

Mohne – Mobile Drohne

Abschlussbericht

Entwicklung eines autonomen Fahrzeuges für extraterrestrische Erkundungsmissionen

Eingereicht bei:

**Fachhochschule Kufstein Tirol Bildungs GmbH**

**Studiengang Smart Products & Solutions**

Verfasser:

**Markus Wiesmüller**

**1710809534**

Abgabedatum:

**02.03.2018**

# Eidesstattliche Erklärung

*„Ich erkläre hiermit, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst und in der Bearbeitung und Abfassung keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe. Die vorliegende Bachelorarbeit wurde noch nicht anderweitig für Prüfungszwecke vorgelegt.“*

Kufstein, den 11. August 2017

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Markus Wiesmüller

Table of Contents

[Eidesstattliche Erklärung II](#_Toc508391957)

[List of Figures V](#_Toc508391958)

[List of Tables VI](#_Toc508391959)

[List of Abbreviation VII](#_Toc508391960)

[Kurzfassung VIII](#_Toc508391961)

[Abstract IX](#_Toc508391962)

[1. System Beschreibung 1](#_Toc508391963)

[1.1 Mechanisches Konzept 1](#_Toc508391964)

[1.1.1 Flugmodus 1](#_Toc508391965)

[1.1.2 Fahrmodus 3](#_Toc508391966)

[1.1.3 Funktionsprinzip der Propeller 3](#_Toc508391967)

[1.2 Sensing Concept 5](#_Toc508391968)

[1.2.1 Picamera 5](#_Toc508391969)

[1.2.2 Ultraschallabstandssensor 5](#_Toc508391970)

[1.2.3 Gesamtsystem (big picture) 6](#_Toc508391971)

[2. Materialien und Methoden 6](#_Toc508391972)

[2.1 Materialien 7](#_Toc508391973)

[2.2 Manufacturing Technologies 7](#_Toc508391974)

[2.3 Software Tools 10](#_Toc508391975)

[2.3.1 Autodesk Inventor Professional 10](#_Toc508391976)

[2.3.2 Autodesk EAGLE 11](#_Toc508391977)

[2.3.3 Arduino IDE 11](#_Toc508391978)

[2.3.4 Raspbian 12](#_Toc508391979)

[2.3.5 Python 12](#_Toc508391980)

[2.3.6 Samba 14](#_Toc508391981)

[2.3.7 Maschinensoftware 14](#_Toc508391982)

[3. System life cycle 15](#_Toc508391983)

[3.1 Systemanforderungen 15](#_Toc508391984)

[3.1.1 Zweck 15](#_Toc508391985)

[3.1.2 Teststrecken 16](#_Toc508391986)

[3.1.3 Ziel 16](#_Toc508391987)

[3.1.4 Einschränkungen 17](#_Toc508391988)

[3.2 Designalternativen 17](#_Toc508391989)

[3.3 Model the Sytem 18](#_Toc508391990)

[3.3.1 System and Subsystem 18](#_Toc508391991)

[3.3.2 Make or Bye 19](#_Toc508391992)

[3.3.3 Sequenzdiagram 20](#_Toc508391993)

[3.3.4 System Architecture 21](#_Toc508391994)

[3.3.5 Functional Decomposition 21](#_Toc508391995)

[3.3.6 Failure Mode and Effects Analysis (FMEA) 21](#_Toc508391996)

[3.4 Integrate System Components 23](#_Toc508391997)

[3.5 Launch the System 23](#_Toc508391998)

[3.5.1 Timeline 23](#_Toc508391999)

[4. Diagrams (obligate) 24](#_Toc508392000)

[4.1 Use Case Diagram (Anwendungsfalldiagramm) 24](#_Toc508392001)

[4.2 Structural Model (Strukturmodelle) 24](#_Toc508392002)

[4.3 Interaction Model (Interaktionsmodell) 24](#_Toc508392003)

[4.4 Behavior Model (Verhaltensmodell) 25](#_Toc508392004)

[5. Testing Software (optional) 25](#_Toc508392005)

[6. Version Control (e.g. Gitlab – obligate) 25](#_Toc508392006)

[7. Design of PCB (obligate) (Max) 25](#_Toc508392007)

[7.1 Breadboard 25](#_Toc508392008)

[7.2 Schematic 25](#_Toc508392009)

[7.3 Board 25](#_Toc508392010)

[pdb- power distribution board hat einen batteriespannungsregler (bec) regeln von 11,1 auf 12 v für arduino und auf 5 v für pi ohne bec regelung zu jedem esc 26](#_Toc508392011)

# ****List of Figures****

[Abb. 1: Visualisierung eines GMs nach Schallmo 6](#_Toc490200752)

[Abb. 2: Bestandteile der Definition von GMI 7](#_Toc490200753)

[Abb. 3: Blöcke zur Entwicklung und Umsetzung der digitalen Transformation 9](#_Toc490200754)

[Abb. 4: Themengebiete der Digitalen Revolution nach MIT und Capgemini Consulting 12](#_Toc490200755)

[Abb. 5: Framework zur Digitalen Transformation von GM 28](file:////Users/Markus/Hochschule/Semester%206/BAII/Bachelorarbeit%20II/Kopie-WING_2014_Wiesmüller_Markus_BA_II.docx#_Toc490200756)

# ****List of Tables****

[Tab. 1: Ausgewählte Definitionen im Kontext der Digitalen Transformation 4](#_Toc490195388)

[Tab. 2: Frameworks zur Digitalen Transformation 15](#_Toc490195389)

[Tab. 3: Thesen aus der Literaturrecherche 20](#_Toc490195390)

# ****List of Abbreviation****

|  |  |
| --- | --- |
| **IoT** | **Internet of Things** |
| **GM** | **Geschäftsmodell** |
| **GMI** | **Geschäftsmodell-Innovation** |
| **DBT** | **Digital Business Transformation** |
| **IT** | **Informationstechnik** |
| **EY** | **Ernst & Young** |
| **ROI** | **Return on Investment** |
| **KPI** | **Key Performance Indicator** |
| **MIT** | **Massachusetts Institute of Technologie** |
| **IKT** | **Informations- und Kommunikationstechnik** |
| **HR** | **Human Resources** |
| **BDI** | **Bundesverband der Deutschen Industrie e.V.** |
| **CEO** | **Chief Executive Officer** |
| **F&E** | **Forschung und Entwicklung** |

# Kurzfassung

sdafsdf

# Abstract

Summarize and describe the project

Management concept?

What concept was apllied?

How was it fabricated?

Performance in the tracks?

Lessons learned?

# System Beschreibung

## Mechanisches Konzept

Die Mohne ist ein sogenannter „Hybrid-Quadrocopter“, der sowohl zum Fahren als auch zum Fliegen konzipiert ist. Durch diese Hybrid-Nutzung ergeben sich große Vorteile. Auf dem Boden spart der Roboter Energie (im Verhältnis zur Fortbewegung in der Luft), während er in der Luft auch meterhohe Hindernisse überwinden kann. Nachfolgend wird das mechanische Konzept beider Varianten näher erläutert sowie und in Abbildung 5 veranschaulicht.

### Flugmodus

Die Mohne gehört zur Gruppe der Quadrocopter, bestehend aus vier Motoren die in H-Konfiguration angeordnet sind. Bei der H-Konfiguration sind die vier Motoren in 45° zur Flugrichtung versetzt, wobei der Abstand vom Kreuzmittelpunkt zu den einzelnen Motoren immer gleich groß ist.

Vier Propeller sind fest an den Motoren montiert und erzeugen den Auftrieb der Drohne, welcher sich über die Drehzahl der einzelnen Motoren verändern lässt. Übersteigt die durch die sich drehenden Propeller erzeugte Auftriebskraft die Anziehungskraft der Erde bzw. des Mars, so beginnt die Drohne zu schweben. Eine Erhöhung bzw. Verringerung der Drehzahl hat ein Steigen bzw. ein Sinken der Drohne zufolge.

Dadurch ergibt sich ein relativ einfacher mechanischer Aufbau, in dem es außer den Rotoren der Motoren keine weiteren beweglichen Teile für den Flugmodus gibt.

Überkreuz liegende Rotoren drehen dabei in die gleiche Richtung (Abbildung 1) und heben somit das um die Mitte der Drohne übertragene Drehmomente auf.

Bei exakt gleicher Drehgeschwindigkeit aller Propeller und keinen äußeren Störeinflüssen, steht die Drohne also exakt lotrecht über dem Boden. Dies eliminiert die Notwendigkeit einer Gierstabilisierung.

Um ein nach vorne fliegen zu ermöglichen, drehen sich die vorderen Rotoren langsamer als die hinteren, was ein nach vorne abkippen der Drohne zur Folge hat (Abbildung 2). Entsprechend wird ein nach hinten fliegen ermöglicht, indem die hinteren Rotoren langsamer als die vorderen drehen (Abbildung 3).

Abbildung 1: Quadrocopter: Motordrehrichtung

A

B

C

D

y

x

A

B

C

D

z

Abbildung 2: Quadrocopter: Vorwärtsbewegung

Abbildung 3: Quadrocopter: Rückwärtsbewegung

A

B

C

D

z

### Fahrmodus

Um mit der Mohne auch fahren zu können, wurde eine Radaufhängung mit Achsen an die Drohne montiert. Allerdings werden dabei die Räder nicht von einem Motor betrieben, stattdessen wird die Schubleistung von den Propellern der Mohne erzeugt. Die hinteren Tragflächen des Quadrocopters lassen sich im Fahrmodus über Zahnräder um nahezu 90° nach vorne und hinten verstellen (Abbildung 4). In diesem Modus drehen sich nur die hinteren Propeller, wodurch ein Vorschub ähnlich wie bei einem Flugzeug bzw. Schiff erzeugt wird. Ein Abbremsen erfolgt indem die Propeller auf Umkehrschub gestellt werden. Sie erzeugen so einen nach vorne gerichteten Schub. Die Verstellung erfolgt dabei elektrisch.

+90°

-90°

Abbildung 4: Tragfläche mit Propeller: Verstellbar um +90° und -90°

### Funktionsprinzip der Propeller

Wichtig für das Verständnis des Funktionsprinzips der Drohne ist, dass man sich grob phänomenologisch die Wirkungsweise von Propellern veranschaulicht, weil sich schon aus recht simplen Überlegungen Hinweise darauf ergeben, wie die Propeller sinnvollerweise ausgelegt werden müssen. Die Wirkungsweise eines Propellers, dessen Aufgabe sowohl im Flug- als auch im Fahrmodus darin besteht eine Drehleistung von optimal in eine Schubleistung umzusetzen, kann aus zwei Anschauungen gewonnen werden, die dann in jeweils unterschiedlichen Berechnungsmodellen münden:

* **Impulsbetrachtung:** Um einen Schub T nach vorne bzw. nach oben zu erzeugen, muss der Propeller die Luft nach hinten bzw. nach unten beschleunigen. Die Änderung des Impulses entspricht dabei dem Propellerschub. Die Theorie, die auf diesem Modell aufbaut, ist die Strahltheorie.
* **Tragflügelbetrachtung:** Die Wirkungsweise eines Propellers entspricht der eines Tragflügels. Der Auftrieb entspricht dabei dem Schub, der Widerstand dem Moment. Das Tragflügelmodell erklärt die Propellerwirkung an den einzelnen Flügelschnitten und ist als Traglinentheorie (nur radiale Diskretisierung) oder Tragflächentheorie bekannt.

Aus diesen beiden Wirkungsweisen lassen sich schon einige grundlegende Dinge qualitativ ableiten, ohne dass genaue mathematische Begründungen nötig sind. Die Auslegung der Propeller erfolgt in Kapitel …???

**Motor**

**+**

**Propeller**

**Motor**

**+**

**Propeller**

**Motor**

**+**

**Propeller**

**Motor**

**+**

**Propeller**

**Rad**

**Rad**

**Rad**

**Rad**

**Servo**

**Servo**

Abbildung 5: Quadrocopter: Mechanisches Konzept

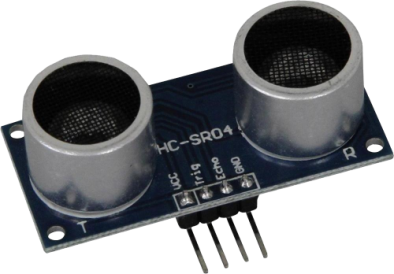
## Sensing Concept

Um die Mohne autonom fliegen bzw. fahren zu lassen und Hindernisse und Ziele zu erkennen, wurde eine Kamera und ein Ultraschallsensor verbaut. Nachfolgend werden beide Sensoren näher beschrieben und das System im Gesamtkontext betrachtet.

### Picamera

### Ultraschallabstandssensor

Mit dem Ultraschallabstandssensor (Abbildung 6) kann man mittels eines Ultraschalllautsprechers und eines Mikrofons den Abstand berührungslos zu einem Objekt messen. Das Prinzip (Abbildung 7) basiert darauf, dass die Schallgeschwindigkeit in der Luft bei gleichbleibender Temperatur nahezu konstant bleibt - bei 20°C beträgt sie 343,2m/s. Aus diesem Fakt kann man die Abstandsmessung in eine Zeitmessung überführen, welche dann von Mikrokontrollern einfach übernommen werden kann.



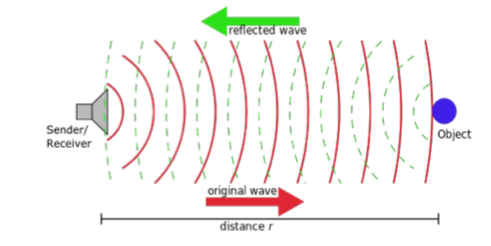
 Abbildung 6: Ultraschallabstandssensor

Abbildung 7: Funktionsprinzip Ultraschallabstandssensor

### Gesamtsystem (big picture)

Die unter Punkt 1.2.1 und 1.2.2 beschriebenen Sensoren unterstützen den Hybrid-Quadrocopter sich in seiner Umgebung zurechtzufinden und sich autonom fortzubewegen, Ziele zu erkennen und Hindernisse gekonnt zu überfliegen. Die Picamera dient dazu ein Ziel zu erkennen und dieses anzusteuern, mit Hilfe des Ultraschallsensors kann ein Hindernis erkannt werden, woraufhin in den Flugmodus gewechselt wird, um das Hindernis zu überfliegen und um nach einer bestimmten Zeit x zu landen und wieder in den Fahrmodus zu wechseln. Wie in Abbildung 8 veranschaulicht sind dabei beide Sensoren mit einem Microcontroller verbunden.

Abbildung 8: Quadrocopter: Sensorkonzept

**Controller**

**Kamera**

**Ultraschallsensor**

# Materialien und Methoden

Um das Projekt zu realisieren und die Mohne zu entwickeln, wurde Seitens des Auftraggebers verschiedene Materialien und Werkzeugmaschinen sowie ein Budget von 40€ zur Verfügung gestellt.

Es gab viele Faktoren, die dazu beitrugen, wie dieser Hybrid-Quadrocopter konstruiert werden sollte. Das Team musste zuerst berücksichtigen, dass die Drohne Testflüge absolvieren musste, von denen sie gegebenenfalls nicht in einem Stück zurückkehrt. Hinzu kam, dass das Gewicht der für die Herstellung der Drohne verwendeten Materialien sorgfältig geprüft werden musste, damit das Gewichtslimit und auch die Größe für den Wettbewerb nicht überschritten wurde. Schließlich musste das Design auf eine Art und Weise erstellt werden, welches leicht mit den zur Verfügung stehenden Materialien und Werkzeugmaschinen umgesetzt werden konnte. Zudem musste das Design die Elektronik vor Wasser schützen. Aufgrund des entscheidensten Faktors, die für die Produktion zur Verfügung stehenden Zeit, wurde entschieden, Lasercutten, 3D-Druck, Schneiden, Schleifen, Fräsen und verbinden mit Schrauben zu verwenden. Dieser Prozess wurde als am schnellsten und am leichtesten reproduzierbar bestimmt, falls erforderlich. Der erste Schritt in diesem Prozess war das Design der Drohne zu entwerfen. Dafür wurde die in Abschnitt 2.3.1 beschriebene CAD-Software Autodesk Inventor verwendet. Im nächsten Schritt wurde bestimmt, wie die einzelnen Komponenten hergestellt und welche Materialen dafür verwendet werden sollten. Komponenten wir Arduino, Raspberry Pi und Sensoren wurden zur Verfügung gestellt. Da jedoch kein Breadboard verwendet werden durfte, wurde mit Hilfe der in 2.3.2 beschriebenen Software ein Leiterplatten-Layout erstellt, welches mit Hilfe einer Fräsmaschine gefertigt wurde.

In diesem Kapitel werden die bereitgestellten Materialien aufgelistet, die verfügbaren Maschinen vorgestellt sowie eine Vorstellung der verwendeten Software.

## Materialien

|  |  |
| --- | --- |
| **Material** | **Maße in mm (l x b x h)** |
| Holz | 800 x 600 x 3,8 |
| Plexiglas | 600 x 400 x 3 |

## Manufacturing Technologies

|  |  |
| --- | --- |
| **Werkzeug / Maschine** |  |
| Bohr-Fräsmaschine Optimum BF 16 Vario |  |
| Schraubstock |  |
| Schleifmaschine Holzstar BTS 150 |  |
| Bandsäge Metabo BAS 317 Precision |  |
| Digitaler Messschieber Scala 150mm |  |
| Ultimaker 2 3D-Drucker |  |
| Ultimaker 3 Dual Extruder 3D-Drucker |  |
| Fräsmaschine Roland SRM-20 |  |
| Lötstation Weller WS 81 |  |
| Lasergravierer Trotec Speedy 360 |  |
| Metallsäge |  |
| Schraubenzieher |  |
| Schraubenschlüssel |  |

## Software Tools

Um das Projekt zu realisieren, wurden zweierlei verschiedene Softwareprogramme verwendet. Zum einen wurde Autodesk Inventor Professional verwendet, um ein 3D CAD-Modell der Mohne zu erstellen, zum anderen wurde für die Konstruktion der Leiterplatte das Programm EAGLE der Firma Autodesk verwendet. Beide Programme werden nachfolgend kurz beschrieben.

### Autodesk Inventor Professional

Autodesk Inventor ist eine von Autodesk entwickelte 3D-CAD-Software für die Volumenmodellierung, mit der digitale 3D-Prototypen erstellt werden können. Es wird für 3D-Konstruktion, Design-Kommunikation, Werkzeugherstellung und Produktsimulation verwendet. Diese Software ermöglicht es Benutzern, genaue 3D-Modelle zu erstellen, um Produkte zu entwerfen, zu visualisieren und zu simulieren, bevor sie gebaut werden.

Die Software ermöglicht zudem 2D- und 3D-Daten in eine einzige Designumgebung zu integrieren und so eine virtuelle Darstellung des Endprodukts zu erstellen, mit der die Form, Passform und Funktion des Produkts vor dem Bau validieren werden kann. Autodesk Inventor enthält leistungsstarke parametrische, direkte Bearbeitungs- und Freiform-Modellierungswerkzeuge sowie Multi-CAD-Übersetzungsfunktionen und DWG Zeichnungen nach Industriestandard, wodurch Entwicklungskosten gesenkt, schnellere Time-to-Market realisiert und großartige Produkte entwickelt werden können.

### Autodesk EAGLE

EAGLE ist eine skriptfähige EDA-Anwendung (Electronic Design Automation) mit Schaltplan-Erfassung, Leiterplatten-Layout, Auto-Router und CAM-Funktionen (Computer Aided Manufacturing). EAGLE steht dabei für Easy Applicable Grafischer Layout-Editor.

Das Programm bietet eine grafische Benutzeroberfläche und ein Menüsystem mit mehreren Fenstern für die Bearbeitung, das Projektmanagement und die Anpassung der Schnittstellen- und Designparameter. Das System kann über Maus, Tastaturkürzel oder durch Eingabe bestimmter Befehle in einer eingebetteten Befehlszeile gesteuert werden. Mehrere Wiederholungsbefehle können zu Skriptdateien kombiniert werden. Es ist auch möglich, Entwurfsdateien unter Verwendung einer EAGLE-spezifischen objektorientierten Programmiersprache zu untersuchen.

### Arduino IDE

Arduino bringt eine eigene integrierte Entwicklungsumgebung (IDE) mit, die auf Wiring IDE basiert. Dabei handelt es sich um eine Java-Anwendung, die für die gängigen Plattformen Windows, Linux und MacOS kostenlos verfügbar ist. Sie basiert auf der IDE von Processing, einer auf die Einsatzbereiche Grafik, Simulation und Animation spezialisierten Entwicklungsumgebung. Die Arduino-IDE bringt einen Code-Editor mit und bindet gcc als Compiler ein. Zusätzlich werden die avr-gcc-Library und weitere Arduino-Librarys eingebunden, die die Programmierung in C und C++ stark vereinfachen. Arduino kann verwendet werden, um eigenständige interaktive Objekte zu steuern oder um Softwareanwendungen auf Computern zu interagieren (z.B. Adobe Flash, Processing, Max/MSP, Pure Data, diverse Skriptsprachen, etc.)

Für ein funktionstüchtiges Programm genügt es, zwei Funktionen zu definieren:

* setup() – wird beim Start des Programms (entweder nach dem Übertragen auf das Board oder nach Drücken des Reset-Tasters) einmalig aufgerufen, um z. B. Pins als Eingang oder Ausgang zu definieren.
* loop() – wird durchgehend immer wieder durchlaufen, solange das Arduino-Board eingeschaltet ist.

### Raspbian

Raspbian ist ein kostenloses Betriebssystem, das auf Debian basiert und für die Raspberry Pi-Hardware optimiert ist. Ein Betriebssystem ist die Menge der grundlegenden Programme und Dienstprogramme, die den Raspberry Pi laufen lassen. Raspbian bietet jedoch mehr als ein reines Betriebssystem: Es enthält mehr als 35.000 Pakete, vorkompilierte Software in einem schönen Format für die einfache Installation auf dem Raspberry Pi. Dieses Betriebssystem basiert auf einem Debian-9-System(Debian Stretch) der ARM-hard-float-Architektur (armhf) mit Anpassungen an den Befehlssatz für den ARMv6-Prozessor. Als grafische Oberfläche wird LXDE vorkonfiguriert. Das etwa 3 GB große Image kann auf SD-Karten mit 4 GB oder mehr übertragen werden. Nach dem Bootvorgang kann die Größe der Raspbian-Partition auf die gesamte SD-Karte erweitert werden. Die Raspberry Pi Foundation erstellt auf Basis der Raspbian-Distribution ein eigenes Raspbian-Image mit passender Firmware für die Raspberry-Pi-Modelle, es wird daher empfohlen, die Distribution immer von der Raspberry Pi Foundation zu beziehen.

### Python

Das Pi im Namen des Raspberry Pi leitet sich von der Programmiersprache Python ab: Die Schöpfer des Minicomputers wollten damit zum Ausdruck bringen, dass diese moderne Skriptsprache besonders zu empfehlen ist.

Python ist eine universelle, üblicherweise interpretierte höhere Programmiersprache. Sie hat den Anspruch, einen gut lesbaren, knappen Programmierstil zu fördern. So werden beispielsweise Blöcke nicht durch geschweifte Klammern, sondern durch Einrückungen strukturiert.

Python unterstützt mehrere Programmierparadigmen, z. B. die objektorientierte, die aspektorientierte und die funktionale Programmierung. Ferner bietet es eine dynamische Typisierung. Wie viele dynamische Sprachen wird Python oft als Skriptsprache genutzt.

Die Sprache hat ein offenes, gemeinschaftsbasiertes Entwicklungsmodell, das durch die gemeinnützige Python Software Foundation, die de facto die Definition der Sprache in der Referenzumsetzung CPython pflegt, gestützt wird.

Wegen ihrer klaren und übersichtlichen Syntax gilt Python als einfach zu erlernen. Sie besitzt eine umfangreiche Standardbibliothek und zahlreiche Pakete im Python Package Index.

Zur Standardinstallation von Python gehört eine integrierte Entwicklungsumge- bung namens IDLE.

Raspbian bringt die Python-Entwicklungsumgebung IDLE bereits vorinstalliert auf dem Desktop mit, sodass bei Bedarf direkt losgelegt werden kann.

IDLE besteht im Wesentlichen aus drei Komponenten:

* **Die Python-Shell:** Beim Starten von IDEL öffnet sich zuerst das Python-Shell- Fenster. Die Shell ist eine Anwendung, mit der Sie direkt mit dem Python-In- terpreter kommunizieren können: Sie können auf der Kommandozeile einzel- ne Python-Anweisungen eingeben und ausführen lassen. Ein Python-Pro- gramm, das eine Bildschirmausgabe liefert, gibt diese in einem Shell-Fenster aus.
* **Der Programmeditor:** Das ist eine Art Textverarbeitungsprogramm zum Schreiben von Programmen. Sie starten den Programmeditor vom Shell-Fens- ter aus (FILE|NEW WINDOW).
* **Der Debugger:** Er dient dazu, den Lauf eines Programms zu kontrollieren, um logische Fehler zu finden.

### Samba

Sabba ist ein freies Programmpaket, das es ermöglicht Windows-Funktionen wie die Datei- und Druckdienste unter anderen Betriebssystemen zu nutzen und die Rolle eines Domain Controllers anzunehmen. Es implementiert hierfür unter anderem das Server Message Block/Common Internet File System-Protokoll (SMB/CIFS-Protokoll).

Da die Software unter der General Public Licence (GPL) kostenfrei verfügbar ist, wird sie als Alternative zu Microsoft-Windows-Server-Betriebssystem eingesetzt.

### Maschinensoftware

# System Life Cycle

Im Rahmen des Projekts Mars Explorer sollen die gewonnenen Erkenntnisse aus den Lehrveranstaltungen Systems Engineering, Embedded Systems und Projektmanagement Anwendung finden sowie eine Fablab Umgebung, mechanische und elektronische Komponentenfertigung und Platinenfertigung kennengelernt werden.

## Systemanforderungen

### Zweck

Das Ziel des Projekts ist, mit den bereitgestellten Komponenten, Materialien, Maschinen und Budget, die Entwicklung eines autonomen Fahrzeuges, welches für extraterrestrische Erkundungsmissionen geeignet ist. Die Eignung wird dabei auf fünf Teststrecken überprüft. Die Anforderungen an das Fahrzeug sind folgende:

* Unbeschadeter Sturz aus einer maximalen Fallhöhe von 0,5 m
* Schutz vor Staub und Wasser
* Abmessungen für den Transport in der Rakete:
  + Länge < 65 cm
  + Breite < 48 cm
  + Höhe < 30 cm
* Gewicht des Fahrzeuges:
  + mRover < 10 kg (Erde)
  + mRover < 4,5 kg (Mars)
* Fortbewegung frei wählbar, jedoch autonom
* Budget: 40 €

### Ziel

Das Projekt soll zu einem autonomen Hybrid-Quadrocopter führen, welcher die unter 3.6 beschriebenen Tracks überwinden kann, indem er zwischen Fahr-und Flugmodus wechselt und seine Umgebung mittels Ultraschallsensor und Kamera erkennt und entsprechend auf unterschiedliche Situationen reagiert. Herzstück des Fahrzeugs ist ein Raspberry Pi sowie ein Arduino, welche Daten empfangen, verarbeiten und Befehle weiterleiten.

Teilziele für das Projekt sind:

* Konstruktion und Fertigung des Chassis
* Drehmechanismus der hinteren Tragflächen konstruieren und fertigen um zwischen Fahr-und Flugmodus zu wechseln
* Platine konstruieren und fertigen
* Programmierung der Motorsteuerung incl. ESC Kalibrierung um die Voraussetzung für das Fliegen zu schaffen
* Autonomer Modus mit Hilfe von Sensoren realisieren (Programmierung für Ultraschallsensor und Kamera)
* Steuerungscode und PIC-Regelung um den Quadrocopter stabil in der Luft zu halten

### Einschränkungen

Das Ein- und Ausschalten erfolgt im manuellen Modus, während dem Fliegen und Fahren agiert die Mohne manuell. Der Abstand zum Boden wird nicht gemessen. Das Überwinden von Hindernissen erfolgt, indem vom Fahrmodus in den Flugmodus gewechselt wird und nach einer vorher eingestellten Zeit wieder in den Fahrmodus übergegangen wird.

Das System muss nur unter guten Wetterbedingungen und auf vorgegebenen Kursen funktionieren, weshalb nur ein geradeausfahren- fliegen realisiert wird.

## Designalternativen

Um über ein Design für dieses Projekt zu entscheiden, mussten bestimmte einschränkende Faktoren berücksichtigt werden, nämlich die maximale Größe und das maximal zulässige Gewicht das das Fahrzeug haben darf, um noch von der Trägerrakete auf den Mars befördert werden zu können. Auch unter der Voraussetzung, dass die elektronischen Komponenten, die irgendwo innerhalb des Rumpfes eingeschlossen und befestigt werden müssen, vor Staub und Wasser geschützt sein müssen, wurden die Möglichkeiten weiter eingeengt. Das vorgegebene Budget und Zeitlimit sowie die zur Verfügung gestellten Materialien und Maschinen limitierten weitere Möglichkeiten. Nachdem einige Überlegungen und Forschung betrieben wurde wie die einzelnen Hürden der Teststrecken am einfachsten bewältigt werden konnten, wurde entschieden, dass ein Hybrid-Quatrocopter entwickelt werden würde. Auch aufgrund der Tatsache, dass sich Drohnen zunehmender Beliebtheit erfreuen und in immer mehr Einsatzbereichen Anwendung finden wollten wir uns dieser Herausforderung stellen und auf die Konstruktion eines konventionellen Mars-Rovers verzichten.

## Model the System

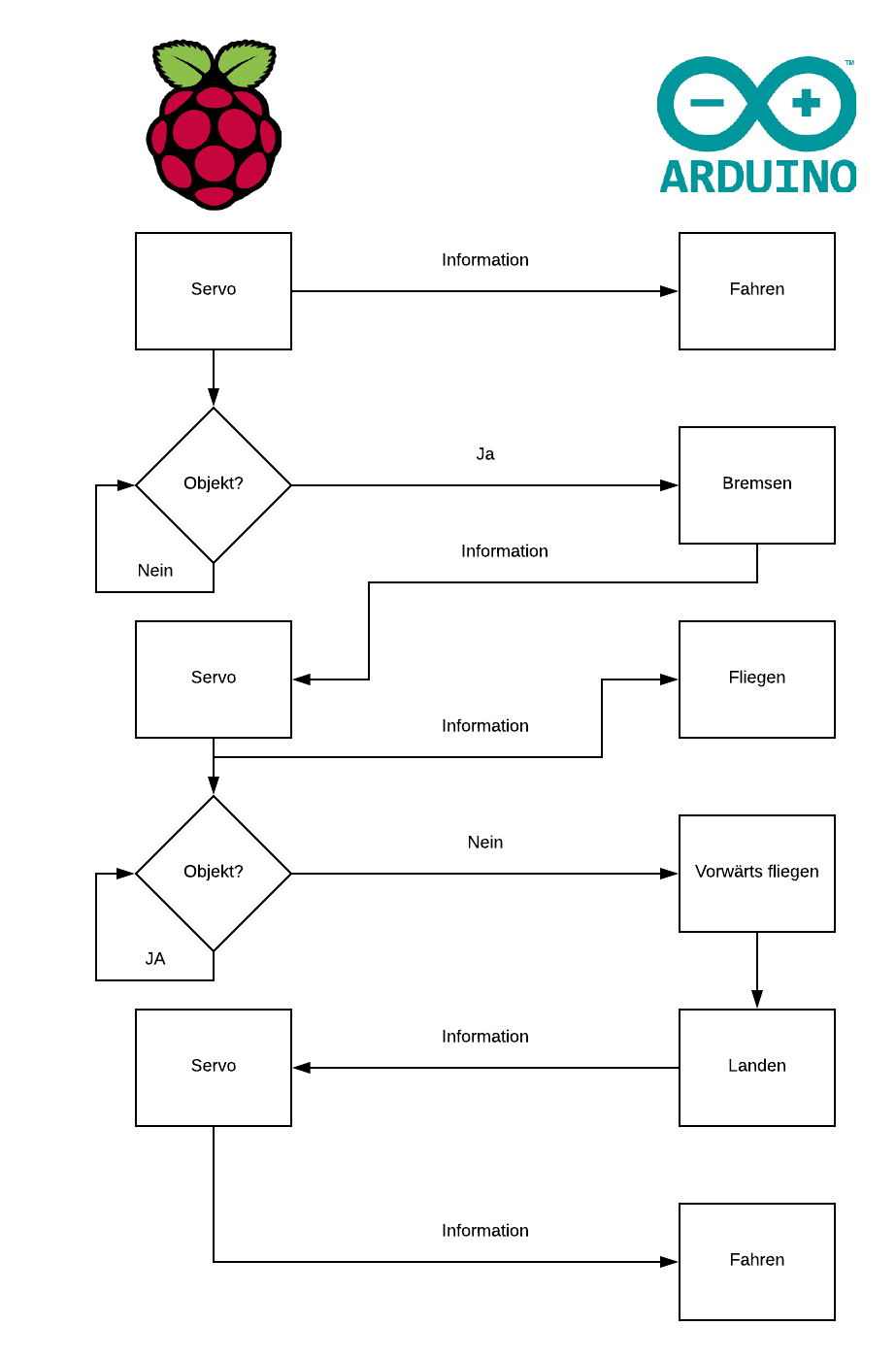
In der Regel werden für nahezu alle alternativen Designs Modelle entwickelt. Das Modell für die bevorzugte Alternative wird dann erweitert und verwendet, um das System über seinen gesamten Lebenszyklus hinweg zu verwalten. Aufgrund der kurzen Entwicklungszeit wurden keine Modelle für alternative Designs entwickelt. In erster Linie sollen die nachstehenden Modelle auch dazu dienen das System zu beschreiben.

### System and Subsystem

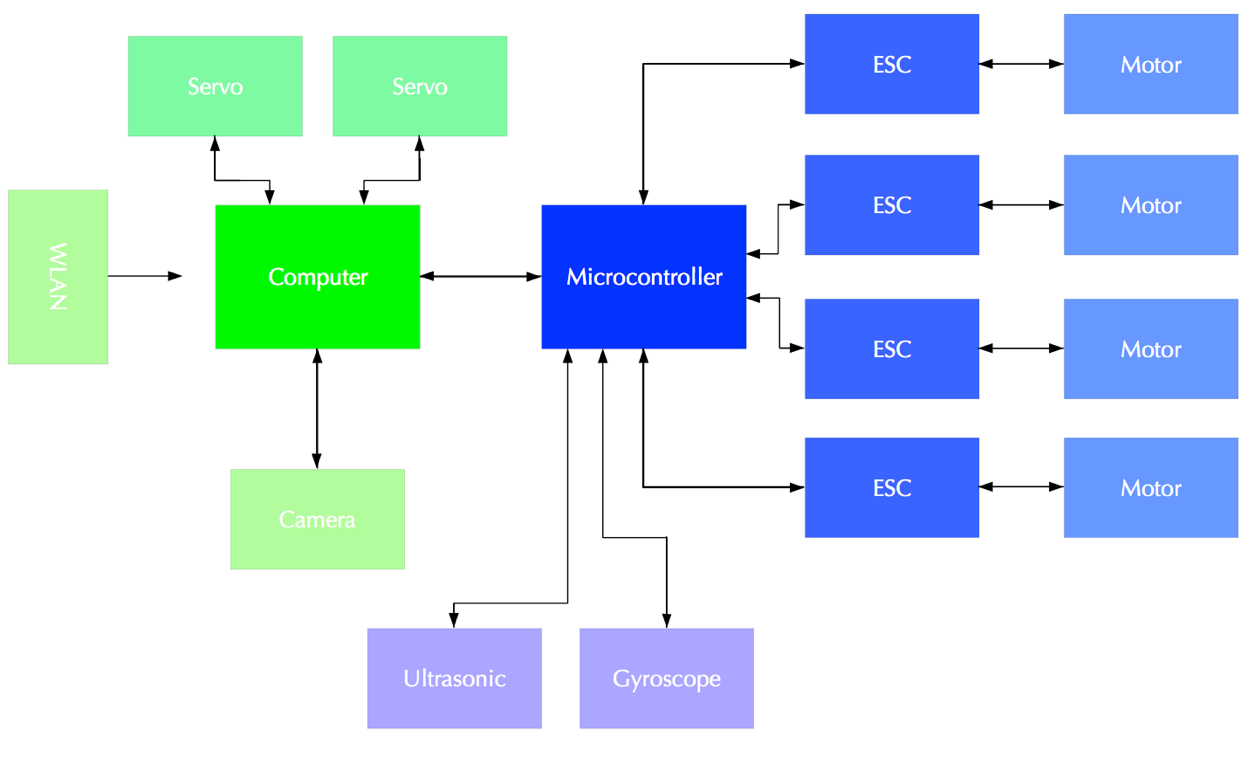
### Make or Bye

|  |  |
| --- | --- |
| Motor | Commercially available subsystem |
| Propeller | Commercially available subsystem |
| ESC | Commercially available subsystem |
| Servo | Commercially available subsystem |
| Sensor | Commercially available subsystem |
| Microcontroller | Commercially available subsystem |
| Computer | Commercially available subsystem |
| Carrier | In-house-design |
| Chassis | In-house-design |
| Rotary mechanism | In-house-design |
| Battery | Commercially available subsystem |
| Wires | Commercially available subsystem |
| Power Distribution Board | Commercially available subsystem |

### Sequenzdiagram



### System Architecture



### Functional Decomposition

|  |  |
| --- | --- |
| **Function** | **Pysical Component** |
| Take-off, drive and land | Wheels |
| Sense objects | Sensors |
| Navigate | Microcomputer |
| Produce horizontal thrust and vertical lift | Propeller and wings |

### Failure Mode and Effects Analysis (FMEA)

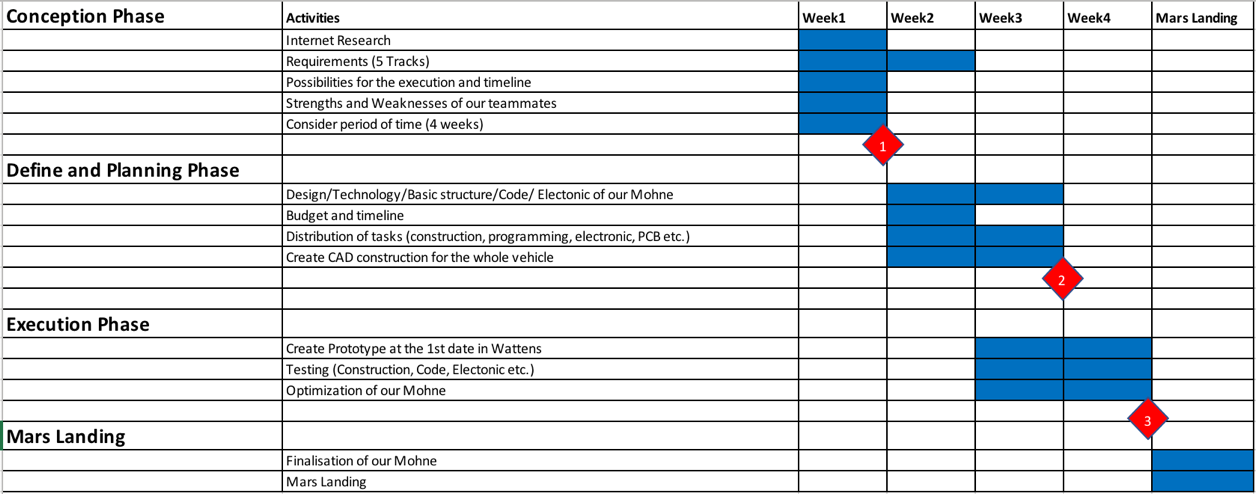
The procedure used to evaluate the various modes of failure in our drone design takes into consideration the severity, probability of occurrence and detectability for each failure mode as well as the root causes. Based on the values assigned to each category, the risk priority and criticality of each cause are calculated. Our failure analysis focused on four failure modes; mobility, navigation, structural and data acquisition. We determined the key failures modes to be mobility and navigation. In each of these two cases, control system and wiring failure are the root causes with the highest risk priorities and criticalities.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Possible Effect** | **Root Cause** | **Potential Indicators** | **Potential Cause(s) of Failure** | **Severity** | **Probability of Occurence** | **Detectability** | **Risk Prirority** | **Criticality** |
| **Mobility** | dead battery | will not power motor | Battery not charged | 5 | 3 | 1 | 15 | 15 |
|  | controls failure | Motors, servos and PID controller failure | Programming error | 5 | 9 | 9 | 405 | 45 |
|  | wiring harness connections | will not start or run correctly | Grounding corrosion | 8 | 8 | 9 | 576 | 64 |
| **Navigation** | Wiring failure | Complete navigation failure | Wiring error | 8 | 7 | 2 | 112 | 56 |
|  | Controls Failure | Complete navigation failure | Programming error | 5 | 9 | 9 | 405 | 45 |
|  | Object sensing hardware failure | Part damage | Wiring control failure | 5 | 3 | 3 | 45 | 15 |
|  | Rotary mechanism failure | Steering failure | Impact on gear | 5 | 5 | 1 | 25 | 25 |
| **Structural** | Frame cracking | No component support | Impact on frame | 10 | 3 | 4 | 120 | 30 |
|  | Engin mount | Engin falls out | Engin mount wear | 2 | 1 | 1 | 2 | 2 |
| **Data Acquisition** | Camera not working | No video recording | Weather proofing controls/wire | 1 | 5 | 2 | 10 | 5 |
|  | Ultrasonic sensor not working | No distance measurement | Weather proofing controls/wire | 1 | 5 | 3 | 15 | 5 |
|  | Incorrect wiring | Will not move | Wiring error | 2 | 2 | 4 | 16 | 4 |
|  | Controls failure | No/incorrect data | Programming error | 1 | 2 | 5 | 10 | 2 |

## Integrate System Components

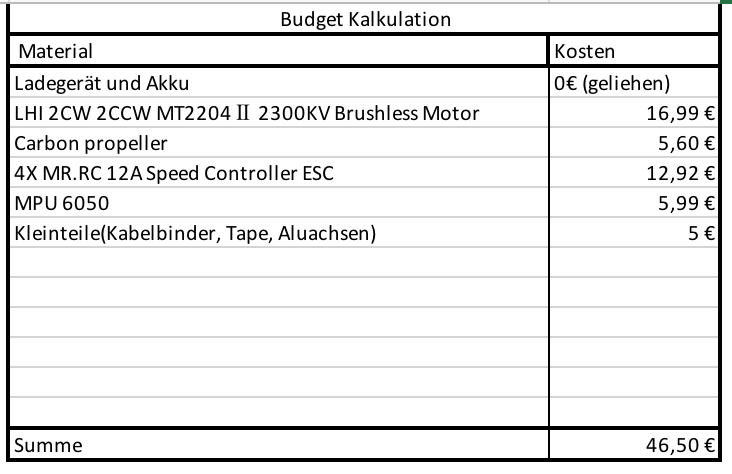
## Launch the System

### Timeline



|  |  |
| --- | --- |
| **Task** | **Team Member(s)** |
| Designentscheidung | alle |
| Chassis konstruieren und fertigen | Duregger, Jungwirth, Gruber |
| Bilderkennung | Paula, Gruber |
| Kamerahalterung | Duregger, Jungwirth |
| Radaufhängung, Achsen und Welle | Duregger, Jungwirth, Gruber |
| Platinelayout | Möbes |
| Elektronik | Möbes, Gruber |
| Servohalterung + Drehmechanismus | Duregger, Jungwirth, Gruber |
| Motorhalterung | Duregger, Jungwirth, Gruber |
| Abstandhalter zwischen Platinen | Duregger, Jungwirth, Gruber |
| ESC Kalibrierung und Motorsteuerung | Möbes, Gruber |
| Steckverbindung HV Akku und Stromplatine | Möbes |
| Steuerungscode | Möbes, Paula, Gruber |
| Projektmanagement und Dokumentation | Wiesmüller |
| Zusammenbau | Wiesmüller, Gruber |
| Kostenanalyse | Gruber |
| Komponentenauswahl und Bestellung | alle |
| Flugtest und Systemtest | alle |

### Budget



## Assess Performance

Während der gesamten Entwicklungsphase wurde die Leistung des Systems immer wieder gemessen und getestet. Diese Messungen waren notwendig um zu verifizieren, dass das System seinen Anforderungen entspricht. Vor allem wurde zu Beginn getestet ob der Rotationsmechanismus wie gedacht funktioniert, was die Basis für den Wechsel zwischen Fahr-und Flugmodus darstellte.

Auch wurden immer wieder Tests durchgeführt um die Bilderkennung und Abstandsmessung sicherzustellen um die autonome Navigation sicherzustellen. Ein weiterer Wichtiger Punkt der immer wieder getestet wurde war die Motorsteuerung um zu gewährleisten, dass sich der Quatrocopter auch fortbewegen kann.

Um letztendlich festzustellen ob das Projekt für einen Einsatz auf dem Mars geeignet ist, sich autonom fortbewegen kann und die Vorstellungen und Wünsche des Auftraggebers erfüllt, mussten 5 Teststrecken absolviert werden, welche nachstehend beschrieben werden.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Track | Abmessungen (l x b) | Oberfläche | Umgebungsbedingung | Ziel |
| 1 | 3m x 1m | Beton | Überdachter Raum | Gerade fahren |
| 2 | 3m x 1m | Beton | Staubsimulation mittels Sprühflaschen | Defektfreie Durchfahrt |
| 3 | 3m x 1m | Holzstufen mit unterschiedlichen Höhen (10cm, 20cm, 30cm, 40cm, 50cm, 60cm) | Überdachter Raum | Hindernisse überwinden |
| 4 | 3m x 1m | Gesteinsbrocken mit unterschiedlicher Höhe (10-60 cm) | Überdachter Raum, Gesteinsbrocken farblich und mit Symbol gekennzeichnet | Track durchqueren, Umfahren der Hindernisse |
| 5 | unbekannt | Unbekannt (Schnee, Eis, Sand, etc.), Gesteinsbrocken mit unterschiedlicher Höhe (10-60cm) | Unter freiem Himmel, Gesteinsbrocken farblich und mit Symbol gekennzeichnet | Überleben |

Zudem wurde von den Auftraggebern inspiziert, ob die geforderten Mindestanforderungen an Größe, Gewicht und Stabilität erfüllt wurden.

## Results, Conclusion and Recommendations

### Results

Dieses Projekt dauerte das 4 Wochen und beinhaltete eine Menge Forschung, Tests und Analysen. Zu Beginn des Projekts erstellte das Team erste Projektspezifikationen wie in Abschnitt 3.1 beschrieben, die als Richtlinien für die Mohne dienten. Es wurde eine Überprüfung durchgeführt, um zu sehen, welche Spezifikationen das Projekt erfüllt und welche nicht.

* Die Drohne erfüllte die maximal zulässige Größe und Gewicht
* Der Falltest aus 0,5m Höhe wurde unbeschadet überstanden
* Die Drohne erfüllte die Anforderungen an die Staub- und Wasserfestigkeit
* Die Drohne erfüllte nicht das Budget von 40 € einzuhalten. Das benötigte Budget betrug 46,50 €
* Die Drohne erfüllte die Anforderung des Not-Stopps
* Die Drohne erfüllte die Anforderung seine Umgebung wahrzunehmen
* Der Wechsel zwischen Flug- und Fahrmodus konnte realisiert werden
* Die Drohne konnte geradeaus fahren
* Die Anforderung zu Fliegen wurde nur bedingt erreicht. Obwohl mehrere Flugversuche durchgeführt wurden, konnte nicht umgesetzt werden, dass alle vier Propeller gleichmäßig mit Strom versorgt wurden und die Drohne somit stabil in der Luft stehen konnte. Das Problem wurde erkannt, die dafür benötigte PID-Regelung konnte jedoch aus Zeitgründen nicht umgesetzt werden.
* Aufgrund des vorherigen Problems konnten die Hindernisse nicht überwunden werden

### Conclusion

Die Endergebnisse zeigten, dass nicht alle Spezifikationen und Ziele, die zu Beginn des Projekts festgelegt wurden, erreicht wurden. Obwohl der vollständig autonome Flug nicht erreicht wurde, hat das Team eine bedeutende Entwicklung hin zu einem kostengünstigen, leichten unbemannten Hybrid-Fahrzeug gemacht, das mit der Implementierung eines Kamerasystems und einem Ultraschallsensor überwacht werden kann. Mehrere Designiterationen wurden abgeschlossen, um ein Fahrzeug zu schaffen, das die Projektziele erfüllen sollte und der endgültige Prototyp zeigt das Potential für den Erfolg der Projektziele. Wie in Abschnitt 3.6 beschrieben, wurden mehrere Tests durchgeführt, um sicherzustellen, dass der entworfene Prototyp erfolgreich fliegen und fahren konnte. Wenn mehr Zeit verfügbar wäre, wäre das Team in der Lage gewesen, ein robusteres Kontrollsystem zu entwerfen, das einen stabilen Flug gewährleisten würde. Insgesamt wurde in diesem Projekt ein unbemanntes Fluggerät vollständig realisiert. Der Hybrid-Quadrocopter zeigte, dass er in der Lage war zu fahren und zu fliegen, aber ihm fehlte die Stabilität. Weitere Iterationen dieses Projekts würden seinen zukünftigen Erfolg sichern.

### Recommendations

Im Laufe des Projekts wurden mehrere Lektionen gelernt und das Team hat mehrere Empfehlungen. Aus Sicht der Steuerung wären mehr Sensoren hilfreich für mehr Präzision oder Fähigkeiten. Ein weiterer Ultraschallsensor könnte dazu dienen um die Flughöhe zu bestimmen. Die Implementierung weitere Kameramodule könnten eine 360 Grad Wahrnehmung realisieren um Hindernisse zu vermeiden. Auch könnte im nächsten Schritt realisiert werden Kurven zu fahren und zu fliegen. Die Durchführung von Simulationen und Berechnungen würde zu einem effektiveren Design führen.

# Diagrams (obligate)

## Use Case Diagram (Anwendungsfalldiagramm)

## Structural Model (Strukturmodelle)

Klassendiagramm oder Kompensationsstrukturdiagramme

## Interaction Model (Interaktionsmodell)

Sequenzdiagramm (Ludi)

## Behavior Model (Verhaltensmodell)

Aktivitätsdiagramme oder Zustandsdiagramme oder Blockdiagramme

# Testing Software (optional)

# Version Control (e.g. Gitlab – obligate)

No source-code report; access to team side

# Design of PCB (obligate) (Max)

## Breadboard

## Schematic

## Board

Stromversorgung

(LiPo Akkus)

Servo

Servo

Ultraschallsensor

Kameramodul

Arduino

Motorregler

Motorregler

Motorregler

Motorregler

Motor mit Propeller

Motor mit Propeller

Motor mit Propeller

Motor mit Propeller

Raspberry Pi

WLAN Not-Stopp

# pdb- power distribution board hat einen batteriespannungsregler (bec) regeln von 11,1 auf 12 v für arduino und auf 5 v für pi ohne bec regelung zu jedem esc