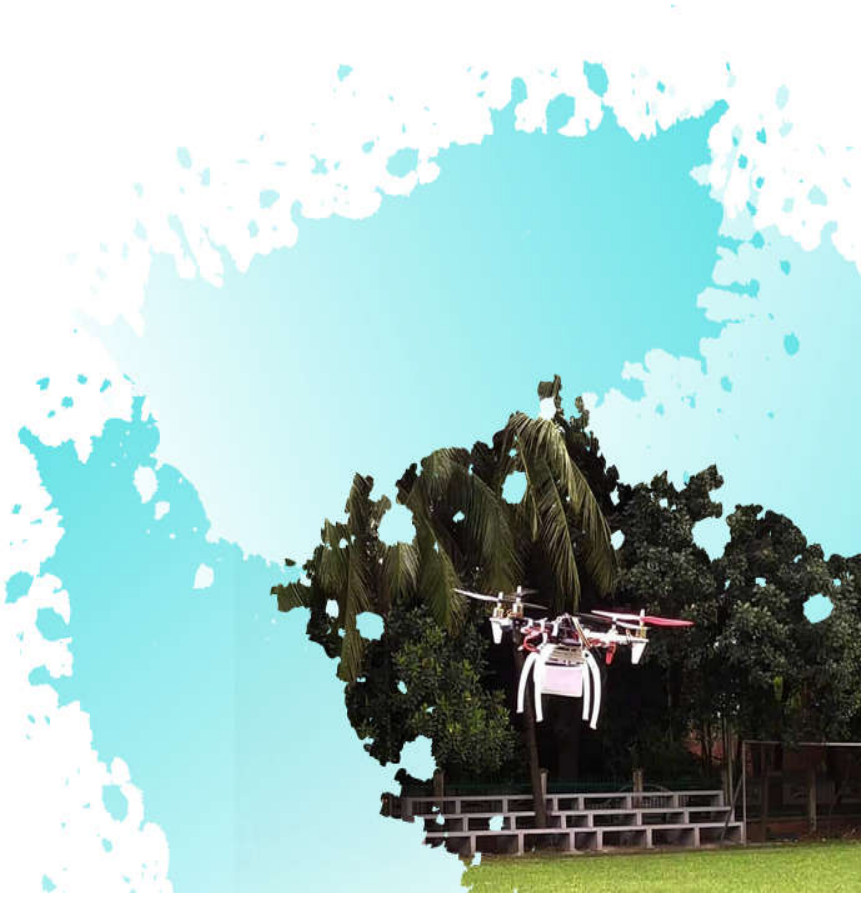
Bangladesh University of Engineering and Technology

EsPro Medicine Delivery

MANUAL

Project on *Electro-Mechanical System Design & Practices*
Course No: *ME 366*
Course Name: *Instrument & Measurement Sessional*



Presented by- ***Group 1***

Partho Kundu (1810011)
Puja Rani Paul (1810032)
Faria Hoque Bhuiya (1810039)
Tahsin-Ul Haque (1810060)

Table of Content

Content	Page No.
Introduction	*1*
Specifications	*1*
Features	*1*
Demonstration	*1*
Components	*2*
Images	*3*
Source Code	*4*
Challenges	*8*
Future Prospects	*9*
References	*9*
Acknowledgement	*9*

Introduction

EsPro is a quadcopter which is used to deliver medicine from local vendor to the customer. It can also detect authorized customer using computer vision based program. The manufacturing and control system of the drone has a user friendly interface so that people with minimum training can be in charge of it.

Specifications

- Weight: 1.5 kg
- Pay Load: 0.5 kg
- Battery Power (Max/Min): 11.1 V/10.5 V
- Thrust Generated: 3600 gm
- Flight Time: 10 minutes
- L×W×H (cm): 45×45×25
- Range: 1.5 km
- Tentative Manufacturing Cost: 20,000-22,000/-TK

Features

- Store to customer medicine delivery
- Navigation by using GPS module
- Customer authentication by QR scanning
- Real time visualization

Demonstration

https://drive.google.com/file/d/1dS_PM8ZE1Zt2CHK4sEKT10xjhz0PtYWt/view?usp=sharing

Components

Hardware:

Frame: Material- Plastic
Wheelbase – 450 mm

Motor: Type: Brushless DC motor, Model: A2212/13T
RPM/V: 1000 kV, Current: 4-10 A (at 75%)
No load current: 10 V/0.5 A
Shaft diameter: 3.17 mm, Dimensions: 27.5*28 mm
Thrust: 900 grams per motor, Total thrust:900*4 = 3.6 kg

ESC: Simonk 30A (current rating)

Battery: 2200 mAh, 11.1 V, 3 cell LiPo Battery

Propeller: Length – 10 inch

Flight controller: Ardupilot APM 2.8 Flight Control Board

GPS Module: Ublox NEO-M8N GPS Module
Features:
Built-in compass GPS module
Main chip: Ublox-M8N
Fast satellite searching speed and high precision

Receiver transmitter: FlySky FS-i6
2.4G 6CH AFHDS RC Transmitter with FS-i6 Receiver
Bandwidth (kHz): 500, No. of channels – 6
Antenna length – 26 mm*2(dual antenna)
Transmitting power – maximum 20 dam
RF Receiver sensitivity -105 dbm

Power module: V2.0, Maximum output current–3A, 3-14S input voltage

Landing gear

ESP 32 camera+ Wi-Fi module

Shock absorber

Arduino UNO

Software

Arduino IDE

Python

Mission Planner 1.3.72

ArduCopter_APM_2.0_Firmware_3.2.1

Images

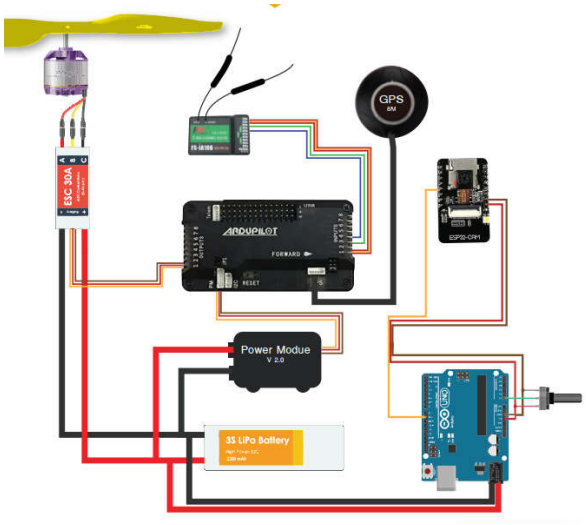


Figure 1: Circuit Diagram

Figure 2: Drone Components

Figure 3: Drone on the way to deliver medicine

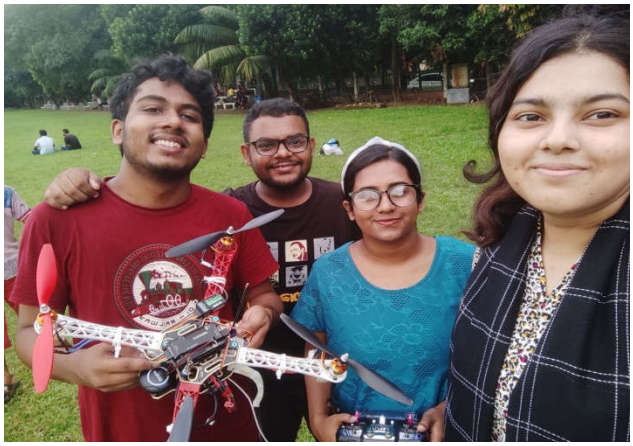
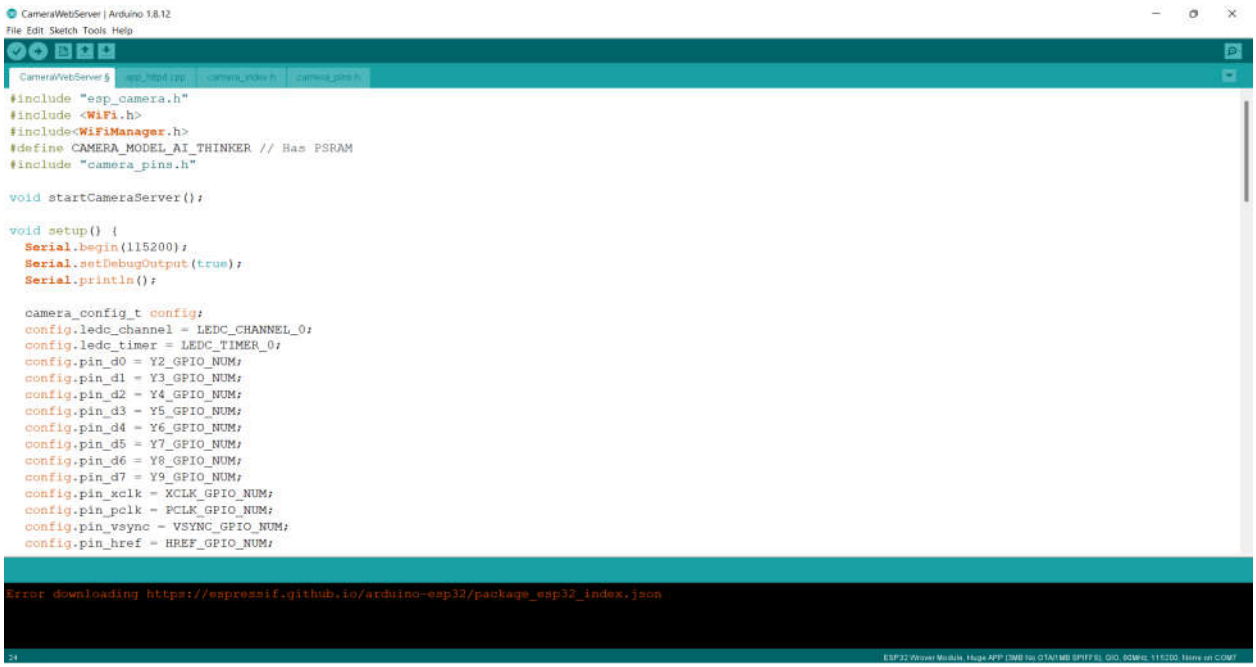


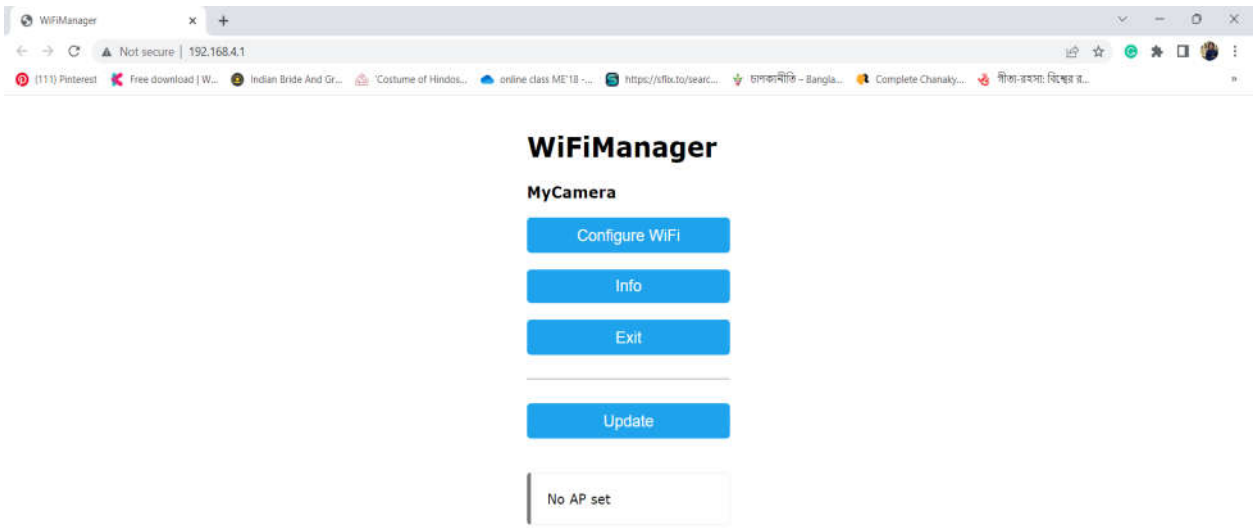
Figure 4: Group 1 members

Source Code

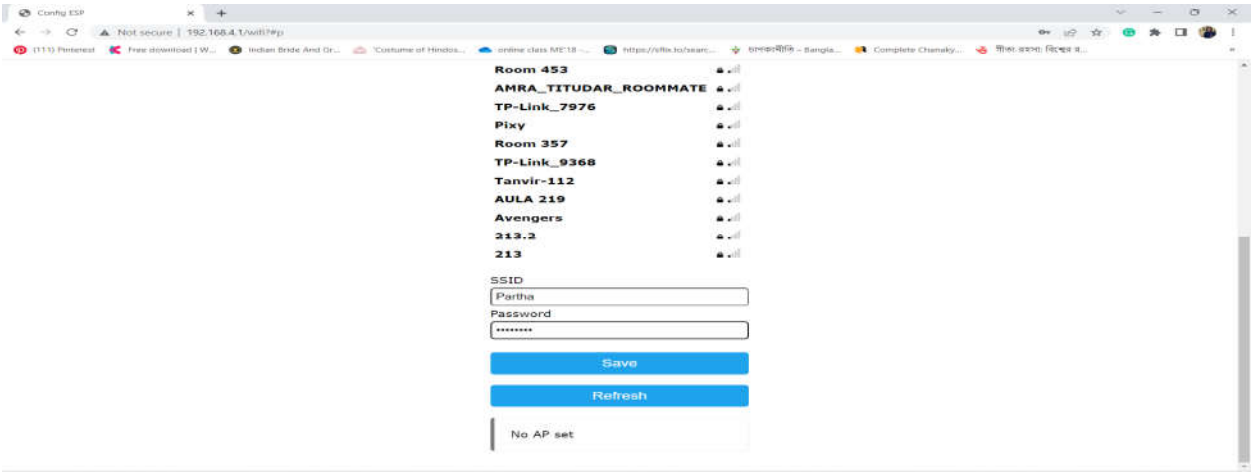
Arduino Code for ESP32 camera module:

The ESP32-CAM is a small size, low power consumption camera module based on ESP32. It comes with an OV2640 camera and provides onboard TF card slot. The ESP32-CAM can be widely used in intelligent IoT applications such as wireless video monitoring, WiFi image upload, QR identification, and so on. The following Arduino commands power up the camera to create its own wifi zone and after few manual steps, the camera can connect itself to any local area network (LAN) to transmit information.

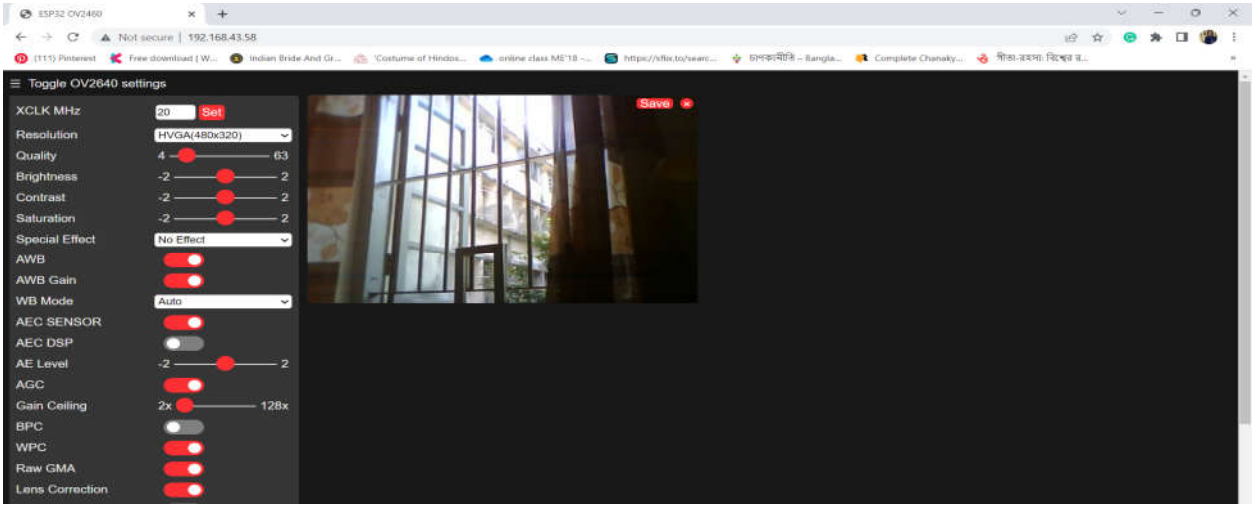




A local network is selected to which the ESP32 module can get connected.

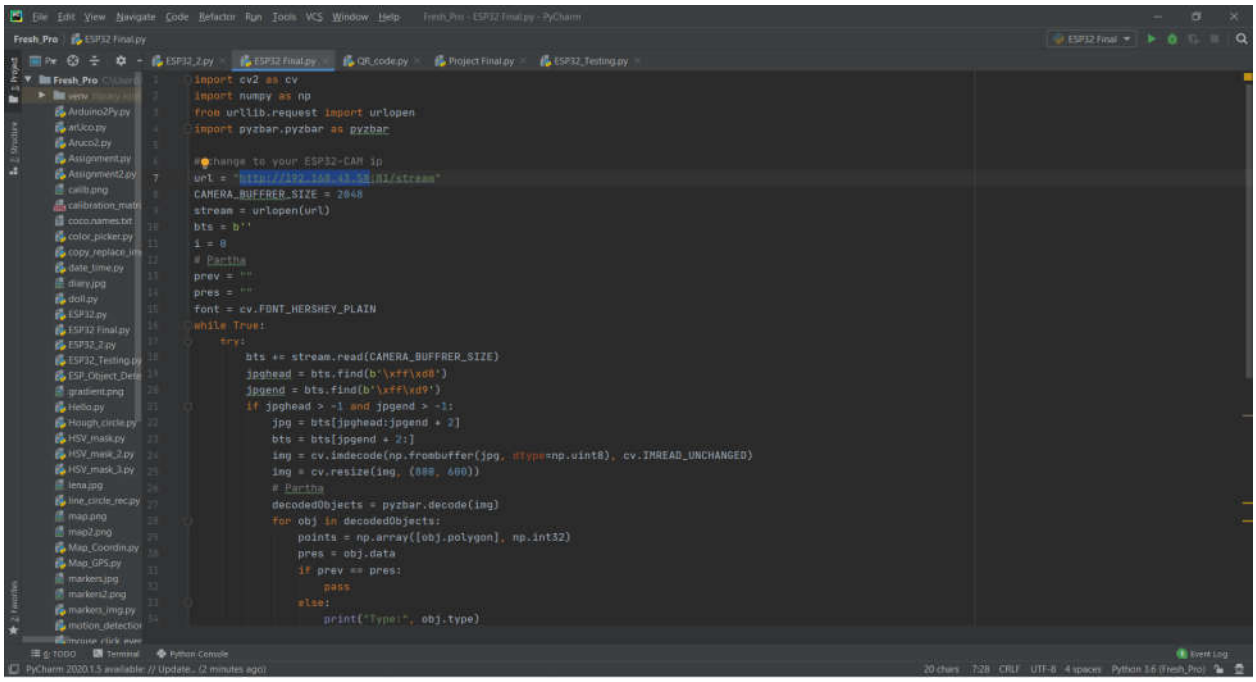


The serial monitor will then return an IP address to access the camera module.



QR Code detection and varification.

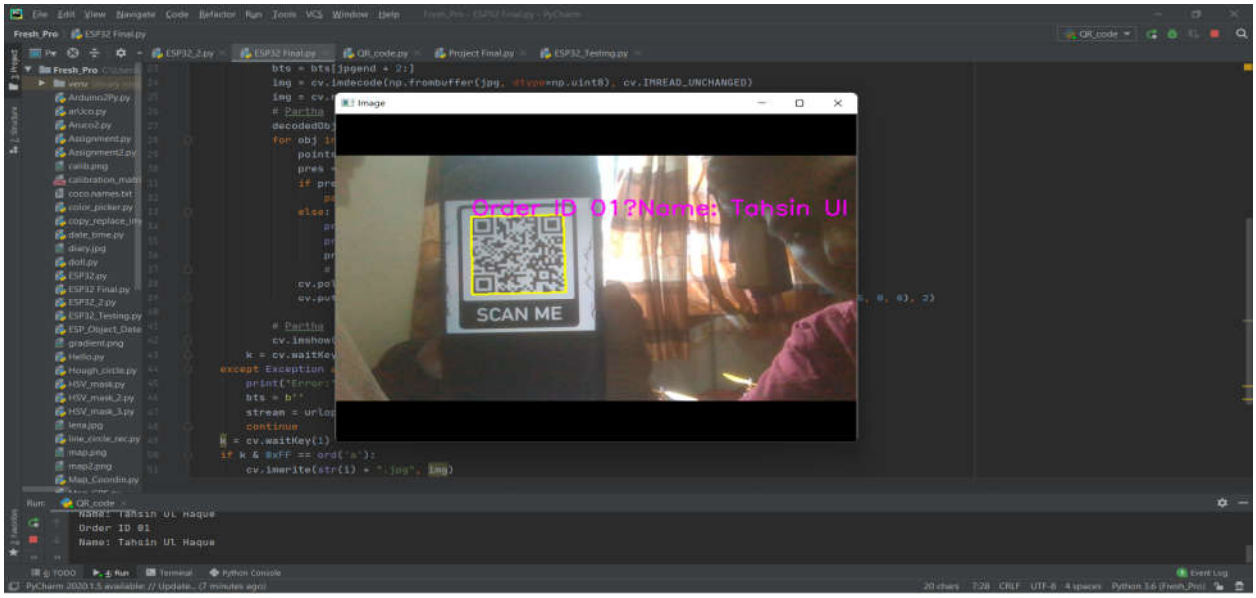
The live camera information is fed into a Python command using OpenCV. OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. If any QR code is visible to the camera, the OpenCV-Python command will detect it. Some kind of action (door opening, green LED blinking etc.) will be executed if the QR code information matches an authorized one.



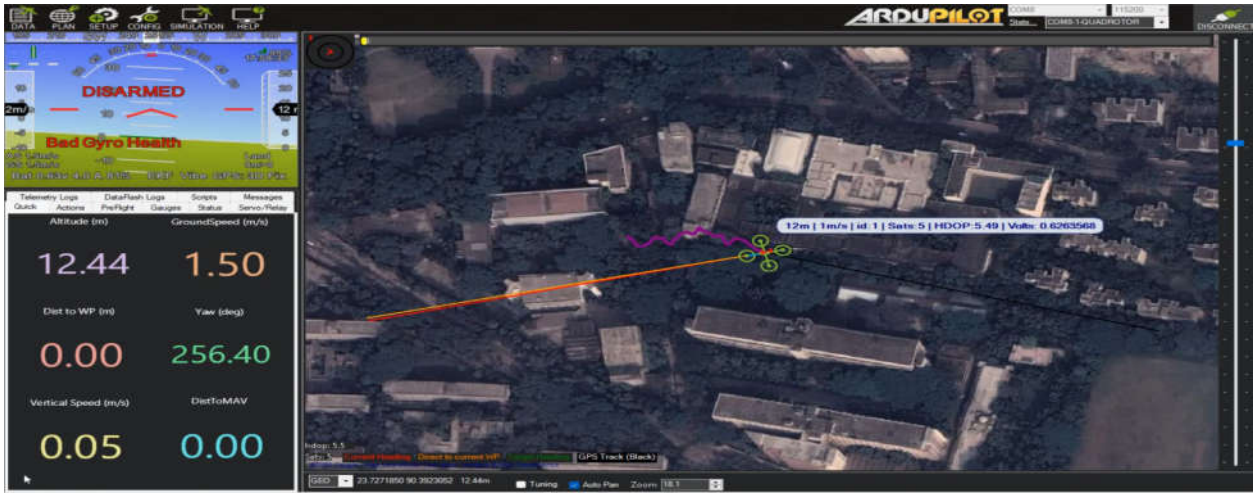
```
import cv2 as cv
import numpy as np
from urllib.request import urlopen
import pyzbar.pyzbar as pyzbar

#change to your ESP32-CAM ip
url = "http://192.168.43.58:81/stream"
CAMERA_BUFFRER_SIZE = 2048
stream = urlopen(url)
bts = b''
i = 0
# Partha
prev = ""
pres = ""
font = cv.FONT_HERSHEY_PLAIN
while True:
    try:
        bts += stream.read(CAMERA_BUFFRER_SIZE)
        jpghead = bts.find(b'\xff\xd8')
        jpgend = bts.find(b'\xff\xd9')
        if jpghead > -1 and jpgend > -1:
            jpg = bts[jpghead:jpgend + 2]
            bts = bts[jpgend + 2:]
            img = cv.imdecode(np.frombuffer(jpg, dtype=np.uint8), cv.IMREAD_UNCHANGED)
            img = cv.resize(img, (800, 600))
            # Partha
            decodedObjects = pyzbar.decode(img)
            for obj in decodedObjects:
                points = np.array([obj.polygon], np.int32)
                pres = obj.data
                if prev == pres:
                    pass
                else:
                    print("Type:", obj.type)
```

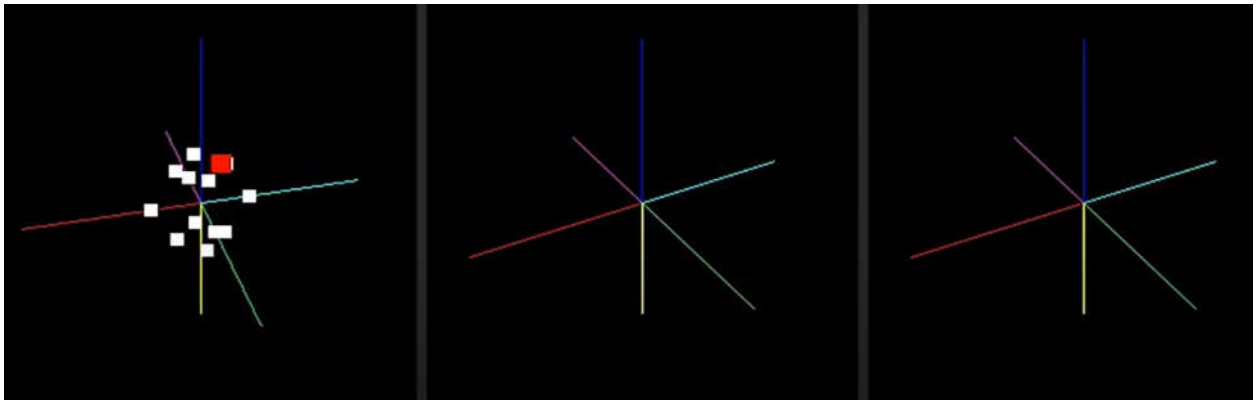
The OpenCV-Python command will extract every information the QR code conveys.

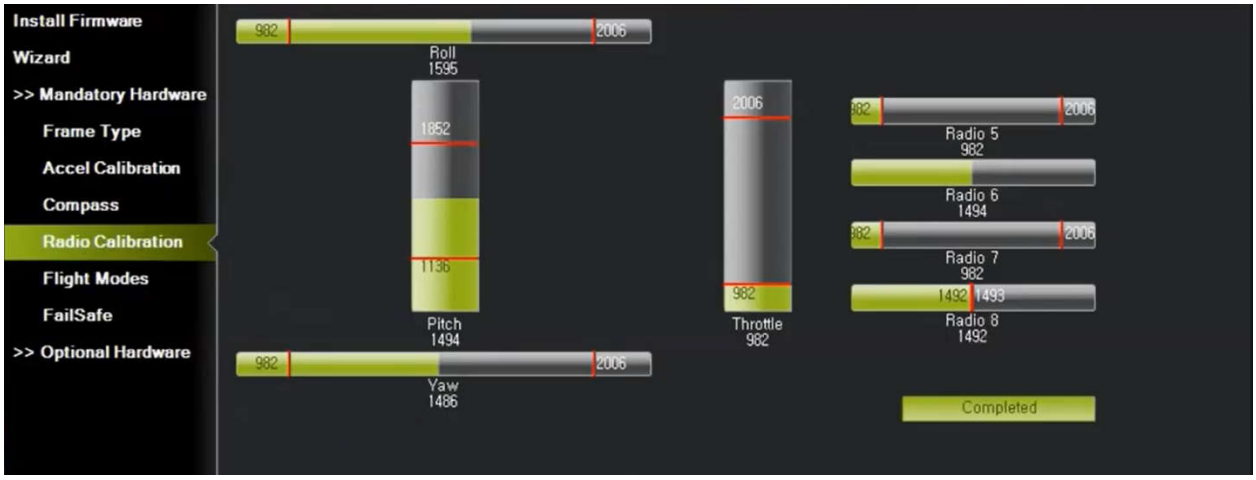


The APM (Flight Controller) settings configurations:

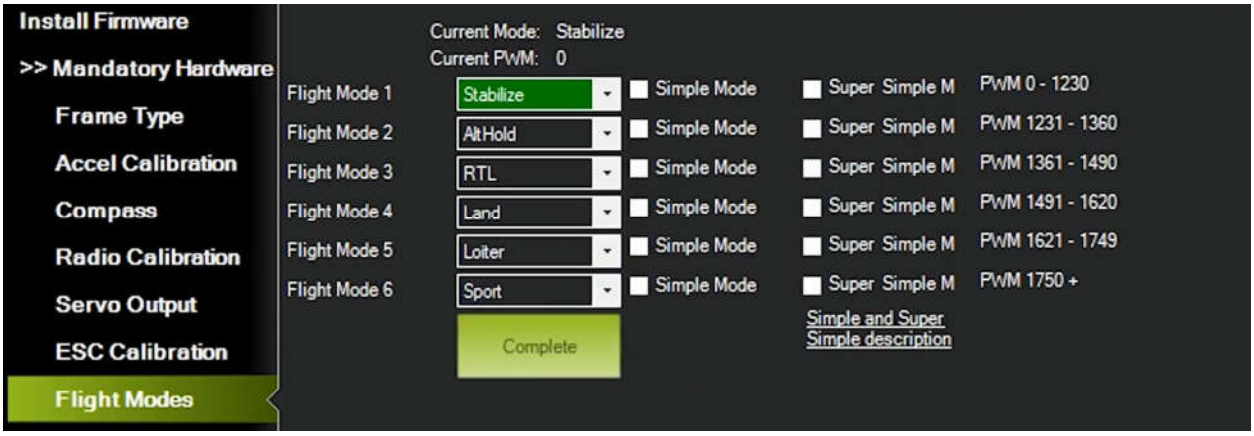


Compass and Radio Calibration:





Flight Modes Settings



Challenges

The main challenge we faced during the project is the shortage of components. Many of our plans had to be reshaped due to the lack of supply of electronic modules in the local market.

Sometimes available but cheap components did not work well and we had to move to their expensive alternatives. This happened to our GPS module and ESP32 module. We tried to use a 6M GPS module but it could not show the location of the drone very well. Therefore, we later used the 8M GPS module. Similarly, the Wi-Fi range, generated by the ESP32 module, was too small for a drone to operate. To extend the range we had to use an additional router module.

Flight controllers of the latest version were not available in Bangladesh. Therefore, it was a big challenge to get the proper firmware for the old version of the flight controller. After a lot of trial and error, we managed to operate the flight controller with the proper firmware.

Uncontrolled flights, sudden malfunctioning of onboard equipment, and drone's dropping down from the sky were big problems. These eventually led us to a broken arm, broken landing gears, and several broken propellers. However, all of those components were replaced with new ones and our final flights were very smooth.

Future Prospects

- Waypoint navigation
- Autonomous flight and landing
- Built in Sim Module
- Face Recognition
- Automated container opening & closing upon verification
- Built in software to collect order and payment electronically

In future with more advancement we can use this drone commercially in hospital sectors or battle field. In hospitals, the drone can move in a predetermined altitude and deliver medicine in wards and cabins by identifying patient records using AI.

References

https://youtu.be/A6hiRMH-sYM
https://youtu.be/IxE4K7ghST0
https://randomnerdtutorials.com/
https://youtu.be/3rU_p4eXPPY
https://youtu.be/A6hiRMH-sYM
https://www.researchgate.net/figure/The-Quadcopter-Reference-Frames_fig4_331298873
https://ieeexplore.ieee.org/document/8533727

Acknowledgement

- Dr. Kazi Arafat Rahman
  Assistant Professor, Department of ME, BUET
- Mr. Mantaka Taimullah
  Lecturer, Department of ME, BUET
- Mr. Shahriar Alam
  Lecturer, Department of ME, BUET
- Mr. Priom Das
  Lecturer (PT), Department of ME, BUET