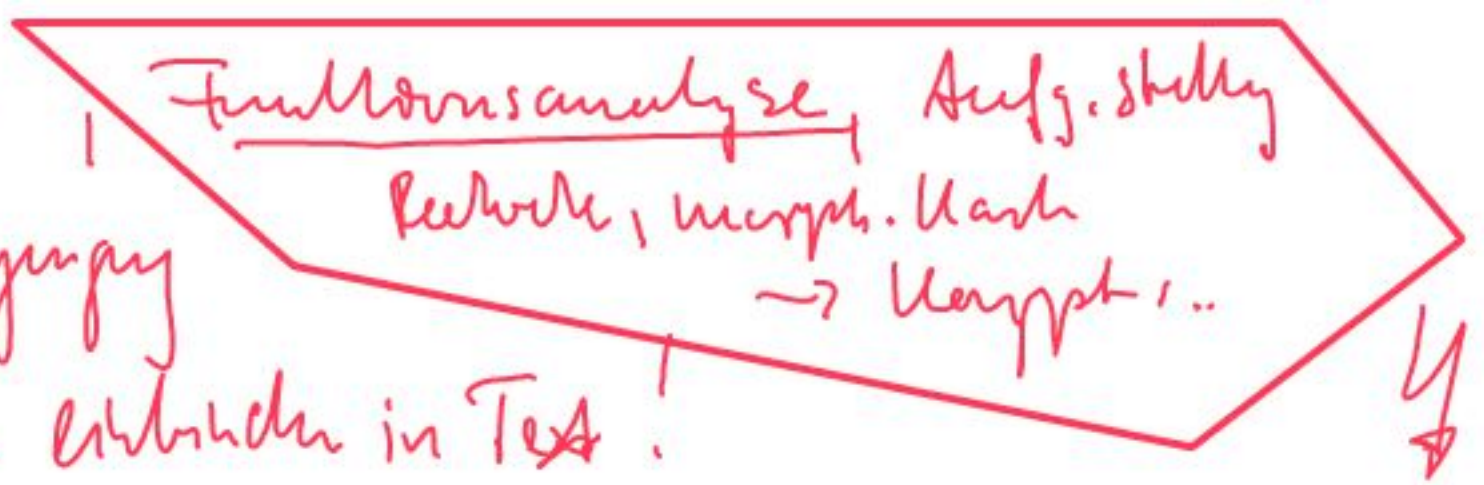


Produktentwicklung 1

1. Abstract.....	3
2. Einleitung	4
3. Konzeptfindung	4
3.1. Teilfunktionenanalyse	5
3.2. Morphologischer Kasten	5
3.3. Auswahl.....	5
3.4. Bewertung	7
3.5. Entschluss	7
4. Lösungskonzept	9
5. Komponentenbeschreibung	10
5.1. Hebemechanismus und Greifer	10
5.1.1. Funktionsweise	10
5.1.2. Detaillierte Beschreibung.....	10
5.1.3. Begründung	10
5.2. Fahrgestell und Lenkung.....	11
5.2.1. Funktionsweise	11
5.2.2. Detaillierte Beschreibung.....	11
5.2.3. Begründung	11
5.3. Entsorgung	13
5.3.1. Berechnung (in Anhang?)	13
5.3.2. Funktionsweise	13
5.3.3. Detaillierte Beschreibung.....	13
5.3.4. Begründung	14
5.4. Sensoren	15
5.4.1. Farbsensoren.....	15
5.4.2. Ultraschallsensor	16
5.5. Energieversorgung	18
5.5.1. Akku	18
5.5.2. Akkuüberwachung	18
5.5.3. Spannungswandlung	19
5.6. Hardware und Software	19
5.6.1. Produktiv Aufbau	19
5.6.2. Kommunikation	20
5.6.3. Entwicklung	21
5.6.4. Deployment	22
5.6.5. Produktiver Betrieb.....	22
5.7. Motoren	23
5.7.1. Auswahl.....	23
5.7.2. Motorensteuerung und Regelung	23
5.7.3. Encoder	24
6. Projektmanagement und Planung	25
6.1. Organigramm.....	25
6.2. Funktionsbeschreibung	25
6.3. Planung.....	25

- Roks Faden
- Faalt, Übergang
- Graphiken einbinden in Text!
- Spezifische Sein! Nachvollziehbarkeit fehlt oft z.B.
weshalb encode, weshalb gezeichnet?
- Software "nicht beschreiben" (Firmware), zu wenig tief, nicht nachvollziehbar
- Sensor okay
- Hardware unklar (analog SW)
- Verbindung zu PC? Start/Stop wo?
- Bilderkennung fehlt ganz



7. Schlussdiskussion	26
7.1. Kosten	26
7.2. Lessons learned	26
7.3. Risiken	26
7.4. Offene Punkte	26
7.5. Ausblick auf PREN2	26
7.6. Fazit	26
8. Abbildungsverzeichnis	27
9. Tabellenverzeichnis	27
10. Anhang.....	28

Betreuender Dozent:

Dipl. El. Ing. ETH Martin Vogel

Team

Manuel Felber

Severin Gabriel

Jonas Hansen

Roman Marti

Pascal Minder

Leila Müller

Valentin Peltier

1. Abstract

2. Einleitung

In den Modulen PREN 1 und PREN 2 der Hochschule Technik und Architektur Luzern werden jedes Jahr interdisziplinäre Teams zusammengestellt, die eine komplexe Aufgabenstellung lösen müssen. Besonderen Wert wird dabei auf die Zusammenarbeit der verschiedenen Fachrichtungen gelegt.

Die Aufteilung des Projekts geschieht wie folgt: Im Modul PREN 1 gilt es, ein Lösungskonzept zu erarbeiten. Dieses wird im zweiten Teil, dem Modul PREN 2, umgesetzt. Zum Abschluss findet ein Wettbewerb statt. Bei diesem werden die Teams nach verschiedenen Kriterien bewertet. Das Team mit den meisten Punkten wird zum Sieger. Diese Dokumentation widmet sich dem Modul PREN 1.

Als erstes wird kurz auf die Lösungsfindung eingegangen. Weiter wird in Kapitel 4 eine Übersicht über das Konzept gegeben. In Kapitel 5 werden die einzelnen Teilfunktionslösungen genau beschrieben. Dann wird kurz auf das Projektmanagement und die Planung eingegangen. Schliesslich werden in der Schlussdiskussion auf die Kosten, Risiken und die offengebliebenen Punkte eingegangen und ein Ausblick auf den zweiten Teil von PREN

3. Aufgabenstellung (grob)

Pflichtauftrag (verweis)

→ Reduziere (verweis)

→ Funktionsanalyse

Zusammenhang
füllt ...

3. Konzeptfindung

In den folgenden Abschnitten wird genauer auf die Findung des Konzepts für die automatische Müllabfuhr eingegangen.

3.1. Teilfunktionenanalyse

Für die Konzeptfindung wurde eine Teilfunktionenanalyse erstellt. Damit wird die Aufgabenstellung besser handhabbar. Diese Teilfunktionen bestehen aus: Fahrgestell und Lenkung, Heben und Greifen, Entsorgen, Energieversorgung, Erkennung der Umgebung. Für diese Teilfunktionen wurden verschiedene Lösungsmöglichkeiten recherchiert und in einem morphologischen Kasten zusammengetragen.

3.2. Morphologischer Kasten

Der Morphologische Kasten (Anhang) erlaubte es dem Team, eine Übersicht über die möglichen Lösungen zu erhalten. Anhand des Inputs der verschiedenen Fachrichtungen erhielten die Teammitglieder Einblick in verschiedene Methoden der Problemlösung. Es wurden mehrere mögliche Gesamtkonzepte zusammengestellt. Dabei wurde auf die Umsetzbarkeit und denkbare Probleme geachtet, um verschiedene plausible Varianten zu finden. So wurden der Hamster als Energiequelle und das Beamen zum Abfalleinsammeln auf Grund technischer Schwierigkeiten und grossem Aufwand ausgeschlossen. In den folgenden Tabellen werden die ausgewählten Varianten präsentiert und anschliessend bewertet.

3.3. Auswahl

Für die Konzeptanalyse wurden verschiedene Zielkriterien wie Geschwindigkeit und Genauigkeit definiert und nach ihrer Wichtigkeit gewichtet. Danach wurden die verschiedenen Lösungsvarianten anhand dieser Kriterien bewertet. Eine niedrige Punktzahl bedeutete, dass das Konzept die Kriterien schlecht erfüllt, eine hohe Punktzahl, dass es sie gut erfüllt. Dabei standen für die Teilfunktionen verschiedene Lösungsmöglichkeiten zur Auswahl. Ein Beispiel ist in Tab.1 zu finden.

erklären was gemacht wurde. Recherche, Teilfunktionenanalyse
→ Morph. Kasten. → Auswahl
(recherche, Hinführung)

Konzept	Kamera + RGB-Sensor	Linienensor + Ultraschall + RGB-Sensor	Stereokamera
Beschreibung	<ul style="list-style-type: none"> • Kamera prüft Spurhaltung • Kamera prüft Rechtsvortritt • RGB-Sensor prüft Farbe Container • Fahrzeug in Position sobald RGB-Sensor nicht mehr blau/grün • Früherkennung Container durch Kamera 	<ul style="list-style-type: none"> • Liniensensor prüft Spurhaltung • Ultraschall prüft Rechtsvortritt • RGB-Sensor prüft Farbe Container • Fahrzeug in Position sobald RGB-Sensor nicht mehr blau/grün • Keine Früherkennung 	<ul style="list-style-type: none"> • Kamera prüft Spurhaltung • Stereokamera prüft Rechtsvortritt • Kamera prüft Farbe Container • Fahrzeugposition berechnet mit Distanz zu Container + Position Greifarm • Früherkennung Container durch Kamera
Vorteile	<ul style="list-style-type: none"> • Guter Kompromiss bezüglich Aufwand und Ertrag • Positionierung bei Container 	<ul style="list-style-type: none"> • Distanzerkennung für Rechtsvortritt • Einfache Implementierung • Positionierung 	<ul style="list-style-type: none"> • Distanzerkennung für grosses Sichtfeld • Alles in einem System • Einfache Ergänzung mittels Sensor
Nachteile	<ul style="list-style-type: none"> • Keine Distanzerkennung • Rechtsvortritt • Distanzerkennung Container nicht möglich 	<ul style="list-style-type: none"> • Keine Objekterkennung 	<ul style="list-style-type: none"> • Rechenintensiv
Risiken	<ul style="list-style-type: none"> • Rechtsvortritt erkennen 	<ul style="list-style-type: none"> • Gestrichelte Linie (Kurve) • Falsches Objekt greifen 	<ul style="list-style-type: none"> • Genauigkeit Containerposition

Tabulle? Text dayes (Kontrolle)

3.4. Bewertung

Alle Teammitglieder hatten die Aufgabe, eine funktionsfähige Lösung zu präsentieren. Diese wurden anhand bereits ausgewählter Gewichtungen bewertet. Diese sind in der untenstehenden Tabelle aufgelistet.

Konzepte											
Zielkriterien	Gewichtungs-Faktor	Variante 1		Variante 2		Variante 3		Variante 4		Variante 5	
		Bewertung		Bewertung		Bewertung		Bewertung		Bewertung	
Geschwindigkeit	0.2	4	0.8	4	0.8	3	0.6	1	0.2	4	0.8
Produktionsaufwand	0.15	4	0.6	2	0.3	3	0.45	1	0.15	3	0.45
Genauigkeit	0.3	4	1.2	4	1.2	5	1.5	5	1.5	5	1.5
Gewicht	0.05	3	0.15	4	0.2	3	0.15	2	0.1	3	0.15
Programmieraufwand	0.25	3	0.75	3	0.75	2	0.5	2	0.5	2	0.5
Energiebedarf	0.05	3	0.15	3	0.15	3	0.15	2	0.1	2	0.1
Summe	1		3.65		3.4		3.35		2.55		3.5

Bewertung		
Zielkriterien	1,,	„5
Geschwindigkeit	sehr langsam	sehr schnell
Produktionsaufwand	hoch	gering
Genauigkeit	ungenau	genau
Gewicht	gross	klein
Programmieraufwand	hoch	simpel
Energiebedarf	hoch	tief
Summe	schlechte Lösung	gute Lösung

3.5. Entschluss

(Festlegung)

Die Bewertung der verschiedenen Varianten erfolgt in **Error! Reference source not found.** Nach den Bewertungen von **Error! Reference source not found.** entspricht das Konzept mit der höchsten Punktzahl den Zielkriterien am besten. Die Variante 1 erhielt die höchste Punktzahl mit 3.65. Es folgen dicht darauf Variante 5 mit 3.5 Punkten und Variante 2 mit 3.4 Punkten. Die Unterschiede der drei Varianten liegen hauptsächlich bei der Wahl der Sensoren, daher sind sie auch recht ähnlich bewertet. Variante 5 unterscheidet sich von der Variante 1 auch bei der Energiequelle. Anstelle des LiPo-Akkus soll hier Solarenergie genutzt werden. Diese liefert jedoch nicht die benötigte Spannung. Ausserdem entsprechen die Lichtverhältnisse beim Wettbewerb eventuell nicht den Bedürfnissen. Daher fällt die Variante 5 weg. Die Variante 4 ist aufgrund des hohen Produktionsaufwandes und der geringen Geschwindigkeit ausgeschieden. Der Spindelkipper lässt sich schlecht mit der Raupe kombinieren und wäre deutlich lang-

samer, als wenn man nur eine Klappe öffnen muss, ebenso das Förderband anstelle des Hebelarms. Daher wurde Variante 1 als Konzept ausgewählt.

Infolgende wird das Kapitel beschrieben.

noch Potential, zu einfach

4. Lösungskonzept

Das Lösungskonzept besteht aus einem Zweipunktgreifer mit Hebelarm, einem Fahrgestell mit vier Rädern und einer Achsschenkellenkung. Zum Transport des Mülls wird eine Mulde verwendet, die eine Klappe zur Entleerung besitzt. Die Energieversorgung wird durch ein LiPo Akku sichergestellt.

Als Rechenwerk werden ein RED Brick und zwei Raspberry Pis 2 eingesetzt. Die Wahl der Sensoren liegt momentan bei einer Kamera und einem RGB-Sensor.

Noch fehlend:

Ablaufdiagramm

Übersicht/Darstellungen

Komponentendiagramm

Tablet? Gesamtsystem

Sensoren? → Bezug auf Funktionsanalyse:
Spur halten, Müll erkennen...
Antrieb?

* Zeichnung mit Legende z.B.

5. Komponentenbeschreibung

In den folgenden Kapiteln werden die Teilfunktionen detaillierter beschrieben.

5.1. Hebemechanismus und Greifer

Berechnen, Größe, Servoanalogie

5.1.1. Funktionsweise

Fehlt noch. Eventuell zusammengefasst unter der Übersicht (Kapitel 4)

5.1.2. Detaillierte Beschreibung

Fehlt noch inkl aktueller Bilder

5.1.3. Begründung

Um den Greifarm zu betätigen, wurden verschiedene Varianten in Erwägung gezogen. So kamen unter anderem eine Zahnradgetriebenen Gelenk, eine Seilwinde, eine Schraube oder ein Servomotor in Frage. Von diesen Möglichkeiten wurde der Servomotor gewählt. Dieser liefert das Mehrfache der benötigten Leistung. Damit können Risiken, wie das Verlieren des Containers beim Heben, gleich ausgeschlossen werden.

Die Wahl des Servomotors hat auch die Gestaltung des Greifers beeinflusst. Würde anstelle eines Servomotors eine Seilwinde gewählt, wäre ein Saugrohr die bessere Wahl, ein Förderband wäre besser mit einem Zahnrad anzutreiben.

In der untenstehenden Abbildung ist ein Greifarm zu sehen. Die beiden roten Pfeile indizieren die Positionen der beiden Servomotoren welche einerseits zum Antrieb des Arms (1) und andererseits zum Betätigen der Klemmen benutzt werden (2). Zu erwähnen ist die fixe Knickung des Greifarms welche aufgrund der volumenspezifischen Produktanforderungen nötig ist.

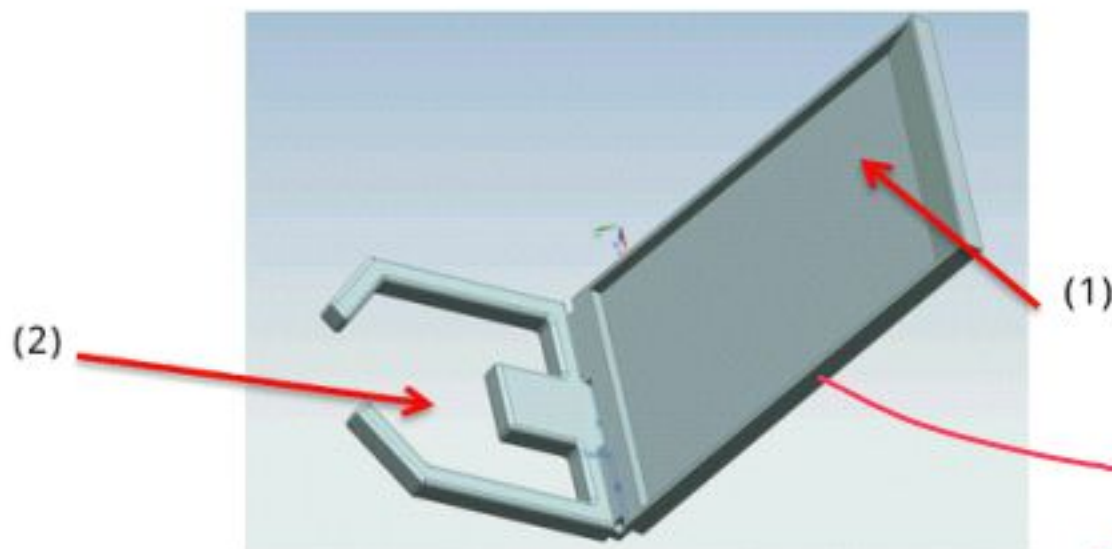


Abbildung 1: Provisorische Darstellung des Greiferarms

Betung zu Gesamtprojekt!

Wahrscheinlich...

Servo drive!

5.2. Fahrgestell und Lenkung

Das Lenkkonzept sieht ein Fahrzeug mit vier Rädern vor. Diese werden mittels einer Achsschenkellenkung gesteuert.

5.2.1. Funktionsweise

Berechnungen?

5.2.2. Detaillierte Beschreibung

Das Fahrgestell mit vier Rädern erhält einen ersten Aufbau, in dem Motor und Akku untergebracht sind. Darauf wird eine Grundplatte befestigt. Diese bietet Platz für die Mulde, die Sensoren, den Greifarm und die Rechenwerke.

Die Lenkung des Fahrzeugs erfolgt über die Vorderräder. Diese werden über eine Achsschenkellenkung gesteuert. Der Servomotor bewegt den Schenkel.

Fehlt noch: Bilder

$\pm 25^\circ$?

Fehlt noch: Beschreibung inkl. Mögliche Auslenkung usw.

5.2.3. Begründung

Das vierrädrige Fahrgestell wurde gewählt, da es eine zentrierte Schwerpunktlage besitzt. Es hat eine größere Stabilität beim Aufsammeln der Container als ein Fahrgestell mit nur drei Rädern. Vergleicht man die Stabilität beider Fahrzeuge in einer Hanglage, welche das Aufsammeln eines Containers simuliert, so stellt sich schnell heraus, dass das Fahrzeug mit drei Rädern schneller ins Kippen kommt als dessen Konkurrent mit vier Rädern.

Für die Lenkung eines autonomen Fahrzeuges mit vier Rädern stehen wiederum zwei Lösungsvorschläge zur Auswahl: Lösungsvorschlag A sieht eine Knicklenkung vor. Als Lösungsvorschlag B kommt eine Schenkellenkung in Frage. Hier wird ein Rad durch einen Hebelarm ausgelenkt.

Der mechanische Aufwand zur Realisierung von Lösungsvorschlag A ist größer als derjenige bei B. Aus diesen Gründen wurde für die Lenkung des autonomen Fahrzeuges eine Schenkellenkung gewählt, welche einfach anzusteuern, kostengünstig, genau und mechanisch einfach realisierbar ist.

Bezug Radius Bahn, Auslenkung

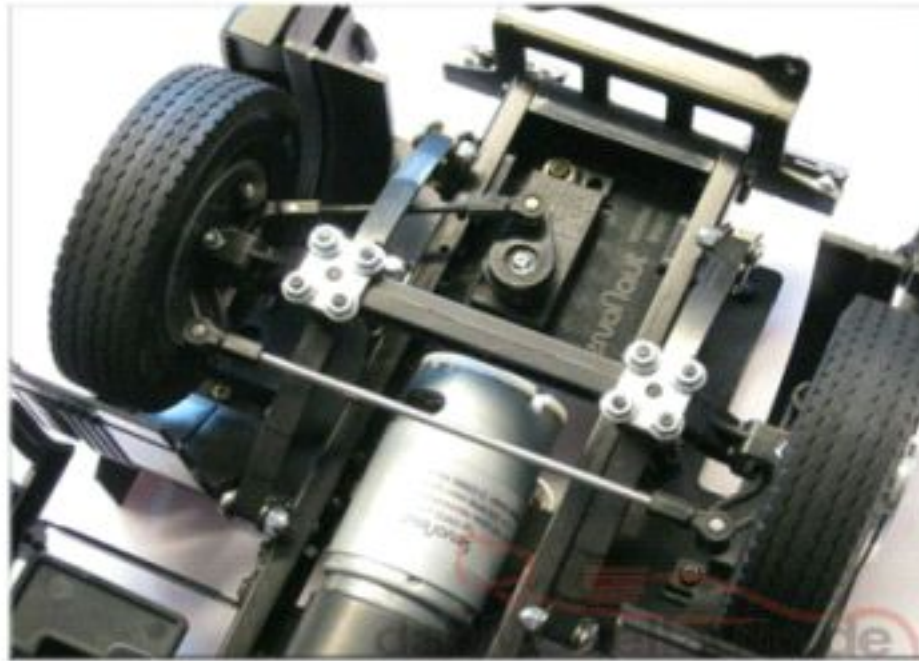


Abbildung 2: Beispiel einer Servolenkung¹

¹ https://www.dasmodellauto.de/yanis42/system/modules/y42_ishop/web/images/product/servonaut-servonaut-svp-SVP-121-img2-800x600px.jpg 05.11.2015

5.3. Entsorgung

5.3.1. Berechnung (in Anhang?)

Masse Klappe = $5\text{cm} \times 6.1\text{cm} \times 0.56\text{cm} = 17.1\text{cm}^3$

Dichte Plexiglas = 1.18 g/cm^3

MMüll = $100\text{g} = 0.1\text{kg}$

Mklappe = $\text{Dichte Plexiglas} \times \text{Masse Klappe} = 1.18 \times 17.1 = 20.2\text{g}$

$F_{\text{gkl}} = M_{\text{klappe}} \times g = 0.0202\text{kg} \times 10\text{m/s}^2 = 0.0202 \times 10 = 0.202\text{N}$

Erforderlich = $F_{\text{gkl}} \times \sin(b) = 0.202\text{N} \times \sin(60^\circ) = 0.175\text{N}$

$F_{\text{gmüll}} = M_{\text{Müll}} + g = 0.1\text{kg} + 10\text{m/s}^2 = 1\text{N}$

$F_{\text{ab}} = F_{\text{gmüll}} \times \sin(b) = 0.87\text{N}$

Moment = $F_{\text{ab}} \times S = 0.87 \times 5 = 4\text{Ncm}$

hier
lassen.
interpretieren ...
Text

Zeichnung fehlt

5.3.2. Funktionsweise

Der gesammelte Müll wird in einer Mulde mit schrägem Boden zum Entsorgungsbecken transportiert. Die Mulde ist mit einer Klappe versehen. Ein Riegel hält die Klappe geschlossen. Beim Entsorgungsbecken hebt ein Servomotor den Riegel an und senkt die Klappe. Durch die Schräge der Mulde wird die Schwerkraft zur Entleerung genutzt. Sobald die Mulde leer ist, senkt der Servomotor den Riegel. Über ein Scharnier wird die Klappe wieder geschlossen.

5.3.3. Detaillierte Beschreibung

Die Mulde besitzt einen Müllauffang, in den der Greifarm die Container entleert. Von dort rutscht der Müll hinab in das Innere der Mulde. Diese besitzt einen Boden mit einer Schräge von **ca. 15°**. Dadurch liegt das Gewicht des Mülls auf der Klappe. Diese wird durch einen Riegel verschlossen gehalten. Die Belastung für den Riegel beläuft sich auf ein Moment von 4 Ncm. Um den Riegel zu öffnen, wird ein Servomotor angesteuert. Die Klappe öffnet sich auf **ca. 105°**.

Der Boden der Mulde ist über eine Achse mit der Klappe verbunden. Um zu verhindern, dass der Müll an der Verbindungsstelle hängen bleibt, wird die Konstruktion mit einer Folie versehen.

Das Schliessen der Klappe erfolgt über

Uchlas
wo?

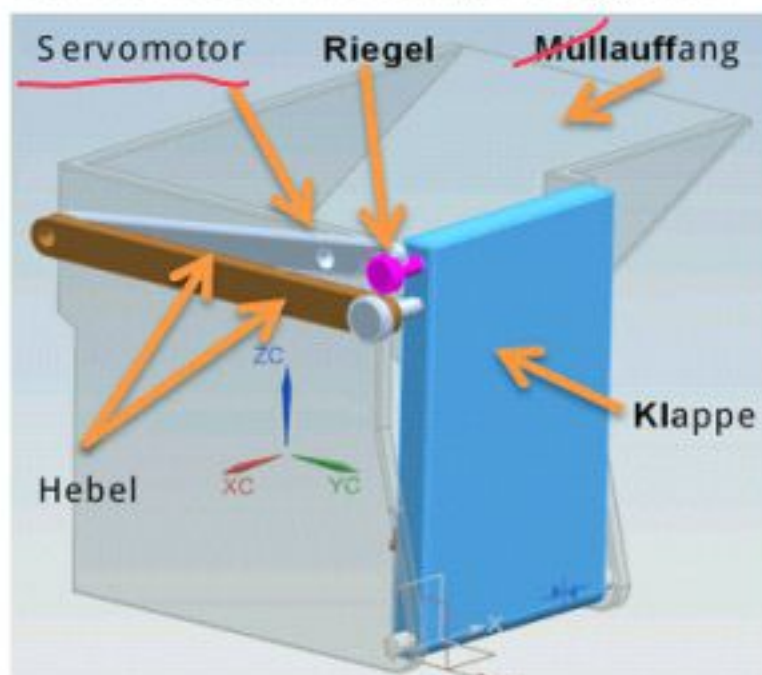


Abbildung 4: Mulde geschlossen

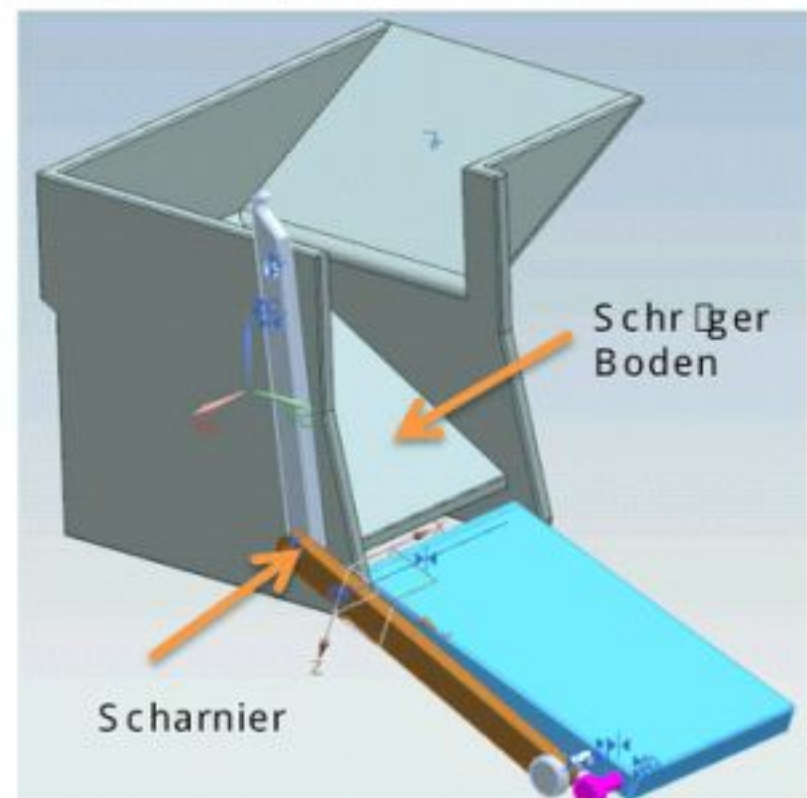


Abbildung 3: Mulde geöffnet

Führung?

denselben Servomotor. Der Riegel ist über ein Scharnier mit der Klappe verbunden. Dieses ist ähnlich konstruiert wie eine Klappenschere. Dadurch wird die zum Schließen benötigte Kraft auf **ca. 0.175 N** eingespart. Die Mulde selbst wird 3D-gedruckt. Für die Klappe kann auch Plexiglas verwendet werden.

Fehlt noch: restliche Beschreibung Bilder

5.3.4. Begründung

Diese Variante erlaubt eine schnelle Entleerung. Im Gegensatz zu einer Gewindespindel ist dieses Konzept mit jedem Fahrwerk kombinierbar. Durch die Klappe wird sichergestellt, dass der Müll im Entsorgungsbecken landet und nicht zwischen Fahrzeug und Becken zu Boden fällt.

*Berechnungen Momente, Gewichte,
Geschwindigkeiten ...*

werthaltig hier?

5.4. Sensoren

Bei der Auswahl der Sensoren wurde der Fokus auf das Sortiment der Firma Tinkerforge gelegt. Das bedeutet, dass das Auslesen der Sensoren sich einfacher gestaltet, da die Firma eine gut dokumentierte API-Schnittstelle für verschiedene Programmiersprachen anbietet.

5.4.1. Farbsensoren

Mit dem Farbsensor soll ausgewertet werden, ob es sich bei dem Container, der sich auf der rechten Seite des Fahrzeuges befindet, um den "richtigen" handelt. Der "richtige" Container, muss von blauer oder grüner Farbe sein. Zudem wird mit dem Farbsensor ausgewertet, wann sich der Container vor dem Hebearm befindet. In der Tabelle unten werden verschiedene Eigenschaften des Sensors TCS34725 aufgezeigt.

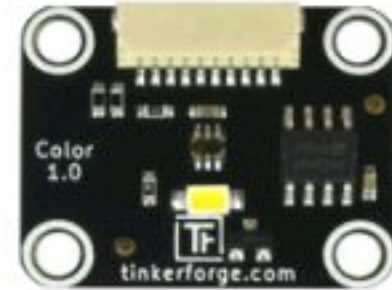


Figure 1: Color-Bricklet

Eigenschaft	Wert
Sensor	TCS34725
Stromverbrauch	0,2mA (LED aus), 5mA (LED an)
Dynamikbereich	3800000:1 <i>→ Helligkeit?</i>
Auflösung Farbe (R,G,B,C)	jeweils 16Bit (0-65535)
Auflösung Farbtemperatur	16Bit (0-65535)
Auflösung Helligkeit	16Bit (0-65535)
Abmessungen (B x T x H)	25 x 20 x 5mm (0,98 x 0,79 x 0,19")
Gewicht	2g

Table 1 - Eigenschaften Color Bricklet

5.4.1.1. Funktionsweise

Der Farbsensor wird durch eine API gesteuert. Diese stellt zum Beispiel die "color_get_color"-Funktion zur Verfügung. Die gemessene Farbe wird von ihr als RGB-Value zurückgeliefert. Um bei verschiedenen Lichtverhältnissen noch gute Messwerte zu erhalten, können die Werte der Integration-Time und dem Gain-Value eingestellt werden. Zudem enthält der Print eine SMD-LED, die auch durch die API an- und abgeschaltet werden kann.

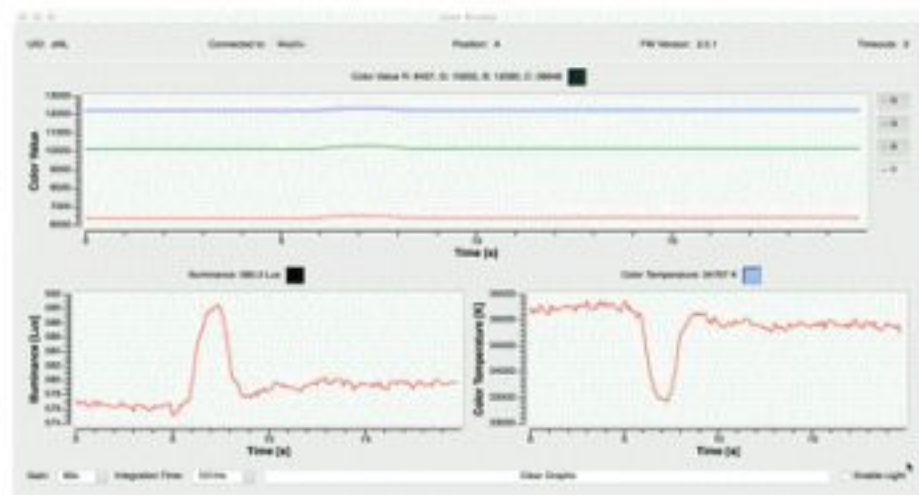


Figure 2: Brick Viewer mit Messwerten des Color-Bricklets

Messaufbau, Abstand, Umgebungshelligkeit etc. ..

5.4.1.2. Test

Beim Testen des Sensors wurde festgestellt, dass bei Umgebungslicht der Sensor die Farbe auf einen Abstand von ca. 2cm noch genau anzeigt. Eine mögliche Lösung, um die Reichweite des Sensors zu erhöhen, besteht in der Verwendung zusätzlicher LEDs für die Beleuchtung.

Für die Auswertung wurde der Sensor mit den verschiedenfarbigen Containern auf immer größer werdende Distanzen

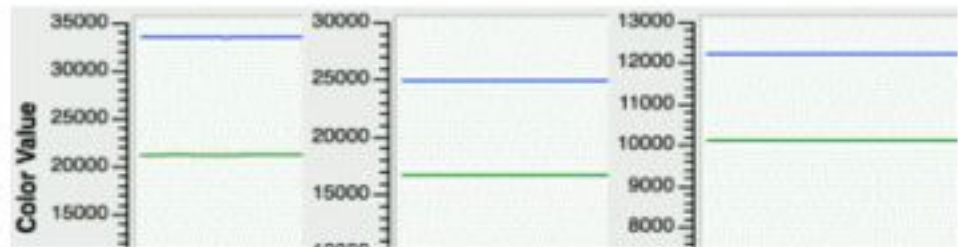


Figure 3: Messwerte des Color-Bricklets für verschiedene Distanzen. (2, 4, 6cm)

getestet. Die gemessenen Distanzen waren 2, 4 und 6cm. Für die Messung wurde der Color-Bricklet mit zwei zusätzlichen weissen LEDs ausgestattet. In der Abbildung #REF sind die verschiedenen Messwerte für die ändernde Entfernung aufgeführt.

Fazit?

5.4.2. Ultraschallsensor

Mit dem Ultraschallsensor soll erkannt werden, ob sich ein Fahrzeug an der Kreuzung befindet. Mit dem verwendeten Sensor können Entfernungen zwischen 2cm und 400cm messen. Die gemessene Entfernung wird als einheitenloser Wert ausgegeben, nicht in mm.

Dies liegt daran, dass das Verhältnis von gemessenem Entfernungswert zu wirklicher Entfernung vom exakten Wert der 5V Versorgungsspannung abhängt. Abweichungen in der Versorgungsspannung führen zu Differenzen in den gemessenen Entfernungswerten.



Figure 4: Das Distance US-Bricklet

Eigenschaft	Wert
Sensor	HC-SR04
Stromverbrauch	8mA
Entfernungen	2cm - 400cm, 12Bit Auflösung
Messwinkel	15°
Aktualisierungsrate	40Hz
Abmessungen (B x T x H)	45 x 20 x 30mm (1,78 x 0,78 x 1,18")
Gewicht	13g

Table 2 - Eigenschaften Ultraschallsensor

5.4.2.1. Funktionsweise

Der Ultraschallsensor kann wie der Farbsensor über eine API angesprochen und konfiguriert werden.

Genau: wie antworten
wie messen?

Fazit, Messung?

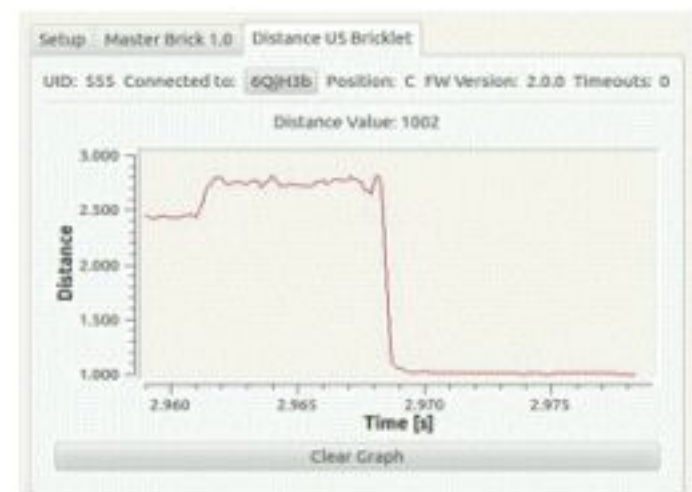


Figure 5: Messwerte des Ultraschallsensors

5.5. Energieversorgung

5.5.1. Akku

Anhand der Konzeptfindung wurde entschieden, dass ein Lithium-Polymer Akku eingesetzt wird. Für die Dimensionierung wurden alle Verbraucher aufgelistet und ihre Leistungen berechnet. In unserer Anforderungsliste ist als Festanforderung eine Akkulaufzeit von mindestens 8 Minuten definiert. Als Wunschzeit wurden 30 Minuten definiert. Für die Berechnung der Akkukapazität wurde eine Akkulaufzeit von 20 Minuten angenommen.

Komponente	Leistung [W]	Zeit t in h
Raspberry Pi 1	5	0.35
Raspberry Pi 2	5	
Antriebsmotor	60	
Servo Lenkung	5	
Servo Greifer	5	
Servo Heber	5	
Servo Mulde	5	
Sensoren	5	
Webcam	7	
	102 W	

Strom 7.29 A @14V Akkuspannung

Akkukapazität 2550.00 mA/h

Tabelle 1 - Leistungen der Komponenten

Für die meisten Komponenten gab es keine Leistungsangabe. Meist musste diese selbst gerechnet oder gar geschätzt werden. Auch werden nicht alle Komponenten die ganze Zeit voll belastet, was eine geringere Leistungsaufnahme zur Folge hat. Zum Beispiel werden die Servo für den Greifer, Heber und Mulde nie gleichzeitig mit dem Antriebsmotor laufen. Anhand diesen Berechnungen und Überlegungen muss der Akku mindestens 20 Minuten halten.

5.5.2. Akkuüberwachung

Lithium-Polymer Akkus sind sehr empfindlich gegenüber Tiefentladung. Deshalb muss der Akku überwacht werden. Sobald die Zellspannung unter 3.6V sinkt, muss er abgeschaltet werden. Diese Funktion wird eine Ladeüberwachungsschaltung mit OPV übernehmen. Die Schaltung ist im Bild unten ersichtlich. Mit dem Potentiometer kann die gewünschte Schaltspannung eingestellt werden. Sobald die Akkuspannung unter $4 \cdot 3.6V = 14.4V$ sinkt, wird ein digitales High Signal am GPIO Eingang des Raspberry Pi angelegt. Dieser gibt den Alarm weiter an das zweite Raspberry Pi. Beide Rechnerwerke führen noch Sicherungen durch und fahren herunter.

wie 14.4V detektiert? A/D 7-Diode oder?

→ siehe Abb.

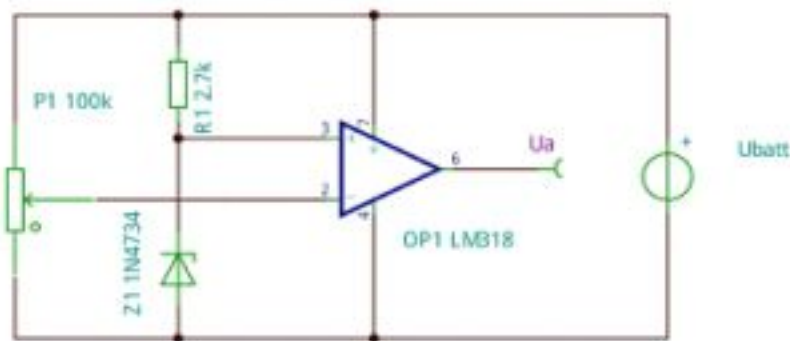


Abbildung 5 - Schaltung Akkuüberwachung

Weshalb?

5.5.3. Spannungswandlung

Der Akku liefert bei voller Aufladung 16.8V. Bei spätestens 14.4V muss der Akku wieder aufgeladen werden. Somit ist die Eingangsspannung zwischen 14.4 bis 16.8V. Aus dieser Spannung muss die 5V Speisespannung für das Raspberry und die 7V Spannung für die Servos generiert werden. Dies wird mit Buck-Convertern gemacht. Es gibt jeweils für die 5V Spannung und die 7V Spannungen eine Buck-Converter Schaltung.



Abbildung 6 - Blockschaltbild Spannungen

Lüftung?

Die Buck-Converter haben einen sehr hohen Wirkungsgrad und sind preiswert. Die Dimensionierung einer solchen Schaltung ist jedoch sehr aufwendig. Deshalb wurde das Webench-Tool von Texas Instruments eingesetzt. Mit diesem Tool kann der Eingangsspannungsbereich, die Ausgangsspannung und der Ausgangsstrom vorgegeben werden und das Tool gibt verschiedene Schaltungsvorschläge. Die daraus resultierenden Schemas sind im Anhang ersichtlich.

Fazit...

5.6. Hardware und Software

5.6.1. Produktiv Aufbau

In Abbildung 7 sind alle Hardware-Module als rechteckige Kasten dargestellt. Die Software, welche auf den jeweiligen Hardware-Modulen laufen, sind als blaue Rechtecke gekennzeichnet. Die Grafik gibt einen Überblick über den Aufbau des IT-Systems des Roboters.

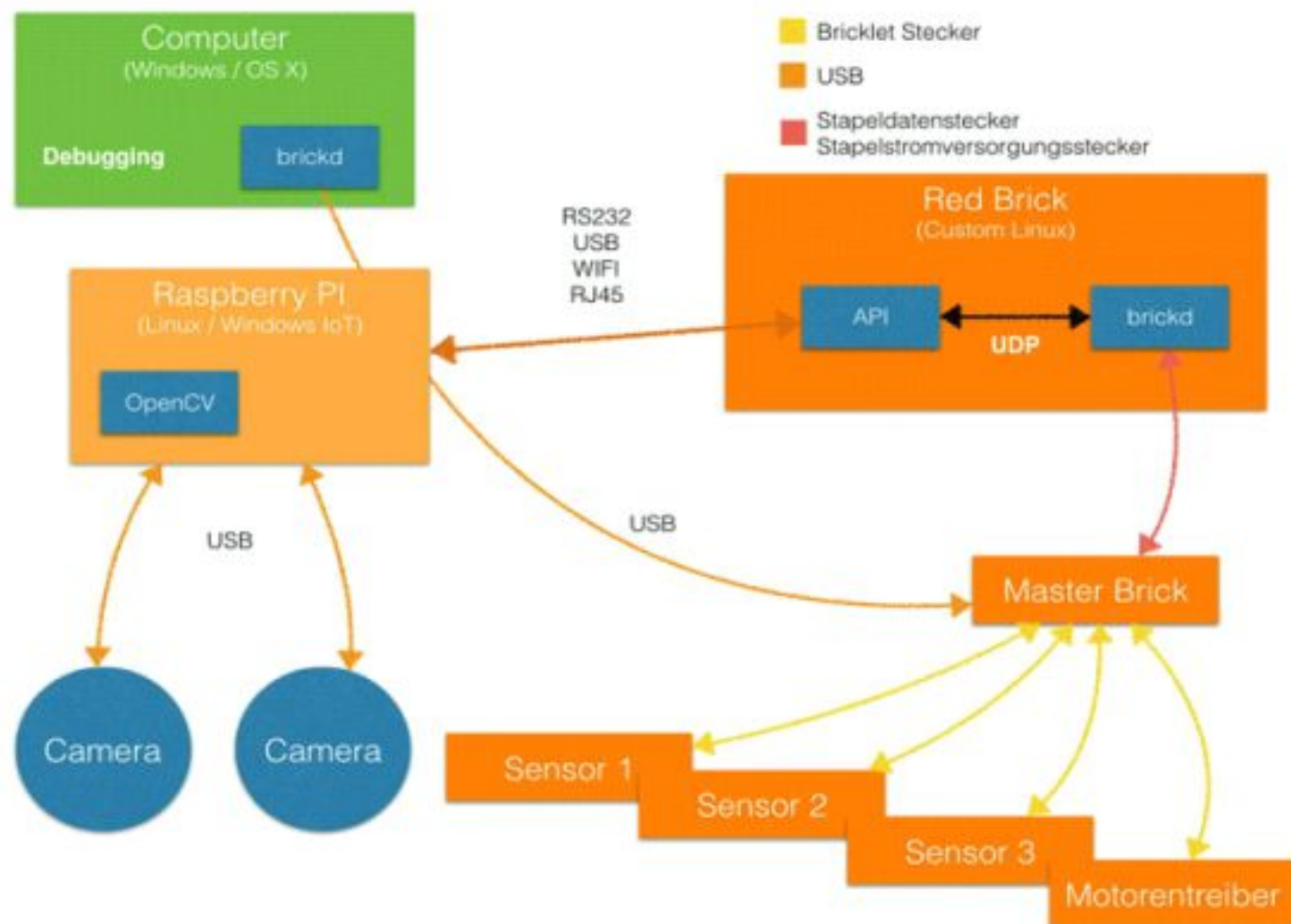


Abbildung 7: Übersicht der Hardware und Software für den Roboter

Für den produktiven Aufbau werden, wie in der Abbildung 7 dargestellt, zwei Computer benutzt. Das Raspberry Pi 2 wird benutzt, um die Bilder der angeschlossenen Kamera auszuwerten. Mit diesen Informationen wird das Fahrzeug gesteuert. Diese Steuerung wird durch die Software auf dem RED Brick abstrahiert. Dadurch müssen die Sensoren nicht direkt vom Raspberry Pi 2 angesteuert werden. Die beiden Software-Komponenten, welche auf dem RED Brick und dem Raspberry Pi 2 laufen, kommunizieren miteinander über eine TCP-Verbindung.

Am RED Brick sind alle Sensoren und Aktuatoren angeschlossen. Die Software (in Abbildung 7 als API bezeichnet) welche auf dem RED Brick läuft, benutzt die API-Bindings, welche von Tinkerforge zur Verfügung gestellt werden. Mit Hilfe dieser Bindings ist es möglich, aus der API-Software heraus die Sensoren und Aktuatoren über einen Daemon (brickd) anzusprechen.

5.6.2. Kommunikation

Um genügend Rechenleistung für das Auswerten der Kamera und das Steuern der Sensoren und Aktoren zu haben, werden zwei Raspberry Pis eingesetzt. Somit werden auch zwei eigenständige Programme laufen. Ein Programm wird für die Auswertung der Kamera und der Steuerung des gesamten Ablaufs zuständig sein. Das zweite Programm wird für die Steuerung der Sensoren und Aktoren zuständig sein.

Die beiden Programme werden über eine Netzwerkverbindung untereinander Daten austauschen. Damit dies möglich ist, wurde ein Kommunikationsprotokoll definiert.

zu knapp, welche Funktionen, threads, python? ...
→ siehe folg. Kap.

Test o b m Kap. Vorw!

1.1.1.1. Protokoll

Message ID	Message Type
Timestamp	
Payload	

Message ID: Inkrementierender Wert (Integer)
 Message Type: ID für die Art der Nachricht (#REF Kapitel) (Integer)
 Time Stamp: Wann die Nachricht abgeschickt wurde (Unix Time Stamp)
 Payload: Je nach Art der Nachricht verschieden

Size?

1.1.1.2. Message Types

Der Message Type ist aus einer Zahl mit 4 Ziffern aufgebaut.

Board Identifier	Command Group Identifier	Command ID	
0-9	0-9	0-9	0-9

Board Identifier: Identifiziert das Board (Raspberry Pi) welches die Nachricht gesendet hat.
 Command Group Identifier: Jede Nachricht gehört zu einer Gruppe. (Drive, Garbage Collection, Garbage Disposal, System State Messages)
 Command ID: Identifiziert das Kommando.

1.1.1.3. Messages

Insert messages here

1.1.1.4. Payloads

Insert Payloads here

5.6.3. Entwicklung

Während der Entwicklung der Software ist es wichtig, eine einfache und gute Möglichkeit zum Debuggen zu haben. Für diesen Zweck unterscheidet sich der Entwicklungsaufbau grundlegend vom Produktivlaufbau. Der Aufbau ist in Abbildung 8 aufgezeigt.

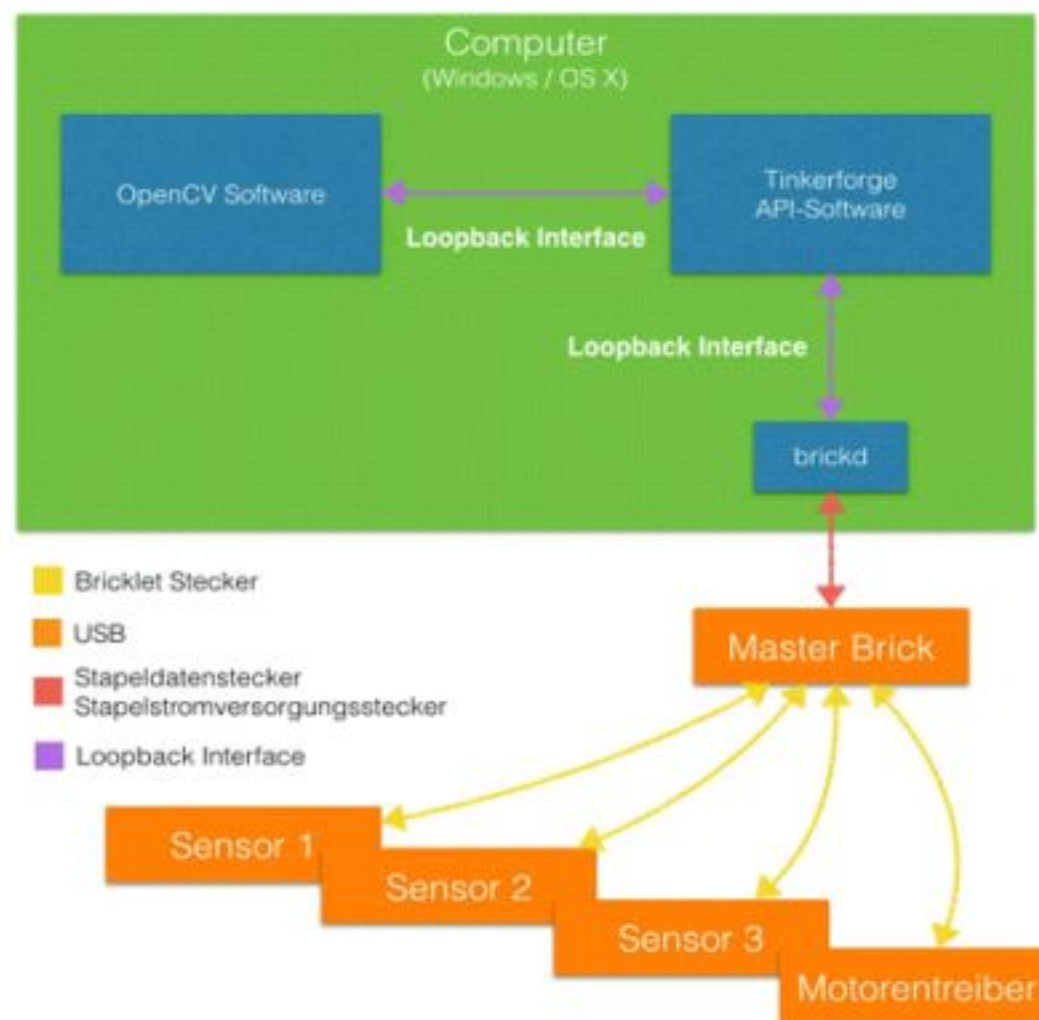


Abbildung 8: Aufbau der Hardware und Software während der Softwareentwicklung

Um die Software zu entwickeln, lässt man beide Software-Komponenten, welche normalerweise auf das Raspberry Pi und den RED Brick aufgeteilt würden, auf demselben Computer laufen. Um die TCP-Verbindung zwischen den beiden Komponenten zu simulieren, benutzt man das Loopback-Interface. Die Bricks von Tinkerforge sind dafür gedacht, vom Computer aus gesteuert zu werden. Daher ist es kein Problem, auch den brickd-Deamon auf dem Computer laufen zu lassen. Das in Abbildung 8 als "OpenCV Software" bezeichnete Rechteck steht für die Softwarekomponente, welche die Auswertung der Kamerabilder vollzieht.

5.6.4. Deployment

Für das Deployment soll eine Anleitung geschrieben werden. Damit soll sichergestellt werden, dass alle nötigen Schritte immer exakt gleich ausgeführt werden. Um allfällige Störungen im produktiven Betrieb nachvollziehen zu können, wird eine Software geschrieben. Diese zeigt während dem Betrieb des Roboters eine Übersicht der Sensorwerte an und kann ein Logfile schreiben.

5.6.5. Produktiver Betrieb

Im produktiven Betrieb läuft die Software jeweils auf dem Raspberry Pi 2 und dem RED Brick. Als Überwachung kann sich ein Computer via Wifi mit dem Raspberry Pi 2 verbinden und in einer Software die Sensorwerte und andere Daten darstellen. Zusätzlich kann auf beiden Linuxboards eine Konsolenausgabe gemacht werden.

5.7. Motoren

5.7.1. Auswahl

In der Auswahl standen lange zwei verschiedene DC Motoren. In der untenstehenden Tabelle sind die Motoren einander gegenübergestellt:

	Modelcraft RB350050	Maxon RE30
Bauart	DC Motor	DC Motor
Getriebe	50:1	4.8:1
Encoder	Nein	Ja, 512 Puls/U
Betriebsspannung	12V	24V
Leistung	10W	60W
Drehmoment	5.39 Nm	85.6mNm
Dimensionen	106mm x 36mm	125mm x 30mm

Tabelle 2 - Vergleich DC Motoren

*WC Skulpt
Encoder?*

Ein sehr wichtiges Kriterium für die Regelung eines Motors ist die Rückführung. Beim Maxon Motor ist ein Encoder direkt auf die Motorenwelle geschraubt und liefert 512 Pulse/U. Der Modelcraft Motor ist nicht mit einem Encoder lieferbar. Encoder für die nachträgliche Montage sind grundsätzlich sehr teuer und passen kaum in das vorgeschriebene Budget. Weiter muss der Encoder zusätzlich mechanisch auf die Antriebswelle angekoppelt werden. Dies ist aus Platzgründen aufwendig. Deshalb fiel die Wahl auf den Maxon Motor mit integriertem Encoder.

5.7.2. Motorensteuerung und Regelung

Um mit dem Fahrzeug Positions- und Geschwindigkeitsgesteuert fahren zu können, wird eine Regelung benötigt. Dazu wird eine Sollgeschwindigkeit vorgegeben und die Geschwindigkeit mithilfe dem Encoder wieder eingelesen. Der Regelkreis ist in der untenstehenden Grafik sichtbar:

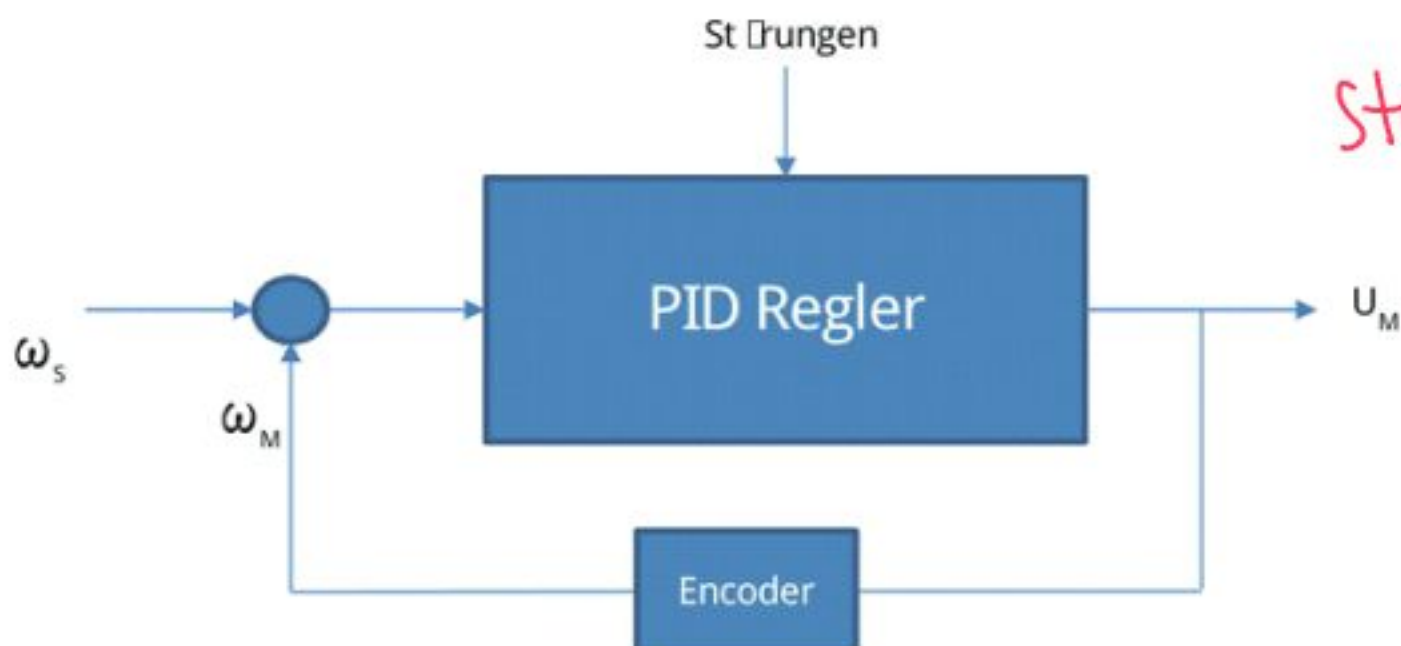


Abbildung 9 - Regelkreis

Sollgeschwindigkeit wird von unserer Steuerung vorgegeben und danach über den

PID-Regler geregelt. Wenn die Steuerung jedoch einen Container sieht, kann sie die Distanz abschätzen. Nun wird der Roboter positionsgesteuert gefahren. Das Fahrzeug fährt also eine definierte Strecke ab. Dies kann gemacht werden, indem die Pulse gezählt werden. Anhand der Berechnung unten ergeben sich 355 Pulse für eine Umdrehung.

$$1 \text{ Umdrehung} \cong \frac{U_{Rad}}{\text{Untersetzung}} = \frac{2 \cdot \pi \cdot 3.3cm}{3 \cdot 4.8} = 1.44cm$$

$$1 \text{ Umdrehung} = 512 \text{ Puls} = 1.44cm$$

$$\frac{1}{1.44cm} \cdot 512 = 355.6 \text{ Puls für } 1cm$$

5.7.3. Encoder

Der Encoder am Maxon Motor liefert 512 Impulse/Umdrehung. Diese Pulse müssen alle aufgezeichnet werden, damit genau gefahren werden kann. Dies wären bei einer Motorendrehzahl von 600 Umdrehungen (bei geschätzter maximalen Geschwindigkeit) pro Sekunde 300'000 Pulse pro Sekunde. Dies würde eine sehr grosse Abtastrate zur Folge haben. Deshalb wird ein Schnittstellen-IC verwendet. Dieses beinhaltet einen 32bit Counter, welche vom Encoder aufwärts/herunter gezählt wird. Nun kann mit dem Raspberry periodisch die Werte herausgelesen werden. Der Zeitpunkt dazu kann selber festgelegt werden, ohne dass Pulse verloren gehen.

6. Projektmanagement und Planung

6.1. Organigramm

6.2. Funktionsbeschreibung

6.3. Planung

7. Schlussdiskussion

7.1. Kosten

7.2. Lessons learned

7.3. Risiken → Projektmanag.

7.4. Offene Punkte

7.5. Ausblick auf PREN2

7.6. Fazit

8. Abbildungsverzeichnis

Abbildung 1: Provisorische Darstellung des Greiferarms	10
Abbildung 2: Beispiel einer Servolenkung	12
Abbildung 3: Mulde geöffnet.....	13
Abbildung 4: Mulde geschlossen	13
Abbildung 5 - Schaltung Akkuüberwachung	19
Abbildung 2 - Blockschaltbild Spannungen.....	19
Abbildung 7: Übersicht der Hardware und Software für den Roboter.....	20
Abbildung 8: Aufbau der Hardware und Software während der Softwareentwicklung....	22
Abbildung 3 - Regelkreis	23

9. Tabellenverzeichnis

Tabelle 1 - Leistungen der Komponenten	18
Tabelle 2 - Vergleich DC Motoren	23

Anhang?

10. Anhang