

# DIPLOMARBEIT

Gesamtprojekt

## Entwicklung eines elektrisch betriebenen Rennkarts

### **Softwareentwicklung - Antriebsstrang**

Klemens Armstorfer      5AHET      Betreuer: Dipl.-Ing. (FH) Roland Holzer

### **Konstruktion und Design**

David Fingerlos      5AHET      Betreuer: Dipl.-Ing. Peter Lindmoser

### **Visualisierung und Mechanik**

Lukas Furtner      5AHET      Betreuer: Dipl.-Ing. Robert Fuchs

### **BMS und Mechanik**

Markus Sinzinger      5AHET      Betreuer: Dipl.-Ing. MBA Adolf Reinhart

ausgeführt im Schuljahr 2016/17

---

Abgabevermerk:

Datum: 04. April 2017

übernommen von:



## Eidesstattliche Erklärung

Wir erklären eidesstattlich, 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.

.....  
Klemens Armstorfer

.....  
David Fingerlos

.....  
Lukas Furtner

.....  
Markus Sinzinger



## Danksagung

Wir möchten uns bei allen an unserem Projekt Mitwirkenden, die uns mit Wissen, Tatenkraft, Komponenten oder einem Budget unterstützt haben, herzlichst bedanken!

Ein besonderes Dankeschön möchten wir an unsere Sponsoren E-Kontakt, Bürklin, CKR, HSCT, Kart-World Ebensee-Salzburg, Schrack Technik und Sigmatek, sowie den jeweils zuständigen Ansprechpartnern und Projektbetreuern aussprechen!

Zudem möchten wir uns bei unseren Projektbetreuern Prof. Dipl.-Ing. (FH) Roland Holzer, Prof. Dipl.-Ing. Peter Lindmoser, Prof. Dipl.-Ing. Robert Fuchs und Prof. Dipl.-Ing. MBA Adolf Reinhart für ihre stetige Hilfe und der Unterstützung mit Fachwissen bei allen möglichen Fragen und Problematiken bedanken. Zudem gilt dieser Dank auch für das Korrekturlesen dieser Diplomarbeit.

Ein weiteres Dankeschön gilt unserem Fachlehrer Franz Rieser, welcher uns bei allen mechanischen Fragen mit Rat und Tat zur Seite stand.



**DIPLOMARBEIT  
DOKUMENTATION**

Namen der Verfasser	Klemens Armstorfer, David Fingerlos, Lukas Furtner, Markus Sinzinger
Jahrgang Schuljahr	5AHET 2016/17
Thema der Diplomarbeit	Entwicklung eines elektrisch betriebsenen Rennkarts
Kooperationspartner	Bürklin GmbH & Co. KG CKR HSCT KARTWORLD SALZBURG Schrack Technik Energie GmbH SIGMATEK GmbH & Co KG
Aufgabenstellung	<p>Das Ziel unseres Projekts ist die Entwicklung eines Renn-Karts mit Elektroantrieb. Unser elektrisch betriebenes Go-Kart soll mit aktuellen Renn-Karts in Sachen Geschwindigkeit und Beschleunigung mithalten können. Dazu sind eine hohe elektrische Antriebsleistung und eine große Speicherkapazität nötig. Als Antrieb dient ausschließlich ein Elektromotor, gebremst wird sowohl elektrisch als auch mechanisch über Scheibenbremsen an allen Rädern. Die Fahrzeit soll für längere Rennen reichen, daher muss der Akku entsprechend dimensioniert werden. Zusätzlich ist eine Rekuperation der Bremsenergie möglich. Darüber hinaus haben wir uns auch Ziele im Regelungstechnikbereich gesetzt: Wir wollen an unserem Kart diverse Fahrassistenten realisieren, unter anderem eine Antriebschlupfregelung und eine intelligente Temperaturüberwachung. Alles in allem soll unser Projekt ein voll funktionstüchtiges Elektrofahrzeug sein, welches vielleicht in Zukunft als Vorbild für neue Konzepte von Elektroautos dient.</p>





Realisierung	<p>Die Grundlage stellt ein Chassis eines Renn-Karts vom italienischen Hersteller CKR dar. Auf dieses Chassis werden alle Komponenten montiert. Für jedes Bauteil wurden eigens Halterungen, Adapter und Abdeckungen gefertigt und montiert. Als Antrieb wird ein Elektromotor des Herstellers Motenergy verwendet. Bei diesem Motor handelt es sich um einen permanenterregten Synchronmotor mit einer niedrigen Nennspannung aber dennoch hohem Drehmoment. Angesteuert wird dieser Motor über einen speziellen Motorcontroller der Firma Sevcon. Für die Energieversorgung werden Akkumulatoren des Herstellers Lipopower verbaut. Dabei handelt es sich um Lithium – Eisenphosphatzellen mit einer Gesamtspannung von 96Volt DC und einem Energieinhalt von ca. 10kWh. Überwacht werden diese Zellen mittels eines Batteriemanagementsystems von EMUS.</p> <p>Gesteuert werden alle Komponenten über eine SPS von Sigmatek. Hier sind alle Sensoren und Bedienelemente angeschlossen. Auch das BMS und der Motorcontroller werden von der SPS angesteuert. Des Weiteren sind hier eine Antriebs-schlupfregelung und eine Temperaturüberwachung implementiert. Visualisiert werden die Fahrdaten über ein Display, welches ebenfalls vom Hersteller Sigmatek gesponsert wird. Dieses Display ist mittels CAN direkt an der SPS angeschlossen.</p>
Ist-Darstellung zu Projektbeginn	<p>Mit dem Projekt wurde bereits in Schuljahr 2015/16 begonnen. Es wurde der Kartrahmen von CKR ausgesucht und gekauft. Damit war die Ausgangslage für das Projekt gesetzt. Im Sommer wurde von zwei Teammitgliedern ein Praktikum bei der Firma Sigmatek absolviert, wodurch die Grundkenntnisse im Umgang mit den benötigten Programmen gelernt wurden. Ebenso wurde bereits ein Sponsoring ausgemacht und die benötigten Bauteile der SPS ausgewählt. Auch wurden erste Berechnungen bezüglich Motorleistung angestellt. Basierend auf diesen Berechnungen und einer ausgiebigen Recherche im Internet wurden die weiteren Komponenten ausgewählt.</p>




Ergebnisse	<p>Das Ergebnis dieses umfangreichen Projekts ist ein voll funktionstüchtiges Elektro-Rennkart. Alle Systeme funktionieren so, wie sie in dieser Diplomarbeit beschrieben sind.</p> 	
Typische Grafik, Foto etc. (mit Erläuterung)	<p>Hier zu sehen ist ein komplettes 3D-Modell des Go-Karts, bei dem bereits alle Komponenten montiert und angeschlossen sind.</p> 	
Teilnahme an Wettbewerben, Auszeichnung	JugendInnovativ Bosch-Technik fürs Leben Preis TÜV-Preis	
Möglichkeit der Einsichtnahme in die Arbeit	Die Diplomarbeit liegt in der Bibliothek der Schule auf.	
Approbation (Datum/Unterschrift)	Prüfer/Prüferin	Abteilungsvorstand



**DIPLOMA THESIS  
DOCUMENTATION**


Author(s)	Klemens Armstorfer, David Fingerlos, Lukas Furtner, Markus Sinzinger
Form Academic year	5AHET 2016/17
Topic	Development of an electric powered race kart
Co-operation Partners	Bürklin GmbH & Co. KG CKR HSCT KARTWORLD SALZBURG Schrack Technik Energie GmbH SIGMATEK GmbH & Co KG
Assignment of Tasks	<p>The aim of this project is to develop an electrical powered Go-cart. This cart should compete with conventional petrol-powered karts in terms of acceleration and speed. Therefore, a powerful motor and high-capacity batteries are needed. The cart is powered by an electric motor, which can also be used as a brake. Mechanical brakes are attached to all wheels. The capacity of the batteries should be enough so that longer races are possible without recharging. The motor can work as a generator, which can recharge the batteries when braking. Furthermore, some drive-assistants are implemented, for example a traction-control system and a temperature-watch. All in all, this project should be a fully operational Go-cart which maybe leads to better concepts for future electric cars.</p>
Realisation	<p>The basic of the project is a chassis of the Italian cart-manufacturer CKR. All components are attached to this chassis; therefore, adapter plates and holders are needed. The motor is a synchronous machine, built by Motenergy. It has a nominal voltage of 96 Volts. The motor is controlled by a special controller made by Sevcon.</p>



	<p>As source of energy, batteries are used. These batteries were bought from the manufacturer Lipopower and have a nominal voltage of 96 Volts and a total energy content of 10 kWh. A battery - management - system from EMUS controls the cells. All components are controlled by a programmable logic controller from the manufacturer Sigmatek. At this PLC all sensors and control devices are connected. Furthermore, the whole program including the drive assistants is implemented here. To visualize some information about the state of the vehicle, a touchpanel is connected to the PLC via CAN-Bus.</p>
Starting point of the project	<p>This project began in the school year 2015/16. The chassis was chosen and bought, therefore the starting point was set. In the summer holidays two project members made an internship at Sigmatek where they learned the basics of programming with "Lasal". Furthermore, a sponsoring was arranged. After that, rst calculations were made which helped to choose the motor and other components which were best suited for the project.</p>
Results	<p>The result of the project is a fully functional electrical vehicle. All systems are working in the correct way, as described in this diploma thesis.</p> 





Illustrative Graph, Photo (incl. explanation)	Shown here is a complete 3D-Model of the Go-Cart. All components are attached and connected. <div data-bbox="636 409 1227 696">  </div>	
Participation in Competitions Awards	JugendInnovativ Bosch-Technik fürs Leben Preis TÜV-Preis	
Accessibility of diploma Thesis	The diploma thesis can be found in the school library.	
Approval (Date/Sign)	Examiner	Department Manager



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Projektteam . . . . .	1
1.2	Projektidee . . . . .	2
1.3	Zielsetzung . . . . .	2
1.4	Nutzen des Projektes . . . . .	3
1.5	Ausgangsposition . . . . .	3
1.6	Sponsoren . . . . .	3
1.7	Zeitplan . . . . .	5
<b>2</b>	<b>Topologie</b>	<b>7</b>
2.1	Elektrisches Konzept als Blockschaltbild . . . . .	7
2.2	Sicherheitskonzept . . . . .	8
2.2.1	Basisschutz . . . . .	8
2.2.2	Fehlerschutz . . . . .	9
2.2.3	Zusatzschutz . . . . .	9
2.2.4	Schutzfunktionen in der Steuerung . . . . .	10
<b>3</b>	<b>Planung und Design</b>	<b>11</b>
3.1	Vorhergehende Berechnungen . . . . .	11
3.2	Planen und Modellieren des Karts . . . . .	14
3.2.1	Grundmodell zusammenbauen und adjustieren . . . . .	14
3.2.2	Anbauen der Erweiterungen . . . . .	17
3.2.2.1	Lenkradeinheit . . . . .	17
3.2.2.2	Antriebseinheit . . . . .	18
3.2.2.3	Batteriesystem . . . . .	20
3.2.2.4	Armaturenbrett . . . . .	20
3.2.2.5	SPS . . . . .	21
3.2.2.6	Verwendete Sensorik . . . . .	22
3.2.2.7	Kettenradantrieb . . . . .	24
3.2.2.8	Sonstige elektrische Bauteile . . . . .	26
3.2.2.9	Allgemeines . . . . .	27
3.2.2.10	Liste der extern bezogenen Bauteile . . . . .	28
3.3	Rendering . . . . .	29
3.3.1	Rendering Gesamtmodell . . . . .	29
3.3.2	Rendering der Mechanikteile . . . . .	30
3.3.3	Renderings für die Visualisierung . . . . .	31
3.3.4	Animationen . . . . .	31

3.3.5	Allgemein . . . . .	31
3.4	Plakate . . . . .	32
3.4.1	Plakat für die BIM . . . . .	32
3.5	Design . . . . .	33
3.5.1	Batterieboxen . . . . .	33
3.5.1.1	Vorderseite Batterieboxen . . . . .	34
3.5.1.2	Rückseite Batterieboxen . . . . .	34
3.5.1.3	Innenseite Batterieboxen . . . . .	34
3.5.2	Vorderverkleidung des Karts . . . . .	35
3.5.2.1	Fahrzeugkennzeichen . . . . .	35
3.5.2.2	Druck . . . . .	36
3.6	Icons für die Visu . . . . .	37
3.6.1	Statusicons . . . . .	37
3.6.1.1	Entwicklung des Grunddesigns . . . . .	37
3.6.1.2	ASR Symbol . . . . .	38
3.6.1.3	Motorcontrollersymbol . . . . .	38
3.6.1.4	Schlüsselsymbol . . . . .	38
3.6.1.5	Softkeys . . . . .	38
3.6.1.6	Fahrmodus . . . . .	39
3.6.1.7	Fehleranzeige . . . . .	39
3.6.1.8	Quittieren . . . . .	39
3.6.1.9	Rennmodus . . . . .	39
3.6.1.10	Status . . . . .	39
<b>4</b>	<b>Hardware</b>	<b>41</b>
4.1	Elektromotor . . . . .	41
4.1.1	Technische Erklärung . . . . .	42
4.1.2	Problemstellung . . . . .	43
4.2	Motorcontroller . . . . .	44
4.2.1	Funktionsweise . . . . .	45
4.2.1.1	Digital In 1 – Key Switch . . . . .	46
4.2.1.2	Digital In 2 – Forward Switch und Digital In 3 – Reverse Switch . . . . .	46
4.2.1.3	Digital In 4 - Brake Switch . . . . .	46
4.2.1.4	Digital In 5 und Digital In 7 - Driveability Select 1 & 2 . . . . .	46
4.2.2	Problemstellung . . . . .	47
4.3	Steuerung . . . . .	48
4.3.1	CP111 & PS101 . . . . .	48
4.3.2	KL180 und KL181 . . . . .	49
4.3.3	DI202 und TO161 . . . . .	50
4.3.4	AI043 & AO081 . . . . .	51
4.3.5	SCP111 und SDM081 . . . . .	52
4.4	Weitere elektrische Bauteile . . . . .	53
4.4.1	Relais . . . . .	53

4.4.2	Isolationsüberwachungssystem . . . . .	54
4.4.3	DC-DC Wandler . . . . .	54
4.4.4	Induktive Näherungssensoren - Drehzahlmesser . . . . .	54
4.4.5	Leitungsschutzschalter . . . . .	55
4.4.6	Gaspedal . . . . .	55
4.4.7	Endschalter für die Bremse . . . . .	55
4.4.8	Schlüsselschalter . . . . .	56
4.4.9	NOT-AUS Taster . . . . .	56
4.4.10	Einschaltknopf für die ASR . . . . .	56
4.4.11	Kippschalter . . . . .	56
4.4.12	Steckverbinder . . . . .	57
4.4.13	Kabel und Kabelschuhe . . . . .	57
4.4.14	Sonstige Komponenten . . . . .	57
4.5	Lageplan . . . . .	58
4.6	Energieversorgung . . . . .	59
4.6.1	Einführung . . . . .	59
4.6.2	Aufbau . . . . .	59
4.6.3	Dimensionierung . . . . .	59
4.6.4	Batteriemanagementsystem . . . . .	60
4.6.4.1	EMUS BMS Control Unit . . . . .	60
4.6.4.2	EMUS BMS Cell Module A/B . . . . .	61
4.6.4.3	EMUS BMS Cell Group Module . . . . .	62
4.6.4.4	EMUS BMS Dual Range Current Sensor . . . . .	63
4.6.4.5	Konfiguration des EMUS BMS . . . . .	64
4.6.4.6	Kommunikation mit der SPS . . . . .	66
4.6.4.7	Ladevorgang . . . . .	67
4.6.4.8	Überblick Gesamtkonzept . . . . .	67
4.7	Schaltpläne . . . . .	69
4.7.1	Steuerteil . . . . .	69
4.7.2	Impedanzwandlung . . . . .	71
4.7.3	Lastteil . . . . .	72
<b>5</b>	<b>Mechanik</b>	<b>75</b>
5.1	Zusammenbau des Roh-Chassis . . . . .	75
5.1.1	Anbringung der Verkleidungen . . . . .	75
5.1.2	Montage des Lenkrades . . . . .	75
5.2	Chassis Verlängerung . . . . .	76
5.2.1	Verlängerungsstücke . . . . .	76
5.2.2	Hintere Bodenplatte . . . . .	76
5.3	Motorhalterung . . . . .	77
5.3.1	Motorhalterung Grundgestell . . . . .	77
5.3.2	Spannvorrichtung für den Kettenspanner . . . . .	78
5.4	Kettenantrieb . . . . .	78
5.4.1	Auslegung . . . . .	79

5.4.2	Bearbeitung der gekauften Komponenten . . . . .	79
5.4.3	Höherlegen des Karts hinten . . . . .	79
5.4.4	Aufstockung der Motorhalterung . . . . .	79
5.4.5	Arbeiten am Kettenspanner . . . . .	80
5.5	Schutz Kettenantrieb . . . . .	80
5.5.1	Seitenabdeckung . . . . .	81
5.5.2	Hauptabdeckung . . . . .	81
5.5.3	Schutzmaßnahme gegen Verbiegen . . . . .	81
5.6	Schutz Motorcontroller . . . . .	82
5.6.1	Abstandshalter . . . . .	82
5.6.2	Abdeckungsgehäuse . . . . .	83
5.7	Sitz . . . . .	83
5.7.1	Montage des Sitzes . . . . .	83
5.7.2	Unterboden . . . . .	83
5.8	Lenkrad . . . . .	84
5.8.1	Displayhalterung . . . . .	84
5.8.2	Diverse Arbeiten am Lenkrad . . . . .	85
5.8.3	Schutz auf der Hinterseite . . . . .	85
5.9	Bedienoberfläche . . . . .	86
5.9.1	Armaturenbrett . . . . .	86
5.9.2	Abdeckung der Kabel auf der Hinterseite . . . . .	86
5.10	Kunststoffgehäuse SPS . . . . .	86
5.11	Gaspedalauswertung . . . . .	87
5.11.1	Adapter auf dem Gaspedal . . . . .	87
5.11.2	Adapter zum Kartrahmen . . . . .	88
5.11.3	Schutz Linearpotentiometer . . . . .	88
5.12	Endschalterbefestigung für die Auswertung der Bremse . . . . .	89
5.12.1	Befestigungsblech . . . . .	89
5.12.2	Abdeckung . . . . .	90
5.13	Halteungsbleche für die induktiven Sensoren . . . . .	90
5.14	Batterieboxen . . . . .	90
5.14.1	Auslegung der Boxen . . . . .	91
5.14.2	Maßnahmen zur Verstärkung des Kartrahmens . . . . .	91
5.14.3	Ebnungsbleche . . . . .	92
5.15	3D-Druck Teile . . . . .	93
5.15.1	Fused Deposition Modeling . . . . .	93
5.15.2	Abdeckung Touchpanel . . . . .	94
5.15.3	Abdeckung Endschalter . . . . .	95
5.15.4	Rückverkleidung der Lenkradeinheit . . . . .	95
5.15.5	Rückverkleidung Armaturenbrett . . . . .	96
5.15.6	Selektives Lasersintern . . . . .	97
5.15.6.1	Abdeckung für das Isolationsüberwachungssystem . . . . .	98

<b>6</b>	<b>Software</b>	<b>99</b>
6.1	Programm der SPS	99
6.1.1	DriveKart	100
6.1.2	EnableDrive	101
6.1.3	ASR	103
6.1.4	TempControl	104
6.1.5	CAN-Object	105
6.1.6	Übergabeklassen für die Visualisierung	105
6.2	Safety - Programm	106
6.3	Programm des Motorcontrollers	108
6.3.1	Übersicht über das Programm	108
6.3.2	Programmierung mit dem Helper	110
6.3.2.1	Input/Output	113
6.3.2.2	TPDO/RPDO	114
6.3.2.3	Download des Programms	116
6.4	Visualisierung	117
6.4.1	Allgemeines	117
6.4.1.1	Auswahl des Displays	117
6.4.1.2	Wichtige Daten des Displays	118
6.4.2	Verbindungseinstellungen	118
6.4.2.1	Herstellung der Verbindung zur CPU	118
6.4.2.2	Einstellung auf der CPU und am Display	120
6.4.2.3	Einstellungen im Lasal Class 2	120
6.4.2.4	Einstellungen im Lasal Screen	121
6.4.3	Screens	122
6.4.3.1	Screenstruktur	122
6.4.3.2	Initialisierung	123
6.4.3.3	Global Screen	124
6.4.3.4	Startscreen	125
6.4.3.5	Fahrscreen	126
6.4.3.6	Statusscreen	128
6.4.3.7	Fehlerscreen	130
6.4.3.8	Weitere Funktionen	132
6.4.4	Übergabevariablen	134
6.4.4.1	Variablenliste	134
6.4.4.2	Alarmliste	135
6.4.4.3	Einstellungen im Lasal Class 2 und Lasal Screen	136
6.4.5	Arbeitsoberfläche	137
6.4.5.1	Grundeinstellungen	138
6.4.5.2	Images	138
6.4.5.3	Texte	139
6.4.5.4	Function Blocks	141
6.4.5.5	Datenanzeige	142
6.4.5.6	Generelle Voraussetzungen zum Fahrbetrieb	144

6.4.6	Notwendige Änderungen . . . . .	145
<b>7</b>	<b>Weitere Realisierungsmöglichkeiten</b>	<b>146</b>
7.1	Ausblick . . . . .	146
<b>A</b>	<b>Anhang</b>	<b>149</b>
A.1	Quellcodes des Programms der SPS . . . . .	149
A.1.1	DriveKart . . . . .	149
A.1.2	EnableDrive . . . . .	153
A.1.3	TempControl . . . . .	156
A.1.4	Antriebsschlupfregelung . . . . .	159
A.1.4.1	SpeedWheelFront . . . . .	159
A.1.4.2	ControlButton . . . . .	160
A.1.4.3	ASR . . . . .	160
A.2	CAD Werkstattzeichnungen . . . . .	163
A.2.1	Verlängerungsadapter . . . . .	163
A.2.2	Hintere Bodenplatte . . . . .	163
A.2.3	Motorhalterung . . . . .	164
A.2.4	Aufstockungsblech . . . . .	166
A.2.5	Spannvorrichtung . . . . .	167
A.2.6	Abdeckung Antrieb . . . . .	168
A.2.7	Schutzverstärkung gegen Verbiegen . . . . .	171
A.2.8	Abdeckung Motorcontroller . . . . .	172
A.2.9	Unterboden . . . . .	174
A.2.10	Displayhalterung . . . . .	175
A.2.11	Displayverkleidung . . . . .	176
A.2.12	Hinterabdeckung Lenkrad . . . . .	176
A.2.13	Armaturenblech . . . . .	177
A.2.14	Hinterabdeckung Armaturenblech . . . . .	178
A.2.15	SPS-Box . . . . .	179
A.2.16	Adapter Gaspedal . . . . .	179
A.2.17	Adapter Kart . . . . .	180
A.2.18	Schutz Linearpotentiometer . . . . .	181
A.2.19	Adapter Endschalter . . . . .	182
A.2.20	Abdeckung Endschalter . . . . .	183
A.2.21	Blech Induktiver Sensor . . . . .	183
A.2.22	Batteriebox rechts . . . . .	184
A.2.23	Batteriebox links . . . . .	184
A.2.24	Aluhülse . . . . .	185
A.2.25	Unterlagsbleche Batterieboxen . . . . .	185
A.2.26	Isolationsüberwachungssystemhalterung . . . . .	186



# Kapitel 1

## Einleitung

### 1.1 Projektteam



Armstorfer Klemens  
11. Februar 1998  
Steinbreite 4  
A – 5112 Lamprechtshausen



Fingerlos David  
05. Juli 1997  
General-Albort-Straße 19  
A – 5061 Elsbethen



Furtner Lukas  
08. März 1998  
Neuhausen 26  
A – 5131 Franking



Sinzinger Markus  
15. März 1998  
Ortholling 4  
A – 5120 Haigermoos

## 1.2 Projektidee

Elektromobilität hat in den letzten Jahren immer mehr an Bedeutung gewonnen. Dabei liegt das Hauptaugenmerk auf Wirtschaftlichkeit und Energieeffizienz. Die Fahrfreude und der Spaß am Fahren kommen dadurch leider oft zu kurz.

Mit unserem Projekt möchten wir mehr Menschen für E-Mobilität begeistern. Wir möchten zeigen, dass Elektromotoren bereits motorsporttauglich sind und dass man mit ihnen viel Spaß auf der Rennstrecke haben kann.

Die Umweltverträglichkeit dieses Go-Karts ist sehr gut. Durch den Elektromotor kommen keine Abgase zustande. Der Wirkungsgrad ist hoch, durch eine zusätzliche Bremsenergieerückspeisung wird dieser noch verbessert. Die Akkus können über den Strom aus der hauseigenen PV-Anlage geladen werden.

Mit diesem Go-Kart könnten in Zukunft ganze Kartbahnen betrieben werden. Solche Hallen mit elektrisch betriebenen Go-Karts kommen ohne Abgasabsaugung oder besondere Belüftung aus. Zudem wird dadurch die Abgasbelastung der Umwelt komplett eingestellt.

## 1.3 Zielsetzung

Das Ziel unseres Projekts ist die Entwicklung eines Renn-Karts mit Elektroantrieb. Unser elektrisch betriebenes Go-Kart soll mit aktuellen Renn-Karts in Sachen Geschwindigkeit und Beschleunigung mithalten können. Dazu ist eine große elektrische Antriebsleistung und eine ebenso hohe Speicherkapazität nötig.

Als Antrieb dient ausschließlich ein Elektromotor, gebremst wird sowohl elektrisch als auch mechanisch über Scheibenbremsen an allen Rädern. Die Fahrzeit soll für längere Rennen reichen, daher muss der Akku entsprechend dimensioniert werden. Zusätzlich realisieren wir eine Rekuperation der Bremsenergie.

Darüber hinaus haben wir uns auch Ziele im Regelungstechnikbereich gesetzt:

Wir wollen an unserem Kart diverse Fahrassistenten realisieren, unter anderem eine Antriebsschlupfregelung und eine intelligente Temperaturüberwachung.

Alles in allem soll unser Projekt ein voll funktionstüchtiges Elektrofahrzeug sein, welches vielleicht in Zukunft als Vorbild für Konzepte von Elektroautos dient.

## 1.4 Nutzen des Projektes

Unser Go-Kart soll der Abteilung Elektrotechnik als Prestigeprojekt dienen. Geplant ist, dass das Go-Kart bei allen Öffentlichkeitsauftritten der Schule gezeigt wird. Besonders beim Tag der offenen Tür, aber auch bei der BIM soll es allen Interessenten zur Schau gestellt werden.

Des Weiteren sind wir in unserer Abteilung ein Vorreiter im Bereich der E-Mobilität. Die nächsten Jahre sollen in der Abteilung Elektrotechnik mehr Projekte zum Thema „Elektrische Fahrzeuge“ umgesetzt werden. Mitunter wäre die Verwirklichung unseres Antriebskonzepts in einem vollwertigen Kraftfahrzeug angedacht.

## 1.5 Ausgangsposition

Mit dem Projekt wurde bereits in Schuljahr 2015/16 begonnen. Es wurde der Kartrahmen von CKR ausgesucht und gekauft. Damit war die Ausgangslage für das Projekt gesetzt. Im Sommer wurde von zwei Teammitgliedern ein Praktikum bei der Firma Sigmatek absolviert, wodurch die Grundkenntnisse im Umgang mit den benötigten Programmen gelernt wurden. Ebenso wurde bereits ein Sponsoring ausgemacht und die Bauteile ausgewählt. Auch wurden erste Berechnungen bezüglich Motorleistung angestellt. Basierend auf diesen Berechnungen wurde bereits zu Beginn des Projekts ein Motor ausgewählt und gekauft. Zur Auswahl wurden ebenso vergleichbare Anwendungen im Internet recherchiert und anhand dessen eine Auswahl getroffen.

## 1.6 Sponsoren

Um bei so einem großartigen Projekt Erfolg zu haben, ist es notwendig, über gewisse finanzielle Mittel zu verfügen. Dazu wurde von Anfang an überlegt, von welchen Firmen aus welchen Branchen welche Bauteile benötigt werden, sowie welche Firmen als Interessenten für ein Sponsoring infrage kämen.

Der Hauptsponsor des Karts ist der Absolventenverein der Abteilung Elektrotechnik. Über den Verein wurden das Chassis, der Motor und der Motorcontroller sowie einige Kleinteile angekauft.

Für die Abteilung Elektrotechnik ist das ein sehr wichtiges Projekt. So soll das Kart einerseits als Prestigeobjekt für alle offiziellen Auftritte der HTL genutzt werden, wie z.B. der BIM. Zum anderen können daraus für die Zukunft wichtige Erfahrungen im Bereich der Elektromobilität gemacht werden.

Um dem Kart eine solide Grundlage zu geben, wurde ein Chassis eines Rennkarts vom italienischen Karthersteller CKR gekauft, das offiziell von der CIK-FIA für die Rennstrecke zugelassen ist. CKR zeigte auch von Anfang an Interesse an diesem Projekt und hat deshalb in Form eines Sponsorings einen 40% - Rabatt auf den Kauf

gewährt. Die Lieferung des Kartrahmens sowie das Kaufen und Beziehen der Reifen übernahm die Firma Kart World Ebensee-Salzburg für uns.

Da bei einem Projekt dieser Art etliche elektrische Bauteile benötigt werden und ein persönlicher Kontakt zur Firma Bürklin bestand, hat sich Bürklin schnell als potentieller Sponsor herauskristallisiert und der Unterstützung unseres Projektes mit einem Budget auch zugestimmt.

Ein Teil des Projekt-Teams hat über die Sommerferien bei der Firma Sigmatek ein Praktikum absolviert und dabei bereits Abmachungen im Hinblick auf ein Sponsoring getroffen. Ebenso wurden hier die Grundlagen für die Software gelernt, wodurch eine Programmierung der Steuerung erst möglich wurde.

Der bei diesem Projekt verwendete Motorcontroller wurde vom deutschen Hersteller Sevcon gebaut. Die Vertriebsfirma des Motorcontrollers, der niederländische Elektronikhändler HSCT, zeigte ebenfalls reges Interesse an diesem Projekt und war deshalb auch sofort einverstanden, als Projekt-Partner einen Preisnachlass zu gewähren.

Da die Firma Schrack schon ein vorhergehendes Projekt unserer Abteilung erfolgreich unterstützt hat, war Schrack sofort bereit, die enormen Kosten des kompletten Batteriesystems zu finanzieren.



Abbildung 1.1: Logos unserer Sponsoren

## 1.7 Zeitplan

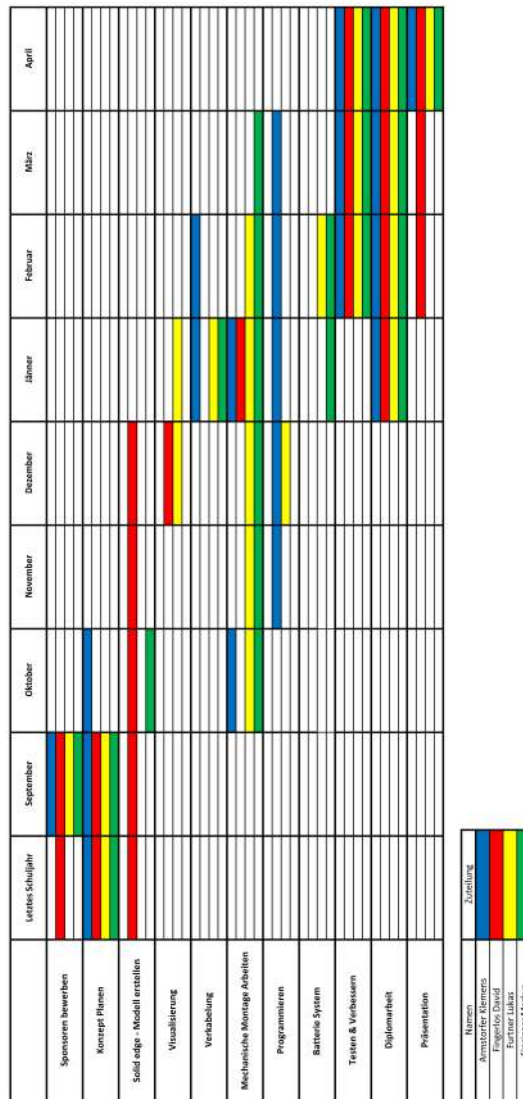


Abbildung 1.2: Zeitplan



# Kapitel 2

## Topologie

### 2.1 Elektrisches Konzept als Blockschaltbild

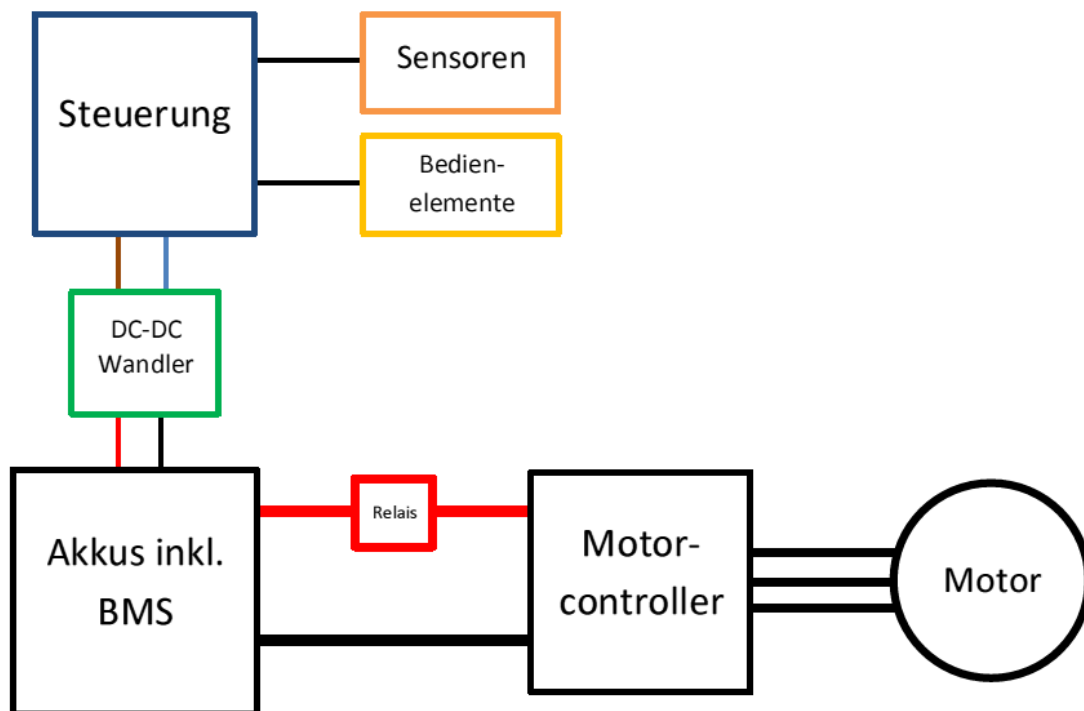


Abbildung 2.1: Blockschaltbild

Hier abgebildet ist das grundlegende Konzept des elektrischen Teils des Antriebs als Blockschaltbild. Dabei werden die Hauptkomponenten als einzelne Rechtecke dargestellt, die anderen Bauteile werden zu funktionellen Gruppen zusammengefasst. Ebenso wird hier komplett auf alle sicherheitsrelevanten Bauteile sowie auf alle Stecker verzichtet, um unnötige Komplexität zu vermeiden. Bei dem Blockschaltbild soll nur das Wesentliche anschaulich dargestellt werden.

Als Ausgangspunkt des Schaltbilds gelten die Akkumulatoren inkl. BMS. Sie stellen die Spannungsversorgung dar, womit alle elektrischen Komponenten mit Energie versorgt werden. Natürlich darf bei den Akkus nie auf das BMS verzichtet werden, da es sonst zu Schäden an den Zellen kommen könnte. Die Akkus werden über ein Relais zum Motorcontroller verbunden.

Der Motorcontroller an sich wird hier ebenfalls nur mit seinen Leistungskontakten dargestellt. Über die Batterieanschlüsse wird er mit Gleichspannung aus den Akkumulatoren versorgt. Der Motor wird aufgrund des benötigten Drehstroms mit seinen drei Phasen angeschlossen. Der gesamte Steuerteil wird nicht direkt über die Akkumulatoren gespeist, da hier die Spannung zu groß wäre. Vielmehr wird mittels DC-DC Wandler die Spannung auf 24V herabgesetzt, einem Potential, mit dem alle Steuerungskomponenten problemlos arbeiten können. Als Steuerung werden hier alle Module der SPS inklusive der Sicherheitsmodule bezeichnet. Sowohl die Sensoren als auch die Bedienelemente für den Fahrer werden aufgrund ihrer Menge lediglich als zusammengefasste Gruppen erwähnt. Als Vorteil sticht hier deutlich das Relais zwischen Akkus und Motorcontroller heraus: Es wird über die Steuerung ein- bzw. ausgeschaltet. Dadurch wird im Fehlerfall der Steuerung sofort der Energiefluss unterbrochen, sodass das Kart nicht ungesteuert weiterfahren kann.

## **2.2 Sicherheitskonzept**

Sicherheit ist bei allen Fahrzeugen ein großes Thema, besonders bei Elektrofahrzeugen. Das Hauptproblem stellt hier die konstante Spannung der Batterien dar, dadurch kann das Fahrzeug nicht wie bei netzgebundenen Anwendungen einfach stromlos geschaltet werden. Besonders bei einem Unfall kann das zu unkontrollierbaren Kurzschlüssen führen. Auch für den normalen Fahrbetrieb sind einige wichtige Sicherheitsmaßnahmen einzuhalten.

### **2.2.1 Basisschutz**

Als Basisschutz werden alle Kabel gegen direktes Berühren geschützt. Der Motorcontroller wird mit einem Gehäuse versehen, um auch hier kein direktes Berühren zu ermöglichen. An allen Klemmstellen der Batteriekabel werden Abdeckungen angebracht. Besonders wichtig ist die Isolierung der Batteriekabel gegeneinander, da hier sonst im Kurzschlussfall kurzzeitig mehrere Tausend Ampere fließen können. Dieser enorme Strom kann zu Funkenflug und im ungünstigsten Fall auch zum Schmelzen von Metall führen. Auch für den menschlichen Körper ist die hohe Gleichspannung nicht ungefährlich, bereits ab 60 Volt können bei Berührung bleibende gesundheitliche Schäden auftreten.



### 2.2.2 Fehlerschutz

Als Fehlerschutz werden bei beiden Batterieboxen Schmelzsicherungen mit einem Nennstrom von 500A angebracht. Hier ist das Problem, dass durch die Serienschaltung von zwei gleichen Sicherungen Selektivität nicht mehr gegeben ist. Jede Batteriebox muss aber eigens abgesichert sein, da sonst bei einem Kurzschluss in der Zuleitung der Batteriebox ein Fehler nicht mehr ausgeschaltet werden könnte. Auch beim Motorcontroller ist eine Flachsicherung direkt vor den Batterieanschluss geklemmt. Diese hat einen Nennstrom von 425A, sie ist also gegenüber den Sicherungen der Batterieboxen selektiv. Beim Steuerungsteil treten im Vergleich zum Leistungsteil viel kleinere Ströme auf. Daher muss die Steuerung auch separat abgesichert werden, da sonst trotz Fehlerfall die Schmelzsicherungen der Batterieboxen nicht reagieren würden. Vor den DC-DC Wandler wird ein Leitungsschutzschalter mit einem Nennstrom von 13A geschaltet. Auf der 24V – Ebene wird direkt hinter dem DC-DC Wandler ebenfalls ein Leitungsschutzschalter verbaut, jedoch mit einem Nennstrom von 10A.

### 2.2.3 Zusatzschutz

Als Zusatzschutz für die Batteriekabel wird ein Isolationsüberwachungssystem verbaut. Dieses System kontrolliert permanent den Isolationswiderstand zwischen B+ und der Karosserie sowie zwischen B- und der Karosserie. Hier sei anzumerken, dass die Karosserie nicht, wie bei herkömmlichen Autos mit Verbrennungsmotor üblich, mit dem Massepol verbunden wird.

## 2.2.4 Schutzfunktionen in der Steuerung

Als sichere Steuerung werden die „Safety“-Module der Sigmatek S-DIAS Produktfamilie verwendet. Diese sind zu normalen Modulen kombinierbar, jedoch arbeiten sie unabhängig und mit höherer Ausfallsicherheit (SIL3<sup>1</sup> Zertifiziert). Diese Safety-Module müssen auch eigenständig programmiert werden:

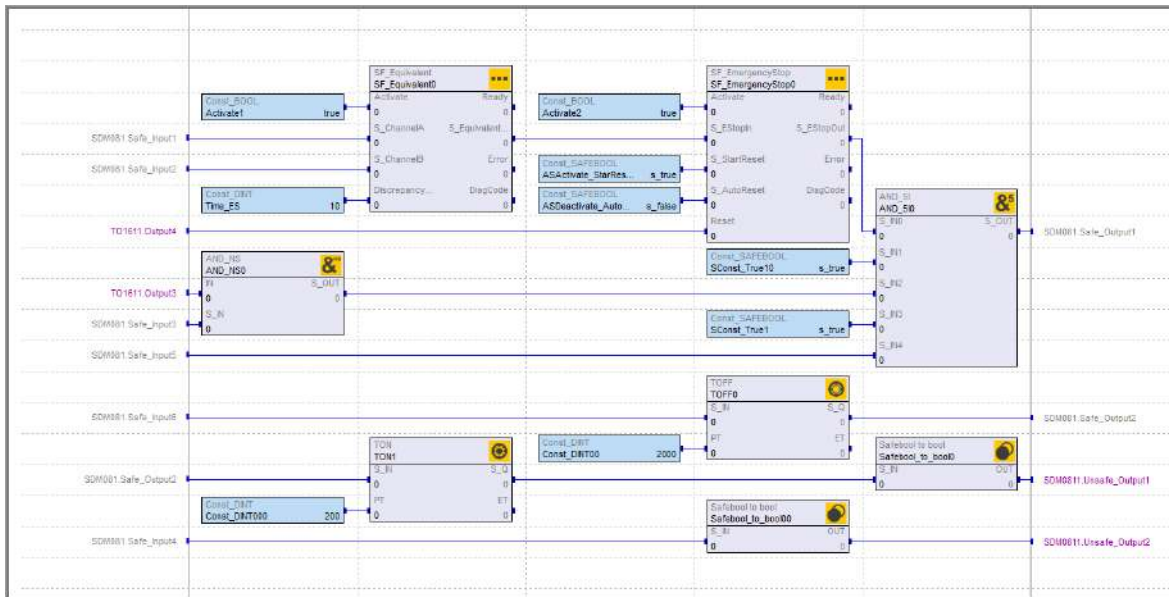


Abbildung 2.2: Programm der Safety-SPS

Verwendet werden eine Safety-CPU sowie ein digitales Ein- und Ausgangsmodul. Auf Eingang Eins und Zwei wird der Not-Aus Taster angeschlossen. Erwähnenswert ist hier auch, dass das Safety-Modul SDM081 eine Querschlusserkennung ermöglicht. Beide Kanäle des Not-Aus Tasters werden über einen Äquivalenzblock eingelesen, dadurch kann auch ein Wackelkontakt erkannt werden. An den Eingängen Drei und Vier werden zwei Schlüsselschalter angeschlossen. Mit dem ersten Schlüssel aktiviert man den gedrosselten Modus, mit dem zweiten den Turbomodus, bei dem das Kart Spitzenleistungen erreicht. Der Eingang Fünf wird direkt an den Fehler-Ausgang des Batteriemanagementsystems angeschlossen. Über den Eingang Nummer Sechs wird eingelesen, ob der Motorcontroller eingeschaltet ist. Am ersten Ausgang wird das Steuerrelais zum Einschalten des Motorcontrollers angeschlossen. Über den zweiten Ausgang wird ein Solid-State-Relais gesteuert, welches mit der Spule des Hauptrelais verbunden ist und somit im Fehlerfall den Energiefluss unterbrechen könnte.

<sup>1</sup>SIL: Safety Integrity Level nach IEC 61508

# Kapitel 3

## Planung und Design

### 3.1 Vorhergehende Berechnungen

Damit von Anfang an eine Abschätzung über die Kennwerte der zu verwendenden Bauteile getroffen werden konnte, wurde zu Beginn eine genaue Berechnung angestellt. Dazu wurden zuerst die Beschleunigungen und Geschwindigkeiten berechnet, welche mit dem zu verwendenden Motor einerseits in dem normalen Fahrmodus (dauerhaft fahrbar) und andererseits in einem Überlastmodus (thermisch nur kurzzeitig betreibbar) erreichbar wären. Nachfolgend musste für die Auslegung des Kettenantriebes errechnet werden, welche Belastungen maximal und dauerhaft auf die Kette wirken dürfen. Als Ausgangslage für die Berechnung der erreichbaren Leistungsdaten des Karts dienten die technischen Kennwerte des Motors sowie die Vorgabe von angestrebten dauerhaft zu fahrenden 80 km/h. Des Weiteren war es notwendig, sowohl den Reibungsfaktor und Durchmesser von Kartreifen (Slick-Reifen) als auch ein geschätztes Gewicht des Karts in die Berechnung miteinzubeziehen. Anhand der Ergebnisse einer ersten Berechnung konnte abgeschätzt werden, welche Leistungswerte das Kart letztendlich erreichen wird, und welches Übersetzungsverhältnis dabei den bestmöglichen Kompromiss zwischen einer guten Beschleunigung und einer ausreichenden Endgeschwindigkeit ergeben würde.

		<b>Normalmodus</b>		<b>Rennmodus</b>	
		Wert	Einheit	Wert	Einheit
Nenndrehmoment	MN	50,00	Nm	85,00	Nm
Nenndrehzahl	NN	3800,00	rpm	6000,00	rpm
Reifenradius	r	13,97	cm	13,97	cm
Reifendurchmesser	d	27,94	cm	27,94	cm
Reifenumfang	u	43,89	cm	87,78	cm
Gewicht	m	250,00	kg	250,00	kg
Maximalgeschwindigkeit	vmax	80,00	km/h	126,32	km/h
Umdrehungsfrequenz	fRot.	25,32	Umdr./s	39,97	Umdr./s
Drehzahl	n	1519,02	rpm	2398,45	rpm
Übersetzungsverhältnis	ü	2,50		2,50	
Wirkendes Drehmoment	M	125,08	Nm	212,64	Nm
Beschleunigungskraft	F	895,35	N	1522,10	N
Beschleunigung	a	3,58	m/s <sup>2</sup>	6,09	m/s <sup>2</sup>

Tabelle 3.1: 1. Berechnung

<b>Berechnung Kettenrad- antrieb</b>	<b>Maximal auftretende Belastungen</b>		
		Wert	Einheit
Übersetzungsverhältnis	ü	2,50	
<b>Achse</b>			
Drehmoment	MAchse	212,64	Nm
Drehzahl	nAchse	2398,45	rpm
Durchmesser Zahnrad max	dZAchse	180,00	mm
<b>Motor</b>			
Drehmoment	MMot.	85,06	Nm
Umdrehungen	mMot.	5996,12	rpm
Durchmesser Zahnrad	dZMot.	72,00	mm
<b>Kette</b>			
Zugkraft	F	2362,64	N
Geschwindigkeit	v	22,60	m/s

Tabelle 3.2: 2. Berechnung

Basierend auf den Ergebnissen der ersten Berechnung wurde die zweite Berechnung angesetzt, in welcher die genauen Belastungen des Kettenradantriebs untersucht wurden.

## 3.2 Planen und Modellieren des Karts

Um das ausgearbeitete Konzept in Form eines 3D-Modells zu verwirklichen, wurde das Kart mittels Solid Edge gezeichnet. Als Erweiterung vom Grundmodell des Karts wurden alle benötigten Halterungen, Komponenten und Bauteile gezeichnet und zum Grundmodell hinzugefügt.

### 3.2.1 Grundmodell zusammenbauen und adjustieren

Als Ausgangslage für das Grundmodell wurde von CRG, einer Partner Firma von CKR, ein CAD-Grundmodell zur Verfügung gestellt. Dieses Grundmodell war sehr wichtig, da einige spezielle Bauteile, wie die Form des Grundrahmens oder die genaue Linienführung der Verkleidungsteile, schwer nachzuzeichnen sind.



Abbildung 3.1: Grundmodell Ausgangslage

Das erhaltene Modell war allerdings eine Standardzeichnung und daher nicht genau auf die Maße des von uns verwendeten Kartrahmens abgestimmt. Um mit der Zeichnung des Grundmodells beginnen zu können, mussten vorweg alle Abweichungen behoben werden. Dazu war es erforderlich, einige im Grundmodell enthaltene Komponenten zu entfernen und andere neu- bzw. umzuzeichnen. Als weiterer Schritt war es nötig, die einzelnen Teile des erhaltenen Grundmodells zusammenzubauen. Dazu wurden auch die neu erstellten Bauteile sowie diverse Schrauben und Kleinteile zum Modell hinzugefügt und assembliert. Nachdem all diese Unzulänglichkeiten bereinigt und die Grundkomponenten richtig platziert waren, war die Ausgangslage für das CAD-Modell geschaffen.



Abbildung 3.2: Grundmodell korrigiert und zusammengebaut

CRG konnte leider aus rechtlichen Gründen keine Zeichnungen von den Leistungsteilen weitergeben, die im Rennsport eine entscheidende Rolle spielen. Deshalb musste das gesamte Bremssystem neu gezeichnet und hinzugefügt werden. Um das Bremssystem, bestehend aus Halterung, Bremshalterung, Bremsscheibe und Bremsbacken, möglichst detailgetreu zu digitalisieren, wurde auf der Website "GrabCAD" nach ähnlichen Systemen gesucht und an dem Bremssystem AP CP6600, welches dem auf dem zugekauften Chassis angebrachten sehr nahe kommt, orientiert. Das von GrabCAD heruntergeladene Modell von einem Bremssattel wurde, um den benötigten Anforderungen gerecht zu werden, stark um-modelliert und verändert. Für die Bremsscheiben konnte ein gut passendes Modell im Internet gefunden werden. Da an den Vorderrädern ein sehr ähnliches, allerdings in der Größe viel kleineres Bremssystem angebracht ist, wurden anhand der Dimensionierungsfunktion die Komponenten des hinteren Bremssystems um den passenden Faktor verkleinert und an die Vorderachse gesetzt. Als weiterer Schritt wurden das zum Bremssystem gehörende, fehlende Hydrauliksystem, die zugehörigen Bremsflüssigkeitsbehälter sowie die Felgenhalterungen gezeichnet und dem Gesamtmodell hinzugefügt.



Abbildung 3.3: Brems- und Hydrauliksystem

Durch anschließendes Hinzufügen der Reifen, den seitlichen Verstrebungen und der kompletten Lenk-Einheit wurde das Grundmodell so komplettiert, dass es dem von uns gekauften Modell entsprach.



Abbildung 3.4: Ausgangsmodell



### 3.2.2 Anbauen der Erweiterungen

Da nun die – wie von CKR gelieferte – Basis vollständig digitalisiert war, konnte mit dem Aufbau und dem Hinzufügen aller benötigten Komponenten begonnen werden.

#### 3.2.2.1 Lenkradeinheit

Um die SPS von Sigmatek zu steuern und zu überwachen, wurde ein 3,5 Zoll großes Touch-Display von Sigmatek bezogen und in das Lenkrad integriert. Dazu wurde zuerst ein Ausschnitt in der oberen Speiche des Lenkrades gemacht. Anschließend musste noch eine Grundplatte, auf der das Touchpanel befestigt wurde, gefertigt werden. Die Zeichnung des Touchpanels war auf der Hersteller-Homepage verfügbar. Da bei Arbeiten am Lenkrad die Abdeckung über dem Touchpanel beschädigt wurde, musste diese am PC neu gezeichnet und durch einen 3D-Drucker von unserer Partner Firma Bürklin gefertigt werden. Des Weiteren wurden in das Lenkrad noch ein Kippschalter zur Wahl der Fahrtrichtung sowie ein Taster mit blauem LED Ring zum Aktivieren der ASR eingebaut. Die Zeichnung des Tasters gibt es auf der Homepage des Herstellers zum Herunterladen. Beim Kippschalter wurde ein Modell von GrapCAD heruntergeladen und verwendet. Um die Kabel dieser Komponenten gebündelt abzuführen und die Rückseiten der Bauteile vor Schmutz und anderen Einwirkungen zu schützen, wurde noch eine Rückverkleidung für das komplette Lenkrad gezeichnet und durch einen 3D-Drucker in unserer Schule produziert.



Abbildung 3.5: Aufbau der Lenkradeinheit

### 3.2.2.2 Antriebseinheit

Als nächstes wurde eine Basis für die Antriebskomponenten (Motor und Motorcontroller) konzipiert. Dazu wurde das Chassis mit zwei Aluminiumstiften verlängert, wodurch sich eine um 10cm vergrößerte Auflagefläche ergibt. Als Unterlage für die Motorkomponenten ist eine Aluminiumplatte in Verwendung. Aluminium ist für diese Anwendung das ideale Material, da es sowohl eine ausreichende Stabilität, ein geringes Gewicht, als auch eine hohe thermische Leitfähigkeit für die Wärmeableitung von Motor und Controller besitzt. In weiterer Folge werden die sich erwärmenden Platten durch die am Unterboden vorbeiströmende Luft gekühlt.

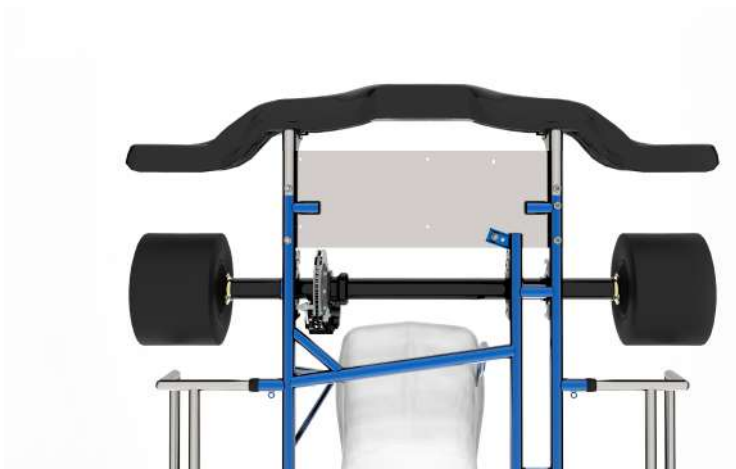


Abbildung 3.6: Umbauten der Basis der Antriebseinheit

Der Motorcontroller wurde einfach auf der Grundplatte platziert und mit vier Schrauben fixiert. Um den überaus leistungsstarken Motor sicher und stabil zu befestigen, wurde eine Halterung aus 8mm dickem Stahl hergestellt. Diese Halterung wurde anschließend auf der Grundplatte befestigt und mehrfach mit dem Kart-Rahmen verschraubt. Dadurch kann eine hohe Festigkeit gewährleistet werden, welche das Drehmoment problemlos aushält. Anschließend wurde der Motor in diese Halterung eingesetzt.

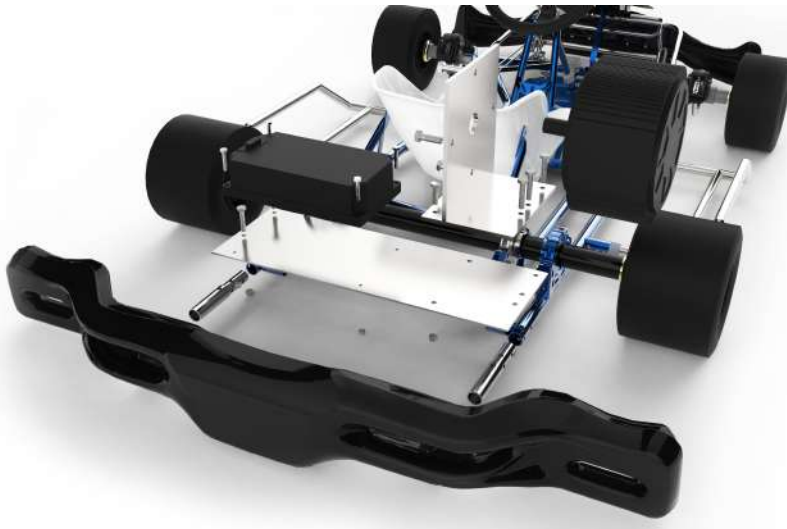


Abbildung 3.7: Aufbau der Antriebseinheit

Um dem Sitz einen besseren Halt zu geben, wurde unter dem Sitz eine Aluminiumplatte als Unterboden befestigt. Ein weiterer Vorteil dieser Konstruktion ist, dass der Luftstrom ohne Verwirbelungen an der Unterseite des Karts entlangströmen kann und somit eine bessere Kühlung erreicht wird.



Abbildung 3.8: Kühlkonzept der Leistungskomponenten

### 3.2.2.3 Batteriesystem

Da die leistungsstarken Batterien im Falle eines Unfalles ein enormes Sicherheitsrisiko darstellen, sind diese sicher in zwei stabilen Metallboxen untergebracht. Zudem ist um alle Batterien eine Isolationsschicht aus Schaumstoff angebracht, welche gegen elektrische Fehler zwischen Batterien und Gehäuse schützen soll. Damit eine sichere Verbindung zwischen den schweren Akkuboxen und dem Kart besteht, wurden diese von unten mit der Karosserie an mehreren Punkten verschraubt. Eine Zeichnung von ähnlichen Batterien wie der von uns verwendeten konnte auf Grapcad gefunden werden. Dieses Modell musste allerdings in den Maßen verändert werden. Dazu wurde erneut das Dimensionierungstool von Solid Edge verwendet. Bei dieser Anwendung wurden Länge, Breite und Höhe mit verschiedenen Faktoren gestreckt bzw. gestaucht. Die richtig dimensionierten Batteriezellen wurden anschließend mit dem Musterwerkzeug in der Batteriebox platziert.



Abbildung 3.9: Aufbau des Batteriesystems

### 3.2.2.4 Armaturenbrett

Die fundamentalen Steuereinheiten sind auf einem Armaturenbrett direkt unterhalb des Lenkrads platziert. Zu finden sind die zwei Schlüsselschalter zum Freischalten des Karts sowie die zwei Kippschalter zum Einschalten. Unterhalb dieses Armaturenbretts wurde der Not-Aus-Taster befestigt. An dieser Stelle ist er im direkten Blickfeld des Fahrers und stets gut erreichbar. Die Bauteilzeichnungen von Kippschalter, Schlüsselschalter und Not-Aus konnten von Grabcad unentgeltlich heruntergeladen werden. Damit die Kabel der Schalter gebündelt abgeführt werden und die Kontakte geschützt sind, wurde eine Abdeckung für die Rückseite des Armaturenbretts mit dem 3D-Drucker gefertigt.

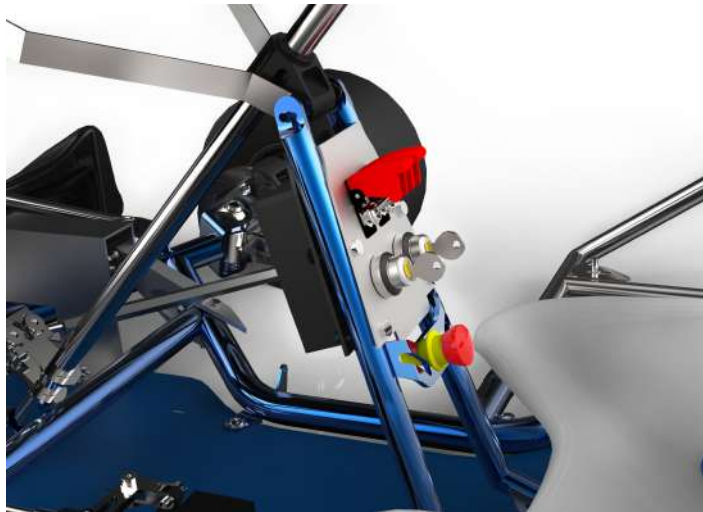


Abbildung 3.10: Armaturenbrett

### 3.2.2.5 SPS

Die SPS sitzt in einer Kunststoffbox unter der Lenkeinheit. Die einzelnen Teilzeichnungen der SPS konnten über die offizielle Seite von Sigmatek heruntergeladen werden. Anschließend wurde die SPS digital zusammengebaut, auf einer Hutschiene platziert und in der Box verschraubt.

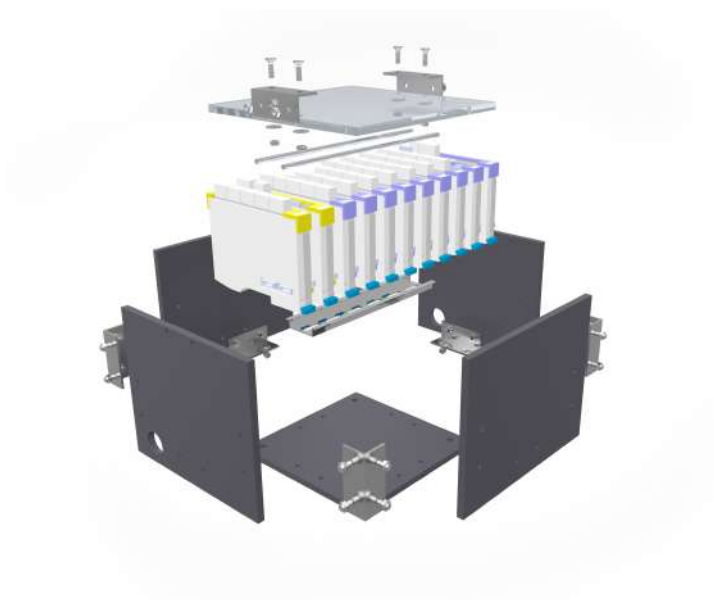


Abbildung 3.11: Aufbau der SPS

### 3.2.2.6 Verwendete Sensorik

Um die in der SPS implementierten Assistenz- und Sicherheitssysteme zu betreiben, braucht die SPS digitalisierte Messdaten vom aktuellen Zustand des Karts. Diesbezüglich sind einige Sensoren am Kart angebracht: An den beiden Vorderrädern wurden induktive Sensoren, welche die aktuelle Drehzahl messen, montiert. Dazu wurde eine kleine Adapterplatte als Befestigung gefertigt.



Abbildung 3.12: Montage des Induktiven Sensors

Damit die Stellung des Gaspedals in die SPS eingelesen werden kann, ist ein Linearpotentiometer am Gaspedal und an einem eigens gefertigten Gegenstück, welches an der Karosserie befestigt wurde, angeschraubt. Da das Linearpotentiometer sehr teuer und nicht besonders stabil ist, wurde noch eine Schutz-Verkleidung darüber befestigt. Da das von CRG erhaltene Modell des Grundrahmens im Vergleich zu dem Rahmen von CKR an der Vorderseite etwas gestaucht ist, musste diese Abdeckung für das Linearpotentiometer in der Solid Edge Version mit einem Ausschnitt versehen werden.



Abbildung 3.13: Gaspedaleinheit

Um mit der SPS auswerten zu können, wann gebremst wird, ist auf den Bremsflüssigkeitsbehältern ein Endschalter platziert, welcher im Falle einer Betätigung der Bremsen gedrückt wird und durchschaltet. Um dieses Bauteil, welches hinter dem linken Vorderrad sitzt und damit im direkten Spritz- und Steinschlagbereich ist, zu schützen, wurde eine Abdeckung in Solid Edge konstruiert und mit einem 3D-Drucker von Bürklin hergestellt.

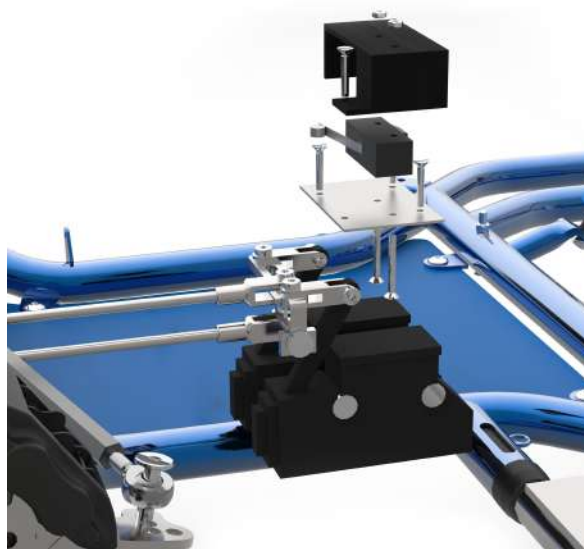


Abbildung 3.14: Endschalter der Bremse



### 3.2.2.7 Kettenradantrieb

Um die Leistung des Motors möglichst verlustfrei auf die Antriebswelle zu übertragen, wurde ein komplexes Kettensystem mit einer speziellen Spannvorrichtung konzipiert. Dabei war es eine große Herausforderung, die Kette in Solid Edge zu zeichnen. Dazu wurde zuerst ein einzelnes Kettenglied gezeichnet, und anschließend der effektive Kettenradius der Zahnräder berechnet. Anhand dieser Radien und den Verbindungslängen zwischen den jeweiligen Tangentialpunkten der Gesamt-Kettenvorrichtung wurde der effektive Pfad, welchem die Kette folgen wird, als 3D-Skizze gezeichnet und über die Musterfunktion von Solid Edge mit lauter aufeinanderfolgenden Kettengliedern gefüllt.

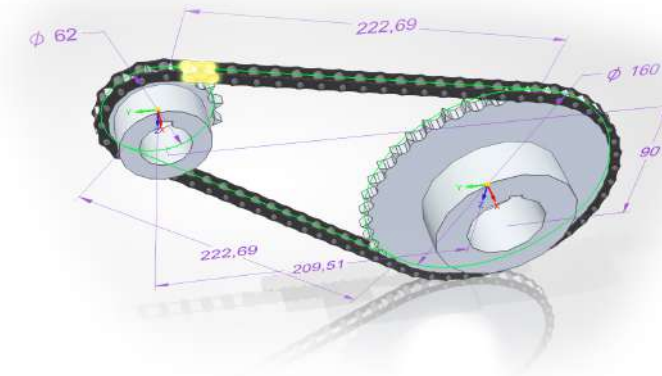


Abbildung 3.15: Screenshot aus Solid Edge: Konstruktion des Kettenmodells

Damit die Kette ruhig läuft und keine Schwingungen auftreten, wurde eine Spannvorrichtung auf der Motorhalterung platziert. Dazu wurde seitlich an der Motorhalterung eine kleine Platte angeschweißt, auf der anschließend die Spannvorrichtung befestigt wurde.





Abbildung 3.16: Kettenradantrieb

Da ein derartiger Kettenantrieb auch ein hohes Sicherheitsrisiko für frei hängende Teile (Kabel, locker hängende Kleidungsstücke, etc...) darstellen kann, wird der direkte Kontakt zum Kettenantrieb durch mehrere Abdeckungen verhindert. Als erstes wurde die Hauptabdeckung gefertigt, welche die Kette von der Seite und von oben umgibt. Um auch mit der Welle passend abzuschließen, wurde auf der Rückseite noch eine kleinere Platte dafür gefertigt und mit der Wellen-Halterung verschraubt. Um den Ausschnitt für die Spannvorrichtung an der Oberseite zu überdecken und gegen direkten Kontakt zu schützen, wurde eine kleine Abdeckung gefertigt und mit der Hauptabdeckung verschraubt.

Da beim Motorcontroller an den Kontakten eine gefährlich hohe Spannung direkt abgegriffen werden könnte, wurde dafür eine stabile Abdeckung konstruiert und darüber montiert.

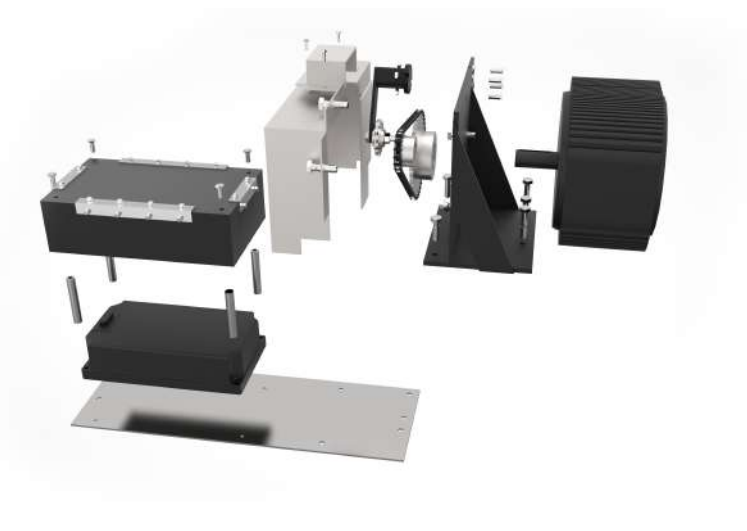


Abbildung 3.17: Aufbau der Antriebskomponenten

### 3.2.2.8 Sonstige elektrische Bauteile

Da rund um den Sitz noch einiges an Freiraum war, wurde dieser Platz genutzt, um einige elektronische Bauteile wie den DCDC-Wandler oder das Relais zu platzieren. Dieser Einbauort hat den zusätzlichen Vorteil, dass die Bauteile dort direkt auf einer gut wärmeleitenden Aluminium-Platte befestigt sind und dadurch von der am Unterboden vorbeiströmenden Luft gekühlt werden.



Abbildung 3.18: Platzierung der weiteren elektrischen Komponenten

Als Zusatzschutz wurde ein Isolationsüberwachungssystem installiert. Da dieses technische Bauteil gegen jedmögliche Einwirkung von Wasser und Schmutz gesichert werden muss, wurde dafür eine Abdeckung in Solid Edge konstruiert und von Bürklin durch einen 3D-Laserdrucker gefertigt. Befestigt wird diese kleine Box auf der Innenseite der linken Akkubox durch zwei Schrauben. Um dabei gegen Verrutschen und Verdrehen zu schützen, sind an den Seitenteilen kleine ineinandergreifende Konturen ausgefeilt worden.

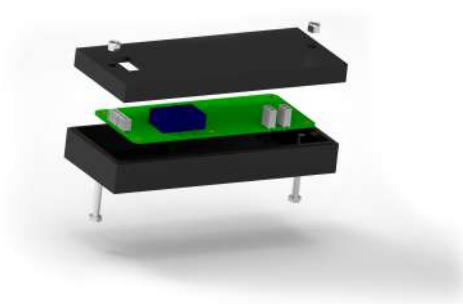


Abbildung 3.19: Aufbau Isolationsüberwachungssystem

Da das BMS für den Betrieb der Batterien ein sehr wichtiges Bauteil ist, wurde dafür eine kleine Vorrichtung zum Schutz vorgesehen. Das Modell zu dieser Vorrichtung wurde in Solid Edge entwickelt und durch einen 3D-Drucker gefertigt. (Bild siehe unter 3D Drucke)

### 3.2.2.9 Allgemeines

Nach der Fertigstellung des gesamten Zusammenbaus der Komponenten ergaben sich in Summe 1650 Einzelteile.



Abbildung 3.20: Screenshot aus der CAD-Modulationssoftware des Gesamtzusammenbaus

Speziell beim Konstruieren der 3D-Drucke gab es einige wichtige Aspekte zu beachten. Beim Abkühlen der Druckerzeugnisse kann es zu leichten Verformungen kommen, so müssen z.B. die Bohrlöcher etwas größer gezeichnet und bei genauen Kanten eine kleine Toleranz miteinbezogen werden. Des Weiteren muss man darauf achten, dass einerseits eine stabile Konstruktion erstellt wird, andererseits nicht zu dicke Flächen und Konturen gedruckt werden, da dadurch sowohl Druckzeit als auch Materialaufwand enorm ansteigen würden. Aus denselben Gründen ist es auch wichtig die Teile so zu konstruieren, dass so wenige Stütz-Vorrichtungen wie möglich benötigt werden und eventuell Fasen etwas steiler konstruiert werden, da der 3D-Drucker bis zu 60° ohne Stützvorrichtung drucken kann.

### 3.2.2.10 Liste der extern bezogenen Bauteile

Einige der verwendeten Bauteile sind industrielle Standardbauteile, deshalb konnten passende 3D-Modelle in Internet-Foren gefunden und heruntergeladen werden. Manche Komponentenzeichnungen konnten auch über die jeweiligen Hersteller direkt bezogen werden. Um die heruntergeladenen Modelle in Solid Edge verwenden zu können, mussten sie zuerst in ein Solid Edge typisches Dateiformat (.par, .asm, etc.) übersetzt werden. Damit sie auch den verwendeten Komponenten gleich sind, mussten bei einigen Modellen noch Veränderungen und Anpassungen vorgenommen werden.

Bauteil	Anbieter	Link
Batteriezeile	GrabCAD	<a href="https://goo.gl/TpVRXo">https://goo.gl/TpVRXo</a>
Bremsblöcke	GrabCAD	<a href="https://goo.gl/GO4S1R">https://goo.gl/GO4S1R</a>
Bremsscheibe	GrabCAD	<a href="https://goo.gl/Ap8dOW">https://goo.gl/Ap8dOW</a>
Diverse Normteile (Schrauben, Muttern, etc.)	Tracepart	<a href="https://goo.gl/IJESpT">https://goo.gl/IJESpT</a>
Induktiver Näherungssensor	Tracepart	<a href="https://goo.gl/ihVSTF">https://goo.gl/ihVSTF</a>
Isolationsüberwachungssystem	Bender GmbH	Anfrage per Mail
Kippschalter	GrabCAD	<a href="https://goo.gl/ous8Wb">https://goo.gl/ous8Wb</a>
Kippschalter Abdeckung	GrabCAD	<a href="https://goo.gl/eQ9B5c">https://goo.gl/eQ9B5c</a>
Klemmdose	GrabCAD	<a href="https://goo.gl/e6izAO">https://goo.gl/e6izAO</a>
Not-Aus Taster	GrabCAD	<a href="https://goo.gl/FnEumw">https://goo.gl/FnEumw</a>
Reihenklemmbock	GrabCAD	<a href="https://goo.gl/4bHZ6A">https://goo.gl/4bHZ6A</a>
Relais	GrabCAD	<a href="https://goo.gl/p4RfhH">https://goo.gl/p4RfhH</a>
Schlüsselschalter	Tracepart	<a href="https://goo.gl/Jye7KT">https://goo.gl/Jye7KT</a>
Sicherungsautomat	GrabCAD	<a href="https://goo.gl/4s7FLw">https://goo.gl/4s7FLw</a>
SPS Komponenten	Sigmathek	Anfrage per Mail
Taster MSM 22	Schurter	<a href="https://goo.gl/L9xR4L">https://goo.gl/L9xR4L</a>
Touchpanel	Sigmathek	Anfrage per Mail

Tabelle 3.3: Liste der extern Bezogenen Bauteile

## 3.3 Rendering

Durch Renderprogramme wird aus einem dreidimensionalen Modell eine Grafik erstellt. In unserem Fall wurde das 3D-Modell in Solid Edge erstellt. Als Renderprogramm wurde Keyshot 6 verwendet. Damit spezielle und kompliziertere Konstruktionen ebenfalls detailliert als Grafik dargestellt werden können, wurden von einzelnen Teilkonstruktionen Explosionszeichnungen mittels Keyshot erstellt. Dazu muss bei einem Zusammenbau, wie zum Beispiel der gesamten Lenkradeinheit, jedes einzelne Bauteil genommen und über das Positionsmenü an die richtige Stelle verschoben werden.

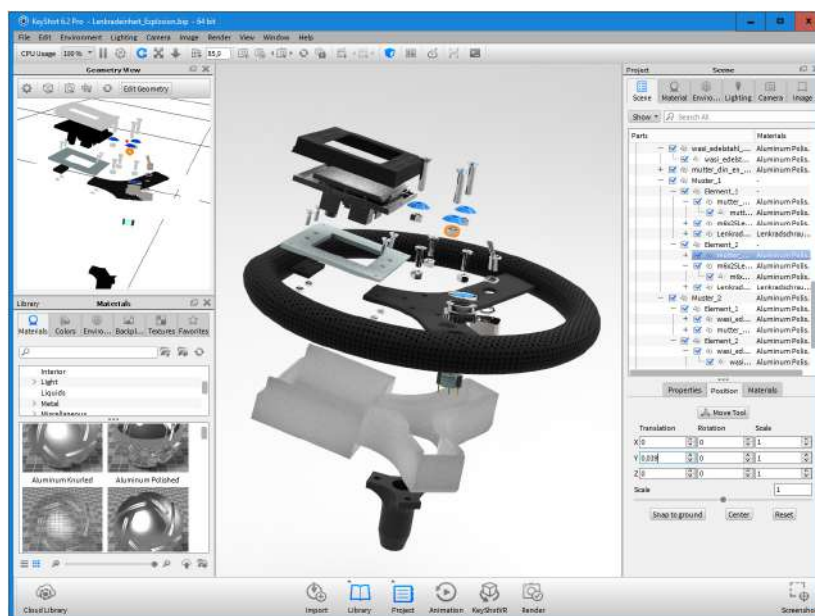


Abbildung 3.21: Screenshot aus KeyShot: Aufbau Lenkradeinheit

### 3.3.1 Rendering Gesamtmodell

Um das gezeichnete Modell aus Solid Edge in Keyshot zu importieren, ist es notwendig, den Zusammenbau in Solid Edge als .bip Datei zu speichern. Diese Bip-Datei kann anschließend mit Keyshot geöffnet werden. Da die Grundgeometrie des von CRG erhaltenen Modells, welches als Basis für alle Modifikationen diente, verschoben ist, war es zu Beginn eines jeden Renderings nötig, dies zu korrigieren und das Kart in den Geometrie-Mittelpunkt zu adjustieren. Mit dem nun importierten Modell und der zentrierten Geometrie kann begonnen werden, alle Bauteile und Modelle mit dem passenden Material und der dazugehörigen Farbe zu versehen. Dabei ist es oft notwendig, das verwendete Material auf die eigenen Bedürfnisse anzupassen, um den Realzustand des Karts genau zu digitalisieren. Nachdem allen Komponenten die passenden Materialien, Farben, Reflektions- und Absorptionseffekte zugeordnet

wurden, ist es nachfolgend notwendig, noch das richtige Lichtszenario sowie einen passenden Hintergrund auszuwählen. Um mit dem Rendern verschiedenster Blickwinkel und Motive des Karts zu beginnen, ist es notwendig, die benötigte Auflösung der Bilder festzusetzen. Dabei muss ein Kompromiss zwischen Auflösungshöhe und der Dauer für ein Rendering eingegangen werden. Als Auflösung für die Renderings des Karts wurde eine Auflösung von 5000x3530 Pixel gewählt. Diese Auflösung ist sehr hoch und benötigt folglich bei einem Rendering des gesamten Karts (alle 1651 Bauteile aktiviert) rund 45 Minuten; dadurch ist allerdings die Möglichkeit gegeben, die Bilder im Anschluss noch umfangreich zu bearbeiten und auch zu vergrößern.

### 3.3.2 Rendering der Mechanikteile

Um für die Diplomarbeit Abbildungen der eigens gefertigten Bauteile zu haben, wurden alle wichtigen Komponenten zusätzlich einzeln gerendert. Bei diesen Renderings war es eine besondere Herausforderung, das richtige Lichtszenario einzustellen, da sich die glatten Flächen sonst im direkten Licht spiegeln und damit nicht mehr klar erkennbar wären.

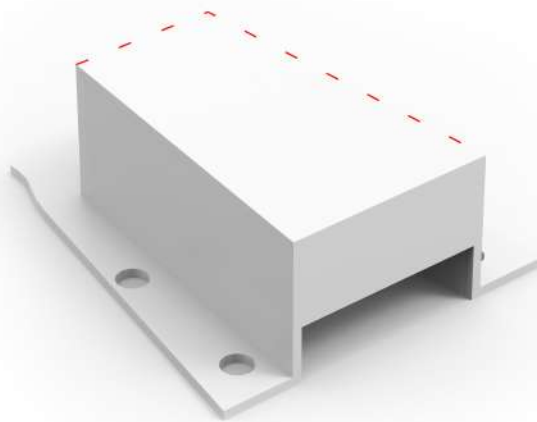


Abbildung 3.22: Spiegelproblematik bei den Einzelbauteilen

### 3.3.3 Renderings für die Visualisierung

Bei den Bildern des Karts, welche in die Visualisierung eingefügt wurden, hat sich eine besondere Herausforderung ergeben: Die Hintergrundfarbe der für die Visualisierung verwendeten Bilder wurde in einem Farbton festgelegt, welcher nicht in dem gerenderten Kart selbst vorkommt (um das Ausschneiden des Karts durch die Sigmatek Software zu ermöglichen). Für diese Hintergrundfarbe wurde ein helles Grün festgelegt. Da das verwendete Touchpanel mit 320x240 Pixel eine sehr niedrige Auflösung besitzt, mussten alle gerenderten Bilder auf die passende Größe herunterskaliert werden. Dabei haben sich die Farben der äußersten Pixel der Kartform mit der Hintergrundfarbe des Bildes vermischt, wodurch sich eine grün-schimmernde Umrandung der Form ergab. Um diesen Effekt zu unterbinden, wurden bei dem hochauflösenden Bild mit Adobe Photoshop alle Grün-Anteile mit dem Farbbereichswerkzeug ausgewählt und entfernt. Anschließend wurde das Kart-Abbild runterskaliert (ohne Hintergrund) und nachfolgend wieder mit einem hellgrünen Hintergrund versehen.

### 3.3.4 Animationen

Um das Kart visuell anschaulich und dynamisch zu präsentieren, wurden mehrere Animationen vom Zusammenbau erstellt. Dabei werden alle Komponenten in einem Film nacheinander angefügt. Dazu mussten für jedes bewegende Bauteil Translations- und Rotationspfade sowie die genauen zeitlichen Abläufe der Bewegungen definiert werden. In einem weiteren Schritt mussten die Perspektiven und die Bewegungen der Kamera ausgewählt werden. Schlussendlich ergab sich für die Hauptanimation eine Länge von 2:31 Minuten mit insgesamt 4529 Frames.

### 3.3.5 Allgemein

Nach dem Rendern wurden alle Grafiken mit Photoshop nachbearbeitet. Dabei wurden alle Bauteile zuerst freigestellt und anschließend auf einem weißen Hintergrund platziert, damit die Grafiken übergangslos in die Diplomarbeit integriert werden konnten. Dabei gilt es zu beachten, dass die Schatten und Reflektionen am Boden richtig ausgeschnitten und gegebenenfalls mit einem weichen Übergang versehen werden.



## 3.4 Plakate

Um Anschauungsmaterial des Projektes auch vor dem Fertigstellen an Messen und Ähnlichem präsentieren zu können, wurde ein Plakat in Photoshop designet und auf ein Roll-Up-Plakat gedruckt.

### 3.4.1 Plakat für die BIM

Als Hintergrund wurde ein eigenes Wallpaper mit einem dynamischen Weiß-Grau-Verlauf gezeichnet. Darauf aufbauend wurden an der Oberseite das Logo unserer Schule und Abteilung platziert. Um den Titel unserer Diplomarbeit etwas hervorzuheben, ist er in demselben sportlichen Rot gehalten, wie auch der „Elektrotechnik“ Schriftzug im Logo. Als Hauptaugenmerk dienen die Renderings des Karts in der Mitte des Plakates. Zur Information sind sowohl die Namen der Teammitglieder sowie die technischen Daten des Karts abgebildet. An der Unterseite des Plakates sind die Sponsoren des Projektes genannt. Dieses Roll-Up wurde sowohl auf der Berufsinformationsmesse, als auch beim Tag der offenen Tür der HTBLuVA-Salzburg für die Abteilung Elektrotechnik präsentiert.



Abbildung 3.23: Plakat für die BIM



## 3.5 Design

### 3.5.1 Batterieboxen

Für die Batterieboxen wurde für alle Seiten ein Design mittels Adobe Photoshop konstruiert und bei der Firma Folien Brunner gedruckt. Das gewählte Design wurde etwas an der von Mercedes-Benz seit 2010 verwendeten Designphilosophie „Aesthetics S“, deren Linienführung seither bei allen Mercedes PKW's ansatzweise erkennbar ist, orientiert. Die Findung und Entwicklung unseres Designs begann grundsätzlich auf der größten und wichtigsten Fläche, den Seitenteilen. Die Vorder- und Hinterseiten wurden anschließend dazu passend ergänzt.

Zu Beginn der Design-Entwicklung wurde das zu verwendende Farbspektrum gesucht. Dazu haben sich ähnliche wie die von CKR verwendeten Farben angeboten, da sie zum Grundrahmen des Karts passen. Basierend auf dieser Farbkonstellation wurde für den Hintergrund ein „Navy-blau“ (RGB: 9,0,139) gewählt. Um etwas Dynamik in die Optik zu bringen, wurde mit dem Plath Tool der „S-Wing“, welcher sich vorne geradlinig an der unteren Kante entlang und ab etwa der Hälfte der Seite mit einem starken Schwung nach oben zieht, dem Design hinzugefügt. Um dem gesamten Design einen stimmigen Eindruck zu verschaffen, wurde der Anstiegswinkel des S-Wings so gewählt, dass er parallel zur Halterung verläuft. Die Kontur des S-Wing ist in einem schlicht hervorstehenden „dark-blue“ (RGB: 3,8,79) gehalten. Damit dieser Figur eine dynamische Tiefe verliehen wird, wurde eine helle Shape-Line in „light-blue“ (RGB: 12,3,139) an der inneren Unterseite des S-Wings hinzugefügt. Damit der Betrachter des Karts dieses sofort mit Geschwindigkeit assoziiert, wurden von links oben beginnende Streamlines gezeichnet. Diese Linien wurden in einem sportlich aggressiv wirkenden Rot (RGB: 216,0,29) eingefärbt. Diese dynamischen Linien beginnen sich nach etwa dem ersten Drittel der Fläche mit einem Faktor von 1,5 auszudehnen, wodurch ein natürlicher Fließ- und Verwirbelungseffekt ausgedrückt wird. Da die Streamlines in aufsteigender Reihenfolge eine abnehmende Dicke von 8 bis 2 Pixel aufweisen, wurde dieser Figur eine dreidimensionale Tiefenwirkung gegeben.

Als weiteres designtechnisches Element wurde der mittige Sponsor-Block kreiert. Diesem wurde eine muskulös und kantig wirkende Form gegeben, welche durch ihre steilen Winkel ideal zu dem sportlichen Design passen. Die Winkel wurden so gewählt, dass die Seiten parallel zu den Konturen des S-Wings verlaufen, wodurch sich dieses Designelement besser in das Gesamtdesign integriert. Als Hintergrundfarbe wurde ein Grau-Weiß-Verlaufen mit einer leichten Struktur gewählt, wodurch die Sponsorblöcke wie aus gebürstetem Aluminium wirken.

### 3.5.1.1 Vorderseite Batterieboxen

Grundsätzlich wurden an der Vorderseite die selben Designelemente wie an den Seitenteilen weitergeführt. Der S-Wing läuft in eine dezente Linie aus. Der Sponsor-Block ist auf der Front der Batteriebox, ident zu den Seitenflächen, nur in einer kleineren Version und mit dem HTBLuVA-Salzburg Logo darauf abgebildet, vorzufinden. Die Streamlines münden in einer Spitze, was diesem Element weitere Sportlichkeit gibt. Zudem werden sie im gleichen Winkel der Kontur des Sponsor-Blocks ausgeführt, wodurch das Design des Vorderteiles einheitlicher wirkt.

### 3.5.1.2 Rückseite Batterieboxen

Auf der Rückseite ist ein Anti-S-Wing zu finden, welcher eine einheitliche Weiterführung der Konturen des Seitenteils ergibt. Des Weiteren werden die Streamlines ebenfalls weitergeführt und immer weiter ausgedehnt.

### 3.5.1.3 Innenseite Batterieboxen

Um die gesamte Optik der Batterieboxen etwas runder wirken zu lassen, wurde auch die Innenseite der Boxen mit Folie beklebt, auf der lediglich die Hintergrundfarbe (Navy-Blau) abgedruckt ist. Da die Zwei Batterieboxen unterschiedliche Abmessungen besitzen, war es notwendig, das Design für die zweite Seite individuell zu adaptieren. Allgemein ist ein sehr schönes, elegantes und vor allem sportliches Design entstanden, welches nicht nur die Blicke auf sich zieht, sondern sich auch wunderbar in das Gesamtkonzept des Rennkarts fügt.

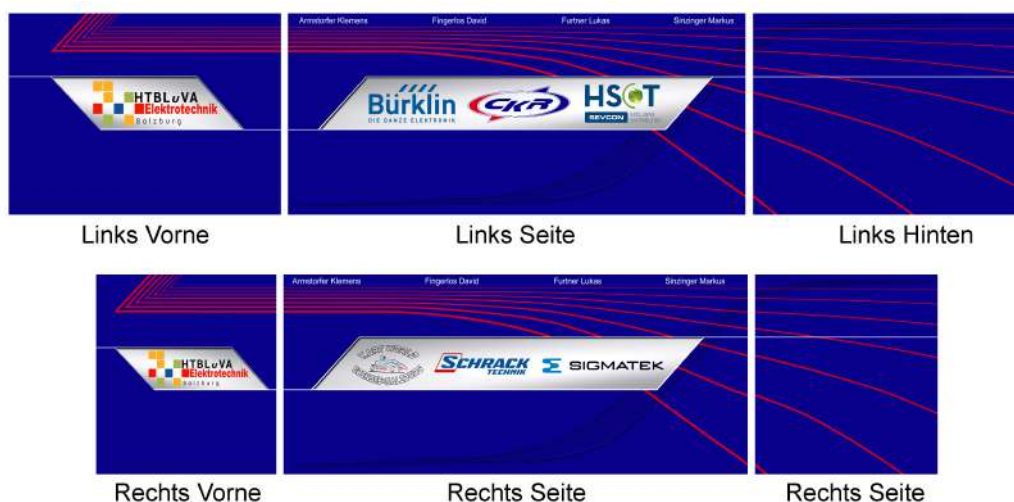


Abbildung 3.24: Design der Folien für die Batterieboxen

### 3.5.2 Vorderverkleidung des Karts

Um dem Kart auch frontal betrachtet einen sportlichen Look zu verleihen, wurde für die Vorderverkleidung eine Folie designed und gedruckt. Dieses Design wurde basierend auf denselben designtechnischen Merkmalen wie die Verkleidung der Seitenkästen entwickelt. Dazu wurde als Hintergrund wieder das Navy-Blau (RGB: 9,0,139) verwendet und die Streamlines von den Seitenkästen kopiert. Diese Streamlines wurden so angeordnet, dass sie entlang der Längsrichtung des Karts verlaufen, am unteren Ende der Frontverkleidung in einem Pfeil münden und über die Vertikallinie gespiegelt sind.

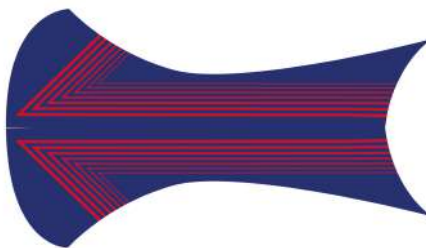


Abbildung 3.25: Design der Folie für die Vorderverkleidung

Diese Folie wurde letztendlich nicht am Kart angebracht, da das dafür bestimmte Chassis-Teil einige Wölbungen und Einkerbungen hat und deshalb ein glattes Anbringen nicht möglich war.

#### 3.5.2.1 Fahrzeugkennzeichen

Um dem Kart den Look eines straßenzugelassenen Fahrzeugs zu geben und gleichzeitig die Abteilung Elektrotechnik der HTBLuVA-Salzburg zu promoten, wurden für Front- und Heckspoiler des Karts Nummerntafeln mit der Aufschrift „Elektrotechnik“ gedruckt und am Kart angebracht. Dazu wurde das Design behördlicher Kennzeichen von für Salzburg zugelassenen Kraftfahrzeugen herangezogen und adaptiert. Grundsätzlich wurden die gleichen designtechnischen Merkmale wie bei normalen Fahrzeugen verwendet. Diese wurden allerdings verändert und zum Teil bezüglich der Maße etwas transformiert. Zuerst wurde für den Hintergrund ein strahlendes Weiß festgelegt. Als Umrandung dienen, wie auch bei einem regulären Autokennzeichen, zwei Farbbalken (an der Ober- und der Unterseite) in Rot-Weiß-Rot. Als weiteres Merkmal wurde auf der linken Seite ein blauer Balken mit einem A (für Austria) und dem Sternkreis der EU hinzugefügt. Um einen authentischen Eindruck zu hinterlassen, wurde die Schriftart (Euro Platte - was eine Variante der FE-Schrift ist), welche auf den Nummerntafeln verwendet wird, heruntergeladen FUBNOTE und dem S (für Salzburg) sowie dem Schriftzug Elektrotechnik zugeordnet. Abschließend wurde noch als Abtrennung zwischen der Statutarstadt und dem Vormerkzeichen

das Salzburger Wappen mit darunter liegendem „Salzburg“-Schriftzug eingefügt.<sup>2</sup>



Abbildung 3.26: Design der Folie des Fahrzeugkennzeichens

### 3.5.2.2 Druck

Beim Drucken der Folien war allgemein zu beachten, dass der Druck mit dem CMYK (subtraktivem) Farbmodell und nicht dem standardmäßigem RGB-Farbmodell ausgeführt wird und deshalb eine Farbtransformation nötig ist. Des Weiteren ist es notwendig, dass der Platz auf der bedruckten Fläche möglichst gut ausgenutzt wird und anschließend nicht zu viel Verschnitt anfällt. Deshalb wurden kleinere unbenutzte Flächen mit Logos von Sponsoren und Partnern versehen, welche am Ende eventuell noch an leeren Flächen angebracht werden können.<sup>3</sup>

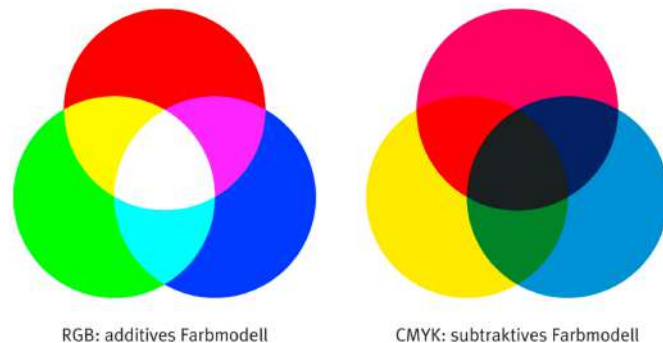


Abbildung 3.27: Gegenüberstellung der Farbmodelle

<sup>2</sup>vgl.: Autokennzeichen-Schriftart, <https://goo.gl/Sc5dwN> (Stand: 23.03.2017)

<sup>3</sup>vgl.: Farbmodelle, <https://goo.gl/7lWXeu> (Stand 23.03.2017)

## 3.6 Icons für die Visu

### 3.6.1 Statusicons

Diese Icons sind am Statusscreen der Visualisierung ersichtlich und dienen der Kontrolle bzw. der Veranschaulichung des aktuellen Zustands einiger Komponenten. Zu diesen zählen:



Abbildung 3.28: Aufstellung aller Statusicons

#### 3.6.1.1 Entwicklung des Grunddesigns

Vorweg wurden einige Button Symbole aus dem Internet begutachtet, um eine Inspiration für das Design der Statusfläche zu erhalten. Als Grundform des Symbols dient eine runde Fläche in der Mitte, auf welcher anschließend die entsprechenden Icons eingefügt werden. Um dieser Fläche einen Tiefeneffekt zu geben, wurde als Umrandung der Mittelfläche ein kleiner, in einer etwas dunkleren Farbe als derjenigen der Mittelfläche aufscheinender Ring eingefügt. Als weiteres Design-Element wurde ein helles Kreissegment an der Unterseite des Kreises eingefügt; dieses Segment ergibt einen Spiegeleffekt. Der äußere Ring wurde in einem metallisch wirkenden Silber mit einem dynamischen Tiefeneffekt gewählt.



Abbildung 3.29: Symbol des Schlüsselschalters

Basierend auf diesem Grunddesign des Symbols wurden für jedes einzelne Icon verschiedene Veränderungen vorgenommen.

### 3.6.1.2 ASR Symbol

Um das Kontroll-Icon für die Anti-Schlupf-Regelung festzulegen, wurde zuerst die Form des typischen ASR-Symbols, welches auch in jedem PKW zu finden ist, im Internet recherchiert. <sup>4</sup> Dieses Symbol wurde anschließend etwas modifiziert und auf dem Symboldesign mit schwarzem Hintergrund platziert. Dieses Symbol gibt es in zwei Ausführungen: einmal als ASR-Ein mit schwarzem Hintergrund und gelben ASR Icon und einmal als ASR-Aus mit schwarzem Hintergrund und dunkelgrauen ASR Icon.

### 3.6.1.3 Motorcontrollersymbol

Um zu verdeutlichen, dass mit diesem Statussymbol der aktuelle Zustand des Motorcontrollers und damit der Motoransteuerung veranschaulicht wird, wurde ein großes M auf dem Basissymbol platziert. Dieses Symbol gibt es in zwei Ausführungen: zum einen als Motorcontroller-Ein mit grünem Hintergrund und zum anderen als Motorcontroller-Aus mit rotem Hintergrund.

### 3.6.1.4 Schlüsselsymbol

Um ein typisches Schlüssel Icon zu verwenden, wurde eine Schlüsseldarstellung mit freier Lizenz aus dem Internet heruntergeladen <sup>5</sup>. Das heruntergeladene Schlüssel Icon wurde anschließend mittig auf dem Grundsymbol platziert. Dieses Symbol gibt es in zwei Ausführungen: einmal als „Schlüsselschalter ist aktiv“ mit grünem Hintergrund und einmal als „Schlüsselschalter ist inaktiv“ mit schwarzem Hintergrund.

### 3.6.1.5 Softkeys

Für die Steuerung der Visualisierung über das Touchdisplay wurden einige Softkeys benötigt, welche am Bildschirm angezeigt werden und durch Drücken eine Aktion ausführen. Diese wurden mit Photoshop erstellt. Zu diesen zählen:



Abbildung 3.30: Aufstellung der Softkeys

<sup>4</sup>vgl.: Fahrdynamikregelung, <https://goo.gl/17G4rA>, (Stand 21.03.2017)

<sup>5</sup>vgl.: Autoschlüssel Kostenlose Icons, <https://goo.gl/77djVI>, (Stand 16.03.17)

### 3.6.1.6 Fahrmodus

Über diesen Softkey ist es möglich, in den normalen Fahrmodus zu wechseln. Um dies bildlich zu veranschaulichen, wurde eine „Chequered Flag“, wie sie aus etlichen Rennserien bekannt ist und welche mit Fahren beziehungsweise Rennfahren assoziiert wird, gezeichnet. Um dieser karierten Flagge eine dynamische Wirkung zu verleihen, wurden die Pixel im Mittelteil etwas nach unten verschoben, wodurch es den Anschein hat, als würde die Flagge geschwungen werden.

### 3.6.1.7 Fehleranzeige

Mit diesem Softkey ist es möglich, in die Fehleranzeige zu wechseln. Als Symbol für etwaige Fehler wurde ein oranges Dreieck mit einem darin integrierten weißen Rufzeichen gewählt. Das soll den Ausruf „Achtung“ signalisieren.

### 3.6.1.8 Quittieren

Dieser Softkey erlaubt eine Quittierung von aufgetreten, aber bereinigten Fehlern. Da es sich dabei thematisch wieder um den Fehlerscreen dreht, wurde das Symbol der Fehleranzeige übernommen. Um aber eine klare Differenzierung dieser zwei Symbole zu gewähren, ist das Symbol in einem dunklen Rot anstelle des Oranges eingefärbt worden. Damit die Funktion des Quittierens mit einem Symbol veranschaulicht werden kann, ist unterhalb des Symbolen ein grünes Häkchen. Das soll ein ‚Akzeptieren‘ darstellen.

### 3.6.1.9 Rennmodus

Durch diesen Softkey kann in den Rennmodus gewechselt werden. Hierzu wurde das Symbol des normalen Fahrmodus übernommen. Die Flagge wurde dabei in einem sportlichen Orange gehalten.

### 3.6.1.10 Status

Über diesen Softkey ist es möglich, in das Statusmenü, in welchem alle wichtigen Informationen des Karts einsehbar sind, zu wechseln. Ein typisches Symbol für Überwachung und Kontrolle stellt das Auge dar, welches auch in unserem Fall herangezogen wurde.





# Kapitel 4

## Hardware

### 4.1 Elektromotor

Eine der wohl wichtigsten Komponenten dieses Projekts ist der Elektromotor. In dieser Anwendung handelt es sich dabei um einen permanenterregten Dreiphasen-Synchronmotor.

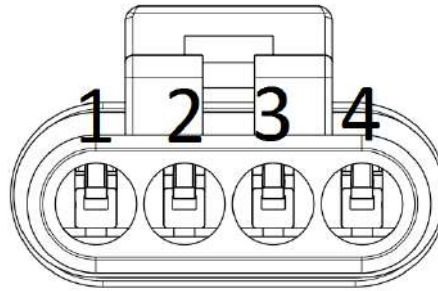
Hersteller	Motenergy
Motorbezeichnung	ME1507
Spannung	100VDC
Nennleistung	17kW
Spitzenleistung	50kW
Nenndrehzahl	3800rpm
Maximaldrehzahl	6000rpm
Nenndrehmoment	50Nm
Spitzendrehmoment	85Nm
Polpaarzahl	5
Gewicht	19,4kg
Encodertyp	Sinus-Cosinus

Tabelle 4.1: Technische Daten des Motors[19]

Die bei diesem Projekt verwendete Maschine besitzt einen Rotor mit Permanentmagnet. Somit kann auf Schleifringe verzichtet werden, was eine etwaige Wartung unnötig macht.

Der Motor hat ein geschlossenes Gehäuse mit außen angebrachten Kühlrippen. Im Motor befinden sich zwei Temperatursensoren des Typs KTY84-130, welche über das Signalkabel angeschlossen werden können. Der sich im Motor befindliche Sinus-Cosinus Encoder wird extern über den Motorcontroller mit Spannung versorgt. Dadurch ist auch im Stillstand jederzeit die Position der Welle bekannt. Als Stecker für den Encoder und die Temperatursensoren werden 2x 4 polige „ampsuperseal“-Stecker von TE-Connectivity verwendet.

Stecker	Pin	Anschluss
1	1	Temp. Sensor 1
	2	Temp. Sensor 2
	3	Gnd. Sensor
	4	Schirm
2	1	+5 V
	2	Encoder - Sinus
	3	Encoder – Cosinus
	4	Ground



Pinbelegung der Stecker

Stecker des Motors

### 4.1.1 Technische Erklärung

Die Synchronmaschine ist eine Drehfeldmaschine, bei der sich Rotor- und Statorfeld mit gleicher Drehzahl drehen. Sie laufen also synchron zueinander.

Synchronmaschinen finden in vielen Bereichen Verwendung, sowohl als Generator als auch als Motor. In der elektrischen Energieerzeugung werden Generatoren fast ausschließlich als Synchronmaschinen ausgeführt. In der Industrie werden sie in Verbindung mit Frequenzumrichtern als sogenannte Servomotoren überwiegend in Werkzeugmaschinen und automatisierten Anlagen verwendet.

Synchronmaschinen bestehen mechanisch aus 2 Hauptkomponenten: Dem Stator und dem Rotor. Der Stator wird stets geblecht und meist mit Drehstromwicklung ausgeführt, dabei werden die Wicklungsdrähte in eingestanzte Nuten gelegt. Die Bauform des Rotors variiert stark bei den unterschiedlichen Maschinentypen. Sie gibt jeder Maschine das charakteristische Betriebsverhalten. Bei der momentenlosen Synchronmaschine liegen die Pole von Stator und Rotor übereinander, die Drehzahlen sind gleich. Soll ein Drehmoment entstehen, müssen die Pole des Rotors um einen gewissen Winkel gegenüber den Statorpolen vor- bzw. nacheilen. Bei der feldorientierten Regelung der Synchronmaschine wird dieser Winkel durch die geeignete Bestromung der Ständerspulen vorgegeben. Eine Synchronmaschine kann normalerweise nicht aus dem Stillstand gestartet werden, da sie nur mit ihrer Synchrondrehzahl betrieben werden kann. Daher wird ein Wechselrichter benötigt, welcher die Statorfrequenz und somit die Synchrondrehzahl von Null weg konstant auf die gewünschte Drehzahl erhöht.<sup>6</sup>

<sup>6</sup>Vgl.: Fischer, Rolf: Elektrische Maschinen, 16. Auflage. München: Hanser Verlag, 2013

### 4.1.2 Problemstellung

In industriellen Anlagen werden sehr viele Synchronmaschinen, auch bekannt als Servomotoren, verwendet. Da aber solche Anlagen überwiegend stationär verbaut sind, werden solche Synchronmotoren hauptsächlich mit 400 Volt bzw. 230 Volt betrieben. In diesem Fall wäre solch eine Spannung jedoch nicht möglich, da sie erheblichen Aufwand hinsichtlich Sicherheit mit sich bringt. Bei einer Speicherung von 230V oder sogar 400 V in einem Akku besteht akute Lebensgefahr. Wird jedoch eine industriell genutzte Synchronmaschine mit einer geringeren Spannung betrieben, sinkt ihre Leistung und somit Drehmoment und Drehzahl. Eine Alternative wäre die Umwandlung von einer geringen Batteriespannung auf 400 V mittels Wechselrichter. Jedoch werden für unsere verwendete Leistung keine Wechselrichter in kompakter Bauweise angeboten. Die Lösung für dieses Problem wurde im Zuge einer Internetrecherche gefunden. Einige Hersteller bauen spezielle Elektromotoren extra für Elektrofahrzeuge. Diese Motoren können mit einer Batteriespannung <100 Volt problemlos betrieben werden.

## 4.2 Motorcontroller

Das Betriebsverhalten von permanenterregten Synchronmaschinen hängt stark von der Ansteuerung ab. Da bei dem von uns verwendeten Motor das Rotorfeld konstant bleibt, kann lediglich das Statorfeld in Drehrichtung, Frequenz und Größe verändert werden. Diese Veränderungen werden mittels Frequenzumrichter erzielt. Solch eine Antriebskombination wird auch Servomotor genannt.

Der von uns verwendete Controller wurde vom Hersteller Sevcon gebaut und trägt die Bezeichnung Gen4 Size 6 – 8055. Dieser Controller wurde speziell für 3-Phasen Drehstrommotoren gebaut, speziell für permanenterregte Synchronmotoren. Die Bezeichnung Size 6 gibt die Größe an, und damit auch, welche Dauerleistung möglich ist. In diesem Fall wären das 220 Ampere Dauerstrom und somit 22 kW bei 100 Volt Batteriespannung, natürlich nur bei ausreichender Kühlung.

Der Motorcontroller kann den Motor so ansteuern, dass ein beliebiges Drehmoment sowie eine daraus resultierende Drehzahl entstehen. Damit kann vom Stillstand bis zum Maximalpunkt jeder Betriebspunkt gefahren werden. Bei Bedarf ergibt sich damit eine lineare M-n Charakteristik, was große Vorteile beim Beschleunigen mit sich bringt. Auch sind so sowohl Motorbetrieb als auch Generatorbetrieb im Rechts- und Linkslauf möglich. Die Motorposition wird über einen Sinus/Cosinus Encoder erfasst. Der verwendete Controller mit der Seriennummer 8055 ist speziell für Sinus – Cosinus Encoder ausgelegt. Andere verfügbare Arten von Encodern wären ein Inkrementalgeber und Hall-Sensoren. Gesteuert wird der Motor über eine feldorientierte Drehmomentenregelung. Das bedeutet, dass der Controller abhängig von der Gaspedalstellung ein Drehmoment im Motor erzeugt.

Hersteller	Sevcon
Typenbezeichnung	Gen4 Size 6
Nennspannung	72V-80VDC
Maximalspannung	116V DC
Dauerstrom	220 A
Maximalstrom	660 A

Tabelle 4.2: Technische Daten[24]

## 4.2.1 Funktionsweise

Über den Encoder des Motors wird die Position des Rotors ermittelt. Über diese Position können Rückschlüsse auf die Richtung des Rotorfelds gezogen werden. Um ein maximales Drehmoment im Motor zu erzeugen, müssen der Flusszeiger des Rotorfelds und der Stromzeiger, bezogen auf ebenfalls dieses Rotorfeld, rechtwinklig aufeinander stehen. Zur Erzeugung eines richtig gerichteten Statorfelds muss der Controller im Ständer des Motors einen bestimmten Strom einprägen. Durch Veränderung des Statorfelds in Größe oder Winkel kann das Drehmoment reduziert werden. Solch eine Regelung wird auch als „feldorientierte“ Regelung bezeichnet. Großer Vorteil ist hier, dass der Controller die feldorientierte Regelung bereits implementiert hat und somit selbstständig ausführt. Das lindert den Programmieraufwand enorm, es muss lediglich ein Analogwert zur Beschleunigung vorgegeben werden.<sup>7</sup>

Auf dem Motorcontroller ist bereits eine Buchse des Typs „ampseal“ montiert. Diese Buchse besitzt 35 Pins, jedoch werden nur 21 davon verwendet. Die folgende Tabelle zeigt die Pinbelegung auf.

Pinnummer	Bezeichnung	Angeschlossenes Gerät
1	Key Switch	Steuerrelais Ausgang 1
2	CAN Term.	Pin 24 – CAN Low
3	Contactor 1	Hauptrelais A2
4	Contactor 1	Hauptrelais A1
15	Encoder – 0V	Motor Stecker 2/ Pin 4
16	CAN High	CP111 – X1/3 CAN A
18	Dig. In 1	TO161 – 9
19	Dig. In 3	TO161 – 11
20	Dig. In 5	TO161 – 13
21	Encoder - Sine	Motor Stecker 2/ Pin 2
22	Analog In 1	AO081 – X1/1
24	CAN Term.	Pin 2 – CAN High
26	Encoder - +5V	Motor Stecker 2/ Pin 1
27	CAN Low	CP111 – X1/4 CAN B
30	Dig. In 2	TO161 – 10
31	Dig. In 4	TO161 – 12
32	Dig. In 7	DI202 – 14
35	Encoder - Cosine	Motor Stecker 2/ Pin 3

Tabelle 4.3: Pinbelegung des Steckers

<sup>7</sup>Vgl.: Fischer, Rolf: Elektrische Maschinen, 16. Auflage. München: Hanser Verlag, 2013

Im Controller sind sowohl Leistungselektronik als auch Steuerung verbaut. Das erleichtert das Zusammenspiel von SPS und Controller. Als Kommunikation zwischen den Komponenten werden ein mehrpoliges Kabel und der CAN-Bus genutzt. Die Befehle von der Steuerung zum Controller werden über Kabel übertragen. Der Motorcontroller sendet seinen Status sowie Temperatur und Drehzahl per CAN retour. Folgende Signale werden dem Controller von der SPS übergeben:

#### **4.2.1.1 Digital In 1 – Key Switch**

Das ist der Hauptschalter des Controllers. Durch Einschalten des Key- Switch wird das Hauptrelais geschlossen und die Leistungselektronik mit Spannung versorgt.

#### **4.2.1.2 Digital In 2 – Forward Switch und Digital In 3 – Reverse Switch**

Diese beiden Eingänge geben an, in welche Richtung gefahren werden soll. Dabei muss beachtet werden, dass jeweils nur ein Eingang aktiv ist, da sonst der Controller in den Fehlerzustand wechseln würde.

#### **4.2.1.3 Digital In 4 - Brake Switch**

Über diesen Schalter wird dem Controller mitgeteilt, ob gebremst wird. Bei einer Bremsung schaltet der Controller von Motor- auf Generatorbetrieb um. Es wird also zurückgespeist.

#### **4.2.1.4 Digital In 5 und Digital In 7 - Driveability Select 1 & 2**

Mit diesen beiden Eingängen wird dem Controller mitgeteilt, in welchem Modus gefahren werden soll. So ist Driveability Select 1 der gedrosselte Modus, Driveability 2 wird für den entdrosselten „Boost“ Modus verwendet.

## 4.2.2 Problemstellung

Durch die eingebaute Steuerung im Controller war anfangs das Zusammenspiel zwischen SPS und Controller schwierig. Auch die Potentialdifferenz zwischen SPS (24V) und Controller (96V) war ein Hindernis. Glücklicherweise sind die digitalen Eingänge des Motorcontrollers „Active-Low“, sie werden also gesetzt, wenn der Eingang auf GND-Potential gezogen wird. Das konnte durch einen Impedanzwandler in Form eines Optokopplers erreicht werden. Im Unterkapitel Schaltpläne ist der Impedanzwandler näher beschrieben.

Ebenfalls Probleme machte das Hauptrelais. Der Controller ist so programmiert, dass er auch nur beim geringsten Verzug zwischen Aktivierungssignal und Einschalten des Relais sofort einen Fehler ausgibt. Daher wurde das Relais direkt angeschlossen, jedoch ein zweites Halbleiterrelais in die Ansteuerleitung als Unterbrecher geschaltet.

## 4.3 Steuerung

Als Steuerung unseres Karts verwenden wir eine SPS des Herstellers Sigmatek. Auf dieser werden das gesamte Programm zur Ansteuerung des Motorcontrollers, die verschiedenen Regler sowie die gesamten Sicherheitsmechanismen implementiert.

### 4.3.1 CP111 & PS101

Das Modul CP111 ist der Hauptrechner unserer Steuerung. Hier wird das Programm verarbeitet. Das PS Modul dient hier der CPU als Stromversorgung. Auf den beiden Modulen befinden sich insgesamt 2 Anschlüsse für den CAN-Bus, eine Varan-Schnittstelle sowie zwei Ethernet-Ports. Der Varan-Bus ist ein eigens von Sigmatek entwickeltes Bus-System. Er wird verwendet, um mehrere dezentrale CPUs miteinander zu verbinden. Da in dieser Anwendung lediglich eine CPU zum Einsatz kommt, wird er nicht benötigt. Für unsere Anwendung sind besonders die beiden CAN-Schnittstellen wichtig, da über diese die SPS mit dem BMS und dem Motorcontroller verbunden wird.

Modul	CP111		
Klemme	Pin	Bezeichnung	Ang. Gerät
	1	CAN A	Display
	2	CAN B	Display
X1	3	CAN A	Controller
	4	CAN B	Controller
	5	Can Gnd.	Masse
	1	+24 V	KL181
X2	2		
	3	Ground	KI180

Tabelle 4.4: Pinbelegung



### 4.3.2 KL180 und KL181

Diese beiden Module sind Klemmmodule, daher werden sie für die Spannungsversorgung aller Steuerungskomponenten verwendet. Dabei wird das KL 180 mit 0V und das KL 181 mit +24V versorgt. Diese Module bieten zusätzlich die Möglichkeit, zu erkennen, wie viele Geräte angeschlossen sind.

Modul	KL180		Modul	KL181	
Klemme	Pin	Bezeichnung	Klemme	Pin	Bezeichnung
X1	1	GND Display	X1	1	+24V Display
	2	LED Taster Lenkrad		2	Taster Lenkrad
	3	Drehzahlsensor 1		3	Fahrtrichtungsschalter
	4	—		4	Drehzahlsensor 1
X2	1	Drehzahlsensor 2	X2	1	Drehzahlsensor 2
	2	—		2	Bremsschalter
	3	AI043 X3 R-		3	—
	4	AI043 X2 R-		4	—
X3	1	AI043 X1 R-	X3	1	—
	2	Isolationsüberwachungssystem		2	—
	3	Impedanzwandler		3	TO161 24V
	4	—		4	AO081 24V
X4	1	TO161 0V	X4	1	SCP111 24V
	2	AO081 0V		2	SDM081 24V
	3	SCP111 0V		3	Isolationsüberwachungssystem
	4	SDM081 0V		4	Impedanzwandler
X5	1	PS101 0V	X5	1	PS101 24V
	2	—		2	—
	3	Versorgung		3	Versorgung
	4			4	

Tabelle 4.5: Pinbelegung KL-Module

### 4.3.3 DI202 und TO161

Der Name DI steht für „Digital Input“, daher handelt es sich dabei um ein digitales Eingangsmodul. Die Besonderheit dieses Moduls besteht darin, dass die ersten vier Eingänge sogenannte „Counter“-Eingänge sind, also eine Zählfunktion aufweisen. Sie besitzen unabhängig vom Hauptprogramm eine Zykluszeit von  $0,25\mu\text{s}$ , wodurch sie schnelle Taktflanken erkennen. An diese Zählereingänge werden die Drehzahlsensoren der Vorderräder angeschlossen. Des Weiteren werden über dieses Modul einige Schalter und Taster sowie die Rückmeldung des Controllers eingelesen. Das Modul TO161 ist ein digitales Ausgangsmodul. Hier wird hauptsächlich die Kommunikation mit dem Motorcontroller abgewickelt. Über gesamt sechs Leitungen werden Fahrtrichtung, Fahrmodus und Fahrerlaubnis übertragen.

Modul	DI202	
Klemme	Pin	Ang. Gerät
X1	1	Ind. Sensor Links
X1	2	Ind. Sensor Rechts
X2	1	ASR – Einschaltknopf
X2	2	Fahrtrichtungsschalter
X2	3	Endschalter Bremse
X3	1	BMS Pin 2 / Übertemperatureingang
X3	2	Isolationsüberwachungssystem
Modul	TO161	
Klemme	Pin	Bezeichnung
X1	1	ASR Einschaltknopf – Beleuchtung
X3	1	Controller Pin 18
X3	2	Controller Pin 19
X3	3	Controller Pin 20
X3	4	Controller Pin 30
X4	1	Controller Pin 31
X4	2	Controller Pin 32
X5	3	+24 V
X5	4	GND

Tabelle 4.6: Pinbelegung der digitalen Module

#### 4.3.4 AI043 & AO081

Diese beiden Module sind Analogmodule. Das Modul AI043 ist ein Eingangsmodul mit vier Eingängen. Eingelesen werden hier das Gaspedal, die Temperatursensoren des Motors und der Akkustand der Batterien. Dabei wäre das Einlesen über eine Dreileiter- oder eine Zweileiter-Messmethode möglich. Alle Analogsensoren werden mittels Zweileiter-Messmethode eingelesen, als zweiter Leiter dient Masse (GND). Über das analoge Ausgangsmodul AO081 wird lediglich ein Wert für das Gaspedal an den Motorcontroller übertragen.

Im Programm „Lasal-Class“ können bei den beiden Modulobjekten mehrere Einstellungen getroffen werden. Über einen maximalen und einen minimalen Widerstandswert werden die Messbereichsgrenzen eingestellt. Sobald diese Grenzen verlassen werden, wird selbstständig ein Fehler ausgegeben.

<b>Modul</b>			<b>AI043</b>		
Klemme		Pin	Angeschlossenes Gerät		
X1		1 - 3	Gaspedal		
X2		1 - 3	Temperatursensor Motor 1		
X3		1 - 3	Temperatursensor Motor 1		
X4		1 - 3	BMS – Pin 3 / Akkustand der Batterien		
<b>Modul</b>			<b>AO081</b>		
Klemme		Pin	Angeschlossenes Gerät		
X1		1 - 2	Motorcontroller Pin 22 – Analog input Throttle		

Tabelle 4.7: Angeschlossene Geräte der Analogmodule

### 4.3.5 SCP111 und SDM081

Diese beiden Module sind für die Sicherheit zuständig. Das Modul SCP111 besitzt eine eigene CPU, wodurch es unabhängig von der Haupt-CPU arbeiten kann. Dadurch kann auch trotz eines Fehlers im Hauptprogramm das Sicherheitsprogramm von der SCP weiter ausgeführt werden. Durch das Modul SDM081 stehen sechs sichere digitale Eingänge und zwei sichere Ausgänge zur Verfügung. Über diese werden alle sicherheitstechnisch relevanten Eingabegeräte wie z.B. der Not-Aus Taster oder die Schlüsselschalter angeschlossen. Auch das Steuer- und Hauptrelais für den Motorcontroller werden hier gesteuert. Eine genauere Auflistung aller sicherheitsrelevanten Funktionen kann im Kapitel Sicherheitskonzept nachgelesen werden.

Modul	SDM081	
Klemme	Pin	Angeschlossenes Gerät
X1	1 - 4	NOT-Aus Taster
X2	1 - 4	Schlüsselschalter 1 & 2
X3	1	BMS Fehlerausgang
X3	2	Steuerrelais Kontakt 14
X4	1	Steuerrelais A1
X4	2	Halbleiter-Trennrelais A1

Tabelle 4.8: Angeschlossene Geräte der Sicherheitsmodule

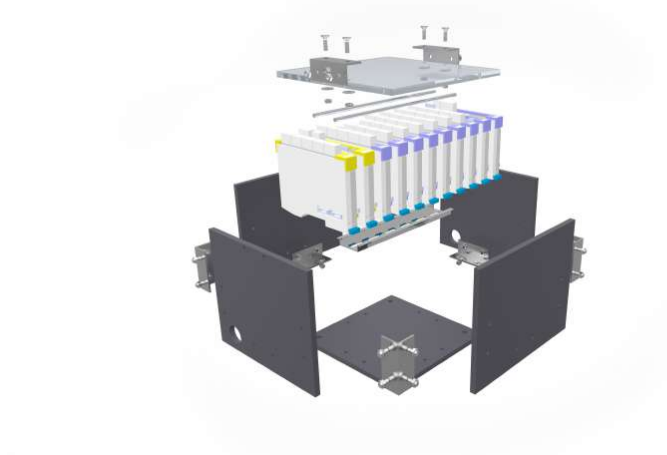


Abbildung 4.1: 3D Ansicht der SPS-Explosionszeichnung

## 4.4 Weitere elektrische Bauteile

Folgende Tabelle zeigt die verwendeten elektronischen Bauteile, welche zuvor nicht erwähnt wurden.

<b>Bauteil</b>		<b>Hersteller</b>	<b>Produktbezeichnung</b>
Relais	Last	TE Connectivity	KILOVAC EV200
	Steuerung	Phoenix Contactors	REL-PR1-24DC/1/MB
	Trenner	Cydom	DR10D12
DC-DC Wandler		Deutronic	DVC251-80-24
Isolationsüberwachungssystem		Bender	ISO3640
Leitungsschutzschalter	96V	Schrack	C13
	24V	ABB	S201 - B10, 1 polig
Linearpotentiometer		TWK-Elektronik	POTI RP13/50-KV-KH-LI
Induktiver Sensor		SICK	IM12-04BPS-ZW1
Schlüsselschalter		RAFI	RAFIX 22QR 1.30.245.202
NOT-AUS Taster		RAFI	RAFIX 22FS+ 1.30.273.301
ASR-Knopf		Schurter	Taster 1241.6634.1124
Kippschalter		Arcoelectric	KS C3900
Endschalter Bremse		Cherry	GPTCRR01
Stecker		TE Connectivity	ampsuperseal, 4 & 6 pins

Tabelle 4.9: Verwendete Bauteile

### 4.4.1 Relais

Insgesamt sind zwei mechanische Relais und ein Halbleiterrelais verbaut. Das Hauptrelais sitzt zwischen Akkus und Motorcontroller. Es schaltet die Energiezufuhr zum Antrieb. Hier fließen Ströme von mehreren hundert Ampere, deshalb muss das Relais groß genug dimensioniert werden. Auch im Kurzschlussfall muss das Relais kurzzeitig den maximalen Kurzschlussstrom aushalten. Verwendet wurde das Relais KILOVAC EV200 vom Hersteller TycoElectronics. Es ist laut Hersteller das kleinste und leichteste Relais in seiner Produktklasse. Das Ausschaltvermögen beträgt bis zu 2000 Ampere bei 320 Volt Gleichspannung. Da die Ansteuerung des Hauptrelais über die SPS Probleme mit dem Motorcontroller bereitete, wurde das Relais direkt an den Controller angeschlossen. Aus sicherheitstechnischen Gründen muss aber das Hauptrelais zusätzlich von der SPS abschaltbar sein. Deshalb wird in den Spulenkreis des Hauptrelais ein Halbleiterrelais geschaltet. Dieses Halbleiterrelais ist im normalen Betrieb immer eingeschaltet. Maximal können hier 24 V DV

anliegen, als Maximalstrom kann von ca. 4 Ampere ausgegangen werden. Wichtig für die Bauteilauswahl war die Kompaktheit sowie eine einfache Montage auf einer Hutschiene. Deshalb wurde sich für das Relais DR10D12 des Herstellers Crytek entschieden. Um die Steuerung des Motorcontrollers ein- und auszuschalten, wird ebenfalls ein Relais benötigt. Hier kann mit Strömen von unter 10 Ampere gerechnet werden, wodurch das Bauteil kleiner dimensioniert werden kann als das Hauptrelais. Zu beachten ist hier jedoch, dass 96 Volt Gleichspannung geschaltet werden müssen, was bei Relais mit Nullpunktlöscher zur Problematik wird. Aus diesem Grund wird das DC – Leistungsrelais Rif 4 von Phoenix Contactors verwendet.

#### **4.4.2 Isolationsüberwachungssystem**

Als Zusatzschutz wurde ein Isolationsüberwachungssystem eingebaut. Dieses Bauteil misst permanent den Widerstand zwischen den Batteriepolen und dem Kartrahmen. Sobald dieser Widerstand geringer als ein eingestellter Wert ist (meist einige 100 kOhm bis MOhm), wird ein Fehler ausgegeben und das Go-Kart gestoppt.

Da solch ein System bereits beim vorherigen Kart der Abteilung für Elektrotechnik angedacht und daher gekauft wurde, jedoch nicht eingebaut worden ist, konnte für dieses Kart das Bauteil vom letzten Jahr verwendet werden.

#### **4.4.3 DC-DC Wandler**

Da die Batteriespannung 96 Volt beträgt, die SPS jedoch nur 24 Volt benötigt, wird ein DC-DC Wandler verwendet, um die Spannung herunterzusetzen. Der in unserem Fall verwendete Wandler trägt die Bezeichnung DVC251-80-24 und wurde vom Hersteller Deutronic gebaut. Zu beachten war dabei, dass der Wandler für genügend Leistung ausgelegt ist, da er sonst an seine thermischen Grenzen stoßen würde. Bei diesem Projekt wurde der Wandler hinsichtlich Leistung überdimensioniert, um auch für etwaige Erweiterungen noch genügend Leistung zu bieten.

#### **4.4.4 Induktive Näherungssensoren - Drehzahlmesser**

Um die Drehzahlen der Vorderräder zu erfassen, werden beidseitig induktive Näherungssensoren an den Vorderrädern angebracht. Diese Sensoren zeigen an, ob sich unmittelbar vor ihnen ein Metall befindet. Zum Erkennen benutzen sie ein Magnetfeld, ähnlich einem Metalldetektor. Der Ausgang ist digital, die Versorgungsspannung wird über die SPS bereitgestellt. Verwendet werden die Sensoren IM12-04BPS-ZW1 der Marke Sick. Der Schaltabstand dieses Typs beträgt 4 - 8 mm, befestigt wird er über ein auf dem Sensor angebrachtes M12 - Gewinde. Die Sensoren zeigen direkt auf die Bremsscheibe. Diese Bremsscheibe hat drei Löcher gleich über ihren

mittleren Umfang verteilt, welche sich pro Umdrehung des Vorderrads am Sensor vorbei bewegen. Der Sensor realisiert das und übermittelt diese Information über eine Signaländerung an die SPS. Dort wird das Signal über einen Counter eingelesen, welcher später vom ASR ausgewertet wird.

#### 4.4.5 Leitungsschutzschalter

Um im Fehlerfall die sensible Elektronik der Steuerung zu schützen sowie um Brände vorzubeugen, wird jedes Potential für die Steuerung separat mittels Leitungsschutzschalter gesichert. Für 96 Volt DC wird ein Schutzschalter der Marke Schrack mit der Auslösecharakteristik C 13 verbaut. Für 24 Volt DC wird einer der Marke ABB mit einer Auslösecharakteristik B 10 verwendet. Die Auslösecharakteristik bzw. der Auslösestrom geben an, ab welchem Kurzschlussstrom bzw. ab welchem Dauerstrom der Stromkreis abgeschaltet wird. Dabei sollte der Nennstrom größer sein als der tatsächliche Strom.

#### 4.4.6 Gaspedal

Für die Ermittlung der Gaspedalstellung wären mehrere Varianten der Realisierung möglich. Zur Verfügung stehen eine Linear- sowie eine Rotationsbewegung. Es wurde sich für die Auswertung der Linearbewegung entschieden, da es sich weitaus einfacher gestaltet.

Dazu wird ein Linearpotentiometer verwendet, welches am Gaspedal und am Rahmen fixiert wird. Je nach Pedalstellung hat das Potentiometer einen gewissen Widerstand. Über ein analoges Eingangsmodul in der SPS wird das Gaspedal eingelesen. Zu beachten sei hier, dass der Widerstand des Potentiometers trotz Ausgangslage des Gaspedals nie null wird. Deshalb muss, um eine konstante Abweichung zu vermeiden, ein Offset im Programm der Steuerung implementiert werden. Als Potentiometer verwendet wird das Bauteil POTI RP13/50-KV-KH-LI vom Hersteller TWK-Elektronik.

#### 4.4.7 Endschalter für die Bremse

Um gleichzeitiges Betätigen von Gas und Bremse zu verhindern, muss erkannt werden, wann gebremst wird. Deshalb wird hinter dem Bremshebel ein Endschalter verbaut, welcher erkennt, ob die Bremse betätigt ist. Unbedingt muss darauf geachtet werden, dass im Falle eines Kabelbruchs das gleiche Signal wie bei einer Bremsung eingelesen wird. Auch ein Umschalten der Fahrtrichtung kann nur bei Betätigen der Bremse durchgeführt werden. Als Schalter wird der GPTCRR01 von Cherry eingebaut.

#### **4.4.8 Schlüsselschalter**

Um nur autorisierten Personen die Verwendung des Go-Karts zu gestatten, sind unter dem Lenkrad zwei Schlüsselschalter angebracht. Dabei handelt es sich beim ersten Schlüssel um den Hauptschlüssel, mit dem das Kart im gedrosselten Modus aktiviert werden kann. Über den zweiten Schlüssel wird der Boost-Modus aktiviert, das Kart ist also ungedrosselt fahrbar. Sollte ein Schlüssel während des Fahrens herausgezogen werden, schaltet der Motor automatisch ab. Als Schlüsselschalter werden die RAFIX 22QR 1.30.245.202 der Marke Rafi verwendet.

#### **4.4.9 NOT-AUS Taster**

Um im Fehlerfall ein sofortiges Abschalten des Motors zu ermöglichen, muss ein NOT-Aus –Schalter verbaut werden. In diesem Fall handelt es sich um einen Pilztaster, der durch Drehen entriegelt wird. Eingelesen wird der Taster zwei-kanalig über ein Safety-Modul. Dadurch sind eine Querschlusserkennung sowie eine Feststellung eines Kabelbruchs bei einem Kanal möglich. Der Taster besitzt die Stopp-Kategorie 0, das bedeutet, dass bei Betätigen sofort der Energiefluss des Antriebssystems unterbrochen wird.<sup>8</sup> Dieser Taster ist ebenfalls von Rafi und trägt die Bezeichnung RAFIX 22FS+ 1.30.273.301.

#### **4.4.10 Einschaltknopf für die ASR**

Im Lenkrad wird ein Druckknopf einbaut, über den sich die Antriebsschlupfregelung bei Bedarf ein- bzw. ausschalten lässt. Das Besondere an diesem Knopf ist ein LED-Ring, der den Taster umgibt. Über diesen Ring wird der aktuelle Einschaltzustand vom ASR angezeigt. Dieser Knopf trägt die Bezeichnung Taster 1241.6634.1124 und wurde von der Firma Schurter hergestellt.

#### **4.4.11 Kippschalter**

Kippschalter werden verwendet, um die SPS und den Motorcontroller separat einzuschalten. Dazu werden zwei in der Armatur unter dem Lenkrad verbaut. Bei der Auswahl des korrekten Bauteils musste besonders auf Schaltspannung und Schaltstrom geachtet werden, da diese bei Gleichspannung oft sehr gering sind. Deshalb finden hier die Kippschalter KS-C3900 von der Marke Arcoelectric Verwendung. Der Nennstrom beträgt 2 Ampere bei einem Ausschaltvermögen von 30 VDC.

---

<sup>8</sup>vgl: Notausschalter, <https://goo.gl/Crf5TP> (Stand: 18.03.2017)



#### 4.4.12 Steckverbinder

Stecker werden bei allen Komponenten benötigt, die nicht fix am Kart montiert sind oder aber deren eingebautes Anschlusskabel nicht lange genug ist. In unserem Fall sind das das Lenkrad, der Motor, der DC-DC Wandler, der Motorcontroller sowie der CAN-Adapter. Abgesehen vom Motorcontroller werden Stecker der Serie ampsuper-seal der Marke TE Connectivity verbaut. Dabei werden Varianten mit verschiedensten Pin-Anzahlen verwendet. Der besondere Vorteil der Stecker liegt in ihrer Robustheit und Wasserfestigkeit, was sie besonders als Automobilstecker auszeichnet. Der Motorcontroller besitzt bereits eine eigene Buchse für alle Anschlüsse der Logik. Diese stammt ebenfalls vom Hersteller TE Connectivity, allerdings aus der Baureihe „ampseal“. Der Stecker hat 35 Pins.

#### 4.4.13 Kabel und Kabelschuhe

Um die großen Ströme problemlos führen zu können, muss der Kabelquerschnitt dementsprechend groß dimensioniert werden. Da in diesem Fall kurzzeitig Ströme von bis zu 500 Ampere auftreten können, wurde ein Querschnitt von 95 mm<sup>2</sup> gewählt. Das Biegen von solchen Kabeln gestaltet sich bereits sehr schwierig, ebenso die Verbindung zu anderen Komponenten. Deshalb wurden auf die Kabelenden Kerbkabelschuhe mit Ring aufgepresst, wodurch sich eine problemlose Montage mittels Schrauben ergab.

#### 4.4.14 Sonstige Komponenten

Wie bei jeder elektrischen Anlage üblich, werden diverse Kleinteile benötigt. Besonders wichtig sind Kabelbinder, Aderendhülse, Kabel, Kabelschuhe, etc. Auch Hutschienen und Klemmen werden verbaut. Eine detaillierte Ausführung wäre jedoch zu umfangreich und nicht sinnvoll, daher soll darauf verzichtet werden.

## 4.5 Lageplan

Diese Abbildung soll die Lage der einzelnen Komponenten anschaulich darstellen. Auf Kabel wurde allerdings verzichtet.

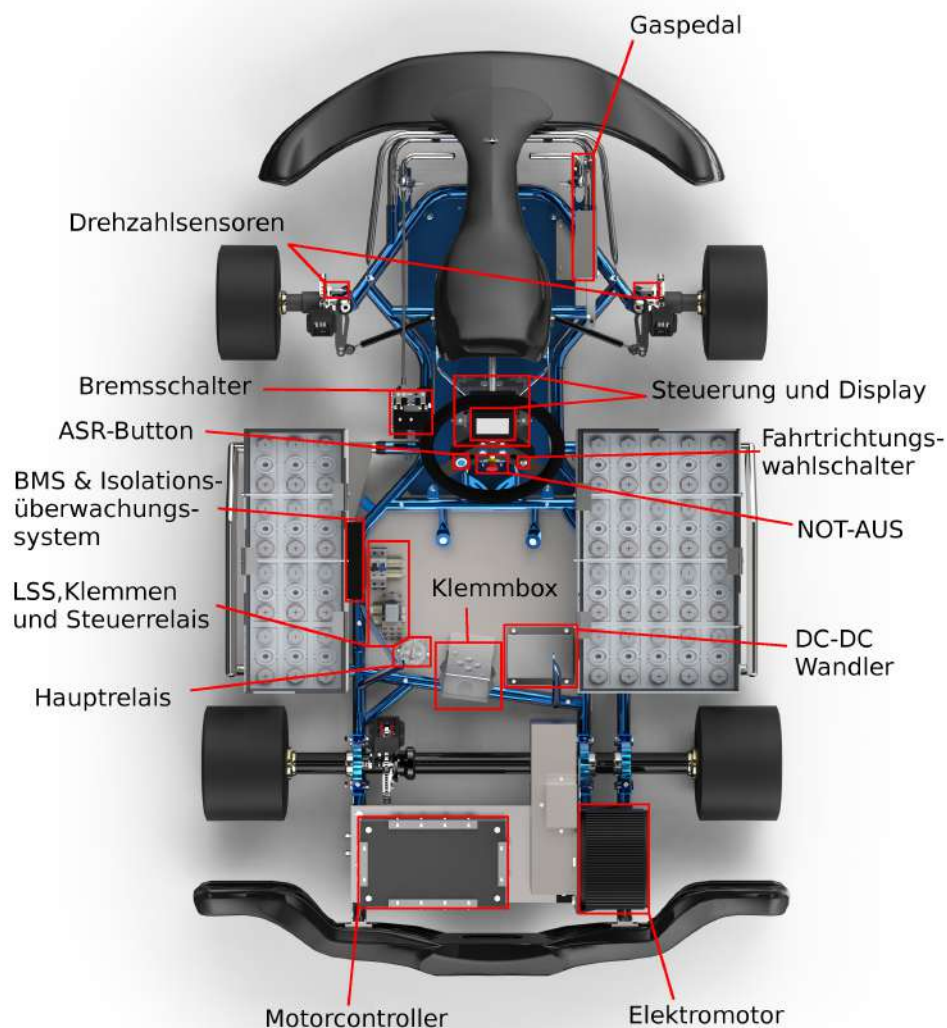


Abbildung 4.2: Lageplan

## 4.6 Energieversorgung

### 4.6.1 Einführung

Wie bei den immer mehr aufkommenden Elektro-Autos soll das Kart mit einer mobilen Spannungsversorgung betrieben werden. Die  $\text{LiFePO}_4$ -Akkumulatoren stellen mit gutem Wirkungsgrad, hoher Energiedichte und großer Eigensicherheit die ideale Spannungsversorgung dar. Sie haben zwar gegenüber  $\text{LiCoO}_2$  eine geringere Energiedichte, neigen jedoch nicht durch mechanische Beschädigungen zum thermischen Durchgehen. Als thermisches Durchgehen wird die selbstständige Entzündung durch Freisetzung von Sauerstoff bezeichnet.

Die größten Nachteile für das Kart sind jedoch der große Platzbedarf und das enorme Gewicht von 110kg. Um die Zellen vor Überladung und Tiefenentladung zu schützen ist eine extra Beschaltung notwendig. Diese Schutzbeschaltung erfolgt durch ein Batteriemanagementsystem, welches im 4.6.4 beschrieben wird.

### 4.6.2 Aufbau

Bei diesem Akkumulator wird an der negativen Elektrode (Anode) Lithium-Eisenphosphat ( $\text{LiFePO}_4$ ), statt wie bei herkömmlichen Lithium-Ionen-Akkus Lithium-Cobalt(III)-oxid ( $\text{LiCoO}_2$ ) verwendet, die Kathode besteht aus Grafit. Um einen Kurzschluss zu verhindern, werden die Elektroden durch einen Separator getrennt, dieser ist aber für die Lithium-Ionen durchlässig.<sup>9</sup>

### 4.6.3 Dimensionierung

Der verwendete Motorcontroller mit Motor benötigt eine Spannung von 96V DC. Dies wird durch die Serienschaltung von 32 Zellen mit je 3,2V erreicht. Um eine angemessene Fahrzeit zu erhalten, werden Zellen mit einer Kapazität von 100Ah eingebaut. Somit errechnet sich eine Dauerbetriebszeit von 30 min. Da im normalen Fahrbetrieb nur die Hälfte der Zeit Vollgas gegeben werden kann, kann die tatsächliche Fahrzeit mit etwa einer Stunde angenommen werden.

---

<sup>9</sup>vgl. Lithium-Eisenphosphat-Akkumulator, <https://goo.gl/W8uJhu>, (Stand: 23.03.2017)

100	Ah	Nennkapazität
3,2	V	Nennspannung
1	C	entspricht 100 A
20	C	max. Entladerate
2	C	Dauer-Entladerate
1	C	Ladestrom
3,6	V	Ladeschlussspannung
142	mm	Länge
67	mm	Breite
219	mm	Höhe
3400	gr.	Gewicht
0,48	Euro	Preis/Wh
320,00	Wh	Nennenergieinhalt
94,12	Wh/kg	Leistungsgewicht
6400	W	max. Spitzenleistung
		Laden mit CC/CV

Tabelle 4.10: Technische Daten der Zellen

#### 4.6.4 Batteriemanagementsystem

Das Batteriemanagementsystem (BMS) dient als Schutz für die Zellen gegen Über- und Tiefenentladung. Es misst die einzelnen Spannungen und Temperaturen der Akkumulatoren. Über die Ausgänge können z.B. Fehler, Übertemperatur sowie der Ladezustand ausgegeben werden. Über den PC kann durch das Control-Panel jederzeit der aktuelle Status angezeigt werden. Dieser beinhaltet ein Balkendiagramm der Spannungs- und Temperaturwerte jeder Zelle. Zusätzlich wird angezeigt, welche Ausgänge ein bzw. ausgeschaltet sind. Bei Spannungsunterschieden im Ladevorgang kann das BMS ausgleichen, sprich „balancen“. Dabei wird Energie von Zellen mit höherer Ladung in Zellen mit niedrigerer Ladung umgeladen<sup>10</sup>.

Im folgenden Abschnitt werden die einzelnen Komponenten des Batteriemanagementsystems der Firma Elektromotus beschrieben.

##### 4.6.4.1 EMUS BMS Control Unit

Dieses Bauteil stellt das Herzstück des Batteriemanagementsystems dar. Es verarbeitet die Daten der einzelnen Zellmodule und kann bei Erreichen unerwünschter Werte Warnungen ausgeben und damit über die SPS die Spannungsversorgung ausschalten. So sollen die einzelnen LiFePO<sub>4</sub> Zellen vor Über- und Tiefenentladung geschützt werden. Die sicherheitsrelevanten Werte werden mit dem Control Panel

<sup>10</sup>vgl. Balancer, <https://goo.gl/qSoMtF>, (Stand: 23.03.2017)



Abbildung 4.3: Controllunit

(einer Software für den PC) konfiguriert und in der Control Unit gespeichert. Die Spannungsversorgung mit +24 V erfolgt durch einen den Akkus nachgeschalteten DC/DC Wandler, der die Spannung von 96 V auf die nötige Versorgungsspannung reduziert.

#### 4.6.4.2 EMUS BMS Cell Module A/B



Abbildung 4.4: Zellmodule A und B

An jeder Batteriezelle wird am negativen Pol ein Zellmodul angebracht. Der rote Draht wird mit Hilfe des Ringkabelschuhs mit dem Pluspol der gleichen Zelle verbunden. Das Zellmodul misst die Spannung und die Temperatur der jeweiligen Zelle des Akkus. Bei dieser Version des Zellmoduls gibt es A- und B- Module. Diese sind gleich aufgebaut, nur die Anschlüsse der Verbindungssignale (UP und DN) sind vertauscht. Dies erlaubt eine einfachere Verkabelung. Die Anschlüsse der Datenleitung können einfach in Serie verbunden werden (siehe Abbildung 4.5). Die Informationen werden über die Datenleitung, welche durch ein Cell Group Module abgeschlossen ist, der Control Unit übergeben.

#### 4.6.4.3 EMUS BMS Cell Group Module

Mit dem Cell Group Module kann der Batterieblock in einzelne Zellgruppen aufgeteilt werden. In jeder Zellgruppe befindet sich dann ein Cell Group Module, welches mit der Control Unit über den CAN-Bus kommuniziert. Das Modul wird so angeschlossen, dass die Enden der Datenleitung (UP, DN) der Zellmodule mit dem Cell Group Module verbunden sind. Der negative Pol der Bottom Cell sowie der negative und positive Pol der Top Cell werden ebenfalls mit dem Modul verbunden (siehe Abbildung 4.5).

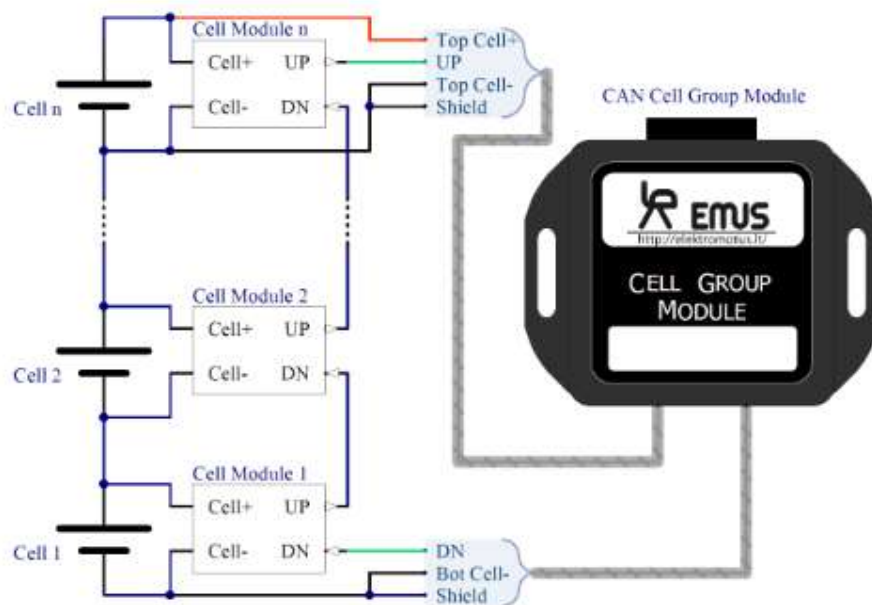


Abbildung 4.5: Datenleitung mit CAN Cell Group Module

Da bei dem Kart durch die Aufteilung in einen rechten und linken Akkublock zwei Cell-Group Module verwendet werden, muss beim Anschließen darauf geachtet werden, dass das Modul mit der kleinsten Seriennummer beim negativsten Potential und das Modul mit der größeren Seriennummer beim positivsten Potential angeschlossen wird.

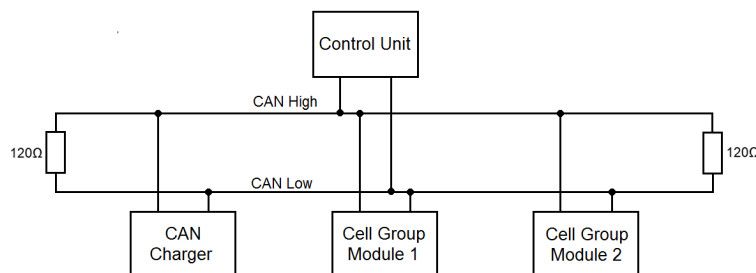


Abbildung 4.6: CAN Bus

#### 4.6.4.4 EMUS BMS Dual Range Current Sensor

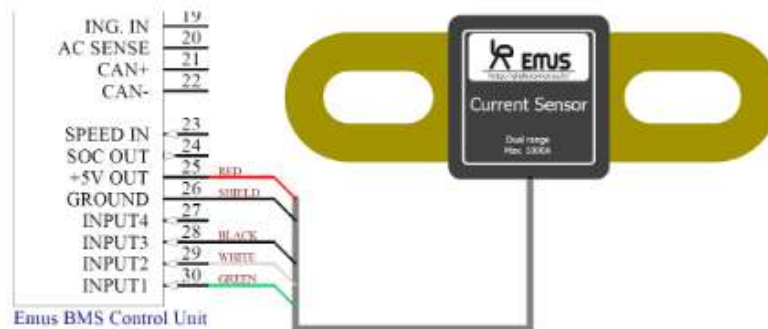


Abbildung 4.7: Stromsensor mit Anschluss

Der Stromsensor besitzt zwei Hallsensoren. Einen zum Messen von 0-100 Ampere und einen für den Messbereich von 100-1000 Ampere. Deshalb wird er auch Dual Range (Zwei-Bereich) Stromsensor genannt. Um bei niedrigen Lasten eine größere Genauigkeit zu erhalten, wird der kleinere Messbereich verwendet. Für die Messung der größtmöglichen Ströme muss auf den zweiten Bereich gewechselt werden. Der Stromsensor kann im Akkublock überall in Serie eingebaut werden. Über die vier Eingänge bei der Control Unit (Input1- Input4) werden die Werte eingelesen und verarbeitet. Jedoch kann es durch Einstreuungen anderer Magnetfelder zu Messfehlern kommen. Um die richtigen Werte für den Strom zu erhalten, soll der Sensor nach dem Einbau kalibriert werden.

„Wird ein [...] Hall-Sensor von einem Strom durchflossen und in ein senkrecht dazu verlaufendes Magnetfeld gebracht, liefert er eine Ausgangsspannung, die proportional zum Produkt aus magnetischer Flussdichte und Strom ist.“<sup>11</sup> Der zu messende Strom erzeugt üblicherweise das Magnetfeld  $B$ , der durch die Sonde fließende Strom ist konstant.

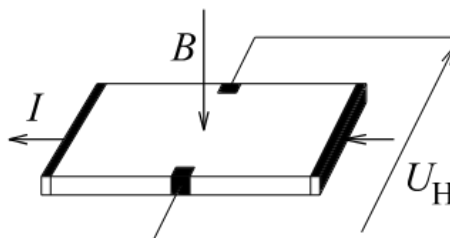


Abbildung 4.8: Aufbau eines Hall Sensors

<sup>11</sup> Pfad: Hallsensor, <https://goo.gl/0khVvR>, (Stand: 23.03.2017)



#### 4.6.4.5 Konfiguration des EMUS BMS

Die Konfiguration des Batterie-Management-Systems erfolgt mittels PC über die Emus Control Panel Software. Um Einstellungen durchführen zu können, wird die Control Unit über USB an den PC angeschlossen. Nach dem Anlegen der Versorgungsspannung und Öffnen der Software wird das BMS automatisch verbunden. Das Programm teilt sich in vier Hauptpunkte, die im nächsten Abschnitt genauer beschrieben sind.

##### 4.6.4.5.1 Status

Im Menü Status können im Reiter „Allgemein“ die aktuellen Daten und Zustände des BMS ausgelesen werden. Im zweiten Reiter „Zellen“ befinden sich 3 Diagramme, welche über die Spannungen, Temperaturen und die Ausgleichsstromstärken der einzelnen Zellen informieren. Hier können ebenfalls die Maximal-, Minimal- und Durchschnittswerte abgelesen werden.

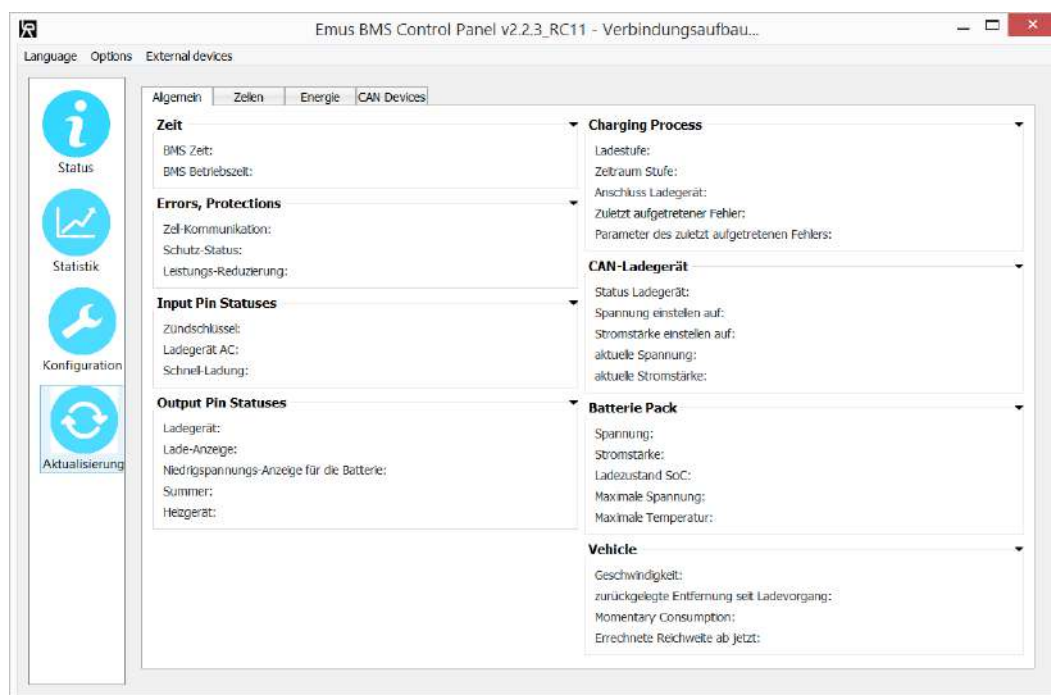


Abbildung 4.9: Status Allgemein

##### 4.6.4.5.2 Statistik

Im Abschnitt Statistik kann eine Ereignishistorie abgerufen werden, welche das Starten des BMS oder auftretende Fehler aufzeichnet.



#### 4.6.4.5.3 Konfiguration

Im Abschnitt Konfiguration werden die einzelnen Werte für einen sicheren Betrieb eingestellt.

Der Reiter „Management“ ermöglicht die Einrichtung eines Passwortes, um nur gewünschten Personen Änderungen zu erlauben. Hier können ebenfalls Standardwerte geladen oder bereits eingestellte Werte gesichert werden.

Im Reiter „Allgemein“ können das Datum und die aktuelle Uhrzeit gesetzt werden. Die Parameter bezüglich des CAN-Busses und die Werte zur Aktualisierung des Statusfensters werden hier festgelegt.

Unter „Cell“ werden alle Spannungs- und Stromwerte eingestellt, um beim Laden und Entladen die Zellen optimal zu schützen.

Im Reiter „Batterieblock“ können die Werte zum Schutz der Zellen eingestellt werden. Zum Beispiel kann beim Erreichen hoher Temperaturen ein Ventilator eingeschaltet werden.

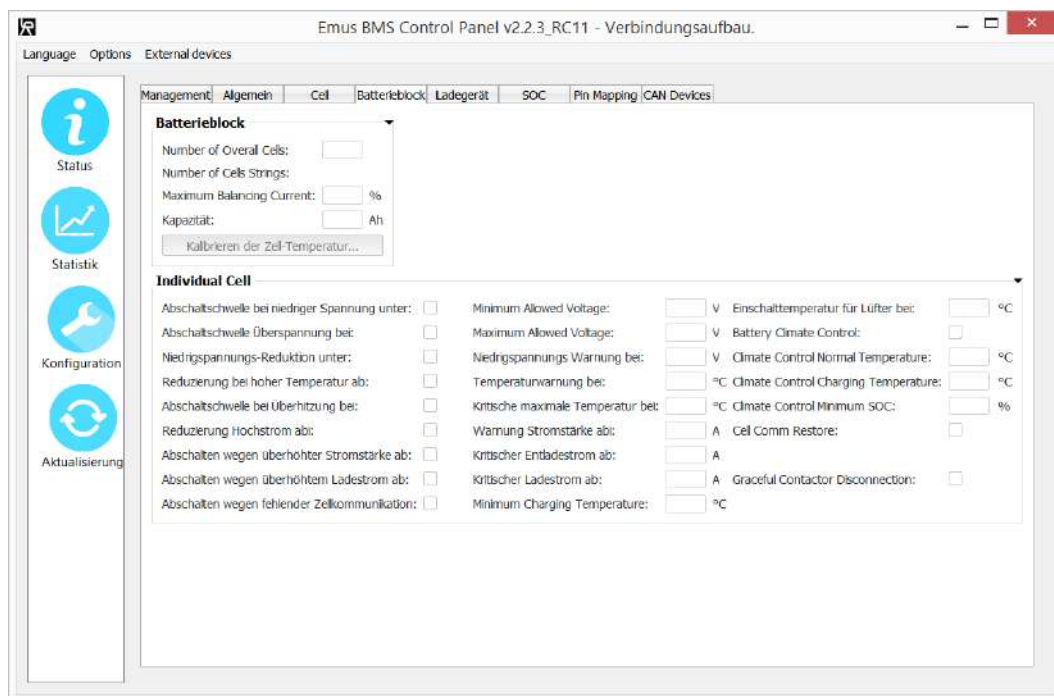


Abbildung 4.10: Konfiguration Batterieblock

Unter „Ladegerät“ werden die Ladestufen und das verwendete Gerät zum Laden festgelegt.

„SOC“ bietet die Möglichkeit, den Stromsensor zu kalibrieren.

Im Reiter „Pin Mapping“ werden den verschiedenen Ein- und Ausgängen deren Funktion zugeteilt.

„CAN Devices“ zeigt alle Geräte, die über den CAN-Bus kommunizieren. Hier werden die zwei verwendeten CAN Cell Group Module und die Anzahl der Zellen angezeigt.

#### **4.6.4.5.4 Aktualisierung**

Hier kann ein Update mit neuerer Firmware durchgeführt werden.

#### **4.6.4.6 Kommunikation mit der SPS**

Die Control Unit kann einige wichtige Daten über den Zustand des Akkumulators für den Fahrer ausgeben. Um die Daten wie Ladezustand und Warnungen zu veranschaulichen, kommuniziert das BMS über seine Ausgänge mit der SPS. Die Daten werden anschließend mit der Steuerung ausgewertet und dem Lenker des Fahrzeuges über das Display am Lenkrad angezeigt.

Folgende Aus- und Eingänge werden verwendet:

##### **SOC OUT**

Dieser Ausgang gibt den aktuellen Ladezustand aus. Mit einem Kondensator  $>1\mu\text{F}$  zwischen Ausgang und Masse wird das pulswidenmodulierte Ausgangssignal in eine analoge Spannung umgewandelt. Diese wird der SPS zugeführt und im Display als Prozentangabe angezeigt.

##### **IGN.IN**

Dies ist ein Eingang für den Zündschlüssel des Fahrzeuges. Wird an diesen Eingang eine Spannung von  $>5\text{V}$  angelegt, wechselt das Batterie Management System vom Ruhezustand in den aktiven Zustand. Dadurch werden die Zellen-, Umfrage- und Anzeigekommunikationsraten gegenüber dem Ruhezustand erhöht. Unabhängig von der Position des Zündschlüssels wird der Aktiv-Zustand auch berücksichtigt, wenn ein CAN- Ladegerät angeschlossen ist.

##### **HEATER**

Für diesen Ausgang wird die Funktion „Battery Fan Output“ verwendet. Dies bedeutet, dass bei einem Überschreiten einer voreingestellten Maximaltemperatur der Ausgang eingeschaltet wird. Somit würde bei üblicher Anwendung ein Kühlgerät, zum Beispiel ein Ventilator, einschalten. Bei diesem Fahrzeug wird dieser Ausgang jedoch an kein Kühlgerät, sondern an einen digitalen Eingang der SPS angeschlossen. Wird der Ausgang geschaltet, wird über das Touch-Panel am Lenkrad eine Warnung ausgegeben, worauf der Fahrbetrieb eingestellt wird.

##### **CHARGER**

Dieser Ausgang wird mit der Funktion „Battery Contactor Output“ verbunden, einer für den Schutz der Zellen relevanten Funktion. Wenn eine Grenzwerte erreicht wird, schaltet der Ausgang auf High. Die Steuerung wertet diesen digitalen Wert aus und gibt über das Display eine Warnung aus.

#### 4.6.4.7 Ladevorgang

Beim Laden des Fahrzeuges wird ein sogenannter „CAN-Charger“ verwendet. Das Ladegerät *1.8KW HK-H Series* kommuniziert während des Ladevorganges über den CAN-Bus mit der Control Unit. Für einen zuverlässigen Betrieb ist ein beidseitiger Abschluss des CAN-Busses mittels 120 Ohm Widerständen nötig. Der Ladevorgang kann mit der EMUS BMS Control Panel Applikation am Laptop oder Computer konfiguriert und überwacht werden. Der Anschluss des Ladegerätes an den CAN-Bus ist in der folgenden Abbildung dargestellt.

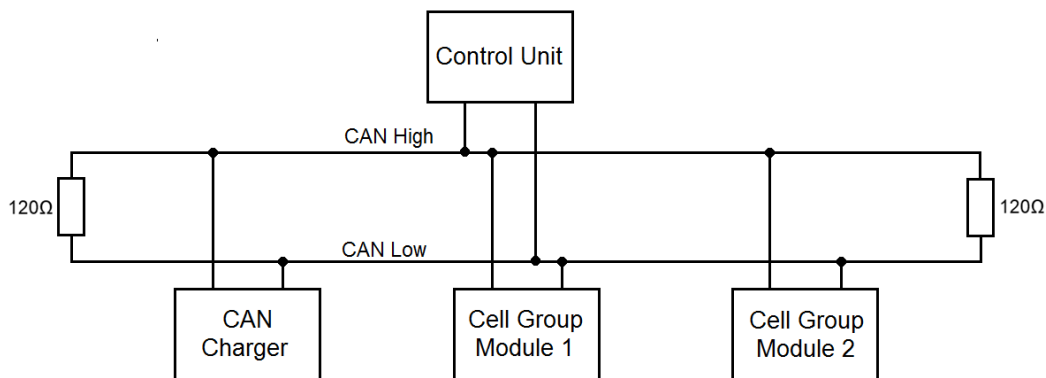


Abbildung 4.11: Anschluss Charger

#### 4.6.4.8 Überblick Gesamtkonzept

Im Anschluss befindet sich eine Übersichtsgrafik, auf der das Gesamtkonzept des Batteriemanagementsystems abgebildet ist.

Abschließend ist noch zu erwähnen, dass die digitalen Ausgänge des Batteriemanagementsystem vom Hersteller mit MOSFET-Endstufen realisiert sind. Deshalb wurde ein Pull-Up Widerstand zur Übertragung der logischen HIGH-Spannung von 24 Volt benötigt. Um die Stromaufnahme dieser Endstufe gering zu halten, wurde ein 8,2 kOhm Widerstand verwendet, da sonst die Verlustleistungen unnötig hoch ausgefallen wären.

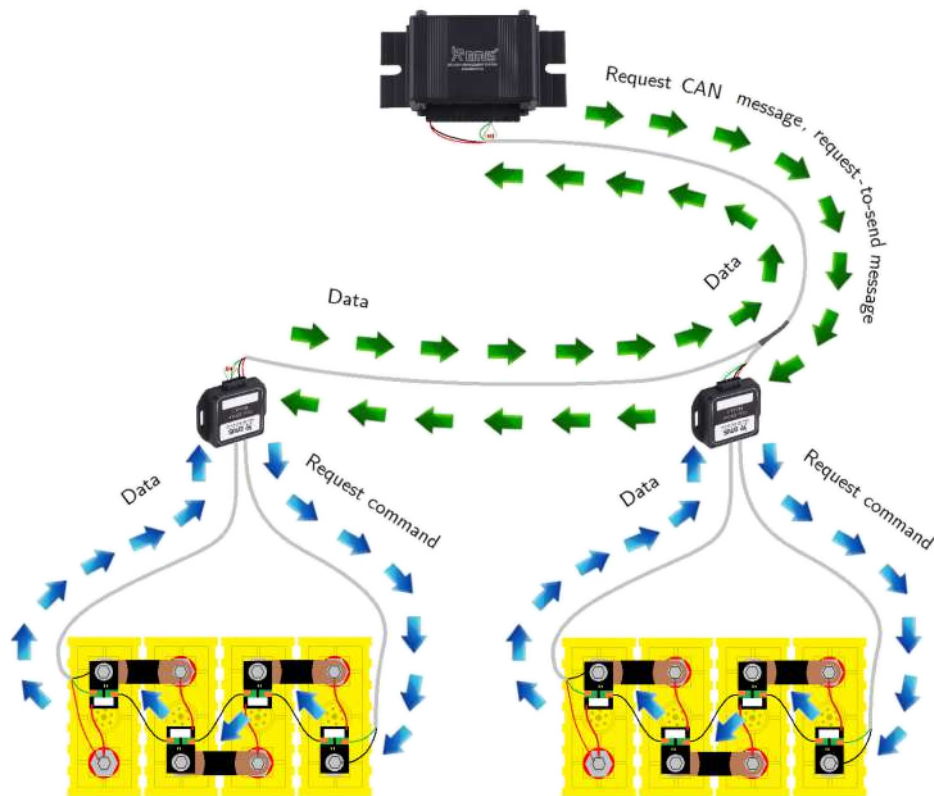


Abbildung 4.12: Gesamtkonzept

## 4.7 Schaltpläne

### 4.7.1 Steuerteil

Dieser Schaltplan beinhaltet die gesamte Steuerung des Go-Karts. Hauptteil bildet hier die SPS mit ihren einzelnen Modulen. Zusätzlich sind alle Bedienelemente, das Display sowie das BMS und das Isolationsüberwachungssystem zu finden. Auch die Schnittstellen zum Controller und zum Motor sind in Form von den passenden Steckern eingezeichnet. Als Stecker wird für den Controller der 35-polige Stecker der Serie AMP - Seal von TE-Connectivity verwendet. Für alle anderen Komponenten, wie z.B. Motor oder DC-DC Wandler, werden wasserdichte 4-polige Stecker der Serie „ampsuperseal“ herangezogen. Solche Stecker sind im Automobilbereich typisch, da sie mechanisch robust und vollständig wasserdicht sind. Für die Kabel wird hauptsächlich der Querschnitt  $1\text{mm}^2$  verwendet. Lediglich die Kabel für die Versorgungsleitungen der KL-Module werden mit mindestens  $3\text{mm}^2$  ausgeführt, um bei größeren Strömen keinen Spannungsabfall zu verursachen. In der Anwendung werden die meisten Aktoren mit jeweils eigenem Kabel angeschlossen. Dabei werden die Standardfarben Braun – Grau – Schwarz – Gelb/Grün verwendet. Somit weichen die realen Farben von den verwendeten Farben im Schaltplan ab. Zur Signalanpassung ist zwischen SPS und Motorcontroller ein Impedanzwandler geschaltet, welcher im Schaltplan lediglich als Rechteck dargestellt ist. Im folgenden Abschnitt soll die Funktionsweise dieses Impedanzwandlers genauer erläutert werden.

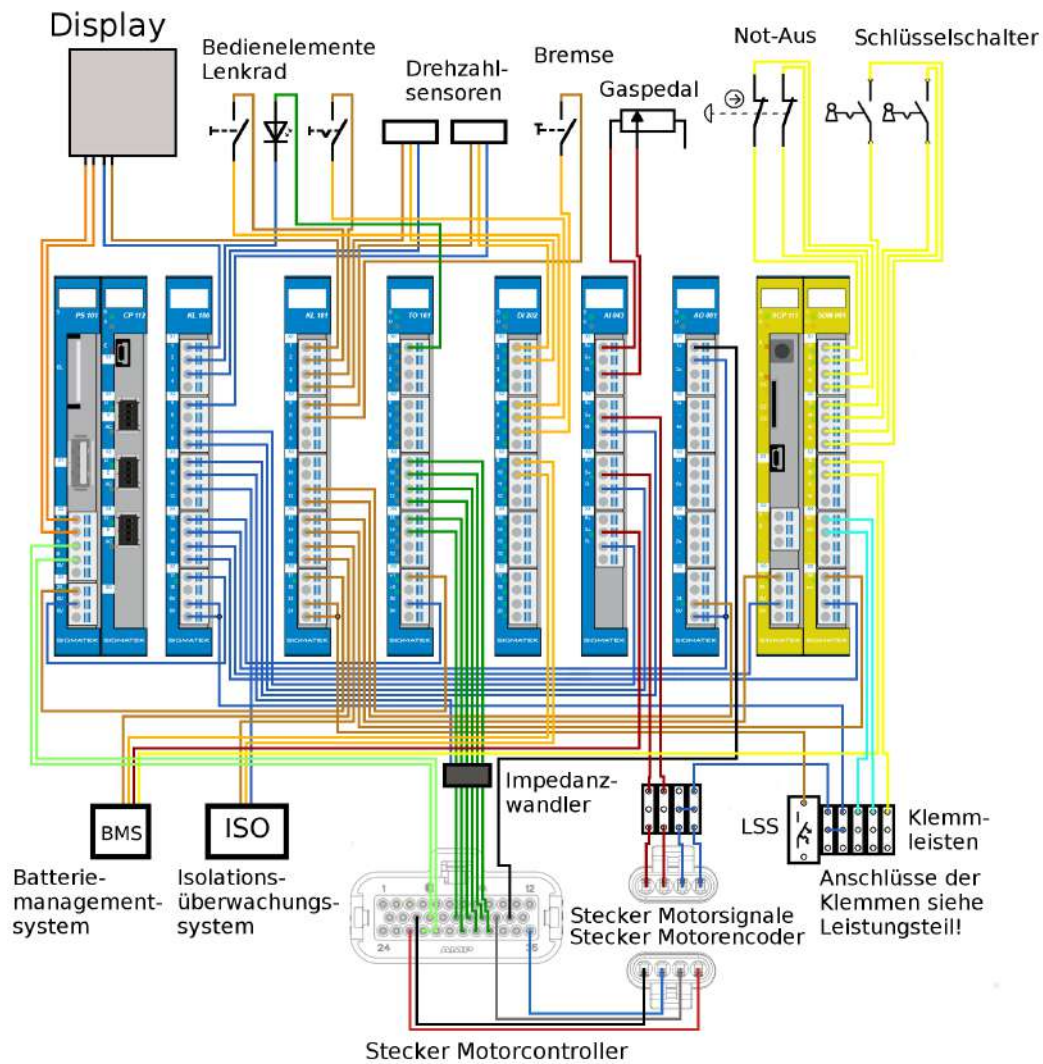


Abbildung 4.13: Schaltplan des Steuerteils

## 4.7.2 Impedanzwandlung

Bei der Kommunikation zwischen der SPS und dem Motorcontroller sind einige Probleme aufgrund von falschen Potentialen aufgetreten. Das digitale Ausgangsmodul TO161 besitzt galvanisch getrennte Ausgänge. Daher schaltet das Modul bei logisch Low zwar auf Null Volt, aber nicht auf GND-Potential durch, wodurch der Controller an seinen Eingängen dieses Low-Signal nicht erkennt. Auch können bei nicht gesetzten Ausgängen Potentialrückwirkungen vom Controller auftreten, welche eventuell das Ausgangsmodul beschädigen könnten. Deshalb muss zwischen SPS und Motorcontroller ein Impedanzwandler geschaltet werden, welcher die Potentiale trennt und auf den richtigen Pegel bringt. In diesem Fall wird dieser Impedanzwandler über einen Optokoppler realisiert. Ein Optokoppler ist ein passives elektronisches Bauteil, welches zwei Signale galvanisch getrennt miteinander verbindet. Dabei wird die Kopplung über eine Infrarotdiode und einen Fototransistor realisiert.

Der Einschaltstatus des Controllers soll direkt über die Spannungsversorgung des Logik-Teils des Controllers erkannt werden. Da hier jedoch 96 Volt verwendet werden, muss ein Vorwiderstand vor den digitalen Eingang der SPS geschaltet werden, um keine Beschädigungen der SPS zu riskieren. Auf einer Schaltung werden beide Signalwandler angebracht. Die Platine dazu ist bewusst sehr kompakt gestaltet, um noch in der Box der SPS Platz zu finden. Über Reihenklemmen können alle Kabel an die Platine angeschlossen werden.

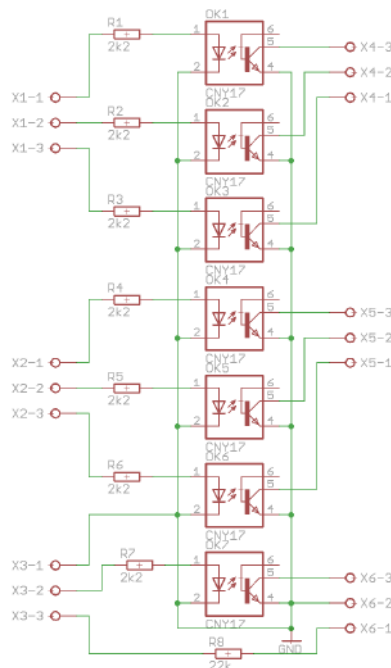


Abbildung 4.14: Platine



### 4.7.3 Lastteil

Dieser Schaltplan beinhaltet den gesamten Leistungsteil des Go-Karts. Hier enthalten sind Motor, Controller, Akkus sowie Relais und Sicherungen. Die Steuerungskabel des Motors und des Controllers werden hier nicht mehr eingezeichnet, da sie bereits im Steuerungsschaltplan enthalten sind. Da für die Spitzenleistung Ströme bis über 500A fließen, müssen die Kabel dementsprechend groß dimensioniert werden. Verwendet wird ein Querschnitt von 95mm<sup>2</sup>. Die Verbindung zu anderen Komponenten wird über Ringkabelschuhe hergestellt. Dabei gestaltet sich die Verlegung bereits problematisch, da sich solch ein Kabel nur noch schwer biegen lässt. Es wird beim Verkabeln sehr darauf geachtet, dass die Farben laut Plan eingehalten werden. Somit haben Kabel mit einem Potential von 96 Volt immer die Farbe Rot, Kabel mit einem Potential von 0 Volt immer die Farbe Schwarz. Die Fahrzeugmasse wird hier bewusst nicht mit dem Potential B- (0 Volt) verbunden, da hier ein Isolationsüberwachungssystem zum Einsatz kommt. Kabel für die Versorgungsspannung der Steuerung werden mit einem Querschnitt von mindestens 3mm<sup>2</sup> ausgeführt, um bei größeren Strömen keinen merklichen Spannungsabfall zu erzeugen. Der Motor ist über drei Kabel mit einem Querschnitt von 50mm<sup>2</sup> am Controller angeschlossen. Hier kann im Vergleich zu den Batteriekabeln unbedenklich ein kleinerer Querschnitt verwendet werden, da sich der Strom gleichmäßig auf die drei Phasen des Motors aufteilt.



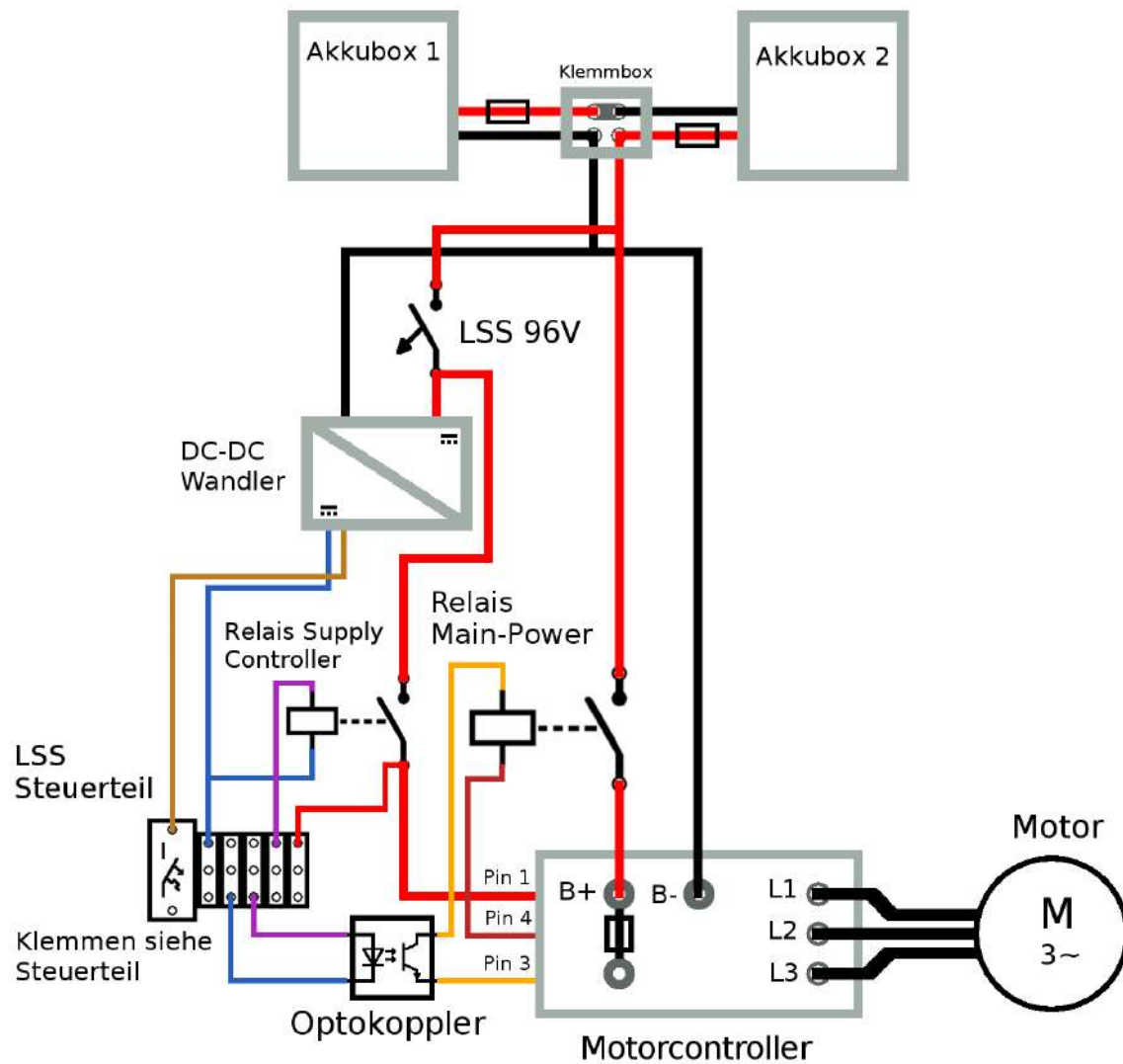


Abbildung 4.15: Schaltplan des Lastteils



# Kapitel 5

## Mechanik

Ein großer Teil des Diplomprojektes war die Adaptierung der Mechanik. Da anfangs nur das Rohchassis zur Verfügung stand, wurden sämtliche folgende Werkstücke angefertigt und montiert. Wo sich diese genau befinden, kann in den Explosionszeichnungen der Konstruktionszeichnungen im Anhang eingesehen werden.

### 5.1 Zusammenbau des Roh-Chassis

Bei dem verwendeten Chassis handelt es sich um ein Produkt der italienischen Firma CKR. Der Rahmen ist für ein Kart ausgelegt, das mit einem Verbrennungsmotor angetrieben wird und bietet deshalb auf der rechten Seite mehr Platz für die Erweiterungen. Das Chassis ist von der CIK-FIA (Commission Internationale de Karting) offiziell als Renn-Chassis zugelassen und wird mit den folgenden mechanischen Teilen erweitert, um alle Komponenten sicher befestigen zu können.

#### 5.1.1 Anbringung der Verkleidungen

Das Chassis wurde größtenteils fertig zusammengebaut geliefert. Jedoch mussten noch einige Teile angeschraubt sowie die Verkleidung befestigt werden. Die Frontschürze wurde zwischen zwei Rohren eingeklemmt, die durch zwei Schellen zusammengepresst werden. Das Frontschild, welches von der Frontschürze bis zum Lenkrad führt, war an drei Stützen mittels Splinten zu befestigen. Die Seitenteile, die die Batteriekästen stützen, wurden in den Rahmen gesteckt und mit jeweils zwei Federn am Herausrutschen gehindert. Um die Federn zu entlasten, wird das Seitenteil durch Verschraubungen gesichert.

#### 5.1.2 Montage des Lenkrades

Das Lenkrad wird mit einem Flansch mit der Lenksäule verbunden. Dieser besitzt ein Bohrloch von 20 mm und wird 50 mm auf die Lenksäule geschoben. Anschließend

wird es mit zwei M6 Schrauben fest verbunden. Um das Lenkrad mit dem Flansch zu verbinden, wird es durch drei um jeweils 120° verschobene M6 Senkkopfschrauben befestigt. Dabei ist auf die richtige Ausrichtung zu achten.

## 5.2 Chassis Verlängerung

Da durch die großen Akkus im Mittelteil der Motor nicht mehr Platz hatte, wurde das Kart hinten um 10 cm verlängert. Diese Verlängerung besteht aus zwei verschiedenen Werkstücken, der Verlängerungsstücke und der hinteren Bodenplatte.

### 5.2.1 Verlängerungsstücke



Abbildung 5.1: Verlängerungsstück

Die Verlängerungsstücke wurden aus einer 30 mm Vollwelle aus Aluminium erstellt. Durch Bearbeiten auf der Drehmaschine wurde der Welle die passende Form gegeben. Dann wurde das Teil hinten in den Kartrahmen gesteckt und mit M8 Schrauben befestigt. Die Heckschürze wurde anschließend an der Verlängerung montiert und durch Zusammenpressen der Gummielemente befestigt.

### 5.2.2 Hintere Bodenplatte

Um das verlängerte Stück nutzen zu können, wurde ein 4 mm dickes Aluminiumblech in der passenden Größe mit M10 Schrauben auf den Verlängerungsstücken befestigt. Links auf diesem Blech befindet sich der Motorcontroller. Zur besseren Wärmeableitung wurde auf der Unterseite eine Wärmeleitpaste aufgetragen. Somit kann diese Bodenplatte als zusätzlicher Kühlkörper genutzt werden und auch auf der Unterseite mit einem zusätzlichen Kühlkörper, falls nötig, erweitert werden. Auf der rechten Seite wurde die Motorhalterung ebenfalls mit M10 Schrauben befestigt. Die Bodenplatte wurde nach der Herstellung sandgestrahlt und mit Klarlack versiegelt.



Abbildung 5.2: Hintere Bodenplatte

## 5.3 Motorhalterung



Abbildung 5.3: Motorhalterung

Um den 20 kW Motor stabil auf dem Kart befestigen zu können, wurde eine eigene Motorhalterung konstruiert. Diese besteht aus einem Grundgestell, einer zusätzlichen Verstärkung und der Spannvorrichtung für den Kettenspanner.

### 5.3.1 Motorhalterung Grundgestell

Das Grundgestell besteht aus zwei 8 mm dicken Stahlteilen, die rechtwinkelig zueinander verschweißt sind. Um beim Schweißen dem Schweißverzug durch die große Hitze zu entgehen, wurden die beiden Teile mit Positionsschlitzen versehen. Der Motor kann anschließend mit 4 Schrauben an der am Kartrahmen befestigten Motorhalterung festgeschraubt werden. Da 8 mm dicker Stahl schwer zu bearbeiten

ist, wurden die beiden Teile der Motorhalterung gelasert und anschließend selbst zusammengeschweißt. Um eine höhere Stabilität gegen ein Verbiegen zu erreichen, wurden seitlich zwei Stützverstrebungen angeschweißt.

### 5.3.2 Spannvorrichtung für den Kettenspanner

Um eine optimale Spannung der Kette zu ermöglichen, wurde an der Motorhalterung ein weiteres 6 mm dickes Stahlstück angeschweißt. In dieses Teil wurden zuvor 4 Langlöcher gefräst. An diesen Löchern wurde der gekaufte Kettenspanner befestigt. Der Kettenspanner kann dort problemlos nach oben oder unten geschoben werden, um eine optimale Spannung der Kette zu erhalten. Nach Beendigung dieser Bearbeitungsschritte wurde eine neue Oberfläche durch Sandstrahlen und Lackieren erzeugt.

## 5.4 Kettenantrieb

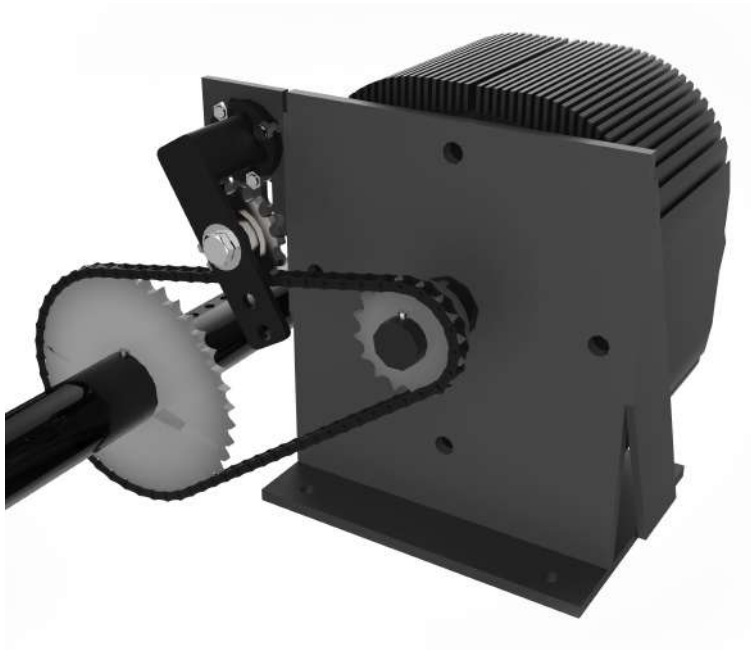


Abbildung 5.4: Gesamtantrieb

Um die Leistung vom Motor auf die Hinterachse zu übertragen, wurde ein Kettenantrieb realisiert, da für einen Zahnriemenantrieb mit gleicher Antriebsleistung zu wenig Platz zur Verfügung stand. Der Kettenantrieb wurde von der Firma SKF erworben, dabei wurde ein Übersetzungsverhältnis von 2,5 gewählt. Der Kettenspanner und das dazugehörige Zahnrad wurden ebenfalls von der Firma SKF erworben.

### 5.4.1 Auslegung

Als Kettenart wurde anhand der berechneten Werte die Art 08B ausgewählt. Laut Datenblatt ist diese Kette den notwendigen Ansprüchen gewachsen und sollte damit ohne Probleme verwendet werden können. Die beiden zugehörigen Zahnräder wurden mit 40 und 16 Zähnen gewählt. Somit konnte die Übersetzung von 2,5 vollständig umgesetzt werden.

### 5.4.2 Bearbeitung der gekauften Komponenten

Die gekauften Zahnräder der Firma SKF wurden auf den richtigen Innendurchmesser der 2 Wellen aufgedreht. Dabei wurde ein Produktionsfehler am größeren Zahnrad festgestellt. Da dieses jedoch schon bearbeitet wurde, konnte es nicht mehr zurückgegeben werden und musste zeitaufwändig nachgebessert werden. Um in die Zahnräder eine Keilnut zu pressen, wurden Führungsbuchsen für die Passfedernadel durch Drehen und Fräsen hergestellt. Nach dem Pressen wurden in jedes Zahnrad drei um jeweils  $120^\circ$  verschobene Gewinde gebohrt, um die Zahnräder an den Wellen fixieren zu können. Des Weiteren trat ein zusätzliches Problem auf, während die Passfeder an der Motorwelle bearbeitet wurde. Die Motorwelle wurde mit einer  $\frac{1}{4}$ " Passfeder ausgestattet. Um die dazugehörige passende Keilnut zu pressen, benötigt man eine  $\frac{1}{4}$ " Passfedernadel, welche in den Werkstätten aber nicht vorhanden war. Deshalb wurde eine 8 mm Keilnut in das kleinere Zahnrad gepresst und eine neue Passfeder angefertigt die auf der einen Seite das  $\frac{1}{4}$ " Maß und auf der anderen Seite die Breite von 8 mm aufweist. Nach diesen Arbeiten konnten die Zahnräder problemlos auf die Wellen geschoben werden, lediglich die Hinterachse des Karts musste dazu ausgebaut werden. Nach der Montage des Zahnrades wurde die Hinterachse wieder zusammengebaut.

### 5.4.3 Höherlegen des Karts hinten

Da bei der Anbringung der Kette festgestellt wurde, dass diese um ein halbes Glied zu kurz oder zu lang war, wurde zum Ausgleich dazu das Kart hinten höhergelegt. Dadurch befand sich die veränderte Position der Motorwelle im Vergleich zur Hinterachse noch höher und der Kettenweg wurde somit verlängert. Dadurch konnte die Kettenlänge ideal angepasst werden.

### 5.4.4 Aufstockung der Motorhalterung

Um den Rest des Kettenweges auch noch anzupassen, wurde der gesamte Motor mit der Motorhalterung mittels Unterlegblechen um 5 mm erhöht. Dies wurde durch zwei

Bleche zum Unterlegen mit zwei und drei Millimeter Dicke realisiert. Diese wurden mit denselben Bohrlöchern, wie jene des unteren Teils der Motorhalterung, versehen.

#### **5.4.5 Arbeiten am Kettenspanner**



Abbildung 5.5: Kettenspanner

Um das Kettenspannrad am Kettenspanner befestigen zu können, wurde eine M10 Schraube verwendet, da sich auf dem Kettenspanner schon vorgebohrte 10 mm Löcher befanden. Da der Innendurchmesser des Spannrades 14 mm betrug, wurde ein Zwischenstück hergestellt. Mit passenden Muttern und Unterlagscheiben wurde das Spannrad mit dem Zwischenstück mittels einer Schraube am Kettenspanner angebracht. Die Kette sollte ursprünglich von innen nach außen gespannt werden. Da aber im Verlauf der Montage des Antriebs festgestellt wurde, dass die Gefahr einer Kollision zweier entgegengesetzt drehender Zahnräder zu groß wäre, wurde sich für die Variante einer Spannung von außen nach innen entschieden. Dafür wurde ein zusätzliches Loch im Kettenspanner gebohrt, in welches die M10 Schraube mit dem Zahnrad eingesetzt wurde.

### **5.5 Schutz Kettenantrieb**

Da eine Berührung der Kette im Fahrbetrieb extrem gefährlich ist und speziell bei hohen Geschwindigkeiten zu ernsthaften Verletzungen führen kann, muss der Antrieb durch eine Abdeckung geschützt werden.





Abbildung 5.6: Seitenabdeckung

### 5.5.1 Seitenabdeckung

Da die Abdeckung des Kettenantriebes nicht in einem Teil ausgeführt werden konnte, weil diese sonst nicht mehr beliebig auf und abmontiert werden könnte, wurden zwei Einzelteile hergestellt, wobei das Seitenteil dauerhaft montiert bleibt. Das Werkstück wurde aus einem 3 mm Alu Blech gefertigt. Das Seitenteil wurde mit den M8 Schrauben, mit denen die Achsaufhängung am Chassis befestigt ist, mitgeschraubt.

### 5.5.2 Hauptabdeckung

Das Hauptstück zum Schutz vor der Kette besteht aus zwei Einzelteilen, welche durch drei Schrauben miteinander verbunden wurden. Diese Teile wurden aus einem drei Millimeter Alu Blech gefertigt. Zur Erstellung waren einige komplexere Arbeitsschritte notwendig, wie zum Beispiel komplizierte Biegearbeiten und Schweißaufgaben. Alle Werkstücke des Schutzes wurden zum Schluss sandgestrahlt und mit Klarlack lackiert. Die Hauptabdeckung wurde mit zwei Schrauben des Motors an die Motorhalterung mitgeschraubt.

### 5.5.3 Schutzmaßnahme gegen Verbiegen

Da der Motor hinter der Achse befestigt wurde und im mittleren Teil des Karts die Akkus den ganzen Platz verbrauchten, wurde, als reine Vorsichtsmaßnahme gegen das Verbiegen der Verlängerung, ein Alu Block passend gefräst und am Seitenteil der Abdeckung befestigt. Diese Schutzmaßnahme soll die vom Motor erzeugte Belastung aufnehmen und an die Achsaufhängung weiterleiten. Das Werkstück wurde abschließend sandgestrahlt und lackiert.

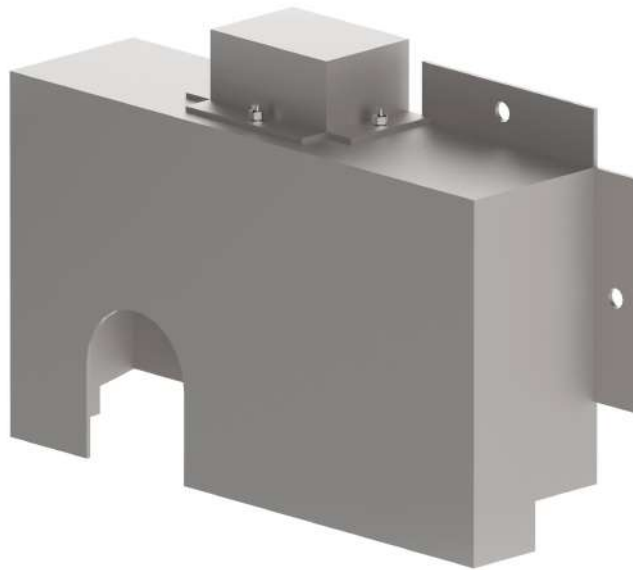


Abbildung 5.7: Hauptabdeckung

## 5.6 Schutz Motorcontroller

Um einen Kurzschluss zu verhindern, welcher nicht ungefährliche Folgen mit sich führen würde, wurde eine Abdeckung zum Schutz für den Motorcontroller konstruiert und gebaut. Diese besteht aus Abstandshaltern und dem Abdeckungsgehäuse.

### 5.6.1 Abstandshalter



Abbildung 5.8: Abstandshalter

Die Abstandshalter wurden aus einer 16 mm Vollwelle aus Aluminium hergestellt. Diese wurde als erstes auf einen Durchmesser von 15 mm gedreht und anschließend auf die passende Länge von 90 mm abgestochen. Danach wurde in die Mitte der Welle ein Loch gebohrt und anschließend ein M8 Gewinde hineingeschnitten. Insgesamt wurden vier Abstandshalter angefertigt.

## 5.6.2 Abdeckungsgehäuse

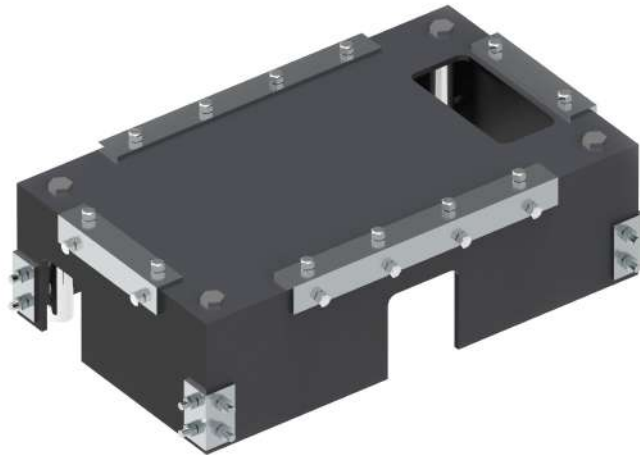


Abbildung 5.9: Abdeckungsgehäuse

Die Abdeckungsbox wurde aus 4 mm dicken Kunststoffplatten angefertigt, welche mit hergestellten Blechwinkeln miteinander verschraubt wurden. Abschließend wurden passende Ausschnitte zur Durchführung der Kabel angefertigt.

## 5.7 Sitz

Bei dem verwendeten Sitz handelt es sich um einen IMAF Racing Seat aus Polyester.

### 5.7.1 Montage des Sitzes

Der Sitz wird seitlich an zwei Stützen und vorne mittels zwei M8 Schrauben befestigt. Für die richtige Montage ist es wichtig, den Sitz mit Hilfe von Kunststoffabstandshaltern richtig auszurichten und möglichst spannungsfrei zu befestigen.

### 5.7.2 Unterboden

Da unter dem Sitz ein großer Freiraum war, wurde dieser durch ein 4 mm dickes Aluminiumblech verbaut. Dieses hat einerseits die Funktion, die Aufhängung des Sitzes zu entlasten und somit den Sitz besser zu stabilisieren. Andererseits wird



Abbildung 5.10: Unterboden

dadurch der Fahrer im Falle eines Aufkommens am Boden besser geschützt, da sich sonst nur der Sitz zwischen Fahrer und Boden befinden würde. Zudem bietet der Unterboden die Möglichkeit, eine Verkabelungsbox anzuschrauben und eine bessere Luftströmung unter dem Kart zu erzeugen, die für das Kühlen des Motors und Motorcontrollers benötigt wird.

## 5.8 Lenkrad

### 5.8.1 Displayhalterung

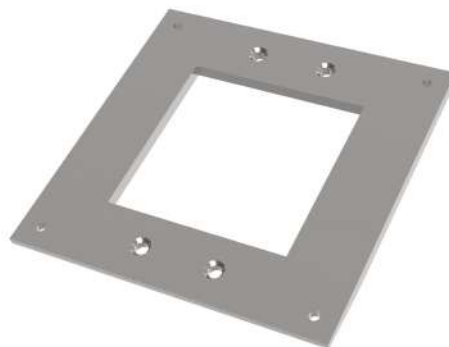


Abbildung 5.11: Displayhalterung

Um einen hohen Fahrkomfort zu erhalten und gleichzeitig möglichst viele Daten während des Fahrens messen zu können, wurde in das bestehende Lenkrad der

Firma CKR ein 3,5" Display der Firma Sigmatek eingebaut. Dazu wurde als Adapterplatte ein 4 mm-Aluminiumblech in der Größe des Displays gefertigt. Die weiteren Arbeitsschritte werden im nächsten Unterpunkt beschrieben.

### 5.8.2 Diverse Arbeiten am Lenkrad

Damit das gefertigte Aluminiumteil am Lenkrad befestigt werden kann, musste ein Teil der drei Verstrebungen des Lenkrads entfernt werden. Anschließend erhielt das Lenkrad vier Bohrungen, damit die Halterung durch M4 Schrauben fest verbunden werden kann. Danach konnte das Bedienfeld auf dieser Halterung mit M3-Schrauben befestigt werden. Da bei dieser Halterung eine Dicke von 4 mm gewählt wurde, wird die Festigkeit des Lenkrads nur geringfügig beeinflusst. Ebenso sind im Lenkrad zwei weitere Bedienelemente verbaut. Für das Rückwärtsfahren ist ein Kippschalter in ein vorhandenes Loch im Lenkrad eingepasst. Um den Taster für die Antriebsschlupfregelung zu befestigen, musste ein Loch auf 25 mm aufgebohrt werden.



Abbildung 5.12: Lenkrad

### 5.8.3 Schutz auf der Hinterseite

Spritzwasserschutz und Schutz für die Bauteile wird durch ein Kunststoffteil aus dem 3D-Drucker erreicht. Der Schutz wurde genau an die Formen des Lenkrads angepasst und verdeckt zahlreiche Kabel und Schrauben, die zur Befestigung notwendig sind.



Abbildung 5.13: Armaturenbrett

## 5.9 Bedienoberfläche

### 5.9.1 Armaturenbrett

Oberhalb der Steuerung an der Halterung des Lenkrads befindet sich das Armaturenbrett mit den notwendigen Schlüsselschaltern und Tastern für diverse Schaltzustände. Darunter sitzt der für die Sicherheit relevante NOT-AUS Schalter. Beim Fertigen des Armaturenbrettes wurde ein 2 mm dickes Aluminiumblech in mehreren Arbeitsschritten zugeschnitten und gebogen, um den vorgegebenen Formen des Karts zu folgen. Da durch das ständige Anpassen die Oberfläche beschädigt wurde, erhielt das Blech durch Sandstrahlen und anschließend Lackieren mit Klarlack wieder eine neue Oberfläche. Die notwendigen Bohrungen wurden erst ganz zum Schluss gemacht, da diese genau an die Schlüsselschalter und Taster angepasst wurden.

### 5.9.2 Abdeckung der Kabel auf der Hinterseite

Durch ein Kunststoffteil aus dem 3D-Drucker werden die elektrischen Bauteile und deren Anschlüsse vor Spritzwasser und Schmutz geschützt. Zur Befestigung werden die gleichen Schrauben wie die des Armaturenbretts verwendet.

## 5.10 Kunststoffgehäuse SPS

Unter dem Lenkrad zwischen den Beinen des Fahrers, wo sich bei herkömmlichen Karts mit Verbrennungsmotor der Treibstoffbehälter befindet, ist bei diesem Kart das Herzstück, die Steuerung der Firma Sigmatek, verbaut. Diese befindet sich in einem



Abbildung 5.14: Kunststoffgehäuse SPS

Kunststoffgehäuse, bestehend aus fünf PVC Teilen und einem Plexiglasdeckel. Diese Box wurde aus Stabilitätsgründen mit mehreren Metallwinkeln zusammengeschaubt und nicht geklebt. An der Bodenplatte ist eine Hutschiene verschraubt, worauf die einzelnen Module der Steuerung befestigt werden. Um eine saubere Verkabelung zu gewährleisten, ist innerhalb des Gehäuses ein Kabelkanal angebracht. Das Gehäuse wird durch vier M6 Senkkopfschrauben fest mit dem Unterboden des Fahrzeuges verbunden.

## 5.11 Gaspedalauswertung

### 5.11.1 Adapter auf dem Gaspedal

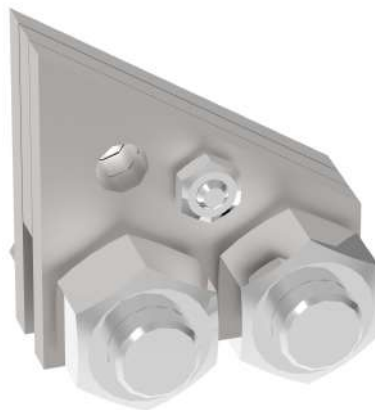


Abbildung 5.15: Gaspedaladapter

Da das Linearpotentiometer zur Gaspedalauswertung nicht einfach so befestigt werden konnte, musste dafür ein Adapter gebaut werden. Dieser besteht unter anderem aus Stabilitätsgründen aus drei einzeln angefertigten 3 mm-Alublechen, wobei die beiden Außenteile identisch sind. Der Adapter wird über zwei 3 mm Löcher zusammengeschraubt und mithilfe von 8 mm Löcher am Gaspedal befestigt. An einem 5 mm Loch wird der Adapter mit einer Schraube mit dem Linearpotentiometer verbunden.

### 5.11.2 Adapter zum Kartrahmen

Der zweite Fixpunkt für das Linearpotentiometer wird durch einen weiteren Adapter am Kartrahmen realisiert. Das Grundgerüst des Adapters ist ein Aluminiumblock mit den Maßen 100x200x25 mm, welcher durch Fräsen die gewünschte Form erhielt. Um das Linearpotentiometer zu befestigen, befindet sich am oberen Ende eine 5 mm Bohrung, durch die eine M5-Zylinderschraube gesteckt wurde. Am unteren Punkt ist eine Kerbung eingefräst, mit der der Adapter auf ein Rohr des Chassis geschoben wird. Mit zwei M5-Senkkopfschrauben wird der Adapter an seinem Platz gehalten.



Abbildung 5.16: Adapter zum Kartrahmen

### 5.11.3 Schutz Linearpotentiometer

Um das für die Auswertung des Gaspedals notwendige Linearpotentiometer vor Beschädigungen zu schützen, wurde eine Abdeckung aus 3mm-Aluminiumblech gefertigt. Um die gewünschte Form und Belastbarkeit zu erhalten, wurde es mehrmals gebogen und anschließend an der vorhandenen Bodenplatte und der eigens gefertigten Halterung des Linearpotentiometers verschraubt. Das Teil erhielt ebenfalls durch Sandstrahlen und anschließendes Lackieren mit Klarlack eine schönere Oberfläche.





Abbildung 5.17: Linearpotentiometerschutz

## 5.12 Endschalterbefestigung für die Auswertung der Bremse

Um das Kart nicht unnötig durch gleichzeitiges Beschleunigen und Bremsen zu belasten, wurde ein Endschalter (Schnappschalter) auf den Bremsflüssigkeitsbehältern montiert, welcher betätigt wird, wenn die Bremse nicht gedrückt ist. Funktionell wird der Schließerkontakt des Schalters ausgewertet und ein digitales High-Signal an die SPS weitergeleitet, wenn nicht gebremst wird.

### 5.12.1 Befestigungsblech



Abbildung 5.18: Befestigungsblech

Um den Schalter montieren zu können, wurde ein Blechstück mit einer Dicke von 3 mm aus Aluminium gefertigt. Dieses wird anschließend über die Schrauben der Bremsflüssigkeitsbehälter angeschraubt. Der Schalter wird mit zwei M4-Senkkopfschrauben befestigt.

### 5.12.2 Abdeckung

Als Abdeckung wird ein Kunststoffteil aus dem 3D-Drucker gefertigt, welches den Endschalter vor Beschädigungen, Schmutz und Spritzwasser schützen soll. An der Vorderseite ist eine Öffnung ausgeführt, wo das Schalterstück auf den Hebel der Bremse drückt. Die Abdeckung wird an drei Punkten befestigt.

## 5.13 Halterungsbleche für die induktiven Sensoren



Abbildung 5.19: Halterung der induktiven Sensoren

Für eine genaue Antriebsschlupfregelung werden die Drehzahlen der Vorder- und Hinterachse benötigt. Die Drehzahl der Hinterachse wird durch den Motorcontroller ermittelt. Für die Drehzahl der Vorderräder werden zwei Drehzahlsensoren mittels eines eigens gefertigten Werkstücks an der Aufhängung des rechten und linken Vorderreifens angebracht. Bei dem Werkstück handelt es sich um ein 2 mm Aluminiumblech. Der induktive Sensor wird durch eine Bohrung der Größe 12 mm geführt und mit Muttern befestigt. Die Halterung wird einerseits durch eine Schraube, an der die Bremsbacke mit der Aufhängung verbunden ist, und andererseits mit einer zusätzlichen M5-Schraube befestigt.

## 5.14 Batterieboxen

Die Akkumulatoren werden in zwei Teile aufgeteilt, wobei der rechte etwas größer und mit 68 kg um 26 kg schwerer als der linke ist. Zu dieser Asymmetrie kommt es, da das Chassis für einen Verbrennungsmotor auf der rechten Seite aus der Sicht des Fahrers ausgelegt ist.



Abbildung 5.20: Batterieboxen

### 5.14.1 Auslegung der Boxen

Für beide Batterieblöcke wurde ein Aluminiumgehäuse als Schutz und Befestigungsmöglichkeit angefertigt. Das Gehäuse ist aus 3 mm Aluminiumblech zusammengeschweißt worden. Es ist etwas größer als das Abmaß der Batterieblöcke, damit zwischen den Zellen und dem Gehäuse zwecks der Isolation eine Kunststoffplatte und eine Dämmung eingepasst werden können. Diese bietet einerseits den Schutz gegen einen Kurzschluss mit dem Gehäuse und andererseits die Möglichkeit, die Zellen so stark aneinander zu pressen, dass sie sich selbstständig nicht mehr lösen können. Die Anschlüsse der Zellen werden durch zwei Kabeldurchführungen aus den Boxen herausgeführt. Der Pluspol führt zu einem an die Box geschraubten Terminal, in dem eine Schmelzsicherung untergebracht ist. Eine weitere Kabeldurchführung verbindet die Sensorleitungen des Batteriemanagement-Systems. Um die Fläche der Batterieboxen zu verschönern, wird eine Folie mit einem eigenen Design angefertigt. Diese zeigt auch die zahlreichen Sponsoren, die das Projekt ermöglichten.

### 5.14.2 Maßnahmen zur Verstärkung des Kartrahmens

Um eine Verbiegung des Rahmens aufgrund der Schwere der Batterien zu verhindern, wurde in den Hohlrohren des Seitenteils eine Aluminiumwelle eingepasst, dadurch soll sich dieser Teil zusätzlich versteifen. Durch das enorme Gewicht der Batterieboxen wurde die Kunststoffhülse, in die das Seitenteil geschoben wird, verformt. Um der Verformung zu entgegnen wurden diese Teile entfernt und durch Aluminiumhülsen in derselben Form ersetzt. Durch eine Presspassung soll kein Spiel auftreten, damit sich die Batterieboxen nicht in Richtung Boden neigen.



Abbildung 5.21: Wellenhalter

### 5.14.3 Ebnungsbleche

Dabei handelt es sich um zwei Bleche auf beiden Seiten, die den Größenunterschied zwischen den Rohren des Chassis und den Rohren der Seitenteile ausgleichen sollen. Es werden zwei Bleche mit der Dicke von 2- und 3 mm verwendet. Des weiteren bieten die Bleche eine möglichst große Auflagefläche für die insgesamt 110kg schweren Batterien.

## 5.15 3D-Druck Teile

Es gibt viele verschiedene Varianten von 3D Druckern und 3D Druckverfahren, allerdings ähneln sich einige dieser Konzepte. Alle zusammen bieten allerdings einen enormen Vorteil gegenüber herkömmlichen Herstellungsverfahren: Sie können mit einer hohen Präzision auch sehr komplexe Konstruktionen herstellen.

Grundsätzlich wird beim dreidimensionalen Druck als Ausgangsdatei ein 3D Modell benötigt, welches im Normalfall mittels CAD Software entworfen wird. Um dieses Modell drucken zu können, muss es zuerst in einem Slicer Programm in einzelne Schichten unterteilt und anschließend an den Drucker übertragen werden. Der jeweilige Drucker trägt anschließend Schicht um Schicht übereinander auf und modelliert so das endgültige Bauteil. Dabei gibt es verschiedene Verfahren, wie diese einzelnen Schichten übereinander aufgetragen werden. Bei diesem Projekt wurden speziell zwei verschiedene Varianten des 3D Druckens zum Herstellen einiger komplexerer Bauteile verwendet: einerseits das mittlerweile sehr bekannte Fused Deposition Modeling (FDM; deutsch: Schmelzschichtung) und andererseits das eher in der Industrie eingesetzte selektive Lasersintern (SLS).

### 5.15.1 Fused Deposition Modeling

Dabei wird mit geschmolzenem Material gedruckt. Dazu wird das verwendete Material, im Normalfall diverse Kunststoffe, in einer Düse bis knapp über den Schmelzpunkt erhitzt und anschließend Schicht für Schicht aufgetragen. Während des Auftrags kühlt der Kunststoff ab und verfestigt sich dabei.<sup>12 13</sup>

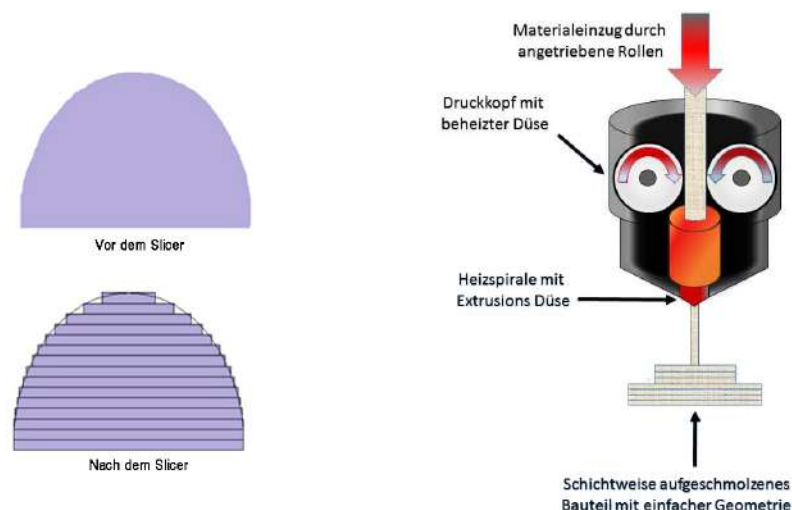


Abbildung 5.22: Funktionsprinzip des „Fused Deposition Modeling“ Verfahrens

<sup>12</sup>vgl: 3D-Drucken/Vor und Nachteile, <https://goo.gl/4zB888> (Stand: 23.03.2017)

<sup>13</sup>vgl: Rapid prototyping slicing, <https://goo.gl/U042Q1> (Stand: 23.03.2017)

Bei der Herstellung der Abdeckungen für das Kart wurden zwei verschiedene Materialien verwendet: der 3D-Drucker an unserer Schule druckt den Kunststoff ABS (Acrylnitril-Butadien-Styrol), der Drucker unseres Sponsors Bürklin verwendet ein Nylon-Karbon Gemisch zum Drucken. Dieses Nylon-Karbon Gemisch ist um einiges stabiler als das Acrylnitril-Butadien-Styrol. Während ABS weißlich ist und nach dem Drucken erst noch lackiert werden muss, ist das von Bürklin verwendete Gemisch schwarz und passt daher bereits ohne lackieren gut zum Kart.

### 5.15.2 Abdeckung Touchpanel

Da die originale Abdeckung des Sigmatek-Touchpanels bei Arbeiten am Lenkrad beschädigt wurde, war es notwendig, eine neue Abdeckung produzieren zu lassen. Davon abgesehen hatte diese Konstruktion einige weitere kleine Schwachstellen, weshalb eine Neu-Entwicklung Sinn machte. Dazu wurde eine von Sigmatek erhaltene CAD Zeichnung des Bauteiles als Grundlage verwendet und an unsere Bedürfnisse angepasst. Darüber hinaus wurden der Name der Schule und die Namen der Projektbeteiligten in diese Abdeckung eingedruckt.



Abbildung 5.23: 3D-Modell der Abdeckung für das Touch Panel

### 5.15.3 Abdeckung Endschalter

Um mit der SPS auswerten zu können, zu welchem Zeitpunkt die Bremsen betätigt sind, wurde auf den Bremsflüssigkeitsbehälter ein Endschalter platziert. Damit dieses Bauteil geschützt wird, wurde dafür eigens eine Abdeckung designed. Diese Abdeckung wird mit den bereits bestehenden Schrauben befestigt.



Abbildung 5.24: 3D-Modell der Abdeckung des Endschalters

### 5.15.4 Rückverkleidung der Lenkradeinheit

Damit eine geordnete und vor direkten Einwirkungen geschützte Verkabelung der einzelnen Schalter und Komponenten der Lenkradeinheit sichergestellt ist, wurde eine Abdeckung für die Rückseite der Lenkradeinheit in Solid Edge entwickelt und mit dem 3D-Drucker gefertigt. Dabei war es eine besondere Herausforderung, dass die drei vom Mittelpunkt wegführenden Speichen in einem Winkel von  $8^\circ$  aufsteigen. Eine weitere speziell beim Drucken des Bauteils bestehende Herausforderung war, dass die großen Flächen Gefahr laufen, ungleich abzukühlen und sich folglich verformen. Dieses Phänomen tritt beim 3D-Drucker unserer Schule speziell in der linken hinteren Ecke auf und konnte nur leicht dadurch korrigiert werden, indem das Druckteil genau in die gegengleiche Ecke verschoben wurde. Da bei einer großen Fläche dennoch eine ähnliche Wölbung aufgetreten ist, wurde diese anschließend mit dem Handfräser nachbearbeitet.



Abbildung 5.25: 3D-Modell der Rückverkleidung der Lenkradeinheit

### 5.15.5 Rückverkleidung Armaturenbrett

Auf der Rückseite des Armaturenbretts wurde ebenfalls eine Verkleidung befestigt. Diese Abdeckung wurde anhand der Maße des Solid Edge Modelles gezeichnet und durch den 3D-Drucker produziert. Hier war problematisch, dass die Schrauben, mit welchen das Armaturenbrett auf der Lenksäule befestigt wurde und mit denen auch diese Rückverkleidung angeschraubt wird, in einem leichten Winkel zur Seite abstehen. Um das zu korrigieren, wurde eine vergrößerte Ausnehmung vorgesehen. Damit die Abdeckung anschließend dennoch stabil hält, wurden zwischen Mutter und Abdeckung Unterlegscheiben verwendet.

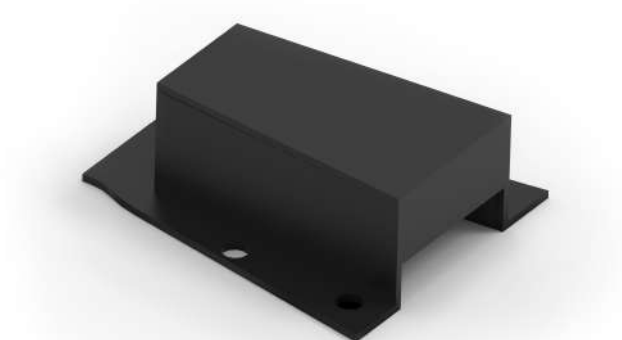


Abbildung 5.26: 3D-Modell der Rückverkleidung des Armaturenbretts



### 5.15.6 Selektives Lasersintern

Bei diesem Druckverfahren liegt das Material, in das gedruckt wird, in Pulverform vor. Dieses Pulver wird an den zu druckenden Stellen durch einen Laser erhitzt und verschmolzen. Anschließend wird wieder eine Schicht Pulver darüber gegeben und erneut verschmolzen. So entsteht das Bauteil Schicht für Schicht. Dieses Druckverfahren hat die Vorteile, dass hier viel kleinere Toleranzen eingehalten werden können und auch keine Stützstrukturen benötigt werden, da das nicht gesinterte Pulver als Stützstruktur dient und nach dem Druck einfach abgeblasen werden kann.<sup>14</sup>

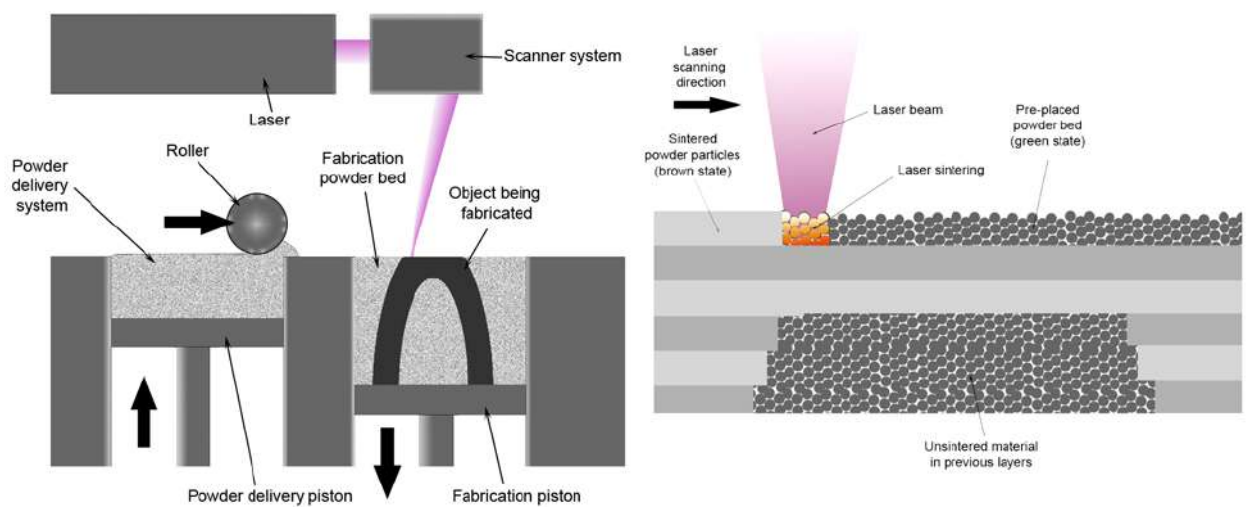


Abbildung 5.27: Funktionsprinzip des „Selektives Lasersintern“-Verfahrens

<sup>14</sup>vgl: Selective Laser Sintering and Melting, <https://goo.gl/76kPbm> (Stand: 23.03.2017)

#### 5.15.6.1 Abdeckung für das Isolationsüberwachungssystem

Da die Abdeckung für das Isolationsüberwachungssystem einerseits sehr lang ist und sich deshalb im 3D Drucker verformen würde, aber andererseits sehr genau sein muss, damit die Platine darauf passend einen sicheren Halt bekommt, wurde dieses Bauteil von Bürklin durch das Verfahren des selektiven Lasersinterns gefertigt.<sup>15</sup>



Abbildung 5.28: 3D-Modell der Abdeckung für das Isolationsüberwachungssystem

---

<sup>15</sup>vgl: Übersicht der aktuellen 3D-Druckverfahren, <https://goo.gl/TPLI05> (Stand: 23.03.2017)

# Kapitel 6

## Software

### 6.1 Programm der SPS

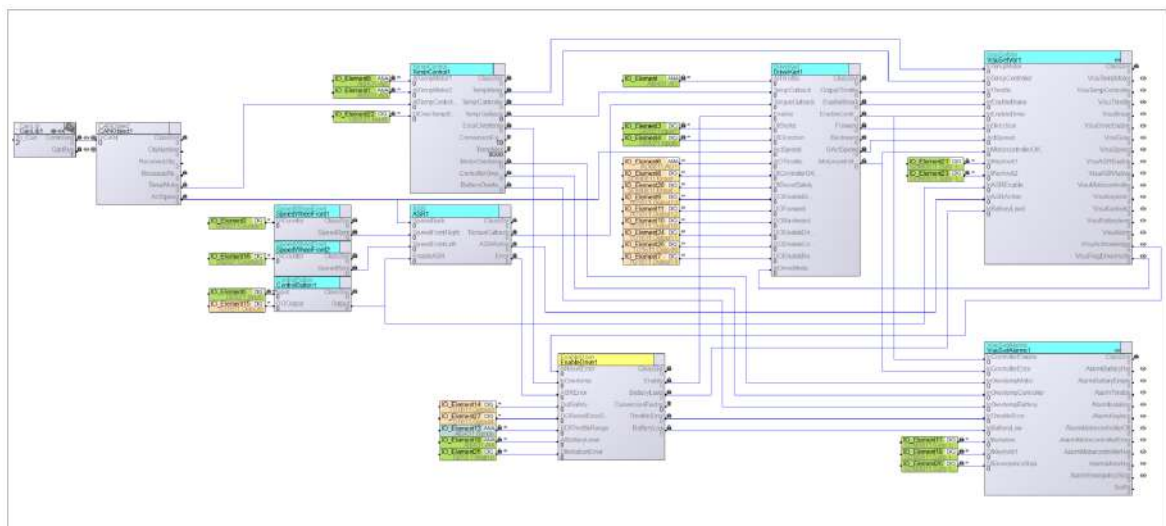


Abbildung 6.1: Programm der SPS - Übersicht

Abbildung 6.1 zeigt das gesamte Programm der Steuerung. Die verwendete Entwicklungsumgebung ist „LASAL Class 2“ von Sigmatek. Geschrieben wurde das Programm in der Programmiersprache Structured Text. Alle erstellten Quellcodes sind im Anhang unter dem Kapitel ?? nachlesbar.

Jedes der hier zu sehenden Rechtecke stellt eine eigene Klasse dar. Diese Klassen beinhalten Clients, Server, Variablen, Funktionen usw.. Über die Clients, welche links in den Klassen abgebildet sind, können Daten mit IOs ausgetauscht werden. Über die Server, welche sich rechts befinden, werden Daten gespeichert und an andere Klassen weitergegeben.

### 6.1.1 DriveKart

In der Mitte rechts befindet sich die Hauptklasse „DriveKart“ (siehe Abbildung 6.2). Über diese wird der Motorcontroller direkt angesteuert. Hier werden außerdem sämtliche I/Os verarbeitet, mit denen der Benutzer das Go-Kart steuert. Über den ersten Client wird der Analogwert des Gaspedals eingelesen. Im Anschluss wird das Gaspedal über einen Offset und einen Faktor auf brauchbare Werte umgerechnet.

Über die Clients „TempCutback“ und „TorqueCutback“ werden die Werte der Regler übertragen. Diese Werte geben eine Verminderung der Motorleistung vor. Gesamt ergibt sich so das aktuelle Gas, welches über den Client „AOThrottle“ dem analogen Ausgangsmodul AO081 übergeben wird. Über die Clients „DIBrake“ und „DIDirection“ werden Bremspedalstellung und Fahrtrichtung eingelesen. Bei der Bremse sei zu beachten, dass der Wert bei einer Betätigung des Pedals Low ist. Dies hat den Grund, dass, falls ein Kabelbruch passieren sollte, zumindest kein Fahren mit gedrückter Bremse möglich sein kann. Ein gleichzeitiges Betätigen von Bremse und Gas könnte sonst zu ernsthaften Problemen und im schlimmsten Fall sogar zum Unfall führen. Die Fahrtrichtung darf nur bei stehendem Fahrzeug umgeschaltet werden können. Deshalb wird überprüft, ob die aktuelle Geschwindigkeit 0 und die Bremse betätigt ist.

Der Motorcontroller ist insgesamt mit sechs digitalen Eingängen am Ausgangsmodul TO161 angeschlossen. Dieses Ausgangsmodul wird mit Werten der „DriveKart“ Klasse gesteuert. Dabei werden Fahrerlaubnis, Bremsstellung, Fahrtrichtung sowie Fahrmodus übertragen. Wichtig ist, dass vor dem Aktivieren eines Ausgangs der Motorcontroller eingeschaltet und startbereit ist. Dies wird über den Client „DIControllerOK“ abgefragt. Ein Starten des Controllers bei aktiviertem Enable-Ausgang würde sonst zu einem Fehler des Controllers führen. Im Anschluss werden die Variablen per Server zur Visualisierung übertragen.

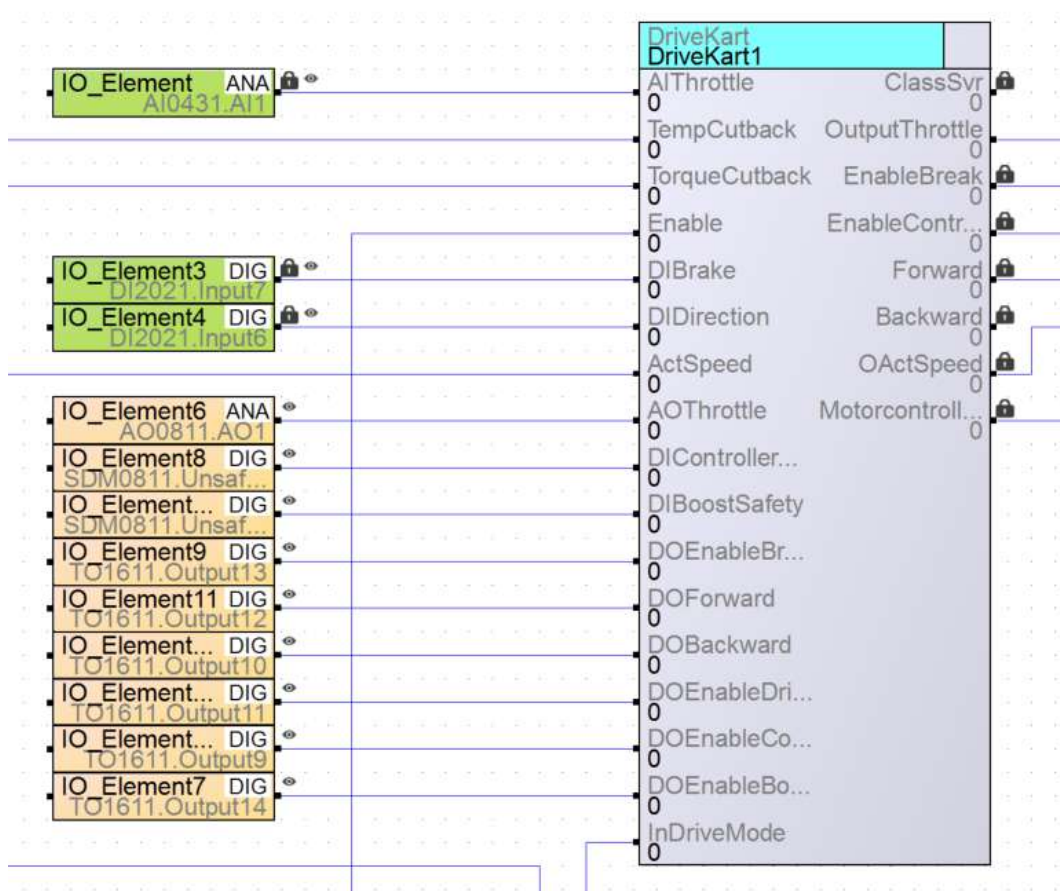


Abbildung 6.2: DriveKart Klasse

## 6.1.2 EnableDrive

Über die Klasse „EnableDrive“ wird die ordnungsgemäße Funktionalität des Fahrzeugs überwacht. Hier ist auch das Sicherheitsprogramm verknüpft. Über den Ausgang „Enable“ bekommt die Hauptklasse „DriveKart“ mit, ob gefahren werden darf. Angehängt an diese Klasse sind die Temperaturüberwachung, das BMS, die ASR, das Isolationsüberwachungssystem und natürlich die Safety-CPU. Der Akkustand wird von BMS geliefert und über das analoge Eingangsmodul AI043 eingelesen. Anschließend wird er dem Client „AIBatteryLevel“ übergeben und über den Faktor des Servers „ConversionFactor“ in Prozent umgerechnet. Sobald der Akkustand eine Grenze unterschreitet, muss das Fahrzeug gestoppt werden.

Das analoge Eingangsmodul bietet die Möglichkeit, für den Wert des Gaspedals obere und untere Grenze zu definieren. Sobald sich der Wert aufgrund eines Kabelbruchs oder eines anderweitigen Fehlers in einem nicht erlaubten Bereich befindet, muss gestoppt werden.

Um die Klasse mit der Safety-SPS zu verbinden, müssen die Übergabevariablen einem Hardwareobjekt zugewiesen werden. Daher wird der Client „OutSafety“ auf

den zweiten Ausgang des digitalen Ausgangsmoduls gelegt, wodurch die Variable von der Safety-SPS eingelesen werden kann. Da der Funktionsblock des NOT-Aus Tasters nach Betätigung des Tasters einen Reset benötigt, wird die Reset – Variable, welche von der Visualisierung übermittelt wird, auf den dritten Ausgang des TO-Moduls gelegt, wodurch der Reset ebenfalls von der Safety-SPS eingelesen werden kann. Da auch die Fehler visualisiert werden sollen, werden hier gleich wie bei der Hauptklasse Übertragungsserver erstellt und am Ende mit Werten beschrieben. Als Zusatz wird über die Visualisierung eine Variable erstellt, mit der ein Fehler nach Auftreten und Bereinigen quittiert werden kann.

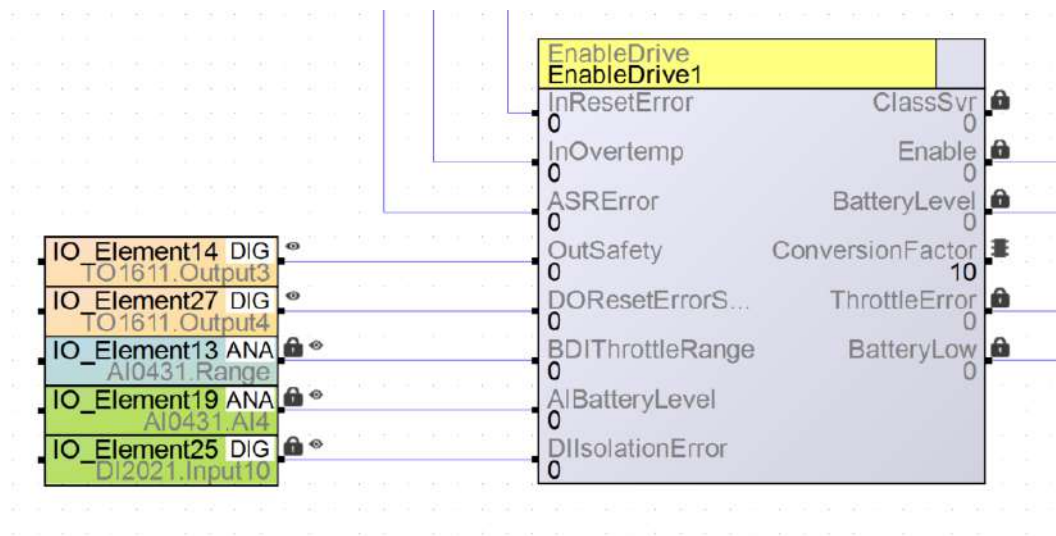


Abbildung 6.3: EnableDrive Klasse

### 6.1.3 ASR

Die Klasse ASR stellt eine Antriebsschlupfregelung dar. Über die beiden Klassen „SpeedWheelFront“ werden die Drehzahlen der Vorderräder eingelesen. Ein CAN-Objekt, welches mit dem Controller kommuniziert, übermittelt der Klasse die Geschwindigkeit der Hinterräder. Über die Klasse „ControlButton“ wird das ASR ein- bzw. ausgeschaltet. Anschließend werden die beiden Drehzahlen verglichen und über einen Regler wird die Verringerung der Motorleistung bestimmt. Dadurch soll verhindert werden, dass die Hinterachse unkontrolliert durchdreht.

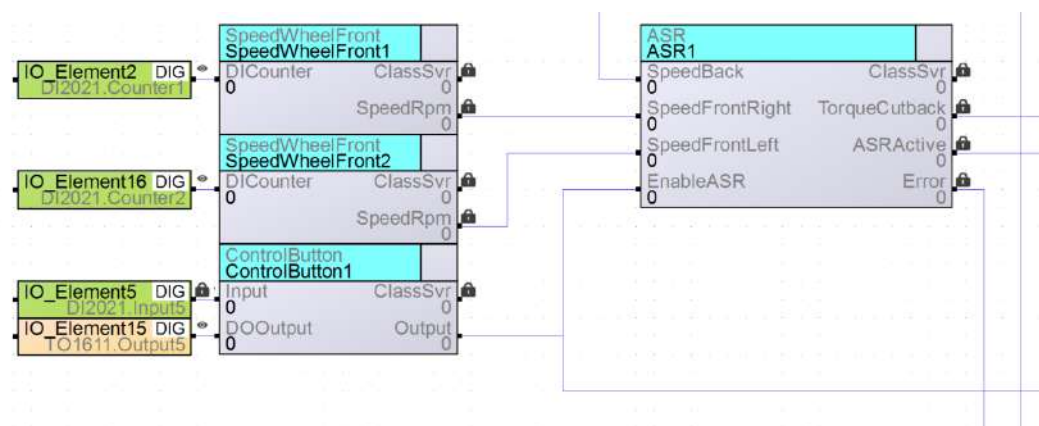


Abbildung 6.4: ASR Klasse

Bei der Klasse „SpeedWheelfront“ wird die Counterfunktion des DI202 Moduls verwendet. Dabei zählt das Modul unabhängig von der Zykluszeit alle steigenden Taktflanken an ihrem Eingang und speichert die Werte live in einem Counter (Zähler-Variable). Da die Bremsscheibe, auf die der Sensor gerichtet ist, dreimal gelocht ist, wird pro Umdrehung dreimal hoch gezählt. Im Anschluss kann auf diesen Counter problemlos zugegriffen werden. Zur Umrechnung des Counters auf die tatsächliche Drehzahl wird über die Systemzeit und einen Merker eine Zeitspanne von einer halben Sekunde erfasst. Anschließend kann bequem über einen Faktor zurückgerechnet werden. Dieser Faktor errechnet sich aus:

$$60 / ((Counts/Turn) * Time) = 60 / 3 + 0,5 = 40 \quad (6.1)$$

Die Klasse „Control Button“ stellt eine Flankenerkennung für den Einschaltknopf bereit. Sie verwaltet den Ausgang für den LED-Ring des Knopfs und übergibt mittels Server, ob die ASR aktiviert ist. Diese Flankenerkennung wird über einen einfachen Merker realisiert. Als Clients werden sowohl das Signal vom Button als auch ein Ausgang für den LED-Ring verlinkt.



## 6.1.4 TempControl

Die Klasse „TempControl“ implementiert eine Temperaturregelung. Die beiden Temperatursensoren des Motors werden analog eingelesen, die Temperatur des Controllers wird über zuvor genanntes CAN-Objekt übermittelt. Das BMS übermittelt über ein digitales Signal, ob die Batterien zu heiß sind. Zuerst werden alle Temperaturen auf ihre richtigen Werte umgerechnet. Dabei können über kaltstartsichere Server Umrechnungsfaktoren eingegeben werden. Im Anschluss werden alle vorliegenden Analogwerte verglichen, um dadurch den Maximalwert herauszufinden. Die Differenz zwischen Maximalwert und maximal erlaubter Temperatur stellt die Zurückstufung der Motorleistung dar, welche an den Controller weitergegeben werden soll. Zusätzlich wird speziell für die Visualisierung eine Übertemperaturerkennung für alle an die Klasse angeschlossenen Bauteile implementiert. Aber auch in der Sicherheitsklasse finden diese Variablen Verwendung: So darf nur gefahren werden, wenn keine Übertemperatur vorliegt. Sobald auch nur ein Bauteil eine gewisse Temperatur überschreitet, wird sofort abgeschaltet. Als Clients angeschlossen sind an die Klasse die drei Temperaturwerte von Motor und Controller sowie ein Eingang für eine Übertemperatur der Batterie. Als Ausgänge sind speziell für die Visualisierung die einzelnen Temperaturen umgerechnet in Grad Celsius sowie die Übertemperaturzustände der einzelnen Bauteile verfügbar. Des Weiteren bietet die Klasse einen Server für die Zurückschaltung der Motorleistung sowie einen für eine generelle Übertemperatur, welcher mit der Sicherheitsklasse „EnableDrive“ verbunden wird.

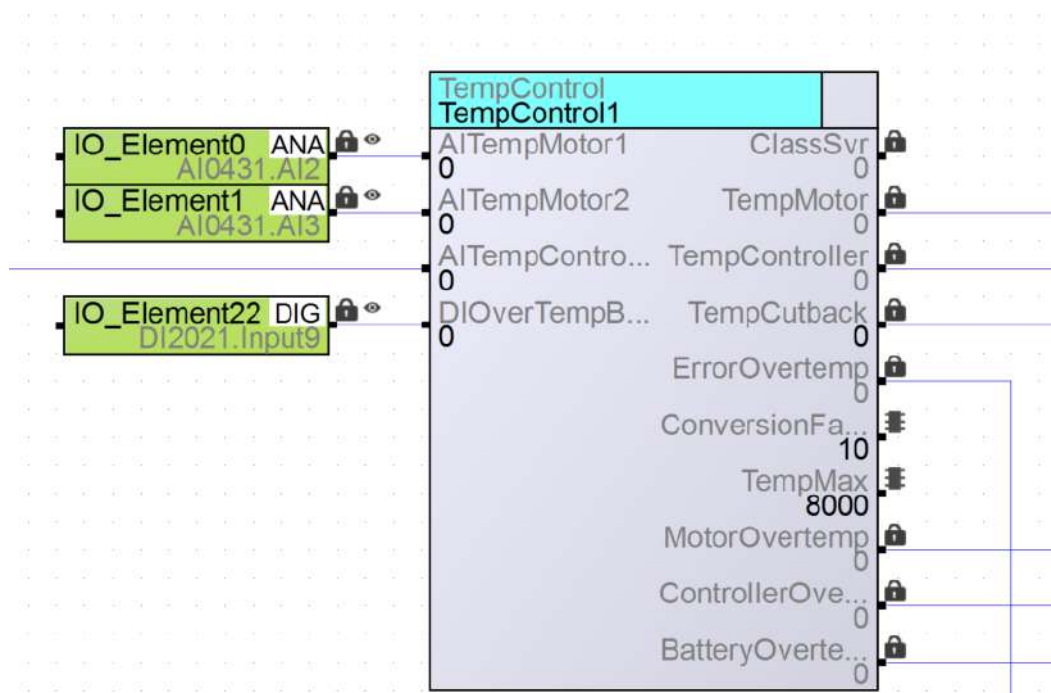


Abbildung 6.5: TempControl Klasse



### 6.1.5 CAN-Object

Zur Kommunikation mit dem Motorcontroller über CAN-Bus wird eine eigene Klasse verwendet. Diese Klasse wurde aus der Standardbibliothek von Sigmatek übernommen und um zwei Server zur Aufnahme von Werten erweitert. Eine nähere Beschreibung soll daher entfallen, diese kann in der Dokumentation der Sigmatek-Library nachgelesen werden.

### 6.1.6 Übergabeklassen für die Visualisierung

Da einige Daten visualisiert werden sollen, müssen sie an die Visualisierung übergeben werden. Dazu kann in den Einstellungen der jeweiligen Server die Option „Visualised“ ausgewählt werden. Zur besseren Überschaubarkeit wurden alle Variablen in einer Klasse übergeben. Ebenso wird für alle Fehler eine eigene Klasse erstellt. Diese beiden Klassen müssen vom normalen Programm mit Werten gefüllt werden. Eine Beschreibung des Quellcodes kann daher entfallen, da nur Werte umkopiert werden.

## 6.2 Safety - Programm

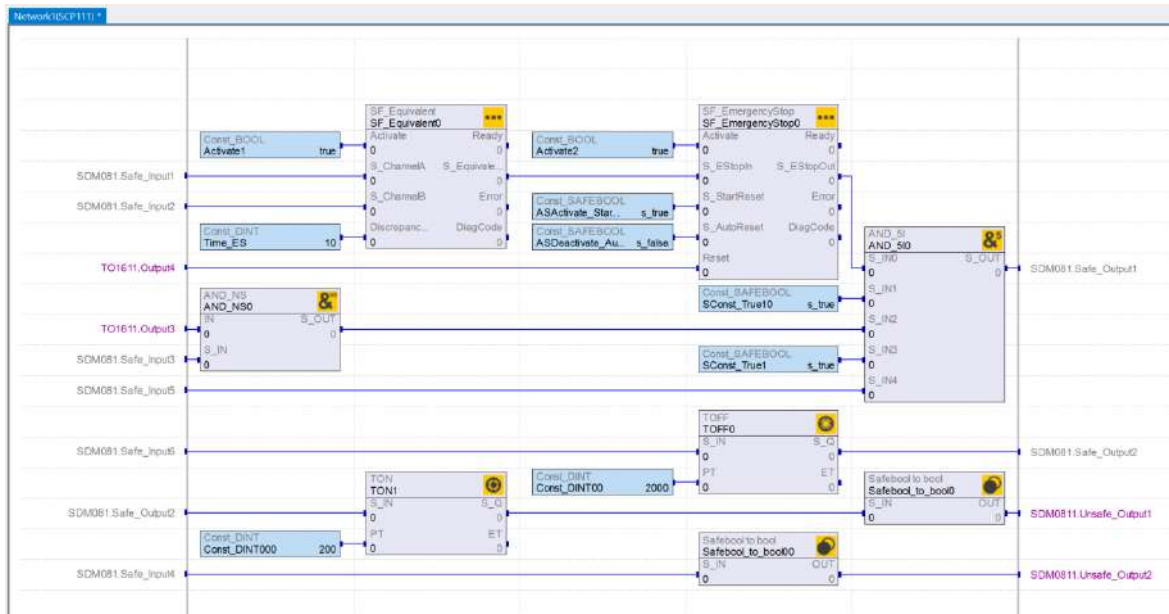


Abbildung 6.6: Programm der Safety - SPS

Gesamt besteht der Safety-Teil der SPS aus einem Hauptrechner (SCP111) sowie einem digitalen Ein- und Ausgangsmodul (SDM081). In dem SCP-Modul wird das Programm gespeichert und abgearbeitet, über das SDM-Modul werden alle sicherheitsrelevanten Bauteile angeschlossen. Im Bild zu sehen ist das gesamte Sicherheitsprogramm. Ganz links befinden sich die Eingänge, rechts die Ausgänge. Dazwischen sind mehrere Funktionsblöcke eingebunden. Programmiersprache ist hier FUB (Function Block Diagramm). Die Verknüpfung zur normalen SPS wird mittels „Unsafe Inputs“, also unsicheren Eingängen hergestellt. Dabei müssen jedoch die zu übertragenden Variablen auf einen Ausgang eines anderen Moduls gelegt werden. In diesem Fall werden dafür die ersten Ausgänge des TO161 – Moduls herangezogen, da diese sonst keine Verwendung finden würden. Um Daten der SPS weiterzugeben, sind „Unsafe Outputs“, also unsichere Ausgänge vorhanden. Auf dem Modul SDM081 sind bereits zwei dieser unsicheren Ausgänge definiert.

Die ersten beiden Eingänge des SDM-Moduls werden an den NOT-Aus Taster angeschlossen. Dabei wird die Funktion des Zwei-kanaligen Einlesens genutzt. Das bedeutet, dass die beiden Kanäle des NOT-Aus Tasters auf einer Seite nicht an 24V angeschlossen werden, sondern direkt an die Kanäle A und B des SDM Moduls. Die andere Seite der Kanäle liegt wie gehabt auf den beiden Eingängen. Dadurch ergibt sich die Möglichkeit einer Querschlusserkennung. In der Software werden die beiden Eingänge des Tasters über einen Äquivalenzblock verknüpft. Dadurch lässt sich ein Zeitverzug beim Einschalten erkennen. Das Signal des Äquivalenzblocks wird an eine vorgefertigte Klasse für einen NOT-Aus Taster weitergeführt. Dieser

Funktionsblock bietet die Möglichkeit eines manuellen Zurücksetzens über einen externen Eingang. Der dritte unsichere Eingang dient dabei als Reset-Schalter.

Eingang Nummer fünf ist an das BMS angeschlossen, der unsichere Eingang Nummer zwei stellt die Verbindung mit der „EnableDrive“ Klasse der Steuerung dar. Über diese beiden Eingänge wird erkannt, ob ein Fehler im Kart oder in den Batterien vorliegt. Sind beide Signale auf High, kann problemlos gefahren werden. Da hier ein unsicherer Eingang verwendet wird, muss ein spezielles UND-Gatter, welches einen sicheren mit einem unsicheren Eingang verknüpft, verwendet werden. Der Eingang drei ist mit dem ersten Schlüsselschalter verbunden, hier ist gleich wie beim NOT-Aus Taster die Querschlusserkennung aktiviert. Die Signale des NOT-Aus Tasters, des BMS, der unsichere Eingang für die Fehlererkennung sowie des ersten Schlüsselschalters werden über ein UND-Gatter verknüpft und ergeben in Summe, ob der Motorcontroller eingeschaltet werden soll. Dabei passiert das Einschalten über den sicheren digitalen Ausgang Nummer Eins, welcher über einen Einschalthebel mit dem Steuerrelais verbunden ist.

Über den sicheren Eingang sechs wird ermittelt, ob das Steuerrelais für den Controller eingeschaltet ist und korrekt funktioniert. Sobald das Steuerrelais angezogen hat, soll auch das Hauptrelais freigegeben werden. Dazu wird der sichere Ausgang zwei auf High geschaltet. Dieser Ausgang ist auf einen Optokoppler verknüpft, welcher direkt in Serie mit der Spule des Hauptrelais geschaltet ist. Die grundsätzliche Ansteuerung des Hauptrelais soll dem Controller überlassen werden, da sich dies weitaus unproblematischer gestaltet. Das Hauptrelais soll nur in Extremsituationen vom Safety-Modul deaktiviert werden. Kritisch wird es beim Abschalten: Sollte das Steuerrelais noch aktiviert, aber das Hauptrelais bereits ausgeschaltet sein, kann es durch Spannungsrückwirkungen aus dem Motor zum Zerstören der Leistungselektronik des Controllers kommen. Um einen solchen Schaden zu verhindern, wird der Ausgang des Hauptrelais über eine Ausschaltverzögerung geschaltet. Das bedeutet, dass trotz Ausschaltens des Steuerrelais das Hauptrelais noch zwei Sekunden aktiv bleiben kann. Da bereits die Steuerspannung des Controllers abgeschaltet ist, findet kein Energiefluss zum Motor mehr statt, der Motor nicht mehr weiter angetrieben werden. Dennoch hat der Motorcontroller genügend Zeit, die Leistungselektronik abzuschalten, um so Schäden durch Spannungsrückwirkungen zu verhindern.

Der digitale Eingang Nummer Vier ist mit dem zweiten Schlüsselschalter verbunden. Dieser sichere Eingang wird zu einer unsicheren Variable konvertiert und anschließend mit dem unsicheren Ausgang zwei verbunden. Dieser Ausgang wird anschließend vom normalen Programm eingelesen und von der Klasse „DriveKart“ verarbeitet. Der Klasse „DriveKart“ des nichts sicherheitsgerichteten Programms muss auch mitgeteilt werden, ob der Controller aktiv ist. Dies wird ebenfalls über einen „unsafe Output“ realisiert. Jedoch muss beachtet werden, dass der Controller eine gewisse Zeit zum Einschalten und Hochfahren benötigt. Um diese Zeit zu berücksichtigen, wird eine Einschaltverzögerung von zwei Sekunden dazu geschaltet.

## 6.3 Programm des Motorcontrollers

Da der Motorcontroller eine eigene Steuerung besitzt, welche unabhängig von der SPS arbeitet, muss er eigens programmiert werden. Der Hersteller Sevcon liefert dafür ein eigenes Programm namens „DVT“ (siehe Abbildung 6.7). Für dieses Programm ist eine eigene Lizenz erforderlich, welche separat beim Kauf des Controllers erworben wurde.

Der Motorcontroller wurde bereits vom Hersteller vorab programmiert. Dadurch ist die Regelung des Motors bereits fertig implementiert, es müssen vom User nur noch die zur Anwendung passenden Parameter eingestellt werden. Für jeden Parameter ist ein eigenes Register vorhanden, welches in DVT direkt über die jeweilige Hex-Adresse gesucht werden kann.

Programmiert wird der Controller über den CAN-Bus. Hierfür ist ein CAN- Adapter für den PC erforderlich. Verwendet wird der Ixxat CAN- USB Adapter, welcher auf einer Seite per USB an den PC und auf der anderen Seite auf drei Pins an den Controller angeschlossen wird.

### 6.3.1 Übersicht über das Programm

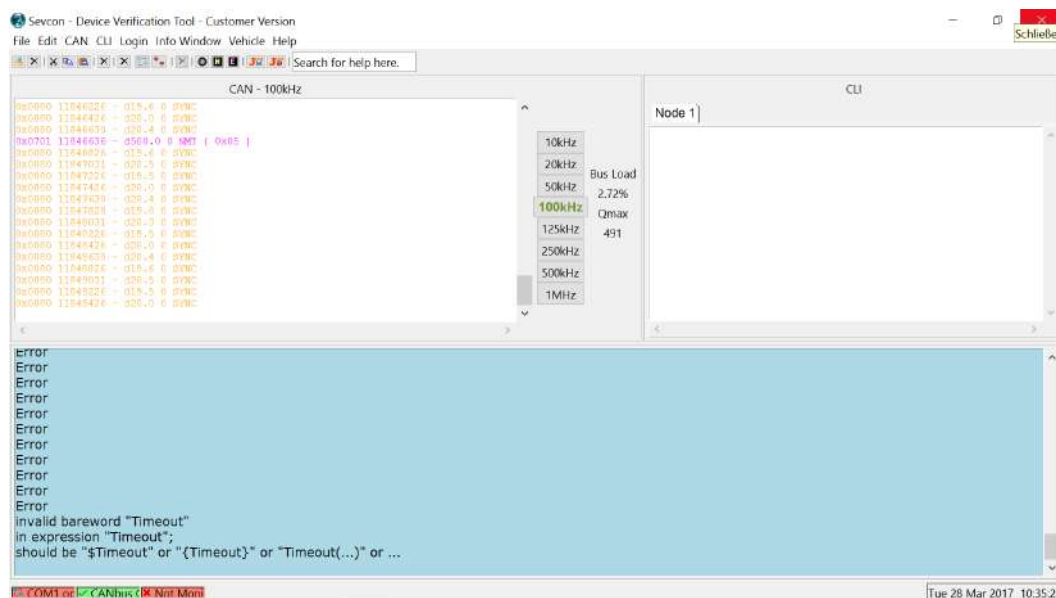


Abbildung 6.7: Übersicht DVT

Im Kasten links oben ist das CAN-Fenster, in dem die gesamte Kommunikation zwischen PC und Controller angezeigt wird. In der Spalte rechts daneben lässt sich die Baud-Rate einstellen. Beim CAN-Bus müssen die Baudraten aller angeschlossenen Komponenten übereinstimmen, da dies sonst in Kommunikationsproblemen ausarten würde.

Im unteren Rechteck befindet sich die Konsole, in der sowohl Befehle eingegeben werden können als auch Nachrichten vom Programm angezeigt werden. Im Rechteck rechts oben befindet sich das CLI-Fenster, welches aber bei der Programmierung keine Verwendung findet.

Die wichtigsten Funktionen des Programms können über die Schaltflächen in der Menüleiste aufgerufen werden. Diese sind:

### **Vehicle Interface**

Über diesen Knopf kann eine Statusanzeige des angeschlossenen Controllers aufgerufen werden. Hier bietet sich die Möglichkeit, alle vom Motorcontroller erfassten Daten anzuzeigen, wie zum Beispiel Temperatur oder Drehzahl. Über die CAN-Einstellungen kann festgelegt werden, welche Daten der Controller übermitteln soll. Die Anleitung zur Einstellung soll später erfolgen. Ein großer Vorteil dieser Funktion ist, dass eine Datenaufzeichnung ermöglicht wird. So lassen sich einzelne Werte wie z.B. die Temperatur über einen längeren Zeitraum betrachten.

### **Helper**

Das ist die Hauptfunktion des Programms. Hier können alle Einstellungen getätigt werden. Allerdings funktioniert dieser Helper nur, wenn der Controller angeschlossen ist. Im Helper sind die wichtigsten Register mit Makros hinterlegt. Auch werden hier Hilfestellungen für die Einstellungen wichtiger Werte z.B. Parametrieren des Encoders gegeben.

### **Editor**

Über den Editor können alle Werte auch offline bearbeitet werden. Da hier kaum Makros hinterlegt sind, müssen die Register direkt gesucht und geändert werden.

### 6.3.2 Programmierung mit dem Helper

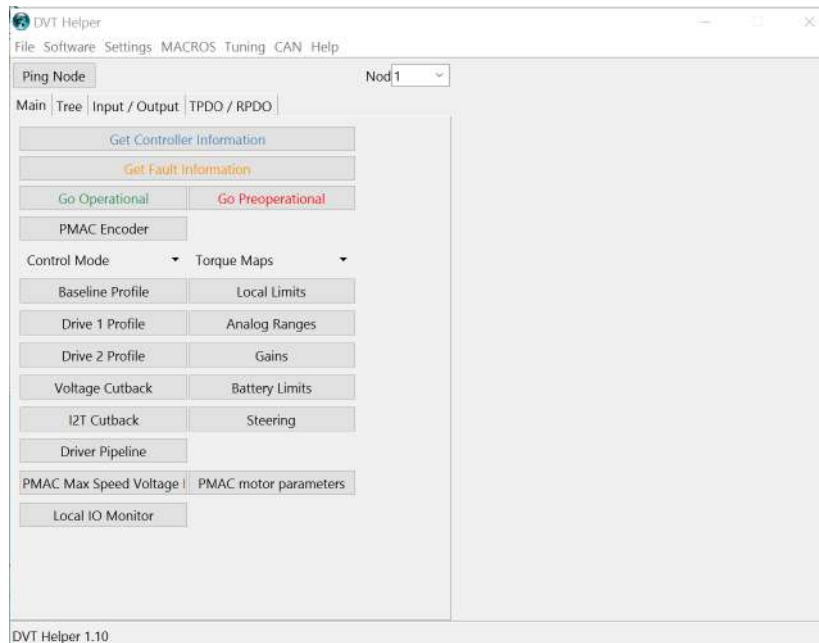


Abbildung 6.8: DVT-Helper

Auf der ersten Seite befinden sich Makros für die wichtigsten Funktionen und Einstellungen. Über den Button „Get Controller Information“ kann ein Fenster aufgerufen werden, wo Seriennummer, Hardware und Software-Version des Controllers angezeigt werden. Auch ersichtlich sind hier Temperatur und Batteriespannung. Der darunterliegende Knopf „Get Fault Information“ öffnet ein Fenster, welches im Fehlerfall die Nummern und den Namen der vorliegenden Fehler anzeigt. Sollten keine Fehler vorliegen, ist der Controller fahrbereit.

Die beiden nebeneinanderliegenden Buttons „Go Operational“ und „Go Preoperational“ setzen den Controller entweder in den Fahrmodus, oder in den Programmiermodus. Zu beachten ist, dass einige Werte nur dann änderbar sind, wenn sich der Controller im Preoperational Mode (PREOP) befindet. Ob eine korrekte Umschaltung des Modus passiert ist, wird im Konsolen-Fenster im Hauptprogramm angezeigt.

Baseline	Driveab. 1	Driveab. 2	
Maximum Torque	100%	60%	100%
Brake - Torque	20%	20%	20%
Maximum Speed Forward	3800 rpm	1000 rpm	6000 rpm
Maximum Speed Backward	1500 rpm	250 rpm	2000 rpm
Ramp up Rate	100%/s	100%/s	250%

Tabelle 6.1: Fahrprofile

Über die Schaltflächen „Baseline Profile“, „Driveability Select 1“ und „Driveability Select 2“ können die unterschiedlichen Fahrmodi parametrisiert werden. Dabei ist „Baseline Profile“ das Grundprofil, welches über den beiden anderen Fahrprofilen steht. Das erste Fahrprofil wird für den gedrosselten Modus verwendet, das zweite für den Renn-Modus. In unserem Fall wird das „Baseline“-Profil mit den maximal fahrbaren Werten gefüllt. Bei den beiden „Driveability“-Profilen werden lediglich das Drehmoment, die Endgeschwindigkeit und die Beschleunigung heruntergesetzt.

Mittels „Local Limits“ können ähnliche Einstellungen für den Motor vorgenommen werden. So ist es möglich, den maximal erlaubten Strom oder die Drehzahl zu begrenzen. Auch hier wird die maximal erlaubte Drehzahl noch einmal auf 6000 rpm gesetzt. Das Maximaldrehmoment beträgt 85 Nm, der Effektivstrom soll 200 Ampere nicht übersteigen.

Unter „Gains“ sind die einzelnen Beschleunigungswerte zu finden. Auch kann hier vorgegeben werden, wie viel das Fahrzeug ohne Gas bremsen soll. Da wir auch eine Rekuperation implementieren, sollen auch die Bremswerte bestimmt werden. Durch die Schaltflächen „Voltage Cutback“ und „Battery Limits“ kann festgelegt werden, ab welchen Batteriespannungen die Motorleistung zurückgeregelt oder ganz abgeschaltet wird. Die Spannung der Batterien sinkt mit der entnommenen Ladung, dadurch müssen gewisse Toleranzen berücksichtigt werden. Grundsätzlich gilt: Batteriespannung  $\pm 20$  Volt für ein Vermindern der Antriebsleistung,  $\pm 30$  Volt für eine komplette Abschaltung.

Über „Analog Ranges“ kann eingestellt werden, welcher Spannungsbereich für die Analogen Eingänge (=Gaspedal) zulässig ist. Da das Analogsignal durch ein analoges Ausgangsmodul AO081 erzeugt wird, können Werte von 0 bis +10 Volt erreicht werden.

Die Schaltfläche „Steering“ ist für eine elektrische Lenkunterstützung (Servolenkung) vorgesehen. Über „I2T“ kann ein Temperaturregler eingestellt werden, welcher die Motorleistung passend herunterregelt. Beide Funktionen werden jedoch in diesem Projekt nicht gebraucht.



Auch zur Überwachung des Antriebssystems sind zwei Funktionen implementiert. Unter „Driver Pipeline“ sind aktuelle Werte von anliegendem Drehmoment und Gaspedal ersichtlich. Über den „Local IO Monitor“ können live die Zustände der analogen und digitalen Eingänge eingesehen werden. Das ist besonders bei Tests wichtig, da hier erkannt werden kann, ob die digitale Verbindung zwischen Controller und SPS einwandfrei funktioniert.

Für die Einstellung der Motorkenndaten sind drei Schaltflächen vorgesehen. In „PMAC Encoder“ kann ein Offset für den Sinus/Cosinus Encoder definiert werden. Über „PMAC Max Speed Voltage“ ist ersichtlich, ob die maximale Motordrehzahl mit der eingestellten Drehzahl möglich ist. Durch „PMAC motor parameters“ können schließlich Werte wie Induktivität oder Widerstand bestimmt werden.

Diese Motorwerte sind entweder vom Hersteller angegeben, oder mussten eigens nachgemessen werden. Im der folgenden Abbildung sind alle eingestellten Motorparameter ersichtlich.

<b>PMAC motor parameters 0x4641</b>		
Maximum Stator Current (Is_max)	200.0	A(RMS)
Minimum Magnetizing Current (Im_min)	-250.0	A(RMS)
Number of Pole Pairs (np)	5.0	
Stator Inductance (Ls)	99.95698928833004	uH
Voltage Constant (Ke)	0.159210205078125	V/rads (line, rms, elec_freq)
Max drive mod index	89.990234375	%
Iq max modindex (PI-Mode IQM)	0.9765625	
Percentage minimum allowed saturation of Ls	39.9993896484375	%
Current control proportional gain (Kp)	4.0	
Current control integral gain (Ki)	1.0	
D-axis current controller proportional gain	1.0	
D-axis current controller integral gain	0.0	
Frequency/Mod index control Kp	0.001556396484375	
Frequency/Mod index control Ki	0.000640869140625	

Abbildung 6.9: Motorparameter

Unter dem Reiter „Tree“ können weitere Einstellungsmöglichkeiten gefunden werden. Auch kann hier über die Suche jedes Register direkt gefunden und beschrieben werden.

Besonders wichtig sind hier die Einstellungen für das Hauptrelais. Es können die Spannungen zum Anziehen und Halten definiert werden. Ein Relais benötigt beim Anziehen zum Aufbau vom Magnetfeld einen großen Strom. Sobald das Magnetfeld aufgebaut ist, kann der Strom und damit auch die Spannung verringert werden. Manche Relais haben bereits eine externe Schaltung für genau diesen Zweck eingebaut. Da das von uns verwendete Relais jedoch keine solche Schaltung besitzt, wird die Funktion des Controllers genutzt.



### 6.3.2.1 Input/Output

Hier zu finden sind alle Ein- und Ausgänge des Controllers. Dabei können sowohl die Anzahl der aktiven Ein- und Ausgänge als auch die Funktionen eingestellt werden.

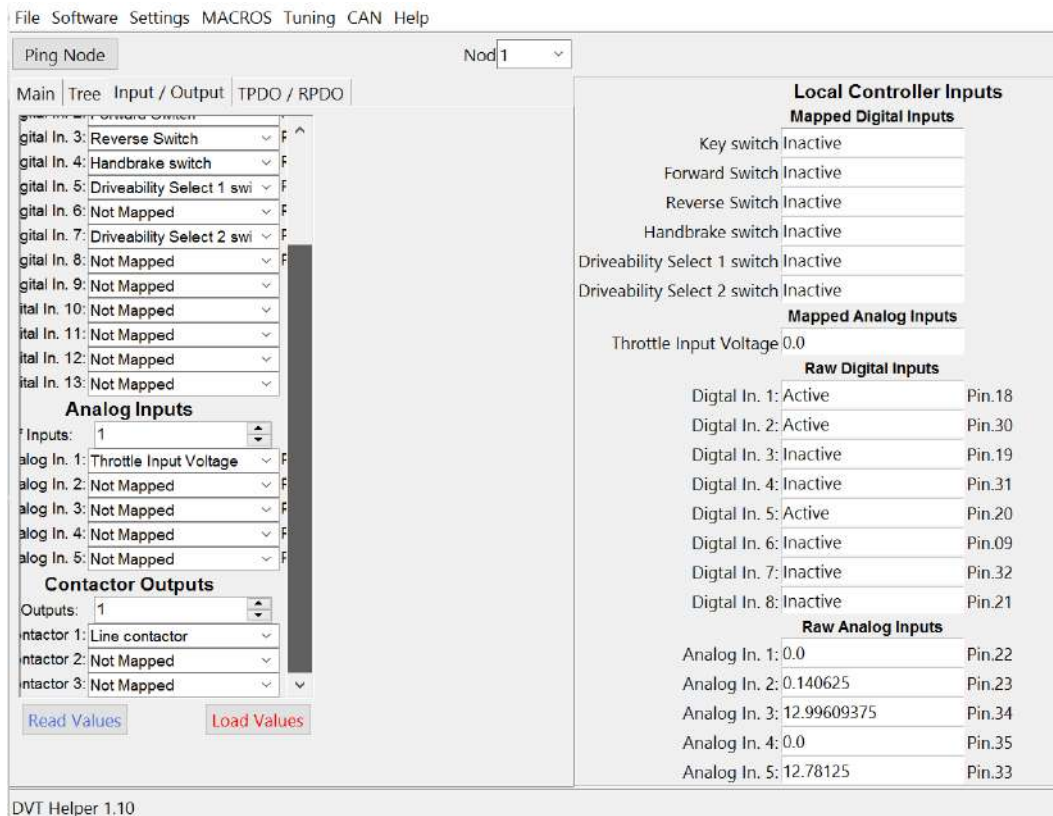


Abbildung 6.10: Verknüpfte IOs mit IOMonitor

### 6.3.2.2 TPDO/RPDO

Über den Reiter TPDO RPDO können die CAN-Nachrichten des Controllers eingestellt werden. Dabei sind TPDO (throw) die gesendeten und RPDO (read) die empfangenen Nachrichten. Diese Nachrichten können mit Objekten verknüpft werden, jede Nachricht umfasst bis zu 64 bits an Daten. In diesem Fall wird nur die erste Nachricht genutzt. Es werden die Temperatur und die Motordrehzahl übertragen.

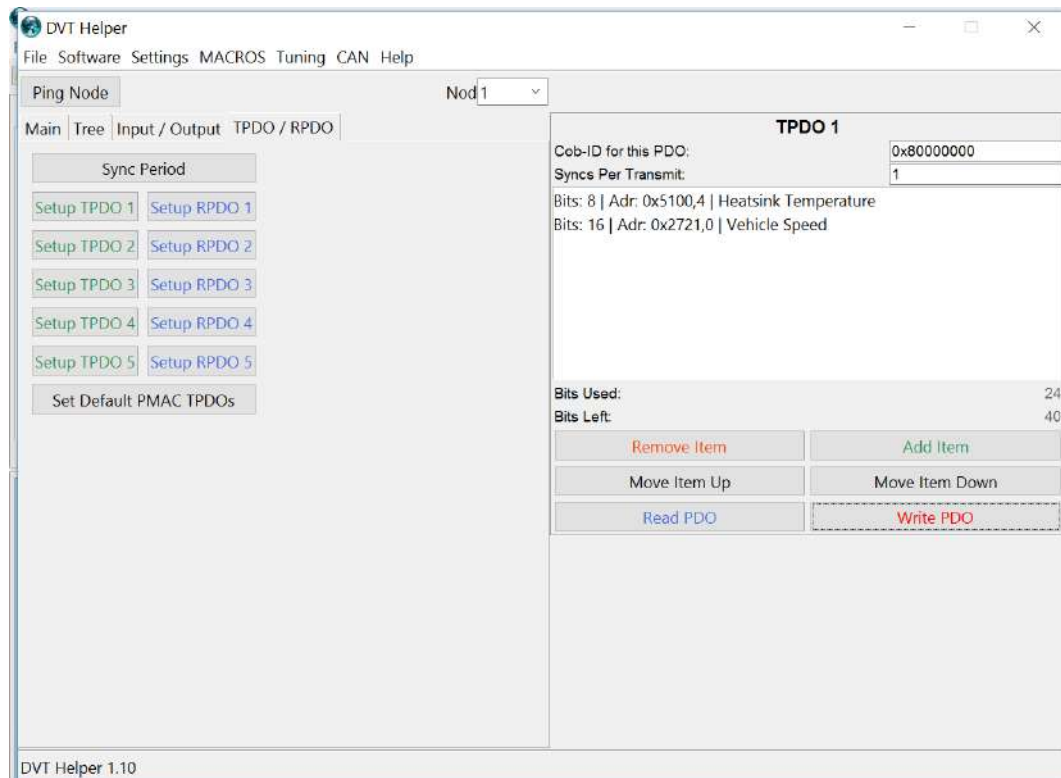


Abbildung 6.11: TPDO/RPDO

Über den Reiter „CAN“ kann die Baudrate des CAN-Busses verändert werden. Um mit der SPS und dem Display übereinzustimmen, wurde eine Baudrate von 100 kHz ausgewählt.

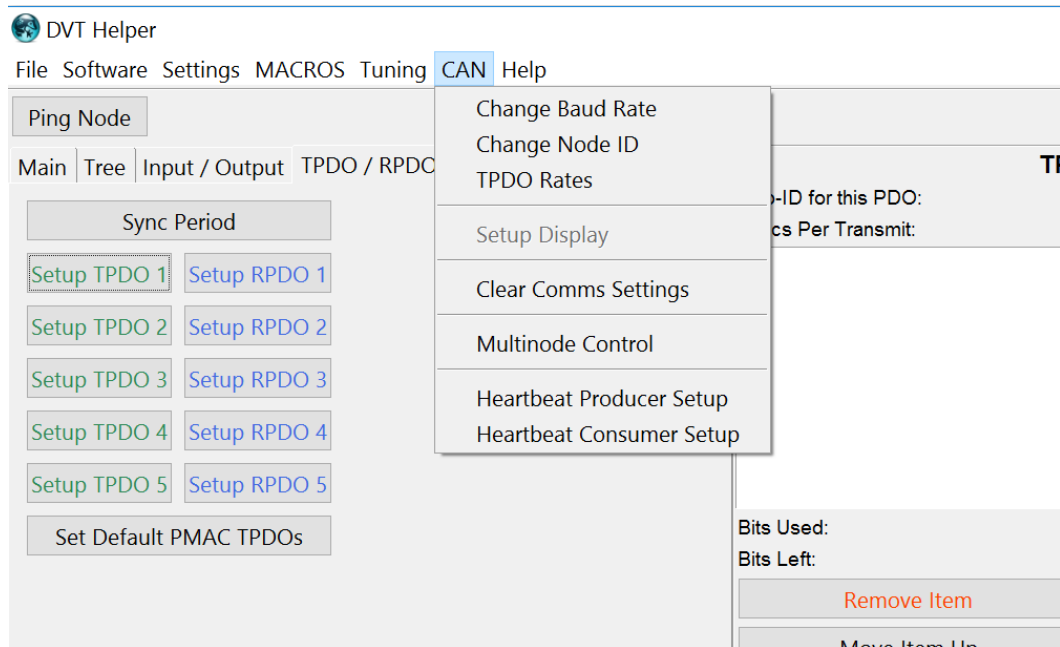


Abbildung 6.12: CAN - Einstellungen

### 6.3.2.3 Download des Programms

Sobald alle Werte eingegeben wurden, kann das Programm über Settings>Save DCF gespeichert werden. Sollte ein Parameter den erlaubten Bereich unter- oder überschreiten, wird dies über einen Fehler namens „Param Dyn Range Error“ mitgeteilt. Sobald das DCF-File gespeichert wurde, kann es über Settings>Send DCF to Unit auf den Controller geschrieben werden. Im Anschluss muss der Button „Go Operational“ angeklickt werden, wodurch der Controller in den Fahrmodus schaltet. Das Kart ist nun einsatzbereit.

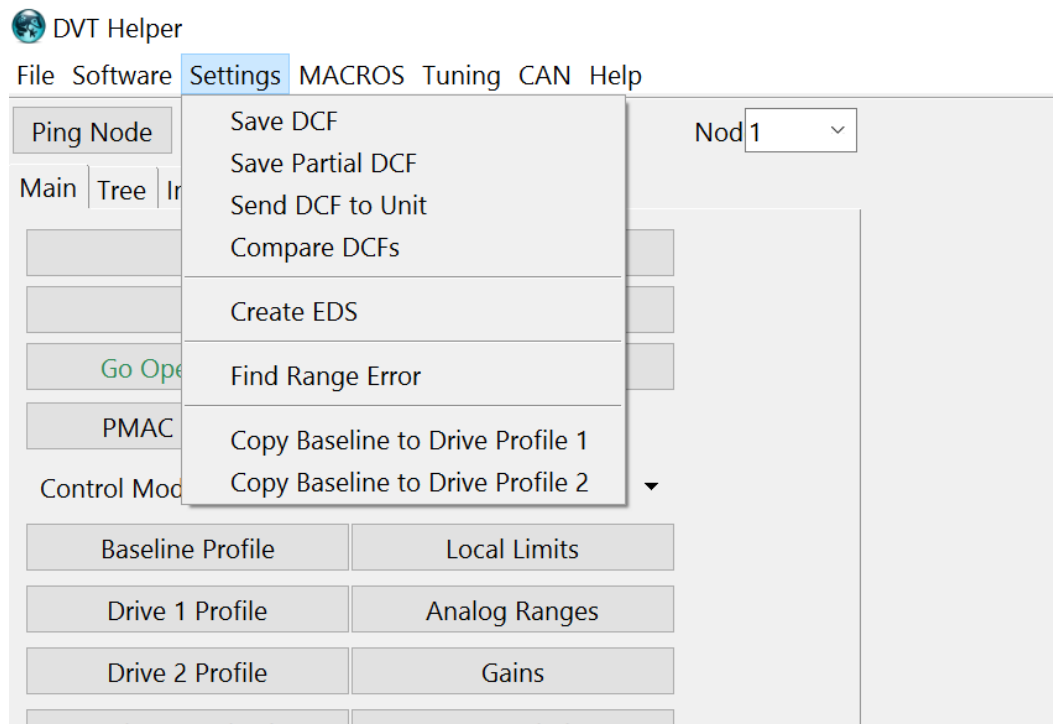


Abbildung 6.13: Upload der Software auf den Controller

## 6.4 Visualisierung

### 6.4.1 Allgemeines

Um die technischen Features darstellen zu können, wurde die Entscheidung getroffen, diverse Daten über ein Display zu visualisieren. Auf dieser Visualisierung sollen neben den wichtigen Fahrdaten auch diverse Status und Fehler sichtbar sein, um das Kart bereits vor dem Fahrbeginn vollständig elektronisch überprüfen zu können.

#### 6.4.1.1 Auswahl des Displays

Um eine Visualisierung verschiedenster Daten, wie zum Beispiel die aktuelle Fahrgeschwindigkeit oder diverse Fehleranzeigen zu ermöglichen und andererseits eine visuelle Umschaltung der Fahrmodi zu gewährleisten, wurde ein Touch-Panel benötigt. Dieses sollte in das Lenkrad des Karts eingebaut werden und musste dementsprechend klein sein. Deshalb wurde das Bedienpanel ETT352 von der Firma Sigmatek ausgewählt, da die SPS, mit der in dem Projekt gearbeitet wird, ebenfalls von der Firma Sigmatek zur Verfügung gestellt wurde. Durch die Auswahl aus dem Produktportfolio ist ein einfacher Datenaustausch der beiden Komponenten möglich.



Abbildung 6.14: DisplayETT352

#### **6.4.1.2 Wichtige Daten des Displays**

Das Bedienpanel ist 3,5“ groß und verfügt über eine Auflösung von 320\*240 Pixel, hat also das Verhältnis von 4:3. Außerdem ist ein Ruhezustand des Displays möglich, welcher sich selbst nach einer gewissen Zeit aktiviert. Der Datenaustausch zwischen SPS und Display erfolgt über einen CAN-Bus.

#### **6.4.2 Verbindungseinstellungen**

Da mit der SPS über CAN-Bus kommuniziert wird, mussten einige Voreinstellungen getroffen werden, um die programmierte Visualisierung auf das Display hinaufladen zu können. Grundsätzlich funktioniert die Kommunikation über CAN-Bus so, dass die Visualisierung auf die CPU der SPS geladen wird und von dort eine Kommunikation zum Display stattfindet, in der das Projekt zum Display weitergeladen und gespeichert wird<sup>16</sup>.

##### **6.4.2.1 Herstellung der Verbindung zur CPU**

Um die Verbindung von der CPU zum Display herstellen zu können, muss zu allererst die Verbindung zur CPU erstellt werden. Dazu wurde eine neue Verbindung unter dem Reiter *Change Online Settings* im Lasal Class erstellt.

---

<sup>16</sup>vgl. 3,5Zoll Terminal Projektrealisierung, <https://goo.gl/nx2F04>, (Stand: 23.03.2017)

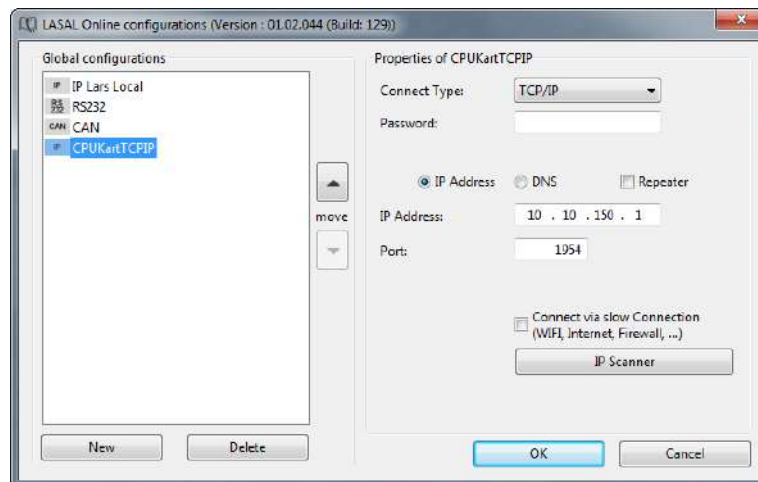


Abbildung 6.15: Online Configurations

Als *Connect Type* muss *TCP/IP* ausgewählt werden, da die Verbindung über ein Netzkabel hergestellt wird. Die IP Adresse der SPS ist nach der Lieferung standardmäßig 10.10.150.1, diese Adresse und der Port wurden auch im Laufe des Projektes nicht verändert. Nach diesen Einstellungen kann über den Reiter *Go Online* auf die SPS *online gegangen werden* und es können Programme, die im Lasal Class 2 programmiert wurden, über *Download* auf die SPS geladen werden.

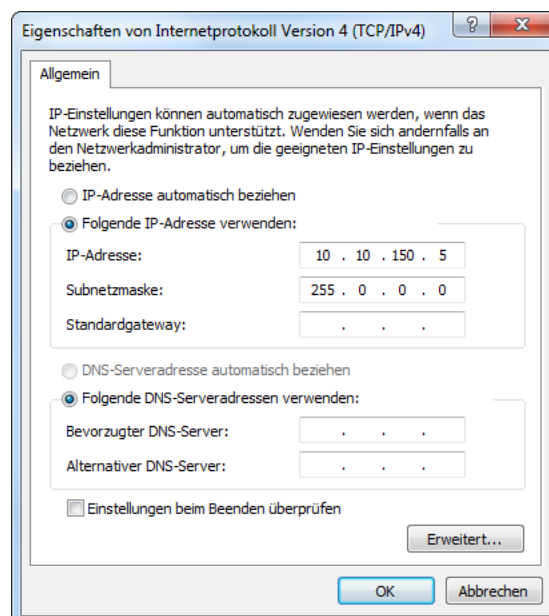


Abbildung 6.16: Neue IPv4 Adresse

Außerdem wurde dem PC, mit dem die Verbindung hergestellt wurde, eine manuelle *TCP/IPv4 Adresse* zugewiesen. Dies war notwendig, um die Kommunikation erfolgreich aufzubauen. Als IP Adresse wird 10.10.150.5 verwendet und die Subnetzmaske lautet 255.0.0.0.

#### 6.4.2.2 Einstellung auf der CPU und am Display

CPU:

Im *Online-Modus* kann im Lasal Class 2 über *Debug/File Transfer/Edit Autoexec.lsl...* auf die Speicherkarte der Steuerung zugegriffen werden. Dabei wurden folgende Werte eingestellt (siehe Abbildung 6.17):

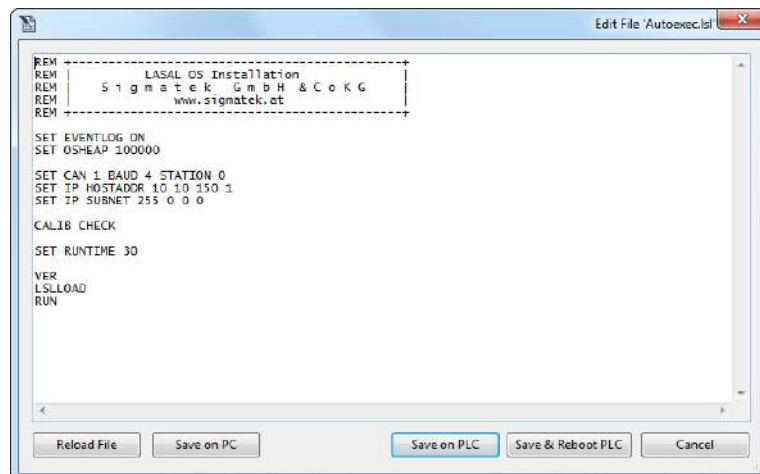


Abbildung 6.17: Autoexec

Der Wert von *CAN* wurde auf 1 gesetzt, der von *BAUD* auf 4 und der von *STATION* auf 0. Dabei beschreibt der Wert nach *CAN*, dass eine Übertragung über CAN-Bus Netzwerk möglich wird. Die Zahl nach *BAUD* legt die Baudrate, also die Übertragungsgeschwindigkeit fest. Dabei wurde sich für die höchstmögliche Kommunikationsgeschwindigkeit entschieden. Diese liegt bei 100 kHz. Der Wert nach *STATION* legt die Stationsnummer von der SPS fest, welche später noch benötigt wird. In dieser Datei befindet sich der gespeicherte Text der Speicherkarte, welche sich in der SPS befindet. Dort könnte zum Beispiel auch die IP Adresse verändert werden, was im Verlauf des Projektes aber nicht benötigt wurde.

Display:

Nach dem Einschalten der Spannungsversorgung kann am Display in die rechte untere Ecke auf den *Setup Button* gedrückt werden. Dabei wird ein *Einstellungsscreen* aufgerufen, in dem die *CanNode* auf 2 gesetzt werden muss. Der Wert von *CanBaud* muss mit dem Wert auf der Speicherkarte übereinstimmen und wird deshalb auf 4 erhöht. Die Hintergrundbeleuchtung wurde nicht verändert. Dieser Screen erscheint auch, wenn der *Setup-Button* nicht sichtbar ist. Dieser *Setup-Button* ist nicht sichtbar, wenn ein Initialisierungsbild vergeben wird.

#### 6.4.2.3 Einstellungen im Lasal Class 2

Im Lasal Screen wurde ein neues Netzwerk erstellt, in das folgende *Klassen (Programmierobjekte)* platziert wurden.



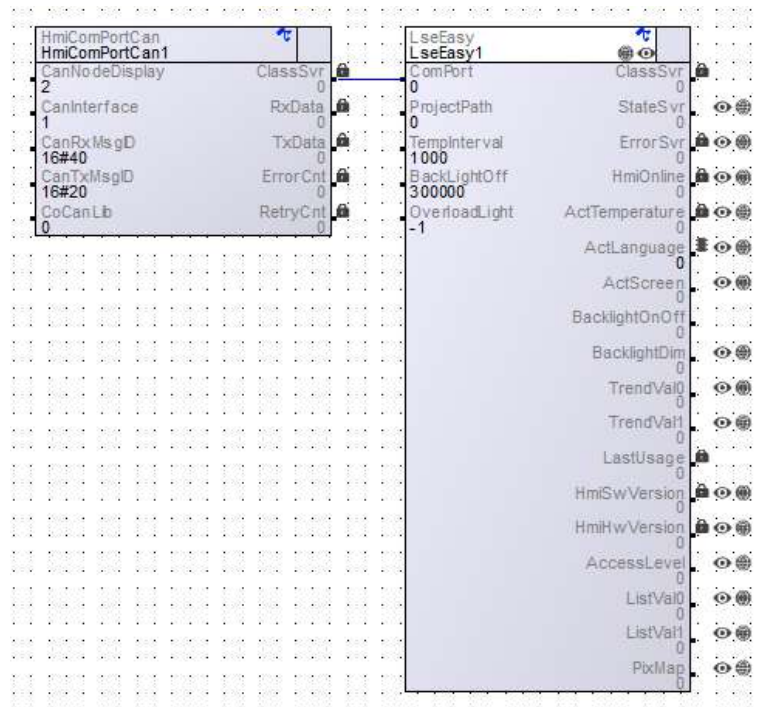


Abbildung 6.18: Platzierte Klassen für die Verbindung

In der Klasse *HmiComPortCan* wurde der Wert von *CanInterface* auf 1 gesetzt, da für die Visualisierung der erste Can-Anschluss verwendet wurde. Danach wurde der Wert vom *CanNodeDisplay* eingetragen. Dieser muss den selben Wert betragen, wie zuvor am Display eingestellt. In der Klasse *LseEasy* wurde die Variable *ProjectPath* auf 0 gesetzt, weil nur ein Display verwendet wurde.

#### 6.4.2.4 Einstellungen im Lascal Screen

Da das Display über CAN-Bus kommuniziert, wurde beim Erstellen des Programms das *Template Easy320x240.lpr* ausgewählt. Anschließend wurden die *Project Setting* geöffnet. Dort wurde der *Target Path* geändert in *C:\MPC\*. Dies war nötig, da im LascalClass die Variable von *ProjectPath* auf 0 gesetzt wurde. Nach diesen Einstellungen konnte mit der Programmierung begonnen werden.

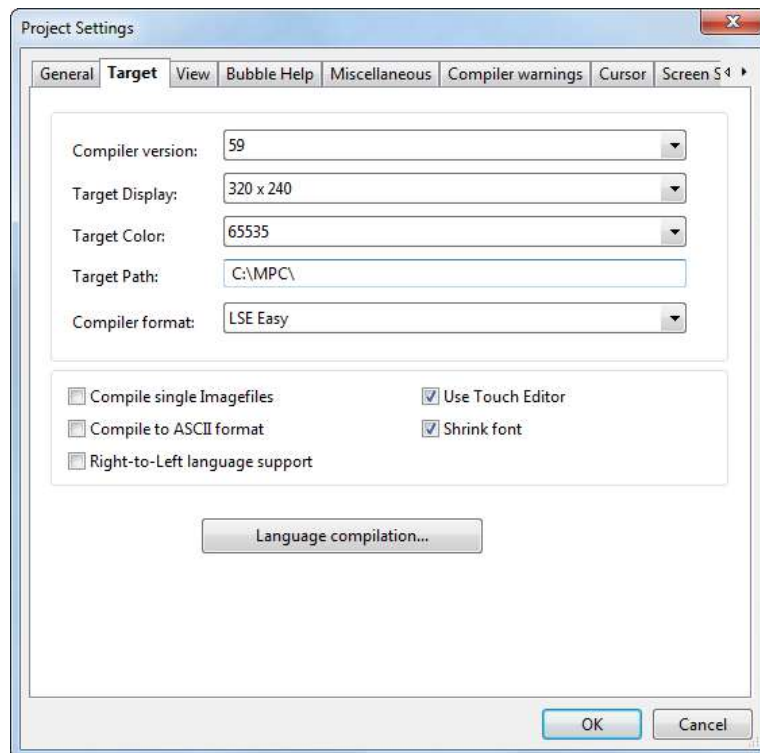


Abbildung 6.19: Project Settings

## 6.4.3 Screens

Als Screens bezeichnet man die einzelnen Anzeigeseiten, welche in der Visualisierung programmiert werden können. In diesem Teil werden nur der Aufbau der einzelnen Screens und deren Funktionen erläutert. Wie die einzelnen Komponenten genau platziert und konfiguriert werden, wird im Kapitel 6.4.5 näher beschrieben.

### 6.4.3.1 Screenstruktur

Generell wurde die Visualisierung so aufgebaut, dass jederzeit direkt oder über einen Zwischenbildschirm in jeden Screen gewechselt werden kann. Die Umschaltung der Screens erfolgt über die Softkeys, welche sich auf der Softkeyleiste im unteren Bereich des Bildschirms befinden. Jeder der Softkeys wurde so erstellt, dass der jeweilige Screenname im Softkey lesbar ist. Diese sind somit selbsterklärend. Nach dem Einschalten wird immer der Startscreen angezeigt. Nachdem dieser umgeschaltet wird (Schlüsselschalter wird dazu benötigt, siehe 6.4.3.4.2) können die Softkeys immer zur Umschaltung verwendet werden.

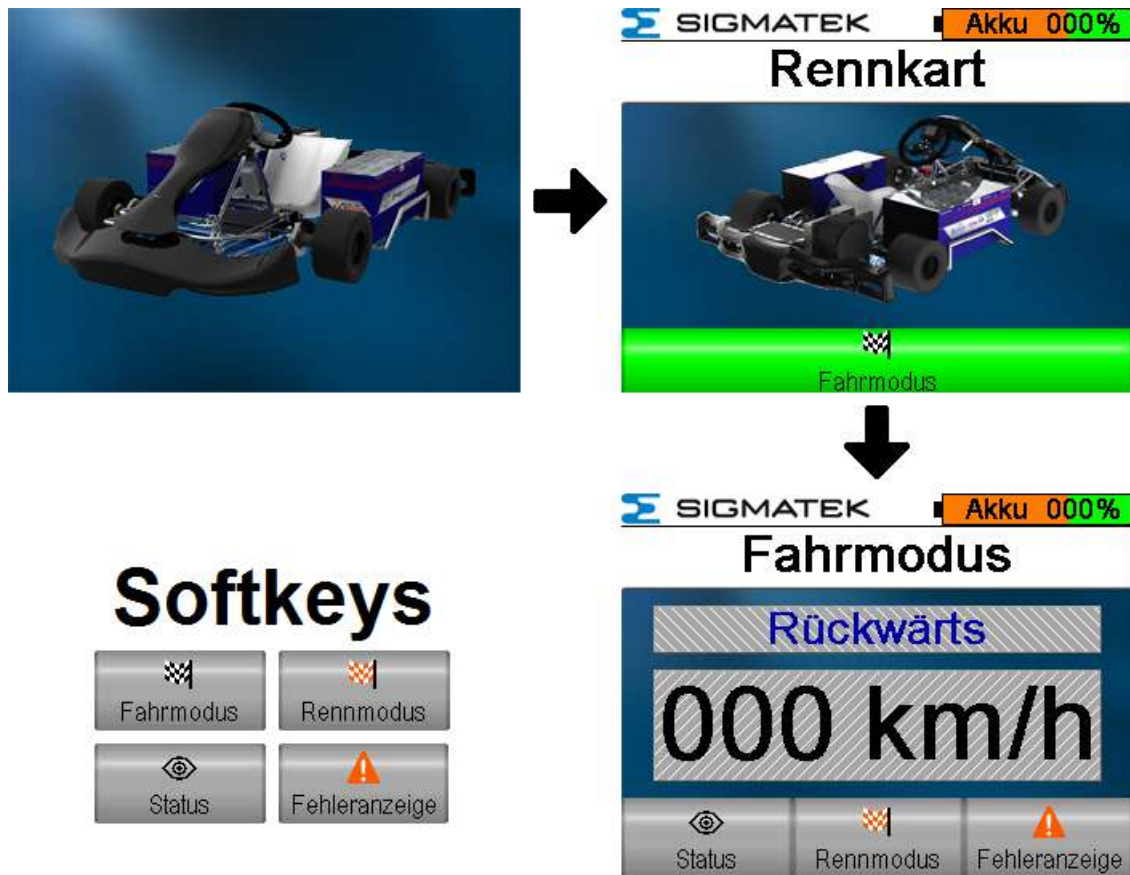


Abbildung 6.20: Screenstruktur

#### 6.4.3.2 Initialisierung

Normalerweise würde bei der Initialisierung des Displays eine Anzeige der Einstellungen erscheinen, welche die Werte der dafür wichtigen Variablen wie die Can Node Nummer und die Baudrate beinhaltet. Da diese Anzeige nach erfolgreichem Test der Verbindung nicht mehr benötigt wird und nicht so gut mit der restlichen Visualisierung harmonisiert, wurde diese durch ein Initialisierungsbild ersetzt. Auf diesem Bild befindet sich das fertig gerenderte Kart aus der erstellten 3D Zeichnung. Dieses Bild wurde als *Image* importiert und *bootimage* genannt. Dieser Name kann auch umbenannt werden, indem man in der LSE-Easy Klasse den Text des *retcode* ändert.



Abbildung 6.21: Initialisierungsbild

```
FUNCTION VIRTUAL GLOBAL LseEasy::GetBootImage  
VAR_OUTPUT  
    retcode : ^CHAR;  
END_VAR  
  
    retcode := "bootimage"; // name of bootimage  
END_FUNCTION
```

Abbildung 6.22: Programmtext Bootimage

### 6.4.3.3 Global Screen

Bei den Screens muss noch zwischen den individuellen Screens und dem *Global Screen* unterschieden werden. Der *Global Screen* dient als Hintergrund der individuellen Screens. Wenn nötig könnte der Global Screen jedoch deaktiviert werden, was aber nicht der Fall war.

#### 6.4.3.3.1 Aufbau

Am Anfang wurde der dynamische Hintergrund festgelegt. Dieses Bild war schon vorgefertigt und wurde von der Firma Sigmatek zur Verfügung gestellt. Das Bild wurde unter *Image* importiert und dann platziert. Anschließend wurde der weiße Balken im oberen Bereich mithilfe eines *Rechteckes* erstellt. Dieses wurde weiß gefärbt und außerdem wurde diesem ein *3D Frame (Rahmen)* zugewiesen, den man am rechten unteren Bereich erkennen kann. Das Sigmatek Logo wurde ebenfalls als Bild importiert und in der linken oberen Ecke des Global Screens platziert. Danach wurde unter dem Logo eine Trennungslinie, welche sich über den ganzen Screen erstreckt, platziert. Zum Schluss wurde die „*Title Text*“ Überschrift und ein *Bar (Balken)* für die Visualisierung des Akkuzustandes eingefügt, welcher den Akkuzustand



Abbildung 6.23: Global Screen

visualisiert. Die Form des Akkus wurde mit einigen einzelnen *Lines* (*platzierte Linien*) nachgebessert und zum Schluss wurden der Text „Akku“ und das Prozentzeichen als Text hinzugefügt. Dazwischen wurde ein *Numedit* (*Nummerische Anzeige*) platziert, welches den prozentualen Akkuzustand darstellt.

#### 6.4.3.3.2 Funktionen

Da dies der *Global Screen* ist, gelten alle Funktionen davon für alle Screens, weil der *Global Screen* bei jedem einzelnen individuellen Screen als Hintergrund verwendet wird. Diese individuellen Screens sind somit übersichtlicher und besser zu erstellen, da nicht so viele Funktionen übereinander liegen. Der „Title Text“ nimmt bei jedem individuellen Screen den Namen an, welcher in den Einstellungen des Screens vergeben wird. Im Statusscreen lautet dieser zum Beispiel „Status“. Des Weiteren wird der platzierte Bar, der den Batteriestand visualisiert, von rechts nach links grün, sodass der Akkuzustand farblich sichtbar wird. Als Hintergrundfarbe dazu wurde Orange gewählt, wie man in Abbildung 6.23 erkennen kann. Mit dem platzierten *Numedit* wird zusätzlich eine genauere Anzeige des Akkuzustandes in Prozent möglich. Dieses *Numedit* nimmt den Wert einer Übergabevariable an, welche einen Wert zwischen 0 und 100 annimmt.

#### 6.4.3.4 Startscreen

Direkt nach der Initialisierung wird automatisch in den Startscreen gewechselt.

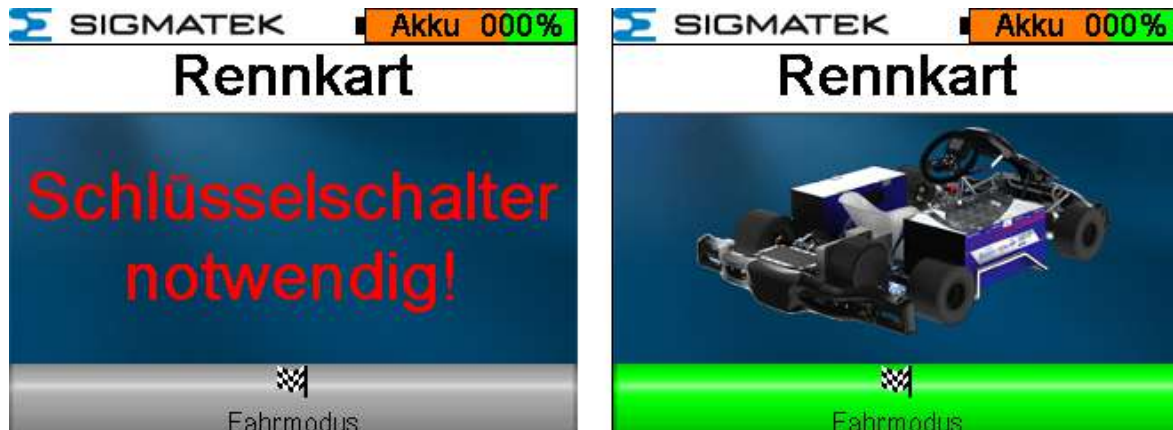


Abbildung 6.24: Startscreens

#### 6.4.3.4.1 Aufbau

Am Startscreen erkennt man im Hintergrund den *Global Screen*, welcher zuvor erstellt wurde. Zudem wurde der *Screename* „Rennkart“ vergeben, welcher oben mittig sichtbar ist. Der Text „Schlüsselschalter notwendig!“ wurde über eine *Textlist* erstellt und passend eingefügt. Im unteren Bereich wurde ein sogenannter *Softkey* erstellt. Dieser wurde mithilfe eines **Buttons** realisiert, in dessen Mitte das Bild mit der Flagge und dem Text Fahrmodus eingestellt wurde. Deckungsgleich mit dem Softkey wurde ein zweiter *Softkey* mit demselben Bild in der Mitte und einer grünen Hintergrundfarbe erstellt, dessen Funktion im Anschluss erläutert wird. Außerdem wurde noch ein Bild der 3D Zeichnung des Karts in der Mitte des Screens platziert.

#### 6.4.3.4.2 Funktionen

Da das Kart ohne Schlüsselschalter nicht betrieben werden darf, wurde auch die Visualisierung so programmiert, dass man nach dem Einschalten der SPS ohne Schlüsselschalter nicht in den Fahrmodus wechseln kann. Deshalb ist der Graue der beiden *Softkeys* nur eine Attrappe, welche keine Funktion beim Betätigen ausführt. Gleichzeitig wird auch der Text „Schlüsselschalter notwendig!“ angezeigt (siehe 6.24). Falls der erste Schlüsselschalter eingesteckt wird oder schon vorhanden ist, erscheint der grüne Softkey und das Bild mit dem Kart. Zeitgleich werden die zuvor sichtbaren Objekte inaktiv geschaltet und verschwinden. Der grüne Softkey kann betätigt werden, um in den Fahrmodus zu wechseln. Diese Funktionen wurden mithilfe von *Checkbits* und *Funktion Blocks* realisiert. Diese werden im Kapitel 6.4.5 erläutert.

#### 6.4.3.5 Fahrscreen

Der Fahrscreen wurde so programmiert, dass dies der einzige Screen ist, welcher der Variable *FlagDriveEnabel* den Wert 1 übergibt. Sobald diese Variable auf 1 ist,



kann losgefahren werden. Dadurch kann auch nur in diesem Screen losgefahren werden.

#### 6.4.3.5.1 Aufbau

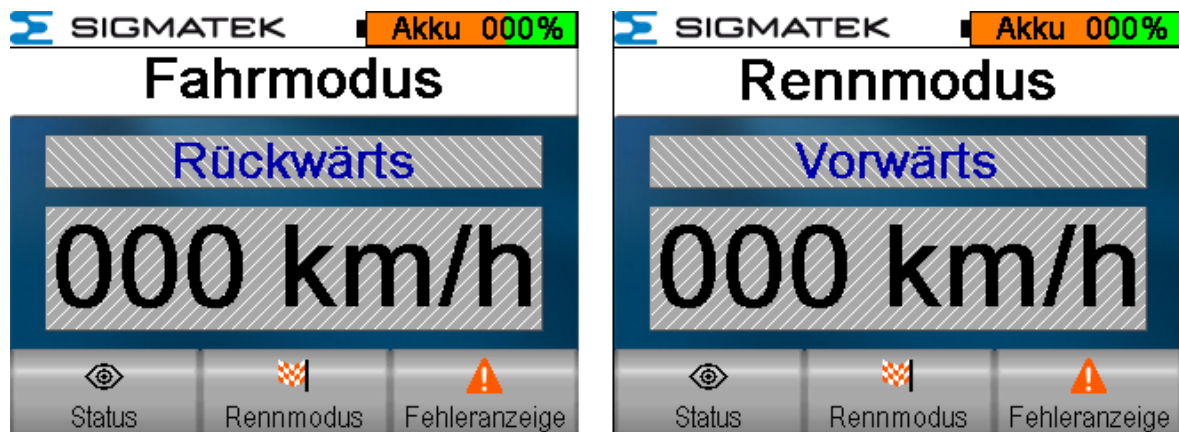


Abbildung 6.25: Fahrscreens

Für diesen Bildschirm wurde kein Name vergeben. Anstatt dessen wurden die Texte „Fahrmodus“ und „Rennmodus“ übereinander als *TextLists* eingefügt und mit den passenden *Checkbits* versehen. Anschließend wurden die beiden grau-weißen Rechtecke platziert, welche eine bessere Erkennung der Gänge und der Geschwindigkeiten ermöglichen sollen. Deckungsgleich mit dem oberen Rechteck wurden die Texte „Rückwärts“ und „Vorwärts“ erstellt und übereinandergelegt. Diese wurden ebenfalls mit *Checkbits* ausgestattet, sodass immer nur ein Text sichtbar ist, abhängig vom Wert der Variablen. Im unteren Rechteck wurde als erstes eine *TextList* mit dem Inhalt „km/h“ eingefügt und anschließend links daneben ein *NumEdit* erstellt. Diesem wurde die Variable der Geschwindigkeit zugewiesen.

#### 6.4.3.5.2 Funktionen

Da den Softkeys vom Rennmodus und vom Fahrmodus nicht nur die Funktion zugewiesen wurde, auf den Fahrscreen zu wechseln, sondern auch implementiert wurde, dass diese bei Betätigung, der Variable *VisuFlagDrivemode* den Wert 1 oder 0 zuweisen, kann im Fahrscreen der Fahrmodus und Rennmodus realisiert werden. Zusätzlich kann durch diese Variable im Lasal Class vom Rennmodus auf den Fahrmodus gewechselt werden, wenn zum Beispiel der zweite Schlüsselschalter abgezogen wird. Die *Checkbits* der Texte „Fahr- und Rennmodus“ wurden ebenfalls über die Variable *VisuFlagDrivemode* realisiert. Der Gang wird durch einen externen Schalter bestimmt. Dieser wird durch die SPS ausgewertet. Von der dazu verwendeten Variable (*VisuGear*) wurde ein *Checkbit* erstellt, welches die beiden Texte

abhängig vom Gang anzeigt. Das eingefügte *Numedit* stellt die aktuell gefahrene Geschwindigkeit dar.

#### 6.4.3.6 Statusscreen

Der Statusscreen dient dazu, die wichtigsten Funktionen im Stillstand des Go-Karts zu überprüfen und die Funktionen des Gaspedals und der Bremse zu testen.

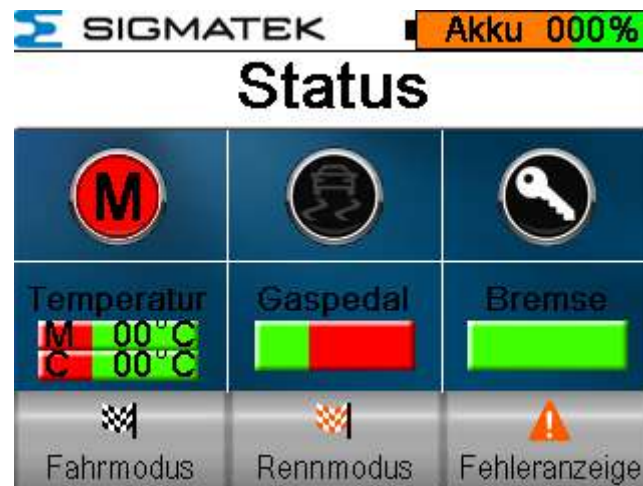


Abbildung 6.26: Statusscreen

##### 6.4.3.6.1 Aufbau

Als Name des Screens wurde „Status“ vergeben. Anschließend wurde die freie blaue Fläche mithilfe von weiß gefärbten Linien in sechs gleichgroße Einzelteile unterteilt. In den oberen drei Bereichen wurden runde Statusanzeigen erstellt, indem immer zwei Symbole übereinander gelegt wurden. Dabei beschreibt das linke Symbol den Motorcontroller, das mittlere Objekt die Antriebsschlupfregelung und mit dem Rechten wird angezeigt, ob der zweite Schlüssel eingesteckt ist. Im linken unteren Bereich wurden zwei *Bars* für die Visualisierung der Temperaturen von Motor und Controller platziert. Im mittleren Bereich wurde ebenfalls ein Balken erstellt, welcher den Zustand des Gaspedals darstellt. Im rechten unteren Bereich wurden zwei Rechtecke in den Farben rot und grün übereinandergelegt. Zudem wurden allen unteren Objekten 3D-Funktionen zugeteilt. Anschließend wurden die ganzen Texte im unteren Bereich erstellt. Dazu wurden noch zwei *Numedit*s platziert, auf denen die beiden Temperaturen des Motors und Controllers abgelesen werden können. Ganz unten befindet sich gleich wie in den anderen Screens die *Softkeyleiste*, die aus den drei einzelnen *Softkeys* mit deren eingefügten Bildern besteht.

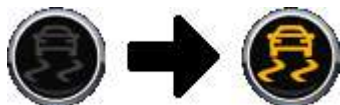


#### 6.4.3.6.2 Funktionen

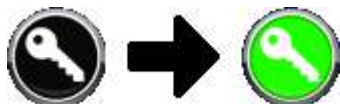
Die Funktionen der oberen drei Kontrolllampen wurden über *Checkbits* realisiert. Alle Bilder, welche im Anschluss vorkommen, wurden einzeln mithilfe von Bildbearbeitungsprogrammen erstellt, im Lasal Screen eingefügt und dort übereinandergelegt.



Der Hintergrund des Motorcontroller wechselt von Rot auf Grün, wenn dieser eingeschaltet wird.



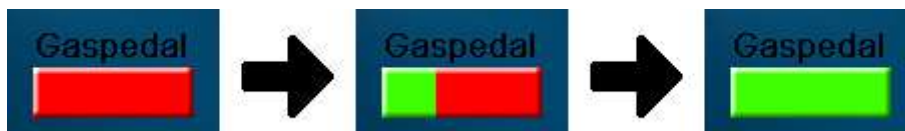
Im ausgeschalteten Zustand ist das Symbol in der Mitte grau gefärbt. Wenn die Antriebsschlupfregelung aktiviert ist, färbt sich das Symbol Gelb.



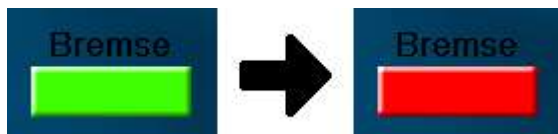
Die Anzeige des Schlüsselschalters wechselt von Schwarz, wenn der Schlüssel nicht vorhanden ist, auf Grün, wenn dieser eingesteckt und betätigt wird.



Mit dem Einschalten des Motorcontrollers färbt sich nicht nur das Statussymbol anders, sondern zusätzlich wird die Temperaturanzeige freigeschaltet, welche im ausgeschalteten Zustand nicht sichtbar ist. Diese Funktion wurde mithilfe von einem *Checkbit* erstellt.



Die Anzeige des Gaspedals dient zur Funktionsprüfung im Stillstand. Dabei färbt sich der Bar von links nach rechts Grün, je nachdem wie weit das Gaspedal betätigt wird.



Die Bremsanzeige dient ebenfalls zur Funktionsprüfung. Wenn die Bremse gedrückt wird, färbt sich das Rechteck von Grün auf Rot. Dieses Farbschema wurde deshalb ausgewählt, da das rote Rechteck mit dem Verkehrszeichen Stopp in Verbindung gebracht werden sollte. Die Funktion wird durch eine Verknüpfung mit einem *Checkbit* realisiert.

Mit den *Softkeys* im unteren Bereich kann auf den Fahrscreen mit Fahr- oder Rennfunktion gewechselt werden und zum Fehlerscreen gesprungen werden.

#### 6.4.3.7 Fehlerscreen

Der Fehlerscreen dient zur Überwachung aller Fehler und Alarmer die über die SPS festgestellt werden können. Anhand dieser Fehleranzeige ist es sehr einfach bei einem Fehler den Grund dafür herauszufinden, warum das Kart nicht fahrbereit ist. Es wurde eine graphische Methode zur Überwachung ausgewählt.

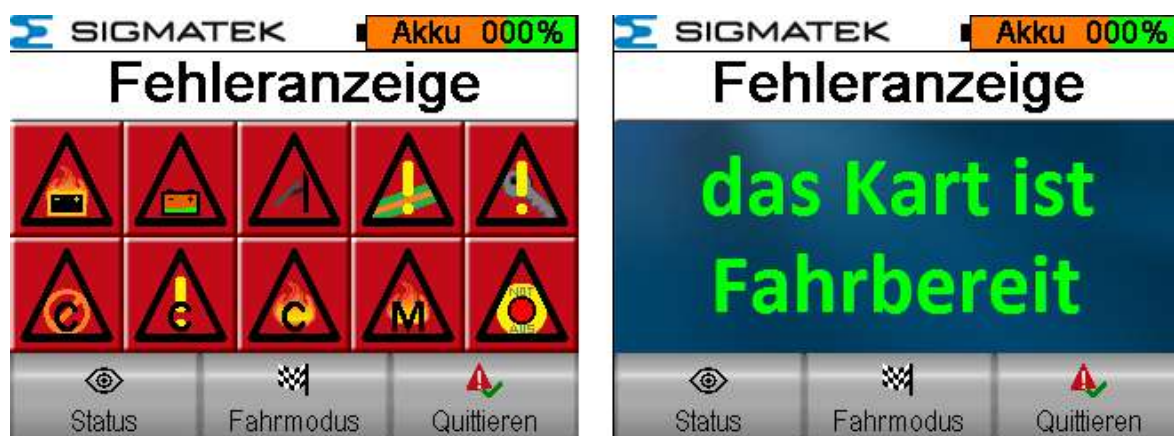


Abbildung 6.27: Fehleranzeigen

##### 6.4.3.7.1 Aufbau

Dem Fehlerscreen wurde der Name „Fehleranzeige“ vergeben. Dann wurden zehn Fehlersymbole mit einer Größe von je 64\*65 Pixel erstellt und platziert. Mittig zu diesen zehn Elementen wurde außerdem der Text „das Kart ist fahrbereit“ in grüner Farbe erstellt und über alle Symbole gelegt. Unten wurde die Softkeyleiste erstellt und mit passenden Bildern versehen.

#### 6.4.3.7.2 Funktionen

Die Anzeige funktioniert so, dass ein Symbol erscheint, wenn dieser Fehler vorliegt. Solang ein Fehler vorliegt und nicht behoben wird, kann auch nicht losgefahren werden. Im Anschluss werden die Fehler aufgelistet und beschrieben.



Dieses Symbol wird angezeigt, wenn die Zelltemperatur der Akkus zu heiß wird. Diese Temperaturgrenze liegt bei 45 Grad Celsius.



Dieses Symbol erscheint, wenn der Akkuzustand unter fünf Prozent sinkt, da ab diesem Zustand nicht mehr gefahren werden darf, da die Akkus auf keinen Fall tiefentladen werden dürfen.



Dieses Symbol leuchtet auf, wenn der Widerstandswert des Linearpotentiometers des Gaspedals außerhalb des Arbeitsbereiches und miteingerechneter Toleranz liegt. Dies kann zum Beispiel durch einen Kabelbruch passieren.



Dieses Symbol wird eingeblendet, wenn das Isolationsüberwachungssystem einen Fehler ausgibt. Vor allem nach einem Crash ist auf dieses Alarmsymbol zu achten, da eine beschädigte Isolation dramatische Auswirkungen hervorrufen könnte.



Dieses Symbol wird angezeigt, wenn der erste Schlüsselschalter nach dem Einschalten und Wechseln in einen anderen Screen entfernt wird.



Dieses Symbol erscheint, wenn der Motorcontroller ausgeschaltet ist, da dieser zum Fahren selbstverständlich benötigt wird.



Dieses Symbol leuchtet auf, wenn der Motorcontroller in den Fehlerzustand schaltet. Dies kann verschiedenste Gründe haben, der Fehler kann daher auch nur von geschultem Fachpersonal behoben werden.



Dieses Symbol wird eingeblendet, wenn die Temperatur des Motorcontrollers auf über 80 Grad Celsius steigt.



Dieses Symbol erscheint, wenn der Motor heißer als 80 Grad Celsius wird.



Dieses Symbol wird angezeigt, wenn der Not-Aus-Schalter betätigt wurde.

Wenn kein Fehler oder Alarm vorliegt, wird der grüne Text angezeigt. Alle diese genannten Funktionen wurden mit *Checkbits* erstellt, die den Bildern und Texten zugewiesen wurden. Ganz unten befindet sich die *Softkeyleiste*, welche ein Wechseln zu den genannten *Screens* ermöglicht.

#### 6.4.3.8 Weitere Funktionen

Um die Visualisierung noch vielfältiger zu erstellen, wurden folgende weitere Funktionen realisiert:



Abbildung 6.28: Antriebsschlupfregelungsanzeige

#### 6.4.3.8.1 Globale Antriebsschlupfregelungsanzeige

Diese besteht aus dem gelben ASR Zeichen, welches so programmiert wurde, dass dies immer erscheint wenn die Antriebsschlupfregelung aktiv ist und den Motor gerade zurückregelt, ähnlich wie bei einem normalen Fahrzeug. Die Funktion wurde mithilfe eines *Checkbits* erstellt und ist theoretisch auf jedem Screen unter dem Akkustand sichtbar.

#### 6.4.3.8.2 Globale Fehleranzeige



Abbildung 6.29: Fehleranzeige

Zu der einzelnen Fehleranzeige im Fehlerscreen wurde noch eine „Globale Fehleranzeige“ eingefügt, da nicht losgefahren werden kann, wenn ein Fehler vorliegt. Anhand

dieser Anzeige kann man sofort erkennen, dass es sich dann um einen Fehler handelt, wenn das rote Warnsymbol aufleuchtet. Die Funktion wurde mit einem Checkbit realisiert. Die Globale Fehleranzeige und Antriebsschlupfregelungsanzeige wurden übereinandergelegt, da die beiden Symbole sowieso nicht gleichzeitig aufleuchten können, da bei einem Fehlerfall nicht gefahren werden kann.

#### 6.4.3.8.3 Bestätigung für Rennmodus



Abbildung 6.30: Fake Softkey

Da der Rennmodus nur mit dem zweiten Schlüssel gefahren werden darf, erscheint in der Visu ein roter *Fake-Softkey* ohne Funktion über dem Softkey des Rennmodus. Dadurch kann nicht mehr in den Rennmodus gewechselt werden.

### 6.4.4 Übergabevariablen

Um die notwendigen Daten von der SPS auf die Visu übergeben zu können, wurden Übergabevariablen erstellt. Diese wurden in 2 Bereiche aufgeteilt.

#### 6.4.4.1 Variablenliste

In der Variablenliste sind alle Variablen enthalten, welche für die ganzen Funktionen zuständig sind.

Name	Von	Zu	Funktion
VisuAcknowledge	Screen	Class	Mit dieser Variable wird übermittelt, ob der Quittieren-Button gedrückt wurde
VisuAlarm	Class	Screen	Mit dieser Variable wird übermittelt, ob ein Fehler vorliegt
VisuASRActive	Class	Screen	Mit dieser Variable wird übermittelt, ob die Antriebsschlupfregelung gerade aktiv ist
VisuASREnable	Class	Screen	Mit dieser Variable wird übermittelt, ob die Antriebsschlupfregelung eingeschaltet ist
VisuBatterylevel	Class	Screen	Mit dieser Variable wird der Akkustand in Prozent übermittelt
VisuBrake	Class	Screen	Mit dieser Variable wird übermittelt, ob gerade gebremst wird
VisuDriveEnable	Screen	Class	Mit dieser Variable wird übermittelt, ob losgefahren werden darf
VisuFlagDrivemode	Class Screen	Screen Class	Mit dieser Variable wird übermittelt, ob sich der User im Rennmodus oder im Fahrmodus befindet und es kann auch zwischen den beiden Modi gewechselt werden
VisuGear	Class	Screen	Mit dieser Variable wird der Gang übermittelt
VisuKeylock1	Class	Screen	Mit dieser Variable wird das Vorhandensein des ersten Schlüsselschalters übermittelt
VisuKeylock2	Class	Screen	Mit dieser Variable wird das Vorhandensein des zweiten Schlüsselschalters übermittelt
VisuMotorcontroller	Class	Screen	Mit dieser Variable wird übermittelt, ob der Motorcontroller eingeschaltet ist
VisuSpeed	Class	Screen	Mit dieser Variable wird die Geschwindigkeit übermittelt

#### 6.4.4.2 Alarmliste

In der Alarmliste befinden sich alle Übergabevariablen für die Alarme, welche auf dem Fehlerscreen angezeigt werden.



VisuTempController	Class	Screen	Mit dieser Variable wird die Temperatur des Motorcontrollers übermittelt
VisuTempMotor	Class	Screen	Mit dieser Variable wird die Temperatur des Motors übermittelt
VisuThrottle	Class	Screen	Mit dieser Variable wird die Stellung des Gaspedales in Prozent übermittelt

Tabelle 6.2: Variablenliste

Übergabevariable	Fehler
AlarmBatteryEmpty	Der Ladezustand der Akkus beträgt weniger als 5 Prozent
AlarmBatteryHot	Die Akkus sind zu heiß
AlarmEmergencyStop	Der Not-Aus Schalter wurde betätigt
AlarmIsolation	Es liegt ein Isolationsfehler vor
AlarmKeylock	Der Schlüsselschalter fehlt
AlarmMotorHot	Der Motor ist zu heiß
AlarmMotorcontrollerError	Der Motorcontroller weist einen beliebigen Fehler auf
AlarmMotorcontrollerHot	Der Motorcontroller ist zu heiß
AlarmMotorcontrollerOff	Der Motorcontroller ist nicht eingeschalten
AlarmThrottle	Der gültige Bereich des Gaspedales wurde verlassen

Tabelle 6.3: Alarmliste

#### 6.4.4.3 Einstellungen im Lasal Class 2 und Lasal Screen

Um die Variablen übergeben zu können, mussten einige Einstellungen gemacht werden.

Im Lasal Class 2 wurden die *Server (Variablen)*, die visualisiert werden sollten, in deren *Property Windows* so eingestellt, dass *Visualized* auf *true* gesetzt wurde. Zudem wurden die Variablen, mit denen Werte von der Visualisierung zur SPS gesendet werden, so eingestellt, dass *WriteProtected (Schreibgeschützt)* auf *false* gesetzt wurde, da die Übergabe sonst nicht möglich wäre. In den *Properties* der erstellten *Klassen (Programmierobjekte)* welche die Übergabevariablen beinhalten, wurde *Visualized* ebenfalls auf *true* gesetzt. Als diese Einstellungen abgeschlossen wurden, erschienen die „Augen“ in der Klasse und neben den Variablen. Diese Augen zeigen an, dass diese Variablen visualisierbar sind und in der Übergabedatei „*MaeExp.txt*“ gespeichert wurden und im Lasal Screen aufgerufen werden können.

Um im Lasal Screen auf diese Variablen zugreifen zu können, wurde unter dem Reiter Project das im Bild 6.32 sichtbare *Reference to Variables-Fenster* geöffnet. In diesem



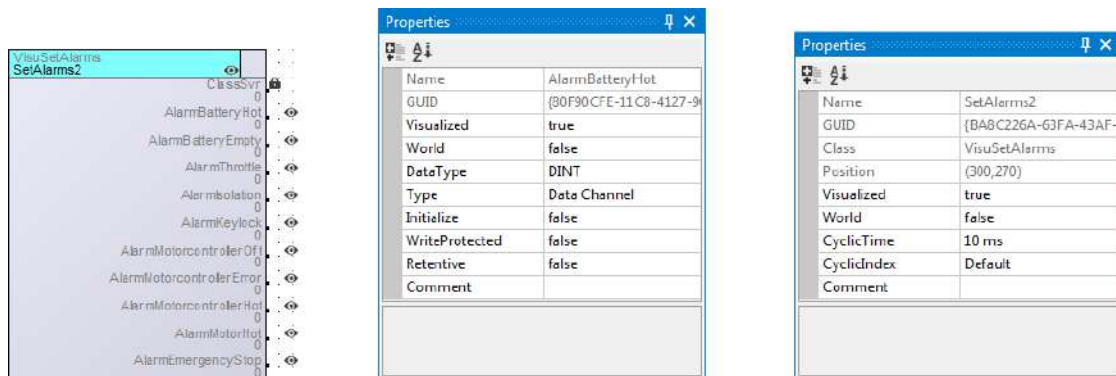


Abbildung 6.31: Variablen und Klasseneinstellungen

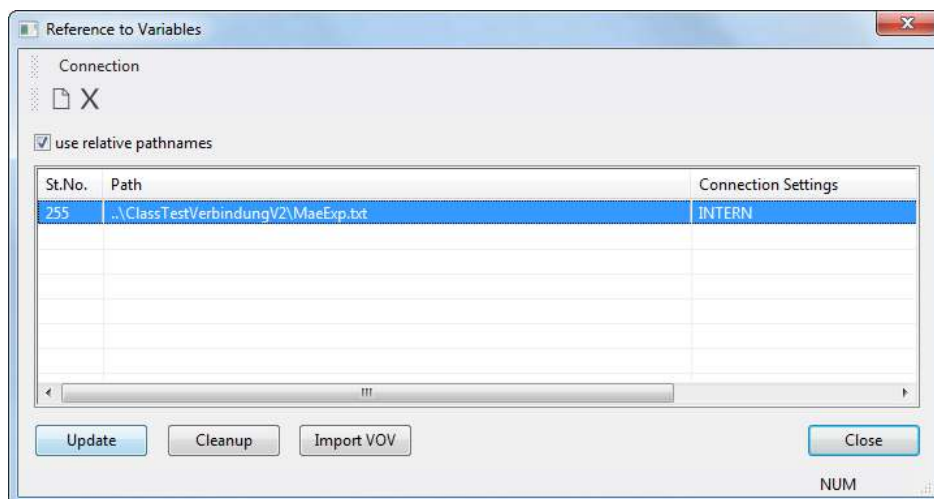


Abbildung 6.32: References to Variables

Fenster wurde unter *Insert New* ein neuer Eintrag erstellt. In diesem wurde die *Station Number* von 255 eingetragen, da das Display keine eigene Zentraleinheit besitzt und sich das Lasal Screen Projekt somit auf der SPS befindet. Unter *Path* wurde die „*MaeExp Textdatei*“ vom dazugehörigen Lasal Class-Projekt ausgewählt. Diese Textdatei befindet sich im automatisch erstellten Ordner, in dem das Class-Projekt gespeichert ist. Unter „Update“ konnten nun die Variablen in das Screen-Projekt hineingespielt und als *Server* verwendet werden.

## 6.4.5 Arbeitsoberfläche

Nachfolgend werden alle verwendeten Objekte und Einstellungen beschrieben, mit denen gearbeitet wurde.

### 6.4.5.1 Grundeinstellungen

Zu allererst wurde beim Erstellen des Projektes die passende Auflösung zum Display gewählt. Dabei musste die *Easy-Variante (für CAN-Bus)* gewählt werden, da es sich um ein CAN-Display handelte. Um das Programm auf die SPS zu spielen, wurden dieselben *Online Settings* eingestellt wie beim Lasal-Class-Programm. Die meisten Objekte konnten mit Rechtsklick auf die Arbeitsoberfläche platziert werden.

### 6.4.5.2 Images

Mit einem Doppelklick auf *Images* auf der rechten Seite im Projektbaum wurde das *Image-Fenster* geöffnet, in dem alle Einstellungen der Bilder gemacht werden konnten.

#### 6.4.5.2.1 Bilder importieren

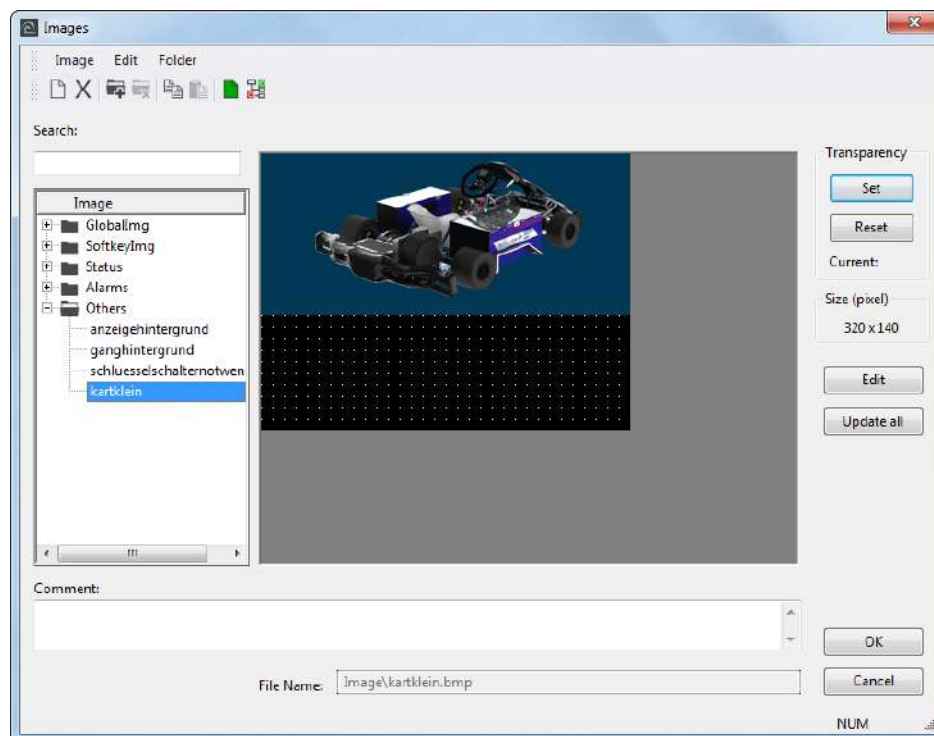


Abbildung 6.33: Images Fenster

Unter *Add new image* wurden die benötigten Bilder eingefügt. Dabei musste beachtet werden, dass nur Icon-, JPEG- und Bitmap-Files importiert werden konnten. Nach dem Importieren konnte unter dem Button *Set* eine Farbe ausgewählt werden, welche transparent gesetzt werden konnte. Dabei musste beachtet werden, dass im Photo-shop keine Übergänge zwischen dem benötigten Bild und dem Hintergrund gemacht

wurden, da man diesen sonst erkennen würde, weil ein Übergang aus mehreren Farben besteht. Zusätzlich könnte das Bild im Lasal Screen kleiner gemacht werden. Dies wurde jedoch nicht gemacht, da für größere Auflösungen mehr Speicherplatz benötigt wird und dieser nur begrenzt vorhanden war.

#### 6.4.5.2 Bilder platzieren

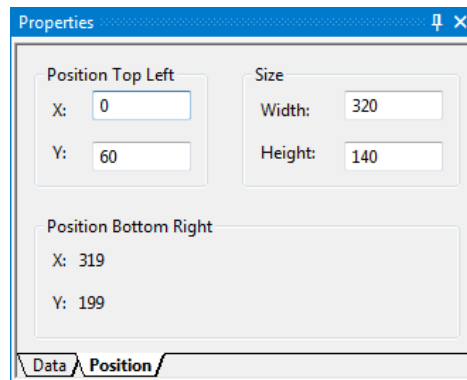


Abbildung 6.34: Properties Positionfenster

Nachdem das Bild mit Rechtsklick und der Auswahl des Bildes platziert wurde, konnte im *Properties-Fenster (Eigenschaften)* im Reiter *Position* die richtige Position festgelegt werden. Alle dortigen Parameter sind in Pixel angegeben. Da die Bilder schon in passender Größe eingefügt wurden, wurden nur noch die Werte von *Top Left* verändert.

#### 6.4.5.3 Texte

Die Erstellung der Texte wurde mit folgendem Schema gemacht: Als erstes wurden *TextLists* erstellt, welche den Inhalt des Textes beschreiben. Nach Platzierung des Textes wurde diesem ein *Font (Schriftgröße und Schriftart)* zugewiesen.

##### 6.4.5.3.1 TextLists

Mit einem Doppelklick auf *Textlists* im Projektbaum auf der rechten Seite wurde das Fenster aus Abbildung 6.35 geöffnet. Unter *Add new textlist* wurde eine neue Liste erstellt und dort wurde ein neuer Text mit dem Button *Add new Text* erstellt. Diese Texte wurden dann beim Einfügen eines neuen Textes mit Rechtsklick ausgewählt. Alternativ wurden die Texte auch bei *Buttons* als Beschriftungen eingefügt.

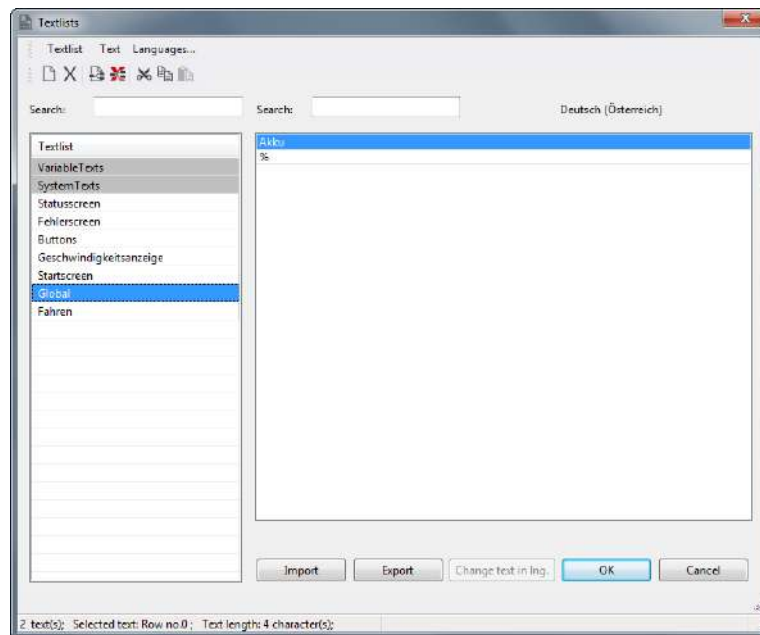


Abbildung 6.35: Textlistsfenster

#### 6.4.5.3.2 Weitere Einstellungen

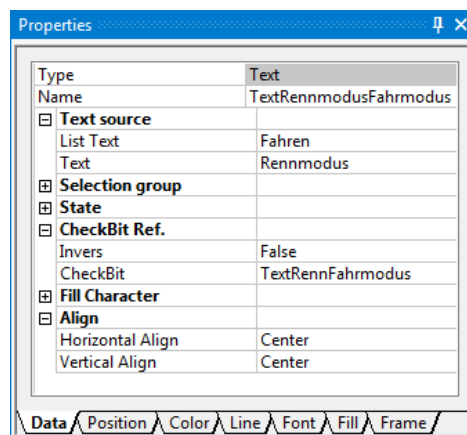


Abbildung 6.36: Propertiesfenster Texte

Wie bei den Bildern konnten im *Properties-Fenster* diverse Einstellungen gemacht werden. Im *Data Fenster* wurden alle notwendigen Einstellungen für die benötigten Funktionen vergeben. Im *Position Fenster* wurde die Position eingestellt. Dieses Fenster ist genauso aufgebaut wie bei den *Images*. Im *Colour Fenster* wurde die Farbe festgelegt und im *Font Fenster* eine neue Schriftart erstellt und ausgewählt. Die anderen Fenster waren für die Realisierung der Visualisierung irrelevant.

Um einen neuen *Font* zu erstellen, wurde das *Font-Fenster* geöffnet. Dort wurde unter *Add new Font* ein neuer Eintrag erstellt. In diesem konnte unter *Font* die

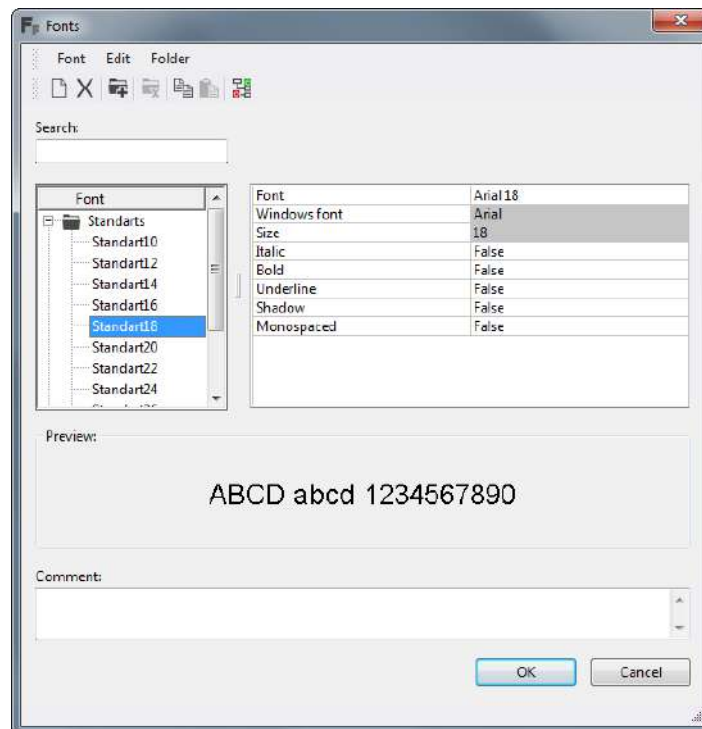


Abbildung 6.37: Fontfenster

Schriftart aus einer Datenbank ausgewählt werden. Bei manchen Texten wurde eine *Fette* Schriftart benötigt. Diese wurde eingestellt, indem *Bold* auf *True* gesetzt wurde.

#### 6.4.5.4 Function Blocks

*Function Blocks* wurden dazu verwendet, um bestimmte Funktionen zu ermöglichen, wie zum Beispiel die Umschaltung zwischen den einzelnen Screens. Diese Funktion wird beispielsweise bei *Softkeys* benötigt. *Function Blocks* können Objekten, wie zum Beispiel Buttons, zugewiesen werden, welche zuvor erstellt wurden. Mit einem Doppelklick auf *Function Blocks* wird das Definitionsfenster geöffnet.

##### 6.4.5.4.1 Softkeys

Unter *Add new function Block* wurde ein neuer Funktionsblock erstellt. In diesem wurde durch *Add new function* eine neue Funktion eingefügt. Diese wurde im Dropdownmenü von *Functions* ausgewählt. Für *Softkeys* wurde die Funktion *Screen* gewählt. In dieser Funktion wurde unter *Direct Choice* der Screen ausgewählt, in den gewechselt werden sollte. Alternativ dazu könnte die *Screennummer* unter Parameter1 eingetragen werden.

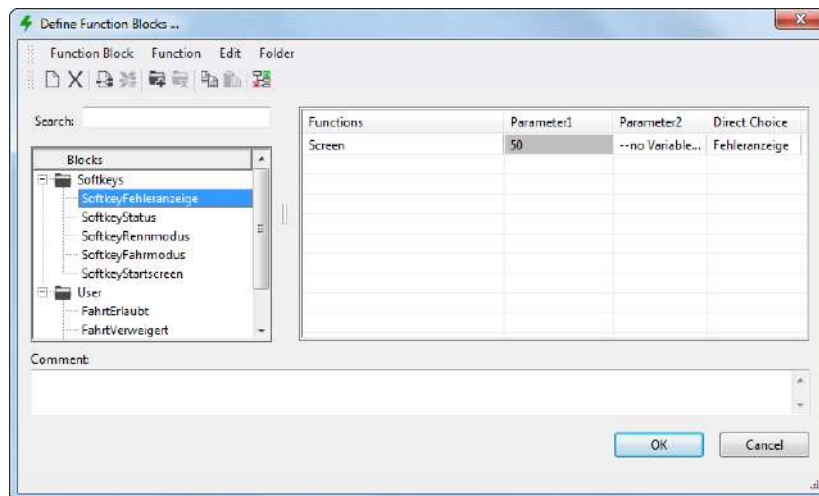


Abbildung 6.38: Functionblock Softkey

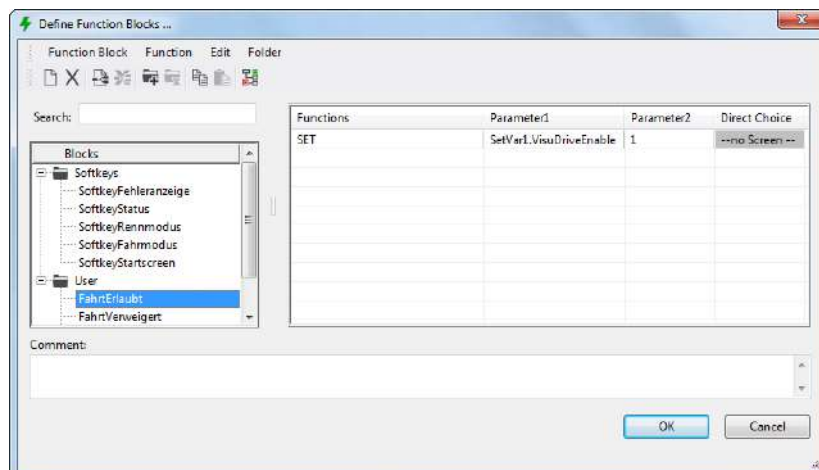


Abbildung 6.39: Functionblock Set

#### 6.4.5.4.2 Set Befehl

Um die Voraussetzung zu erfüllen, dass nur losgefahren werden darf, wenn sich die Visualisierung im Fahrscreen befindet, wurde der *SET-Befehl* verwendet. Dazu wurde ein neuer Funktionsblock erstellt, in dem eine neue Funktion zugefügt wurde. Unter *Function* wurde *SET* ausgewählt. Als Parameter1 wurde die Variable ausgewählt, welche gesetzt werden soll und unter Parameter2 wurde der Wert festgelegt, welchen die Variable annehmen sollte.

#### 6.4.5.5 Datenanzeige

Um diverse Daten zu realisieren, wurden Elemente der Datenanzeige eingefügt. Die wichtigsten drei werden im Anschluss beschrieben.

#### 6.4.5.5.1 NumEdits

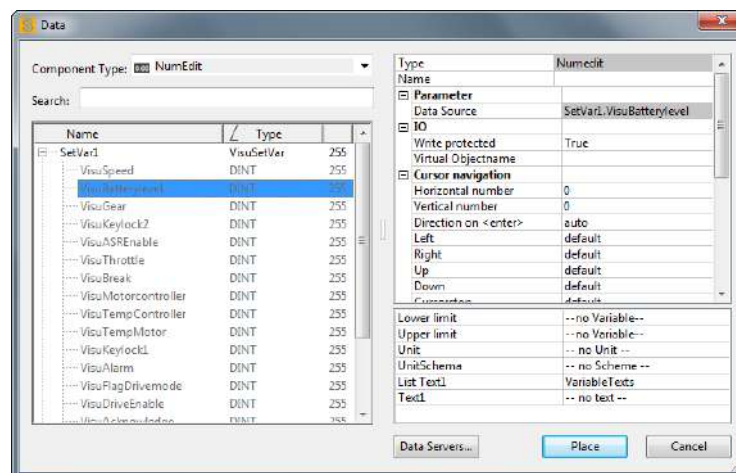


Abbildung 6.40: NumEdit

Mit *NumEdits* wurden alle Numerischen Zahl-Ausgaben erstellt. Diese wurden mit *Rechtsklick/Place/Data* eingefügt. Dort konnte unter *Parameter/Data Source* die zu visualisierende Variable eingestellt werden. Anschließend wurden die *NumEdits* richtig platziert. Diese Einstellung konnte im *Properties Fenster* vorgenommen werden. Dort wurde außerdem der dazugehörige *Font* vergeben.

#### 6.4.5.5.2 Bars

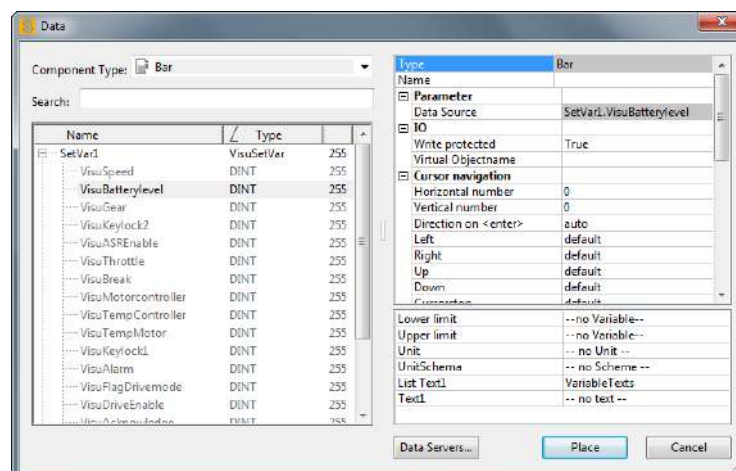


Abbildung 6.41: Bar

Durch einen *Bar (Balken)* wurde zum Beispiel die Anzeige des Akkus realisiert. Diese werden gleich wie die *NumEdits* platziert, lediglich dass *Bar* als *Function* ausgewählt wird. Unter *Data Source* konnte die zu visualisierende Variable eingestellt werden. Die Farben und Position des Balkens wurden im *Property Fenster* eingestellt.



### 6.4.5.5.3 Checkbits

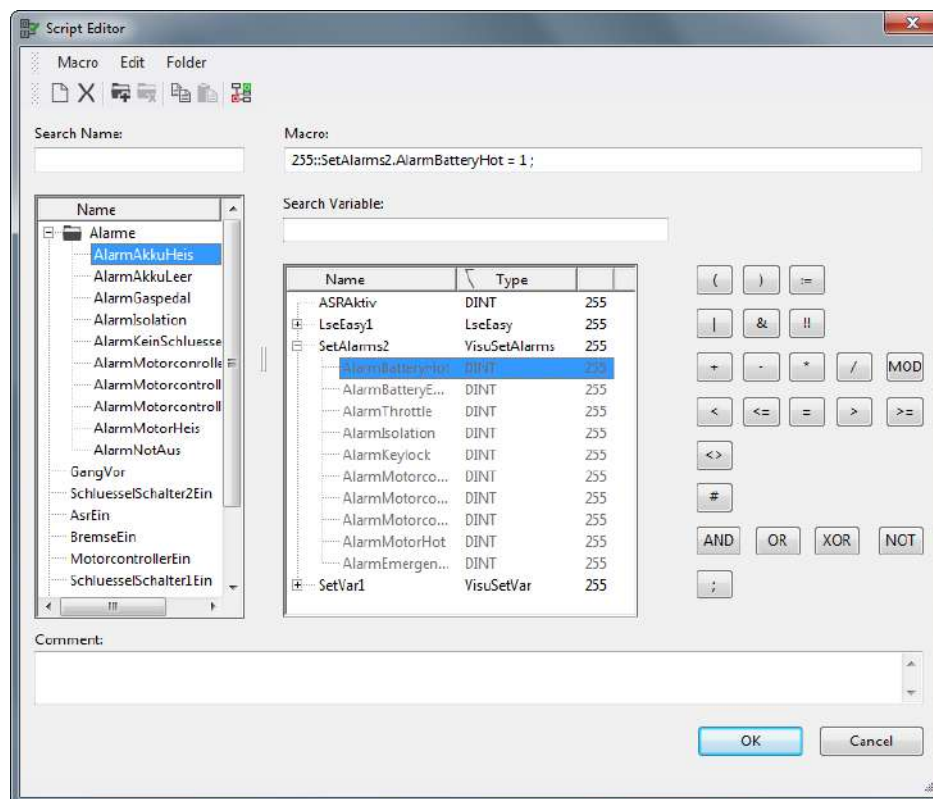


Abbildung 6.42: Checkbit

*Checkbits* bieten die Funktion, Objekte in der Visualisierung auf inaktiv zu schalten. Diese wurden hauptsächlich für Anzeigen verwendet, welche nur unter gewissen Voraussetzungen angezeigt werden sollten, wie zum Beispiel die Anzeigen der Alarme. Mit einem Doppelklick auf *Checkbits* öffnet sich das oben angezeigte Fenster. Dort konnte unter *Add new Checkbit* ein neuer Eintrag erstellt werden. Im Bereich unter *Macro* wurde die benötigte Prüfung programmiert. Dazu konnten die Variablen, welche vom Class-Projekt importiert wurden, verwendet werden. Diese Variablen befinden sich im mittlerem Fenster und können durch Doppelklick verwendet werden. Das erstellte *Checkbit* wurde dann einem Element in dessen *Properties Fenster* zugewiesen. Dabei könnte dieses, wenn erforderlich, invertiert werden.

### 6.4.5.6 Generelle Voraussetzungen zum Fahrbetrieb

Um losfahren zu können, muss der Fahrscreen aktiv sein, denn im Statusscreen kann zum Beispiel nur geprüft werden. Außerdem muss der erste Schlüsselschalter eingesteckt sein. Zudem darf kein Fehler vorliegen, welcher im rechten oberen Bereich der Visualisierung sichtbar wäre. Für eine genauere Information des Fehlers muss in den Fehlerscreen gewechselt werden. Der Rennmodus kann nur gefahren



werden, wenn die Losfahrbedingungen erfüllt sind und der zweite Schlüsselschalter aktiv ist.

## 6.4.6 Notwendige Änderungen

Im Verlauf der Programmierung und der anschließenden Prüfung wurden einige Fehler festgestellt. Viele *Fonts* unter Schriftgröße 14 konnten nicht richtig dargestellt werden. Deshalb wurden diese bei den *Softkeys* im Photoshop bearbeitet und als Bild mit dem darüber liegenden Zeichen eingefügt. Ebenfalls wurde festgestellt, dass bei der internen Bildbearbeitung nur genau eine einzige Farbe ausgeblendet werden kann. Dafür mussten einige Bilder nachbearbeitet werden. Ansonsten konnte die Visualisierung ohne größere Probleme erstellt werden.

# Kapitel 7

## Weitere Realisierungsmöglichkeiten

### 7.1 Ausblick

Als mechanische Erweiterung wäre denkbar die Fertigung von weiteren Abdeckungen.

So sollen über die Räder der Hinterachse Kotflügel aus Plexiglas montiert werden. Des Weiteren würde sich anbieten, die kompletten Antriebseinheiten mit einer durchgehenden Abdeckung zu versehen, um den Zugriff auf drehende Komponenten an der Hinterachse weitgehend zu Verhindern und so die Sicherheit zu erhöhen. Um den Fahrbetrieb auch bei widrigsten Wetterbedingungen zu ermöglichen, könnten alle Komponenten abgedichtet werden.

Auch Interessant wäre die Messung der Elektromagnetischen Wellen bei Fahrbetrieb und die damit resultierenden Einwirkungen auf den Fahrer. Bei Strömen  $>500A$  könnte das entstehende EMF bereits gesundheitsschädlich sein. Generell sollten Personen mit einem Herzschrittmacher mit diesem Kart nicht mehr fahren.

Ein weiterführendes Projekt unserer Anwendung wäre die Verwirklichung dieses Antriebskonzepts in einem vollwertigen PKW. Der Motor wäre mit seinem Drehmoment von  $85Nm$  bereits ausreichend, um einen Kleinwagen zu bewegen.

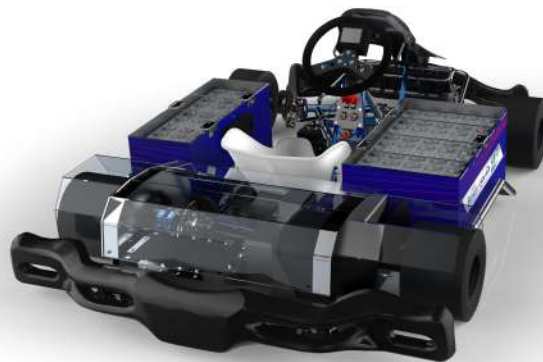


Abbildung 7.1: Geplante Abdeckungen (Hinten)



# Anhang A

## Anhang

### A.1 Quellcodes des Programms der SPS

#### A.1.1 DriveKart

```
FUNCTION VIRTUAL GLOBAL DriveKart::CyWork
  VAR_INPUT
    EAX    : UDINT;
  END_VAR
  VAR_OUTPUT
    state (EAX)    : UDINT;
  END_VAR

  // Einlesen der Variablen

  AIThrottle := AIThrottle.Read();
  TempCutback := TempCutback.Read();
  TorqueCutback := TorqueCutback.Read();
  Enable := Enable.Read();
  DIBrake := DIBrake.Read();
  DIDirection := DIDirection.Read();
  ActSpeed := ActSpeed.Read();
  DIControllerOK := DIControllerOK.Read();
  InDriveMode := InDriveMode.Read();
  DIBoostSafety.Write(input := DIBoostSafety);

  // Berechnung Gas
  // Dabei werden beide Werte der Regler beruecksichtigt

  // Einlesen der Pedalstellung
  Throttle := AIThrottle;
```

```
// Offset
Throttle -=8500;

//Umwandlung in einen passenden Zahlenbereich
Throttle /=5;
Throttle *=TempCutback/1000;
Throttle *=TorqueCutback/1000;

//Ueberpruefung , ob Boost-Modus von der Visu
//freigegeben wurde

if (InDriveMode=1)&(DIBoostSafety=1) then

    DOEnableBoost:=1;
    DOEnableDrive:=0;

else

    DOEnableBoost:=0;
    DOEnableDrive:=1;

end_if;

//Auswahl der Fahrtrichtung
//die Fahrtrichtung kann nur gewechselt werden,
//wenn das Fahrzeug steht
//und die Bremse gedruickt ist

if (DIBrake=1) & (ActSpeed=0) then

    if DIDirection=1 then

        Forward:=1;
        Backward:=0;

    elsif DIDirection=0 then

        Forward:=0;
        Backward:=1;

    end_if;
end_if;
```

```
//Bremsung
//Gas muss 0 sein

if DIBrake=0 then

    EnableBreak:=1;
    Throttle:=0;

else

    EnableBreak:=0;

end_if;

//Setzen der Ausgaenge
//Das darf erst passieren, wenn
//der Controller eingeschaltet ist
//und das Fahren freigegeben wurde

    if (DIControllerOK=1)&(Enable=1) then

        //Setzen und beschreiben des Aktivierungsausgangs
        EnableController:=1;
        DOEnableController.Write (DOEnableController);

        //Ausgeben des Gaspedals
        OutputThrottle:=Throttle;
        OutputThrottle.Write(input:=OutputThrottle);

        //Uebermittlung an Visu, dass
        //Controller eingeschaltet ist
        MotorcontrollerOK:=1;

        //Ausgeben des Gaspedals
        AOThrottle.Write (OutputThrottle);

        //Beschreiben der Ausgaenge
        DOEnableBreak.Write (EnableBreak);
        DOForward.Write (Forward);
        DOBackward.Write (Backward);
        DOEnableDrive.Write (DOEnableDrive);
        DOEnableBoost.Write (DOEnableBoost);
```

```
else

    MotorcontrollerOK:=0;

    //Setzen der Ausgaenge auf ihren Low Wert,
    //um fliegende Potentiale zu vermeiden
    AOThrottle.Write (0);
    DIControllerOK.Write (0);
    DOEnableBreak.Write (0);
    DOForward.Write (0);
    DOBackward.Write (0);
    DOEnableDrive.Write (0);
    DOEnableController.Write (0);
    EnableController:=0;
    OutputThrottle:=0;

end_if;

//Aktualisieren der Server
EnableBreak.Write(EnableBreak);
MotorcontrollerOK.Write(MotorcontrollerOK);

state := READY;

END_FUNCTION
```



## A.1.2 EnableDrive

```
FUNCTION VIRTUAL GLOBAL EnableDrive :: RtWork
  VAR_INPUT
    EAX          : UDINT;
  END_VAR
  VAR_OUTPUT
    state (EAX)   : UDINT;
  END_VAR

  // Einlesen der Eingaenge

  ASRError := ASRError.Read();
  InOvertemp := InOvertemp.Read();
  InResetError := InResetError.Read();
  DIIsolationError.Write (DIIsolationError);

  // Ueberpruefung des Gaspedals
  boThrottleError := DetectThrottleError();

  // Ueberpruefen des Akkustands
  CheckBatteryLevel();

  // Abfrage ob Fehler vorliegt

  if (ASRError = 1) | (InOvertemp = 1) | (BatteryLow = 1)
    | (boThrottleError = 1) | (DIIsolationError = 1) then

    boError := 1;

  // Wenn Fehler bereinigt ist,
  // kann ueber die Visualisierung quittiert werden
  elsif (boError = 1) & (InResetError = 1) then

    boError := 0;
    DOResetErrorSafety := 1;

  else

    DOResetErrorSafety := 0;

  end_if;
```

```
//Enable ist nur High, wenn kein Fehler vorliegt
Enable := (1 - boError);

//Uebergabevariable an die Safety-CPU
OutSafety := Enable;

//Schreiben der Ausgaenge

Enable.Write(Enable);
DOResetErrorSafety.Write(DOResetErrorSafety);
OutSafety.Write(OutSafety);

        state := READY;

END_FUNCTION

FUNCTION EnableDrive :: DetectThrottleError
VAR_OUTPUT
    result : BOOL;
END_VAR

//Funktion zur Ueberpruefung des Gaspedals
//Bei Verlassen des erlaubten Wertebereichs
//wird ein Fehler ausgegeben

//Einlesen des Fehlerregisters
BDIThrottleRange := BDIThrottleRange.Read();

//Ueberpruefung, ob sich Wert des Gaspedals
//im erlaubten Bereich befindet

if (BDIThrottleRange = 8)|(BDIThrottleRange = 128) then

    ThrottleError:=1;
    result:=1;

else

    ThrottleError:=0;
    result:=0;
```

```
end_if;

// Direktes Beschreiben des Servers ,
// somit kein Rueckgabewert benoetigt
ThrottleError . Write ( ThrottleError );

END_FUNCTION

FUNCTION EnableDrive :: CheckBatteryLevel
    VAR_OUTPUT
        result : BOOL;
    END_VAR

    // Diese Funktion ueberprueft den
    // Akkustand und gibt bei Unterschreiten
    // eines gewissen Werts einen Fehler aus

    // Einlesen des Batterielevels von analogem Eingang
    AIBatteryLevel := AIBatteryLevel . Read();

    // Umrechnung auf Prozent
    uiBatteryLevel := AIBatteryLevel * ConversionFactor;

    // Bei Ladezustand < 5% wird Fehler ausgegeben
    if uiBatteryLevel <= 5 then

        BatteryLow := 1;

    else

        BatteryLow := 0;

    end_if;

    // Direktes Beschreiben des Servers ,
    // somit kein Rueckgabewert benoetigt
    BatteryLow . Write ( input := BatteryLow );

END_FUNCTION
```

### A.1.3 TempControl

```
FUNCTION TempControl :: CalculateValues
```

```
    //Umrechnen der Werte  
    //der Temperatursensoren des Motors auf C  
    diTempMotor1 := AITempMotor1 * ConversionFactorMotor;  
    diTempMotor2 := AITempMotor2 * ConversionFactorMotor;
```

```
END_FUNCTION
```

```
FUNCTION TempControl :: CalculateTempMax
```

```
    VAR_OUTPUT  
        result : DINT;  
    END_VAR
```

```
    //Funktion zur Berechnung des Maximalwerts der Temperatur
```

```
    if (diTempMotor1 > diTempMotor2) &  
        (diTempMotor1 > diTempController) then
```

```
        result := diTempMotor1;
```

```
    elsif (diTempMotor2 > diTempController) then
```

```
        result := diTempMotor2;
```

```
    else
```

```
        result := diTempController;
```

```
    end_if;
```

```
    //Rueckgabewert ist die maximale Temperatur  
    // aller Komponenten
```

```
END_FUNCTION
```

```
FUNCTION VIRTUAL GLOBAL TempControl :: CyWork
```

```
    VAR_INPUT
```

```
EAX    : UDINT;
END_VAR
VAR_OUTPUT
    state (EAX) : UDINT;
END_VAR

// Einlesen der Variablen
AlTempMotor1 := AlTempMotor1.Read();
AlTempMotor2 := AlTempMotor2.Read();
AlTempController := AlTempController.Read();
DIOOverTempBattery := DIOOverTempBattery.Read();

// Berechnen der richtigen Werte
CalculateValues();

// Berechnen der Maximaltemperatur
diTempMax := CalculateTempMax();

// Implementierung des tatsächlichen Reglers
// Die Differenz zwischen Maximaltemperatur
// und aktueller Temp. bestimmt die Drosselung

if ((diTempMax > TempMax) & (TempCutback > 100)) |
    ((diTempMax < TempMax) & (TempCutback < 1000)) then

    TempCutback := (TempMax - diTempMax) * 10;
    ErrorOvertemp := 0;

elsif (diTempMax > TempMax) & (TempCutback < 100) then

    ErrorOvertemp := 1;

end_if;

// Feststellung ob eine Komponente zu heiß ist
if (diTempMotor1 > TempMax) | (diTempMotor2 > TempMax) then

    MotorOvertemp := 1;

else

    MotorOvertemp := 0;
```

```
end_if;  
  
if diTempController > TempMax then  
    ControllerOvertemp:=1;  
else  
    ControllerOvertemp:=0;  
end_if;  
  
//Das BMS uebermittelt mit einem digitalen Ausgang,  
//ob eine Uebertemperatur vorliegt  
BatteryOvertemp:=DIOverTempBattery  
  
//Beschreiben der Ausgaenge (Server)  
TempCutback.Write(input:=TempCutback);  
ErrorOvertemp.Write(input:=ErrorOvertemp);  
MotorOvertemp.Write(input:=MotorOvertemp);  
ControllerOvertemp.Write(input:=ControllerOvertemp);  
BatteryOvertemp.Write(input:=BatteryOvertemp);  
TempMotor.Write(input:=diTempMotor1);  
TempController.Write(input:=diTempController);  
  
state := READY;  
  
END_FUNCTION  
  
FUNCTION VIRTUAL GLOBAL TempControl::Init  
  
    //Initialisieren des Reglers und setzen auf Ausgangswert  
    TempCutback:=1000;  
  
END_FUNCTION
```

## A.1.4 Antriebsschlupfregelung

### A.1.4.1 SpeedWheelFront

```
FUNCTION VIRTUAL GLOBAL SpeedWheelFront :: CyWork
  VAR_INPUT
    EAX : UDINT;
  END_VAR
  VAR_OUTPUT
    state (EAX) : UDINT;
  END_VAR

  // Einlesen des Eingangs
  DICounter := DICounter.Read();

  // Nach jeder 1/2 Sekunde wird die Berechnung ausgeführt
  if ops.tAbsolute > (udTimeMark + 500) then

    // Umrechnungsfaktor 40=60/(3*0,5)
    SpeedRpm:= DICounter*40;

    // Zurücksetzen des Zählers um Überläufe zu verhindern
    DICounter:=0;
    DICounter.Write (DICounter);

    // Zurücksetzen des Zeitzählers
    udTimeMark:=ops.tAbsolute;

  end_if;

  state := READY;
END_FUNCTION

FUNCTION VIRTUAL GLOBAL SpeedWheelFront :: Init

  // Erstmaliges setzen des Zeitzählers
  udTimeMark:=ops.tAbsolute;

END_FUNCTION
```

#### A.1.4.2 ControlButton

```
FUNCTION VIRTUAL GLOBAL ControlButton::CyWork
  VAR_INPUT
    EAX : UDINT;
  END_VAR
  VAR_OUTPUT
    state (EAX) : UDINT;
  END_VAR

  // Einlesen der Eingänge
  Input := Input.Read();

  // Einfache Flankenerkennung mittels Merker
  IF Input > boFlagIn THEN

    Output := 1 - boFlagOut;

  END_IF;

  DOOutput := Output;
  DOOutput.Write (DOOutput);

  // Umspeichern der Merker
  boFlagIn := Input;
  boFlagOut := Output;

  state := READY;

END_FUNCTION
```

#### A.1.4.3 ASR

```
FUNCTION VIRTUAL GLOBAL ASR::CyWork

  VAR_INPUT
    EAX : UDINT;
  END_VAR
  VAR_OUTPUT
    state (EAX) : UDINT;
  END_VAR

  // Einlesen der Eingänge
```



```
SpeedBack := SpeedBack.Read();
SpeedFrontRight := SpeedFrontRight.Read();
SpeedFrontLeft := SpeedFrontLeft.Read();
EnableASR := EnableASR.Read();

//Berechnung des Mittelwerts
//der Drehzahlen der Vorderräder
SpeedFrontAverage:=(SpeedFrontLeft+SpeedFrontRight)/2;

//Vergleich der Drehzahlen der Vorderräder
//Bei eineinhalbfacher Abweichung
//regelt die Motorleistung herunter

if (SpeedFrontLeft > 1.5*SpeedFrontRight) |
    (SpeedFrontRight > 1.5*SpeedFrontLeft) then

    Cutback -=20;

else

    Cutback +=1;

end_if;

//Vergleich der Drehzahlen von Vorne und Hinter
//Bei eineinhalbfacher Abweichung wird
//die Motorleistung heruntergeregelt

if (SpeedFrontAverage <1.5*SpeedBack) |
    (1.5*SpeedFrontAverage > SpeedBack) then

    Cutback -=20;

else

    Cutback +=1;

end_if;

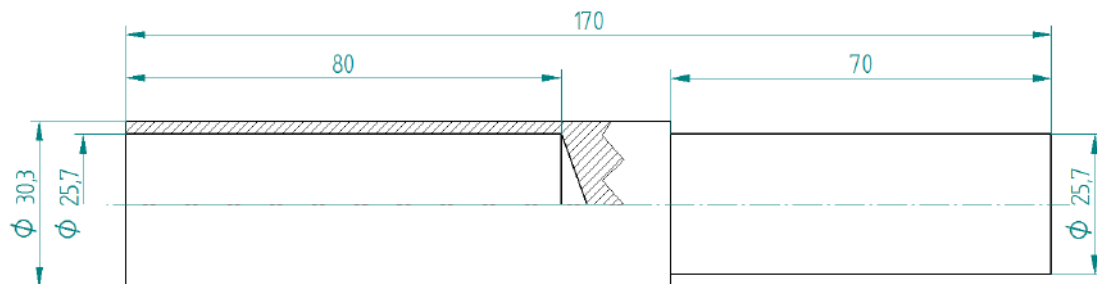
//Bevor die Motorleistung zu weit absinkt,
//wird ein Fehler ausgegeben
//Dadurch wird der Motor abgeschaltet

if Cutback<=100 then
```

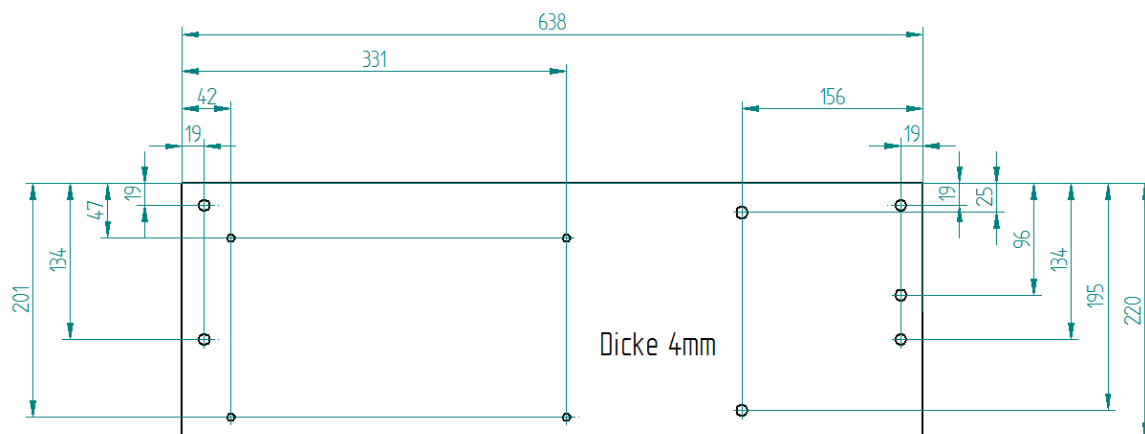
```
Cutback:=100;  
Error:=1;  
  
//Zuruecksetzen des Reglerwerts  
//Sonst wuerde bei normalem Betrieb  
//dieser Wert ins Unendliche steigen  
  
elsif Cutback>=1000 then  
  
    Cutback:=1000;  
  
else  
  
    Error:=0;  
  
end_if;  
  
//Schreiben der Ausgaenge  
TorqueCutback:=Cutback;  
TorqueCutback.Write(input:=TorqueCutback);  
Error.Write(input:=Error);  
  
state := READY;  
  
END_FUNCTION  
  
FUNCTION VIRTUAL GLOBAL ASR::Init  
  
    Cutback:=1000;  
  
END_FUNCTION
```

## A.2 CAD Werkstattzeichnungen

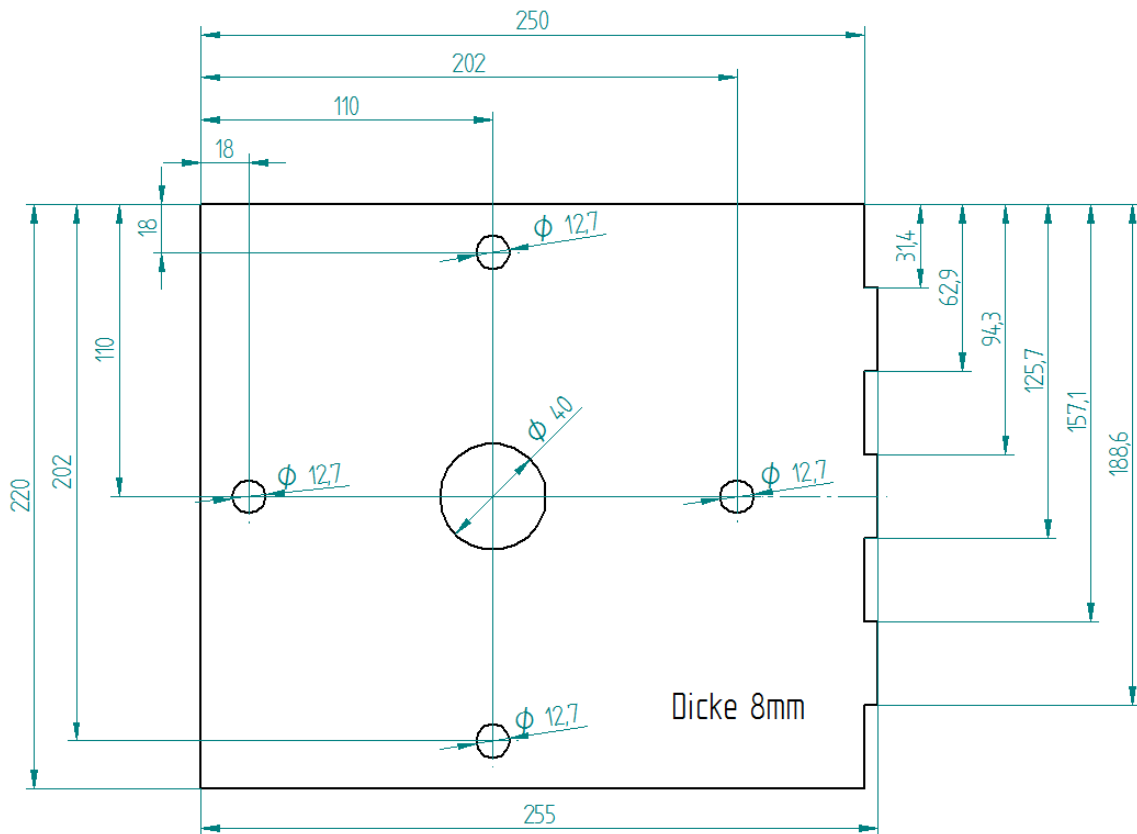
### A.2.1 Verlängerungsadapter

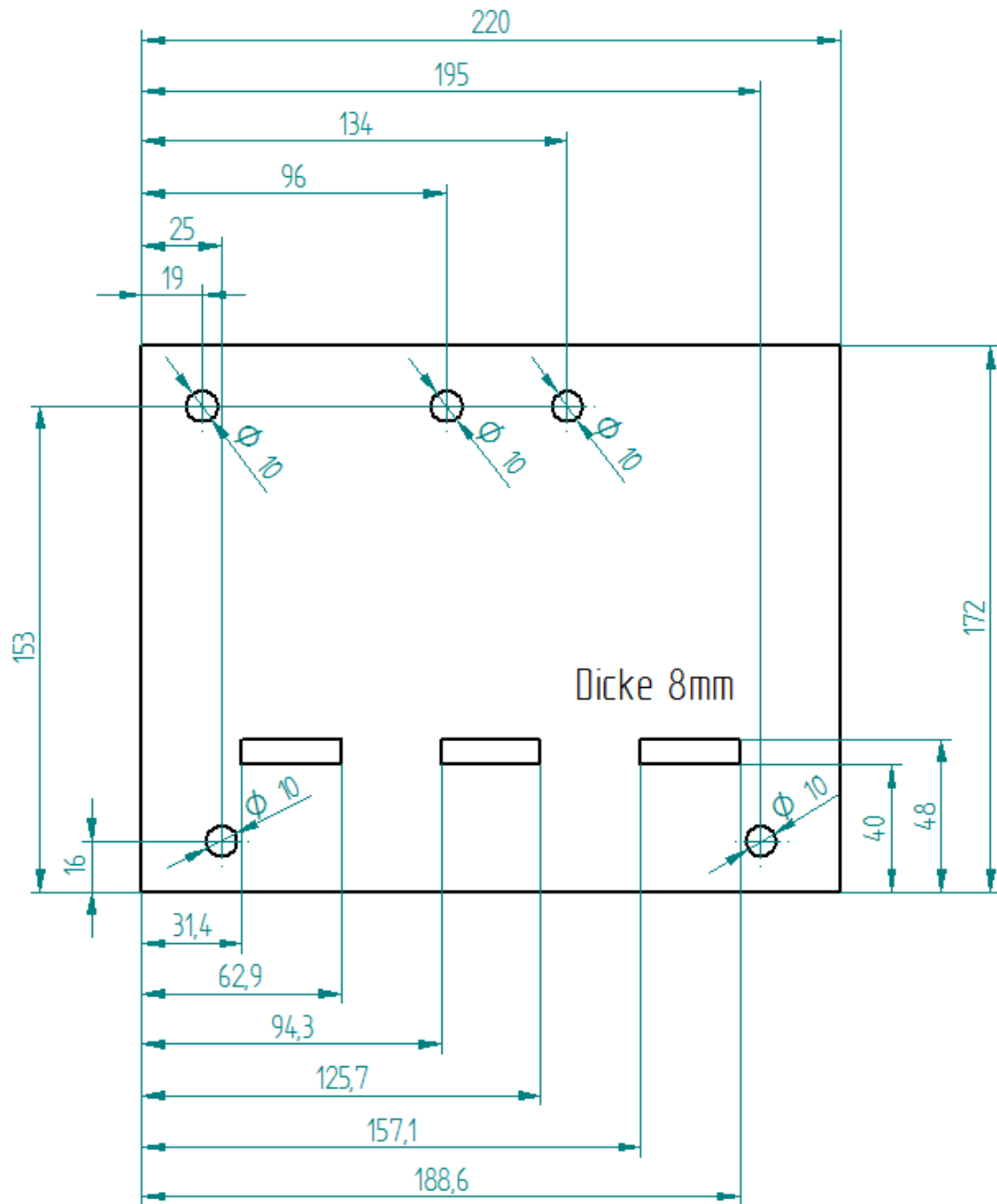


### A.2.2 Hintere Bodenplatte

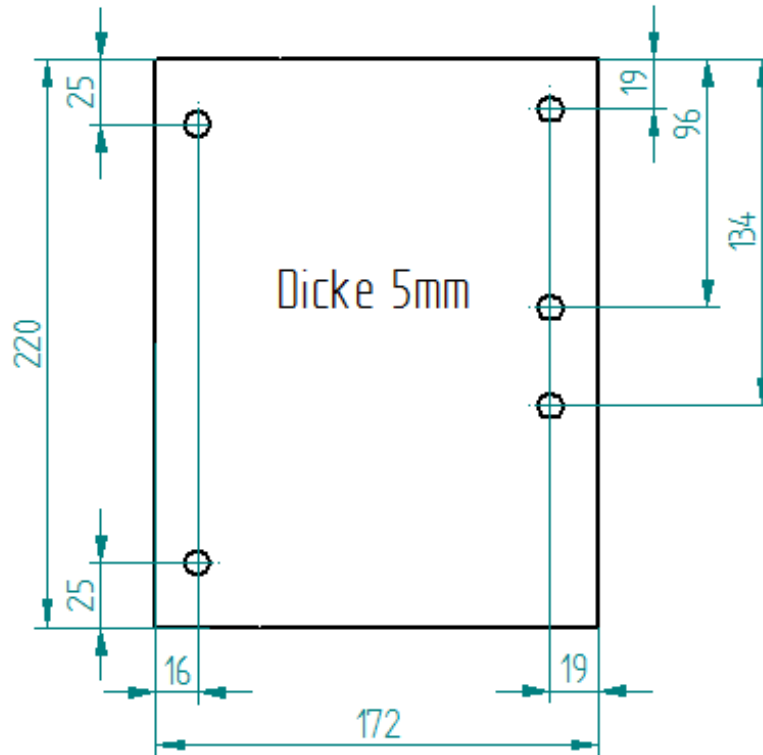


### A.2.3 Motorhalterung

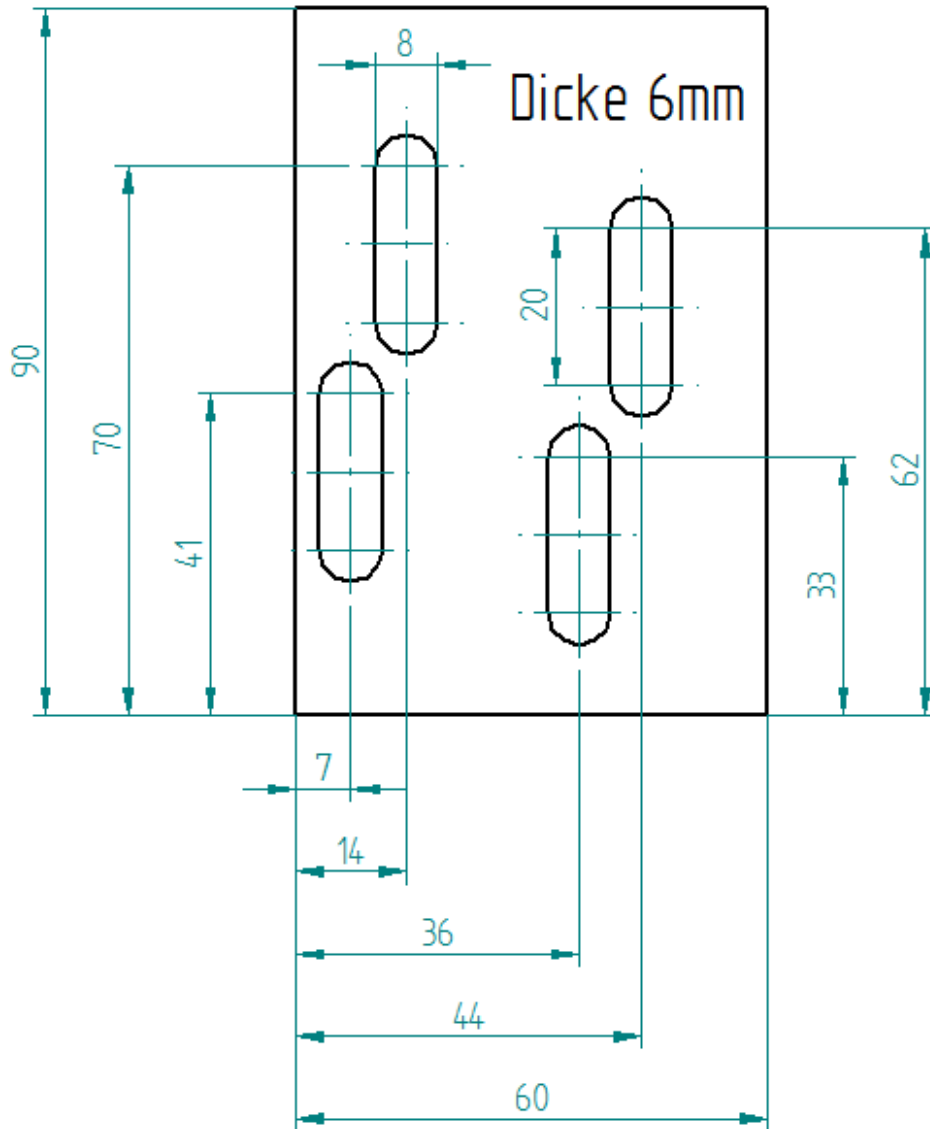




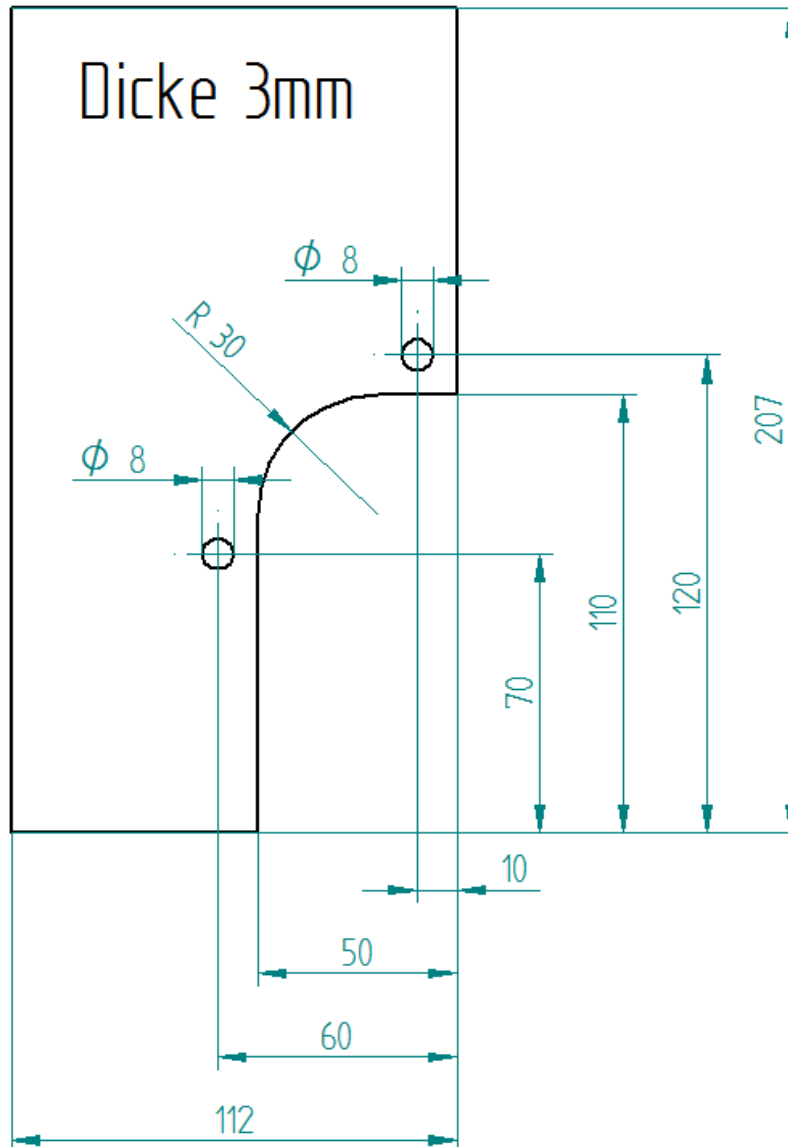
## A.2.4 Aufstockungsblech



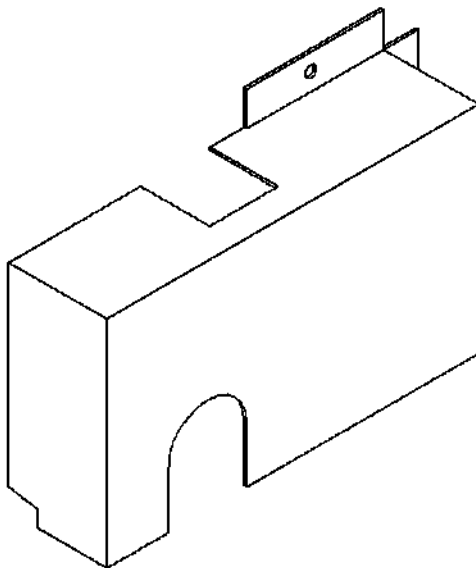
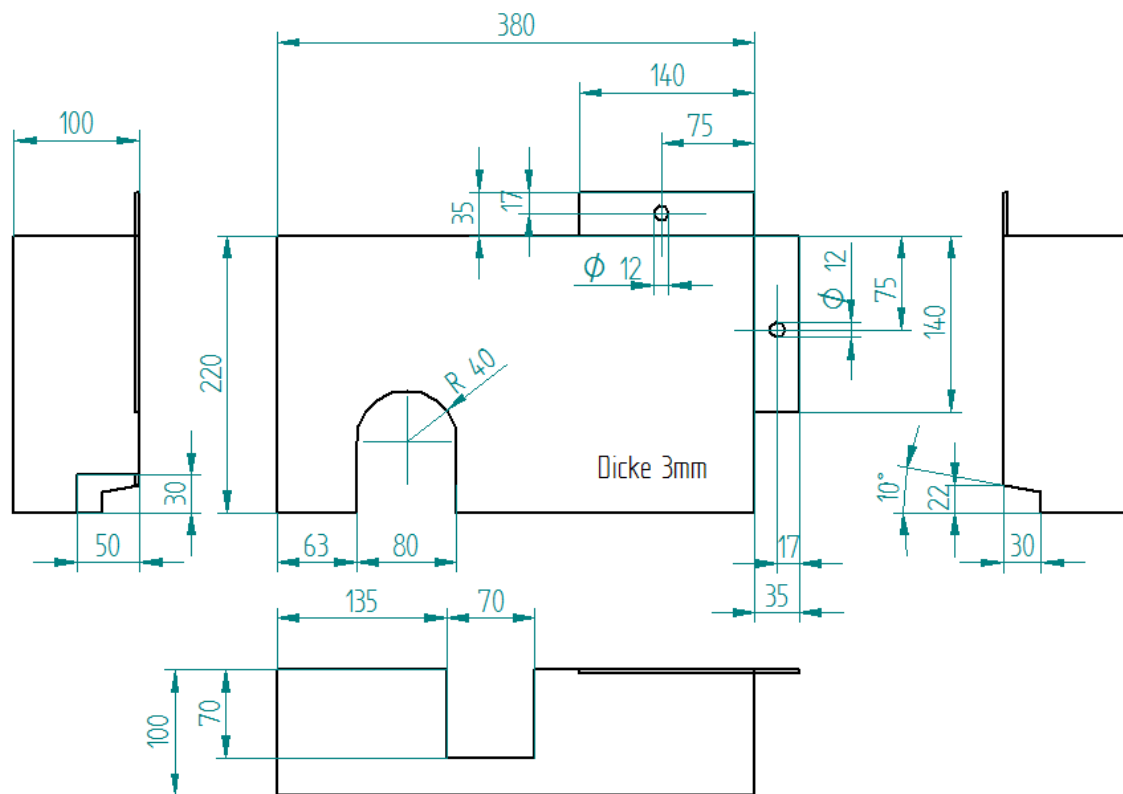
## A.2.5 Spannvorrichtung

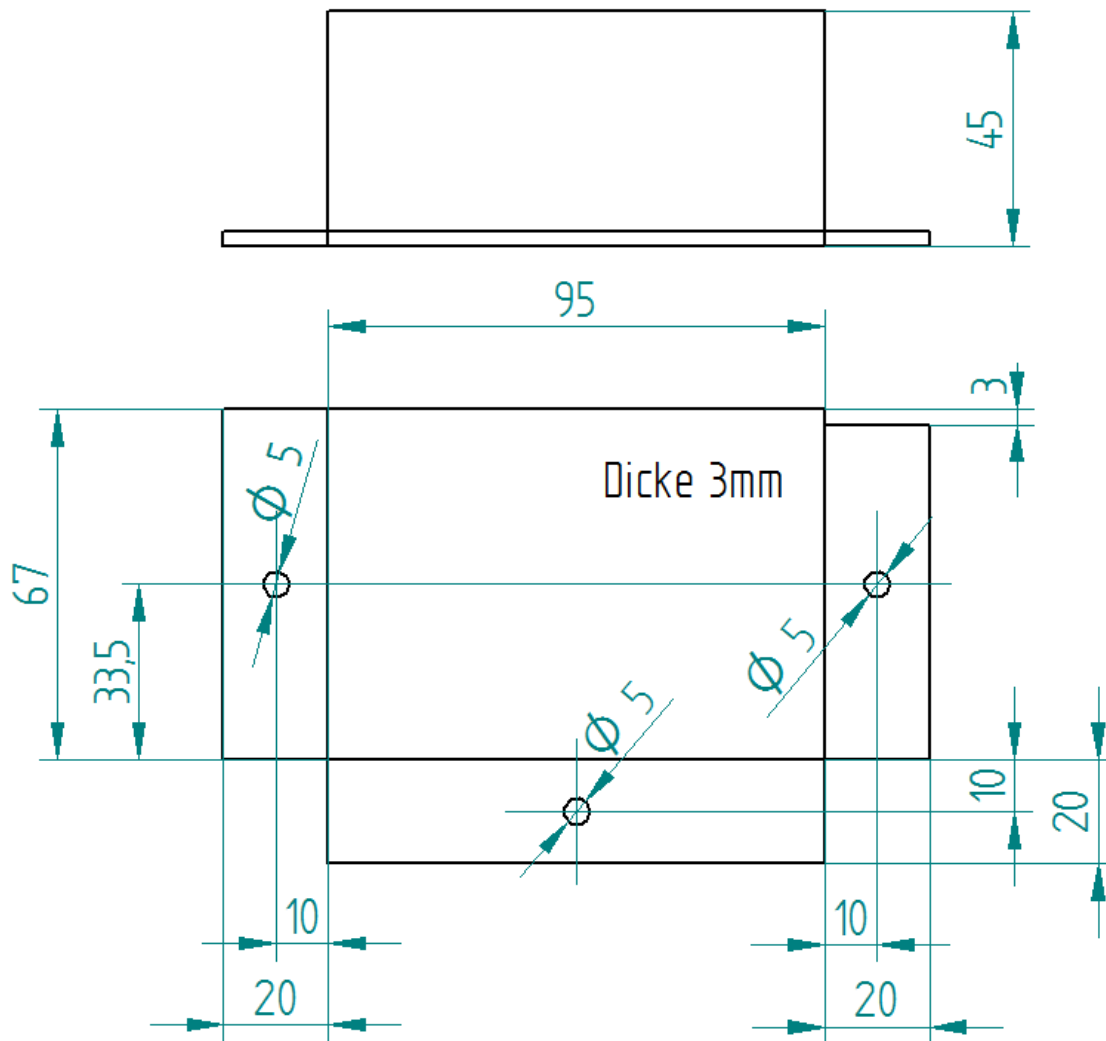


## A.2.6 Abdeckung Antrieb

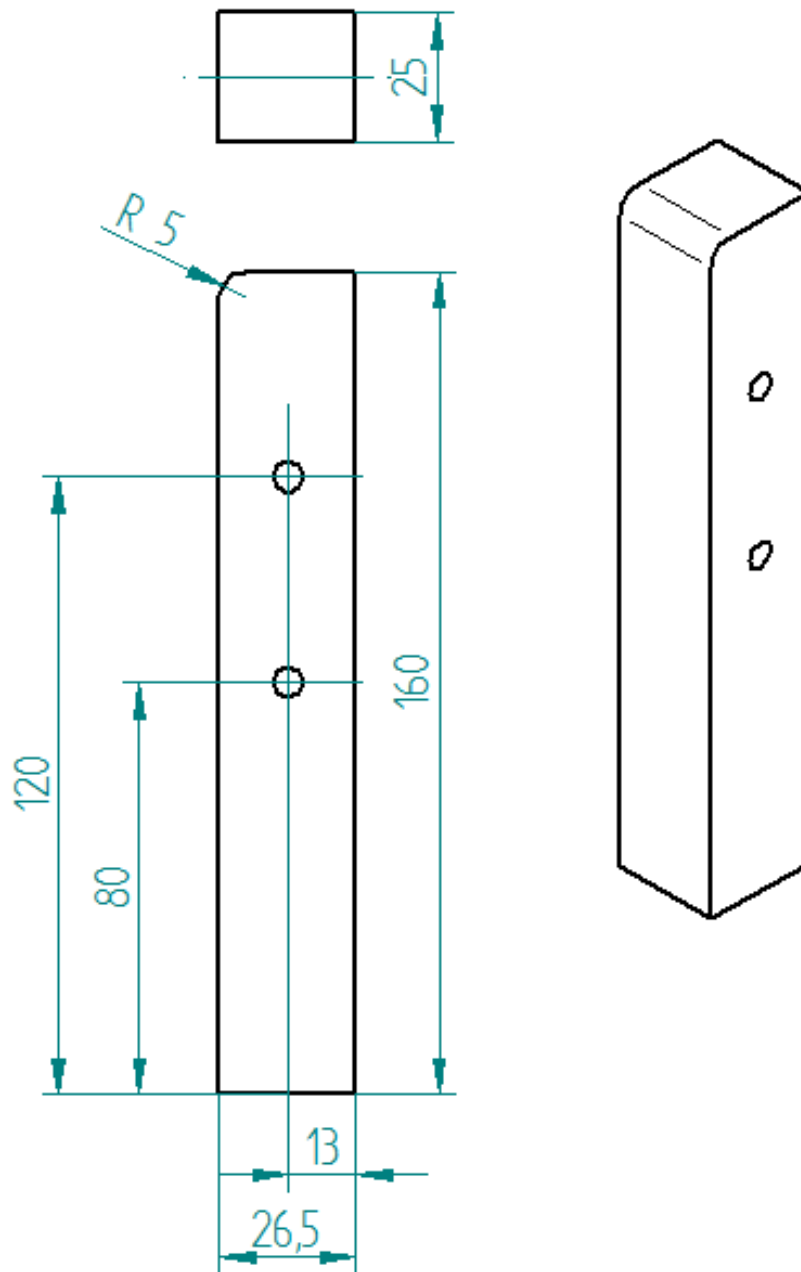




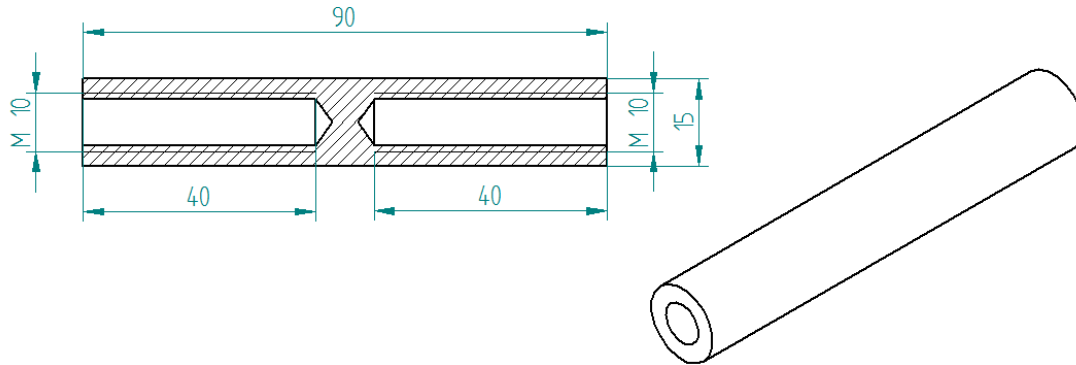


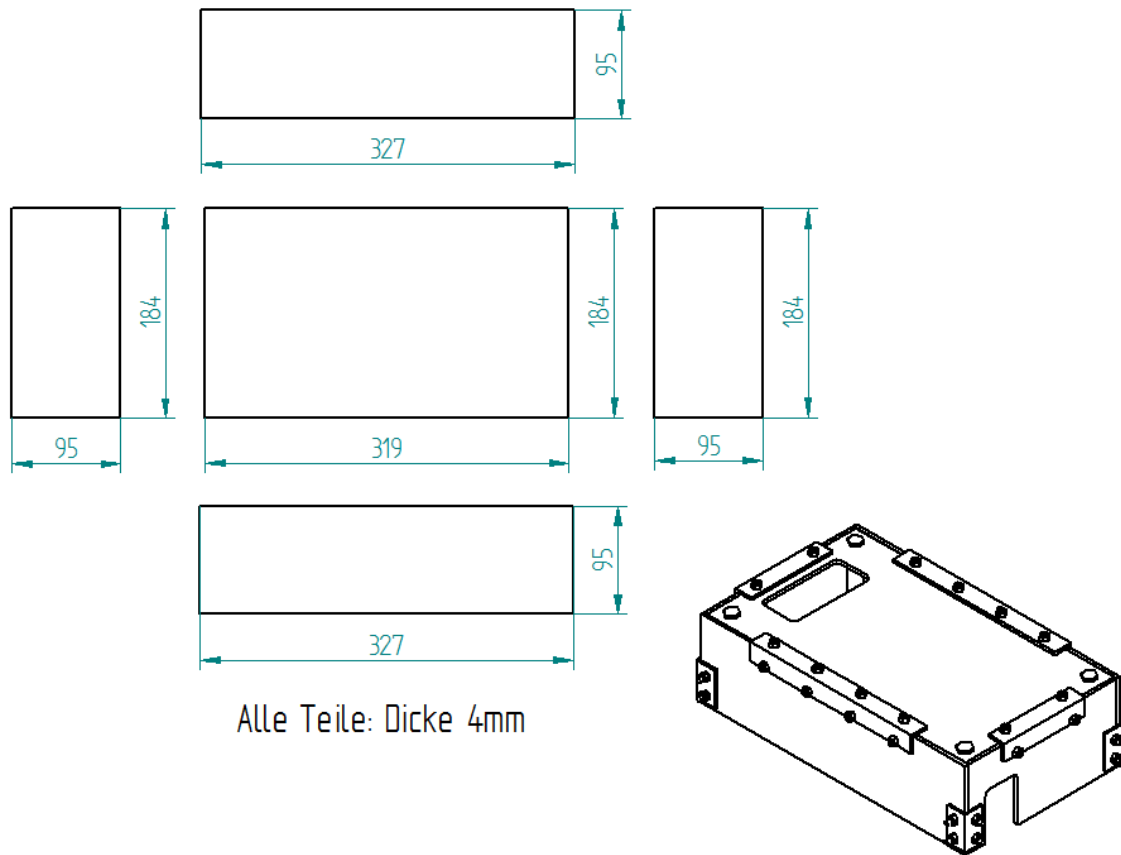


## A.2.7 Schutzverstärkung gegen Verbiegen

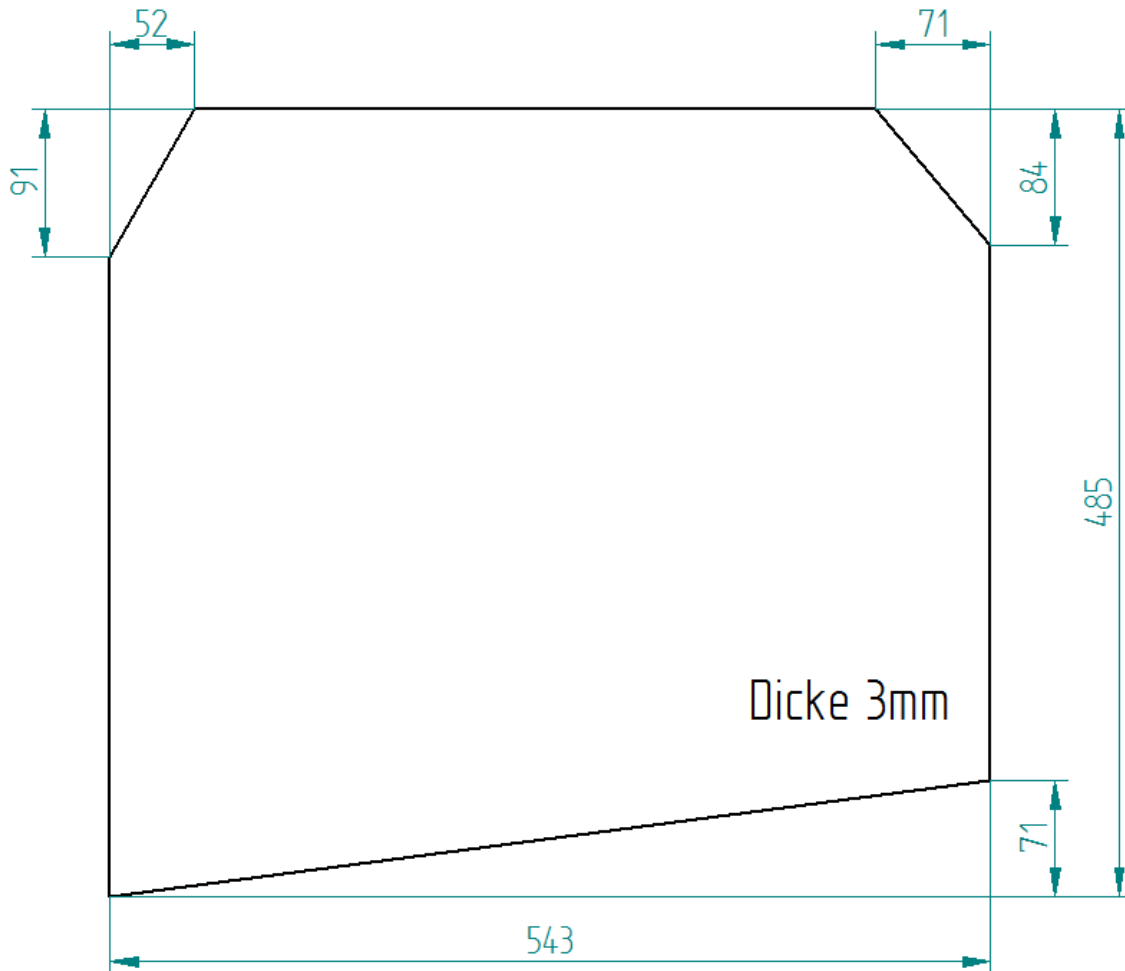


## A.2.8 Abdeckung Motorcontroller

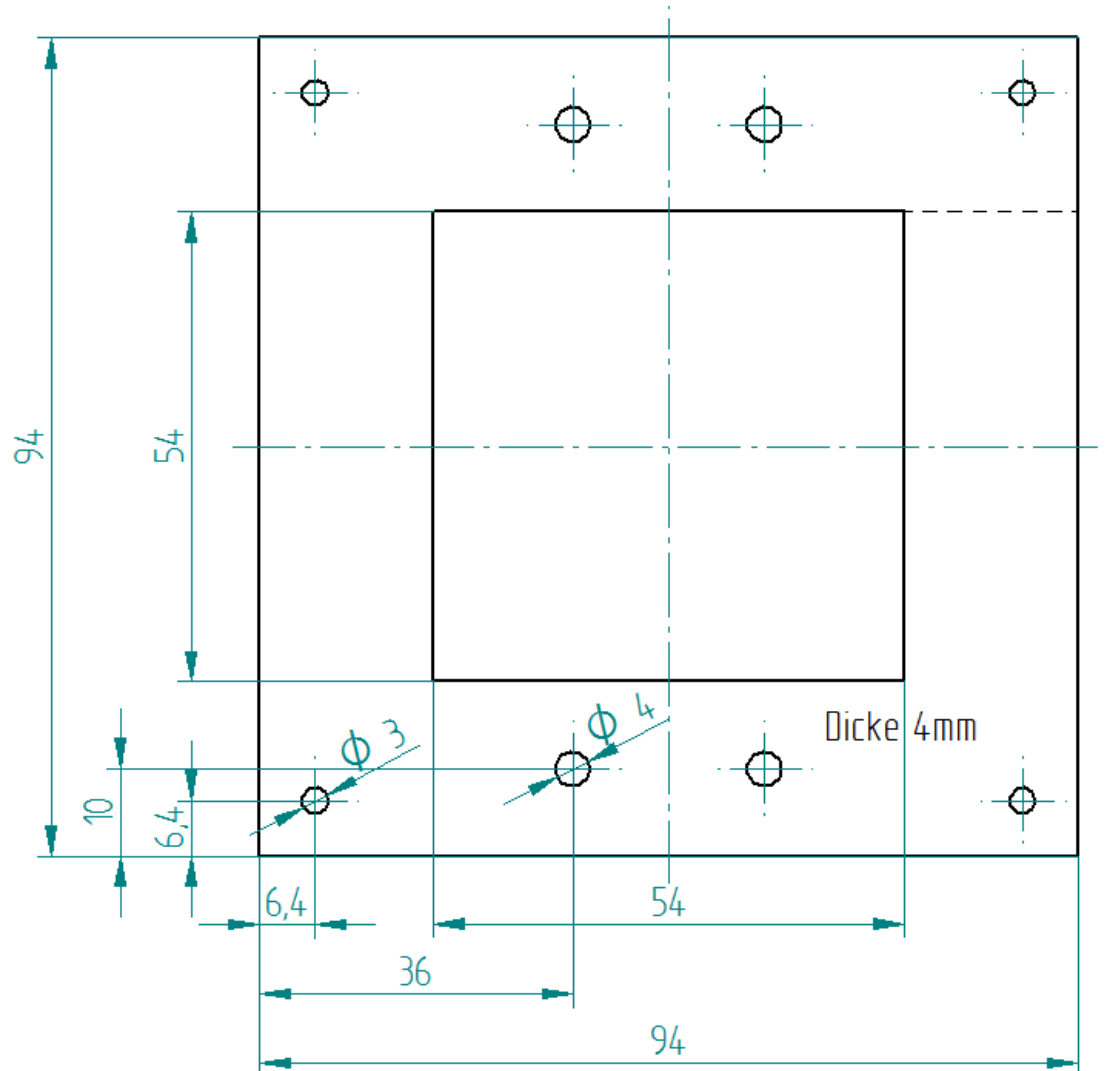




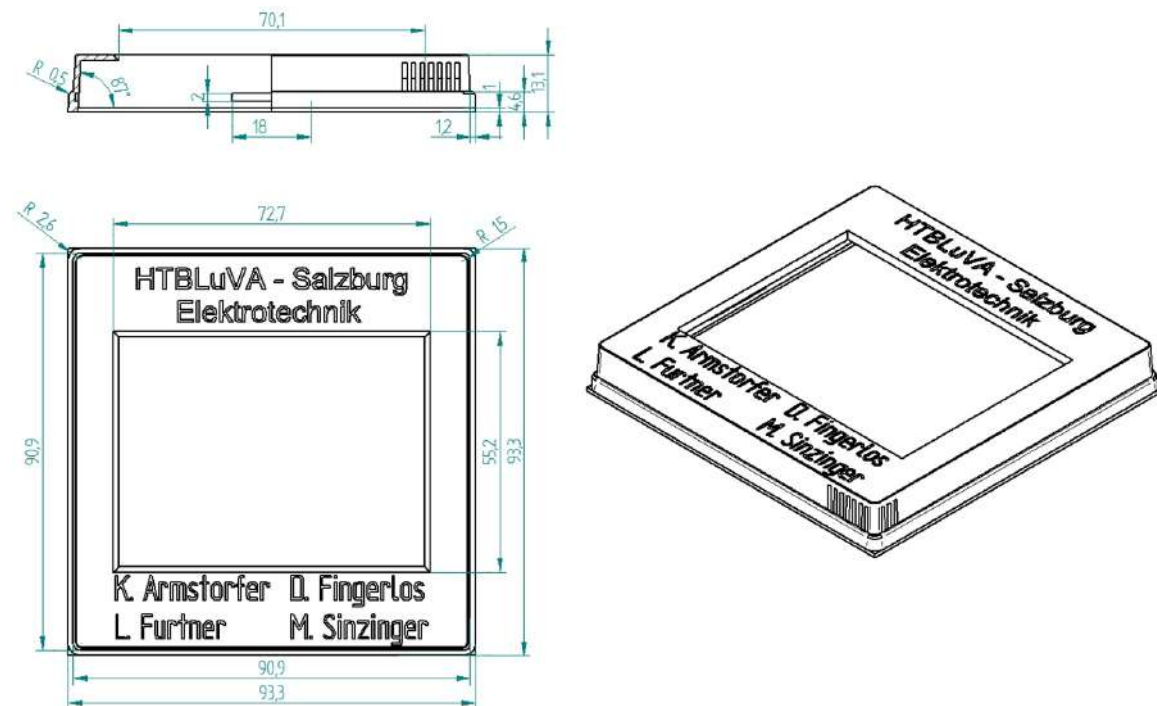
## A.2.9 Unterboden



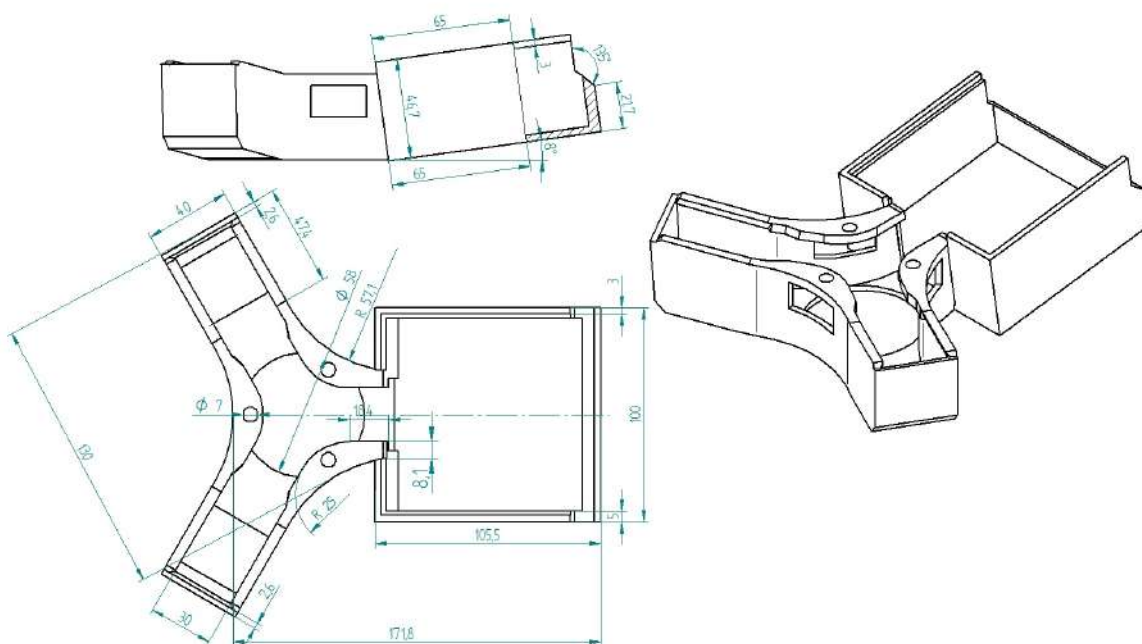
## A.2.10 Displayhalterung



### A.2.11 Displayverkleidung

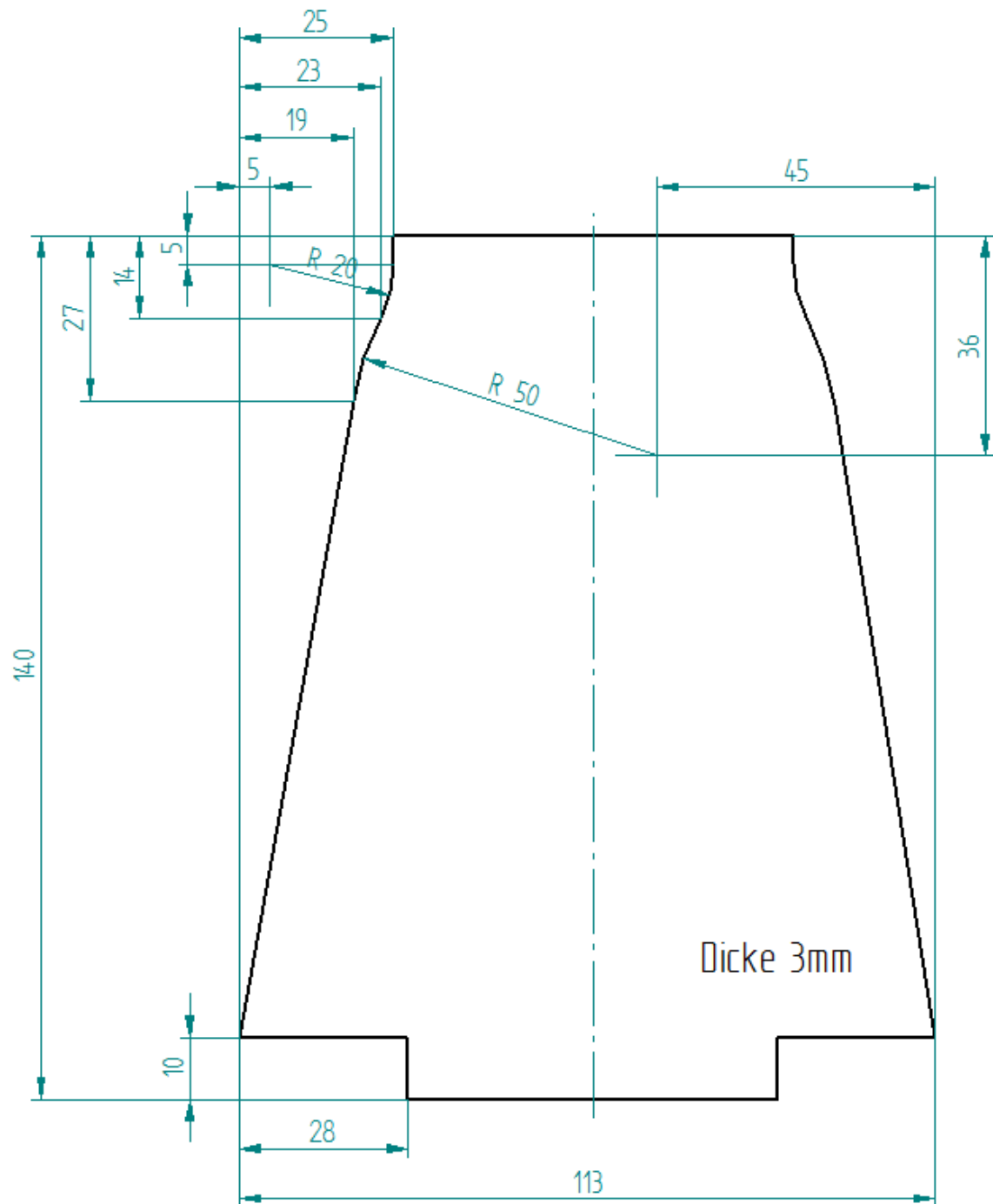


### A.2.12 Hinterabdeckung Lenkrad

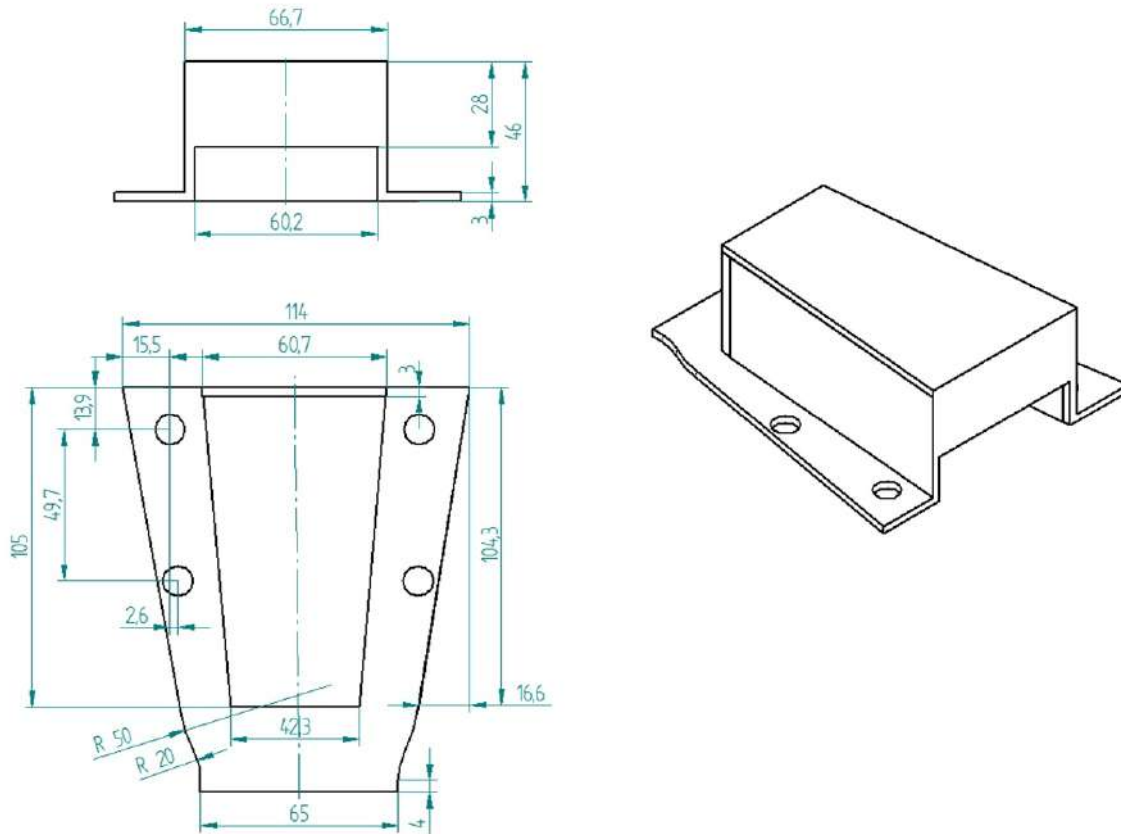




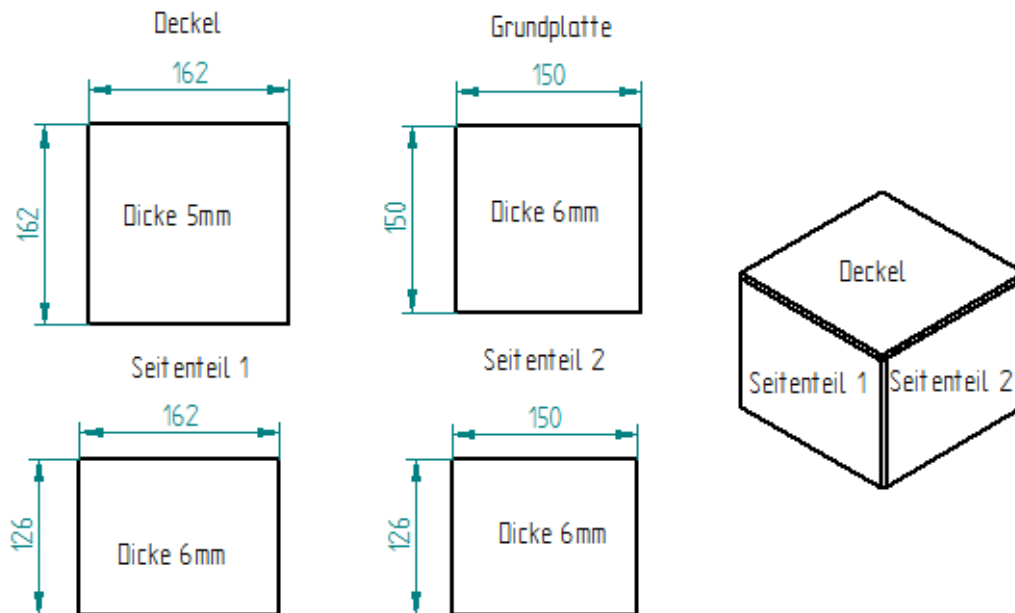
### A.2.13 Armaturenbrett



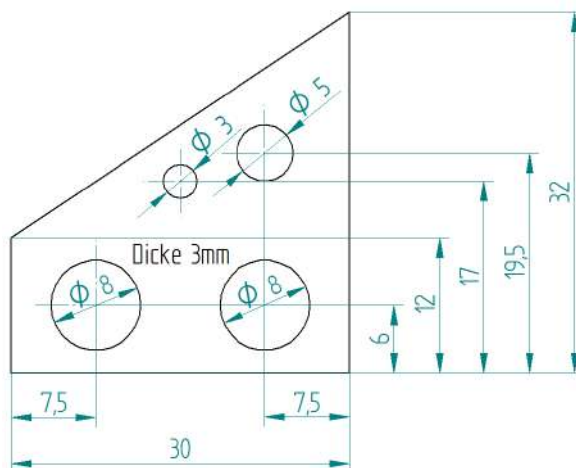
## A.2.14 Hinterabdeckung Armaturenbrett



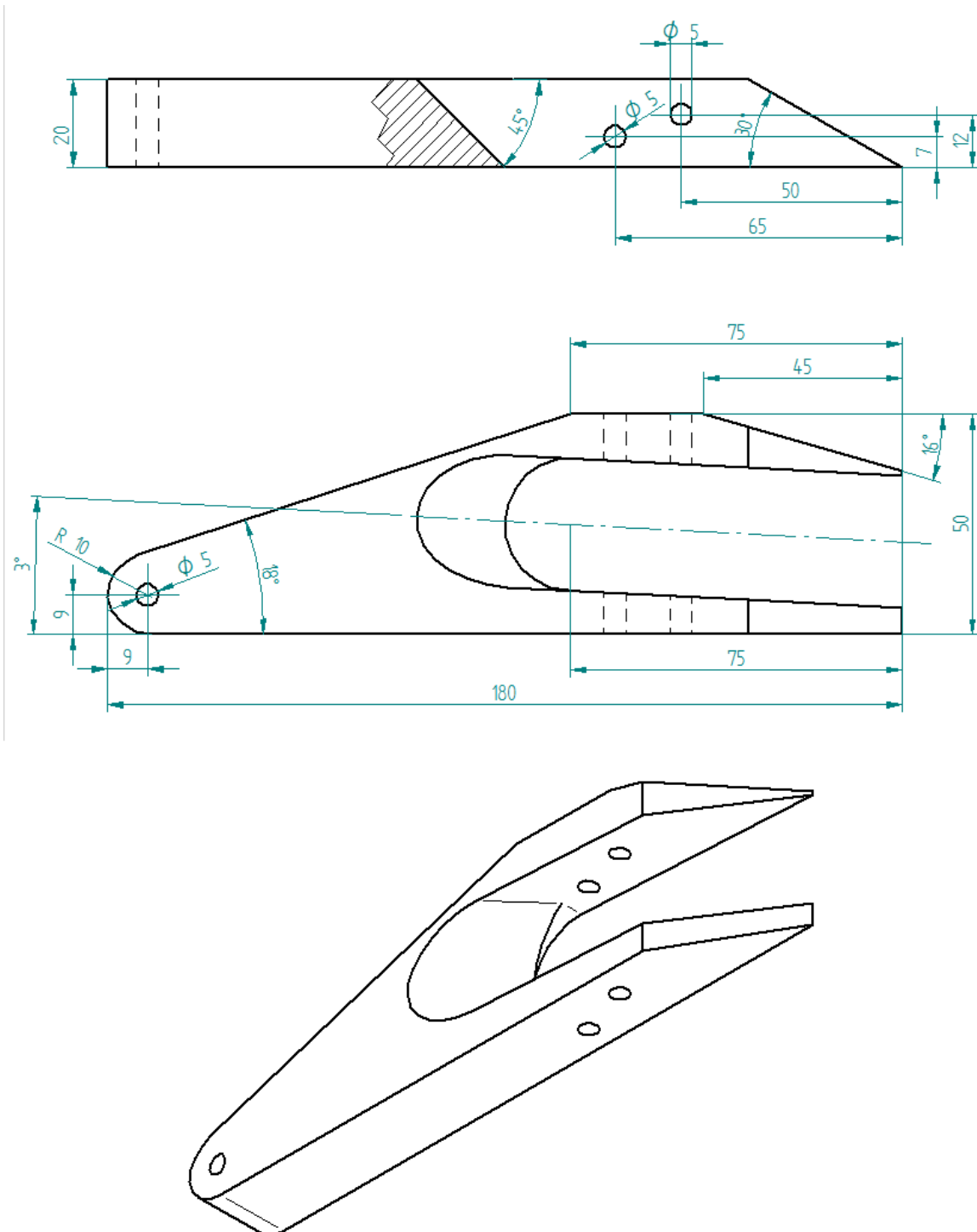
## A.2.15 SPS-Box



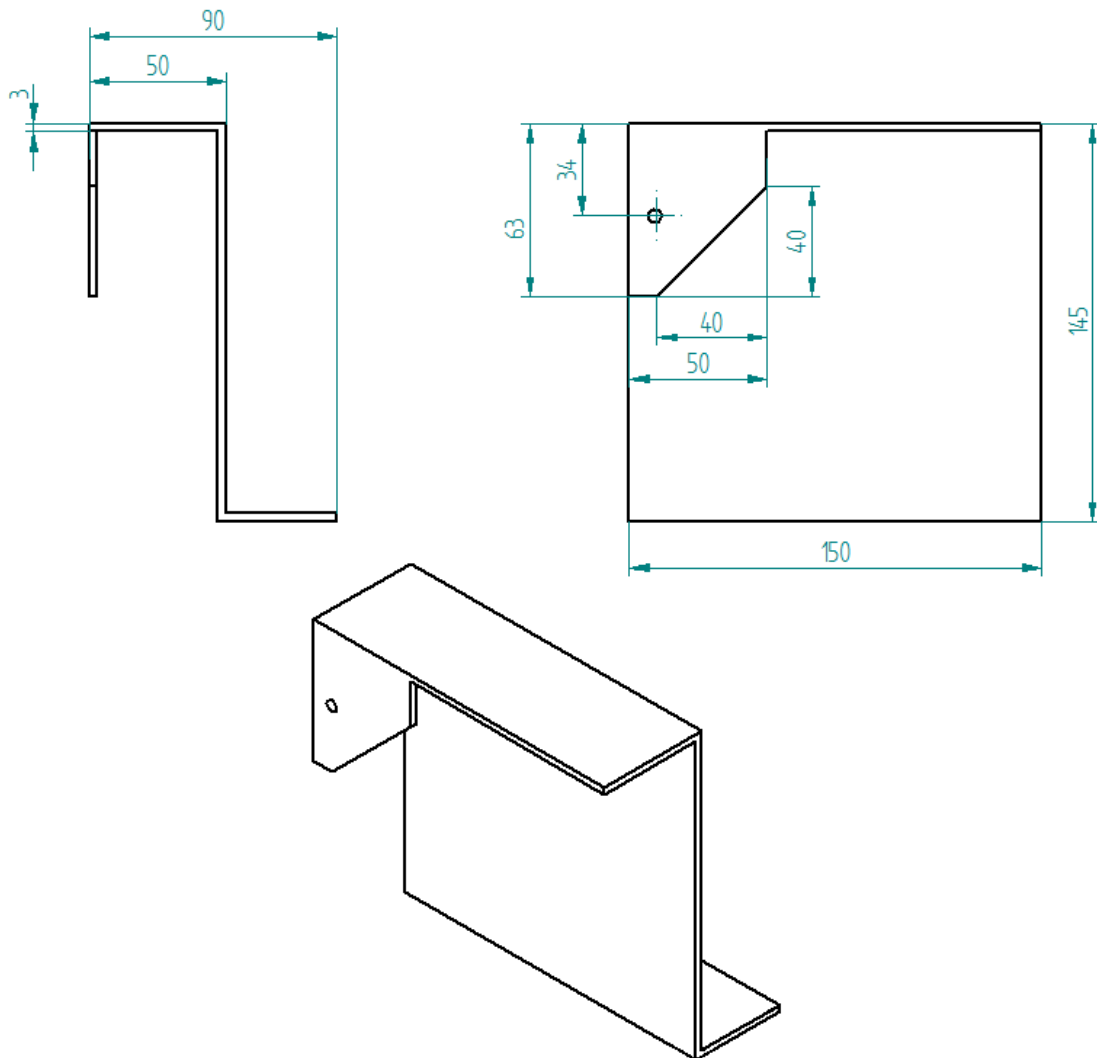
## A.2.16 Adapter Gaspedal



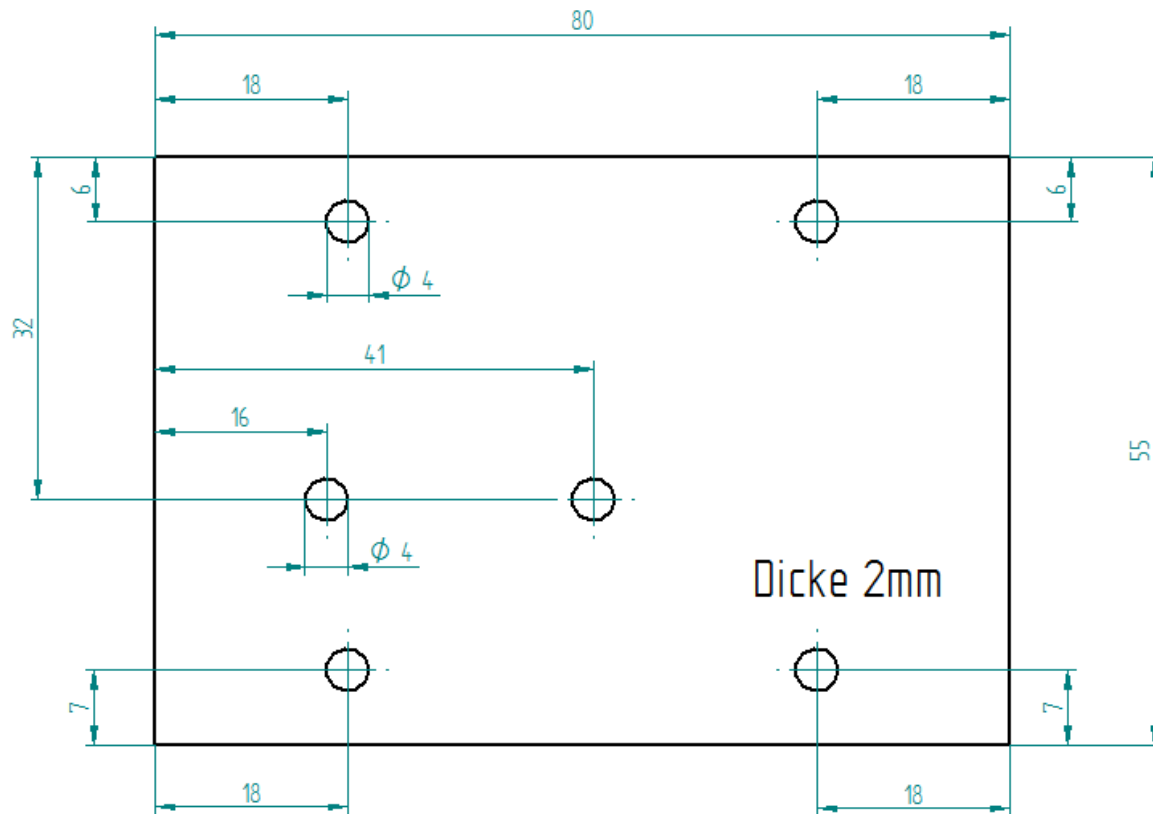
## A.2.17 Adapter Kart



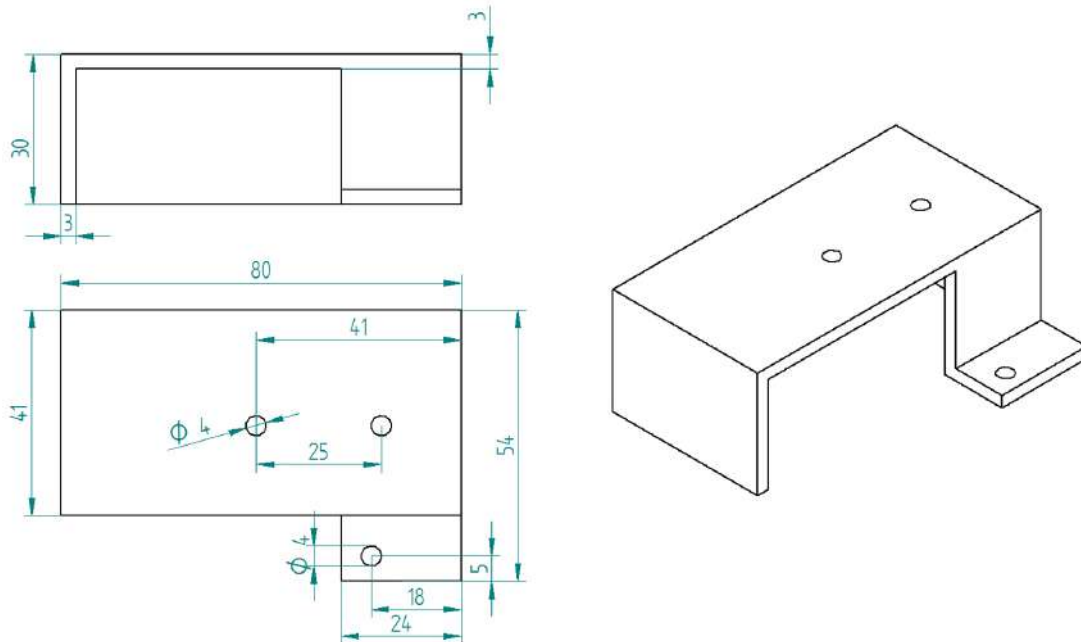
## A.2.18 Schutz Linearpotentiometer



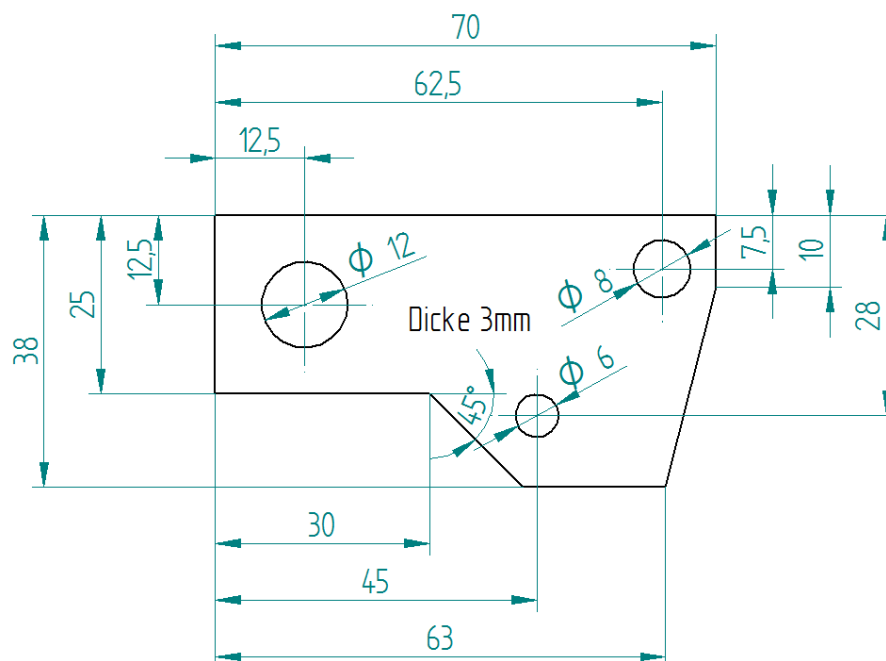
## A.2.19 Adapter Endschalter



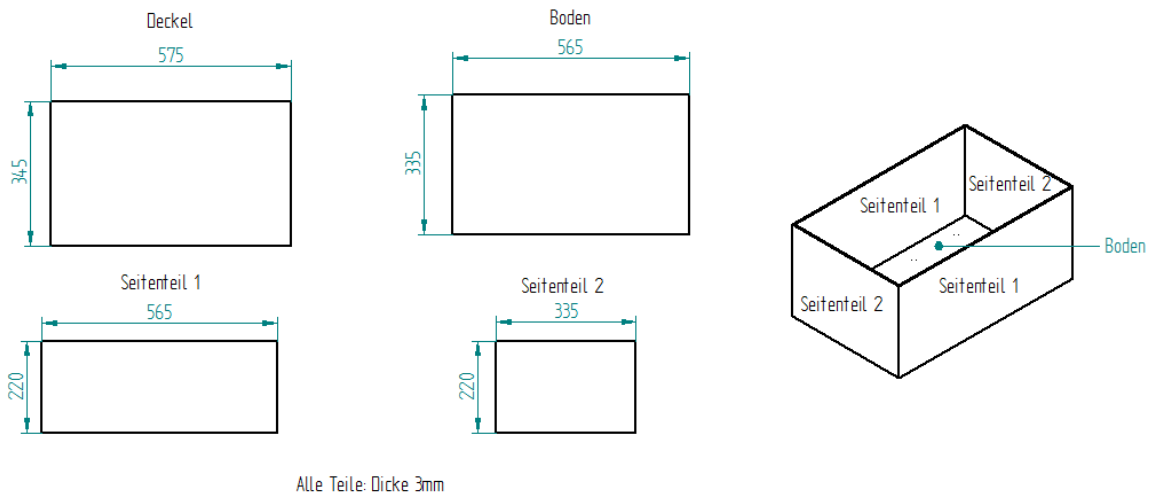
## A.2.20 Abdeckung Endschalter



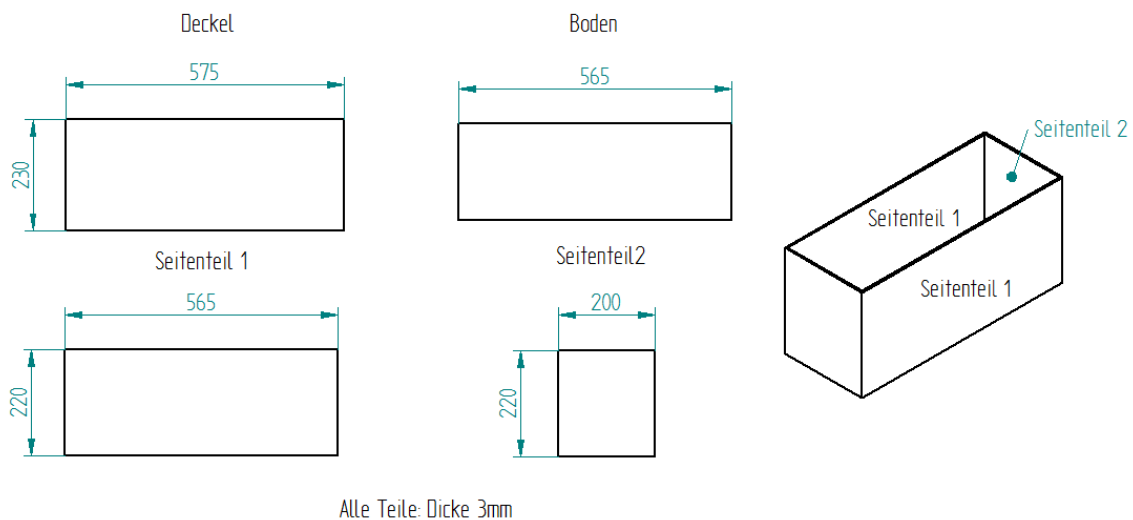
## A.2.21 Blech Induktiver Sensor



## A.2.22 Batteriebox rechts

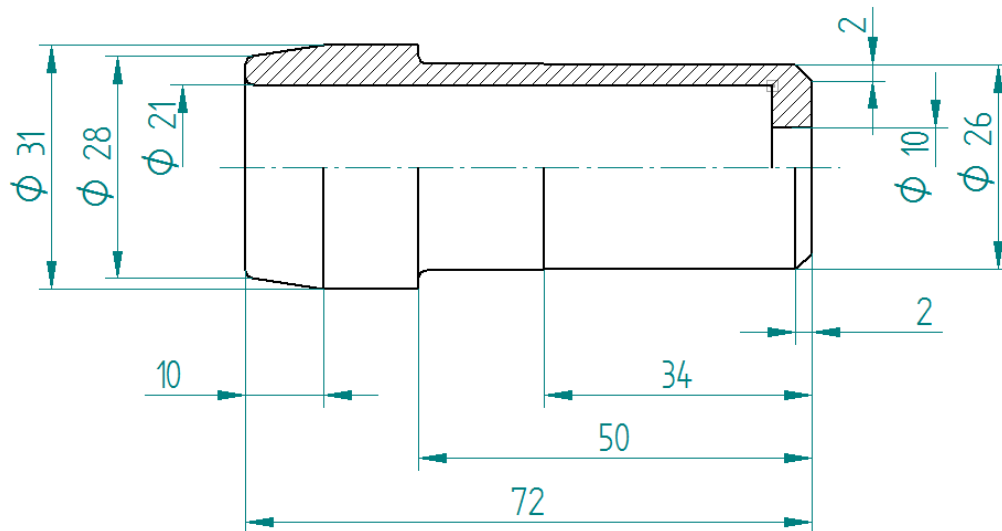


## A.2.23 Batteriebox links

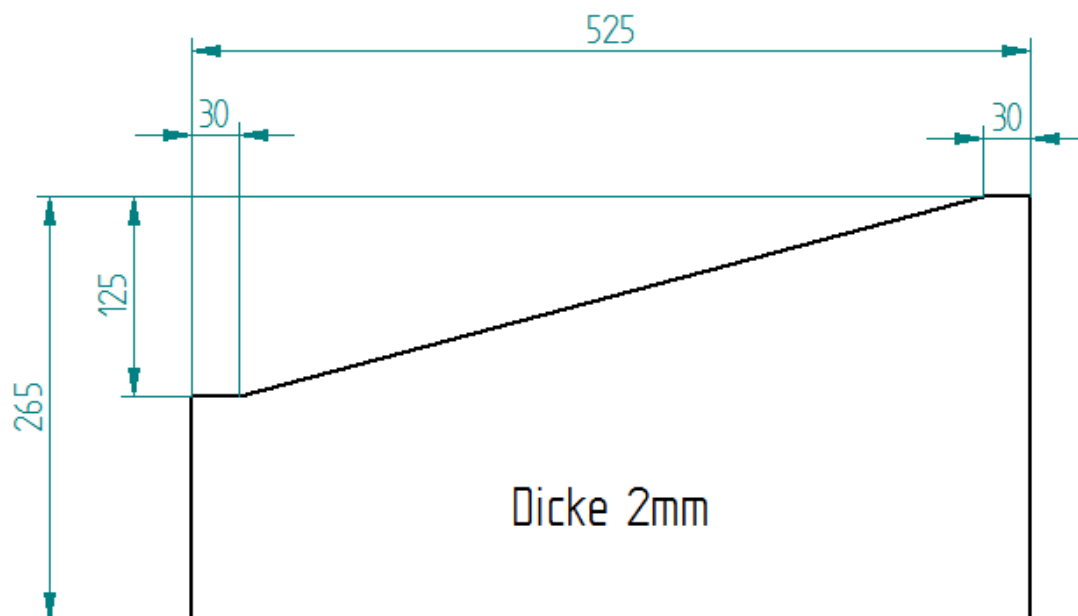


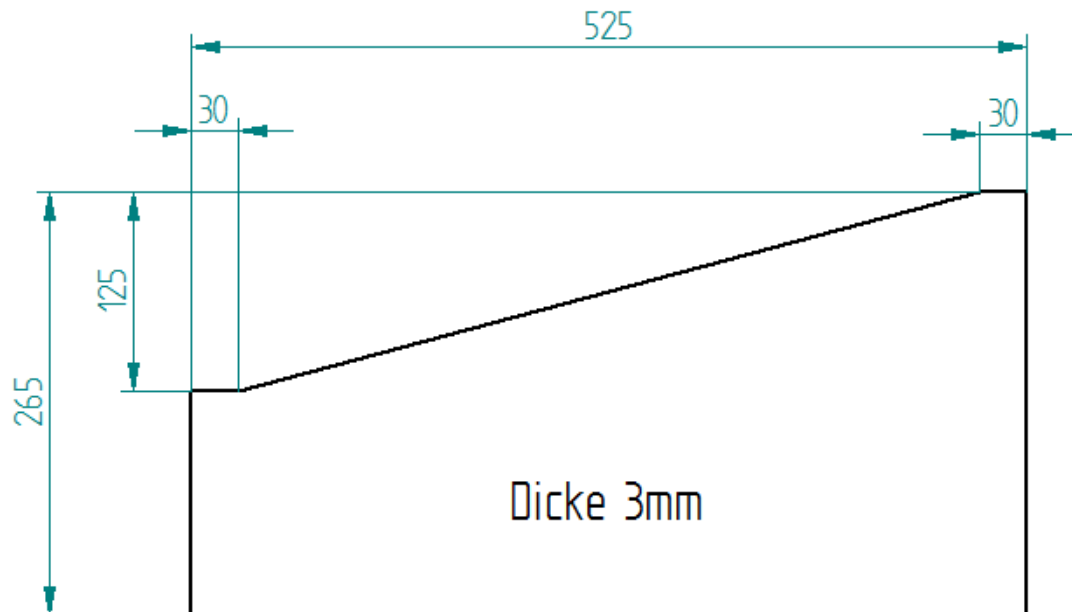


### A.2.24 Aluhülse

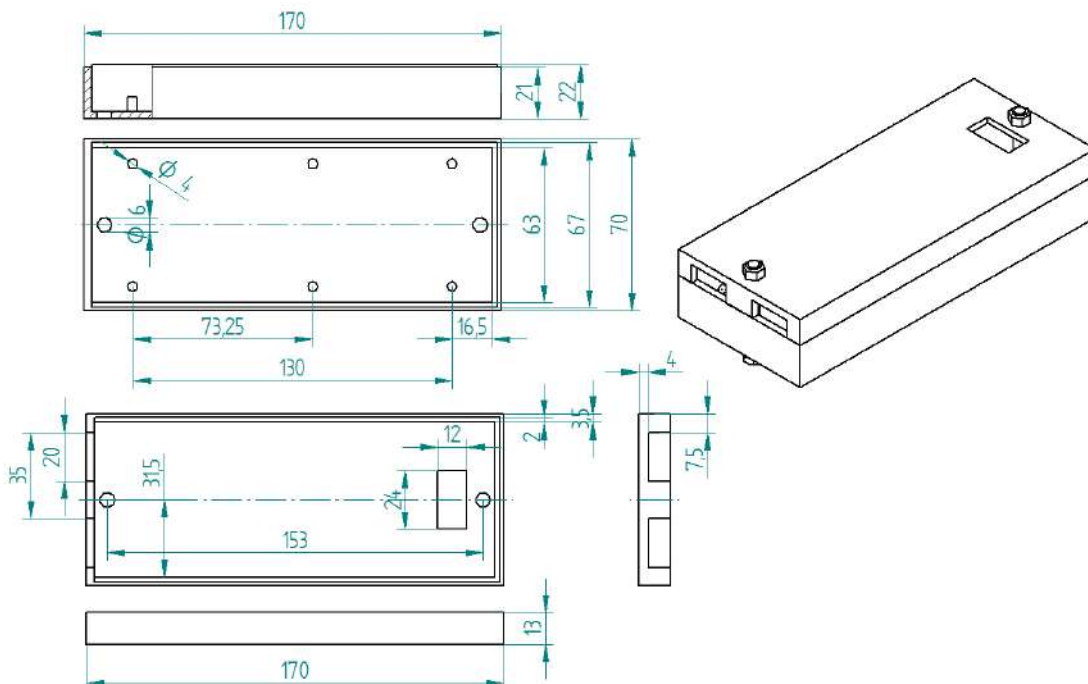


### A.2.25 Unterlagsbleche Batterieboxen





### A.2.26 Isolationsüberwachungssystemhalterung



# Abbildungsverzeichnis

1.1	Logos unserer Sponsoren . . . . .	4
1.2	Zeitplan . . . . .	5
2.1	Blockschaltbild . . . . .	7
2.2	Programm der Safety-SPS . . . . .	10
3.1	Grundmodell Ausgangslage . . . . .	14
3.2	Grundmodell korrigiert und zusammengebaut . . . . .	15
3.3	Brems- und Hydrauliksystem . . . . .	15
3.4	Ausgangsmodell . . . . .	16
3.5	Aufbau der Lenkradeinheit . . . . .	17
3.6	Umbauten der Basis der Antriebseinheit . . . . .	18
3.7	Aufbau der Antriebseinheit . . . . .	19
3.8	Kühlkonzept der Leistungskomponenten . . . . .	19
3.9	Aufbau des Batteriesystems . . . . .	20
3.10	Armaturenbrett . . . . .	21
3.11	Aufbau der SPS . . . . .	21
3.12	Montage des Induktiven Sensors . . . . .	22
3.13	Gaspedaleinheit . . . . .	23
3.14	Endschalter der Bremse . . . . .	23
3.15	Screenshot aus Solid Edge: Konstruktion des Kettenmodells . . . . .	24
3.16	Kettenradantrieb . . . . .	25
3.17	Aufbau der Antriebskomponenten . . . . .	25
3.18	Platzierung der weiteren elektrischen Komponenten . . . . .	26
3.19	Aufbau Isolationsüberwachungssystem . . . . .	26
3.20	Screenshot aus der CAD-Modulationssoftware des Gesamtzusammenbaus . . . . .	27
3.21	Screenshot aus KeyShot: Aufbau Lenkradeinheit . . . . .	29
3.22	Spiegelproblematik bei den Einzelbauteilen . . . . .	30
3.23	Plakat für die BIM . . . . .	32
3.24	Design der Folien für die Batterieboxen . . . . .	34
3.25	Design der Folie für die Vorderverkleidung . . . . .	35
3.26	Design der Folie des Fahrzeugkennzeichens . . . . .	36
3.27	Gegenüberstellung der Farbmodelle . . . . .	36
3.28	Aufstellung aller Statusicons . . . . .	37
3.29	Symbol des Schlüsselschalters . . . . .	37
3.30	Aufstellung der Softkeys . . . . .	38

4.1	3D Ansicht der SPS-Explosionszeichnung . . . . .	52
4.2	Lageplan . . . . .	58
4.3	Controllunit, Quelle: [9] . . . . .	61
4.4	Zellmodule A und B, Quelle: [8] . . . . .	61
4.5	Datenleitung mit CAN Cell Group Module, Quelle: [11] . . . . .	62
4.6	CAN Bus . . . . .	62
4.7	Stromsensor mit Anschluss, Quelle: [12] . . . . .	63
4.8	Aufbau eines Hall Sensors, Quelle: [16] . . . . .	63
4.9	Status Allgemein . . . . .	64
4.10	Konfiguration Batterieblock . . . . .	65
4.11	Anschluss Charger . . . . .	67
4.12	Gesamtkonzept, Quelle: [10] . . . . .	68
4.13	Schaltplan des Steuerteils . . . . .	70
4.14	Platine . . . . .	71
4.15	Schaltplan des Lastteils . . . . .	73
5.1	Verlängerungsstück . . . . .	76
5.2	Hintere Bodenplatte . . . . .	77
5.3	Motorhalterung . . . . .	77
5.4	Gesamtantrieb . . . . .	78
5.5	Kettenspanner . . . . .	80
5.6	Seitenabdeckung . . . . .	81
5.7	Hauptabdeckung . . . . .	82
5.8	Abstandshalter . . . . .	82
5.9	Abdeckungsgehäuse . . . . .	83
5.10	Unterboden . . . . .	84
5.11	Displayhalterung . . . . .	84
5.12	Lenkrad . . . . .	85
5.13	Armaturenbrett . . . . .	86
5.14	Kunststoffgehäuse SPS . . . . .	87
5.15	Gaspedaladapter . . . . .	87
5.16	Adapter zum Kartrahmen . . . . .	88
5.17	Linearpotentiometerschutz . . . . .	89
5.18	Befestigungsblech . . . . .	89
5.19	Halterung der induktiven Sensoren . . . . .	90
5.20	Batterieboxen . . . . .	91
5.21	Wellenhalter . . . . .	92
5.22	Funktionsprinzip des „Fused Deposition Modeling“ Verfahrens . . . . .	93
5.23	3D-Modell der Abdeckung für das Touch Panel . . . . .	94
5.24	3D-Modell der Abdeckung des Endschalters . . . . .	95
5.25	3D-Modell der Rückverkleidung der Lenkradeinheit . . . . .	96
5.26	3D-Modell der Rückverkleidung des Armaturenbretts . . . . .	96
5.27	Funktionsprinzip des „Selektives Lasersintern“ -Verfahrens . . . . .	97
5.28	3D-Modell der Abdeckung für das Isolationsüberwachungssystem . . . . .	98

6.1	Programm der SPS - Übersicht . . . . .	99
6.2	DriveKart Klasse . . . . .	101
6.3	EnableDrive Klasse . . . . .	102
6.4	ASR Klasse . . . . .	103
6.5	TempControl Klasse . . . . .	104
6.6	Programm der Safety - SPS . . . . .	106
6.7	Übersicht DVT . . . . .	108
6.8	DVT-Helper . . . . .	110
6.9	Motorparameter . . . . .	112
6.10	Verknüpfte IOs mit IOMonitor . . . . .	113
6.11	TPDO/RPDO . . . . .	114
6.12	CAN - Einstellungen . . . . .	115
6.13	Upload der Software auf den Controller . . . . .	116
6.14	DisplayETT352, Quelle: [13] . . . . .	117
6.15	Online Configurations . . . . .	119
6.16	Neue IPv4 Adresse . . . . .	119
6.17	Autoexec . . . . .	120
6.18	Platzierte Klassen für die Verbindung . . . . .	121
6.19	Project Settings . . . . .	122
6.20	Screenstruktur . . . . .	123
6.21	Initialisierungsbild . . . . .	124
6.22	Programmtext Bootimage . . . . .	124
6.23	Global Screen . . . . .	125
6.24	Startscreens . . . . .	126
6.25	Fahrscreens . . . . .	127
6.26	Statusscreen . . . . .	128
6.27	Fehleranzeigen . . . . .	130
6.28	Antriebsschlupfregelungsanzeige . . . . .	133
6.29	Fehleranzeige . . . . .	133
6.30	Fake Softkey . . . . .	134
6.31	Variablen und Klasseneinstellungen . . . . .	137
6.32	References to Variables . . . . .	137
6.33	Images Fenster . . . . .	138
6.34	Properties Positionfenster . . . . .	139
6.35	Textlistsfenster . . . . .	140
6.36	Propertiesfenster Texte . . . . .	140
6.37	Fontfenster . . . . .	141
6.38	Functionblock Softkey . . . . .	142
6.39	Functionblock Set . . . . .	142
6.40	NumEdit . . . . .	143
6.41	Bar . . . . .	143
6.42	Checkbit . . . . .	144
7.1	Geplante Abdeckungen (Hinten) . . . . .	147



# Tabellenverzeichnis

3.1	1. Berechnung . . . . .	12
3.2	2. Berechnung . . . . .	13
3.3	Liste der extern Bezogenen Bauteile . . . . .	28
4.1	Technische Daten des Motors[19] . . . . .	41
4.2	Technische Daten[24] . . . . .	44
4.3	Pinbelegung des Steckers . . . . .	45
4.4	Pinbelegung . . . . .	48
4.5	Pinbelegung KL-Module . . . . .	49
4.6	Pinbelegung der digitalen Module . . . . .	50
4.7	Angeschlossene Geräte der Analogmodule . . . . .	51
4.8	Angeschlossene Geräte der Sicherheitsmodule . . . . .	52
4.9	Verwendete Bauteile . . . . .	53
4.10	Technische Daten der Zellen, vgl. [7] . . . . .	60
6.1	Fahrprofile . . . . .	111
6.2	Variablenliste . . . . .	136
6.3	Alarmliste . . . . .	136





# Quellenverzeichnis

- [1] 3,5Zoll Terminal Projektrealisierung, <https://goo.gl/nx2F04>, (Stand: 23.03.2017).
- [2] 3D-Drucken/Vor und Nachteile, <https://goo.gl/4zB888>, (Stand: 23.03.2017).
- [3] Autokennzeichen-Schriftart, <https://goo.gl/Sc5dwN>, (Stand: 23.03.2017).
- [4] Autoschlüssel Kostenlose Icons, <https://goo.gl/77djVI>, (Stand 16.03.17).
- [5] Balancer, <https://goo.gl/qSoMtF>, (Stand: 23.03.2017).
- [6] Übersicht der aktuellen 3D-Druckverfahren, <https://goo.gl/TPLI05>, (Stand: 23.03.2017).
- [7] Daten Akkumulatoren, <http://bit.ly/2n8FmxV>, (Stand: 23.03.2017).
- [8] EMUS BMS Cell Module, <http://bit.ly/2nVu2pZ>, (Stand: 23.03.2017).
- [9] EMUS BMS Controll Unit, <http://bit.ly/2mUxj79>, (Stand: 23.03.2017).
- [10] Emus\_BMS\_User\_Manual\_v.1.0.pdf, Seite 14, (Stand: 23.03.2017).
- [11] Emus\_BMS\_User\_Manual\_v.1.0.pdf, Seite 35, (Stand: 23.03.2017).
- [12] Emus\_BMS\_User\_Manual\_v.1.0.pdf, Seite 37, (Stand: 23.03.2017).
- [13] ETT352 Touchpanel, <https://goo.gl/YsQVfh>, (Stand: 23.03.2017).
- [14] Fahrdynamikregelung, <https://goo.gl/17G4rA>, (Stand 21.03.2017).
- [15] Farbmodelle, <https://goo.gl/7lWXeu>, (Stand 23.03.2017).
- [16] Hallsensor, <http://bit.ly/2mY8Uyw>, (Stand: 23.03.2017).
- [17] Hallsensor, <https://goo.gl/0khVuR>, (Stand: 23.03.2017).
- [18] Lithium-Eisenphosphat-Akkumulator, <https://goo.gl/W8uJhu>, (Stand: 23.03.2017).
- [19] ME1507 PMSM brushless motors, <http://kartmasters.fr/shop/en/synchronous-motors/1817-me1507-pmsm-brushless-motors.html>, (Stand: 01.04.2017).
- [20] Notausschalter, <https://goo.gl/Crf5TP>, (Stand: 18.03.2017).
- [21] Rapid prototyping slicing, <https://goo.gl/U042Q1>, (Stand: 23.03.2017).
- [22] Selective Laser Sintering and Melting, <https://goo.gl/76kPbm>, (Stand: 23.03.2017).



- [23] Rolf Fischer. *Elektrische Maschinen*. München: Hanser Verlag, 2013.
- [24] Sevcon. *Gen4 Product Manual*. Sevcon, 2013.