

# **User Manual**

**For the FINken Quadrocopters**

Sebastian Mai

26. August 2014

# Inhaltsverzeichnis

<b>1</b>	<b>Hardware</b>	<b>3</b>
1.1	Komponenten . . . . .	3
1.1.1	Aktuatoren . . . . .	3
1.1.2	Flight Control . . . . .	3
1.1.3	Sonar . . . . .	3
1.1.4	Farbsensor . . . . .	4
1.1.5	Bluetooth . . . . .	4
1.2	Fernbedienung . . . . .	4
1.2.1	Manuelle Steuerung . . . . .	4
1.2.2	Autopiloten-Modi . . . . .	4
1.3	Akkus und Ladegerät . . . . .	4
<b>2</b>	<b>Software</b>	<b>6</b>
2.1	Paparazzi-Center . . . . .	6
2.1.1	Groundstation . . . . .	6
2.1.2	Data-Link . . . . .	6
2.1.3	Messages . . . . .	6
2.2	Code . . . . .	7
2.3	Modell . . . . .	7
2.3.1	Sensormodell . . . . .	9
2.3.2	Aktuatoren . . . . .	9
2.3.3	Systemmodell . . . . .	9
2.3.4	Umgebungsmodell . . . . .	9

# 1 Hardware

## 1.1 Komponenten

### 1.1.1 Aktuatoren

Die Motoren des Quadrocopters (und letztendlich auch die Propeller) werden von sogenannten Speedcontrollern (ESCs) gesteuert, die aus dem Gleichstrom der Batterie einen 3-Phasen Wechselstrom machen der die Motoren antreibt. Als Steuereingang verwenden die Speedcontroller ein PWM-Signal, das vom FlightController erzeugt wird. Sind an den Propellern Stresszeichen (herausgeschlagene Ecken, weiß verfärbter Kunststoff) zu sehen, sind die Propeller unverzüglich zu tauschen, um weitere Schäden am Flugroboter zu vermeiden. Beim Austausch von Propellern ist auf korrekte Orientierung (Beschriftung oben) und Drehrichtung (siehe Abb. 1) zu achten.

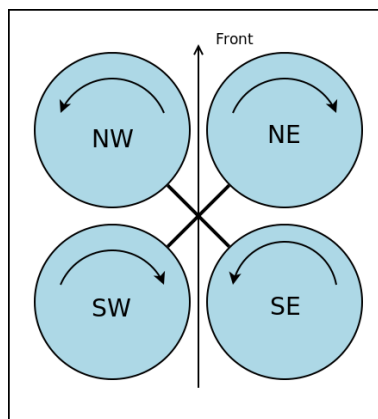


Abbildung 1: Drehrichtung der Rotoren

### 1.1.2 Flight Control

Als FlightControl-Board verwenden wir ein Board vom Typ Paparazzi-Lia, auf dem bereits ein 10-DOM Lagesensor (IMU) integriert ist und auf dem die Paparazzi-Firmware läuft. Hier werden sowohl die Low-Level-Lageregelung als auch logisch höhere Funktionen wie Navigation umgesetzt und alle Aktuatoren und Sensoren angeschlossen.

### 1.1.3 Sonar

Die Sonarsensoren sind neben der IMU die wichtigsten Sensoren. Die Lagesensoren sind über I<sup>2</sup>C an die Flightcontrol angebunden. Die I<sup>2</sup>C-Adressen der Sonarsensoren sind 0x71 bis 0x74, wobei 0x71 die Adresse des vorderen Sensors ist und im Uhrzeigersinn nummeriert wird.

Der Code für die Sensortreiber befindet sich in `sw/airborne/modules/sonar/sonar_array.c`.

#### 1.1.4 Farbsensor

Der Farbsensor ist genau wie die Sonarsensoren per I<sup>2</sup>C angebunden. Ein Treiber muss noch geschrieben werden<sup>1</sup>.

#### 1.1.5 Bluetooth

Bluetooth wird zum Empfang der Telemetriedaten und zur Steuerung des Autopiloten<sup>2</sup> genutzt. Um Bluetooth nutzen zu können muss der Quadrocopter mit dem Rechner mit der Bodenstation gepairt werden – das ist leider bei jedem Start von neuem notwendig, obwohl der Bluetoothstandart auch automatisches Pairing vorsieht. *Die PIN zum Pairing ist 1234.*

### 1.2 Fernbedienung

Die Fernbedienung dient zur manuellen Steuerung des Quadrocopters und zum Umschalten der Autopiloten-Modi.

#### 1.2.1 Manuelle Steuerung

Der Wichtigste Schalter ist open Rechts: Der Kill-Schalter. Dieser Schalter funktioniert quasi als Not-Aus – alle Motoren werden sofort ausgeschalten. Zum Losfliegen muss dieser Schalter deaktiviert werden. Wichtig sind außerdem die beiden Steuerknüppel der Fernbedienung: Links ist Schub und Yaw, rechts Pitch und Roll.

#### 1.2.2 Autopiloten-Modi

Mit dem drei Wege-Schalter oben rechts an der Fernbedienung (neben dem Kill-Schalter) kann der Modus des Autopiloten gewechselt werden. Momentan sollten die Stellung unten – „Attitude Mode“ zum manuellen Fliegen mit Fernsteuerung und die Stellung oben – „Nav Mode“ zum autonomen Flug nach „flight-plan“. Im Autopilotmodus wird je nach Einstellung der Schub durch den Schubhebel an der Fernsteuerung limitiert. *Schaltet man wieder auf manuellen Modus um, versucht der Quadrocopter sofort diese Obergrenze als aktuellen Schub umzusetzen!*

### 1.3 Akkus und Ladegerät

Die Akkus die wir benutzen sind 2-Zellen Litium-Polymer-Akkus mit 950mAh Kapazität. Zum Laden müssen die Akkus mit beiden Steckern mit dem Ladegerät verbunden werden, danach kann man mit dem LiPo-Balance-Programm die Akkus laden. Wichtig ist, dass im Ladegerät 2S ausgewählt ist und die Stromstärke nicht über 2A eingestellt

---

<sup>1</sup>Zur Orientierung kann der Treiber für die Sonarsensoren genutzt werden.

<sup>2</sup>Es geht als nicht um Manuellen Flug, sondern um Kommandos wie SStarten, Landen, Wegpunkt anfliegen etc.

wird. (mit 1A Stromstärke zu laden ist schonender für die Akkus, über 4A können die Akkus anfangen zu brennen) Die Akkus sollten nicht ohne Aufsicht geladen werden.

Am Quadrocopter wird der Akku mit dem JST-Connector angeschlosse. Die Spannungsüberwachung ist intern angeschlossen.

Um neue Firmware zu flashen muss der Quadrocopter über USB mit dem Rechner verbunden werden, der Akku sollte dabei nicht angeschlossen sein, um ein unkontrolliertes Loslaufen der Motoren zu verhindern.

## 2 Software

### 2.1 Paparazzi-Center

Das Paparazzi Center bietet bequeme Möglichkeiten zum Kompilieren und Flashen der Firmware und zum starten der zum Flug benötigten Tools. Um die Anwendung auszuführen muss im Paparazzi-Verzeichnis der Befehl „./paparazzi“ eingegeben werden. Wenn Änderungen an der Konfiguration für das Bodensegment gemacht werden muss das make-Target „gcs“ neu kompiliert werden.

Wenn man mit dem Quadrocopter fliegen will sollte man in etwa folgendes tun:

1. Upload der neuen Firmware (falls neue Firmware geschrieben wurde)
2. Trennen von USB und verbinden der Batterie und Akkuüberwachung
3. Herstellen der Bluetooth-Verbindung (über das Betriebssystem)
4. Start einer Session (Button oben Rechts)
5. Fliegen
6. Kill-Switch auf „Kill“, abstecken der Batterie (und Akkuüberwachung)

#### 2.1.1 Groundstation

Die Groundstation dient zum steuern des Quadrocopters – hier sieht man diverse Informationen wie aktueller Batteriestand, Navigationsmodus ... Von hier können auch Kommandos wie „Takeoff“, „Land“, etc. gegeben werden.

#### 2.1.2 Data-Link

Die Groundstation bekommt ihre Informationen über Bluetooth 4.0. Da in BT4.0 keine Serielle Verbindung im Protokoll enthalten ist, muss diese Verbindung zuerst über ein externes Programm hergestellt werden. Danach kann sich das Data-Link-Tool von Paparazzi auf das neu erstellte Pseudoterminal `/dev/pts/X` verbinden.

Dazu muss in `/home/finken/bt-bridge/mkpty.sh` die MAC-Adresse des aktuellen Bluetoothmoduls einkommentiert werden. Danach kann das entsprechende Device (der Pfad von `mkpty.sh` angezeigt) in den entsprechenden Parameter des Data-Link-Programmes im Paparazzi-Center eingetragen werden. Danach kann das Data-Link-Programm gestartet werden und die Groundstation erhält alle Informationen über den Quadrocopter.

#### 2.1.3 Messages

Unter tools  $\Rightarrow$  messages befindet sich ein nützliches Werkzeug mit dem sich die Telemetriedaten ausgeben lassen – unter anderem auch die ausgabe der Sonar-Sensoren. Die angezeigten Daten können mit dem Real-Time-Plotter auch visualisiert werden.

## 2.2 Code

### Versionsverwaltung

Der Code wird aktuell verwaltet mit der Versionskontrollsoftware "git". Eine gute Referenz für die Benutzung von git findet man unter <http://www.ndpsoftware.com/git-cheatsheet.html>. Der Code wird gehostet unter <https://github.com/ovgu-FINken/paparazzi> und kann heruntergeladen werden mit „git clone“<sup>3</sup>.

### Organisation

Der Code im Paparazzi-Repository teilt sich für uns in zwei Bereiche ein: Einerseits die Konfigurationsdateien in `conf` und andererseits der Quelltext für die Firmware unter `sw/airborne`. Idealerweise sollte der Coder für die Firmware also für mehrere verschiedene Konfigurationen funktionieren, so dass z.B. statt einem Quadrocopter auch ein Flugzeug eingesetzt werden kann.

### Airframe

Im Airframe sind alle Spezifika des Fluggerätes zusammengefasst – also alles von der Drehrichtung der Motoren bis zur Adressierung einzelner Sensoren. Änderungen müssen hier eigentlich nur gemacht werden, sofern Hardware neu angebunden wird oder die Hardware selbst verändert wird. Unter anderem sind hier auch die PID-Werte für die Lageregelung festgelegt, die angepasst werden sollten wenn sich das Gewicht des Quadrocopters ändert. Unser Airframefile befindet sich unter `conf/airframes/ovgu/quadrotor_lia_ovgu.xml`.

### Flightplan

Im Flightplan(`conf/flight_plans/ovgu.xml`) wird quasi eine Mission für den Quadrocopter festgelegt. Paparazzi sieht dafür im wesentlichen Wegpunktnavigation vor, aber unter anderem können hier "Exceptions" definiert werden, so dass der Quadrocopter vor einer Wand ausweicht, falls er im normalen Flugmodus sonst dagegenstoßen würde. Momentan wird die eigentliche Steuerung von einer Funktion im Sonartreiber ausgeführt, die über den Flightplan aufgerufen wird.

## 2.3 Modell

Um Strukturierter Arbeiten zu können, folgt die Software dem folgenden Modell der Copter.

---

<sup>3</sup>Da mit git mehrere Features parallel entwickelt werden können, muss u.U. noch ein Branch ausgecheckt werden

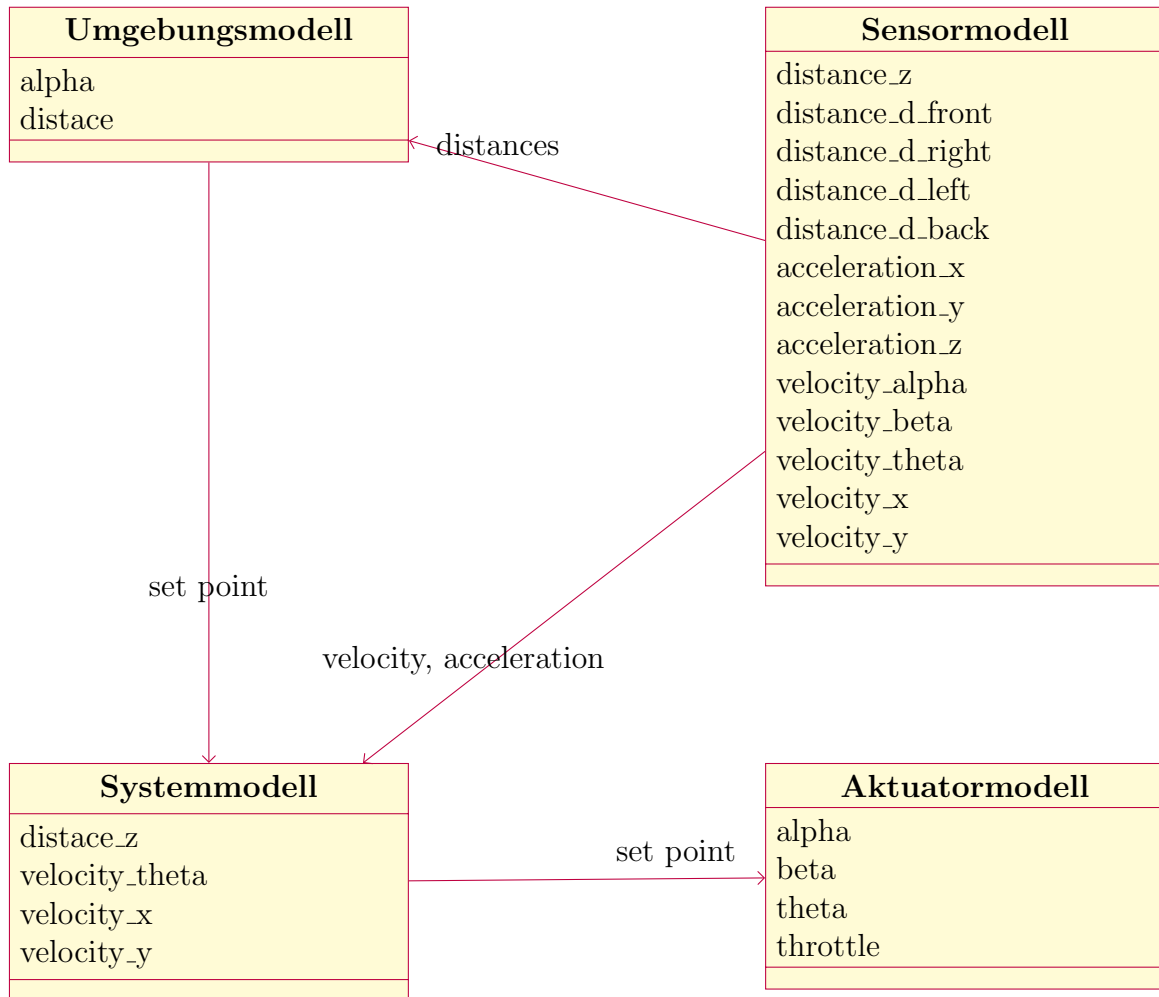


Abbildung 2: Modelle in UML



### 2.3.1 Sensormodell

Diverse Variablen lassen sich durch Sensorik und Virtuelle Sensoren (also z.B. Sensorfusion oder Filter) beobachten. Dazu gehören: Geschwindigkeiten und Beschleunigungen in allen Achsen außer der Höhe, die Absolute Höhe, die Messungen der Sonarsensoren und noch einige weitere Werte.

### 2.3.2 Aktuatoren

Das Aktuatormodell ist beschränkt auf die Angabe einer Fluglage, die durch die Lageregelung des Quadrocopters umgesetzt werden soll. Dazu gehören Pitch- und Roll-Winkel, die Ableitung des Yaw-Winkels und der Schub.

### 2.3.3 Systemmodell

Der Zustand des Quadrocopters lässt sich beschreiben aus  $\dot{x}$ ,  $\dot{y}$ , dem Yaw-Winkel( $\phi$ ), dem Aktuellen Schub ( $t$ ) und der aktuellen Höhe. Der Zusammenhang zwischen Aktuatoren und  $\dot{x}$  bzw.  $\dot{y}$  kann dabei ungefähr beschrieben werden wie folgt:

$$\dot{x} = k_t \cdot t \sin \alpha$$

$$\dot{y} = k_t \cdot t \sin \beta$$

$k_t$  ist dabei eine Konstante die sich aus der Masse des Quadrocopters und dem Verhältnis von vorgegebenen Schub und der tatsächlich erzeugten Kraft ergibt.

Die Werte für  $\dot{x}$  und  $\dot{y}$  werden dabei mittels eines Kalman-Filters berechnet, der die Sensorwerte für die Beschleunigungen mit einbezieht. Die kinematischen Gleichungen geben dabei den folgenden Übergang vor:

$$\begin{pmatrix} y_{t+1} \\ \dot{y}_{t+1} \end{pmatrix} = \begin{pmatrix} y_t & \dot{y}_t \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \quad (1)$$

### 2.3.4 Umgebungsmodell

Die Umwelt des Quadrocopters soll zu nächst durch ein sehr einfaches Modell beschrieben werden: Durch den Abstand und Winkel des nächstgelegenen Objektes.