Duale Hochschule Baden-Württemberg

Fachhochschule Nordwestschweiz

Université de Haute-Alsace

QuadroMotion



Semesterarbeit

Studenten: Betreuer:

Stephan Alexis Matthias Bachmann

Urech Gabriel

Henzmann Simon

**Änderungsdokument**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Datum** | **Bearbeiter** | **Beschreibung** |
| 0.1 | 22.02.2016 | Henzmann | Entwurf erstellt |
| 0.2 | 23.02.2016 | Henzmann | Einleitung |
| 0.3 | 25.02.2016 | Stephan | Aufbau Inhalt |
| 0.4 |  |  |  |
| 0.5 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 1.0 |  |  |  |

**Basisdokumente**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dokument** | **Beschreibung** | **Autor** | **Herkunft** |
| FHNW/ DHBW/UHA | Projektrichtlinien | Studiengangleitung Trinat | Trinat.net |

**Danksagung**

Das Projektteam bedankt sich bei der IUT Mulhouse, DHBW Lörrach und der FHNW Muttenz für das ermöglichen des Projekts.

Ein grosser Dank geht an Herrn Matthias Bachmann für das Betreuen dieses Projekts.

Stakeholder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Funktion | Kontaktdaten | Verfügbarkeit | Relevanz |
| M.Bachmann | Dozent | matthias.Bachmann@fhnw.ch | Nach Vereinbarung | Hoch |
| G. Urech | Student | gabrielurech¦@gmail.com | MO-FR | Hoch |
| S. Henzmann | Student | simon.henzmanngmail.com | MO-FR | Hoch |
| A.Stephan | Student | asteph61@gmail.com | MO-FR | Hoch |

**Plagiat**

Inhaltsverzeichnis

[1 Einleitung 5](#_Toc451200945)

[1.1 Projektbeschreibung 5](#_Toc451200946)

[1.2 Vision 5](#_Toc451200947)

[1.3 Danksagung 6](#_Toc451200948)

[1.1 Stakeholder 6](#_Toc451200949)

[2 Projektplanung und Organisation 7](#_Toc451200950)

[2.1 Projektteam 7](#_Toc451200951)

[2.2 Projekt Betreuung 7](#_Toc451200952)

[2.3 Features 7](#_Toc451200953)

[2.4 Risikoanalyse 7](#_Toc451200954)

[2.5 Rahmenbedingungen 7](#_Toc451200955)

[2.7 Terminplan 8](#_Toc451200956)

[2.8 Budget 9](#_Toc451200957)

[2.9 Projektaufteilung 10](#_Toc451200958)

[3 Konzept 11](#_Toc451200959)

[3.1 Systemübersicht 11](#_Toc451200960)

[3.2 Blockdiagramm 12](#_Toc451200961)

[3.3 Zustandsdiagramm (muss überarbeitet werden) 13](#_Toc451200962)

[3.4 Software Architektur – Klassendiagramm 16](#_Toc451200963)

[3.5 Schnittstellen 17](#_Toc451200964)

[3.6 Anforderungen 18](#_Toc451200965)

[3.6.1 Funktionale Anforderungen 18](#_Toc451200966)

[3.6.2 Nicht funktionale Anforderungen 19](#_Toc451200967)

[3.7 Stand der Technik 20](#_Toc451200968)

[3.7.1 Existierende Eingabegeräte 20](#_Toc451200969)

[3.7.2 Die Gesten-Erkennungsmöglichkeiten 20](#_Toc451200970)

[3.7.3 Datenverarbeitung 21](#_Toc451200971)

[3.7.4 Quadrocopter 21](#_Toc451200972)

[3.7.5 Programmiersprache 21](#_Toc451200973)

[3.8 Varianten 22](#_Toc451200974)

[3.8.1 Morphologischer Kasten 22](#_Toc451200975)

[3.9 Nutzwertanalyse 23](#_Toc451200976)

[3.9.1 Kriterien 23](#_Toc451200977)

[4 Umsetzung 26](#_Toc451200978)

[4.1 Programmierung 26](#_Toc451200979)

[4.1.2 Analyse und Test der verschiedenen Umrechnungsmöglichkeit 26](#_Toc451200980)

[4.1.3 Analyse und Test der verschiedenen Gesten Konfigurationen 28](#_Toc451200981)

[6 Testberichte 29](#_Toc451200982)

[8 Fazit 36](#_Toc451200983)

[8.1 Testberichte I 36](#_Toc451200984)

[8.2 Testberichte II 37](#_Toc451200985)

[8.2.1 Testbericht III 38](#_Toc451200986)

[8.2.2 Testbericht IV 39](#_Toc451200987)

[9 Anhang 40](#_Toc451200988)

[9.1 Abbildungsverzeichnis 40](#_Toc451200989)

[9.2 Tabellenverzeichnis 40](#_Toc451200990)

[9.3 Abkürzungen 40](#_Toc451200991)

[9.4 Quellenverzeichnis 40](#_Toc451200992)

# Einleitung

In diesem Dokument beschreibt das Projektteam die Planung, die Entwicklung, die Programmierung und die Herstellung unserem Mechatronischen Projekt. Wir erhielten die Möglichkeit, ein Projekt von A bis Z zu planen und zu realisieren. Dieses Projekt wird im sechsten Semester des Studienganges Mechatronik Trinational erarbeitet.

## Projektbeschreibung

Quadrokopter sind beliebte Flugobjekte und verhältnismäßig einfach zu steuern. Die Steuerung erfolgt einerseits durch eine Fernsteuerung mit Steuerknüppel, was mit mehr oder weniger grossem Aufwand zu erlernen ist, andererseits kann ein Quadrokopter auch mit dem Smartphone gesteuert werden.

LeapMotion ist ein Gerät, das für die Bedienung eines Computers mittels Handbewegungen konzipiert wurde. Es erkennt eine oder zwei Hände, mit denen dann durch Gesten in der Luft der Computer bedient wird.

Die erste Idee dieser Projektarbeit war die Kombination dieser beiden Geräte. Dies bedeutet, dass schlussendlich mit der Hand über das LeapMotion der Quadrokopter gesteuert werden kann. Diese Steuerung soll so intuitiv sein, dass selbst ein unerfahrener Pilot das Steuer übernehmen kann. Im Rahmen dieser Arbeit werden weitere Eingabegeräte evaluiert, die eine Steuerung eines Quadrokopters mittels Gestenbewegung ermöglichen. Ziel des Projektes ist es, eine Gestenorientierte Steuerung für einen Quadrokopter zu realisieren.

## Vision

Die Vision von unserem Projekt QuadroMotion ist es, dem Piloten ein außergewöhnliches Flugerlebnis mit einer Drohne zu ermöglichen. Bisher kann man Drohnen mittels Ipod, Tablet, Smartphone oder anderen Geräten ansteuern. Der Bediener hat dabei jedoch ein «Objekt in den Händen», welches sein Gefühl zu der Drohne beeinflussen kann. Unsere Idee ist, die Drohnen einzig mit den Hand- und Armbewegungen zu steuern, ohne «ein Objekt in den Händen» zu halten, so dass sich der Bediener näher an der Technik fühlt. Des Weiteren soll die Steuerung intuitiv, einfach und schnell erlernbar sein.

## Projektplanung und Organisation

In diesem Kapitel wird der ganze Aufbau, die Organisation und die genaue Planung des Projekts aufgezeigt.

## Projektteam

Das Projektteam besteht aus drei Studenten die alle Mechatronik Trinational studieren und sich im sechsten Semester befinden. Die Studenten sind Gabriel Urech (CH), Alexis Stephan (F) und Simon Henzmann (CH).

## Projekt Betreuung

Das Projekt QuadroMotion wird vom Dozenten Herr Matthias Bachmann betreut.

## Features

|  |  |
| --- | --- |
| **Beschreibung** | **Erreicht** |
| Die Drohne muss steuerbar sein, ohne ein Gerät in den Händen. | Ja |
| QuadroMotion muss im Indoor Bereich einwandfrei funktionieren. | Ja |
| Die Verbindung zur Drohne muss kabellos sein. | Ja |
| Die Bedienung sollte möglichst einfach sein. |  |
| QuadroMotion sollte als Plug-and-Play System funktionieren. |  |
|  |  |
|  |  |
|  |  |

Tabelle 1 Features

## Risikoanalyse

Die Risikoanalyse zeigt mögliche Risiken und Probleme auf, die im Laufe dieses Projekts auftreten könnten.

|  |  |  |
| --- | --- | --- |
| **Risiko** | **Beschreibung** | **Beschreibung** |
| 1 | Die Gesten werden vom Eingabegerät nicht bzw. nicht richtig erkannt |  |
| 2 | Die Drohne fliegt nicht stabil |  |
| 3 | Die Drohne kann nicht genau genug gesteuert werden |  |
| 4 | Die Sicherheit kann in stressigen Situationen nicht gewährleistet werden |  |
| 5 | Das Budget wird überschritten |  |
| 6 | Die Lieferzeit kann nicht eingehalten werden |  |
| 7 | Mechanische defekte der Drohne können das Projekt verzögern |  |

## Rahmenbedingungen

Das Projekt hat einige Rahmenbedingungen an die sich das Projektteam halten muss. Der Abgabetermin des Projekts und der Dokumentation wurde auf den 03. Juni 2016 vereinbart. An diesem Datum ist auch die Präsentation. Es wird ein Video verlangt, das auch am diesem Datum abgegeben werden muss.

## Terminplan

Gantt dia.

## Budget

Beim Budget hat das Projekt zwei verschiedene Versionen. Dies lässt sich dadurch erklären, dass das Projektteam bei der Lösungssuche entdeckt hat, dass die Vorgängerpromotion auch eine Ar Drone 2.0 verwendet hat. Das Projektteam hat bei der Studiengangleitung und beim Laborverantwortlichen angefragt, ob es möglich wäre diese Drohne wieder zu verwenden. Die Antwort auf diese Frage wurde durch beide Parteien positiv beantwortet. Somit kann das alte Budget von 415.8 CHF auf 113.3 CHF reduziert werden. Im aktuellen Budget enthält noch einen kleinen Mehraufwand von 25 Euro für einen neuen Akku.



Tabelle 2 Budget Version 1.0



Tabelle 3 Ursprüngliche Budget Version 0.1

## Projektaufteilung

Folgende Teilaufgaben können unterteilt werden:

* Recherche Stand der Technik

Der Stand der Technik wird recherchiert, um einen guten Überblick zu erhalten, was bereits umgesetzt wurde

Erwartetes Resultat: Stand der Technik ist dokumentiert

* Anforderungen definieren
* Evaluation Drohne/Eingabegerät/Gesten Variantensuche  
  Verschiedene Lösungsvarianten werden gesucht und dokumentiert. Inputs liefert die Recherche des Stands der Technik.

Erwartetes Resultat: Verschiedene Varianten zur Zielführung werden aufgezeigt

* Lösungsfindung  
  Aus den verschiedenen Varianten wird eine geeignete Lösung ausgearbeitet.

Erwartetes Resultat: Eine geeignete und umsetzbare Lösung wird gefunden.

* Programmierung des Eingabegeräts

Das Eingabegerät sendet gewisse Daten, die es aus den erfassten Gesten generiert. Diese gilt es einzulesen und zu interpretieren, um sie dann weiterverarbeiten zu können.  
Erwartetes Resultat: Die Daten des Eingabegeräts können ausgewertet werden

* Programmierung der Ansteuerung der Drohne

Die Drohne benötigt bestimmte Befehle, um eine Bewegung auszuführen. Diese gilt es zu programmieren. Erwartetes Resultat: Die Drohne kann über den Sender gesteuert werden

# Konzept

## Systemübersicht

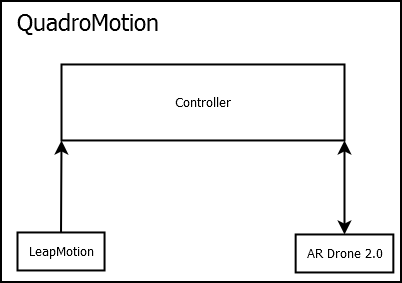


Abbildung 1 Systemübersicht

## Blockdiagramm

Abbildung 2 Blockschaltdiagramm QuadroMotion

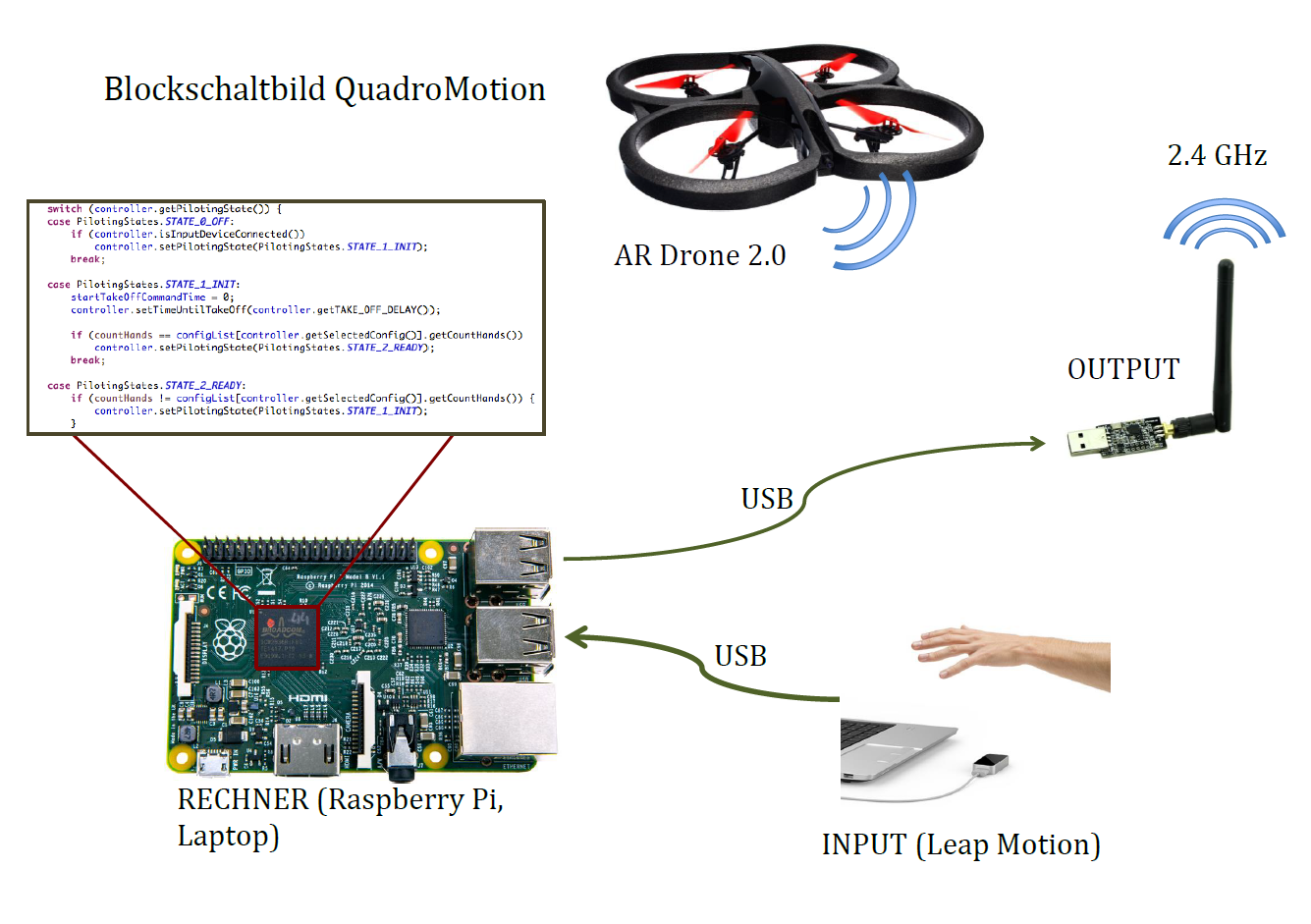
(Quelle: AR Drone <http://drohnenvergleich.ch/drohnenmarken/parrot/parrot-ar-drone-2-0/>

Raspberry Pi <https://de.wikipedia.org/wiki/Raspberry_Pi>

LeapMotion <http://www.robotshop.com/blog/en/explore-virtual-reality-with-leap-motion-3d-motion-controller-16806>

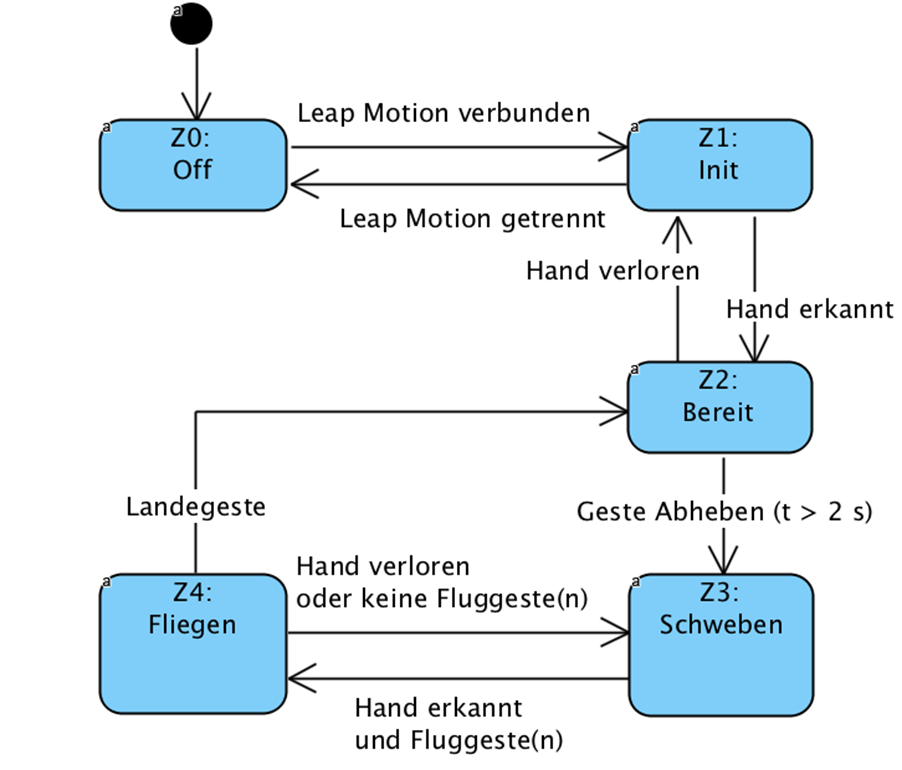
Crasyradio <http://www.seeedstudio.com/depot/Crazyradio-24Ghz-nRF24LU1-USB-radio-dongle-with-antenna-BCCR01A-p-1525.html>

Besucht am 16.05.2016)



## Zustandsdiagramm (muss überarbeitet werden)

Abbildung 3 Zustandsdiagramm



**Z0: Off**

Das Leap Motion ist nicht mit dem Computer verbunden. Es kann nichts gemacht werden.

**Z1: Init**

Das LeapMotion ist mit dem Computer verbunden, es wurde aber keine Hand erkannt. Die Drohne kann noch nicht geflogen werden.

**Z2: Bereit**

Es wurde die richtige Hand bzw. beide Hände erkannt. Dies kann vor und nach einem Flug sein. Die Drohne kann nun gestartet werden.

**Z3: Schweben**

*Nach Z2*: Die Geste „Abheben“ wurde länger als 2 Sekunden erkannt. Der Abhebe-Befehl wurde an die Drohne gesendet.

*Nach Z4:* Die Hand wurde bzw. die Hände wurden nicht erkannt.

Die Drohne schwebt an Ort und Stelle.

**Z4: Fliegen**

Mit der Hand bzw. mit beiden Händen werden Flugsteuerungsgesten ausgeführt, die Drohne fliegt dementsprechend.

Noch ein Diagramm

## Software Architektur – Klassendiagramm

## Schnittstellen

## Anforderungen

### Funktionale Anforderungen

Alle Anforderungen wurden aus der Sicht der Drohne erläutert:

|  |  |  |
| --- | --- | --- |
| **FR-ID** | **Anforderung** | **Akzeptanzkriterium** |
| FR-01 | Alle vorgegebenen Gesten müssen vom Steuerungsprogramm erkannt werden | Alle Gesten werden vom System erkannt |
| FR-02 | Das System muss innerhalb von 500 ms auf Gesten reagieren | Das System reagiert innerhalb von 500 ms |
| FR-03 | Das System darf keine Geste falsch interpretieren | Das System interpretiert jede Geste richtig |
| FR-04 | Die Verbindung zur Drohne soll innerhalb von 10 s hergestellt sein | Die Verbindung ist nach 10 s abgeschlossen |
| FR-05 | Der Status „Drohne verbunden“ muss korrekt visualisiert werden | Der Status „Drohne verbunden“ wird korrekt visualisiert |
| FR-06 | Das System darf den Befehl „Drohne starten“ nicht falsch interpretieren | Das System interpretiert den Befehl „Drohne starten“ nicht falsch |
| FR-07 | Beim Ausführen der Geste „Drohne starten“ muss die Drohne abheben und 0.8 m über dem Boden an Ort schweben | Die Drohne hebt beim Ausführen der Geste „Drohne starten“ ab und schwebt 1 m über dem Boden. |
| FR-08 | Durch die Geste „Drohne vorwärts bewegen“ muss sich die Drohne vorwärts bewegen | Die Drohne bewegt sich vorwärts beim Ausführen der Geste „Drohne vorwärts bewegen“ |
| FR-09 | Durch die Geste „Drohne rückwärts bewegen“ muss sich die Drohne rückwärts bewegen | Die Drohne bewegt sich rückwärts beim Ausführen der Geste „Drohne rückwärts bewegen“ |
| FR-10 | Durch die Geste „Drohne nach links bewegen“ muss sich die Drohne nach links bewegen | Die Drohne bewegt sich nach links beim Ausführen der Geste „Drohne nach links bewegen“ |
| FR-11 | Durch die Geste „Drohne nach rechts bewegen“ muss sich die Drohne nach rechts bewegen | Die Drohne bewegt sich nach rechts beim Ausführen der Geste „Drohne nach rechts bewegen“ |
| FR-12 | Durch die Geste „Drohne nach oben bewegen“ muss sich die Drohne nach oben bewegen | Die Drohne bewegt sich nach oben beim Ausführen der Geste „Drohne nach oben bewegen“ |
| FR-13 | Durch die Geste „Drohne nach unten bewegen“ muss sich die Drohne nach unten bewegen | Die Drohne bewegt sich nach unten beim Ausführen der Geste „Drohne nach unten bewegen“ |
| FR-14 | Durch die Geste „Drohne landen“ muss die Drohne sicher gelandet werden | Die Drohne landet beim Ausführen der Geste „Drohne landen“ |
| FR-15 | Ist die Drohne gestartet und wird keine Geste erkannt, muss die Drohne an Ort und Stelle schweben | Die Drohne schwebt an Ort und Stelle wenn sie gestartet ist und keine Geste erkannt wird |
| FR-16 | Die Drohne muss sich im Schwebeflug selber stabilisieren | Das Flugobjekt darf während der Schwebeflugphase nicht ausser Kontrolle geraten und abstützen |
| Fr-17 | Die Verbindung zur Drohne sollte mit jedem rechnerähnlichen Gerät möglich sein | Die Verbindung der Drohne und des Controllers muss kompatibel sein. So kann ein Z.B ein Rechner oder ein Rasperry Pi die Drohne steuern. |

### Nicht funktionale Anforderungen

|  |  |  |
| --- | --- | --- |
| **NFR-ID** | **Anforderung** | **Akzeptanzkriterium** |
| NFR-01 | Die Sicherheit muss zu jedem Zeitpunkt gewährleistet sein | Die Sicherheit ist zu jedem Zeitpunkt gewährleistet |
| NFR-02 | Die Drohne darf keine zufälligen Bewegungen machen | Die Drohne macht keine zufällige Bewegungen |
| NFR-03 | Die Bedienung soll intuitiv und einfach sein | Ungeschulte Personen können die Drohne ohne Training problemlos fliegen |
| NFR-04 | Alle Komponenten zusammen dürfen den Betrag von 500 CHF nicht überschreiten | Das Budget ist genau definiert und kann nur durch die Bewilligung der Studiengangs Leitung angepasst werden |
| NFR-05 | Der mechanische Aufbau muss mit einem Zeitaufwand von 2 Arbeitstagen realisierbar sein | Der Aufbau der Komponenten darf nicht zu komplex sein. |
| NFR-06 | Die Lieferzeit von den Komponenten muss innerhalb von 3 Wochen liegen | Um die Termin Planung nicht zu gefährden sollte die Lieferzeit nicht überschritten werden |
| NFR-07 | Die Software sollte Objektorientiert geschrieben sein | Der Aufbau ist übersichtlich gestaltet. |

## Stand der Technik

Bei der Analyse der Stand der Technik realisierte das Projektteam das schon einige ähnliche Projekte, von anderen Studenten und Technikern, gibt. Auf diese Weise konnte das Projektteam einige Projekte analysieren und ein eigenes Projekt gestalten.

### Existierende Eingabegeräte

Heutzutage gibt es schon beliebig viele Möglichkeiten eine Drohne zu steuern. Nach einige Recherche hat sich rausgegeben, dass die am meist verwendete Steuerungen noch die Klassische 2.4GHz Steuerung sind, die selbe die im Model bau im Einsatz sind. Der Hersteller von der AR Drone hat eine Gesten-Steuerung mit dem Myio Gerät schon auf dem Markt. Es gibt auch immer mehr Smartphone App um Drohne steuern zu können. Da die Drohne momentan so im Trend ist, versuchen die Hersteller die Bedienung von denen immer zu vereinfachen so dass jeder Mensch die Möglichkeit hat sich eine Drohne zu kaufen und diese zu beherrschen ohne großen Aufwand und/oder Bildung.

### Die Gesten-Erkennungsmöglichkeiten

Um unser Projekt zu Starten haben wir verschiedene eingabe-Möglichkeiten gesucht. Nach einige Recherche haben wir uns für zwei vermarktete, und eine selbst zu entwickelnde Möglichkeit.

#### Leap Motion



Abbildung 4 Leap Motion

Quelle: <http://dulce.com/blog/leap-motion-orion-dulce-dotcom/> (16.05.2016)

Leap Motion [[1]](#footnote-1)ist ein Produkt vom Start Up Unternehmen Motion Control. Die Vision von Motion Control ist die Gestensteuerung zu optimieren und zu perfektionieren in der Zukunft. Nach einem Jahr Entwicklungszeit wurde das Gerät Leap Motion entwickelt. Leap Motion ist ein sehr kleines Eingabegerät das via USB-Kabel mit dem Rechner verbunden ist. Dieses Gerät kann die Handbewegungen im Raum präzise registrieren. Wie auf dem Bild sehr gut zu erkennen ist, simuliert Leap Motion die Hände auf dem Rechner. Die Verbindung mit dem Rechner ist perfekt angepasst.



Abbildung 5 Myo

Quelle:

<https://drjpau.wordpress.com/2015/02/09/myo-by-thalmic-labs-review/> (16.05.2016)

#### Myo

Das Myo[[2]](#footnote-2) ist ein Gadget-Armband das mit seiner Hilfe mit simplen Armbewegungen eine Präsentation, Musik, Videos und Programme steuern werden kann. Das Gadget wird auf der Oberseite des Oberarms platziert, dort registriert es die Muskelbewegungen der Hand und des Arms. Die Synchronisation ist relative einfach. Die Anwendung dagegen braucht viel Übung.

#### Handschuh

Eine weitere Idee ist einen Handschuh selber zu bauen, der mittels Dehnungsmessstreifen die Winkel des Handgelenkes erkennt. Dies wäre Marke Eigenbau.

### Quadrocopter

Als Flugobjekt haben wir uns für einen Quadrocopter entschieden. Aus dem Grund, weil ein solcher einfach zu steuern ist und an Ort und Stelle schweben kann. Als Quadrocopter kommen für uns zwei Modelle in Frage: Die AR Drone 2.0 von Parrot und die Crazyflie 2.0. Zu beiden gibt es Bibliotheken

#### AR Drone 2.0

Die AR Drone 2.0 [[3]](#footnote-3)wird von dem französischen Unternehmen Parrot SA hergestellt. Diese Drohne kann schon via Apps gesteuert werden. Die Drohne besitzt mehrere Sensoren und Trägheitsnavigationseinheit die ihr erlauben zu nicken, rollen und gieren. Andere Trägheitsmesser sind für die automatische Stabilisation der Manöver sowie assistierte Neigungskontrolle zuständig. Hiermit werden realistische AR-Effekte erzeugt. Die Ultraschalltelemetrie ermöglicht Höhenmessung und automatische Höhenstabilisierung sowie assistierte Geschwindigkeitskontrolle

#### Crazyflie 2.0

Die Craszyflie 2.0 [[4]](#footnote-4)ist eine Hand grosse, 27 Gramm schwere Drohne die von dem Unternehmen Bitcraze hergestellt wird. Die Drohne ist ziemlich einfach mit einem mobilen Gerät zu fliegen und ist hauptsächlich für den Indoor Bereich gedacht.

### Programmiersprache

Zu der Entwicklung von unserer Software können verschiedene Programmiersprachen verwendet werden. Kriterien sowie Hardware, Knowhow, informations-quellen, etc… Müssen wir für unsere Entscheidung beachten.

#### Java

#### Python

### Datenverarbeitung

## Varianten

Das Projektteam hat einige einzelne Komponenten betrachtet und analysiert. Da es viele mögliche Varianten gibt, hat sich das Projektteam entschieden ein paar Tools anzuwenden um eine optimale Variante zu evaluieren.

### Morphologischer Kasten

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Ausprägung 1** | **Ausprägung 2** | **Ausprägung 3** | **Ausprägung 4** |
| **Drohne** | Crazyflie 2.0 | AR Drone 2.0 |  |  |
| **Steuerung** | Leap Motion | Myo | Handschuh | Tablet |
| **Rechner** | Raspberry Pi 2.0 | Banana Pi | Laptop | Ardunio Mega |
| **Übertragung** | Crazyradio | WLAN | Bluetooth |  |
| **Energieversorgung** | Powerbank | Netz | Batterie |  |
| **Programm Sprache** | JAVA | C/C++ | Python | C- Sharp |
| **Programmierumgebung** | Visual studio | Eclipse Mars 2.0 | Arduino |  |

Beim Morphologischer Kasten haben wir verschiede Möglichkeiten angeschaut und die 3 besten aufgezeichnet.

## Nutzwertanalyse

Mit der Nutzwertanalyse haben wir die drei verschiedenen Varianten, die wir anhand des Morphologischen Kasten erhalten haben, miteinander verglichen. Wir haben unsere Anforderungen in verschiedenen Klassifikationen eingeteilt.

(M): Stehen für die Anforderungen die unbedingt erreicht werden müssen

(R): Stehen für die Anforderungen die unbedingt erreicht werden müssen haben aber einen gewissen Toleranzbereich

(W): Stehen für die Anforderungen, die nicht unbedingt vorhanden sein müssen aber trotzdem nice to have sind.

(O): Stehen für die Anforderungen, die nicht unbedingt vorhanden sein müssen aber trotzdem nice to have sind mit einem gewissen Toleranzbereich.

### Kriterien

1. Der Preis darf 500 CHF nicht überschreiten. (R)
   1. Der Preis für das Eingabegerät darf nicht höher als
   2. Der Preis für den Quadrocopter darf nicht höher als 200 CHF
   3. Preis Rechner
2. Die Gesten müssen zu 95 % erkannt werden. (R)
3. Die Lieferzeit sollte nicht länger als 3 Wochen betragen. (O)
4. Die Einarbeitungszeit sollte nicht mehr als 45 Stunden sein. Davon 20 Stunden Gestenerkennung, 15 Stunden Drohnenansteuerung und 10 Stunden für die Programmiersprache. (O)
5. Der Zeitaufwand für den Zusammenbau der mechanischen Elemente sollte nicht mehr als 12 Stunden betragen. Davon sind 8 Stunden für das Eingabegerät und 4 für das Flugobjekt. (O)
6. Das Flugobjekt muss mindestens 3 Sekunden ohne menschlichen Eingriff am gleichen Ort schweben können (R)
7. Das Flugobjekt und das Eingabegerät müssen Bibliotheken zur Verfügung stellen (M)
8. Die Kommunity sollte mehr als 10 Antworten auf eine Frage bereitstellen. (O)

#### Gewichtung

|  |  |  |  |
| --- | --- | --- | --- |
| **NR** | **Kurzbezeichnung** | **Art Kriterium** | **Gewichtung** |
| Kriterium 1a | Preis Eingabegerät | R | 9 |
| Kriterium 1b | Preis Quadrocopter | R | 7 |
| Kriterium 2 | Zuverlässigkeit Gestenerkennung | R | 23 |
| Kriterium 3a | Lieferzeit Eingabegerät | O | 6 |
| Kriterium 3b | Lieferzeit Quadrocopter | O | 6 |
| Kriterium 4a | Einarbeitungszeit Gestenerkennung | O | 10 |
| Kriterium 4b | Einarbeitungszeit Ansteuerung | O | 7 |
| Kriterium 4c | Einarbeitungszeit Programmierung | O | 5 |
| Kriterium 5a | Zusammenbau Eingabegerät | O | 2 |
| Kriterium 5b | Zusammenbau Quadrocopter | O | 3 |
| Kriterium 6 | Flugstabilität | R | 20 |
| Kriterium 7 | Verfügbarkeit Bibliothek | M | - |
| Kriterium 8 | Kommunity | O | 2 |
| **Summe** |  |  | **100** |

#### Bestimmung des Nutzwertes Beschreibung



Tabelle 4 Nutzwertanalyse

Bei der Nutzwertanalyse hat sich rausgestellt, dass die Variante 1 mit 806 Punkten am besten auf unsere Anforderungen passt. Mit dieser Auswertung hat sich das Projektteam für diese Lösung entschieden. Hinsichtlich der unsere Analyse hat sich etwas Interessantes ergeben. Die Variante drei, mit dem Handschuh, hat unerwartet den Platz zwei erreicht

# Umsetzung

## Programmierung

Die ganze Software würde mit eclipse Entwickelt und in Java Sprache geschrieben. Um die Arbeit im Team zu vereinfachen würde Git verwendet für die Verteilung der arbeit.

#### Leap Motion

Zur Programmierung Stand uns den Offizielle SDK von LeapMotion zu Verfügung. Das hat uns geholfen da es schon vorprogrammierte Klassen zur Erkennung der Gesten und der Bewegungen von den Armen und Händen zu verfügung gestellt hat. Die Inbetriebnahme dieses SDK hat am Anfang ziemlich viel Zeit gekostet, weil wir es mit verschiedene Entwicklungs-umgebungen und Programmiersprachen ausprobiert haben (Visual Studio mit Python und eclipse für JAVA). Parallel zu der Inbetriebnahme des LeapMotion-SDK, haben wir uns mit der Inbetriebnahme des AR-Drone SDK beschäftigt. Die ersten Tests zur Gesten-Erkennung haben mittels ein Programm mit Konsole-Ausgabe stattgefunden. Somit könnten wir schon eine Idee haben welche Gesten Brauchbar sein könnten für die Umsetzung. Anhand von diesen Informationen könnten wir die Ersten Gesten-Kombinationen bilden die wir später programmieren wollen.

#### AR Drone 2.0

### Analyse und Test der verschiedenen Umrechnungsmöglichkeit

**Die Formel:**

X = Variable

A = Angle Offset

B = Max Angle

C = Speed Offset

D = Max Speed

### Analyse und Test der verschiedenen Gesten Konfigurationen

# Testberichte

Test Nummer: 20160513

Verhältnisse: Kein Wind, künstliches Licht (Indoor)

Testaufbau: PC mit Win 7, AR Drone 2.0, LeapMotion, Program Version 7ec91f9

Tester: Alexis Stephan, Gabriel Urech, Simon Henzmann, Jens Gais, Patrick Grubert, Jost v. Segesser, Kyle Correal,

|  |  |
| --- | --- |
| **TF1** | |
| Anforderung | Alle vorgegebenen Gesten müssen vom Steuerungsprogramm erkannt werden |
| Req-ID | FR-01 |
| Akzeptanzkriterium | Alle Gesten werden vom System erkannt |
| Tester | Die Gesten wurden von allen Tester getestet |
| Testablauf | Drohne verbunden mit dem Rechner |
| Beschreibung | Folgende Gesten wurden auf ihre funktionsweise getestet: Starten, Landen, Vorwärts, Rückwärts, Links, Rechts, ClockWise drehen, Counter ClockWise drehen |
| Status | Alle Gesten wurden erfolgreich erkannt |

|  |  |
| --- | --- |
| **TF2** | |
| Anforderung | Das System muss innerhalb von 500 ms auf Gesten reagieren |
| Req-ID | FR-02 |
| Akzeptanzkriterium | Das System reagiert innerhalb von 500 ms |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **TF3** | |
| Anforderung | Das System darf keine Geste falsch interpretieren |
| Req-ID | FR-03 |
| Akzeptanzkriterium | Das System interpretiert jede Geste richtig |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **TF4** | |
| Anforderung | Die Verbindung zur Drohne soll innerhalb von 10 s hergestellt sein |
| Req-ID | FR-04 |
| Akzeptanzkriterium | Die Verbindung ist nach 10 s abgeschlossen |
| Tester | Stephan, Urech, Henzmann |
| Testablauf | Drohne ist aktiviert und der Rechner ist mit dem W\_LAN der Drohne verbunden. Eclipse gestartet. |
| Beschreibung | Wenn der Button “Drohne verbinden“ gedrückt wird muss die Verbindung innerhalb 10 s erstellt sein. Dies wurde mit einer stopp Uhr gemessen. |
| Status | Wurde erfüllt. |

|  |  |
| --- | --- |
| **TF5** | |
| Anforderung | Der Status „Drohne verbunden“ muss korrekt visualisiert werden |
| Req-ID | FR-05 |
| Akzeptanzkriterium | Der Status „Drohne verbunden“ wird korrekt visualisiert |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **TF6** | |
| Anforderung | Das System darf den Befehl „Drohne starten“ nicht falsch interpretieren |
| Req-ID | FR-06 |
| Akzeptanzkriterium | Das System interpretiert den Befehl „Drohne starten“ nicht falsch |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **TF7** | |
| Anforderung | Beim Ausführen der Geste „Drohne starten“ muss die Drohne abheben und 1 m über dem Boden an Ort schweben |
| Req-ID | FR-07 |
| Akzeptanzkriterium | Die Drohne hebt beim Ausführen der Geste „Drohne starten“ ab und schwebt 1 m über dem Boden. |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **TF8** | |
| Anforderung | Durch die Geste „Drohne vorwärts bewegen“ muss sich die Drohne vorwärts bewegen |
| Req-ID | FR-08 |
| Akzeptanzkriterium | Die Drohne bewegt sich vorwärts beim Ausführen der Geste „Drohne vorwärts bewegen“ |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **TF9** | |
| Anforderung | Durch die Geste „Drohne vorwärts bewegen“ muss sich die Drohne vorwärts bewegen |
| Req-ID | FR-09 |
| Akzeptanzkriterium | Die Drohne bewegt sich rückwärts beim Ausführen der Geste „Drohne rückwärts bewegen“ |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **TF10** | |
| Anforderung | Durch die Geste „Drohne nach links bewegen“ muss sich die Drohne nach links bewegen |
| Req-ID | FR-10 |
| Akzeptanzkriterium | Die Drohne bewegt sich nach links beim Ausführen der Geste „Drohne nach links bewegen“ |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **TF11** | |
| Anforderung | Durch die Geste „Drohne nach rechts bewegen“ muss sich |
| Req-ID | FR-11 |
| Akzeptanzkriterium | Die Drohne bewegt sich nach rechts beim Ausführen der Geste „Drohne nach rechts bewegen“ |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **TF12** | |
| Anforderung | Durch die Geste „Drohne nach oben bewegen“ muss sich die Drohne nach oben bewegen |
| Req-ID | FR-12 |
| Akzeptanzkriterium | Die Drohne bewegt sich nach oben beim Ausführen der Geste „Drohne nach oben bewegen“ |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **TF13** | |
| Anforderung | Durch die Geste „Drohne nach unten bewegen“ muss sich die Drohne nach unten bewegen |
| Req-ID | FR-13 |
| Akzeptanzkriterium | Die Drohne bewegt sich nach unten beim Ausführen der Geste „Drohne nach unten bewegen“ |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **TF14** | |
| Anforderung | Durch die Geste „Drohne landen“ muss die Drohne sicher gelandet werden |
| Req-ID | FR-14 |
| Akzeptanzkriterium | Die Drohne landet beim Ausführen der Geste „Drohne landen“ |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **TF15** | |
| Anforderung | Ist die Drohne gestartet und wird keine Geste erkannt, muss die Drohne an Ort und Stelle schweben |
| Req-ID | FR-15 |
| Akzeptanzkriterium | Die Drohne schwebt an Ort und Stelle wenn sie gestartet ist und keine Geste erkannt wird |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **TF16** | |
| Anforderung | Die Drohne muss sich im Schwebeflug selber stabilisieren |
| Req-ID | FR-16 |
| Akzeptanzkriterium | Das Flugobjekt darf während der Schwebeflugphase nicht außer Kontrolle geraten und abstützen |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **TF17** | |
| Anforderung | Die Verbindung zur Drohne sollte mit jedem rechnerähnlichen Gerät möglich sein |
| Req-ID | FR-17 |
| Akzeptanzkriterium | Die Verbindung der Drohne und des Controllers muss kompatibel sein. So kann ein Z.B ein Rechner oder ein Rasperry Pi die Drohne steuern. |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **NTF-1** | |
| Anforderung | Die Verbindung zur Drohne sollte mit jedem rechnerähnlichen Gerät möglich sein |
| Req-ID | NFR-1 |
| Akzeptanzkriterium | Die Verbindung der Drohne und des Controllers muss kompatibel sein. So kann ein Z.B ein Rechner oder ein Rasperry Pi die Drohne steuern. |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **NTF-2** | |
| Anforderung | Die Drohne darf keine zufälligen Bewegungen machen |
| Req-ID | NFR-2 |
| Akzeptanzkriterium | Die Drohne macht keine zufällige Bewegungen |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **NTF-3** | |
| Anforderung | Die Bedienung soll intuitiv und einfach sein |
| Req-ID | NFR-3 |
| Akzeptanzkriterium | Ungeschulte Personen können die Drohne ohne Training problemlos fliegen |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **NTF-4** | |
| Anforderung | Alle Komponenten zusammen dürfen den Betrag von 500 CHF nicht überschreiten |
| Req-ID | NFR-4 |
| Akzeptanzkriterium | Das Budget ist genau definiert und kann nur durch die Bewilligung der Studiengangs Leitung angepasst werden |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **NTF-5** | |
| Anforderung | Der mechanische Aufbau muss mit einem Zeitaufwand von 2 Arbeitstagen realisierbar sein |
| Req-ID | NFR-5 |
| Akzeptanzkriterium | Der Aufbau der Komponenten darf nicht zu komplex sein. |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **NTF-6** | |
| Anforderung | Die Lieferzeit von den Komponenten muss innerhalb von 3 Wochen liegen |
| Req-ID | NFR-6 |
| Akzeptanzkriterium | Um die Termin Planung nicht zu gefährden sollte die Lieferzeit nicht überschritten werden |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |  |
| --- | --- |
| **NTF-7** | |
| Anforderung | Die Software sollte Objektorientiert geschrieben sein |
| Req-ID | NFR-7 |
| Akzeptanzkriterium | Der Aufbau muss übersichtlich gestaltet werden. |
| Tester |  |
| Testablauf |  |
| Beschreibung |  |
| Status |  |

|  |
| --- |
| Tester: Gabriel Urech, Alexis Stephan, Simon Henzmann  Testdatum: |
| **Testfall** | | **Erfüllt?** | **Besondere Vorkommnisse** |
| TF1  Alle vorgegebenen Gesten müssen vom Steuerungsprogramm erkannt werden | |  |  |
| TF-2¨  Das System muss innerhalb von 500 ms auf Gesten reagieren | |  |  |
| TF-3  Das System darf keine Geste falsch interpretieren | |  |  |
| TF-4  Die Verbindung zur Drohne soll innerhalb von 10 s hergestellt sein | |  |  |
| TF-5  Der Status „Drohne verbunden“ muss korrekt visualisiert werden | |  |  |
| TF-6  Das System darf den Befehl „Drohne starten“ nicht falsch interpretieren | |  |  |
| TF-7  Beim Ausführen der Geste „Drohne starten“ muss die Drohne abheben und 1 m über dem Boden an Ort schweben | |  |  |
| TF-8  Durch die Geste „Drohne vorwärts bewegen“ muss sich die Drohne vorwärts bewegen | |  |  |
| TF 9  Durch die Geste „Drohne vorwärts bewegen“ muss sich die Drohne vorwärts bewegen | |  |  |
| TF 10  Durch die Geste „Drohne nach links bewegen“ muss sich die Drohne nach links bewegen | |  |  |
| TF-11  Durch die Geste „Drohne nach rechts bewegen“ muss sich die Drohne nach rechts bewegen | |  |  |
| TF 12  Durch die Geste „Drohne nach oben bewegen“ muss sich die Drohne nach oben bewegen | |  |  |
| TF 13  Durch die Geste „Drohne nach unten bewegen“ muss sich die Drohne nach unten bewegen | |  |  |
| TF 14  Durch die Geste „Drohne landen“ muss die Drohne sicher gelandet werden | |  |  |
| TF 15  Ist die Drohne gestartet und wird keine Geste erkannt, muss die Drohne an Ort und Stelle schweben | |  |  |
| TF 16  Die Drohne muss sich im Schwebeflug selber stabilisieren | |  |  |
| TF 17  Die Verbindung zur Drohne sollte mit jedem rechnerähnlichen Gerät möglich sein | |  |  |
| NTF-1  Die Sicherheit muss zu jedem Zeitpunkt gewährleistet sein | |  |  |
| NTF-2  Die Drohne darf keine zufälligen Bewegungen machen | |  |  |
| NTF-3.  Die Bedienung soll intuitiv und einfach sein | |  |  |
| NTF-4  Alle Komponenten zusammen dürfen den Betrag von 500 CHF nicht überschreiten | |  |  |
| NTF 5  Der mechanische Aufbau muss mit einem Zeitaufwand von 2 Arbeitstagen realisierbar sein | |  |  |
| NTF 6  Die Lieferzeit von den Komponenten muss innerhalb von 3 Wochen liegen | |  |  |
| NTF 7  Die Software sollte Objektorientiert geschrieben sein | |  |  |

# Fazit

## Testberichte I

|  |  |  |
| --- | --- | --- |
| Test Nr. 20160430 | | |
| Name : Alexis Stephan | Datum : 30.04.2016 | Ort : F-Sessenhiem |

**Test Nr. 20160430**

Lichtverhältnisse: künstliches Licht (neon), kein wind (indoor).

**Testaufbau:**

* PC mit Windows 7 64 Bits
* AR Drone 2.0
* LeapMotion
* Dritte Person als Hilfe im Fall eines Unfalls

**Getestete Funktionen:**

Starten, Landen, Vorwärts, Rückwärts, Links, Rechts, ClockWise drehen, Counter ClockWise drehen

Test Ergebnisse:

|  |  |  |
| --- | --- | --- |
| **Funktionen** | **Ergebnis** | **Bemerkung** |
| Starten | i.o | Startet nicht immer, die Drohne muss oft neugestartet werden |
| Landen | i.o | Wenn die Drohne fliegt wird der Landungs-befehl erkennt und mit eine geringe Verzögerung gesendet |
| Vorwärts | n.i.o | Die Drohne reagiert nicht auf den Befehl. |
| Rückwärts | n.i.o | Die Drohne reagiert nicht auf den Befehl. |
| Links | i.o | Die Drohne reagiert auf den Befehl „Links“. Eine unnormale Verzögerung ist aber zu erkennen. |
| Rechts | n.i.o | Die Drohne reagiert nicht auf den Befehl. |
| Turn ClockWise | i.o | Die Drohne reagiert sofort auf den Befehl. |
| Turn Counter ClockWise | i.o | Die Drohne reagiert sofort auf den Befehl. |

Allgemeine Bemerkung:

* Verbindung ist schwer erreichbar nach Abbruch.
* Take- Off Befehl wird nicht immer verarbeitet/ oder bekommen von der Drohne.
* Neu – Starten war öfters notwendig.

## Testberichte II

|  |  |  |
| --- | --- | --- |
| Test Nr. 20160430 | | |
| Name : Alexis Stephan | Datum : 30.04.2016 | Ort : F-Sessenhiem |
|  |  |  |

Test Nr. 20160430

Lichtverhältnisse: künstliches Licht (neon).

Luftverhältnisse: kein wind (indoor).

**Testaufbau:**

* PC mit Windows 7 64 Bits
* AR Drone 2.0
* LeapMotion
* Programm Version 7ec91f9
* Dritte Person als Hilfe im Fall eines Unfalls

**Getestete Funktione**n:

Starten, Landen, Vorwärts, Rückwärts, Links, Rechts, ClockWise drehen, Counter ClockWise drehen

**Test Ergebnisse:**

|  |  |  |
| --- | --- | --- |
| **Funktionen** | **Ergebnis** | **Bemerkung** |
| Starten | i.o | Startet nicht immer, die Drohne muss oft neugestartet werden |
| Landen | i.o | Wenn die Drohne fliegt wird der Landungs-befehl erkennt und mit eine geringe Verzögerung gesendet |
| Vorwärts | n.i.o | Die Drohne reagiert nicht auf den Befehl. |
| Rückwärts | n.i.o | Die Drohne reagiert nicht auf den Befehl. |
| Links | i.o | Die Drohne reagiert auf den Befehl „Links“. Eine unnormale Verzögerung ist aber zu erkennen. |
| Rechts | n.i.o | Die Drohne reagiert nicht auf den Befehl. |
| Turn ClockWise | i.o | Die Drohne reagiert sofort auf den Befehl. |
| Turn Counter ClockWise | i.o | Die Drohne reagiert sofort auf den Befehl. |

Allgemeine Bemerkung : Keine

### Testbericht III

Datum: 04.05.2016

Ort: St. Louis

Bereich: Outdoor

Wetter: Leicht windig und unebener Boden

Tester: Alexis Stephan, Gabriel Urech, Simon Henzmann

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **T** | **Zeit** | **Vorkommnisse** | **Arbeiten** | **Tester** |
| 1 | 1846-1847 | Hover Ok, Macht vor und zurück Bewegungen die er nicht sollte. Abbruch durch umkehren. Handerkennung nicht optimal | Das Programm wird überarbeitet | A.S |
| 2 | 1900-1901 | Start OK; Hover OK; +/- Y geht nicht; Vorn- und rückwärtsgeht. Test Abbruch. | Einige Änderungen am Programm  Aschen vertauscht | A.S |
| 3 | 1925-1926 | Start OK; Rest ist noch unkontrollierbar; Test Abbruch | Einige Änderungen am Programm | A.S |
| 4 | 1927-1927 | Start OK, Landung OK | Freischaltung der Achsen | G.U |
| 5 | 1928-1928 | Start OK; Auswertung nicht brauchbar wegen zu viel Wind |  | G.U |
| 6 | 1932-1933 | Start Ok; Landung OK; Z Achse geht |  | G.U |
| 7 | 1934-1934 | Start OK; Landung OK; Hover OK; | Freischaltung der X- Achse | A.S |
| 8 | 1950-1951 | Start OK; Landung OK; Hover OK; X und Y verwechselt |  | A.S |
| 9 | 1953-1953 | Start OK; Landung OK; Hover OK; Y +/- Nicht Optimal |  | A.S |
| 10 | 1957-1957 | Y +/- zu wenig Speed |  | G.U |
| 11 | 1958-1958 | Start OK; Landung OK; Hover OK; Vor- und Rückwärts OK; Links Recht noch nicht so OK | 0/100 nicht gut | A:S |

Hauptproblem:

* 0/100 machte Schwierigkeiten
* Arrays machte Probleme
* +/- Vertauscht
* X und Y vertauscht

### Testbericht IV

Datum: 13.05.2016

Ort: Muttenz

Bereich: Labor

Wetter: Kein Wind, Begrenzter Bereich

Tester: Alexis Stephan, Gabriel Urech, Simon Henzmann, Jens Gais, Patrick Grubert

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **T** | **Zeit** | **Vorkommnisse** | **Arbeiten / Erkenntnisse** | **Tester** |
| 1 | 0958-1000 | Start: Ok, Hover Ok, Land Ok; Links/Rechts Ok; Vor-und rückwärts ok, Turn clockwise und counterclockwise Nok, | clockwise und counterclockwise Nok, (es war ein Bedienungsfehler) | A.S |
| 2 | 1000-1002 | Start: Ok, Hover Ok, Land Ok; Links/Rechts Ok; Vor-und rückwärts ok, CC ok |  | S.H |
| 3 | 1005-1010 | Start: Ok, Hover Ok, Land Ok; Links/Rechts Ok; Vor-und rückwärts ok, CC ok  Sicherheitslandung ok  Hoch u runter Reaktion nicht optimal. |  | S.H |
| 4 | 1030-1035 | Start: Ok, Hover Ok, Land Ok; Links/Rechts Ok; Vor-und rückwärts ok, CC ok  Sicherheitslandung ok  Hoch u runter ok | Wechsel auf Exponentialfunktion | J.G |
| 5 | 1040-1045 | Start: Ok, Hover Ok, Land Ok; Links/Rechts Ok; Vor-und rückwärts ok, CC ok  Sicherheitslandung ok  Hoch u runter ok |  | S.H |
| 6 | 1932-1933 | Start: Ok, Hover Ok, Land Ok; Links/Rechts Ok; Vor-und rückwärts ok, CC ok  Sicherheitslandung ok  Hoch u runter ok |  | P.G |
| 7 | 1934-1934 |  |  |  |
| 8 | 1950-1951 |  |  |  |
| 9 | 1953-1953 |  |  |  |
| 10 | 1957-1957 |  |  |  |
| 11 | 1958-1958 |  |  |  |

# Anhang

## Abbildungsverzeichnis

## Tabellenverzeichnis

## Abkürzungen

## Quellenverzeichnis

[Abbildung 1 Systemübersicht 12](file:///C:\Users\Simon\Dropbox\Projekt%206.%20Semester\3_Dokumentation\Dokumentation_QuadroMotion_00V13.docx#_Toc451202514)

[Abbildung 2 Blockschaltdiagramm QuadroMotion 13](file:///C:\Users\Simon\Dropbox\Projekt%206.%20Semester\3_Dokumentation\Dokumentation_QuadroMotion_00V13.docx#_Toc451202515)

[Abbildung 3 Zustandsdiagramm 14](file:///C:\Users\Simon\Dropbox\Projekt%206.%20Semester\3_Dokumentation\Dokumentation_QuadroMotion_00V13.docx#_Toc451202516)

[Abbildung 4 Leap Motion 21](file:///C:\Users\Simon\Dropbox\Projekt%206.%20Semester\3_Dokumentation\Dokumentation_QuadroMotion_00V13.docx#_Toc451202517)

[Abbildung 5 Myo 21](file:///C:\Users\Simon\Dropbox\Projekt%206.%20Semester\3_Dokumentation\Dokumentation_QuadroMotion_00V13.docx#_Toc451202518)

1. <http://www.pctipp.ch/tests/hardware/zubehoer/artikel/bewegungssteuerung-leap-motion-im-fuchteltest-69216/> (16.05.2016) [↑](#footnote-ref-1)
2. <http://www.digitaltrends.com/pc-accessory-reviews/myo-gesture-control-armband-review/> [↑](#footnote-ref-2)
3. <http://drohnenvergleich.ch/drohnenmarken/parrot/parrot-ar-drone-2-0/> (16.05.2016) [↑](#footnote-ref-3)
4. <https://wiki.bitcraze.io/projects:crazyflie2:index> (16.05.2016) [↑](#footnote-ref-4)