

HOCHSCHULE LUZERN

PREN 1 TEAM 10

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

## Abgabe Meilenstein 3

6. Dezember 2024

**Versionsverlauf**

| Version | Datum      | Verfasser | Änderungen    |
|---------|------------|-----------|---------------|
| 1.0     | 06.12.2024 | Team 10   | Erste Fassung |

## Inhaltsverzeichnis

|  |           |
|--|-----------|
| <b>1. Einleitung</b>   | <b>1</b>  |
| 1.1. Aufgabenstellung . . . . .                                    | 1         |
| <b>2. Produktbeschreibung</b>                                      | <b>2</b>  |
| 2.1. Produktbeschreibung . . . . .                                 | 2         |
| 2.2. Hardfacts . . . . .   | 3         |
| <b>3. Entwicklung &amp; Dimensionierung</b>                        | <b>4</b>  |
| 3.1. Mechanik . . . . .  | 4         |
| 3.1.1. Chassis und Fahrwerk . . . . .                              | 4         |
| 3.1.2. Greifeinheit . . . . .                                      | 6         |
| 3.2. Elektronik . . . . .  | 7         |
| 3.2.1. Steuerungstopologie und Gesamtübersicht . . . . .           | 7         |
| 3.2.2. Bordnetz . . . . .  | 8         |
| 3.2.3. Antriebe und Dimensionierung . . . . .                      | 10        |
| 3.2.4. Liniensor . . . . .   | 14        |
| 3.2.5. Liniensor als PCB . . . . .                                 | 16        |
| 3.2.6. Abstandssensoren . . . . .                                  | 17        |
| 3.2.7. Sensorik Strecken Rückverfolgung . . . . .                  | 18        |
| 3.3. Informatik . . . . .  | 21        |
| 3.3.1. Beschreibung & Entscheid Algorithmen . . . . .              | 21        |
| 3.3.2. Einbindung in Roboter . . . . .                             | 23        |
| 3.3.3. Simulation . . . . .  | 25        |
| 3.3.4. Tests . . . . .   | 30        |
| <b>4. Projektmanagement</b>  | <b>31</b> |
| 4.1. Organisation . . . . .  | 31        |
| 4.2. Projektplanung . . . . .                                      | 31        |
| 4.3. Risikoanalyse . . . . .                                       | 33        |
| <b>5. Nachhaltigkeit</b>   | <b>34</b> |
| 5.1. Nachhaltigkeitsbetrachtung . . . . .                          | 34        |
| 5.1.1. Ziele der Nachhaltigkeitsbetrachtung . . . . .              | 34        |
| 5.1.2. Abgrenzung . . . . .  | 34        |
| 5.1.3. Bezug zu den Sustainable Development Goals (SDGs) . . . . . | 34        |
| 5.2. Entscheide . . . . .  | 35        |
| 5.2.1. Elektronik . . . . .  | 35        |
| 5.2.2. Mechanik . . . . .  | 35        |
| 5.2.3. Informatik . . . . .  | 35        |
| 5.3. Reflexion . . . . .   | 36        |
| <b>6. Schlussdiskussion</b>  | <b>37</b> |
| 6.1. Erfahrungen & Lessons learned . . . . .                       | 37        |
| 6.2. Erfüllungsgrad Anforderungsliste . . . . .                    | 38        |
| 6.3. Fazit und Ausblick . . . . .                                  | 39        |
| <b>7. Verzeichnisse</b>  | <b>40</b> |
| 7.1. subsection . . . . .  | 40        |

|   |           |
|---|-----------|
| <b>8. Anhang</b>                                    | <b>41</b> |
| <b>A. Anforderungsliste Version 2</b>               | <b>41</b> |
| A.1. Allgemeine Anforderungen . . . . .             | 41        |
| <b>B. Konzepterstellung Greifereinheit</b>          | <b>47</b> |
| <b>C. Konzepterstellung Energieversorgung</b>       | <b>49</b> |
| <b>D. Konzepterstellung Antriebe</b>                | <b>57</b> |
| <b>E. Konzepterstellung Hinderniserkennung</b>      | <b>61</b> |
| <b>F. Konzepterstellung Strecken Rückverfolgung</b> | <b>66</b> |
| <b>G. Liniensensor</b>                              | <b>68</b> |
| <b>H. Risikomanagement</b>                          | <b>74</b> |
| <b>I. Blockschaltbild Elektrokonzept</b>            | <b>82</b> |

## 1. Einleitung

### 1.1. Aufgabenstellung

Die aktuelle Aufgabenstellung des Moduls Produktentwicklung PREN 1 der Hochschule Luzern stellt uns als interdisziplinäres Team vor die Herausforderung, ein autonomes Fahrzeug zu entwickeln, das sich in einem vorgegebenen Wegenetzwerk optimal navigieren kann. Ziel ist es, das Fahrzeug so zu konzipieren, dass es Hindernisse erkennt, gesperrte Bereiche meidet und unter Berücksichtigung der vorab unbekannten Einschränkungen die kürzeste Route vom Start- zum Zielpunkt findet.

Unser Team setzt sich aus Studierenden der Studiengänge Elektrotechnik, Informatik und Maschinenbau zusammen, was uns die Möglichkeit bietet, verschiedene technische und methodische Kompetenzen zu vereinen. In der Projektphase von PREN 1 entwickeln wir ein Gesamtkonzept, das auf einem morphologischen Kasten basiert und verschiedene Lösungsvarianten systematisch vergleicht. Ziel ist es, die technische Machbarkeit zu bewerten und erste funktionale Prototypen von Teilsystemen zu erstellen, die als Validierungsbasis für das weiterführende Modul PREN 2 dienen.

Die Aufgabe erfordert die Entwicklung eines autonomen Systems, das auf einem Netzwerk aus Leitlinien operiert. Zu den wichtigsten Anforderungen gehören:

- **Erkennung gesperrter Wegpunkte:** Diese sind durch Pylonen gekennzeichnet und müssen vom Fahrzeug selbstständig detektiert werden.
- **Bewältigung von Hindernissen:** Das Fahrzeug soll Hindernisse aktiv von der Strecke entfernen und diese an die ursprüngliche Position zurückstellen.
- **Anpassung an veränderte Bedingungen:** Nicht vorhandene Streckenabschnitte müssen vom System zuverlässig als nicht passierbar erkannt werden.
- **Autonome Zielfindung:** Über eine Zielauswahl vor dem Start (Positionen A, B oder C) soll das Fahrzeug den kürzesten und effizientesten Weg zum Zielpunkt 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 zentrale wissenschaftliche Herausforderung für unser Team besteht darin, Sensorik, Elektronik und Algorithmen so zu entwickeln und zu kombinieren, dass eine zuverlässige Wegfindung und Hindernisbewältigung gewährleistet wird. Dabei legen wir besonderen Wert auf eine methodische Herangehensweise, die neben technischer Präzision auch Nachhaltigkeitsaspekte und Ressourceneffizienz in den Entwicklungsprozess integriert.

Diese Aufgabe bietet uns die Möglichkeit, theoretisches Wissen praktisch anzuwenden und gleichzeitig Kompetenzen in interdisziplinärer Zusammenarbeit und systematischer Produktentwicklung zu vertiefen.

## 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, Abstandssensoren und Kameras ermöglicht es dem Fahrzeug, die Leitlinien des Wegenetwerks zu verfolgen, Hindernisse zu erkennen und die Position genau zu bestimmen. 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.

## 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 behandelt die Evaluierung des Fahrwerks in Bezug auf die Radauswahl und die Chassis-Konstruktion. Dabei werden verschiedene Konzeptlösungen diskutiert und getestet. Abschliessend wird die finale Entscheidung auf Basis der Testergebnisse dokumentiert.

#### Anforderungen

##### Gewicht:

Das Gewicht stellt, wie auch bei allen anderen Baugruppen, einen kritischen Faktor dar. Daher basiert das Chassis auf dem Konzept einer einzigen Grundplatte, anstatt einen Volumenkörper zu entwerfen, der sämtliche Aktoren, Sensoren und Steuerungskomponenten umschliesst. Wie eine Abdeckung für das Gerät aussehen wird, steht noch offen. Zudem wird angestrebt, die Konstruktion auf zwei Räder zu beschränken. Die Drehbewegung wird mittels Skid Steering realisiert. Um die Stabilität zu gewährleisten, benötigt der Roboter jedoch einen dritten Auflagepunkt, dessen genaue Umsetzung im Nachfolgemodul PREN 2 durch verschiedene Tests untersucht wird.

##### Geschwindigkeit:

Da die Motoren bereits ausgewählt sind und ihre Drehzahl technisch begrenzt ist, kann die maximale Fahrgeschwindigkeit lediglich über die Radgrösse definiert werden. Von einem Getriebe wird bewusst abgesehen, um Gewicht und Kosten zu reduzieren.

##### Kosten:

Die Anbauteile des Chassis werden mit einem 3D-Drucker gefertigt, da dem Team 25 Stunden Druckzeit des HSLU-T&A-Druckers kostenfrei zur Verfügung stehen. Die Kosten für die Räder sollen unter 20 CHF bleiben, um ein grösseres Budget für kritische Funktionen bereitzustellen.

#### Konzeptionierung

Die Grundplatte des Chassis wird aus einer MDF-Platte gefertigt, in die sämtliche Ausschnitte und Öffnungen mittels Lasergravur eingebracht werden. Komponenten werden modular mit Anbauteilen aus dem 3D-Druck auf der Grundplatte montiert. Diese modulare Konstruktion ermöglicht einfache Anpassungen und hilft, das Gewicht zu reduzieren. Bereits früh in der Projektphase wurde entschieden, dass die Fortbewegung des Pfadfinders mit zwei Rädern und einem dritten Auflagepunkt realisiert werden soll. Abbildung 1 zeigt eine erste Konzeptskizze des Chassis, einschliesslich der Räder und einer Rutschfläche als drittem Auflagepunkt.

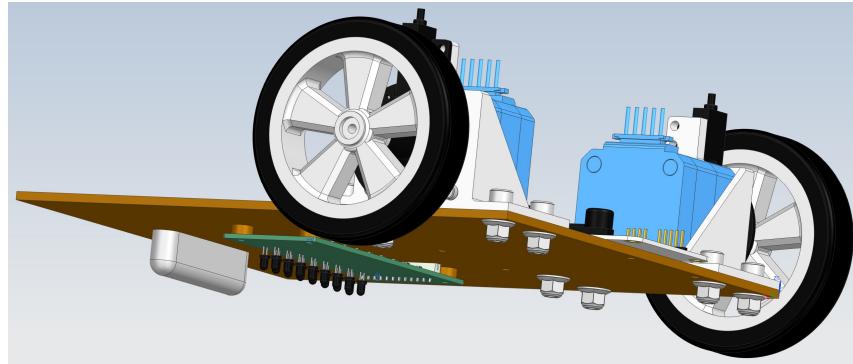


Abbildung 1: Chassis Konzept in Siemens NX

Die Räder dürfen eine maximale Breite von 20 mm nicht überschreiten, um ausreichend Platz auf der Grundplatte für Motoren und Motorentreiber bereitzustellen. Der Mindestdurchmesser beträgt 80 mm, da andernfalls die Liniensensoren nicht wie vorgesehen montiert werden können. Zudem ermöglicht ein grösserer Durchmesser von 80 mm eine maximale Geschwindigkeit von 1.676 m/s, was den Projektanforderungen entspricht. Die Berechnung erfolgt gemäss folgender Formel:

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

Auf Grundlage der genannten Anforderungen wurde auf verschiedenen Webseiten nach passenden Rädern gesucht. Die Ergebnisse sind in Tabelle 3 zusammengefasst.

| Hersteller | Herst. Nr. | Preis    | Durchmesser | Material Lauffläche | Gewicht |
|------------|------------|----------|-------------|---------------------|---------|
| ServoCity  | 595660     | 9.05 CHF | 80.3 mm     | Silikon             | 19 g    |
| DF-Robot   | FIT0500    | 2.62 CHF | 80 mm       | Silikon             | -       |

Tabelle 3: Rad Parameter

### Entscheidung und Fazit

Von den beiden untersuchten Rädern wurde das Modell FIT0500 von DF-Robot ausgewählt. Obwohl die Aufnahme des ServoCity-Rades technisch besser geeignet wäre, wurden die erheblichen Lieferkosten von 70 CHF als entscheidender Nachteil gewertet.

**3.1.2. Greifeinheit**

Entscheid Freitag bezüglich Konzept 1 oder Konzept 2 sonstiges  
Platzierung von Kamera, Endschalter/Lichtschranke für Motor, Lichtschranke für Hindernis,  
Bilder in Anhang (Anhang Dokumentation fast vollständig)

### 3.2. Elektronik

#### 3.2.1. Steuerungstopologie und Gesamtübersicht

Anhang I zeigt die Gesamtübersicht der Steuerung zum Zeitpunkt von PREN1. 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.

#### 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 ausserdem 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 laufende Motoren, Schaltregler und andere Leistungselektronik auf engem Raum mit Kommunikations- und anderer feinfühliger Elektronik verbaut werden müssen, wird die UART-Kommunikation über eine RS422-Schnittstelle differentiell übertragen. Die Kommandos und das Protokoll werden erst im Nachfolgemodul beim Inbetriebnehmen dieser PCBs definiert. Angedacht ist eine Kommunikation über eine Kommandolänge von beispielsweise 32- oder 64-Bit, worin eine Adresse definiert ist, ein Kommando und darüber

hinaus noch genügend Platz für zu übertragende Parameter. Mögliche Parameter können dabei eine Fahrzeuggeschwindigkeit, ein Drehwinkel oder etwa Errorcodes, welche zurück zum Raspberry Pi übertragen werden, sein.

Die Adressenvergabe ist geplant, da der Raspberry Pi auf Basis dieses Blockschaltbilds grundsätzlich nicht in der Lage ist, auf direktem Wege mit dem Greifcontroller zu kommunizieren. Dies muss er grundsätzlich auch nicht. Um allerdings trotzdem eben diesen Fall abzudecken, sollen Adressen von 0...2 (2-Bit) vergeben werden. So kann eine einfache Punkt-zu-Punkt-Übertragung ausreichend sein, um die drei Steuerungen miteinander kommunizieren zu lassen.

## Software

!!! (ABBILDUNG ERSTELLEN GEPLANTE TASKS IN FREERTOS / EMBEDDED SOFTWARE) !!!

## Boardnetz

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 beschäftigt sich mit der Energieversorgung des Pfadfinders. Wie die Evaluation der Energieversorgung ergeben hat, siehe Anhang C, wird der Roboter über einen 4S-LiPo mit Energie versorgt. Der Evaluierte Akku beinhaltet  $\approx 19Wh$  Energie und bietet damit genug Energie, um den Roboter im Worst-Case bei voller Leistung für ca. 20min zu betreiben. Die Bordnetzspannung ist auf 12V gesetzt, um auf Sensorik und Aktorik, so zum Beispiel Lichtschranken, aus dem Industriellen Umfeld zurückgreifen zu können. Außerdem kann so eine grössere Bandbreite an Motoren abgedeckt werden, falls sich die eingeplanten als nicht geeignet erweisen würden.

Die Ausgangsspannung von 14.4V wird auf dem selbst entwickelten DC-PowerBoard mittels eines Abwärtsstellers auf 12V geregelt. Dieses Board bietet außerdem die Möglichkeit, über ein Netzteil in einem Bereich von 12 bis 18 V versorgt zu werden und dabei den Akku selbstständig zu trennen. Dadurch ist der Roboter bei Entwicklungs- und Einrichtungsaufgaben nicht zwingend auf die Energieversorgung via Batterie angewiesen, sondern kann auch über ein Netzteil betrieben werden.

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

Die Batterieschutzbeschaltung überwacht die Zellspannungen des LiPo-Akkus jeweils getrennt und trennt die Batterie vom Roboter, falls eine der Zellen in den Unterspannungsbereich gerät. Dieser Zustand wird über eine rot leuchtende LED signalisiert. Die Zell-Balancierung ist nicht auf dem Roboter selbst umgesetzt. Für diesen Zweck ist ein Ladegerät aus dem Modellbaubereich gekauft, welches die Zellen aktiv ausbalancieren kann.

Das Schema dieses PCB's, sowie sein Layout sind im Anhang beigelegt. Abbildung 2 zeigt eine 3D-Ansicht dieses PCB's.

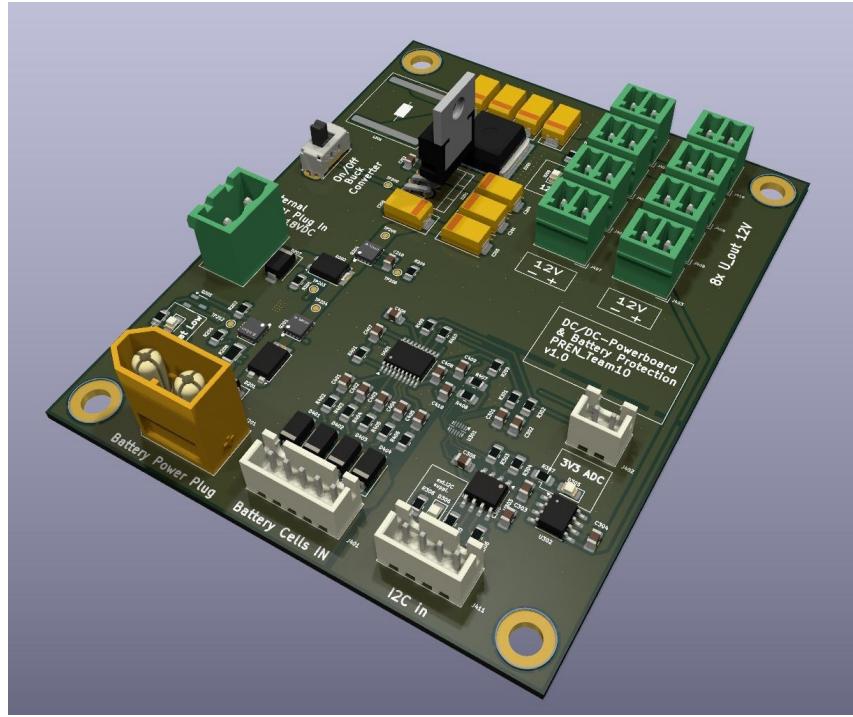


Abbildung 2: Ansicht PowerBoard

**Bordnetz - Spannungsverteilung** Jeder PCB wird über 12V mit Spannung versorgt. Auf den entsprechenden PCB's werden die geregelten 12V zuerst über einen DC-DC Abwärtssteller auf  $\approx 6V$  heruntergebracht und dann über einen LDO auf 5V bzw. 3V3 aktiv gefiltert. Dadurch wird verhindert, dass sich Spannungsspitzen oder -ripples durch zum Beispiel die anlaufenden Motoren im gesamten System verteilen und andere Elektronik stört. Für die LDOs wird ein PSSR angestrebt, welcher bei entsprechenden Taktraten der Abwärtssteller eine angemessene Dämpfung von mindestens 20dB aufweist. Das Blockschaltbild in Abbildung 3 zeigt die Spannungsverteilung konzeptionell.

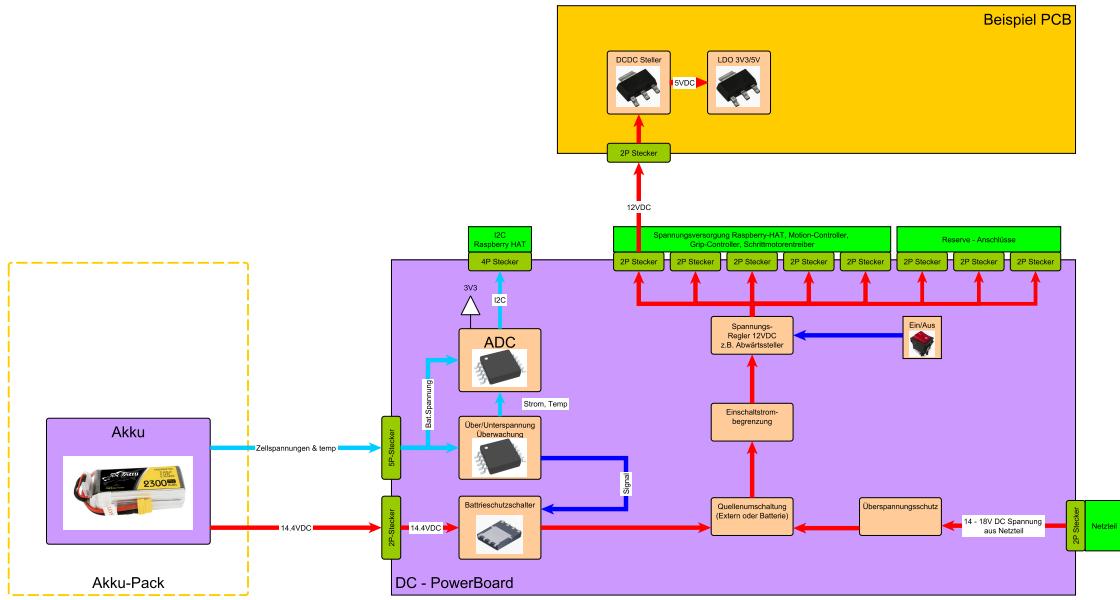


Abbildung 3: PowerBoard Blockschaltbild

### 3.2.3. Antriebe und Dimensionierung

Dieser Abschnitt beschäftigt sich mit der Auswahl eines passenden Antriebs mitsamt Steuerung für den Roboter. Die Evaluation der Antriebe, siehe Anhang D, hat ergeben, dass ein Schrittmotor der Firma DFRobot, siehe Abbildung 4, eingesetzt wird. Dieser ist in der Lage, mit 1.7A den Roboter auf bis zu  $2\frac{m}{s}$  zu beschleunigen.



Abbildung 4: Schrittmotor

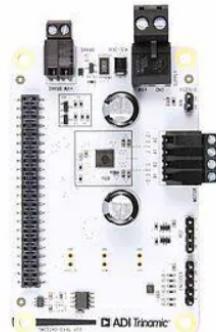


Abbildung 5: Evaluationboard TMC5240

Angesteuert werden diese Motoren über einen Vollintegrierten Schrittmotorentreiber der Firma ADI-Trinamic. Um weniger Entwicklungsaufwand zu haben, wird auf 2 Evaluation-Boards des Treiber-IC's *TMC-5240* zurückgegriffen, gezeigt in Abbildung 5. Eines der Team-Mitglieder hat bereits Erfahrungen mit diesem speziellen Treiber und kann auf entsprechende Treiber-Evaluation-Boards aus seinem beruflichen Umfeld zurückgreifen. Die Abbildung 6 zeigt schematisch, wie diese Treiber angesteuert werden.

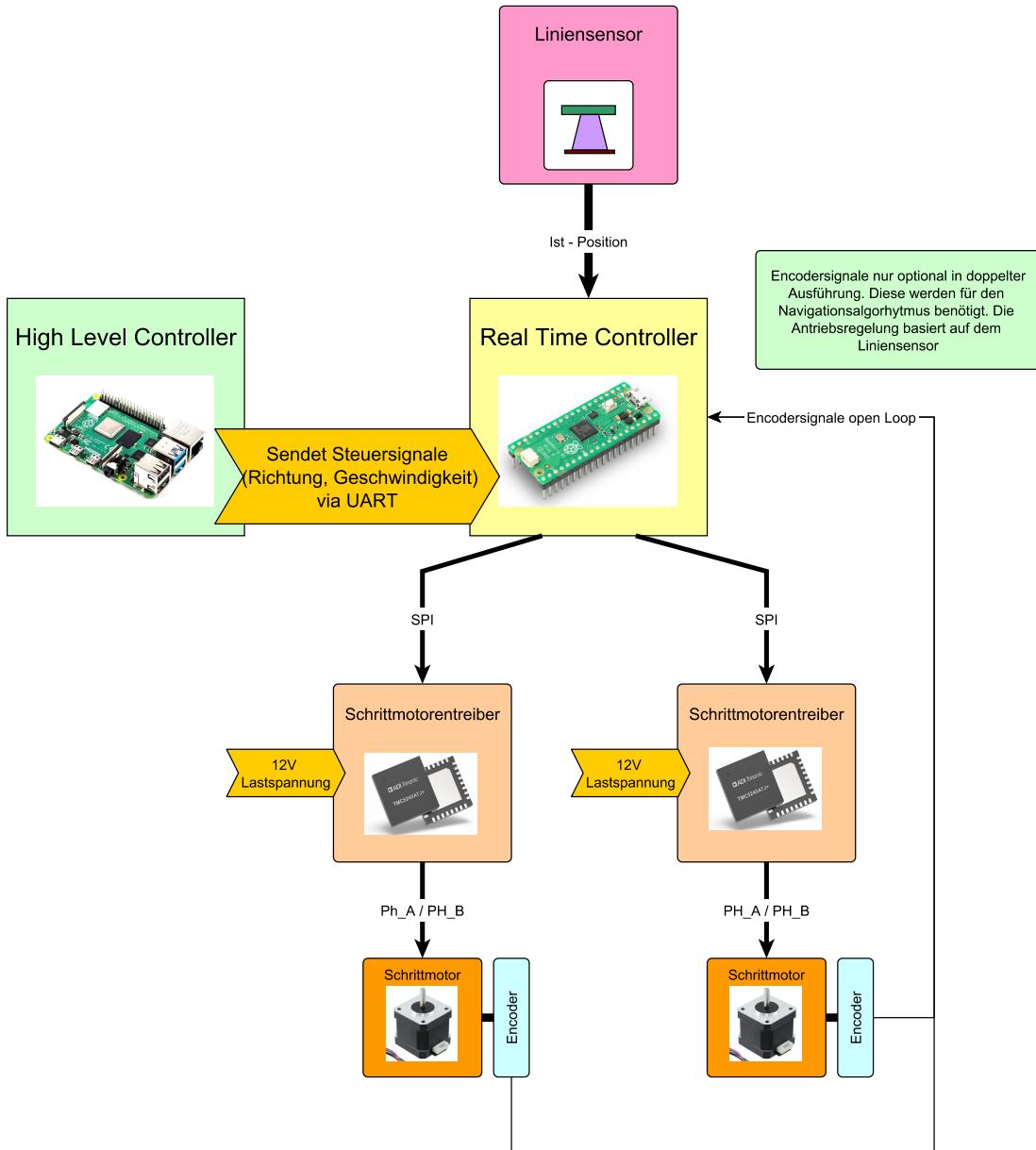


Abbildung 6: Ansteuerungstopologie Schrittmotoren

Die Antriebsregelung wird im Sinne der Gewaltenteilung auf einem eigenen PCB umgesetzt. Eingangssignale für diesen sind der Liniensensor und 1 oder sogar 2 Encoder. Die Regelung findet allerdings ohne Encoder statt. Die Eingangsgröße dafür ist lediglich die Position des Fahrzeugs auf der Linie. Angesteuert werden die Schrittmotorentreiber über den SPI-Bus des Mikrocontrollers.

Die Encoder werden benötigt, damit der Navigationscomputer nachvollziehen kann, welche Strecke zurückgelegt wurde. Aus Gründen der Echtzeitfähigkeit beim Auszählen der Encoder werden diese allerdings trotzdem auf der Antriebssteuerung ausgewertet. Die Encoder sind zum aktuellen Stand des Projekts eine Fall-back-Lösung. Im Nachfolgemodul wird noch evaluiert, ob die getätigten Schritte, welche aus dem Motorentreiber ausgelesen werden können, ausreichen können, um die gefahrene Strecke nachvollziehen zu können.

Signale, in welche Richtung das Fahrzeug bewegt werden soll, sowie Start- und Stoppsignale erhält der Mikrocontroller vom Navigationscomputer über eine UART-Schnittstelle. Der Greifer-Controller kann dem Antriebs-Controller ebenfalls via UART, Signale übermitteln, welche die Antriebe anhalten - das Fahrzeug drehen oder eben weiterfahren lassen.

Abbildung 7 zeigt schematisch, welche Funktionsgruppen das entsprechende PCB enthält.

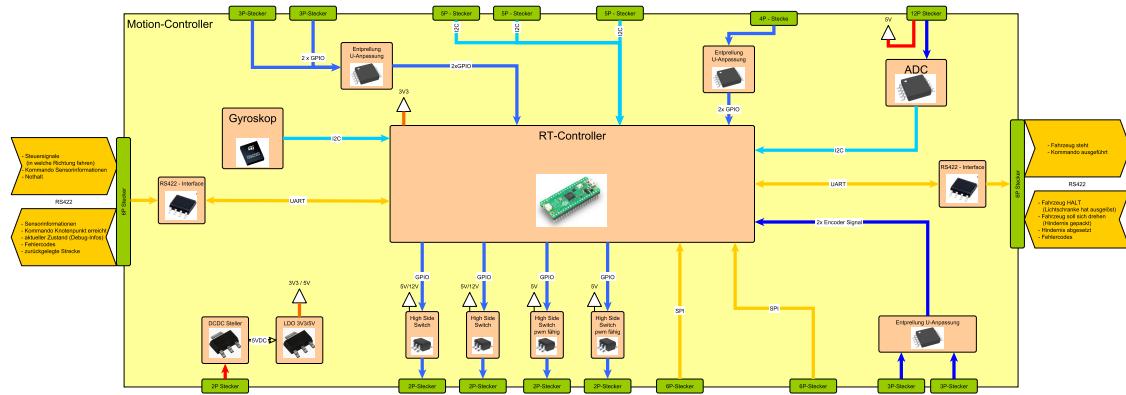


Abbildung 7: Blockschaltbild Motion Controller

Der Motion Controller besitzt sowohl digitale Eingänge als auch Ausgänge mit umschaltbaren Spannungsniveaus als Reserve, falls im Laufe des Folgemoduls PREN2 Sensorik angepasst oder zusätzliche Aktorik hinzugefügt werden muss.

Abbildung 8 zeigt das Motion-Controller-PCB in einer 3D-Ansicht. Es besitzt 3 I2C - Verbindungen, ein spezielles HC-SR04 Ultraschallsensor Interface sowie die erwähnten digitalen Ein-/Ausgänge. Darüber hinaus ist eine Schnittstelle für den Liniensensor vorgesehen und ausserdem ein Gyroskop zur Winkelerfassung verbaut. Es wird wie eine Brücke auf die beiden Evaluation-Boards von Trinamic aufgesteckt sein, wofür sich an den äusseren Rändern die breiten Steckerleisten befinden.

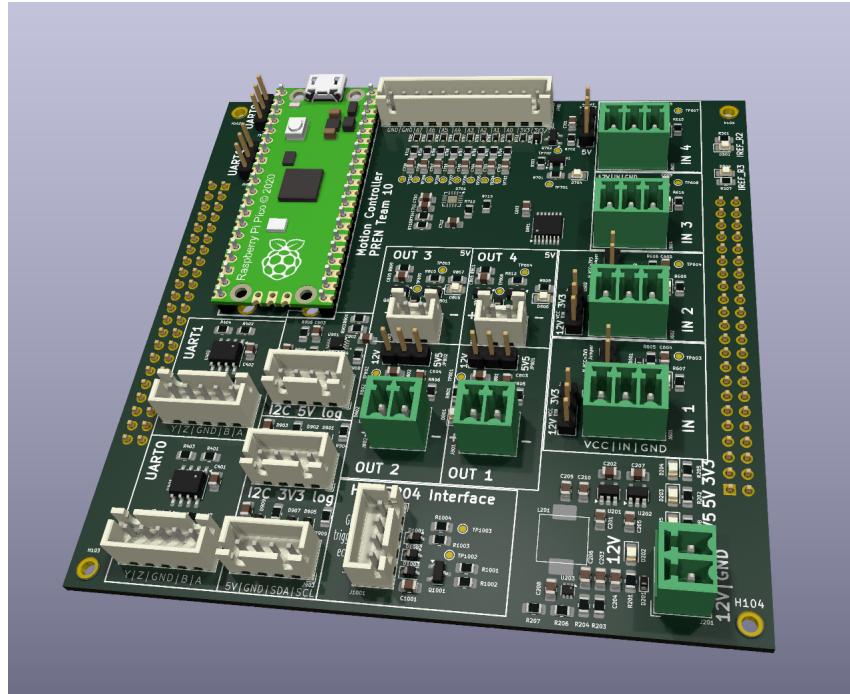


Abbildung 8: 3D-Ansicht Motion Controller

### 3.2.4. Liniensensor

Im folgenden Abschnitt ist die Entwicklung und Evaluierung eines Liniensensors dokumentiert. Ziel ist es, einen einfach auszuwertenden Sensor zu entwickeln, der das vorgegebene Klebeband (*Tesa Gewebeband 4651*) problemlos vom Wettkampf-Untergrund unterscheiden kann.

Ein eigener Liniensensor wird entwickelt, weil das Verlassen der Strecke als ein hohes Risiko empfunden wird. Daher ermöglicht das eigene Designen eines Liniensensors eine hohe Flexibilität. Dies mit der Begründung, dass für ein massgeschneidertes Produkt alle Komponenten selber erlesen werden können.

#### Anforderungen

Das Klebeband muss auf einem rötlich gefliesten Untergrund detektiert werden. Eine besondere Herausforderung stellen hierbei längs und quer verlaufende Fliesenfugen dar, welche eine ähnliche Farbe aufweisen. Dieser Untergrund ist in Abbildung 9 gezeigt.



Abbildung 9: Untergrund während des Wettkampfs

### Aufbau und Auswertung

Der Liniensor wird aus acht einzelnen Messzellen aufgebaut. Es werden acht Messzellen gewählt, damit eine möglichst breite Fläche durch den Sensor abgedeckt werden kann, aber dennoch nicht zu viele Pins für die Auswertung benötigt werden. Dabei sollten stets genau zwei Messzellen direkt über dem Klebeband ausgerichtet sein. Jede einzelne Messzelle hat eine emittierende Diode und einen dazugehörigen Fototransistor. Je mehr Licht auf den Fototransistor einwirkt, desto höher ist der Strom, welcher durch ihn fließt. Weil der Fototransistor als konstante Stromquelle interpretiert werden kann, wird der Spannungsabfall über ihn ausgewertet. Ein hoher Strom durch den Fototransistor bewirkt einen hohen Spannungsverlust an einem davor geschalteten Widerstand. Somit sollte die Spannung über dem Fototransistor gegen Ground gezogen werden. Ist der Strom jedoch klein, so fällt gemäss dem Ohmschen Gesetz wenig Spannung über dem Vorwiderstand ab was einen erhöhten Spannungsabfall über dem Fototransistor zur Folge hat. Diese Spannungen werden dann mittels Analog-Digital-Wandlern (ADCs) ausgewertet. In Abbildung 10 ist die Auswertung schematisch festgehalten.

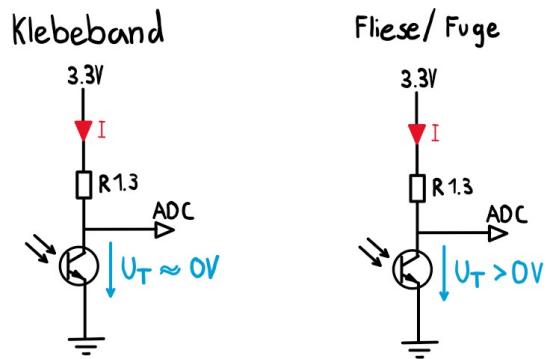


Abbildung 10: Konzept der Auswertung mittels ADC

### 3.2.5. Liniensor als PCB

Damit der Liniensor möglichst praxisnah getestet werden kann, wird dieser als PCB mit Kicad erstellt. Dabei werden die oben formulierten Anforderungen und Dimensionierungen eingehalten. In Abbildung 11 ist die Draufsicht und in Abbildung 12 die Untersicht dargestellt.

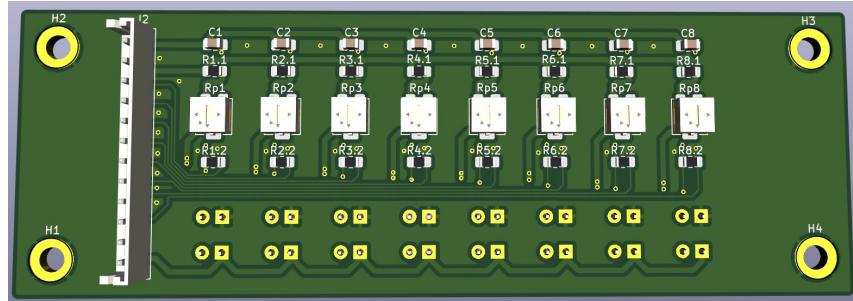


Abbildung 11: Liniensor in Kicad von oben

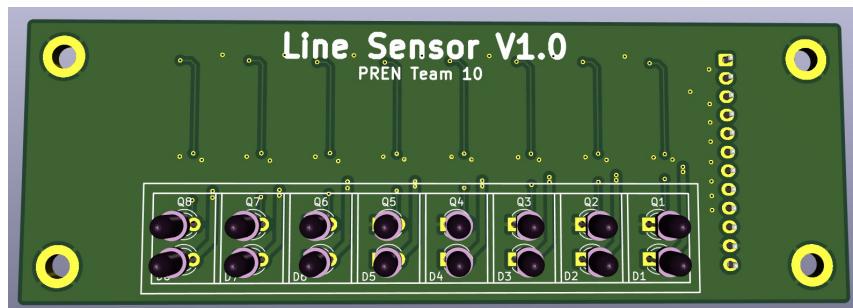


Abbildung 12: Liniensor in Kicad von unten

**Versuche** Im Anhang ist ein Versuch dokumentiert, welcher ein geeignetes Lichtspektrum für den Liniensor liefert. Außerdem wurden die Ströme auf den unterschiedlichen Untergründen (Klebeband, Fliese und Fuge) gemessen. Mittels eines Arduino wurden alle Eingänge ausgewertet und der Liniensor validiert. Alle Messungen sind im Anhang G dokumentiert.

**Entscheidung und Fazit** Durch die Messungen wurde das Konzept des Liniensors überprüft und validiert. Das folgende Bild bestätigt, dass das Klebeband von der Fliese deutlich unterscheiden werden kann. Die Abbildung 13 zeigt die Spannungsauswertung mittels ADCs. Diese Auswertung wurde mit einem Arduino gemacht. Die hohen Werte stellen die Fliese dar und die tiefen Werte das Klebeband.

|         |         |         |        |         |         |         |         |
|---------|---------|---------|--------|---------|---------|---------|---------|
| 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 13: Die tiefen Werte stellen das Klebeband dar und die hohen die Fliesen

Auf der Abbildung ist ersichtlich, dass sich A3 direkt überhalb dem Klebeband befindet. Die Pins A2 und A4 sind teilweise auch über dem Klebeband. Der Wert ist allerdings grösser, weil nur ein Teil des Klebebandes direkt darunter liegt.

Anfangs wurde der Unterschied von Klebeband zu Fugen als eine Schwierigkeit betrachtet. Der Liniensensor ist fähig, Fugen von Klebeband zu unterscheiden. Dies ist im Anhang G genauer dokumentiert.

Aufgrund der eindeutigen Messwerten, wird dieser Liniensensor verwendet.

### 3.2.6. Abstandssensoren

Wie die Konzeption der Abstandssensoren (siehe Anhang E) ergeben hat, wird die Detektion der Umwelt beim ersten Prototyp redundant ausgeführt, welche im Nachfolgemodul dann noch genauer ausgearbeitet werden.

Das in PREN1 ausgearbeitete Konzept sieht vor, anhand eines LIDAR über die Hindernisse hinweg den Abstand zu potenziellen Pylonen zu detektieren. Sind diese näher als 2 m - ist die betrachtete Strecke wahrscheinlich nicht befahrbar. Ein zweiter Distanzsensor, welcher über Ultraschall arbeiten wird, detektiert Hindernisse, welche sich näher als 0.5m befinden. Dadurch kann das Fahrzeug frühzeitig abgebremst werden. Wird die beim Greifer positionierte Lichtschranke ausgelöst, kann mit Sicherheit davon ausgegangen werden, dass das Fahrzeug nun beginnen kann und soll, das Hindernis zu greifen und umzupositionieren. Diese Funktion könnte allerdings auch vom Ultraschallsensor übernommen werden, dies wird im Folgemodul weiter untersucht. Ebenfalls befindet sich auf dem Fahrzeug eine Kamera, welche die vorliegenden Streckenpunkte analysieren kann. Mit eben dieser Kamera könnten ebenfalls Hindernisse detektiert werden, was allerdings auch erst im Folgemodul, wenn ein erster Prototyp aufgebaut wurde, weiter untersucht wird.

**Fazit** Das in Abbildung 14 gezeigte Konzept stellt den aktuellen Stand der Entwicklung dar und ist so für den ersten Aufbau eingeplant.

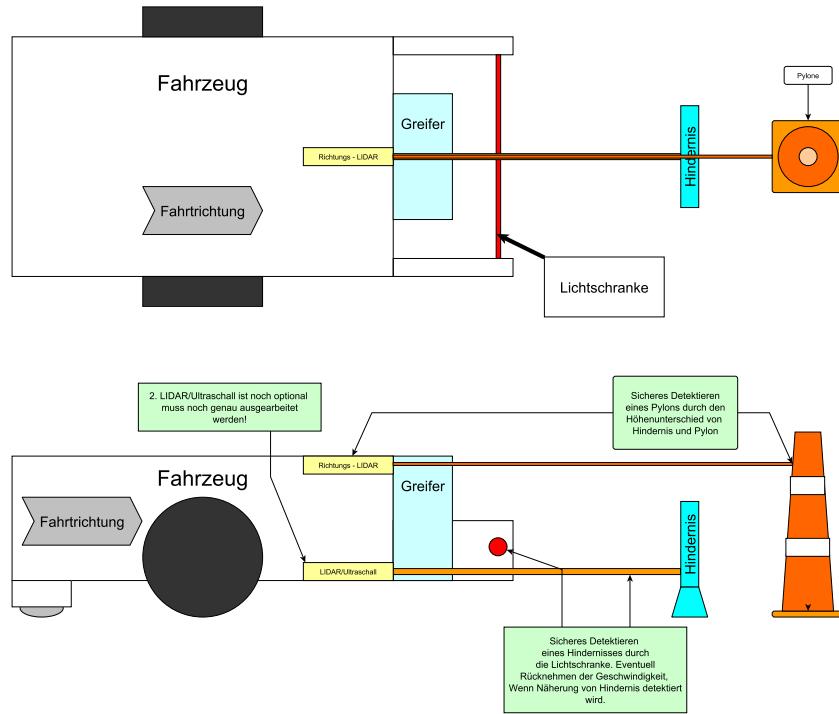


Abbildung 14: Konzept Hinderniserkennung

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

### 3.2.7. Sensorik Strecken Rückverfolgung

Der Algorithmus setzt voraus, dass das Roboter immer und zu jeder Zeit seine Orientierung als absoluten Winkel ab dem Startpunkt weiss und die zurückgelegte Strecke messen kann. Dieser Abschnitt behandelt, wie der Roboter eben diese Information erfasst. Abbildung 15 zeigt schematisch auf, wie diese Funktion umgesetzt wird. Die danach folgende Beschreibung führt dieses Schema nochmal aus.

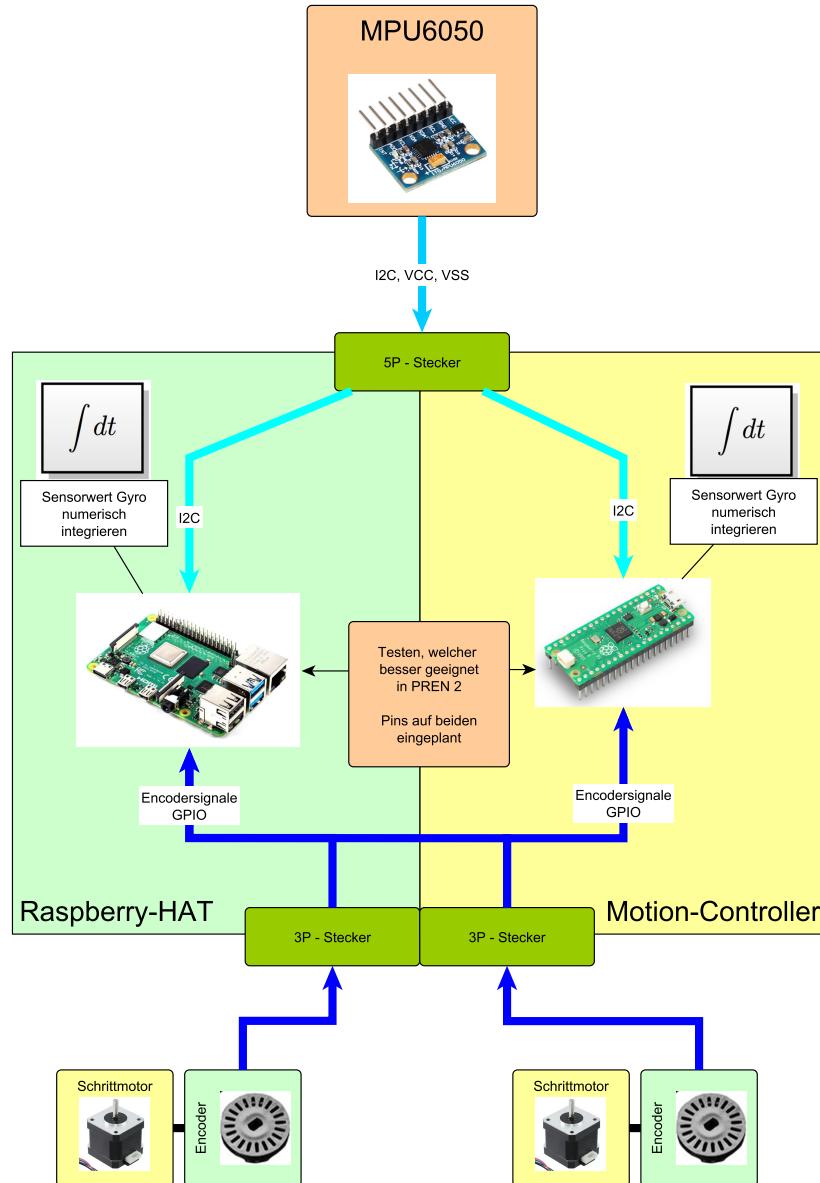


Abbildung 15: Blockschaltbild Sensorik Streckenrückverfolgung

**Winkelerfassung** Der Winkel des Pfadfinders wird, wie in Anhang F konzeptioniert, über ein Gyroskop ermittelt. Dieses erfassst die Änderungsrate des aktuellen Winkels zu jeder Zeit und integriert sie jede  $25\mu s$  numerisch auf. Dadurch kann eine sehr genaue Angabe darüber getroffen werden, in welcher Orientierung sich das Roboter aktuell befindet. Mangels Echtzeitfähigkeit ist es nicht sicher, ob der Raspberry Pi in der Lage ist, das Gyroskop ausreichend genau und häufig auszulesen. Deshalb wird ein entsprechender Steckplatz sowohl auf dem Antriebs-Controller, als auch auf dem Raspberry-HAT vorgesehen, um dies im Nachfolgemodul PREN2 noch zu testen.

**Zurückgelegte Strecke** Wie in der Konzeption beschrieben, wird die zurückgelegte Strecke des Pfadfinders mithilfe eines Encoders erfasst. Die verwendete Lochscheibe besitzt 36 Löcher, was einer Auflösung von 10 Grad pro Schritt entspricht. Die Dimensionen der

Löcher wurden basierend auf den 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 16).

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

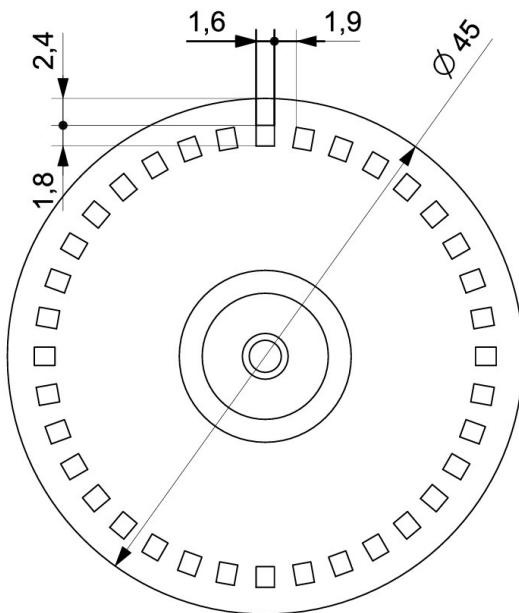


Abbildung 16: Masse der Lochscheibe

In einem zweiten Versuch soll im Folgemodul PREN2 getestet werden, ob es ausreichen könnte, die zurückgelegten Schritte des Schrittmotorentreibers für diese Referenz zu verwenden. Dieser Versuch kann allerdings erst ausreichend aussagekräftig durchgeführt werden, wenn ein erster Roboter gebaut ist.

### 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 jedoch, dass die Anwendung von klassischen Shortest Path-Algorithmen 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 im Graphen problematisch. Der Roboter fährt los, ohne die vollständige Umgebung zu kennen, und kann auf unvorhergesehene Situationen treffen, wie zum Beispiel:

- **Sackgassen und Umwege:** Der Roboter kann in Sackgassen gelangen, aus denen er wieder herausfinden muss. Dabei sind klassische Algorithmen wie Dijkstra und A\* nicht flexibel genug, da sie voraussetzen, dass eine vollständige Karte und alle möglichen Wege bekannt sind.
- **Blockierte Knoten hinter freien Knoten:** Ein blockierter Knoten (z. B. mit einem Pylon) kann sich direkt hinter einem freien Knoten befinden. Ohne Kenntnis dieser Blockade im Vorfeld kann der Algorithmus falsche Entscheidungen treffen.
- **Unsicherheit bei entfernten Strecken:** Wenn eine Strecke im Graph entfernt wurde, ist es schwierig, genau zu wissen, welche Alternativwege existieren. Algorithmen wie Dijkstra und A\* erfordern, dass der genaue Standort des Roboters und alle möglichen Verbindungen zu jedem Zeitpunkt bekannt sind.

Diese Probleme machten es für die Gruppe schwierig, klassische Shortest Path-Algorithmen auf unsere Problemstellung anzuwenden.

**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 intelligenter Heuristik, 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. Einbindung in Roboter

Benötigte Schnittstellen, Sensorik & Co. Ausserdem: Wo stehen noch Risiken? Thema Messtoleranzen & Co. ?

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

**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:

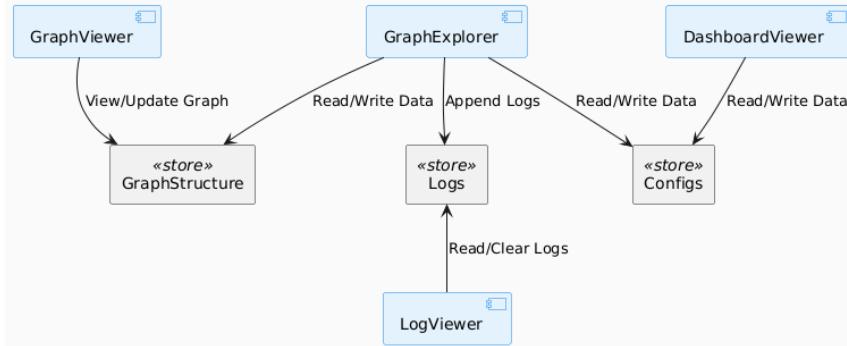


Abbildung 17: Aufbau des Simulators und dessen Komponenten

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 18:

- **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 Stati wie *probed* (gelb), *visited* (grün), *restricted* (rot) oder *finished* (blau) werden visuell hervorgehoben und erlauben eine präzise Nachverfolgung des Algorithmus.

## Graph

Randomized graph

Reset graph

Reset state

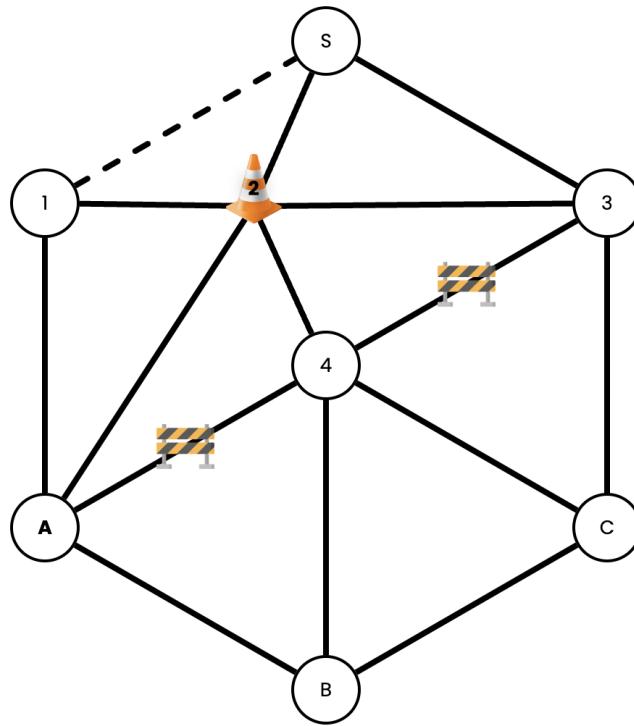


Abbildung 18: Section für Graph im Dashboard

**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 [18]:

- **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 aktuellen Zustand klar darzustellen.

**Funktionen und Modus-Optionen** Der Simulator bietet drei verschiedene Betriebsmodi. Diese werden durch Tabs voneinander graphisch getrennt 19:

- **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.

## Dashboard

### Execution mode:

|                 |                   |             |
|-----------------|-------------------|-------------|
| Interactive run | Parameterized run | Explore map |
|-----------------|-------------------|-------------|

### Vehicle-Parameters:

|                                    |   |
|------------------------------------|---|
| Time to traverse edge (cm/s):<br>1 | Time to traverse edge with barrier (cm/s):<br>2 |
|------------------------------------|---|

### Algorithm-Selection:

#### Algorithm to apply:

Abbildung 19: Bereich für Betriebsmodi und deren Konfiguration

**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 20:

- **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.

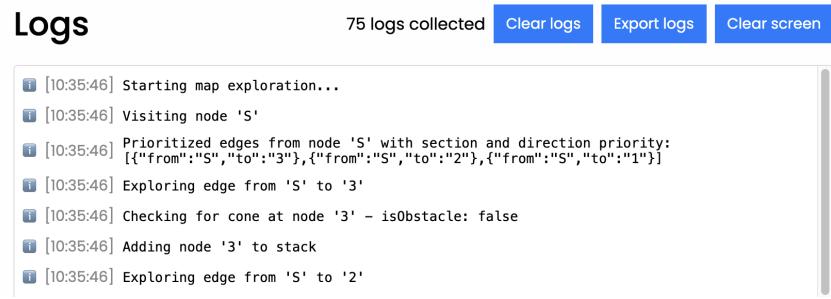


Abbildung 20: Bereich für Anzeige, Export und Löschung der Logs

**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 21 zeigt eben diese Struktur des Teams.

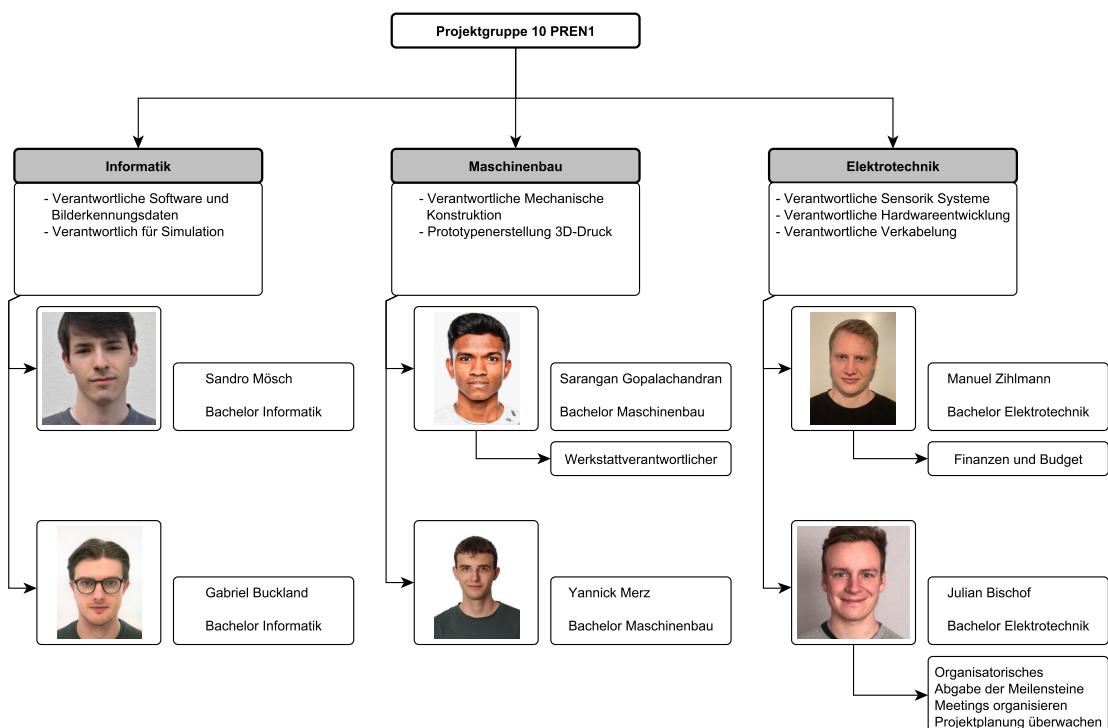


Abbildung 21: Organigramm Team 10

### 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 22 zeigt die verschiedenen Menüs sowie das eingerichtete Kanban-Board für das Projektteam.

Abbildung 22: 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.

#### **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 Wettkampfplatz 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.

## 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 unseres Projekts – Elektronik, Mechanik und Informatik – dargestellt.

### 5.2.1. Elektronik

- **Komponentenwahl:** asdf
- **Wiederverwendbarkeit:** asdf
- **Lieferantenwahl:** adsf

### 5.2.2. Mechanik

- **Materialauswahl:** asdf
- **Modulares Design:** asdf
- **Energieeffiziente Antriebssysteme:** asdf

### 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**

Team-Reflektion zum Kompetenzerwerb\*, Meinung, Ausblick

## 6. Schlussdiskussion

- Budgetplanung
- Gewichtsplanung
- morphologischer Kasten

### 6.1. Erfahrungen & Lessons learned

Text here

## **6.2. Erfüllungsgrad Anforderungsliste**

Text here

### **6.3. Fazit und Ausblick**

Text here

## **7. Verzeichnisse**

### **7.1. subsection**

Text here

## 8. Anhang

### A. Anforderungsliste Version 2

Die nachfolgende Tabelle zeigt die Anforderungsliste des Roboter-Fahrzeugs.

#### Legende

F = Festanforderung

M = Mindestanforderung

W = Wunschanforderung

#### A.1. Allgemeine Anforderungen

| Nr.  | F<br>M<br>W | Bezeichnung          | Daten<br>Werte<br>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 Materialien.   |
| 1.10 | W           | Lieferwege           | Wenn möglich Materialien und Kaufteile aus Europäischen Lagern beziehen.   |

**Roboter**

| Nr.  | F<br>M<br>W | Bezeichnung           | Daten<br>Werte<br>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 vom Fahrzeug erkannt werden.  |
| 2.6  | W           | Hindernis auf Strecke | Mögliche Hindernisse sollten vom Fahrzeug erkannt werden.  |
| 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           | Schutzkasse           | Mindestens IP-10 sollte gewährleistet sein.  |

| Nr.  | F<br>M<br>W | Bezeichnung               | Daten<br>Werte<br>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<br>M<br>W | Bezeichnung                   | Daten<br>Werte<br>Erläuterungen   |
|------|-------------|-------------------------------|---|
| 3.1  | F           | Wege-Netzwerk                 | Das Graph-Topologie und der Startknoten sind bekannt. (Abbildung ??)  |
| 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. (Abbildung ??) |
| 3.3  | F           | Wegpunkte                     | Insgesamt gibt es acht Wegpunkte. Die Wegpunkte sind aufgeklebte Vollkreise (weiss) mit einem Durchmesser von 7 bis 12 cm. (Abbildung ??)   |
| 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. (Abbildung ??)  |
| 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<br>M<br>W | Bezeichnung               | Daten<br>Werte<br>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, 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           | Hindernistypen            | Verschiedene Arten von Hindernissen (beweglich und statisch) sollen simuliert werden können.  |
| 4.10 | W           | Fahrzeugparameter         | Fahrzeugparameter (Geschwindigkeit, Wendekreis, Sensorreichweite, etc.) sollen editierbar sein.   |
| 4.11 | W           | Datenexport               | Die Daten, welche während der Simulation generiert werden, sollen exportierbar sein.  |
| 4.12 | 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<br>M<br>W | Bezeichnung                     | Daten<br>Werte<br>Erläuterungen   |
|-----|-------------|---------------------------------|---|
| 5.1 | W           | Materialbeschaffung             | Materialien und Komponenten sollen vorzugsweise von folgenden Lieferanten bestellt werden:<br>- Conrad Electronic<br>- Distrelec<br>- Mädler<br>- Farnell   |
| 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:<br>- maximal 25 h Maschinenlaufzeit der 3D-Drucker<br>- maximal 1 h Maschinenlaufzeit des Lasergeräts<br>- maximal 10 Arbeitsstunden des Werkstattpersonals Elektrotechnik<br>- maximal 10 Arbeitsstunden des Werkstattpersonals Maschinentechnik |

## B. Konzepterstellung Greifereinheit

In diesem Abschnitt wird die Entwicklung des Greifarmkonzepts 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  $0.235 \text{ Nm}$ .

**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  $0.2 \text{ Nm}$ .

**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

### **Konzept 2 - Greifer und Höhenverstellung - ein Motor**

Bild

In diesem Konzept wird ein einzelner Motor sowohl für die Greiffunktion als auch für die Höhenverstellung eingesetzt. Die Höhenverstellung erfolgt mit einem Zahnrad, welches von einem Motor angetrieben wird und einer Zahnstange, die sich vertikal nach oben oder unten bewegt.

Der Greifer besteht aus zwei Backen: die vordere Backe ist fest montiert, während die hintere Backe beweglich auf einer Gleitführung sitzt. Die Backen sind mit einem Gummiband vorgespannt, sodass sie im Ruhezustand geschlossen bleiben. An den Seiten der hinteren Backe befinden sich Mitnehmer, die an einer linearen Nocke auf beiden Seiten geführt wird. Diese Nocken bewegen die hintere Backe an die gewünschte Position. Dies ermöglicht die Öffnung und Schließung des Greifers

Bild lineare Nocke (evtl. Bewegungsablauf erklären und noch Klappe)

Hier ist ein Bewegungsablauf... Klappe für verschiedenen Weg...

### **Konzept 1 - Greifer und Höhenverstellung - zwei Motoren**

Bild

In diesem Konzept werden jeweils separate Motoren für die Greiffunktion und für die Höhenverstellung eingesetzt. Der Greifer wird als Parallelgreifer umgesetzt. Ein Motor, der mit einem Zahnrad gekoppelt ist, treibt zwei Zahnstangen an, wodurch sich die beiden Greiferbacken horizontal bewegen.

Die Höhenverstellung erfolgt über die Ausgangswelle eines weiteren Motors, die fest mit einem Zahnrad verbunden ist. Dieses Zahnrad treibt eine Zahnstange an, die eine vertikale Bewegung nach oben oder unten ermöglicht.

### **Konzept 3 - Gabelstapler - ein Motor**

Bild

In diesem Konzept wird das Prinzip eines Gabelstaplers nachgebildet. Die Gabeln werden in die dafür vorgesehene Öffnung des Hindernisses eingeführt. Anschließend wird das Hindernis mithilfe eines Motors vertikal angehoben, wobei die Hebebewegung durch ein Zahnstangen-Zahnrad-System realisiert wird.

## C. Konzepterstellung Energieversorgung

In diesem Abschnitt wird sich mit der Energieversorgung des Pfadfinders auseinandergesetzt. 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 4 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 4: 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 23 zeigt ein Blockschaltbild dieser Variante. Im folgenden Unterabschnitt werden die einzelnen Unterbaugruppen kurz kommentiert.

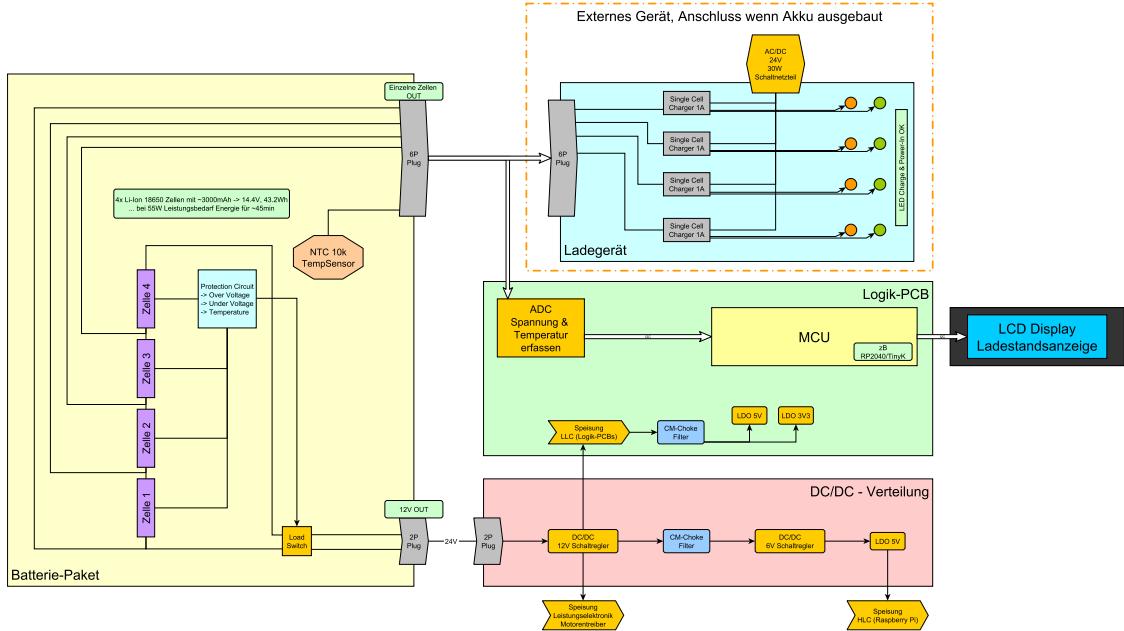


Abbildung 23: Blockschaltbild des Akku-Konzepts 12V Variante

**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 5 zeigt. Hier könnte allerdings noch am Gewicht des Gehäuses gespart werden.

**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 6 und Tabelle 7 zu entnehmen.

| 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%                 |   |     | $\approx$<br>320g |

Tabelle 5: Gewichtsschätzung eines Akku-Pakets 12V Variante

| 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<br>Temperaturfühler                | Vishay Dale         | CRCW080510K0JNTC | Digikey             | 1 | 0.12  | 0.12       |
| MOSFET<br>Leistungsschalter<br>P-Kanal | Diodes Incorporated | DMP3036SFV-13    | DigiKey             | 2 | 0.55  | 1.1        |
| MOSFET Gate<br>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,<br>C's, L's, ...)  |                     |                  |                     | 1 | 3     | 3          |
| PCB 4 Lagen<br>@100x100mm              |                     |                  | JLCPCB              | 1 | 8     | 8          |
| Gehäuse                                |                     |                  | 3D Druck,<br>leicht |   |       | 0          |
| Versandpauschale                       |                     |                  |                     | 1 | 10    | 10         |
| SUMME Battery Pack<br>+ 10%            |                     |                  |                     |   |       | 40.61      |

Tabelle 6: 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<br>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,<br>C's, L's, ...) |                       |                 |             | 1 | 3     | 3          |
| PCB 2 Lagen<br>@100x100mm             |                       |                 | JLCPCB      | 1 | 8     | 8          |
| Versandpauschale                      |                       |                 |             | 1 | 10    | 10         |
| SUMME Ladegerät<br>+10%               |                       |                 |             |   |       | 61.655     |

Tabelle 7: 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*, abgebildet in Abbildung ???. 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 8 berechneten Preis. Die Kosten für das Ladegerät kann Tabelle 9 entnommen werden.

### 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.

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

Tabelle 8: 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 9: Kostenschätzung des Ladegeräts 12V Variante mit Kaufteilen

**Kostenschätzung** Die Akkuzelle beläuft sich auf den in Tabelle 10 berechneten Preis. Die Kosten für das Ladegerät können Tabelle 9 entnommen werden.

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

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

## Fazit und Entscheidung der Konzeptphase

Tabelle 11 zeigt einen Vergleich der verschiedenen Konzepte, um sie direkt miteinander gegenüberstellen zu können.

In einer gemeinsamen Sitzung ist der Entscheid auf das Konzept 3 gefallen. Hauptargumente sind der etwas reduzierte Entwicklungsaufwand, der günstige Preis für eine angemessene

| 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 11: Vergleichstabelle der verschiedenen Konzepte

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 Detektierter Unterspannung einer Batteriezelle einen MOSFET abzuschalten, womit die Batterie vom Netz getrennt wird. 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.

**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. Der MOSFET *Q201* hat die Aufgabe, die Batterie vom Netz zu trennen, sobald ein Netzteil angeschlossen ist.

**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. Ziel ist es, den Spannungsanstieg im System zu begrenzen, da dadurch direkt der Strom folgt, welcher in die Kapazitäten fliesst. Das ist ersichtlich am Zusammenhang

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

Der Spannungsanstieg beim Einschalten des MOSFETS *Q204* bezweckt eine Spannungsänderung über dem Kondensator *C210*. Durch den Zusammenhang

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

ergibt sich, dass sich über dem Widerstand *R204* 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 'selbst abstellt'. Aus den beiden Formeln folgt:

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

und daraus widerrum

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

Es wird nachfolgend mit dem gleichen MOSFET gerechnet, wie er bereits bei der Quellenumschaltung eingesetzt wird. Der maximale Strom durch den Widerstand *R204* wird mit  $10mA$  angenommen, wodurch ein Widerstand von  $1.2k\omega$  resultiert. Der Inrush-Strom soll auf  $7A$  begrenzt sein, bei einer Threshold-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. Die Impulsdauer wird stark vereinfacht, mit der Annahme, 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 bringen kann. Zur Dimensionierung der Kapazitäten, der Induktivität und der Diode wurde sich auf die Dimensionierungs-Empfehlung des Datenblatts berufen.

**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 können. 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 Störungen, verursacht durch z.B. den Buck-Konverter, über eben diese Leitung an der Filterung vorbei auf den Navigationscomputer gelangen können. 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.

**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 Motorenkonzepte diskutiert, als auch potenziell Einsatzfähige Motoren herausgesucht. Im Abschluss wird noch Konzeptionell zusammengefasst, wie diese Antriebe anzusteuern sind.

### Anforderungen

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

**Gewicht** Gewicht ist, wie bei allen anderen Baugruppen ebenfalls, ein sehr kritischer Punkt bei der Entwicklung des Pfadfinders. Als *Gewichtsbudget* wurde für die Antriebseinheit festgelegt, dass das Gewicht nicht grösser als total  $500g$  sein sollte.

**Kosten** Das Budget dieses Projekts ist sehr beschränkt. Antriebe können hierbei einen wesentlichen Kostenpunkt darstellen, weshalb dem Antrieb inklusive der Ansteuerung und Räder ein Budget von  $100CHF$  zugewiesen wurde.

**Nennspannung** Die benötigte Nennspannung des Boardnetzes richtet sich mehrheitlich nach der benötigten Spannung der Motoren. Wie im Kapitel ?? bereits erwähnt wurde, würde eine Spannung von  $24V$  ein nicht umsetzbares Gewicht darstellen. Daher beträgt die Maximalspannung für Motoren hier  $12V$  Nennspannung.

**Technologie** Die Ansteuerung der Motoren ist mit geeigneten Treibern in den meisten Fällen relativ einfach umzusetzen. Aufgrund bestehender Erfahrungswerte sollen vorzüglich Schrittmotoren eingesetzt werden.

**Schnittstelle** Die Schnittstelle zur Ansteuerung der Motoren soll einem einfach umzusetzenden Standard entsprechen. Das betrifft die Ansteuerung über bekannte Bus-Protokolle wie  $I^2C$ , *SPI*, *UART* oder aber auch PWM- oder Step/Dir Interfaces.

Die Anforderungen an den Akku schliessen aufgrund der  $12V$ -Boardnetzspannung Industrietaugliche  $24V$ -Motoren bereits aus. Diese sind zwar sehr robust, allerdings in den meisten Fällen auch sehr schwer.

### Leistungsanforderungen an Motoren

---

Text Yanik zu Dimensionierung

---

### Konzeption

Aus den Leistungsanforderungen an die Motoren heraus lässt sich auf dem Markt schauen, welche Motoren potenziell zur Verfügung stehen. Diese lassen sich wiederum in den Kriterien Kosten, Gewicht, Leistung und Entwicklungsaufwand gegeneinander vergleichen.

### Konzept 1 - Bipolarer Schrittmotor

Die Wahl für einen bipolaren Schrittmotor fällt, da er bei gleichem Gewicht eine grössere Ausnutzung der Spulen bringt - woraus eine höhere Leistung bei gleichem Gewicht im Vergleich zu einem unipolaren Schrittmotoren resultiert.

Ein Vergleich verschiedener Schrittmotoren hat gezeigt, dass Schrittmotoren, welche sich in einem Nennstrombereich von 1.2A bis 2A in einem Bereich befinden, welche den Leistungsanforderungen bereits ohne Zwischengetriebe genügen können.

Typische, günstige Vertreter aus dieser Reihe sind die in Tabelle 12 aufgeführten.

| Hersteller           | Herst. Nr      | Distributor | $I_{nenn}$<br>[A] | $U_{nenn}$<br>[V] | Preis<br>[CHF] | Gewicht<br>[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 12: Verschiedene Schrittmotoren

Dieser Vergleich zeigt, dass sich eigentlich nur der Motor *FIT0278* von *DFRobot* in einem akzeptablen Gewichts-Leistungs-Verhältnis befindet. Preisgünstig sind alle aufgeführten Motoren.

Die Motoren sollen über eingekaufte Treiber angesteuert werden. Zwar ist eine Eigenentwicklung eines Motorentreibers mit nicht allzu viel Aufwand umzusetzen, jedoch sind fertige Treiber-Endstufen sehr günstig zu haben und einfach anzusteuern. Ein Team-Mitglied kann auf Erfahrung aus seinem beruflichen Umfeld zurückgreifen bei der Ansteuerung des Schrittmotorentreibers *TMC5240* von *ADI-Trinamic*. Diese Treiber stehen darüber hinaus für dieses Projekt in zweifacher Ausführung zur Verfügung. Der Vergleich dieses Treibers mit einem einfach anzusteuernden Treibers zeigt die Tabelle 13.

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

Tabelle 13: Verschiedene Schrittmotorentreibер

### Konzept 2 - Radnabenmotor

Es gibt fertig zu kaufende Radnaben auf dem Markt, welche zum Teil bereits sowohl Endstufen als auch eine Ansteuerung beinhalten.

Als Beispiel zu nennen wäre da der folgende:

| Hersteller | Herst. Nr | Distributor | $I_{nenn}$<br>[A] | $U_{nenn}$<br>[V] | Preis<br>[CHF] | Gewicht<br>[kg] |
|------------|-----------|-------------|-------------------|-------------------|----------------|-----------------|
| DFRobot    | FIT1001   | DigiKey.ch  | 0.5               | 14.4              | 25.71          | 0. 216          |

Tabelle 14: Radnabenmotor DFRobot

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

### Konzept 3 - BLDC-Motor

BLDC Motoren bieten den grossen Vorteil, dass sie bei kleinem Gewicht eine hohe Leistung erreichen können. Ihr Drehmoment hingegen ist eher gering - woraus die Notwendigkeit an ein Zwischengetriebe besteht. Dieses Zwischengetriebe treibt das Gewicht des Antriebs weiter in die Höhe. BLDC Motoren kommen darüber hinaus häufig mit einem höheren Preis daher.

Die nachfolgende Tabelle zeigt eine Aufstellung verschiedener Motoren, welche in Frage kommen könnten.

| Hersteller      | Herst. Nr     | Distributor | $I_{nenn}$<br>[A] | $U_{nenn}$<br>[V] | Preis<br>[CHF] | Gewicht<br>[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 15: Radnabenmotor DFRobot

Der Motor von DFRobot wird zwar mit seiner Geschwindigkeit von maximal  $159 \frac{1}{min}$  die gewünschte Drehzahl zwar nicht erreichen, das geringe Gewicht bei entsprechender Leistung sowie sein bereits integrierten Treiber macht diesen Antrieb allerdings trotzdem Wert, im Detail beurteilt zu werden.

Generell lässt sich sagen, die Suche nach Bauteilen hat ergeben, dass BLDC-Motoren eine höhere Nennspannung benötigen, als mit dem Dimensionierten Akku-Pack zur Verfügung gestellt werden kann. Natürlich arbeiten Motorentreiber als Stromquellen, allerdings sind Motoren im entsprechend benötigten Leistungsbereich schnell zu teuer und schwer.

Generell lassen sich BLDC-Motoren, ähnlich wie die Schrittmotoren, über einen selbst entwickelten Treiber ansteuern, allerdings sind fertige Motorensteuerung ebenfalls unschlagbar günstig und über ein PWM-Interface sehr einfach anzusteuern. Als Vertreter dieser Produktgruppe ist der folgende zu nennen.

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

Tabelle 16: BLDC-Treiber

### Fazit und Entscheid aus der Konzeptionsphase

In der Gruppe ist der Entscheid darauf gefallen, den Schrittmotor von DFRobot, gezeigt in Abbildung 25, in Kombination mit den *Trinamic*-Schrittmotorentreibern, gezeigt in Abbildung 24, 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 zwingend die gewünschten Geschwindigkeits- und Beschleunigungswerte erreicht werden, dafür ist seine Ansteuerung sehr einfach, sie haben ihre Treiber bereits integriert und dazu sind sie auch leicht.

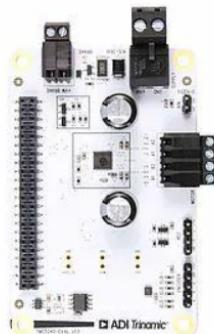


Abbildung 24: TMC 5240 Evaluation-Board



Abbildung 25: DFROBOT FIT0278 Schrittmotor

Geregelt werden soll dieser Motor von einem echtzeitfähigen Mikroprozessor, welcher ebenfalls über die Sensordaten des Liniensensors verfügt. Der Antrieb soll auf das Feedback eben dieses Sensors geregelt werden. Dies ist auf Abbildung 26 nochmals verdeutlicht. Zusätzlich wird noch mindestens ein Encoder vorgesehen, welcher mit einer einfachen Lochscheibe und einer Gabellichtschranke umgesetzt wird. Dieser hat allerdings keine Anwendung auf die Fahrzeugregelung. Mit ihm soll lediglich die zurückgelegte Strecke erfasst werden.

Die gewählten Treiber liessen sich direkt über den High-Level-Controller ansteuern. Die Antriebsregelung wurde aber bewusst auf einen Mikroprozessor verschoben, um dem Ansatz der Gewaltentrennung gerecht zu werden. So gibt es einen Echtzeitfähigen Prozessor, welcher sich ausschliesslich mit der Positionsregelung befasst, während der High-Level-Controller lediglich Entscheidungen trifft, in welche Richtung gefahren werden muss.

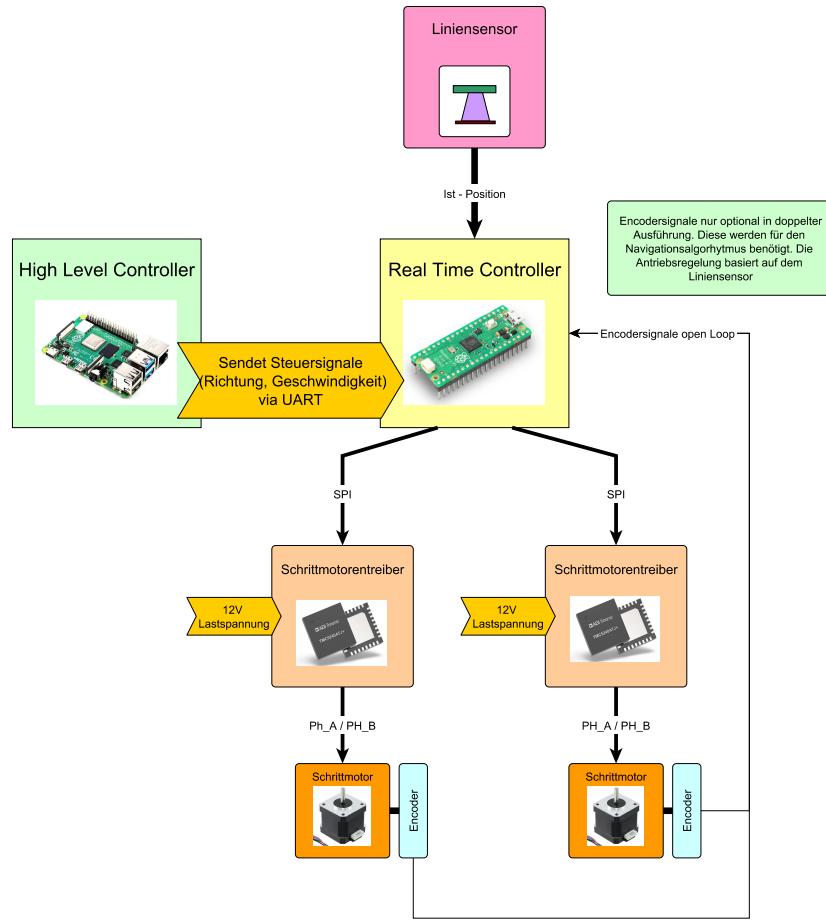


Abbildung 26: Konzept für die Ansteuerung der Schrittmotoren

## 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  $15^\circ$ , was auf eine Distanz von  $2m$  bereits einen Messkegel von ca.  $1m$  Radius bedeutet. Abbildung 27 zeigt dies nochmals.

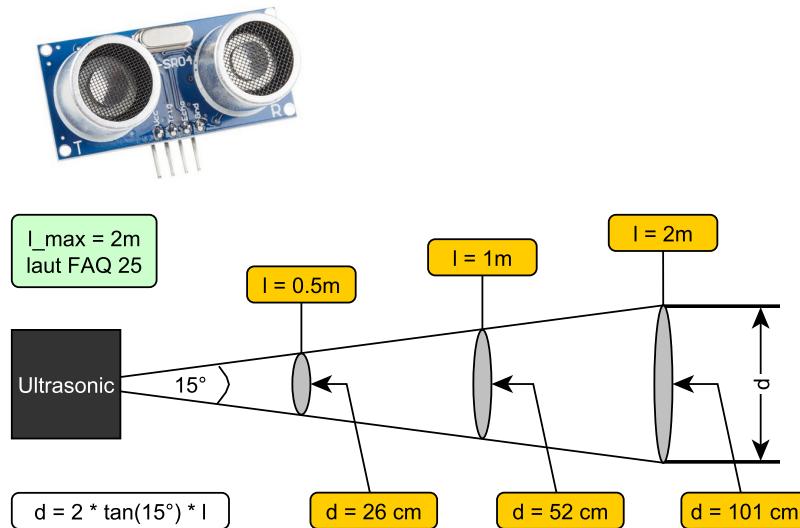


Abbildung 27: Messgenauigkeit Ultraschallsensor

## 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  $\approx 2$  breit, womit tatsächlich über ein Hindernis hinweg geschaut werden kann. Abbildung 28 zeigt dies nochmals.

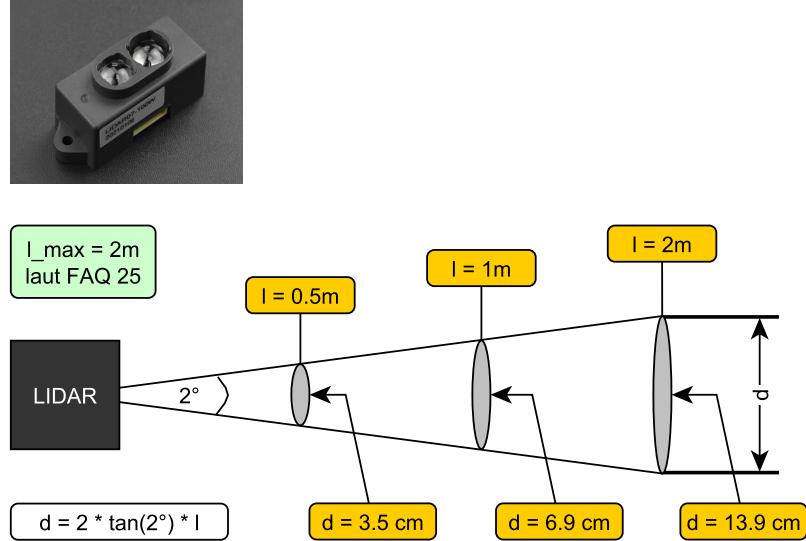


Abbildung 28: Messgenauigkeit LIDAR

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 29 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 29: Lichtschranke 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 ?? gezeigt.



Abbildung 30: 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 31 zeigt das erste angestrebte Konzept nochmals skizziert.

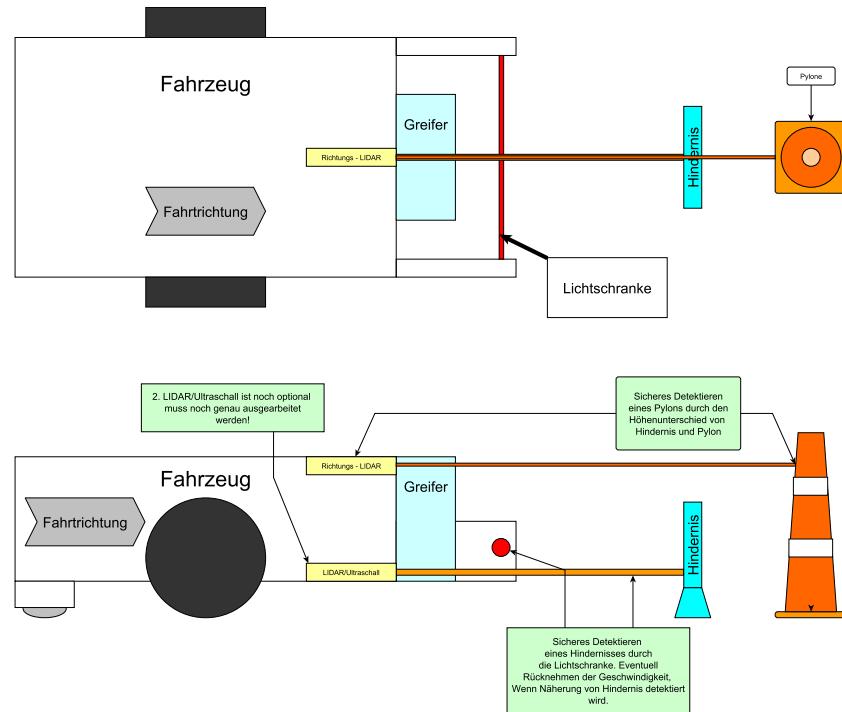


Abbildung 31: Konzept Hinderniserkennung

## 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 grossen 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 *I<sup>2</sup>C* Protokoll. Abbildung 32 zeigt eben dieses Gyroskop als Evaluation Board - welches auf einem Steckbrett zu Testzwecken eingesetzt wird.

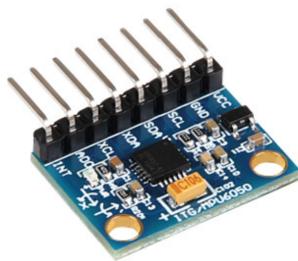


Abbildung 32: MPU6050 Gyroskop EVAL-Board

**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.

===== Text  
Yannik Dimensionierung Lochscheibe =====

Entsprechende Signale auf dem Mikroprozessor erzeugen Interrupts, mit welchen die Anzahl der Pulse ausgezählt werden. 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 befinden.

**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 17: 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  $360^\circ$  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 33 zeigt die konzeptionellen Überlegungen einer Messzelle des Liniensensors. Gemäss dem Datenblatt hat der UV-Emitter (Sender) einen Abstrahlwinkel von  $15^\circ$  und der Fototransistor (Empfänger) einen Einfallswinkel von  $60^\circ$ . 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.

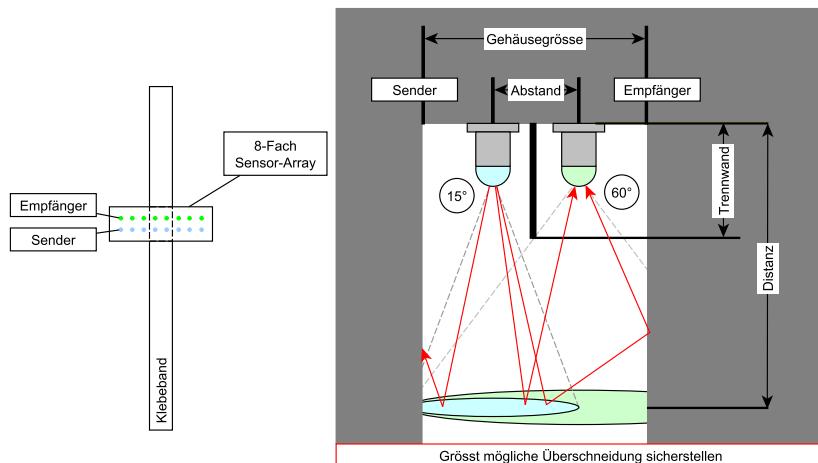


Abbildung 33: Konzept des Liniensensors

### Dimensionierung Gehäuse

Damit der Empfänger möglichst wenig auf Umwelteinflüsse reagiert, wird ein Gehäuse dimensioniert, welches den Sensor abschirmt. Die nachfolgende Abbildung 34 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 35 dargestellt.

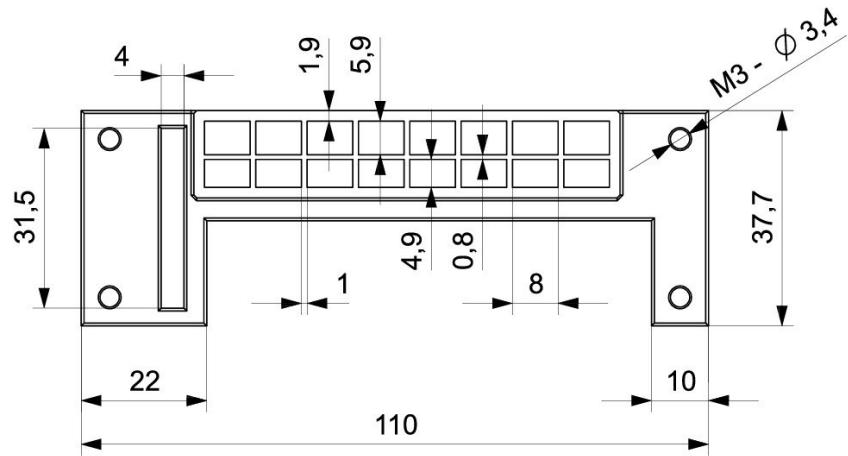


Abbildung 34: Vermassung des Gehäuses in Siemens NX

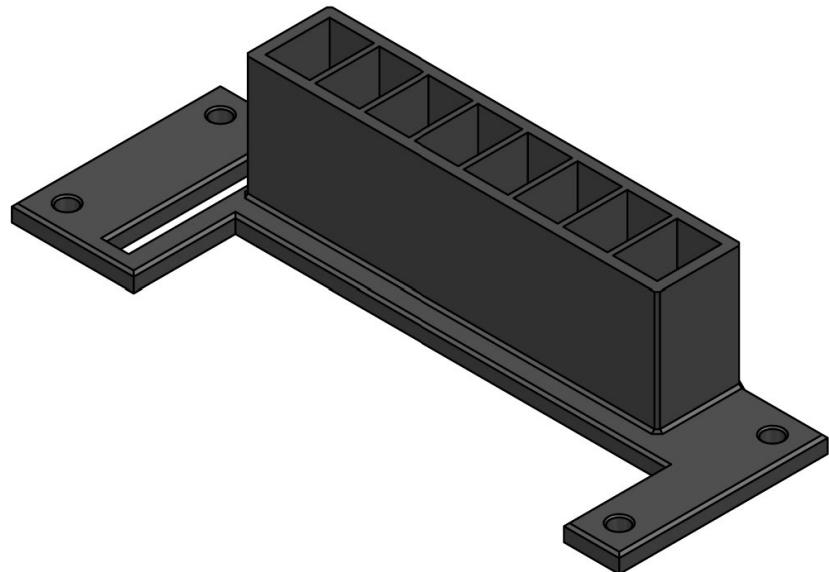


Abbildung 35: Isometrische Ansicht des Gehäuses in Siemens NX

### Schema einer Messzelle

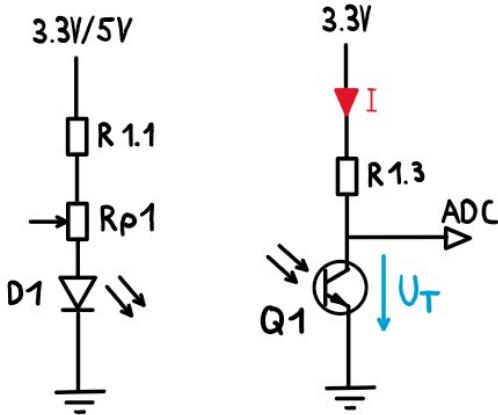


Abbildung 36: Schema einer Messzelle

Abbildung 36 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, dass 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  $200\Omega$  und ein  $R_{1.1}$  von  $33\Omega$ . 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  $200\Omega$  und ein  $R_{1.1}$  von  $82\Omega$ , 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 18 festgehalten. Dabei resultierte der Abstand von ungefähr 2.7mm von Messfläche zu Liniensor. Das Potentiometer wird mittig eingestellt, sprich auf  $100\Omega$ .

| 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 18: 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 19 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 19: 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 20 dargestellt.

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

Tabelle 20: 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 37 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 37: Vergleich der Kennzahlen von Fliese (oben) und Klebeband (unten) des Arduinos

Die Abbildung 38 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 38: Vergleich der Kennzahl Klebeband (vor allem A3) und der Fliese

In Abbildung 39 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 39: 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 21: 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 22: 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 23: 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 24: 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 25: 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 26: 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 27: 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 28: Erfasste Massnahmen für allgemein betreffende Risiken

Abbildung 40 zeigt schematisch, wie die getroffenen Massnahmen entsprechende Risiken für den Projekterfolg reduzieren.

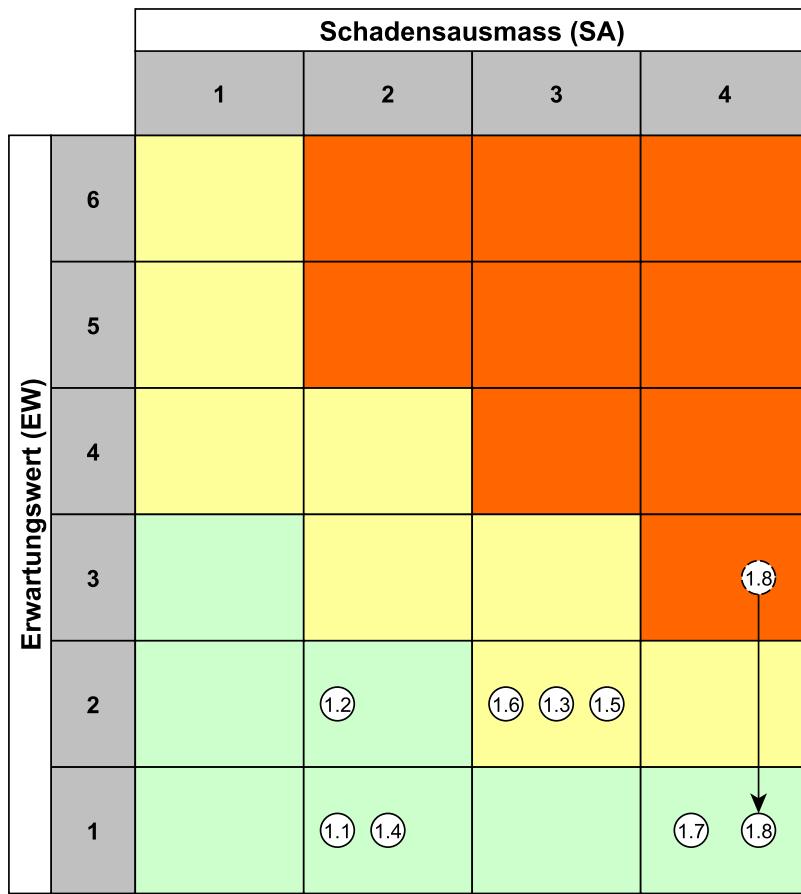


Abbildung 40: Grafische Darstellung der Risikoanalyse Allgemein

## 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 29: Erfasste Massnahmen für die Mechanik betreffende Risiken

Abbildung 41 zeigt schematisch, wie die getroffenen Massnahmen entsprechende Risiken für den Projekterfolg reduzieren.

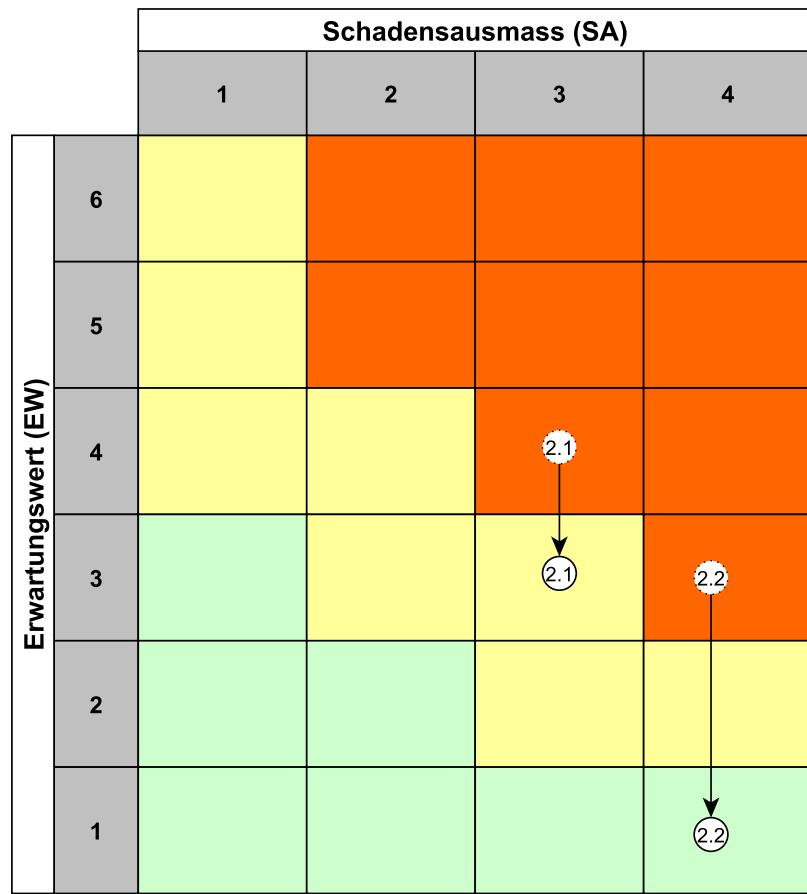


Abbildung 41: Grafische Darstellung der Risikoanalyse Mechanik

## **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 30: Erfasste Massnahmen für Risikoanalyse

Abbildung 42 zeigt schematisch auf, wie die getroffenen Massnahmen entsprechende Risiken auf den Projekterfolg reduzieren.

## **Informatik**

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

Tabelle 31: Erfasste Massnahmen für Risikoanalyse

Abbildung 43 zeigt schematisch auf, wie die getroffenen Massnahmen entsprechende Risiken auf den Projekterfolg reduzieren.

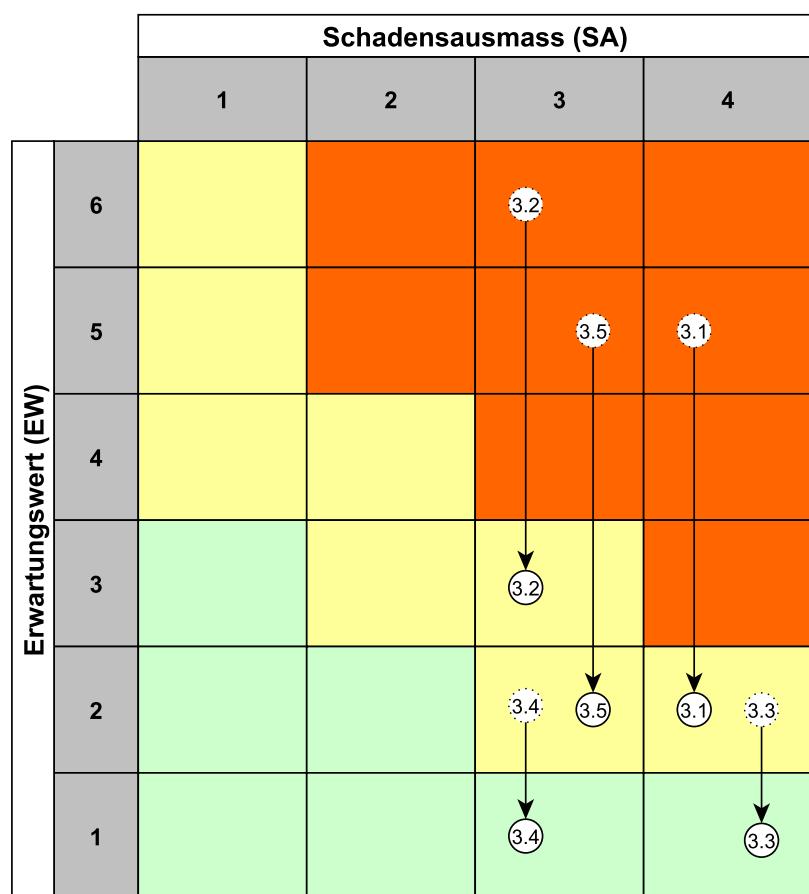


Abbildung 42: Grafische Darstellung Risikoanalyse Elektrotechnik

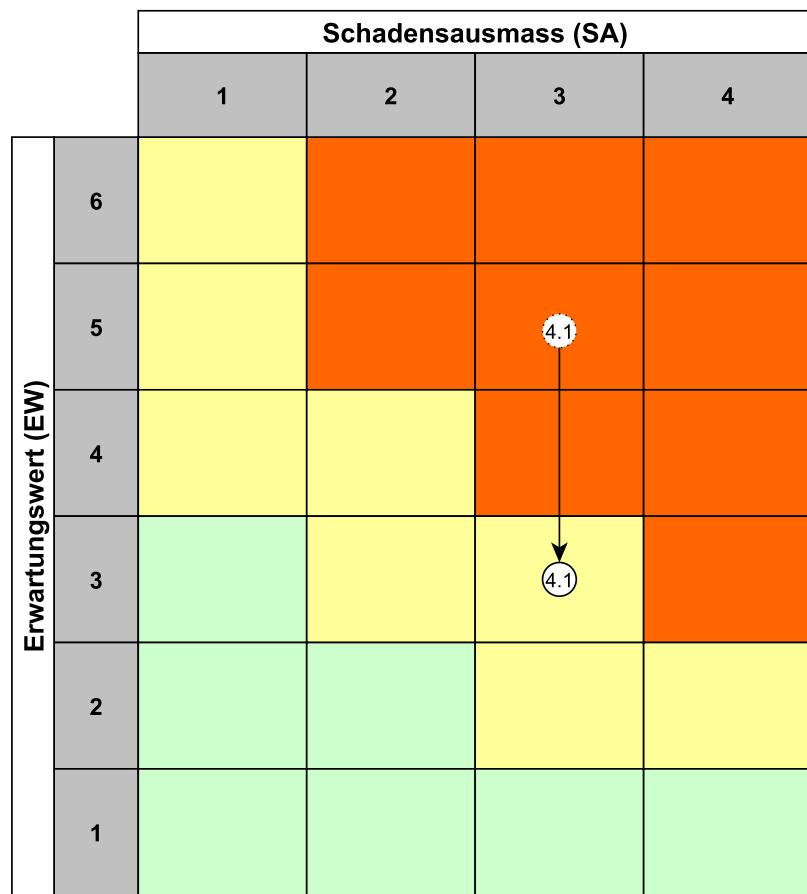


Abbildung 43: Grafische Darstellung Risikoanalyse Informatik

## I. Blockschaltbild Elektrokonzept

