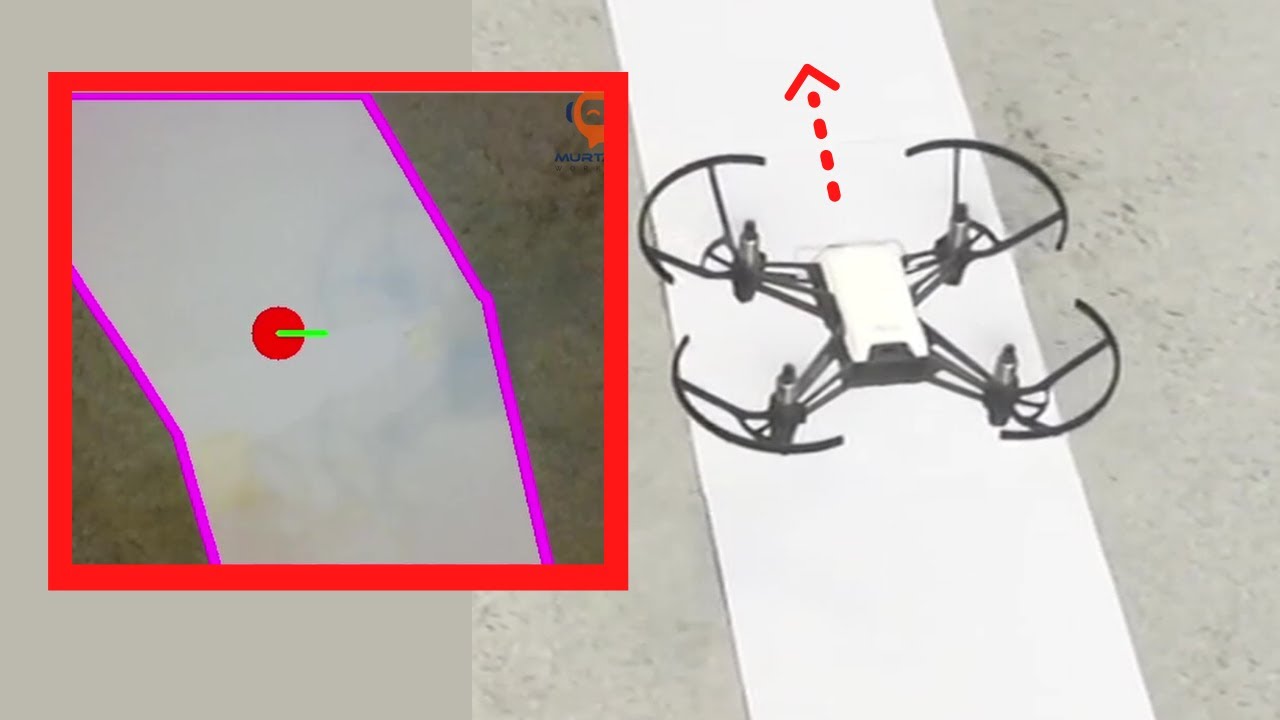
**ARIES PROJECT**

**TOPIC-Line Following Drone**



**Name-Aniket Raj**

**Electrical Engineering**

Basic Drone Training

Components in basic drone:

`1. Hexa-X frame:structure in which all parts are fitted such that centre of gravity is uniformly distributed and

to apply



2.Motors: these are essential for uplifting the drone by providing thrust in upward direction and the motors are aligned in opposite direction alternately to eliminate the net torque.We have used 1400kV motors to run the hexa-x drone.



3.ESC(electronic speed controller):these are attached with every motor in drone and it varies the speed of motors.



4.Radio Transmitters:this is used to give instructions to drone from user.We have used flysky transmitter.



5.Reciever:this is binded to the transmitter and carry the information to implement to the drone.we used **FLY SKY FS IA6B RF 2.4GHz 6CH PPM.**

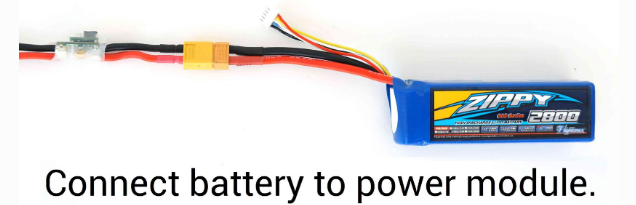
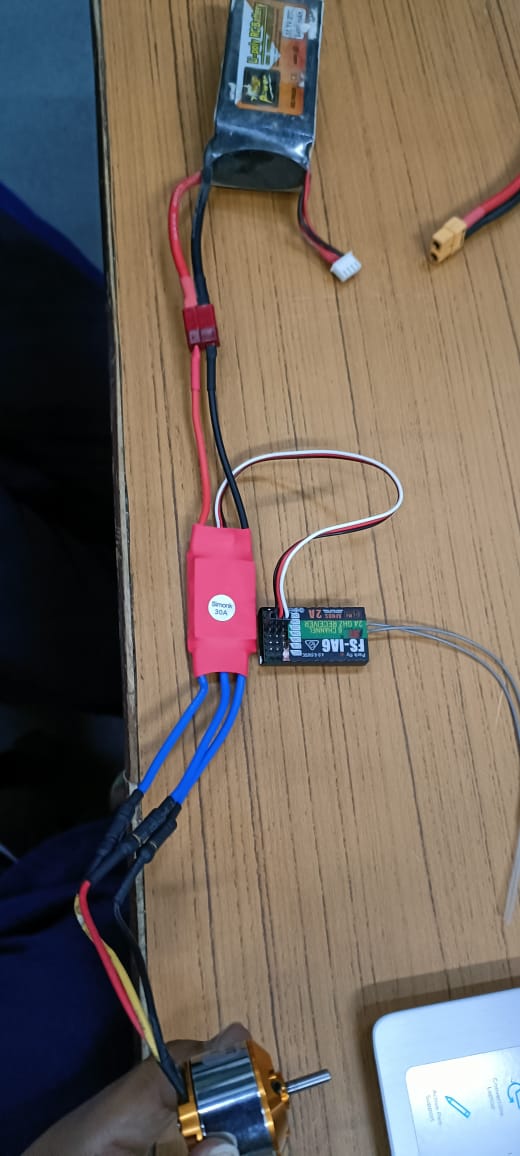
****

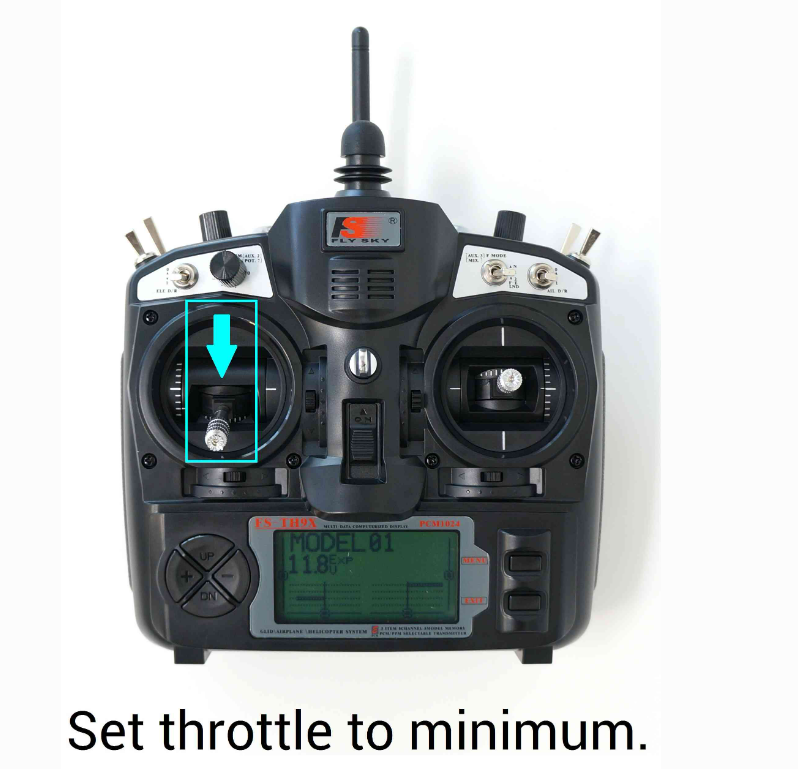
6.Pixhawk:this is the main control part which control all other components of a drone.

Q.What we have done to fully functional drone and what difficulty we faced.

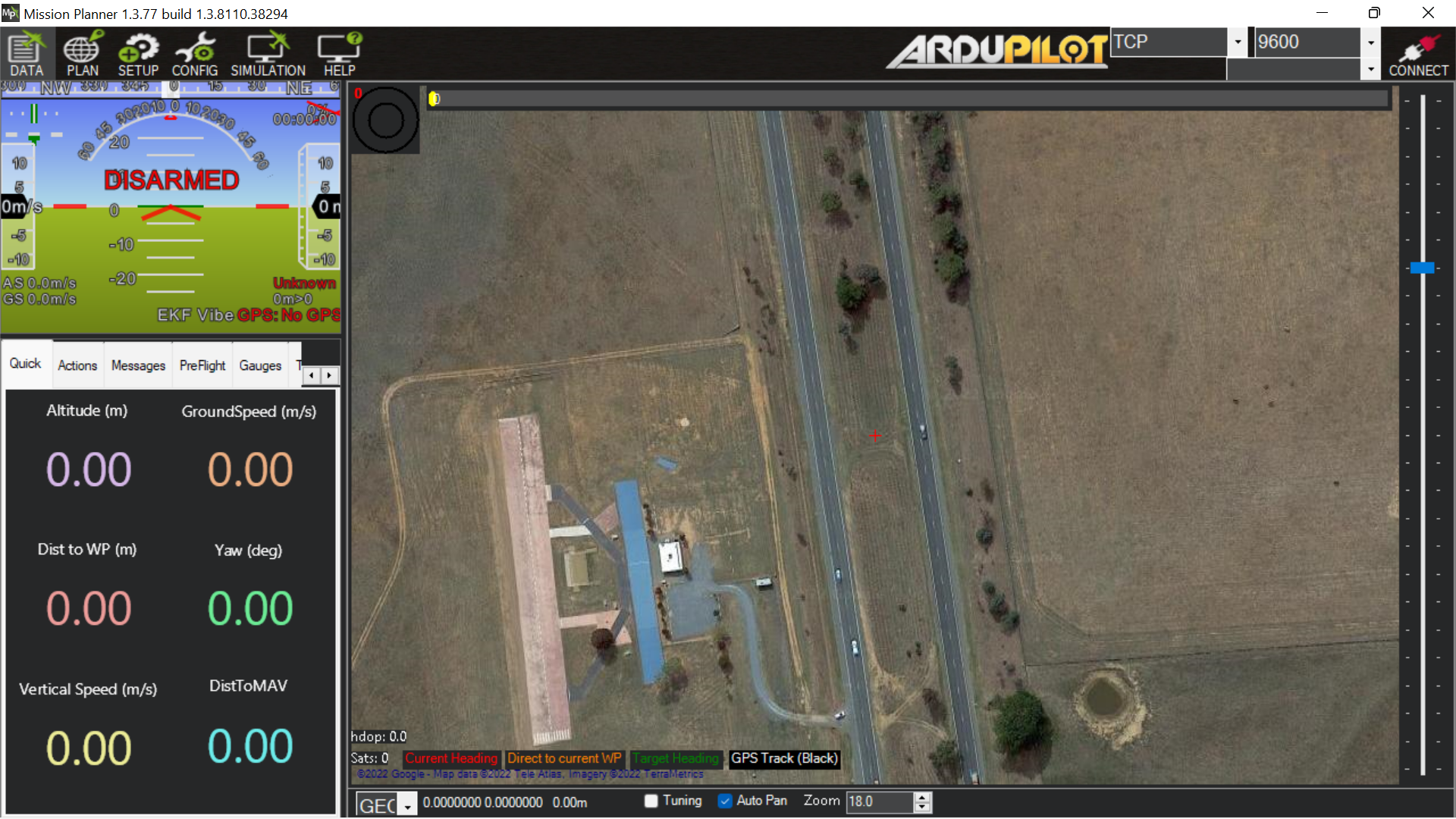
Step1:First in Rx binding the receiver to transmitter we have short circuited the channel 1 and make connections of esc in channel 3 of receiver and now we turn on the transmitter by pressing the Rx binding button and remove the short circuit and the transmitter got binded.

Problems:not shorting after binding,no proper connections,not connecting battery to the Esc’s.



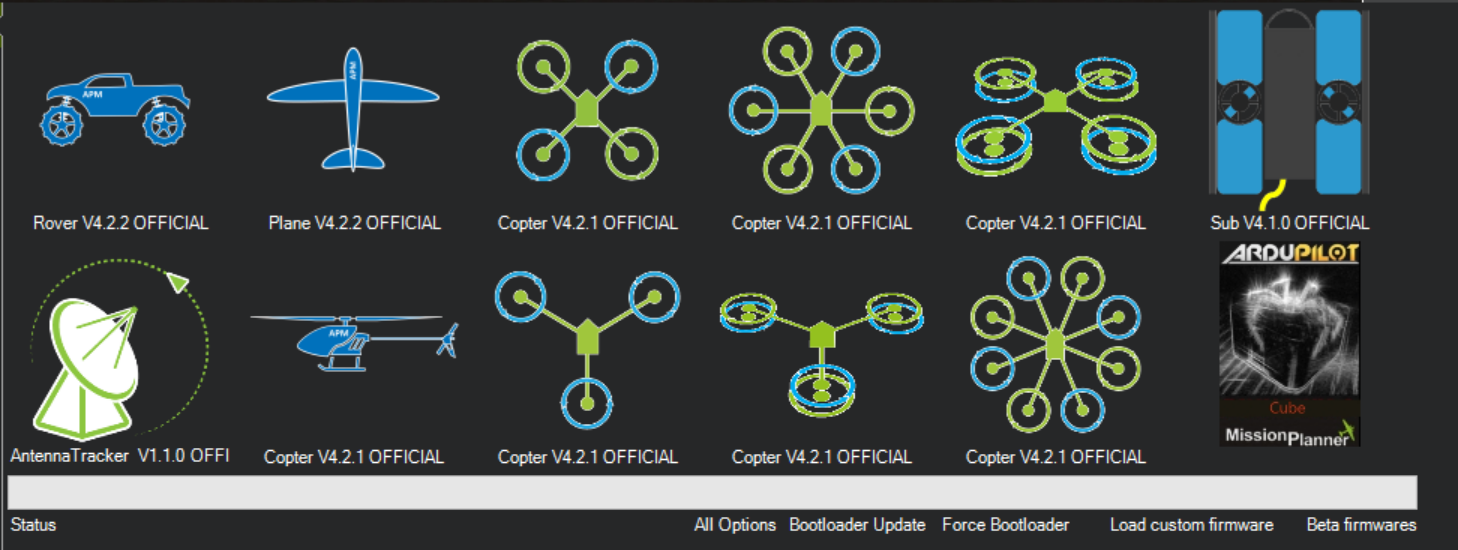


Step2:we worked with mission planner.

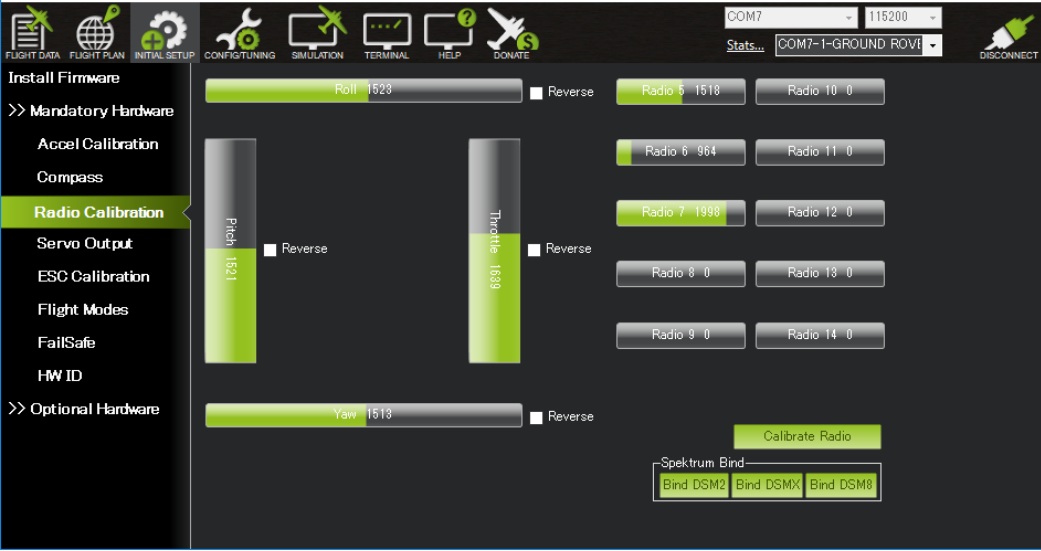


Firstly we setup the drone and connecting it through usb or telemetry.

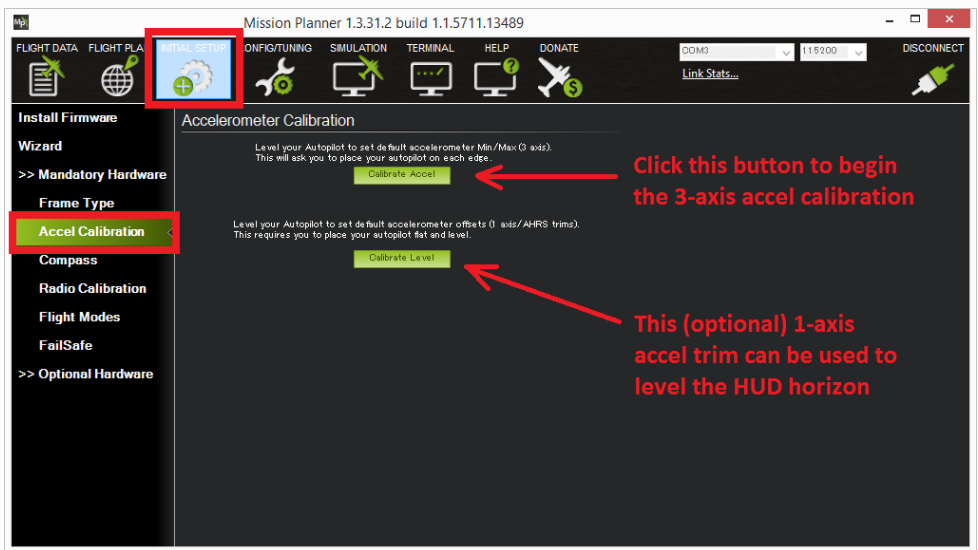
1.installing firmware-hexa-x

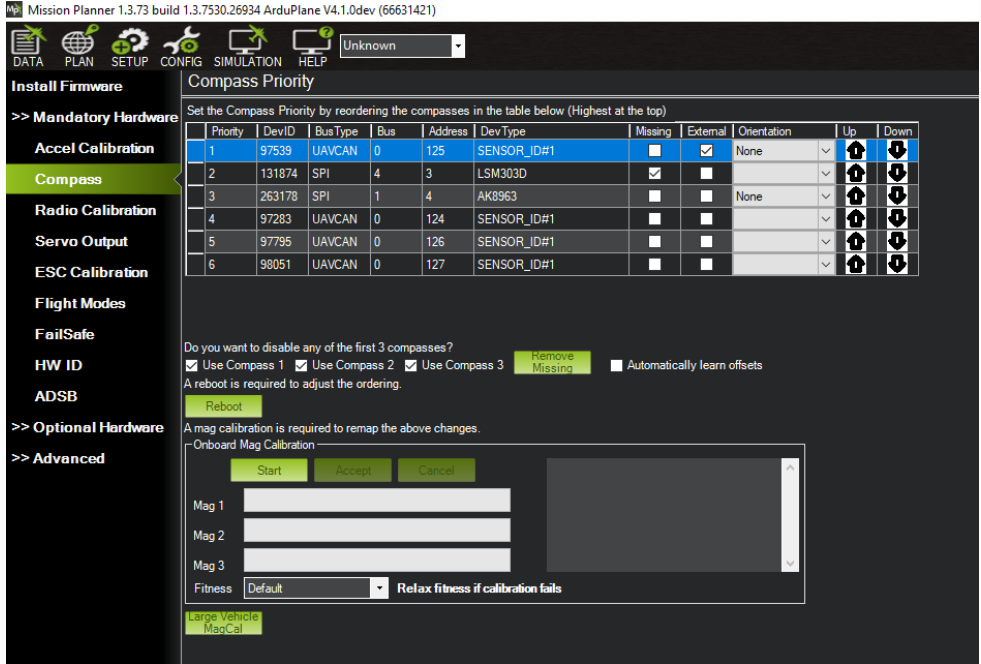


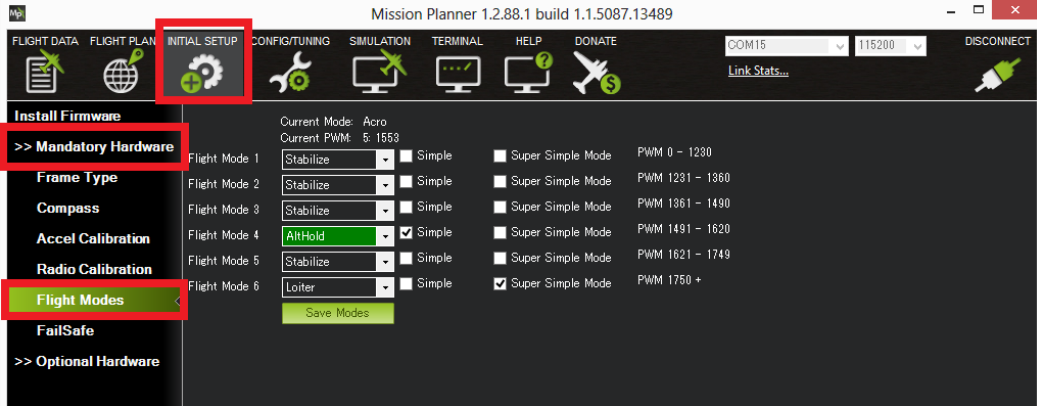
2.Calibration:



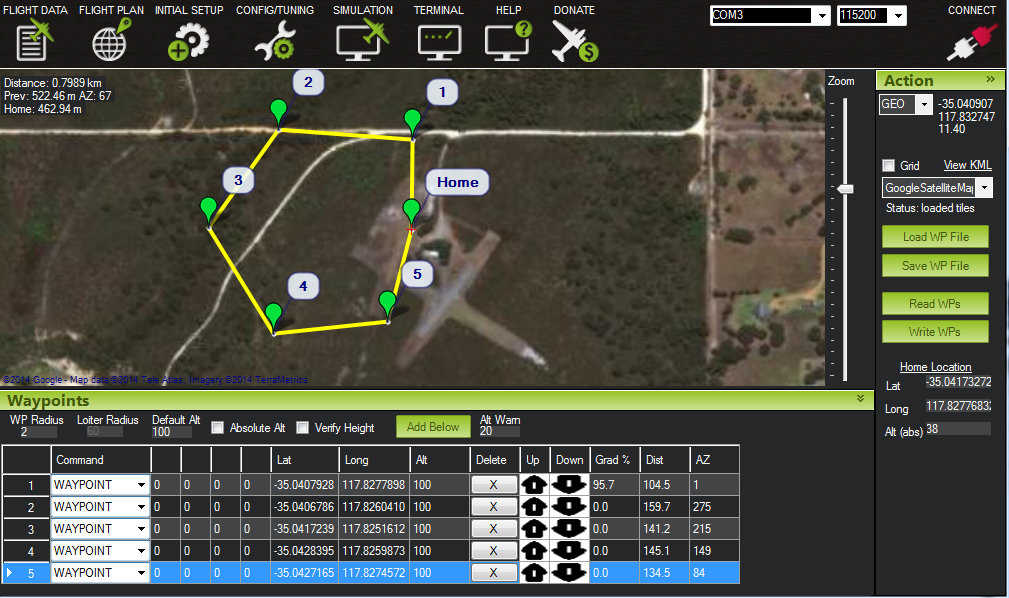
We calibrated compass and radio for compass calibration we needed a open ground and different step such as nose up,nose down,left,right,bottom facing.and also level calibration was needed.



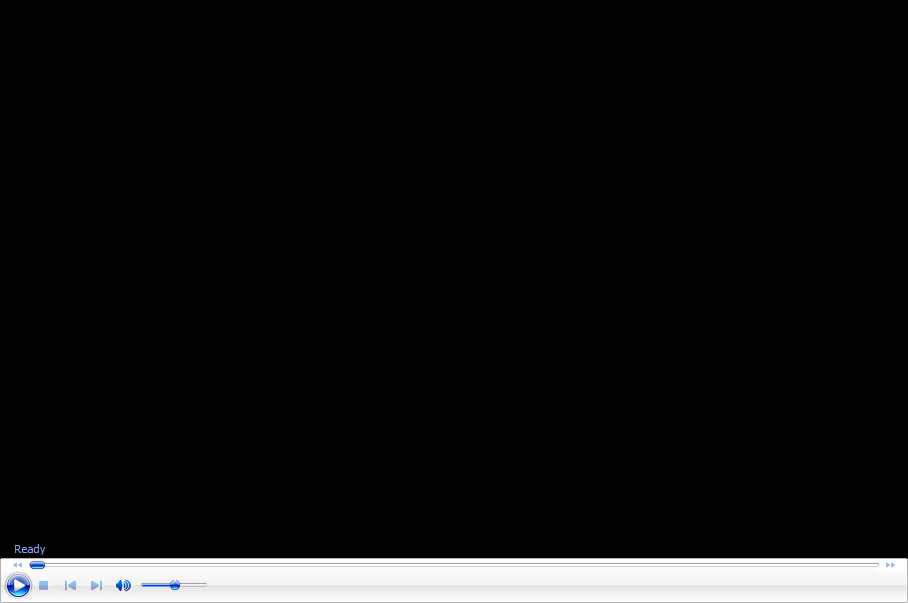




finally we checked the connections and bindings and mark the location on mission planner, and see for all gps and other components are armed and ready to fly and slowly throttling up.



Final flight:



Line following drone

How this is useful in daily life:

Uses:

1.In shopping malls:as for particular identity the drone picks up the product and send it to user on counter.

2.For identification of cracks in certain line(such as roads)and preventing any accidents.

How to work on it?

The work presented here ﬁnds an application within the

IFAC2020 MathWorks Minidrone competition [19], where

the use of a model-based design approach is the aim of

the speciﬁc contest. Path error and mission time are used as

evaluation metrics for the algorithm. The whole process is the

following: a quad-rotor UAV follows a pre-established red

path by using its downward facing camera to get feedback

from the environment. Images are updated according to

the position and orientation of the vehicle simulated in the

MATLAB VR world. No prior knowledge on the path and

the surrounding scenario is given. The drone takes off and

starts its motion looking at the path, and the mission stops

with the recognition of an end-marker. At that time, the drone

lands staying within the delimited area. Figure 2 shows the

considered scenario

The work presented here ﬁnds an application within the

IFAC2020 MathWorks Minidrone competition [19], where

the use of a model-based design approach is the aim of

the speciﬁc contest. Path error and mission time are used as

evaluation metrics for the algorithm. The whole process is the

following: a quad-rotor UAV follows a pre-established red

path by using its downward facing camera to get feedback

from the environment. Images are updated according to

the position and orientation of the vehicle simulated in the

MATLAB VR world. No prior knowledge on the path and

the surrounding scenario is given. The drone takes off and

starts its motion looking at the path, and the mission stops

with the recognition of an end-marker. At that time, the drone

lands staying within the delimited area. Figure 2 shows the

considered scenario

The work presented here ﬁnds an application within the

IFAC2020 MathWorks Minidrone competition [19], where

the use of a model-based design approach is the aim of

the speciﬁc contest. Path error and mission time are used as

evaluation metrics for the algorithm. The whole process is the

following: a quad-rotor UAV follows a pre-established red

path by using its downward facing camera to get feedback

from the environment. Images are updated according to

the position and orientation of the vehicle simulated in the

MATLAB VR world. No prior knowledge on the path and

the surrounding scenario is given. The drone takes off and

starts its motion looking at the path, and the mission stops

with the recognition of an end-marker. At that time, the drone

lands staying within the delimited area. Figure 2 shows the

considered scenario

Where the use of a model-based design approach is the aim of the speciﬁc contest. Path error and mission time are used as evaluation metrics for the algorithm. The whole process is the following: a quad-rotor UAV follows a pre-established Redpath by using its downward-facing camera to get feedback from the environment. Images are updated according to the position and orientation of the vehicle simulated in the theMATLAB VR world. No prior knowledge on the path and the surrounding scenario is given. The drone takes off and starts its motion looking at the path, and the mission stops with the recognition of an end-marker. At that time, the drone lands staying within the delimited area.

TOOLS USED IN THIS PROJECT:

1.Matlab R2022A

2.Simulink

3.Add-ons

(i)Simulink Support package for Parrotminidrones

(ii)UAV toolbox

(iii)Control system toolbox

(iv)Stateflow

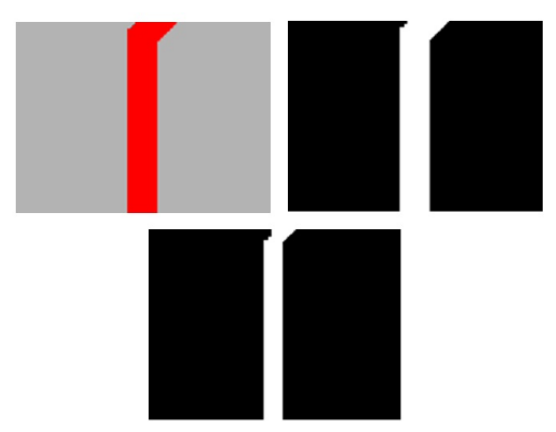
(v)Image processing toolbox

(vi)Computer vision toolbox

(vii)Aerospace toolbox

(viii)Aerospace blockset

1.Image processing toolbox: the main aim is to identify the path as follows.



Using image processing we can binarize the picture and get the specified picture.

Using code:

I = imread("IMG\_006.jpg");

gs = im2gray(I);

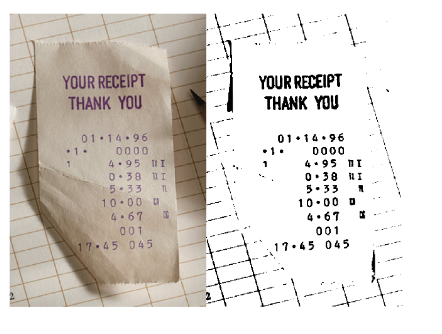
gsAdj = imadjust(gs);

BW = imbinarize(gsAdj,"adaptive","ForegroundPolarity","dark");

imshowpair(I,BW,"montage")

F = fspecial("average",*n*)

This converts the image into binarize and adjustable iso to correctly identify the path.



I = imread("IMG\_002.jpg");

gs = im2gray(I);

gs = imadjust(gs);

H = fspecial("average",3);

gs = imfilter(gs,H,"replicate");

SEdisk = strel("disk",8);

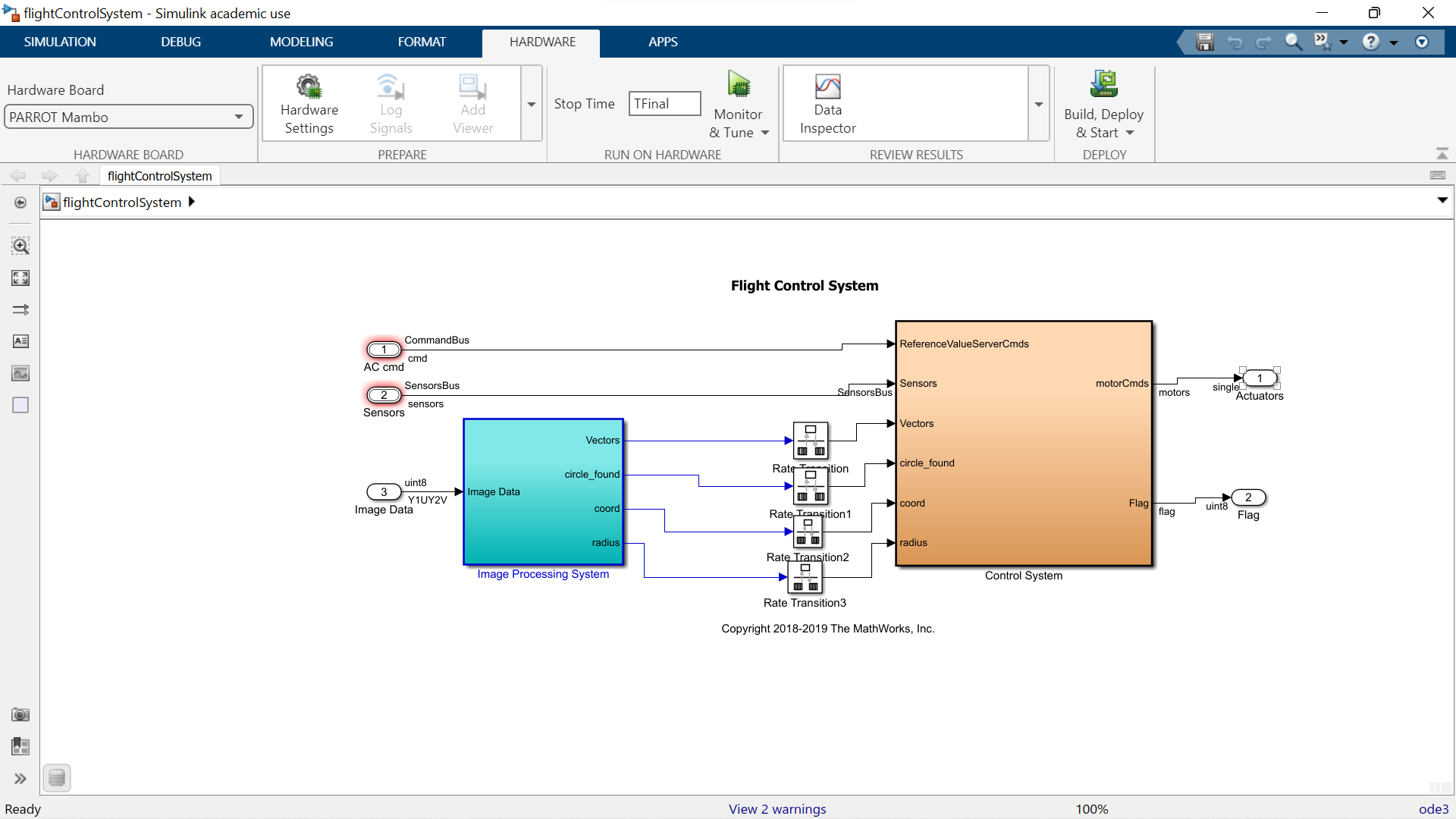
Ibg = imclose(gs,SEdisk);

gsSub = Ibg - gs;

BW = ~imbinarize(gsSub);

imshowpair(I,BW,"montage")

2.Simulink



Using simulink we created a flow how the components re connected across the Pixhawk and how it functions to follow a certain path.

(flight control system)

