

Name

class school

5.10.2023 - 23.05.2024

FÜLA

Projekt: HO(TA)S

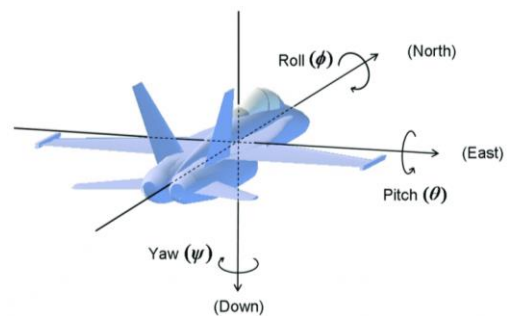
Hands on (Throttle and) Stick



Inhalt	
Projektübersicht .....	3
Hauptziel.....	3
Optionale Ziele .....	4
Optional 1 .....	4
Optional 2.....	4
Umsetzung .....	4
Projekt-Planung .....	4
Recherche 3D Model.....	4
3D Zeichnung/Anpassung .....	5
Elektronischen Komponenten .....	6
Elektronischen Umsetzung.....	6
Schematic der Frontplatine: .....	7
Schematic der Daumenplatine:.....	8
Praktische Verkabelung .....	9
Software .....	9
Finalisierung und Testphase .....	10
Quellen: .....	11
Programme .....	11
3D-Modelle.....	11
Bilder.....	11

## Projektübersicht

Ein **Hands On Throttle And Stick** (Abbildung rechts) ist ein Computereingabegerät welches optisch an die Kontrollflächen, welche in einem Flugzeug/Helikopter gefunden werden können, erinnert. Es wird hauptsächlich verwendet, um in Simulatoren das Fliegen von echten Fluggeräten zu Lernen und zu trainieren, seit einigen Jahren werden sie auch im Unterhaltungssektor zum Lenken von Flugobjekten (Flugzeuge, Hubschrauber, Raumschiffe etc.) in Computerspielen und Simulationen wie DCS, WarThunder etc. Das Hauptmodul des HOTAS, der Flightstick oder Joystick, übernimmt die Hauptsächlichen Funktionen von „Roll“ und „Pitch“ (Abbildung rechts). Dadurch dass diese Achsen in einem HOTAS analog korrigiert werden, erreicht man eine viel höhere Präzision als man mit einer Tastatur erreichen würde. Desweiteren finden sich auf dem Stick noch Knöpfe zur Eingabe von weiteren Befehlen.



## Hauptziel

Das Ziel des Projektes war es ein außerhalb der Schulzeit angefangenes Projekt auszuarbeiten und fertigzustellen. Es soll aus zwei Arduinos bestehen, wovon einer sich in der Basis befindet und ein weiterer im Joystick. Diese zwei  $\mu C$  sollen über I2C die benötigten Daten austauschen und über die USB-Schnittstelle an den PC weitergeben. Die Wahl für zwei Arduinos bietet neben den offensichtlichen Vorteilen wie, dass nun mehr Pins zur Verfügung stehen, auch den Vorteil, dass man über den Arduino in der Basis noch weitere Module über den I2C Bus, mit wenigen Zeilen Code, hinzufügen kann.

## Optionale Ziele

### Optional 1

Als optionale Ziele wurden vorerst noch die Addition eines weiteren Modules vorgesehen. Dieses Modul wäre damit der Throttle oder zu Deutsch der Schubregler. Er befindet sich in echten Fluggeräten meist zur Linken des Piloten und erlaubt eine Stufenlose Kontrolle des Schubes in Flugzeugen oder Pitch der Rotoren in Hubschraubern sowohl als Kontrolle über Drehzahl der Turbine.

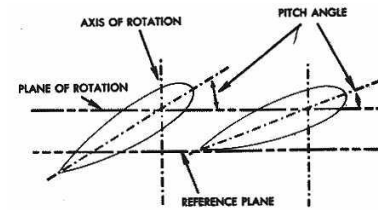


FIGURE 4.—The pitch angle of a rotor blade is the angle between the chord line and a reference plane determined by the rotor hub or the plane of rotation.

### Optional 2

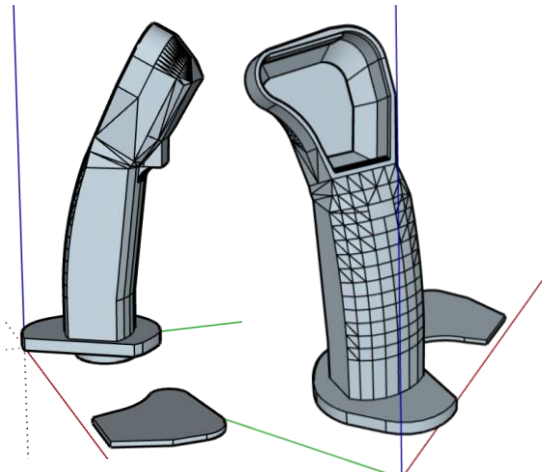
Ein weiteres Optional war die Ausgabe von Daten, welche über Dritt Software aus der Simulation geholt werden können, an Digitalen oder Analogen Anzeigen, welche in der Nähe des Joysticks montiert, werden könnten. Dieses Ziel war jedoch von Anfang an nur eine grobe Idee und wurde nie wirklich angefangen. Es wäre jedoch ein guter Platzhalter, wenn am Ende des Jahres noch viel Zeit übrig wäre. Dies war jedoch nicht der Fall und so wurde dieses optionale Ziel schon Ende Januar verworfen.

## Umsetzung

### Projekt-Planung

#### Recherche 3D Model

Gestartet wurde das Projekt mit Recherchen nach verschiedenen Joystickmodellen, um ein Gefühl dafür zu bekommen was alles nötig ist auf einem Joystick zu haben. Nach einigen ersten eigenen Zeichenversuchen (Bilder rechts), wurde sich jedoch entschieden ein open source 3D-Modellen von einem F-16 / A10 Joystick zu verwenden. Da sich herausstellte, diesen komplett selbst zu zeichnen Unmengen An Zeit aufbrauchen würde. Es wurde sich für das 3D Modell von [JFlyer81](#) entschieden, da diese auf einem 3D-Scan eines echten Joysticks basiert.



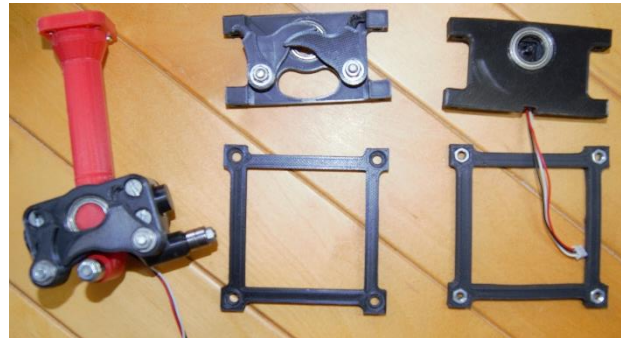
Da der schlussendliche Joystick jedoch nicht nur aus einem Joystick besteht, sondern auch einem Teil zur Montage des Joysticks, dem sogenannten Gimbal. Die Grundidee hier war ein Konstrukt, welches sich um zwei Achsen drehen kann. Bei den ersten Versuchen ein eigenes Gimbal zu designen wurde schnell festgestellt, dass es hier sehr schwer ist ein Konzept zu zeichnen welches die Kraft, welche der Bewegung des Joysticks entgegenwirkt, linear zu halten und ein spürbares „Einrasten“ in der Mittelposition zu verhindern, was optimal wäre für einen Joystick.

Nach längerer Recherche und Austausch mit einem Designer ([olukelo >>](#) Thingiverse) welcher schon ein Gimbal designt hat, welches ca. den Ansprüchen entsprach. Dieses Gimbal wurde dann als Basis verwendet und weiter modifiziert.

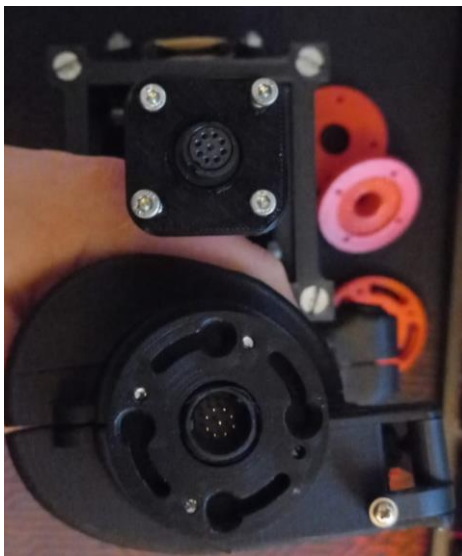
### 3D Zeichnung/Anpassung

Diese Modifikationen bezogen sich hauptsächlich auf die Anpassung der Ausbeutungsmethode von Hall Sensoren zu schlechteren/Unpräziseren Potentiometern und unterschiedlichen kleineren Anpassungen was Kugellager, Kabelschächte und Befestigungspunkte für Muttern etc. angingen. Die nächste große und wichtige Anpassung war die Änderung vom Joystick-Schaft welcher Gimbal und Joystick fest verbindet. Die originale Verbindung war eine reine

Steckverbindung welche sich reine auf Reibung der Materialien verlies, des Weiteren gab es noch eine Verbindungsoption, welche auf zwei M5 Schrauben basiert, jedoch fand ich auch diese unpassend da der Joystick in meiner Vorstellung die Möglichkeit bietet, bei Bedarf, schnell ausgetauscht zu werden.



*1 Einzelnen Komponenten des Gimbals erkennbar sind Kugellager sowie Potentiometer*



*2 Twiston Quickconnect, Gimbal oben, Joystick unten*

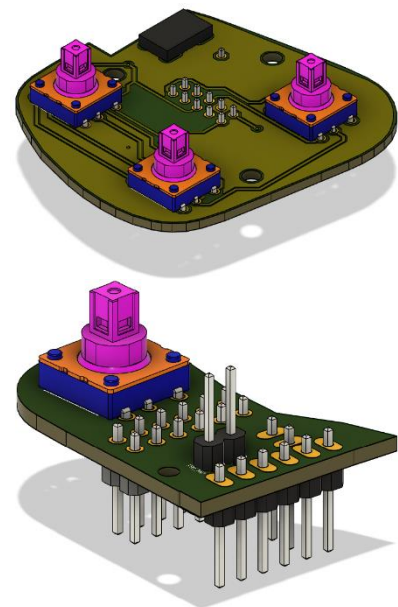
Aufgrund dessen wurde der Verbindungsmechanismus von mir in eine Variante umgezeichnet, welche verriegelt. Der Joystick wird mit einer 45° Drehung zur Basis an den Verbinder gedrückt und zurück in eine 0° Position gedreht, dabei verriegelt der Joystick sich mit dem Schaft und ist fest damit verbunden. Die elektronische Verbindung übernimmt hierbei nach einem weder gescheiterten noch geglückten Versuch mit einer Provisorischen Platine, ein Druck-Stecker, welcher normalerweise für Audiosysteme im Avionik Bereich verwendet wird (Siehe Bild links). Nachdem das Grundmodell des kompletten Joystick-Zusammenbaus stand, wurde mit der der Planung der elektrischen Komponenten gestartet.

## Elektronischen Komponenten

Die Grundidee war hier, einen eigenständigen Arduino im Joystick zu haben welcher sich darum kümmert die verschiedenen Knöpfe auszulesen und an einen anderen Arduino zu schicken welcher sich darum kümmert, dass diese Daten dann am PC ankommen. Das Auslesen der 26 Knöpfe im Joystick sollte sich noch als eine Hürde erweisen jedoch dazu später mehr. In der Basis des Joysticks befindet sich ein Arduino Micro, welcher über die Eigenschaften verfügt als eine Tastatur/Maus/Controller unter Windows zu erscheinen. Dieser liest die gesendeten Daten vom Arduino Nano aus und übergibt diese, des Weiteren gibt er noch die Werte der Potentiometer weiter.

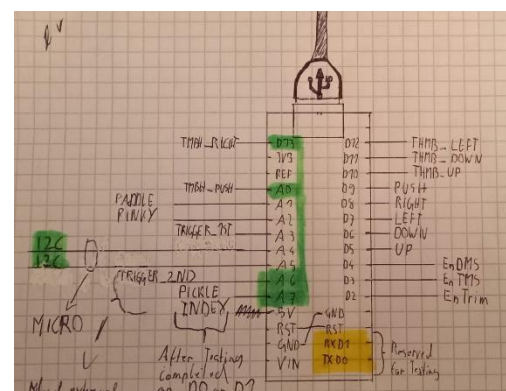
Die verwendeten Potentiometer waren hier Potentiometer welche über einen Drehbereich von ca. 90°, anstelle von den klassischen 300°-320°, hatten, was fast genau dem möglichen Drehwinkel vom Gimbal entsprach.

Die Knöpfe waren im ersten Prototyp kleine per Schrauben im Gehäuse angebrachte Halterungen für fünf 5\*5\*3 mm Knöpfe. Diese Gebilde waren jedoch sehr komplex, unzuverlässig und Fehleranfällig. Deshalb wurde sich entschieden diese Halterungen durch einen integrierten 5-Wege Knopf zu ersetzen, welcher jedoch eine Platine benötigt. Nächster Schritt war also eine beziehungsweise zwei Platinen zu designen welche dem Joystickgehäuse entsprachen, im Gehäuse wurde ein Einschnitt gemacht, damit die Platinen fest im Gehäuse verbaut werden können. Dieser Vorgang wurde für die seitliche Platine wiederholt und dementsprechend fortgeführt. Da es eine Front-Platine gibt und eine Thumb-Platine, welche sich an den unterschiedlichen Positionen befinden (links wo der Daumen ist und oben auf der Frontplatte).



## Elektronischen Umsetzung

Da der Arduino Nano nur über 14 digitale und 8 Analoge Pins verfügt, davon jedoch schon zwei für die Kommunikation der zwei Arduinos über I2C verloren gehen, bleiben nur 20 Pins übrig. Wie vorher schon erwähnt, befinden sich jedoch 26 Knöpfe im Joystick. Da die 5 Wegekнопfe jedoch über einen gemeinsamen Versorgungspin verfügen, können sie sequenziell ein und ausgeschaltet werden und die 5 verschiedenen Leitungen der jeweiligen

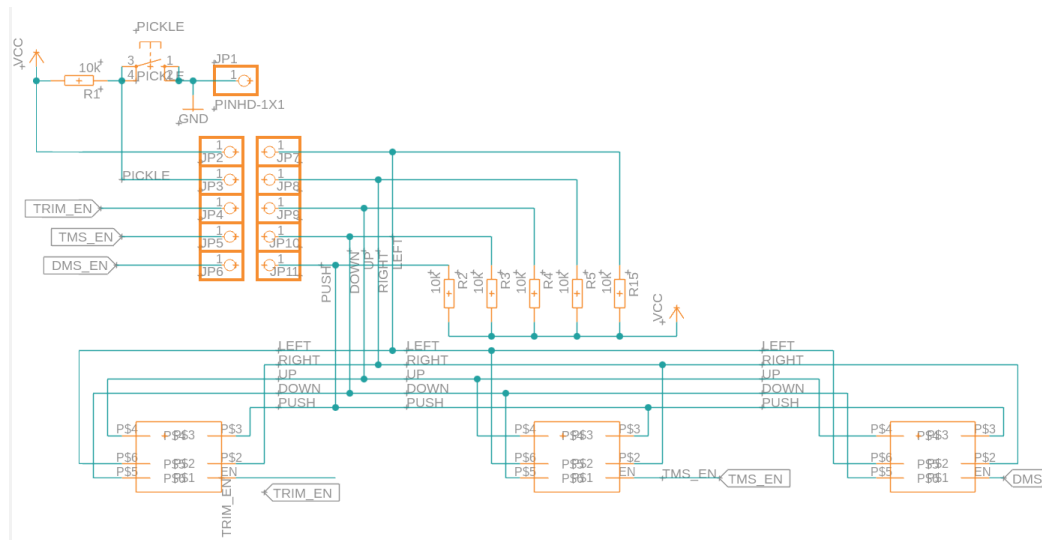


3 Pinout/Belegung des Nanos



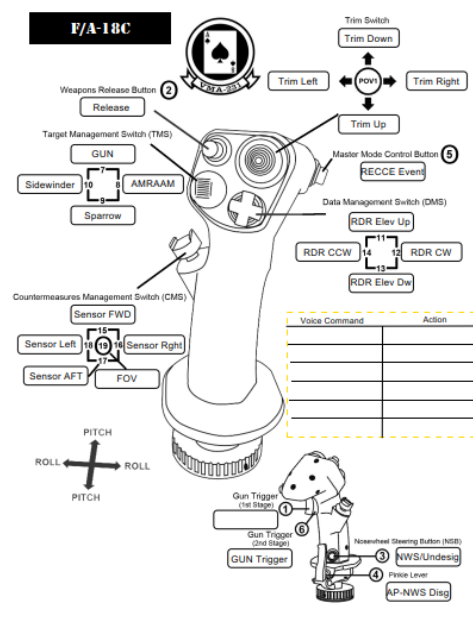
Pins können mit den dementsprechenden zusammengelegt werden. Dadurch können die 15 Pins, die dafür verwendet werden müssten auf 5 Pins zum Auslesen der Zustände und 3 Pins um Ein und Ausschalten der Knopfeinheiten reduziert werden. Bei den anderen Knöpfen wurden diese nicht über dasselbe System gelöst, sondern einzeln auf den  $\mu C$  gezogen, da nun genügend Pins zur Verfügung standen und die Verkabelung nicht viel einfacher geworden wäre.

## Schematic der Frontplatine:



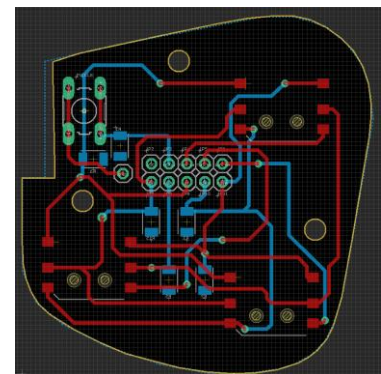
4 Schematic der Frontal verbauten Platine

Hier lässt sich nochmal gut erkennen wie die gleichen die Ausgänge der verschiedenen Knopfeinheiten alle auf die selben fünf Leiter gehen. Dabei ist zu beachten, da die Pins alle auf Pullups konfiguriert sind, dass eben immer nur diese Einheit die Ausgänge kontrollieren kann, welche



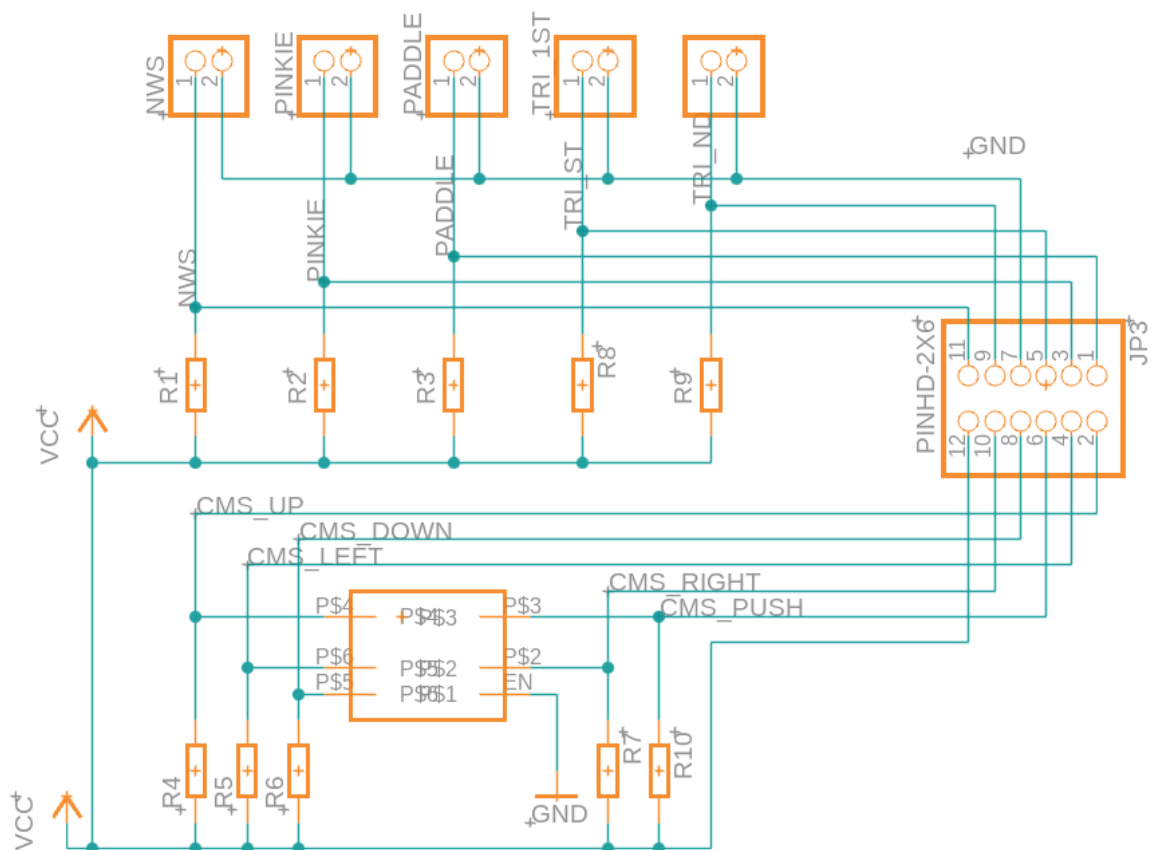
gerade vom  $\mu C$  auf GND geschaltet wird.

Weiters ist hier noch zu erwähnen dass, die Namen TRIM, TMS, DMS sich auf Kontrolleigenschaften des Flugzeuges bezieh, dabei beziehe ich mich auf die Abbildung rechts, und diese nur Orientierungshalber übernommen wurden.



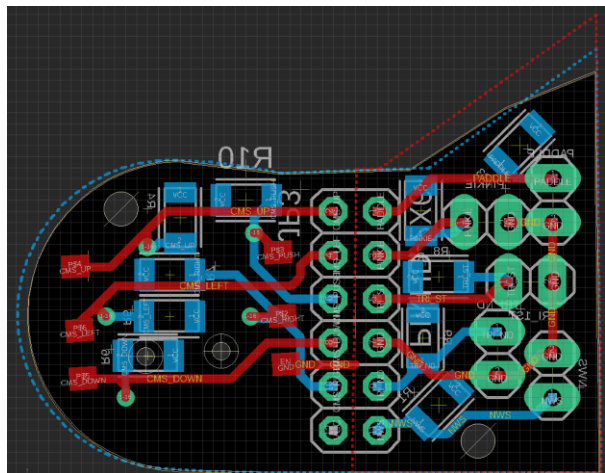
5 Dazugehöriges Board

### Schematic der Daumenplatine:



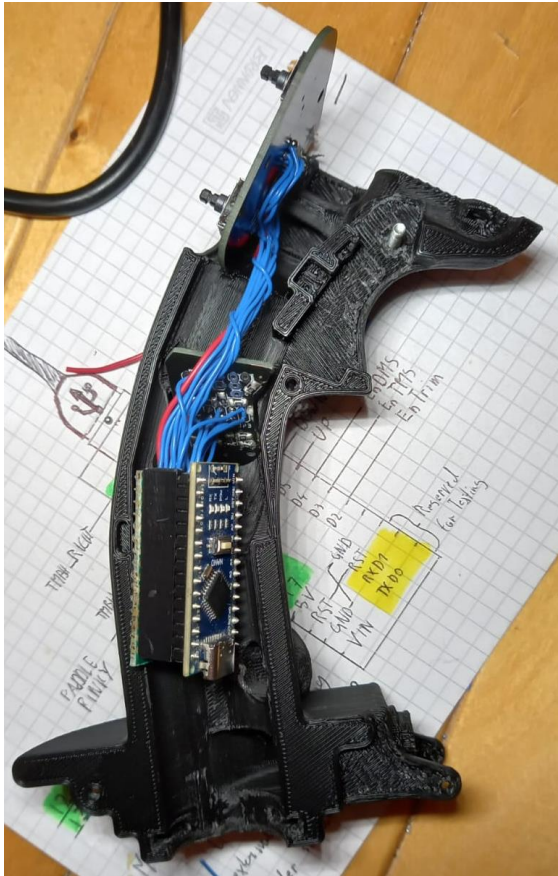
## 7 Schematic der an der Seite Verbauten Platine

Die and der Seite für den Daument 5Way verbaute Platine diente auch als eine verteilerplatine damit nicht jeder andere Knopf welcher sonst noch verbaut war einen eigenen GND kabel braucht und so schlussendlich nur drei GND kable vom Nano ausgangen einer zumr Frontal verbaute Platine, einer zur seitlichen und einer zur Basis.



## 6 Dazugehöriges Board





8 Joystick während des Lötens

## Praktische Verkabelung

In der Praxis sah die Verkabelung nicht mehr so sauber und ordentlich aus, da alles in einem sehr beschränkten raum Platz finden musste. Rechts ist zu erkennen, dass auch trotz der Reduzierung auf 10 Kabel es doch ein wenig unübersichtlich wurde.

## Software

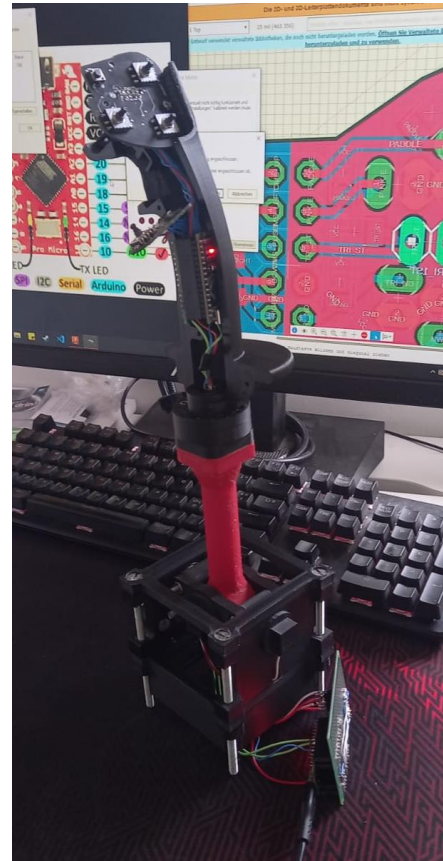
Zeitgleich zur Planung des Aufbaus wurde auch schon die auf den Arduinos laufende Software entwickelt. Wenn man bei der Basis anfängt, befindet sich auf dem Arduino Micro einerseits eine Bibliothek, welche sich darum kümmert, dass der Micro als HID aufscheint. Diese wird noch mit dem nötigen Joystick code erweitert, damit der Micro als Joystick arbeiten kann bzw. erkannt wird.

Der Micro schreibt die über die Potentiometer ausgelesenen Daten in die jeweiligen Achsen X/Y und übernimmt die über I2C eingehenden Daten der Knöpfe und gibt diese an den PC weiter.

Der Nano im Joystick übernimmt hierbei die unterliegende Rolle und schick auf Anfrage des Micros die Zustände der Knöpfe über I2C. Diese Kommunikation findet über die Arduino eigene I2C Bibliothek statt.

## Finalisierung und Testphase

Die eigentliche Testphase dieses Projekts war immer kontinuierlich neben der Entwicklung, da eben schon ein erster Prototyp existierte. Dieser bot eine super Ausgangsbasis, da man während man etwas plante auch vom Digitalen auf das Physische übergehen konnte und sich ein gutes Bild erzeugen konnte was noch geändert und angepasst werden muss. In der Phase der Finalisierung ging leider bei der Lieferung der Platinen über DHL etwas schief wodurch die Platinen erst eine Woche verspätet ankamen, was den Zeitplan etwas einschränkte aber bis zum Schluss nicht weiters erwähnenswert. Nachdem alle Teile vor Ort waren konnte mit dem Zusammenbau begonnen werden und nach ca. zwei bis drei Nachmittagen war der finale Aufbau fertig. Danach wurde der schon im März fertiggeschriebene Code ein wenig angepasst und auf die Controller geladen. Nach einigen kleinen Anpassungen (Feilen, Löten) war der Finale Prototyp fertig und das Protokoll wurde auch finalisiert.



9 Bild aus der finalen Testphase

Quellen:

Programme

[verwendete Joystick-Library](#)

[GitHub mit dem Code zu diesem Projekt](#)

3D-Modelle

[Stick Basis unbearbeitet](#)

[Gimbal unbearbeitet](#)

[GitHub zu den eigenen 3D Modellen](#)

Bilder

[https://shop.thrustmaster.com/it\\_it/hotas-warthog.html](https://shop.thrustmaster.com/it_it/hotas-warthog.html)

[https://www.researchgate.net/figure/An-illustration-of-the-three-angles-yaw-pitch-and-roll-returned-by-the-Hyper-IMU\\_fig1\\_329603549](https://www.researchgate.net/figure/An-illustration-of-the-three-angles-yaw-pitch-and-roll-returned-by-the-Hyper-IMU_fig1_329603549)

<https://www.digitalcombatsimulator.com/upload/iblock/49a/HOTAS.png>

- Eigene Bilder sollten auch im GitHub zu finden sein.