

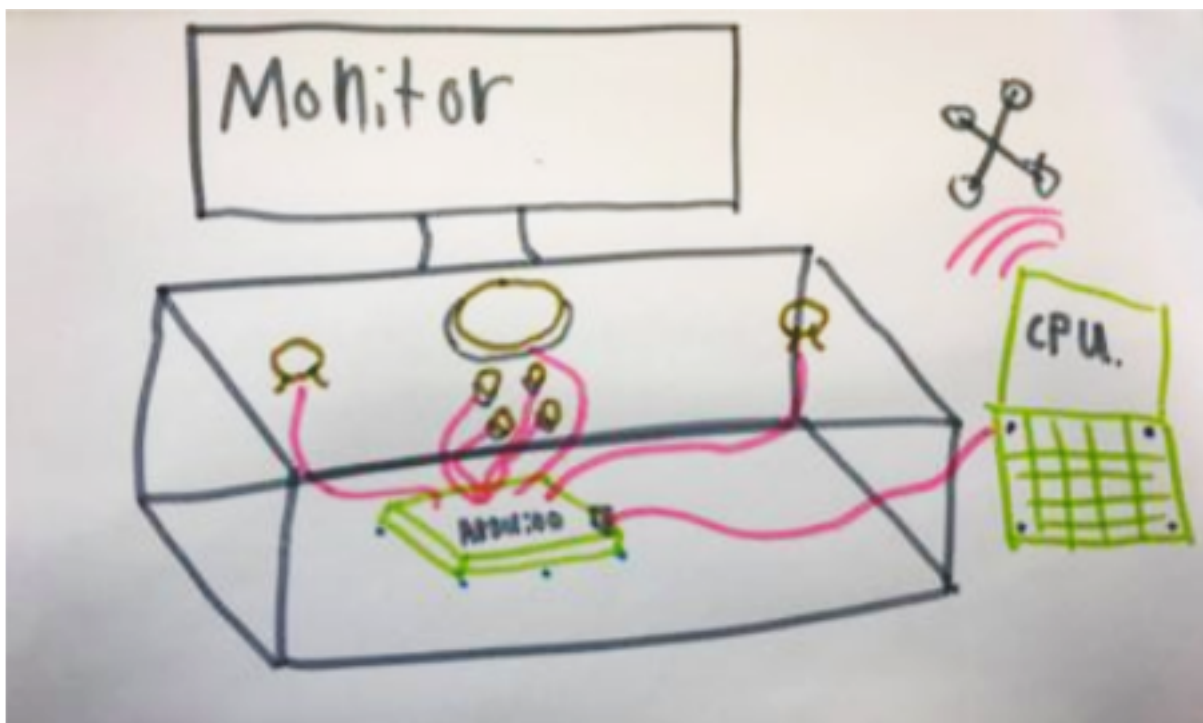
# Tello-drone controller: Iteration 1

## Indledning

- Dette er anden iteration af vores projekt med en controller til tello-droner. Vi har valgt at gå nostalgis vej, og lave en arkade-stil controller. Det betyder at dronen bliver styret med joysticks og knapper, som sidder under en skærm. På skærmen vises live video-feed af hvad dronen ser, sådan at brugeren styrer dronen som var den del af et computerspil. I dette dokument er forklaring af projektet, status på projektet i nuværende stadie og planer for forløbet indtil næste, og sidste, iteration.

## Forklaring af projekt:

- Som tidligere nævnt, skal vores system fungere ligesom en god gammeldags arkade. Det betyder store knapper, joysticks og en skærm, alt sammen pakket ind i et træskelet. I sidste iteration var skitsering over hvordan sådan et system kunne se ud. Nu hvor projektet er mere detaljeret, har vi fremstillet en skitse over hvordan knapper og joysticks sættes op i controlleren. Derudover har vi tildelt funktioner til 3 ud af 5 (7 hvis tryk på joysticks tæller med), af knapperne.
- **Følgende funktioner er tildelt:**
  - Stor knap (0) - Abort!
  - Lille knap (1) - Let
  - Lille knap (2) - Land
  - Joystick (0) - Rotation og højde
  - Joystick (1) - Horisontal bevægelse
- **Skitse over layout af knapper og joysticks** (I siderne er joysticks, midten er knapper. Den store knap i midten er abort-knappen):



## Status på tjekliste - Faser til vores controller-byggeri:

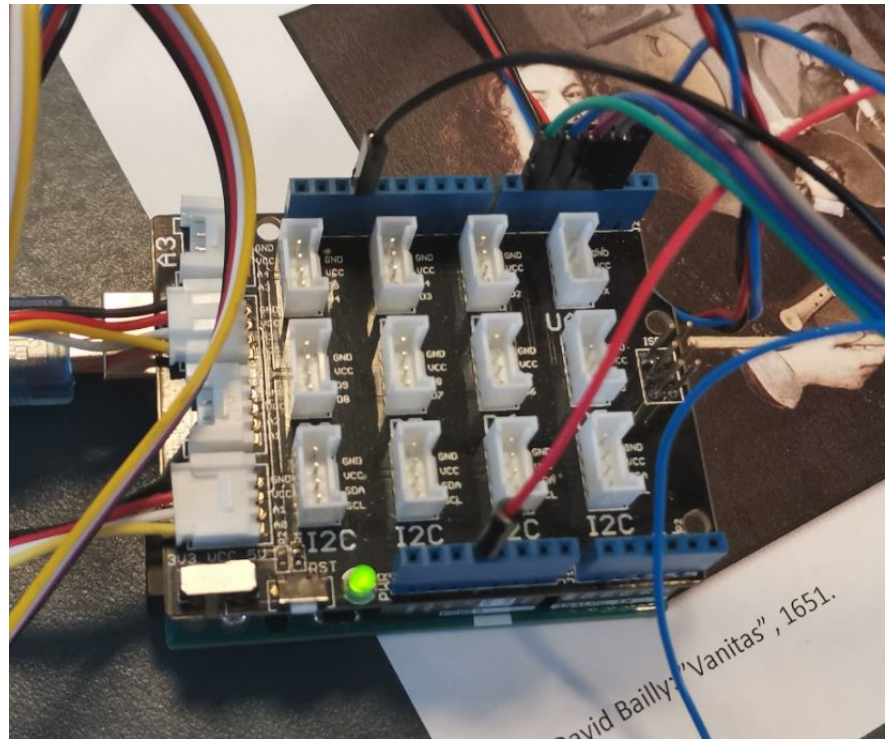
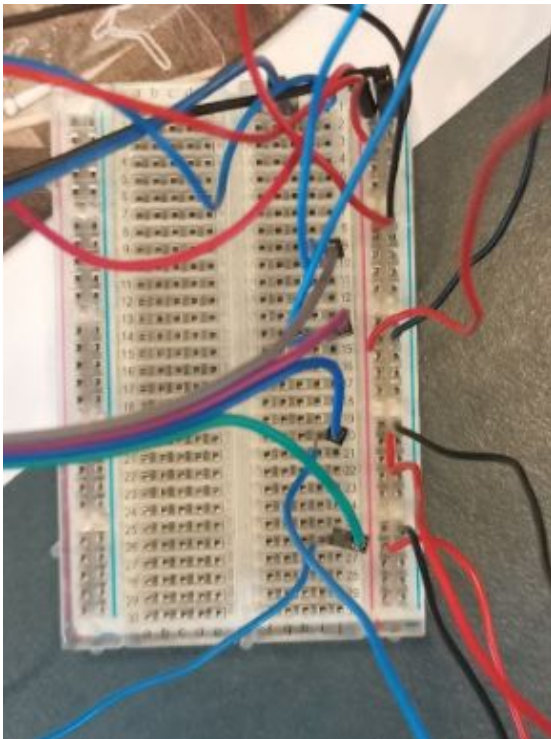
- I iteration 0 var følgende tjekliste over de forskellige trin i udviklingen af controlleren. Der er her en status over hvor langt gennem tjeklisten vi er kommet:
- **(Fed betyder at steppet er fuldført)**
- **(Fed kursiv betyder at de er under arbejde/snart færdige. Status på individuelle ting kan findes længere nede)**
- Normal tekst betyder at det praktiske arbejde ikke er påbegyndt endnu.

1. **Controller bygges på arduino**
2. **Input til PC fra arduino**
3. **Oversættelse fra c++ til python**
4. Output fra pc - kommandoer til drone
5. Drone skal kunne sende video til computer, som skal fremvise det.
6. Æstetik - ***Custom Arcade Joysticks***, Classic Arcade

### 1. Controller bygges på arduino

- Det første step i konstruktionen af controlleren var at opsætte knapper og joysticks på den rette måde på selve arduinoen. Der bruges en "sandwich" som sættes ovenpå arduinoen. Denne bruges til at kunne få analog-input fra joysticks. Derudover benyttes en separat del som sætter ledningerne sammen i baner. På den måde kan alle knapper få 5 volt til strøm og jordforbindelse.

#### Billeder af opsætning:



## 2. Input til PC fra arduino

- Den næste del i processen var at kunne læse på computeren når knapperne blev trykket på, og kunne læse værdier fra de to joysticks. Dette er koden som arduinoen kører, som så kan læses af computeren.

**Her er den færdige kode, som printer alle inputs som serial-data i en lang string til sidst. Alle data er separeret med et “,” , for at det kan indeles i python:**

```
#include <Arduino.h>

void setup() {
  Serial.begin(9600);

  //Pins (knapper) defineres (Port, Type af Input)
  pinMode(2, INPUT_PULLUP);
  pinMode(3, INPUT_PULLUP);
  pinMode(4, INPUT_PULLUP);
  pinMode(5, INPUT_PULLUP);
  pinMode(6, INPUT_PULLUP);
}

void loop() {
  //De to joysticks akser defineres som floats
  float Joy0x = float(analogRead(A0));
  float Joy0y = float(analogRead(A1));
  float Joy1x = float(analogRead(A2));
  float Joy1y = float(analogRead(A3));

  //Data printes i serial

  Serial.println(String(digitalRead(2))+","+String(digitalRead(3))+","+String(digitalRead(4))+","+String(digitalRead(5))+","+String(digitalRead(6))+","+String(Joy0x)+","+String(Joy0y)+","+String(Joy1x)+","+String(Joy1y)+",");
}
```

### 3. Oversættelse fra c++ til python - Protokol dokumentation

Data fra arduino sendes via serial til vores pythonscript. Pythonscriptet køres på værtscomputeren som benyttes til systemet.

**Her er dokumentation for protokollen vi bruger når data bliver sendt gennem serial. (Som tidligere nævnt er det hele i en lang string, så et sorteringssystem skal laves til de forskellige variabler):**

Nr.# liste	Variabel	Data	Type	Værdi-variation
0	Knap0 (Stor Rød Knap)	Boolean	Digital	0/1 (Sand/Falsk)
1	Knap1	Boolean	Digital	0/1 (Sand/Falsk)
2	Knap2	Boolean	Digital	0/1 (Sand/Falsk)
3	Knap3	Boolean	Digital	0/1 (Sand/Falsk)
4	Knap4	Boolean	Digital	0/1 (Sand/Falsk)
5	Joy_0_x	Float	Analog	160-650 (700 ved tryk)
6	Joy_0_y	Float	Analog	160-650
7	Joy_1_x	Float	Analog	160-650 (700 ved tryk)
8	Joy_1_y	Float	Analog	160-650

**Dataet som bliver sendt kunne for eksempel se ud på følgende måde:**

- "1, 0, 1, 1, 255.00, 255.00, 700.00, 300.00"
- Næste step er at separere dataet for hvert komma der er, og oprette variabler for de forskellige værdier. Variablerne ville i dette tilfælde se ud på følgende måde:
  - btn0 = True
  - btn1 = False
  - btn2 = True
  - btn3 = True
  - joy0x = 255.00
  - joy0y = 255.00
  - joy1x = 700.00
  - joy1y = 300.00

**Her er koden som oversætter serial data til python. Den udprinter den string som er modtaget i terminalen. (Vi må være ærlige og indrømmer at koden mere eller mindre er stjålet med arme og ben):**

```
#pip3 install pyserial
import serial
import sys

serialPortName = "COM6" #Skal sandsynligvis ændres.
baudRate = 9600 #Defineret i main.cpp
try:
    s = serial.Serial(serialPortName,baudRate,timeout=1)
except:
    print("Kunne ikke finde data gennem", serialPortName, "Husk at tjekke hvilket
port arduino kører igennem")
    sys.exit(1)

print("Tilslutning lavet - Venter på serial data")

try:
    while(s.is_open):

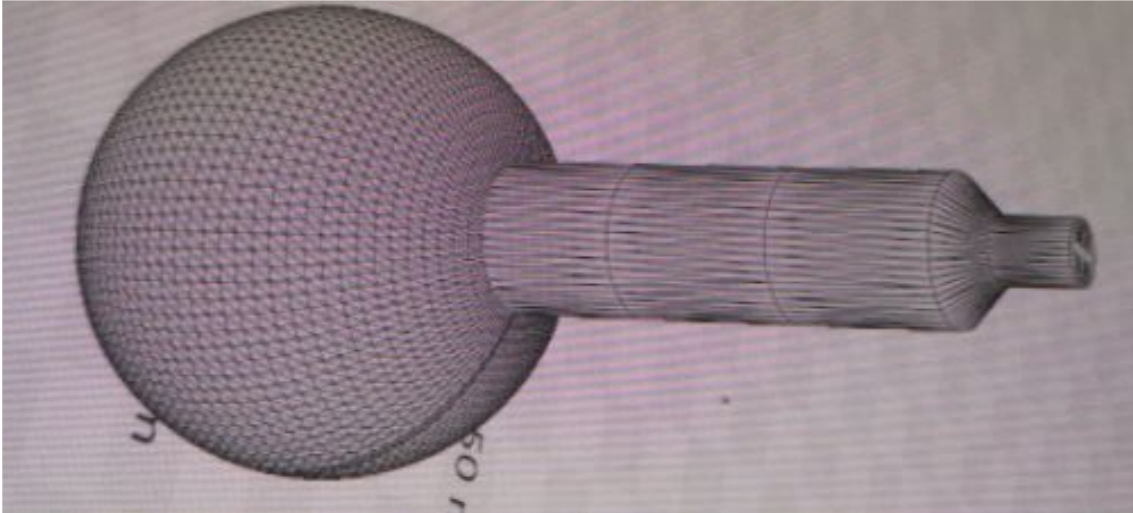
        if(s.in_waiting>0):
            rxLine=s.readline().decode("ascii").strip()
            print("Recieved:",rxLine)

            #Her skal laves system med udpakning af data (Måske lavet nyt script, så
tingene er opdelt lidt mere?)
            # rxsplit = rxline.split()
            # ax = float(rxsplit[0]) etc...
except:
    print("Fejl efter tilslutning til serial (Fejlopsætning af arduino!)"
```

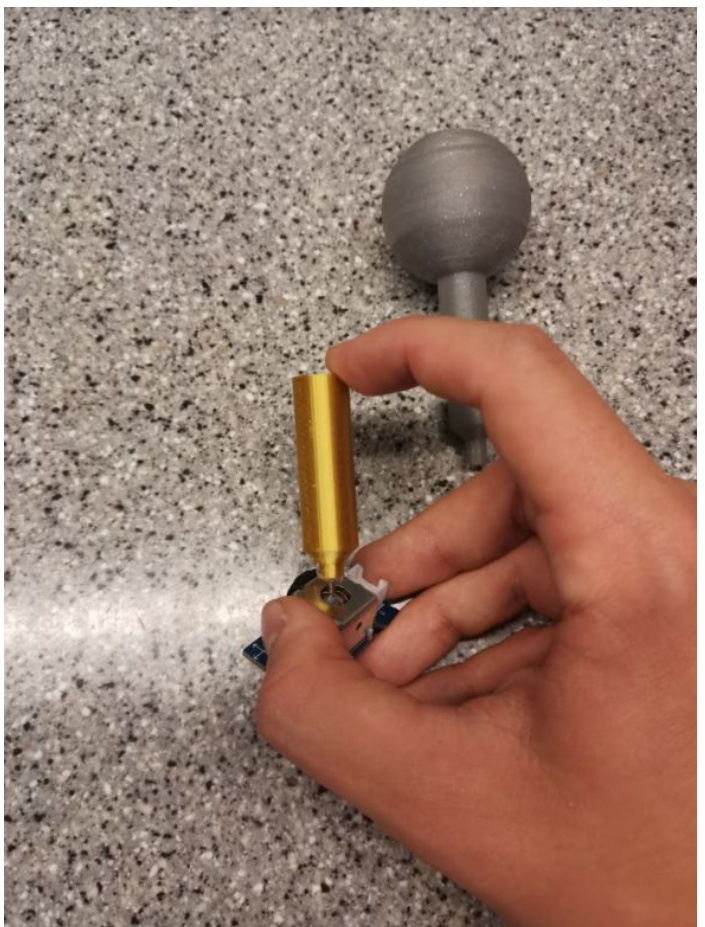


## 6. Æstetik - Custom arcade joysticks

- Vores arbejde på joysticks er begyndt, da de er en af de mere udfordrende dele af projektet, og fordi de kræver 3d-print. Vi vil meget gerne lave den typiske form med en stang, som har en kugle ovenpå, da det er en af de mest klassiske typer joystick som bliver brugt i arkader. Vi er kommet op med følgende design:

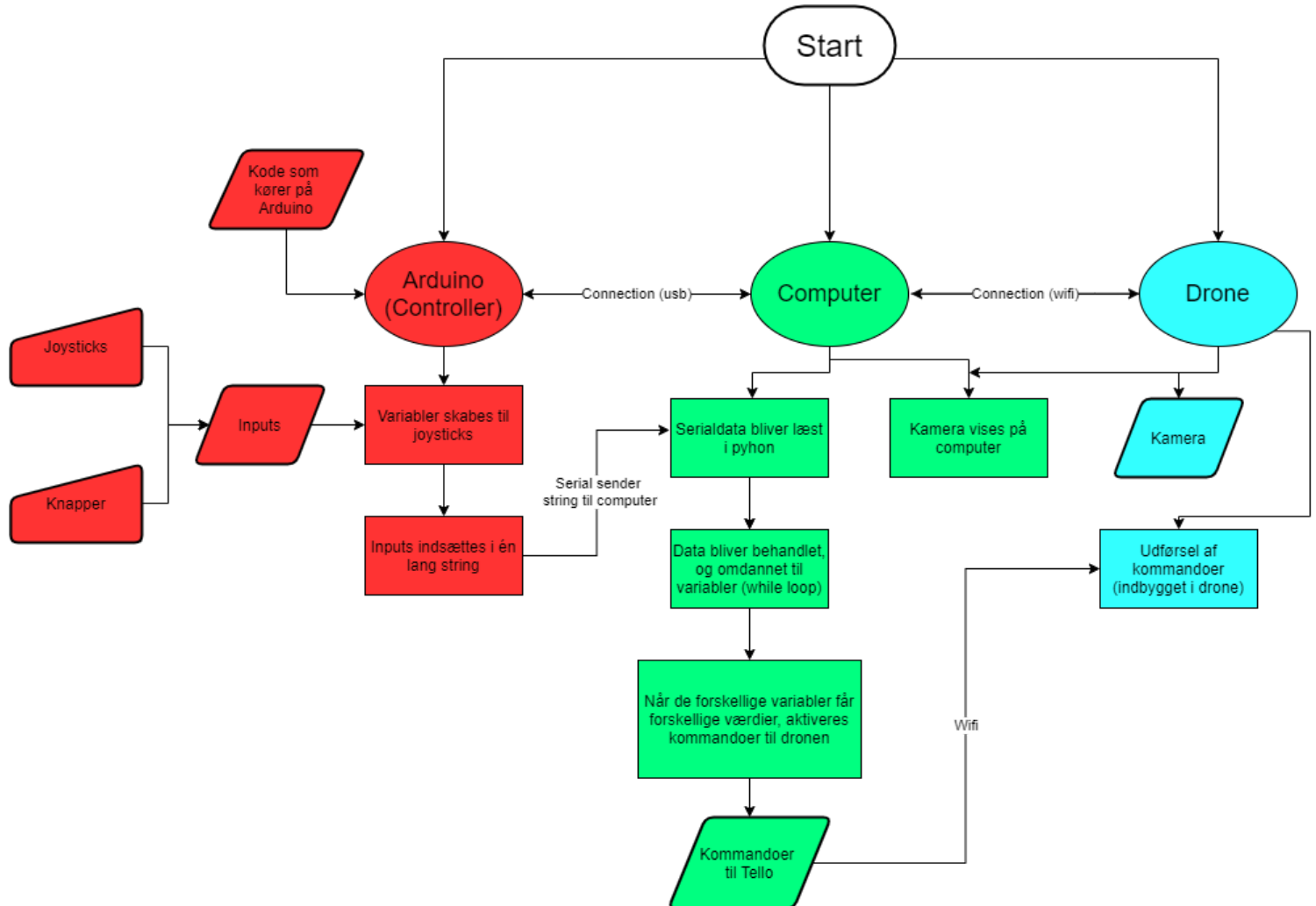


- Designet har vi printet to gange, i guld og sølv. Dog gik den i guld i stykker, da vi kom til at tabe den på gulvet. Den i sølv er sat på selve joysticket, og vi havde succes med monteringen af den, dog vejer den for meget med kugle på. Løsning til dette skal findes senere i processen. Her kan de to joysticks ses:



## Opdateret flowchart over samlet system:

- Flowchartet herunder er en opdatering på flowchartet som kunne findes i iteration 0. Nogle af blokkene er blevet opdateret til at stemme overens med hvordan systemet kører i virkeligheden.



## Næste steps i processen:

- Det næste som skal laves i processen er, at computeren skal kunne sende kommandoer til dronen. Dette er den mest kritiske del vi har tilbage af systemet som mangler, da det giver os muligheden for at have et produkt som virker i praksis. Derudover skal vi også lave en brugerundersøgelse af vores system, fremvise live video fra dronen, og lave selve arkade-maskinen. Vi vil gerne have færdiggjort en model for vores controllers form. selve grundideen for vores controller var at dronen skulle være styret af, hvad der ligner en gammeldags arkade konsol. Det er derfor vigtigt at vores controller ikke ender med at ligne noget andet. Ellers har vi ikke fuldført vores mission. Planen er lige nu at laserskære delene til selve konsollen, men denne plan kan ændre sig. Dog er det den mest simple, hurtigste og mest præcise måde vi kan gøre det på. Om der forekommer problematikker ved dette valg finder vi ud af på et senere tidspunkt.