Fachhochschule Nordwestschweiz
Hochschule für Technik

Drohne für Landwirte

Projekt 5: Bericht

Projektmitglieder	Haberkorn Jonas Schellhorn Antoine
Projektnummer	EIT-HS17-_____ *(noch nicht bekannt)*
Auftraggeber	Nouri T., Prof. Dr. Dip. Eng. Elec./Phys.
Studiengang	Bachelor Elektro- und Informationstechnik – I.C.S.
Datum	Mittwoch, 4. Oktober 2017

Zusammenfassung

Die Verbreitung von Drohnen für verschiedene Anwendungsbereiche und Preisklassen nimmt laufend zu. Die Drohnen, mit eingebauter Kameratechnik, als modernes „Spielzeug“ oder für Industrielle Anwendung, werden ständig erweitert. Drohnen sind ein wesentliches Element der Präzisionslandwirtschaft sowie von „Smart Farming“-Konzepten. Ausgestattet mit speziellen Sensoren und Kameras liefern sie zeit- und ortsbezogen detaillierte Information zum Zustand von Pflanzen und Böden. Mit diesem Wissen können Landwirte ihre Ressourcen deutlich effizienter einsetzen. Unser Interesse an den vielfältigen Möglichkeiten der Innovation in diesem Bereich haben uns motiviert, dieses Projekt zu wählen.

Als dies ein komplexes Bereich der Technologie ist, erfolgt das gesamt Ziel dieses Semesterprojekt: ein Betriebsbereit Hexakopter mit vollständigeren Hardware und geeigneter Software zu bauen. Die gesammelten Erfahrungen während dieses Projekt des 5. Semester, werden uns erlauben, ein effizienter Arbeit für unsere Thesis zu realisieren.

Keywords: Drohne, UAV, Hexakopter, Multikopter, ArduCopter, Ardupilot, APM, Fernsteuerung, GPS, Mission Planner, Flight Plan, Flug Simulation

Inhaltsverzeichnis

Inhaltsverzeichnis **2**

1. Einleitung **4**

2. Hardware & Anlagen **5**
2.1 Hexakopter 5
2.1.1 Hexakopter und Quadkopter Vergleichung 5
2.1.2 HMF S550 5
2.1.3 ArduCopter 6
2.1.4 Fernbedienung und Empfänger 7
2.1.5 Motoren und ESC 7
2.1.6 GPS 8
2.1.7 Stromverteiler 9
2.1.8 Batterie und Ladegeräte 10
2.1.9 Plattform für GPS und APM 10
2.1.10 Power Modul 12
2.2 Antrieb 12

3. Software **14**
3.1 Eingänge lesen und Ausgänge steuern - mit dem Code 14
3.1.1 Lesung eines Eingangswert 14
3.1.2 Steuerung eines Ausgangs Pin 14
3.2 Eingänge lesen und Ausgänge steuern - mit Mission Planner 15
3.3 Flight Plan 16
3.4 Flugsimulation mit Mission Planner 17
3.5 Warum den Code mit Linux auf den APM zu Laden 17

4. Tests **19**
4.1 Betrieb der Drohne 19
4.2 Ausgänge Bedienung 20
4.2.1 Relaisausgang – Ein und Aus 20
4.2.2 Pulsweitenmodulation Ausgang - PWM 21
4.3 Abflug, Landung und Flight Plan Simulation 23

5. Reflexion **25**

6. Schlusswort **26**

7. Ehrlichkeitserklärung **27**

Literaturverzeichnis **28**

Abbildungsverzeichnis **29**

Tabellenverzeichnis **30**

A. Programmcode **31**

a. UserCode.pde 31

b. UserCode.h 33

B. Fertigungsdaten Mechanik **34**

a. GPS / APM Plattform 34

Projektausschreibung Originalversion P5/P6 ..... **35**

Plichtenheft **37**

Benutzermanual **52**

1. Einleitung

Die Landwirte benutzen neue Technologien um ihrer Bewirtschaftung effizienter zu verwalten. Heutzutage wenden sich die Landwirte immer mehr zu den Drohnen um ihren Anforderungen zu beantworten. Uns wurde das benötigte Material zum Aufbau und Steuerung für eine Drohne zur Verfügung gestellt.

Hexakopter können verwendet werden, um Düngemittel auf Gemüse, Pflanzen, Bäumchen usw. zu verteilen. Diese Aufgabe kann auch von einem Drohnenschwarm realisiert werden. Um ein solches Projekt durchzuführen, muss man zuerst, der Betrieb eine Drohne beherrschen.

Die Endziele unser 5. Semester Projekt sind:

- Eine Drohne aufbauen (Grundplatte, Elektronik, Speisung, Fernbedingung, Antrieb, usw.).
- Betrieb der Softwareumgebung beherrschen.
- Das Steuerungsprogramm erfassen und Änderungsmöglichkeiten des Programms einarbeiten.
- Tests durchführen für die Inbetriebnahme bereit zu sein.
- Ein- und Ausschaltung eines L.E.D. auf dies Ausgängen des Flight Controller mit der Fernbedienung kontrollieren zu können.
- Automatik Abflug und Landung der Drohne
- Mit der Drohne eine Liste von vordefinierte Punkte überfliegen

Dieses Semester wird zuerst eine Drohne gebaut und seine Hardware bemeistert um eine Weiterentwicklung zu erleichtern. Deshalb haben wir Hardware, Software und Steuerungsmethoden der Drohne studiert um ein Benutzermanual zu erzeugen. Im Folgenden finden Sie eine Zusammenfassung unseren Forschungen und Arbeiten.

2. Hardware & Anlagen

Für dieses Projekt haben wir verschiedenes Hardware benötigt, wir werden es hier unten präsentieren.

2.1 Hexakopter

Hier werden wir die Wahl der Drohne erklären und die verschiedenen Komponenten präsentieren.

2.1.1 Hexakopter und Quadkopter Vergleichung

Es gibt mehrere Vorteile eines Hexakopter-Rahmens gegenüber einem Quadkopter-Rahmen. Zum Beispiel gibt es natürlich mehr Motoren mit einem Hexakopter, die mehr Leistung bedeuten, also größere Batterien. Diese Zunahme an Batterien führt zu einer Erhöhung der Flugzeit, was auch ein Vorteil ist. Aufgrund dieser zusätzlichen Motoren ist ein Hexakopter auch stabiler als ein Quadkopter. Wenn ein Quadkopter aus irgendeinem Grund einen Motor verliert, sinkt er ab. Mit einem Hexakopter können Sie in den meisten Fällen sicher mit einem oder sogar zwei beschädigte Motoren landen.

Mit 6 Motoren, müssen auch 6 Ausgang Kanal auf den Kontroller zu Verfügung sein, wo ein Quadkopter nur 4 benötigt.

Zum Schluss, hat der Hexakopter all die guten Dinge, die Quadkopter haben, zusätzlich zu mehr Kraft und Hubkapazität.

2.1.2 HMF S550

Als Gehäusegestell für unsere Hexakopter haben wir der Happymodel „HMF S550“ Rahmen benutzt. Diese Rahmen sind größtenteils aus einem robusten Kunststoff gefertigt, der ihn haltbar macht, aber auch leicht ist. Eine eingebaute Stromverteilungsplatine ist auch ein Schlüsselmerkmal dieses Rahmen, das bedeutet, dass wir die ESCs einfach direkt auf die Zentralplatte löten können.

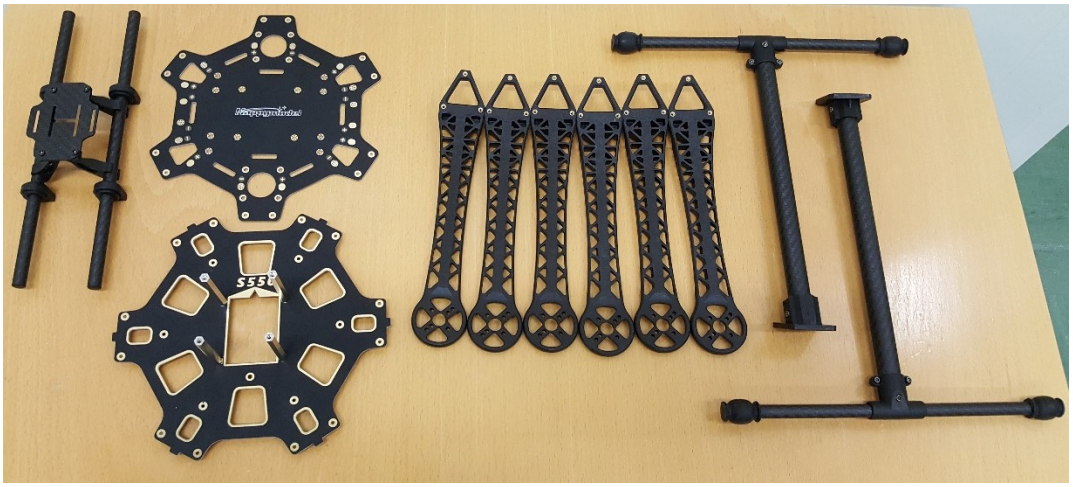


Abbildung 1: *Happymodel HMF S550*

2.1.3 ArduCopter

Als Steuerungshardware haben wir ein „ArduPilot Mega 2.5“ mit den Software ArduCopter benutzt, es ist ein Remote Control spezialisierte Arduino Mega 2560. Die APM 2.5 ist eine Open-Source-Flugkarte. Es wird ständig weiterentwickelt und bietet eine sehr breite Palette von Geräten, Funktionen und Einstellungen. Zunächst bietet das APM eine große Auswahl an Stützen, RC-Car, Flugzeug, Hubschrauber, Trikopter, Quadkopter, Hexakopter und Octokopter. Es ermöglicht Pilotenassistenz, Flugsicherheit, Hinzufügen von Features und kann durch Programmierung autonom werden. Das APM hat unter anderem ein integrierter Kompass.

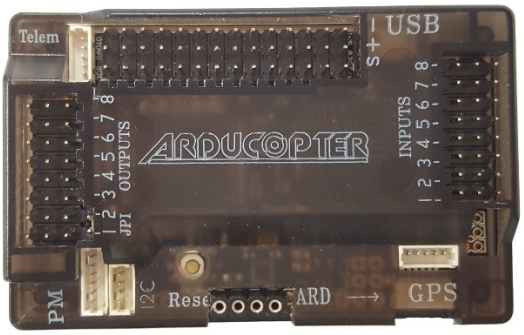


Abbildung 2: *ArduPilot Mega 2.5*

2.1.4 Fernbedienung und Empfänger

Als Fernbedienung haben wir die Graupner MX-12 HoTT benutzt. Diese Fernbedienung hat 6 Steuerfunktionen und ermöglicht das Steuern zahlreicher Modelle. Als Empfänger benutzen wir der Graupner GR-12 HoTT, es ist ein Empfänger mit 6 Kanälen für Flugzeuge und Helikopter. Mit der HoTT-Technologie verfügt er über ein neues Hopping Telemetry Transmission Hochfrequenzteil mit modernsten Komponenten und modernster Software. Ohne Zusatz-Sensorik werden die Parameter Empfängerspannung und -temperatur sowie Signalstärke übermittelt. Der Einsatz der Graupner HoTT 2,4 GHz Übertragungstechnologie gewährleistet hohe Funktionssicherheit durch bidirektionale Kommunikation zwischen Sender und Empfänger mit integrierter Telemetrie und ultraschnellen Reaktionszeiten.

Sobald die Fernbedienung eingeschaltet ist, versucht sie, eine Verbindung mit dem Empfänger herzustellen. Solange die Verbindung nicht hergestellt ist, piepst die Fernbedienung.

Abbildung 3: *Fernbedienung Graupner GR-12 HoTT*

2.1.5 Motoren und ESC

Als Motoren benutzen wir sechs „RCX BC2836-9“ die 880KV Brushless Motoren sind. Sie wiegen nur 70g und haben eine maximale Leistung von 243W und ein maximaler Zug von 1,1 Kg. Sie können mit 12x6, 10x7 oder 9x6 Propeller benutzen werden. Der Anschlüsse durch drei Kabeln ist sehr einfach (Rot für positiv, Schwartz für negativ und Blau für den Betrieb Signal).

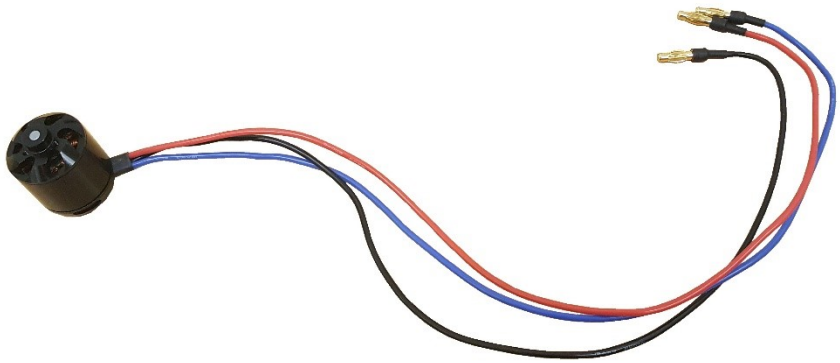


Abbildung 4: *Motor RCX BC2836-9 Brushless*

Für die ESC (Electronic Speed Controller) haben wir die 20Amp von 3D Robotics genutzt. Dies sind für brushless Motoren geeignet.

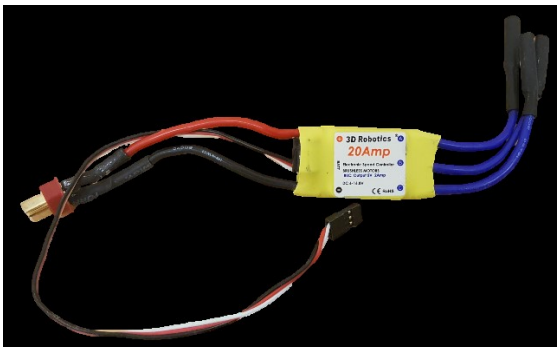


Abbildung 5: *ESC 3D Robotics*

Wenn die Motoren mit die ESCs angeschlossen sind und danach mit Strom versorgt, ertönt ein Signalton, um das Einschalten zu signalisieren.

2.1.6 GPS

Als GPS benutzen wir ein „M8N“ GPS der eine Kompass enthält. Er hat punktgenau einer Navigationsempfindlichkeit von –167 dBm (was sehr gut ist), und ist einfach zu einstellen.

Abbildung 6: *M8N GPS*

2.1.7 Stromverteiler

Unser PDB (Power Distribution Board), die E-Max 6 Axis, ist geeignet für ein Hexakopter und sehr praktisch zu benutzen. Wir haben anderen Anschlüsse gelötet um leichter auf diese zugreifen zu können, um es mehr handlich zu machen (siehe Abbildung 8).

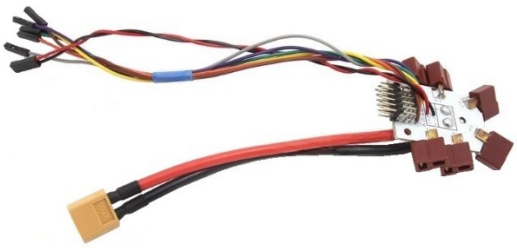


Abbildung 7: *Power Distribution Board*
Quelle: https://www.kooltoyz.co.uk/

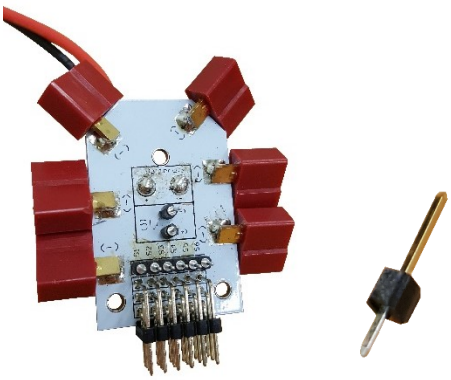


Abbildung 8: *neuen Pins auf der PDB gelötet*

2.1.8 Batterie und Ladegeräte

Wir benutzen ein Orion 4000mAh 1.1V Lithium Polymer Batterie. Die hat eine lange Lebensdauer und bringt die geeignete Leistung. Die Batterie hat eine eingebettete Ladezustandsanzeige was sehr praktisch ist.

Der Ladegeräte ist ein Turnigy Accucell 6, 50W Balance Ladegerät. Wir können dieses Ladegerät auch mit einem USB-Kabel an unseren PC anschließen, so dass wir während des Ladevorgangs über Echtzeitdiagramme Packungsspannung, Zellspannung und andere Daten überwachen können.

Abbildung 9: *Batterie und Ladegeräte*

2.1.9 Plattform für GPS und APM

Da wir auf der Drohne keinen Platz hatten, haben wir uns entschieden, eine erhöhte Plattform zu schaffen. Diese Plattform kann auf aerodynamischen Gründen nicht grösser als 8cm x 8cm sein. Da das APM mit Klettverschluss auf der Drohne hängt (wegen praktische Gründen), können Schrauben überschreiten (siehe Abbildung 12).

Die technische Zeichnung diese Plattform befindet sich in den Fertigungsdaten Mechanik.

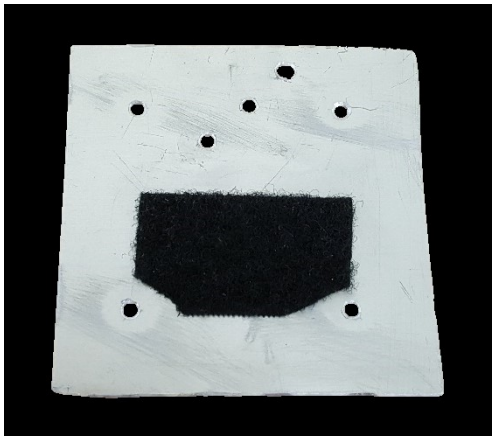


Abbildung 10: *Plattform für GPS und APM*

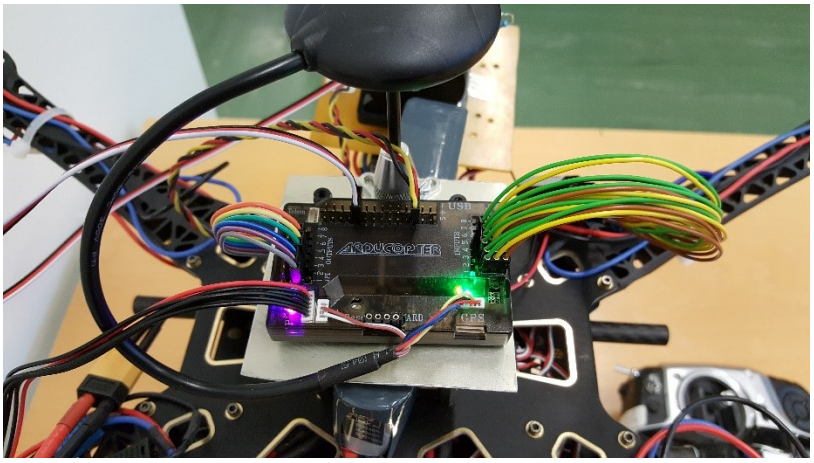


Abbildung 11: *draufsicht der Plattform*

Abbildung 12: *Vorderansicht der Plattform*

2.1.10 Power Modul

Um des APM und die Power Distribution Board in Strom zu versorgen, haben wir ein 3DR Power Module benutzt.

Damit wird das APM mit einem geeigneten Strom versorgt und den PDB kann den benötigten Strom auf die Batterie ziehen.

Abbildung 13: *Power Modul*

2.2 Antrieb

Dank unsere Code können wir verschiedenen Leistungsantriebe mit der ArduCopter steuern. Abbildung 15, können wir ausser die bereits zitierte Elementen, die Montage von eine LED und Servomotor auf den Ausgängen des APM sehen.

Wir benutzen eine LED und einen Servomotor, weil wir damit die zwei möglichen Ausgängen Typ des APM testen können (Relaisausgang und Pulsweitenmodulation Ausgang).

Als Servomotor benutzen wir der Digital «Karbonite Ultra Speed Servo» Hitec HS-6965HB, mit einer Reaktionszeit von 20ms und ein Gewicht von 50g ist er für Anwendungen auf ein Hexakopter geeignet.

Abbildung 14: *Servomotor Hitec HS-6965HB*
Quelle: https://servodatabase.com/images/servos/hitec-hs-6965hb.jpg

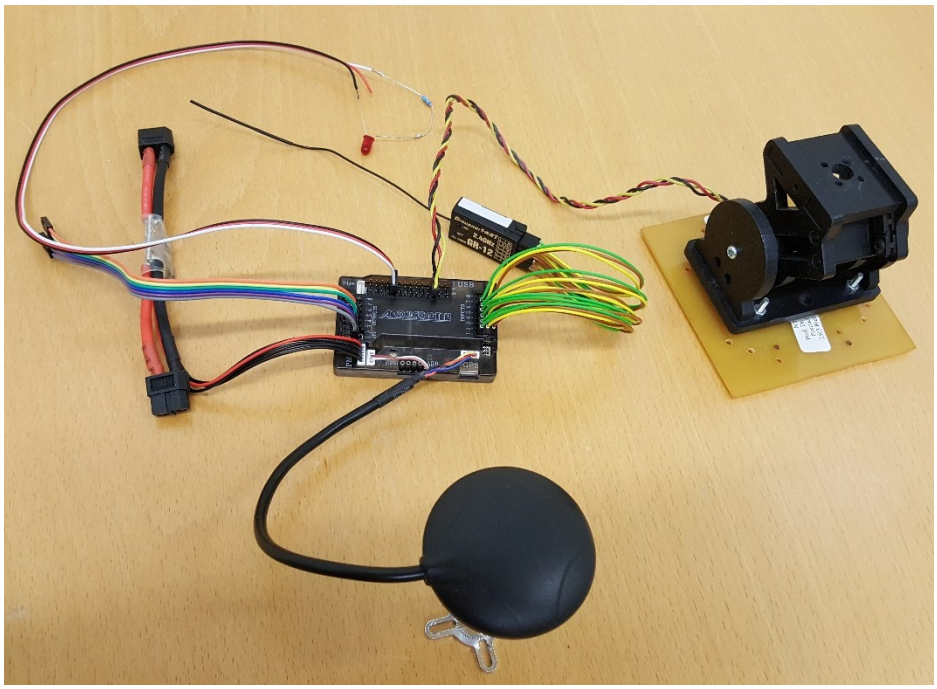


Abbildung 15: *gesamt Übersicht des Antriebes*

3. Software

Als Software haben wir Mission Planner und den ArduCopter-3.2.1 Quellcode benutzt. Den ArduCopter Code haben wir unten Linux geändert und auf dem Board geladen. Mehr Infos über der Installation und Benutzung sind im Benutzermanual zu finden.

3.1 Eingänge lesen und Ausgänge steuern - mit dem Code

Um, mit dem Code, Eingängen zu lesen und Ausgängen zu steuern, müssen wir unsere eigene Funktion schreiben. Dafür haben wir unseren Code in das „UserCode.pde“ Datei geschrieben. Es gibt 6 Default Funktionen die mit verschiedene Frequenz aufgerufen sind. Unseren Code haben wir in die schnellste Funktion geschrieben, die „userhook_FastLoop()“.

3.1.1 Lesung eines Eingangswert

Um ein Eingang zu lesen benutzen wir die folgende Anweisung:

$$int\ value = g.rc_6.radio_in;$$

Hier wird das Eingangswert für das 6.Kanal gelesen und in ein Variable zugewiesen.

3.1.2 Steuerung eines Ausgangs Pin

Um einen Ausgang zu steuern, benutzen wir die folgende Anweisung:

$$hal.gpio\text{->}write(LED_A4, HAL_GPIO_LED_ON);$$

LED_A4 und *HAL_GPIO_LED_ON* sind Integer Variablen. *HAL_GPIO_LED_ON* kommt von der Datei „GPIO.h“ in „ardupilot-ArduCopter-3.2.1\libraries\AP_HAL“ und ist Standardmässig genutzt als Zustand Variable für die LED (selbst wenn es nur ein Integer ist). Ebenfalls, existiert die Variable *HAL_GPIO_LED_OFF* auch, um die LED zu ausschalten.

LED_A4 haben wir in das „UserVariables.h“ Datei so definiert:

$$const\ int\ LED_A4 = 4+OFFSET_OUTPUT;\ //PIN\ A4$$

Wo auch OFFSET_OUTPUT so definiert ist:

$$const\ int\ OFFSET_OUTPUT = 54;\ //to\ set\ in\ OUTPUT\ Digital + 54$$

Um den Nummer des Ausgangs Pin zu kennen, haben wir die folgende Abbildung gemach.

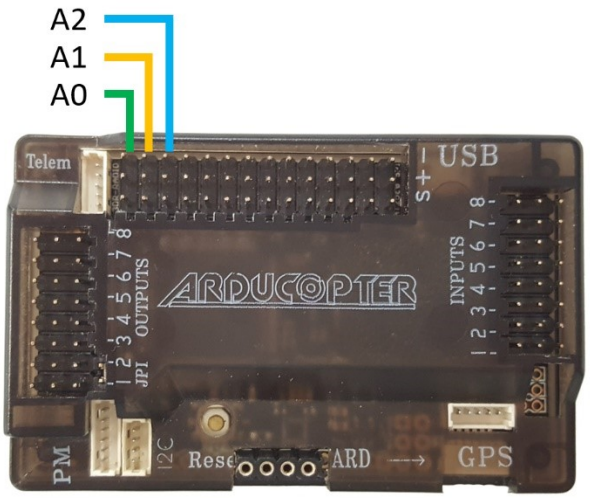


Abbildung 16: *APM 2.5 Ausgängen numerierung*

Das Offset von 54 sorgt dafür, dass das GPIO („General Purpose Input/Output“) als Ausgang benutzt wird (die GPIO können auch als Eingang genutzt werden, dafür wird das Pin Nummer normal in dem Code genutzt).

3.2 Eingänge lesen und Ausgänge steuern - mit Mission Planner

Um die Eingänge und Ausgänge des APM mit Mission Planner zu konfigurieren, muss das APM während die Änderungen an den Computer verbunden sein (siehe das Benutzermanual).

Mission Planner ändert die Variablen direkt auf das APM, deswegen ist es nicht möglich sogenannten „Profile“ im Voraus vorzubereiten. Um ein PWM Ausgang (Pulse Width Modulation) zu steuern soll darauf ein Eingang zugewiesen. Es gibt Ausgänge die standartmässig für PWM benutzt sind (zum Beispiel auf die Abbildung 17, die Pins A11).

Das einfachste ist die Konfiguration des Kamera Kontrolle zu ändern. Hier werden das Tilt, Roll und Pan mit PWM beherrschen, auch wenn es keine Kamera, sondern zum Beispiel ein Motor, der am Ausgang angeschlossen ist.

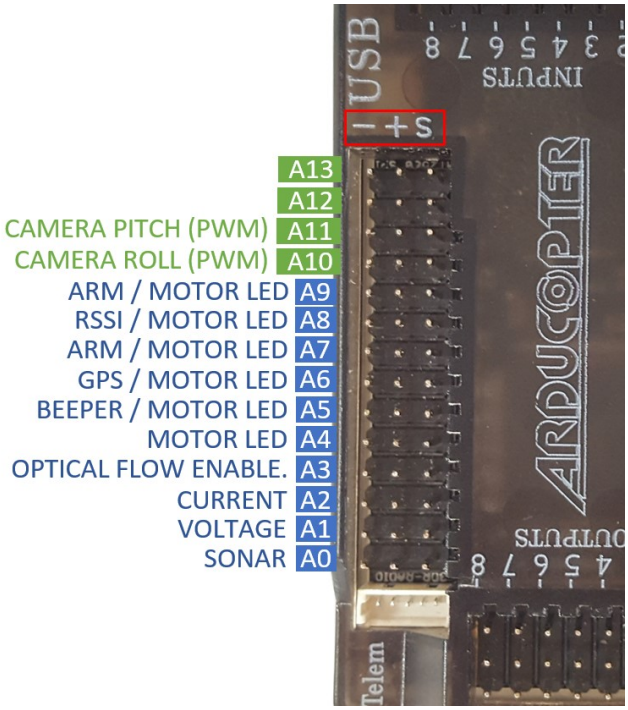


Abbildung 17: *Standardkonfiguration der Ausgänge*

3.3 Flight Plan

Mit Mission Planner ist es möglich Flight Pläne zu definieren (siehe das Benutzermanual). Dies können auch genutzt werden um bestimmte Aktion zu realisieren (zum Beispiel: abfliegen, landen, Geschwindigkeit ändern, Kamera einsteuern, und so weiter).

Beliebig viele Flight Pläne können auf den Rechner gespeichert werden aber nur ein auf den APM.

Wichtig zu betrachten ist, dass der Drohne eine gute Höhe hat während sein Flug damit er nicht abstürzt.

3.4 Flugsimulation mit Mission Planner

Wen man Änderungen an den ArduCopter Code durchführt, ist es gut dieses zuerst auf ein virtuelle Copter zu Probieren. Da Mission Planner nur die Firmware benutzt, braucht man das APM nicht für die Simulation.

Mit der Simulation, kann man zum Beispiel Flight Plan testen und das Verhalten der Drohne. Die Daten die angezeigt sind, sind genau dieselben als wenn es eine reelle Drohne wäre.

3.5 Warum den Code mit Linux auf den APM zu Laden

Zuerst haben wir mit Arduino für ArduCopter probiert. Trotzt die Bestätigung, dass das Firmware auf dem Board gut geladen wird, ist immer noch das APM mit der alten Version der Firmware programmiert.

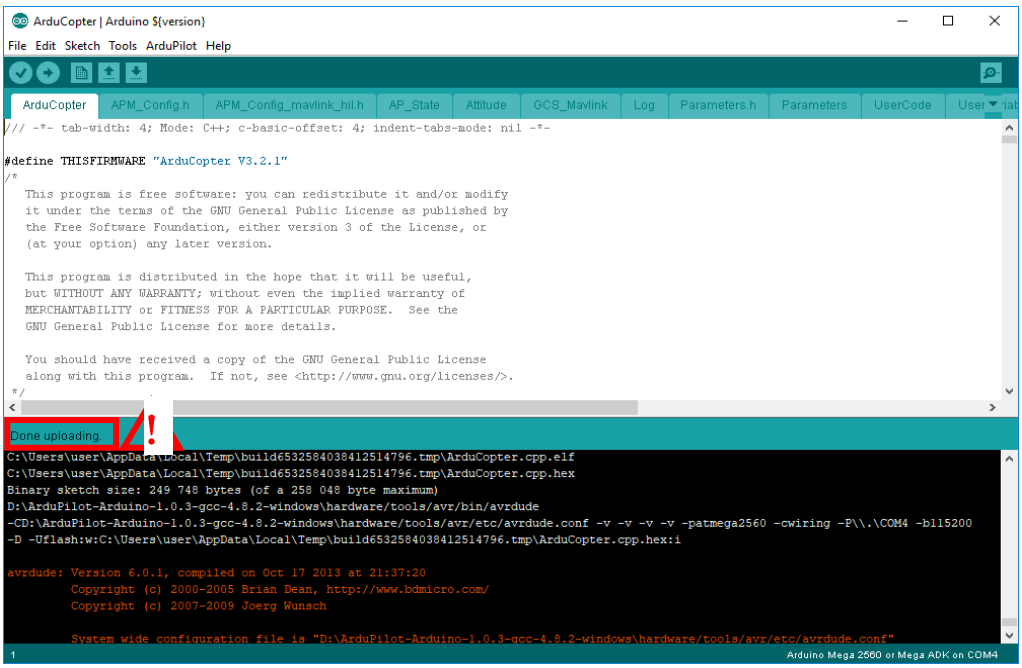


Abbildung 18: *falsche Bestätigung des Hochladung - Arduino*

Wir haben auch mit Mission Planner getestet. Dafür muss den Custom Firmware zuerst auf das APM geladen werden (siehe das Benutzermanual) und dann auf Mission Planner. Während dieses Teil muss das APM an dem Rechner angeschlossen sein aber nicht mit Mission Planner gebunden sein (siehe Abbildung 20). Dieses Firmware (.hex Datei) ist mit der Kompilierung auf Arduino für ArduCopter erzeuget und ist in das „Temp“ Folder gespeichert (siehe Abbildung 19).

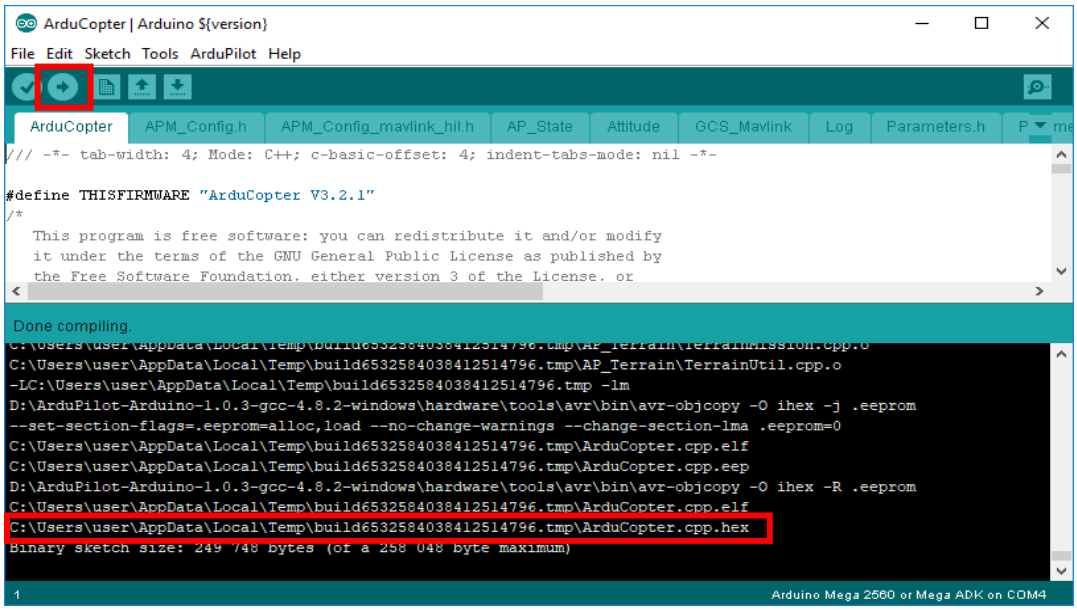


Abbildung 19: *Hochladung und Folder des .hex Datei - Arduino*

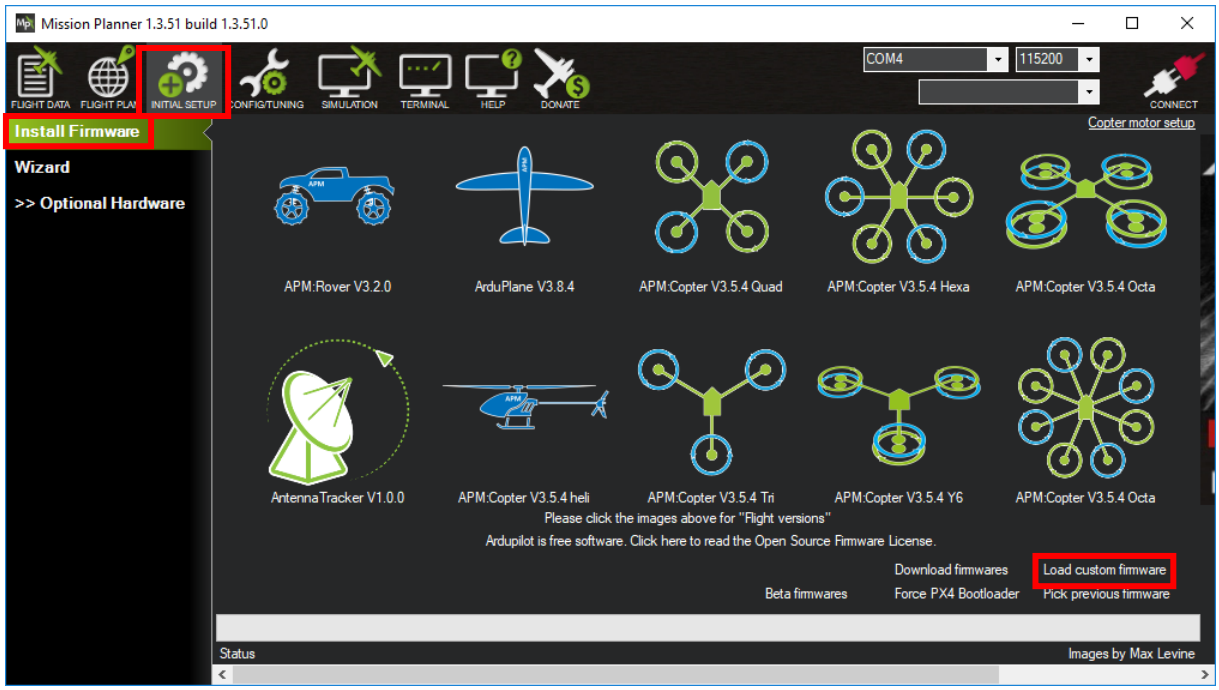


Abbildung 20: *Neue Firmware hochladen – Mission Planner*

Es kann sein das am Ende der Installation das Fehler «Upload succeeded, but verify failed» auftritt, das ist der Grund warum wir es mit Mission Planner nicht geschafft haben, auch wenn das Bitrate, Parity, u.s.w für das COM-Port richtig konfiguriert waren. Die Firmware war gut auf das Board geladen aber könnte nicht verifizieren werden, deswegen das APM hat noch die letzte Version und die neue war gelöscht.

4. Tests

Um zu prüfen ob unsere Ziele erreicht wurden, haben wir verschiedene Tests durchgeführt. Diese war sehr wichtig, um die Meilensteine unseres Projekts zu repräsentieren.

4.1 Betrieb der Drohne

Dieser Test war eine Verbindung von mehreren kleinen Teste die, alle zusammen das gesamt Betrieb der Drohne überprüft. Hier ist eine Tabelle mit den verschiedenen Komponenten die wir geprüft haben und wie wir es gemacht haben:

Komponenten	**Testmethode**
Fernbedienung und Empfänger	Wir haben der Empfänger an den APM angeschlossen und den APM mit Mission Planner verbunden. Im Menü „Radio Calibration“ haben wir geprüft ob sich die Eingänge ändern, wenn man die Fernbedienung nutzen.
GPS und Kompass	Wir haben den GPS an den APM angeschlossen und den APM mit Mission Planner verbunden. Im Menü „Filght Data“ haben wir auf die Karte geprüft ob die Position der Drohne übereinstimmt und ob die Orientierung richtig ist, auch wenn wir die Drohne drehen.
Stromversorgung des APM und den Motoren	Wir haben die geladene Batterie auf dem Leistungsmodul angeschlossen. Die Motoren müssen auf den ESC angeschlossen sein, und die ESC an das Leistungsmodul. Das Leistungsmodul versorgt das APM und das Stromverteiler. Wenn die LED des APM leuchten, dann ist das APM gut in Strom versorget (Das USB Kabel soll auf des APM nicht angeschlossen sein). Der Stromverteiler wird dann die ESC mit Strom versorgen, wenn die Motoren ein Signalton machen dann ist es gut versorgt.
Motoren Bedienung	Dafür haben die Kalibration des ECS die im Benutzermanual erklärt sind, gefolgt. Wenn sich die Motoren drehen, dann funktioniert die Bedienung.

Tabelle 1 : *Testmethoden für verschiedene Komponenten*

4.2 Ausgänge Bedienung

Um die Ausgänge Bedienung zu testen, haben wir zwei Test Geräten benutzt, weil wir zwei Arten von Ausgängen haben.

4.2.1 Relaisausgang – Ein und Aus

Mit Relaisausgänge sind nur zwei Werten in Ausgang möglich: Ein und aus (0V und 5V). Dafür haben wir ein Trigger Werte definiert. Wenn die 6. Eingänge Kanal grösser ist als dieser Wert, dann ist der Ausgang auf 5V gesetzt, sonst auf 0V. Um dies zu prüfen benutzen wir eine LED mit einem geeigneten Wiederstand die wir auf Kabeln zusammengelötet haben.

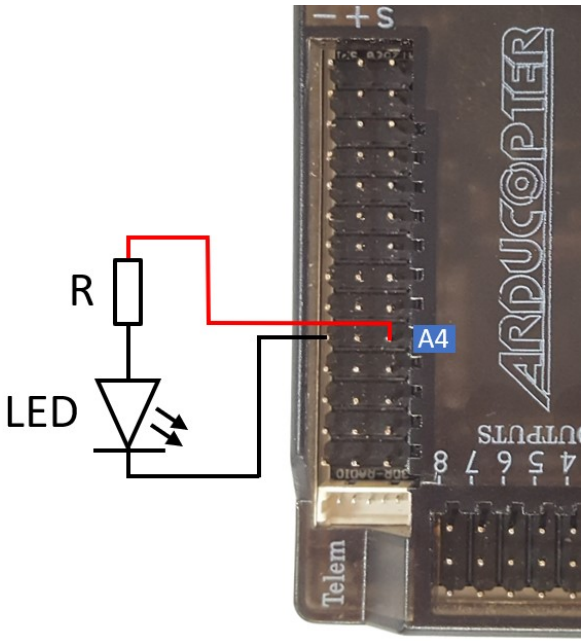


Abbildung 21: *Schaltplan des LED auf das APM*

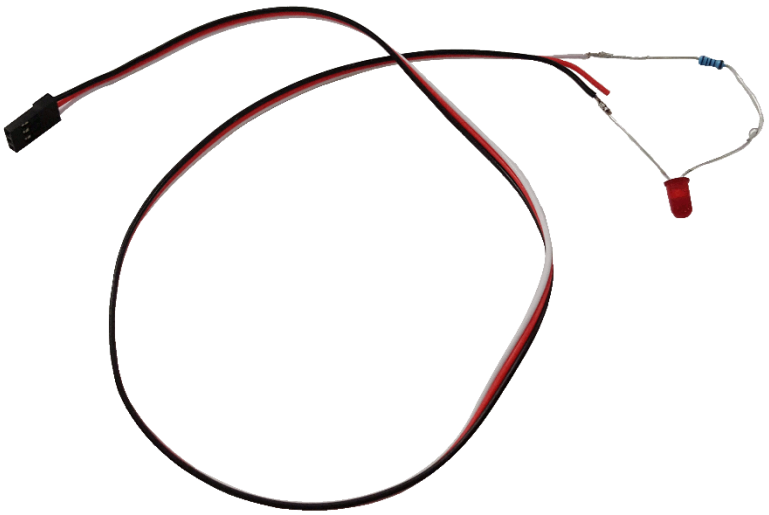


Abbildung 22: *Test Gerät für Relaisausgang*

Wenn die LED entsprechend dem Wert von Kanal 6 leuchtet oder nicht, funktioniert die Bedienung des Relaisausgangs.

Den Code der wir benutzt haben ist im Anhang zu finden.

4.2.2 Pulsweitenmodulation Ausgang - PWM

Mit Pulsweitenmodulation Ausgänge können wir verschiedene Geräten steuern (zum Beispiel: die Winkelposition oder die Rotationsgeschwindigkeit eines Motors). Wir haben der Servomotor HS-6965HB Hitec genutzt um seine Winkelposition mit die 6. Eingange Kanal zu bedienen. Dieser Servomotor war schon in einem Kamerasupport eingebaut aber den Betrieb ist das Gleiche.

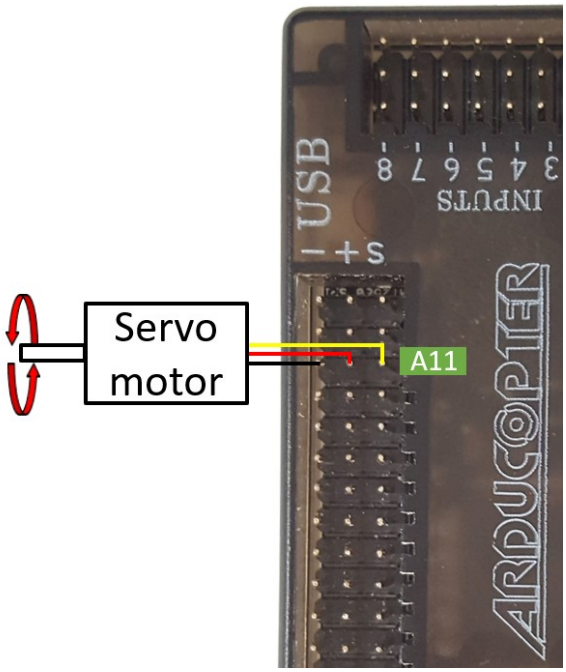


Abbildung 23: *Schaltplan des Servomotors auf das APM*

Abbildung 24: *Test Gerät für Pulsweitenmodulation Ausgang*

Wenn der Servomotor die Änderung des gewählte eingangs Kanal folgt, dann funktioniert die Bedienung des PWM Ausgang gut.

Den Code der wir benutzt haben ist im Anhang zu finden.

4.3 Abflug, Landung und Flight Plan Simulation

Aus Sicherheitsgründen könnten wir diese Tests mit der Drohne nicht durchführen, deswegen haben wir es mit Mission Planner simuliert.

Zuerst haben wir ein Flight Plan definiert (wie im Benutzermanual erklärt ist). Dann sind wir in das Menu „Simulation“ gegangen und haben die Firmware für den Hexakopter geladen.

Abbildung 25: *Start der Simulation*

Im Menü Action haben wir «Mission_Start» gewählt und dann folgt die Simulation den vordefinierte Flight Plan. Dafür benutzt er den automatischen Abflug von den Code ArduCopter (seine Höhe und anderen Informationen sind oben links zu sehen). Dann folgt die Simulation den Flight Plan, und am Ende haben wir einen Landung Punkt definiert, wo die Simulation die automatische Landung des ArduCopter Code benutzt.

Diese Simulation war erfolgreich.

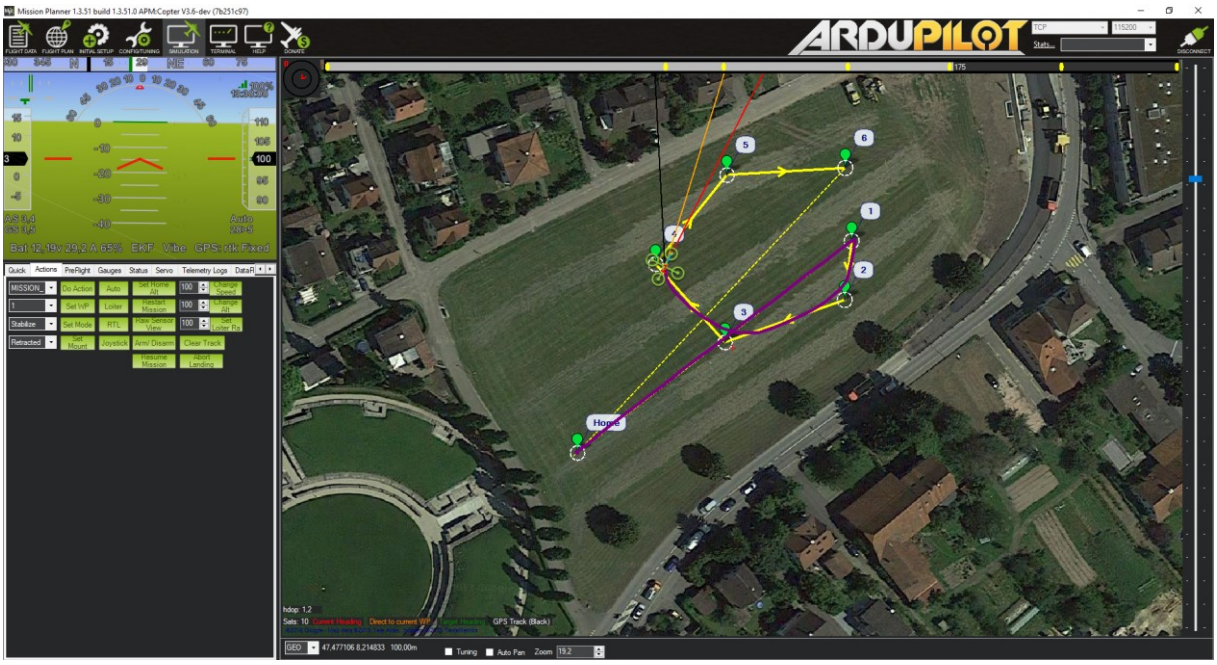


Abbildung 26: *während der Simulation*

5. Reflexion

Unsere Projekte befasst sich mit Technologie die Heutzutage immer mehr genutzt sind, und mit dem Bereich des Eingebettetes System. Unsere Motivation war während dieses Projekt immer vorhanden, auch wenn wir auf Probleme gestossen sind. Wir haben viel Zeit für die Ladung der Firmware auf das Board verloren, aber wir könnten parallel auf anderen Teilen des Projekts arbeiten. Deswegen könnten wir es in der Zeit abschiessen.

Während des Projekt wurde mit Dropbox des Codes, die Dokumentation, die Quellen, die Bilder und Videos gearbeitet und gespeichert. Dies hat uns erlaubt, auf unsere Rechner in Frankreich weiter zu arbeiten.

Wir haben sehr viel über die Drohnen gelernt und sind bereit über einer Erweiterung der Optionen eine Drohne zu Arbeiten. Unsere Motivation für die Thesis ist im Laufe dieses Projekts gewachsen.

Danksagung:

Wir möchten uns bei Prof. Dr. Nouri, unserem Auftraggeber und Betreuer, für die Empfehlungen bedanken.

6. Schlusswort

Die Hauptziele dieses Semesterprojekt könnten erreicht werden. Ein Hexakopter wurde aufgebaut, vollständig ausgerüstet und alle nötigen mechanischen Teile wurden gefertigt. Durch die durchgeführten Tests könnten das Betrieb der Drohne und die Ausgänge des APM nachgewiesen werden. Aus Sicherheitsgründen könnten wir den automatischen Abflug und Landung der Drohne nicht probieren. Deswegen haben wir es mit Mission Planner simuliert.

Unsere grösseren Probleme war den Code auf den APM zu laden. Wie erklärt, hat es mit Mission Planner und Arduino nicht funktioniert. Wir sind jetzt am Ende dieses Projekt und wissen leider immer noch nicht warum das Firmware nicht auf das Board geladen war. Die Lösung mit Linux die wir benutzen haben, ermöglicht uns ein spezielle Virtuelle Umgebung benutzen, der nur für dieses Projekt reserviert ist.

Das gesamt Umgebung dieses Projekt (der Drohne, Mission Planner, Kompilation mit Linux, usw.) ist grundsätzlich beherrscht. Für die Thesis geht es darum, einige Details zu vertiefen, je nachdem, wie die Drohne eingesetzt wird.

7. Ehrlichkeitserklärung

Mit der Unterschrift bestätigen die Unterzeichnenden, dass das Dokument selbst geschrieben worden ist.

Projektmitglieder

Haberkorn J.	Ort, Datum
Schellhorn A.	Ort, Datum

Literaturverzeichnis

- [**3D Robotics(2017)**] I. 3D Robotics. Electronic speed controller 20amp, 2017.
  URL : https://3drobotics.zendesk.com/hc/en-us/articles/207358766-ESC-20-Amp-with-SimonK.
- [**Arducopter(2012)**] Ardcopter. Connecting everything for arducopter, 2012.
  URL : http://www.arducopter.co.uk/all-arducopter-guides/2connectingeverything-for-arducopter
- [**Arducopter(2017)**] Arducopter. Arducopter website, 2017.
  URL : http://www.arducopter.co.uk/.
- [**Ardupilot(2017)a**] ardupilot. Code overview (copter), 2017.
  URL : http://ardupilot.org/dev/docs/apmcopter-code-overview.html.
- [**Ardupilot(2017)b**] Ardupilot. Ardupilot firmware, 2017.
  URL : http://firmware.ardupilot.org/
- [**Ardupilot(2017)b**] Ardupilot. Ardupilot website, 2017.
  URL : http://ardupilot.org/
- [**Chiaretta(2012)**] S. Chiaretta. Radio channels/inputs for arducopter 2.8, 2012.
  URL : http://diydrones.com/forum/topics/radio-channels-inputs-for-arducopter-2-8.
- [**EHIROBO**] EHIROBO. Copterx (cx-m3530-14-kv1100) m3530 1100kv brushless motor.
  URL : http://www.ehirobo.com/copterx-cx-m3530-14-kv1100-m3530-1100kv-brushless-motor.html
- [**Electronics(2013)**] M. Electronics. Users' Manual For MAYTECH Brushless Electronic Speed Controller (Version 2013.05) Harrier-Suprem Series / Harrier-Suprem Economical Series / Tiger-Shack Series, 2013.
- [**fpvracing**] fpvracing.ch. Multicopter komponenten.
  URL : https://fpvracing.ch/de/content/8-multicopter-komponenten
- [**GRAUPNER(2017)**] GRAUPNER. Fernsteuerungs-set mx-12 hott, empfänger gr-12
  URL : https://www.graupner.de/Fernsteuerungs-Set-mx-12-HoTT-GB-und-Empfaenger-GR-12/33112.EN/
- [**I/O(2016)**] V. I/O. Connecting external leds to arducopter, 2016.
  URL : https://visser.io/2014/10/connecting-external-leds-to-arducopter/
- [**"Do it Yourself" Drones(2017)**] Do it Yourself" Drones. Diy drones website, 2017.
  URL : http://diydrones.com/
- [**KG(2011)**] G. G. C. KG. GETTING STARTED - Groupner HoTT, 2011.
- [**KG(2012)**] G. G. C. KG. 33112.mx-12 HoTT.2.fr, 2012.
- [**KG(2014)**] G. G. C. KG. S1012 GR-12L HoTT, 2014.
- [**Kletti(2014)**] T. P. F. Kletti. Dezentralisierte drohnen kommunikation, 2014.
- [**MYRCMART**] MYRCMART. Rcx bc2836-9 880kv outrunner brushless motor.
  URL : http://www.myrcmart.com/rcx-bc28369-880kv-outrunner-brushless-motor-airplane-free-mounts-p-3993.html
- [**3D Robotics(2013)**] I. 3D Robotics. 3DR RADIO V2 - QUICK START GUIDE, 2013.

Abbildungsverzeichnis

Abbildung 1: Happymodel HMF S550 6
Abbildung 2: ArduPilct Mega 2.5 6
Abbildung 3: Fernbedienung Graupner GR-12 HoTT 7
Abbildung 4: Motor RCX BC2836-9 Brushless 8
Abbildung 5: ESC 3D Robotics 8
Abbildung 6: M8N GPS 9
Abbildung 7: Power Distribution Board 9
Abbildung 8: neuen Pins auf der PDB gelötet 9
Abbildung 9: Batterie und Ladegeräte 10
Abbildung 10: Plattform für GPS und APM 11
Abbildung 11: draufsicht der Plattform 11
Abbildung 12: Vorderansicht der Plattform 11
Abbildung 13: Power Modul 12
Abbildung 14: Servomotor Hitec HS-6965HB 13
Abbildung 15: gesamt Übersicht des Antriebes 13
Abbildung 16: APM 2.5 Ausgängen numerierung 15
Abbildung 17: Standardkonfiguration der Ausgänge 16
Abbildung 18: falsche Bestätigung des Hochladung - Arduino 17
Abbildung 19: Hochladung und Folder des .hex Datei - Arduino 18
Abbildung 20: Neue Firmware hochladen – Mission Planner 18
Abbildung 21: Schaltplan des LED auf das APM 20
Abbildung 22: Test Gerät für Relaisausgang 21
Abbildung 23: Schaltplan des Servomotors auf das APM 22
Abbildung 24: Test Gerät für Pulsweitenmodulation Ausgang 22
Abbildung 25: Start der Simulation 23
Abbildung 26: während der Simulation 24

Tabellenverzeichnis

Tabelle 1 : Testmethoden für verschiedene Komponenten.....19

A. Programmcode

a. UserCode.pde

```
1  /// -*- tab-width: 4; Mode: C++; c-basic-offset: 4; indent-tabs-mode: nil -*-
2
3  #ifdef USERHOOK_INIT
4  void userhook_init()
5  {
6      // put your initialisation code here
7      // this will be called once at start-up
8  }
9  #endif
10
11 #ifdef USERHOOK_FASTLOOP
12 void userhook_FastLoop()
13 {
14     // put your 100Hz code here
15     if (g.rc_6.radio_in > LED_TRIGGER)
16     {
17       hal.gpio->write(LED_A4, HAL_GPIO_LED_ON); // turn on the LED
18     }
19     if(g.rc_6.radio_in < LED_TRIGGER){
20       hal.gpio->write(LED_A4, HAL_GPIO_LED_OFF); // turn off the LED
21     }
22 }
23 #endif
24
25 #ifdef USERHOOK_50HZLOOP
26 void userhook_50Hz()
27 {
28     // put your 50Hz code here
29 }
30 #endif
31
32 #ifdef USERHOOK_MEDIUMLOOP
33 void userhook_MediumLoop()
34 {
35     // put your 10Hz code here
36 }
```

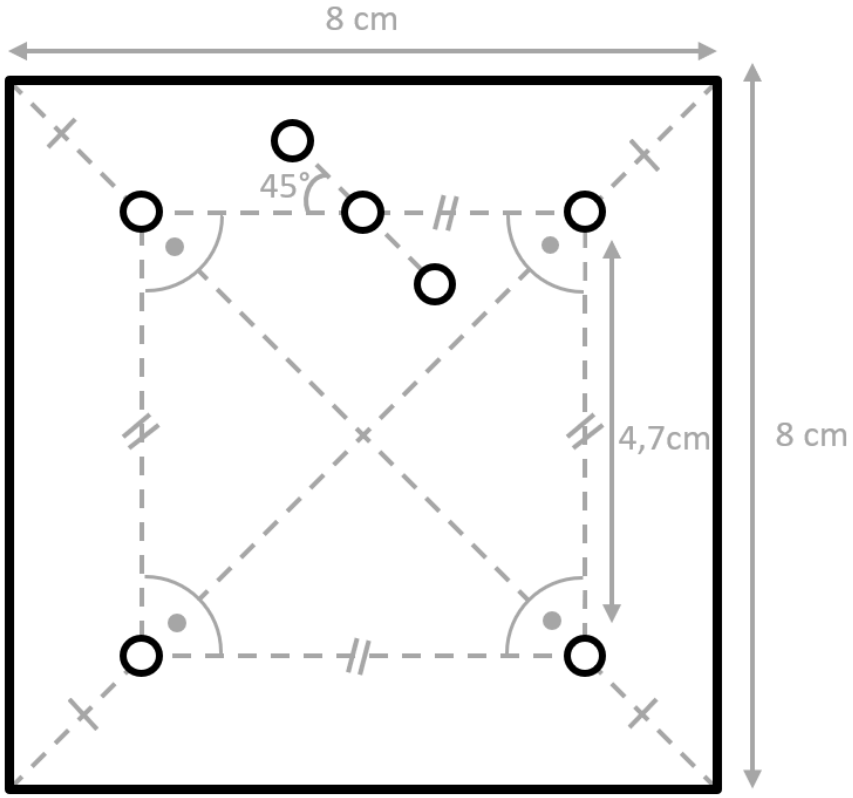
```
37  #endif
38
39  #ifdef USERHOOK_SLOWLOOP
40  void userhook_SlowLoop()
41  {
42      // put your 3.3Hz code here
43      hal.console->printf_P(PSTR("\nCHANNEL 6 : %d"), g.rc_6.radio_in);
44  }
45  #endif
46
47  #ifdef USERHOOK_SUPERSLOWLOOP
48  void userhook_SuperSlowLoop()
49  {
50      // put your 1Hz code here
51      gcs_send_text_P(SEVERITY_HIGH, PSTR("\nTEST gcs\n"));    // sends the text to the
52  Ground Station Control, usually used for important messages (ex: errors)
53  }
54  #endif
```

b. UserCode.h

```
1  /// -*- tab-width: 4; Mode: C++; c-basic-offset: 4; indent-tabs-mode: nil -*-
2
3  // user defined variables
4
5  #ifdef USERHOOK_VARIABLES
6
7  int LED_TRIGGER_HIGH = 1700;
8  int LED_TRIGGER_LOW = 1200;
9  int LED_TRIGGER = 1500;
10 bool buzzer_state = true;
11
12 bool state_LED = true;
13 const int OFFSET_OUTPUT = 54; //to set in OUTPUT Digital + 54
14 const int LED_A4 = 4+OFFSET_OUTPUT; //PIN A4
15
16 /* // example variables used in Wii camera testing - replace with your own variables
17 #if WII_CAMERA == 1
18 WiiCamera          ircam;
19 int                WiiRange=0;
20 int                WiiRotation=0;
21 int                WiiDisplacementX=0;
22 int                WiiDisplacementY=0;
23 #endif  // WII_CAMERA */
24
25 #endif  // USERHOOK_VARIABLES
```

B. Fertigungsdaten Mechanik

a. GPS / APM Plattform



Projektausschreibung Originalversion P5/P6

Titel: Drohne für Landwirte
Betreuer: Prof. Dr. T. Nouri Taoufik.Nouri@fhnw.ch
Auftraggeber:
Aufgabenbeschreibung:

Ausgangslage:
Die Landwirte brauchen neue Technologien wie Drohnen usw. Mehrere Quadcopter werden den Studierenden zur Verfügung gestellt(mit Basis Software).
Diese Quadcopter werden verwendet um Düngemittel auf Gemüse, Pflanzen, Bäumschen usw. zu verteilen. Diese Aufgabe kann auch von einer Drohnenschwarm realisiert werden

Zielsetzung:

1. Kommunikation zwischen dem Drohnenschwarm
2. Automatische Koordination zwischen dem Drohnenschwarm
3. Planung durch Überflug von Parzellen, die behandelt werden sollen
4. Aufnahme (Foto / Video), Codierung und Übertragung der behandelten Land Zone zu Basisstation
5. Entwicklung des Düngemittelverteilungsmechanismus

Schlüsselwörter:
Embedded-System, Kommunikation, Programmierung
Digital/Analog electronic

Eignung:			
X	P5 (180 h)	X	P6 (Bachelor-Thesis, 360 h)
O	Profil Energie- und Antriebssysteme	X	Profil Mikroelektronik und Kommunikationssysteme
Anforderungen:			
X	Ein Studierender	X	Zwei Studierende
Besuchte Module: Andere Vorkenntnisse oder Anforderungen:			
Weitere Angaben:			
X	Das Projekt steht für die zweite Phase der Ausschreibung zur Verfügung, soweit es nicht schon vorher vergeben wurde.		
X	Eine Fortsetzung des Projekts P5 als P6 ist vorgesehen.		
Bemerkungen:			

Plichtenheft

Fachhochschule Nordwestschweiz
Hochschule für Technik

Drohne für Landwirte

Projekt 5: Pflichtenheft

Projektmitglieder	Haberkorn Jonas Schellhorn Antoine
Projektnummer	EIT-HS17-_____ *(noch nicht bekannt)*
Auftraggeber	Nouri T., Prof. Dr. Dip. Eng. Elec./Phys.
Studiengang	Bachelor Elektro- und Informationstechnik – I.C.S.
Datum	Mittwoch, 4. Oktober 2017

Inhaltsverzeichnis

1. Einleitung 2

2. Ziele 3

2.1 Muss-Ziele 3
2.2 Wunsch-Ziel 3
2.3 Nicht-Ziele 3

3. Lieferobjekte 4

4. Projektumfeld 5

4.1 Fernbedienung 5
4.2 Software 6
4.3 Flight Controller 7

5. Testkonzept mit Meilensteine 10

5.1 Kommunikation zwischen die Fernbedienung und das ArduPilot 10
5.2 Kompilation und Hochladung des Programms für das ArduPilot 10
5.3 Fernkontrollierung einen Ausgang des ArduPilot 10
5.4 Eichung des ArduPilot und Start und Steuerung den Motoren 10
5.5 Automatische Abflug und Landung 11
5.6 Flight Plan 11

6. Zeitplan 12

7. Projektvereinbarung 13

1. Einleitung

Drohnen werden immer mehr benutzt um speziellen Bedarf zu antworten. Sie sind klein, schnell, leichtgängig und können fliegen. Diese letzte Charakteristik ist auch die wichtigste und auch die komplizierte, da, die Stabilisierung schwierig ist. Trotzdem sind die Anwendungsmöglichkeiten einer Drohne sehr breite. Der rote Faden für unser Projekt und die Thesis ist die Anwendung von Drohnen um Anforderung von Landwirten zu beantworten. Dafür sollten wir zuerst eine Drohne aufbauen und Inbetriebsetzen. Dass ist das Ziel dieses Projekt 5.

Drohnen existieren in viel verschiedene Formen. Die bekannteste Drohne Familie (Multikopter) enthält die Trikopter, Quadrokopter, Hexakopter, usw. Es gibt auch verschiedene Architekturen, zum Beispiel: die 6 Propeller eines Hexakopter können gleichmässig herum der Mitte aufgeteilt sein, aber sie können auch nur in 3 Richtungen aufgeteilt sein mit immer 2 Propeller aufeinander die sich in Gegenrichtungen drehen.

Eine Drohne kann auch anderen Geräten tragen, wie zum Beispiel, ein GPS-Modul, Kameras, Sensoren usw. Deswegen, sind sie sehr modulierbar.

Um eine Drohne aufzubauen und zu programmieren existieren spezifische Geräten und Softwares die das ganze erleichtern. Diese sind meisten im Hobbybereich des Modellfliegens benutzt. Es existieren vielen Artikeln, Website, Forums, usw. die Informationenmine sind.

2. Ziele

Unsere Zielsetzung besteht aus verschiedene Elementen, diese sind -je nach ihren Wichtigkeiten- in drei Kategorien aufgeteilt (Muss-, Wunsch- und Nicht-Ziele).

2.1 Muss-Ziele

- Eine Drohne aufbauen (Grundplatte, Elektronik, Speisung, Fernbedingung, Antrieb, usw.).
- Betrieb der Softwareumgebung beherrschen.
- Das Steuerungsprogramm erfassen und Änderungsmöglichkeiten des Programms einarbeiten.
- Tests durchführen um für die Inbetriebnahme bereit zu sein.
- Ein- und Ausschaltung eines L.E.D. auf dies Ausgängen des Flight Controller mit der Fernbedienung kontrollieren zu können.
- Automatik Abflug und Landung der Drohne
- Mit der Drohne eine Liste von vordefinierte Punkte überfliegen

2.2 Wunsch-Ziel

- Weitere Recherchen und Teste in Richtung unsere Thesis.

2.3 Nicht-Ziele

- Wir würden keine Drohne komplett neu herstellen, sondern würden wir ein (spezifisch zu Modellierung Anbausatz) benutzen.
- Es wird kein Fliegen-Verwaltungssoftware entwickelt, sondern würden wir Open-Source Software benutzen und möglicherweise erweitern um unsere Ziele zu erreichen.

3. Lieferobjekte

Die Lieferobjekte werden vom Auftraggeber zum einem definierten Datum erwartet und müssen bis dann fertiggestellt sein.

Lieferobjekt	**Datum**
Pflichtenheft	Kalenderwoche 42/43 2017
Fachbericht	18. Januar 2018

Während des Projektes würden wir ein Journal halten wo wir unsere Fortschritte, Überlegung und Erwartungen für die nächst Woche eintragen.

4. Projektumfeld

4.1 Fernbedienung

4.1.1 Graupner HOTT mx-12

Um unsere zu kontrollieren, brauchen wir eine Fernbedienung. Sehr verwendet in Modellierung, die mx-12 von Graupner hat ein leichter Handgriff und ist leicht programmierbar. Sie hat folgende Eigenschaften:

- 6 Kanal Sender
- SD-Karte Slot
- Telemetrie
- Grosses Graphische Display
- Umfangreiche Ausstattung
- Freie Zuordnung aller Schalter
- Freie Zuordnung aller Drehregler
- Ergonomisch gestaltetes Gehäuse
- 2.400 bis 2.484,5 MHz (Funkwellen)

Geeignet für Drohnen, Helikopter-, Flugzeug- und Auto-Modellen. Sie ist auch Bezeichnet als die „Ideale Basis“ zu haben für Anfängern.

Abbildung 1: ***Graupner HOTT mx-12 Fernbedienung***

4.2 Software

4.2.1 APM Planner und Mission Planner

Wir benutzen die Open-Source Softwares APM 2.0 und Mission Planner 1.3.49 die bieten beide:

- Basic Einstellung („initial Setup"), das heißt das Gyroskop, die Motoren, die Fernbedienung kalibrieren
- Failsafe Mode der sich automatisch startet, wenn die Batterien fast leer sind.
- Es gibt auch optional Hardware Einstellungen wie GPS, Radio, Bluetooth oder eingebettet Kamera.

Beiden Softwares bieten auch die Möglichkeit ein Flugplan zu setzen, der von der Drohne gefolgt wird. Wir können auch Daten während des Flugs im Umgebung „Flight Data" nachsehen (zum Beispiel die GPS Koordinaten, die Position auf der Weltkarte, das Orientierung, die Höhe und die Geschwindigkeit, usw.) Es ist auch möglich falls eine Kamera vorhanden ist ein direktes Feedback zu erhalten.

APM Planner bietet zusätzlich im Umgebung „Flight Data" die echte Höhe, Batterie Status, Tension (V) und Intensität des Strohm (A). Nach dem Flug können wir nur auf dieser Software Graphs von dem Flug erzeugen.

Abbildung 2: *Mission Planner Umgebung*

4.3 Flight Controller

4.3.1 ArduPilot Mega

ArduPilot Mega ist ein Open-Source Projekt, es ist ein Board die verschiedenen RC Modellen steuern kann durch ESC und mit Hilfe eine Software oder eine Fernbedienung. Für unser Projekt wird nur der HexaKopter Drohne Model benutzt. Beim ArduPilot Mega sind mit diesem Model verschiedene Flight Modes anwählbar, über das PWM Signal des Empfängers können so die vorher einprogrammierten Modi angewählt werden. Folgende stehen aktuell zur Verfügung:

- *Acro*

Bei diesem Flugmodus sind die Stabilisierungen ausgeschaltet, vergleichbar mit einem Heli Modell muss jede Bewegung manuell ausgeglichen werden und auch wieder gestoppt werden. Lässt man einfach die Knüppel los, fliegt der Kopter in die vorher gelenkte Richtung weiter. Daher nur für erfahrene Piloten.

- *Stabilize Mode*

Hier funktionieren die Stabilisierungen. Der Kopter bringt sich automatisch in eine Horizontale Ausrichtung. Lediglich Yaw Eingaben werden beibehalten, fliegt man jedoch nach Vorne und lässt die Knüppel los, stoppt der Kopter und bleibt auf der Position (Windstille vorausgesetzt, durch Wind wird der Kopter in die jeweilige Richtung getrieben). Wir würden für das Projekt dieser Flug Modus am häufigsten benutzen.

- *Alt hold*

Hier wird der Kopter versuchen die Höhe zu halten, beim Einschalten des Modes wird die aktuelle Höhe zum neuen Nullpunkt. Man kann trotzdem in alle Richtungen weiterfliegen, jedoch wird der Kopter die Höhe halten.

- *Simple Mode*

Der Simple Mode ist besonders für Anfänger gut geeignet. Er wird auch als „Care free“ bezeichnet. In diesem Modus spielt die Ausrichtung des Kopters keine Rolle, er fliegt bei Stick Eingabe immer in die Richtung vom Piloten ausgesehen. Wenn also Yaw um 90° im Uhrzeigersinn gedreht wird und die rechte Seite des Kopters zu sich zeigt, fliegt er vom Piloten weg wenn die Stick Eingabe „vorwärts“ lautet anstatt nach rechts zu fliegen. Der Kopter könnte also

durchgängig gedreht werden und trotzdem würde er in eine Richtung fliegen. Achtung: Der Pilot muss dabei unbedingt seine Ausrichtung beibehalten.

- *Auto Mode*

Dies ist einer der interessantesten Flugmodus. Hier wird der Autopilot aktiviert, über den Computer beziehungsweise Notebook und, muss vorher im Mission Planner 1.3 oder APM 2.0 eine Flugroute ausgewählt werden (Wegpunkte) sowie eine Endfunktion. Wird keine Endfunktion angegeben wird der Kopter in den Leiter Mode schalten und am letzten Weg punkt verharren.

- *RTL (Return to launch)*

Beim RTL Modus fliegt der Kopter zum Ausgangspunkt zurück. Interessant zum Beispiel als Failsafe, wenn der Akku leer geht. Allerdings gibt es hier einiges zu beachten: Für RTL wird GPS benötigt und der „Launch Punkt“ wird vom Kopter dort gesetzt wo er sein GPS fix hat, d.h. wenn die blaue LED am APM dauerhaft leuchtet. Wenn also auf „Return to launch“ geschaltet wird sollte man sich sicher sein wo der Startpunkt war. Der Kopter wird auf eine vorher definierte Höhe steigen, sich in die Richtung zum Launch punkt drehen und losfliegen, dort angekommen wird er auf eine niedrigere Höhe absteigen und in den Loiter Mode schalten. Nach 20 Sekunden dann landen. Alle diese Parameter können über den Mission Planner eingestellt werden.

- *Guide Mode*

Dieser ist ähnlich dem RTL Flight Mode, mit dem Unterschied das der Launch Punkt in diesem Fall auf der Karte im Mission Planner festgelegt wird. Dort angekommen hovert er im Loiter Mode und wartet auf weitere Befehle.

- *Loiter Mode*

Der Loiter Mode ist eine Art Position Hold in der die aktuelle Höhe als neuer Nullpunkt fingiert. Der Kopter wird die aktuelle Position in Höhe und zu allen Seiten halten, GPS gestützt. Dennoch kann geflogen werden und die neue Position wird dann gehalten. Lediglich landen und disarmen geht nicht, dafür muss in den Stabilize Mode geschaltet werden.

- *OF_Loiter Mode*

Gleich wie der Loiter Mode jedoch ohne GPS und nur möglich mit vorhandenen Optical Flow Sensor über den der Kopter die Position dann hält.

- *Position Hold*

Ähnlich wie Loiter Mode ohne die Höhe als neuen Nullpunkt zu nehmen.

- *Circle Mode*

Dieser Flugmodus ist vor allem für Filmaufnahmen interessant in dem ein Objekt um gekreist werden soll. In einem vorher definierten Radius (Mission Planner), kreist der Kopter um den Punkt auf den er gerade zeigt. Dabei bleibt die Yaw Ausrichtung auf dem Punkt oder Objekt.

- *Follow me*

Das „Follow me“ Modus ist -wie der Name schon sagt- ein Modus in dem der Kopter dem Piloten autonom folgt. Die Voraussetzung dafür ist zum einen das Telemetrie Modul, sowie ein Notebook mit GPS oder ein Android Handy oder Tablet mit GPS. Sehr interessant für Film Aufnahmen von Auto- beziehungsweise Motorradfahrten oder auch über Wasser.

5. Testkonzept mit Meilensteine

Während unsere Projekt würden verschiedene Testen durchgeführt. Diese Tests würden als Meilensteine benutzt, um bestimmten Recherchen- beziehungsweise Entwicklungsphase zu beenden.

5.1 Kommunikation zwischen die Fernbedienung und das ArduPilot

Um den Drohne drahtlos zu kontrollieren, müssen wir eine Funkkommunikation zwischen die Fernbedienung und das ArduPilot erstellen. Dafür müssen wir das Empfänger an den ArduPilot anschliessen. Wenn die Kommunikation gut etabliert ist, sind wir in der Lage des Kopter zu kontrollieren.

5.2 Kompilation und Hochladung des Programms für das ArduPilot

Um den ArduPilot zu programmieren müssen wir eine bestimmte Software Umgebung installieren. Dann müssen wir uns damit ausüben und suchen wie man ein Standardcode damit Kompiliert.

Wenn das gemacht ist, müssen wir das kompilierte Programm auf das ArduPilot hochladen.

5.3 Fernkontrollierung einen Ausgang des ArduPilot

Für unsere Thesis ist es wichtig zu wissen: wie man die Ausgänge den ArduPilot benutzt und auch wie man es mit der Fernbedienung kontrollieren kann. Dafür möchten wir mit einem bestimmten Signal des Graupner mx-12 (zum Beispiel, wenn man eine Taste drückt) ein L.E.D. auf Ausgänge des ArduPilots ein- und ausschalten.

5.4 Eichung des ArduPilot und Start und Steuerung den Motoren

Wenn das Drohne fertig aufgebaut ist, müssen wir das ArduPilot eichen. Dafür würden die Motoren (ohne Propeller) bei der Software (APM Planner oder Mission Planner) gestartet. Beim erfolgreichste Eichung wird ein wichtiger Meilenstein erfüllen.

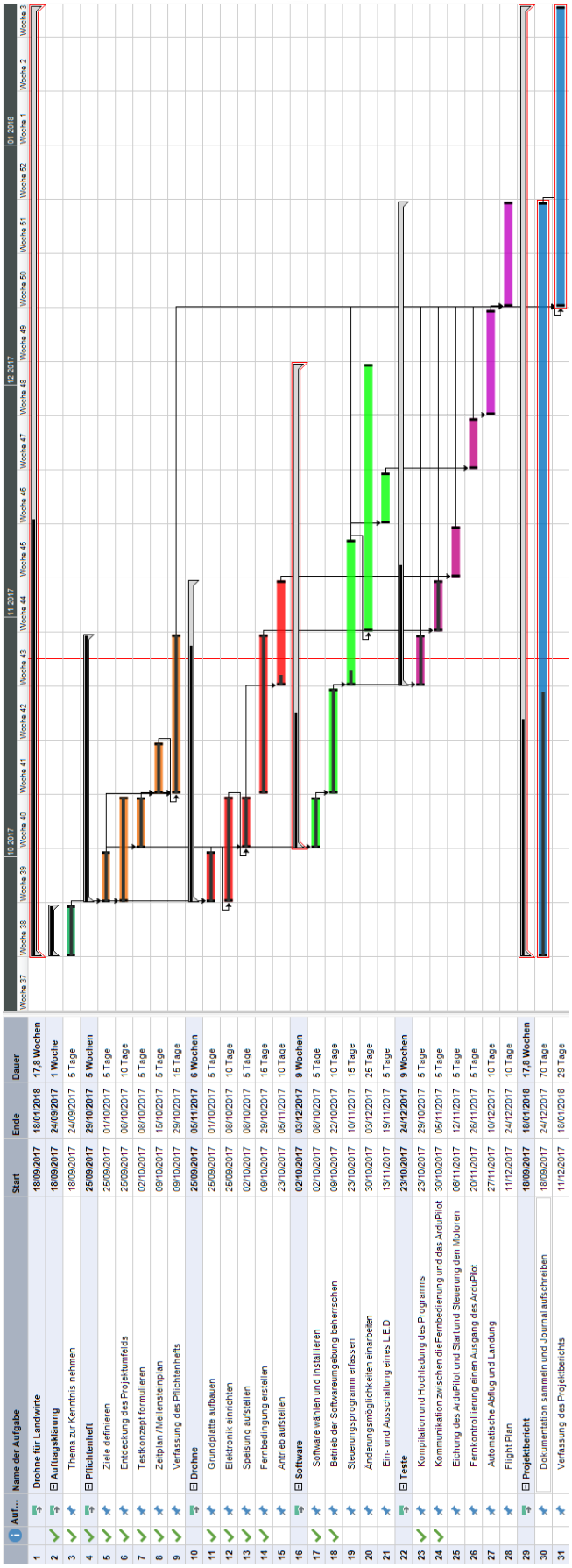
5.5 Automatische Abflug und Landung

Das erstes Ziel bezüglich der Fliegen ist den Abflug und die Landung beherrschen. Manchmal ist es notwendig ein Notfall Landung zu machen. Deswegen wird ein Automatische Abflug und Landung vor der Inbetriebsetzung getestet.

5.6 Flight Plan

Um ein Drohne für eine spezielle Anwendung benutzen zu können, muss sich die Drohne bewegen. Mit ein Flight Plan kann man es um verschiedene Zustand testen.

6. Zeitplan



	Auf...	Name der Aufgabe	Start	Ende	Dauer
1		**Drohne für Landwirte**	**18/09/2017**	**18/01/2018**	**17,8 Wochen**
2		**Auftragsklärung**	**18/09/2017**	**24/09/2017**	**1 Woche**
3		Thema zur Kenntnis nehmen	18/09/2017	24/09/2017	5 Tage
4		**Pflichtenheft**	**25/09/2017**	**29/10/2017**	**5 Wochen**
5		Ziele definieren	25/09/2017	01/10/2017	5 Tage
6		Entdeckung des Projektumfelds	25/09/2017	08/10/2017	10 Tage
7		Testkonzept formulieren	02/10/2017	08/10/2017	5 Tage
8		Zeitplan / Meilensteinplan	09/10/2017	15/10/2017	5 Tage
9		Verfassung des Pflichtenhefts	09/10/2017	29/10/2017	15 Tage
10		**Drohne**	**25/09/2017**	**05/11/2017**	**6 Wochen**
11		Grundplatte aufbauen	25/09/2017	01/10/2017	5 Tage
12		Elektronik einrichten	25/09/2017	08/10/2017	10 Tage
13		Speisung aufstellen	02/10/2017	08/10/2017	5 Tage
14		Fernbedingung erstellen	09/10/2017	29/10/2017	15 Tage
15		Antrieb aufstellen	23/10/2017	05/11/2017	10 Tage
16		**Software**	**02/10/2017**	**03/12/2017**	**9 Wochen**
17		Software wählen und installieren	02/10/2017	08/10/2017	5 Tage
18		Betrieb der Softwareumgebung beherrschen	09/10/2017	22/10/2017	10 Tage
19		Steuerungsprogramm erfassen	23/10/2017	10/11/2017	15 Tage
20		Änderungsmöglichkeiten einarbeiten	30/10/2017	03/12/2017	25 Tage
21		Ein- und Ausschaltung eines LED	13/11/2017	19/11/2017	5 Tage
22		**Teste**	**23/10/2017**	**24/12/2017**	**9 Wochen**
23		Kompilation und Hochladung des Programms	23/10/2017	29/10/2017	5 Tage
24		Kommunikation zwischen die Fernbedienung und das ArduPilot	30/10/2017	05/11/2017	5 Tage
25		Eichung des ArduPilot und Start und Steuerung den Motoren	06/11/2017	12/11/2017	5 Tage
26		Fernkontrollierung einen Ausgang des ArduPilot	20/11/2017	26/11/2017	5 Tage
27		Automatische Abflug und Landung	27/11/2017	10/12/2017	10 Tage
28		Flight Plan	11/12/2017	24/12/2017	10 Tage
29		**Projektbericht**	**18/09/2017**	**18/01/2018**	**17,8 Wochen**
30		Dokumentation sammeln und Journal aufschreiben	18/09/2017	24/12/2017	70 Tage
31		Verfassung des Projektberichts	11/12/2017	18/01/2018	29 Tage

7. Projektvereinbarung

Mit der Unterschrift bestätigen die Unterzeichnenden, dass sie das Dokument gelesen und den Inhalt verstanden haben.

Projektmitglieder

Haberkorn J.	Ort, Datum
Schellhorn A.	Ort, Datum

Auftraggeber

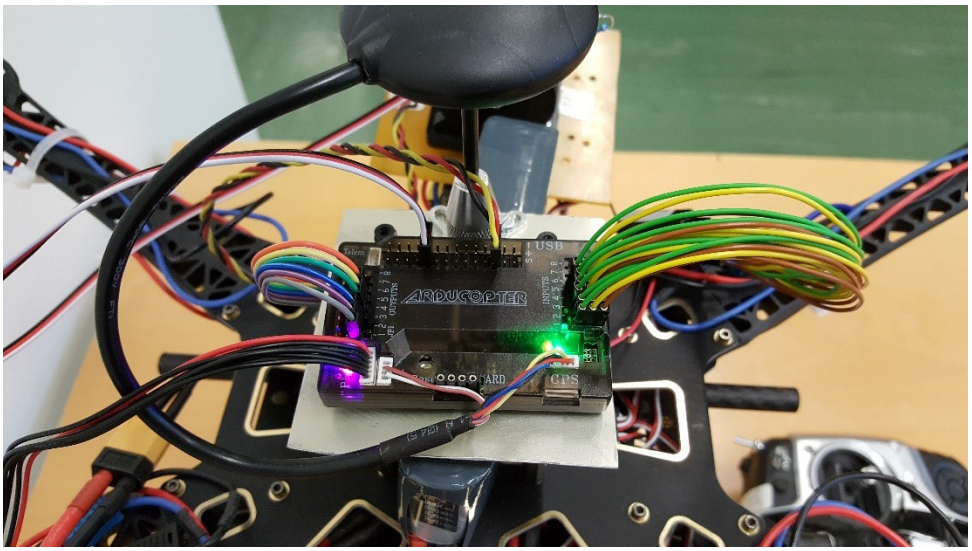
Nouri T.	Ort, Datum

Benutzermanual

Fachhochschule Nordwestschweiz
Hochschule für Technik

Drohne für Landwirte

Projekt 5: Benutzermanual



Projektmitglieder	Haberkorn Jonas Schellhorn Antoine
Projektnummer	EIT-HS17-_____ *(noch nicht bekannt)*
Auftraggeber	Nouri T., Prof. Dr. Dip. Eng. Elec./Phys.
Studiengang	Bachelor Elektro- und Informationstechnik – I.C.S.
Datum	Dienstag, 28. November 2017

Inhaltsverzeichnis

1 Einleitung 2

2 Das Material 3

3 Aufstellungen 7

3.1 Aufbau der Drohne 7

3.2 Softwareinstallation 14

3.2.1 Windows 14

3.2.1.1 Arducopter-3.2.1 Quellcode 14

3.2.1.2 MHV_AVR_Tools – Cross-Compiler (optional) 15

3.2.1.3 Arduino – die neuste Version (optional) 18

3.2.1.4 Arduino – für ArduPilot (optional) 20

3.2.1.5 Mission Planner 24

3.2.2 Linux 27

4 Kalibrierung und erste Verwendungen 29

4.1 Verbindung zum Controller 29

4.2 RC-Kanäle kalibrieren 30

4.2.1 Kalibrierung die Beschleunigungsmesser 31

4.2.2 Kalibrierung den ESC 32

4.2.3 Arming und disarming der Drohne 32

5 Änderungen 34

5.1 Durch den Code 34

5.1.1 Vorbereitung 34

5.1.2 Bearbeitung 35

5.1.3 Einsetzung 36

5.2 Durch Mission Planner 36

5.2.1 Aktivierung das erweiterte Layout 36

5.2.2 Steuerung einen PWM-Ausgange 37

5.2.3 Flight Plan 39

6 Quellen 40

1 Einleitung

In dieses Dokument wird erklärt:

- welche Material benutzen
- wie der Drohne eingebaut wird
- wie man die Softwares und Cross-Compiling Tools installiert und benutzt
- wie man Änderungen an den Code und Konfiguration vornimmt

Es soll einen schnellen und einfachen Handgriff der Drohne ermöglichen.

/!\ *Ein Windows Rechner und eine Virtuelle Maschine mit Linux* ***ist notwendig*** *um das Programm auf den Kontroller hochzuladen und ein Feed-back währen den Flug zu bekommen* /!\

Dieses Benutzermanual ist für den Kontroller APM 2.5 geeignet. Für neuere Kontrollers ist die Vorgehensweise nicht dieselbe, erkundigen Sie sich auf Internet für die einschlägige Methode.

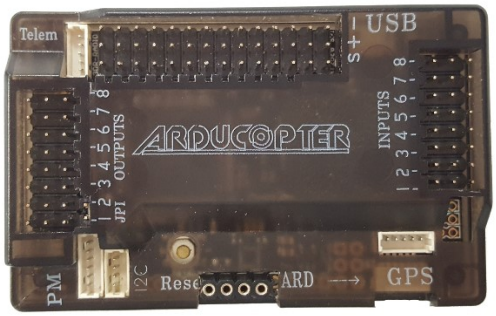
Falls Sie unsere Virtual Maschine bekommen, ist das Password „fhnw“.

2 Das Material

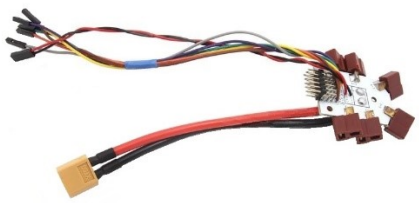
Benötigtes Material:

1. ein Drohnenskelett (hier das Happymodel – HMF S550)

2. ein Controller (hier den APM 2.5)

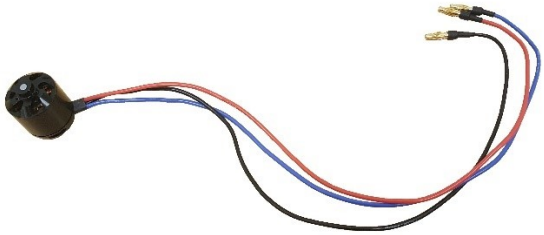


3. ein Stromverteiler / Power Distribution Board (PDB)



4. eine Batterie (mit seinem Ladegerät)

5. Motoren



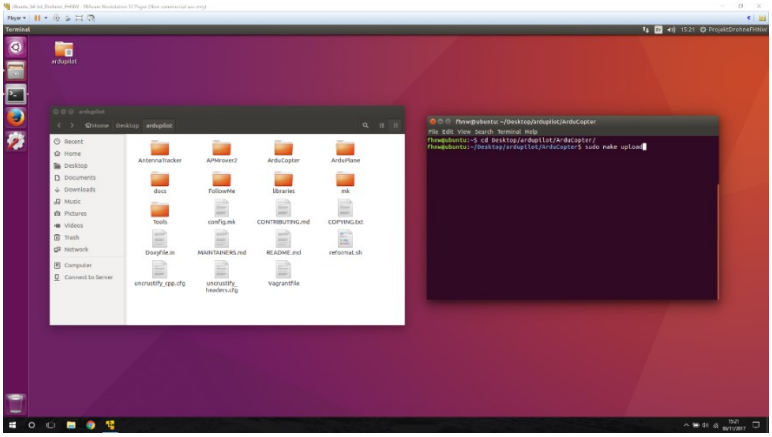
6. Geschwindigkeitswandler / Electronic Speed Controller (ESC)

7. ein GPS mit einem digitalen Kompasse

8. eine Fernbedienung (mit seinem Empfänger)

9. ein Leistungsmodul (für unsere APM 2.5)

10. ein Windows Rechner, der eine Virtuelle Maschine mit Linux hat



11. ein USB - mikroUSB Kabel

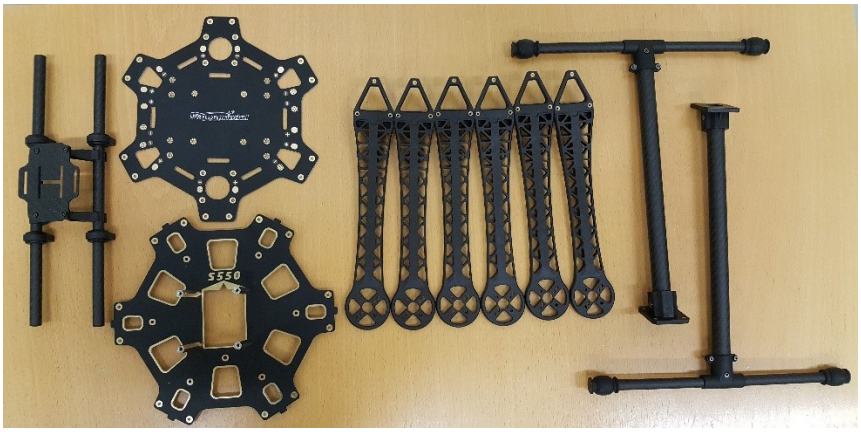
12. Jumper Kabeln

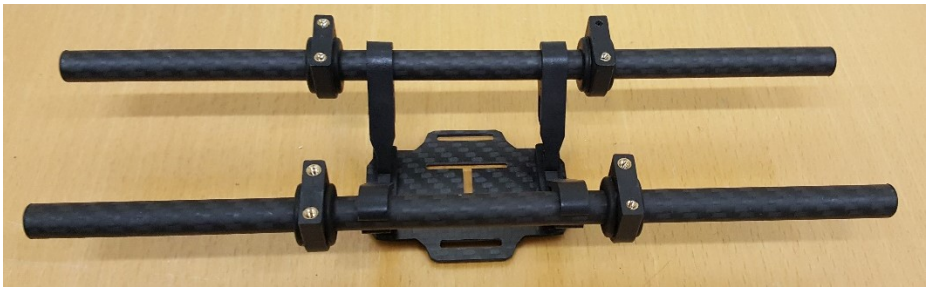
3 Aufstellungen

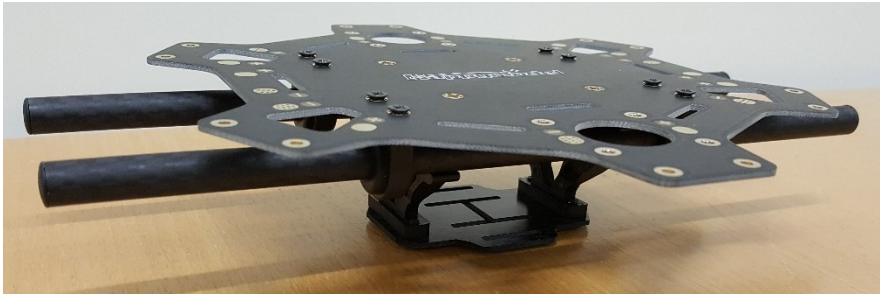
Hier wird den Aufbau der Drohne und die Softwareinstallation erklärt. Diese gelten für ein Hexa X Drohne mit ein APM 2.5 als Controller.

3.1 Aufbau der Drohne

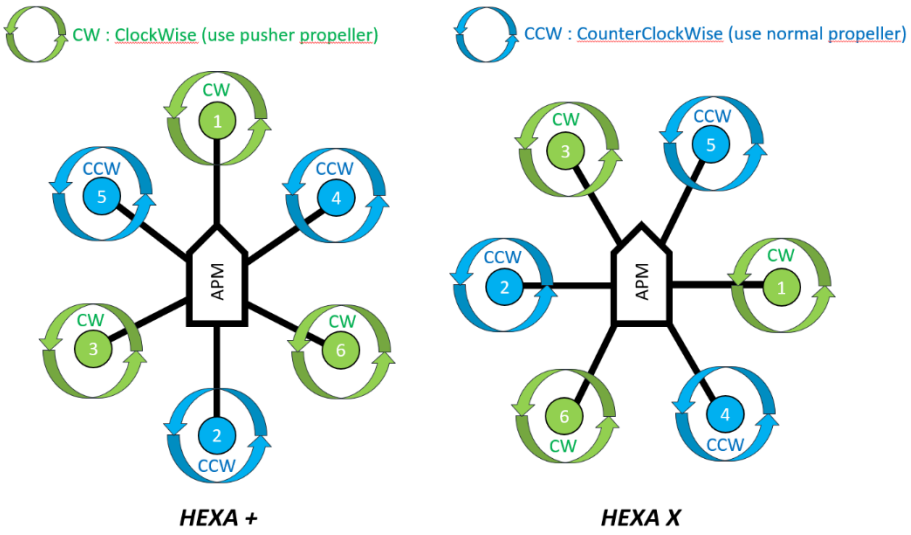
Zuerst, stellen Sie das Gehäuse zusammen, ohne die obere Platte. Beziehen Sie sich auf die folgenden Bilder.





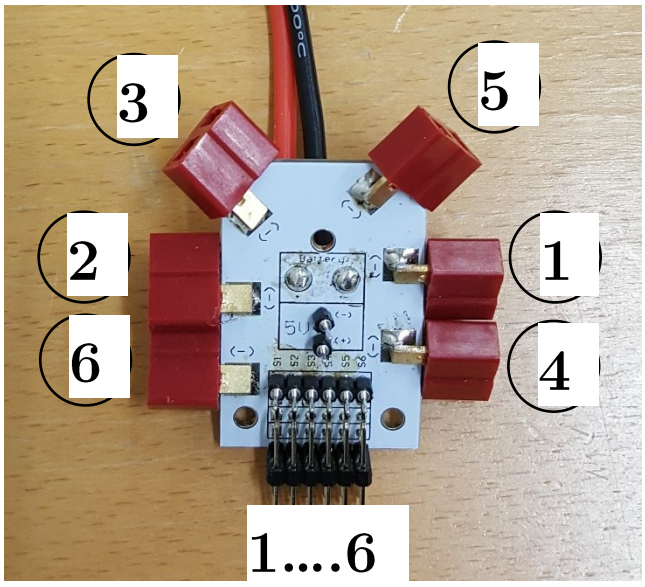


Bevor man mit der Verdrahtung beginnt, muss man darauf aufpassen zu welchen Arten von Drohnen diese Gebäude gehört. Die Folgende Abbildung bezeichnet für die beiden Konfigurationen (Hexa X und Hexa +) welche Nummer jeden Motor hat und in welche Richtung sich die Propeller drehen. (Achtung! Die Propeller müssen nur während überwachte Test fixiert werden!)

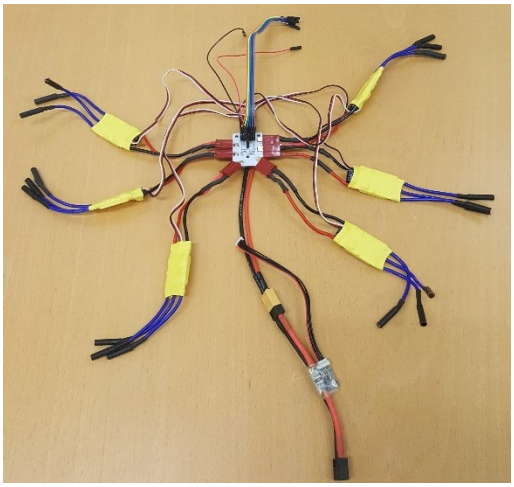


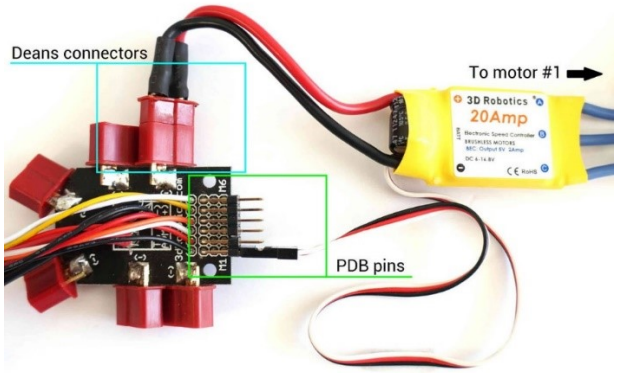
Quelle: http://www.ardupilot.org

Wir benutzen ein Hexa X Copter, für unsere Power Distribution Board sind die Ein- und Ausgänge so nummeriert:

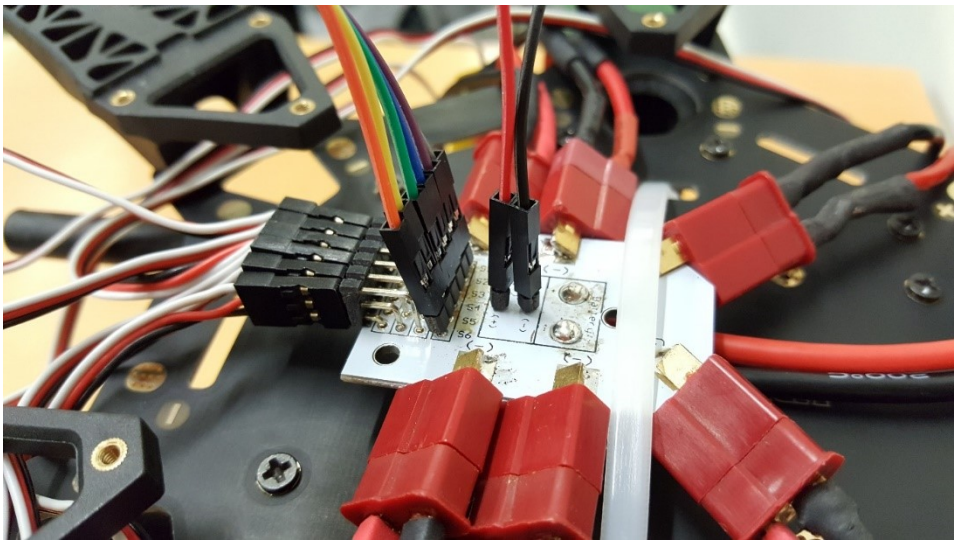


Danach müssen den ESC an die Ausgänge des PDB angeschlossen sein. Die Jumper und das Leistungsmodul können auch schon angeschlossen werden.

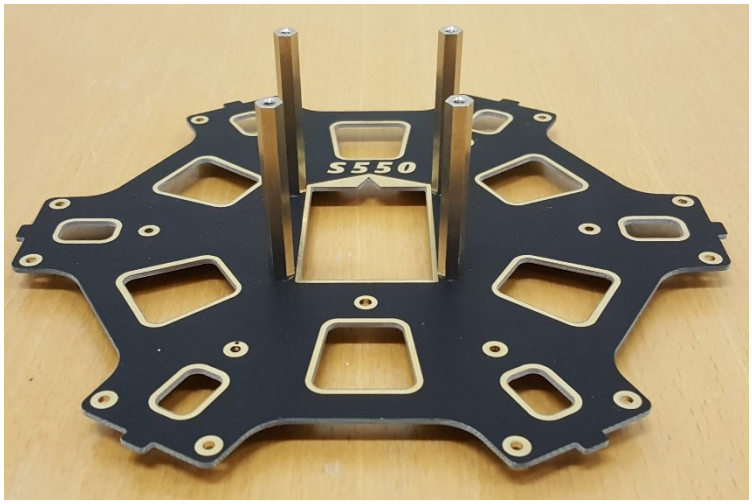


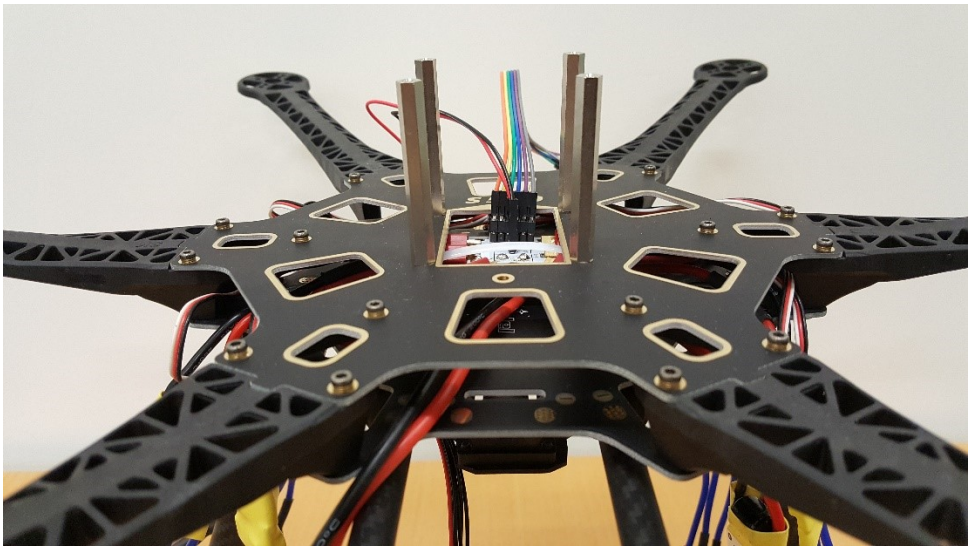


Fixieren Sie den PDB auf der Drohne. Achten Sie auf die Ausrichtung! Stellen Sie sich vor in welcher Direktion der Drohne fliegen wird und vergleichen Sie mit den obigen Abbildungen.

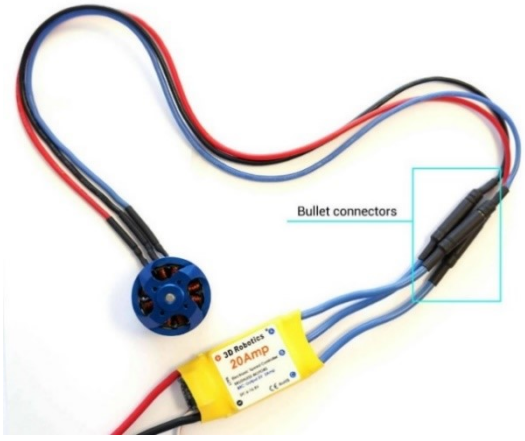


Zunächst, fixieren Sie den obere Platte. Wie haben vier Stützen eingeschraubt um mehr Platzt zu haben (wird weiter erklärt).

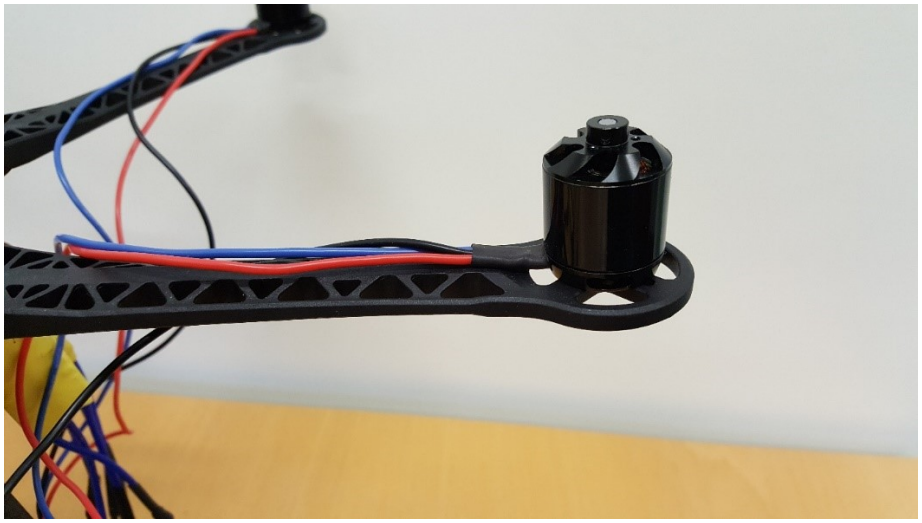




Sie können jetzt die Motoren anschrauben (aber immer ohne Propeller) und mit den ESC verbinden (rot des Motors gegenüber den schwarz Eingang des ESC, schwarz des Motor in die Mitte und blau oder gelb des Motor gegenüber den rot Eingang des ESC).

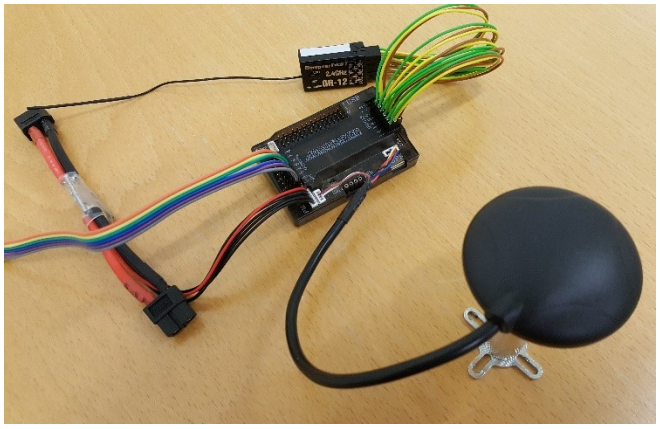


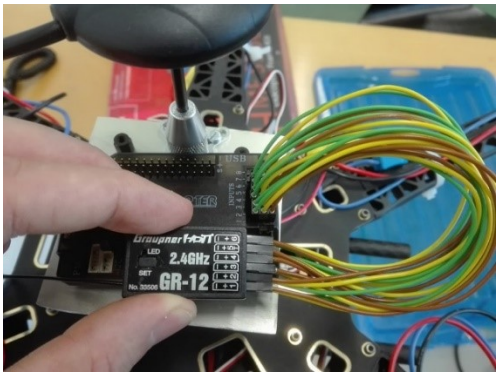
Quelle: http://www.instructables.com

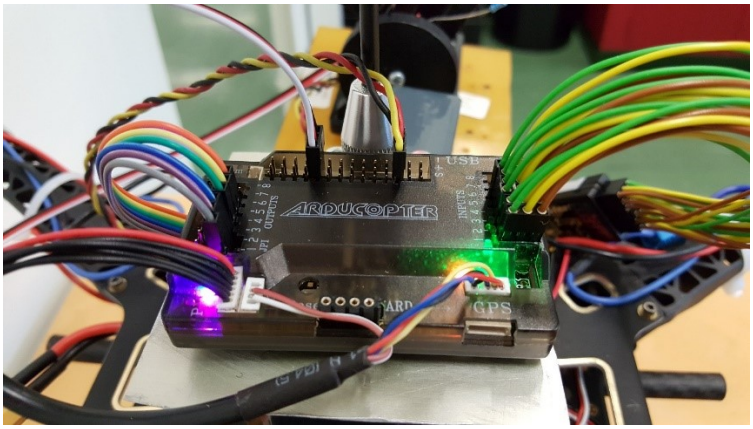


Um mehr Platzt zu haben, können Sie eine kleine Platte (nicht grösser als 8cm x 8cm für aerodynamischer Grunde) über der Drohne anschrauben.

Verbinden Sie den Receiver mit den Eingängen des ArduCopter, die Ausgänge des ArduCopter mit den PDB, den GPS und das Leistungsmodul mit den ArduCopter. Beziehen Sie sich auf den folgenden Bildern, die Pins sind jeweils so organisiert (von Zentrum zu Seite des APM) Signal-Positiv-Negativ. Deswegen müssen Sie nur den PDB auf den innersten Pins verbinden.







Zuletzt, schliessen Sie die Stromversorgung mit der Batterie an und Ihre Drohne ist bereit für die Kalibrierung.

Für ästhetische und praktische Gründen haben wir die Motoren und ESC Kabel mit ein paar Kabelbinder and das Gehäuse festgebunden.

3.2 Softwareinstallation

Hier wird alles erklärt um die Softwares und Cross-Compiling Tools zu einstellen.

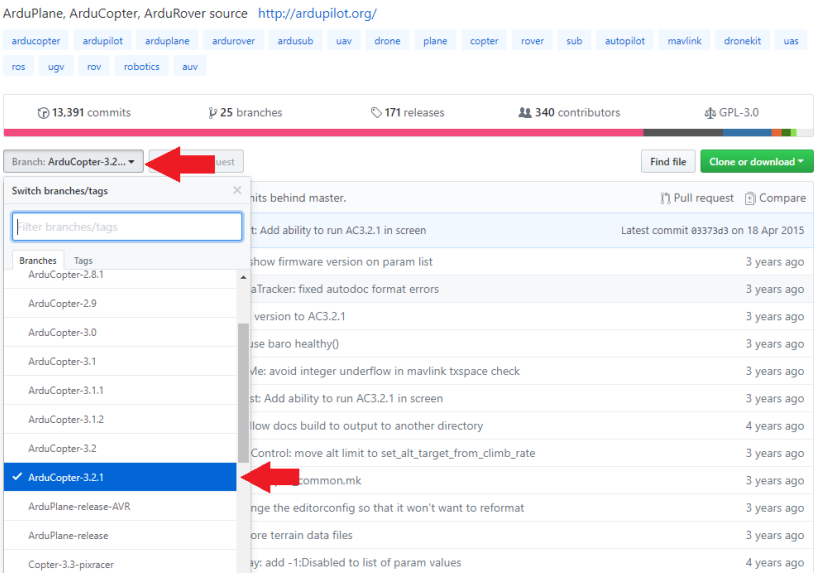
3.2.1 Windows

Auf Windows können wir den Code Kompilieren um die Richtigkeit zu verifizieren und ein Feedback von dem Controller zu bekommen, aber leider nicht den Code auf den Kontroller hochladen. Deswegen können Sie nur Mission Planner (Kapitle 3.2.1.5) installieren, wenn Sie den Code nur auf Linux editieren möchten.

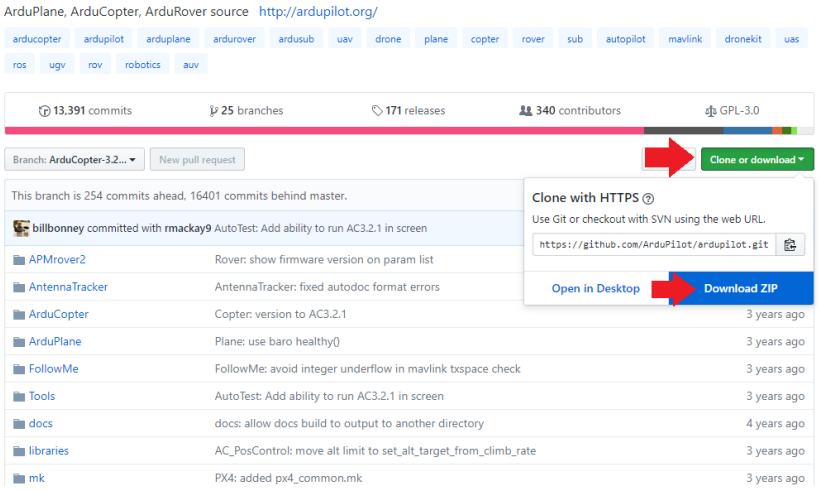
3.2.1.1 Arducopter-3.2.1 Quellcode

1. Gehen Sie auf *https://github.com/ArduPilot/ardupilot/*

2. Wählen Sie den Zweig “ArduCopter-3.2.1



3. Herunterladen und entpacken Sie es, wo Sie es wollen. Denken Sie daran, dass Sie den Code dort bearbeiten werden.

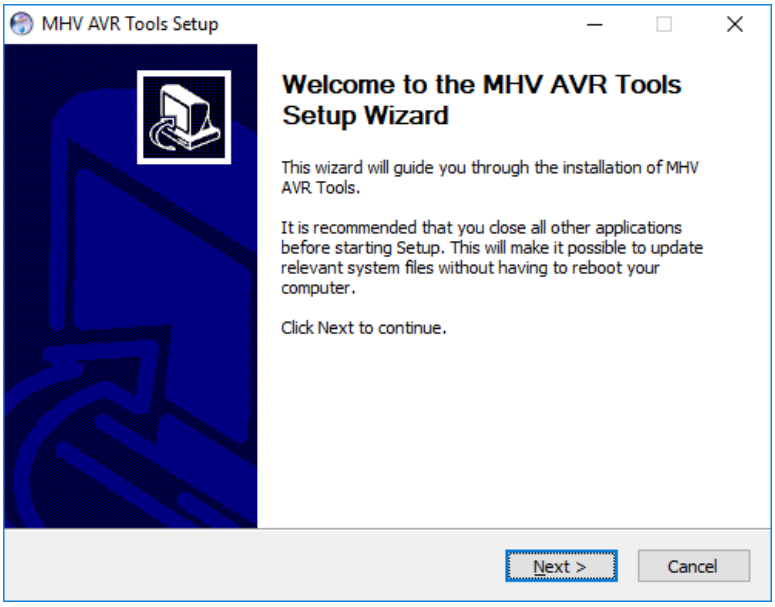


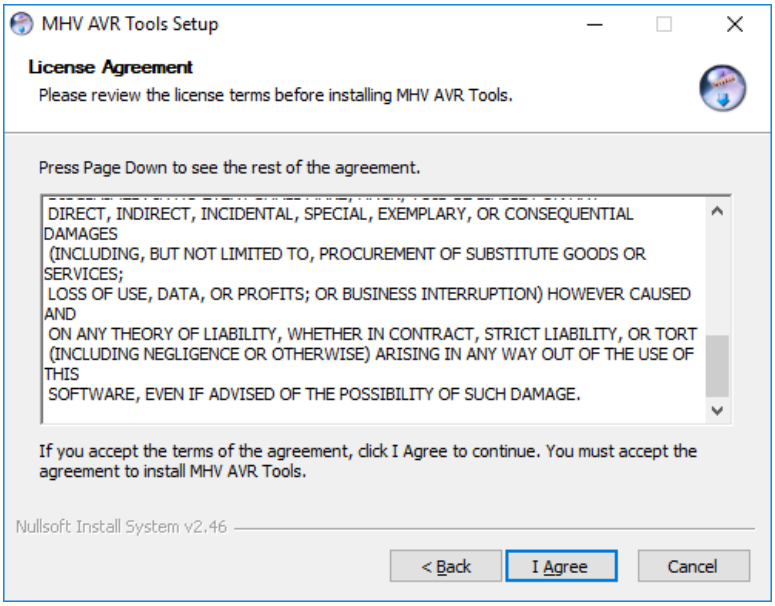
3.2.1.2 MHV_AVR_Tools – Cross-Compiler (optional)

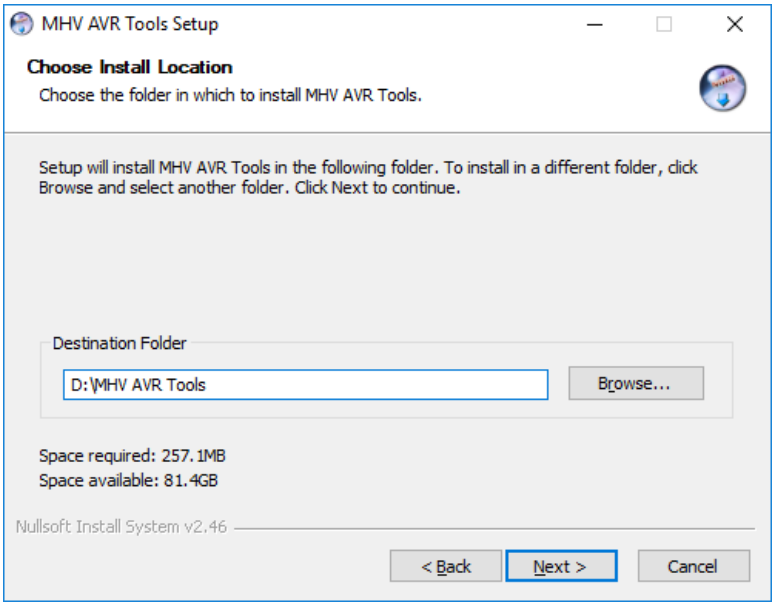
Dieses Cross-Compiler brauchen Sie nur wenn Sie Arduino für ArduPilot installieren möchten.

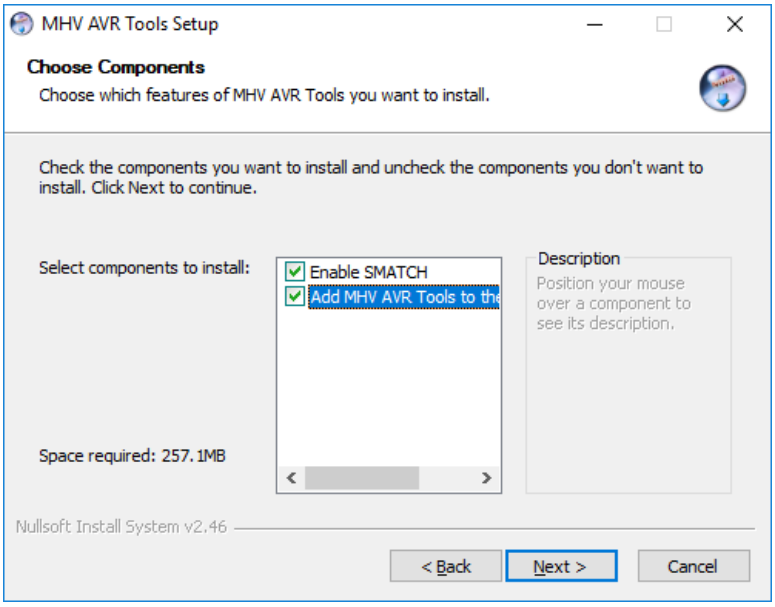
1. Laden Sie runter und installieren Sie die MHV_AVR Tools:

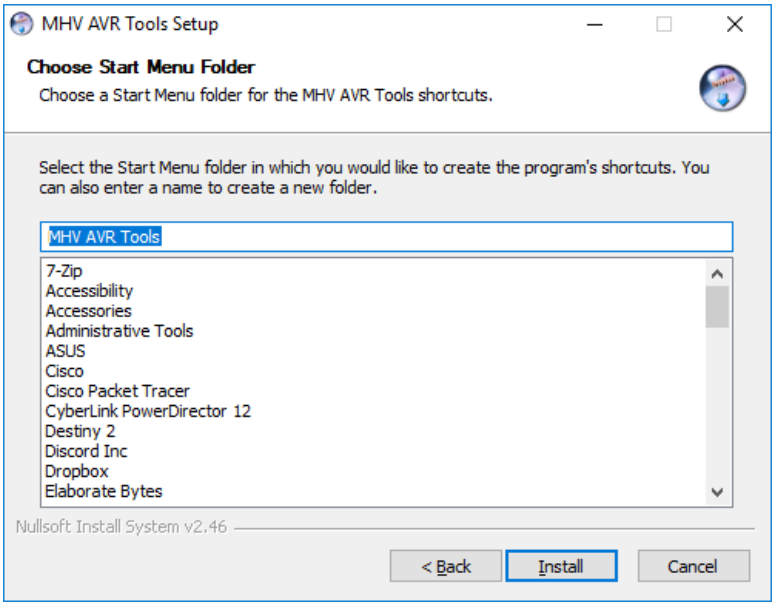
http://firmware.ardupilot.org/Tools/Arduino/MHV_AVR_Tools_20121007.exe

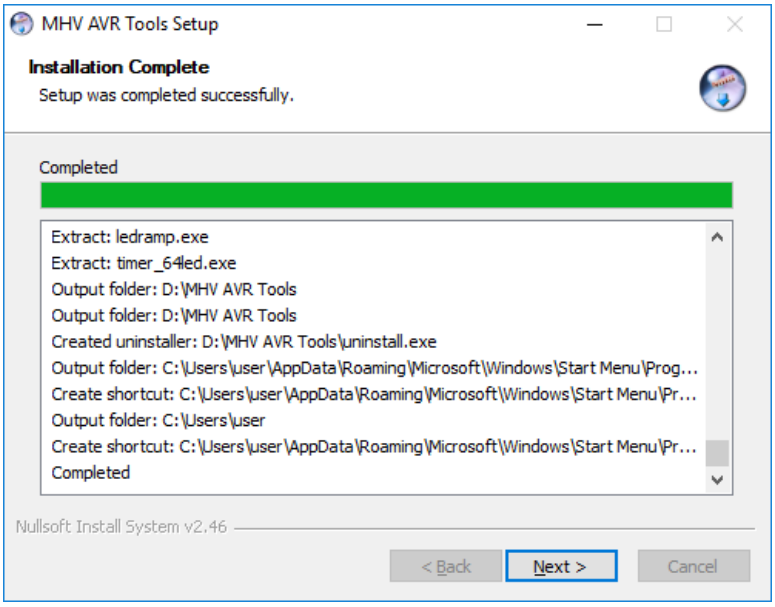


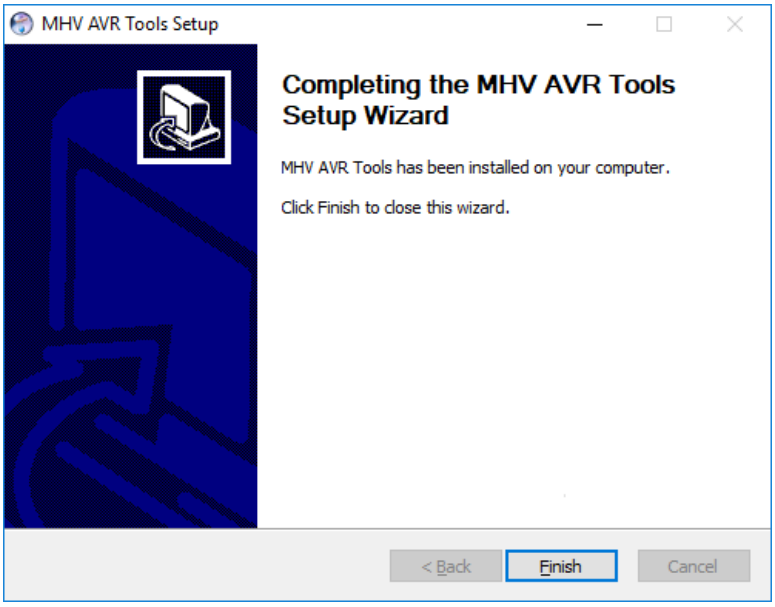

MHV AVR Tools Setup
License Agreement
Please review the license terms before installing MHV AVR Tools.
Press Page Down to see the rest of the agreement.
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install MHV AVR Tools.
Nullsoft Install System v2.46
< Back
I Agree
Cancel


MHV AVR Tools Setup
Choose Install Location
Choose the folder in which to install MHV AVR Tools.
Setup will install MHV AVR Tools in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
D:\MHV AVR Tools
Browse...
Space required: 257.1MB
Space available: 81.4GB
Nullsoft Install System v2.46
< Back
Next >
Cancel


MHV AVR Tools Setup
Choose Components
Choose which features of MHV AVR Tools you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select components to install:
Enable SMATCH
Add MHV AVR Tools to the
Description
Position your mouse over a component to see its description.
Space required: 257.1MB
Nullsoft Install System v2.46
< Back
Next >
Cancel


MHV AVR Tools Setup
Choose Start Menu Folder
Choose a Start Menu folder for the MHV AVR Tools shortcuts.
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.
MHV AVR Tools
7-Zip
Accessibility
Accessories
Administrative Tools
ASUS
Cisco
Cisco Packet Tracer
CyberLink PowerDirector 12
Destiny 2
Discord Inc
Dropbox
Elaborate Bytes
Nullsoft Install System v2.46
< Back
Install
Cancel

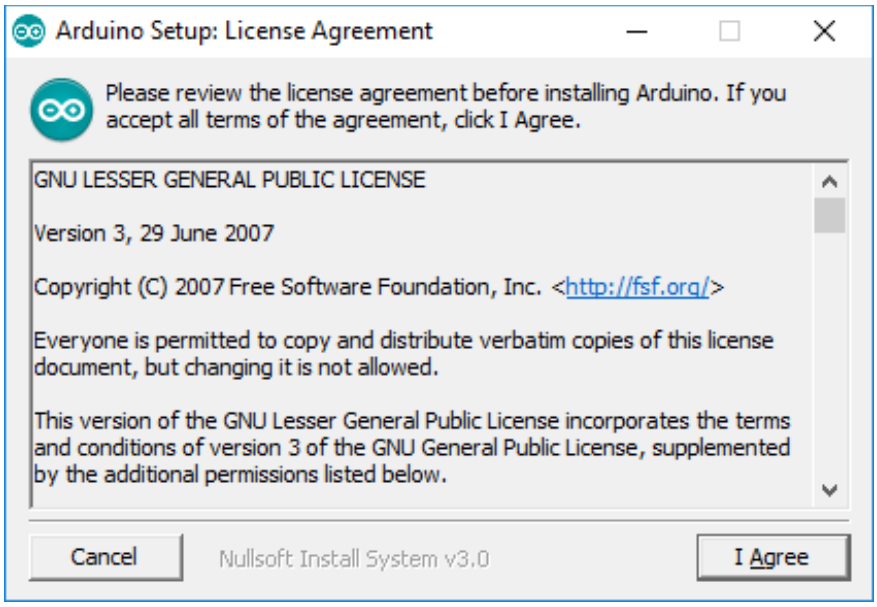

MHV AVR Tools Setup
Installation Complete
Setup was completed successfully.
Completed
Extract: ledramp.exe
Extract: timer_64led.exe
Output folder: D:\MHV AVR Tools
Output folder: D:\MHV AVR Tools
Created uninstaller: D:\MHV AVR Tools\uninstall.exe
Output folder: C:\Users\user\AppData\Roaming\Microsoft\Windows\Start Menu\Prog...
Create shortcut: C:\Users\user\AppData\Roaming\Microsoft\Windows\Start Menu\Pr...
Output folder: C:\Users\user
Create shortcut: C:\Users\user\AppData\Roaming\Microsoft\Windows\Start Menu\Pr...
Completed
Nullsoft Install System v2.46
< Back
Next >
Cancel

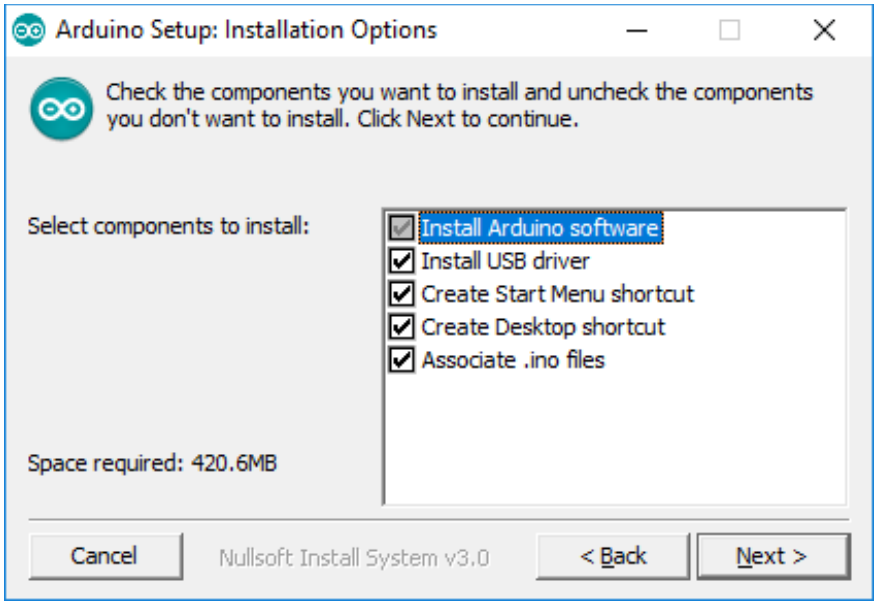

MHV AVR Tools Setup
Completing the MHV AVR Tools Setup Wizard
MHV AVR Tools has been installed on your computer.
Click Finish to close this wizard.
< Back
Finish
Cancel

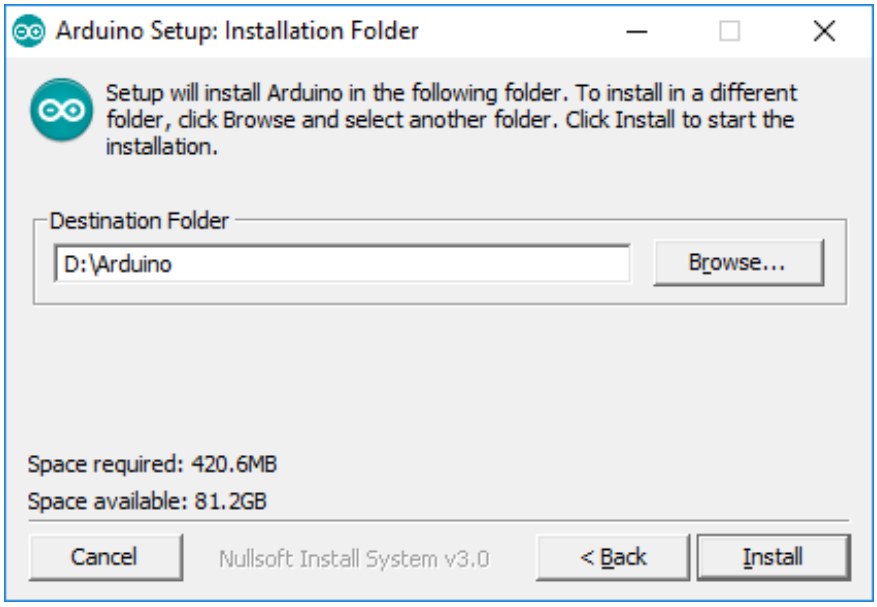
3.2.1.3 Arduino – die neuste Version (optional)

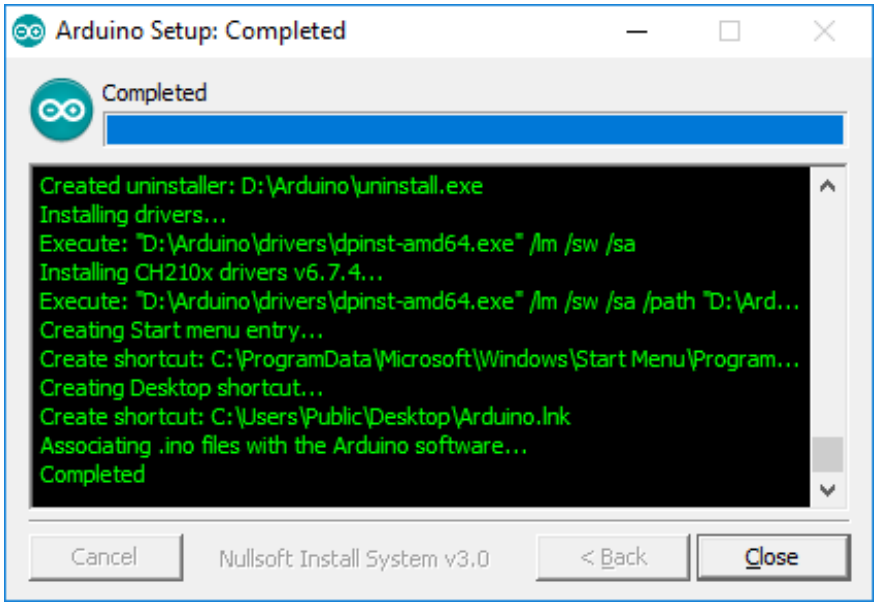
Aufgrund eines Navigationsproblems zwischen den Tabs in der für APM 2.5 angepassten Version und wenn Sie eine aktuelle Version für andere Anwendungen verwenden müssen, empfehlen wir Ihnen, zuerst, die letzte Version der Software zu installieren.

1. *https://www.arduino.cc/en/main/software*






Arduino Setup: Installation Folder
Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
Destination Folder
D:\Arduino
Browse...
Space required: 420.6MB
Space available: 81.2GB
Cancel
Nullsoft Install System v3.0
< Back
Install


Arduino Setup: Completed
Completed
Created uninstaller: D:\Arduino\uninstall.exe
Installing drivers...
Execute: "D:\Arduino\drivers\dpinst-amd64.exe" /lm /sw /sa
Installing CH210x drivers v6.7.4...
Execute: "D:\Arduino\drivers\dpinst-amd64.exe" /lm /sw /sa /path "D:\Ard...
Creating Start menu entry...
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Program...
Creating Desktop shortcut...
Create shortcut: C:\Users\Public\Desktop\Arduino.lnk
Associating .ino files with the Arduino software...
Completed
Cancel
Nullsoft Install System v3.0
< Back
Close

3.2.1.4 Arduino – für ArduPilot (optional)

Arduino wird nur benutzt um den Code zu ändern und verifizieren, wenn Sie den Code direkt auf Linux ändern wollen, brauchen Sie nicht Arduino zu installieren.

1. Das ArduPilot-Arduino Paket runterladen und entpacken da wo Sie möchten:

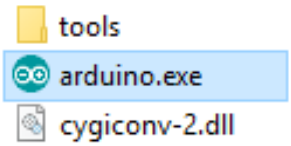
   http://firmware.ardupilot.org/Tools/Arduino/ArduPilot-Arduino-1.0.3-gcc-4.8.2-windows.zip

   Hinweis: Dies ist ein spezielles Arduino-Arduino-Paket, das gcc 4.8.2 enthält.

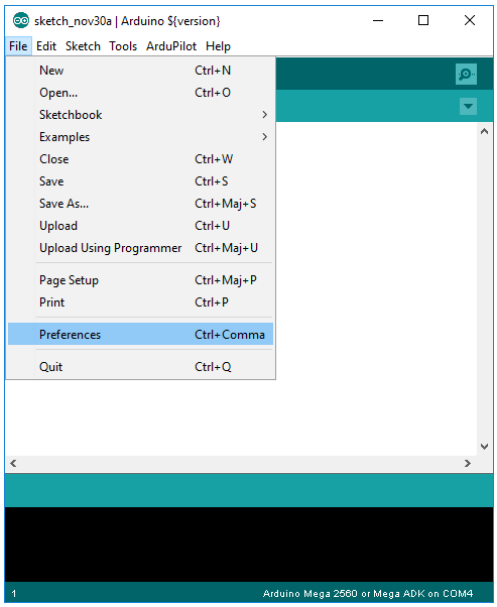
2. Gehen Sie zu Ihrem Arduino-Ordner.

3. Um sicherzustellen, dass keine Probleme mit den Bibliotheken geschehen kann, kopieren Sie den Inhalt des Ardupilot-3.2.1-Ordners „libraries“ in den Ordner „libraries“ des Arduino-1.0.3.

4. Doppelklick auf Arduino.exe.



5. Dateimenü wählen und „Preferences“ anklicken.

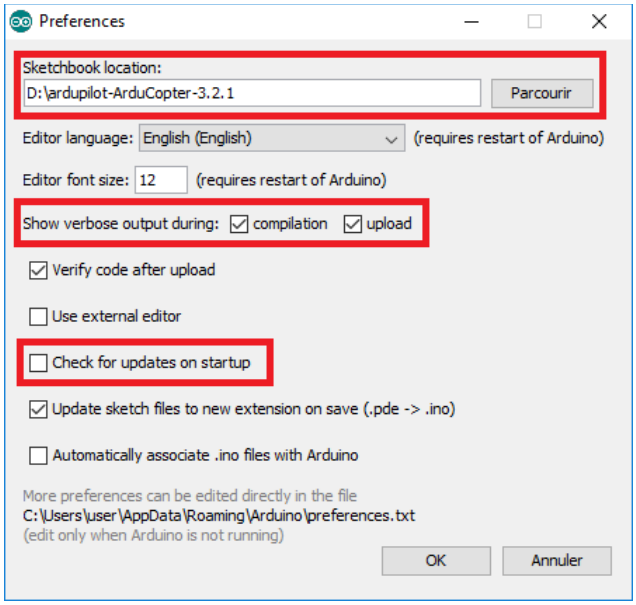


6. Stellen Sie den Sketchbook-Speicherort an Ihrem ArduPilot-Verzeichnis.

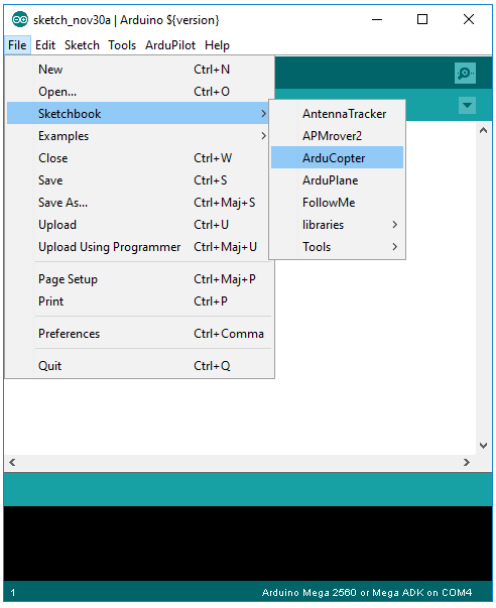
7. Setzen Sie „Show verbose output“ für die Kompilierung und das Hochladung.

8. NICHT nach Updates beim Start suchen ... (Denken Sie daran, dies ist eine spezielle Version nur für ArduPilot).

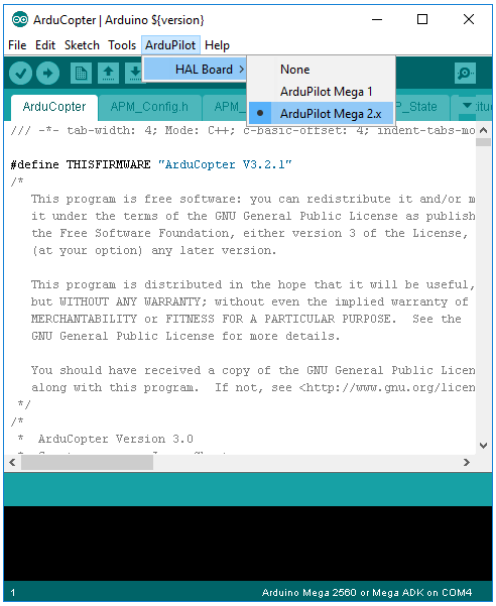
9. Klicken Sie auf OK, und schließen Sie Arduino.



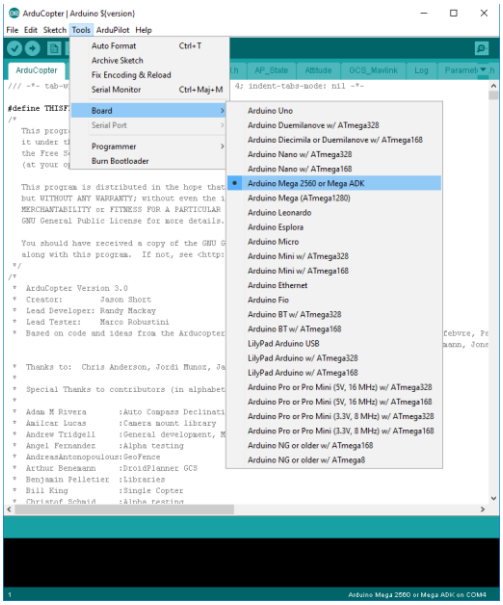
10. Öffnen Sie wieder ArduPilot und klicken Sie unter „File“ auf „Sketchbook“ und dann auf ArduCopter.



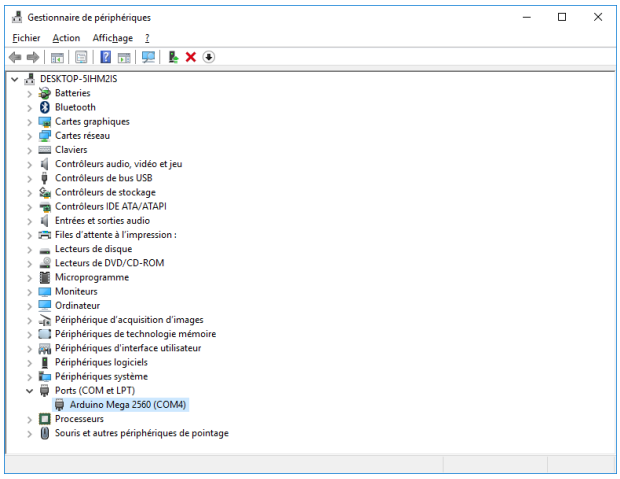
11. Klicken Sie nach dem Laden auf „ArduPilot“ und wählen Sie ArduPilot Mega 2.x aus den HAL-Optionen (weil wir mit ein APM 2.5 arbeiten).

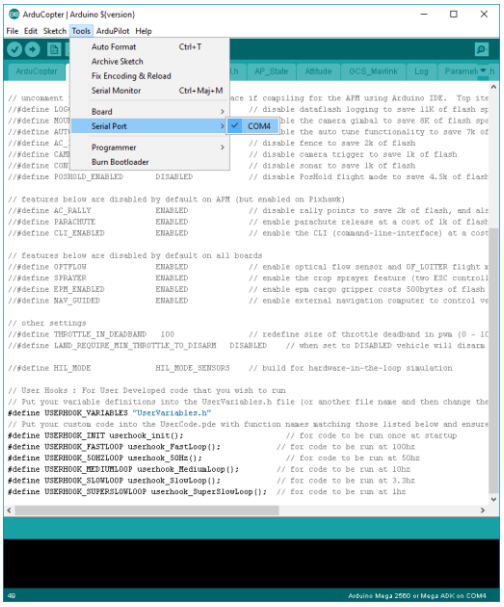


12. Klicken Sie dann auf „Tools“ und wählen Sie „Arduino Mega 2560 or Mega ADK“ aus den Boards.



13. Wählen Sie als Nächstes im Menü „Tools“ → „Serial Port“ (Seriellen Anschluss), den Anschluss, mit dem Ihr APM verbunden ist. In unserem Fall war es COM4, aber schauen Sie unter Gerätemanager → Ports nach, um es auf Ihrem System herauszufinden.



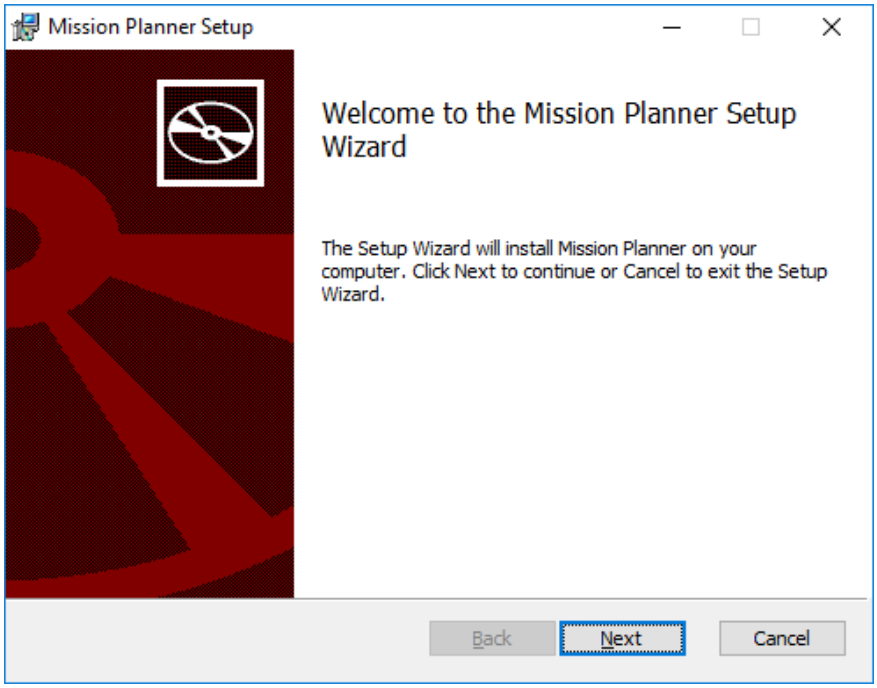


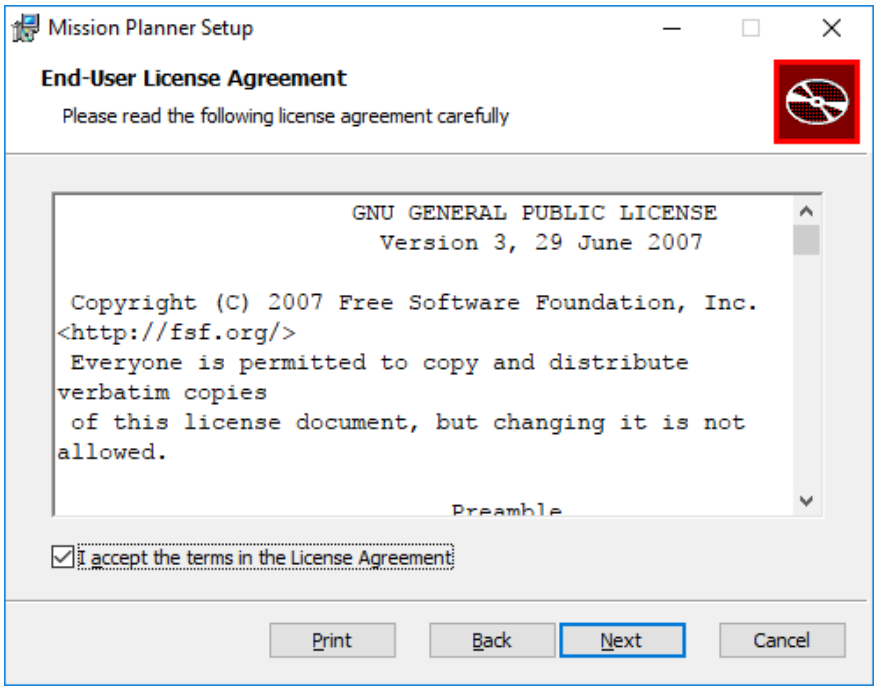
14. Sie sind jetzt bereit um den Code zu editieren (sehen Kapitle 5).

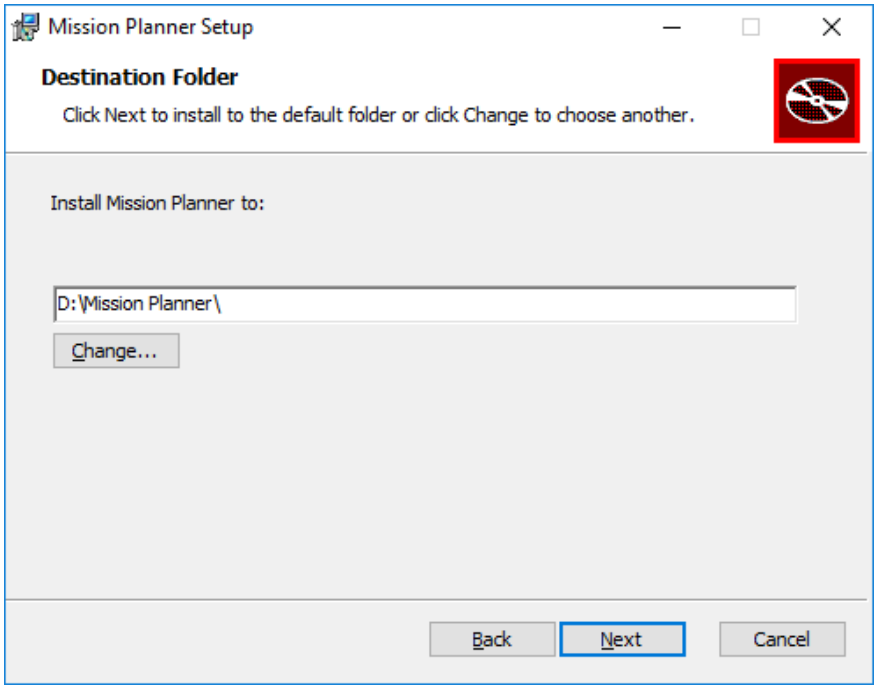
3.2.1.5 Mission Planner

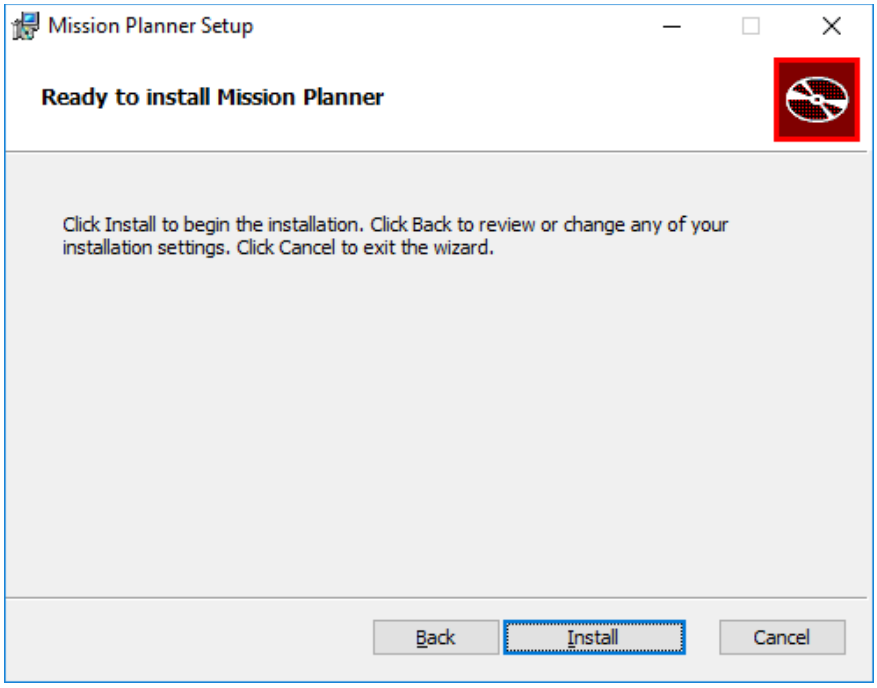
1. Laden Sie die Software auf diese Adresse herunter und führen Sie das Installationsprogramm aus:

   http://firmware.ardupilot.org/Tools/MissionPlanner/MissionPlanner-latest.msi

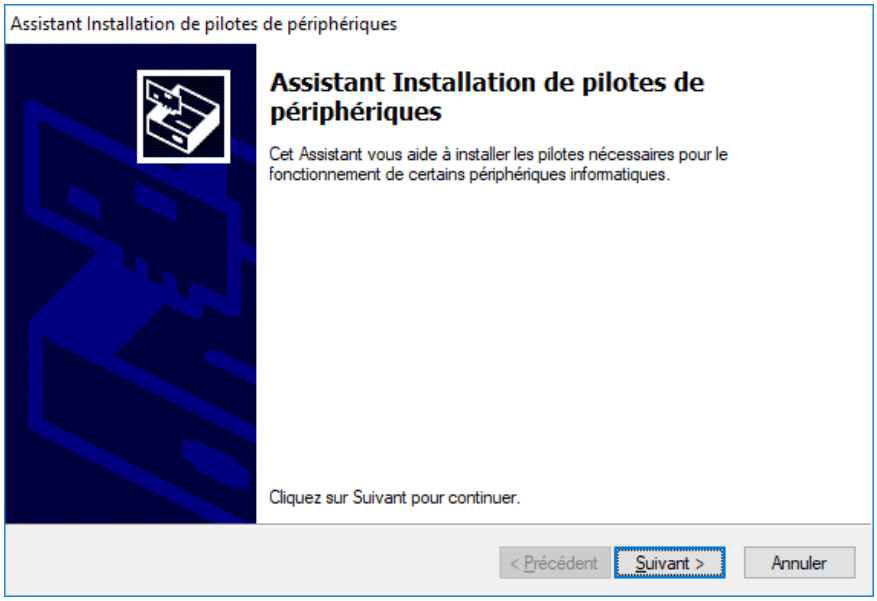


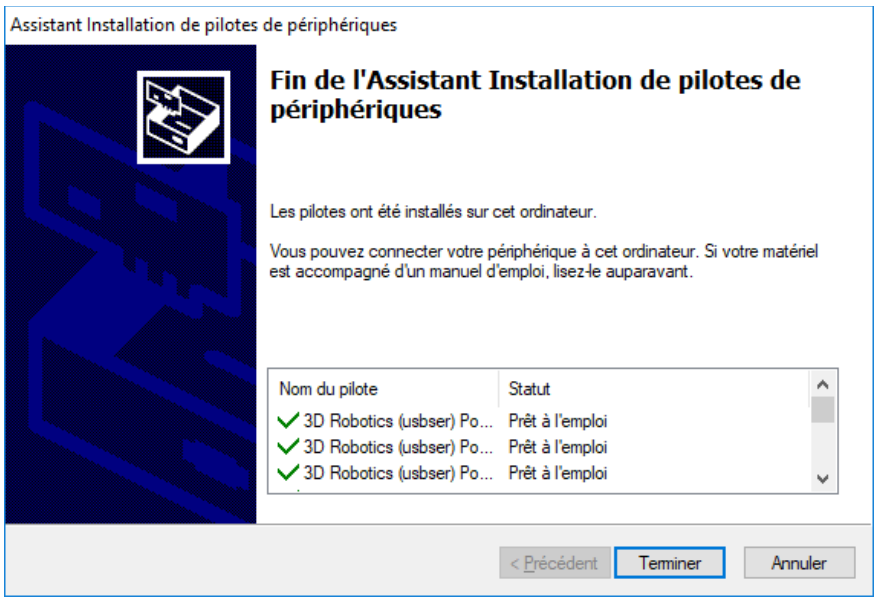


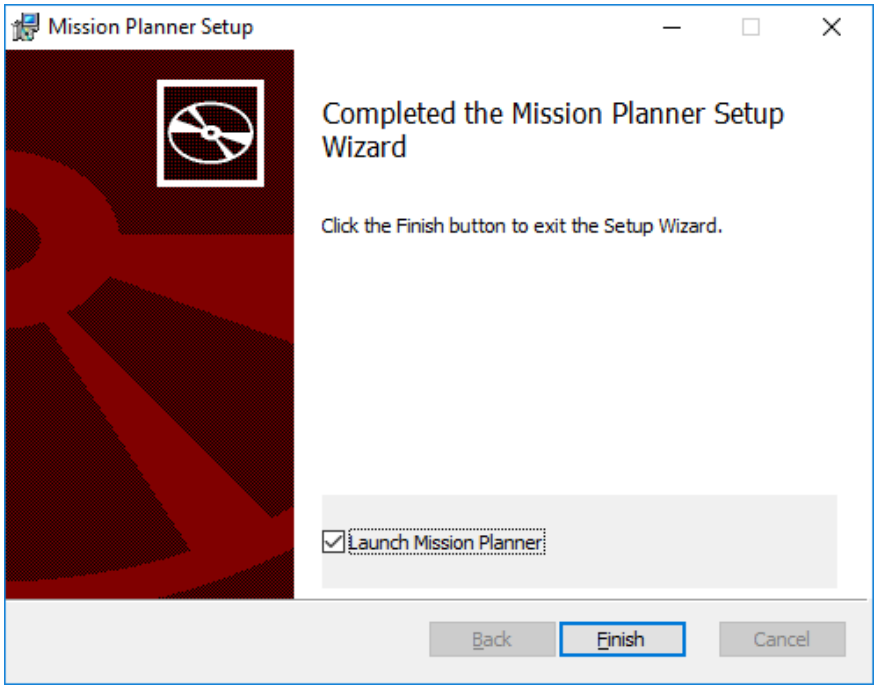




2. Sie werden automatisch nach der Berechtigung gefragt, Treiber zu installieren:


Assistant Installation de pilotes de périphériques
Assistant Installation de pilotes de périphériques
Cet Assistant vous aide à installer les pilotes nécessaires pour le fonctionnement de certains périphériques informatiques.
Cliquez sur Suivant pour continuer.
< Précédent
Suivant >
Annuler


Assistant Installation de pilotes de périphériques
Fin de l'Assistant Installation de pilotes de périphériques
Les pilotes ont été installés sur cet ordinateur.
Vous pouvez connecter votre périphérique à cet ordinateur. Si votre matériel est accompagné d'un manuel d'emploi, lisez-le auparavant.
Nom du pilote
Statut
3D Robotics (usbser) Po...
Prêt à l'emploi
3D Robotics (usbser) Po...
Prêt à l'emploi
3D Robotics (usbser) Po...
Prêt à l'emploi
< Précédent
Terminer
Annuler



3.2.2 Linux

1. Aktualisieren Sie zuerst den Paket-Manager:

   sudo apt-get update

   sudo apt-get upgrade

2. Um die Installation zu starten, benötigen wir git (es dauert eine wenig Zeit, bis es etwas anzeigt):

   sudo apt-get -qq -y install git

3. Gehen Sie dann in den Ordner, in dem Sie den Code speichern möchten, und geben Sie folgendes ein:

   sudo git clone -b ArduCopter-3.2.1 https://github.com/ArduPilot/ardupilot.git

4. Gehen Sie ins Ordner für den ArduCopter und initialisieren Sie git:

   cd ardupilot/ArduCopter/

   sudo git init

5. Dann gehen Sie zu dem Ordner „scripts“ und geben Sie folgendes ein:

   sudo ../Tools/scripts/install-prereqs-ubuntu.sh -y

   sudo apt-get install arduino arduino-core

   sudo make configure

6. Installieren Sie den Cross-Compiler:

   sudo apt-get install gawk

   sudo apt-get install gcc-avr avrdude avr-libc binutils-avr

   sudo apt-get install python-serial python-wxgtk2.8 python-matplotlib python-opencv python-pexpect python-scipy

7. Jetzt sind Sie bereit um den Code zu bearbeiten, zu kompilieren und auf dem Board zu laden (sehen Kapitle 5.1).

Falls es Probleme gibt:

- Wenn die Arduino-Installation nicht mit dem Befehl funktioniert, installieren Sie die normale Version von Arduino unter Linux über die Arduino-Website:

  https://www.arduino.cc/en/Main/Software

- Wenn die git-Befehle automatisch eine neuere Version installiert, löschen Sie diese ArduPilot-Version und laden Sie den Zip mit der finalen Version für APM 2.5 (ArduCopter-3.2.1) herunter:

  https://github.com/diydrones/ardupilot/archive/ArduCopter-3.2.1.zip

4 Kalibrierung und erste Verwendungen

Vor die erste Verwendung muss man ein Paar Kalibrierung durchführen und dafür muss den APM mit Mission Planner verbunden sein.

4.1 Verbindung zum Controller

1. Zuerst schließen Sie mit ein USB-Kabel den Kontroller mit dem Rechner an.

2. Um herauszufinden, welchem Port der Controller zugewiesen ist, siehe Kapitel 3.2.1.4.

3. Wählen Sie den Port aus der Liste (hier haben wir den COM4-Port) und setzen Sie die Baudrate auf 115200.

4. Klicken Sie auf „Connect“.

4.2 RC-Kanäle kalibrieren

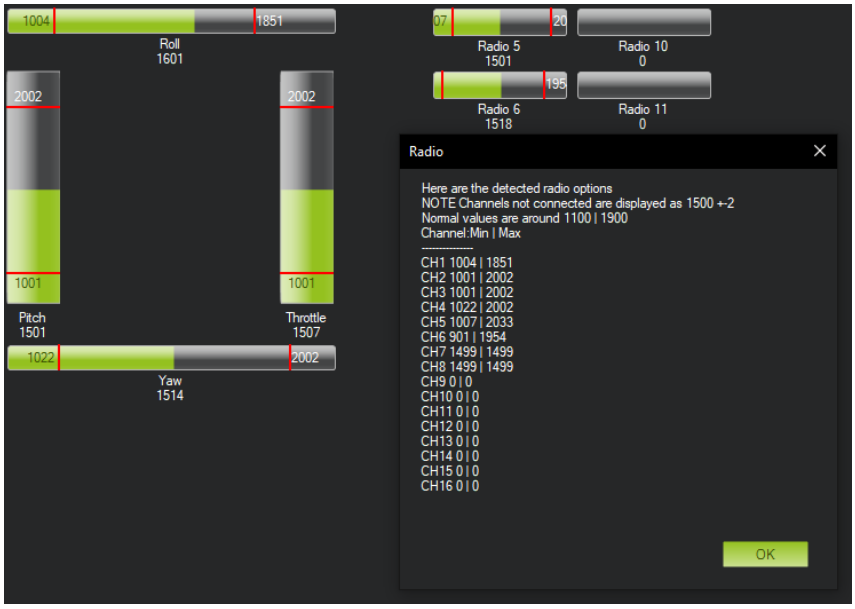
Um die RC-Kanäle zu kalibrieren, ist es zuerst notwendig, dass der Receiver und dem Controller zusammen verbunden sind und dass die Übertragung zwischen dem Controller und ihm hergestellt ist (kein Piepton von der Fernbedienung mehr und den Receivers LED leuchtet).

Dann gehen Sie einfach zum Mission Planner und klicken Sie auf „INITIAL SETUP“ -> „MADATORY HARDWARE“ -> „RADIO CALIBRATION“.

Hier sind 14 Kanale repräzentiert, den APM 2.5 hat nür 8 Eingang, deswegen sind die 6 anderen Kanale auf 0. Unsere receiver hat 6 Kanale, dass heisst die Kanale 7 und 8 werden immer nahe 1500.

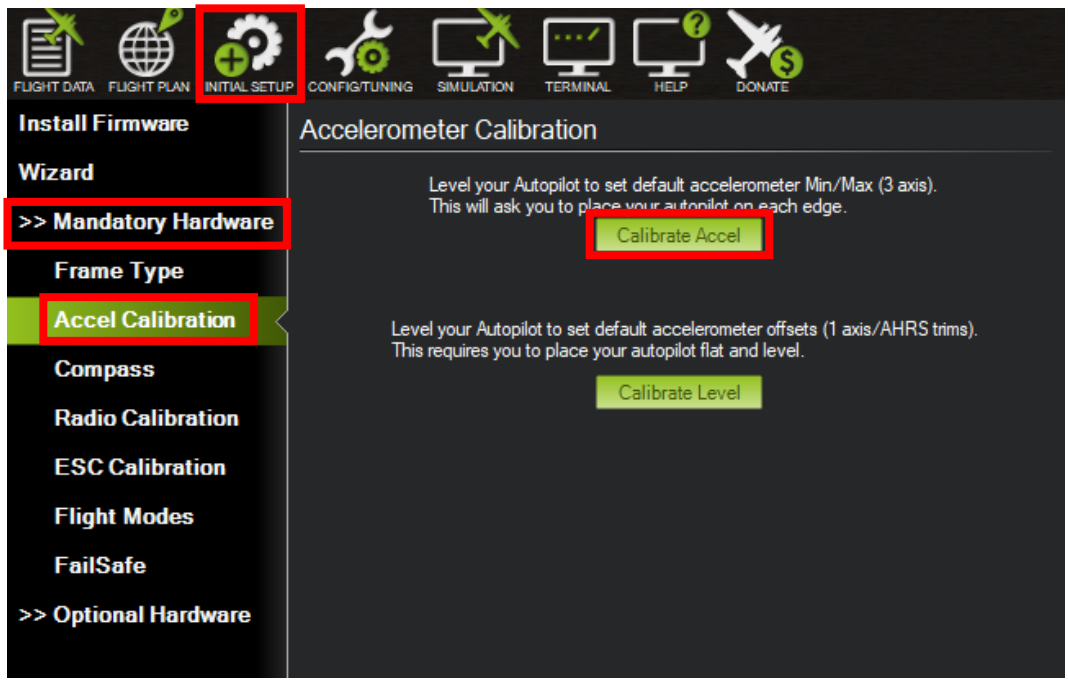
Nachdem Sie einfach auf „Calibrate Radio“ geklickt haben, folgen Sie den Anweisungen auf dem Bildschirm. Vergessen Sie nicht, den Throttle am Ende auf das Minimum zu stellen um Unfall zu vermeiden falls die gesamte Drohne an den APM angeschlossen ist.

Am Ende sollte dieses Fenster erscheinen (sie zeigt welche Intervallwert jeden Kanal hat):



4.2.1 Kalibrierung die Beschleunigungsmesser

Um die Beschleunigungsmesser zu kalibrieren, müssen Sie im Mission Planner den Controller anschließen, gehen Sie dann zu INITIAL SETUP -> MANDATORY HARDWARE --> ACCEL CALIBRATION und klicken Sie auf „Calibrate Accel“.



Folgen Sie danach einfach die angezeigten Anweisungen.

4.2.2 Kalibrierung den ESC

Es ist unmöglich ESCs über Mission Planner zu kalibrieren. Trennen Sie die Drohne von Ihrem Computer. Die Drohne muss von der Batterie versorgt sein und die ESC müssen an die Motoren angeschlossen sein. Bevor Sie die Batterie anschliessen, müssen Sie den Throttle auf den Maximum stellen und halten. Starten Sie die Drohne und warten Sie bis die LEDs zyklisch „blau-gelb-rot“ blinken, das bedeutet, dass Sie sich im Kalibriermodus bei nächsten Start befinden.

Mit den Throttle immer noch auf Maximum, trennen Sie die Batterie und schließen Sie sie wieder an, lassen die Motoren laufen, bis die Steuerung einen deutlichen Signalton abgibt.

Spielen sie ein bisschen am Gas, so dass der ESC das Kraftfeld des Gaspedals kennt und dann stellen Sie den Throttle auf Minimum ein, trennen Sie die Batterie, halten Sie die Throttle auf dem Minimum und schließen die Batterie wieder an.

Wenn die LEDs wieder normal blinken, ist die Kalibrierung den ESC erfolgreich.

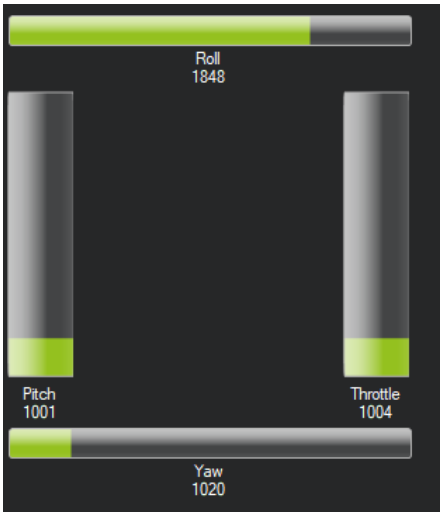
Auf dieser Seite finden Sie alle detaillierten Informationen und ein Video:

http://ardupilot.org/copter/docs/esc-calibration.html

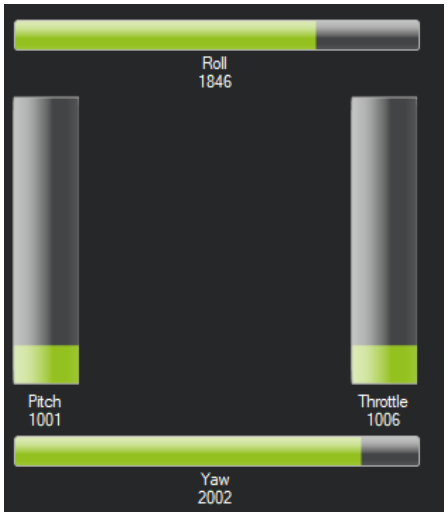
4.2.3 Arming und disarming der Drohne

Um die Drohne in den Zustand „armed“ zu bringen, muss den Pitch, Throttle und Yaw auf das Minimum und den Roll auf das Maximum gesetzt sein. Um die Drohne in den Zustand „disarmed“ zu bringen, muss den Pitch, Throttle auf das Minimum und den Roll, Yaw auf das Maximum gesetzt sein. Sie müssen jeweils ca. 5 Sekunden warten bis die Drohne disarmed oder armed wird.

disarming



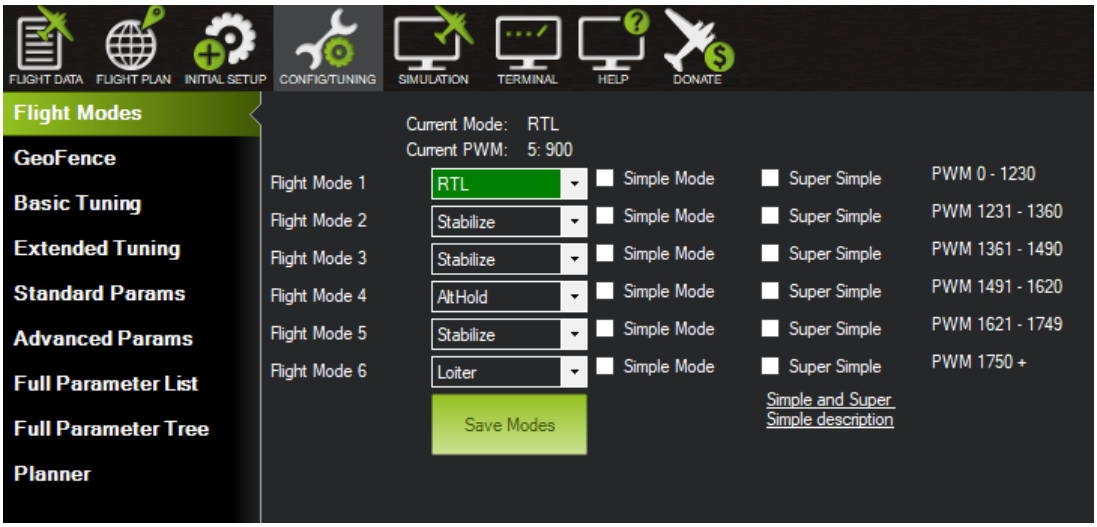
arming



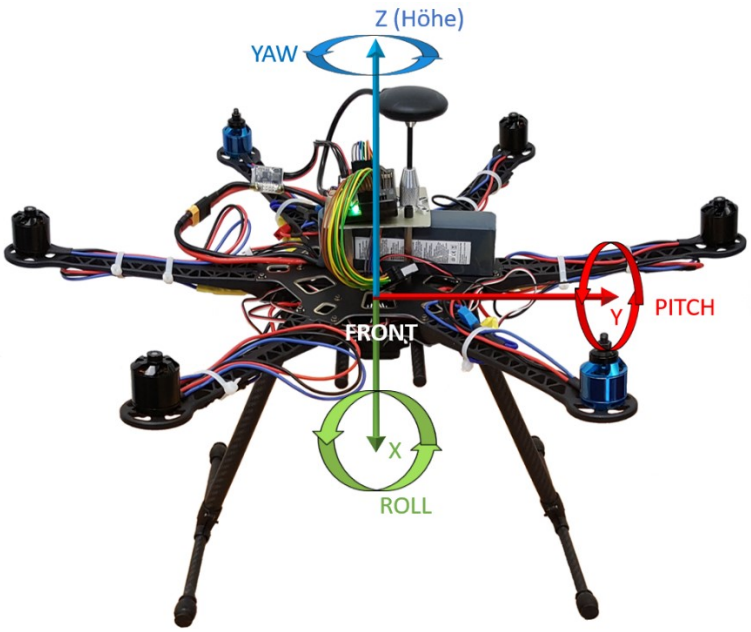
Sie können die Drohne nur in den Modi „stabilize“, „ACRO“, „AltHold“, „Loiter“ und „PosHold“ armen (achten Sie darauf, dass Sie nicht in „Auto“ Modus sind).

Sobald die Dronhe armed ist, muss man nur noch den Throttle erhöhen damit die Drohne abfliegt. Die verschiedenen Flugmodi sind konfigurierbar, Sie können sie im Mission Planner konfigurieren und während den Flug mit dem Kanal 5 auswählen.

Es gibt 6 konfigurierbare Modi, defaultmässig können Sie mit Kanal 5 von Ihre Fernbedienung den Modus wählen. Wenn Sie diesen Kanal auf einem Schalter haben, dann kann die Anzahl von Modi, die Sie wählen können, begrenzt sein.



Für die Steuerung der Drohne, ist es wichtig zu verstehen was „Roll“, „Pitch“ und „Yaw“ sind. „Throttle“ erhebt die Drohne (bezieht sich auf die Höhe, „Altitude“ auf diese Abbildung).



5 Änderungen

Um Änderungen am Code und am Verhalten der Drohne vorzunehmen, gibt es zwei Vorgehensweisen: mit dem Code oder mit Mission Planner.

5.1 Durch den Code

Durch die Änderung des Codes haben wir die Möglichkeit, Funktionen mit verschiedenen Frequenzen (1 Hz, 3,3 Hz, 10 Hz, 50 Hz und 100 Hz) auszuführen und eine die nur am Initialisierung ausgeführt wird.

Alle Änderungen, die in diesem Unterkapitel besprochen werden, beziehen sich auf die Dateien im Ordner „ArduCoper“ von „ArduPilot-3.2.1“.

5.1.1 Vorbereitung

Damit Änderungen am Code berücksichtigt werden können, müssen die verwendeten Funktionen und die Header-Datei deklariert werden.

Öffnen Sie dafür das Dokument „APM_Config.h“ und entfernen Sie die Kommentarzeichnen den Deklarationen von Usermethodes und von der Datei „UserVariables.h“ wie im Screenshot angezeigt.

```
// User Hooks : For User Developed code that you wish to run
// Put your variable definitions into the UserVariables.h file (or another file name and then change the #define below).
#define USERHOOK_VARIABLES "UserVariables.h"
// Put your custom code into the UserCode.pde with function names matching those listed below and ensure the appropriate #define below is uncommented below
#define USERHOOK_INIT userhook_init();                    // for code to be run once at startup
#define USERHOOK_FASTLOOP userhook_FastLoop();           // for code to be run at 100hz
#define USERHOOK_50HZLOOP userhook_50Hz();                // for code to be run at 50hz
#define USERHOOK_MEDIUMLOOP userhook_MediumLoop();       // for code to be run at 10hz
#define USERHOOK_SLOWLOOP userhook_SlowLoop();           // for code to be run at 3.3hz
#define USERHOOK_SUPERSLOWLOOP userhook_SuperSlowLoop(); // for code to be run at 1hz
```

Ausserdem, muss die Art von Drohne angeben sein. Entfernen Sie dafür die Kommentarzeichnen aus die „define FRAME_CONFIG“ Kommando am Anfang dieser Datei. Wir verwenden einen HEXA_FRAME, so wird es aussehen:

```
#define FRAME_CONFIG HEXA_FRAME
/*  options:
 *  QUAD_FRAME
 *  TRI_FRAME
 *  HEXA_FRAME
 *  Y6_FRAME
 *  OCTA_FRAME
 *  OCTA_QUAD_FRAME
 *  HELI_FRAME
 *  SINGLE_FRAME
 *  COAX_FRAME
 */
```

5.1.2 Bearbeitung

Um den Code zu bearbeiten, öffnen Sie die Dateien „UserCode.pde“ (oder „UserCode.ino“, wenn Sie das Projekt bereits mit Arduino kompiliert haben) und „UserVariables.h“.

Abhängig von der Funktion, in der Sie Ihren Code schreiben, wird sie entweder nur bei der Initialisierung oder bei verschiedenen Frequenzen (1 Hz, 3,3 Hz, 10 Hz, 50 Hz und 100 Hz) ausgeführt.

Einige Beispiele für nützliche Anweisungen:

- Um eine Nachricht an die Ground Control Station zu senden (für wichtige Meldungen, wie Fehler usw.):

  gcs_send_text_P(SEVERITY_HIGH, PSTR("Text"));

- Um eine Nachricht an die Konsole zu senden:

  hal.console->printf_P(PSTR("Text\n"));

- Um einen Ausgang des Controllers zu steuern (zum Beispiel für eine LED):

  hal.gpio->write(OUTPUT_NUMBER, HAL_GPIO_LED_ON);

- Um den Wert eines Eingangskanals (hier Kanal 6 von Empfänger) zu lesen:

  int value = g.rc_6.radio_in;

5.1.3 Einsetzung

Dazu müssen Sie sich im ArduPilot-3.2.1 „ArduCopter"-Ordner befinden. Für das erste Mal, ist es nötig eine „*sudo make configure*" vor die folgenden Anweisungen durchzuführen. Achten Sie darauf, dass das APM an dem Linux Maschine angeschlossen ist falls sie eine Virtuelle Maschine benutzen.

1. Um den Code zu kompilieren:

   sudo make

2. Um den Code zu kompilieren und auf den Kontroller hochladen:

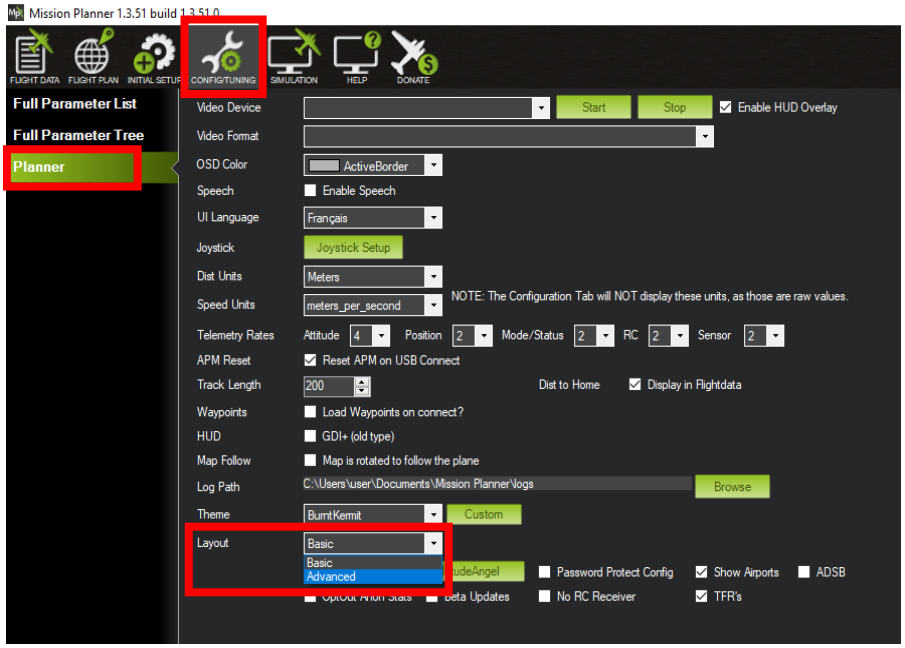
   sudo make upload

5.2 Durch Mission Planner

Mit Mission Planner ist es möglich die Steuerung von Ausgängen zu ändern und ein Flight Plan zu definieren. Dafür muss das erweiterte Layout aktiviert sein und den Kontroller mit Mission Planner gebunden sein.

5.2.1 Aktivierung das erweiterte Layout

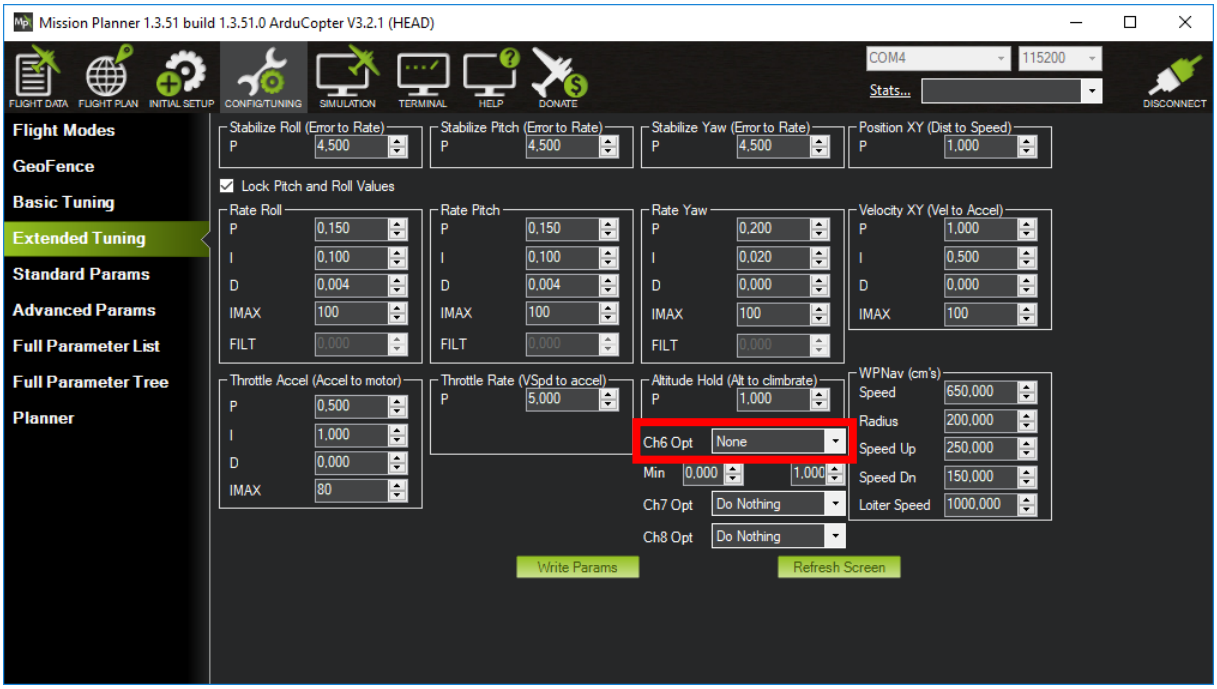
Mit dieser Option haben Sie Zugriff auf alle Konfigurationsoptionen und auf die Konsole, mit der Sie die gesendeten Nachrichten vom Befehl „*hal.console->printf_P();*" sehen können.



5.2.2 Steuerung einen PWM-Ausgange

Wenn der Kontroller mit Mission Planner verbunden ist, klicken Sie auf das Menü „Config/Tuning“ und dann auf den Reiter „Extended Tuning“.

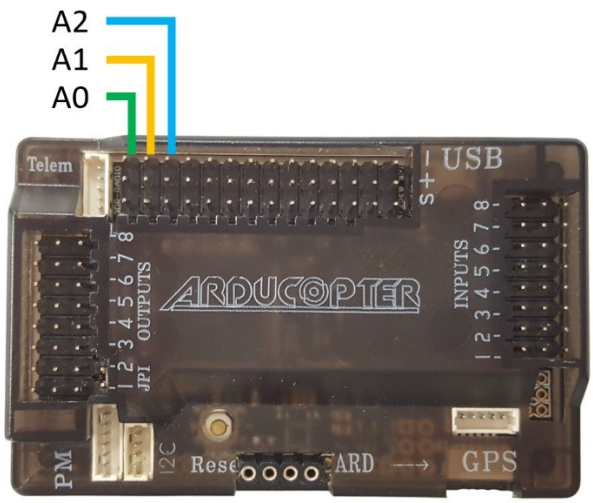
Setzen Sie die Option Kanal 6 auf „None“. Wir benützen den Kanal 6, weil die 4 ersten sind für die Steuerung reserviert und den Kanal 5 um den Flight Mode zu wechseln. Falls sie mehr als 6 Kanale zu Verfügung haben, können Sie diese benutzen.



Gehen Sie jetzt zu Menü „Initial Setup“ und dann ins Reiter „Optional Hardware“ und „Camera Gimbal“.

Für die Option „Tilt“ wählen Sie die Ausgangspins die Sie verwenden möchten (RCXX = AXX – Siehe Folgende Abbildungen). Wir haben RC11 gewählt, und der Motor mit A11 verbunden. Das Option „stabilise Tilt“ bedeutet: wenn sich die Drohne lehnt, kompensiert automatisch der Motor.

Definieren Sie jetzt den Input Channel. Mit „Servo Limits“, „Angle Limits” und « Reverse” können Sie den Verhältnisse des Ausgangs im Detail kontrollieren.



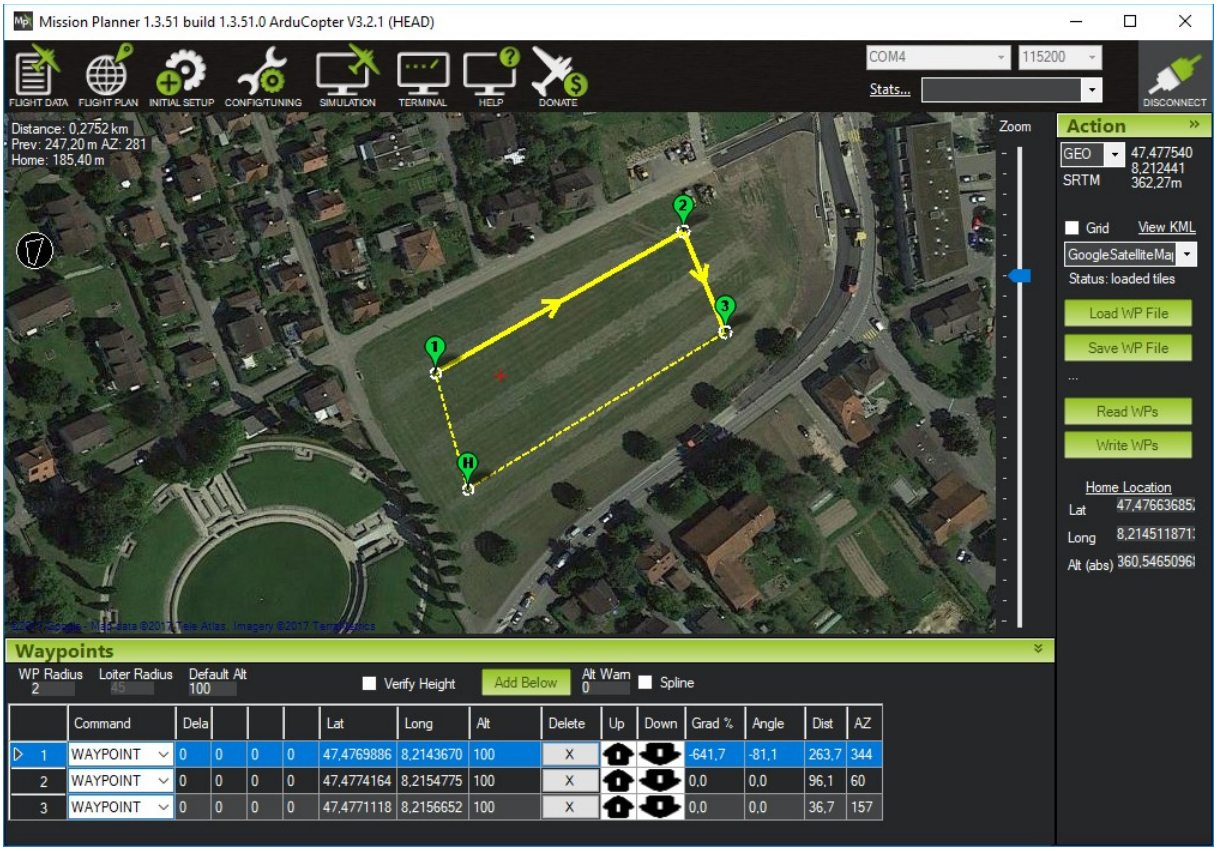
Sie brauchen nicht etwas zu kompilieren, weil diese Änderungen sind direkt auf den Kontroller gemacht. Falls wirklich ein Problem gibt, trennen Sie den Kontroller von ihrem Rechner und verbinden Sie sie wieder zusammen.

Jetzt können Sie mit dem Potentiometer der zu Ihren 6. Kanale, ein PWM-Ausgang Steuern.

5.2.3 Flight Plan

Um ein Flight Plan zu definieren, gehen Sie in das Menü „Flight Plan“.

Klicken Sie irgendwo auf die Karte um ein Waypoint zu definieren. Unten haben Sie alle Informationen über des Flight Plan und die verschiedene Waypoints. Diese können manuell geändert werden und Sie können eine bestimmte Aktion programmieren in den Spalt „Command“.



Wenn Ihren Flight Plan bereit ist, können Sie es unter „Action“ (Recht von Fenster) und „Save“ den Flight Plan speichern. Hier wird es auch mit „Load“ wieder geladen.

Um den Flight Plan auf den Kontroller zu senden klicken Sie auf „Write WPs“. Mit „Read WPs“ lesen Sie den Flight Plan der aktuell auf den Kontroller ist.

Für mehrere Infos über das Thema „Flight Plan“, nützen Sie den Folgende Link:

http://ardupilot.org/planner/docs/common-planning-a-mission-with-waypoints-and-events.html#common-planning-a-mission-with-waypoints-and-events

6 Quellen

Hier sind die Quellen die wir benutzt haben um dieses Benutzermanual zu schreiben:

- ARDUPILOT, a: *Installing Mission Planner*, Online: http://ardupilot.org/copter/docs/common-install-mission-planner.html [abgerufen am 02. Oktober 2017]

- ARDUPILOT, b: *Archived: APM 2.5 and 2.6 Overview*, Online: http://ardupilot.org/copter/docs/common-apm25-and-26-overview.html [abgerufen am 04. Oktober 2017]

- ARDUPILOT, c: *Introducing Copter, Online*: http://ardupilot.org/copter/docs/introduction.html [abgerufen am 04. Oktober 2017]

- ARDUPILOT, d: *Building ArduPilot for APM2.x on Linux with Make*, Online: http://ardupilot.org/dev/docs/building-the-code-onlinux.html [abgerufen am 05. Oktober 2017]

- ARDUPILOT, e: *Electronic Speed Controller (ESC) Calibration*, Online: http://ardupilot.org/copter/docs/esc-calibration.html [abgerufen am 04. November 2017]

- ARDUPILOT, f: *Mission Planner Flight Plan*, Online: http://ardupilot.org/planner/docs/mission-planner-flight-plan.html [abgerufen am 20. Oktober 2017]

- GR33NONLINE : *Compiling ArduCopter/ArduPilot with the Arduino IDE*, Online: https://gr33nonline.wordpress.com/2016/09/26/compiling-arducopterardupilot-with-the-arduino-ide/ [abgerufen am 17. November 2017]

- MAPASSIONDUMODELISME, *Qu'est-ce que ArduCopter*, Online: https://mapassiondumodelisme.jimdo.com/technique/arducopter/ [abgerufen am 18. Oktober 2017]

- MECHSCIENCE, *Transmitter Control for Pitch, Yaw and Roll*, Online picture: http://www.mechscience.com/wp-content/uploads/2017/11/image-235.png [abgerufen am 02. Januar 2018]