1.1	Input	117
3.2	Umsetzung	118
3.2.1	Lenkerschalter	118
3.2.2	Beleuchtung	119
3.2.3	Output	120
3.3	Software	120

3.3.1	gpiozero	120
3.3.2	threading	120
4	Benutzeroberfläche	121
4.1	Hardware	121
4.1.1	Befestigung	122
4.2	Software	122
4.2.1	Aufbau	122
4.2.2	Nutzer / Berechtigungen	123
4.3	Komponenten	123
4.3.1	Navigationsmenü	123
4.3.2	Balken Anzeige	124
4.3.3	Modus Anzeige	124
4.3.4	Graph	125
4.3.5	Weitere Komponenten	125
4.4	Programm Fenster	126
4.4.1	Login	126
4.4.2	Fahrdaten	126
4.4.3	Akku- und Ladedaten	127
4.4.4	Fahrdaten Diagnose	127
4.4.5	Fehler	128
4.4.6	Nutzer und Berechtigungen	128
4.5	Realisierung der Benutzeroberfächer	129
4.5.1	QML	129
4.5.2	Qt-Quick	129
4.5.3	Slots und Signals	129
4.5.4	Bridge	130
5	Kommunikation	131
5.1	Hardware	131
5.1.1	CAN-Modul	131
5.1.2	Netzwerkstruktur	131
5.2	Listener	132
5.2.1	Konfigurieren der Schnittstelle	132
5.2.2	Empfangen der Daten	132
6	Fahrdatenspeicher	133
6.1	Datenbankstruktur	133
6.1.1	Benutzer System	133
6.1.2	Motor Daten	133
6.1.3	Fehler Tabelle	135
6.1.4	Akku Daten	135
6.2	Handler	135
6.2.1	Konfigurieren der Schnittstelle	135
6.2.2	Cursor	136
6.2.3	SELECT Befehl	136
6.2.4	INSERT Befehl	136
	VII Ergebnis	137
A	Allgemeines	138
1	Zeitplan	139
2	Kosten	140
B	Programmcode	141
1	Antriebsstrang	141
2	HICS	141
2.1	Benutzeroberfläche	141
C	CAD-Zeichnungen	142
1	Mechanik	143
2	Antriebsstrang	154

3	HCIS	156
D	Simulationen	157
1	Mechanik	157
E	Schaltpläne	172
1	Antriebsstrang	172
2	HCIS	174
F	Datenblätter	175
1	HCIS	176
1.1	Mean Well RSD-30H-5	176
1.2	Mean Well SD-350C-12	186
1.3	Raspberry Pi 4 Moddel B	189
2	Antrieb	202
2.1	Ashwoods Elektro-Motor IPM-200-50	202
2.2	Hochleistungs-Relais: KILOVAC LEV200 A4ANA	204
2.3	Hochgeschwindigkeits-Schmelzsicherung: FWA-400B	206
2.4	Leistungsdaten Hirschmann elektronischer Gasdrehgriff v1.0	208
2.5	VCL Common Functions	235
	Literaturverzeichnis	269
	Abbildungsverzeichnis	269
	Tabellenverzeichnis	271
	Codeverzeichnis	272

Kapitel I

Einführung

1 Projektteam



Martin Kronberger



Jakob Lackner



Simon Kern



Schmeisser Tobias

2 Projektbetreuer

Prof. Dipl.-Ing. Reinhold Benedikter

unterstützte Simon Kern bei der Entwicklung des Akku- und Ladesystems

Prof. Dipl.-Ing. (FH) Johannes Ferner

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

Prof. Dipl.-Ing. Adolf Reinhart, MBA

unterstützte Jakob Lackner bei der Entwicklung des Antriebssystems

Prof. Dipl.-Ing. Peter Lindmoser

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

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

Simon Kern

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

Tobias Schmeisser

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

Kapitel II

Einleitung

1 Motivation

Seit kurzem ist die E-Mobilität unter anderem ein neuer schulautonomer Schwerpunkt in der Abteilung Elektrotechnik, an der HTBLuVA-Salzburg. Passend zu diesem Schwerpunkt ergab sich die Idee ein Projekt zum Bereich E-Mobilität zu realisieren, um zukünftige Schüler daran zu interessieren. Die E-Mobilität umfasst Elemente aus der Mechanik, Elektrotechnik, Kunststofftechnik und Informatik, das Projekt sollte so viele wie möglich davon abdecken und darstellen können. Gleichzeitig soll das Projekt, eine Art von Prototyp für neu entwickelte Elektromotorräder sein und nicht nur als Abschlussprojekt dienen.

Das Umdenken der Gesellschaft von Verbrennungsmotoren auf Elektromotoren hat zur Folge, dass alltägliche Maschinen immer öfter mit Elektromotoren ausgestattet werden. Motorräder in dieser Größenordnung sind bis dato nicht vorhanden oder erhältlich. Es sind einige Firmen mit der Entwicklung dieser beschäftigt und es werden immer öfter neue Produkte präsentiert, doch diese Zeitalter ist erst am Anfang und benötigt noch Zeit.

Für die Herstellung eines zugelassenen Motorrades ist der Umfang einer Diplomarbeit, aus zeitlichen, sowie budgetären Gründen nicht realisierbar und aus diesem Grund Ist diese Projekt ein Prototyp eines vielleicht zukünftigen Elektro-Sport-Motorades, welches die Verwirklichung des Systems wiedergeben soll.

2 Zielsetzung

Ziel dieses Projektes war ein funktionierendes Motorrad, ohne Licht, Hupe oder sonstigen Zusatzfunktionen. Es sollte einfach das System funktionieren, welches die Fortbewegung ermöglichen soll. Das Betriebssystem soll Funktionsfähig und die Versorgung über die Akkus sichergestellt sein.

Am Ende soll ein fahrtüchtiges Elektromotorrad mit den selben Fähigkeit, wie das Original das Ergebnis sein. Das Gewicht sollte wenn möglich reduziert, aber niemals erhöht werden. Die Zulassung wird in den darauffolgenden Jahren mit der Erweiterung von Licht, Hupe, und so weiter in der Studienzeit erlangt werden. Diese Projekt soll das Interesse des Projektteams an einer Entwicklung eines solchen Produkts bei Firmen zeigen, um aus diesen Prototypen einmal ein Produkt werden lassen zu können.

Die Allgemeinen Funktionen sind die allbekannten. Fahren mit Spitzen bis zu 130 km/h, Bremsen, ABS. Bei elektrischen Antrieben kann die Bremsleistung beim bergabfahren wieder in den Akku eingespeist werden. Diese Funktion nennt sich Rekuperation und wird bei Elektroautos schon verwendet und soll natürlich auch bei dem Elektromotorrad vorhanden sein.

3 Topologie des Gesamtsystems

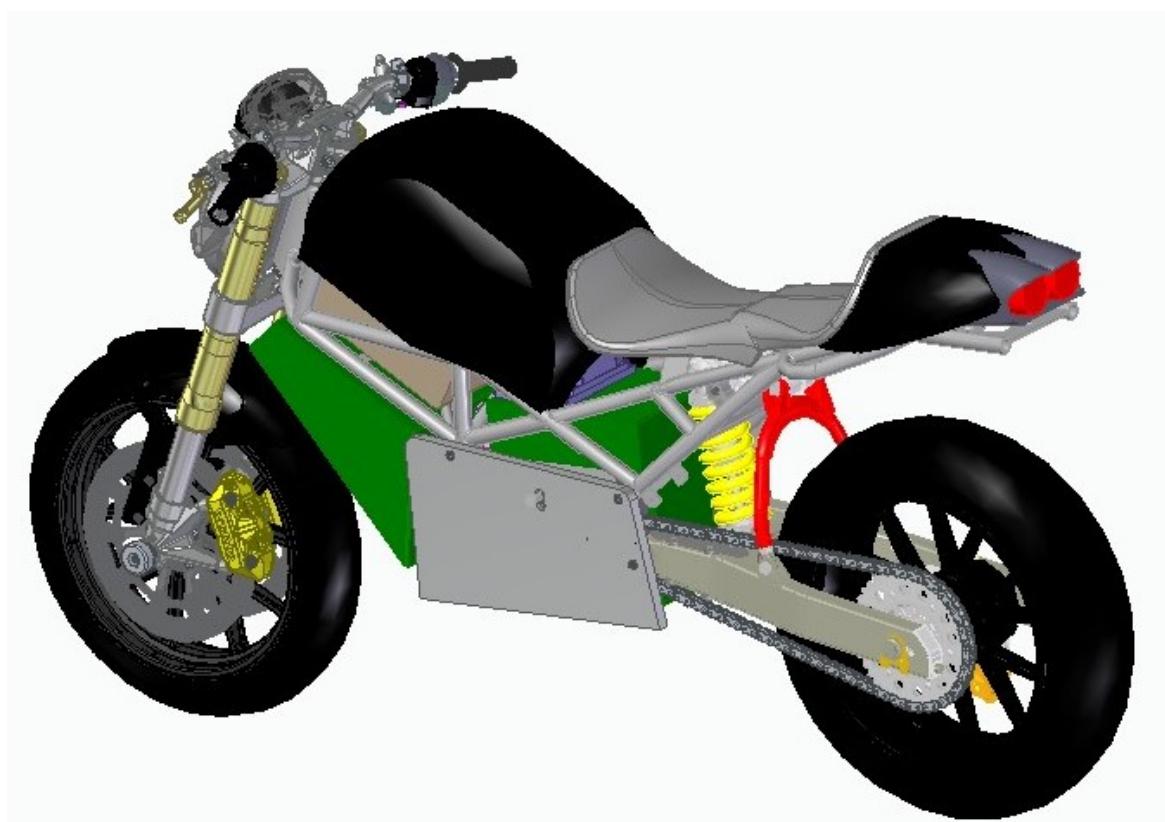


Abbildung II.1: Topologiekonzept 1

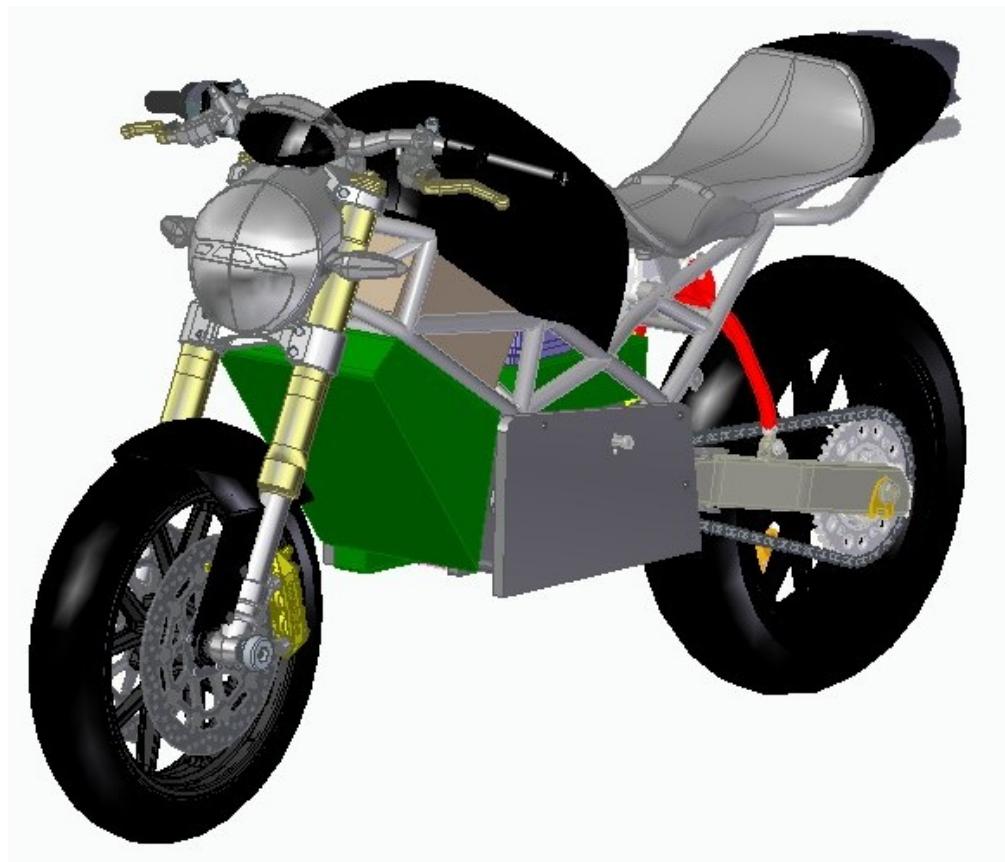


Abbildung II.2: Topologiebild 2

4 Leitfaden

Die folgenden Kapitel dieses Dokuments beschreiben die verschiedenen Aspekte des Projekts im Detail.

In Kapitel 1.5 ist der Stand der Technik, der verwendeten Technologien, sowie kleinere Auszüge von Normen vorhanden. Dieses Kapitel soll einen Überblick über die Technologien gegeben. Auf dieses Kapitel folgen die Kapitel zur Beschreibung des „Emissionslosen Sportmotorrades“.

In Kapitel IV wird als erstes auf die mechanische Umsetzung von Gehäuse, Getriebe, Akkupacks, Akkukühlung und der Zusammenbau dieser Komponenten eingegangen.

Kapitel V handelt vom Akku- und Ladekonzept, in dem eine Übersicht über die Aufgaben der Energieversorgung und des Batteriemanagement gegeben, sowie Die Energieversorgung dokumentiert ist.

Anschließend beschreibt Kapitel VI die Übersicht, den Hardwareaufbau, den Softwareaufbau und die Inbetriebnahme des Antriebsstranges und gibt Einblicke in die Entwicklung der Motorsteuerung.

Als letztes wird in Kapitel VII Das Human-Computer-Interaction-System Beschrieben. Diese Kapitel beinhaltet die Beschreibung der Zentralsteuerung mit Fahrdatenspeicher, der Benutzeroberfläche am Display und Kommunikation mit der Motorsteuerung, die für den Betrieb entwickelt worden sind.

In nachfolgenden Teilbereichen gilt in den einzelnen Kapiteln eine interne Referenzierung auf einen Unterpunkt in selbigem Kapitel – heißt: „siehe Abschnitt 2.3“ bedeutet, dass besagter Punkt innerhalb des Kapitels nachzuschlagen ist. Wird allerdings auf Abschnitte oder Abbildungen verwiesen, die sich in einem anderen Kapitel befinden, so wird dies beispielsweise als „siehe Kapitel Stand der Technik Abschnitt 2.1.3“ angegeben. Die Namen der Kapitel sind dabei stets mit kursiver Schrift hervorgehoben, um eine bessere Lesbarkeit zu schaffen.

Kapitel III

Stand der Technik

1 Kühlung

Bei dem Vorgang der Kühlung oder Abkühlung, wird einem System Wärme, oder thermische Energie entzogen. (Deshalb auch Entwärmung genannt) Unter Kühlung versteht man die Übertragung von Wärme einer technischen Komponente, an die Umwelt. Das Phänomen der Kühlung wird und kann auch beabsichtigt hervorgerufen werden, um bestimmte temperaturabhängige Eigenschaften erreichen und erhalten, aber auch Systeme vor Überhitzung schützen zu können.

1.1 Thermodynamische Grundlagen

Bei Feststoffen und Flüssigkeiten geht der Entzug von Wärme durch Wärmeübertragung entsprechend einem Temperaturgradienten vonstatten. Die wesentlichen Prozesse sind dabei Wärmeleitung und Wärmestrahlung, eingeschränkt auch die Konvektion. Diese Prozesse laufen spontan ab und deshalb, entsprechend der Gesetzen der Thermodynamik, einen Temperaturausgleich zur Folge haben. Diese Phänomene, kann für eine erwünschte Kühlung ausgenutzt werden, jedoch ist hierfür viel Energie notwendig.

Weil durch den Wärmeaustausch die Wärme nach Außen gelangt, hat dies eine Erhöhung der Temperatur, der Umgebung zur Folge. Dies bedeutet eine Umwandlung von Energieformen höherer Ordnung, in thermische Energie. Eine Kühlung im Sinne einer Reduzierung der thermischen Energie eines abgeschlossenen Systems ist daher nicht möglich, was sich in der Praxis zum Beispiel darin äußert, dass auch Kühlschränke letztlich die Temperatur (der Umgebung) erhöhen und nicht senken, wenn dies auch lokal der Fall sein mag.

Bei Festkörpern, ähnlich wie die elektrische Leitfähigkeit, gibt es auch bei der Wärmeleitung Einflussfaktoren. Diese sind hier Wärmeleitkoeffizient, Wärmeübergangskoeffizient und Wärmekapazität. Bei Flüssigkeiten spielt die Wärmeleitung und Wärmestrahlung ebenfalls eine Rolle, hinzu kommt jedoch die Konvektion als wesentlicher Prozess des Temperaturausgleichs. Bei Gasen dominiert die Konvektion.

1 2

¹vgl. [Kuehlung1] S.8-28
²vgl. [Kuehlung2]

1.2 Technische Anwendung

Kühlsysteme können nach dem verwendeten Wärmeträgermedium unterteilt werden. Die geläufigsten Arten der Kühlung sind:

- Wasserkühlung und
- Luftkühlung.
- Ölkühlung z. B. im Automotor und in Hydrauliksystemen (hydraulischen Antrieben)
- Natriumkühlung in Kernkraftwerken
- Kühlung durch Peltier-Elemente Kühlung von Prozessoren

1.3 Funktionsweise

Die Kühlung verwendet diese in den Thermodynamischen Grundlagen genannten Phänomene und führt diese künstlich herbei. Diese basiert auf der Wärmeleitung, vom zu kühlenden Körper mit einem Kühlstoff, durch Gas oder Flüssigkeit und des Abtransports, also der Wärmeströmung. Um Platz zu sparen, werden in manchen Anwendungen, als Abtransport, Heatpipes verwendet. Bei Verbrennungs- und Elektromotoren werden oft³ Flüssigkeitskühlungen verwendet.

1.4 Technische Kühlungsbeispiele

- Kühlsysteme von Kraftwerken und chemischen Prozessen
- Kühlung in der Klimatechnik
- Öl- und Ladeluftkühler im Turbodiesel-Motor
- Abgaskühlung in AGR-Systemen (zur Emissionsreduzierung (NOx))
- Wasserkühlung eines Automotors
- Luftkühlung eines Prozessors
- Kühlung für Steuerungen

3 4

³vgl. [Kuehlung1] S.8-28

⁴vgl. [Kuehlung2]

1.5 Wasserkühlung

Bei einer Wasserkühlung (Flüssigkeitskühlung), wird logischerweise Wasser als primäres Kühlmittel verwendet. Für die Wasserkühlung gibt es sehr viele verschiedene Einsatzmöglichkeiten. Diese Art von Kühlung wird in Elektro- wie auch in Verbrennungsmotoren, Hochöfen, Stromrichtern, Kraftwerken, Computern und vielen weiteren Anwendungen verwendet.

1.5.1 Wasserkühlung bei elektronischen Geräten

Röhren bestückte Endstufen werden seit fasst 100 Jahren mit einer Wasserkühlung gekühlt. Um Fehler von und deren Folgen, also Zerstörungen, zu vermeiden, wird bei dieser Art von Wasserkühlung kein normales Wasser verwendet, sondern nur destilliertes deionisiertes Wasser. Herkömmliches Wasser wäre bei dioesen großen Spannungen zu gefährlich. Bei einer Kühlung mit destillierten deionisierten Wasser, wird die Wärme durch einen Wärmeübertrager an einen zweiten Kühlkreislauf, welcher keine bestimmte Reinheit des Wassers besitzen muss, ab. Dieser Kreislauf ist von den hohen Spannungen so weit entfernt, dass keine Überschläge oder Kurzschlüsse mit dem Kühlwasser möglich sind.

Eine weiter Kühlungsmethode verwendet die Siedekondensationskühlung, welche bei Hochleistungsröhren eingesetzt wird. Hier sind Dampferzeugung und Kondensation räumlich nicht voneinander getrennt. Die Kühlflüssigkeit, also das Kühlwasser fließt durch einen Kühlkanal, welcher mit Nuten versehen ist. Die Nuten sind hin zur Anodeninnenseite gerichtet. In diesen Nuten entsteht durch die Erwärmung oder Erhitzung Wasserdampf. Der Wasserdampf gelangt nun in den Hauptkühlkanal, wo er durch Verwirbelung wieder kondensiert wird.

Dieser Vorgang geschieht bei Temperaturen von über 100°C und nützt zu gleich den Aggregatzustand flüssig zu gasförmig aus. Durch die hier benötigte Hitze für die Verdampfung, kann durch diese Kühlungsmethode auch bei kleinen Röhren große Wärmemengen abgeführt werden.

Neuere Modelle dieser Sender sind nicht mehr mit einer Wasserkühlung ausgestattet.

Auch in der Leistungselektronik wird eine Flüssigkeitskühlung mit Wasser verwendet. Beispiele für die Leistungselektronik sind Stromrichter für Hochspannungs-Gleichstrom-Übertragung oder auch Traktionsstromrichter in Schienenfahrzeugen.

⁵vgl. [Wasserkuehlung]

1.5.2 Wasserkühlung in Personal Computern

Sehr oft sind mittlerweile auch Wasserkühlungen in PC-Systemen zu finden. Diese Methode ist um einiges leiser und effizienter um einzelne Komponenten zu kühlen, als die herkömmlichen Luftkühlungen mit Ventilatoren. Mit der Wasserkühlung wird meist der Hauptprozessor gekühlt. Neben dem Hauptprozessor, können noch Grafikkarten, Hauptplatinenchipsätze, Festplatten, Netzteile mit dem Transformator, Spannungswandler und RAM-Bausteine in die Wasserkühlung mit einbezogen werden. Durch die Effizienz der Flüssigkeitskühlung mit Wasser und der immer weiter steigenden Leistungen, auch von Personalcomputern, ist um die Wasserkühlung bei Rechnern ein Markt entstanden.

Vorteile von einer Wasserkühlung sind unter anderem die schon erwähnte Effektivität der Kühlung der Hardware für Übertaktungsspielraum der CPU, durch die verbesserte Kühlung. Wie ebenfalls schon erwähnt funktioniert diese Kühlung sehr leise und beginnt nicht zu laut blasen, wenn eine große Rechenleistung vom PC verlangt wird. Bei der Wasserkühlung wird als Wärmeübertrager, ein Radiator, mit großen, langsam drehenden Lüftern oder ein passiver Radiatoren ohne Lüfter eingesetzt werden. Noch nicht genannt, erhöht sich mit einer Wasserkühlung die Zuverlässigkeit und Lebensdauer der durch die Kühlung, gekühlten Bausteine. Durch die Verwendung der richtigen Pumpe kann die Methode der Wasserkühlung eine sehr stromsparende sein

Nachteile bringt diese Methode natürlich auch mit sich. Die Wasserkühlung, hat einen erheblich größeren Installationsaufwand und damit auch vergleichsweise hohe Kosten zur Folge. Mit den Genannten Problemen kommt auch noch die Wartung hinzu, weil nicht nur der Ventilator geputzt werden müssen. Durch den Verzicht der Kühlung von einzelnen Komponenten, kann dies zur Überhitzung der selbigen führen. Mit hinzu kommt auch noch der erheblich größere Platzbedarf im Gehäuse.

Weil destilliertes Wasser, verglichen mit anderen Kühlungsflüssigkeiten bei Zimmertemperatur, den größten Wärmeleitkoeffizienten besitzt, ist diese Flüssigkeit die erste Wahl bei einer Flüssigkeitskühlung.

⁶vgl. [Wasserkuehlung]

1.6 Luftkühlung

Elektronische Bauelemente der Leistungselektronik, Verbrennungsmotoren und Klimaanlagen werden bei der Luftkühlung, durch die an den Oberflächen vorbei strömende Luft gekühlt. Die bei einer Luftkühlung benötigte Luftströmung kann durch verschiedene Arten erreicht werden. Unter andern durch Konvektion, Gebläse, also Ventilatoren, oder bei Fahrzeugen durch den Fahrtwind. Das Luftgekühlte Bauteil steht entweder so frei, dass es von Luft umströmt wird oder der Luftstrom wird mittels Kanäle um die Komponente geleitet. Oft sind freistehende Komponenten mit Kühlkörper aus einem Metall oder Kühlrippen als Wärmeübertrager versehen. Durch Rippen wird die Oberfläche vergrößert und dies hat einen größeren Wärmeabtransport zur Folge.

1.6.1 Luftkühlung bei Personalcomputern

Mit der Größe eines Prozessors, steigt auch die Wärmeentwicklung. In aktuellen PCs eingebaute leistungsstarke Mikrochips, erzeugen erhebliche Verlustwärmе. Diese Wärme wird meist durch eine Luftkühlung abgeführt, um eine thermische Überlastung der Chips zu verhindern.

Um die Wärmeentwicklung zu verringern, wurde die Betriebsspannung der Bauteile reduziert, weil die gesamte Aufnahmefähigkeit in Wärme umgesetzt wird. Ebenfalls werden bei normalen Betrieb nicht benötigten Bauteilen, die Taktfrequenz verringert oder die Bauteile auch ganz abgeschaltet um weniger oder keine unnötige Energie zu verbrauchen oder unnötige Wärme zu erzeugen. Trotz dieser Methoden, reicht die Abfuhr der Wärme durch Strahlung und Konvektion nicht mehr aus. Aus diesem Grund müssen Kühlkörper und Ventilatoren benutzt werden. Die Ventilatoren, sind am Rand des Gehäuses mit Luftkontakt nach Außen angebracht. Diese bewirken einen Luftstrom bei den zu kühlenden Komponenten und bläst die Warme Luft nach außen. Gleichzeitig wird kalte Luft von der anderen Seite angesaugt. Dies kann durch Luflöcher erzwungen werden, die an einem kühltechnisch sinnvollen Platz gewählt worden sind, damit alle Bauteile, die eine Kühlung benötigen, umströmt werden. Durch die mit gerippten Kühlkörpern vergrößerten Oberfläche der Bauteile ist die Kühlung ausreichend und ein zuverlässiger Betrieb ist möglich.

In einem Rechner wird die CPU am stärksten gekühlt. Neben der CPU werden auch, wie schon in der Wasserkühlung erwähnt, Prozessoren von Grafikkarten und auch das Netzteil mit Transformator mit gekühlt. Bei einer Luftkühlung natürlich mit der Luft.

Weitere Anwendung:

- HiFi-Verstärker Kunststoff

7 8

⁷vgl. [Luftkuehlung1]
⁸vgl. [Luftkuehlung2]

1.7 Aktive Kühlung

Bei einer aktiver Kühlung wird eine Kühlung durch eine Pumpe oder einem Lüfter künstlich erzwungen und wird eingesetzt, wenn die Abfuhr der Wärme durch Strahlung und Konvektion nicht mehr ausreicht.

1.7.1 Aufbau

Eine aktive Kühlung besteht meist aus den folgenden Komponenten:

- Kühlmittel
 - Wasser
 - Luft
 - Öl
 - speziellen Kühlflüssigkeit

Das Kühlmittel sorgt oft auch für die Schmierung der Bauteile.

- Antreibende Komponente
 - Lüfter
 - Pumpe

Diese Komponenten erzwingen einen Kühlmittelstrom. Bei einem Lüfter wird die Luft zu einem Luftstrom gezwungen und umströmt das zu kühlende Bauteil. Bei einer Pumpe fließt das Kühlmittel nicht nur zum zu kühlenden Objekt, sondern ebenfalls durch den Radiator, welcher die Flüssigkeit auf Ausgangstemperatur herunter kühlt.

Funktionsprinzip einer Aktiven Kühlung entspricht beispielsweise der Wasserkühlung.

Mittels einer Pumpe wird das Kühlmedium (Wasser, Kühlmittel) durch einen Kühlkreislauf, der aus Schläuchen oder Bohrungen (Nuten) besteht durch gepumpt. Am anderen Ende der Zubringleitung befindet sich das Objekt, welches gekühlt werden soll und wird von der Kühlflüssigkeit umflossen. Das Kühlmittel nimmt die Wärme auf und transportiert sie ab. Bei der Rückleitung fließt das Wasser durch den Radiator, welcher die Temperatur des Wassers reduziert und eine Erhitzung der Pumpe verhindert. Der Radiator gibt die Wärme durch Ventilatoren oder bei einem passiven Radiator durch Strahlung an die Umgebung ab. Dafür ist aber immer ein Temperaturgefälle notwendig, weil sonst eine Abgabe nicht möglich ist. und je größer das Gefälle ist, desto mehr Wärme kann pro Zeit abgegeben werden.

9

1.8 Passive Kühlung

Die passive Kühlung ist im Grunde jede natürliche Abkühlung von Körpern. Hier wird das Phänomen der Wärmekonvektion verwendet. Diese Kühlung wird verwendet um die Erwärmung eines Motors, oder einer Maschine und elektrischen Geräten an die Umwelt abzugeben. Der Effekt für die Konvektion kann durch Kühlrippen verstärkt werden. Bei jeder aktiven Kühlung, sind also auch passive Kühlungen vorhanden, weil man die Wärme noch nicht "abwringen" kann von einem Körper an die Umgebung abzugeben.

Eine passive Kühlung benützen zum Beispiel Kühlrippen oder Radiatoren. Diese Elemente bestehen aus Metall und geben die Wärme des wärmeren Mediums an das Kältere weiter. Um eine Beschleunigung zu erlangen, wird eine passive Kühlung mit eingesetzt.

10

⁹vgl. [AktiveKuehlung]

¹⁰vgl. [PassiveKuehlung]

2 Schutzarten Elektrischer Betriebsmittel

Die Schutzart eines Betriebsmittels gibt an, wie gut dieses geschützt ist vor:

- Zugang von Personen zu gefährlichen Teilen innerhalb eines Gebäudes
- Eindringen von fremden Festkörpern
- Eindringen von Wasser

Die Angabe der Schutzart erfolgt in Form eines Kurzzeichens (IP-Code / International-Protection-Code) Dieser Schutz entspricht den Regeln von ÖVE/ÖNORM EN 60529 (und IEC)

Die erste Kennziffer, gibt den Schutz gegen Berührung von gefährlichen Teilen innerhalb des Gehäuses und den Schutz gegen das Eindringen von festen Fremdkörpern an. Die zweite Kennziffer gibt den Schutz gegen Eindringen von Wasser an.¹¹

Schutzart	Betriebsmittelschutz	Personenschutz	Anwendung
IP6X	staubgeschützt	gegen eindringen von Draht	Deckelverschluss der Akkupacks
IPX7	zeitweilig wasserdicht	-	Gesamtes Akkupack
IPX8	dauerhaft wasserdicht	-	Akkupack-Korpus

Tabelle III.1: Auszug IP Schutzarten

¹¹vgl. [IPSchutzarten] S.63,64

3 Solid Edge



Abbildung III.1: Solid Edge Logo

Dank 1,5 jähriger Ausbildung in den ersten beiden Jahren in der HTBLuVA-Salzburg, konnte als Planungs- und Berechnungsprogramm, Solid Edge verwendet werden. Es wurden alle Teile des Motorrades in diesem Programm erstellt und geplant. Bei den Seitenplatten war noch eine Berechnung der Stabilität notwendig; um die Sicherheit prüfen zu können.

3.1 Erklärung

Solid Edge ist ein 3D-Zeichenprogramm aus dem Hause Siemens. Auf einfachsten Wege, können virtuell Teile erstellt, auf Zeichenblättern für die Fertigung gedruckt und mit anderen Teilen zu Geräten miteinander verbunden werden.

Mit der Funktion "DIN Metrisch Part" können Einzelteile erstellt und dimensioniert werden. In dieser Funktion sind einige Zeichenhilfen vorhanden, die einem das Zeichnen und planen erleichtern. Unter dem Abteilung "Volumenkörper" kann festgelegt werden, wie das Teil später einmal hergestellt werden soll. Beispielsweise wäre ein Rotationsausschnitt eine Fertigungsweise, einer Drehmaschine.



Abbildung III.2: Solid Edge Volumenkörper

Mit "Bemaßen" werden Maße des Formkörpers festgelegt.



Abbildung III.3: Solid Edge Bemaßen

In der selben Funktion, lassen sich auch Berechnungen durchführen. Unter der Registerkarte Simulationen, sind alle hierfür notwendigen Werkzeuge zu finden. Mit "Strukturelle Lasten", können an den gewünschten Punkten, beliebige Arten und Größen von Kräften angelegt werden.



Abbildung III.4: Solid Edge Strukturelle Lasten

Der nächste Schritt besteht darin eine "Vernetzung" durchzuführen, um die Berechnung zu ermöglichen. Durch die Vernetzung, wird die Kontur des Bauteiles vernetzt, welche bei der Berechnung des Verhaltes des Teiles auf die Kräfte, notwendig ist.



Abbildung III.5: Solid Edge Vernetzung

Mit "Berechnen", wird die Berechnung gestartet und das Ergebnis ausgegeben. Im Ergebnis können die Verschiebung des Teiles, auftretende Spannungen und viele weitere Ergebnisse abgerufen und auch mit "Animation", animiert beobachtet und abgelesen werden.



Abbildung III.6: Solid Edge Animation

Die Funktion "DIN Metrische Zeichnung", kann das Bauteil in 2D Ansichten, für Zeichnungen umgewandelt werden, um das Bauteil fertigen lassen zu können. Die umgewandelten Ansichten können bemaßt und geschnitten werden, um ein bestmögliches Verständnis der Fertigungsabteilung zu versichern. Mit "Ansichtsassistent" kann das gewünschte Teil ausgewählt und umgewandelt werden.



Abbildung III.7: Solid Edge Zeichnungsansichten

Mit der Funktion "DIM Metrische Baugruppe", können Einzelteile zu einem virtuellen Gerät zusammengebaut werden. Hier wird unter anderem ermöglicht Simulationen von Bewegungs-, oder Getriebeabläufen zu erstellen. Durch die "Komponentenmontage" werden Beziehungen zwischen Teilen festgelegt und fixiert.



Abbildung III.8: Solid Edge Zusammenbau

4 Akkusysteme

Verschiedene Speicher für elektrische Energie, die auf einer elektrochemischen Basis basieren, nennt man Batterien oder auch Akkumulatoren. Elektrochemische Speicher haben in den vergangenen Jahren immer mehr an Bedeutung gewonnen und werden auch in Zukunft immer öfter Gebrauch finden. Möglich wurde unsere heutige elektronische Mobilität erst mit der Erfindung der galvanischen Zelle. Seit dieser Erfindung, die mit Hilfe eines Stromkreises chemische Energie in elektrische Energie umzuwandeln, hat sich über die Jahrzehntelange Weiterentwicklung der Batterie einiges getan. Die Einsatzmöglichkeiten von Akkumulatoren sind extrem vielfältig. Kleine Lithium-Ionen Akkus werden zum Beispiel als Knopfzellen in Smartphones verwendet. Jedoch können sie auch bis hin zu großen stationären Energiespeichern für erneuerbare Energien benutzt werden. Wie bereits vorher erwähnt, sind elektrische Energiespeicher ein wichtiger Bestandteil für den Erfolg der Elektromobilität geworden. Es gibt unzählig viele verschiedene Arten von Batterien, die sich im chemischen Aufbau, ihrer Form und natürlich in ihren Einsatzmöglichkeiten unterscheiden. Durch die äußerst besonderen chemischen Eigenschaften und die vielseitigen Anwendungsbereiche, hat sich der Lithium-Ionen Akku durchgesetzt.

5 Batteriearten

5.1 Bleiakkumulator

Die ersten Versuche, einen auf Blei basierenden Akkumulator zu entwickeln, wurden am Anfang des 19. Jahrhunderts durchgeführt. Industriell wurde der Bleiakku interessant, als Forscher und Chemiker zusammen 1880 ein Verfahren entwickelten, bei dem der Bleiakkumulator bereits nach wenigen Ladezyklen, eine hohe Kapazität erreichte. Der erste technisch einsetzbare Bleiakkumulator wurde 1886 von Henri Tudor entwickelt. Dieser besitzt eine Zellspannung von ungefähr 2V (abhängig vom Ladezustand), was eine durchaus große Spannung für sogenannte wässrige Systeme ist. Der Ausdruck wässrige Systeme leitet sich von dem Elektrolyt ab. Bei Bleiakkumulatoren wird wässrige Schwefelsäure als Elektrolyt verwendet. Im entladenen Zustand bestehen beide Pole aus Blei(II)-sulfat ($PbSO_4$). Weiters besteht die Kathode aus Blei und die Anode aus Bleioxid. Bleiakkumulatoren sollten keinesfalls tiefenentladen werden, da dies zu Schäden führt und den Akku unbrauchbar macht. Ein extrem großer Nachteil ist das Gewicht, da nur 30 bis 40Wh/kg erreicht werden können. Diese Art von Akku zeichnet sich durch das kurzzeitige Zulassen hoher Ströme aus die zum Beispiel für Fahrzeug -bzw. Starterbatterien notwendig sind. Unter anderem sind 50 Prozent des Batterimarkets von Bleiakkumulatoren belegt. Wie vorher bereits erwähnt werden diese oftmals in Autos, LKWs oder auch Motorräder verbaut.

5.2 Nickel-Metallhybrid Akkumulatoren

Die technischen Grundlagen des Nickel-Metallhybrid Akkumulator wurden von Stanford R. Ovshinsky und Masahiko Oshitani ab 1962 bis 1982 zur marktreifen Zelle entwickelt. Seit dem Jahr 2006 sind spezielle NiMH-Akkumulatoren auf dem Markt, die sich gegenüber herkömmlichen NiMH-Akkus durch eine deutlich reduzierte Selbstentladung auszeichnen. Die positive Elektrode eines Nickel-Metallhybrid Akkumulators(NiMH) besteht aus Nickel(II)-hydroxid wogegen sich die negative Elektrode aus einem Metallhybrid zusammensetzt. Als Elektrolyt verwendet dieser Akkumulator eine Wasserstoffspeicherlegierung aus Nickel und seltenen Erden. NiMH-Akkus erreichen bis zu 80Wh/kg. Sie sind vielfach in den üblichen Bauformen von Standardbatterien verbreitet und liefern pro Zelle eine Spannung von 1,2V. Oftmals werden sie als wiederaufladbare Alternative der gängigen Alkalibatterien in haushaltsüblichen Geräten eingesetzt. Ein großer Vorteil gegenüber den Nickel-Cadmium Batterien ist es, dass der NiMH Akku nicht aus giftigen Cadmium besteht und er außerdem eine höher Energiedichte aufweist. Der Anwendungsbereich von NiMH Akkumulatoren ist sehr vielfältig. Vorzugsweise kommen sie wie NiCd Akkus überall dort zur Anwendung, wo ein hoher Energiebedarf besteht und hohe Batteriekosten vermeiden werden sollten. Typische Anwendungsbereiche sind zum Beispiel Foto- Videogeräte, Elektroautos, Elektrowerkzeuge und noch viele mehr. NiMH Akkus werden außerdem oft als Energiespeicher für Notbeleuchtungsanlagen verwendet.

5.3 Nickel-Cadmium Akkumulatoren

1899 wurde der Nickel-Cadmium Akku von dem Schweden W. Jungner entwickelt. NiCd Akkus zeichnen sich dadurch aus, dass sie einen eingebauten Ent- und Überladeschutz integriert haben. Das hat zur Folge, dass man keine aufwendige elektronische Schaltung durchführen muss. Als Material für die Kathode dieses Akkus verwendet man Nickeloxidhydroxid. Die Anode dagegen besteht aus dem giftigen Material Cadmium, welches jedoch eine äußerst hohe spezifische Ladung (478Ah/kg) besitzt. Bei Nickel-Cadmium Akkumulatoren besteht das Elektrolyt aus Kalilauge. Die typische Nennspannung ist exakt die selbe wie bei NiMH Akkus, 1,2V. Aus dieser Zellenspannung ergibt sich eine spezifische Energie von ungefähr 60Wh/kg. Eine Eigenschaft die man bei anderen Technologien nur selten antrifft ist das hervorragende Tieftemperaturverhalten von NiCd Akkus. Selbst bei einer Temperatur von -40°C ist eine Inbetriebnahme noch möglich. Im Jahr 2004 wurde jedoch die Verwendung von Nickel-Cadmium Akkus wegen dem giftigen Material auf medizinische und sicherheitrelevante Bereiche begrenzt. Diese Akkumulatoren sind in 2 verschiedenen Bauformen verfügbar, die sich durch die unterschiedlichen Anwendungsbereiche unterscheiden. Die offene Bauweise wird meist für Starterbatterien für Verbrennungsmotoren und Traktionsbatterien für Elektrofahrzeuge verwendet. Bei der anderen Bauweise, werden die Zellen gasdicht verschlossen. Oftmals werden sie für zentrale Stromversorgungssysteme für Notbeleuchtung verwendet.

5.4 Lithium-Ionen Batterie

5.4.1 Geschichte

Schon bereits in dem Jahr 1970 wurde von Jürgen Otto Besenhard und anderen das grundlegende Funktionsprinzip der Alkalimetallionen-Interkalation in Kohlenstoff-Elektroden sowie auch in oxidischen Elektroden erforscht und veröffentlicht. Ebenfalls wurde dabei die Anwendung in Lithium Batterien untersucht auch wenn zu der Zeit die praktische Anwendbarkeit als Elektroden für Lithium Batterien noch nicht erkannt wurde. Der erste auf dem Markt erhältliche Lithium-Ionen Akkumulator wurde von Sony im Jahr 1991 angeboten. Dieser Lithium-Cobaltdioxid Akku wurde in einer Videokamera verbaut. Die Batterie, die eine Spannung von 7,2V aufwies, bestand aus zwei seriell verschalteten Zellen und weiste etwa eine Kapazität von 1200mAh auf. Sogar bis heute wird diese Bauform von Akkumulatoren mit Kapazitäten bis zu 6900mAh angeboten und in äußerst vielen Geräten eingesetzt. Drei Physiker bzw. Chemiker (Whittingham, Goodenough und Yoshino) erhielten 2019 sogar den Nobelpreis für Chemie, für die Entwicklung der Lithium-Ionen Batterie. Forscher einer Universität fanden im Jahr 2020 heraus, dass durch die Zugabe von dem Element Kalium die Lithium Akkumulatoren langlebiger und sicherer werden. Außerdem verhindert das Kalium in dem Akku unerwünschte chemische Nebenreaktionen.

5.4.2 Allgemeines

Es gibt zahlreiche verschiedene Bauformen von Lithium-Ionen Akkumulatoren. Diese Batterien unterscheiden sich nicht nur in ihrer Größe und der Bauform, sondern auch in der chemischen Zusammensetzung ihrer Komponenten und haben unter anderem auch verschiedene Spannungsbereiche. Kenndaten wie Zellenspannung, Lade- und Entladeschlussspannung, Temperaturrempfindlichkeit und der maximal zulässige Lade- oder Entladestrom variieren bauartbedingt und sind wesentlich vom eingesetzten Elektrodenmaterial und den Elektrolyten abhängig. Eine Eigenschaft die alle Lithium-Ionen Akkumulatoren gemeinsam haben ist, dass sie gasdicht versiegelt sein müssen und außerdem lageunabhängig betrieben werden können. Die spezifische Energiedichte liegt ungefähr in der Größenordnung von 150Wh/kg und weist eine Energiedichte von 400Wh/l auf. Durch diese Eigenschaften findet diese Art von Batterie besonderen Einsatz in der mobilen Branche als elektrischer Energiespeicher. Ein weiteres wichtiges Merkmal aller Lithium-Ionen Akkumulatoren ist, dass sie Überladungen nicht verkraften können. Wenn man mehrere Zellen zum Beispiel in Reihe schaltet, um eine höhere elektrische Spannung zu erzielen, müssen zum Ausgleichen der Toleranzen in der Kapazität zwischen den Zellen meistens zusätzlich ein Batteriemanagementsystem (BMS) und ein Balancer vorgesehen werden.

5.4.3 Prinzip der Lithium-Ionen Batterie

Ein Lithium-Ionen Akkumulator erzeugt durch die Verschiebung von Lithium-Ionen eine elektromotorische Kraft. Beim Ladevorgang wandern positiv geladene Lithium-Ionen durch einen Elektrolyten hindurch von der positiven Elektrode zur negativen, während der Ladestrom die Elektronen über den äußeren Stromkreis liefert. Eine negative Elektrode aus Lithium-Metall ist elektrochemisch optimal, für einen Akku aber ungeeignet. Da sich die Elektrode beim Entladevorgang genauso wie bei einer Lithium-Batterie auflöst, besteht beim Ladevorgang keine Möglichkeit mehr, ihre Geometrie zu rekonstruieren.

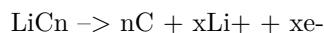
Aufbau:

Die negative Elektrode eines gänigen Lithium-Ionen Akkus besteht meist aus Graphit. Die positive Elektrode hingegen enthält meist Lithium-Metalloxide in Schichtstruktur wie Lithiumcobaltoxid (LiCoO_2). Der Lithium-Ionen Akkumulator muss wasserdicht sein, da es sonst zu einer Nebenreaktion zwischen dem Wasser (H_2O) mit dem Leitsalz (LiPF_6) zu Flußsäure (HF) reagieren kann. Das am häufigsten verwendete Elektrolyt in Lithium-Ionen Akkumulatoren besteht aus einer Mischung zwischen wasserfreien Lösungsmitteln (Ethylencarbonat, Propylencarbonat) mit Alkylcarbonaten/Äthern (Dimethylcarbonat, Diethylcarbonat) und natürlich mit Lithiumsalzen.

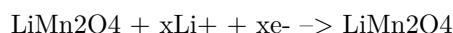
Beim Aufladen der Lithium-Ionen Akkus, d.h. anlegen einer äußeren Potenzials, fließen Lithium-Ionen zwischen die Graphitebenen ($n\text{C}$). Zusammen mit dem Kohlenstoff bilden diese Ionen eine Interkalationsverbindung ($\text{Li}_{x\text{n}}\text{C}$). Anders als beim Aufladen, wandern die Lithium-Ionen beim Entladen wieder in das Metalloxid und die Elektronen der Batterie können über einen äußeren Stromkreis wieder zur positiven Elektrode fließen. Ausschlaggebend für diese Interkalationsverbindung ist die Ausbildung einer schützenden Deckschicht auf der negativen Elektrode. Für die Lithium-Ionen ist diese Schicht durchlässig, jedoch die Lösungsmittelmoleküle können diese Deckschicht nicht durchdringen. Es kann passieren, dass diese Deckschicht nicht genügend ausgebildet worden ist. Das hat zur Folge, dass die Lithium-Ionen mit den Lösungsmittelmolekülen interkalieren, wodurch die Graphitelektrode stark beschädigt, oder sogar zerstört wird.

Reaktionsgleichungen:

- **Negative Elektrode (Entladung):**



- **Positive Elektrode (Entladung):**



- **Redox Gleichung:**

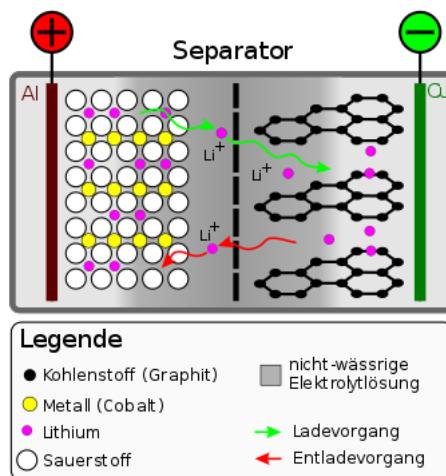
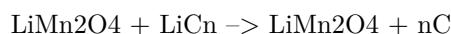


Abbildung III.9: Grundaufbau einer Lithium-Ionen Zelle [[GrundaufbauLiionenZelle]]

5.4.4 Lagerung und Sicherheitshinweise

Auch wenn Lithium nur als Li-Verbindungen in Lithium-Ionen Akkumulatoren vorhanden sind, sind die Komponenten eines solchen Akkus extrem leicht entzündbar, da Lithium ein hochreaktives Metall ist. Beim Überladen sind Ausgleichsreaktionen (z.B. die Zersetzung von Wasser), bei Lithium-Ionen Akkus nicht möglich, wie bei anderen Akkumulatoren dies der Fall ist. Schutzschaltungen die intern verbaut sein sollten, müssen eine solche Verpufferung verhindern. Anderfalls wird sonst die Funktionsfähigkeit des Akkus zerstört und er wird unbrauchbar. Jedoch kann es nicht nur zu inneren Beschädigungen kommen. Im Falle von inneren Kurzschlüssen, kann es passieren, dass mechanische Beschädigungen entstehen. Der hohe Kurzschlussstrom lässt zum Beispiel das Gehäuse schmelzen oder sogar in Flammen aufgehen. Es kann auch passieren, dass man den Defekt nicht unmittelbar erkennen kann. Doch kurze Zeit später kann es bereits zum Ausbruch eines Feuers kommen.

Lagerung:

Im Idealfall, sollten Lithium Ionen Akkus bei einem Ladezustand zwischen 40 - 60 Prozent kühl aufbewahrt und gelagert werden.

Sicherheitshinweise:

- Es ist wichtig, dass Lithium-Ionen Akkus nur mit passenden Ladegeräten aufgeladen werden. Schnell-Ladegeräte für Lithium Akkumulatoren können beispielweise eingesetzt werden. Man muss darauf achten, dass sie immer unter Aufsicht und möglichst nicht in der Nähe von brennbaren Materialien benutzt werden.
- Lithium-Ionen Akkus sind zwar hermetisch gekapselt, dennoch sollten sie unter keinen Umständen in Wasser getaucht werden. Besonders defekte und vollgeladene Lithium-Zellen reagieren meist heftig mit Wasser.
- Lithium-Ionen Akkumulatoren sind mechanisch sehr empfindlich. Durch einen internen Kurzschluss und einem Kontakt mit Luft können sie sich schnell entzünden.
- Eine Lithium-Ionen Zelle die in Flammen steht, wenn möglich mit Sand und nicht mit Wasser löschen, da dies zu einer heftigen Reaktion führen kann.
- Lithium-Zellen sollten niemals über 4,2V geladen und nicht unter 2,5V pro Zelle entladen werden. Außerdem dürfen Zellen niemals kurzgeschlossen werden. Bei einem Ladvorgang ist auf eine gute Wärmeabfuhr zu achten (nicht in die Sonne legen).
- Mehrere Lithium-Zellen sollten nur dann gleichzeitig geladen werden, wenn eine Schutzschaltung vorhanden ist.
- Die Elektrolytflüssigkeit ist brennbar. Sollte aus einer Zelle Elektrolytflüssigkeit austreten, diese am besten sofort ordnungsgerecht entsorgen.
- Man sollte versuchen die Lithium-Zellen bei einer Restkapazität von 20 Prozent nachzuladen.

5.4.5 Anwendungsbereiche von Lithium-Ionen Akkumulatoren

Lithium-Ionen Akkus versorgten anfangs hauptsächlich tragbare Geräte mit hohem Energiebedarf, für die herkömmliche Nickel-Cadmium- oder Nickel-Metallhydrid Akkus zu schwer oder zu groß waren, beispielsweise Mobiltelefone, Tablets, Digitalkameras, Camcorder, Notebooks, Handheld-Konsolen oder Taschenlampen. In der heutigen Zeit sind Lithium-Ionen Akkumulatoren fast in allen denkbaren Bereichen aufzufinden. In der Elektromobilitätsbranche dienen sie oftmals als Energiespeicher für Elektroautos, moderne elektronisch betriebene Rollstühle und auch für Hybridfahrzeuge. Auch im Modellbau haben sie schon früh Verwendung gefunden. Dadurch, dass Lithium Akkus ein deutlich geringeres Gewicht als andere Batteriearten aufweisen, sind sie in Verbindung mit bürstenlosen Gleichstrommotoren und den entsprechenden Reglern, gut als Antriebseinheit im Flugmodellbau geeignet. Schon seit Anfang des 21. Jahrhunderts, gibt es Lithium-Ionen Akkus auch in Elektrowerkzeugen wie zum Beispiel Akkuschraubern. Auch im Flugbetrieb haben diese Batterien Verwendung gefunden. In der Boeing 787 werden ebenfalls Lithium-Kobaltoxid-Akkus (LiCoO_2) verwendet. Zum Großen Teil werden Lithium-Ionen-Batterie-Systeme auch in Batterie-Speicherkraftwerken und Solarbatterien eingesetzt.

5.5 Batteriemanagementsystem

Batteriemanagementsysteme (BMS) sind elektronische Regelschaltungen, die Akkumulatoren oder Akkupacks auf Ladung und Entladung überwachen und ebenfalls regeln. Die Batteriekennwerte die man überwachen kann bzw. möchte, hängen oftmals von dem Projekt ab. Zu den häufigsten Batteriekennwerten gehören die Erkennung des Batterietyps, die Batteriespannung, die Spannung sowie die Temperatur einzelner Batteriezellen, die Akkukapazität, der Ladzustand, die Restbetriebszeit, die Stromentnahme und einige Kennwerte mehr. Die Hauptaufgabe von BMS-Systemen besteht darin, sicherzustellen, dass die Restenergie in einer Zelle optimal genutzt wird. Um keine Beschädigungen an Zellen zu bekommen, schützt das Batteriemanagementsystem die Batterien vor Tiefenentladung, vor Überspannung, vor zu schneller Ladung (begrenzen des Ladestroms) und ebenfalls vor einem zu hohen Entladestrom. Bei Akkupacks, d.h. Akkumulatoren mit mehreren Zellen, sorgt das Batteriemanagementsystem außerdem für ein sogenanntes Balancing, das sich darin ausdrückt, dass die verschiedenen Batteriezellen gleiche Ladezustände und Entladezustände haben.

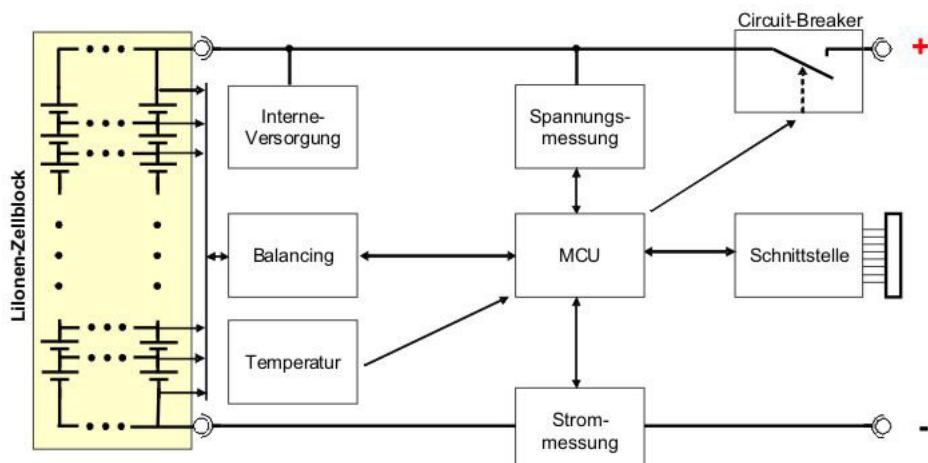


Abbildung III.10: Grundschaltung eines Batteriemanagementsystems[[GrundschaltungBMS]]

Grundsätzlich setzt sich ein vollständiges Batteriemanagementsystem aus folgenden Komponenten zusammen:

- **Cell Supervising Circuit (CSC):**

Die Aufgabe des CSC besteht darin die Zellen auf Spannung und Temperatur zu überwachen.

- **Kontrolleinheit:**

Die Kontrolleinheit berechnet die State of Charge (SOC), die State of Health (SOH) und überprüft auch die Funktionalität der Batterien (State of Function). Außerdem wird von der Kontrolleinheit auch der Ladeausgleich gesteuert und kommuniziert über eine serielle Schnittstelle mit einem Prozessor.

- **Circuit Breaker:**

Der Circuit Breaker trennt im Fehlerfall die Batterie von der Last. Dies erfolgt mithilfe eines HS-Kontaktors.

- **Strommessvorrichtung:**

Diese Komponente ist für die Messung des Stromes zuständig.

- **Temperaturüberwachung:**

Diese Komponente überprüft, ob sich die Batterien oder die Akkupacks in einem zulässigen Temperaturbereich befinden.

5.5.1 Komponenten eines BMS

Cell Supervising Circuit (CSC): Die erste Komponente eines Batteriemanagementsystem ist für die Spannungs- und Temperaturüberwachung der einzelnen Zellen zuständig und wird als Cell Supervising Circuit (CSC) bezeichnet. Ein Akkusystem besteht immer aus mindestens 2 einzelnen Zellen, oftmals jedoch aus mehreren. Deswegen ist die Spannungs- und Temperaturüberwachung jeder einzelnen Zelle nicht möglich. Es kommt durchaus vor, dass mehrere einzelne Zellen zu sogenannten Akkupacks zusammengeschraubt oder zusammengeschweißt werden. Man hat dann wiederum die Möglichkeit, jedes Akkupack für sich mithilfe eines CSC zu überwachen.

Kontrolleinheit: Die nächste Komponente wird auch als Kontrolleinheit bezeichnet. Die Aufgabe dieser Komponente liegt darin, die SOC (State of Charge) und auch die SOH (State of Health) zu berechnen. Außerdem steuert die Kontrolleinheit auch den Ladeausgleich der einzelnen Zellen oder der Akkupacks. Diese Komponente übernimmt auch die Kommunikation des Batteriemanagementsystems mit anderen angeschlossenen Einheiten. Diese Kommunikation erfolgt über eine serielle Schnittstelle wie den I2C-Bus oder dem CAN-Bus. Die Werte (SOC und SOH) und auch die SOF (State of Function) werden dann an einen Prozessor übermittelt auf dem die BMS-Software läuft und außerdem der SOC-Algorithmus implementiert ist. Die Kontrolleinheit ist fähig sich selbst in einen Ruhezustand zu versetzen um den eigenen Stromverbrauch um ein Minimum zu reduzieren.

- **State of Health (SOH):**

Beschreibt den aktuellen Alterungszustand der Batterie. Ein Kriterium dafür ist, welche Ladungsmengen die Zellen noch aufnehmen. Je älter die Akkumulatoren werden, desto weniger Aufnahmevermögen haben sie.

- **State of Charge (SOC):**

Beschreibt den momentanen Ladezustand der Batterie. Außerdem gibt er Auskunft darüber, wie viel beziehungsweise wie lange die Batterie noch Energie bereitstellt. Beim Aufladen der Akkus gibt es an, wie viel Energie er noch aufnehmen kann.

- **State of Function (SOF):**

Gibt die Funktionalität der Batterie an. State of Function beschreibt die Leistungsfähigkeit der Batterie. Also wie viel kW der Energiespeicher dem Motor zum Beispiel bereitstellen kann. Die Leistungsfähigkeit lässt mit zunehmendem Batteriealter nach.

Circuit-Breaker: Tritt ein Fehler bei einer einzelnen Zelle oder auch einem ganzen Akkupack auf, wird diese Batterie mithilfe eines HS-Kontaktors (Circuit-Breaker) von der Last getrennt. Dies schützt den Akkumulator. Dieser Kontaktor übernimmt außerdem noch die Trennung eines Akkumulators im Ruhezustand. Fehler die durch Trennen der Last beseitigt werden können sind zum Beispiel Kurzschlüsse oder auch Übertemperatur. Für den Kurzschlussfall sind meistens auch Schmelzsicherungen (eine Art Sollbruchstelle im Stromkreis; die Wärmewirkung des Stromes wird ausgenutzt) verbaut die verhindern, dass Leitungen oder auch das Gehäuse in Flammen aufgehen.

Strommessvorrichtung: Diese Komponente ist wie der Name schon beschreibt, für die Messung des Stromes zuständig. Oftmals werden dazu zwei voneinander unabhängige Systeme verwendet. Um den Strom messen zu können, wird ein Messsensor verwendet. Dies ist meist ein einfacher Widerstand. Bei der zweiten Methode wird der Strom über das elektromagnetische Feld gemessen.

Temperaturüberwachung: Die letzte Komponente, die benötigt wird, um das Batteriemanagementsystem zu vervollständigen, ist für den Temperaturausgleich zuständig. Das heißt, es wird überwacht ob sich der Akkumulator in einem zulässigen Temperaturbereich befindet. Sollte das nicht der Fall sein, können innere sowie auch äußere Schäden an der Batterie entstehen. Außerdem wirkt sich die Temperatur auf die Lebensdauer der Akkus auf.

5.5.2 Battery-Balancing

Der Ausdruck Balancing bezogen auf Akkumulatoren bedeutet so viel wie Ladeausgleich. Ohne Battery-Balancing bestimmt in einem Mehrzellen-Akku immer die schwächste Zelle darüber, welche Kapazität oder Spannung das Gesamtsystem aufweist. Das ergibt sich daraus, da sich jede Zelle minimal von einer anderen Zelle, durch ihre chemischen Struktur, unterscheidet. Einzelne Batteriezellen können auch unterschiedlich altern und deswegen kann man nie sicherstellen, dass jede Zelle exakt die identische Kapazität aufweist. Manche Zellen laden etwas schneller oder langsamer als andere. Wiederrum andere Batterien entladen sich etwas zügiger oder eben auch langsamer. In der Regel gibt es zwei verschiedene Arten von Battery-Balancing.

- **Passives Battery-Balancing**
- **Aktives Battery-Balancing**

Akkupacks:

Cluster oder Akkupacks bestehen zur Erhöhung der Nennspannung in der Regel aus mehreren in Reihe geschalteten Einzelzellen oder Zellblöcken. Fertigungs- und alterungsbedingt gibt es hierbei Schwankungen in der Kapazität, im Innenwiderstand und weiteren Parametern dieser Zellen. Die schwächste Zelle ist dabei bestimmend, wie viel geladen oder entladen werden darf. Im praktischen Einsatz von mehrzelligen in Reihe verschalteten Akkus führt dieser Umstand dazu, dass die Zellen in Reihe unterschiedlich geladen und entladen werden.

Es kommt dann im Verbund zu kritischer Tiefentladung oder bei der Ladung zu einer Überladung und Überschreiten der Ladeschlussspannung einzelner Zellen. Je nach Atyp kann es dabei zu einer irreversiblen Schädigung einzelner Zellen kommen. Die Folge: das gesamte Akkupack verliert an Kapazität¹². Um das zu verhindern, spielen im Batteriemanagementsystem die Balancer eine wichtige Rolle.

¹²[BalancingderPacks]

6 Synchronmaschine mit Dauermagneterregung

6.1 Allgemeines

Die permanenterregte Synchronmaschine benötigt keine zusätzlichen Komponenten für die Einspeisung des Erregersystems. Neben dem Wegfall der Schleifringe bzw. des Kommutators hat die PSM noch weitere Vorteile:

- Aufgrund der Erregung mit Permanentmagneten ist die Einspeisung einer flussbildenden Blindstromkomponente, wie bei der Asynchronmaschine bekannt ist, nicht notwendig. Somit ist der einzuspeisende Strom und damit der Energieverbrauch für vergleichbare Maschinen kleiner¹³.
- Weiteres erlauben die fehlende Läufernutung und die hohe Remanenzflussdichte der Dauermagneten höhere Luftspaltflussdichten, dadurch lassen sich auch höhere Leistungsdaten realisieren.

Verglichen mit Asynchronmaschinen haben permanenterregte Synchronmotoren also eine mindestens doppelt so große Bemessungsleistung bei gleicher Baugröße, zudem besitzen sie ebenfalls weniger Gesamtverluste. Die PSM müssen jedoch anwendungsspezifischer ausgelegt werden, denn es entfallen alle Einflussmöglichkeiten auf die Betriebsdaten, speziell auf Betrag und Phasenlage des Ständerstroms, die sonst durch die Änderung des Feldes über den Erregerstrom gegeben sind.

6.2 Aufbau

Um eine gute Ausnutzung des Bauvolumens realisieren zu können, wird der Ständer meist mit einer sechspoligen Drehstromwicklung ausgeführt. Der Läufer ist dabei wie in folgender Abbildung beschrieben mit Dauermagneten ausgeführt. Große Aussparungen im Läuferblech (2) reduzieren das Trägheitsmoment und führen zu besseren dynamischen Eigenschaften¹⁴.

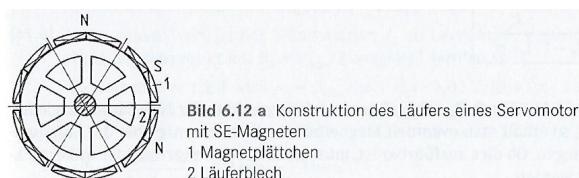


Abbildung III.11: Läufer einer dauerstärkerregten Synchronmaschine [Fischer]

6.3 Funktionsweise

Im Motorbetrieb wird an die Klemmen der Ständerwicklungen ein Drehstrom angelegt und dadurch ein sinusförmiges Drehfeld erzeugt. Da sich ungleiche magnetische Pole anziehen, wird der Rotor mit seinem zugehörigen Magnetfeld synchron mit dem angelegten Drehfeld mitbewegt. Im Leerlauf stehen sich also der Nordpol des Stator-Drehfelds und der Südpol des permanenterregten Rotor-Magnetfelds exakt gegenüber, die Drehzahl des Läufers ist also synchron mit der Drehzahl des angelegten Drehfelds. Bei bremsender Last eilt das Magnetfeld des Permanentmagneten dem Stator-Drehfeld um den Polradwinkel nach. Die Größe des Winkels stellt sich also abhängig von der Stärke des Lastdrehmoments ein. Dies kann jedoch nur bis zum sogenannten Kippmoment (Polradwinkel $\vartheta = 90^\circ$) gesteigert werden. Übersteigt das Drehmoment der Last das Kippmoment, so spricht man vom „Abreißen“ der magnetischen Felder und die Synchronmaschine bleibt stehen. Das selbe gilt für den Generatorbetrieb, hier eilt aber das Drehfeld des Rotors vor. Bei Frequenzumrichterspeisung mit Feldorientierung kann jedoch dauerhaft mit $\vartheta = +/- 90^\circ$ gefahren werden, man spricht von sogenannter Querstromspeisung, der Zusammenhang zwischen (Läufer)-Feld und (Ständer)-Strom entspricht dann jenem der fremderregten Gleichstrommaschine¹⁵.

¹³vgl. [Dissertation], S. 15ff.

¹⁴vgl. [Fischer], S. 348ff; S. 405

¹⁵vgl. [Synchronmaschine]

6.4 Auswertung der Antriebswelle (Encoder)

Für den Betrieb einer permanentmagneterregten Synchronmaschine wird immer ein zugehöriger Frequenzumrichter benötigt. Um eine gute Regelung einer Synchronmaschine gewährleisten zu können, benötigt dieser die Rückmeldung von einem Resolver (Encoder) oder einem hochauflösenden Inkrementalgeber, welcher im Motorgehäuse integriert ist. Denn im Betrieb mit sinusförmigen Strömen muss die exakte Läuferfeldlage bekannt sein, um eine feldorientierte Regelung durchführen zu können¹⁶.

Die Übertragung des elektrischen Signals erfolgt in Synchronmaschinen mit eingebauten Dauermagneten am häufigsten mit einem Sinus/Cosinus-Sensor. Der Sensor besteht hierbei aus zwei Versorgungsleitungen (positive Spannungsversorgung = 5V; Ground = 0V) und zwei Sensordrähten. Über die zwei Sensordrähte wird das Spannungssignal (0-5V) abhängig vom aktuellen Stand der Motorwelle ausgegeben. Betrachtet man die Spannungswerte über den gesamten Winkel der Motorwelle, erkennt man zwei sinusförmige Kurven, die um 90° versetzt sind. Bei einer korrekten Auswertung dieser Signale kann der aktuelle Stand der Motorwelle in jeder beliebigen Position absolut bestimmt werden.

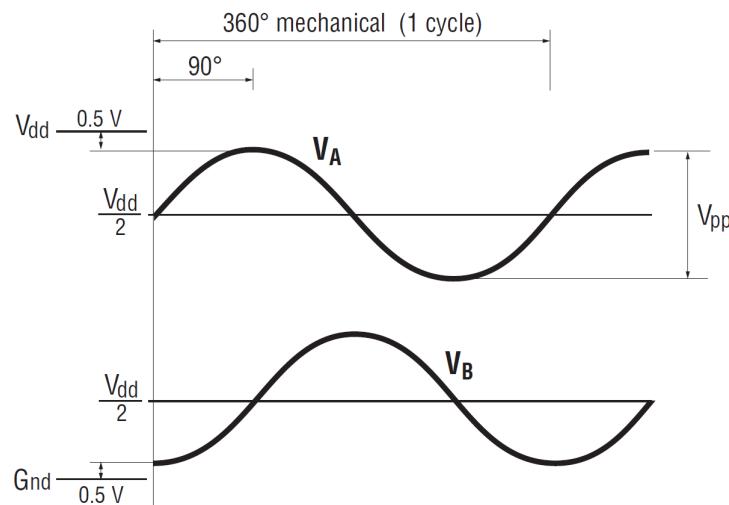


Abbildung III.12: Ausgabe eines Sinus/Cosinus-Sensors [Manual]

¹⁶vgl. [Fischer], S. 405.

7 Curtis Controller

7.1 Allgemeines

Der Curtis Controller ermöglicht eine genaue, zuverlässige und hocheffiziente Regelung von Drehzahl und Drehmoment bei Wechselstrom-Induktionsmotoren (ACIM) und Permanentmagnet-Synchronmotoren (PSM). Dieser Motorcontroller verfügt über zwei unabhängige Mikroprozessoren, um gleichzeitig eine außergewöhnliche Leistungsfähigkeit und funktionale Sicherheit gewährleisten zu können.

Der primäre Mikroprozessor führt eine fortschrittliche feldorientierte AC-Motorregelung durch, ebenso beinhaltet er einen weiteren integrierten Logik-Prozessor für eine zeitgleiche Operation durch die VCL-Software. Der zweite Mikroprozessor überwacht kontinuierlich den Betrieb des Gesamtsystems. Er misst Eingänge und kontrolliert die zugehörigen Ergebnisse, weiters überprüft er kritische Zeitpunkte und Operationen.

Die Vehicle-Control-Language (VCL) ist eine innovative Programmiersprache, welche von Curtis entwickelt wurde. Sehr viele einzigartige Funktionen sind in dieser Programmiersprache enthalten, weiters können bei Bedarf selbst Funktionen erstellt und konfiguriert werden. VCL eröffnet neue Möglichkeiten der Anpassung, sodass bestimmte fahrzeugspezifische Anwendungen schnell und einfach in der Motorsteuerung selbst erstellt werden können, um den Einbau von zusätzlichen Steuer-Modulen verhindern zu können.

Für die umfassende Einsatzmöglichkeit dieses Controllers sind ebenfalls sehr wichtige Kommunikationsmöglichkeit über den CAN-Bus gegeben. Ein- und Ausgänge können im gesamten System geteilt und optimal gemeinsam genutzt werden, wodurch die Verkabelung minimiert und zusätzliche Applikationsmöglichkeiten erstellt werden können, die häufig die Kosten des Gesamtsystems senken.

Dieser Motorcontroller ist eine ideale Lösung für Motorantriebe in der Anwendung für Elektrofahrzeuge, Hebevorrichtungen und Doppelmotorantriebe¹⁷.



Abbildung III.13: Übersicht der häufigsten Bauformen eines Curtis Controllers [Manual]

¹⁷vgl. [Manual], S. 1.

7.2 Feldorientierte Regelung

Die herkömmliche Technik zur Regelung von PSM wird als Sechsstufen- oder Trapezsteuerung bezeichnet. Dabei wird der Stator in einem sechsstufigen Prozess angesteuert, der Schwankungen des erzeugten Drehmoments hervorruft. Jedes Wicklungspaar wird einzeln erregt, bis der Rotor die nächste Position erreicht, dann wird der Motor zum nächsten Schritt kommutiert. Skalare Techniken sind jedoch für Anwendungen mit dynamisch wechselnden Lasten nicht präzise genug. Durch den Einsatz der feldorientierten Regelung lässt sich die Präzision, Effizienz und Lebensdauer des Antriebs erheblich steigern, weiteres lassen sich bei hohen und niedrigen Drehzahlen leistungsstarke Drehmomentregelungen realisieren¹⁸.

Die feldorientierte Regelung (auch Vektorregelung genannt) ist ein sinusförmiges Kommutierungsverfahren. Der Motorcontroller hat die Aufgabe die Statorströme zu berechnen, um den Rotor basierend auf der Stromrückkoppelung des Motors ansteuern zu können. Abgeleitet von der Raumzeigerdarstellung für Wechselspannungen und -strömen und deren Bezug zueinander wird ein rotierendes Koordinatensystem mit den Größen Drehmoment und magnetische Flussdichte als Achsen erstellt. Das Verfahren wandelt die Statorströme über dieses Koordinatensystem in magnetische Flussdichte und Drehmoment erzeugende Komponenten um. Zudem regelt es die Statorströme über einen PID-Regler und wandelt sie dann in drei Spannungswerte zurück, aus denen wiederum der PWM-Ausgang erstellt wird. Dieser ist dann auch für das sinusförmige Kommutierungssignal verantwortlich¹⁹.

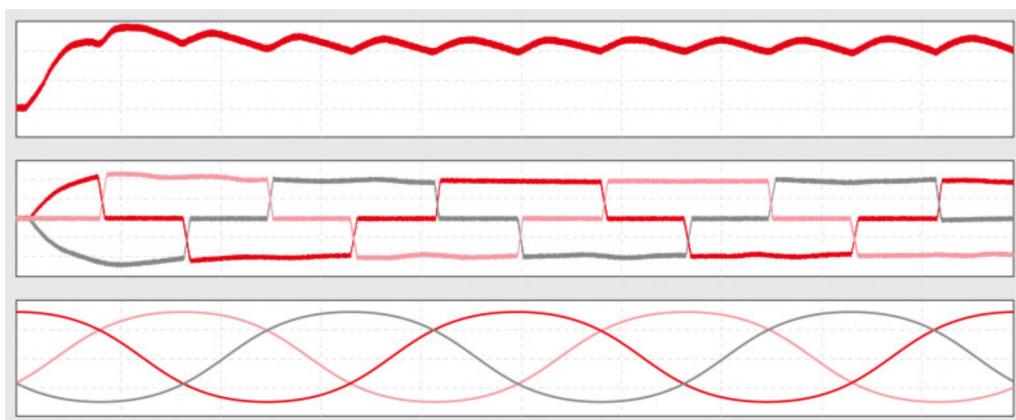


Abbildung III.14: Kommutierung mittels Trapezsteuerung [Feldorientierte-Antriebssteuerung]

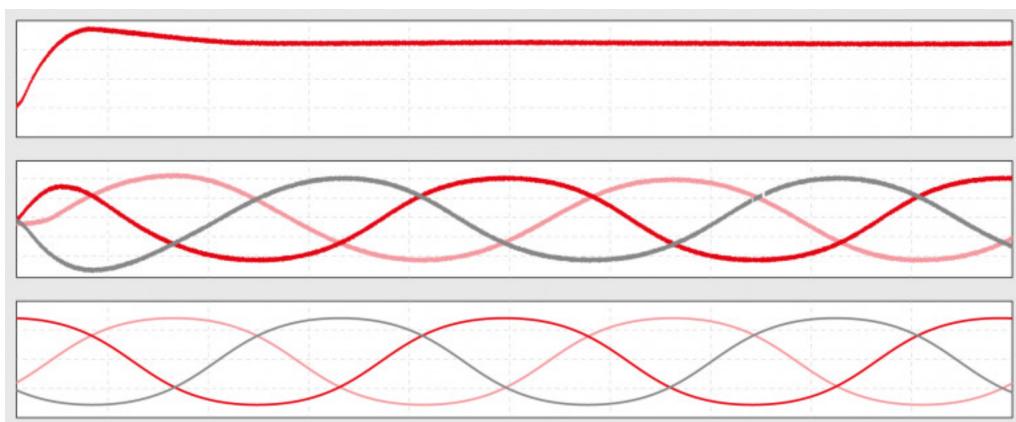


Abbildung III.15: Kommutierung mittels Vektorregelung [Feldorientierte-Antriebssteuerung]

¹⁸vgl. [Feldorientierte-Regelung]

¹⁹vgl. [Feldorientierte-Antriebssteuerung]

8 Der Regler

8.1 Einleitung

"Regelungen sind ein Bestandteil unseres Lebens und das nicht nur seit Erfindung der Dampfmaschine.

Allein schon der aufrechte Gang funktioniert nur mit Regelung. Dabei wirken die Sinne als Sensoren, das Gehirn als Regler und die Muskeln als Aktoren. Weitere Regelungen in unserem Körper sind z.B. die Konstanthaltung der Körpertemperatur, der Blutdruck, die Anpassung der Pupille auf Helligkeitsänderungen usw.

Der Begriff Regelung ist zu unterscheiden von dem im allgemeinen Sprachgebrauch oft synonym gebrauchten Begriff der Steuerung. Das Steuern ist ein rein vorwärts gerichteter Prozess ohne Rückkopplung. Die Ausgangsgröße wird dabei nicht überwacht und kann sich durch Störungen von außen verändern. Ein Beispiel ist die Steuerung eines Elektromotors mit einer einstellbaren Spannung. Durch Laständerungen wird sich die Drehzahl des Motors ändern. Soll nun die Drehzahl konstant gehalten werden, bedarf es einer Rückkopplung, um über die Spannung die Drehzahl anzupassen. Diese Rückkopplung ist das Kennzeichen einer Regelung.

Das Regeln ist ein Vorgang, bei dem die Ausgangsgröße, im Beispiel die Drehzahl, fortlaufend überwacht wird und bei Abweichung über die Stellgröße, im Beispiel die Spannung, korrigiert wird. Der sich dabei ergebende Wirkungsablauf findet in einem geschlossenen Kreis, dem Regelkreis, statt.

8.2 Der Regelkreis

Das Prinzip einer Regelung ist das fortlaufende Messen – Vergleichen – Stellen.

- **Messen**

Die Regelgröße wird mittels Sensoren gemessen.

- **Vergleichen**

Der Wert der Regelgröße wird mit dem Sollwert verglichen. Die Differenz ist die Regelabweichung.

- **Stellen**

Aus der Regelabweichung wird unter Berücksichtigung der dynamischen Eigenschaften der Regelstrecke die Stellgröße bestimmt.

Ein Regelkreis dient dazu, eine vorgegebene physikalische Größe, die Regelgröße, auf einen gewünschten Wert (Sollwert) zu bringen und dort zu halten, unabhängig von eventuell auftretenden Störungen. Um die Regelungsaufgabe zu erfüllen, muss der Augenblickswert der Regelgröße – der Istwert – gemessen und mit dem Sollwert verglichen werden. Auftretende Abweichungen müssen in geeigneter Art und Weise nachgestellt werden.

Um nun diese Aufgabe technisch zu lösen, gibt es die Regelungstechnik. Sie baut im wesentlichen auf die mathematische Beschreibung und Modellbildung des Systems Regelkreis. Zur Modellierung, Beschreibung und Simulation werden Blockschaltbilder mit diskreten Signalgliedern verwendet²⁰.

²⁰vgl. [Regelungstechnik]

Ein Regelkreis besteht entsprechend des vereinfachten Blockschaltbildes, wie es oft in der Regelungstechnik verwendet wird, aus den Hauptteilen Regler und Regelstrecke:

- **Regler G_R**

Um die Regelabweichung zu minimieren müssen Korrekturmaßnahmen ergriffen werden.

- **Regelstrecke G_S**

Das zu regelnde System

- **Führungsgröße w (Sollwert)**

Vorgegebener Wert, auf dem die Regelgröße durch die Regelung gehalten werden soll.

- **Regelgröße x (Istwert)**

Die Ausgangsgröße der Regelstrecke wird für den Vergleich mit dem Sollwert bzw. zur Berechnung der Regelabweichung rückgekoppelt.

- **Regelabweichung e**

Differenz zwischen Führungsgröße und Regelgröße $e = w - x$, bildet die eigentliche Eingangsgröße des Reglers.

- **Stellgröße y**

Die Ausgangsgröße des Reglers wird vom zu regelnden System beeinflusst.

- **Störgröße z**

Es können unvorhersehbare Störgrößen auftreten, die ebenfalls ausgeregelt werden müssen.

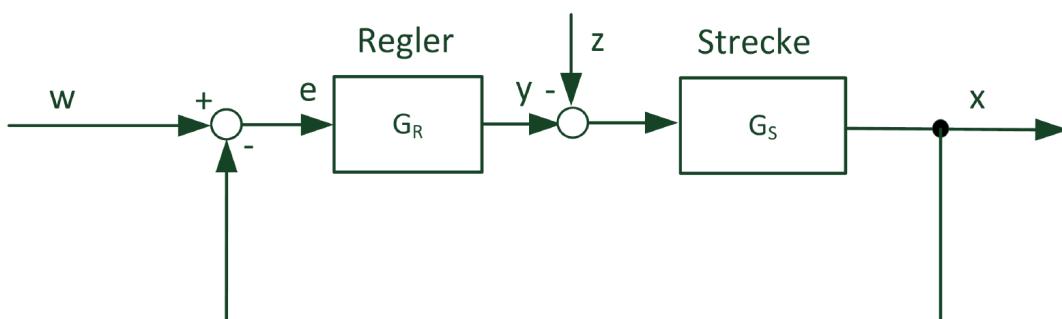


Abbildung III.16: Das Blockschaltbild eines allgemeinen Regelkreises [AUT5]

8.3 Der PID-Regler

Der PID-Regler vereinigt die guten Eigenschaften der anderen klassischen Regler. Der PID-Regler ist genau und sehr schnell, in den meisten Anwendungen kommt deshalb dieser Regler zum Einsatz. Vorab sollte man jedoch überprüfen, ob ein einfacherer bzw. billigerer Regler für die Realisierung der gewünschten Anwendung ebenfalls ausreicht.

Der PID-Regler besteht grundlegend aus drei einzelnen Reglern, die mit einander kombiniert werden:

P-Anteil

Der proportionalwirkende Regler multipliziert die Regelabweichung mit seinem Verstärkungsfaktor K_p und gibt das Ergebnis unverzögert weiter. Der P-geregelte Kreis ist einfach und mittelschnell im Vergleich zu anderen Regelungen. Das Problem hierbei ist die bleibende Regelabweichung!

$$\text{Reglergleichung: } y(t) = K_P \cdot e(t) \quad \text{bzw.} \quad G_R = K_P$$

I-Anteil

Der integralwirkende Regler summiert die Regelabweichung über der Zeit auf und multipliziert die Summe (d.h. das Integral) mit dem Faktor K_i . Je länger eine Regelabweichung ansteht, desto größer wird die Stellgröße des I-Reglers. Der I-geregelte Kreis ist langsam im Vergleich zu anderen Regelungen. Er hat aber den Vorteil, dass die Abweichung vollständig eliminiert wird.

$$\text{Reglergleichung: } y(t) = K_I \cdot \int_0^t e(t) dt \quad \text{bzw.} \quad G_R = \frac{K_I}{s}$$

D-Anteil

Der D-Anteil ist ein Differenzierer, der nur in Verbindung mit anderen Reglern eingesetzt wird. Er reagiert nicht auf die Regelabweichung, sondern nur auf deren Änderungsgeschwindigkeit. Je schneller sich das Signal ändert, desto größer ist die Ausgangsgröße, diese wird noch mit K_d multipliziert.

Ein Nachteil aller Regler mit D-Anteil kann die Unruhe im Kreis sein. Ist das Sensorsignal verrauscht, so wird dieses Rauschen durch das Differenzieren weiter verstärkt und wieder in den Kreis hineingegeben. Dadurch wird der Aktor stärker belastet. Solche Probleme können aber durch Anpassung der Regelparameter oder mit dem Hinzufügen eines Filters behoben werden²¹.

$$\text{Reglergleichung: } y(t) = K_D \cdot \frac{de(t)}{dt} \quad \text{bzw.} \quad G_R = K_D \cdot s$$

²¹vgl. [AUT5], Kapitel 4

Kombiniert man also die drei unterschiedlichen Regler miteinander, so erhält man den sogenannten PID-Regler, dieser vereint die guten Eigenschaften der einzelnen Regler. Der P-Regler ist für die schnellen Korrekturen zuständig, er reagiert ohne Verzögerung auf die Regelabweichung. Der I-Regler überwacht die Regelung des P-Reglers und korrigiert diese über die Zeit, um eventuelle Regelabweichungen auszubessern. Der D-Anteil verstärkt bei schnellen Veränderungen beide Regelkreise, um das Ansprechverhalten zu verbessern²².

Reglerleichung: $y(t) = K_P \cdot e(t) + K_I \cdot \int_0^t e(t) dt + K_D \cdot \frac{de(t)}{dt}$ bzw. $G_R = K_P + \frac{K_I}{s} + K_D \cdot s$

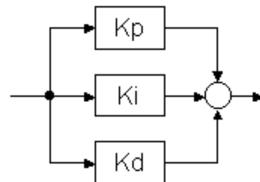


Abbildung III.17: Zusammensetzung des PID-Reglers [Regelungstechnik]

Hier sind für ein besseres Verständnis die Sprungantworten der unterschiedlichen klassischen Reglertypen dargestellt. Der Regelkreis ist geschlossen, ein PT2-Glied bildet die Regelstrecke.

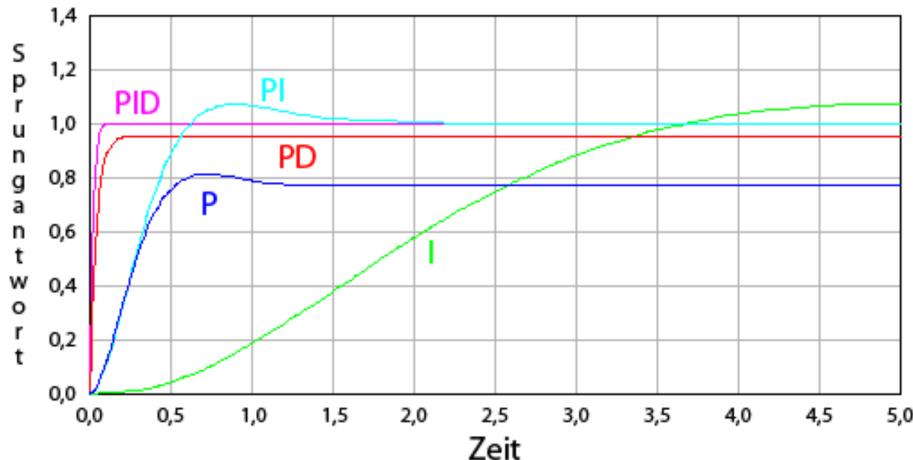


Abbildung III.18: Vergleich der verschiedenen Reglertypen [PID-Regler]

²²vgl. [PID-Regler]

9 Bussysteme

9.1 CAN-Bus

Controller Area Network ist eine serielle Netzwerktechnologie, die ursprünglich für die Automobilindustrie, insbesondere für europäische Autos, entwickelt wurde. Der CAN-Bus wird hauptsächlich in embedded systems verwendet und ist eine Netzwerktechnologie, die eine schnelle Kommunikation zwischen Mikrocontrollern bis hin zu Echtzeitanforderungen ermöglicht.

CAN ist ein Zweidraht-Halbduplex-Hochgeschwindigkeitsnetzwerksystem²³, das herkömmlichen seriellen Technologien wie RS232 in Bezug auf Funktionalität und Zuverlässigkeit weit überlegen ist und dennoch sind CAN-Implementierungen kostengünstiger.²⁴

Es gibt im Gegensatz zu dem SPI-Bus hier keinen Master und Slave. Zur Übertragung wird eine Twisted-Pair Leitung mit einem positiven und einem negativen CAN-Signal (CAN-High/CAN-Low) benötigt. Parallel dazu werden die Bus-Teilnehmen (CAN-Nodes) geschalten.

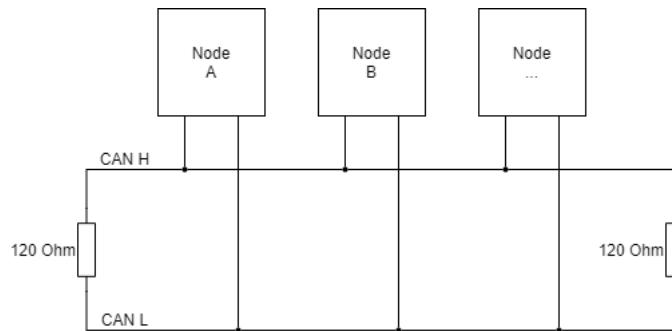


Abbildung III.19: Aufbau einer CAN-Bus Leitung

9.2 SPI-Bus

Der SPI-Bus (Serial Peripheral Interface) wurde von Motorola entwickelt, um eine synchrone serielle Vollduplex-Kommunikation zwischen Master- und Slave-Geräten zu ermöglichen. Er wird üblicherweise für die Kommunikation mit Flash-Speichern, Sensoren, Echtzeituhren, Analog-Digital-Wandlern und mehr verwendet.

Eine Standard-SPI-Verbindung umfasst einen Master, der über die Leitungen Serial Clock (SCK), Master Out Slave In (MOSI), Master In Slave Out (MISO) und Slave Select (SS) mit Slaves verbunden ist. Die SCK-, MOSI- und MISO-Signale können von Slaves gemeinsam genutzt werden, während jeder Slave über eine eindeutige SS-Leitung verfügt.²⁵

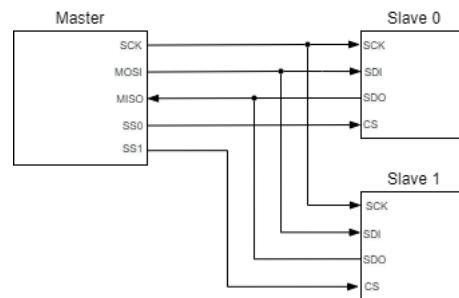


Abbildung III.20: Aufbau einer SPI-Bus Verbindung

²³Halb-Duplex und Voll-Duplex bestimmen, ob Hosts Daten gleichzeitig senden und empfangen können.

²⁴vgl. [can]

²⁵vgl. [spi], Abschnitt 1

10 Programmiersprachen

10.1 Python

Python ist eine interpretierte, interaktive, objektorientierte Programmiersprache. Sie enthält Module, Ausnahmen, dynamische Typisierung, dynamische Datentypen auf sehr hoher Ebene und Klassen. Der Code wird automatisch zu Bytecode kompiliert und ausgeführt. Daher eignet sich Python zur Verwendung als Skriptsprache, Implementierungssprache für Webanwendungen. Ebenso kann sie in C und C ++ erweitert werden, wodurch Python die Geschwindigkeit bereitstellen kann, die für rechenintensive Aufgaben erforderlich ist. Aufgrund seiner starken Strukturierungskonstrukte und seiner konsequenten Verwendung von Objekten und objektorientierter Programmierung ermöglicht Python das Schreiben klarer, logischer Anwendungen für kleine und große Aufgaben. Es verfügt über Integrierte Datentypen auf hoher Ebene: Zeichenfolgen, Listen, Wörterbücher, sowie die üblichen Kontrollstrukturen: if, if-else, if-elif-else, while und noch viel mehr. Ebenso gibt es eine sehr große Anzahl an Bibliotheken, welche sonst schwere Aufgaben sehr einfach gestalten.²⁶

```
print("HelloWorld")
```

Code Listing III.1: Hello World Programm in Pyhton

10.2 JavaScript

JavaScript ist eine benutzerfreundliche Skriptsprache für Objekte, mit der Live-Online-Anwendungen erstellt werden können, die Objekte und Ressourcen auf Clients und Servern miteinander verknüpfen. Während Java von Programmierern zum Erstellen neuer Objekte und Applets verwendet wird, wurde JavaScript für HTML-Seitenautoren und Entwickler von Unternehmensanwendungen entwickelt, um das Verhalten von Objekten, die entweder auf dem Client oder auf dem Server ausgeführt werden, dynamisch zu skripten. JavaScript ist kann von Personen mit wenig oder keiner Programmiererfahrung verwendet werden, um schnell komplexe Anwendungen zu erstellen. Das Design von JavaScript ist speziell für das Internet und die Erstellung netzwerkzentrierter Anwendungen entwickelt worden.²⁷

```
console.log('HelloWorld');
```

Code Listing III.2: Hello World Programm in JavaScript

²⁶vgl. [python], Abschnitt 1.1

²⁷vgl. [javascript], Einleitung

Kapitel IV

Mechanische Umsetzung

1 Gehäuse

1.1 Anforderung an das Gehäuse

Da das Grundprodukt eine Ducati S4 Monster 2001 mit Verbrennungsmotor war, musste der Motor entfernt werden, um die benötigten Teile für eine Umrüstung auf E-Antrieb unterbringen zu können. Bei allen Motorrädern ist der Motorblock ein Teil des Gehäuses beziehungsweise des Rahmens. Aus diesem Grund musste, statt des Verbrennungsmotors, ein platzsparender Ersatz gefunden werden. Das neue Gehäuse muss also die, bei einer Fahrt, auftreffenden Kräfte und Spannungen aufnehmen können. Die Schraubpunkte, mit denen der Rahmen des Motorrades zuvor zusammengehalten wurde, mussten ebenfalls im selben Bauteil ersetzt werden. Gleichzeitig mussten neue Schraubpunkte für Getriebe, Motor, Akkupacks, und weiteren Komponenten geschaffen werden, da sonst ein sicherer Betrieb nicht möglich wäre. Weiters sollte das neue Gehäuse ein Schlag- und Spritzschutz für die dahinter, im inneren liegenden Bauteile und Komponenten des Motorrades sein. Neben alldem soll natürlich auch der Fahrer nicht mit Körperteilen, oder sonstigen, am Körper anliegenden Gegenständen, in das Getriebe, oder beispielsweise die Motorsteuerung, gelangen können.

Zusammenfassend soll der Motorblockersatz folgendes übernehmen:

- Anfallende Kräfte in Normal-, sowie in Unfallsituationen verkraften.
- Schraubpunkte des Motorblocks ersetzen, ohne die das Motorrad in sich zusammenfallen würde.
- Neue Anschraubpunkte für Bauteile und Getriebe vorsehen.
- Schlag- und Spritzschutz für Bauteile und wichtige Komponenten, sowie Schutz für den Fahrer.

1.2 Dimensionierung

Die Dimensionierung richtete sich im Großen und Ganzen nach dem Platzbedarf der Bauteile und dem Gewicht, welches noch belegt werden darf, bis der Grenzwert erreicht ist. Nach der Wahl des Gesamtkonzepts beziehungsweise nachdem feststand, was alles für ein funktionstüchtiges Motorrad benötigt wird, könnte mit der Dimensionierung des Gehäuses begonnen werden.

1.3 Gewicht

Da die Akku-Zellen mit Abstand das schwerste am neuen Motorrad sind, musste, um nicht die Grenzwerte des vorhergehenden Motorrades zu sprengen, der Rest so Gewichtsparend wie möglich geplant werden. Anstatt von Stahl konnte durch Berechnungen auch Aluminium als ausreichendes Material ausfindig gemacht werden. Da Aluminium leichter als Stahl ist, wurde neben Kosten auch Gewicht gespart.

1.4 Gesamtgewicht des Zero-Emission-Power-Bikes:

Bezeichnung	Gewicht in kg
Akkuzellen	39,200
Akkupacks	3,269
Getriebe (ohne Seitenplatte)	13,031
Seitenplatte Links	5,067
Seitenplatte Rechts	5,821
BMS	1,100
Motorsteuerung	1,200
Akkusteuerung	0,700
Sonstiges (Schrauben, Leitungen,...)	2,900
Motorrad-Grundgerüst	79,000
Motor	13,000
Gesamt	164,288

Tabelle IV.1: Gewichtstabelle

1.5 Festlegung der Maße

Nach Absprache mit einigen Lehrern, Maschinenbau-Ingenieuren und dem Projektteam wurde als sinnvollster Ersatz für den Motorblock, zwei Platten ausgewählt. Die Platten enthielten zu Anfang nur die je 3 Bohrungen, als Ersatz für den Rahmen. Mit Fortlauf der Zeit und des Projektes wurden immer mehr Details, vor allem an der Linken Platte vorgenommen. Nach Fertigstellung der Seitenplatten musste das Material und die Stärke ausgewählt werden. Nach einigen statischen Berechnungen, war eine 20 mm Aluminiumplatte ausreichen für die anfallenden Kräfte. Mit der Entscheidung Aluminiumplatten zu verwenden, konnten Kosten, wie auch Gewicht gespart werden.

1.6 Material

Aluminiumlegierung:	3.3547 , AlMg4.5Mn, EN-AW 5083
Dichte:	2,660 kg/dm
Wärmeausdehnungskoeffizient:	0,0000 /c
Wärmeleitfähigkeit:	0,204 kW/m-C
Spezifische Wärme:	940,000 J/kg-C
Elastizitätsmodul:	70000,000 MPa
Poissonscher Beiwert:	0,390
Streckgrenze:	270,000 MPa
Zugfestigkeit:	345,000 MPa

Tabelle IV.2: Aluminium: AlMg4.5Mn Materialdaten

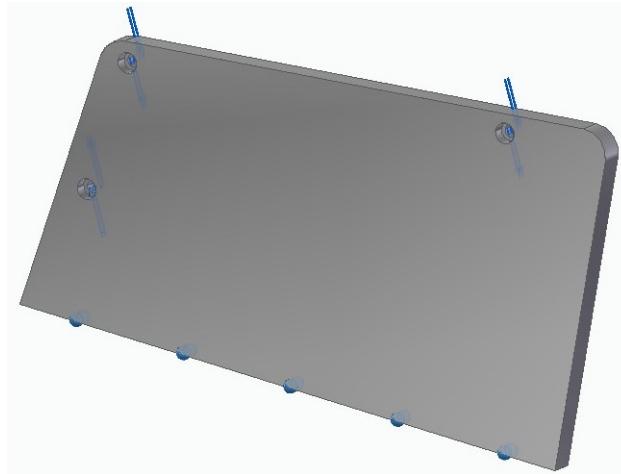


Abbildung IV.1: Seitenplatte Rechts

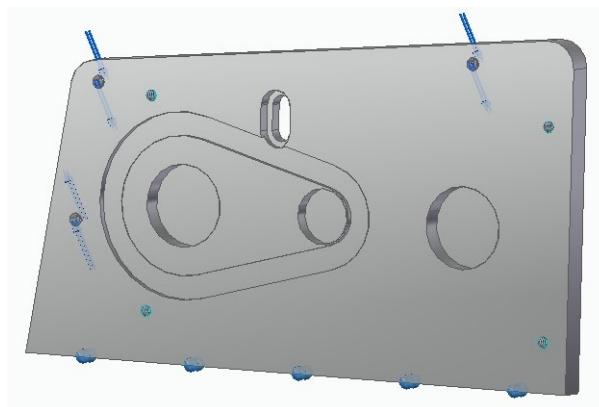


Abbildung IV.2: Seitenplatte Links Innenansicht

Neben den Platten, waren noch weitere kleinere Gehäuseteile nötig um den Rahmen neu in Betrieb zu nehmen. Dabei handelt es sich um den Achsenersatz für den Schwingarm, an dem das Hinterrad angebracht ist.



Abbildung IV.3: Wellenersatz

2 Getriebe

2.1 Aufgabe des Getriebes

Durch die hohen Umdrehungswerte des Elektromotors musste ein Getriebe diese herunterwandeln, weil ansonsten beim Start ein zu geringes Drehmoment vorhanden gewesen wäre und bei maximaler Drehzahl, eine theoretische Geschwindigkeit von gerundet 300 km/h, das Ergebnis gewesen wäre. Im Projekt wurde dieses Getriebe mit einem Kettengetriebe realisiert um ein Zahnradgetriebe zu vermeiden. Der Grund dafür wird unten näher erläutert.

2.2 Das Getriebe

Bei dem Getriebe wurde entschieden, dass kein Zahnradgetriebe verwendet werden wird, da ein solches in Öl getränkt sein muss und dafür die Getriebebox vollständig wasserdicht sein müsste. Stattdessen wurde ein Kettengetriebe als ausreichend empfunden. Nach reichlicher Recherche wurde klar, dass nicht nur eine zusätzliche Kette, sondern sogar zwei zusätzliche Ketten notwendig waren um ein Übertragungsverhältnis von 1: 9,25 zu erhalten. Die Zahnräder wären aufgrund der großen Drehmomente, der hohen Anzahl an Zähnen und des damit verbundenen Durchmessers zu groß gewesen. Somit entstand ein Fahrwerk mit insgesamt drei Ketten, sechs Kettenrädern und 2 Kettenspannern. Bei den Kettenrädern, sowie auch bei der Kette musste der Platzbedarf zu den auftretenden Kräften abgewogen werden. Die originale Kette ist eine aus der Normreihe B-1 10. Da aber in dieser Norm die Zahnräder größer sind, um größeren Kräften standhalten zu können, hätte ein größeres Getriebe eingeplant werden müssen. Nach Berechnungen könnte eine kleinere Norm ebenfalls als ausreichend befunden werden. Die maximalen Drehmomente in der Norm B-1 08, sind immer noch ausreichen für Extrem- und Alltagssituationen. Mit dieser Erkenntnis könnte wieder einiges an Platz gespart werden.

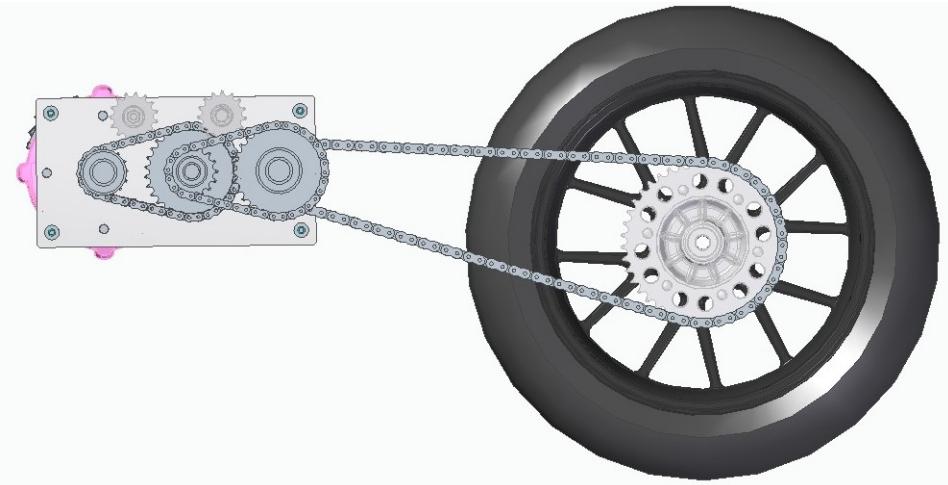


Abbildung IV.4: Getriebeansicht Seitlich

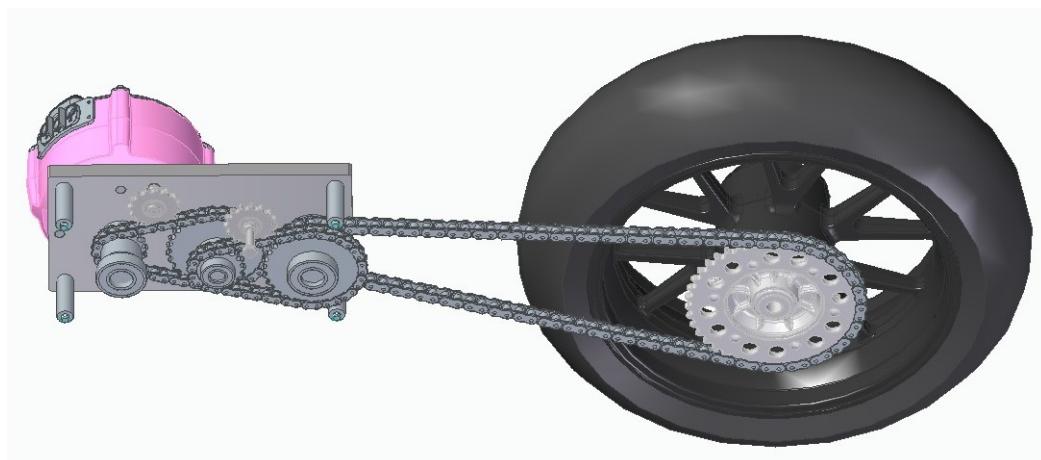


Abbildung IV.5: Getriebeansicht Schräg

Einzelteile

Die Zahnräder sind auf insgesamt 3 Achsen montiert, welche die Kraft von einem Kettenrad zum nächsten übertragen soll. Um dies zu gewährleisten, werden diese aus Stahl gefertigt, um das Risiko eines Reißens der Achse zu vermeiden. Die Zahnräder werden mit Passfedern und Schrauben an die Achse gepresst. Die Achsen sind Kugelgelagert in der linken Seitenplatte und der Aufbauplatte versenkt.

Achse 1 / Antriebsachse

Achse 1/Antriebsachse : Verbindet die Motorwelle mit dem ersten Zahnrad (15 Zähne B-1 08). Am anderen Ende befindet sich ein Kugellager.

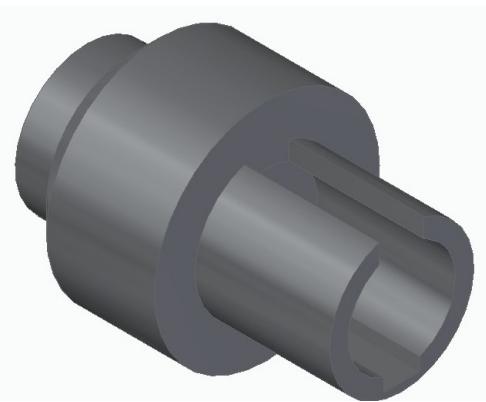


Abbildung IV.6: Achse 1/Antriebsachse

Achse 2

Am größerem Zylinder sitzt ein Zahnrad mit 30 Zähnen (B-1 08), welches mittels einer Kette mit dem ersten Zahnrad (15 Zähne B-1 08) von der 1. Achse verbunden ist. Über die Achse wird das Drehmoment und die Drehzahl auf das Nächste Zahnrad (16 Zähne B-1 08) übertragen. An beiden Enden ist jeweils ein Kugellager, wobei auf der rechten Seite, von dieser Perspektive aus gesehen, ein kleineres notwendig war, um den Kettenverlauf des Zahnrades nicht zu behindern. (Eine genauere Erklärung folgt.)



Abbildung IV.7: Achse 2

Achse 3

Dies ist die letzte Achse, bevor das Hinterrad angetrieben wird. Am größeren Zylinder befindet sich wieder ein 30-zähniges Kettenrad der Norm B-1 08 und am kleineren Zylinder ein 15-zähniges Zahnrad der Norm B-1 10. Bei dem Kettenrad mit 30 Zähnen war ein Eingriff in die

Norm-Bauform notwendig, um die Antriebskette nicht zu blockieren (Eine genauere Erklärung später). An beiden Seiten der Achse befindet sich jewils ein Kugellager der selben Bauform.

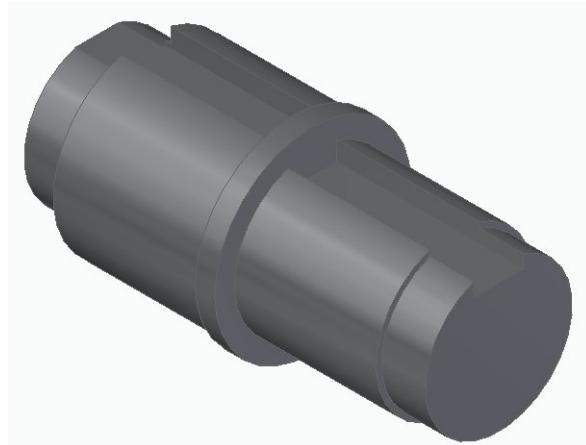


Abbildung IV.8: Achse 3

Getriebegegenplatte und Abstandhalter

Für das Getriebe war, wie schon erwähnt, eine Gegenplatte notwendig. An diese Platte wird auch der Motor und das mittlere Akkupack befestigt. Mit 4 Schrauben und dazugehörigen Abstandhalter wird das Getriebe zusammengehalten. Die Abstandhalter sind 4 Stahlrohre mit Innendurchmesser 10 mm und Außendurchmesser 20 mm.



Abbildung IV.9: Getriebegegenplatte



Abbildung IV.10: Abstandhalter

Kettenspanner

Wenn Ketten zu schwingen beginnen, werden diese sehr schnell kaputt und beeinträchtigen das Fahrverhalten. Um diese Problem zu lösen werden bei Kettenantrieben immer Spannräder benötigt. Bei den ersten beiden Ketten, vom Motor ausgehend, ist jeweils eines vorhanden. Bei der letzten Kette kann diese am Anschraubpunkt der Hinterachse gespannt oder entspannt werden und deshalb ist kein Spannrad notwendig. Die beiden Kettenräder haben 15 Zähne und stammen aus der Norm B-1 08. Die beiden Spannräder werden mittels einer Außensechskantschraube M16 und 3 Muttern M16 befestigt. Das, vom Motor aus, erste Kettenrad ist an der Aufbauplatte montiert, dort wo auch der Motor befestigt ist. Das zweite Kettenrad ist auf der Seitenplatte befestigt. Für die Befestigung sind an beiden Platten Nuten eingefräst.

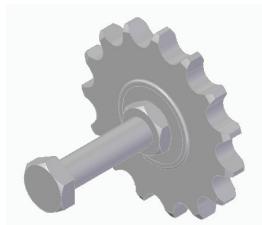


Abbildung IV.11: Kettenspanner

Kettenräder

Das Getriebe umfasst insgesamt fünf Kettenräder (ohne Antriebskettenrad an der Hinterachse) mit verschiedenen Größen, Bezogen auf die Anzahl der Zähne, aus zwei verschiedenen Normen, um das Getriebe mit an die originale Kette anpassen zu können:

ISO 08 B-1 Teilung 1/2 x 5/16"

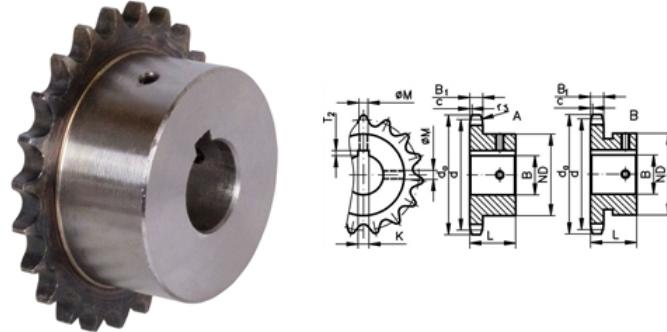


Abbildung IV.12: ISO 08 B-1

Werkstoff: Stahl C45. Zähne gefräst und induktiv gehärtet (ca. HRC 50). Einbaufertig, für diverse Wellendurchmesser. Fertigbohrung H7 -Rauwert Ra 1,6, Nut nach DIN 6885/1 auf Zahnspitze ausgerichtet, 2 Stellschraubengewinde, einmal auf Nut ausgerichtet, einmal 90° versetzt. Allgemeine Abmessungen: B1 = 7,2 mm, c = 1,3 mm, r3 = 13 mm. Weiter Daten sind aus der folgenden Tabelle zu entnehmen.

Produkt	Zähne	B H7	Type	da	d	ND	L	KH9	T2	M	Gewicht
		mm		mm	mm	mm	mm	mm	mm	mm	kg
10681532	15	32	B	65,5	61,09	49	28	10	3,3	M8	0,300
10581632	16	32	A	69,5	65,10	53	28	10	3,3	M8	0,334
10583038	30	38	A	126,1	121,50	80	30	10	3,3	M8	1,219

Tabelle IV.3: Auszug Wertetabelle ISO 08 B-1

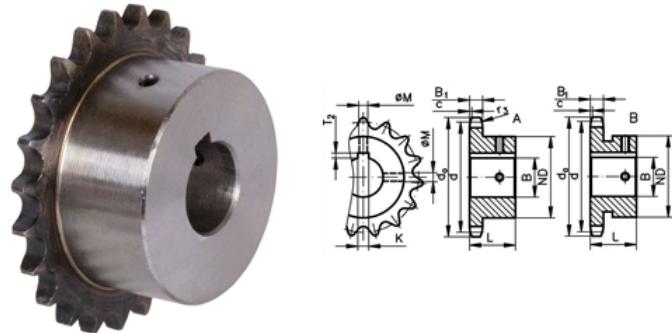
ISO 10 B-1 Teilung 5/8 x 3/8"


Abbildung IV.13: ISO 10 B-1

Werkstoff: Stahl C45. Zähne gefräst und induktiv gehärtet (ca. HRC 50). Einbaufertig, für diverse Wellendurchmesser. Fertigbohrung H7 -Rauwert Ra 1,6, Nut nach DIN 6885/1 auf Zahnspitze ausgerichtet, 2 Stellschraubengewinde, einmal auf Nut ausgerichtet, einmal 90° versetzt. Allgemeine Abmessungen: B1 = 9,1 mm, c = 1,6 mm, r3 = 16 mm. Weiter Daten sind aus der folgenden Tabelle zu entnehmen.

Produkt	Zähne	B H7	Type	da	d	ND	L	KH9	T2	M	Gewicht
		mm		mm	mm	mm	mm	mm	mm		kg
10681532	15	32	B	83,0	76,36	57	30	8	3,3	M6	0,501

Tabelle IV.4: Auszug Wertetabelle ISO 10 B-1

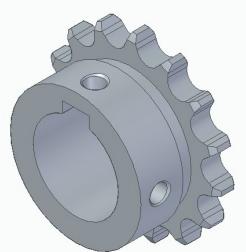


Abbildung IV.14: Kettenrad Z=15 B-1 08

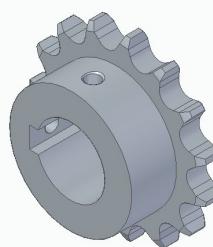


Abbildung IV.15: Kettenrad Z=15 B-1 10

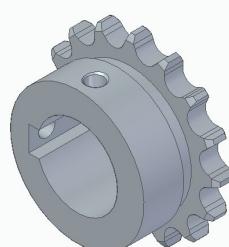


Abbildung IV.16: Kettenrad Z=16 B-1 08

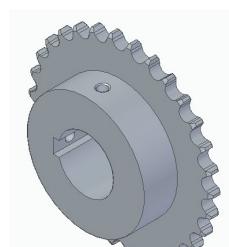


Abbildung IV.17: Kettenrad Z=30 B-1 08



Abbildung IV.18: Ducati Kettenrad Z=37 B-1 10

Kugellager

Kugellager: Innendurchmesser 20 mm Außendurchmesser 42 mm Stärke 12 mm



Abbildung IV.19: Kugellager 1

Kugellager: Innendurchmesser 30 mm Außendurchmesser 62 mm Stärke 16 mm



Abbildung IV.20: Kugellager 2

Vor Beginn der Planung, muss die Verfügbarkeit der Bauteile, durch Lieferanten und Firmen sichergestellt werden. Nach fixer Zusage von Lieferanten sowie verschiedenen Firmen wurde das Getriebe unter Hochdruck geplant. Da im Getriebe sehr hohe Kräfte aufkommen, muss garantiert sein, dass alles in der Genauigkeit eines Hundertstelmillimeter passt. Ebenfalls dürfen die Zahnräder nicht an Seitenplatten anstoßen oder streifen, weil ein enorm hoher Verschleiß die Folge wäre.

Das Getriebe ist eine Spezialanfertigung, welches für diese Projekt erstellt wurde. Alle Abstände von Zahnrad zu Zahnrad, sind so klein wie möglich. Die Befestigungspunkte für den Aufbau, sind ebenfalls mit dem geringsten Abstand zu den Kettenrädern angebracht. Durch sämtliche Optimierungen konnte Geld für das Material und Gewicht für das Motorrad gespart und dennoch ein sicherer Betrieb ermöglicht werden.

Seitenplatte mit Getriebe-Fräslungen

Diese ist die komplizierteste Platte bei diesem Projekt. Wie die rechte Seitenplatte, enthält diese auch die drei Bohrungen für die Rahmen- und Schwingarmaufhängung. Aber unter anderem ist diese auch Teil des Getriebes. Das Getriebe ist ausschließlich auf dieser Platte aufgehängt und trägt somit auch den Motor und einen Teil des mittleren Akkus. Es war ein Mittelweg zu finden, der es ermöglicht Kugellager, welche die Achsen beinhalten, Form-Fräslungen für den Verlauf von Ketten, weitere zusätzliche Bohrungen zu erstellen und noch die erforderliche Festigkeit für den Motorblock zu erhalten. Mit jeder Entfernung von Material vom Bauteil, wird seine Festigkeit verringert und damit die Gefahr eines Zusammenbruchs größer. Dieses Bauteil erforderte große Sorgfalt und genaue Planung um all dies zu vermeiden. Berechnungen (siehe Anhang).

Problemstellung des Getriebes

Wie oben schon erwähnt, musste eines der beiden 30-Zahn-Kettenräder bearbeitet werden. Wie im Bild zu sehen, würde ohne Bearbeitung die Kette zerstört werden, beziehungsweise würde das Getriebe nicht funktionieren. Es muss ein kleiner Teil weggedreht werden, um einen fehlerfreien Betrieb gewährleisten zu können.

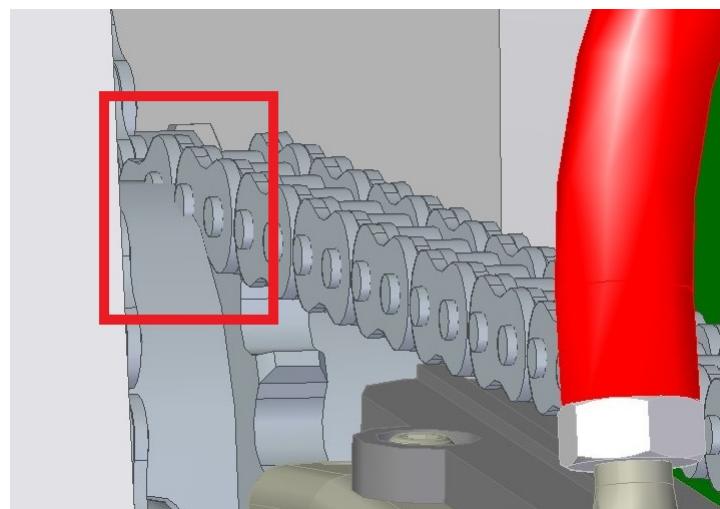


Abbildung IV.21: Getriebeproblem 1

Das zweite Problem war aufgrund eines Kugellagers, welches das selbe Problem zur Folge hätte, wie bei dem Kettenrad von oben. Die Lösung war ein kleineres Kugellager, welches einen kleineren Durchmesser hat, die Zahnscheibe des Kettenrades.

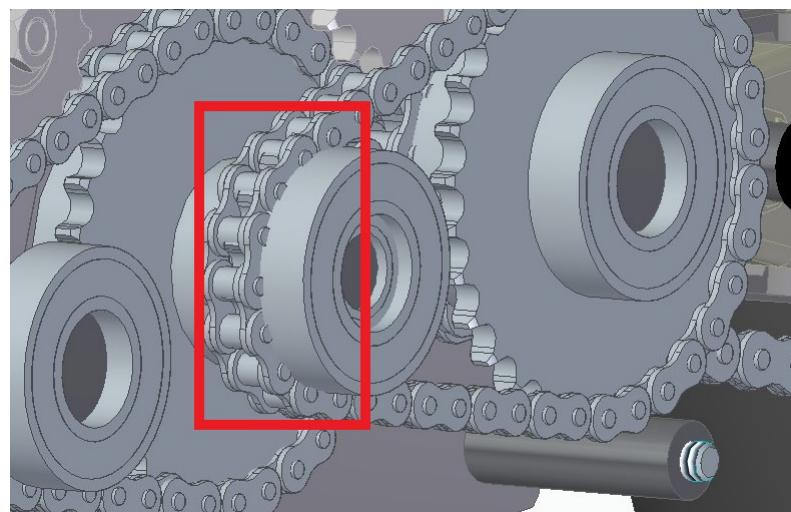


Abbildung IV.22: Getriebeproblem 2

3 Akkupacks

Als am besten geeignetes Material für die Packs ergab sich Polycarbonat. Dieses Material lässt sich verkleben und zurechtschneiden, ist dann aber doch nahezu unzerstörbar. Mit 6 mm Stärke ist das für dieses Projekt ausgewählte Material Schlagsicher. Mit 8 mm wäre es schon kugelsicher. Zudem wiegen diese Platten auch nicht sehr viel und sind daher ein perfektes Material. Die Packs sind so entworfen, dass sich ein Vielfaches von 40 Zellen einfach zusammenschließen lässt. Neben den Akkus muss natürlich auch eine Kühlmöglichkeit Platz finden. Das Design ist zu erst einmal darauf ausgelegt, dass so viele Zellen wie nur möglich in den Packs Platz haben, sportliches Aussehen oder Design ist zweitranig.

3.1 Akkupack Vorderseite

Die vordere Akkubox beinhaltet 240 Akkuzellen. Das Design richtet sich ein wenig nach der Originalform der Ducati S4 und bildet einen Übergang auf die Seitenplatten. Befestigt wird diese Box mit Schrauben nach oben weg. Die Steuerbox und eine weitere Platte bilden ein Gegenstück für Schrauben. Um sicher gehen zu können wird die Box noch von einem Metalgurt, der unten um dem Körper führt, welcher dann am Rahmen befestigt wird. Die Verbindung von Akku zur Akku- und Motorsteuerung wird mit Leitungen ermöglicht. Damit die Bohrungen für die Leitungen keine undichte Stelle wird, wird hier eine Kabelverschraubung verwendet um die Stelle abzudichten.

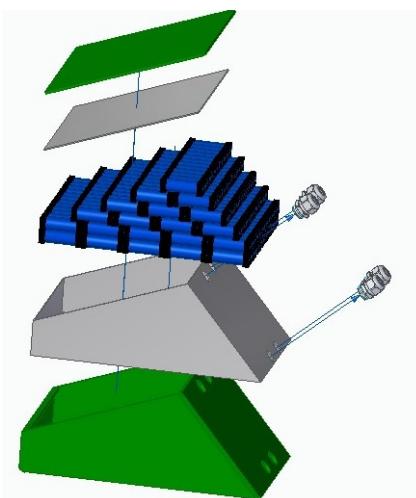


Abbildung IV.23: Akku Vorderseite

3.2 Akkupack Motorblock

Die große Akkubox in der Mitte beinhaltet 220 Akkuzellen, also um 20 zu wenig, oder zu viel. Diese 20 Zellen werden mit 20 Zellen aus der oberen liegenden Box verbunden um die gewünschte Spannung zu erreichen. Dieses Akkupack ist in den Freiraum zwischen den beiden Platten (Rechte Seitenplatte und Aufbauplatte vom Getriebe) und dem Motor eingepasst. Dieser Akku wird dann mit Schrauben an der linken und rechten Seite befestigt. Damit eine Verbindung zu dem vorderen Akku geschaffen werden kann, ist auch hier eine Kabelverschraubung notwendig, durch die die Leitung verlaufen soll.

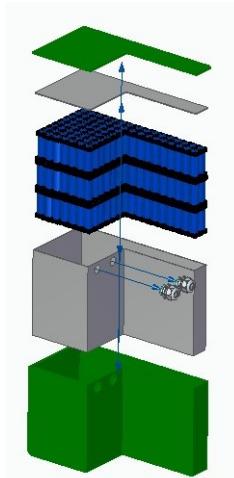


Abbildung IV.24: Akku Motorblock

3.3 Akkupack Mitte

Die kleine Akkubox in der Mitte des Motorrades, direkt unter dem Sitz, beinhaltet die oben schon erwähnten fehlenden 20 Zellen des großen Akkupacks und den Rest auf die Endsumme von 560 Zellen. Diese Box ist im Rahmen so eingepasst, dass der Platz am besten genutzt ist und der Stoßdämpfer, vom Schwingarm die Box nicht beschädigt. Die Verbindung vom oberen zum unteren Pack wird mit einer Bohrung hergestellt. Weil diese beiden Boxen gemeinsam verschraubt werden, müssen keine weiteren Dichtungsmaßnahmen vorgesehen werden.

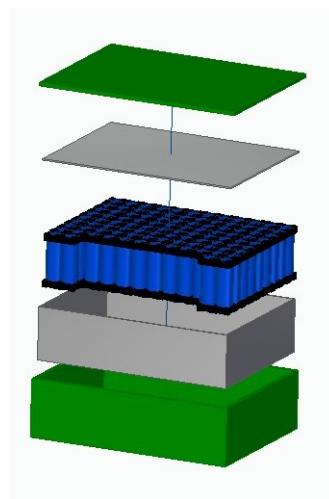


Abbildung IV.25: Akku Mitte

4 Akkukühlung

Für den Prototypen, wurde das Kühlungssystem so einfach, wie nur möglich gestaltet. Die ohnehin zu geringe Zeitvorgabe, war der Grund, warum eine Wärmeübertragungskühlung gewählt wurde. Diese Kühlungsmethode entspricht in etwa dem der Peltier-Elemente. Im Inneren der Akkupacks sind die Wände mit Aluminiumplatten ausgekleidet. Zusätzlich sind noch Platten mitten hindurch eingeplant, die verschiedene Teile des gesamten Packs unterteilen. Sämtliche Aluminiumplatten, sind thermisch und somit auch elektrisch, miteinander verbunden. Mithilfe von Leitfähigen Leitern, werden diese Kühlungen mit Teilen außerhalb des Akkus verbunden. Teile, die am Motorrad angebracht sind, werden vom Fahrtwind umströmt und dadurch gekühlt. Dieser Effekt, wird bei diesem System ausgenutzt. Die Wärmeenergie der Akkuzellen, soll also an die Umgebung abgegeben werden. Bei zukünftigen Modellen, wird ein Flüssigkeitssystem in Form einer Wasserkühlung verwendet werden, um die bestmögliche Kühlung gewährleisten zu können.

Siehe Akkukühlung im Motorblock-Pack

Siehe Akkukühlung im Mittel-Pack

Siehe Akkukühlung im Vorderseiten-Pack

4.1 Warum eine Kühlung notwendig ist

Wenn ein Akku eine gewisse Größe erreicht, muss dieser gekühlt werden, um nicht zu überhitzen. Wenn eine Akkuzelle Leistung abgibt, erwärmt sich diese. Die logische Schlussfolgerung ist nun, je mehr Akkuzellen vorhanden sind, desto größer ist die Erwärmung in diesem Raum. Es würde mit dem Fahrtwind eine Luftkühlung völlig ausreichend sein, nur die Packs sind in Boxen untergebracht. Durch die zusätzliche Hülle, um die Zellen vor Wasser, Steinschlag, oder Ähnlichem zu schützen, kann eine Luftkühlung nicht realisiert werden. Gleichzeitig wirkt die Box wie eine Isolierung, wodurch sich die Packs immer schneller erhitzen. Wenn nun keine Kühlung verwendet wird, erhitzt sich das Innere so lange, bis sich sämtliche Zellen aufzulösen beginnen. Das Polycarbonat würde extrem beansprucht werden. Wenn die Zellen immer weiter Leistung abgeben, kommt es bei Lithium-Ionen-Zellen schlussendlich zu einer Explosion. Gegen solche Fälle ist im Betriebssystem ein Warnsystem vorhanden. Ebenfalls sind Schutzeinrichtungen wie Sicherungen eingebaut, die ein solches Szenario verhindern sollen. Weil keine 100 prozentige Sicherheit für alle Fälle besteht, ist das Material, wie oben schon erwähnt, ebenfalls auf Extremsituationen ausgelegt und kann die Wärmeentwicklung für einen kleinen Zeitraum standhalten, um im schlimmsten Fall ein Entfernen vom Motorrad zu ermöglichen.

5 Zusammenbau

Die beiden Seitenplatten, sind mit jeweils 3 Sechskantschrauben mit Rahmen und dem Schwingarm verbunden. Die Schrauben werden mit Sicherheitsmuttern und einer Zulegscheibe festgeschraubt. Bei dem Ersatzteil für die Welle, ist ein Gewinde integriert.

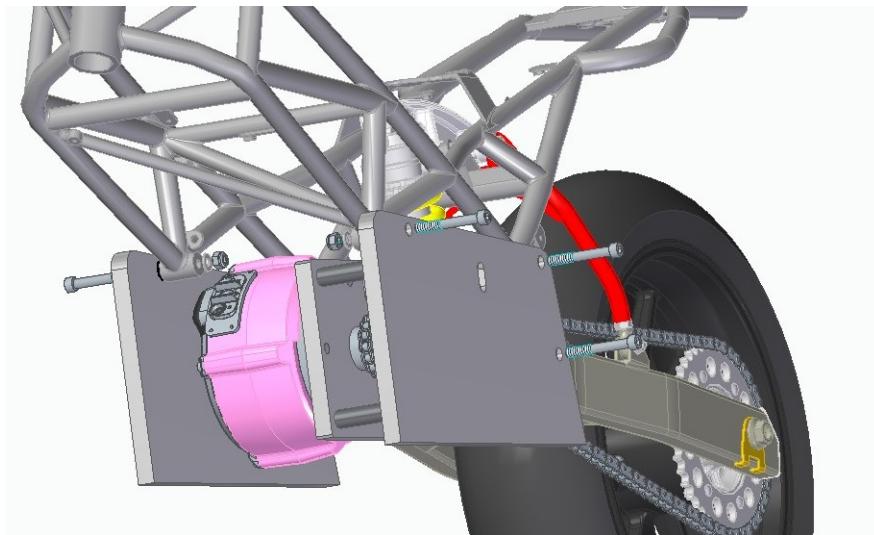


Abbildung IV.26: Befestigung der Seitenplatten

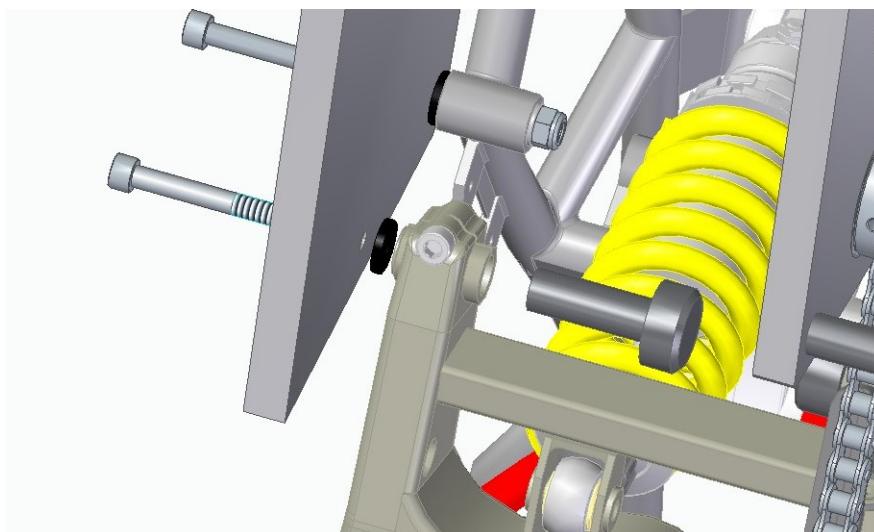


Abbildung IV.27: Befestigung des Schwingarmes

In den folgenden Bildern ist das Getriebe mit all seinen Einzelteilen abgebildet. In der Mitte die drei Achsen. Jeweils zwei beziehungsweise ein Kettenrad links und rechts der Achsen und zwischen den Kettenrädern und den beiden Aluminiumplatten noch Kugellager. sämtliche Schrauben, die das Getriebe zusammenhalten sollen, sind auch sichtbar gemacht worden.

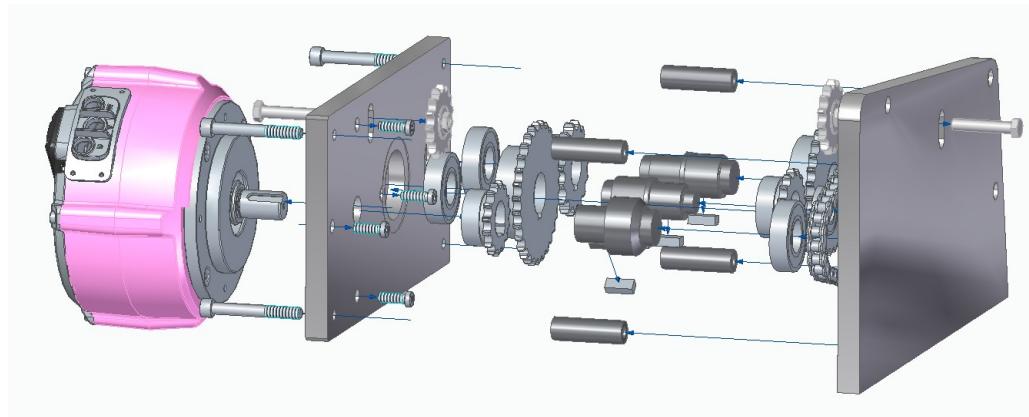


Abbildung IV.28: Explosionsansicht des gesamten Getriebes

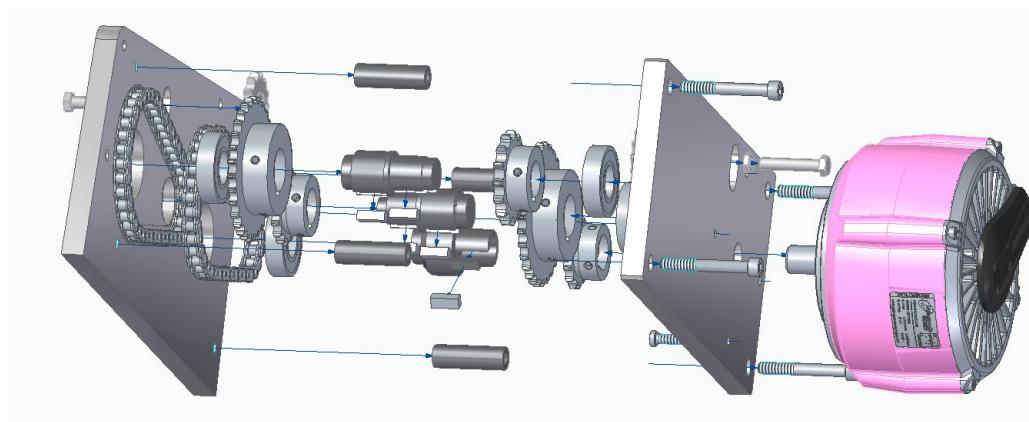


Abbildung IV.29: Detailansicht Getriebe

Kapitel V

Akku und Ladekonzept

1 Übersicht

1.1 Aufgaben der Energieversorgung

In den Bereichen der Wirtschaft und der Technik bedeutet der Ausdruck Energieversorgung so viel wie, die Belieferung von Verbrauchern mit Energie. Es gibt unterschiedliche Arten von Energieträgern. Einerseits gibt es leitungsgebundene Energieträger wie den elektrischen Strom andererseits gibt es auch feste Energieträger wie Kohle oder Holz. In naher Zukunft werden regenerative Energie vermehrt an Bedeutung gewinnen.

Eine stabil Energieversorgung, die mit allen Verbrauchern verbunden ist, wird benötigt, um dem emissionsfreien Sportmotorrad Mobilität zu verleihen. Die Aufgabe der Energieversorgung besteht daher darin, alle Komponenten mit ausreichend Spannung zu versorgen und noch wichtiger, diese Komponenten bei elektrischem Versagen vor Schäden zu schützen. Daher ist die Energieversorgung ein äußerst wichtiger Bestandteil eines solchen Projekts. Im weiteren Verlauf dieses Kapitel wird erklärt, wie die Energieversorgung geplant und in weiterer Folge umgesetzt wurde.

1.2 Aufgaben des Batteriemanagement

Wie bereits im Kapitel 2.5 erklärt wurde, bestehen die Aufgaben des Batteriemanagements darin, Kennwerte der Akkumulatoren aufzunehmen und an eine Steuereinheit weiterzuleiten. Weiters ist es auch für den Schutz der Akkumulatoren zuständig und ebenfalls für einen geregelten Ladevorgang.

Ein Bestandteil des Batteriemanagements ist das sogenannte Batteriemanagementsystem (BMS). Das BMS hat die Aufgabe die Akkumulatoren auf ihre Ladung und Entladung zu überwachen und ebenfalls zu regeln. Die Hauptaufgabe des Batteriemanagementsystem besteht jedoch darin, die Restenergie in den Zellen optimal zu nutzen. Um keine Beschädigungen an Zellen zu bekommen, schützt das Batteriemanagementsystem die Batterien vor Tiefenentladung, vor Überspannung, vor zu schneller Ladung (begrenzen des Ladestroms) und ebenfalls vor einem zu hohen Entladestrom. Bei Akkupacks, d.h. Akkumulatoren mit mehreren Zellen, sorgt das Batteriemanagementsystem außerdem für ein sogenanntes Balancing, das sich darin ausdrückt, dass die verschiedenen Batteriezellen gleiche Ladezustände und Entladezustände haben. Das Batteriemanagementsystem liest und verarbeitet alle Daten, die anschließend über ein Bussystem an die Zentralsteuerung weitergegeben werden.

Um die Akkumulatoren beziehungsweise die Akkupacks wieder aufladen zu können, wird ein externes Ladegerät benötigt. Bei Bedarf wird dieses Ladegerät mit den Akkumulatoren verbunden um den Ladevorgang in Gang zu setzen.

2 Energieversorgung

2.1 Dimensionierung der Akkuzellen

Es ist besonders wichtig die Daten der Zellen vorher zu beachten um die Dimensionierung gut zu planen und schließlich auch umzusetzen. Um die Energie effizient speicher zu können wurden Lithium-Ionen Akkumulatoren verwendet. Da der Elektromotor mit einer Spannung von 50,4V gespeist werden muss, haben wir uns deswegen auch auf ein solches Versorgungssystem geeinigt. Weiters ist bei der Dimensionierung besonders wichtig auf die Lade- und Entladeströme zu achten. Einerseits muss beim Ladestrom nur auf den Strom des Ladegeräts geachtet werden. Andererseits ist es beim Entladestrom etwas komplizierter. Um den Richtwert des Entladestrom einzuhalten, muss der Nennstrom aller Verbraucher in einem System addiert werden.

Der Verbrauch der Steuerelemente setzt sich aus dem verwendeten Raspberry, dem Batteriemanagementsystem und dem Motorcontroller zusammen. Der Raspberry benötigt eine Spannung von 5V und darf eine maximale Leistung von ca. 8W aufweisen. Der Motorcontroller weist eine Nennspannung von 50,4V auf und benötigt eine Leistung von ungefähr 19,5kW. In den folgenden Formeln wird der Strom dieser Komponenten berechnet.

$$I_{Raspberry} = \frac{P_{Raspberry}}{U} = \frac{8W}{5V} = 1,6A$$

$$I_{Motorcontroller} = \frac{P_{Motorcontroller}}{U} = \frac{19,5kW}{50,4V} = 387A$$

Natürlich müsst auf der Stromverbrauch des Batteriemanagementsystem mit eingerechnet werden. Dieser ist jedoch vernachlässigbar, weil dieser bei wenigen mA liegt.

In der Nachfolgenden Abbildung kann man die Kenndaten einer einzelnen Lithium-Ionen Zelle betrachten. Durch diese Kenndaten setzt sich weitere Dimensionierung des Akkukonzeptes zusammen.

EAN / GTIN:	8438493108880
Marke:	Samsung
Batteriegröße:	21700
Batteriechemie:	Li-Ion
Batterie:	Wiederaufladbar
Stromspannung	3,6V
Mindest. Kapazität in mAh:	4.900,00
Batterieverision:	Flache Oberseite
Entladestrom:	9,80
Lithium - Schutzschaltung:	Ungeschützt
Höhe:	70,80
Durchmesser in mm:	21.25

Abbildung V.1: Eigenschaften und Kennwerte einer Lithium Ionen Zelle[[ZellenEigenschaften]]

Wie man in der Tabelle erkennen kann, liegt die Spannung einer Lithium Ionen Zelle bei 3,6V. Da jedoch ein Versorgungssystem mit 50,4V notwendig ist müssen die Zellen beziehungsweise die Akkupacks seriell sowie auch parallel verschalten werden.

Die Nennkapazität einer Zelle beträgt 4900mAh. Um die Kapazität zu erhöhen wurden innerhalb eines Akkupacks 40 einzelne Akkumulatoren parallel verschalten.

$$Q_{Gesamt} = Q_{Akkupack} = Q_{Zellen} \cdot 40 = 4900 \text{ mAh} \cdot 40 = 196.000 \text{ mAh}$$

Die Nennkapazität einer Lithium Ionen Zelle beträgt 3,6V. Innerhalb der Akkupacks werden diese 40 Zellen parallel verschalten, was nicht zu einer Erhöhung der Spannung führt. Da wir aber ein Versorgungssystem mit 50,4V benötigen, musst wir die 14 Akkupacks seriell verschalten um die Versorgungsspannung zu erreichen.

$$U_{Gesamt} = U_{Akkupack} \cdot 14 = U_{Zelle} \cdot 14 = 3.6 \text{ V} \cdot 14 = 50.4 \text{ V}$$

2.2 Zusammenstellung der Batteriezellen

Die Aufgabe besteht darin, die einzelnen Zellen zu sogenannten Akkupacks zusammenzuschalten. Dafür wurde Abstandshalter verwendet um die Zellen anzuzuordnen und anschließend mit einem Hiluminband zusammenzuschweißen.



Abbildung V.2: Explosionsdarstellung einer Doppelzelle

Im nächsten Schritt wurden diese Doppelzellen zusammengeschalten, die vereint ein Akkupack ergeben. Die Schwierigkeit dabei war, die Doppelzellen so anzuzuordnen, dass der Platz in der Schutzbox für die Akkumulatoren möglichst effizient genutzt wird. Jedoch waren wir beim Designen der Akkubox relativ eingeschränkt, da wir darauf achten mussten, diese Boxen auch an unser Motorrad anzubringen. Die Boxen zusammen mit dem Inhalt sind auch extrem schwer. Deswegen war es wichtig, diese Boxen möglichst unten und in der Mitte zwischen den Achsen anzubringen um den Schwerpunkt des emissionsfreien Sportmotorrads nicht drastisch zu verändern.

Wie bereits vorher erwähnt müssen diese Akkupacks geschützt werden. Dies ist notwendig um die Lebenszeit der Akkumulatoren zu verlängern und um sie vor äußeren Einflüssen zu schützen. Diese Akkuboxen haben unterschiedliche Formen, da sie an unterschiedlichen Positionen an dem Sportmotorrad angebracht werden. Grundsätzlich haben wir 3 verschiedene Akkuboxen designt und angefertigt.

1. Akkubox:

Die erste Akkubox besteht aus 2 Akkupacks. Das heißt insgesamt beinhaltet sie 80 Zellen.

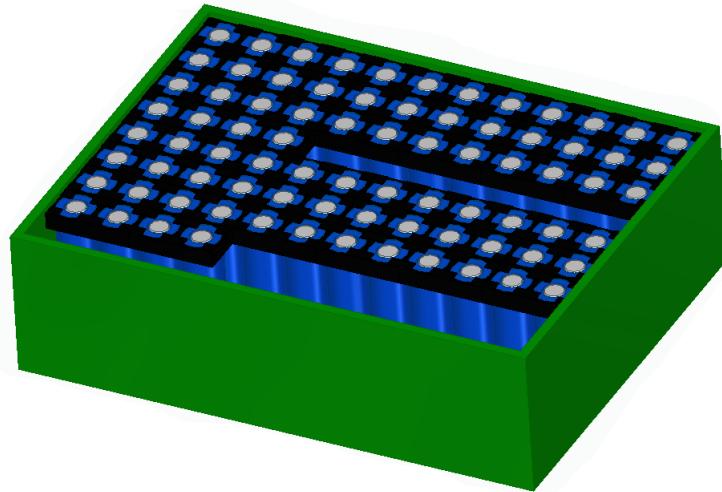


Abbildung V.3: 1. Akkubox

2. Akkubox:

Die zweite Akkubox besteht aus insgesamt 6 Akkupacks. Diese Akkubox ist in drei Ebenen aufgeteilt wobei in einer Ebene 2 Akkuboxen Platz finden. Die nächsten beiden Akkuboxen liegen auf der ersten Ebene auf. Schlussendlich befinden sich die letzten 2 der 6 Akkuboxen auf der letzten, der dritten Ebene.

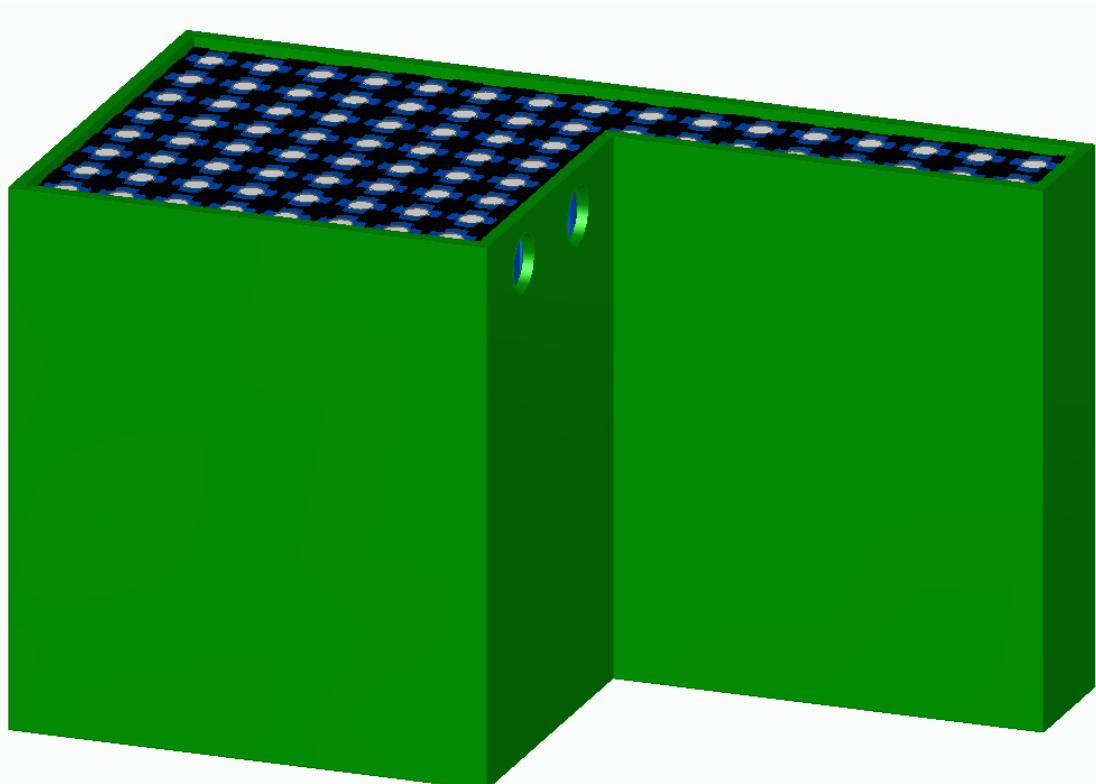


Abbildung V.4: 2. Akkubox

3. Akkubox:

Die dritte und somit letzte Akkubox beinhaltet insgesamt 6 Akkupacks. Diese Box wird an der Vorderseite, genauer parallel zu den Stoßdämpfern, angebracht. Es ist wichtig die Box so nahe es geht am Mittelpunkt und möglichst weit in Bodennähe anzubringen, um den Schwerpunkt des Motorrads nicht drastisch zu verändern.

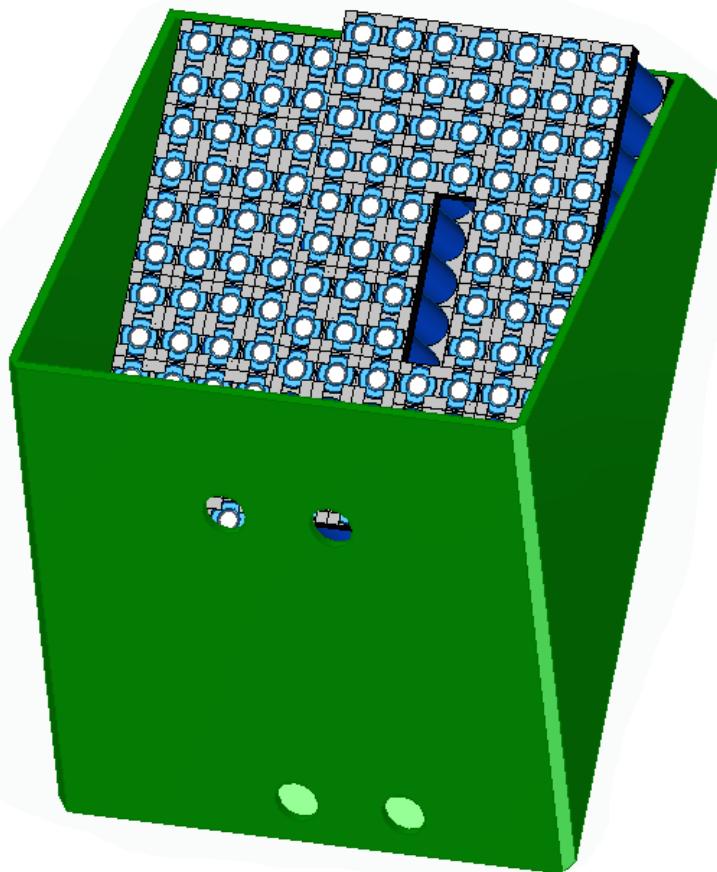


Abbildung V.5: 3. Akkubox

Ergebnis: Zuerst wird die Anzahl der Akkupacks im gesamten System berechnet.

$$\text{Akkupacks}_{\text{Gesamt}} = 1.\text{Akkubox} + 2.\text{Akkubox} + 3.\text{Akkubox} = 2 \text{ Packs} + 6 \text{ Packs} + 6 \text{ Packs} = 14 \text{ Akkupacks}$$

Anschließend werden die Zellen jeder einzelnen bereits oben beschriebenen Akkubox berechnet.

$$\text{Zellen}_{1.\text{Akkupack}} = 1.\text{Akkubox} = 2 \text{ Packs} = 80 \text{ Zellen}$$

$$\text{Zellen}_{2.\text{Akkupack}} = 2.\text{Akkubox} = 6 \text{ Packs} = 240 \text{ Zellen}$$

$$\text{Zellen}_{3.\text{Akkupack}} = 3.\text{Akkubox} = 6 \text{ Packs} = 240 \text{ Zellen}$$

Schlussendlich wird die Zellanzahl der verschiedenen Akkuboxen miteinander addiert und ergeben die Gesamtanzahl der verwendeten Zellen in dem Konzept.

$$\begin{aligned} \text{Zellen}_{\text{Gesamt}} &= \text{Zellen}_{1.\text{Akkupack}} + \text{Zellen}_{2.\text{Akkupack}} + \text{Zellen}_{3.\text{Akkupack}} \\ &= 80 \text{ Zellen} + 240 \text{ Zellen} + 240 \text{ Zellen} = 560 \text{ Zellen} \end{aligned}$$

2.3 Verschaltung der Batteriezellen

Bei der Verschaltung der Batteriezellen war es wichtig auf die Kennwerte der Komponenten zu achten. Da unser elektrischer Motor mit einer Spannung von 50,4V versorgt werden muss, haben wir uns daher auch auf ein 50,4V Versorgungssystem geeinigt. Im weiteren Verlauf wurde geplant und schließlich auch umgesetzt wie die Zellen am besten verschaltet werden sollten.

Grundsätzlich gibt es 3 verschiedene Möglichkeiten einzelne Zellen zu verschalten:

- **Serienschaltung**
- **Parallelschaltung**
- **Kombination aus Serien-und Parallelschaltung**

2.3.1 Serienschaltung

Das Prinzip der Serienschaltung ist relativ einfach. Der Minuspol der einen Batterie wird einfach mit dem Pluspol der anderen Batterie verbunden. Das hat zur Folge, dass alle Zellen mit dem selben Strom durchflossen werden. Jedoch addieren sich alle Teilspannungen der Zellen zu einer Gesamtspannung. Die Serienschaltung von Batterien wird häufig auch als Hintereinanderschaltung oder Reihenschaltung bezeichnet.

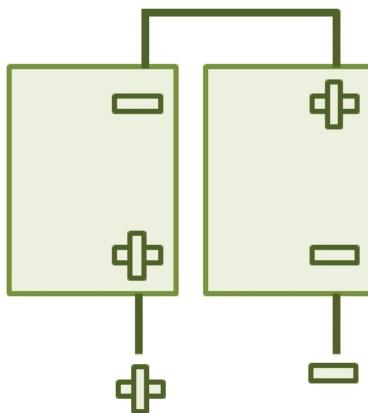


Abbildung V.6: Serienschaltung beliebig vieler Zellen[[ReihenschaltungZellen]]

Beispiel: Werden zwei Batterien mit jeweils 300Ah (Amperestunden) und 9V (Volt) in Reihe geschaltet, ergibt sich eine Ausgangsspannung von 18V mit einer Kapazität von 300 Ah.

Vorteile und Nachteile:

Durch die Reihenschaltung wird ermöglicht, höhere Gesamtspannungen zu erzeugen. Ein bedeutender Nachteil ist jedoch, dass die schwächste Zelle die Leistung der gesamten Reihenschaltung beeinflusst. Außerdem kann es vorkommen, dass wenn eine einzelne Batteriezelle defekt ist, die gesamte Batteriereihe ausfällt. Daher ist es notwendig zusätzliche Sicherungen hinzu zu schalten. Im Normalfall können nur Batteriezellen vom gleichen Hersteller, Typ und von der selben Batterietechnik kombiniert und in Serie geschalten werden.

2.3.2 Parallelschaltung

Das Prinzip der Parallelschaltung ist relativ ähnlich zu der Serienschaltung. Der große Unterschied jedoch besteht darin, dass der Minuspol der einen Batterie mit dem Minuspol der anderen Batterie verschaltet wird. Dadurch summieren sich die Ladekapazitäten (Ah) der einzelnen Zellen zu einer gesamten Ladekapazität. Die Gesamtspannung entspricht jedoch der Spannung einer einzelnen Batterie.

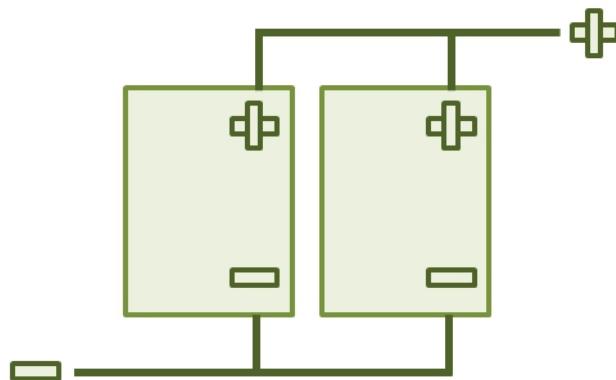


Abbildung V.7: Parallelschaltung beliebig vieler Zellen[[ParallelschaltungZellen]]

Beispiel: Werden zwei Batterien mit jeweils 300 Ah und 9V parallel geschaltet, so ergibt sich eine Ausgangsspannung von 9V und eine Gesamtkapazität von 600Ah.

Vorteile und Nachteile:

Die Parallelschaltung führt dazu, die Leistungsfähigkeit und außerdem die Lebensdauer zu steigern. Jedoch ist die Laderegelung bei einer Parallelschaltung nicht unkompliziert, da jede Zelle unterschiedlich schnell altern und damit eine Fehlerquelle darstellen kann. Deshalb kann es vorteilhaft sein, eine größere Batterie anstatt mehrerer einzelner Zellen zu verwenden. Sollte trotzdem einer Parallelschaltung mehrerer Zellen vorgenommen werden, sollte unbedingt darauf geachtet werden Batterien mit gleicher Kapazität, Bauart und Ladezustand einzusetzen. Außerdem ist es von Vorteil möglichst kurze Leitungen einzusetzen, um Spannungsverlust zu verhindern.

2.3.3 Kombination aus Serien- und Parallelschaltung

Durch eine Kombination aus einer Serien- und Parallelschaltung ermöglicht man eine größeres Flexibilität zur Erreichung einer bestimmten Spannung und auch Leistung. Wie bereits vorher erwähnt, erreicht man durch eine Parallelschaltung die benötigte Gesamtkapazität und durch die Serienschaltung die gewünschte Betriebsspannung.

Diese Methode wird in der Praxis am häufigsten eingesetzt und ist auch bei unserem Akkusystem zum Einsatz gekommen.

2.3.4 Verschaltung der Zellen zu einem Akkupack

Da wir uns auf ein 50,4V Versorgungssystem geeinigt haben, und jede einzelne Zelle eine Spannung von 3,6V aufweist haben wir jeweils 40 einzelne Lithium Ionen Akkus zusammengeschaltet. Die einzelnen Batterie werden in den Akkupacks parallel verschaltet, da wir so einen deutlich höheren Leistungsfähigkeit erreichen und die Lebensdauer dadurch verlängert wird.

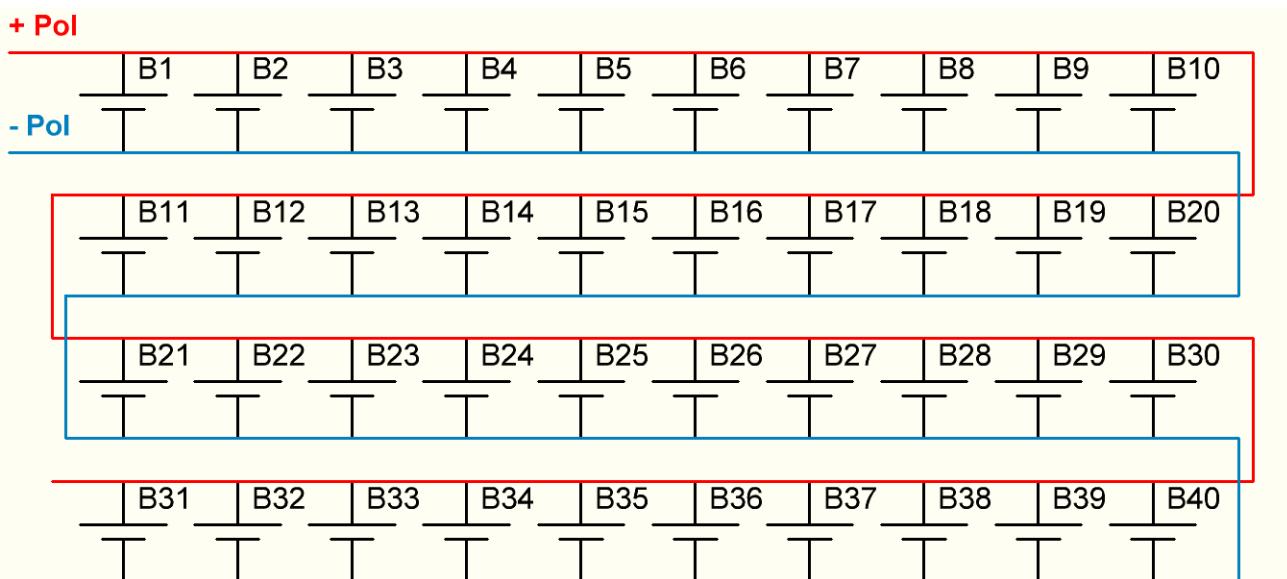


Abbildung V.8: Verschaltung der einzelnen Zellen innerhalb der Akkupacks

2.3.5 Verschaltung der Akkupacks

Durch die Parallelschaltung innerhalb der Akkupacks wird jedoch nicht die Spannung erhöht. Wir haben eine Anzahl von 560 einzelnen Lithium Ionen Zellen. Ein Akkupack beinhaltet 40 Zellen was dazu führt, dass wir insgesamt 14 Akkupacks haben. Diese 14 Akkupacks werden seriell verschalten um die Spannung von 3,6V auf die gewünschten 50,4V zu vergrößern.

Die separaten Zellen werden mithilfe eines Hiluminband Punktgeschweißt um so eine Verbindung zwischen den Zellen herzustellen. Wichtig ist darauf zu achten, dass das Hiluminband nur bis zu einer Stromstärke von 30A zulässig ist. Da wir jedoch weitaus höhere Ströme zu erwarten haben, muss wir dieses Hiluminband verstärken. Dies erfolgt über eine Metallplatte die jeweils an der Oberseite sowie auch an der Unterseite angebracht wird. Dadurch wird die Schweißverbindung gestärkt, damit in weiterer Folge keine Fehler auftreten.

$$U_{Gesamt} = U_{Akkupack} \cdot 14 = U_{Zelle} \cdot 14 = 3.6 \text{ V} \cdot 14 = 50.4 \text{ V}$$

Aus Darstellungsgründen wurden die Akkupacks 4 - 13 durch das Akkupack ... ersetzt.

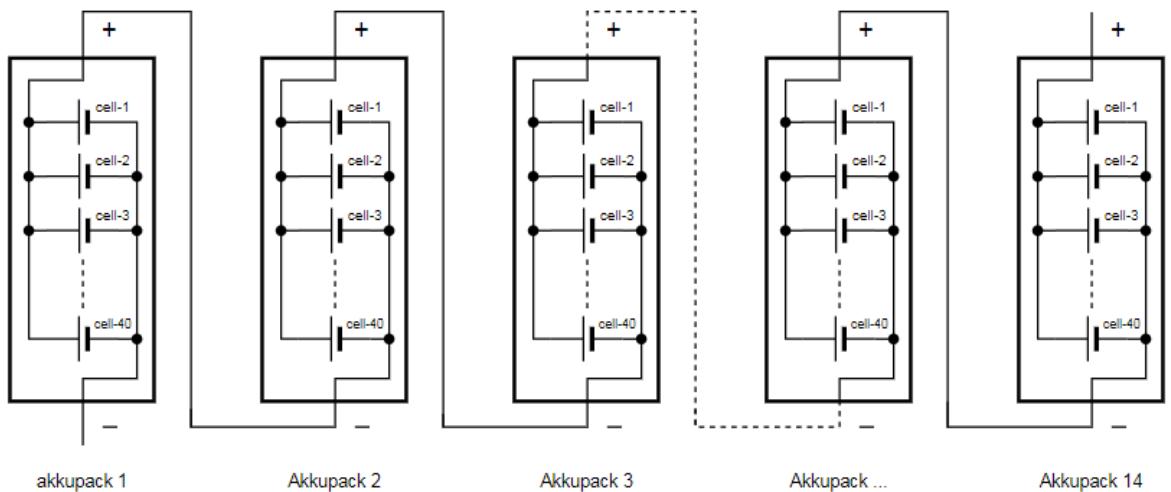


Abbildung V.9: Verschaltung der Akkupacks

2.3.6 Geschätzte Betriebszeit des Akkumulators

In diesem Abschnitt wurden alle relevanten Daten gesammelt, um schließlich eine Näherung für die Laufzeit des Akkumulators durchzuführen. Diese Daten wurden bereits im Kapitel Dimensionierung teilweise berechnet.

Im ersten Schritt wird die Energie, die der Akkumulator aufnehmen kann berechnet.

$$W_{Akkumulator} = U_{Gesamt} \cdot Q_{Gesamt} = 50.4 \text{ V} \cdot 196.000 \text{ mAh} = 9880 \text{ Wh}$$

Als nächstes wird die Leistung berechnet. Da sich das Motorrad jedoch nicht immer mit voller Geschwindigkeit fortbewegen wird, haben wir eine Auslastung von 75 Prozent angenommen und eine realistische Akkulaufzeit zu berechnen.

$$P_{Gesamt} = P_{Motorcontroller} \cdot P_{Raspberry} = 19.5 \text{ kW} \cdot 8 \text{ W} = 19.508 \text{ kW}$$

Wie bereits vorher erwähnt müssen von dieser berechneten Gesamtleistung 25 Prozent abgezogen werden. Da jedoch die Leistung des Raspberry so gering im Vergleich zur Leistung des Motorcontrollers ist, kann diese vernachlässigt werden.

$$P_{Gesamt,real} = P_{Gesamt} \cdot 0,75 = 19.5 \text{ kW} \cdot 0.75 = 14.63 \text{ kW}$$

Schlussendlich kann die ungefähre Laufzeit des emissionsfreien Sportmotorads berechnet werden.

$$t_{Laufzeit} = \frac{W_{Akkumualtor}}{P_{Gesamt,real}} = \frac{9880 \text{ Wh}}{14.63 \text{ kW}} = 0,68 \text{ h} = 41 \text{ min}$$

Die geschätzte durch eine Näherung berechnet Laufzeit des Akkumulator beträgt in etwa 41min.

3 Batteriemanagementsystem

Der zweite Teil, der für ein funktionierendes Akkusystem notwendig ist, ist das Batteriemanagementsystem. Die Aufgabe des BMS ist es die Akkumulatoren zu überwachen und an die aufgenommenen Daten über eine serielle Schnittstelle an die Zentralsteuerung weiterzugeben.

In unserm Projekt wurde ein BMS Mini des Hersteller EMUS verwendet. Wir haben uns für das EMUS BMS Mini entschieden, da es ein eigenständiges Lithium-Batteriemanagementsystem mit vielen bereits integrierten Bauteilen und die passenden Spezifikationen für unser Projekt hat. Unter anderem passen die Anwendungsbereiche des Batteriemanagementsystems genau zu unserer Anwendung. Außerdem sind all diese BMS-Funktionen in einem einzigen Gerät integriert.



Abbildung V.10: Mini Batteriemanagementsystem der Firma EMUS[[EmusBmsMini]]

Funktionen des EMUS BMS Mini:

- Messung der Zellenspannungen und Zellentemperaturen
- Batteriestrommessung
- Ausgleich der Zellen (Battery-Balancing)
- Integriertes Schütz zum Schutz der Batterie
- Integrierter Vorladekreis für den Sanftanlauf der Motorsteuerung
- Konnektivität zu verschiedenen Arten von Ladegeräten
- Konnektivität zu Eingängen zur Steuerung des BMS und der Batterie
- Konnektivität zu den Ausgängen zur Anzeige des Batteriestatus und der Steuerung
- Gegebenfalls Konnektivität zu Smartphones zur Steuerung und Konfiguration

Spezifikationen des BMS:

- Anzahl der Zellen(Akkupacks): 6-16
- Einzellenspannung: 1,00 - 4,95 V
- Batteriepackspannung: DC 15,0 - 67,2 V
- Betriebstemperatur: -40°C bis +80°C
- Verbrauchte Leistung: im aktiven Modus (11mA bei 24V)
- Verbrauchte Leistung: im Leerlaufmodus (4mA bei 24V)
- Verbrauchte Leistung: im Tiefschalf (1mA bei 24V)
- Ausgleichsstrom: bis zu 200mA
- Dauerstrom 45A laden/entladen (Spitze 100A - 5s)
- Maße: 97 x 67,1 x 16,6mm

Verwendbare Schnittstellen:

- CAN v2.0 A/B
- RS-232 zur Fernüberwachung
- Drahtlose Smartphone - Konnektivitätsschnittstelle

Ladegeräte die das BMS unterstützt:

- Schützgesteuertes NonCAN-Ladegerät
- Elcon J1939 Ladegerät
- DeltaQ CANopen Ladegeräte

Wie bereits vorher erwähnt beinhaltet unser Akkusystem 560 Zellen. Die Bezeichnung 16s bedeutet, dass das Batteriemanagement insgesamt maximal 16 Zellen überwachen kann. Daher ist es nicht möglich jede Zelle einzeln zu überwachen da so viele Anschlüsse nicht vorhanden sind. Jedes Akkupack beinhaltet 40 Zellen.

$$\text{Akkupacks} = \frac{\text{Zellen}_{\text{gesamt}}}{\text{Zellen}_{\text{Akkupack}}} = \frac{560 \text{ Zellen}}{40 \text{ Zellen}} = 14 \text{ Akkupacks}$$

3.1 Funktionen des Batteriemanagementsystems

Unter dem Begriff Batteriemanagementsystem versteht man die Überwachung und Regelung von Akkumulatoren beim Laden und ebenfalls Entladen. Ein BMS ist nichts anderes als eine elektronische Regelschaltung. Zu den häufigsten Batteriekennwerten gehören die Erkennung des Batterietyps, die Batteriespannung, die Spannung der Zellen sowie die Temperatur einzelner Akkumulatoren oder gesamter Akkupacks, die Akkukapazität, der Ladezustand, die Restbetriebszeit, die Stromentnahme und einige Kennwerte mehr. Die Aufgabe des Batteriemanagementsystems besteht also darin, sichzugehen, dass die Energie in einer Zellen effizient genutzt wird. Um Beschädigungen von Akkus zu vermeiden, haben die Akkus einen Tiefentladungsschutz, einen Überspannungsschutz und außerdem schützt das BMS auch vor zu schneller Ladung und ebenfalls vor einem zu hohen Entladestrom. Oft ist es der Fall, dass in einem Projekt mehrere einzelne Akkus oder sogar Akkupacks zum Einsatz kommen. In einem solchen Fall sorgt das Batteriemanagementsystem für ein sogenanntes Balancing. Das Battery-Balancing sorgt dafür, dass die Akkupacks oder einzelne Zellen gleiche Ladezustände und Entladezustände haben.

Die folgenden Funktionen des Batteriemanagementsystems werden in den weiteren Seite genauer erklärt:

- **Balancing**
- **Spannungsmessung**
- **Strommessung**
- **Temperaturüberwachung**

3.1.1 Balancing

Grundsätzlich wurde das Battery-Balancing bereits im Kapitel 5.5.2 erklärt. Im weiteren Verlauf wird das Battery-Balancing im Zusammenhang mit den Akkupacks genauer erklärt und welche verschiedenen Arten es davon gibt.

Balancing in Zusammenhang mit den Akkupacks:

Muss noch zitiert werden: Cluster oder Akkupacks bestehen zur Erhöhung der Nennspannung in der Regel aus mehreren in Reihe geschalteten Einzelzellen oder Zellblöcken. Fertigungs- und Alterungsbedingt gibt es hierbei Schwankungen in der Kapazität, im Innenwiderstand und weiteren Parametern dieser Zellen. Die schwächste Zelle ist dabei bestimmend, wie viel geladen oder entladen werden darf. Im praktischen Einsatz von mehrzelligen in Reihe verschalteten Akkus führt dieser Umstand dazu, dass die Zellen in Reihe unterschiedlich geladen und entladen werden.

Es kommt dann im Verbund zu kritischer Tiefentladung oder bei der Ladung zu einer Überladung und Überschreiten der Ladeschlussspannung einzelner Zellen. Je nach Akkutyp kann es dabei zu einer irreversiblen Schädigung einzelner Zellen kommen. Die Folge: das gesamte Akkupack verliert an Kapazität. (Ende des Zitates) Um das zu verhindern, spielen im Batteriemanagementsystem die Balancer eine wichtige Rolle. Beim Battery-Balancing gibt es zwei verschiedene Arten.

- **Passives Battery-Balancing**
- **Aktives Battery-Balancing**

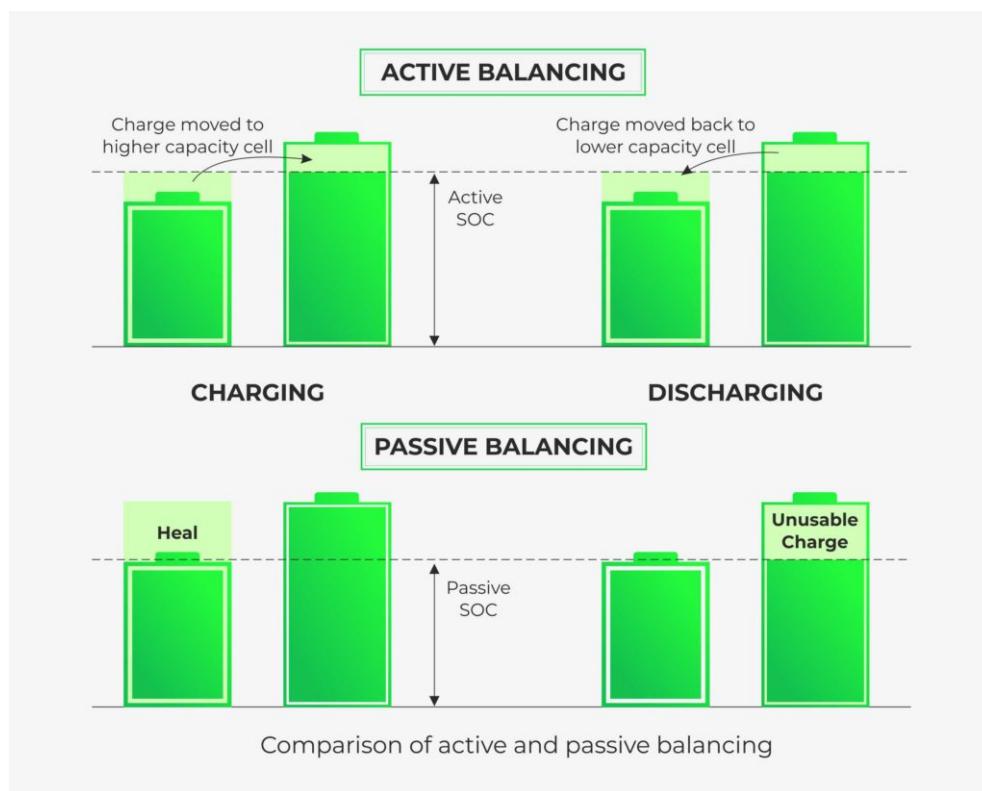


Abbildung V.11: Vergleich zwischen Aktiven- und Passiven-Battery Balancing[[VergleichBalancing]]

Passives Battery-Balancing:

Eine technische weit verbreitete Methode für das Balancing ist das Passive Battery-Balancing. Dabei arbeitet es nur im Bereich des Ladeschlusses. Der Ausdruck Ladeschluss bedeutet soviel, dass wenn die Akkumulatoren fast vollständig aufgeladen sind das Balancing zu arbeiten beginnt. Sobald die Zellen die Ladeschlussspannung erreicht haben, wird durch den Balancer ein Widerstand parallel dazugeschalten, um so die Spannung auf die Ladeschlussspannung zu begrenzen. Zellen, welche diese Spannung bereits erreicht haben, werden dann nur noch geringfügig weitergeladen oder teilweise sogar etwas entladen. Die Zellen, die jedoch noch in der Reihenschaltung verschalten sind und die Ladeschlussspannung noch nicht erreicht haben, werden weiterhin mit dem Ladestrom versorgt und somit weitergeladen. Wichtig ist darauf zu achten, dass die Leistung des Parallelwiderstandes auf den Ladestrom angepasst werden muss, da sonst zuviel Energie in Form von Wärme am Widerstand auftreten wird.

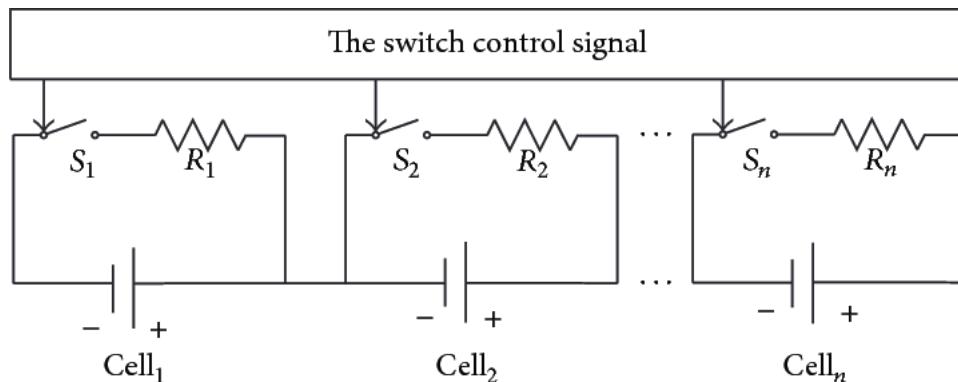


Abbildung V.12: Funktionsweise des Passiven Battery-Balancing[[FunktionsweisePassivBalancing]]

Vorteile des Passiven Battery-Balancing:

- sehr kostengünstig
- technisch relativ leicht realisierbar

Nachteile des Passiven Battery-Balancing:

- Ladevorgang kann extrem lange dauern, da man warten muss bis die schwächste Zelle den geforderten State of Charge (SOC) erreicht hat
- viel Energie verpufft in Form von Wärme
- Diese Verlustwärme wirkt sich negativ auf die Lebensdauer der Akkumulatoren aus
- nicht unerhebliche Brandgefahr

Aktives Battery-Balancing:

Diese Methode des Balancing ist etwas komplexer als beim Passiven Battery-Balancing, jedoch ist sie deutlich effizienter. Bei den aktiven Balancern wird ein Ladungstransfer von Zellen untereinander realisiert. Das bedeutet, dass die Energie der Zellen die bereits eine höheren Ladung aufweisen, auf die Akkumulatoren mit niedrigerer Ladung übertragen werden. Beim Aktiven Battery-Balancing werden sogenannte Ladereglungen benötigt. Ladereglungen sind im Prinzip speziell auf eine Anwendung optimierte Schaltregler, die pro Zelle arbeiten und aktiv die Energie übertragen. Dieser Vorgang kann bereits während des Ladezyklus erfolgen. Standardmäßig wird dieser Vorgang jedoch erst im Bereich des Ladeschlusses aktiv (gleich wie beim Passiven Battery-Balancing). Eine weiterentwickelte Form dieses Systems wird bidirektionale Balancer-System genannt. Hierbei ist es möglich, dass der Ladungsaustausch sowohl beim Entladen als auch beim Aufladen der Zellen stattfinden kann. Deswegen sind diese bidirektionale Balancer noch deutlich effizienter. Das Aktive Battery-Balancing wird heutzutage meistens bei größeren Leistungen angewandt, wie zum Beispiel im Bereich der Elektromobilität.

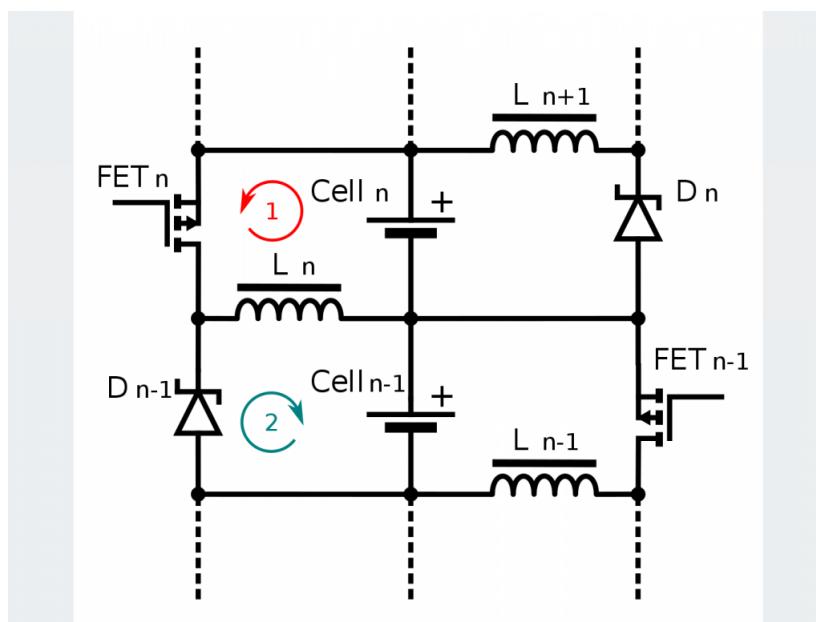


Abbildung V.13: Funktionsweise des Aktiven Battery-Balancing[[FunktionsweiseAktivBalancing]]

In der obigen Abbildung kann man die Prinzipschaltung eines aktiven Balancers mit zwei Stufen sehen. Innerhalb von zwei Schaltvorgängen kann dabei die Energie aus der Akkuzelle Cell n über den FET n in die Spule L n übertragen werden (Schleife in rot, 1). Im zweiten Schaltvorgang (Schleife in blau, 2) wird die Energie in der Spule L n über Diode D n-1 in die Cell n-1 geladen und Cell n-1 aufgeladen.

Vorteile des Aktiven Battery-Balancing:

- deutlich höherer Wirkungsgrad als beim passiven Battery-Balancing
- übergeordnete Ladereglung mit intelligenter und lernfähiger Software
- Lebensdauer der Akkumulatoren kann durch die Methode der Ladungsumverteilung deutlich erhöht werden
- überschüssige Energie wird nur zu einem geringen Grad in Wärme umgewandelt
- geringeres Risiko für eine Entzündung

Nachteile des Aktiven Battery-Balancing:

- höherer Verschaltungsaufwand, dadurch erhöhte Initialkosten

3.1.2 Spannungsmessung

Lithium Ionen haben die Eigenschaft, dass sie Überladungen (Überspannung) sowie auch Unterspannung nicht gut verkraften. Im schlimmsten Fall, kann das sogar zur vollständigen Zerstörung einer Zelle kommen. Deswegen hat das Batteriemanagementsystem die Aufgabe jedes Akkupack zu messen und bei gegebener Über- und Unterspannung sofort abzuschalten. Der Schutz vor Überspannung wird zum Beispiel dann eingeschalten, sollte es zu einer ungewollten Rückspeisung kommen. Der Unterspannungsschutz hingegen tritt ein, wenn der vorgegebene Unterspannungsparameter überschritten wird. Sollte dies der Fall sein wird der Akkumulator von der Last getrennt, sodass der Akku nicht weiter entladen oder auch überladen werden kann. Die Spannungsmessung ist außerdem dafür zuständig die Spannung der Akkupacks zu messen.

3.1.3 Strommessung

Bei dem Daly Batteriemanagementsystem ist eine Strommesseinrichtung bereits integriert. Diese Strommesseinrichtung misst einerseits den Entladestrom und andererseits den Ladestrom. Sollte es wie beim Überspannungsschutz zu einer ungewollten Rückspeisung kommen, greift der Ladestromschutz ein. Sollte es dazu kommen, dass der Akkumulator zu viel Strom abgeben würde, wird der Entladeschutz aktiviert. Zudem wird die Strommesseinrichtung dazu benötigt, den Ladezustand des Akkumulator anzugeben.

3.1.4 Temperaturüberwachung

Lithium Ionen Akkumulatoren neigen wie auch viele andere Akkus dazu, sich bei einer zu großen Stromentnahme stark zu erwärmen. Das kann zu inneren sowie auch zu äußereren Schäden am Akkumulator führen. Deswegen ist es wichtig, die Temperatur der Akkupacks zu überwachen um Schäden vorzubeugen. Sollte der Akkumulator unter oder über einen gewissen Temperaturbereich gelangen, wird dieser von der Last getrennt um eine weitere Erwärmung und dadurch größere Schäden zu vermeiden.

3.2 Verschaltung des Batteriemanagementsystems

3.2.1 Anschließen der Akkupacks an das BMS

Die Hersteller des BMS mini empfehlen die Akkumulatoren in einem möglichst geringen Abstand zur Batterie zu installieren, um den Verdrahtungsprozess zu vereinfachen. Da es notwendig ist die Spannung jedes Akkupacks zu überprüfen muss auch jedes einzelne an das Batteriemanagementsystem angeschlossen werden. Um mit einer sicheren Installation zu starten muss am Anfang sichergestellt werden, dass der Minuspol des ersten Akkumulators mit dem Terminal (-) am BMS verbunden wird. Anschließend wird genau anders herum der Pluspol des Akkus mit dem Terminal (+) verbunden. Danach werden die einzelnen Akkupacks mit dem Batteriemanagementsystem verbunden.

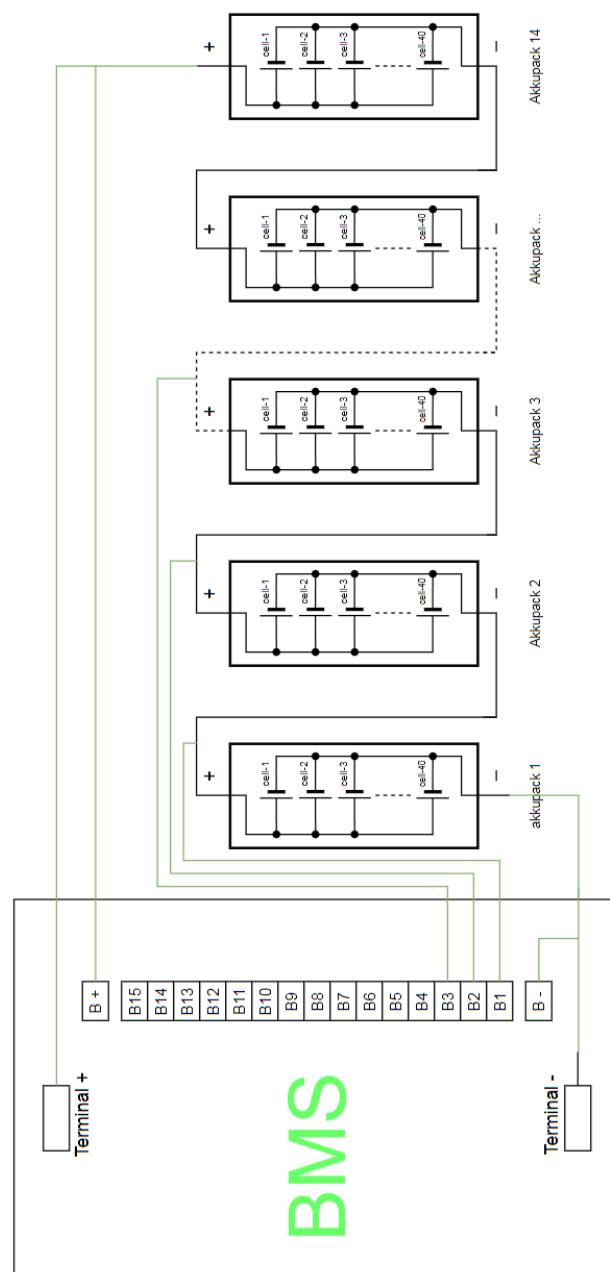


Abbildung V.14: Anschlussplan der Akkupacks an das BMS

3.2.2 Temperaturmessung mithilfe des BMS

Im nächsten Schritt werden Temperatursensoren gleichmäßig verteilt und an die Akkupacks angebracht. Es ist nicht notwendig an jedem einzelnen Akkupack einen Temperatursensor anzubringen. Jedoch sollteman sich bewusst sein, dass mithilfe solcher Sensoren teilweise Fehler schon im vorhinein behoben werden können bevor es noch zu größeren Schäden im System kommt.

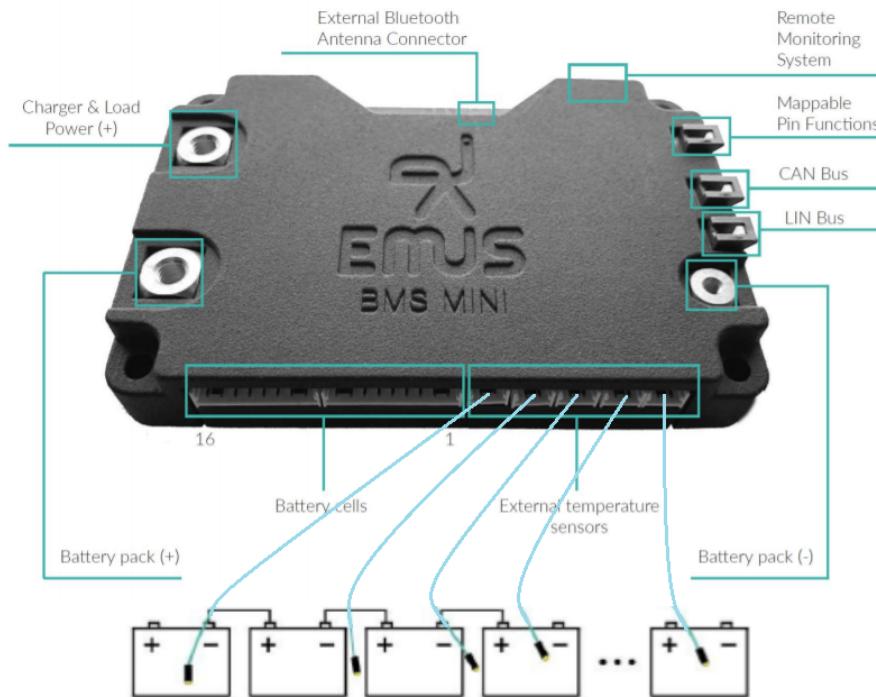


Abbildung V.15: Anschlussplan der Temperatursensoren[[AnschlussplanTemp]]

3.2.3 Anschlussplan der Last

Im letzten Schritt wird die Last an dem Batteriemanagementsystem angeschlossen. Die Last entspricht entweder dem Verbrauchen oder anderfalls dem Ladegrät. Um die Last an das BMS anzuschließen muss das Power (+) Terminal mit dem Pluspol der Last (Load/Motor (+) Terminal) verbunden werden. Der zweite Pol der Last (Load/Motor (-) Terminal) wird mit dem Battery (-) Terminal verbunden. Sehr wichtig ist dabei darauf zu achten, dass die Last nicht unmittelbar mit Strom versorgt werden muss. Dazu wird ein Leistungsschutzschalter zwischen Last und dem Power (+) Terminal dazwischengeschalten.

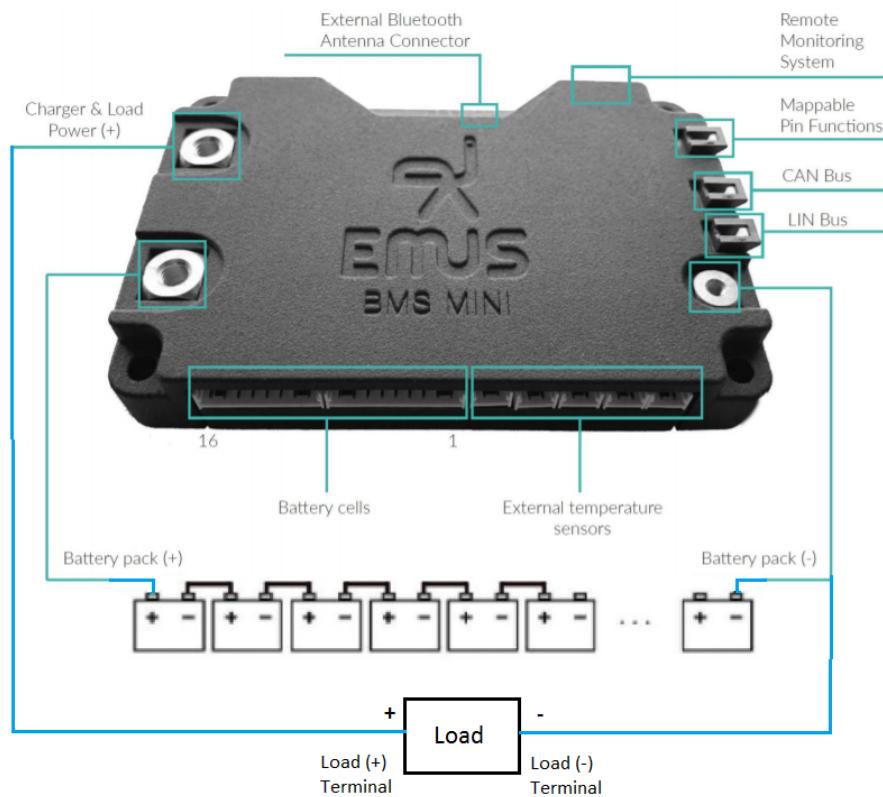


Abbildung V.16: Anschlussplan der Last an das BMS[[AnschlussplanLast]]

3.2.4 Kommunikation mittels CAN-Bus

Der CAN-Bus ist aufgrund seiner Stärken und seines Kommunikationsmechanismus in vielen Anwendungsbereichen weit verbreitet. Wichtig ist, dass der CAN-Bus richtig eingerichtet wird, da man sonst seine Zuverlässigkeit stark gefährdet. Dies ist bereits bei den EMUS BMS-Komponenten welche mit einem CAN ausgestattet sind der Fall und weiters ist es sogar in der Norm ISO 11898-2 festgelegt. Meistens wird der CAN-Bus in Hochgeschwindigkeits Netzwerken verwendet. In der Norm ist eine Standard definierte Netzwerktopologie mit einer einzelnen Leistungstruktur abgebildet. Die Kernaussage dieser Norm ist, dass am Ende zwischen CAN-High und CAN-Low ein Leitungsabschlusswiderstand integriert sein muss. Wie bereits vorher erwähnt kommuniziert das Batteriemanagementsystem mit der Zentralsteuerung mittel CAN-Bus. In der folgenden Abbildung ist die Topologie eines CAN-Netzwerkes abgebildet.

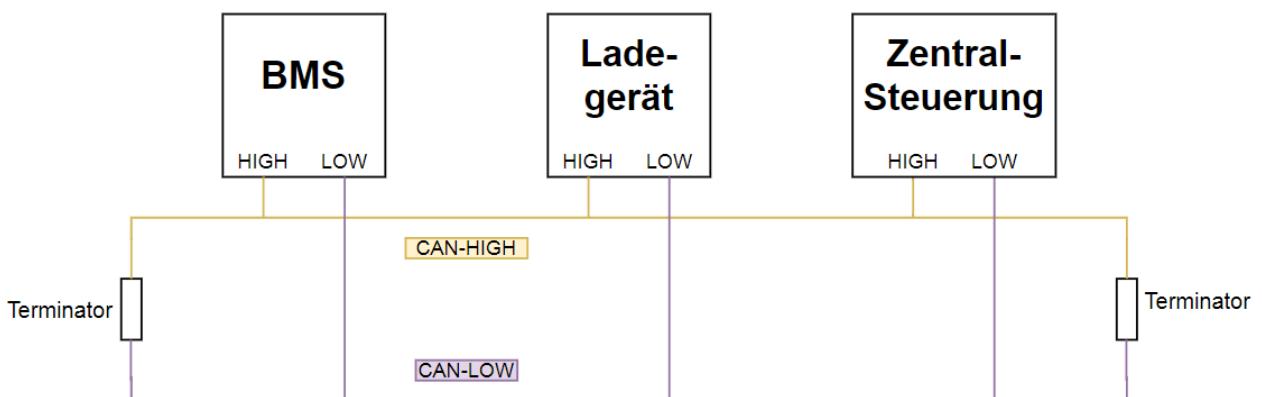


Abbildung V.17: Topologie eines High Speed CAN-Netzwerks

Wenn man Daten von dem Batteriemanagementsystem zur SPS sendet, kann es zu Übertragungsproblemen kommen. Um diese Übertragungsprobleme zu minimieren muss zwischen CAN High und CAN Low ein Leitungsabschlusswiderstand (Skizze: Terminator) eingebaut werden. Dieser Terminator wird jeweils an dem Batteriemanagementsystem (Bus-Ende"links) und ebenfalls an der SPS (Bus-Ende"rechts) angebracht, da diese zwei Komponenten ununterbrochen miteinander verbunden sind.

CAN-Bus Stecker an dem Batteriemanagementsystem:

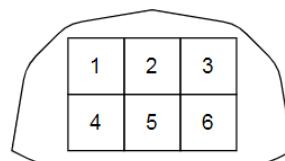


Abbildung V.18: Steckerbelegung des CAN-Buses an dem BMS

- Pin 1: Versorgung
- Pin 2: CAN Terminator (Leitungsabschlusswiderstand)
- Pin 3: CAN - HIGH
- Pin 4: Ground (Masse)
- Pin 5: CAN Terminator (Leitungsabschlusswiderstand)
- Pin 6: CAN - LOW

3.3 Applikation des BMS Mini

Die Hersteller, EMUS, des Batteriemanagementsystems haben eine eigene Anwendung, die auf jedem Smartphone und Tablet verfügbar ist, erstellt. Die Anwendung Namens EMUS EVGUI ist eine frei verfügbare App für Android- und iOS-Geräte, die speziell für die Überwachung des Batteriestatus in einem Elektrofahrzeug entwickelt worden ist. Das Smartphone kann einerseits über ein drahtloses BT-Protokoll mit dem BMS Mini verbunden werden. Dazu wird jedoch die EMUS G1 Smartphone-Konnektivität benötigt. Die zweite Möglichkeit besteht darin, das Smartphone über ein einfachs USB Kabel mit dem Batteriemanagementsystem zu verbinden. In der Hauptansicht der EMUS EVGUI App sind die wichtigsten Anzeigen abgebildet:

Hauptanzeige der EMUS EVGUI App:

- Temperatur und Ladezustand der Akkumulatoren
- Restspannung der Akkupacks
- Lade- Entladestrom
- Momentane Leistung des Elektromotors
- Momentaner Energieverbrauch des Sportmotorrads pro km
- Reichweite

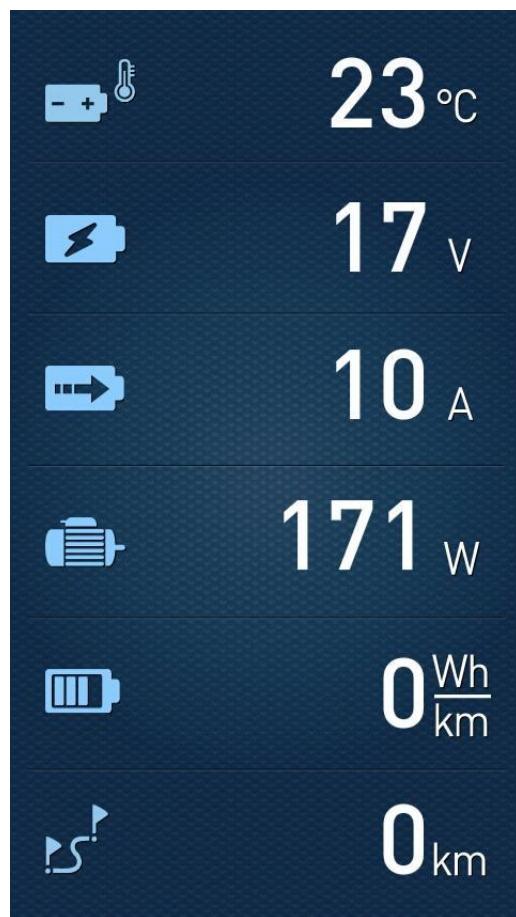


Abbildung V.19: Hauptansicht der EMUS EVGUI App[[**HauptansichtApp**]]

Sobald man in dem Hauptansichtsfenster einen beliebigen Unterpunkt anklickt, öffnet sich ein Unterfenster, wo man die Anzeige detaillierter erkennt.



Abbildung V.20: Detaillierte Ansicht der EMUS EVGUI App[[**DetaillierteApp**]]

3.3.1 Konfiguration der EMUS EVGUI App

Wie bereits vorher beschrieben, gibt es zwei verschiedene Ansichtsarten. In der detaillierten Ansicht gibt es im Linken unteren Eck einen Button mit dem Namen Menü.

Klickt man diesen Button an, scheinen 4 weitere Unterpunkte auf.

- Battery Status
- Bms Status
- Event Log
- Settings



Abbildung V.21: Menü der EMUS EVGUI App[[MenüApp]]

Die wichtigsten zwei Punkte sind hierbei der Battery Status und der Bms Status. Auf diese zwei Punkte wird mithilfe von Bildern in den folgenden Seiten etwas genauer eingegangen.

Battery Status:

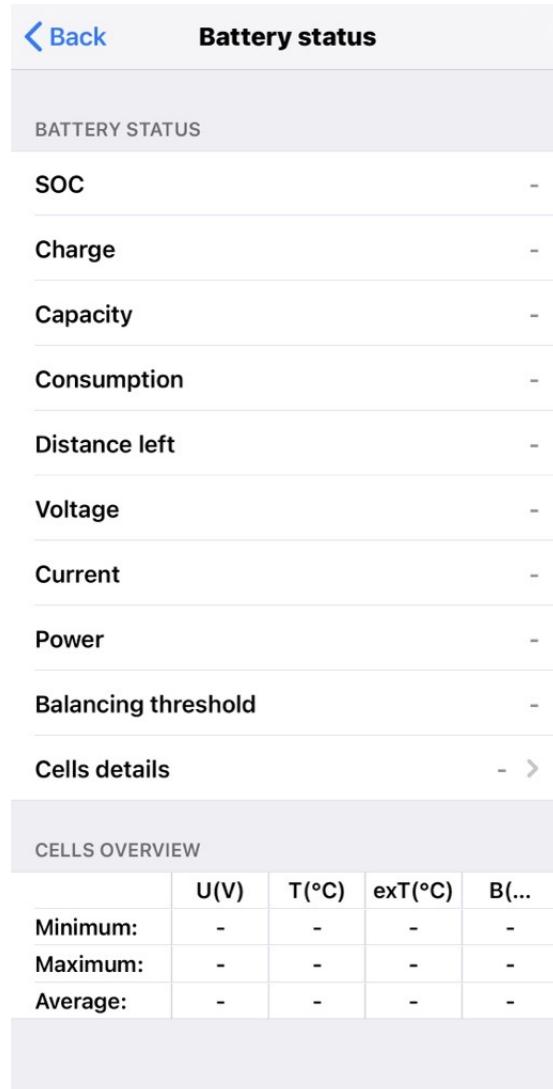


Abbildung V.22: Battery Status Anzeige der EMUS EVGUI App[[BatteryStatusApp]]

Bms Status:

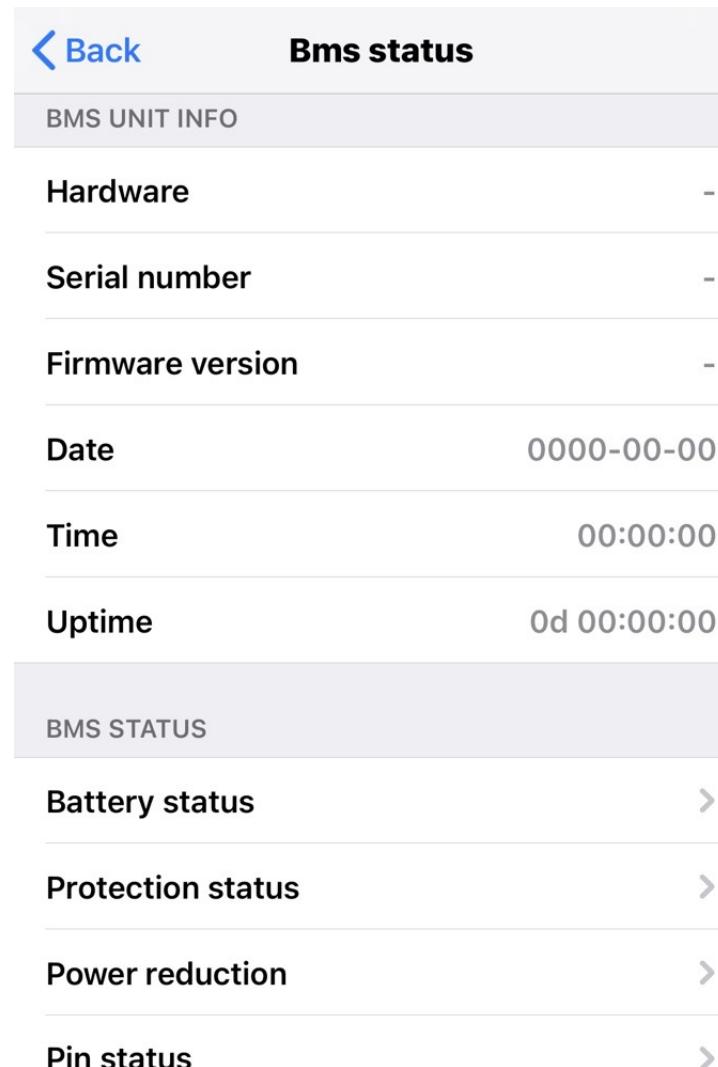


Abbildung V.23: BMS Status Anzeige der EMUS EVGUI App (Seite 1)[[BM]]



Abbildung V.24: BMS Status Anzeige der EMUS EVGUI App (Seite 2)

In der Kategorie BMS Status kann man wiederum verschiedene Unterpunkte auswählen um diese im Detail anzeigen zu lassen.

- Battery Status
- Protection Status
- Power Reduction
- Pin Status

3.4 Laderegelung

Die Steuerung des Ladevorgangs ist eine der wichtigsten Aufgaben eines Batteriemanagementsystems. Es ist von besonderer Bedeutung, diesen Ladevorgang richtig durchzuführen um einen sicheren Betrieb der Lithiumakkumulatoren zu gewährleisten. Die Auswahl eines bestimmten Ladegeräts sollte immer unter der Berücksichtigung verschiedener Aspekte der Anwendung durchgeführt werden. Die Ausgabeparameter sind aus der Sicht des BMS am wichtigsten. Die maximale Ausgangsspannung sollt geringfügig höher sein als die von der Zelle angegebene Ladespannung multipliziert mit der Anzahl der in Reihe geschaltenen Zellen bei maximaler Leistung. Der Strom sollte den maximalen Ladestrom nie überschreiten. In den folgenden Kapiteln werden die verschiedenen Ladegerättypen, die von der EMUS BMS Mini unterstützt werden, beschrieben. Grundsätzlich gibt es drei verschiedene Ladetypen die mit diesem Batteriemanagementsystem kompatibel sind.

- CAN based chargers
- Non Can based chargers
- Analog signal-controlled chargers

Bei unserem Projekt entschieden wir uns für ein Can basiertes Ladegerät. Bei dieser Methode der Laderegelung wird ein Ladegerät verwendet, dass zusätzlich mittel CAN-Bus mit dem BMS verbunden wird. Diese Methode ist mit Abstand am effizientesten, da das Batteriemanagementsystem auf den Ladestrom sowie auch auf die Ladespannung zugreifen bzw. vorgeben kann. Außerdem ist dies die schnellste und sicherste Methode die Akkumulatoren zu laden. Zur Anwendung ist schlussendlichen ein TC Can-Charger gekommen, da dieser kompatibel mit dem EMUS BMS Mini ist.

3.4.1 Verschaltung

Das Ladegerät wird mithilfe des CAN-Buses mit dem Batteriemanagementsystem verbunden. Diese Verbindung dient jedoch nur zur Kommunikation für die Regelung. Das Ladegerät und die Akkumulatoren werden jedoch über das BMS verbunden. Die grau markierte Linie in dem folgenden Bild sollte keine Leitung darstellen. Es sollte nur zeigen, dass sich der 6-polige Stecker direkt auf dem Batteriemanagementsystem befindet.

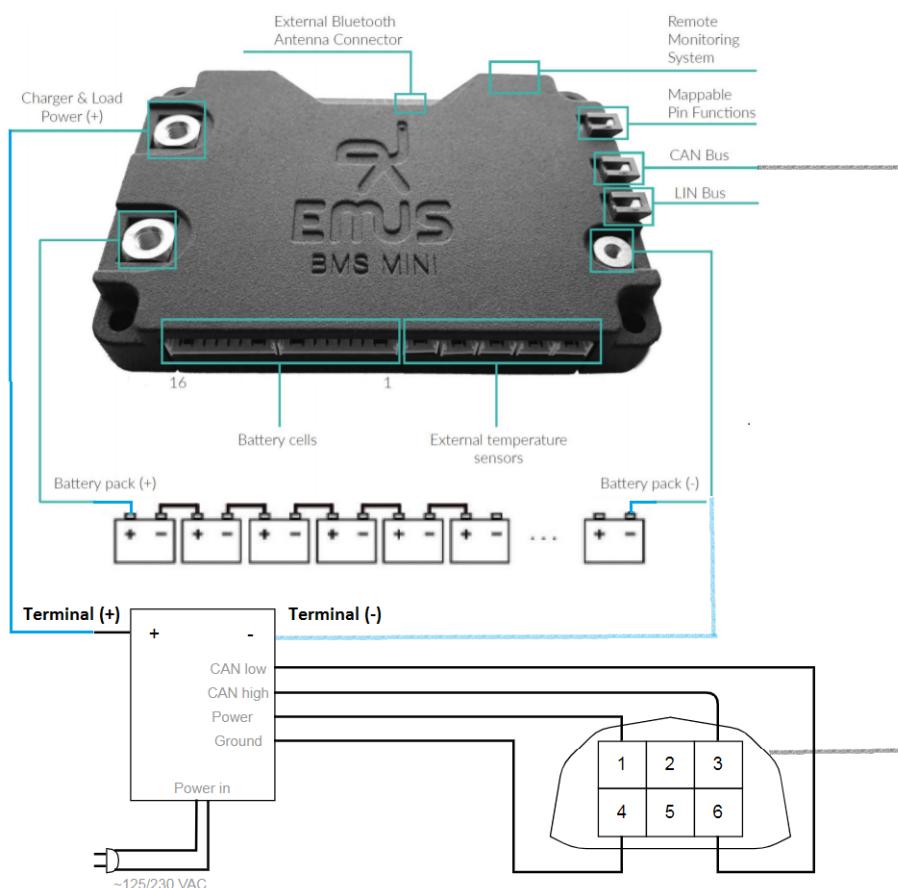


Abbildung V.25: Verschaltung des Ladegeräts über das BMS

Kapitel VI

Antriebsstrang

1 Übersicht

Die Hauptaufgabe des Antriebssystems ist die Umwandlung der, vom Akkumulator zur Verfügung gestellten, elektrischen Energie in die kinetische Antriebsenergie. Diese tritt zuerst rotatorisch am Motor auf und wird zunächst über das Direkt-Getriebe umgeformt bzw. auf die passende Drehzahl gebracht, anschließend wird die Rotationsenergie 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, sie fungiert als Bindeglied zwischen dem Human-Computer Interacting System und den elektrischen Anforderungen an das Gesamtsystem.

1.1 Grundfunktionen des Systems

Die geplanten Aufgaben des Antriebssystems lassen sich grob in zwei Grundfunktionen einteilen:

- **Der Antrieb**

Translation ist eine Grundfunktion eines jeden Verkehrsmittels.

Durch die Umwandlung der elektrischen Energie in kinetische Energie erfährt das gesamte System eine Beschleunigung in Fahrtrichtung.

- **Die Steuereinheit**

Steuerung und Kommunikation mit anderen Betriebsmitteln,
realisiert durch In- und Outputs, Datenübertragung mithilfe des CAN-Busses.

Um auf die einzelnen Details des Antriebssystems besser eingehen zu können, unterscheiden wir zwischen dem Hardwareaufbau und dem Softwareaufbau des Antriebssystems.

2 Hardwareaufbau des Antriebssystems

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

- **Mechanische Umsetzung (Kraftübertragung und Montage)**
Umfasst das Getriebe und die Befestigung aller Komponenten am Rahmen.
- **Der Laststromkreis (2.2)**
Beinhaltet die Verbindung des Motorcontrollers mit dem Motor und dem Akkumulator.
- **Der Steuerstromkreis (2.3)**
Beinhaltet alle elektrischen Verbindungen, welche mithilfe des 35-poligen Niederleistungs-Steckers mit dem Motorcontroller verbunden sind.

2.1 Mechanische Umsetzung

Die Fertigung des Getriebes und die Montage der einzelnen Betriebsmittel wurde vollständig von Tobias Schmeisser übernommen. (Kapitel: IV)

2.2 Der Laststromkreis

Der Laststromkreis beinhaltet alle leistungsführenden Betriebsmittel des Antriebssystems. Hierbei unterscheiden wir zwischen den zwei wichtigsten Grundfunktionen:

- **Elektrische Energieübertragung [2.2.1]**

Umfasst die elektrische Verbindung von Motor, Motorcontroller und Akkumulator. Realisiert durch einfache Leitungen, um Leistungen übertragen zu können.

- **Schutz der Komponenten vor Beschädigungen (Leitungsschutzorgane [2.2.2])**

Beinhaltet eine Schmelzsicherung zum Schutz vor Überströmen und ein Hochleistungs-Relais, um im Fehlerfall den Laststromkreis öffnen zu können und damit eine galvanische Trennung des Antriebs und der Energieversorgung gewährleisten zu können.

Im folgenden Bild ist der Grundaufbau des Laststromkreises ausführlich beschrieben:

R1 ... Vorladewiderstand

F1 ... Schmelzsicherung

S4.1 ... Hauptschütz (Hochleistungs-Relais)

S4.2 ... Vorladeschütz

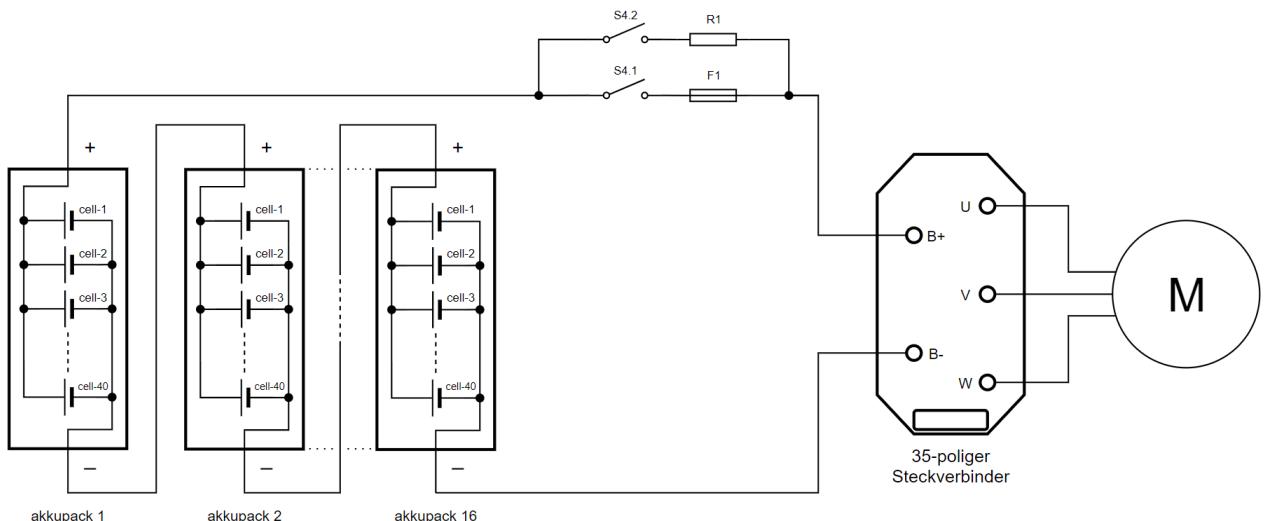


Abbildung VI.1: Grundaufbau des Laststromkreises [E.1]

2.2.1 Elektrische Energieübertragung

Um die benötigte elektrische Energie übertragen zu können, müssen die Leitungen an den Leistungsverbrauch des Verbrauchers (Motor) angepasst werden. Bei einer zu hohen Stromaufnahme (Überlast) des Motors kann es zu einer übermäßigen Erwärmung der Leitungen bis hin zu dauerhaften Beschädigungen, wie Durchschmoren der Isolierung, oder sogar einen Leitungsbrand führen. Um dies verhindern zu können, müssen die Leitungen an die Stromaufnahme des Motors angepasst werden. Das heißt, der zulässige Dauerstrom der Leitungen muss den maximalen Dauerstrom des Motors bzw. den maximalen Dauerstrom, welcher durch den Akkumulator zur Verfügung gestellten werden kann, übersteigen.

Auswahlkriterien:

Bei der Auswahl von Leitungen muss man grundsätzlich den Querschnitt und die Länge der Leitung an die benötigte Stromaufnahme des Motors anpassen. Bestimmte weitere Anforderungen, wie zum Beispiel ein maximal zulässiger Spannungsabfall, müssen ebenfalls berechnet und berücksichtigt werden. Ein gutes Zusammenspielen der Leitungen mit den anderen Leitungsschutzeinrichtungen ist ebenfalls sehr wichtig, jedoch können diese Bauteile auch sehr gut an die bereits festgelegten Eigenschaften der Leitungen angepasst werden.

Berechnung:

In diesem Abschnitt befassen wir uns zunächst mit der Berechnung aller Ströme, die wir zur Auswahl der richtigen Leitungen benötigen. Da wir den maximalen Strom, welcher durch eine Zelle zur Verfügung gestellt werden kann, bereits aus dem Datenblatt kennen, können wir von der Seite des Akkumulators mit der Berechnung beginnen. Zuerst berechnet man also den maximalen Strom, welcher sich aus der Parallelschaltung der Zellen ergibt. Der Motor erhält nahezu die selbe Leistung, weshalb der Motorstrom durch die Anordnung der Phasen ebenfalls bestimmt werden kann. Da die Größenordnungen der beiden Ströme im Akkumulator-Stromkreis und im Motor-Stromkreis sehr unterschiedlich sind, können ebenfalls verschiedene Leitungsquerschnitte verwendet werden. Die Berechnung des Leitungswiderstands lässt nun auch auf den Spannungsabfall schließen, diese Berechnung muss jedoch bei beiden Stromkreisen erfolgen.

- **Berechnung der Ströme:**

Maximaler Strom, der von einer Zelle zur Verfügung gestellt werden kann: 14 A
Anzahl der Zellen die in einem Akkupack parallel verschaltet werden: 40

- **Akkumulator-Stromkreis:**

Querschnitt $A_A = 70 \text{ mm}^2$
Leitungslänge $l_A = 1\text{m}$
Spezifischer Widerstand von Aluminium: 0,0278

- **Motor-Stromkreis:**

Querschnitt $A_M = 35 \text{ mm}^2$
Leitungslänge $l_M = 1\text{m}$
Spezifischer Widerstand von Kupfer: 0,01786

Berechnung der Ströme:

$$I_{Zmax} = 14 \text{ A}$$

$$I_{Amax} = 40 \cdot I_{Zmax} = 40 \cdot 14 \text{ A} = 560 \text{ A}$$

$$I_{Mmax} = \frac{I_{Amax}}{\sqrt{3}} = \frac{560 \text{ A}}{\sqrt{3}} = 323 \text{ A}$$

Spannungsabfall Akkumulator-Stromkreis:

$$R_A = \frac{\delta_A \cdot l_A}{A_A} = \frac{0,0278 \frac{\Omega \cdot \text{mm}^2}{\text{m}} \cdot 1 \text{ m}}{70 \text{ mm}^2} = 0,397 \text{ m}\Omega$$

$$\Delta U = R_A \cdot I_{Amax} = 0,397 \text{ m}\Omega \cdot 560 \text{ A} = 222 \text{ mV}$$

$$U_{VB} = U_Q - \Delta U = 50,4 \text{ V} - 222 \text{ mV} = 50,178 \text{ V}$$

Spannungsabfall Motor-Stromkreis:

$$R_M = \frac{\delta_K \cdot l_M}{A_M} = \frac{0,01786 \frac{\Omega \cdot \text{mm}^2}{\text{m}} \cdot 1 \text{ m}}{35 \text{ mm}^2} = 0,51 \text{ m}\Omega$$

$$\Delta U = R_M \cdot I_{Mmax} = 0,51 \text{ m}\Omega \cdot 323 \text{ A} = 165 \text{ mV}$$

$$U_{VB} = U_Q - \Delta U = 50,4 \text{ V} - 165 \text{ mV} = 50,235 \text{ V}$$

Fazit:

Bei der Anwendung als E-Motorrad spielen der Widerstand und der damit verbundene Spannungsabfall eigentlich keine Rolle, da die Leitungslängen bezogen auf den Querschnitt klein sind. Hinzu kommt noch, dass die Leitungslängen in dieser Berechnung sicherheitshalber größer angenommen wurden, als sie in Wirklichkeit umgesetzt werden, da die Leitungslängen vorab schwer abgeschätzt werden können und nach der Montage ebenso variieren werden. Der Curtis Controller ist ebenfalls sehr unempfindlich gegenüber Spannungsschwankungen, welche so oder so durch das Laden und Entladen des Akkumulators entstehen.

Der Aluminiumleiter verfügt über eine maximale Dauerbelastung von 160A, was deutlich unter dem Maximalstrom liegt. Bei unserer Anwendung als Motorrad ist die Dauerbelastung aber eher unwichtig, da nur kurzzeitig hohe Leistungen benötigt werden (Beschleunigungsvorgang). Weiters wird der vom Motor angeforderte Strom dauerhaft vom Batterie-Management-System und vom Curtis Controller überprüft und reguliert bzw. im Notfall abgeschaltet. Kommt es zu starken Erwärmungen im Motor, der Motorsteuerung oder dem Akkumulator, wird die Maximalleistung gedrosselt.

2.2.2 Leitungsschutzorgane

Die Aufgabe der Leitungsschutzorgane ist es, bei unerwarteten Überströmen oder in einem Fehlerfall den Laststromkreis zu öffnen, um damit den Motor bzw. Motorcontroller vom Akkumulator galvanisch zu trennen. Diese Maßnahme wird ergriffen, um mögliche Beschädigungen an den Komponenten oder an den Leitungen verhindern zu können. Da jedoch unerwünschte Fehlauslösungen zum sofortigen Stillstand des Motorrades führen und eventuell sogar benötigte Wartungen (Wechsel der durchgebrannten Schmelzsicherung) nach sich ziehen, müssen diese Leitungsschutzorgane sehr sorgfältig ausgewählt werden. Eine Überdimensionierung ist ebenso unerwünscht, denn dadurch steigen die Anschaffungskosten der Bauteile. Das eher größere Problem entsteht jedoch bei der Überdimensionierung der Schmelzsicherung, denn diese löst nun zu spät aus und hat damit nur mehr eine nicht geeignete Schutzfunktion.

Hochleistungs-Relais:

Bei der Auswahl des Hochleistungs-Relais muss vor allem der maximale Strom, der vom Relais geschaltet werden kann, höher als der Verbraucherstrom bei maximaler Auslastung sein. Ebenfalls werden die zu erwarteten Lebenszyklen mithilfe einer speziellen Kennlinie abgeschätzt, welche diese Zyklen abhängig von bestimmten Strom- und Spannungswerten angibt. Das im Endeffekt ausgewählte Bauteil stammt von der Firma „TE Connectivity / Raychem“ und hat die Bezeichnung „KILOVAC LEV200 A4ANA“. Bei diesem Hochleistungs-Relais ergibt sich bei einer Gleichspannung von 120V und einem Strom von 600A eine ungefähre Lebenszeit von 5.000 Schaltvorgängen. Die oben genannten Werte sind aber entsprechend der Kennlinie deutlich höher als die realen Leistungswerte, zudem ist der Schaltvorgang meist nahezu unbelastet, da die Kondensatoren vorgeladen werden und das Hochleistungs-Relais nur in einem Fehlerfall während des Betriebs geöffnet wird. Aufgrund dessen kann auf eine Lebenszeit von deutlich über 10.000 Schaltvorgängen geschlossen werden. Natürlich muss auch die Spulenspannung und der zugehörige Leistungsverbrauch für die gewünschte Anwendung passen. Die Spulenspannung wurde passend zu den anderen Bauteilen für 12V ausgewählt, der Leistungsverbrauch befindet sich im Bereich von wenigen Watt und ist damit vernachlässigbar¹.

Hochgeschwindigkeits-Schmelzsicherung:

Bei der Auswahl der Schmelzsicherung ist es vor allem wichtig, dass der Stromkreis bei einem Kurzschlussfall in kurzer Zeit unterbrochen wird, die zulässigen Nennströme jedoch dauerhaft geleitet werden können. Das bedeutet, der Kurzschlussstrom muss um ein Vielfaches größer sein, als der Nennstrom der Schmelzsicherung. Da jedoch die Kurzschlussimpedanz im Bereich von wenigen Milliohm liegt, ist der Kurschlussstrom mindestens 10 mal so groß wie der Nennstrom der Sicherung, was bei der ausgewählten Schmelzsicherung „FWA-400B“ von der Firma „Bussmann / Eaton“ eine Ausschaltzeit von wenigen Millisekunden bedeutet. Normalerweise ist es ebenfalls in Bezug auf die Leitungen wichtig, dass der zulässige Dauerstrom der Leitungen größer als der Nennstrom der Schmelzsicherung ausgewählt wird, um Beschädigungen an den Leitungen verhindern zu können. In unserem Fall ist dies aber eher nebensächlich, da der vom Motor angeforderte Strom, wie bereits erwähnt, dauerhaft durch das Batterie-Management-System überprüft und begrenzt wird².

Fazit:

benötigt?

¹Hochleistungs-Relais Datenblatt: [2.2]

²Hochleistungs-Schmelzsicherung Datenblatt: [2.3]

2.2.3 Motorbeschreibung

Die Nennleistung von Motor und Motorcontroller übersteigen die Nennleistung des Akkumulators, weshalb die Nennwerte dieser Bauteile unwichtig für die Dimensionierung der Leitungen und Leitungsschutzorgane ist. Da jedoch Motor und Motorsteuerung die zentralen Elemente des E-Motorrades darstellen, werden unter diesem Punkt bestimmte Eigenschaften und Nenndaten genauer beschrieben.

Für die Anwendung in einem E-Motorrad bot sich die Verwendung eines bürstenlosen permanenterregten Gleichstrommotors an. Da jedoch eine Motorsteuerung die drei Phasen des Motors mit Drehstrom versorgt, ist dieser Motor eigentlich eine Synchronmaschine mit einem Permanentmagneten im Läufer.

Der Motor wurde von der Firma Ashwoods gebaut und hat die Bezeichnung „IPM-200-50“. Er besitzt eine mechanische Spitzenleistung von 16 kW und ein maximales Drehmoment von 74 Nm. Die maximale Drehzahl beträgt 8500 U/min und der Wirkungsgrad liegt bei circa 85%³.

Die Motorsteuerung wurde von der Firma Curtis gebaut, der Controller ist ein Prototyp mit der Modellnummer „AC-F4-A-Proto-002“. Er besitzt einen maximalen Motorstrom von 450A RMS. Im Betrieb hat der Motorcontroller einen Leistungsverbrauch von rund 500W⁴.

Betrachtet man also beide Bauteile gemeinsam, ergibt das eine maximale elektrische Leistung von 19,5 kW, sprich einen maximalen Strom von 385A bei einer Nennspannung von 50,4V.

$$P_{max} = P_{Motor} + P_{Controller} = \frac{16 \text{ kW}}{0,85} + 500 \text{ W} \approx 19,5 \text{ kW}$$

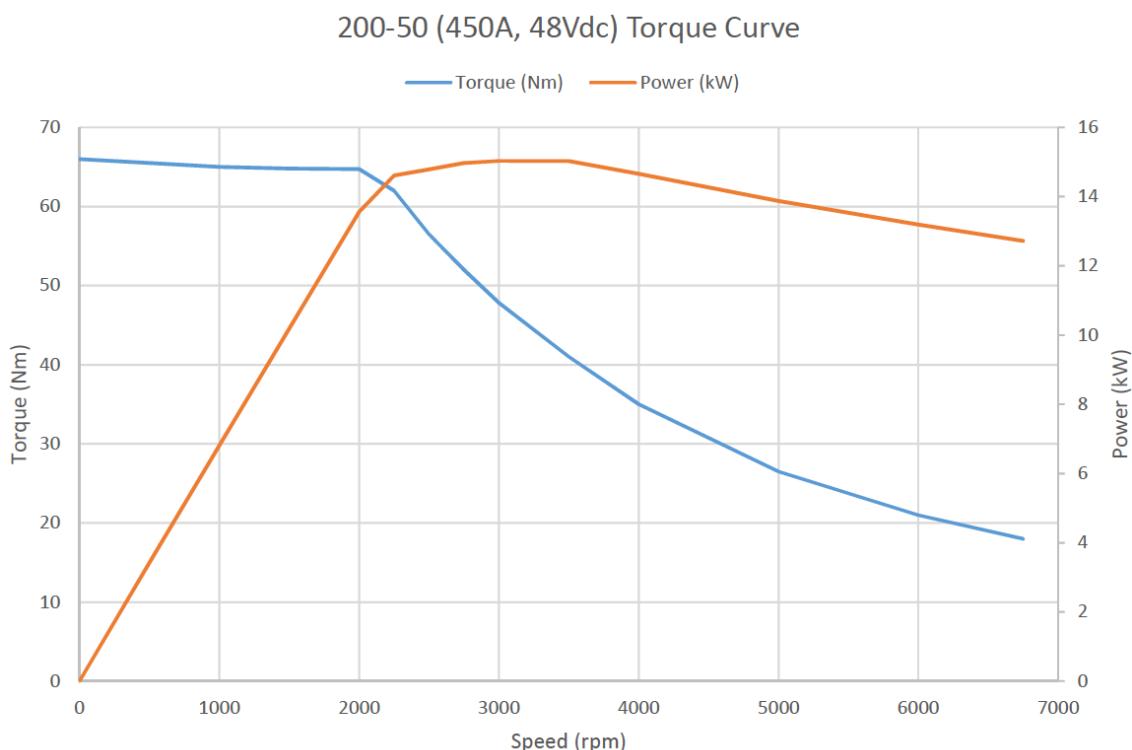


Abbildung VI.2: Ashwoods-Motor Kennlinie

³Ashwoods Motor CAD-Zeichnung: [C.12]

⁴Curtis Controller CAD-Zeichnung: [C.13]

2.3 Der Steuerstromkreis

2.3.1 Übersicht Ein- und Ausgänge

Der Steuerstromkreis umfasst alle elektrischen Verbindungen, welche über den 35-poligen Niederleistungs-Stecker mit dem Motorcontroller verbunden sind. Hierbei unterscheiden wir grundsätzlich zwischen drei verschiedenen Ports, welche nochmals unterkategorisiert werden können:

- **Eingänge (Inputs)**

- Digitale Eingänge (Digital Inputs [2.3.2])
- Analoge Eingänge (Analog Inputs [2.3.3])
- Gas- und Bremseingänge (Throttle and Brake Inputs [2.3.4])
- Positionsrückmeldung vom Encoder (Position-feedback Input [2.3.5])
- Prozessorschaltung und Spulenrücklauf (KSI and Coil Return [2.3.6])

- **Ausgänge (Outputs)**

- Analoge Ausgänge (Analog Outputs [2.3.7])
- Digitale und Pulsweltenmodulierbare Ausgänge (Digital and PWM Outputs [2.3.8])
- Spannungsversorgungs-Ausgänge (Power Supply Outputs [2.3.9])

- **Kommunikation (Communication [2.3.10])**

- CAN-Bus (CAN-Port)
- Serielle Schnittstelle (Serial-Port)

Der Motorcontroller verfügt über viele Pins, welche über mehrere Funktionen verfügen, es muss jedoch eine dieser Funktionen ausgewählt werden. Pin 6 zum Beispiel wird eigentlich als digitaler und phasenmodulierbarer Ausgang verwendet, bei richtiger Konfiguration kann er jedoch auch als digitaler Input verwendet werden. Weiteres kann frei konfiguriert werden, ob man mit diesem Ausgang zum Beispiel das Hochleistungs-Relais, eine Pumpe oder Bremslichter ansteuern möchte. Je nach gewünschter Anwendung gibt es auch unterschiedliche, für diese Applikation vorgefertigte Funktionen. Um den passenden Pin für eine Anwendung auswählen zu können, muss man jedoch die elektrischen Eigenschaften der Pins genauer unter die Lupe nehmen. Oftmals haben auch die Pins der selben Unterkategorie verschiedene Funktionen, Eingangsimpedanzen oder Toleranzen.

In dieser Abbildung kann man den prinzipiellen Aufbau des Steuerstromkreises genauer betrachten. Die einzelnen Ein- und Ausgänge (Schalter und Induktivitäten) sind mit Ausnahme von Encoder, Throttle-Pot und CAN-Bus nur mit der allgemeinen Bezeichnung des Curtis Controllers versehen. Da der Curtis Controller ein umfassendes Einsatzgebiet abdeckt, werden nicht alle Funktionen bzw. Pins für die Anwendung als Motorrad benötigt, somit fällt die Verdrahtung dieser Pins weg. Die zugehörigen Eigenschaften und ausgewählten Funktionen der einzelnen Pins kann man auf dieser Abbildung nicht erkennen, diese sind aber in folgenden Punkten erklärt.

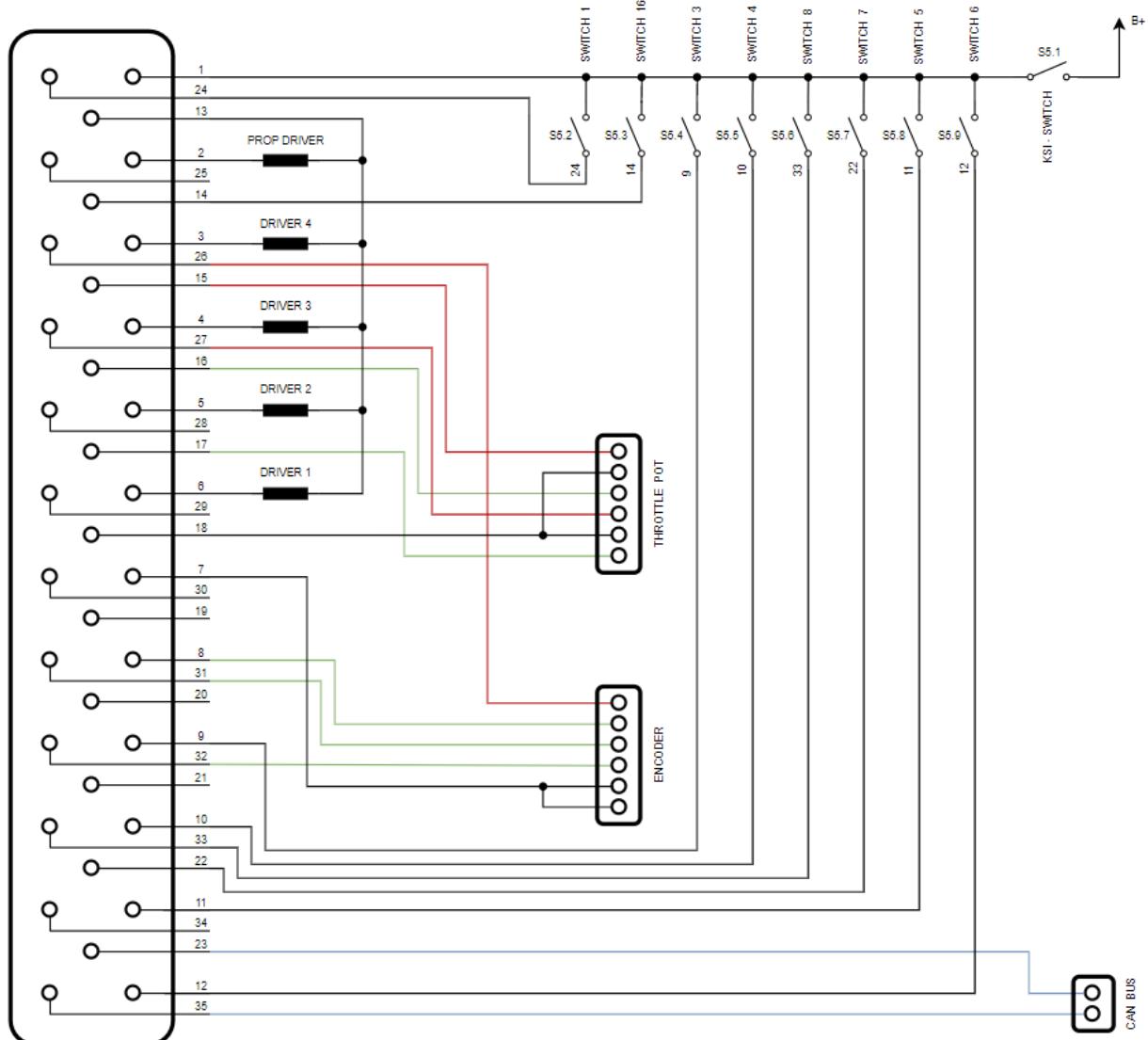
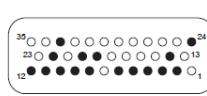


Abbildung VI.3: Grundaufbau des Steuerstromkreises [E.2]

2.3.2 Digitale Eingänge (Digital Inputs)

Es gibt insgesamt 16 Pins, die als digitale Eingänge genutzt werden können, jedoch werden die letzten sieben Pins davon eigentlich für die Verwendung als Ausgänge konfiguriert. Der Switch 1 wird im Normalfall für die Funktion „Emergency Reverse“ verwendet, da die Realisierung dieser Funktion bei einem Motorrad aber eher unwichtig ist, wird der zugehörige Pin 24 vorerst nicht verwendet (wird eventuell bei späteren Verbesserungen hinzugefügt). Switch 2 wird als analoger Eingang verwendet und ist in diesem Punkt genauer erklärt. Switch 3 wird für die Betätigung von Interlock verwendet. Interlock wird vom Raspberry PI angesteuert und muss aktiviert sein, damit der Motorcontroller in den Fahrbetrieb wechseln kann. Der NOT-Aus befindet sich auf der Lenkstange und steuert den Switch 4 an, bei Betätigung wird jeglicher Betrieb sofort eingestellt. Der Schalter zur Auswahl des gewünschten Antriebsmodus befindet sich ebenfalls auf der Lenkstange und ist mit dem Switch 5 verbunden. Die Parameter „Forward“ und „Reverse“, welche die Auswahl der gewünschten Fahrtrichtung ermöglichen, werden normalerweise auf Switch 7 und 8 gelegt.

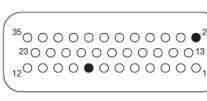


DIGITAL INPUT SPECIFICATIONS						
Name	Pin	Logic Thresholds	Input impedance*	Voltage range†	ESD Tolerance	
Switch 1	24	Rising edge= 4.4V max Falling edge= 1.5V min	48-80V models: 25.2 kΩ, 27.3 kΩ	-10V to (MaxV + 10 V)	$\pm 8 \text{ kV}$ (direct strike)	
Switch 2	8					
Switch 3	9					
Switch 4	10					
Switch 5	11					
Switch 6	12					
Switch 7	22					
Switch 8	33			-5V to (MaxV + 10 V)		
Switch 16	14					
Digital Out 6	19					
Digital Out 7	20					
Driver 1	6					
Driver 2	5					
Driver 3	4					
Driver 4	3					
Prop Driver	2					

Abbildung VI.4: Digital Input Specifications [[Manual]]

2.3.3 Analoge Eingänge (Analog Inputs)

Es gibt insgesamt zwei Pins, die als analoge Eingänge verwendet werden können. Analog 2 wird hierbei im Normalfall für den Motortemperatur-Sensor verwendet. Die Eingänge, die für das Gas- und Bremspotentiometer verwendet werden, sind in dieser Kategorie nicht aufgelistet, obwohl diese ebenfalls als analoge Eingänge genutzt werden. Diese Pins sind jedoch speziell für die Gas- und Bremssteuerung konfiguriert und sollten im Normalfall auch dafür verwendet werden.

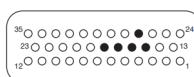


ANALOG INPUT SPECIFICATIONS					
Signal Name	Pin	Operating Voltage	Input impedance*	Protected Voltage	ESD Tolerance
Analog 1	24	0 to 10V in 1024 steps	48-80 V models: 23.8 kΩ, 28.1 kΩ	-10 V to (MaxV + 10 V)	$\pm 8 \text{ kV}$ (direct strike)
Analog 2	8				

Abbildung VI.5: Analog Input Specifications [[Manual]]

2.3.4 Gas- und Bremseingänge (Throttle and Brake Inputs)

Die zwei Gas- und Bremssteuerungs-Eingänge sind wegen der speziellen Auslegung von den analogen Eingängen abgegrenzt und können unabhängig voneinander programmiert werden. Sie sind optimiert für die Anwendung mittels Spannungssteuerung, 2-Draht Widerstandssteuerung oder 3-Draht Widerstandssteuerung. Bei der Spannungssteuerung benötigt man die Pins Pot Wiper und I/O Ground, bei der 2-Draht Widerstandssteuerung Pot Wiper und Pot Low und bei der 3-Draht Widerstandssteuerung Pot High, Pot Wiper und Pot Low. In unserem Fall benutzen wir beide Steuerungs-Eingänge für die 3-Draht Widerstandssteuerung, da der Gasdrehgriff über eine Drahtbrucherkennung verfügt. Das heißt, der Gasdrehgriff hat zwei unabhängige 3-Draht Potentiometer-Ausgänge verbaut, welche beide für den Gaseingang benutzt werden.



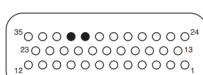
THROTTLE INPUT SPECIFICATIONS

Signal Name	Pin	Operating Voltage	Input Impedance	S/Sink Current	Protected Voltage	ESD Tolerance
Throttle Pot High	15	0 V (shorted to Pot Low) 5 V (open circuit)	N/A	1 mA nominal (source)	-0.5 V to (MaxV + 10 V)	$\pm 8 \text{ kV}$ (direct strike)
Pot2 High	27					
Throttle Pot Wiper	16	0 to 6.25 V	100 k Ω min	0.76 mA nominal (source, 2-wire)		
Pot2 Wiper	17					
Pot Low	18	0 to 0.25 V	20 Ω nom.	Faults if above 15 mA (sink)	-1 V to (MaxV + 10 V)	

Abbildung VI.6: Throttle Input Specifications [[Manual]]

2.3.5 Positionsrückmeldung vom Encoder (Position-Feedback Input)

Diese zwei Pins sind intern dafür konfiguriert, die aktuelle Position der Motorwelle einzulesen, um eine optimale feldorientierte Ansteuerung des Motors durchführen zu können. Dabei gibt es die Möglichkeiten über einen Quadratur-Encoder oder einen Sin/Cos-Encoder. Da im Ashwoods-Motor ein Sin/Cos-Sensor verbaut ist, wurde dies vorab beim Motorcontroller eingestellt.



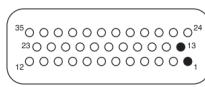
SIN/COS SENSOR INPUT SPECIFICATIONS

Signal Name	Pin	Operating Voltage	Input Impedance	Max. Frequency	Protected Voltage	ESD Tolerance
Position Feedback A	31	0 to 5 V	150 k Ω for voltages \leq 5 V 75 k Ω for voltages $>$ 5 V	500 Hz	-5 V to (MaxV + 10 V)	$\pm 8 \text{ kV}$ (direct strike)
Position Feedback B	32					

Abbildung VI.7: Sin/Cos Sensor Input Specifications [[Manual]]

2.3.6 Prozessorversorgung und Spulenrücklauf (KSI and Coil Return)

Der KSI-Eingang stellt die elektrische Versorgung aller Niederleistungs-Schaltkreise zur Verfügung. Dies beinhaltet ebenfalls die Versorgung aller Ausgänge und die Kondensator-Vorlade-Funktion, welche dazu dient, die Kondensatoren über einen Widerstand vorzuladen, um hohe Einschaltströme zu verhindern. Der Spulenrücklauf stellt die Versorgung der pulsweitenmodulierbaren Ausgänge zur Verfügung und hat die gleiche Spannung wie der KSI-Pin. Die elektrische Trennung von KSI und Coil Return muss jedoch aufrechterhalten werden, um einen Verpolungsschutz gewährleisten zu können.

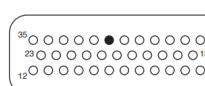


KSI and COIL RETURN INPUT SPECIFICATIONS					
Signal Name	Pin	Operating Voltage	Input Current	Protected Voltage	ESD Tolerance
KSI	1	Between under- and overvoltage cutbacks	13 A max * continuous	$\pm (\text{MaxV} + 10 \text{ V})$	$\pm 8 \text{ kV}$ (direct strike)
Coil Return	13		10 A or 12 A max **	(KSI - 0.3 V) to (MaxV + 10 V)	

Abbildung VI.8: KSI and Coil Return Input Specifications [[Manual]]

2.3.7 Analoge Ausgänge (Analog Outputs)

Der analoge Ausgang kann ein Spannungssignal von 0 bis 10V ausgeben. Dieser Ausgang ist für die Datenausgabe über bestimmte Anzeigegeräte vorgesehen, wie zum Beispiel eine Anzeige für den aktuellen Ladestand des Akkumulators, vorgesehen. Da alle benötigten Fahrdaten mithilfe des Touchdisplays visualisiert werden, wird dieser Pin nicht verwendet.

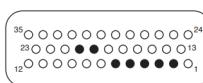


ANALOG OUTPUT SPECIFICATIONS					
Signal Name	Pin	Output Voltage	Output Current	Protected Voltage	ESD Tolerance
Analog Out	30	0 to 10 V	10 mA	-1 V to (MaxV + 10 V)	$\pm 8 \text{ kV}$ (direct strike)

Abbildung VI.9: Analog Output Specifications [[Manual]]

2.3.8 Digitale und Pulsweitenmodulierbare Ausgänge (Digital and PWM Outputs)

Es gibt insgesamt sieben digitale Ausgänge, wovon jedoch nur die ersten fünf für eine Pulsweitenmodulation konfiguriert werden können. Diese Ausgänge sind für induktive Lasten, wie zum Beispiel den Hauptschütz oder eine elektromagnetische Bremse, vorgesehen. Rein ohm'sche Lasten können ebenfalls angesteuert werden, jedoch darf der zulässige Spitzenstrom nicht überschritten werden. Der Proportional-Driver kann bei richtiger Konfiguration auch für die Anzeige eines Tachometers oder ähnlichem hergenommen werden. Generell kann jeder Pin dieser Gruppe ebenfalls als digitaler Eingang benutzt werden.



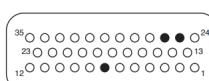
DIGITAL and PWM OUTPUT SPECIFICATIONS

Name	Pin	PWM	PV Current	Frequency	Output Current	Protected Voltage	ESD Tolerance
Driver 1	6	0 to 100% Duty Cycle	N/A	120 to 1000 Hz *	2A Max	−0.5 V to (MaxV + 10 V)	± 8 kV (direct strike)
Driver 2	5				3A Max		
Driver 3	4				2A Max		
Driver 4	3		0 to 2A in 607 nominal steps	18 kHz			
Prop Driver	2	On / Off	N/A	N/A	1A Max		
Digital Out 6	19						
Digital Out 7	20						

Abbildung VI.10: Digital and PWM Output Specifications [[Manual]]

2.3.9 Spannungsversorgungs-Ausgänge (Power Supply Outputs)

Um kleine Schaltkreise, wie zum Beispiel einen LED-Indikator oder die Positionsrückmeldung des Encoders, mit Spannung versorgen zu können, gibt es zwei dafür vorgesehene Spannungsversorgungs-Ausgänge mit je einem Pin für 5V und 12V. Für diese Anwendungen gibt es ebenfalls noch einen Rücklauf, der als I/O Ground definiert wurde. Die 5V-Spannungsversorgung wird in unserem Fall für den Encoder verwendet, die 12V-Versorgung wird aufgrund des vorhandenen Spannungswandlers nicht benötigt.



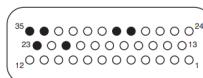
POWER SUPPLY OUTPUT SPECIFICATIONS

Signal Name	Pin	Output Voltage	Output Current	Protected Voltage	ESD Tolerance
+12 V Out	25	11.5 to 14.5 V	100 mA max for +12 Out 100 mA max for +5 Out 200 mA max (combined total)	−1 V to (MaxV + 10 V)	± 8 kV (direct strike)
+5 V Out	26	5 V ±10%			
I/O Ground	7	n/a	500 mA max	not protected	

Abbildung VI.11: Power Supply Output Specifications [[Manual]]

2.3.10 Kommunikations-Ports

Für die Kommunikation mit anderen Betriebsmitteln stellt der Motorcontroller zwei Möglichkeiten zur Verfügung, den CAN-Bus und die serielle Schnittstelle. Da sich unser Projektteam auf die Nutzung des CAN-Busses geeinigt hat, wird die serielle Schnittstelle nicht verwendet. Die zwei Pins „CAN Term High“ und „CAN Term Low“ werden ebenfalls nicht benötigt, denn diese dienen nur dazu, den CAN-Bus vorübergehend funktionsunfähig zu schalten. Programmtechnisch gibt es zwei Möglichkeiten zur Konfiguration des CAN-Busses, dies wird jedoch im Punkt Software genauer erklärt. Der CAN-Bus wird also über die Pins „CAN H“ und „CAN L“ realisiert, dieser wird mit dem Batterie-Management-System und der Zentralsteuerung verbunden.



COMMUNICATIONS PORT SPECIFICATIONS					
Signal Name	Pin	Supported Protocol / Devices	Data Range	Protected Voltage	ESD Tolerance
CAN H	23	CANopen, other 11-bit or 29-bit identifier protocols	up to 1 Mbit/s	–0.5 V to (MaxV + 10 V) (no connection to external wiring)	$\pm 8 \text{ kV}$ (direct strike)
CAN L	35				
CAN Term H	21				
CAN Term L	34				
Serial TX	28	Curtis 840 Display, 1313 Handheld Programmer, 1314 PC Programming Station	as required, 9.6 kbit/s to 56 kbit/s	–0.3 V to 12 V	
Serial RX	29				

Abbildung VI.12: Communications Port Specifications [[Manual]]

3 Softwareaufbau des Antriebssystems

Der Softwareaufbau des Antriebssystems kann grob in drei Grundfunktionen unterteilt werden:

- **Parameterbasierte Programmierung (Programmer [3.1])**

Umfasst alle Parameter und die vornehmbaren Konfigurationsmöglichkeiten.

- **Drehmomentsteuerung (Torquecontrol [3.2])**

Beinhaltet alle Parametereinstellungen, welche für die Drehmomentsteuerung ausschlaggebend sind.

- **Vehicle-Control-Language Programmierung (VCL [3.3])**

Umfasst das gesamte Programm, welches mit der Vehicle-Control-Language realisiert wurde. Ein Großteil dieses Programms beschäftigt sich hierbei mit der Kommunikation zwischen der Motorsteuerung und dem Raspberry PI.

3.1 Parameterbasierte Programmierung (Programmer)

3.1.1 Allgemeines

Im Curtis Integrated Toolkit befindet sich unter anderem die Funktion „Programmer“, welche sehr viele Applikationseinstellungen für die Controllerprogrammierung zur Verfügung stellt. Es gibt mehrere verschiedene Menüs, welche die jeweils zugehörigen Parametergruppen auflisten. Viele Variablen können konfiguriert werden, andere werden nur für die Ausgabe aktueller Werte, wie zum Beispiel die aktuelle Geschwindigkeit oder Temperatur, verwendet. Die Parameter können je nach Anwendung und Funktion verschiedene Einheiten und einstellbare Wertebereiche besitzen. Sehr vorteilhaft sind hierbei die meist vorhandenen Kurzbeschreibungen im Curtis Integrated Toolkit oder die Parameterdeklaration in der Anleitung.

3.1.2 Funktionen

Die folgenden Parametermenüs sind auf bestimmte Funktionen angepasst und optimiert worden. Muss eine Funktion angesteuert werden, so kann sie einen beliebigen Pin zugewiesen werden.

Throttle and Brake

Die Hauptfunktion dieser Parametergruppen ist die genaue Einstellung der Potentiometer-Eingänge für den Bremsvorgang, Vorwärts- und Rückwärtsbetrieb. Es können der Start- und Endwert des Potentiometer festgelegt werden, ebenso bestimmte Formänderungen der Eingangskurve, welche die Intensität des Gas- bzw. Bremsbefehls abhängig von der Potentiometerstellung beschreibt. Ebenso gibt es mehrere mögliche Sicherheitskonfigurationen, Filtereinstellungen und auch die Möglichkeit der Ansteuerung mittels VCL.

CAN Interface

Das CAN-Interface wird in unserer Anwendung nicht benötigt, denn die Kommunikation wurde mittels VCL-CAN umgesetzt. Die vorgefertigte Version CANopen bietet jedoch auch mehrere Einstellungsmöglichkeiten, wie ID und Taktfrequenzanpassungen. Ebenfalls gibt es jeweils 4 Datenpakete, welche für den Senden- und Empfangsvorgang konfiguriert werden können.

Battery Setup

In diesem Parametermenü kann die Nennspannung festgelegt werden, ebenfalls gibt es mehrere Einstellungsmöglichkeiten, welche in einem Über- oder Unterspannungsfall von Bedeutung sind. Um den Ladestand des Akkumulators besser abschätzen zu können, sollten hier auch bestimmte Eigenschaften festgelegt werden.

Main Contactor

Dieses Menü dient zur Konfiguration Hauptschütz (Hochleistungs-Relais). Die Ansteuerungsspannung wird hier genau beschrieben, nach dem Schließen des Schützes kann eine kleinere Spannung zum Halten der Kontakte verwendet werden. Ebenfalls kann eine Zeitverzögerung eingestellt werden, welche bestimmt, wie lange der KSI-Pin geöffnet sein darf, bevor der Hauptschütz geöffnet wird. Diese Funktion kann sehr wichtig sein, denn ein Wechsel des Antriebsmodus kann somit ebenfalls während des Betriebs erfolgen (Gaseingang muss aber 0% sein). Weiteres können die Vorladefunktion und auch mehrere sicherheitsrelevante Funktionstests eingeschaltet und konfiguriert werden.

Pump Contactor

Dieses Menü dient zur Konfiguration einer Pumpe für den Kühlkreislauf. Es kann eine erhöhte Versorgungsspannung im Anlaufbetrieb festgelegt werden.

EM Brake Control

Mit diesem Parametermenü kann eine zusätzlich verbaute, elektromagnetische Bremse angesteuert werden. Da in unserem Motorrad jedoch nur herkömmliche mechanische Bremsen verbaut sind, ist diese Funktion für unsere Anwendung nicht zu gebrauchen. Für die elektromagnetische Bremse gibt es jedoch eine große Menge an Konfigurationsmöglichkeiten, für verschiedene Ereignisse können bestimmte Drehmoment- und Zeitwerte festgelegt werden. Ebenso können viele sicherheitsrelevante Einstellungen getätigt werden.

Emergency Reverse (EMR)

Diese Funktion ist für die Erstellung eines Notfall-Rückwärtsgangs gedacht, welche bei einem Motorrad nicht gebraucht wird. Das Fahrzeug wird für eine bestimmte Zeit auf die festgelegte Geschwindigkeit im Rückwärtsbetrieb gebracht. Wenn sich das Motorrad währenddessen Forwärts bewegt, wird es zuerst abgebremst. Die Werte für die Verzögerung bzw. Beschleunigung können festgelegt werden, ebenso wie weitere Feineinstellungen.

Interlock Braking

Wenn die Funktion Interlock Braking aktiviert wurde, wird das Motorrad durch Rekuperation zum Stillstand gebracht, wenn der Eingang Interlock deaktiviert wurde. Wird Interlock Braking deaktiviert, so rollt das Fahrzeug im Leerlauf weiter. Die gewünschte Stärke der Verzögerung kann definiert werden, ebenso wie weiter sicherheitsrelevante Einstellungen.

Dual Drive

Diese Funktion dient dazu, ein Fahrzeug mit zwei Motoren betreiben zu können. Die zwei zugehörigen Motorcontroller interagieren über ein Master-Slave-System miteinander.

Vehicle

Über dieses Parametermenü können allgemeine Einstellungen, wie zum Beispiel die Einheit der Geschwindigkeit oder der Raddurchmesser festgelegt werden. Zusätzlich sind sehr viele unterschiedliche Parameter für eine genaue Auswertung und Ausgabe aktueller Fahrdaten vorhanden.

Supervision

Diese Parameter dienen der Einstellung der Geschwindigkeits- und Strom- bzw. Frequenzüberwachung. Wenn die zu kontrollierenden Werte ein definiertes Maximum übersteigen, wird ein Fehler ausgelöst.

3.2 Drehmomentsteuerung (Torquecontrol)

Die Drehmomentsteuerung wird mithilfe von zwölf unterschiedlichen Parametern beschrieben. Generell können diese Parameter eher als Feineinstellungen angesehen werden, da der grundsätzliche Vorgang immer der gleiche bleibt. Mit diesen Parametern werden die maximale Geschwindigkeit, die Reaktionszeit und die Aggressivität des Motors genau definiert, um das gewünschte Beschleunigungsmuster erhalten zu können.

3.2.1 Parameter

Grundsätzlich lassen sich diese Parameter in zwei Hauptgruppen unterscheiden:

- **Geschwindigkeitsbegrenzer-Parameter (Speed-Limiter)**

Bestimmen die maximale Geschwindigkeit und die Begrenzungsparameter des KPI-Reglers.

- **Reaktions-Parameter (Response)**

Bestimmen die verschiedensten Ansprechzeiten und Drehzahlen für bestimmte Ereignisse.

Diese Parameter werden in den folgenden Tabellen noch genau erklärt. Für ein besseres Verständnis können ebenfalls die englischen Tabellen „SPEED LIMITER MENU“ und „RESPONSE MENU“, aber auch die zugehörigen Abbildungen „Figure 9-11“ aus der Anleitung herangezogen werden.

Parameter	Einstellungsbereich	Beschreibung
Max Speed Max_Speed_TrqM	500 - 8000 U/min 500 - 8000	Definiert die maximale Geschwindigkeit in U/min, welche mittels der Drehmomentsteuerung angesteuert werden kann. (Unabhängig von der Gasdrehgriff-Stellung)
Kp Kp_TrqM	0 – 100 % 0 – 8192	Legt fest, wie aggressiv der Drehzahlregler versucht, die Motordrehzahl auf die maximale Drehzahl zu begrenzen. Größere Werte sorgen für eine genauere Kontrolle, können jedoch zu Regelschwingungen führen. Bei einem zu niedrigen Kp kann die maximale Geschwindigkeit den Parameter Max Speed überschreiten.
Ki Ki_TrqM	5 – 100 % 50 – 1000	Mit diesem Parameter kann die Integralregelung genauer beschrieben werden, welche die Drehzahlabgrenzung unterschiedlich stark beeinflusst, abhängig von der aktuellen Regelabweichung. Größere Werte ermöglichen eine schnellere Regelung, können jedoch zu Regelschwingungen führen. Bei zu niedrigen Werten kann es lange dauern, bis sich der Motor bei einer Überdrehzahl dem Parameter Max Speed nähert.
Kd Kd_TrqM	0 – 100 % 0 – 8192	Beschreibt die Dämpfung, wenn sich das Fahrzeug der Höchstgeschwindigkeit nähert, dadurch werden Überschwingungen verringert. Bei einem zu hohen Kd kann es lange dauern, bis die Höchstgeschwindigkeit erreicht wird. Wenn Kd zu niedrig eingestellt ist, kann die Höchstgeschwindigkeit überschritten werden, insbesondere beim Bergabfahren.

Tabelle VI.1: Speed-Limiter Parameter [[Manual]]

Parameter	Einstellungsbereich	Beschreibung
Accel Rate Accel_Rate_TrqM	0.1 - 30.0 s 100 - 30000	Legt fest, wie lange es dauert, bis das Motordrehmoment bei Vollgas auf das Maximum angestiegen ist. Größere Werte bedeuten eine langsamere Reaktion.
Accel Release Rate Accel_Release_Rate_TrqM	0.1 - 2.0 s 100 - 2000	Legt fest, wie schnell die Verzögerung des Fahrzeugs bei einem Loslassen des Gasdrehgriffs eingeleitet wird. Bei einem geringen Wert wird der Übergang abrupt eingeleitet. Ist der Wert zu hoch eingestellt, kann es sich anfühlen, wie wenn das Fahrzeug für kurze Zeit weiterfährt.
Brake Rate Brake_Rate_TrqM	0.1 - 5.0 s 100 - 5000	Legt fest, wie lange es dauert, bis das Bremsmoment bei einem startenden Bremsvorgang oder einem Fahrtrichtungswechsel aufgebaut wird. Größere Werte bedeuten einen schonenderen Bremsvorgang.
Brake Release Rate Brake_Release_Rate_TrqM	0.1 - 2.0 s 100 - 2000	Beschreibt, wie schnell sich das Bremsmoment löst, wenn das Fahrzeug vom Bremsvorgang zum Fahrbetrieb wechselt. Bei zu hohen Werten wird der Bremsvorgang noch kurzzeitig fortgeführt.
Neutral Braking Neutral_Braking_TrqM	0 – 100 % 0 - 32767	Der neutrale Bremsvorgang tritt auf, wenn der Gasdrehgriff losgelassen wird oder keine Fahrtrichtung gewählt wurde. Der neutrale Bremsstrom ist von 0 bis 100% des maximalen Rekuperationsstromes einstellbar.
Neutral Taper Speed Neutral_Taper_Speed_TrqM	200 - 6000 U/min 200 - 6000	Bei dieser Motordrehzahl wird der neutrale Bremsstrom rückgespeist, wenn der Gasdrehgriff losgelassen wird. Bei einem zu geringen Wert kann es zu Regelschwingungen kommen.
Forward Full Restraint Speed Forward_Full_Restraint_Speed_TrqM	100 – 32000 U/min 100 - 32000	Legt den Geschwindigkeitspunkt fest, an dem der volle Rekuperationsstrom rückgespeist wird, um das Vorwärtsrollen des Fahrzeugs zu verhindern. Kann auch als Parameter für die Rückhaltestärke angesehen werden. Bei einem zu geringen Wert kann es zu Regelschwingungen kommen.
Back Full Restraint Speed Back_Full_Restraint_Speed_TrqM	100 – 32000 U/min 100 – 32000	Legt den Geschwindigkeitspunkt fest, an dem der volle Rekuperationsstrom rückgespeist wird, um das Rückwärtsrollen des Fahrzeugs zu verhindern. Kann auch als Parameter für die Rückhaltestärke angesehen werden. Bei einem zu geringen Wert kann es zu Regelschwingungen kommen.

Tabelle VI.2: Response Parameter [[Manual]]

3.2.2 ECO- und Sportmodus (Speed-Mode-Select)

Die oben genannten Parameter werden für die Erstellung der unterschiedlichen Antriebsmodi verwendet. Für den ECO-Modus werden für die Parameter höhere Reaktionszeiten und weniger Aggressivität gewählt, daraus ergibt sich ein gemütlicherer und schonenderer Fahrbetrieb, welcher gleichzeitig weniger Leistung verbraucht. Für den Sport-Modus hingegen werden kürzere Reaktionszeiten und höhere Aggressivität gewählt, daraus ergibt sich ein sportlicheres Fahrverhalten, die gewünschte Geschwindigkeit wird schneller erreicht. Natürlich zieht dieser Antriebsmodus einen erhöhten Leistungsverbrauch nach sich. Die einzelnen Modi können beliebig konfiguriert werden und auch im Nachhinein mittels des Curtis Integrated Toolkit bzw. VCL-Studio sehr schnell und leicht verändert oder angepasst werden. Vorerst wird es nur zwei unterschiedliche Antriebsmodi geben, da in diesem Fall nur ein Pin für die Selektierung benötigt wird (ein digitaler Eingang). Wenn in Zukunft jedoch weiterhin Interesse an weiteren unterschiedlichen Antriebsmodi besteht, können diese mit relativ wenig Aufwand hinzugefügt werden.

Hier eine Tabelle der einzelnen Parameter im ECO- und im Sportmodus:

Parameter	ECO-Modus	SPORT-Modus
Max Speed	7000 U/min	8000 U/min
Kp	30 %	40 %
Ki	30 %	40 %
Kd	15 %	10 %
Accel Rate	2 s	1 s
Accel Release Rate	1 s	0.4 s
Brake Rate	2 s	1 s
Brake Release Rate	1 s	0.4 s
Neutral Braking	15 %	10 %
Neutral Taper Speed	500 U/min	800 U/min
Forward Full Restraint Speed	800 U/min	500 U/min
Back Full Restraint Speed	800 U/min	500 U/min

Tabelle VI.3: Antriebsmodi

Konventionelle Umsetzung

In konventionellen E-Motorrädern wird bei der Auswahl zwischen den Antriebsmodi ECO-Normal-Sport eine 7-Punkte-Map hinterlegt, welche die Drehzahl abhängig vom Drehmoment beschreibt. Da dies für die schulische Umsetzung aber ebenfalls zu zeitintensiv wäre, wird dies wahrscheinlich bei späteren Optimierungen hinzugefügt.

3.3 Vehicle-Control-Language (VCL) Programmierung

3.3.1 Grundfunktion

Die Vehicle-Control-Language ist ein eigener Abschnitt der Programmierung und von der Grundidee auch sehr unterschiedlich. Die normale Parameterprogrammierung vereinfacht die generelle Programmierung des Curtis Controllers zwar sehr, beschränkt die Anzahl der Möglichkeiten jedoch auf die Anzahl der Parameter und deren Konfigurationsmöglichkeiten. Die VCL-Programmiersprache bietet hierbei eine sehr gute Erweiterungsmöglichkeit und bei Kombination der beiden Programmierarten kann die Programmierung des Curtis Controllers optimiert und enorm erweitert werden. Die Vehicle-Control-Language ist eine Programmiersprache auf der Basis von C und das VCL-Studio sieht der Benutzeroberfläche eines normalen Programms mit integriertem Editor und Compiler, wie zum Beispiel Dev-C++, sehr ähnlich. Da in diesem Fall jedoch jeder beliebige Programmcode verfasst werden kann, bietet VCL nahezu endlose Möglichkeiten. Die wichtigsten Grundlagen, auf die man bei der Programmierung von VCL aufpassen muss, sind in den verlinkten Skripten „WIN VCL User’s Guide“⁵, „VCL Programmer’s Guide“⁶ und „VCL Common Functions“⁷ genauer erklärt. Da diese Skripten nicht über eine öffentliche Online-Ausgabe verfügen, findet man im Anhang ebenso eine kurze Zusammenfassung der Kapitel, welche für die Programmierung am häufigsten benötigt wurden [2.5]. Das Wichtigste ist jedoch die Kenntnis darüber, dass mit dem VCL-Studio nicht nur die vorhandenen oder selbsterstellten VCL-Variablen verändert werden können. Die meisten Variablen, die in den Parametermenüs vorhanden sind, können ebenfalls eingelesen und beschrieben werden. Hier ist natürlich Vorsicht geboten, denn viele Parameter dürfen in keinem Fall während des Betriebs geändert werden. Bei wichtigen Variablen muss daher auch ein Aus- und Einschalten des KSI-Pins durchgeführt werden, um die Veränderung wirksam zu machen und Komplikationen zu vermeiden. Generell ist die Hauptaufgabe einer ausgefeilten VCL-Programmierung jedoch die Automatisierung des gesamten E-Motorrades. Zum Beispiel kann beim Auftritt eines Fehlers automatisch ein zugehöriger Fehlerbehebungsvorgang eingeleitet werden. Da wir in unserem Projekt jedoch nur ein sehr kleines Zeitfenster zur Verfügung haben, konnte das VCL-Programm nicht derartig ausgereift werden. In unserem Fall beschränkt sich die VCL-Programmierung fast ausschließlich auf die Kommunikation mit dem Raspberry PI, denn die Lösung über VCL-CAN schien uns einfacher und vor allem viel flexibler als die vorgefertigte Programmierung „CANopen“ zu verwenden.

⁵WIN VCL User’s Guide: [\[WIN_VCL\]](#)

⁶VCL Programmer’s Guide: [\[VCL_Programmer\]](#)

⁷VCL Common Functions: [\[VCL_Common_Functions\]](#)

3.3.2 Kommunikation (CAN-Bus)

Die Kommunikation erfolgt grundsätzlich nur in eine Richtung, da dies nicht nur einfacher zu realisieren war, sondern für unsere Anwendungen auch völlig ausreicht. Der Curtis Controller sendet die ausgewählten Parameter immer in der selben Reihenfolge an den Raspberry PI, dafür wurden drei unterschiedliche Datenpakete mit jeweils einem einzigartigen Identifier und sieben folgenden Parametern erstellt. Ein Parameter enthält immer vier Hexadezimal-Zahlen, also insgesamt zwei Byte. Wenn ein Fehler am Motorcontroller erkannt wurde, wird nach dem dritten Datenpaket noch ein zusätzliches Fehler-Datenpaket eingeschoben. Wenn zum Beispiel zwei Fehler am Controller vorliegen, werden diese zwei Fehler nach dem Fehler-Identifier (FFFF) gesendet, danach werden die fünf weiteren freien Plätze des Fehler-Datenpakets mit dem Fehler-Identifier aufgefüllt. Nachdem das Fehler-Datenpaket gesendet wurde, werden die ersten sieben Fehler vom Datenspeicher gelöscht, liegt der Fehler jedoch weiterhin vor, kann dieser auch nicht gelöscht werden. Hier werden die Datenpakete mit den zugehörigen Parametern ausführlich dargestellt:

	Datenpaket 1	Datenpaket 2	Datenpaket 3
Identifier	FFFC	FFFD	FFFE
Parameter 1	Vehicle_Speed	Vehicle_Speed	Vehicle_Speed
Parameter 2	Current_RMS	Capacitor_Volts	Vehicle_Acceleration
Parameter 3	Controller_Temperature	BDI_Percentage	Vehicle_Odometer
Parameter 4	Motor_Temperature	Interlock	Time_to_Capture_Speed_1
Parameter 5	Motor_Power	Throttle_Command	Time_to_Capture_Distance_1
Parameter 6	Motor_Torque	Brake_Command	Braking_Distance_Captured
Parameter 7	Modulation_Depth	Switch_5	Distance_Since_Stop

Tabelle VI.4: Datenpakete Deklaration

Der gesamte VCL-Programmcode befindet sich im Anhang, auffällig ist hierbei unter anderem die enorm lange Zeitverzögerung, welche den selben Delay-Befehl 3330 mal beinhaltet. Die gewünschte CAN-Bus Übertragungsrate beträgt 0,1 Sekunden, da jedoch der integrierte Logic-Controller eine sehr viel kürzere applikationsabhängige Taktzeit besitzt, erwies sich das Aussenden von Parametern in der selben Reihenfolge vorerst als sehr schwierig. Es wurden viele unterschiedliche Methoden versucht, jedoch erwies sich die Synchronisation der beiden Taktfrequenzen als die einfachste. Da im Internet auch sehr wenig hilfreiche Informationen zu finden waren, wurde keine alternative Realisierung mehr versucht. Die Vereinfachung bzw. Verkürzung der langen Zeitverzögerung mit einer Schleife oder ähnlichem erwies sich ebenfalls als schwierig und fehleranfällig, wurde deshalb weggelassen⁸.

⁸VCL-Programmcode: [VCL_Programmcode]

3.3.3 Speed-Mode-Select

Die Speed-Mode-Select Programmierung gehört eigentlich auch zum Punkt Drehmomentsteuerung, dieses Programm wurde aber ebenfalls in der Vehicle-Control-Language realisiert. Die zugehörigen Parameter, welche ausschlaggebend für die Drehmomentsteuerung sind, wurden bereits genau erklärt. Grundsätzlich ist das VCL-Programm sehr einfach und kurz aufgebaut, die Parameter zur Drehmomentsteuerung werden abhängig von einem digitalen Eingang verändert. Der digitale Eingang wird mittels eines externen Schalters realisiert, dieser wurde auf der Lenkstange befestigt. Wenn der Schalter auf den ECO-Antriebsmodus geschaltet wird, werden die Parameter für den ECO-Antriebsmodus an den Drehmomentergler weitergegeben. Bei der Auswahl des Sport-Antriebsmodus werden die Parameter für den Sport-Antriebsmodus an den Regler weitergegeben. Der Wechsel des Antriebsmodus bedeutet jedoch eine Veränderung kritischer Parameter, weshalb der KSI-Pin aus- und eingeschaltet werden muss, um die Änderung der Parameter wirksam zu machen. Ein Ausschalten des KSI-Pins führt nach 0,1 Sekunden zum sofortigen Einleiten des Bremsvorgangs, bis die Geschwindigkeit Null ist. Bei richtiger Konfiguration kann der KSI-Pin in dieser Zeit aus- und eingeschalten werden, um einen Wechsel des Antriebsmodus während des Fahrbetriebs wirksam zu machen. Aus sicherheitstechnischen Gründen kann der KSI-Schaltvorgang jedoch nur erfolgen, wenn der Gas-Eingang Null ist.

Wie im folgenden Programmausschnitt zu erkennen ist, ist der Aufbau dieses Programms äußerst einfach. Aufpassen muss man bei der Überschreibung der Parameter, da nicht die Werte mit den Einheiten, sondern die absoluten Werte verwendet werden müssen. Vor allem bei den Parametern des PID-Reglers sind unterschiedliche Absolutwerte vorgegeben, diese müssen auf den gewünschten Prozentwert skaliert werden.

```
;Speed-Mode-Select (Sport/ECO)
if(Switch_5 = 0) {
    Max_Speed_TrqM = 7000 ;ECO-Antriebsmodus
    Kp_TrqM = 2458 ;7000rpm
    Ki_TrqM = 300 ;30%
    Kd_TrqM = 1229 ;15%
    Accel_Rate_TrqM = 2000 ;2s
    Accel_Release_Rate_TrqM = 1000 ;1s
    Brake_Rate_TrqM = 2000 ;2s
    Brake_Release_Rate_TrqM = 1000 ;1s
    Neutral_Braking_TrqM = 4915 ;15%
    Neutral_Taper_Speed_TrqM = 500 ;500rpm
    Forward_Full_Restraint_Speed_TrqM = 800 ;800rpm
    Back_Full_Restraint_Speed_TrqM = 800 ;800rpm
}
else if(Switch_5 = 1) ;Sport-Antriebsmodus
    Max_Speed_TrqM = 8000 ;8000rpm
    Kp_TrqM = 3277 ;40%
    Ki_TrqM = 400 ;40%
    Kd_TrqM = 819 ;10%
    Accel_Rate_TrqM = 1000 ;1s
    Accel_Release_Rate_TrqM = 400 ;0.4s
    Brake_Rate_TrqM = 1000 ;1s
    Brake_Release_Rate_TrqM = 400 ;0.4s
    Neutral_Braking_TrqM = 3276 ;10%
    Neutral_Taper_Speed_TrqM = 800 ;800rpm
    Forward_Full_Restraint_Speed_TrqM = 500 ;500rpm
    Back_Full_Restraint_Speed_TrqM = 500 ;500rpm
}
```

Abbildung VI.13: ECO/Sport-Select Programmierung[VCL_Programmcode]

4 Inbetriebnahme

4.1 Leonard-Versuchsaufbau

Grundidee:

Der Bau des Li-Ionen-Akkumulators hat sich leider sehr verzögert. Um effektiv an der Inbetriebnahme des Antriebssystems weiterarbeiten zu können, musste vorübergehend eine alternative Spannungsversorgung gefunden werden, welche für die Anforderungen der Motorsteuerung geeignet ist. Da der Motor und die Motorsteuerung jedoch eine bipolare Spannungsquelle (beide Energierichtungen) benötigen, um ordnungsgemäß in Betrieb genommen werden zu können, erwies sich dies vorerst schwieriger als gedacht. Die Beschaffung eines bipolaren Netzteils erwies sich als zu kosten- und zeitintensiv, weshalb der Betreuungslehrer vorschlug, die bipolare Spannungsquelle mittels eines Leonard-Umformers zu realisieren. Dabei treibt eine vom Netz gespeiste Synchronmaschine einen Gleichstrommotor an. Die Statorwicklung der Gleichstrommaschine kann hierbei für die Spannungsquelle verwendet werden, mit der Erregerspannung kann der gewünschte Spannungswert eingestellt werden.

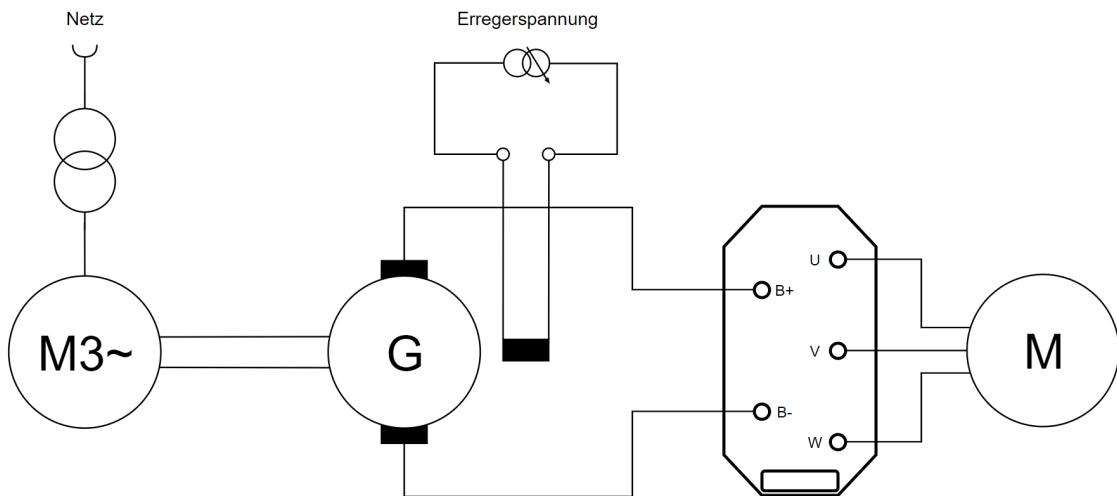


Abbildung VI.14: Leonardumformer Versuchsaufbau

Fazit:

Der erste Schritt bei der Inbetriebnahme des Motors ist der Testlauf (Commissionierung) zur Ausmessung und Einstellung der Motorparameter. Bei der Durchführung dieses Testlaufs stoppte die Motorsteuerung jedoch immer wieder nach kurzer Zeit. Auch viele weitere Versuche bei geänderten Testparametern oder zusätzlichen parallelgeschalteten Kondensatoren brachten keine weiteren Erkenntnisse. Aufgrund des schwankenden Spannungspiegels während der gescheiterten Test-Durchläufe nahmen wir genauere Messungen mittels eines Oszilloskops vor, um mögliche Fehlerursachen herausfinden zu können. Bei der Untersuchung der Gleichspannungs-Speisung konnten wir feststellen, dass in einem Zeitbereich von circa 40ms ein unerwarteter Einschwingvorgang zu beobachten war, welcher einer negativen Sinus-Schwingung sehr ähnelte. Es trat zuerst eine negative Flanke in einem Zeitbereich von 12ms und mit einer Spannungsunterhöhung von circa 30V auf, dann folgte eine Spannungsüberhöhung mit etwa den selben Werten. Nach weiteren 16ms war der Schwingungsvorgang wieder auf die Eingangsspannung von 50V zurückgefallen. Aufgrund dessen konnten wir rückschließen, dass unsere Leonard-Spannungsquelle zu träge für die Motorsteuerung ist. Außerdem entstand aus den hohen Induktivitäten des Motors und den langen Leitungen kombiniert mit den großen Kondensatoren der Motorsteuerung eine Art Schwingkreis, welcher den Trägheitseffekt zusätzlich verstärkte.

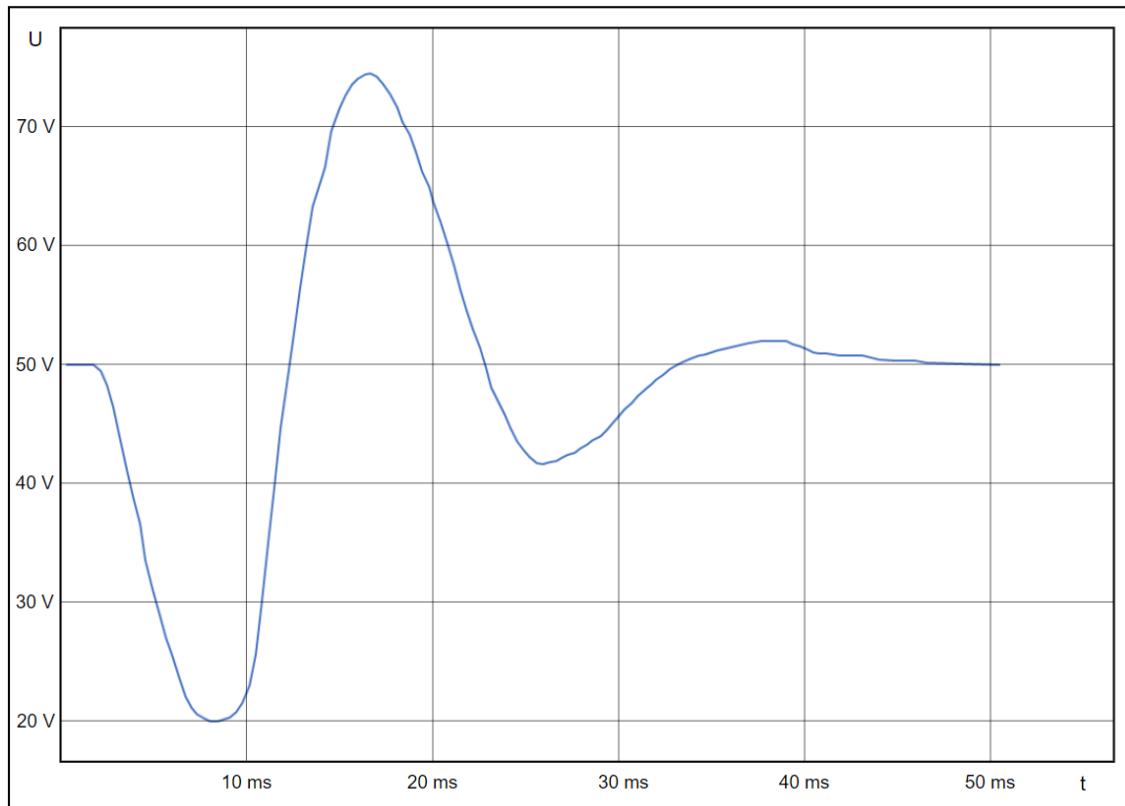


Abbildung VI.15: Leonardumformer Spannungsüberhöhung

4.2 Bleiakku-Versuchsaufbau

Grundidee:

Der Leonard-Versuchsaufbau hat aufgrund der Trägheit der Spannungsquelle nicht funktioniert. Übrig blieb deshalb nur die Realisierung der Spannungsquelle mithilfe eines Ersatz-Akkumulators. Zufälligerweise konnten wir im Projektraum vier Bleiakkumulatoren ausfindig machen, welche wir nun seriell zu einer 48V-Spannungsquelle verschalten wollten. Die Bleiakkus waren zwar teils angeschlagen bzw. sehr tief entladen, mittels eines intelligenten Ladegeräts konnten aber drei von vier Akkus wieder erfolgreich aufgeladen werden. Mit einer von Zuhause mitgebrachten Autobatterie und zusätzlichen Starterkabeln konnte letztendlich aber die 48V-Spannungsquelle realisiert werden.

Fazit:

Der Aufbau mit den Bleiakkumulatoren hat vorübergehend ganz gut funktioniert. Ein Problem stellten vorerst aber die großen Einschaltströme dar, welche bei der Schließung des Stromkreises zu Ladestromstößen führten. Um Beschädigungen an den Kondensatoren zu verhindern, konnte dieses Problem jedoch durch Vorladen der Kondensatoren mithilfe eines 48V Netzgerätes (durch Verwendung der Strombegrenzung) behoben werden.

Erste Inbetriebnahme:

Der Testlauf zur Einstellung der Motorparameter hat mithilfe der Bleiakkumulatoren beim ersten Versuch erfolgreich funktioniert. Der Motor konnte nach weiteren Konfigurationen letztendlich auch eine bestimmte Drehzahl abhängig von einem Spannungssignal anfahren.

Kapitel VII

Human-Computer Interaction System

1 Übersicht

Das Human-Computer Interaction System ist, wie der Name schon sagt, 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 HCIS¹ lassen sich grob in vier Grundfunktionen einteilen.

- **Steuerung der Peripherie**

Die Schalter und Buttons am Lenker, welche zuvor über den Kabelbaum die Leuchten, Blinker und die Hupe gesteuert haben, werden nun über die General-purpose input/output (GPIO) des Raspberry Pi Mikrocomputers 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 Anzeige 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 die Daten für Datenauswertung und Testberichte zu speichern. Ebenso bezieht das Diagnosesystem der Benutzeroberfläche die Daten über diese Schnittstelle.

¹ Abkürzung: Human-Computer Interaction System

1.2 Steuereinheit

Als Basis zur Auswahl der Steuereinheit wurden die zuvor erläuterten Grundfunktionen herangezogen genommen. Die Ausgewählte Steuereinheit sollte diese erfüllen können und ebenso Potential zur Erweiterung der Funktionen bieten. Genauso wichtig war das eine große Flexibilität und Individualität erreicht werden kann, um nicht in der Umsetzung unserer Ideen eingeschränkt zu sein. Zur Auswahl standen verschiedene Speicherprogrammierbare Steuerungen und Mikrocomputer, doch letzten Endes überzeugte der Mikrocomputer Raspberry Pi², welcher über die Interpretersprache Python³ programmiert werden kann und die Programmierung um einiges einfacher gestaltet.

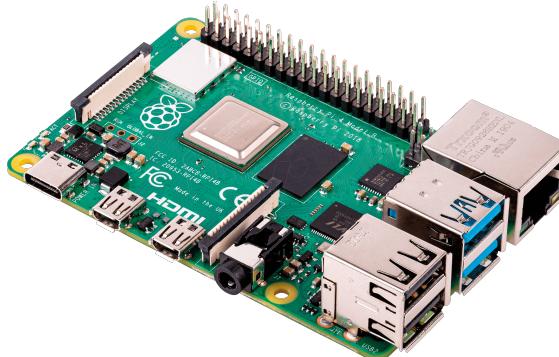


Abbildung VII.1: Raspberry Pi - Steuereinheit des HCIS

1.3 Grundaufbau des Systems

In der Abbildung VII.2 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 Motorrades.
- User Input - Die vorhandenen Schalter am Lenker des Motorrads werden direkt mit den Eingängen des Raspberry Pis verbunden.
- Peripherie - Die Grundkomponenten des Motorrades wie Scheinwerfer oder Hupe. Diese werden über Relais, welche an die Ausgänge des Raspberry Pis angeschlossen sind, gesteuert.
- Dashboard - Der Bildschirm zur Anzeige der verarbeiteten Informationen. Dieser wird über HDMI und USB mit dem Raspberry Pi verbunden.

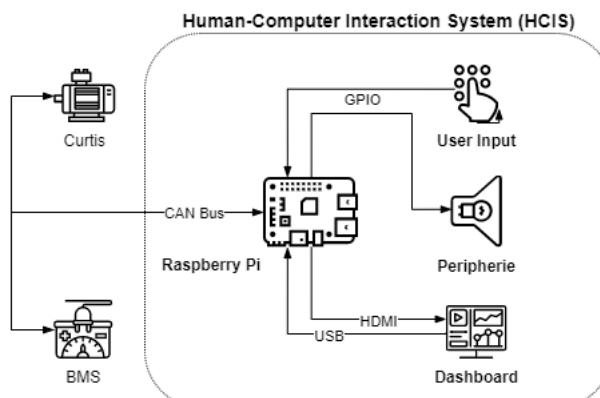


Abbildung VII.2: Grundaufbau des Human-Computer Interaction Systems

²siehe Anhang 1.3 oder <https://www.raspberrypi.org/>

³siehe <https://www.python.org/>

2 Spannungsversorgung

2.1 Aufbau des Versorgungssystems

Das Versorgungssystem des Motorrades besteht aus zwei Spannungsebenen: Einer 12V Ebene zur Versorgung der Peripherie des Motorrades und einer 5V Ebene, welche nur den Raspberry Pi und seine Komponenten beinhaltet. Diese Ebenen werden durch DC-DC Wandler erzeugt, welche direkt an den Akku des Motorrades angeschlossen werden.

Wichtig hierbei ist, dass alle ausgewählten Spannungswandler über einen Kurzschluss- und Überstrom-Schutz verfügen. Dies macht es uns möglich diese Versorgungssysteme, solange die Drähte auch nach dem maximalen Strom der Spannungswandler dimensioniert wurden, ohne jegliche Leistungs- und Überstrom-Schutzorgane aufzubauen. Die Spannungswandler schalten bei jeglichen Fehlern ab und verbrennen die überschüssige Leistung über einen eingebauten Widerstand. Sobald der Fehler behoben wurde, schalten sich die Spannungswandler automatisch wieder ein.

2.1.1 12V Versorgungssystem

Um den Spannungswandler dimensionieren zu können mussten vorher alle Bauteile, welche über die 12V versorgt werden sollten, zusammengefasst werden, um die mindestens benötigte Leistung des Spannungswandlers zu errechnen.

Bauteilbezeichnung	Spannung	Leistung
Tagfahrlicht	12V	10W
Abblendlicht	12V	10W
Aufblendlicht	12V	20W
Hupe	12V	10W
Rücklicht	12V	21W
Kennzeichenbeleuchtung	12V	5W
Blinker links	12V	2 x 10W
Blinker rechts	12V	2 x 10W
Bildschirm	12V	12W
Gesamt	12V	128W

Tabelle VII.1: Berechnung der Leistung des 12V-Systems

Der Spannungswandler wurde nun nach der größt möglichen Leistung, welche auftritt wenn alle Bauteile gleichzeitig auf Höchstleistung betrieben werden, ausgelegt. Diese maximale Leistung beträgt, wie in der Tabelle VII.1 zu sehen, 128 Watt. Um noch Ausbaumöglichkeiten zu gewährleisten und uns nicht dem Leistungslimit des Wandlers zu nähern, haben wir uns für einen 48V-12V, 300 Watt DC-DC Wandler von Mean Well⁴ entschieden.

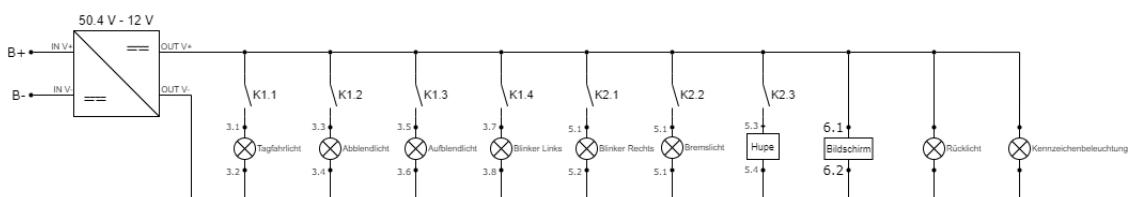


Abbildung VII.3: Aufbau des 12V Versorgungssystems

Diese Abbildung ist nur zur Veranschaulichung und kann im Anhang in original Größe gefunden werden.⁵

⁴Datenblatt: siehe Anhang 1.2

⁵siehe Anhang

2.1.2 5V Versorgungssystem

Die Leistung des Raspberry Pis ist mit einem Maximum von 6.2 Watt sehr klein und daher ist die Wahl des Spannungswandlers in diesem Fall nicht wirklich davon abhängig. Auch die Komponenten, welche angeschlossen werden, haben grundsätzlich keine erwähnenswerte Wirkleistung und müssen daher nicht genau berechnet werden. Nun entschied nur mehr das Preis-Leistungs-Verhältnis sowie die Ausfallsicherheit des Spannungswandlers die Wahl. Daher haben wir uns für einen 48V-5V, 30 Watt DC-DC Wandler von Meanwell⁶ entschieden.

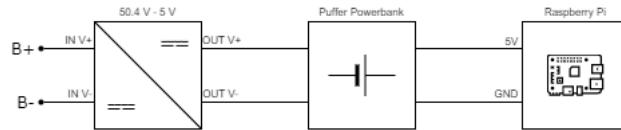


Abbildung VII.4: Aufbau des 5V Versorgungssystems

2.1.3 Abschalten der Spannungswandler

Das Abschalten der Spannungswandler ist nicht notwendig, da diese - wie schon in Abschnitt 2.1 erklärt - bei einem anliegenden Fehler automatisch abschalten. Ebenso wird beim Abschalten des Motorrades über die BMS jegliches andere Bauteil von der Spannungsversorgung getrennt. Was die Spannungswandler vom Entladen des Akkus abhält.

⁶Datenblatt: siehe Anhang 1.1

3 Steuerung der Peripherie

Die Grundfunktionen wie Beleuchtung, Hupe und Blinker werden hier als Peripherie bezeichnet. Diese sollten so einfach wie möglich und vom Lenker aus zu bedienen sein. Ebenso müssen sie verlässlich gesteuert werden können. Daher haben wir uns entschieden diese Funktionen ebenso über den Raspberry Pi zu steuern, da dieser bei einem Fehler der Motorsteuerung über den eingebauten Puffer gespeist werden kann und daher diese wichtigen Funktionen bis zu einem sicheren Stillstand weiter betrieben und gesteuert werden können.

Dennoch ist in der Plan in Zukunft die Motorsteuerung, welche ebenso in der Lage wäre die Ausgänge abhängig von den Eingängen zu schalten, diese Aufgabe übernehmen zu lassen, solange die Ausfallsicherheit ebenso gegeben wäre. Der Vorteil dieser Methode ist die Schaffung einer Zentralen Steuereinheit, welche alle Steueraufgaben in einem Bauteil vereinen kann.

3.1 Hardware

3.1.1 Input

Wenn wir eine digitale Schaltung wie die GPIO⁷ Pins betrachtet, sind die Pins immer entweder 0 oder 1. In diesem Fall muss der Zustand von 0 auf 1 oder von 1 auf 0 geändert werden. In beiden Fällen muss der digitale Pin entweder auf 0 gehalten und dann auf den Status auf 1 geändert werden oder andersherum. In beiden Fällen muss man den GPIO Pin entweder auf "Hochöder" Niedrigsetzen, er kann aber nicht schwebend gelassen werden. Dafür werden Pullup- oder Pulldown Widerstände verwendet. Standardmäßig werden hier 10kOhm Widerstände benutzt. Ob Pullup oder Pulldown ist grundsätzlich gleichgültig und hängt meistens von der Anwendung ab. Wir benutzen für das Einlesen der Eingänge 10k Pulldown Widerstände, um nicht immer eine Spannung an den Eingängen des Raspberry Pis anliegen zu haben.

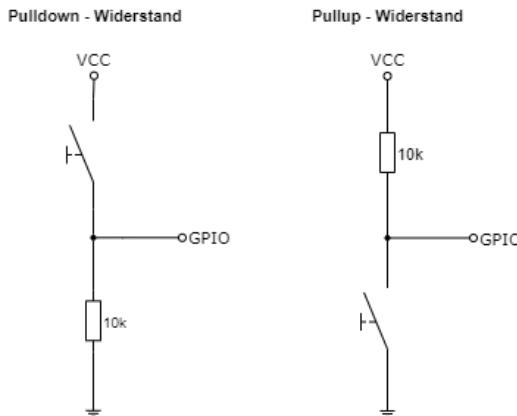


Abbildung VII.5: Anschlussplan Eingänge

Pullup Widerstand

Hier kann der digitale Logikeingangspin mit dem Schalter von logisch 1 auf logisch 0 umgeschaltet werden. Der Widerstand wirkt als Pull-up-Widerstand. Es ist mit der Logikspannung von der Versorgungsquelle von 3,3 Volt verbunden. Wenn der Schalter nicht gedrückt wird, hat der logische Eingangspin immer eine Standardspannung von 3,3 Volt oder der Pin ist immer hoch, bis der Schalter gedrückt und der Pin gegen Masse kurzgeschlossen wird, wodurch er logisch 0 wird.

Pulldown Widerstand

Bei einem Pulldown Widerstand passiert genau das Gegenteil. Der Widerstand ist mit Masse verbunden. Somit wird der digitale Logikpegel als Standard 0 gesetzt, bis der Schalter gedrückt wird und der Pegel logisch 1 wird. In diesem Fall fließt die geringe Strommenge mithilfe des geschlossenen Schalters und des Pulldown-Widerstands von den 3,3 Volt zur Erde, wodurch verhindert wird, dass der GPIO Pin mit der 3,3 Volt Quelle kurzgeschlossen wird. Weshalb wir uns auch für diese Variante entschieden haben

⁷general-purpose input/output

3.2 Umsetzung

Um diese Theorie auch an unser Projekt anwenden zu können, müssen zuerst die Taster und Schalter, welche sich normalerweise am Lenker des Motorrades befinden Ausgemessen werden. Dies wird gemacht um herauszufinden welche Drähte mit welchen Schalter verbunden ist. Das Gleiche gilt für die Frontbeleuchtung. Nur so können diese auch wieder richtig angeschlossen und verwendet werden.

3.2.1 Lenkerschalter

Die einzelnen Schalter und deren Schaltstellungen sowie die dazugehörigen Drähte können mit Durchgangsmessung herausgefunden werden. Dafür muss nur ein Schalter umgelegt werden und mit Trial and Error nach dem geschlossenen Kontakt an den Drähten gesucht werden.

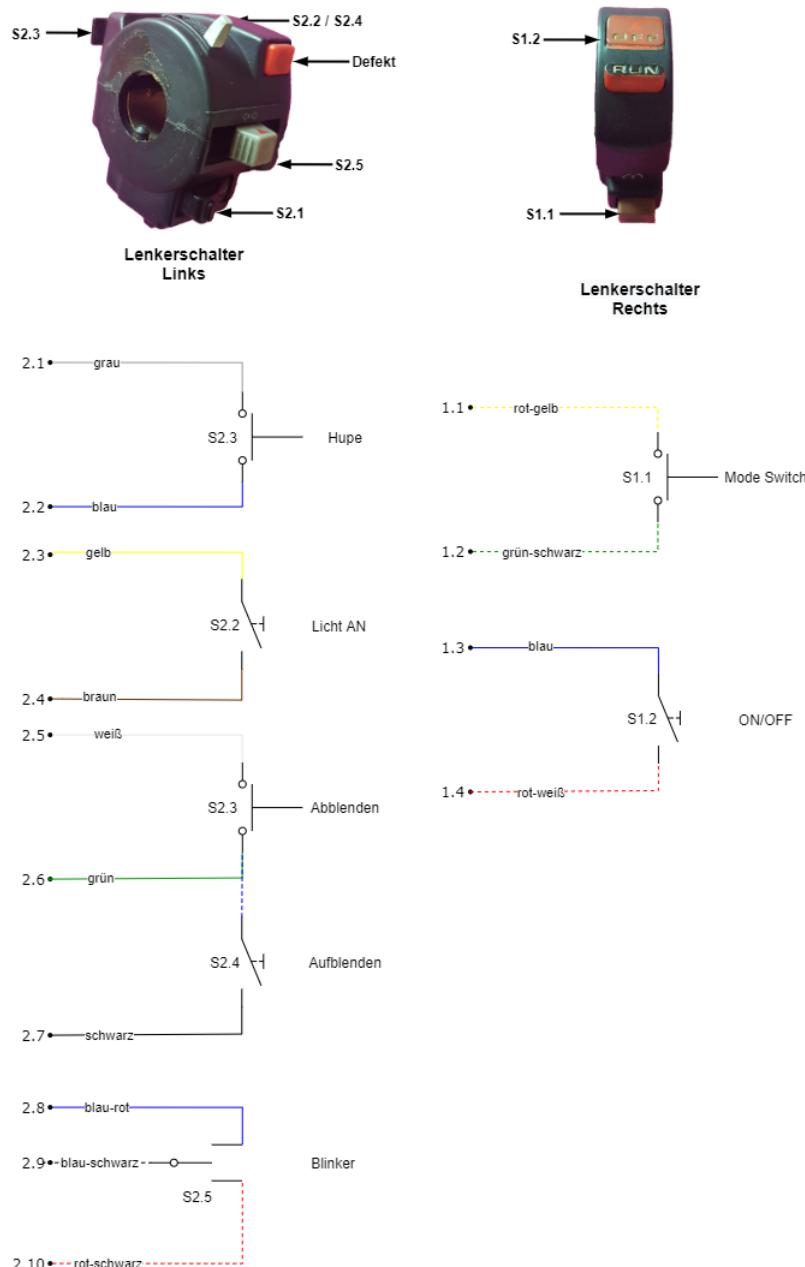


Abbildung VII.6: GUI Komponente - Modus Anzeige

3.2.2 Beleuchtung

Ähnlich konnten die Drähte den Richtigen Leuchten des Frontscheinwerfers zugeordnet werden.

3.2.3 Output

Hierbei werden die GPIOs als Ausgang verwendet. Sie sind verbunden mit den Eingängen eines 4 Channel Relais Moduls, welches über die 5V direkt von dem Raspberry Pi gespeist wird. Hiermit ist es nun möglich die 12V der Peripherie zu schalten und

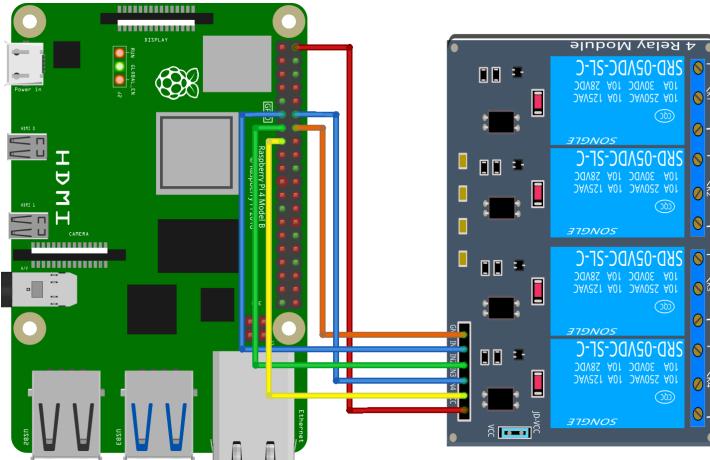


Abbildung VII.7: Anschlussplan Relais

3.3 Software

Wichtig bei der Programmierung war in diesem Fall die dauerhafte Verfügbarkeit der Grundfunktionen sowie die einfache Integrierbarkeit von alten und neuen Bauteilen. Dies wurde erreicht durch die Anwendung verschiedener Bibliotheken.

Die Wichtigsten in dieser Anwendung waren:

3.3.1 gpiozero

Eine Bibliothek, welche das einfache und schnelle Integrieren neuer Ein- oder Ausgänge ermöglicht. Dadurch können schnell und einfach Änderungen an dem Steuerverhalten der Peripherie gemacht werden. Hierzu kann die Klasse Button und DigitalOutputDevice verwendet werden. Dieser können Parameter wie die gewünschte Debounce Zeit und verwendete Pull-Widerstände übergeben werden. Es werden nur drei Zeilen Code benötigt um einen Aus- oder Eingang zu definieren und anzusteuern⁸

3.3.2 threading

Eine Bibliothek, welche es ermöglicht, einen eigenen Thread⁹ für das Programm zu öffnen, wodurch die Anwendung ohne Einflüsse oder Unterbrechungen anderer Programme weiter arbeiten kann. Der folgende Programmcode zeigt einen Ausschnitt der Klasse zum Steuern der GPIOs.

```
def start(self):
    self.thread=Thread(target=self.runner)
    self.thread.daemon=True
    self.thread.start()
```

Code Listing VII.1: Code zum Starten eines Threads

In diesem Beispiel wird ein Demon Thread erzeugt und gestartet. Das bedeutet dieser Thread wird beim schließen des Programms automatisch mit geschlossen und muss dadurch nicht mehr überwacht werden. Das Target ist die Funktion, welche unabhängig ausgeführt werden soll.

⁸siehe: Anhang

⁹gleichzeitig laufende Aufgabe

4 Benutzeroberfläche

Die Benutzeroberfläche stellt die Verbindung zwischen dem Nutzer und dem Motorrad dar. Sie sollte während der Fahrt die Instrumententafel des Motorrades ersetzen und dem Nutzer die wichtigsten Fahrinformationen anzeigen. Sobald das Motorrad zum Stillstand gekommen ist, wird es möglich Einstellungen zu ändern und die aufgezeichneten Fahrdaten anzeigen zu lassen. Ebenso können der Akkuladestatus und Informationen über Fehler im System entnommen werden.

4.1 Hardware

Zur Anzeige und Bedienung wird ein 11.6 Zoll kapazitives Touch LCD Display verwendet. Es besitzt eine Full HD Auflösung (1920x1080), was für eine professionelle Darstellung essentiell ist. Ebenso hat es ein schützendes ABS Gehäuse, welches trotz fehlender IP Zertifizierung das Abdichten ermöglicht. Die Versorgungsspannung beträgt 12V, was ident zu den anderen Komponenten am Motorrad ist und daher die Versorgung sehr vereinfacht, es kann also über den gleichen Spannungswandler versorgt werden.



Abbildung VII.8: Paneel Maße

Die Auflösung und die Größe des Paneels wirkt sich stark auf das Design der Benutzeroberfläche aus. Es muss die Größe der Icons und der anderen Designelemente so angepasst werden, dass sie einerseits gut ersichtlich und andererseits einfach über Berührung zu bedienen sind.

4.1.1 Befestigung

In das Gehäuse des Paneels sind M4 Verschraubungen in einem Raster von 75mm x 75mm integriert und kann daher einfach an Wänden oder Platten verschraubt werden. Um den Bildschirm nun in einer ähnlichen Position wie die Instrumententafel zu befestigen wurde eine 100mm x 210mm x 1.5mm Aluminium Platte - wie in der Abbildung zu sehen - gebogen und mit Löchern versehen. Um diese Halterung nun an dem Motorrad zu befestigen werden die Verschraubungen der alten Instrumententafel verwendet.¹⁰



Abbildung VII.9: Befestigung des Displays

4.2 Software

Bevor die Software für die Benutzeroberfläche verfasst wurde, mussten das Design, die Funktionen sowie die angezeigten Informationen geplant werden, um einen reibungslosen Arbeitsablauf beim Entwickeln des Frontends zu gewährleisten. Design Elemente wurden zuvor in Adobe Illustrator vorgefertigt. In den folgenden Seiten wird das Ergebnis dieses Prozesses erläutert.

4.2.1 Aufbau

Die nachfolgende Abbildung zeigt den grundsätzlichen Programmaufbau der Benutzeroberfläche. Die einzelnen Fenster werden als Tabelle mit ihren angezeigten Informationen dargestellt. Dies ist wichtig da jede dieser Informationen vom Backend an das Frontend gesendet werden müssen. Ebenso sind in den letzten Zeilen der Tabellen die QML-Elemente zur Navigation zwischen den einzelnen Fenstern niedergeschrieben. Diese müssen auch schon in der frühen Phase der Entwicklung der Benutzeroberfläche definiert werden.

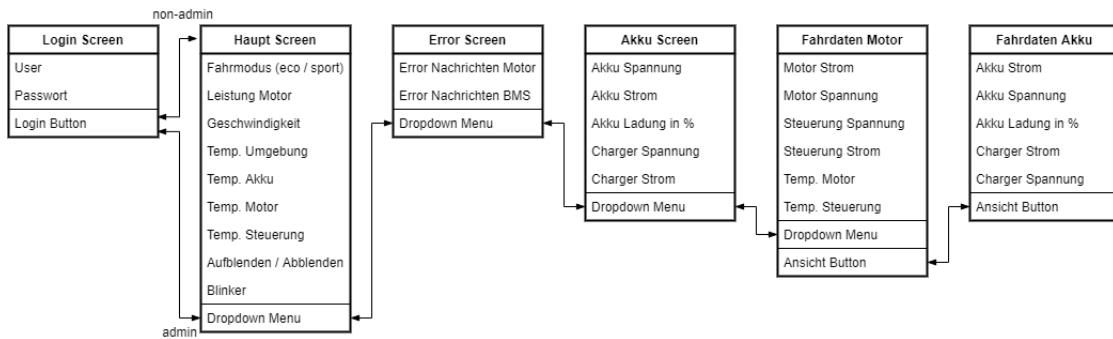


Abbildung VII.10: Aufbau der Graphischen Benutzeroberfläche

¹⁰siehe Anhang 3

4.2.2 Nutzer / Berechtigungen

4.3 Komponenten

Komponenten sind wiederverwendbare, gekapselte QML-Elemente mit genau definierten Schnittstellen. Komponenten werden häufig durch Komponenten-Dateien definiert, das heißt durch QML-Dateien. Wichtig ist dabei die Definition von Schnittstellen sowie Properties und Signals.

Properties

Einer Property eines Objektes kann ein statischer Wert zugewiesen werden, der konstant bleibt, bis ihm explizit ein neuer Wert zugewiesen wird. Um QML und seine integrierte Unterstützung für dynamisches Objektverhalten optimal zu nutzen, verwenden die meisten QML-Objekte jedoch Propertybindings.

Dies sind eine Kernfunktion von QML, mit der Beziehungen zwischen verschiedenen Objekteigenschaften festgelegt werden können. Wenn sich die Abhängigkeiten einer Property im Wert ändern, wird die Eigenschaft automatisch gemäß der angegebenen Beziehung aktualisiert. Hinter den Kulissen überwacht die QML-Engine die Abhängigkeiten der Eigenschaft. Wenn eine Änderung erkannt wird, wertet die QML-Engine den Bindungsausdruck erneut aus und wendet das neue Ergebnis auf die Eigenschaft an.

Java-Script-Funktionen

Programmlogik kann auch in Java-Script-Funktionen definiert werden. Diese können in QML-Dokumenten definiert und von Signalhandlern, Eigenschaftsbindungen oder Funktionen in anderen QML-Objekten aufgerufen werden. Solche Methoden werden häufig als Inline-Java-Script-Funktionen bezeichnet, da ihre Implementierung im QML-Dokument statt in einer externen Java-Script-Datei enthalten ist.

4.3.1 Navigationsmenü

Das Navigations-Menü ist ein Dropdown-Menü, welches zur Navigation zwischen den verschiedenen Fenstern benutzt wird. Sobald man sich eingeloggt hat wird das Menü angezeigt und die einzelnen Untermenüs können aufgerufen werden. Das Menü wird abhängig von den Berechtigungen des Benutzers angepasst.



Abbildung VII.11: GUI Komponente - Navigation Menü

Buttons

Die Navigation wird über das QML-Element *Mousearea*, welche direkt über den Icons der einzelnen Navigationselemente platziert wurde, gesteuert. Nun kann mit dem Befehl *onClicked* eine Funktion aufgerufen werden, welche das gewünschte Fenster sichtbar macht, sowie das Navigationsmenü wieder nach oben fahren lässt.

In dieser Funktion wird ebenso die Berechtigung des Nutzers über eine Globale Variable, welche beim Anmelden durch ein Signal gesetzt wird, abgefragt. Falls die Berechtigung die ausgewählte Funktion nicht zulässt wird ein Informationstext ausgegeben und das Menü wiederum geschlossen.

Abmelden

Wird der Abmeldebutton gedrückt, werden die Anmeldeinformationen zurückgesetzt und dem Nutzer wird wieder das Anmeldefenster angezeigt, wo er sich nun mit anderen Anmelde-Informationen einloggen kann.

4.3.2 Balken Anzeige

Die Komponente Balken Anzeige wird in der Benutzeroberfläche zur Visualisierung verschiedener Daten verwendet. Mit ihr können diese übersichtlicher dargestellt werden. Diese Komponente wird in mehreren QML-Dateien verwendet, daher sind Properties zur Anpassung notwendig. Die wichtigsten davon sind:

- Wert - Der aktuelle Wert, welcher am Balken angezeigt werden sollte.
- Anfangswinkel - Der Winkel an dem der Balken entspringt.
- maximaler Wert - Der maximal zu erreichende Wert. Dieser bestimmt die Länge des Hintergrundbalkens
- Hintergrundfarbe - Farbe des Hintergrundbalkens (in der Abbildung grau)
- Balkenfarbe - Farbe des Anzeigebalkens (in der Abbildung blau)

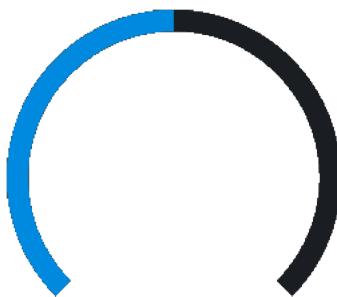


Abbildung VII.12: GUI Komponente - Balken Anzeige

Über ein *Signal* kann nun über das Backend der Wert des Balkens verändert werden und in Echtzeit angezeigt werden.

4.3.3 Modus Anzeige

Diese Komponente befindet sich auf der Instrumententafel. Der Modus kann über einen Button am Lenker, welcher mit der Curtis Steuerung verbunden ist, geändert werden. Sie dient zum Anzeigen der derzeitig gewählten Fahrmodi und wird über ein *Signal* aus dem Backend gesteuert, welches mit der Listener Klasse verbunden ist.



Abbildung VII.13: GUI Komponente - Modus Anzeige

Die Modus Anzeige ist eine einfache Komponente. Sie ist ein Item QML-Typ, dadurch wird das Nutzen von *States* möglich. Durch diese können verschiedene Eigenschaften gespeichert und über einen kurzen Befehl wiederhergestellt werden.

In diesem Fall wird der Punkt ausgefüllt und die Farbe des Textes geändert.

4.3.4 Graph

Die Graph Komponente befindet sich auf dem Diagnose Fenster und dient zum Anzeigen der in der Datenbank gespeicherten Fahrdaten. Sie verfügt über einen Ladebalken, welcher nach dem Auswählen der anzuseigenden Daten den Fortschritt des Auslesens der Datenbank anzeigt. Dieses Auslesen läuft über eine Funktion der Fetcher Klasse.

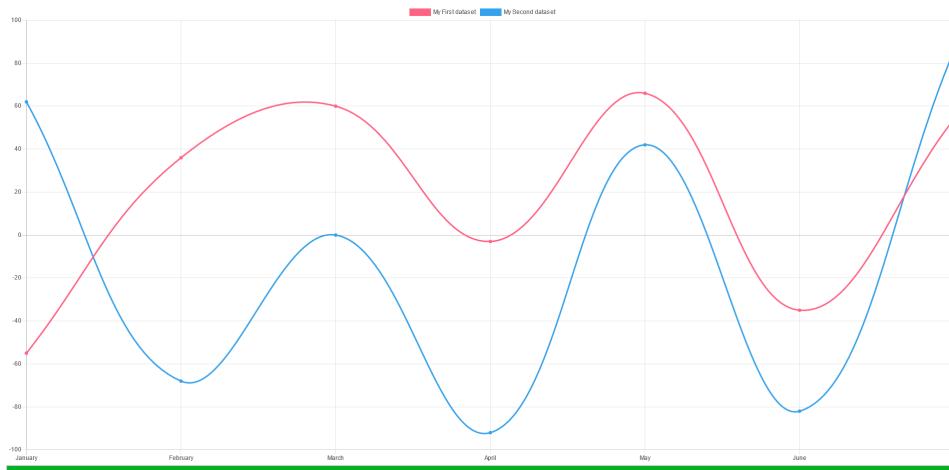


Abbildung VII.14: GUI Komponente - Graph

Realisiert wurde diese Komponente über öffentliche Bibliothek *Charts.js*¹¹, welche durch die Java Script Unterstützung von *QT*, mit der Installation eines Adapters¹² mit dem QML Programm kompatibel ist. Dadurch wird nun das Zeichnen von verschiedenen Graphen ermöglicht.

4.3.5 Weitere Komponenten

Wie schon bei der Fahr Modi Komponente, wurden weitere Objekte, welche zwischen zwei Zuständen hin und her gestalten werden müssen oder eine bestimmte Handlung ausführen sollten, über eine eigene QML Komponente verwirklicht. Diese ermöglicht nun wiederum das wiederholte verwenden dieser Komponente, sowie das Ansteuern über *States* und *Signale*. Diese Komponenten sind:

Umschaltkomponenten

Dies sind Komponenten, welche über ein *Signal* angesteuert werden und dann mithilfe von *States* ihr Aussehen verändern.

- Blinker Rechts
- Blinker Links
- Tagfahrlicht
- Fernlicht

Touch Komponenten

Dies sind Komponenten, welche eine *Mousearea* mit einer Graphik verbindet. Wird diese Komponente nun mit einem Mausklick ausgewählt, kann eine Funktion ausgeführt oder ein *Signal* ausgesendet werden.

- Abmeldebutton
- Navigationsmenübuttons
- Menü-Öffnen-Button

¹¹siehe <https://www.chartjs.org/>

¹²siehe <https://github.com/MichaelVoelkel/ChartJs2QML>

4.4 Programm Fenster

4.4.1 Login

Das Login Fenster dient zur Autorisierung des Benutzers. Über die User Kombobox kann der gewünschte Nutzer mit der dazugehörigen Berechtigung ausgewählt werden (siehe Abschnitt). Das Fenster dient ebenso zur Darstellung der Logos unserer Sponsoren



Abbildung VII.15: GUI Fenster - Login Menü

Um sich einzuloggen muss nur mehr das Passwort im *Textfeld* darunter eingegeben werden. Nach dem Drücken des Login Pfeiles (Neben dem Passwort Textfeld) werden die Login Daten an das Backend versendet. Diese vergleicht die Daten über die Datenbankschnittstelle¹³ und loggt, insofern das richtige Passwort gegeben ist, den Benutzer ein. Wird jedoch das Passwort falsch eingegeben, wird eine rote Fehlermeldung angezeigt.

4.4.2 Fahrdaten

Dieses Fenster dient als Ersatz für die Instrumententafel des Motorrades. Es ist daher das einzige während der Fahrt ersichtliche Fenster. Es Zeigt alle wichtigen Fahrdaten wie Leistung, Geschwindigkeit und Akkuladestand, sowie den Aktuellen Fahrmodus an. Diese Anzeige kann jedem Nutzer, unabhängig der Berechtigung, angezeigt werden.

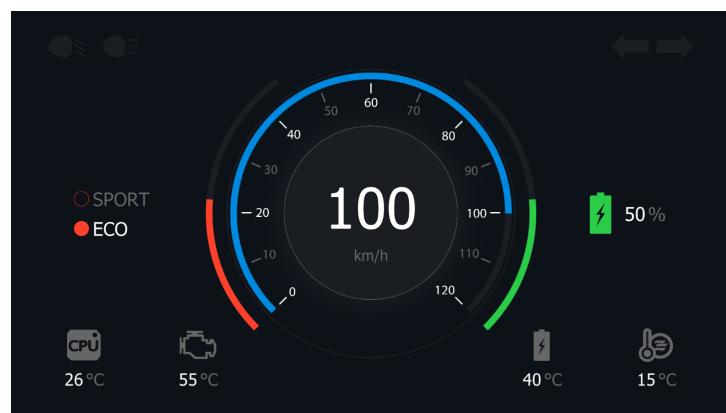


Abbildung VII.16: GUI Fenster - Fahrdaten

Der Status der Blinker und der Beleuchtung wird ebenso wie die Temperaturen der verschiedenen Komponenten des Motorrades angezeigt. Derzeit ist diese Ansicht starr und kann noch nicht geändert werden. Doch es ist geplant in Zukunft weitere Designs, welche in den Einstellungen geändert werden können, zu implementieren.

¹³siehe Abschnitt

4.4.3 Akku- und Ladedaten

Dieses Fenster dient Ähnlich wie das der Fahrdaten zur Darstellung wichtiger Daten. Es können alle wichtigen Daten bezüglich Akku und BMS auf einem Blick abgelesen werden.

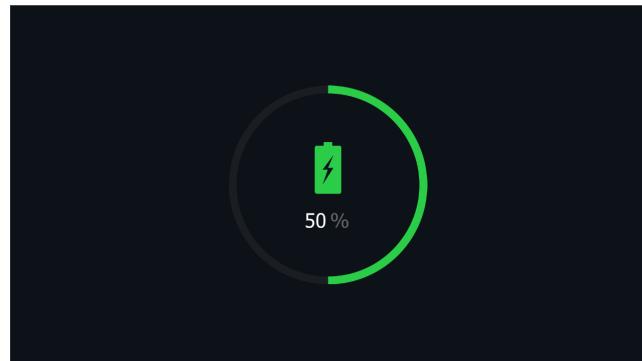


Abbildung VII.17: GUI Fenster - Akkudaten

Dieses Fenster wird Standardmäßig während dem Laden des Fahrzeuges angezeigt. Es benötigt also keine bestimmte Berechtigung und kann von jedem Nutzer über das Navigations-Menü angezeigt werden.

4.4.4 Fahrdaten Diagnose

Hier können die Fahrdaten, welche während der Fahrt dauerhaft von der Motorsteuerung versendet und vom Raspberry Pi verarbeitet und in einer Datenbank gespeichert werden, in Graphen angezeigt werden. Hierzu wird die *Graph Komponente*¹⁴ verwendet.

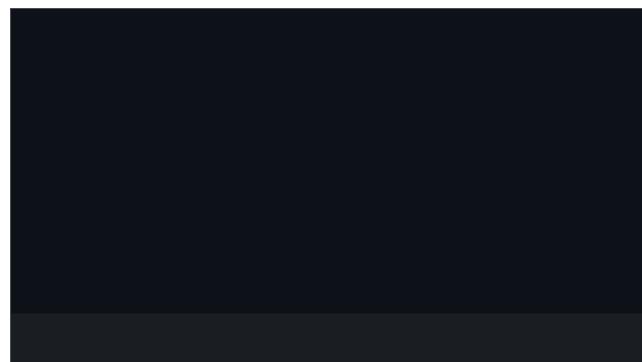


Abbildung VII.18: GUI Fenster - Fahrdaten Diagnose

Diese wird, um während der Fahrt die volle Prozessorleistung zu gewährleisten, erst auf Knopfdruck geladen. In der Menüleiste am unteren Ende des Fensters können verschiedene Vorlagen ausgewählt werden, welche über fixe Datensätze verfügen. Das bedeutet es werden erst beim Auswählen des anzuseigenden Datensatzes die Daten der letzten Stunde ausgelesen und in den Graphen geladen.

¹⁴siehe Abschnitt 4.3.4

4.4.5 Fehler

Sobald ein Fehlerdatenpaket über den CAN-Bus versendet wird, gibt der Listener Klasse die Fehlercodes an die Bridge Klasse weiter, um sie in diesem Fenster anzuzeigen. In einer *Listview* werden nun die Aktiven Fehler angezeigt. Solange Fehler anliegen werden diese angezeigt und erst sobald die zuvor angezeigten Fehler nicht mehr anliegen werden diese aus der Liste gelöscht.

FEHLER Nr.	NAME	BESCHREIBUNG

Abbildung VII.19: GUI Fenster - Fehler Liste

Der Name und die Beschreibung der Fehler, wird über eine Fehlertabelle¹⁵ in der Datenbank ausgelesen. Diese werden während dem Testen des Motorrades gesammelt und häufig auftretende Fehler werden in diese Tabelle eingetragen.

4.4.6 Nutzer und Berechtigungen

Um sicherzustellen das das Nutzen der Benutzeroberfläche nicht für jeden uneingeschränkt möglich ist wurde ein Anmeldesystem mit drei verschiedenen Benutzern, welche jeweils verschiedene Berechtigungen besitzen, umgesetzt.

ADMIN - Passwort: 53AC2

Dieser Benutzer verfügt über uneingeschränkte Rechte und kann auf jede verfügbare Funktion sowie Funktionen, welche noch in Entwicklung sind zugreifen.

USER - Passwort: 5AHET

Dieser Benutzer verfügt über wenig eingeschränkte Rechte. Er kann auf jede fertig entwickelte Funktion zugreifen kann jedoch keine Änderungen an der Benutzeroberfläche vornehmen und hat ebenso keinen Zugriff auf experimentelle Funktionen.

GUEST - Passwort: 00000

Dieser Benutzer verfügt nur über sehr eingeschränkte Rechte. Er wird zum Präsentieren der Maschine benutzt und lässt den unerfahrenen Nutzer nur auf die Fahrdaten und Akku- und Ladedaten zugreifen. Ebenso sollte in Zukunft die Leistung des Motorrades beim einloggen dieses Benutzers stark eingeschränkt werden, um Unfälle zu verhindern.

¹⁵siehe Abschnitt 6.1.3

4.5 Realisierung der Benutzeroberfächer

4.5.1 QML

QML¹⁶ 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 einfache Wiederverwendung und Anpassung von Komponenten innerhalb einer Benutzeroberfläche erlaubt. Es bietet Syntax mit Unterstützung für Java-Script-Ausdrücke in Kombination mit dynamischen Eigenschaftsverbindungen.

4.5.2 Qt-Quick

Das Qt-Quick-Modul¹⁷ 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 Objektinstanziieren. Es können problemlos flüssige, animierte Benutzeroberflächen in QML erstellt werden. Diese Benutzeroberflächen können mit beliebigen Backend Bibliotheken verbunden werden.

4.5.3 Slots und Signals

Slots und Signals werden in QML zur ereignisgesteuerten Kommunikation zwischen Frontend und Backend verwendet. In der folgenden Illustration wird diese anhand eines einfachen Beispiels erklärt.

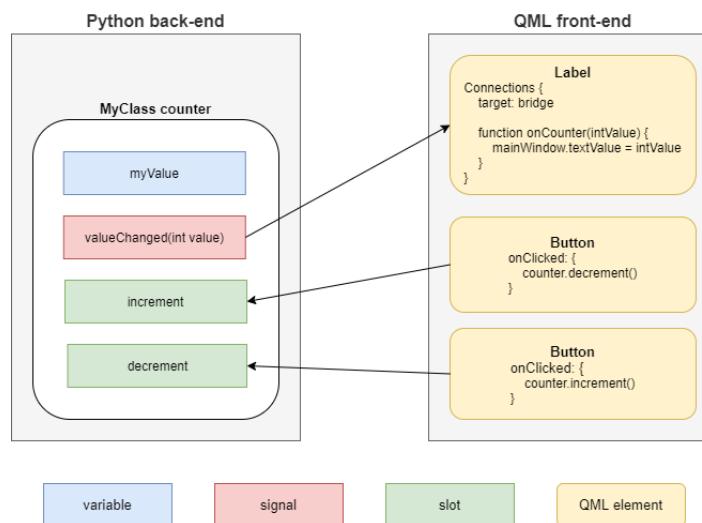


Abbildung VII.20: Slots und Signals Konzept

Signale

Diese können als Mitteilungen angesehen werden, welche über das Aufrufen der `signal.emit()` Funktion vom Backend an das Frontend gesendet wird. Im Frontend wird wiederum eine eigens definierte Funktion benötigt um dem Wert einem Property eines QML Elements zuzuweisen.

Slots

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

¹⁶Qt Modeling Language

¹⁷siehe <https://www.qt.io/>

4.5.4 Bridge

Die Bridge Klasse wird für die Kommunikation zwischen den einzelnen Modulen des Backends mit denen der graphischen Benutzeroberfläche. Die Kommunikation mit QtQuick findet mittels dem oben genannten Signal-Slot-Konzept statt.

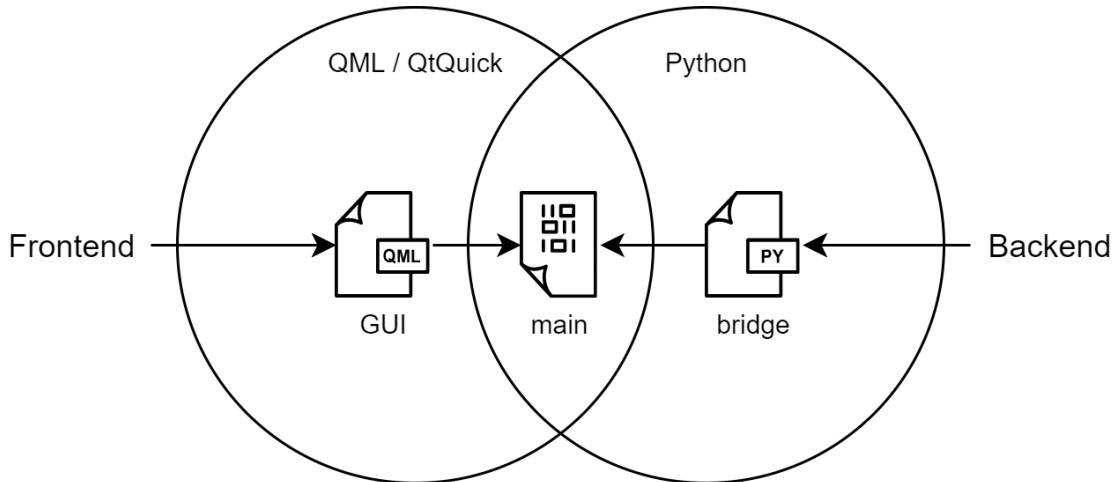


Abbildung VII.21: Verbindung Frontend zu Backend

Sie beinhaltet folgende Aufgaben:

- Kommunikation mit der Datenbankschnittstelle
- Verarbeitung der Übergebenen Daten in angepasste Modelle für QML
- Echtzeitverarbeitung einiger Daten von der CAN-Bus Schnittstelle

5 Kommunikation

Um Daten zwischen den mehreren Steuereinheiten des Motorrades zu versenden, muss eine Echtzeit-Kommunikation über ein Bussystem gewährleistet werden. Die Entscheidung ist auf das Controller Area Network Bussystem (CAN-Bus) gefallen. Ausschlaggebend für diese Entscheidung war der Curtis Motorcontroller, dieser verfügt über eine serielle Schnittstelle (RS-232) sowie ein CAN-System. Für unserer Anwendung bietet das CAN-System eine größere Ausbaufähigkeit sowie größere Übertragungsraten, weshalb wir uns letztendlich auch dafür entschieden haben.

5.1 Hardware

5.1.1 CAN-Modul

Da der Raspberry Pi selbst nicht über ein CAN-System verfügt, erfolgt der Anschluss an die Busleitung über ein externes CAN-Modul, welches über das Serial Peripheral Interface (SPI) mit dem Raspberry Pi kommuniziert, welches wie folgt angeschlossen werden muss:

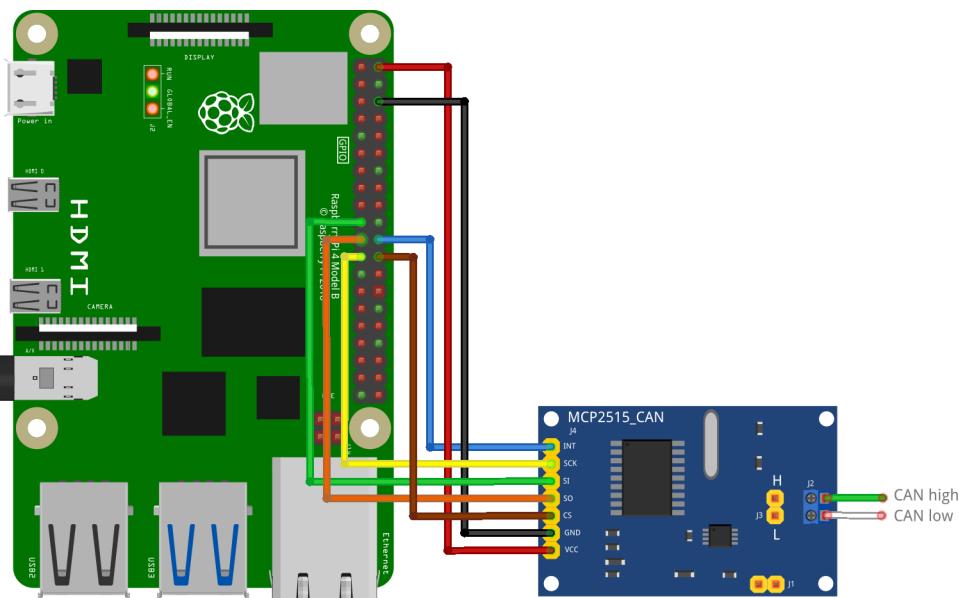


Abbildung VII.22: Anschlussplan CAN-Modul

Die Kommunikation wird über zwei Komponenten ermöglicht. Einen MCP2562 Transceiver, welcher für die Verarbeitung der Nachrichten zuständig ist und ein MCP2515 CAN Interface, welches die Daten zwischenspeichert und sich um das Versenden der Nachrichten kümmert. Gemeinsam mit dem Mikrocomputer ergibt dies nun einen CAN Node, welcher fähig ist Nachrichten zu versenden und zu empfangen.

5.1.2 Netzwerkstruktur

Ein CAN-Netzwerk wird standardmäßig als Bus- oder Sternkopplung aufgebaut. Wir haben uns bewusst für die Bustopologie entschieden, da Sternkopplungen nur in bestimmten Anwendungen Gebrauch finden und noch dazu markante Nachteile besitzt.

Es müsste zum Beispiel eine zentrale Steuereinheit den Nachrichtenverkehr steuern, ebenso gibt uns die niedrige Anzahl an Teilnehmern im Netzwerk nicht einmal die Möglichkeit ein anderes System zu verwenden. Erst wenn wir in Zukunft das CAN-Netzwerk um Sensoren und Aktoren erweitern würden, müsste weitere Zeit in die Planung des Netzwerks investiert werden.

5.2 Listener

Die Listener Klasse ist dafür zuständig den Datenverkehr am Bus zu überwachen und geordnet an die Datenbankschnittstelle¹⁸ sowie die Schnittstelle zum Frontend (Bridge)¹⁹ weiterzugeben. Nach dem Konfigurieren werden die einzelnen Nachrichten nach dem Empfangen des Identifiers in eine Liste geordnet und an die Bridge und den Handler weitergegeben, um weiter verarbeitet zu werden.

5.2.1 Konfigurieren der Schnittstelle

Für die Verbindung mit dem CAN-Modul wird die Bibliothek *pyhton-can*²⁰ verwendet. In unserem Fall wird diese Bibliothek mit einer bestimmten CAN-Schnittstelle verwendet. Diese kann im Code angegeben, aus Konfigurationsdateien oder Umgebungsvariablen gelesen werden.

Das *can-Objekt* macht ein *rc-Wörterbuch* verfügbar, mit dem die Schnittstelle und der Kanal vor dem Import aus *can.interfaces* festgelegt werden können.

```
import can
can.rc['interface'] = 'socketcan'
can.rc['channel'] = 'vcan0'
can.rc['bitrate'] = 500000
from can.interface import Bus

bus = Bus()
```

Code Listing VII.2: Konfigurieren des CAN Adapters vor Import

Es kann aber auch die Schnittstelle und der Kanal für jede Businstanz angegeben werden:

```
import can

bus = can.interface.Bus(bustype='socketcan', channel='vcan0', bitrate=500000)
```

Code Listing VII.3: Konfigurieren des CAN Adapters in Code

In der weiteren Entwicklung kann das CAN-Modul über eine Konfigurationsdatei instanziert werden.

5.2.2 Empfangen der Daten

Um die Daten, welche am Bus versendet werden auszulesen wird wiederum die *cantools* Bibliothek verwendet. Mit der *can.recv* Funktion kann der Listener gestartet werden. Das anschließende Beispiel ist ein einfacher Listener, welche die Nachrichten auf dem CAN-Bus auslesen kann.

```
listener = Listener()
meass = my_bus.recv()

listener.on_message_received(msg)

# Important to ensure all outputs are flushed
listener.stop()
```

Code Listing VII.4: Konfigurieren des CAN Adapters in Code

In der weiteren Entwicklung wird ein Buffer und weitere Erweiterungen umgesetzt.

¹⁸siehe Abschnitt 6.2

¹⁹siehe Abschnitt 4.5.4

²⁰siehe <https://python-can.readthedocs.io/en/master/index.html>

6 Fahrdatenspeicher

Das Motorrad sollte seine Fahrdaten abspeichern und darstellen können, um eine Diagnose des gesamten Aufbaus zu gewährleisten. Dafür wird eine Datenbank verwendet, nämlich die weit verbreitete Software MySQL verwendet.

6.1 Datenbankstruktur

6.1.1 Benutzer System

Diese Tabelle wird zum Einloggen in das Programm verwendet und ist mit keiner weiteren Tabelle verbunden. Der Inhalt dieser Tabelle²¹ sollte nicht verändert werden. Die Zuweisung der Rechte läuft über den Benutzernamen im Frontend. Die id ist nicht signiert und ist der Primärschlüssel mit Auto-Inkrement.

users		
Attribut	Datentyp	Beschreibung
id	int	Identifikationsnummer
username	varchar(50)	Name des Benutzers
password	varchar(255)	Eingegebenes Passwort

Tabelle VII.2: Datenbankstruktur der Benutzer Tabelle

6.1.2 Motor Daten

Fehler Datenpacket

Dieses Paket ist analog zu den Motordatenpaketen, welche in dem nächsten Abschnitt beschrieben werden. Es beinhaltet jedoch 7 ungeordnete Daten Attribute, welche die an der Motorsteuerung der anliegenden Fehler beschreibt.

Attribut	Datentyp	Beschreibung
id	int	Identifikationsnummer
date_time	datetime	Datum der Erstellung
error_1	int	Fehlercode 1
error_2	int	Fehlercode 2
error_3	int	Fehlercode 3
error_4	int	Fehlercode 4
error_5	int	Fehlercode 5
error_6	int	Fehlercode 6
error_7	int	Fehlercode 7

Tabelle VII.3: Datenbankstruktur der Fehler-Datenpaket Tabelle

²¹siehe Abschnitt 4.4.6

Motor Datenpakete

Das id Attribut ist nicht signiert und ist der Primärschlüssel, welcher mit der Auto-Inkrement Funktion versehen ist. date_time ist ein Zeitattribut und wird mit der Funktion NOW() beschrieben, um die Zeit des Beschreibens festzuhalten. Dies ist wichtig für das Auslesen der Daten beim Zeichnen eines Graphen in der Benutzeroberfläche. Der kaskadierte Fremdschlüssel id_data_X wird für die Verbindung der Tabellen mit der JOIN Funktion verwendet. Die Restlichen Attribute sind die geordneten Daten, welche über die CAN-Schnittstelle empfangen werden. Die vier verschiedenen Datenpakete - inklusive dem Fehlerdatenpaket - sind gleich aufgebaut. Nur die 7 Datenattribute sind an die Pakete des Curtis Controllers angepasst.

Datenpacket 1

Attribut	Datentyp	Beschreibung
id	int	Identifikationsnummer
date_time	datetime	Datum der Erstellung
vehicle_speed	int	Geschwindigkeit des Motorrades
current_rms	int	Ausgangstrom der Motorsteuerung
controller_temp	int	Temperatur des Kontrollers
motor_temp	int	Temperatur des Motors
motor_torque	int	Drehmoment am Motor
modulation_depth	int	Modulationsgrad des Kontrollers

Tabelle VII.4: Datenbankstruktur der Datenpaket 1 Tabelle

Datenpacket 2

Attribut	Datentyp	Beschreibung
id	int	Identifikationsnummer
id_data_1	int	Fremdschlüssel von Datenpaket 1
date_time	datetime	Datum der Erstellung
vehicle_speed	int	Geschwindigkeit des Motorrades
capacitator_volts	int	Eingangsspannung der Motorsteuerung
bdi_percentage	int	Ladestand des Akkus
interlock	int	Muss aktiv sein um zu fahren
throttle_command	int	Stellung des Gasgriffes
brake_command	int	Stellung der Motorbremse (software)
driving_mode	int	Aktueller Fahrmodus

Tabelle VII.5: Datenbankstruktur der Datenpaket 2 Tabelle

Datenpacket 3

Attribut	Datentyp	Beschreibung
id	int	Identifikationsnummer
id_data_2	int	Fremdschlüssel von Datenpaket 2
date_time	datetime	Datum der Erstellung
vehicle_speed	int	Geschwindigkeit des Motorrades
vehicle_acceleration	int	Eingangsspannung der Motorsteuerung
vehicle_odometer	int	Vom Motorrad zurückgelegte Strecke
time_to_capture_speed	int	Zeit bis zur eingestellten Geschwindigkeit
time_to_capture_distance	int	Zeit bis zur eingestellten Distanz
braking_distance_captured	int	Zeit von Bremsbeginn bis zu Stillstand
distance_since_stop	int	Zurückgelegte Distanz seit dem Stillstand

Tabelle VII.6: Datenbankstruktur der Datenpaket 3 Tabelle

6.1.3 Fehler Tabelle

Diese Tabelle ist ebenso in der Datenbank gespeichert. Sie beinhaltet die Fehlercodes der Motorsteuerung mit weiteren Informationen wie Name und Beschreibung. Sie muss per Hand angelegt werden und sollte während dem Testen der Maschine mit dokumentiert, um sie dann in das Fehler Fenster der Benutzeroberfläche zu implementieren.

Attribut	Datentyp	Beschreibung
error_code	hex	Errorcode von Steuerung in HEX format
error_name	varchar(50)	Kurze Benennung des Fehlers
error_description	varchar(255)	Kurze Beschreibung des Fehlers

6.1.4 Akku Daten

Da die Kommunikation der BMS noch nicht umgesetzt wurde, werden von Ihr keine Daten auf den CAN-Bus gelegt und daher muss auch keine Tabelle dafür erstellt werden. Somit werden alle Daten über Akku und Ladestand über die Motorsteuerung empfangen.

6.2 Handler

Die Handler Klasse hat die Aufgabe eine Verbindung zwischen der Datenbank und dem Programm herzustellen. Dazu verwendet sie die MySQL-Connector Bibliothek. Mit dieser können SQL Skripten direkt in die Python Klasse integriert werden ohne dass externe SQL Skript Dateien geöffnet werden müssen.

Dieser bekommt in Echtzeit eine Liste der aktuell Empfangenen Daten der Listener Klasse ²² und schreibt diese in die dafür vorgesehenen Tabelle der Datenbank. Hierfür wird ein eigener Thread ²³ geöffnet, um diese Aufgabe unabhängig von dem Restlichen Programm ausführen zu können.

Ebenso verfügt er über Funktionen, welche über das Diagnose Fenster ²⁴ angewählt werden können und Daten aus der Datenbank auslesen, um sie in einem Graph anzeigen zu lassen.

6.2.1 Konfigurieren der Schnittstelle

Um auf eine Datenbank zu zugreifen wird muss ein *connect* Objekt, mit den Anmeldedaten der lokalen Datenbank als Parameter, erstellt werden. Mit diesem Objekt kann nun auf Subklassen zugegriffen werden und Daten können aus- und eingelesen werden.

```
# Database credentials
dbhost = "hostname"
dbuser = "username"
dbpass = "password"
dbname = "databasename"

# Connection to the database
con = mysql.connector.connect(host=dbhost, user=dbuser, password=dbpass,
                               database=dbname)

# Creating cursor for database queries
cursor = con.cursor()
```

Code Listing VII.5: Konfiguration der Datenbankschnittstelle

²²siehe Abschnitt 5.2

²³siehe Abschnitt 3.3.2

²⁴siehe Abschnitt 4.4.4

6.2.2 Cursor

Die Cursor Klasse instanziert Objekte, die Operationen wie SQL-Anweisungen ausführen können. Cursor interagieren mit dem Server mithilfe eines Connection Objekts. Ein paar der Wichtigsten Befehle sind:

- *cursor.execute()*
Diese Methode führt die angegebene Abfrage aus. Die in den Tupel- oder Wörterbuchparametern gefundenen Parameter sind an die Variablen in der Operation gebunden. Geben Sie Variablen mit dem Parameterstil von % s oder % (Name) an.
- *cursor.fetchall()*
Diese Methode ruft alle Zeilen einer Abfrage ab und gibt eine Liste von Tupeln²⁵ zurück. Wenn keine Zeilen mehr verfügbar sind, wird eine leere Liste zurückgegeben.
- *cursor.close()*
Diese Methode wird verwendet wenn ein Cursor nicht mehr benötigt wird. Diese Methode schließt den Cursor, setzt alle Ergebnisse zurück und stellt sicher, dass das Cursorobjekt keinen Verweis auf sein ursprüngliches Verbindungsobjekt hat.

6.2.3 SELECT Befehl

Der SELECT Befehl wird zum Abrufen von Daten einer Tabelle verwendet und besitzt mehrere Parameter.

```
# Querying the data from the database
sql = "SELECT * from users WHERE username=%s and password=%s"
cursor.execute(sql, [(username), (password)])
results = cursor.fetchall()
```

Code Listing VII.6: SELECT Befehl über MySQL Connector

Der Befehl kann in zwei Bereiche unterteilt werden. In der ersten Zeile werden die Attribute, die von der Tabelle ausgegeben werden sollen, definiert. Die anzuzeigenden Ergebnisse werden durch den WHERE Befehl gefiltert werden.

Nützlich ist auch der Befehl JOIN durch den Tabellen anhand der Fremdschlüssel verbunden werden.

6.2.4 INSERT Befehl

Der INSERT Befehl wird zum Schreiben von Daten in eine Tabelle verwendet.

```
# Inserting data into the database
sql = "INSERT INTO `users`(`id`, `username`, `password`) VALUES(1,%s,%s);"
cursor.execute(sql, [(username), (password)])
```

Code Listing VII.7: INSERT Befehl über MySQL Connector

Dieser Befehl kann in zwei Teile unterteilt werden. In der ersten Zeile wird die Zieltabelle mit den Attributen angegeben. In der zweiten Zeile folgen die Werte, die in der Datenbank erfasst werden sollen. Hierbei ist die Reihenfolge der Attribute zu beachten. Das Attribut id wird standardmäßig von dem Datenbanksystem vergeben, es ist daher möglich das Attribut im Befehl zu vernachlässigen.

Als Wert wird ebenso die Funktion *NOW()* unterstützt, wodurch intern der aktuelle Zeitstempel abgespeichert wird. Dadurch kümmert sich das Datenbanksystem um die korrekten Werte für das date_time Attribut²⁶.

²⁵ein Datensatz (eine Zeile) einer Datenbank

²⁶siehe Abschnitt

Kapitel VIII

Endergebnis

Zum Abgabezeitpunkt befindet sich das Motorrad noch im selben Zustand, wie nach der Ausschlachtung. Alle Konzepte für die Fertigung und Fertigstellung des Motorrades sind fertig geplant und werden zu diesem Zeitpunkt auch schon Durchgeführt.

Die Zentralsteuerung ist mittlerweile schon einsatzbereit und kann sobald der Prototyp fertig ist eingesetzt werden. Die Benutzeroberfläche ist am selben Stand und konnte schon simuliert werden.

Ebenfalls schon fertig ist das Antriebskonzept. Motor und Motorsteuerung kommunizieren und der Motor ist von der Motorsteuerung steuerbar und Einsatzbereit, welches ebenfalls schon simuliert werden konnte. Sobald das Getriebe fertiggestellt wurde, kann der Motor auch schon montiert werden.

Das Akku und Ladekonzept ist ebenfalls auf einem guten Weg und mit dem Bau der Akkuboxen kann nach Fertigstellung des Getriebes begonnen werden.

Der Prototypen wird aufgrund vieler Probleme mit Sponsorensuche und späten Zusagen, beziehungsweise sehr oft Absagen, um 1 bis 2 Monaten später fahrbereit sein.

Wie unter der Unterüberschrift 4 Akkukühlung zu lesen ist, sind wie hier auch bei anderen Themen Zeit und Geld sparende Methoden angewendet worden um innerhalb eines Schuljahres fertig zu werden. Nach Schulende am 30. April wird der Prototyp fertiggestellt und in der Studienzeit noch erweitert werden, um möglicherweise die Straßenzulassung zu erhalten. Mit der Straßenzulassung sollen bestmögliche Optimierungen vorgenommen werden m auch eine Kühlung der Steuerung und damit einen sicheren Betrieb versichern zu können. Erweiterungen stehen immer offen und werden auch vorgenommen werden.

Diese Projekt ist deshalb sehr interessant, weil die Mobilbranche in diese Richtung expandieren und Techniker für diese Entwicklung benötigen wird. Neben möglicher Interessen von Firmen an uns, soll diese Projekt auch ein Forschungsobjekt für private Interessen sein, die in der Wirtschaft möglicherweise einmal von Bedeutung sein können. Diese Projekt ist ein weiterer kleiner Schritt in die Zukunft. Mit einer möglichen Verbindung des Projektes der Wasserstoffzelle eines Kollegen, kann eine Art der Fortbewegung erschaffen werde, die diesen Planeten schützen kann.

Anhang A

Allgemeines

1 Zeitplan

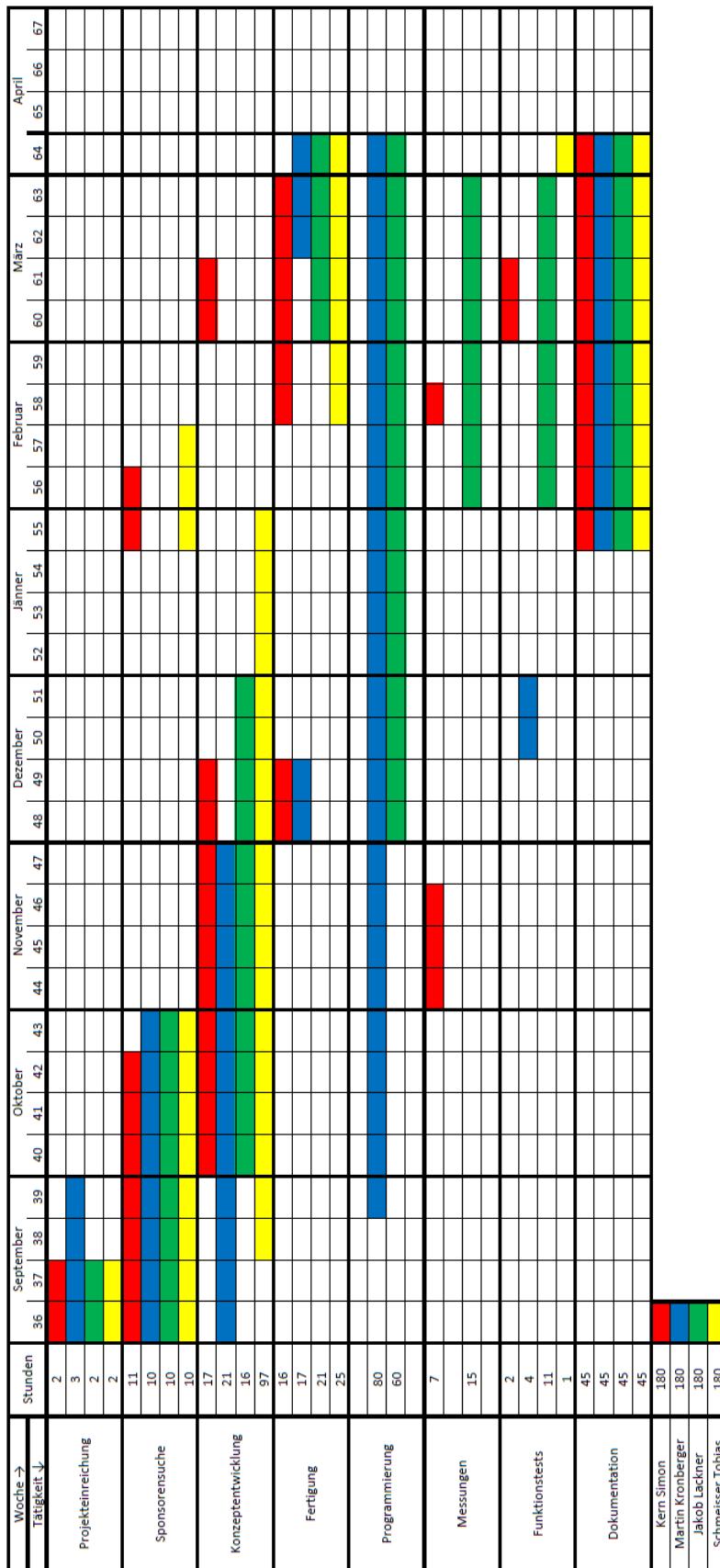


Abbildung A.1: Zeitplan und Arbeitszeiten

2 Kosten

Abbildung A 2: Kostenaufstellung und Stückliste

Anhang B

Programmcode

1 Antriebsstrang

2 HICS

2.1 Benutzeroberfläche

Anhang C

CAD-Zeichnungen

1 Mechanik

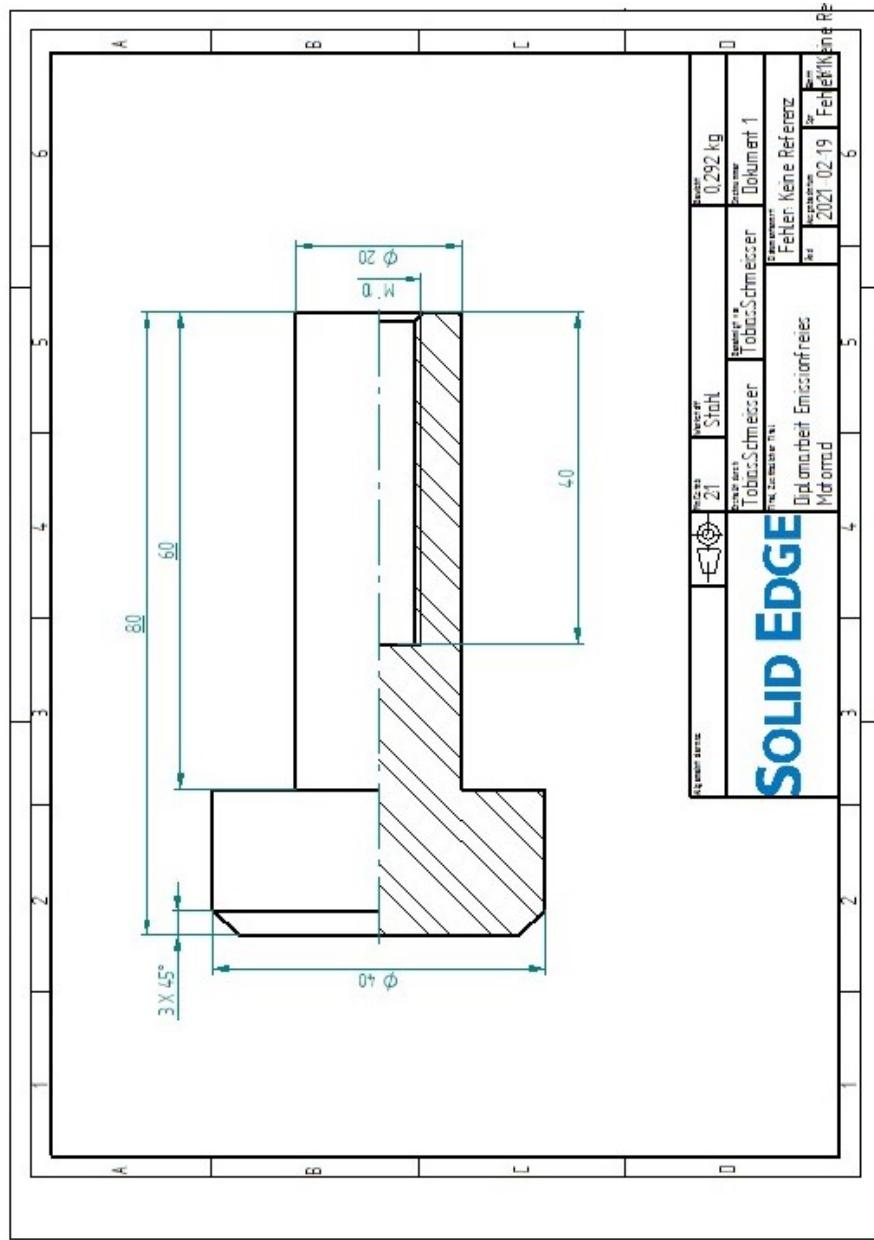


Abbildung C.1: Wellenersatz

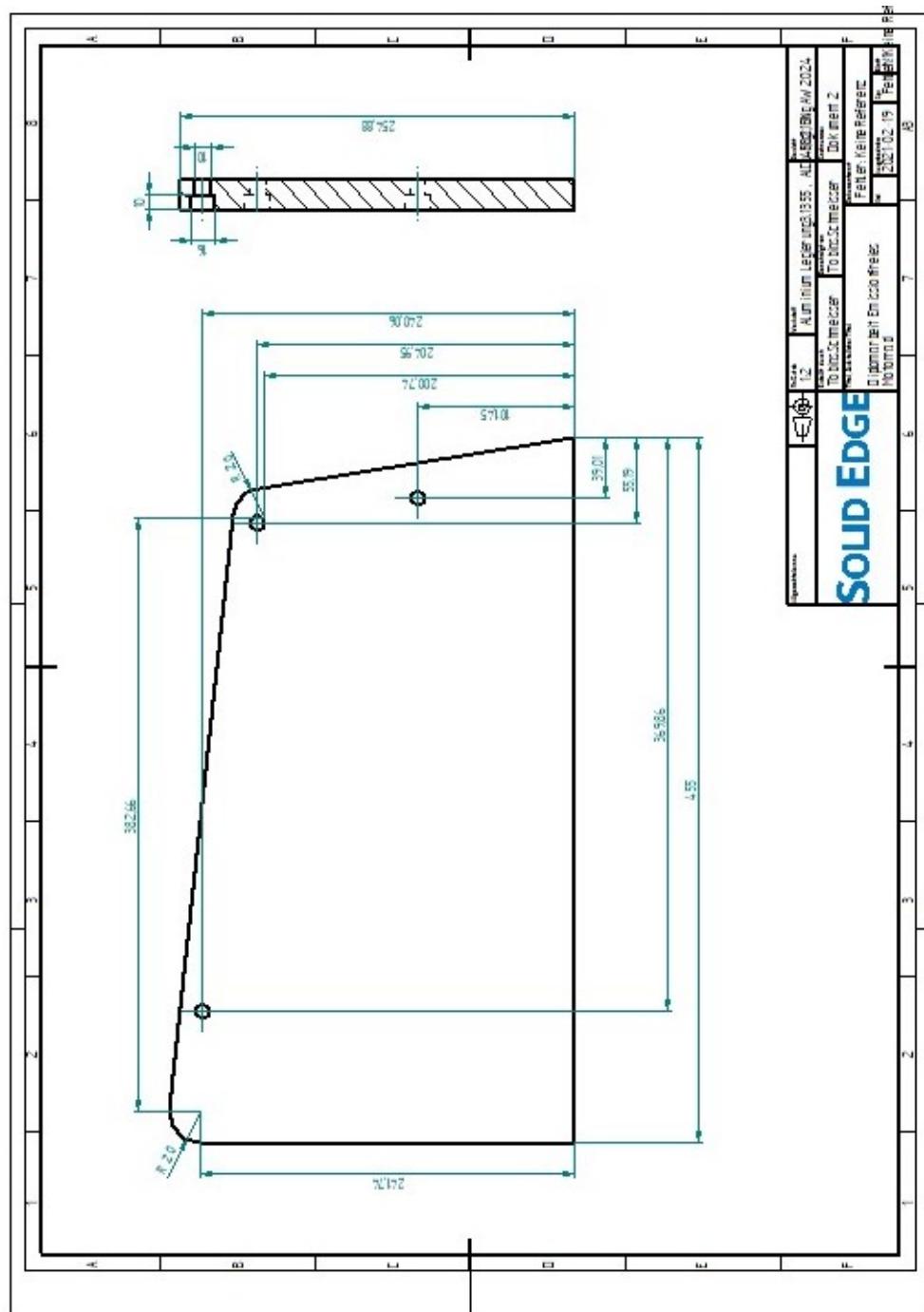


Abbildung C.2: Seitenplatte Rechts

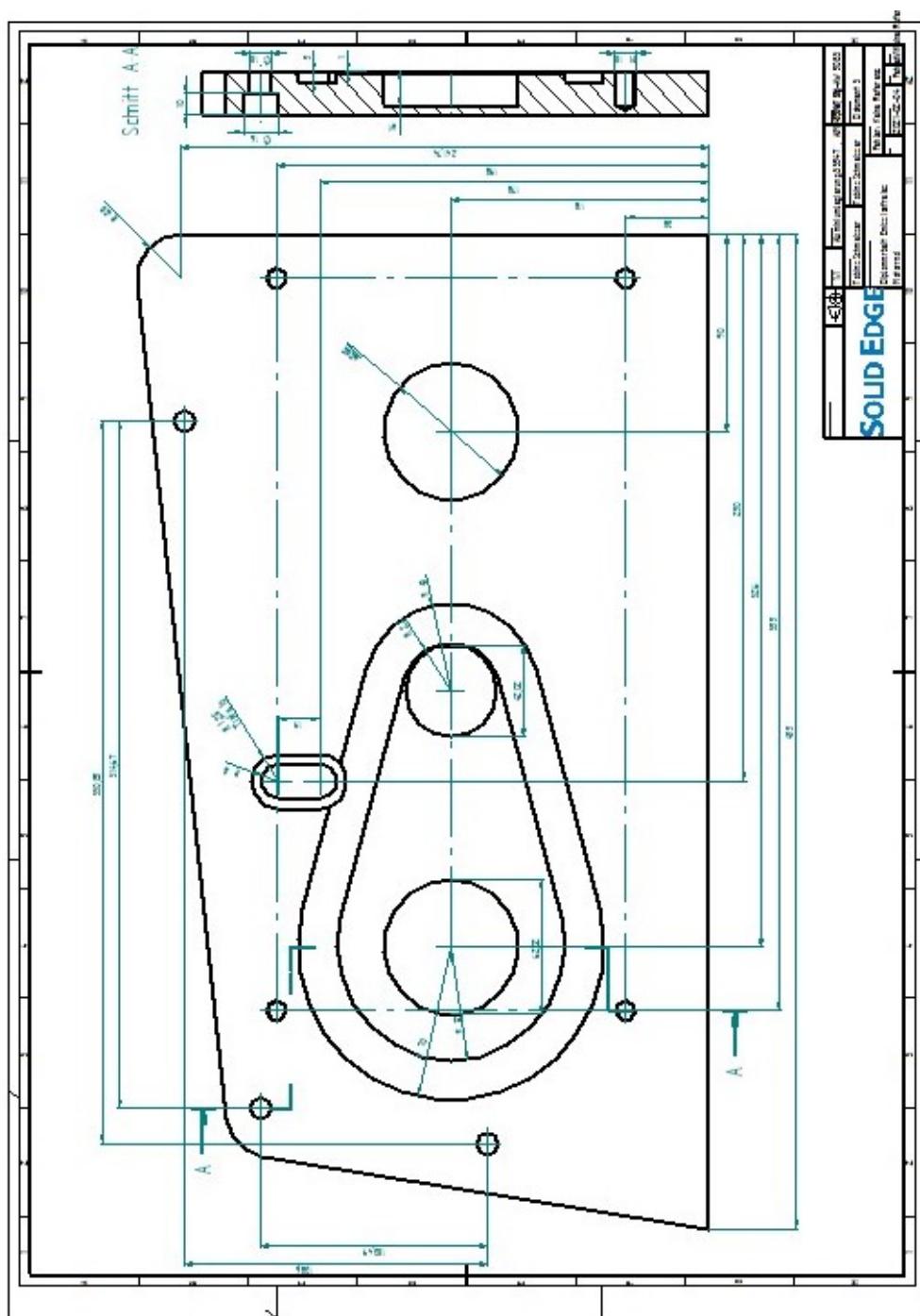


Abbildung C.3: Seitenplatte Links

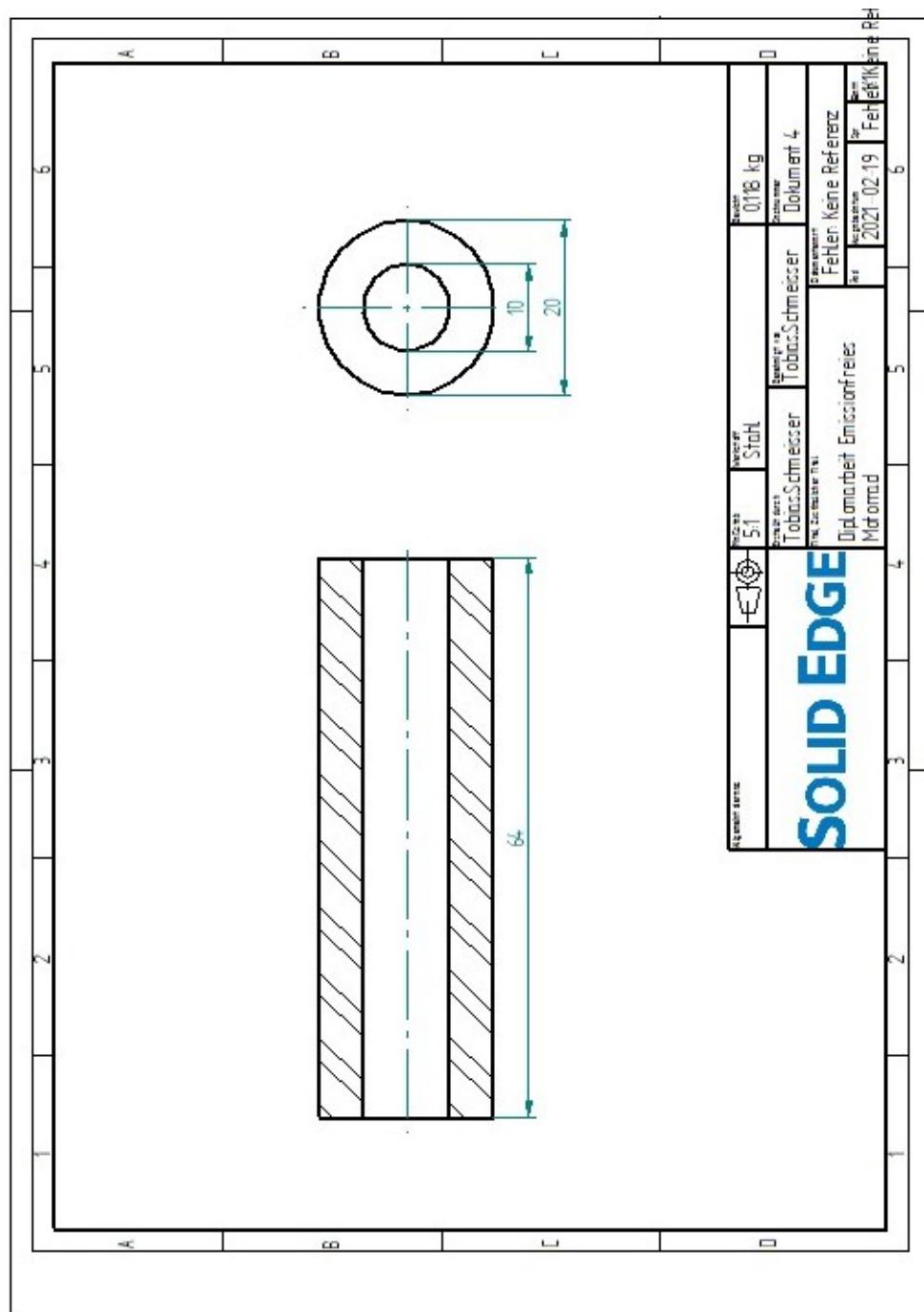


Abbildung C.4: Abstandhalter

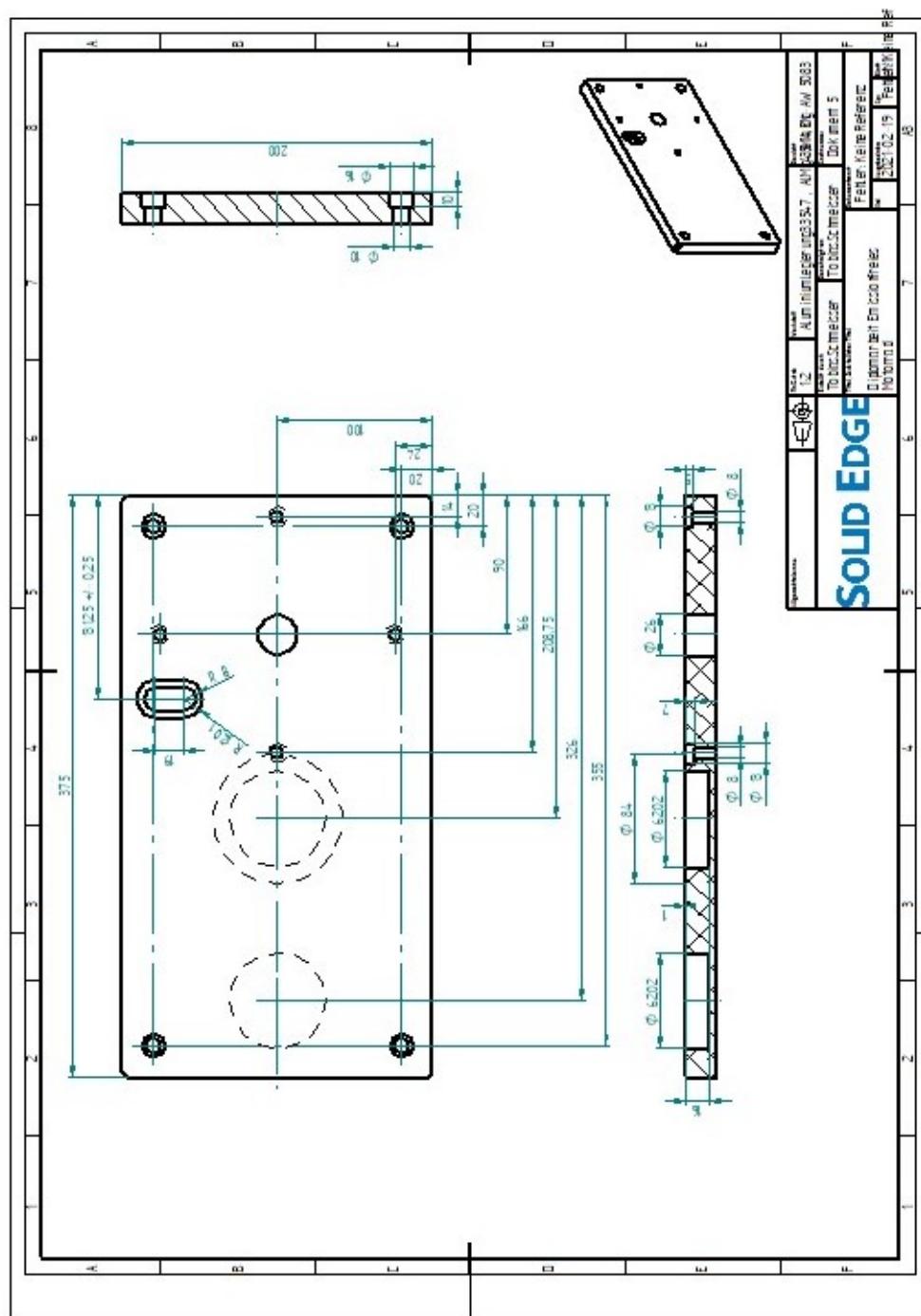


Abbildung C.5: Aufbau/Zusatzplatte

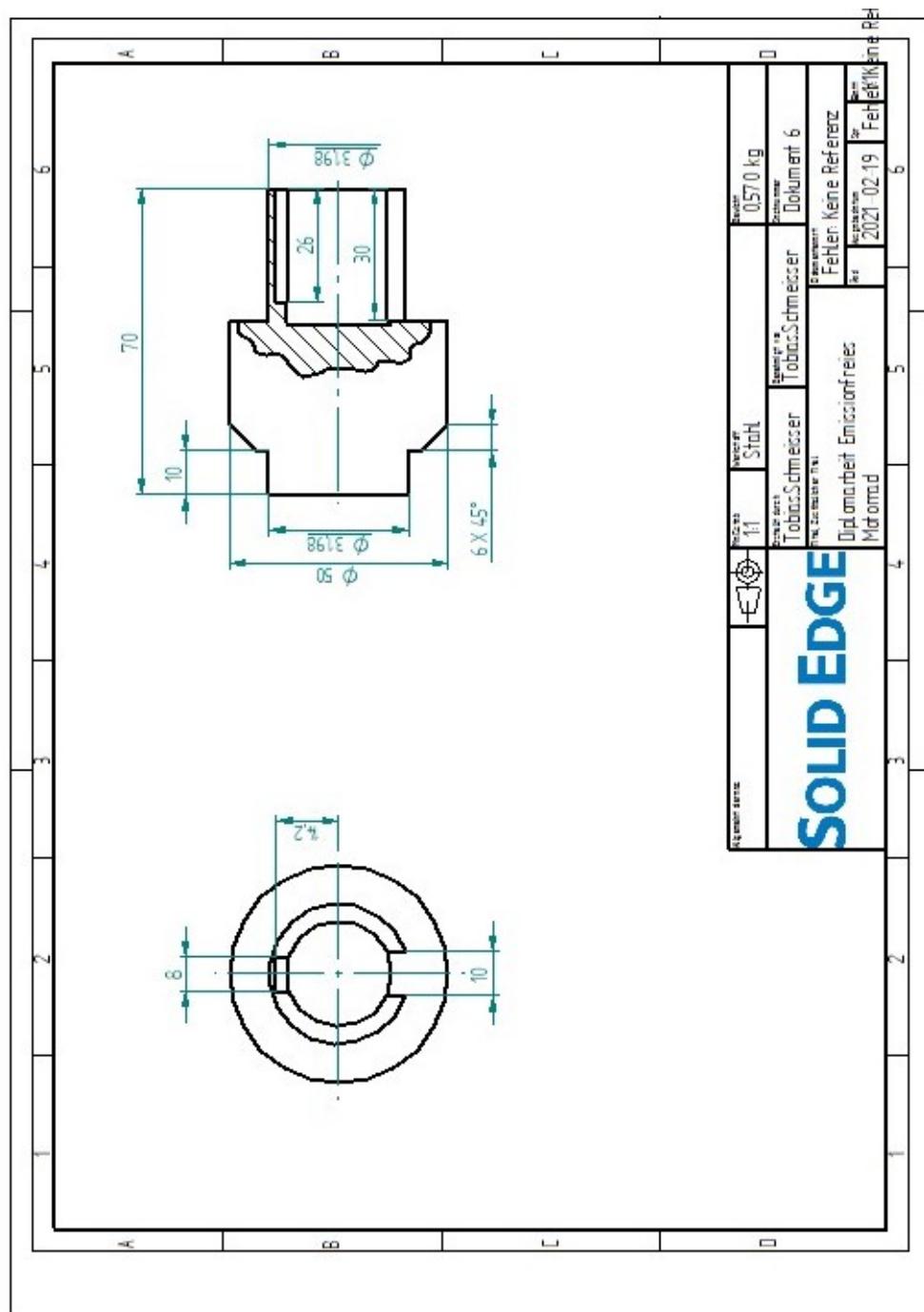


Abbildung C.6: Achse 1/Antriebsachsen

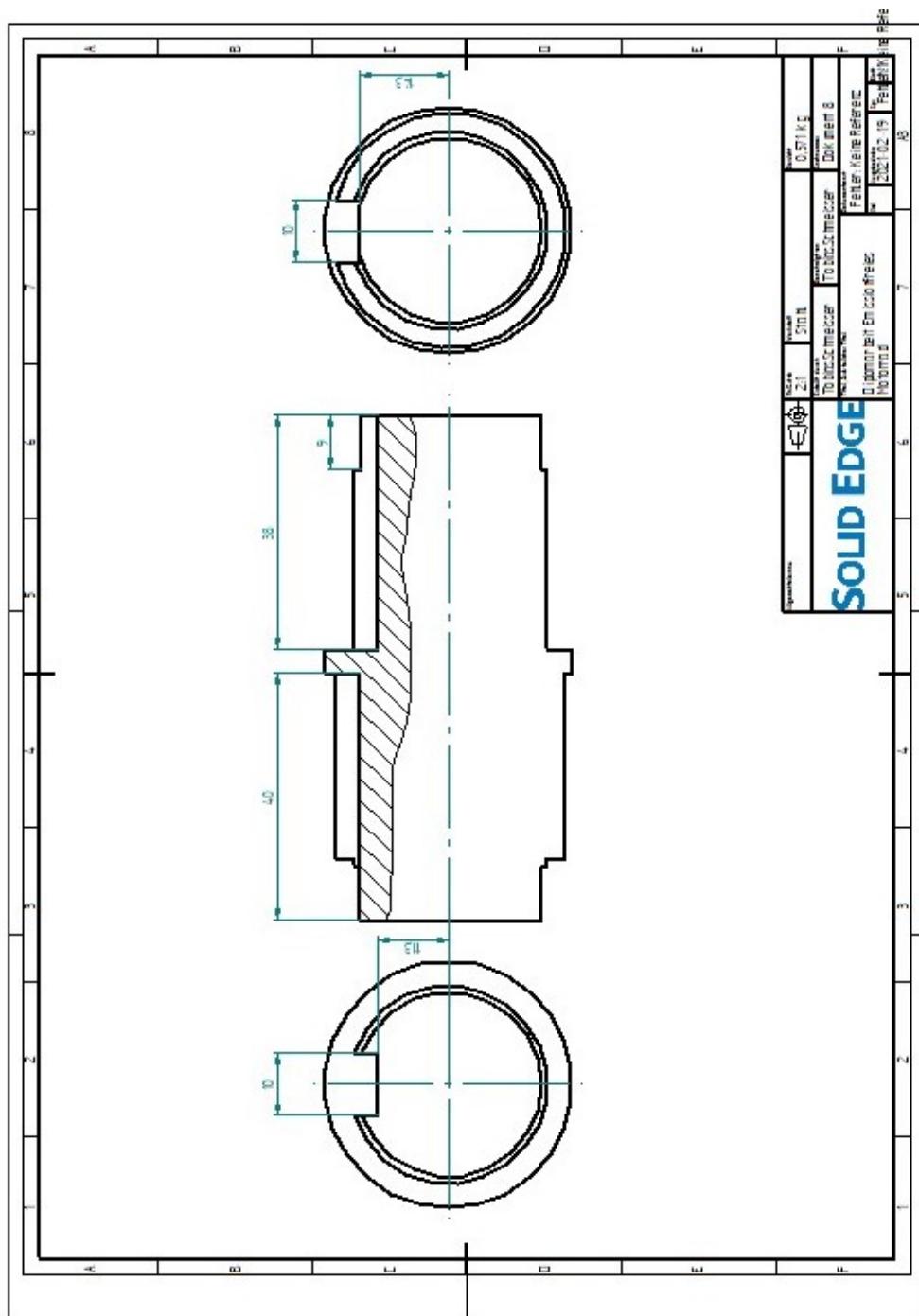


Abbildung C.7: Achse 3

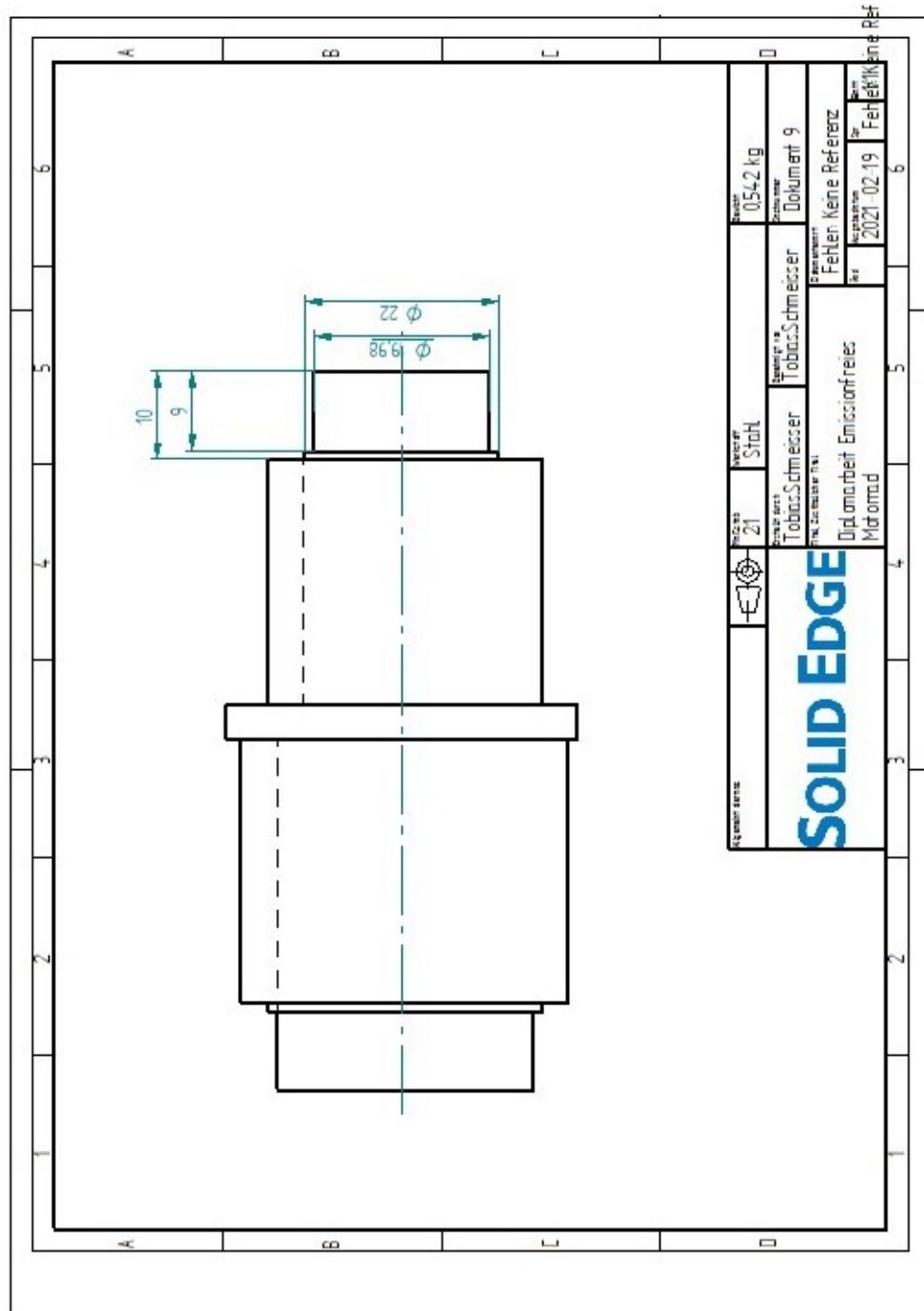


Abbildung C.8: Achse 2

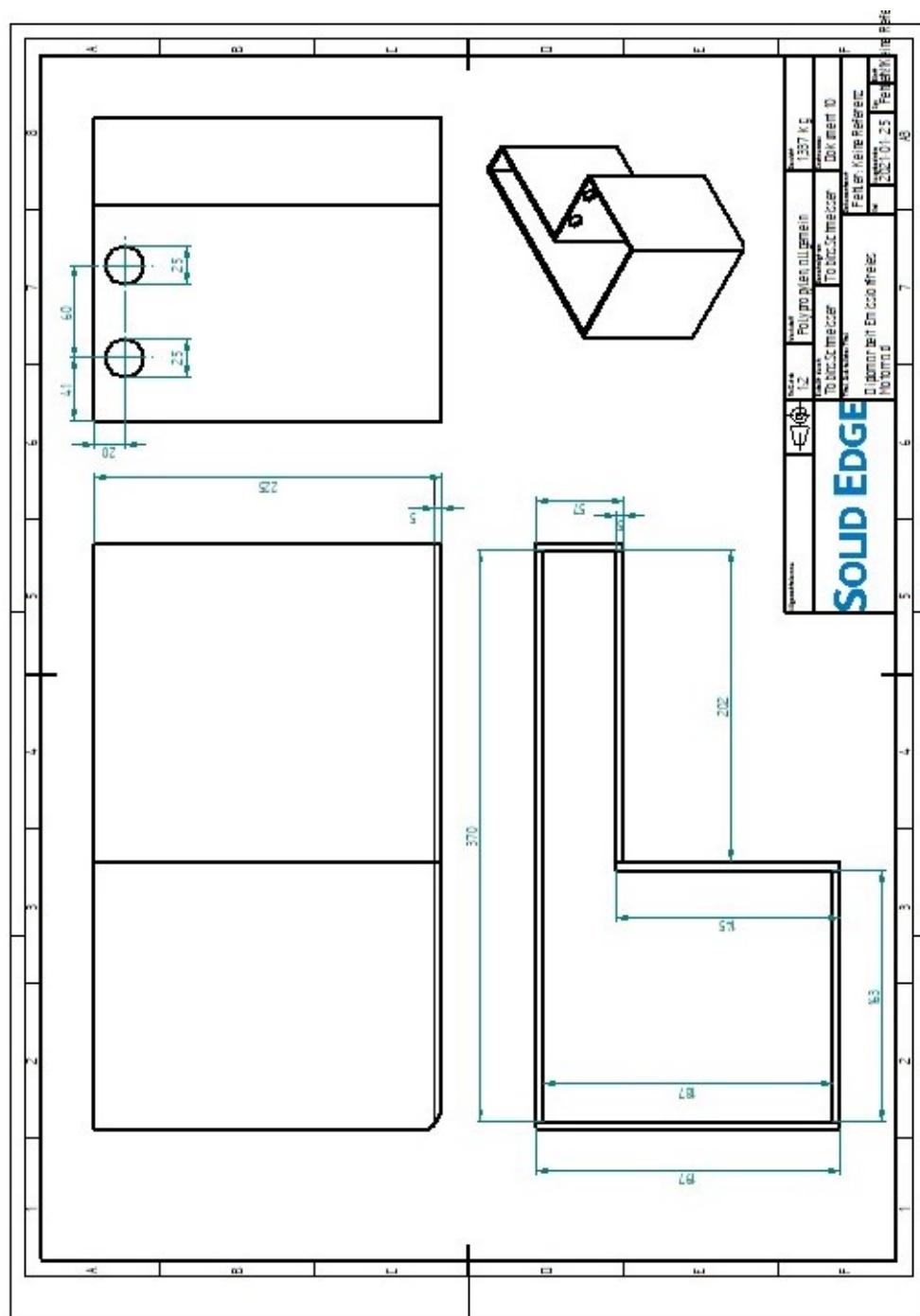


Abbildung C.9: Akkubox Motorblock

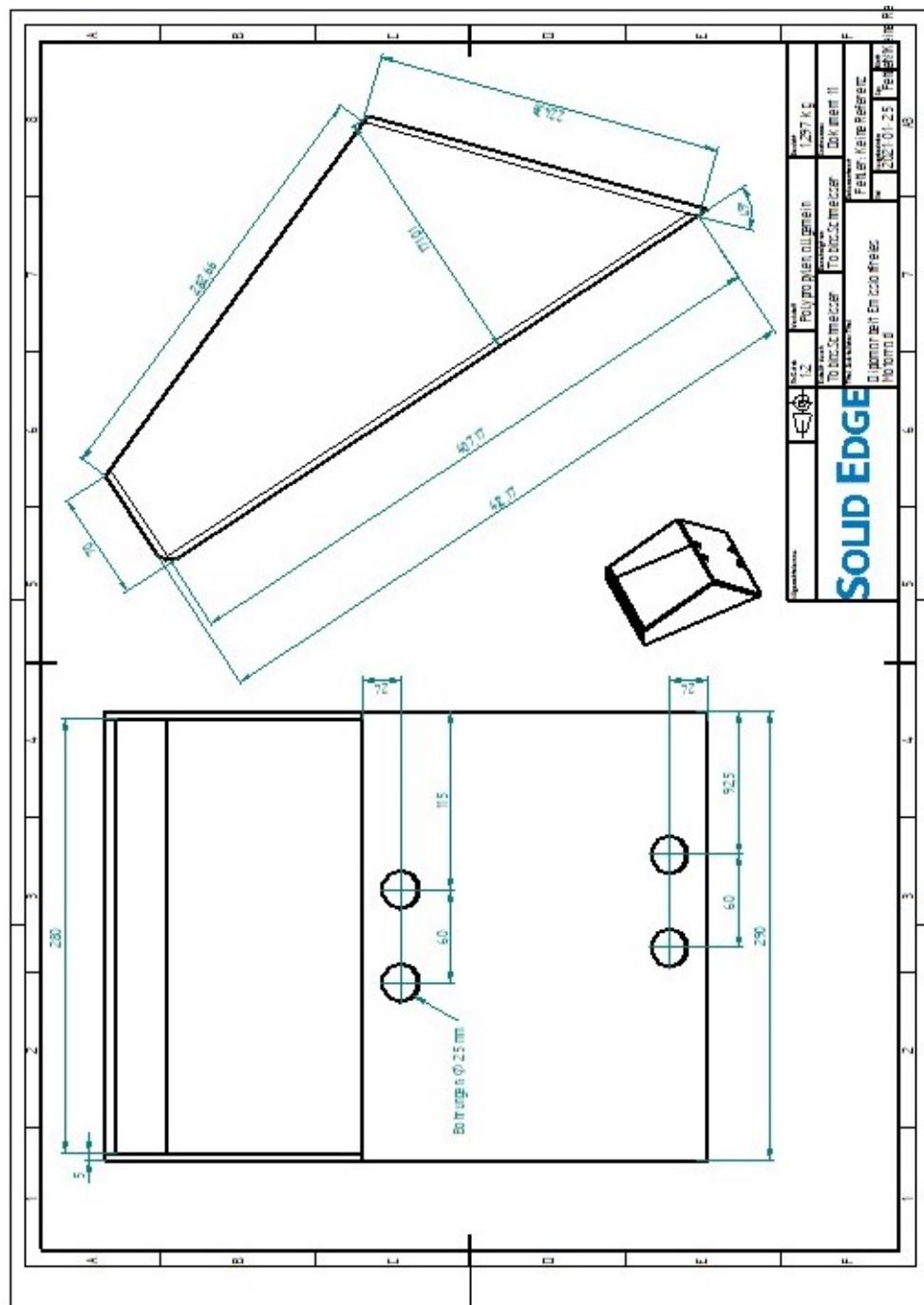


Abbildung C.10: Akkubox Vorderseite

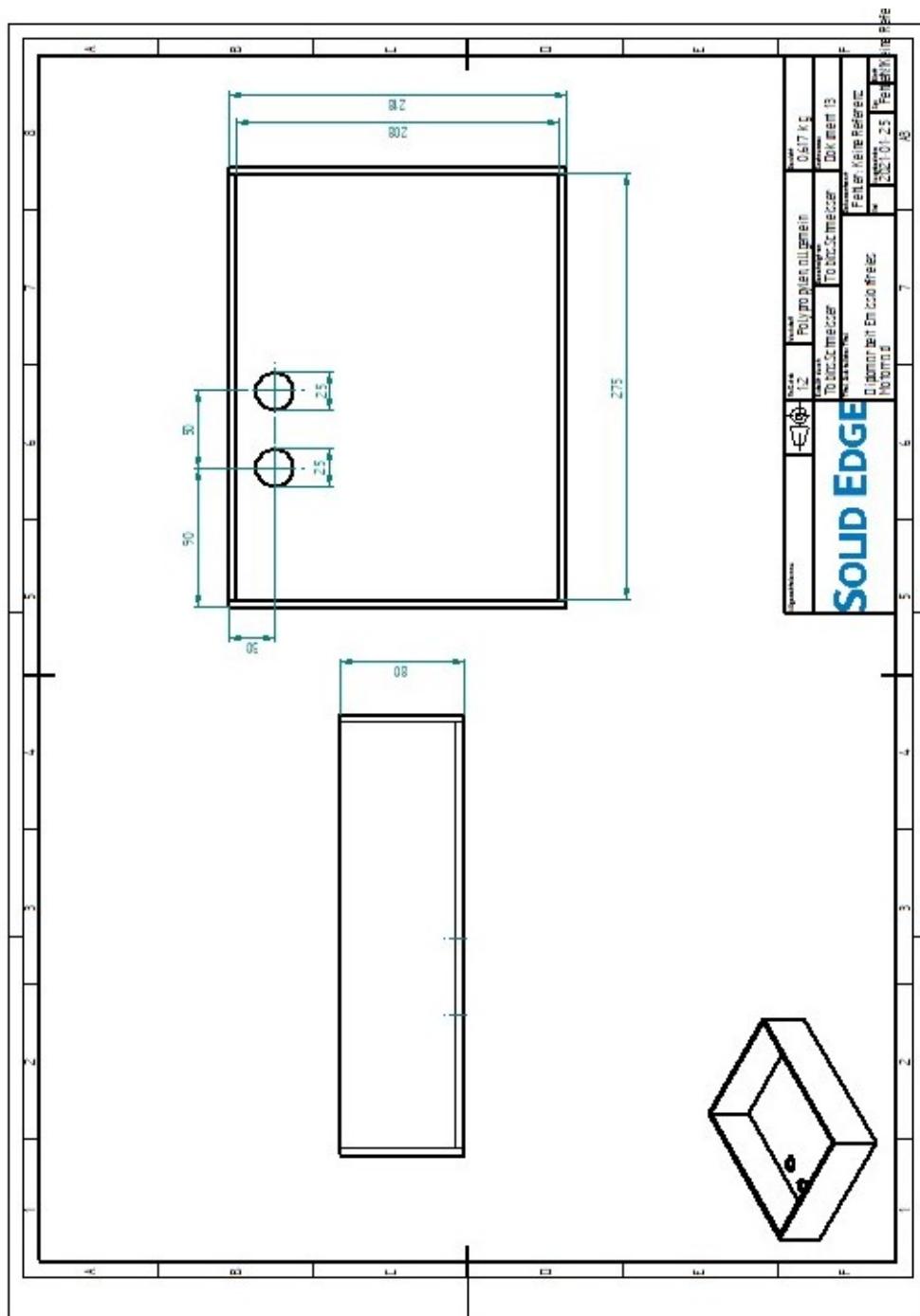


Abbildung C.11: Akkubox Mitte

2 Antriebsstrang

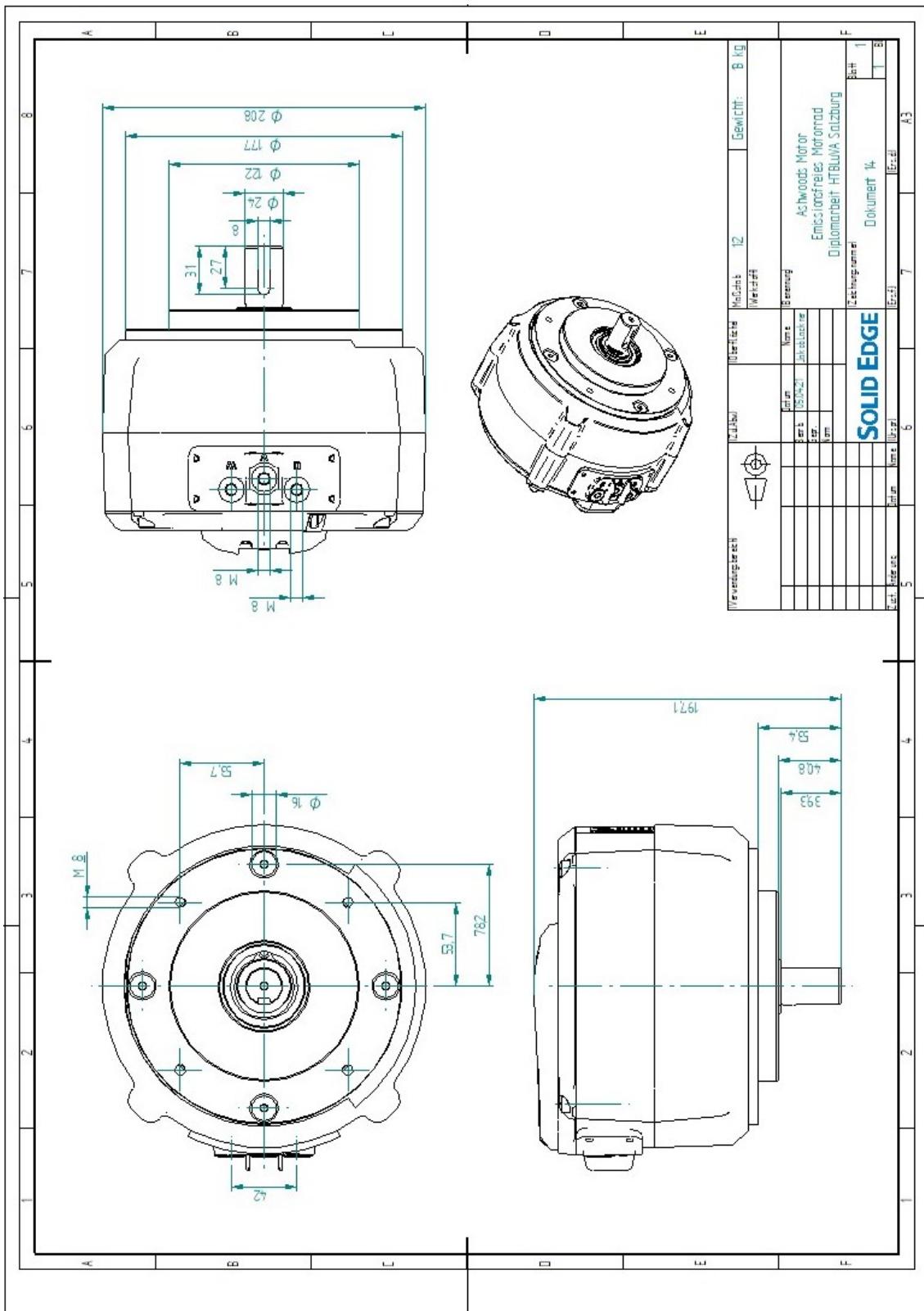


Abbildung C.12: Ashwoods Motor

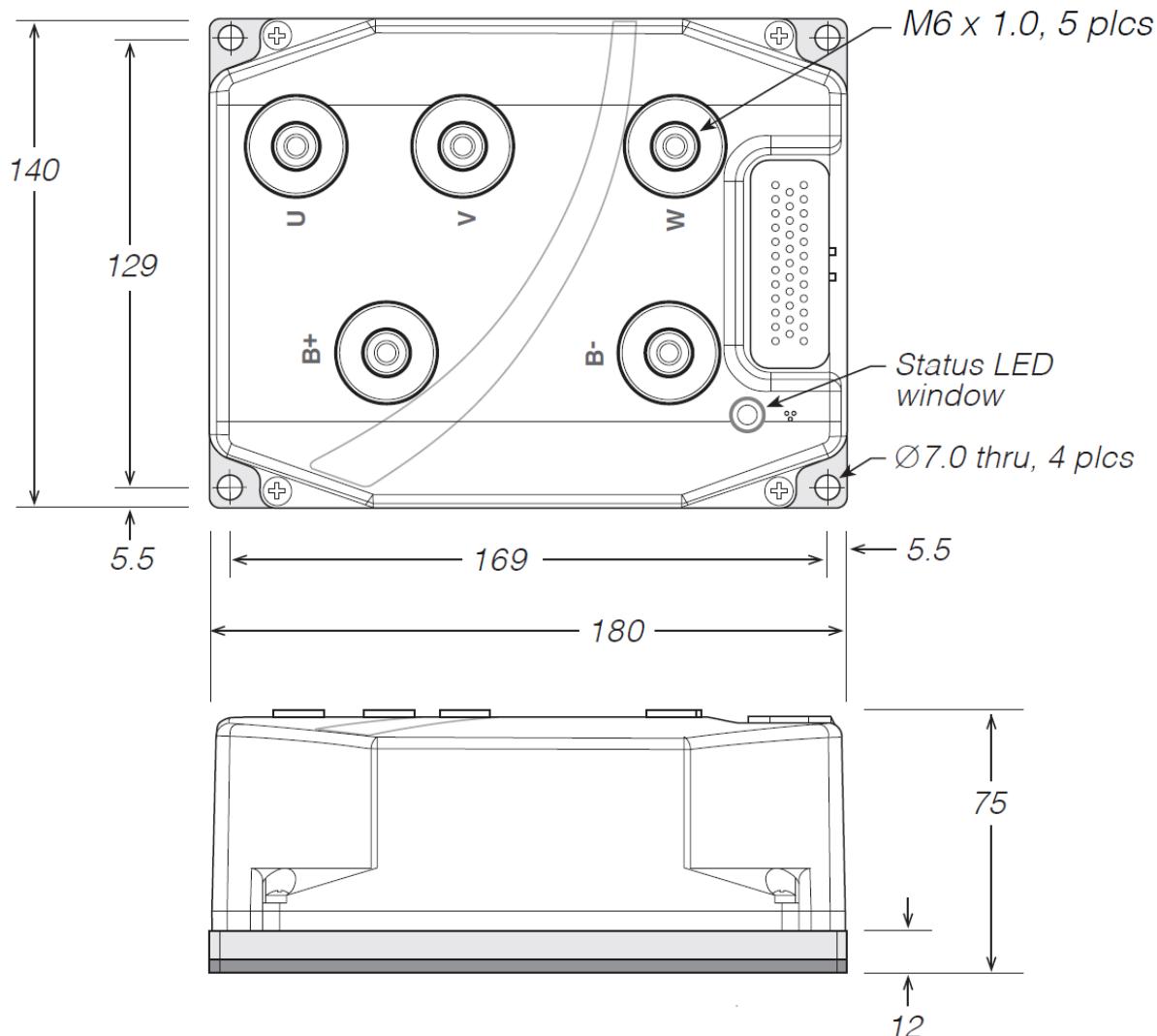


Abbildung C.13: Curtis Controller

3 HCIS

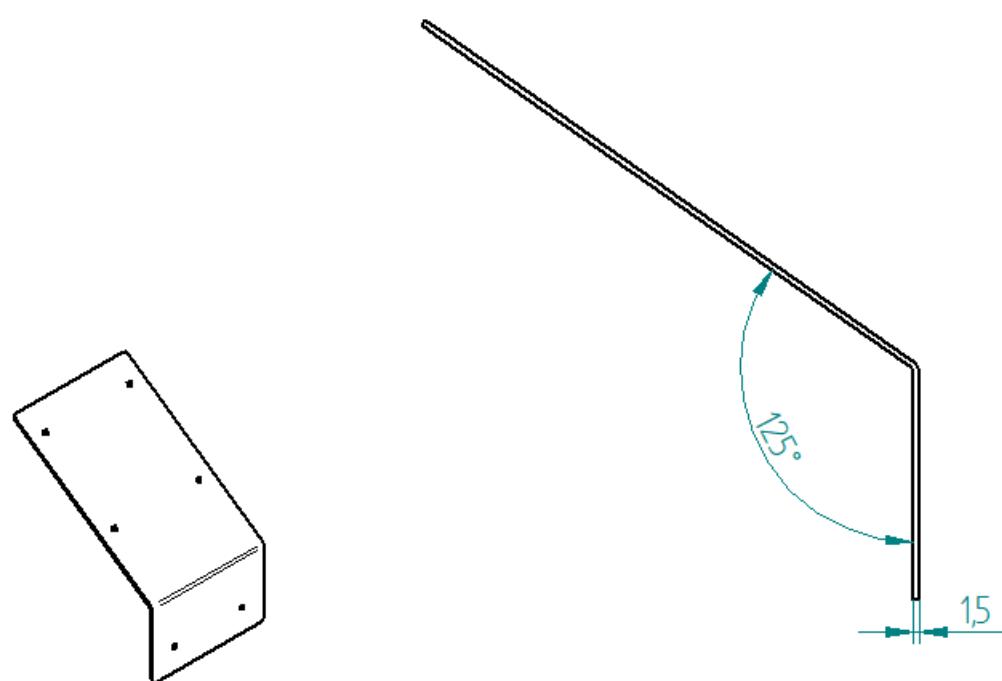
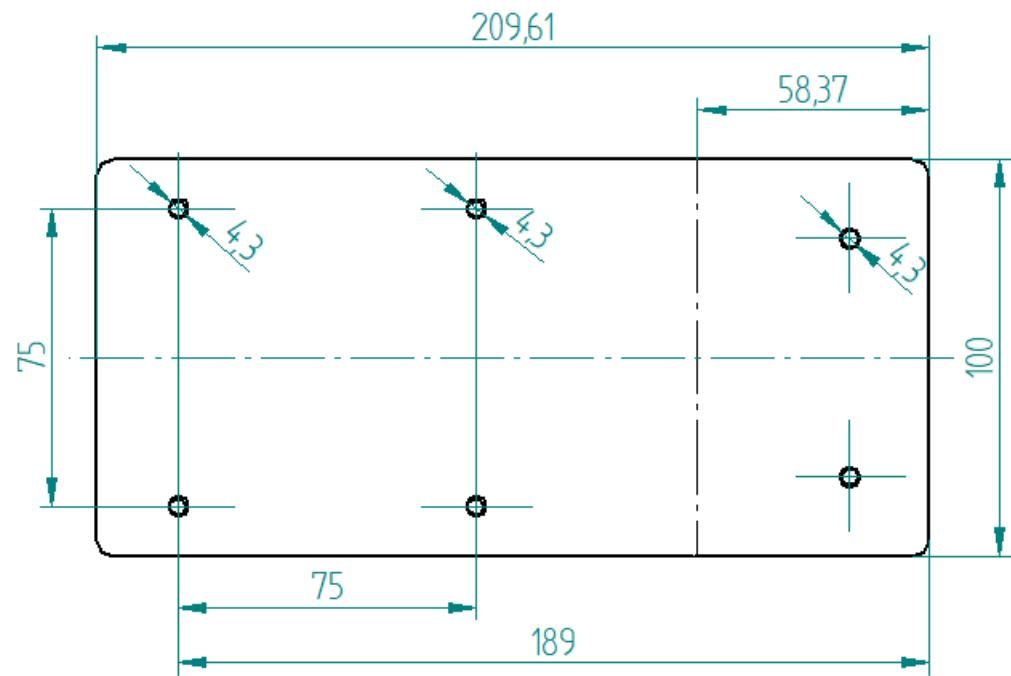


Abbildung C.14: Akkubox Mitte

Anhang D

Simulationen

1 Mechanik

Simulationsbericht_Seitenplatte_Links_ALU 5754

Unternehmen

Autor

Tobias.Schmeisser

Datum

Donnerstag, 18. Februar 2021

Verwendete Software

Solid Edge (219.00.00.091 x64)
Femap (11.3.1)

Verwendeter Solver

NX Nastran (11.0)

Inhaltsverzeichnis

1. [Einführung](#)
 2. [Modellinformationen](#)
 3. [Analyseeigenschaften](#)
 4. [Berechnete Geometrie](#)
 5. [Materialeigenschaften](#)
 6. [Überschreibungseigenschaften](#)
 7. [Lasten](#)
 8. [Randbedingungen](#)
 9. [Vernetzungsinformationen](#)
 10. [Ergebnisse](#)
 11. [Optimierungen](#)
 12. [Schluss](#)
 13. [Haftungsausschluss](#)
-

1. Einführung

2. Modellinformationen

Dokument	C:\Users\Tobias.Schmeisser\Desktop\Später löschn\Projekt\Bauteile\Seitenplatte_L\Seitenplatte_Links.par
----------	---

3. Berechnungseigenschaften

Berechnungseigenschaft	Wert
Berechnungsname	Statische Berechnung 1
Berechnungstyp	Linear statisch
Vernetzungstyp	Tetraeder
Iterativer Solver	Ein
NX Nastran-Geometrieprüfung	Ein
NX Nastran-Befehlszeile	
NX Nastran-Berechnungsoptionen	
Von NX Nastran generierte Optionen	
NX Nastran-Standardoptionen	
Option-Nur Flächenergebnisse	Ein

4. Berechnete Geometrie

4.1 Körper

Name	Material	Masse	Volumen	Gewicht
Seitenplatte_Links.par	Aluminiumlegierung:3.3535 , AlMg3, EN-AW 5754	5,106 kg	1,919 dm ³	50036,229 mN

5. Materialeigenschaften

5.1 Aluminiumlegierung:3.3535 , AlMg3, EN-AW 5754

Eigenschaft	Wert
Dichte	2,660 kg/dm ³
Wärme- Ausdehnungskoeff.	0,0000 /c
Wärmeleitfähigkeit	0,204 kW/m-C
Spezifische Wärme	940,000 J/kg-C
Elastizitätsmodul	70000,000 MegaPa
Poissonscher Beiwert	0,390
Streckgrenze	250,000 MegaPa
Zugfestigkeit	290,000 MegaPa

Bruchdehnung %	0,000
----------------	-------

6. Überschreibungseigenschaften

7. Lasten

Nam e	Lastenty p	Lastenwer t	Lastverteilun g	Lastenrichtun g	Lastenrichtungsoptio n
Kraft 1	Kraft	Fx: 2e+06 mN, Fy: 0 mN, Fz: 4e+06 mN Fx: -2e+06	Pro Entität		Komponenten
Kraft 2	Kraft	mN, Fy: 0 mN, Fz: - 4e+06 mN Fx: -2e+06	Pro Entität		Komponenten
Kraft 3	Kraft	mN, Fy: 0 mN, Fz: - 4e+06 mN	Pro Entität		Komponenten

8. Randbedingungen

Name	Randbedingungstyp	Freiheitsgrade
Fixiert 1	Fixiert	Freie Freiheitsgrade: Keine

9. Vernetzungsinformationen

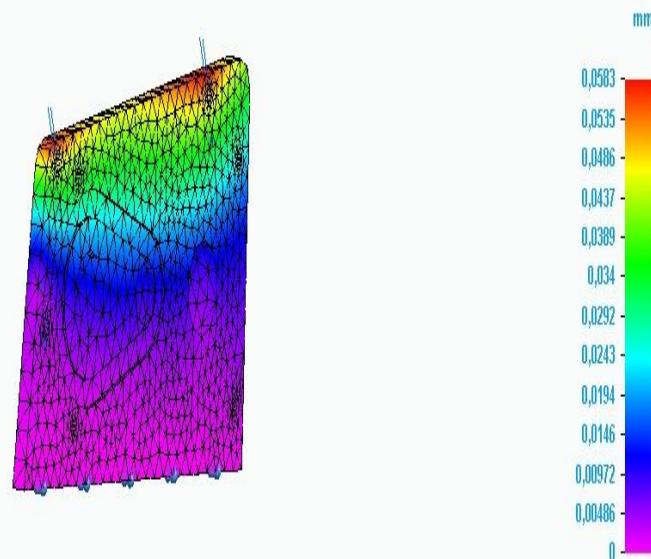
Vernetzungstyp	Tetraeder
Gesamtzahl der vernetzten Körper	1
Gesamtzahl der Elemente	6.645
Gesamtzahl der Knoten	11.781
Subjektive Netzgröße (1-10)	1

10. Ergebnisse

10.1 Verschiebungsergebnisse

Ergebniskomponente: Gesamtverschiebung					
Ausmaß	Wert	x	y	z	
Mindestwert	0 mm	11,583 mm	-0,000 mm	-231,679 mm	
Höchstwert	0,0583 mm	-57,198 mm	-0,000 mm	22,140 mm	

Seitenplatte_Links.par, Statische Berechnung 1, Aluminiumlegierung 3.3535, AlMg3, EN-AW 5754
 Verschiebung - Knoten
 Kontur: Resultierende Verschiebung
 Verformung: Gesamtverschiebung
 Datum: Donnerstag, 18. Februar 2021 18:56

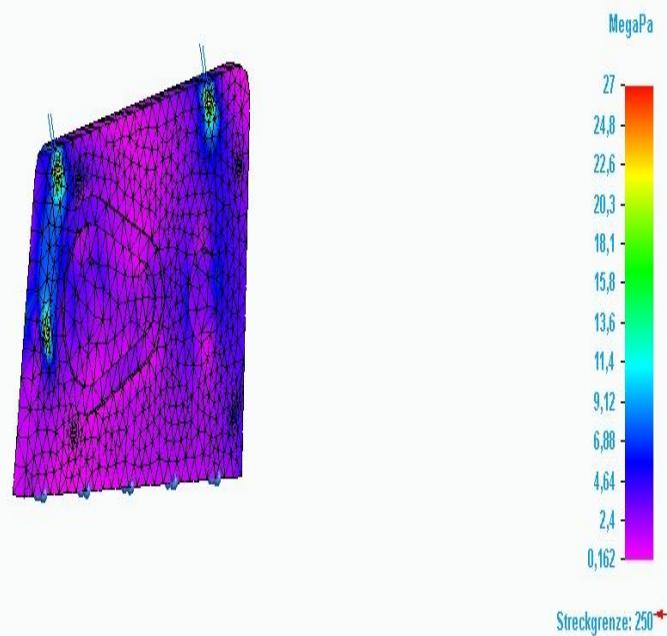


Gesamtverschiebung

10.2 Spannungsergebnisse

Ergebniskomponente: Von Mises					
Ausmaß	Wert	x	y	z	
Mindestwert	0,162 MegaPa	-145,584 mm	-0,000 mm	24,940 mm	
Höchstwert	27 MegaPa	252,593 mm	-0,000 mm	-26,734 mm	

Seitenplatte Links.par, Statische Berechnung 1, Aluminiumlegierung 3.3535, AlMg3, EN-AW 5754
 Spannung - Elemente
 Kontur: Von Mises
 Verformung: Gesamtverschiebung
 Datum: Donnerstag, 18. Februar 2021 18:56

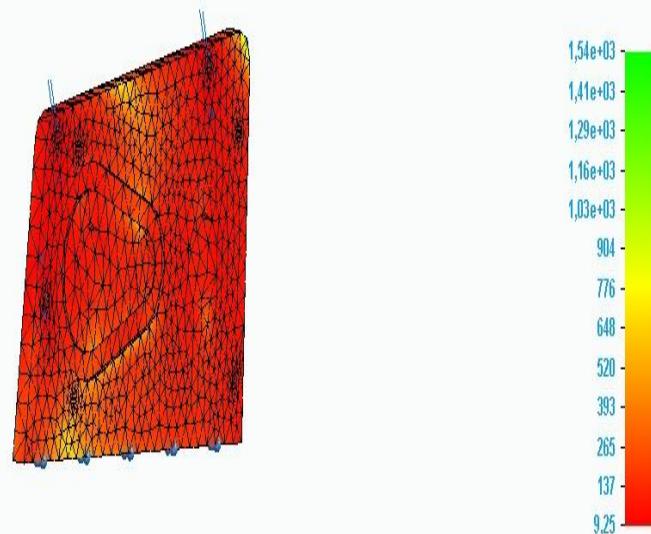


Von Mises

10.3 Sicherheitsfaktorergebnisse

Ergebniskomponente: Sicherheitsfaktor					
Ausmaß	Wert	x	y	z	
Mindestwert	9,25	252,593 mm	-0,000 mm	-26,734 mm	
Höchstwert	1,54e+03	-145,584 mm	-0,000 mm	24,940 mm	

Seitenplatte Links.par, Statische Berechnung 1, Aluminiumlegierung:3.3535 , AlMg3, EN-AW 5754
Spannung - Elemente
Kontur: Sicherheitsfaktor
Verformung: Gesamtverschiebung
Datum: Donnerstag, 18. Februar 2021 18:56



Sicherheitsfaktor

11. Optimierungen

12. Schluss

13. Haftungsausschluss

Wichtige Informationen

Dieser Auszug sollte nicht als einziges Mittel zur Bewertung einer Entwurfsidee hinsichtlich ihrer Eignung für einen bestimmten Satz von Umgebungsbedingungen verwendet werden. Siemens folgt dem Bestreben, immer ausreichend Hilfen und Benutzerführungen für Produkte zur Verfügung zu stellen. Diese ersetzen jedoch nicht ein gutes Beurteilungs- und Einschätzungsvermögen bei der Konstruktion, das in der Verantwortung unserer Anwender liegt. Ein qualitativer Ansatz bei der Konstruktion sollte sicherstellen, dass die Ergebnisse dieser Bewertung zusammen mit der praktischen Erfahrung von Konstrukteuren und Analytikern sowie experimentellen Testdaten in Betracht gezogen wird. Die in diesem Auszug enthaltenen Ergebnisse werden als verlässlich betrachtet, sollten jedoch nicht als Garantie jeglicher Art für die Tauglichkeit eines Zwecks ausgelegt werden.

Simulationsbericht_Seitenplatte_Rechts

Unternehmen

Autor

Tobias.Schmeisser

Datum

Mittwoch, 20. Januar 2021

Verwendete Software

Solid Edge (219.00.00.091 x64)

Femap (11.3.1)

Verwendeter Solver

NX Nastran (11.0)

Inhaltsverzeichnis

1. [Einführung](#)
 2. [Modellinformationen](#)
 3. [Analyseeigenschaften](#)
 4. [Berechnete Geometrie](#)
 5. [Materialeigenschaften](#)
 6. [Überschreibungseigenschaften](#)
 7. [Lasten](#)
 8. [Randbedingungen](#)
 9. [Vernetzungsinformationen](#)
 10. [Ergebnisse](#)
 11. [Optimierungen](#)
 12. [Schluss](#)
 13. [Haftungsausschluss](#)
-

1. Einführung

2. Modellinformationen

Dokument	C:\Users\Tobias.Schmeisser\Desktop\Später löschn\Projekt\Bauteile\Seitenplatte_R\Seitenplatte_Rechts.par
----------	--

3. Berechnungseigenschaften

Berechnungseigenschaft	Wert
Berechnungsname	Statische Berechnung 1
Berechnungstyp	Linear statisch
Vernetzungstyp	Tetraeder
Iterativer Solver	Ein
NX Nastran-Geometrieprüfung	Nur Warnung
NX Nastran-Befehlszeile	
NX Nastran-Berechnungsoptionen	
Von NX Nastran generierte Optionen	
NX Nastran-Standardoptionen	
Option-Nur Flächenergebnisse	Ein

4. Berechnete Geometrie

4.1 Körper

Name	Material	Masse	Volumen	Gewicht
Seitenplatte_Rechts.par	Stahllegierung:1.7005 , 45Cr2	16,331 kg	2,094 dm ³	160048,125 mN

5. Materialeigenschaften

5.1 Stahllegierung:1.7005 , 45Cr2

Eigenschaft	Wert
Dichte	7,800 kg/dm ³
Wärme- Ausdehnungskoeff.	0,0000 /c
Wärmeleitfähigkeit	0,014 kW/m-C
Spezifische Wärme	440,000 J/kg-C
Elastizitätsmodul	190000,000 MegaPa
Poissonscher Beiwert	0,280
Streckgrenze	540,000 MegaPa
Zugfestigkeit	780,000 MegaPa

Bruchdehnung %	0,000
----------------	-------

6. Überschreibungseigenschaften

7. Lasten

Nam e	Lastenty p	Lastenwer t	Lastverteilun g	Lastenrichtun g	Lastenrichtungsoptio n
Kraft 2	Kraft	Fx: -2e+06 mN, Fy: 0 mN, Fz: - 4e+06 mN Fx: 2e+06 mN, Fy: - 1e+03 mN, Fz: 4e+06 mN	Insgesamt		Komponenten
Kraft 3	Kraft		Pro Entität		Komponenten

8. Randbedingungen

Name	Randbedingungstyp	Freiheitsgrade
Fixiert 4	Fixiert	Freie Freiheitsgrade: Keine

9. Vernetzungsinformationen

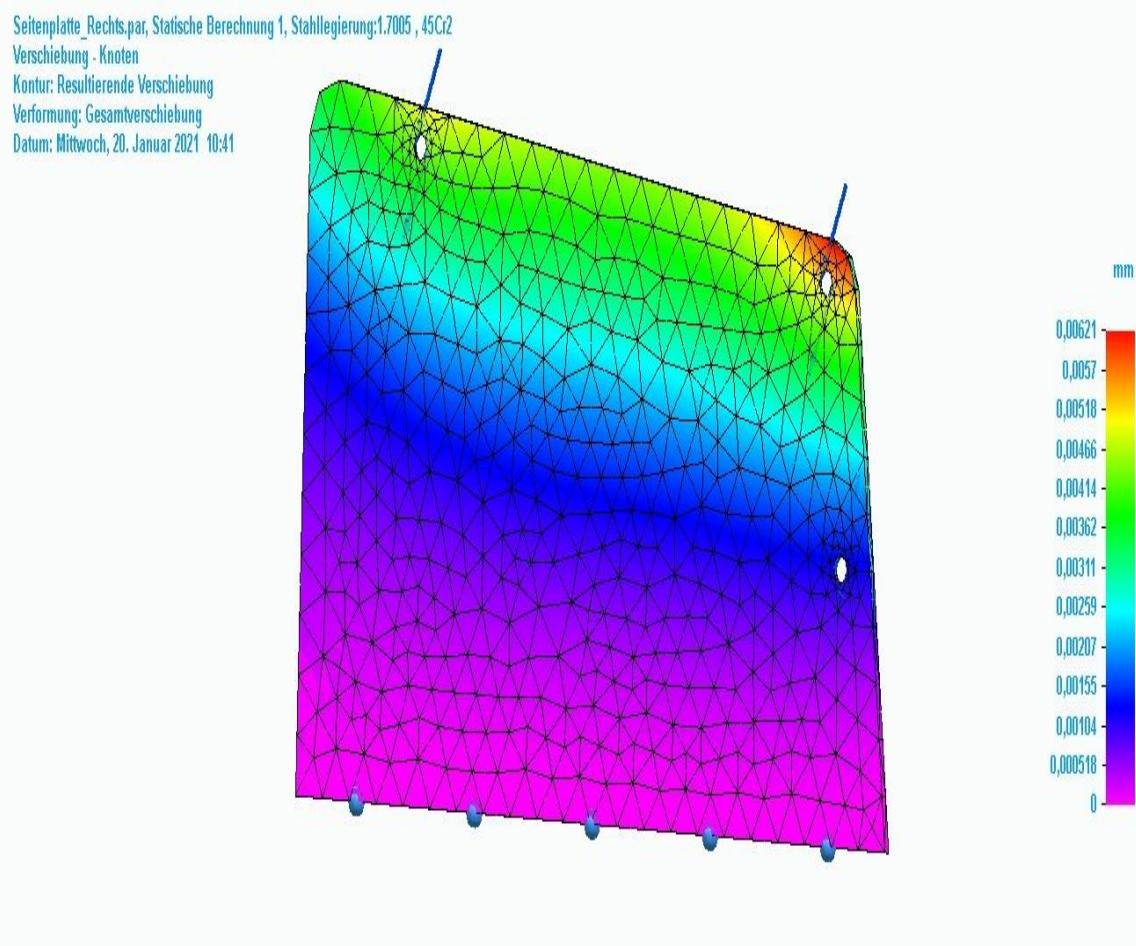
Vernetzungstyp	Tetraeder
Gesamtzahl der vernetzten Körper	1
Gesamtzahl der Elemente	15.836
Gesamtzahl der Knoten	27.527
Subjektive Netzgröße (1-10)	1

10. Ergebnisse

10.1 Verschiebungsergebnisse

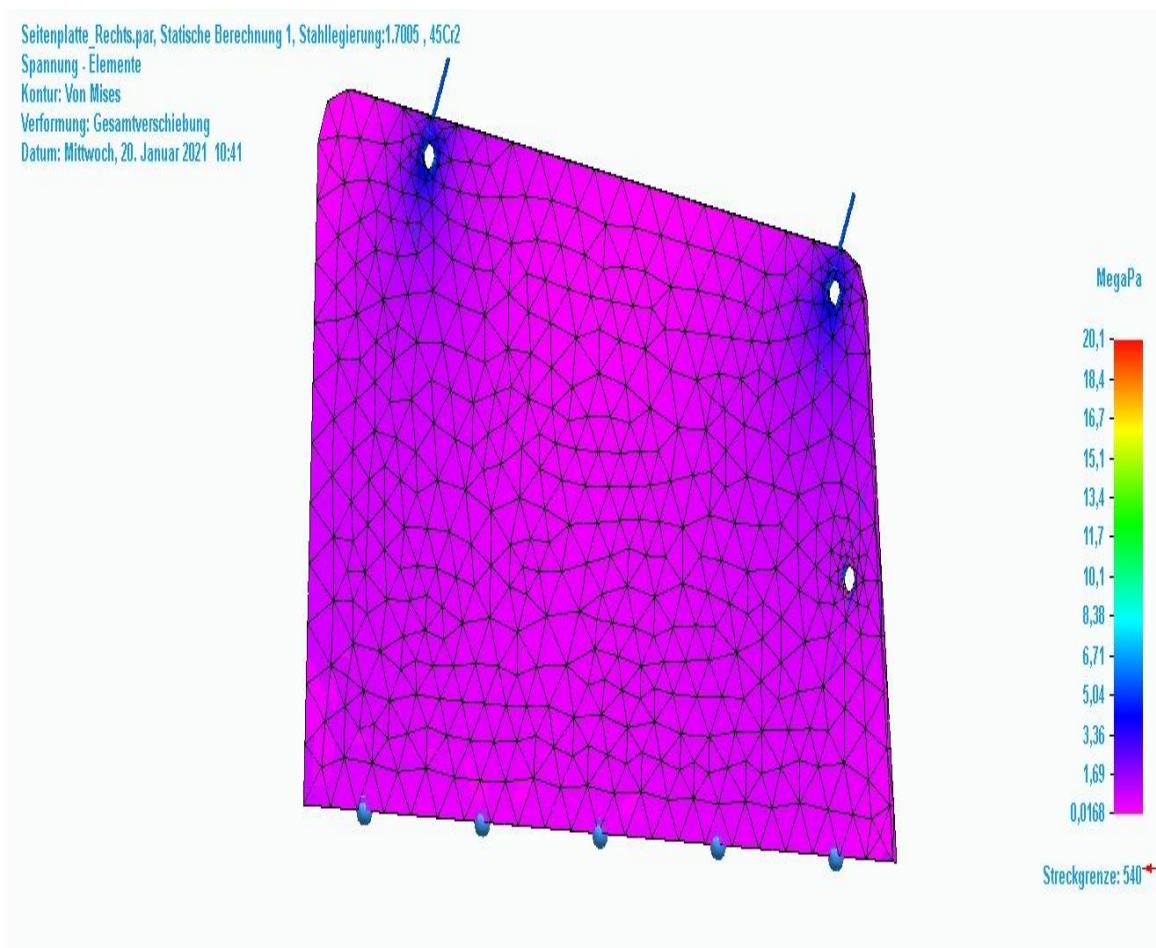
Ergebniskomponente: Gesamtverschiebung
--

Ausmaß	Wert	x	y	z
Mindestwert	0 mm	-75,140 mm	0,000 mm	-101,201 mm
Höchstwert	0,00621 mm	184,882 mm	-20,000 mm	119,429 mm



10.2 Spannungsergebnisse

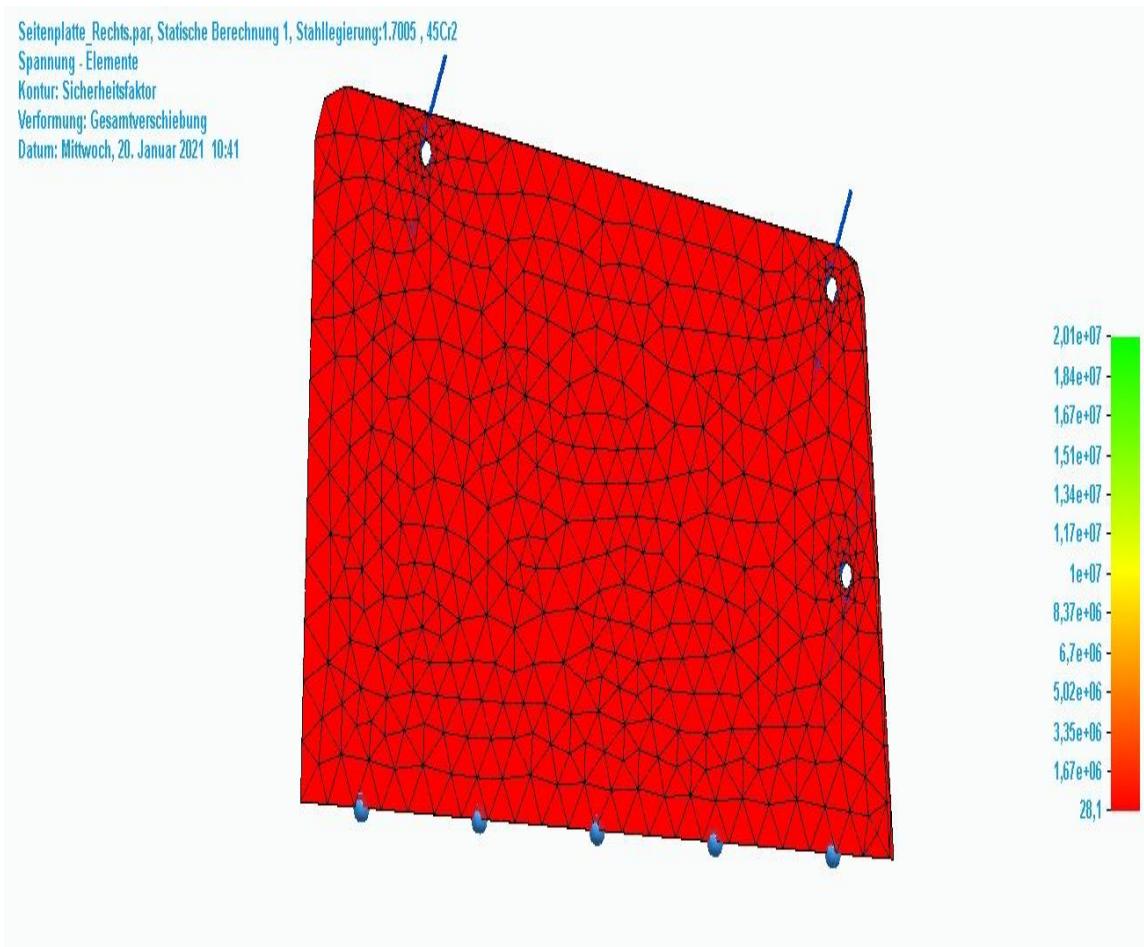
Ergebniskomponente: Von Mises					
Ausmaß	Wert	x	y	z	
Mindestwert	0,0168 MegaPa	-219,913 mm	0,000 mm	74,612 mm	
Höchstwert	20,1 MegaPa	192,541 mm	-10,000 mm	3,780 mm	



Von Mises

10.3 Sicherheitsfaktorergebnisse

Ergebniskomponente: Sicherheitsfaktor					
Ausmaß	Wert	x	y	z	
Mindestwert	28,1	191,076 mm	-20,000 mm	0,244 mm	
Höchstwert	2,01e+07	192,541 mm	-10,000 mm	3,780 mm	



11. Optimierungen

12. Schluss

13. Haftungsausschluss

Wichtige Informationen

Dieser Auszug sollte nicht als einziges Mittel zur Bewertung einer Entwurfsidee hinsichtlich ihrer Eignung für einen bestimmten Satz von Umgebungsbedingungen verwendet werden. Siemens folgt dem Bestreben, immer ausreichend Hilfen und Benutzerführungen für Produkte zur Verfügung zu stellen. Diese ersetzen jedoch nicht ein gutes Beurteilungs- und Einschätzungsvermögen bei der Konstruktion, das in der Verantwortung unserer Anwender liegt. Ein qualitativer Ansatz bei der Konstruktion sollte sicherstellen, dass die Ergebnisse dieser Bewertung zusammen mit der praktischen Erfahrung von Konstrukteuren und Analytikern sowie experimentellen Testdaten in Betracht gezogen wird. Die in diesem Auszug enthaltenen Ergebnisse werden als verlässlich betrachtet, sollten jedoch nicht als Garantie jeglicher Art für die Tauglichkeit eines Zwecks ausgelegt werden.

Anhang E

Schaltpläne

1 Antriebsstrang

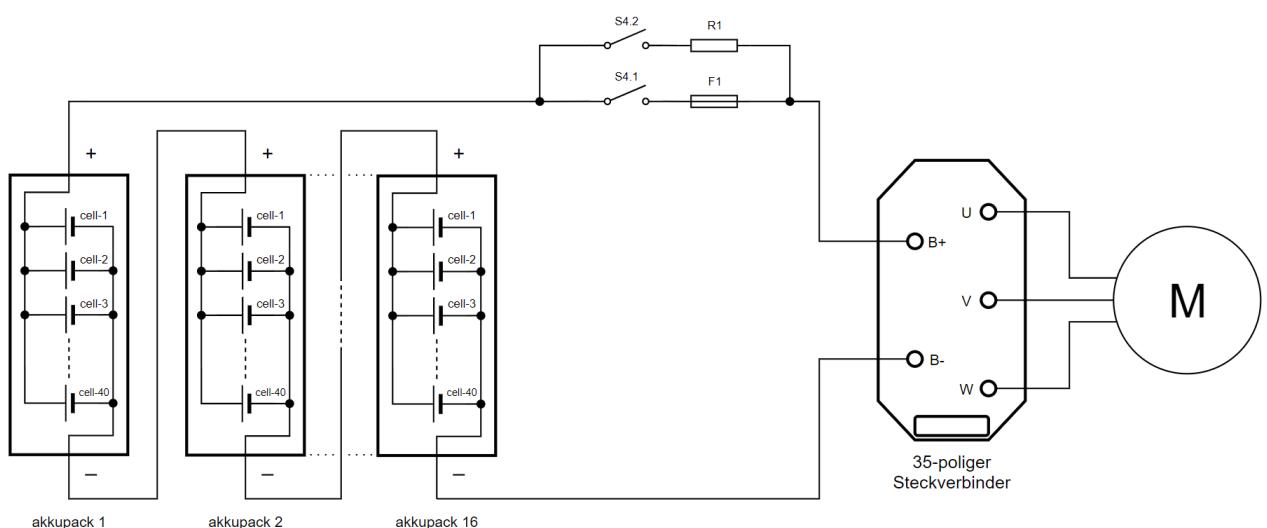


Abbildung E.1: Grundaufbau des Laststromkreises

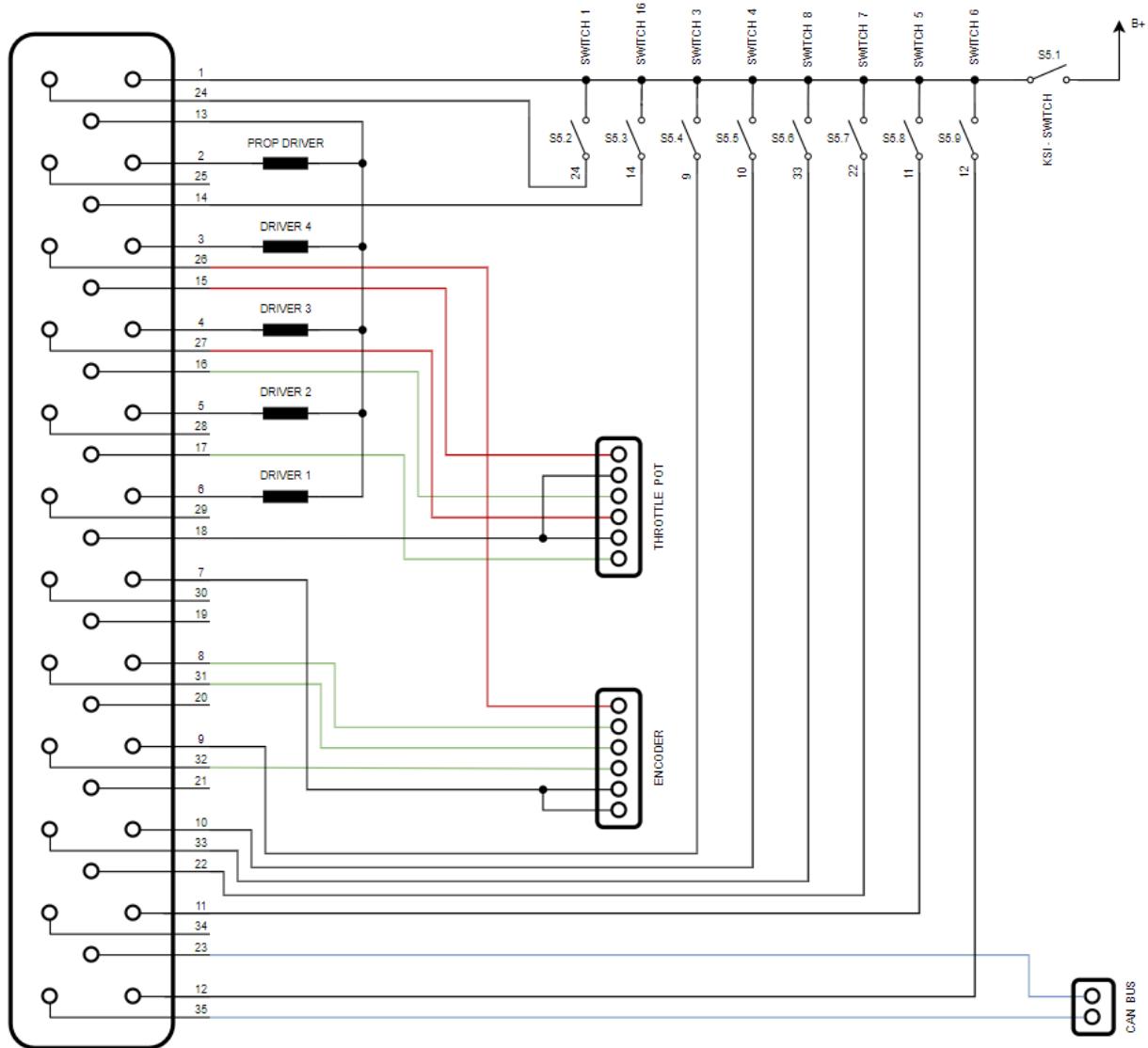


Abbildung E.2: Grundaufbau des Steuerstromkreises

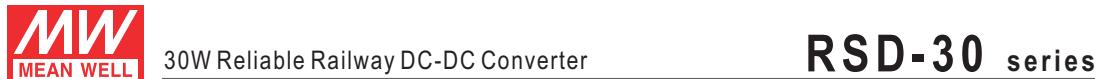
2 HCIS

Anhang F

Datenblätter

1 HCIS

1.1 Mean Well RSD-30H-5



■ Features

- Compliance to EN50155 and EN45545-2 railway standard
- Ultra compact and 1U low profile(25mm)
- 4:1 wide input range
- No minimum load required
- Protections: Short circuit / Overload / Over voltage / Input reverse polarity
- 4000VDC I/O isolation (reinforced isolation)
- Half encapsulated , cooling by free air convection
- -40~+70°C wide working temperature
- Built-in constant current limiting circuit
- LED indicator for power on
- 3 years warranty

■ Applications

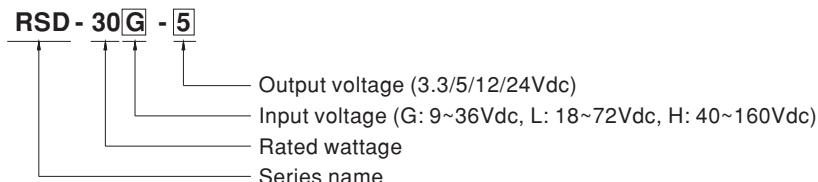
- Bus,tram,metro or railway system
- Wireless network
- Telecom or datacom system
- Highly vibrating, highly dusty, extremely low or high temperature harsh environment

■ Description

RSD-30 is a 30W enclosed type DC-DC reliable railway converter. This series is compliant with EN50155/IEC60571 railway standard, constituting three types of models with 4:1 wide but different input ranges 9~36V/18~72V/40~160V, suitable for railway and all kinds of transportation systems exploiting the frequently used standard input voltages such as 12V, 24V, 36V, 48V, 72V, 96V and 110V. Various output voltages, 3.3V, 5V, 12V and 24V are available for selection.

This series has the capability of working under -40~+70°C, low ripple and noise, supreme EMC characteristics, 4KVDC I/P-OP, low enclosure profile 25mm and an interior with semi-potted silicone. It does not only well fits the in-car systems or the facilities by rails for railway, trams and buses but also can be used in the harsh environment with high vibration, high dust, extremely low or high temperature, etc.

■ Model Encoding



File Name:RSD-30-SPEC 2017-02-22



30W Reliable Railway DC-DC Converter

RSD-30 series**SPECIFICATION**

MODEL	RSD-30G-3.3	RSD-30G-5	RSD-30G-12	RSD-30G-24	RSD-30L-3.3	RSD-30L-5	RSD-30L-12	RSD-30L-24							
OUTPUT	DC VOLTAGE	3.3V	5V	12V	24V	3.3V	5V	12V	24V						
	RATED CURRENT	6A	6A	2.5A	1.25A	6A	6A	2.5A	1.25A						
	CURRENT RANGE	0 ~ 6A	0 ~ 6A	0 ~ 2.5A	0 ~ 1.25A	0 ~ 6A	0 ~ 6A	0 ~ 2.5A	0 ~ 1.25A						
	RATED POWER	19.8W	30W	30W	30W	19.8W	30W	30W	30W						
	RIPPLE & NOISE (max.) Note.2	70mVp-p	70mVp-p	60mVp-p	50mVp-p	70mVp-p	70mVp-p	60mVp-p	50mVp-p						
	VOLTAGE TOLERANCE Note.3	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%	±2.0%						
	LINE REGULATION	±0.5%	±0.5%	±0.3%	±0.2%	±0.5%	±0.5%	±0.3%	±0.2%						
	LOAD REGULATION	±0.5%	±0.5%	±0.3%	±0.2%	±0.5%	±0.5%	±0.3%	±0.2%						
	SETUP, RISE TIME	120ms, 85ms at full load													
HOLD UP TIME (Typ.)		G type comply with S1 level(3ms) @full load, S2 level(10ms) @80% load; L type comply with S2 level(10ms) @full load													
INPUT	VOLTAGE RANGE CONTINUOUS	9 ~ 36VDC				18 ~ 72VDC									
	EFFICIENCY (Typ.)	84%	85%	86.5%	89%	84%	86%	90%	91%						
	DC CURRENT (Typ.)	1.1A/24V				0.52A/48V									
	INRUSH CURRENT (Typ.)	20A/24VDC				20A/48VDC									
PROTECTION	OVERLOAD	105 ~ 135% rated output power													
		Protection type : Constant current limiting, recovers automatically after fault condition is removed													
	OVER VOLTAGE	3.8 ~ 4.5V	5.75 ~ 7V	13.8 ~ 16.2V	27.6 ~ 32.4V	3.8 ~ 4.5V	5.75 ~ 7V	13.8 ~ 16.2V	27.6 ~ 32.4V						
ENVIRONMENT	WORKING TEMP.	-40 ~ +55°C (no derating); +70°C @ 60% load by free air convection; +70°C (no derating with external base plate)													
	WORKING HUMIDITY	5 ~ 95% RH non-condensing													
	STORAGE TEMP., HUMIDITY	-40 ~ +85°C, 10 ~ 95% RH non-condensing													
	TEMP. COEFFICIENT	±0.03%/°C (0 ~ 50°C)													
	VIBRATION	10 ~ 500Hz, 5G 10min./1cycle, 60min. each along X, Y, Z axes; Mounting : compliance to IEC61373													
SAFETY & EMC (Note 4)	SAFETY STANDARDS	Meet IEC60950-1 (LVD)													
	WITHSTAND VOLTAGE	I/P-O/P:4KVDC I/P-FG:2.5KVDC O/P-FG:2.5KVDC													
	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:100M Ohms / 500VDC / 25°C / 70% RH													
	EMC EMISSION	Parameter	Standard	Test Level / Note											
		Conducted	EN55032	Class A											
		Radiated	EN55032	Class B											
		Harmonic Current	EN6100-3-2	Class A											
		Voltage Flicker	EN6100-3-3	-----											
	EMC IMMUNITY	Parameter	Standard	Test Level / Note											
		ESD	EN61000-4-2	Level 3, ±8KV air; Level 3, ±6KV contact											
		Radiated Field	EN61000-4-3	Level X											
		EFT / Burst	EN61000-4-4	Level 3, 2KV at power											
		Surge	EN61000-4-5	Level 3, 1KV Line-Line, Level 3, 2KV Line-Earth											
		Conducted	EN61000-4-6	Level 3											
	RAILWAY STANDARD	Compliance to EN45545-2 for fire protection; Meet EN50155 / IEC60571 including IEC61373 for shock & vibration, EN50121-3-2 for EMC													
OTHERS	MTBF	396.9K hrs min. MIL-HDBK-217F (25°C)													
	DIMENSION	113*60*25mm (L*W*H)													
	PACKING	0.25Kg; 56pcs/15Kg/0.83CUFT													
NOTE	1. All parameters NOT specially mentioned are measured at 24.48VDC input, rated load and 25°C of ambient temperature. 2. Ripple & noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uf & 47uf parallel capacitor. 3. Tolerance : includes set up tolerance, line regulation and load regulation. 4. The power supply is considered a component which will be installed into a final equipment. All the EMC tests are been executed by mounting the unit on a 360mm*360mm metal plate with 1mm of thickness. The final equipment must be re-confirmed that it still meets EMC directives. For guidance on how to perform these EMC tests, please refer to "EMI testing of component power supplies." (as available on http://www.meanwell.com) 5. Strongly recommended that external output capacitance should not exceed 5000uF.														



30W Reliable Railway DC-DC Converter

RSD-30 series**SPECIFICATION**

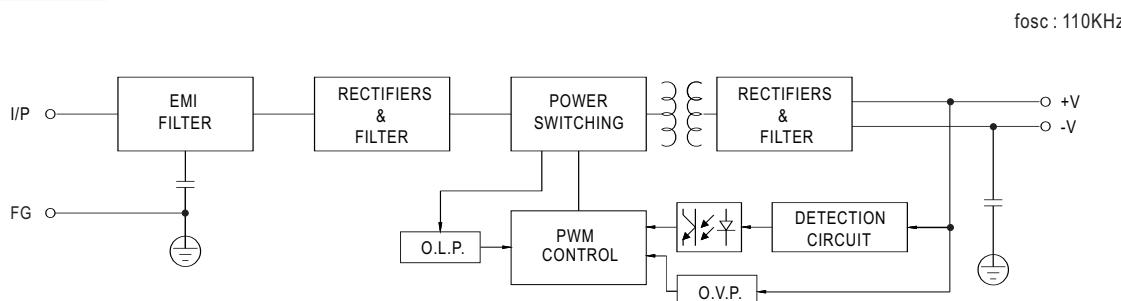
MODEL	RSD-30H-3.3	RSD-30H-5	RSD-30H-12	RSD-30H-24
OUTPUT	DC VOLTAGE	3.3V	5V	12V
	RATED CURRENT	6A	6A	2.5A
	CURRENT RANGE	0 ~ 6A	0 ~ 6A	0 ~ 2.5A
	RATED POWER	19.8W	30W	30W
	RIPPLE & NOISE (max.) Note.2	70mVp-p	70mVp-p	60mVp-p
	VOLTAGE TOLERANCE Note.3	±2.0%	±2.0%	±2.0%
	LINE REGULATION	±0.5%	±0.5%	±0.3%
	LOAD REGULATION	±0.5%	±0.5%	±0.3%
	SETUP, RISE TIME	120ms, 85ms at full load		
	HOLD UP TIME (Typ.)	H-type comply with S2 level(10ms) @ full load		
INPUT	VOLTAGE RANGE CONTINUOUS	40 ~ 160VDC		
	EFFICIENCY (Typ.)	87%	89%	89%
	DC CURRENT (Typ.)	0.23A/110V	0.35A/110V	
	INRUSH CURRENT (Typ.)	20A/110VDC		
PROTECTION	OVERLOAD	105 ~ 135% rated output power		
		Protection type : Constant current limiting, recovers automatically after fault condition is removed		
	OVER VOLTAGE	3.8 ~ 4.5V	5.75 ~ 7V	13.8 ~ 16.2V
ENVIRONMENT	OVER VOLTAGE	27.6 ~ 32.4V		
	Protection type : Shut down o/p voltage, re-power on to recover			
	WORKING TEMP.	-40 ~ +55°C (no derating) ; +70°C @ 60% load by free air convection ; +70°C (no derating with external base plate)		
	WORKING HUMIDITY	5 ~ 95% RH non-condensing		
	STORAGE TEMP., HUMIDITY	-40 ~ +85°C, 10 ~ 95% RH non-condensing		
SAFETY & EMC (Note 4)	TEMP. COEFFICIENT	±0.03%/°C (0 ~ 50°C)		
	VIBRATION	10 ~ 500Hz, 5G 10min./1cycle, 60min. each along X, Y, Z axes ; Mounting : compliance to IEC61373		
	SAFETY STANDARDS	Meet IEC60950-1 (LVD)		
	WITHSTAND VOLTAGE	I/P-O/P:4KVDC	I/P-FG:2.5KVDC	O/P-FG:2.5KVDC
	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:100M Ohms / 500VDC / 25°C / 70% RH		
	EMC EMISSION	Parameter	Standard	Test Level / Note
		Conducted	EN55032	Class A
		Radiated	EN55032	Class B
		Harmonic Current	EN6100-3-2	Class A
		Voltage Flicker	EN6100-3-3	-----
	EMC IMMUNITY	Parameter	Standard	Test Level / Note
		ESD	EN61000-4-2	Level 3, ±8KV air ; Level 3, ±6KV contact
		Radiated Field	EN61000-4-3	Level X
		EFT / Burst	EN61000-4-4	Level 3, 2KV at power Level 4, 2KV at signal
		Surge	EN61000-4-5	Level 3, 1KV Line-Line, Level 3, 2KV Line-Earth
		Conducted	EN61000-4-6	Level 3
OTHERS	RAILWAY STANDARD	Compliance to EN45545-2 for fire protection ; Meet EN50155 / IEC60571 including IEC61373 for shock & vibration, EN50121-3-2 for EMC		
	MTBF	396.9K hrs min. MIL-HDBK-217F (25°C)		
	DIMENSION	113*60*25mm (L*W*H)		
	PACKING	0.25Kg; 56pcs/15Kg/0.83CUFT		
NOTE	1. All parameters NOT specially mentioned are measured at 110VDC input, rated load and 25°C of ambient temperature. 2. Ripple & noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uf & 47uf parallel capacitor. 3. Tolerance : includes set up tolerance, line regulation and load regulation. 4. The power supply is considered a component which will be installed into a final equipment. All the EMC tests are been executed by mounting the unit on a 360mm*360mm metal plate with 1mm of thickness. The final equipment must be re-confirmed that it still meets EMC directives. For guidance on how to perform these EMC tests, please refer to "EMI testing of component power supplies." (as available on http://www.meanwell.com) 5. Strongly recommended that external output capacitance should not exceed 5000uF.			



30W Reliable Railway DC-DC Converter

RSD-30 series

■ Block Diagram



■ Input Fuse

There is one fuse connected in series to the positive input line, which is used to protect against abnormal surge. Fuse specifications of each model are shown as below.

Type	Fuse Type	Reference and Rating
G	Time-Lag	CONQUE MST, 6.3A, 250V
L	Time-Lag	CONQUE MST, 3.15A, 250V
H	Time-Lag	CONQUE MST, 2A, 250V

■ Input Reverse Polarity Protection

There is a MOSFET connected in series to the negative input line. If the input polarity is connected reversely, the MOSFET opens and there will be no output to protect the unit.

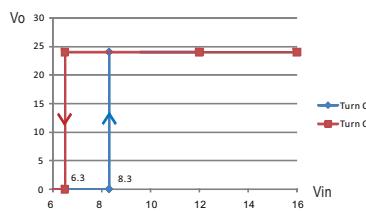
■ Input Range and Transient Ability

The series has a wide range input capability. With $\pm 40\%$ of rated input voltage, it can withstand that for 1 second.

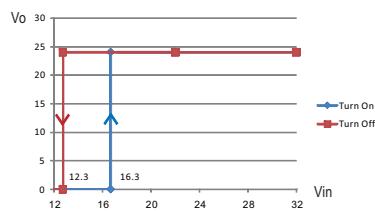
■ Input Under-Voltage Protection

If input voltage drops below V_{min} , the internal control IC shuts down and there is no output voltage. It recovers automatically when input voltage reaches above V_{min} , please refer to the curve below.

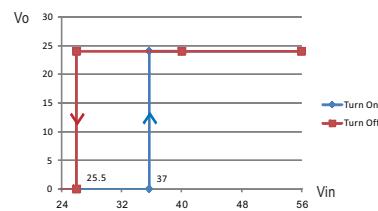
RSD-30G-24



RSD-30L-24



RSD-30H-24



■ Inrush Current

Inrush current is suppressed by a resistor during the initial start-up, and then the resistor is bypassed by a MOSFET to reduce power consumption after accomplishing the start-up.



30W Reliable Railway DC-DC Converter

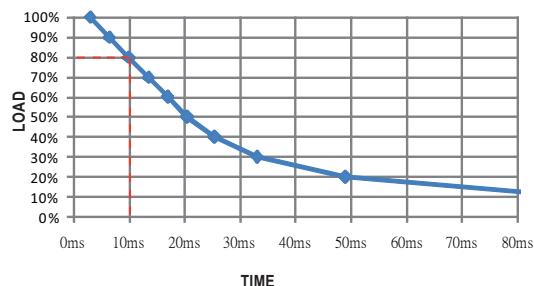
RSD-30 series

■ Hold-up Time

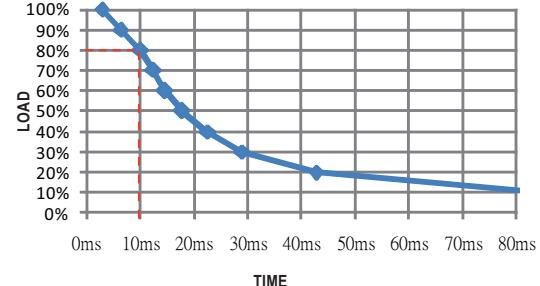
H type is in compliance with S2 level (10ms), while G and L types are in compliance with S1 level (3ms) at full load output condition.

To fulfil the requirements of S2 level (10ms), G types require de-rating their output load to 80%, please refer to the curve diagrams below.

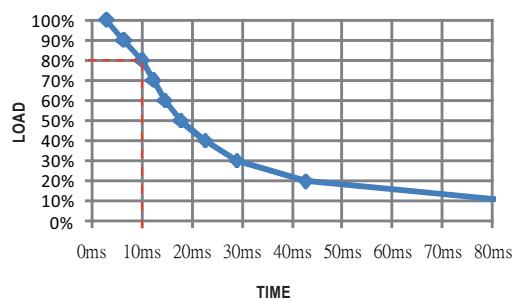
RSD-30G-3.3



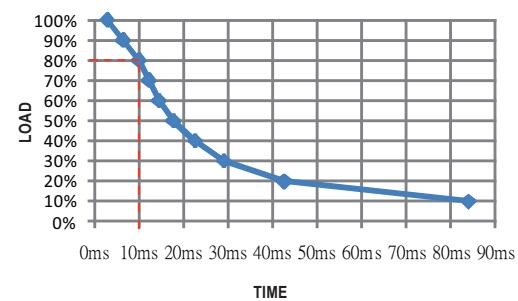
RSD-30G-5



RSD-30G-12



RSD-30G-24



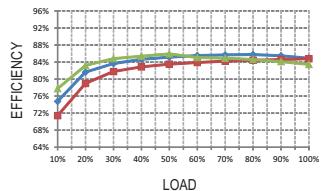
■ Output Voltage Adjustment

This function is optional, which the standard product does not have it. If you do need the function, please contact MW for details.

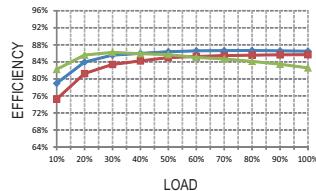
■ Efficiency vs Load & Vin Curve

The efficiency vs load & Vin curves of each model are shown as below.

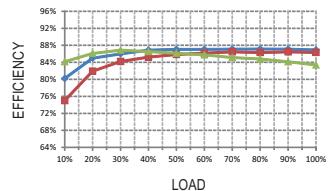
RSD-30G-3.3



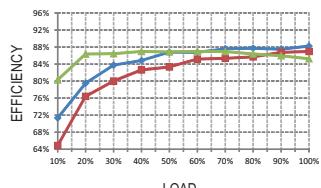
RSD-30G-5



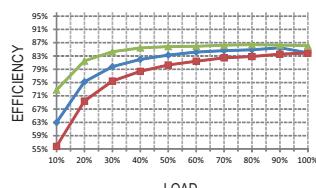
RSD-30G-12



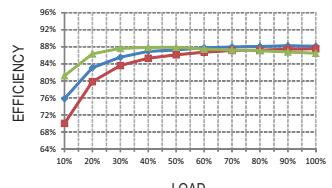
RSD-30G-24



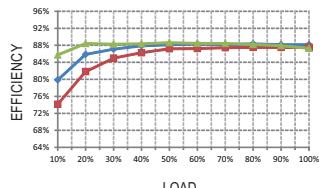
RSD-30L-3.3



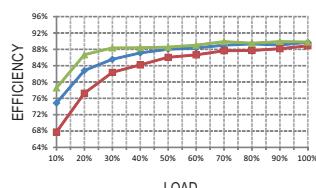
RSD-30L-5



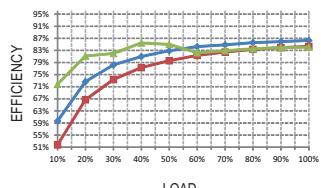
RSD-30L-12



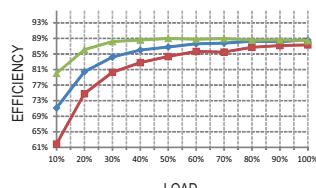
RSD-30L-24



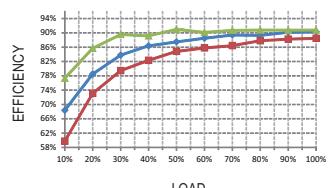
RSD-30H-5



RSD-30H-12



RSD-30H-24

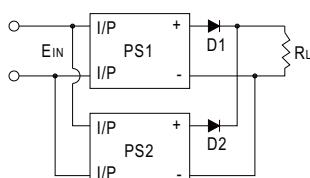


■ Parallel and Series Connection

A. Operation in Parallel

Since RSD-30 series don't have built-in parallel circuit, it can only use external circuits to achieve the redundant operation but not increase the current rating.

- Add a diode at the positive-output of each power supply (as shown as below), the current rating of the diode should be larger than the maximum output current rating and attached to a suitable heat sink. This is only for redundant use (increase the reliability of the system) and users have to check suitability of the circuit by themselves.

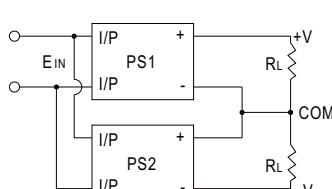


- When using S.P.S. in parallel connection, the leakage current will increase at the same time. This could pose as a shock hazard for the user. So please contact the supplier if you have this kind of application.

B. Operation in Series

RSD-30 can be operated in series. Here are the methods of doing it:

- Positive and negative terminals are connected as shown as below. According to the connection, you can get the positive and negative output voltages for your loads.

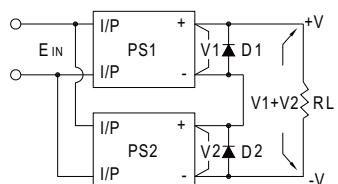




30W Reliable Railway DC-DC Converter

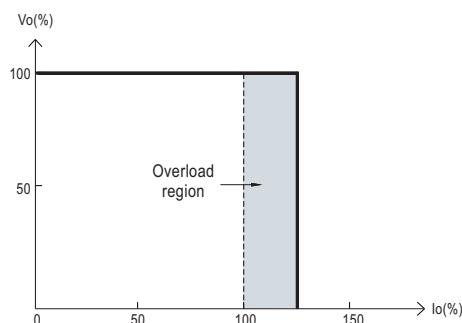
RSD-30 series

2.Increase the output voltage (current does not change). Because RSD-30 series have no reverse blocking diode in the unit, you should add an external blocking diode to prevent the damage of every unit while starting up. The voltage rating of the external diode should be larger than V_1+V_2 (as shown as below).



■ Overload Protection

If the output draw up to 105~135% of its output power rating, the converter will go into overload protection which is constant current mode. After the faulty condition is removed, it will recover automatically. Please refer to the diagram below for the detail operation characteristic. Please note that it's not suitable to operate within the overload region continuously, or it may cause to over temperature and reduce the life of the power supply unit or even damage it.



■ Over Voltage Protection

The converter shuts off to protect itself when the output voltage drawn exceeds 115~140% of its output rating. It must be repowered on to recover.

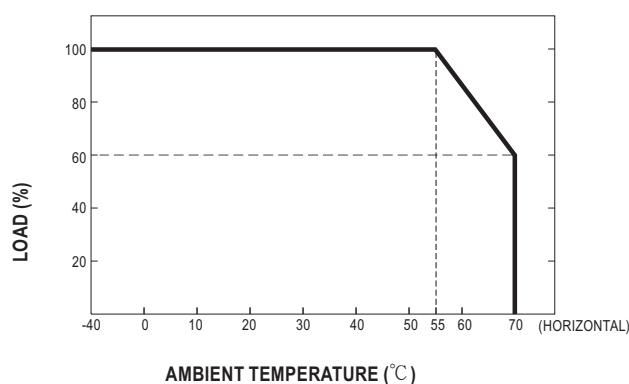
■ LED Indicator

Equipped with a built-in LED indicator, the converter provides an easy way for users to check its condition through the LED indicator.
Green : normal operation; No signal: no power or failure.

■ Derating Curve

a.Single unit operation

If the unit has no iron plate mounted on its bottom, the maximum ambient temperature for the unit will be 55°C as operating under full load condition. It requires de-rating output current when ambient temperature is between 55~70°C, please refer to the de-rating curve as below.

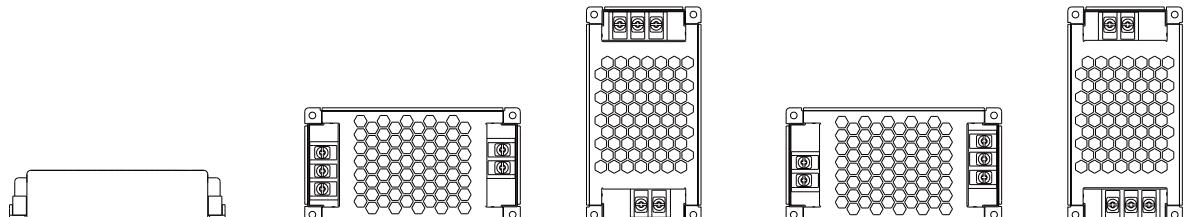




30W Reliable Railway DC-DC Converter

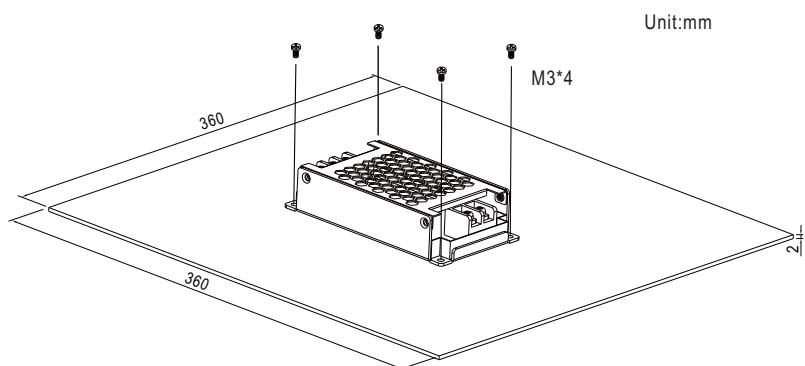
RSD-30 series

Suitable installation methods are shown as below. Since RSD-30 is a semi-potted model, its thermal performances for the following installation methods are similar and share the same derating curve.

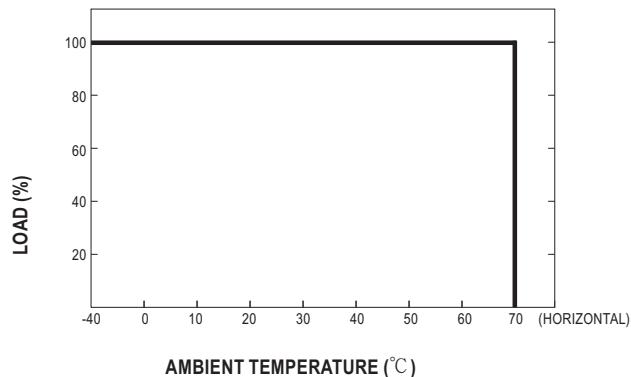


b. Operate with additional iron plate

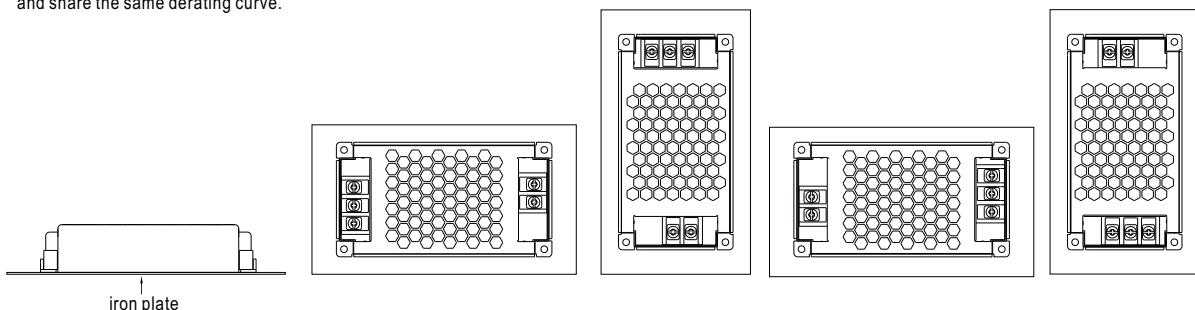
If it is necessary to fulfil the requirements of EN50155 TX level that operate the unit fully-loaded at 70°C, RSD-30 series must be installed onto an iron plate on the bottom. The size of the suggested iron plate is shown as below. In order for optimal thermal performance, the iron plate must have an even & smooth surface and RSD-30 series must be firmly mounted at the center of the iron plate.



The load vs ambient temperature curve is shown as below.



Suitable installation methods are shown as below. Since RSD-30 is a semi-potted model, its thermal performances for the following installation methods are similar and share the same derating curve.





30W Reliable Railway DC-DC Converter

RSD-30 series

■ Immunity to Environmental Conditions

Test method	Standard	Test conditions	Status
Cooling Test	EN 50155 section 12.2.3 (Column 2, Class TX) EN 60068-2-1	Temperature: -40°C Dwell Time: 2 hrs/cycle	No damage
Dry Heat Test	EN 50155 section 12.2.4 (Column 2, Class TX) EN 50155 section 12.2.4 (Column 3, Class TX & Column 4, Class TX) EN 60068-2-2	Temperature: 70°C / 85°C Duration: 6 hrs / 10min	PASS
Damp Heat Test, Cyclic	EN 50155 section 12.2.5 EN 60068-2-30	Temperature: 25°C ~ 55°C Humidity: 90% ~ 100% RH Duration: 48 hrs	PASS
Vibration Test	EN 50155 section 12.2.11 EN 61373	Temperature: 19°C Humidity: 65% Duration: 10 mins	PASS
Increased Vibration Test	EN 50155 section 12.2.11 EN 61373	Temperature: 19°C Humidity: 65% Duration: 5 hrs	PASS
Shock Test	EN 50155 section 12.2.11 EN 61373	Temperature: 21 ± 3°C Humidity: 65 ± 5% Duration: 30ms * 18	PASS
Low Temperature Storage Test	EN 50155 section 12.2.3 (Column 2, Class TX) EN 60068-2-1	Temperature: -40°C Dwell Time: 16 hrs	PASS
Salt Mist Test	EN 50155 section 12.2.10 (Class ST4)	Temperature: 35°C ± 2°C Duration: 96 hrs	PASS

■ EN45545-2 Fire Test Conditions

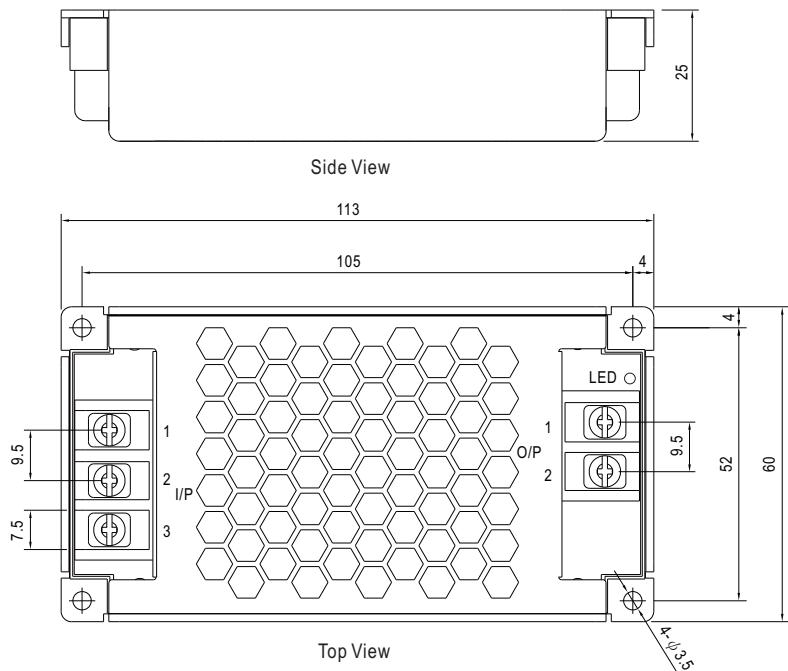
Test Items		Hazard Level			
Items		Standard	HL1	HL2	
R24	Oxygen index test	EN 45545-2:2013+A1:2015 EN ISO 4589-2:1996	PASS	PASS	PASS
R25	Glow-wire test	EN 45545-2:2013+A1:2015 EN 60695-2-11:2000	PASS	PASS	PASS
R26	Vertical flame test	EN 45545-2:2013+A1:2015 EN 60695-11:2003	PASS	PASS	PASS



30W Reliable Railway DC-DC Converter

RSD-30 series**■ Mechanical Specification**

Case No.253A Unit:mm



Input Terminal Pin No. Assignment :

Pin No.	Assignment
1	DC INPUT V+
2	DC INPUT V-
3	FG \pm

Output Terminal Pin No. Assignment :

Pin No.	Assignment
1	DC OUTPUT -V
2	DC OUTPUT +V

■ Installation ManualPlease refer to : <http://www.meanwell.com/manual.html>

File Name:RSD-30-SPEC 2017-02-22

1.2 Mean Well SD-350C-12



350W Single Output DC-DC Converter

SD-350 series



■ Features :

- 2:1 wide input range
- Protections: Short circuit / Overload / Over voltage / Over temperature
- 1500VAC I/O isolation
- Forced air cooling by built-in DC Fan
- 100% full load burn-in test
- 24V and 48V input voltage design refer to LVD
- 2 years warranty



EN62368-1

CB (for D type only)
IEC62368-1


TPTC004

SPECIFICATION

MODEL	SD-350B				SD-350C				
OUTPUT	DC VOLTAGE	5V	12V	24V	48V	5V	12V	24V	48V
	RATED CURRENT	57A	27.5A	14.6A	7.3A	60A	27.5A	14.6A	7.3A
	CURRENT RANGE	0 ~ 57A	0 ~ 27.5A	0 ~ 14.6A	0 ~ 7.3A	0 ~ 60A	0 ~ 27.5A	0 ~ 14.6A	0 ~ 7.3A
	RATED POWER	285W	330W	350.4W	350.4W	300W	330W	350.4W	350.4W
	ripple & noise (max.) Note.2	100mVp-p	120mVp-p	150mVp-p	200mVp-p	100mVp-p	120mVp-p	150mVp-p	200mVp-p
	VOLTAGE ADJ. RANGE	4.5 ~ 5.5VDC	11 ~ 16VDC	23 ~ 30VDC	43 ~ 53VDC	4.5 ~ 5.5VDC	11 ~ 16VDC	23 ~ 30VDC	43 ~ 53VDC
	VOLTAGE TOLERANCE Note.3	±2.0%	±1.0%	±1.0%	±1.0%	±2.0%	±1.0%	±1.0%	±1.0%
	LINE REGULATION	±0.5%	±0.3%	±0.2%	±0.2%	±0.5%	±0.3%	±0.2%	±0.2%
	LOAD REGULATION	±1.0%	±1.0%	±1.0%	±1.0%	±1.0%	±1.0%	±1.0%	±1.0%
INPUT	SETUP, RISE TIME	300ms, 50ms at full load							
	VOLTAGE RANGE	B:19 ~ 36VDC	C:36 ~ 72VDC	D:72 ~ 144VDC					
	EFFICIENCY (Typ.)	74%	80%	80%	84%	76%	81%	81%	82%
	DC CURRENT (Typ.)	14.4A/24V	16A/24V	17.6A/24V	17.6A/24V	7.6A/48V	8.8A/48V	9.0A/48V	9.0A/48V
	INRUSH CURRENT (Typ.)	C:45A/48VDC D:45A/96VDC							
	PROTECTION	OVERLOAD	105 ~ 135% rated output power Protection type : Shut down o/p voltage, re-power on to recover						
		OVER VOLTAGE	5.75 ~ 6.75V	16.8 ~ 20V	31.5 ~ 37.5V	53 ~ 65V	5.75 ~ 6.75V	16.8 ~ 20V	31.5 ~ 37.5V
	OVER TEMPERATURE Protection type : Shut down o/p voltage, re-power on to recover								
ENVIRONMENT	WORKING TEMP.	-20 ~ +60°C (Refer to "Derating Curve")							
	WORKING HUMIDITY	20 ~ 90% RH non-condensing							
	STORAGE TEMP., HUMIDITY	-40 ~ +85°C, 10 ~ 95% RH							
	TEMP. COEFFICIENT	±0.03%/°C (0 ~ 50°C)							
	VIBRATION	10 ~ 500Hz, 2G 10min./1cycle, 60min. each along X, Y, Z axes							
SAFETY & EMC (Note 4)	SAFETY STANDARDS	IEC62368-1 CB approved by TUV (for D type only), EAC TP TC 004 approved							
	WITHSTAND VOLTAGE	I/P-O/P:1.5KVAC I/P-FG:2KVAC O/P-FG:0.5KVAC							
	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:100M Ohms / 500VDC / 25°C/ 70% RH							
	EMC EMISSION	Compliance to EN55032 (CISPR32) Class B, EAC TP TC 020							
OTHERS	EMC IMMUNITY	Compliance to EN61000-4-2,3,4,6,8, light industry level, criteria A, EAC TP TC 020							
	MTBF	209.4 hrs min. MIL-HDBK-217F (25°C)							
	DIMENSION	215*115*50mm (L*W*H)							
NOTE	PACKING	1.1Kg; 12pcs/14.4Kg/0.92CUFT							
1. All parameters NOT specially mentioned are measured at 24,48,96VDC input, rated load and 25°C of ambient temperature. 2. Ripple & noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uf & 47uf parallel capacitor. 3. Tolerance : includes set up tolerance, line regulation and load regulation. 4. The power supply is considered a component which will be installed into a final equipment. All the EMC tests are been executed by mounting the unit on a 360mm*360mm metal plate with 1mm of thickness. The final equipment must be re-confirmed that it still meets EMC directives. For guidance on how to perform these EMC tests, please refer to "EMI testing of component power supplies." (as available on http://www.meanwell.com) 5. The ambient temperature derating of 3.5°C/1000m with fanless models and of 5°C/1000m with fan models for operating altitude higher than 2000m(6500ft)									

File Name:SD-350-SPEC 2020-03-37



350W Single Output DC-DC Converter

SD-350 series

■ Features :

- 2:1 wide input range
- Protections: Short circuit / Overload / Over voltage / Over temperature
- 1500VAC I/O isolation
- Forced air cooling by built-in DC Fan
- 100% full load burn-in test
- 24V(B) and 48V(C) input voltage design refer to LVD
- 2 years warranty

**CB** (for D type only)
IEC62368-1**SPECIFICATION**

MODEL	SD-350D			
OUTPUT	DC VOLTAGE	5V	12V	24V
	RATED CURRENT	60A	29.2A	14.6A
	CURRENT RANGE	0 ~ 60A	0 ~ 29.2A	0 ~ 14.6A
	RATED POWER	300W	350.4W	350.4W
	RIPPLE & NOISE (max.) Note.2	100mVp-p	120mVp-p	150mVp-p
	VOLTAGE ADJ. RANGE	4.5 ~ 5.5VDC	11 ~ 16VDC	23 ~ 30VDC
	VOLTAGE TOLERANCE Note.3	±2.0%	±1.0%	±1.0%
	LINE REGULATION	±0.5%	±0.3%	±0.2%
	LOAD REGULATION	±1.0%	±1.0%	±1.0%
	SETUP, RISE TIME	300ms, 50ms at full load		
INPUT	VOLTAGE RANGE	B:19 ~ 36VDC	C:36 ~ 72VDC	D:72 ~ 144VDC
	EFFICIENCY (Typ.)	78%	83%	87%
	DC CURRENT (Typ.)	6A/96V	6A/96V	6A/96V
	INRUSH CURRENT (Typ.)	C:45A/48VDC	D:45A/96VDC	
PROTECTION	OVERLOAD	105 ~ 135% rated output power		
		Protection type : Shut down o/p voltage, re-power on to recover		
	OVER VOLTAGE	5.75 ~ 6.75V	16.8 ~ 20V	31.5 ~ 37.5V
		Protection type : Shut down o/p voltage, re-power on to recover		
ENVIRONMENT	OVER TEMPERATURE	Shut down o/p voltage, recovers automatically after temperature goes down		
	WORKING TEMP.	-20 ~ +60°C (Refer to "Derating Curve")		
	WORKING HUMIDITY	20 ~ 90% RH non-condensing		
	STORAGE TEMP., HUMIDITY	-40 ~ +85°C, 10 ~ 95% RH		
	TEMP. COEFFICIENT	±0.03%/°C (0 ~ 50°C)		
SAFETY & EMC (Note 4)	VIBRATION	10 ~ 500Hz, 2G 10min./cycle, 60min. each along X, Y, Z axes		
	SAFETY STANDARDS	IEC62368-1 CB approved by TUV (for D type only), EAC TP TC 004 approved		
	WITHSTAND VOLTAGE	I/P-O/P:1.5KVAC	I/P-FG:2KVAC	O/P-FG:0.5KVAC
	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:100M Ohms / 500VDC / 25°C/ 70% RH		
	EMC EMISSION	Compliance to EN55022 (CISPR22) Class B, EAC TP TC 020		
OTHERS	EMC IMMUNITY	Compliance to EN61000-4-2,3,4,6,8, light industry level, criteria A, EAC TP TC 020		
	MTBF	209.4K hrs min. MIL-HDBK-217F (25°C)		
	DIMENSION	215*115*50mm (L*W*H)		
NOTE	PACKING	1.1Kg; 12pcs/14.4Kg/0.92CUFT		
	1. All parameters NOT specially mentioned are measured at 24,48,96VDC input, rated load and 25°C of ambient temperature. 2. Ripple & noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uf & 47uf parallel capacitor. 3. Tolerance : includes set up tolerance, line regulation and load regulation. 4. The power supply is considered a component which will be installed into a final equipment. All the EMC tests are been executed by mounting the unit on a 360mm*360mm metal plate with 1mm of thickness. The final equipment must be re-confirmed that it still meets EMC directives. For guidance on how to perform these EMC tests, please refer to "EMI testing of component power supplies." (as available on http://www.meanwell.com) 5. The ambient temperature derating of 3.5°C/1000m with fanless models and of 5°C/1000m with fan models for operating altitude higher than 2000m(6500ft).			

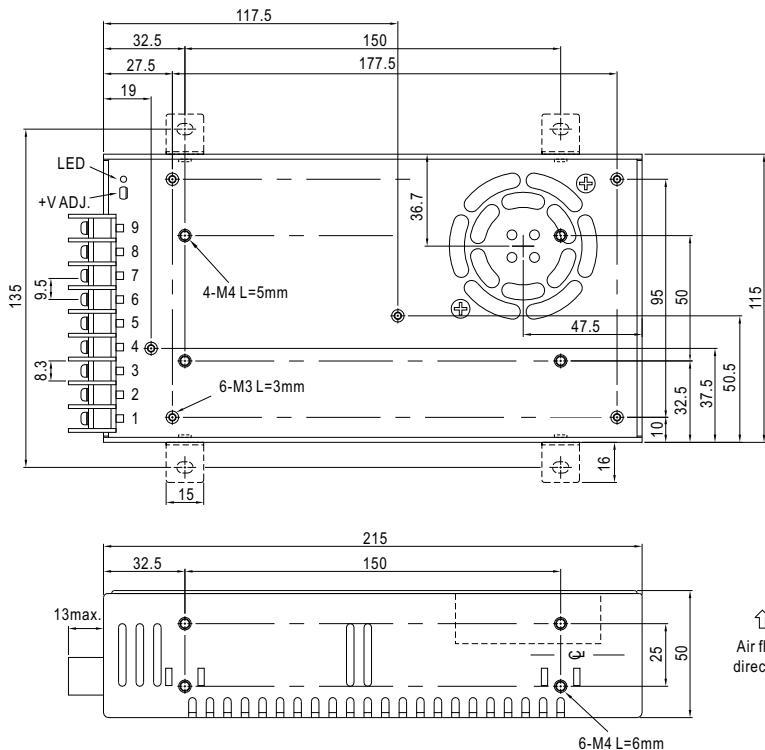


350W Single Output DC-DC Converter

SD-350 series

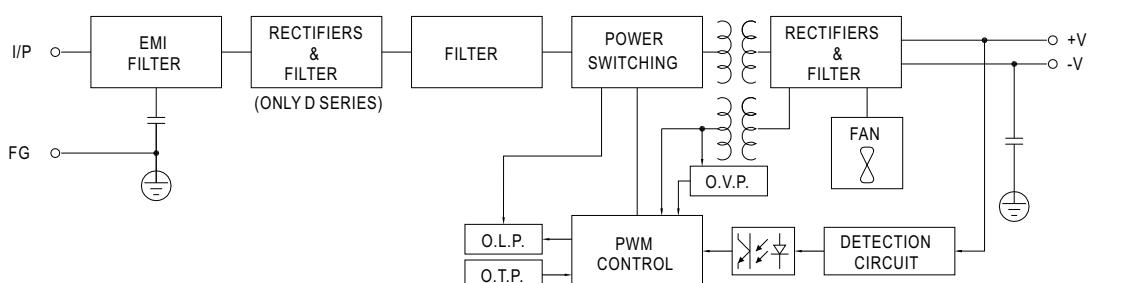
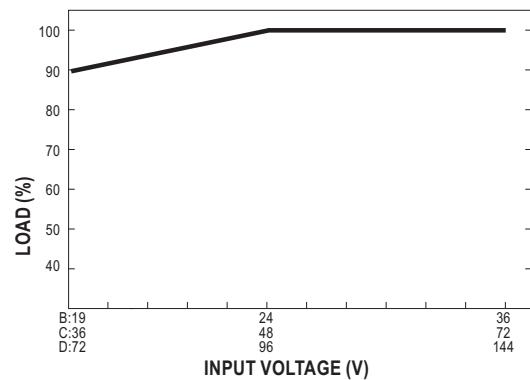
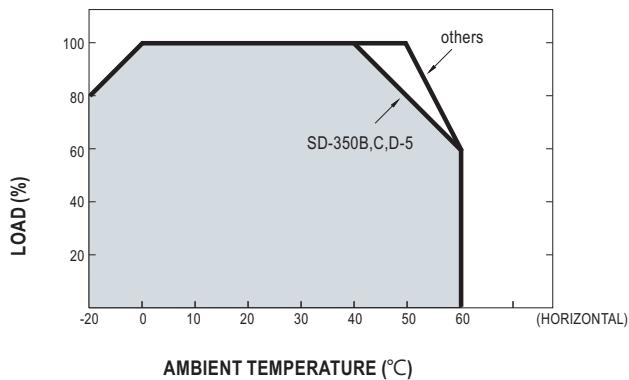
Mechanical Specification

Case No. 912B Unit:mm



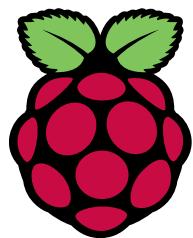
Terminal Pin No. Assignment:

Pin No.	Assignment	Pin No.	Assignment
1	DC INPUT V+	4,5,6	DC OUTPUT V-
2	DC INPUT V-	7,8,9	DC OUTPUT V+
3	FG \pm		

Block Diagram**Derating Curve****Static Characteristics**

1.3 Raspberry Pi 4 Model B

DATASHEET



Raspberry Pi 4 Model B

Release 1

June 2019

Copyright 2019 Raspberry Pi (Trading) Ltd. All rights reserved.



Raspberry Pi 4 Model B Datasheet
Copyright Raspberry Pi (Trading) Ltd. 2019

Table 1: Release History

Release	Date	Description
1	21/06/2019	First release

The latest release of this document can be found at <https://www.raspberrypi.org>



Contents

1	Introduction	5
2	Features	6
2.1	Hardware	6
2.2	Interfaces	6
2.3	Software	7
3	Mechanical Specification	7
4	Electrical Specification	7
4.1	Power Requirements	9
5	Peripherals	9
5.1	GPIO Interface	9
5.1.1	GPIO Pin Assignments	9
5.1.2	GPIO Alternate Functions	10
5.1.3	Display Parallel Interface (DPI)	11
5.1.4	SD/SDIO Interface	11
5.2	Camera and Display Interfaces	11
5.3	USB	11
5.4	HDMI	11
5.5	Audio and Composite (TV Out)	11
5.6	Temperature Range and Thermals	11
6	Availability	12
7	Support	12



Raspberry Pi 4 Model B Datasheet
Copyright Raspberry Pi (Trading) Ltd. 2019

List of Figures

1	Mechanical Dimensions	7
2	Digital IO Characteristics	8
3	GPIO Connector Pinout	9



List of Tables

1	Release History	1
2	Absolute Maximum Ratings	8
3	DC Characteristics	8
4	Digital I/O Pin AC Characteristics	8
5	Raspberry Pi 4 GPIO Alternate Functions	10



Raspberry Pi 4 Model B Datasheet
Copyright Raspberry Pi (Trading) Ltd. 2019

1 Introduction

The Raspberry Pi 4 Model B (Pi4B) is the first of a new generation of Raspberry Pi computers supporting more RAM and with significantly enhanced CPU, GPU and I/O performance; all within a similar form factor, power envelope and cost as the previous generation Raspberry Pi 3B+.

The Pi4B is available with either 1, 2 and 4 Gigabytes of LPDDR4 SDRAM.



2 Features

2.1 Hardware

- Quad core 64-bit ARM-Cortex A72 running at 1.5GHz
- 1, 2 and 4 Gigabyte LPDDR4 RAM options
- H.265 (HEVC) hardware decode (up to 4Kp60)
- H.264 hardware decode (up to 1080p60)
- VideoCore VI 3D Graphics
- Supports dual HDMI display output up to 4Kp60

2.2 Interfaces

- 802.11 b/g/n/ac Wireless LAN
- Bluetooth 5.0 with BLE
- 1x SD Card
- 2x micro-HDMI ports supporting dual displays up to 4Kp60 resolution
- 2x USB2 ports
- 2x USB3 ports
- 1x Gigabit Ethernet port (supports PoE with add-on PoE HAT)
- 1x Raspberry Pi camera port (2-lane MIPI CSI)
- 1x Raspberry Pi display port (2-lane MIPI DSI)
- 28x user GPIO supporting various interface options:
 - Up to 6x UART
 - Up to 6x I2C
 - Up to 5x SPI
 - 1x SDIO interface
 - 1x DPI (Parallel RGB Display)
 - 1x PCM
 - Up to 2x PWM channels
 - Up to 3x GPCLK outputs



2.3 Software

- ARMv8 Instruction Set
 - Mature Linux software stack
 - Actively developed and maintained
 - Recent Linux kernel support
 - Many drivers upstreamed
 - Stable and well supported userland
 - Availability of GPU functions using standard APIs

3 Mechanical Specification

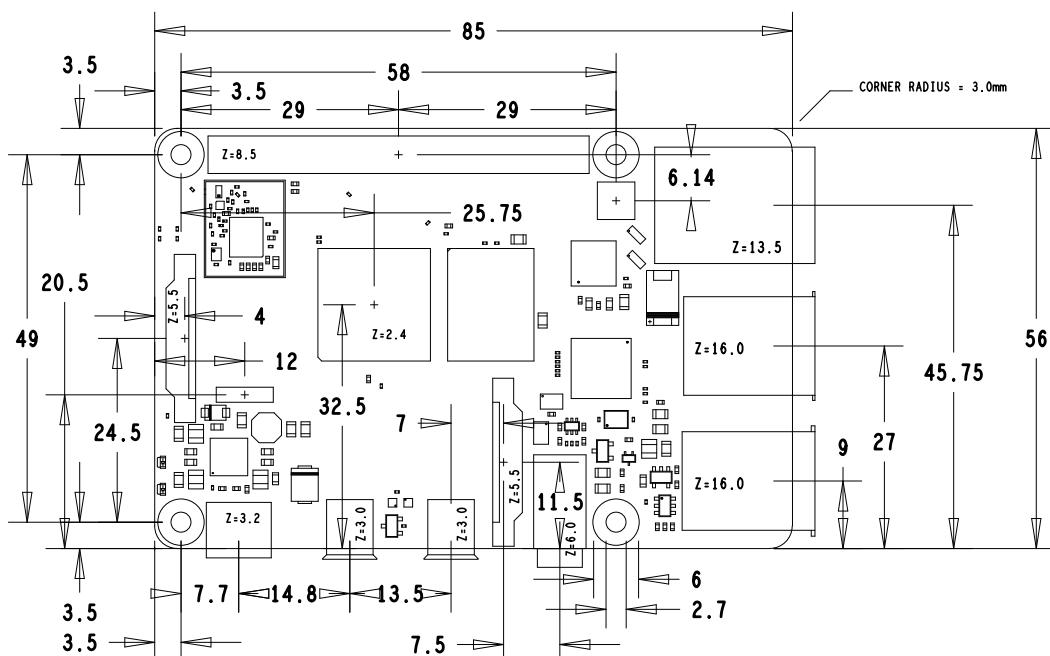


Figure 1: Mechanical Dimensions

4 Electrical Specification

Caution! Stresses above those listed in Table 2 may cause permanent damage to the device. This is a stress rating only; functional operation of the device under these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.


 Raspberry Pi 4 Model B Datasheet
 Copyright Raspberry Pi (Trading) Ltd. 2019

Symbol	Parameter	Minimum	Maximum	Unit
VIN	5V Input Voltage	-0.5	6.0	V

Table 2: Absolute Maximum Ratings

Please note that VDD_IO is the GPIO bank voltage which is tied to the on-board 3.3V supply rail.

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{IL}	Input low voltage ^a	$VDD_IO = 3.3V$	-	-	TBD	V
V_{IH}	Input high voltage ^a	$VDD_IO = 3.3V$	TBD	-	-	V
I_{IL}	Input leakage current	$TA = +85^\circ C$	-	-	TBD	μA
C_{IN}	Input capacitance	-	-	TBD	-	pF
V_{OL}	Output low voltage ^b	$VDD_IO = 3.3V, IOL = -2mA$	-	-	TBD	V
V_{OH}	Output high voltage ^b	$VDD_IO = 3.3V, IOH = 2mA$	TBD	-	-	V
I_{OL}	Output low current ^c	$VDD_IO = 3.3V, VO = 0.4V$	TBD	-	-	mA
I_{OH}	Output high current ^c	$VDD_IO = 3.3V, VO = 2.3V$	TBD	-	-	mA
R_{PU}	Pullup resistor	-	TBD	-	TBD	k Ω
R_{PD}	Pulldown resistor	-	TBD	-	TBD	k Ω

^a Hysteresis enabled

^b Default drive strength (8mA)

^c Maximum drive strength (16mA)

Table 3: DC Characteristics

Pin Name	Symbol	Parameter	Minimum	Typical	Maximum	Unit
Digital outputs	t_{rise}	10-90% rise time ^a	-	TBD	-	ns
Digital outputs	t_{fall}	90-10% fall time ^a	-	TBD	-	ns

^a Default drive strength, $CL = 5pF, VDD_IO = 3.3V$

Table 4: Digital I/O Pin AC Characteristics



Figure 2: Digital IO Characteristics



4.1 Power Requirements

The Pi4B requires a good quality USB-C power supply capable of delivering 5V at 3A. If attached downstream USB devices consume less than 500mA, a 5V, 2.5A supply may be used.

5 Peripherals

5.1 GPIO Interface

The Pi4B makes 28 BCM2711 GPIOs available via a standard Raspberry Pi 40-pin header. This header is backwards compatible with all previous Raspberry Pi boards with a 40-way header.

5.1.1 GPIO Pin Assignments

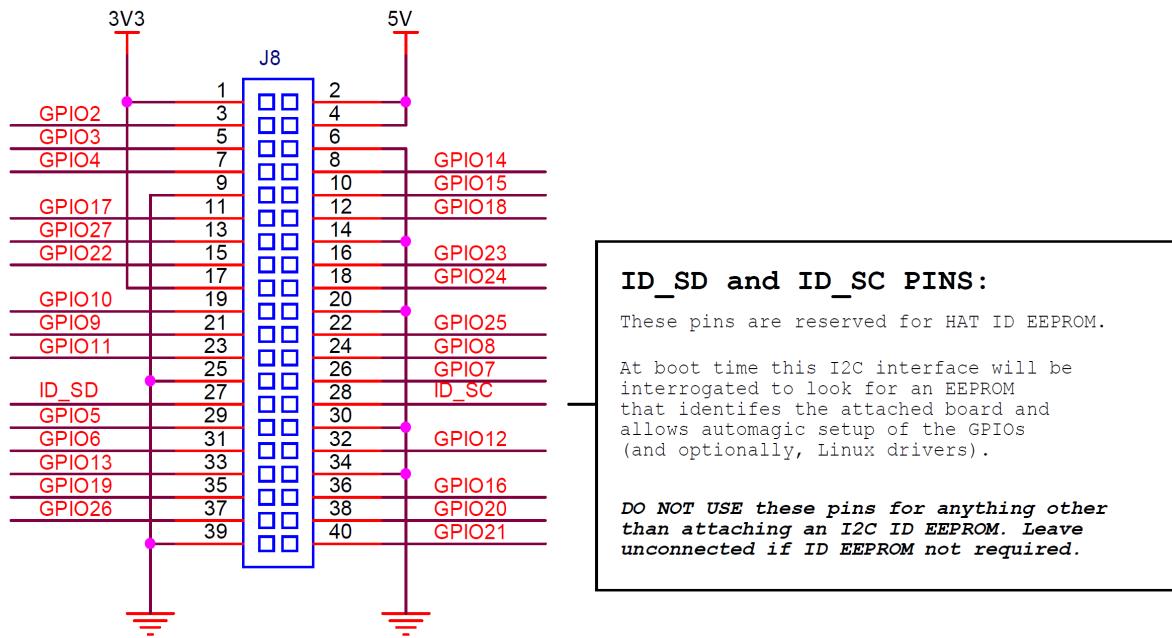


Figure 3: GPIO Connector Pinout

As well as being able to be used as straightforward software controlled input and output (with programmable pulls), GPIO pins can be switched (multiplexed) into various other modes backed by dedicated peripheral blocks such as I2C, UART and SPI.

In addition to the standard peripheral options found on legacy Pis, extra I2C, UART and SPI peripherals have been added to the BCM2711 chip and are available as further mux options on the Pi4. This gives users much more flexibility when attaching add-on hardware as compared to older models.



5.1.2 GPIO Alternate Functions

GPIO	Pull	Default					
		ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
0	High	SDA0	SA5	PCLK	SPI3_CE0_N	TXD2	SDA6
1	High	SCL0	SA4	DE	SPI3_MISO	RXD2	SCL6
2	High	SDA1	SA3	LCD_VSYNC	SPI3_MOSI	CTS2	SDA3
3	High	SCL1	SA2	LCD_HSYNC	SPI3_SCLK	RTS2	SCL3
4	High	GPCLK0	SA1	DPI_D0	SPI4_CE0_N	TXD3	SDA3
5	High	GPCLK1	SA0	DPI_D1	SPI4_MISO	RXD3	SCL3
6	High	GPCLK2	SOE_N	DPI_D2	SPI4_MOSI	CTS3	SDA4
7	High	SPI0_CE1_N	SWE_N	DPI_D3	SPI4_SCLK	RTS3	SCL4
8	High	SPI0_CE0_N	SD0	DPI_D4	-	TXD4	SDA4
9	Low	SPI0_MISO	SD1	DPI_D5	-	RXD4	SCL4
10	Low	SPI0_MOSI	SD2	DPI_D6	-	CTS4	SDA5
11	Low	SPI0_SCLK	SD3	DPI_D7	-	RTS4	SCL5
12	Low	PWM0	SD4	DPI_D8	SPI5_CE0_N	TXD5	SDA5
13	Low	PWM1	SD5	DPI_D9	SPI5_MISO	RXD5	SCL5
14	Low	TXD0	SD6	DPI_D10	SPI5_MOSI	CTS5	TXD1
15	Low	RXD0	SD7	DPI_D11	SPI5_SCLK	RTS5	RXD1
16	Low	FL0	SD8	DPI_D12	CTS0	SPI1_CE2_N	CTS1
17	Low	FL1	SD9	DPI_D13	RTS0	SPI1_CE1_N	RTS1
18	Low	PCM_CLK	SD10	DPI_D14	SPI6_CE0_N	SPI1_CE0_N	PWM0
19	Low	PCM_FS	SD11	DPI_D15	SPI6_MISO	SPI1_MISO	PWM1
20	Low	PCM_DIN	SD12	DPI_D16	SPI6_MOSI	SPI1_MOSI	GPCLK0
21	Low	PCM_DOUT	SD13	DPI_D17	SPI6_SCLK	SPI1_SCLK	GPCLK1
22	Low	SD0_CLK	SD14	DPI_D18	SD1_CLK	ARM_TRST	SDA6
23	Low	SD0_CMD	SD15	DPI_D19	SD1_CMD	ARM_RTCK	SCL6
24	Low	SD0_DAT0	SD16	DPI_D20	SD1_DAT0	ARM_TDO	SPI3_CE1_N
25	Low	SD0_DAT1	SD17	DPI_D21	SD1_DAT1	ARM_TCK	SPI4_CE1_N
26	Low	SD0_DAT2	TE0	DPI_D22	SD1_DAT2	ARM_TDI	SPI5_CE1_N
27	Low	SD0_DAT3	TE1	DPI_D23	SD1_DAT3	ARM_TMS	SPI6_CE1_N

Table 5: Raspberry Pi 4 GPIO Alternate Functions

Table 5 details the default pin pull state and available alternate GPIO functions. Most of these alternate peripheral functions are described in detail in the BCM2711 Peripherals Specification document which can be downloaded from the hardware documentation section of the website.



5.1.3 Display Parallel Interface (DPI)

A standard parallel RGB (DPI) interface is available the GPIOs. This up-to-24-bit parallel interface can support a secondary display.

5.1.4 SD/SDIO Interface

The Pi4B has a dedicated SD card socket which supports 1.8V, DDR50 mode (at a peak bandwidth of 50 Megabytes / sec). In addition, a legacy SDIO interface is available on the GPIO pins.

5.2 Camera and Display Interfaces

The Pi4B has 1x Raspberry Pi 2-lane MIPI CSI Camera and 1x Raspberry Pi 2-lane MIPI DSI Display connector. These connectors are backwards compatible with legacy Raspberry Pi boards, and support all of the available Raspberry Pi camera and display peripherals.

5.3 USB

The Pi4B has 2x USB2 and 2x USB3 type-A sockets. Downstream USB current is limited to approximately 1.1A in aggregate over the four sockets.

5.4 HDMI

The Pi4B has 2x micro-HDMI ports, both of which support CEC and HDMI 2.0 with resolutions up to 4Kp60.

5.5 Audio and Composite (TV Out)

The Pi4B supports near-CD-quality analogue audio output and composite TV-output via a 4-ring TRS 'A/V' jack.

The analog audio output can drive 32 Ohm headphones directly.

5.6 Temperature Range and Thermals

The recommended ambient operating temperature range is 0 to 50 degrees Celcius.

To reduce thermal output when idling or under light load, the Pi4B reduces the CPU clock speed and voltage. During heavier load the speed and voltage (and hence thermal output) are increased. The internal governor will throttle back both the CPU speed and voltage to make sure the CPU temperature never exceeds 85 degrees C.

The Pi4B will operate perfectly well without any extra cooling and is designed for sprint performance - expecting a light use case on average and ramping up the CPU speed when needed (e.g. when loading a webpage). If a user wishes to load the system continually or operate it at a high temperature at full performance, further cooling may be needed.



Raspberry Pi 4 Model B Datasheet
Copyright Raspberry Pi (Trading) Ltd. 2019

6 Availability

Raspberry Pi guarantee availability Pi4B until at least January 2026.

7 Support

For support please see the hardware documentation section of the Raspberry Pi website and post questions to the Raspberry Pi forum.

2 Antrieb

2.1 Ashwoods Elektro-Motor IPM-200-50



Smaller, lighter, more efficient

Ashwoods design and manufacture smaller, lighter and more efficient permanent magnet electric motors across a range of sectors including the automotive, materials handling and off-highway vehicle markets.

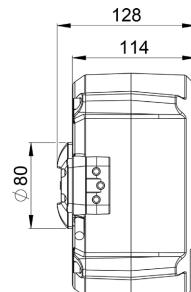
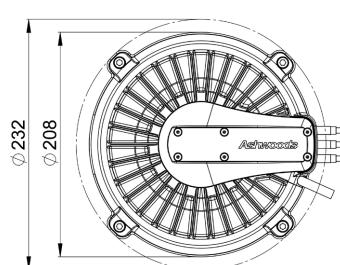
IPM-200-50

Functional Data

Peak Torque @ 350Arms	74 Nm
Voltage	48 - 80V
Peak Power	10kW - 16kW
Continuous Power	up to 8kW
Max Speed	8500 rpm
Peak Efficiency	94%
Weight (motor)	13kg



Applications



Interfaces

Standard



A standard 24MM shaft with a keyway can be provided to allow easy integration in many applications.

Traction



Ashwoods works closely with a variety of transmission suppliers to provide interfaces for any application.

Hydraulic



Pump interfaces have been developed to enable our motors to work with a range of pumps.

Motor Features



Sales and Customer Service

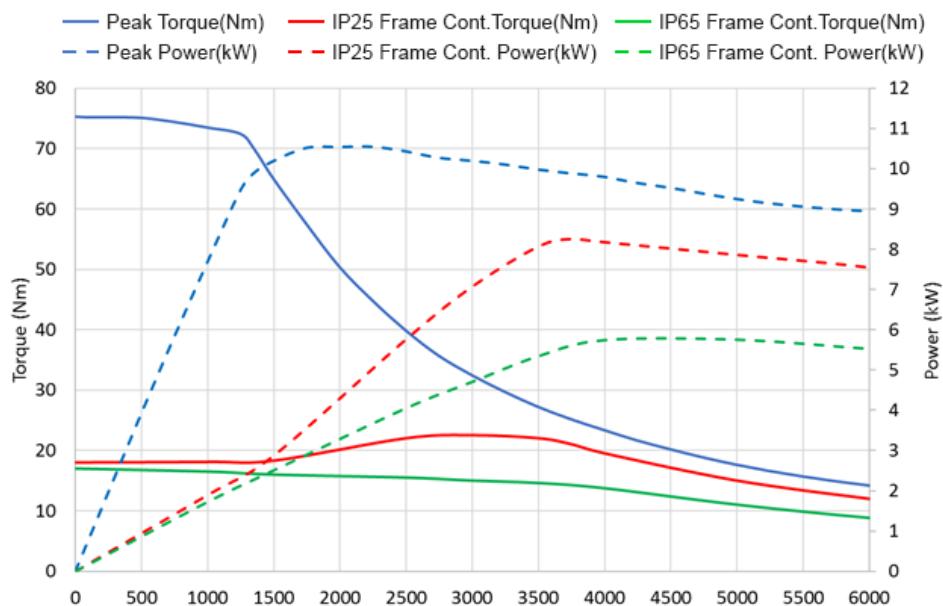
Telephone: +44 1392 340184
(Monday to Friday, 8.30am-5.30pm)

info@ashwoodselectricmotors.com
www.ashwoodselectricmotors.com

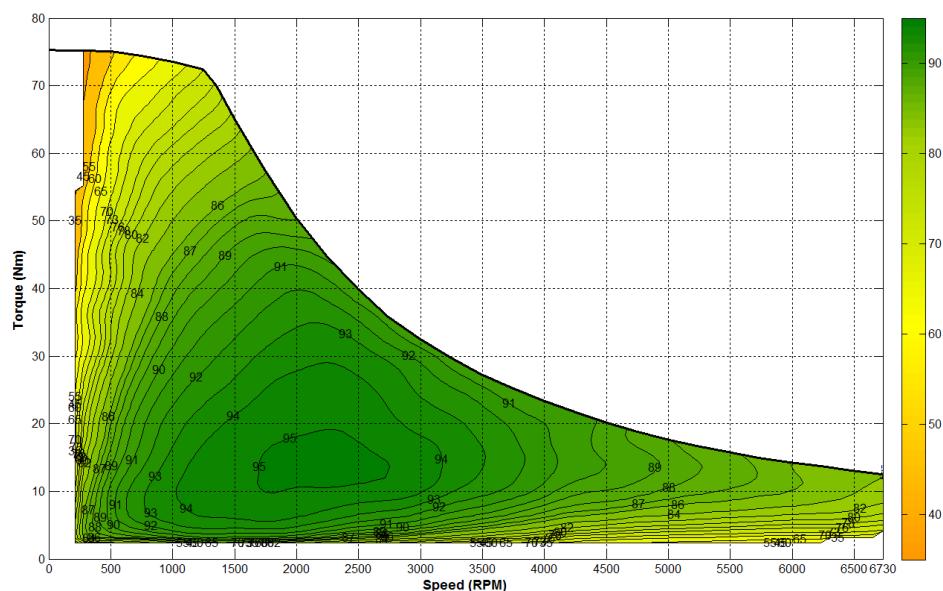
IPM-200-50

48V Performance Graphs

IPM-200-50 Perfomance Curves @48V 350Arms



Ashwoods IPM Motor Efficiency



Sales and Customer Service

Telephone:+44 1392 340184
(Monday to Friday, 8.30am-5.30pm)

info@ashwoodselectricmotors.com
www.ashwoodselectricmotors.com

2.2 Hochleistungs-Relais: KILOVAC LEV200 A4ANA



KILOVAC High Voltage DC Contactors

KILOVAC LEV200 Series Contactor With 1 Form X Contacts Rated 500+ Amps, 12-900Vdc

Product Facts

- Designed to be the lowest cost sealed contactor in the industry with its current rating (500+A carry, 2000A interrupt at 320Vdc)
- Available with bottom or side mounting — not position sensitive
- Optional auxiliary contact for easy monitoring of power contact position
- Hermetically sealed — intrinsically safe, operates in explosive/harsh environments with no oxidation or contamination of coils or contacts, including long periods of non-operation
- Typical applications include battery switching and backup, DC voltage power control, circuit protection and safety
- Versatile coil/power connections
- Designed and built in accordance to AIAG QS9000
- RoHS compliant



Coil Data (Valid Over Temperature Range) 4

Nominal Voltage	12Vdc	24Vdc	48Vdc	72Vdc
Pickup Voltage (Will Operate)	9.0Vdc	19.0Vdc	38.0Vdc	57.0Vdc
Voltage (Max.)	15Vdc	30Vdc	60Vdc	90Vdc
Dropout Voltage	0.75 - 2.0Vdc	1.0 - 5.0Vdc	2.0 - 7.0Vdc	3.0 - 12.0Vdc
Coil Resistance @ 25° (Typ.)	11 ohms	40 ohms	145 ohms	357 ohms

Ordering Information

Typical Part Number ►

LEV200 A 4 N A A

Series: _____

LEV200 = 500+ Amp, 12-900Vdc Contactor

Contact Form: _____

A = Normally Open

H = Normally Open with Aux. Contacts. (Option "H" requires option "A" in Coil Wire Length and option "N" in Coil Terminal Connector.)

Note: Other auxiliary contact forms available. Consult factory.

Coil Voltage: _____

4 = 12Vdc 5 = 24Vdc B = 28Vdc

6 = 48Vdc K = 72Vdc

8 = 96Vdc L = 110Vdc O = 115Vac 9 = 240Vac

Notes: Consult factory for detailed specifications and availability of coils not listed in "Coil Data" table above. In coil voltage codes, 115Vac is designated by the letter "O" rather than the numeral "0."

Coil Wire Length: _____

A = 15.3 in (390 mm) N = None (Requires option "A" in next step.)

Coil Terminal Connector: _____

N = None, stripped wires

(Requires option "A" in previous step.)

A = Studs, #10-32 Threaded (Electrical connection is made to the tab at the base of the stud.)

Note: Specify option A, stripped wires, for coil voltages > 96Vdc

Mounting & Power Terminals: _____

A = Bottom Mount & Male 10mm x M8 Threaded Terminals

F = Side Mount & Male 10mm x M8 Threaded Terminals

Consult factory regarding other available mountings and power terminals.

For factory-direct application assistance,
dial 800-253-4560, ext. 2055, or
805-220-2055.

7-32

Catalog 5-1773450-5
Revised 3-13
www.te.com

Dimensions are shown for reference purposes only.
Specifications subject to change.

Dimensions are in millimeters unless otherwise specified.

USA: +1 800 522 6752
Asia Pacific: +86 0 400 820 6015
UK: +44 800 267 666

For additional support numbers
please visit www.te.com

Performance Data

Contact Arrangement, Power Contacts — 1 Form X (SPST-NO-DM)

Rated Operating Voltage — 12 - 900 VDC

Continuous (Carry) Current, Typical — 500 A @ 65°C, 400 mcm conductors
Consult TE for required conductors for higher (500+ A) currents

Make/Break Current at Various Voltages 1 — See graph next page

Break Current at 320VDC 1 — 2,000 A, 1 cycle 3

Contact Resistance, Typ. (@200A) — 0.2 mohms

Load Life — See graph next page

Mechanical Life — 1 million cycles

Contact Arrangement, Auxiliary Contacts — 1 Form A (SPST-NO)

Aux. Contact Current, Max. — 2A @ 30VDC / 3A @ 125VAC

Aux. Contact Current, Min. — 100mA @ 8V

Aux. Contact Resistance, Max. — 0.417 ohms @ 30VDC / .150 ohms @ 125VAC

Operate Time @ 25°C — Close (includes bounce), Typ. — 25 ms
Bounce (after close only), Max. — 7 ms
Release (includes arcing), Max @ 2000A — 12 ms

Dielectric Withstanding Voltage — 2,200 VRMS @ sea level (leakage <1mA)

Insulation Resistance @ 500VDC — 100 megohms 2

Shock, 11ms 1/2 Sine, Peak, Operating — 20 G

Vibration, Sine, 80-2000Hz., Peak — 20 G

Operating Ambient Temperature — -40°C to +85°C

Weight, Typical — 1.3 lb. (.60 kg)

Notes:

1 Main power contacts

2 50 at end of life

3 Does not meet dielectric & IR after test, 1700 amp for unit with Aux. Contacts

4 Contacts will operate with $0.8V_{nom} < V_{coil} < 1.1V_{nom}$ over temperature range.

Invalid Combinations/Reason

LEV200H-NA_

No auxiliary function with coil studs

LEV200_ONA_

No coil studs with rectifier circuit

LEV200_9NA_

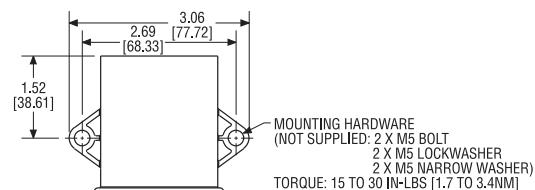
No coil studs with rectifier circuit

LEV200_O_F

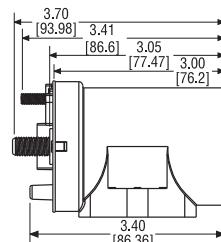
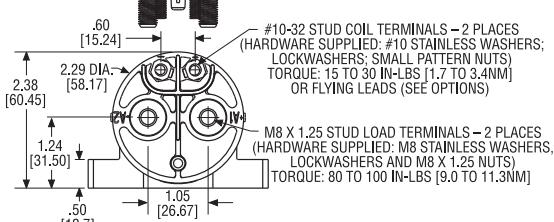
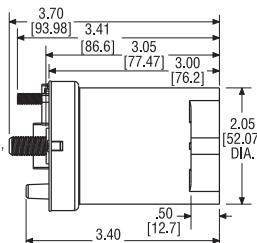
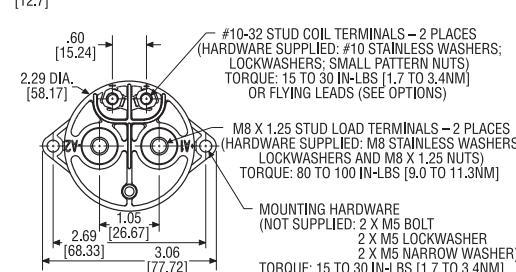
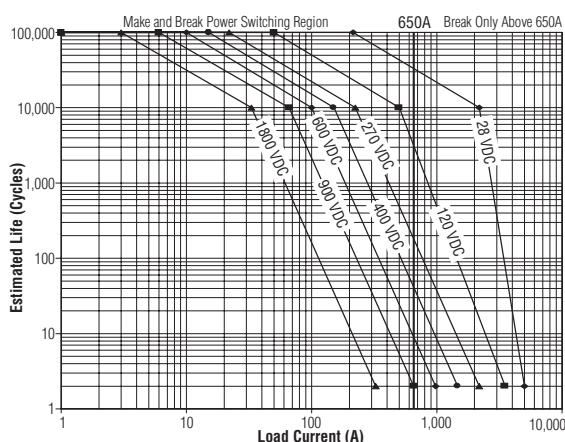
No side mont with rectifier circuit

LEV200_9_F

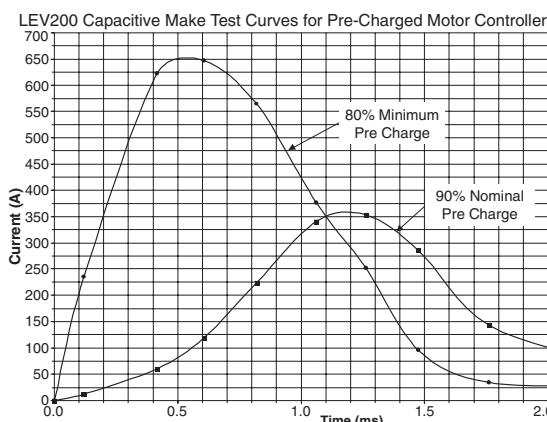
No side mount with rectifier circuit


KILOVAC High Voltage DC Contactors
KILOVAC LEV200 Series (Continued)
Outline Dimensions


Note: When stud terminals are specified for coil connections, the electrical connection is made at the tab located at the base of the stud.

Side Mount Enclosure

Bottom Mount Enclosure

Estimated Make & Break Power Switching Ratings

NOTES:

- 1) For resistive loads with 300µH maximum inductance. Consult factory for inductive loads.
- 2) Estimates based on extrapolated data. User is encouraged to confirm performance in application.
- 3) End of life when dielectric strength between terminals falls below 50 megohms @ 500VDC.
- 4) The maximum make current is 650A to avoid contact welding.



7
KILOVAC High Voltage
DC Contactors

For factory-direct application assistance,
dial 800-253-4560, ext. 2055, or
805-220-2055.

7-33

Catalog 5-1773450-5
Revised 3-13

www.te.com

Dimensions are shown for
reference purposes only.
Specifications subject
to change.

Dimensions are in millimeters
unless otherwise specified.

USA: +1 800 522 6752
Asia Pacific: +86 0 400 820 6015
UK: +44 800 267 666

For additional support numbers
please visit www.te.com

2.3 Hochgeschwindigkeits-Schmelzsicherung: FWA-400B

High Speed Fuses

Bussmann
by EATON

North American — FWA 150V: 70-1000A

FWA

Specifications

Description: North American style stud-mount fuses.

Dimensions: See Dimensions illustrations.

Ratings:

Volts: — 150Vac/dc*

Amps: — 70-1000A

IR: — 100kA Sym. (70-400A)

— 200kA Sym. (450-1000A)

— 20kA @150Vdc (70-800A)

— 100kA @ 80Vdc (70-1000A)

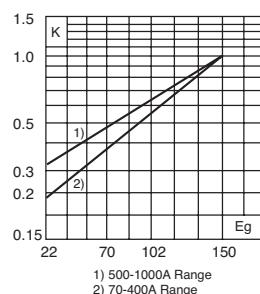
*1000A rated @ 80Vdc.

Agency Information: CE, UL Recognized JFHR2.E91958

Electrical Characteristics

Total Clearing I^2t

The total clearing I^2t at rated voltage and at power factor of 15% are given in the electrical characteristics. For other voltages, the clearing I^2t is found by multiplying by correction factor, K, given as a function of applied working voltage, E_g , (rms).



Dimensions - in

Fig. 1: 70-400A

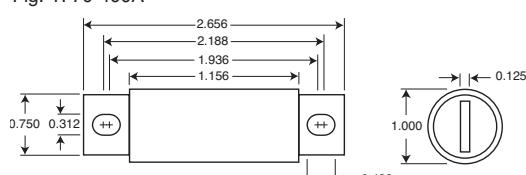
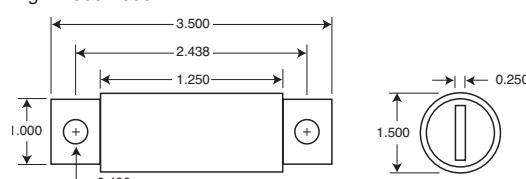


Fig. 2: 500-1000A

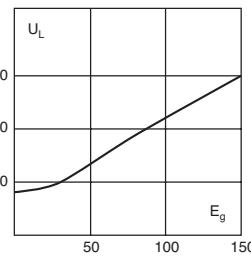


1mm = 0.0394" / 1" = 25.4mm

Data Sheet: 720002

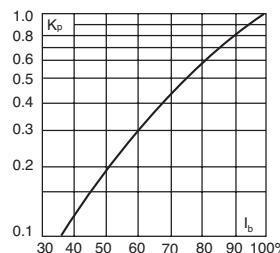
Arc Voltage

This curve gives the peak arc voltage, U_L , which may appear across the fuse during its operation as a function of the applied working voltage, E_g , (rms) at a power factor of 15%.



Power Losses

Watts loss at rated current is given in the electrical characteristics. The curve allows the calculation of the power losses at load currents lower than the rated current. The correction factor, K_p , is given as a function of the RMS load current, I_b , in % of the rated current.



Catalog Numbers

Catalog Number	Rated Current RMS-Amps	Electrical Characteristics		
		Pre-arc	I^2t (A ² Sec)	Clearing at 150V
FWA-70B	70	470	4000	6.9
FWA-80B	80	670	6000	7.7
FWA-100B	100	1200	12000	9.0
FWA-125B	125	1870	18000	11.2
FWA-150B	150	2700	26000	13.5
FWA-200B	200	4780	45000	17.6
FWA-250B	250	7470	70000	22.5
FWA-300B	300	10760	100000	27.0
FWA-350B	350	15700	140000	30.6
FWA-400B	400	20300	180000	35.2
FWA-500A	500	39000	120000	35.0
FWA-600A	600	46000	140000	47.0
FWA-700A	700	75000	220000	49.0
FWA-800A	800	92000	280000	58.0
FWA-1000A	1000	170000	510000	60.0

• Watts loss provided at rated current.

• See accessories on page 141.

Features and Benefits

- Excellent DC performance
- Low arc voltage and low energy let-through (I^2t)
- Low watts loss
- Superior cycling capability

Typical Applications

- DC Common bus
- DC Drives
- Power converters/rectifiers
- Reduced voltage starters

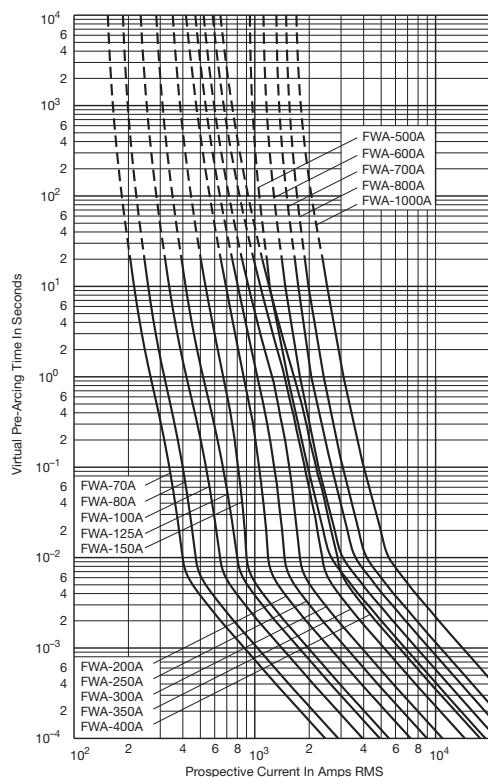
High Speed Fuses

Bussmann
by 

North American — FWA 150V: 70-1000A

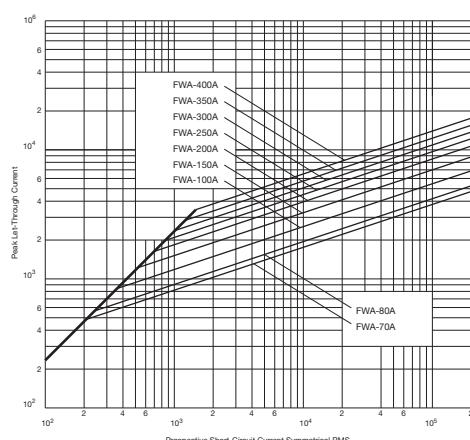
FWA 70-1000A: 150V

Time-Current Curve



High Speed
Fuses

Peak Let-Through Curve



Data Sheet: 35785310

For product data sheets, visit www.cooperbussmann.com/DatasheetsEle

129

2.4 Leistungsdaten Hirschmann elektronischer Gasdrehgriff v1.0

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

Leistungsdaten Hirschmann elektronischer Gasdrehgriff v1.0

Performance data Hirschmann electronic throttle grip v1.0.



Funktion <i>Function</i>	Abteilung <i>Departement</i>	Name / Unterschrift <i>Name / Signature</i>	Datum: <i>Date</i>
Entwicklung <i>development</i>	D_SC_ST	Mähr Dietmar /	19.09.2017
	D_SC_ST	Grohs Thomas /	19.09.2017
Qualitätsabteilung <i>Quality Departement</i>	Q_S_A	Holzer Thomas /	19.9.2017
Projektleiter <i>Project Manager</i>	OP_SC	Zelenovic Novica /	19.9.2017

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	1 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

Änderungsdokumentation/ *Change documentation:*

Nr.	Kurzbeschreibung <i>short description</i>	Datum <i>date</i>	Bearbeiter <i>Editor</i>
1.0	Erstellung <i>creation</i>	05.09.2017	Mähr D. / Grohs T.

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	2 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip

C.02160 Hirschmann E-Gas 65°

EPS-100120

1 Inhalt

Leistungsdaten Hirschmann elektronischer Gasdrehgriff v2.0	1	9.23 Störfestigkeit gegenüber eingestrahlte magnetische Felder (Antennen Methode)	25
Änderungsdokumentation	2	9.24 Störfestigkeit gegenüber eingestrahlte magnetische Felder (BCI Methode)	26
1 Inhalt	3	9.25 Störfestigkeit gegenüber niederfrequenten Magnetfeldern	26
2 Einleitung	5	9.26 Messung elektromagnetisch Abstrahlung	27
2.1 Vertraulichkeit	5	9.27 Störfestigkeit gegenüber elektrostatischer Entladung - Handlingtest	27
2.2 Zweck des Dokumentes	5	9.28 Störfestigkeit gegenüber elektrostatischer Entladung – Systemtest bestromt	27
2.3 Abkürzungen	5		
3 Allgemeine Beschreibung	6		
4 Betriebsbedingungen	6		
5 Elektrische Eigenschaften	7		
5.1 Kontaktbelegung	7		
5.2 Ausgangssignale	7		
5.3 Beschaltung Baugruppe	8		
5.4 Linearitätsfehler	8		
5.5 Gleichlauffehler	8		
5.6 Diagnosebereich	8		
5.7 Elektrische Grenzwerte	9		
5.8 Elektrische Eigenschaften	9		
6 Mechanische Eigenschaften	10		
6.1 Mechanische Parameter	10		
6.2 Stecker	10		
6.3 Mechanische Grenzwerte	10		
6.4 Parameter nach Alterung über Lebenszeit	11		
6.5 Fixierung des E-Gas Drehgriffes	11		
7 Funktionale Sicherheit	12		
8 Validierung	15		
9 Definition der Validierungsprüfungen	16		
9.1 Großer Funktionstest	16		
9.2 Kleiner Funktionstest	16		
9.3 Temperaturschockprüfung	17		
9.4 Temperaturwechselprüfung	17		
9.5 Lagerung trockene Wärme	18		
9.6 Feuchte Wärme zyklisch Lagerung	18		
9.7 Thermoschockprüfung	19		
9.8 Salznebelprüfung	19		
9.9 IP6KX Staubprüfung	20		
9.10 IPX9K Dampfstrahlprüfung	20		
9.11 Isolationswiderstandsprüfung	20		
9.12 Durchschlagsfestigkeitsprüfung	21		
9.13 UV Beständigkeit	21		
9.14 Chemische Beständigkeit	21		
9.15 Falltest	22		
9.16 Kabelzugprüfung	22		
9.17 Dauerlaufprüfung	22		
9.18 Vibration Breitbandrauschen	23		
9.19 Vibration Dauerschock	23		
9.20 Prüfung Rückstellzeit	24		
9.21 Beständigkeit gegenüber Kontaktierungsfehler	24		
9.22 Störfestigkeit gegenüber Transienten	25		

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)		
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	3	von	27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip

C.02160 Hirschmann E-Gas 65°

EPS-100120

1 Index

Performance data Hirschmann electronic throttle grip v2.0	1
<i>Change documentation:</i>	2
1 Index	4
2 Introduction	5
2.1 Confidentiality	5
2.2 Purpose of the documents	5
2.3 Acronym	5
3 General Description	6
4 Operating Conditions	6
5 Electrical properties	7
5.1 Pinning	7
5.2 Output signal	7
5.3 Wiring assembly	8
5.4 Linearity error	8
5.5 Synchronism error	8
5.6 Diagnosis area	8
5.7 Electrical limits	9
5.8 Electrical characteristics	9
6 Mechanical characteristics	10
6.1 Mechanical parameters	10
6.2 Connector	10
6.3 Mechanical limits	10
6.4 Parameter after aging over lifetime	11
6.5 Fixation of ETG	11
7 Functional safety	12
8 Validation	15
9 Definition of validation test	16
9.1 Large function test	16
9.2 Small function test	16
9.3 Temperature shock test	17
9.4 Temperature cycle	17
9.5 Aging in dry heat	18
9.6 Humidity heat cyclic	18
9.7 Thermal shock	19
9.8 Salt spray fog test	19
9.9 IP6Kx dust test	20
9.10 IPX9K pressure washer test	20
9.11 Insulation resistance	20
9.12 Dielectric strength test	21
9.13 UV Resistance test	21
9.14 Chemical resistance	21
9.15 Drop test	22
9.16 Resistance to harness tearing	22
9.17 Fatigue stop test	22
9.18 Vibration broadband random	23
9.19 Vibration endurance shock test	23
9.20 Return time test	24
9.21 Resistance to accidental connection	24
9.22 Immunity against induced transient disturbances	25
9.23 Immunity against electromagnetic disturbances (Antenna Method)	25

9.24 Immunity against RF current on cable cluster (BCI Methode)	26
9.25 Immunity against low frequency magnetic fields	26
9.26 Radiated emissions	27
9.27 Immunity against electrostatic discharge - handling	27
9.28 Immunity against electrostatic discharge - powered up	27

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)		
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	4	von	27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

2 Einleitung

2.1 Vertraulichkeit

Das vorliegende Dokument ist nur zur Einsicht für direkt am Projekt beteiligte Personen bestimmt. Eine Weitergabe am Dritte ist nicht erlaubt.

2.2 Zweck des Dokumentes

Das vorliegende Dokument dient als Definition der Leistungsdaten des elektronischen Gasdrehgriffes der Firma Hirschmann Automotive.

Konstruktive Details sind den entsprechenden Freigabezeichnungen zu entnehmen. In folgender Reihenfolge sind die Dokumente zu priorisieren (1=höchste Prioritäten):

- 1- Freigabezeichnung (906-716-..., 906-822-...)
- 2- Leistungsdaten
- 3- Qualitätsvorschriften
- 4- Mitgeltende Unterlagen

Die Leistungsdaten dienen als Ergänzung und Erweiterung der Freigabezeichnung.

2.3 Abkürzungen

Abkürzung	Bedeutung
E-Gas Drehgriff	Elektronischer Gas Drehgriff

2 Introduction

2.1 Confidentiality

This document is intended only for inspection for directly involved people of the project. Forwarding to third parties is not allowed.

2.2 Purpose of the documents

This document is intended as a definition of the performance data of the electronic throttle grip from Hirschmann Automotive.

Construction details are given in the corresponding release drawings. In the following order, the documents are to prioritize (1= highest priority):

- 1- Released drawing (906-716-..., 906-822-...)
- 2- Performance data
- 3- Quality regulations
- 4- Applicable documents

Performance data serve as a complement and extension of released drawing.

2.3 Acronym

Acronym	Meaning
ETG	Electronic throttle grip
ECU	Electronic control unit

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	5 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip

C.02160 Hirschmann E-Gas 65°



EPS-100120

3 Allgemeine Beschreibung

Die Aufgabe des E-Gas Drehgriffes ist die Messung des Drehwinkels des Gasdrehgriffes welcher durch den Bediener erzeugt wird. Die daraus resultierende Winkelfunktion wird in ein elektrisches Signal umgewandelt welches dem Steuergerät zur weiteren Verarbeitung zur Verfügung gestellt wird. Das Messprinzip der Sensorik basiert auf Basis der HALL Technologie, und ist somit berührungslos und verschleißfrei. Die Sensorik ist redundant ausgeführt um eine Plausibilisierung der Signale im Steuergerät zu ermöglichen. Die Signale sind als analoge ratiometrische Spannungen ausgeführt.

4 Betriebsbedingungen

Der E-Gas Drehgriff ist unter folgenden Betriebsbedingungen zu verwenden.

Parameter	Wert
Betriebstemperatur (Top)	-20°C ... +85°C
Lagertemperatur (Tstor)	-30°C ... +90°C
Versorgungsspannung (Vdd1,2)	+5,0 ±0,5 V
Ausgangssignal (Vout1,2)	Siehe Grafik1
Versorgungsstrom pro Sensor (Icc1,2)	< 15mA
Belastung Ausgang (Rload)	> 1,0kΩ
Drehmoment Betrieb (Mop)	-2...+1Nm
Drehmoment Mißbrauch (Mmu)	+/-15Nm

3 General Description

The purpose of the ETG is to measure the rotation angle of the throttle grip which is generated by the operator. This resulting rotation angle function is converted into an electrical signal which is provided to the ECU for further processing. The measuring principle of the sensor is based on the HALL technology and is therefore contactless and wearless.

The sensor is designed with redundancy to enable a plausibility check of the signals in the control unit. The signals are designed as analog ratiometric voltages.

4 Operating Conditions

The E - throttle grip must be used under the following operating conditions

Parameter	Value
Operating temperature (Top)	-20°C ... +85°C
Storage temperature (Tstor)	-30°C ... +90°C
Supply voltage (Vdd1,2)	+5,0 ±0,5 V
Output signal (Vout1,2)	see graph1
Supply current per sensor (Icc1,2)	< 15mA
Output load (Rload)	> 1kΩ
Operating torque (Mop)	-2...+1Nm
Misuse torque (Mmu)	+/-15Nm

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	6 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

5 Elektrische Eigenschaften

5.1 Kontaktbelegung

Kontakt	Bezeichnung	Funktion
1	Vdd 1	Versorgung1
2	Gnd 1	Masse1
3	Out 1	Signal 1
4	Vdd 2	Versorgung 2
5	Gnd 2	Masse 2
6	Out 2	Signale 2

5.2 Ausgangssignale

Die beiden Sensorsignale sind als analoge Ausgangsstufen mit ratiometrischem Signalverhalten ausgeführt. Die Sensorsignale haben folgende Charakteristiken.

Drehwinkel	Signal1	Signal 2
-10 °	10,8 [%Vdd]	5,4 [%Vdd]
0 °	20,0 [%Vdd]	10,0 [%Vdd]
65 °	80,0 [%Vdd]	40,0 [%Vdd]

5 Electrical properties

5.1 Pinning

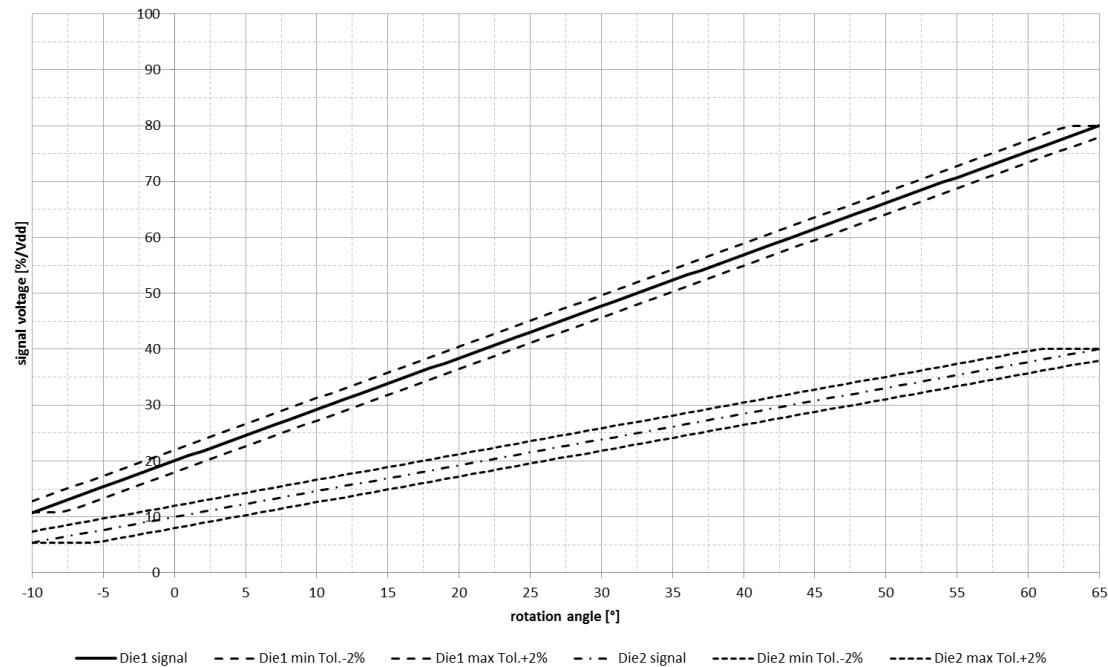
Pin	Description	Function
1	Vdd 1	Supply1
2	Gnd 1	Ground1
3	Out 1	Signal 1
4	Vdd 2	Supply2
5	Gnd 2	Ground2
6	Out 2	Signal2

5.2 Output signal

The two sensor signals are designed as an analog output stages with ratiometric signal behavior. The sensor signals have the following characteristics.

Rotation angle	Signal1	Signal 2
-10 °	10,8 [%Vdd]	5,4 [%Vdd]
0 °	20,0 [%Vdd]	10,0 [%Vdd]
65 °	80,0 [%Vdd]	40,0 [%Vdd]

sensor signal characteristic +/- 2% linearity error tolerance



graph1: sensor signal characteristic

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	7 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff Performance data Hirschmann electronic throttle grip

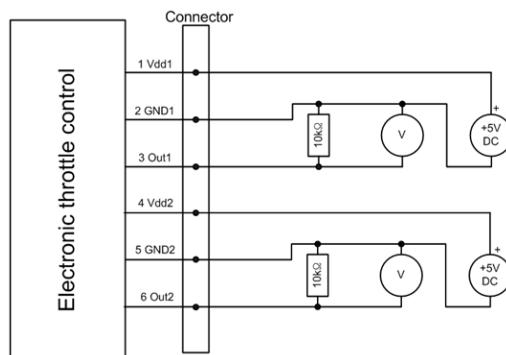


C.02160 Hirschmann E-Gas 65°

EPS-100120

5.3 Beschaltung Baugruppe

Der E-Gas Drehgriff ist entsprechend Grafik2 zu beschalten.



Grafik2: Beschaltung Baugruppe

5.4 Linearitätsfehler

Für die beiden Sensorsignale ist ein Linearitätsfehler von +/-2% zulässig. Zur Berechnung des Linearitätsfehlers dient die Versorgungsspannung als Referenz.

$$\text{lin_error} = \frac{(V_{\text{signal}} - V_{\text{ideal}})}{V_{\text{cc}}} * 100\%$$

5.5 Gleichlauffehler

Um eine Plausibilisierung im Steuergerät zu ermöglichen, werden die beiden Sensorsignale zueinander mit Hilfe des Gleichlauffehlers bewertet. Der Gleichlauffehler ist mit einer maximalen Toleranz von +/-2% definiert und wird nach folgender Formel berechnet:

$$\text{sync_error} = \frac{(V_{\text{signal1}} - V_{\text{signal2}})^2}{V_{\text{cc}}} * 100\%$$

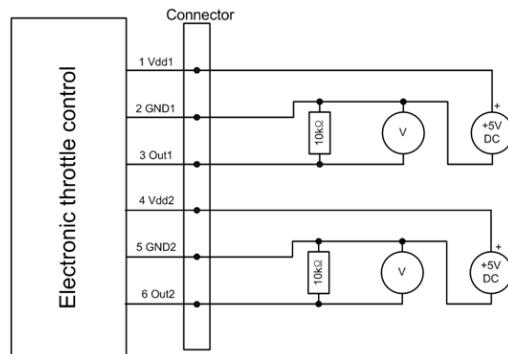
5.6 Diagnosebereich

Zusätzlich zum normalen Betriebsbereich der Sensorsignale ist ein Diagnosebereich definiert. Im Fehlerfall wechselt der Sensor in diesen Signalbereich um dem Steuergerät eine Fehlfunktion des Sensors anzudecken.

Bezeichnung	Wertebereich
DiagLow	0...4 [%Vdd]
DiagHigh	96..100 [%Vdd]

5.3 Wiring assembly

The ETG has to be wired according to the graph2.



graph2: wiring assembly

5.4 Linearity error

For the two sensor signals a linearity error of +/- 2% is permissible. In order to calculate the linearity error, the supply voltage is used as reference.

$$\text{lin_error} = \frac{(V_{\text{signal}} - V_{\text{ideal}})}{V_{\text{cc}}} * 100\%$$

5.5 Synchronism error

In order to allow a plausibility check in the control unit, the two sensor signals are comparable to each other using the synchronization error. The synchronization error is defined with a maximum tolerance of +/- 2 % and is calculated by the following formula:

$$\text{sync_error} = \frac{(V_{\text{signal1}} - V_{\text{signal2}})^2}{V_{\text{cc}}} * 100\%$$

5.6 Diagnosis area

In addition to the normal operating range of the sensor signals a diagnostic range is defined. If an error occurs, the sensor switches in this signal range to indicate the ECU a malfunction of the sensor.

Parameter	Value area
DiagLow	0...4 [%Vdd]
DiagHigh	96..100 [%Vdd]

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	8 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

5.7 Elektrische Grenzwerte

Der Sensorbaustein besitzt folgende elektrischen Grenzwerte, welche nicht überschritten werden dürfen. Eine Überschreitung der Grenzwerte kann eine permanente Beschädigung der Baugruppe zur Folge haben.

Parameter	Wert
Überspannung Vdd	+24V
Verpolspannung Vdd	-12V
Überspannung Vout	+18V
Verpolspannung Vout	-0,3V
Ausgangstrom Iout	+30mA

5.8 Elektrische Eigenschaften

Der Sensorbaustein besitzt folgende elektrischen Eigenschaften im Normalbetrieb.

Parameter	Min	Typ	Max	Einheit
elektrische Parameter				
Versorgungsspannung (Vdd)	4,5	5,0	5,5	V
Versorgungsstrom (Icc)		13,5	15	mA
Belastung Ausgang (Rload)	1,0	10	∞	k Ω
zeitliches Verhalten				
Auffrischungsrate		400		μ s
Startzeit			10	ms
Überwachungszeit			4,58	ms
Anstiegsgeschwindigkeit Ausgang		2,5		V/ms
Genauigkeit				
Auflösung Ausgangsstufe		0,025		%Vdd/ LSB
EMV/ESD				
ESD Festigkeit Luft Entladung			+/-15	kV
ESD Festigkeit Kontakt Entladung			+/-4	kV
EMV (stripline)			+/-200	V/m

5.7 Electrical limits

The sensor system has the following electrical limits, which aren't allowed to be exceeded. Exceeding of the critical limits can cause a permanent damage to the sensor system.

Parameter	Value
Oversupply Vdd	+24V
Reverse Voltage Vdd	-12V
Oversupply Vout	+18V
Reverse Voltage Vout	-0,3V
Output current Iout	+30mA

5.8 Electrical characteristics

The sensor system has the following electrical characteristics in the normal operation mode.

Parameter	Min	Typ	Max	Einheit
electrical parameter				
Supply voltage (Vdd)	4,5	5,0	5,5	V
Supply current (Icc)		13,5	15	mA
Output load resistor (Rload)	1,0	10	∞	k Ω
timing parameter				
Refresh rate		400		μ s
Startup time			10	ms
Watchdog			4,58	ms
Slew rate output		2,5		V/ms
Accuracy				
Output resolution		0,025		%Vdd/ LSB
EMV/ESD				
ESD stability air discharge			+/-15	kV
ESD stability contact discharge			+/-4	kV
EMV (stripline)			+/-200	V/m

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	9 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

6 Mechanische Eigenschaften

6.1 Mechanische Parameter

Der E-Gas Drehgriff besitzt im normalen Betriebszustand folgende mechanische Eigenschaften.

Parameter	Min	Typ	Max	Einheit
Drehwinkel				
Leerlauf	-2	0	2	°
Vollgas	63	65	67	°
Tempomat	-12	-10	-8	°
Drehmoment (Neuzustand)				
Drehmoment 5°	0,28	0,40	0,52	Nm
Drehmoment 75°	0,47	0,60	0,73	Nm
Drehmoment -10°	1,40	1,75	2,10	Nm
Mechanische Beständigkeit				
Anzahl Zyklen 0°...80°			1. Mio.	Zyklen
Anzahl Zyklen 0°...-10°			10k	Zyklen

6.2 Stecker

Der Stecker ist ein Bauteil der "Seal Star" Familie, welcher eine integrierte Aderabdichtung besitzt. Die Einzelkontakte des 6 poligen Steckers haben ein Rastermaß von 4,0mm. Die Stiftkontakte haben die Maße 1,2 x 0,6mm.

6.3 Mechanische Grenzwerte

Der E-Gas Drehgriff ist für ein Drehmoment im Missbrauchsfall von maximal **15Nm** ausgelegt. Eine Überschreitung der Missbrauchsgrenze ist nicht erlaubt. Bei Überschreitung der mechanischen Grenze, kann es zu Veränderungen einzelner technischer Parameter kommen oder sogar zur mechanischen Zerstörung der Baugruppe.

6 Mechanical characteristics

6.1 Mechanical parameters

The ETG has the following mechanical properties in normal operating condition.

Parameter	Min	Typ	Max	Unit
Rotation angle				
idle position	-2	0	2	°
full throttle position	63	65	67	°
cruise control position	-12	-10	-8	°
Torque (new condition)				
torque 5°	0,28	0,40	0,52	Nm
torque 75°	0,47	0,60	0,73	Nm
torque -10°	1,40	1,75	2,10	Nm
Mechanical durability				
Number cycles 0°...80°			1. Mio.	cycles
Number cycles 0°...-10°			10k	cycles

6.2 Connector

The connector is a component of the "Seal Star" family, which has an integrated wire seal. The contacts of the 6-pin connector have a pitch of 4,0mm. The pin contacts have the dimensions 1,2 x 0,6mm.

6.3 Mechanical limits

The ETG is designed for a torque in the case of misuse of a maximum of 15Nm. Exceeding the misuse limit is not allowed. If this mechanical limit is exceeded, there may be changes of individual technical parameters or even mechanical destruction of the ETG.

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	10 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip

C.02160 Hirschmann E-Gas 65°

EPS-100120

6.4 Parameter nach Alterung über Lebenszeit

Folgende Parameter gelten für den E-Gas Drehgriff nach einer Alterung von 1Mio Zyklen.

Parameter	Min	Typ	Max	Einheit
Sensorsignal				
Linearitätsfehler	-3	0	3	%/Vdd
Drehmoment				
Drehmoment 5°	0,28	0,40	0,70	Nm
Drehmoment 75°	0,47	0,60	1,10	Nm
Drehmoment -10°	1,40	1,75	2,30	Nm

Durch die Alterung im Betriebsfall kann es zu einer Abnützung der Weichkomponente kommen.

6.5 Fixierung des E-Gas Drehgriffes

Der E-Gas Drehgriff wird über ein Klemmprofil auf das Lenkerrohr befestigt. Die Außenmaße des Lenkerrohrdurchmessers müssen innerhalb des Toleranzbereiches von 22,0mm bis 22,15mm liegen. Ist der Lenkerrohrdurchmesser größer oder kleiner als der definierte Toleranzbereich, kann keine ausreichende Fixierung des Gasdrehgriffes gewährleistet werden.

Für die Fixierung des E-Gas Drehgriffes auf dem Lenkerrohr, muss die Klemmprofilschraube mit einem Drehmoment von 3,0Nm +/-0,1Nm angeschraubt werden.

Die Fixierung der Baugruppe auf einer Lenkerrohrstange durch das Klemmprofil darf nur einmal erfolgen. Ein zweiter Verschraubungsprozess des Klemmprofiles ist nicht zulässig.

Im Falle einer Fehlfunktion oder Beschädigung ist der E-Gas Drehgriff zu ersetzen. Eine Reparatur des E-Gas Drehgriffes ist nicht erlaubt.

6.4 Parameter after aging over lifetime

The following parameters are valid to the ETG after aging for 1 million cycles.

Parameter	Min	Typ	Max	Unit
Sensor signal				
Linearity error	-3	0	3	%/Vdd
torque				
torque 5°	0,28	0,40	0,70	Nm
torque 75°	0,47	0,60	1,10	Nm
torque -10°	1,40	1,75	2,30	Nm

Through the aging over Lifetime a wear and tear of the soft component can be caused.

6.5 Fixation of ETG

The ETG is fastened via a clamping profile on the handlebar. The dimensions of the outer handlebar diameter must be within the tolerance range of 22,0mm to 22,15mm. If the outer handlebar diameter is larger or smaller than the defined tolerance range, a correct fixation of the ETG can't be ensured.

For the fixation of the ETG on the handlebar, the clamping profile screw must be tightened with a torque of 3,0 Nm +/- 0,1Nm.

Fixing the ETG on a handlebar through the clamping profile is only once allowed. A second screwing process of the clamping profile screw is not permitted.

In case of malfunction or defects the ETG has to be replaced. Reparation of the ETG is not allowed.

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	11 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

7 Funktionale Sicherheit

Zur Bewertung der funktionalen Sicherheit des E-Gas Drehgriffes im Sinne der ISO26262 wurden Erfahrungen und Bewertungen aus einem E-Gas Vorgängerprojekt berücksichtigt.

ASIL Level:

Für die Bewertung der funktionalen Sicherheit vom E-Gas Drehgriff wurden die Anforderungen der ASIL Level B Bewertungsstufe herangezogen.

Sicherheitsziel Gesamtsystem:

Ein Selbstbeschleuniger soll verhindert werden.

Sicherer Zustand Gesamtsystem:

Die Fahrzeugbeschleunigung ist kleiner oder gleich der gewünschten Beschleunigung des Fahrers.

Zugehörige Fehlfunktion:

Drosselklappe ist weiter geöffnet als vom Fahrer gewünscht (Selbstbeschleuniger).

Funktionale Sicherheitsanforderungen:

Zur Erreichung der Sicherheitsziele sind folgende funktionale Sicherheitsanforderungen für den E-Gas Drehgriff definiert:

1. Der E-Gas Drehgriff soll ein Sensorsignal (Signal1) zur Verfügung stellen, welches dem Drehwinkel des E-Gas Drehgriffes entspricht.
2. Der E-Gasdrehgriff soll ein Sensorsignal (Signal2) zur Verfügung stellen, welches dem Drehwinkel des E-Gas Drehgriffes entspricht und dazu geeignet ist Fehler des Sensorsignales Signal1 im gesamten Arbeitsbereich durch einen Vergleich im Steuergerät zu erkennen.
3. Der Signalpfad der beiden Sensorsignale ist unabhängig voneinander auszuführen.

7 Functional safety

Experiences and reviews from a previous ETG project were taken into account in order to evaluate the functional safety of the ETG within the meaning of ISO26262.

ASIL Level:

For the evaluation of functional safety of the ETG the requirements of the ASIL Level B have been used.

Safety goal complete system:

A self-accelerator has to be prevented

Safe state complete system:

The vehicle acceleration is less than or equal to the desired acceleration of the driver.

Related malfunction:

Throttle is opened further as desired by the driver (self-accelerator).

Functional safety requirements:

For the achievement of the safety goals following functional safety requirements for the ETG are defined.

1. The ETG should provide a sensor signal (Signal1) which corresponds to the rotation angle of the throttle grip.
2. The ETG should provide a sensor signal (Signal2) which corresponds to the rotation angle of the throttle grip and is suitable to detect errors of the sensor signal Signal1 in the entire working range by comparison in the ECU.
3. The signal path of the two sensor signals have to be independent from each other.

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)		
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	12	von	27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip

C.02160 Hirschmann E-Gas 65°

EPS-100120

Sicherheitsrelevante Fehler des E-Gas Drehgriffes

1. **Gefährlicher Fehler:** Das Sensorsignal (Signal1) liegt innerhalb des plausiblen Signalbereiches und entspricht einem Drehwinkel welcher größer ist, als der tatsächliche Drehwinkel des Gasdrehgriffes.
2. **Latenter Fehler:** Das Sensorsignal (Signal2) liegt innerhalb des plausiblen Signalbereichs und entspricht einem Drehwinkel welcher größer ist, als der tatsächliche Drehwinkel des Gasdrehgriffes.

Diagnosemaßnahmen zur Erkennung sicherheitsrelevanter Fehler

Um sicherheitsrelevante Fehler erkennen zu können müssen folgende Sicherheitsmaßnahmen im Steuergerät umgesetzt werden:

1. Sensorsignale des E-Gas Drehgriffes unterhalb des Minimalwertes oder oberhalb des Maximalwertes sind als Fehler zu erkennen.
2. Abweichungen des Sensorsignals Signal1 zum Sensorsignal Signal2, die größer einer Fehlererkennungsschwelle sind, werden als Fehler erkannt.

Fehlerreaktionszeit

Im Falle von internen Fehlern und Unterbrechungen der Spannungsversorgung oder der Masseanbindung wechselt der E-Gasdrehgriff innerhalb von 40ms in den sicheren Zustand.

Sicherer Zustand

Folgende Betriebszustände sind als sicherer Zustand zu bewerten:

1. Mindestens eines der Sensorsignale liegt unterhalb oder oberhalb des Minimal- bzw. Maximalwertes der Diagnosemaßnahme 1.
2. Die Abweichung des Sensorsignales Signal1 zum Sensorsignal Signal2 ist größer als die Fehlererkennungsschwelle der Diagnosemaßnahme 2.

Safety related fault of the ETG

1. **Dangerous fault:** The sensor signal (signal 1) is within the plausible signal range and corresponds to a rotation angle which is larger than the actual rotation angle of the throttle grip.
2. **Latent fault:** The sensor signal (Signal2) is within the plausible signal range and corresponds to a rotation angle which is larger than the actual rotation angle of the throttle grip.

Diagnostic measures to detect safety relevant faults

To be able to detect safety relevant faults, the following safety measures have to be implemented in the ECU:

1. Sensor signals of the ETG below the minimum value or above the maximum value have to be detected as a fault.
2. Deviations of the sensor signal Signal1 to the sensor signal Signal2 which are greater than the fault detection threshold have to be detected as a fault.

Fault reaction time

In case of internal errors and interruptions of the power supply or the ground connection, the ETG will switch within 40 ms into the safe state.

Safe State:

The following operating states have to be rated as a safe state:

1. At least one of the sensor signals are below or above the minimum or maximum value of the diagnostic measure 1.
2. The deviation of the sensor signal Signal1 to the sensor signal Signal2 is greater than the fault detection threshold of the diagnostic measure 2.

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)		
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	13	von	27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

Quantitative Bewertung der Sicherheitsanforderungen

Die folgenden Werte sind die Ergebnisse aus der FMEDA für die "Single Point Fehler" und die "Latent Multiple Point Fehler":

$$\lambda_{resfault} = 0,9653 \text{ [FIT]}$$

$$\lambda_{latfault} = 0,9653 \text{ [FIT]}$$

Qualitative Bewertung der Sicherheitsanforderungen

Die folgenden Metriken ergeben sich aus der FMEDA.

1. SPF-Metriken: 99,65%
2. LF-Metriken: 99,65%

Quantitative evaluation of the safety requirements

The following values are the results of the FMEDA for the "Single Point Fault" and the „latent Multiple Point Fault“:

$$\lambda_{resfault} = 0,9653 \text{ [FIT]}$$

$$\lambda_{latfault} = 0,9653 \text{ [FIT]}$$

Qualitative evaluation of the safety requirements

The following metrics are the results of the FMEDA.

1. SPF-Metriken: 99,65%
2. LF-Metriken: 99,65%

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)		
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	14	von	27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff

Performance data Hirschmann electronic throttle grip

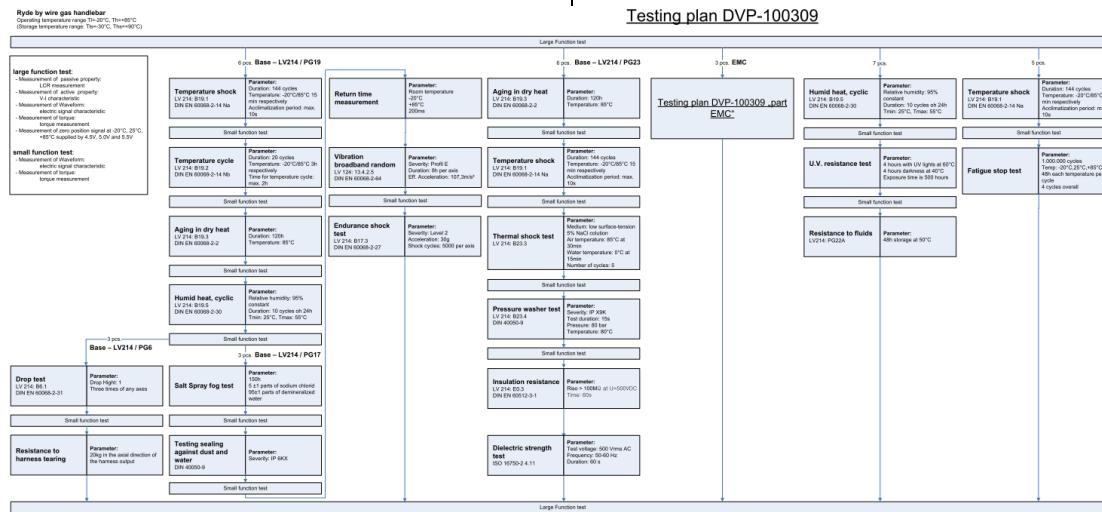


C.02160 Hirschmann E-Gas 65°

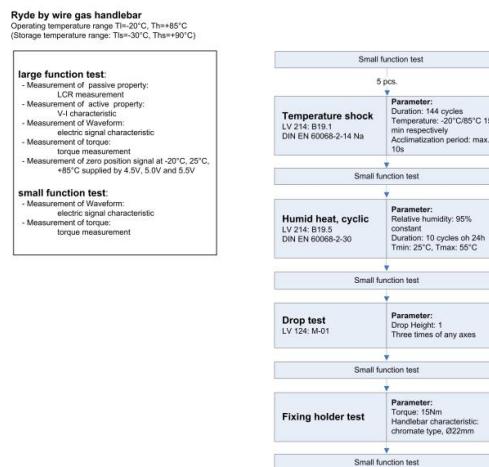
EPS-100120

8 Validierung

Das Produkt wurde entsprechend folgendem Validierungsplan geprüft.



Testing plan DVP-100361 65° variante



Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	15 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

9 Definition der Validierungsprüfungen

Die Validierungsprüfungen wurden entsprechend folgender Definitionen durchgeführt

9.1 Großer Funktionstest

Prüfungsdefinition:

Folgende Prüfungen wurden beim großen Funktionstest durchgeführt:

- Messung Sensor Signal Charakteristik
- Messung passiver Eigenschaften mittels LCR Meter
- Messung der Halbleitereigenschaften mittels U-I Messgerät
- Messung des Drehmomentverlaufes
- Messung der Leerlaufposition bei -20°C, +25°C und +85°C bei einer Versorgungsspannung von 4.5V, 5.0V und 5.5V

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen.

9.2 Kleiner Funktionstest

Prüfungsdefinition:

Folgende Prüfungen wurden beim kleinen Funktionstest durchgeführt:

- Messung Sensor Signal Charakteristik
- Messung des Drehmomentverlaufes

Prüfungskriterium:

Die Sensoreigenschaften dürfen die spezifizierten Toleranzgrenzen nicht überschreiten.

9 Definition of validation test

The validation tests were executed according following definitions.

9.1 Large function test

Test definition:

The following tests were executed for the large function test:

- Measurement Sensor signal characteristic
- Measurement passive characteristics with LCR Meter
- Measurement active characteristics with U-I Meter
- Measurement of torque characteristic
- Measurement of zero position signal at -20°C, +25°C and +85°C supplied by 4.5V, 5.0V and 5.5V

Test criteria:

The measurement data's fulfill the specified tolerances.

9.2 Small function test

Test definition:

The following tests were executed for the large function test:

- Measurement Sensor signal characteristic
- Measurement of torque characteristic

Test criteria:

The measurement data's fulfill the specified tolerances.

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)		
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	16	von	27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

9.3 Temperaturschockprüfung

Prüfungsdefinition:

Die Temperaturschockprüfung wurde in Anlehnung an die Norm LV214:B19.1 durchgeführt.

Parameter	Wert
Temperatur min	-20°C
Temperatur max	+85°C
Verweildauer	15 min.
Umlagerungsdauer	max 10s
Zyklen	144

Während der Prüfung wurden das Sensorsignal und die Stromaufnahme überwacht.

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen.

9.4 Temperaturwechselprüfung

Prüfungsdefinition:

Die Temperaturwechselprüfung wurde in Anlehnung an die Norm LV214:B19.2 durchgeführt.

Parameter	Wert
Temperatur min	-20°C
Temperatur max	+85°C
Verweildauer	3h
Zyklen	20

Während der Prüfung wurden das Sensorsignal und die Stromaufnahme überwacht.

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen.

9.3 Temperature shock test

Test definition:

The temperature shock was tested in dependence to the norm LV214:B19.1

Parameter	Value
Temperature min	-20°C
Temperature max	+85°C
Duration	15 min.
Acclimatization period	max 10s
Cycles	144

During the test the sensor signal and the sensor current was monitored.

Test criteria:

The measurement data's fulfill the specified tolerances.

9.4 Temperature cycle

Test definition:

The temperature cycle test was tested in dependence to the norm LV214:B19.2

Parameter	Value
Temperature min	-20°C
Temperature max	+85°C
Retention time	3h
Cycles	20

During the test the sensor signal and the sensor current was monitored.

Test criteria:

The measurement data's fulfill the specified tolerances.

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	17 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

9.5 Lagerung trockene Wärme

Prüfungsdefinition:

Die Alterung in trockener Wärme wurde in Anlehnung an die Norm LV214:B19.3 durchgeführt.

Parameter	Wert
Temperatur	+85°C
Verweildauer	120h

Während der Prüfung wurden das Sensorsignal und die Stromaufnahme überwacht.

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen.

9.6 Feuchte Wärme zyklisch Lagerung

Prüfungsdefinition:

Die zyklische Feuchte Wärme Lagerung wurde in Anlehnung an die Norm LV214:B19.5 durchgeführt.

Parameter	Wert
Temperatur min	+25°C
Temperatur max	+55°C
Relative Luftfeuchtigkeit	95%
Zyklusdauer	24h
Zyklen	10

Während der Prüfung wurden das Sensorsignal und die Stromaufnahme überwacht.

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen

9.5 Aging in dry heat

Test definition:

The aging in dry heat test was tested in dependence to the norm LV214:B19.3

Parameter	Value
Temperature	+85°C
Retention time	120h

During the test the sensor signal and the sensor current was monitored.

Test criteria:

The measurement data's fulfill the specified tolerances.

9.6 Humidity heat cyclic

Test definition:

The humidity heat test was tested in dependence to the norm LV214:B19.5

Parameter	Value
Temperature min	+25°C
Temperature max	+55°C
Relative humidity	95%
Cycle duration	24h
cycles	10

During the test the sensor signal and the sensor current was monitored.

Test criteria:

The measurement data's fulfill the specified tolerances

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	18 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

9.7 Thermoschockprüfung

Prüfungsdefinition:

Die Thermoschockprüfung wurde in Anlehnung an die Norm LV214:B23.3 durchgeführt.

Parameter	Wert
Medium	5% NaCl Lösung
Lufttemperatur	+85°C
Wassertemperatur	0°C
Verweildauer Luft	30 min
Verweildauer Wasser	15 min
Zyklen	5

Während der Prüfung wurden das Sensorsignal und die Stromaufnahme überwacht.

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen.

9.8 Salznebelprüfung

Prüfungsdefinition:

Parameter	Wert
Medium	5+/-1 Teile Natriumchlorid 95+/- Teile demineralisiertes Wasser
Prüfungsdauer	150h

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen.

9.7 Thermal shock

Test definition:

The thermal shock test was tested in dependence to the norm LV214:B23.3

Parameter	Value
Medium	5% NaCl dilution
Air temperature	+85°C
Water temperature	0°C
Retention time air	30 min
Retention time water	15 min
cycles	5

During the test the sensor signal and the sensor current was monitored.

Test criteria:

The measurement data's fulfill the specified tolerances

9.8 Salt spray fog test

Test definition:

Parameter	Value
Medium	5+/-1 parts sodium chloride 95+/- 1 parts demineralized water
Test duration	150h

Test criteria:

The measurement data's fulfill the specified tolerances

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	19 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

9.9 IP6KX Staubprüfung

Prüfungsdefinition:

Die IP6KX Staubprüfung wurde entsprechend der Norm DIN 40050-9 durchgeführt.

Parameter	Wert
Teststaub	DIN40050-9 7.3.1
Bestaubungsdauer	6s
Wartezeit	900s
Zyklen	20

Prüfungskriterium:

Kein Staubeintritt in die Baugruppe.

9.10 IPX9K Dampfstrahlprüfung

Prüfungsdefinition:

Die IPX9K Dampfstrahlprüfung wurde entsprechend der Norm LV214:B23.4 durchgeführt.

Parameter	Wert
Wasserdruck	80 bar
Wassertemperatur	80°C
Prüfungsdauer	15s
Zyklen	3

Prüfungskriterium:

Kein Wassereintritt in die Baugruppe. Die Messwerte erfüllen die spezifizierten Toleranzgrenzen.

9.11 Isolationswiderstandsprüfung

Prüfungsdefinition:

Die Isolationswiderstandsprüfung wurde entsprechend der Norm LV214:E 0.3 durchgeführt.

Parameter	Wert
Spannung	500 VDC
Prüfungsdauer	60s

Prüfungskriterium:

Der Isolationswiderstand der Baugruppe ist größer 100MΩ.

9.9 IP6KX dust test

Test definition:

The dust test was tested in dependence to the norm DIN 40050-9.

Parameter	Value
Dust	DIN40050-9 7.3.1
Dusting duration	6s
Retention time	900s
Cycles	20

Test criteria:

No dust entry into the product

9.10 IPX9K pressure washer test

Test definition:

The pressure washer test was tested in dependence to the norm LV214:B23.4

Parameter	Value
Water pressure	80 bar
Water temperature	80°C
Test duration	15s
Cycles	3

Test criteria:

No water entry into the product. The measurement data's fulfill the specified tolerances

9.11 Insulation resistance

Test definition:

The insulation resistance test was tested in dependence to the norm LV214:E 0.3.

Parameter	Value
Testing voltage	500 VDC
Retention time	60s

Test criteria:

The insulation resistance test must be greater than 100MΩ.

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	20 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip

C.02160 Hirschmann E-Gas 65°

EPS-100120

9.12 Durchschlagsfestigkeitsprüfung

Prüfungsdefinition:

Die Durchschlagsfestigkeitsprüfung wurde entsprechend der Norm ISO 16750-2 4.11 durchgeführt.

Parameter	Wert
Testspannung	500VAC
Frequenz	50Hz
Wartezeit	60s

Prüfungskriterium:

Kein elektrischer Überschlag

9.12 Dielectric strength test

Test definition:

The dielectric strength test was tested in dependence to the norm ISO 16750-2 4.11

Parameter	Value
Test voltage	500VAC
Frequency	50Hz
Retention time	60s

Test criteria:

The insulation resistance test must be greater than 100MΩ

9.13 UV Beständigkeit

Prüfungsdefinition:

Die UV Beständigkeitsprüfung wurde entsprechend folgender Parameter durchgeführt.

Parameter	Wert
UV Lampen	8 Stk. 313nm UVB
Bestrahlungsdauer	4h
Pause	4h
Prüfungsdauer	500h

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen

9.13 UV Resistance test

Test definition:

The UV resistance test was tested in dependence to the following parameters.

Parameter	Value
UV lights	8 pcs. 313nm UVB
Radiation time on	4h
Radiation time off	4h
Exposer time	500h

Test criteria:

The measurement data's fulfill the specified tolerances

9.14 Chemische Beständigkeit

Prüfungsdefinition:

Die Prüfung der chemischen Beständigkeit wurde entsprechend der Norm LV214:PG22A durchgeführt.

Chemikalie	Beaufschlagung / Alterung
Cockpitspray	aufsprühen / 48h bei 50°C
Rostlöser	aufsprühen / 48h bei 50°C
Unverdünntes Scheiben-frostschutzmittel	begießen / 48h bei 50°C
Isopropanol	begießen / 48h bei RT
Schmierfett	Einreiben / 48h bei 50°C

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen

9.14 Chemical resistance

Test definition:

The chemical resistance test was tested in dependence to the norm LV214:PG22A

Chemically	Appliance / aging
Cockpit spray	spraying / 48h at 50°C
Penetrating oil	spraying / 48h at 50°C
Undiluted washer fluid anti-freeze	douse / 48h at 50°C
Isopropanol	douse / 48h at RT
Grease	rubbing / 48h at 50°C

Test criteria:

The measurement data's fulfill the specified tolerances

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	21 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

9.15 Falltest

Prüfungsdefinition:

Die Falltest wurde entsprechend der Norm LV214: B6.1 durchgeführt.

Parameter	Wert
Testhöhe	1m
Wiederholungen	3 mal je Raumachse

Prüfungskriterium:

Keine Beschädigung der Baugruppe

9.15 Drop test

Test definition:

The drop test was tested in dependence to the norm LV214: B6.1

Parameter	Value
Drop height	1m
cycles	3 times each axis

Test criteria:

No damage off the product

9.16 Kabelzugprüfung

Prüfungsdefinition:

Die Kabelzugprüfung wurde entsprechend folgender Parameter durchgeführt.

Parameter	Wert
Belastung	200N in axialer Richtung des Kabelaustrittes

Prüfungskriterium:

Keine Beschädigung der Baugruppe

9.16 Resistance to harness tearing

Test definition:

The resistance to harness tearing was tested according following parameters.

Parameter	Value
Strain	200N in axial direction of the harness output

Test criteria:

No damage off the product

9.17 Dauerlaufprüfung

Prüfungsdefinition:

Die Dauerlaufprüfung wurde entsprechend folgender Parameter durchgeführt.

Parameter	Wert
Zyklen	1.000.000
Temperaturprofil	-20°C, +25°C,+85°C jeweils für 48h lagern
Temperaturzyklen	min. 4

Prüfungskriterium:

Keine Beschädigung der Baugruppe

9.17 Fatigue stop test

Test definition:

The fatigue stop test was tested according following parameters.

Parameter	Value
cycles	1.000.000
Temperature profile	-20°C, +25°C,+85°C each temperature 48h
Temperature cycles	min. 4

Test criteria:

No damage off the product

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)		
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	22	von	27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff <i>Performance data Hirschmann electronic throttle grip</i>	 HIRSCHMANN AUTOMOTIVE
C.02160 Hirschmann E-Gas 65°	EPS-100120

9.18 Vibration Breitbandrauschen

Prüfungsdefinition:

Die Vibrationsprüfung mit Breitbandrauschen wurde in Anlehnung an die Norm LV124: 13.4.2.5 durchgeführt.

Parameter	Wert	
Vibrationserregung	Breitbandrauschen	
Prüfungsdauer	8h je Raumachse	
RMS Beschleunigung	107,3 m/s ²	
Vibrationsprofil	Frequenz [Hz]	Effektive Beschleunigung (m/s ²)/Hz
	20	200
	40	200
	300	0,5
	800	0,5
	1000	3
	2000	3

Während der Prüfung wurde das Sensorsignal überwacht.

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen

9.19 Vibration Dauerschock

Prüfungsdefinition:

Die Vibrationsprüfung mit Dauerschockbelastung wurde in Anlehnung an die Norm LV214: B17.3 durchgeführt.

Parameter	Wert
Anzahl positive Schocks	2500
Anzahl negative Schocks	2500
Beschleunigung	30g
Impulszeit	6ms

Während der Prüfung wurde das Sensorsignal überwacht.

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen

9.18 Vibration broadband random

Test definition:

The broadband random vibration test was tested according following norm LV124: 13.4.2.5

Parameter	Value	
Vibration	Broad band random	
Testing duration	8h each axis	
RMS acceleration	107,3 m/s ²	
Vibration profile	frequency [Hz]	Eff. acceleration (m/s ²)/Hz
	20	200
	40	200
	300	0,5
	800	0,5
	1000	3
	2000	3

During the test the sensor signal and the sensor current was monitored.

Test criteria:

The measurement data's fulfill the specified tolerances

9.19 Vibration endurance shock test

Test definition:

The endurance shock test was tested according following norm LV214: B17.3

Parameter	Value
number positive Shocks	2500
number negative Shocks	2500
Acceleration	30g
impulse time	6ms

During the test the sensor signal and the sensor current was monitored.

Test criteria:

The measurement data's fulfill the specified tolerances

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)		
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	23	von	27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

9.20 Prüfung Rückstellzeit

Prüfungsdefinition:

Die Prüfung der Rückstellzeit wurde entsprechend folgender Parameter durchgeführt:

Parameter	Wert
Prüfungstemperatur	-20°C / +85°C
Rückstellzeit	200ms
Prüfwinkel 1	80° -> 0°
Prüfwinkel 2	5° -> 0°

Während der Prüfung wurde das Sensorsignal überwacht.

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen

9.21 Beständigkeit gegenüber Kontaktierungsfehler

Prüfungsdefinition:

Die Prüfung wird entsprechend folgender Parameter durchgeführt.

Parameter	Wert
Kontaktierungsvarianten	Vdd -> Vbat
	Vdd -> -Vbat
	Out -> Gnd
	Out -> Vdd

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen

9.20 Return time test

Test definition:

The endurance shock test was tested according following parameters

Parameter	Value
Testing temperature	-20°C / +85°C
Return time	200ms
Testing angle 1	80° -> 0°
Testing angle 2	5° -> 0°

During the test the sensor signal and the sensor current was monitored.

Test criteria:

The measurement data's fulfill the specified tolerances

9.21 Resistance to accidental connection

Test definition:

The endurance shock test was tested according following parameters

Parameter	Value
Accidental connection	Vdd -> Vbat
	Vdd -> -Vbat
	Out -> Gnd
	Out -> Vdd

Test criteria:

The measurement data's fulfill the specified tolerances

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	24 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

9.22 Störfestigkeit gegenüber Transienten

Prüfungsdefinition:

Die Prüfung wurde in Anlehnung an die Norm ISO 7637-3 durchgeführt.

Parameter	Wert
Amplitude	3a: -300V 3b: 300V
Prüfzeit	30 min (10 Minuten Einzellitze, 20 Minuten Leitungssatz)

Während der Prüfung wurde das Sensorsignal überwacht

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen

9.23 Störfestigkeit gegenüber eingestrahlte magnetische Felder (Antennen Methode)

Prüfungsdefinition:

Die Prüfung wurde in Anlehnung an die Norm ISO 11452-5 durchgeführt.

Parameter	Wert
Frequenzbereich	10 MHz ... 1,0 GHz
Feldstärke	200 V/m

Während der Prüfung wurde das Sensorsignal überwacht

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen

9.22 Immunity against inducted transient disturbances

Test definition:

The immunity against inducted transient test was tested according following norm ISO 7637-3.

Parameter	Value
Amplitude	3a: -300V 3b: 300V
Testing duration	30 min (10 Minute single wire, 20 Minute cluster of cable)

During the test the sensor signal and the sensor current was monitored.

Test criteria:

The measurement data's fulfill the specified tolerances

9.23 Immunity against electromagnetic disturbances (Antenna Method)

Test definition:

The immunity against electromagnetic disturbances test was tested according following norm ISO 11452-5

Parameter	Wert
Frequency range	10 MHz ... 1,0 GHz
Magnetic field	200 V/m

During the test the sensor signal and the sensor current was monitored.

Test criteria:

The measurement data's fulfill the specified tolerances

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	25 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

9.24 Störfestigkeit gegenüber eingestrahlte magnetische Felder (BCI Methode)

Prüfungsdefinition:

Die Prüfung wurde in Anlehnung an die Norm ISO 11452-4 durchgeführt.

Parameter	Wert
Prüfstrom	200mA f < 10MHz 200mA 10MHz < f < 400MHz
Signalart	CW AM 1kHz, 80% Modulation

Während der Prüfung wurde das Sensorsignal überwacht

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen

9.25 Störfestigkeit gegenüber niedrfrequenten Magnetfeldern

Prüfungsdefinition:

Die Prüfung wurde in Anlehnung an die Norm ISO 11452-8 durchgeführt.

Parameter	Wert
Magnetfeld	170 dB μ A/m 10Hz < f < 1kHz 170-(30*log(f/1000)) dB μ A/m 1kHz < f < 10kHz 140 dB μ A/m 10kHz < f < 150kHz

Während der Prüfung wurde das Sensorsignal überwacht

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen

9.24 Immunity against RF current on cable cluster (BCI Methode)

Test definition:

The immunity against RF current on cable clusters test was tested according following norm ISO 11452-4.

Parameter	Value
Current	200mA f < 10MHz 200mA 10MHz < f < 400MHz
Signal mode	CW AM 1kHz, 80% Modulation

During the test the sensor signal and the sensor current was monitored.

Test criteria:

The measurement data's fulfill the specified tolerances

9.25 Immunity against low frequency magnetic fields

Test definition:

The immunity against low frequency magnetic fields was tested according following norm ISO 11452-8

Parameter	Value
Magnetic field	170 dB μ A/m 10Hz < f < 1kHz 170-(30*log(f/1000)) dB μ A/m 1kHz < f < 10kHz 140 dB μ A/m 10kHz < f < 150kHz

During the test the sensor signal and the sensor current was monitored.

Test criteria:

The measurement data's fulfill the specified tolerances

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	26 von 27

Leistungsdaten Hirschmann elektronischer Gasdrehgriff
Performance data Hirschmann electronic throttle grip



C.02160 Hirschmann E-Gas 65°

EPS-100120

9.26 Messung elektromagnetisch Abstrahlung

Prüfungsdefinition:

Die Prüfung wurde in Anlehnung an die Norm CISPR_25 durchgeführt.

Parameter	Wert
Prüfschärfe	Level 5

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen

9.27 Störfestigkeit gegenüber elektrostatischer Entladung - Handlingtest

Prüfungsdefinition:

Die Prüfung wurde in Anlehnung an die Norm ISO 10605:2008-9 durchgeführt.

Parameter	Wert
Kapazität	150pF
Widerstand	2kΩ
Luftentladung	±15kV
Kontaktentladung	±4kV

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen

9.28 Störfestigkeit gegenüber elektrostatischer Entladung – Systemtest bestromt

Prüfungsdefinition:

Die Prüfung wurde in Anlehnung an die Norm ISO 10605:2008-8 durchgeführt.

Parameter	Wert
Kapazität	150pF
Widerstand	2kΩ
Luftentladung	±15kV
Kontaktentladung	±4kV
Indirekte Entladung	±4kV

Während der Prüfung wurde das Sensorsignal überwacht

Prüfungskriterium:

Die Messwerte erfüllen die spezifizierten Toleranzgrenzen

9.26 Radiated emissions

Test definition:

The immunity against low frequency magnetic fields was tested according following norm CISPR_25

Parameter	Value
Emission level	Level 5

Test criteria:

The measurement data's fulfill the specified tolerances

9.27 Immunity against electrostatic discharge - handling

Test definition:

The immunity against electrostatic discharge test was tested according following norm ISO 10605:2008-9

Parameter	Wert
capacity	150pF
resistance	2kΩ
Air discharge	±15kV
Contact discharge	±4kV

Test criteria:

The measurement data's fulfill the specified tolerances

9.28 Immunity against electrostatic discharge – powered up

Test definition:

The immunity against electrostatic discharge test was tested according following norm 10605:2008-8

Parameter	Wert
Capacity	150pF
Resistance	2kΩ
Air discharge	±15kV
Contact discharge	±4kV
Indirect discharge	±4kV

During the test the sensor signal and the sensor current was monitored.

Test criteria:

The measurement data's fulfill the specified tolerances

Dokumenten- Nummer	erstellt / geändert	Status	Seite / Seite (n)
170905_Leistungsdaten Hirschmann E-Gas Drehgriff v1.0.doc	05.09.2017	released	27 von 27

2.5 VCL Common Functions

VCL Common Functions

MANUAL

VEHICLE CONTROL LANGUAGE

Common Functions

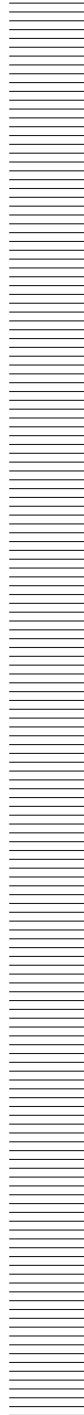
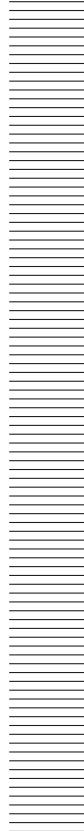
© 2005 CURTIS INSTRUMENTS, INC.

VCL Common Functions, p/n 37314
Rev. A: September 2005

**CURTIS INSTRUMENTS, INC.**
200 Kisco Avenue
Mt. Kisco, New York 10509 USA
Tel. 914.666.2971
Fax 914.666.2188
www.curjisnstruments.com

Contents

CONTENTS

1.0 OVERVIEW	1	
1.1 Application (how the functions are used)	1	
1.2 Data: Constants and Variables.....	1	
1.3 Functions.....	2	
A • MATH FUNCTIONS	5	
2.0 ABS — ABSOLUTE VALUE	6	
2.1 Using the Absolute Value Functions	6	
2.2 ABS Data.....	6	
2.3 ABS Functions	6	
automate_ABS() get_ABS()		
3.0 MTD — MULTIPLY THEN DIVIDE	7	
3.1 Using MTD	7	
3.1 MTD Data.....	7	
3.2 MTD Functions.....	7	
automate_muldiv() get_muldiv()		
4.0 RND — RANDOM NUMBER GENERATOR	9	
4.1 Using the Random Number Generator	9	
4.2 RND Data	9	
4.3 RND Functions	9	
random() random_seed()		
5.0 SCL — SCALING	10	
5.1 Using the SCL Functions	10	
5.2 SCL Data	10	
5.3 SCL Functions	10	
automate_scale() scale_value() setup_scale_factor()		
B • SIGNAL PROCESSING FUNCTIONS.....	12	
6.0 CPY — COPY ONE VARIABLE TO ANOTHER	12	
6.1 CPY Data.....	12	
6.2 CPY Data.....	12	
6.3 CPY Functions.....	12	
automate_copy() automate_block_copy() automate_limited_block_copy()		

Contents

7.0	FLT — FILTERS	15
7.1	Using Filters.....	15
7.2	FLT Data	15
7.3	FLT Function..... automate_filter()	15
8.0	LIM — LIMITS	16
8.1	Using Limits	16
8.2	LIM Data.....	16
8.3	LIM Functions..... automate_limit() get_limit() set_limit() setup_limit()	16
9.0	MAP — TWO-DIMENSIONAL MAPS	18
9.1	Using Maps.....	18
9.2	MAP Data.....	20
9.3	MAP Functions..... automate_map() get_map_output() get_map_segment() setup_map()	20
10.0	RMP — RAMPING	22
10.1	Using Ramps.....	22
10.2	RMP Data.....	24
10.3	RMP Functions..... automate_ramp() set_ramp_hold() set_ramp_rate() set_ramp_target() setup_ramp()	25
11.0	SEL — SELECTOR SWITCH	27
11.1	Using Selector Switches	27
11.2	SEL Data	27
11.3	SEL Functions..... automate_select() set_select() setup_select() automate_4p_select() set_4p_select() setup_4p_select()	28

Contents

C • TIMING FUNCTIONS	31		
12.0 DLY — TIME DELAYS	32		
12.1 Using Delays	32		
12.2 DLY Data	32		
12.3 DLY Functions	32		
setup_delay()			
setup_delay_prescale()			
13.0 TMR — TIMERS (HOURMETERS)	34		
13.1 Using Timers	34		
13.2 TMR Data	34		
13.3 TMR Functions	35		
disable_timer()			
enable_timer()			
reset_timer()			
setup_timer()			
D • CONTROL FUNCTIONS	36		
14.0 PID — PROPORTIONAL, INTEGRAL, DERIVATIVE	37		
14.1 Using PID Functions	37		
14.2 PID Data	37		
14.3 PID Functions	38		
automate_PID()			
reset_PID()			
E • MEMORY ORGANIZATION FUNCTIONS	39		
15.0 NVM — NONVOLATILE MEMORY	40		
15.1 Using Nonvolatile Memory	40		
15.2 NVM Data	44		
15.3 NVM Functions	45		
NVM_block_format()			
NVM_block_read()			
NVM_block_write()			
NVM_nvuser_restore()			
set_NVM_save_rate()			
NVM_write_parameter()			
16.0 SYS — SYSTEM-LEVEL GROUP	48		
16.1 Using the System-Level Functions	48		
16.2 SYS Data	48		
16.3 SYS Functions	49		
clear_diaghist()			
get_fault_code()			
reset_controller()			
set_sys_save_rate()			

Contents

17.0 VCL INTERPRETER	51
17.1 Using the VCL Functions.....	51
17.2 VCL Data	51
17.3 VCL Functions	52
VCL_get_size()	
VCL_get_byte()	
VCL_put_byte()	
F • COMMUNICATIONS FUNCTIONS	53
18.0 SER — SERIAL PORT CONTROL	54
18.1 Using the Serial Port Function	54
18.2 SER Data	54
18.3 SER Function.....	54
setup_serial()	
19.0 SPY — SPYGLASS SUPPORT	55
19.1 Using the Spyglass Functions.....	55
19.2 SPY Data	55
19.3 SPY Functions.....	56
put_spy_LED()	
put_spy_message()	
put_spy_text_offset()	
put_spy_timer()	
setup_spy_protocol()	
19.4 Superceded SPY Functions	58
put_spy_bin()	
put_spy_dec()	
put_spy_hex()	
put_spy_mixed()	
put_spy_text()	
G • CONTROLLER AREA NETWORK (CAN) FUNCTIONS	61
20.0 CANopen	62
20.1 Using CANopen	62
20.2 CANopen Data	62
20.3 VCL Functions Applicable to CANopen	66
enable/disable_CANopen()	
enable/disable_CANopen_emergency()	
enable/disable_CANopen_heartbeat()	
enable/disable_CANopen_nmt()	
enable/disable_CANopen_pdo()	
enable/disable_CANopen_sdo()	

Contents

21.0 VCL CAN.....	70	
21.1 Using VCL CAN	70	
21.2 VCL CAN Data.....	79	
21.3 VCL CAN Functions	83	
CAN_last_error()		
CAN_set_cyclic_rate()		
enable/disable_mailbox()		
send_mailbox()		
setup_CAN()		
setup_CAN_resync_width()		
setup_CAN_sample_delay()		
setup_CANRO_mask()		
setup_mailbox()		
setup_mailbox_byte_select()		
setup_mailbox_data()		
shutdown_CAN()		
shutdown_CAN_cyclic()		
startup_CAN()		
startup_CAN_cyclic()		
21.0 COMBINING CANOPEN & VCL CAN	90	

10 — RMP (Ramping)

SIGNAL PROCESSING FUNCTIONS

Using the ramp in normal operation

There are also functions to independently control the ramp's target value, hold control, and rate. Be aware that these functions only apply to ramps setup using the normal setup routine.

For example, if you setup the ramp with the hold control set to 0 (no-hold = movement enabled), you could use the `set_ramp_target()` function to install a new target value, which will cause the current value to immediately start moving toward the new target.

The ability to hold the ramp at a particular value is important in certain applications. For example, the speed of certain types of machines is controlled using Up and Down speed buttons. In this case, as long as the Up button is held, the machine will increase in speed until it reaches its maximum speed. As soon as the button is released, the machine's speed will no longer increase, but will remain at its current value. The hold control is very useful for this type of application. For example, you would have code to determine which button was pressed and set the target value accordingly: 0 if the Down button was pressed, 100% if the Up button was pressed. Once the target has been set, you would clear the hold control, allowing the current value to approach the target until the button was released, at which point you will again assert the hold control.

Ramp rate control is also useful in certain applications. It is sometimes the case that the rates for increasing values are different from the rates for decreasing values. This is easy to accommodate using the rate control function.

Setting up the ramp to run automatically

The automatic version of the ramp is useful in a number of situations—in particular, when setting up signal chains. The automatic version of the ramp is mainly different in that all of the parameters are controlled indirectly.

Before you setup a ramp to run automatically, you will have to decide on which user variables you will use to hold its parameters. More to the point, you will need to initialize the user variables *before* you run the automate ramp function. You must also initialize the appropriate ramp output (e.g., `RMP2_output=123`). This is very important because the output of the function will begin changing as soon as the setup function is executed. If you don't properly initialize all of the elements first, you will have an undefined output — not a good thing!

The values specified by the `automate_ramp()` function are very similar to those of the individual normal functions:

<code>RMP#</code>	ID of the ramp block to be setup to run automatically.
<code>@Target</code>	Predefined target (final) value.
<code>@Hold</code>	Predefined ramp hold control value.
<code>@Rate</code>	Predefined ramp rate value.

10 — RMP (*Ramping*)

SIGNAL PROCESSING FUNCTIONS

There are two critical differences to notice. Most obviously, the values are specified indirectly. Second, only the target value is specified (not the target and current value). This means that it is up to you to set the target equal to the current value (using `set_ramp_target()` or `setup_ramp()`) before you run the automate function.

Using the ramp in automatic operation

In automatic mode, the individual Set functions for *Target*, *Hold*, and *Rate* have no effect. It is up to you (or the other components in your signal chain) to control these elements directly (i.e., by directly altering the identified user variables). Other than that, there is no difference between the functionality offered by the normal and automatic versions of the ramp functions.

10.2 RMP DATA

CONSTANT GROUP ⇒

RMP# Block ID

These constants define the four RMP data blocks: RMP1, RMP2, RMP3, RMP4.

VARIABLE GROUP ⇒

RMP#_output Output value

These variables hold the output from the corresponding data blocks; for example, the output of RMP3 is placed in the variable `RMP3_output`.

VARIABLE GROUP ⇒

RMP#_hold Hold control

These variables hold flags for the corresponding data blocks; for example, the flag for RMP3 is placed in the variable `RMP3_hold`. A flag is used to hold the ramp at its present value. If the value is zero, the ramp is free to move. If the value is non-zero, the ramp is held at its current value.

10 — RMP (Ramping)

SIGNAL PROCESSING FUNCTIONS

10.3 RMP FUNCTIONS

FUNCTION ⇒ **automate_ramp(4)** *Setup ramping to be done automatically*

This function allows you to setup a ramp that runs automatically on each clock tick. You can specify one variable that will be the target and another that will control the ramp hold. A value of zero for a given parameter allows you to maintain program control over it (by using the setup function or by writing the variable directly). For example, to maintain local control over the target but have automatic control over the ramp hold (using whatever value is written to user37), you'd set *Target* to 0 and *Hold* to user37.

Parameters

RMP# Ramp block ID (e.g., RMP2).

@Target Ramp target (final) variable.

@Hold Ramp hold control variable.

@Rate Ramp rate variable.

Returns: 1, 0.

Reported Errors: BAD_ID, PT_RANGE.

FUNCTION ⇒ **set_ramp_hold(2)** *Set the ramp hold value*

This function installs a new value for the hold control. A value of zero enables the ramps. Any non-zero value will hold the ramp at its present value.

Parameters

RMP# Ramp block ID (e.g., RMP2).

Hold Ramp hold value (0=move, !0=hold)

Returns: 1, 0.

Reported Errors: BAD_ID, AUTO_RUN.

15 — NVM (Nonvolatile Memory)

MEMORY ORGANIZATION

E•15

NVM

NONVOLATILE MEMORY

15.1 USING NONVOLATILE MEMORY

Two sections of nonvolatile memory are accessible to VCL: the section made up of the 40 NVM user blocks, and the section containing the persistent parameters.

All but one of the functions in this group apply to the nonvolatile memory section containing the NVM user blocks; the last function applies only to the persistent parameter section. Some of the variables apply to only one or the other of these sections, while many apply to both.

Access to the nonvolatile memory is scheduled. So, unlike most operations in VCL, it will usually take some time for a given operation to complete. Consequently, you must monitor the return status of the function call to see if the request was accepted and then monitor the variable NVM_status to find out when the operation has completed. Nonvolatile memory functions generally return a non-zero result if the requested operation was actually initiated. The NVM_status value will remain non-zero until the operation completes.

For example, let us assume that you wish to read the contents of NVM4, elements 3 through 8, into the user variables user23 through user28. The proper way to do this (assuming that your program can tolerate hanging) is by writing:

```
while (NVM_block_read(NVM4, 3, 8, user23) = 0) {}
while (NVM_status <> 0) {}
```

The first line waits for the operation to be accepted. The second line waits for the operation to be completed. (A similar operation could be done in the persistent parameter section, by substituting NVM_write_parameter() for NVM_block_read() in this example.)

NVM User Block SectionRestoring the NVUser# variables

The values NVuser1 through NVuser15 can be used as you would any of the other user variables (user1 through user120). However, this 15-element block of variables is different in that it is automatically saved to nonvolatile memory periodically and on power failure. If your nonvolatile memory needs are modest (15 variables or fewer), then all you need to do is to use the NVM_nvuser_restore() function to satisfy your application's needs for nonvolatile memory.

By default, once every six minutes, the NVuser variables are saved to NVM2. This rate can be altered using the set_NVM_save_rate() function; however, care should be taken to not exceed the maximum number of writes to the EEPROM within the lifetime of the system. For the memory currently in use, the maximum

15 — NVM (Nonvolatile Memory)

MEMORY ORGANIZATION

number of writes is 1 million. Thus, one write every 6 minutes implies a service life of 100,000 hours.

When the system detects that the power has failed (see the `setup_power_fail()` routine description) the NVuser variables are saved to NVM1.

Although NVuser1 through NVuser15 are automatically saved, they are not automatically *restored*. To do this, you must execute the `NVM_nvuser_restore()` function. This function attempts to restore the NVuser variables from NVM1 first. If that fails, the function will attempt to restore the NVuser variables from NVM2.

The reason for this recovery strategy is the expectation that the save on power-failure will occasionally be unsuccessful. It is quite possible that the combination of loads on a system will cause the power to a given system to fail so quickly there is simply not enough time to write the NVuser variables. In this case, we simply fall back to the last periodic save—which, on average, will be current within 3 minutes.

Here is an example of how to restore the NVuser variables:

```

while (NVM_nvuser_restore() == 0) {}
while (NVM_status <> 0) {}

if (NVM_restore_source = NVM_RESTORED_NVM1)
{
    ; restored from NVM1
}
else if (NVM_restore_source = NVM_RESTORED_NVM2)
{
    ; restored from NVM2
}
else
{
    ; NVM_restore_source will = NVM_RESTORED_NONE (neither NVM1 nor
    NVM2 was restored)
}

```

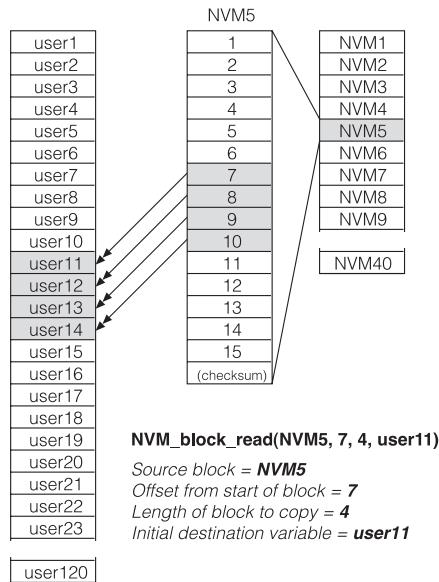
Finally, be aware that the process of restoring the NVuser variables will cause the destruction of NVM1. When the `NVM_nvuser_restore()` routine successfully recovers the information from NVM1, it intentionally corrupts the block. The behavior helps to guarantee that the most recent block (NVM1-power-fail-save-block or NVM2-periodic-save-block) is restored. Consider the following example. The system has been running for eight hours when the power fails so quickly that the system is unable to even start to save to the power-fail-save block (NVM1). If NVM1 were not intentionally corrupted, it could still be read successfully. In this case, the contents of NVM1, which are eight hours old, would be restored. The contents of NVM2 (the periodic-save block), which are only 6 minutes old, would be ignored.

Reading from and writing to NVM user blocks

The most important thing to keep in mind is that read and write operations to the nonvolatile memory take time. This means that if you request an operation, you must not assume that it has been accepted. Instead, you must test the return value of the function to make sure it was accepted.

15 — NVM (Nonvolatile Memory)

MEMORY ORGANIZATION

Reading from NVM user blocks:

Aside from timing considerations, nonvolatile reads and writes are easy. You must specify the nonvolatile block you want to access (NVM1 through NVM40), the offset within that block (0 to 15), the number of consecutive items you want to read/write (again, 0 to 15), and the initial destination variable (for reads) or initial source variable (for writes).

If your application can wait for the nonvolatile operation to complete, the code becomes quite simple (see the example in the introduction to this section). If your application cannot wait, your code will be a little more complex, depending on your needs.

For example, consider the case where you only occasionally need to write data to nonvolatile memory (i.e., you don't have to stack requests). The typical strategy is to put the nonvolatile memory access routines in their own subroutine. Within this subroutine you would have a state variable. When you need to access the nonvolatile memory, you call this routine after setting a request flag.

```
write_NVM_block:
  ; Request the operation
  if (NVM_request = 1)
  {
    if (NVM_block_write(NVM4,3,8, user23) <> 0)
    {
      NVM_request = 2
    }
  }
  ; Determine if the operation has completed
  else if (NVM_request = 2)
  {
    if (NVM_status <> 0)
    {
      NVM_request = 0
    }
  }
  return;
```

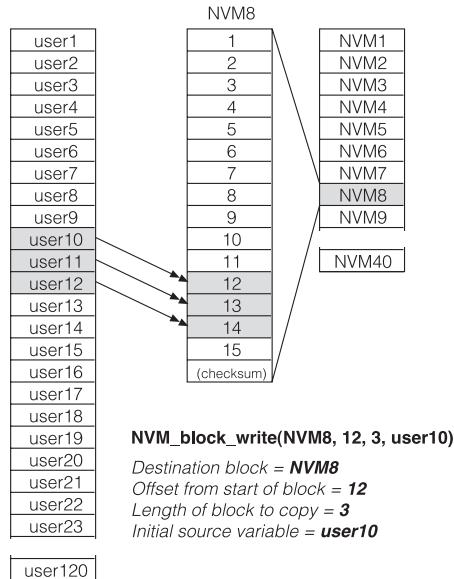
Formatting NVM user blocks

The nonvolatile memory is organized as blocks of 15 variables with a checksum. The format command operates on an entire blocks in one of two ways.

You can use the format command to initialize a block, setting all the elements to a common value and recalculating the checksum. For example, to format NVM block 3 with the value 0xFFFF (or -1 in decimal), you would write:

```
while (NVM_block_format(NVM3, 1, 0xFFFF) = 0) {}
```

Alternatively, you can use the format command in an attempt to recover a corrupted block by having it only recalculate the

Writing to NVM user blocks:

15 — NVM (Nonvolatile Memory)

MEMORY ORGANIZATION

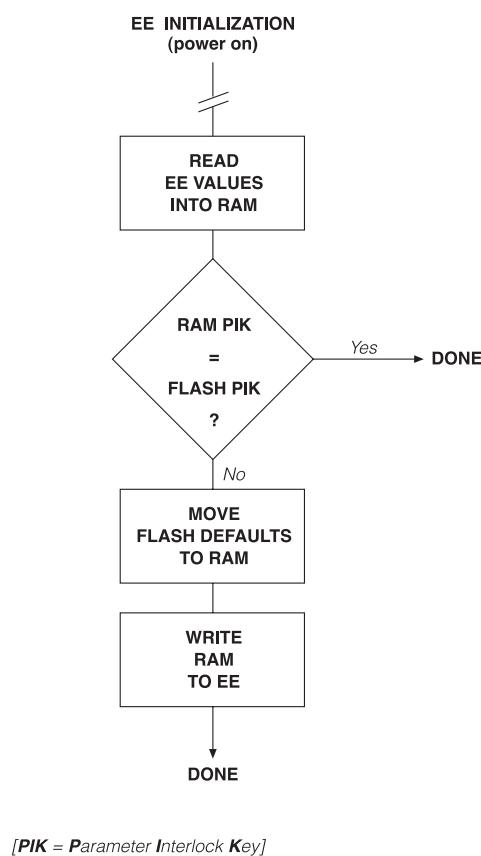
checksum. For example, to try to recover NVM block 3 without altering the contents, you would write:

```
while (NVM_block_format(NVM3, 0, 0) == 0) {}
```

Although the last parameter is ignored, it is set to zero by convention.

Persistent Parameter Section

This section of nonvolatile memory holds a variety of parameters, which have in common the characteristic of being persistent. These parameters include 1311 parameters, CAN objects, and variables that have Curtis-predefined default values.



Initialization of persistent parameters

When the nonvolatile memory (EE) is initialized on startup, the variable `parameter_interlock_key` is compared to the corresponding parameter interlock key (PIK) in Flash; if the two PIKs match, the existing EE values for the persistent parameters are maintained. If the two PIKs do *not* match, the Flash default values for the persistent parameters are written into EE, replacing the earlier non-matching values. This ensures that the persistent parameter values are the correct ones for the current OS.

The variable `parameter_defaults_restore_source` indicates the source from which the persistent parameters were restored:

DEFAULTS_FROM_EE	<i>Restored from EE</i>
DEFAULTS_FROM_FLASH	<i>Restored from Flash</i>
DEFAULTS_NOT_RESTORED	<i>Defaults were not restored.</i>

Writing persistent parameters

The function `NVM_write_parameter()` can be used to write a new value of a persistent parameter to nonvolatile memory. This function allows you to modify the value of any persistent parameter; the new value will be written from RAM to nonvolatile memory and will replace the previous value.

15 — NVM (Nonvolatile Memory)

MEMORY ORGANIZATION

15.2 NVM DATA

CONSTANT GROUP ⇒

NVM# Block ID

These constants identify the 40 NVM user data blocks: NVM1 through NVM40. The first two are predefined:

NVM1 *Holds NVuser variables upon power fail*

NVM2 *Holds NVuser variables from most recent periodic write.*

NVM3 through NVM40 are user defined.

VARIABLE ⇒

NVM_restore_source Restore source indicator

This variable shows the source of the restored values when the NVM_nvuser_restore() function is executed. The following codes are set in this variable:

NVM RESTORED_NVM1	<i>Restored from power-fail-save block (NVM1)</i>
NVM RESTORED_NVM2	<i>Restored from periodic-save block (NVM2)</i>
NVM RESTORED_NONE	<i>Unable to restore from either NVM1 or NVM2; NVuser variables are set to zero.</i>

VARIABLE ⇒

NVM_result Result of the current operation

This variable shows the result of the intrinsic nonvolatile memory access routines. These routines are used by the user nonvolatile memory access routines and the error handler, and are also used by the system function group (SYS) in the process of saving and restoring the system block.

The following codes are set in this variable:

NV_OP_ACCESS	<i>Couldn't access SPI bus</i>
NV_OP_CHECKSUM	<i>Checksum error</i>
NV_OP_MIN_KSI	<i>KSI too low to allow write</i>
NV_OP_SPI	<i>SPI Bus error</i>
NV_OP_SUCCESS	<i>Successful completion.</i>

VARIABLE ⇒

NVM_status Current NVM access status

This variable indicates the current NVM operation being performed, using the following codes:

NVM_FORMAT	<i>Format in progress</i>
NVM_IDLE	<i>No operation in progress</i>
NVM_PARAMETER	<i>Writing parameter</i>
NVM_READ	<i>Read in progress</i>
NVM_RESTORE	<i>Restore in progress</i>
NVM_WRITE	<i>Write in progress.</i>

20 — CANopen

CAN INTERFACE

E•20

CANOPEN

20.1 USING CANOPEN

CANopen provides a ready-made library of CAN constants and variables, allowing you to interface the devices in your CAN network. With these constants and variables, you can set the characteristics of CAN system devices, assess their status, and setup data transfers.

Note: On a very few specific controller models, CANopen has been disabled by default. If your controller is one of these, you will need to enable CANopen; see `suppress_CANopen_init`, page 65.

For more information on the CANopen interface, see Curtis's OEM Technical Reference, *Generic CANopen Implementation*.

20.2 CANOPEN DATA

CONSTANT GROUP ⇒

See List CANopen mailbox IDs

These constants define the CANopen mailbox assignments.

CANOPEN_EMERGENCY_MAILBOX
CANOPEN_NMT_MAILBOX
CANOPEN_HEARTBEAT_MAILBOX
CANOPEN_SDO_MISO_MAILBOX
CANOPEN_SDO莫斯I_MAILBOX
CANOPEN_PDO_MISO_1_MAILBOX
CANOPEN_PDO_MOSI_1_MAILBOX
CANOPEN_PDO_MISO_2_MAILBOX
CANOPEN_PDO_MOSI_2_MAILBOX

CONSTANT GROUP ⇒

See List CANopen mailbox masks

These constants are used with the `CAN_msg_recd`, `CAN_msg_sent`, and `CAN_response_error` variables (see page 81) to isolate a selected object. For example, you could use a mask to find out whether an NMT has been received.

CANOPEN_EMERGENCY_MASK
CANOPEN_NMT_MASK
CANOPEN_HEARTBEAT_MASK
CANOPEN_SDO_MISO_MASK
CANOPEN_SDO莫斯I_MASK
CANOPEN_PDO_MISO_1_MASK
CANOPEN_PDO_MOSI_1_MASK
CANOPEN_PDO_MISO_2_MASK
CANOPEN_PDO_MOSI_2_MASK

Example: if (CAN_msg_recd & CANOPEN_NMT_MASK) >> 0, the NMT was received.

20 — CANopen

CAN INTERFACE

VARIABLE GROUP ⇒

See List CANopen PDO MISO # specification and mapping variables

These variables specify and map PDO MISO 1 and PDO MISO 2 objects.

CAN_pdo_miso_#_map_errors structure	<i>Bit array indicating mapping errors</i>
CAN_pdo_miso_#_struct_length	<i>Number of PDO MISO structure variables (= 1)</i>
CAN_pdo_miso_#_cob_ID	<i>PDO MISO COB ID</i>
CAN_pdo_miso_#_length	<i>PDO MISO byte length (max=8)</i>
CAN_pdo_miso_#_map_1	<i>PDO object ID 1</i>
CAN_pdo_miso_#_map_2	" " " 2
CAN_pdo_miso_#_map_3	" " " 3
CAN_pdo_miso_#_map_4	" " " 4
CAN_pdo_miso_#_map_5	" " " 5
CAN_pdo_miso_#_map_6	" " " 6
CAN_pdo_miso_#_map_7	" " " 7
CAN_pdo_miso_#_map_8	" " " 8

VARIABLE GROUP ⇒

See List CANopen PDO MOSI # specification and mapping variables

These variables specify and map PDO MOSI 1 and PDO MOSI 2 objects.

CAN_pdo_mosi_#_map_errors structure	<i>Bit array indicating mapping errors</i>
CAN_pdo_mosi_#_struct_length	<i>Number of PDO MOSI structure variables (= 1)</i>
CAN_pdo_mosi_#_cob_ID	<i>PDO MOSI COB ID</i>
CAN_pdo_mosi_#_length	<i>PDO MOSI byte length (max=8)</i>
CAN_pdo_mosi_#_map_1	<i>PDO object ID 1</i>
CAN_pdo_mosi_#_map_2	" " " 2
CAN_pdo_mosi_#_map_3	" " " 3
CAN_pdo_mosi_#_map_4	" " " 4
CAN_pdo_mosi_#_map_5	" " " 5
CAN_pdo_mosi_#_map_6	" " " 6
CAN_pdo_mosi_#_map_7	" " " 7
CAN_pdo_mosi_#_map_8	" " " 8

20 — CANopen

CAN INTERFACE

VARIABLE GROUP ⇒

See List CANopen general status variables

These read-only variables indicate message/process timing.

CANopen_system_status_flags Returns the following flags:

CANOPEN_NMT_ENABLED

CANOPEN_SDO_ENABLED

CANOPEN_PDO_ENABLED

CANOPEN_HEARTBEAT_ENABLED

CANOPEN_EMERGENCY_ENABLED

CANOPEN_ENABLED

For example, if (CANopen_system_status_flags & CANOPEN_SDO_ENABLED) ><0, the SDOs are enabled.

CANopen_bus_cycle_time CANopen processing rate (in ms)**CAN_pdo_response_period** Max response time to PDO (in CANopen_bus_cycle_time units)**CAN_nmt_state** 0=initialization, 4=stopped, 5=operational, 127=pre-operational

VARIABLE GROUP ⇒

See List CANopen general control variables

The rates and periods in this group are in CANopen_bus_cycle_time increments; the value of CANopen_bus_cycle_time is in milliseconds. Thus if CANopen_bus_cycle_time is set at 4 ms and CANopen_heart_beat_rate is set at 5, heartbeats will be produced every 20 ms.

CANopen_heart_beat_rate Sets heartbeat rate**CANopen_emergency_rate** Sets emergency rate**CAN_pdo_timeout_period** Sets PDO timeout period

VARIABLE GROUP ⇒

See List CANopen SDO block transfer variables

The variables in this group are used to initiate and control SDO block transfers.

start_CAN_flash_download Initiates a flash download (OS or APP)**start_CAN_flash_upload_os** Initiates a flash OS upload**start_CAN_flash_upload_app** Initiates a flash APP upload

20 — CANopen

CAN INTERFACE

VARIABLE GROUP ⇒

See List CANopen EE write status and control variables

These variables are related to the interaction between CANopen objects and nonvolatile memory.

CANopen_ee_status_flags

Indicates the status of the EE write, using the following flags:

CANOPEN_EE_WRITE_IN_PROGRESS

CANOPEN_EE_WRITE_OVERFLOW

CAN_ee_writes_enabled

To enable automatic EE write of a CANopen object, set this variable to a non-zero value

VARIABLE ⇒

suppress_CANopen_init CANopen suppression variable

To disable CANopen initialization, set this variable to:

SUPPRESS_CANOPEN_KEY

If your controller has CANopen disabled by default and you want to run CANopen, you will have to change the setting of this variable. The most direct way to do this is by including the following parameter block definition in your VCL application:

```
; parameter_entry      "Suppress CANopen Init"
;   type                no_display
;   address             suppress_canopen_init
;   default             0
; end
```

When the VCL is built and downloaded, this will permanently change the nonvolatile memory's default value from CANopen disabled to CANopen enabled.

20 — CANopen

CAN INTERFACE

20.3 VCL FUNCTIONS APPLICABLE TO CANOPEN

Although CANopen runs independent of VCL, you may want to use VCL to limit its scope of operation. You can execute the VCL functions disable/enable_CANopen()—or individually execute the disable/enable functions for individual CANopen components (emergency, heartbeat, nmt, pdo, sdo).

FUNCTION ⇒ **enable_CANopen(0)** *Enable all CANopen processing*

This function enables all CANopen processing.

Parameters: none.

Returns: 1.

Reported Errors: none.

FUNCTION ⇒ **disable_CANopen(0)** *Disable all CANopen processing*

This function disables all CANopen processing.

Parameters: none.

Returns: 1.

Reported Errors: none.

FUNCTION ⇒ **enable_CANopen_emergency(0)** *Enable CANopen emergency processing*

This function enables CANopen emergency processing by setting up the following mailbox:

CANOPEN_EMERGENCY_MAILBOX.

Parameters: none.

Normal Returns: 1, 0.

Reported Errors: none.

FUNCTION ⇒ **disable_CANopen_emergency(0)** *Disable CANopen emergency mailbox*

This function disables CANopen emergency mailbox processing by releasing the following mailbox:

CANOPEN_EMERGENCY_MAILBOX.

Parameters: none.

Returns: 1.

Reported Errors: none.

20 — CANopen

CAN INTERFACE

FUNCTION ⇒ **enable_CANopen_heartbeat(0)** *Enable CANopen heartbeat processing*

This function enables CANopen heartbeat processing:
CANOPEN_HEARTBEAT_MAILBOX.

Parameters: none.

Normal Returns: 1, 0.

Reported Errors: none.

FUNCTION ⇒ **disable_CANopen_heartbeat(0)** *Disable CANopen standard heartbeat mailbox*

This function releases the following mailbox from CANopen heartbeat processing:
CANOPEN_HEARTBEAT_MAILBOX.

Parameters: none.

Normal Returns: 1.

Reported Errors: none.

FUNCTION ⇒ **enable_CANopen_nmt(0)** *Enable CANopen NMT processing*

This function enables CANopen NMT processing in the following mailbox:
CANOPEN_NMT_MAILBOX.

Parameters: none.

Normal Returns: 1, 0.

Reported Errors: none.

FUNCTION ⇒ **disable_CANopen_nmt(0)** *Disable CANopen standard NMT mailbox*

This function releases the following mailbox from CANopen NMT processing:
CANOPEN_NMT_MAILBOX.

Parameters: none.

Normal Returns: 1.

Reported Errors: none.

20 — CANopen

CAN INTERFACE

FUNCTION ⇒ **enable_CANopen_pdo(0)** *Setup CANopen PDO processing*

This function sets up the standard CANopen PDO processing in these four mailboxes:

CANOPEN_PDO_MISO_1_MAILBOX
CANOPEN_PDO莫斯I_1_MAILBOX
CANOPEN_PDO_MISO_2_MAILBOX
CANOPEN_PDO莫斯I_2_MAILBOX.

Parameters: none.

Normal Returns: 1, 0.

Reported Errors: none.

FUNCTION ⇒ **disable_CANopen_pdo(0)** *Disable all CANopen standard PDO mailboxes*

This function releases these four mailboxes from CANopen PDO processing:

CANOPEN_PDO_MISO_1_MAILBOX
CANOPEN_PDO莫斯I_1_MAILBOX
CANOPEN_PDO_MISO_2_MAILBOX
CANOPEN_PDO莫斯I_2_MAILBOX.

Parameters: none.

Normal Returns: 1.

Reported Errors: none.

20 — CANopen

CAN INTERFACE

FUNCTION ⇒ **enable_CANopen_sdo(0)** *Enable CANopen SDO processing*

This function enables CANopen SDO processing, by allocating these two mailboxes to CANopen SDO processing:

CANOPEN_SDO_MISO_MAILBOX
CANOPEN_SDO莫斯I_MAILBOX.

Parameters: none.

Normal Returns: 1, 0.

Reported Errors: none.

FUNCTION ⇒ **disable_CANopen_sdo(0)** *Disable CANopen standard SDO mailboxes*

This function releases these two mailboxes from CANopen SDO processing:

CANOPEN_SDO_MISO_MAILBOX
CANOPEN_SDO莫斯I_MAILBOX.

Parameters: none.

Normal Returns: 1.

Reported Errors: none.

E•21

VCL CAN

21.1 USING VCL CAN

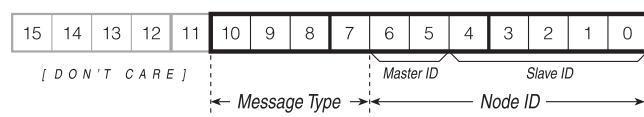
As an alternative to the ready-made CANopen interface, you can create your own customized CAN interface using the constants, variables, and functions of VCL CAN. (We assume CANopen is disabled (see `suppress_CANopen_init`, page 65.)

Section 21.1 is in four parts: (a) COB-ID structure, (b) a brief introduction to the key VCL CAN variables, (c) how to use the core VCL CAN functions, and (d) a CAN system example.

COB-ID Structure

Each message has an 11-bit message identifier, called a COB-ID (Communication Object ID). The most common COB-ID structure consists of three fields. The upper four bits hold the message type. The next two bits hold the ID of the master device. The last five bits hold the ID of the slave device.

11-bit message identifier field:



Alternative allocation and assignment of ID bits is possible, but you may require Curtis support and modification if your node IDs are to be accessible or programmable using a Curtis 1311 programmer where maximum values must be defined.

Using VCL CAN Variables

These two VCL CAN variables provide status information:

CAN_bus_on	CAN Bus Status (0=off, non-0=on)
CAN_error_status	Bit0 = Tx or Rx error count > 96, Bit1 = CAN bus Off.

These three VCL CAN variables provide information about the status of a given mailbox; examples of bit definitions are given in parentheses:

CAN_response_error	Response timeout flags (bit=CAN1_TIMEOUT).
CAN_msg_sent	Sent flags (bit=CAN1_SENT).
CAN_msg_recd	Received flags (bit=CAN1_RECEIVED).

21 — VCL CAN

CAN INTERFACE

The CAN_response_error variable holds fourteen bit-addressable timeout flags (CAN1_TIMEOUT through CAN14_TIMEOUT) that are set to 1 when a CAN response is not received within the response timeout period set by the setup_mailbox() function. For example, assume you setup a mailbox with the following parameters:

```
Setup_MailBox(CAN1, PDO_MOSI, MasterID, SlaveID, C_CYCLIC,
C_XMT, 8, CAN2)
```

This mailbox will be sent every 20ms (assuming the cyclic clock remains set to its default value). When the message is sent, a timer is started. If a response message doesn't arrive in the CAN2 mailbox within 32ms, a CAN1_TIMEOUT flag will be set.

Similarly, the CAN_msg_sent variable holds fourteen bit-addressable flags (CAN1_SENT through CAN14_SENT) that are set each time the corresponding mailbox is sent. These flags are very useful if you want to send messages manually (i.e., using the send_mailbox() function) as rapidly as possible. There are no queues or stacks for CAN buffers; consequently, you must either wait at least 4ms (the service time for the CAN subsystem) or wait on the CAN#_SENT flag for that mailbox to guarantee that you don't overwrite a message before it's sent.

There are two warnings about using the CAN_msg_sent flags. First, these flags are only set by the system, never cleared; you must therefore remember to clear the flag before you use it. For example, you might write:

```
CAN3_SENT = OFF
send_mailbox(CAN3)
while (CAN3_SENT == OFF) {}
```

This clears the message-sent flag, tells the system that you want to send the contents of mailbox 3, and then waits until the system sets the flag saying that mailbox 3 has actually been sent.

A second warning concerning the use the message sent flags is that many CAN nodes may only be able to process messages at a limited rate. For example, the 1243 CAN controller will only process CAN messages every 20ms. In other words, sending messages more frequently than every 20ms to a 1243 CAN controller will result in lost messages.

The CAN_msg_recd variable also holds a group of twelve flags, indicating that a message has been received in the specified mailbox. As with the CAN_msg_sent flags, CAN_msg_recd flags are only set by the system, never cleared.

21 — VCL CAN

CAN INTERFACE

Using VCL CAN Functions• **setup_CAN(5)**

The first VCL CAN command that should be issued is the `setup_CAN()` function. It initializes the CAN system and allows you to tailor its operation according to the following parameters:

<i>Baud</i>	Holds the index <code>baud_rate</code> (<code>CAN_125KBAUD</code> , <code>CAN_250KBAUD</code> , <code>CAN_500KBAUD</code>).
<i>Sync</i>	Holds the sync period (0=off, n=number of 4ms ticks between syncs).
<i>Reserved</i>	Not currently used (set to 0).
<i>Reserved</i>	Not currently used (set to 0).
<i>Restart</i>	CAN bus auto-restart after bus-off (0=yes, !0=no).

The *Baud* rate must, of course, be the same for all elements on the CAN bus.

The *Sync* period refers to one of two periodic rates available to schedule mailboxes to be sent automatically (the other rate is the cyclic period). The default rate for the sync period is the same as for the cyclic period: 20ms (i.e., `Sync=5`).

The *Restart* parameter determines what the CAN subsystem does when it encounters bus errors. If set to 0, the CAN subsystem will automatically try to restart the CAN bus in the event of a bus error. If set to non-zero, the CAN subsystem will not try to restart the bus (i.e., the CAN bus will become inactive if an error is encountered) and will set a flag indicating a CAN bus error.

• **setup_mailbox(8)**

This function sets up the initial parameters for a mailbox. **This function must be called before defining the data for the mailbox** (see `setup_mailbox_data`) or trying to use the mailbox. This function accepts the following parameters:

<i>CAN#</i>	Mailbox block ID (e.g., CAN3).
<i>Message Type</i>	Type of message (e.g., PDO_MOSI) goes into address bits 7-10.
<i>Master ID</i>	Master ID goes into address bits 5-6.
<i>Slave ID</i>	Slave ID goes into address bits 0-4.
<i>Service</i>	When to send this message (<code>C_EVENT</code> , <code>C_CYCLIC</code> , or <code>C_SYNC</code>).
<i>Direction</i>	Data direction (<code>C_RCV</code> or <code>C_XMT</code>).
<i>Timeout</i>	Replay timeout in 4ms increments (e.g., 5 = 20ms timeout).
<i>Reply</i>	Handle of reply buffer (e.g., CAN2).

21 — VCL CAN

CAN INTERFACE

CAN# refers to the predefined ID of the mailbox you wish to setup. There are presently fourteen mailboxes (CAN1 through CAN14) available. All mailboxes behave identically.

The next three parameters—*Message Type*, *Master*, and *Slave*—are combined to make a single 11-bit identifier, as shown on page 70.

The 4-bit *Message Type* field determines both how the message will be processed and its priority. The message types are listed below in priority order:

NMT	Network Management Transmission				
SYNC_ERR	Both Sync (ID=0) & Error (ID = Error) messages				
PDO_MISO	Process Data Object (PDO), Master In Slave Out				
PDO_MOSI	"	"	"	"	Master Out Slave In
PDO_SYS1	"	"	"	"	System Broadcast 1
PDO_SYS2	"	"	"	"	System Broadcast 2
PDO_SYS3	"	"	"	"	System Broadcast 3
PDO_SYS4	"	"	"	"	System Broadcast 4
SDO_MISO	Service Data Object (SDO), Master In Slave Out				
SDO_MOSI	"	"	"	"	Master Out Slave In
NODE	Node Guard Message				

As you can see there are two groups of types (PDO and SDO) plus three unique message types (NMT, SYNC_ERR, and NODE). In the most general sense, the types simply define the priority level with NMT being the highest, and NODE being the lowest. However, according to common agreement, each group and each unique type has a specific use.

In many of these messages, you see MOSI and MISO appended to the basic message type (e.g., PDO_MOSI). These tags reinforce the intended flow of data on the CAN bus. The master transmits a PDO_MOSI (Master Out, Slave In), while the slave receives the PDO_MOSI. The slave may then respond with a PDO_MISO.

NMT: The NMT message type is used to control bus activity and initiate certain actions within nodes connected to the bus. For example, you would use an NMT message to bring a Curtis controller's CAN system online.

SYNC_ERR: This type is used for two purposes. Receipt of a SYNC message should cause you to send sync information (e.g., information in response to a master clock/requestor). ERROR messages are used to broadcast the advent of an error within a controller to other bus elements.

PDO: A Process Data Object is a collection of CAN objects; usually these are parameters that reflect the changing state of the machine. Because of

21 — VCL CAN

CAN INTERFACE

the rigorous error checking in a CAN message, the data portion of the message is usually 50% or less. PDOs allow up to 8 CAN objects to be sent all at once. The destination of the data (the index or address) for each element of a PDO is pre-defined at design time; consequently, all that must be sent is the data. It is often the case that receipt of a PDO carries the expectation that a PDO will be sent in response. For example, the default configuration will cause a PDO_MISO and a PDO_SYS1 to be sent in response to each PDO_MOSI sent.

SDO: A **Service Data Object** has a format that is identical to that of a PDO; however, SDOs generally refer to parameters within the machine that do not change as the machine runs (i.e., system constants). Often, SDOs refer to data that is stored in nonvolatile memory.

NODE: The node type is reserved for automatic responses in response to a node-guarding message, usually sent out at the sync frequency. These messages are used to guarantee that nodes are still available.

The 2-bit *Master ID* field is the next part of the ID field. This identifies which of the four possible bus masters is generating the message.

The final part of the ID field is the *Slave ID* field, which identifies the intended recipient of the message.

VCL uses all three fields in the 11-bit identifier. For most cases, you will enter a value for each field: i.e., (. . . message type, master id, slave id, . . .). The Message Type field is shifted automatically into the top 4 bits. The Master ID field is shifted into a 2-bit field just below the Message Type, and the Slave ID field is placed in the lowest 5 bits. Both the Message Type and Master ID fields are masked before they are shifted into place; however, the Slave ID field is simply or'd into the final identifier. This allows you to set both the Message Type and Master ID fields to 0 and then place a pre-built 11-bit identifier into the Slave ID field: i.e., (. . . 0, 0, cob-id, . . .). This is necessary to implement unique COB-IDs (communication object IDs) and can also be used to accommodate other protocol ID strategies.

The *Service* parameter is used by the system to determine when to send the data in the mailbox. There are three possibilities:

C_SYNC	Send on sync tick
C_CYCLIC	" " clock tick
C_EVENT	" " event

As mentioned previously, there are two periodic clocks in the CAN subsystem, the cyclic clock and the sync clock. These clocks can be set independently (the default for both is 20ms). Usually, periodic events are scheduled using the cyclic clock (C_CYCLIC), with the sync clock being reserved for node-guarding. Node-guarding refers to the practice of sending a message out to each node in

21 — VCL CAN

CAN INTERFACE

the system and then checking that the node responds (i.e., to see if a node has failed). This use of the two periodic clocks is the convention; however, it is not enforced by the VCL implementation. You can, if you wish, use the sync clock as a second cyclic clock.

The C_EVENT flag is used to tell the system that the data in the mailbox is to be sent on demand, using the send_mailbox() command.

The *Direction* parameter tells the system whether the data in this mailbox is to be transmitted (C_XMT) or received (C_RCV).

The *Timeout* and *Reply* parameters are used together; their interpretation depends on the direction flag (C_XMT or C_RCV).

If the data direction is *out* (C_XMT), you can tell the system to expect a reply within a certain period of time. Set the timeout parameter equal to the maximum time (in 4ms increments) that you're willing to wait for the reply. Then set the reply parameter to the mailbox ID (e.g., CAN2) in which you expect the reply to be received.

If, on the other hand, the data direction is *in* (C_RCV), you can set the reply field to the ID of a mailbox that is to be automatically transmitted in response to the message you've just received. In this case, set the timeout field to 0.

If neither of the two previous conditions are of interest to you (you don't demand a response and you don't automatically reply), then set the timeout parameter and reply parameter both to 0.

- **setup_mailbox_data(10)**

This function is used to define the data pointers for a given mailbox. Remember: The setup_mailbox function() must have been called before defining the data with this function. This function takes the following parameters:

<i>CAN#</i>	Holds the ID of the mailbox (e.g., CAN4).
<i>Count</i>	Holds the number of bytes actively used by this mailbox.
<i>@Data1</i>	Address of byte 1's value.
<i>@Data2</i>	Address of byte 2's value.
<i>@Data3</i>	Address of byte 3's value.
<i>@Data4</i>	Address of byte 4's value.
<i>@Data5</i>	Address of byte 5's value.
<i>@Data6</i>	Address of byte 6's value.
<i>@Data7</i>	Address of byte 7's value.
<i>@Data8</i>	Address of byte 8's value.

CAN# is the same predefined mailbox identifier that you used in the setup_mailbox() function call (e.g., CAN2)

21 — VCL CAN

CAN INTERFACE

The *Count* parameter tells the system how many bytes you will be sending/receiving with this mailbox. Depending on the format, the message length can be anything from 0 to 8 bytes of data.

Remember that all variables within VCL are 16-bits long; however, the mailbox data pointers refer to 8-bit values. If you just give the name of a variable (e.g., DLY1_output) you will be referring to the lower 8 bits of that variable. If you want to refer to the upper 8 bits, you must use the USEHB operator. To reference the upper 8 bits of DLY1_output, you would write @DLY1_output+USEHB.

Note: In most cases, you can safely omit the address operator “@” in your code; the program automatically inserts the operator where appropriate. However, you **must** type the address operator whenever an expression is involved (e.g., @DLY1_output+USEHB).

Because all VCL functions have fixed length parameter lists, you must always supply values for them all. By convention, any unused parameters should be set to 0.

- **startup_CAN(0)**

When the system first starts up, the CAN system is disabled. Disabled in this sense means that the subsystem does not send any messages, nor does it check if any messages have been received.

Generally, you will execute the setup_CAN() function first, followed by one or more pairs of setup_mailbox() and setup_mailbox_data() functions. Once you have setup all the mailboxes, you can startup the CAN system by using the startup_CAN() function.

- **startup_CAN_cyclic(0)**

Executing the startup_CAN() function allows the CAN subsystem to send messages on demand and begin automatically checking for messages received; however, it does not enable automatic (cyclic or sync) message transmissions. To do this, you must execute the startup_CAN_cyclic() function.

The reason that we don't start up both the CAN system and its cyclic functions has to do with the way we normally use the system. It is often the case that after you've setup the CAN system (setup_CAN()) and configured all your mailboxes (setup_mailbox() and setup_mailbox_data()) you will need to configure other elements of the CAN network before you begin full automatic operation. For example, you may have to send one or more NMT messages to startup CAN services on another device. Or you may have to send a series of SDOs to make sure that another device on the bus is properly configured. Initiating cyclic messaging at this point would, at best, delay these operations (because of the increased bus traffic).

21 — VCL CAN

CAN INTERFACE

Consequently, you will often see the startup_CAN() function call followed by one or more NMT messages (or even a call to an extensive set of SDO message invocations) before you see the startup_CAN_cyclic() function call.

- **send_mailbox(1)**

This function forces the mailbox contents to be sent. Its only parameter is the block ID of the mailbox you are sending (e.g., CAN4). You will need to use this function for any mailbox that has a service field set to C_EVENT (as opposed to C_CYCLIC or C_SYNC).

CAN System Example

This code shows how to setup the CAN buffers to send SDOs on demand. This requires five CAN-related function calls: setup_CAN(), setup_mailbox(), setup_mailbox_data(), startup_CAN(), and send_mailbox(). We also discuss the setup_CAN_cyclic() function.

The definitions section

In this section, we define all user variable synonyms and constants. In this case, we need two variables: one to hold the ID part of the SDO and one to hold the data part of the SDO. As you can see, these have been put into user1 and user2.

We also need to define the Master and Slave IDs. This is done using constant definitions. Defining these values once, using a constant, makes it much easier to modify these values should the need arise.

```

;=====
; CAN System Demonstration
;=====
;
; -This code demonstrates how to setup the CAN system to send SDOs on
; demand.
;
;-----
; Local Definitions
;-----
;
; System Constants
;-----
;
; CAN Bus IDs
MASTER_ID    constant 2      ; ID of the master unit
SLAVE_ID     constant 5      ; ID of the slave unit
;
; PDO Variables
;-----
;
; These variables are used when sending and receiving SDOs.
sdo_id       equals user1   ; Holds the ID when sending SDOs
sdo_data     equals user2   ; Holds the data to be sent
;
sdo_in_id    equals user1   ; Holds the ID when receiving SDOs
sdo_in_data  equals user2   ; Holds the data received

```

21 — VCL CAN

CAN INTERFACE

The one-time setup section

This is the classic initialization section. This code is executed once, before we fall into the main loop.

```
;-----
; One Time Setup
;-----
; -In this section of code we do all one-time initializations.

call Startup_CAN_System      ; Start CAN System Messaging
user3 = 0                     ; User3 triggers sending the mailbox
```

The main loop

The main loop code is continuously executed. If user3's value becomes non-zero, mailbox 6 is sent (explicitly, using the send_mailbox() function).

```
;-----
; Main Program Loop
;-----
; -Each time user3 is set, the mailbox will be sent. You can examine
; the CAN_msg_sent variable to see that the bit corresponding to CAN6
; has been set.

; main_loop:
    if (user3 <> 0)
    {
        user3 = 0
        send_mailbox(CAN6)
    }
goto main_loop
```

System support subroutines

There is only one support routine in this example: the CAN initialization subroutine that is called in the one-time setup section.

```
;-----
; Startup_CAN_System - Startup the CAN subsystem
;-----
; -This routine starts up the CAN system. We send out NMTs and SDOs
; here to initially configure the system.
;
Startup_CAN_System:

setup_can(CAN_125KBAUD,0,0,0,1) ;125kb,no sync,null,null,no auto-
restart

; Setup an SDO mailbox so we can Send Data
setup_mailbox(CAN6,SDO_MOSI,MASTER_ID,SLAVE_ID,C_EVENT,C_XMT,0,0
@sdo_id+USEHB,
sdo_id,
@sdo_data+USEHB,
sdo_data,
0,0,0,0)

; Setup an SDO mailbox so we can Receive Data
setup_mailbox(CAN7,SDO_MISO,MASTER_ID,SLAVE_ID,C_EVENT,C_RCV,0,0
@sdo_in_id+USEHB,
sdo_in_id,
@sdo_in_data+USEHB,
sdo_in_data,
0,0,0,0)

; Start CAN Messaging (non-cyclic only)
startup_can()

return
```

21.2 VCL CAN DATA

CONSTANT GROUP ⇒

CAN# Block ID

These constants identify the fourteen CAN mailboxes: CAN1 to CAN14. (The fifteenth mailbox is reserved as read-only: “CANRO”.)

CONSTANT GROUP ⇒

See List Message type specifier

These constants are used in the process of setting up a mailbox; they are listed in priority order.

NMT	<i>Network Management</i>		
SYNC_ERR	<i>Both Sync (ID=0) and Error (ID=Error)</i>		
PDO_MISO	Process Data Object	(Master In Slave Out)	
PDO莫斯I	” ” ”	(Master Out Slave In)	
PDO_SYS1	” ” ”	(System Broadcast 1)	
PDO_SYS2	” ” ”	(System Broadcast 2)	
PDO_SYS3	” ” ”	(System Broadcast 3)	
PDO_SYS4	” ” ”	(System Broadcast 4)	
SDO_MISO	Service Data Object	(Master In Slave Out)	
SDO莫斯I	” ” ”	(Master Out Slave In)	
NODE	<i>Node Guard Message</i>		

CONSTANT GROUP ⇒

See List Message execution schedule

These constants are used to schedule transmission of messages.

C_SYNC	<i>Send on sync tick</i>
C_CYCLIC	” ” <i>clock tick</i>
C_EVENT	” ” <i>event</i>

CONSTANT GROUP ⇒

See List Transfer direction

These values are used to set the direction of transfer (see `setup_mailbox()`).

C_RCV	<i>Receive bit</i>
C_XMT	<i>Transmit bit</i>

CONSTANT ⇒

USEHB Use the high byte of a variable

This is used in conjunction with the `setup_mailbox_data()` function to tell the system to use the high byte of a variable. When using `setup_mailbox_data()`, you must specify each byte of data to be transmitted. By default, the system will access the low byte of the word specified (each word is 16 bits). Using this constant, you can tell the system that you want to use the high byte instead; see page 87.

21 — VCL CAN

CAN INTERFACE

VARIABLE \Rightarrow **CAN_baud_rate** *Data transmission rate*

This variable sets the effective baud rate, using the following constants:

CAN_125KBAUD

CAN_250KBAUD

CAN_500KBAUD.

VARIABLE \Rightarrow **CAN_bus_on** *Bus status*

The CAN_bus_on variable has two values, 0 and 1. The value is zero until the startup_CAN() routine is run, at which point CAN_bus_on is set to 1.

If the CAN bus ever goes Off (i.e., CAN_error_status = 3), CAN_bus_on is reset to 0. CAN_bus_on will stay at 0 until the next time startup_CAN() is called. Because of this, if you leave the CAN Bus auto-restart enabled (see setup_CAN()) you can still use CAN_bus_on to tell whether the bus was ever forced off.

VARIABLE \Rightarrow **CAN_error_mailbox** *Bus error mailbox ID*

This variable provides the mailbox ID at the time the CAN bus error occurred. A value of -1 indicates “no mailbox”.

VARIABLE \Rightarrow **CAN_error_status** *Bus error status*

This variable is used to provide information about the number of errors and the current status of the CAN bus. The variable's two bits have the following values:

bit 0 If non-zero, there have been more than 96 transmit or receive errors.

bit 1 If non-zero, the CAN bus has been automatically turned off.

This yields the following status values for CAN_error_status:

0 = No error.

1 = There have been more than 96 transmit or receive errors.

3 = The CAN bus is Off (there have been more than 256 errors).

VARIABLE \Rightarrow **CAN#_length** *Length of last message*

This variable holds the number of data bytes received in the indicated mailbox. For example, CAN4_length holds the number of data bytes received in the CAN4 mailbox.

21 — VCL CAN

CAN INTERFACE

BIT-VARIABLE ⇒ **CAN_msg_recd** *Message received flags*

This variable is used to indicate that a message has been received in the specified mailbox. Each of these bits has a predefined bit constant consisting of the block ID plus the word “received”:

`CAN#_RECEIVED`

For example, the message-received bit definition for CAN5 is `CAN5_RECEIVED`.

Note: Flags are set—but not cleared—by the system. It is up to you to clear them.

BIT-VARIABLE ⇒ **CAN_msg_sent** *Message transmitted flags*

This variable is used to indicate that a message has been sent. CAN messages are sent out periodically rather than instantly; a message may be sent later than the time your program specifies the message is to be sent. This delay could be a problem when you need to send a number of messages in sequence; however, using this variable, you can “loop” until the Sent flag for the buffer of interest is set, thus avoiding data overruns.

Each of these bits has a predefined bit constant consisting of the block ID plus the word “SENT”:

`CAN#_SENT`

For example, the message-sent bit definition for CAN5 is `CAN5_SENT`.

Note: Flags are set—but not cleared—by the system. It is up to you to clear them before issuing new Send commands.

BIT-VARIABLE ⇒ **CAN_response_error** *Response timeout flags*

This variable is used to indicate that there was a message response timeout. When you setup a mailbox, you can also specify that you expect a response within a certain time period (see the `setup_mailbox()` function’s reply and timeout parameters). If the response is not received within the specified time period, the bit corresponding to the mailbox is set.

Each of these bits has a predefined bit constant consisting of the block ID plus the word “timeout”:

`CAN#_TIMEOUT`

For example, the timeout bit definition for CAN5 is `CAN5_TIMEOUT`.

Note: Flags are set—but not cleared—by the system. It is up to you to clear them.

Abbildungsverzeichnis

II.1 Topologiebild 1	6
II.2 Topologiebild 2	7
III.1 Solid Edge Logo	16
III.2 Solid Edge Volumenkörper	16
III.3 Solid Edge Bemaßen	16
III.4 Solid Edge Strukturelle Lasten	16
III.5 Solid Edge Vernetzung	16
III.6 Solid Edge Animation	17
III.7 Solid Edge Zeichnungsansichten	17
III.8 Solid Edge Zusammenbau	17
III.9 Grundaufbau einer Lithium-Ionen Zelle [[GrundaufbauLiionenZelle]]	22
III.10Grundschaltung eines Batteriemanagementsystems[[GrundschaltungBMS]]	25
III.11Läufer einer dauer magnet erregten Synchronmaschine [Fischer]	28
III.12Ausgabe eines Sinus/Cosinus-Sensors [Manual]	29
III.13Übersicht der häufigsten Bauformen eines Curtis Controllers [Manual]	30
III.14Kommutierung mittels Trapezsteuerung [Feldorientierte-Antriebssteuerung]	31
III.15Kommutierung mittels Vektorregelung [Feldorientierte-Antriebssteuerung]	31
III.16Das Blockschaltbild eines allgemeinen Regelkreises [AUT5]	33
III.17Zusammensetzung des PID-Reglers [Regelungstechnik]	35
III.18Vergleich der verschiedenen Reglertypen [PID-Regler]	35
III.19Aufbau einer CAN-Bus Leitung	36
III.20Aufbau einer SPI-Bus Verbindung	36
IV.1 Seitenplatte Rechts	40
IV.2 Seitenplatte Links Innenansicht	40
IV.3 Wellenersatz	40
IV.4 Getriebeansicht Seitlich	42
IV.5 Getriebeansicht Schräg	42
IV.6 Achse 1/Antriebsachse	43
IV.7 Achse 2	43
IV.8 Achse 3	44
IV.9 Getriebegegenplatte	44
IV.10Abstandhalter	44
IV.11Kettenspanner	45
IV.12ISO 08 B-1	46
IV.13ISO 10 B-1	47
IV.14Kettenrad Z=15 B-1 08	48
IV.15Kettenrad Z=15 B-1 10	48
IV.16Kettenrad Z=16 B-1 08	48
IV.17Kettenrad Z=30 B-1 08	48
IV.18Ducati Kettenrad Z=37 B-1 10	48
IV.19Kugellager 1	49
IV.20Kugellager 2	49
IV.21Getriebeproblem 1	50
IV.22Getriebeproblem 2	51

IV.23Akku Vorderseite	52
IV.24Akku Motorblock	53
IV.25Akku Mitte	53
IV.26Befestigung der Seitenplatten	55
IV.27Befestigung des Schwingarmes	55
IV.28Explosionsansicht des gesamten Getriebes	56
IV.29Detailansicht Getriebe	56
V.1 Eigenschaften und Kennwerte einer Lithium Ionen Zelle[[ZellenEigenschaften]]	59
V.2 Explosionsdarstellung einer Doppelzelle	60
V.3 1. Akkubox	61
V.4 2. Akkubox	61
V.5 3. Akkubox	62
V.6 Serienschaltung beliebig vieler Zellen[[ReihenschaltungZellen]]	63
V.7 Parallelschaltung beliebig vieler Zellen[[ParallelschaltungZellen]]	64
V.8 Verschaltung der einzelnen Zellen innerhalb der Akkupacks	65
V.9 Verschaltung der Akkupacks	66
V.10 Mini Batteriemanagementsystem der Firma EMUS[[EmusBmsMini]]	68
V.11 Vergleich zwischen Aktiven- und Passiven-Battery Balancing[[VergleichBalancing]]	71
V.12 Funktionsweise des Passiven Battery-Balancing[[FunktionsweisePassivBalancing]]	72
V.13 Funktionsweise des Aktiven Battery-Balancing[[FunktionsweiseAktivBalancing]]	73
V.14 Anschlussplan der Akkupacks an das BMS	75
V.15 Anschlussplan der Temperatursensoren[[AnschlussplanTemp]]	76
V.16 Anschlussplan der Last an das BMS[[AnschlussplanLast]]	77
V.17 Topologie eines High Speed CAN-Netzwerks	78
V.18 Steckerbelegung des CAN-Buses an dem BMS	78
V.19 Hauptansicht der EMUS EVGUI App[[HauptansichtApp]]	79
V.20 Detaillierte Ansicht der EMUS EVGUI App[[DetaillierteApp]]	80
V.21 Menü der EMUS EVGUI App[[MenÄijApp]]	81
V.22 Battery Status Anzeige der EMUS EVGUI App[[BatteryStatusApp]]	82
V.23 BMS Status Anzeige der EMUS EVGUI App (Seite 1)[[BM]]	83
V.24 BMS Status Anzeige der EMUS EVGUI App (Seite 2)	84
V.25 Verschaltung des Ladegeräts über das BMS	86
VI.1 Grundaufbau des Laststromkreises [E.1]	89
VI.2 Ashwoods-Motor Kennlinie	93
VI.3 Grundaufbau des Steuerstromkreises [E.2]	95
VI.4 Digital Input Specifications [[Manual]]	96
VI.5 Analog Input Specifications [[Manual]]	96
VI.6 Throttle Input Specifications [[Manual]]	97
VI.7 Sin/Cos Sensor Input Specifications [[Manual]]	97
VI.8 KSI and Coil Return Input Specifications [[Manual]]	98
VI.9 Analog Output Specifications [[Manual]]	98
VI.10Digital and PWM Output Specifications [[Manual]]	99
VI.11Power Supply Output Specifications [[Manual]]	99
VI.12Communications Port Specifications [[Manual]]	100
VI.13ECO/Sport-Select Programmierung[VCL_Programmcode]	109
VI.14Leonardumformer Versuchsaufbau	110
VI.15Leonardumformer Spannungsüberhöhung	111
VII.1Raspberry Pi - Steuereinheit des HCIS	114
VII.2Grundaufbau des Human-Computer Interaction Systems	114
VII.3Aufbau des 12V Versorgungssystems	115
VII.4Aufbau des 5V Versorgungssystems	116
VII.5Anschlussplan Eingänge	117
VII.6GUI Komponente - Modus Anzeige	118
VII.7Anschlussplan Relais	120
VII.8Paneel Maße	121
VII.9Befestigung des Displays	122

VII.1	Aufbau der Graphischen Benutzeroberfläche	122
VII.1.1	GUI Komponente - Navigation Menü	123
VII.1.2	GUI Komponente - Balken Anzeige	124
VII.1.3	GUI Komponente - Modus Anzeige	124
VII.1.4	GUI Komponente - Graph	125
VII.1.5	GUI Fenster - Login Menü	126
VII.1.6	GUI Fenster - Fahrdaten	126
VII.1.7	GUI Fenster - Akkudaten	127
VII.1.8	GUI Fenster - Fahrdaten Diagnose	127
VII.1.9	GUI Fenster - Fehler Liste	128
VII.2	Slots und Signals Konzept	129
VII.2.1	Verbindung Frontend zu Backend	130
VII.2.2	Anschlussplan CAN-Modul	131
A.1	Zeitplan und Arbeitszeiten	139
A.2	Kostenaufstellung und Stückliste	140
C.1	Wellenersatz	143
C.2	Seitenplatte Rechts	144
C.3	Seitenplatte Links	145
C.4	Abstandhalter	146
C.5	Aufbau/Zusatzplatte	147
C.6	Achse 1/Antriebsachsen	148
C.7	Achse 3	149
C.8	Achse 2	150
C.9	Akkubox Motorblock	151
C.10	Akkubox Vorderseite	152
C.11	Akkubox Mitte	153
C.12	Ashwoods Motor	154
C.13	Curtis Controller	155
C.14	Akkubox Mitte	156
E.1	Grundaufbau des Laststromkreises	172
E.2	Grundaufbau des Steuerstromkreises	173

Tabellenverzeichnis

III.1 Auszug IP Schutzarten	15
IV.1 Gewichtstabelle	39
IV.2 Aluminium: AlMg4.5Mn Materialdaten	39
IV.3 Auszug Wertetabelle ISO 08 B-1	46
IV.4 Auszug Wertetabelle ISO 10 B-1	47
VI.1 Speed-Limiter Parameter [[Manual]]	104
VI.2 Response Parameter [[Manual]]	105
VI.3 Antriebsmodi	106
VI.4 Datenpakete Deklaration	108
VII.1 Berechnung der Leistung des 12V-Systems	115
VII.2 Datenbankstruktur der Benutzer Tabelle	133
VII.3 Datenbankstruktur der Fehler-Datenpaket Tabelle	133
VII.4 Datenbankstruktur der Datenpaket 1 Tabelle	134
VII.5 Datenbankstruktur der Datenpaket 2 Tabelle	134
VII.6 Datenbankstruktur der Datenpaket 3 Tabelle	134

Code Listings

III.1 Hello World Programm in Pyhton	37
III.2 Hello World Programm in JavaScript	37
VII.1Code zum Starten eines Threads	120
VII.2Konfigurieren des CAN Adapters vor Import	132
VII.3Konfigurieren des CAN Adapters in Code	132
VII.4Konfigurieren des CAN Adapters in Code	132
VII.5Konfiguration der Datenbankschnittstelle	135
VII.6SELECT Befehl über MySQL Connector	136
VII.7INSERT Befehl über MySQL Connector	136