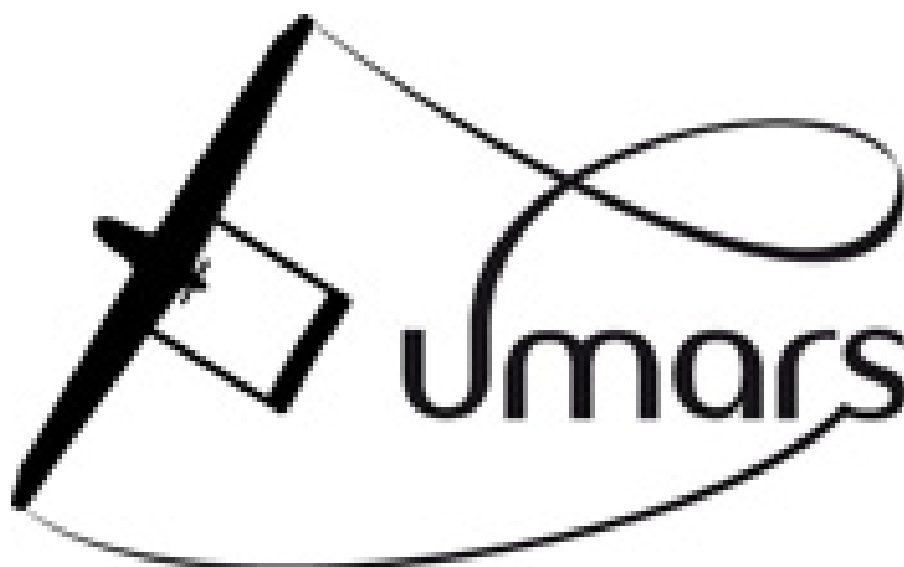


Vertiefung:
SYAT2

Zürcher Hochschule
für Angewandte Wissenschaften



Flugregelung einer Kleindrohne für Forschungszwecke



Autor: Schmidhauser Emilio, MI07a
schmiemi@students.zhaw.ch

Chaney René, MI07a
chaneren@students.zhaw.ch

Datum: 20.09.2010

Dozent: Siegl Walter
siew@zhaw.ch

Projektleitung Umars: Ensslin Oliver
enso@zhaw.ch

Abstract

Diese Projektarbeit wurde im Vertiefungsfach SYAT im letzten Semester an der ZHAW erstellt. Die Aufgabe war eine Marktrecherche, das Prüfen der Verfügbarkeit und der Einbau eines Autopiloten in eine Drohne, die zu Forschungszwecken entwickelt wurde. Die Drohne soll nach dem Programmieren einer Flugroute in der Lage sein, diese möglichst genau abzufliegen. Dabei muss sie aber sicher betrieben werden können. Das heisst es muss jederzeit von einem Operator die Kontrolle übernommen werden können. Es gibt viele kommerzielle Autopiloten auf dem Markt, die diese Aufgabe erfüllen können. Es gibt aber auch diverse Opensource Projekte, die sehr viel günstiger sind. Die Preise für die kommerziellen Produkte sind einiges höher als die Opensource Projekte. Im Laufe dieser Projektarbeit wurde der Autopilot Paparazzi, einer der ältesten Autopilot-Opensource-Projekte, gewählt, da er den kommerziellen Produkten bezüglich der Funktionen in fast nichts nachsteht. Ein weiterer wichtiger Aspekt ist, dass der Code zugänglich ist. So kann in weiterführenden Arbeiten, wie zu Beispiel die Anbindung eines IMU, ohne zusätzliche Kosten die Software verändert werden.

Das System ist sehr komplex, was eine grössere Einarbeitungszeit mit sich bringt. Der Aufbau und die Inbetriebnahme des Autopiloten gestalteten sich schwieriger als angenommen, da Kabel und einzelne Elemente falsch konfiguriert geliefert wurden und sehr viel Detailarbeit gemacht werden musste. Ein weiteres Ziel dieser Projektarbeit war es, alle Funktionen der Soft- und Hardware zu dokumentieren, da die Dokumentation im Internet etwas mangelhaft ist und ansonsten keine mitgeliefert wurde.

Bei Abgabe der PA konnte das angestrebte Ziel, ein Testflugzeug namens MAJA mit dem Autopiloten zu fliegen, nicht erreicht werden. Das System ist aber beinahe fertig aufgebaut und betriebsbereit. Alle wichtigen Konfigurationen und Verbindungen sind erstellt und könnten mit wenig Arbeit im Flugzeug verbaut werden.

Inhaltsverzeichnis

1	Aufgabenstellung	5
1.1	Ausgangslage	5
1.2	Die Drohne	6
1.3	Anforderungen an den Autopiloten	6
2	Auswahl eines Autopiloten	7
2.1	Opensource Autopiloten	7
2.2	Kommerzielle Autopiloten	8
3	Einführung in das Paparazzi System	9
3.1	Warum Paparazzi?	9
3.2	Paparazzi im Überblick	9
3.3	Systemüberblick.....	10
3.4	Blockschaltbild	11
4	Tiny V2.11	12
4.1	Schematische Darstellung des Controlboards	12
4.2	Anschlüsse am Control Board	13
4.3	Die Komponenten	14
5	Die Lagesensoren	14
5.1	IR-Sensoren.....	15
5.2	Gyro	15
5.3	Beschleunigungssensoren.....	15
5.4	IMU (Inertial Measurement Unit).....	16
6	Schnittstellen.....	17
6.1	2.4 GHz RC-Steuerung und Paparazzi verbinden	17
7	Vorbereitungen / Alle Elemente besorgen	20
7.1	Welche Paparazzi-Elemente werden benötigt?	20
7.2	Bestelliste.....	20
7.3	AWG 28...26.....	21
8	Software Installation.....	22
8.1	Was braucht es für die Software Installation?.....	22
8.2	Installation mit Live-CD	22
8.3	Installation von Grund auf	22
8.4	Updates.....	23
9	Zusätzliche Kabel und Verbindungen erstellen.....	24
9.1	Serielles Verbindungskabel	24
9.2	USB→Seriell Kabel.....	24
9.3	Anschluss der Funkmodem	25

9.4	Diverse Servokabel.....	26
9.5	Empfänger→Paparazzi.....	26
10	Paparazzi Inbetriebnahme.....	27
10.1	Flashen des Tiny	27
10.2	Datenverbindung Paparazzi →Tiny erstellen	28
10.3	Serielle Verbindung mit Kabel	28
10.4	Verbindung über X BEE Modem	29
10.5	GPS Konfiguration.....	32
10.6	Inbetriebnahme	32
11	GUI	33
11.1	Aufstarten	33
11.2	Menu	33
11.3	Bereiche	38
12	Operationsarten.....	39
12.1	Simulation.....	39
12.2	Replay	39
12.3	Flug	39
12.4	Hardware in the Loop HITL	39
13	Konfigurations-Dateien	40
13.1	Airframe	40
13.2	Flightplan.....	46
13.3	Settings	48
13.4	Radio.....	48
13.5	Telemetry	48
14	GCS.....	49
14.1	Menu-Leiste.....	50
14.2	Karte.....	51
14.3	Überwachungsfeld.....	51
14.4	Einstellungen.....	51
14.5	Meldeanzeige	52
15	Anhang	53
15.1	Abbildungsverzeichnis.....	53
15.2	Tabellenverzeichnis.....	54

1 Aufgabenstellung

1.1 Ausgangslage

Der Markt an Kleindrohnen bis 30 Kilogramm Eigengewicht ist sehr überschaubar. Der Einsatz solcher Drohnen ist in der Schweiz zurzeit ohne Bewilligung möglich. Geringe Betriebskosten gegenüber bemannten Missionen machen ein solches Flugzeug für meteorologische sowie geodätische (Geodäsie: Wissenschaft der Ausmessung der Erdoberfläche) Einsätze interessant. Der Vorteil einer solchen Drohne liegt in der Fähigkeit, in Bereichen zu fliegen, die für einen bemannten Flug zu gefährlich sind, wie z.B. in niedriger Höhe oder über unlandbarem Gebiet. Die Verwendung ist aber nicht auf diese Bereiche festgelegt. Ein Einsatz als Versuchsträger für flugmechanische Eigenschaften, in der Luftchemie oder als Impfdrohne gegen Hagel ist durchaus möglich. Die Auslegung in der Modellflug-Gewichtsklasse erlaubt eine vielfältige Verwendung. Es kann 7 kg Nutzlast geladen werden. Verschiedene Öffnungen im Rumpf und Behälter an den Flügelenden machen das Flugzeug vielseitig.¹

Für die Weiterentwicklung der Drohne, die während der Bachelorarbeit an der Zhaw von Oliver Ensslin und Lukas Staub entwickelt worden ist, soll nun ein Autopilot implementiert werden. Dieser Autopilot soll es jedermann ermöglichen, ohne grosse Ausbildung die Drohne zu verwenden. Das Flugzeug soll autonom eine festgelegte Strecke abfliegen können (Positionierung durch GPS) und jederzeit durch einen gewöhnlichen Handsender wie er im Modellbau üblich ist übernommen werden können. In weiteren Schritten soll das Flugzeug auch autonom starten und landen können. Für diese Anforderungen sind auf dem Markt diverse Autopiloten vorhanden. Es soll nun abgeklärt werden ob ein solcher in Frage käme, oder ob gewisse Elemente oder sogar alles selber ausgelegt werden muss.

¹ Quelle: Bachelorarbeit von Oliver Ensslin, Lukas Staub, Erstellt am 7.8.2009

1.2 Die Drohne

Die Drohne ist mit einer Spannweite von ca. 5 m ziemlich imposant. Sie ist aerodynamisch so ausgelegt, dass sie bis 5000 müM operieren kann. Sie kann abhängig von Höhe und Geschwindigkeit bis 7 kg Nutzlast mit sich führen. Angetrieben wird sie von einem 8.5 PS starken Benzinmotor der sie auf einen Cruise Speed von ca. 70 km/h beschleunigt. Die Landegeschwindigkeit liegt bei ca 50 km/h. Die 7 kg Nutzlast können im Rumpf (Flugzeugspitze) mitgeführt werden. Um die Sicht für Kameras oder Messsonden nach vorne frei zu haben, ist der Antrieb am Heck angebracht. Im Moment ist die Drohne mit einem Fahrwerk ausgestattet, das sich nicht einfahren lässt und darum einen ziemlich grossen Luftwiderstand generiert. Dieses Fahrwerk wird voraussichtlich noch demontiert, wobei aber bis zum jetzigen Punkt noch nicht klar ist wie das Flugzeug landen wird. Für Transportzwecke können die Flügel demontiert werden.

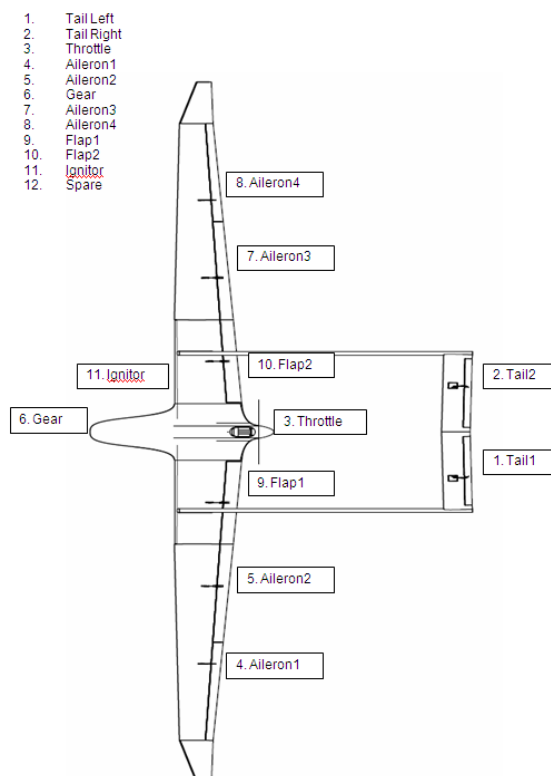


Abbildung 1-1: Umars

Für weitere Informationen verweisen wir Sie hier auf www.umars.ch.

1.3 Anforderungen an den Autopiloten

Die genauen Anforderungen sind in der Anforderungsliste aufgeführt (siehe Anhang, Anforderungsliste). Um aber einen groben Überblick zu geben sollen hier die wichtigsten Funktionen kurz zusammengefasst werden. Die Drohne soll auch für wenig versierte Piloten gut handhabbar sein, so dass sie mit möglichst wenig Schulung geflogen werden kann. Der Autopilot soll so programmiert werden können, dass er eine vorgegebene Route selbständig abfliegen kann. Dabei soll die Drohne jederzeit von einer gewöhnlichen RC-Handsteuerung übernommen werden können. In weiteren Schritten soll sie selbständig landen und starten können, so dass während einer Mission nicht eingegriffen werden muss.

2 Auswahl eines Autopiloten

Es sind viele verschiedene Autopiloten auf dem Markt verfügbar die verschiedenste Ansprüche befriedigen. Zum einen sind einige Open-Source-Produkte verfügbar die entweder selber zusammengebaut werden können, oder über einen Hersteller bezogen werden können. Zum anderen gibt es sehr viele kommerzielle Produkte, die sich in unterschiedlichen Preislagen bewegen. In diesem Abschnitt sollen nun einige Autopiloten vorgestellt werden, um einen groben Überblick zu geben. Es sind hier bei weitem nicht alle Autopiloten dabei, aber die grossen, oder die nach unserer Meinung erwähnenswerten, werden hier kurz vorgestellt. Für weitere Informationen sind die entsprechenden Internetseiten angegeben.

2.1 Opensource Autopiloten

2.1.1 Ardupilot

Der Ardupilot basiert auf der Arduino-Plattform, ist sehr preisgünstig und sehr gut dokumentiert. Er ist eher für Modellbauer und Einsteiger ausgelegt, besitzt aber schon die Möglichkeit IMU's für die Lageerkennung anzuschliessen.



<http://diydrone.com/profiles/blogs/ardupilot-main-page>

2.1.2 Openpilot

Der Openpilot ist ein neues Projekt, das grossen Wert auf die einfache Verwendung legt. Es ist ein sehr vielversprechendes Projekt, das weiterverfolgt werden sollte. Zum Zeitpunkt der Entscheidung und Bestellung eines Autopiloten war die Hardware aber noch nicht verfügbar und schied somit aus.



http://wiki.openpilot.org/Main_Page

2.1.3 Paparazzi

Der Paparazzi Autopilot ist eines der ältesten Opensource Autopilot Projekten. Es wird seit ca. 7 Jahren daran gearbeitet und verbessert. Hard- und Software sind Opensource und können im der Paparazzi-Wiki heruntergeladen werden. Die grosse Paparazzi-Community treibt das Projekt ständig vorwärts und verbessert das System.



www.paparazzi.enac.fr

2.2 Kommerzielle Autopiloten

2.2.1 Micropilot

Micropilot, der selbsternannte Weltmarktführer in Miniatur UAV Autopiloten ist ein Kanadischer Hersteller der seit 1995 Autopiloten baut.



<http://www.micropilot.com/>

2.2.2 Kerstel Autopilot

Ein amerikanischer Hersteller, der ein sehr robustes und leistungsstarkes System herstellt.



<http://www.procerusuav.com/>

2.2.3 Piccolo

Dies dürfte der Rolls Royce unter den für Privatpersonen kommerziell erhältlichen Autopiloten sein. Ein sehr leistungsfähiges, professionelles System, das keine Wünsche offen lässt. Preislich allerdings spielt dieses Produkt auch in einer eigenen Liga.



<http://www.cloudcaptech.com/>

2.2.4 Diverse gute Links und andere Autopiloten

Vielversprechender Autopilot aber noch in den Kinderschuhen: www.gluonpilot.com

Gute Informationsseite: <http://diydrone.com/>

Kommerzieller Aupilot <http://www.attopilotinternational.com/>

3 Einführung in das Paparazzi System

3.1 Warum Paparazzi?

Es gibt diverse Autopiloten auf dem Markt, die schon sehr weit ausgereift sind. Ein solches System selber zu konzipieren und zu bauen würde sehr viel Zeit in Anspruch nehmen und würde in absehbarer Zeit nicht zum Erfolg führen. Wir entschieden uns daher ein fertiges System zu übernehmen und allenfalls an unsere Bedürfnisse anzupassen. Unsere Wahl ist auf Paparazzi gefallen, weil schon über 7 Jahren Erfahrung und Entwicklungsarbeit von verschiedenen Universitäten und Nutzern darin stecken. Ausserdem ist es komplett Opensource. Das heisst die Hard- und Software sind frei zugänglich und lassen eventuelle Weiterentwicklungen zu. Verglichen mit den anderen Opensource-Produkten ist der Paparazzi-Autopilot sehr weit entwickelt und kostet nur ein Bruchteil eines kommerziellen Systems.

3.2 Paparazzi im Überblick

Paparazzi beinhaltet ein komplettes System für den autonomen Flug. Darin enthalten sind die in der Luft benötigten Komponenten(exklusive RC Zubehör), sowie die Bodenstation, die mittels einer seriellen Verbindung mit der Drohe verbunden ist. Die Bodenstation wird verwendet um die Flugroute zu planen, aber auch für die Echtzeit-Verfolgung der Drohne in der Luft. Im Flugzeug müssen folgende Komponenten platziert werden:

- Autopilot Control-Board
- Batterie für die Versorgung des Autopiloten
- Datalink (Modem und Antenne)
- GPS Empfänger
- IR-Sensoren
- Motor mit Controller
- RC Empfänger mit Antenne
- Aktuatoren (Servos)

Der Autopilot muss genau wissen, wie das Flugzeug in der Luft liegt. Die auf der Homepage aufgezeigte Variante besteht aus vier bis sechs Infrarotsensoren, die den warmen Boden sowie die kalte Luft erkennen können. Der Vorteil dieser Variante ist das geringe Gewicht, was den Einsatz des Autopiloten vor allem für sehr kleine Flugmodelle (bis ca. 1 kg) möglich macht. Eine genaue Abfrage des Pitch, Roll sowie Heading sind mit dem IR-Sensor aber nicht möglich. Paparazzi bietet aber die Möglichkeit, viele beliebige Standardsensoren (z.B. für die Lageerkennung in Modellhubschraubern (Gyro)) zu verwenden, somit auch solche, welche die oben genannten Parameter herausgeben. Über ein Radio-Modem werden die

aktuellen Daten an die Bodenstation gesendet. Auch Videosignale und andere Parameter können übermittelt werden. Paparazzi ist OpenSource, und läuft unter Linux.²

3.3 Systemüberblick

Man benötigt ein flugfähiges Modellflugzeug mit allen Komponenten wie Servos, Empfänger, Regler und Motor. Das Paparazzi-System wird wie in übergeordnet dazwischen geschaltet. Die Fernsteuerung wird nur noch als Safetylink verwendet. Bei Problemen während des autonomen Fluges kann über einen Schalter an der Fernbedienung die Kontrolle des Flugzeuges wieder übernommen werden. Vor dem Start



Abbildung 3-1: Überblick Paparazzi System

werden alle Flugrelevanten Informationen sowie die Flugroute in den Flashspeicher des Autopiloten geladen. Das Flugzeug ist völlig autonom und kann ohne Eingriff von aussen den programmierten Weg abfliegen. Die Drohne regelt sich bei z.B. Seitenwind selbständig aus und steuert das Flugzeug wieder auf die angepeilten Koordinaten. Der Datalink zwischen GroundStation wird eigentlich nur für den Empfang von Telemetriedaten und für Änderungen an der Flugroute während des Fluges gebraucht.

- **A**utopilot Control Board
- **B**attery
- **D**atalink Radio-Modem & Antenna
- **G**PS Receiver
- **I**R Sensor Board
- **M**otor & Controller
- **R**C Receiver & Antenna
- **S**ervos
- **P**ayload = Camera & Video Transmitter

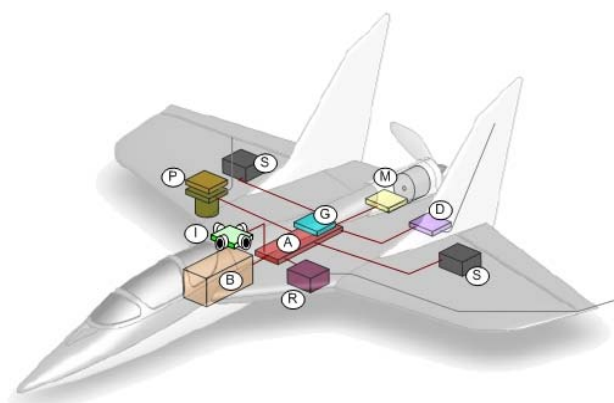


Abbildung 3-2: Komponente des Paparazzi Systems

² Quelle: Bachelorarbeit von Oliver Ensslin, Lukas Staub, Erstellt am 7.8.2009

3.4 Blockschaftbild

Dieses Blockschaftbild verdeutlicht wie das System im bestehenden Modellflugzeug an die RC-Elektronik angeschlossen wird. Die untere Hälfte des Bildes zeigt die konventionelle Modellbausteuerung. Das Paparazzi-System kann einfach zwischen den Empfänger und die Servos bzw. das Servointerface angeschlossen werden. Die grösste Hürde beim Anschliessen der Elemente ist, dass das Signal aus dem Empfänger das Summensignal mit den Informationen für alle Sevos sein muss, sprich nicht schon aufgetrennt auf die verschiedenen Kanäle. Die Stelle von der dieses Signal abgenommen werden kann ist von Empfänger zu Empfänger anders und muss in den Schaltplänen ermittelt werden. Mehr Dazu in Kapitel:6.

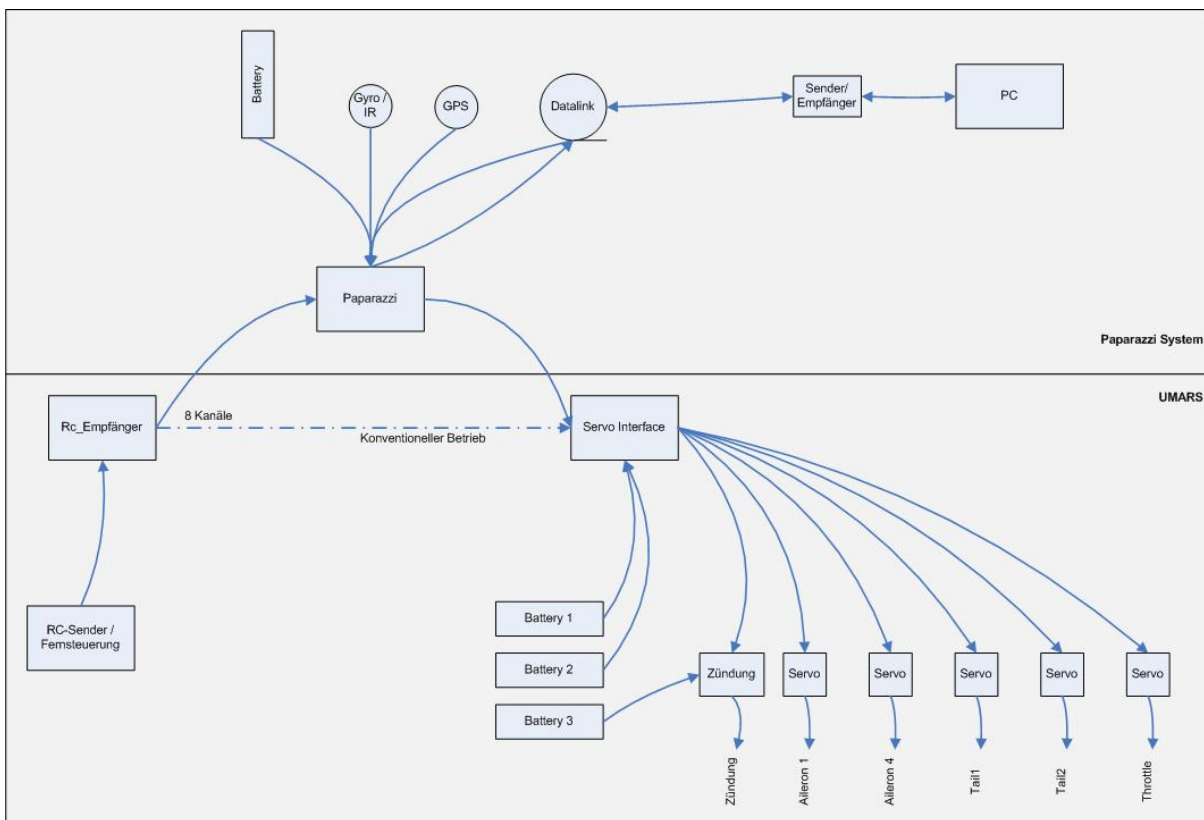


Abbildung 3-3: Blockschaftbild Paparazzi

4 Tiny V2.11

Dieses fertige Bundle kann auf der Seite <https://www.ppz UAV.com> bestellt werden. Wir entschieden uns ein Bundle zu kaufen damit wir möglichst schnell erste Erfahrungen mit dem System sammeln können. Dies ersparte uns langwieriges und fehlerbehaftetes Bestellen und Ausprobieren der Komponenten ausserdem konnten wir uns möglichst schnell mit der Software und den Schnittstellen befassen. Zudem sind die einzelnen Komponenten getestet und aufeinander abgestimmt und sollten funktionieren.

4.1 Schematische Darstellung des Controlboards

Hier ein kleiner Überblick über den Autopiloten und dessen I/O Steckplätze.

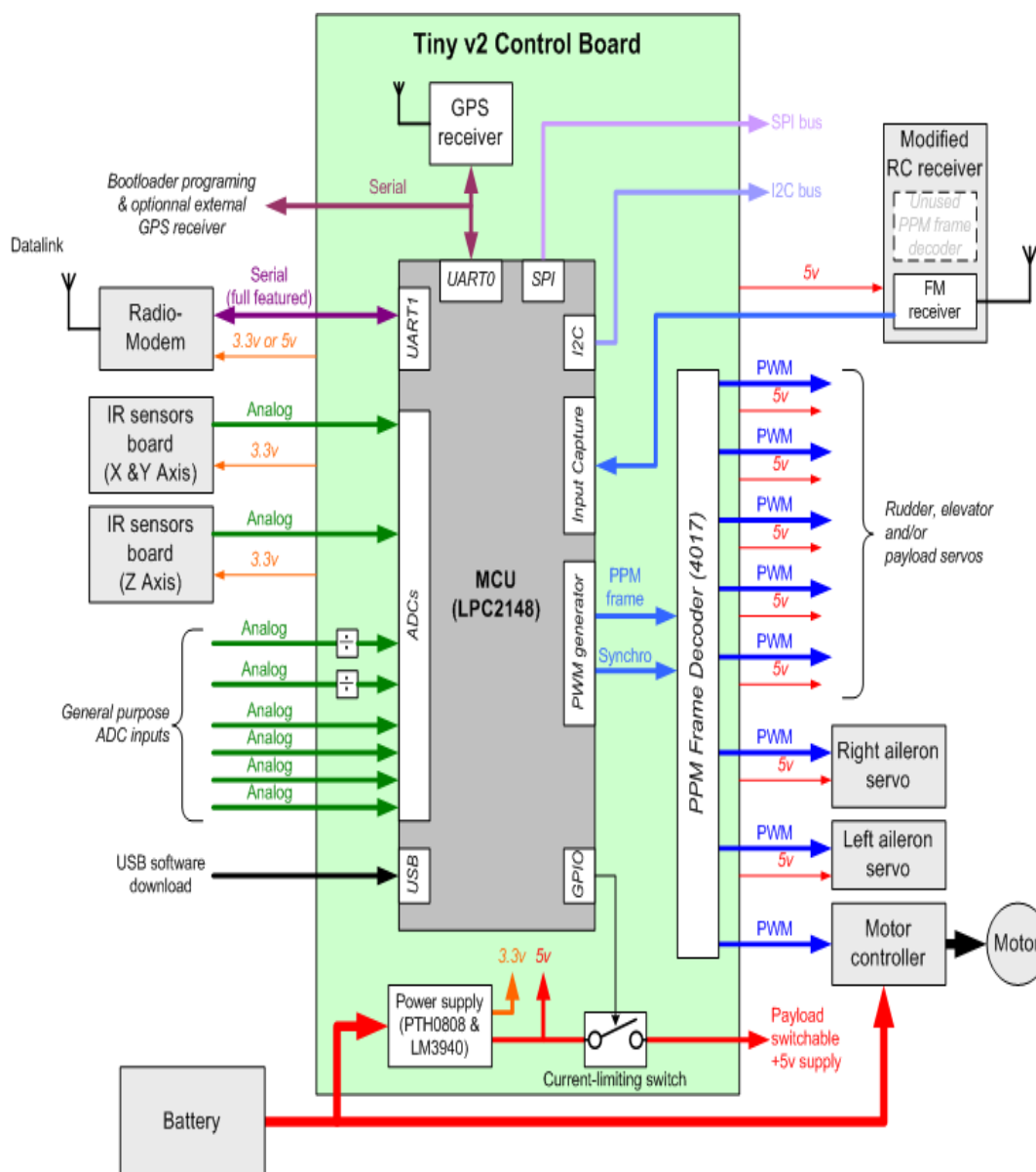


Abbildung 4-1: Schema Tiny V2.11

4.2 Anschlüsse am Control Board

Auf der folgenden Grafik sind die Anschlussmöglichkeiten am Autopiloten dargestellt.

IRV	Infrarot vertikal
IRH	Infrarot horizontal
ADC2	Analog-Digitalwandler 2
ADC1	Analog-Digitalwandler 1
DownLoad	xxxxx
CAM	Anschluss für eine Camera die das Bild auf den Monitor bringen kann
I ² C	I ² C Schnittstelle
USB	USB-Anschluss um Modem zu flashen
Serial	Serielle Schnittstelle (Anschluss des Modems)
SPI	SPI Schnittstelle
PPM	Eingang für Summensignal
SRVX	Steckplätze für die Servos

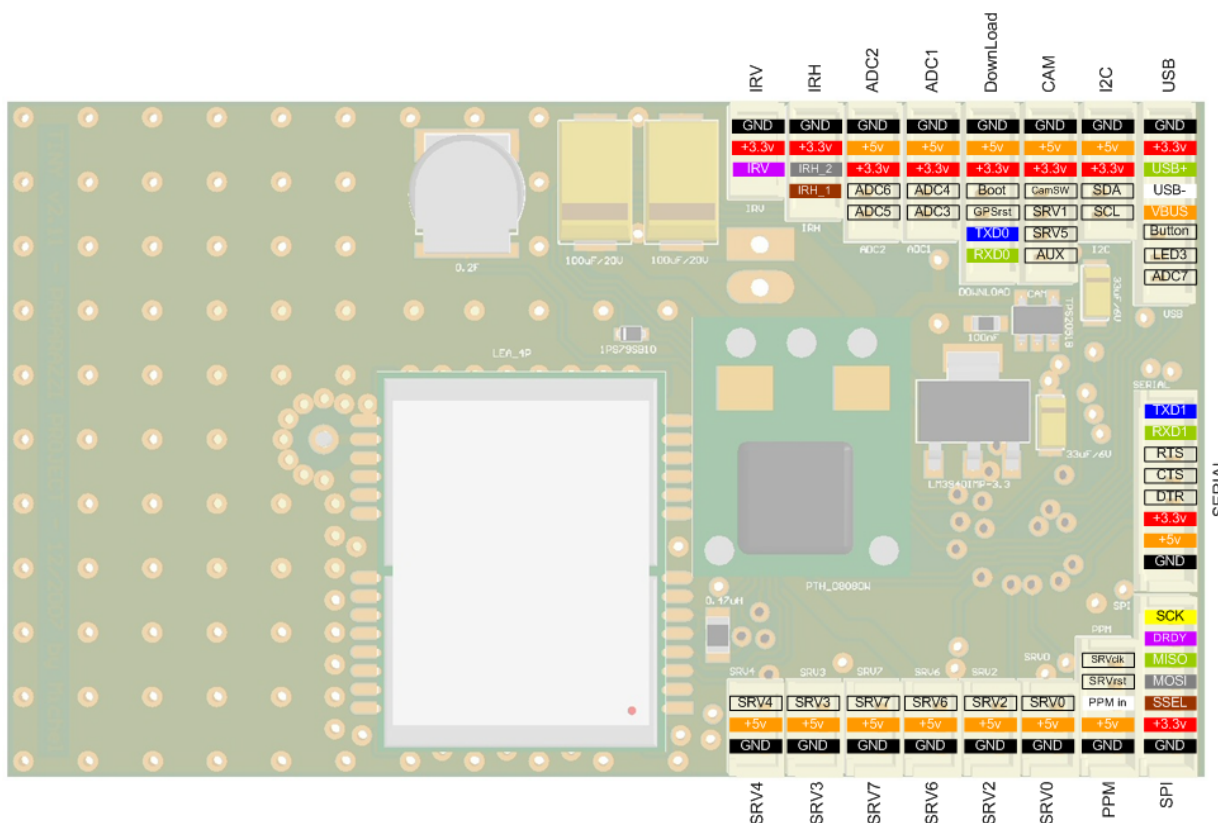


Abbildung 4-2: Ein- und Ausgänge des Tiny V2.11

4.3 Die Komponenten

Hier sehen Sie alle Elemente des Systems. Die beiden Servos und der Empfänger sind im Tiny v2.11 Bundle nicht dabei.

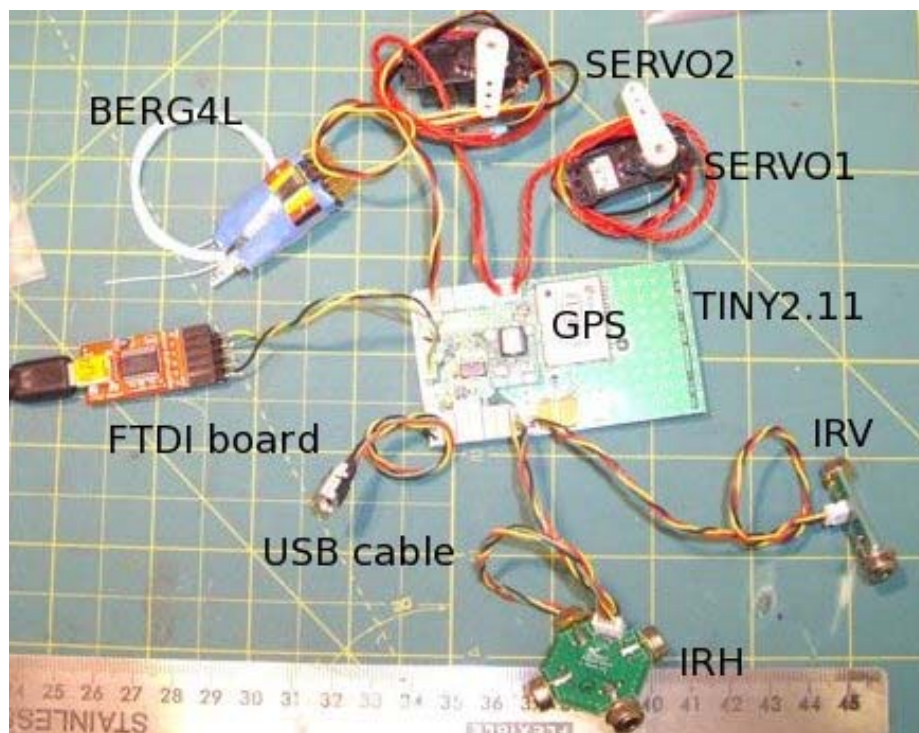


Abbildung 4-3: Komponente des Tiny V2.11

5 Die Lagesensoren

UAV's werden entweder mit einem Kreisel, Pyrometer oder wie im Fall des Paparazzi Autopiloten mit Infrarot Sensoren ausgestattet. IMU Gyro-Signale werden mit einem Kalmanfilter für Flugwerk Vibrationen geglättet, damit bei jedem Wetter geflogen werden kann. Infrarot basierte Steuerungstechnik wird nicht durch Vibrationen beeinträchtigt, aber bei bestimmten Wetterbedingungen oder Bergketten können die Messungen beeinflusst werden. Nach Abklärungen mit dem Hersteller des Autopiloten wurden diese Ängste aber relativiert. Für die ersten Versuche mit dem System sollte der IR-Sensor ausreichen, später während weiteren Arbeiten soll aber ein IMU verbaut werden.

Weiter führende Informationen zur Lageerkennung finden sie im Anhang Unter Kapitel 1 Einleitung der Diplomarbeit³ von Andreas Dei.

³ Andreas Dei; Entwicklung und Erprobung eines Autopiloten mit inertialer Messeinheit; 8. April 2009

5.1 IR-Sensoren

Die Infrarotsensoren, die zum Standard-Paparazzi-System mitgeliefert werden, arbeiten nach dem Prinzip, dass der Himmel kälter ist als der Boden. Dieses System soll aber in einem weiteren Schritt durch ein IMU ersetzt werden, da Wetterlagen möglich sind bei denen dieses System mit IR Sensoren nicht mehr funktionieren kann.

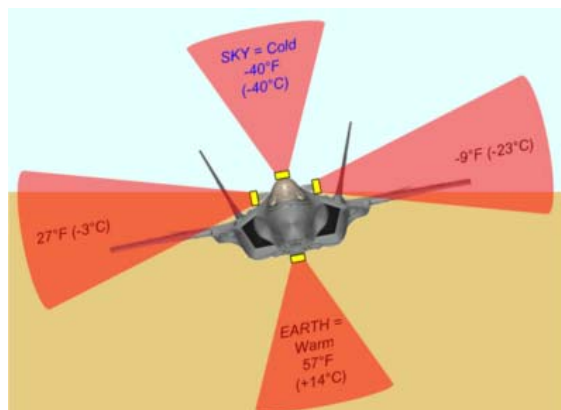


Abbildung 5-1: Funktionsweise der IR-Sensoren

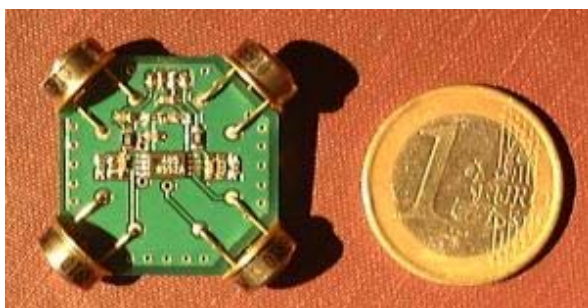


Abbildung 5-2: Horizontaler IR-Sensor

Im Bundle sind zwei Infrarotbauteile vorhanden. Der Links im Bild abgebildete Sensor ist zuständig für die Erfassung der horizontalen Lage, das heißt er informiert den Autopiloten ob das Flugzeug gerade steigt oder sinkt respektive links oder rechts fliegt. Der Vertikal-IR-Sensor gibt dem Autopiloten Aufschluss ob das Flugzeug gerade auf dem Rücken fliegt oder nicht.

5.2 Gyro

Ein Gyro misst die Winkelgeschwindigkeit um eine bestimmte Drehachse. Der Winkel kann durch eine Integration des Signals über der Zeit angenähert werden, dies birgt aber immer einen Fehler. Jeder

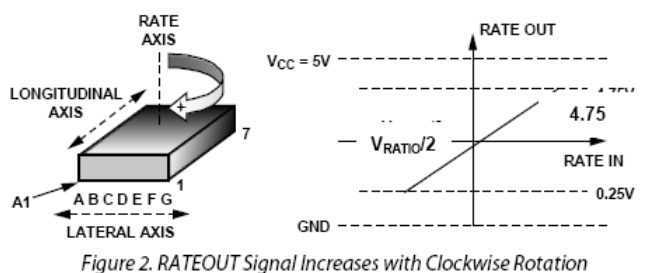


Abbildung 5-3: Funktionsprinzip Gyro

Gyro hat aber einen sogenannten Drift, das bedeutet, dass ein zusätzlicher Fehler aufsummiert wird. Aufgrund dieses Signals alleine kann die Fluglage für ein UAV nicht sicher bestimmt werden.

5.3 Beschleunigungssensoren

Beschleunigungssensoren messen die Beschleunigung in eine bestimmte Richtung. Ein 3-Achsen Beschleunigungssensor gibt Auskunft über die Orientation einer Plattform bezüglich der Erde. Wenn das Flugzeug aber eine Kurve fliegt gibt der Sensor die Information

Beschleunigung gegen Flugzeugboden aus (Steigflug). Dies macht die Beschleunigungssensoren alleine unbrauchbar für diese Anwendung.

5.4 IMU (Inertial Measurement Unit)

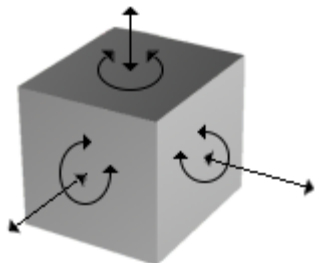


Abbildung 5-4: 6 DOF

Gyro und Beschleunigungssensoren müssen zusammen verwendet werden, da die 6 Freiheitsgrade abgedeckt werden müssen. Gyros messen bei kurzen Impulsen gut, neigen aber zum Driften bei langen Impulsen. Wobei Beschleunigungssensoren bei langen Beschleunigungen richtig messen bei kurzen Impulsen aber zu Störungen neigen. Diese beiden Messmethoden decken einander nur teilweise ab. Die Gierbewegung (Yaw) kann so noch nicht überprüft werden und bedarf noch eines weiteren Messmittels. Hierfür kommen GPS oder Magnetometer zum Einsatz. Ein IMU kombiniert die Werte dieser verschiedenen Messeinrichtungen und kann so Aussagen über alle 6 Freiheitsgrade im Raum geben.

6 Schnittstellen

6.1 2.4 GHz RC-Steuerung und Paparazzi verbinden

Eine wichtige Schnittstelle ist zwischen RC-Empfänger und dem Paparazzi-Controlboard. Um die Drohne im Notfall zu übernehmen muss gewährleistet sein, dass die RC-Steuerung die Kontrolle über das Flugzeug jederzeit übernehmen kann. Über einen Schalter an der RC-Fernsteuerung kann dem Paparazzi-Controlboard mitgeteilt werden welches Signal an die Servos weitergeleitet werden soll. Dem Paparazzi Autopiloten muss das Summensignal weitergeleitet werden. Alle üblichen RC-Empfänger splitten

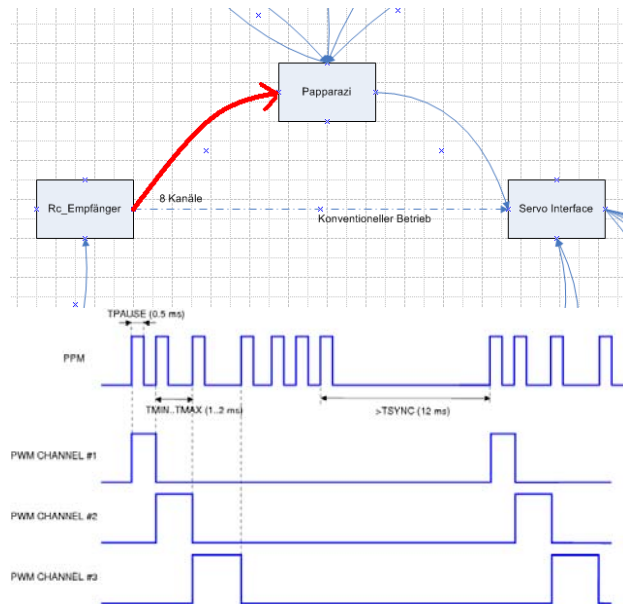


Abbildung 6-1: Schnittstellen Signal

aber das Empfänger-PPM-Signal in die PWM-Signale für die einzelnen Kanäle auf und man muss das Signal intern abziehen. Bei den neuen 2.4 GHz Fernsteuerungen geht das aber nicht mehr so einfach, da das Signal intern anders erstellt wird. Aus Sicherheitsgründen haben wir uns für eine 2.4 GHz Steuerung entschieden. Sie haben den Vorteil, dass sie nicht einfach mit Hilfe der richtigen Frequenz durch eine Drittperson übernommen werden können. Die 2.4GHz Steuerungen und Empfänger erkennen sich gegenseitig durch den GUID (Globally Unique Identifier). Diese Bindung ergibt gegenüber herkömmlichen Systemen einen grossen Sicherheitsvorsprung, der bei einem solchen Flugzeug sicher nicht vernachlässigt werden darf. Es gibt mehrere Möglichkeiten um diese Verbindung vorzunehmen und es sollen hier die wichtigsten vorgestellt werden.

6.1.1 Futaba FASST 7-Kanal Empfänger

Eine Möglichkeit ist das Signal im Empfänger abzunehmen. Dies ist aber nur bei wenigen Modellen möglich die auf dem Markt erhältlich sind, ausserdem kann man meist nicht alle Kanäle abnehmen. Beispielsweise beim Futaba FASST 7-Kanal Empfänger, der auf der Paparazzi-Seite erwähnt wird, können nur 5 Kanäle abgezogen werden. Mit den restlichen



beiden Kanälen könne nur direkt Servos angesprochen werden. Dies ermöglicht zwar ein Flugzeug zu fliegen, ist aber gerade bei der Umars Drohne wohl etwas knapp, da für ein solches Projekt keine halben Sachen gemacht werden sollten. Der unten angefügte Link kann weitere Informationen vermitteln.

http://paparazzi.enac.fr/wiki/Other_Hardware#Futaba_FASST_7-channel_receiver

6.1.2 Summensignalempfänger für 2.4Ghz

Eine weitere interessante Möglichkeit ist ein sogenannter Summensignalempfänger. Er decodiert das Signal gar nicht in seine Einzelkomponenten, sondern gibt es vollständig aus. Dies hat den Vorteil dass keine Zwischenelemente benötigt werden und der Empfänger direkt an den Autopiloten angeschlossen werden kann. Diese Systeme sind durch den wachsenden



Mikrokopter/Quatrokopter- Markt immer populärer geworden. Seit 2009 ist seitens Futaba der Summsignal-Empfänger R6007SP ohne Umbaunotwendigkeit erhältlich. Dieser hat eine relativ geringe Reichweite, das Modell R6107SP mit erhöhter Reichweite folgte 2010.

Allen Fernsteuersystemen ist gemeinsam, dass sie nur mit Marken-Eigenen Empfängern betrieben werden können, da sich die Übertragungsverfahren Herstellerspezifisch grundlegend unterscheiden. Dies schränkt aber die Flexibilität ein, da die einzelnen Komponenten aufeinander abgestimmt werden müssen. Ausserdem wäre für die Testphase des Autopiloten ein System, das aus mehreren Einzelkomponenten besteht besser geeignet, weil man allfällige Messungen oder Versuche einfacher machen könnte. Für eine grössere Serie sollte man aber diese Lösung sicher im Auge behalten.

6.1.3 Channel PPM Encoder

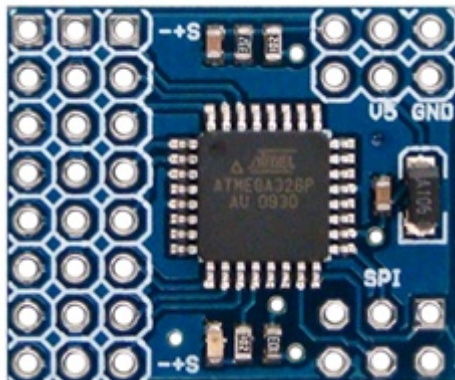


Abbildung 6-4: Encoder

Autopiloten halten, da die Quadrocopter für Failsafe-Situationen andere Anforderungen haben. Wir entschieden uns für den Encoder von dem wir auch das Paparazzi-Controboard bestellt hatten.

http://www.ppz UAV.com/osc/product_info.php?products_id=91

Es gibt aber auch die Möglichkeit eines Channel Encoders. Dieser macht nichts anderes als alle 8 Einzelsignale wieder zusammen zu mixen. Es werden alle 8 Ausgangskanäle des Empfängers in den Encoder geführt, der dann das Signal an einem seiner Ausgänge ausgibt. Es gibt dieses Element von verschiedensten Herstellern aus der Autopiloten- oder Quadrocopter Umgebung. Wer das Programm des Controlers aber nicht umschreiben will, sollte sich aber einfachheitshalber an die Hersteller von

7 Vorbereitungen / Alle Elemente besorgen

7.1 Welche Paparazzi-Elemente werden benötigt?

Wie schon erwähnt, ist in diesem Projekt ein fertiges Packet mit allen benötigten Hardwareelementen bestellt worden. Bei dem Bundle mit dem wir gearbeitet haben, handelt es sich um den Tiny V2.11 von PPZUAV. Darin sind folgende Bauteile enthalten:

- Tiny V2.11 Platine
- FTDI Board
- XBee Pro 2 Modem
- Diverse Stecker (PicoBlade, USB...)
- IR-Sensor (X-Y)
- IR-Sensor (Z)

7.2 Bestellliste

Um die nötigen Verbindungen zu erstellen braucht es die Folgenden Stecker und Bauteile. Diese Konfiguration wurde zu Testzwecke erstellt und würde für das Endprodukt noch angepasst werden.

Für Elektronikspezifischen Stecker entschieden wir uns alles bei Distrelec zu bestellen. Für die entsprechenden Stecker die eher modellbauspezifisch sind, haben wir bei Conrad alle Teile bestellt, da sie bei Distrelec nicht zu bekommen waren.

7.2.1 Distrelec

Pos	Überbegriff	Beschreibung	Anz.	Art. Nr.	Preis
1	PicoBlade	3 Pol	10	116876	1.- / Stück
2	PicoBlade	4 Pol	2	116877	1.- / Stück
3	PicoBlade	5 Pol	5	116878	1.- / Stück
4	PicoBlade	7 Pol	5	116880	1.- / Stück
5	PicoBlade	8 Pol	2	116881	1.- / Stück
6	PicoBlade	Crimps AWG 28...26	130	116905	0.5.- / Stück
7	Stift / Buchsenleiste	Leiste für Antenne	2	127220	1.- / Stück

Tabelle 7-1: Distrelec Bestellung

7.2.2 Conrad

Pos	Hersteller	Beschreibung	Anz	Bestellnummer	Preis
8	Futaba	Servostecker für Tiny - Servos (übergangsstück)	6	065604 - 62	5.50 / Paar
9	Modelcraft	Servosteckerkabel Encoder-Empfänger Weiblein	6	227278 - 62	3.65 / stk
10	Modelcraft	Servosteckerkabel Tiny - Servos Männchen	9	223726 - 62	3.95 / stk
11	Futaba	Servokabel Verlängerung	1	065597 - 62	5.50 / 2m

Tabelle 7-2: Conrad Bestellung

7.3 AWG 28...26

Die Picoblade Steckverbindungen, die auf dem Paparazzi-ControlBoard sind, benötigen bestimmte Drahtquerschnitte.

Wir entschieden uns für die grösstmöglichen Crimps die es für die Stecker gab. Also AWG 28. Die Crimpzange haben wir aus dem Elektrolabor bekommen: Molex Order Nr.: 638110200 für AWG 28-32 Male and Female. Für die Montage der Crimps auf die Stecker wird hier wärmstens eine entsprechende Crimpzange empfohlen, da die Verbindungen ansonsten nicht stabil werden oder sie Kurzschlüsse entstehen könnten, die während einem Flug zum Absturz führen könnten.

	Nr	Querschnitt [mm ²]	Leiter-Æ [mm]
AWG	25	0,163	0,455
AWG	26	0.128	0.404
AWG	27	0.128	0.404
AWG	28	0.102	0.364
AWG	29	0.0804	0.32
AWG	30	0.0507	0.254
AWG	31	0.0401	0.226
AWG	32	0.0324	0.203

Tabelle 7-3: Übersicht AWG Grössen

8 Software Installation

8.1 Was braucht es für die Software Installation?

Es braucht einen Computer mit einer Debian Distribution, am Besten einer aktuellen Ubuntu Version. Der Computer muss nicht der schnellste und modernste sein, aber es empfiehlt sich nicht gerade den ältesten zu nehmen. Ausserdem sollte er genügend Akkulaufzeit haben um bei den Flügen nicht in problematische Situationen zu kommen. Das System kann auch auf eine virtuelle Maschine wie VM-Ware oder der VirtualBox (Achtung USB wird benötigt) installiert werden. Der Computer sollte die folgenden minimalen Anforderungen erfüllen.

- Akkulaufzeit
- Optisches Laufwerk (CD/DVD)
- Mindestens 1 USB port
- Mindestens 256 Mb RAM
- Möglichkeit für Internetanschluss

Die benötigte Software kann auf zwei Arten Installiert werden. Linuxexperten werden vermutlich die Eigenhändige Installation wählen wobei aber Linuxanfänger eher die Variante mit der Live-CD wählen werden. Zum Schluss spielt es keine Rolle welche Variante man gewählt hat.

Auf der Internetseite: <http://paparazzi.enac.fr/wiki/Installation> sind jeweils die aktuellsten Informationen zur Installation notiert. Alle nötigen Dateien können auf der folgenden Internetseite heruntergeladen werden. <http://paparazzi.enac.fr/wiki/Downloads>

8.2 Installation mit Live-CD

Die einfachste Möglichkeit Paparazzi zu installieren ist mit Hilfe der Live-CD. Sie enthält alle Pakete die für den Betrieb von Paparazzi benötigt werden. Die CD erfährt nur selten ein Update, es sollte darum möglichst nach der Installation ein Update durchgeführt werden.

Die CD kann auf der Seite <http://paparazzi.enac.fr/wiki/Downloads> heruntergeladen werden. Dieses iso-File kann dann auf eine CD gebrannt werden und installiert werden.

8.3 Installation von Grund auf

Die Installation von Grund auf ist etwas komplizierter und dauert etwas länger. Auf folgendem Link ist eine sehr gute Beschreibung.

<http://www.rescuerobotics.com.au/getting-started-with-linux-and-the-paparazzi-ground-control-station>

8.4 Updates

Die neuste Version von Paparazzi kann vom Wiki mit dem Link <http://www.recherche.enac.fr/paparazzi/tarball/> heruntergeladen werden. Laden Sie die Datei einfach herunter und extrahieren sie in ein Verzeichnis ihrer Wahl. Ersetzen Sie nun den Paparazzi-Ordner im Home Verzeichnis durch den neuen, extrahierten Paparazzi3/ Ordner. Nun ist das System auf dem neusten Stand.

9 Zusätzliche Kabel und Verbindungen erstellen

In diesem Abschnitt wird beschrieben welche Kabel benötigt werden und wie man sie selber baut. Um diesen Abschnitt etwas einfacher zu halten, wird nur von z.B. 4 Polige Picoblade Stecker geschrieben. Um alle Verbindungen zu erstellen werden noch diverse Kabel benötigt. Die genauen Stecker mit Bestellnummern sind im Abschnitt 7 angegeben.

9.1 Serielles Verbindungskabel

Dieses Kabel wird für den Betrieb des Systems ohne die Funk-Verbindung benötigt. Natürlich kann der Tiny so nur für Testzwecke am Boden verwendet werden. Wer sich entscheidet das serielle Kabel gar nicht zu verwenden, muss dieses auch gar nicht unbedingt bauen.

Es werden folgende Komponenten benötigt:

1. 3 poliger PicoBladestecker (Weibchen)
2. 8 poliger PicoBladestecker (Weibchen)
3. Drei Drähte ($A=0.1\text{mm}^2$)

Es ist wichtig, dass die beiden Drähte TX und RX gekreuzt gesteckt werden. Ansonsten startet das Paparazzi GCS nicht richtig und es kann keine Verbindung hergestellt werden. Bildschirm im GCS bleibt schwarz.

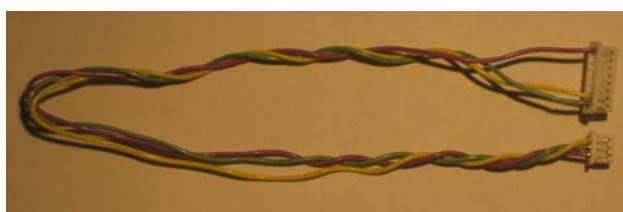


Abbildung 9-1: Verbindungskabel serielle Kommunikation

- RX FTDI → TX Paparazzi
- TX FTDI → RX Paparazzi

9.2 USB→Seriell Kabel

Dieses Kabel kommt im Tiny V2.11 Bundle mit. Es handelt sich hier aber um einen 8 poligen Picoblade und einen Mini USB Stecker. Dazu werden wieder 4 Drähte mit 0.1mm^2 benötigt.



Abbildung 9-2: USB to Picoblade Kabel

9.3 Anschluss der Funkmodem

Es werden zwei Sender/Empfänger benötigt. Einer bleibt am Boden am Computer der andere wird im Flugzeug befestigt. Beim Tiny V2.11 Bundle war es so, dass das Groundmodem nur auf der FTDI-Platte eingesteckt werden konnte.

9.3.1 Autopilot-Modem

Aus diesen Bildern und der Liste ist ersichtlich wie das Modem an den Autopiloten angeschlossen werden muss.

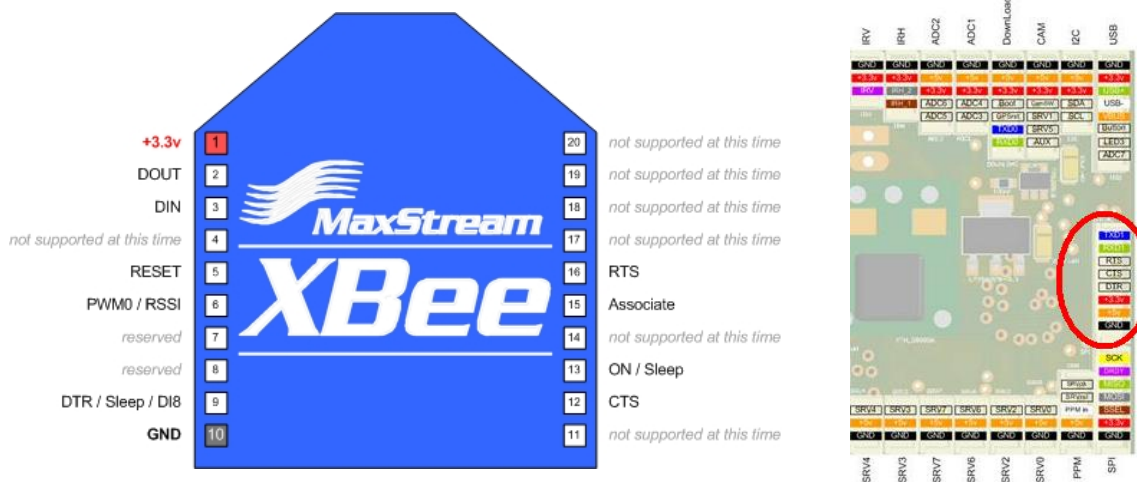


Abbildung 9-3: Anschlüsse von XBee Modem und Autopilot

Pin Nr	Name	Notes	Suggested Color
1	+3.3V	Power	Red
2	DOUT	Tx output - connect to Autopilot Rx	Green
3	DIN	Rx input - connect to Autopilot Tx	Blue
10	GND	Ground	Black

Tabelle 9-1: Verbindungen zwischen XBee Modem und Autopilot

Um die Verbindungen einfach lösen zu können haben wir einen 2 mm Stift/Buchsenleiste verwendet an der wir die Drähte angelötet haben.

Als Anschluss an den Autopiloten wird ein 8 poliger Picoblade Stecker verwendet.



Abbildung 9-4: Modem

9.3.2 Bodenstationsmodem

Das Modem für die Bodenstation kann bei unserem Bundle einfach in die dafür vorgesehene Halterung eingesetzt werden.

Das XBee Modem kommt ohne Antenne für die Bodenstation. Es ist aber eine handelsübliche 2.4 GHz Antenne die an jedem Internetrouter für den Heimgebrauch dran ist. Unsere Antenne stammt von einem solchen Modem/Router.



Abbildung 9-5: Bodenstations Modem

9.4 Diverse Servokabel

Da auf dem Tiny nur die Picoblade-Steckverbindungen montiert sind, müssen alle Verbindungen für die Sevos gekrimpt werden. Es ist essentiell, dass diese Verbindungen sehr sorgfältig erstellt werden, da bei einem Ausfall einer Verbindung der Servo nicht mehr funktioniert und so das Flugzeug abstürzen könnte.

1. drei poliger Picobladestecker (Weibchen)
2. Futaba-Servo-Stecker (Weibchen)



Abbildung 9-6: Servokabel

9.5 Empfänger→Paparazzi

Wie in Abschnitt 6 schon erwähnt, kann die Verbindung vom RC-Empfänger verschiedenartig erstellt werden. Wir entschieden uns für einen Encoder, da hier die Vorteile für uns klar überwiegen. Es gibt natürlich verschiedene Möglichkeiten diese Verbindungen zu erstellen. Um möglichst wenige Schwachstellen und fest montierte Elemente am System zu haben, entschieden wir uns möglichst alle Verbindungen steckbar zu machen. Das Encoder Board haben wir darum mit Pins versehen, um so handelsübliche Futaba-Patchkabel verwenden zu können.

10 Paparazzi Inbetriebnahme

10.1 Flashen des Tiny

Das Flashen des Tiny beinhaltet, dem Autopiloten alle relevanten Informationen für den Flug in den Speicher des Mikrocontrollers zu laden. Das bedeutet es werden z.B. die Konfiguration des Flugzeugs, Servoeinstellungen, Flugplan und auch Fernsteuerungseinstellungen übermittelt. Um genau zu sein sind das folgende Files:

- Airframe
- Flightplan
- Settings
- Radio
- Telemetry
- Diverse C-Codes

Nach dem Start des Paparazzi Centers, erscheint folgendes Fenster. Im Linken Abschnitt des Fensters können Änderungen an der Konfiguration des Autopiloten gemacht werden. Stellen Sie sicher, dass alle Einstellungen in den Configurationsfiles stimmen (siehe Abschnitt 13). Für die ersten Versuche empfiehlt es sich eine Demonstrations Setup zu verwenden, indem alle Einstellungen schon richtig vorhanden sind. Wir verwendeten den TJ1(Twin Jet) mit dem funjet.xml Airframe das alle Einstellungen für den Tiny V 2.11 vorkonfiguriert hat. Stellen Sie sicher, dass das GCS und die Configurations Files auf dem neusten Stand sind (Siehe Abschnitt 8.4). Wenn sie sich sicher sind, dass sie die richtigen

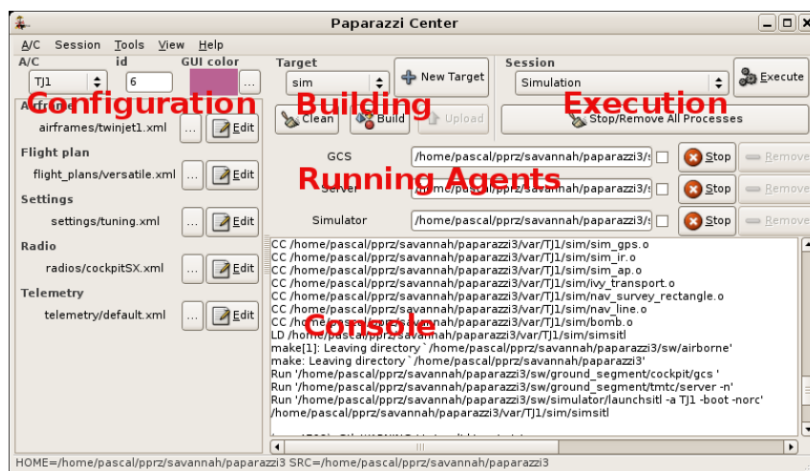


Abbildung 10-1: Servokabel

Files ausgewählt haben, selektieren sie das Ziel (Target), wählen Sie „ap“ (Autopilot) und klicken Sie auf „Build“. In der Konsole sehen sie den Output zur Kompilierung der Einstellungen. Fahren Sie nur weiter wenn der Vorgang fehlerlos abgeschlossen wurde.

Falls dies der Fall ist verbinden Sie den Tiny (nicht das UtilityBoard) per USB-Kabel mit dem Computer. Stellen Sie sicher, dass die Stromversorgung des Tiny erst nach dem Verbinden des USB-Kabels angeschlossen wird. Auf der Rückseite des Tiny sollte jetzt kein LED blinken. Dies signalisiert, dass der Tiny neu geflasht werden kann. Falls doch ein LED blinkt stimmt etwas mit der Verbindung nicht. Nun kann das Programm mit dem Kopf „Upload“ in den Tiny geladen werden. Wenn der Ladevorgang beendet ist, beginnt eine rote LED zu blinken. Dies signalisiert dass der Autopilot betriebsbereit ist.

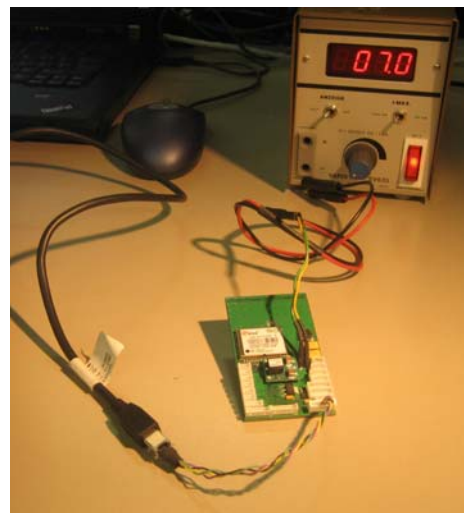


Abbildung 10-2: Aufbau zum Flashen des AP

10.2 Datenverbindung Paparazzi → Tiny erstellen

Es gibt zwei Arten um die serielle Verbindung zum Tiny herzustellen. Einerseits kann mit entsprechenden seriellen Kabeln eine direkte Verbindung zum Tiny hergestellt werden, oder aber auch mit den XBee Modem. Das serielle Kabel ist für den Flugbetrieb natürlich nicht brauchbar und dient nur zu Testzwecken oder Konfigurationsarbeiten. Wir brauchten die Verbindung mit Kabeln nur für erste Gehversuche und um bei der Problemsuche mit den XBee Modem den Fehler einkreisen zu können. Für die ersten Versuche empfehlen wir das Airframe „funjet.xml“ zu verwenden, da dort alle wichtigen Einstellungen für den Tiny eingestellt sind.

10.3 Serielle Verbindung mit Kabel

Wie schon erwähnt ist die Verbindung über das Kabel nur für Konfigurations- und Testzwecken geeignet. Wenn sie aber nicht sicher sind ob es sich um ein Problem mit der Funkverbindung oder es sich um Einstellungen in der Konfiguration handelt ist dies ein einfacher und guter Weg dies zu ermitteln. Stellen Sie Sicher, dass Sie entweder alle Updates gemacht haben und wie unter Punkt Tiny Flashen (Abschnitt 10.1 und Abschnitt 13) beschrieben den Tiny geflasht haben oder stellen Sie sicher dass folgende Einstellungen im Airframe gemacht sind. Falls Sie etwas am Airframe ändern, müssen sie den Tiny neu flashen.

- 1) Abschnitt der GPS Einstellungen: UART0
- 2) Abschnitt transparent: UART1

Verwenden Sie das Serielle Kabel und verbinden Sie es mit der Groundstation resp. dem FTDI-Utility Board. Verbinden Sie das Board nun mit dem Computer per USB. Schalten Sie nun die Spannungsversorgung für den Tiny an und Starten sie das Paparazzi GroundControl Center, wenn es noch nicht gestartet ist. Wählen sie nun die richtigen Verbindungseinstellungen (Demo Einstellungen: „Flight USB Serial 9600“) und drücken Sie Execute. Wenn die Verbindung erfolgreich erstellt werden konnte, wird das Center nun die Telemetriedaten im Fenster anzeigen. Auch der Flugplan wird angezeigt und es können die Google-Maps-Karten heruntergeladen werden. Falls die Verbindung nicht erfolgreich erstellt werden konnte bleibt der Bildschirm schwarz und es werden keine Telemetriedaten angezeigt. Falls dies der Fall ist kontrollieren sie ob alle Stecker in Ordnung sind und ob TX / RX am seriellen Kabel gekreuzt sind (siehe Abschnitt 9).

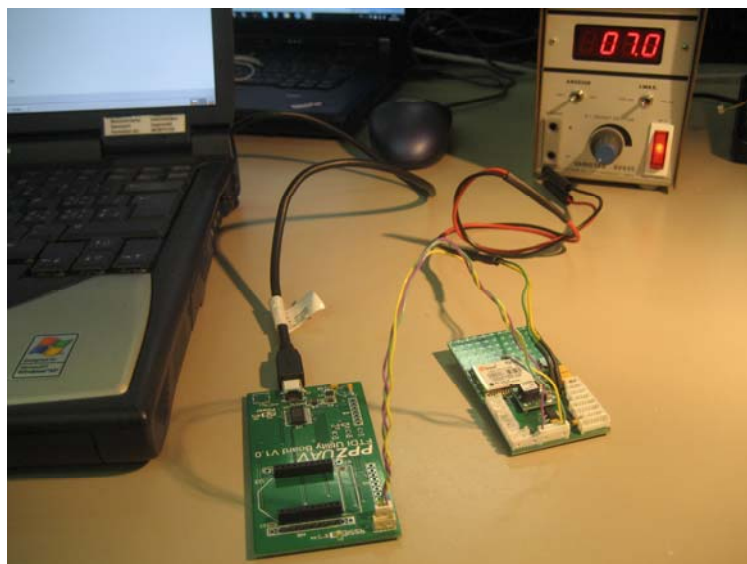


Abbildung 10-3: Aufbau für Kommunikation mit AP

10.4 Verbindung über X BEE Modem

Für einen Flug mit dem Autopiloten an Bord des Flugzeuges wird natürlich eine drahtlose serielle Verbindung benötigt.

Bei unserem Tiny V2.11 Bundle wurden XBee Pro Series 2 Funkmodule der Firma Digi (ehemals Maxstream) mitgeliefert. Die Module ermöglichen preisgünstige und zuverlässige Kommunikationslösungen, auch ohne Spezialkenntnisse von Funkverbindungen. Beide Module sind Sender und Empfänger um Daten in beide Richtungen schicken zu können. Neben einfachen Point-to-Point oder Point-to-Multipoint Verbindungen sind damit auch komplexere Kommunikationsstrukturen möglich, z.B. die Weitergabe von Informationen weit entfernter Module über dazwischen liegende Netzwerkknoten. Dies ermöglicht zum Beispiel auch das steuern mehrerer Flugzeuge mit einer Bodenstation. Die Stromaufnahme im

aktiven Zustand (Senden/Empfangen) beträgt etwa 40mA bei 3,3V für eine Sendeleistung von 50mW. Dies entspricht in etwa einer Reichweite von 3000m auf offenem Feld.



Abbildung 10-4:
Modem Bodenstation



Abbildung 10-5:
Modem Flugstation

10.4.1 XBEE konfigurieren

Die Module kommen vorkonfiguriert und sie sollten eigentlich plug and play sein. Falls dies bei Ihnen nicht der Fall ist oder wenn Sie eine höhere Baudrate (9600bps Standardwert) für die Telemetriedaten wünschen, müssen Sie die Module neu konfigurieren. Eine höhere Baudrate ist für den einfachen Flugbetrieb nicht nötig. Wenn aber viele Telemetriedaten oder ein Videosignal übertragen werden soll muss die Baudrate erhöht werden.

Zur Konfiguration der Module kann die XCTU-Software heruntergeladen werden. Dies ist ein Programm, das dem Benutzer erlaubt über ein GUI auf die XBEE-Modul Konfiguration zuzugreifen und Einstellungen daran vorzunehmen.

1. Laden sie die X-CTU Software bei www.digi.com herunter und installieren Sie sie. Die Software funktionierte bei uns nur auf einem Windows XP Computer einwandfrei.
2. Stecken Sie nun ihr FTDI-Utility Board mit der Bodenstationsantenne ein und installieren Sie die Treiber.
3. Zuerst muss die Baudrate des Modems am Computer eingestellt werde um die Kommunikation zwischen PC und XBee zu ermöglichen. Je nach Einstellung die im Modem gespeichert ist muss diese zuerst

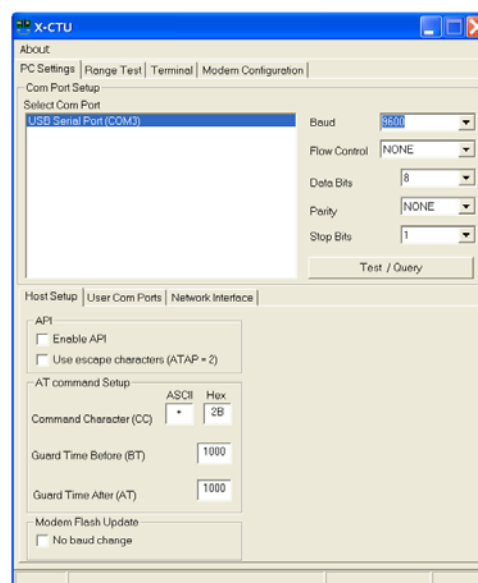


Abbildung 10-6: XCTU GUI

gesucht werden. Wenn Sie die gewünschte Baudrate eingestellt haben drücken Sie auf „Test/Query“. Wenn hohe Raten konfiguriert sind, muss eventuell bei „Enable API“ ein Haken gesetzt werden.

- 1) Bei erfolgreichem Verbindungsaufbau erscheint dieses Fenster. Der Verbindungsaufbau war nur erfolgreich wenn Modemtyp und die Firmware Version angezeigt werden. Der Computer ist nun in der Lage die Daten mit dem Modem auszutauschen und sie zu verändern.

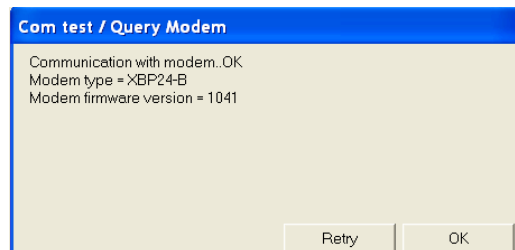


Abbildung 10-7: Meldung für funktionierende Kommunikation

- 2) Die Installation der neusten Updates hat bei unserer Softwareversion nicht funktioniert und musste von Hand erledigt werden. Zuerst müssen auf der Digi Internetseite die neuesten Firmware Versionen heruntergeladen werden. Den Modemtyp erhalten sie wenn Sie im Reiter „Modem Configuration“. „Read“ drücken. Klicken Sie auf „Download new versions“ und selektieren Sie die entsprechenden Dateien mit denen sie updaten wollen.

- 3) Wechseln Sie nun in den Reiter „Modem Configuration“. Hier kann nun mit „Read“ die gespeicherte Konfiguration ausgelesen werden. Stellen Sie nun sicher, dass sie die neuste Firmware heruntergeladen und installiert haben. Falls sie ein Firmware update durchführen müssen empfehlen wir zuerst das Update durchzuführen und erst später neue Einstellungen zu speichern. Sie können nun die Änderungen vornehmen die sie wollen.

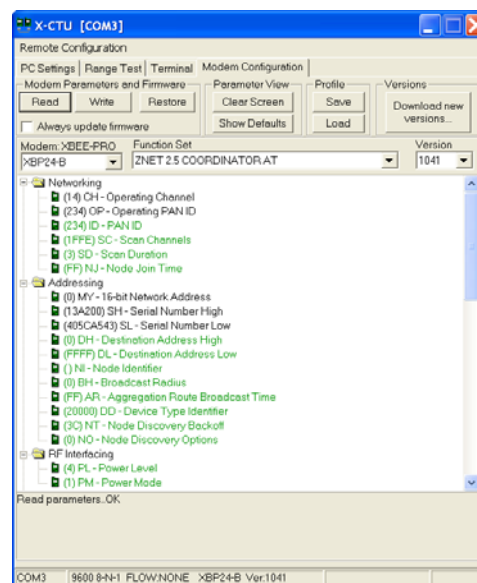


Abbildung 10-8: Modem Konfiguration

- 4) Rangetest

Der Rangetest kann für eine Kontrolle ob alles in Ordnung ist sehr hilfreich sein. Sie müssen aber dazu am Flugzeugmodem (kurze Antenne) die beiden Pins RX und TX verbinden (Abschnitt 9.3) und es mit Strom versorgen. Die Groundstation verbinden Sie per USB mit dem Computer.

Eine sehr gute Anleitung ist auf der Seite www.Digi.com erhältlich, sie befindet sich aber auch im Anhang. In dieser Dokumentation ist jeder Schritt genau erklärt.

10.4.2 Verbindung zum Autopiloten erstellen

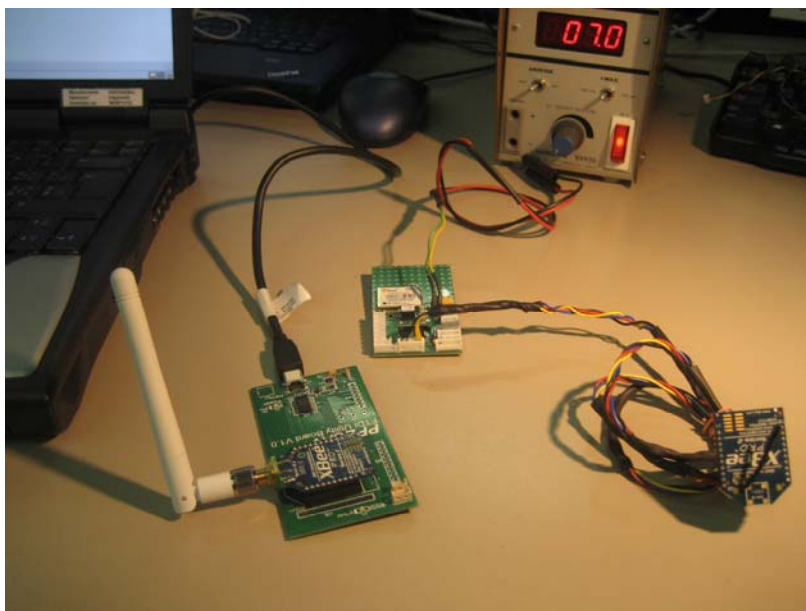


Abbildung 10-9: Aufbau für Kommunikation via Funk

erstellt worden. Falls dies nicht funktioniert hat, überprüfen sie das Airframe und ob das Modem richtig konfiguriert wurde.

Verbinden Sie dafür die Bodenstation (FTDI-Board mit Modem) mit dem Computer. Schliessen Sie das Flugmodem an den Autopiloten an. Versorgen sie den Autopiloten nun mit Spannung. Nun kann das Paparazzi-Groundcenter gestartet werden. Wenn Sie sehen, dass die Telemetriedaten empfangen werden, ist die Verbindung erfolgreich

10.5 GPS Konfiguration

Um die GPS-Daten vollständig zu erhalten müssen im Airframe (siehe Kapitel 13.1) noch einige Anpassungen gemacht werden. Stellen Sie sicher, dass die Baudrate und die Schnittstellen (UART) korrekt auf ihren Autopiloten eingestellt sind. Falls Sie das GPS-Modul anders konfigurieren wollen, können sie über den UART-Tunnel und der Software u-Center von der Firma U-Blox das GPS-Modul neu konfigurieren.

10.6 Inbetriebnahme

Nun kann alles verbunden werden und wenn alles richtig funktioniert hat, sollte Sie jetzt im GCS den Künstlichen Horizont sehen und ihren Flugplan. Falls Sie draussen sind sollte innerhalb von ca. 30 Sekunden ein GPS-Fix erstellt sein. Falls Sie in einem Raum sind, denken Sie daran, dass der künstliche Horizont fehlerhafte Resultate anzeigt und er nur im Freien richtig funktioniert.

11 GUI

11.1 Aufstarten

Um das Programm zu Starten, wählt man über das Menu Application → Sonstige → Paparazzi oder über das entsprechende Icon aus der Taskliste das Paparazzi Center. Erscheinen wird ein Bild entsprechend der Abbildung 11-1. Das GUI ist in fünf Felder aufgeteilt, die vom Anwender der Reihe nach bearbeitet werden sollten. Im Folgenden wird jede Funktion beschrieben.

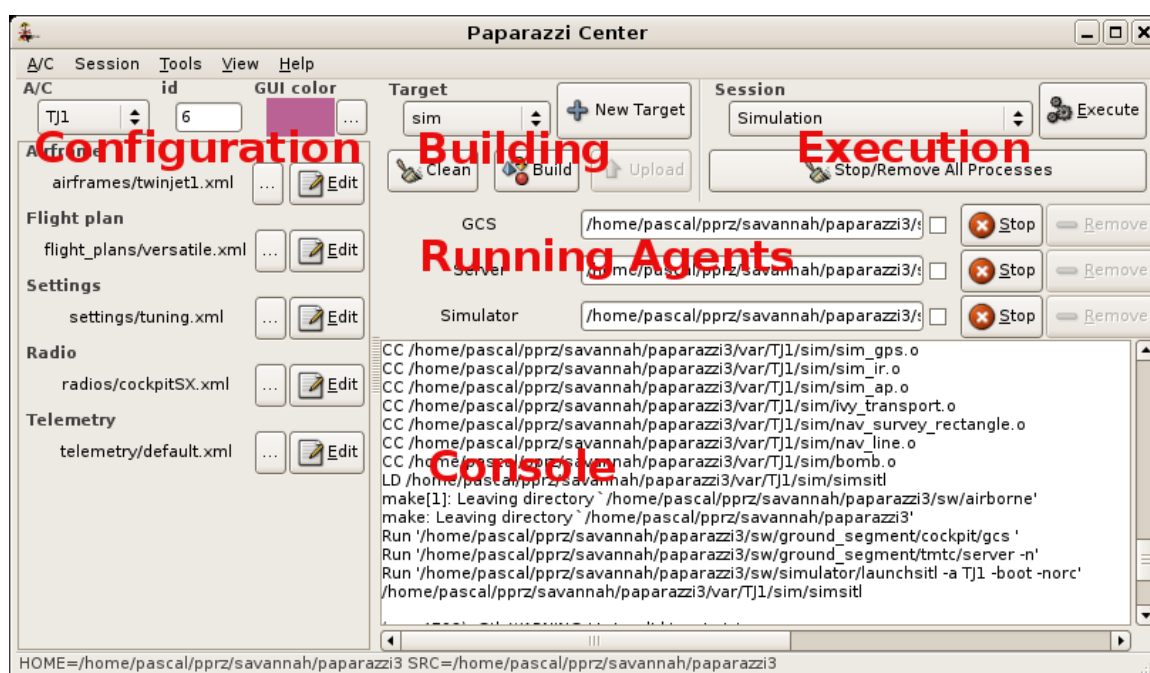


Abbildung 11-1: Paparazzi Center

11.2 Menu

11.2.1 A/C

Unter dem Menu A/C kann ein neues Aircraft mit der Entsprechenden Konfiguration erstellt, gespeichert oder gelöscht werden. Auch das Beenden vom Paparazzi Center kann über dieses Menu erfolgen.

11.2.2 Session

In diesem Menu kann eine Session gelöscht oder erstellt werden. Als Session bezeichnet Paparazzi die Art der Kommunikation. Für weitere Information siehe Kapitel 11.3.3 Session.

11.2.3 Tools

Hier können alle Prozesse gestartet werden. Interessant für den Benutzer sind vor allem der Flight Plan Editor, die Messages, der Real-time Plotter und der Environment Simulator. Wobei der Letzte nur bei der Simulation eingesetzt werden kann. Zum Ausführen einer Simulation oder bei einem Flug werden automatisch der GCS, der Server und eventuell der Simulator respektive der Data Link gestartet.

- Data Link

Der Data Link wird benötigt um die Daten per Funk oder auch über das serielle Kabel



Abbildung 11-2: Data Link

zwischen dem Autopiloten und der Bodenstation zu übermitteln.

- Environment Simulator (nur Simulation)

In der Simulation kann mit dem Environment Simulator (alte Bezeichnung Gaia) die Umwelt simuliert werden. So kann man zum Beispiel die Richtung und Geschwindigkeit des Windes eingestellt werden oder das GPS-Signal ausschalten. Wobei zu beachten ist, dass eigentlich der Autopilot ohne GPS keine Koordinaten senden kann. Somit könnte im GCS theoretisch keine Position des Flugzeuges angezeigt werden.

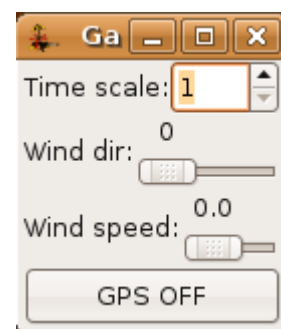


Abbildung 11-3: Gaia

- Flight Plan Editor

Es wird empfohlen den Flugplan mit der graphischen Oberfläche zu bearbeiten. Für Fortgeschrittene kann es jedoch hilfreich sein die grundlegenden Arbeiten wie setzen von Punkten usw. händisch in die entsprechende XML-Datei zu schreiben.

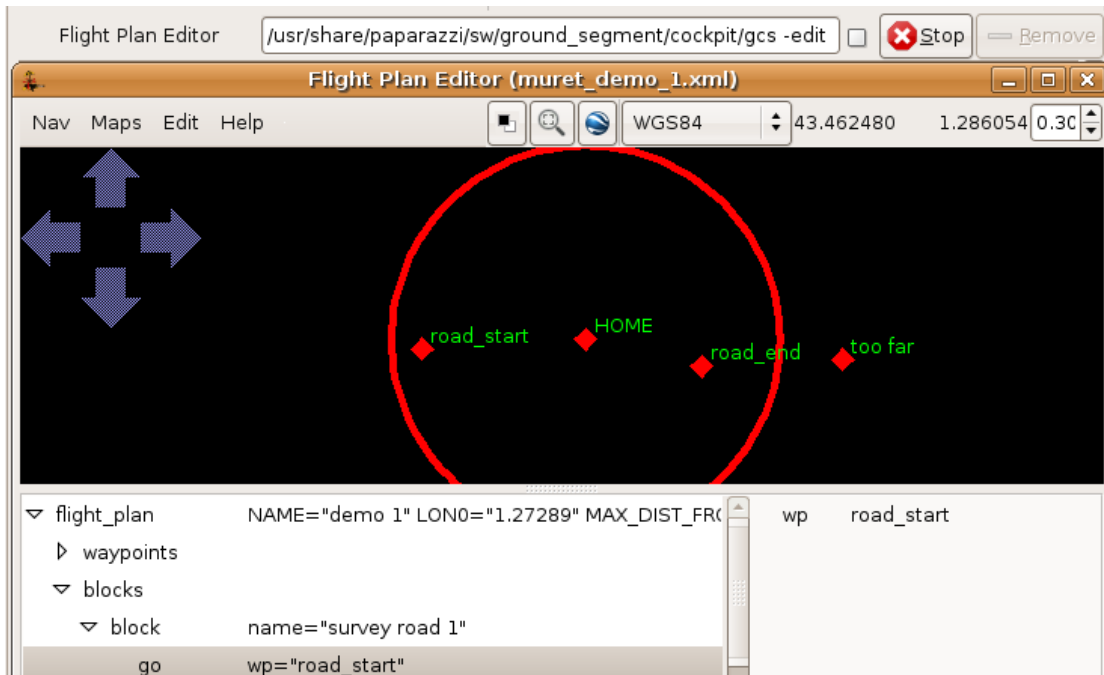


Abbildung 11-4: Flight Plan Editor

- GCS
In diesem Prozess wird das Cockpit dargestellt. Jeglichen Flugeinstellungen werden hier betätigt und überwacht. Einen ausführlichen beschrieb ist unter dem Kapitel 14 zu finden.
- Hardware in the Loop
<http://paparazzi.enac.fr/wiki/HITL>
- http Server
- Log File Player
Für einen Replay wird der Log File Player benötigt. Dabei wird die Kommunikation die in der Log-Datei notiert wurde gelesen und im GCS simuliert.
- Log Plotter
mit dem Plotter kann eine bestehende Log-Datei geöffnet werden. Wählt man im Menu den gewünschten Datensatz aus, so wird er im Plotter über der Zeit aufgezeichnet.



Abbildung 11-5: Log File Player

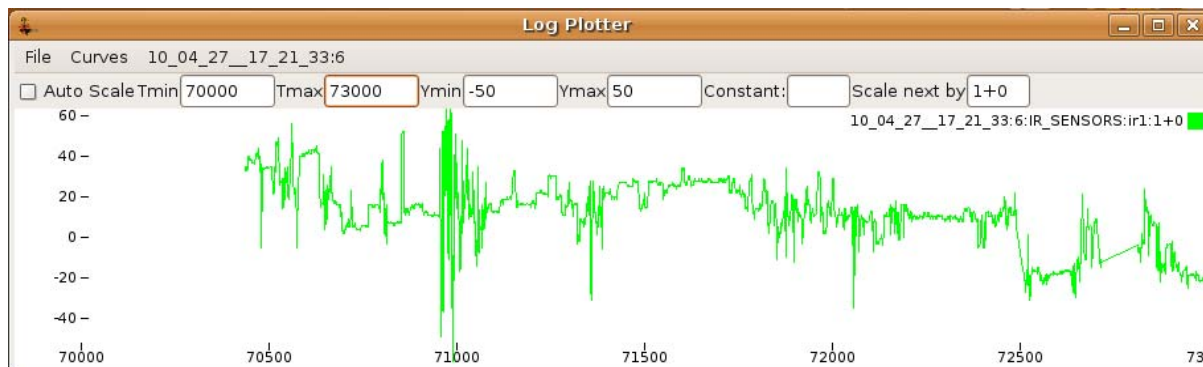


Abbildung 11-6: Log Plotter

- Messages

Im geöffneten Fenster können die Daten beobachtet werden. Jedes mal, wenn ein Feld farbig hinterlegt wird, hat die Bodenstation dieses Datenpaket empfangen. Dieser Prozess wird auch benötigt um ein Datensatz per Drag n' Drop in den Real-time Player oder in das GCS zu ziehen.

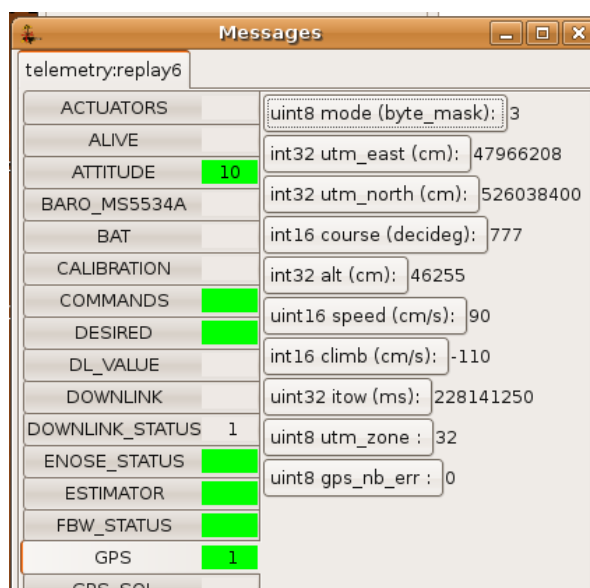


Abbildung 11-7: Messages

- Real-time Plotter

Dieser wird benötigt um einen Verlauf eines Datensatzes aufzuzeichnen. Der gewünschte Parameter kann per Drag n' Drop von dem Message-Prozess abgezogen werden.

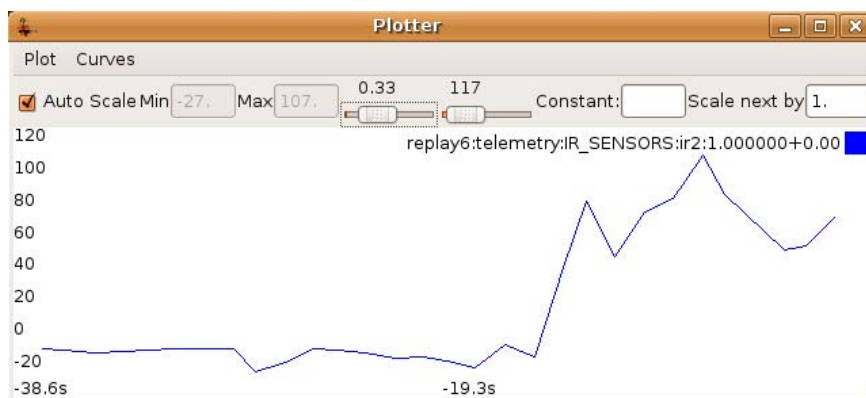


Abbildung 11-8: Real-time Plotter

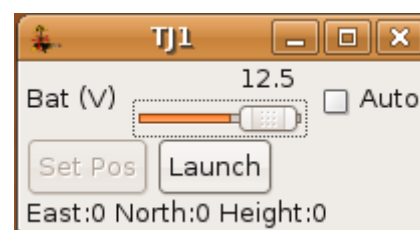
- Server

Der Server wird bei der Simulation und bei dem Flug benötigt. Er bereitet die Daten für das GCS auf und schreibt die Daten in eine Log-Datei. Diese Datei ist unter dem Pfad ... zu finden

- Settings

- Simulator

Der Simulator wird nur bei der Simulation benötigt. Er beinhaltet einen Schieberegler für die Regulierung des Batterielevels und ein Knopf für das Starten des Motors (Launch). Mit der Checkbox Auto kann der Verbrauch der Batterie simuliert werden. Jedoch muss dafür im Airframe ein Wert angegeben werden, der beschreibt wie viel der Verbrauch pro Sekunde beträgt. Wird der Prozess ohne das Attribut boot gestartet, so kann man den Startpunkt der Simulation festlegen. Dies geschieht durch die Eingabe von den WGS84 Koordinaten und der Starthöhe. Jedoch ist darauf zu achten, dass die Koordinaten mit dem Airframe übereinstimmen müssen.



Launchsitl -a TJ1 -boot -norc

Abbildung 11-9: Simulator

11.3 Bereiche

11.3.1 A/C

Im Bereich A/C des GUI's kann ein bestehendes, programmiertes Flugobjekt ausgewählt werden. Mit der Auswahl eines A/C's stellen sich automatisch die dazugehörige Nummer, die Farbe des GUI's und die XML-Dateien ein.

Um ein A/C zu erstellen kann das Menü benutzt werden oder man programmiert das neue A/C direkt im XML-File. Die Datei findet man unter `~/paparazzi/conf/conf.xml`

Die XML-Dateien können über den Button mit den drei Punkten ausgewählt werden oder mit dem Edit-Button bearbeitet werden. Die Ordner sind unter dem Pfad `~/paparazzi/conf/` zu finden. Die Beschreibung der Dateien wird im Kapitel 13 beschrieben.

11.3.2 Target

Die Auswahlliste unter der Überschrift Target beinhaltet die möglichen Ziele. Zur Simulation gehört der Eintrag sim und ap steht für Autopilot. Mit der Schaltfläche Build werden die XML-Einstellungen kompiliert und der Code entsprechend dem Target erzeugt. Der Vorgang ist abgeschlossen, wenn der make-Prozess aus der Liste entfernt wurde. Ist ein Autopilot angehängt, so wird der erstellte Code mit dem Button Upload auf den Chip geschrieben. Um den Workspace aufzuräumen betätigt man die Clean-Schaltfläche.

11.3.3 Session

Mit der Auswahlliste Session kann ein Eintrag ausgewählt werden, der die Kommunikation zwischen dem Autopiloten und der Bodenstation festlegt. Des Weiteren wird mit der Auswahl auch festgelegt welche Prozesse mit welchen Attributen gestartet werden. Die Daten sind in der Datei unter `~/paparazzi/conf/control_panel.xml` in der Sektion sessions gespeichert.

Wird die Schaltfläche Excecute angewählt, so werden die Prozesse gestartet und unterhalb aufgelistet. Jeder einzelne Prozess kann gestoppt, entfernt und neu gestartet werden. Bei einem Neustart könne auch die entsprechenden Attributen gesetzt werden. Die Schaltfläche Stop/Remove All Prozesses entfernt alle Prozesse. Ist die Checkbox neben der Stop-Taste angewählt, so wird der entsprechende Dienst nach seiner Beendigung automatisch neu gestartet.

12 Operationsarten

12.1 Simulation

Um eine Simulation durchzuführen, wählt man nach dem Starten der Applikation das gewünschte A/C aus. Danach stellt man das Target auf sim und betätigt den Build-Knopf. Nach dem erfolgreichem generieren des Codes – zwei make Einträge am Ende des Plots – muss die Session auf Simulation gestellt werden. Somit kann die Simulation gestartet werden, in dem man auf Execute klickt. Dabei öffnen sich das GCS, der Server und der Simulator. Es sollten vier blaue Pfeile im oberen linken Eck und diverse Punkte in der Farbe des GUI color Feld erscheinen.

12.2 Replay

Mit dem Replay kann ein bestehendes Log-File wiedergegeben werden. In der Session wird der Eintrag Replay ausgewählt und mit Execute ausgeführt. Es wird der Log File Player, der Server und das GCS gestartet. Im Player kann unter dem Menu File eine Log-Datei geöffnet werden, die man abspielen will. Nach dem Öffnen und Starten richtet sich der GCS ein und spielt den Flug ab. Wobei mit dem Environment Simulator die Geschwindigkeit eingestellt werden kann. Um die Scrollbar des Log File Player zu nutzen, muss die Wiedergabe mit dem Menu gestoppt werden.

12.3 Flug

Der Autopilot muss zu Beginn mit den aktuellen Daten beschrieben werden. Dafür hängt man den Autopiloten direkt mit dem USB an den Laptop, öffnet das Paparazzi Center und generiert den Code mit der Build-Schaltfläche. Danach wird der Autopilot mit der Spannung versorgt. Der AP entdeckt das angeschlossene USB Kabel und wartet auf einen Upload. Nach der Betätigung der Upload-Taste wird der Upload durchgeführt und mit einem blinkenden Led am Autopiloten bestätigt der AP den erfolgreichen Upload. Danach kann das USB Kabel an die Bodenstation gehängt werden und das Target „Flight USB-serial@9600“ ausgeführt werden. Es startet dabei der Data Link, der Server und das GCS. Bei erfolgreicher Kommunikation wird der Standort des Flugzeuges (wenn GPS-Signal vorhanden) und der Flugplan angezeigt.

12.4 Hardware in the Loop HITL

<http://paparazzi.enac.fr/wiki/HITL>

13 Konfigurations-Dateien

13.1 Airframe

Die Datei, die das Airframe beschreibt ist unter dem Pfad `~/paparazzi/conf/airframes` zu finden. Bei der Auswahl ist darauf zu achten, dass die XML-Datei für den Tiny 2.1 oder höher ist. Ansonsten wird der Upload nicht funktionieren.

13.1.1 Struktur

Die Struktur der Airframes ist in der Datei `~/paparazzi/conf/airframes/airframe.dtd` festgelegt. Sie ist in etwa der nebenstehenden Abbildung zu entnehmen. Um den genauen, notwendigen Syntax zu verstehen, empfiehlt es sich die `airframe.dtd` Datei zu studieren und allenfalls den Wikipedia-Eintrag über Dokumenttypdefinition nachzuschlagen.

(<http://de.wikipedia.org/wiki/Dokumenttypdefinition>)

```
airframe
  servos
    servo
  commands
    axis
  rc_commands
    set
  ap_only_commands
    copy
  command_laws
    let
    set
  section
    define
    linear
  makefile
  #PCDATA
```

Abbildung 13-1: airframe Struktur

13.1.2 Beispiel

Im Folgenden wird das Airframe anhand des Funjets erläutert.

- Airframe

```
<airframe name="Funjet 1 Tiny 2.1">
```

- Servos

```
<servos>
  <servo name="MOTOR" no="0" min="1000" neutral="1000" max="2000"/>
  <servo name="AILEVON_LEFT" no="2" min="1130" neutral="1500" max="1880"/>
  <servo name="AILEVON_RIGHT" no="6" min="1980" neutral="1515" max="1170"/>
</servos>
```

Im Element Servos werden die einzelnen, physikalisch vorhandenen Servos deklariert und justiert. Der Name ist frei wählbar und die Nummer entspricht dem Ausgang auf dem Autopilot. Den Attribute `min`, `neutral` und `max` können einen Wert zwischen $1000\mu\text{s}$ bis $2000\mu\text{s}$ zugewiesen werden. Dies entspricht der Maximalen, respektiv Minimalen Impulsdauer des PWM-Signals.



Abbildung 13-2: Funjet

- Commands

```
<commands>  
  <axis name="THROTTLE" failsafe_value="0"/>  
  <axis name="ROLL" failsafe_value="0"/>  
  <axis name="PITCH" failsafe_value="0"/>  
</commands>
```

Die drei Kommandos werden zur Berechnung im Kontroller gebraucht. Sie werden in Abschnitt der `command_laws` weiter verarbeitet. Der `failsafe_value` kommt zum Einsatz, wenn der Kontroller nicht läuft wie zum Beispiel beim Aufstarten. Der Wert kann zwischen ± 9600 eingestellt werden.



Abbildung 13-3: Achsen

- RC Commands

```
<rc_commands>  
  <set command="THROTTLE" value="@THROTTLE"/>  
  <set command="ROLL" value="@ROLL"/>  
  <set command="PITCH" value="@PITCH"/>  
</rc_commands>
```

Diese `set`-Befehle sind notwendig, um die Kanäle des Empfängers den Achsen (siehe Abbildung 13-3) zu zuweisen.

Will man im Auto1- und Auto2-Modus die Fernsteuerung nutzen können, so muss man ein Bereich `auto_rc_commands` erstellen und die Werte mit dem `set`-Befehl übergeben.

Reine Autopilot-Kommandos werden in einem Bereich `ap_only_commands` definiert. Sie können nicht mit der Fernsteuerung manipuliert werden, jedoch immer (unabhängig vom Modus) vom GCS gesteuert werden können. Dies kann zum Beispiel als Schutz der Kamera genutzt werden, die beim Unterschreiten einer gewissen Höhe das Objektiv einfährt.

- Mixer

```
<section name="MIXER">  
  <define name="AILEVON_AILERON_RATE" value="0.75"/>  
  <define name="AILEVON_ELEVATOR_RATE" value="0.75"/>  
</section>
```

mit dem Mixer können Variablen eingestellt werden. Jedoch wäre es auch möglich diese einfache Rechnung direkt im `command_laws`-Bereich durchzuführen.

- Command Laws

```
<command_laws>
  <let var="aileron" value="@ROLL * AILEVON_AILERON_RATE"/>
  <let var="elevator" value="@PITCH * AILEVON_ELEVATOR_RATE"/>
  <set servo="MOTOR" value="@THROTTLE"/>
  <set servo="AILEVON_LEFT" value="$elevator + $aileron"/>
  <set servo="AILEVON_RIGHT" value="$elevator - $aileron"/>
</command_laws>
```

Die Ausgänge des Autopiloten werden skaliert und einem theoretischen Leitwerk zugewiesen. Durch die set-Funktion werden die theoretischen Werte anschliessend den effektiven Servomotoren zugewiesen. Laut dem Wiki von Paparazzi sollte die maximale Auslenkung der Servomotoren immer ausgenützt werden.

- Auto1

```
<section name="AUTO1" prefix="AUTO1_">
  <define name="MAX_ROLL" value="0.85"/>
  <define name="MAX_PITCH" value="0.6"/>
</section>
```

Um die erhöhte Stabilität des Auto1-Modus zu erreichen, werden hier die maximalen Werte der Befehle angegeben. Die Werte werden als Radiant erwartet, können jedoch mit der Funktion RadOfDeg() umgerechnet werden.

- Analog-Digital-Converter

```
<section name="adc" prefix="ADC_CHANNEL_">
  <define name="IR1" value="ADC_1"/>
  <define name="IR2" value="ADC_2"/>
  <define name="IR_TOP" value="ADC_0"/>
  <define name="IR_NB_SAMPLES" value="16"/>
  <define name="GYRO_ROLL" value="ADC_3"/>
  <define name="GYRO_NB_SAMPLES" value="16"/>
</section>
```

Dieser Abschnitt befasst sich mit dem Deklarieren der Eingänge des Autopiloten. Also mit der Frage welcher Sensor an welchem Stecker angehängt ist. Um einen Eingang zu aktivieren muss dieser auch im Makefile-Bereich aktiviert werden. Siehe dafür das Wiki.

http://paparazzi.enac.fr/wiki/Airframe_Configuration#Hardware_definitions_-_Makefile

- IR-Sensoren

```
<section name="INFRARED" prefix="IR_">
  <define name="ADC_IR1_NEUTRAL" value="512"/>
  <define name="ADC_IR2_NEUTRAL" value="512"/>
  <define name="ADC_TOP_NEUTRAL" value="520"/>
```

In den ersten drei Zeilen werden die Sensoren kalibriert. Als Sensor wird immer ein Paar angeschaut, dass zum Beispiel links und rechts vergleicht. Ein optimaler Sensor

sollte den Wert 512 ausgeben, wenn er auf beiden Seiten das Gleiche misst. Um sie zu kalibrieren, legt man sie am Besten in eine Schachtel und liest den durchschnittlich, gemessenen Wert mit dem Real-time Plotter ab. Der neue Wert wird nach der unten stehenden Formel gerechnet.

$$value := value + Messung$$

```
<define name="IR2_SIGN" value="-1"/>
<define name="TOP_SIGN" value="-1"/>
```

Diese zwei Zeilen definieren das Vorzeichen der Sensoren. Laut der Definition von Paparazzi sollte der Messwert positiv sein, wenn die Temperatur unter dem Flugzeug wärmer ist als oberhalb. Für die horizontale Ausrichtung sollte es bei positivem Vorzeichen hinten rechts wärmer sein.

```
<define name="HORIZ_SENSOR_TILTED" value="1"/>
```

Mit dieser Zeile wird die Ausrichtung des IR-Sensors festgelegt. Mit dem Statement Horiz_Sensor_Tilted = 1 legt man fest, dass die Achsen des horizontalen Sensor mit einem Winkel von 45° zu den Achsen des Flugzeugs stehen. Will man die Achse des Sensors mit der Achse des Flugzeugs übereinander legen, so wählt man das Statement: Horiz_Sensor_Aligned = 1

```
<define name="LATERAL_CORRECTION" value="1."/>
<define name="LONGITUDINAL_CORRECTION" value="1."/>
<define name="VERTICAL_CORRECTION" value="1.5"/>
<define name="ROLL_NEUTRAL_DEFAULT" value="-2.5" unit="deg"/>
<define name="PITCH_NEUTRAL_DEFAULT" value="6" unit="deg"/>
<define name="CORRECTION_UP" value="1."/>
<define name="CORRECTION_DOWN" value="1."/>
<define name="CORRECTION_LEFT" value="1."/>
<define name="CORRECTION_RIGHT" value="1."/>
</section>
```

Diese Werte sind alles Faktoren, die beim Tuning eingestellt werden. Die drei ersten Werte werden genutzt, um den drei Achsen ähnliche Kontraste zuzuweisen. Für einen Sensoranordnung von 45° sind die notierten Werte als Standard zu betrachten. Wird nichts eingestellt, so wird der Default-Wert 1 angenommen. Die zwei „Neutral“-Parameter werden so eingestellt, dass das Flugzeug gerade (Roll_Neutral_Default) und auf gleicher Höhe (Pitch_Neutral_Default) fliegt. Die Tuningparameter können während dem Flug im GCS eingestellt werden. Siehe dazu das Wiki.

<http://paparazzi.enac.fr/wiki/Tuning>

- Batterie

```
<section name="BAT">
  <define name="MILLIAMP_PER_PERCENT" value="0.86"/>
  <define name="CATASTROPHIC_BAT_LEVEL" value="2.0" unit="V"/>
</section>
```

Diese Sektion dient der Sicherheit des Flugzeuges. Werden hier die Daten richtig eingetragen, so kann der Autopilot diverse Sicherheitsfunktionen realisieren und den Batteriestand anzeigen. Im Beispiel wurde nur zwei von vielen Variablen gebraucht. Der Eintrag Catastrophic_Bat_Level bestimmt den Batteriestand, bei dem der Autopilot den Motor auf 0% stellt. Zusätzlich wird in der Konsole des GCS eine Warnung ausgegeben. Um nicht von dieser Warnung überrascht zu werden, können die Variablen Critic_Bat_Level, Low_Bat_Level und Max_Bat_Level verwendet werden. Der kritische und tiefe Stand lösen nur eine Nachricht aus.

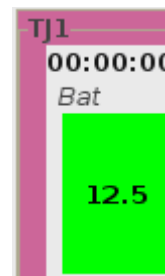


Abbildung 13-4:
Batterie Niveau

Standard mässig sind sie auf 10V und 10.5V gesetzt. Der Max_Bat_Level wird genutzt um die Anzeige im GCS zu kalibrieren. In der Abbildung 13-4 wurde der Standardwert von 12.5V nicht verändert und zeigt somit eine voll geladene Batterie an. Die Anderen Variablen sind uns momentan noch nicht verständlich.

- Horizontal Control

```
<section name="HORIZONTAL CONTROL" prefix="H_CTL_">
  <define name="COURSE_PGAIN" value="-0.8"/>
  <define name="ROLL_MAX_SETPOINT" value="0.6" unit="radians"/>
  <define name="PITCH_MAX_SETPOINT" value="0.5" unit="radians"/>
  <define name="PITCH_MIN_SETPOINT" value="-0.5" unit="radians"/>
  <define name="PITCH_PGAIN" value="-10000."/>
  <define name="PITCH_DGAIN" value="1.5"/>
  <define name="ELEVATOR_OF_ROLL" value="2500"/>
  <define name="ROLL_ATTITUDE_GAIN" value="-7500"/>
  <define name="ROLL_RATE_GAIN" value="-1500"/>
</section>
```

Die Attribute Course_PGain, Pitch_PGain, Pitch_DGain, sowie Roll_Attitude_Gain und Roll_Rate_Gain werden in der Regelung der einzelnen Loops benötigt. Sie werden während dem Tuning eingestellt und sind im GCS unter den Settings/control/nav und Settings/control/attitude zu finden. Die Berechnung des Reglers findet in der Datei ./usr/share/Paparazzi/sw/airborne/fw_h_ctl.c statt.

- Vertical Control

```
<section name="VERTICAL CONTROL" prefix="V_CTL">
  <define name="POWER_CTL_BAT_NOMINAL" value="11.1" unit="volt"/>
  <!-- outer loop proportional gain -->
  <define name="ALTITUDE_PGAIN" value="-0.04"/>
  <!-- outer loop saturation -->
  <define name="ALTITUDE_MAX_CLIMB" value="2."/>
  <!-- auto throttle inner loop -->
  <define name="AUTO_THROTTLE_NOMINAL_CRUISE_THROTTLE" value="0.5"/>
  <define name="AUTO_THROTTLE_MIN_CRUISE_THROTTLE" value="0.35"/>
  <define name="AUTO_THROTTLE_MAX_CRUISE_THROTTLE" value="0.80"/>
  <define name="AUTO_THROTTLE_LOITER_TRIM" value="1500"/>
  <define name="AUTO_THROTTLE_DASH_TRIM" value="-1000"/>
  <define name="AUTO_THROTTLE_CLIMB_THROTTLE_INCREMENT" value="0.1" unit="%/(m/s)"/>
  <define name="AUTO_THROTTLE_PGAIN" value="-0.02"/>
  <define name="AUTO_THROTTLE_IGAIN" value="0.1"/>
  <define name="AUTO_THROTTLE_PITCH_OF_VZ_PGAIN" value="0.5"/>
  <!-- auto pitch inner loop -->
  <define name="AUTO_PITCH_PGAIN" value="-0.05"/>
  <define name="AUTO_PITCH_IGAIN" value="0.075"/>
  <define name="AUTO_PITCH_MAX_PITCH" value="0.35"/>
  <define name="AUTO_PITCH_MIN_PITCH" value="-0.35"/>
  <define name="THROTTLE_SLEW" value="0.5"/>
</section>
```

Altitude_PGain, Auto_Trottle_Climb_Throttle_Increment, Auto_Trottle_PGain, Auto_Trottle_IGain, Auto_Trottle_Pitch_Of_VZ_PGain werden für die Vertikale-Regelung benötigt. Die Einstellung während des Tunings kann im GCS unter Settings/control/alt sowie Settings/control/auto_throttle durchgeführt werden. Die Datei für die Berechnung des Reglers ist unter dem Pfad `./usr/share/Paparazzi/sw/airborne/fw_v_ctl.c` zu finden.

- Misc

```
-<section name="MISC">
  <define name="NOMINAL_AIRSPPEED" value="15." unit="m/s"/>
  <define name="CARROT" value="5." unit="s"/>
  <define name="KILL_MODE_DISTANCE" value="(1.5*MAX_DIST_FROM_HOME)"/>
  <define name="CONTROL_RATE" value="60" unit="Hz"/>
  <!--
  <define name="XBEE_INIT" value="\rATPL2\rATRN1\rATTT80\r"/>
  -->
  <define name="NO_XBEE_API_INIT" value="TRUE"/>
  <define name="ALT_KALMAN_ENABLED" value="TRUE"/>
  <define name="TRIGGER_DELAY" value="1."/>
  <define name="DEFAULT_CIRCLE_RADIUS" value="80."/>
  <define name="MIN_CIRCLE_RADIUS" value="50."/>
</section>
```

Der Wert Nominal_airspeed wird nur für die Simulation benötigt, damit das Programm abschätzen kann, wie schnell sich in etwa das Flugzeug bewegen wird. Carrot beschreibt die Entfernung der Wegmarke (Carrot) vom Flugzeug. Mit dem Wert Kill_Mode_Distance wird ein Kreis um die Home-Position gerechnet. Verlässt das Flugzeug aus einem Grund diesen Kreis, so wird der Motor abgeschaltet. Dies führt zu einer definierten Reduzierung der Höhe ohne Motor. Als vierter Wert ist die Control_Rate anzuschauen. Sie bestimmt wie oft pro Sekunde die Regelkreise gerechnet werden. Als Werte werden 60 oder 20 Herz zugelassen.

- Make

Als nächster Abschnitt muss das Makefile geschrieben werden. Da wir bis jetzt nur den Abschnitt über das Modem und GPS bearbeiten mussten, wird hier auf das Wiki von Paparazzi verwiesen. Siehe dazu

http://paparazzi.enac.fr/wiki/Airframe_Configuration#Hardware_definitions_-_Makefile

13.2 Flightplan

Für die Programmierung des Flugplanes gibt es zwei Möglichkeiten. Entweder man programmiert direkt in der XML-Datei, die unter `~/Paparazzi/conf/flight_plans` zu finden ist, oder man benutzt das GCS dafür. Beide Arten haben ihre Vor- und Nachteile. So kann man zum Beispiel beim Arbeiten im XML-File keine optische Referenz auf eine Karte brauchen, kommt jedoch mit einer gewissen Übung schneller voran.

13.2.1 Struktur

Die Datei hat immer die Struktur, wie sie in der Datei `~/paparazzi/conf/flight_plan.dtd` beschrieben wird. Dazu gehört als Erstes die Referenz auf die `.dtd` Datei. Danach kommen die Lokalisierung der Ausgangsposition, sowie diverse andere Attribute. Anschliessend werden die Wegpunkte, Sektoren, exceptions und blocks definiert.

- Einleitung

```
<!DOCTYPE flight_plan SYSTEM "flight_plan.dtd">
```

```
<flight_plan alt="439" ground_alt="439" lat0="47.496872" lon0="8.729987" max_dist_from_home="1500" name="winti" security_height="0">
```

```
<header>
```

```
#include "datalink.h"
```

```
</header>
```

Die ersten drei Attribute vom `flight_plan` TAG markieren die Ausgangskoordinaten des Flugplans. Die Variable `max_dist_from_home` erzeugt einen Kreis mit dem angegebenen Radius um diese Koordinaten und dient der Sicherheit. Der Radius sollte den Bereich begrenzen, in

dem die Verbindungen noch kommunizieren können und das Flugzeug in Sichtweite ist. Sollte es passieren, dass ein Flugkörper ausserhalb des Bereichs gerät, so steuert der Autopilot das Flugzeug zurück zu der Home-Position, kreist um den Punkt und wechselt in den Home-Modus. Um wieder in den Auto2-Modus zu gelangen, klickt man im GCS auf die jetzt rote Fläche mit der Beschriftung HOME.

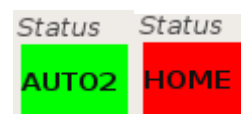


Abbildung 13-5: Modus

- Waypoints

```
<waypoints>
  <waypoint name="HOME" x="0" y="0"/>
  ...
  <waypoint alt="439" name="AF" x="200" y="-10"/>
</waypoints>
```

Die Wegpunkte müssen immer einen Name aufweisen, der anschliessend auch auf der Karte im GCS angezeigt wird. Sinnvollerweise gehören auch die X und Y Koordinaten relativ zur HOME-Position dazu. Eine Höhe (alt) kann optional dazu gegeben werden.

- Sektoren

```
<sectors>
  <sector name="Square">
    <corner name="_1"/>
    <corner name="_2"/>
    <corner name="_3"/>
    <corner name="_4"/>
  </sector>
</sectors>
```

Um einen Sektor zu definieren muss man die Eckpunkte der Fläche angeben. Die Corner referenzieren dabei auf definierte Waypoints. In diesem Beispiel müsste also unter dem Tag <waypoints> die Punkte _1, _2, _3 und _4 vorhanden sein.

- Exceptions

```
<exception cond="10 > PowerVoltage()" deroute="go_down"/>
<exception cond="(ground_alt+10 > estimator_z)" deroute="go_up"/>
<exception cond="(estimator_flight_time > 840)" deroute="quick_land"/>
```

Die Exceptions stellen Bedingungen dar, die periodisch geprüft werden. Trifft eine Bedingung zu, so kann der aktive Block des Flugplans geändert werden. Als Beispiel sind drei Bedingungen aufgeführt. In der zweiten Zeile wird geprüft, ob eine Mindesthöhe unterschritten wird. Trifft dies zu, so wird das Flugzeug in die Routine „go_up“ geleitet.

- Blocks

sind der Hauptteil des Flugplans. Sie verknüpfen die einzelnen Wegpunkt mit einander und legen die Route des Flugzeuges fest.

```
<block name="descent" key="D" strip_button="Descent" strip_icon="descent.png"
  pre_call="function_to_call_before()" post_call="function_to_call_after()">
  <circle wp="HOME" throttle="0.0" pitch="-15" vmode="throttle"/>
</block>
```

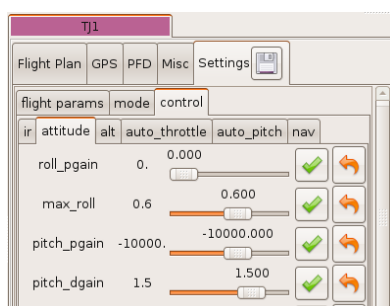
Dieses Beispiel zeigt ein Block, der bei der Ausführung zuerst eine andere Funktion aufruft. Anschliessend kreist er um den angegebenen Punkt, stellt den Motor ab und sinkt mit einem Pitch von -15°. Nach Beendigung wird wiederum eine andere Funktion aufgerufen. Um weitere Informationen über die Möglichkeiten der Programmierung zu erhalten, empfiehlt sich ein Blick in das Wiki:

http://paparazzi.enac.fr/wiki/Flight_Plans#Blocks

Mit den Attributen `strip_button` und `strip_icon` kann eine Schaltfläche im GCS programmiert und angezeigt werden.

13.3 Settings

```
<dl_settings NAME="attitude">  
  <dl_setting MAX="25000" MIN="000" STEP="250" VAR="h_ctl_roll_pgain" shortname="roll_pgain" module="fw_h_ctl"/>  
  <dl_setting MAX="1" MIN="0" STEP="0.05" VAR="h_ctl_roll_max_setpoint" shortname="max_roll" param="H_CTL_ROLL_MAX_SETPOINT"/>  
  <dl_setting MAX="000" MIN="-25000" STEP="250" VAR="h_ctl_pitch_pgain" shortname="pitch_pgain" param="H_CTL_PITCH_PGAIN"/>  
  <dl_setting MAX="2" MIN="0." STEP="0.1" VAR="h_ctl_pitch_dgain" shortname="pitch_dgain" param="H_CTL_PITCH_DGAIN"/>  
  <dl_setting MAX="5000" MIN="0" STEP="100" VAR="h_ctl_elevator_of_roll" shortname="elevator_of_roll" param="H_CTL_ELEVATOR_OF_ROLL"/>  
  <dl_setting MAX="5000" MIN="0" STEP="100" VAR="h_ctl_aileron_of_throttle" shortname="aileron_of_throttle"/>
```



In den Settings wird konfiguriert, was im GCS für Tuning Faktoren vorhanden sein sollen. Als Beispiel werden die Einstellungen der Fluglage von der Datei `tuning.xml` gezeigt. Man definiert mit dem XML-TAG `dl_settings` einen neuen Tab. Um den Tab `attitude` im Tab `control` zu platzieren muss die Struktur wie folgt aussehen:

```
<dl_settings Name="control">  
  <dl_settings Name="attitude">
```

Abbildung 13-6: Settings im GCS

```
    <dl_setting MAX.../>
```

```
  </dl_settings>
```

```
</dl_settings>
```

Die Einstellung des `roll_pgain` bekommt ein Maximum, Minimum sowie eine Step zugewiesen. Zusätzlich wird sie mit der Variabel `h_ctl_roll_pgain` der Datei `fw_h_ctl.c` verknüpft.

13.4 Radio

In dieser Datei werden die Kanäle der Fernsteuerung den Funktionen (roll, pitch, ...) zugewiesen. Dazu werden die Extrem- und Neutralstellungen der Servomotoren angegeben.

13.5 Telemetry

Um den Datalink zu konfigurieren wird die Konfigurationsdatei verwendet. Man kann angeben in welchem Intervall ein bestimmter Wert gesendet werden soll.

14 GCS

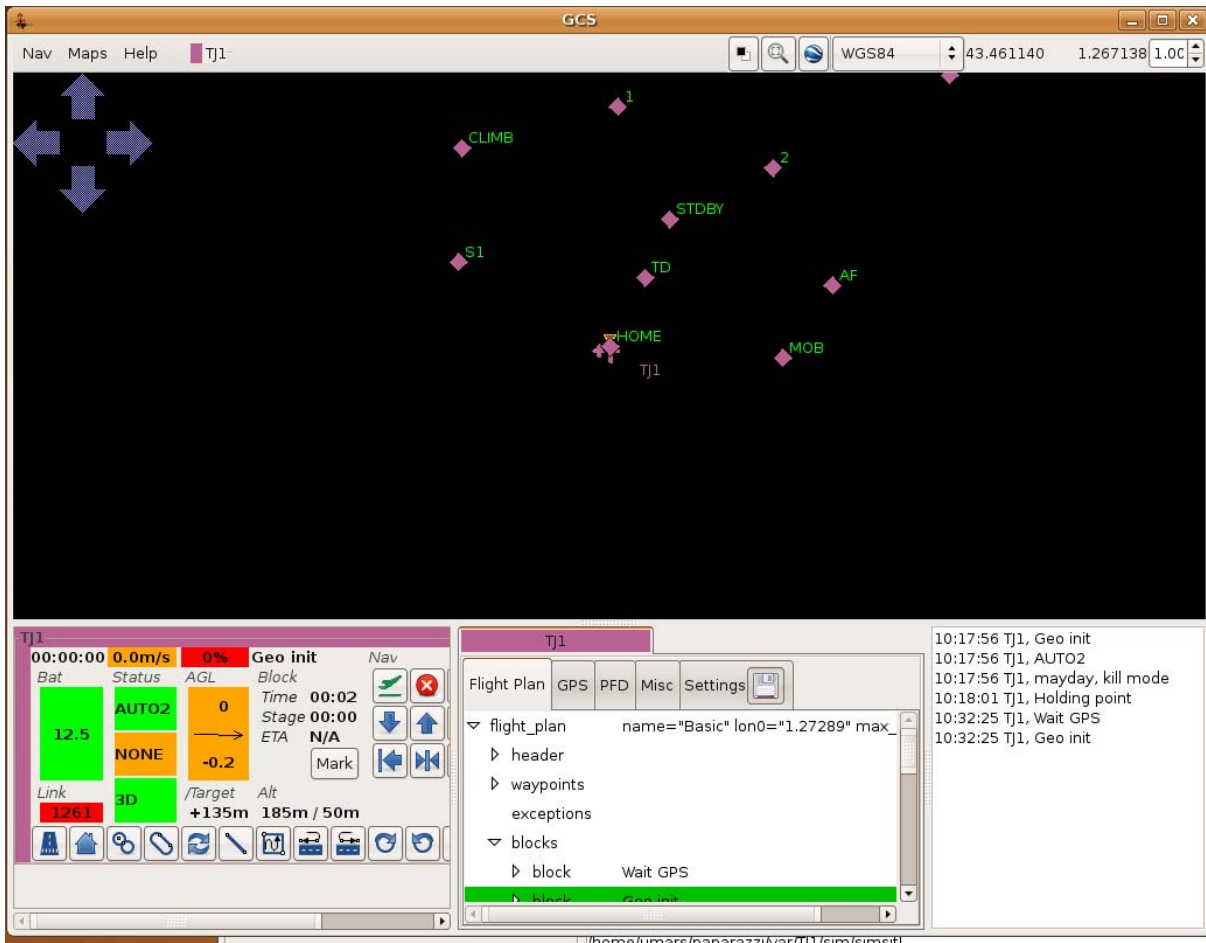


Abbildung 14-1: GCS-Fenster

Das GCS-Fenster ist in fünf Bereiche aufgeteilt. Unter der Menüleiste ist die Karte zu finden. Und am Unteren Rand ist von links nach rechts das Überwachungsfeld, die Einstellungen und die Meldungsanzeige angeordnet. Um das GUI selber zu konfigurieren oder zu ergänzen kann in dem Ordner `~/paparazzi/conf/gcs` eine Datei bearbeitet respektive erstellt werden. Für den Aufruf einer speziellen Konfiguration kann im Paparazzi Center der Text für den Start des Prozesses vom GCS angepasst werden.

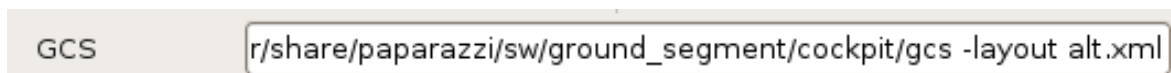


Abbildung 14-2: GCS mit Höhenangabe aufrufen

Um selber ein Layout zu erstellen, empfiehlt es sich das GCS zu öffnen, vom Message-Fenster die gewünschte Variable in den Karten-Bereich des GCS ziehen, Variable

formatieren und Layout über das Nav-Menü speichern. Beim nächsten Aufruf muss man dann nur noch das Layout-Atributt wie in Abbildung 14-2 einstellen.

14.1 Menu-Leiste

14.1.1 Nav

Im ersten Menü könne diverse Anzeige Optionen ein- oder ausgeschaltet oder die Karte angepasst werden. Die interessanten, nicht selbst erklärenden Funktionen werden im Folgenden erläutert.

- SRTM (Shuttle Radar Topography Mission)
Diese Option ermöglicht das Hinterlegen der Erdoberfläche in Meter über Meer. Dazu muss man den entsprechenden ZIP-Ordner in den Pfad `~/paparazzi/data/srtm` kopieren. Die verpackten Daten dafür findet man auf der nachfolgenden Internetseite: http://dds.cr.usgs.gov/srtm/version2_1/SRTM3/
Versucht man die Option zu aktivieren, ohne dass das entsprechende ZIP-Paket im Verzeichnis zu finden ist, so gibt die erscheinende Fehlermeldung Aufschluss über die fehlenden Koordinaten. Nach dem erfolgreichen Einschalten der SRTM-Option wird am linken Rand der Menu-Leiste, zusätzlich zu den Mauskoordinaten auch die dazugehörige Höhe in Meter über Meer angegeben. Ist die Option aktiviert, so erkennt die Simulation einen allfälligen Absturz durch einen Programmierfehler im Flugplan.
- Background
Im Background verstecken sich unter Anderem die hinterlegte Karte. Mit dieser Funktion kann also die Karte aus respektive eingeblendet werden. Um sie zu laden benötigt man allerdings das Menu Maps.
- Save Layout
Hat man sich das GUI zu recht geschoben oder angepasst, so kann mit dieser Funktion die momentane Konstellation des GCS abgespeichert werden. Zum Aufrufen des gespeicherten Standes muss man jedoch beim Starten vom GCS das Attribut layout einstellen. Siehe dazu Abbildung 14-2.

14.1.2 Maps

Um eine Karte in das GCS zu laden gibt es die Möglichkeit, dass man das Kartenmaterial von Google einbindet. Dabei muss man allerdings darauf achten, dass man nicht eine zu grosse Datenmenge anfragt, denn sonst wird der Dienst für die nächsten 24 Stunden durch Google gesperrt. Da Google jedoch ihr Protokoll geändert haben, funktioniert diese Funktion

momentan nicht. Abhilfe kann eine eigene Karte sein. Dazu benötigt man eine Bild mit zwei Punkten, von denen man möglichst genau die Koordinaten im WSG84 Format kennt. Dazu muss man eine XML-Datei schreiben, die festlegt welches Pixel zu welcher Koordinate gehört. Anschliessend verwendet man die Funktion Maps/load und wählt die gewünschte XML-Datei an.

```
<map file="winti.jpg" projection="Mercator">
  <point x="258" y="175" geo="WSG84 47.497595 8.728772"/>
  <point x="484" y="375" geo="WSG84 47.496872 8.729987"/>
</map>
```

14.2 Karte

Die Karte kann mit den Pfeiltasten der Tastatur oder den blauen Pfeilen geschoben werden. Zum Zoomen benötigt man das Mausrad oder die PgUp respektive PgDn Tasten. Die aktuellen Koordinaten der Maus können rechts in der Menuleiste abgelesen werden.

14.3 Überwachungsfeld

Das Überwachungsfeld, auch aus Notizblock bezeichnet, zeigt die wichtigsten Flugdaten an. Die Felder sind mit einer Farbe hinterlegt, die den Status signalisieren. Dabei gilt das Farbkonzept der Ampel (grün, orange, rot). Wird die Maus über einer Anzeige stehen gelassen, so erscheint eine Notiz mit einer Erklärung über die dargestellte Variable. Die Tasten am unteren Rand wurden im Flugplan programmiert und dienen zum Aufrufen der Flugblöcke. Die neun Tasten mit dem Titel Nav dienen der Steuerung des Flugzeuges. Die Erste Zeile steuern den Motor, mit der Mittleren kann auf die Flughöhe Einfluss genommen werden und mit der untersten Zeile kann der Durchmesser der Standartkreise verändert werden.

14.4 Einstellungen

Die Tabs dieses Feldes lauten der Reihe nach Flight Plan, GPS, PFD, Misc und Settings.

Im Flug Plan kann der aktuell ausgeführte Block – grün hinterlegt - angeschaut werden oder jeder andere Block gestartet werden. Der nächste Tab, GPS gibt grafisch über den Empfang und die zu erwartende Genauigkeit des GPS-Signals Auskunft. PFD steht für primary flight display und ist die Abkürzung für den künstlichen Horizont, welcher auch in Flugzeugen Auskunft über die Lage des Fliegers gibt. Im Feld Misc wird die berechnete Windgeschwindigkeit und –Richtung angezeigt. Dazu gibt es ein Feld mit der durchschnittlichen Geschwindigkeit und der daraus berechneten Zeit bis zur Home-Position. Der letzte und wichtigste Tab für das Tuning sind die Settings. Hier können die festgelegten Parameter justiert werden.

14.5 Meldeanzeige

Im Feld rechts werden die wichtigsten Meldungen ausgegeben. Dazu gehören die Angaben über den auszuführenden Flugplan-Block, Warnungen des Batteriestandes und Annäherung an den Boden.

15 Anhang

- 1) Zeitplan PA
- 2) Anforderungsliste PA
- 3) Anforderungsliste BA
- 4) Presentation Workshop
- 5) Paparazzi User Manual
- 6) XBEE Setup XCTU
- 7) Servo2PPM_Encoder Manual
- 8) Entwicklung und Erprobung eines Autopiloten mit inertialer Messeinheit
- 9) UtahStateUniversity_UAV_Project

15.1 Abbildungsverzeichnis

Abbildung 1-1: Umars.....	6
Abbildung 3-1: Überblick Paparazzi System	10
Abbildung 3-2: Komponente des Paparazzi Systems	10
Abbildung 3-3: Blockschaltbild Paparazzi	11
Abbildung 4-1: Schema Tiny V2.11	12
Abbildung 4-2: Ein- und Ausgänge des Tiny V2.11	13
Abbildung 4-3: Komponente des Tiny V2.11	14
Abbildung 5-1: Funktionsweise der IR-Sensoren	15
Abbildung 5-2: Horizontaler IR-Sensor.....	15
Abbildung 5-3: Funktionsprinzip Gyro	15
Abbildung 5-4: 6 DOF	16
Abbildung 6-1: Schnittstellen Signal.....	17
Abbildung 6-2: Modifikation an Futaba FASST 7	18
Abbildung 6-3: Summensignalempfänger	18
Abbildung 6-4: Encoder	19
Abbildung 9-1: Verbindungskabel serielle Kommunikation	24
Abbildung 9-2: USB to Picoblade Kabel.....	24
Abbildung 9-3: Anschlüsse von XBee Modem und Autopilot	25
Abbildung 9-4: Modem	25
Abbildung 9-5: Bodenstations Modem	26
Abbildung 9-6: Servokabel	26
Abbildung 10-1: Servokabel	27

Abbildung 10-2: Aufbau zum Flashen des AP	28
Abbildung 10-3: Aufbau für Kommunikation mit AP	29
Abbildung 10-4: Modem Bodenstation	30
Abbildung 10-5: Modem Flugstation.....	30
Abbildung 10-6: XCTU GUI	30
Abbildung 10-7: Meldung für funktionierende Kommunikation.....	31
Abbildung 10-8: Modem Konfiguration	31
Abbildung 10-9: Aufbau für Kommunikation via Funk	32
Abbildung 11-1: Paparazzi Center	33
Abbildung 11-2: Data Link	34
Abbildung 11-3: Gaia.....	34
Abbildung 11-4: Flight Plan Editor.....	35
Abbildung 11-5: Log File Player	35
Abbildung 11-6: Log Plotter.....	36
Abbildung 11-7: Messages.....	36
Abbildung 11-8: Real-time Plotter	37
Abbildung 11-9: Simulator	37
Abbildung 13-1: airframe Struktur	40
Abbildung 13-2: Funjet	40
Abbildung 13-3: Achsen	41
Abbildung 13-4: Batterie Niveau.....	44
Abbildung 13-5: Modus	46
Abbildung 13-6: Settings im GCS.....	48
Abbildung 14-1: GCS-Fenster	49
Abbildung 14-2: GCS mit Höhenangabe aufrufen.....	49

15.2 Tabellenverzeichnis

Tabelle 7-1: Distrelec Bestellung.....	20
Tabelle 7-2: Conrad Bestellung.....	21
Tabelle 7-3: Übersicht AWG Grössen	21
Tabelle 9-1: Verbindungen zwischen XBee Modem und Autopilot	25