

# GüselStar XXI

## Gesamtkonzept

---

Horw, 3. März 2016



*Autoren:*

Patrizio Brantschen  
Stefan Häfliger  
Tobias Kreienbühl  
Joël Meloni  
Silvan Ritz  
Lars Walther  
Adrian Würsch

*Supervisor:*  
Jürg Habegger

## **Inhaltsverzeichnis**

## Abstract

Die vorliegende Arbeit befasst sich mit der Konzeptfindung für das Modell eines autonomen Müllfahrzeugs. Im Rahmen des Moduls „Produktentwicklung 1“ an der Hochschule Luzern wird in einer interdisziplinären Zusammenarbeit zwischen Studierenden der Fachrichtungen Informatik, Elektrotechnik und Maschinenbau dieses Konzept erarbeitet. Das Fahrzeug wird im Modul „Produktentwicklung 2“ hergestellt und muss eine vorgegebene Aufgabe ausführen können. Es wird aufgezeigt wie mehrere mögliche Konzepte entwickelt werden und schlussendlich ein Lösungskonzept zur Herstellung ausgewählt wird. Der Fokus liegt auf Projektmanagement, Budgetplanung, Technologierecherchen, Konzepterarbeitung, interdisziplinärer Zusammenarbeit und Entscheidungsfindung. Anhand von Tests und Auswertungen wird das Chassis mit vier Rädern, von einem Servomotor angetriebenen Achsschenkellenkung und einem DC-Getriebemotor für den Antrieb bestimmt. Für die Erzeugung der Bilder wird eine drehbare Raspberry Pi Cam verwendet. Die Bilder für die Fahrbahnerkennung werden mit OpenCV über ein Raspberry Pi 2 ausgewertet. Eine weitere Aufgabe des Raspberry Pi ist die Informationsverteilung. Für die Kommunikation mit den Hardwarekomponenten wird ein Mikrocontrollerboard von Freescale eingesetzt. Die genaue Detektierung des zu leerenden Containers erfolgt mittels Infrarotsensoren. Um den Container zu greifen wird ein Schwenkarm benutzt, welcher mit einem Rundgreifer ausgestattet ist. Das Schüttgut wird über einen Behälter mit abgewinkelten Bodenflächen für das Entladen vorbereitet. Durch das Öffnen der Behälterklappe wird das Schüttgut in den Endbehälter entleert.

## 1 Einleitung

Busfahrer/in und Bus, Lokführer/in und Zug, Chauffeur/in und Auto, all diese Begriffe waren bis vor kurzer Zeit unmittelbar miteinander verbunden. Ohne den jeweils Anderen konnte man sich nicht fortbewegen. Doch genau dies ändert sich im Moment. Autonome Fahrzeuge sind längst nicht mehr nur wilde Fantasien, sondern werden von vielen Unternehmen entwickelt. Wenn man auf den Straßen von Kalifornien unterwegs ist, kann es bereits geschehen, dass ein solches autonom fahrendes Auto neben dem eigenen Wagen an der Ampel steht. Diese Entwicklung ist keineswegs neu, sondern hielt schlechend Einzug in die Automobil-Branche. Ob mit dem Tempomat oder den automatischen Notbremssystemen, die Grundbausteine wurden bereits in der Vergangenheit gelegt. Trotzdem ist das autonome Fahren eine neue Dimension und erfordert sehr viel neue Technik. Das Potential und die Chancen für solche autonomen Fahrzeuge sind riesig. Jedoch sind auch die technischen Anforderungen sehr hoch, damit auch jeder Passagier sicher an sein Ziel kommt und alle Passanten sich angstfrei bewegen können.

Mit genau diesen Problemen ist auch die PREN Aufgabe 2015/2016 verbunden. Die vorliegende Arbeit soll aufzeigen, wie es möglich sein soll, ein autonomes Entsorgungsfahrzeug zu bauen. Dieses muss in der Lage sein auf einem achtförmigen Kurs selbstständig zwei Miniatur-Abfallcontainer zu entleeren und das Schüttgut anschliessend auf einer Deponie zu entsorgen.

Für das optimale Lösungskonzept wurde die Aufgabenstellung in viele kleine Teilprobleme aufgeteilt. Zu jedem Teilproblem wurden verschiedene Lösungsvarianten gesucht. Die erfolgversprechendsten Varianten wurden näher betrachtet und teilweise mit Funktionsmuster ausgetestet. Diese Arbeit zeigt zum einen auf, welche Teilaufgaben bearbeitet und zum anderen, welche Lösungen ausgewählt wurden. Des Weiteren sind die Projektstruktur und die Projektplanung aufgeführt. Diese Arbeit beschränkt sich auf die grundlegenden Konzepte. Detaillierte Konzepte oder Implementationen werden Bestandteil von PREN2 sein.

## 2 Lösungskonzept

### 2.1 Gesamtübersicht

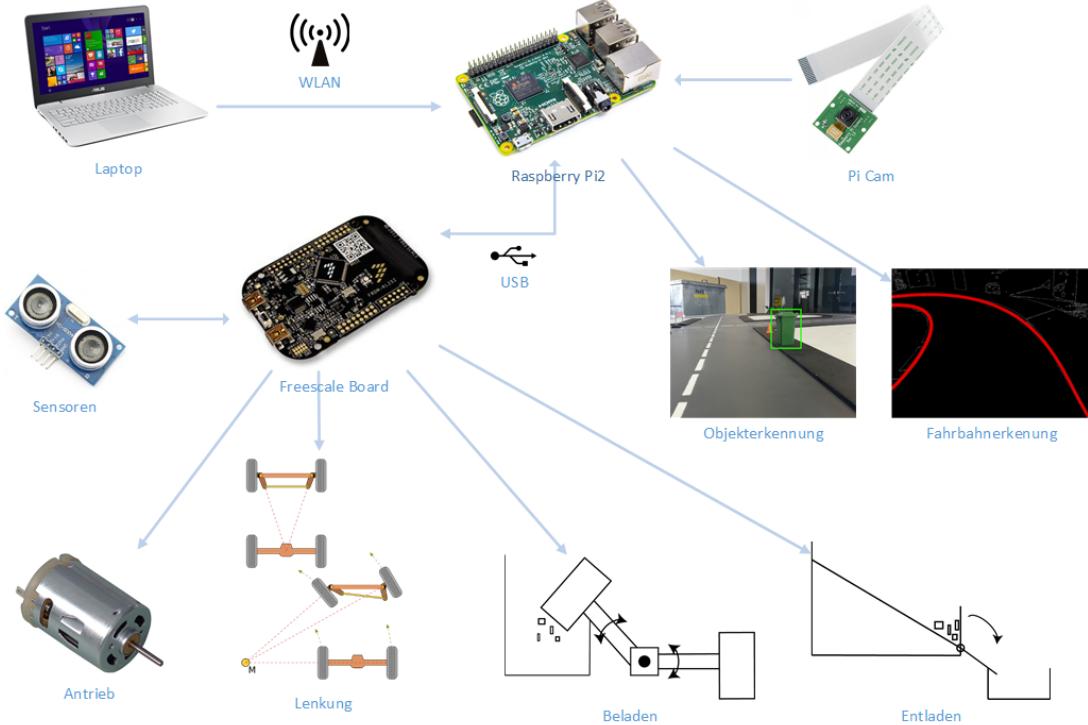


Abbildung 1: Übersichtszeichnung

#### 2.1.1 Zusammenspiel

Wie die obige Darstellung zeigt, besteht zwischen allen Komponenten ein enges Zusammenspiel. So wird beispielsweise die gesamte Bilderkennung über den Minicomputer gesteuert. Dieser erhält seine Daten durch die Kamera, aber auch von den Sensoren, welche über den Mikrocontroller an ihn gesendet werden. Nachdem die Informationen bearbeitet wurden, werden Befehle an den Mikrocontroller geschickt, welcher die Ansteuerung der Motoren übernimmt. Die Motoren wiederum, lösen die mechanischen Bewegungen aus, wie z.B. die Schwenkung der Kamera oder das Senken des Greifarmes.

### 2.1.2 Schnittstellen

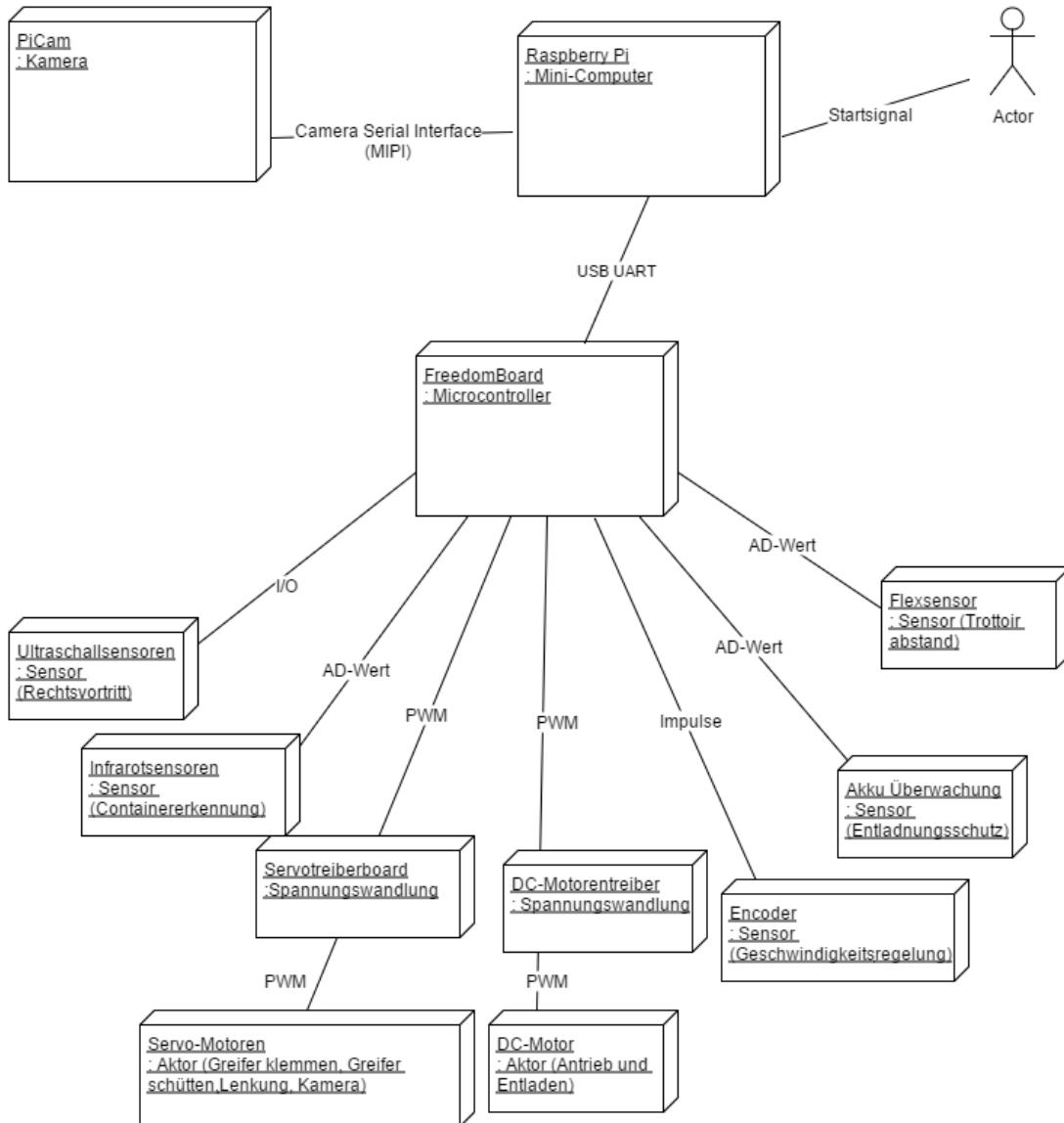


Abbildung 2: Schnittstellenübersicht

In Abbildung 2 wird eine detaillierte Ansicht der Komponente dargestellt, insbesondere was die Schnittstellen betrifft. So ist erkennbar, dass der Mikrocontroller als Hauptknoten fungiert und als Schnittstelle für analoge wie auch digitale Signale dient. Die Wahl der Schnittstellen ist bei fast allen Bauteilen vorgegeben. Zum Beispiel müssen die Ultraschallsensoren über I/O Pins angesteuert und ausgelesen werden. Anders sieht es bei der Verbindung Minicomputer und Mikrocontroller aus. Diese Schnittstelle war frei wählbar. Durchgesetzt hat sich UART<sup>1</sup> über USB. Dies hat den Vorteil, dass die Speisung und die Kommunikation über das selbe USB Kabel läuft. Zudem gibt es für UART bereits implementierte Funktionen auf Seiten- Mikrocontroller, wie auch Minicomputer.

<sup>1</sup>UART= Asynchrones serielles Kommunikationsprotokoll

### 2.1.3 Regelkreise

#### Geschwindigkeitsregelung

Damit das Fahrzeug mit einer konstanten Geschwindigkeit fahren kann, muss ein Regler eingebaut werden.

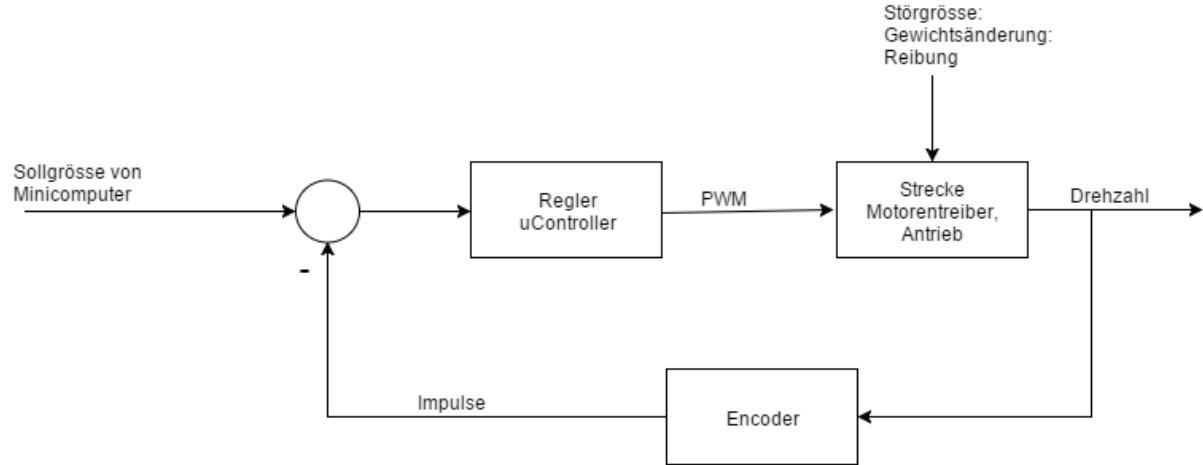


Abbildung 3: Grober Antriebsregelkreis

Hier ist der grobe Regelkreis des Antriebs zu sehen. Der Sollwert wird vom Minicomputer vorgegeben und an das Mikrocontrollerboard weitergeleitet. In diesem wird dann die Regelung implementiert. Der Regelausgang ist ein PWM<sup>2</sup>-Signal, welches die Geschwindigkeit des Antriebsmotors bestimmt. Die Drehzahl des Antriebmotors wird über einen Encoder zurück in den Mikrocontroller gelesen. Der Encoder misst die Drehzahl am Ausgang des DC-Antriebsmotors.

#### Lenkregelung

Damit das Fahrzeug auf der Strecke bleiben kann, braucht es einen Regelkreis für die Steuerung. Der grobe Regelkreis ist hier aufgezeigt.

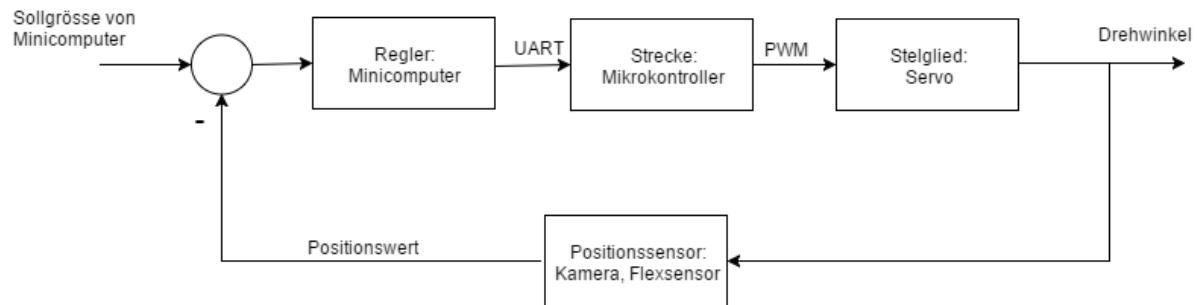


Abbildung 4: Grober Steuerungs Regelkreis

Wie hier zu sehen ist, befindet sich der Regler auf dem Minicomputer. Der Regelausgang wird über UART auf das Mikrocontrollerboard weitergeleitet. Dies wird ein Winkel in Grad sein, um wie viel sich der Servo drehen soll. Der Mikrocontroller ändert daraufhin das PWM Signal zum Servo. Die Positionsdetektion wird über die Kamera/Flexsensor realisiert. Das ist nur eine Übersicht über den Regelkreis.

<sup>2</sup>PWM= Pulsweiten Moduliertes Signal

## 2.2 Chassis

### Funktionsbeschrieb

Im Lösungskonzept wird ein Chassis mit vier Rädern verwendet. Die Hinterräder werden mit einem DC-Getriebemotor angetrieben. Der Motor ist über ein Differentialgetriebe mit der Hinterachse verbunden. Das Differentialgetriebe erlaubt den beiden Hinterrädern in der Kurve unterschiedlich schnell zu drehen. Die Vorderräder sind frei drehbar. Der Achsabstand soll ein Mass von 160mm nicht überschreiten. Für die Lenkung wird eine Achsschenkellenkung verwendet. Diese wird mittels eines Servomotors angetrieben. Wegen dem maximalen Achsabstand von 160mm ist der Servomotor zum Antrieb der Lenkung vor dem Fahrzeug verbaut.

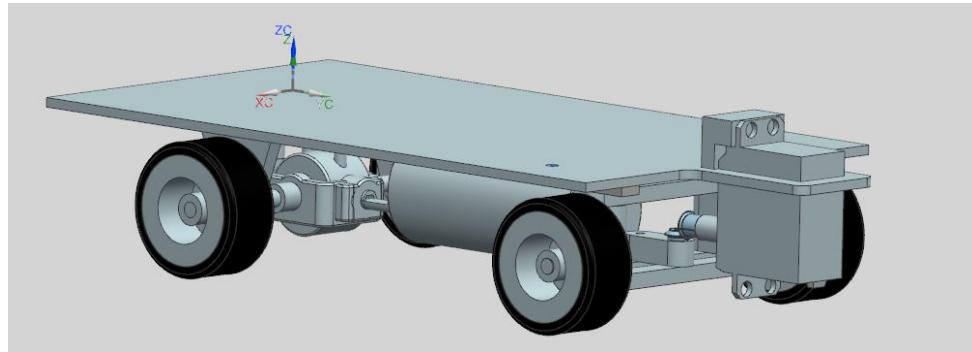


Abbildung 5: Konzept des Chassis

### Komponentenbeschrieb

Der Antriebsmotor ist ein 12 V Getriebemotor mit einer Getriebeuntersetzung von 1:18. Er hat somit eine Lastdrehzahl von 317 U/min. Zusammen mit der 2:1 Untersetzung des Differentialgetriebes, drehen die Hinterräder mit Durchmesser 43mm mit 158 U/min. Somit ist die maximale Geschwindigkeit des Fahrzeugs bei 35 cm/s.

Als Differentialgetriebe wurde ein Produkt vom Modellbauhersteller Relly gewählt.

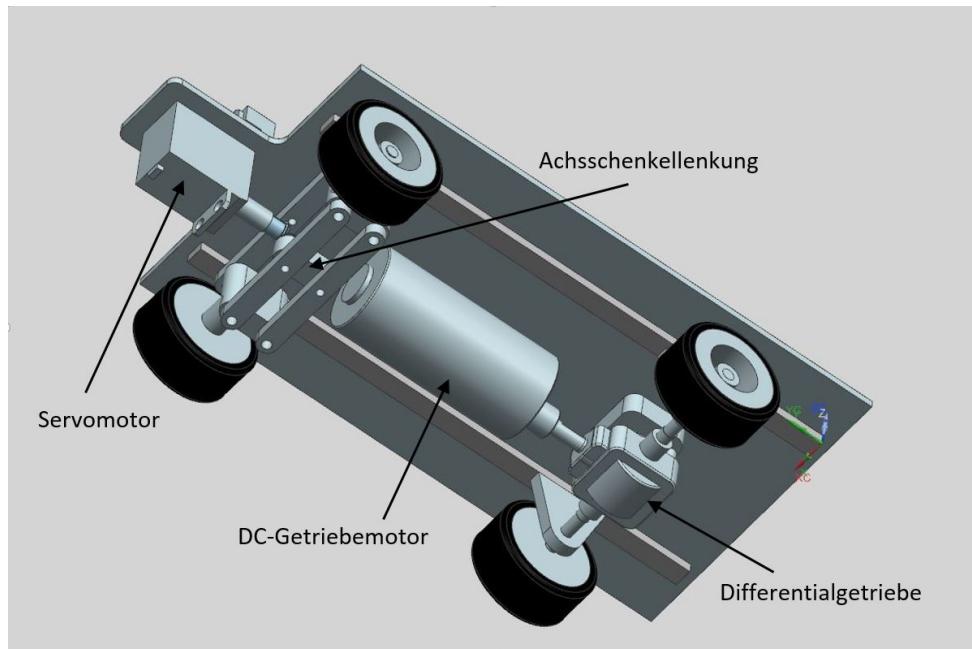


Abbildung 6: Konzept des Chassis mit Komponentenbeschriftung

## Begründung

Das Konzept mit vier Rädern, einem Antriebsmotor und einer Achsschenkellenkung hat sich durchgesetzt, weil die Regelung für die Spurhaltung wesentlich ruhiger ist als mit zwei angetriebenen Rädern. Des Weiteren sind die Lenkung und der Antrieb klar getrennt. Beim Antriebsmotor kann auf Schrittmotoren verzichtet werden. Ausserdem muss die Geschwindigkeit der Räder nicht sehr genau bestimmt werden.

## 2.3 Lenkung

### Funktionsbeschrieb

Die Lenkung ist eine Achsschenkellenkung. Sie wird heute in fast allen PKWs, LKWs, Omnibussen und sonstigen Nutzfahrzeugen eingesetzt. Bei dieser Art der Lenkung befinden sich die Räder auf einzeln lenkbaren Achsschenkeln, die jeweils mit einem Spurstangenhebel versehen sind. Die Spurstangenhebel sind ungefähr senkrecht zur Vorderachse bzw. ungefähr parallel zur Längsachse des Fahrzeugs ausgerichtet und durch eine Spurstange miteinander verbunden. Deshalb ist es möglich beide Räder gleichzeitig zu lenken.

### Komponentenbeschrieb

Mit einem Servomotor wird über eine Kegelradverbindung der Lenkschubhebel angetrieben. Dieser treibt wiederum die Spurstange an, welche die Bewegung über die Spurstangenhebel an die Räder weitergibt.

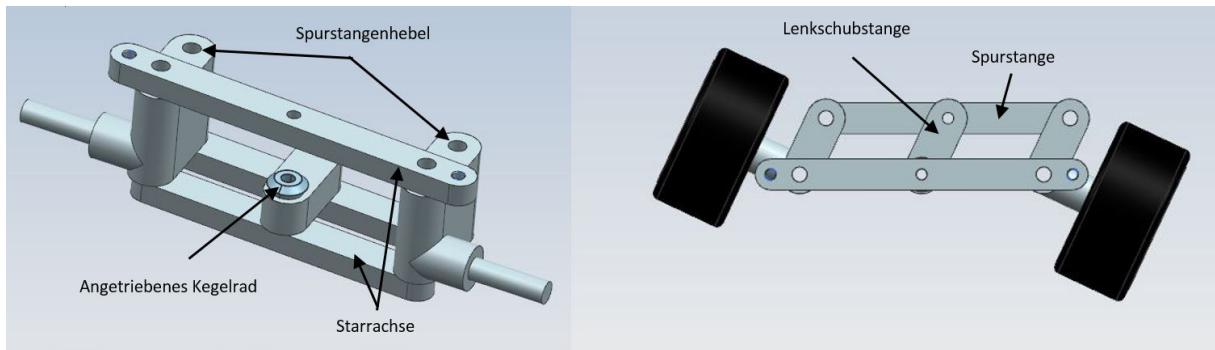


Abbildung 7: Konzept der Achsschenkellenkung

## Begründung

Für die Bildverarbeitung ist das Lenken und das konstante Ausgleichen der Fahrbahn und Fahrtgeschwindigkeit mit zwei Motoren, welche jeweils ein Rad antreiben ein Nachteil. Darum ist das Lösungskonzept mit einer Achsschenkellenkung ausgestattet. Von der konstruktiven Seite ist die Achsschenkellenkung im Vergleich zwar aufwändiger, aber für die Aufgabenstellung besser geeignet. Da es das Ziel ist, einen möglichst konstanten Abstand zum Trottoir der Fahrbahn zu halten, ist die Regelung einer Achsschenkellenkung einfacher. Weitere Gründe, welche für die Achsschenkellenkung sprechen, wurden schon im Kapitel „Chassis“ erläutert. Die Begründung für die Wahl des Servomotors ist, dass die Lenkung keine 360°-Bewegungen durchführen muss. Zudem ist die Drehzahl des Servos leicht zu steuern, ohne dass zusätzlich eine Regelung eingebaut werden muss.

## Berechnungen

### Berechnung Drehmoment Servo Lenkung

- Gewichtskraft auf jedes Rad  $F_g = \frac{m \cdot g}{4}$
- Abstand  $l = 30mm$
- $M = 2 \cdot F_g \cdot l = 0.37Nm$
- Sicherheitsfaktor = 1.5 (da Beschleunigung- und Bremskräfte vernachlässigt werden)
- $M = 0.37 \cdot 1.5 = 0.555Nm = 55.5Ncm$

## 2.4 Antrieb

### Funktionsbeschrieb

Als Antrieb wird ein DC-Getriebemotor verwendet, der an der Unterseite montiert wird und für die Vor- sowie Rückwärtsbewegungen des Fahrzeugs zuständig ist. Die aktuelle Drehzahl des Motors wird von einem Encoder erfasst. Dieser wiederum sendet die gemessenen Daten an den Mikrocontroller der schlussendlich die Anzahl Umdrehungen reguliert.

### Komponentenbeschrieb



Abbildung 8: Antrieb (Quelle:<http://www.conrad.ch>)

Als voraussichtlicher Favorit wurde ein Hochleistungsgetriebemotor von Modelcraft ausgewählt. Die technischen Daten sind folgende:

- Leerlaufstrom: 0.32 A
- Last-Drehzahl: 317 U/min
- Leerlauf-Drehzahl: 333 U/min
- Betriebsspannung: 12 V DC
- Spitzendrehmoment: 2.23 Nm
- Max. Laststrom: 0.7 A

### Begründung

Der ausgewählte Motor punktet vor allem aufgrund seiner kleinen und kompakten Bauform. Da der Montageort für den Antrieb am Fahrzeug nicht verändert werden kann, ohne umständliche Wellen zu installieren, ist die Baugröße relativ beschränkt. Ein weiterer Punkt ist, dass er sich als Gleichstrommotor leicht ansteuern lässt und damit die Handhabung vereinfacht. Zudem zeichnet er sich mit einer hohen Drehzahl und einem grossen Drehmoment aus und ist dennoch verhältnismässig günstig in der Anschaffung.

## 2.5 Beladen und Greifer

### Funktionsbeschrieb

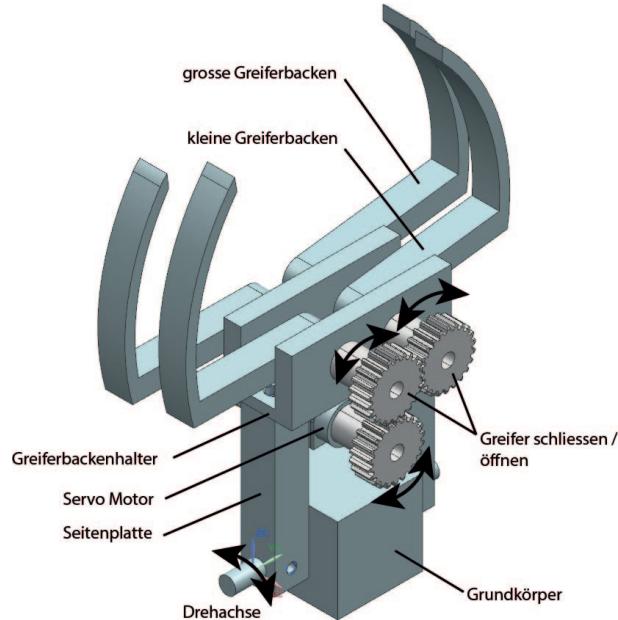


Abbildung 9: Greifer

Der Greifer ist auf der Grundplatte des Fahrzeuges montiert. Im Grundkörper wird die Welle für die Drehung des Greifers montiert. Die Welle wird von einem Servo Motor angetrieben. Um gute Laufeigenschaften zu erreichen werden Lagerbüchsen eingebaut. Auf der Drehachse sind die beiden Seitenplatten des Greifers montiert. Auf den Seitenplatten wird der Greifbackenhalter verschraubt. Der Servo Motor für die Funktion „Greifer schliessen/öffnen“ wird am Greifbackenhalter befestigt. Auf dem Servo wird ein Zahnrad montiert, das über die gleichen Zahnräder die Greiferbacken öffnet bzw. schliesst. Die Greiferbacken sind so konzipiert, dass die 2 längeren und 2 kürzeren an verschiedenen Flächen greifen. Damit wird erreicht, dass der Container während des ganzen Einladeprozesses stabil bleibt.

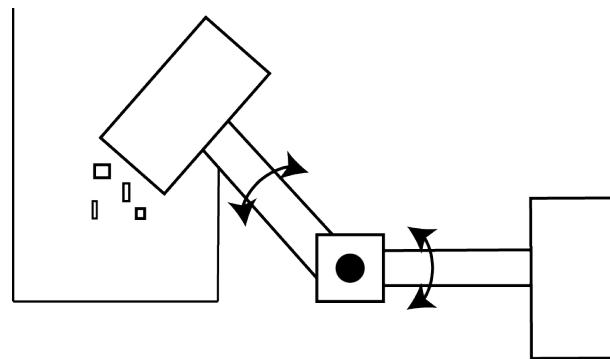


Abbildung 10: Einladeprozess

Der Greifer ist während des Fahrens in der oberen Position gelagert, damit die Abmasse der Aufgabenstellung eingehalten werden können. Der Greifer wird erst auf horizontale Richtung bewegt, nachdem der Wagen still steht und richtig positioniert ist. Die rechte Fahrzeugkante wird auf 25mm +/- 5mm zum Trottoirrand positioniert. In Fahrtrichtung beträgt die maximale zulässige Positionierungsgenauigkeit +/- 10mm.

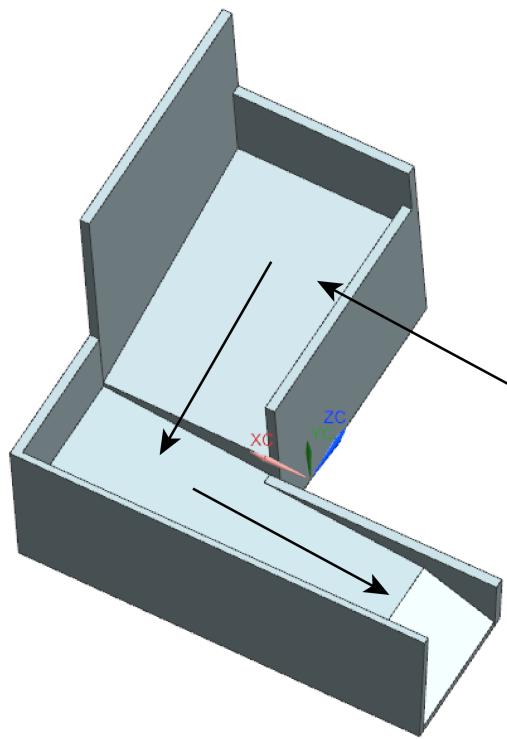


Abbildung 11: Schüttgutfluss in den beiden Behältern

Der Container wird in den ersten Behälter entleert. Durch den schrägen Boden rutscht das Schüttgut in den zweiten Behälter. Am Ende des zweiten Behälters wird eine Klappe montiert, damit die Schrauben und Muttern nicht herausfallen. Die Klappe wird durch einen DC-Motor gehalten und erst am Schluss beim Entladetank geöffnet.

### Komponentenbeschrieb

- Zahnräder aus Kunststoff, Durchmesser = 15 (Einkaufsteil)
- Greiferbacken aus Kunststoff (Druckteil)
- Grundkörper, Seitenplatte, Greifbackenhalter und Wellen werden aus Aluminium gefertigt.
- Behälter aus Acrylglas 3mm Dicke (Laserzuschnitt)
- Technische Daten des Motors Greifer:
  - Stellzeit bei 4.8 Volt: 0.1 sec ( $50^\circ$ )
  - Stellzeit bei 6 Volt: 0.08 sec ( $50^\circ$ )
  - Betriebsspannung: 4.6/6 V
  - Stell-Moment bei 4.8 Volt: 18 Ncm
  - Stell-Moment bei 6 Volt: 20 Ncm

### Berechnungen

Servo Motor für Greifer schliessen / Berechnung des benötigten Drehmoments:

- Haftriebung auf 0.7 geschätzt
- Masse Container  $m = 0.075\text{kg}$
- Gewichtskraft =  $m \cdot g = 0.74\text{N}$

- Kraft  $F = \frac{0.5 \cdot 0.74}{0.7} = 0.53N$
- Abstand  $l = 80mm$  (Drehpunkt zu Greiffläche)
- $M = F \cdot l = 0.04Nm = 4Ncm$
- Sicherheitsfaktor = 2
- $M = 4 \cdot 2 = 8Ncm$

Servo Motor für Greifarm drehen:

- Gewichtskraft Container  $F_c = 0.075kg \cdot 9.81 = 0.74N$
- Gewichtskraft Greifarm:  $F_{gr} = 0.3kg \cdot 9.81 = 2.94N$
- Benötigtes Moment:  $M = F_{gr} \cdot 0.05m + F_c \cdot 0.1m = 0.147Nm + 0.07Nm = 0.217Nm = 21.7Ncm$
- Sicherheitsfaktor = 2 wegen Abschätzung des Greifergewichts und vernachlässigter Trägheitskräfte
- $M = 2 \cdot 21.7 = 43.4Ncm$

## 2.6 Entladen

### Funktionsbeschrieb

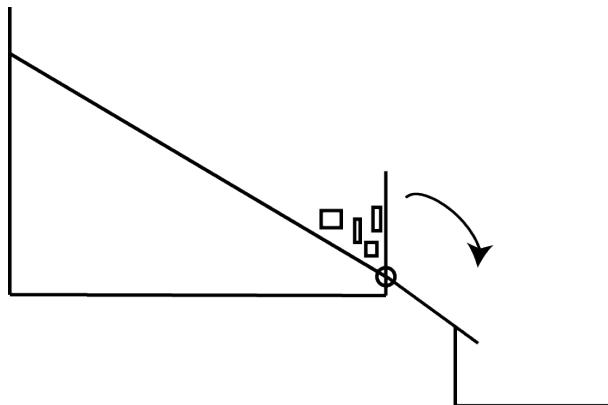


Abbildung 12: Entladen

Beim Entladevorgang fährt das Fahrzeug bis auf 25mm +/- 5mm an den Rand der rechten Seitenlinie. Anschliessend wird die Klappe gelöst und sie fällt auf den Rand des Entsorgungsbeckens. Über die Klappe rutscht das Schüttgut in das Entsorgungsbecken. Nach erfolgter Abladung wird die Klappe wieder nach oben gezogen und ist wieder in der Ausgangsstellung. Damit die Klappe wieder in Ausgangsstellung gebracht werden kann, werden oben rechts bzw. links Bohrungen angebracht. Dort können Seile befestigt werden. Diese werden auf Rollen auf- bzw. abgewickelt. Die beiden Rollen werden auf einer Welle befestigt, die von einem DC-Motor angetrieben wird.

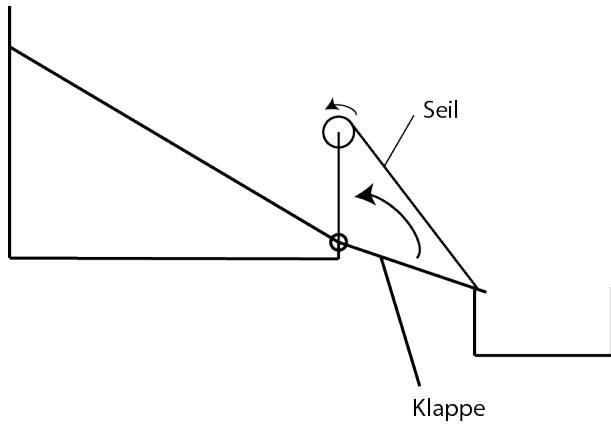


Abbildung 13: Klappe schliessen

### Komponentenbeschrieb

Die Klappe besteht aus einem handelsüblichen Scharnier auf dem eine Platte aus Acrylglass befestigt ist.

### Berechnungen

DC-Motor für Entladeklappe:

- $m = 0.15\text{kg}$
- $l = 0.1\text{m}$
- $M = m \cdot g \cdot l = 0.015\text{Nm} = 1.5\text{Ncm}$
- Sicherheitsfaktor =2
- $M = 1.5 \cdot 2 = 3\text{Ncm}$

## 2.7 Energieversorgung

### Funktionsbeschrieb

Das autonome Entsorgungsfahrzeug muss mit Energie versorgt werden. Dazu werden Akkumulatoren eingesetzt, welche das Gerät während dem gesamten Einsatz mit Strom versorgen.

### Komponentenbeschrieb

Um Störungen, die z.B. durch den erhöhten Anlaufstrom der Motoren verursacht werden könnten, zu vermeiden, werden die intelligenten Systeme (Microcontroller, Mini-Computer,etc.) möglichst getrennt von den Motoren gespiesen. Dazu werden zwei Lithium-Polymer-Akkumulatoren mit einer voraussichtlichen Spannung von 11.1 und 7.4 Volt verwendet, die galvanisch getrennt sind. Der 11.1-Volt Akkumulator wird die Versorgung der Motoren gewährleisten und eine Kapazität von 2000-2400 mAh aufweisen. Für die empfindlicheren Systeme wird der 7.4 Volt LiPo zuständig sein, der eine Kapazität von 500-1000 mAh besitzt. Die genaue Evaluation der Kapazitäten ist noch nicht abgeschlossen und bezieht sich lediglich auf die Resultate der Berechnungen, welche im Verlauf des Kapitels noch beschrieben werden.

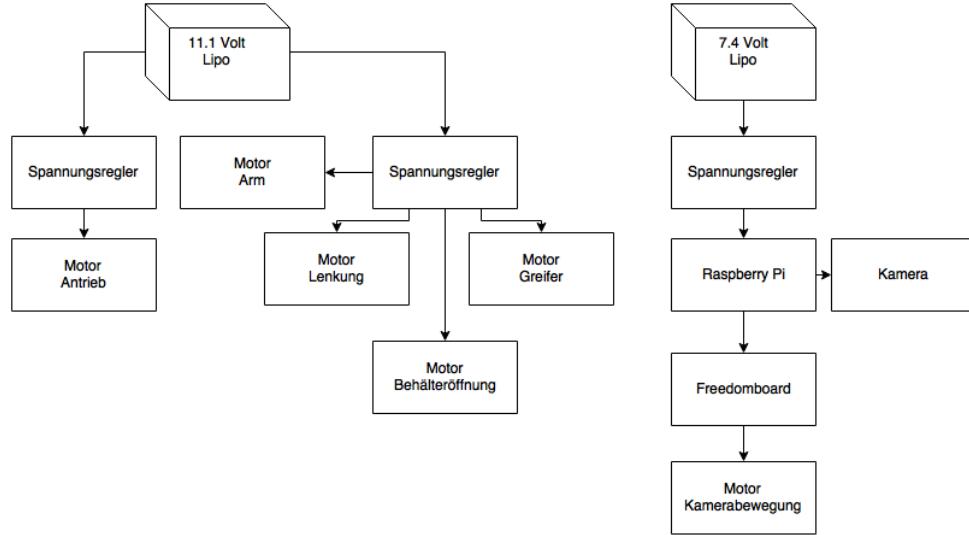


Abbildung 14: Aufteilung der Akkumulatoren

### Begründung

Die Entscheidung wird damit begründet, dass Lithium-Polymer-Akkumulatoren in einem vielfältigen Sortiment erhältlich sind und damit eine grosse Flexibilität bei der Auswahl ermöglichen. Somit würde sich auch die Suche nach einem Ersatz bei allfälligen Änderungen vereinfachen. Außerdem weisen LiPos, im Vergleich zu anderen Akkumulatoren, eine wesentlich kompaktere Bauform auf, was entscheidend für die Auswahl war.

### Berechnungen

Da die Wahl der Motoren noch nicht definitiv feststeht, wurden für die Berechnungen solche genommen, die den Anforderungen am nächsten kommen.

Leistungsberechnungen:

- Servomotor für die Lenkung:

$$P = 2 \cdot \pi \cdot N \cdot M \rightarrow 2 \cdot \pi \cdot 0.92 \frac{U}{sec} \cdot 0.65 Nm = 3.76W$$

- Servomotoren für Kamerabewegung und Greifer:

$$P = 2 \cdot \pi \cdot N \cdot M \rightarrow 2 \cdot \pi \cdot 0.72 \frac{U}{sec} \cdot 0.18 Nm = 0.81W$$

- Servomotor für den Greiferarm:

$$P = 2 \cdot \pi \cdot N \cdot M \rightarrow 2 \cdot \pi \cdot 1.14 \frac{U}{sec} \cdot 0.32 Nm = 2.5W$$

- DC-Motor für den Antrieb:

$$P = 2 \cdot \pi \cdot N \cdot M \rightarrow 2 \cdot \pi \cdot 5.258 \frac{U}{sec} \cdot 2.23 Nm = 73.98W$$

- DC-Motor für die Behälteröffnung:

$$P = 2 \cdot \pi \cdot N \cdot M \rightarrow 2 \cdot \pi \cdot 16.6 \frac{U}{sec} \cdot 0.028 Nm = 2.9W$$

Kapazitätenberechnungen:

- DC-Motor für den Antrieb:

$$\frac{P \cdot t}{U} \rightarrow \frac{73.98W \cdot 0.25h}{12V} = 1.54Ah$$

- DC-Motor für die Behälteröffnung:

$$\frac{P \cdot t}{U} \rightarrow \frac{2.9W \cdot 0.25h}{4.8V} = 0.15Ah$$

- Servomotor für die Lenkung:

$$\frac{P \cdot t}{U} \rightarrow \frac{3.76W \cdot 0.25h}{4.8V} = 0.19Ah$$

- Servomotor für den Greiferarm:

$$\frac{P \cdot t}{U} \rightarrow \frac{2.5W \cdot 0.25h}{4.8V} = 0.13Ah$$

- Servomotoren für Kamerabewegung und Greifer:

$$\frac{P \cdot t}{U} \rightarrow \frac{0.81W \cdot 0.25h}{4.8V} = 0.04Ah$$

- Mini-Computer:

$$I \cdot t \rightarrow 2A * 0.25h = 0.5Ah$$

- Benötigte Kapazität für den 11.1 Volt Akkumulator:

$$1.54Ah + 0.15Ah + 0.19Ah + 0.13Ah + 0.04Ah = 2.05Ah$$

- Benötigte Kapazität für den 7.4 Volt Akkumulator:

$$0.5Ah$$

## 2.8 Intelligente Systeme

### 2.8.1 Softwarekonzept

#### Funktionsbeschrieb

Das Softwarekonzept basiert auf dem MVC<sup>3</sup>-Prinzip, wobei sich die View auf einen reinen Debug-Zweck beschränkt. Dementsprechend soll die View über eine Schnittstelle nach aussen verfügbar gestaltet werden, um auf dem Notebook entsprechende Debug-Komponenten betreiben zu können. Ebenfalls über Methoden nach aussen offen gelegt, sollen die Regelungsparameter zur Laufzeit angepasst werden können. Im produktiven Betrieb ist diese Schnittstelle gemäss Aufgabenstellung geschlossen. Dabei hat der Controller alle Subprozesse sowie die Schnittstelle zur Hardwaresteuerung, via Mikrocontroller-Board realisiert, unter Kontrolle. Die parallel laufenden Subprozesse sind:

- Bilderzeugung,
- Objekterkennung und
- Fahrbahnerkennung.

---

<sup>3</sup>MVC = Model-View-Control

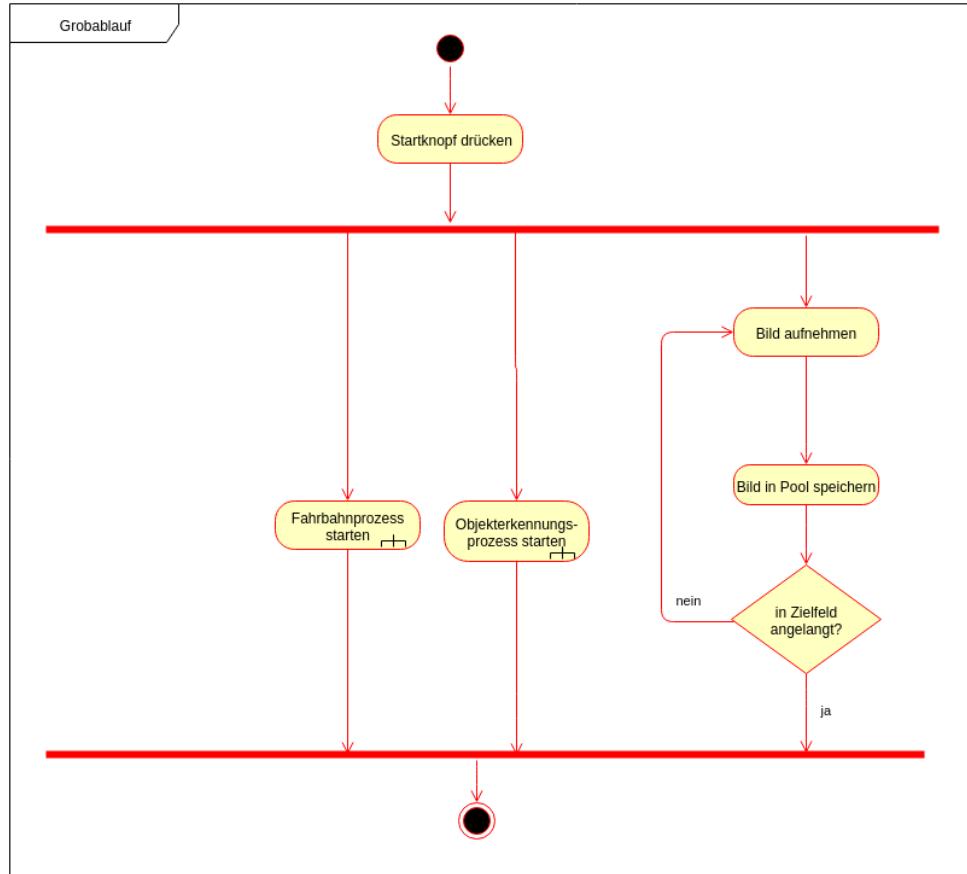


Abbildung 15: Aktivitätendiagramm Grobablauf

Der Zustand des Fahrzeuges wird im Model gespeichert und steht allen Prozessen zur Verfügung. Die Datenstruktur ist dabei so aufgebaut, dass keine Zugriffskonflikte entstehen sollten.

Der Controller behandelt alle Anweisungen der Prozesse. Meldet beispielsweise die Objekterkennung einen Container, hat dieser Priorität vor dem normalen Fahren und das Mikrocontroller-Board erhält die entsprechenden Anweisungen.

### Komponentenbeschrieb

Umgesetzt wird das Konzept auf dem Minicomputer in C++. Dabei werden die zuvor beschriebenen Prozesse als eigene Threads realisiert. Um eine hohe Softwarequalität sicherzustellen, soll möglichst nach TDD<sup>4</sup> gearbeitet werden. Für die automatisierten Tests werden die Open-Source-Pakete Google-Test und Google-Mock verwendet, welche in C++ realisiert und optimiert sind und sich direkt in das Projekt integrieren lassen.

Das Mikrocontroller-Board wird aus Effizienzgründen mit der Programmiersprache C programmiert.

### Begründung

Der geplante Minicomputer verfügt über einen Vierkernprozessor, weswegen eine Parallelisierung zwingend notwendig wird, um diesen Prozessor auch vollumfänglich nutzen zu können. Bei der Programmiersprache ist C++ aufgrund der Performance und der einfachen Kopplung an C gewählt worden. Zudem sind keine Zusatzttools wie virtuelle Maschinen notwendig. Die verwendete Bildverarbeitungssoftware OpenCV ist ebenfalls in C++ entwickelt und somit direkt nutzbar.

TDD soll sicherstellen, dass vor allem bei kurzfristigen Änderungen keine unerwünschten Nebeneffekte entstehen und jederzeit sichergestellt ist, dass die Software wunschgemäß läuft.

<sup>4</sup>TDD = Test Driven Development

### 2.8.2 Mini-Computer

#### Funktionsbeschrieb

Der Mini-Computer beinhaltet die Logik- und Ablaufsteuerung, nimmt Bilder von der Kamera entgegen und verarbeitet diese um Container zu orten und in der Mitte der Fahrbahn zu bleiben. Die Lenkkorrektur und weitere Befehle zur Steuerung der Mechanik werden via USB Schnittstelle an das Mikrocontroller-Board weitergeleitet.

#### Komponentenbeschrieb

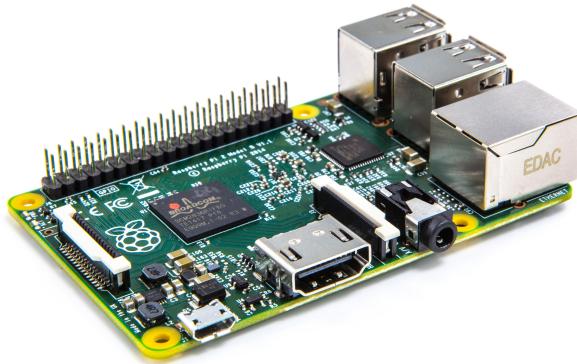


Abbildung 16: Raspberry Pi2 Model B (Quelle:<http://pcworld.com/>)

Das Raspberry Pi2 bietet mit einem 900MHz Quadcore Prozessor, einem eigenem Grafikchip und 1GB Ram viel Leistung und ist ideal geeignet.

#### Begründung

Dem Team war von Anfang an klar, dass die Bildverarbeitung viele Ressourcen verbrauchen wird. Gleichzeitig darf der Energieverbrauch die Kapazität eines Akkus nicht übersteigen. Das Raspberry Pi2 besitzt genügend Leistung und ist trotzdem sparsam. Mit einem Quadcore und 1GB RAM ist es ausreichend leistungsfähig um Bilder zeitig zu verarbeiten. Zusätzlich ist es einfach zu verwalten und zu bedienen, da es über ein eigenes Betriebssystem verfügt.

### 2.8.3 Mikrocontroller-Board

#### Funktionsbeschrieb

Das Mikrocontrollerboard bildet die Schnittstelle zwischen der Hardware (Motoren und Sensoren) und dem Minicomputer. Das Mikrocontrollerboard übernimmt die Ansteuerung und Auswertung der einzelnen Komponenten und stellt die verarbeiteten Informationen dem Minicomputer über eine Schnittstelle zur Verfügung.

## Komponentenbeschrieb



Abbildung 17: Freedomboard KL25 von Freescale (Quelle:<http://ch.farnell.com/>)

Als Mikrocontrollersystem wurde das Freedomboard KL25Z von Freescale ausgewählt. Das Entwicklungsgesboard bietet vieles, wie ausreichend I/O's, AD Wandler und Timerausgänge. Das Board wird mit der Programmiersprache C programmiert.

### Begründung

Das Board überzeugt durch eine gute Rechenperformance zu einem kleinen Preis. Das Hauptargument für das Freedomboard ist die einfache Programmierung. Für viele Komponenten wie Servos oder Ultraschalsensoren steht ein Tool namens ProcessorExpert zu Verfügung. Dieses Tool ermöglicht eine relativ einfache Anbindung solcher Komponenten. Dies spart sehr viel Entwicklungsaufwand. Das Freedomboard hat sich gegen das Tinkerforgesystem durchgesetzt. Der Hauptgrund ist die Flexibilität des Freedomboards gegenüber dem Tinkerforgesystem. Dieses ist sehr einfach zu bedienen solange alle Komponenten von Tinkerforge zur Verfügung gestellt werden. Fehlt aber ein Modul wird es sehr schnell kompliziert. Beim Freedomboard ist der Aufwand für eine Komponente etwas höher, dafür können die meisten Systeme angeschlossen werden.

### Testergebnisse

Das Freedomboard wurde als Funktionsmuster bereits in Betrieb genommen. Es hat sich gezeigt, dass die Anbindung der Komponenten tatsächlich sehr einfach ist. Es wurden bereits ein Ultraschallsensor, mehrere Servos, eine UART Kommunikation und ein Infrarotsensor angebunden. Das System lieferte dabei sehr vielversprechende Ergebnisse.

### Software Grundaufbau

Das Tool ProcessorExpert bietet eine Komponente FreeRTOS an. Das ist ein Betriebssystem, welches das Programmieren sehr vereinfacht. Damit können verschiedene Tasks definiert werden und quasi parallel ausgeführt werden. Die Software wird damit geplant. Folgende Tasks sind geplant:

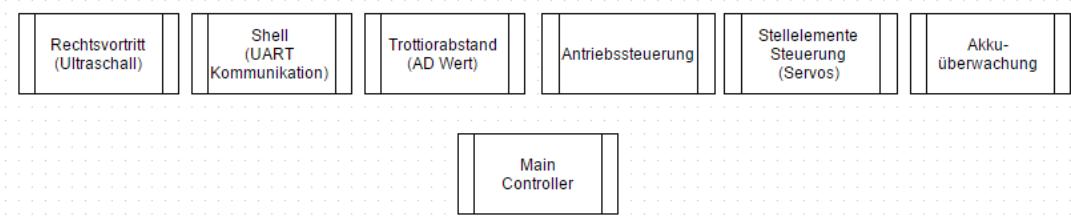


Abbildung 18: Übersicht MC Tasks

Die Implementierung der Tasks wird im Modul Produkteentwicklung zwei (PREN2) umgesetzt.

## 2.9 Bilderzeugung

### Funktionsbeschrieb

Die Bilderzeugung stellt die Schnittstelle zur Kamera sicher und speichert die erzeugten Bilder fortlaufend in einer von OpenCV zu Verfügung gestellten Datenstruktur. Dabei wird den übrigen Prozessen jeweils nur das aktuellste Bild der Kamera zur Verfügung gestellt.

Als Kamera kommt die Raspberry Pi CAM zum Einsatz (Abbildung 18), die über eine MIPI Schnittstelle direkt an den Minicomputer angeschlossen und kostengünstig erworben werden kann. Das Bildformat lässt sich über den Open Source Treiber einstellen, was zu einer Performanceverbesserung beiträgt. Der Winkel von 53° ist etwas knapp, daher wird die Kamera drehbar auf dem Fahrzeug montiert. Diese Vorrichtung soll für die Kurvenfahrt sowie den Rechtsvortritt genutzt werden. Der Neigungswinkel der Kamera lässt sich ebenfalls einstellen, um die benötigte Flexibilität der Sichtposition der Kamera sicherzustellen.

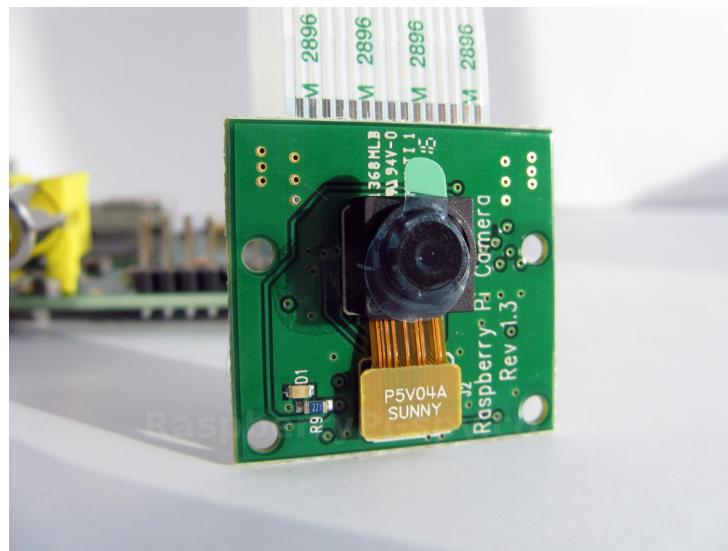


Abbildung 19: Raspberry Pi Kameramodul

### Komponentenbeschrieb

Die Bilderzeugung ist ein eigener als Thread realisierter Subprozess und liefert über die Methode `GetImage()` den Pointer auf das Bild zurück. Die verfügbare Datenstruktur ist dabei `cv::Mat` in Farbe mit einer Auflösung von 640x320 Pixel. Die verwendenden Prozesse können sich dann den Bildinhalt zeitgleich und unabhängig voneinander abgreifen und bedarfsgerecht weiterverarbeiten. Gestartet wird der Prozess automatisch bei der Objekterzeugung im Konstruktor und kann über die Methode `StopRecording()` angehalten werden, sobald das Ziel erreicht worden ist. Dem Konstruktor wird als Übergabeparameter der Pointer auf den `PrenController` mitgegeben, so dass im Störungsfall entsprechende Meldungen an den Controller übergeben werden können.

### Begründung

Der beschriebene Lösungsansatz bietet den Vorteil, dass die Fahrbahnerkennung und die Objekterkennung, welche diesen Prozess benutzen, unabhängig voneinander ihre Weiterverarbeitung parallel durchführen können. So wird der Mehrkernprozessor des Minicomputers optimal genutzt. Weiter sollte eine Synchronisation der `GetImage()`-Methode nicht notwendig sein, da das erzeugte Bild nicht verändert, sondern nur genutzt wird, was sich wiederum positiv auf die Performance auswirkt.

### Testergebnisse

Die Tests mit dem Minicomputer und der entsprechenden Kamera zeigen, dass ca. 20-30 Frames pro Minute zu Verfügung gestellt werden. Die Bilder sind dabei in guter Qualität und mit wenig Glanzeffekten entstanden. Es könnte dennoch von Vorteil sein, den Glanz wegzufiltern, um Störungen zu vermeiden. Die parallelen Zugriffe haben sich in den Tests bewährt und konfliktfrei funktioniert.

## 2.10 Fahrbahnerkennung

### Funktionsbeschrieb

Die Fahrbahnerkennung soll primär mittels Kamera umgesetzt werden. Dazu werden Bilder aus dem zur Verfügung gestellten Bilderpool entnommen, mit OpenCV in Graustufen umgewandelt und anschliessend einer Kantenerkennung unterzogen. Dazu wird ein eigener Algorithmus verwendet, der das ganze Bild als Funktion  $z = f(x, y)$  anschaut, wobei  $x$  der Pixelkoordinate der Spalte,  $y$  der Pixelkoordinate der Zeile und  $z$  dem Graustufenwert des Pixels entspricht (Abbildung 19).



Abbildung 20: Bild nach Kantenerkennung

Anhand der vorhandenen Informationen kann die Fahrbahn und demzufolge laufend der Korrekturvektor ermittelt werden. Die eigentliche Korrektur soll mittels PID-Regelung realisiert werden, um eine ruhige Fahrt zu erreichen. Die Korrekturanweisungen werden dabei vom Controller berechnet, und in Grad an das Mikrocontrollerboard weitergegeben.

### Komponentenbeschrieb

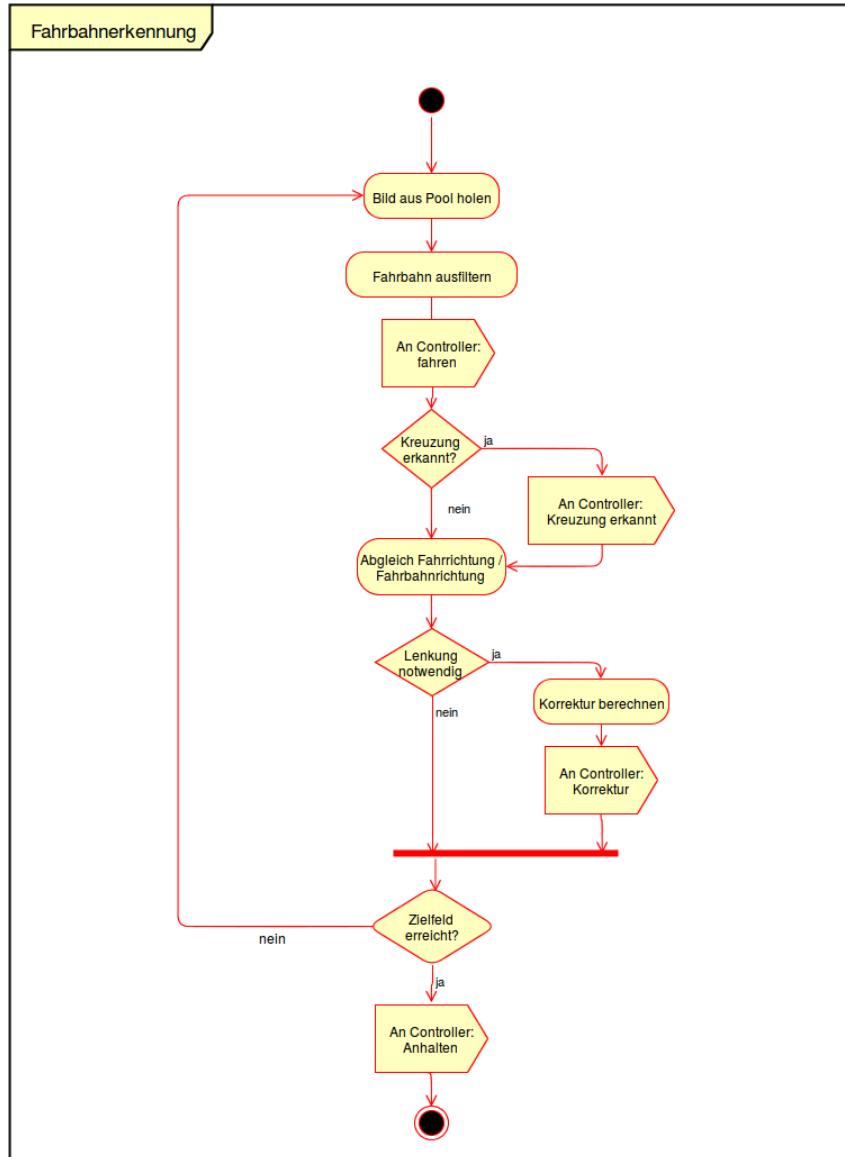
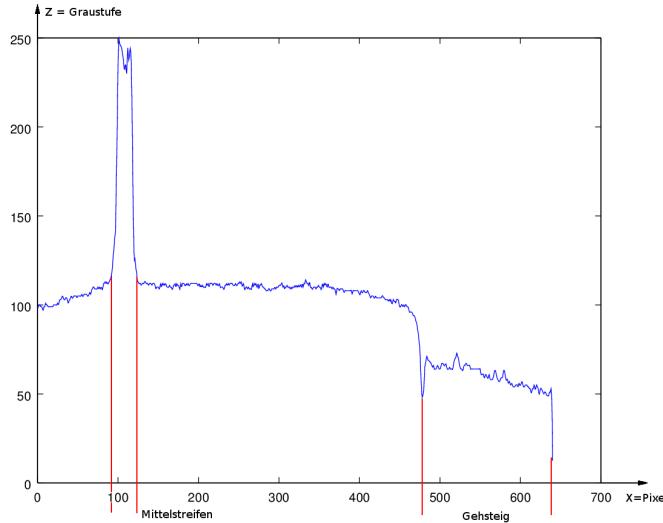


Abbildung 21: Aktivitätendiagramm Fahrbahnerkennung

Die Fahrbahnerkennung wird als eigener Thread auf dem Raspberry Pi realisiert und in C++ objekt-orientiert umgesetzt. Die erhaltenen Bilder werden vor der Analyse auf ein Format von 320x230 Pixel verkleinert. Für die Kantenerkennung ein Bild zeilenweise durchlaufen und die Änderungsrate der Graustufe des nächsten Pixels rechts davon ( $x$ -Richtung) und aus der darüberliegenden Zeile ( $y$ -Richtung) betrachtet. Überschreitet diese den Filterwert, wird eine Kante erkannt und auch deren Richtung festgehalten (Abbildung: 21).

Im Anschluss wird im unteren Bereich des Bildes von innen nach aussen die Fahrbahn gesucht. Dazu werden jeweils von der Bildmitte aus, in beide Richtungen, die ersten Kanten gesucht, über mehrere Punkte fixiert und danach alle übrigen Kanten entfernt.

Abbildung 22: Graph der Graustufenwerte einer Bildzeile in  $x$ -Richtung

Da die Fahrbahn keine sprunghaften Änderungen aufweisen kann, wird die kantendetektionierte Matrix von unten nach oben durchgesehen und alle Kanten, die in  $x$  - Richtung eine zu grosse Abweichung von der erkannten Fahrbahn aufweisen entfernt, so dass nur noch die rechte und linke Fahrbahnbegrenzung übrig bleibt (Abbildung 22 und 23). Lücken werden geschlossen, indem in Richtung des letzten Elementes fortgefahren wird.

Für die anschliessende Suche der Ideallinie wird von den gefundenen Kanten die Mitte genommen. Die Abweichung wird ermittelt, indem der Fahrtrichtungsvektor mit dem Richtungsvektor der Ideallinie abgeglichen wird. Bei einem zu grossen Winkelfehler wird eine Korrektur eingeleitet. In der Kurve wird dazu der Fahrtrichtungsvektor durch einen Bogen, entsprechend dem eingeschlagenen Radius der Lenkung, ersetzt und mit dem Sollradius der Fahrbahn abgeglichen. Die Abweichung wird dann über die Tangentenfunktion in einen Winkel umgewandelt, um für die Regelung nur einen Wertetyp auswerten zu müssen. Da die Kurven, insbesondere auf der Innenbahn, sehr eng sind, verliert die Kamera die Fahrbahn aus der Sicht. Um dieses Problem zu lösen wird die Kamera beim Kurveneingang auf einen festgelegten und konstanten Winkel gedreht, der dann wiederum in der Regelung beachtet werden muss. Am Ende der Kurvenfahrt wird die Kamera wieder auf die Null-Grad-Position zurückgeschwenkt. Nebst der besseren Fahrbahnsicht werden auch Objekte auf diese Weise früher erkannt.

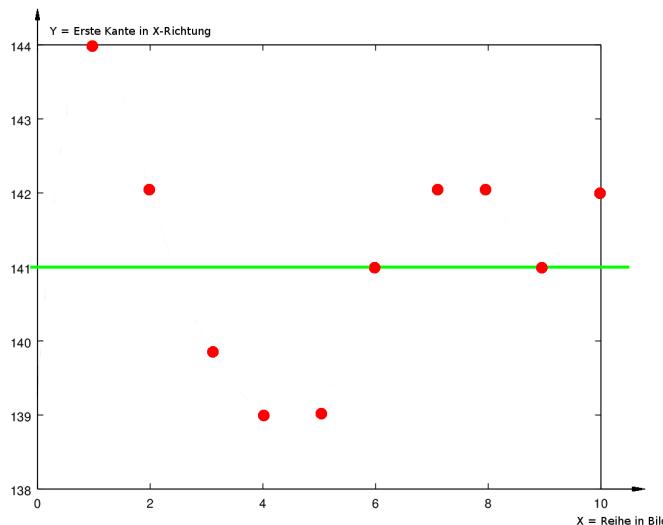


Abbildung 23: Erste Kante links der Mitte

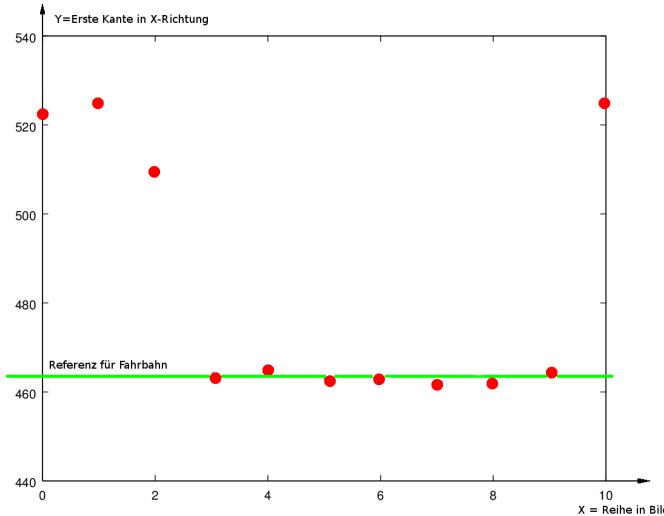


Abbildung 24: Erste Kante rechts der Mitte

### Begründung

Mit der beschriebenen Vorgehensweise kann der Aufwand pro Bild gegenüber OpenCV deutlich reduziert werden, da die Operationen ausschliesslich für die Fahrbahnerkennung erstellt werden. Die Tests haben die Realisierbarkeit dieser Vorgehensweise als machbar und flexibel aufgezeigt.

### Berechnungen

Abschätzung der Komplexität des Algorithmus: Jedes Bild wird als eine  $n \times m$  Matrix  $M$  betrachtet, wobei  $n = \text{Spaltenzahl}$  und  $m = \text{Zeilenzahl}$ . Die Resultate der Kantenerkennung werden in einer zweiten  $n \times m$  Matrix  $M_1$  gespeichert, welche für die weiteren Berechnungen verwendet wird. Anhand des Bildformats gilt:  $n > m$ . Die Umwandlung des Bildes in Graustufen, von OpenCV durchgeführt, ist in der Abschätzung nicht enthalten.

- Iteration für Kantenerkennung:

$$n \cdot m \leq n \cdot n \text{ wenn } n \geq m \rightarrow \mathcal{O}(n^2) \text{ mit den Zeugen } C = 1 \text{ und } k = 1$$

- Iteration für Kantenfindung Fahrbahn:

$$\frac{1}{2}(10n) + \frac{1}{2}(10n) = 5n + 5n = 10n \rightarrow \mathcal{O}(n) \text{ mit den Zeugen } C = 10 \text{ und } k = 1$$

- Iteration für Fahrbahnausfilterung und das Erzeugen der Sollspur:

$$n \cdot m \leq n \cdot n \text{ wenn } n \geq m \rightarrow \mathcal{O}(n^2) \text{ mit den Zeugen } C = 1 \text{ und } k = m$$

- Zusammengezogene Schätzung der Komplexität:

$$\begin{aligned} 2(n^2) + 10n &\leq 2n^2 + 10n^2 \text{ wenn } n \geq 1 \\ &= 12n^2 \\ &\rightarrow \mathcal{O}(n^2) \text{ mit den Zeugen } C = 12 \text{ und } k = m \end{aligned}$$

Formeln für die Berechnungen:

- Kantenfindung  $h = \text{Grenzwert für die Änderungsrate um eine Kante zu ermitteln und } z_1 \text{ der Wert des Pixels in der Quellmatrix } M: z = f(x, y) = \text{Graustufen des Pixels } 0 \leq z \leq 255$

$$\frac{\partial z}{\partial x} = \frac{f(x + \Delta x) - f(x)}{\Delta x} = \frac{f(x + 1) - f(x)}{1} = f(x + 1) - f(x)$$

$$\frac{\partial z}{\partial y} = \frac{f(y + \Delta x) - f(y)}{\Delta y} = \frac{f(y + 1) - f(y)}{1} = f(y + 1) - f(y)$$

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial z}{\partial x} \\ \frac{\partial z}{\partial y} \end{bmatrix}$$

$$\|\nabla f(x, y)\| = \sqrt{(f(x + 1) - f(x))^2 + (f(y + 1) - f(y))^2}$$

Wenn nun der  $\|\nabla f(x, y)\| \geq h$ , wird in die Zielmatrix  $M_1$  an die Stelle  $(x, y)$  der Wert 255 geschrieben, sonst 0.

- Finden der Fahrbahnreferenz am unteren Bildrand:

Wie zuvor beschrieben wird für die Fahrbahn von innen nach aussen gesucht. Dabei wird jeweils die erste Kante genommen, für die gilt:

$$\frac{\partial z}{\partial y} > h_y$$

wobei  $h_y$  für die minimale erwartete Änderung in  $y$ -Richtung steht. Im Anschluss werden die  $\min_{(x,y)}$ - und  $\max_{(x,y)}$ -Koordinaten der gefundenen Limits folgendermassen angeglichen:

$$\min_{(x,y)} = 10 \cdot \left\lfloor \frac{\min_{(x,y)}}{10} + 0.5 \right\rfloor$$

respektive:

$$\max_{(x,y)} = 10 \cdot \left\lceil \frac{\max_{(x,y)}}{10} + 0.5 \right\rceil$$

Die Fahrbahnkanten werden im Anschluss anhand der ermittelten Werte gesucht. Ist eine Kante in der Nähe der ermittelten Fahrbahn, wird sie behalten, ansonsten entfernt. Der Grenzwert muss noch genau ermittelt werden.

Die Parameter für die Regelung werden mit folgender Formel ermittelt:

$$G_R = K_p \left( 1 + \frac{1}{T_{ns} + T_{vs}} \right)$$

Danach werden die ermittelten Werte in die Korrekturformel eingesetzt:

$$y = K_{PR} \left( e + \frac{1}{T_n} \int edt + T_v \dot{e} \right)$$

wobei  $y$  dem zu korrigierenden Winkelwert in Grad entspricht. Die Parameter  $T_v$ ,  $T_n$  und  $K_{PR}$  müssen für die Feineinstellung justierbar sein. Dies soll für die Debugphase über eine auf dem Notebook betriebene Software geschehen.

### Testergebnisse

In den bisher durchgeföhrten Tests konnten die Bilder der Kamera mit einer Framerate von 10 - 20 Bilder auf dem Minicomputer verarbeitet werden. Die Lichteinflüsse haben sich dabei als handhabbar präsentiert, wenn die Beleuchtung mindestens der im Testgebäude vorhandenen Standardbeleuchtung entspricht. Hellere Lichtverhältnisse haben sich positiv auf die Kantenerkennung ausgewirkt.

Die Parallelisierung hat sich ebenfalls erfolgreich testen lassen, gemeinsam mit der Objekterkennung, allerdings nur auf dem Notebook aufgrund einer defekten Testkamera.

## Fahrbahnerkennung Unterstützung

### Funktionsbeschrieb

Wird die Fahrbahn auf einer Seite ganz verloren, speziell in engen Kurven, dann wird die Erkennung an den Flexsensor übergeben, der mit dem seitlichen Abstand zum Trottoir die Fahrspur halten soll.

### Komponentenbeschrieb



Abbildung 25: Flexsensor

Der Flexsensor ändert seinen Widerstandswert abhängig davon, wie stark der Sensor gebogen wird.

### Begründung

Für die Distanzmessung zum Trottoir wurden noch andere Sensoren (Ultraschall und Infrarot-Sensor) in Betracht gezogen. Jedoch sind die Ultraschall-Sensoren zu unpräzise und die Infrarot-Sensoren haben Mühe mit der schwarzen Oberfläche des Trottoirs. Beides wurde mit einem Funktionsmuster ausgetestet. Der Flexsensor hat bei beiden diesen Punkten bessere Eigenschaften.

### Testergebnisse

Diese Kennlinie stellt den Widerstandswert in Funktion von dem Biegewinkel dar.

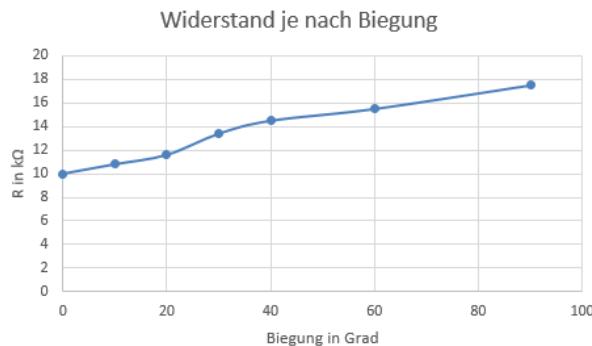


Abbildung 26: Widerstand in Funktion des Biegewinkels

Mit dieser Widerstandsänderung kann auf die Distanz geschlossen werden. Die Berechnung oder Auswertung welcher Widerstandswert zu welcher Distanz passt muss noch im PREN2 gemacht werden. Der Widerstandswert des Flexsensors kann über einen einfachen Spannungsteiler und einem AD-Wandler eingelesen werden. Ein Risiko besteht darin, dass der Flexsensor auf das Trottoir springt, anstatt sich zubiegen. Jedoch sollte dies durch eine optimale Positionierung vermieden werden können.

## 2.11 Objekterkennung

Die Objekterkennung hat das Ziel die richtigen Objekte zu erkennen und das genaue Positionieren des Fahrzeugs zu ermöglichen. Dabei wird die Containererkennung in zwei Teilaufgaben aufgeteilt: Die „Objekterkennung grob“ ist für die Erkennung der richtigen Objekte mittels Farb- und Formerkennung zuständig. Die „Objekterkennung präzise“ ist für das Positionieren des Fahrzeugs notwendig. Diese zwei Teilaufgaben werden separat angeschaut.

### 2.11.1 Objekterkennung grob

Die Aufgabe der „Objekterkennung grob“ ist es, die aufzuladenden Container, kreuzenden Fahrzeuge und das Entleerungsbecken zu erkennen. Sobald ein solches Objekt erkannt wird, wird der zentrale Controller darüber informiert, damit die Informationen an die „Objekterkennung“ präzise weitergegeben werden können.

#### Container und Entleerungsbecken

##### Funktionsbeschrieb

Bei der Container- und Entleerungsbeckenerkennung werden einerseits grüne und blaue Container und andererseits, im letzten Teil der Strecke, das Entleerungsbecken erkannt. Wird eines dieser Objekte erkannt, wird der Controller darüber informiert, damit sich das Fahrzeug auf das Auf- bzw. Abladen vorbereiten kann.

##### Komponentenbeschrieb

Sobald die ersten Bilder mit der Kamera aufgenommen wurden, werden diese laufend aus dem Bilderpool abgefragt und untersucht. Dabei läuft die Erkennung von grünen und blauen Containern parallel ab. Erst sobald die Kreuzung zum zweiten Mal passiert wurde, wird auch nach dem Entleerungsbecken gesucht.

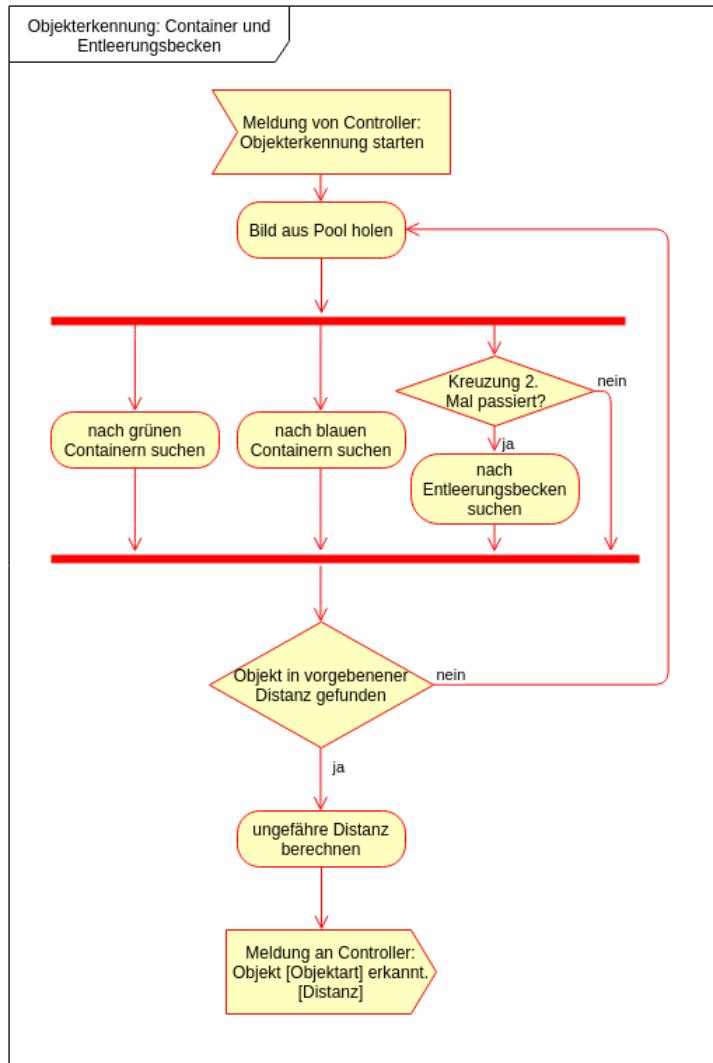


Abbildung 27: Aktivitätendiagramm Objekterkennung: Container und Entleerungsbecken

Die Erkennung läuft dabei immer gleich ab. Als erstes wird das Bild mit OpenCV nach der entsprechenden Farbe gefiltert. Sprich: Es wird nur nach Grün-, Blau- und Weisstönen gesucht. Anschliessend werden mit Hilfe einer Konturerkennung störende und falsche Objekte entfernt. Mit der Höhe dieser Objekte kann dann ausgerechnet werden, wie weit das Objekt noch entfernt ist. Diese Distanzberechnung liefert nur ein ungefähres Ergebnis. Ist das Objekt zu weit entfernt, wird der zentrale Controller nicht informiert. Ansonsten wird eine Meldung an den Controller gesendet, welche die Objektart und Distanz enthält.

### Begründung

Da der Blickwinkel auf die Container durch die Kurven variieren kann, wäre es wenig sinnvoll die Konturerkennung vor der Farberkennung durchzuführen. Es bestehen zu viele Möglichkeiten, dass ein störendes Objekt als Container oder Entleerungsbecken identifiziert werden könnte. Dadurch, dass zuerst nach Farben gefiltert wird, können bereits viele falsche Objekte ausgeschlossen werden.

OpenCV stellt Methoden zur Verfügung, mit welcher ein ganzer Bereich mit Farbwerten im HSV-Format angegeben werden kann. Das bedeutet, dass auch bei unterschiedlichen Lichtverhältnissen die Farbtöne korrekt erfasst werden können.

### Testergebnisse

Die Containererkennung wurde bereits soweit getestet, dass grüne und blaue Container bis auf eine bewusst eingeschränkte Distanz erkannt werden können.

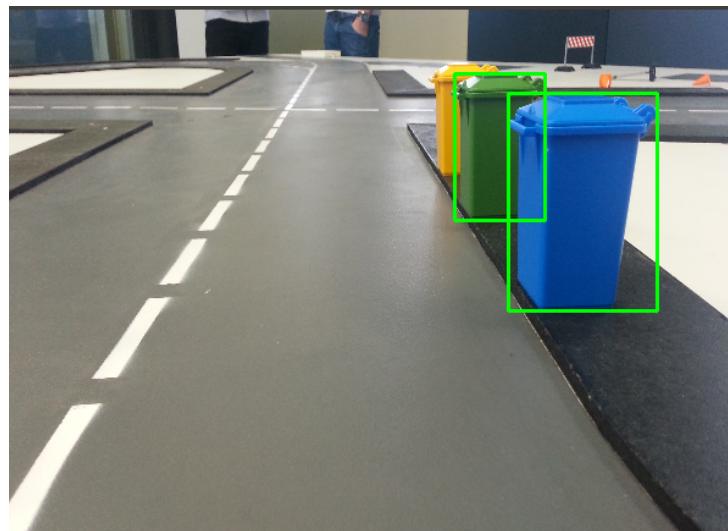


Abbildung 28: erfolgreicher Testversuch

Auch bei aufgenommenen Bildern, welche gleichfarbige Container mit unterschiedlichen Lichtverhältnissen zeigen, konnten die Container korrekt erkannt werden:

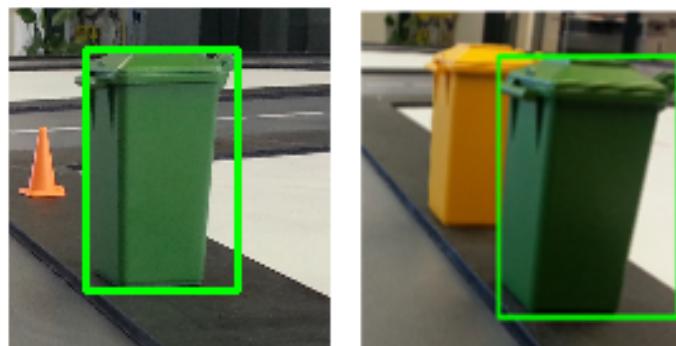


Abbildung 29: unterschiedliche Helligkeiten

Bei den Testversuchen wurden jedoch auch Probleme und Risiken aufgedeckt. Falls zum Beispiel im Hintergrund ein Objekt steht, welches eine grüne oder blaue Farbe hat und dessen Masse einem Container entsprechen könnten, kann es zu einer Fehlererkennung kommen. Folgendes Bild zeigt einen solchen Fall:

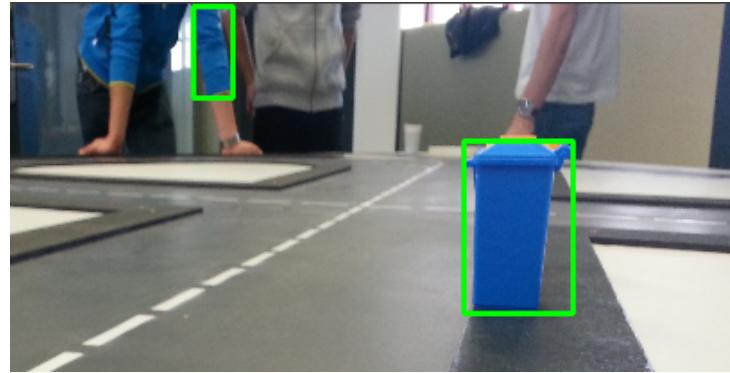


Abbildung 30: fehlerhafter Testversuch

Um solche Fälle möglichst zu vermeiden, wird versucht den abzusuchenden Bereich einzuschränken, so dass nur solche Bildbereiche untersucht werden, wo sich auch wirklich ein Container befinden könnte.

### Rechtsvortritt

#### Funktionsbeschrieb

Falls sich an der Kreuzung ein gegnerisches Fahrzeug von rechts nähert, hat dieses gemäss Aufgabenstellung Vortritt. Die Objekterkennung muss ein solches Fahrzeug erkennen und den Controller darüber informieren, damit das Fahrzeug gestoppt wird.

#### Komponentenbeschrieb

Sobald die Objekterkennung vom Controller informiert wird, dass das Fahrzeug gleich an der Kreuzung ankommen wird, wird Ausschau nach einem gegnerischen Fahrzeug auf der von rechts kommenden Spur gehalten. Wird ein Fahrzeug erkannt, wird dies dem Controller gemeldet. Der Vorgang wird wiederholt, bis das Fahrzeug nicht mehr im Bereich liegt. Nachdem dies geschehen ist, wird wieder der Controller informiert, damit das Fahrzeug weiterfahren kann.

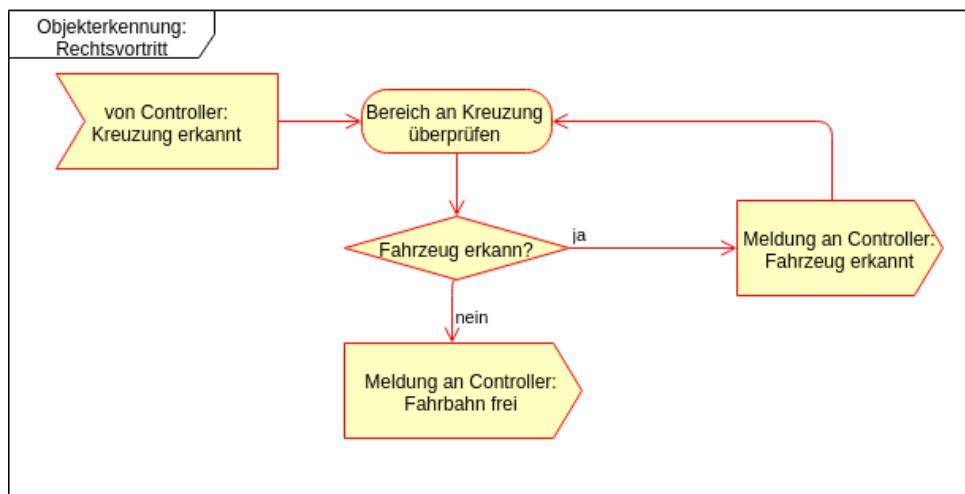


Abbildung 31: Aktivitätendiagramm Objekterkennung: Rechtsvortritt

Da nicht nur angehalten werden muss, wenn ein gegenerisches Fahrzeug an der Kreuzung steht, sondern auch, wenn es sich von rechts nähert, muss der zu überprüfende Bereich eine gewisse Größe haben. Der ungefähre Bereich, welcher abgesucht werden soll, ist in der folgenden Grafik markiert:

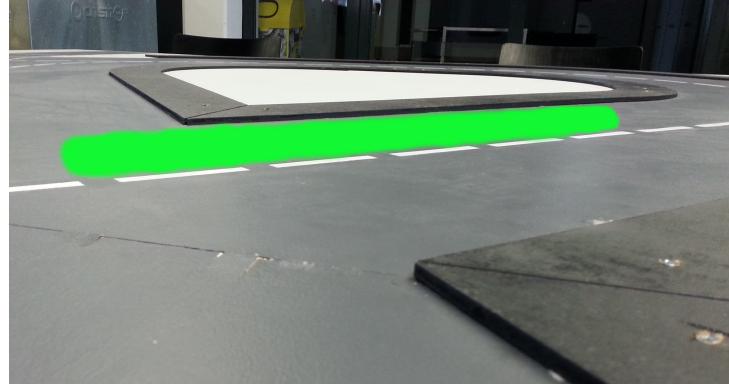


Abbildung 32: Abgedeckter Bereich beim Rechtsvortritt

### Begründung

Bei der Vortrittserkennung kann nicht nach Farben oder Konturen gesucht werden, da keine Angaben vorhanden sind, wie das gegnerische Fahrzeug aussieht. Daher ist es nötig, dass erkannt werden kann, ob ein definierter Straßenabschnitt verdeckt ist oder nicht. Falls der Abschnitt verdeckt ist, heißt das, dass sich ein Fahrzeug darauf befindet, welches Vortritt hat.

#### 2.11.2 Objekterkennung präzise

##### Containererkennung

###### Funktionsbeschrieb

Die präzise Containererkennung wird für das genaue Stoppen des Fahrzeugs gebraucht. Dieses Modul gibt den Trigger zum Anhalten.

###### Komponentenbeschrieb

Die Containererkennung wird mit Hilfe von Distanzsensoren realisiert. Dafür sind Infrarot-Distanzmesser oder Infrarot-Reflexkoppler vorgesehen. Diese werden auf der rechten Seite des Fahrzeugs befestigt. Sobald ein Objekt in der Nähe ist, ändert sich der Spannungswert, was einer Distanzänderung gleichkommt.

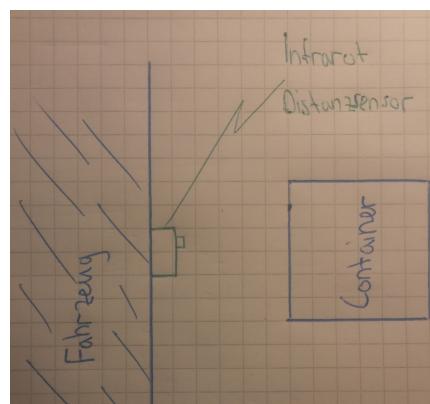


Abbildung 33: Beispielhafte Containererkennung mit einem Infrarotsensor

### Begründung

Die Positionierung mit einem Infrarotsensor ist die ideale Lösung. Sie ist einfach zu realisieren im Vergleich zu einer Kamera oder einem Farbesensor und präziser als ein Ultraschallsensor.

## Testergebnisse

Für die Ermittlung des besten Sensors wurde ein Ultraschall und Infrarotsensor als Funktionsmuster getestet. Der relativ grosse „Empfangswinkel“, welcher der Ultraschallsensor aufweist, ist in dieser Anwendung nicht gewollt. Der Infrarotsensor ist diesbezüglich besser. Dieser ist zwar lichtempfindlich, jedoch sollte dieses Problem gelöst werden können.

## Rechtsvortritt

### Funktionsbeschrieb

Die „Rechtsvortritterkennung detailliert“ hat die Aufgabe das Fahrzeug vor einer Kollision mit einem zweiten Fahrzeug zu bewahren. Wenn rechts oder vor dem Fahrzeug etwas im Weg ist, soll das Fahrzeug anhalten.

### Komponentenbeschrieb



Abbildung 34: Getesteter Ultraschallsensor (Quelle: [www.generationrobots.com](http://www.generationrobots.com))

Diese Aufgabe wird mit einem Ultraschallsensor gelöst.

## Begründung

Ein Ultraschallsensor ist für unter fünf Franken zu haben und erfüllt mit seinem grossen Abstrahlwinkel und seiner Reichweite die geforderten Bedingungen. Zudem ist die Anbindung an das Freedomboard gut möglich.

## Testergebnisse

Der Ultraschallsensor wurde bereits erfolgreich an das Freedomboard angeschlossen und ausgewertet. Die Anbindung wurde mit Hilfe einer Anleitung von [mcuoneclipse.com](http://mcuoneclipse.com) realisiert. Die Distanzauswertung funktionierte bei dem Test relativ genau ( $\approx \pm 1\text{cm}$ ).

## Entleerungbecken Erkennung

### Funktionsbeschrieb

Damit das Schüttgut im Entleerungsbecken entleert werden kann, muss das Entleerungsbecken richtig detektiert werden können.

### Komponentenbeschrieb

Die Container und Fahrbahnerkennung benötigen bereits Sensoren. Dies wären der Flexsensor und der Infrarotsensor. Für die Detektion des Entleerungsbehälters ist kein weiterer Sensor mehr nötig. Die Detektion ist mit beiden Sensoren möglich. Bei Tests mit den beiden Sensoren wird sich zeigen, welcher dieser beiden besser geeignet ist.

## Begründung

Die Einsparung von Sensoren entlastet das Budget und spart zusätzlichen Implementierungsaufwand.

### 3 Projektmanagement

Bei einem Projekt, wie es in den Modulen PREN1 und PREN2 durchgeführt wird, macht es Sinn, dass Entscheidungen zum Thema Projektmanagement getroffen werden. Wichtig ist, dass unter den Teammitgliedern Verantwortungen verteilt werden, mögliche Risiken erkannt und Massnahmen dazu gefunden werden und die Ressourcen Zeit und Geld genau eingeplant werden.

#### 3.1 Team

##### Maschinenbau



**Stefan Häfliger**  
Beladen und Entladen



**Joël Meloni**  
Chassis und Lenkung

##### Elektrotechnik



**Silvan Ritz**  
Servos, Sensoren und Freescale-Board

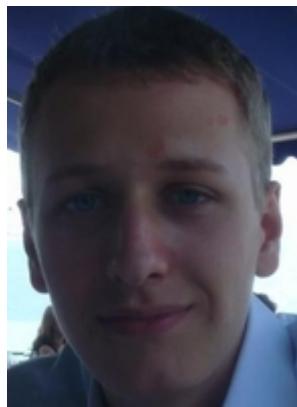


**Lars Walther**  
Energieversorgung und Antrieb

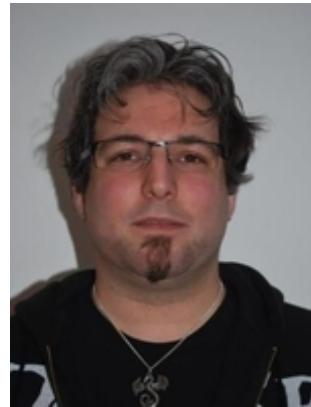
## Informatik



**Patrizio Brantschen**  
Objekterkennung und Rechtsvortritt



**Adrian Würsch**  
Schnittstellen und Dokumentation



**Tobias Kreienbühl**  
Fahrbahnerkennung und Mini-Computer

## 3.2 Zeitplanung

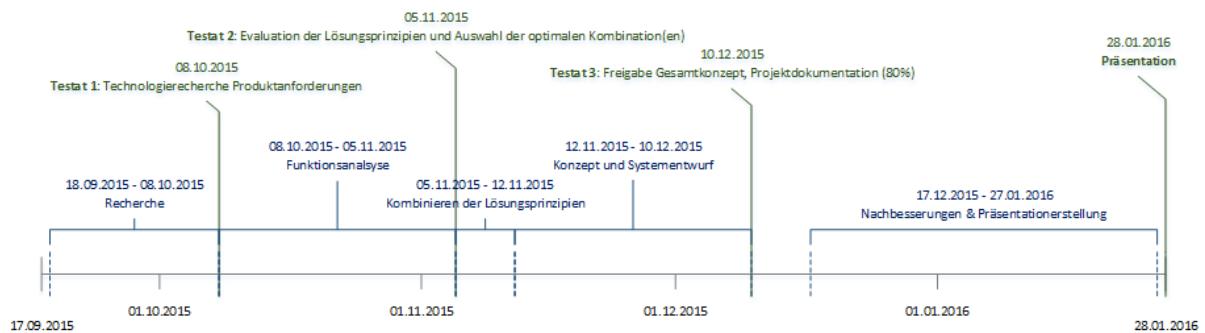


Abbildung 35: Zeitplan

## 3.3 Risikomanagement

Die Risiken, die während der Projektdurchführung entstehen könnten, sind im Folgenden aufgelistet. Um zu wissen, wie hoch ein Risiko eingeschätzt werden muss, wurde eine Grafik erstellt. Zu jedem Risiko sind auch Massnahmen aufgelistet, welche im Eintrittsfall unternommen werden können.

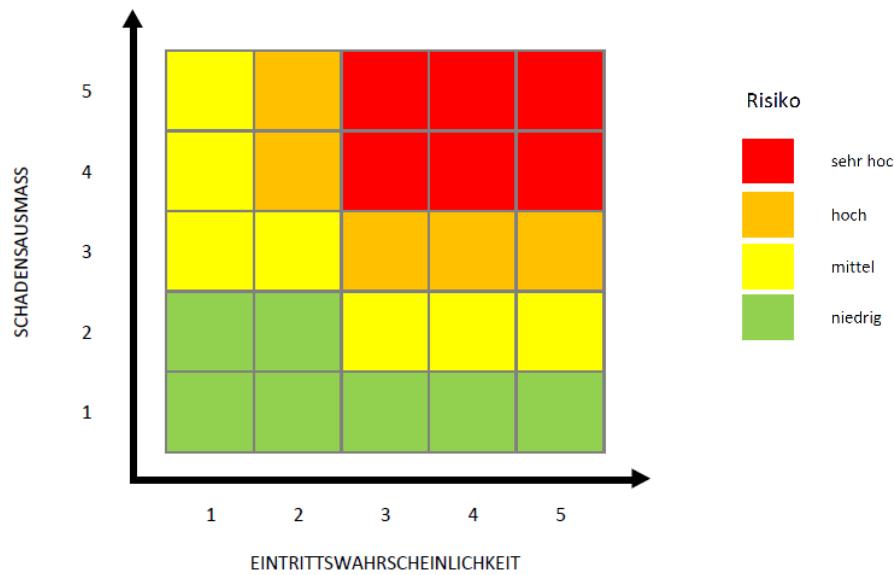


Abbildung 36: Risikomatrix

### 3.3.1 Risiken im Team

Risiko	Wahrscheinlichkeit	Schadensausmass	Massnahmen
Mitglied verlässt das Team	1	4	Neuverteilung der Arbeiten
Unzuverlässigkeit eines Teammitglieds	2	3-4	Absprache im Team, um eine Lösung zu finden
Unstimmigkeiten unter den Teammitgliedern	3	4	Absprache im Team, um eine Lösung zu finden, evtl. auch Rat von Supervisor einholen.
Mitglied überlastet	4	3	Neuverteilung der Arbeiten
Kommunikationsprobleme	4	3	verbesserte Absprachen, Frequenz der Sitzungen erhöhen
Fehlende fachliche Kompetenz eines Mitglieds	2	2	Neuverteilung der Arbeiten
Fehlende soziale Kompetenz eines Mitglieds	1	3	Absprache im Team, um eine Lösung zu finden

Tabelle 1: Risiken im Team

### 3.3.2 Risiken im Projektmanagement

Risiko	Wahrscheinlichkeit	Schadensausmass	Massnahmen
Missverständnisse bei den Anforderungen	3-4	3	Absprache mit Supervisor
Ineffizientes Arbeiten	3-4	2-3	Neuorganisierung der Vorgehensweise
Überschreitung des Budgets	1	4	Absprache mit Supervisor

Tabelle 2: Risiken im Projektmanagement

### 3.3.3 Risiken bei der Realisierung

Risiko	Wahrscheinlichkeit	Schadensausmass	Massnahmen
Idee funktioniert nicht wie gewünscht	2	4	neue Ideenfindung, Rat einholen
Eingekaufte Komponente erfüllt Anforderung nicht	2	5	andere Komponente einkaufen
Projektanforderungen sind nicht erfüllt	1	5	Anpassungen planen

Tabelle 3: Risiken bei der Realisierung

### 3.3.4 Andere Risiken

Risiko	Wahrscheinlichkeit	Schadensausmass	Massnahmen
Lange Lieferzeiten bei Bestellungen	3	2	andere Arbeiten aufnehmen, früher bestellen
Github fällt aus	1	4	Lokal weiterarbeiten

Tabelle 4: andere Risiken

### 3.4 Budgetplanung

Gemäss Anforderung steht für das gesamte Projekt 500 Fr. zur Verfügung, wobei 200 Fr. bereits in PREN1 verwendet werden können.

Bauteil	Beschreibung	ungefähre Kosten (Fr.)	Kosten PREN1 (Fr.)
<b>el. Bauteile</b>			
Raspberry Pi 2		45.00	45.00
Kamera		33.00	33.00
Kameragehäuse		10.00	0.00
Freedom-Board		20.00	20.00
Sensoren	Je nach Einsatz (Ultraschallsensor, Infrarotsensor, Flexsensor)	70.00	20.00
Akkus	2 Stück	80.00	0.00
Antriebs-Motor	DC-Motor	35.00	0.00
Servo-Motoren	4 Stück	55.00	10.00
Motor für Klappe	DC-Motor	15.00	0.00
Encoder		15.00	0.00
Spannungsregler		15.00	0.00
Motorentreiber		10.00	0.00
<b>mech. Bauteile</b>			
Differentialgetriebe	inkl. Gehäuse und Kegelräder	30.00	30.00
div. mech. Komponenten	Zahnräder, Lagerbüchsen etc.	40.00	0.00
Lenkung	3D-Drucker	0.00	0.00
Räder	Sponsoring	0.00	0.00
Greifer	Fräsen	0.00	0.00
Behälter	Plexiglas, Lasern	0.00	0.00
Chassis		0.00	0.00
<b>Total</b>		<b>473.00</b>	<b>158.00</b>

Tabelle 5: Budgetplanung

### 3.5 Projektunterstützung

Während der Durchführung des Projekts soll der administrative Aufwand möglichst klein gehalten werden können. Weiter sollen alle Projektmitglieder gleichzeitig an der Software weiterentwickeln oder an der Dokumentation weiterschreiben können. Darum wird auf diverse Software-Produkte und -Systeme zurückgegriffen, die sich in der Praxis bewährt haben.

#### 3.5.1 Software

Kommunikation	E-Mail, Whatsapp
Dokumenterstellung	LaTeX
Versionsverwaltungssoftware	Git
Filehosting-Dienst	GitHub
CAD-Software	NX
Entwicklungsumgebung	Eclipse, Kinetis Design Studio

Tabelle 6: Software zur Projektunterstützung

LaTeX ermöglicht es, die gesamte Dokumentation in mehrere Teildokumente zu unterteilen. All diese Dokumente sind in GitHub eingeccheckt. Somit wird sichergestellt, dass mehrere Teammitglieder zur selben Zeit an der Dokumentation arbeiten können und dabei möglichst wenig Konflikte entstehen. Weiter wird

auf dem selben Git-Repository auch der Quellcode der Software abgelegt.

### 3.5.2 Aufgabenverteilung

Um die Aufgabenverteilung festzuhalten wurde ein Scrum-Board erstellt. Darauf sind alle Team-Mitglieder aufgelistet. Zudem sind, wie bei Scrum üblich, die drei Zustände „ToDo“, „Doing“ und „Done“ definiert.

	ToDo	Doing	Done
Adrian	Verantwortung Anhang		
Lars	Berechnung Alkoholeistung	Verantwortung Lösungskonzept	
Joel	Verantwortung Management Summary		
Stefan	Berechnung Brummen Servos	Verantwortung Lösungskonzept	
Silvan	Regelkreis automatisierung (speed)	Berechnung Endkugel + Ultrahall Zielserzung	Funktionsmuster Distanzsenoren
Tobias	Berechnung Farbdiff Pixel	Verantwortung Schlussdiskussion	Regelkreis aufrechterhalten (Steigung)
Patrizio	Verantwortung Projektmanagement	Kosten - abschätzung	Projektplan
Alle	Habitate 3D	Allgemein	Alle Quellen

Abbildung 37: Scrum-Board

## 4 Schlussdiskussion

### 4.1 Lessons Learned

Bei einer Projektarbeit mit interdisziplinärer Zusammenarbeit ist es wichtig, sich früh auf einen Konsens zu einigen. Für das Team war es wichtig, sich immer gemeinsam zu entscheiden und so immer alle möglichen Auswirkungen im Blick zu haben. Durch das Definieren von Schnittstellen zu Beginn, konnten viele Missverständnisse schon in der Planungsphase bereinigt werden.

Gewisse OpenSource Lösungen bieten zwar eine breite und solide Palette von Funktionen an, sind aber schlecht auf eine spezifische Aufgabe skalierbar. Ein interessiertes Mitglied der Gruppe hat sich die Zeit genommen und mit vertieftem mathematischen Wissen einen sehr effizienten Algorithmus entwickelt. Dieser schlanke, auf die Aufgabe zugeschnittene, Algorithmus ermöglicht das Auswerten des Bildmaterials nahezu in Echtzeit.

Einige Mitglieder des Teams haben ein persönliches Engagement für Lego Bauteile. Dies machte sich nützlich, um ein erstes Fahrzeug zu visualisieren. Mit diesem provisorischen Lego-Fahrzeug konnte die Strecke abgefahren werden. So wurden einige Problemstellen aufgedeckt, welche vorher nicht beachtet wurden. Durch diese simple Idee konnte die zukünftige Umsetzung massiv beeinflusst werden.

Zu Beginn setzte sich das Team zusammen und diskutierte über alle benötigten Funktionen und Bauteile. Dies ermöglichte den Maschinenbauern ein Fahrzeug zu modellieren, welches genug Platz bietet und innerhalb der Größenordnung bleibt. Durch dieses Vorgehen ist frühzeitig sichergestellt, dass alle Funktionalitäten abgedeckt sind und zeitig Fertigungsteile bestellt werden können.

Um eine Gruppe zu bilden und erfolgreich zu führen braucht es zwingend zwei Dinge: Die gleichen Ziele und die gleichen Regeln für alle Mitglieder. Doch wo es Regeln gibt, wird es auch Verstöße geben. Deshalb hat sich das Team früh auf eine adäquate Strafe geeinigt. Kommt jemand zu spät oder gar nicht, so soll derjenige das Team mit Getränken versorgen. So ist sichergestellt, dass genügend Motivation vorhanden ist und auch die anderen bei solch einem Missgeschick profitieren.

### 4.2 Ausblick PREN2

Ende PREN1 wurde ein Lösungskonzept ausgewählt und detailliert ausgearbeitet. Die Aufgaben sind klar verteilt und das Team versteht sich gut. PREN2 geht nun nahtlos weiter und setzt die Planung in die Realität um. Besonders ins Auge sticht die komplexe Navigation auf dem Spielfeld und das Zusammenspiel der vielen Sensoren. Das Team freut sich auf die neuen Herausforderungen, die umso grösser sind, da ein Mitglied die Gruppe verlässt.

### 4.3 Fazit

Ziel von PREN1 ist es ein Funktionskonzept zu einer Aufgabenstellung in Teamarbeit zu erstellen. Am Ende von PREN1 hat die Gruppe nun ein plausibles und umsetzbares Konzept ausgearbeitet und ist selbst überzeugt davon. Es lässt sich sagen, dass die Projektplanung ein Erfolg ist. Besonders die variable Erkennung von Fahrbahn und Containern bereitete Sorgen. Deshalb wurde bereits in der Planung viel Zeit für die Problemlösung eingesetzt. Nun ist das Team zuversichtlich die Aufgabe korrekt und zeitig abzuschliessen und im Wettbewerb einen Platz in den vorderen Rängen zu belegen. Die Mitglieder arbeiten eingespielt und ecken nur selten an. So lässt sich das Projekt auch bezüglich Sozialkompetenz als erfolgreich bezeichnen.

## Abbildungsverzeichnis

1	Übersichtszeichnung . . . . .	4
2	Schnittstellenübersicht . . . . .	5
3	Grober Antriebsregelkreis . . . . .	6
4	Grober Steuerungs Regelkreis . . . . .	6
5	Konzept des Chassis . . . . .	7
6	Konzept des Chassis mit Komponentenbeschriftung . . . . .	7
7	Konzept der Achsschenkellenkung . . . . .	8
8	Antrieb (Quelle: <a href="http://www.conrad.ch">http://www.conrad.ch</a> ) . . . . .	9
9	Greifer . . . . .	10
10	Einladeprozess . . . . .	10
11	Schüttgutfluss in den beiden Behältern . . . . .	11
12	Entladen . . . . .	12
13	Klappe schliessen . . . . .	13
14	Aufteilung der Akkumulatoren . . . . .	14
15	Aktivitätendiagramm Grobalauf . . . . .	16
16	Raspberry Pi2 Model B (Quelle: <a href="http://pcworld.com/">http://pcworld.com/</a> ) . . . . .	17
17	Freedomboard KL25 von Freescale (Quelle: <a href="http://ch.farnell.com/">http://ch.farnell.com/</a> ) . . . . .	18
18	Übersicht MC Tasks . . . . .	18
19	Raspberry Pi Kameramodul . . . . .	19
20	Bild nach Kantenerkennung . . . . .	20
21	Aktivitätendiagramm Fahrbahnerkennung . . . . .	21
22	Graph der Graustufenwerte einer Bildzeile in <i>x</i> -Richtung . . . . .	22
23	Erste Kante links der Mitte . . . . .	22
24	Erste Kante rechts der Mitte . . . . .	23
25	Flexsensor . . . . .	25
26	Widerstand in Funktion des Biegewinkels . . . . .	25
27	Aktivitätendiagramm Objekterkennung: Container und Entleerungsbecken . . . . .	26
28	erfolgreicher Testversuch . . . . .	27
29	unterschiedliche Helligkeiten . . . . .	27
30	fehlerhafter Testversuch . . . . .	28
31	Aktivitätendiagramm Objekterkennung: Rechtsvortritt . . . . .	28
32	Abgedeckter Bereich beim Rechtsvortritt . . . . .	29
33	Beispielhafte Containererkennung mit einem Infrarotsensor . . . . .	29
34	Getesteter Ultraschallsensor (Quelle: <a href="http://www.generationrobots.com">www.generationrobots.com</a> ) . . . . .	30
35	Zeitplan . . . . .	32
36	Risikomatrix . . . . .	33
37	Scrum-Board . . . . .	36

## Tabellenverzeichnis

1	Risiken im Team . . . . .	33
2	Risiken im Projektmanagement . . . . .	34
3	Risiken bei der Realisierung . . . . .	34
4	andere Risiken . . . . .	34
5	Budgetplanung . . . . .	35
6	Software zur Projektunterstützung . . . . .	35

## **Anhang**

- I** Technologierecherche (26 Seiten)
- II** Aufgabenstellung (8 Seiten)
- III** Anforderungsliste (5 Seiten)
- IV** Morphologischer Kasten (44 Seiten)

# GüselStar XXI

## Technologierecherche

---

*Autoren:*

Patrizio Brantschen  
Stefan Häfliger  
Tobias Kreienbühl  
Joël Meloni  
Silvan Ritz  
Lars Walther  
Adrian Würsch

*Team-Coach:*  
Jürg Habegger

## Inhaltsverzeichnis

<b>1 Autonomes Fahren</b>	<b>3</b>
1.1 Erkennung des rechten Randes mit Sensoren . . . . .	3
1.2 Fest einprogrammierte Strecke ohne Sensoren . . . . .	4
1.3 Rechtsvortritt . . . . .	4
1.3.1 Mit der Kamera . . . . .	4
1.3.2 Mit Sensoren . . . . .	4
<b>2 Beladen</b>	<b>4</b>
2.1 Erkennung des Containers . . . . .	4
2.2 Mechanismen die bei Müllfahrzeugen eingesetzt werden . . . . .	5
<b>3 Entladen</b>	<b>6</b>
3.1 Kippen des Behälters . . . . .	6
3.2 Seitliche Klappe öffnen . . . . .	6
3.3 Mit Greifer ausladen . . . . .	6
<b>4 Greifer</b>	<b>7</b>
4.1 mechanisch . . . . .	7
4.2 pneumatisch . . . . .	8
4.3 magnetisch . . . . .	8
<b>5 Antrieb</b>	<b>8</b>
5.1 Elektromotor . . . . .	8
5.1.1 Bürstenloser Gleichstrommotor . . . . .	9
5.1.2 Gleichstrommaschine . . . . .	9
5.1.3 Schrittmotor . . . . .	9
5.2 Verbrennungsmotor . . . . .	10
5.3 Pneumatik . . . . .	10
5.4 Dampfmaschine . . . . .	10
<b>6 Lenkung</b>	<b>10</b>
6.1 Knicklenkung . . . . .	10
6.2 Achsschenkellenkung . . . . .	11
6.2.1 Lenktrapez . . . . .	11
6.3 Zweiradlenkung . . . . .	12
6.4 Lenkung für Kettenfahrzeuge . . . . .	12
<b>7 Mikrocontroller-Board</b>	<b>12</b>
7.1 Anforderungen . . . . .	12
7.2 Tinkerforge . . . . .	12
7.3 HCS08 Devlopperboard . . . . .	13
7.4 Arduinoboard . . . . .	13
7.5 Freedomboard . . . . .	13
<b>8 Distanzsensoren</b>	<b>13</b>
8.1 Anforderungen . . . . .	13
8.2 Ultraschallsensoren . . . . .	14
8.3 Infrarotsensoren . . . . .	14
<b>9 Farbsensoren</b>	<b>14</b>
9.1 Anforderungen . . . . .	14
9.2 Eigenschaften . . . . .	14

<b>10 Stromversorgung</b>	<b>15</b>
10.1 Anforderungen . . . . .	15
10.2 Batterien . . . . .	15
10.3 NiCd (Nickel-Cadmium) . . . . .	15
10.4 Lithium-Akku (LiPo, Lilon) . . . . .	16
10.5 NiMH (Nickel-Metallhydrid) . . . . .	16
<b>11 Boardcomputer</b>	<b>16</b>
11.1 Raspberry Pi . . . . .	16
11.2 Raspberry Pi 2 . . . . .	17
11.3 Banana Pi . . . . .	17
11.4 BeagleBone Black . . . . .	17
11.5 ODROID-XU4 . . . . .	18
<b>12 Bilderkennung</b>	<b>18</b>
12.1 Anforderungen . . . . .	18
12.2 OpenCV . . . . .	18
12.3 MatLab . . . . .	18
12.4 SimpleCV . . . . .	19
12.5 LibCCV . . . . .	19
<b>13 Kamerasysteme</b>	<b>19</b>
13.1 Raspberry Pi CAM . . . . .	19
13.2 Logitech Webcam C525 . . . . .	20
13.3 Logitech Webcam C615 . . . . .	20
13.4 Logitech Webcam C920 . . . . .	20
13.5 Zusammenfassung . . . . .	21
13.6 Vor- und Nachteile . . . . .	21
13.6.1 Raspberry Pi CAM . . . . .	21
13.6.2 Webcams . . . . .	21
13.7 Steuerung . . . . .	21
13.8 Ein Steuerungsboard . . . . .	22
13.9 Bordcomputer und Hardwarecontroller . . . . .	22
<b>14 Morphologischer Kasten</b>	<b>22</b>
<b>15 Recherchequellen</b>	<b>23</b>

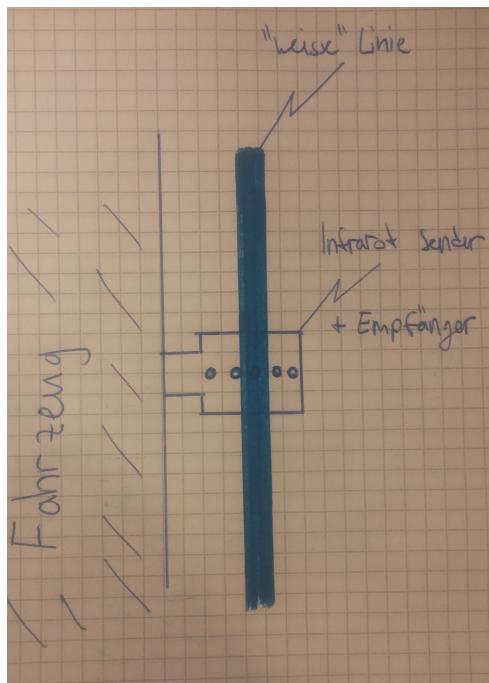
# 1 Autonomes Fahren

## 1.1 Erkennung des rechten Randes mit Sensoren

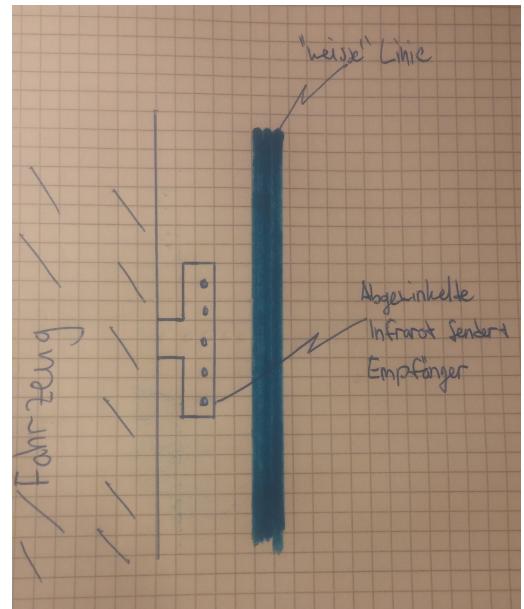
Der rechte Rand des Parcours ist für die Spurerkennung vielversprechend. Er ist entweder durch eine weisse Linie, ein 5mm hohes Trottoir oder für kurze Zeit auf der Kreuzung nicht begrenzt. Für die unterschiedlichen Bedingungen werden verschiedene Möglichkeiten aufgezeigt.

### Erkennung der weissen Linie

Die rechte Begrenzungslinie ist 1cm breit und weiß. Es gibt viele Minaturfahrzeuge die einer Linie folgen. Jedoch ist bei den meisten Linienfolger die zu folgende Linie in der Mitte des Fahrzeuges. Das hat der Vorteil das man links und rechts der Linie die Linie detektieren kann. Dies ist bei dieser Aufgabenstellung nicht oder nur sehr begrenzt möglich. Dies ist für beide Lösungsvarianten eine Herausforderung. Für beide wäre ein Funktionsmuster nötig. Hier zwei Möglichkeiten:



(a) Variante 1. Sensoren über der Linie



(b) Variante 2. Sensoren neben der Linie

Abbildung 1: Mögliche Linienerkennungen mit Infrarotsensoren

**Erkennung des Trottoirs** Das Trottoir ist die längste und wichtigste (rechte) Begrenzung des Tracks. Es ist der längste Straßenabschnitt und die Positionsbestimmung muss am präzisesten funktionieren. Für das die Trottoir Erkennung sind diese zwei Varianten im Vergleich.

### Möglichkeiten bei keiner Begrenzung (Kreuzung)

Die Kreuzung ist der Spezialfall des Tracks. Es gibt keine Begrenzung am rechten Rand. Da keine Begrenzung vorhanden ist, kann man mit Sensoren auch nicht detektieren. Die Lösung ohne Sensoren ist bei der Kreuzung stur geradeaus zu fahren. Dies könnte man einprogrammieren, wenn weder eine Linie noch ein Trottoir vorhanden ist. Dies setzt natürlich die richtige Ausrichtung des Fahrzeugs vor der Kreuzung voraus.

## 1.2 Fest einprogrammierte Strecke ohne Sensoren

Zu der Variante mit vielen Sensoren gibt es die Möglichkeit die Strecke fest einzuprogrammieren. Da die Strecke immer gleich bleibt (ausser den zwei Möglichen Startplätze) ist es möglich die Strecke bereits fest einzuprogrammieren. **Vorteile**

- Keine Sensoren für die Spurerkennung notwendig.
- Kostenersparnis
- Weniger Fehlerquellen

### Nachteile

- Keine Fehlerkorrektur (einmal falsch, immer falsch)
- Braucht extrem genauer Antrieb + Lenkung

## 1.3 Rechtsvortritt

Damit das Fahrzeug verkehrssicher unterwegs ist, muss ein von rechts kommendes Fahrzeug erkannt werden.

### 1.3.1 Mit der Kamera

Falls eine Kamera für die Spurerkennung eingesetzt würde, wäre auch die Erkennung von rechts kommenden Fahrzeugen denkbar. Somit müsste die Kamera auch den rechten Teil der Spur aufnehmen.

### 1.3.2 Mit Sensoren

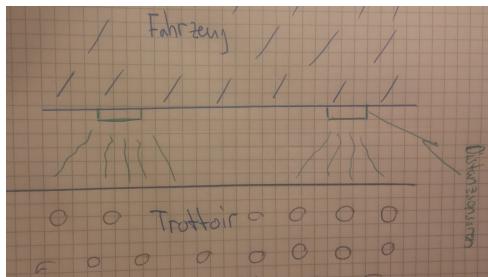
Ein von rechts kommendes Fahrzeug könnte mit Sensoren erkannt werden. Diese Sensoren müssten an der rechten Seite des Fahrzeugs befestigt werden. Es müsste möglich sein von rechts kommende Fahrzeuge von Fußgängern und Hindernissen zu unterscheiden.

## 2 Beladen

### 2.1 Erkennung des Containers

#### Mit Distanzsensoren und Farbsensoren

- Unbekannte Genauigkeit muss getestet werden.
- Benötigt AD Wandler (Wenn Infrarot- oder Ultraschallsensor)
- Kein mechanischer Kontakt



(a) Variante 1. Distanzsensoren als Abstands-messung



(b) Variante 2. Distanzhalter als mechanische Bregrenzung (z.B Räder)

Abbildung 2: Mögliche Spurhalte Konzepte beim Trottoir

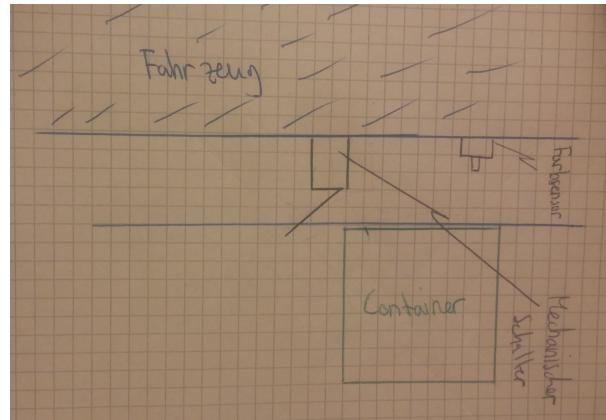


Abbildung 3: Distanzsensoren an der rechten Seite des Fahrzeugs

### Mechanische Detektion und Farbsensoren

- Unbekannte Genauigkeit muss getestet werden.
- Klarer Wert
- Mechanischer Kontakt (darf nichts ausser den Container berühren)

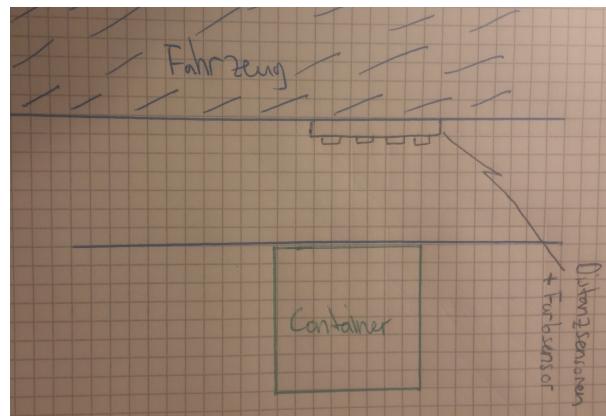


Abbildung 4: Ein Schalter an der rechten Seite des Fahrzeugs

## 2.2 Mechanismen die bei Müllfahrzeugen eingesetzt werden

- Der Greifer kann senkrecht zur Fahrbahn ausgefahren werden. Zudem ist der Greifer fähig eine Rotation um ca. 135° durchzuführen.  
Der genaue Mechanismus ist auf youtube (<https://www.youtube.com/watch?v=LTUjiLxzDQs>) von 0.08 bis 0.23 ersichtlich.
- Auf dem selben Video (bei 4:33-4:57) ist ein weiterer Mechanismus zu sehen, der mit einem Knickarm funktioniert.
- Bei 5:20-5:50 ist ein sehr einfache und direkte Lösung zu sehen. Es dürfte schwierig sein, diesen Vorgang autonom zu realisieren.

### 3 Entladen

#### 3.1 Kippen des Behälters



Seitliches Entladen des Behälters

Vorteile:

- einfache Realisierung
- wird in der Praxis angewandt

Nachteile:

- Nahes Heranfahren an Entladebehälter notwendig

#### 3.2 Seitliche Klappe öffnen

Vorteil:

- sehr einfach realisierbar

Nachteile:

- nicht automatisch wieder verschliessbar
- fraglich ob Zielbereich einhaltbar ist

#### 3.3 Mit Greifer ausladen

Vorteil:

- Greifer wird für 2 Funktion gebraucht (Einladen/Ausladen)

Nachteile:

- schwierig gesamtes Schüttgut zu erwischen
- aufwändiger Greifer
- schwierig realisierbar

## 4 Greifer

### 4.1 mechanisch

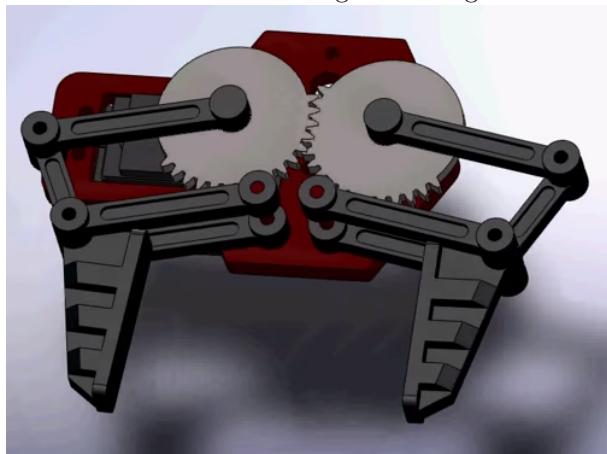
Greifer besteht aus einer festen Seite und einer beweglichen Seite



Eigenschaften:

- sehr einfacher Aufbau
- wenige Teile notwendig
- nicht präzise Lösung

Greifer mittels 2 Zahnräder gleichmässig schliessen



Eigenschaften:

- präzise Lösung
- aufwändiger Aufbau
- symmetrischer Aufbau

## 4.2 pneumatisch

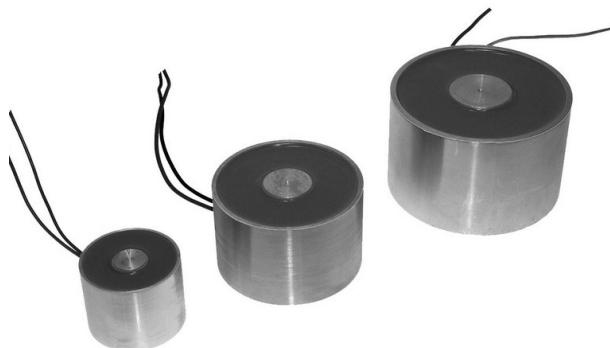


Eigenschaften:

- Druckluft-Aggregat notwendig
- hohe Präzision
- teuer

## 4.3 magnetisch

Greifen der M4 Schrauben und Muttern mittels eines Elektromagneten



Eigenschaften:

- einfacher Aufbau
- einfache Integration
- fraglich ob genügend stark

## 5 Antrieb

### 5.1 Elektromotor

- Vorwissen vorhanden
- Leicht (weniger als 200g)
- Klein
- Günstig (10.- bis 130.-)

- Leistungsstark (Mehr als 100 Watt)
- Hohe Umdrehungszahl
- Einfache Energieversorgung
- Einfach anzusteuern
- Sauber und leise
- <http://www.modellbau-friedel.com>

### 5.1.1 Bürstenloser Gleichstrommotor

- Gut geeignet für permanente Drehung (Fahren)
- Extra Regelung notwendig
- Nur bedingt präzise (Schwieriges Bremsen)
- Preis ab 100.-
- [https://de.wikipedia.org/wiki/Bürstenloser\\_Gleichstrommotor](https://de.wikipedia.org/wiki/Bürstenloser_Gleichstrommotor)

### 5.1.2 Gleichstrommaschine

- Gut geeignet für permanente Drehung (Fahren)
- Extra Regelung notwendig
- Gutes Regelverhalten
- Nur bedingt präzise (Schwieriges Bremsen)
- Preis ab 80.-
- <https://de.wikipedia.org/wiki/Gleichstrommaschine>

### 5.1.3 Schrittmotor

- Genaue Positionsrückmeldung ohne Sensoren
- Exakte Drehung/Kontrolle
- Schrittverlust bei Überlastung führt zu ungenauer Position
- Hohe Wärmeentwicklung
- Hohe Betriebsspannung
- Preis ab 30.-
- <https://de.wikipedia.org/wiki/Schrittmotor>

## 5.2 Verbrennungsmotor

- Keine Erfahrung auf diesem Gebiet
- Teuer (100.- bis 400.-)
- Keine Größenangabe
- Komplex
- Extra Tank nötig
- Treibstoffkosten
- Zu Kräftig/Ungenau 2-4 PS
- Abgase
- Schwer
- Lärm
- <http://www.modellbau-friedel.com>

## 5.3 Pneumatik

- Nicht in Modellgrösse erhältlich
- Komplex
- Extra Drucktank nötig
- Keine Erfahrung

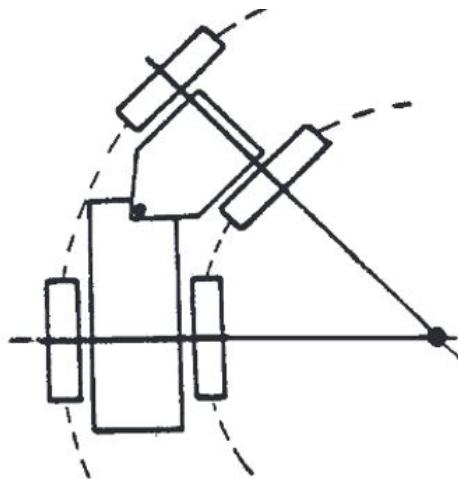
## 5.4 Dampfmaschine

- Stationärer Aufbau (Umbau nötig)
- Teuer (100.- oder mehr)
- Gross (200 x 140 mm)
- Hohes Gewicht (ca. 1kg)
- Keine Leistungsangabe
- [www.modell-dampfmaschinen.de](http://www.modell-dampfmaschinen.de)

# 6 Lenkung

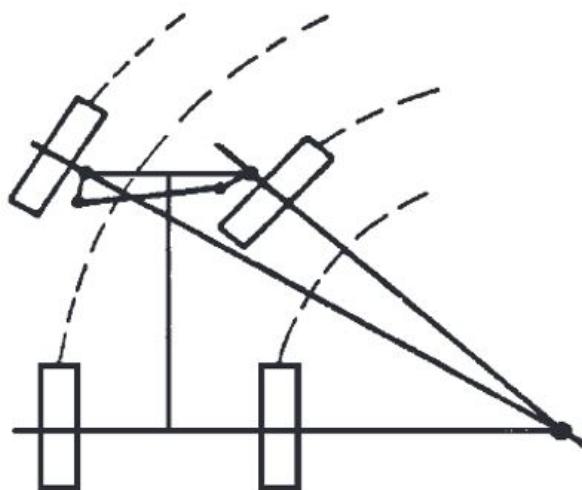
## 6.1 Knicklenkung

- Die Hinterachse volgt immer der gleichen Spur wie die Vorderachse.



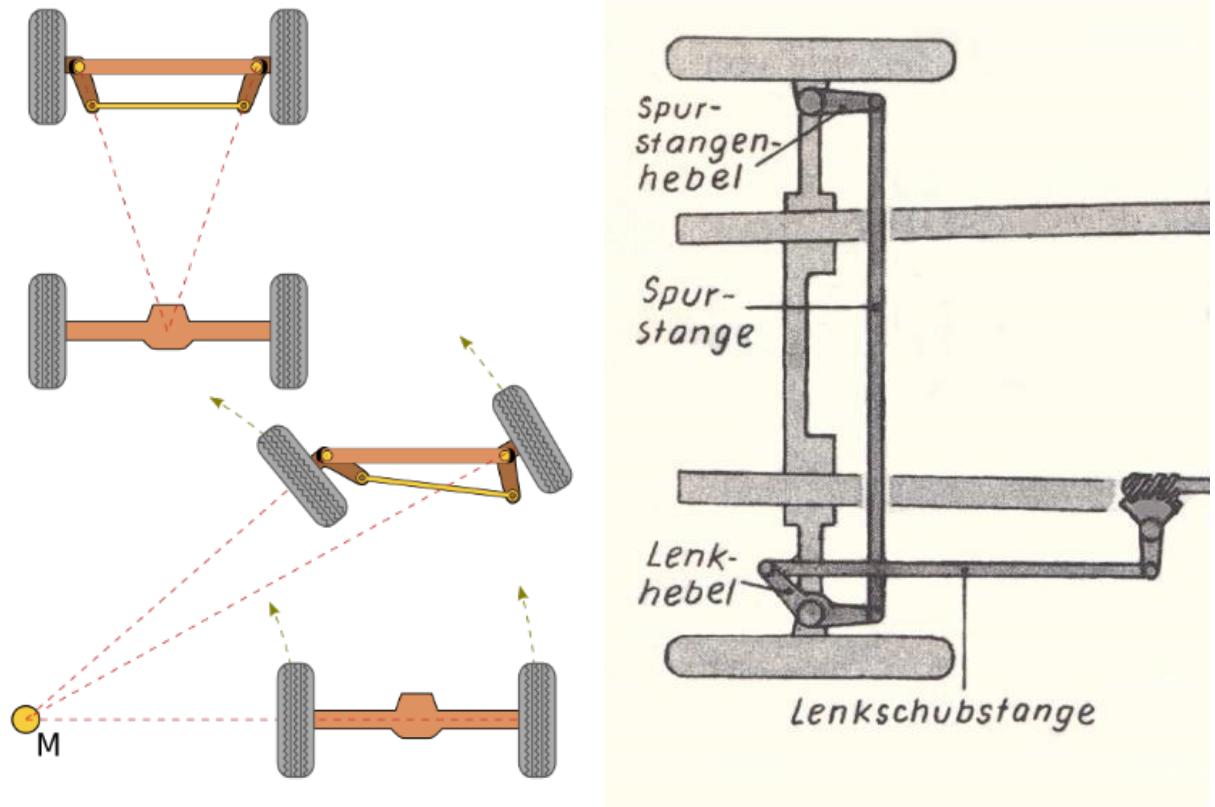
## 6.2 Achsschenkellenkung

- Die Standfestigkeit wird auch bei vollem Lenkeinschlag nicht beeinträchtigt.
- Heutige zweiachsige PWS und Nutzfahrzeuge sind fast alle mit Achsschenkellenkung gebaut.
- Das kurveninnere Rad ist stärker eingeschlagen als das kurvenäussere.



### 6.2.1 Lenktrapez

- Ermöglicht unterschiedliche Einschlagwinkel der Vorderräder.
- Ermöglicht einfaches einstellen eines Spurwinkels.
- Zur Berechnung von Lenktrapezen gibt es vorgefertigte EXCEL Tabellen im Web.



### 6.3 Zweiradlenkung

- Lenkung mit einem Zwei- oder Dreiradfahrzeug durch unterschiedlich schnell rotierende Räder.

### 6.4 Lenkung für Kettenfahrzeuge

- Die Lenkung kann durch unterschiedlich schnelles laufen lassen der Ketten realisiert werden.

## 7 Mikrocontroller-Board

### 7.1 Anforderungen

- Schnittstelle zu möglichem Boardcomputer (UART,I2C,SPI,Ethernet)
- Gute Rechenperformance (besser mehr als zu wenig)
- AD Wandler
- Möglichst Kostengünstig (Richtpreis 20Fr.)

### 7.2 Tinkerforge

#### Vorteile

- Einfache und modulare Schnittstelle zum Boardcomputer
- Einfach erweiterbar mit zusätzlichen Modulen (z.B Sensoren)

- Viele benötigte Module Vorhanden: Schrittmotoren, Distanzsensoren, Liniensensoren und Farbsensoren

### Nachteile

- Grosser Aufwand für eigene Module (Abhängigkeit zu Tinkerforge)

## 7.3 HCS08 Devlopperboard

### Vorteile

- Positive Erfahrungen im MC Modul (Einarbeitung in die Grundfunktionen entfällt)
- Starterkit von der Hochschule erhältlich

### Nachteile

- Begrenzte Rechenpower

## 7.4 Arduinoboard

### Vorteile

- Im Preisrahmen, ein Board gibt es ab 20Fr.-
- Sehr grosse Community
- Einfache Programmierung

### Nachteile

- Besitzt ein Betriebssystem ⇒ weniger Hardware nahe

## 7.5 Freedomboard

### Vorteile

- Im Preisrahmen, ein Board gibt es ab 20Fr.-
- Wurde an der Hochschule bereits eingesetzt. (Mögliche Ansprechpersonen)
- Einfache Programmierung

### Nachteile

- Besitzt ein Betriebssystem ⇒ weniger Hardware nahe

## 8 Distanzsensoren

### 8.1 Anforderungen

- Einfaches auslesen der Daten muss möglich sein
- Genaue Positionsdaten, je nach Anforderungen +/- 1mm
- Preis darf 10Fr. nicht übersteigen
- Möglichst Störungsunabhängig
- Distanz 0.5 - 10cm (je nach Einsatz)
- Möglichst eindeutige Aussage (geringer Streuwinkel)

## 8.2 Ultraschallsensoren

### Vorteile

- Erschwinglich, ein Sensor kostet ungefähr 5Fr
- Unempfindlich auf Störeinflüsse (Ausser andere Ultraschallsensoren)
- Grosse Distanz (>10cm)

### Nachteile

- Kann nicht als Liniensor eingesetzt werden
- Keine klaren Grenzen (relativ grosser Abstrahlwinkel)

## 8.3 Infrarotsensoren

### Vorteile

- Kostengünstig (Sensor und Empfänger kosten zusammen 1.7Fr)
- Kann als Liniensor und Rad-Encoder eingesetzt werden.
- Eher klaren Grenzen (kleiner Abstrahlwinkel)

### Nachteile

- Benötigt zwei AD Eingänge
- Sind empfindlich auf UV-Licht
- Geringe Reichweite (je nach Typ nur bis 5mm)
- Als Liniensor: Nicht senrektes Abtasten ist problematisch

## 9 Farbsensoren

### 9.1 Anforderungen

- Einfaches auslesen der Daten muss möglich sein
- Genaue Farberkennung (Unterschied Gelb zwischen Blau und Grün muss möglich sein)
- Preis darf 10Fr. nicht übersteigen
- Distanz mindestens 5cm

### 9.2 Eigenschaften

#### Vorteile

- Im Preisrahmen, ein Farbsensor kostet ungefähr 10.-

#### Nachteile

- Benötigt AD Eingänge

## 10 Stromversorgung

### 10.1 Anforderungen

- Muss die Energieversorgung während der gesamten Fahrt gewährleisten können.
- Darf sich aufgrund seiner Baugröße und Gewichts nicht negativ auf das Fahrzeug auswirken

### 10.2 Batterien

- Einfache Beschaffung.
- Günstig.
- Relativ unempfindlich.
- Platzsparende Montage möglich.
- Einfache Energieregulierung durch Serie-, Parallelschaltung.
- Geringe Selbstentladung.

#### Nachteile

- Nicht wieder aufladbar.
- Kurze Lebensdauer.
- Geringe Energiedichte.
- Müssen regelmässig ersetzt werden.
- Für den längeren und intensiven Gebrauch eher ungeeignet

### 10.3 NiCd (Nickel-Cadmium)

- Große Auswahl an verschiedenen Bauformen.
- Verhältnismässig billig.
- Lange Lebensdauer.
- Relativ unempfindlich.
- Simple und billige Ladegeräte.

#### Nachteile

- Relativ schwer.
- Geringe Energiedichte.
- Mittelmässiger Ladewirkungsgrad.

## 10.4 Lithium-Akku (LiPo, Lilon)

- Grosse Auswahl von verschiedenen Bauformen.
- Sehr hohe Energiedichte.
- Geringe Selbstentladung.
- Lange Lebensdauer.
- Hoher Ladewirkungsgrad.
- Relativ leicht.

### Nachteile

- Empfindlich bei Über- und Unterschreiten der Spannungsgrenze.
- Teuer.
- Benötigt teure Ladegeräte.

## 10.5 NiMH (Nickel-Metallhydrid)

- Billig.
- Lange Lebensdauer.
- Relativ unempfindlich.
- Simple und billige Ladegeräte.

### Nachteile

- Relativ schwer.
- Mittelmässige Energiedichte.
- Mittelmässiger Ladewirkungsgrad.

# 11 Boardcomputer

## 11.1 Raspberry Pi

- Günstig (32.-)
- CPU 700MHz
- GPU Dual Core (OpenGL, OpenVG)
- RAM 512MB
- 85 x 56 x 17mm
- 4x USB 2.0
- Stromversorgung Micro USB 5V, 2A
- <https://www.pi-shop.ch/raspberry-pi-model-b>

## 11.2 Raspberry Pi 2

- Günstig (44.-)
- CPU 900MHz Quad-Core Cortex-A7
- GPU Dual Core (OpenGL, OpenVG)
- RAM 1GB DDR2
- 86 x 56 x 20 cm
- 4x USB 2.0
- Stromversorgung Micro USB 5V, 2A
- <https://www.pi-shop.ch/raspberry-pi-2-model-b>

## 11.3 Banana Pi

- Günstig (40.-)
- CPU 1GHz Dual-Core
- GPU ARM Mali400MP2
- RAM 1GB DDR3 (Shared with GPU)
- 92 mm x 60 mm
- 2x USB 2.0
- Stromversorgung Micro USB 5V
- <https://www.pi-shop.ch/banana-pi>

## 11.4 BeagleBone Black

- Teuer (60.-)
- AM335x 1GHz ARM® Cortex-A8
- 512MB DDR3 RAM
- 4GB Flash Speicher
- 3D Grafikbeschleunigung
- Ubuntu, Debian und Android unterstützt
- 210-460 mA 5V
- <http://www.beagleboard.org/BLACK>

## 11.5 ODROID-XU4

- Teuer (80.-)
- Exynos5422 Cortex A15 2Ghz 8 Kern CPU
- Cortex A7 8 Kern CPU
- Mali-T628 MP6(OpenGL ES 3.0/2.0/1.1)
- 2GB LPDDR3 RAM
- 82x58x22mm
- 4A 5V
- [http://www.hardkernel.com/main/products/prdt\\_info.php](http://www.hardkernel.com/main/products/prdt_info.php)

Für diese Boards wird Zubehör benötigt, z.B. Stromkabel oder evtl. ein Gehäuse.

# 12 Bilderkennung

## 12.1 Anforderungen

- Objekterkennung (Container, Fahrzeug von rechts).
- Linie erkennen.
- Winkel der Linie erkennen.
- Farbe erkennen.
- Muss auf Linux oder Windows lauffähig sein.
- Muss mit Java, C++ oder C-Sharp kompatibel sein.
- Muss gut dokumentiert sein oder eine gute Community haben.

## 12.2 OpenCV

- Läuft auf den meisten Plattformen.
- Stellt eine grosse Bibliothek mit vielen Zusatzmodulen zur Verfügung.
- Ist mit vielen grossen anderen Bibliotheken kompatibel.
- Hat eine grosse Community.
- Ist gratis und open source (BSD) lizenziert.
- Ist gut dokumentiert.
- Ist kompatibel mit Java, C, C++, Python usw..
- Läuft sehr schnell.

### Nachteile

- Enthält auch viele Funktionalitäten, welche nicht benötigt werden (komplex).

## 12.3 MatLab

### Nachteile

- sehr hohe Lizenzkosten.

## 12.4 SimpleCV

- Läuft auf den meisten Plattformen.
- Ist mit OpenCV kompatibel.
- Ist gratis und open source.
- Hat eine grosse Community.
- Ist gut dokumentiert.
- Ist kompatibel mit Python.
- Läuft schnell.
- Leicht zu erlernen.

### Nachteile

- Deckt möglicherweise nicht alle Anforderungen.

## 12.5 LibCCV

- Läuft auf allen Plattformen.
- Ist gratis und open source (BSD) lizenziert.

### Nachteile

- Die Dokumentation ist mässig.

# 13 Kamerasyteme

## 13.1 Raspberry Pi CAM

- Kosten: ±30.-
- Baugrösse: 25x20x9mm
- Auflösung: 5 Megapixel, bis 30 Bilder pro Sekunde, 1080x720 HD
- Schnittstelle: 15 Pin Flachband MIPI Kameraschnittstelle.
- Lieferzeit: Innert 24h (Digitec)
- Kamerawinkel Horizontal:  $53.5 \pm 0.13$  Grad.
- Kamerawinkel Vertikal:  $41.41 \pm 0.11$  Grad.
- Bildformate: JPEG (beschleunigt), JPEG + RAW, GIF, BMP, PNG, YUV420, RGB888
- Brennweite:  $3.6\text{mm} \pm 0.01$
- Fixer Fokus: 1m bis unendlich
- Software: keine, Kompatibel zu RaspberryPi 1 und 2

### 13.2 Logitech Webcam C525

- Kosten: ±70.-
- Baugröße: 60x 40x 20mm geschätzt, keine genauen Angaben auf Herstellerseite.
- Auflösung: 8 Megapixel, 1280x720 HD
- Schnittstelle: USB 2.0
- Lieferzeit: Innert 24h (Fust, Microspot, Logitech)
- Kamerawinkel Horizontal: Keine Angaben
- Kamerawinkel Vertikal: Keine Angaben
- Bildformate: Keine Angaben
- Brennweite: Keine Angaben
- Fokus: Autofokus
- Software: Logitech Software nur Windows Vista, 7 und 8.

### 13.3 Logitech Webcam C615

- Kosten: ±100.-
- Baugröße: 80x 40x 20mm geschätzt, keine genauen Angaben auf Herstellerseite.
- Auflösung: 8 Megapixel, 1920x1080 Full HD
- Schnittstelle: USB 2.0
- Lieferzeit: Innert 24h (Fust, Microspot, Logitech)
- Kamerawinkel Horizontal: Keine Angaben
- Kamerawinkel Vertikal: Keine Angaben
- Bildformate: Keine Angaben
- Brennweite: Keine Angaben
- Fokus: Autofokus
- Software: Logitech Software nur Windows Vista, 7 und 8.

### 13.4 Logitech Webcam C920

- Kosten: ±100.-
- Baugröße: 80x 40x 20mm geschätzt, keine genauen Angaben auf Herstellerseite.
- Auflösung: 5 Megapixel, 1080x720 HD
- Schnittstelle: USB 2.0, VID\_046D&PID\_0821
- Lieferzeit: Innert 24h (Fust, Microspot, Logitech)
- Kamerawinkel Diagonal: 83 Grad
- Bildformate: Keine Angaben
- Brennweite: 4.3mm
- Fokus: Autofokus
- Software: Logitech Software nur Windows Vista, 7, 8 und 10.

## 13.5 Zusammenfassung

Zusammenfassend können zwei Kamerasysteme verwendet werden. Dies sind einerseits die RaspberryPi Cam und andererseits eine Webcam, aus den Aufgeführten Modellen oder von weiteren Anbietern. Gemäss Tabelle xxx funktionieren diverse Modelle mit dem Raspberry Pi oder vergleichbaren Boards.

## 13.6 Vor- und Nachteile

### 13.6.1 Raspberry Pi CAM

#### Vorteile:

- Preis.
- Open Source und sehr Detaillierte Funktionsangaben.
- API vorhanden.
- Baublöcke und Befestigung.
- Schnittstelle auf Raspberry Pi abgestimmt.

#### Nachteile:

- Fixer Fokus ab 1m.
- Winkel mit 53.5 Grad etwas klein.
- Schnittstelle, wenn kein Raspberry Pi verwendet wird.

### 13.6.2 Webcams

#### Vorteile:

- Autofokus.
- Hohe Auflösung.
- Größere Auswahl an Modellen.

#### Nachteile:

- Preis.
- Baugröße oft nicht genau angegeben.
- Geringere Herstellerinformationen.
- Fehlende API.
- Befestigung, Gehäuse muss evtl entfernt werden.

## 13.7 Steuerung

Die Steuereinheit ist Haupteinheit für das autonome Fahren. In der Steuerung werden die Spurdaten eingelesen, verarbeitet und die entsprechenden Massnahmen getroffen. Ohne eine gute Steuerung ist das autonome Fahren nicht möglich.

### 13.8 Ein Steuerungsboard

Ein Board für die Verarbeitung der Spurdaten und für den Zugriff auf die Peripherie. **Vorteile**

- Keine Kommunikationsschnittstelle nötig (Fehlerquelle eliminiert)
- Preisersparnis

#### Nachteile

- Arbeitsaufteilung schwierig
- Weniger Rechenperformance

### 13.9 Bordcomputer und Hardwarecontroller

Ein Board für die Verarbeitung der Spurdaten und Steuerung und ein anderes Board für den Zugriff auf die Peripherie und alternative Aufgaben. **Vorteile**

- Getrennte Aufgabenbereiche
- Genügend Rechenperformance
- Weniger Seiteneffekte

#### Nachteile

- Braucht zwei Boards
- Schnittstelle nötig

## 14 Morphologischer Kasten

Energie Versor-gung	Elektrisch (Ak-ku)	Brennstoff	Pneumatik (di-rekt)	Pneumatik (Drucktank)
Startsignal	Knopf	Laptop	Smartphone	
Stoppsignal	Akustisch	Licht (Blinken)	Laptop Rück	Smartphone Rück
Antrieb	Elektromotor	Verbrennungs-motor	Pneumatik (Drucktank)	
Lenkung	Trapezlenkung	Achsenlenkung	Raupenlenkung	MC Car
Chassis	2 Achsen 4 Rä-der	2 Raupen	2 Räder 1 Stütze	
Anhalten	Bremsen	Motor stoppen		
Spurerkennung	Optisch	Infrarot	Ultraschall	
Containererkennung	Optisch	Infrarot	Ultraschall	Farbsensor (z.B. Lego)
Greifen	Gelenkarm	Kettengreifer		
Leerung	Schräge	Kippen	Stossen	
Lagerung	Becken	Beweglicher Container		

Tabelle 1: Morphologischer Kasten

## 15 Recherchequellen

Themengebiet	Beschreibung	Quelle	Bewertung (1-5)
<b>Bilderkennung</b>	OpenCV Beschreibung	<a href="http://docs.opencv.org/master/d1/dfb/intro.html#gsc.tab=0">http://docs.opencv.org/master/d1/dfb/intro.html#gsc.tab=0</a>	3
	Vergleich von OpenCV und SimpleCV	<a href="http://simplecv.tumblr.com/post/19307835766/opencv-vs-matlab-vs-simplecv">http://simplecv.tumblr.com/post/19307835766/opencv-vs-matlab-vs-simplecv</a>	4
	SimpleCV Beschreibung	<a href="http://simplecv.org/">http://simplecv.org/</a>	3
	Linienerkennung mit Open CV	<a href="https://www.youtube.com/watch?v=aGGehlgiZoQ">https://www.youtube.com/watch?v=aGGehlgiZoQ</a>	3
<b>Boardcomputer</b>	Raspberry Pi	<a href="https://www.pi-shop.ch/raspberry-pi-model-b">https://www.pi-shop.ch/raspberry-pi-model-b</a>	4
	Raspberry Pi 2	<a href="https://www.pi-shop.ch/raspberry-pi-2-model-b">https://www.pi-shop.ch/raspberry-pi-2-model-b</a>	4
	Banana Pi	<a href="https://www.pi-shop.ch/banana-pi">https://www.pi-shop.ch/banana-pi</a>	4
<b>Antrieb</b>	Elektromotor	<a href="http://www.modellbau-friedel.com">http://www.modellbau-friedel.com</a>	3
	Verbrennungsmotor	<a href="http://www.modellbau-friedel.com">http://www.modellbau-friedel.com</a>	3
	Dampfmaschine	<a href="http://www.modell-dampfmaschinen.de">www.modell-dampfmaschinen.de</a>	3
<b>Lenkung</b>	Beschreibung Knicklenkung und Bild	<a href="http://www.portmanns.ch/Repetition/Fahrwerk/Lenkungsarten.pdf">http://www.portmanns.ch/Repetition/Fahrwerk/Lenkungsarten.pdf</a>	3
	Beschreibung und Bild Achsschenkellenkung	<a href="http://www.urlaub-und-hobby.de/metallbaukasten/so09dt.html">http://www.urlaub-und-hobby.de/metallbaukasten/so09dt.html</a>	4
	Lenktrapez	<a href="http://portmanns.ch/Repetition/Fahrwerk/Achssch.pdf">http://portmanns.ch/Repetition/Fahrwerk/Achssch.pdf</a>	2
<b>Stromversorgung</b>	Vergleich von Akkutypen	<a href="http://www.akku-abc.de/akku-vergleich.php">http://www.akku-abc.de/akku-vergleich.php</a>	3
	Grundlagenbeschreibung Akkutypen	<a href="http://rn-wissen.de/wiki/index.php/Akku-Grundlagen">http://rn-wissen.de/wiki/index.php/Akku-Grundlagen</a>	4
	Vergleich von Batterien und Akkus	<a href="http://www.conrad.de/ce/de/content/ti_AkkusBatterien/Nickel-Zink-Akkus-die-neue-Alternative-zu-den-herkömmlichen-Batterien">http://www.conrad.de/ce/de/content/ti_AkkusBatterien/Nickel-Zink-Akkus-die-neue-Alternative-zu-den-herkömmlichen-Batterien</a>	3
	Vergleich von Batterien und Akkus	<a href="http://www.computerbild.de/artikel/cb-Tests-PC-Hardware-Teure-Marken-gegen-Billig-Batterien-Mignon-AA-Micro-AAA-4640760.html">http://www.computerbild.de/artikel/cb-Tests-PC-Hardware-Teure-Marken-gegen-Billig-Batterien-Mignon-AA-Micro-AAA-4640760.html</a>	2
<b>Liniensensoren</b>	Beispiel eigene Liniensensoren mit Tipps	<a href="http://www.cs.hs-rm.de/~linn/vpdv0708/asuro1/das_projekt_liniensensoren.html">http://www.cs.hs-rm.de/~linn/vpdv0708/asuro1/das_projekt_liniensensoren.html</a>	3

	Opensource Beispiel Liniesensor	<a href="https://www.tinkerforge.com/de/shop/bricklets/line-bricklet.html">https://www.tinkerforge.com/de/shop/bricklets/line-bricklet.html</a>	4
	Opensource Beispiel Infrarot Linienfolger Arduino	<a href="http://www.instructables.com/id/Arduino-Line-Following-Robot-for-Beginners/">http://www.instructables.com/id/Arduino-Line-Following-Robot-for-Beginners/</a>	4
	Opensource Beispiel Infrarot Linienfolger Selfmade + Regelung	<a href="http://www.societyofrobots.com/member_tutorials/book/export/html/350">http://www.societyofrobots.com/member_tutorials/book/export/html/350</a>	2
<b>Distanzsensoren</b>	Arduino Ultraschallsensor Überblick	<a href="https://www.youtube.com/watch?v=U75vH-VfaPQ">https://www.youtube.com/watch?v=U75vH-VfaPQ</a>	3
	Allgemeiner Sensor Überblick	<a href="http://www.robotc.de/ev3-sensoren/">http://www.robotc.de/ev3-sensoren/</a>	5
	IR Sensor (von Tinkerforge)	<a href="https://www.tinkerforge.com/de/shop/accessories/sensors/infrared-sensor-gp2y0a41sk0f.html">https://www.tinkerforge.com/de/shop/accessories/sensors/infrared-sensor-gp2y0a41sk0f.html</a>	1
	IR Sensor (von Tinkerforge, kurze Distanz)	<a href="http://www.tinkerforge.com/de/doc/Hardware/Bricklets/Line.html">http://www.tinkerforge.com/de/doc/Hardware/Bricklets/Line.html</a>	1
<b>Farbsensor</b>	Farbsensormodul	<a href="http://www.amazon.de/RGB-Farbsensor-mit-Filter-Arduino/dp/B00CYOFN2K">http://www.amazon.de/RGB-Farbsensor-mit-Filter-Arduino/dp/B00CYOFN2K</a>	3
	Farbsensormodul	<a href="https://www.openimpulse.com/blog/products-page/product-category/tcs230-color-sensor-module-2/">https://www.openimpulse.com/blog/products-page/product-category/tcs230-color-sensor-module-2/</a>	3
<b>Mikrocontroller Module</b>	MC Modul mit Erweiterungen	<a href="https://www.tinkerforge.com/de/">https://www.tinkerforge.com/de/</a>	4
	Freedomboard	<a href="http://www.freescale.com/products/arm-processors/kinetis-cortex-m:KINETIS">http://www.freescale.com/products/arm-processors/kinetis-cortex-m:KINETIS</a>	5
	Freedomboard Überblick KL-Line	<a href="http://cache.freescale.com/files/microcontrollers/doc/selector_guide/KINETISLMCUSELGD.pdf?fpss=1&amp;WT_TYPE&gt;Selecto">http://cache.freescale.com/files/microcontrollers/doc/selector_guide/KINETISLMCUSELGD.pdf?fpss=1&amp;WT_TYPE&gt;Selecto</a>	4
	Arduino Board	<a href="https://www.arduino.cc/en/Main/Boards">https://www.arduino.cc/en/Main/Boards</a>	2
<b>Greifer</b>	Beschreibung diverser Greifer	<a href="http://www.zwahlenag.ch/produkte/greifer-pneumatische-greifzangen.php">http://www.zwahlenag.ch/produkte/greifer-pneumatische-greifzangen.php</a>	3
<b>Beladen</b>	Video zu diversen Belademechanismen die in der Praxis verwendet werden	<a href="https://www.youtube.com/watch?v=LTUjiLxzDQs">https://www.youtube.com/watch?v=LTUjiLxzDQs</a>	4
<b>Entladen</b>	Bild einer möglichen Variante	<a href="http://www.asia.ru/de/ProductInfo/1423164.html">http://www.asia.ru/de/ProductInfo/1423164.html</a>	2
<b>Kamerasysteme</b>	Technische Doku Raspberry Pi CAM	<a href="https://www.raspberrypi.org/documentation/hardware/camera.md">https://www.raspberrypi.org/documentation/hardware/camera.md</a>	4
	Logitech- Produkteseite Kamera	<a href="http://support.logitech.com/de_ch/home">http://support.logitech.com/de_ch/home</a>	2
	Raspberry Pi Infrarotkamera	<a href="https://www.raspberrypi.org/blog/pi-noir-infrared-camera-now-available/">https://www.raspberrypi.org/blog/pi-noir-infrared-camera-now-available/</a>	2
10. Dezember 2015	Spurerkennung Kamera	<a href="http://weelug.blogspot.ch/2011/08/spurerkennung.html">http://weelug.blogspot.ch/2011/08/spurerkennung.html</a>	Seite 24 von 25 2
	Infoseite iOnRoad App	<a href="https://play.google.com/store/apps/details?id=com.nicitud.iOnRoad">https://play.google.com/store/apps/details?id=com.nicitud.iOnRoad</a>	3

