

HOCHSCHULE LUZERN

PREN 1 TEAM 10

JULIAN BISCHOF
GABRIEL BUCKLAND
SARANGAN GOPALACHANDRAN
YANNICK MERZ
SANDRO MÖSCH
MANUEL ZIHLMANN

Abgabe Meilenstein 4

9. Januar 2025

Versionsverlauf

Version	Datum	Verfasser	Änderungen
1.0	06.12.2024	Team 10	Erste Fassung
2.0	07.01.2025	Team 10	Abgabe der Schlussdokumentation PREN 1

Tabelle 1: Versionsverlauf der Dokumentation

Abstract

Im interdisziplinären Projektmodul PREN 1 der Hochschule Luzern wurde ein Konzept für einen autonomen Roboter entwickelt, der ein vorgegebenes Wegenetzwerk selbstständig navigieren kann. Ziel war es, Hindernisse zu erkennen, gesperrte Wegpunkte zu meiden und dynamisch die effizienteste Route vom Start- zum Zielpunkt zu ermitteln. Der Roboter nutzt Sensordaten von LIDAR, Kamera und Ultraschallsensoren, um eine präzise Umgebungserkennung zu gewährleisten, und basiert auf einem heuristikbasierten Algorithmus, der flexibel auf unbekannte Graphen reagieren kann.

Besonderer Fokus lag auf der Konstruktion eines stabilen, gewichtsoptimierten Chassis, das den strengen Vorgaben des Projekts entspricht, sowie auf der Entwicklung eines motorisierten Greifers zur aktiven Hindernisbeseitigung. Zur Unterstützung der Konzeptentwicklung wurde ein Simulator implementiert, der eine frühzeitige Validierung der Navigationsalgorithmen in einer virtuellen Umgebung erlaubt. Nachhaltigkeitsaspekte wie energieeffiziente Komponenten, ressourcenschonende Materialwahl und der Bezug zu den Zielen für nachhaltige Entwicklung (SDGs) wurden systematisch berücksichtigt.

Die im Modul PREN 1 erarbeiteten Ergebnisse bilden die Grundlage für die praktische Umsetzung und den Bau des Roboters im nachfolgenden Modul PREN 2. Dort wird das Konzept weiter optimiert und die technische Machbarkeit unter realen Bedingungen evaluiert, bevor ein funktionales Gesamtsystem präsentiert wird.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Aufgabenstellung	1
2. Produktbeschreibung	2
2.1. Produktbeschreibung	2
2.2. Hardfacts	4
3. Entwicklung & Dimensionierung	5
3.1. Mechanik	5
3.1.1. Chassis und Fahrwerk	5
3.1.2. Greifeinheit	8
3.2. Elektronik	11
3.2.1. Steuerungstopologie und Gesamtübersicht	11
3.2.2. Bordnetz	13
3.2.3. Antriebe und Dimensionierung	16
3.2.4. Liniensor	21
3.2.5. Liniensor als PCB	23
3.2.6. Abstandssensoren	24
3.2.7. Sensorik Strecken Rückverfolgung	25
3.2.8. Grip Controller	27
3.2.9. Schnittstelle Raspberry Pi	28
3.3. Informatik	31
3.3.1. Beschreibung & Entscheid Algorithmen	31
3.3.2. Sensorik und Anforderungen zur Algorithmusintegration	33
3.3.3. Simulation	35
3.3.4. Tests	40
4. Projektmanagement	41
4.1. Organisation	41
4.2. Projektplanung	41
4.3. Risikoanalyse	43
4.4. Reflexion	43
5. Nachhaltigkeit	45
5.1. Nachhaltigkeitsbetrachtung	45
5.1.1. Ziele der Nachhaltigkeitsbetrachtung	45
5.1.2. Abgrenzung	45
5.1.3. Bezug zu den Sustainable Development Goals (SDGs)	45
5.2. Entscheide	46
5.2.1. Elektronik	46
5.2.2. Mechanik	47
5.2.3. Informatik	47
5.3. Reflexion	48
6. Schlussdiskussion	49
6.1. Finanzplanung und Ausgaben	49
6.1.1. Ausgaben Produktentwicklung 1	50
6.2. Gewichtsbudget	52

6.3. Erfahrungen & Lessons learned	53
6.3.1. Mechanik	53
6.3.2. Elektronik	53
6.3.3. Informatik	54
6.4. Fazit	55
6.4.1. Erfüllung der Anforderungen	55
6.4.2. Ausblick und Risiken	55
7. Verzeichnisse	57
7.1. Abbildungsverzeichnis	57
7.1. Tabellenverzeichnis	58
8. Anhang	60
A. Konzepterstellung Greifereinheit	60
B. Blockschaltbild Elektrokonzept	65
C. Konzepterstellung Energieversorgung	66
D. Konzepterstellung Antriebe	78
E. Konzepterstellung Hinderniserkennung	94
F. Konzepterstellung Strecken Rückverfolgung	99
G. Liniensor	101
H. Risikomanagement	107
I. Recherche	115
J. Anforderungsliste Version 2	119
K. Morphologischer Kasten	127
L. Nutzwertanalyse	130
M. Digitaler Anhang	132

1. Einleitung

1.1. Aufgabenstellung

Die Aufgabenstellung des Moduls Produktentwicklung PREN 1 der Hochschule Luzern fordert uns als interdisziplinäres Team heraus, ein autonomes Fahrzeug zu entwickeln, das ein vorgegebenes Wegenetzwerk optimal navigieren kann. Das Fahrzeug soll Hindernisse erkennen, gesperrte Bereiche meiden und die kürzeste Route zum Ziel unter Berücksichtigung unbekannter Einschränkungen finden.

Unser Team aus Studierenden der Elektrotechnik, Informatik und Maschinenbau vereint verschiedene technische Kompetenzen. In der Projektphase von PREN 1 entwickeln wir ein Gesamtkonzept basierend auf einem morphologischen Kasten, um Lösungsvarianten systematisch zu vergleichen. Dabei bewerten wir die technische Machbarkeit und erstellen erste Prototypen als Grundlage für das Modul PREN 2.

Zu den wichtigsten Anforderungen zählen:

- **Erkennung gesperrter Wegpunkte:** Pylonen müssen selbstständig detektiert werden.
- **Bewältigung von Hindernissen:** Hindernisse sollen aktiv entfernt und zurückgestellt werden.
- **Anpassung an veränderte Bedingungen:** Fehlende Streckenabschnitte müssen als nicht passierbar erkannt werden.
- **Autonome Zielfindung:** Über eine Zielauswahl vor dem Start soll das Fahrzeug die effizienteste Route finden.

Zudem werden bei der Umsetzung strenge Vorgaben zu Dimensionen, Gewicht und Autonomie des Systems berücksichtigt. Dies umfasst die Integration sämtlicher Hardware-Komponenten in das Fahrzeug sowie die Sicherstellung eines störungsfreien und sicheren Betriebs. Weiterhin erarbeiten wir in PREN 1 erste Simulationen, die das Verhalten des Systems unter realitätsnahen Bedingungen analysieren und optimieren.

Die wissenschaftliche Herausforderung besteht darin, Sensorik, Elektronik und Algorithmen so zu kombinieren, dass eine zuverlässige Wegfindung und Hindernisbewältigung gewährleistet ist. Dabei legen wir besonderen Wert auf eine methodische Herangehensweise, die auch Nachhaltigkeitsaspekte berücksichtigt.

Das Projekt bietet uns die Möglichkeit, theoretisches Wissen praktisch anzuwenden und unsere Kompetenzen in interdisziplinärer Zusammenarbeit und systematischer Produktentwicklung zu vertiefen.

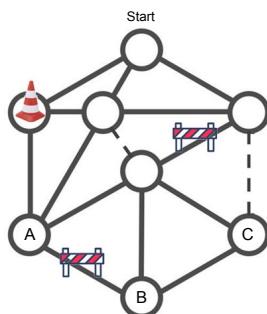


Abbildung 1: Beispiel Wegenetzwerk aus Aufgabenstellung

2. Produktbeschreibung

2.1. Produktbeschreibung

Das in diesem Projekt entwickelte System ist ein autonomes Fahrzeug, das in der Lage ist, sich selbstständig durch ein Wegenetzwerk zu navigieren. Es wurde speziell für die Aufgabe konzipiert, dynamisch auf Hindernisse und Einschränkungen zu reagieren, um den optimalen Weg von einem definierten Startpunkt zu einem Zielpunkt zu finden.

Das Fahrzeug zeichnet sich durch eine modulare Bauweise aus, die es ermöglicht, verschiedene Technologien effizient zu integrieren. Zu den zentralen Komponenten gehören:

- **Chassis und Antrieb:** Das Fahrzeug verwendet ein leichtes, stabiles Chassis, das auf einem dreirädrigen Konzept basiert. Zwei Antriebsräder sorgen für die Fortbewegung, während ein stabilisierender Auflagepunkt zusätzliche Balance bietet. Die Antriebssteuerung erfolgt über Schrittmotoren, die eine präzise Bewegung und Manövriertfähigkeit gewährleisten.
- **Sensorik:** Ein Set aus Liniensensoren, Kamera, Ultraschallsensor, Lichtschranke und einem LiDAR-System ermöglicht es dem Fahrzeug, die Umgebung präzise zu analysieren. Die Liniensensoren dienen ausschließlich der Verfolgung von Leitlinien. Schranken werden mithilfe von Ultraschallsensoren und/oder Lichtschranken erkannt, während LiDAR auf einer spezifischen Höhe dazu eingesetzt wird, Pylonen (gesperrte Wegpunkte) zu identifizieren. Die Kamera wird ausschließlich dafür genutzt, Knotenpunkte frühzeitig zu erkennen, bevor sie überfahren werden, sowie Zielknoten eindeutig zu identifizieren. Diese Sensoren arbeiten zusammen, um eine Echtzeitbewertung der Umgebung durchzuführen.
- **Steuerungseinheit:** Ein Mikrocontroller-basiertes System übernimmt die Verarbeitung der Sensordaten und die Steuerung der Motoren. Durch die Integration eines speziell entwickelten Algorithmus wird das Fahrzeug in die Lage versetzt, dynamisch auf Veränderungen der Umgebung zu reagieren.
- **Greifeinheit:** Für das Entfernen von Hindernissen ist das Fahrzeug mit einem motorisierten Greifer ausgestattet. Dieser kann Objekte sicher greifen, anheben und präzise zurücklegen.
- **Energieversorgung:** Das System wird durch einen kompakten Lithium-Polymer-Akku betrieben, der eine ausreichende Betriebsdauer für die vorgegebene Aufgabenstellung sicherstellt. Zusätzliche Sicherheitsfunktionen wie eine Überwachung der Zellspannung und ein Not-Aus-Knopf garantieren die Betriebssicherheit.
- **Benutzerinterface:** Über einen Wahlschalter kann vor dem Start die Zielposition ausgewählt werden. Visuelle und akustische Signale zeigen den Abschluss der Aufgabe an.
- **Simulator:** Zur Unterstützung der Entwicklung und zur frühzeitigen Validierung von Algorithmen hat das Informatik-Team einen Simulator in *Svelte.js* entwickelt. Der Simulator ermöglicht es, das Verhalten des Fahrzeugs in einer virtuellen Umgebung zu testen, bevor physische Prototypen gebaut werden. Mit Funktionen wie der Simulation von Hindernissen, gesperrten Wegpunkten und alternativen Routen erlaubt der Simulator eine präzise Analyse der Navigations- und Steuerungsalgorithmen. Zudem bietet das Tool eine interaktive Benutzeroberfläche, um Szenarien flexibel zu konfigurieren und Ergebnisse in Echtzeit zu visualisieren.

Das Fahrzeug wurde so konzipiert, dass es sowohl den technischen Anforderungen als auch den strengen Gewichtsvorgaben gerecht wird. Dank seiner modularen Struktur und der klaren Trennung von Hardware- und Softwarekomponenten ist es flexibel erweiterbar und anpassungsfähig. Dies ermöglicht es, das System im weiteren Projektverlauf iterativ zu verbessern und zu optimieren.

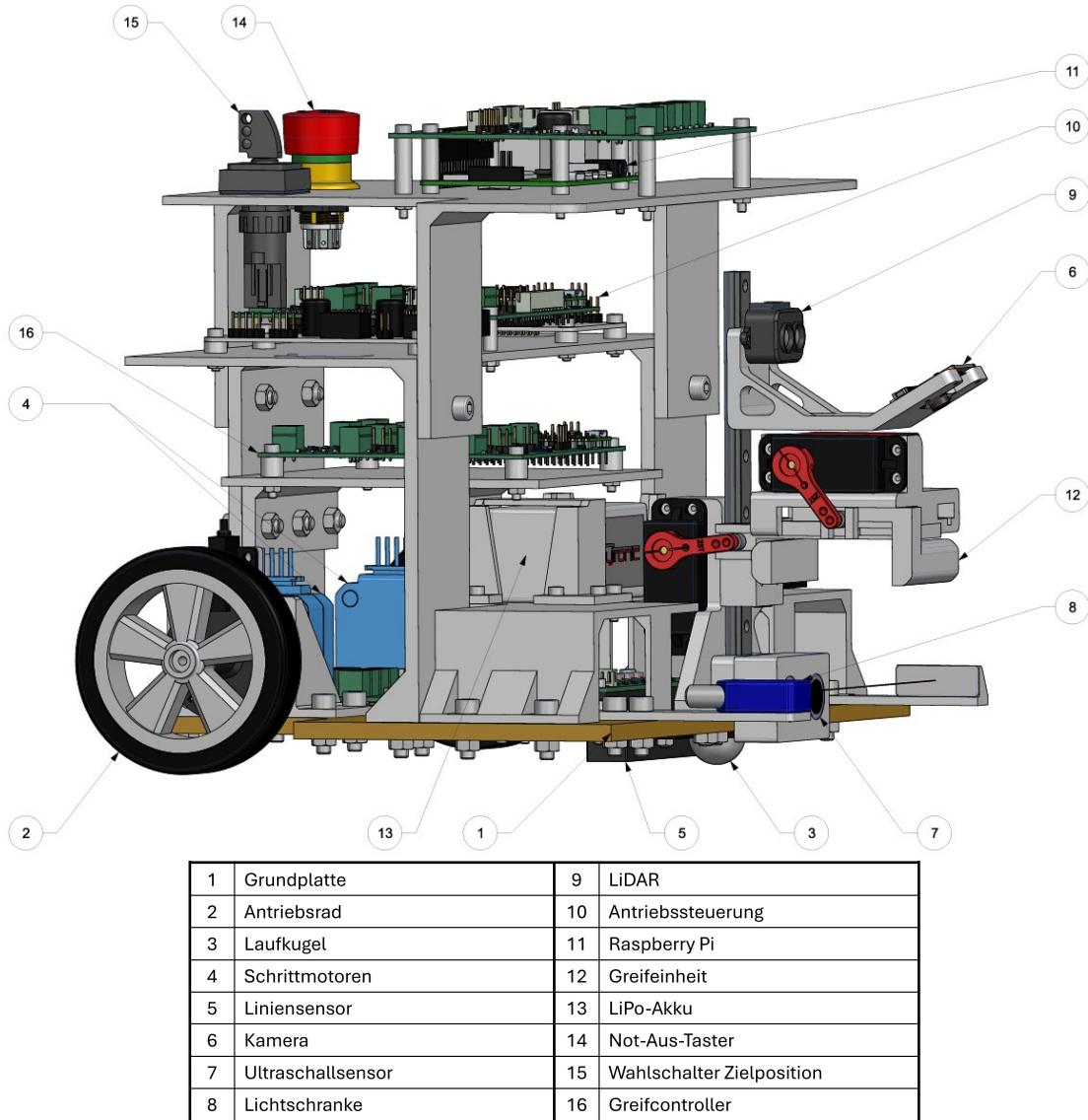


Abbildung 2: Prototyp des Fahrzeugs mit Bildlegende in Siemens NX

2.2. Hardfacts

Die folgenden technischen Daten beschreiben die wichtigsten Eigenschaften und Leistungsparameter des entwickelten Fahrzeugs:

Eigenschaft	Wert
Energieversorgung	4S LiPo-Akku (14.4V, 1300 mAh)
Betriebsdauer	ca. 20 Minuten (durchschnittlich)
Maximale Traglast der Greifeinheit	300 g
Abmessungen	30 cm x 30 cm x 30 cm
Gewicht	2 kg
Navigationsalgorithmus	Heuristikbasierter DFS
Sensoren	Liniensor, Abstandssensoren, Kamera, Lichtschranken, Encoder, Gyroskop
Kommunikationsschnittstellen	UART, I2C
Steuerungseinheit	Raspberry Pi 4
Greifermechanik	Motorisierter Parallelgreifer
Entwicklungstool für Simulation	Svelte.js-basierter Simulator

Tabelle 2: Technische Daten des autonomen Fahrzeugs

3. Entwicklung & Dimensionierung

3.1. Mechanik

3.1.1. Chassis und Fahrwerk

Dieser Abschnitt beschreibt die Evaluierung des Fahrwerks sowie die Konstruktion des Chassis. Dabei werden die Anforderungen an Gewicht, Kosten und Funktionalität sowie die konstruktiven Entscheidungen im Detail erläutert.

Anforderungen

Gewicht:

Das Gewicht stellt bei allen Baugruppen einen kritischen Faktor dar. Da leistungsstarke Schrittmotoren und der Akku bereits einen Grossteil des Gewichtsbudgets beanspruchen, verbleibt für das Chassis lediglich ein Budget von 200 Gramm. Dies erfordert eine leichte, aber dennoch stabile Konstruktion. Die Gewichtsoptimierung wurde bereits in der frühen Planungsphase berücksichtigt, um sicherzustellen, dass das Fahrzeug die Anforderungen hinsichtlich Stabilität und Belastbarkeit erfüllt.

Kosten:

Um die Kosten niedrig zu halten, soll auf die frei verfügbaren Ressourcen der Hochschule Luzern zurückgegriffen werden. Das Team kann 25 Stunden Druckzeit am HSLU-T&A-Drucker sowie 1 Stunde Maschinenlaufzeit des Lasersystems kostenfrei nutzen. Diese Ressourcen werden optimal ausgeschöpft, um den Einsatz von Einkaufsteilen zu minimieren. Ziel ist es, das verfügbare Budget vor allem für funktionskritische Baugruppen wie den Greifmechanismus und die Antriebssteuerung einzusetzen.

Chassiskonstruktion

Das Chassis basiert auf einer 6 mm dicken Grundplatte aus MDF, einem leichten und kostengünstigen Material. MDF bietet den Vorteil, dass es einfach zu bearbeiten ist und alle notwendigen Ausschnitte präzise mit dem Lasersystem der Hochschule gefertigt werden können. Abbildung 3 zeigt das Design der Grundplatte des ersten Prototyps. Die Bauteilplatzierung ist vorläufig und wird in PREN 2 voraussichtlich weiter optimiert. In dieser Phase wird die Grundplatte auch mit zusätzlichen Aussparungen versehen, um weiteres Gewicht einzusparen.

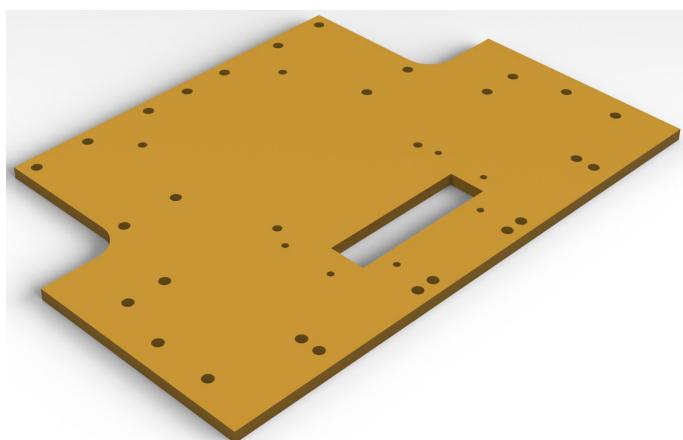


Abbildung 3: Grundplatte in Siemens NX

Abbildung 4 zeigt das Chassis mit allen Anbauteilen, die mit dem 3D-Drucker gefertigt werden. Diese Konstruktion ist für Prototypentests gedacht und daher nicht platzsparend ausgelegt. Sie ermöglicht eine einfache Verkabelung der Leiterplattensowie den schnellen Austausch einzelner Komponenten. Der modulare Aufbau erlaubt eine problemlose Anpassung in PREN 2. Obwohl das aktuelle Design das Gewichtsbudget knapp überschreitet, bestehen zahlreiche Möglichkeiten zur weiteren Optimierung. So ist geplant, die Anbauteile mit Aussparungen zu versehen und mit einer Art Wabenstruktur aufzubauen, um das Gewicht zu reduzieren, ohne die Stabilität zu beeinträchtigen. Diese Strukturen sind besonders geeignet, da sie eine hohe Festigkeit bei minimalem Materialeinsatz bieten und sich im 3D-Druck effizient umsetzen lassen.

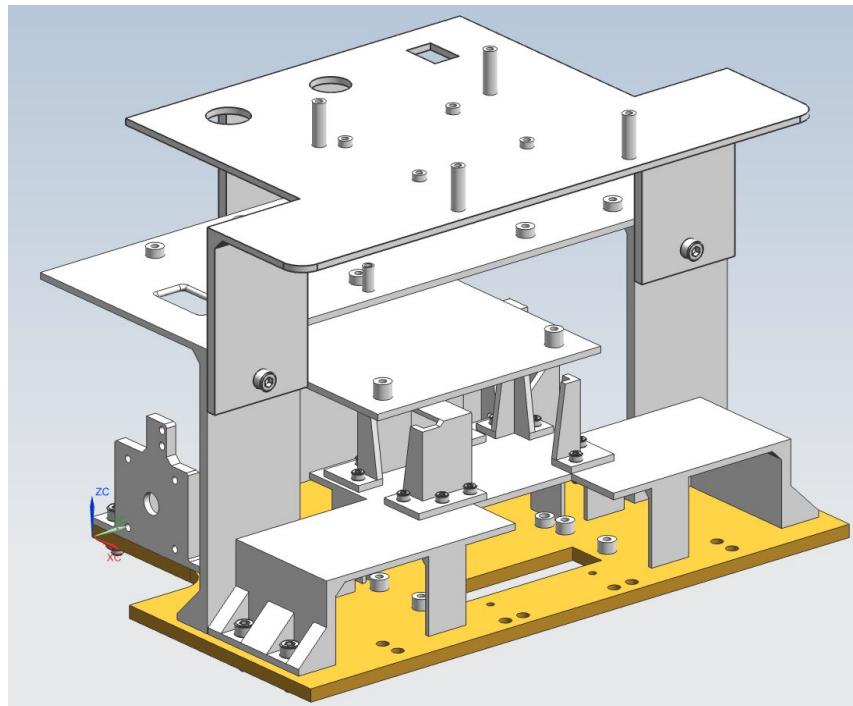


Abbildung 4: Chassis mit sämtlichen Anbauteilen in Siemens NX

Fahrwerk

Das Fahrwerk basiert auf einem Design mit zwei angetriebenen Rädern und einer Laufkugel als dritten Auflagepunkt, um Stabilität zu gewährleisten. Diese Lösung minimiert die Anzahl der Einkaufsteile und damit die Kosten. Zudem bietet sie eine hohe Wendigkeit des Fahrzeugs. Abbildung 5 zeigt die Umsetzung mit den beiden Hinterrädern und der vorderen Laufkugel. Die Platzierung der angetriebenen Räder an der Hinterachse ermöglicht präzise Kurskorrekturen, da das Fahrzeug durch kleine Bewegungen der Hinterräder seine Ausrichtung leicht ändern kann. Dies erlaubt es dem Fahrzeug, auf Knotenpunkten effizient zu drehen und verschiedene Wegabzweigungen zu überprüfen.

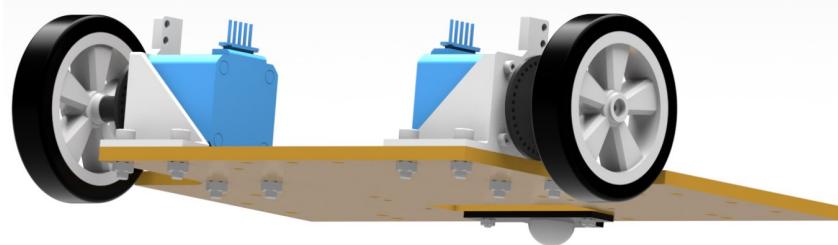


Abbildung 5: Grundplatte mit Rädern und Laufkugel in Siemens NX

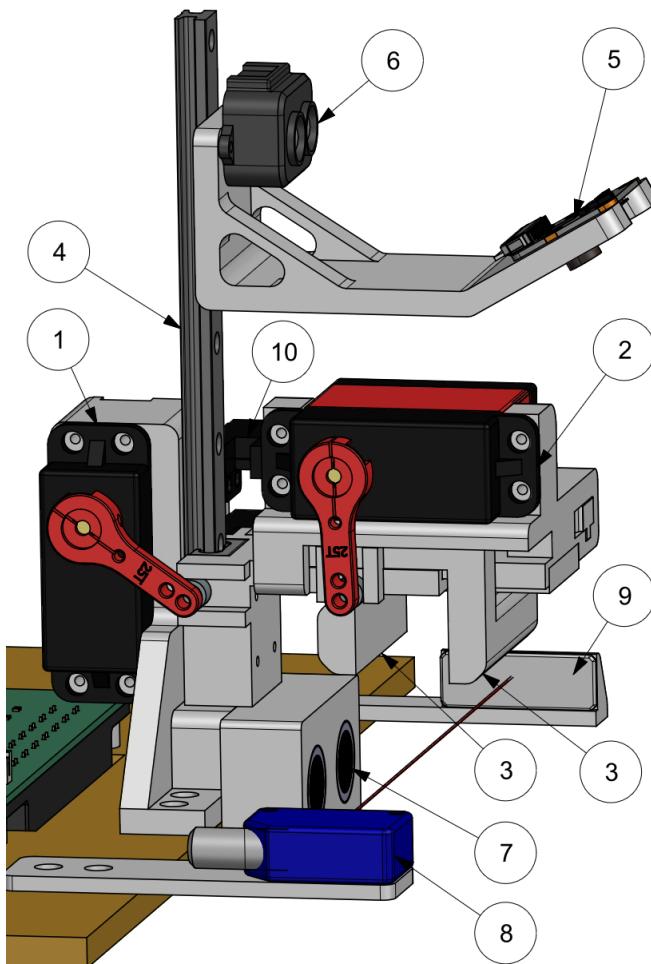
Die Radgrösse wurde auf 80 mm festgelegt, wodurch eine maximale Geschwindigkeit von 1,676 m/s erreicht wird. Die Berechnung erfolgt nach folgender Formel:

$$v_{max} = n_{Motor} \cdot d_{Rad} \cdot \pi = 1.676 \frac{\text{m}}{\text{s}}$$

Die Räder vom Typ FIT0500 von DFRobot wurden über eine Sammelbestellung der Hochschule Luzern beschafft, um die Umweltbelastung durch Lieferungen zu reduzieren. Die endgültige Auswahl der Laufkugel steht noch aus. Vier verschiedene Modelle wurden bestellt, die Tests und die finale Entscheidung erfolgen jedoch erst in PREN 2, da ein Test ohne das komplette Fahrzeug nicht aussagekräftig wäre. In dieser Phase werden auch die Auswirkungen der Laufkugel auf das Fahrverhalten genauer untersucht, um sicherzustellen, dass das Fahrzeug unter verschiedenen Bedingungen zuverlässig und stabil agiert.

3.1.2. Greifeinheit

Die Abbildung 6 veranschaulicht das Greiferkonzept des Roboters. Dieses Konzept wurde basierend auf der Evaluierung im Anhang J weiterentwickelt. Im folgenden Abschnitt wird die Funktionsweise des Greifers im Detail beschrieben.



01	linker Servomotor
02	rechter Servomotor
03	Greiferbacken
04	IGUS Gleitführung
05	Raspberry Pi Kamera
06	Lidar
07	Ultraschallsensor
08	Lichtschranke (Hindernis)
09	Reflektor (Hindernis)
10	Lichtschranke (Höhenverstellung)

Abbildung 6: Übersicht der Greifereinheit im NX Siemens

Der Greifer verfügt über zwei Backen, die parallel zueinander verschoben werden, um das Hindernis sicher zu greifen. Dieser Vorgang wird von einem Servomotor gesteuert. Konkret bewegt der Hebel vom Servomotor eine der beiden Backen horizontal. Diese Backe ist mit einer Zahnstange ausgestattet, die ein innenliegendes Zahnrad antreibt. Das Zahnrad überträgt die Bewegung auf die gegenüberliegende Backe, sodass sich beide Backen synchron, aber in entgegengesetzter Richtung bewegen. Die Höhenverstellung ist ebenfalls mit einem Servomotor gelöst. Dieser bewegt den Greifer entlang einer Gleitführung von IGUS nach oben und unten. Die benötigten Drehmomente für den Servomotor befinden sich im Anhang J.

Die Abbildung 7 zeigt den Ablauf des Greifvorgangs in mehreren Bildern. Die Bewegung des Motors werden mit einem gelben Pfeil dargestellt:

Positionierung der Greifeinheit Von Position 1 zu 2 dreht der linke Servomotor die Greifeinheit entlang einer Gleitführung nach unten, nachdem ein Hindernis erkannt wurde.

Greifen des Hindernisses Von Position 2 zu 3 bewegt der rechte Servomotor die Backen horizontal, wodurch sich beide Backen parallel schliessen und das Hindernis greifen.

Anheben des Hindernisses Von Position 3 zu 4 dreht der linke Servomotor die Greifeinheit leicht nach oben, sodass das Hindernis vom Boden angehoben wird. Dadurch wird ein sicherer Transport des Hindernisses gewährleistet.

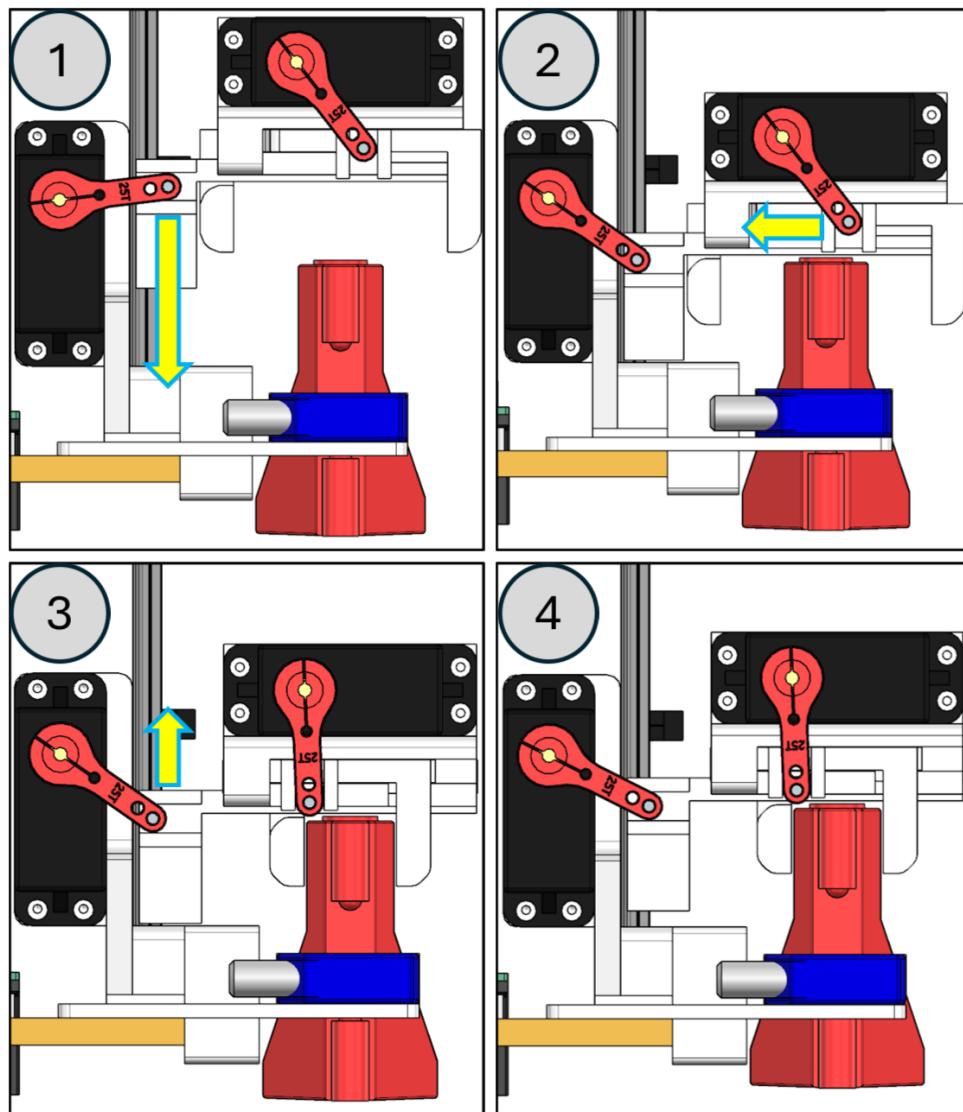


Abbildung 7: Greiferablauf - Anheben des Hindernis

Die folgende Abbildung 8 zeigt einen funktionierenden Greiferprototyp aus einer vorherigen Iteration.

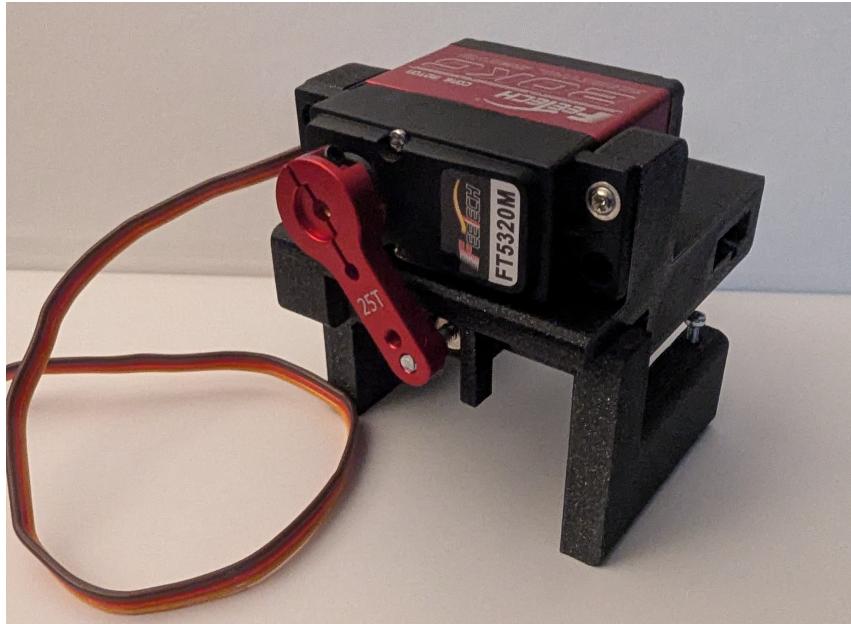


Abbildung 8: Greiferprototyp

Sensorikplatzierung

In diesem Abschnitt werden die verschiedenen Sensoren an der Greifereinheit beschrieben.

Raspberry Pi Kamera - Knotenpunkt Die Raspberry Pi Kamera ist auf Höhe 130 mm in einem Winkel von 30° montiert. In dieser Position kann sie Objekte mit einem Durchmesser von circa 260 mm auf dem Boden erkennen. Dadurch wird sichergestellt, dass die 120 mm grossen Knotenpunkte sowie die zugehörigen Knotenabgänge erfasst werden.

Ultraschallsensor und Lichtschranke - Hinderniss Der Ultraschallesnor ist auf Höhe 25 mm montiert. Er erkennt Hindernisse frühzeitig, um Kollisionen mit dem Roboter zu vermeiden. Er sorgt zusätzlich dafür, dass der Roboter an der richtigen Position stoppt, um das Hindernis anzuheben. Falls die Positionserkennung für das Greifen des Hindernisses im PREN 2 ungenau ist, wird eine weitere Lichtschranke eingesetzt. Die genaue Funktionsweise wird im Kapitel 3.2.6 ausführlicher erklärt.

Lidar - Pylon Der Lidar ist in einer Höhe von 150 mm montiert, wodurch er bis zu einer Entfernung von 2 Metern einen Pylon erkennen kann. Diese Höhe wird benötigt, sodass die Streuung des Lidars auf 2 Metern keinen Hindernis erfasst. Die genaue Funktionsweise wird im Kapitel 3.2.6 ausführlicher erklärt.

Lichtschranke - Höhenverstellung des Greifers Die Lichtschranken sind an den Endpositionen der Höhenverstellung montiert. Sie stellen sicher, dass die Greifer die richtige Höhe erreicht und verhindern ein Überdrehen des Servomotors, um Beschädigungen an den Komponenten zu vermeiden.

3.2. Elektronik

3.2.1. Steuerungstopologie und Gesamtübersicht

Abbildung 9 zeigt das Elektrokonzept in einem kleinen Detaillierungsgrad. Ziel ist es, im Sinne der Gewaltenteilung jede Steuerungseinheit für sich abgeschlossen zu betrachten. So kann sowohl die Zuständigkeit der beiden Elektronikentwickler gut aufgeteilt werden, als auch Schnittstellen zwischen Funktionseinheiten sehr eindeutig benannt werden.

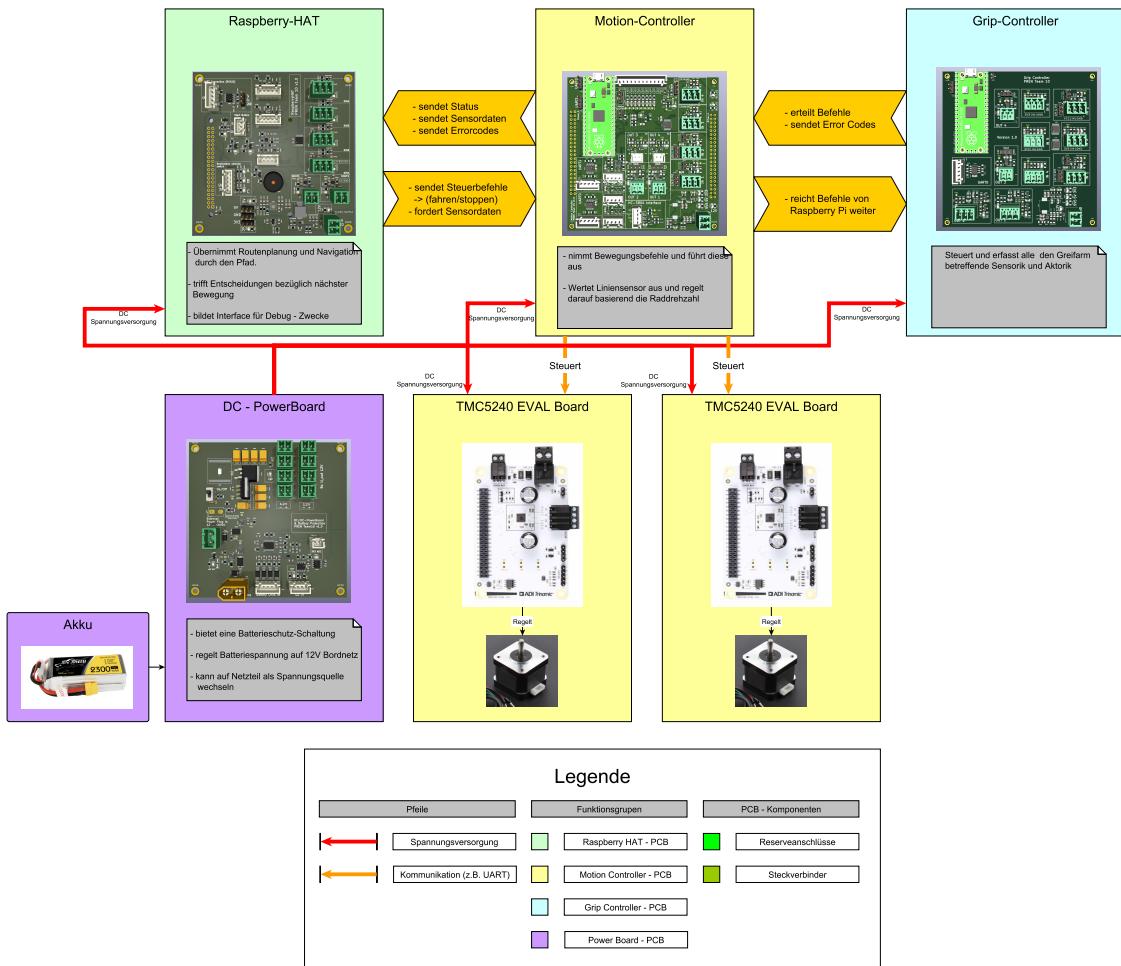


Abbildung 9: Vereinfachte Gesamtübersicht Elektronikkonzept

Anhang B zeigt die Gesamtübersicht der Steuerung zum Zeitpunkt von PREN1 nochmals in einer höheren Detaillierung.

Die verschiedenen Funktionsgruppen und ihre PCBs

Es werden total die 4 folgenden Funktionsgruppen unterschieden:

Raspberry HAT Dieser PCB hat die Aufgabe, die nötige Peripherie für den Raspberry Pi zur Verfügung zu stellen. Dies beinhaltet die Anbindung verschiedener Taster an die entsprechenden GPIOs sowie eine Spannungsversorgung. Der Raspberry Pi soll aus-

serdem über einen Summer Signale wie das Erreichen des Zielpunktes signalisieren können. Weiter wird sein UART-Kanal für die Kommunikation aufbereitet.

Motion Controller Dieser PCB bietet die Schnittstelle aller Steuereinheiten zu den Antrieben. Via UART werden diesem PCB Steuerkommandos wie *fahren mit Geschwindigkeit xxx, Fahrzeug um xxx° drehen, Hindernis erkannt: Anhalten und warten* und viele weitere mitgeteilt. Er selbst beinhaltet Hardware zum Auszählen der Encoder sowie des Gyroskops und noch einige digitale Ein-/Ausgänge und Reserve-Sensorschnittstellen.

Grip Controller Dieser PCB steuert, wie der Name erahnen lässt, alles rund um das Greifen und um den Umgang mit Hindernissen. Er erkennt das Hindernis und kann dem Motion Controller via UART Kommandos erteilen, wie das Anhalten oder dass das Fahrzeug gedreht werden soll.

Power Board Das Power Board bietet die Spannungsaufbereitung für den Roboter. Hier werden aus den 14.4V Batteriespannung 12V Boardnetzspannung eingeregelt. Weiter wird die Batterie überwacht und bei drohender Unterspannung getrennt. Ausbalanciert wird eben diese Batterie über das externe Ladegerät. Wird ein 14...18V Netzteil an diesem Board angeschlossen, wird die Batterie vom Fahrzeug getrennt, was Einricht- und Entwicklungsarbeiten erlaubt, ohne dass dazu eine Batterie benötigt wird.

Zum Stand von PREN1 wäre der Motion Controller grundsätzlich in der Lage, die Aufgabe des Greifers ebenfalls zu übernehmen. Wenn sich der noch nicht abgeschlossen getestete Greifmechanismus nicht eignet und mehr Motoren oder Sensoren benötigt, hätte es auf diesem nicht mehr genug Schnittstellen für Sensorik und Aktorik zur Verfügung. Daher wird mit dem separaten Grip Controller geplant.

Kommunikationsschnittstellen

Da Motoren, Schaltregler und andere Leistungselektronik auf engstem Raum mit Kommunikations- und anderer empfindlicher Elektronik untergebracht werden müssen, wird die UART-Kommunikation differentiell über eine RS422-Schnittstelle übertragen. Die Kommandos und das Protokoll werden erst im Nachfolgemodul bei der Inbetriebnahme dieser Leiterplatten endgültig definiert. Angedacht ist eine Kommunikation über eine Kommandolänge von z.B. 64 Bit, in der eine Adresse, ein Kommando und zusätzlich noch genügend Platz für zu übertragende Parameter definiert sind. Abbildung 10 zeigt den Aufbau einer solchen Nachricht. Mögliche Parameter können eine Fahrzeuggeschwindigkeit, ein Drehwinkel oder auch Fehlercodes sein, die an den Raspberry Pi zurückgesendet werden.

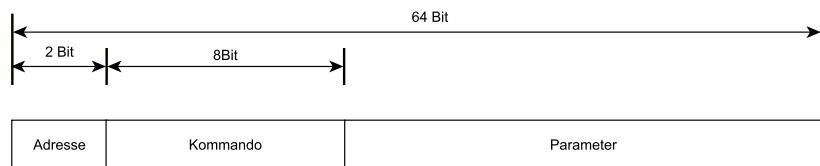


Abbildung 10: Aufbau einer Nachricht

Die Adressvergabe, dargestellt in Abbildung 11, ist vorgesehen, da der Raspberry Pi aufgrund der Kommunikationstopologie, dargestellt im Blockschaltbild 9, grundsätzlich nicht

in der Lage ist, direkt mit dem Greifcontroller zu kommunizieren. Dies ist prinzipiell auch nicht notwendig. Um aber genau diesen Fall abzudecken, sollten Adressen von 0...2 (2-Bit) vergeben werden. So kann eine einfache Punkt-zu-Punkt-Übertragung ausreichen, um die drei Steuerungen miteinander kommunizieren zu lassen.

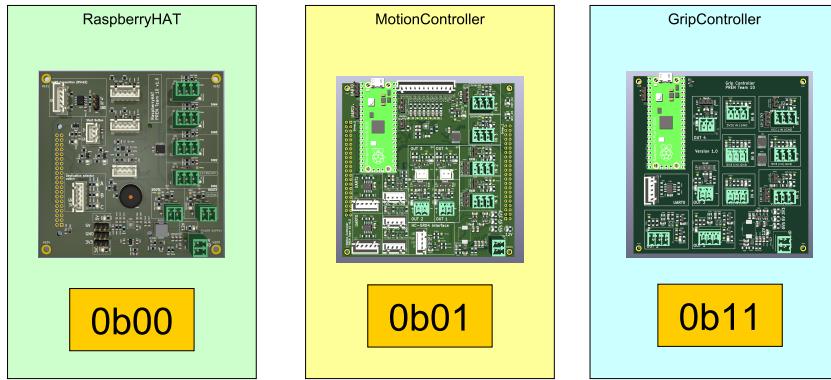


Abbildung 11: Vergebene Adressen der Prints

Bordnetz

Auf dieses wird in der Sektion im Detail eingegangen. Kurz ausgedrückt besitzt jedes Board eine eigene Spannungswandlung von 12V auf 5V respektive 3.3V, welche vom Power Board zur Verfügung gestellt werden.

3.2.2. Bordnetz

Der folgende Abschnitt befasst sich mit der Energieversorgung des Scouts. Wie die Evaluation der Energieversorgung gezeigt hat, siehe Anhang C, wird der Roboter über einen 4S-LiPo mit Energie versorgt. Der evaluierte Akku enthält $\approx 19Wh$ Energie und bietet damit genügend Energie, um den Roboter im ungünstigsten Fall für ca. 20min mit voller Leistung zu betreiben. Die Bordnetzspannung ist auf 12V festgelegt, um auf Sensorik und Aktorik, wie z.B. Lichtschranken, aus dem industriellen Umfeld zurückgreifen zu können. Außerdem kann so eine grössere Bandbreite an Motoren abgedeckt werden, falls sich die vorgesehenen als ungeeignet erweisen sollten.

Bordnetz - Spannungsverteilung Jede Leiterplatte wird mit einer Spannung von 12 V versorgt. Auf den jeweiligen Leiterplatten werden die geregelten 12V zunächst über einen DC-DC Abwärtsregler auf $\approx 6V$ abgesenkt und dann über einen LDO aktiv auf 5V bzw. 3V3 gefiltert. Dadurch wird verhindert, dass sich Spannungsspitzen oder -ripples, z.B. durch anlaufende Motoren, im gesamten System ausbreiten und andere Elektronik stören. Außerdem wird das System dadurch robuster gegenüber Bordnetzschwankungen, die z.B. durch anlaufende Motoren verursacht werden. Für die LDOs wird ein PSSR angestrebt, der bei entsprechenden Taktraten der Abwärtsregler eine ausreichende Dämpfung von mindestens 20dB aufweist. Das Blockschaltbild in Abbildung 12 zeigt konzeptionell die Spannungsverteilung. Es zeigt das entsprechende *PowerBoard*, das eine *Beispiel-Platine* mit 12V versorgt. Auf dieser Leiterplatte (gelb dargestellt) wird die Spannung dann entsprechend auf 5V reduziert. Es wird angestrebt, die Spannungsversorgung für alle PCB's mit den gleichen Bauteilen zu realisieren.

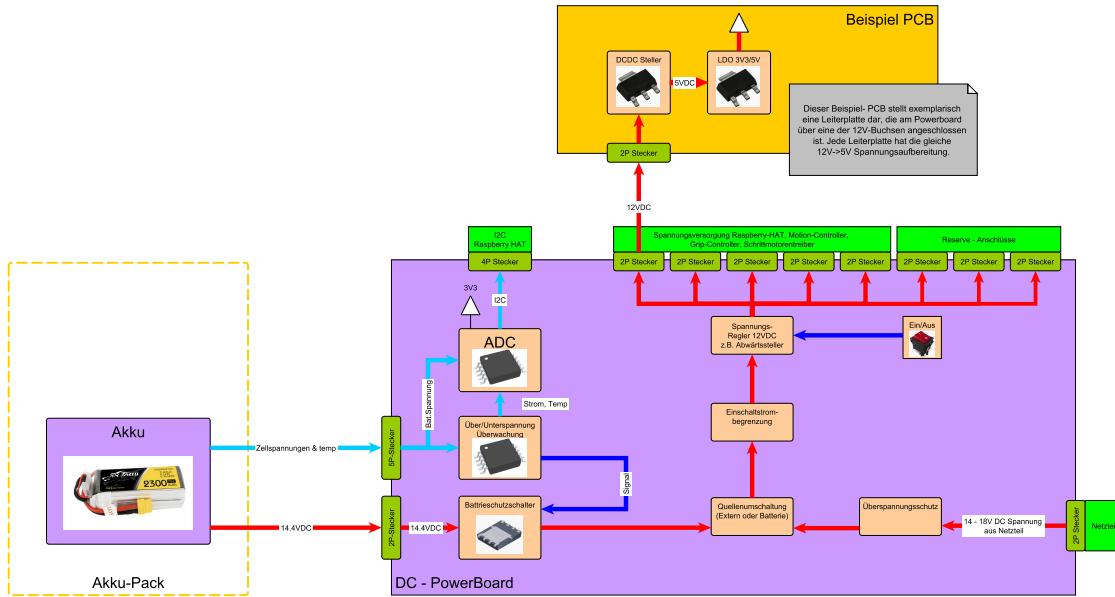


Abbildung 12: PowerBoard Blockschaltbild

PowerBoard-PCB Die Ausgangsspannung von 14,4V wird auf dem selbstentwickelten PowerBoard mit einem Abwärtsregler auf 12V geregelt. Dieses Board bietet auch die Möglichkeit, von einem Netzteil im Bereich von 12 bis 18V versorgt zu werden, wobei die Batterie automatisch abgeschaltet wird. Damit ist der Roboter bei Entwicklungs- und Einrichaufgaben nicht zwingend auf die Energieversorgung über die Batterie angewiesen, sondern kann auch über ein Netzteil betrieben werden.

Batteriespannung und -strom werden über einen ADC überwacht und können über I^2C vom Raspberry Pi ausgelesen werden.

Die Batterieschutzsitzung überwacht die Zellspannungen des LiPo-Akkus einzeln und trennt den Akku vom Roboter, wenn eine der Zellen in den Unterspannungsbereich gerät. Dieser Zustand wird durch eine rot leuchtende LED angezeigt. Der Zellenausgleich wird nicht am Roboter selbst durchgeführt. Zu diesem Zweck wurde ein Ladegerät aus dem Modellbaubereich beschafft, das in der Lage ist, die Zellen aktiv auszubalancieren.

Der Schaltplan dieser Platine und das Layout sind im Anhang beigelegt. Abbildung 13 zeigt eine 3D-Ansicht dieser Platine.

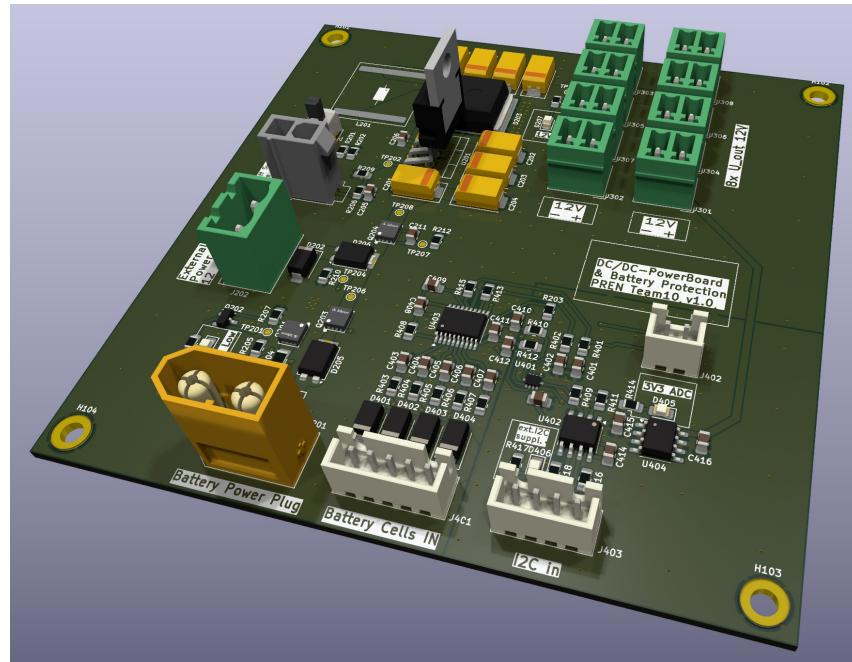


Abbildung 13: Ansicht PowerBoard

GND-Konzept Abbildung 14 zeigt das GND-Konzept des Roboters. Im Anschluss wird dieses noch ein wenig ausgeführt.

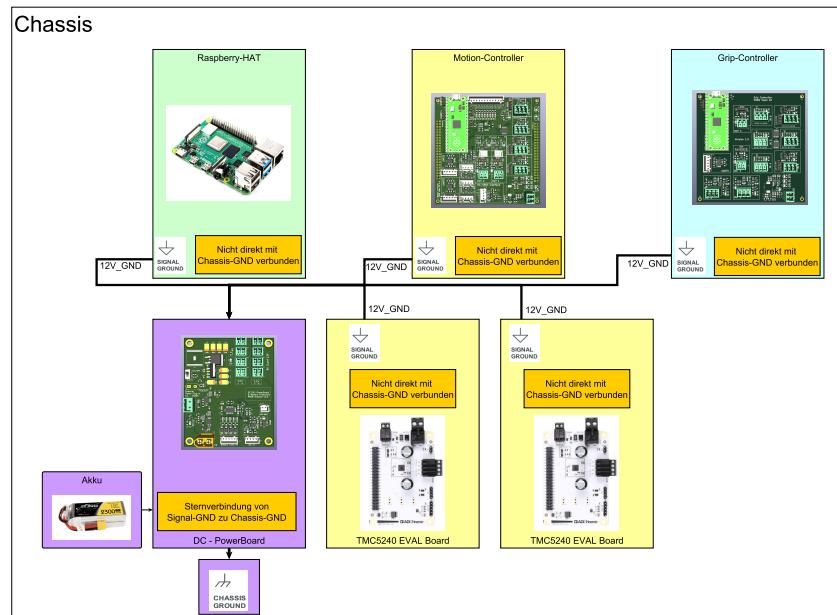


Abbildung 14: Blockschaltbild GND-Konzept

Da sich auf dem Roboter mehrere Leiterplatten befinden, die miteinander kommunizieren und interagieren, muss sichergestellt werden, dass sich diese auf einem ähnlichen Potential befinden. Einfache Überlegungen führen schnell zu der zu der Schlussfolgerung, dass jede Leiterplatte mit ihren Befestigungsschrauben auf GND-Potential mit einem leitfähigen

Chassis verbunden wird. Wird so verfahren, können Signalrückführungen nicht genau vorhergesagt werden und es können sich GND-Schleifen bilden. In Anbetracht der Tatsache, dass sich Motoren und damit sich ändernde Magnetfelder auf dem Gerät befinden, müssen genau diese GND-Schleifen vermieden werden und es kann nicht einfach jede Platine leitend mit der Grundplatte verschraubt werden. Dem wird dadurch entgegengewirkt, dass jede Leiterplatte eine 2-polige Spannungsversorgung erhält und nur über diese eine Leitung mit dem GND-Netz verbunden ist. Die Befestigungslöcher sollten isoliert mit der Grundplatte verbunden werden, damit sich keine GND-Schleifen über das Gehäuse bilden können. In diesem Fall laufen alle GND-Verbindungen sternförmig auf dem PowerBoard zusammen. Leichte Potentialunterschiede werden dadurch ausgeglichen, dass die Kommunikationssignale über eine RS422-Schnittstelle übertragen werden und somit resistent gegen Potentialunterschiede von $\pm 15V$ sind.

Wird anstelle eines organischen Materials ein Metallgehäuse für das Chassis verwendet, so ist der GND eines freien PowerBoard-Ausgangs mit diesem zu verbinden. Die Leiterplatten selbst werden mit Kunststoffverschraubungen an das Gehäuse geschraubt, um die Befestigungslöcher von diesem zu isolieren. Auf diese Weise laufen alle GND-Potentiale sternförmig an einem Punkt auf dem Chassis zusammen und das Chassis kann nicht als Antenne fungieren.

3.2.3. Antriebe und Dimensionierung

Dieser Abschnitt befasst sich mit der Auswahl eines geeigneten Antriebs mit Steuerung für den Roboter. Die Evaluation der Antriebe, siehe Anhang D, hat ergeben, dass ein Schrittmotor der Firma DFRobot, siehe Abbildung 15, eingesetzt wird. Dieser ist in der Lage, den Roboter mit 1.7A auf bis zu $2\frac{m}{s}$ zu beschleunigen.



Abbildung 15: Schrittmotor

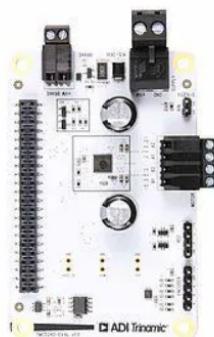


Abbildung 16: Evaluationboard TMC5240

Die Ansteuerung dieser Motoren erfolgt über einen vollintegrierten Schrittmotortreiber der Firma ADI-Trinamic. Um den Entwicklungsaufwand geringzu halten, wird auf 2 Evaluation-Boards des Treiber-IC's *TMC-5240* zurückgegriffen, wie in Abbildung 16 gezeigt. Eines der Teammitglieder hat bereits Erfahrung mit diesem speziellen Treiber und kann auf entsprechende Treiber-Evaluation-Boards aus seinem beruflichen Umfeld zurückgreifen.

Abbildung 17 zeigt schematisch, wie diese Treiber angesteuert werden.

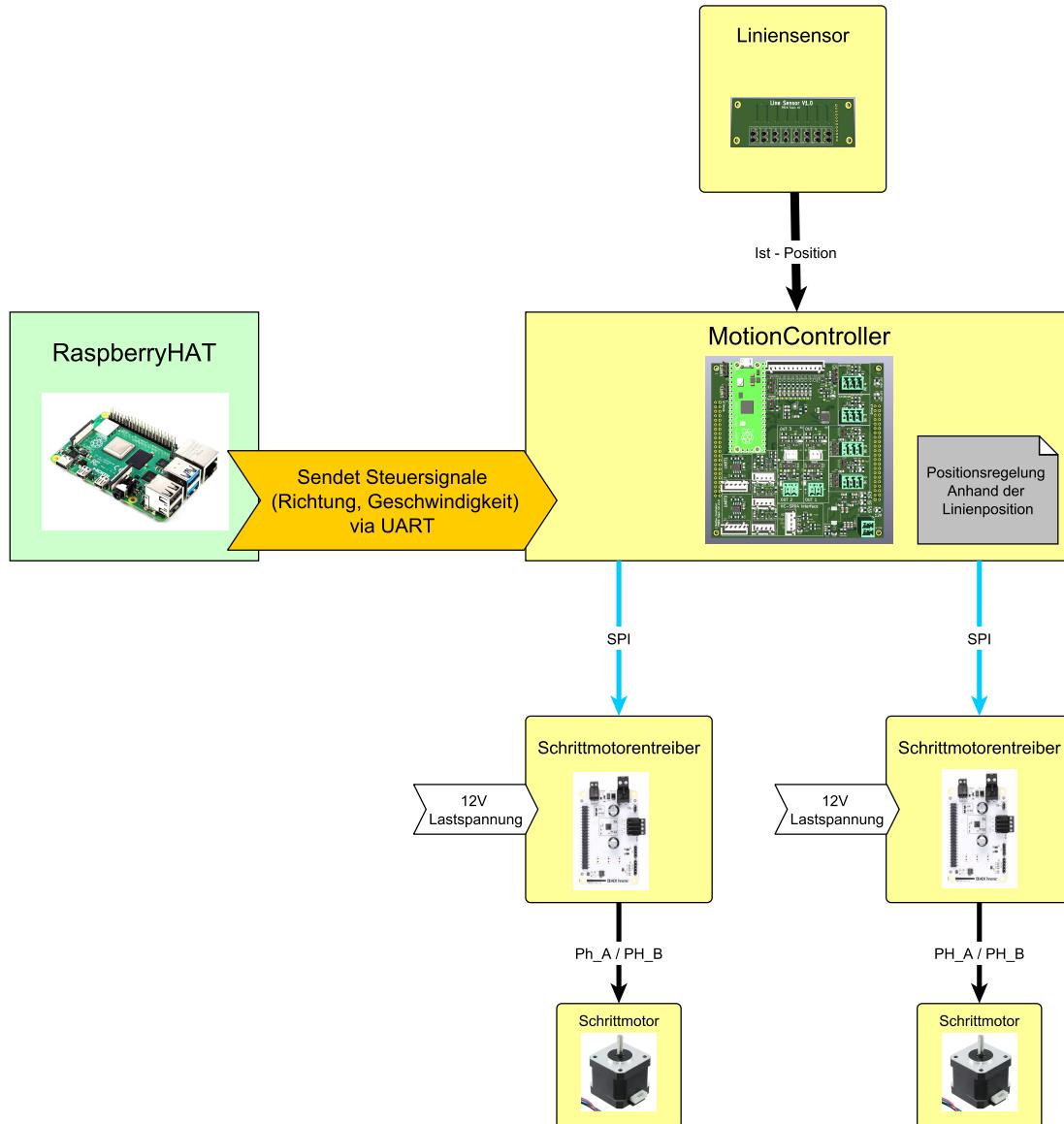


Abbildung 17: Ansteuerungstopologie Schrittmotoren

Die Antriebsregelung ist auf einer separaten Leiterplatte realisiert. Als Eingangssignal für diese Regelung dient nur der Liniensensor. Die Ansteuerung der beiden Schrittmotortreiber erfolgt über den SPI-Bus des Mikrocontrollers.

An den Rädern sind zusätzlich Encoder vorgesehen, damit der Navigationsrechner den zurückgelegten Weg verfolgen kann. Aus Gründen der Echtzeitfähigkeit bei der Auswertung der Encoder werden diese auf der Antriebssteuerung und nicht auf dem Navigations-

rechner ausgewertet. Die Encoder stellen im aktuellen Projektstand eine Fallback-Lösung dar. Im Folgemodul wird noch evaluiert, ob die gefahrenen Schritte, die aus dem Motorcontroller ausgelesen werden können, ausreichen, um die gefahrene Strecke rekonstruieren zu können.

Signale, in welche Richtung das Fahrzeug bewegt werden soll, sowie Start- und Stoppsignale erhält der Mikrocontroller vom Navigationsrechner über eine UART-Schnittstelle. Ebenfalls über UART kann der Greifcontroller Signale an den Antriebscontroller senden, die die Antriebe stoppen - das Fahrzeug drehen oder weiterfahren lassen.

Abbildung 18 zeigt schematisch, welche Funktionsgruppen auf der entsprechenden Platine enthalten sind.

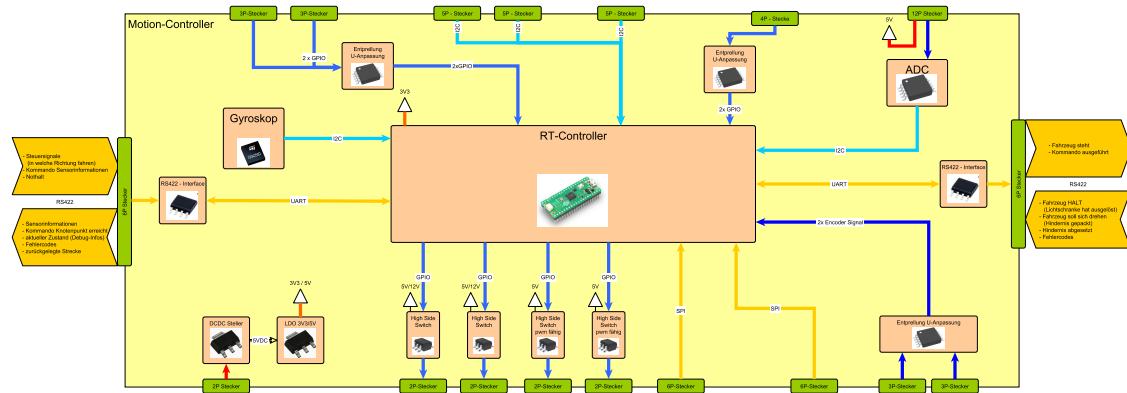


Abbildung 18: Blockschaltbild Motion Controller

Der Motion Controller verfügt über digitale Ein- und Ausgänge mit umschaltbaren Spannungspegeln als Reserve, falls im Verlauf des Folgemoduls PREN2 Sensoren angepasst oder zusätzliche Aktoren hinzugefügt werden müssen. Abbildung 19 zeigt das Motion Controller PCB in einer 3D-Ansicht. Es verfügt über 3 I²C-Anschlüsse, ein spezielles HC-SR04-Ultraschallsensor-Interface und die bereits erwähnten digitalen Ein-/Ausgänge. Zusätzlich ist eine Schnittstelle für den Liniensensor vorgesehen und ein Gyroskop zur Winkelerfassung integriert. Es wird wie eine Brücke auf die beiden Evaluation Boards von Trinamic gesteckt, wofür sich die breiten Steckerleisten an den Außenkanten befinden.

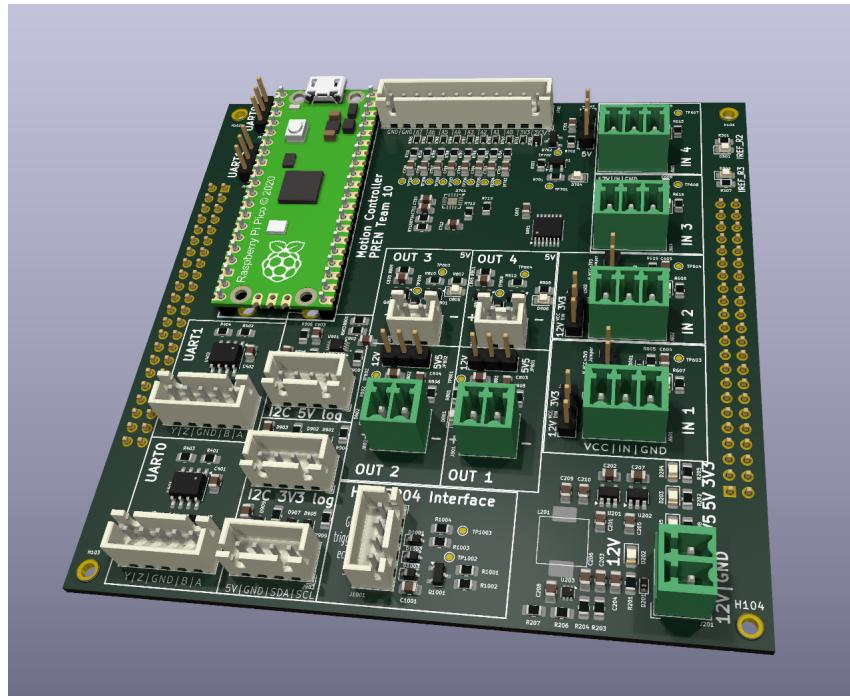


Abbildung 19: 3D-Ansicht Motion Controller

Nach dem derzeitigen Stand der Konzeption von PREN1 wäre dieser Controller prinzipiell in der Lage, auch die Aufgabe des Greifers zu übernehmen. Es ist jedoch noch nicht abschliessend getestet, inwieweit der Greifer in seinem jetzigen Zustand implementiert werden kann und ob noch weitere Ein-/Ausgänge benötigt werden. Daher ist für das Folgemodul ein eigener Controller für die Greifeinheit vorgesehen, der eine strombegrenzte Spannungsversorgung für die Servomotoren bietet. Der MotionController besitzt dagegen nur Lastschalter, deren Stromausgang nicht begrenzt ist.

Software

Im Nachfolgemodul ist die Entwicklung der Firmware des MotionControllers in C++ auf Basis des FreeRTOS-Kernels geplant.

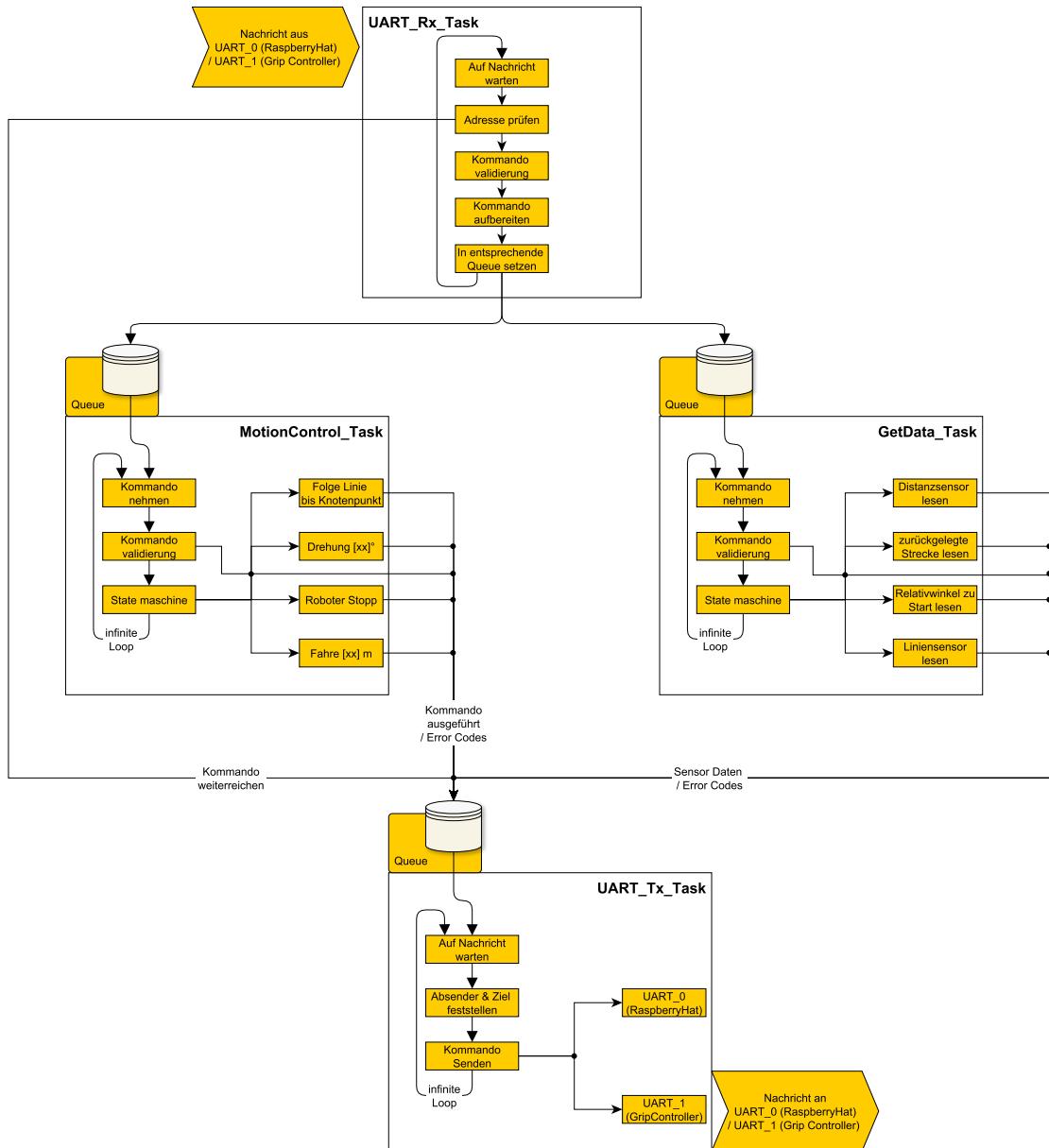


Abbildung 20: Steuerablauf Motion Controller Konzept

Abbildung 20 zeigt schematisch, wie die Firmware des Motion Controllers zu implementieren ist. Es wird einen *UART_RX*- und einen *UART_TX*-Task geben, die nur auf eingehende und ausgehende Signale warten. Dadurch wird verhindert, dass mehrere Tasks gleichzeitig auf dieselbe UART-Schnittstelle zugreifen. Wenn der Befehl für die Steuerung bestimmt und validiert ist, wird er in eine Queue eingereiht. Die zugehörige Task arbeitet dann in Form einer Zustandsmaschine die eingehenden Befehle in der je eigenen Queue ab. Durch diese Struktur ist die Firmware sehr modular aufgebaut, was spätere Anpassungen oder Erweiterungen wesentlich vereinfacht.

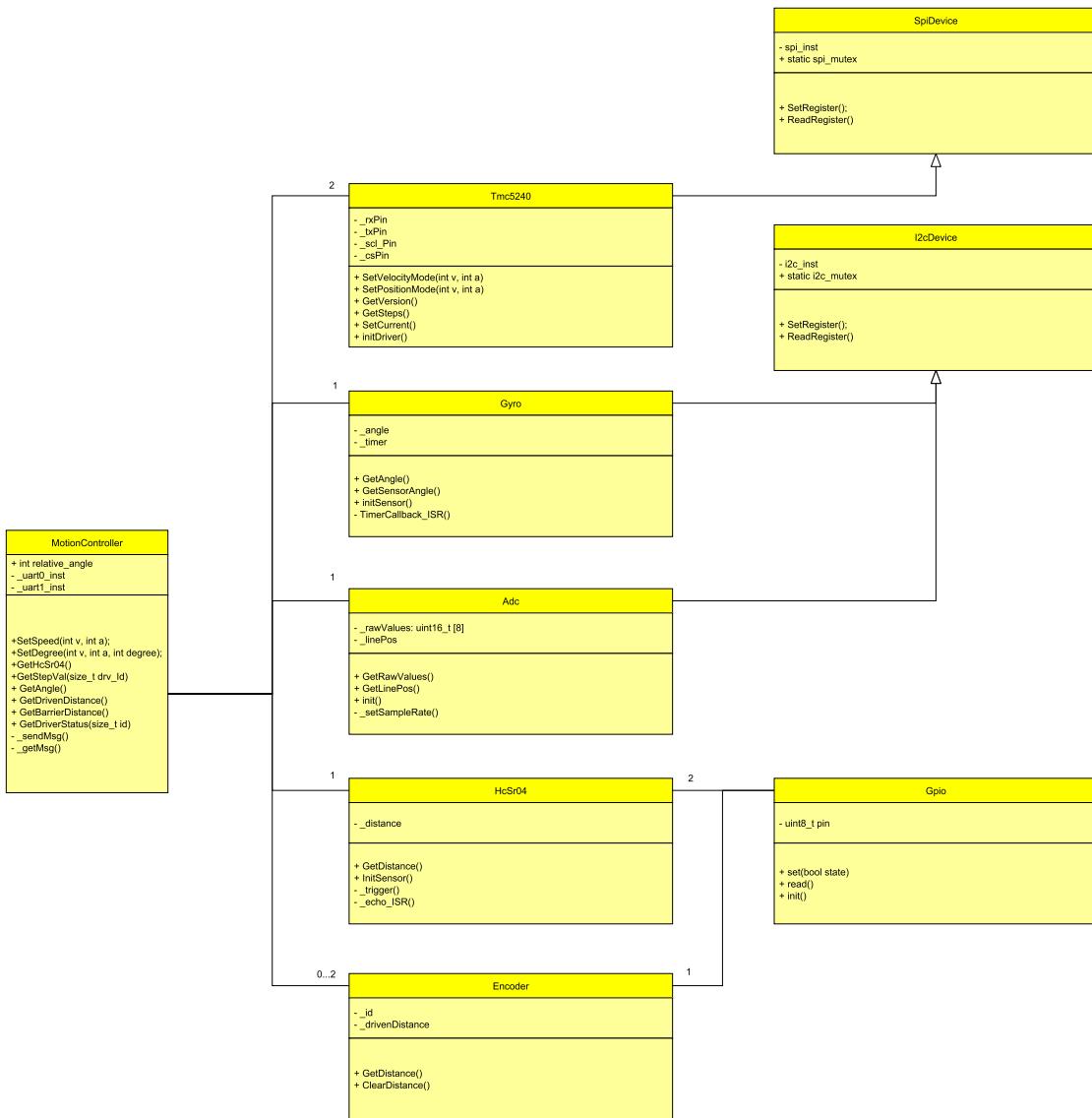


Abbildung 21: Klassendiagramm Konzept Motion Controller

In Abbildung 21 ist konzeptionell dargestellt, wie die noch zu erstellenden Klassen und Objekte zueinander in Beziehung stehen werden.

3.2.4. Liniensensor

In der vorliegenden Abhandlung wird die Entwicklung und Evaluierung eines Liniensensors dargelegt. Das Ziel besteht in der Konzeption eines Sensors, der eine unkomplizierte Auswertung ermöglicht und dazu geeignet ist, das vorgegebene Klebeband (*Tesa Gewebeband 4651*) zuverlässig von der Wettkampfbahn zu differenzieren. Die Entwicklung eines eigenständigen Liniensensors wird vorangetrieben durch das Bewusstsein, dass das Verlassen der Strecke ein signifikantes Risiko birgt. Die Eigenentwicklung eines Liniensensors ermöglicht eine hohe Flexibilität, da alle Komponenten selbst ausgewählt werden können.

Anforderungen

Das Klebeband soll auf einem rötlich gefliesten Untergrund detektiert werden. Eine besondere Herausforderung stellen dabei die gleichfarbigen Längs- und Querfugen der Fliesen dar. Dieser Untergrund ist in Abbildung 22 dargestellt.



Abbildung 22: Untergrund während des Wettkampfs

Aufbau und Auswertung

Der Liniensor bestehst aus acht einzelnen Messzellen. Diese Anzahl wurde so gewählt, dass ein möglichst grosser Bereich vom Sensor abgedeckt wird, ohne dass zu viele Pins für die Auswertung benötigt werden. Es ist sicherzustellen, dass stets genau zwei Messzellen direkt über dem Klebeband ausgerichtet sind. Jede Messzelle besteht aus einer Leuchtdiode und einem zugehörigen Fototransistor. Je mehr Licht auf den Fototransistor fällt, desto höher ist der Strom, der durch ihn fliesst. Da der Fototransistor als Konstantstromquelle interpretiert werden kann, wird der Spannungspegel zwischen Widerstand und Transistor ausgewertet.

Ein hoher Strom durch den Fototransistor bewirkt, dass der Transistor versucht, mehr Strom zu liefern, als ihm der Widerstand erlaubt. Dies führt dazu, dass er in Sättigung gerät und das Potential stark gegen GND gezogen wird. Ist die Reflexion des Lichts und damit der Strom hingegen gering, lässt der Fototransistor weniger Strom durch, als der Widerstand zulassen würde. In diesem Fall bewegt sich der Spannungspegel in Richtung 3,3V. Die Spannungen werden anschliessend mit Analog-Digital-Wandlern (ADCs) ausgewertet. Abbildung 23 zeigt die schematische Darstellung der Auswertung.

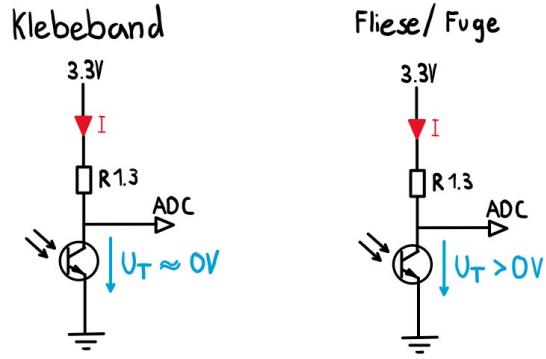


Abbildung 23: Konzept der Auswertung mittels ADC

3.2.5. Liniensensor als PCB

Damit der Liniensensor möglichst praxisnah getestet werden kann, wird er als PCB mithilfe von KiCad entworfen. Dabei werden die oben genannten Anforderungen und Dimensionierungen eingehalten. In Abbildung 24 ist die Draufsicht, und in Abbildung 25 die Untersicht des PCBs dargestellt.

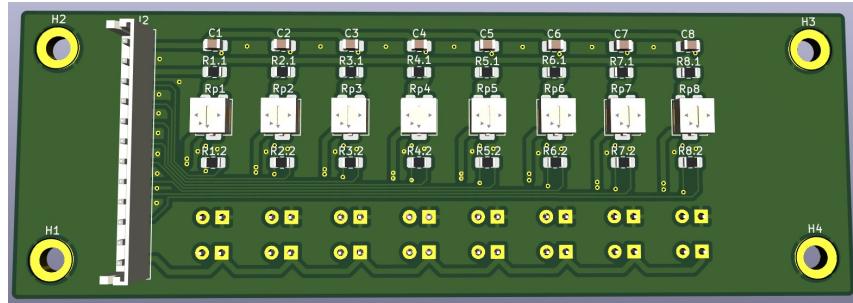


Abbildung 24: Liniensensor in Kicad von oben

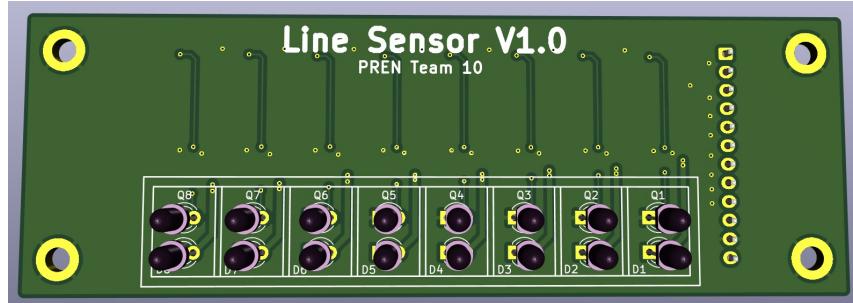


Abbildung 25: Liniensensor in Kicad von unten

Versuche Im Anhang ist ein Versuch dokumentiert, der ein geeignetes Lichtspektrum für den Liniensensor identifiziert. Zusätzlich wurden die Ströme auf den unterschiedlichen Untergründen (Klebeband, Fliese und Fuge) gemessen. Mithilfe eines Arduino wurden alle

Eingänge ausgewertet und der Liniensor validiert. Die vollständigen Messungen sind im Anhang G detailliert dargestellt.

Entscheidung und Fazit Durch die Messungen konnte das Konzept des Liniensors erfolgreich überprüft und validiert werden. Das folgende Bild bestätigt, dass das Klebeband deutlich von der Fliese unterschieden werden kann. In Abbildung 26 ist die Spannungs- auswertung mittels ADCs dargestellt. Diese Auswertung wurde mithilfe eines Arduino durchgeführt. Die hohen Spannungswerte repräsentieren die Fliese, während die tiefen Werte das Klebeband anzeigen.

```
A0: 594 A1: 569 A2: 519 A3: 67 A4: 76 A5: 613 A6: 515 A7: 642
A0: 601 A1: 569 A2: 429 A3: 69 A4: 218 A5: 519 A6: 487 A7: 735
A0: 601 A1: 586 A2: 415 A3: 69 A4: 242 A5: 511 A6: 567 A7: 766
A0: 584 A1: 553 A2: 448 A3: 68 A4: 211 A5: 504 A6: 517 A7: 711
A0: 601 A1: 591 A2: 442 A3: 69 A4: 240 A5: 524 A6: 530 A7: 705
A0: 653 A1: 635 A2: 491 A3: 72 A4: 250 A5: 570 A6: 612 A7: 805
```

Abbildung 26: Die tiefen Werte stellen das Klebeband dar und die hohen die Fliesen

In Abbildung 26 ist erkennbar, dass sich die Zelle A3 direkt über dem Klebeband befindet. Auch die Pins A2 und A4 liegen teilweise über dem Klebeband. Der gemessene Wert ist jedoch höher, da nur ein Teil des Klebebandes direkt darunter positioniert ist.

In der Risikobewertung wurde die Unterscheidung zwischen Klebeband und Fugen als potenzielles Risiko identifiziert. Der entwickelte Liniensor ist jedoch in der Lage, Fugen zuverlässig von Klebebändern zu unterscheiden. Diese Eigenschaft ist im Anhang G ausführlich dokumentiert. Aufgrund der klaren und reproduzierbaren Messwerte wird dieser Liniensor eingesetzt.

3.2.6. Abstandssensoren

Wie die Konzeption der Abstandssensoren (siehe Anhang E) gezeigt hat, wird die Umfelderkennung im ersten Prototypen redundant ausgeführt und im Folgemodul weiter verfeinert.

Das in PREN1 erarbeitete Konzept sieht vor, den Abstand zu potentiellen Pylonen mittels LIDAR über die Hindernisse hinweg zu erfassen. Sind diese näher als 2 m - ist die betrachtete Strecke wahrscheinlich nicht befahrbar. Ein zweiter Abstandssensor, der mit Ultraschall arbeitet, erkennt Hindernisse, die näher als 0,5 m sind. Dadurch kann das Fahrzeug rechtzeitig abgebremst werden. Wird die am Greifer positionierte Lichtschranke ausgelöst, kann mit Sicherheit davon ausgegangen werden, dass das Fahrzeug nun beginnen kann und soll, das Hindernis zu greifen und umzupositionieren.

Die Funktion der Lichtschranke könnte auch vom Ultraschallsensor übernommen werden, was im Folgemodul weiter untersucht wird. Ebenso befindet sich auf dem Fahrzeug eine Kamera, die die vorhandenen Wegpunkte analysieren kann. Mit dieser Kamera könnten auch Hindernisse erkannt werden, dies wird aber ebenfalls erst im Folgemodul weiter untersucht, wenn ein erster Prototyp aufgebaut ist.

Fazit Das in Abbildung 27 gezeigte Konzept stellt den aktuellen Stand der Entwicklung dar und ist so für den ersten Aufbau eingeplant.

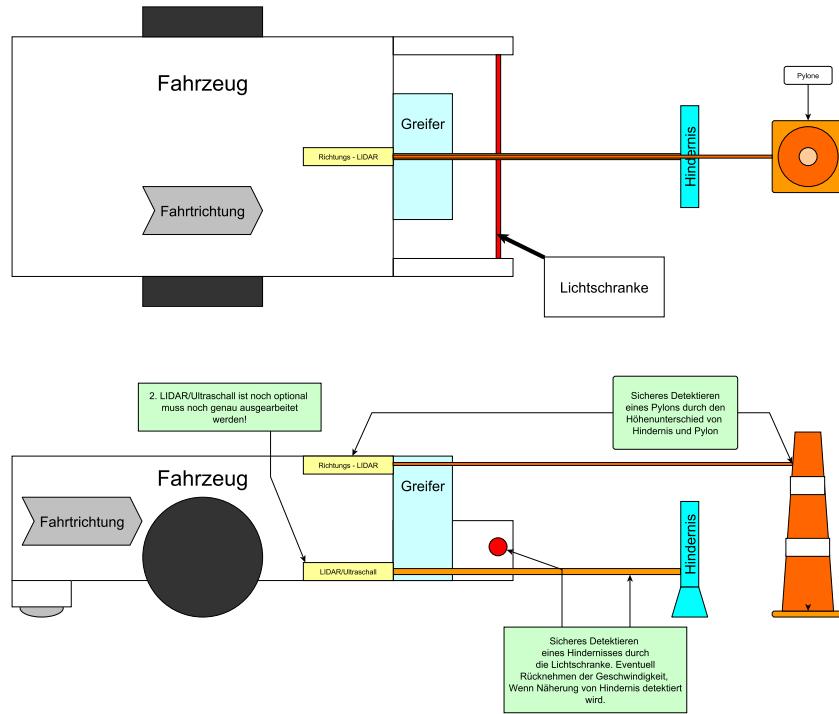


Abbildung 27: Konzept Hinderniserkennung

Eingesetzte Sensoren Verwendet wird der Ultraschallsensor *HC-SR04*, sowie der Richtungs-LIDAR *TFLuna*.

3.2.7. Sensorik Strecken Rückverfolgung

Der Algorithmus setzt voraus, dass der Roboter zu jedem Zeitpunkt seine Orientierung als absoluten Winkel vom Startpunkt kennt und die zurückgelegte Strecke messen kann. In diesem Abschnitt wird beschrieben, wie der Roboter diese Informationen erhält. Abbildung 28 zeigt schematisch, wie diese Funktion implementiert ist. In der folgenden Beschreibung wird dieses Schema noch einmal erläutert.

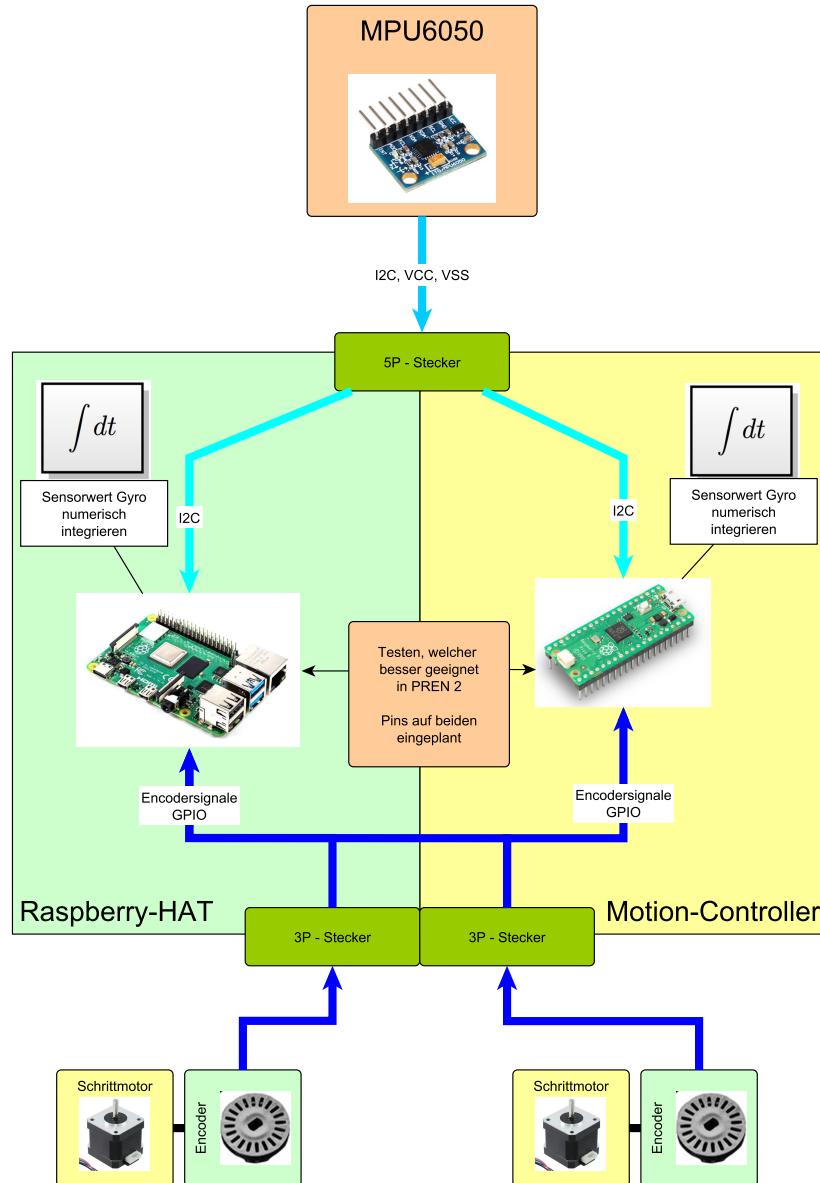


Abbildung 28: Blockschaltbild Sensorik Streckenrückverfolgung

Winkelerfassung Der Winkel des Fahrzeugs wird, wie in Anhang F beschrieben, mit einem Gyroskop gemessen. Dieses erfasst zu jedem Zeitpunkt die Änderungsrate des aktuellen Winkels und integriert diese alle $25\mu s$ numerisch. Dies ermöglicht eine sehr genaue Aussage über die aktuelle Orientierung des Roboters.

Mangels Echtzeitfähigkeit ist nicht sicher, ob der Raspberry Pi in der Lage ist, das Gyroskop ausreichend genau und häufig auszulesen. Daher ist sowohl auf dem Antriebscontroller als auch auf dem Raspberry-HAT ein entsprechender Steckplatz vorgesehen, um dies im Nachfolgemodul PREN2 zu testen.

Fahrstrecke Wie in der Konzeption beschrieben, wird die zurückgelegte Wegstrecke des Roboters mit Hilfe eines Encoders erfasst. Die verwendete Lochscheibe hat 36 Löcher, was einer Auflösung von 10 Grad pro Schritt entspricht. Die Abmessungen der Löcher

wurden anhand der Herstellerangaben aus dem entsprechenden Datenblatt festgelegt. Laut Hersteller beträgt die minimale Erfassungsgröße $1,8 \times 0,8$ mm, weshalb Löcher mit den Abmessungen $1,8 \times 1,6$ mm gewählt wurden (siehe Abbildung 29).

Sollte sich während der Tests in PREN 2 herausstellen, dass die Auflösung nicht ausreicht, könnte der Durchmesser der Lochscheibe vergrößert werden, um bei gleicher Lochgröße mehr Löcher pro Umdrehung unterzubringen.

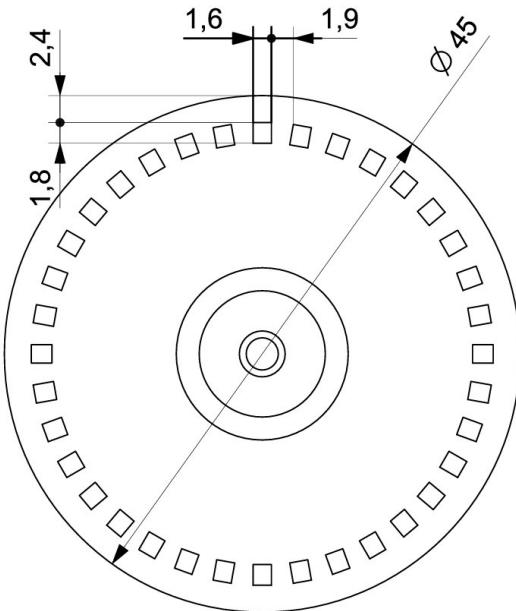


Abbildung 29: Masse der Lochscheibe

In einem zweiten Versuch soll im Folgemodul PREN2 getestet werden, ob es ausreicht, für diese Referenz die vom Schrittmotortreiber zurückgelegten Schritte zu verwenden. Dieser Versuch kann jedoch erst nach dem Bau eines ersten Roboters mit ausreichender Aussagekraft durchgeführt werden.

3.2.8. Grip Controller

Wie im Abschnitt 3.2.1 erwähnt, steuert der Grip Controller sämtliche Funktionen im Zusammenhang mit dem Greifen und dem Umgang mit Hindernissen. Die selbstentwickelte Leiterplatte erfüllt folgende Aufgaben:

Digitale Ein- & Ausgänge Die Leiterplatte ist mit digitalen Eingangs- und Ausgangsfunktionalitäten ausgestattet. Es existieren mehrere digitale Eingänge, die eine flexible Konfiguration zwischen Spannungsspeichern von 3,3 V oder 12 V ermöglichen. Die entprellten Eingänge dienen der Detektion des Anschlags des Greifers mittels Endschaltern oder Lichtschranken. Des Weiteren sind zwei Anschlüsse für Servos vorhanden. Diese Ausgänge sind mit einer Strombegrenzung belegt, welche bei einem maximalen Strom von 1,5 A begrenzt wird. Beim Erreichen der Strombegrenzung wird dies dem Mikrocontroller mitgeteilt. Zusätzlich sind zwei flexible Ausgänge integriert, die eine Einstellung zwischen den Spannungsspeichern von 3,3 V oder 12 V ermöglichen. Einer der beiden Ausgänge ist für die Detektion von Hindernissen mittels einer Lichtschranke vorgesehen.

Schnittstelle Kommunikation Die Kommunikation wird durch den Einsatz von UART realisiert. Die differenzielle Kommunikation zwischen dem Grip Controller und dem Motion Controller erfolgt direkt über Steuerbefehle. Bei möglichen Fehlercodes steht der Grip Controller via Motion Controller mit dem RaspberryHAT-PCB in Kontakt.

Spannungsversorgung Der Grip Controller transformiert die Versorgungsspannung mittels eines 12V-DC/DC-Wandlers auf einen Spannungswert von 5,5 V. Durch den Einsatz von zwei LDOs werden gefilterte Spannungspiegel von 3,3 V und 5 V bereitgestellt.

Sonstige Besonderheiten Bei der Konzeption dieser Leiterplatte wurde bewusst eine grosszügige Reserve an Anschlüssen eingeplant, um auf unvorhergesehene Änderungen dennoch genügen Anschlüsse zu bieten. Darauf hinaus ist die Leiterplatte mit LEDs ausgestattet, die im Falle einer Fehlersuche den Betriebsstatus der jeweiligen Komponenten-Gruppen anzeigen.

Die Abbildung 30 zeigt die 3D-Ansicht des Grip Controllers.

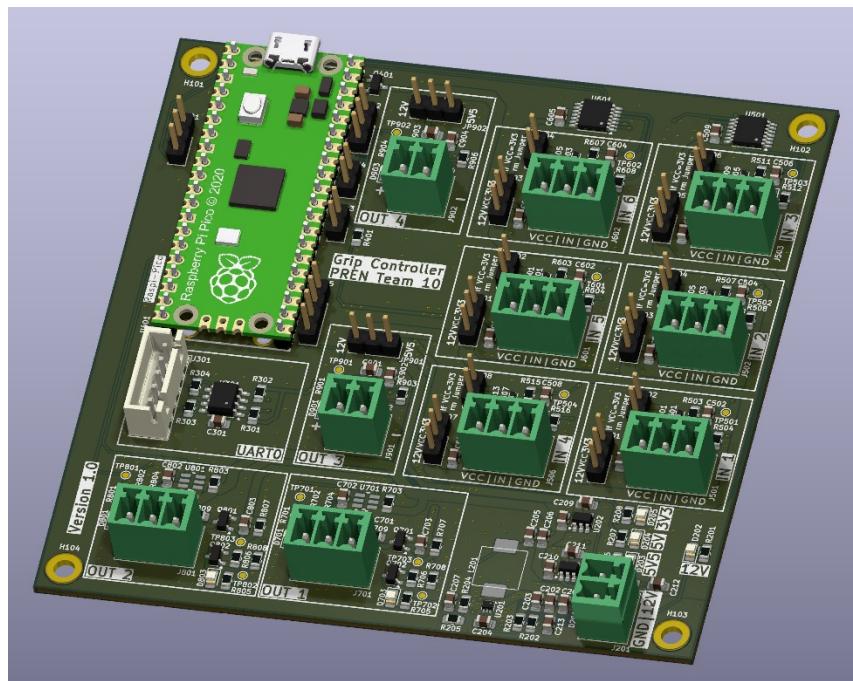


Abbildung 30: 3D-Ansicht Grip Controller

Bei der Entwicklung dieser Leiterplatte wurde auf Teilbaugruppen des Motion Controllers zurückgegriffen. Alle relevanten Designfiles sind im digitalen Anhang verfügbar.

3.2.9. Schnittstelle Raspberry Pi

Wie Eingangs im Kapitel 3.2.1 erwähnt, wird die Steuerungseinheit, der Raspberry Pi, mit einer eigens entwickelten Leiterplatte in das Gesamtsystem eingebunden. Diese erfüllt die folgenden Aufgaben

Digitale Ein- & Ausgänge Im Hinblick auf die digitale Ein- und Ausgangs-Funktionalität der Leiterplatte ist festzustellen, dass diese über vier digitale Eingänge verfügt, die

auf einem Spannungspegel von 12 V arbeiten. Zwei digitale Ausgänge sind in der Lage, auch grössere Lasten bis zu 3 A zu schalten, wobei hier auf den Energiebedarf des Gesamtsystems geachtet werden muss.

Schnittstellen Schalter Im Hinblick auf die Schnittstellen ist festzuhalten, dass sowohl ein Wahlschalter als auch ein Starttaster mit dem Raspberry Pi verbunden werden können. Der Starttaster wird mit einem Komparator und einem RC-Filter bereits in Hardware entprellt.

Schnittstelle Kommunikation Die Schnittstelle zur Kommunikation wird durch den UART-Kanal des Raspberry Pis mit einer RS422-Schnittstelle zur Kommunikation mit dem Gesamtsystem aufbereitet.

Sensorschnittstellen Darüber hinaus können über die I^2C -Schnittstellen einfache Sensoren angeschlossen werden. Es sind sowohl Schnittstellen für 5V und 3.3V Logikpegel vorhanden, was die Leiterplatte sehr vielseitig einsetzbar macht.

Spannungsversorgung Im Hinblick auf die Spannungsversorgung wird die 12-Volt-Versorgungsspannung mit einem 5-Volt-3-Ampere-DCDC-Steller und LDO für den Raspberry Pi aufbereitet und gefiltert.

Sonstige Besonderheiten Darüber hinaus bietet die RaspberryHAT-Leiterplatte einige optische Anzeigeelemente, die zu Debugzwecken einfache digitale Signale optisch darstellen können. Daneben ist das Board mit einem Piezo Summer ausgestattet, welcher die akustische Signalisierung von Betriebszuständen, wie z.B. das Erreichen eines Zielpunkts zulässt.

Die Abbildungen 31 und 32 zeigen das RaspberryHAT - PCB in einer Draufsicht und in einer 3D-Ansicht. In Abbildung ist ersichtlich, wie der Raspberry Pi an dieses Board angeschlossen wird.

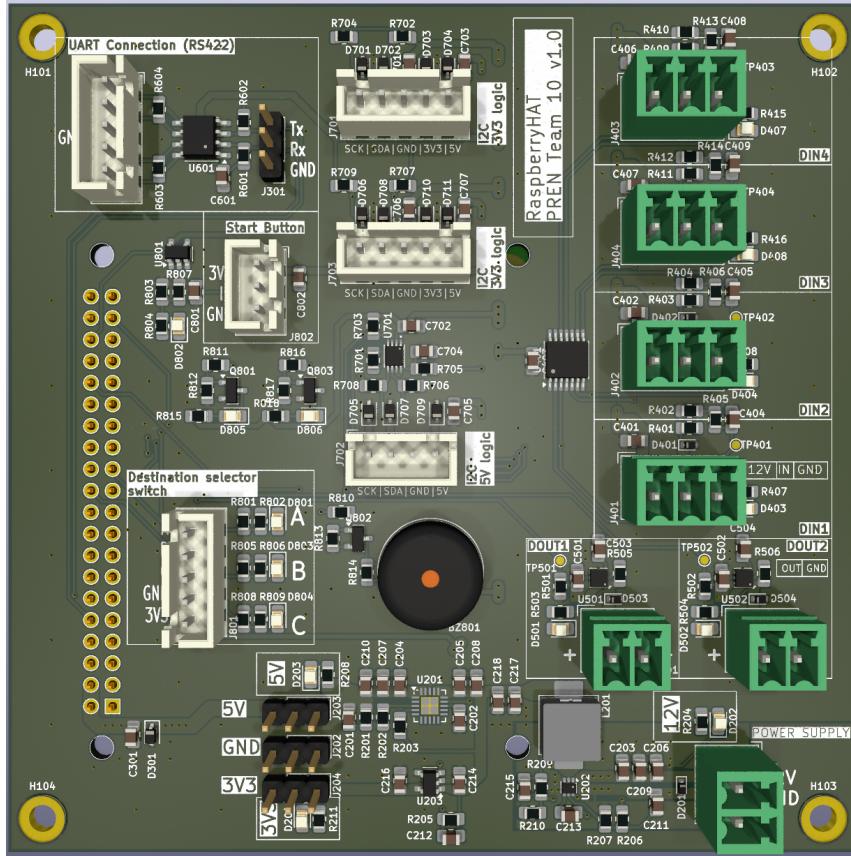


Abbildung 31: RaspberryHAT in Draufsicht

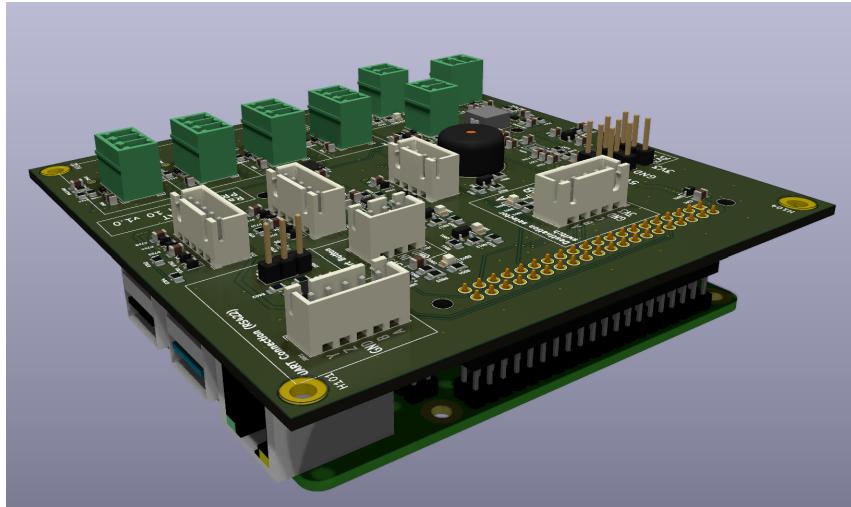


Abbildung 32: RaspberryHAT in 3D Ansicht

Bei der Entwicklung dieser Leiterplatte ist zum grössten Teil auf Teilbaugruppen des Motioncontrollers sowie des Gripcontrollers zurückgegriffen worden. Aus Gründen der Übersichtlichkeit ist auf eine separate Schaltungsdimensionierung und -beschreibung verzichtet. Die entsprechenden Designfiles befinden sich nichtsdestotrotz im digitalen Anhang.

3.3. Informatik

3.3.1. Beschreibung & Entscheid Algorithmen

Zu Beginn der Entwicklung des Simulators wurden verschiedene Algorithmen evaluiert, wobei der Fokus zunächst auf „Shortest Path“-Algorithmen lag. Hierfür wurden die Algorithmen **Dijkstra**, **A*** und **D*Lite** implementiert und vollständig in den Simulator integriert. Diese Algorithmen eignen sich grundsätzlich hervorragend für die Berechnung kürzester Wege in statischen Graphen.

Schwierigkeiten bei der Anwendung von Shortest Path-Algorithmen Im Laufe der Entwicklung zeigte sich, dass die Anwendung von klassischen Shortest Path-Algorithmen wie Dijkstra und A* in unserem Szenario besondere Herausforderungen mit sich bringt. Obwohl der Graph und seine Einschränkungen (z. B. blockierte Knoten oder entfernte Strecken) einmal zu Beginn definiert werden, ist die tatsächliche Navigation problematisch, da der Roboter den Graphen erst schrittweise entdecken muss. Die fehlende Kenntnis über Teile des Graphen führt zu spezifischen Herausforderungen:

- **Sackgassen und Umwege:** Der Roboter kann in Sackgassen gelangen, aus denen er sich befreien muss. Dijkstra und A* setzen voraus, dass der gesamte Graph bekannt ist, was bei einem schrittweisen Entdecken nicht der Fall ist. Darüber hinaus können diese Algorithmen keine Gewichte dynamisch anpassen oder neue Kanten und Knoten während der Navigation integrieren. Sie sind daher ungeeignet für Szenarien, in denen Informationen über den Graphen schrittweise gesammelt und verarbeitet werden müssen.
- **Blockierte Knoten hinter freien Knoten:** Blockierte Knoten (z. B. durch Pylonen) können sich direkt hinter freien Knoten befinden. Ohne Kenntnis dieser Blockade im Vorfeld könnten klassische Algorithmen falsche Entscheidungen treffen, da sie nicht auf unbekannte Hindernisse reagieren können.
- **Unvollständige Anfangskenntnis des Graphen:** Algorithmen wie Dijkstra und A* benötigen eine vollständige Kenntnis aller Knoten und Kanten sowie deren Gewichte. In unserem Szenario wird der Graph jedoch erst nach und nach durch den Roboter erkundet, was die Anwendung dieser Algorithmen ausschließt.

Beurteilung der Algorithmen: Dijkstra und A* sind für vollständig bekannte Graphen optimiert und daher nicht auf Szenarien ausgelegt, bei denen ein Graph schrittweise entdeckt wird. D*-Lite wäre theoretisch besser geeignet, da dynamische Anpassungen möglich sind. Allerdings setzt D*-Lite voraus, dass Kanten mit dynamischen Gewichten bewertet werden können und dass Hindernisse oder Schranken sofort erkannt und in die Berechnung einbezogen werden. Da unsere Sensoren die Entfernung zu Knoten nicht erfassen und Schranken initial ignoriert werden, können keine aussagekräftigen Gewichte gesetzt oder dynamisch angepasst werden. Aufgrund dieser Einschränkungen und der spezifischen Anforderungen unseres Projekts haben wir uns für die Entwicklung eines massgeschneiderten Algorithmus entschieden.

Entscheidung für einen neuen Algorithmus Da das Projekt bestimmten Rahmenbedingungen unterliegt, die weitgehend bekannt sind, und da die genannten Einschränkungen berücksichtigt werden müssen, entschied sich die Gruppe, einen **massgeschneiderten**

und pragmatischen Algorithmus zu entwickeln. Dieser Ansatz ermöglicht es, die spezifischen Herausforderungen des Problems effizient zu adressieren und gleichzeitig die vorhandenen Ressourcen optimal zu nutzen.

Obwohl der **D*Lite**-Algorithmus aufgrund seiner Fähigkeit, dynamische Graphen zu verarbeiten, länger in Betracht gezogen wurde, stellte sich heraus, dass auch dieser Algorithmus nicht vollständig auf die spezifischen Anforderungen unserer Problemstellung abgestimmt ist. Daher wurde ein neuer Algorithmus entwickelt, der speziell auf die Herausforderungen des Simulators zugeschnitten ist.

High-Level-Analyse des Algorithmus Der entwickelte Algorithmus basiert auf einer Kombination aus **Depth First Search (DFS)** und einer **heuristikbasierten Optimierung**. Dieser Ansatz erlaubt es dem Roboter, in einem vorab definierten, aber nur teilweise bekannten Graphen effizient zu navigieren, auch wenn Sackgassen oder andere Hindernisse auftauchen.

Funktionsweise

1. **Initialisierung:** Der Algorithmus startet am definierten Startknoten und initialisiert wichtige Datenstrukturen wie den Knoten- und Kantenstatus sowie die besuchten Knoten und Kanten. Der `vectorToGoal` wird berechnet, um eine Richtung zum Ziel zu definieren. Dieser Vektor ist entscheidend für die Heuristiken.
2. **Iterative Knotenexploration:** Mit einem *Stack* wird die Navigation durchgeführt, wobei Knoten nacheinander besucht und von möglichen Kanten aus weitergeführt werden. Gesperrte oder bereits besuchte Knoten und Kanten werden übersprungen.
3. **Traversal der Kanten:** Jede Kante wird vor dem Überqueren markiert. Falls eine Kante zu einem Pylon führt, wird sie als unpassierbar markiert.
4. **Heuristik-basierte Navigation:** Der Graph wird in drei verschiedene *Sektionen* (Bereich des Graphen) geteilt: Links (Zielknoten A), Mitte (Zielknoten B) und Rechts (Zielknoten C). Der Algorithmus geht aus einer Kombination aus *Richtung* (zum Zielvektor) und *Sektion* in die Tiefe. Basierend auf diesen Heuristiken werden den Kanten Gewichte zugewiesen und anhand dieser Gewichte, die beste Kante ausgewählt. Der Algorithmus priorisiert Kanten, die direkt auf das Ziel hinweisen, und bestraft Bewegungen in entgegengesetzte Richtungen. Zusätzlich versucht der Algorithmus wege in der korrekten *Section* zu nehmen. Wenn beispielsweise A der Zielknoten ist, versucht der Algorithmus möglichst Kanten zu wählen, welche in die vermutete Richtung von Knoten A zeigen und zusätzlich möglichst links zu halten.
5. **Zielerreichung und Backtracking:** Sobald das Ziel erreicht ist, wird der Prozess abgebrochen. Falls ein Knoten keine weiteren möglichen Kanten bietet, erfolgt ein Backtracking, um alternative Wege zu erkunden.

Fazit Der entwickelte Algorithmus kombiniert die Flexibilität eines DFS mit Heuristiken, um vorab definierte, aber nur teilweise bekannte Graphen effizient zu navigieren. Dieser pragmatische Ansatz erlaubt es dem Roboter, dynamisch auf Hindernisse wie Sackgassen oder blockierte Knoten zu reagieren und sich schrittweise dem Ziel zu nähern. Durch die spezifische Anpassung des Algorithmus an die Anforderungen des Projekts und dem *Proof of Concept* im Simulator erscheint diese Lösung robust und effektiv.

3.3.2. Sensorik und Anforderungen zur Algorithmusintegration

Der beschriebene Algorithmus zur Navigation und Exploration eines unbekannten Graphen benötigt spezifische Schnittstellen und Sensoren, um seine Funktionalität sicherzustellen. Diese Elemente bilden die Grundlage für die präzise Erfassung der Umgebung und die Entscheidungsfindung des Roboters.

Benötigte Schnittstellen und Sensorik Um den Algorithmus korrekt auszuführen, werden folgende Sensoren und Schnittstellen benötigt:

- **Gyroskop:** Das Gyroskop liefert Feedback zur Ausrichtung des Roboters und ermöglicht es, die Bewegungsrichtung präzise zu bestimmen. Dies ist essenziell, um die interne Karte zu zeichnen und die Orientierung innerhalb des Graphen zu bewahren.
- **Motor-Feedback:** Die Informationen aus den Motoren, wie die zurückgelegte Distanz, sind entscheidend, um die Länge der Kanten im Graphen zu kennen. Dadurch kann der Algorithmus die relative Position von Knoten und Kanten in der internen Karte bestimmen.
- **Lidar:** Der Lidar-Scanner dient der frühzeitigen Erkennung von Hindernissen wie Pylonen. Diese Informationen werden verwendet, um Kanten im Graphen als unpassierbar zu markieren, bevor der Roboter diese streift. Dies erhöht die Effizienz und Sicherheit bei der Navigation.

Durch die Kombination dieser Sensoren kann der Roboter eine dynamisch aktualisierte Karte erstellen, die sowohl besuchte als auch unbesuchte Knoten und Kanten enthält.

Umgang mit Schranken Der Algorithmus behandelt Strecken mit Schranken vorerst bewusst wie reguläre Strecken, ohne sie bei der Pfadauswahl stärker zu gewichten. Dies reduziert die Komplexität bei der Routenplanung. Je nach Geschwindigkeit, mit der Hindernisse umplatziert werden können, könnte dieser Ansatz jedoch reevaluiert werden, um Schranken in Zukunft in die Gewichtung einzubeziehen.

Risiken und Messtoleranzen Trotz der robusten Integration von Sensoren gibt es spezifische Risiken und Herausforderungen, die berücksichtigt werden müssen:

- **Messtoleranzen:** Abweichungen bei den Sensorwerten können zu Fehlern in der internen Karte führen. Beispielsweise können:
 - **Schlupf an den Rädern** die zurückgelegte Distanz falsch messen (Motor-Feedback).
 - **Drift im Gyroskop** zu Abweichungen in der Orientierung führen.

Lösung: Im Algorithmus nicht mit exakten Werten, sondern Wertebereichen rechnen. Dies erlaubt Fehler in den Messungen.

- **Fehlinterpretation von Pylonen:** Temporäre Hindernisse, wie ein Bein eines Zuschauers, könnten vom Lidar fälschlicherweise als Pylonen erkannt werden. Dies würde dazu führen, dass Strecken oder Knoten unnötig als gesperrt markiert werden. **Lösung:** Zuschauer bitten Abstand zu nehmen und eine Verifizierung, bei der Hindernisse mehrfach von verschiedenen Seiten erkannt werden müssen, bevor sie dauerhaft in die Karte aufgenommen werden.

- **Sonderfall: Gesperrter Knoten hinter einem freien Knoten:** Wenn sich ein gesperrter Knoten direkt hinter einem freien Knoten befindet (innerhalb von 2 Metern), könnte der Algorithmus fälschlicherweise auch den freien Knoten als gesperrt markieren.

Lösung: Der Roboter kann den Knoten mit vermutetem Pylon aus einem anderen Winkel reevaluieren, um eine fehlerhafte Markierung zu vermeiden. Dies minimiert das Risiko und erhöht die Genauigkeit der internen Karte.

Fazit Die Kombination der Sensoren Gyroskop, Motor-Feedback und Lidar bildet die Grundlage für die präzise Navigation des Roboters. Trotz einiger Risiken, wie Messtoleranzen und Fehlinterpretationen, bietet der Algorithmus durch Reevaluation und rechnen mit Wertebereichen eine robuste Lösung, um ungenaue Messungen abzufangen. Diese Massnahmen reduzieren die genannten Risiken und erlauben eine zuverlässigere Navigation.

3.3.3. Simulation

Die Simulation dient als Werkzeug zur Validierung und Optimierung der Navigationsalgorithmen in einer virtuellen Umgebung. Sie ermöglicht es, das Verhalten des Roboters unter verschiedenen Szenarien zu testen, bevor physische Prototypen entwickelt werden. Der implementierte Simulator unterstützt Funktionen wie die Simulation von Hindernissen, gesperrten Wegpunkten und alternativen Routen.

Der vollständige Sourcecode des Simulators ist im zugehörigen GitHub-Repository abrufbar: <https://github.com/tramasys/simpren>

Konzeption Der Simulator wurde so konzipiert, dass er eine flexible und intuitive Plattform zur Analyse und Optimierung von Navigationsalgorithmen bereitstellt. Die Architektur basiert auf vier Hauptkomponenten, die über gemeinsam genutzte Datenstrukturen und Stores miteinander kommunizieren. Das folgende Diagramm verdeutlicht den Aufbau des Simulators und die Interaktion zwischen den Komponenten:

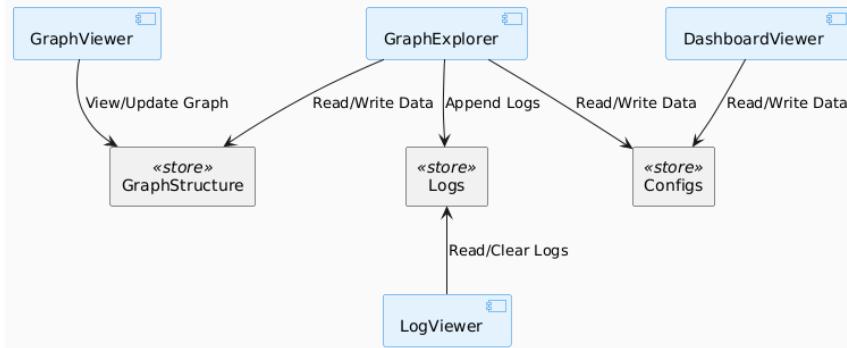


Abbildung 33: Aufbau des Simulators und dessen Komponenten

Die zentralen Komponenten sind:

- **GraphViewer:** Visualisiert den Graphen und ermöglicht die Aktualisierung der Knoten und Kanten im *GraphStructure*-Store. Nutzer:innen können über diese Komponente den aktuellen Zustand des Graphen einsehen und manuelle Änderungen vornehmen.
- **DashboardViewer:** Bietet eine Benutzeroberfläche zur Konfiguration der Simulation und interagiert mit dem *Configs*-Store, um Simulationsparameter zu lesen oder zu schreiben. Hier können Betriebsmodi ausgewählt und Simulationseinstellungen angepasst werden.
- **GraphExplorer:** Die zentrale Komponente des Simulators, die den Graphen analysiert, Pfade berechnet und Logs im *Logs*-Store speichert. Der *GraphExplorer* übernimmt die Hauptlogik der Simulation und entscheidet über die Navigationsstrategien.
- **LogViewer:** Ermöglicht die Ansicht und Analyse der im *Logs*-Store gespeicherten Simulationslogs. Nutzer:innen können die Entscheidungen und den Fortschritt des Algorithmus nachvollziehen.

Die Kommunikation zwischen den Komponenten erfolgt über drei zentrale Stores:

- **GraphStructure:** Speichert den Zustand des Graphen, inklusive der Knoten und Kanten, und wird von *GraphViewer* und *GraphExplorer* genutzt.

- **Logs:** Enthält alle Simulationslogs, die von *GraphExplorer* geschrieben und von *LogViewer* gelesen werden können. Diese Logs dokumentieren die Schritte und Entscheidungen während der Simulation.
- **Configs:** Speichert die Simulationsparameter, die von *DashboardViewer* konfiguriert und von *GraphExplorer* gelesen werden. Dazu gehören Einstellungen wie Betriebsmodi, Parameter für Algorithmen und benutzerdefinierte Anpassungen.

Allgemeine Funktionen Der entwickelte Simulator dient der Visualisierung, Analyse und Optimierung von Algorithmen zur Navigation eines Roboters in einem unbekannten Graphennetzwerk. Er wurde mit dem Ziel konzipiert, eine flexible und intuitive Plattform für die Simulation unterschiedlicher Szenarien zu bieten. Die Funktionen sind so ausgelegt, dass sowohl die interaktive Bedienung als auch automatisierte Simulationen und visuelle Darstellungen unterstützt werden.

Es wurden verschiedene Algorithmen im Simulator implementiert, wie Dijkstra, A* und D*Lite. Letztendlich hat sich das Informatikteam dafür entschieden, einen eigenen Algorithmus zu entwickeln. Im Laufe der Entwicklung wurde immer mehr klar, dass es durch das Wegnehmen von Strecken auf dem Graph zu einem Navigationsproblem kommen kann. Deswegen hat man sich dafür entschieden, nicht mit den Algorithmen Dijkstra und D*Lite weiterzufahren. Vollständigkeitshalber sind sie aber noch im Simulator enthalten.

Visualisierung und Steuerung Die Steuerung und Visualisierung des Graphen erfolgt über ein benutzerfreundliches Dashboard. Es stehen drei wesentliche Optionen zur Verfügung 34:

- **Randomized Graph:** Generiert einen neuen, zufällig aufgebauten Graphen, der als Grundlage für die Simulation dient.
- **Reset Graph:** Setzt den Graphen in seinen Ursprungszustand zurück, entfernt alle Barrieren, Pylonen und andere Modifikationen.
- **Reset State:** Stellt den Zustand aller Knoten und Kanten auf ihre Standardeinstellungen zurück. Unterschiedliche Status wie *probed* (gelb), *visited* (grün), *restricted* (rot) oder *finished* (blau) werden visuell hervorgehoben und erlauben eine präzise Nachverfolgung des Algorithmus.

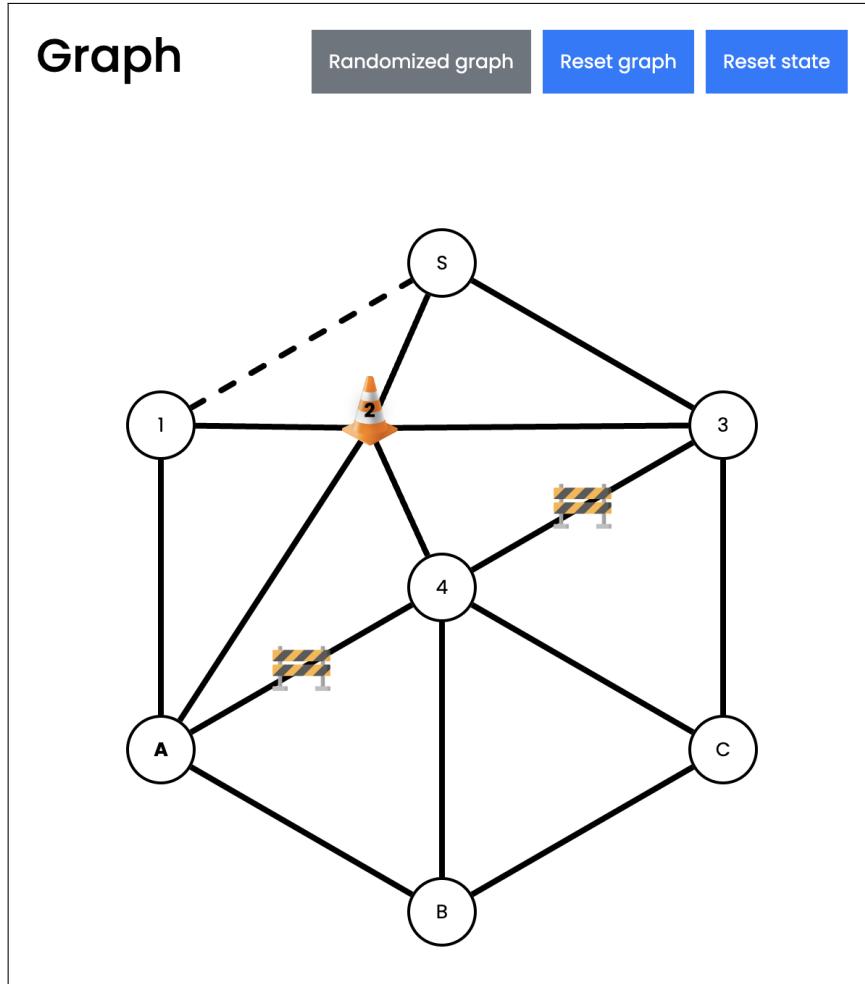


Abbildung 34: Screenshot des Bereichs für den Graph im Simulator (GUI)

Manuelle Bearbeitung von Knoten und Kanten Der Simulator erlaubt eine intuitive manuelle Bearbeitung von Knoten und Kanten im Graphen direkt über die Benutzeroberfläche. Dies ermöglicht es dem Nutzer, die Simulation individuell anzupassen und spezifische Szenarien zu testen 34:

- **Knoten bearbeiten:** Durch einfaches Klicken auf einen Knoten wird ein Pylon auf diesen gesetzt. Der Pylon repräsentiert ein Hindernis auf dem Knoten. Ein erneutes Klicken entfernt den Pylon wieder, wodurch der Knoten frei wird.
- **Kanten bearbeiten:** Kanten durchlaufen drei Zustände, die durch wiederholtes Klicken in einem zyklischen Ablauf geändert werden können:
 1. *Normale Strecke*: Die Kante ist ohne Hindernis befahrbar.
 2. *Entfernte Strecke*: Die Kante wird entfernt, was im UI durch eine gestrichelte Linie angezeigt wird.
 3. *Strecke mit Schranke*: Die Kante ist befahrbar, hat jedoch eine Schranke, die zusätzliche Fahrzeit oder eine andere Einschränkung symbolisiert.

Nach einem weiteren Klick kehrt die Kante wieder in den Ausgangszustand als normale Strecke zurück. Dieser Zyklus wird im UI visuell hervorgehoben, um den aktu-

ellen Zustand klar darzustellen.

Funktionen und Modus-Optionen Der Simulator bietet drei verschiedene Betriebsmodi. Diese werden durch Tabs voneinander graphisch getrennt 35:

- **Interactive run:** Dieser Modus erlaubt es Graph Algorithmen, wie Dijkstra, A* und D*Lite interaktiv zu testen, wobei spezifische Parameter durch den Benutzer angepasst werden können.
- **Parameterized run:** In diesem Modus können mehrere Durchläufe automatisiert durchgeführt werden. Es ist möglich, Parameter wie die Fahrzeit auf verschiedenen Strecken (normale Strecken vs. Strecken mit Barrieren), die Anzahl der zu simulierenden Durchläufe (z. B. 100 Runs) und andere Variablen zu spezifizieren. Der Simulator generiert dabei zufällige Graphen und löst diese mittels eines heuristikbasierten Tiefensuchalgorithmus (DFS).
- **Explore Map:** In diesem Modus wird der heuristikbasierte DFS-Algorithmus visuell dargestellt. Der Nutzer kann das Ziel pro Durchlauf selbst bestimmen, wodurch der Graph Schritt für Schritt aufgedeckt wird. Diese Visualisierung identifiziert auch Knoten, die durch mathematische Berechnungen erkannt werden, obwohl der Roboter diese nicht direkt besucht hat. Wenn eine neu entdeckte Kante eine bereits bekannte Kante schneidet, wird anhand linearer Algebra auf die Existenz eines Knotens geschlossen.

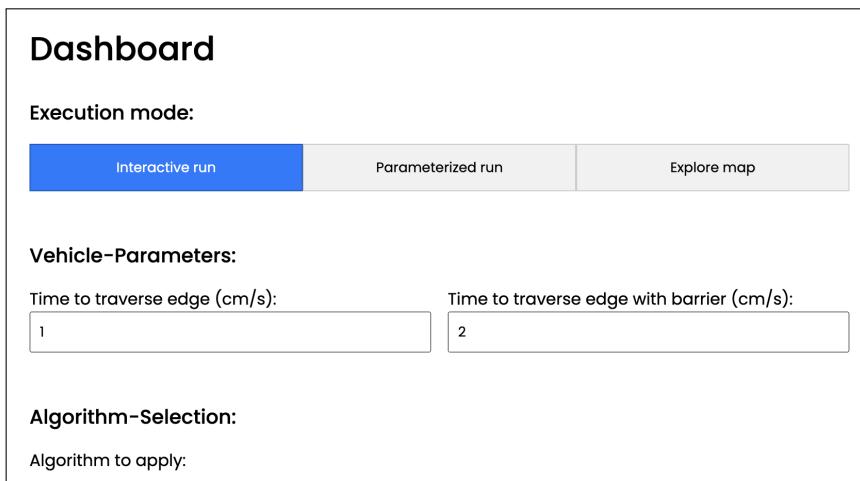


Abbildung 35: Screenshot des Bereichs für Betriebsmodi und Konfiguration im Simulator

Log-Analyse Der Simulator enthält ein integriertes Log-System, das die Entscheidungen des Algorithmus dokumentiert. Die Logs können direkt im Dashboard gelesen oder als CSV-Datei exportiert werden. Folgende Funktionen stehen zur Verfügung 36:

- **Clear Logs:** Setzt die Logs vollständig zurück.
- **Export Logs:** Exportiert die Logs als CSV-Datei.
- **Clear Screen:** Entfernt die Logs aus der Ansicht, ohne diese zu löschen.

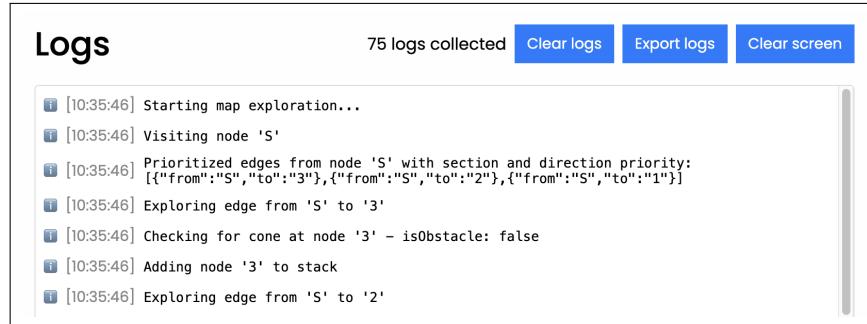


Abbildung 36: Screenshot des Bereichs für Logs im Simulator

Fazit Der Simulator bietet eine vielseitige Plattform für die Simulation und Analyse von Navigationsalgorithmen. Durch die Kombination aus visueller Darstellung, flexiblen Betriebsmodi und präzisem Logging ermöglicht er ein umfassendes Verständnis der Funktionsweise und Leistungsfähigkeit der eingesetzten Algorithmen.

3.3.4. Tests

Die Qualität und Funktionalität des Simulators wurden umfassend geprüft, wobei ein besonderer Schwerpunkt auf der zentralen Komponente, dem *GraphExplorer*, lag. Diese Komponente ist entscheidend für die Navigation und Entscheidungsfindung im Graphen und wurde mittels detaillierter Unit-Tests und Usertests validiert.

Unittests Die Unit-Tests für den *GraphExplorer* deckten die wichtigsten Kernfunktionen ab, um sicherzustellen, dass die Algorithmen korrekt implementiert sind und konsistent arbeiten. Dabei wurde das Framework *Vitest* verwendet, welches aufgrund seiner schnellen Performance und nahtlosen Integration mit Svelte ideal für die Testumgebung geeignet ist. Die wichtigsten getesteten Funktionen umfassen:

- **Berechnung von Vektoren und Ausrichtung:** Validierung der korrekten Berechnung des Zielvektors zwischen Knoten und der Ermittlung der Ausrichtung von Vektoren, um sicherzustellen, dass die Navigation priorisiert wird.
- **Klassifikation von Knoten:** Tests zur korrekten Einteilung von Knoten in die Sektionen *links*, *rechts*, *Mitte*.
- **Ermittlung möglicher Kanten:** Prüfung, ob die möglichen Verbindungen eines Knotens korrekt ermittelt und priorisiert werden.
- **Markierung besuchter Knoten und Kanten:** Überprüfung, ob Knoten und Kanten korrekt als besucht markiert werden, um redundante Wege zu vermeiden.
- **Intersektionserkennung:** Validierung der Berechnung von Schnittpunkten zwischen Kanten, um die Konsistenz des Graphen zu gewährleisten.
- **Verarbeitung eingeschränkter Knoten:** Sicherstellung, dass Knoten mit Einschränkungen (z. B. *restricted*) korrekt behandelt werden.

Die Tests wurden unter Verwendung von Mock-Daten durchgeführt, die feste Knoten und Kanten (*fixedNodes*, *fixedEdges*) im Graphen repräsentierten. Das Verhalten des Systems wurde mit den erwarteten Ergebnissen verglichen, um die Genauigkeit der Algorithmen sicherzustellen.

Usertests Zusätzlich zu den automatisierten Tests wurde der Simulator durch umfangreiche Usertests evaluiert. Hierfür wurden spezifische, „schwierige“ Graphen erstellt, die besondere Herausforderungen wie Sackgassen, blockierte Knoten oder entfernte Strecken beinhalteten.

Das Verhalten des Simulators wurde mit den theoretisch erwarteten Ergebnissen verglichen, um sicherzustellen, dass der Algorithmus auch in komplexen Szenarien zuverlässig arbeitet. Dabei lag der Fokus auf:

- der korrekten Navigation durch den Graphen,
- und der Einhaltung der Prioritäten bei der Pfadwahl.

Fazit Die Kombination aus automatisierten Unit-Tests und praxisnahen Usertests gewährleistete, dass der Simulator sowohl auf funktionaler Ebene als auch in der Anwendung robust ist. Die Tests trugen wesentlich dazu bei, die Zuverlässigkeit des Systems zu validieren und potenzielle Fehler frühzeitig zu identifizieren.

4. Projektmanagement

4.1. Organisation

Das Projektteam ist agil organisiert. Die Idee ist, dass jedes Teammitglied seine Verantwortung und die Zuständigkeit für eine bestimmte Teilfunktion des Gerätes innehat. Die entsprechenden Inkremeante der Entwicklung werden jeweils Freitags den anderen Teammitgliedern vorgestellt und in der offenen Runde diskutiert. Dabei wird sichergestellt, dass wichtige Weichen für Schnittstellen, beispielsweise mechanische Umsetzung mit elektrischer Ansteuerung, frühzeitig gestellt werden und bei auf kommenden Problemen auch früh reagiert werden kann. Darüber hinaus sind Ämter verteilt, wie die allgemeine Organisation, Mechanik-Werkstatt-Beauftragter sowie die Budgetplanung.

Abbildung 37 zeigt eben diese Struktur des Teams.

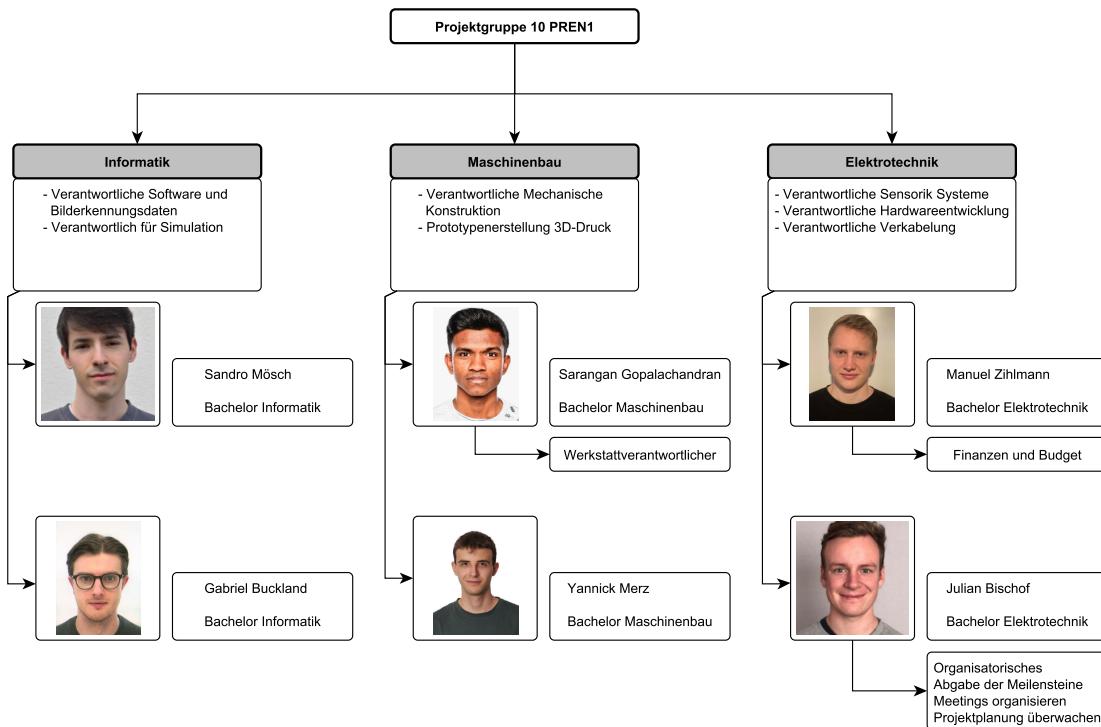


Abbildung 37: Organigramm Team 10

4.2. Projektplanung

Das Projekt wird in Sprints geplant. Ein Sprint dauert jeweils 2 Wochen, sodass im Laufe des Semesters 7 Sprints entstehen. Für die einzelnen Sprints sind die wichtigen Meilensteine, die Abgaben, bereits im Voraus eingeplant. Wichtige Tätigkeiten, die eben in den entsprechenden Sprints stattfinden sollen, sind ebenfalls im voraus bereits im Projektplan eingetragen. So ist immer klar, welches Ziel zum Ende des Sprints erreicht werden soll. Eine detaillierte Planung findet immer zu Sprint-Beginn statt - bei dem offene Aufgaben entsprechend im Team verteilt werden. Diese Planung lebt und wird laufend korrigiert dadurch, dass jeweils freitags in der Diskussionsrunde der aktuelle Stand abgeholt, sowie gegebenenfalls Korrekturen an dieser Planung vorgenommen werden.

Meetings innerhalb des Teams finden regelmäßig einmal pro Woche, jeweils am Freitag, statt. In dieser Gesprächsrunde werden zuerst die abgeschlossenen Tasks besprochen, bevor

im Anschluss eventuelle Knackpunkte im Team besprochen werden können. So weiss jeder, woran der jeweils andere aktuell am Arbeiten ist und es wird darüber hinaus eine Plattform geboten, Ideen und Anmerkungen aus anderen Fachbereichen einfließen zu lassen sowie Abhängigkeiten frühzeitig zu erkennen und Teilentwicklungen aufeinander anzupassen.

Planungstool

Das Projekt ist mit dem *GitHub Projects* Tool geplant. Dieses Tool stellt im Wesentlichen ein Kanban-Board dar, welches mit den vielen individuellen Optionen eine feine Abstimmung auf verschiedene Projekte bietet.

Es ist für jedes Team-Mitglied eine Übersicht über den Gesamt-Backlog und eine Ansicht für den individuellen Backlog eingerichtet. Die Ansicht *Roadmap* bietet eine Darstellung aller Tasks auf einem Zeitstrahl. Grundsätzlich könnte zu jedem Issue gleich auch Dokumentation in Form von Kommentaren geschrieben werden. In dieser Zusammenarbeit aber wird allerdings in einem separaten Projektordner dokumentiert. Abbildung 38 zeigt die verschiedenen Menüs sowie das eingerichtete Kanban-Board für das Projektteam.

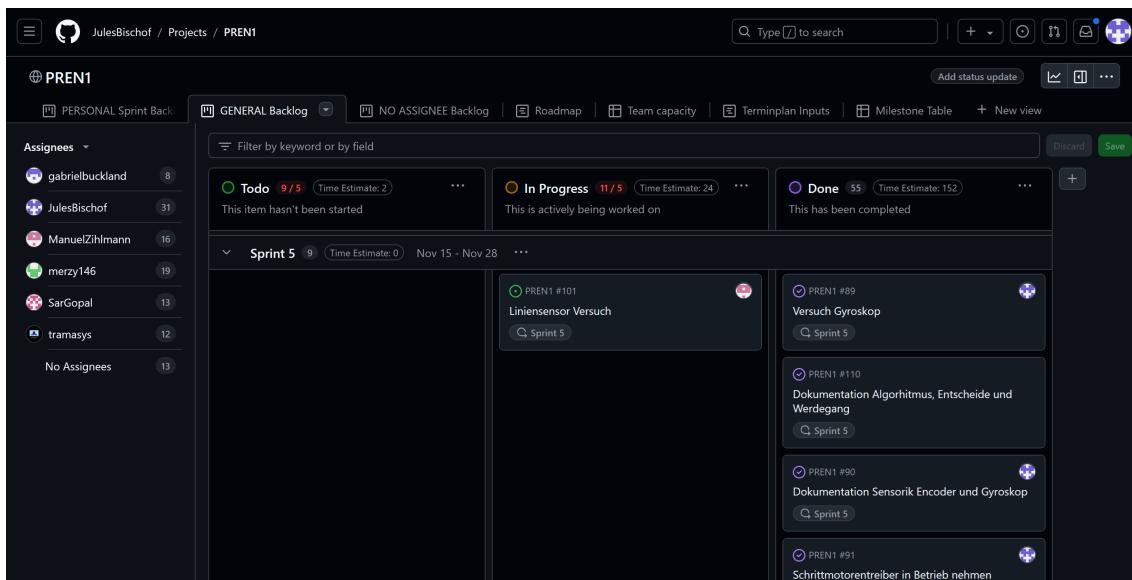


Abbildung 38: GitHub Projects Tool

Datenmanagement

Es ist ein Projektordner eingerichtet, in welchem jedes Team-Mitglied seine Arbeit ablegen kann. Zugriff darauf haben die Team-Mitglieder via *OneDrive*. Dabei gibt es für die Disziplinen Elektronik sowie Mechanik jeweils eigene Ordner, in denen Konstruktionsdaten, CAD-Files oder etwa Bauteilauslegungen und PCB-Planungen abgelegt werden. Die Dokumentation selbst wird in einem eigens dafür eingerichteten *git Repository* geführt und versioniert. Dadurch wird ermöglicht, dass jedes Team-Mitglied zur gleichen Zeit an der in *LATEX*geschriebenen Dokumentation weiterarbeiten kann.

Der Teamleiter erstellt zu jedem Meilenstein bereits frühzeitig eine Vorlage mit allen geforderten Unterdateien in Form einer Disposition. Jedes geforderte Kapitel wird dabei durch eine eigene Unterdatei repräsentiert. So kommt es nur sehr unwahrscheinlich zu Merge-Konflikten innerhalb des Repositorys und es kann mit *Trunk-based development* gearbeitet werden - was den Einstieg in *git* für unerfahrene Mitglieder vereinfacht.

Projektplan

Der mit *GitHub Projects* erstelle Projektplan ist über den Link <https://github.com/users/JulesBischof/projects/5/views/4> einsehbar. Werden wider erwarten Berechtigungen für diese Ansicht benötigt, so kann diese über die E-Mail-Adresse julian.bischof@stud.hslu.ch angefragt werden.

4.3. Risikoanalyse

Anhang H zeigt Risiken auf, welche zu Projektbeginn erfasst wurden. Diese sind nach der ALARP-Methode erfasst und gemeinsam bewertet worden. Durch das sind sehr frühzeitig schon kritische Knackpunkte der Aufgabenstellung erfasst worden und es konnte der Fokus während des Semesters auf entsprechende Teilgebiete gelegt werden. Entsprechend ist viel Zeit in die Ausarbeitung eines Liniensensors gesteckt worden, neben Tests, Objekte via Bilderkennung in schwierigem Umgebungslicht, wie dieses eben am Wettkampf zu erwarten ist, zu testen.

Weiter wurde bei jedem Entwicklungsschritt bereits das Gewicht berücksichtigt, wodurch einige der vorgeschlagenen Konzepte nicht weiter verfolgt werden mussten, was wiederum Zeit eingespart hat.

4.4. Reflexion

In einer gemeinsamen Diskussionsrunde wurde die Arbeitsweise bei der Projektorganisation und -planung offen diskutiert und entsprechende Massnahmen für das Folgemodul abgeleitet. Im Folgenden wird stichpunktartig festgehalten, welche Punkte als positiv und welche als negativ bewertet wurden. Anschliessend werden entsprechende Massnahmen für das Folgemodul festgehalten.

Was ist gut gelaufen

- Das Team ist mit dem Stand der Arbeit am Ende von PREN1 zufrieden. Ein realistisches Konzept wurde produktiv entwickelt.
- Es wurde viel bereits während des Semesters erarbeitet und die Dokumentation ist gegen Ende des Jahres bereits auf einem guten Stand. Es besteht daher kein Zeitdruck am Ende des Semesters, die Abschlussdokumentation fertig zu stellen. Außerdem gibt es keine offenen Fragen mehr zur Lösung bestimmter Teilprobleme der Aufgabenstellung.
- Die Teamdynamik hat gut gepasst, es gab keine Streitigkeiten und es konnte produktiv an der Lösung der Aufgabe gearbeitet werden.

Was nicht so gut gelaufen

- Der Gesamtüberblick über den aktuellen Stand des Konzepts war nicht allen Teammitgliedern jederzeit klar.
- Die Einteilung der Planung in Sprints war bei der gewählten Arbeitsweise etwas überflüssig, da keine konkreten Ziele für das jeweilige Sprintende definiert wurden.
- Es wurde viel Zeit in die Entwicklung eines Greifers mit nur einem Motor investiert, der im Nachhinein nicht umsetzbar war. Eine einfachere Lösung war von Anfang an vorhanden und wird nun für das Projekt verwendet.

- Die Lernkurve einiger Teammitglieder für den Einstieg in die Dokumentation mit L^AT_EX und die Zusammenarbeit über das Tool *git* war etwas steil.
- Da der Zeitaufwand und die Arbeitspakete nicht von Anfang an klar waren, hatte die Vorplanung des Projekts den Charakter eines "Blindflugs", der durch die Meilensteine geführt wurde.

Fazit der Zusammenarbeit und Massnahmen für PREN2

Da der Aufwand und die zu lösenden Teilaufgaben nicht ganz klar waren, gestaltete sich die Planung etwas schwierig. Glücklicherweise gab es in der Konzeptphase nur wenige Abhängigkeiten zwischen den einzelnen Fachbereichen und Teilfunktionen, so dass jedes Teammitglied an seiner *eigenen* Teilfunktion weiterarbeiten und entsprechende *ToDos*'s vorplanen konnte. Es ist zu erwarten, dass diese Art der Planung für das Nachfolgemodul PREN2 nicht zielführend und effizient sein wird, da beim Aufbau, der Inbetriebnahme und dem Testen des Prototyps sehr viele Abhängigkeiten zwischen den einzelnen Teilfunktionen bestehen.

Es ist nun klarer, welche Arbeitspakete bis zum fertigen Produkt existieren und welche Abhängigkeiten auf dem Weg dorthin auftreten. Deshalb wird in den ersten Wochen des neuen Semesters ein in Sprints aufgeteilter Wochenplan erstellt, in dem klare Sprintziele definiert werden. Diese werden z.B. sein *Einzelteile für Prototyp beschafft*, *Prototyp fahrfertig aufgebaut* oder etwa *Firmware Antriebsregelung fertiggestellt*. Damit sind die Aktivitäten bis zum Sommer gut planbar und es kann produktiv gearbeitet werden. Weiter wird dadurch vermieden, dass das gesamte Projektteam aufgrund unterschiedlicher Vorlauf-/Wartezeiten auf bestimmte Komponenten warten muss.

Der Umgang mit den Dokumentationswerkzeugen wie L^AT_EX und *git* hat zwar anfangs zu Schwierigkeiten geführt - da der Umgang damit aber nun klar ist, kann und soll dies auch so beibehalten werden.

5. Nachhaltigkeit

5.1. Nachhaltigkeitsbetrachtung

5.1.1. Ziele der Nachhaltigkeitsbetrachtung

Das Ziel dieser Nachhaltigkeitsbetrachtung ist es, die Entwicklung und den Einsatz unseres autonomen Fahrzeugs unter ökologischen und ökonomischen Gesichtspunkten zu analysieren. Dabei sollen nachhaltige Praktiken und Materialien identifiziert und integriert werden, um negative Auswirkungen auf die Umwelt zu minimieren und eine eventuelle Wiederverwendung der einzelnen Komponenten im privaten Umfang oder bei nachfolgenden Durchführungen dieses Moduls zu ermöglichen.

5.1.2. Abgrenzung

In PREN1 fokussieren wir uns insbesondere auf die Wiederverwendbarkeit der eingesetzten Komponenten, wobei wir besonderes Augenmerk auf die Elektronik legen. Wo immer möglich, verwenden wir High- und Low-Level-Controller wie beispielsweise einen Raspberry Pi aus bereits vorhandenem Bestand. Wir streben eine Optimierung des Energieverbrauchs des autonomen Fahrzeugs an, da ein geringerer Stromverbrauch nicht nur der Umwelt zugutekommt, sondern auch bei der Erstellung des Fahrzeugs Vorteile bietet. Zudem legen wir Wert auf kurze Lieferwege für bestellte Komponenten und bevorzugen Artikel, die wir Nearshore beziehen können.

Aspekte wie die Entsorgung oder das Recycling von nicht wiederverwendbaren Komponenten am Ende ihrer Lebensdauer sowie die detaillierte Analyse der gesamten Lieferkette der verwendeten Materialien werden in dieser Betrachtung nicht ausführlich behandelt. Ebenso werden soziale Nachhaltigkeitsaspekte und die Auswirkungen auf globale Nachhaltigkeitsziele ausserhalb des Projektumfangs nicht vertieft analysiert.

5.1.3. Bezug zu den Sustainable Development Goals (SDGs)

Die Entwicklung des autonomen Fahrzeugs steht in Verbindung mit zwei der von den Vereinten Nationen definierten Sustainable Development Goals:

- **SDG 9 - Industrie, Innovation und Infrastruktur:** Durch die Entwicklung eines innovativen autonomen Systems tragen wir zur Förderung von nachhaltiger Industrie und Infrastruktur bei.
- **SDG 12 - Nachhaltiger Konsum und Produktion:** Durch die Verwendung leichter und effizienter Materialien sowie energiesparender Komponenten unterstützen wir nachhaltige Produktionsmuster.

5.2. Entscheide

In diesem Abschnitt werden die Nachhaltigkeitsaspekte bei den Entscheidungsprozessen in den drei Hauptbereichen des Projekts – Elektronik, Mechanik und Informatik – dargestellt.

5.2.1. Elektronik

- **Komponentenwahl:** Im Rahmen der Komponentenwahl finden Mikrocontroller Verwendung, die sich bereits im Besitz einzelner Teammitglieder befinden. Auch die Bauteile und Sensoren für den Prototypenbau und Tests befinden sich zum grössten Teil im Besitz der Teammitglieder und können somit wiederverwendet werden. Dadurch wird verhindert, dass bei der Herstellung des Endproduktes Bauteile bestellt werden, die später keine Anwendung im Endprodukt finden. Des Weiteren verfügen die entwickelten Leiterplatten über Reserve-Schnittstellen, wie z. B. überzählige digitale Ein- und Ausgänge. Somit müssen bei Änderungen am bestehenden Konzept die Leiterplatten nicht neu entwickelt und bestellt werden. Die elektronische Plattform ist modular aufgebaut. Die Kommunikationsschnittstelle sowie die Möglichkeit, die einzelnen Controller über eine Adressierung individuell anzusprechen, ermöglichen eine einfache und schnelle Systemerweiterung. So kann im Falle nicht ausreichender Ein-/Ausgänge zum Beispiel problemlos ein zweiter MotionController, welcher entsprechende Ausgänge zur Verfügung stellt, eingesetzt werden. Komponenten von weit entfernten Lieferanten müssen so nur einmal für die gesamte Projektlaufzeit bestellt werden. Es wird angestrebt, für sämtliche Leiterplatten, soweit dies realisierbar ist, identische Bauteile für Schnittstellen, Steckverbinder und Schalter-ICs zu verwenden. Die Verwendung identischer Bauteile für Schnittstellen, Steckverbinder und Schalter-ICs ermöglicht die Bereitstellung identischer Ersatzteile für alle Teilsteuерungen und reduziert die Komplexität, was die Wiederverwendung der Boards begünstigt. Weiter werden vorzugsweise Komponenten verwendet, welche sich in den Lagern des Elektrolabors in der Hochschule Luzern befinden - bevor diese neu bezogen werden.
- **Wiederverwendbarkeit:** Der modulare Aufbau der Elektronik ermöglicht die Wiederverwendung der Leiterplatten durch andere Teams aus späteren Jahrgängen. Im Anschluss an das Projekt erfolgt die Veröffentlichung der entsprechenden Baugruppen und Fertigungspläne auf Studenten-Plattformen wie *Studentbox*, um nachfolgenden Jahrgängen zur Verfügung zu stehen. Einige Komponenten, wie der Raspberry Pi und die Mikrocontroller aus persönlichen Beständen, werden für zukünftige private Projekte wiederverwendet.
- **Lieferantenauswahl:** Es wird auf Lieferanten zurückgegriffen, deren Komponenten die Hochschule Luzern auch für eigene Entwicklungen verwendet. Dadurch können grössere Bestellungen mit anderen Teams und der Hochschule gemeinsam getätigt werden, was Lieferungen in grösseren Mengen erlaubt. Zu diesen zählen insbesondere *Mouser*, *DigiKey* und *Conrad*. Obwohl die Leiterplatten vom asiatischen Hersteller PCBWAY bezogen werden, wird durch den modularen Aufbau der Elektronik die Umweltbelastung so gering wie möglich gehalten.

5.2.2. Mechanik

- **Materialauswahl:** Das Chassis besteht aus MDF-Holzplatten und dem 3D-Druck-Material PLA. MDF wird üblicherweise aus Rest- und Abfallfasern hergestellt, was eine ressourcenschonende Nutzung ermöglicht. Die thermische Verwertung von MDF-Platten ist derzeit die gängigste Recyclingmethode, bei der die Platten zur Energiegewinnung genutzt werden. Nach Abschluss des Projekts besteht jedoch die Möglichkeit, diese Materialien in neuen Anwendungen weiter zu verwerten. Ergänzend kommt das 3D-Druck-Material PLA zum Einsatz, das aus nachwachsenden Rohstoffen wie Maisstärke gewonnen wird und unter industriellen Bedingungen biologisch abbaubar ist. Durch das additive Fertigungsverfahren wird Material gezielt nur an den benötigten Stellen aufgetragen, wodurch weniger Abfall im Vergleich zu spanenden Verfahren entsteht. Beide Materialien können lokal beschafft und verarbeitet werden, was den Transportaufwand und damit verbundene Emissionen minimiert.
- **Modulares Design:** Ein weiterer zentraler Ansatz zur Förderung der Nachhaltigkeit war das modulare Design des Systems. Die Konstruktion in austauschbaren Baugruppen ermöglicht es, einzelne Komponenten im Falle eines Defekts gezielt zu ersetzen, anstatt das gesamte System erneuern zu müssen. Dieses Vorgehen verlängert die Lebensdauer des Produkts und reduziert den Ressourcenverbrauch erheblich. Am Ende des Projekts können die Baugruppen entweder in neuen Projekten wieder verwendet oder sortenrein recycelt werden. Ein modularer Aufbau hat zudem den Vorteil, dass bei zukünftigen Iterationen nicht alle Bauteile neu gefertigt werden müssen, was den Materialeinsatz weiter reduziert. Trotz des anfänglichen Mehraufwands in der Entwicklungsphase zahlt sich dieses Designprinzip langfristig aus, da es den Bedarf an neuen Ressourcen minimiert und so einen wesentlichen Beitrag zur Ressourcenschonung und Reduktion von Abfall leistet.

5.2.3. Informatik

- **Verwendung von Open-Source-Software:** Für die Entwicklung des Simulators wird ausschließlich Open-Source-Software eingesetzt. Dies fördert die Nachhaltigkeit durch Reduzierung von Lizenzkosten und Abhängigkeiten von proprietärer Software und unterstützt gleichzeitig die Open-Source-Gemeinschaft. Die Nutzung frei verfügbarer Ressourcen ermöglicht es, den Quellcode an spezifische Anforderungen anzupassen, was zu effizienteren und ressourcenschonenderen Anwendungen führt.
- **Nutzung vorhandener Hardware:** Es werden Raspberry Pi Einplatinencomputer verwendet, die bereits verfügbar sind. Durch die Wiederverwendung bestehender Hardware wird der Bedarf an neu produzierten elektronischen Geräten minimiert, was zur Reduzierung von Elektroschrott und zur Schonung natürlicher Ressourcen beiträgt. Diese Entscheidung ist sowohl ökologisch sinnvoll als auch kosteneffizient und unterstützt die Nachhaltigkeitsziele des Projekts.

5.3. Reflexion

Während zu Beginn der Projektarbeit etwaige Nachhaltigkeitsaspekte vernachlässigt wurden, rückte der Nachhaltigkeitsgedanke gegen Ende der Konzeptionsphase immer mehr in den Vordergrund. Die anfängliche Vernachlässigung lässt sich hauptsächlich auf ein Fehlen einer konkreten Lösung zurückführen. Ohne handfeste Konzepte zur Realisierung des autonomen Fahrzeugs, gestaltete es sich schwierig Punkte zu identifizieren, wo wir nachhaltig handeln könnten. Erst als die Konzeptionen konkreter wurden, konnten wir uns Massnahmen vorstellen, die in die eventuelle Realisierung einfließen könnten. Wir stellten jedoch schnell fest, dass wir mit unserem limitierenden Budget nur sehr geringfügig Entscheidungen treffen konnten, welche die Nachhaltigkeit des Projekts signifikant beeinflussen würden. Ein Punkt den wir hier hervorheben wollen ist, dass wir bei der Beschaffung von Materialien und Komponenten mehrheitlich nicht auf Übersee zurückgreifen. Diese Entscheidung stand aber in starkem Konflikt mit dem einzuhaltenden Budget. Ein weiterer Punkt warum sich die Auseinandersetzung mit der Nachhaltigkeit auf das Semesterende verschob, war die Tatsache, dass die operativen Bewertungskriterien viel höher gewichtet werden. Es wird schlussendlich nur einen minimalen Einfluss auf die Bewertung haben, wenn man sich für eine maximal nachhaltige Lösung entscheidet. Die Abzüge bei Nichteinhalten der Anforderungen an das Fahrzeug sind hier um einiges grösser. Die investierte Zeit wird demnach priorisiert auf die Erfüllung der Anforderungen gelegt, während die Nachhaltigkeit als Nachgedanke behandelt wird. Die Sensibilisierung für das Nachhaltigkeitsthema fanden wir dennoch wertvoll. Eine seriöse Auseinandersetzung mit dem Thema kann im richtigen Rahmen durchaus zu Lösungen führen, die welche nicht nur ökologisch, sondern auch ökonomisch wertvoll sind.

6. Schlussdiskussion

In diesem Abschnitt werden das Finanz- und das Gewichtsbudget erläutert. Es wird auf die Finanzplanung des Gesamtprojektes eingegangen sowie auf die getätigten Ausgaben in Produktentwicklung 1. Ausserdem wird das Gewicht des Roboters in allen Einzelteilen aufgelistet.

6.1. Finanzplanung und Ausgaben

Gemäss der Aufgabenstellung ist das zur Verfügung stehende finanzielle Budget für das gesamte Projekt auf 500 CHF beschränkt. Das Modul Produktentwicklung 1 darf dabei maximal 200 CHF beanspruchen. Die restlichen 300 CHF können im Folgemodul beantragt werden. Die Komponenten, welche aus dem privaten Umfeld stammen, müssen gemäss der Vorgabe zum halben Preis verrechnet werden. Aufgrund der limitierten finanziellen Mitteln ist eine sorgfältige Finanzplanung für das Gesamtprojekt bereits in der Planungsphase unerlässlich. Im folgenden Kapitel wird auf die Finanzplanung für das Gesamtprojekt sowie auf bereits getätigte Ausgaben in Produktentwicklung 1 eingegangen.

Finanzplanung

Die Tabelle 3 veranschaulicht die finanzielle Budgetplanung für das gesamte Projekt. Die Tabelle ist in Kategorien wie Antrieb, Greifeinheit, Chassis und Energieversorgung unterteilt, um eine bessere Übersicht zu gewährleisten. Die geschätzten Kosten basieren auf aktuellen Marktpreisen und enthalten auch Kleinteile sowie Leiterplatten. Gewisse Komponenten sind schon im Besitz von den Entwicklern. Gemäss der Vorgabe werden diese Komponenten direkt mit dem halben Marktpreis in der Tabelle angegeben. Eigenkonstruktionen sind mit einem realistischen Schätzwert angegeben. Zu beachten ist, dass sich die Preise ständig ändern. Die Tabelle dient daher als Richtwert.

Komponente	Hersteller	Hersteller Nr.	Besitz	Anzahl	Kosten [CHF/stk]	Kosten total [CHF]
Antrieb						
Schrittmotoren	DFRobot	FIT0278	Nein	2	12.65	25.3
Schrittmotorentreiber	ADI-Trinamic	TMC5240 Eval Board	Ja	2	31.5	63
Greifeinheit						
Lichtschranke Hinder-niss	SICK	WL100-P3430	Ja	1	15.25	15.25
Servo	DFRobot	SER0063	Nein	2	13.67	27.34
PCB	PCBWY	-	Nein	5	4	4
Chassis						
LIDAR	Benewake	TF Luna I2C	Nein	1	22.6	22.6
Liniensensor	Eigenkonstruktion	-	Nein	1	40	40
Ultraschallsensor	OSEPP Electro-nics	HC-SR04	Ja	1	1.55	1.55
PCB	PCBWY	-	Nein	5	7	7
Antriebsräder	DFRobot	FIT0500	Nein	2	1.37	2.74
Laufkugeln	Polulu	Polulu Item: 954	Nein	1	4.5	4.5
Lichtschranken Enco-der	Panasonic	PM-F45-P	Ja	2	5.75	11.5
Controller						
Raspberry Pi 4 2GB RAM	Raspberry Pi	SC0193(9)	Ja	1	26	26
Raspberry Pi Kamera 12MP	Raspberry Pi	RASPBERRY PI CA-MERA 3	Nein	1	25.35	25.35
Raspberry Pico	Raspberry Pi	SC0915	Ja	2	1.8	3.6
PCB	PCBWY	-	Nein	5	7	7
Energieversorgung						
Akkupack	Swaytronic	7640182625221	Nein	1	41.7	41.7
Ladegerät	Voltcraft	1597950-UP	Nein	1	39.95	39.95
PCB	PCBWY	-	Nein	1	7	7
Elektronik						
Kleinteile Elektronik	Elektro-Werkstatt HSLU	-	Nein	-	-	40
Sonstige Produktions-kosten	-	-	Nein	-	-	20
Total						435.38

Tabelle 3: Finanzplanung

Aus der vorliegenden Tabelle geht hervor, dass die Gesamtkosten 415,38 CHF betragen. Dies liegt unter dem ursprünglich veranschlagten Budget von 500 CHF. Das resultierende Restbudget beträgt 64,62 CHF, welches für unvorhergesehene Ausgaben oder zusätzliche Anschaffungen genutzt werden kann.

6.1.1. Ausgaben Produktentwicklung 1

Im Rahmen des Moduls Produktentwicklung 1 wurden die in Tabelle 4 aufgeführten Ausgaben getätigt. Es wird an dieser Stelle nochmals darauf hingewiesen, dass das zur Verfügung stehende Budget für Produktentwicklung 1 auf 200 CHF begrenzt ist. Die Komponenten, welche mit Eigenbesitz markiert sind, stammen direkt aus dem privaten Umfeld der Entwickler. Bei diesen Komponenten ist bereits der halbe Marktpreis in der Tabelle eingetragen.

Komponente	Lieferant	Hersteller	Hersteller Nr.	Anzahl	Kosten [CHF/stk]	Kosten total [CHF]
Antrieb						
Schrittmotoren	Digikey	DFRobot	FIT0278	2	12.65	25.3
Greifeinheit						
Lichtschranke	Eigenbesitz	SICK	WL100-P3430	1	15.25	15.25
Servo	Mouser	DFRobot	SER0063	2	13.67	27.34
Chassis						
LIDAR	Mouser	Benewake	TF Luna I2C	1	22.6	22.6
Ultraschallsensor	Eigenbesitz	OSEPP Electronics	HC-SR04	1	1.55	1.55
Antriebsräder	Mouser	DFRobot	FIT0500	2	1.37	2.74
Liniensensor						
Trimmerpotentiometer 200Ohm SMD	Mouser	Vishay	TS3YJ201MR15	10	1.85	18.5
UV-Emitter 395nm	Digikey	Bivar Inc.	UV3TZ-395-15	10	1.573	15.73
UV-Fototransistor 630nm	Digikey	Everlight Electronics Co Ltd	ALS-PT204-6C/L177	10	0.496	4.96
IR-Emitter 940nm	Digikey	würth Elektronik	15400394F3590	2	0.351	0.702
IR-Fototransistor 940nm	Digikey	würth Elektronik	1540031NC6090	2	0.329	0.658
PCB-Liniensensor	PCB-Way	PCBWay	-	5	7	7
Produktion						
3D-Druck PLA	Pauschale	-	-	0.2 Kg	23 CHF/Kg	4.6
3D-Druck PETG	Pauschale	-	-	0.2 Kg	23 CHF/Kg	4.6
Total						151.53

Tabelle 4: Ausgaben PREN1

Die Gesamtausgaben aus der Tabelle 4 belaufen sich auf 151,53 CHF und liegen damit innerhalb des vorgesehenen Budgets von 200 CHF. Mit diesen Komponenten konnten erste Prototypen entwickelt und grundlegende Funktionen getestet werden. Die Entwicklung des Liniensensors fällt finanziell höher aus als erwartet. Dies ist darauf zurückzuführen, dass der Preis der UV-Emitter und der Trimmerpotentiometer bei der Budgetkalkulation unterschätzt wurden.

6.2. Gewichtsbudget

Gemäss der Aufgabenstellung ist das Gewicht des Fahrzeugs auf 2 Kilogramm begrenzt. Daher ist eine sorgfältige Gewichtsanalyse bereits in der Planungsphase von entscheidender Bedeutung. In Tabelle 5 sind die Gewichte der verschiedenen Komponenten aufgeführt. Für Zukaufteile wurden die Gewichte den Herstellerdaten entnommen, während bei Eigenkonstruktionen die Gewichtswerte anhand von Abschätzungen im CAD unter Berücksichtigung des verwendeten Materials berechnet wurden. Die Gewichtsanalyse zeigt, dass das Fahrzeug knapp unter der Gewichtslimite ist, leider ist diese zu knapp bemessen.

Komponente	Hersteller	Hersteller Nr.	Anzahl	Gewicht [g]/stk	Gewicht total[g]
Motor					
Schrittmotoren	DFRobot	FIT0278	2	287	574
Schrittmotorentreiber	DFRobot	DRI0043	2	38	76
Servomotor			2	80	160
Elektronik					
Akkupack	Swaytronic		1	167	167
LCD			0	70	0
zusätzliche PCBs	JLCPCB		4	60	240
Sensorpauschale	diverse		9	10	90
Microcontroller	Raspberry Pico		2	4	8
Controller	Raspberry Pi 5		1	46	46
Mechanik					
Räder	Pauschale		2	40	80
Chassis	Pauschale		1	200	200
Greifeinheit			1	100	175
Verkabelungspauschale			1	100	100
Schraubenpauschale			1	50	50
Summe					1996

Tabelle 5: Gewichtsbudget

Während der Entwicklung wurde deutlich, dass die Greifereinheit und das Chassis schwerer sind als ursprünglich im Gewichtsbudget angenommen. In der nächsten Iteration wird das Chassis vollständig im 3D-Druckverfahren gefertigt, voraussichtlich mit einem Wabenmuster, um durch gezielte Aussparungen Gewicht zu reduzieren. Die PCBs werden in einem kompakten Turm mithilfe von Disanzbuchsen übereinander gestapelt, wodurch mehrere Stützen und 3D-Druck Platten eingespart werden. Jedoch werden diese Gewichtsoptimierungen erst im PREN2 umgesetzt.

6.3. Erfahrungen & Lessons learned

6.3.1. Mechanik

Zu Beginn schien die Aufgabenstellung vergleichsweise unkompliziert und wenig zeitintensiv zu sein. Im Laufe des Projekts zeigte sich jedoch, dass einige Aspekte deutlich komplexer waren als ursprünglich erwartet.

Die Ideenfindung sowie die Einigung auf ein Gesamtkonzept verliefen im Team insgesamt reibungslos und zielorientiert. Erste Tests mit physischen Prototypen der Greifeinheit zeigten, dass ein simpler Mechanismus mit zwei Motoren erheblich zuverlässiger ist als ein komplexes System mit nur einem Motor.

Ein wesentlicher Aspekt, der während der Konstruktion unterschätzt wurde, war der Entwicklungsaufwand für das Chassis. Die Platzierung der verschiedenen Komponenten im begrenzten Bauraum, insbesondere in enger Abstimmung mit den Elektrotechnikern, erwies sich als deutlich zeitintensiver und anspruchsvoller als angenommen. Während dieses Prozesses zeigte sich, wie wichtig eine enge interdisziplinäre Zusammenarbeit ist, um Konflikte zwischen mechanischen und elektrischen Anforderungen zu minimieren und die Effizienz in der Umsetzung zu steigern.

Eine besondere Herausforderung stellte zudem die Einhaltung des maximalen Systemgewichts dar. Diese Einschränkung erwies sich als besonders herausfordernd, da sie die Auswahl der Materialien und die Gestaltung des Aufbaus stark beeinflusst. Dieser Punkt wird im Rahmen von PREN2 ein zentraler Schwerpunkt sein, um gezielt Gewichtseinsparungen umzusetzen und das Design weiter zu optimieren. Die frühzeitige Berücksichtigung solcher Randbedingungen ist eine wertvolle Erkenntnis, die wir in die weitere Entwicklung mitnehmen.

6.3.2. Elektronik

Die Entwicklung eines Systems, das für viele mögliche Lösungen offen sein soll, ist schwierig, wenn nicht klar ist, welchen Umfang das System später haben wird. Konkret geht es um die Anzahl und Form der benötigten Sensorschnittstellen und Kommunikationsschnittstellen. Was zu Anfang stark unterschätzt wurde, ist der Energiebedarf, den ein Computer wie der Raspberry Pi 5 mit sich bringt. Der Raspberry Pi 5 mit einer Leistungsaufnahme von 25W ist ein enormer Stromverbraucher in unserem System, was angesichts seiner Grösse überrascht. Ausserdem sind die Preisunterschiede bei der Vergabe von Fertigungsaufträgen an asiatische Leiterplattenhersteller erheblich. Werden die exakten Baumasse von 100 mm × 100 mm sowie die vorgegebenen Via-Grössen nicht eingehalten, steigen die Preise für diese vermeintlich günstigen Leiterplatten rasant.

Die Entwicklung des Liniensensors hat viel Zeit in Anspruch genommen und zu einigen spannenden Erkenntnissen geführt, als die Ströme für verschiedene Lichtspektren verglichen wurden. Es war von entscheidender Bedeutung, sich nicht auf einen Lösungsansatz festzusetzen, sondern lösungsoffen für mehrere Optionen zu sein. Die Entscheidung, die digitale Auswertung der Entladzeit eines Kondensators durch die analoge Auswertung eines Spannungspiegels zu ersetzen, hat sich als ausserordentlich vorteilhaft erwiesen, da so wesentlich schnellere Messzeiten zu erreichen sind. Diskussionen mit den anderen Teams erweisen sich als sehr spannend, da alle vor den gleichen Problemen stehen, aber eigene Lösungsansätze verfolgen.

6.3.3. Informatik

Das Entwickeln von Software auf constrainter Hardware beansprucht die Evaluation von Faktoren, welche bei klassischen Programmieraufgaben nicht in Betracht gezogen werden müssen. Die Erfahrung und damit zu überwindenden Herausforderungen sind wertvoll und ermöglichen eine differenzierte Auseinandersetzung mit einem sonst kaum besprochenen Themengebiet in Rotkreuz. Im Folgekurs PREN2 wird sich dieser Umstand sicherlich noch massgebend auf die eventuelle Realisierung auswirken. Die kollaborative Entwicklung des Simulators ging reibungslos von statten, was auch dem Zustand geschuldet ist, dass wir uns bereits früh gegenseitig auf unsere Stärken und Schwächen geachtet und entsprechend geplant haben. Aufgrund von diesen Planungen haben wir uns aber auch Mehraufwand in PREN2 eingefahren. Der für den Simulator geschriebene Code kann voraussichtlich nicht für das physische Fahrzeug wiederverwendet werden. Daraus resultiert ein bemerkbarer Aufwand beim Implementieren des Pfadfindungsalgorithmus, da der geschriebene JavaScript Code für den Simulator manuell in eine Sprache transpiliert werden muss, welche auf unseren Low -und Highlevel Controllern effizient lauffähig ist.

6.4. Fazit

Das PREN 1-Projekt ermöglichte es unserem Team, die Grundlagen für die Entwicklung eines autonomen Roboters zu schaffen. In der frühen Phase der Ideenentwicklung war es naturgemäß herausfordernd, fachübergreifende Abhängigkeiten im Voraus vollständig zu planen. Viele Verknüpfungen zwischen Mechanik, Elektronik und Informatik wurden erst im Projektverlauf sichtbar, wodurch die Arbeitsweise häufig reaktiv gestaltet werden musste.

Trotz dieser Herausforderungen konnten wir ein solides Gesamtkonzept entwickeln, das den Anforderungen gerecht wird und die Grundlage für die nächste Projektphase legt. Die Fähigkeit, flexibel auf neue Situationen zu reagieren, sowie die kreative Problemlösung im Team waren entscheidend für den Erfolg in diesem Modul. Gleichzeitig wurde deutlich, dass für PREN 2 eine gezielte Verbesserung in der Planung und Abstimmung erforderlich ist, um effizienter zu arbeiten und Synergien besser zu nutzen.

6.4.1. Erfüllung der Anforderungen

Der Erfüllungsgrad der Anforderungen in PREN 1 zeigt, dass viele zentrale Ziele erreicht wurden. Dennoch gibt es einige offene Punkte, insbesondere bei den Festanforderungen, die in PREN 2 priorisiert bearbeitet werden müssen (siehe Anhang J).

Zu den offenen Festanforderungen gehören vor allem das Gewicht und die Sicherheit. Das System liegt aktuell knapp innerhalb des Gewichtslimits, jedoch ist eine gezielte Optimierung notwendig, um Spielraum für zukünftige Anpassungen zu schaffen. Zusätzlich fehlen wichtige Sicherheitsmassnahmen, wie robuste Notfallmechanismen, die für einen sicheren Betrieb unverzichtbar sind.

Eine für die Testphase besonders relevante Wunschanforderung ist die Akkulaufzeit. Diese konnte in PREN 1 noch nicht ausreichend evaluiert werden, was im nächsten Modul priorisiert erfolgen sollte, um realistische Tests des Systems zu ermöglichen. Weitere Wunschanforderungen, wie die Zeitauswertung während des Parcours oder die Debug-Schnittstelle, bleiben sekundär und können je nach Projektverlauf ergänzt oder verworfen werden.

Zusammenfassend zeigt der Erfüllungsgrad der Anforderungen, dass das Projekt auf einem soliden Fundament steht. Die offenen Punkte bieten klare Handlungsfelder für PREN 2 und können gezielt weiterentwickelt werden, um die vollständige Erfüllung aller Anforderungen sicherzustellen.

6.4.2. Ausblick und Risiken

Basierend auf den Erfahrungen aus PREN 1 ergeben sich für PREN 2 konkrete Handlungsfelder sowie potenzielle Risiken, die es zu adressieren gilt:

Elektronik

- **Leiterplatten bestellen und zusammenbauen:** Verzögerungen bei der Beschaffung könnten die Inbetriebnahme der Elektronik verzögern. Eine rechtzeitige Bestellung ist daher essenziell.
- **Firmwareentwicklung:** Die Firmware für die Steuerung der Boards und Antriebe muss entwickelt und getestet werden, wobei Fehler in der Programmierung die Funktionalität beeinträchtigen könnten.

- **Strombedarf evaluieren:** Eine fehlerhafte Bewertung des Strombedarfs könnte zu Problemen mit der Energieeffizienz oder einer unzureichenden Batteriekapazität führen.

Mechanik

- **Überarbeitung von Chassis und Gewicht:** Das Design des Chassis muss überarbeitet werden, um das Gewicht zu optimieren.
- **Prototyp fertigen und testen:** Ein früher Prototyp ist notwendig, um potenzielle Probleme rechtzeitig zu identifizieren. Verzögerungen könnten die Tests und die Integration mit anderen Disziplinen beeinträchtigen.
- **Räder mit unzureichendem Grip:** Sollten die Räder nicht genügend Traktion bieten, könnte dies die Navigation und Manövriertfähigkeit des Roboters negativ beeinflussen.

Informatik

- **Karten erstellen und optimieren:** Die Entwicklung einer präzisen Kartierungsstrategie ist essenziell. Fehlerhafte Karten könnten die erfolgreiche Navigation behindern.
- **Charakter- und Abgangserkennung:** Die Kamerafunktionen zur Erkennung von Wegabgängen und Buchstaben müssen umfangreich getestet werden. Unzureichende Ergebnisse könnten die Zielerreichung gefährden.
- **Algorithmusanpassung:** Der Algorithmus muss auf eine ressourcenschonende Sprache für den Raspberry Pi umgeschrieben werden, um Speicher- und Rechenressourcen effizient zu nutzen.

Allgemeines

- **Projektplanung verbessern:** Massnahmen zur besseren Abstimmung zwischen den Disziplinen sind entscheidend, um Leerlaufzeiten zu minimieren. Eine unzureichende Planung könnte Verzögerungen verursachen.
- **Abhängigkeiten erkennen:** Die Identifikation und Berücksichtigung fachübergreifender Abhängigkeiten ist notwendig, damit alle Teilbereiche parallel arbeiten können.

Durch die frühzeitige Berücksichtigung dieser Risiken und die Umsetzung der genannten Massnahmen wird es möglich sein, die offenen Punkte effizient zu bearbeiten und den Roboter in PREN 2 erfolgreich weiterzuentwickeln.

7. Verzeichnisse

7.1. Abbildungsverzeichnis

Abbildungsverzeichnis

1.	Beispiel Wegenetzwerk aus Aufgabenstellung	1
2.	Prototyp des Fahrzeugs mit Bildlegende in Siemens NX	3
3.	Grundplatte in Siemens NX	5
4.	Chassis mit sämtlichen Anbauteilen in Siemens NX	6
5.	Grundplatte mit Rädern und Laufkugel in Siemens NX	7
6.	Übersicht der Greifereinheit im NX Siemens	8
7.	Greiferablauf - Anheben des Hindernis	9
8.	Greiferprototyp	10
9.	Vereinfachte Gesamtübersicht Elektronikkonzept	11
10.	Aufbau einer Nachricht	12
11.	Vergebene Adressen der Prints	13
12.	PowerBoard Blockschaltbild	14
13.	Ansicht PowerBoard	15
14.	Blockschaltbild GND-Konzept	15
15.	Schrittmotor	16
16.	Evaluationboard TMC5240	16
17.	Ansteuerungstopologie Schrittmotoren	17
18.	Blockschaltbild Motion Controller	18
19.	3D-Ansicht Motion Controller	19
20.	Steuerablauf Motion Controller Konzept	20
21.	Klassendiagramm Konzept Motion Controller	21
22.	Untergrund während des Wettkampfs	22
23.	Konzept der Auswertung mittels ADC	23
24.	Liniensensor in Kicad von oben	23
25.	Liniensensor in Kicad von unten	23
26.	Die tiefen Werte stellen das Klebeband dar und die hohen die Fliesen	24
27.	Konzept Hinderniserkennung	25
28.	Blockschaltbild Sensorik Streckenrückverfolgung	26
29.	Masse der Lochscheibe	27
30.	3D-Ansicht Grip Controller	28
31.	RaspberryHAT in Draufsicht	30
32.	RaspberryHAT in 3D Ansicht	30
33.	Aufbau des Simulators und dessen Komponenten	35
34.	Screenshot des Bereichs für den Graph im Simulator (GUI)	37
35.	Screenshot des Bereichs für Betriebsmodi und Konfiguration im Simulator	38
36.	Screenshot des Bereichs für Logs im Simulator	39
37.	Organigramm Team 10	41
38.	GitHub Projects Tool	42
39.	Konzept 1 im NX Siemens	61
40.	Konzept 2 im NX Siemens	62
41.	Konzept 3 im NX Siemens	63
42.	Blockschaltbild Elektrokonzept Gesamtübersicht	65
43.	Blockschaltbild des Akku-Konzepts 12V Variante	67
44.	MOSFETS für Batterietrennung	73

45.	Quellenumschaltung	74
46.	Inrush Limiter Schaltung	75
47.	DC/DC Konverter 12V, 60W	76
48.	ADC zur Batterieüberwachung	77
49.	Berechnungsformel aus Maschinenbau Formelsammlung	79
50.	TMC 5240 Evaluation-Board	83
51.	DFROBOT FIT0278 Schrittmotor	83
52.	Konzept für die Ansteuerung der Schrittmotoren	84
53.	Auszug Schema Spannungsversorgung	86
54.	Auszug Schema TMC5240 - Anbindung	87
55.	Auszug Schema Kommunikationsschnittstellen	88
56.	Auszug Schema Digitaler Eingang	89
57.	Auszug Schema Analogeingänge	90
58.	Auszug Schema Digitale Ausgänge	91
59.	Auszug Schema I2C Verbindungen	92
60.	Auszug Schema Gyroskop und HCSR-04	93
61.	Messgenauigkeit Ultraschallsensor	95
62.	Messgenauigkeit LIDAR	95
63.	Lichtschranke SICK	96
64.	TFLuna LIDAR	97
65.	Konzept Hinderniserkennung	98
66.	MPU6050 Gyroskop EVAL-Board	99
67.	Konzept des Liniensensors	101
68.	Vermassung des Gehäuses in Siemens NX	102
69.	Isometrische Ansicht des Gehäuses in Siemens NX	102
70.	Schema einer Messzelle	103
71.	Vergleich der Kennzahlen von Fliese (oben) und Klebeband (unten) des Arduinos	106
72.	Vergleich der Kennzahl Klebeband (vor allem A3) und der Fliese	106
73.	Vergleich der Kennzahl Fuge (A6) und dem Klebeband (A6) umrandet von der Fliese	106
74.	Grafische Darstellung der Risikoanalyse Allgemein	110
75.	Grafische Darstellung der Risikoanalyse Mechanik	111
76.	Grafische Darstellung Risikoanalyse Elektrotechnik	113
77.	Grafische Darstellung Risikoanalyse Informatik	114
78.	Morphologischer Kasten Gerät	127
79.	Morphologischer Kasten Gerät	128
80.	Morphologischer Kasten Gerät	128
81.	Morphologischer Kasten Simulation	129
82.	Nutzwertanalyse Gerät	130
83.	Nutzwertanalyse Gerät	130
84.	Nutzwertanalyse Simulation	131

7.1. Tabellenverzeichnis

Tabellenverzeichnis

1.	Versionsverlauf der Dokumentation	2
2.	Technische Daten des autonomen Fahrzeugs	4
3.	Finanzplanung	50

4.	Ausgaben PREN1	51
5.	Gewichtsbudget	52
6.	Überschlagsrechnung zum Energiebedarf	66
7.	Gewichtsschätzung eines Akku-Pakets 12V Variante	68
8.	Kostenschätzung des Akku-Pakets 12V Variante	68
9.	Kostenschätzung des Ladegeräts der 12V Variante	69
10.	Kostenschätzung des Akku-Pakets 12V Variante mit Kaufteilen	70
11.	Kostenschätzung des Ladegeräts 12V Variante mit Kaufteilen	70
12.	Kostenschätzung des Akku-Pakets 12V Variante mit Kaufteilen	71
13.	Vergleichstabelle der verschiedenen Konzepte	71
14.	Verschiedene Schrittmotoren	81
15.	Verschiedene Schrittmotorentreiber	81
16.	Radnabenmotor DFRobot	82
17.	Radnabenmotor DFRobot	82
18.	BLDC-Treiber	82
19.	Initialisierungsparameter MPU6050	100
20.	Strom durch die beiden Fototransistor in verschiedenen Lichtspektren	104
21.	Strom durch den Fototransistor auf den verschiedenen Messuntergründen	105
22.	Spannungsauswertung des Arduinos über dem Fototransistor	105
23.	Legende Eintrittswahrscheinlichkeit	107
24.	Legende Schadensausmass	107
25.	Legende Bereichsdefinition	107
26.	Erfasste Risiken mit Bewertung	108
27.	Erfasste Risiken im Bereich Mechanik	108
28.	Erfasste Risiken im Bereich Elektrotechnik	109
29.	Erfasste Risiken im Bereich Informatik	109
30.	Erfasste Massnahmen für allgemein betreffende Risiken	109
31.	Erfasste Massnahmen für die Mechanik betreffende Risiken	110
32.	Erfasste Massnahmen für Risikoanalyse	112
33.	Erfasste Massnahmen für Risikoanalyse	113
34.	Quellensammlung	115
34.	Quellensammlung	116
34.	Quellensammlung	117
34.	Quellensammlung	118

8. Anhang

A. Konzepterstellung Greifereinheit

In diesem Abschritt wird die Entwicklung des Greifkonzepts behandelt. Zunächst werden die benötigten Anforderungen erfasst. Danach werden die verschiedenen Konzepte kommentiert. Schliesslich wird eine Entscheidung auf Grundlage der Testergebnisse getroffen und dokumentiert.

Anforderungen

Greifkraft Die Greifkraft muss ausreichend dimensioniert sein, um das Hindernis sicher greifen zu können. Dabei sind die Haftreibung und die Anpresskraft zu berücksichtigen.

$$F_{erforderlich} = \frac{m \cdot g}{\mu_{hr}}$$

$$M_{erforderlich} = F_{erforderlich} \cdot Hebel \cdot Sicherheit$$

Das erforderliche Drehmoment beträgt 0.235 Nm .

Höhenverstellung Die Höhenverstellung muss an das Gewicht des zu hebenden Hindernisses, des Greifers und der Elektronik ausgelegt sein.

$$M_{erforderlich} = (m_{Hindernis} + m_{Elektronik} + m_{Greifer}) \cdot Hebel \cdot Sicherheit$$

Das erforderliche Drehmoment beträgt 0.2 Nm .

Genauigkeit Der Greifer muss den gesamten Bewegungsablauf mit hoher Wiederholgenauigkeit ausführen. Dadurch wird sichergestellt, dass das Hindernis innerhalb des Toleranzbereichs zurückplatziert wird.

Gewicht Das Gewicht der Greifereinheit ist gering zu halten, um das Gesamtgewicht des Fahrzeugs zu reduzieren. Eine Gewichtsreduktion ist ebenfalls bei der bewegten Massen des Greifers wichtig, da sie direkte Auswirkungen auf die Motorauswahl, die Geschwindigkeit und die Energieeffizienz hat.

Konzeption

Die Ideen für die verschiedenen Konzepte wurden durch der Anwendung eines Brainstorming und eines Morphologischen Kastens (Anhang K) entwickelt.

Konzept 1 - Greifer und Höhenverstellung - ein Motor

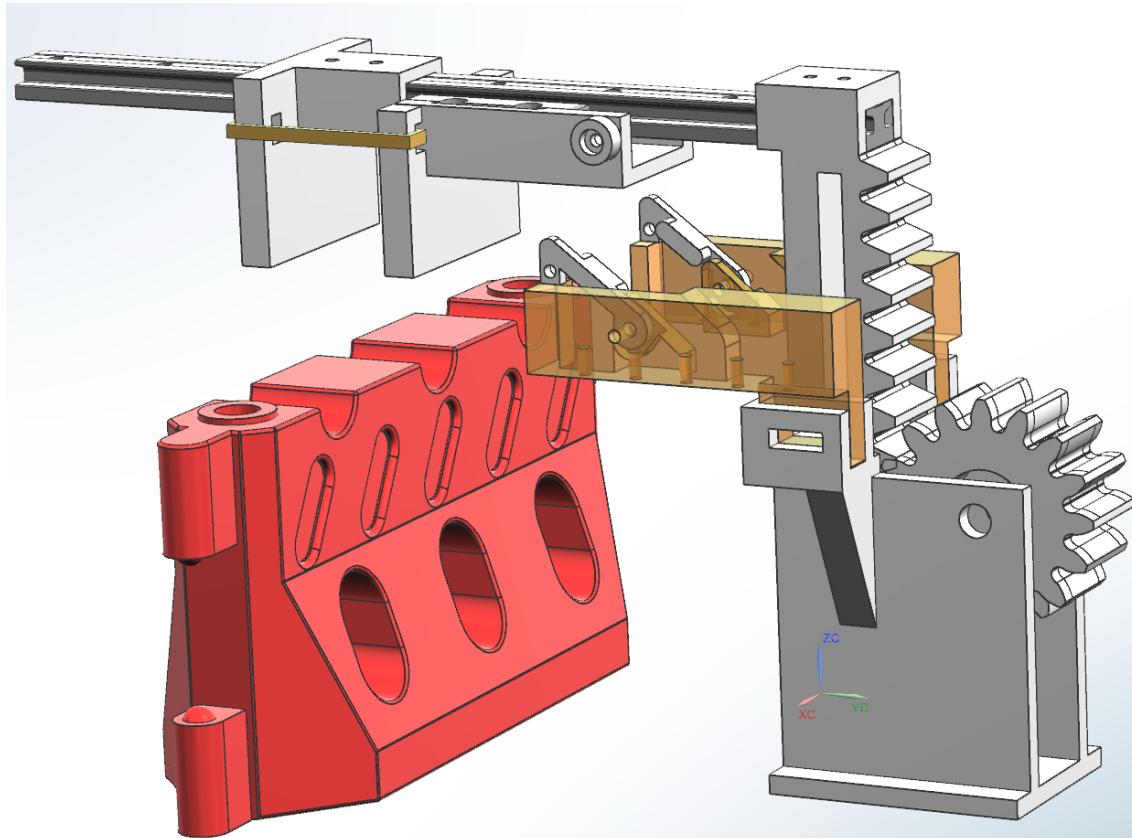


Abbildung 39: Konzept 1 im NX Siemens

Das Ziel dieses Konzepts aus der Abbildung 39 ist es, einen einzigen Motor sowohl für die Greiffunktion als auch für die Höhenverstellung zu nutzen. Der Servomotor treibt ein Zahnrad an, das eine Zahnstange vertikal bewegt und so die Höhe des Greifers einstellt. Der Greifer besteht aus zwei Backen: einer fest montierten vorderen Backe und einer hinteren Backe, die horizontal auf einer Gleitführung beweglich ist. Die beiden Backen werden durch ein Gummiband vorgespannt, sodass sie im Ruhezustand geschlossen bleiben. An den Seiten der hinteren Backe befinden sich Mitnehmer, die entlang einer linearen Nocke geführt werden. Je nach eingestellter Höhe des Greifers bewegen sich die Mitnehmer in der Nocke so, dass die hintere Backe horizontal verschoben wird. Dadurch öffnet und schließt sich der Greifer in Abhängigkeit von seiner Höhe.

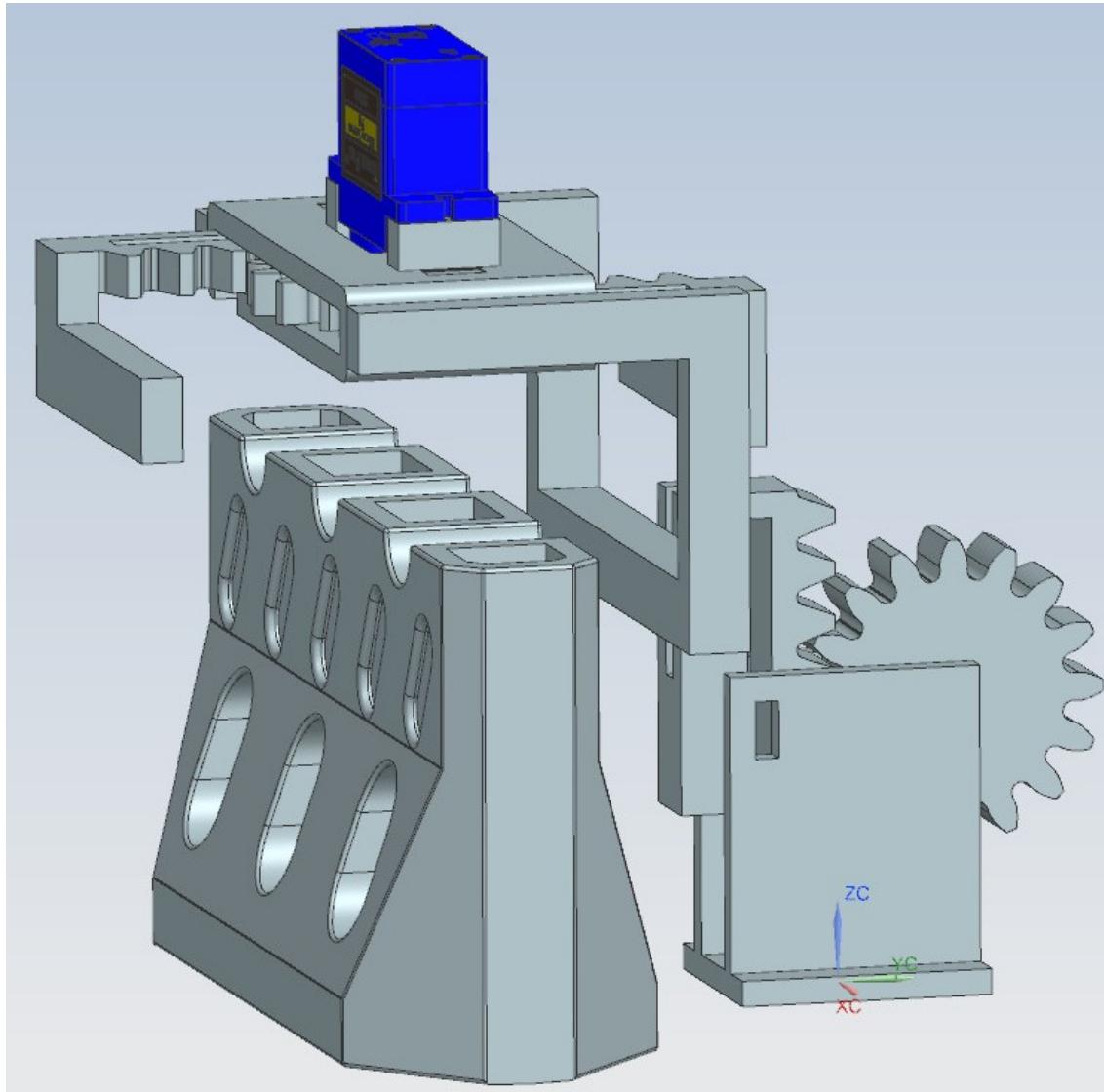
Konzept 2 - Greifer und Höhenverstellung - zwei Motoren

Abbildung 40: Konzept 2 im NX Siemens

Das Ziel dieses Konzepts aus der Abbildung 40 ist die Umsetzung eines einfachen und zuverlässigen Mechanismus. Aus diesem Grund werden separate Motoren für die Greiffunktion und die Höhenverstellung eingesetzt. Der Greifer ist als Parallelgreifer konstruiert. Ein Servomotor treibt ein zentrales Zahnrad an, das die Bewegung auf zwei Zahnstangen überträgt. Diese bewegen sich in entgegengesetzte Richtungen, wobei an jeder Zahnstange eine Greiferbacke befestigt ist. Die Höhenverstellung erfolgt über einen weiteren Servomotor, der ein Zahnrad antreibt, das eine Zahnstange vertikal bewegt. Dadurch kann der Greifer präzise auf die gewünschte Höhe eingestellt werden.

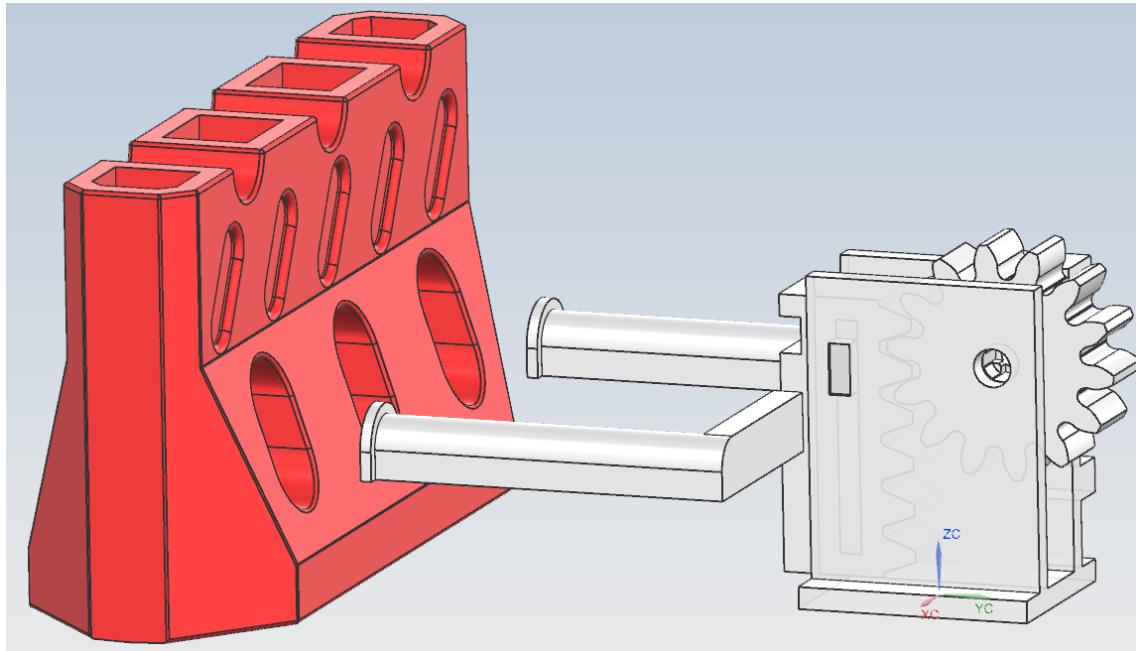
Konzept 3 - Gabelstapler - ein Motor

Abbildung 41: Konzept 3 im NX Siemens

Das Konzept aus der Abbildung 41 basiert auf dem Prinzip eines Gabelstaplers. Die Gabeln werden in die Öffnung des Hindernisses eingeführt. Anschliessend hebt ein Servomotor, der ein Zahnrad antreibt, eine Zahnstange vertikal an und hebt so das Hindernis an.

Fazit und Entscheidung der Konzeptphase

Die Nutzwertanalyse (Anhang L) hat ergeben, dass die Variante mit Parallelgreifer und Höhenverstellung weiterverfolgt werden soll. Daher werden die Konzepte 1 und 2 weiterentwickelt und anschliessend getestet.

Versuche

Konzept 1 Nach mehreren Iterationen wurde ein Prototyp entwickelt, der teilweise funktioniert, jedoch noch nicht den gewünschten Anforderungen entspricht. Das Grundprinzip ist zwar funktionsfähig, jedoch führt der Einsatz eines Gummibands zur Vorspannung zu einer leichten Deformation des Materials. Dies erhöht die innere Reibung, welche der Servomotor nicht überwinden kann. Für die nächste Iteration stellt sich daher die Frage, ob ein anderes Material anstelle des 3D-Drucks besser geeignet wäre.

Konzept 2 Bereits nach einer Iteration hat der Prototyp gute Ergebnisse geliefert. Die Reibung kann jedoch durch eine leichte Erhöhung der Toleranzen im CAD weiter reduziert werden.

Fazit und Ausblick

Die Testergebnisse haben gezeigt, dass Konzept 1 schwer umzusetzen ist, insbesondere im Vergleich zu Konzept 2, das bereits funktionsfähig wäre. Aufgrund des deutlich geringeren Risikos eines Fehlschlags wird daher Konzept 2 weiterverfolgt.

B. Blockschaltbild Elektrokonzept

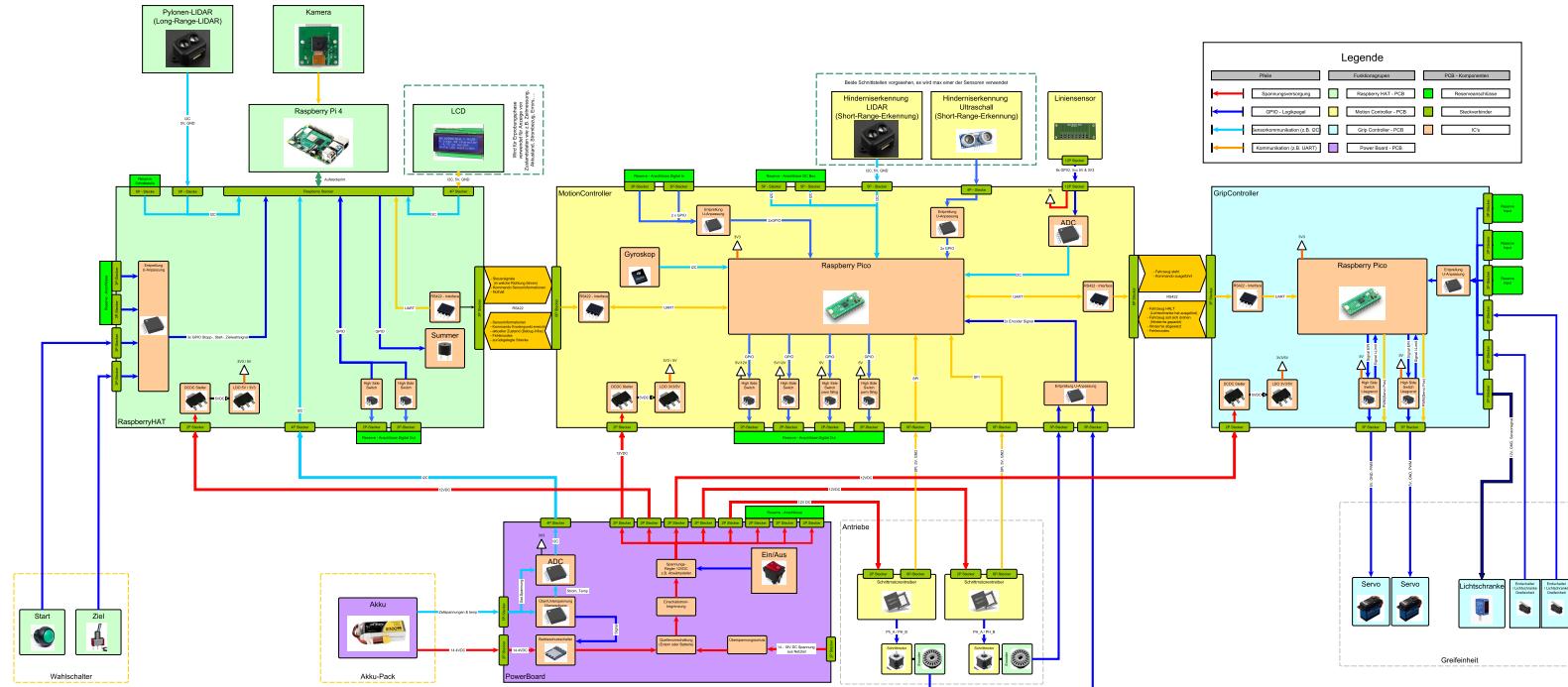


Abbildung 42: Blockschaltbild Elektrokonzept Gesamtübersicht

C. Konzepterstellung Energieversorgung

In diesem Abschnitt wird sich mit der Energieversorgung des Pfadfinders auseinandergezettet. In einem ersten Schritt werden Anforderungen zusammengestellt, dann verschiedene Konzepte kommentiert sowie sich am Ende für ein Konzept entschieden.

Anforderungen

Autonome Versorgung Das Fahrzeug muss autonom fahren können. Deshalb ist eine kabellose Speisung des Gefährts unerlässlich. Die benötigte Spannung im Bordnetz hängt direkt mit den eingesetzten Motoren und ihren Nennspannungen zusammen.

Bordnetz-Spannung Bauteilrecherchen haben ergeben, dass 24V-Akkus im benötigten Leistungsbereich praktisch nicht mit geringem Gewicht erhältlich sind. Ein Vorteil einer höheren Bordnetz-Spannung ist, dass auf Bauteile und Motoren aus dem Industriellen Umfeld zurückgegriffen werden kann. Da diese allerdings in jedem Fall ebenfalls zu schwer sind, wird eine Bordnetz-Spannung von 24V ausgeschlossen. 12V sind dafür ausreichend, um entsprechende Sensorik anzusteuern und gleichzeitig noch genügend Spielraum bei der Auswahl von Motoren zu haben.

Energiebedarf Tabelle 6 zeigt eine überschlägige Rechnung zum benötigten Leistungsbedarf einzelner Teilgruppen. Zusätzliche 25% Toleranz geben den voraussichtlichen Energiebedarf des Gerätes mit $\approx 55W$ an.

Komponente	P [W]	Anzahl	P total [W]
Schrittmotor	12	2	24
Raspberry Pi 5	8.6	1	8.6
RTC & Sensoren	10	1	10
Summe			42.6
Summe inkl. 25% Sicherheit			53.25

Tabelle 6: Überschlagsrechnung zum Energiebedarf

Gewicht Da das Gewicht des Fahrzeugs mit $2kg$ stark beschränkt ist, darf das Akkupaket nicht mehr als $0.5kg$ wiegen.

Strombelastbarkeit Aus dem Leistungsbedarf ergibt sich eine minimale Strombelastbarkeit der Akkumulatoren wie folgt:

$$I_{discharge} = \frac{P}{U_{Bat}}$$

Für die Batterien ergeben sich also mindestens $4.6A$ Strombelastbarkeit für die Batterien.

Konzeption

Das Gewicht ist eine grosse Einschränkung in der Entwicklung des Geräts. Aufgrund seiner hohen Energiedichte werden lediglich Lithium-Ionen sowie Lithium-Polymer betrachtet.

Konzept 1 - 12V Li-Ion-Akku - Eigenentwicklung

Abbildung 43 zeigt ein Blockschaltbild dieser Variante. Im folgenden Unterabschnitt werden die einzelnen Unterbaugruppen kurz kommentiert.

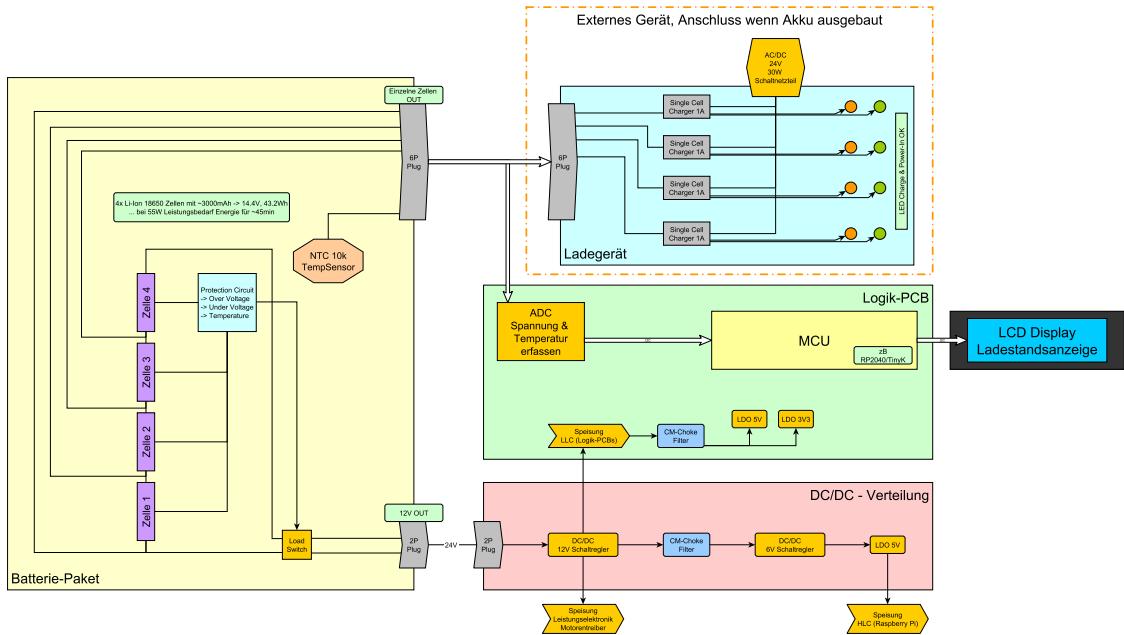


Abbildung 43: Blockschaltbild des Akku-Konzepts 12V Variante

Akkuzelle Es sind 4 Lithium-Ionen-Zellen der Bauform 18650 in Reihe geschaltet, um die gewünschte Spannung zu erreichen. Diese Bauform ist in der Industrie gängig vertreten und entsprechend einfach erhältlich. Die Nennspannung ergibt sich zu $4 \cdot 3.6V = 14.4V$. Die Akkuzelle enthält

$$E_{bat} = 4 \cdot 3.6V \cdot 3Ah = 43.2Wh$$

Energie, die für einen Betrieb von ca. 45 Minuten ausreichen würde.

Mit einem DC/DC-Abwärtssteller werden daraus die benötigten 12V generiert. Lithium-Ionen-Akkus benötigen eine Schutzbeschaltung, um sie vor Unter- oder Überspannung zu schützen. Diese Ereignisse führen nicht nur zu Schäden an der Akkuzelle selbst, sondern auch zum Entflammen des Akkupakets. Dazu ist auf dem Akkupaket selbst ein IC der Firma Texas-Industries verbaut, welcher mit MOSFETS den Akku von der Last trennen kann, falls sich die Spannung einem gefährlichen Bereich nähern würde. Die Schutzbeschaltung wird auf dem Akku selbst vorgesehen, damit sich Ladegerät und Roboter dieselbe Schutzbeschaltung teilen können.

Das Gewicht dieses Akkupakets beläuft sich auf $\approx 320g$, wie Tabelle 7 zeigt. Hier könnte allerdings noch am Gewicht des Gehäuses gespart werden.

Beschreibung	n	[g]	\sum [g]
Batteriezelle	4	30	120
PCB 4 Lagen @100x100mm	1	60	60
Kleinmaterial (Stecker, IC's, Schrauben)	1	15	10
Gehäuse			100
SUMME Battery Pack + 10%			\approx 320g

Tabelle 7: Gewichtsschätzung eines Akku-Pakets 12V Variante

Kostenschätzung Um dieses Konzept bezüglich seiner Kosten bewerten zu können, ist eine Kostenabschätzung erstellt worden. Dazu sind beispielhaft ICs herausgesucht und mit pauschalen gerechnet. Die Summe enthält zu guter Letzt noch einen Aufschlag von 10%, womit ein realistischer Preis gut angenähert werden kann. Die Kostenschätzungen sind Tabelle 8 und Tabelle 9 zu entnehmen.

Beschreibung	Hersteller	Hersteller Nr.	Distributor	n	CHF	\sum CHF
Batteriezelle	Tenpower	INR18650-32HE	Nikon.nl	4	2.35	9.4
Batteriezellenhalter	MPD	BK-18650-CLIP	Digikey	8	0.458	3.66
NTC Temperaturfühler	Vishay Dale	CRCW080510K0JNTC	Digikey	1	0.12	0.12
MOSFET Leistungsschalter P-Kanal	Diodes Incorporated	DMP3036SFV-13	DigiKey	2	0.55	1.1
MOSFET Gate Treiber N-Kanal	Nexperia USA Inc.	2N7002NXAKR	DigiKey	2	0.09	0.18
Battery Protection IC	Texas Instruments	BQ296907DSGR	Digikey	1	0.33	0.33
6P Stecker	TE	2350514-6	Digikey	1	0.83	0.83
2P Stecker	Würth Elektronik	691321300002	Digikey	1	0.3	0.3
Kleinmaterial (R's, C's, L's, ...)				1	3	3
PCB 4 Lagen @100x100mm			JLCPCB	1	8	8
Gehäuse			3D Druck, leicht			0
Versandpauschale				1	10	10
SUMME Battery Pack + 10%						40.61

Tabelle 8: Kostenschätzung des Akku-Pakets 12V Variante

Beschreibung	Hersteller	Hersteller Nr.	Distributor	n	CHF	\sum CHF
Ladegerät IC	3Peak	TPB4056A20-ES1R	DigiKey	4	0.17	0.68
LEDs Ladegerät	generisch 3mm		Digikey	8	0.15	1.2
AC/DC Wandler Ladegerät	Mean Well USA	1866-3343-ND	DigiKey	1	11.44	11.44
6P Buchse	TE Connectivity	2350398-6	DigiKey	1	3.93	3.93
Gehäuse	Hammond Manufacturing	1455N1601	Reichelt	1	17.8	17.8
Kleinmaterial (R's, C's, L's, ...)				1	3	3
PCB 2 Lagen @100x100mm			JLCPCB	1	8	8
Versandpauschale				1	10	10
SUMME Ladegerät +10%						61.655

Tabelle 9: Kostenschätzung des Ladegeräts der 12V Variante

Strombelastbarkeit Die Akkuzellen sind belastbar mit einem Entladestrom von 8A, was vollkommen ausreichend ist für diese Anwendung.

Ladegerät Um die Akkuzelle zu laden, muss auch ein Ladegerät entwickelt werden. Dieses wird mit ICs der Firma 3-Peak realisiert, welche jede Akkuzelle einzeln auf 4.2V mit einem konstanten Strom von 1A lädt. Dadurch, dass jede Zellen über ein eigenes Ladegerät verfügt, werden die Zellen auch bereits bei jedem Ladevorgang balanciert. Versorgt werden diese einzelnen Ladegeräte über ein AC-DC-Schaltnetzteil, welches 5W zur Verfügung stellt.

Die benötigte Leistung berechnet sich wie folgt:

$$P_{crg} = 4 \cdot I_{crg} \cdot U_{crg} = 4 \cdot 1A \cdot 4.2V = 16.8W$$

Das Ladegerät ist nicht teil des Geräts und wird deshalb nicht in seinem Gewicht beurteilt.

Akku Monitoring Die Anforderungsliste des Pfadfinders sieht vor, den Ladezustand des Akkus über ein LCD darzustellen. Um dies zu ermöglichen, werden die einzelnen Zellenanschlüsse nochmals separat herausgeführt, um sie auf dem Roboter selbst mit einem ADC auswerten und entsprechend anzeigen zu können.

Konzept 2 - 12V LiPo Akku - Kaufteil

Eine Akkuzelle muss nicht zwingend selbst entwickelt werden. Aus dem Modellbau-Bereich kann aus einer breiten Palette ein passender Akku ausgewählt und eingesetzt werden. Ein günstiger Anbieter dieser Akku-Pakete ist die Firma *Swaytronic*. Dieser Hersteller bietet nicht nur Akkumulatoren an, sondern auch Lösungen, wie diese aufgeladen werden können. In den Parametern Gewicht, Kapazität und Baugröße bewegen sich die meisten Hersteller in einem ähnlichen Rahmen, weshalb hier auf Akkus der Firma *Swaytronic* stellvertretend für alle Modellbau-Akkuherrsteller eingegangen wird.

Akkuzelle Mit sogenannten 4S-Akkupacks erhält man eine Nennspannung von

$$U_{bat} = 4 \cdot 3.6V = 14.4V$$

Um diese Akkupakete mit den Eigenentwicklungen vergleichen zu können, wird in einem ähnlichen Kapazitätsbereich von 3000mAh geschaut. Diese Akkupacks haben praktisch

den gleichen Energieinhalt wie die Eigenentwicklung mit $43.2Wh$. Herausgesucht wurde der Akku *SWAY-EL LiPo 4S 14.8V 3000mAh 35C/70C EC3* von *Swaytronic*. Die Schutzbeschaltung wird auch hier auf dem Roboter umgesetzt werden müssen, da die Akkus eine solche nicht integriert haben. Zugriff auf die Zellspannungen erhält man über den separat herausgeführten Steckverbinder, welcher eben genau diese Anschlüsse bereitstellt.

Kostenschätzung Die Akkuzelle beläuft sich auf den in Tabelle 10 berechneten Preis. Die Kosten für das Ladegerät kann Tabelle 11 entnommen werden.

Beschreibung	Hersteller	Hersteller Nr.	Distributor	n	CHF	$\sum CHF$
Batteriepaket	Swaytronic	7640182625344	Swaytronic.ch	1	58.35	58.35
MOSFET Leistungsschalter P-Kanal	Diodes Incorporated	DMP3036SFV-13	DigiKey	2	0.55	1.1
MOSFET Gate Treiber N-Kanal	Nexperia USA Inc.	2N7002NXAKR	DigiKey	2	0.09	0.18
Battery Protection IC	Texas Instruments	BQ296907DSGR	Digikey	1	0.33	0.33
Steckverbinder Pauschale				2	1.5	3
Kleinmaterial (R's, C's, L's, ...)				1	3	3
Gehäuse			3D Druck, leicht			0
Versandpauschale				1	10	10
SUMME Battery Pack + 10%						83.56

Tabelle 10: Kostenschätzung des Akku-Pakets 12V Variante mit Kaufteilen

Beschreibung	Hersteller	Hersteller Nr.	Distributor	n	CHF	$\sum CHF$
Ladegerät	Swaytronic	7640159368274	Swaytronic.ch	1	73.15	73.15

Tabelle 11: Kostenschätzung des Ladegeräts 12V Variante mit Kaufteilen

Konzept 3 - 12V LiPo Akku - Kaufteil mit weniger Kapazität

Dieses Akkupaket besitzt eine geringere Kapazität. Für eine Betriebsdauer von $\approx 20min$ würde ein Energiegehalt von $19.20Wh$ bereits ausreichen. Für einen 4S-Akku sind das in etwa $1300mAh$.

Akkuzelle Das Akkupack *SWAY-FPV LiPo 4S 14.8V 1300mAh 95C/190C XT60* ist mit 167g und 23,75 CHF ein bezahlbares und vergleichsweise leichtes Modell mit ausreichender Akkukapazität.

Kostenschätzung Die Akkuzelle beläuft sich auf den in Tabelle 12 berechneten Preis. Die Kosten für das Ladegerät können Tabelle 11 entnommen werden.

Beschreibung	Hersteller	Hersteller Nr.	Distributor	n	CHF	\sum CHF
Batteriepaket	Swaytronic	7640182620585	Swaytronic.ch	1	23.75	25.55
MOSFET Leistungsschalter P-Kanal	Diodes Incorporated	DMP3036SFV-13	DigiKey	2	0.55	1.1
MOSFET Gate Treiber N-Kanal	Nexperia USA Inc.	2N7002NXAKR	DigiKey	2	0.09	0.18
Battery Protection IC	Texas Instruments	BQ296907DSGR	Digikey	1	0.33	0.33
Steckverbinder Pauschale				2	1.5	3
Kleinmaterial (R's, C's, L's, ...)				1	3	3
Gehäuse			3D Druck, leicht			0
Versandpauschale				1	10	10
SUMME Battery Pack + 10%						45.67

Tabelle 12: Kostenschätzung des Akku-Pakets 12V Variante mit Kaufteilen

Fazit und Entscheidung der Konzeptphase

Tabelle 13 zeigt einen Vergleich der verschiedenen Konzepte, um sie direkt miteinander gegenüberstellen zu können.

Kriterium	Konzept 1	Konzept 2	Konzept 3
Enthaltene Energie [Wh]	42.2	44.4	32.56
Betriebsdauer [min]	40	45	30
Bordnetzspannung [V]	12	12	12
Maximaler Entladestrom [A]	8	105	77
Gewicht [g]	310	300	235
Kosten Akkupack [CHF]	40	83	66
Kosten Ladegerät [CHF]	60	73	73
Kosten Total [CHF]	100	156	139

Tabelle 13: Vergleichstabelle der verschiedenen Konzepte

In einer gemeinsamen Sitzung ist der Entscheid auf das Konzept 3 gefallen. Hauptargumente sind der etwas reduzierte Entwicklungsaufwand, der günstige Preis für eine angemessene Kapazität sowie ein geringes Gewicht. Bei eben dieser Besprechung hat sich ebenfalls der Wunsch an eine Möglichkeit ergeben, den Roboter auch über ein externes Netzteil speisen zu können. Dadurch entschärft sich ebenfalls die geringere Akkukapazität, da das Roboter bei Einrichttätigkeiten oder Versuchen nicht zwingend vom Akkubetrieb abhängig sein wird. Es kann in solchen Fällen dann auch stationär an einem Arbeitsplatz über ein Netzkabel betrieben werden.

Entwicklung und Dimensionierung

Entwickelt wird eine Schaltung, welche die Akkuspannung im Netz verteilen kann, sowie eine gewisse Schutzbeschaltung für den Akku bietet. Nachfolgend sind Anforderungen an diese Schaltung aufgeführt.

Anforderungen

Schutz vor Über-/ Unterspannung Die verwendete Akkutechnologie kann in Brand geraten, falls der Akku ausserhalb seines sicheren Arbeitsbereichs betrieben wird. Eine entsprechende Schutzbeschaltung muss also zwingend vorgesehen werden.

Quellenumschaltung Die Schaltung muss automatisch, sobald ein Netzteil angehängt wird, die Batterie vom Bordnetz trennen. Die verwendete Spannungsquelle kann optisch über eine LED signalisiert werden.

Einschaltstrombegrenzung Im gesamten Schaltkreis befinden sich Kapazitäten. Der Einschaltstrom muss daher begrenzt werden.

Akkuzellenüberwachung Aus der Anforderungsliste des Gesamtgeräts ergibt sich, dass die einzelnen Zellen überwacht werden sollen.

Ein- und Abschalten Über einen Schalter muss die Spannungsversorgung des Geräts abgeschaltet werden können. Der Betriebszustand kann optisch über eine LED dargestellt werden.

Anzahl der Ausgänge Die Schaltung muss genügend Ausgänge bieten, um jegliche Elektronik auf dem Roboter anschliessen zu können. Dabei soll ein in der Industrie üblicher, einfacher Steckverbinder eingesetzt werden.

Verpolungsschutz Der Anschluss soll einen Verpolungsschutz sowohl beim Netzteilanschluss, als auch beim Batterieanschluss bieten. Verpolte Anschlüsse sind anzuzeigen, dies kann zum Beispiel optisch über eine LED erfolgen.

Schaltungsbeschreibung

Batterieschutz Als Batterieschutz-IC wird der *BQ7790400PWR* der Firma Texas-Industries eingesetzt. Dieser bietet die Funktion, bei detekterter Unterspannung einer Batteriezelle die Batterie über einen MOSFET, dargestellt in Abbildung 44 vom Bordnetz zu trennen. Im Schema schaltet der IC *U401* also die beiden MOSFETS *Q203* und *Q205*. Zweiterer ist lediglich dafür da, eine LED zu schalten, falls eine Unterspannung detektiert wird. Der Ausgang des ICs kann nicht sonderlich viel Strom treiben, weshalb für diese LED ebenfalls ein MOSFET eingesetzt wird. Widerstandswerte und Kapazitäten wurden weitestgehend aus dem Datenblatt übernommen.

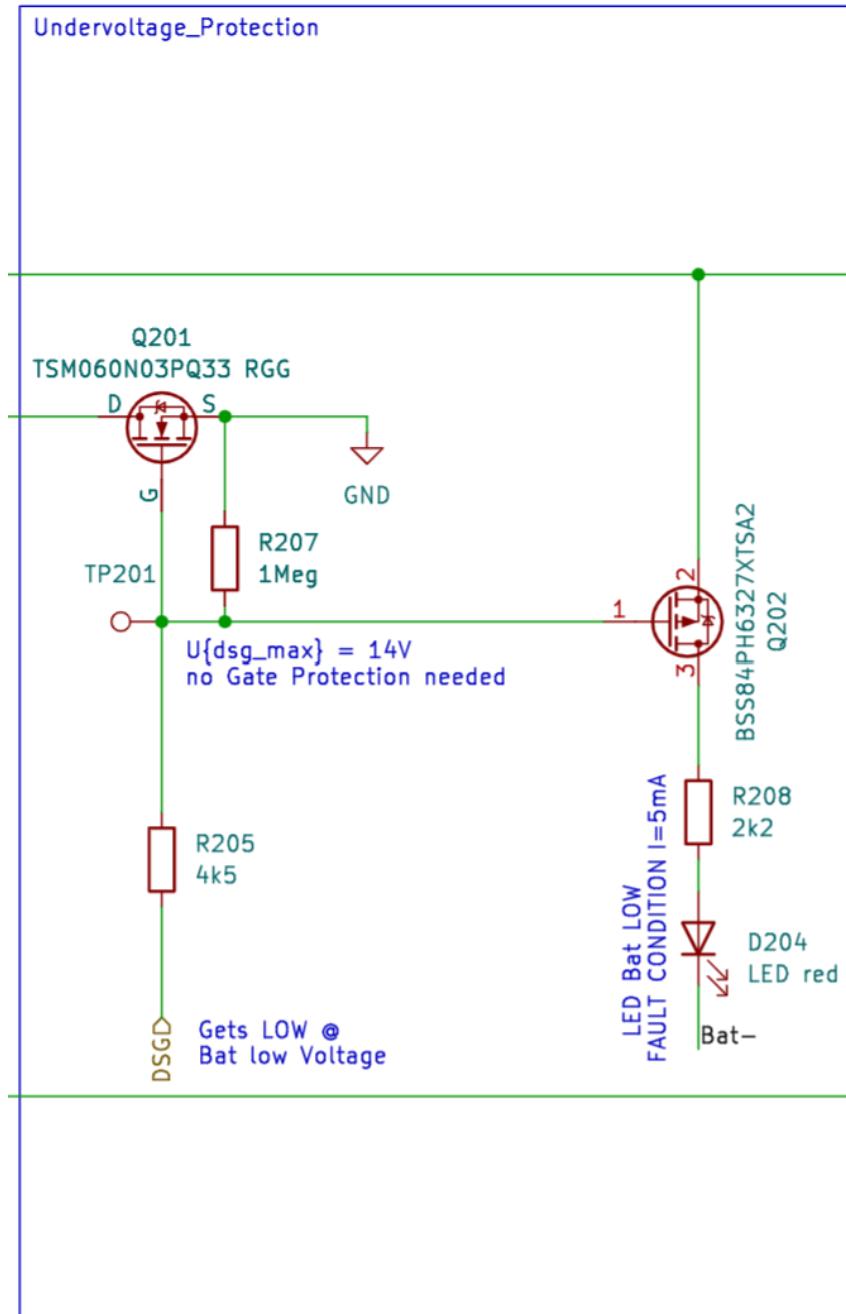


Abbildung 44: MOSFETS für Batterietrennung

Quellenumschaltung Beide Spannungsversorgungen, das Akku-Paket und ein externes Netzteil, werden über Dioden geführt ($D201 \& D202$), was verhindert, dass Strom aus dem Netzteil in die Batterie gelangen kann. Diese Schaltung wird in Abbildung 45 gezeigt. Der MOSFET $Q201$ hat die Aufgabe, die Batterie vom Netz zu trennen, sobald ein Netzteil angeschlossen ist.

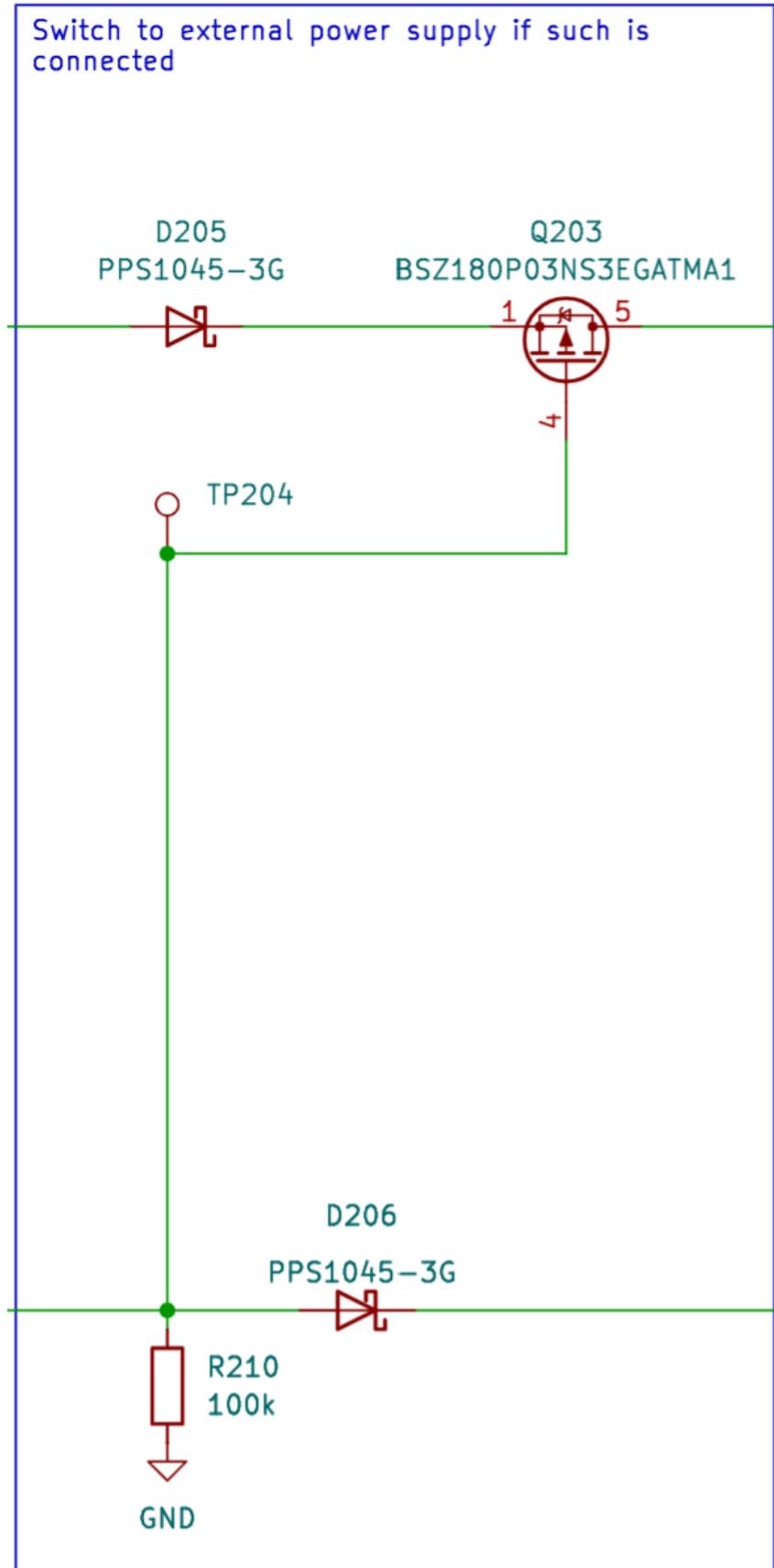


Abbildung 45: Quellenumschaltung

Einschaltstrombegrenzung Innerhalb des Roboters befinden sich viele Kapazitäten, alleine schon aufgrund der beiden Motorentreiber. Die Gesamtkapazität wird dabei auf $C_l = 1\mu F$ angenommen. Um den Stromstoss beim Einschalten des Geräts zu begrenzen, muss der Spannungsanstieg im System begrenzt sein. Daraus ergibt sich durch den Zusammenhang $I_{C_{Last}} = \frac{dU}{dt} \cdot C_{Last}$ direkt der Stromstoss durch das System.

Die Schaltung, welche im Anschluss diskutiert wird, ist in Abbildung 46 gezeigt.

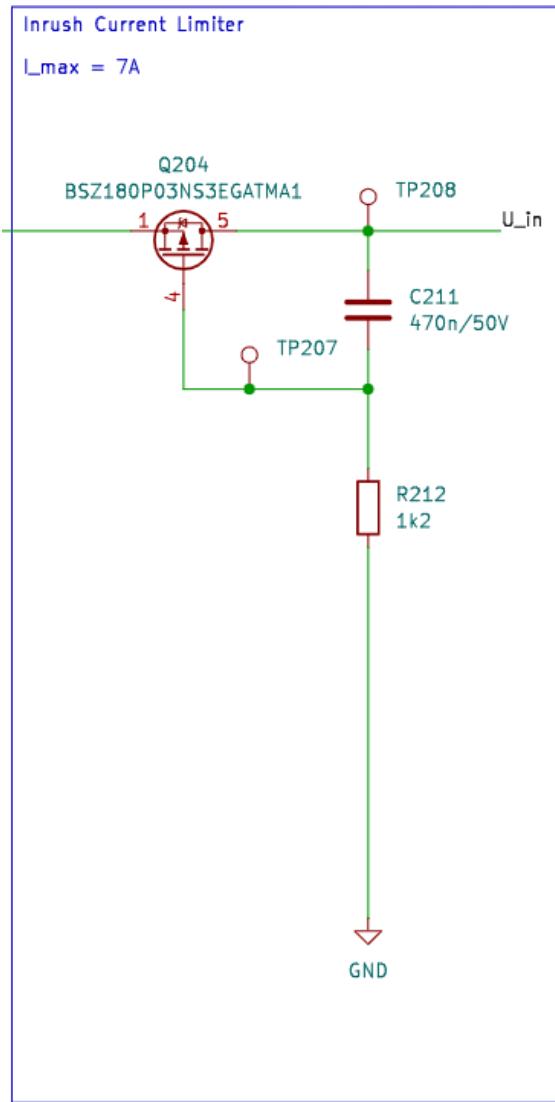


Abbildung 46: Inrush Limiter Schaltung

Der Spannungsanstieg beim Einschalten des MOSFETS $Q204$ bezweckt einen Strom durch Kondensator $C210$. Durch den Zusammenhang

$$I_{C_{210}} = \frac{dU}{dt} \cdot C_{210}$$

folgt, dass sich über dem Widerstand R_{204} eine Spannung einstellt, welche proportional zum Spannungsanstieg des Systems ist.

$$U_{R_{204}} = I_{C_{210}} \cdot R_{204}$$

Es ist richtig anzunehmen, dass sich die Spannung über dem Widerstand auf die Spannung U_{th} des MOSFETS einstellt, da er sich bei diesem Wert beginnt, 'selbst abzustellen'. Aus den beiden Formeln folgt:

$$\frac{dU}{dt} = \frac{U_{th}}{C_{210} \cdot R_{204}}$$

und daraus wiederrum

$$I_{inrush} = \frac{U_{th}}{C_{210} \cdot R_{204}} \cdot C_{Last}$$

Es wird nachfolgend mit dem gleichen MOSFET gerechnet, wie er bereits bei der Quellenumschaltung eingesetzt wird. Der maximal erlaubte Strom durch den Widerstand R_{204} wird mit $10mA$ angenommen, wodurch ein Widerstand von $1.2k\Omega$ resultiert. Der Inrush-Strom soll auf $7A$ begrenzt sein bei einer Treshhold-Spannung von $2.5V$. Daraus ergibt sich $C_{210} = 470nF$.

Im Anschluss muss noch die Verlustleistung des MOSFETS untersucht werden. Eingesetzt wird der *BSZ180P03NS3EGATMA1* P-Kanal-FET von Infineon Technologies. Der Impuls wird stark vereinfacht mit der Annahme betrachtet, dass der Stromimpuls während der gesamten Prozessdauer $I_{inrush,max} = 7A$ bei $U = 12V$ beträgt.

Durch den Zusammenhang

$$I_{C_{Last}} = \frac{dU}{dt} \cdot C_{Last}$$

lässt sich der Ladevorgang auf $t \approx 2.6ms$ abschätzen. Ein Blick in das Datenblatt des Herstellers zeigt, dass sich der FET trotz dieser Worst Case Annahme noch vollkommen in einer SOA befindet und bedenkenlos eingesetzt werden kann.

DC-DC Converter Eingesetzt wird der Buck-Konverter *LM2677* der Firma *Texas-Industries*. Es ist ein sehr günstiger Abwärtssteller, welcher 60W Leistung liefern kann. Zur Dimensionierung der Kapazitäten, der Induktivität und der Diode wurde sich auf die Dimensionierungsempfehlung des Datenblatts berufen. Der entsprechende Ausschnitt des Schemas ist in Abbildung 47 gezeigt.

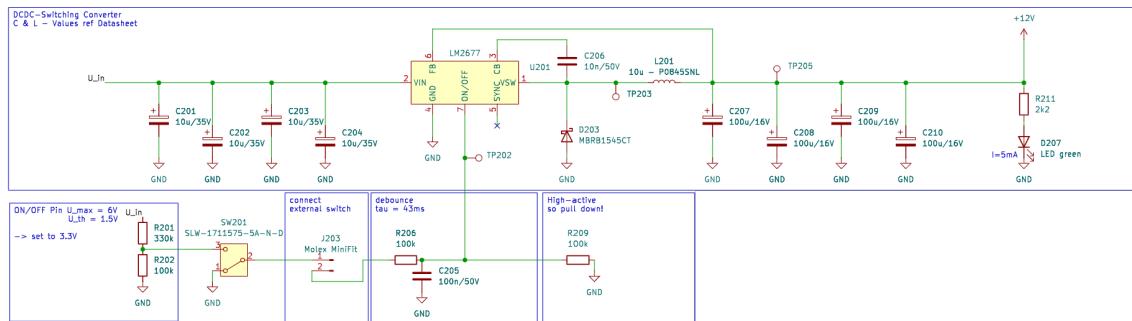


Abbildung 47: DC/DC Konverter 12V, 60W

ADC - Battery-Monitoring Aus der Anforderungsliste ergab sich der Wunsch nach der Möglichkeit, die Batterie überwachen zu können. Dazu wird ein via I^2C ansprechbarer ADC verbaut, welcher den Strommesswiderstand, den NTC sowie die Gesamtspannung des Batterieschutz-ICs überwachen kann. Die entsprechende Schaltung ist in Abbildung 48 dargestellt. Um den single-ended I^2C Signalen einen eindeutigen Rückweg bieten zu können, um Störungen zu vermeiden, muss das GND des Navigationscomputers ebenfalls mit auf das Powerboard geführt werden. Dadurch entsteht allerdings die Gefahr, dass

sich ungewollte GND-Schleifen im gesamten System bilden. Um dieses Problem zu umgehen, wird ein I^2C Isolator der Firma *Texas Industries* eingesetzt. Das setzt voraus, dass sich auf dem Powerboard ebenfalls eine 3V3-Spannungsquelle befindet. Diese wird mit einem LDO zur Verfügung gestellt. Es ist mit nur sehr geringen Strömen zu rechnen, daher ist nicht mit hoher Verlustleistung über den LDO zu rechnen und dieser kann die 12V direkt in die benötigten 3V3 umwandeln.

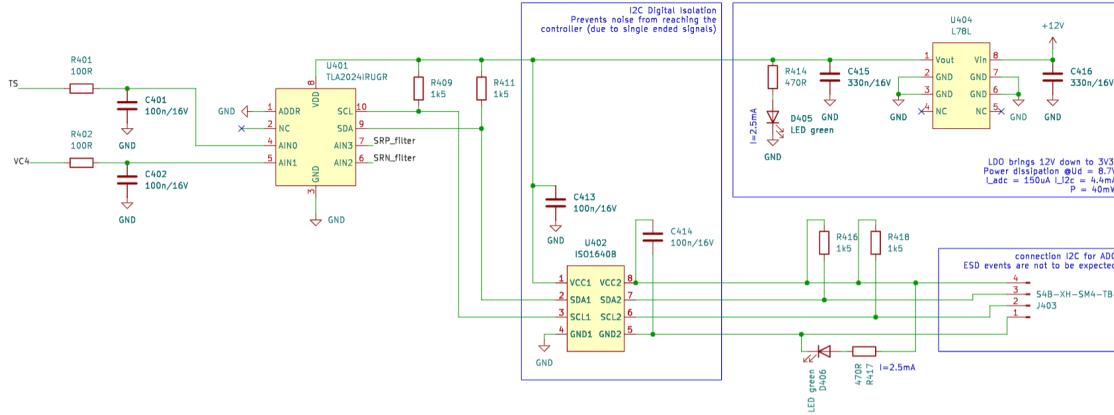


Abbildung 48: ADC zur Batterieüberwachung

Steckverbinder Jede Funktion bringt einen eigenen Steckverbinder mit, um Verwechslungen und falsches Einsecken zu vermeiden.

D. Konzepterstellung Antriebe

Dieser Abschnitt befasst sich mit den Antrieben des Pfadfinders. Dabei werden sowohl verschiedene Antriebskonzepte diskutiert als auch potentiell einsetzbare Antriebe ausgewählt. Abschliessend wird konzeptionell zusammengefasst, wie diese Antriebe angesteuert werden sollen.

Anforderungen

Beschleunigung Eine ehrgeizige Anforderung an den Roboter ist es, ihn innerhalb einer Sekunde auf eine Geschwindigkeit von $2\frac{m}{s}$ beschleunigen zu können.

Gewicht Das Gewicht ist, wie bei allen anderen Baugruppen, ein sehr kritischer Punkt bei der Entwicklung des Pfadfinders. Als *Gewichtsbudget* wurde für die Antriebseinheit festgelegt, dass das Gewicht nicht mehr als insgesamt 500g betragen darf.

Kosten Das Budget für dieses Projekt ist sehr begrenzt. Der Antrieb kann ein wesentlicher Kostenfaktor sein, daher wurde für den Antrieb inklusive Steuerung und Räder ein Budget von 100CHF festgelegt.

Nennspannung Die benötigte Nennspannung des Bordnetzes hängt weitgehend von der benötigten Spannung der Motoren ab. Wie bereits im Kapitel C erwähnt, würde eine Spannung von 24V ein nicht realisierbares Gewicht darstellen. Daher liegt die maximale Spannung für Motoren hier bei 12V Nennspannung.

Technik Die Ansteuerung der Motoren ist mit geeigneten Treibern in den meisten Fällen relativ einfach zu realisieren. Aufgrund vorhandener Erfahrungswerte sollten vorzugsweise Schrittmotoren eingesetzt werden.

Schnittstelle Die Schnittstelle zur Ansteuerung der Motoren sollte einem einfach zu realisierenden Standard entsprechen. Dies betrifft die Ansteuerung über bekannte Busprotokolle wie I^2C , SPI, UART oder auch PWM- oder Step/Dir-Schnittstellen.

Die Anforderungen an die Batterie schliessen aufgrund der 12V Bordnetzspannung bereits industrietaugliche 24V Motoren aus. Diese sind zwar sehr robust, aber in den meisten Fällen auch sehr schwer.

Leistungsanforderungen an Motoren

Um die Leistungsanforderungen der Motoren zu bestimmen, wurde das Beschleunigungsmoment berechnet, da während der Beschleunigung die grösste Last auf die Motoren wirkt. Abbildung 49 zeigt die zur Berechnung verwendete Formel, die aus einer Maschinenbau-Formelsammlung entnommen wurde.

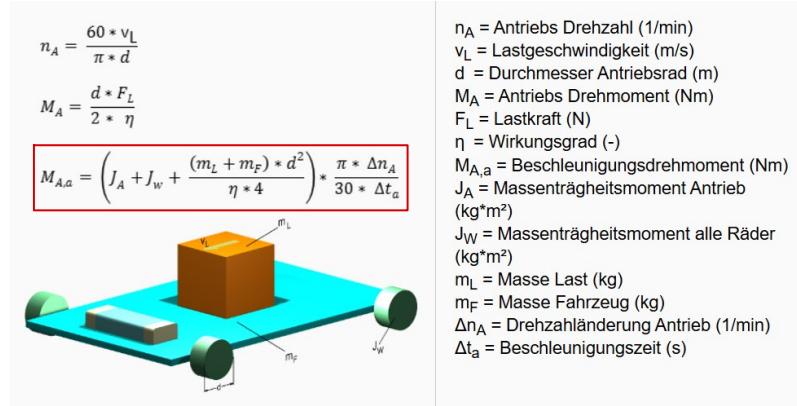


Abbildung 49: Berechnungsformel aus Maschinenbau Formelsammlung

Die Berechnungen wurden für alle evaluierten Motoren durchgeführt und dienten der Überprüfung, ob ein bestimmter Motor das erforderliche Drehmoment für die Anwendung liefern kann. Als Beispiel ist auf der folgenden Seite die Berechnung für den FIT0278-Schrittmotor von DFRobot aufgeführt.

Annahmen:

Beschleunigung:

$$a = 10 \frac{\text{m}}{\text{s}^2}$$

Beschleunigungsduer:

$$t_{\text{Beschleunigung}} = 0.2 \text{ s}$$

Wirkungsgrad des Antriebs:

$$\eta_{\text{Antrieb}} = 0.8$$

Max. Motorenendrehzahl (keine Herstellerangabe im Datenblatt):

$$n_{\text{Motor}} = 400 \text{ min}^{-1}$$

Gegeben:

Fahrzeuggewicht:

$$m_{\text{Fahrzeug}} = 2 \text{ kg}$$

Raddurchmesser:

$$d_{\text{Rad}} = 0.08 \text{ m}$$

Masse pro Rad:

$$m_{\text{Rad}} = 0.04 \text{ kg}$$

Trägheitsmoment Motor:

$$I_{\text{Motor}} = 3.5 \cdot 10^{-6} \text{ kg} \cdot \text{m}^2$$

Erdbeschleunigung:

$$g = 9.81 \frac{\text{m}}{\text{s}^2}$$

Max. Drehmoment:

$$M_{\text{Motor, max}} = 0.343 \text{ N} \cdot \text{m}$$

Berechnungen:

Massenträgheitsmoment (zwei Räder):

$$I_{\text{Rad}} = 2 \cdot \frac{1}{2} \cdot m_{\text{Rad}} \cdot \left(\frac{d_{\text{Rad}}}{2} \right)^2 = (6.4 \cdot 10^{-5}) \text{ kg} \cdot \text{m}^2$$

Maximale Geschwindigkeit:

$$v_{\text{max}} = n_{\text{Motor}} \cdot d_{\text{Rad}} \cdot \pi = 1.676 \frac{\text{m}}{\text{s}}$$

Beschleunigungsmoment (worst case):

$$M_{\text{max}} = \left(I_{\text{Motor}} + I_{\text{Rad}} + \frac{m_{\text{Fahrzeug}} \cdot d_{\text{Rad}}^2}{\eta_{\text{Antrieb}} \cdot 4} \right) \cdot \frac{\pi \cdot n_{\text{Motor}}}{t_{\text{Beschleunigung}}} = 0.426 \text{ N} \cdot \text{m}$$

Drehmoment pro Motor:

$$M_{\text{Motor}} = \frac{M_{\text{max}}}{2} = 0.213 \text{ N} \cdot \text{m}$$

Konzeption

Ausgehend von den Leistungsanforderungen an die Motoren kann am Markt recherchiert werden, welche Motoren potenziell verfügbar sind. Diese können wiederum nach den Kriterien Kosten, Gewicht, Leistung und Entwicklungsaufwand miteinander verglichen werden.

Konzept 1 - Bipolarer Schrittmotor

Die Wahl fällt auf einen bipolaren Schrittmotor, da dieser bei gleichem Gewicht eine bessere Ausnutzung der Spulen und damit eine höhere Leistung bei gleichem Gewicht im Vergleich zu einem unipolaren Schrittmotor ermöglicht. Ein Vergleich verschiedener Schrittmotoren hat gezeigt, dass Schrittmotoren, die in einem Nennstrombereich von 1.2A bis 2A liegen, in einem Bereich liegen, der die Leistungsanforderungen bereits ohne Zwischengänge erfüllen kann. Typische, kostengünstige Vertreter dieser Reihe sind die in Tabelle 14 aufgeführten.

Hersteller	Herst.Nr	Distributor	I_{nenn} [A]	U_{nenn} [V]	Preis [CHF]	Gewicht [kg]
Olimex LTD	SM-42HB34F08AB	DigiKey.ch	1. 33	12	9.36	0.400
DFRobot	FIT0278	DigiKey.ch	1. 7	3.4	12	0.269
SparkFun Electronics	ROB-10846	DigiKey.ch	1. 7	3	16.77	0.356

Tabelle 14: Verschiedene Schrittmotoren

Dieser Vergleich zeigt, dass sich eigentlich nur der Motor *FIT0278* von *DFRobot* in einem akzeptablen Verhältnis von Gewicht und Leistung befindet. Alle aufgeführten Motoren sind preisgünstig. Die Motoren müssen über gekaufte Treiber angesteuert werden. Die Eigenentwicklung eines Motortreibers ist zwar mit wenig Aufwand zu realisieren, fertige Treiberendstufen sind jedoch sehr preisgünstig und einfach anzusteuern. Bei der Ansteuerung des Schrittmotorentreibers *TMC5240* von *ADI-Trinamic* kann ein Teammitglied auf Erfahrungen aus seinem beruflichen Umfeld zurückgreifen. Diese Treiber stehen auch für dieses Projekt in zweifacher Ausführung zur Verfügung. Der Vergleich dieses Treibers mit einem einfach anzusteuernden Treiber zeigt die Tabelle 15.

Hersteller	Herst.Nr	Distributor	I_{nenn} [A]	U_{nenn} [V]	Preis [CHF]	Gewicht [kg]
ADI	TMC5240-EVAL	Komax AG	2	36	63	0.036
ACT Motor	ACT DM430	reichelt.ch	3	32	17.42	0.180

Tabelle 15: Verschiedene Schrittmotorentreiber

Konzept 2 - Radnabenmotor

Auf dem Markt sind fertige Radnaben erhältlich, die zum Teil bereits sowohl Endstufen als auch eine Steuerung enthalten:

Der gezeigte Motor ist einfach über den UART-Bus anzusteuern, kostengünstig, leicht und hat einen integrierten Encoder. Ein grosser Nachteil ist jedoch, dass mit diesem Motor das ehrgeizige Ziel einer Beschleunigung von $2 \frac{m}{s^2}$ voraussichtlich nicht erreicht werden kann.

Hersteller	Herst.Nr	Distributor	I_{nenn} [A]	U_{nenn} [V]	Preis [CHF]	Gewicht [kg]
DFRobot	FIT1001	DigiKey.ch	0.5	14.4	25.71	0.216

Tabelle 16: Radnabenmotor DFRobot

Konzept 3 - BLDC-Motor

BLDC-Motoren haben den grossen Vorteil, dass sie eine hohe Leistung bei geringem Gewicht erreichen können. Allerdings ist ihr Drehmoment eher gering, weshalb ein Zwischengetriebe notwendig ist. Dieses Zwischengetriebe erhöht das Gewicht des Antriebs zusätzlich. Ausserdem sind BLDC-Motoren oft teurer. In der folgenden Tabelle sind verschiedene Motoren aufgelistet, die in Frage kommen.

Hersteller	Herst.Nr	Distributor	I_{nenn} [A]	U_{nenn} [V]	Preis [CHF]	Gewicht [kg]
DFRobot	FIT0441	DigiKey.ch	0.7	12	17.11	0.070
Lin Engineering	BL17E19-01-RO	DigiKey.ch		24	79.18	0.320

Tabelle 17: Radnabenmotor DFRobot

Der Motor von DFRobot wird mit seiner Geschwindigkeit von maximal $159 \frac{1}{min}$ zwar nicht die gewünschte Drehzahl erreichen, aber das geringe Gewicht bei entsprechender Leistung sowie der bereits integrierte Treiber machen diesen Antrieb dennoch wert, im Detail evaluiert zu werden. Generell hat die Komponentenrecherche ergeben, dass BLDC-Motoren eine höhere Nennspannung benötigen, als mit dem dimensionierten Akkupack zur Verfügung gestellt werden kann. Als Stromquelle dienen natürlich Motortreiber, jedoch werden Motoren im entsprechend benötigten Leistungsbereich schnell zu teuer und zu schwer. Generell können BLDC-Motoren, ähnlich wie Schrittmotoren, über einen selbst entwickelten Treiber angesteuert werden, aber auch fertige Motorsteuerungen sind unschlagbar günstig und über eine PWM-Schnittstelle sehr einfach anzusteuern. Stellvertretend für diese Produktgruppe sei der folgende Motor genannt.

Hersteller	Herst.Nr	Distributor	I_{nenn} [A]	U_{nenn} [V]	Preis [CHF]	Gewicht [kg]
ACT Motor	BLDC-8015A-5	reichelt.ch	15	50	39.1	0.432

Tabelle 18: BLDC-Treiber

Fazit und Entscheid aus der Konzeptionsphase

In der Gruppe wurde beschlossen, den in Abbildung 51 dargestellten Schrittmotor von DFRobot in Kombination mit den in Abbildung 50 dargestellten Schrittmotorentreibern von Trinamic genauer zu analysieren und zu verfolgen. Als *Plan-B*, auch für den Fall, dass das gewünschte Gewicht nicht eingehalten werden kann, soll der Radnabenmotor, ebenfalls von DFRobot, evaluiert werden. Damit können zwar nicht unbedingt die gewünschten Geschwindigkeits- und Beschleunigungswerte erreicht werden, aber die Ansteuerung ist sehr einfach, die Treiber sind bereits integriert und zudem sind sie leicht.

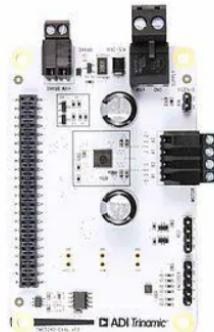


Abbildung 50: TMC 5240 Evaluation-Board



Abbildung 51: DFROBOT FIT0278 Schrittmotor

Dieser Motor soll durch einen echtzeitfähigen Mikroprozessor geregelt werden, dem auch die Sensordaten des Liniensensors zur Verfügung stehen. Der Antrieb soll auf das Feedback dieses Sensors geregelt werden. Dies ist in Abbildung 52 nochmals verdeutlicht.

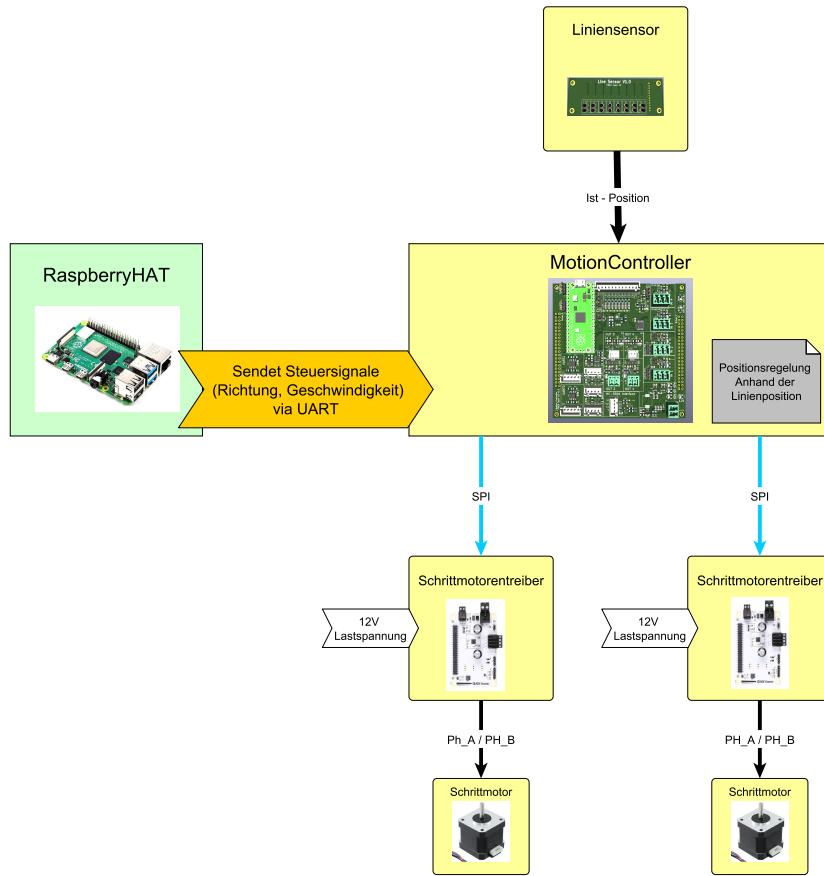


Abbildung 52: Konzept für die Ansteuerung der Schrittmotoren

Zusätzlich ist mindestens ein Encoder vorgesehen, der mit einer einfachen Lochscheibe und einer Gabellichtschranke realisiert ist. Dieser hat jedoch keine Anwendung in der Fahrzeugregelung. Er dient lediglich zur Erfassung der zurückgelegten Wegstrecke. Die ausgewählten Antriebe könnten direkt über den High-Level-Controller angesteuert werden. Die Antriebsregelung wurde jedoch bewusst auf einen Mikroprozessor verlagert, um dem Ansatz der Gewaltentrennung gerecht zu werden. So gibt es einen echtzeitfähigen Prozessor, der sich ausschliesslich um die Lageregelung kümmert, während der High-Level-Controller lediglich die Richtungsentscheidungen trifft.

Motion Controller

Für die Integration der Schrittmotortreiber in das Gesamtsystem wird eine eigene Leiterplatte entwickelt. Der komplette Schaltplan befindet sich im Anhang. Im Folgenden wird auf einige Baugruppen näher eingegangen.

Anforderungen

Spannungsversorgung Die Leiterplatte muss mit 12V versorgt werden können.

Digitale Ein- und Ausgänge Die Leiterplatte muss über digitale Ein- und Ausgänge mit Spannungsversorgung verfügen. Es müssen sowohl industrietaugliche 12V als auch

3V3 Sensoren betrieben werden können. Ausgänge sollen sowohl Lasten Schalten, aber auch logische Signale übertragen können.

Analoge Eingänge Der PCB muss in der Lage sein, mit 8 analogen Eingängen den Liniensensor auswerten zu können.

Kommunikationsschnittstellen Der Motion Controller muss 2 Punkt zu Punkt UART Kommunikation über eine RS422 Schnittstelle aufbauen können.

Motorentreiber Schnittstellen Die Motorentreiber sollen möglichst einfach angeschlossen werden können, es ist erwünscht, dass der MotionController einfach aufgesteckt werden kann.

Gyroskop Das PCB muss über eine Möglichkeit der Winkelerfassung verfügen.

HC-SR04 Schnittstelle Das PCB muss einen Ultraschallsensor auswerten können.

Debugfunktionen Einzelne Signale können zusätzlich auf LEDs gelegt werden, um sie für Debugzwecke sichtbar zu machen.

Baugrösse Die Platine darf maximal 100mm x 100mm gross sein, da dadurch massiv Kosten gespart werden können.

Schaltungsbeschreibung

Das Gesamte Schema befindet sich im Abgabedokument beigefügt als digitalen Anhang in Form eines eigenständigen pdf Dokuments.

Spannungsversorgung Die Leiterplatte wird mit einer Spannung von 12V versorgt. Diese 12V werden zunächst mit einem DC-DC-Wandler auf 5,5V heruntergeregt, bevor sie über 2 LDOs zunächst auf 5V und dann nochmals auf 3,3V heruntergeregt werden. Abbildung 53 zeigt genau diese Schaltung.

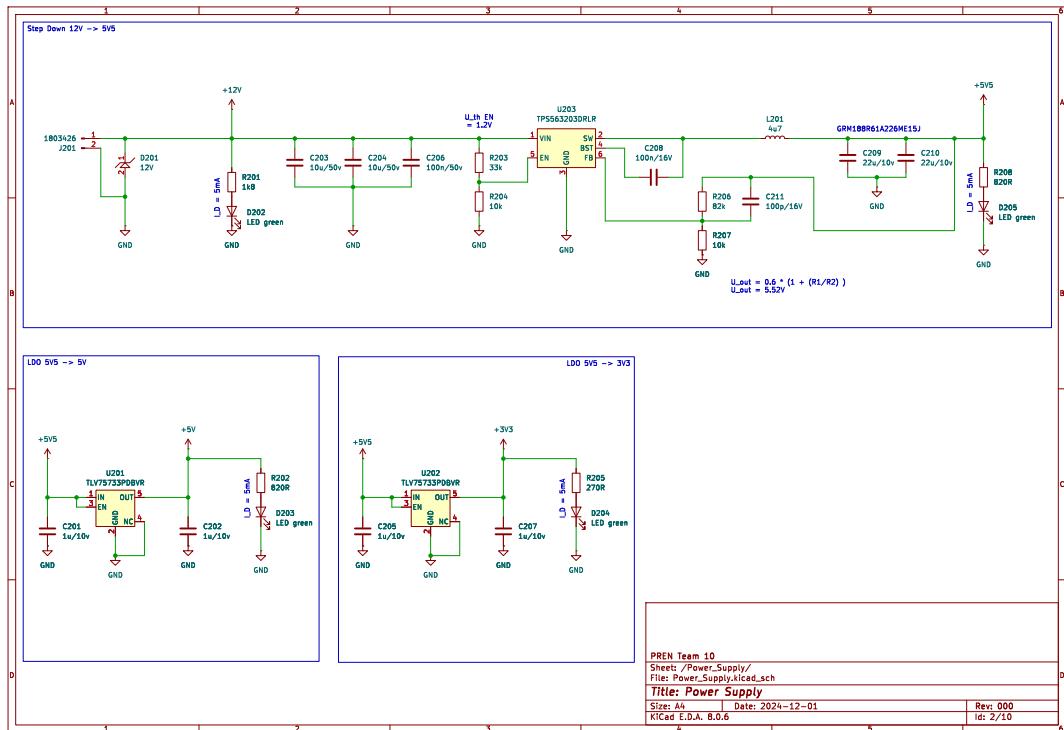


Abbildung 53: Auszug Schema Spannungsversorgung

Verbindungen an TMC5240 Die Abbildung 54 zeigt, wie die Motortreiber über 2 44-polige Steckerleisten einfach auf den Motion Controller aufgesteckt werden können. 2 LEDs zeigen die aktuell gewählte Strombegrenzung an.

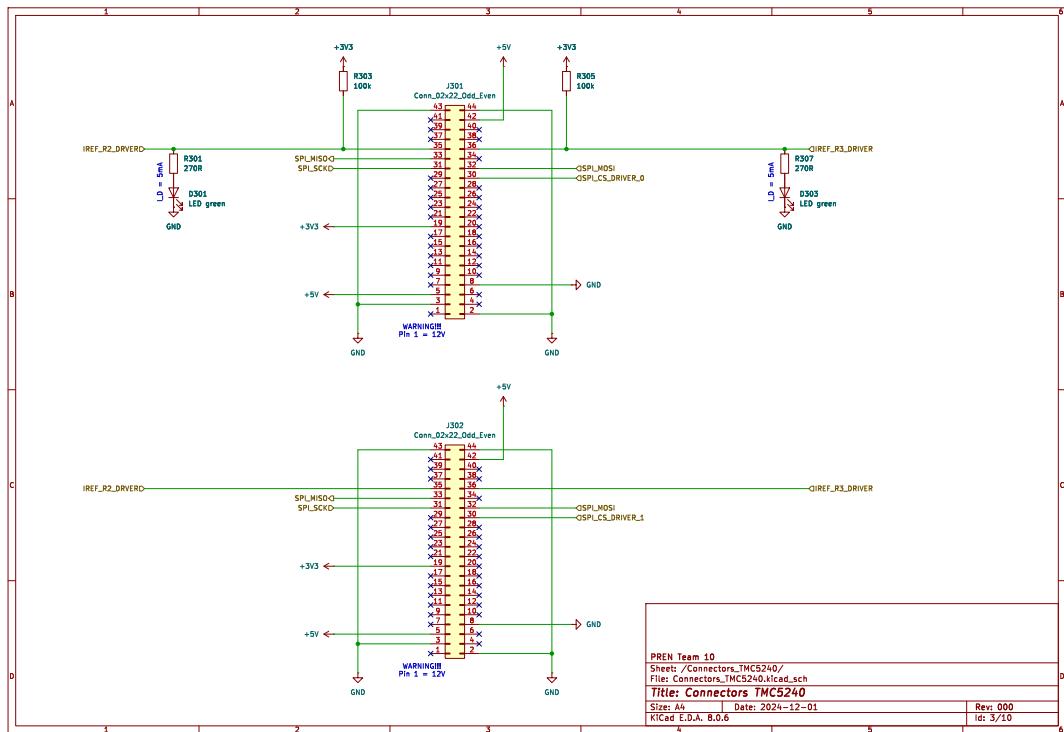


Abbildung 54: Auszug Schema TMC5240 - Anbindung

Kommunikationsschnittstellen Abbildung 55 zeigt den Anschluss der beiden Kommunikationskanäle *UART_0* und *UART_1* an zwei separate RS422-Schnittstellen. Prinzipiell sind UART-Kanäle Push-Pull-Stufen - allerdings sind diese Zustände beim Systemstart noch nicht zwingend definiert. Um Bit-Banging auf der Kommunikationsleitung zu vermeiden, sind diese Signale dennoch mit Pullup-Widerständen versehen.

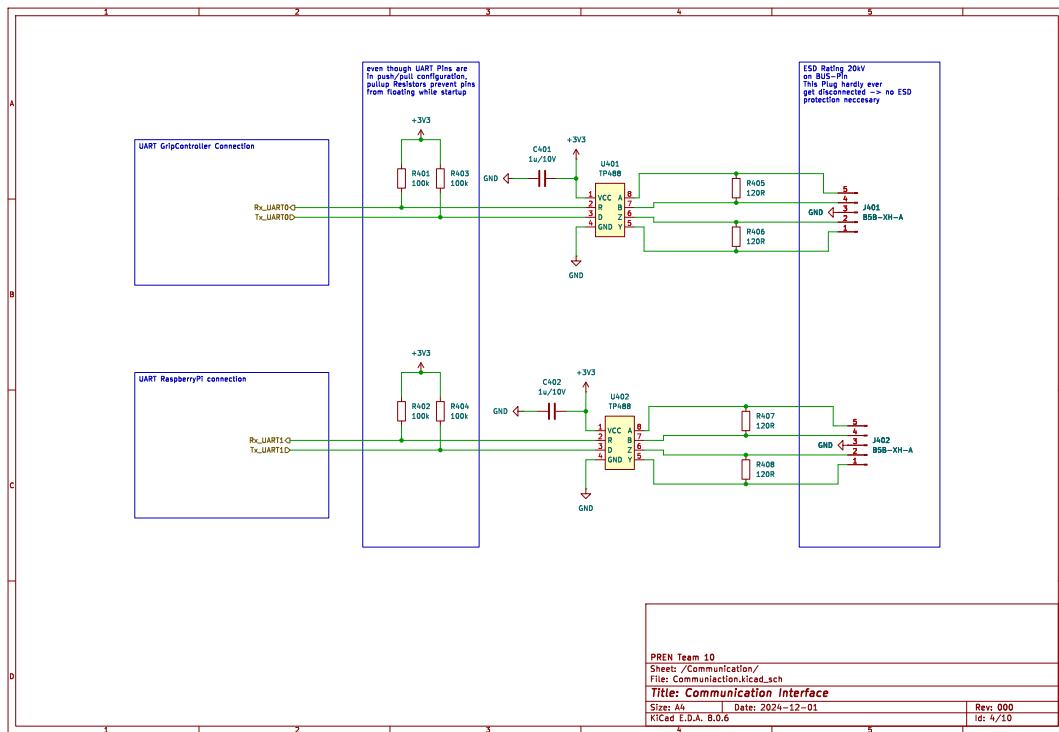


Abbildung 55: Auszug Schema Kommunikationsschnittstellen

Digital Eingänge Abbildung 56 zeigt einen digitalen Eingang des Motion Controllers. 2 der 4 Eingänge können sowohl mit 12V als auch mit 3V3 versorgt und geschaltet werden. Um den Eingang auf 3V3 umzukonfigurieren müssen lediglich die beiden Jumper am Eingang gesetzt werden, die anderen beiden digitalen Eingänge sind fest auf 12V konfiguriert. Das eingehende Signal wird über ein RC-Glied und einen Schmitt-Trigger entprellt. Eine LED zeigt an, dass der Eingang geschaltet ist.

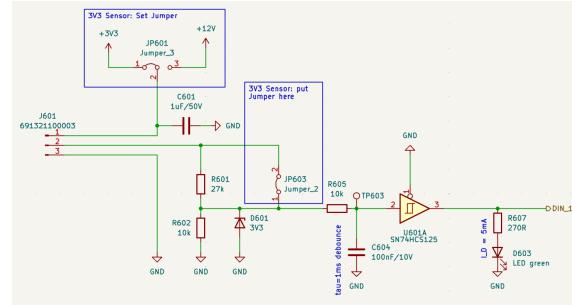


Abbildung 56: Auszug Schema Digitaler Eingang

Analoge Eingänge Abbildung 57 zeigt den Anschluss der Analogsignale an den Raspberry Pi Pico. Es wird ein I^2C -fähiger ADC mit 8 Eingängen verwendet. Die eingehenden Signale werden über einen RC-Tiefpass gefiltert. Weiterhin kann die Versorgungsspannung der Liniensor-Sensor-Sender ein- und ausgeschaltet werden. Dadurch ist es möglich, den Sensor bei Bedarf nur dann einzuschalten, wenn gerade gemessen wird. UV-Strahlung ist für das menschliche Auge schädlich und kann so zum Personenschutz beitragen.

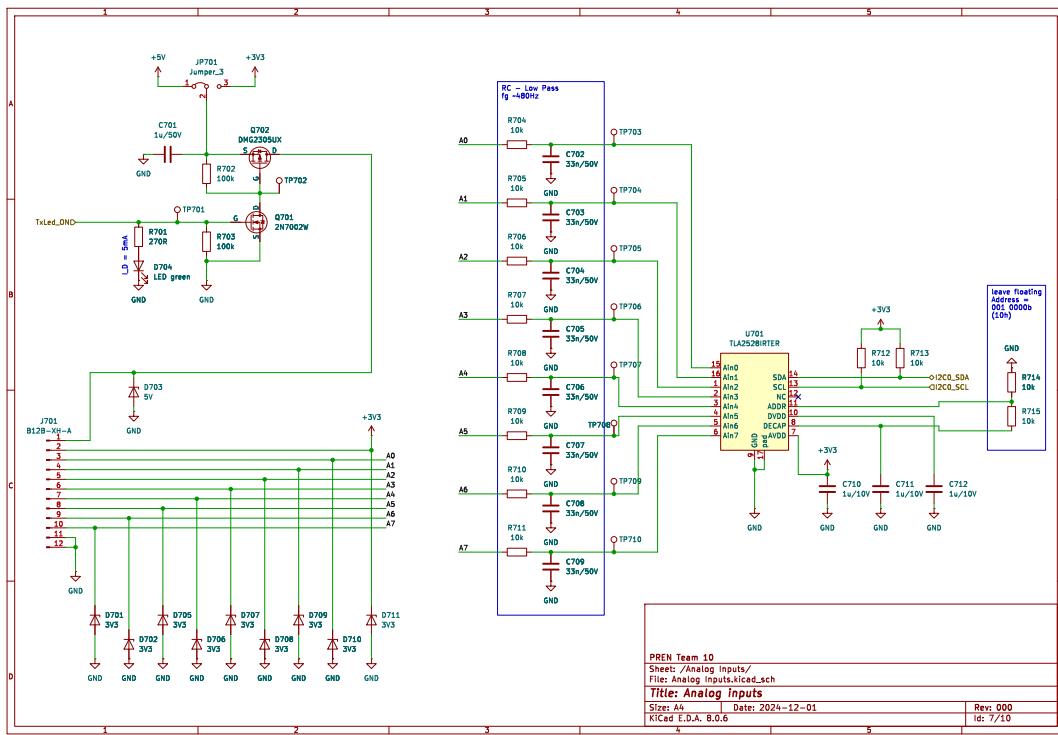


Abbildung 57: Auszug Schema Analogeingänge

Digitale Ausgänge Abbildung 58 zeigt die 4 digitalen Ausgänge des Motion Controllers. Über die beiden Ausgänge DOUT_1 und DOUT_2 können grössere Ströme mit jeweils 5V5 bzw. 12V geschaltet werden. Der gezeigte High-Side Switch erlaubt Ströme bis zu 3A. Die beiden rechten Ausgänge sind für logische Ausgangspegel vorgesehen. Über sie kann z.B. auch ein PWM-Signal ausgegeben werden. Die Pull-Down-Widerstände ermöglichen einen schnellen Übergang nach GND, wenn logische Rechtecksignale schnell geschaltet werden müssen.

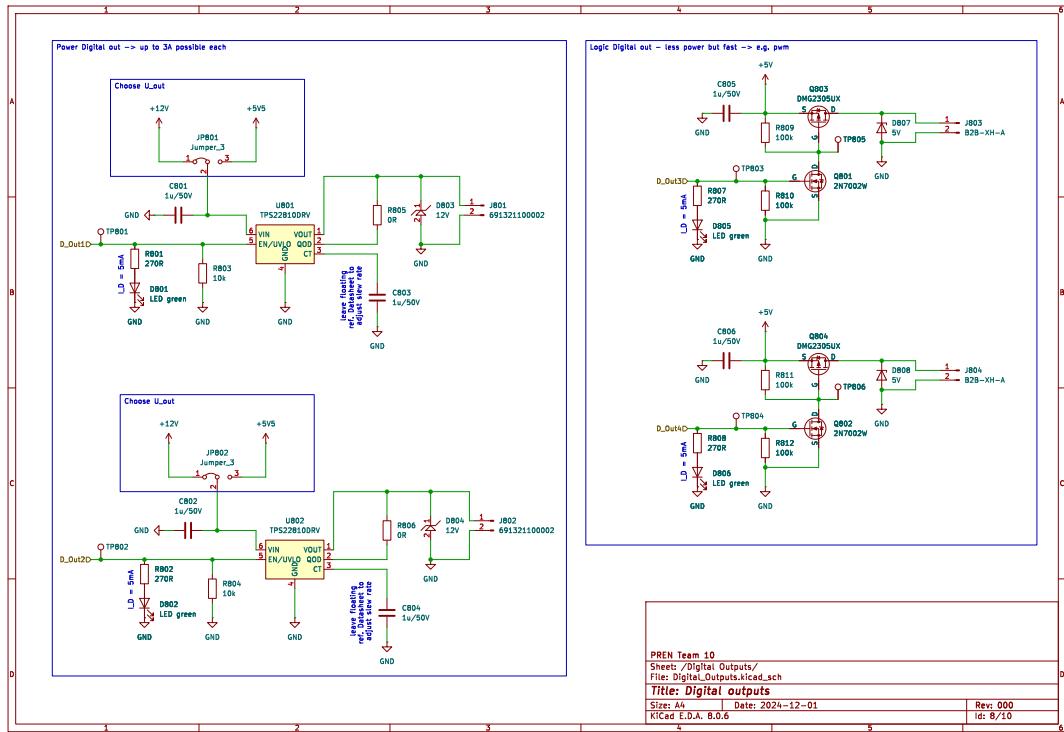


Abbildung 58: Auszug Schema Digitale Ausgänge

I²C Schnittstellen In der Abbildung 59 sind 3 I²C-Schnittstellen dargestellt. Der Motion Controller verfügt über 2 I²C-Schnittstellen, die auf einem 3V3-Pegel arbeiten und eine, die auf einem 5V-Pegel arbeitet. An diese Schnittstellen können Sensoren wie Lidar oder auch LCD Displays angeschlossen werden.

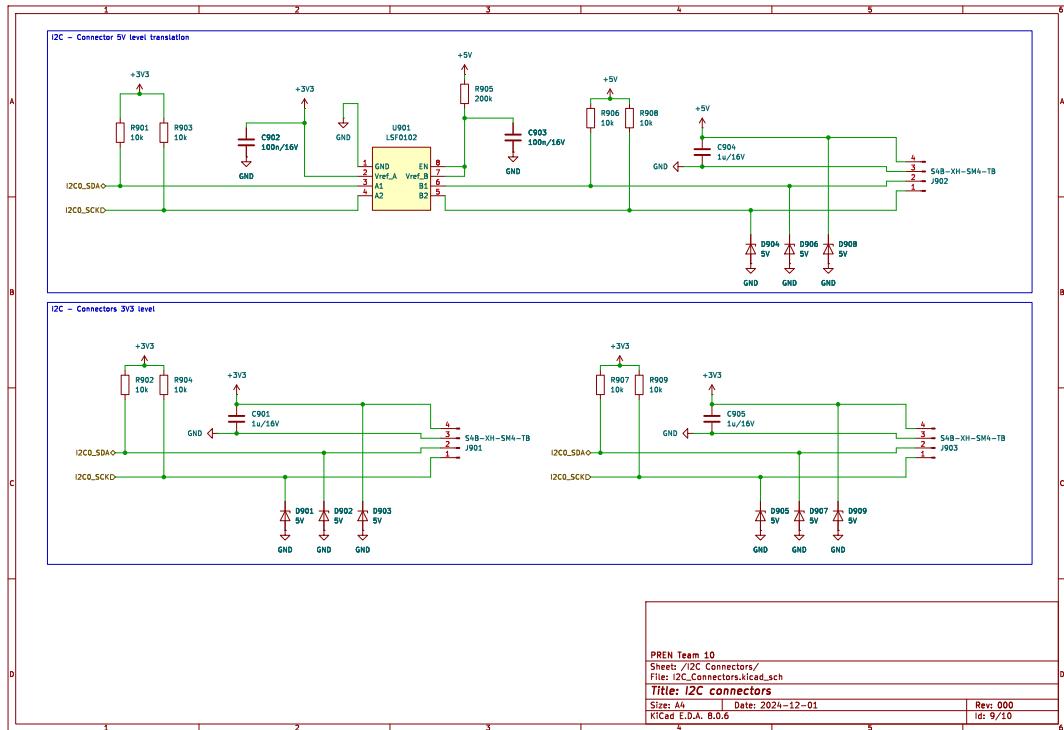


Abbildung 59: Auszug Schema I²C Verbindungen

Gyroskop und Ultraschallsensor Diese Schaltung ist in Abbildung 60 dargestellt. Das Gyroskop ist sehr rudimentär angeschlossen und verzichtet auf viele Funktionen, da es diese schlichtweg nicht benötigt. Falls wider Erwarten doch mit einem Interrupt gearbeitet wird, ist zumindest einer angeschlossen. Der Ultraschallsensor HC-SR04 arbeitet auf einem 5V-Pegel und wird über einen MOSFET angesteuert. Damit das Echo-Signal den Eingang nicht beschädigt, wird der Spannungsspeicher über einen Spannungsteiler auf 3,3V reduziert.

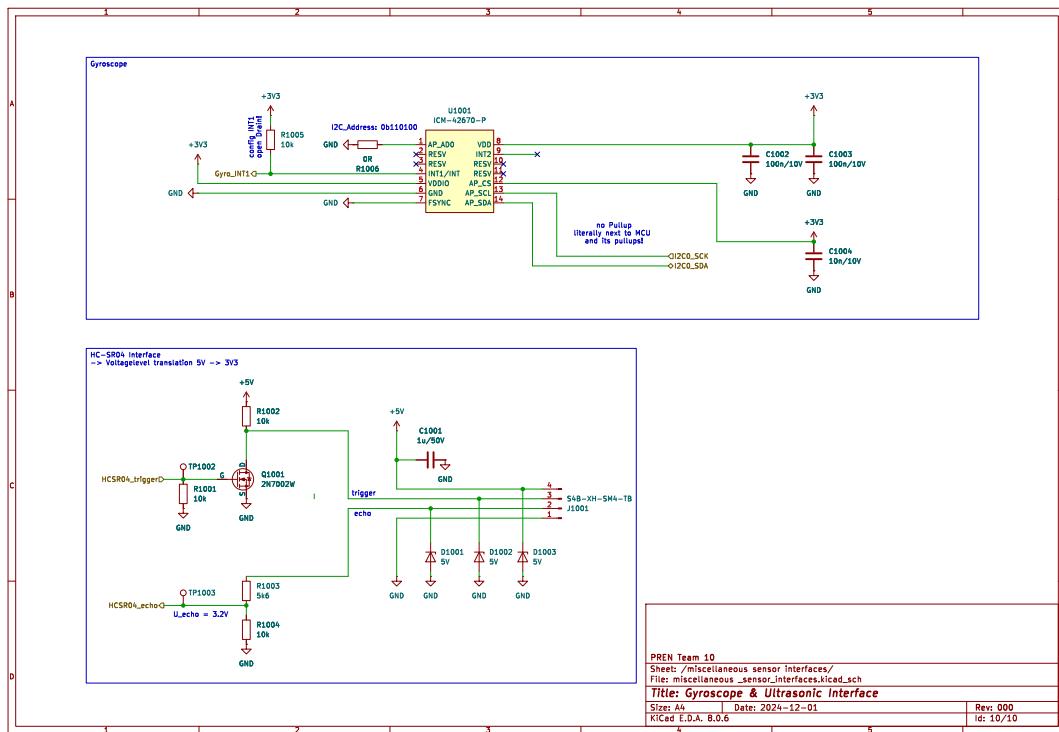


Abbildung 60: Auszug Schema Gyroskop und HCSR-04

E. Konzepterstellung Hinderniserkennung

Dieser Abschnitt beschäftigt sich mit der Evaluation verschiedener Distanzsensoren, welche die Anwesenheit eines Hindernisses detektieren sollen.

Anforderungen

Distanz Die Streckenlängen belaufen sich laut *FAQ* auf $0.5m \dots 2m$. Daher muss zur Pylonen-Erkennung ein Sensor gefunden werden, welcher in eben diesem Bereich plausible Werte zurückgeben kann. Bei Hindernissen wird davon ausgegangen, dass sich diese in etwa in der Mitte der Streckenlänge befinden. Ein entsprechender Sensor muss also auf mindestens $0.25m$ genau Distanzen detektieren können.

Genauigkeit Bei Pylonen muss lediglich die Anwesenheit detektiert werden, um entsprechende Wegstrecken als nicht befahrbar zu bewerten. Eine Genauigkeit von $\pm 50mm$ muss also ausreichend sein, um diese zu erkennen. Der Sensor, welcher Barrieren erkennen sollte, muss dagegen ein wenig genauer arbeiten, da das Fahrzeug zwecks Positionierung des Greifmechanismus gut ausgerichtet werden muss. Hier wird also eine Detektion mit einer Genauigkeit von $\pm 10mm$ angestrebt.

Gewicht und Baugrösse Das Gewicht ist in allen Fällen eine Einschränkung für jede Funktionseinheit. Für die Sensorik ist dies zwar kein gravierender Faktor, da entsprechende Bauteile in der Regel sehr leicht sind, nichtsdestotrotz gilt: Je kleiner und leichter, desto besser.

Kosten Das Budget für die Entwicklung des Pfadfinders ist begrenzt. Daher soll ein Budget von $30CHF$ nicht überschritten werden beim Zusammenstellen dieser Sensorik.

Konzeption

Erkennen von Hindernissen und Pylonen

Geplant ist, mit einem Sensor über die Hindernisse hinweg zu schauen, um so Pylonen zu erkennen. Ein zweiter Sensor, welcher sich etwas tiefer befindet, soll Hindernisbarrieren erkennen können. So wird über die Höhendifferenz zwischen Pylonen und Hindernissen unterschieden.

Ultraschallsensor HC-SR04

Der Ultraschallsensor HC-SR04 ist ein sehr einfacher, sehr günstiger Distanzsensor, welcher, wie der Name bereits anmuten lässt, als Sonar Distanzen detektieren kann. Seine Ansteuerung und Auswertung sind auf einem Mikroprozessor mit 2 GPIOs sehr einfach umzusetzen und mit einer Messgenauigkeit von einigen mm auch sehr genau.

Ein grosses Problem dieser Sensoren ist ihr grosser *Messkegel*. Der Winkel der ausgesendeten Schallwellen beträgt 15°, was auf eine Distanz von $2m$ bereits einen Messkegel von ca. $1m$ Radius bedeutet. Abbildung 61 zeigt dies nochmals.

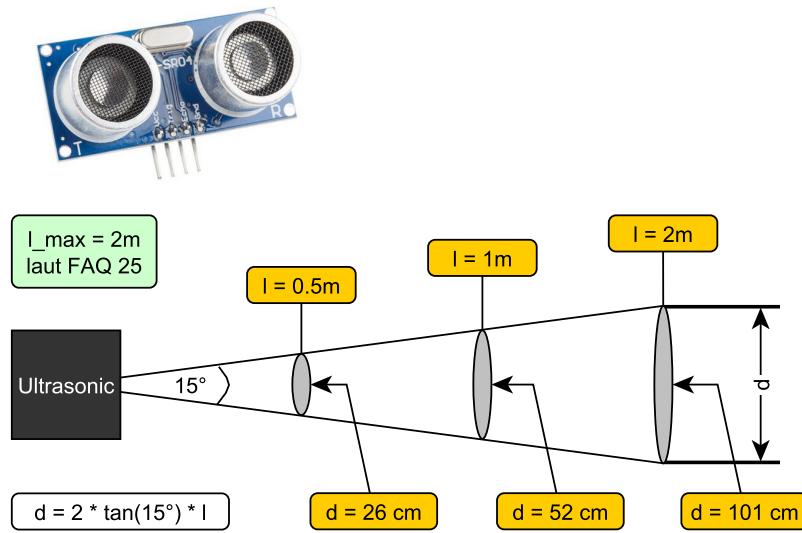


Abbildung 61: Messgenauigkeit Ultraschallsensor

LIDAR

Lidar-Sensoren sind tendenziell sehr teuer, zumindest wenn es darum geht Sensoren, mit welchen das gesamte Umfeld erkannt werden kann, zu betrachten. Auf dem Markt sind allerdings auch solche Sensoren erhältlich, welche ebenfalls als LIDAR arbeiten, allerdings nur die direkte Distanz in eine einzige Richtung detektieren können. Preislich bewegen sich diese in einem Rahmen von 20 – 30CHF, was also noch innerhalb des Budgets für diese Sensorik liegen würde. Ausgestrahlte Kegel sind bei Standard-Sensoren häufig ≈ 2 breit, womit tatsächlich über ein Hindernis hinweg geschaut werden kann. Abbildung 62 zeigt dies nochmals.

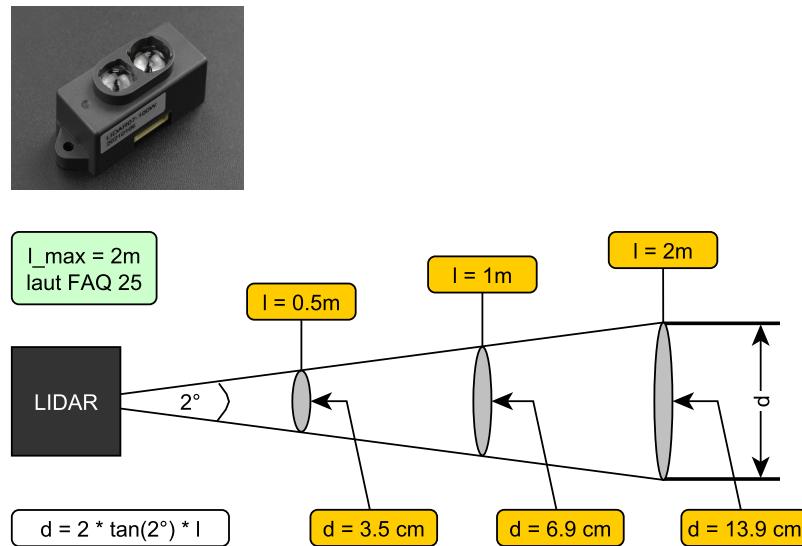


Abbildung 62: Messgenauigkeit LIDAR

Auf die Distanz sind diese Sensoren allerdings lediglich auf $20mm - 30mm$ sensitiv - eine grössere Auflösung wird allerdings auch nicht benötigt für diese Anwendung. Der Messbereich üblicher Sensoren beginnt häufig erst bei $200mm - 300mm$. Diese Sensoren können häufig sehr einfach über Registerzugriffe via I^2C , UART oder SPI ausgelesen werden.

IR-Sensor

Infrarotsensoren, wie der *GP2Y0A02YK0F* von der Firma *SHARP* geben die gemessene Distanz als analoges Signal zurück. Sie sind sensitiv ab einer Distanz von $200mm$ bis auf $1500mm$. Sie arbeiten auf der Basis eines Infrarotlichtes - welches vom gegenüberliegenden Objekt reflektiert wird. Diese Sensoren sind bereits ab *15CHF* auf dem Markt erhältlich. Aufgrund ihres sehr eingeschränkten Messbereichs werden diese Sensoren nicht genauer untersucht - da sie weder im Nahbereich für Hindernisse, noch im Fernbereich für Pylonen eingesetzt werden könnten.

Lichtbarriere

Eine Lichtbarriere, wie sie in Industriellen Anwendungen gerne eingesetzt wird, kann zwar nicht eingesetzt werden, um direkt Distanzen zu Objekten zu messen, aber dafür, die Anwesenheit eines Objektes sehr schnell und genau zu melden. Damit kann sichergestellt werden, dass sich das Hindernis zum entsprechenden Zeitpunkt ganz sicher an der richtigen Position vor dem Fahrzeug befindet.

Ein Team-Mitglied kann aus seinem beruflichen Umfeld auf solche Lichtbarrieren der Firma *SICK* zurückgreifen - wodurch diese folglich sehr günstig für dieses Projekt erhältlich sind. Abbildung 63 zeigt eine solche Lichtschranke.

Kamera und Bilderkennung

Der Roboter wird in jedem Fall eine Kamera verbaut haben, da mit dieser versucht werden soll, verschiedene Linienabgänge zu detektieren. Mit dieser Kamera ist es durchaus ebenfalls möglich, Hindernisse zu erkennen. Kameras sind allerdings im Folgemodul noch genauer zu untersuchen, gerade in Bezug auf Störempfindlichkeit bei unterschiedlichen Lichtverhältnissen.



Abbildung 63: Lichstchranke SICK

Fazit und Entscheidung der Konzeptphase

Um Pylonen in der Ferne zu erkennen, wird ein LIDAR eingesetzt. Genauer gesagt soll der Sensor *TFLuna* von *Benewake* eingesetzt werden. Er bietet eine hohe Resistenz gegen Sonneneinstrahlung, einen kleinen Messkegel und einen sehr überzeugenden Preis. Dieser Sensor ist in Abbildung 64 gezeigt.



Abbildung 64: TFLuna LIDAR

In einem ersten Prototypenaufbau werden mehr Sensoren eingesetzt als in Zukunft vermutlich benötigt werden. So soll grundsätzlich die Anwesenheit eines Hindernisses in der Position, in der es gegriffen werden kann, anhand einer Lichtschranke detektiert werden. Trotzdem wird redundant ein Ultraschallsensor auf Höhe der Hindernisse vorgesehen, welcher im Nahbereich bis zu $0.5m$ Hindernisse erkennen soll. Dadurch kann frühzeitig der Bremsvorgang eingeleitet werden. Falls sich bei Versuchen mit dem Prototyp herausstellt, dass auch die Positionierung des Hindernisses anhand des Ultraschallsensors genau bestimmt werden kann, wird auf die zusätzliche Lichtschranke verzichtet. Parallel dazu werden Auswertemöglichkeiten mit der Kamera weiter untersucht, womit ebenfalls die Orientierung und Position des Hindernisses ermittelt werden kann. Abbildung 65 zeigt das erste angestrebte Konzept nochmals skizziert.

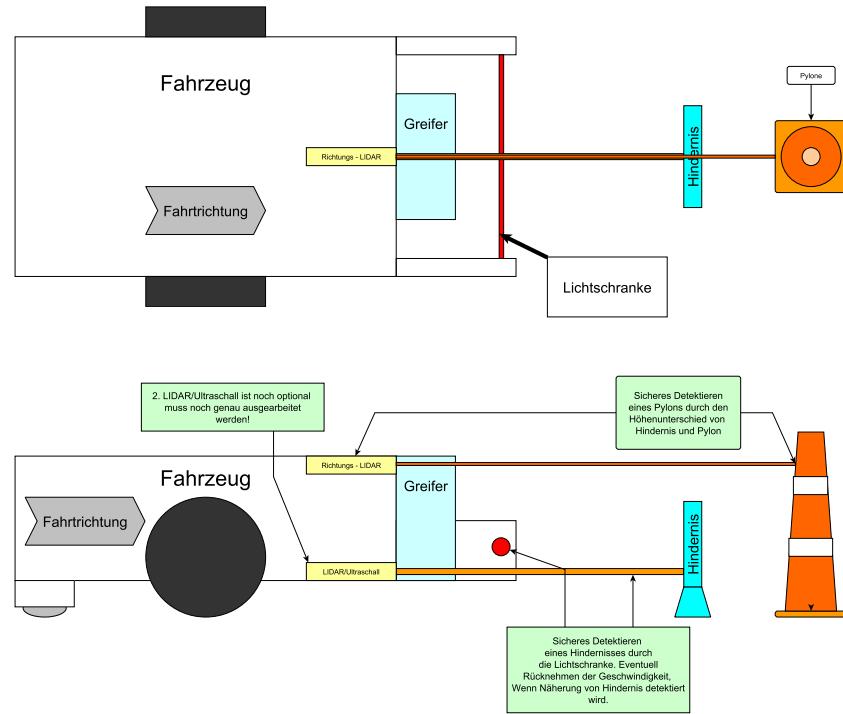


Abbildung 65: Konzept Hinderniserkennung

F. Konzepterstellung Strecken Rückverfolgung

Dieser Abschnitt befasst sich damit, wie der Roboter seinen Weg, den er durch das Wege- netz genommen hat, zurückverfolgen kann und so auf seine Position Rückschliessen kann.

Anforderungen Der Algorithmus setzt voraus, dass das Roboter immer und zu jeder Zeit seine Orientierung als absoluten Winkel ab dem Startpunkt weiss und die zurückgelegte Strecken messen kann.

Konzeption Wineklerfassung Betreffend dem Winkel könnte angenommen werden, dass ein einfacher Kompass auf dem Roboter bereits ausreichen würde, um die Orientierung feststellen zu können. Da sich allerdings Motoren in der direkten Umgebung der Elektronik befinden, welche ebenfalls starke Magnetfeldänderungen verursachen, bestehen zu grosse Bedenken, dass der Winkel so eine willkürliche Form annimmt.

Anstelle dessen wird der Ansatz verfolgt, die momentane Änderungsrate des Winkels über ein Gyroskop zu erfassen und in einem ausreichend geringen Zeitintervall numerisch zu integrieren. Für die ersten Versuche dieses Verfahrens wird ein einfaches Gyroskop des Typs *MPU6050* eingesetzt, welches viele Einstellmöglichkeiten im Sinne von Samplingraten und eines integrierten Tiefpassfilters besitzt. Ausgelesen werden kann dieser über ein einfaches *I²C* Protokoll. Abbildung 66 zeigt eben dieses Gyroskop als Evaluation Board - welches auf einem Steckbrett zu Testzwecken eingesetzt wird.

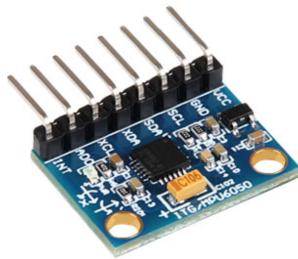


Abbildung 66: MPU6050 Gyroskop EVAL-Board

Konzeption Streckenerfassung Zur Rückverfolgung der zurückgelegten Strecke bieten sich gleich mehrere Möglichkeiten. Der eingesetzte Schrittmotorentreiber wird nicht direkt über ein Step/Dir Interface angesteuert, bietet allerdings trotzdem die Möglichkeit, zurückgelegte Schritte via *SPI* auszulesen. Da dies erst ausgiebig getestet werden kann, wenn der Roboter ein erstes Mal zusammengebaut wurde, werden vorerst noch separate Encoder vorgesehen. Diese sind sehr einfach umgesetzt anhand einer Lichtschranke, welche eine ausreichend gelochte Lochscheibe auszählt.

Entsprechende Signale dieser Lichtschranke erzeugen auf dem Mikroprozessor Interrupts, mit welchen die Anzahl der Pulse ausgezählt wird. Grundsätzlich wird, wie im Kapitel für Antriebe bereits erwähnt, der Antrieb nicht über Encoder geregelt, sondern über den Liniensor. Daher wäre es naheliegend, dass der Raspberry-Pi eben diese Encoder auszählt, da auch nur er auf diese Informationen zugreifen muss. Es ist allerdings nicht sicher, wie gut der Raspberry Pi mangels Echtzeitfähigkeit dafür geeignet ist, weshalb sich Encoder-Schnittstellen sowohl auf dem Antriebscontroller als auch auf dem Raspberry-HAT befindet.

Versuch Wineklerfassung Bei Versuchen wurde mit verschiedenen Parametern variiert, um ein gutes Ergebnis zu erreichen. Die folgenden Parameter liefern ein sehr zufriedenstellendes Ergebnis:

Parameter	Wert
Abfragehäufigkeit Mikrocontroller	$25\mu s$
Samplingrate MPU6050	$1kHz$
Digital Low Pass Filter (DLPF)	$f_g = 42Hz$

Tabelle 19: Initialisierungsparameter MPU6050

Per Software ist noch ein Threshold-Wert eingestellt, mit welchem ganz leichtes Zittern des Sensorwertes ignoriert werden kann. Mit diesen Werten können Winkeländerungen von 360° in weniger als einer Sekunde problemlos erfasst werden, ohne dass der Winkel in der Summe davon driftet. Diese Technik hat sich also bewährt und wird auch so im Roboter eingesetzt.

Der *Motion Board* PCB wird ein Gyroskop verbaut haben, welches einer neueren Generation als das bei diesen Versuchen eingesetzten Gyroskop entspricht (*ICM42670*). Dies mit der Begründung, dass dieses Gyroskop mittlerweile obsolet ist. Die Art der Ansteuerung unterscheidet sich jedoch kaum von der des vorgängig getesteten. Das *Motion Board* wird genügend I^2C Schnittstellen besitzen, um zu einem späteren Zeitpunkt wieder auf den hier getesteten Sensor zurückzuwechseln.

G. Liniensensor

Anbei folgen diverse Unterkapitel, welche Berechnungen sowie Messungen für die Erstellung eines Liniensensors beinhalten.

Geometrische Überlegungen

Abbildung 67 zeigt die konzeptionellen Überlegungen einer Messzelle des Liniensensors. Gemäss dem Datenblatt hat der UV-Emitter (Sender) einen Abstrahlwinkel von 15° und der Fototransistor (Empfänger) einen Einfallswinkel von 60° . Für eine möglichst störungsfreie Detektion muss auf dem Boden eine möglichst grosse Fläche des Empfänger-Kreises durch den Sender-Kreis ausgefüllt werden. Der passende Abstand zwischen Liniensensor und Boden muss in der Praxis mit Versuchsmessungen eruiert werden.

Diese einzelnen Messzellen sollen untereinander und von der Umwelt abgekapselt werden. Dies mit der Begründung, dass die Fototransistoren, gegebenenfalls empfindlich auf Umwelteinflüsse reagieren könnten. Daher wird ein Gehäuse für die Abschirmung von Umwelteinflüssen konzipiert.

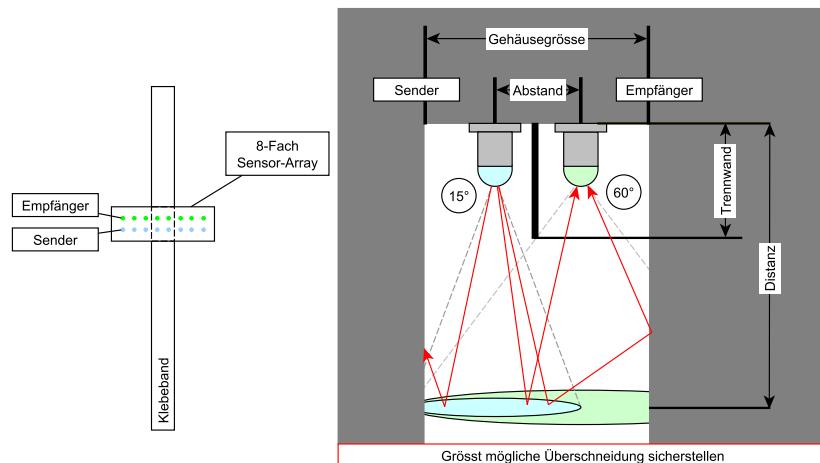


Abbildung 67: Konzept des Liniensensors

Dimensionierung Gehäuse

Damit der Empfänger möglichst wenig auf Umwelteinflüsse reagiert, wird ein Gehäuse dimensioniert, welches den Sensor abschirmt. Die nachfolgende Abbildung 68 zeigt eine Skizze, welche den Aufbau des Gehäuses repräsentiert. In dieser Skizze wird jeder Emitter und Fototransistor von allen anderen abgeschottet. Die vier Löcher passen genau auf das PCB des Liniensensors. Die Höhe des Gehäuses, stellt den Abstand zwischen Liniensensor und Boden dar. Die grösste Stromdifferenz wurde in einer Höhe von 2.5 cm gemessen. Daher wird die Höhe des Gehäuses auf 2.5 cm festgelegt. Die isometrische Ansicht des Gehäuse ist in Abbildung 69 dargestellt.

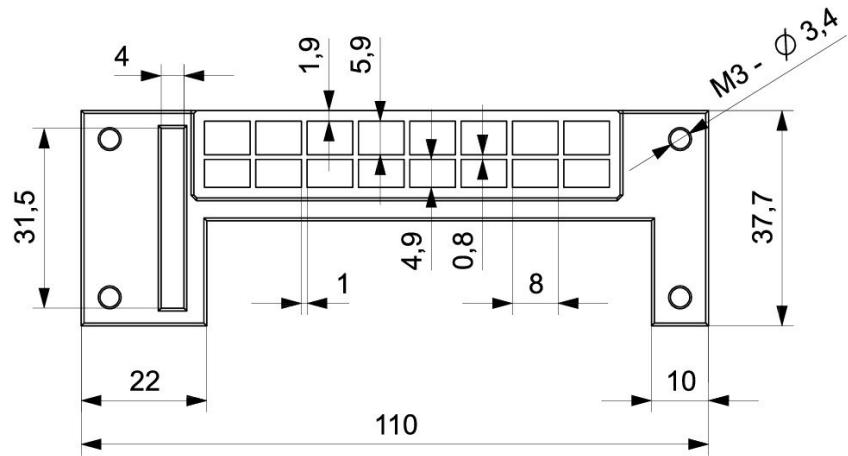


Abbildung 68: Vermassung des Gehäuses in Siemens NX

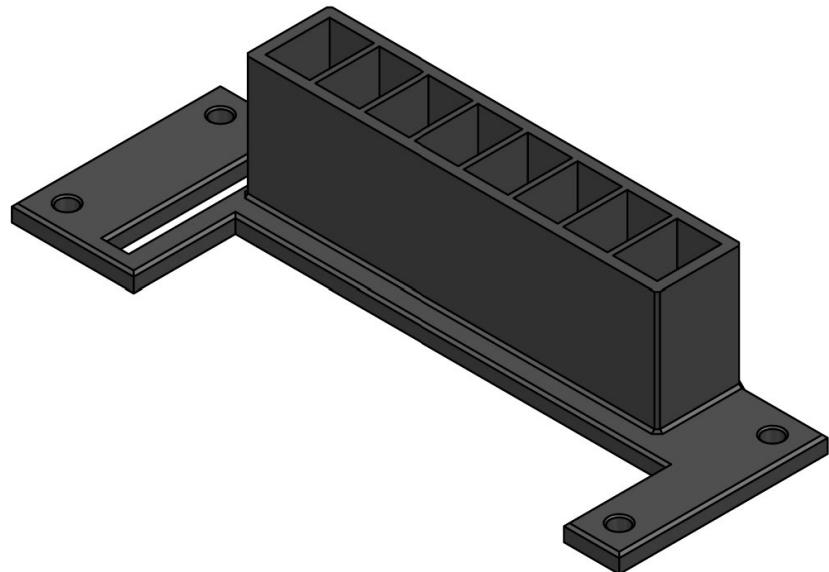


Abbildung 69: Isometrische Ansicht des Gehäuses in Siemens NX

Schema einer Messzelle

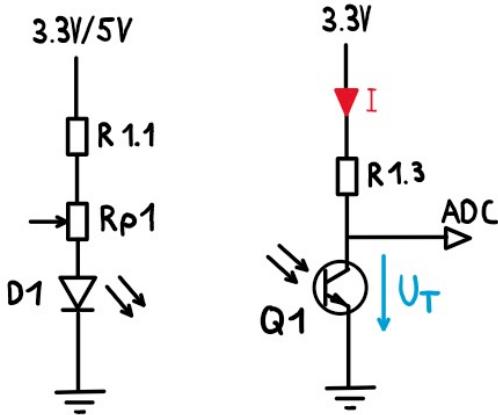


Abbildung 70: Schema einer Messzelle

Abbildung 70 zeigt das Schema einer einzelnen Messzelle. Die Speisespannung beträgt 3.3V bis 5V. Zudem ist als Sicherheitsmaßnahme beim Messabgang über dem Fototransistor ein Platz für einen Widerstand vorgesehen, der bei allfälligen Stromspitzen den Strom limitiert. Da dies mit der verwendeten Auswertung nicht notwendig ist, wird ein $0\ \Omega$ Widerstand eingebaut. Die Potentiometer werden eingebaut, dass allfällige Bauteiltoleranzen eliminiert werden können. In den folgenden Abschnitten wird erläutert, wie die Werte der Komponenten $R_{1.1}$, R_{p1} und $R_{1.3}$ dimensioniert sind.

Wahl des Lichtspektrums

Damit der Unterschied zwischen dem Wettkampfuntergrund und dem Klebeband möglichst drastisch hervorgehoben werden kann, ist zuerst eine Evaluierung des optimalen Lichtspektrums nötig. Auf der Folgeseite sind Versuche dokumentiert, welche die Spannungen über dem Fototransistor im Infrarot-Spektrum (IR) und im Ultraviolet-Spektrum (UV) aufzeigen. Für die Infrarot-Variante wird ein Emitter und ein Fototransistor im IR-Spektrum (beide 940 nm Peak-Wellenlänge) verwendet. Beim Versuch im UV-Spektrum wird ein UV-Emitter (395nm Peak-Wellenlänge) mit einem Fototransistor (630nm Peak-Wellenlänge) im sichtbaren Spektrum verwendet. Dabei wird eine fluoreszierende Wirkung des Klebebandes erwartet.

Dimensionierung IR-Messzelle

In diesem Abschnitt wird eine IR-Messzelle dimensioniert. Gemäß Datenblatt kann über dem IR-Emitter von 1.3 V bis 2 V abfallen. Die Speisespannung beträgt 3.3 V. Der maximale Strom beträgt 100 mA. Der Strom wird gewählt, dass er ungefähr in den Bereich von 10 mA bis 60 mA fällt. Dies wurde so gewählt, das nicht zu viel Leistung verbraucht, aber der Emitter trotzdem mit genügend Strom versorgt wird. Daraus resultiert die folgende Widerstandsberechnung:

$$R_{max} = \frac{U_q - U_{LED}}{I_{LEDmin}} = \frac{3.3V - 1.3V}{0.01A} = 200\Omega$$

$$R_{min} = \frac{U_q - U_{LED}}{I_{LEDmax}} = \frac{3.3V - 1.3V}{0.03A} = 33.33\Omega$$

Daraus resultiert ein Potentiometer R_{p1} von 200Ω und ein $R_{1.1}$ von 33Ω . Der Widerstand $R_{1.3}$ kann erst durch Tests dimensioniert werden, da der Strom durch den Widerstand noch unklar ist.

Dimensionierung UV-Messzelle

In diesem Abschnitt wird die UV-Messzelle dimensioniert. Über die Speisespannung von 5 V wird ein UV-Emitter mit Vorwiderstand bestromt. Der erlaubte Strom wird gemäss dem Datenblatt des UV-Emitters von 10 mA bis 20 mA vorgeschlagen. Der Strom nicht 30mA übersteigen. Daher resultiert:

$$R_{max} = \frac{U_q - U_{LED}}{I_{LEDmin}} = \frac{5V - 2.9V}{0.01A} = 210\Omega$$

$$R_{min} = \frac{U_q - U_{LED}}{I_{LEDmax}} = \frac{5V - 2.9V}{0.03A} = 70\Omega$$

Daraus eignet ein Potentiometer R_{p1} von 200Ω und ein $R_{1.1}$ von 82Ω , welches ungefähr im zuvor festgelegten Strombereich liegt:

$$I_{max} = \frac{U_R}{R_{min}} = \frac{2.1}{82\Omega} = 0.0256A = 25.6mA$$

$$I_{min} = \frac{U_R}{R_{max}} = \frac{2.1}{282\Omega} = 0.00745A = 7.45mA$$

Aufgrund dieser Widerstandsaufteilung kann der Strom in dem empfohlenen Bereich frei eingestellt werden. Der Widerstand $R_{1.3}$ kann erst später dimensioniert werden, da der Strom durch ihn unklar ist.

Versuchsmessungen

Bei den beiden oben dimensionierten Varianten sind je eine Messzelle auf ein PCB gelötet und im Anschluss der Strom durch den Fototransistor gemessen. Die gemessenen Werte sind in der folgenden Tabelle 20 festgehalten. Dabei resultierte der Abstand von ungefähr 2.7mm von Messfläche zu Liniensor. Das Potentiometer wird mittig eingestellt, sprich auf 100Ω .

Lichtspektrum	Strom Klebeband [mA]	Strom Fuge [mA]	Strom Fliese [mA]
Infrarot	1.368	1.248	1.92
Ultraviolett	0.2024	0.0691	0.0494

Tabelle 20: Strom durch die beiden Fototransistor in verschiedenen Lichtspektren

Aus der Tabelle ist zu sehen, dass im UV-Spektrum der Strom über dem Klebeband geringerer ist als beim IR-Spektrum. Jedoch ist der Differenzfaktor beim UV-Spektrum zu den verschiedenen Messuntergründen etwa doppelt so hoch im Vergleich zum IR-Spektrum. Aufgrund dieser Messresultate wird auf das UV-Spektrum zurückgegriffen.

Dimensionierung des Widerstandes $R_{1.3}$

Im oberen Teil wurde der Strom durch einen Fototransistor gemessen. Wie bereits erwähnt, wurde wegen der besseren Messwerte auf das UV-Spektrum zurückgegriffen. Für den Versuch wurde jedoch nur eine Messzelle gemessen. Bei nun acht Messzellen beeinflussen sich die Messzellen trotz möglichst genauer Abschirmung gegenseitig. Abschirmung immer noch gegenseitig. Das heisst, mehr Licht entspricht einer höheren Ansteuerung der Basis des Fototransistors und damit einem höheren Strom. Dieser muss nun erneut gemessen werden. Bei allen acht Messzellen werden die verschiedenen Ströme auf allen Messuntergründen gemessen. In der Tabelle 21 sind die maximalen Ströme, die auf der Platte und der Fuge gemessen wurden. Mit den Potentiometern werden die Ströme durch die Emitter so eingestellt, dass sie auf dem Klebeband durch den Fototransistor immer etwa gleich hoch sind (480 mA). Der Grund dafür ist, dass ein möglichst gleichmässiger Strom durch den Fototransistor fließt, wenn sich dieser auf dem Klebeband befindet.

Messuntergrund	Strom in [mA]
Klebeband	0.48
Fliese	0.34
Fuge	0.23

Tabelle 21: Strom durch den Fototransistor auf den verschiedenen Messuntergründen

Weil nun die Ströme durch den Fototransistor bekannt sind, kann nun der passende Widerstandswert für $R_{1.3}$ berechnet werden. Weil der Strom durch den Fototransistor auf dem Klebeband nie ganz genau identisch sein wird, wird ein Schwellwert angenommen. Es wird nun ein Strom I_F von $400 \mu A$ angenommen. Dieser Wert bietet noch genügend Differenz zum Fliesen-/Fugenstrom. Daher kann der Widerstand $R_{1.3}$ berechnet werden.

$$R_{1.3} = \frac{U_q}{I_{Ph}} = \frac{3.3 \text{ V}}{400 \mu\text{A}} = 8.25 \text{ k}\Omega$$

Daraus resultiert ein Widerstand $R_{1.3}$ von 8200 Ohm. Das bedeutet, sobald der Strom grösser als $400 \mu A$ ist, geht die Spannung über dem Fototransistor annähernd zu Ground. Unter dem Schwellwert wird ein Spannungsoffset registriert.

Messwerte des Liniensensors

Um diese Berechnungen zu verifizieren, wird der Liniensor mit einem Arduino geprüft. Die Auswertung des Arduinos ist in der Tabelle 22 dargestellt.

Kennzahl aus Software	Spannung in [V]
0	0V
205	1V
1023	5V

Tabelle 22: Spannungsauswertung des Arduinos über dem Fototransistor

Es kann also ein Spannungsbereich von 0 V bis 5 V über dem Fototransistor ausgewertet werden. Anbei sind nun einige Versuchsmessungen dokumentiert.

Bei der Abbildung 71 ist der Vergleich Fliese (oben) zu Klebeband (unten) ersichtlich. Die Werte auf dem Boden schwanken zwar, sind aber nie unter 350, also sprich 1.707 V. Die Spannung auf dem Klebeband wird jedoch immer sauber gegen Ground gezogen.

```
A0: 975 A1: 976 A2: 971 A3: 909 A4: 910 A5: 854 A6: 756 A7: 827
A0: 672 A1: 662 A2: 549 A3: 354 A4: 535 A5: 492 A6: 402 A7: 666
A0: 653 A1: 662 A2: 624 A3: 508 A4: 689 A5: 666 A6: 644 A7: 720
A0: 707 A1: 675 A2: 683 A3: 620 A4: 650 A5: 547 A6: 378 A7: 417
A0: 80 A1: 80 A2: 84 A3: 73 A4: 86 A5: 81 A6: 77 A7: 188
A0: 66 A1: 70 A2: 71 A3: 66 A4: 74 A5: 76 A6: 68 A7: 74
A0: 67 A1: 70 A2: 72 A3: 66 A4: 74 A5: 75 A6: 67 A7: 73
A0: 67 A1: 70 A2: 71 A3: 66 A4: 73 A5: 74 A6: 67 A7: 72
```

Abbildung 71: Vergleich der Kennzahlen von Fliese (oben) und Klebeband (unten) des Arduino

Die Abbildung 72 zeigt, wie eine mögliche Strecke erfasst werden kann. Dabei wird vor allem der Pin A3 gegen Ground gezogen. Außerdem ist ersichtlich, dass die Pins A2 und A4 auch das Klebeband erfassen. Diese jedoch nicht so deutlich wie der Pin A3. Dies liegt daran, dass diese beiden Messzellen nicht direkt auf das Klebeband ausgerichtet sind.

```
A0: 594 A1: 569 A2: 519 A3: 67 A4: 76 A5: 613 A6: 515 A7: 642
A0: 601 A1: 569 A2: 429 A3: 69 A4: 218 A5: 519 A6: 487 A7: 735
A0: 601 A1: 586 A2: 415 A3: 69 A4: 242 A5: 511 A6: 567 A7: 766
A0: 584 A1: 553 A2: 448 A3: 68 A4: 211 A5: 504 A6: 517 A7: 711
A0: 601 A1: 591 A2: 442 A3: 69 A4: 240 A5: 524 A6: 530 A7: 705
A0: 653 A1: 635 A2: 491 A3: 72 A4: 250 A5: 570 A6: 612 A7: 805
```

Abbildung 72: Vergleich der Kennzahl Klebeband (vor allem A3) und der Fliese

In Abbildung 73 wird eine Auswertung aufgezeigt, bei der die Fuge auch miteinbezogen wurde. Diese ist bei dem Pin A2 leicht sichtbar. Dies fällt aber fast nicht auf. Im Vergleich zu A6, bei dem das Klebeband angezeigt wird. Die restlichen Pins stellen die Fliese dar.

```
A0: 724 A1: 849 A2: 572 A3: 674 A4: 805 A5: 408 A6: 80 A7: 525
A0: 722 A1: 849 A2: 569 A3: 672 A4: 803 A5: 406 A6: 79 A7: 528
A0: 707 A1: 846 A2: 553 A3: 666 A4: 791 A5: 395 A6: 78 A7: 515
A0: 702 A1: 845 A2: 547 A3: 663 A4: 789 A5: 398 A6: 78 A7: 511
A0: 681 A1: 840 A2: 545 A3: 660 A4: 784 A5: 397 A6: 77 A7: 516
A0: 684 A1: 842 A2: 533 A3: 657 A4: 770 A5: 388 A6: 76 A7: 500
A0: 715 A1: 847 A2: 550 A3: 664 A4: 788 A5: 388 A6: 78 A7: 531
A0: 720 A1: 845 A2: 552 A3: 664 A4: 773 A5: 380 A6: 74 A7: 380
```

Abbildung 73: Vergleich der Kennzahl Fuge (A6) und dem Klebeband (A6) umrandet von der Fliese

H. Risikomanagement

Das Risikomanagement wird nach der ALARP-Methode (*engl. as low as reasonably possible*) durchgeführt. Dafür werden Risiken im ersten Schritt identifiziert und anschliessend durch risikomindernde Massnahmen auf ein Niveau reduziert, das ein angemessenes Mass an Sicherheit bietet. Die Bewertung erfolgt im gesamten Team und basiert auf einer subjektiven Einschätzung zur Erfüllung der Aufgabe. Ziel ist es, möglichst früh im Projektverlauf kritische Punkte zu identifizieren und den Fokus auf diese zu legen.

Eintrittswahrscheinlichkeit (EW)

Die Eintrittswahrscheinlichkeit ist ein Mass für die Wahrscheinlichkeit, mit der ein Ereignis eintreten könnte.

EW	Bezeichnung	%
6	häufig	> 90%
5	wahrscheinlich	> 70%
4	gelegentlich	> 50%
3	entfernt vorstellbar	> 30%
2	unwahrscheinlich	> 15%
1	unvorstellbar	> 5%

Tabelle 23: Legende Eintrittswahrscheinlichkeit

Schadensausmass (SA)

Das Schadensausmass ist ein Mass dafür, wie fatal ein eintretendes Ereignis für den Projekterfolg ist.

SA	Bezeichnung	Auswirkung
4	katastrophal	Wettbewerb abgebrochen
3	kritisch	Gefährdung für Projekterfolg
2	geringfügig	Minderung des Projekterfolgs
1	unwesentlich	Störung des Projekterfolgs

Tabelle 24: Legende Schadensausmass

Bereichsdefinition

Die entsprechenden Risiken sind mit der folgenden Farbgebung codiert, um die Notwendigkeit von Massnahmen zu kennzeichnen.

Farbcodierung	Bedeutung
grün	Akzeptabler Bereich
gelb	ALARP-Bereich
rot	Inakzeptabler Bereich

Tabelle 25: Legende Bereiche

Erfasste Risiken

Die nachfolgenden Tabellen zeigen die identifizierten Risiken bis zum aktuellen Zeitpunkt.

Allgemeines

#	Risiko	SA	EW	Auswirkungen
1.1	Unterschiedliche Erwartungen an den Projekterfolg	2	1	Enttäuschung bei Teammitgliedern, Kommunikationsprobleme
1.2	Verpasste Abgaben aufgrund mangelhaften Zeitmanagements	2	2	Testate werden nicht erteilt. Unstimmigkeiten zwischen Teammitgliedern
1.3	Konzept wurde bis zum Ende des Semesters nicht vollständig durchdacht	3	2	Probleme treten in PREN 2 auf, die früher hätten erkannt werden können
1.4	Budget wird knapp, da Aufwände unterschätzt wurden	4	3	Es muss an kritischen Stellen gespart werden
1.5	Personeller Ausfall durch Teamwechsel in PREN2 oder Krankheit	3	2	Aufgaben müssen umverteilt werden
1.6	Kommunikationsprobleme und fehlerhafte Absprachen unter Teammitgliedern	3	2	Missverständnisse, ineffizientes Arbeiten führt zu Zeitverlust
1.7	Vergessene Anforderungen aus der Aufgabenstellung	4	1	Aufgabenstellung wird nicht vollständig erfüllt
1.8	Fehlende Motivation durch unzureichende Teilerfolge bei Konzeption und Prototypen Aufbau	2	1	Stimmung im Team leidet darunter

Tabelle 26: Erfasste Risiken mit Bewertung

Mechanik

#	Risiko	SA	EW	Auswirkungen
2.1	Fahrzeug kann Hindernis erfassen und aufnehmen, jedoch nicht exakt positionieren.	3	4	Punktabzug bei der Bewertung
2.2	Fahrzeug überschreitet das zulässige Gesamtgewicht; 2 kg ist ein enger Rahmen.	4	3	Disqualifikation

Tabelle 27: Erfasste Risiken im Bereich Mechanik

Elektrotechnik

#	Risiko	SA	EW	Auswirkungen
3.1	Fahrzeug kann Linie nicht erkennen und verlässt daher die Strecke.	4	5	Disqualifikation
3.2	Fahrzeug wird durch Umwelteinflüsse wie Lichtverhältnisse gestört.	3	6	Punktabzug durch Abkommen von der Linie oder Kollisionen mit Hindernissen
3.3	Akku reicht nicht für beide Läufe mit Tests, da aufgrund des Gewichts zu knapp dimensioniert.	4	2	Fahrzeug kann das Ziel nicht erreichen.
3.4	Kommunikation zwischen verschiedenen Mikrocontrollern und dem Hauptrechner wird durch Umwelteinflüsse gestört.	3	2	Möglicherweise fehlerhafte oder keine Steuersignale
3.5	Lastregelung der Spannungsversorgung unzureichend, da Leistungselektronik zu viel Strom benötigt.	2	2	Prozessoren und Sensorik könnten unversorgt sein und neu starten müssen

Tabelle 28: Erfasste Risiken im Bereich Elektrotechnik

Informatik

#	Risiko	SA	EW	Auswirkungen
4.1	Fahrzeug verliert die Orientierung im Parcours und erreicht das Ziel nicht.	3	5	Fahrzeug benötigt lange, bis es das Ziel zufällig erreicht.

Tabelle 29: Erfasste Risiken im Bereich Informatik

Erfasste Massnahmen

Allgemein

Risiko #	Massnahme	Neu EW
1. 4	Kostenpunkte bei jedem Entwicklungsschritt frühzeitig berücksichtigen und als wichtiges Kriterium für Technologieentscheide werten.	1

Tabelle 30: Erfasste Massnahmen für allgemein betreffende Risiken

Abbildung 74 zeigt schematisch, wie die getroffenen Massnahmen entsprechende Risiken für den Projekterfolg reduzieren.

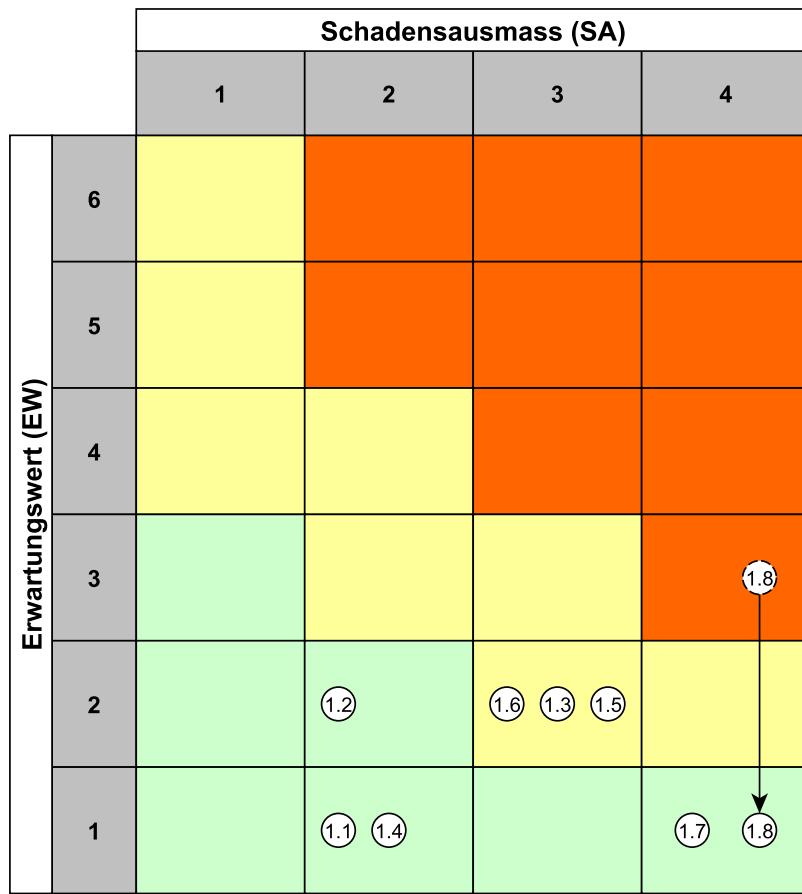


Abbildung 74: Grafische Darstellung der Risikoanalyse Allgemein

Mechanik

Risiko #	Massnahme	Neu EW
2. 1	Höhere Gewichtung auf dieses Detail in der Konzeptbewertung.	3
2. 2	Gewicht der Bauteile frühzeitig überschlagen und bei jedem Entwicklungsschritt berücksichtigen. Liste für bereits bekannte Gewichte führen und Budget für Baugruppen festlegen.	1

Tabelle 31: Erfasste Massnahmen für die Mechanik betreffende Risiken

Abbildung 75 zeigt schematisch, wie die getroffenen Massnahmen entsprechende Risiken für den Projekterfolg reduzieren.

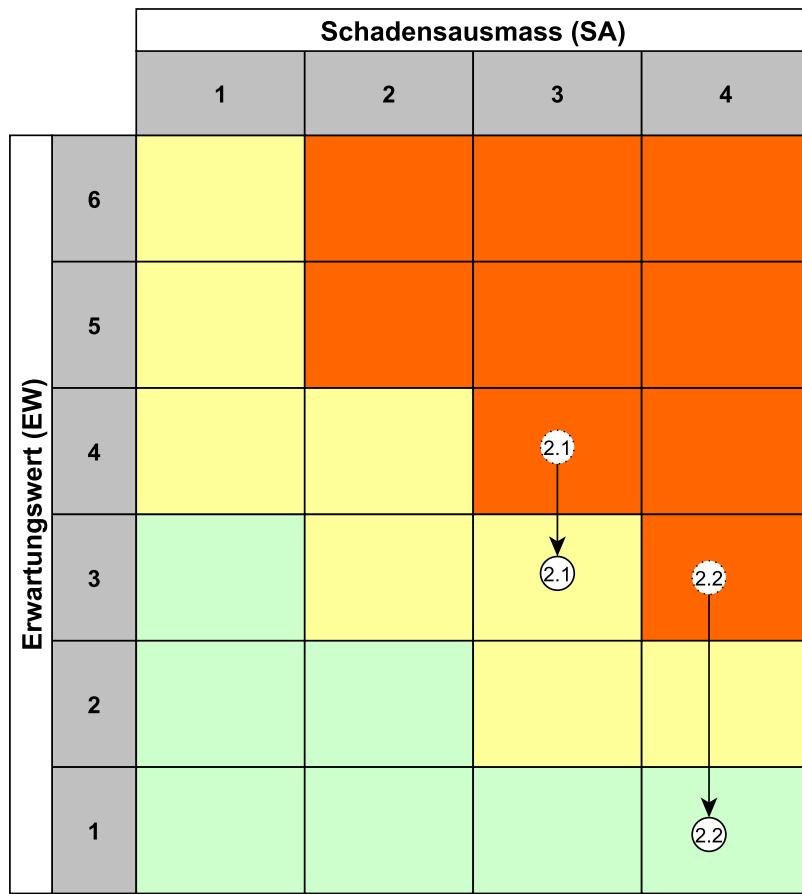


Abbildung 75: Grafische Darstellung der Risikoanalyse Mechanik

Elektrotechnik

Risiko #	Massnahme	Neu EW
3. 1	Frühzeitiges Testen und Optimieren der Genauigkeit der Sensorik. Möglichkeit bieten, das Fahrzeug nach Fahrzeugwinkel und gefahrenen Strecken zu regeln. Technologieentscheid erst nach ausgiebigem Testen.	2
3. 2	Sensorik, die optisch arbeitet, möglichst abgekapselt von der Umwelt betreiben. Alternativ Wellenlängen, die das sichtbare Licht beinhalten, vermeiden.	3
3. 3	Akku doppelt herstellen/einkaufen, Ladestation extern ausführen, um immer einen vollgeladenen Akku bereit zu haben.	1
3. 4	Kommunikationsleitungen mindestens als Twisted-Pairs mit GND – besser aber geschirmt ausführen.	1
3. 5	Genügend Spannungsreserve des Akkus „nach oben“ einplanen. Vor der Spannungsversorgung für High-Level-Controller und Mikrocontroller aktive Filter (z.B. LDO oder ähnliches) vorsehen.	2

Tabelle 32: Erfasste Massnahmen für Risikoanalyse

Abbildung 76 zeigt schematisch auf, wie die getroffenen Massnahmen entsprechende Risiken auf den Projekterfolg reduzieren.

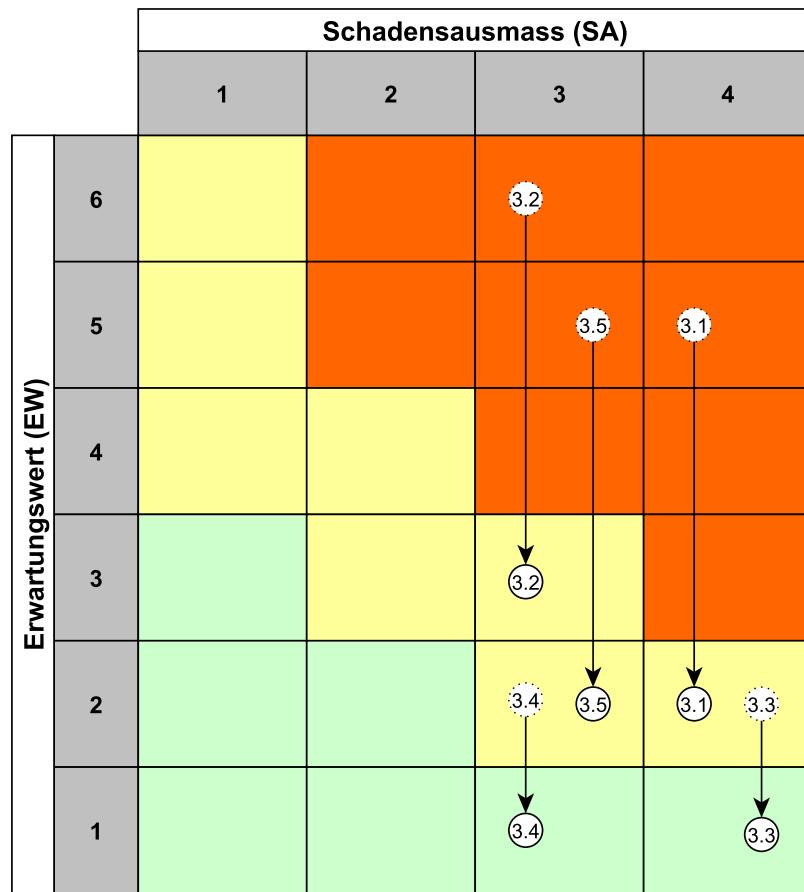


Abbildung 76: Grafische Darstellung Risikoanalyse Elektrotechnik

Informatik

Risiko #	Massnahme	Neu EW
4. 1	Fallback-Lösung: Fahrzeug fährt immer links.	3

Tabelle 33: Erfasste Massnahmen für Risikoanalyse

Abbildung 77 zeigt schematisch auf, wie die getroffenen Massnahmen entsprechende Risiken auf den Projekterfolg reduzieren.

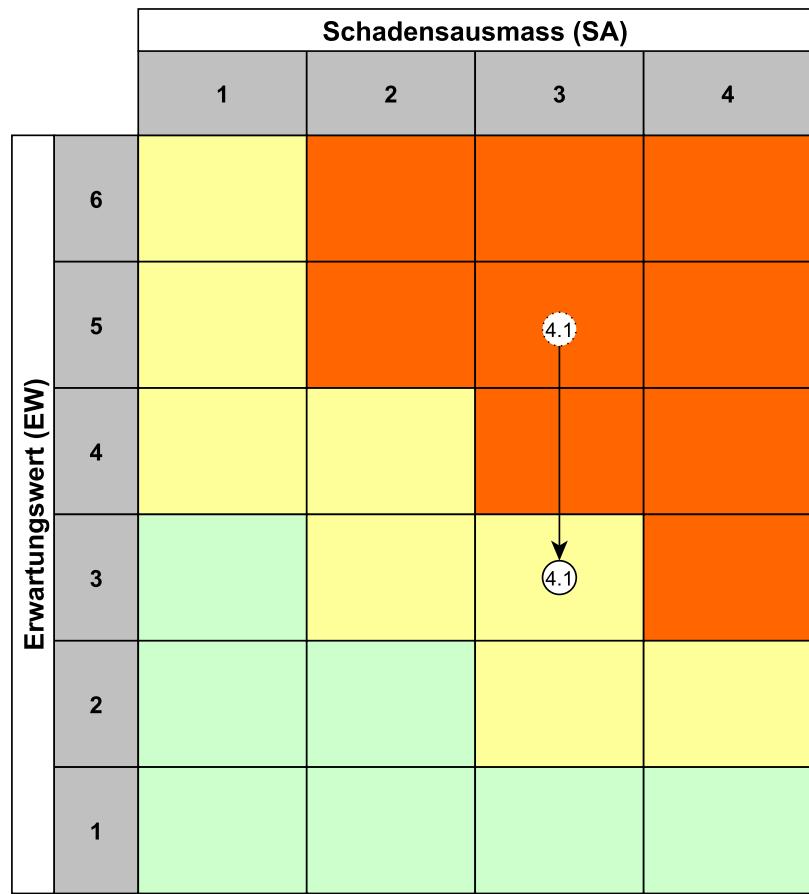


Abbildung 77: Grafische Darstellung Risikoanalyse Informatik

I. Recherche

Die nachfolgende Quellsammlung in Tabelle 34 dient als Übersicht zur Technologierecherche und wird im Laufe des Projekts weitergeführt, um die verwendeten Quellen im Ausblick auf die Schlussdokumentation zu sammeln. Die unter Grade aufgeführten Werte dienen zur Bewertung der Relevanz der Quellen für das Projekt und deren weiterführende Benutzung.

Quellen

Thema	Stichwort	Grade	Quelle	Beschreibung
Simulator	Pfadfindung	2	Link	Visualisierung verschiedener Pfadfindungsalgorithmen.
Simulator	Pfadfindung	5	Link	Performance Evaluation von Pfadfindungsalgorithmen.
Simulator	Graph	3	Link	Erstellung von 2D Graphen.
Simulator	2D-Simulation für autonome Fahrzeuge	4	Link	Simulationstool für Visualisierung.
Simulator	Sensoren und KI	4	Link	Programmierung von Sensoren und neuronalen Netzen in Javascript.
Simulator	Physik Auto	4	Link	Simulation eines realistischen Fahrverhaltens.
Simulator	Editierbare Benutzerober- flächen	5	Link	Benutzerfreundliche Oberfläche.
Simulator	Pfadfindung, Berechenbarkeit	8	Link	Übersicht und Visualisierung verschiedener fortgeschrittener Pfadfindungsalgorithmen.
Simulator	Pfadfindung	6	Link	State Space Exploration: Grundlagen der Graphenexploration.
Simulator	Pfadfindung	5	Link	Übersicht über Model Predictive Path Integral (MPPI).
Simulator	Pfadfindung, Optimierungen	7	Link 1, Link 2	Markov Decision Processes (MDP): Modellierungen von Entscheidungen bei ungewissem Ausgang, welcher Weg ist wahrscheinlich der schnellste im Graph.

contd

Tabelle 34: Quellsammlung

Tabelle 34 – Fortsetzung

Thema	Stichwort	Grade	Quelle	Beschreibung
Simulator	Pfadfindung	7	Link	Detaillierte Beschreibung des D* Lite Algorithmus.
Simulator	Pfadfindung	4	Link	Euclidean Distance Transform für heuristische Entscheidungen bei Graphenproblemen.
Sensorik	Raumwahrnehmung, Image Processing	5	Link	Depth Perception: Grundlagen für Raumwahrnehmung bei der Bildverarbeitung.
Sensorik	Homographie, Image Processing	5	Link	Informationen, um verzerrte Bilder in verschiedene Perspektiven zu transformieren.
Sensorik	Kantenerkennung, Image Processing	9	Link 1, Link 2	Erkennung von Kanten in Bildern, ermöglicht rudimentäre Kollisionserkennung.
Sensorik	Image Processing	8	Link	Analyse von mehreren SLAM Algorithmen.
Elektrotechnik - Antriebe	BLDC Grundlagen	10	Link	Application Note: Grundlagen BLDC Motoren.
Elektrotechnik - Antriebe	BLDC Grundlagen	6	Link	Application Note: Grundlagen BLDC Motoren.
Elektrotechnik - Antriebe	Brushless DC Motor Fundamentals	7	Link	Application Note: Grundlagen BLDC Motoren.
Elektrotechnik - Antriebe	Stepping Motors Fundamentals	10	Link	Application Note: Grundlagen Schrittmotoren.
Elektrotechnik - Antriebe	Stepping Motors Fundamentals	7	Link	Application Note: Grundlagen Schrittmotoren.
Elektrotechnik - Antriebe	Stepper Motor Reference	7	Link	Application Note: Grundschaltungen Schrittmotoren.
Elektrotechnik - Energiemanagement	Li-Ion Batterie	5	Link	Buch: Lithium-Ionen Batterien.
Elektrotechnik - Energiemanagement	Li-Ion Basics	8	Link	Buch: Batterietypen.
Elektrotechnik - Energiemanagement	Recycling Li-Ion; Li-Ion	/	Link	Buch: Recycling.

contd

Tabelle 34: Quellensammlung

Tabelle 34 – Fortsetzung

Thema	Stichwort	Grade	Quelle	Beschreibung
Elektrotechnik - Energiemanagement	Li-Ion Battery	10	Link	Buch: Verschiedene Batterietypen.
Elektrotechnik - Energiemanagement	NiCad Battery Charge	5	Link	Beschreibung: NiCad vs. NiMH Batterien.
Elektrotechnik - Energiemanagement	NiCad Battery Basics	10	Link	Buch: Grundlagen Nickel-Batterien, Ladevorgänge.
Elektrotechnik - Energiemanagement	Lead Acid Batteries; Batteries; Ni-Cd Batteries	10	Link	Buch: Verschiedene Batterietypen sowie Ladeverfahren.
Elektrotechnik - Energiemanagement	Lead Acid Battery	6	Link	Research Paper über Blei-Akkumulatoren.
Elektrotechnik - Energiemanagement	Lead Acid Battery Charge	4	Link	Application Note über Ladeverfahren zu Blei-Akkumulatoren.
Elektrotechnik - Energiemanagement	Lead Acid Battery	2	Link	Research Paper zu Blei-Akkumulatoren.
Elektrotechnik - Energiemanagement	Battery Management; Li-Ion Battery	8	Link	Buch über Batteriemanagementsysteme für Li-Ion Akkus.
Elektrotechnik - Energiemanagement	Battery Management; Li-Ion Battery	7	Link	Buch über Batteriemanagement und Li-Ion Akkus.
Elektrotechnik - Energiemanagement	Battery Management; Li-Ion Battery	6	Link	Buch über Batteriemanagement und Li-Ion Akkus.
Elektrotechnik - Sensoren	LiDAR und Ultraschall	3	Link	Unterschied von LiDAR und Radar für Abstandsmessung.
Elektrotechnik - Sensoren	Abstandsmessung	4	Link	Möglicher LiDAR Sensor mit Time-of-Flight.
Elektrotechnik - Sensoren	Abstandsmessung	4	Link	Möglicher Ultraschallsensor.
Elektrotechnik - Sensoren	Pfadfindung	5	Link	Verschiedene Sensoren für die Pfadfindung.

contd

Tabelle 34: Quellensammlung

Tabelle 34 – Fortsetzung

Thema	Stichwort	Grade	Quelle	Beschreibung
Elektrotechnik - Sensoren	Pfadfindung	4	Link	Möglicher Infrarotsensor für die Pfadfindung.
Elektrotechnik - Sensoren	Pfadfindung	3	Link	Geschwindigkeit und Strecke berechnen mit Hallsensor.
Elektrotechnik - Sensoren	Streckenerkennung	8		
Maschinenbau	Mecanum Wheels Overview	5	Link	Überblick über Mecanumräder und deren Verwendungszweck in der Industrie.
Maschinenbau	Räder	7	Link	Überblick und Auswahl verschiedener Rädertypen für einen Roboter.
Maschinenbau	Greifer	7	Link	Funktionsweise von verschiedenen Greifermechanismen.
Maschinenbau	Greifer	4	Link	Auswahl an Greifern und Linearführungen.
Maschinenbau	Greifer	6	Link	Funktionsweise von verschiedenen Greifermechanismen.
Maschinenbau	Linearführung	5	Link	Überblick an Linearführungen.
Maschinenbau	Material	2	Link	Materialauswahl für Chassis.
Maschinenbau	Roboterkinematik	7	Link	Roboterkinematik für fahrende Systeme inklusive Linienverfolgung.
Maschinenbau	Bewegungsarten	5	Link	Verschiedene Bewegungsarten für Roboter.
Maschinenbau	Robotik	6	Link	Grundlagen der Robotik.

Tabelle 34: Quellensammlung

J. Anforderungsliste Version 2

Die nachfolgende Tabelle zeigt die Anforderungsliste des Roboter-Fahrzeugs.

Legende

F = Festanforderung

M = Mindestanforderung

W = Wunschanforderung

Allgemeine Anforderungen

Nr.	F M W	Bezeichnung	Daten Werte Erläuterungen
1.1	W	Wettbewerb	Team 10 wird im Wettbewerb einen Podestplatz erreichen.
1.2	F	Wettbewerbsort	Voraussichtlich wird der Wettbewerb im Foyer der Mensa HSLU Technik und Architektur in Horw durchgeführt.
1.3	F	Projektabgabe	Der PREN 1 Schlussbericht ist bis zum 10. Januar 2025 abzugeben.
1.4	F	Eigenkonstruktion	Einzelne Systemkomponenten wie z.B. Räder, Servos, Motoren, Mikrocontroller, Kamera, etc. dürfen zugekauft und eingesetzt werden. Das zu realisierende Fahrzeug als Ganzes muss jedoch zwingend eine Eigenkonstruktion sein.
1.5	F	Software	Es dürfen Software-Komponenten und Software-Services von Fremd-Herstellern verwendet werden.
1.6	F	Eingriffe	Ein Eingreifen auf das Fahrzeug ist nach dem Start nicht mehr erlaubt.
1.7	F	Sicherheit	Das Team ist während sämtlichen Betriebs- und Test-Phasen verantwortlich für die Sicherheit des Fahrzeuges und den Schutz der Personen.
1.8	W	Nachhaltigkeit	Erfüllt Anforderungen nach SDG 12. Insbesondere die Unterziele 12.4 und 12.5.
1.9	W	Materialien Mechanik	Mechanikkonstruktionen vorzüglich aus wiederverwendbaren Materialien
1.10	W	Lieferwege	Wenn möglich Materialien und Kaufteile aus Europäischen Lagern beziehen.

Roboter

Nr.	F M W	Bezeichnung	Daten Werte Erläuterungen
2.1	F	Autonomität	Das Fahrzeug muss den vorgegebenen Parcours von Start bis Ziel ohne Zugriff von aussen absolvieren können.
2.2	F	Hardwarekomponenten	Alle zum Betrieb benötigten Hardware-Komponenten wie z.B. Sensoren, Aktoren, Steuergeräte, Kamera, etc. müssen sich im oder auf dem Fahrzeug befinden.
2.3	F	Softwarekomponenten	Alle Berechnungen und Software muss auf dem Roboter betrieben werden.
2.4	M	Betriebsbereitschaft	Das Fahrzeug muss innerhalb von maximal einer Minute im Startbereich manuell platziert und aufgebaut werden, sowie betriebsbereit sein.
2.5	W	Gesperrte Wegpunkte	Die gesperrten Wegpunkte sollten aus der Distanz vom Fahrzeug erkannt werden können.
2.6	W	Hindernis auf Strecke	Mögliche Hindernisse sollten vom Fahrzeug aus der Distanz erkannt werden können.
2.7	F	Hindernisbewältigung	Befährt das Fahrzeug eine Strecke mit einem Hindernis, so muss dieses erkannt und aktiv von der Strecke aufgenommen werden. Sobald das Fahrzeug die besagte Stelle passiert hat, muss das Hindernis wieder an die Ursprungsposition zurückgestellt werden. Die Toleranzzone beim Zurückstellen des Hindernisses beträgt 20 mm (umlaufend).
2.8	F	Auswahl Zielposition	Die Zielposition (1, 2 oder 3) muss am Fahrzeug mittels einem Wahlschalter ausgewählt werden können.
2.9	F	Startbefehl	Der Startbefehl wird mittels einem Schalter oder Taster am Fahrzeug erteilt. Gleichzeitig wird die Sicht auf die Strecke freigegeben und die Zeitmessung gestartet.
2.10	F	Leitlinien	Das Fahrzeug muss sich während des gesamten Parcours auf den vorgegebenen Leitlinien bewegen.
2.11	F	Not-Aus	Das Fahrzeug muss über einen leicht zugänglichen Not-Aus-Knopf oder -Schalter verfügen, der alle mechanisch-dynamischen Prozesse sofort unterbricht.
2.12	F	Gewicht	Das Fahrzeug darf das Maximalgewicht von 2 kg nicht überschreiten.
2.13	W	Schutzklasse	Mindestens IP-10 sollte gewährleistet sein.

Nr.	F M W	Bezeichnung	Daten Werte Erläuterungen
2.14	F	Dimensionen	Das Fahrzeug darf die Dimensionen 30×30 cm zu jeder Zeit, ausser beim Bewegen von Hindernissen, nicht überschreiten. Zudem ist die Höhe des Fahrzeugs (oder allfälliger Anbauteile) auf maximal 80 cm beschränkt.
2.15	F	Zielposition	Das Erreichen der Zielposition muss vom Fahrzeug in einer passenden Form visuell oder akustisch angezeigt werden. Zudem muss das Fahrzeug innerhalb eines Kreises von 30 cm Durchmesser um den Zielpunkt zum Stehen kommen.
2.16	W	Energieversorgung	Die Energieversorgung soll mit einem Akku realisiert werden, der über eine physische, steckbare Schnittstelle innerhalb von 6h wieder aufgeladen werden kann.
2.17	W	Akkulaufzeit	Im aktiven Betrieb des Fahrzeugs soll eine Akkulaufzeit von mindestens 25 Minuten gewährleistet sein.
2.18	W	Debug-Schnittstelle	Die Elektronik des Fahrzeugs soll über eine Debug-Schnittstelle verfügen, die es ermöglicht, aktuelle Zustände und Signale auszulesen.
2.19	W	Controlling-Schnittstelle	Die Elektronik des Fahrzeugs soll über Schnittstelle verfügen, über welche die Aktoren aktiv angesteuert werden können.
2.20	W	Zeitmessung	Das Roboter bietet die Möglichkeit die verstrichene Zeit seit Start anzugeben. Diese Zeitmessung wird optisch oder über eine Programmschnittstelle an das Team weitergegeben.

Parcours

Nr.	F M W	Bezeichnung	Daten Werte Erläuterungen
3.1	F	Wege-Netzwerk	Das Graph-Topologie und der Startknoten sind bekannt.
3.2	F	Zielpunkte	Die möglichen Zielpunkte sind bekannt, doch der definitive Zielpunkt wird erst unmittelbar vor dem Start des Parcours von der Juri bekannt gegeben und ist vorher nicht bekannt.
3.3	F	Wegpunkte	Insgesamt gibt es acht Wegpunkte. Die Wegpunkte sind aufgeklebte Vollkreise (weiss) mit einem Durchmesser von 7 bis 12 cm.
3.4	F	Untergrund	Der Untergrund entspricht dem Bodenbelag des Foyers der Mensa auf dem Campus der Hochschule Luzern für Technik und Architektur in Horw.
3.5	F	Leitlinien	Die Wegpunkte sind mit hellen Leitlinien (aufgeklebtes Klebeband) verbunden. Die Breite der Leitlinien beträgt ca. 20 mm.
3.6	F	Abmessungen	Der Abstand der Wegpunkte ist variabel zwischen 0.5 bis 2.0 m. Die Gesamtfläche des Wege-Netzwerkes beträgt ca. 4.5 x 4.5 m.
3.7	F	Gesperrte Wegpunkte	Die gesperrten Wegpunkte dürfen nicht befahren werden. Sie sind bis zum Start unbekannt und mittels einem Leitkegel gekennzeichnet.
3.8	F	Hindernis auf Strecke	Die Strecke darf befahren werden, doch das Hindernis muss aktiv von der Strecke aufgenommen und am gleichen Ort wieder zurückgestellt werden.
3.9	F	Nicht vorhandene Teilstrecken	Leitlinien können aus dem Wege-Netzwerk entfernt werden. Die entsprechenden Verbindungen können nicht befahren werden.
3.10	F	Streckenbedingungen	Die Streckenbedingungen (Sperrung, Hindernisse, nicht vorhandene Teilstrecke) sind bis zum Start unbekannt.
3.11	F	Startbereich	Die Grösse des Startbereichs beträgt 30 x 30 cm. Das Fahrzeug darf diese Dimensionen nicht überschreiten.
3.12	F	Start	Sobald die Sicht auf die Strecke freigegeben wird, beginnt ebenfalls die Zeitmessung.
3.13	M	Parcours-Laufzeit	Die Laufzeit von Start bis Ziel darf maximal vier Minuten betragen. Wird das Ziel innert vier Minuten nicht erreicht, ist der Lauf ungültig.

Simulation

Nr.	F M W	Bezeichnung	Daten Werte Erläuterungen
4.1	F	Betriebssystem	Die Simulation muss auf Windows ausführbar sein.
4.2	W	Betriebssystem	Die Simulation soll auf Linux und auch Windows ausführbar sein.
4.3	W	Benutzeroberfläche	Den Wegstrecken und Netzwerknoten können beliebige Streckenereignisse zugewiesen werden.
4.4	M	Darstellung	Die Simulation muss 2-dimensional dargestellt werden.
4.5	W	Darstellung	Die Simulation kann 3-dimensional dargestellt werden.
4.6	W	Pfadfindungsalgorithmen	In der Simulation sollen verschiedene Pfadfindungsalgorithmen (z.B. Dijkstra, A*-Algorithmus, etc.) implementiert werden für eine direkte Gegenüberstellung in Zuverlässigkeit und Schnelligkeit.
4.7	W	Zeitauswertung	In der Simulation soll eine approximierte Zeitauswertung, basierend auf heuristischen Abschätzungen und vorgegebene Fahrzeugparameter, möglich sein.
4.8	W	Visualisierung des Pfades	Der vorgeplante Pfad soll während der Simulation angezeigt werden, um das Verhalten des Fahrzeugs besser nachvollziehen zu können.
4.9	W	Fahrzeugparameter	Fahrzeugparameter (Geschwindigkeit, Wendekreis, Sensorreichweite, etc.) sollen editierbar sein.
4.10	W	Datenexport	Die Daten, welche während der Simulation generiert werden, sollen exportierbar sein.
4.11	W	Error-Handling	Der Simulator muss robust auf Fehler reagieren und darf keinesfalls abstürzen. Zudem sollen Fehlerzustände abgefangen und klar dokumentiert werden.

Herstellungsressourcen

Nr.	F M W	Bezeichnung	Daten Werte Erläuterungen
5.1	W	Materialbeschaffung	Materialien und Komponenten sollen vorzugsweise von folgenden Lieferanten bestellt werden: - Conrad Electronic - Distrelec - Mädler - Farnell - Mouser - DigiKey - Reichelt
5.2	F	Budget	Für die Realisierung des Projekts stehen dem Team insgesamt 500 CHF zur Verfügung. Davon dürfen maximal 200 CHF in PREN 1 ausgegeben werden.
5.3	F	Normteile ab HSLU Lagerbestand	Normteile (Schrauben, Lager, Rohmaterial, Widerstände, Kondensatoren, etc.) aus dem HSLU Lagerbestand dürfen kostenlos verwendet werden.
5.4	F	Persönlicher Drucker	Wird für das Projekt ein persönlicher 3D-Drucker verwendet, so muss die verarbeitete Menge ausgewiesen werden.
5.5	F	Herstellungsressourcen der HSLU	Dem Team stehen für die Umsetzung des Projekts (PREN 1 und PREN 2) die folgenden Ressourcen der HSLU zur Verfügung: - maximal 25 h Maschinenlaufzeit der 3D-Drucker - maximal 1 h Maschinenlaufzeit des Lasergeräts - maximal 10 Arbeitsstunden des Werkstattpersonals Elektrotechnik - maximal 10 Arbeitsstunden des Werkstattpersonals Maschinentechnik

Erfüllungsgrad Anforderungen

Nr.	F M W	Bezeichnung	Erfüllt ?
1.1	W	Wettbewerb	offen
1.2	F	Wettbewerbsort	ja
1.3	F	Projektabgabe	ja
1.4	F	Eigenkonstruktion	ja
1.5	F	Software	ja
1.6	F	Eingriffe	ja
1.7	F	Sicherheit	offen
1.8	W	Nachhaltigkeit	ja
1.9	W	Materialien Mechanik	ja
1.10	W	Lieferwege	offen

Roboter

Nr.	F M W	Bezeichnung	Erfüllt ?
2.1	F	Autonomität	ja
2.2	F	Hardwarekomponenten	ja
2.3	F	Softwarekomponenten	ja
2.4	M	Betriebsbereitschaft	ja
2.5	W	Gesperrte Wegpunkte	offen
2.6	W	Hindernis auf Strecke	offen
2.7	F	Hindernisbewältigung	ja
2.8	F	Auswahl Zielposition	ja
2.9	F	Startbefehl	ja
2.10	F	Leitlinien	ja
2.11	F	Not-Aus	ja
2.12	F	Gewicht	offen
2.13	W	Schutzklasse	offen
2.14	F	Dimensionen	ja
2.15	F	Zielposition	ja
2.16	W	Energieversorgung	ja
2.17	W	Akkulaufzeit	offen
2.18	W	Debug-Schnittstelle	offen
2.19	W	Controlling-Schnittstelle	ja
2.20	W	Zeitmessung	nein

Parcours

Nr.	F M W	Bezeichnung	Erfüllt ?
3.1	F	Wege-Netzwerk	ja
3.2	F	Zielpunkte	ja
3.3	F	Wegpunkte	ja
3.4	F	Untergrund	ja
3.5	F	Leitlinien	ja
3.6	F	Abmessungen	ja
3.7	F	Gesperrte Wegpunkte	ja
3.8	F	Hindernis auf Strecke	ja
3.9	F	Nicht vorhandene Teilstrecken	ja
3.10	F	Streckenbedingungen	ja
3.11	F	Startbereich	ja
3.12	F	Start	ja
3.13	M	Parcours-Laufzeit	offen

Simulation

Nr.	F M W	Bezeichnung	Erfüllt ?
4.1	F	Betriebssystem	ja
4.2	W	Betriebssystem	ja
4.3	W	Benutzeroberfläche	ja
4.4	M	Darstellung (2D)	ja
4.5	W	Darstellung (3D)	nein
4.6	W	Pfadfindungsalgorithmen	ja
4.7	W	Zeitauswertung	ja
4.8	W	Visualisierung des Pfades	ja
4.9	W	Fahrzeugparameter	teilweise
4.10	W	Datenexport	ja
4.11	W	Error-Handling	ja

Herstellungsressourcen

Nr.	F M W	Bezeichnung	Erfüllt ?
5.1	W	Materialbeschaffung	ja
5.2	F	Budget	ja
5.3	F	Normteile ab HSLU Lagerbestand	ja
5.4	F	Persönlicher 3D-Drucker	ja
5.5	F	Herstellungsressourcen der HSLU	ja

K. Morphologischer Kasten

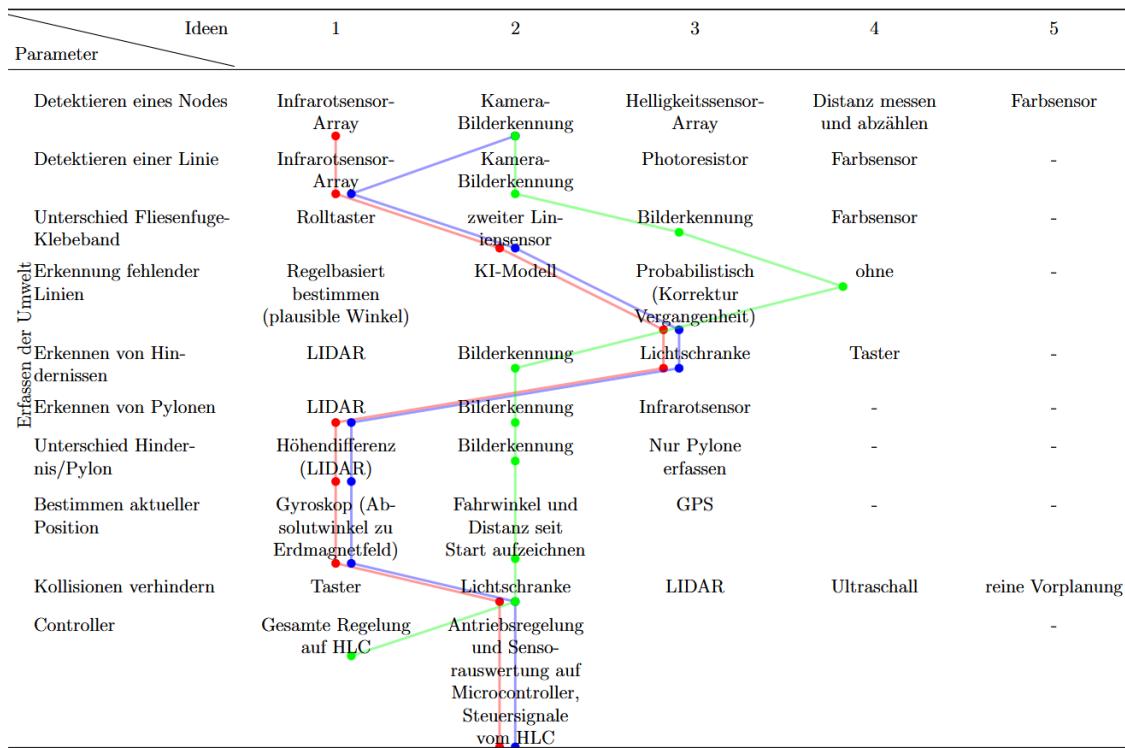


Abbildung 78: Morphologischer Kasten Gerät

	Ideen	1	2	3	4	5
Parameter						
Navigation	Weg durch Netz bestimmen	Wege gewichten	Wege vordefinieren	Brute force	bis zum nächsten Knoten vorrausschauen	-
Ansatz der Kartenerkennung	Rein auf auftretende Ereignisse reagieren	Bilderkennung und Route vorplanen	Nächste Knoten mit LIDAR vorplanen	Probabilistischer Algorithmus und Graph aufzeichnen	-	-
Energie	Li-ion Akku	Ni-Cd Akku	Bleisäure Akku	Kraftstofftank	Druckluft	-
Ladestandsanzeige	Mehrere LED's	1 LED (Akku low)	Feedback WebUI	Analog	LCD 32x4	-
Ziel	Piezo-Alarm	Lautsprecher-Ton	Optische Anzeige (LED's)	LCD Display	-	-
Zielerkennung	Bilderkennung (Beschriftung)	Reine Navigation	-	-	-	-

Abbildung 79: Morphologischer Kasten Gerät

	Ideen	1	2	3	4	5
Parameter						
Fortbewegungsmittel	Gummiräder	Mecanumräder	4 Beine (gehend)	Propeller (fliegend)	Gummiraupen	-
Antrieb	DC-Motor	Dampfmaschine	Schrittmotor-Bidirektional	Stirlingmotor	Verbrennungsmotor	-
Fahrwerk Lenkung	2 Rädrig (Achse mittig) mit 2 Kugeln	2 Rädrig mit einer Kugel	3 Rädrig mit lenkendem Einzelrad	4 Rädrig mit lenkender Achse	Panzerlenkung	-
Hindernis Aufnahmeverrichtung	von oben ohne Winkelausrichtung	von oben mit Winkelausrichtung	von oben frei rotierend und Arretierung	Hindernis seitlich aufnehmen	durch die mittleren Löcher aufnehmen	-
Chassis	Gabelstapler	Klemmgreifer	Saugnapf	Schnappverschluss	-	-
Hindernis aufnehmen	Über das Fahrzeug drüber	Um das Fahrzeug herum	Linearförderung durch das Fahrzeug	Fahrzeug 180° Drehung	Fahrzeug 180° Drehung (symmetrisch)	-
Hindernis-Handling	anhand gefahren-der Distanz und Liniensensor	per Kamera	-	-	-	-
Hindernis positionieren	Roboterarm	Linearmodul	Hebebühne	Pneumatik	Hubmagnet	-
Z-Positionierung	Buzzer	Taster	-	-	-	-
Notstopp	-	-	-	-	-	-

Abbildung 80: Morphologischer Kasten Gerät

Parameter \ Ideen	1	2	3	4	5
Betriebssystem	Windows	Linux	MacOS	Alle OS	-
Darstellung	2-D	3-D	mixed Reality	2-D mit Sensorfeedback	-
Pfadfindungsalgo	Mehrere Algos	A*-Algorithmus	D*Lite	Partikelfilter	-
Fahrzeugparameter	editierbar	fix	lernfähige	dynamisch	-
Daten-Export	Log-File	CSV	SQ-Lite	JSON	-
Programmiersprache Frontend	JavaScript	Java (JavaFX)	C++(QT)	Python(Tkinter)	-
Programmiersprache Backend	JavaScript	Java	C++	Python	-
UI Technologie	Webapplikation	Desktop App	CLI	3-D Simulation mit Interaktion	-
Auswertung	Diagramme in MS Excel	Log-Auswertung im Tool	Live Dashboard mit Updates	-	-
Simulationsgeschwindigkeit	Echtzeit	Zeitraffer	Verlangsame Bewegung	Anpassbar	-
Interaktionsmöglichkeiten	Passiv beobachten	Eingreifen per Klick	Eingreifen via Konsole	Vollinteraktives Umfeld	-
Fahrverhalten	konstant	dynamisch (anpassbar)	lernfähig	-	-

Abbildung 81: Morphologischer Kasten Simulation

L. Nutzwertanalyse

	Kriterien	Gewichtung {%}	Punkte {1...10}	Punkte gewichtet	Punkte {1...10}	Punkte gewichtet	Punkte {1...10}	Punkte gewichtet
Gesamt- system			rote Linie		grüne Linie		blaue Linie	
	Einrichtungsaufwand	15	8	12	5	7.5	6	9
	Entwicklungsaufwand	25	6	15	3	7.5	8	20
	Geschwindigkeit	20	7	14	5	10	9	18
	Kosten	30	8	24	4	12	6	18
	Nachhaltigkeit	10	5	5	3	3	4	4
	Summe	100		70		40		69
Antrieb			Schrittmotor		DC-Motor		Schrittmotor	
	Geschwindigkeit	20	8	16	8	16	8	16
	Verhältnis Leistung zu Gewicht	30	4	12	7	21	4	12
	Wendefähigkeit	20	8	16	10	20	8	16
	Genauigkeit	30	7	21	4	12	7	21
	Summe	100		65		69		65

Abbildung 82: Nutzwertanalyse Gerät

Hindernis- handling			Greifer ohne Ausrichtung		Gabelstapler		Greifer mit Ausrichtung	
	Platziergenauigkeit	25	7	17.5	4	10	8	20
		Sicherheit	45	6	27	7	31.5	6
		Robustheit	30	8	24	7	21	6
		Summe	100		68.5		62.5	65
Energie- versorgung			Li-ion Akku		Ni-Cd Akku		Li-ion Akku	
	Laufzeit	40	9	36	7	28	9	36
	Gewicht	60	9	54	7	42	9	54
	Summe	100		90		70		90
Umwelt- detektion			Lidar		Bilderkennung		Bilderkennung	
	Detektion der Fahrlinie	30	9	27	7	21	7	21
	Positionsbestimmung	15	6	9	2	3	8	12
	Detektion der Pylonen / Hindernisse	25	8	20	6	15	6	15
	Abhängigkeit von Umwelteinflüssen	30	8	24	3	9	3	9
	Summe	100		80		48		57
Gesamtsumme				74.7		57.9		69.2

Abbildung 83: Nutzwertanalyse Gerät

	Kriterien	Gewichtung {%}	Punkte {1...10}	Punkte gewichtet	Punkte {1...10}	Punkte gewichtet
Plattform				Nativ	Web-based	
	Entwicklungs-aufwand	40	2	8	8	32
	Performance	20	7	14	4	8
	Bedienbarkeit	40	4	16	7	28
	Summe	100		38		68
Visualisier-ung				3D	2D	
	Entwicklungs-aufwand	40	3	12	8	32
	Interaktions-möglichkeit	10	6	6	5	5
	Realitätsnähe	20	8	16	3	6
	Summe	100		49		58
Auswertung				In-tool	Export + Scripts	
	Entwicklungs-aufwand	30	5	15	5	15
	Flexibilität	40	4	16	6	24
	Integration in Dokumentati-on	30	2	6	8	24
	Summe	100		37		63
Wieder-verwendbar-keit				Splitting Front-/Backend	Kein Splitting	
	Entwicklungs-aufwand	40	3	12	7	28
	Austauschbar-keit	30	8	24	2	6
	Transpilation	30	8	24	2	6
	Summe	100		60		40
Gesamtsumme				46	57.25	

Abbildung 84: Nutzwertanalyse Simulation

M. Digitaler Anhang

Dem Abgabedokument beigelegt ist ein digitaler Anhang. Darin befinden sich Schema und Layout Daten zu den verschiedenen Leiterplatten sowie Konstruktionsdaten und CAD-Modelle des aktuellen Arbeitsstands.

Das Verzeichnis ist der README Datei zu entnehmen, welche sich im Anhang befindet.