Tello-drone controller: Iteration 1.5

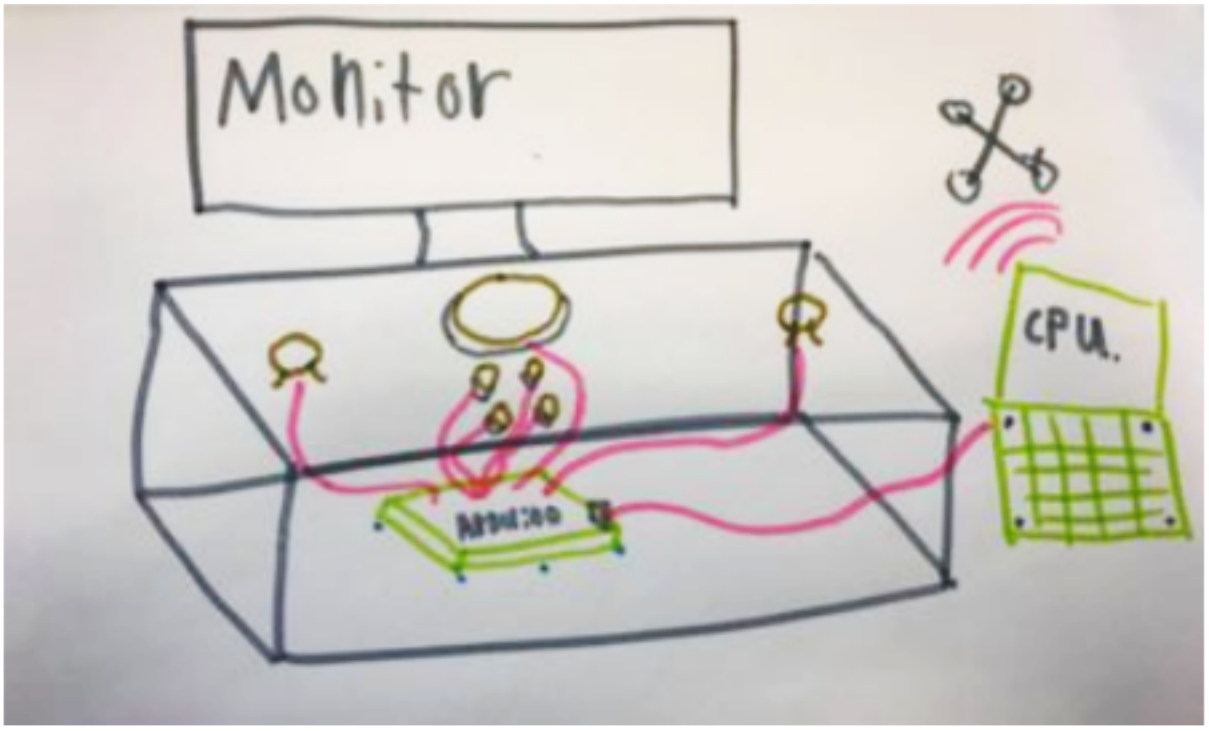
Indholdsfortegnelse

Indledning

- Dette er iteration 1.5 af vores projekt med en controller til tello-droner. Vi har valgt at gå nostalgiens 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:

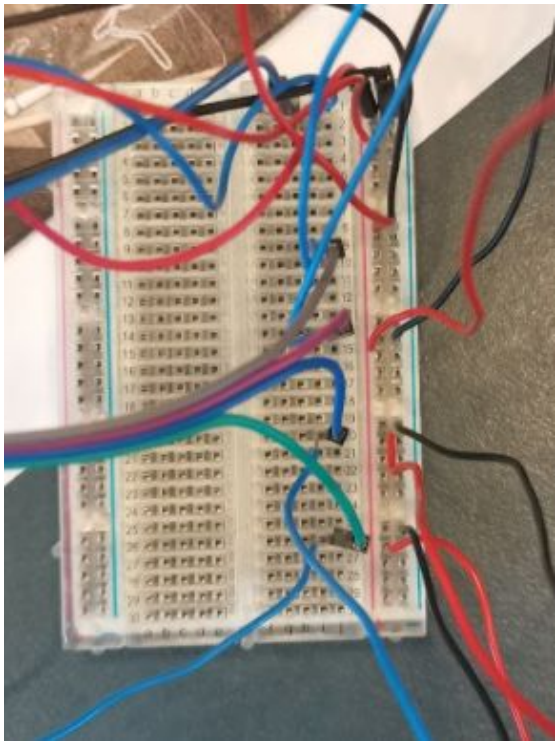
- I iteration 0 var følgende tjekliste over de forskellige trin i udviklingen af controlleren. Der er her en status over hvor langt gennem tjeklisten vi er kommet:
- **(Fed betyder at steppet er fuldført)**
- ***(Fed kursiv betyder at de er under arbejde/snart færdige. Status på individuelle ting kan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

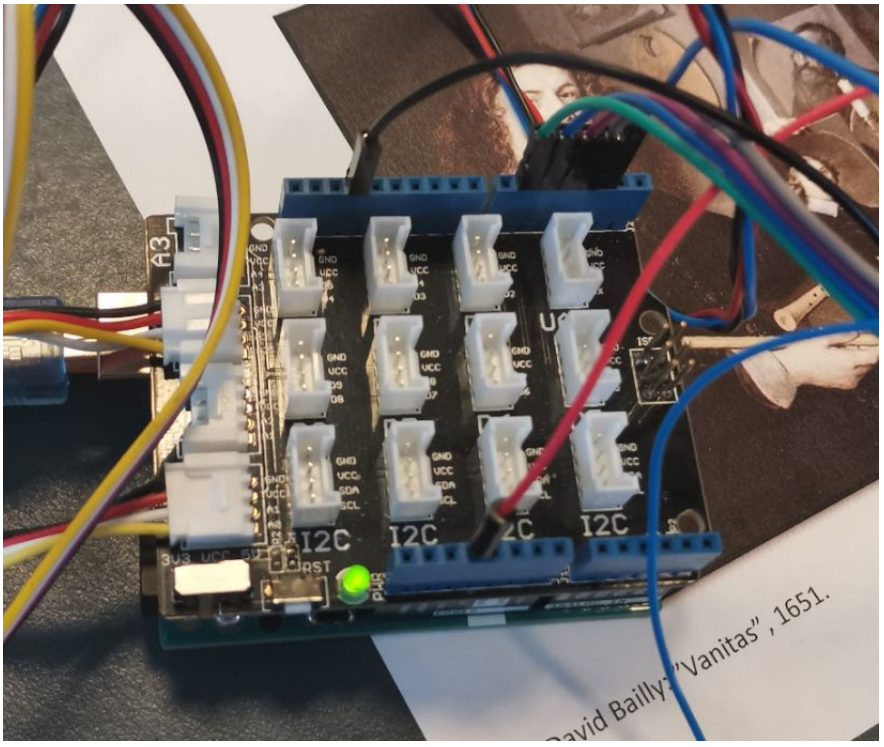
1. **Controller bygges på arduino**
2. **Input til PC fra arduino**
3. **Oversættelse fra c++ til python**
4. Output fra pc - kommandoer til drone
5. **Drone skal kunne sende video til computer, som skal fremvise det.**
6. ***Æstetik - Custom Arcade Joysticks, Classic Arcade***

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 inddeles 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. Dog har vi skåret ned i koden og tilpasset den til vores brugssituation):*

```
#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!)")
```

4. Output fra pc - kommandoer til drone

- Vi er begyndt at skrive koden som oversætter inputtet fra controlleren til de kommandoer som dronen skal modtage. Dog har vi ikke haft mulighed for at afprøve koden, da vi har haft controlleren skilt ad under konstruktionen af de æstetiske elementer. Derfor ved vi heller ikke meget om kodens egentlige funktionalitet på nuværende tidspunkt. Dog ved vi at alle kommandoer til dronen som er skrevet i koden fungerer på den måde vi forestiller os. Der mangler alt med joysticks.
  Her er koden på nuværende tidspunkt (Nogle kommentarer slettet):

```
import tellopy
drone = tellopy.Tello(port=9000) #Drone defineres
drone.connect() #Connection mellem program og drone etableres
droneFlying = False
btn0Value = 0
def main(btn0, btn1, btn2, btn3, btn4, joy0x, joy0y, joy1x, joy1y):
    try:
        if droneFlying:
                btn0Value += 1
                if btn0Value > 20:
                    drone.land()
                    btn0Value = 0

            elif not btn1 and not btn2 and not btn3 and not btn4: #Trykket på alle fire
                pass #Do smthn?
            elif not btn1 and not btn2 and btn3 and btn4: #Trykket på de to nederste
                drone.flip_back()
            elif not btn1 and btn2 and not btn3 and btn4: #Trykket på to knapper til
                drone.flip_left()
            elif btn1 and btn2 and not btn3 and not btn4: #Trykket på to øverste knapper
                drone.flip_forward
            elif btn1 and not btn2 and btn3 and not btn4: #Trykket på to knapper til højre
                drone.flip_right()
            elif not btn1 and btn2 and btn3 and btn4: #Trykket på knap 1
                drone.flip_backleft()
            elif btn1 and not btn2 and btn3 and btn4: #Trykket på knap 2
                drone.flip_forwardleft()
            elif btn1 and btn2 and not btn3 and btn4: #Trykket på knap 3
                drone.flip_forwardright()
            elif btn1 and btn2 and btn3 and not btn4: #Trykket på knap 4
                drone.flip_backright()
        else:
            if not btn0:
                drone.takeoff()
                drone.sleep(3)
                droneFlying = True
            else: pass
    except: print(Exception)
```

5. Drone skal kunne sende video til computer, som skal fremvise det.

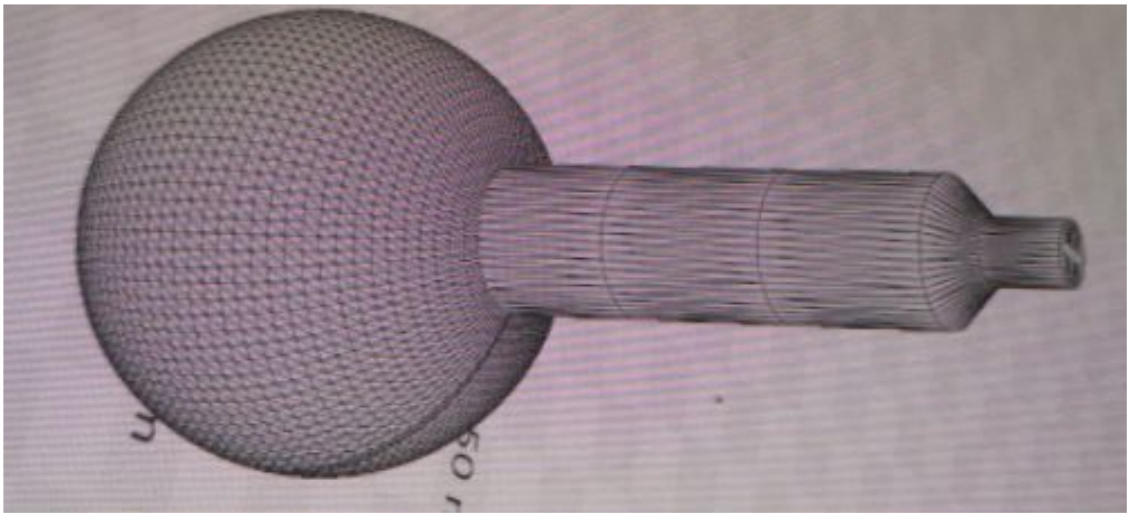
- Vi fik koden til video inputtet til at virke, ved at låne koden fra internettet, koden stammer nemlig fra Tellopy som er det library/python package vi bruger til tello-dronens kommandoer.
  Her er koden på nuværende tidspunkt (Forkortet og med færre kommentarer):

```
import sys, traceback, tellopy, av, cv2.cv2 as cv2, numpy, time
def main():
    drone = tellopy.Tello()
    try:
        drone.connect()
        drone.wait_for_connection(60.0)
        retry = 3
        container = None
        while container is None and 0 < retry:
            retry -= 1
            try:
                container = av.open(drone.get_video_stream())
            except av.AVError as ave:
                print(ave)
                print('retry...')
        frame_skip = 300  # skip first 300 frames
        while True:
            for frame in container.decode(video=0):
                if 0 < frame_skip:
                    frame_skip = frame_skip - 1
                    continue
                start_time = time.time()
                image = cv2.cvtColor(numpy.array(frame.to_image()), cv2.COLOR_RGB2BGR)
                cv2.imshow('Original', image)
                cv2.imshow('Canny', cv2.Canny(image, 100, 200))
                cv2.waitKey(1)
                if frame.time_base < 1.0/60:
                    time_base = 1.0/60
                else:
                    time_base = frame.time_base
                frame_skip = int((time.time() - start_time)/time_base)
    except Exception as ex:
        exc_type, exc_value, exc_traceback = sys.exc_info()
        traceback.print_exception(exc_type, exc_value, exc_traceback)
        print(ex)
    finally:
        drone.quit()
        cv2.destroyAllWindows()
```

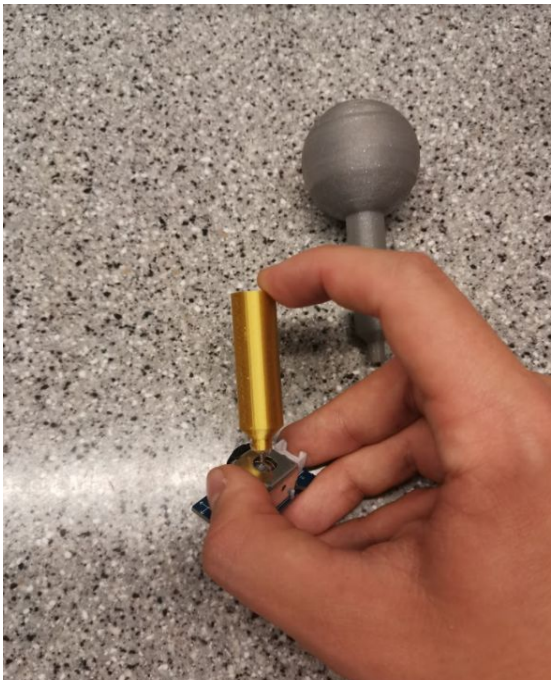
6. Æstetik - Custom arcade

Joysticks

Vores arbejde på joysticks begyndte før færdiggørelsen af iteration 1, 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:

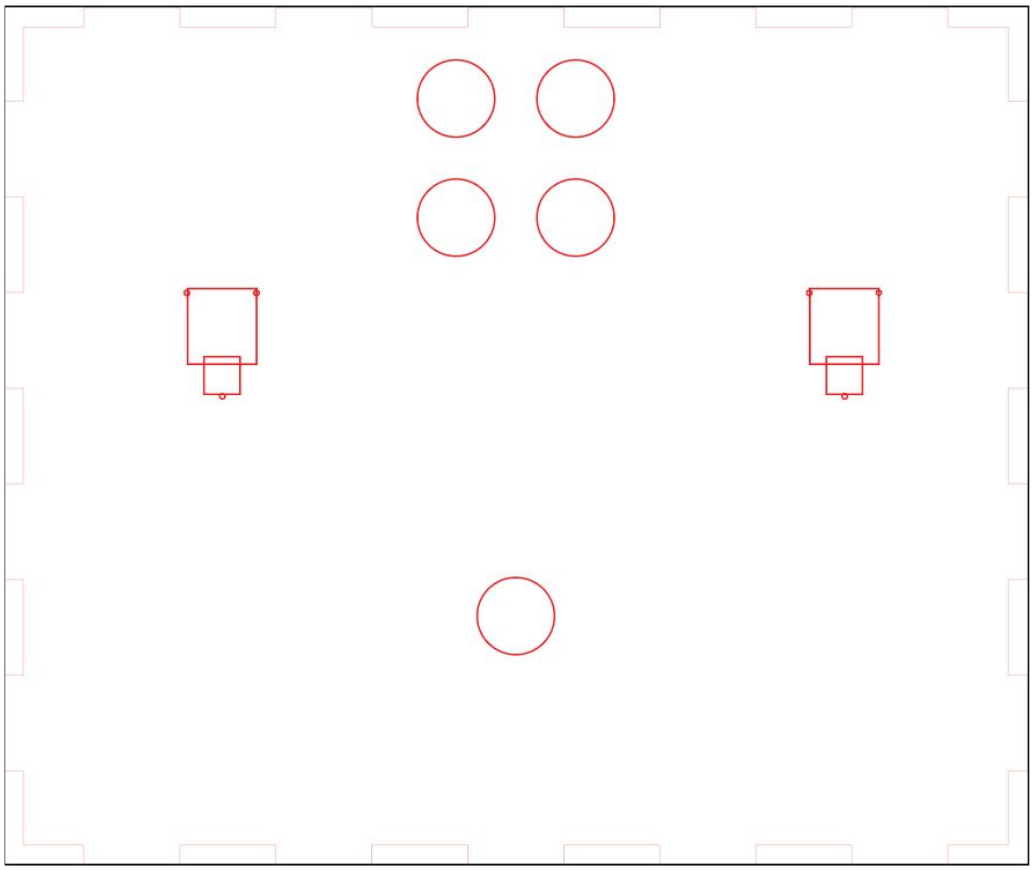


Designet har vi printet to gange, i guld og sølv. Dog gik den i guld i stykker, da vi kom til at tabe den på gulvet. Den i guld er sat på selve joysticket, og vi havde succes med monteringen af dem, dog vejer de for meget med kugle på. Løsning til dette skal findes senere i processen. Her kan de to joysticks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

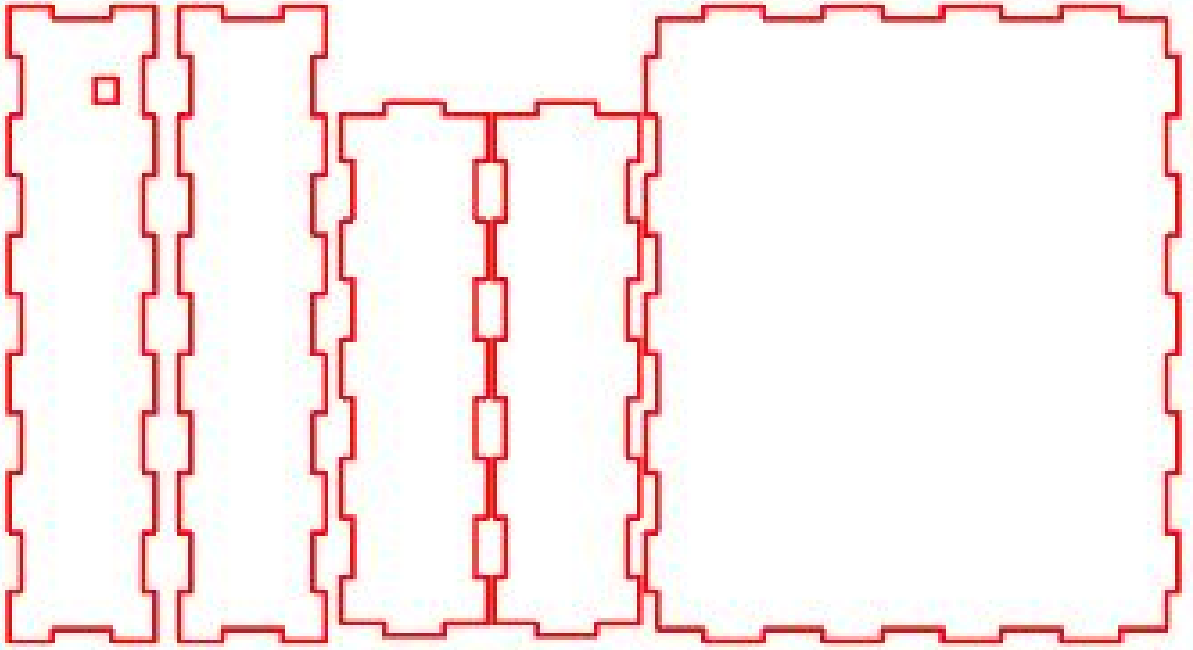


Controllerkassen

- Vi valgte at bruge akrylplade til skelettet af vores controller. Det gode ved at bruge akryl var at man kunne se ind til elektronikken og derved se eventuelle fejl hvis det nu skulle forekomme Vi syntes også at den klare gennemsigtige glas struktur så godt ud.



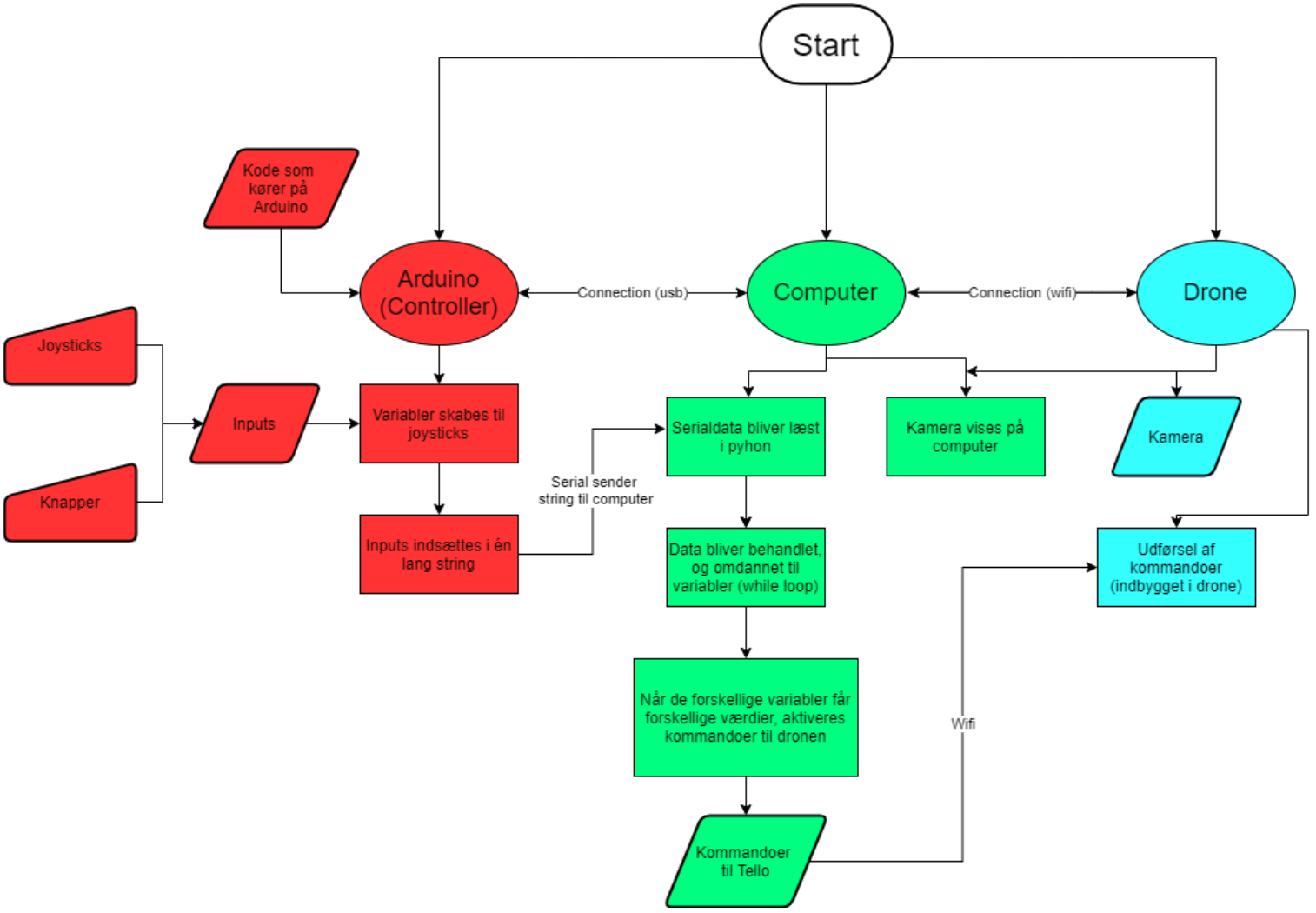
- Efter vi havde skåret toppen på vores controllere ud, fandt vi ud af at de huller hvor knapperne og joystickene skulle være, derfor tog det flere udskæringer for at finde ud af hvor vi præcist ville have vores knapper og joysticks. Det vidste vi på forhånd og valgte derfor at laver de forskellige udskæringer i pap i stedet for af spilde dyre akrylplader.



- Måden vi lavede pladerne på, var ved at bruge en laser cutter til at beskære en stor akrylplade med en 1/10 af en millimeters præcision. Vi skar den store plade ud i seks mindre stykker. 4 sider og en top og bund. alle siderne var skåret således at man kunne sæte dem samme med akryllim. D.s.v havde vi ikke noget akryllim, så vi prøvede at svejse det i stedet således at akrylen visse smelte sammen. Det gik ikke så godt, da hele kasse blev skæv, og knækkede da vi forsøgte at montere låget. så vi skar en ny, som vi formentlig ender med at lime sammen.

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.



Brugerundersøgelse:

Opbygning af brugerundersøgelse

Vi har også lavet en brugerundersøgelse som vi vil gennemføre når vi har en færdig prototype af vores controller. Det betyder at den i skal virke med mindre små fejl. Vi er endnu ikke nået punktet hvor vi har en færdig prototype og har derfor ikke kunnet udføre undersøgelsen. Når vi undersøger vores prototype tester vi ud fra følgende 6 heuristikker:

Brugerkontrol og frihed

- *Brugeren skal være i stand til at afbryde systemet i ethvert øjeblik. Dette bliver ekstra vigtigt i et system som dette, hvor både personskader og materielle skader hurtigt kan forekomme hvis man ikke kan afbryde systemet.*

Kontinuitet og symmetri

- *Man skal kunne se hvilke dele af controlleren som "hører sammen", altså skal knapper og joysticks være inddelt efter deres funktioner.*

Fejlprævention

- *Programmet skal have svært ved at crashe. Det skal være sådan at der ikke forekommer crashes på grund af en brugers input. I stedet skal programmet kunne udvælge*

Genkendelse i stedet for hukommelse

- *Man skal kunne kigge på knapperne på controlleren, og med det samme være i stand til at genkende deres funktionalitet, i stedet for at være nødt til at huske det.*

Erfaring

- *Brugere som har prøvet lignende systemer skulle gerne* skulle gerne ret instinktivt kunne vide hvordan systemet skal styres.

Sammenligning mellem systemet og virkeligheden

- *Systemet skulle kunne kommunikere med brugeren gennem det samme sprog, ord, sætninger og koncepter som brugeren er meget velkendt med*

Med en undersøgelse som har disse 6 temaer i fokus vil vi bygge videre og finjustere den prototype som skal bruges til den færdige controller. ***Opbygningen af selve undersøgelsen findes i afleveringen af brugerundersøgelsen.***

Resultater af brugerundersøgelse

- Her indsættes uddrag af resultaterne fra brugerundersøgelse

Næste steps i processen:

- Det næste som skal laves i processen er, at computeren skal kunne sende de rette kommandoer til dronen ud fra inputtet til controlleren. 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 iterere vores joysticks, da de på nuværende tidspunkt er for store, og har et meget svagt punkt omkring basen som knækker meget nemt.