



University of Applied Sciences

**HOCHSCHULE  
EMDEN·LEER**

Fachbereich Technik  
Abteilung Elektrotechnik und Informatik

# ENTWICKLUNG EINES FAHRZEUGES FÜR DEN HYPERLOOP (STEUERUNG UND SCHALTPLAN)

PROJEKTARBEIT  
Studiengang Elektrotechnik

Vorgelegt von  
Oliver Schmidt  
Studiengang Elektrotechnik  
Matr. Nr. 7023462

Emden, 26. Oktober 2024

Betreut von  
Prof. Dr.-Ing. Kane  
Prof. Dr.-Ing. Thomas Schüning  
Prof. Dr. rer. nat. Walter Neu

# Rechtliche Erklärung

## Erklärung

- [ja|nein] Die vorliegende Arbeit enthält vertrauliche / kommerziell nutzbare Informationen, deren Rechte außerhalb der Hochschule Emden/Leer liegen. Sie darf nur den am Prüfungsverfahren beteiligten Personen zugänglich gemacht werden, die hiermit auf ihre Pflicht zur Vertraulichkeit hingewiesen werden (Sperrvermerk).
- [ ja|nein-] Soweit meine Rechte berührt sind, erkläre ich mich einverstanden, dass die vorliegende Arbeit Angehörigen der Hochschule Emden/Leer für Studium / Lehre / Forschung uneingeschränkt zugänglich gemacht werden kann.

## Eidesstattliche Versicherung

Ich, der Unterzeichnende, erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Alle Quellenangaben und Zitate sind richtig und vollständig wiedergegeben und in den jeweiligen Kapiteln und im Literaturverzeichnis wiedergegeben. Die vorliegende Arbeit wurde nicht in dieser oder einer ähnlichen Form ganz oder in Teilen zur Erlangung eines akademischen Abschlussgrades oder einer anderen Prüfungsleistung eingereicht.

Mir ist bekannt, dass falsche Angaben im Zusammenhang mit dieser Erklärung strafrechtlich verfolgt werden können.

Emden, 26. Oktober 2024

*Oliver Schmidt*

# Inhaltsverzeichnis

|  |           |
|--|-----------|
| Rechtliche Erklärung . . . . .                     | I         |
| Abbildungsverzeichnis . . . . .                    | III       |
| Tabellenverzeichnis . . . . .                      | IV        |
| <b>1 Einleitung</b>                                | <b>1</b>  |
| 1.1 Motivation der Hyperloop Technologie . . . . . | 1         |
| 1.1.1 Institute of Hyperloop Technology . . . . .  | 2         |
| 1.2 Aufgabenstellung . . . . .                     | 3         |
| 1.3 Aufbau der Projektdokumentation . . . . .      | 3         |
| <b>2 Konzept</b>                                   | <b>4</b>  |
| <b>3 Stand der Technik</b>                         | <b>5</b>  |
| 3.1 Antrieb – Golden Motor . . . . .               | 5         |
| 3.1.1 BLDC Motor . . . . .                         | 5         |
| 3.1.1.1 Funktionsweise . . . . .                   | 7         |
| 3.1.2 Vector Controller . . . . .                  | 8         |
| 3.1.2.1 Funktionsweise . . . . .                   | 9         |
| 3.2 Echtzeitfähige Steuerung – Speedgoat . . . . . | 10        |
| 3.2.1 I/O-Modul – IO397-50k . . . . .              | 12        |
| 3.2.1.1 Pin Mapping – IO397-50k . . . . .          | 12        |
| 3.2.1.2 IO397-50k – Konfigurieren . . . . .        | 14        |
| 3.2.2 I/O-Modul – IO691 . . . . .                  | 17        |
| 3.2.2.1 Pin Mapping – IO691 . . . . .              | 18        |
| 3.2.2.2 IO691 Module Konfigurieren . . . . .       | 18        |
| <b>4 Implementierung</b>                           | <b>22</b> |
| 4.1 Schaltplan . . . . .                           | 22        |
| 4.1.1 Betriebsmittelbezeichnung . . . . .          | 22        |
| 4.1.1.1 Funktionskennzeichen . . . . .             | 23        |
| 4.1.1.2 Ortskennzeichen . . . . .                  | 23        |
| 4.1.1.3 Betriebsmittelkennzeichen . . . . .        | 23        |
| 4.1.2 Verdrahtung Konventionen . . . . .           | 25        |

|          |  |           |
|----------|--|-----------|
| 4.2      | Simulation mit Simulink . . . . .            | 26        |
| 4.2.1    | Funktionsweise der Steuerung . . . . .       | 26        |
| 4.2.1.1  | Controllpanel . . . . .                      | 26        |
| 4.2.2    | Automatensteuerung . . . . .                 | 28        |
| 4.2.2.1  | Automat für die manuelle Steuerung . . . . . | 30        |
| 4.2.2.2  | Automat für die automatische Steuerung . . . | 32        |
| 4.3      | Distanzmessung . . . . .                     | 34        |
| 4.3.1    | Verbindung . . . . .                         | 34        |
| 4.3.2    | Kodierung der Steuersignale . . . . .        | 36        |
| 4.3.3    | Steuerung für die Sensordaten . . . . .      | 36        |
| <b>5</b> | <b>Konklusion</b>                            | <b>37</b> |
| <b>A</b> | <b>Anhang</b>                                | <b>39</b> |
| A.1      | Schaltplan . . . . .                         | 39        |

# Abbildungsverzeichnis

|      |  |    |
|------|--|----|
| 1.1  | Hyperloop der Hochschule Emden-Leer . . . . .                  | 1  |
| 2.1  | Konzept – Aufbauplan des Fahrzeugs . . . . .                   | 4  |
| 3.1  | Golden Motor – 10 KW BLDC Motor Liquid Cooled . . . . .        | 6  |
| 3.2  | Golden Motor – VECTOR 500 Motor Controller . . . . .           | 8  |
| 3.3  | Speedgoat – Baseline Real-Time Target Machine . . . . .        | 10 |
| 3.4  | Speedgoat – Ansicht der Modulanschlüsse . . . . .              | 11 |
| 3.5  | Speedgoat – I/O Module 397 [4] . . . . .                       | 12 |
| 3.6  | Simulink – Driver Block – IO397-50k . . . . .                  | 15 |
| 3.7  | Speedgoat – TTL I/O Interface – IO397-50k [4, S. 12] . . . . . | 15 |
| 3.8  | Simulink – Pull-Widerstände – IO397-50k [4, S. 12] . . . . .   | 16 |
| 3.9  | Speedgoat – I/O Module 691 [5] . . . . .                       | 17 |
| 3.10 | Simulink – Driver Block – I/O Module 691 [5] . . . . .         | 18 |
| 3.11 | Simulink – Baudrate – I/O Module 691 [6] . . . . .             | 20 |
| 4.1  | Automat – Controllpanel . . . . .                              | 27 |
| 4.2  | Automat – Start Zustand »Idle« . . . . .                       | 28 |
| 4.3  | Automat – Funktion Driving . . . . .                           | 29 |
| 4.4  | Automat – Übersicht der manuellen Steuerung . . . . .          | 31 |
| 4.5  | Automat – »Forwards« -Stateflow der manuellen Steuerung .      | 32 |
| 4.6  | Automat – Übersicht der automatischen Steuerung . . . . .      | 35 |
| 4.7  | PARKSIDE – Distanzsensor – Innenaufbau . . . . .               | 36 |

# Tabellenverzeichnis

|      |  |    |
|------|--|----|
| 3.1  | Pin Mapping – BLDC-Motor . . . . .   | 7  |
| 3.2  | Pin Mapping – Vector Controller . . . . .                                    | 9  |
| 3.3  | Speedgoat – Anschlüsse der Module . . . . .                                  | 12 |
| 3.4  | IO397-50k Pin Mapping – Terminal Board A: analog I/O [4,<br>S. 15] . . . . . | 13 |
| 3.5  | IO397-50k Pin Mapping – Terminal Board B: analog I/O [4,<br>S. 15] . . . . . | 14 |
| 3.6  | Simulink – Pull-Widerstand – Reaktionszeit . . . . .                         | 17 |
| 3.7  | IO691 Pin Mapping . . . . .  | 18 |
| 3.8  | Simulink – Baudraten Parameter . . . . .                                     | 19 |
| 4.1  | Schaltplan – Funktionskennzeichen (=) . . . . .                              | 23 |
| 4.2  | Schaltplan – Ortskennzeichen (+) . . . . .                                   | 23 |
| 4.3  | Schaltplan – Betriebsmittelkennzeichen (-) [8] . . . . .                     | 24 |
| 4.4  | Verdrahtung Konvention – Aderfarben . . . . .                                | 25 |
| 4.5  | Automat – Wahlschalter für den manuellen Betrieb . . . . .                   | 26 |
| 4.6  | Ausgänge – Idle . . . . .  | 29 |
| 4.7  | Ausgänge – Emergency stop . . . . .  | 30 |
| 4.8  | Ausgänge – der Zustände »Forwards« und »Backwards« –<br>»slow« . . . . .     | 31 |
| 4.9  | Ausgänge – der Zustände »Forwards« und »Backwards« – »fast«                  | 32 |
| 4.10 | Ausgänge – Automatic . . . . .   | 33 |
| 4.11 | Ausgänge – Drive . . . . .   | 33 |
| 4.12 | Ausgänge – Distance . . . . .  | 34 |
| 4.13 | Ausgänge – Stop . . . . .  | 34 |
| 4.14 | PARKSIDE – Pin Mapping – Distanzsensor . . . . .                             | 35 |

# Kapitel 1

## Einleitung

### 1.1 Motivation der Hyperloop Technologie

Der Hyperloop ist ein innovatives Transportkonzept, dass eine ökonomische, klimafreundliche und schnellere Alternative zu herkömmlichen Verkehrsmitteln wie Lastkraftwagen, Zügen und Flugzeugen bietet.

Derzeit stehen herkömmlichen Transportmitteln zwei wesentliche Hinder-



**Abbildung 1.1:** Hyperloop der Hochschule Emden-Leer

nisse im Weg, um Personen oder Güter schnell und emissionsarm zu befördern: Zum einen der hohe Luftwiderstand, der bei hohen Geschwindigkeiten den Energieverbrauch stark erhöht, und zum anderen der Rollwiderstand der Räder, der ebenfalls zu einem höheren Energiebedarf führt.

Der Hyperloop löst diese Probleme, indem er Güter und Personen in einem Fahrzeug, das sich in einer Vakuumröhre bewegt, wie in Abbildung 1.1 dargestellt ist. Zudem wird das Fahrzeug, wie bei einer Magnetschwebebahntechnik angehoben. Somit lassen sich Roll- und Luftwiderstand fast vollkommen aufheben.

### 1.1.1 Institute of Hyperloop Technology

Im Rahmen dieser Forschung wurde an der Hochschule Emden eine Teststrecke mit einer Länge von 27 Metern errichtet (siehe Abbildung 1.1). Auf dieser Strecke soll das Fahrzeug (Pod) unter realistischen Bedingungen getestet und weiterentwickelt werden. Die Teststrecke besteht aus einem Schienensystem und einem Linearmotor.

## 1.2 Aufgabenstellung

To-do

- Ablauforientiert erklären. Also erst die Bestellung, dann der Schaltplan und dann die Simulation mit Simulink.
- Aufgabenstellung in der Vergangenheit formulieren.
- Den Leser in der Doku struktur Einführen. Am enden in 1.x

Die Motivation für dieses Projekt liegt in der Entwicklung eines Hyperloop-Fahrzeugs, das mit einer Batterie und einem Motor betrieben wird. Für die Steuerung des Fahrzeugs wurde ein echtzeitfähiges Steuerungssystem der Firma Speedgoat vorgeben, welches in Abschnitt 3.2 vertieft wird.

Im Rahmen des Projekts wird ein Fahrzeug (Pod) für den Hyperloop mit einer Bordspannung von 48 V konzipiert. Ziel ist es, die Realisierbarkeit dieser Spannung zu überprüfen und umzusetzen. Dazu gehören die Planung und Simulierung, die Integration der erforderlichen Sensorik sowie die Beschaffung der notwendigen Bauteile. Die Logik- und Signalverarbeitung wird mithilfe von Simulink auf dem echtzeitfähigen Speedgoat-System durchgeführt. Die Steuerung erfolgt über Simulink, ein Modul von MATLAB, und umfasst die Erfassung von Position und Beschleunigung des Fahrzeuges. Der Motor wird über ein zusätzliches Steuergerät angesteuert. Die Steuerung soll in Form einer Automatensteuerung umgesetzt werden. Die Verdrahtung des Pods wird entsprechend der Bordspannung von 48 V ausgelegt. Hierfür wird mit der Software QElectroTech ein Schaltplan (siehe Anhang A.1) erstellt. Alle erforderlichen Bauteile für die Umsetzung der Bordspannung, die Verdrahtung und die Sensorik müssen beschafft werden. Textergebnisse und Inbetriebnahme entfallen.

## 1.3 Aufbau der Projektdokumentation

To-do

- Wird am Ende geschrieben.

## Kapitel 2

# Konzept

Im Folgenden wird der Aufbau des Fahrzeugs erklärt und welche Sensoren verbaut werden. Dabei übernimmt das Steuerungs- und Datenerfassungskonzept von Speedgoat eine zentrale Rolle.

Wie in Abbildung 2.1 dargestellt, werden die Baugruppen des Fahrzeugs an

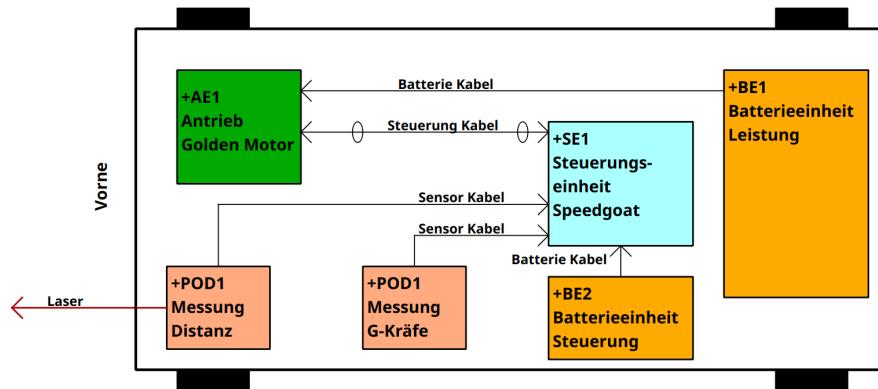


Abbildung 2.1: Konzept – Aufbauplan des Fahrzeugs

verschiedenen Stellen miteinander verbunden. Dabei übernimmt die Steuereinheit (+SE1) eine zentrale Rolle. Mit der echtzeitfähigen Steuerung von Speedgoat werden alle digitalen und analogen Ein- und Ausgangssignale gesteuert, einschließlich der Distanzmessung und der G-Kraft-Messungen. Die Distanzmessung ist für die Positionsermittlung notwendig, während die Beschleunigungskräfte-Messung für Forschungszwecke genutzt werden soll. Die Steuereinheit (+SE1) wird von der Batterieeinheit (+BE2) mit Energie versorgt.

Der Antrieb (+AE1) erfolgt über einen BLDC-Motor. Dieser wird mittels eines zusätzlichen Steuergeräts, einem Vector-Controller, angesteuert. Für die Energieversorgung des Antriebs wird eine Leistungsbatteie (+BE1) verwendet.

## Kapitel 3

# Stand der Technik

### To-do

- Bilder neu machen aus dem Schaltplan.
- Vector Controller, Funktionsweise nur kurz erklären und mehr auf die Anschlüsse eingehen.
- PIN-Mapping als Tabelle erstellen, für den Motor und den Vector Controller.
- QElectroTeck.
- PIN-Mapping als Tabelle erstellen, für den Motor und den Vector Controller.
- Auf die PDF verweisen.

Im folgenden Abschnitt werden die verwendeten Bauteile, welche in Kapitel 2 dargestellt worden sind, näher beschrieben.

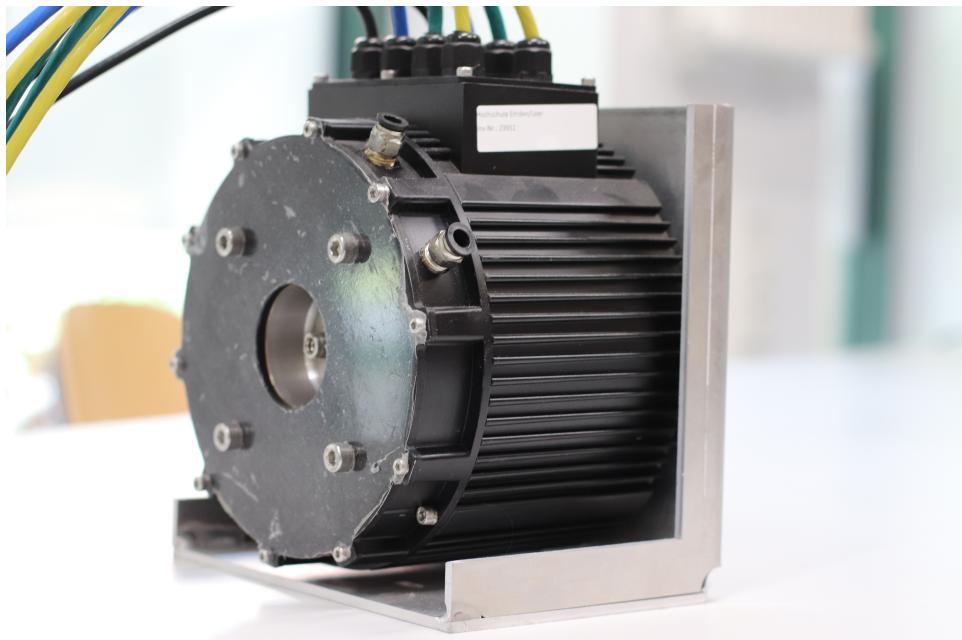
### 3.1 Antrieb – Golden Motor

Golden Motor bietet eine Gesamtlösung bestehend aus einem BLDC-Motor und einem Vektor-Controller an. Der Controller besitzt eine Schnittstelle für alle notwendigen Steuer- und Kontrollsignale, um den Motor anzutreiben. Die Anschlüsse sowie die grundlegende Funktionsweise werden in den Abschnitten 3.1.1 und 3.1.2 näher erläutert.

#### 3.1.1 BLDC Motor

Der in Abbildung 3.1 dargestellte BLDC-Motor verfügt über eine Leistung von 10 kW und kann optional durch eine Ölkühlung gekühlt werden, was

jedoch für das Fahrzeug derzeit nicht relevant ist. Die Drehzahl ist variabel und kann, mittels des Vector Controllers, welcher in Abschnitt 3.1.2 erklärt wird, zwischen 2.000 und 6.000 U/min eingestellt werden. Das Nenndrehmoment beträgt 26 Nm, während das maximale Drehmoment bei 85 Nm liegt. Der Wirkungsgrad des Motors beträgt 91 % [1].



**Abbildung 3.1:** Golden Motor – 10 KW BLDC Motor Liquid Cooled

Die Anschlüsse des Motors sind in Tabelle 3.1 aufgelistet. Der Motor verfügt über zwei Hall-Sensor-Kabel, die jeweils drei Hall-Sensoren sowie einen Temperatursensor enthalten. Zusätzlich sind ein GND- und ein +5V-Versorgungsanschluss in den Hall-Sensor-Kabeln integriert. Die Spulen des Motors werden über sechs separate Kabel angeschlossen: U, V und W.

**Tabelle 3.1:** Pin Mapping – BLDC-Motor

| <b>Anschluss</b>             | <b>Funktionalität</b> | <b>Farbe</b> |
|------------------------------|-----------------------|--------------|
| <b>Anschluss Adern Motor</b> |                       |              |
| U1                           | Spule 1               | Gelb         |
| V1                           | Spule 2               | Grün         |
| W1                           | Spule 3               | Blau         |
| U2                           | Spule 1               | Gelb         |
| V2                           | Spule 2               | Grün         |
| W2                           | Spule 3               | Blau         |
| <b>Motor Hall Kabel 1</b>    |                       |              |
| Hall A                       | Hall Sensor           | Gelb         |
| Hall B                       | Hall Sensor           | Grün         |
| Hall C                       | Hall Sensor           | Blau         |
| Temp                         | Temperatur Sensor     | Weiß         |
| +5V                          | Spannungsversorgung   | Rot          |
| GND                          | Masse                 | Schwarz      |
| <b>Motor Hall Kabel 2</b>    |                       |              |
| Hall A                       | Hall Sensor           | Gelb         |
| Hall B                       | Hall Sensor           | Grün         |
| Hall C                       | Hall Sensor           | Blau         |
| Temp                         | Temperatur Sensor     | Weiß         |
| +5V                          | Spannungsversorgung   | Rot          |
| GND                          | Masse                 | Schwarz      |

### 3.1.1.1 Funktionsweise

Ein BLDC-Motor (Brushless DC Motor) unterscheidet sich grundlegend von einem herkömmlichen Gleichstrommotor. Während bei einem traditionellen DC-Motor die Polumschaltung (Kommutierung) mechanisch über Kohlebürsten erfolgt, übernimmt beim BLDC-Motor eine elektronische Steuerung diese Aufgabe. Dadurch entfällt die Notwendigkeit von Kohlebürsten, was den Motor effizienter und langlebiger macht[2].

### 3.1.2 Vector Controller

Der Motor wird mithilfe eines Vektor-Controllers angesteuert, der in Abbildung 3.2 dargestellt ist. Der Controller hat die Aufgabe, den Motor basierend auf den Eingangssignalen so zu steuern, dass beispielsweise eine gewünschte Drehzahl erreicht wird. Die genaue Funktionsweise des Vektor-Controllers wird kurz unter 3.1.2.1 erläutert.

Das Pin-Mapping ist in Tabelle 3.2 aufgeführt. Bei den Steuersignalen muss auf den Spannungsbereich geachtet werden, da der Controller eine Störung meldet, wenn ein Signal außerhalb des zulässigen Bereichs liegt. Dies ist ein integriertes Sicherheitssystem des Controllers. Dies ist ein integriertes Sicherheitssystem des Controllers. So liegt beispielsweise bei der Bremse das logische Signal bei 0, wenn die Spannung 1 V beträgt. Tritt ein Kabelbruch ein, wird die Spannung am Controller bei 0 V liegen und eine Warnung wird durch ein periodisches Tonsignal ausgegeben.



**Abbildung 3.2:** Golden Motor – VECTOR 500 Motor Controller

**Tabelle 3.2:** Pin Mapping – Vector Controller

| Anschluss               | Funktionalität           | Farbe      | U in V     |
|-------------------------|--------------------------|------------|------------|
| <b>Batterie Eingang</b> |                          |            |            |
| B+                      | Versorgungsspannung      | RT         | 48 V       |
| B-                      | Masse                    | SW         | 0 V        |
| <b>Motor Ausgang</b>    |                          |            |            |
| U                       | Spule 1                  | GE         |            |
| V                       | Spule 2                  | GN         |            |
| W                       | Spule 3                  | BL         |            |
| <b>Steuerungskabel</b>  |                          |            |            |
| 1                       | CANL - CAN-Bus           | Vorhanden? |            |
| 2                       | CANH – CAN-Bus           | Vorhanden? |            |
| 3                       | Motor Temperatur         | BL         |            |
| 4, 5, 6                 | Masse                    | SW         | 0 V        |
| 7                       | Tempomat                 | Gr         | 0 V - 5 V  |
| 10                      | Elektronisches Schloss   | OR         | 0 V - 5 V  |
| 11                      | Hall C – Sensor          | BL         |            |
| 12                      | Hall B – Sensor          | GN         |            |
| 13                      | Hall A – Sensor          | GE         |            |
| 14                      | +5 V                     | RT         | 5 V        |
| 15                      | +12V Bremse              | GN&WS      | Vorhanden? |
| 16                      | Bremse                   | BL&WS      | 1 V - 5 V  |
| 20                      | Hohe Geschwindigkeit     | BL         | Vorhanden? |
| 21                      | Masse                    | SW         | 0 V        |
| 24                      | Niedrige Geschwindigkeit | BL         | Vorhanden? |
| 25                      | Masse                    | SW&WS      | 0 V        |
| 26                      | Gaspedal                 | GN&WS      | 1 V - 4 V  |
| 27                      | Masse                    | SW         | 0 V        |

### 3.1.2.1 Funktionsweise

Der Vektor-Controller verwendet die Motorsteuerungstechnologie der feldorientierten Regelung (Field-Oriented Control, FOC). Dabei wird die Position des Rotors im Motor mithilfe von Hall-Sensoren ermittelt. Mit der FOC-Technik ist es möglich, das Verhältnis von Drehmoment pro Ampere zu maximieren. Dafür muss der Rotorflussvektor im rechten Winkel (90 Grad) zum Statorflussvektor ausgerichtet sein [3].

## 3.2 Echtzeitfähige Steuerung – Speedgoat

### To-do

- I/O Module 691 zuende erklären.
- Echtzeitfähigkeit auf die DIN Norm beziehen und besser erklären auf unserer Situation.

Der Baseline Real-Time Target Machine, wie in Abbildung 3.3 dargestellt, ist ein echtzeitfähiges System, welches von der Firma Speedgoat hergestellt und vertrieben werden. Es wird hauptsächlich in der Entwicklung und beim Testen von komplexen Steuerungs- und Regelungssystemen eingesetzt, bei denen die Echtzeitfähigkeit eine zentrale Rolle spielt.



Abbildung 3.3: Speedgoat – Baseline Real-Time Target Machine

Speedgoat-Systeme integrieren sich nahtlos mit Simulink, einer Software von MathWorks, die häufig zur Modellierung und Simulation dynamischer Systeme verwendet wird. Durch eine große Auswahl an I/O-Modulen lassen sich verschiedene Sensoren und Aktuatoren problemlos an die Systeme anschließen. Echtzeitsysteme von Speedgoat sind Computer, die in der Lage sind, Aufgaben innerhalb festgelegter Zeitintervalle auszuführen. Diese Sys-

teme sind entscheidend für Anwendungen, bei denen die zeitliche Genauigkeit von großer Bedeutung ist, wie beispielsweise bei der Steuerung von Motoren, der Regelung von Prozessen oder der Simulation von physikalischen Systemen.



**Abbildung 3.4:** Speedgoat – Ansicht der Modulanschlüsse

In Abbildung 3.4 sind die Anschlüsse der I/O-Module des Speedgoat-Systems dargestellt. Die Zuordnung der Module zu den entsprechenden Anschlüssen ist in Tabelle 3.3 ersichtlich.

Das Modul IO397-50k-50k kann über ein Sensor-Aktor-Kabel von Phoenix Contact extern angeschlossen werden, wie in Abbildung 3.5 dargestellt. Zusätzlich wird das Modul IO691 mittels eines 9-Pin D-Sub-Steckers ange schlossen.

**Tabelle 3.3:** Speedgoat – Anschlüsse der Module

| I/O Module | Name | Funktion              |
|------------|------|-----------------------|
| 1          | A    | IO397-50k analog I/O  |
|            | B    | IO397-50k digital I/O |
| 2          | A    | IO691 CAN             |
|            | B    | IO691 CAN             |
| 3          | A    | IO397-50k analog I/O  |
|            | B    | IO397-50k digital I/O |
| 4          | A    | Frei                  |
|            | B    | Frei                  |

### 3.2.1 I/O-Modul – IO397-50k

Das in Abbildung 3.5 dargestellte IO397-50k I/O-Modul verfügt über zahlreiche analoge und digitale Ein- sowie Ausgänge. Die Anschlüsse werden auf Modul A, das die analogen Schnittstellen bereitstellt, und Modul B, welches die digitalen Schnittstellen umfasst, verteilt. Die Anschlussbelegung werden in den Tabellen 3.4 und 3.5 dargestellt. Die Spannungen der Ein- und Ausgänge können softwaregesteuert angepasst werden [4].

**Abbildung 3.5:** Speedgoat – I/O Module 397 [4]

#### 3.2.1.1 Pin Mapping – IO397-50k

Das Terminal Board A ist für die Verarbeitung analoger Signale konzipiert, sowohl für Eingänge als auch für Ausgänge. Die Tabelle 3.4 zeigt das Pin-Mapping von Terminal Board A. Dieses Board verfügt über Anschlüsse für insgesamt vier analoge Eingänge (Pins A1 bis A8) und vier analoge Ausgänge (Pins A9 bis A12). Die Ausgänge sind an einen Digital-zu-Analog-Wandler

(DAC) angeschlossen, während die Eingänge mit einem Analog-zu-Digital-Wandler (ADC) verbunden sind.

**Tabelle 3.4:** IO397-50k Pin Mapping – Terminal Board A: analog I/O [4, S. 15]

| Pin | Funktionalität   | Type      |
|-----|------------------|-----------|
| A1  | Analog input 1 + | ADC       |
| A2  | Analog input 1 - | ADC       |
| A3  | Analog input 2 + | ADC       |
| A4  | Analog input 2 - | ADC       |
| A5  | Analog input 3 + | ADC       |
| A6  | Analog input 3 - | ADC       |
| A7  | Analog input 4 + | ADC       |
| A8  | Analog input 4 - | ADC       |
| A9  | Analog output 1  | DAC       |
| A10 | Analog output 2  | DAC       |
| A11 | Analog output 3  | DAC       |
| A12 | Analog output 4  | DAC       |
| A13 | GND              | GND       |
| A14 | GND              | GND       |
| A15 | 0 V              | 0 V       |
| A16 | 5 V DC           | 5 V DC    |
| A17 | GND              | GND       |
| SH  | SH               | Shielding |

Terminal Board B hingegen bietet eine flexible Konfiguration für digitale Ein- und Ausgänge. Die Tabelle 3.5 zeigt das Pin Mapping von Terminal Board B. Die Pins B3 bis B16 sind für konfigurierbare I/O-Funktionalitäten ausgelegt und nutzen TTL-Signale (Transistor-Transistor-Logik), was sie besonders für schnelle digitale Schaltvorgänge geeignet macht. Dieses Board ermöglicht es, die Anschlüsse flexibel zu nutzen, je nach den Anforderungen der Applikation. Auch hier gibt es mehrere Spannungs- und Ground-Anschlüsse, sowie eine Abschirmung (Shielding) für das M12-Kabel, um elektromagnetische Störungen zu minimieren.

**Tabelle 3.5:** IO397-50k Pin Mapping – Terminal Board B: analog I/O [4, S. 15]

| Pin | Funktionalität | Type      |
|-----|----------------|-----------|
| B1  | 0 V            | 0 V       |
| B2  | 5 V DC         | 5 V DC    |
| B3  | I/O 0          | IN/OUT    |
| B4  | I/O 1          | IN/OUT    |
| B5  | I/O 2          | IN/OUT    |
| B6  | I/O 3          | IN/OUT    |
| B7  | I/O 4          | IN/OUT    |
| B8  | I/O 5          | IN/OUT    |
| B9  | I/O 6          | IN/OUT    |
| B10 | I/O 7          | IN/OUT    |
| B11 | I/O 8          | IN/OUT    |
| B12 | I/O 9          | IN/OUT    |
| B13 | I/O 10         | IN/OUT    |
| B14 | I/O 11         | IN/OUT    |
| B15 | I/O 12         | IN/OUT    |
| B16 | I/O 13         | IN/OUT    |
| B17 | GND            | GND       |
| B18 | SH             | Shielding |

### 3.2.1.2 IO397-50k – Konfigurieren

In dem Driver Block, wie in Abbildung 3.6 gezeigt, können am dem Pull-Widerstand des I/O-Moduls verschiedene Spannungen eingestellt werden. Zur Auswahl stehen:

- Pull-up 3,3 V
- Weak Pull-up 5,0 V
- Pull-down
- Floating

welche im Folgenden erklärt werden.

Pull-Widerstände werden verwendet, um klare digitale Schaltzustände zu gewährleisten. Es gibt drei verschiedene Zustände: »High« entspricht »logisch 1« Low entspricht »logisch 0«. »Floating« hat die Bedeutung, dass der Widerstand nicht an »GND« oder an einer Spannung angeschlossen ist. So mit ist der Widerstand hochohmig. Die Zustände zu den Einstellungen sind in der Tabelle 3.6 aufgeführt. In Abbildung 3.7 ist im rot markierten Bereich der Pull-Widerstand mit  $4,7\text{ k}\Omega$  an einen  $47\text{-}\Omega$ -Widerstand am I/O-Anschluss



Abbildung 3.6: Simulink – Driver Block – IO397-50k

angeschlossen. Der Pull-Widerstand kann, wie bereits erwähnt, konfiguriert werden. Dabei wird intern am Pull-Widerstand entweder eine Spannung von 3,3 V, 5 V, GND oder kein Kontakt, also Floating (hochohmig), geschaltet.

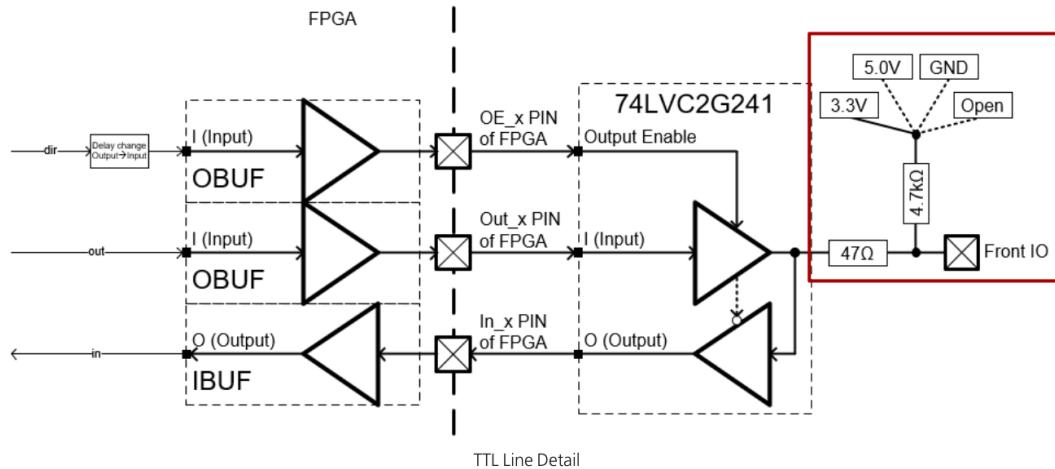
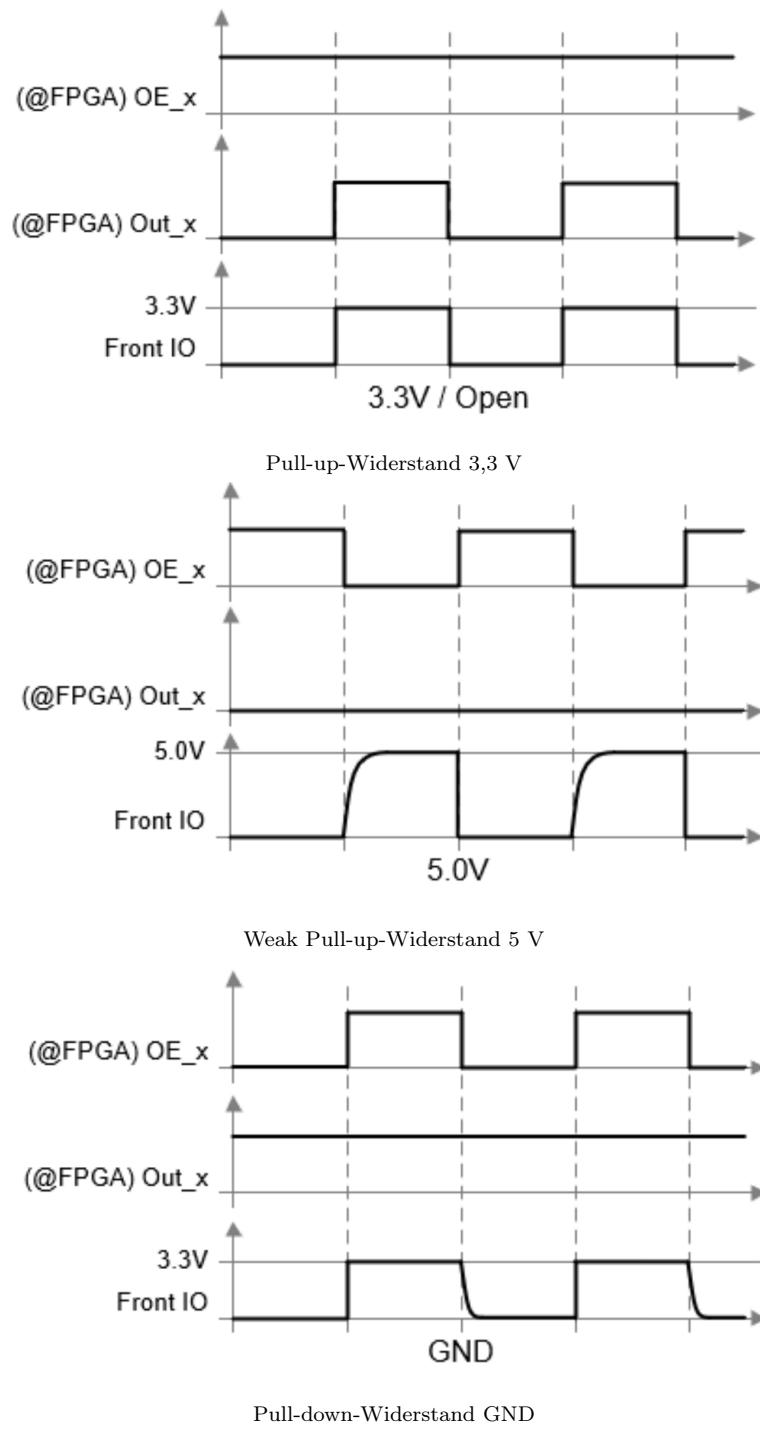


Abbildung 3.7: Speedgoat – TTL I/O Interface – IO397-50k [4, S. 12]

Bei der Einstellung auf 3,3 V oder 5 V fungiert der Widerstand als Pull-up-Widerstand. Bei der Pull-down-Konfiguration liegt am Widerstand Ground (GND) an.

In Tabelle 3.6 können die Reaktionszeiten der digitalen Ausgänge verglichen werden. Bei der Einstellung »Weak Pull-up 5,0 V« bedeutet »weak«, dass der Ausgang eine langsamere Anstiegszeit hat. In Abbildung 3.8 sind die Reaktionszeiten dargestellt.



**Abbildung 3.8:** Simulink – Pull-Widerstände – IO397-50k [4, S. 12]

**Tabelle 3.6:** Simulink – Pull-Widerstand – Reaktionszeit

| Einstellung       | Logischer zustand | Reaktionszeit der Ausgänge |         |
|-------------------|-------------------|----------------------------|---------|
|                   |                   | Anstieg                    | Abfall  |
| Pull-up 3,3V      | I/O = 1           | Schnell                    | Schnell |
| weak pull-up 5,0V | I/O = 1           | Langsam                    | Schnell |
| Pull-down         | I/O = 0           | Schnell                    | Langsam |
| Floating          | I/O = 0 - 1       | -                          | -       |

### 3.2.2 I/O-Modul – IO691

Das IO691 I/O-Modul bietet eine intelligente CAN-Schnittstelle mit zwei Kanälen, die sowohl flexible Datenrate CAN (CAN FD) als auch High-Speed CAN (CAN HS) unterstützen. Es ist kompatibel mit CAN 2.0A/B-Netzwerken und unterstützt SAE J1939 sowie ASAM XCP für Bypassing. Alle Signale sind über 9-polige D-Sub-Front-CAN-Anschlüsse zugänglich [5, S. 4].

**Abbildung 3.9:** Speedgoat – I/O Module 691 [5]

### 3.2.2.1 Pin Mapping – IO691

Tabelle 3.7: IO691 Pin Mapping

| Pin | Farbe   | An Sensor       | DB9 Connector A/B, Signal |
|-----|---------|-----------------|---------------------------|
| 1   | Braun   | PIN 2 - Weiß    | Vcc                       |
| 2   | Rot     | PIN 5 - Grau    | CAN-low                   |
| 3   | Orange  | PIN 3 - Blau    | GND                       |
| 4   | Gelb    |                 | -                         |
| 5   | Grün    |                 | -                         |
| 6   | Blau    | PIN 3 - Blau    | GND                       |
| 7   | Viollet | PIN 4 - Schwarz | CAN-high                  |
| 8   | Grau    |                 | -                         |
| 9   | Schwarz |                 | -                         |
| SH  |         | PIN 1 - Braun   | Shield                    |

### 3.2.2.2 IO691 Module Konfigurieren

Im »Driver Block«, siehe Abbildung 3.10, können die »CAN Channels« 1 und 2 konfiguriert werden. Wie bereits in Abschnitt 3.2.2 erwähnt, lässt sich der Protokollmodus jedes Kanals separat einstellen, wobei zwischen »CAN HS«, »CAN FD« und »Disable« gewählt werden kann. Dabei entspricht »CAN HS« der Norm ISO 11898-2 und »CAN FD« der Norm ISO 11898-1 [5, S. 4].

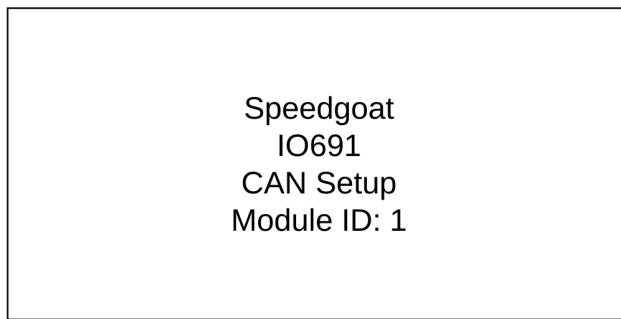


Abbildung 3.10: Simulink – Driver Block – I/O Module 691 [5]

Die »Initialization«- und »Termination«-CAN-Nachrichten können individuell angepasst werden. Im nachfolgenden Matlab-Skript (3.1) werden die Variablen »initCAN« und »termCAN« definiert.

**Listing 3.1:** Initialization and Termination Structure [6]

```
% create Initialization structure
initCAN.Channel      = 1;                                % 'CAN' or 'CAN-FD'
initCAN.Protocol     = 'CAN-FD';
initCAN.BRS          = 1;                                % 0 or 1
initCAN.Type         = 'Standard';                      % 'Standard' or 'Extended'
initCAN.Identifier   = 99;                               % ID must be double
initCAN.Data         = [8 8 8 8 8 8 8];
initCAN.Pause        = 0.05;

% create Termination structure
termCAN.Channel      = 1;                                % 'CAN' or 'CAN-FD'
termCAN.Protocol     = 'CAN-FD';
termCAN.BRS          = 1;                                % 0 or 1
termCAN.Type         = 'Extended';                      % 'Standard' or 'Extended'
termCAN.Identifier   = double(0xFFFF);                  % ID must be double
termCAN.Data         = [1 1 1 1 1 1 1];
termCAN.Pause        = 0.05
```

Die Baudrate kann in zwei Kategorien unterteilt werden: »Nominal Baud Rate« und »Data Baud Rate«. Wenn im »CAN Channels«-Menü der Modus »CAN FD« ausgewählt wird, können beide Baudraten konfiguriert werden. Im Modus »CAN HS« ist die Einstellung der »Nominal Baud Rate« möglich. Es können entweder die voreingestellten Baudraten verwendet oder eigene Werte über ein Array [BRP, SJW, TSEG1, TSEG2] konfiguriert werden, deren Elemente im Folgenden kurz erläutert werden, siehe Tabelle 3.8.

**Tabelle 3.8:** Simulink – Baudraten Parameter

| Abkürzung | Bedeutung                  |
|-----------|----------------------------|
| BRP       | Bit Rate Prescaler         |
| SJW       | Synchronization Jump Width |
| TSEG1     | Time Segment 1             |
| TSEG2     | Time Segment 2             |

Die sogenannte »CAN Nominal Bit Time«, wie in Abbildung 3.11 dargestellt, besteht aus vier Segmenten: dem »Sync Segment«, dem »Propagation Segment«, dem »Phase Buffer Segment 1« und dem »Phase Buffer Segment 2«. Das »Sync Segment« dient zur Synchronisation der Knoten auf dem Bus und hat eine Länge von genau einem Zeitquantum. Im »Propagation Segment« werden die physikalischen Verzögerungen im Busnetzwerk ausgeglichen [7, S. 1].

Im »Phase Buffer Segment 1« werden positive Phasenfehler kompensiert, und das Segment kann während der Resynchronisation verlängert werden.

Im »Phase Buffer Segment 2« hingegen werden negative Phasenfehler kompensiert, und das Segment kann während der Resynchronisation verkürzt werden.

Der »BRP« ist ein Parameter, der das Zeitquant  $t_q$  festlegt siehe Gleichung 3.1. Der »SJW« ist ein Parameter, der die Länge von »TSEG1« verlängern und die Länge von »TSEG2« verkürzen kann, um die Synchronisation zwischen den Knoten zu gewährleisten, dabei darf »SJW« kleiner gleich »TSEG2« sein [6].

Der Sample Point (SP) ist der Punkt innerhalb eines CAN-Bits, an dem die tatsächliche Datenabtastung stattfindet. Dieser Punkt kann durch das Verhältnis zwischen »TSEG1« und »TSEG2« angepasst werden.

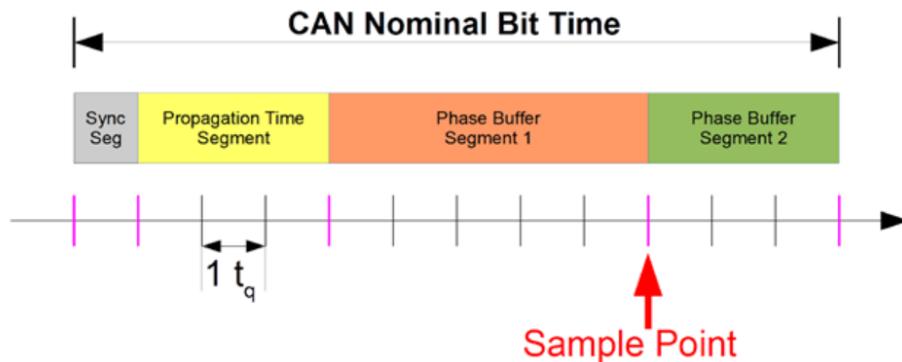


Abbildung 3.11: Simulink – Baudrate – I/O Module 691 [6]

Mit den weiteren Gleichungen können die Zeiten innerhalb des »CAN Nominal Bit Time« ermittelt werden.

$$t_q = \frac{BRP}{CAN_{Clk}} \quad (3.1)$$

$$t_{BS1} = t_q \cdot TSEG1 \quad (3.2)$$

$$t_{BS2} = t_q \cdot TSEG2 \quad (3.3)$$

$$\begin{aligned} NominalBitTime &= t_q + t_{BS1} + t_{BS2} \\ &= t_q \cdot (1 + TSEG1 + TSEG2) \end{aligned} \quad (3.4)$$

$$CAN_{Baudrate} = \frac{1}{NominalBitTime} \quad (3.5)$$

$$\begin{aligned} SP &= \frac{(1 + TSEG1)}{(1 + TSEG1 + TSEG2)} \\ &= \frac{(t_q + t_{BS1})}{NominalBitTime} \end{aligned} \quad (3.6)$$

Am CAN-Bus ist es wichtig, an beiden Enden Abschlusswiderstände von  $120 \Omega$  zu installieren. Diese Widerstände verhindern Reflexionen von Signalen, die auf der Leitung entstehen können, wenn die Impedanz des Busses nicht korrekt abgestimmt ist.

Reflexionen können das ursprüngliche Signal beeinträchtigen und zu Fehlern in der Datenkommunikation führen. Die Abschlusswiderstände gewährleisten eine angepasste Impedanz und minimieren somit die Wahrscheinlichkeit von Signalreflexionen [6].

## Kapitel 4

# Implementierung

### 4.1 Schaltplan

To-do

- Funktion (=Anlage) und Ortskennzeichen (+Ort) definieren.  
Funktionales Engineering.
- Bauteilkennzeichnung (BMK) festlegen.
- Aderfarben und Querschnitt definieren.
- Ausgewählte Bauteile in die Schaltung integrieren.
- Nicht vorhandene Bauteile selbst erstellen.
  - Bauteile sortiert hinzufügen: Spannungsversorgung, Steuerung (Eingang, dann Ausgang), Lastkreis und Sonderfunktionen.

Zu Beginn der Erstellung des Schaltplans sollten die Funktions- und Ortskennzeichen sowie die Beschriftung der Bauteile und Leitungen festgelegt werden. Bei der Verdrahtung ist es erforderlich, die Aderfarben den entsprechenden Spannungspotenzialen zuzuordnen.

#### 4.1.1 Betriebsmittelbezeichnung

Ein Betriebsmittel besteht aus verschiedenen Kennzeichen: dem Funktionskennzeichen (=), dem Ortskennzeichen (+) und dem Betriebsmittelkennzeichen (-). Früher wurde das Funktionskennzeichen als „Anlage“ bezeichnet. In der neuen DIN EN IEC 81346-2 [8] wurde diese Bezeichnung jedoch auf „Funktion“ geändert.

#### 4.1.1.1 Funktionskennzeichen

Das Funktionskennzeichen beschreibt eine bestimmte Funktion im Schaltplan, wie beispielsweise die Spannungsversorgung oder die Steuerung (siehe Tabelle 4.1).

**Tabelle 4.1:** Schaltplan – Funktionskennzeichen (=)

| Abkürzung | Bezeichnung         |
|-----------|---------------------|
| SV        | Spannungsversorgung |
| ES        | Eingänge Steuerung  |
| AS        | Ausgänge Steuerung  |
| KO        | Kommunikation       |
| AT        | Antrieb             |
| NA        | Not-Aus             |

#### 4.1.1.2 Ortskennzeichen

Das Ortskennzeichen gibt an, an welchem Ort ein Bauteil installiert ist. Beispielsweise gibt es im Fahrzeug eine Steuereinheit und eine Batterieeinheit. Im Schaltplan wird dadurch deutlich, welche Verbindungen zwischen verschiedenen Orten bestehen. Das Ortskennzeichen erleichtert zudem die Wartung und Installation des Systems.

**Tabelle 4.2:** Schaltplan – Ortskennzeichen (+)

| Abkürzung | Bezeichnung     |
|-----------|-----------------|
| POD       | Fahrzeug        |
| BE        | Batterieeinheit |
| SE        | Steuereinheit   |
| AE        | Antriebseinheit |

#### 4.1.1.3 Betriebsmittelkennzeichen

Das Betriebsmittelkennzeichen bezeichnet das spezifische Bauteil. Die genaue Zuordnung der Bezeichnungen ist in der DIN EN IEC 81346-2 geregelt. Die für das Projekt relevanten Betriebsmittelkennzeichen sind in Tabelle 4.3 aufgeführt.

Bei einer vollständigen Betriebsmittelbezeichnung sieht das Kennzeichen des Betriebsmittels folgendermaßen aus:

=ES2+SE1-10QA1

**Tabelle 4.3:** Schaltplan – Betriebsmittelkennzeichen (-) [8]

| <b>Abkürzung</b> | <b>Klassenname</b>   | <b>Allgemeine Bedeutung</b> |
|------------------|--|-----------------------------|
| FC               | Überstromschutzobjekt  | Sicherung                   |
| GB               | Erzeugungsobjekt für elektrische Energie durch chemische Energie | Batterie                    |
| KE               | Elektrische Signale verarbeitendes Objekt                        | Steuerung                   |
| MA               | Elektromagnetisches Rotationsantriebsobjekt                      | Motor                       |
| QA               | Stromsteuerungsobjekt  | Relais                      |
| RL               | Bewegungsbegrenzungsobjekt                                       | Bremse                      |
| SF               | Gesichtsinteraktionsobjekt                                       | Schalter                    |
| TB               | Stromkonvertierungsobjekt  | Transformator               |
| WD               | Niederspannungsenergie Leitobjekt                                | Leitung/Kabel               |
| XD               | Niederspannungs-Verbindungsobjekt                                | Klemme, Stecker oder Buchse |

Die Betriebsmittelbezeichnung =ES2+SE1-10QA4 besagt, dass es sich um die Eingänge der Steuerung (=ES) handelt. Hierbei gibt es zwei Funktionen. Der Ort ist in der Steuereinheit (+SE) verortet, und das Betriebsmittel selbst ist ein Relais, welche auf der Seite 10 des Schaltplans zu finden ist.

Das Kennzeichen umfasst sowohl das Funktions- als auch das Ortskennzeichen, die jeweils mit einer fortlaufenden Nummer versehen sind. Das Betriebsmittelkennzeichen enthält zwei Nummerierungen: Die erste gibt an, auf welcher Seite des Schaltplans sich das Betriebsmittel befindet, während die zweite die fortlaufende Nummerierung des Bauteils darstellt.

Im Schaltplan wird das Betriebsmittel lediglich durch das Betriebsmittelkennzeichen (-) dargestellt, während das Funktions- und Ortskennzeichen im Zeichnungskopf angegeben ist.

#### 4.1.2 Verdrahtung Konventionen

Für die Verdrahtung müssen den Spannungspotenzialen verschiedene Farben zugewiesen werden. Diese Zuordnungen sind in Tabelle 4.4 aufgeführt.

**Tabelle 4.4:** Verdrahtung Konvention – Aderfarben

| Name     | Fabe | U in V | A in mm <sup>2</sup> | Funktion |
|----------|------|--------|----------------------|----------|
| Rot      |      | x      |                      | 1        |
| Blau     |      | x      |                      | 1        |
| Schwarz  |      | x      |                      | 1        |
| Schwarz  |      | < 48   | 50                   |          |
| Viollett |      | x      |                      | 1        |

## 4.2 Simulation mit Simulink

To-do

.

### 4.2.1 Funktionsweise der Steuerung

Die Steuerung des Fahrzeugs bietet die Möglichkeit, zwischen Automatik- und manuellem Betrieb zu wählen. Im manuellen Betrieb kann der Benutzer das Fahrzeug sowohl vorwärts als auch rückwärts fahren lassen, wobei zwei Geschwindigkeitsstufen – schnell und langsam – zur Verfügung stehen. Diese Funktion soll die möglichkeit geben<sup>666</sup>

Der Automatikmodus wird über den Startknopf aktiviert. Das Fahrzeug beschleunigt auf die einstellbare Geschwindigkeit und bremst automatisch ab, sobald die Distanz zum Enddeckel der Röhre kleiner wird. Es hält schließlich am Ende der Strecke an.

#### 4.2.1.1 Controllpanel

Über das Control Panel (siehe Abbildung 4.1) können alle Eingaben via Simulink gesteuert werden. Mit dem Wahlschalter kann, wie bereits erwähnt, zwischen Automatik- und manuellem Betrieb gewechselt werden. Ist der manuelle Betrieb ausgewählt, kann mit dem Drehschalter (Manuell) zwischen Vorwärts, Rückwärts und Stopp geschaltet werden (siehe Tabelle 4.5, die Werte werden später für die Automatensteuerung benötigt).

**Tabelle 4.5:** Automat – Wahlschalter für den manuellen Betrieb

| Schalterbezeichnung | Zustände       | Wert |
|---------------------|----------------|------|
| Rück+               | Forwards fast  | 1    |
| Rück                | Forwards slow  | 2    |
| STOP                | Manual idle    | 3    |
| Vor                 | Backwards slow | 4    |
| Vor+                | Backwards fast | 5    |

Im Automatikbetrieb kann die Geschwindigkeit über das Drehrad »Geschwindigkeit Automatik« voreingestellt oder während der Fahrt angepasst werden. Das Fahrzeug beginnt zu fahren, sobald der Druckknopf »Start« betätigt wird. Die Automatik kann über den Druckknopf »Stop« gestoppt

werden.

Im Notfall kann sowohl im Automatik- als auch im manuellen Betrieb der Druckknopf »Emergency stop« betätigt werden. Das Fahrzeug führt eine Vollbremsung mit der Motor- und der mechanischen Bremse durch. Der Notfallmodus kann durch Drücken des »Reset« -Knopfs beendet werden.

Die Funktionsweise der manuellen Steuerung wird in Abschnitt 4.2.2.1 erklärt, und die der automatischen Steuerung in Abschnitt 4.2.2.2

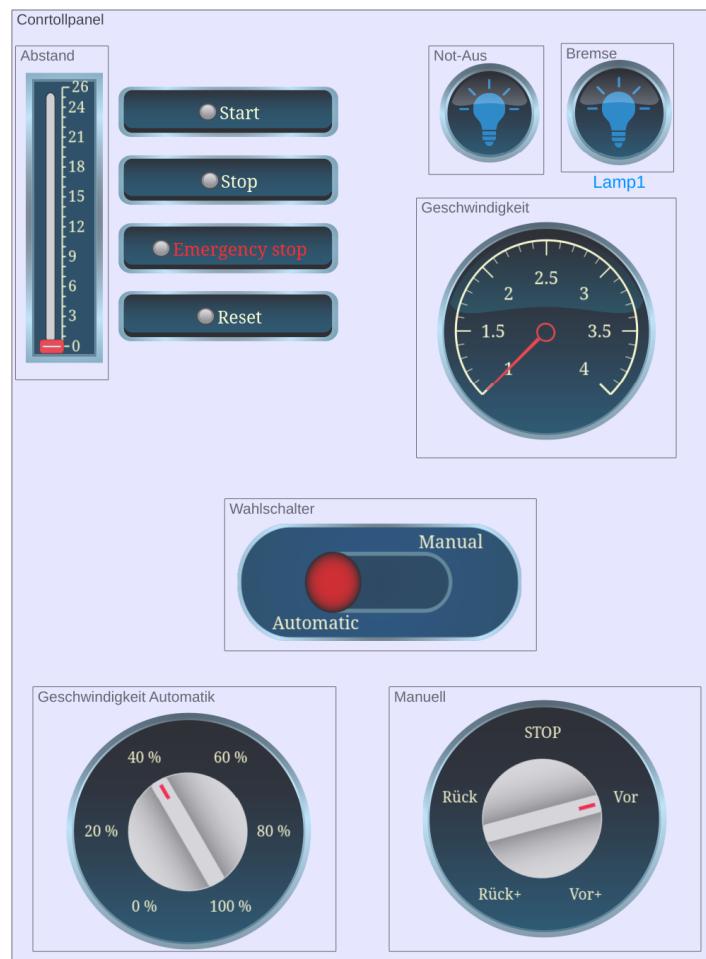


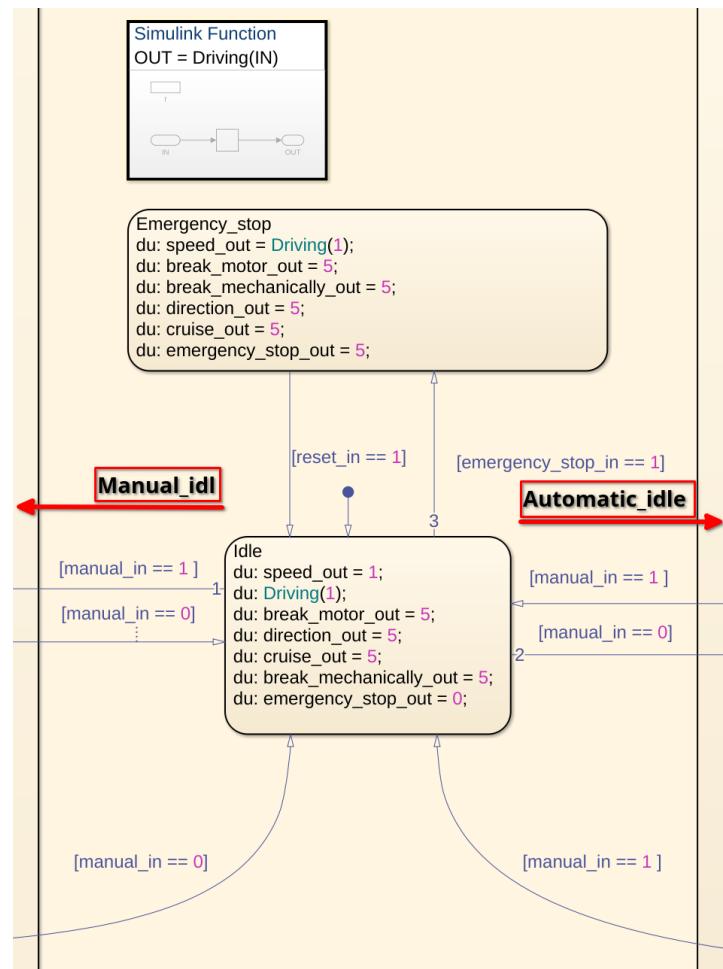
Abbildung 4.1: Automat – Conrtolpanel

### 4.2.2 Automatensteuerung

Die Automatensteuerung beschreibt die logische Steuerung eines Systems durch einen Automaten, wie etwa einen Moore- oder Mealy-Automaten, der auf Basis von Eingaben und aktuellen Zuständen Ausgaben steuert.

Das Fahrzeug wird mittels eines Moore-Automaten gesteuert, wobei die Ausgänge erst dann aktiviert werden, wenn sich das System im entsprechenden Zustand befindet.

Der Automat startet im Zustand »Idle« siehe Abbildung 4.6, und die Abläufe für den automatischen oder manuellen Modus werden von »Idle« aus angesteuert. Im Zustand Idle werden die Ausgänge (siehe Tabelle 4.6) geschaltet.

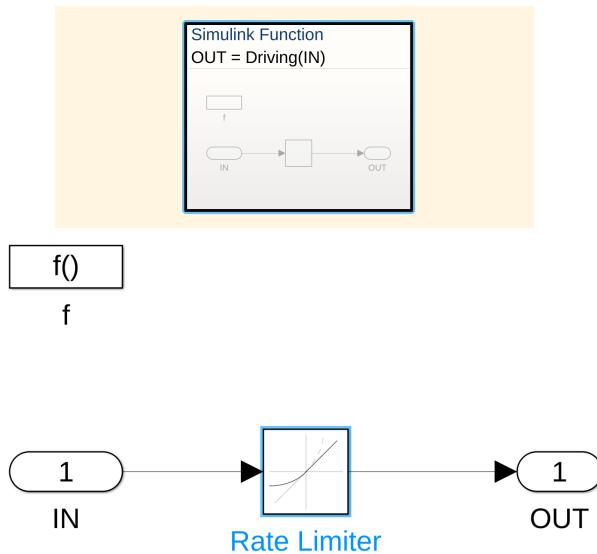


**Abbildung 4.2:** Automat – Start Zustand »Idle«

**Tabelle 4.6:** Ausgänge – Idle

| Ausgänge               | I/O Module     | Wert  |
|------------------------|----------------|-------|
| speed out              | IO397-50k – A1 | 1     |
| Driving(1)             |                | Reset |
| break motor out        | IO397-50k – B3 | 5     |
| direction out          | IO397-50k – B4 | 5     |
| cruise out             | IO397-50k – B5 | 5     |
| break mechanically out | IO397-50k – B6 | 5     |
| emergency stop out     | IO397-50k – B7 | 0     |

Die Funktion »Driving«, siehe Abbildung 4.3, erhält einen Eingangswert, der durch einen »Rate Limiter« begrenzt und am Ausgang ausgegeben wird. Die Funktion speichert den letzten Wert, daher muss sie in den Zuständen, in denen das Fahrzeug angehalten wird, zurückgesetzt werden. Dies geschieht, indem die Funktion »Driving« den Wert »1« erhält.

**Abbildung 4.3:** Automat – Funktion Driving

Der Zustand »Emergency stop« kann von jedem Zustand aus durch den Eingang »emergency stop in« erreicht werden, welcher sowohl über das Controlpanel als auch extern über das IO397-50k – B11 ausgelöst werden kann. Der Zustand kann nur durch ein Reset-Signal verlassen werden und führt dann in einen der Idle-Zustände zurück: »Idle« »Manual idle« oder »Automatic idle«. Im Zustand »Emergency stop« werden die Ausgänge gemäß Tabelle 4.7 geschaltet.

**Tabelle 4.7:** Ausgänge – Emergency stop

| Ausgänge               | I/O Module<br>IO397-50k | Wert  |
|------------------------|-------------------------|-------|
| speed out=Driving()    | A1                      | Reset |
| break motor out        |                         | 5     |
| direction out          | B4                      | 5     |
| cruise out             | B5                      | 5     |
| break mechanically out | B6                      | 5     |
| emergency stop out     | B7                      | 5     |

#### 4.2.2.1 Automat für die manuelle Steuerung

Die manuelle Steuerung bietet dem Nutzer die Flexibilität, das Fahrzeug sowohl innerhalb als auch außerhalb der Röhre zu steuern.

Der Automat des manuellen Ablaufs, wie in Abbildung 4.4 dargestellt, ist in zwei Stateflows unterteilt: »Forwards« und »Backwards«. In Abbildung 4.5 ist der »Forwards« -Stateflow zu sehen, der »Backwards« -Stateflow ist äquivalent, wobei sich lediglich der Ausgang »Direction Out« ändert. Daher wird nur der »Forwards« -Stateflow erklärt. Der Zustand »Emergency stop 1« ist der gleiche Zustand wie »Emergency stop«. Die Ausgänge für die Zustände »Forwards« und »Backwards« sind in der Tabelle 4.8 für die Zustände »slow« und in der Tabelle 4.9 für die Zustände »fast« aufgelistet.

Mit dem Wahlaster (siehe Tabelle 4.5) kann zwischen den Zuständen gewechselt werden. In der Abbildung 4.5 ist der Ausschnitt des »Forwards« -Stateflow zu sehen. Wenn sich der Automat in dem Stateflow »Forwards« befindet, gibt es keine direkte Verknüpfung zu dem »Backwards« Stateflow, somit muss man erst in den Zustand »Manual idle« wechseln, um in den Stateflow »Backwards« zu gelangen.

Der Zustand »Collector manual in« fängt das Schaltsignal des Wahlschalters ab, wenn mitten im Betrieb auf Automatik geschalten wird und führt zu dem »Idle « Zustand.

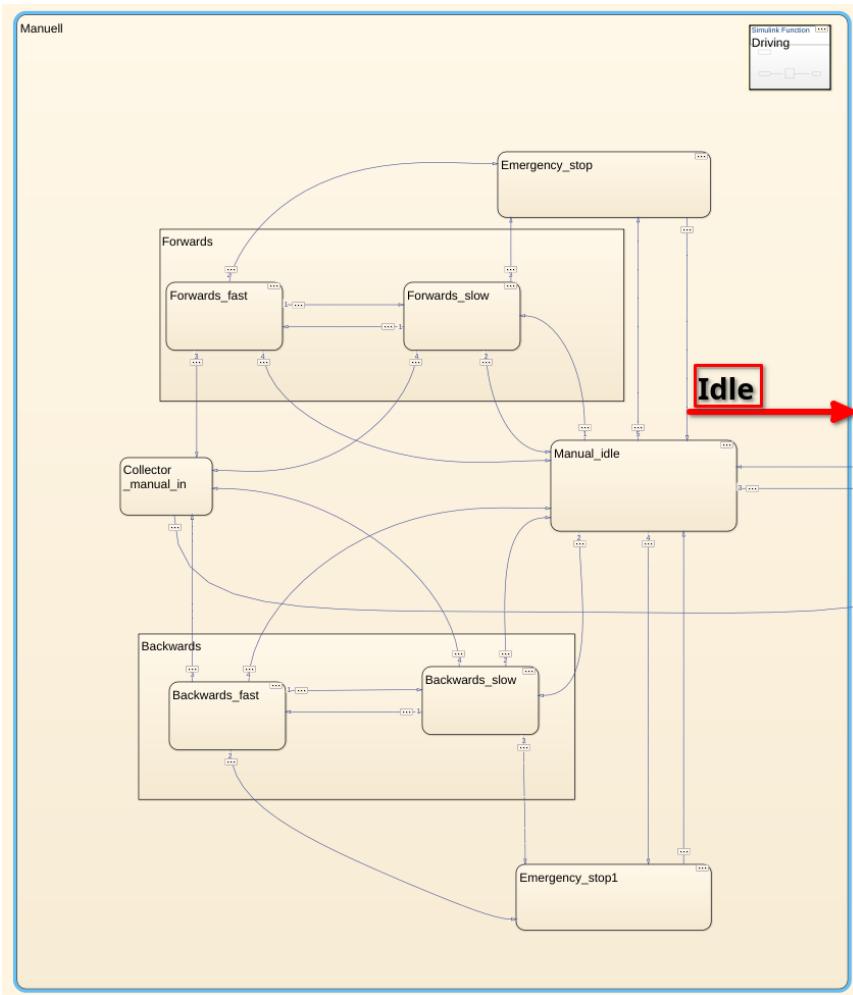


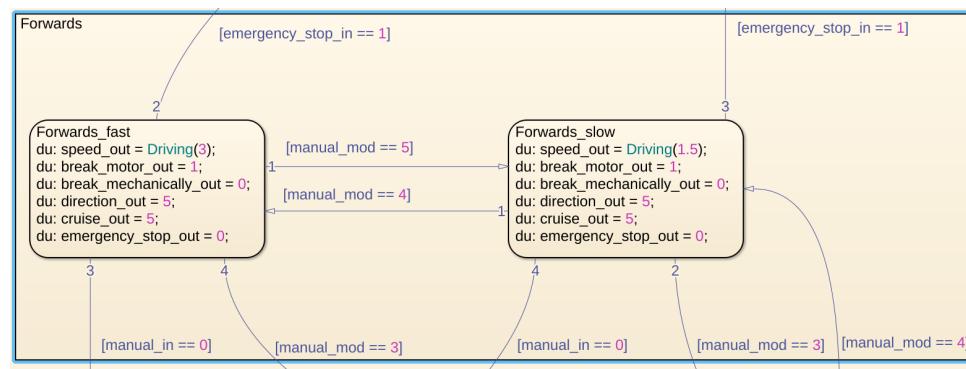
Abbildung 4.4: Automat – Übersicht der manuellen Steuerung

Tabelle 4.8: Ausgänge – der Zustände »Forwards« und »Backwards« – »slow«

| Ausgänge               | I/O Module<br>IO397-50k | Werte<br>»Forwards« | Werte<br>»Backwards« |
|------------------------|-------------------------|---------------------|----------------------|
| speed out=Driving(1,5) | A1                      | 1,5                 | 1,5                  |
| break motor out        | B3                      | 1                   | 1                    |
| direction out          | B4                      | 5                   | 0                    |
| cruise out             | B5                      | 5                   | 5                    |
| break mechanically out | B6                      | 0                   | 0                    |
| emergency stop out     | B7                      | 0                   | 0                    |

**Tabelle 4.9:** Ausgänge – der Zustände »Forwards« und »Backwards« – »fast«

| Ausgänge               | I/O Module<br>IO397-50k | Werte      |             |
|------------------------|-------------------------|------------|-------------|
|                        |                         | »Forwards« | »Backwards« |
| speed out=Driving(3)   | A1                      | 3          | 3           |
| break motor out        | B3                      | 1          | 1           |
| direction out          | B4                      | 5          | 0           |
| cruise out             | B5                      | 5          | 5           |
| break mechanically out | B6                      | 0          | 0           |
| emergency stop out     | B7                      | 0          | 0           |

**Abbildung 4.5:** Automat – »Forwards« -Stateflow der manuellen Steuerung

#### 4.2.2.2 Automat für die automatische Steuerung

Die automatische Steuerung soll das Fahrzeug innerhalb der Röhre bis zur eingestellten Geschwindigkeit beschleunigen. Ab einer bestimmten Distanz zum Ende der Röhre soll es allmählich abbremsen und schließlich zum Stillstand kommen.

Der Automat »Automatic« besteht aus den Zuständen »Automatic«, »Drive«, »Distance«, »Stop« und »Emergency stop«, die in Abbildung 4.6 dargestellt sind.

Der Zustand »Automatic« beschreibt den Idle-Zustand der automatischen Steuerung. Von hier aus wird der Zustand »Drive« angesteuert, sobald der Startknopf des »Control Panels« (siehe Abbildung 4.1) betätigt wird. Die Ausgänge werden gemäß Tabelle 4.10 gesetzt, solange der Zustand »Automatic« nicht verlassen wird.

Im Zustand »Drive« wird die Geschwindigkeit vom Nutzer mittels des Drehrads »Geschwindigkeit Automatik« eingestellt. Wird hingegen der »Stop« Taster betätigt, wird der Zustand »Stop« angesteuert. Ab einer Distanz von 10 m wird der Zustand »Distance« aktiviert. In Tabelle 4.11 sind die Aus-

gänge des Zustands »Drive« dargestellt.

Wie bereits erwähnt, wird die Geschwindigkeit in Abhängigkeit zum Abstand »distance« vom Ende der Röhre verringert, hier bei wird in dem Zustand »Distance« der Augang der Geschwindigkeit mittels der Gleichung:

$$speed_{out} = distance_{in} \cdot \frac{speed_{in} - 1}{distance} + 1 \quad (4.1)$$

ermittelt.

Soll die Automatik beendet werden, kann dies über den Stop-Taster auf dem Control Panel ausgelöst werden. Dabei werden die in Tabelle 4.13 aufgeführten Signale geschaltet.

**Tabelle 4.10:** Ausgänge – Automatic

| Ausgänge               | I/O Module<br>IO397-50k | Wert      |
|------------------------|-------------------------|-----------|
| speed out              | A1                      | Driving() |
| Driving(1)             |                         | Reset     |
| break motor out        | B3                      | 5         |
| direction out          | B4                      | 5         |
| cruise out             | B5                      | 5         |
| break mechanically out | B6                      | 5         |
| emergency stop out     | B7                      | 5         |

**Tabelle 4.11:** Ausgänge – Drive

| Ausgänge               | I/O Module<br>IO397-50k | Wert              |
|------------------------|-------------------------|-------------------|
| speed out              | A1                      | Driving(speed in) |
| Driving(speed in)      |                         | speed in          |
| break motor out        | B3                      | 5                 |
| direction out          | B4                      | 5                 |
| cruise out             | B5                      | 5                 |
| break mechanically out | B6                      | 5                 |
| emergency stop out     | B7                      | 5                 |

**Tabelle 4.12:** Ausgänge – Distance

| Ausgänge               | I/O Module<br>IO397-50k | Wert        |
|------------------------|-------------------------|-------------|
| speed out              | A1                      | gl. 4.1     |
| Driving(1)             |                         | distance in |
| break motor out        | B3                      | 1           |
| direction out          | B4                      | 5           |
| cruise out             | B5                      | 5           |
| break mechanically out | B6                      | 0           |
| emergency stop out     | B7                      | 0           |

**Tabelle 4.13:** Ausgänge – Stop

| Ausgänge               | I/O Module<br>IO397-50k | Wert      |
|------------------------|-------------------------|-----------|
| speed out              | A1                      | Driving() |
| Driving(Restet)        |                         | Reset     |
| break motor out        | B3                      | 1         |
| direction out          | B4                      | 5         |
| cruise out             | B5                      | 5         |
| break mechanically out | B6                      | 0         |
| emergency stop out     | B7                      | 0         |

## 4.3 Distanzmessung

Der Sensor von Pepperl+Fuchs wurde ursprünglich für die Distanzmessung angeschafft. Da dieser Sensor jedoch sehr teuer ist und wir nicht das Risiko eines möglichen Schadens im Vakuum eingehen wollten, wurde nach einer kostengünstigen Alternative gesucht.

Die Idee bestand darin, ein Entfernungsmessgerät von PARKSIDE zu verwenden und den Sensor aus dem Gerät zu entfernen. Die Messwerte werden anschließend mit einer MCU zu decodiert.

### 4.3.1 Verbindung

Der Sensor ist über ein Flachbandkabel mit der Hauptplatine verbunden, wie in Abbildung 4.7 zu sehen ist. Auf der Hauptplatine befinden sich vier ungenutzte Lötstellen. Mithilfe eines Oszilloskops wurden diese Lötstellen analysiert und es wurde festgestellt, dass Leitung »1« eine Spannung von 3,3 V führt und Leitung »4« als GND dient. Die Leitungen »2« und »3«

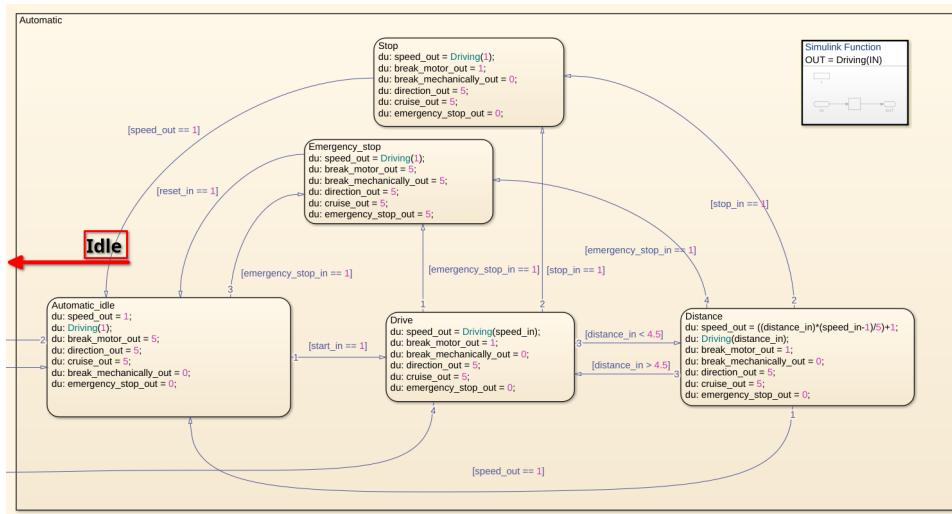


Abbildung 4.6: Automat – Übersicht der automatischen Steuerung

übertragen digitale Signale und fungieren als Datenleitungen.

Wenn zwei Leitungen für die Datenübertragung vorhanden sind, kann die Kommunikation bei einer zweiadrigten Verbindung entweder synchron oder asynchron erfolgen. Ist die Kommunikation synchron, dient eine der Leitungen als Takteleitung (Clock). Bei einer asynchronen Kommunikation, ist eine Leitung der Sender (TX) und die andere der Empfänger (RX). Dies ermöglicht eine Vollduplex-Übertragung.

Tabelle 4.14: PARKSIDE – Pin Mapping – Distanzsensor

| Pin | Farbe   | Funktion         |
|-----|---------|------------------|
| 1   | Rot     | 3V3              |
| 2   | Weiß    | RX (receiver)    |
| 3   | Gelb    | TX (transmitter) |
| 4   | Schwarz | GND              |

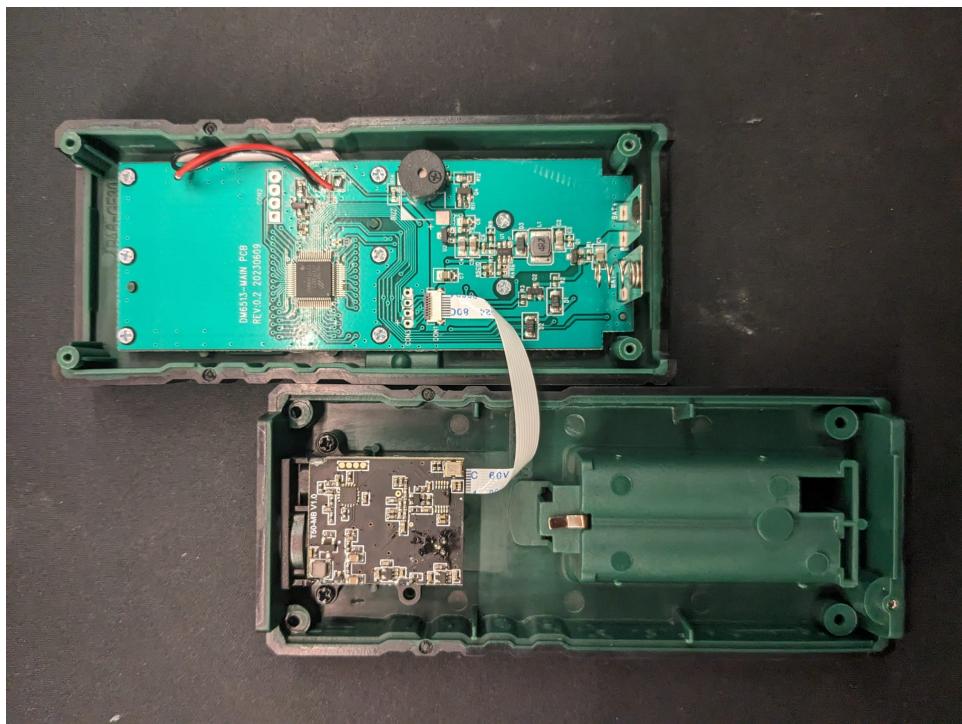


Abbildung 4.7: PARKSIDE – Distanzsensor – Innenaufbau

#### 4.3.2 Kodierung der Steuersignale

Der Sensor sendet verschiedene Steuersignale, die über die Busleitung übertragen werden. Jede Nachricht beginnt mit dem Bitmuster »0x24« (Nachricht-Start) und endet mit dem Bitmuster »0x26« (Nachricht-Ende), zwischen den beiden Bitmuster werden die Steuersignale und Messwerte kodiert.

Um eine Messung durchzuführen, müssen zwei Signale an den Sensor übertragen werden »Laser on« und »Laser start«, danach sendet der Sensor die gemessene Distanz »distanceValue«.

```
24 30 30 30 33 32 36 30 30 32 39 26 (Laser on)
24 30 30 30 33 32 36 30 31 33 30 26 (Laser start)
24 30 30 30 32 32 31 32 33 26 (Messen)
```

#### 4.3.3 Steuerung für die Sensordaten

Um die Werte von dem Distanzsensor zu erhalten wird eine ESP verwendet.

## Kapitel 5

# Konklusion

To-do

- Endeschnittstellen für die weiterführung des Projekts.
- weiterführung mit Testsenarien.

# Literatur

- [1] Golden Motor. *Motor Wiring*. PDF. [Online; Zugriff 05-September-2024]. 2024.
- [2] mathworks. *Introduction to Brushless DC Motor Control*. Abschnitt: Types of DC Motors; abgerufen am 11-08-2024. 2024. URL: <https://de.mathworks.com/campaigns/offers/next/introduction-to-brushless-dc-motor-control.html>.
- [3] Golden Motor. *Vector Controller (FOC) for BLDC Motors*. PDF. [Online; Zugriff 06-September-2024]. 2024.
- [4] Speedgoat. *IO397 – FPGA I/O Module – Hardware Reference Manual*. PDF. 2024.
- [5] Speedgoat. *IO691 – FPGA I/O Module – Hardware Reference Manual*. PDF. 2024.
- [6] Speed goat. *CAN Message Types*. [Online; Zugriff 13-Oktober-2024]. 2024. URL: [https://www.speedgoat.com/help/slrt/page/io\\_main/refentry\\_can\\_ofa\\_usage\\_notes](https://www.speedgoat.com/help/slrt/page/io_main/refentry_can_ofa_usage_notes).
- [7] Pat Richards. *Understanding Microchip's CAN Module Bit Timing*. PDF. 2001.
- [8] DIN EN IEC 81346-2 *Industrielle Systeme, Anlagen und Ausrüstungen und Industrieprodukte – Strukturierungsprinzipien und Referenzkennzeichnung – Teil 2*. Norm. 2020.

## Anhang A

# Anhang

### A.1 Schaltplan

To-do

- BMK beschriftung anpassen
- Bauteile einpflegen