**UAV SWARM**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the*

*Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

**K SAI SHRUTHI<1602-20-737-036>**

**AKSHITHA DOODALA<1602-20-737-003>**

# JIBA NAFEES FATHIMA<1602-20-737-017>



**Department of Information Technology**

**Vasavi College of Engineering**

**(Autonomous)**

# VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)

**(AFFILIATED TO OSMANIA UNIVERSITY) HYDERABAD - 500 030**

## Department of Information Technology



## DECLARATION BY CANDIDATE

We, **K SAI SHRUTHI, AKSHITHA DOODALA, JIBA NAFEES FATHIMA,** bearing hall ticket number, **1602-20-737-036,1602-20-737-003,1602-20-737-017** hereby declare that the project report entitled **”UAV SWARM”** Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by me and the results embodied in this project report has not been submitted to any other university or institute for the award of any other degree or diploma.

**K SAI SHRUTHI (1602-20-737-036)**

**AKSHITHA DOODALA (1602-20-737-003) JIBA NAFEES FATHIMA (1602-20-73-017)**

**VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)**

**(AFFILIATED TO OSMANIA UNIVERSITY) HYDERABAD - 500 030**

## Department of Information Technology



## BONAFIDE CERTIFICATE

This is to certify that the project entitled “**UAV SWARM**” being submitted by

**K SAI SHRUTHI, AKSHITHA DOODALA, JIBA NAFEES FATHIMA** bearing **1602-20-737-036, 1602-20-737-003, 1602-20-737-017**, in partial fulfilment of the requirements for the completion of MINI PROJECT of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

Internal Guide External Examiner Dr.K Ram Mohan Rao

Mrs Sireesha Panyam HOD, IT

**ACKNOWLEDGEMENT**

We thank the department of INFORMATION TECHNOLOGY, for introducing the subject “Mini Project-2” in BE fifth semester.

We would also like to take this opportunity to express our gratitude to our guide who initiated this project, **Dr K. Raghavendra**, Scientist SG, ADRIN (ISRO).

We would also like to show our appreciation to our Honourable principal, Dr S V Ramana sir, our HOD K. Ram Mohan Rao for supporting us and our mini project lecturer, Mrs Sireesha mam, for letting us properly understand the process of doing a project and for providing valuable insight and expertise that has greatly assisted us in the making of the project.

**TABLE OF CONTENTS:**

## 1 INTRODUCTION---------------------------------------------------------------6

* 1. PURPOSE -------------------------------------------------------------------7
  2. INTENDED AUDIENCE ------------------------------------------------7
  3. PRODUCT SCOPE--------------------------------------------------------7
  4. PROBLEM DEFINITON------------------------------------------------8

## 2. RELATED WORK -------------------------------------------------------------8

## 3. PROPOSED WORK------------------------------------------------------------9

### 3.1 USE CASES---------------------------------------------------------------9

### 3.2 UI PROTOTYPES--------------------------------------------------------10

### 3.3 ARCHITECTURE--------------------------------------------------------11

### 3.4 DESIGN -------------------------------------------------------------------13

### 3.5 IMPLEMENTATION----------------------------------------------------14

3.5.1 PACKAGES USED------------------------------------------------14

3.5.2 CODE AND GITHUB LINKS------------------------------------15

4. RESULTS ------------------------------------------------------------------------27

5. DISCUSSION AND FUTURE WORK --------------------------------------31

6. REFERENCES-------------------------------------------------------------------32

# ABSTRACT

* Design and development of Swarm UAV system using Cellular (GPRS) communication.
* Design of Graphical User Interfaces (GUI) for sending and receiving messages from Ground Control Station (GCS)/Laptop to the IOT board (Raspberry pi).
* Testing the Swarm in lab environment for one UAV and can be extendable to 8 numbers or more.

## CHAPTER 1 INTRODUCTION

**What is a UAV SWARM?**

UAV SWARM is the collection of drones working together or individually depending on the user’s specification. The software being designed is used to control this Swarm in order to make it easier for the user by building a GUI for it to work automatically.

**Why do we need a UAV SWARM software?**

* 1. This software makes it easier to control a swarm.
  2. It can be used to map or survey large areas in a short period of time, providing vital information for tactical operations, precision agriculture, utility inspection, entertainment purpose and more.

**1.1 PURPOSE**

UAV SWARM software basically provides an interface for the user or the people who want to control a swarm of drones for any possible purpose like-

Agriculture, Military purposes, Entertainment and many more.

It precisely overcomes the problem of controlling a swarm manually which requires a lot of work.

**1.2 INTENDED AUDIENCE**

The intended audience for this project is everyone who wants to work with numerous drones altogether using software.

**1.3 PRODUCT SCOPE**

UAV SWARM software provides the user/market with a GUI which has its end connected to the hardware unit of drones which are in use. This can be used by the market on large scale for various purposes like a search and rescue, surveillance, traffic monitoring, weather monitoring, agriculture and many more purposes.

**1.4 PROBLEM DEFINITION**

# A swarm of drones usually are controlled manually using remotes by specifying one remote for each drone flying. To overcome this issue, we have decided to build a simplified solution by making a software to fly swarm of drones from anywhere on a single click.

# CHAPTER 2

# RELATED WORK –

There has been a lot of work done in the domain of controlling a swarm of drones. Many software have been updated in order to easily control a swarm.

Our agenda is to design a more reliable software to control drones with an effective GUI connecting the back end of the hardware using socket programming.

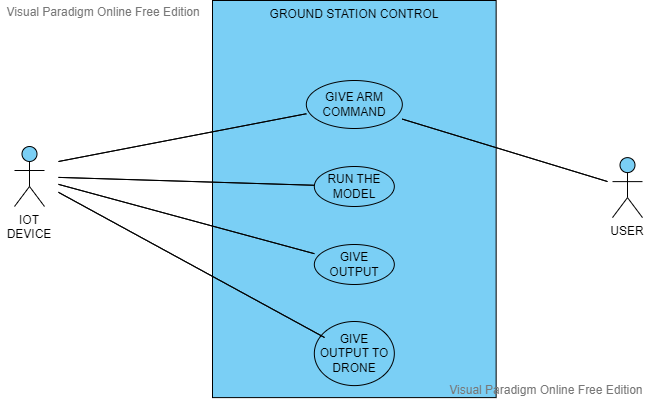
In most cases each individual UAV is simultaneously controlled by a GCS. Traditional UAV swarms use a computer as a GCS running a ground control software.

The computers are equipped with a transceiver that sends and receives telemetry data from connected UAVs. Traditionally these transceivers use unlicensed radio frequency bands, such as 900 MHz, to send and receive the data. Higher levels of autonomy would allow UAVs to make decisions using on-board computers.

## CHAPTER 3

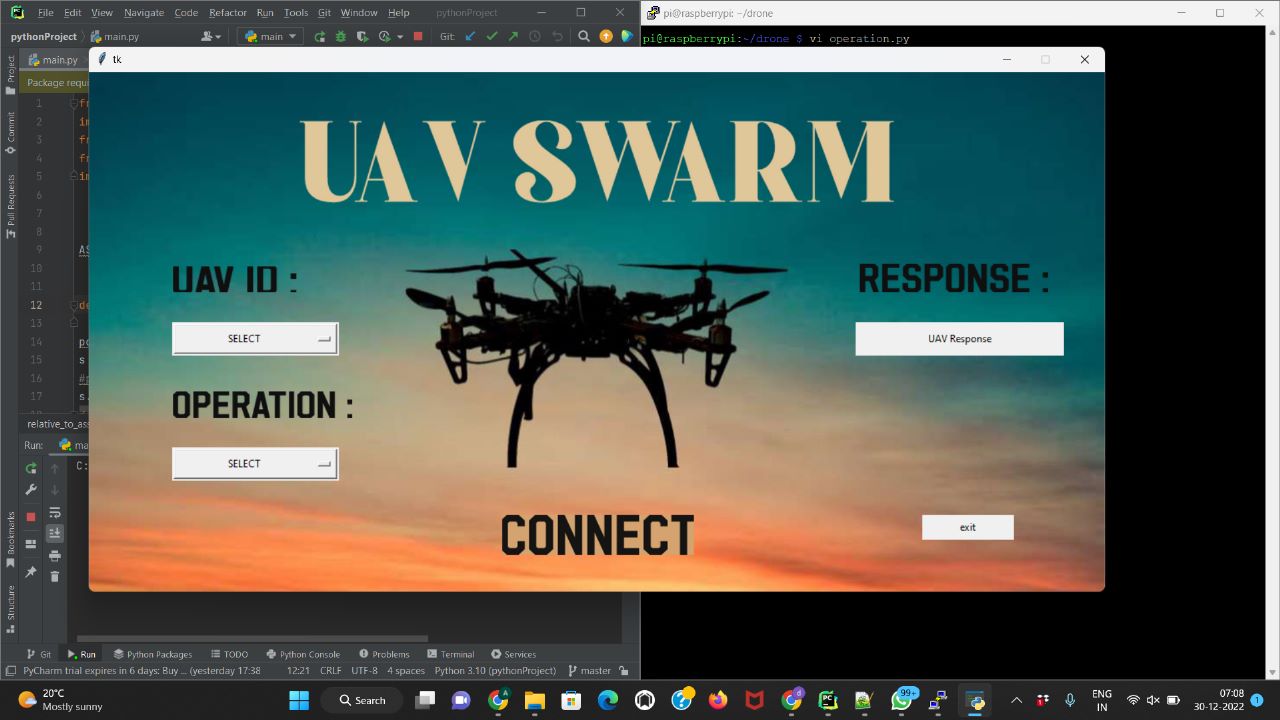
## PROPOSED WORK –

## 3.1 Use cases –

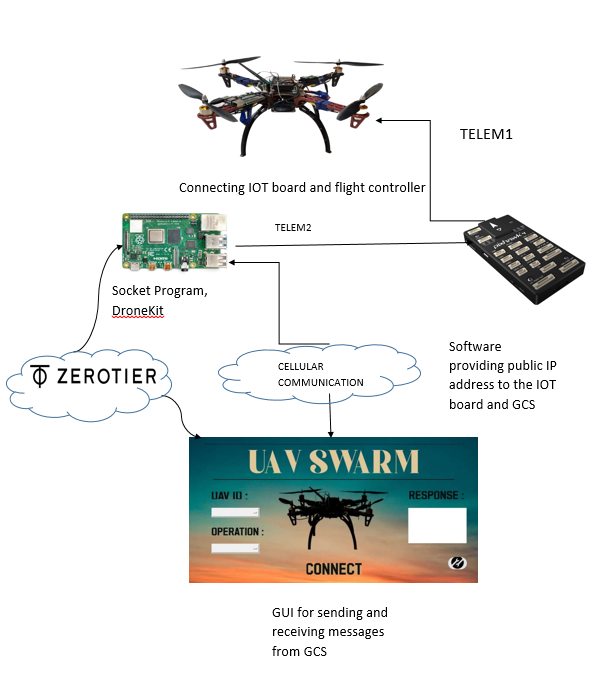
****

## 3.2 UI prototypes or screenshots





## 3.3 Architecture and Technology used –



To begin with, there are five main components in our architecture.

Namely:

1. **GUI connect to the Ground Control Station (GCS):** This takes the messages from the Ground station and sends it to the GUI. The messages indicate commands being passed from the GCS.

2. **IOT Board (Raspberry pi):** It acts as an operating system to connect the GUI to the drone. It is socket programmed with packages installed from Drone-Kit to make a connection with the flight controller.

3. **Pixhawk (Flight controller):** It is the interface which connects the IOT Board to the drone using the ports TELEM1 and TELEM2.

4. **Drone:** The UAV which performs various actions depending on the commands passed by the user like ARM, DISARM.

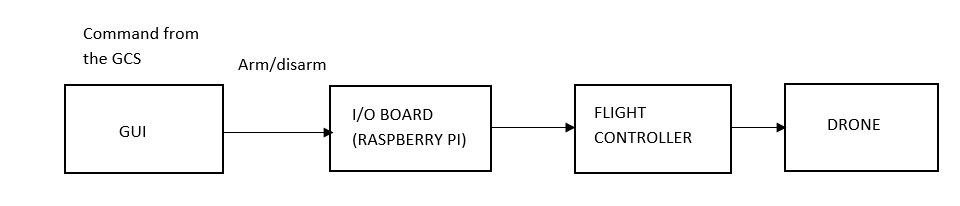
5. **Zero-Tier:** Software which provides public IP addresses to the IOT Board and GUI connected to the GCS.

**Technology used –**

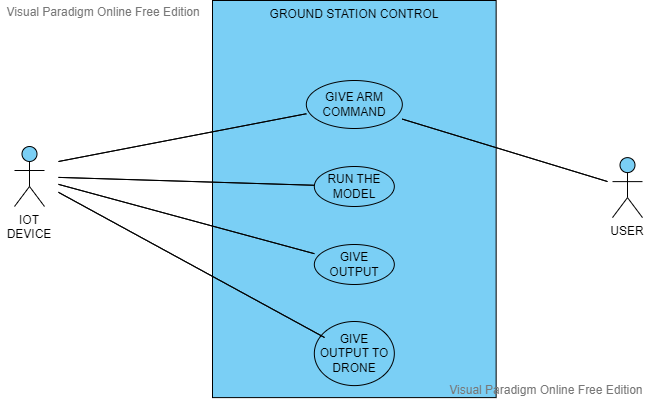
The technology which is used includes the software requirements. The software used is the Drone-Kit, acting as the interface which provides packages to be installed onto the raspberry pi in order to establish a connection between the Raspberry pi and the Flight controller.

Python3.9 using Tkinter allowing us to build an effective GUI.

**3.4 Design – DATA FLOW DIAGRAM**



**USE CASE DIAGRAM**



## 3.5 Implementation –

At its current stage, the chatbot is trained to answer various kinds of queries posed by outside visitors.

3.5.1- Packages used –

INSTALLATIONS

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install python3-pip3

sudo apt-get install python3-dev

sudo pip install future

sudo apt-get install screen python3-wxgtk4.0 python3-lxml

sudo pip3 intsall pyserial

sudo pip3 install dronekit

sudo pip3 install MAVProxy

**3.5.2 – Code –**

//DRONE-KIT ARM CODE

from dronekit import connect, VehicleMode, LocationGlobalRelative,APIException

import time

import socket

import math

import argparse

def connectMyCopter():

parser=argparse.ArgumentParser(description='commands')

parser.add\_argument('--connect')

args=parser.parse\_args()

connection\_string=args.connect

baud\_rate=57600

vehicle=connect(connection\_string,baud=baud\_rate,wait\_ready=True)

return vehicle

def arm():

vehicle.armed=True

print("DONE")

return None

s = socket.socket();

port = 12307

s.bind(('172.26.24.107', port))

s.listen(5)

c, addr = s.accept()

print("CONNECTED")

datarecv=s.recv(1024).decode()

print(datarecv);

vehicle=connectMyCopter()

cond='arm'

if datarecv==cond:

arm()

//SERVER.PY

from dronekit import connect, VehicleMode, LocationGlobalRelative,APIException

import time

import socket

import math

import argparse

def connectMyCopter():

parser=argparse.ArgumentParser(description='commands')

parser.add\_argument('--connect')

args=parser.parse\_args()

connection\_string=args.connect

baud\_rate=57600

vehicle=connect(connection\_string,baud=baud\_rate,wait\_ready=True)

return vehicle

def arm():

vehicle.armed=True

print("DONE")

return None

s = socket.socket();

port = 12307

s.bind(('172.26.24.107', port))

s.listen(5)

c, addr = s.accept()

print("CONNECTED")

datarecv=s.recv(1024).decode()

print(datarecv);

vehicle=connectMyCopter()

cond='arm'

if datarecv==cond:

arm()

//RASPBERRY PI

from dronekit import connect, VehicleMode, LocationGlobalRelative,APIException

import time

import socket

import math

import argparse

def connectMyCopter():

parser=argparse.ArgumentParser(description='commands')

parser.add\_argument('--connect')

args=parser.parse\_args()

connection\_string=args.connect

baud\_rate=57600

vehicle=connect(connection\_string,baud=baud\_rate,wait\_ready=True)

return vehicle

def arm():

vehicle.armed=True

print("DONE")

return None

s = socket.socket();

port = 12307

s.bind(('172.26.24.107', port))

s.listen(5)

c, addr = s.accept()

print("CONNECTED")

datarecv=s.recv(1024).decode()

print(datarecv);

vehicle=connectMyCopter()

cond='arm'

if datarecv==cond:

arm()

//GUI CODE:

//ALONG WITH GUI

from pathlib import Path

import socket

from tkinter import \*

from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage

import tkinter.messagebox

ASSETS\_PATH = Path(r"C:\Users\91630\Desktop\SEM 5\MPI\frame0\frame0\frame0\frame0")

def relative\_to\_assets(path: str) -> Path:

return ASSETS\_PATH / Path(path)

port=2348

s = socket.socket()

#port = 12345

s.connect(('172.26.24.107', port))

def onClick():

text1=clicked.get()

text2=click.get()

if (text1=='UAV 1' and text2=='ARM')or(text1=='UAV 2' and text2=='ARM')or(text1=='UAV 3' and text2=='ARM')or(text1=='UAV 4' and text2=='ARM')or(text1=='UAV 5' and text2=='ARM')or(text1=='UAV 6' and text2=='ARM')or(text1=='ALL UAVS' and text2=='ARM'):

tkinter.messagebox.showinfo("ARM","WILL ARM IN 20sec")

my\_string\_var.set("The operation of selected UAV is: ARM")

s.send('arm'.encode())

elif (text1=='UAV 1' and text2=='DISARM')or(text1=='UAV 2' and text2=='DISARM')or(text1=='UAV 3' and text2=='DISARM')or(text1=='UAV 4' and text2=='DISARM')or(text1=='UAV 5' and text2=='DISARM')or(text1=='UAV 6' and text2=='DISARM')or(text1=='ALL UAVS' and text2=='DISARM'):

tkinter.messagebox.showinfo("DISARM","WILL DISARM SOON")

my\_string\_var.set("The operation of selected UAV is: DISARM")

s.send('disarm'.encode())

elif (text1=='UAV 1' and text2=='TAKEOFF')or(text1=='UAV 2' and text2=='TAKEOFF')or(text1=='UAV 3' and text2=='TAKEOFF')or(text1=='UAV 4' and text2=='TAKEOFF')or(text1=='UAV 5' and text2=='TAKEOFF')or(text1=='UAV 6' and text2=='TAKEOFF')or(text1=='ALL UAVS' and text2=='TAKEOFF'):

tkinter.messagebox.showinfo("TAKEOFF","READY TO TAKEOFF")

my\_string\_var.set("The operation of selected UAV is: TAKEOFF")

s.send('takeoff'.encode())

elif (text1=='UAV 1' and text2=='LAND')or(text1=='UAV 2' and text2=='LAND')or(text1=='UAV 3' and text2=='LAND')or(text1=='UAV 4' and text2=='LAND')or(text1=='UAV 5' and text2=='LAND')or(text1=='UAV 6' and text2=='LAND')or(text1=='ALL UAVS' and text2=='LAND'):

tkinter.messagebox.showinfo("LAND","ALL SET TO LAND")

my\_string\_var.set("The operation of selected UAV is: LAND")

s.send('land'.encode())

s.close()

def close():

s = socket.socket()

s.connect(('172.26.24.107', port))

s.send('exit'.encode())

s.close()

window.quit();

window = Tk()

window.geometry("1219x623")

window.configure(bg = "#FFFFFF")

canvas = Canvas(

window,

bg = "#FFFFFF",

height = 623,

width = 1219,

bd = 0,

highlightthickness = 0,

relief = "ridge"

)

canvas.place(x = 0, y = 0)

image\_image\_1 = PhotoImage(

file=relative\_to\_assets("image\_1.png"))

image\_1 = canvas.create\_image(

609.0,

343.0,

image=image\_image\_1

)

image\_image\_2 = PhotoImage(

file=relative\_to\_assets("image\_2.png"))

image\_2 = canvas.create\_image(

175.0,

248.0,

image=image\_image\_2

)

image\_image\_3 = PhotoImage(

file=relative\_to\_assets("image\_3.png"))

image\_3 = canvas.create\_image(

208.0,

399.0,

image=image\_image\_3

)

#DROPBOXES

clicked = StringVar()

clicked.set("SELECT")

drop1 = OptionMenu(window, clicked, "UAV 1","UAV 2","UAV 3","UAV 4","UAV 5","UAV 6","ALL UAVS")

drop1.place(

x=100,

y=300,

width=200,

height=40

)

click = StringVar()

click.set("SELECT")

drop2 = OptionMenu(window, click, "ARM","DISARM","TAKEOFF","LAND")

drop2.place(

x=100,

y=450,

width=200,

height=40

)

#TEXT BOX FOR RESPONSE

my\_string\_var = StringVar()

my\_string\_var.set("UAV Response")

my\_label = Label(window,textvariable=my\_string\_var).place(x=920.0,y=300.0,width=250.0,height=40.0)

image\_image\_5 = PhotoImage(

file=relative\_to\_assets("image\_5.png"))

image\_5 = canvas.create\_image(

1038.0,

247.0,

image=image\_image\_5

)

image\_image\_6 = PhotoImage(

file=relative\_to\_assets("image\_6.png"))

image\_6 = canvas.create\_image(

609.0,

107.0,

image=image\_image\_6

)

button\_image\_1 = PhotoImage(

file=relative\_to\_assets("button\_1.png"))

button\_1 = Button(

image=button\_image\_1,

borderwidth=0,

highlightthickness=0,

command=onClick,

relief="flat"

)

button\_1.place(

x=496.0,

y=531.0,

width=230.0,

height=48.0

)

button\_2 = Button(text="exit",

borderwidth=0,

highlightthickness=0,

command=close,

relief="flat"

)

button\_2.place(

x=1000.0,

y=531.0,

width=110.0,

height=30.0

)

window.resizable(False, False)

window.mainloop()

**3.5.3 – GitHub Links –**

<https://github.com/Shruthi-Kovvur/UAV-SWARM>

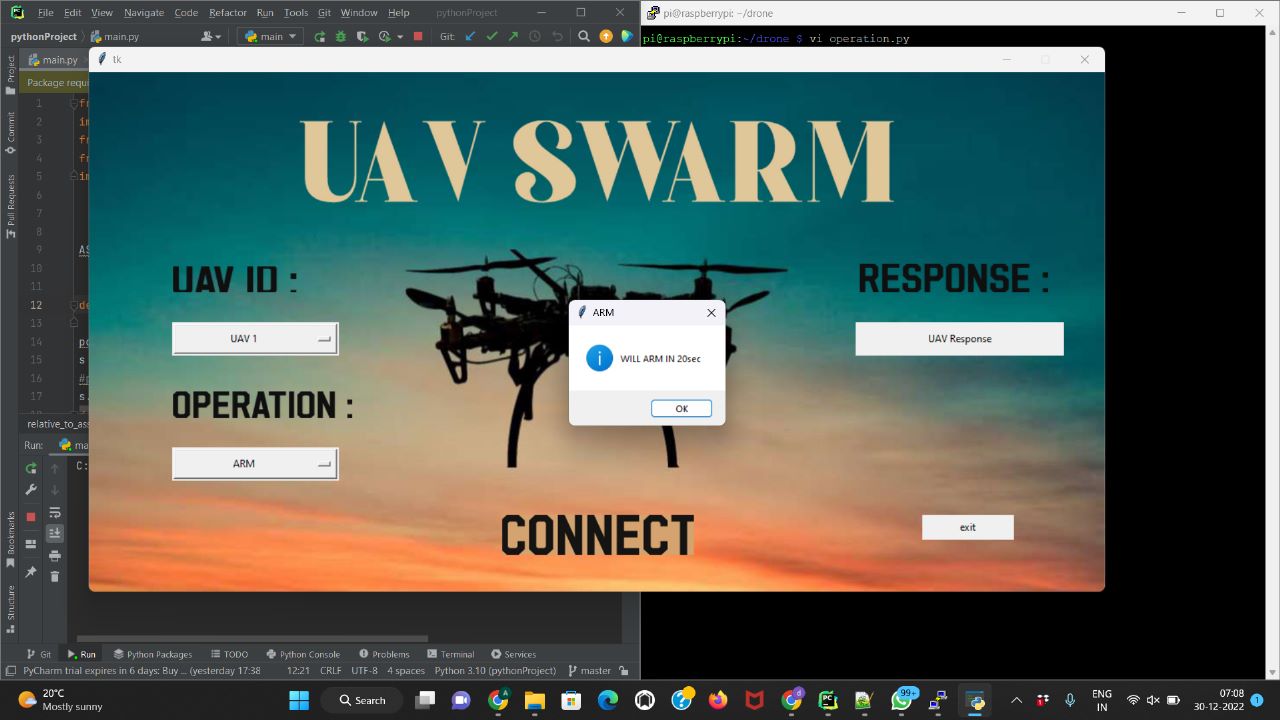
## 

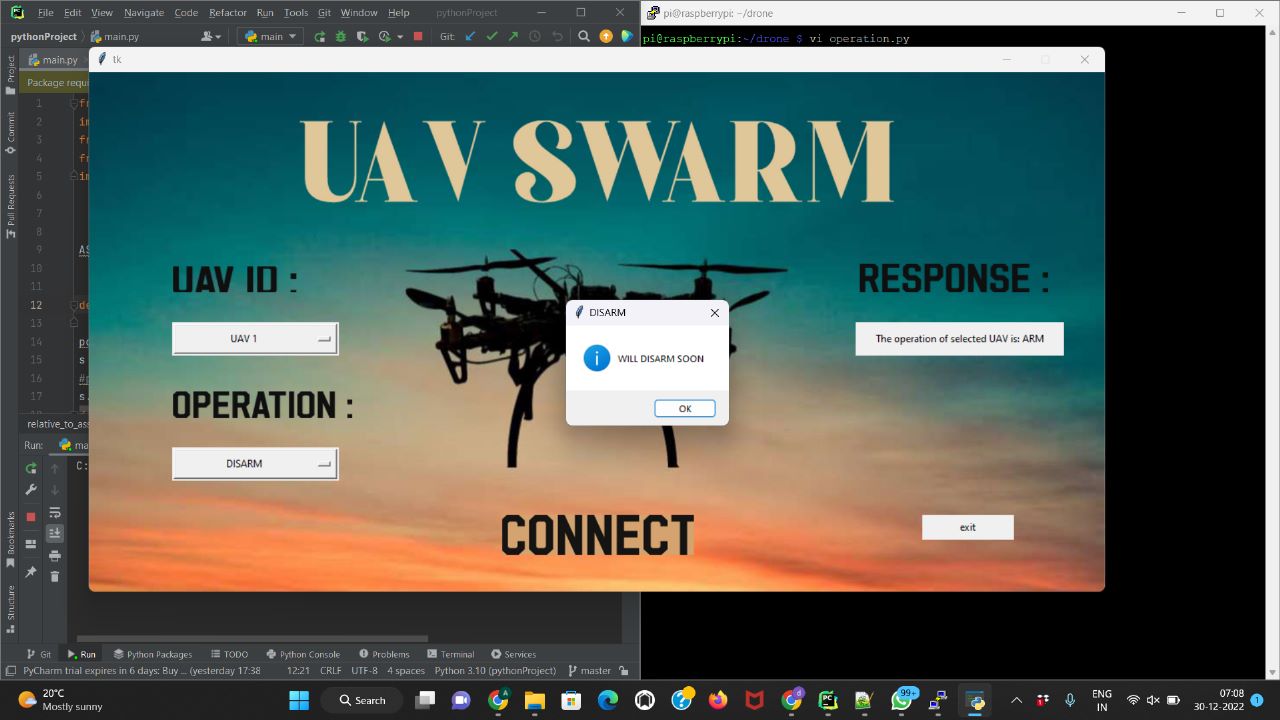
## CHAPTER 4 –

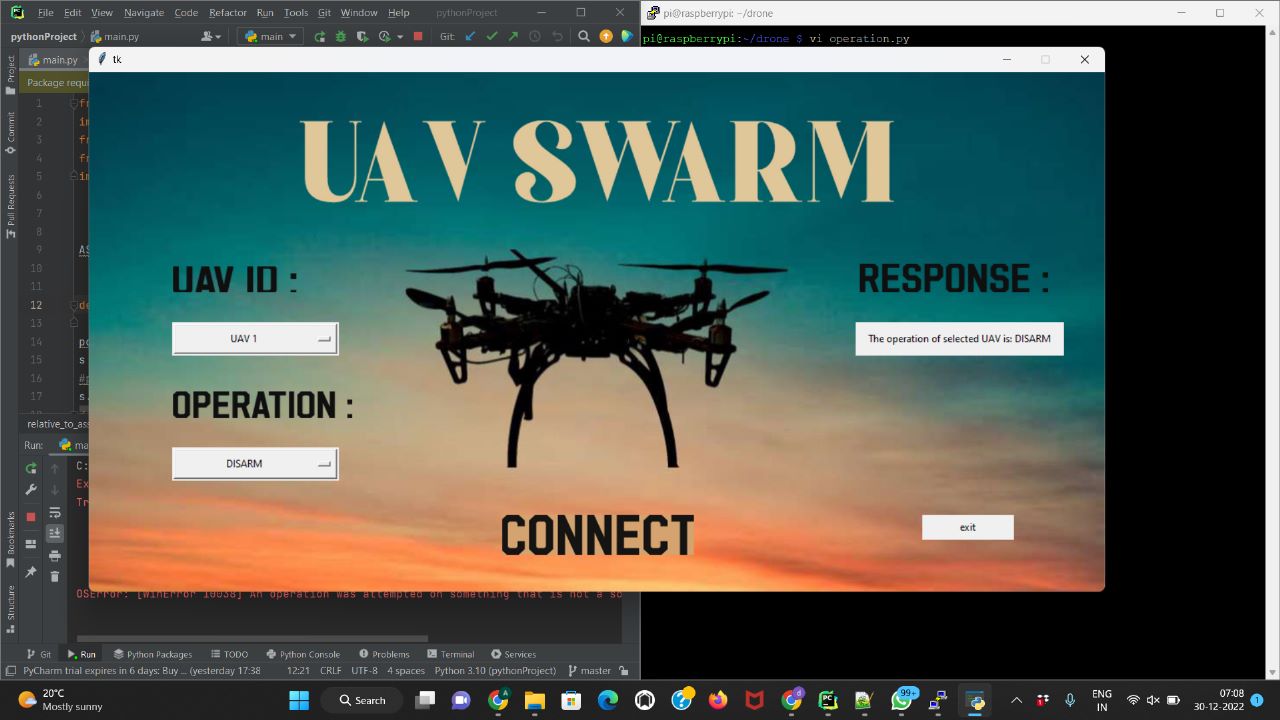
## RESULTS

The following are the results obtained after implementation –

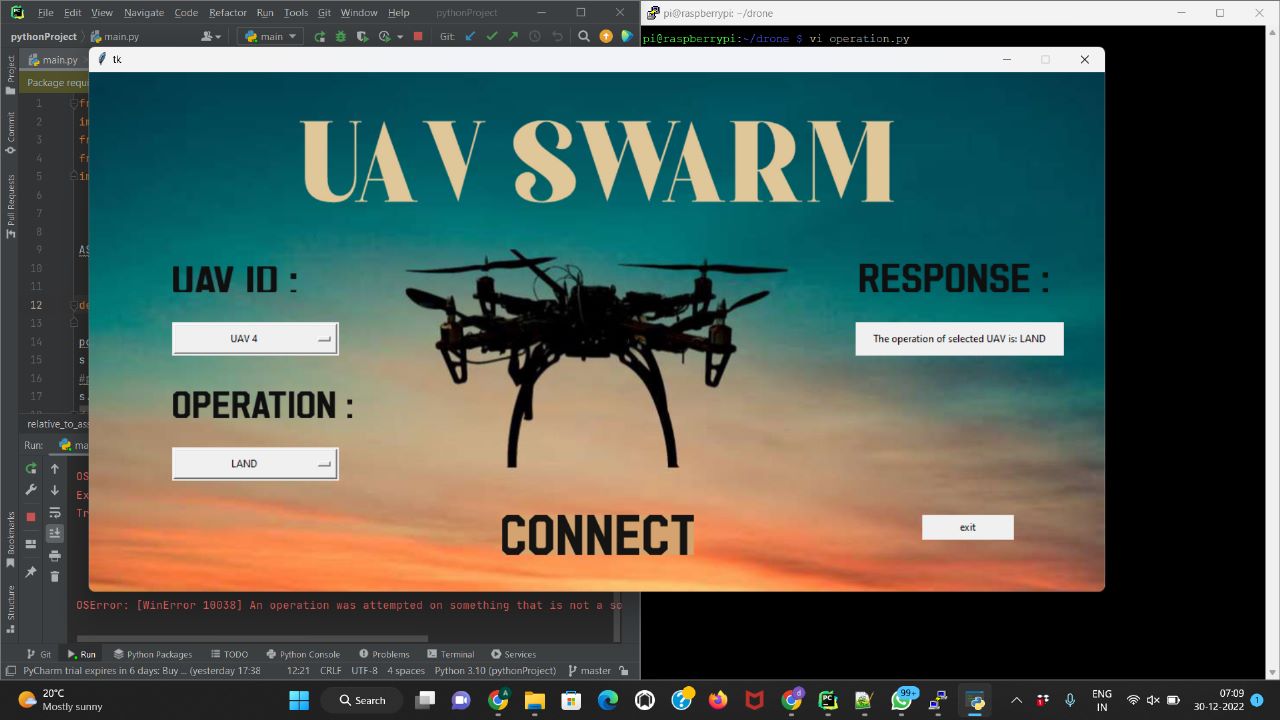
1. Selecting the UAV 1 and ARM Operation

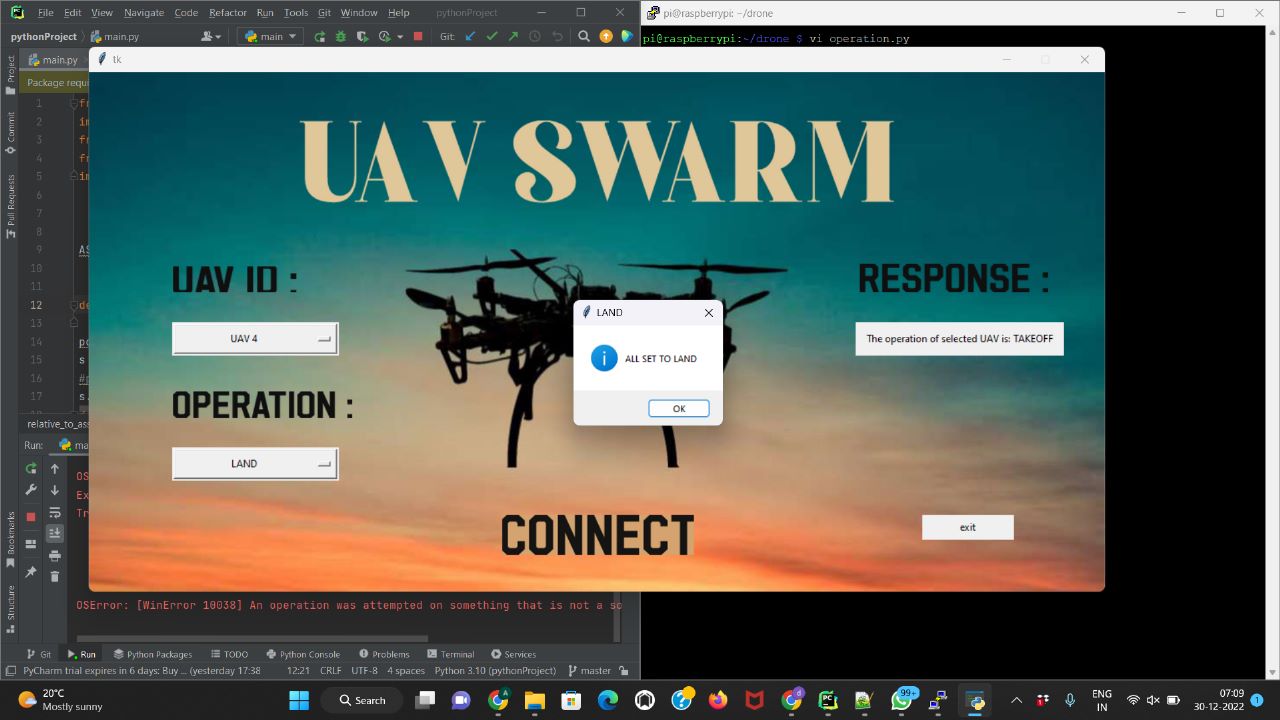


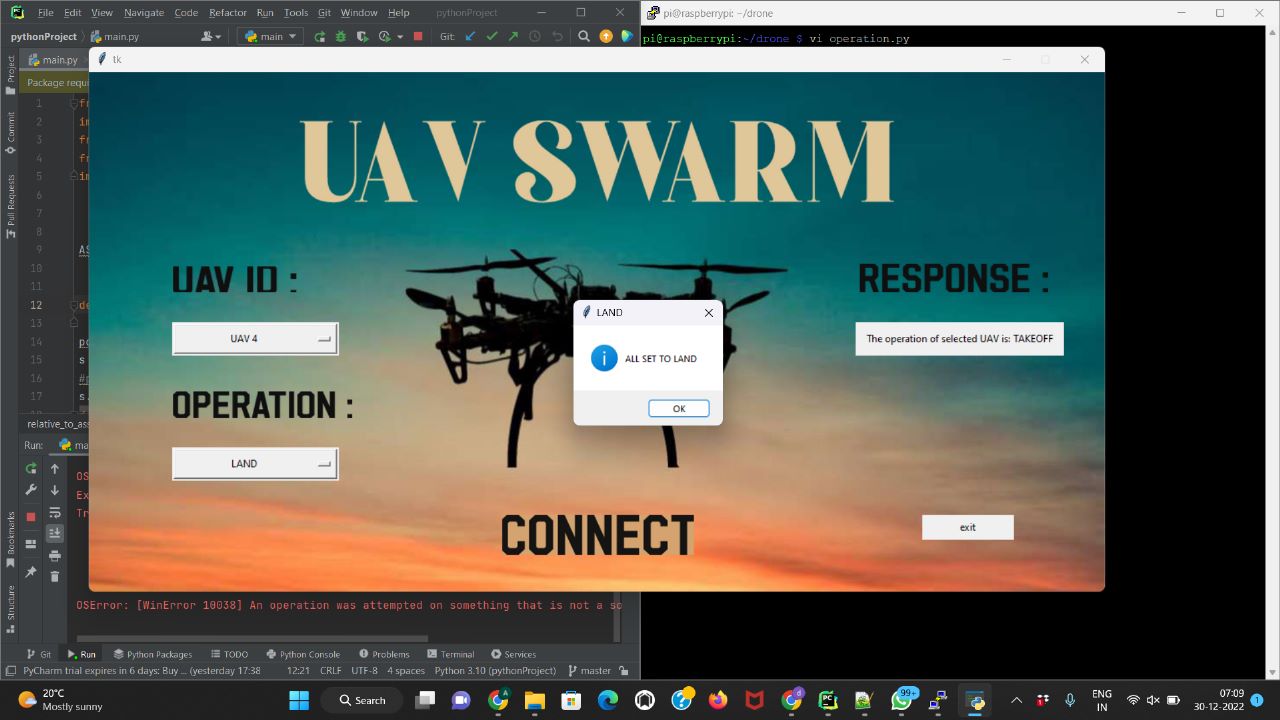
2.The drone will perform DISARM after a few seconds.



3.Selecting for other UAV’s







**CHAPTER 5**

**DISCUSSION AND FUTURE WORK –**

Drone/UAV SWARM technology is always evolving, **the future for drones is very exciting,** these can be used by the market on large scale for various purposes like a search and rescue, surveillance, traffic monitoring, weather monitoring, agriculture and many more purposes.

The UAV SWARM project can be extended to have an advanced Graphical User Interface and can be tested with more real-time UAVS to use for drone light show or any other applications.

The software developed should further be able generate a pattern for simulating a swarm of drones.

## CHAPTER 6

## REFERENCES –

* https://dronekit-python.readthedocs.io/en/latest/develop/sitl\_setup.html
* https://dronekit.io/
* https://www.raspberrypi.org/
* <https://www.youtube.com/watch?v=CUEbTx1Fh1>w
* <https://youtube.com/playlist?list=PLgiealSjeVyx3t4N9GroE29SbVwhYrOtL>
* https://www.digitalocean.com/community/tutorials/python-socket-programming-server-client