

# **Silisloth**

## **Konzept einer autonomen Laufkatze**

PREN Gruppe 7

2. Juli 2018

### **Versionierung**

Version	Datum	Bemerkung
0.1	Do, 26.10.2017	Aufsetzen des Dokuments
0.2	Do, 23.11.2017	Restrukturierung gemäss Vorlage
0.3	Fr, 01.12.2017	Management Summary, Analyse Aufgabenstellung, Komponenten
0.4	Sa, 02.12.2017	Überarbeitung, korrekte Referenzierung
0.5	Mi, 13.12.2017	Anhang, weitere Komponenten, Verweise
0.5	Do, 14.12.2017	Grobkonzept, CAD-Modelle, Fazit
1.0	Fr, 15.12.2017	Vorbereitung zur Testatabgabe
1.1	Di, 09.01.2018	Finale Überarbeitung gemäss Feedback

### **Betreuer**

Marcel Joss, Elektrotechnik

### **Autoren**

Sandro BERTOZZI, Informatik

Christoph BINKERT, Maschinenbau

Patrick BUCHER, Informatik

Alex DUONG, Elektrotechnik

Quentin FREI, Maschinenbau

Jan GREBER, Elektrotechnik

Marko LOVRINOVIC, Maschinenbau

Johannes TOGAN, Maschinenbau

## Teamvorstellung Gruppe 7

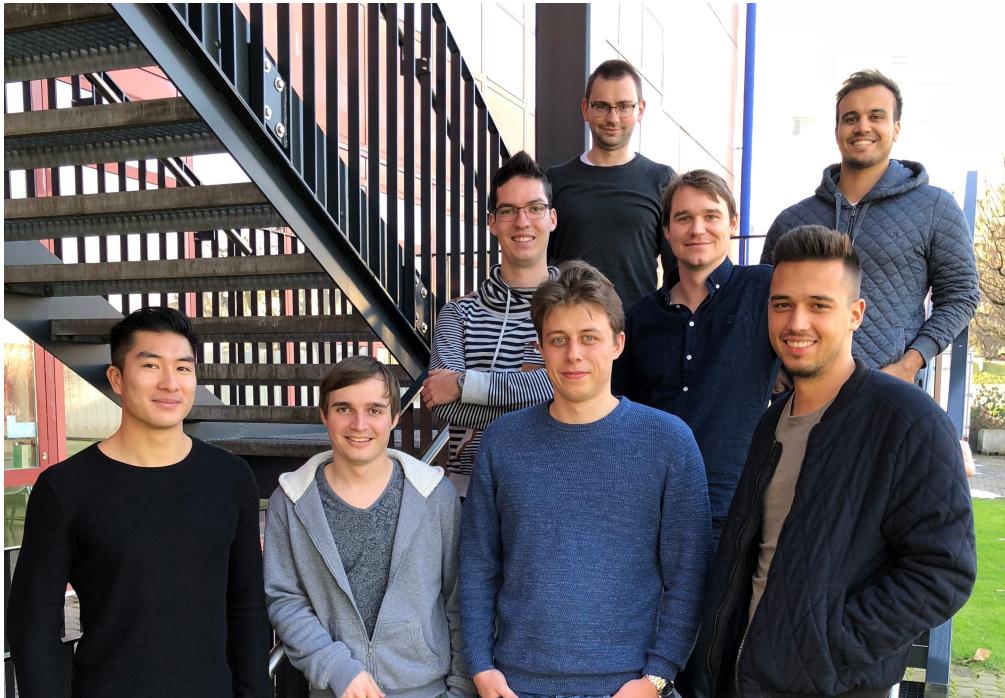


Abbildung 1: Gruppenfoto (von links nach rechts): Alex Duong, Christoph Binkert, Sandro Bertozzi, Quentin Frei (vorne), Patrick Bucher (hinten), Jan Greber, Marko Lovrinovic (vorne), Johannes Togan (hinten)

- Alex DUONG (Elektrotechnik)  
[alex.duong@stud.hslu.ch](mailto:alex.duong@stud.hslu.ch)
- Christoph BINKERT (Maschinenbau)  
[christoph.binkert@stud.hslu.ch](mailto:christoph.binkert@stud.hslu.ch)
- Sandro BERTOZZI (Informatik)  
[sandro.bertozzi@stud.hslu.ch](mailto:sandro.bertozzi@stud.hslu.ch)
- Quentin FREI (Maschinenbau)  
[quentin.frei@stud.hslu.ch](mailto:quentin.frei@stud.hslu.ch)
- Patrick BUCHER (Informatik)  
[patrick.bucher@stud.hslu.ch](mailto:patrick.bucher@stud.hslu.ch)
- Jan GREBER (Elektrotechnik)  
[jan.greber@stud.hslu.ch](mailto:jan.greber@stud.hslu.ch)
- Marko LOVRINOVIC (Maschinenbau)  
[marko.lovrinovic@stud.hslu.ch](mailto:marko.lovrinovic@stud.hslu.ch)
- Johannes TOGAN (Maschinenbau)  
[johannes.togan@stud.hslu.ch](mailto:johannes.togan@stud.hslu.ch)

## Management Summary

Dieses Dokument wurde ihm Rahmen des Moduls PREN 1 im Herbstsemester 2017 von der Gruppe 7 erstellt und beschreibt ein Konzept für eine autonome Laufkatze namens *Silisloth*, wie sie im Frühlingssemester 2018 im Rahmen des Moduls PREN 2 umgesetzt werden soll. *Silisloth* ist ein Gerät, das an einem Seil hängend nach erteiltem Startsignal eine Last lokalisieren, aufnehmen, über Hindernisse hinweg befördern und in einem durch konzentrische Rechtecke markierten Zielbereich absetzen soll. Das hier beschriebene Konzept erörtert Fragestellungen der Mechanik (Aufhängung, Greifmechanismus, Gehäuse), der Elektrotechnik (Sensorik, Stromversorgung, Mikrocontroller) und der Informatik (Bildverarbeitung, Entwicklerboard, Softwarearchitektur). Die einzelnen Komponenten sind darin genauer beschrieben; Tests mit Prototypen dieser Komponenten dokumentiert. Weiter beschreibt das Konzept die Kombination der einzelnen Komponenten zu einem autonomen Gerät. Das Konzept folgt der Maxime «eine sichere, zuverlässige und für alle verständliche Lösung [zu] erarbeiten» (siehe *Anhang I: Gruppenziele*), die dennoch originell und für alle Beteiligten lehrreich ist und zur Zusammenarbeit über die Fachgrenzen hinweg anregt.

## **Inhaltsverzeichnis**

<b>1. Einleitung</b>	<b>6</b>
<b>2. Analyse der Aufgabenstellung</b>	<b>7</b>
2.1. Pflichtenheft . . . . .	7
2.2. Funktionale Zerlegung . . . . .	7
2.3. Funktionales Blockschaltbild . . . . .	8
2.4. Zeitlicher Ablauf . . . . .	9
<b>3. Gesamtkonzept</b>	<b>11</b>
<b>4. Komponenten</b>	<b>14</b>
4.1. Maschinenbau . . . . .	14
4.1.1. Aufhängung . . . . .	14
4.1.2. Seilwinde . . . . .	15
4.1.3. Silikongreifer . . . . .	16
4.2. Elektrotechnik . . . . .	16
4.2.1. Motoren . . . . .	16
4.2.2. Ultraschallsensor . . . . .	17
4.2.3. Mikrocontroller . . . . .	20
4.2.4. Endschalter/Drucktaster . . . . .	20
4.2.5. Stromversorgung . . . . .	21
4.3. Informatik . . . . .	22
4.3.1. Entwicklerboard . . . . .	23
4.3.2. Betriebssystem . . . . .	24
4.3.3. Programmiersprache . . . . .	24
4.3.4. Kamera . . . . .	24
4.3.5. Bildverarbeitung . . . . .	26
4.3.6. Stromversorgung . . . . .	26
4.3.7. Smartphone-App . . . . .	27
4.4. Alternativen . . . . .	27
4.4.1. Mikrocontroller: Arduino/Freedom-Board . . . . .	29
4.4.2. Greifmechanismus: Silikongreifer/Elektromagnet . . . . .	30
4.5. Schnittstellen . . . . .	30
4.5.1. Entwicklerboard/Mikrocontroller . . . . .	32
4.5.2. Netzwerk . . . . .	33
4.6. Einschätzung des Konzepts gemäss Anforderungen . . . . .	34

## *Inhaltsverzeichnis*

<b>5. Projektmanagement und Projektplanung</b>	<b>38</b>
5.1. Risikomanagement . . . . .	38
5.1.1. Erläuterungen und Gegenmassnahmen . . . . .	38
5.2. Ressourcenbudget . . . . .	39
<b>6. Schluss</b>	<b>41</b>
6.1. Weiteres Vorgehen . . . . .	41
6.2. Lessons Learned . . . . .	41
6.3. Fazit . . . . .	42
<b>A. Projektauftrag</b>	<b>43</b>
<b>B. Recherche</b>	<b>43</b>
<b>C. Anforderungsliste</b>	<b>43</b>
<b>D. Konzeptvarianten</b>	<b>43</b>
<b>E. Detaillierte Berechnungen</b>	<b>43</b>
E.1. Reibung . . . . .	43
<b>F. Versuche</b>	<b>44</b>
F.1. Silikongreifer . . . . .	44
F.1.1. Herstellung . . . . .	44
F.1.2. Belastungstest . . . . .	46
F.2. Ultraschallsensor . . . . .	46
F.3. Elektromagnet . . . . .	48
<b>G. Meilensteinberichte</b>	<b>49</b>
<b>H. Protokolle</b>	<b>49</b>
<b>I. Gruppenziele</b>	<b>49</b>
<b>J. Projektplan</b>	<b>49</b>
<b>Quellen</b>	<b>50</b>
<b>Abbildungsverzeichnis</b>	<b>51</b>
<b>Tabellenverzeichnis</b>	<b>52</b>

## 1. Einleitung

Das vorliegende Dokument mit Anhängen wurde von der Gruppe 7 im Rahmen des Projektmoduls «Produktentwicklung» an der Hochschule Luzern – Technik & Architektur erstellt. Es stellt einerseits das Projekt-Zwischenergebnis nach dem Herbstsemester 2017 und dem ersten Modulteil PREN 1 dar, soll andererseits im Frühlingssemester 2018 im zweiten Modulteil PREN 2 als Grundlage für die Umsetzung des im Folgenden beschriebenen Geräts dienen: einer autonomen Laufkatze namens *Silisloth*. Der Name setzt sich aus «Silikon» und «Sloth» zusammen, denn *Silisloth* arbeitet wie ein Faultier kopfüber hängend und verfügt über eine wichtige Komponente aus Silikon.

Grundlage für das Projekt ist der im *Anhang A: Projektauftrag* angeführte Projektauftrag. Dieser beschreibt die Rahmenbedingungen der zu erstellenden Apparatur und des Wettbewerbs, an dem diese im Sommer 2018 antreten soll. *Silisloth* soll an einem Seil hängend eine Last in der Form eines Holzwürfels erkennen, aufnehmen, anheben, dem ansteigenden Seil entlang fahrend über zufällig positionierte Hindernisse hinwegführen und in einem markierten Zielbereich möglichst genau absetzen – und schliesslich zum Endmast fahren und anhalten. Der Vorgang muss in einem gesetzten Zeitrahmen abgeschlossen sein; auch die Vorbereitungszeit und Entwicklungskosten sind begrenzt.

Im ersten Teil wird der Projektauftrag analysiert (siehe *Abschnitt 2: Analyse der Aufgabenstellung, Seite 7*) und daraus eine Anforderungsliste (*Anhang C: Anforderungsliste*) abgeleitet. Das Gesamtkonzept (siehe *Abschnitt 3: Gesamtkonzept, Seite 11*) bietet einen Überblick über die Bauweise von *Silisloth*. In den weiteren Abschnitten (4.1: *Maschinenbau, Seite 14*; 4.2: *Elektrotechnik, Seite 16* und 4.3: *Informatik, Seite 22*) werden die einzelnen Komponenten erklärt. Diese wurden mithilfe einer Recherche (*Anhang B: Recherche*) ermittelt, teils in Versuchen (*Anhang F: Versuche*) erprobt und kombiniert zu verschiedenen Konzeptvarianten (*Anhang D: Konzeptvarianten*) evaluiert. Da die Wahl einiger Komponenten mit einem hohen Risiko verbunden ist (siehe *Abschnitt 5.1: Risikomanagement, Seite 38*), wurden Alternativkomponenten ermittelt (siehe *Abschnitt 4.4: Alternativen, Seite 27*).

Das Projektmanagement umfasst neben dem Projektplan (*Anhang J: Projektplan*) auch die Bereiche Risikomanagement (siehe *Abschnitt 5.1: Risikomanagement, Seite 38*) und Budgetplanung (siehe *Abschnitt 5.2: Ressourcenbudget, Seite 39*). Im Schlusswort (siehe *Abschnitt 6: Schluss, Seite 41*) wird neben dem weiteren Vorgehen und einem kurzen Fazit auf die «*Lessons learned*» eingegangen; *Silisloth* ist zwar das eigentliche Projektziel, der Lerneffekt aber der übergeordnete Zweck dieser Projektarbeit.

Die Gruppe 7 hat sich zu Beginn des Moduls PREN 1 Ziele gesetzt (siehe *Anhang I: Gruppenziele*), die bei der Entwicklung des hier beschriebenen Lösungsvorschlags mitberücksichtigt wurden. Dabei ist ein Vorschlag für eine «*sichere, zuverlässige und für alle verständliche Lösung*» (Ziel 5) entstanden, wobei durch die Wahl innovativer Komponenten der «*Spaß an der Arbeit*» (Ziel 2) nicht zu kurz kam, fachbereichübergreifend etwas Neues gelernt wurde (Ziel 3) und aufgrund einer sinnvollen Aufgabenverteilung sichergestellt wurde, dass «*jedes Gruppenmitglied seinen Beitrag leistet*». Dies soll auch für das Modul PREN 2 gelten, das die Gruppe 7 in gleicher Zusammensetzung bestreiten will.

## 2. Analyse der Aufgabenstellung

Die Aufgabenstellung (siehe *Anhang A: Projektantrag*) beschreibt die Rahmenbedingungen (Versuchsaufbau, Punktevergabe) des Wettbewerbs, an dem *Silislot* im Sommer 2018 teilnehmen soll. Durch eine ausführliche Analyse der einzelnen Anforderungen, die zu Beginn des Semesters vorgenommen wurde, konnte daraus ein Pflichtenheft abgeleitet werden.

### 2.1. Pflichtenheft

Das Pflichtenheft beschreibt, *was* das zu erstellende System können muss, nicht *wie* oder *womit* dies zu bewerkstelligen ist. Deshalb wurde das Pflichtenheft als verbale Wortkette formuliert, da diese Form kein Subjekt beinhaltet, und somit noch keine Annahmen über die Lösungskomponente enthält. Das Pflichtenheft umfasst folgende Punkte:

- das Startsignal erkennen
- sich an einem Seil fortbewegen
- die Last erkennen
- die Last aufnehmen
- die Last heben und senken
- das Zielfeld erkennen
- den Zielmast erkennen
- die Lastkoordinaten erfassen
- die Lastkoordinaten übertragen
- die Lastkoordinaten anzeigen

### 2.2. Funktionale Zerlegung

Nachdem geklärt wurde, was das System leisten soll, wurde abgeklärt, durch welche Komponenten, also *womit*, die jeweilige Teilaufgabe gelöst werden soll. Diese Komponenten wurden zunächst abstrakt formuliert, da die Recherche und Evaluation der tatsächlichen Lösungskomponenten erst in einer späteren Phase erfolgten. Die funktionale Zerlegung wurde eins zu eins anhand des Pflichtenheftes vorgenommen, indem den einzelnen Anforderungen ein Subjekt zugeordnet wurde:

- *die Startsignalerkennung* erkennt das Startsignal
- *der Fortbewegungsmechanismus* bewegt [Silislot] an einem Seil fort
- *die Lasterkennung* erkennt die Last
- *die Lastaufnahme* nimmt die Last auf
- *die Zielfelderkennung* erkennt das Ziel
- *die Zielmasterkennung* erkennt den Zielmast
- *die Lastkoordinatenerfassung* erfasst die Lastkoordinaten
- *die Lastkoordinatenübertragung* überträgt die Lastkoordinaten
- *die Lastkoordinatenvisualisierung* zeigt die Lastkoordinaten an

## *2. Analyse der Aufgabenstellung*

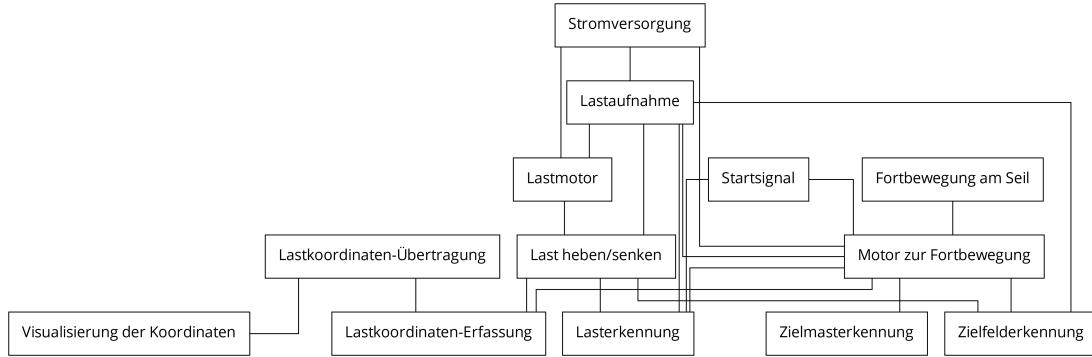


Abbildung 2: Funktionales Blockschaltbild

## 2.3. Funktionales Blockschaltbild

Zu den damit gefundenen (abstrakten) Lösungskomponenten wurde eine Recherche (siehe *Anhang B: Recherche*) durchgeführt. Aufgrund dieser Recherche konnten diese Lösungskomponenten konkretisiert und teils weiter aufgeteilt werden. Auch sekundäre Anforderungen (Stromversorgung) wurden dabei berücksichtigt. Es wurden folgende konkrete Lösungskomponenten ermittelt:

- Startignal
  - Fortbewegung am Seil
  - Motor zur Fortbewegung
  - Lasterkennung
  - Lastaufnahme
  - Lastmotor
  - Last heben/senken
  - Zielfelderkennung
  - Stromversorgung
  - Zielmasterkennung
  - Lastkoordinatenerfassung
  - Lastkoordinatenübertragung
  - Visualisierung der Koordinaten

Weiter wurde ermittelt, welche Komponenten miteinander interagieren oder sich gegenseitig beeinflussen. Aufgrund dieser Analyse konnte ein funktionales Blockschaltbild (siehe Abbildung 2, Seite 8) erarbeitet werden, wobei die einzelnen Lösungskomponenten die Knoten und die Schnittstellen dazwischen die Kanten repräsentieren.

Zu den einzelnen Lösungskomponenten wurden mögliche Lösungsvarianten gesucht, wobei die Recherche (siehe *Anhang B: Recherche*) als Grundlage diente. Die dabei gefundenen Lösungsvarianten wurden in einem morphologischen Kasten gesammelt. In einer Gruppensitzung wurden mit Hilfe dieses morphologischen Kastens die einzelnen Komponenten zu Konzeptvarianten kombiniert. Zu

## *2. Analyse der Aufgabenstellung*

diesem Zweck wurden vorher sogenannte Leitkriterien definiert, die vorgaben, nach welchem Kriterium die einzelnen Komponenten für ein Konzept ausgewählt werden sollten, beispielsweise «günstigste Lösung», «technisch anspruchsvollste Lösung» (siehe *Anhang D: Konzeptvarianten*).

Weiter wurde eine Kompromissvariante ausgearbeitet, die zwar auch technisch anspruchsvoll und originell sein sollte, jedoch einige problematische Lösungsansätze vermied. Diese Kompromissvariante schnitt in der Nutzwertanalyse am besten ab. Eine grobe Aufwandsschätzung ergab, dass diese Kompromissvariante für die verfügbaren Gesamtressourcen der Gruppe einerseits und für die Ressourcenverteilung auf die verschiedenen Fachbereiche verteilt andererseits am ehesten erfolgreich umgesetzt werden könnte. Diese Variante – fortan als *Silisloth* bezeichnet – wird im Weiteren genauer erläutert.

### **2.4. Zeitlicher Ablauf**

*Abbildung 3, Seite 10* stellt den zeitlichen Ablauf dar. Zum Zeitpunkt *Start* ist *Silisloth* bereits in seine Ausgangsposition gebracht worden.

1. *Silisloth* erwartet das Startsignal. Dies wird durch den Schiedsrichter erteilt. Sobald dies erfolgt ist, wird die Starttaste betätigt.
2. *Silisloth* fährt nun los und nähert sich der Last. Das Tempo bleibt konstant, solange die Last noch nicht erkannt worden ist.
3. Wurde die Last erkannt, kann diese aufgenommen werden. Dazu muss diese gegriffen und angehoben werden. Die Bewegung wird erst fortgesetzt, sobald die Last soweit wie möglich angehoben wurde, um keine unnötigen Schwingungen beim Anfahren zu erzeugen.
4. Die Last wird nun über die Hindernisse hinweg in Richtung Endmast zum Ziel transportiert. Sobald das Ziel erkannt wird, kann der Abstand zum Zielfeld berechnet werden. Befindet sich *Silisloth* exakt über dem Zielfeld, wird die Bewegung gestoppt und die Last abgesetzt.
5. *Silisloth* kann nun zum Ziel fahren. Die Greifapparatur muss dazu nicht erneut angehoben werden. Sobald *Silisloth* den Endmast berührt, kann die Bewegung gestoppt werden. Der Ablauf ist nun zu Ende.

2. Analyse der Aufgabenstellung

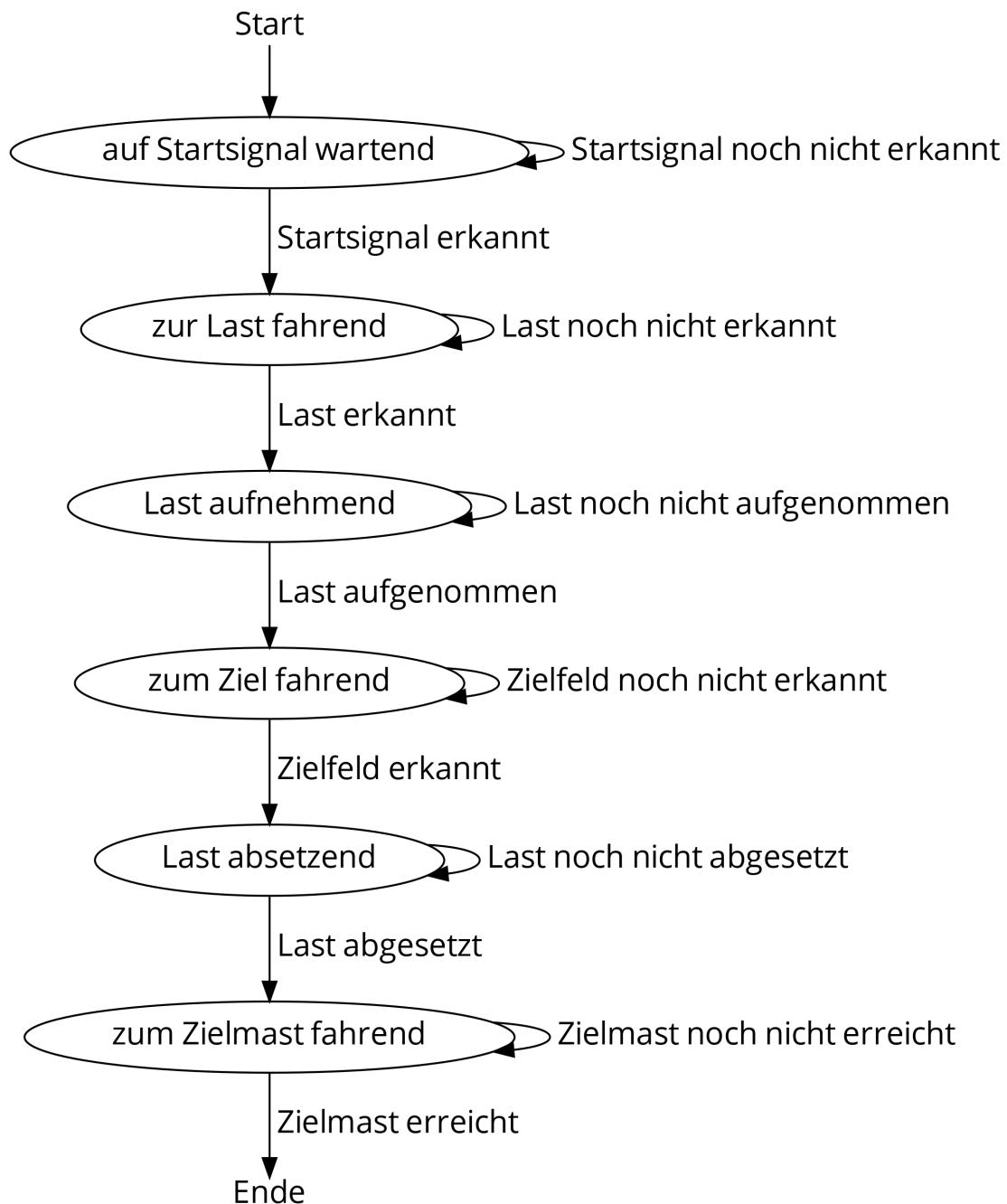


Abbildung 3: Der zeitliche Ablauf als Zustandsdiagramm

### 3. Gesamtkonzept

*Silisloth* besteht aus drei Baugruppen: Aufhängung (A), Kontrollbox (B) und Greifmechanismus (C). Der Aufbau von *Silisloth* ist in *Abbildung 4, Seite 12* (Ansicht links) und *Abbildung 5, Seite 13* (Ansicht rechts) als CAD-Modell ersichtlich.

In der Aufhängung wird das Antriebsrad von einem Getriebemotor (1) mit einer 1:1-Übersetzung angetrieben. Das Seil wird an den beiden Laufrädern (2, 3) und am Antriebsrad (4) umgelenkt. Die Kontrollbox ist mit einem Bolzen gelenkig mit der Aufhängung gelagert, damit der Greifer immer horizontal liegt. Der Greifmechanismus wird mit einem Flaschenzug (5) auf und ab bewegt. Das Seil für den Flaschenzug wird durch eine 3-Punkt-Umlenkung an einer Seiltrommel (6) aufgewickelt. Diese wird durch einen Stepper-Motor (7) mit einer Übersetzung von 1:2 angetrieben. Die Luftzufuhr für den Silikongreifer (8) erfolgt durch die Luftpumpe (9) und wird über das magnetische Ventil (10) geregelt.

Die Ultraschallsensoren (11, 12) sind so positioniert, dass keine Komponente von *Silisloth* während der Fahrt die Messdaten der Sensoren verfälschen könnte. Der erste Ultraschallsensor (11) misst die Distanz zum Boden, der zweite Ultraschallsensor (12) die Distanz zum Endmast.

Das Entwicklerboard (13) ist auf der rechten Seite angebracht. Die Akkus (14) und der Mikrocontroller (15) befinden sich im Inneren des Gehäuses. Vorne ist eine Kamera (16) angebracht.

3. Gesamtkonzept

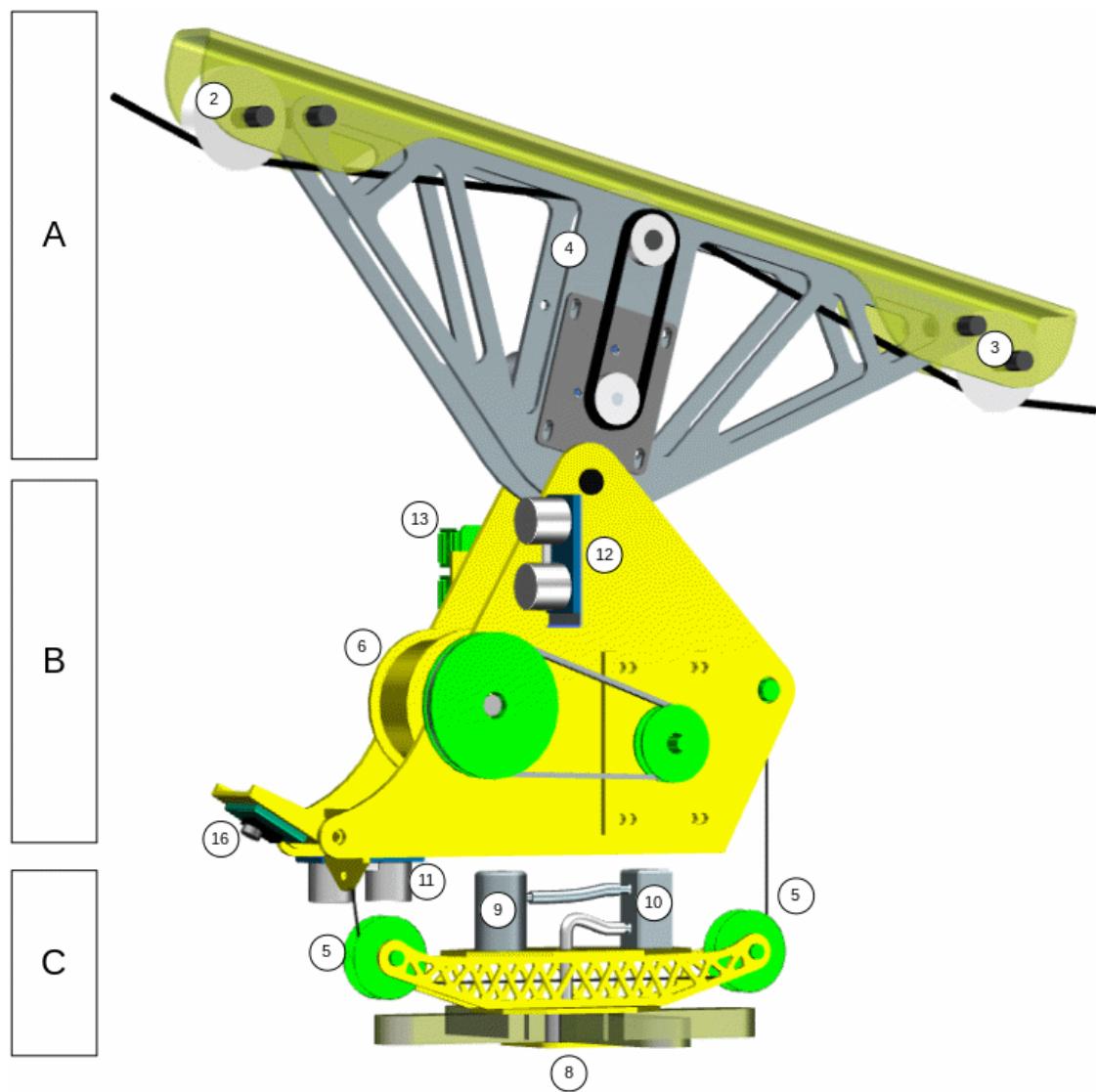


Abbildung 4: CAD-Modell von *Silisloth* (Ansicht links)

3. Gesamtkonzept

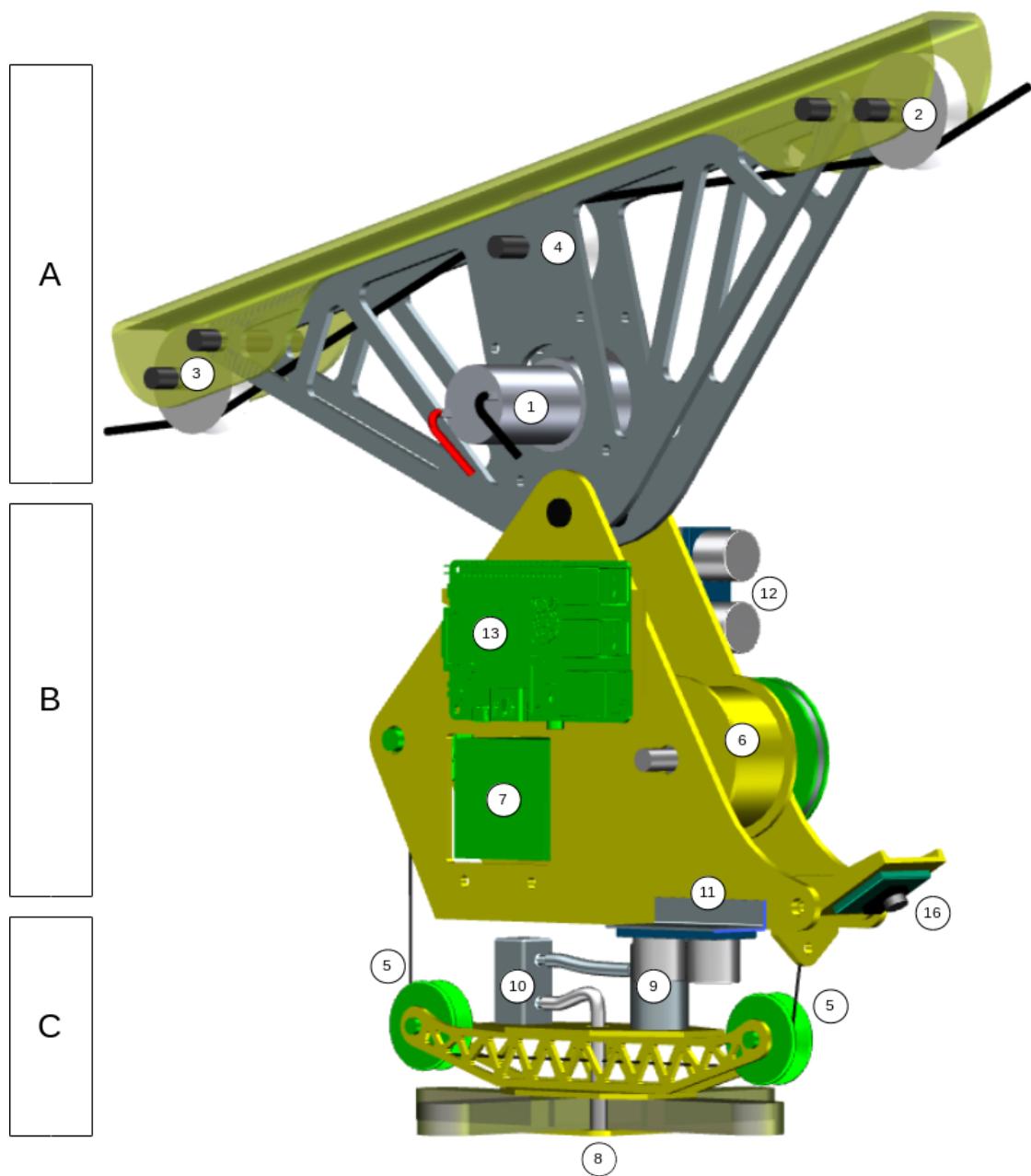


Abbildung 5: CAD-Modell von *Silisloth* (Ansicht rechts)

#### 4. Komponenten

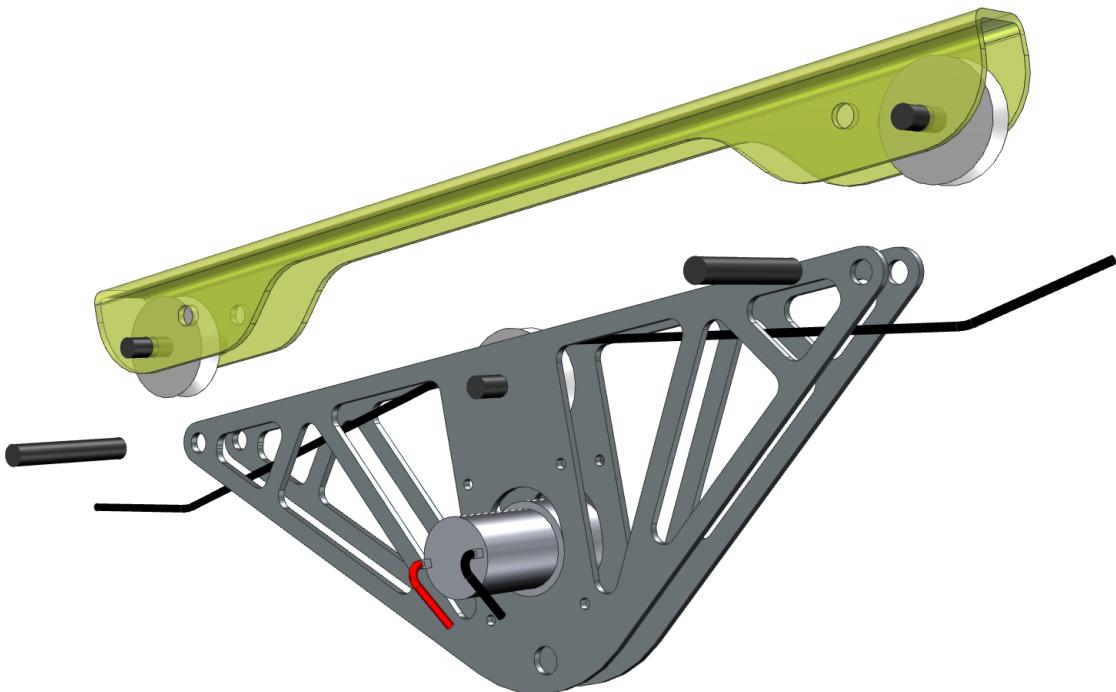


Abbildung 6: CAD-Modell der Aufhängung (Explosionsansicht)

## 4. Komponenten

Im Folgenden werden die einzelnen Komponenten genauer vorgestellt. Die Erläuterungen werden nach Fachbereich gegliedert.

### 4.1. Maschinenbau

Neben dem Gehäuse, das alle anderen Komponenten zusammenhält und so den Körper von *Silisloth* bildet, werden im Bereich Maschinenbau diverse mechanische Komponenten benötigt. Die Motoren, die Gegenstand der beiden Fachbereiche Maschinenbau und Elektrotechnik sind, werden hierbei der Elektrotechnik zugerechnet.

#### 4.1.1. Aufhängung

**Fortbewegung** Die Befestigung und Fortbewegung am Stahlseil wird durch Räder realisiert. Sie bestehen aus Kunststoff, sind 3D gedruckt und haben eine V-förmige Lauffläche. Diese Form führt dazu, dass mehr als nur die Gewichtskraft von *Silisloth* für die Reibung vorhanden ist. So mit kann auch bei einem kleinen Reibungskoeffizienten von 0.13 die Haftriebung garantiert werden, die eine reibschlüssige Fortbewegung am Seil ermöglicht. (Genauere Berechnungen

#### 4. Komponenten

werden in *Anhang E.1: Reibung* angestellt.) Das treibende Rad wird durch einen Getriebemotor, der mit einem Zahnriemen mit dem Rad verbunden ist, angetrieben. Durch den Synchronantrieb kann die momentane Position auf der Längsachse berechnet werden. Der Motor wird über eine Platte auf dem Fachwerk befestigt, damit die Vorspannung des Zahnriemens auch bei sich ändernder Seilsteigung gleich bleibt. Die Platte dient dazu, dass der Zahnriemen ge- und entspannt werden kann. Dies ist bei Zahnriemen nötig, um die geforderte Leistung übertragen zu können. Die Achsen und die Welle sind mit Rillenkugellagern gelagert, um geringe Reibungsverluste zu erzielen.

**Gerüst** Die Räder werden an einem Fachwerk-ähnlichen Gerüst befestigt (siehe *Abbildung 6, Seite 14*). Das Fachwerk bietet hohe Biege-Widerstände für wenig Material/Gewicht. Durch das Abstützen am Seil an zwei Orten entsteht ein Biegemoment im Gerüst. Jenes steigt, je weiter die beiden Räder voneinander entfernt sind. Damit ein Montieren am Seil innerhalb von zwei Minuten möglich ist, kann der Deckel der Aufhängung durch das Entfernen von zwei Stiften geöffnet werden. Das Fachwerk wird aus Aluminium lasergeschnitten und der Deckel 3D gedruckt. Die beiden tragenden Räder werden fertig im Deckel montiert, ebenso ist das Fachwerk mit dem treibenden Rad und dem Getriebemotor fest verbaut. Die beiden Aluminiumplatten des Fachwerks werden durch Distanzhülsen auf der gewünschten Breite gehalten.

**Schwingungsdämpfung** Schwingungen in Seilrichtung wird mit Dämpfern entgegengewirkt. Sie werden am Gerüst und am Hauptteil befestigt. Der Hauptteil ist durch einen Bolzen mit dem Gerüst verbunden. Somit liegt der Hauptteil immer waagerecht unter dem Seil. Damit kann garantiert werden, dass der Holzwürfel am gewünschten Ort auf- und abgeladen werden kann. Schwingungen um die Seilachse sollten nicht entstehen können, da sämtliche Kräfte direkt unter dem Seil angreifen. Sollten dennoch solche Schwingungen auftreten, sind sie klein und werden durch die Reibung am Seil gedämpft.

##### 4.1.2. Seilwinde

Zur vertikalen Bewegung des Greifers wird eine Seilwinde eingesetzt. Um das Haltemoment des Schrittmotors, welcher die Seiltrommel antreibt, nicht zu überschreiten, wird ein Flaschenzug mit dem Verhältnis 1:2 zur Anwendung kommen. Dies bringt zusätzlich den positiven Nebeneffekt, dass die Last nicht unkontrolliert umherpendeln kann. Die Seiltrommel sowie deren Antrieb sind am Gehäuse angebracht. Als Kraftübertragungsmittel ist ein Zahnriemen vorgesehen, welcher die Möglichkeit bietet, mit den Riemenrädern eine zusätzliche Unter- bzw. Übersetzung zu realisieren.

### 4.1.3. Silikongreifer

Ziel des Silikongreifers ist es, einen Holzwürfel mit einer Kantenlänge von ca. 5cm und einem Gewicht von 80g zu fassen. Der Greifer ist an einem Seil befestigt, wobei dieses mit der restlichen Vorrichtung verbunden ist. Die Greifvorrichtung verfügt über einen integrierten Luftkanal, welcher durch gezielte Luftzufuhr den eigentlichen Greifmechanismus auslöst. Durch ein Ventil an der Pumpe kann die Luft wieder abgelassen werden, wobei sich die Greifhand langsam öffnet und die Last loslässt. Die Herstellung des Silikongreifers ist in *Anhang F.1.1: Herstellung* ausführlich beschrieben.

## 4.2. Elektrotechnik

Der Bereich Elektrotechnik, der logisch zwischen dem Maschinenbau und der Informatik angesiedelt ist, umfasst im vorliegenden Projekt neben den bereits erwähnten Motoren auch deren Stromversorgung, einen Mikrocontroller und Sensoren. Die Stromversorgung des Mikrocontrollers ist Gegenstand der Informatik, da dieser über das Entwicklerboard gespeist wird.

### 4.2.1. Motoren

Zur Fortbewegung von *Silisloth* und für die Lastaufnahme werden Motoren benötigt. Für das Fahren am Seil kommt ein DC-Motor zum Einsatz, welcher über ein Getriebe *Silisloth* antreibt. Bei der Lastaufnahme werden zwei Motoren benötigt. Ein Motor bringt den Greifarm zur Last, während der zweite Motor den Greifer selbst antreibt. Um die Motoren mit genügend Strom zu versorgen, werden diese über einen Motorentreiber betrieben.

**Motorentreiber** Da *Silisloth* für den Betrieb drei Motoren benötigt, bietet sich das Motorshield von Adafruit (*Adafruit Motor Shield*, 2015) als Motorentreiber an. Über das Board können 4 Motoren unabhängig voneinander angesteuert werden. Jeder Ausgang liefert bis zu 600mA Strom (1.2A Spitze) bei 4.5 bis 25V Spannung (siehe *Abbildung 7, Seite 17*).

**Lastaufnahme** Die Lastaufnahme erfolgt über einen Silikongreifer, der mit Druckluft betrieben wird. Der Greifer wird als Einheit mit einer Miniluftpumpe (*Mini Air Pump DC 6V and DC 12V Motor 370 for Aquarium*, o.J.) zur Last gebracht. Die Luftpumpe wird von einem 6/12V-DC-Motor angetrieben und befördert ca. 1.8l/min bei einer Stromaufnahme von 300mA (siehe *Abbildung 8, Seite 18*). Um einen Anhaltspunkt für die Dimensionierung des Motors für die Seilwinde zu erhalten, wurde das Gewicht der Greifeinheit mit einer Last von 1kg grosszügig geschätzt. Die Greifeinheit wird über einen Flaschenzug zu ihrem Ziel gebracht. Als erstes wird die Solldistanz von einem Ultraschallsensor ermittelt, dann bringt der Schrittmotor die Einheit zu ihrem Einsatzort. Der Schrittmotor vom Typ Emis E547-52500 (*Schrittmotor Emis E547-52500*,

#### 4. Komponenten

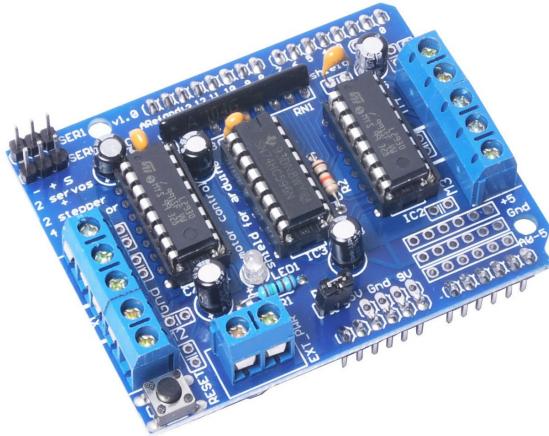


Abbildung 7: Motor Shield HP-ASH-L293D

o.J.) weist ein Haltemoment von  $0.25\text{Nm}$  auf und wird mit  $12\text{V}$  und  $600\text{mA}$  betrieben. Dabei wird eine Umdrehung in 200 Schritte unterteilt, was eine sehr genaue Platzierung der Greifseinheit ermöglicht (siehe Abbildung 9, Seite 18).

**Antrieb** Die Fortbewegung am Seil wird mit einem DC-Motor realisiert, der *Silisloth* über ein Getriebe antreibt. Da herkömmliche DC-Motoren viel zu hohe Drehzahlen aufweisen und passende Getriebe schwer zu finden oder zu teuer sind, fiel die Wahl auf den Getriebemotor Typ33-125 vom Hersteller Igarashi (*Getriebemotor 12 V Igarashi TYP 33G-125*, 1999), der die hohen Drehzahlen des DC-Motors mit einem integrierten Getriebe untersetzt. Der Igarashi Typ 33-125 wird mit  $12\text{V}$  und  $300\text{mA}$  betrieben und liefert  $0.3\text{Nm}$  Drehmoment bei  $38\text{U/min}$  (siehe Abbildung 10, Seite 19).

#### 4.2.2. Ultraschallsensor

Um zu jedem Zeitpunkt zu wissen, wo sich die aufgenommene Last befindet, werden zwei Ultraschallsensoren des Typs HC-SR04 (*Ultrasonic Ranging Module: HC-SR04*, 2010) verwendet (siehe Abbildung 11, Seite 19). Diese sind in der Lage den Abstand zu einem Objekt im Bereich von 3 bis  $400\text{cm}$  zu messen. Der eine Sensor misst dabei den Abstand zum Masten am Ende des Seils, während der andere nach unten gerichtet ist und die jeweilige Höhe der Last über dem Boden ermittelt. Der Ultraschallsensor arbeitet mit einer Spannung von  $5\text{VDC}$ , bei einer Stromaufnahme von  $< 2\text{mA}$ .

Eine Messung mit dem HC-SR04 läuft wie folgt ab (Plate, 2017): Über den Trigger-Anschluss wird eine Messung gestartet. Die fallende Flanke eines min.  $10\text{s}$  langen High-Impulses löst eine Messung aus (siehe Abbildung 12, Seite 19). Der HC-SR04 sendet daraufhin ein  $40\text{kHz}$ -Burst-Signal bestehend aus acht Impulsen aus. Danach geht der Echo-Ausgang sofort auf einen High-Pegel und wartet auf

#### *4. Komponenten*



Abbildung 8: Miniluftpumpe mit Motor



Abbildung 9: Schrittmotor Emis E547-52500

*4. Komponenten*



Abbildung 10: Getriebemotor Igarashi Typ33-125

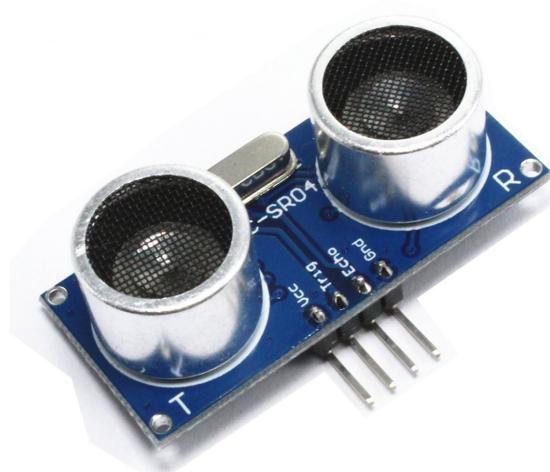


Abbildung 11: Ultraschallsensor HC-SR04

#### 4. Komponenten

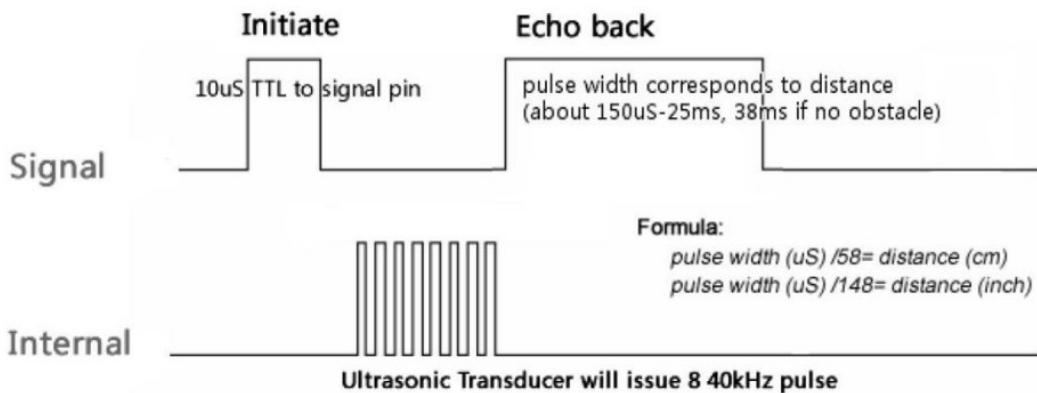


Abbildung 12: Ansteuerung des HC-SR04

das Empfangen eines Echos/Signals. Wird ein solches Echo detektiert, geht der Ausgang wieder auf Low. 20ms nach der Triggerung kann erneut eine Messung gestartet werden. Es können somit 50 Messungen pro Sekunde durchgeführt werden. Bleibt jedoch ein Echo aus, bleibt der Ausgang für 38ms auf High und signalisiert so eine Fehlmessung.

Die gemessene Entfernung ist proportional zur Echo-Puls-Weite und kann wie folgt berechnet werden (*Ultraschall-Sensor HC-SR04 und kompatible Ultraschall-Module, o.J.*):

$$\text{Distanz} = \frac{\text{EchoSignalLänge} * c}{2}, c \approx 340 \frac{\text{m}}{\text{s}} \quad (1)$$

Der Divisor 2 berücksichtigt, dass das Signal den doppelten Weg zurücklegt: zum Objekt hin und wieder zurück. Wird weiter die Schallgeschwindigkeit bei ca. 20°C verwendet, und die Echo-Signal-Länge in  $\mu\text{s}$  kann die Formel folgendermassen vereinfacht werden, um die gemessene Distanz in cm zu erhalten.

$$\text{Distanz} = \frac{\text{EchoSignalLänge}}{58}, \text{Distanz in cm} \quad (2)$$

Die besten Messergebnisse werden bei einer Reflexion an glatten, ebenen Objekten erzielt. Ab einer Entfernung von über einem Meter muss darauf geachtet werden, dass der Ultraschallsensor möglichst genau auf das zu messende Objekt ausgerichtet wird. Hindernisse, welche sich dann im Sendekegel (ca. 15°) befinden, können die Messung beeinflussen und somit verfälschen. Des Weiteren ist darauf zu achten, dass sich mehrere Ultraschallsensoren, welche nahe beieinander liegen, nicht gegenseitig beeinflussen. Dies und die Messgenauigkeit des Sensors wurden in einem Versuch erprobt (siehe *Abschnitt F.2: Ultraschallsensor, Seite 46*).

#### 4. Komponenten



Abbildung 13: Das Freedom-Board FRDM-KL25Z

##### 4.2.3. Mikrocontroller

Für die hardwarenahe Programmierung wird das Freedom Development Board FRDM-KL25Z (*Kinetis KL25 Sub-Family, 2014*) von NXP verwendet (siehe *Abbildung 13, Seite 21*). Dieses Mikrokontroller-Board ist besonders kostengünstig. Das Herz bildet ein 32-bit ARM Cortex-M0+-Prozessor mit einer Betriebsfrequenz von  $48\text{MHz}$ . Des Weiteren besitzt es diverse analoge und digitale Peripherie und mehrere UART-Schnittstellen.

Das Freedom-Board wird verwendet um die Motoren des Antriebs und der Lasthebung über Motorrentreiber anzusteuern. Zusätzlich kann es auch verwendet werden um die Messungen der Ultraschallsensoren auszuwerten und um das Signal des Endschalters zu empfangen.

Programmiert wird das FRDM-KL25Z über die integrierten Entwicklungsumgebungen (IDE) von Freescale, Kinetis Design Studio oder CodeWarrior. Als Programmiersprache kommt C zum Einsatz. Zusätzlich wird Processor Expert verwendet: ein Tool, welches Embedded-Komponenten verwendet um einfach Quellcode zu generieren.

##### 4.2.4. Endschalter/Drucktaster

Um sicherzustellen, dass *Silisloth* am Ende des Seils stoppt und den Antriebsmotor abschaltet, wird ein Mikroendschalter (*Mikroendschalter, o.J.*) verwendet (siehe *Abbildung 14, Seite 22*). Dieser wird an der Vorderseite des Geräts angebracht und betätigt, sobald er den Masten am Ende des Seils berührt. Die Signalleitung am Pin des Freedom-Boards wechselt dann von High auf Low und der Antriebsmotor wird angehalten.



Abbildung 14: Mikroendschalter

#### 4.2.5. Stromversorgung

Damit *Silisloth* seine Aufgabe erfüllen kann, muss es mit Strom versorgt werden. Um einen reibungslosen Ablauf zu garantieren, werden die Informatik- und Elektronikkomponenten von einer separaten Quelle über eine USV mit Powerboost versorgt. Die Motoren werden von einer anderen Quelle gespeist, um bei einem möglichen Spannungsabfall beim Anfahren keinen Neustart der Rechenkomponenten (Mikrocontroller und Entwicklerboard) zu riskieren.

**Akku für Mikrocontroller und Entwicklerboard** Das Entwicklerboard versorgt den Mikrocontroller und den Ultraschallsensor mit Strom. Das Entwicklerboard bezieht die Energie von einem Lithium-Polymer-Akku über eine USV mit Spannungshochsetzer (siehe *Abschnitt 4.3.6: Stromversorgung, Seite 26*). Die einzelnen Komponenten und ihr Energieverbrauch sind in *Tabelle 1: Berechnung der Akku-Kapazität für die Rechenkomponenten, Seite 22*, aufgelistet<sup>1</sup>.

**Akku für Motoren** *Silisloth* benötigt für seinen Betrieb drei Motoren: zwei für die Lastaufnahme und einen für den Antrieb. Die Motoren werden über ein Motorshield von einem Lithium-Polymer-Akku mit Energie versorgt. Der Energieverbrauch der einzelnen Motoren ist in *Tabelle 2: Berechnung der Akku-Kapazität für die Motoren und die Luftpumpe, Seite 23* aufgelistet.

**Berechnung der Akkukapazität** Aus den *Tabelle 1: Berechnung der Akku-Kapazität für die Rechenkomponenten, Seite 22* und *Tabelle 2: Berechnung der Akku-Kapazität für die Motoren*

---

<sup>1</sup>Das Freedom-Board wird über den Raspi gespiesen. Darum werden hier einige Angaben weggelassen.

#### 4. Komponenten

Komponente	Anzahl	Spannung [V]	Strom [A]	Leistung [W]	Laufzeit [s]	Energie [Wh]
Raspi mit Kamera	1	5	2.5	12.5	180	0.625
Freedom-Board	1	5	-	-	-	-
Ultraschallensor	2	5	0.015	0.075	180	0.00375
Total	4	5	2.515	12.575	360	0.62875

Tabelle 1: Berechnung der Akku-Kapazität für die Rechenkomponenten

Komponente	Anzahl	Spannung [V]	Strom [A]	Leistung [W]	Laufzeit [s]	Energie [Wh]
Getriebemotor	1	12	0.38	4.56	180	0.228
Schrittmotor	1	12	0.6	7.2	150	0.299
Mini-Luftpumpe	1	12	0.3	3.6	60	0.06
Total	3	12	1.28	15.36	490	0.58

Tabelle 2: Berechnung der Akku-Kapazität für die Motoren und die Luftpumpe

und die Luftpumpe, Seite 23 wird ersichtlich, dass die Rechenkomponenten 630mWh und die Motoren 587mWh (Angaben gerundet) Energie brauchen. Die Akkukapazitäten werden mit Formel 3, Seite 22 berechnet. Dabei wird bei den Motoren mit einer Spannung von 12V und bei den Rechenkomponenten mit 5V gerechnet. Die Berechnungen ergeben eine Mindestkapazität von 126mAh für die Rechenkomponenten, und für die Motoren ist ein Akku mit mindestens 49mAh Kapazität notwendig.

$$Q = \frac{E}{U}, Q = \text{Ladung in Ah}, E = \text{Energie in Wh}, U = \text{Spannung in V} \quad (3)$$

### 4.3. Informatik

In der Informatik ist das Entwicklerboard die zentrale Komponente. Dieses schränkt nicht nur die Wahl der Hard- und Softwarekomponenten (Kamera, Betriebssystem, Programmierungsumgebung) ein, sondern bedarf auch einer bestimmten Stromversorgung. Das Netzwerk ist dabei von besonderer Bedeutung. Da es sich hierbei nicht um eine eigentliche Komponente, sondern um ein geräteübergreifendes Kommunikationssystem handelt, wird dieses unter Schnittstellen (siehe Abschnitt 4.5.2: Netzwerk, Seite 33) beschrieben.



Abbildung 15: Raspberry Pi Model B Version 3

#### 4.3.1. Entwicklerboard

Als Entwicklerboard wird der Raspberry Pi Model B Version 3 (fortan als *Raspi* bezeichnet) verwendet. Mit seiner 4-Core-ARM-CPU mit einer Taktrate von  $1200MHz$  und  $1GB$  RAM (*Raspberry Pi Model 3*, 2016) bietet dieses Entwicklerboard genügend Leistung für rechen- und speicherintensive Aufgaben wie Bildverarbeitung und bleibt dank der Multicore-CPU dennoch auch bei hoher Last ansprechbar. Dieses Entwicklerboard bietet ein integriertes WiFi-Interface, womit das Konnektivitätsproblem gelöst wird (siehe *Abschnitt 4.5.2: Netzwerk*, Seite 33).

Es gibt auch andere Entwicklerboards, die ähnliche oder gar noch höhere Leistungsmerkmale aufweisen, diese sind aber wesentlich teurer als der Raspi. Die OpenCV-Library benötigt nicht nur viel Leistung bei der Ausführung, sondern auch viel persistenten Speicherplatz (ca.  $2GB$ ). Mit austauschbaren SD-Karten kann dieses Problem sehr gut gelöst werden. Es ist auch möglich, dasselbe Entwicklerboard mit verschiedenen Konfigurationen zu testen, indem einfach die SD-Karte ausgetauscht wird. Zum Einsatz kommen soll eine SD-Karte mit  $32GB$  Speicherkapazität. Dies genügt nicht nur für Betriebssystem mit den verschiedenen Entwicklertools (Python, OpenCV), sondern auch zur Speicherung verschiedenster Testdaten, die bei den Tests mit der Kamera und Bildverarbeitung entstehen (v.a. Video- und Bilddateien). Zudem bietet der Raspi ein Kamerainterface, womit die Raspi-Cam (siehe *Abschnitt 4.3.4: Kamera*, Seite 24) einfach angeschlossen und in Betrieb genommen werden kann.

#### 4.3.2. Betriebssystem

Die Wahl des Betriebssystems ist stark durch die Auswahl des Entwicklerboards eingeschränkt. Das Flaggschiff unter den Raspi-Betriebssystemen ist Raspbian – ein Debian-Derivat speziell für den Raspi. Debian zeichnet sich durch hohe Stabilität und weite Verbreitung (im Linux-Umfeld) aus und verfügt über 35000 Softwarepakete (*Raspbian FAQ*, o.J.) mit entsprechender Dokumentation dazu<sup>2</sup>.

<sup>2</sup>Tatsächlich umfasst Raspbian gegenwärtig über 37198 Softwarepakete (Stand 08.12.2017)

## 4. Komponenten

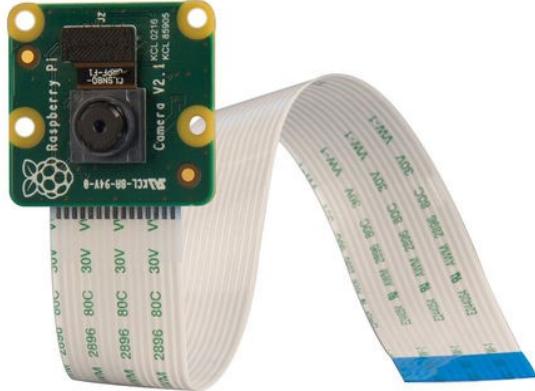


Abbildung 16: Das Raspberry Pi Camera Module V2

Raspbian gibt es in zwei Varianten: Lite und Desktop. Zwar wäre eine grafische Oberfläche bei Experimenten mit der Bildverarbeitung sehr hilfreich, diese konsumiert aber sehr viele Ressourcen, was für den praktischen Betrieb von Nachteil ist. Aus diesem Grund soll Raspbian Lite verwendet werden.

### 4.3.3. Programmiersprache

Als Programmiersprache auf dem Raspi wird Python 3.5 verwendet (*Python. A Programming Language Changes the World*, 2017). Python hat den Vorteil, dass viele mächtige Bibliotheken dafür existieren. Mit diesen Bibliotheken können wichtige Kernfunktionen, wie zum Beispiel die Zielfelderkenntnis, sehr einfach implementiert werden. Für die Bildverarbeitung wird die Bibliothek OpenCV verwendet (*About OpenCV*, 2017). OpenCV bietet bereits ein Python-Interface an, womit die Bildverarbeitung viel einfacher möglich ist als mit anderen Sprachen. Des Weiteren ist der Python-Quellcode leicht lesbar und kann direkt auf dem Raspi editiert und ausgeführt werden. Zur Entwicklung genügt ein einfacher Texteditor. Da Python eine interpretierte Sprache ist, fällt der Kompilierungsschritt weg, was schnelle Entwicklungs- und Testzyklen erlaubt. Aufgrund der vielen genannten Vorteile wird auf dem Raspi die Programmiersprache Python verwendet.

#### 4.3.4. Kamera

Für die Erkennung der Last und des Zielfelds wird eine Kamera benötigt. Mit dieser Kamera ist es möglich, auf dem Raspi Bilder und Videos zu machen. Diese Aufnahmen werden analysiert um die Last und das Zielfeld zu erkennen. Für den Raspi existiert bereits ein geeignetes Kameramodul: das *Raspberry Camera Module V2* (*Camera Module V2*, o.J.)

Die Kamera eignet sich besonders dadurch, dass auf dem Raspi bereits ein direkter Anschluss vorhanden ist. Diese wird über ein Flachbandkabel (siehe *Abbildung 16, Seite 25*) direkt mit der passenden

#### 4. Komponenten



(a) Ein Foto vom Zielfeld

(b) Das Zielfeld mit erkannten Konturen

Abbildung 17: Das fotografierte Zielfeld (links) mit erkannten Konturen (rechts)

Schnittstelle mit dem Raspi verbunden. Die Kamera ist bereits für unter CHF 30.- erhältlich<sup>3</sup>. Die Kamera hat 8 Megapixel und eine maximale Auflösung von  $3280 \times 2464$  Pixel für Fotos. Bei Videos sind je nach Auflösung mehr oder weniger Bilder pro Sekunde möglich. Bei  $1920 \times 1080$  Pixel sind 30 Bilder pro Sekunde möglich. Reduziert man die Auflösung auf  $1280 \times 720$  Pixel, sind es bereits 60 Bilder pro Sekunde.

Mittels dieses Kameramoduls wurden in der Verifizierungsphase Aufnahmen gemacht um zu überprüfen, ob diese Kamera geeignet ist. Abbildung 17a, Seite 25 zeigt ein Bild des Zielfelds, welches vom Raspi über das Kameramodul gemacht wurde.

Da in der Aufnahme das Zielfeld gut erkennbar und die Kamera bereits sehr günstig erhältlich ist, wird für das Projekt die offizielle Raspberry-Kamera verwendet werden.

#### 4.3.5. Bildverarbeitung

Für die Erkennung des Lastwürfels und des Zielfelds wird die freie Library OpenCV (*About OpenCV*, 2017) verwendet. OpenCV wurde unter anderem für die Programmiersprache Python geschrieben, durch welche *Silisloth* gesteuert wird. Um OpenCV verwenden zu können, muss zuerst die aktuelle Version auf den Raspi heruntergeladen werden. Anschliessend wird OpenCV auf dem Raspi kompiliert und installiert. Aufgrund der enormen funktionalen Möglichkeiten, die OpenCV bietet, entsteht nach dem Kompilieren ein Buildverzeichnis von über 2GB. Nachdem die Installation erfolgreich durchgeführt worden war, wurde ein erster Test zur Erkennung des Zielfelds über OpenCV unternommen. Dazu wurde ein Bild verwendet, welches bereits bei der Verifizierung des Kameramoduls entstanden ist und dadurch direkt über den Raspi und das dazugehörige Kameramodul gemacht wurde.

<sup>3</sup><https://www.pi-shop.ch/raspberry-pi-kamera-module-v2>

#### 4. Komponenten

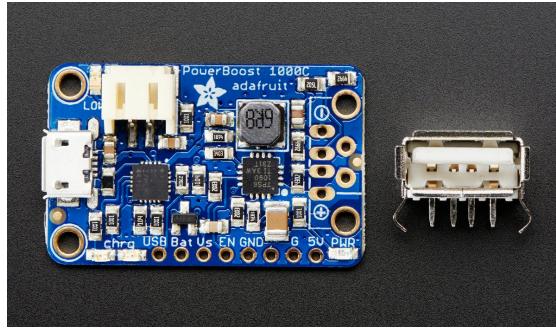


Abbildung 18: USV Adafruit PowerBoost 1000C

Als erstes wird das Bild mit OpenCV geladen. Anschliessend wird das komplette Bild verarbeitet, wobei per OpenCV alle Konturen markiert werden, die weiss oder schwarz sind. Das komplette Python-Skript, welches die Bilderkennung durchführt, ist gerade einmal 14 Zeilen lang. Die Konturenerkennung ist nur deshalb mit so wenigen Zeilen möglich, da das Finden und Zeichnen dieser Konturen über die OpenCV-Library erfolgt. Das Ergebnis der Zielfelderkennung ist auf *Abbildung 17b, Seite 25* zu sehen.

#### 4.3.6. Stromversorgung

Wie für das Netzwerk gilt auch bei der Stromversorgung, dass sie in der Entwicklungsphase per Kabel erfolgen kann, diese Möglichkeit aber im Test- und Wettbewerbsbetrieb nicht mehr besteht. Darum muss der Raspi über einen Akku betrieben werden. Ein Raspi bezieht je nach Rechenlast (und angeschlossenen Peripheriegeräten) zwischen 1A und 2.5A Strom bei einer Spannung von 5V. Ein Akku mit 3000mAh sollte bis zu 72 Minuten (siehe *Berechnung 4, Seite 26*) und somit bei Weitem für die Wettbewerbssituation ausreichen, bei der *Silisloth* während nur weniger Minuten vom Stromnetz getrennt werden muss.

$$t = \frac{3000mAh}{2.5A} = 1200 \frac{mAh}{A} = 1.2h = 72\text{min} \quad (4)$$

Damit der Raspi während der Einrichtungsphase nicht neu gestartet werden muss, was aufgrund sporadisch durchzuführender Startup-Tasks (Dateisystem-Integritätsprüfung, Aktualisierung des Dateiindex etc.<sup>4</sup>) oder unverhofft auftretenden Probleme zu Verzögerungen führen kann, soll eine USV (unterbruchsfreie Stromversorgung) zum Einsatz kommen (siehe *Abbildung 18, Seite 27*). Diese verfügt über drei Anschlüsse:

1. USB-Mini B (links): Hier wird die USV über ein Netzteil an das Stromnetz angeschlossen.

---

<sup>4</sup>Derlei Aufgaben könnten natürlich deaktiviert werden, was jedoch zeitaufwändig ist und zu Fehlern führen könnte. Schliesslich gibt es einen Grund dafür, warum gewisse Prüfungen beim Aufstarten gelegentlich ausgeführt werden.

#### *4. Komponenten*

2. Akku (oben): Hier wird ein Lithium-Polymer-Akku angeschlossen.
3. USB-Anschluss (rechts, noch nicht aufgelötet): Hier wird der Raspi per Kabel angeschlossen.

Die USV funktionierte zunächst nicht, da ein Netzteil mit einer Stromstärke von  $2.5A$  verwendet wurde. Zwar wurde der Akku geladen und entladen, beim Ein- und Ausstecken des Netzteils kam es aber jeweils zu einem Neustart des Raspis. Mit einem Netzteil der Stromstärke  $1A$  funktionierte die USV tadellos, sprich unterbrechungsfrei.

#### **4.3.7. Smartphone-App**

Mit einer Smartphone-App wird eine Verbindung zwischen Smartphone und *Silisloth* hergestellt. Dies geschieht über eine WiFi-Verbindung. Um *Silisloth* zu starten, wird eine Taste auf der App gedrückt. Ab jetzt läuft *Silisloth* völlig autonom. Sobald *Silisloth* dieses Startsignal empfangen hat, wird der Antrieb gestartet und der Lastwürfel gesucht. Wenn der Lastwürfel aufgenommen wurde, beginnt die App die aktuellen x- und z-Koordinaten des Würfels anzuzeigen. Durch eine konstante TCP/IP-Verbindung zwischen Smartphone und *Silisloth* werden diese Koordinaten laufend aktualisiert. Dies geschieht so lange, bis die Last wieder abgesetzt wurde. Auf *Abbildung 19, Seite 28* ist das User-Interface der Smartphone-App schematisch dargestellt:

#### **4.4. Alternativen**

Wie im Risikomanagement erörtert wird (siehe *Abschnitt 5.1: Risikomanagement, Seite 38*), stellen einige Komponenten bzw. die fehlende Erfahrungen damit eine Gefährdung des Projektziels dar. Für diese Komponenten sollen darum Alternativen gefunden werden, auf die im Problemfall zurückgegriffen werden kann.

##### **4.4.1. Mikrocontroller: Arduino/Freedom-Board**

Im Bereich Embedded Systems wurden zwei Mikrocontroller untersucht und verglichen: den Arduino Uno Rev3 (*Arduino Uno Rev3, 2017*), der auf *Abbildung 20, Seite 30* abgebildet ist, und das FRDM-KL25Z Freedom-Board von NXP. Arduino ist im Einsteiger- und Hobbybereich sehr beliebt und weit verbreitet. Mit dem Freedom-Board hingegen wird an der HSLU – Technik & Architektur gearbeitet. Beide Varianten haben Vor- und Nachteile (siehe *Tabelle 3: Vor- und Nachteile des Arduino Uno Rev3 gegenüber dem FRDM-KL25Z, Seite 29*).

Was die Performance und die nötige Peripherie betrifft, sind sich die beiden Mikrocontroller-Boards ebenbürtig. Die Aufgabenstellung ist sicherlich mit beiden zu bewältigen. Wie bereits erwähnt (siehe *Abschnitt 4.2.3: Mikrocontroller, Seite 20*) ist die Wahl auf das FRDM-KL25Z Freedom-Board gefallen. Da davon auszugehen ist, dass relativ viel Code geschrieben werden muss um die Motoren anzusteuern

*4. Komponenten*

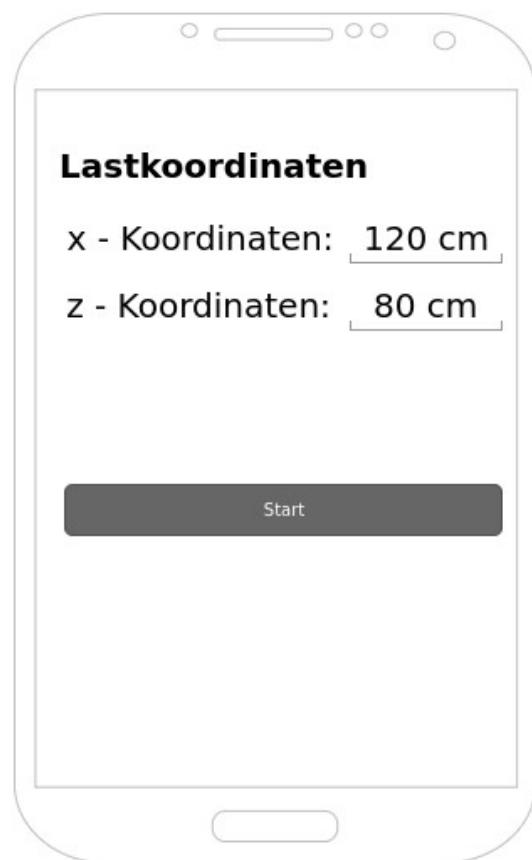


Abbildung 19: Das User-Interface der Smartphone-App (schematisch)

#### 4. Komponenten

	ARDUINO UNO REV3	FRDM-KL25Z
Vorteile	<ul style="list-style-type: none"> <li>• viele Codebeispiele und fertig implementierte Libraries online vorhanden</li> <li>• grosse, hilfreiche Community und Online-Blogs</li> <li>• vereinfachte Programmierumgebung</li> </ul>	<ul style="list-style-type: none"> <li>• Lerneffekt hoch</li> <li>• übersichtliche Programmierumgebung</li> <li>• Debugger vorhanden</li> <li>• Unterstützung durch HSLU-Dozenten vor Ort möglich</li> </ul>
Nachteile	<ul style="list-style-type: none"> <li>• online gefundener Code muss auf seine Funktionalität überprüft werden</li> <li>• kein Debugger vorhanden</li> <li>• Lerneffekt eher gering</li> <li>• begrenzte Übersichtlichkeit in der Programmierumgebung</li> </ul>	<ul style="list-style-type: none"> <li>• lange Einarbeitungszeit in Programmierumgebung nötig</li> <li>• weniger Codebeispiele online vorhanden</li> <li>• geringeres Know-How vorhanden</li> </ul>

Tabelle 3: Vor- und Nachteile des Arduino Uno Rev3 gegenüber dem FRDM-KL25Z

und um mit dem Raspi zu kommunizieren, wird die Übersichtlichkeit in der Programmierumgebung hoch gewichtet. Die lange Einarbeitungszeit und das fehlende Know-How wird im Austausch gegen den hohen Lerneffekt in Kauf genommen.

Sollten während der Umsetzung unerwartet Probleme mit dem Freedom-Board auftreten, welche nicht fristgerecht gelöst werden können, wird die Option freigehalten auf den Arduino zu wechseln. Der Wechsel auf den Arduino UNO oder Arduino Mega 2560 R3 kann problemlos garantiert werden, da alle verwendeten Komponenten auch zum Arduino kompatibel sind.

#### 4.4.2. Greifmechanismus: Silikongreifer/Elektromagnet

Für den Greifmechanismus musste die Entscheidung zwischen dem selbstgebauten Silikongreifer und einem gekauften Elektromagneten getroffen werden. Der Silikongreifer besteht aus zwei verschiedenen Komponenten, welche flüssig zusammengeführt wurden und in einer vorgedruckten Giessform aushärten konnten. Beim Elektromagneten kann eine Kraft von bis zu 300 Newton aufgenommen werden, was für die aufzunehmende Last mehr als ausreichend ist. Eine Gegenüberstellung von Silikongreifer und Elektromagneten (siehe *Tabelle 4: Vor- und Nachteile des Silikongreifers gegenüber dem Elektromagneten, Seite 32*) ergab, dass der Silikongreifer die besser geeignete Variante ist, der Elektromagnet aber eine verlässliche Alternative dazu darstellt.

Sollten in der Umsetzung unerwartete Probleme mit dem Silikongreifer auftreten, welche nicht fristgerecht gelöst werden können, wird die Option freigehalten auf den Elektromagneten zu wech-

#### 4. Komponenten



Abbildung 20: Arduino Uno Rev3

	SILIKONGREIFER	ELEKTROMAGNET
Vorteile	<ul style="list-style-type: none"> <li>• kleine Positionierungsfehler können aufgrund der grossen Spannweite noch während des Greifvorgangs ausgeglichen werden</li> <li>• nicht-magnetische Last kann angehoben werden</li> <li>• optisch interessant</li> <li>• innovativ</li> </ul>	<ul style="list-style-type: none"> <li>• einfach in der Anwendung</li> <li>• weniger Funktionsschritte</li> </ul>
Nachteile	<ul style="list-style-type: none"> <li>• viele und aufwändige Herstellungsschritte</li> <li>• braucht eine gesteuerte Luftversorgung</li> </ul>	<ul style="list-style-type: none"> <li>• Position beim Heben muss viel genauer sein als beim Silikongreifer, da der magnetische Teil der Last sehr klein ist</li> <li>• nur magnetische Teile können angehoben werden</li> </ul>

Tabelle 4: Vor- und Nachteile des Silikongreifers gegenüber dem Elektromagneten

#### *4. Komponenten*

seln. Der Wechsel auf den Elektromagneten kann problemlos garantiert werden, denn dieser wurde in einem Versuch (siehe *Anhang F.3: Elektromagnet*) erfolgreich getestet.

### **4.5. Schnittstellen**

Für die volle Funktionalität des Systems muss dieses in Komponenten aufgeteilt werden. Diese Komponenten erfüllen Teilaufgaben um die Gesamtaufgabe zu lösen. Dabei kommunizieren die Komponenten über Schnittstellen miteinander. *Abbildung 21, Seite 31* zeigt, wie die einzelnen Komponenten miteinander verbunden sind.

#### **4.5.1. Entwicklerboard/Mikrocontroller**

Die Hardware-Architektur zeigt, dass die Steuerung des Systems über den Raspi erfolgt. Dieses kommuniziert mit dem Freedom-Board KL25Z, um die Antriebsmotoren, den Motor zur Anhebung der Last und die Mini-Luftpumpen für den Silikongreifer anzusteuern. Dabei erfolgt die Verbindung des Raspi mit dem Freedom-Board über eine USB-CDC-Schnittstelle<sup>5</sup>, welche gleichzeitig das Mikrocontrollerboard mit Strom versorgt. USB CDC implementiert ein «Serial over USB»-Protokoll und ist ein einfacher und eleganter Weg um die Kommunikation der beiden Boards sicherzustellen.

Das übertragene Signal vom Raspi zum Freedom-Board enthält die Antriebsgeschwindigkeit und die Antriebsrichtung. Das Freedom-Board leitet diese Befehle an die Motoren weiter. Um die Höhe der Last zu erfahren, kann die aktuelle Position des Schrittmotors, der zum Anheben der Last benötigt wird, abgefragt werden.

Eine weitere Schnittstelle besteht zwischen dem Raspi und dem Smartphone. Die Kommunikation erfolgt hierbei über ein Wireless-Netzwerk. Vom Smartphone wird das Startsignal an den Raspi gesendet. In die Gegenrichtung werden fortlaufend die Koordinaten der aufgenommenen Last an das Smartphone gesendet. Somit erfährt der Anwender jederzeit die Position der Last über sein Smartphone mit der entsprechenden App.

Der Raspi selbst enthält neben der Steuerungslogik auch die Bildverarbeitung. Dazu wird mit einer angeschlossenen Kamera fortlaufend ein Bild des Untergrunds gemacht. Dadurch erkennt das System wann es sich über der Last oder über dem Zielfeld befindet. Sobald dies der Fall ist, wird der Schrittmotor angesteuert um den Greifarm zu senken. Das Öffnen und Schliessen des Greifarms wird ebenfalls vom Raspi aus gesteuert.

#### **4.5.2. Netzwerk**

Der Raspi verfügt nicht nur über eine Ethernet-Schnittstelle, sondern auch über einen integrierten WiFi-Chip. Ist die Ethernet-Schnittstelle bei der initialen Konfiguration sehr hilfreich, kommt beim

---

<sup>5</sup>Wie diese Verbindung hergestellt wird, ist in einem Blogeintrag (Styger, 2012) hervorragend erklärt.

#### 4. Komponenten

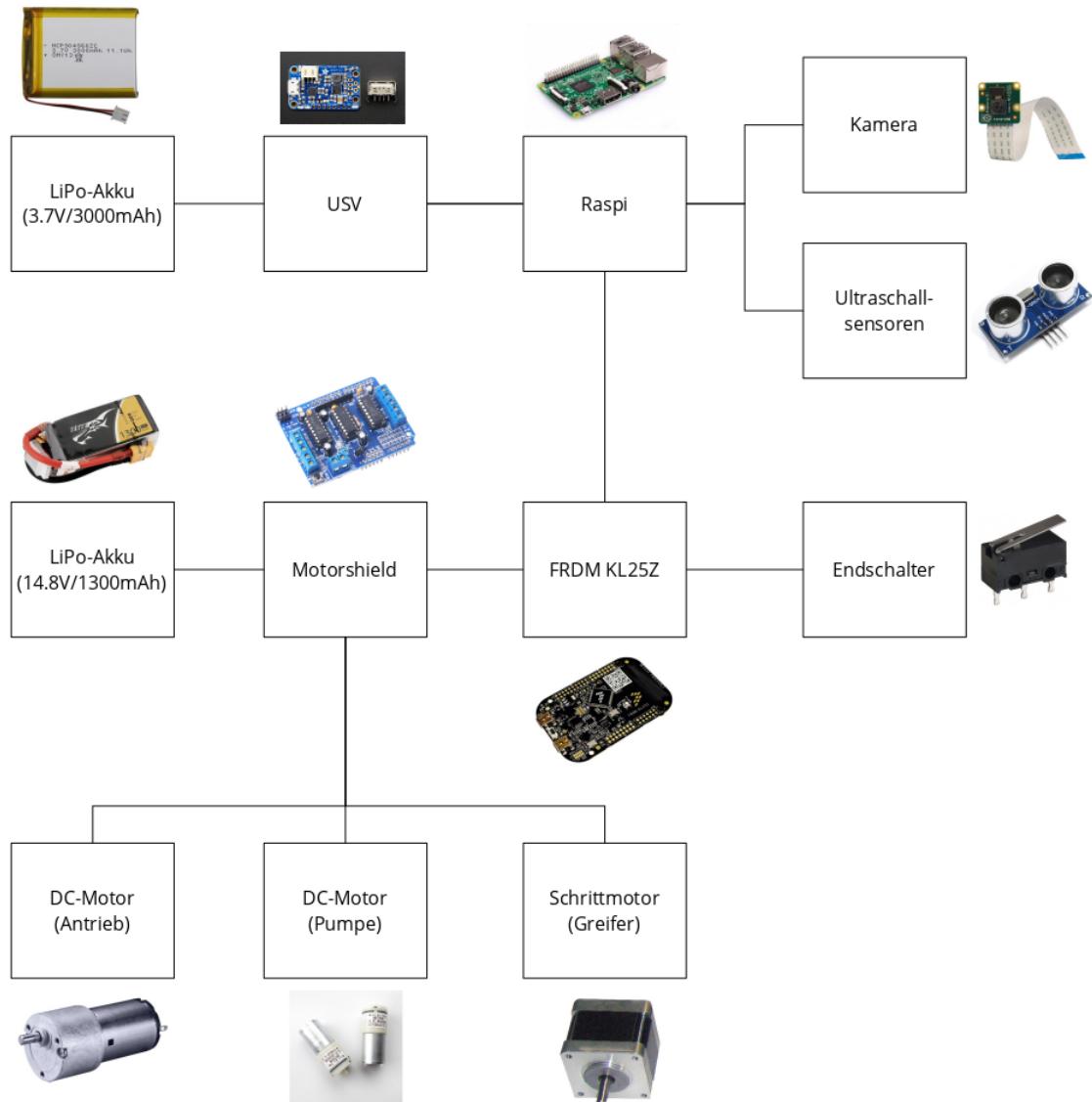


Abbildung 21: Die Hardware-Architektur: Komponenten und Schnittstellen

#### 4. Komponenten

späteren Betrieb von *Silisloth* nur eine kabellose Verbindung in Frage. Bei der Wahl des Netzwerkes bieten sich grundsätzlich zwei Möglichkeiten an:

1. Es wird direkt auf das HSLU-Netzwerk zugegriffen. In diesem Fall muss das Passwort eines der Gruppenmitglieder (verschlüsselt) der Netzwerkkonfiguration hinterlegt werden. Der Nachteil ist, dass das HSLU-WiFi – mit seinen tausenden von verbundenen Geräten<sup>6</sup> und den am Wettbewerbstag zu hunderten auf einen einzelnen Access Point zugreifenden Geräten – unter hoher Last steht und dadurch instabil werden könnte. Der Vorteil am HSLU-Netz ist jedoch, dass mit dem Raspi auf das Internet zugegriffen werden kann, was bei der Installation von Software sehr hilfreich ist. Die Variante HSLU-Netzwerk ist für die Entwicklungsphase vorteilhaft.
2. Es wird ein eigenes Netzwerk eingerichtet, entweder mit einem eigens dafür mitgebrachten Wireless Access Point – oder mit einem Gerät (Laptop oder Smartphone), das ein Ad-hoc Wireless-Netzwerk zur Verfügung stellt. Der Vorteil an einem dedizierten Wireless Access Point ist, dass dieser tendenziell stabiler und performanter arbeitet als Laptops<sup>7</sup> oder Smartphones, die einen Access Point mit einer einzigen Netzwerkkarte softwaremäßig umsetzen müssen. Dafür ist es mit einem einzigen Wireless Access Point nur bedingt möglich, auch aufs Internet zuzugreifen. Für den Test- und Wettbewerbsbetrieb dürfte der dedizierte Wireless Access Point aber die beste Lösung sein.

Die Kommunikation mit dem Raspi funktioniert über ssh. Dateien können per scp vom und auf den Raspi hin und her kopiert werden. Wird das HSLU-Netzwerk verwendet, ist die IP-Adresse des Raspi nicht konstant<sup>8</sup>. Es ist auch nicht möglich, im HSLU-Netz über den Hostnamen auf den Raspi zuzugreifen, da dieser nicht eindeutig ist. Aus diesem Grund wurde ein Skript entwickelt, das die IP-Adresse des Raspi anhand seiner (konstanten) MAC-Adresse ermittelt und diese sogleich dafür verwendet, eine Verbindung per ssh aufzubauen:

```
#!/bin/sh
mac='b8:27:eb:0b:44:f1' # enter your Raspberries MAC address here
ip=`sudo arp-scan -I wlp2s0 -l | grep -i "$mac" | head -n 1 | cut -f1` 
echo "$ip"
ssh pi@"$ip"
```

Hierbei handelt es sich um einen Aufruf des Programms ssh, der den Hostnamen per arp-scan ermittelt und die dabei gefundene IP von der restlichen Ausgabe herausfiltert. Ein Aufruf dieses Skripts

<sup>6</sup>Mittels ARP-Scan wurden an einem Freitagvormittag 2612 Geräte gefunden.

<sup>7</sup>Eine leicht modifizierte Variante der Software create\_ap wurde bereits erfolgreich getestet, wobei die eher schlechte Durchsatzrate etwas negativ aufgefallen ist.

<sup>8</sup>Es wurde aber schon die Beobachtung gemacht, dass ein Raspi am Morgen in Horw und am Nachmittag in Rotkreuz jeweils die gleiche IP zugewiesen bekam.

#### 4. Komponenten

ist zwar nicht besonders performant, die dabei ermittelte IP-Adresse ist aber für eine gewisse Zeit konstant und kann für weitere Aufrufe über Stunden wiederverwendet werden, wozu sie auch zusätzlich auf das Terminal ausgegeben wird.

### 4.6. Einschätzung des Konzepts gemäss Anforderungen

Die Anforderungsliste (*Anhang C: Anforderungsliste*) wurde zu Beginn des Projekts festgelegt und geht dem hier vorliegenden Konzept und den zuvor erarbeiteten Konzeptvarianten (*Anhang D: Konzeptvarianten*) voraus. Zwar war die Anforderungsliste während des ganzen Semesters gut sichtbar an einer Wand der Teaminsel angebracht, bei der Arbeit mit konkreten Komponenten und beim Finden (vermeintlicher) Lösungen können die teils einschränkenden Anforderungen schon einmal vergessen gehen. Darum soll das hier erarbeitete Konzept noch einmal kritisch darauf geprüft werden, ob es den Anforderungen auch wirklich gerecht wird. Dabei werden die Anforderungen in den ersten drei Spalten aufgeführt; in der vierten Spalte wird die Erfüllung der jeweiligen Anforderung gemäss dem Konzept eingeschätzt. Auf Wunsch- und Unteranforderungen wird dabei nicht eingegangen.

NR.	BEZEICHNUNG	DETAILS	EINSCHÄTZUNG GEMÄSS KONZEPT
<b>1 RAHMENBEDINGUNGEN</b>			
1.1	Aufbau	<i>Silisloth</i> lässt sich in max. zwei Minuten aufbauen.	Mit konzipierter Aufhängung und dank USV möglich
1.2	Autonomie	<i>Silisloth</i> und das I/O-Gerät arbeiten nach dem Startsignal autonom.	Nach Startsignal (per Smartphone-App) kein weiteres Eingreifen nötig
1.3	Temperaturbereich	Die Geräte sind in einem Temperaturfenster von $0^{\circ}\text{C}$ bis $70^{\circ}\text{C}$ einsatzfähig.	Mit konzipierten Komponenten möglich
1.4	Lichtverhältnisse	<i>Silisloth</i> funktioniert bei 1'000 – 100'000 lux.	Kamera passt Helligkeit nach wenigen Frames automatisch an
1.5	Zeitrahmen	<i>Silisloth</i> erledigt ihre Aufgaben innerhalb von vier Minuten.	Parameter wie Motorenleistung, Gewicht, Steigung, Rechenkapazität etc. sollten dies erlauben
<b>2 DIMENSIONEN</b>			
2.1	Länge	max. 480mm	400mm (CAD-Modell)
2.2	Breite	max. 480mm	200mm (CAD-Modell)
2.3	Höhe	max. 580mm	350mm (CAD-Modell)
2.4	Gewicht	max. 7'000g Leergewicht	4'000g (CAD-Modell, Herstellerangaben)
<b>3 ANTRIEB</b>			
3.1	Höhenüberwindung	<i>Silisloth</i> ist in der Lage eine Steigung von max. $40^{\circ}$ zu überwinden.	$45^{\circ}$ gemäss Berechnung möglich
3.2	Einsatzbereich	<i>Silisloth</i> kann auf einem Seil mit Durchmesser 2 – 4mm montiert werden.	Aufhängung für 2 – 4mm dicke Seile konzipiert

#### 4. Komponenten

3.3	Ziel	<i>Silisloth</i> stoppt nach Berührung des Endpfostens.	Drucktaster meldet Berührung
3.4	Geschwindigkeit	<i>Silisloth</i> bewegt sich durchschnittlich mit mindestens $15 \frac{mm}{s}$ .	ca. $30 \frac{mm}{s}$ möglich (gemäss Gewicht, Motorleistung, Reibung)
3.5	Fahrtrichtung	<i>Silisloth</i> ist in der Lage sich vorwärts und rückwärts am Seil zu bewegen.	Richtung per Motorentreiber steuerbar
4 LASTBEFÖRDERUNG			
4.1	Greifen	<i>Silisloth</i> kann eine quaderförmige Last mit den Kantenlängen von mindestens 45mm und höchstens 55mm und einem Gewicht von bis zu 200g greifen.	Test mit 50mm Kantenlänge und ca. 200g Last erfolgreich
4.2	Heben/Senken	<i>Silisloth</i> kann eine Last von bis zu 200g heben und senken.	per Schrittmotor geregelt
4.3	Abladen	<i>Silisloth</i> kann die Last auf dem Zielfeld absetzen.	Greifer wird per Luftventil gelöst
5 SENSORIK			
5.1	x-Koordinate	Die x-Koordinate muss mit einer Toleranz von $\pm 20mm$ bestimmt werden können.	Messgenauigkeit der Ultraschallsensoren: $\pm 20mm (> 100cm), \pm 2.5% (< 100cm)$
5.2	z-Koordinate	Die z-Koordinate muss mit einer Toleranz von $\pm 20mm$ bestimmt werden können.	Messgenauigkeit der Ultraschallsensoren: $\pm 20mm (> 100cm), \pm 2.5% (< 100cm)$
5.3	Zielerkennung	Das spezifizierte Zielfeld muss mit einer Toleranz von $\pm 20mm$ erkannt werden können.	noch nicht erprobt
5.4	Lasterkennung	Die Last muss mit einer Toleranz von $\pm 15mm$ erkannt werden können.	Erkennung noch nicht erprobt, doch der Silikongreifer hat eine Toleranz von $\pm 40mm$
6 KOMMUNIKATION			
6.1	Startsignal	<i>Silisloth</i> empfängt das Startsignal.	Per WiFi möglich
6.2	Koordinaten	<i>Silisloth</i> sendet die Koordinaten an das Ausgabegerät.	Per WiFi möglich
7 I/O-GERÄT			
7.1	Startsignal	Das Gerät sendet beim Start das Signal an <i>Silisloth</i> .	Per Smartphone-App möglich
7.2	Koordinaten	Das Gerät gibt die x- und z-Koordinaten der Last an.	Per Ultraschallsensoren und Smartphone-App möglich
8 AUSNAHMEBEHANDLUNG			
8.1	Lasterkennung	Falls <i>Silisloth</i> den Zielmast nicht erkennt, soll es durch einen Nothalt beim Berühren des Mastes anhalten.	Drucktaster beendet Ablauf endgültig

#### 4. Komponenten

8.2	Lasterkennung	Falls <i>Silisloth</i> die Last nicht erkennt, fährt es automatisch zum Ziel.	Drucktaster beendet Ablauf endgültig
8.3	Zielerkennung	<i>Silisloth</i> lässt die Last beim Zielmast fallen, falls es das Zielfeld nicht erkennt.	Drucktaster beendet Ablauf endgültig
8.4	Schwingen	<i>Silisloth</i> hält an, falls die Schwingung in y-Richtung $20^\circ$ überschreitet	durch statistische Auswertung der Ultraschallsensor-Messwerte möglich
8.5	Lastverlust	<i>Silisloth</i> erkennt, wenn es seine Last verliert.	Lösung mit Schrittmotor prüfen

Tabelle 5: Anforderungen mit Einschätzung aufgrund des Konzepts

Nach Betrachtung von *Tabelle 5: Einschätzung des Konzepts gemäss Anforderungen, Seite 36* kann unterm Strich festgestellt werden, dass sich das Konzept im Rahmen der Anforderungen bewegt. Kleinere Abweichungen, wie etwa die Messungenauigkeit der Ultraschallsensoren bei grossen Distanzen, sollten für die Realisierung kein Problem darstellen. Schwierigkeiten können sich bei der Umsetzung trotzdem ergeben, gerade in folgenden Bereichen:

**Zusammenspiel der Komponenten** Die Komponenten wurden bisher eher isoliert betrachtet. Zwar wurden auch Abklärungen und Versuche im Bereich der Schnittstellen gemacht, bei diesen wurden aber auch immer nur einzelne Komponenten kombiniert. Ein Prototyp, der nahezu alle Komponenten miteinander verbindet, wurde bisher nicht gebaut.

**Bildverarbeitung** Der ganze Hardware- und Software-Stack zur Erkennung des Zielfeldes funktioniert grundlegend. Die benötigte Implementierung, die auch den Abstand zu Zielfeld und Lastwürfel berechnen können muss, dürfte aber um einiges schwieriger ausfallen, zumal in der Gruppe 7 noch wenig Wissen in diesem Bereich vorhanden ist. Die Bildverarbeitung ist auch aus Sicht der Geschwindigkeit der Flaschenhals, denn je weniger Bilder pro Sekunde ausgewertet werden können, desto langsamer muss sich *Silisloth* fortbewegen.

**Silikongreifer** Der Silikongreifer ist sehr tolerant was Form, Oberfläche, Gewicht und Position der Last angeht. Er ist aber auch fehleranfällig, denn sobald auch nur ein kleiner Riss entsteht und Luft entweicht, kann die Last nicht aufgenommen werden bzw. fällt sie sofort herunter. Weitere Erfahrungen in den Bereichen Herstellung, Umgang und Ausbesserung sind hier vonnöten.

**Schwingungen** Sollten bei Tests stärkere Schwingungen in y-Richtung auftreten, müssen diese zur Laufzeit erkannt werden können. Dies kann mit statistischen Auswertungen der Ultraschallsensor-Messwerte oder mit einem zusätzlichen Neigungssensor umgesetzt werden.

## 5. Projektmanagement und Projektplanung

### 5.1. Risikomanagement

In der Risikomatrix (siehe *Tabelle 6: Die Risikomatrix, Seite 38*) sind die erkannten Risiken aufgelistet. Dabei werden die Risiken nach Eintrittswahrscheinlichkeit und möglichem Schadensausmass eingeteilt. Für Risiken, die sich im gelben Bereich befinden, werden mögliche Gegenmassnahmen erläutert um das Risiko zu minimieren.

Häufig				
Wahrscheinlich				
Gelegentlich		Seilschwingung	Greiftechnik Hindernisse	
Unwahrscheinlich			Gesamtgewicht Geschwindigkeit	Kosten Freedom-Board
Unvorstellbar				Gegengewicht
	Unwesentlich	Geringfügig	Kritisch	Katastrophal
SCHADENSAUSMASS				

Tabelle 6: Die Risikomatrix

#### 5.1.1. Erläuterungen und Gegenmassnahmen

**Seilschwingung** Durch die Fortbewegung von *Silisloth* und/oder das Heben der Last entsteht eine Schwingung am Seil. Die Last muss zudem noch hoch genug angehoben werden, damit sie trotz Schwingungen nicht die Hindernisse berührt. Sollten die Seilschwingungen zu stark sein, wird analysiert, wo diese Schwingungen auftreten. Sollten die Schwingungen beim Heben oder Senken der Last entstehen, müsste dieser Vorgang verlangsamt werden. Entstehen die Schwingungen während der Fahrt, müsste die Fortbewegungsgeschwindigkeit gesenkt werden, sofern *Silisloth* seine Aufgabe noch innerhalb der maximal definierten Zeit von vier Minuten erfüllt. Wird die Zeit von vier Minuten überschritten, und die Seilschwingungen sind immer noch vorhanden, muss der Schwerpunkt gesenkt werden ohne dabei die Hindernisse zu berühren. Reicht das Senken der Last nicht zum Senken des Schwerpunkts, muss *Silisloth* umkonstruiert werden um den Schwerpunkt tiefer platzieren zu können.

**Hindernisse** Die Hindernisse verfälschen den Z-Wert (die Höhe über der Plattform) der Last. Sind es nur einzelne Hindernisse, können ungültige Z-Werte durch Ausschliessen ignoriert werden. Sollten die Hindernisse jedoch eine konstante Steigung haben und auf der ganzen Strecke vorhanden sein, wird es schwierig, diese mittels des Ultraschalldistanzsensors zu erkennen und zu

ignorieren. Aufgrund der Möglichkeit eines Hindernisses mit konstanter Steigung wird der Punkt «Hindernisse» als kritisch eingestuft. Sollte ein solches Hindernis verwendet werden, würde als Gegenmassnahme der Z-Wert nicht mehr über den Ultraschalldistanzsensor gemessen, sondern anhand des gemessenen X-Wertes (die zurückgelegte horizontale Strecke) und der Steigung des Seils berechnet.

**Gesamtgewicht** *Silisloth* ist schwerer als das maximal definierte Gewicht, weshalb das Seil zu stark durchhängt und dadurch die Hindernisse berührt werden. Dieses Risiko wird als unwahrscheinlich eingestuft. Darum sind hier keine Gegenmassnahmen vorgesehen.

**Kosten** Die Kosten für das Projekt sind höher als die maximal verfügbaren 500 Franken. Eine Kostenüberschreitung wäre katastrophal, da damit das Projekt nicht umgesetzt werden könnte. Als Gegenmassnahme müssten die Kosten wo immer möglich reduziert werden. Aufgrund der Kostenschätzung der genutzten Komponenten wird eine Kostenüberschreitung jedoch als unwahrscheinlich eingestuft (siehe siehe *Abschnitt 5.2: Ressourcenbudget, Seite 39*).

**Gegengewicht** *Silisloth* ist zu schwer für das Gegengewicht, wodurch es den Boden berührt. Aufgrund des hohen maximalen Gewichts wird dieses Risiko als unvorstellbar eingestuft. Darum sind hier keine Gegenmassnahmen vorgesehen.

**Geschwindigkeit** *Silisloth* fährt zu langsam und erreicht deshalb das Ziel nicht in der vorgegebenen Zeit von unter vier Minuten. Da vier Minuten mehr als genug für das Erledigen der Aufgabe erscheinen, wird das Risiko als unwahrscheinlich eingestuft.

**Greiftechnik** Die Last kann über die Greiftechnik mit dem Softrobotics-Greifer nicht angehoben werden oder geht während der Fahrt verloren. Sollte sich diese Greiftechnik nicht bewähren, müsste die Last mit einem Magneten angehoben werden. Die Greiftechnik mit einem Magneten hat sich in einem ersten Versuch bereits als verlässliche Alternative bewährt.

**Freedom-Board** Das Freedom-Board wird für die Ansteuerung der Motoren verwendet. Sollte das Freedom-Board nicht zum Laufen gebracht werden, wäre das katastrophal, da dann *Silisloth* notwendige Funktionen nicht ausführen könnte. Als Alternative könnte das Arduino-Board für die Motorenansteuerung verwendet werden. Dieses hat sich in verschiedenen Tests schon als leicht zugänglich erwiesen.

## 5.2. Ressourcenbudget

Gesamthaft stehen für PREN 1 und PREN 2 CHF 500 zur Verfügung. Davon sind CHF 200 für das erste Semester bzw. PREN 1 vorgesehen, um im Rahmen der Konzeptphase Lösungsansätze zu erproben. Mit diesem Budget sollen auch die erforderlichen Komponenten gekauft werden. Zusätzlich

## 5. Projektmanagement und Projektplanung

	Kostenbudget PREN 1	CHF 200.00
Informatik	Raspberry Pi	CHF –38.90
	Kamera	CHF –29.90
	SD-Karte	CHF –19.90
	Akku	CHF –19.90
	USV	CHF –29.90
	Netzteil	CHF –10.00
Elektrotechnik	Freedom Board	CHF –21.95
	Motorensteuerung	CHF –6.95
Maschinenbau	Luftpumpe 2×	CHF –5.00
	Silikon (anteilmässig)	CHF –7.00
	Restbudget	CHF 12.60

Tabelle 7: Das Kostenbudget für PREN 1

wurde damit auch der Zweikomponenten-Silikon beschafft, welcher für den Silikongreifer benötigt wird. Wie *Tabelle 7: Das Kostenbudget für PREN 1, Seite 40* zeigt, wurde das Budget grösstenteils aufgebraucht. Die Komponenten wurden nach vertiefter Recherche bewusst ausgesucht, sodass diese auch in PREN 2 weiterverwendet werden können.

Zusätzliche Mittel, wie zum Beispiel Maschinenlaufzeiten oder Arbeitsstunden, wurden (mit Ausnahme eines privaten 3D-Druckers) noch nicht in Anspruch genommen. Somit sind noch genügend Ressourcen für PREN 2 vorhanden, wie *Tabelle 8: Das Restbudget für verschiedene Hilfsmittel (PREN 1 & PREN 2), Seite 40* zeigt.

MITTEL	ZEITBUDGET	BEZUG	RESTBUDGET
3D-Drucker (privat)	-	8h	-
3D-Drucker (HSLU)	25h	0h	25h
Lasergerät	1h	0h	1h
Werkstattpersonal Elektrotechnik	10h	0h	10h
Werkstattpersonal Maschinenbau	10h	0h	10h

Tabelle 8: Das Restbudget für verschiedene Hilfsmittel (PREN 1 & PREN 2)

## 6. Schluss

### 6.1. Weiteres Vorgehen

Das in PREN 1 Erarbeitete bildet den Grundstein für das nächste Semester, mit dem Ziel, das Konzept umzusetzen und das Projekt erfolgreich zum Abschluss zu bringen. In PREN 1 konnten schon einige Schwierigkeiten frühzeitig erkannt und grösstenteils ausgeräumt werden. Das Konzept genügt den Anforderungen weitgehend. Für kritische Komponenten stehen Alternativen zur Verfügung. Nun ist es wichtig, die erarbeiteten Pläne zu präzisieren, damit die Konstruktion weiter an Form gewinnt, so dass am Ende vom Modul PREN 2 das voll funktionsfähige *Silisloth* herauskommt, das mehr leistet, als nur kopfüber von einem Seil zu hängen und dabei Ressourcen zu verbrauchen.

### 6.2. Lessons Learned

**Anforderungen** Anforderungen können sich im Verlauf eines Projektes ändern. Betreffend Zugwiderstand des Seils unterschieden sich zu Beginn die Abschätzungen erheblich vom aufgestellten Parcours, was sich aber aufgrund der später ausgetauschten Laufrollen wieder etwas angeglichen. Darum ist es nötig, bei wichtigen Parametern (wie etwa dem Gewicht) nicht ans absolute Limit zu gehen, sondern etwas Spielraum einzuberechnen.

**Bildverarbeitung** Für die Bildverarbeitung lohnt es sich genügend Zeit in die Recherche zu investieren. Dadurch lassen sich nicht nur weit verbreitete Bildverarbeitungs-Libraries wie OpenCV finden, sondern auch passende Beispiele dazu.

**Dokumentation** Typografisch hochstehende Ergebnisse erreicht man mit L<sup>A</sup>T<sub>E</sub>X, wenn man bereit ist den entsprechenden Aufwand auf sich zu nehmen. Gerade die Darstellung von Tabellen ist sehr anspruchsvoll und erfordert eine hohe Lernbereitschaft. Bei bekannten Problemen sollte man sich besser nicht an eigene Lösungen heranwagen, sondern etablierte Lösungen verwenden, auch wenn diese auf den ersten Blick als zu kompliziert erscheinen mögen. Es lohnt sich für häufig wiederkehrende Aufgaben eine eigene Makro-Sammlung anzulegen, gerade im Hinblick auf weitere Dokumente und Projektarbeiten.

**Freedom-Board** Der Einstieg in die Arbeit mit dem FRDM-KL25Z stellt eine grosse Hürde dar, gerade im Gegensatz zur Arduino-Plattform. In Hinsicht auf noch folgende Module im Bereich Embedded Systems lohnt es sich hier aber Zeit zu investieren. Ist diese Einstiegshürde erst einmal überwunden, kann einiges übersichtlicher und verständlicher programmiert und allfällige Fehler können einfacher aufgespürt werden.

**Grafiken** Schematische Grafiken lassen sich mit Graphviz einfach als textuelle Struktur beschreiben. Für spezielle Grafiken, wie z.B. die Hardware-Architektur, erreicht man aber mit einem

## *6. Schluss*

WYSIWYG-Werkzeug wie LibreOffice Draw schnelle und akzeptable Ergebnisse.

**Kommandozeile** Die Interaktion mit dem Raspi erfordert Kenntnisse im Linux-Bereich und Vertrautheit im Umgang mit der Kommandozeile. Diese Themen werden an der HSLU – Informatik eher stiefmütterlich behandelt. Darum war es ein Glücksfall, dass die Informatik-Studenten der Gruppe 7 Vorkenntnisse in diesem Bereich hatten.

**Netzwerk** In einem grossen Netzwerk, wie in demjenigen der HSLU, ist es nicht ganz einfach, ein bestimmtes Gerät zu finden. Am einfachsten, wenn auch nicht am schnellsten, geht es mittels ARP-Scan und der Filterung nach einer bestimmten MAC-Adresse.

**Schnittstellen** Es ist wichtig ganzheitlich zu denken und nicht nur im jeweiligen Fachbereich. Dazu ist es notwendig die Schnittstellen zwischen den einzelnen Gebieten klar zu definieren und zu benennen. Dadurch kann sichergestellt werden, dass die einzelnen Komponenten und Abläufe reibungslos ineinander greifen.

**Silikongreifer** Beim Silikongreifer brauchte es mehrere Anläufe bis ein zufriedenstellendes Ergebnis erzielt werden konnte. Bei Defekten musste der Fehler analysiert und entsprechend behoben werden. Mehrmals sind die Rippen im inneren des Greifers ausgerissen, weshalb die Gussform angepasst wurde. Neu wurden weniger Rippen und mit einem grösseren Abstand dazwischen gedruckt, damit die Flächen grösser werden und die Stabilität dadurch erhöht wird.

**Stromversorgung** Ein Ladekabel mit 2.5A mag zwar einen Akku schneller laden als eines mit nur 1A, beim Ein- und Ausstecken kann es aber zu raschen Spannungsschwankungen kommen, die einen unterbrechungsfreien Betrieb verunmöglichen.

### **6.3. Fazit**

Die Zusammenarbeit im Team funktionierte über das ganze Semester hinweg sehr gut. Die aufgeteilten Aufgaben wurden stets zuverlässig bis zu dem abgemachten Termin erledigt. Naturgemäß verdichtete sich der Aufwand für die zu erstellenden Dokumente vor den Testatterminen. Darauf folgten jeweils kreative Phasen des Experimentierens, was für eine willkommene Abwechslung sorgte. Das Arbeitsklima war immer angenehm, und die Kommunikation funktionierte. Außerdem konnten die Gruppenmitglieder bereits im Modul PREN 1 viel voneinander profitieren und lernen. Dies soll auch im Folgemodul PREN 2 beibehalten und weitergeführt werden.

## A. Projektauftrag

Der Projektauftrag wurde zu Beginn des Moduls PREN I ausgehändigt und wird hier unverändert als separates Dokument (Anhang-A\_Projektauftrag.pdf) abgegeben.

## B. Recherche

Die Recherche war das Ergebnis des ersten Testats. Die Rechercheergebnissen werden (mit kleineren kosmetischen Änderungen) als separates Dokument (Anhang-B\_Recherche.pdf) abgegeben.

## C. Anforderungsliste

Die Anforderungsliste, die ebenfalls Gegenstand des ersten Testats war, wird als separates Dokument (Anhang-C\_Anforderungsliste.pdf) abgegeben. Auch hier wurden nur kosmetische Änderungen vorgenommen.

## D. Konzeptvarianten

Das Dokument «Konzeptvarianten» (Anhang-D\_Konzeptvarianten.pdf) ist das Ergebnis des zweiten Testats und wird nahezu unverändert abgegeben.

## E. Detaillierte Berechnungen

### E.1. Reibung

Das maximale Gewicht von *Silisloth* beträgt 4kg. Bei einer maximalen Seilsteigung von 45° beträgt die durch Hafreibung kompensierende Kraft:

$$F_r = \cos(45^\circ) \cdot 4\text{kg} \cdot 9.81 \frac{\text{m}}{\text{s}^2} = 27.75\text{N} \quad (5)$$

Die Hafreibung ist definiert als:

$$F_r \leq F_N \cdot \mu \quad (6)$$

Mit  $\mu = 0.13$  ergibt sich für die Normalkraft:

$$F_N \geq \frac{F_r}{\mu} \geq \frac{27.75\text{N}}{0.13} \geq 213.46\text{N} \quad (7)$$

Die Normalkraft auf dem treibenden Rad berechnet sich mit:

$$F_N = 2 \cdot 15kg \cdot 9.81 \frac{m}{s^2} \cdot \sin(\tan^{-1}(\frac{40mm}{155mm})) = 75N \quad (8)$$

Da das treibende Rad eine V-förmige ( $30^\circ$ ) Lauffläche besitzt, ergibt sich als die effektive Normalkraft:

$$F_N = 2 \cdot \frac{75N}{\sin(30^\circ)} = 300N \quad (9)$$

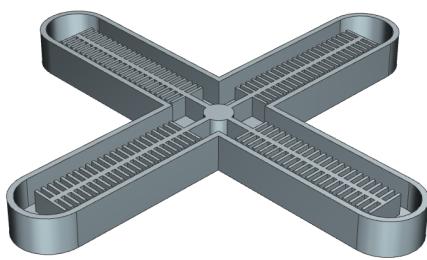
## F. Versuche

### F.1. Silikongreifer

#### F.1.1. Herstellung

Der auf zwei verschiedenen Silikonen basierende Greifer wurde in sechs Schritten hergestellt (Finio, 2014).

1. In einem ersten Schritt wurde die eigentliche Gussform in CAD geplant und anschliessend 3D gedruckt (siehe *Abbildung 22a, Seite 45*).
2. Anschliessend wurden je 30g von den zwei Silikonkomponenten in einem Plastikbecher zusammengeführt und rund drei Minuten lang vermischt (siehe *Abbildung 22b, Seite 45*). Dabei entstanden kleine Luftblasen, welche unbedingt entfernt werden mussten, da sie allenfalls den eigentlichen Greifarm schwächen können und dadurch keine vollständige Dehnung des Greifers gewährleistet gewesen wäre.
3. Um die Luftblasen zu entfernen, wurde das Silikongemisch unter eine Vakuumglocke gestellt (siehe *Abbildung 22c, Seite 45*). Die Bläschen wurden dabei aus dem Silikon gesogen. Sobald alle Bläschen beseitigt waren, wurde das Gemisch in die zuvor angefertigte Form gegossen (siehe *Abbildung 22d, Seite 45*). Diese wurde bei Zimmertemperatur insgesamt vier Stunden stehen gelassen, sodass sich das Silikongemisch verfestigen konnte.
4. Weitere 30g desselben Silikongemisches wurden in einer dünnen Schicht auf ein Stück Stoff gegossen. Das bereits feste Silikon teil wurde auf die noch flüssige Silikonschicht platziert. Dadurch wurde die Greifhand rückseitig verschlossen. Der Zweck des Stoffes bestand darin, die gewünschte Greifbewegung zu ermöglichen. Durch den Stoff wird verhindert, dass sich die untere Seite des Greifarms ausdehnen kann. Somit verformt sich nur die obere Seite des Greifers, wodurch die gewünschte Greiffingerkrümmung gewährleistet wird und die gegebene Last gefasst werden kann.



(a) Die Gussform (CAD-Modell)



(b) Die beiden Silikon-Komponenten



(c) Die Silikonmischung unter der Vakuumglocke



(d) Das Giessen des Silikongreifers

Abbildung 22: Die Herstellung des Silikongreifers

5. Um die bereits oben genannte Luftzufuhr zu ermöglichen, wurde ein zuvor ebenfalls aus Silikon angefertigter Zylinder mit einem Luftkanal an der oberen Seite der Greifhand befestigt. Erstaunlicherweise konnten alle Silikonteile, ob nun bereits in verfestigtem oder noch flüssigem Zustand, mit Silikonresten nahtlos verbunden werden.
6. Die vollständige Greifhand wurde nach vier Stunden in einem abschliessenden Schritt für rund zwei Stunden bei  $80^{\circ}C$  in einem Industrieofen erwärmt, um eine maximale Dehnung und Festigkeit zu gewährleisten. Dies wurde vom Silikonhersteller empfohlen (*Ecoflex® oo-30*, o.J., Curing/Post Curing).

### **F.1.2. Belastungstest**

Der Silikongreifer wurde darauf getestet, ob im aufgeblasenen Zustand die Dehnung und Festigkeit ausreicht. Für das gewünschte Volumen musste 5 mal gepumpt werden, der Riss erfolgt erst bei 30 Mal pumpen. Dies zeigte, dass die fünf notwendigen Pumpvorgänge weit entfernt vom kritischen Volumen sind und die Festigkeit und Dehnung den Anforderungen mehr als gerecht werden. Da einige Rippen im inneren ausgerissen sind, wurde die Gussform angepasst. Die Anzahl Rippen wurde verringert, dafür sind sie in der neueren Form breiter. Ob genug Reibung zwischen Greifer und Last ist, konnte direkt an der Last getestet werden, indem der Greifer aufgeblasen und übertrieben starken Schwingungen ausgesetzt wurde.

### **F.2. Ultraschallsensor**

Um die Messgenauigkeit des Ultraschallsensors zu Testen, wird der HC-SR04 an den Raspi angeschlossen und über ein Python-Skript in Betrieb genommen. Die gemessenen Distanzen werden direkt auf der Konsole ausgegeben. Getestet wird direkt auf dem Parcours. Der Sensor ist auf einem Steckbrett angebracht und auf den Masten am Ende des Seils gerichtet. Es werden immer 14 Messungen durchgeführt und aufgelistet. Zuerst wird die gesamte Länge (337cm) gemessen. Danach ab 200cm in jeweils in Schritten von 20cm bis zu 10cm. Nun werden noch Messungen bei 5cm, 2.5cm, und 1.5cm durchgeführt. Von allen Messgruppen wird jeweils der Durchschnitt (Mean) und Median bestimmt.

Zwischendurch kann es zu Messungen kommen, die komplett falsch sind. So z.B bei 180cm, als ein Wert von 59.1cm gemessen wurde. Diese sind aber extrem selten. Um dem entgegenzuwirken, kann der Median der Messgruppe verwendet werden. Anders als beim Mittelwert (Mean) bewirkt ein solcher «Ausreißer» keine Änderung des Endresultates. Der Median ist also wesentlich robuster als der Mittelwert. Ab einer Distanz von 100cm misst der Ultraschallsensor tendenziell eher zu wenig: ca. 2cm fehlen meist. Unter 100cm sind die gemessenen Distanzen erstaunlich genau. Der Fehler des Medians beträgt nie mehr als 2.5%. Schön zu sehen ist auch die minimale Messdistanz des Sensors bei ca. 3cm. Unterhalb dieser Grenze können keine genauen Messdaten mehr geliefert werden. Dies stellt

### Distanzmessung: von Hand und per Ultraschallsensor

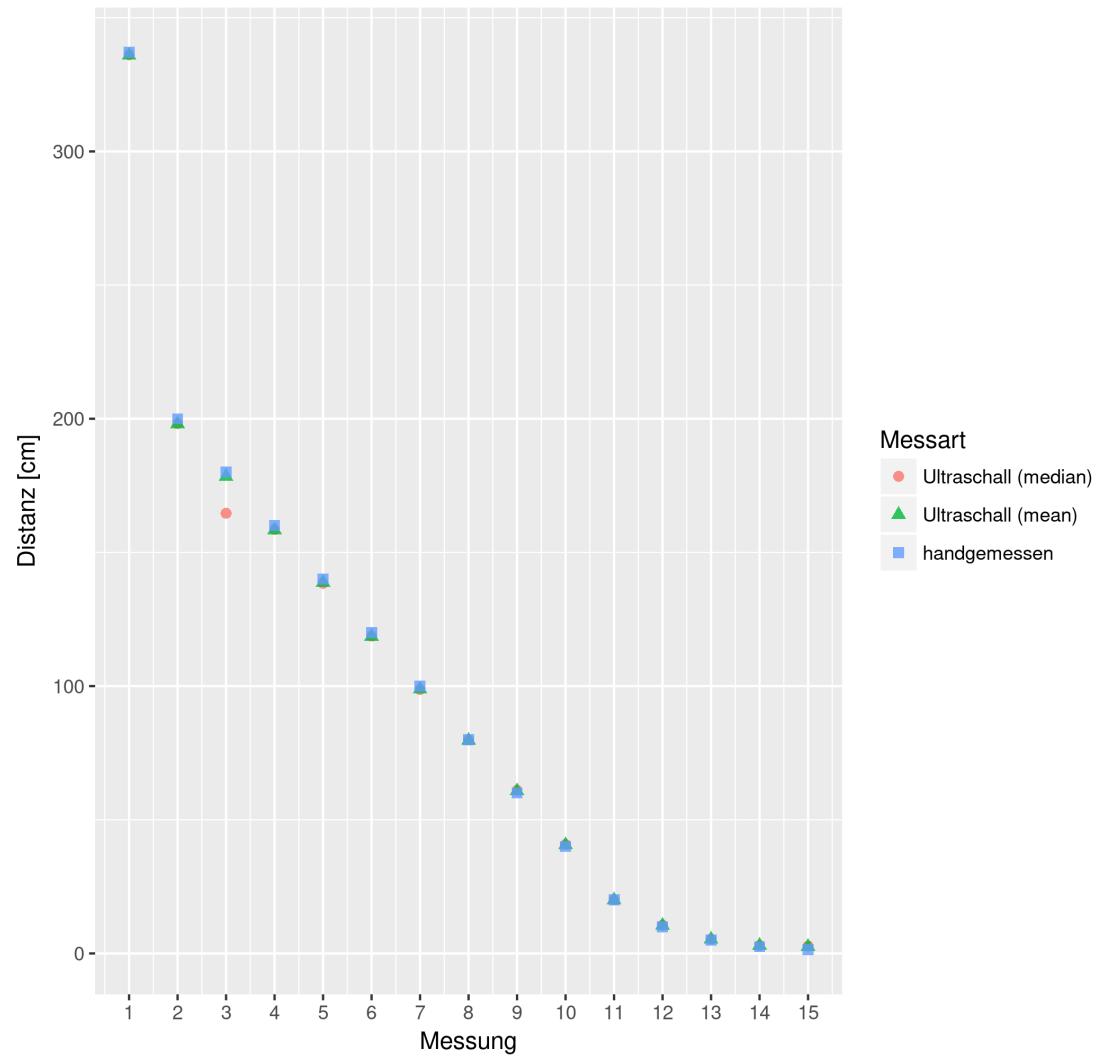


Abbildung 23: Die Messungen mit dem Ultraschallsensor

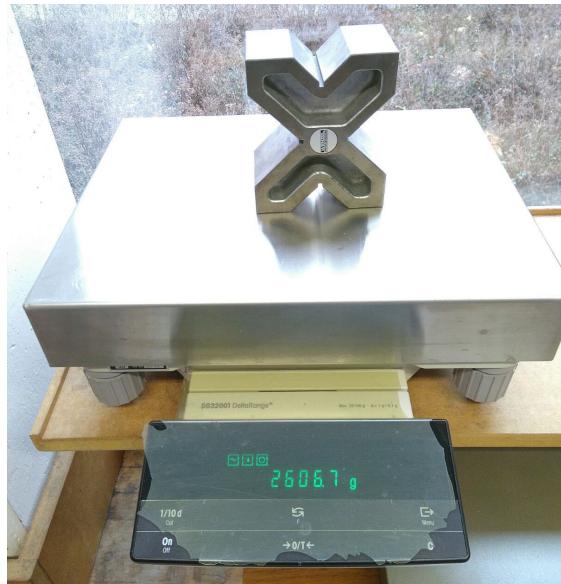


Abbildung 24: Das Gewicht der Last, die mit dem Elektromagneten angehoben wurde

aber kein Problem dar, da der Ultraschallsensor einfach so am Gerät platziert wird, dass dieser nie unter dieser Messgrenze messen muss.

*Abbildung 23, Seite 47* veranschaulicht die Messungen. Pro Messpunkt werden drei Werte einander gegenübergestellt: die handgemessene Distanz, der Median der Ultraschallmessung und der Mittelwert der Ultraschallmessung. Die drei Punkte liegen dermassen nahe beieinander, dass von blossem Auge kaum Abweichungen auszumachen sind. Einzig bei der dritten Messung ist beim Mittelwert eine grosse Abweichungen zu sehen. Das liegt daran, dass der Sensor teilweise fehlerhafte Daten liefert, und dass der Mittelwert sehr anfällig für «Ausreisser» ist. Aus diesem Grund soll sich die Steuerung auf den Median verlassen.

### F.3. Elektromagnet

Als Alternative zum Silikongreifer wurde ein Elektromagnet getestet. Als Modell wurde der Intertec ITS-PE3529-24VD (*Elektromagnet 300 N 24 V/DC 28 W Intertec ITS-PE3529-24VDC*, o.J.) verwendet. Dieser Elektromagnet kann bis zu 300N anheben. Für das Abladen werden 24V und 1.16A benötigt. Die Speisung wird jedoch nur für das Abladen der Last benötigt. Solange man den Magneten nicht speist, wird die Last voll angezogen. Für den Versuch wurde eine Last verwendet, welche etwas mehr als 2.6kg wiegt (siehe *Abbildung 24, Seite 48*).

Die Last von 2.6kg konnte ohne Probleme angehoben werden, wie *Abbildung 25, Seite 49* zeigt.



Abbildung 25: Die vom Elektromagneten angehobene Last

## G. Meilensteinberichte

Die Meilensteinberichte werden als separates Dokument (Anhang-G\_Meilensteinberichte.pdf) abgegeben.

## H. Protokolle

Die Protokolle werden als separates Dokument (Anhang-H\_Protokolle.pdf) abgegeben. Sie beinhalten eine wöchentliche Übersicht über die Tätigkeiten aller Gruppenmitglieder.

## I. Gruppenziele

1. Wir wollen gut zusammen arbeiten – auch über die Fachbereiche hinweg.
2. Wir wollen Spass an der Arbeit haben.
3. Wir wollen etwas lernen – in unserem Fachbereich und fachbereichübergreifend.
4. Wir wollen, dass jedes Gruppenmitglied seinen Beitrag leistet.
5. Wir wollen eine sichere, zuverlässige und für alle verständliche Lösung erarbeiten, nicht den Wettbewerb um jeden Preis gewinnen.

## J. Projektplan

Der Projektplan wird als separates Dokument (Anhang-J\_Projektplan.pdf) abgegeben.

## Quellen

- About OpenCV.* (2017). <https://opencv.org/about.html>.
- Adafruit Motor Shield.* (2015). <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-motor-shield.pdf>.
- Arduino Uno Rev3.* (2017). <https://store.arduino.cc/usa/arduino-uno-rev3>.
- Camera Module V2.* (o.J.). <https://www.raspberrypi.org/products/camera-module-v2/>.
- Ecoflex® 00-30.* (o.J.). <https://www.smooth-on.com/products/ecoflex-00-30/>.
- Elektromagnet 300 N 24 V/DC 28 W Intertec ITS-PE3529-24VDC.* (o.J.). <https://www.conrad.ch/de/elektromagnet-300-n-24-vdc-28-w-intertec-its-pe3529-24vdc-506164.html>.
- Finio, B. (2014). Air-Powered Soft Robotic Gripper.* <http://www.instructables.com/id/Air-Powered-Soft-Robotic-Gripper/>.
- Getriebemotor 12 V Igarashi TYP 33G-125.* (1999). <https://www.conrad.ch/de/getriebemotor-12-v-igarashi-typ-33g-125-244031.html>.
- Kinetis KL25 Sub-Family.* (2014). <https://www.nxp.com/docs/en/data-sheet/KL25P80M48SF0.pdf>.
- Mikroendschalter.* (o.J.). <http://ml-tec-shop.ch/index.php?a=495>.
- Mini Air Pump DC 6V and DC 12V Motor 370 for Aquarium.* (o.J.). <https://www.wish.com/c/570fc11a9df4c35cd9428f82>.
- Plate, J. (2017). Raspberry Pi: Ultraschall-Entfernungsmessung mit HC-SR04.* <http://www.netzmafia.de/skripten/hardware/RasPi/Projekt-Ultraschall/index.html>.
- Python. A Programming Language Changes the World.* (2017). <http://brochure.getpython.info/media/releases/prerelases/psf-python-brochure-vol-1-final-content-preview>. (Brochure)
- Raspberry Pi Model 3.* (2016). <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.  
(Technische Spezifikation)
- Raspbian FAQ.* (o.J.). <https://www.raspbian.org/RaspbianFAQ>.
- Schrittmotor Emis E547-52500.* (o.J.). <https://www.conrad.ch/de/schrittmotor-emis-e547-52500-025-nm-06-a-wellen-durchmesser-5-mm-198398.html>.
- Styger, E. (2012). Tutorial: USB CDC with the KL25Z Freedom Board.* <https://mcuoneclipse.com/2012/10/07/tutorial-usb-cdc-with-the-kl25z-freedom-board/>.
- Ultraschall-Sensor HC-SR04 und kompatible Ultraschall-Module.* (o.J.). <http://www.mikrocontroller-elektronik.de/ultraschallsensor-hc-sr04/>.
- Ultrasonic Ranging Module: HC-SR04.* (2010). <http://cdn2.boxtec.ch/pub/iteadstudio/HC-SR04.pdf>.

# Abbildungsverzeichnis

1.	Gruppenfoto (von links nach rechts): Alex Duong, Christoph Binkert, Sandro Ber tozzi, Quentin Frei (vorne), Patrick Bucher (hinten), Jan Greber, Marko Lovrinovic (vorne), Johannes Togan (hinten) . . . . .	2
2.	Funktionales Blockschaltbild . . . . .	8
3.	Der zeitliche Ablauf als Zustandsdiagramm . . . . .	10
4.	CAD-Modell von <i>Silisloth</i> (Ansicht links) . . . . .	12
5.	CAD-Modell von <i>Silisloth</i> (Ansicht rechts) . . . . .	13
6.	CAD-Modell der Aufhängung (Explosionsansicht) . . . . .	14
7.	Motor Shield HP-ASH-L293D . . . . .	17
8.	Miniluftpumpe mit Motor . . . . .	18
9.	Schrittmotor Emis E547-52500 . . . . .	18
10.	Getriebemotor Igarashi Typ33-125 . . . . .	19
11.	Ultraschallsensor HC-SR04 . . . . .	19
12.	Ansteuerung des HC-SR04 . . . . .	19
13.	Das Freedom-Board FRDM-KL25Z . . . . .	21
14.	Mikroendschalter . . . . .	22
15.	Raspberry Pi Model B Version 3 . . . . .	23
16.	Das Raspberry Pi Camera Module V2 . . . . .	25
17.	Das fotografierte Zielfeld (links) mit erkannten Konturen (rechts) . . . . .	25
18.	USV Adafruit PowerBoost 1000C . . . . .	27
19.	Das User-Interface der Smartphone-App (schematisch) . . . . .	28
20.	Arduino Uno Rev3 . . . . .	30
21.	Die Hardware-Architektur: Komponenten und Schnittstellen . . . . .	31
22.	Die Herstellung des Silikongreifers . . . . .	45
23.	Die Messungen mit dem Ultraschallsensor . . . . .	47
24.	Das Gewicht der Last, die mit dem Elektromagneten angehoben wurde . . . . .	48
25.	Die vom Elektromagneten angehobene Last . . . . .	49

## **Tabellenverzeichnis**

1.	Berechnung der Akku-Kapazität für die Rechenkomponenten . . . . .	22
2.	Berechnung der Akku-Kapazität für die Motoren und die Luftpumpe . . . . .	23
3.	Vor- und Nachteile des Arduino Uno Rev3 gegenüber dem FRDM-KL25Z . . . . .	29
4.	Vor- und Nachteile des Silikongreifers gegenüber dem Elektromagneten . . . . .	32
5.	Anforderungen mit Einschätzung aufgrund des Konzepts . . . . .	36
6.	Die Risikomatrix . . . . .	38
7.	Das Kostenbudget für PREN 1 . . . . .	40
8.	Das Restbudget für verschiedene Hilfsmittel (PREN 1 & PREN 2) . . . . .	40