

# PRODUKTENTWICKLUNG 1

HOCHSCHULE LUZERN

TECHNIK & ARCHITEKTUR

---

## Lösungskonzept

Solution Concept

---

*Autoren:*

Adriano VALSANGIACOMO  
Christian SPYCHER  
Christian SCHÜRCH  
Ervin MAZLAGIĆ  
Fabian WÜTHRICH  
Alexander SUTER

*Modulbetreuer:*

Martin VOGEL

Horw  
5. Dezember 2014

## Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>4</b>
<b>2 Konzeptwahl</b>	<b>5</b>
<b>3 Funktionsübersicht</b>	<b>7</b>
3.1 Energieversorgung . . . . .	7
3.1.1 Akku . . . . .	7
3.1.2 Drucktank . . . . .	7
3.1.3 Dampf . . . . .	7
3.1.4 Netzteil . . . . .	7
3.2 Ball-Lagerung . . . . .	8
3.2.1 Netz . . . . .	8
3.2.2 Magazin . . . . .	8
3.2.3 Korb . . . . .	8
3.2.4 Rohr . . . . .	8
3.3 Kommunikation . . . . .	8
3.3.1 ZigBee . . . . .	8
3.3.2 Bluetooth . . . . .	8
3.3.3 WLAN (eigenes Netz / Ad-hoc) . . . . .	9
3.4 Externe Steuerungseinheit . . . . .	9
3.4.1 Notebook . . . . .	9
3.4.2 Mobile . . . . .	9
3.5 Ortung des Korbs . . . . .	9
3.5.1 Optik . . . . .	9
3.5.2 Ultraschall . . . . .	10
3.5.3 Laser . . . . .	10
3.5.4 Wärmebild . . . . .	10
3.5.5 Radar . . . . .	10
3.6 Positionierung . . . . .	10
3.6.1 Fix . . . . .	10
3.6.2 Springt in Korb . . . . .	11
3.6.3 Fliegen . . . . .	11
3.6.4 Geradeaus fahren / Rollt . . . . .	11
3.7 Balltransport . . . . .	11
3.7.1 Luft . . . . .	11
3.7.2 Feder . . . . .	11
3.7.3 Drehräder . . . . .	11
3.7.4 Zylinder (Pneumatisch) . . . . .	11
3.7.5 Zylinder (Elektromagnetisch) . . . . .	11
3.8 Ausrichtung . . . . .	12
3.8.1 horizontale Ausrichtung . . . . .	12
3.8.2 vertikale Ausrichtung . . . . .	12
3.9 Bordcomputer . . . . .	12
3.9.1 Raspberry Pi . . . . .	13
3.9.2 LowLevel Fallback . . . . .	14
3.9.3 Alternativen . . . . .	15

<b>4 Übersicht Lösungsideen</b>	<b>16</b>
4.1 Idee 1 . . . . .	16
4.2 Idee 2 . . . . .	16
4.3 Idee 3 . . . . .	17
4.4 Idee 4 . . . . .	17
4.5 Idee 5 . . . . .	18
4.6 Idee 6 . . . . .	18
4.7 Idee 7 . . . . .	19
4.8 Idee 8 . . . . .	19
4.9 Auswahl . . . . .	19
<b>5 Konzept Drehrad</b>	<b>20</b>
5.1 Idee . . . . .	20
5.2 Annahmen . . . . .	20
5.3 Vor- und Nachteile . . . . .	20
<b>6 Konzept Druckluft</b>	<b>21</b>
6.1 Idee . . . . .	21
6.2 Annahmen . . . . .	21
6.3 Vor- und Nachteile . . . . .	21
<b>7 Konzept Propeller</b>	<b>22</b>
7.1 Idee . . . . .	22
7.2 Annahmen . . . . .	22
7.3 Vor- und Nachteile . . . . .	22
<b>8 Konzept-unabhängige Funktionen</b>	<b>23</b>
8.1 Übersicht . . . . .	23
8.2 Bordcomputer . . . . .	23
8.3 Kommunikation . . . . .	23
8.4 Ortung des Korbes . . . . .	23
<b>9 Schluss</b>	<b>24</b>
<b>Literatur</b>	<b>25</b>
<b>Anhang</b>	<b>26</b>
<b>A Ultraschallmodul HC-SR04</b>	<b>i</b>
A.1 Prüfeinrichtung . . . . .	i
A.2 Messgenauigkeit . . . . .	i
A.3 Messempfindlichkeit . . . . .	i
A.4 Fazit . . . . .	ii
<b>B Bildverarbeitung</b>	<b>iv</b>
B.1 Java . . . . .	iv
B.2 Python . . . . .	iv
<b>C Kameramodul Pixy (CMUcam5)</b>	<b>vi</b>
C.1 Eckdaten . . . . .	vi
C.2 Objekterkennung . . . . .	vi
C.3 PixyMon . . . . .	vi
C.4 Schnittstelle . . . . .	vi

C.5 Erweiterbarkeit . . . . .	vi
C.6 Versuch . . . . .	vii
<b>D Kommunikation</b>	<b>viii</b>
D.1 Ziel . . . . .	viii
D.2 Versuch . . . . .	viii
D.3 Fazit . . . . .	ix
<b>E Berechnungen</b>	<b>x</b>

## 1 Einleitung

Die Evaluation eines Lösungskonzeptes ist eine zentrale und kritische Phase eines jeden Entwicklungsprojektes. Kritisch ist diese insbesondere aufgrund der Tatsache, dass funktionelle Entscheidungen getroffen werden aufgrund der Rechercheergebnisse und der bisher gemachten Erfahrungen der einzelnen Projektteilnehmer. Diese Entscheidungsgrundlagen gilt es während dieser Phase in möglichst effizienter und dennoch angemessener Weise zu festigen, mit Hilfe von gezielten Detailrecherchen, praktischen Erfahrungen und theoretischen Abschätzungen.

Dieses Dokument beschreibt den Prozess der durchlaufen wurde, welcher zur Festigung und Erweiterung der besagten Entscheidungsgrundlagen diente. Dieser stellt eine starke Konkretisierung der ersten Rechercheergebnisse dar welche direkt zur Zusammensetzung von Funktionen zu Lösungskonzepten führt.

Im ersten Teil dieses Dokumentes wird spezifisch auf die verschiedenen Funktionen und deren mögliche Realisierungen eingegangen. Dabei wird klar gemacht, welche Kriterien zum Tragen kamen, bei der Beurteilung der jeweiligen Realisierung. Im darauf folgenden Teil werden die verschiedenen Lösungskonzepte näher erläutert, welche mittels der Evaluation nach folgenden Strategien zustande kamen.

### **Konzept Drehrad *Das Motivationspferd***

Das erste Konzept beschreibt die ideale Lösung nach welcher sich das ganze Team richtet. Dieses versucht möglichst allen Wunschvorstellungen gerecht zu werden ohne aus dem Rahmen des Vorstellbaren und Möglichen zu fallen.

### **Konzept Druckluft *Der Kompromiss***

Dieses Konzept dient als erste Alternative zum ersten Konzept und streicht von vornherein jene Funktionen welche ausschliesslich der Erfüllung von Wunschanforderungen dienen und die voraussichtlich grosse Realisierungsaufwände erfordern könnten.

### **Konzept Propeller *Das Fallnetz***

Dieses Konzept beschreibt eine Lösung welche allen Mindestanforderungen gerecht wird und dabei jeglicher Mehraufwand zur Erfüllung von Wunschanforderungen extrahiert ist.

Nebst den verschiedenen Paradigmen unterscheiden sich alle drei Konzepte auch in der Art der Realisierung von Grundfunktionen bei denen die dazu erforderliche Modularität gegeben ist. Beispielsweise ist es irrelevant ob eine Entfernungsmessung per Ultraschall oder Laser erfolgt. Ob das Gerät stationär ist oder sich bewegt, beeinflusst aber sehr wohl viele andere Teifunktionen und somit auch Systementscheide in beträchtlicher Art und Weise.

## 2 Konzeptwahl

Alle Konzepte werden nach einem einheitlichem Bewertungsraster bewertet. Jeder Punkt im Raster ist mit einer Gewichtung versehen. Diese Gewichtung ist ein Wert von 1 bis 3. Jeder Punkt wird selbst mit einem Wert von 0 bis 3 bewertet. Nachfolgend alle Punkte und die Bedeutung der Bewertung:

- Preis (Gewichtung 3)
  - 0 Punkte - Preis liegt nicht im Budget
  - 3 Punkte - Preis liegt im Budget
- Balltransport (Gewichtung 3)
  - 0 Punkte - Kann 2 oder weniger Bälle transportieren
  - 1 Punkt - Kann mind. 3 Bälle transportieren
  - 2 Punkte - Kann mind. 4 Bälle transportieren
  - 3 Punkte - Kann alle Bälle transportieren
- Robustheit (Gewichtung 3)
  - 0 Punkte - Umgebungsbedingungen haben starken Einfluss
  - 1 Punkte - Umgebungsbedingungen haben geringen Einfluss
  - 3 Punkte - Umgebungsbedingungen haben keinen Einfluss
- Zeit (Gewichtung 2)
  - 0 Punkte - Vorgang dauert länger als Anforderungen
  - 1 Punkt - Vorgang dauert lange
  - 2 Punkte - Vorgang dauert schnell
  - 3 Punkte - Vorgang dauert sehr schnell
- Gewicht (Gewichtung 1)
  - 0 Punkte - Mehr als 4 Kilogramm
  - 1 Punkte - Maximal 4 Kilogramm
  - 3 Punkte - Maximal 2 Kilogramm
- Umsetzbarkeit (Gewichtung 2)
  - 0 Punkte - Viele Fragezeichen wie es umgesetzt wird
  - 1 Punkte - Fragezeichen wie es umgesetzt wird vorhanden
  - 3 Punkte - Keine Fragezeichen wie es umgesetzt wird
- Aufwand (Gewichtung 2)
  - 0 Punkte - Nicht möglich in der gegeben Zeit umzusetzen
  - 1 Punkte - Sehr aufwändig umzusetzen
  - 3 Punkte - Aufwand hält sich im Rahmen
- Anforderungen (Gewichtung 3)
  - 0 Punkte - Erfüllt Pflichtanforderungen nicht
  - 2 Punkte - Erfüllt Pflichtanforderungen
  - 3 Punkte - Erfüllt Pflicht- und Wunschanforderungen

- Bauchgefühl (Gewichtung 3)

0 Punkte - Schlechtes Bauchgefühl

1 Punkte - Gemischtes Bauchgefühl

3 Punkte - Gutes Bauchgefühl

Nach dem Bewerten der einzelnen Konzepte werden die konkrete Punktgebung mit den Gewichtungen multipliziert. Alle Multiplikationen werden summiert.

Bewertungspunkt	Gewichtung	Konzept Drehrad	Konzept Druckluft	Konzept Propeller
Preis	3	3 (9)	3 (9)	3 (9)
Balltransport	3	2 (6)	1 (3)	1 (3)
Robustheit	3	3 (9)	3 (9)	3 (9)
Zeit	2	2 (4)	0 (0)	3 (6)
Gewicht	1	1 (1)	3 (3)	0 (0)
Umsetzbarkeit	2	1 (2)	1 (2)	0 (0)
Aufwand	2	3 (6)	1 (2)	1 (2)
Anforderungen	3	2 (6)	2 (6)	2 (6)
Bauchgefühl	3	3 (0)	0 (0)	1 (3)
Total		46	34	36

Tabelle 1: Bewertung

Das Konzept Drehrad hat nach der Bewertung die höchste Punktzahl erreicht und wird somit ausgewählt.

### 3 Funktionsübersicht

Die verschiedenen Funktionen und deren möglichen Implementierungen sind in einem morphologischen Kasten zusammengestellt (siehe Tabelle 2). In dieser Tabelle sind bereits ausselektierte Implementierungen durchstrichen dargestellt. Diese werden aufgrund der bisherigen Rechercheergebnisse nicht weiter betrachtet.

Die Interpretation des vorliegenden morphologischen Kastens gibt vor, dass sämtliche Implementierungen als *Module* zu betrachten sind und sich gegenseitig nicht ausschliessen oder negativ beeinflussen.

Funktion	Variante A	Variante B	Variante C	Variante D	Variante E
Energieversorgung	Akku	Druckluft	Dampf	Netz	
Balllagerung	Netz	Magazin	Korb	Rohr	
Kommunikation	ZigBee	Bluetooth	WLAN	Ad-Hoc	
Ortung des Korbes	Optik	Ultraschall	Laser	Wärmebild	Radar
Positionierung	fix	springt	fliegt	fährt linear	rollt
Balltransport	Luft	Feder		Drehräder	Zylinder
Ausrichtung	vertikal	horizontal			
Externe Steuerungseinheit	Notebook	Mobile			
Bordcomputer	Raspberry Pi	BeagleBone			

Tabelle 2: Morphologischer Kasten der Funktionen

#### 3.1 Energieversorgung

Der autonome Ballwerfer muss mit Energie versorgt werden. Die Anforderungen beschränken die Art der Energieversorgung nicht. Da das Gewicht der Energieversorgung nicht dem Gesamtgewicht angerechnet wird, sollte sie möglichst einfach entnehmbar sein. In den folgenden Abschnitten werden verschiedene Arten einer möglichen Energiezuführung kurz erläutert.

##### 3.1.1 Akku

Ein Akku als Stromlieferant ist vor allem bei einem fahrbaren Gerät Pflicht. Jedoch haben wir uns geeinigt dass unser Gerät nicht fahrbar ist und somit verliert diese Art der Energieversorgung an Relevanz. Ein Netzteil eignet sich besser, weil für mehr Leistung weniger Kosten anfallen. Gerade bei Elektromagneten oder starken Motoren geht ein normaler Akku schnell an seine Grenzen.

##### 3.1.2 Drucktank

Ein Drucktank eignet sich dafür ein mobiles Gerät zu Versorgen oder kurzfristig einen hohen Druck zur Verfügung zu stellen. Wie im oberen Abschnitt schon erwähnt haben wir uns gegen eine mobile Lösung entschieden. Jedoch kann ein Drucktank den nötigen Druck liefern um einen Ball direkt mit Luft abzuschiessen oder einen starken Pneumatikzylinder zu betreiben. Zudem kann der Drucktank verwendet werden um mögliche Schwankungen auf dem Luftdrucknetz auszugleichen.

##### 3.1.3 Dampf

Die Zeit der Dampflokomotiven ist wohl definitiv vorbei. Diese Möglichkeit ist während des Ideenfindungsprozesses entstanden und wird nur der Vollständigkeit halber erwähnt.

##### 3.1.4 Netzteil

Ein Netzteil liefert den nötigen Strom für eine Vielzahl von Komponenten und wird definitiv in unserem Gerät verwendet. Da die Kosten eines vorhandenen Netzteils nicht angerechnet werden, kann man hier genügend

Leistung zu einem akzeptablen Budget bereitstellen. Das Netzteil wird das schwächste Glied in der Kette, dem trotz seiner Einfachheit genügend Beachtung geschenkt werden sollte. Den ohne das Netzteil läuft der Bordcomputer nicht und auch die restlichen elektrischen Komponenten versagen ihren Dienst.

### 3.2 Ball-Lagerung

Die Art der Balllagerung hängt zu einem grossen Teil vom gewählten Abwurfmechanismus ab. Der Hauptzweck der Lagerung besteht darin, die Bälle sicher dem Abwurfmechanismus zuzuführen.

#### 3.2.1 Netz

Eine Balllagerung in einem Netz ist nur sinnvoll, wenn alle Bälle zusammen in den Korb geworfen werden. Ansonsten ergibt sich kein Vorteil durch diese Art der Balllagerung. Bei den von uns erarbeiteten Konzepten macht diese Balllagerung keinen Sinn, da die Bälle geordnet dem Abwurfmechanismus zugeführt werden müssen.

#### 3.2.2 Magazin

Bei dieser Variante werden die Bälle nacheinander dem Abwurfmechanismus zugeführt. Dies geschieht je nach gewählten Konzept mittels einer Feder oder einer Zahnstange. Der Vorteil dieser Variante besteht darin, dass die Bälle geordnet und geführt transportiert werden. Durch diese Vorteile ist diese Variante wohl die beste Lösung.

#### 3.2.3 Korb

Ähnlich wie die Variante Netz. Diese Variante wird nicht weiterverfolgt, weil sich keine Vorteile gegenüber der Variante „Magazin“ ergeben.

#### 3.2.4 Rohr

Ähnlich wie die Variante „Magazin“. Der Unterschied besteht darin, dass die Bälle nur mittels Schwerkraft dem Abwurfmechanismus zugeführt werden. Wenn die Schwerkraft ausreicht um die Bälle dem Abwurfmechanismus zuzuführen ist dies klar der einfachste Lösungsansatz.

### 3.3 Kommunikation

Die externe Steuerungseinheit muss mit der Maschine kommunizieren. Als Pflichtanforderung muss sowohl Start- wie auch Stoppsignal übertragen werden. Jedoch ist es möglich, dass die Maschine zusätzlich noch mit der Steuerungseinheit in Kontakt treten muss, falls diese die Aufgaben nicht selbst bewältigen kann. In den Abschnitten 3.4 und 3.9 sind die beiden Kommunikationspartner genauer beschrieben. Da beide Teilnehmer das gleiche Kommunikationsprotokoll unterstützen müssen ist die endgültige Entscheidung auch abhängig von der Teilnehmerwahl.

#### 3.3.1 ZigBee

ZigBee kommt bei Sensoren-Netzwerken zum Einsatz und ist optimiert für Low-Power Sensoren. Dies bedeutet auch, dass der Datendurchsatz möglichst gering sein soll. Diese Umsetzungsvariante scheidet aus. Es ist nicht für unseren Anwendungsfall konzipiert und zudem müssten beide Kommunikationspartner über spezielle Hardware Module verfügen.

#### 3.3.2 Bluetooth

Bluetooth ist ein weitverbreiteter Standard. Beinahe alle Smartphones und Tablets haben Bluetooth integriert, jedoch fehlt dieses Modul bei einigen Notebooks. Die Ansteuerung der Bluetooth-Module aus Hochsprachen wie Java oder Python stellt sich nicht als Problem heraus.

### 3.3.3 WLAN (eigenes Netz / Ad-hoc)

Mit WLAN hat man den gesamten TCP/IP Stack zur Verfügung. Diese Variante wird bevorzugt, denn man kann schnell eine Komponenten-Architektur auf Seite der Software aufziehen. Nahezu alle Mobile-Devices und Notebooks verfügen über einen WLAN-Adapter.

WLAN kann in verschiedenen Modi betrieben werden. Es wäre möglich mit einem Access Point ein eigenes Netzwerk zu erstellen in welchem sich die Teilnehmer anmelden können. Der andere Weg nennt sich Ad-hoc. Beide Teilnehmer können ohne ein zur Verfügung gestelltes Netz kommunizieren, denn diese erfolgt direkt untereinander. Jedoch ist der Konfigurationsaufwand höher siehe dazu <http://www.informationsarchiv.net/articles/1203/>. Zudem kann nicht jede WLAN Hardware mehrere Clients in einem solchen Ad-hoc Netzwerk aufnehmen. Daher ist die Variante 'eigenes Netz' zu favorisieren.

Siehe ausserdem Anhang D. Darin wurde erfolgreich versucht einen Webservice in Python auf dem Raspberry PI von einem Notebook aufzurufen (über TCP/IP).

## 3.4 Externe Steuerungseinheit

Die externe Steuerungseinheit ist mindestens dafür verantwortlich, dass diese ein Startsignal an die Maschine sendet und am Ende des Vorgangs ein Stoppsignal erhält. Weitere Gedanken gehen dahin, dass die externe Steuerungseinheit aufwändige Berechnungen zur Laufzeit vornehmen kann. Eine zusätzliche Idee ist es, dass diese Einheit Statusmeldungen von der Maschine erhält. Eine Statusmeldung könnte sein Örtung des Korbes abgeschlossen oder "Ball 4 abgefeuert".

Ausserdem muss die externe Steuerungseinheit das gewählte Kommunikationsprotokoll unterstützen.

### 3.4.1 Notebook

Ein Notebook als externe Steuerungseinheit ist zu favorisieren. Start-, Stoppsignal und Statusmeldungen zu empfangen sind kein Problem. Zusätzlich bietet heute ein handelsüblicher Notebook bereits sehr viel Rechenpower. Falls die Umsetzung aufwendige Berechnungen erfordert, sollte die Rechenleistung eines modernen Notebooks ausreichen.

### 3.4.2 Mobile

Mobile steht für ein Smartphone oder auch für ein Tablet. Auch hier sind Start-, Stoppsignal und Statusmeldungen problemlos möglich. Doch diese Variante gilt als "nice-to-have", denn ein solches Gerät verfügt heute noch nicht über die eventuell nötige Rechenleistung eines Notebooks für aufwendige Berechnungen.

## 3.5 Ortung des Korbs

Die Fragestellung lautet: "Wo ist der Standort des Korbes?". Um Standorte von Objekten herauszufinden gibt es mehrere Technologien. Die Erkennung hat folgende Herausforderungen:

- Distanz 2 Meter
- Objekt ist schwarz (guter Kontrast zu Hintergrund)
- Beleuchtung nicht konstant

### 3.5.1 Optik

Die Umsetzung mittels Optik erfordert auf der Seite der Software-Entwicklung erhöhten Aufwand im Gegensatz zu den anderen Umsetzungsmöglichkeiten. Die Erkennung von Objekten mit einer Kamera hängt zudem stark von den Umgebungsbedingungen ab. Weist ein Objekt z.B. zu wenig Kontrast zu seinem Hintergrund auf, wird die Erkennung schwierig. Auch ein Scheinwerfer ist ein möglicher Störfaktor, der zu falschen Messergebnissen

führen kann. Diese Probleme können mit einer Justierung der Kamera und der Algorithmen behoben werden. Von den möglichen Störfaktoren abgesehen liefert diese Art der Korrbortung sehr präzise Ergebnisse.

Für die Erstellung der Bilder kann eine herkömmliche Webcam verwendet werden. Für den Raspberry Pi wird außerdem eine günstige und praktische Kamera angeboten. Mittels der Kamera kann ein Bild fotografiert werden, welches ausgewertet wird um den Ort des Korbes zu bestimmen. Geprüft wurden Auswertungsmethoden auf Basis von Java und Python. In beiden Sprachen gibt es mehrere Bibliotheken, welche es erlauben Bildverarbeitung durchzuführen. Im Bereich Java wurde ImageJ (<http://imagej.nih.gov/ij/>) und in Python wurde OpenCV (<http://docs.opencv.org/>) angesehen. In beiden Programmiersprachen war es möglich Objekte zu erkennen (siehe Anhang B).

Eine weitere interessante Lösung für die Objekterkennung ist Pixy. Pixy vereint einen Bildsensor mit einem Prozessor und umgeht so das Problem mit den grossen Datenmengen die ein Bordcomputer (z.B. Raspberry Pi) verarbeiten muss. Zudem sind der Bildsensor extrem schnell und hochauflösend. Bei unseren Recherchen hat sich jedoch herausgestellt, dass Pixy mit dem Standartalgorithmus nur farbige Objekt erkennt. Für die Erkennung des Korbes muss der vom Hersteller gelieferte Algorithmus angepasst werden. Die Anpassung ist mit einem gewissen Aufwand und Risiken verbunden und muss gut überlegt werden. Detaillierte Informationen können dem Anhang C entnommen werden.

### 3.5.2 Ultraschall

Ultraschall bietet sich an als sehr einfache und preiswerte Technik für ein Ortungssystem. Verschiedene Tests zeigten, dass typische Ultraschallmodule für die vorgesehene Anwendung nicht optimal sind (siehe Anhang A).

### 3.5.3 Laser

Laser- bzw. Infrarotmessmodule bieten sich ähnlich wie Ultraschallmodule als einfache und preiswerte Ortungssysteme an. Auch hier haben praktische Testst gezeigt, dass einige Schwierigkeiten auftreten im projektnahen Umfeld (Mazlagić & Winz, 2014).

### 3.5.4 Wärmebild

Diese Umsetzung kommt aus mehreren Gründen nicht in Frage. Auf dem Markt gibt es keine Wärmebild Kameras, welche ins Budget passen. Andererseits wird sowohl der Korb wie auch die Wand dahinter dieselbe Wärme aufweisen.

### 3.5.5 Radar

Ein Radargerät sendet elektromagnetische Wellen aus und empfängt das reflektierte Echo. Aus den empfangenen, vom Objekt reflektierten Wellen kann man die Entfernung und die Bewegung eines Objektes bestimmen. Radar erkennt sehr gut Bewegungen auf den Sensor zu. Nicht so gut werden Bewegungen erkannt, die um den Sensor herum geschehen. Auch bei den grossen Elektronikhändler werden nur Bewegungsmelder mit einem Schaltausgang angeboten, der die Position eines Objektes nicht erkennen kann. Deshalb eignet sich die Technologie nicht für die Erkennung des Korbes.

## 3.6 Positionierung

In der Funktion Positionierung geht es darum wie die Maschine sich ausrichten soll.

### 3.6.1 Fix

Dies ist die einfachste Variante. Es werden keine zusätzliche Komponenten benötigt damit die Maschine sich in irgendeiner Form bewegt. Das setzt dann aber Voraus, dass bewegliche Achsen implementiert werden, damit die Maschine auf die Variabilität des Standortes des Korbes agieren kann.

### 3.6.2 Springt in Korb

Diese konkrete Umsetzung meint, dass die Maschine alle fünf Bälle packt und sich selbst mit den Bällen in den Korb transportiert bzw. springt. Diese Variante wird ausgeschlossen, da es uns gegenüber anderen möglichen Umsetzungen zu komplex erscheint.

### 3.6.3 Fliegen

Mit dieser Methode startet die Maschine ihre Propeller und transportiert die Bälle im Flug in den Korb. Auch diese Variante wird ausgeschlossen, da das Fliegen eine ganze Palette an Unbekannten mitbringt in welchem dem Team das Know-how fehlt. Beispielsweise ist ein stabiler Flug zu implementieren einfach gesagt als getan.

### 3.6.4 Geradeaus fahren / Rollt

Möglich ist es, dass die Maschine bis zur Grenze fährt um dem Korb näherzukommen. Auch dies wird ausgeschlossen, den der Aufwand dafür ist zu gross um nur 'etwas' näher zu sein. Es sollte sich lohnen mehr Aufwand in eine gute Abschussvorrichtung zu investieren.

## 3.7 Balltransport

Die Bälle müssen die Maschinen mit einer konstanten Geschwindigkeit verlassen und das Ziel möglichst genau treffen. Um die Bälle zu beschleunigen, gibt es verschiedene Möglichkeiten.

### 3.7.1 Luft

Eine Möglichkeit um Bälle zu beschleunigen besteht darin, Druckluft zu verwenden. Die Bälle befinden sich in einer Röhre und werden direkt mit Druckluft beschleunigt. Mit einer präzisen Regulierung des Druckes, der Luft und des Volumenstromes kann die nötige Kraft erreicht werden. Dieses Methode könnte geeignet sein für unseres Projekt.

### 3.7.2 Feder

Die Benutzung der Federn erlaubt ein einfache Berechnung der Kräfte und Energien. Die grösste Schwierigkeit bei der Benutzung der Federn ist die korrekte Wiederpositionierung beim Nachladen, da die Federn sich verbiegen können. Damit das Problem des Nachladens wegfällt, könnten mehrere Abschussrampen gebaut werden.

### 3.7.3 Drehräder

Drehräder sind die am meisten verwendeten Beschleunigungsarten in Ballwerfsystemen. Bei der Benutzung von Drehräder wird die tangentiale Geschwindigkeit der Drehräder auf den Objekt übertragen. Drehräder können auf verschiedene Arten gebaut werden: zwei Gegenseitig drehende Räder oder ein Rad mit einer Führung.

### 3.7.4 Zylinder (Pneumatisch)

Nach einer vertieften Recherche über die existierenden pneumatische Zylindermodelle, haben wir entschlossen dass kein Zylinder unsere Anforderungen erfüllt. Zudem ergeben sich keine Vorteile gegenüber der Variante mit der Druckluft.

### 3.7.5 Zylinder (Elektromagnetisch)

Nach einer vertieften Recherche über die existierenden elektromagnetische Zylindermodelle, haben wir entschlossen, dass kein Zylinder unsere Anforderungen erfüllt. Die Modelle die benutzbaren wären, brauchen zu viel Energie während einer kurzen Zeitdauer.

### 3.8 Ausrichtung

Um Zeit einzusparen und die Komplexität zu reduzieren, wird die Ballwurfmaschine in der Mitte des Spielfeldes positioniert. Aus diesem Grund muss das Gerät in horizontaler wie auch in vertikaler Richtung ausgerichtet werden können, nachdem der Korb positioniert wurde.

#### 3.8.1 horizontale Ausrichtung

Die horizontale Ausrichtung geschieht mit einem Schrittmotor. Dabei wird der Abwurfmechanismus so ausgerichtet, dass er in Richtung des Korbes zeigt. Um den Abwurfmechanismus und die Kamera miteinander zu verbinden, wird ein Visier vor die Kamera montiert. Das Visier wird auf den Korb ausgerichtet und somit ist auch der Abwurfmechanismus in der richtigen Position zum Abschuss.

#### 3.8.2 vertikale Ausrichtung

Damit möglichst wenige Motoren verbaut werden, ist es sinnvoll, die horizontale und vertikale Ausrichtung Mechanisch zu koppeln. Dabei wird die Ballwurfmaschine in horizontaler Richtung mit einem Schrittmotor gedreht. Bei dieser Drehbewegung wird der Abwurfmechanismus über eine Kurvenscheibe geführt, welche den Abschusswinkel einstellt. Dies ist nötig, damit die Tennisbälle mit konstanter Kraft abgeschossen werden können.

### 3.9 Bordcomputer

Die hier als Bordcomputer bezeichnete Komponente steht allgemein für eine Einheit, welche die *Intelligenz* des Gerätes enthält. Man assoziert mit dem Bordcomputer gewisse Fähigkeiten wie

- Kommunikation (*digitale Datenübertragung*)
- Aktoren-Ansteuerung
- Sensorauswertung
- Regelung

Die Evaluation eines solchen Bordcomputers kann sehr unterschiedlich ausfallen in Abhängigkeit von den gewählten Eigenschaften bzw. Ansprüchen an das Gerät. Hier gibt es einige typische Strategien bei der Auswahl, welche je nach Einsatzgebiet gewählt werden.

**Industrieanwendung** Die Einheit sollte möglichst Einsatzbereit eingekauft werden können und nur eine geringfügige Parametrierung erfordern — der Preis spielt keine Rolle.

**Massenproduktion** Die Einheit sollte weitestgehend eine Eigenentwicklung sein welche genau auf die Aufgaben der Applikation zugeschnitten ist — es muss möglichst preiswert sein.

**Einzelanfertigung** Die Einheit sollte möglichst aus gebrauchsfertigen Modulen (Funktionsgruppen) zusammengestellt werden, welche die jeweiligen Anforderungen optimal erfüllen — es soll schnell zum Ziel führen.

Für das vorliegende Projekt ist die dritte Strategie gewählt worden, welche sich daran orientiert, möglichst effizient ans Ziel zu gelangen. Diese *Effizienz* muss für eine konkrete Evaluation jedoch noch genauer spezifiziert werden. Im vorliegenden Fall ist dies definiert durch die folgenden Kriterien:

- Fähigkeiten
- Einsatzbereitschaft
- Verbreitungsgrad
- Leistung

- Modifizierbarkeit
- Dokumentation
- Preis

Unabhängig von den oben genannten Kriterien ist ein einschneidender Entscheid getroffen worden, welcher definiert, dass der Bordcomputer fähig sein muss ein vollwertiges und gängiges Betriebssystem zu unterstützen. Dieser Entscheid ist zustande gekommen mit der Absicht, die Disziplinen Elektrotechnik und Informatik so nahe wie möglich zusammenzuführen. Dies vereinfacht die Zusammenarbeit, verhindert redundante Arbeitsschritte und erleichtert das Teilen von Ressourcen. Insbesondere erhofft man sich durch diesen Entscheid eine relevante Entlastung der Elektrotechnik, da diese ohnehin mit minimaler Manpower in der Projektgruppe vertreten ist.

### 3.9.1 Raspberry Pi

Der Raspberry Pi ist ein Einplatinencomputer welcher explizit für das Experimentieren mit Soft- und Hardware entwickelt worden ist von der britischen Raspberry Pi Foundation (Anonymous, 2014a).

Im Gegensatz zu üblichen Computern hat dieser alle für den Betrieb relevanten Systemkomponenten fix installiert auf einer Platine. Hiervon ausgenommen ist die Spannungsversorgung und der Massenspeicher, welche jedoch mit gängigen Standards implementiert sind (MicroUSB bzw. MicroSD).

Der Raspberry Pi ist inzwischen sehr weit verbreitet. Dies spiegelt sich auch wider im grossen Angebot an Literatur, Projekten und Tutorials um und über den Raspberry Pi.



Abbildung 1: Der Raspberry Pi in der neusten Generation B+ (Anonymous, 2014d)

Der Raspberry Pi ist als erste Wahl für die Rolle des Bordcomputers gewählt worden aufgrund der gegebenen Kriterien, zu denen folgende Ergebnisse zustande kamen.

**Fähigkeiten** Der Raspberry Pi ist in der Lage direkte LowLevel Ansteuerung durchzuführen mittels einer Vielzahl von Programmiersprachen. Dies ist insbesondere gegeben durch die Tatsache, dass ein vollwertiges Betriebssystem eingesetzt wird. Durch den Einsatz eines modernen Betriebssystems sind auch viele Erweiterungen für Hard- und Software möglich, welche auch auf gewöhnlichen Computern eingesetzt werden. Diese Eigenschaften erfüllen das Kriterium der disziplinären Zusammenführung in optimaler Weise.

**Einsatzbereitschaft** Der Raspberry Pi ist ein Einplatinencomputer welcher (ausgehend vom Originalzustand wie er vom Hersteller angeboten wird) in wenigen Schritten in Betrieb genommen werden kann.

**Verbreitungsgrad** Der Raspberry Pi ist einer der populärsten Einplatinencomputer mit über 3 Millionen verkauften Exemplaren (Upton, 2014).

**Leistung** Der Raspberry Pi ist nicht der leistungsstärkste Vertreter unter den Einplatinencomputern, bietet dennoch gute Kennwerte und besitzt das beste Preis-Leistungsverhältnis (Anonymous, 2014b).

**Modifizierbarkeit** Der Raspberry Pi bietet ein vorbereitetes Interface an für eine Hardwareerweiterung, welches direkt die GPIO und Peripherie-IO zur Verfügung stellt. Auch für die Software ist eine hohe Modifizierbarkeit gewährleistet durch den Einsatz von freien und konfigurierbaren GNU/Linux Betriebssystemen.

**Dokumentation** Da der Raspberry Pi explizit zu Bildungs- und Experimentierzwecken entwickelt wurde, bietet der Hersteller auch relevante Daten an für dessen Einsatz (Adams, 2014). Die Entwicklung im Bereich der vorhandenen Literatur oder der frei verfügbaren Tutorials korreliert mit den Verkaufszahlen und es existieren grosse Benutzergemeinschaften.

**Preis** Die Ziele der Raspberry Pi Foundation bei der Entwicklung des Raspberry Pi waren Kompaktheit und ein niedriger Preis (Anonymous, 2014a). Diese Ziele zeigen sich auch im entstandenen Produkt, welches beim offiziellen Distributor Farnell für 32.57 SFr. erworben werden kann. Dies ist ein auffallend geringer Preis im Bereich der Einplatinencomputer mit ähnlichen Eigenschaften.

Für die Evaluation ist ein aktuelles Modell eines Raspberry Pi mit ArchLinux aufgesetzt worden. Dieses hat man überprüft auf die Lauffähigkeit gewisser Softwarekomponenten und Programmiersprachen, was bisher positive Ergebnisse geliefert hat. Die bisherigen Ergebnisse führen zudem zum Fazit, dass der Raspberry Pi ein für die Projektaufgabe vielseitig einsetzbaren Bordcomputer darstellt, der hinsichtlich seiner Einsatzfähigkeit nicht weiter untersucht werden muss.

Wie jede andere Komponente eines Systems, kann auch die des Bordcomputers falsch eingeschätzt werden. Um ein späteres Nachsehen zu vermeiden wird zu dieser Phase auch gleich evaluiert, wie man kritische Funktionen aus- bzw. umlagern kann, falls der Raspberry Pi in diesen versagen sollte.

### 3.9.2 LowLevel Fallback

Sollte der Fall eintreten, dass gewisse LowLevel Funktionalitäten nicht oder nur unter ungünstigen Bedingungen mit dem Raspberry Pi realisiert werden können, wird für diese Funktionen ein dediziertes LowLevel Mikrocontroller System hinzugezogen. Solche LowLevel Funktionen können Hardware-Interrupts oder direkte Peripheriekomponenten des Mikrocontrollers betreffen auf dem auch das Betriebssystem läuft. Diese LowLevel Funktionen verlangen im Extremfall ein Anpassen des Kernels bzw. das Erstellen eigener Kernelmodule (Treiberprogramme). Der Overhead, welcher sich beim Kerneldevelopment ergeben kann, ist im Verhältnis zum Ergebnis suboptimal. Um auf solche Probleme ein Fallback zu haben wird ein einfaches Mikrocontrollersystem evaluiert welches ohne Betriebssystem auskommt und dadurch wesentlich einfacher und dedizierter programmiert werden kann.



Abbildung 2: Das Freedomboard FRDM-KL25Z von Freescale

Aktuell wird das Freedomboard FRDM-KL25Z evaluiert. Diese wird im Rahmen des offiziellen Curriculum der Hochschule Luzern eingesetzt sowie auch in einem neuen aussencurricularem Kurs für Programmierersteiger. Die Vorteile, welche sich dadurch ergeben sind erheblich, denn es besteht eine grosse Benutzergruppe inklusive

Dozenten auf dem Campus Horw welche Einführungen, Kursunterlagen und sonstige Informationen und Hilfen anbieten können. Mit Hilfe dieser Unterlagen war es bereits möglich die Buildutils der Hochschule auf Linux zu portieren (Mazlagić, 2014). Mit einer Firmware des Herstellers und des OpenSDA ist es nun möglich ohne aufwändige Tools eigene Programme zu entwickeln und auf das Target zu flashen. Hierzu werden auch keine weiteren Geräte wie Debugger verwendet. Im Falle eines plötzlichen Bedarfs für ein Fallback, sind dies entscheidende Kriterien. Zudem ergibt sich wiederum die vorteilhafte Situation, dass auch die direkte Programmierung des Mikrocontrollers von den Informatik-Studenten erfolgen kann und die geringe Manpower der Elektrotechnik entlastet werden kann falls dies nötig ist.

### 3.9.3 Alternativen

Für den Raspberry Pi gibt es grundsätzlich eine Vielzahl von Alternativen (Anonymous, 2014b). Die Kriterien welche für die Definition einer geeigneten Alternative angewendet wurden sind die Folgenden.

- Systemkompatibilität
- Freie Hard- und Software
- Benutzergemeinschaft
- Leistungsfähigkeit
- Preis

Mit Hilfe dieser Kriterien hat sich eine Alternative für den Raspberry Pi herauskristallisiert.



Abbildung 3: BeagleBone Black (Halfacree, 2014)

**BeagleBone Black** Das BeagleBone Black stellt ein direktes Konkurrenzprodukt dar für den Raspberry Pi, denn es bewegt sich nahe an dessen Preissegment und bietet im Vergleich deutlich mehr Features (insbesondere im LowLevel Bereich) und eine etwas Leistungsstärkere Prozessorarchitektur. Zudem ist die gesamte Plattform offen im Sinne von OpenHardware, denn es werden offiziell sämtliche Daten zum BeagleBone Black angeboten. Diese Plattform ist zudem auch systemkompatibel mit dem Raspberry Pi, somit kann ebenfalls ein ArchLinux auf dem Gerät installiert werden und alle bestehende Software kann übertragen werden. Das BeagleBoard verfügt über eine kleinere Benutzergemeinschaft und auch weniger Literatur und sonstige Referenzen im Vergleich zum Raspberry Pi.

## 4 Übersicht Lösungsideen

Die Beschleunigung eines Balles lässt sich durch viele Methoden erreichen. Dementsprechend wurden während der Ideensuche auch viele unterschiedliche Möglichkeiten dazu gefunden. Hier sind die acht vernünftigsten Ideen aufgeführt.

### 4.1 Idee 1

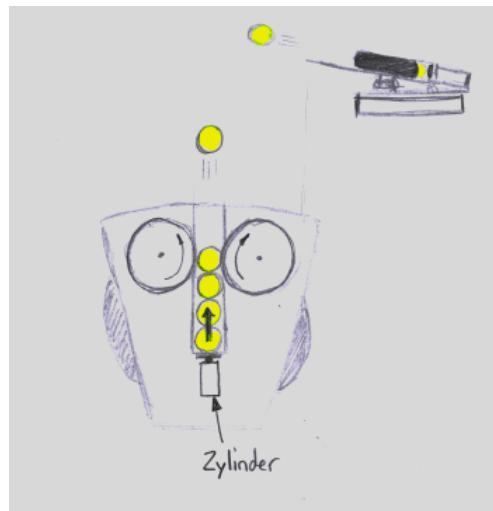


Abbildung 4: Aufbau mit zwei Rädern

Durch die Technologierecherchen im Internet wurde sehr schnell eine erste Idee gefunden. Die Bälle werden durch Einklemmen zwischen zwei entgegengesetzte drehenden Rädern in Drehrichtung beschleunigt.

### 4.2 Idee 2



Abbildung 5: Sprungfähiges Gefährt

Auch die zweite Idee beruht auf einer schon bestehenden Technologie. Analog dem SandFlea (Anonymous, 2013) genannten Roboter von Boston Dynamics, wäre die Maschine fähig in die Luft zu springen. Die Bälle könnten während dem gesamten Ablauf im Gerät verbleiben, da der Roboter mitsamt Bällen in den Korb springen würde.

#### 4.3 Idee 3

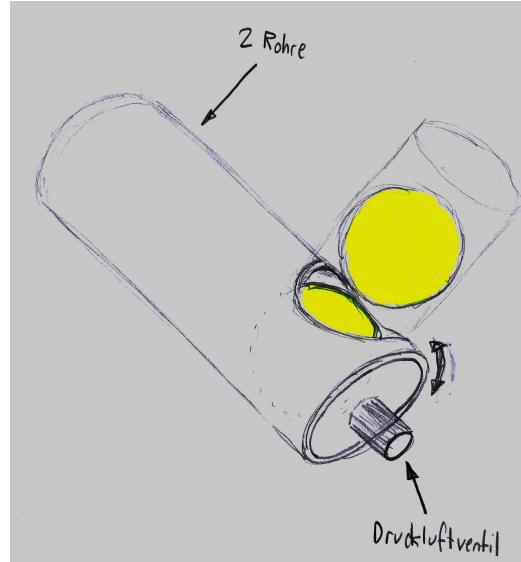


Abbildung 6: Druckluftrohr

Bei der dritten Idee wird das Wurfgeschoß mittels einer schnellen Druckbeaufschlagung durch Druckluft beschleunigt. Jeweils ein Ball wird in ein einseitig offenes Rohr mit einem Ventil am geschlossenen Ende gelegt. Bei einem schlagartigen Öffnen, des an ein Druckluftsystem angeschlossenen Ventiles, strömt Luft ins Rohr und beschleunigt den Ball.

#### 4.4 Idee 4

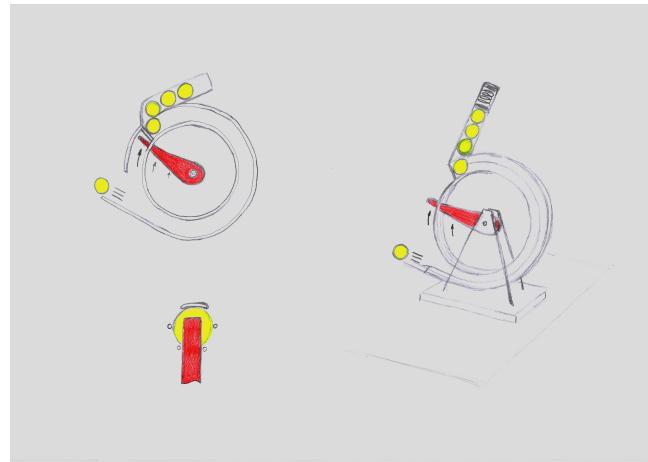


Abbildung 7: Drehrad mit Mitnehmer

Um einem allfälligen Schlupf, wie bei Idee 1 möglich, entgegenzuwirken, wird bei der vierten Idee der Ball durch einen schnell in Rotation versetzten Mitnehmer beschleunigt.

#### 4.5 Idee 5

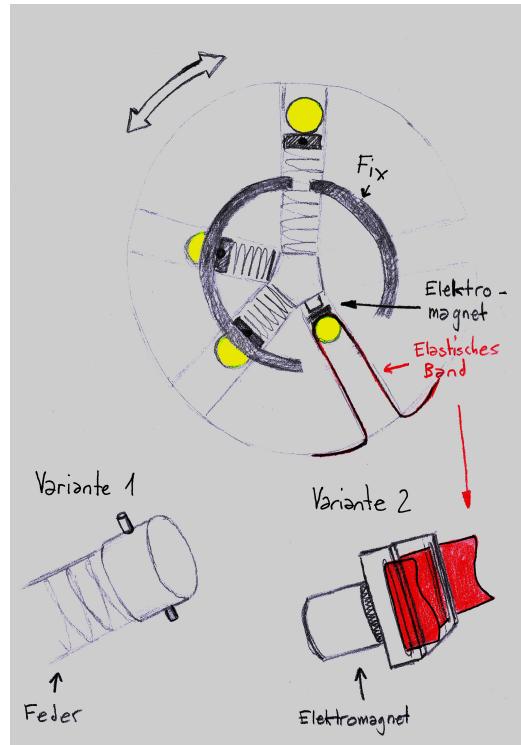


Abbildung 8: Rotierender Federbeschleuniger

Anders als bei den meisten bisherigen Ideen wird bei der fünften Idee auf ein Nachladeprozess verzichtet. Dies ist möglich, weil für jeden Ball ein beschleunigendes Element (Feder oder Gummiband) schon vorgespannt vorliegt. Für den ersten Ball wird die Vorrichtung ausgerichtet und für jeden weiteren Ball dann nur noch gedreht.

#### 4.6 Idee 6



Abbildung 9: Fallrohr

Diese Idee bedient sich der Schwerkraft als Beschleuniger. Die Bälle werden aus einer Höhe fallen gelassen und einer Führung entlang Richtung Korb gelenkt.

#### 4.7 Idee 7

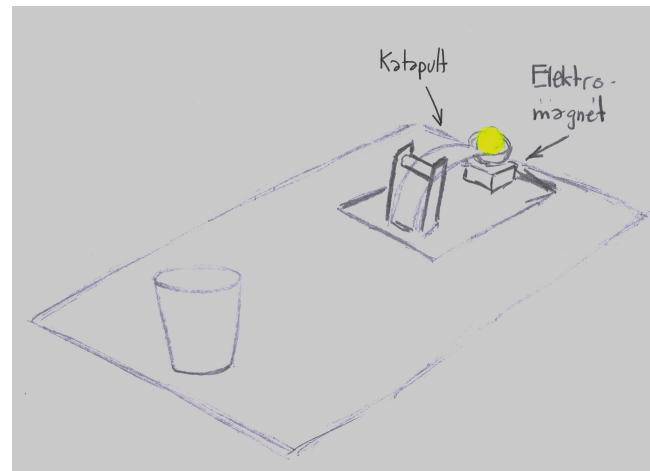


Abbildung 10: Katapult

Bei der siebten Idee werden die Bälle durch ein Katapult in den Korb geworfen. Das Katapult würde durch ein Elektromagnet in Wurfposition gehalten.

#### 4.8 Idee 8

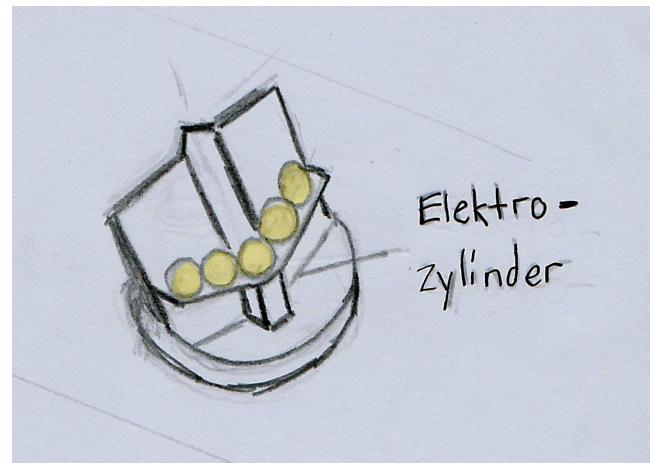


Abbildung 11: Elektrozylinder

Die achte Idee beinhaltet einen elektrischen Zylinder, der durch schnelles Ausfahren jeweils einen Ball beschleunigt.

#### 4.9 Auswahl

Da wir nur drei Ideen weiterverfolgen, überprüften wir bei jeder Idee die Realisierbarkeit und fokussierten uns so auf die für uns besten Lösungen. Zu aufwändig oder zu schwer machbar sind die Ideen 2, 5 und 7. Bei den Ideen 6 und 8 stellten wir nach tiefergrifenden Überlegungen fest, dass sie die Aufgabe nicht genügend gut erfüllen können. Wir entschieden uns daher für die Ideen 1, 3 und 4, welche nachfolgend genauer beschrieben werden.

## 5 Konzept Drehrad

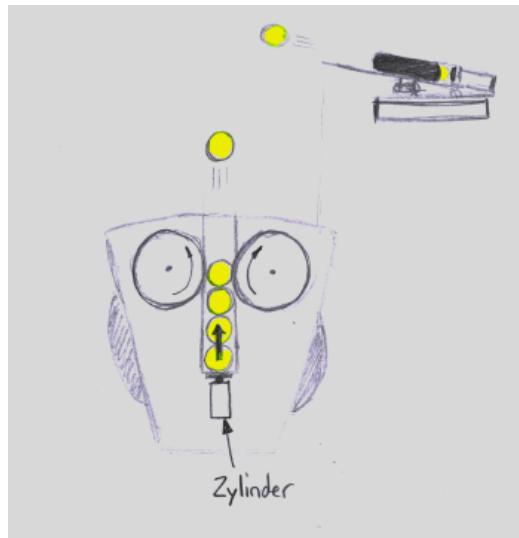


Abbildung 12: Aufbau mit zwei Rädern

### 5.1 Idee

Das dritte Konzept bedient sich einer unter Ballwurfmaschinen weit verbreiteten Methode Bälle zu beschleunigen. Zwei koaxial zueinander stehende Räder werden zum Drehen gebracht. Die Bälle müssen somit jeweils nur noch zwischen die Räder gebracht werden, um sich in Richtung der Drehrichtung zu beschleunigen. Als Variante dazu, würde sich ein Aufbau mit nur einem Drehrad anbieten.

### 5.2 Annahmen

Wir gehen davon aus, dass die Größenunterschiede der Bälle keinen signifikanten Einfluss auf die Wurfweite habe. Weiter schätzen wir die zwischen Rad und Ball entstehende Reibung als genügend stark für die Ballbeschleunigung ein.

### 5.3 Vor- und Nachteile

Vorteile:

- Schnelligkeit
- Einstellbare Wurfweite
- Einfacher Nachladeprozess

Nachteile:

- Bei einer Ausführung mit zwei Rädern müssen die Räder genau gleich schnell drehen, um am Ball entstehender Drall zu verhindern.
- Ungenügend genau einstellbare Drehzahlen könnten die Wurfweite schwer steuerbar machen.

## 6 Konzept Druckluft

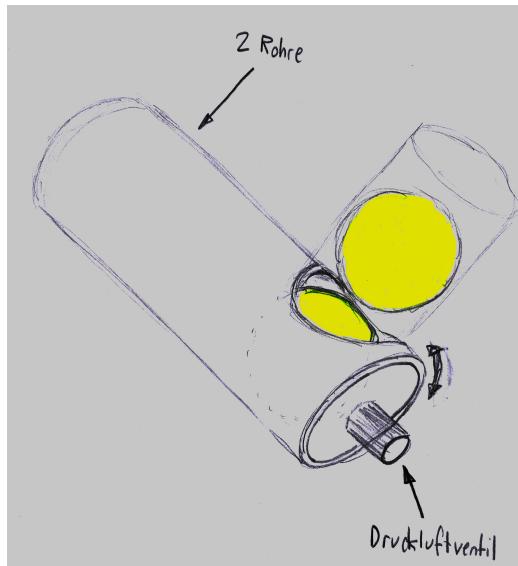


Abbildung 13: Druckluftrohr

### 6.1 Idee

Die Idee ist Druckluft zu benutzen, um ein Ball aufs mal durch ein Rohr zu beschleunigen. Es wird der zur Verfügung stehende Druckluftanschluss benutzt. Da die Bälle eine Differenz im Durchmesser bis zu 1 cm haben können, kann das Rohr nicht dicht sein. Aus diesem Grund werden wir ein Rohr benutzen dessen Durchmesser nur wenig grösser ist, als der grösste Ball. Wir werden das Rohr so modifizieren, dass es gerade nach der Aufladezone enger ist, so dass sich alle Bälle dort blockieren. Dank dieser Verengung blockiert sich der Ball und der Luftdruck hinter dem Ball wird grösser, bis genug Druck vorhanden ist um den Ball durch die Verengung zu drücken. Dieser Prozess wird dem Ball genügend Energie geben um bis zum Korb zu fliegen. Die Nachladung der Bälle kann mit einem zweiten Rohr gemacht werden, der sich um das erste Rohr dreht. Diese Rohre sind beide so geschnitten, dass wenn beide Öffnungen übereinstimmen ein neuer Ball ins erste Rohr fällt. Wenn die beide Schnitte nicht übereinstimmen ist das erste Rohr dicht.

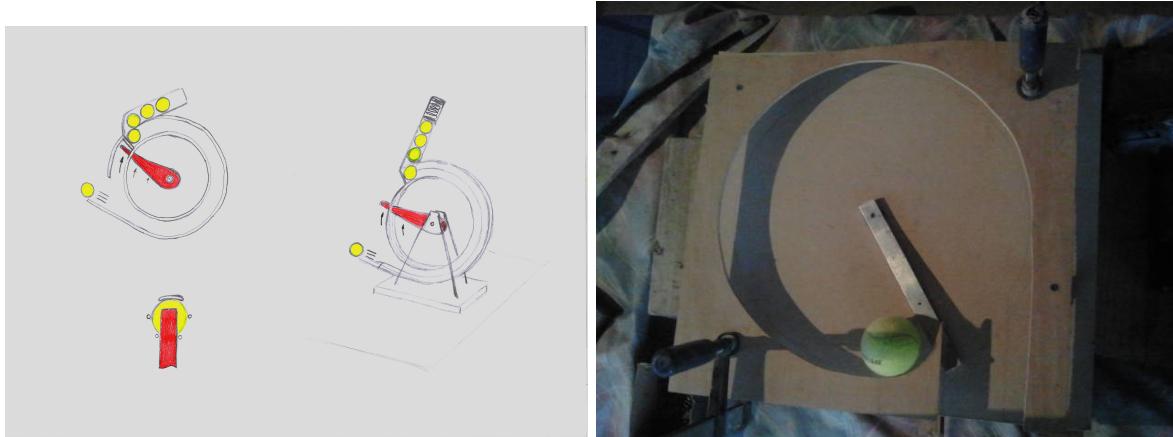
### 6.2 Annahmen

Um sicherzustellen dass das Konzept funktioniert, muss die Schrumpfung und der Ball eng anliegend, so dass die Druckluft genügend Druck aufbauen kann, um den Ball abzuschiessen. Deswegen darf der Durchmesser der Bälle nicht zu unterschiedlich sein.

### 6.3 Vor- und Nachteile

Falls der Durchmesser der Bälle zu unterschiedlich ist, kann es bei einigen Bällen passieren, dass sie nicht den Korb treffen, weil der Druck nicht genügend oder zu gross war. Weil das Rohr etwas grösser ist als die Bälle, gibt es die Möglichkeit, dass die Bälle beim Wurf links oder rechts weg fliegen oder sogar einklemmen. Der Vorteil dieses Konzept sind die Kosten. Da hier kein Motor für die Beschleunigung genutzt wird, werden die Kosten von dieses Teil ausfallen und auch die nötige Batterie für die Schrittmotoren kann deutlich kleiner sein.

## 7 Konzept Propeller



(a) Drehrad mit Mitnehmer

(b) Prototyp Drehrad mit Mitnehmer

Abbildung 14: Testaufbau

### 7.1 Idee

Bei dieser Variante werden die Tennisbälle mit einem Mitnehmer innerhalb von einer Umdrehung auf die nötige Abwurfgeschwindigkeit gebracht. Die Bälle werden nacheinander dem Mitnehmer zugeführt. Es besteht die Möglichkeit, dass der Teller mit dem Mitnehmer horizontal oder vertikal im Gerät eingebaut wird. Da eine hohe Umfangsgeschwindigkeit nötig ist, um den Ball auf die geforderte Geschwindigkeit zu beschleunigen müssen die Bälle in kurzer Zeit dem Mitnehmer zugeführt werden. Die Bälle werden in Umfangsrichtung mit einer Führung bis zur Abschussposition in Position gehalten. Als Antrieb wird ein Getriebemotor eingesetzt.

### 7.2 Annahmen

Ob das Konzept mit dem Mitnehmer funktioniert, wurde mit einem Prototyp überprüft. Dabei wurde ein Motor verwendet, welcher die geforderte Drehzahl nicht erreicht. Dementsprechend war die erreichte Wurfentfernung nur rund 40 cm. Dies sollte mit einem passenden Motor kein Problem mehr darstellen. Beim Versuchsaufbau wurde der Tennisball nur in Umfangsrichtung und auf einer Platte geführt. Dies führte dazu, dass der Ball teilweise gegen oben aus der Vorrichtung heraußsprang. Dieses Problem kann mit einer geschlossenen Bauweise behoben werden.

### 7.3 Vor- und Nachteile

**Vorteile:** Diese Art von Abwurfvorrichtung ist weitgehend unabhängig von der Ballgrösse (im vorgegebenen Bereich gemäss Aufgabenstellung). Die Bälle können in einem sehr geringen Zeitrahmen beschleunigt werden. Dadurch ist diese Variante sehr schnell, was die Abwurfgeschwindigkeit betrifft.

**Nachteile:** Beim Versuchsaufbau landeten die Tennisbälle in einem Feld von 30x30 cm. Somit liegt die Genauigkeit unter unseren Ansprüchen. Die benötigte Umfangsgeschwindigkeit ist so hoch, dass die Bälle nicht nur mittels der Gewichtskraft zugeführt werden können. Dies erfordert eine aufwendige Magazinkonstruktion. Damit für alle Tennisbälle identische Abwurfbedingungen vorhanden sind, muss der Motor zuerst auf die gewünschte Drehzahl gebracht werden. Dies führt wiederum zu einer aufwendigen Magazinkonstruktion. Durch die Konstruktion mit den Führungen wird das Gewicht der Konstruktion eher hoch.

## 8 Konzept-unabhängige Funktionen

Die Konzepte unterscheiden sich mechanisch, denn jedes einzelne Konzept verfolgt ein andere Idee um den Ball in den Korb zu befördern. Doch die Funktionen 'Bordcomputer', 'Kommunikation' und 'Ortung des Korbes' sind für alle vereinheitlicht. Jedes Konzept wird mit den gleichen konkreten Umsetzungen in diesen Bereichen ausgerüstet.

### 8.1 Übersicht

Auf der Grafik sieht man die gewählten Umsetzungen. In den nachfolgenden Kapitel sind diese begründet.

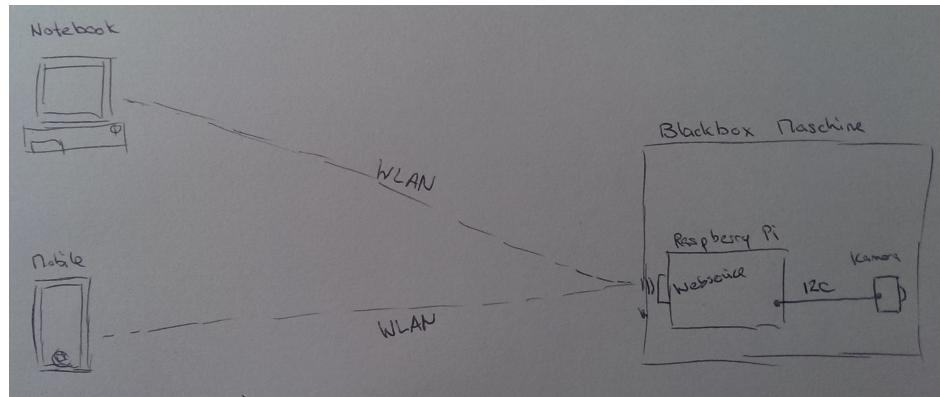


Abbildung 15: Übersicht konzept-unabhängige Funktionen

### 8.2 Bordcomputer

Der Einsatz eines Bordcomputers ist unabdingbar. Im Kapitel 3.9.1 wird die Verwendung des 'Raspberry PIs' empfohlen. Kostengünstig, weit verbreitet und ein grosser Funktionsumfang sind schlagkräftige Argumente. Möglicherweise stellen sich Low-Level Operationen als möglicher Knackpunkt heraus. Für dieses Problem kann schnell auf einen 'Freedomboard FRDM-KL25Z' zugegriffen werden, welcher die Aufgaben übernehmen kann (siehe Abschnitt 3.9.2).

### 8.3 Kommunikation

Die präferierte Variante für die Kommunikation ist das Benutzen des WLANs. Sowohl auf Seite der Maschine steht mit dem Bordcomputer ein TCP/IP fähiges Gerät zur Verfügung. Auf der anderen Seite der Steuerungseinheit besitzt heute nahezu jedes Gerät TCP/IP (Notebooks, Mobile Devices). Als möglicher Fallback kann auch Bluetooth gewählt werden, dies aber nur, wenn einer der beiden Teilnehmer kein WLAN unterstützt (siehe Abschnitt 3.3). Die Kommunikation an sich war in den Versuchen kein Problem und konnten problemlos durchgeführt werden (siehe Anhang D).

### 8.4 Ortung des Korbes

Bei der Ortung des Korbes wird auf Optik gesetzt. Denn die Versuche mit Ultraschall- sowie Lasermodulen waren nicht zufriedenstellend (siehe Abschnitt 3.5). An dem 'Raspberry PI' kann eine Kamera angeschlossen werden. Das Bild wird entweder direkt auf dem 'Raspberry PI' ausgewertet oder an eine externe Einheit geschickt, welche die benötigte Rechenpower zur Verfügung stellt.

## 9 Schluss

Innerhalb von vier Wochen hat das Team 39 drei Grobkonzepte erarbeitet, welche auf den Anforderungen und den Rechercheergebnisse der Wochen zuvor basieren. Die Analyse der benötigten Funktionen des Gesamt-Systems war der Startschuss für das vorliegende Lösungskonzept. Die Ergebnisse daraus wurden in einem morphologischen Kasten festgehalten. Jeder hat für jede Funktion konkrete Umsetzungsmöglichkeiten genannt, welche nachfolgend entsprechend detaillierter angeschaut wurden. Einige waren wortwörtliche 'Hirngespinster' (z.B. beim Balltransport: Beamen) andere wiederum wurden in die Grobkonzepte eingearbeitet.

Nachdem klar wurde wie man welche Funktion möglicherweise umsetzen kann, wurden Lösungsskizzen gezeichnet. So kamen einige interessante Ideen zusammen. Mit der Zeit und nach diversen Versuchen hat sich auch herauskristallisiert, dass es konzept-unabhängige Funktionen gibt. Diese Teilaufgaben werden in jedem Konzept gleich implementiert. Darunter fällt die 'Ortung des Korbes', der 'Bordcomputer' und die 'Kommunikation'. Aufwendige Versuche mit Laser-, Ultraschall- und Optikmodulen sowie Kommunikationstests wurden durchgeführt und lieferten wichtige Erkenntnisse. So war man überrascht, dass kostengünstige Laser- und Ultraschallapparaturen ungenau und störanfällig sind. Auf Seite der Mechanik hat man sich auf drei konkrete Lösungsideen fixiert und diese in unabhängigen Versuchen überprüft ob die gewählten Varianten im Grundsatz auch umsetzbar ist.

Im letzten Teil wurden die drei Grobkonzepte nach einem einheitlichen Raster bewertet. Der Gewinner ist das Konzept 'Drehrad'. Wichtiger als die Bewertung selbst ist, dass das ganze Team an das Gewinner-Konzept glaubt und mit einem guten Bauchgefühl die nächsten Aufgaben angehen kann. Nach einer intensiven und spannenden Zeit können wir nun drei Grobkonzepte mit gutem Gewissen präsentieren.

## Literatur

- Adams, J. (2014). *Raspberry Pi — Schematics for the various Raspberry Pi board versions*. <http://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/README.md>. (Online; abgerufen am 31. Oktober 2014)
- Anonymous. (2013). *Sandflea - leaps small buildings in a single bound*. [http://www.bostondynamics.com/robot\\_sandflea.html](http://www.bostondynamics.com/robot_sandflea.html). (Online; abgerufen am 6. November 2014)
- Anonymous. (2014a). *About us — The making of Pi*. <http://www.raspberrypi.org/about/>. Raspberry Pi Foundation. (Online; abgerufen am 31. Oktober 2014)
- Anonymous. (2014b). *ARM Einplatinencomputer* (Nr. 5). <http://www.raspberrypi.org/raspberry-pi-at-buckingham-palace-3-million-sold/>. (Online; abgerufen am 31. Oktober 2014)
- Anonymous. (2014c). *Freescale FRDM-KL25Z — Freedom Board (mbed-enabled) mit Kinetis L und ARM Cortex-M0+*. <http://elmicro.com/de/frdm-kl25z.html>. Elektronik Laden — Development Tools für Entwickler von Embedded Systemen. (Online; abgerufen am 31. Oktober 2014)
- Anonymous. (2014d). *Raspberry Pi B+ gets more USBs and microSD for the same tiny price*. <http://www.cnet.com/products/raspberry-pi-model-b-plus/>. c|net. (Online; abgerufen am 31. Oktober 2014)
- Halfacree, G. (2014). *BeagleBone Black Review*. <http://www.linuxuser.co.uk/reviews/beaglebone-black-review>. (Online; abgerufen am 2. November 2014)
- Mazlagić, E. (2014). *Material for a Crash Course in C Programming (Fork)*. <https://github.com/ninux/CrashCourseC>. (Online; abgerufen am 31. Oktober 2014)
- Mazlagić, E. & Winz, D. (2014). *Gp2y0a710k0f*. <https://github.com/accefa/doku/wiki/GP2Y0A710K0F-%28Lasermodul%29>. (Online; abgerufen am 3. November 2014)
- Upton, L. (2014). *Raspberry Pi at Buckingham Palace, 3 Million sold*. <http://www.raspberrypi.org/raspberry-pi-at-buckingham-palace-3-million-sold/>. Raspberry Pi Foundation. (Online; abgerufen am 31. Oktober 2014)

# Anhang

## A Ultraschallmodul HC-SR04

Das Ultraschallmodul HC-SR04 ist ein sehr verbreitetes Modul. Die Popularität dieses Moduls ist gegeben durch die folgenden Eigenschaften:

- unteres Preissegment (<30 CHF)
- Mittlerer Messbereich (2 - 400cm)
- Digitales Interface (TTL)
- Geringer Energiebedarf (75mW)
- Einfache Ansteuerung

### A.1 Prüfeinrichtung

Für die Prüfung des Ultraschallmoduls ist ein projektnahes Umfeld erstellt worden im Elektroniklabor B332b. Hierzu ist ein freistehender Tisch aufgestellt worden in Richtung des Ultraschallmoduls (siehe Abbildung 16). Auf diesem ist ein passender Eimer platziert worden. Dieser hat die folgenden Eckdaten:

Eigenschaft	Wert
Durchmesser oben	38cm
Durchmesser unten	33cm
Höhe	48cm
Farbe	Schwarz (matt)
Material	Kunststoff
Hersteller	Helit
Typ	61062

Tabelle 3: Testeimer Eckdaten

Um auch eine gewisse Relevanz und statistische Macht zu erreichen, sind sämtliche Prüfungen auch statistisch analysiert worden mit mindestens tausend Einzelmessungen. Hierzu sind die Geräte des Elektroniklabor verwendet worden. Dabei diente ein Frequenzgenerator als Ansteuerung und ein Digitales Speicheroszilloskop als Messmittel, welches auch gleich die statistische Auswertung durchführte.

### A.2 Messgenauigkeit

Um die Messgenauigkeit zu ermitteln ist der Testeimer direkt in Messrichtung des Ultraschallmoduls aufgestellt worden zu verschiedenen Abständen. Dieser Test zeigte auf, dass die Reichweite des Moduls lediglich 170 beträgt mit dem gegebenen Aufbau. Dies liegt deutlich unter dem idealen Wert aus dem Datenblatt.

Ein Referenzversuch mit einer Sperrholzplatte zeigte auf, dass das Signalrauschen deutlich verringert wird im Vergleich zum Testeimer. Bei einem Abstand vom 180cm wurde ein mittlerer Impuls von 10.45ms mit einer Standardabweichung von  $9.7\mu\text{s}$  gemessen.

### A.3 Messempfindlichkeit

Die Messung der Messempfindlichkeit beschreibt, wie das Modul Objekte erfassen kann, welche nicht direkt anvisiert werden. Hierzu sind Messungen jeweils in  $5^\circ$  Schritten durchgeführt worden. Der notierte Wert ist jener, bei welchem das Modul den Testeimer gerade noch erkennt, d.h. bei kleineren Distanzen wird der Testeimer erkannt.

Interessant ist hier, dass das Ultraschallmodul einen blinden Bereich aufweist bei welchem der Testeimer nicht erkannt wird. Dieser Bereich liegt zwischen einem Winkel vom  $25^\circ$ - $30^\circ$  und zu einer Distanz von 75-90cm.

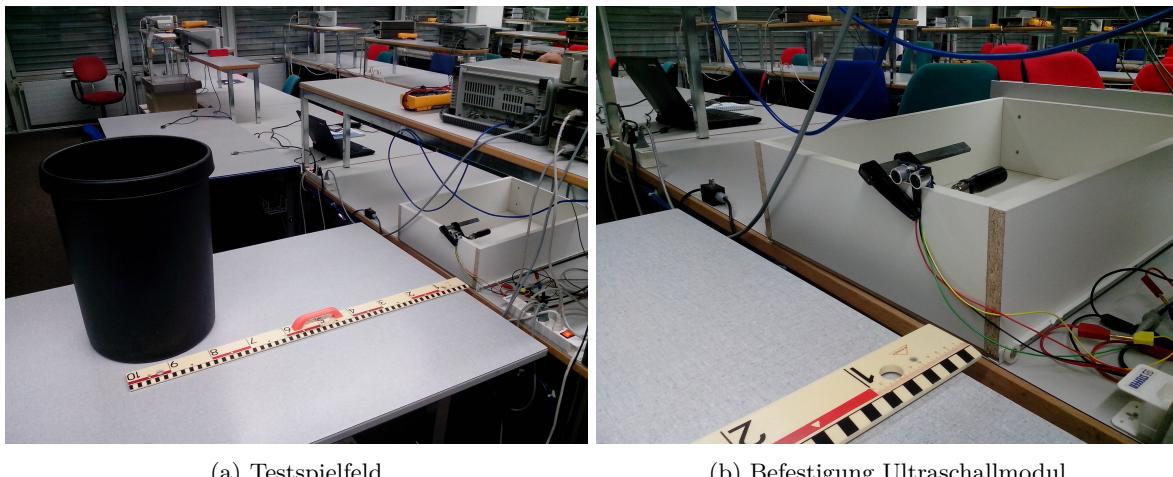


Abbildung 16: Testaufbau

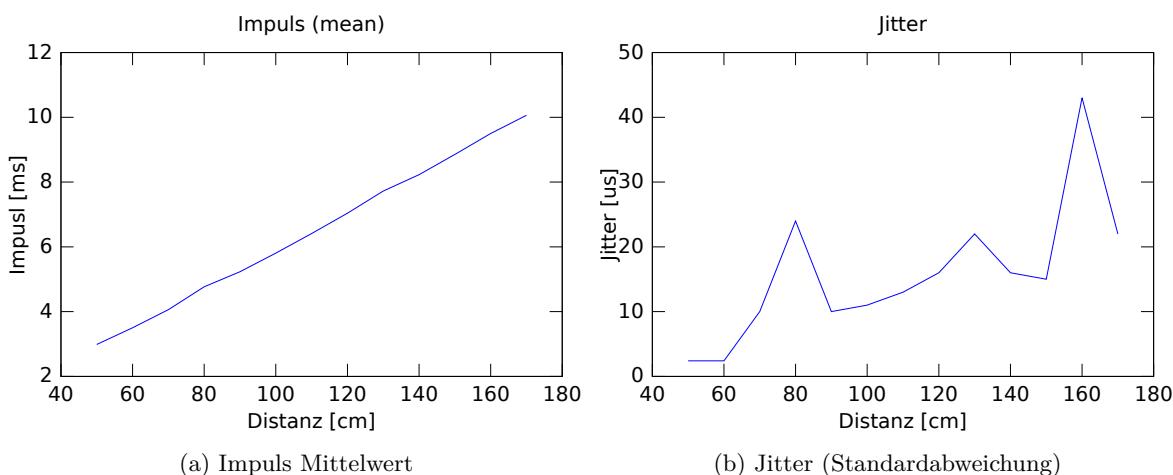


Abbildung 17: Messergebnisse der Impulsmessung

Als einfacher Versuch zur Eingrenzung des Messbereichs ist je ein selbsterstelltes Papierrohr mit einer Länge von 3cm mittels eines Kupferdrahtes auf die Sonden des Ultraschallmoduls fixiert worden. Die Testreihe wurde dann in gleicher Weise wiederholt und zeigte, dass sich der Messbereich deutlich reduziert auf ca. 15°. Dieser Effekt kann extrapoliert werden in der Weise, dass längere Röhren zu einem weiteren Einschnüren des Messbereich führen sollten. Diese Hypothese wurde jedoch keinem Test unterzogen.

#### A.4 Fazit

Das Ultraschallmodul HC-SR04 bietet auch unter projektnahen Bedingungen eine hohe Linearität (siehe Abbildung 17a). Dies bedeutet, dass sich das Modul sehr wohl eignet für eine direkte Distanzmessung mit geringer Toleranz. Dies funktioniert jedoch nur in einem Bereich von bis zu 170cm, was deutlich unter dem Datenblattwert liegt und knapp ist für die geplante Applikation.

Die weiteren Messungen zeigten zudem, dass das Modul insbesondere bei indirekter Messung des Teststellers deutliche Schwierigkeiten aufweist. Die polare Darstellung dieser Messergebnisse zeigt, dass ein eigentlicher Messkegel nicht vorhanden ist (siehe Abbildung 18). Dies eliminiert den Einsatz des Moduls für das Abtasten

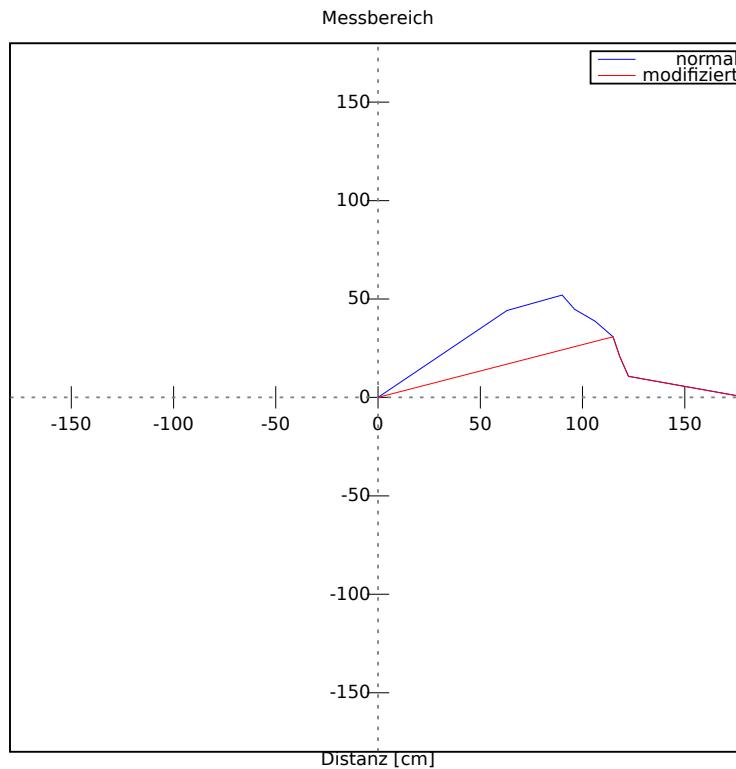


Abbildung 18: Seitliche Messempfindlichkeit

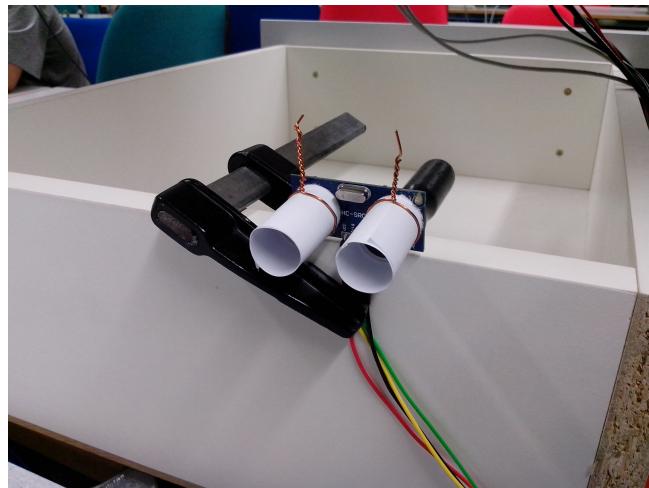


Abbildung 19: Modifiziertes Ultraschallmodul

des Spielfelds nach dem Zielobjekt, insbesondere dann, wenn zu dem nichtidealen seitlichen Messverhalten des Moduls auch Reflexionen und dadurch ausgelöste Interferenzen berücksichtigt werden. Für das Abtasten des Spielfeldes gilt es ein Messverfahren zu verwenden, welches entweder unempfindlich ist gegenüber Reflexionen und sonstigen Interferenzen von Aussen oder sehr punktuell operieren kann, was Reflexionen von vornherein ausschliesst. Ein typischer Vertreter solcher Messmittel sind Laserbasierte Geräte.

## B Bildverarbeitung

Die Bildverarbeitung wurde auf Basis von Java und Python geprüft. In beiden Programmiersprachen gibt es viele Bibliotheken, welche Bildoperationen unterstützen. Es beginnt bei einfachen Funktionen wie das Zuschneiden oder Komprimieren von Bildern bis hin über integrierte Objekterkennung.

### B.1 Java

In Java wurde die Bibliothek ImageJ<sup>1</sup> angeschaut. ImageJ ist modular aufgebaut und lässt sich sowohl Nutzen als auch Erweitern. Zweiteres macht die Bibliothek attraktiv, denn es gibt bereits einige Erweiterungen im Netz. Eine solche Erweiterung nennt sich FeatureFinder<sup>2</sup>. Diese Erweiterung verlangt als Input das zu suchende Objekt und das Bild auf welchem das Objekt gefunden werden muss. In folgendem Beispiel wird gezeigt wie einfach sich ein Bild zuschneiden lässt.

Listing 1: Bild zuschneiden mit ImageJ

```

1 public void crop(ImagePlus imp, int targetWidth, int targetHeight) {
2     ImageProcessor ip = imp.getProcessor();
3     int cropX = ip.getWidth() / 2;
4     int cropY = ip.getHeight() / 2;
5     ip.setRoi(cropX, cropY, targetWidth, targetHeight);
6     ip = ip.crop();
7     BufferedImage croppedImage = ip.getBufferedImage();
8     ImageIO.write(croppedImage, "jpg", new File("cropped.jpg"));
9 }
```

Es ist auch möglich das Bild, auf welchem der Korb zu suchen ist, mit eigenen Low-Level Bildverarbeitung Operationen zu verarbeiten. Beispielsweise kann von links nach rechts die Pixel untersucht werden. Dort wo der Korb steht weisen die einzelnen Pixel eine anderen Charakteristika auf. Auch das Auslesen eines Pixels auf dem Bild ist simpel.

Listing 2: Auslesen eines Pixels mit ImageJ

```

1 private void readSomePixel() {
2     ImagePlus im = new ImagePlus("C:/tmp/ALLUSB/Bilder/Scannen0001.jpg");
3     ImageProcessor imp = im.getProcessor();
4
5     int[] rgb = new int[3];
6     imp.getPixel(5, 5, rgb);
7     System.out.println(Arrays.toString(rgb));
8 }
```

### B.2 Python

Soll das Bild direkt auf dem Raspberry Pi verarbeitet werden, wird Python als Programmiersprache verwendet. Diese Sprache ist schnell zu erlernen und bringt auch alle notwendigen Funktionen mit. Zudem wird Python vom Raspberry Pi optimal unterstützt.

Um Bilder mit Python zu bearbeiten wird die Python Image Library<sup>3</sup> verwendet. Die Bibliothek ist ein Bestandteil der Programmiersprache und kann ohne zusätzliche Installationen verwendet werden. Die Python Image Library bietet alle Funktionen die von einer modernen Bildbearbeitungsbibliothek erwartet werden. Sie bietet auch diverse Filter an und die Pixel eines Bildes lassen sich einzeln bearbeiten.

<sup>1</sup><http://imagejdocu.tudor.lu/>

<sup>2</sup><http://imagejdocu.tudor.lu/doku.php>

<sup>3</sup><http://effbot.org/imagingbook/overview.htm>

Für eine erweiterte Objekterkennung wird OpenCV<sup>4</sup> unterstützt. OpenCV ist eine Sammlung von Bildalgorithmen mit denen praktisch jegliche Anwendung von einfachen Objekterkennungen bis hin zu maschinellem Lernen möglich ist. Mithilfe von OpenCV kann man so eine Gesichtserkennung mit wenigen Codezeilen realisieren.

Listing 3: Gesichtserkennung mit OpenCV

```

1 import numpy as np
2 import cv2
3
4 face_cascade = cv2.CascadeClassifier('haarcascade_face.xml')
5 eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
6
7 img = cv2.imread('sachin.jpg')
8 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
9 faces = face_cascade.detectMultiScale(gray, 1.3, 5)
10 for (x,y,w,h) in faces:
11     cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
12     roi_gray = gray[y:y+h, x:x+w]
13     roi_color = img[y:y+h, x:x+w]
14     eyes = eye_cascade.detectMultiScale(roi_gray)
15     for (ex,ey,ew,eh) in eyes:
16         cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
17
18 cv2.imshow('img',img)
19 cv2.waitKey(0)
20 cv2.destroyAllWindows()

```

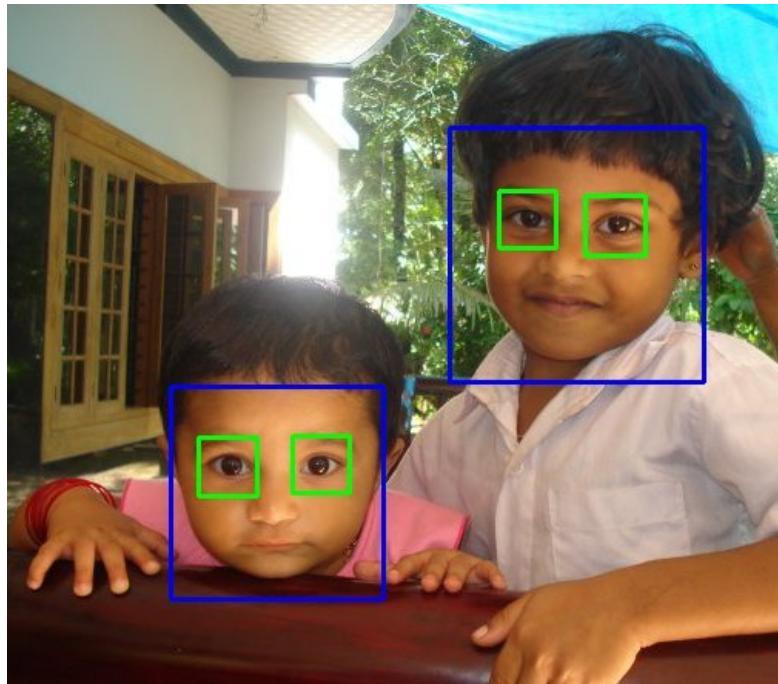


Abbildung 20: Das Ergebnis der Gesichtserkennung

<sup>4</sup>[http://docs.opencv.org/trunk/doc/py\\_tutorials/py\\_tutorials.html](http://docs.opencv.org/trunk/doc/py_tutorials/py_tutorials.html)

## C Kameramodul Pixy (CMUcam5)

Pixy ist ein schneller Bildsensor, welcher Objekte erkennen kann und das Resultat (Position/Fläche) an einen beliebigen Controller sendet.

Bildsensoren sind sehr flexibel, haben jedoch den Nachteil dass grosse Datenmengen anfallen und ein starker Prozessor für die Berechnung der Algorithmen benötigt wird. Pixy umgeht dieses Problem indem es einen Bildsensor mit einem Prozessor kombiniert. Pixy verarbeitet die Bilder vom Bildsensor und schickt die relevanten Informationen an den Controller (z.B. Raspberry Pi).

### C.1 Eckdaten

- Prozessor: NXP LPC4330, 204 MHz, dual core
- Bild Sensor: Omnivision OV9715, 1/4", 1280x800
- Framerate: 50Hz
- Sichtfeld: 75° horizontal (auf 2m Abstand 3m Sicht), 47° vertikal (auf 2m Abstand 1.7m Sicht)
- Linsen Typ: standard M12 (several different types available)
- Stromverbrauch: 140 mA (typisch)
- Spannung: 5V (bis zu 10V)
- RAM: 264K bytes
- Flash: 1M bytes
- Verfügbare Schnittstellen: UART serial, SPI, I2C, USB, digital, analog
- Abmessungen: 5.3cm x 4.4cm x 3.5cm

### C.2 Objekterkennung

Pixy berechnet den Farbton und die Sättigung von jedem RGB-Pixel und filtert so das gewünschte Objekt heraus. Der Algorithmus ist robust gegen Lichtveränderungen (laut Hersteller). Ein Objekt kann in Pixy per Knopfdruck eingelernt werden. Es können bis zu 7 Objekte eingelernt werden.

### C.3 PixyMon

PixyMon ist die Konfigurationssoftware für Pixy. Verbindet man seinen Laptop per USB-Kabel mit Pixy erhält man in PixyMon eine Liveansicht der Bilder (gut für Debugging).

### C.4 Schnittstelle

Alle 20ms sendet Pixy eine Liste von *Object Blocks* über die serielle Schnittstelle (UART, SPI, I<sup>2</sup>C) an den Controller. Die Object Blocks werden nach Objektgrösse sortiert. Der Aufbau des Blockes kann der unteren Abbildung entnommen werden.

### C.5 Erweiterbarkeit

Die gesamte Hardware und Software von Pixy ist frei verfügbar (GPL) und erweiterbar. Die Firmware von Pixy ist in C geschrieben und PixyMon basiert auf dem QT-Framework.

Data label	Size	Description
sync	2 bytes	Synchronization tag, indicates start of object block
checksum	2 bytes	Simple checksum for the rest of the object block data
signature	2 bytes	Color signature of object (1-7) or list of signatures for color code
x_position	2 bytes	X position of object center in image
y_position	2 bytes	Y position of object center in image
size	2 bytes	Size (area) of object in image
angle	2 bytes	Angle of object (optional, only available for color codes)

Abbildung 21: Aufbau eines Datenblockes

## C.6 Versuch

Da Pixy sehr vielversprechend unser Problem mit der Objekterkennung lösen konnte wurde ein simpler Versuch durchgeführt. Bei normalen Tageslicht (bedeckter Himmel) wurde versucht den schwarzen Abfalleimer vor einer weissen Wand einzulernen. Dabei stellte sich heraus das der Algorithmus von Pixy die Farbsättigung zur Erkennung der Objekte verwendet. Der Abfalleimer ist schwarz und wird deshalb nicht von Pixy erkannt. Ein farbiges Objekt, wie z.B. ein Tennisball wird sehr zuverlässig und schnell erkannt. Das Bild zeigt wie Pixy einen Leuchtstift erkennt. Der gesamte Einlernprozess hat gerade einmal 5 Minuten in Anspruch genommen.

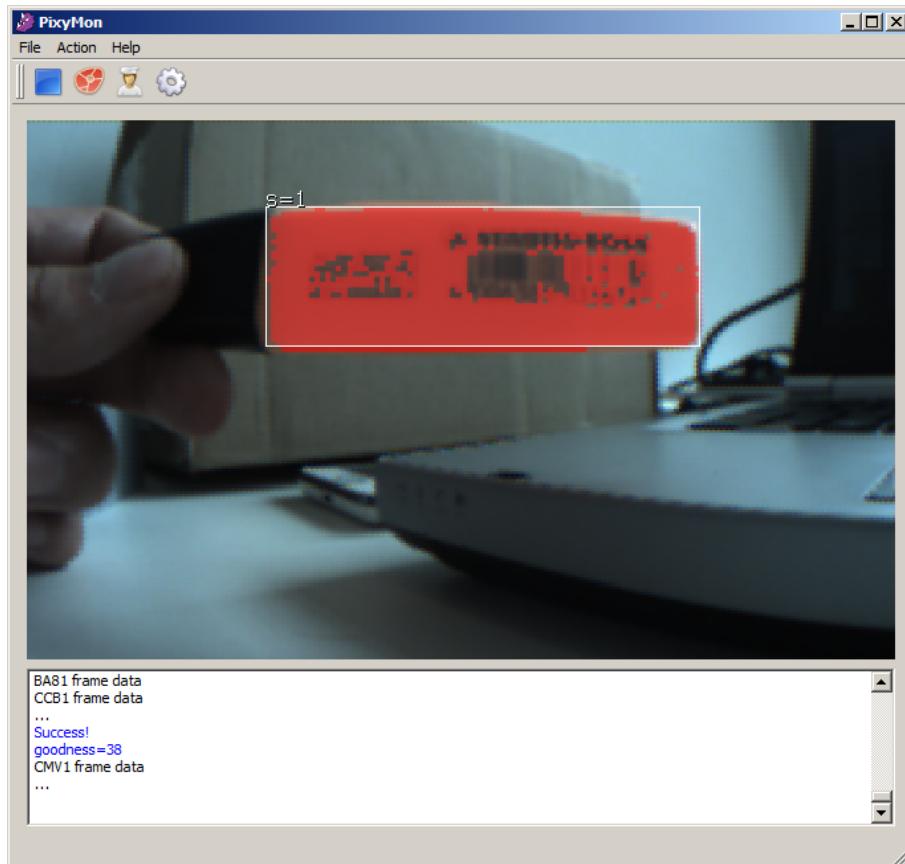


Abbildung 22: Pixy erkennt einen Leuchtstift

## D Kommunikation

Der ganze Versuch basiert auf dem Tutorial von dreamsyssoft.com <http://www.dreamsyssoft.com/blog/blog.php?/archives/6>Create-a-simple-REST-web-service-with-Python.html>

### D.1 Ziel

Das Ziel des Versuches ist es über den TCP/IP Stack von einem externen Gerät Funktionen auf dem Raspberry PI aufzurufen.

### D.2 Versuch

In einem ersten Schritt wurde auf dem Raspberry PI ein Python Webservice implementiert und gestartet. Die Wahl Python ist dadurch begründet, da mittels Python relativ einfach serielle Ports angesteuert werden können.

Dazu haben wir auf dem Raspberry PI die Python-Library web.py installiert. Diese Library ermöglicht es einen Webservice in Python zu schreiben. Nachfolgend haben wir den Webservice auf dem PI angelegt und anschließend gestartet.

```

1 #!/usr/bin/env python
2 import web
3 import xml.etree.ElementTree as ET
4
5 tree = ET.parse('user_data.xml')
6 root = tree.getroot()
7
8 urls = (
9     '/users', 'list_users',
10    '/users/(.*)', 'get_user',
11 )
12
13 app = web.application(urls, globals())
14
15 class list_users:
16     def GET(self):
17         output = 'users:[';
18         for child in root:
19             print 'child', child.tag, child.attrib
20             output += str(child.attrib) + ','
21             output += ']';
22         return output
23
24 class get_user:
25     def GET(self, user):
26         for child in root:
27             if child.attrib['id'] == user:
28                 return str(child.attrib)
29
30     if __name__ == "__main__":
31         app.run()
```

Nun läuft der Webservice unter dem Port 8080 und kann von einem xbeliebigen Gerät im gleichen Netzwerk aufgerufen werden.

### D.3 Fazit

Es ist problemlos möglich über TCP/IP auf das Raspberry PI zuzugreifen. Sogar Webservices lassen sich in Python implementieren, welche von irgendeinem Client aufgerufen werden können.

## E Berechnungen

Damit wir wissen, mit welchen Geschwindigkeiten wir die Bälle abwerfen müssen, haben Berechnungen gemacht. Hier sind nur die Hauptergebnisse angezeigt; alle Berechnungen befinden sich hier.

Mit der Benutzung und der Anwendung der schrägen Wurf Formel, könnten wir die nötige Geschwindigkeit ausrechnen und damit die notwendige Energie.

Y0: Abwurfhöhe	Y: Korbhöhe g	a (Winkel)	a (rad)	cos	sin	tan	x1: 1.75 m	x2: 1.86 m	V1(x1 abstand)	V2 (x2 abstand)	Energie J	Elastische Konstante k mit l = 0,1m
0,3	0,4	9,81	5 0,0873	1 0,09	0,09	1,75	1,85	16,88	16,54	8,550887	1710,177452	
0,3	0,4	9,81	10 0,1745	0,98	0,17	0,18	1,75	1,85	8,62	8,75	2,227804	445,5608109
0,3	0,4	9,81	15 0,2618	0,97	0,26	0,27	1,75	1,85	6,61	6,74	1,309264	261,852724
0,3	0,4	9,81	20 0,3491	0,94	0,34	0,36	1,75	1,85	5,63	5,76	0,950457	190,0914443
0,3	0,4	9,81	25 0,4363	0,91	0,42	0,47	1,75	1,85	5,05	5,18	0,766211	153,242286
0,3	0,4	9,81	30 0,5236	0,87	0,5	0,58	1,75	1,85	4,69	4,81	0,660025	132,0050395
0,3	0,4	9,81	35 0,6108	0,82	0,57	0,7	1,75	1,85	4,46	4,57	0,59678	119,3560986
0,3	0,4	9,81	40 0,6981	0,77	0,64	0,84	1,75	1,85	4,33	4,44	0,56187	112,2374087
0,3	0,4	9,81	45 0,7854	0,71	0,71	1	1,75	1,85	4,27	4,38	0,546239	109,2477273
0,3	0,4	9,81	50 0,8727	0,64	0,77	1,19	1,75	1,85	4,28	4,39	0,549309	109,8617288
0,3	0,4	9,81	55 0,9599	0,57	0,82	1,43	1,75	1,85	4,36	4,48	0,570922	114,18436
0,3	0,4	9,81	60 1,0472	0,5	0,87	1,73	1,75	1,85	4,53	4,65	0,614989	122,9978041
0,3	0,4	9,81	65 1,1345	0,42	0,91	2,14	1,75	1,85	4,80	4,93	0,690722	138,1444971
0,3	0,4	9,81	70 1,2217	0,34	0,94	2,75	1,75	1,85	5,22	5,37	0,818255	163,650951
0,3	0,4	9,81	75 1,309	0,26	0,97	3,73	1,75	1,85	5,90	6,07	1,046067	209,2133455
0,3	0,4	9,81	80 1,3963	0,17	0,98	5,67	1,75	1,85	7,12	7,32	1,521159	304,23186
0,3	0,4	9,81	85 1,4835	0,09	1	11,4	1,75	1,85	9,97	10,25	2,980813	596,1625803
0,4	0,4	9,81	5 0,0873	1	0,09	0,09	1,75	1,85	9,94	10,22	2,965911	593,1821536
0,4	0,4	9,81	10 0,1745	0,98	0,17	0,18	1,75	1,85	7,08	7,28	1,505832	301,1664722
0,4	0,4	9,81	15 0,2618	0,97	0,26	0,27	1,75	1,85	5,86	6,02	1,03005	205,01
0,4	0,4	9,81	20 0,3491	0,94	0,34	0,36	1,75	1,85	5,17	5,31	0,801237	160,2473328
0,4	0,4	9,81	25 0,4363	0,91	0,42	0,47	1,75	1,85	4,73	4,87	0,672317	134,4634776
0,4	0,4	9,81	30 0,5236	0,87	0,5	0,58	1,75	1,85	4,45	4,58	0,5947	118,939929
0,4	0,4	9,81	35 0,6109	0,82	0,57	0,7	1,75	1,85	4,27	4,39	0,548078	109,6156315
0,4	0,4	9,81	40 0,6981	0,77	0,64	0,84	1,75	1,85	4,18	4,29	0,52297	104,5940182
0,4	0,4	9,81	45 0,7854	0,71	0,71	1	1,75	1,85	4,14	4,26	0,515025	103,006
0,4	0,4	9,81	50 0,8727	0,64	0,77	1,19	1,75	1,85	4,18	4,29	0,52297	104,5940182
0,4	0,4	9,81	55 0,9599	0,57	0,82	1,43	1,75	1,85	4,27	4,39	0,548078	109,6156315
0,4	0,4	9,81	60 1,0472	0,5	0,87	1,73	1,75	1,85	4,45	4,58	0,5947	118,939929
0,4	0,4	9,81	65 1,1345	0,42	0,91	2,14	1,75	1,85	4,73	4,87	0,672317	134,4634778
0,4	0,4	9,81	70 1,2217	0,34	0,94	2,75	1,75	1,85	5,17	5,31	0,801237	160,2473328
0,4	0,4	9,81	75 1,309	0,26	0,97	3,73	1,75	1,85	5,86	6,02	1,03005	205,01
0,4	0,4	9,81	80 1,3963	0,17	0,98	5,67	1,75	1,85	7,08	7,28	1,505832	301,1664722
0,4	0,4	9,81	85 1,4835	0,09	1	11,4	1,75	1,85	9,94	10,22	2,965911	593,1821536

Abbildung 23: Berechnungen der Benötigten Abwurfgeschwindigkeiten und der nötigen Federkonstante

Um eine Idee von den nötigen Drehmomenten, die unseres Motor braucht, wurden die benötigten Trägheitsmomente berechnet. Anhand der Trägheitsmomente konnte die Leistungen und die Drehzahlen berechnet werden.

Trägheitsmoment		Trägheitsmoment	
Holkugel		Holkugel	
$I = (2/3) * m * r^2$		$I = (2/3) * m * r^2$	
m	0,06000 Kg	m	0,06000 Kg
r	0,03500 m	r	0,03500 m
I	0,00005 Kg*m^2	I	0,00005 Kg*m^2
$I_p = I_{cm} + m * h^2$		$I_p = I_{cm} + m * h^2$	
d	0,30000 m	d	0,35000 m
h	0,11350 m	h	0,13850 m
$I_p$	0,00082 Kg*m^2	$I_p$	0,00120 Kg*m^2
$I_{stange} = 1/3 * m * l^2$		$I_{stange} = 1/3 * m * l^2$	
l	0,11350 m	l	0,11350 m
m	0,30000 Kg	m	0,30000 Kg
I	0,00129 Kg*m^2	I	0,00129 Kg*m^2
Igesamt	0,00211 Kg*m^2	Igesamt	0,00249 Kg*m^2
v	5 m/s	v	5 m/s
Drehmoment		Drehmoment	
$I_p * \alpha$	0,652 N*m	$I_p * \alpha$	0,516 N*m
alpha	308,865 rad/s^2	alpha	207,425 rad/s^2
Drehzahl		Drehzahl	
M	420,674 rpm	M	344,740 rpm
w	44,053 rad/s	w	36,101 rad/s
f	7,011 Hz	f	5,746 Hz

Abbildung 24: Notwendige Drehzahlen und Drehmomente vom Motor für das Konzept Propeller

Für das Konzept Drehräder wird ein Motor mit einer Drehzahl von mindestens 1000rpm benötigt. Die genauen Berechnungen sind auch im Berechnungsfile auf GitHub zu finden.