# FIRST RESPONSE DRONE

**A PROJECT REPORT**

*Submitted by*

**S. PREM RAJ (170701101)**

**S. LOKESH (170701074)**

**SV. KUMARAN (170701071)**

***in partial fulfilment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**IN**

ELECTRONICS AND COMMUNICATION ENGINEERING

**SRI VENKATESWARA COLLEGE OF ENGINEERING**

**(An Autonomous Institution; Affiliated to Anna University, Chennai -600 025)**

**ANNA UNIVERSITY :: CHENNAI 600 025**

**MAY 2021**

# SRI VENKATESWARA COLLEGE OF ENGINEERING

**(An Autonomous Institution; Affiliated to Anna University, Chennai -600 025)**

# ANNA UNIVERSITY, CHENNAI – 600 025

## BONAFIDE CERTIFICATE

Certified that this project report "**FIRST RESPONSE DRONE**" is a bonafide work of "**S. PREM RAJ (170701101), S. LOKESH (170701074), SV. KUMARAN (170701071)**" who carried out the project work under my supervision.

**SIGNATURE**

Dr. S. Muthukumar, M.E., Ph.D.

**HEAD OF THE DEPARTMENT**

Electronics & Communication Engineering

**SIGNATURE**

Dr. S. Ganesh Vaidyanathan, M.E., Ph.D.

**SUPERVISOR**

Professor

Electronics & Communication Engineering

Submitted to the Project viva-voice Examination held on ……………………...

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We thank our supervisor **Prof. Dr. S. GANESH VAIDYANATHAN,** Principal, Sri Venkateswara College of Engineering for his continuous support and guidance in the completion of our project.

We thank **Dr. S. MUTHUKUMAR,** Head of the Department, Department of Electronics and Communication Engineering for his constant support to complete our project.

We wish to thank the project coordinators, **Mrs. K S SUBHASHINI, Mr. S P SIVAGANA SUBRAMANIAN, Dr. S. SUTHA** for encouraging us thought out our project.

We also wish to thank **Dr. S. R. MALATHI and Dr. R . GAYATHRI** for their valuable feedback.

We also extend our sincere thanks to all teaching and non-teaching staff of the Department of Electronics and Communication Engineering for their valuable support.

# ABSTRACT

An emergency/response vehicle is any vehicle that is designated to respond to an emergency in a life-threatening situation. The conventional emergency/response vehicles have been our main preference for all these years, but the main draw-back is the wait time. Studies show that in India the average time for a response vehicle to reach the scene is about 15 minutes. This is mainly because of the extreme population density and the poor traffic management. With UAVs (Unmanned Aerial Vehicles)/drones this delay can be reduced down to a significant amount. Our aim is to aid and support Civil Service Departments such as Medical Department, Fire and Rescue and Police Department. Once the user reports an emergency, our server will pick it up and map the user to the nearest available FRD (First Response Drone). The assigned FRD will autonomously reach the addressed destination. It will have a container carrying medicines in a cold state with a Peltier module and an onboard camera to provide a bird's eye view of the area in case of a police or fire emergency. The drone sends a live feed to the administrator which then is passed through an object detection algorithm to detect human beings.

## TABLE OF CONTENTS

## LIST OF FIGURES

# LIST OF ABBREVIATIONS

| ABBREVIATION | EXPANSION |
|---|---|
| UAV | Unmanned Aerial Vehicle |
| FRD | First Response Drone |
| 3D | 3 Dimensional |
| CAD | Computer-aided design |
| DC | Direct Current |
| RBC | Red Blood Cell |
| BVLOS | Beyond Visual Line of Sight |
| HD | High Definition |
| ARM | Advanced Risk Machines |
| SoC | System on a Chip |
| LPDDR2 | Low-Power Double Data Rate |
| SDRAM | Synchronous Dynamic Random-Access Memory |
| CSI | Camera Serial Interface |
| DSI | Display Serial Interface |
| PLA | Polylactic acid |
| PVC | Polyvinyl chloride |
| ESC | Electronic Speed Controllers |
| LiPo | Lithium-ion Polymer |
| A | Ampere |
| V | Volt |
| VCC | Voltage Common Collector |
| GND | Ground |
| GPS | Global Positioning System |
| CV | Computer Vision |

| | |
|---|---|
| UART | Universal asynchronous Receiver-transmitter |
| GPIO | General-purpose input/output |
| PWM | Pulse Width Modulation, |
| LED | Light-emitting diode |
| HDMI | High-Definition Multimedia Interface |
| USB | Universal Serial Bus |
| PETG | Polyethylene Terephthalate Glycol |
| ABS | Acrylonitrile Butadiene Styrene |
| CAN | Controller Area Network |
| GPU | Graphics Processing Unit |
| UI | User Interface |
| SDK | Software Development Kit |
| WSGI | Web Server Gateway Interface |
| Py | Python |
| SITL | Software in The Loop |
| GUI | Graphical User Interface |
| API | Application programming interface |
| ETA | Estimated time of arrival |

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

## 1.1 UNMANNED AERIAL VEHICLE

Unmanned aerial vehicles (UAVs) are gaining significance these days. Detailed information and techniques of UAVs in smart and modern cities are brought out. Having a human commandeer an aerial vehicle especially during an emergency like accidents can be a very challenging task. By deploying a drone, the task can be carried out almost instantaneously. The drones can be used for a wide range of applications ranging from defence to agriculture.

## 1.2 AUTONOMY OF THE DRONE

A typical unmanned aircraft is made of light composite materials to reduce weight and increase manoeuvrability. The unmanned aerial vehicle is capable of navigating autonomously without any real-time input from the user and also is programmed to follow a specified path. The algorithm enables a control technique by which the drone is facilitated to fly autonomously. The trajectory of the drone, accurate altitude holds performance and its smooth motion can all be monitored autonomously.

## 1.2GENERAL ELEMENTS
## 1.3

Few important elements of our project are

- Raspberry Pi for data collection and processing
- Pixhawk for flight control
- Object Detection
- 3D Printing

### 1.3.1 OBJECT DETECTION

It is necessary to view and analyse the state of the drone's environment in order to give the civil service officials an "birds eye" view of the emergency situation. The process works by passing the video frames through a server and retrieving it in our system and performing object detection with the help of Haarcascade classifiers. The video stream will be in real time and be fed continuously after the drones has reached the addressed coordinates, the camera used to stream is a Raspberry PI Cam V1.3.

### 1.3.2 3D PRINTING

3D printing, or additive manufacturing, is the construction of a three-dimensional object from a CAD model or a digital 3D model. The term "3D printing" can refer to a variety of processes in which material is deposited, joined or solidified under computer control to create a three-dimensional object, with material being added together (such as plastics, liquids or powder grains being fused together), typically layer by layer.

## 1.4 INTEGRAL PARTS

The autonomous hexacopter drone is controlled using Pixhawk flight controller, which is powered by a Lithium Polymer 2200mAh battery. The body of the drone is made of lightweight S550 frame which has a carbon fibre power distribution board which is used to ensure equal distribution of power to all the four brushless DC motors. The Gyro stabilization technology is the most essential component that makes the drone to maintain a smooth flight even through strong winds and gusts. The Raspberry Pi camera v1.3 is used for object detection.

### 1.5 NEED FOR DRONE

The main purpose of the drone is to minimize response time and to support civil service departments. During emergencies like fire accidents or robbery, the response time of civil service departments like Medical, Police and fire department haven’t been promising in INDIA. Therefore, to provide aid before the arrival of civil service departments drones can be deployed. These drones can be quick and efficient when navigating through rough conditions and unpredictable terrain.

# CHAPTER 2

# LITERATURE SURVEY

**Piotr Kardasz, Jacek Doskocz, Mateusz Hejduk, Paweł Wiejkut and Hubert Zarzycki, "Drones and Possibilities of Their Using "** This paper proposes the different methodologies used for the construction of drones and their elements such as frame, propellers, engine, system of power, electronic control and communication system. It also explains about how drones can be used for public services like police, fire brigades and border guards, delivery, by army, industry and filming. The paper also discusses about the danger of using unmanned aerial vehicles, which is due to the limited power supply, damage caused by weather conditions, hitting an obstacle, discharge from batteries if there is any fault, danger of drone falling from a height due to battery failure. The paper overall discusses about the construction and risks in practicing an autonomous drone in civilian sectors

**Anna Konert, Jacek Smereka, and Lukasz Szarpak, "The Use of Drones in Emergency Medicine: Practical and Legal Aspects"** This paper discusses about how air transport is widely used in military and civil emergency medicine owing to speed of action, lack of restrictions characteristic of ground vehicles and the ability to reach distant, otherwise inaccessible places. The paper talks about how drones carry loads at several hundred meters above ground. It also explains about the survey made with blood transport to urban areas with drones and how there is no loss in platelet counts, RBC count and temperature. It explains about how drones can be used in Beyond Visual Line of Sight (BVLOS) and discusses about how various countries are implementing tasks with drones.

**Kyle Stelmack,"Weaponized Police Drones and Their Effect on Police Use of Force"** Here they discuss about how unlikely an actual drone with lethal weaponry will rise, and how non-lethal weapons systems being attached to police drones is not out of the realm of possibility. Weaponry like fire tasers, beanbags also known as "Stun Batons" are suggested to be used. They also discuss about weaponry designs, which are similar to paint ball barrels which are capable of firing 20- 30 balls per second, and how these balls are to be replaced or filled with pepper spray, coloured dye or any solid plastic bullets and how the drone should also be equipped with blinding lasers. They also have stated about 'Skunk Riot Control Copter' which can fire up to 80 balls per second. They also have discussed about having a thermal camera, a full HD colour camera and a telemetry for long distance functions, including a real-time video link. They also strongly suggest that using drones would be of a great advantage to the department, due to the response time and limitations of human interaction towards the situation and how drones will move beyond strictly surveillance uses and move to intervention in situations on the ground.

# CHAPTER 3

# PROPOSED SYSTEM

## 3.1 PROPOSED SYSTEM ARCHITECTURE

The below diagram depicts the system architecture.

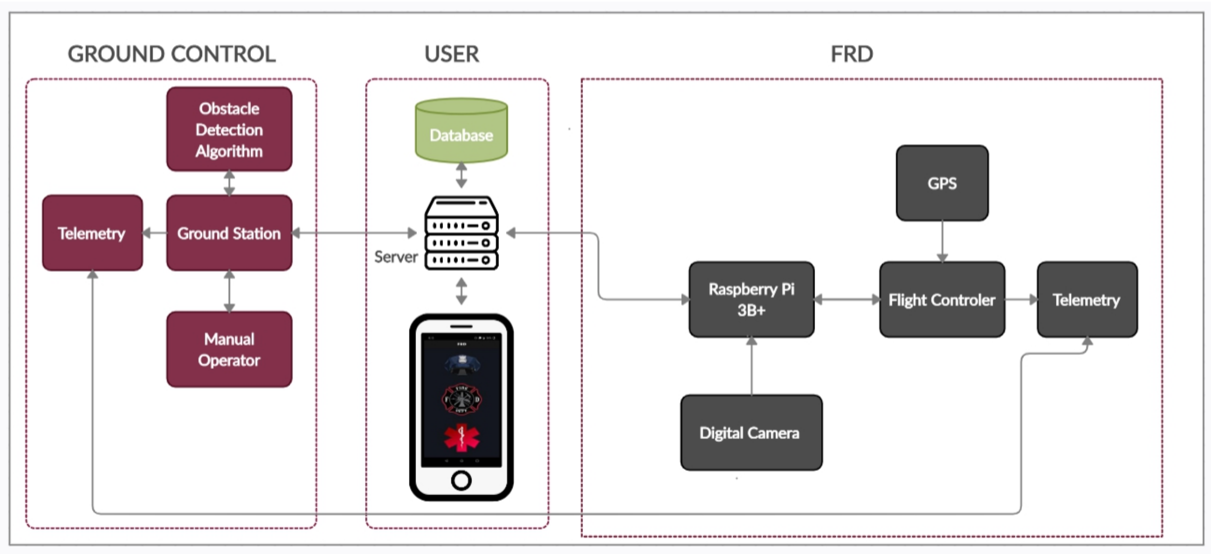


*Fig 3.1 Proposed system architecture*

The drone acts as instant response medium for medical, police and fire emergency. The raspberry pi is used to send commands to Pixhawk for autonomous flight, and also uses a raspberry pi cam v1.3 for capturing frames and streaming it to the admin.

Features of the raspberry pi 3b+ microprocessor

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading the operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

Features of Pixhawk flight controller

- Main SoC STM32F427 - 180MHz ARM® Cortex® M4 with single-precision FPU 256 KB SRAM
- Failsafe SoC STM32F100 24MHZ ARM Cortex M3 8KB SRAM
- ST Micro L3GD20H 16-bit gyroscope

- ST Micro LSM303D 14-bit accelerometer / magnetometer
- Invensense MPU 6000 3-axis accelerometer/gyroscope
- MEAS MS5611 barometer
- 5x UART (serial ports), one high-power capable, 2x with HW flow control
- 2x CAN (one with internal 3.3V transceiver, one on expansion connector)
- PPM sum signal input
- Spektrum DSM / DSM2 / DSM-X® Satellite compatible input
- I2C, SPI

# CHAPTER 4

# AUTONOMOUS DRONE

## 4.1 INTRODUCTION

Unmanned aerial vehicles are a relatively new technology that has spread its applications recently. In this chapter, the development of these vehicles and their implementation are presented. Unmanned aerial vehicles can perform repetitive tasks with high precision and accuracy whatever may be the condition be even in complete darkness. With their small size as an advantage, the practical applications of these vehicles are vast compared to other aerial vehicles.

## 4.2 WHY DRONES?

Drones have a wide range of applications that have led industries to invest in this technology. They have proven to be safe to the environment and also to be a cost saving technology at the same time. They are also flexible for quick inspections. Drone markets are considered to be a great commercial opportunity, since the market is expected to grow in the following years.

## 4.3 HARDWARE

A multicopter, more commonly known as a drone, is a helicopter with more than two propellers. Some of the different types of multicopters are quadcopter, hexacopter and an octocopter. Hexacopter type has been selected for FRD. It has been made as light weight as possible and as durable as possible. The materials used are aluminium, PLA, PVC and carbon fibre.

### 4.3.1 MOTORS

The most widely used motor for small multicopters is the Brushless DC motor, it differs from the conventional DC motors as it doesn't contain brushes. It works by the electromotive forces generated by the permanent magnets located in the rotor and the coil wound poles in the stators. When current passes through the coils it induces a magnetic field, the current direction determines whether it's an attractive force or a repulsive force. The advantage of this motor is that it doesn't produce magnetic interferences. Their small size, low cost and the above-mentioned properties have led us to use this motor for our hexacopter. The hexacopter uses 6 1000KV BLDC motors.

### 4.3.2 BLADES

One of the most important parts of the drone are the blades. These spinning blades are the wings to the craft, the very part that creates the airflow that lifts the machine into the air. Drone propellers come in many different shapes and sizes – they all serve the same overall purpose. The blade material used in the hexacopter drone is plastic, it has been selected for its durability, light weight and cost.

### 4.3.3 ELECTRONIC SPEED CONTROLLERS

The main aim of Electronic Speed Controllers (ESC) is for controlling the velocity of the motor according to the PWM (Pulse Width Modulation) inputs that they are receiving, it consists of switches, which energises two out of the 3 phases which are coming from the motor at a time, and switches phases accordingly with the Back EMF from the motors or Hall Effect which enables the ESC to know the precise location of the permanent magnet in the rotor.

### 4.3.4 BATTERY

Multi-copters commonly use LiPo battery as an energy source. The main reasons for this tendency are their lightweight, the large variety of shapes that they can adopt, their great capacities in comparison with their small size and their high discharging rates. Different capacities can be chosen for this type of batteries. Capacity is an indication of how much power can be stored by the battery, and it is measured in mAh (milliamperes per hour). For ensuring flight missions of 15 minutes, LiPo batteries of 2200mAh have been used with maximum current rating and voltage rating of 30A and 7.4V.

### 4.3.5 RADIO CONTROLLER

It's a device that connects the pilot with the aircraft and it is recommendable to use when the drone reaches the particular location where there is a fire emergency and the officials can control it once it arrives to the location to search the whole area to get proper insight; it can also be used to recover the hexa-copter control in case it's needed. It has a minimum of four channels and it transmits at frequencies in the range of 2.5GHz. The radio control has two sticks, one on the left which helps with the arming and controls the throttle or vertical acceleration and the yaw and the right one controls the roll and the pitch.

### 4.3.6 TELEMETRY AND GPS

The function of the telemetry is to provide a channel to transfer flight data at real time. The GPS is used to estimate the global position of the drone by measuring the relative positions with respect to several satellites. The GPS selected for this project includes an inbuild compass.

## 4.4 HEXACOPTER MODEL

The mathematical model for the hexacopter is developed in this section. The hexacopter is considered a rigid body, which consists of the main airframe and four arms. This, the links between the arms and the airframe are assumed to be ideal constraints, it means, there is no dissipation on them. The arms meet at the centre at right angles. The centre of gravity of the hexacopter is located at the centre of the airframe.

### 4.4.1 THRUST

As the propulsion element for the hexacopter, the motor is the most important component. However, before determining the capacity of the motor applied in the design, it is necessary to know the total weight that will be lifted and the thrust which is required to lift the heavy load of hexacopter. The calculation to determine the thrust per motor as in the following equation:

$$Thrust = \frac{total\,weight\ x\ 2}{number\,of\ \ motor}$$

### 4.4.2 PROPELLER PERFORMANCE

The distance streamed by the fluid due to one rotation of propeller's blade is defined as the pitch parameter that is often noticed in the selection of propellers. Thus, if the pitch and diameter of the propeller are larger, so the motor rotation will be slower and the lifting force that is produced is large. So, if the hexacopter can lift the heavy load, it required large diameter propellers and large pitch. The thrust style equation of this propeller is

$$F_{TH} = \rho C_t n^2 D^4$$

Where ɋ is the air density, n is the rotational speed of the propeller, Ct is the thrust propeller coefficient, and D in meters is the diameter of the propeller. For each speed, this Ct value varies with a small value so it can be ignored. While the power that is generated from the propeller can be calculated

$$P_p = \rho C_p n^3 D^5$$

Cp is the power coefficient of the propeller that is obtained from the rotation. This Cp value changes with speed. For the torque on the propeller is generated based on the following equation:

$$T_q = \frac{P_p}{\omega}$$

Where ω is the propeller's angular speed.

### 4.4.3 CAPACITY OF THE BATTERY

Batteries are the power source to run all the components on the hexacopter. The battery also affects the flight time so that proper calculations are required to produce optimal results. Therefore, to get the proper power and load combinations, the batteries that are used must have more current than motor currents. The parameter to be considered in the selection of the battery is the number of cells, discharge, and capacity. The number of cells determines the voltage of the battery in an empty state. Then the discharge shows how much current rating / current velocity can be released, and the capacity shows how long the battery can work on certain amperes.

### 4.4.4 FORCE MODELS

Actions affecting the hexacopter include its own weight and the aerodynamic forces produced by its blades. These aerodynamic forces and moments must be modelled using the Momentum Theory

$$M_j = \frac{1}{2} C_D \rho A \Omega^2$$

$$T_j = C_T \rho \pi R^4 \Omega^2$$

ρ refers to the air density, Ω is the angular velocity of the blade, R the radius of the blade, A the cross-section of the blade, Tj the thrust and Mj the torque produced by the motors. Once the thrust model is established, it can be a model all the forces acting on the hexacopter.

### 4.4.5 MASS AND MOMENT OF INTERTIA

The mass moment of inertia of an object (J) plays a similar role in the rotational motion to the role that mass plays in translational motion: the mass moment of inertia determines how the rotational velocity is affected by the applied torque. This of course depends not only on the mass of the object but also on how the mass is distributed around the rotation axis. It is important to note here that the hexacopter is assumed to be perfectly symmetric about the x y and z axes and to have its centre of mass at the geometric centre of the arms. With these assumptions, the matrix $J_b$ becomes a diagonal matrix (note that this is related to our choice of the x- and y-axis positions). The $J_{xx}$ and $J_{yy}$ terms are also taken to be identical owing to this symmetry.

$$J_b = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix}$$

| COMPONENTS | QUANTITY | MASS (Q) | TOTAL MASS (g) |
|---|---|---|---|
| Upper frame plate | 1 | 63.2 | 63.2 |
| Distribution board | 1 | 73.8 | 73.8 |
| Arms | 6 | 51.2 | 307.2 |
| Pixhawk | 1 | 38 | 38 |
| Motors | 6 | 33.8 | 202.8 |
| ESCs | 6 | 24 | 144 |
| Telemetry | 1 | 9.3 | 9.3 |
| Propellers | 6 | 8 | 48 |
| GPS | 1 | 23 | 23 |
| PI Camera | 1 | 3 | 3 |

| | | | |
|---|---|---|---|
| Landing gear | 6 | 138.4 | 830.4 |
| Medicine Container | 1 | 150 | 100 |
| Peltier Module | 2 | 11.43 | 22.86 |
| Raspberry Pi | 1 | 39.1 | 39.1 |
| LiPo Battery | 2 | 53 | 106 |

*Table 4.1 Mass of components*

Then, adding the mass of all the above-mentioned items:

Final mass: 2.01kg

### 4.4.6 EQUATIONS OF MOTION

The equations of motion, according to Newton-Euler Formalism, are described with the following expression:

$$\begin{bmatrix} mJ & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{v}_B \\ \dot{\omega}_B \end{bmatrix} + \begin{bmatrix} \omega_B \wedge m v_B \\ \omega_B \wedge I\omega \end{bmatrix} = \begin{bmatrix} F_B \\ M_B \end{bmatrix}$$

Time derivatives must be done in the inertial Earth reference frame. Nonetheless, since vectors v and ω are projected in the body reference frame, Coriolis Theorem must be used, which has been taken into account in the second matrix of the equation. Finally, after a series of algebraic operations that can be checked in, this set of equations of motion is achieved.

$$m\ddot{x} = (sin\psi sin\phi + cos\psi sin\theta cos\phi)\sum_{i=1}^{4} T_i$$

$$m\ddot{y} = (-cos\psi sin\phi + sin\psi sin\theta cos\phi)\sum_{i=1}^{4} T_i$$

$$m\ddot{z} = mg - cos\psi cos\phi \sum_{i=1}^{4}(T_i)$$

$$I_{xx}\ddot{\phi} = \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) + J_r\dot{\theta}\Omega_r + l(-T_2 + T_4)$$

$$I_{yy}\ddot{\theta} = \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) + J_r\dot{\phi}\Omega_r + l(T_1 - T_3)$$

$$I_{zz}\ddot{\psi} = \dot{\phi}\dot{\theta}(I_{xx} - I_{yy}) + J_r\Omega_r + (-1)^i\sum_{i=1}^{4} Q_i$$

## 4.5 HEXACOPTER ASSEMBLY

The assembly process of the hexacopter model is going to be discussed, its divided into two sections; the hardware section and the wiring section.

The components used in the hexacopter are the following

- 6 x Hexa-Frame Arm
- Hexa distribution board
- Pixhawk 3DR flight controller
- Raspberry PI 3B+
- GPS receiver
- 3 x Clockwise propellers
- 3 x Counter-clockwise propellers
- 3 x Clockwise brushless dc motors
- 3 x Counter-clockwise brushless dc motors
- 1 x Battery eliminator circuit
- 30 x M2.5*6 screws
- 24 x M3*10
- Telemetry
- Camera

- Double-side tape
- PVC with PLA joints

### 4.5.1 ESC AND BATTERY SOLDERING

Firstly, the ESCs should be tested whether they are working properly, to do this, supply power to each of them and measure the voltage at the other end. ESCs should then be soldered to the lower part of the power distribution board. Soldering iron and lead are used for this. The red cable from the ESC should be soldered to the positive (+) ones and black cables should be soldered to the negative (-) ones in the power distribution board. Do this process for all the 6 ESCs. ESC to distribution board soldering connection can be seen in Fig 4.1. After successfully soldering, it is advised to use a hot glue gun to cover the soldering part.

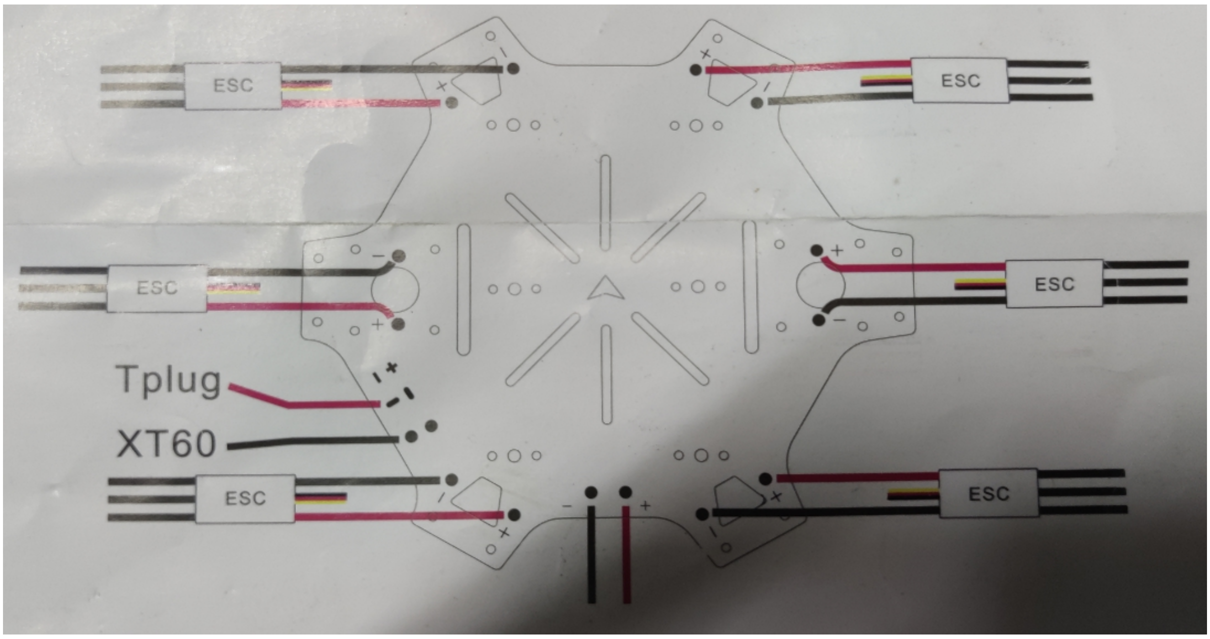


*Fig 4.1 ESC and Distribution board soldering*

### 4.5.2 HEXACOPTER FRAME ASSEMBLY

The 6 arms must be attached now to the bottom frame. Four of M2. 5*6 mounting screws must be screwed for each arm using an Allen wrench. Make sure the ESC's wires are taken through the gap between the distribution board and the arms. After all the 6 arms and the distribution board are attached it'll look like the figure shown below.

*Fig: 4.2 Graphical representation of arms and distribution board*

#### 4.5.2.1 LANDING GEAR ASSEMBLY

Landing gear plays an important role in elevating the drone above ground level while landing and also provides space for carrying loads. Six landing gears have been designed for 6 propellers. The landing gear's main tube is a 30cm tall PVC and the top and bottom connectors are 3D printed with PLA (Polylactic Acid). The top and bottom connectors are connected to the PVC and the top connector is screwed exactly under the motors with M3*10 screws. After connecting all 6 landing gears, the hexacopter will look like the below figure Fig 4.3.

*Fig: 4.3 Hexacopter with landing gear*

### 4.5.3 MOTOR ASSEMBLY

The six motors must be assembled at extreme of the arms using the M2. 5*6 mounting screws. When placing each motor, it has to be taken into account that the motor rotation criterion which was established after developing theoretically the hexacopter model, due to the fact that the spinning of the motor vehicle around itself is to be avoided. Thus, the motors with the same rotation direction (either clockwise or counter-clockwise) must be located in opposite arms. There are two types of motor configuration, one is Hexa X and another is Hexa +, the FRD is of Hexa x type, which has its nose between two arms as shown in the below figure Fig 4.4.

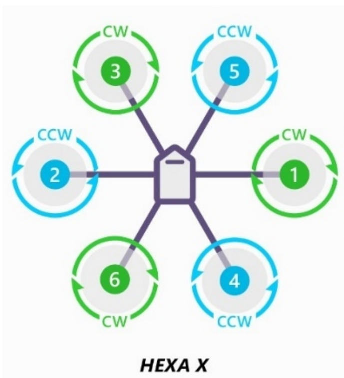


*Fig 4.4: HEXA X type configuration*

Once the motors are mounted, the same criteria should be followed for the blades, the ones motors which spin clockwise must be screwed with clockwise blades and the motors which spin anticlockwise should be screwed with counter-clockwise blades. After this make sure to zip tie the ESC's to their respective arms, this prevents them from shaking freely during flight. Once all the connections are done, the hexacopter will look like the below figure.

*Fig 4.5 Hexacopter with motors and propellers connected*

### 4.5.4 PIXHAWK, TELEMETRY AND GPS ASSEMBLY

The Pixhawk flight controller will be placed on an anti-vibration stand, to reduce vibration produced by the multicopter, the GPS module is connected to the GPS port in the flight controller and the telemetry will be connected to the telemetry port of the flight controller. After assembly it is recommended to check that the structure is rigid enough.

### 4.5.5 WIRING AND CONNECTIONS

For the hexacopter to be ready for flight, the only remaining final step is proceeding with wiring and the connections

#### 4.5.5.1 ESC CONNECTION

The ESCs have to be connected to the motors and the flight controller, each motor has to be connected to the respective esc and remember the rotation of motors, it should be the same as discussed before in Fig 4.3. The esc to motor connection can be seen in the shown Fig 4.6

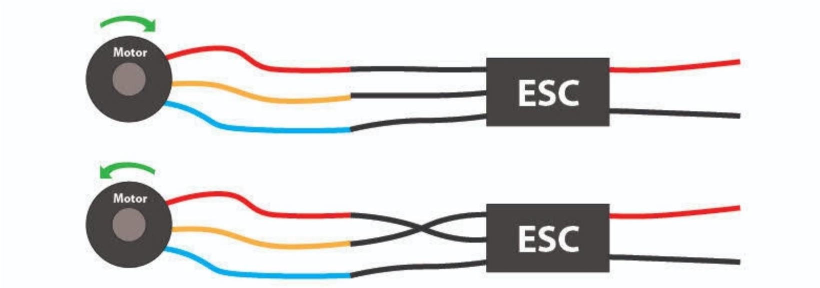


*Fig 4.6 ESC to motor connection*

The ESCs have to be then connected to the flight controller, in the other side of the ESCs, combo of 3 cables will be there, they are VCC, GND and Signal cables. They have to be connected at appropriate channels to the Pixhawk flight controller. The channels are shown in the below figure Fig 4.7.


*Fig 4.7 Pixhawk Output channel pins*

#### 4.5.5.2 PIXHAWK AND RASPBERRY PI CONNECTION

Raspberry Pi 3B+ microprocessor is programmed in python to retrieve the coordinates of the emergency situation and the nearby FRDs coordinates and map the nearest FRD. Then the microprocessor sends flight instructions to the flight controller with the help of DroneKit library, this transfer is done through UART communication. GPIO 14 and GPIO 15 pins of raspberry pi are UART_TX and UART_RX pins respectively, these pins are connected to the tele2 pins in the Pixhawk flight controller along with VCC and GND.

#### 4.5.5.3 PELTIER MODULE CONNECTION

Peltier module follows Peltier effect, the cooling of one junction and heating of other when the electric current is maintained in a circuit. The primary advantages of a Peltier cooler compared to a vapor-compression refrigerator are its lack of moving parts or circulating liquid, very long life, invulnerability to leaks, small size, and flexible shape. The Peltier module is placed inside the medicine container and is supplied through a separate battery to keep the medicines inside in a cold state. The module has two cables VCC and GND which are connected accordingly to the battery.

### 4.6 CALIBRATION

The calibration of the software of all the quadcopter components is necessary before the flight. Furthermore, the calibrated results parameters will be used also for the later simulation of the quadcopter flights. Mission Planner has been selected as the software for doing the calibration. It is going to be explained in detail the complete calibration process. Firstly, the APM board must be connected to the computer via USB. Then, the APM must be linked to Mission Planner. After initializing Mission Planner, it is done by pressing the Connect button on the top right of the screen. For the APM to be recognized, it must be chosen before pressing the button the communication port. This can be done automatically by Mission Planner if the AUTO option is selected, or by clicking on the COM4 option and setting up the data rate to 115200. It is important to remark that Disconnect button must be used before unplug the APM. Once the APM is connected, it has to be gone to Initial Setup Wizard

### 4.6.1 FRAME TYPE CONFIGURATION

The frame of the hexacopter and the type should be selected. The Mission Planner screen can be seen in Fig 4.8

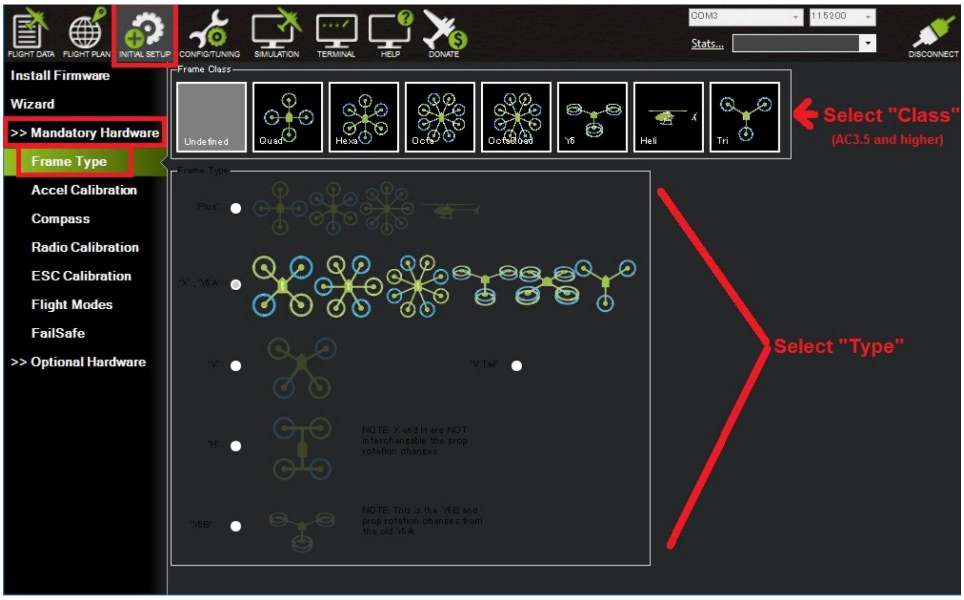


*Fig 4.8 Frame type configuration*

### 4.6.2 ACCELEROMETER CALIBRATION

- Calibrate Accelerometer: The autopilot will have to be placed on each edge: level, on right side, on left side, nose down, nose up and on its back.
- Calibrate level It requires placing the autopilot horizontal. After finishing these steps, the accelerometer setting will be saved automatically. It is important to ensure that the autopilot is kept still just after pressing the key for each accelerometer calibration step.

### 4.6.3 RADIO CALIBRATION

The radio controller transmitter and receiver must be calibrated now. It is needed to teach the autopilot to work with it. Select Calibrate Radio button to start the calibration process. Then, both sticks must be moved in the largest possible circle in order to be attained their complete range of motion. The range of set values will be pointed in the screen with a red line. Same procedure must be followed with channel 5 and 6 toggle sticks. Press end calibration to save the calibrated parameters. Note that if the screen bars are moving in the opposite direction to the radio control orders, it must be selected the reverse channel option on the radio transmitter


*Fig 4.9 Radio control calibration*

### 4.6.4 ESC CALIBRATION

The minimum and maximum PWM values of the ESCs for maximum throttle radio controller commands. Before starting the ESCs calibration, the autopilot must be disconnected from the computer and the rotor blades must be disassembled. Then, it must be followed the next steps. Turn on the Radio Controller and put the throttle stick at maximum.30 Connect the LiPo battery. The autopilot's red, blue and yellow LEDs should light up in a cyclical pattern. If this condition is succeeding, the APM will start in ESC calibration mode the next time the battery is plugged in. Keeping the throttle stick high, disconnect and connect the reconnect the battery. The ESC calibration should start. Wait for the ESCs to emit the musical tone. It depends on the battery's cell count, being that number the regular number of beeps that must be emitted (i.e., 3 for 3S, 4 for 4S). Then, additional two beeps to indicate that the maximum throttle has been captured. Pull the throttle stick down to its minimum position The ESCs will emit a long tone indicating that the minimum throttle has been captured. Then, the calibration is complete. Check now that the motor spins by raising the throttle a bit and then lowering it again. Set the throttle to minimum and disconnect the battery. The APM will exit the ESC calibration mode.

## 4.7 LIVE STREAMING AND OBJECT DETECTION

### 4.7.1 LIVE STREAMING

Once the user interacts with the mobile app, selects the type of emergency and reports one, the request will be placed in the database and the server will catch and assign a drone to the location. The drone will instantly pick it up and will navigate to the addressed coordinates. Once reached, the raspberry pi camera will pick up the frames and send a post request to the server, this happens in real time.

### 4.7.2 OBJECT DETECTION

The administrator application will be using a get request in a loop to get these frames. These frames are nothing but images, these images are then passed through a Haarcascade Frontal Face classifier, Object Detection using Haar feature-based cascade classifiers is an effective object detection method. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

This process is achieved with the help of OpenCV library both in the Raspberry Pi as well as in the server backend. The OpenCV library is used to pick up the frames from the camera and sends a post request to the server, the administrator then pings the server to get these images and is converted to grayscale. The new converted gray image is then passed through the Haar classifier, the classifier then returns four-pixel coordinates as a result, these coordinates will enable the algorithm to plot a box over the coloured frame to indicate the object has been detected visually. This happens in a loop, so the administrator will be able to identify the person in the frame with minimal effort. This will be efficient method during fire and police emergencies

# CHAPTER 5

# HARDWARE AND SOFTWARE REQUIREMENTS

## 5.1 HARDWARE

### 5.1.1 RASPBERRY PI 3B+

Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom. The Raspberry Pi project originally leaned towards the promotion of teaching basic computer science in schools and in developing countries. The original model became more popular than anticipated, selling outside its target market for uses such as robotics. It is widely used in many areas, such as for weather monitoring, because of its low cost, modularity, and open design. It is typically used by computer and electronic hobbyists, due to its adoption of HDMI and USB devices.

The final revision of our third-generation single-board computer

1.4GHz 64-bit quad-core processor, dual-band wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and Power-over-Ethernet support (with separate PoE HAT)

The Raspberry Pi 3 Model B+ is the final revision in the Raspberry Pi 3 range.

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading the operating system and storing data
- 5V/2.5A DC power input

- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

Raspberry Pi 3B+ model and its 40 pin GPIO pinout can be seen with Fig: 5.1 and Fig 5.2 respectively

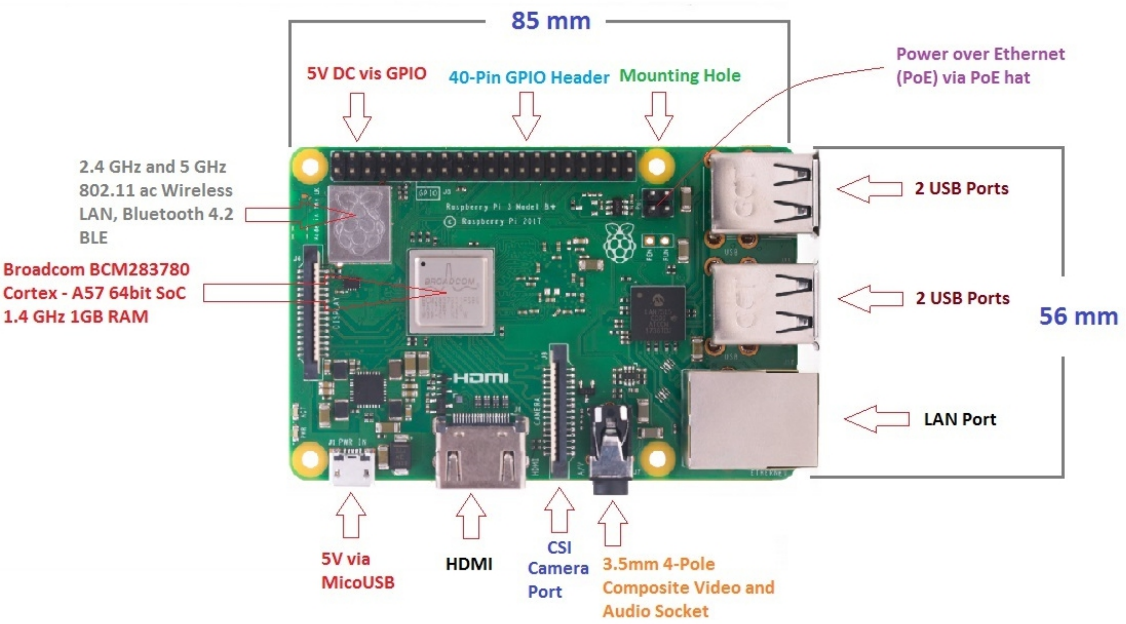


*Fig: 5.1 Description of Raspberry Pi 3B+ Kit*

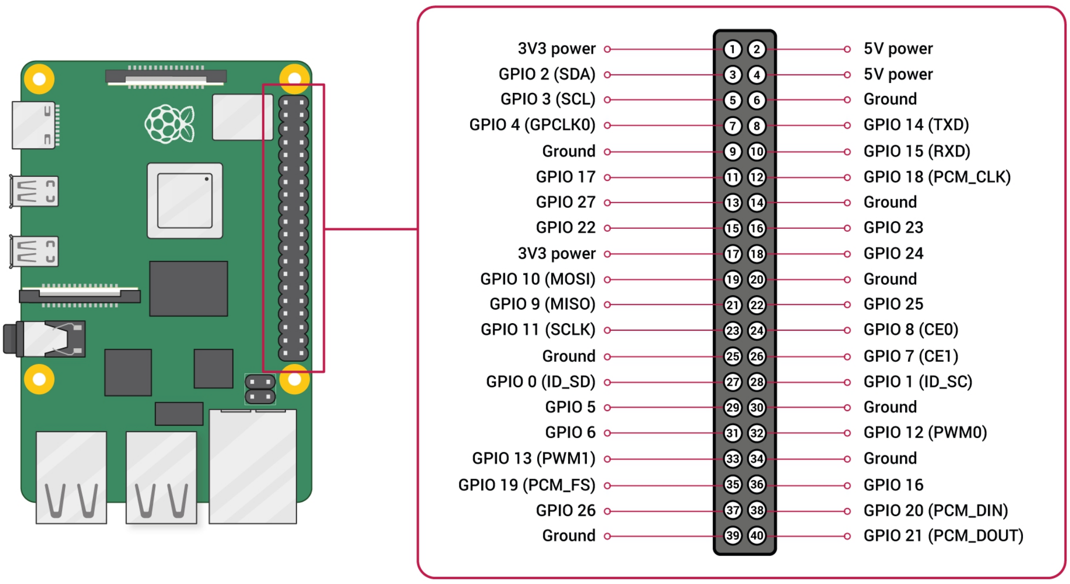

*Fig: 5.2 Raspberry pi 3B+ GPIO connector pins*

#### 5.1.1.1 PRODUCT DESCRIPTION

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading the operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)
- Memory Card: 16Gb Micro Sd Card

### 5.1.2 PIXHAWK 3DR

Pixhawk autopilot is an open-source autopilot system oriented toward inexpensive autonomous aircraft. Low cost and availability enable hobbyist use in small remotely piloted aircraft. PX4 is the *Professional Autopilot*. Developed by world-class developers from industry and academia, and supported by an active worldwide community, it powers all kinds of vehicles from racing and cargo drones through to ground vehicles and submersibles.

### 5.1.2.1 PRODUCT DESCRIPTION

- Main SoC STM32F427 - 180MHz ARM® Cortex® M4 with single-precision FPU 256 KB SRAM
- Failsafe SoC STM32F100 24MHZ ARM Cortex M3 8KB SRAM
- ST Micro L3GD20H 16-bit gyroscope
- ST Micro LSM303D 14-bit accelerometer / magnetometer
- Invensense MPU 6000 3-axis accelerometer/gyroscope
- MEAS MS5611 barometer
- 5x UART (serial ports), one high-power capable, 2x with HW flow control
- 2x CAN (one with internal 3.3V transceiver, one on expansion connector)
- PPM sum signal input
- Spektrum DSM / DSM2 / DSM-X® Satellite compatible input
- I2C, SPI

### 5.1.3 CREALITY ENDER 3

Creality Ender 3 is an extraordinary 3D printer, it has a build volume of 220 x 220 x 250mm, a BuildTak-like heated build plate, power recovery mode and a tight filament pathway that makes it easier to print with flexible materials. IT works with PLA, PETG, ABS and other exotic filaments. Creality also offers an "Ender 3 Pro", which has a detachable magnetic heated bed and improvements to the Y-axis to achieve a better print quality.

*Fig 5.3 Creality Ender 3 3D printer*

### 5.1.3.1 ASSEMBLY

Step 1: Put the frame together with the pair of 20×40 aluminium extrusion pieces, which will be fastened to the frame using two screws on each side. There are pre-drilled holes in the base to mount the Creality Ender 3's aluminium extrusions.

Step 2: Attach the power supply box and LCD screen to the base.

Step 3: Install the Z-Limit switch and Z motor to the left side of the frame, The Z-Limit switch must be mounted to the side of the vertical beam precisely 32mm from the bottom of the Creality Ender 3 base.

Step 4: The Z-axis Motor is attached directly to a pre-installed bracket on the back of the left vertical beam.

Step 5: The next step is the X-axis assembly. Starting at the open side of the X-axis beam, slide the gantry into place by positioning the wheels at the top and bottom. The wheels should be smoothly sliding back and forth on the rail.

Step 6: Install the GT2 belt on the X-axis. Each end of the pre-measured belt has a gold clamp, which will be positioned into slots below the extruder carriage.

Step 7: Install the belt onto the Creality Ender 3, carefully roll the extruder carriage over the belt, which will slide the belt under the wheel and into the empty space. Do the same procedure for the second wheel, positioning the other end of the belt beneath the carriage. There is a pully on each end of the X-axis to wrap the belt around, which will ensure smooth motion of the extruder.

## 5.2 SOFTWARE

### 5.2.1 RASPBIAN JESSY

Raspberry Pi OS (formerly Raspbian) is a Debian-based operating system for Raspberry Pi. It has been officially provided by the Raspberry Pi Foundation as the primary operating system for the Raspberry Pi family of compact single-board computers. Raspbian comes pre-installed with plenty of software for education, programming, and general use. It has Python, Scratch, Sonic Pi, Java, Mathematica and more.

#### 5.2.1.1 Installation

It is required to download 2 software's and 1 OS i.e., Raspbian, for this complete process.

**Downloads:**

- 1st software: The first software is Win32 Disk Imager.

  https://sourceforge.net/projects/win32diskimager/

- 2nd software: Second software is SD Card Formatter.

  https://www.sdcard.org/downloads/formatter_4/

- Raspbian OS: This is the Main operating system of the Pi.

  https://www.raspberrypi.org/downloads/raspbian/

Extract all files to the desktop.

**Installing Raspbian Jessie by downloading the image**

- Insert the micro-SD card into the card reader and connect it with the PC.
- Format the SD card with FAT file system. To format the SD card, SD Card Formatter is used.
- Download the Raspbian Jessie image from the official site. Two options are available on the landing page - Raspbian Jessie and Raspbian Jessie Lite. Raspbian Jessie has been selected and the full image is downloaded. Extract the zip file.
- For the next step, a software is needed to write the operating system image into the SD card.
- For this purpose, the Win32DiskManager tool is used.

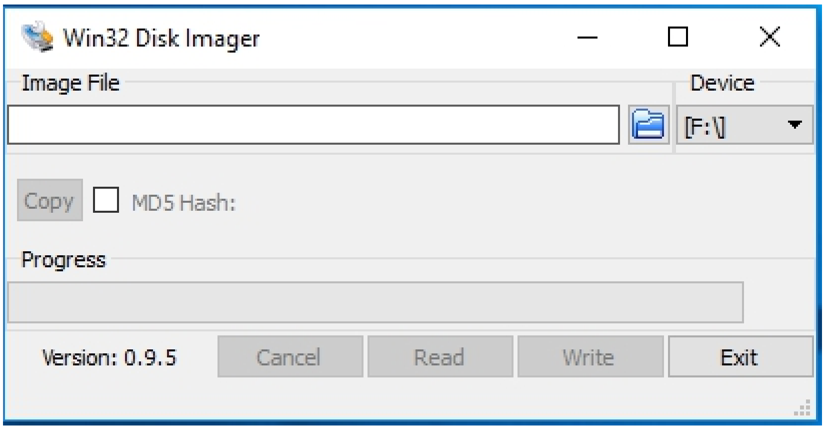


*Fig 5.4: Win32 Disk Imager*

- Select the SD card from available device options.
- Select the location of the Raspbian Jessie's image file wherever it was downloaded on the PC.
- Click Write button in Win32 Disk Imager.
- After some time, the memory card will be ready. On completion, remove the SD card and insert it into the Raspberry Pi's memory card slot. (The slot can be found on the backside of the Raspberry Pi).
- Power on the Raspberry Pi. The first boot will take some time (5 mins approx.).
- After the booting, it'll be the same as the below figure.

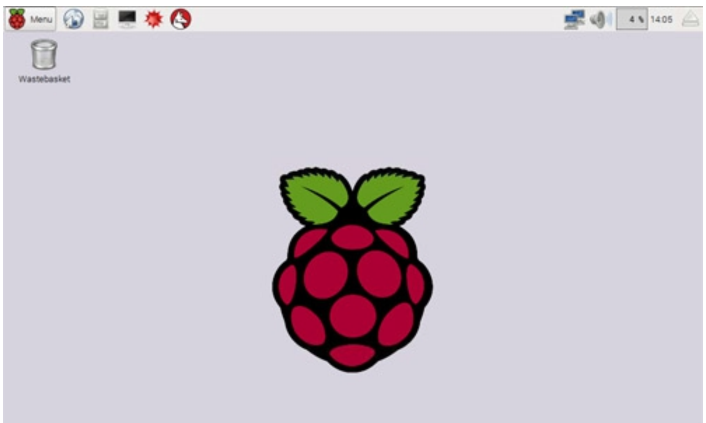

*Fig 5.5 Raspbian Jessie desktop screen*

### 5.2.2 OpenCV

OpenCV (Open-Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. The library is cross platform and free for use under the open-source BSD license. OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation. Wrappers in other languages such as C#, Perl, Ch, Haskell and Ruby have been developed to encourage adoption by a wider audience. All the new developments and algorithms in OpenCV are now developed in the C++ interface. OpenCV runs on a variety of platforms. Desktop: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD; Mobile: Android, iOS, Maemo, BlackBerry. In version 3.4, JavaScript bindings for a selected subset of OpenCV functions was released as OpenCV.js, to be used for web platforms

OpenCV's application areas include:

- 2D and 3D feature toolkits
- Ego motion estimation
- Facial recognition system
- Gesture recognition
- Human–computer interaction (HCI)
- Mobile robotics
- Motion understanding

- Object detection
- Segmentation and recognition
- Stereopsis stereo vision: depth perception from 2 cameras
- Structure from motion (SFM)
- Motion tracking
- Augmented reality
- To support some of the above areas, OpenCV includes a statistical machine learning library that contains:
- Boosting
- Decision tree learning
- Gradient boosting trees
- Expectation-maximization algorithm
- k-nearest neighbor algorithm
- Naive Bayes classifier
- Artificial neural networks
- Random forest
- Support vector machine (SVM)
- Deep neural networks (DNN)

### 5.2.2.1 INSTALLATION

#### 5.2.2.1.1 Installing OpenCV in Raspberry PI 3B+

**1. Installing required packages and libraries**

Step 1: sudo raspi-config, expand file system

Step 2:  sudo reboot

Step 3: df -h

Step 4: sudo apt-get purge wolfram-engine

Step 5: sudo apt-get upgrade

Step 6: sudo apt-get update

Step 7: sudo apt-get install build-essential cmake pkg-config

Step 8: sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev

Step 9: sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev

Step 10: sudo apt-get install libavcodec-dev libx264-dev

Step 11: sudo apt-get install libgtk2.0-dev

**2. Installing Python 3**

Step 1: sudo apt-get install libatlas-base-dev gfortran

Step 2: $ sudo apt-get install python3-dev

Step 3: cd ~

Step 4: wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip

Step 5: unzip opencv.zip

Step 6: wget -O opencv_contrib.zip
https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip

Step 7: wget https://bootstrap.pypa.io/get-pip.py

Step 8: sudo python get-pip.py

Step 9: sudo pip install virtualenv virtualenvwrapper

Step 10: sudo rm -rf ~/.cache/pip

Step 11: export WORKON_HOME=$HOME/.virtualenvs

Step 12: source /usr/local/bin/virtualenvwrapper.sh

**3. Creating Python virtual environment**

Step 1: mkvirtualenv cv -p python3

Step 2: source ~/.profile

Step 3: workon cv

**4. Installing Numpy python package**

Step 1: pip install numpy

**5. Compile and Install OpenCV**

Step 1: cd ~/opencv-3.1.0/

Step 2: mkdir build

Step 3: cd build

Step 4: cmake -D CMAKE_BUILD_TYPE=RELEASE \

Step 5: -D CMAKE_INSTALL_PREFIX=/usr/local \

Step 6: -D INSTALL_PYTHON_EXAMPLES=ON \

Step 7: -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.1.0/modules \

Step 8: -D BUILD_EXAMPLES=ON ..

Step 9: make -j4

**6. Finish Installing OpenCV in the Raspberry PI**

Step 1: sudo make install

Step 2: sudo ldconfig

Step 3: cd ~/.virtualenvs/cv/lib/python3.4/site-packages/

Step 4: ln -s /usr/local/lib/python3.4/site-packages/cv2.so cv2.so

**7. Verifying OpenCV 3.0 install**

Step 1: source ~/.profile

Step 2: workon cv

Step 3: python

```
>>> import cv2
>>> cv2.__version__
```

'3.1.0'

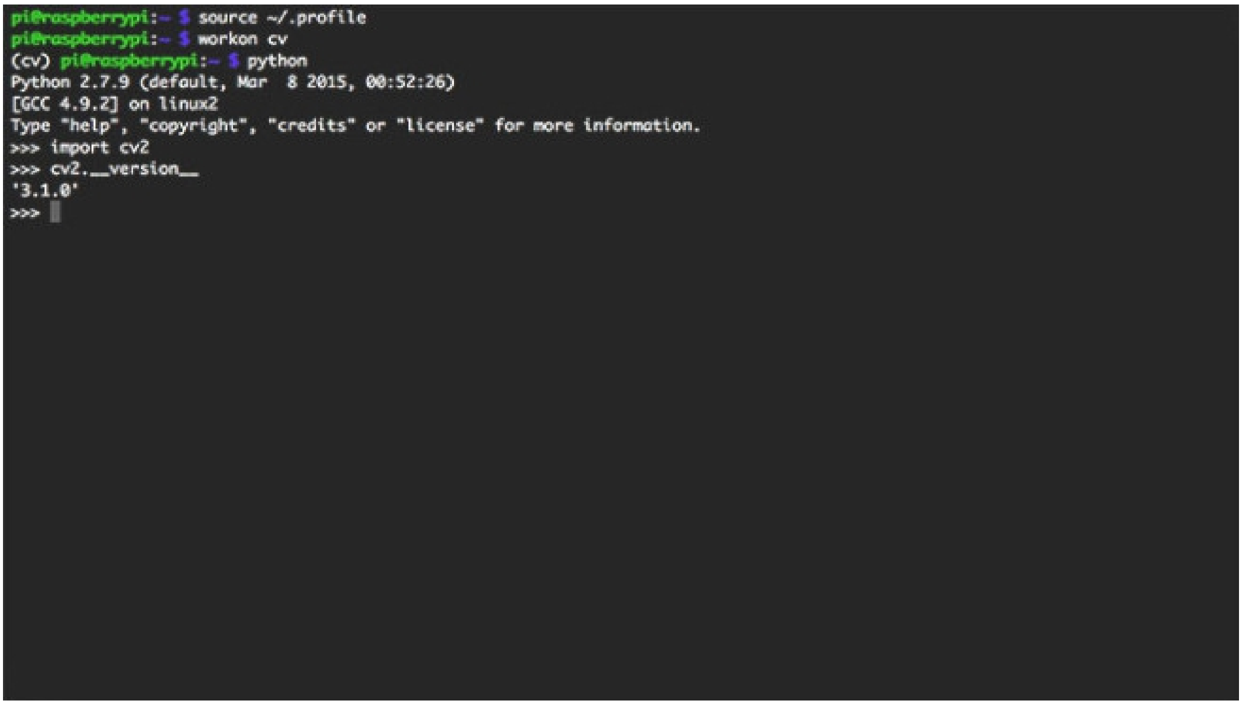


*Fig 5.6 Raspberry Pi terminal*

### 5.2.3 VNC

RealVNC is a company that provides remote access software. The software consists of a server and client application for the Virtual Network Computing (VNC) protocol to control another computer's screen remotely. For a desktop-to-desktop connection RealVNC runs on Windows, on Mac OS X, and on many Unix-like operating systems. A RealVNC client also runs on the Java platform and on the Apple iPhone, iPod touch and iPad and Google Android devices. A Windows-only client, designed to interface to the embedded server on Intel

AMT chipsets found on Intel vPro motherboards. Now, the Raspberry Pi can be accessed from any client by punching in this unique IP address and port number to gain access to the Raspberry Pi. The Username and Password has to be entered to take control of the device. RealVNC even has the option to chat with the server and FTP services as well.

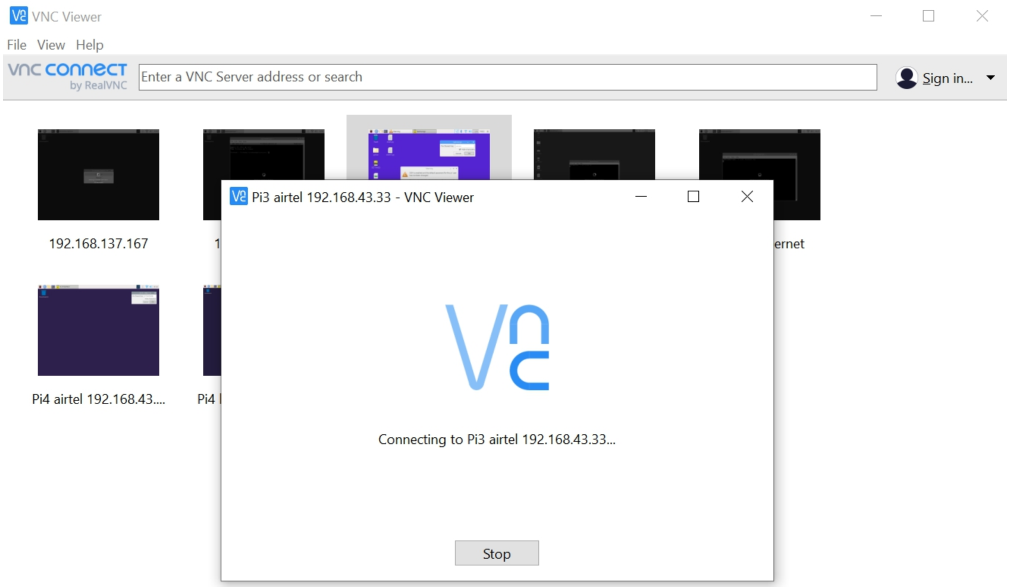


*Fig 5.7 VNC Viewer*

### 5.2.4 PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

Python supports modules and packages, which encourages program modularity and code reuse. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. Python uses duck typing and has typed objects but untyped variable names. Type constraints are not checked at compile time; rather, operations on an object may fail, signifying that the given object is not of a suitable type.

### 5.2.4.1 LIBRARIES

A Python library is a reusable chunk of code that can be included in programs/ projects. Compared to languages like C++ or C, a Python library do not pertain to any specific context in Python. Here, a 'library' loosely describes a collection of core modules. Python Package Index contains more than 72,000 packages offering a wide range of functionality, including:

- Graphical user interfaces, web frameworks, multimedia, databases, networking, and communications
- Test frameworks, automation and web scraping, documentation tools, system administration

### 5.2.4.1 IMPLEMENTATIONS

- The Raspberry Pi single-board computer project has adopted Python as its principal user-programming language.
- The python package FLASK is used for the server-side backend.

### 5.2.5 FLUTTER AND DART

### 5.2.5.1 What is Flutter?

Flutter is Google's UI toolkit for building beautiful, natively compiled applications for mobile (iOS and Android), web and desktop from a single codebase. Flutter SDK is a collection of tools that are helps us to develop our applications. This includes tools to compile the code into native machine code which includes (iOS and Android for mobile). Flutter's framework is a collection of reusable UI elements (buttons, text inputs, sliders, and so on) that can be personalize for our own needs. Flutter's layered architecture allows for full customization, which results in incredibly fast rendering and expensive and flexible designs. Flutter's widgets incorporate all critical platform differences such as scrolling, navigation, icons and fonts, and the Flutter code is compiled to native ARM machine code using Dart's native compilers. For further information related to flutter, refer the official documentation www.flutter.dev

### 5.2.5.2 What is Dart language?

Dart language was created by Google in October 2011. Dart focuses on front-end development and can be used to create mobile and web applications. Dart is a typed object programming language; its syntax is very similar to JavaScript. Dart AOT-compiles apps to native machine code for instant start up. Dart's async-await is used for user interfaces containing event-driven code. Dart's Hot-Reload feature helps with instant update with UI and features, it works by injecting updated source code files into the running Dart Virtual Machine.

### 5.2.5.3 WINDOWS INSTALLATION

Step 1: Download the latest stable release of Flutter SDK.

Step 2: Download and install 7-zip file extractor.

Step 3: Extract the downloaded Flutter SDK zip file.

Step 4: Place the extracted file in any drive folder, for e.g.: C, D, F etc.

Step 5: Add Flutter to PATH environment variable, type env in the search bar.

Step 6: Select edit environment variables.

Step 7: Under user variables, check if there is an entry called Path, if yes append full path to flutter\bin using; as a separator from existing values.

Step 8: If entry doesn't exit, create a new user variable named Path with full path to flutter\bin as its value.

Step 9: Open command prompt and type where flutter dart

```
C:\Users\HP>where flutter dart
F:\flutter\bin\flutter
F:\flutter\bin\flutter.bat
F:\flutter\bin\dart
F:\flutter\bin\dart.bat
F:\flutter\bin\cache\dart-sdk\bin\dart.exe
```

*Fig 5.8 Flutter path verification in CMD*

### 5.2.5.4 ANDROID STUDIO SETUP

Step 1: Download and install the latest stable version of Android Studio from www.developer.android.com/studio

Step 2: Launch android studio

Step 3: Go through the Android Studio Setup Wizard and check yes for latest Android SDK, Android SDK Command-line Tools and Android SDK Build-Tools.

Step 4: Go to File->Settings->SDK->Path and enter the path to flutter/bin

### 5.2.5.5 ANDROID DEVICE SETUP

Android device should be running at Android 4.1 (API level 16) or higher.

Step 1: Enable developer options by tapping the build number 7 times

- Android 9 (API level 28) and higher: Settings > About Phone > Build Number
- Android 8.0.0 (API level 26) and Android 8.1.0 (API level 26): Settings > System > About Phone > Build Number
- Android 7.1 (API level 25) and lower: Settings > About Phone > Build Number

Step 2: Go to developer options now and enable USB Debugging

### 5.2.5.6 SETUP VERIFICATION

Step 1: Connect the mobile device.

Step 2: Open command prompt

Step 3: Type flutter doctor

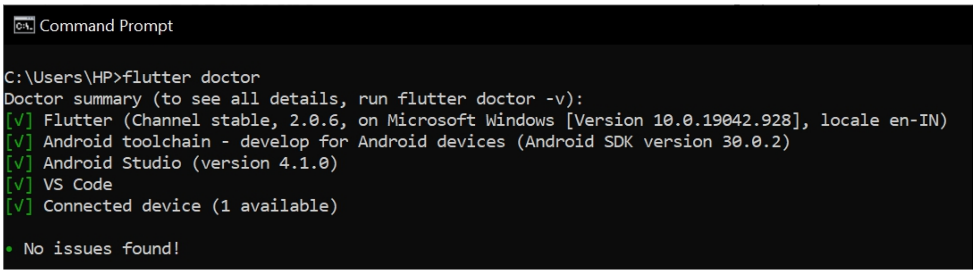


*Fig 5.9 Verifying flutter installation in CMD*

### 5.2.6 CAD

Computer-Aided Design (CAD) is the use of an application to help create or optimize a design. Therefore, CAD software allows engineers, architects, designers, and others to create precision drawings or technical illustrations in 2D or 3D. This category of software can increase productivity, improve quality, and maximize organization by creating a documentation database for manufacturing.

CAD software has a host of applications, including the design of manufacturing parts, electronic circuit boards, prototypes for 3D printers, and buildings. Typically, this software uses either traditional vector-based graphics or raster graphics which show how finished objects would actually look.

CAD software can also facilitate the flow from the design process to the manufacturing process. This software can simulate the movement of a part through the manufacturing process in three dimensions. As this software becomes ever better at simulating the manufacturing process, specialized software for designing the manufacturing process and controlling machine tools called Computer-Aided Manufacturing (CAM) has become integrated with CAD as a single platform.

Some CAD programs are specially designed for specific industries, or have add-on sets to make your CAD experienced tailored to your application. For example, Autodesk tools are really popular within the animation and art industries, whereas SketchUp sees a lot of popularity amongst architects given its tools for building design.

### 1) Features & Capabilities

Professional-level software packages usually have the following features:

- 2D/3D Design
- Electrical design
- CAM integration
- Simulation and analysis, such as simulating real world use of an item to identify areas with a propensity for thermal stress and buckling
- Augmented reality
- Data management
- Additive manufacturing

### 5.2.7 FUSION 360

Fusion 360 helps students and educators prepare for the future of design. It's the first 3D CAD, CAM, and CAE tool of its kind, connecting the entire product development process into one cloud-based platform. Download the software today, then turn ideas into reality.

### 5.2.8 SLICER

A slicer is a program that converts digital 3D models into printing instructions for a given 3D printer to build an object. In addition to the model itself, the instructions contain user-entered 3D printing parameters, such as layer height, speed, and support structure settings.

Every 3D printing technology creates 3D objects by adding material layer-by-layer. Slicer software is therefore appropriately named because it virtually “cuts” 3D models into many horizontal 2D layers that will later be printed, one at a time.

Cura is an open-source printer. It was created by David Braam who was later employed by Ultimaker, a 3D printer manufacturing company, to maintain the software. Cura is available under LGPLv3 license. Cura was initially released under the open source Affero General Public License version 3, but on 28 September 2017 the license was changed to LGPLv3. This change allowed for more integration with third-party CAD applications. Development is hosted on GitHub.Ultimaker Cura is used by over one million users worldwide and handles 1.4 million print jobs per week. It is the preferred 3D printing software for Ultimaker 3D printers, but it can be used with other printers as well.

Ultimaker Cura works by slicing the user’s model file into layers and generating a printer-specific g-code. Once finished, the g-code can be sent to the printer for the manufacture of the physical object.

The open source software, compatible with most desktop 3D printers, can work with files in the most common 3D formats such as STL, OBJ, X3D, 3MF as well as image file formats such as BMP, GIF, JPG, and PNG.

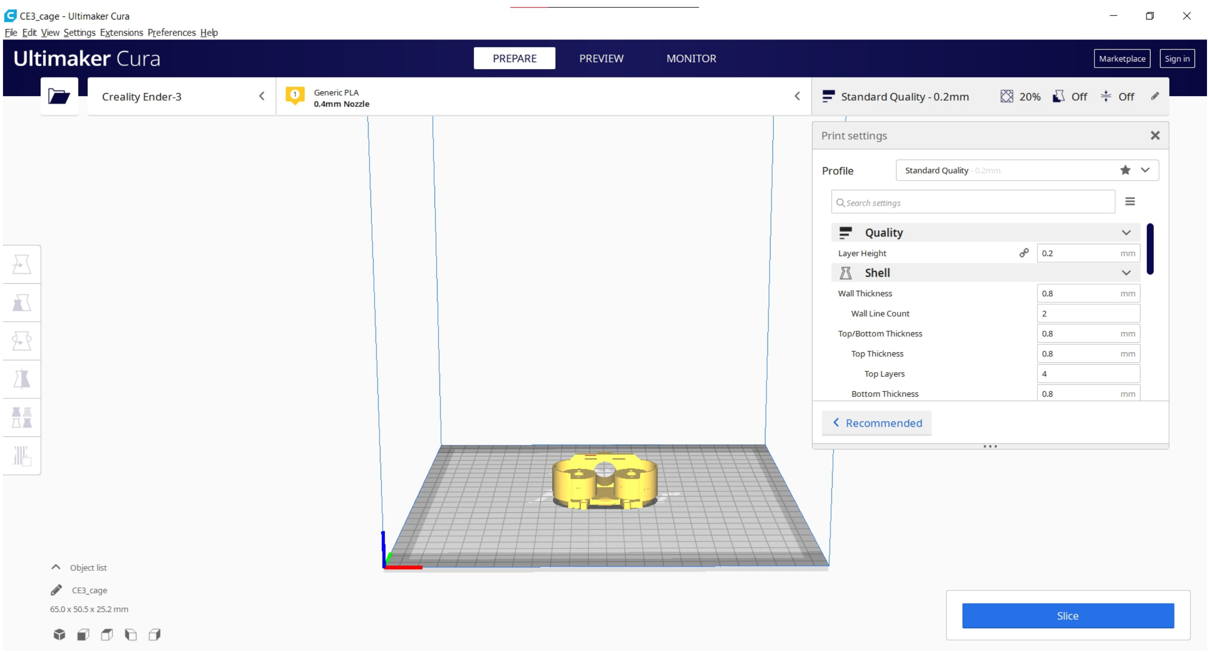


*Fig 5.10 Slicer application*

### 5.2.9 FLASK

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

### 5.2.9.1 INSTALLATION

**1. Python Version**

It's recommended to use the latest version of Python 3. Flask supports Python 3.5 and PyPy.

**2. Dependencies**

These distributions will be installed automatically when installing Flask.

- Werkzeug implements WSGI, the standard Python interface between applications and servers.
- Jinja is a template language that renders the pages the application serves.
- MarkupSafe comes with Jinja. It escapes untrusted input when rendering templates to avoid injection attacks.
- ItsDangerous securely signs data to ensure its integrity. This is used to protect Flask's session cookie.
- Click is a framework for writing command line applications. It provides the flask command and allows adding custom management commands.

**3. Virtual environments**

Use a virtual environment to manage the dependencies for the project, both in development and in production. Virtual environments are independent groups of Python libraries, one for each project. Packages installed for one project will not affect other projects or the operating system's packages.

Python 3 comes bundled with the venv module to create virtual environments.

**4. On Windows:**

py -3 -m venv venv

**5. Activate the environment**

On Windows:

> >venv\Scripts\activate

**6. Install Flask**

Within the activated environment, use the following command to install Flask:

>>pip install Flask

### 5.2.10 DRONEKIT

DroneKit-Python allows developers to create apps that run on an onboard companion computer and communicate with the ArduPilot flight controller using a low-latency link. Onboard apps can significantly enhance the autopilot, adding greater intelligence to vehicle behavior, and performing tasks that are computationally intensive or time-sensitive (for example, computer vision, path planning, or 3D modelling). DroneKit-Python can also be used for ground station apps, communicating with vehicles over a higher latency RF-link.

The API communicates with vehicles over MAVLink. It provides programmatic access to a connected vehicle's telemetry, state and parameter information, and enables both mission management and direct control over vehicle movement and operations. Drone Kit-Python is an open source and community-driven project.

DroneKit-Python is compatible with vehicles that communicate using the MAVLink protocol (including most vehicles made by 3DR and other members of the Drone Code foundation). It runs on Linux, Mac OS X, or Windows.

**1. API**

The API provides classes and methods to:

- Connect to a vehicle (or multiple vehicles) from a script
- Get and set vehicle state/telemetry and parameter information.
- Receive asynchronous notification of state changes.
- Guide a UAV to specified position (GUIDED mode).
- Send arbitrary custom messages to control UAV movement and other hardware (GUIDED mode).
- Create and manage waypoint missions (AUTO mode).
- Override RC channel settings.

### 5.2.10.1 WINDOWS INSTALLATION

Step 1: pip install dronekit

Step 2: pip install dronekit-sitl

### 5.2.11 DroneKit-SITL

DroneKit-SITL is the simplest, fastest and easiest way to run SITL on Windows, Linux (x86 architecture only), or Mac OS X. It is installed from Python's pip tool on all platforms, and works by downloading and running pre-built vehicle binaries that are appropriate for the host operating system.

**1. Setting up a Simulated Vehicle (SITL)**

The SITL (Software in The Loop) simulator creates and test DroneKit-Python apps without a real vehicle.

SITL can run natively on Linux (x86 architecture only), Mac and Windows, or within a virtual machine. It can be installed on the same computer as DroneKit, or on another computer on the same network.

The sections below explain how to install and run SITL, and how to connect to DroneKit-Python and Ground Stations at the same time.

**2. Installation**

Step 1: pip install dronekit-sitl -UI

**3. Running SITL**

To run the latest version of Copter for which the binaries are available (downloading the binaries if needed), call:

Step 1: dronekit-sitl copter

SITL will then start and wait for TCP connections on 127.0.0.1:5760.

### 5.2.12 MAVProxy

A UAV ground station software package for MAVLink based systems

MAVProxy is a fully-functioning GCS for UAV's, designed as a minimalist, portable and extendable GCS for any autonomous system supporting the MAVLink protocol (such as one using ArduPilot). MAVProxy is a powerful command-line based "developer" ground station software. It can be extended via add-on modules, or complemented with another ground station, such as Mission Planner, APM Planner 2, QGroundControl etc, to provide a graphical user interface.

It has a number of key features, including the ability to forward the messages from the UAV over the network via UDP to multiple other ground station software on other devices.

MAVProxy is commonly used by developers (especially with SITL) for testing new builds.

MAVProxy was first developed by CanberraUAV, to enable the use of companion computing and multiple datalinks with ArduPilot. It has grown to be one of the most versatile tools in the ArduPilot ecosystem, and many of the features' users now see in other GCS tools can trace their origins to MAVProxy.

**1. Features**

It is a command-line, console-based app. There are plugins included in MAVProxy to provide a basic GUI.

Can be networked and run over any number of computers.

It's portable; it should run on any POSIX OS with python, pyserial,
and select() function calls, which means Linux, OS X, Windows, and others.

The light-weight design means it can run on small netbooks with ease.

It supports loadable modules, and has modules to support console/s, moving maps, joysticks, antenna trackers, etc

Tab-completion of commands.

**2. Installation**

Step 1: pip install MAVProxy

# CHAPTER 6

# DATABASE AND MAPS

## 6.1 FIREBASE DATABASE

Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure.

Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.

Key Features:

1. Authentication
2. Realtime database
3. Analytics
4. Performance Monitoring

### 6.1.1 CREATING NEW PROJECT:

Step 1: Go to www.firebase.google.com

Step 2: Login to a google account

Step 3: Select Go to Console on the top right corner.

Step 4: Select create new project

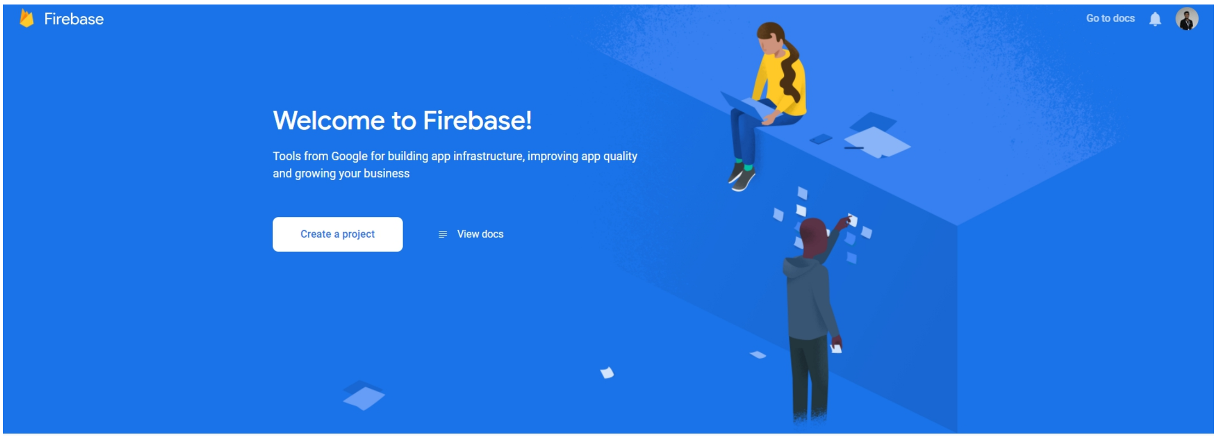


*Fig 6.1 New project firebase*

Step 5: Enter the project name, hit continue

Step 6: Check yes for enable analytics, hit continue.

Step 7: Select the country, accept the terms.

Step 8: Select Create Project

Step 9: Wait till the project is created.

### 6.1.2 ANDROID APP INTEGRATION

Step 1: Select android in main screen

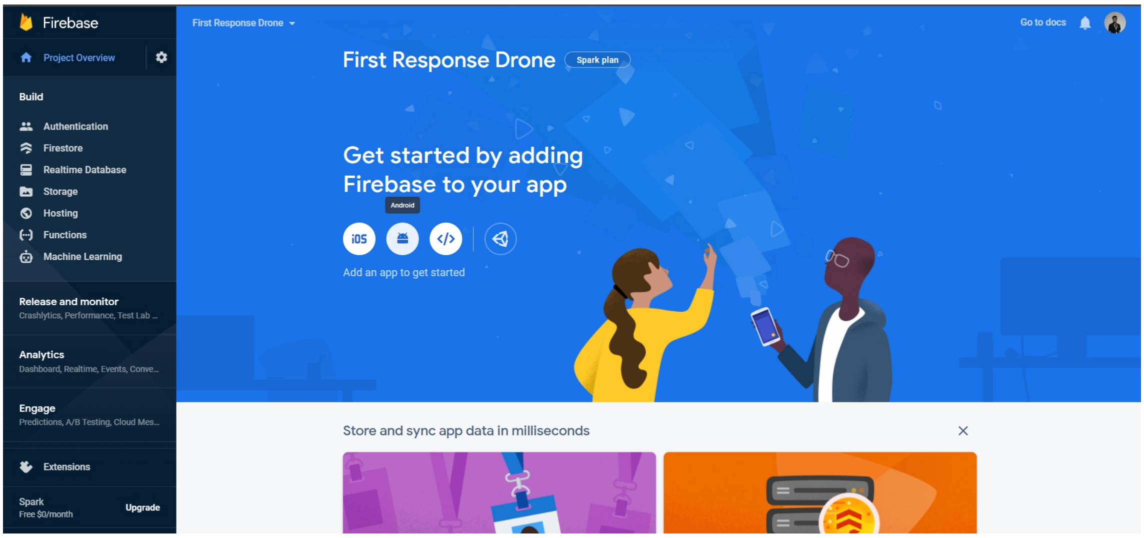


*Fig 6.2 Firebase console for android*

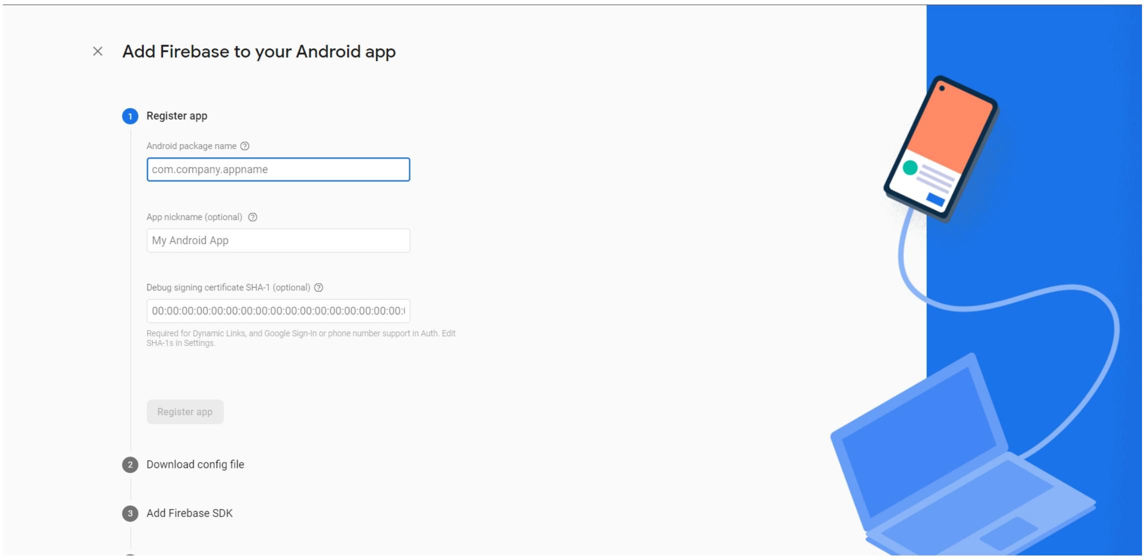


*Fig 6.3 Firebase android setup*

Step 2: Open the created flutter project

Step 3: Go to App-> Build.grade file->Copy the package name

Step 4: Paste the copied package name in the firebase console

Step 5: Open terminal inside the flutter project and execute this command

keytool -list -v -alias androiddebugkey -
keystore %USERPROFILE%\.android\debug.keystore

Step 6: Enter password as "android" and hit enter

Step 7: Copy-paste the sha-1 certificate in firebase console

Step 8: Download the google-services.json file

Step 9: Paste the google-services.json inside app folder in the flutter project

Step 10: Paste the lines shown project level build.gradle

Step 11: Paste the lines shown in app level build.gradle

Step 12: Hit next and finish, the app will now be registered to firebase.

Step 13: Rebuild the flutter app.

### 6.1.3 WEB APP INTEGRATION

Step 1: Go to www.firebase.google.com

Step 2: Login to the google account

Step 3: Select Go to Console on the top right corner.

Step 4: Select the Project and Choose Add App

Step 5:  Select Web App in main screen

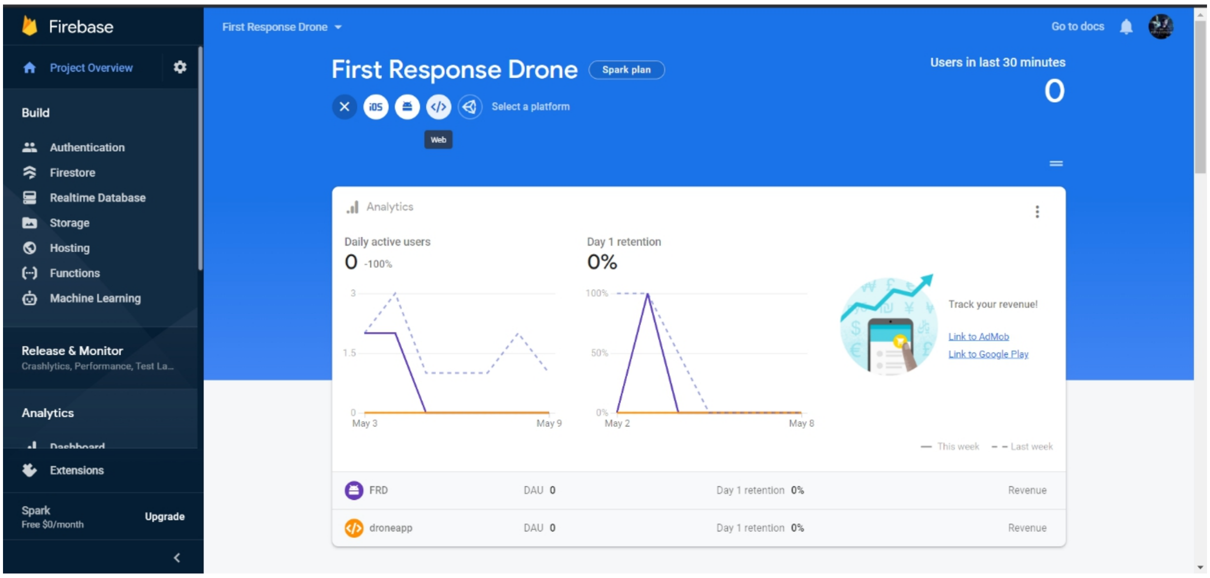


*Fig 6.4 Firebase console for web*

Step 6: Register the App

Step 7: Copy the SDK given To the Java Script File of the Front End to connect it to the Firebase

Step 8:  Download the google-services.json file from the Project console

Step 9:  Add the google-services.json to the python Back End File to connect it to the  Firebase

Now Both the Front End and Back End will be registered to Firebase.

### 6.1.4 AUTHENTICATION SETUP

Step 1: Select Authentication under Build options on the left

Step 2: Select get started

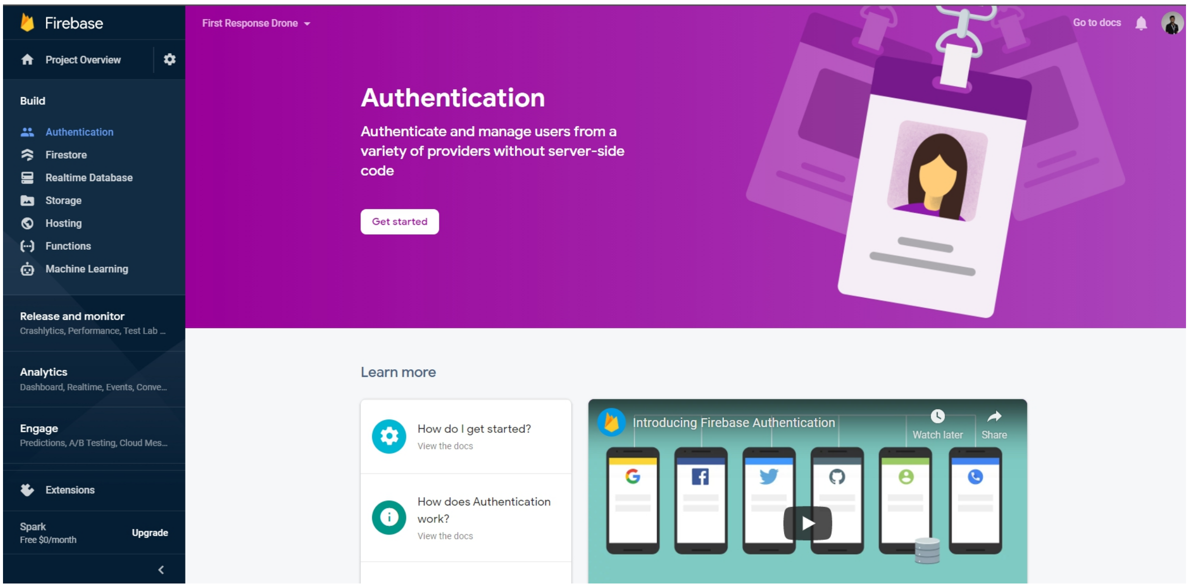


*Fig 6.5 Firebase authentication screen*

Step 3: Enable any authentication from the different available options

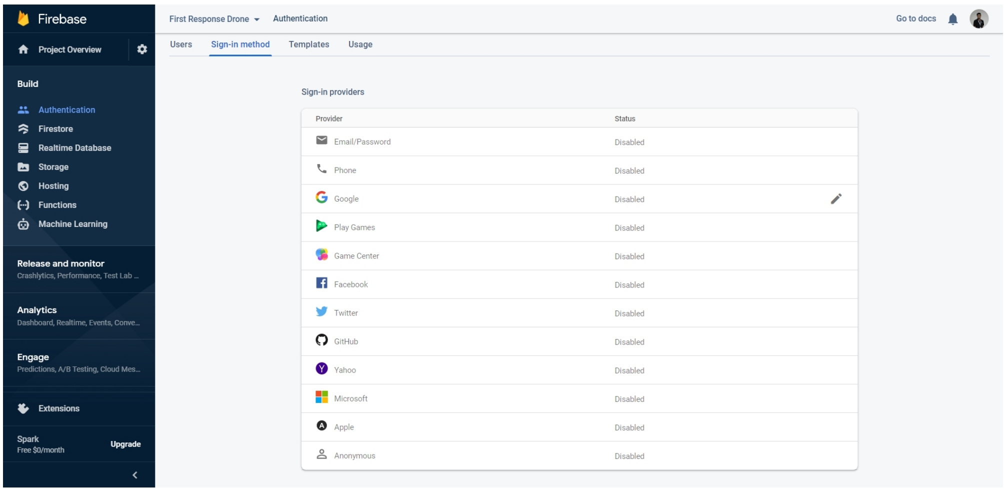


*Fig 6.6 Firebase authentication types*

### 6.1.5 REALTIME DATABASE SETUP:

Step 1: Select Realtime Database under Build options on the left side.

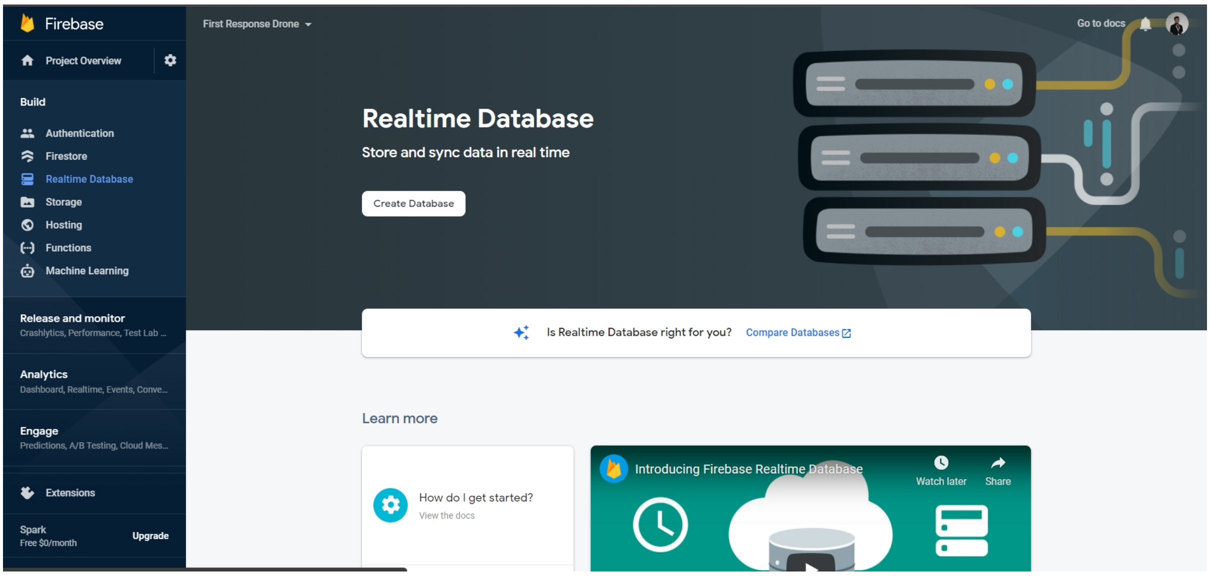


*Fig 6.7 Firebase Realtime Database console*

Step 2: Select create database

Step 3: Navigate to rules on the top bar, change the rules to the following and select publish

## 6.2 GOOGLE MAPS API SETUP

Google Maps as the name suggests is a maps API developed and maintained by Google. The API is an essential tool which is required for the custom-made simulator, Drone markers, Emergency markers, Polylines (Path traced by the drone) are marked and visualised in real-time in google maps.

### 6.2.1 CONFIGURATION

Step 1: Go to the google cloud platform console

Step 2: Login to the google account

Step 3: The page will redirect to the overview page, select create new project

Step 4: Enter a project name and select create.

Step 5: Select Maps SDK for Android in the APIs section and select enable.

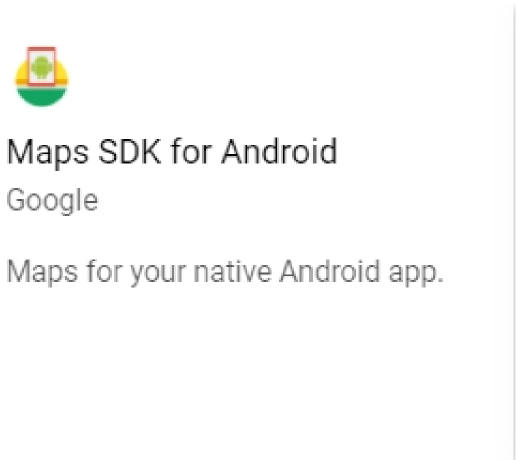


*Fig 6.8 Google maps SDK for android*

Step 6: The API key will be ready under credentials section

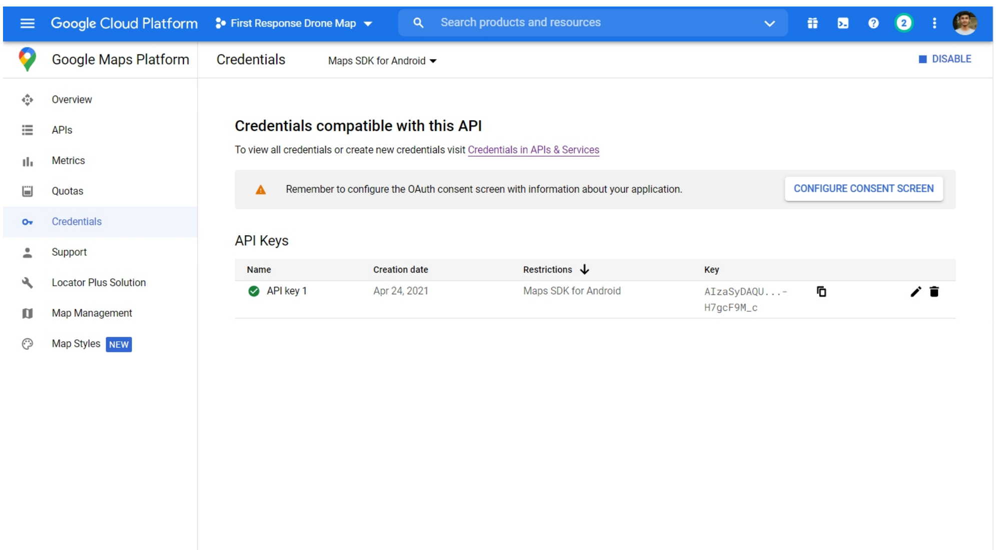


*Fig 6.9 Google maps API key console*

Step 7: Select edit API key option, check yes for restrict key and select Maps SDK for android and hit save.

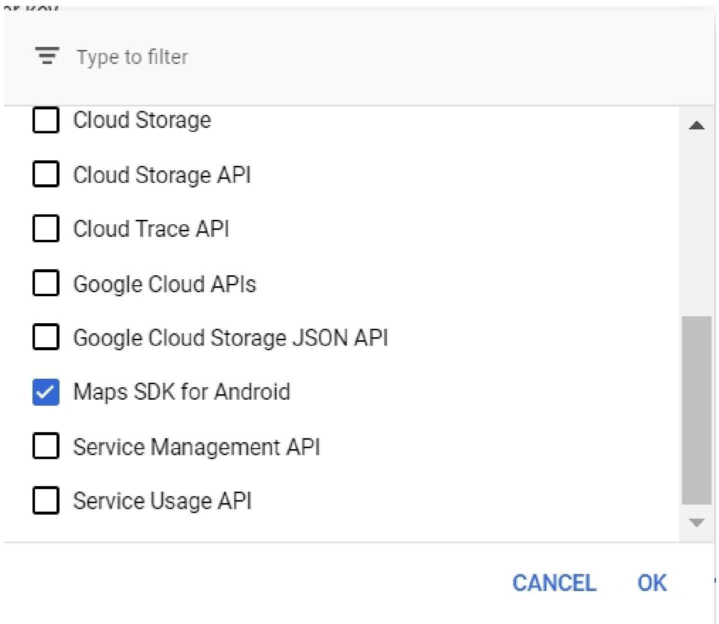


*Fig 6.10 Google Maps API key restriction*

# CHAPTER 7

# IMPLEMENTATIONS AND RESULT

## 7.1 WORKING

In case of an emergency, a report is reported with the help of the FRD mobile app made with Flutter and Dart. The type and exact cause of emergency is also reported and sent to the database, once reported the server will pick it up. The server will then map the emergency to the nearest drone in a 3000m radius. The drone will be carrying emergency medicines in a container which is cooled by a Peltier module, it'll also be streaming in real-time to the server, and the server will ping to this stream and perform object detection with the help of Haar classifier. Once the FRD is mapped to an emergency, the server will intimate the FRD for launch with the destination coordinates. With the help of DroneKit, the processor on board will get these coordinates and send flight instructions to the FRD, so that the drone reaches the destination on its own. The full model of the FRD can be seen in the below figure

*Fig 7.1 FRD side view and top view*

## 7.2 DART GUN

Since the FRD is designed to assist police department, a non-lethal, reloadable dart-gun is designed. The dart-gun was designed and 3D printed with Fusion 360 and Creality Ender 3. The current version wasn’t implemented due to hardware limitations. The model and the prototype can be seen in the below figures.

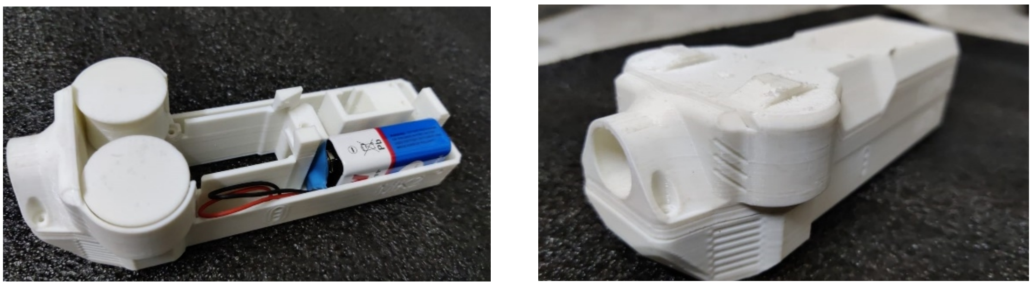

*Fig 7.2 Dart Gun*

## 7.3 SIMULATED RESULT

The simulator is entirely been designed and developed from scratch due to certain limitations with existing simulators and certain custom requirements. The administrator side was designed with Flask as its backend and implements OpenCV and Google Maps API. The customer side mobile application was designed with Flutter SDK and implements Geolocation and Google Maps API.

The step-by-step results are shown in the below figures. They are split into two modules, Client side and Administrator Side.

### 7.3.1 CLIENT SIDE

Once the client opens the app, they have to login with their Google Account, Google auth is few of the most secure authentication there is. Once registered they're redirected to the emergency screen. This can be seen in the below figures.

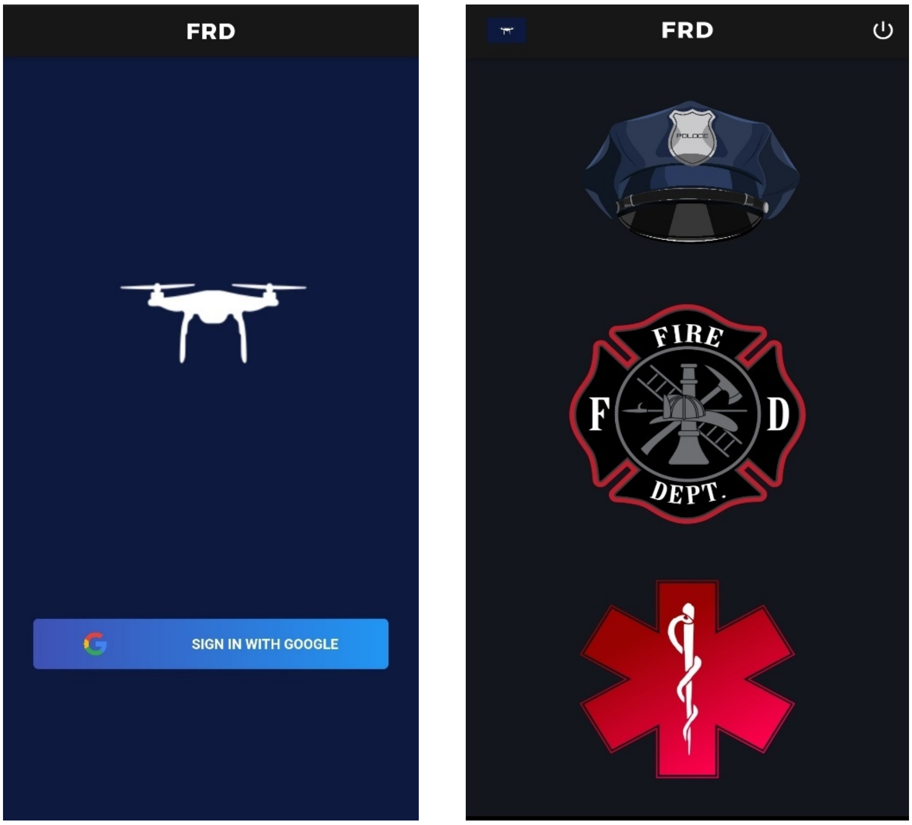


*Fig 7.3 Login and Emergency Screen*

Then the client has to select the type of emergency, each type of emergency has a pre-set list of names and the client has to select any one of the existing ones there is, the client application will also assist with a haptic vibration on selection and will also give a voice note on what is currently selected. This can be seen in the below figures.

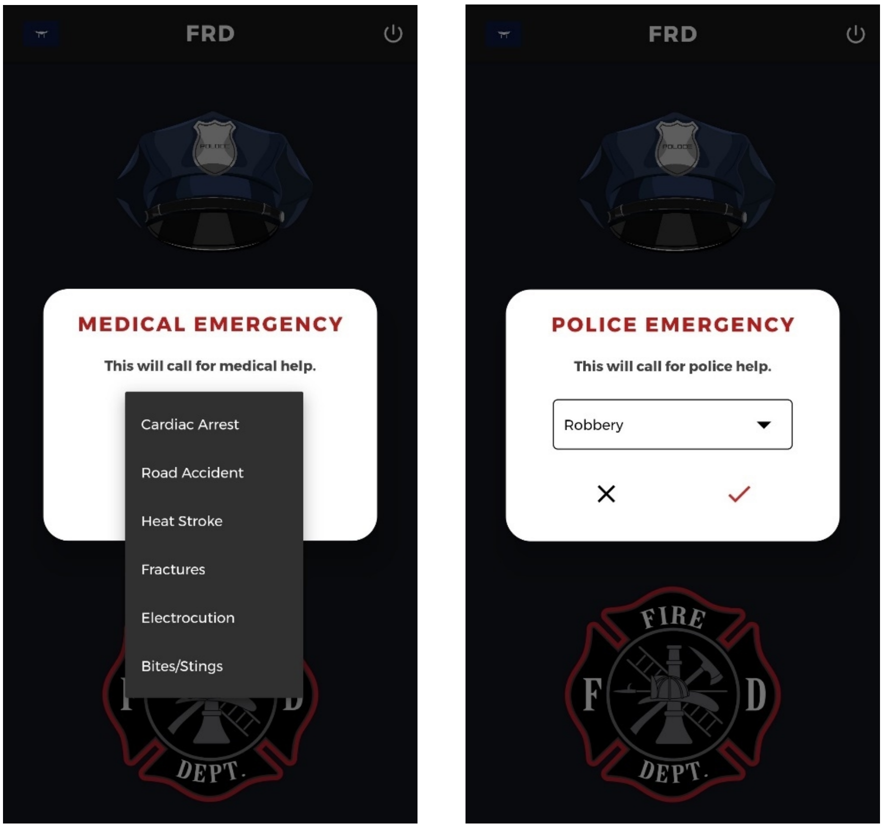


*Fig 7.4 Registering emergency*

Once the type of emergency is selected, the algorithm will fetch the current latitude and longitude coordinates with the help of Geo Location library and set the necessary details in the database. Then it is pushed to the waiting screen while the algorithm assigns the nearest FRD. Meanwhile, with the current coordinates and the type of emergency, the algorithm also identifies the nearest police station or hospital or fire station with the data in the database and maps the nearest appropriate station to the reported area and notifies them about the emergency. This can be seen in the below figures.

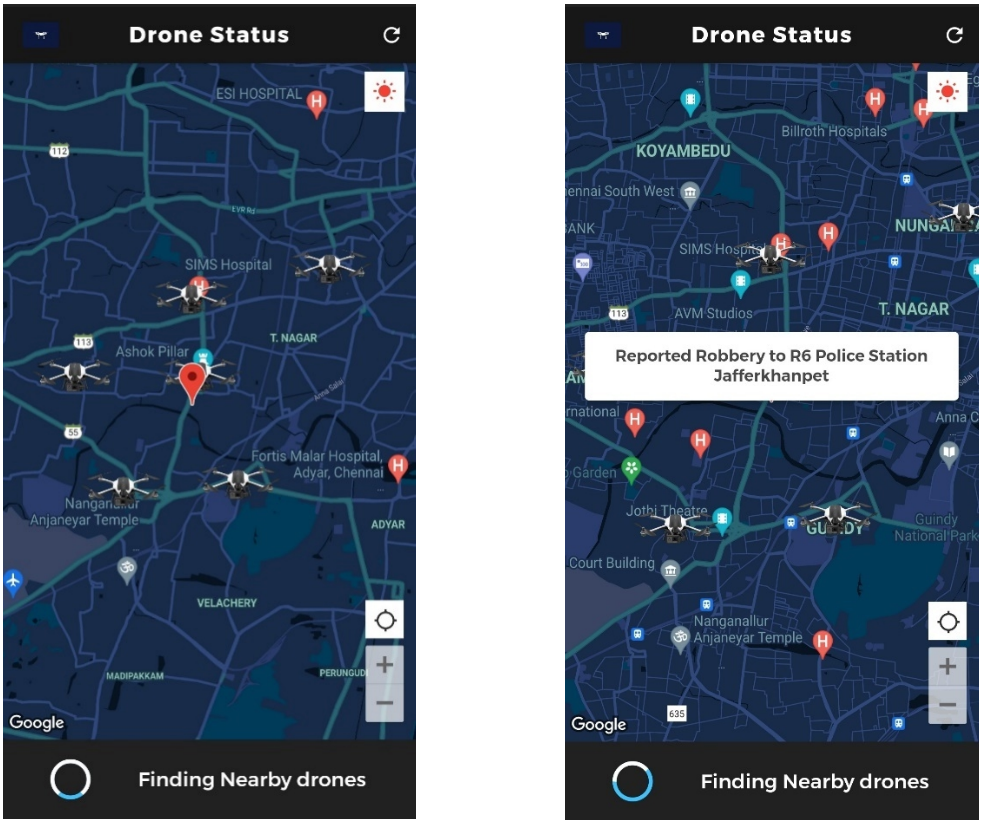


*Fig 7.5 Waiting screen*

Once the nearest FRD is assigned to the response, the administrator sends a conformation and the screen changes to a poly line screen, where the current status and location of the FRD, the path traced by it and the ETA are updated to the client in real time.

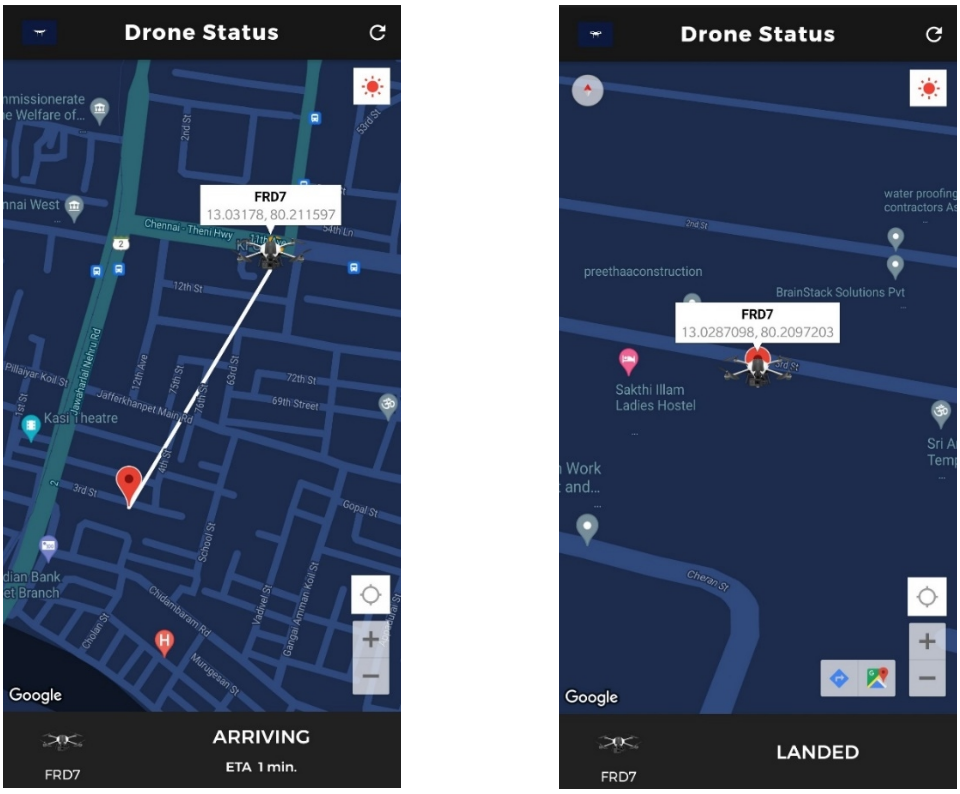


*Fig 7.6 FRD in pursuit*

Once the emergency situation is addressed the screen pops back to emergency screen and the data is deleted from the database for maintaining database health.


*Fig: 7.7 Post emergency*

### 7.3.2 ADMINISTRATOR SIDE

The administrator side has been developed with JavaScript and Flask. It uses Google Maps API and OpenCV with Haar Classifier for object detection in real time. Here, the administrator will be able to view all the emergency requests in one screen and the screen refreshes every 5 seconds automatically to check for any new requests. This can be seen in the below figure.

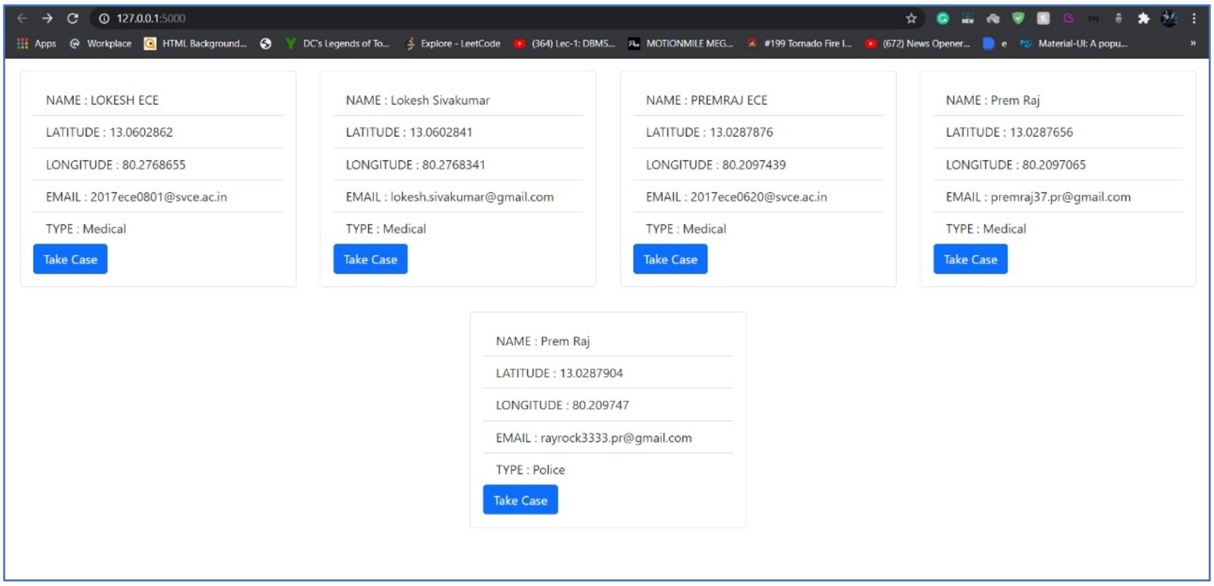


*Fig 7.8 Emergency details screen*

Then the administrator in the has to select any one of the emergencies to start the simulation. Once launched the MAVProxy runs with SITL to fetch the current FRD coordinates from the drone, current status and continuously feeds it to the database to update the maps on both the client side and administrator side. After successfully addressing the emergency situation the algorithm updates the status and clears the current record from the database to maintain health of the database. These can be viewed in the below figures.

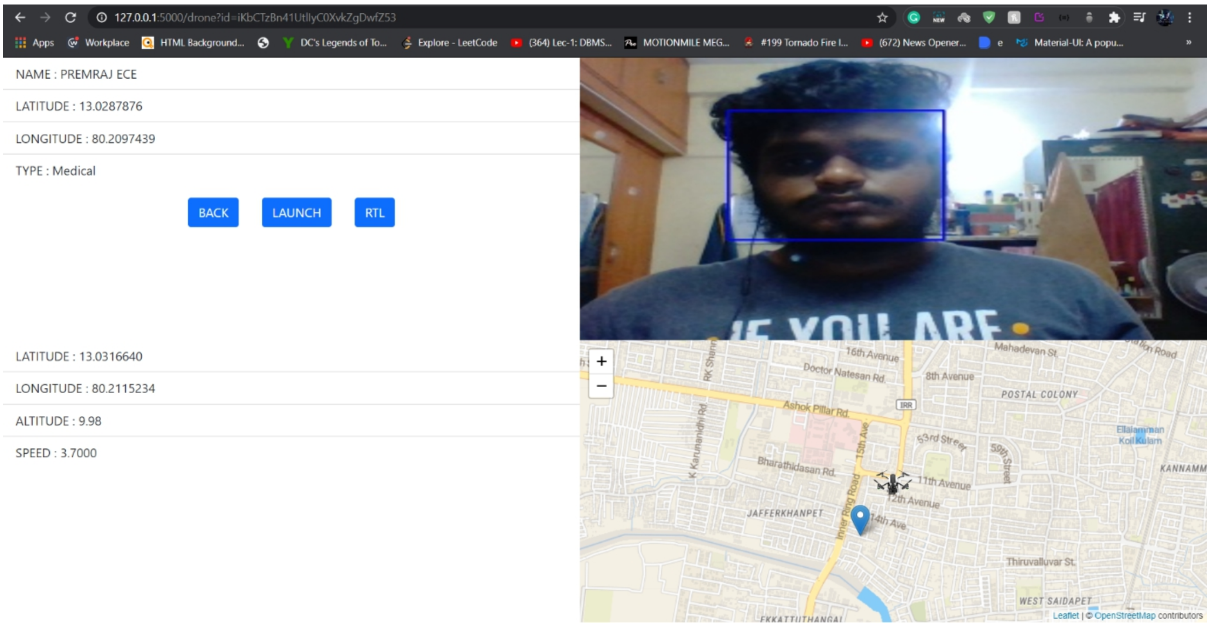


*Fig 7.9 Simulator main screen.*

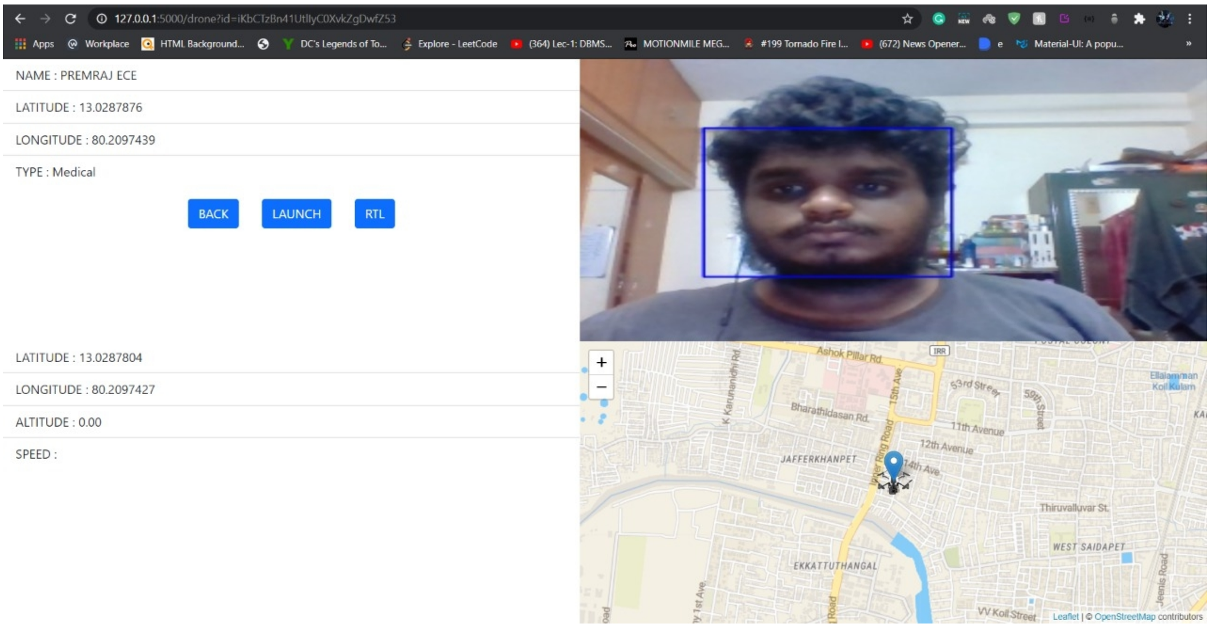


*Fig 7.10 Simulator on successful addressal of the emergency*

# CHAPTER 8

# CONCLUSION AND FUTURE SCOPE

The outcome of this project is to propose an innovative addition to the pre-existing UAVs and drones by equipping it with cutting edge autonomy. The features and methodologies proposed will display a significant elevation in terms of performance and efficiency when compared to other UAV’s. The core aspects of the drone are to carry a first aid box in a cold state and provide a bird’s eye view of the situation and autonomous flight. The camera equipped in the drone, provides a live feed to the server which is tapped and being fed to the object detection algorithm in real time stands out compared to other existing models. The autonomous aspect of the drone is a huge plus, this will result in much less response times than the manually controlled UAVs which are available.

As discussed earlier the application of the drone is to provide aid and support to civil service departments like Medical, Police and Fire department in India. However, over the years, the structure of the drones can be enhanced in order to support payloads of greater weight for a longer duration, better implementation of non-lethal weapons and flight path. The future scope of the drone involvement is vast. The FRD will be a valuable piece of technology which can be used to save a lot of lives.

# REFERENCES


[1] Piotr Kardasz, Jacek Doskocz, Mateusz Hejduk, Paweł Wiejkut and Hubert Zarzycki in Drones and Possibilities of Their Using, Journal of Civil & Environmental Engineering 2016.

[2] Anna Konert, Jacek Smereka, and Lukasz Szarpak in The Use of Drones in Emergency Medicine: Practical and Legal Aspects, Hindawi Emergency Medicine International Volume 2019.

[3] Kyle Stelmack in Weaponized Police Drones and Their Effect on Police Use of Force, Pittsburgh Journal of Technology, Spring 2015.

[4] Abinesh.D.V, Deepak.A.K in Fire Fighting Drone, IJIERE, Volume 4, Special Issue 1, 2017.

[5] Bhakti Yudho Suprapto, M Ary Heryanto, Herwin Suprijono, Jemie Muliadi, Benyamin Kusumoputro in Design and Development of Heavy-lift Hexacopter for Heavy Payload, iSEMANTIC Conference, 2017.

[6] Dang-Khanh Le, Taek-Kun Nam in A study on the modeling of a hexacopter, Journal of the Korean Society of Marine Engineering (JKOSME), Vol. 39, No. 10 pp. 1023~1030, 2015.