



UNIVERSITÄT AUGSBURG

Fakultät für Angewandte Informatik

Praktikum für Produktionstechnik

Entwicklungsprozess eines lenkbaren Fahrzeuges TEGGLA

vorgelegt von:	Jonas Wilfert: 1541778 Niklas Paprotta Johannes Evertz: 1463672 Marcel Khodabakhsh: 1333430
eingereicht am:	30.07.2019
Studiengang:	B.Sc. Ingenieurinformatik
Anfertigung am Lehrstuhl:	Produktionsinformatik
	Fakultät für Angewandte Informatik
Verantwortlicher Professor:	Prof. Dr. Ing. Johannes Schilp
Wissenschaftliche Betreuer:	M.Sc. Michael Aumüller M.Sc. Fabian Herzer M.Sc. Shuang Lu M.Sc. Paul Haase

Danksagung

... Danksagung einfügen...

Kurzfassung

... Hier später die eigene deutsche Kurzfassung einfügen...

Dieses Dokument soll als Gerüst für eigene L^AT_EX Dokumente dienen und gleichzeitig Beispiele für häufig verwendete Konstrukte wie Tabellen, Formeln oder Grafiken liefern. Es empfieilt sich, diese Elemente per Cut&Paste zu kopieren und einzufügen.

Abstract

... Please insert your english abstract here...

Inhaltsverzeichnis

1	Problemstellung	1
1.1	Anheben eines Objektes mithilfe von Ultraschall	1
1.2	lalala	1
1.3	Beispiele zur Gliederung: section	2
1.3.1	Unterkapitel subsection	2
1.3.1.1	Unterkapitel subsubsection	2
1.3.2	Ein weiteres Unterkapitel	2
1.4	Formeln	2
1.5	Listen und Aufzählungen	3
1.6	Querverweise	4
2	Vergleich der Technologien	5
2.1	Smartphone als Augmented Reality Gerät	5
3	Software	6
3.1	Installation des Basissystems	6
3.1.1	Windows	6
3.1.2	Linux	7
3.2	Übersetzung	8
3.2.1	<code>latex</code> vs. <code>pdflatex</code>	8
3.2.2	Überblick über die Kommandos	8
3.3	Verwendung von Entwicklungsumgebungen	9
3.3.1	<code>Kile</code> – GNU/Linux	9
3.3.2	<code>TeXnicCenter</code> – Windows	10
3.4	Dateien dieser Formatvorlage	10
4	Allgemeine Hinweise	13
4.1	<code>LATeX</code> -bezogen	13
4.2	Inhaltlich	14

A Quellcode	16
A.1 ESP32	16
A.2 Webpage	29
B Technische Zeichnungen	41
C Software und Konfigurationsdateien	56
C.1 Software A	56
C.2 Software B	56
Literaturverzeichnis	57
Listings	59
Abbildungsverzeichnis	60
Tabellenverzeichnis	61

1 Problemstellung

1.1 Anheben eines Objektes mithilfe von Ultraschall

Durch gezielte Steuerung von mehreren Ultraschall Lautsprechern kann ein Kraftfeld erzeugt werden, durch das leichte Objekte berührungslos in Schweben gehalten werden können. Diese sogenannte akustische Falle resultiert aus der Überlagerung der Schallwellen.

Da Schallwellen natürlich unsichtbar sind entsteht hier das Problem, dass man nicht sieht, wo man Objekte platzieren kann. In dieser Arbeit soll dieses Problem durch die Visualisierung des Schalldruckfelds in Augmented Reality gelöst werden.

1.2 lalala

Bei L^AT_EX schreibt man seinen Text einfach als reinen Text in einem Texteditor seiner Wahl herunter. Umlaute können direkt als „äÄöÖüÜß“ eingegeben werden. Bei Anführungszeichen wird im deutschen zwischen zwei „Versionen“ unterschieden. „Amerikanische“ Anführungszeichen können natürlich ebenfalls verwendet werden.

Absätze mit neuem Einzug werden durch Freilassung einer Zeile im Quelltext erzeugt. Dabei ist es egal, ob man eine oder mehrere Leerzeilen einfügt. Ebenso ist es egal ob man im Text Leerzeichen einstreut, die Zeile bis zum Rand vollschreibt oder nicht. Einen Zeilenumbruch ohne Beginn eines neuen Absatzes

kann man ebenfalls erzwingen, auch wenn dies im Fließtext nicht immer Sinn ergibt. Diverse Textauszeichnungen sind möglich, sollten aber konsistent verwendet werden. So bietet es sich beispielsweise an, ein einheitliches Schema für die Einführung von *Abkürzungen* (Abk.), wie beispielsweise *Personal Computer* (PC), zu verwenden. **Fette Buchstaben** sind bei Bedarf vorhanden, **Schreibmaschinenschrift** eignet sich für die Nennung von Programmnamen. Für URLs bietet sich ein spezielles Kommando an, wie z.B. <http://www.tu-ilmenau.de/kn>.

Literaturverweise setzen eine oder mehrere Literaturdatenbanken voraus. Diese werden als Textdateien mit der Endung **.bib** angelegt und von L^AT_EX verarbeitet. Dies kann

man beispielsweise in dem gut geeigneten Nachschlagewerk [5] nachlesen. Unter [9] findet man ein „Kochbuch“ für L^AT_EX.

Fussnoten sind eine feine Sache, können aber bei zu häufigem Gebrauch nerven¹.

1.3 Beispiele zur Gliederung: section

Kein Text...

1.3.1 Unterkapitel subsection

Kein Text...

1.3.1.1 Unterkapitel subsubsection

Kein Text...

Paragraph Kein Text...

Subparagraph Kein Text...

Ein Unterkapitel ohne laufende Nummer

Es macht nicht immer Sinn ein Kapitel oder Unterkapitel mit einer laufenden Nummer auszustatten. Manchmal soll nur eine Gliederungshilfe eingefügt werden, ohne aber im Inhaltsverzeichnis aufzutreten. Man erreicht dies, indem man ein Sternchen an den Gliederungsbefehl anhängt.

1.3.2 Ein weiteres Unterkapitel

Kein Text...

1.4 Formeln

Formeln sind eine Stärke von L^AT_EX. Sie können einerseits im Fließtext hinterlegt werden, was bei kleinen Formeln wie $E = mc^2$ oder bei $a^2 + b^2 = c^2$ noch gut funktioniert. Bei größeren Formeln und Herleitungen macht es dagegen Sinn, diese abgesetzt vom Text aufzuführen.

¹Praktisch, stört aber den Lesefluss.

$$U = R \cdot I \quad (1.1)$$

$$R = \frac{U}{I} \quad (1.2)$$

Die laufende Nummerierung kann dabei auch unterdrückt werden:

$$A \approx \int_1^{\infty} \frac{1}{x} dx$$

Für mehrzeilige Herleitungen eignet sich auch:

$$(x + y)(x - y) = x^2 - xy + xy - y^2 \quad (1.3)$$

$$= x^2 - y^2 \quad (1.4)$$

$$(x + y)^2 = x^2 + 2xy + y^2 \quad (1.5)$$

1.5 Listen und Aufzählungen

Listen und Aufzählungen braucht man öfters, beispielsweise die so genannten „Bullet“-Listen:

- Erster Punkt
- Zweiter Punkt
- Dritter Punkt
- – Erster Unterpunkt mit Startbullet
 - Zweiter Unterpunkt mit Startbullet
 - Erster Unterpunkt ohne Startbullet
 - Zweiter Unterpunkt ohne Startbullet

Echte Aufzählungen sehen so aus.

1. Erster Punkt
2. Zweiter Punkt

3. Dritter Punkt
4.
 - a) Erster Unterpunkt mit übergeordneter Nummer
 - b) Zweiter Unterpunkt mit übergeordneter Nummer
 - a) Erster Unterpunkt ohne übergeordneter Nummer
 - b) Zweiter Unterpunkt ohne übergeordneter Nummer

Aufzählungen eignen sich auch gut zur Gliederung innerhalb eines Kapitels:

- **Argument A:**

Blah...

Fazit:

Funktioniert, weil ...

- **Argument B:**

Fasel...

Fazit:

Funktioniert nicht, weil ...

Zudem gibt es auch noch die Description-Umgebung:

Schlagwort So kann man einzelne Begriffe der Reihe nach einführen und dabei auch gleich erklären. Nach einem Zeilenunbruch wird eingerückt.

Noch ein Begriff Dabei findet aber keine horizontale Ausrichtung statt.

1.6 Querverweise

Ein Dokument kann Querverweise enthalten. Diese können sich unter anderem auf Grafiken, Tabellen, Formeln oder Absätze beziehen. Der Verweis kann dabei entweder die Nummerierung des Objektes oder dessen Seitenzahl zurückliefern. Der aktuelle Abschnitt lautet beispielsweise 1.6 und beginnt auf Seite 4. Die dazu notwendigen „Anker“ (**labels**) enthalten einen Kenner, welcher zwar frei wählbar ist, aber aus Gründen der Übersicht nach einem einheitlichen Schema konsistent gebildet werden sollte.

Eine Grafik befindet sich beispielsweise in Kapitel 3.3.1, ihre Bezeichnung lautet 3.3 und zu finden ist Sie auf Seite 10. Das Ohmsche Gesetz wird in Formel 1.1 auf Seite 3 wiedergegeben.

2 Vergleich der Technologien

2.1 Smartphone als Augmented Reality Gerät

Als Alternative zu den später verglichenen Augmented Reality Brillen, kann auch ein Smartphone [10] benutzt werden um Virtuelle Objekte in die Realität zu projizieren.

3 Software

Dieses Kapitel zeigt, welche Programme zur Nutzung von L^AT_EX benötigt werden. L^AT_EX ist für alle Plattformen verfügbar, allerdings unterscheiden sich die Paketnamen teilweise.

3.1 Installation des Basissystems

Eine vollständige L^AT_EX-Installation besteht aus mehreren Komponenten. Zusätzlich zur absolut notwendigen Basisinstallation empfiehlt sich noch die Installation diverser Hilfsprogramme, wie z.B. den PDF-Betrachter *Acrobat Reader*, welcher weder Bestandteil des L^AT_EX-Paketes ist noch zum Installationsumfang eines Windowssystems gehört.

3.1.1 Windows

Bei Verwendung von Microsoft Windows werden mehrere Programmkomponenten benötigt, siehe Tabelle 3.1.

Programmname	Aufgabe	Quelle
MiKTeX	L ^A T _E X Distribution, DVI-Betrachter	[6]
Ghostview, Ghostscript	Betrachter für PS-Dateien	[2]
Acrobat Reader	Betrachter für PDF-Dateien	[1]
TeXnicCenter	Entwicklungsumgebung	[7, 8]

Tabelle 3.1: Benötigte Programme unter Windows

MiKTeX ist die L^AT_EX-Distribution für Microsoft Windows. Sie enthält u.a. das Kommando `latex`, mit dessen Hilfe die `.tex` Dateien in `.dvi` Dateien übersetzt werden können. Diese *Device Independent* (DVI) Dateien können bereits mit Hilfe des DVI-Viewers (– TODO –) angezeigt werden. Sinnvoll ist allerdings eine anschließende Wandlung in Postscript (→ `dvips`) oder in ein PDF (– TODO –). „BibTeX“ zur Handhabung von Literaturverzeichnissen, `makeindex` zur Erzeugung von Abkürzungsverzeichnissen sowie `pdflatex` zur direkten Erzeugung von PDF’s aus T_EX-Quellcodes sind ebenfalls bereits in MiKTeX enthalten.

Um Postscript-Dateien unter Windows anzeigen zu können, werden die Programme „Ghostview“ und „Ghostscript“ benötigt. Beide sind unter [2] frei erhältlich.

Zur Anzeige von PDF's, egal ob direkt durch `pdflatex` erzeugt oder durch Wandlung eines `.dvi` bzw. `.ps` entstanden, wird der Adobe Acrobat Reader [1] benötigt.

Zusätzlich wird die integrierte Entwicklungsumgebung „TeXnicCenter“ empfohlen. Sie ist frei unter [7] bzw. [8] verfügbar und bietet eine bequeme Oberfläche zum Umgang mit \LaTeX .

3.1.2 Linux

Die Software für Linuxsysteme ist meist Bestandteil der Basisinstallation. Sollte dennoch eine Komponente (siehe Tabelle 3.2) fehlen, kann sie mit Hilfe des distributionsspezifischen Paketmanagers nachinstalliert werden

Programmname	Aufgabe
<code>TeX</code>	\LaTeX Distribution
<code>xdvi</code>	DVI Betrachter für KDE
<code>kghostview</code>	PS Betrachter für KDE
<code>kpdf</code>	PDF Betrachter für KDE
<code>acroread</code>	Acrobat Reader
<code>kile</code>	\LaTeX Umgebung für KDE [4]
<code>convert</code>	Bildkonverter (ImageMagick) [3]

Tabelle 3.2: Benötigte Programme unter GNU/Linux

Das Paket, welches die \LaTeX Distribution enthält, heißt in der Regel „TeX“ oder auch „LaTeX“. Es enthält ähnlich wie MiKTeX unter Windows sämtliche Programme zur Wandlung von \LaTeX -Quelltexten in `.dvi`'s. Auch hier hat man anschließend die Wahl zwischen `latex` und `pdflatex`.

Die Anzeige von DVI's kann mit Hilfe der Programme `xdvi` oder `kdvi` (unter KDE) erfolgen. Postscriptdokumente können per `gsview` oder `kghostview` (für KDE) angezeigt werden; für PDF's stehen `acroread`, `xpdf`, `kpdf` (für KDE ab 3.4 empfohlen) und `kghostview` (für ältere KDE Versionen) bereit.

Auch für Linux (speziell KDE) gibt es eine sehr schöne Entwicklungsumgebung namens `kile` [4]. Sie erleichtert ähnlich wie TeXnicCenter für Windows die Arbeit mit den Dokumenten.

3.2 Übersetzung

3.2.1 `latex` vs. `pdflatex`

Beide Kommandos sind sich ähnlich, auch wenn sie leicht unterschiedliche Eingabeformate benötigen. Sie wandeln beide `.tex`-Eingabedateien in ein grafisches Ausgabeformat um. Bei Verwendung des Befehles `latex` ist dies eine `.dvi`-Datei, welche anschließend in ein `.ps` oder ein `.pdf` gewandelt werden kann. Die dabei erzeugten PDFs sind von der Qualität her vergleichbar mit der des Outputs von `pdflatex`, welches direkt ein `.pdf` als Ausgabeformat erzeugt. Allerdings unterstützt `pdflatex` diverse PDF-Erweiterungen wie beispielsweise die Möglichkeit, Querverweise als echte Hyperlinks einzubetten.

Ein Unterschied besteht in der Art und Weise, wie einzubindende Grafiken vorliegen müssen. Bei `latex` müssen diese vorher in ein `.eps` (Encapsulated Postscript) gewandelt werden, bei `pdflatex` in ein `.pdf` (Portable Document Format). Wenn man sich an diese Regel hält und sämtliche Grafiken sowohl als `.eps` als auch als `.pdf` ablegt, kann man jederzeit zwischen den beiden Befehlen wählen.

3.2.2 Überblick über die Kommandos

Fehlt noch...

Aufgabe	Linux / Windows
LaTeX Aufruf	<code>latex dokument.tex</code>
PDFLatex Aufruf	<code>pdflatex dokument.tex</code>
BibTeX Aufruf	<code>bibtex dokument</code>
Makeindex Aufruf	<code>makeindex dokument.nlo -s nomenclist -o dokument.nls</code>
Wandlung <code>.dvi</code> → <code>.ps</code>	<code>dvips dokument.dvi -o dokument.ps</code> <code>dvips dokument.dvi</code>
Wandlung <code>.dvi</code> → <code>.pdf</code>	<code>dvipdf dokument.dvi</code> Windows: (?)
Wandlung <code>.ps</code> → <code>.pdf</code>	<code>ps2pdf dokument.ps</code> Windows: (?)
Wandlung Grafik → <code>.eps</code>	<code>convert grafik.jpg grafik.eps</code> ? Evtl. Bildbetrachter, oder auch ImageMagick
Wandlung Grafik → <code>.pdf</code>	<code>convert grafik.jpg grafik.pdf</code> ? Evtl. Bildbetrachter, oder auch ImageMagick

Tabelle 3.3: Kommandos zum manuellen LaTeX-Aufruf

```
makeindex dokument.nlo -s nomenclist -o dokument.nls
bibtex dokument
latex dokument.tex
latex dokument.tex

dvips dokument.dvi -o dokument.ps
ps2pdf dokument.ps
dvipdf dokument.dvi
```

Abbildung 3.1: Komplette Übersetzung mit Hilfe von `latex`

```
makeindex dokument.nlo -s nomenclist -o dokument.nls
bibtex dokument
pdflatex dokument.tex
pdflatex dokument.tex
```

Abbildung 3.2: Komplette Übersetzung mit Hilfe von `pdflatex`

3.3 Verwendung von Entwicklungsumgebungen

Alternativ zur Verwendung eines reinen Texteditors mit manuell zu startendem L^AT_EX-Durchlauf empfiehlt es sich eine integrierte Entwicklungsumgebung zu verwenden. Für GNU/Linux gibt es beispielsweise das Programm **Kile** [4], für Microsoft Windows das Programm **TeXnicCenter** [7, 8]. Beide Programme sind quasi „aufgebohrte“ Texteditoren mit Schaltflächen zum direkten L^AT_EX-Aufruf aus einem Texteditor heraus. Symbole und Tabellen können mit Hilfe von Assistenten ausgewählt und erstellt werden. Die Verwendung einer solchen Entwicklungsumgebung wird empfohlen, macht aber im Endergebnis keinen Unterschied.

3.3.1 Kile – GNU/Linux

Unter Linux, speziell KDE, kann die Entwicklungsumgebung **kile** verwendet werden. Kile organisiert mehrere Teildokumente zu einem Projekt und bietet damit einen einfachen Zugriff auf alle Teildokumente einer Ausarbeitung. Syntaxhighlighting ist ebenfalls vorhanden, sowohl für L^AT_EX als auch für die BibTeX-Literaturdatenbanken. Für den Start eines L^AT_EX-Durchlaufes sowie den verschiedenen Konvertierungsmöglichkeiten gibt es einzelne Knöpfe. Ein direktes Hin- und Herspringen zwischen DVI- und TEX-Ansicht, wobei an die korrekte Stelle gesprungen wird, ist möglich. Dies vereinfacht Korrekturen speziell bei umfangreicherer Dokumenten.

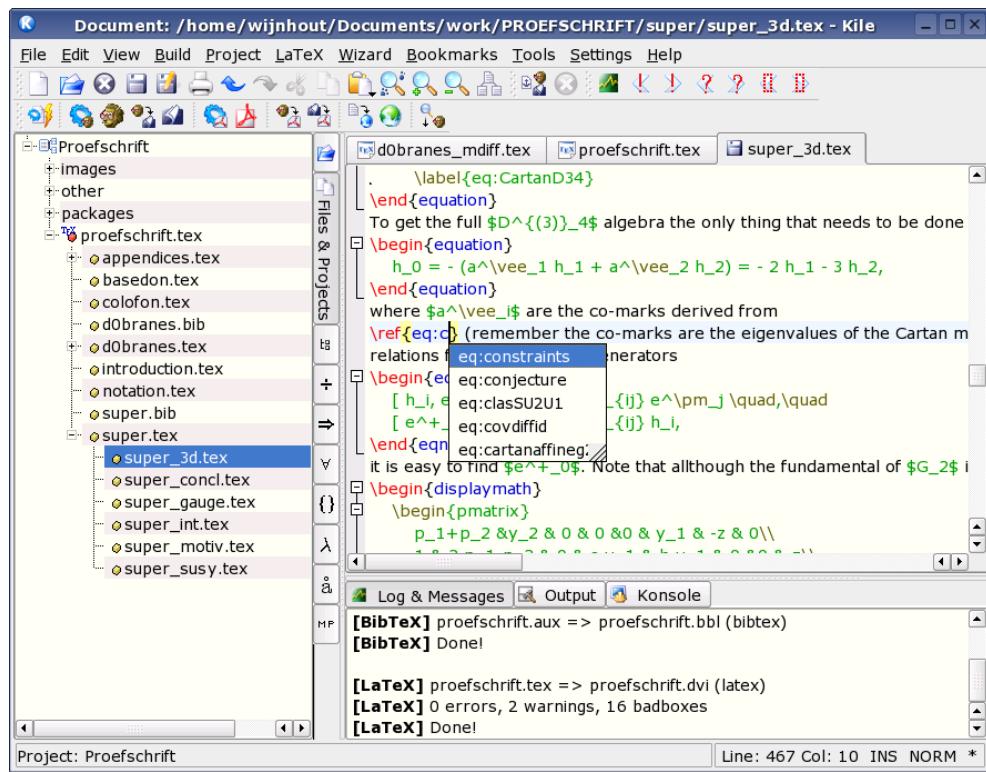


Abbildung 3.3: Bildschirmfoto kile

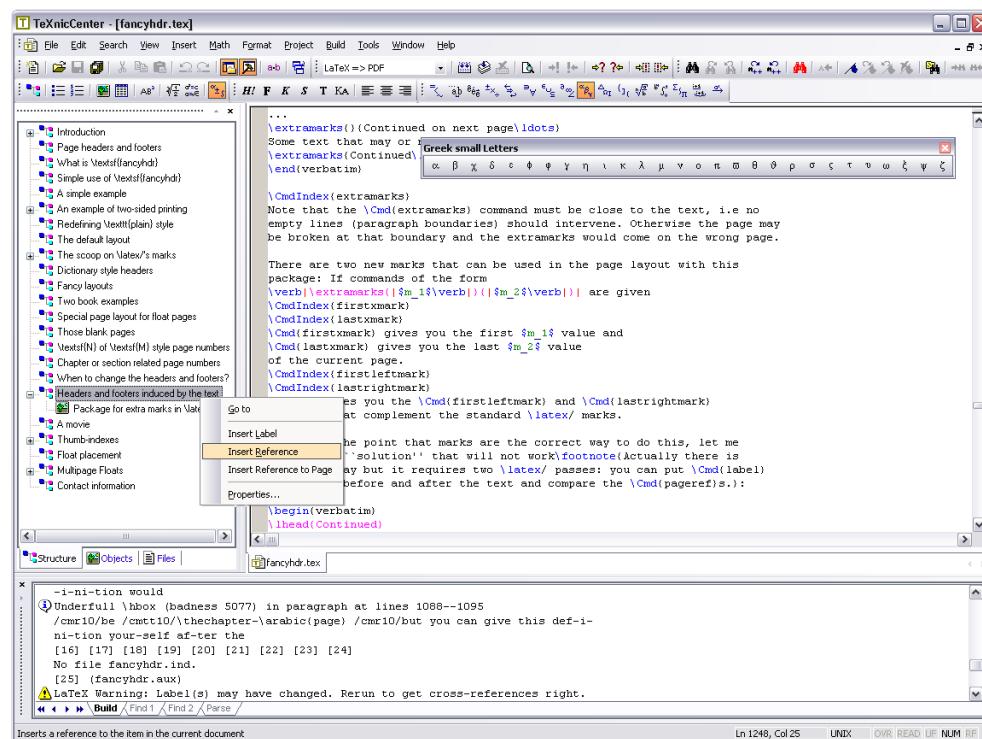
3.3.2 TeXnicCenter – Windows

Ein Bildschirmfoto von TeXnicCenter zeigt Abbildung 3.4.

Der Editor bietet Syntaxhighlighting für die verschiedenen Latexbefehle. Kurz gesagt bietet diese Entwicklungsumgebung die selben Features wie die im letzten Abschnitt vorgestellte Software „Kile.“

3.4 Dateien dieser Formatvorlage

Siehe Tabelle 3.4.

Abbildung 3.4: Bildschirmfoto **texniccenter**

Dateiname	Beschreibung
abschlusserklaerung.tex	L ^A T _E X Teildokument
anhang_abbildungsverzeichnis.tex	L ^A T _E X Teildokument
anhang_abkuerzungsverzeichnis.tex	L ^A T _E X Teildokument
anhang_literaturverzeichnis.tex	L ^A T _E X Teildokument
anhang_messwerte.tex	L ^A T _E X Teildokument
anhang_programm_a.tex	L ^A T _E X Teildokument
anhang_programm_b.tex	L ^A T _E X Teildokument
anhang_protokoll.tex	L ^A T _E X Teildokument
anhang_tabellenverzeichnis.tex	L ^A T _E X Teildokument
bilder/	Hier alle Bilder ablegen!
dokument.dvi	Ergebnis des L ^A T _E X Durchlaufes
dokument.pdf	Erzeugtes PDF-Dokument
dokument.ps	Erzeugtes Postscript-Dokument
dokument.tex	L ^A T _E X Hauptdokument
itmabbrv.bst	Formatvorlage
itmalpha.bst	Formatvorlage
kapitel1.tex	L ^A T _E X Teildokument
kapitel2.tex	L ^A T _E X Teildokument
kapitel3.tex	L ^A T _E X Teildokument
kurzfassung.tex	L ^A T _E X Teildokument
latexvorlage.kilepr	Projektdatei für Kile
latexvorlage.tcp	Projektdatei für TeXnicCenter
literatur.bib	Die Literaturdatenbank
thesen.tex	L ^A T _E X Teildokument
titelblatt.tex	L ^A T _E X Teildokument
vorwort.tex	L ^A T _E X Teildokument

Tabelle 3.4: Relevante Dateien im Paket

4 Allgemeine Hinweise

4.1 L^AT_EX-bezogen

Abkürzungsverzeichnis Sollte das Abkürzungsverzeichnis nach dem Hinzufügen eines `nomenclature`-Kommandos nicht aktualisiert werden, muss der `makeindex`-Aufruf manuell in der Konsole gestartet werden. Manche Entwicklungsumgebungen machen dies aber schon automatisch. Bitte die genannten Parameter nicht vergessen!

Bei Benutzern der GUI Kile kann es vorkommen, dass der `makeindex`-Befehl nicht automatisch ausgeführt wird, scheint ein Bug zu sein. In diesem Fall kann der Index auch manuell durch Aufruf von `makeindex` aktualisiert werden.

Thesenpapier Für die Thesen wurde mit der Version 0.8 an ein eigenständiges Dokument namens `thesen-handout.tex` hinzugefügt. Es bindet ebenso wie das Hauptdokument die Datei `thesen.tex` ein, erzeugt aber eben nur dieses eine Blatt ohne eine Seitenzahl.

Beidseitiger Druck Im Zentraldokument `dokument.tex` kann das Layout auf doppelseitigen Druck umgeschaltet werden (Option `twoside` statt `oneside`). Allerdings verlangen manche Prüfungsämter explizit einen einseitigen Druck! Neue Kapitel (`chapter`) beginnen dabei automatisch auf einer Vorderseite (→ rechte Seite). Die Ränder sind dabei innen nur halb so breit wie außen, was aber Absicht ist: Zusammen ergeben die linke und die rechte Seite innen einen „weißen Streifen“, der genauso breit ist wie die äußeren Ränder.

Überlange Kapitelüberschriften Manchmal müssen Überschriften sehr lang sein, so dass sie von L^AT_EX umgebrochen werden. Dieses Verhalten ist aber weder im Inhaltsverzeichnis noch in der Kopfzeile erwünscht! Daher kann man zu einer überlangen Überschrift auch eine Kurzform mit angeben, welche dann im Inhaltsverzeichnis und im Dokumentenkopf verwendet wird:

```
\chapter[Kurzform]{Langform}
```

Einzüge Bitte *nicht!* die Einzüge ändern oder abschalten. Das ist so gewollt und verbessert den Lesefluss! (Stichwort `\setlength{\parindent{0pt}}!`)!

BibTeX-Einträge mit mehreren Autoren Sollen mehrere Autoren angegeben werden, so sind diese einzeln als *Vorname Nachname* anzugeben und durch `and` voneinander zu trennen. BibTeX ersetzt das `and` dann durch das deutsche „und“:
`author = {Adam Riese and Eva Zwerg},`

4.2 Inhaltlich

- Überschriften im Inhaltsverzeichnis nie tiefer als vier Ebenen. Dies geht mit L^AT_EX auch gar nicht anders, da `subsubsection` bereits die niedrigste Schachtelungstiefe darstellt, welche noch im Inhaltsverzeichnis aufgeführt wird.
- Die Kapitel sollten in der späteren Ausarbeitung anders benannt werden als in dieser Formatvorlage. Eine Diplomarbeit *kann* beispielsweise aus der folgenden Aufteilung bestehen:
 1. Problemstellung
 2. Theoretische Grundlagen
 3. Herleitung
 4. Der Prototyp
 5. Zusammenfassung
 6. Ausblick
- Eine Inventarisierungsnummer ist nicht bei jeder Art der Ausarbeitung gegeben. Dazu kann im Zentraldokument (→ `dokument.tex`) eine leere Zeichenfolge hinterlegt werden.
- Es empfiehlt sich, ein Programm zur Rechtschreibprüfung zu installieren. Alternativ zu einer L^AT_EX-fähigen Rechtschreibkorrektursoftware kann ein Abschnitt auch in bspw. Microsoft Word getippt und geprüft werden, bevor er dann in das L^AT_EX-Dokument eingefügt wird.
- Für Diplomarbeiten wird generell ein englischer „Abstract“ benötigt!

Anhang

A Quellcode

A.1 ESP32

```
1 #include "communication.h"
2
3 AsyncWebSocketClient *ws_client = nullptr;
4
5 void Communication::onWSData(AsyncWebSocket *server, AsyncWebSocketClient *client,
6     AwsEventType type, uint8_t *data, size_t len) {
7     ws_client = client;
8
9     int16_t *d16 = (int16_t *)data;
10    int16_t id = d16[0];
11
12    switch (id) {
13        case OmniMessageType::DRIVE:
14            onDrive(d16[1], d16[2], d16[3], d16[4]);
15            break;
16
17        default:
18            Serial.printf("No such command %i\n", id);
19            break;
20    }
21
22 void Communication::onDrive(int16_t x1, int16_t y1, int16_t x2, int16_t y2) {
23     // Serial.printf("Driving with speed: %i, %i and %i, %i now\n", x1, y1, x2,
24     // y2);
25     Movement::drive(y1, x1, x2);
26 }
27
28 void Communication::sendCurrGyro(XYZ speed, XYZ accel, XYZ rot) {
29     int16_t buff[10] = {OmniMessageType::CURR_GYRO, speed.x, speed.y, speed.z,
30     accel.x, accel.y, accel.z, rot.x, rot.y, rot.z};
31     ws->binaryAll((uint8_t *)buff, 20);
```

```

30 }
31
32 void Communication::sendCurrGyBatComb(int16_t sX, int16_t sY, int16_t sZ, int16_t
33     aX, int16_t aY, int16_t aZ, int16_t rX, int16_t rY, int16_t rZ, int16_t bat,
34     int16_t temp) {
35     int16_t buff[12] = {OmniMessageType::CURR_GY_BAT_COMB, sX, sY, sZ, aX, aY, aZ,
36     rX, rY, rZ, bat, temp};
37     ws->binaryAll((uint8_t *)buff, 24);
38 }
39
40 void Communication::sendCurrMotor(int16_t vl, int16_t vr, int16_t hl, int16_t hr) {
41     int16_t buff[5] = {OmniMessageType::CURR_MOTOR, vl, vr, hl, hr};
42     // Serial.println("should send now");
43     ws->binaryAll((uint8_t *)buff, 10);
44 }
45
46 void Communication::sendCurrBattery(int16_t cell1, int16_t cell2) {
47     int16_t buff[5] = {OmniMessageType::CURR_BATTERY, cell1, cell2};
48     // Serial.println("should send now");
49     ws->binaryAll((uint8_t *)buff, 10);
50 }
51
52 AsyncWebSocket *Communication::ws;

```

Listing A.1: communication.cpp

```

1 #pragma once
2
3 #include "movement.h"
4 #include <ESPAsyncWebServer.h>
5
6 /**
7 * Limitations due to js:
8 * Buffer has to be all the same type
9 *
10 * Type is int16 as well at the start of the array
11 *
12 */
13 enum OmniMessageType {
14     DRIVE = 0,           //L4 {x1} int16 [-1023, 1023] :: {y1} int16 [-1023,
15     1023] :: {x2} int16 [-1023, 1023] :: {y2} int16 [-1023, 1023]

```

```

15     CURR_GYRO = 1,           //L18 {speed m/s} (x) int16 (y) int16 (z) int16 :: 
16     {accel m/s^2} (x) int16 (y) int16 (z) int16 :: {rot} (x) int16 (y) int16 (z)
17     int16
18     CURR_BATTERY = 2,       //L1 {cell1} int16 [0, 2^12]
19     CURR_MOTOR = 3,         //L4 {vl} int16 [-1023, 1023] :: {vr} int16 [-1023,
20     1023] :: {hl} int16 [-1023, 1023] :: {hr} int16 [-1023, 1023]
21     CURR_GY_BAT_COMB = 4,   //L20 {speed m/s} (x) int16 (y) int16 (z) int16 :: 
22     {accel m/s^2} (x) int16 (y) int16 (z) int16 :: {rot} (x) int16 (y) int16 (z)
23     int16 :: {bat} int16 [0, 2^12] :: {temp} int16
24 };
25
26
27 struct XYZ {
28     int16_t x;
29     int16_t y;
30     int16_t z;
31 };
32
33
34 class Communication {
35 public:
36     static void onWSData(AsyncWebSocket *server, AsyncWebSocketClient *client,
37     AwsEventType type, uint8_t *data, size_t len);
38     static void onDrive(int16_t x1, int16_t y1, int16_t x2, int16_t y2);
39     static void sendCurrGyro(XYZ speed, XYZ accel, XYZ rot);
40     static void sendCurrBattery(int16_t cell1, int16_t cell2);
41     static void sendCurrGyBatComb(int16_t sX, int16_t sY, int16_t sZ, int16_t aX,
42     int16_t aY, int16_t aZ, int16_t rX, int16_t rY, int16_t rZ, int16_t bat,
43     int16_t temp);
44     static void sendCurrMotor(int16_t vl, int16_t vr, int16_t hl, int16_t hr);
45
46     static AsyncWebSocket *ws;
47 };

```

Listing A.2: communication.h

```

1 #include "movement.h"
2
3 // -----
4 // ----- Motor Class
5 // -----
6 // ##### Functions #####
7 void Motor::setSpeed(int16_t speed) {
8     // disable before changing direction
9     // (probably not needed) safety to prevent shoutthrough

```

```
8     ledcWrite(channel, 0);
9     if (speed > 0) {
10         digitalWrite(pin_dir1, LOW);
11         digitalWrite(pin_dir2, HIGH);
12     } else if (speed < 0) {
13         digitalWrite(pin_dir1, HIGH);
14         digitalWrite(pin_dir2, LOW);
15     } else {
16         digitalWrite(pin_dir1, LOW);
17         digitalWrite(pin_dir2, LOW);
18         return;
19     }
20
21     ledcWrite(channel, abs(speed));
22 }
23
24 void Motor::stop() {
25     ledcWrite(channel, 0);
26     digitalWrite(pin_dir1, LOW);
27     digitalWrite(pin_dir2, LOW);
28 }
29
30 // ----- Movement Class
31 // ##### Attributes #####
32 Motor Movement::MOTOR_VL = Motor(CH_MOTOR_VL, PIN_MOTOR_VL_DIR1,
33                             PIN_MOTOR_VL_DIR2);
34 Motor Movement::MOTOR_VR = Motor(CH_MOTOR_VR, PIN_MOTOR_VR_DIR1,
35                             PIN_MOTOR_VR_DIR2);
36 Motor Movement::MOTOR_HL = Motor(CH_MOTOR_HL, PIN_MOTOR_HL_DIR1,
37                             PIN_MOTOR_HL_DIR2);
38 Motor Movement::MOTOR_HR = Motor(CH_MOTOR_HR, PIN_MOTOR_HR_DIR1,
39                             PIN_MOTOR_HR_DIR2);
40
41 // ##### Functions #####
42 void Movement::initPWM() {
43     Serial.println("Initing PWMs");
44
45     // Setting all pins to output
46     pinMode(PIN_LED, OUTPUT);
47     pinMode(PIN_MOTOR_VL_EN, OUTPUT);
48     pinMode(PIN_MOTOR_VL_DIR1, OUTPUT);
49     pinMode(PIN_MOTOR_VL_DIR2, OUTPUT);
```

```
46
47     pinMode(PIN_MOTOR_VR_EN, OUTPUT);
48     pinMode(PIN_MOTOR_VR_DIR1, OUTPUT);
49     pinMode(PIN_MOTOR_VR_DIR2, OUTPUT);
50
51     pinMode(PIN_MOTOR_HL_EN, OUTPUT);
52     pinMode(PIN_MOTOR_HL_DIR1, OUTPUT);
53     pinMode(PIN_MOTOR_HL_DIR2, OUTPUT);
54
55     pinMode(PIN_MOTOR_HR_EN, OUTPUT);
56     pinMode(PIN_MOTOR_HR_DIR1, OUTPUT);
57     pinMode(PIN_MOTOR_HR_DIR2, OUTPUT);
58
59 // Disable all pins by default to not possibly cause shootthrough
60 digitalWrite(PIN_MOTOR_VL_EN, LOW);
61 digitalWrite(PIN_MOTOR_VL_DIR1, LOW);
62 digitalWrite(PIN_MOTOR_VL_DIR2, LOW);
63
64 digitalWrite(PIN_MOTOR_VR_EN, LOW);
65 digitalWrite(PIN_MOTOR_VR_DIR1, LOW);
66 digitalWrite(PIN_MOTOR_VR_DIR2, LOW);
67
68 digitalWrite(PIN_MOTOR_HL_EN, LOW);
69 digitalWrite(PIN_MOTOR_HL_DIR1, LOW);
70 digitalWrite(PIN_MOTOR_HL_DIR2, LOW);
71
72 digitalWrite(PIN_MOTOR_HR_EN, LOW);
73 digitalWrite(PIN_MOTOR_HR_DIR1, LOW);
74 digitalWrite(PIN_MOTOR_HR_DIR2, LOW);
75
76 // configure LED PWM functionalitites
77 ledcSetup(CH_LED, PWM_FREQ, PWM_RESOLUTION);
78 ledcSetup(CH_MOTOR_VL, PWM_FREQ, PWM_RESOLUTION);
79 ledcSetup(CH_MOTOR_VR, PWM_FREQ, PWM_RESOLUTION);
80 ledcSetup(CH_MOTOR_HL, PWM_FREQ, PWM_RESOLUTION);
81 ledcSetup(CH_MOTOR_HR, PWM_FREQ, PWM_RESOLUTION);
82
83 // attach the channel to the GPIO2 to be controlled
84 ledcAttachPin(PIN_LED, CH_LED);
85 ledcAttachPin(PIN_MOTOR_VL_EN, CH_MOTOR_VL);
86 ledcAttachPin(PIN_MOTOR_VR_EN, CH_MOTOR_VR);
87 ledcAttachPin(PIN_MOTOR_HL_EN, CH_MOTOR_HL);
88 ledcAttachPin(PIN_MOTOR_HR_EN, CH_MOTOR_HR);
```

```
89     Serial.println("Initiated PWMs");
90 }
91
92
93 void Movement::drive(int controlFront, int controlSide, int controlTurn) {
94     if (abs(controlFront) < CONTROLLER_LOWER_LIMIT && abs(controlSide) <
95         CONTROLLER_LOWER_LIMIT && abs(controlTurn) < CONTROLLER_LOWER_LIMIT) {
96         // Serial.println("Stopping Motors");
97         MOTOR_VL.stop();
98         MOTOR_VR.stop();
99         MOTOR_HL.stop();
100        MOTOR_HR.stop();
101
102        Communication::sendCurrMotor(0, 0, 0, 0);
103        return;
104    }
105
106    int speedVL = 0;
107    int speedVR = 0;
108    int speedHL = 0;
109    int speedHR = 0;
110
111    if (abs(controlFront) > 0 && controlSide == 0 && controlTurn == 0) {
112        speedVL = controlFront;
113        speedVR = controlFront;
114        speedHL = controlFront;
115        speedHR = controlFront;
116    } else if (abs(controlSide) > 0 && controlFront == 0 && controlTurn == 0) {
117        speedVL = controlSide;
118        speedVR = -controlSide;
119        speedHL = -controlSide;
120        speedHR = controlSide;
121    } else if (controlSide == 0 && controlFront == 0 && abs(controlTurn) > 0) {
122        speedVL = controlTurn;
123        speedVR = -controlTurn;
124        speedHL = controlTurn;
125        speedHR = -controlTurn;
126    } else {
127        driveAlgorithm(controlFront, controlSide, controlTurn, &speedVL, &speedVR,
128                      &speedHL, &speedHR);
129    }
130}
```

```

129     speedVL = speedVL / 1023.0 * USEABLE_UPPER_LIMIT + (1023 -
130     USEABLE_UPPER_LIMIT) * sgn(speedVL);
131     speedVR = speedVR / 1023.0 * USEABLE_UPPER_LIMIT + (1023 -
132     USEABLE_UPPER_LIMIT) * sgn(speedVR);
133     speedHL = speedHL / 1023.0 * USEABLE_UPPER_LIMIT + (1023 -
134     USEABLE_UPPER_LIMIT) * sgn(speedHL);
135     speedHR = speedHR / 1023.0 * USEABLE_UPPER_LIMIT + (1023 -
136     USEABLE_UPPER_LIMIT) * sgn(speedHR);
137
138
139     MOTOR_VL.setSpeed(speedVL);
140     MOTOR_VR.setSpeed(speedVR);
141     MOTOR_HL.setSpeed(speedHL);
142     MOTOR_HR.setSpeed(speedHR);
143
144
145     Communication::sendCurrMotor(speedVL, speedVR, speedHL, speedHR);
146 }
147
148 void Movement::driveAlgorithm(int controlFront, int controlSide, int controlTurn,
149     int *speedVL, int *speedVR, int *speedHL, int *speedHR) {
150     double phi = atan2(controlSide, controlFront);
151
152     int vd = min((int)sqrt(controlFront * controlFront + controlSide *
153     controlSide), 1023);
154     vd -= controlTurn / 2;
155     int vphi = controlTurn / 2;
156
157     double s = vd * sin(phi + PI / 4);
158     double c = vd * cos(phi + PI / 4);
159
160     *speedVL = s + vphi;
161     *speedVR = c - vphi;
162     *speedHL = c + vphi;
163     *speedHR = s - vphi;
164 }
```

Listing A.3: movement.cpp

```

1 #pragma once
2
3 #include "communication.h"
4 #include "util.h"
5 #include <Arduino.h>
```

```
6 #include <math.h>
7
8 class Motor {
9 private:
10     char channel;
11     char pin_dir1;
12     char pin_dir2;
13
14 public:
15     Motor(char channel, char pin_dir1, char pin_dir2) : channel(channel),
16         pin_dir1(pin_dir1), pin_dir2(pin_dir2){};
17     ~Motor() {}
18
19     /**
20      * speed in [-1023; 1023]
21      */
22     void setSpeed(int16_t speed);
23
24     void stop();
25 };
26
27 class Movement {
28 private:
29     /* data */
30 public:
31     static const char PIN_LED = 2;
32
33     static Motor MOTOR_VL;
34     static Motor MOTOR_VR;
35     static Motor MOTOR_HL;
36     static Motor MOTOR_HR;
37
38     /** Amount of values down from 1023 which can be used */
39     static const int USEABLE_UPPER_LIMIT = 700;
40     /** Wont move below this limit */
41     static const int CONTROLLER_LOWER_LIMIT = 50;
42
43     static const char PIN_MOTOR_VL_EN = 23;
44     static const char PIN_MOTOR_VL_DIR1 = 18;
45     static const char PIN_MOTOR_VL_DIR2 = 19;
46
47     static const char PIN_MOTOR_VR_EN = 32;
48     static const char PIN_MOTOR_VR_DIR1 = 12;
```

```

48     static const char PIN_MOTOR_VR_DIR2 = 13;
49
50     static const char PIN_MOTOR_HL_EN = 16;
51     static const char PIN_MOTOR_HL_DIR1 = 17;
52     static const char PIN_MOTOR_HL_DIR2 = 5;
53
54     static const char PIN_MOTOR_HR_EN = 4;
55     static const char PIN_MOTOR_HR_DIR1 = 3;
56     static const char PIN_MOTOR_HR_DIR2 = 15;
57
58     static const int PWM_FREQ = 500;
59     static const char CH_LED = 0;
60     static const char CH_MOTOR_VL = 1;
61     static const char CH_MOTOR_VR = 2;
62     static const char CH_MOTOR_HL = 3;
63     static const char CH_MOTOR_HR = 4;
64     static const char PWM_RESOLUTION = 10; //Resolution 8, 10, 12, 15
65
66     static void initPWM();
67
68     static void driveMotor(char channel, char dir1, char dir2, int speed);
69
70 /**
71  * Converts the given controls into wheel speeds
72  * Algorithm source:
73  * http://eprints.utm.edu.my/16543/1/Omni%20Directional%20Control%20Algorithm%20For%20Mecanu
74  */
75
76     static void drive(int controlFront, int controlSide, int controlTurn);
77
78 private:
79     static void driveAlgorithm(int controlFront, int controlSide, int controlTurn,
80     int *speedVL, int *speedVR, int *speedHL, int *speedHR);
81 };

```

Listing A.4: movement.h

```

1 #include "util.h"
2
3 ****
4 * high precision sine/cosine
5 *
6 * Source: https://gist.github.com/geraldyeo/988116

```

```
7  *
8  *
9  ****
10 void cossin(float x, float *outCos, float *outSin) {
11     float sin, cos;
12
13     //always wrap input angle to -PI..PI
14     if (x < -3.14159265)
15         x += 6.28318531;
16     else if (x > 3.14159265)
17         x -= 6.28318531;
18
19     //compute sine
20     if (x < 0) {
21         sin = 1.27323954 * x + .405284735 * x * x;
22
23         if (sin < 0)
24             sin = .225 * (sin * -sin - sin) + sin;
25         else
26             sin = .225 * (sin * sin - sin) + sin;
27     } else {
28         sin = 1.27323954 * x - 0.405284735 * x * x;
29
30         if (sin < 0)
31             sin = .225 * (sin * -sin - sin) + sin;
32         else
33             sin = .225 * (sin * sin - sin) + sin;
34     }
35
36     //compute cosine: sin(x + PI/2) = cos(x)
37     x += 1.57079632;
38     if (x > 3.14159265)
39         x -= 6.28318531;
40
41     if (x < 0) {
42         cos = 1.27323954 * x + 0.405284735 * x * x;
43
44         if (cos < 0)
45             cos = .225 * (cos * -cos - cos) + cos;
46         else
47             cos = .225 * (cos * cos - cos) + cos;
48     } else {
49         cos = 1.27323954 * x - 0.405284735 * x * x;
```

```

50
51     if (cos < 0)
52         cos = .225 * (cos * -cos - cos) + cos;
53     else
54         cos = .225 * (cos * cos - cos) + cos;
55     }
56
57     *outSin = sin;
58     *outCos = cos;
59 }
60
61 /**
62 * Branchless signum function
63 */
64
65 int sgn(int val) {
66     return (0 < val) - (val < 0);
67 }
```

Listing A.5: util.cpp

```

1 #pragma once
2
3 void cossin(float x, float *outCos, float *outSin);
4
5 int sgn(int val);
```

Listing A.6: util.h

```

1 #include "MPU6050.h"
2 #include "communication.h"
3 #include "movement.h"
4 #include <Arduino.h>
5 #include <ESPAsyncWebServer.h>
6 #include <SPIFFS.h>
7 #include <WiFi.h>
8 #include <Wire.h>
9
10 // Replace with your network credentials
11 const char *ssid = "ESP32-OmniMove";
```

```
12 const char *password = "123456789";
13 // const char* ssid      = "moto g(6) 2970";
14 // const char* password = "428d1382abf1";
15
16 // const byte DNS_PORT = 53;
17 const IPAddress apIP = IPAddress(192, 168, 4, 1);
18
19 AsyncWebServer server(80);
20 AsyncWebSocket ws("/ws");
21 MPU6050 mpu;
22
23 void onEvent(AsyncWebSocket *server, AsyncWebSocketClient *client, AwsEventType
24 type, void *arg, uint8_t *data, size_t len) {
25
26     switch (type) {
27         case WS_EVT_CONNECT:
28             Serial.printf("Client connected from %s\n",
29             client->remoteIP().toString().c_str());
30             break;
31
32         case WS_EVT_DATA:
33             Communication::onWSData(server, client, type, data, len);
34             break;
35
36         default:
37             break;
38     }
39 }
40
41 void setup() {
42     // enableCore1WDT();
43     Serial.begin(115200);
44
45     Wire.begin();
46
47     mpu.initialize();
48
49     // mpu.CalibrateAccel_MP6500(6);
50     // mpu.CalibrateGyro(6);
51
52     // mpu.PrintActiveOffsets_MP6500();
53     mpu.setXGyroOffset(96);
54     mpu.setYGyroOffset(92);
```

```
53     mpu.setZGyroOffset(-20);
54
55     // Serial.printf("\n");
56     Movement::initPWM();
57
58     // enable AP with dns
59     WiFi.mode(WIFI_AP);
60
61     // Setup websockets
62     ws.onEvent(onEvent);
63     server.addHandler(&ws);
64     Communication::ws = &ws;
65     ws.enable(true);
66
67     // Initialize SPIFFS
68     if (!SPIFFS.begin(true)) {
69         while (true) {
70             Serial.println("An Error has occurred while mounting SPIFFS");
71             delay(1000);
72         }
73         return;
74     } else {
75         Serial.println("Mounted SPIFFS successfully");
76     }
77
78     // Route for root / web page
79     server.on("/", HTTP_GET, [] (AsyncWebRequest *request) {
80         // Serial.println("request on index");
81         request->send(SPIFFS, "/index.html", String(), false, nullptr);
82     });
83
84     // Route to load style.css file
85     server.on("/style.css", HTTP_GET, [] (AsyncWebRequest *request) {
86         // Serial.println("request on style");
87         request->send(SPIFFS, "/style.css", "text/css");
88     });
89
90     // Route to load code.js file
91     server.on("/code.js", HTTP_GET, [] (AsyncWebRequest *request) {
92         // Serial.println("request on code");
93         request->send(SPIFFS, "/code.js", "text/javascript");
94     });
95
```

```

96  // Route to load code.js file
97  server.on("/progressbar.min.js", HTTP_GET, [] (AsyncWebServerRequest *request) {
98      // Serial.println("request on progressbar");
99      request->send(SPIFFS, "/progressbar.min.js", "text/javascript");
100 });
101
102 // WiFi.softAPConfig(apIP, apIP, IPAddress(255, 255, 255, 0));
103 WiFi.setSleep(false);
104 WiFi.softAP(ssid, password, 6, 1);
105 delay(1000);
106
107 server.begin();
108 }
109
110 void loop() {
111
112 // put your main code here, to run repeatedly:
113 // dnsServer.processNextRequest();
114 uint16_t v = analogRead(39);
115 int16_t x, y, z, gx, gy, gz, temp;
116
117 temp = mpu.getTemperature();
118 mpu.getMotion6(&x, &y, &z, &gx, &gy, &gz);
119
120 // Serial.printf("x: %6.2fg, y: %6.2fg, z: %6.2fg, gx: %6.2f°/s, gy:
121 // %6.2f°/s, gz: %6.2f°/s, temp: %6.2f°C\r", x / 16384.0, y / 16384.0, z /
122 16384.0, gx / 250.0, gy / 250.0, gz / 250.0, temp / 340.0 + 36.53);
123 // power = (v / 4095 * 3.1 + .1) * 3;
124 Communication::sendCurrGyBatComb(0, 0, 0, x, y, z, gx, gy, gz, v, temp);
125 delay(1000);
126 }

```

Listing A.7: main.cpp

A.2 Webpage

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>

```

```
5  <title>ESP32 Web Server</title>
6  <meta charset="utf-8" />
7  <meta name="viewport" content="width=device-width, initial-scale=1">
8  <link rel="icon" href="data:, ">
9  <link rel="stylesheet" type="text/css" href="style.css">
10 </head>
11
12 <body>
13 <h1>ESP32 Web Server</h1>
14 <p>WebSocked Connected: <strong id="wsState">CONNECTING</strong></p>
15 <p><button class="button" onclick="sendSpeed()">Test Button</button></p>
16
17
18 <div class="wrapper">
19   <div class="left">
20     <h2>Messurements</h2>
21     <p>Accelleration: <strong id="valAccelTotal">0</strong> m/s2
22       <br /> <span id="valAccel">[x: 0, y: 0, z: 0]</span>
23     </p>
24
25     <p>Speed: <strong id="valSpeedTotal">0</strong> m/s
26       <br /> <span id="valSpeed">[x: 0, y: 0, z: 0]</span></p>
27
28     <p>Rotation: <strong id="valRotTotal">0</strong> rad/s2
29       <br /> <span id="valRot">[x: 0, y: 0, z: 0]</span></p>
30
31   </div>
32
33   <div class="right">
34     <h2>Battery</h2>
35     <progress id="progBattery" value="60" max="100"></progress>
36
37     <strong id="valBatteryTotal">-1</strong>%
38     <p>Voltage <strong id="valVolt">-1</strong> V</p>
39
40     <h2>Temprature</h2>
41     <p><strong id="valTemp">-1</strong> °C</p>
42   </div>
43 </div>
44
45
46 <div>
```

```
47   <svg viewBox="0 0 205 10" preserveAspectRatio="none" style="width: 100%;  
48   height: 100%;">  
49     <path id="pathVL" d="M50,2 L100,2 0,2 50,2" stroke="#FFEA82"  
50     stroke-width="4" fill-opacity="0"  
51     style="stroke-dasharray: 100, 100; stroke-dashoffset: 0;"/></path>  
52     <path id="pathHL" d="M50,8 L100,8 0,8 50,8" stroke="#FFEA82"  
53     stroke-width="4" fill-opacity="0"  
54     style="stroke-dasharray: 100, 100; stroke-dashoffset: 0;"/></path>  
55     <path id="pathVR" d="M155,2 L205,2 105,2 155,2" stroke="#FFEA82"  
56     stroke-width="4" fill-opacity="0"  
57     style="stroke-dasharray: 100, 100; stroke-dashoffset: 0;"/></path>  
58     <path id="pathHR" d="M155,8 L205,8 105,8 155,8" stroke="#FFEA82"  
59     stroke-width="4" fill-opacity="0"  
60     style="stroke-dasharray: 100, 100; stroke-dashoffset: 0;"/></path>  
61     <path d="M 50 0 L 50 4" stroke="black" stroke-width="0.5" fill="none" />  
62     <path d="M 50 6 L 50 10" stroke="black" stroke-width="0.5" fill="none" />  
63     <path d="M 155 0 L 155 4" stroke="black" stroke-width="0.5" fill="none" />  
64     <path d="M 155 6 L 155 10" stroke="black" stroke-width="0.5" fill="none" />  
65   </svg>  
66 </div>  
67 <div>  
68   <h5>Select Controller</h5>  
69   <button class="buttonsmall" onclick="updateControllerList()">Test  
70   Controller</button>  
71   <input type="checkbox" id="controllerEnabled" checked>  
72   <label for="controllerEnabled">Enable Controller?</label>  
73   <ul id="controllerList">  
74     <li onclick="onControllerSelected(0)">Test</li>  
75   </ul>  
76 </div>  
77 </body>  
78 </html>  
79 <script type="text/javascript" src="progressbar.min.js"></script>  
80 <script type="text/javascript" src="code.js"></script>  
81 </body>  
82 </html>
```

Listing A.8: index.html

```
1 let ws = new WebSocket("ws://192.168.4.1/ws");
2 // let ws = new WebSocket("ws://demos.kaazing.com/echo");
3 ws.binaryType = 'arraybuffer';
4
5 let barVL = new ProgressBar.Path('#pathVL', { easing: 'easeInOut', duration: 140,
6   });
7 let barVR = new ProgressBar.Path('#pathVR', { easing: 'easeInOut', duration: 140,
8   });
9 let barHL = new ProgressBar.Path('#pathHL', { easing: 'easeInOut', duration: 140,
10  });
11 let barHR = new ProgressBar.Path('#pathHR', { easing: 'easeInOut', duration: 140,
12  });
13
14 barVL.set(0.1)
15 barVR.set(0.1)
16 barHL.set(0.1)
17 barHR.set(0.1)
18
19
20 //##region >>> WebSocket
21 ws.onopen = (event) => {
22   document.getElementById("wsState").innerHTML = "OPEN";
23 }
24
25 ws.onerror = (event) => {
26   document.getElementById("wsState").innerHTML = "ERROR";
27 }
28
29 ws.onclose = (event) => {
30   document.getElementById("wsState").innerHTML = "CLOSED";
31 }
32
33 ws.onmessage = function (event) {
34   // console.log("WebSocket message received:", event);
```

```

35 let dv = new DataView(event.data);
36 let id = dv.getInt16(0, true);
37
38 switch (id) {
39     case 1: //gyro
40         let a = [dv.getInt16(2, true), dv.getInt16(4, true), dv.getInt16(6, true)];
41         let s = [dv.getInt16(8, true), dv.getInt16(10, true), dv.getInt16(12, true)];
42         let r = [dv.getInt16(2, true), dv.getInt16(4, true), dv.getInt16(6, true)];
43
44         document.getElementById("valAccelTotal").innerHTML = Math.sqrt(a[0] * a[0] +
45             a[1] * a[1] + a[2] * a[2]).toFixed(2);
46         document.getElementById("valSpeedTotal").innerHTML = Math.sqrt(a[0] * s[0] +
47             s[1] * s[1] + s[2] * s[2]).toFixed(2);
48         document.getElementById("valRotTotal").innerHTML = Math.sqrt(r[0] * r[0] +
49             r[1] * r[1] + r[2] * r[2]).toFixed(2);
50
51         document.getElementById("valAccel").innerHTML = "[x: " + a[0] + ", y: " +
52             a[1] + ", z: " + a[2] + "]";
53         document.getElementById("valSpeed").innerHTML = "[x: " + s[0] + ", y: " +
54             s[1] + ", z: " + s[2] + "]";
55         document.getElementById("valRot").innerHTML = "[x: " + r[0] + ", y: " + r[1] +
56             ", z: " + r[2] + "]";
57         break;
58
59     case 2: //battery
60         let c1 = (dv.getInt16(2, true) / 4095 * 3.1 + .1) * 3;
61
62         document.getElementById("valVolt").innerHTML = c1.toFixed(2);
63         break;
64
65     case 3: //motors
66         const UPPER_LIMIT = 300;
67
68         let vl = dv.getInt16(2, true) / (1023 * 4);
69         let vr = dv.getInt16(4, true) / (1023 * 4);
70         let hl = dv.getInt16(6, true) / (1023 * 4);
71         let hr = dv.getInt16(8, true) / (1023 * 4);
72
73         barVL.animate(vl);
74         barVR.animate(vr);
75         barHL.animate(hl);
76         barHR.animate(hr);
77         break;
78
79     default:
80         break;
81
82 }

```

```

72     case 4: //gyroBatComb
73         let s1 = [dv.getInt16(2, true), dv.getInt16(4, true), dv.getInt16(6, true)];
74         let a1 = [dv.getInt16(8, true) / 16384.0, dv.getInt16(10, true) / 16384.0,
75         dv.getInt16(12, true) / 16384.0];
76         let r1 = [dv.getInt16(14, true) / 250.0, dv.getInt16(16, true) / 250.0,
77         dv.getInt16(18, true) / 250.0];
78
79
80         document.getElementById("valAccelTotal").innerHTML = Math.sqrt(a1[0] * a1[0]
81         + a1[1] * a1[1] + a1[2] * a1[2]).toFixed(2);
82         document.getElementById("valSpeedTotal").innerHTML = Math.sqrt(s1[0] * s1[0]
83         + s1[1] * s1[1] + s1[2] * s1[2]).toFixed(2);
84         document.getElementById("valRotTotal").innerHTML = Math.sqrt(r1[0] * r1[0] +
85         r1[1] * r1[1] + r1[2] * r1[2]).toFixed(2);
86
87         document.getElementById("valAccel").innerHTML = "[x: " + a1[0].toFixed(2) +
88         ", y: " + a1[1].toFixed(2) + ", z: " + a1[2].toFixed(2) + "]";
89         document.getElementById("valSpeed").innerHTML = "[x: " + s1[0].toFixed(2) +
90         ", y: " + s1[1].toFixed(2) + ", z: " + s1[2].toFixed(2) + "]";
91         document.getElementById("valRot").innerHTML = "[x: " + r1[0].toFixed(2) + ",
92         y: " + r1[1].toFixed(2) + ", z: " + r1[2].toFixed(2) + "]";
93
94         let bat = (dv.getInt16(20, true) / 4095 * 3.1 + .1) * 3;
95         let batPercent = (bat - 6.5) / (7.8 - 6.5) * 100;
96         let temp = dv.getInt16(22, true) / 340.0 + 36.53;
97
98         document.getElementById("valVolt").innerHTML = bat.toFixed(2);
99         document.getElementById("progBattery").value = batPercent.toFixed(2);
100        document.getElementById("valBatteryTotal").innerHTML = batPercent.toFixed(0);
101
102        document.getElementById("valTemp").innerHTML = temp.toFixed(2);
103        break;
104
105    default:
106        break;
107    }
108};

//endregion
109
110//#region >>> Controller
111window.addEventListener("gamepadconnected", (event) => {
112    console.log("A gamepad connected:");
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
871
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1
```

```
107  console.log(event.gamepad);
108
109  updateControllerList();
110 });
111
112 window.addEventListener("gamepaddisconnected", (event) => {
113  console.log("A gamepad disconnected:");
114  console.log(event.gamepad);
115
116  updateControllerList();
117 });
118
119
120
121
122 // Lets the user select which connected controller to use
123 function updateControllerList() {
124  let inner = ""
125
126  let pads = navigator.getGamepads();
127  console.log(pads);
128
129  for (let i = 0; i < pads.length; i++) {
130    const element = pads[i];
131    if (element === null)
132      continue;
133    if (selectedControllerIndex == i) {
134      inner += "<li onclick=\"onControllerSelected(" + i + ")\"><strong>" +
135      element.id + "</strong></li>";
136    } else {
137      inner += "<li onclick=\"onControllerSelected(" + i + ")\">" + element.id +
138      "</li>";
139    }
140  }
141
142  document.getElementById("controllerList").innerHTML = inner;
143
144 // gets called when a Controller gets selected in the List
145 function onControllerSelected(index) {
146  selectedControllerIndex = index;
147  updateControllerList();
148 }
```

```
148
149
150 let lastX1 = 0, lastX2 = 0, lastY1 = 0, lastY2 = 0;
151 function controllerFunc() {
152     if (!document.getElementById("controllerEnabled").checked)
153         return;
154
155     let pads = navigator.getGamepads();
156     if (pads === null || pads === undefined || pads[selectedControllerIndex] === null
157         || pads[selectedControllerIndex] === undefined)
158         return;
159     x1 = Math.floor(pads[selectedControllerIndex].axes[0] * 1023);
160     y1 = -Math.floor(pads[selectedControllerIndex].axes[1] * 1023);
161     x2 = Math.floor(pads[selectedControllerIndex].axes[2] * 1023);
162     y2 = -Math.floor(pads[selectedControllerIndex].axes[3] * 1023);
163
164     if (Math.abs(x1) < 200)
165         x1 -= x1;
166     if (Math.abs(y1) < 200)
167         y1 -= y1;
168     if (Math.abs(x2) < 200)
169         x2 -= x2;
170     if (Math.abs(y2) < 200)
171         y2 -= y2;
172
173     if (lastX1 != x1 || lastY1 != y1 || lastX2 != x2 || lastY2 != y2) {
174         console.log(x1 + ", " + y1 + " : " + x2 + ", " + y2);
175         if (ws.readyState == WebSocket.OPEN) {
176             let buf = new Int16Array([0, x1, y1, x2, y2])
177             ws.send(buf);
178         }
179         lastX1 = x1;
180         lastX2 = x2;
181         lastY1 = y1;
182         lastY2 = y2;
183     }
184
185 }
186 //endregion
187
188 //#region >>> Keyboard Control
189 let dirX = 0;
```

```
190 let dirY = 0;
191 let rotX = 0;
192 //a: 65, s: 83, d:68, w:87
193 document.addEventListener('keydown', function (event) {
194     let c = event.keyCode;
195     switch (event.keyCode) {
196         case 65:
197             dirX = -1;
198             break;
199         case 68:
200             dirX = 1;
201             break;
202         case 87:
203             dirY = 1;
204             break;
205         case 83:
206             dirY = -1;
207             break;
208         case 39:
209             rotX = 1;
210             break;
211         case 37:
212             rotX = -1;
213             break;
214         case 27:
215             dirY = 0;
216             dirX = 0;
217             break;
218         default:
219             break;
220     }
221
222     if (lastX1 != dirX * 1023 || lastY1 != dirY * 1023 || lastX2 != rotX * 1023 ||
223         lastY2 != 0) {
224         console.log(dirX * 1023 + ", " + dirY * 1023 + " : " + rotX * 1023 + ", " + 0);
225         if (ws.readyState == WebSocket.OPEN) {
226             let buf = new Int16Array([0, dirX * 1023, dirY * 1023, rotX * 1023, 0])
227             ws.send(buf);
228         }
229         lastX1 = dirX * 1023;
230         lastY1 = dirY * 1023;
231         lastX2 = rotX * 1023;
232         lastY2 = 0;
233     }
234 }
```

```
232     }
233
234     // console.log("x: " + dirX + " y: " + dirY);
235 });
236
237 document.addEventListener('keyup', function (event) {
238     let c = event.keyCode;
239     switch (event.keyCode) {
240         case 65:
241         case 68:
242             dirX = 0;
243             break;
244         case 87:
245         case 83:
246             dirY = 0;
247             break;
248         case 39:
249         case 37:
250             rotX = 0;
251             break;
252         default:
253             break;
254     }
255     console.log("x: " + dirX + " y: " + dirY);
256
257     if (lastX1 != dirX * 1023 || lastY1 != dirY * 1023 || lastX2 != rotX * 1023 ||
258         lastY2 != 0) {
259         console.log(dirX * 1023 + ", " + dirY * 1023 + " : " + rotX * 1023 + ", " + 0);
260         if (ws.readyState == WebSocket.OPEN) {
261             let buf = new Int16Array([0, dirX * 1023, dirY * 1023, rotX * 1023, 0])
262             ws.send(buf);
263         }
264         lastX1 = dirX * 1023;
265         lastY1 = dirY * 1023;
266         lastX2 = rotX * 1023;
267         lastY2 = 0;
268     }
269 });
270
271
272
273 function sendSpeed() {
```

```
274 console.log("speed");
275 let arr = new Int16Array([2, Math.random() * 4048, Math.random() * 4048]);
276 let arr2 = new Int16Array([1,
277     Math.random() * 1023, Math.random() * 1023, Math.random() * 1023,
278     Math.random() * 1023, Math.random() * 1023, Math.random() * 1023,
279     Math.random() * 1023, Math.random() * 1023, Math.random() * 1023]);
280 ws.send(arr);
281 ws.send(arr2);
282 }
```

Listing A.9: code.js

```
1 html {
2     font-family: Helvetica;
3     display: inline-block;
4     margin: 0px auto;
5     text-align: center;
6 }
7 h1{
8     color: #0F3376;
9     padding: 2vh;
10 }
11 p{
12     font-size: 1.5rem;
13 }
14 .button {
15     display: inline-block;
16     background-color: #008CBA;
17     border: none;
18     border-radius: 4px;
19     color: white;
20     padding: 16px 40px;
21     text-decoration: none;
22     font-size: 30px;
23     margin: 2px;
24     cursor: pointer;
25 }
26
27 .buttonsmall {
28     display: inline-block;
29     background-color: #008CBA;
30     border: none;
```

```
31 border-radius: 4px;  
32 color: white;  
33 padding: 8px 20px;  
34 text-decoration: none;  
35 font-size: 16px;  
36 margin: 2px;  
37 cursor: pointer;  
38 }  
39  
40 .button2 {  
41 background-color: #f44336;  
42 }  
43  
44 .wrapper {  
45 display: flex;  
46 }  
47  
48 .left {  
49 flex: 0 0 65%;  
50 }  
51  
52 .right {  
53 flex: 1;  
54 }
```

Listing A.10: style.css

B Technische Zeichnungen

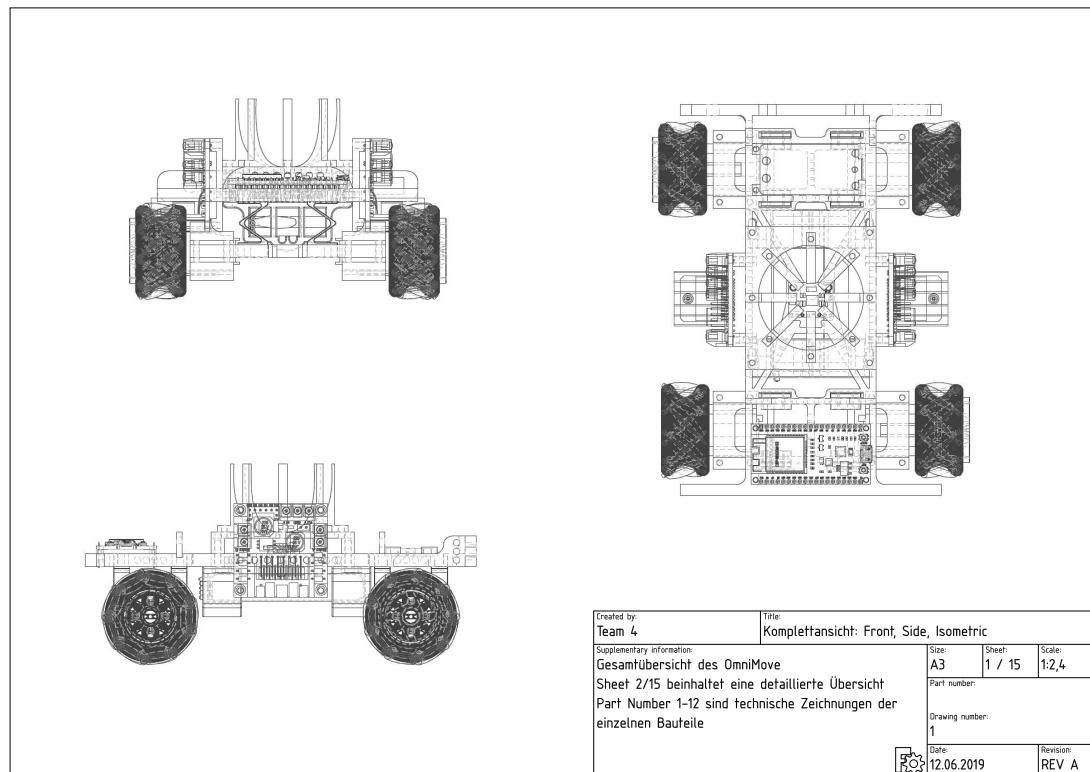


Abbildung B.1: Komplettansicht

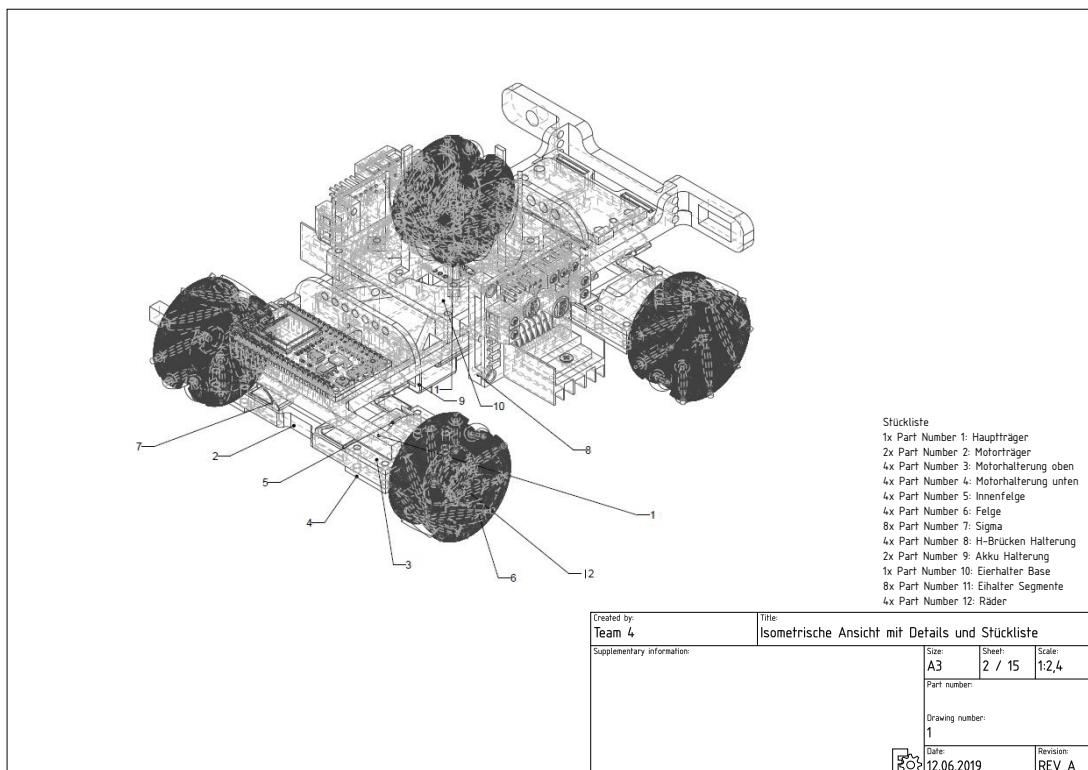


Abbildung B.2: Isometrisch und Stückliste

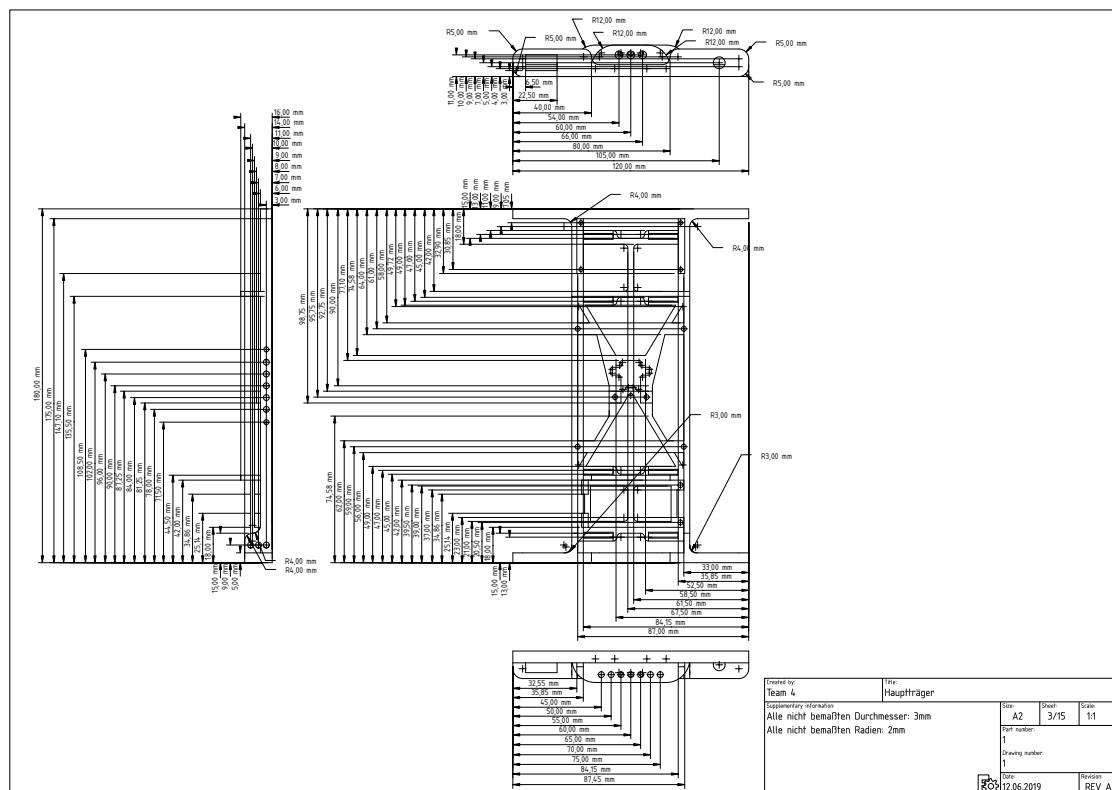


Abbildung B.3: Hauptträger

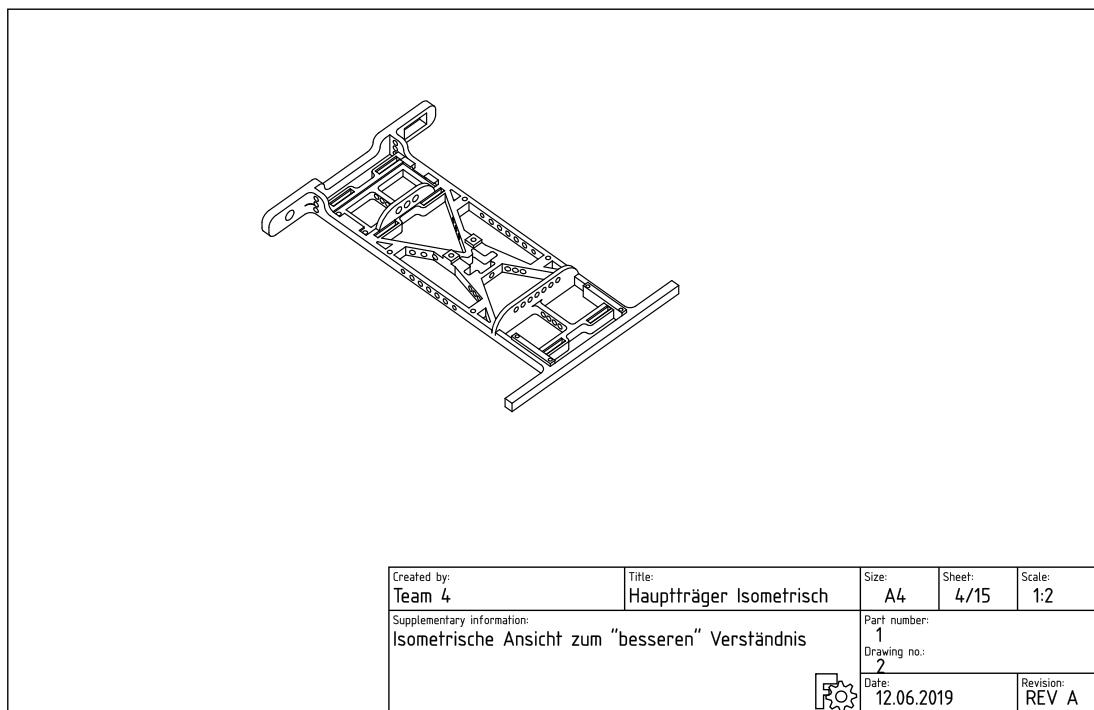


Abbildung B.4: Hauptträger Isometrisch

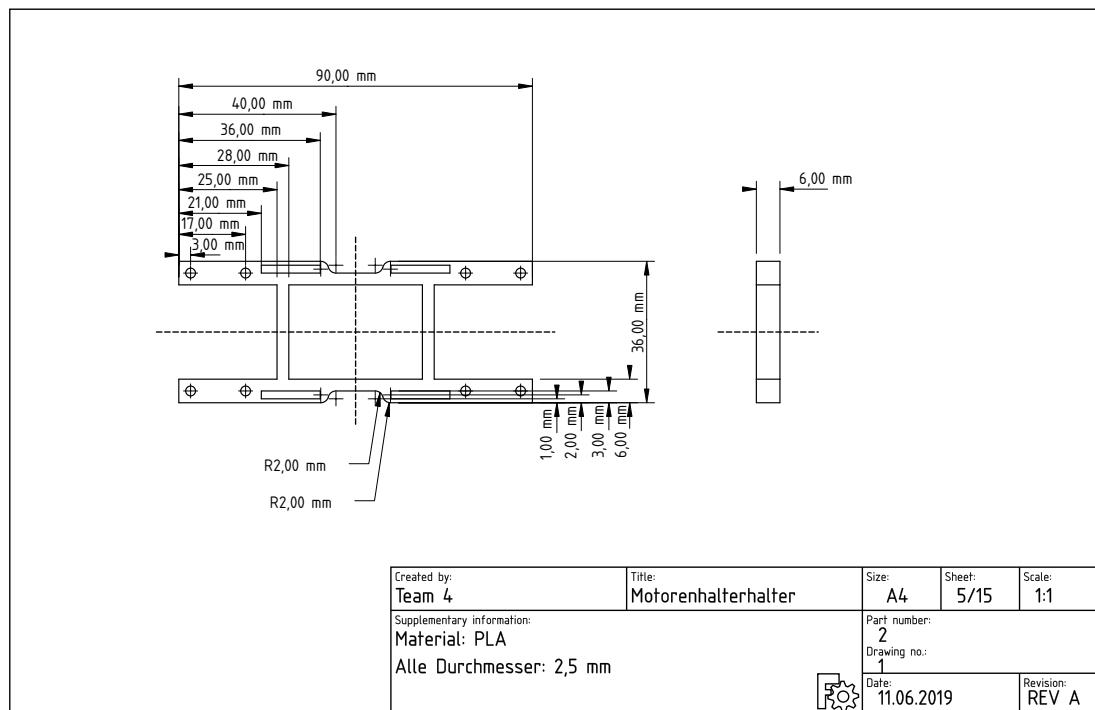


Abbildung B.5: Motorenhalterhalter

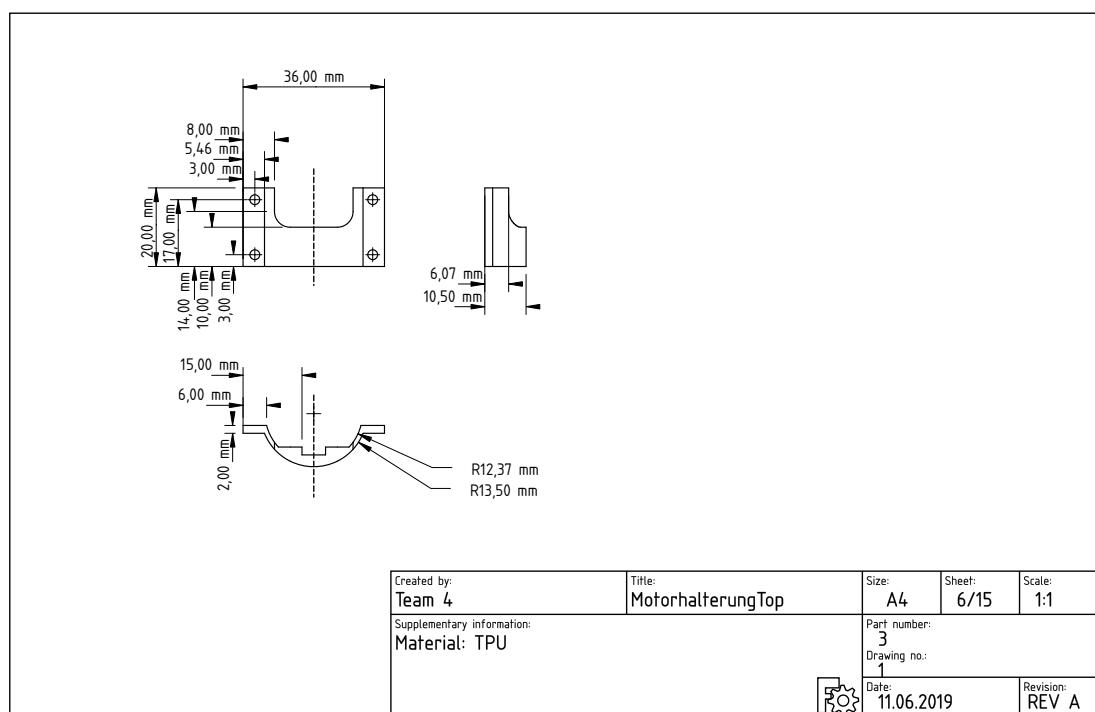


Abbildung B.6: Motorenhalterung Oben

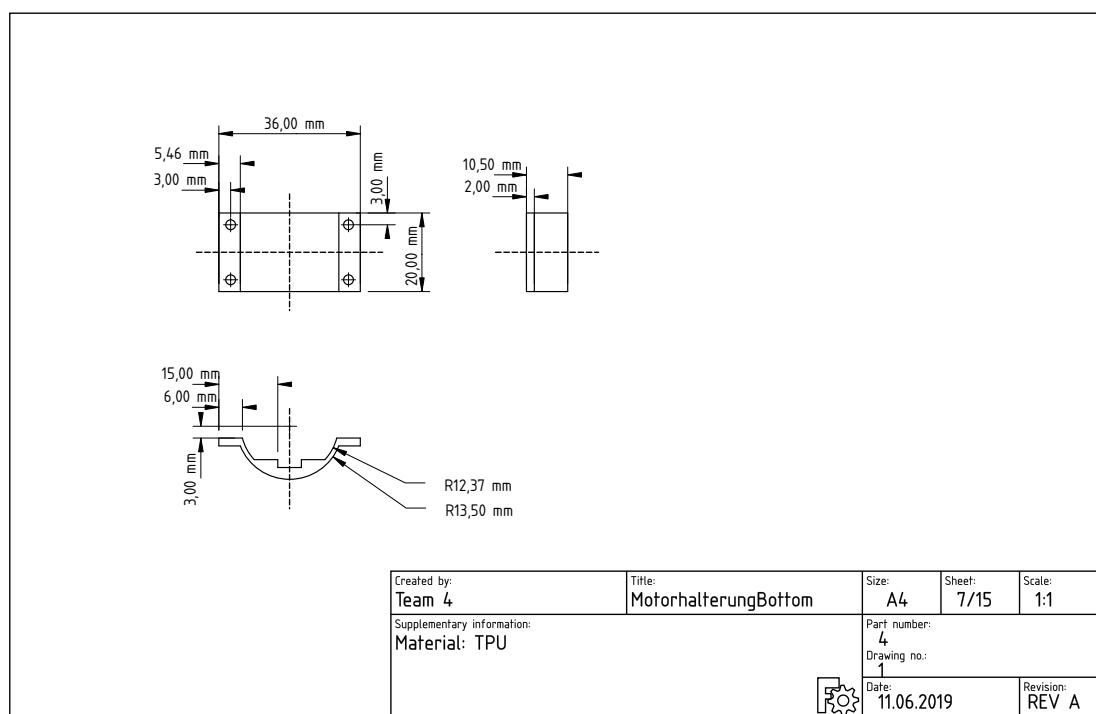


Abbildung B.7: Motorenhalterung Unten

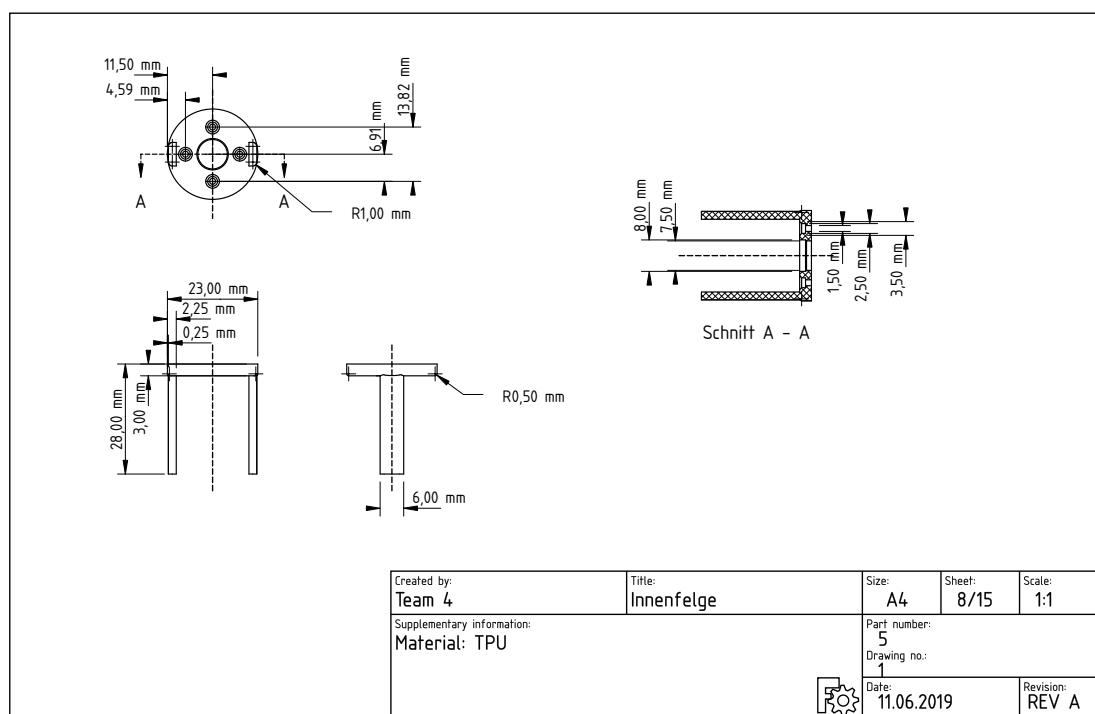


Abbildung B.8: Innenfelge

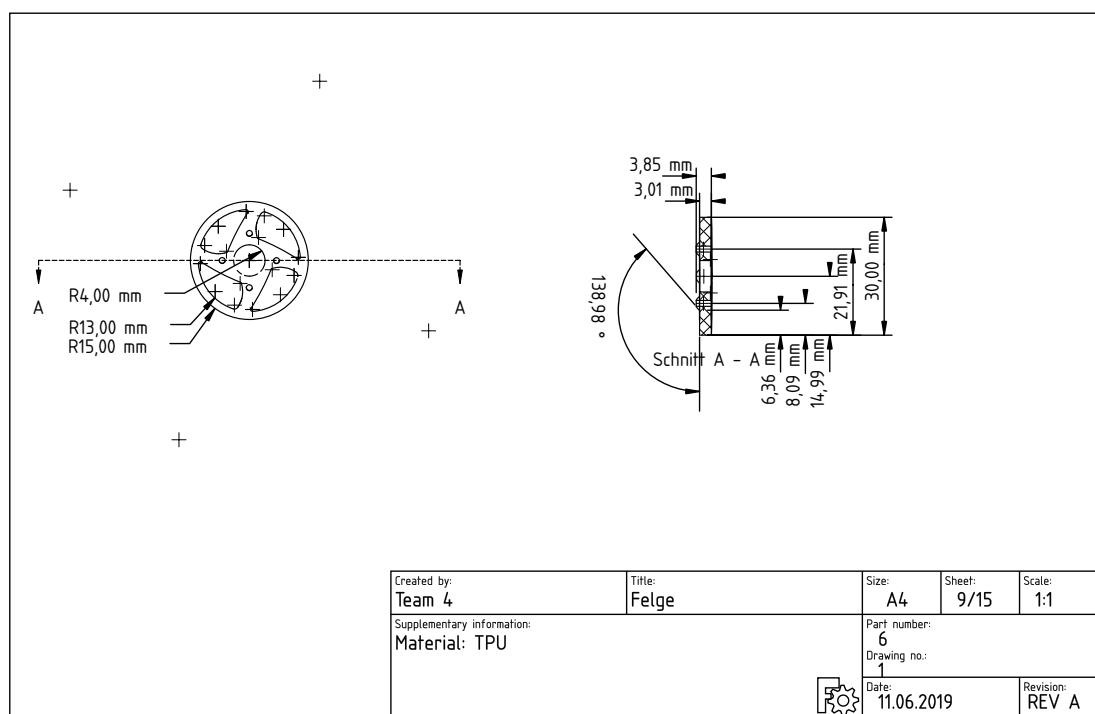


Abbildung B.9: Felge

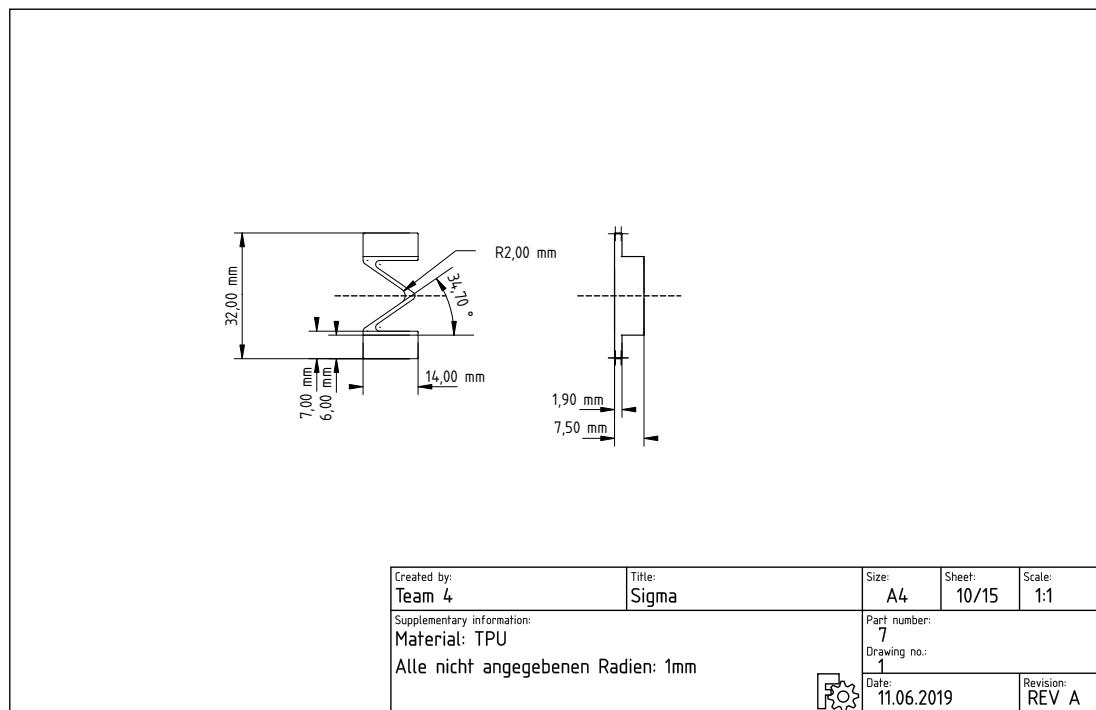


Abbildung B.10: Sigma

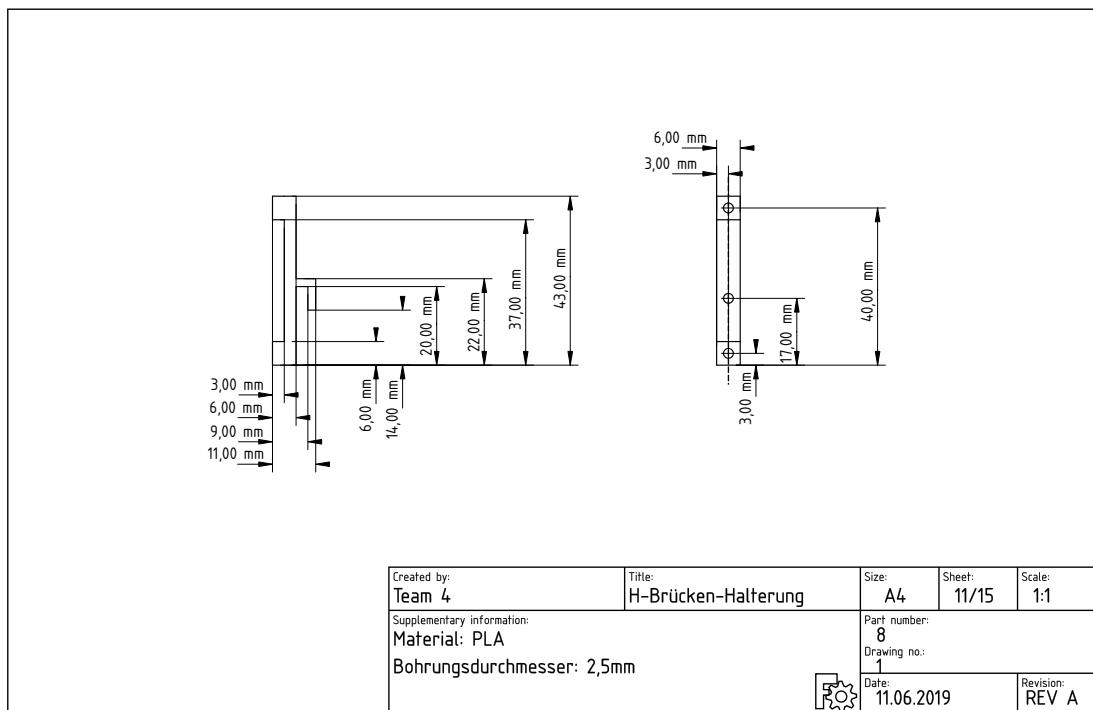


Abbildung B.11: H-Brücken-Halterung

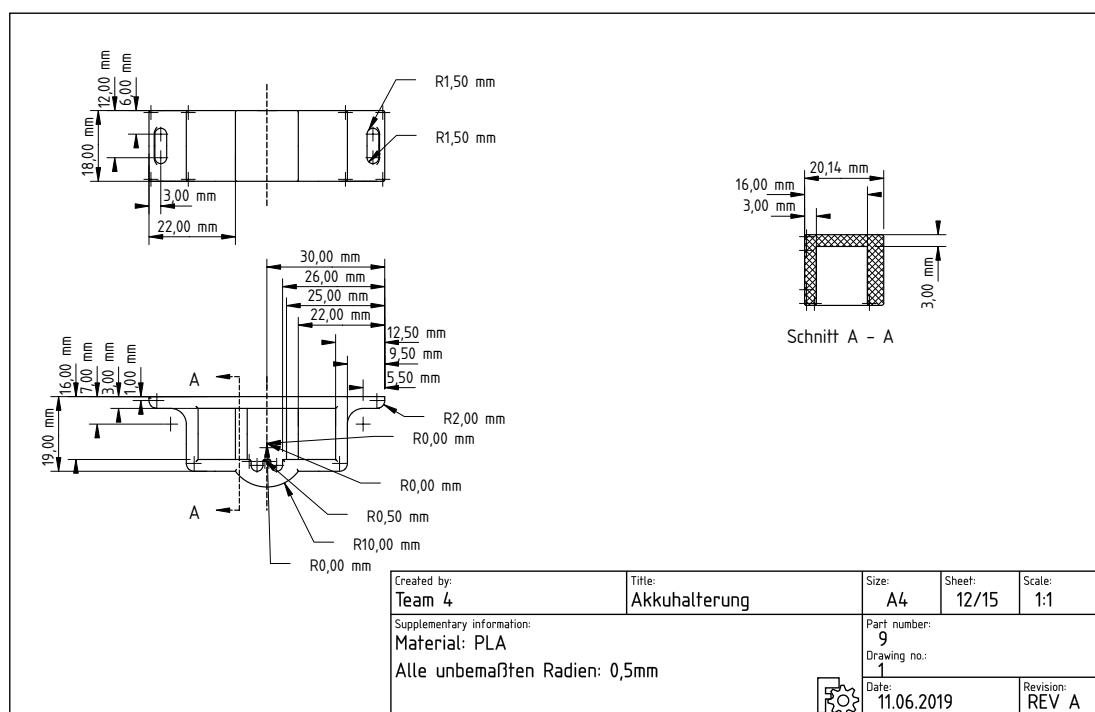


Abbildung B.12: Akkuhalterung

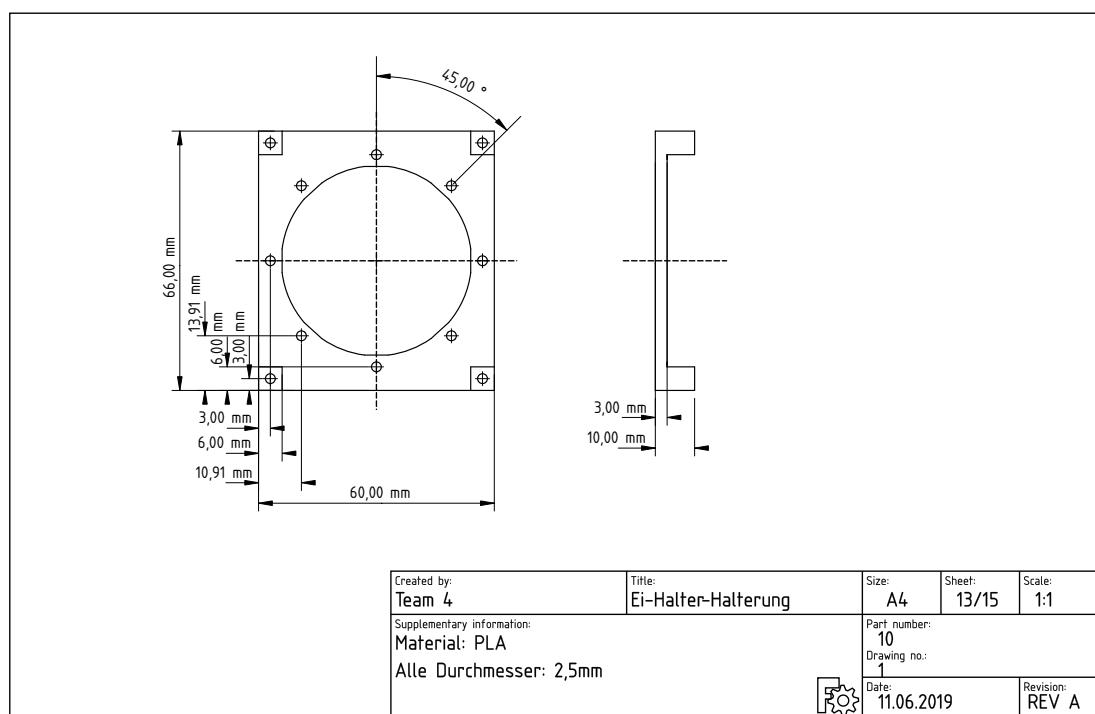


Abbildung B.13: Ei-Halterung-Halterung

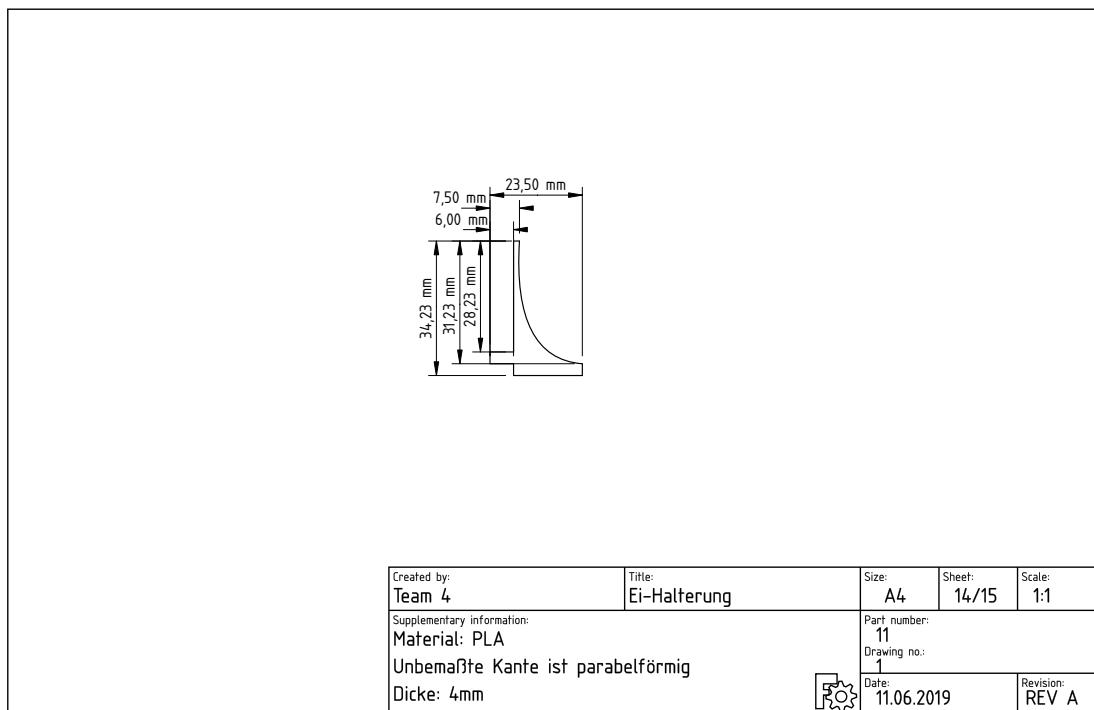


Abbildung B.14: Ei-Halterung

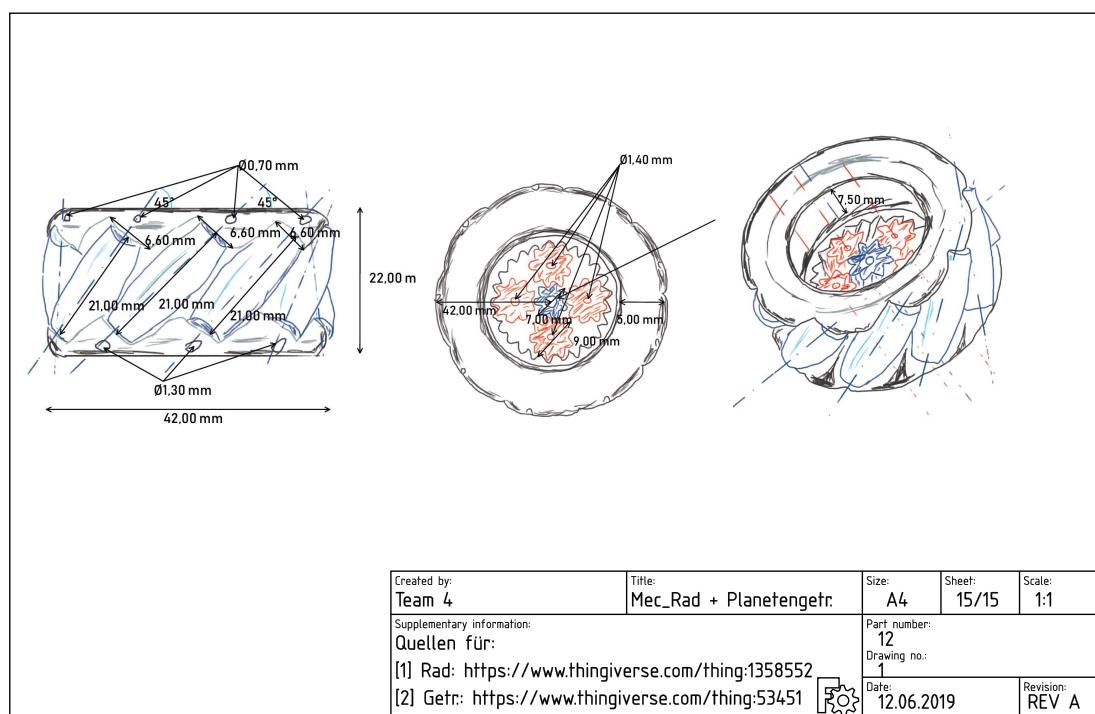


Abbildung B.15: Reifen und Planetengetriebe

C Software und Konfigurationsdateien

Beispieltext

C.1 Software A

C.2 Software B

Beispieltext

Literaturverzeichnis

- [1] ADOBE (Hrsg.): *Adobe Reader - Herunterladen.* <http://www.adobe.de/products/acrobat/readstep2.html>: Adobe, August 2005. – Der PDF-Betrachter Acrobat Reader
- [2] GHOSTVIEW (Hrsg.): *Ghostscript, Ghostview and GSview.* <http://www.cs.wisc.edu/~ghost>: Ghostview, August 2005. – Welcome to the Home Page for Ghostscript, an interpreter for the PostScript language and for PDF, and related software and documentation.
- [3] IMAGEMAGICK STUDIO LLC (Hrsg.): *ImageMagick: Convert, Edit and Compose Images.* <http://www.imagemagick.org/script/index.php>: ImageMagick Studio LLC, August 2005. – ImageMagick, version 6.2.3, is a free software suite to create, edit, and compose bitmap images. It can read, convert and write images in a large variety of formats.
- [4] KDE (Hrsg.): *Kile - an integrated LaTeX environment.* <http://kile.sourceforge.net>: KDE, August 2005. – Homepage der LaTeX-Entwicklungsumgebung Kile für KDE bzw. GNU/Linux
- [5] KOPKA, Helmut: *Latex Einführung.* 3. Addison Wesley, 2000 (Scientific Tools). – 521 Seiten
- [6] SOURCEFORGE (Hrsg.): *MiKTeX Project Page.* <http://www.miktex.org>: Sourceforge, August 2005. – MiKTeX is an up-to-date TeX implementation for the Windows operating system.
- [7] SOURCEFORGE (Hrsg.): *Project: TeXnicCenter: Summary.* <http://sourceforge.net/projects/texniccenter>: Sourceforge, August 2005. – Projektseite der Software TeXnicCenter, einer Entwicklungsumgebung für LaTeX-dokumente unter Microsoft Windows

- [8] TOOLSCENTER.ORG (Hrsg.): *TeXnicCenter*. <http://www.texniccenter.org>: ToolsCenter.org, August 2005. – Homepage des Projektes TeXnicCenter, einer Entwicklungsumgebung für LaTeXdokumente unter Microsoft Windows
- [9] UNIVERSITÄT GIESSEN (Hrsg.): *Kochbuch für LaTeX*. <http://www.uni-giessen.de/hrz/tex/cookbook/cookbook.html>: Universität Gießen, Januar 1995. – Dieser Text beschreibt in Form eines Kochbuches LaTeX-Textteile.
- [10] VRODO (Hrsg.): *Testberichte und News zu AR-Brillen*. <https://vrodo.de/augmented-reality-brillen-vergleich/>: VRODO, Dezember 2018

Listings

A.1	communication.cpp	16
A.2	communication.h	17
A.3	movement.cpp	18
A.4	movement.h	22
A.5	util.cpp	24
A.6	util.h	26
A.7	main.cpp	26
A.8	index.html	29
A.9	code.js	32
A.10	style.css	39

Abbildungsverzeichnis

3.1	Aufruf von <code>latex</code>	9
3.2	Aufruf von <code>pdflatex</code>	9
3.3	Bildschirmfoto <code>kile</code>	10
3.4	Bildschirmfoto <code>texniccenter</code>	11
B.1	Komplettansicht	41
B.2	Isometrisch und Stückliste	42
B.3	Hauptträger	43
B.4	Hauptträger Isometrisch	44
B.5	Motorenhalterhalter	45
B.6	Motorenhalterung Oben	46
B.7	Motorenhalterung Unten	47
B.8	Innenfelge	48
B.9	Felge	49
B.10	Sigma	50
B.11	H-Brücken-Halterung	51
B.12	Akkuhalterung	52
B.13	Ei-Halterung-Halterung	53
B.14	Ei-Halterung	54
B.15	Reifen und Planetengetriebe	55

Tabellenverzeichnis

3.1	Benötigte Programme unter Windows	6
3.2	Benötigte Programme unter GNU/Linux	7
3.3	Kommandos zum manuellen L ^A T _E X-Aufruf	8
3.4	Relevante Dateien im Paket	12