



**FACHHOCHSCHUL-BACHELORSTUDIENGANG**

**Automatisierungstechnik**

---

# GRAPHICAL USER INTERFACE FÜR QUADROPTER RESCUE DROHNE

**ALS BACHELORARBEIT EINGEREICHT**

**zur Erlangung des akademischen Grades**

**Bachelor of Science in Engineering**

**von**

**Ing. Daniel Rumetshofer**

**21.04.2011**

---

Betreuung der Bachelorarbeit durch

Dr. Clemens Derndorfer



Campus Wels

---

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt, die den benutzten Quellen entnommenen Stellen als solche kenntlich gemacht habe und dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

.....  
Ing. Daniel Rumetshofer

Steyregg, 21.04.2011

### ***Kurzbeschreibung***

Um das Potential eines gewöhnlichen Quadrocopters zu erweitern, wurde, angelehnt an den Bewerb RoboCup Rescue League, eine Quadrocopter Rescue Drohne aufgebaut.

In dieser Arbeit wird das Konzept und die benötigte Funktionalität des Gesamtsystems einer Rescue Drohne vorgestellt.

Um die bestehende Plattform des einfachen Quadrocopters von NG-UAVP<sup>1</sup>, entsprechend erweitern zu können, wurde eine Windowsoberfläche entwickelt, die eine Implementierung von GPS, Videoübertragung, digitaler Datenübertragung und weiterer Funktionen ermöglicht.

Weiters wurden Möglichkeiten gesucht, die manuelle Steuerung per RC-Fernbedienung durch ein intuitiveres Konzept zu ersetzen.

Schlüsselbegriffe: GUI, Quadrocopter

### ***Abstract***

To improve the potential of an ordinary quadrocopter, a prototype of a quadrocopter rescue drone was built, based on the Rules of the competition RoboCup Rescue League.

In this paper the concept and the necessary functionalities of the overall system is introduced.

To improve the existing platform, in this case the NG-UAVP, a windows graphical user interface was developed to allow the implementation of GPS, video transmission, digital data transmission and further functions.

Another goal was to replace the manual control by a more intuitive system.

Keywords: GUI, quadrocopter



Abbildung: flugfähiger Quadrocopter Prototyp (ohne Rotorblätter)

---

<sup>1</sup> NG-UAVP: Next Generation Universal Video Plattform; Quelle: [ng.uavp.ch](http://ng.uavp.ch)

## *INHALTSVERZEICHNIS*

|       |   |    |
|-------|---|----|
| 1.    | Einleitung.....   | 5  |
| 2.    | NG-UAVP Plattform .....   | 6  |
| 2.1   | Konzept.....  | 6  |
| 2.2   | Hardware .....  | 7  |
| 2.2.1 | LPC Board.....  | 7  |
| 2.2.2 | Flightcontrol Board .....   | 8  |
| 2.2.3 | Sensor Board.....   | 8  |
| 2.3   | Software.....   | 9  |
| 2.4   | Weitere Informationen.....  | 9  |
| 3.    | Anforderungen an die Drohne hinsichtlich der RoboCup Rescue League..... | 10 |
| 3.1   | Modellierung des Konzepts .....   | 10 |
| 3.1.1 | MiniPC.....   | 11 |
| 3.1.2 | Miniboard .....   | 12 |
| 3.1.3 | Headtracking.....   | 12 |
| 3.1.4 | Kamera und Videoübertragung.....  | 12 |
| 3.1.5 | LRF - Laser Range Finder .....  | 12 |
| 3.1.6 | Erweiterungen.....  | 13 |
| 4.    | Innovative Steuerung.....   | 14 |
| 4.1   | Manuelle Steuerung.....   | 14 |
| 4.1.1 | Nintendo Wii-Mote .....   | 15 |
| 4.1.2 | Joystick .....  | 15 |
| 4.1.3 | SpaceNavigator - 3D Maus.....   | 16 |
| 4.2   | Autonomer Flug.....   | 17 |
| 5.    | Entwicklung der Windowsoberfläche / GUI .....                           | 18 |
| 5.1   | Modellierung der GUI.....   | 18 |
| 5.1.1 | Visualisierung.....   | 18 |
| 5.1.2 | Control.....  | 19 |
| 5.1.3 | Datenübertragung.....   | 19 |
| 5.2   | Programmierung.....   | 21 |
| 5.2.1 | Treiber der 3D-Maus .....   | 21 |
| 5.2.2 | 3D Modell .....   | 21 |
| 5.3   | Fertige Oberfläche .....  | 22 |
| 6.    | Ausblick.....   | 23 |
| 7.    | Quellenverzeichnis .....  | 24 |

# 1. EINLEITUNG

Bei dem Bewerb der RoboCup Rescue League gibt es seit 2010 eine eigene Kategorie/Arena für Quadcopter Drohnen. Um neben anderen Disziplinen auch in dieser Kategorie des Bewerbs teilnehmen zu können, beschloss das Roboter Team der FH-Wels einen eigenen Quadcopter aufzubauen.

Der Quadcopter sollte zwar nicht ausschließlich für diesen Bewerb entwickelt werden, sondern universell auch für andere Aufgaben eingesetzt werden, doch gibt der Bewerb sehr anspruchsvolle und reale Anforderungen an die Roboter vor. Daher wird die geforderte Funktionalität zunächst stark an das Reglement des Bewerbs angelehnt.

Da es bereits viele Open Source Projekte flugfähiger Quadcopter gibt, wurde beschlossen, keine Drohne von Grund auf neu zu entwickeln, sondern auf eine bestehende Plattform aufzusetzen und diese zu erweitern.<sup>2</sup>

Die Drohnen sind in der Regel für manuelle Fernbedienung konzipiert. Viele dieser Projekte sind bereits in der Lage, auf GPS-Koordinaten zu reagieren. Dies stellt aber noch keine ausreichende Funktionalität für eine Rescue-Drohne dar, da noch Systeme wie Bilderkennung, und intelligenter Indoor-Flug fehlen.

Ziel dieser Arbeit ist die Entwicklung einer Windowsoberfläche (GUI, graphical user interface), als Basis, um die Implementierung komplexer Funktionen zu ermöglichen.

Zudem soll auch die manuelle Steuerung mittels RC-Fernbedienung durch ein intuitiveres Steuerungskonzept ersetzt werden.

Als erstes wird die NG-UAVP Plattform der Drohne vorgestellt, und wie das Konzept hinsichtlich der hohen Anforderungen erweitert werden muss. Anschließend wird das Steuerungskonzept analysiert, und daraus schließlich die erforderliche Windows Oberfläche entwickelt.

---

<sup>2</sup> Der Auswahlprozess und die Auswahlkriterien einer geeigneten Plattform werden in der Bachelorarbeit von Andreas Berger [1] näher behandelt.

## 2. NG-UAVP PLATTFORM

Die NG-UAVP Plattform ist eine OpenSource Plattform, die entwickelt wurde, um die komplexe Lageregelung eines Quadrocopters zu übernehmen. Das Konzept umfasst Hardware und Software.

### 2.1 KONZEPT

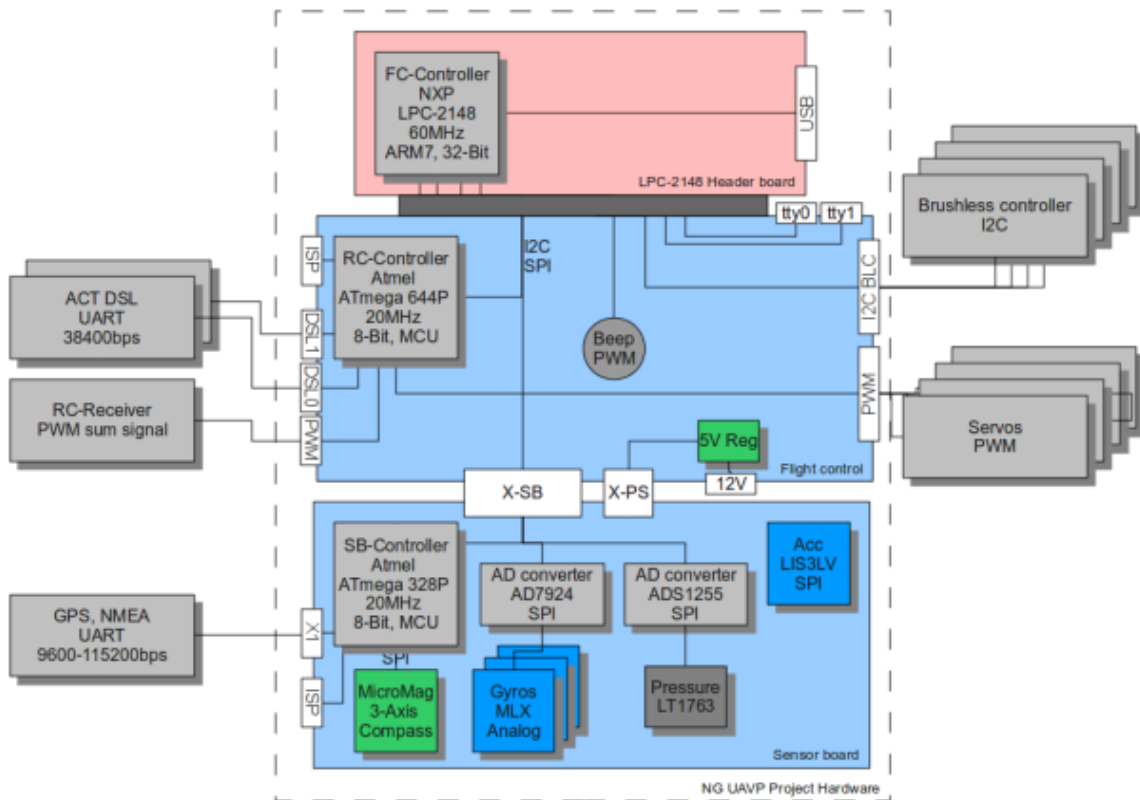


Abbildung: Prinzipieller Aufbau der NG UAVP Hardware<sup>3</sup>

<sup>3</sup> Quelle: [ng.uavp.ch/moin/Documentation/Construction](http://ng.uavp.ch/moin/Documentation/Construction)

## 2.2 HARDWARE

Es wurden von ng.uavp.ch die Platinen der Hardwareversion 0.22 bestellt, und die Bauteile manuell bestückt.

Die Hardware besteht im Wesentlichen aus folgenden Teilen:

- LPC Development Board
- Flightcontrol Board
- Sensor Board

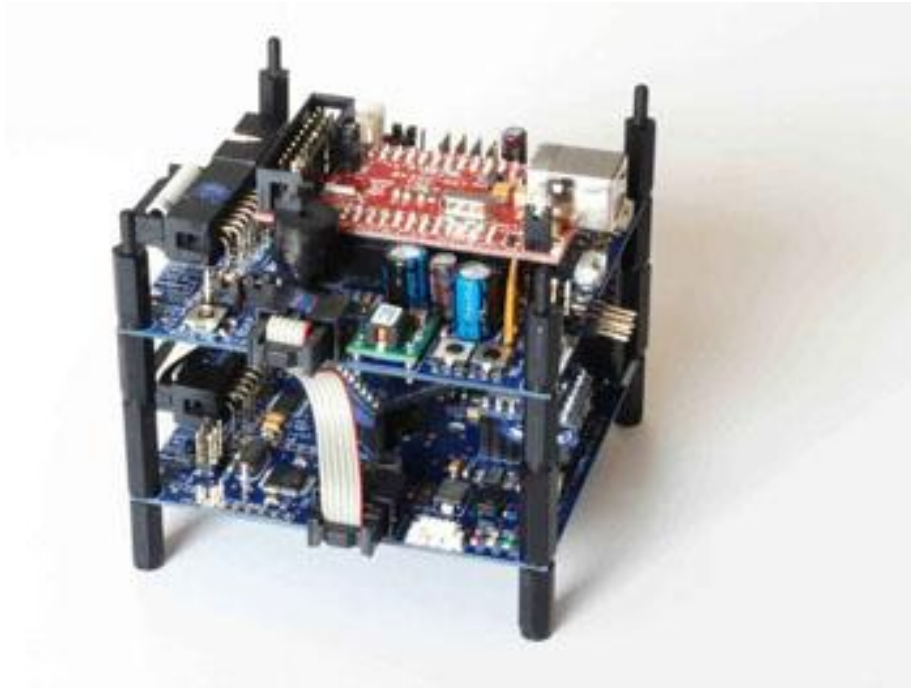


Abbildung: vollständige Hardware der NG-UAVP Plattform<sup>4</sup>

### 2.2.1 LPC BOARD

Das LPC Board ist eine kompakte Entwicklerplatine mit ARM7 CPU. Auf diesem Board werden sämtliche Grundfunktionen des Fliegens realisiert.



Abbildung: LPC Board<sup>5</sup>

---

<sup>4, 4</sup> Quelle: ng.uavp.ch

## 2.2.2 FLIGHTCONTROL BOARD

Auf dem Flightcontrol Board befindet sich das LPC-Board als Rucksackplatine. Weiters befindet sich ein Atmel 644p und die Spannungsversorgung des Systems auf der Platine. Der Atmel 644p kann für die Steuerung von RC-Servos benutzt werden, wurde von uns bislang aber nicht verwendet.



Abbildung: Flightcontrol Board inkl. LPC Board<sup>6</sup>

## 2.2.3 SENSOR BOARD

Auf dem Sensor Board befinden sich sämtliche Sensoren des Systems

- Gyroscope Drehwinkelsensoren (Gyros)
- 3 Achsen Beschleunigungssensor (Acc)
- GPS Empfänger
- 3 Achsen Magnetsensor
- Barometer

Zusätzlich befindet sich ein Atmel 328 Prozessor auf dem Board. Dieser Atmel-Prozessor muss programmiert werden, um den GPS-Empfänger bzw. den Magnetsensor verwenden zu können.

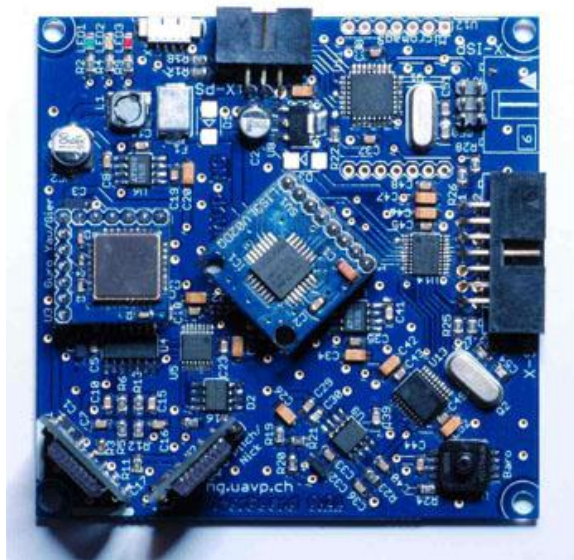


Abbildung: Sensorboard<sup>7</sup>

---

<sup>6, 6</sup> Quelle: ng.uavp.ch

## 2.3 SOFTWARE

Zur Grundfunktion des normalen Fliegens ist die Programmierung des LPC-Board notwendig.  
Softwareversion: NGOS Version 0.60

Die Software steht unter der Lizenz GPLv3+, eine gängige Lizenz für OpenSource Projekte.

## 2.4 WEITERE INFORMATIONEN

Weiter Informationen zur NG-Plattform sind der Bachelorarbeit von Andreas Berger [1], bzw. der Entwicklerhomepage [3] zu entnehmen.

### 3. ANFORDERUNGEN AN DIE DROHNE HINSICHTLICH DER ROBOCUP RESCUE LEAGUE

Aufgrund der komplexen Anforderungen an die Drohne hinsichtlich des Wettbewerbs RoboCup Rescue League ist es nicht möglich, die Drohne einfache per RC-Fernbedienung zu steuern. Vielmehr ist eine effektive manuelle Steuerung nötig, und es müssen Möglichkeiten für einen autonomen Flug geschaffen werden.

#### 3.1 MODELLIERUNG DES KONZEPTS

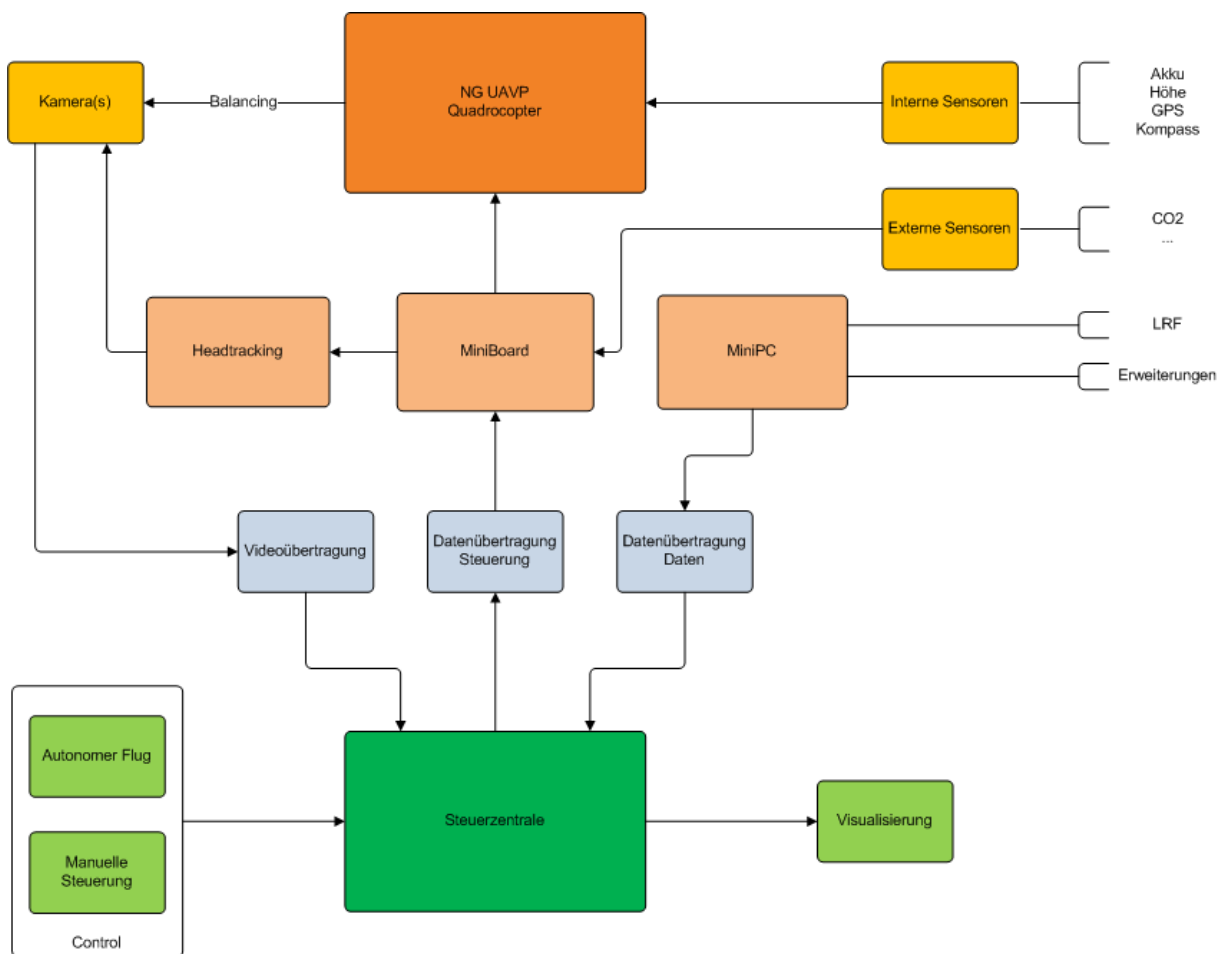


Abbildung: Modellierung Gesamtkonzept

### 3.1.1 MINIPC

Für rechenintensive Operationen ist eine kompakte, leichte x86 Plattform auf der Drohne erforderlich.

Der Nutzen besteht darin, bereits vorhandene Windowsprogramme und Treiber übernehmen zu können. Dies ist in erster Linie beim LaserRangeFinder und für das SDK der Microsoft Kinect Kamera nötig.

Als Hardware kommt ein Via-Epia-Pico-ITX PX10000G<sup>8</sup> Mainboard zum Einsatz. Dieses Mainboard stellt eine äußerst kompakte und leichte x86 Plattform zur Verfügung.

Wichtige Kenndaten:

- 1Ghz Via C7 CPU
- DDR2 SODIMM Sockel
- Onboard Grafik

Wichtige Schnittstellen:

- RJ45 Ethernet
- VGA
- SATA
- 4 x USB
- RS232



Abbildung: Pico ITX Mainboard

<sup>8</sup> Quelle: <http://cartft.com/catalog/il/822>

### 3.1.2 MINIBOARD

Zur Datenübertragung der Steuerkommandos an die Drohne ist eine zusätzliche Platine nötig. Das sogenannte Miniboard ist eine Atmel Microcontrollerplatine. Diese erhält per XBee-Übertragung die Steuerkommandos und wandelt diese in ein PWM-Signal um, welches von der Drohne interpretiert werden kann.

Zusätzlich können die freien Anschlüsse des Microcontrollers für die Auswertung zusätzlicher Sensoren verwendet werden.

Nähere Informationen sind der Bachelorarbeit von Hr. Zangenfeind Manuel [2] zu entnehmen.

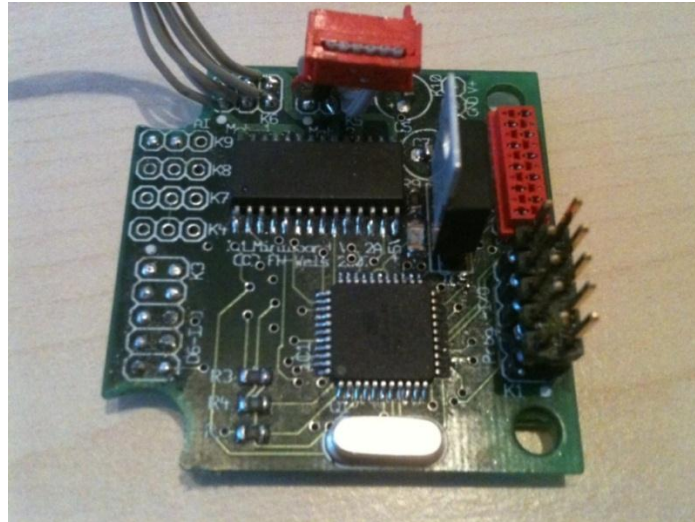


Abbildung: Miniboard

### 3.1.3 HEADTRACKING

Es soll für optimale Ausnutzung der Onboardkamera ein Headtrackingsystem entwickelt und implementiert werden. Mit einem derartigen System ist es möglich, die Kamera durch Kopfsteuerung zu positionieren. In Verbindung mit einer Videobrille entsteht dadurch ein sehr realistisches Gefühl der freien Sicht.

### 3.1.4 KAMERA UND VIDEOÜBERTRAGUNG

Eine Onboardkamera soll mithilfe einer geeigneten Videoübertragungsstrecke Livebilder von der Drohne an die Steuerzentrale schicken.

Die Kamera wird durch 2 Servos von der NG-Plattform waagrecht ausgeregelt. Dadurch entsteht ein ruhiges Bild unabhängig von der Lage der Drohne.

Zusätzlich soll die Kamera durch das Headtracking System gesteuert werden.

#### 3.1.4.1 Analoges System

Analoge Videokamera und 5 Ghz Videoübertragungssystem mit TX an der Drohne, und RX an der Bodenstation

#### 3.1.4.2 Digitales System

Netzwerkkamera bzw. USB-Kamera angeschlossen an den MiniPC und Datenübertragung über die breitbandige Datenverbindung (Wlan, UMTS)

### 3.1.5 LRF – LASER RANGE FINDER

Damit sich die Drohne im Indoorflug an Wänden orientieren kann, wird ein LRF eingesetzt. Mit dem LRF ist es möglich eine 2D bzw. 3D Karte eines Gebäudes zu erstellen. Diese Informationen können ausgewertet werden und als Unterstützung der Navigation dienen.

### 3.1.6 ERWEITERUNGEN

Eine mögliche, sinnvolle Erweiterung stellt die Microsoft Kinect Kamera dar. Dieses Kamerasystem wurde für die Microsoft Xbox360 entwickelt und kann Gesten des menschlichen Körpers sehr genau im dreidimensionalen Raum erkennen.

Diese Erweiterung wird in dieser Arbeit nicht näher behandelt.

## 4. INNOVATIVE STEUERUNG

### 4.1 MANUELLE STEUERUNG

Einer der wichtigsten Punkte dieser Entwicklung war die Verbesserung der Steuerung. Das Fliegen per RC-Fernbedienung funktioniert für den geübten RC-Piloten zwar gut, jedoch ist es für Anfänger beinahe unmöglich, die Drohne gezielt zu navigieren.

Die Anforderung an die Steuerung ist, dass sie intuitiv und möglichst einfach zu erlernen ist. Zudem muss sie ausreichende Möglichkeiten für die Funktionalität bieten.

Für die Steuerung der Drohne sind min. 4 analoge Achsen nötig

- Nick (oder auch Pitch genannt)
- Roll
- Yaw (oder auch Gier genannt)
- Throttle (Gas, Höhe)



Abbildung: Drehachsen<sup>9</sup>

Hierfür wurden verschiedene Steuerungskonzepte in Erwägung gezogen, und analysiert

- Nintendo Wii-Mote
- Joystick
- 3D-Maus: SpaceNavigator von 3D Connexion

<sup>9</sup> Quelle: <http://www.qc-copter.de/wiki/images/a/a5/Quadrocopter-schema.jpg>

### 4.1.1 NINTENDO WII-MOTE



Abbildung: Nintendo Wii-Mote<sup>10</sup>

VT:

- Viele unterschiedliche Steuerelemente

NT:

- Treiber nur bedingt verfügbar
- Auf einer Hand zu wenige analoge Achsen, daher beide Hände zur Navigation nötig
- Nicht intuitiv
- Sensoren nicht exakt genug

### 4.1.2 JOYSTICK



Abbildung: Joystick<sup>11</sup>

VT:

- Hochwertig
- Viele zusätzliche Tasten und Steuerelemente

NT:

- Nicht intuitiv
- zu wenig Achsen, bzw. ungünstige Anordnung der Steuerelemente

---

<sup>10</sup> Quelle: <http://www.gamekool.com/images/GKWA-00001.jpg>

<sup>11</sup> Quelle: [http://regmedia.co.uk/2007/11/16/flightstickx\\_prod1.jpg](http://regmedia.co.uk/2007/11/16/flightstickx_prod1.jpg)

### 4.1.3 SPACENAVIGATOR – 3D MAUS



Abbildung: 3D Connexion Spave Navigator<sup>12</sup>

VT:

Hochwertig

Hohe Genauigkeit

USB Treiber vorhanden

eine Hand frei

Genügend Achsen + 2 Buttons

Intuitiv da passend zum Drohnenmodell

Die 3D-Maus von 3D Connexion besteht durch die hochwertige Verarbeitung. Die Anordnung der Achsen ist identisch mit den realen Drehachsen der Drohne, und daher sehr intuitiv. Auch dass die Drohne dadurch mit nur einer Hand vollständig steuerbar ist kann in bestimmten Situationen ein Vorteil sein.

Dieser Controller hat ideale Voraussetzungen als geeignete Steuerung eines Quadrocopters. Ob sich die Erwartungen auch im realen Flug erfüllen, wird sich erst nach vollständiger Implementierung und Testflügen zeigen.

---

<sup>12</sup> Quelle: <http://nicogeisler.de/wp-content/uploads/2009/03/3dx-spacnavigator.jpg>

## 4.2 AUTONOMER FLUG

Um autonomen Flug der Drohne zu ermöglichen, gibt es unterschiedliche Möglichkeiten, die je nach Aufgabe individuell betrachtet werden müssen.

Eine einfache Möglichkeit wäre beispielsweise, eine Liste von GPS-Koordinaten, sogenannte Waypoints, vorzugeben. Die Drohne kann diese Waypoints dann der nach Reihe anfliegen.

Die Abbildung zeigt ein einfaches Beispiel, wie eine Navigation per Wegpunkte aussehen kann.

Dabei geben die grünen Punkte die vorgegebenen Wegpunkte an.

Der blaue Punkt markiert die Startzone.

Der rote Punkt ist die aktuelle Position der Drohne.



Abbildung: Beispiel für autonomen Flug per GPS-Waypoints<sup>13</sup>

Es müssen aber auch noch weitere Schnittstellen bei autonomem Flug berücksichtigt werden. So kann beispielsweise die Karte des LRF verwendet werden, um Indoor zu navigieren. Auch die MS-Kinect Kamera kann Objekte erkennen und darauf autonom reagieren.

<sup>13</sup> Quelle: <http://www.mkstation.de/public/thumbs/318.jpg>

## 5. ENTWICKLUNG DER WINDOWSOBERFLÄCHE / GUI

### 5.1 MODELLIERUNG DER GUI

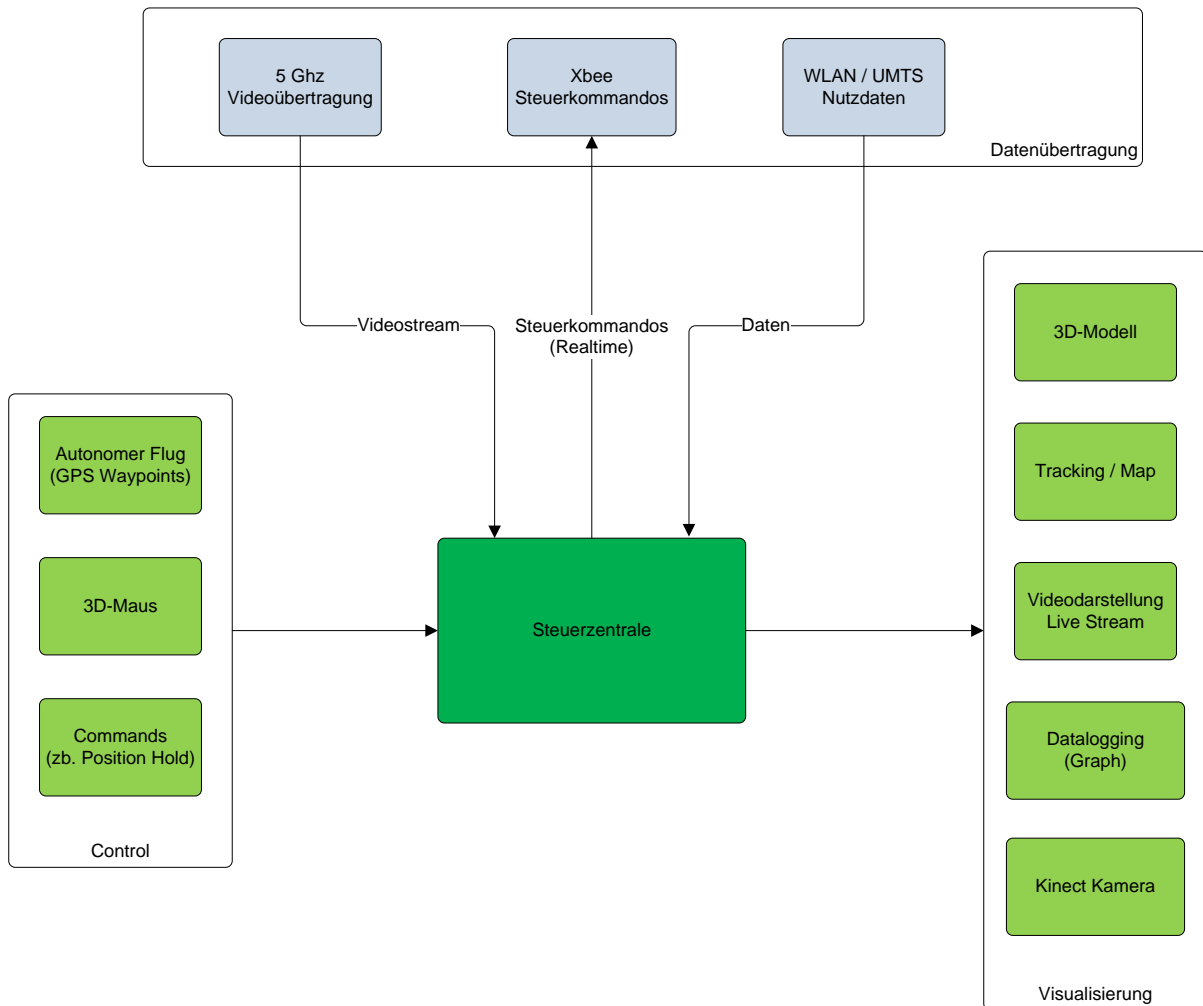


Abbildung: Modell der Steuerzentrale

#### 5.1.1 VISUALISIERUNG

Es sollen relevante Daten und Informationen der Drohne visualisiert werden.

- Videoübertragung der Kamera
- 3D Modell der Drohne und deren Lage
- Tracking der Flugroute in einer Karte (zb. Google Maps) bei Outdoorflight
- 2D Mapping bei Indoorflight
- Logging der Sensorwerte (zb. Flughöhe)
- Darstellen der Informationen der MS Kinect Kamera

## 5.1.2 CONTROL

Es sollen unterschiedliche Möglichkeiten zur Steuerung der Drohne ermöglicht werden.

- Manuelle Steuerung durch 3D-Maus
- Autonom durch Vorgabe von GPS – Wegpunkten
- Zusätzliche Übertragung von Kommandos wie Position Hold, Coming Home, Height Hold usw.

Denkbar wäre auch eine Kombination aus den oben genannten Steuerarten. Zum Beispiel sollte eine rechnerunterstützte manuelle Steuerung die Kollisionen verhindert. Diese Möglichkeiten werden in dieser Arbeit nicht weiter betrachtet.

## 5.1.3 DATENÜBERTRAGUNG

### 5.1.3.1 Steuerkommandos

Um die Drohne vom PC aus zu steuern, erfolgt die Übertragung der Steuerkommandos über eine digitale XBee Funkverbindung.

Die Steuerkommandos werden von der Zentrale an ein XBee Modul übergeben, welches die Kommandos per Funk an den Empfänger zur Drohne übermittelt. Mithilfe des Miniboards werden die Kommandos ausgewertet und ausgeführt.

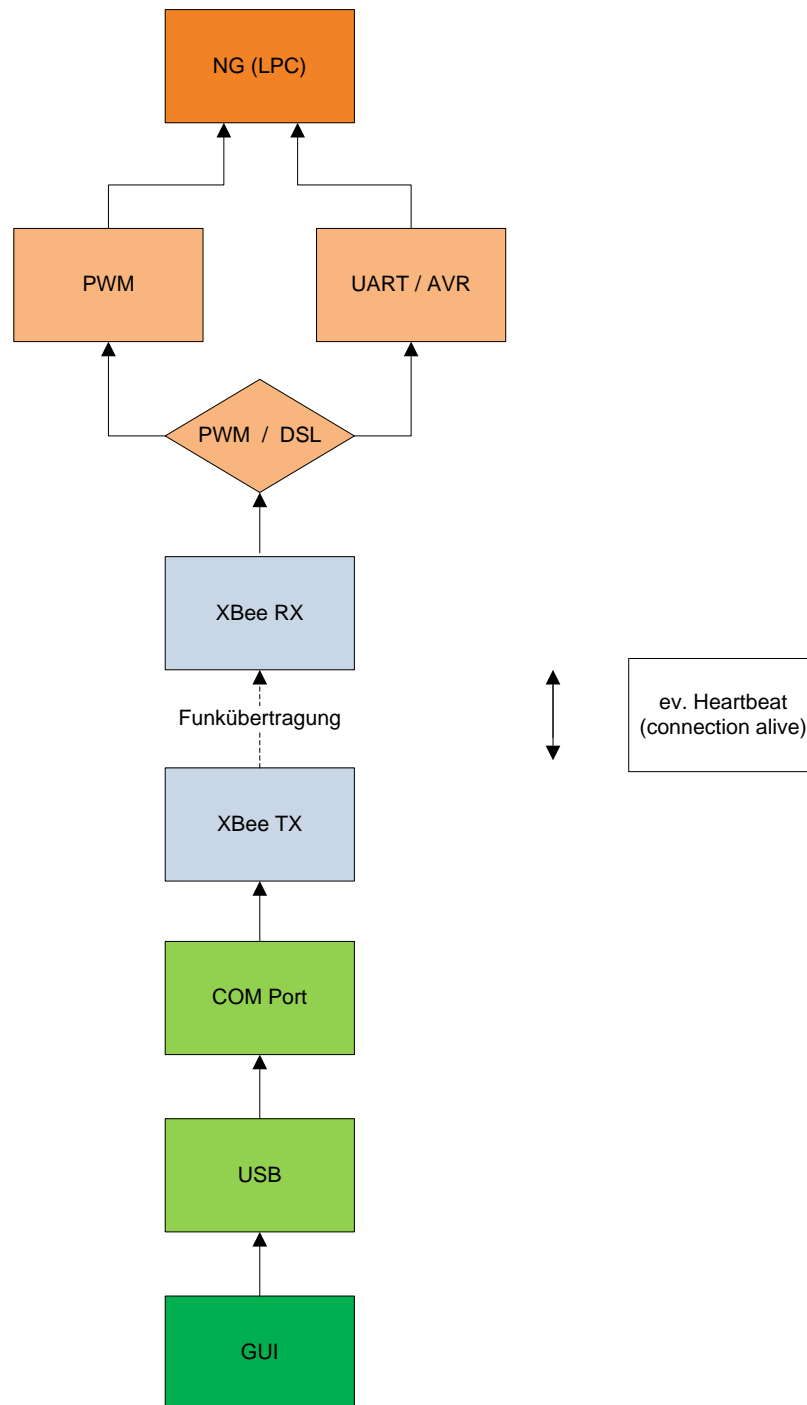


Abbildung: Verbindungsaufbau per XBee

### 5.1.3.2 Nutzdaten

Komponenten wie LRF erzeugen große Datenmengen, die zur Zentrale übertragen werden müssen. Dafür ist eine breitbandige Übertragungsstrecke nötig.

Im Indoorbetrieb kann dies per WLAN erfolgen.

Im Outdoorbetrieb kann mithilfe eines UMTS-Modems eine breitbandige Internetverbindung hergestellt werden.

## 5.2 PROGRAMMIERUNG

Die Programmierung der Oberfläche wurde mithilfe von Microsoft Visual Studio 2010 Express durchgeführt.

Es handelt sich dabei um eine Windows Forms Anwendung, deren Funktionen in C# programmiert wurden.

### 5.2.1 TREIBER DER 3D-MAUS

Der Treiber für die 3d-Maus konnte aus einem Testprogramm TDxInput kopiert und eingebunden werden.

```
using TDx.TDxInput;
```

Die Methoden des Treibers werden durch geeignete Eventhandler aufgerufen. Details sind dem Code zu entnehmen.

### 5.2.2 3D MODELL

Die Realisierung des 3D-Modells der Drohne wurde aus einem Testprogramm von Microsoft abgeleitet und entsprechend adaptiert. Die graphische Darstellung verwendet Routinen aus dem XNA Game Environment.

```
using Microsoft.Xna.Framework;  
using Microsoft.Xna.Framework.Content;  
using Microsoft.Xna.Framework.Graphics;
```

Die Eingabewerte der 3D-Maus werden eingelesen, und die Lage des Drohnenmodells damit angepasst, sodass das Modell seine Lage ähnlich verändert wie die echte Drohne.

### 5.3 FERTIGE OBERFLÄCHE

Die Erkenntnisse und Anforderungen aus der Modellierung der Steuerzentrale wurden in einer Windows-Forms Anwendung realisiert. Die Oberfläche, siehe Abbildung, wurde so gestaltet, dass alle Kontroll- und Anzeigemöglichkeiten übersichtlich am Bildschirm angeordnet sind. Die Visualisierung, welche viel Platz in Anspruch nimmt, wurde in Tabs kategorisiert. Es wurden bei der Gestaltung der Oberfläche die Grundlagen von Mensch-Computer-Interfaces lt. [8] berücksichtigt. Die sogenannten „8 golden rules“ gewährleisteten eine benutzerfreundliche Oberfläche.

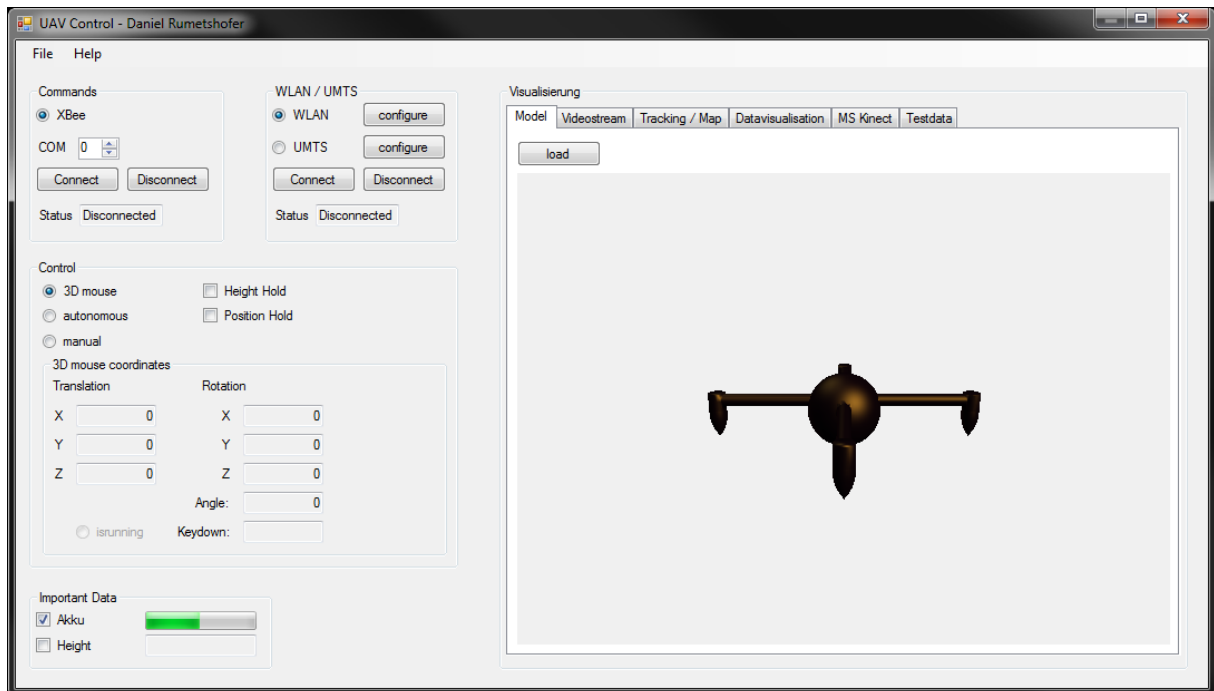


Abbildung: Fertige Windowsoberfläche als Steuerzentrale der Drohne

## 6. AUSBLICK

Zunächst werden die Steuerkommandos aus C# an den Com-Port zum XBee-Sender übertragen. Dies ist einfach zu realisieren. Doch muss berücksichtigt werden, dass ein C# Programm unter Windows normalerweise nicht den Echtzeitanforderungen gerecht werden kann. Es wird zwar davon ausgegangen, dass die Performance ausreichend sein wird, aber das muss sich in den ersten Tests noch beweisen.

Sollte es Probleme mit der Echtzeitfähigkeit geben, so werden die Steuerkommandos der 3D-Maus in Labview eingelesen, und verschickt. Labview-Realtime ist auch unter Windows Echtzeitfähig.

Ein entsprechendes Programm wurde in Labview bereits vorbereitet. Die Mausdaten können als Joystickdaten akquiriert werden. Auch das Senden an den Com-Port ist in Labview einfach zu realisieren. Die relativen Mausdaten müssen noch in absolute Koordinaten umgerechnet werden.

Weitere Programmieraufgaben:

- Einlesen der Sensordaten und visualisieren
- Einlesen der GPS-Daten und visualisieren (Google-Maps / Google-Earth verwenden)
- Entwicklung der Funktionen für Autonomen Flug. (modular, für unterschiedliche Herausforderungen)
- Empfindlichkeit der 3D-Maus einstellbar (unempfindlich für Anfänger)

## 7. QUELLENVERZEICHNIS

- [1] Andreas Berger, „Recherche, Gegenüberstellung und Bewertung von Hardwarekomponenten für den Aufbau eines Multikopters“, 2011
- [2] Zangenfeind Manuel, „Datenübertragung und Signalmodulation zur Steuerung eines Quadropters“, 2011
- [3] [ng.uavp.ch](http://ng.uavp.ch), „Open Source Next Generation Multicopter“, 16.04.2011
- [4] [www.cartft.com](http://www.cartft.com), Pico ITX Mainboard, 05.04.2011
- [5] [www.robocuprescue.org](http://www.robocuprescue.org), RoboCup Rescue 2011, 02.04.2011
- [6] [www.robocup.org](http://www.robocup.org), RoboCup, 02.04.2011
- [7] Harald Störrle, „UML 2 für Studenten“, Pearson Studium, München, 2005
- [8] Ben Shneiderman, „Designing the User Interface“, Pearson Higher Education, 5. Auflage, 2010