



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, 4. November 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, 4. November 2024

Oliver Schmidt

Inhaltsverzeichnis

Rechtliche Erklärung	I
Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
1 Einleitung	1
1.1 Motivation der Hyperloop Technologie	1
1.2 Aufgabenstellung	3
1.3 Aufbau der Projektdokumentation	3
2 Konzept	5
3 Stand der Technik	7
3.1 Antrieb – Golden Motor	7
3.1.1 BLDC Motor	7
3.1.2 Vector Controller	9
3.2 Echtzeitfähige Steuerung – Speedgoat	11
3.2.1 I/O-Modul – IO397-50k	13
3.2.1.1 Pin Mapping – IO397-50k	14
3.2.1.2 IO397-50k – Konfigurieren	15
3.2.2 I/O-Modul – IO691	19
3.2.2.1 Pin Mapping – IO691	20
3.2.2.2 IO691 Module konfigurieren	20
4 Implementierung	24
4.1 Schaltplan	24
4.1.1 Betriebsmittelbezeichnung	24
4.1.1.1 Funktionskennzeichen	24
4.1.1.2 Ortskennzeichen	24
4.1.1.3 Betriebsmittelkennzeichen	25
4.1.2 Verdrahtung Konventionen	26
4.2 Simulation mit Simulink	27
4.2.1 Funktionsweise der Steuerung	27
4.2.1.1 Controlpanel	27

4.2.2	Automatensteuerung	29
4.2.2.1	Automat für die manuelle Steuerung	31
4.2.2.2	Automat für die automatische Steuerung	33
4.3	Distanzmessung	36
4.3.1	Verbindung	36
4.3.2	Dekodierung der Steuersignale	38
5	Konklusion	42
A	Anhang	44
A.1	Schaltplan	44

Abbildungsverzeichnis

1.1	Hyperloop der Hochschule Emden-Leer	1
2.1	Konzept – Aufbauplan des Fahrzeugs	5
3.1	Golden Motor – 10 KW BLDC Motor Liquid Cooled	8
3.2	Golden Motor – VECTOR 500 Motor Controller	10
3.3	Speedgoat – Baseline Real-Time Target Machine	12
3.4	Speedgoat – Ansicht der Modulanschlüsse	13
3.5	Speedgoat – I/O Module 397 [4]	14
3.6	Simulink – Driver Block – IO397-50k	16
3.7	Speedgoat – TTL I/O Interface – IO397-50k [4, S. 12]	17
3.8	Simulink – Pull-Widerstände – IO397-50k [4, S. 12]	18
3.9	Speedgoat – I/O Module 691 [5]	19
3.10	Simulink – Driver Block – I/O Module 691 [5]	20
3.11	Simulink – Baudrate – I/O Module 691 [6]	22
4.1	Automat – Controlpanel	28
4.2	Automat – Start Zustand »Idle«	29
4.3	Automat – Funktion Driving	30
4.4	Automat – Übersicht der manuellen Steuerung	32
4.5	Automat – »Forwards« -Stateflow der manuellen Steuerung .	33
4.6	Automat – Übersicht der automatischen Steuerung	36
4.7	Messung – zur Überprüfung der Kommunikationsart: Synchron oder Asynchron	37
4.8	PARKSIDE – Distanzsensor – Innenaufbau	38
4.9	Distanzmessung – Signalverlauf der Steuersignale	39

Tabellenverzeichnis

3.1	Pin Mapping – BLDC-Motor	9
3.2	Pin Mapping – Vector Controller	11
3.3	Speedgoat – Anschlüsse der Module	13
3.4	IO397-50k Pin Mapping – Terminal Board A: analog I/O [4, S. 15]	15
3.5	IO397-50k Pin Mapping – Terminal Board B: analog I/O [4, S. 15]	16
3.6	Simulink – Pull-Widerstand – Reaktionszeit	17
3.7	IO691 Pin Mapping	20
3.8	Simulink – Baudraten Parameter	21
4.1	Schaltplan – Funktionskennzeichen (=)	25
4.2	Schaltplan – Ortskennzeichen (+)	25
4.3	Schaltplan – Betriebsmittelkennzeichen (-) [8]	26
4.4	Verdrahtung Konvention – Aderfarben	26
4.5	Automat – Wahlschalter für den manuellen Betrieb	27
4.6	Ausgänge – Idle	30
4.7	Ausgänge – Emergency stop	31
4.8	Ausgänge – der Zustände »Forwards« und »Backwards« – »slow«	32
4.9	Ausgänge – der Zustände »Forwards« und »Backwards« – »fast«	33
4.10	Ausgänge – Automatic	34
4.11	Ausgänge – Drive	34
4.12	Ausgänge – Distance	35
4.13	Ausgänge – Stop	35
4.14	PARKSIDE – Pin Mapping – Distanzsensor	37
4.15	Distanzmessung – Signale die an den Sensor gesendet werden (TX)	40
4.16	Distanzmessung – Signale die von dem Sensor gesendet werden 1 (RX)	40
4.17	Distanzmessung – Signale die von dem Sensor gesendet werden 2 (RX)	41

Kapitel 1

Einleitung

1.1 Motivation der Hyperloop Technologie

Der Hyperloop ist ein innovatives Transportkonzept, das 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 und Güter schnell und emissionsarm zu befördern. Zum einen der hohe Luftwiderstand, der bei hohen Geschwindigkeiten den Energieverbrauch stark erhöht.

Der Hyperloop löst diese Probleme, indem er Güter und Personen in einem Fahrzeug, das sich in einer fast Vakuumröhre bewegt, wie in Abbildung 1.1 dargestellt ist.

Institute of Hyperloop Technology

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

Die Forschung des »Institute of Hyperloop Technology«, fokussiert sich auf die Güterbeförderung.

1.2 Aufgabenstellung

Das Ziel dieses Projektes ist die Entwicklung eines Hyperloop-Fahrzeuges, das mit einer Batterie und einem Motor betrieben wird. Für die Steuerung des Fahrzeugs wurde ein echtzeitfähiges Steuerungssystem der Firma Speedgoat vorgegeben, welches in Abschnitt 3.2 vertiefend erläutert 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 beinhaltet die Erfassung von Position und Beschleunigung des Fahrzeugs. 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

Die Projektdokumentation beginnt mit einer Einführung in die Hyperloop-Technologie und der Aufgabenstellung dieses Projekts, die die Motivation und Zielsetzungen erläutern. Es wird dargelegt, warum der Hyperloop als innovative, umweltfreundliche Transportlösung eine wichtige Alternative zu bestehenden Systemen darstellt und welche spezifischen Herausforderungen und Ziele mit der Entwicklung eines funktionsfähigen Prototyps verbunden sind.

Im weiteren Verlauf wird das grundlegende Fahrzeugkonzept vorgestellt. Dabei wird beschrieben, wie die verschiedenen Baugruppen miteinander verbunden sind und welche Rolle das zentrale Steuerungssystem von Speedgoat übernimmt. Auch die verwendeten Sensoren und deren Einbindung in das Gesamtsystem werden erläutert.

Der Dokumentationsteil zur Implementierung zeigt die technische Realisierung der Steuerung und der elektrischen Verdrahtung des Fahrzeugs. Die Erstellung des Schaltplans, die Konfiguration der Steuerlogik sowie die detaillierte Beschreibung der Automatik- und manuellen Steuerungsmodi geben Einblicke in die Funktionsweise und den Aufbau des Systems. Ergänzend

wird die Integration der Distanzmessung beschrieben, die eine präzise Positionsermittlung innerhalb der Teststrecke ermöglicht.

Abschließend fasst die Konklusion die Ergebnisse der Projektarbeit zusammen und gibt Hinweise auf mögliche zukünftige Entwicklungen. Hierbei wird auch die Offenheit der Systemarchitektur betont, die eine flexible Weiterentwicklung ermöglicht..

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

Hyperloop - CargoPOD

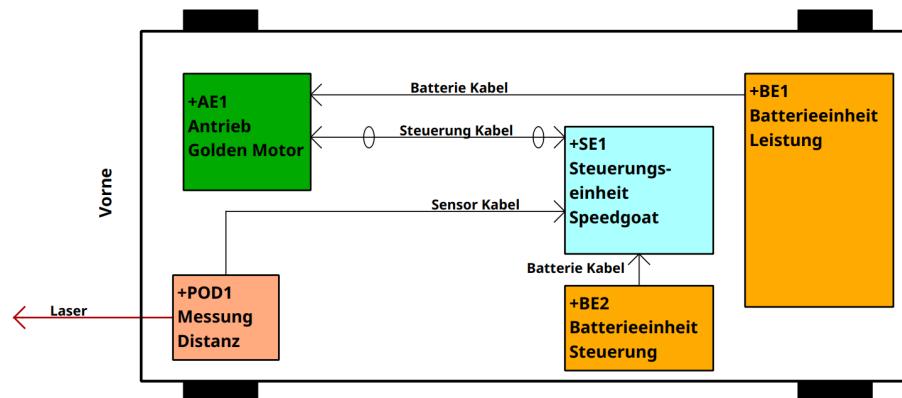


Abbildung 2.1: Konzept – Aufbauplan des Fahrzeugs

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. Die Distanzmessung ist für die Positions- und Geschwindigkeitsermittlung notwendig. 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

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

3.1 Antrieb – Golden Motor

Golden Motor bietet eine Gesamtlösung bestehend aus einem BLDC-Motor und einem Vector-Controller an. Der Controller besitzt eine Schnittstelle für alle notwendigen Steuer- und Kontrollsäle, 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].

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.

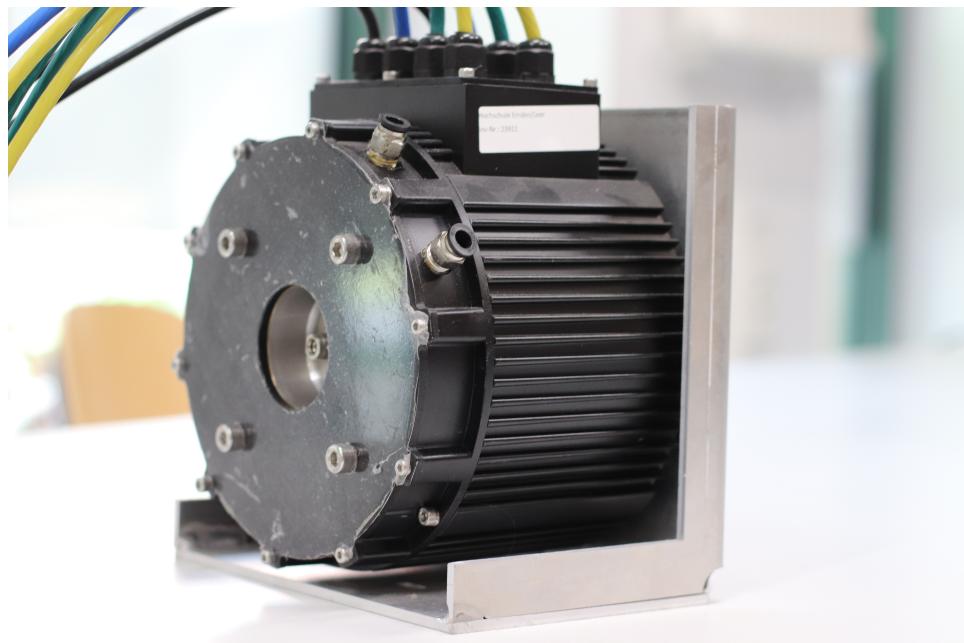


Abbildung 3.1: Golden Motor – 10 KW BLDC Motor Liquid Cooled

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].

Tabelle 3.1: Pin Mapping – BLDC-Motor

Anschluss	Funktionalität	Farbe
Anschluss Adern Motor		
U1	Spule 1	Gelb
V1	Spule 2	Grün
W1	Spule 3	Blau
U2	Spule 1	Gelb
V2	Spule 2	Grün
W2	Spule 3	Blau
Motor Hall Kabel 1		
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
Motor Hall Kabel 2		
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

3.1.2 Vector Controller

Der Motor wird mithilfe eines Vector-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 Vector-Controllers wird kurz unter 3.1.2 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. 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

Funktionsweise

Der Vector-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 RotorflussVector im rechten Winkel (90 Grad) zum StatorflussVector ausgerichtet sein [3].

Tabelle 3.2: Pin Mapping – Vector Controller

Anschluss	Funktionalität	Farbe	U in V
Batterie Eingang			
B+	Versorgungsspannung	RT	48 V
B-	Masse	SW	0 V
Motor Ausgang			
U	Spule 1	GE	
V	Spule 2	GN	
W	Spule 3	BL	
Digitale und Analoge I/O			
5	Masse - Bremse	SW	0 V
6	Masse	SW	0 V
6	Masse - Tempomat	SW	0 V
7	Tempomat	GR	0 V oder 5 V
10	elektronisches Schloss	RS	0 V oder 5 V
16	Bremse	BL&WS	1 V oder 5 V
17	Fahrtrichtung	BR	0 V oder 5 V
26	Masse	SW&WS	0 V
27	Geschwindigkeit	GN&WS	1 V - 4 V
28	+5 V	RT&WS	5 V
Hallsensor Kabel			
(3)	Motor Temperatur	WS	
(11)	Hall C – Sensor	BL	
(12)	Hall B – Sensor	GN	
(13)	Hall A – Sensor	GE	
(14)	+5 V	RT	5 V
(21)	Masse	SW	0 V

3.2 Echtzeitfähige Steuerung – Speedgoat

Der Baseline Real-Time Target Machine, wie in Abbildung 3.3 dargestellt, ist ein echtzeitfähiges System, welches von der Firma Speedgoat hergestellt und vertrieben wird. 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 in 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 Systeme 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.

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 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.



Abbildung 3.4: Speedgoat – Ansicht der Modulanschlüsse

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- und Ausgänge. Die Anschlüsse werden auf Modul A, das die analogen Schnittstellen bereitstellt, und Modul B, welches die digitalen Schnittstellen umfasst, verteilt. Die Anschlussbelegung wird 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.

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 3.4: 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

3.2.1.2 IO397-50k – Konfigurieren

In dem Driver Block, wie in Abbildung 3.6 gezeigt, können an 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

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

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

**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.

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«,

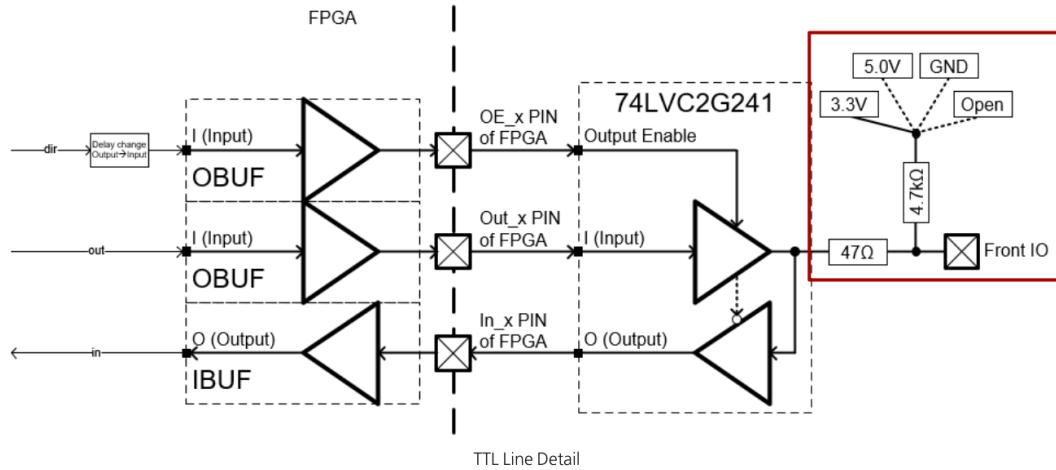


Abbildung 3.7: Speedgoat – TTL I/O Interface – IO397-50k [4, S. 12]

dass das Ausgangssignal eine längere Anstiegszeit hat. In Abbildung 3.8 sind die Reaktionszeiten dargestellt.

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	-	-

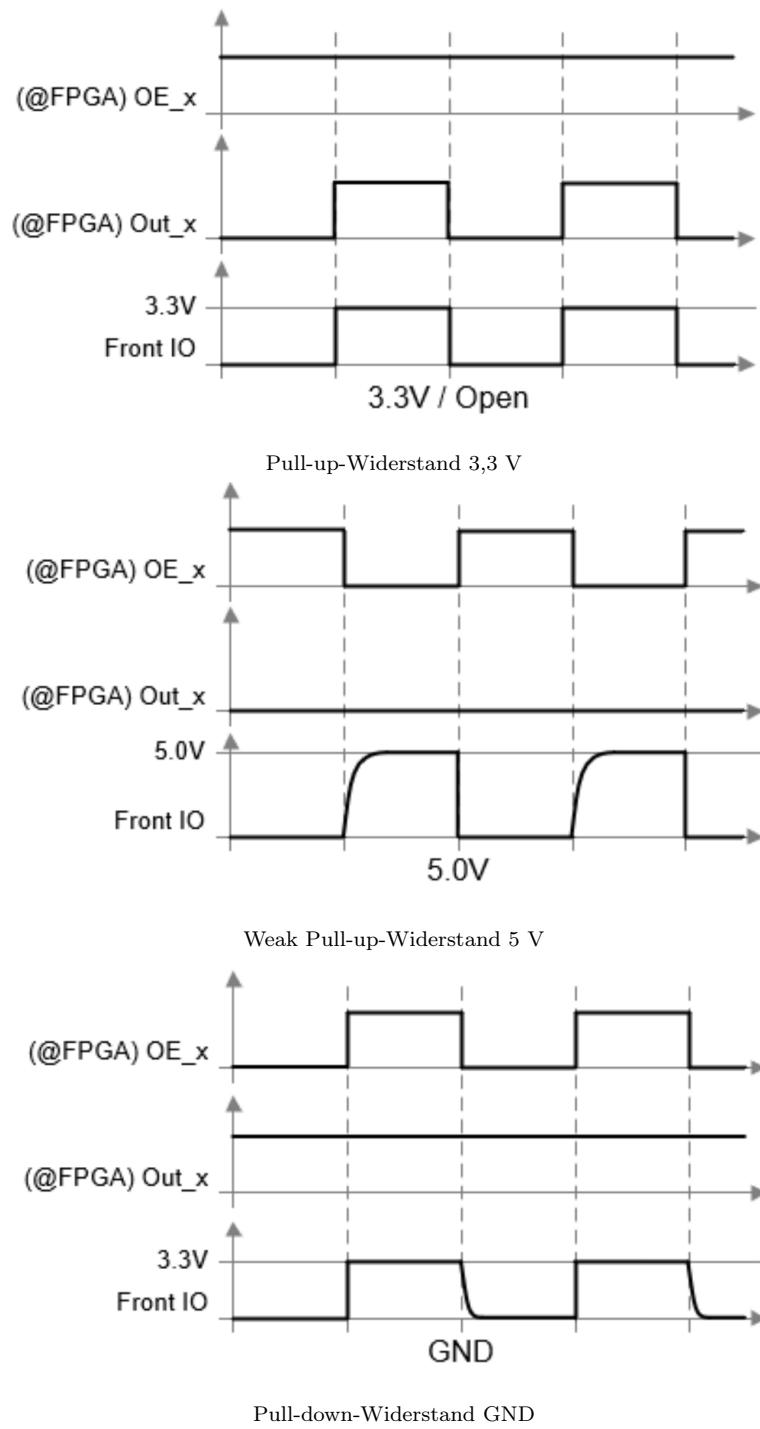


Abbildung 3.8: Simulink – Pull-Widerstände – IO397-50k [4, S. 12]

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« ausschließlich 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.

»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 oder gleich groß »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.

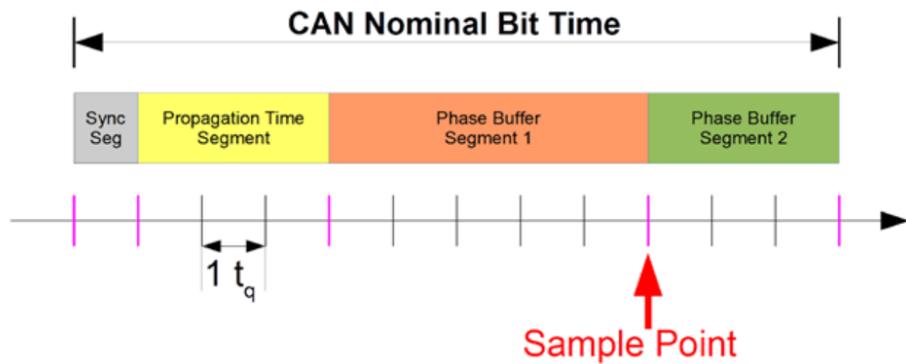


Abbildung 3.11: Simulink – Baudrate – I/O Module 691 [6]

Mit den weiteren Gleichungen können die Zeiten innerhalb der »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 120Ω 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

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).

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 4.1: Schaltplan – Funktionskennzeichen (=)

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

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

Die Betriebsmittelbezeichnung =ES2+SE1-10QA4 besagt, dass es sich um die Eingänge der Steuerung (=ES) handelt und es gibt zwei Funktionen in dem Schaltplan. 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 werden. 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.

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

Abkürzung	Klassenname	Allgemeine Bedeutung
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	Fingerinteraktionsobjekt	Schalter
TB	Stromkonvertierungsobjekt	Transformator
WD	Niederspannungsenergie Leitobjekt	Leitung/Kabel
XD	Niederspannungs-Verbindungsobjekt	Klemme, Stecker oder Buchse

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

Farbe	U in V	A in mm²	Funktion
Rot	24	1	Steuerspannung
Blau	x	1	
Schwarz	GND	1	Masse Steuerspannung
Schwarz	< 48	50	Batteriespannung (Last)
Viollett	5	1	Steuerung

4.2 Simulation mit Simulink

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 bewegen, wobei zwei Geschwindigkeitsstufen – schnell und langsam – zur Verfügung stehen. Diese Funktion erlaubt dem Nutzer, das Fahrzeug innerhalb der Röhre individuell zu steuern und es hinein- und herauszufahren.

Im Automatikmodus beschleunigt das Fahrzeug auf eine einstellbare Geschwindigkeit und bremst automatisch ab, sobald eine bestimmte Distanz zum Enddeckel der Röhre erreicht ist. Schließlich hält es am Ende der Strecke an.

4.2.1.1 Controlpanel

Über das Control Panel (siehe Abbildung 4.1) können alle Eingaben via Simulink gesteuert werden. Mit dem Wahlschalter kann, 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

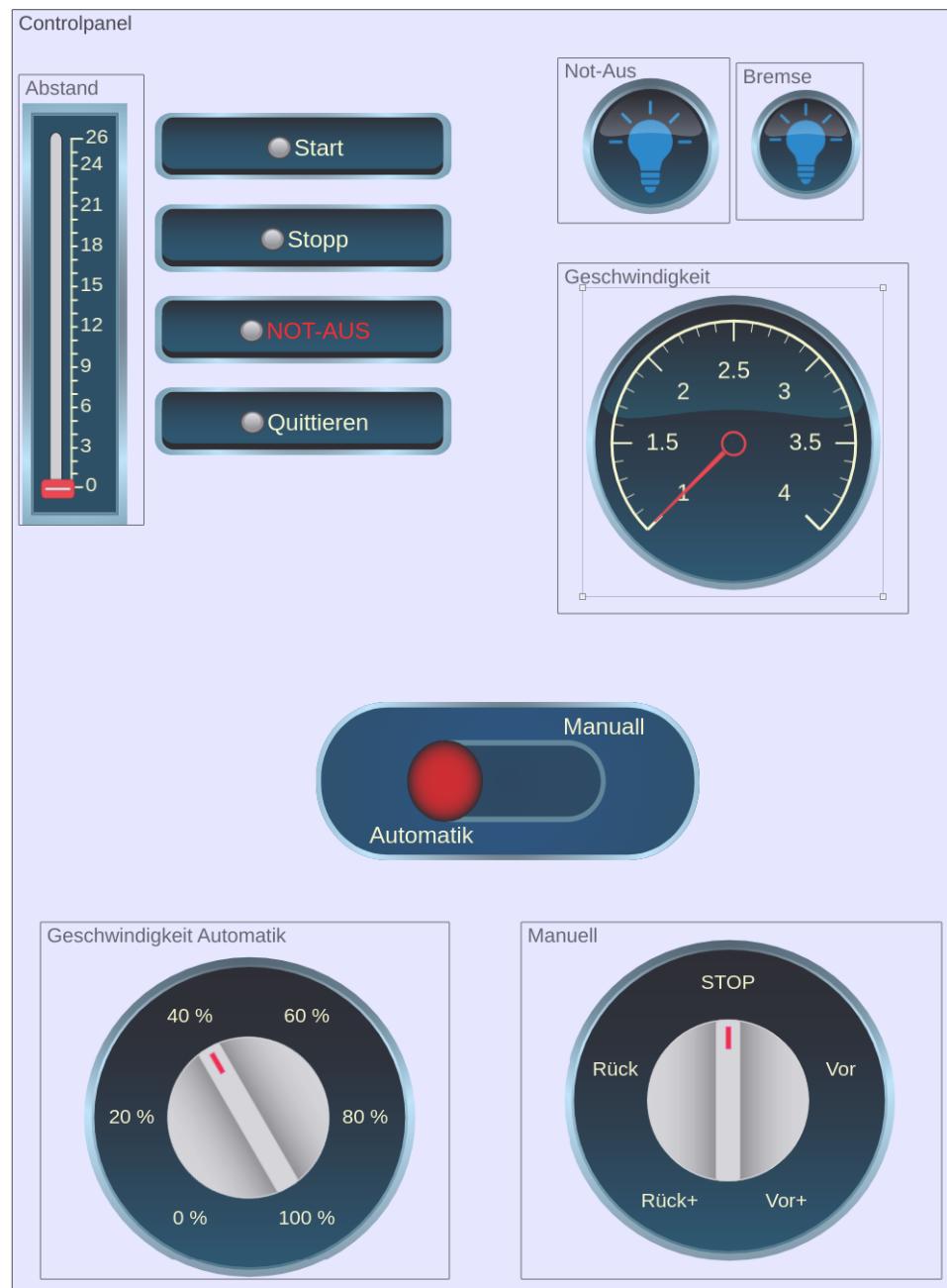


Abbildung 4.1: Automat – Controlpanel

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.

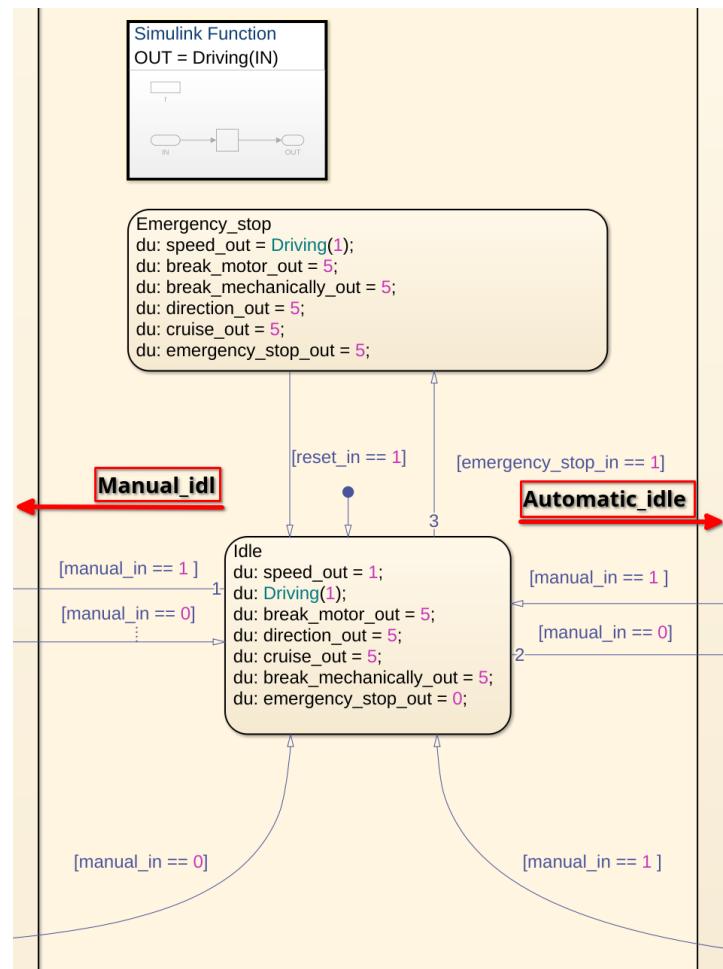
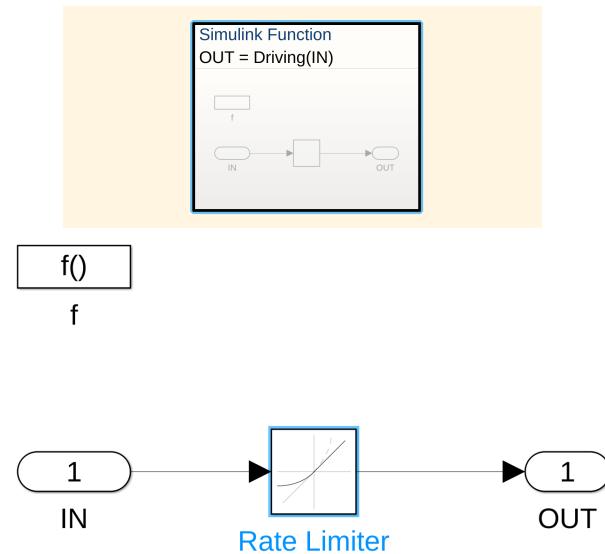


Abbildung 4.2: Automat – Start Zustand »Idle«

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 die Quittierung 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 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 2 Tabellen aufgeführt 4.8 und 4.9 aufgeführt, hierbei ist die erste Tabelle die Zustände »slow« und die zweite Tabelle für die Zustände »fast«.

Mit dem Wahltaster (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 in den »Idle « Zustand.

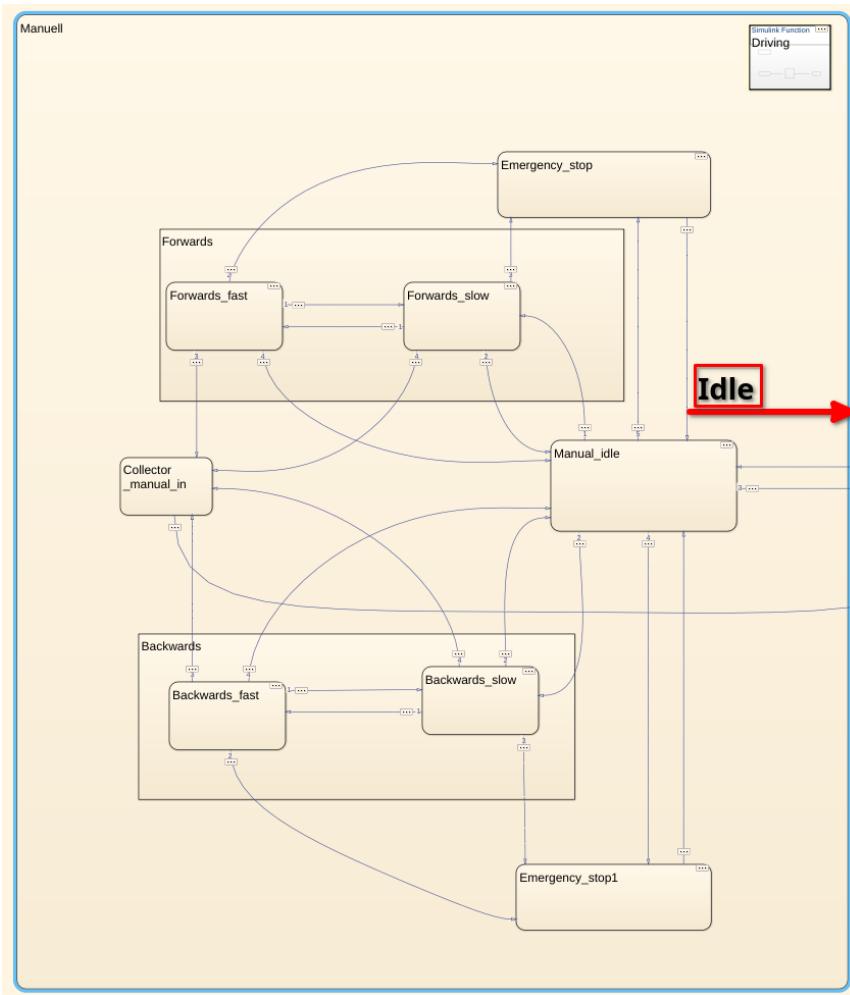


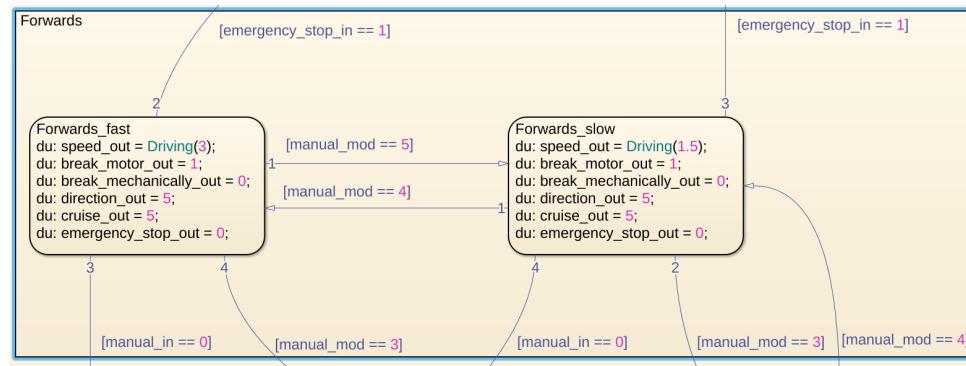
Abbildung 4.4: Automat – Übersicht der manuellen Steuerung

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

Ausgänge	I/O Module IO397-50k	Werte »Forwards«	Werte »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 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 vom Abstand »distance« zum Ende der Röhre verringert, hierbei wird in dem Zustand »Distance« der Augang der Geschwindigkeit mittels folgender Gleichung ermittelt:

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

Soll die Automatik beendet werden, kann dies über den Stopp-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 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 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 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 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

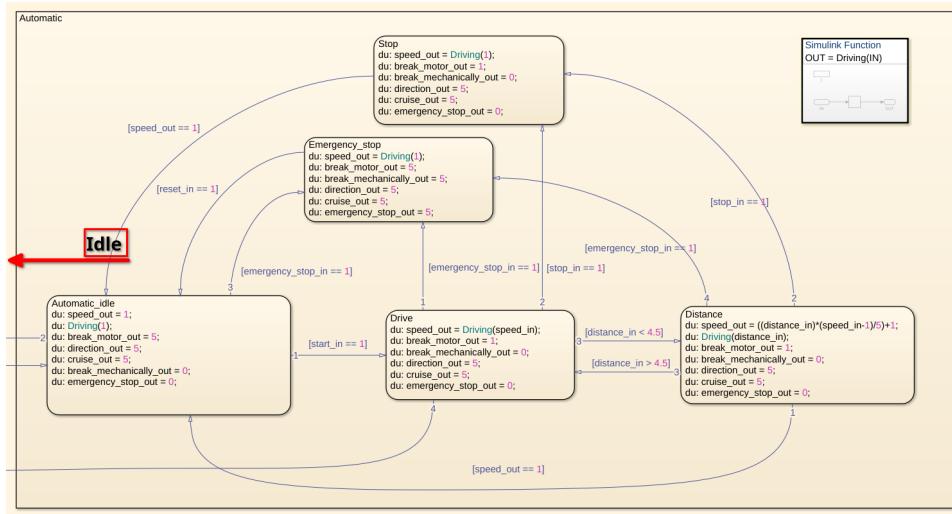


Abbildung 4.6: Automat – Übersicht der automatischen Steuerung

4.3 Distanzmessung

Ursprünglich wurde ein Sensor von Pepperl+Fuchs 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.

Dafür wurde ein günstiges Entfernungsmessgerät für den Heimbedarf angekauft und analysiert, ob es für die Anwendung geeignet ist.

4.3.1 Verbindung

Der Sensor ist über ein Flachbandkabel mit der Hauptplatine verbunden, wie in Abbildung 4.8 zu sehen ist. Auf der Hauptplatine befinden sich vier ungeutzte Lötstellen. Mithilfe eines Oszilloskops und eines Multimeters wurden die Signale an den Lötstellen analysiert, wobei festgestellt wurde, dass Leitung »1« eine Spannung von 3,3 V führt und Leitung »4« als GND dient. Die Leitungen »2« und »3« übertragen digitale Signale und fungieren als Datenleitungen.

Die Kommunikation kann bei einer zweiadrigigen Verbindung entweder synchron oder asynchron erfolgen. Bei synchroner Kommunikation dient eine der Leitungen als Taktleitung (Clock). Bei asynchroner Kommunikation ist eine Leitung der Sender (TX) und die andere der Empfänger (RX). Zur Überprüfung der Kommunikationsart wurde ein Oszilloskopogramm (siehe Abbildung 4.7) aufgenommen. Im Oszilloskopogramm wurde kein Taktsignal festgestellt, sodass die Datenübertragung asynchron und Vollduplex erfolgt. In Tabelle 4.14 ist die Pinbelegung aufgelistet.

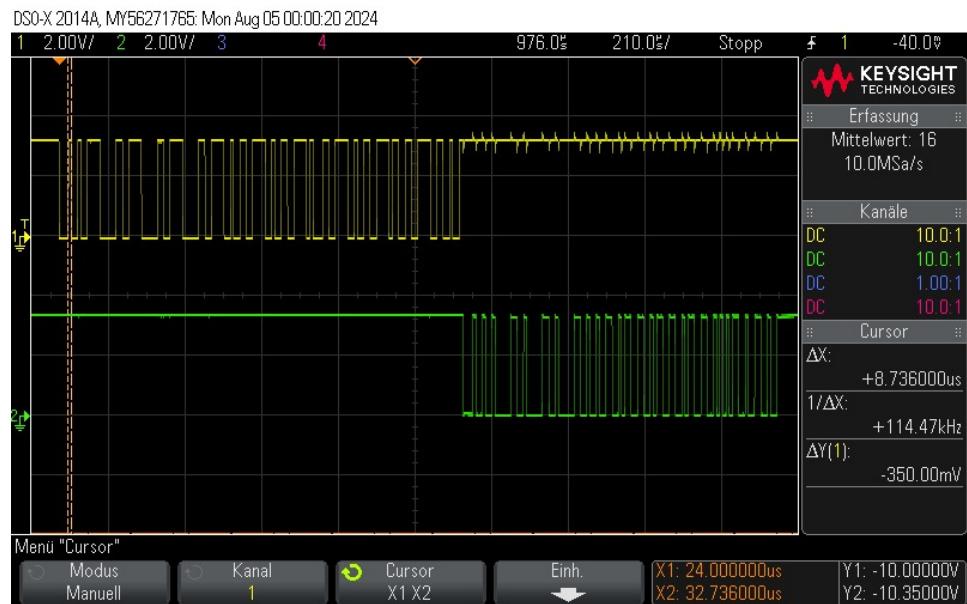


Abbildung 4.7: Messung – zur Überprüfung der Kommunikationsart: Synchron oder Asynchron

Tabelle 4.14: PARKSIDE – Pin Mapping – Distanzsensor

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

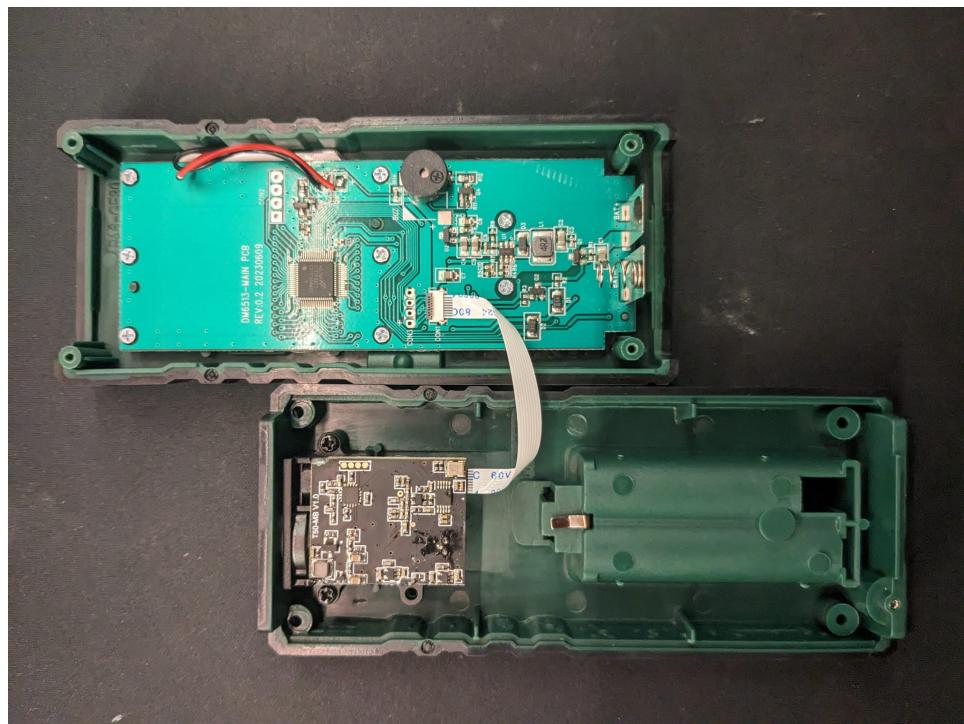


Abbildung 4.8: PARKSIDE – Distanzsensor – Innenaufbau

4.3.2 Dekodierung der Steuersignale

Um die Steuersignale zu analysieren, wurden die mit dem Oszilloskop aufgenommenen Messwerte in den Tabellen 4.15, 4.16 und 4.17 dokumentiert. Bei den Messungen fiel auf, dass jede Nachricht mit dem Bitmuster »0x24« (Nachricht-Start) beginnt und mit dem Bitmuster »0x26« (Nachricht-Ende) endet. Zwischen diesen beiden Bitmustern sind die Steuersignale und Messwerte kodiert, welche im Folgenden erklärt werden.

Die Signalübertragung in den Tabellen ist wie folgt aufgebaut: In Tabelle

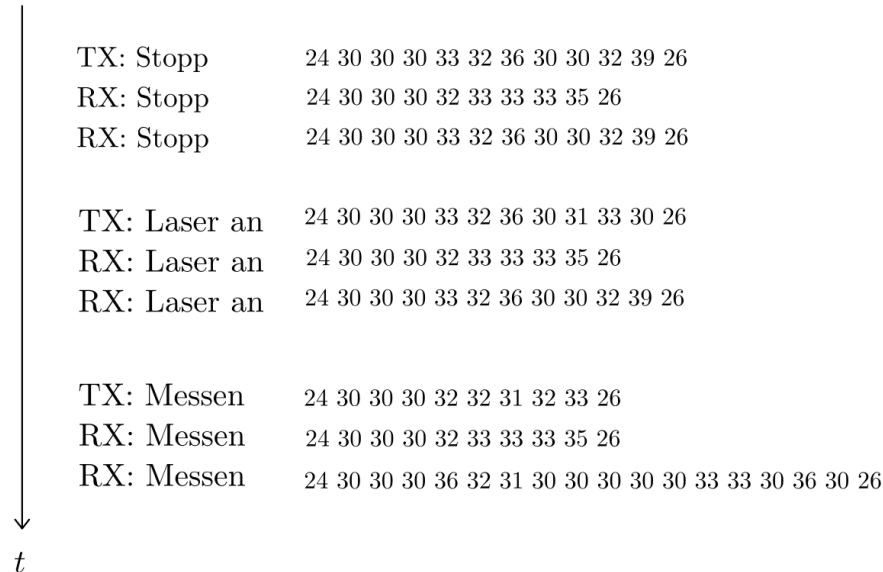


Abbildung 4.9: Distanzmessung – Signalverlauf der Steuersignale

4.15 zeigt die Spalte »Stopp« die erste Nachricht, die an den Sensor gesendet wird. Der Sensor antwortet stets mit zwei Nachrichten: Die erste Nachricht ist in Tabelle 4.16 dargestellt, die zweite in Tabelle 4.17, in der Abbildung 4.9 ist der Verlauf noch einmal aufgeführt.

Die gelb hinterlegten Felder in den Tabellen weisen auf ein identisches Bitmuster hin. Nach jedem Start-Byte werden drei identisch kodierte Bytes gesendet, die als Delay fungieren. Der Sensor antwortet nach jedem »Transmitter« Signal mit derselben Kodierung, was als Bestätigung dient, dass er sendebereit ist. Erst wenn das Bitmuster »Messen« an den Sensor gesendet wird, überträgt der Sensor die Messdaten, die in den grün hinterlegten Feldern in Tabelle 4.17 als ASCII-Zeichen dargestellt sind.

Tabelle 4.15: Distanzmessung – Signale die an den Sensor gesendet werden (TX)

Bytes	TX Stopp		TX Laser an		TX Messen	
	Nachricht	Bdtg.	Nachricht	Bdtg.	Nachricht	Bdtg.
0	24	Start	24	Start	24	Start
1	30		30		30	
2	30		30		30	
3	30		30		30	
4	33		33		32	
5	32		32		32	
6	36		36		31	
7	30		30		32	
8	30		31		33	
9	32		33		26	Stop
10	39		30			
11	26	Stop	26	Stop		

Tabelle 4.16: Distanzmessung – Signale die von dem Sensor gesendet werden 1 (RX)

Bytes	RX Stopp		RX Laser an		RX Messen	
	Nachricht	Bdtg.	Nachricht	Bdtg.	Nachricht	Bdtg.
0	24	Start	24	Start	24	Start
1	30		30		30	
2	30		30		30	
3	30		30		30	
4	32		32		32	
5	33		33		33	
6	33		33		33	
7	33		33		33	
8	35		35		35	
9	26	Stop	26	Stop	26	Stop
10						
11						

Tabelle 4.17: Distanzmessung – Signale die von dem Sensor gesendet werden 2 (RX)

Bytes	RX Stopp		RX Laser an		RX Messen	
	Nachricht	Bdtg.	Nachricht	Bdtg.	Nachricht	Bdtg.
0	24	Start	24	Start	24	Start
1	30		30		30	
2	30		30		30	
3	30		30		30	
4	33		33		36	
5	32		32		32	
6	36		36		31	
7	30		30		30	10^{+4}
8	30		30		30	10^{+3}
9	32		32		30	10^{+2}
10	39		39		30	10^{+1}
11	26	Stop	26	Stop	30	10^{+0}
12					33	10^{-1}
13					33	10^{-2}
14					30	10^{-3}
15					36	10^{-4}
16					30	10^{-5}
17					26	Stop

Kapitel 5

Konklusion

In dieser Projektarbeit wurde erfolgreich ein funktionsfähiges Fahrzeug für den Hyperloop-Prototyp entwickelt und dessen Aufbau sowie die Steuerung detailliert beschrieben. Die Implementierung umfasste mehrere wesentliche Schritte: Von der Erstellung eines Schaltplans für die elektrische Verdrahtung über die Integration der Sensorik bis hin zur Programmierung und Simulation der Steuerungslogik.

Die Implementierung der Steuerung erfolgte über ein Speedgoat-System, das Echtzeitsteuerung ermöglicht. Die Simulation der Steuerungslogik in Simulink stellte sicher, dass das Fahrzeug sowohl im manuellen als auch im automatischen Modus betrieben werden kann. Im manuellen Modus kann der Benutzer das Fahrzeug vorwärts und rückwärts fahren lassen, während der automatische Modus das Fahrzeug beschleunigt und bei Annäherung an das Streckenende abbremst. Zudem wurde die Distanzmessung durch einen alternativen Sensor kostengünstig und zuverlässig realisiert, was die Positionierung des Fahrzeugs in der Röhre ermöglicht.

Die Wahl der Bauteile, wie des BLDC-Motors und der IO-Module, sowie die Verdrahtungskonventionen wurden sorgfältig dokumentiert und implementiert, um eine sichere und stabile Betriebsumgebung zu gewährleisten. Die entwickelten Endschnittstellen bieten eine Grundlage für zukünftige Erweiterungen und Testdurchläufe.

Zukünftige Arbeiten könnten darauf abzielen, das Fahrzeug unter realistischen Bedingungen zu testen, um das Zusammenspiel von Steuerung und Sensorik weiter zu optimieren. Die durchgeführten Implementierungen und die offene Architektur des Systems bieten somit eine flexible Basis für die Weiterentwicklung des Hyperloop-Konzepts und die Erforschung einer innovativen, energieeffizienten Transportlösung.

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.
- [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

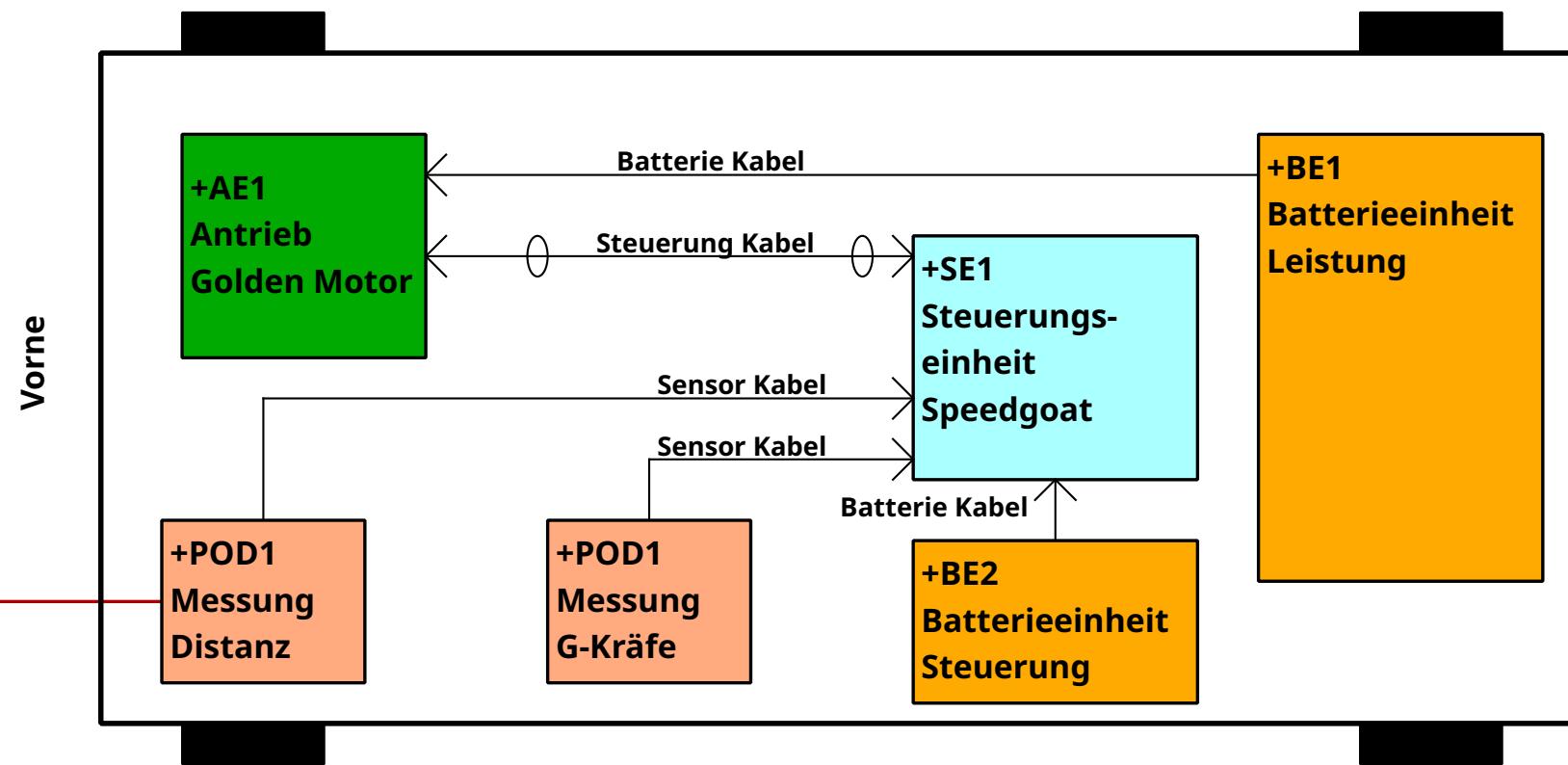
Anhang

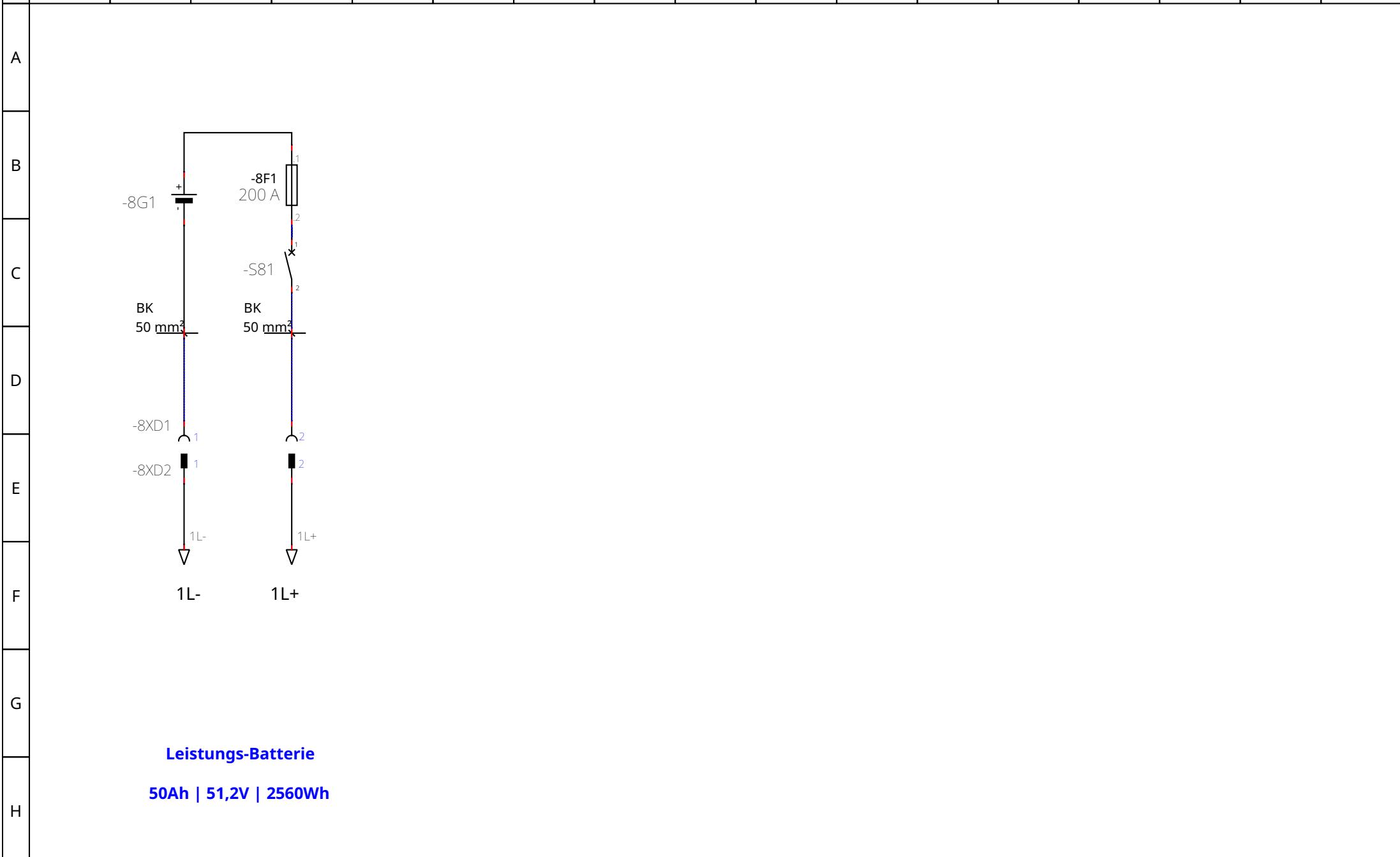
A.1 Schaltplan

To-do

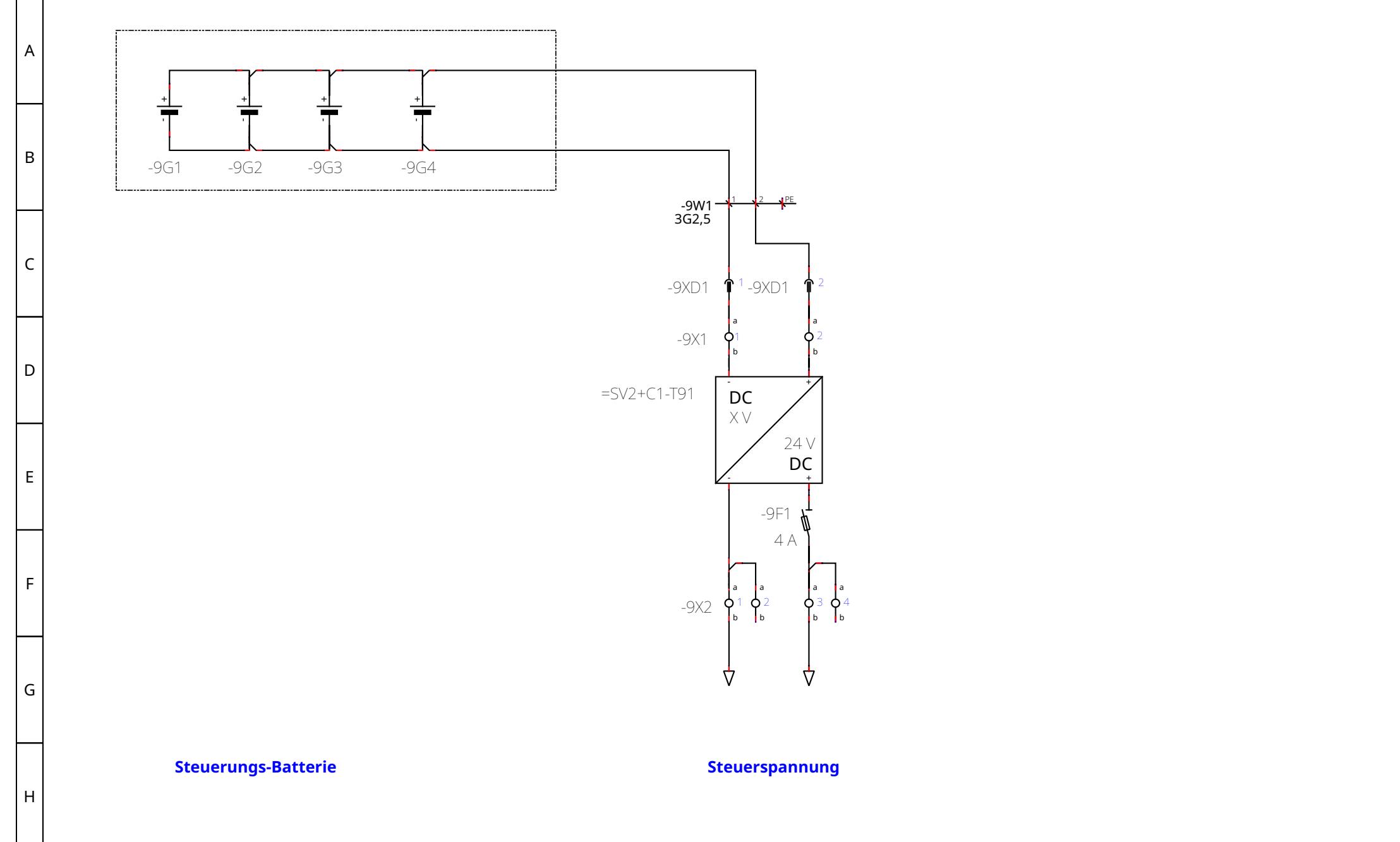
- BMK beschriftung anpassen
- Bauteile einpflegen

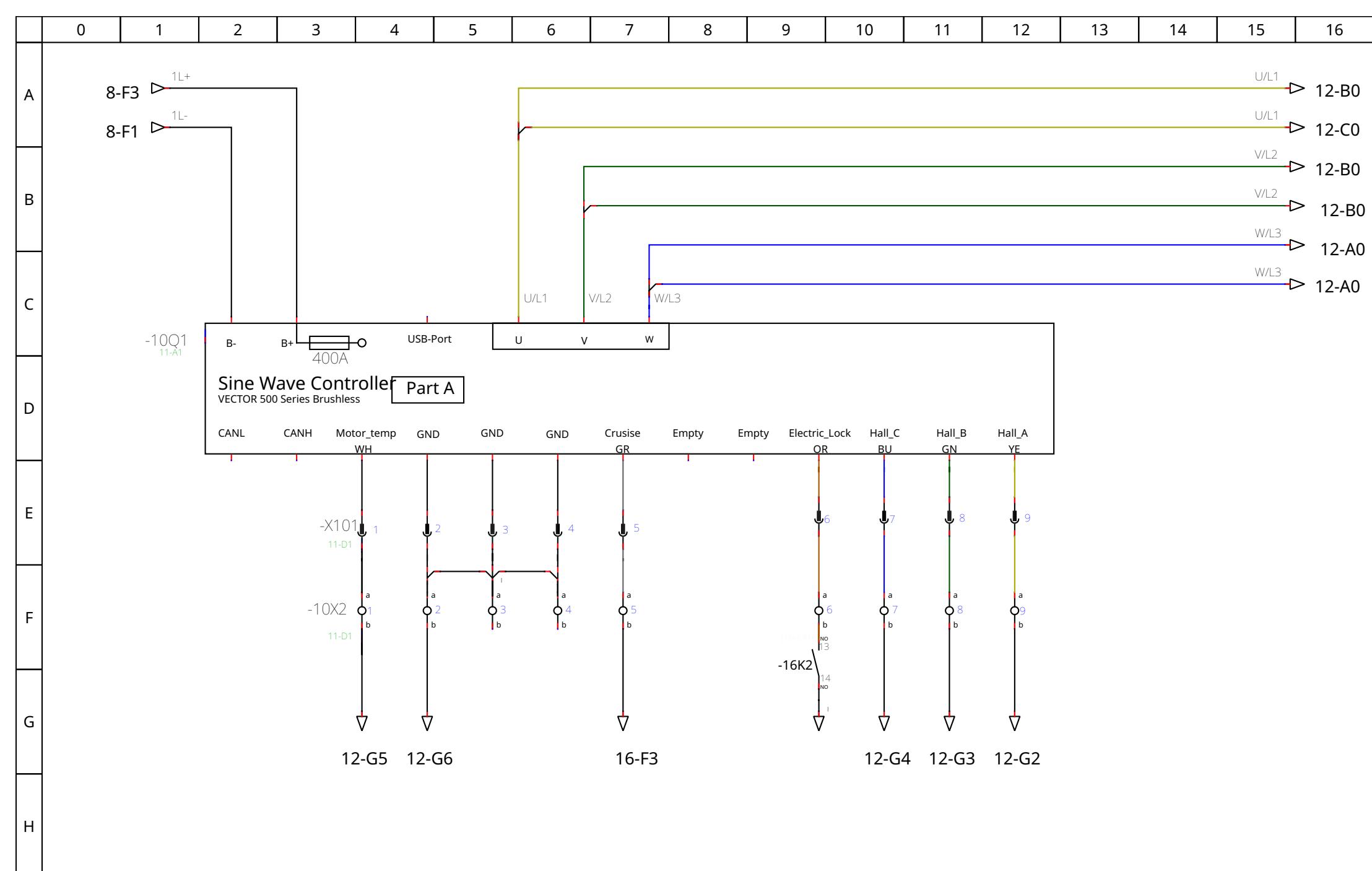
Hyperloop – POD

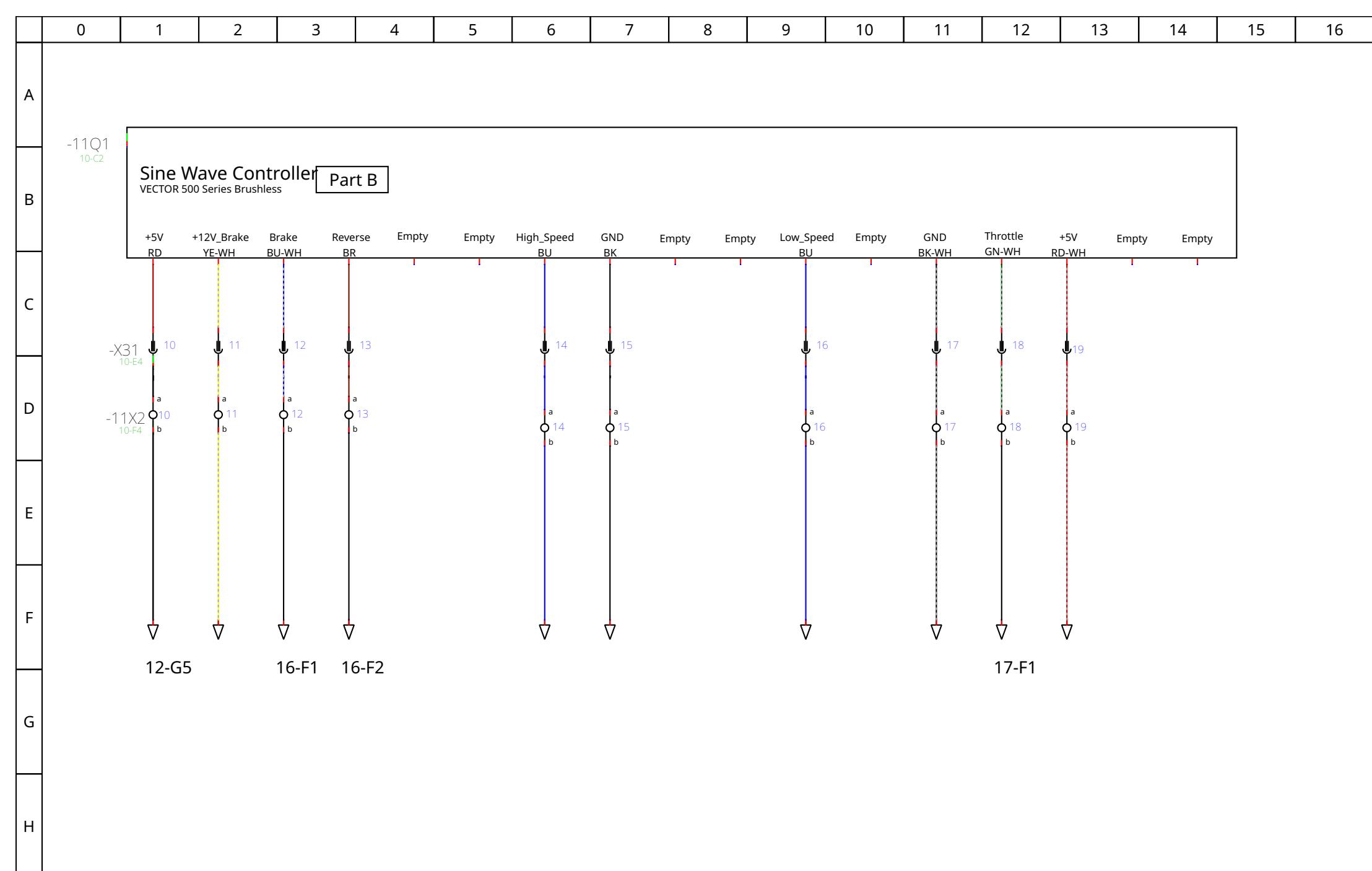


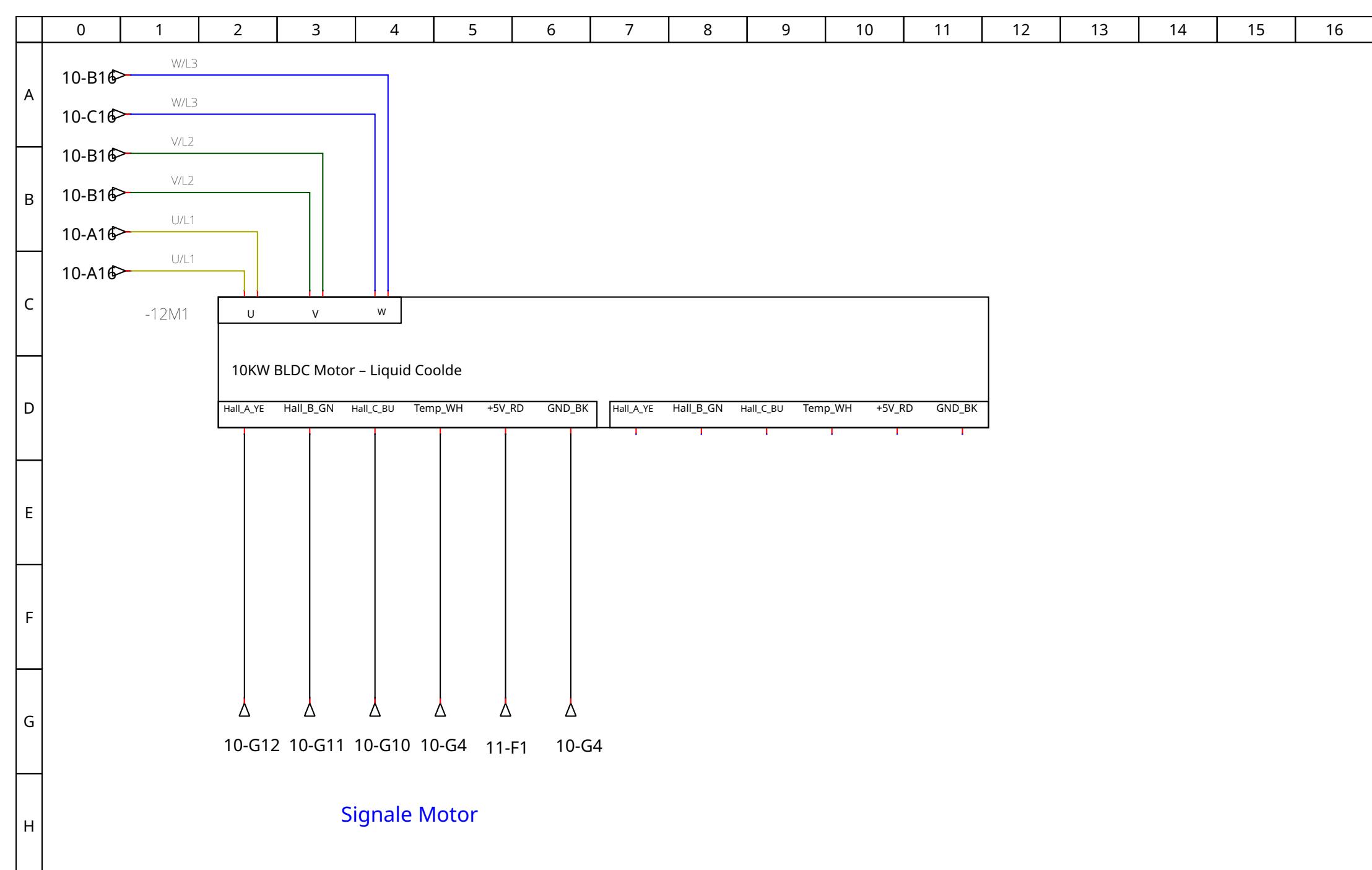


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
--	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

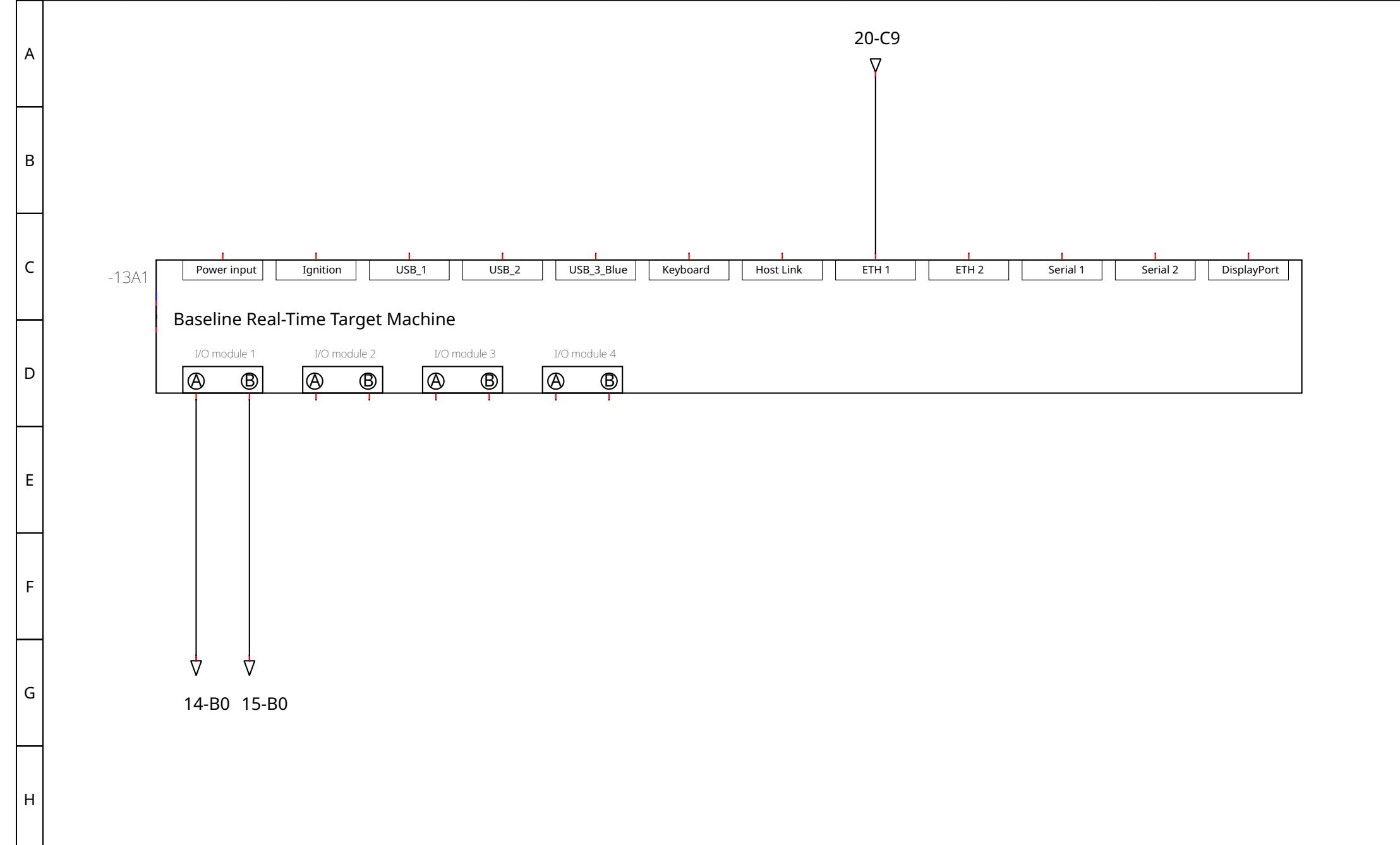




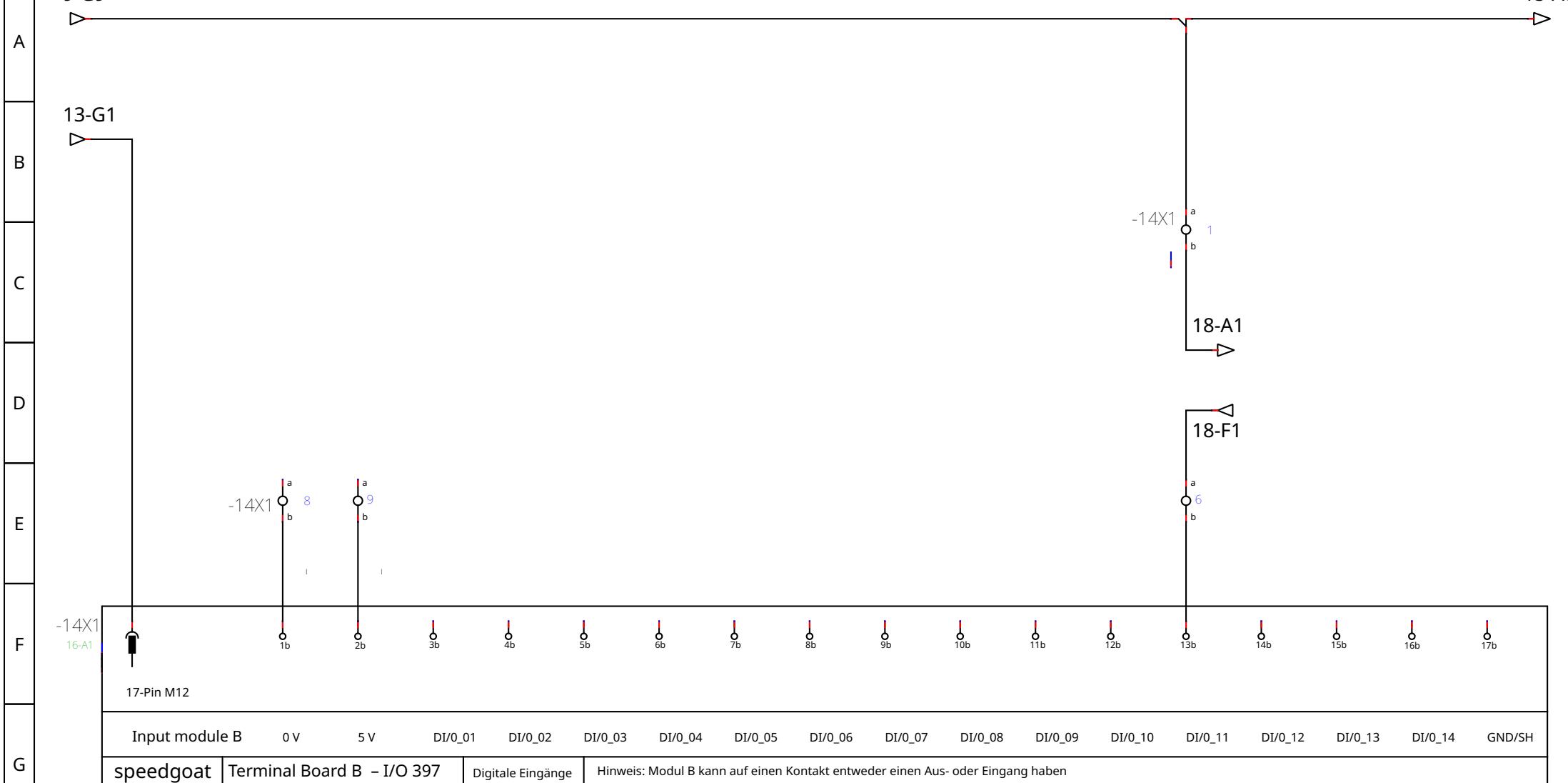




	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
--	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

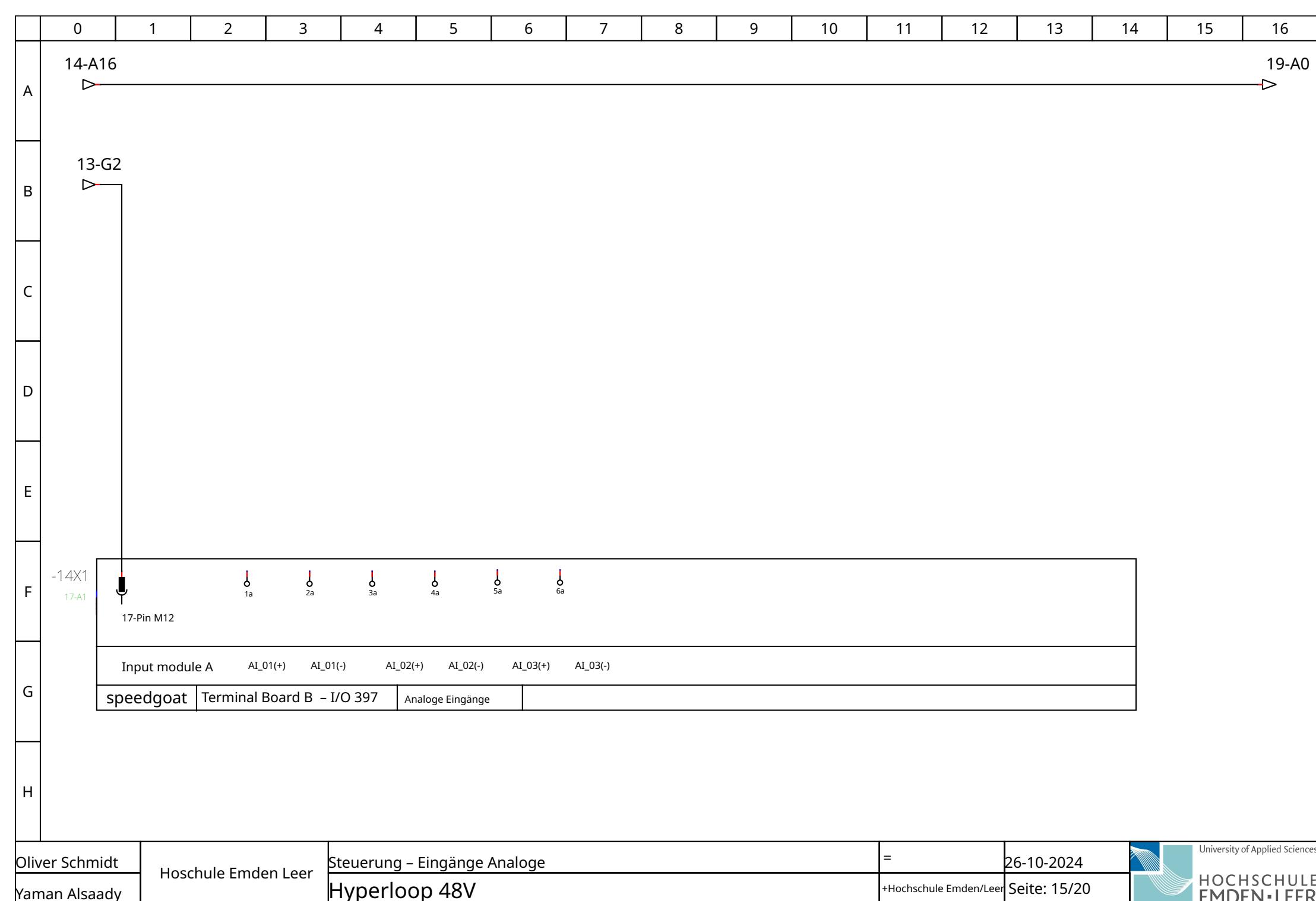


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
--	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

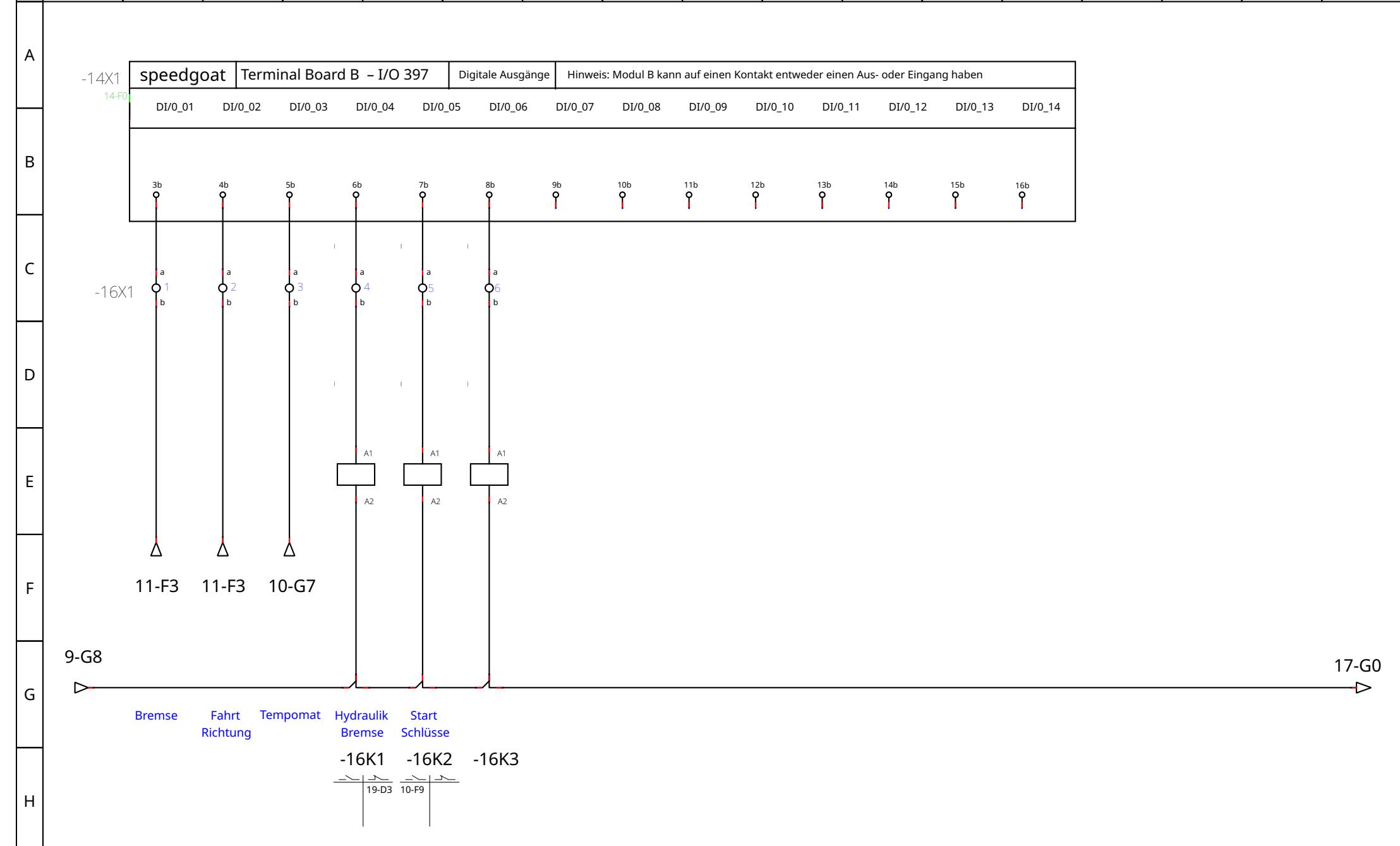


Not
Aus

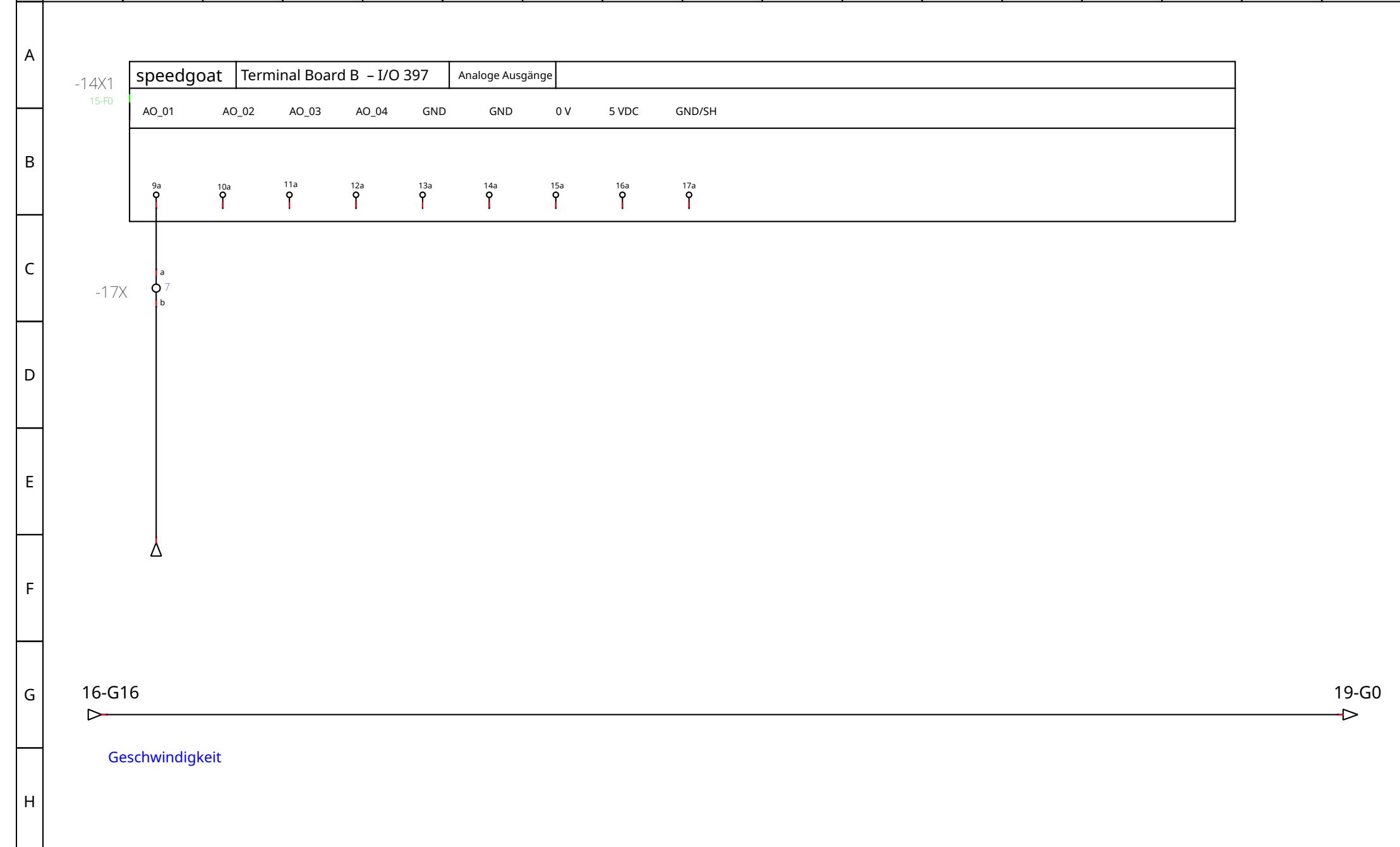
Oliver Schmidt	Hochschule Emden Leer	Steuerung - Eingänge Digitale Hyperloop 48V	=1D1 (POD)	26-10-2024	 University of Applied Sciences HOCHSCHULE EMDEN·LEER
Yaman Alsaady			+1C1 (Controllunit)	Seite: 14/20	



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
--	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
--	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
--	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

