American International University-Bangladesh (AIUB)

Controlling a UAV Using Remote Framebuffer (RFB) Protocol.

An approach to find the simplest, cost effective and efficient way to build and control a drone from anywhere in the world without using autopilot.

Mahmood, Shakur
(ID: 22-92338-1)

A Thesis submitted for the degree of Master of Science (MSc) in Computer Science (CS) at American International University- Bangladesh in August, 2023

Faculty of Science and Technology (FST)

Abstract

Drone technologies are getting more attention to the researchers due to it's growing popularity. It is an easier and efficient way to explore the remote areas of the world. Specially, it's easier control mechanism is the real reason behind it's growing popularity. An unmanned aerial vehicle (UAV) system has two parts, the drone itself and the control station at ground which is called ground control systems (GSC). FPV (First Person View) drones are most commonly used where one person views the world through a camera fitted onto the drone, and use that 'first person' perspective to fly the drone. Here, the system is limited to one person's view only. If we can control a drone using client-server protocol, it enlarges our scope and widens the area to control it. In this case, cell phone towers can help us to establish such type of communication in UAVs. This research will show the possibility of creating such drone using Remote Framebuffer (RFB) protocol and will discuss the comparative analysis with the other existing methods to control it. As it will work using a client-server based model and this thesis wants to control the UAV from anywhere in the world, so using 4g/5g internet (it is the internet provided by the cell phone towers) will be an efficient and cost effective way of communication with the UAV than other alternatives like the satellite internet or the balloon mesh network are not as efficient or cost-effective. Using the RFB protocol, multiple permitted users can access the drone at the same time. To control the drone through the internet using client-server based RFB protocol, it is needed to use proper devices that support this protocol. To do this we can use arm based computer like raspberry pi as a flight controller. In this approach we can build and control a UAV using only 4 types of equipment- brushless DC (BLDC) motors, electronic speed controllers (ESC), a power source like lithium polymer (LIPO) battery and a raspberry pi with internet dongle. As a ground station, any device(s) that support(s) RFB protocol (Android, IOS, Windows, MAC) with internet connection can be used. As raspberry pi has a webcam connection port, one can attach a webcam if he/she wants to collect photos or, videos. It is the simplest and cost effective way to control a drone from anywhere in the world. This thesis uses pulse width modulation technique in python to control the BLDC motors those are used in this thesis. Later, a software can be build for the purpose of user-friendliness. Using RBF protocol, a user can get full satisfaction of controlling a UAV due to it's visual appearance of what is happening inside the UAV and fast live streaming. The latency (or lag time) tends to be in the 5-20 seconds range with HLS live streams. In contrast, using rfb protocol, the same live video or audio reaches to the audience instantly.

Declaration by author

This thesis is composed of My original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution of others to My thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, financial support and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of Thesis.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis and have sought permission from co-authors for any jointly authored works included in the thesis.

Shakur Mahmood
ID: 22-92338-1

Approval

The thesis titled **"Controlling a UAV Using Remote Framebuffer (RFB) Protocol"** has been submitted to the following respected members of the board of examiners of the department of computer science in partial fulfillment of the requirements for the degree of Master of Science in Computer Science on **(28/8/2023)** and has been accepted as satisfactory.

Afsah Sharmin (Supervisor)
Associate Professor
Department of Computer Science
American International University-Bangladesh

Dr. Afroza Nahar (External)
Associate Professor
Department of Computer Science
American International University-Bangladesh

Professor Dr. Tabin Hasan
Head (Graduate)
Department of Computer Science
American International University-Bangladesh

Mashiour Rahman
Sr. Associate Professor & Dean-in-charge
Department of Computer Science
American International University-Bangladesh

Publications included in this thesis

No publications included.

Submitted manuscripts included in this thesis

No manuscripts submitted for publication.

Other publications during candidature

No other publications.

Research involving human or animal subjects

No animal or human subjects were involved in this research.

Contributions by authors to the thesis

List the significant and substantial inputs made by different authors to this research, work and writing represented and/or reported in the thesis. These could include significant contributions to: the conception and design of the project; non-routine technical work; analysis and interpretation of research data; drafting significant parts of the work or critically revising it so as to contribute to the interpretation.

Shakur Mahmood
ID: 22-92338-1

	Contribution (%)
Conceptualization	100 %
Data curation	100 %
Formal analysis	100 %
Investigation	100 %
Methodology	100 %
Implementation	100 %
Validation	100 %
Theoretical derivations	100 %
Preparation of figures	100 %
Writing – original draft	100 %
Writing – review & editing	100 %

Acknowledgments

I am thankful to my thesis supervisor Afsah Sharmin for her constant guidance and support. This thesis would not have been possible without her immense support, knowledge, guidance and patience. Her constant motivation helped me a lot to complete this thesis.

I would also like to thank my board of examiners, Dr. Afroza Nahar, Professor Dr. Tabin Hasan, Dr. Mashiour Rahman, and Dr. Carmen Z. Lamagna for helping me throughout my thesis. Their informative discussions, feedback, and cooperative nature have motivated and guided me.

Dhaka, Bangladesh
August 28, 2023.

Shakur Mahmood
ID: 22-92338-1

Keywords

Drone, UAV, Remote Framebuffer protocol, client-server model, cell phone tower, raspberry pi, ESC, BLDC motor, python pulse width modulation, microprocessor.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Flowcharts

List of Circuit Diagrams

List of Photos

List of Symbols and Abbreviations

Symbols	
P	Power
Q	Torque
f	Produced thrust
A_p	Area swept by the rotor
V	Voltage
I	Current
ρ	Density of air
G	Gas constant
T	Temperature in kelvin
g	Gravitational force
L	Lapse rate
h	Height
Δ	Difference

Abbreviations	
GSC	Ground Control Systems
UAV	Unmanned Arial Vehicle
RFB	Remote FrameBuffer
BLDC	Brushless DC
LIPO	Lithium Polymar
PWM	Pulse Width Modulation
FC	Flight Controller
FTP	File Transfer Protocol
SMTP	Simple Mail Transfer Protocol

HTTP	Hypertext Transfer Protocol
ESC	Electronic Speed Controller
VHF	Very High Frequency
HF	High Frequency
VNC	Virtual Network Computing
GUI	Graphical User Interfaces
TCP	Transmission Control Protocol
IP	Internet Protocol
MAS	Multi-Agent System
MARL	Multi-Agent Reinforcement Learning
SAGIN	Space-Air-Ground Integrated Networks
QoE	Quality-of-Experience
CRAN	Cloud Radio Access Network
ESN	Echo State Network
BS/BSs	Base Station(s)
HAP	High Altitude Platform
CSI	Channel State Information
IAB	Integrated Access and Backhaul
BH	Backhaul
SU	Secondary User
PU	Primary User
SNR	Signal-to-Noise Ratio
RPM	Rotations Per Minute
BPS	Bits Per Second
RTT	Round Trip Time
SSH	Secure SHell
WebRTC	Web Real-Time Communication
RTMP	Real Time Messaging Protocol
RTSP	Real Time Streaming Protocol

Chapter 1

Introduction

The client-server model is a distributed application structure that divides workloads between service requesters, also known as clients, and resource or service providers, known as servers. A distributed system is one whose parts are spread across numerous networked computers and which coordinate and communicate by sending messages to one another. An unmanned aerial vehicle's (UAV) communication protocols can be divided into 3 parts. These are TX protocols, RX protocols and ESC protocols. TX protocols work in between radio transmitter and radio receiver. Radio transmitter is controlled by the user. RX protocols work in between radio receiver and flight controller. And ESC protocols work in between a flight controller and ESCs. TX protocol is wireless. Other 2 protocols are wired. This thesis addresses the challenges of TX protocols and proposes an alternating way of communication between the user and the ESCs.

1.1 Problem Statement

In TX protocol, the range of control is limited by the distance between the receiver inside the drone and the transmitter. When the receiver inside the drone is so far away from the transmitter that it can no longer clearly receive signals from the transmitter, this is called the limit of the control range or limit of the radio link. Typically, this range is about 1 kilometer in normal conditions. Radio wave is good for long distance communication but stable communication actually depends on the surrounding other factors like the strength of the devices, weather condition, geographical location etc. For stable long distance communication, previously established and proven communication techniques are best like the communication between the cell towers and our mobile phones. If we can connect our FPV drones with the cell towers then we can communicate with our drones very efficiently and effectively as well. So, the following questions arises-

1. *"Can one connect a UAV with the cell phone towers and how?"*
2. *"Is it efficient and effective to communicate with the UAV through the cell phone towers?"*
3. *"Is it possible to use client-server model to communicate with the UAV from the ground station? If possible then how?"*
4. *"Is it efficient and effective to communicate using client-server model?"*
5. *"Is it possible and effective to control a UAV using RFB protocol?"*
6. *"Can one use RFB protocol to get access of a UAV from a long distance?"*

1.2 Motivation

Creating an UAV is expensive. It is due to the high cost of materials and components, significant investment in research and development, regulatory requirements and certifications necessary to manufacture and operate drones safely, and limited competition in the market. The technology used in drones is expensive, and drone companies have to spend a lot on research and development. Government regulations also add to the cost of drones. In this context, my approach is to reduce the building cost of a drone. More specifically, to make and control a drone using lowest possible equipment but which will fulfill a users demands.

Another fact that motivated the research is the complex structure of a drone. For an average person it is difficult to understand how a drone works and which component is doing which work and how they are working. On the other side, assembling the components properly is another difficult task. The drone model that is represented in this research is so simple that any average person will understand how a drone works and which component is responsible for which work and assembling the components are also much easier than the other known ways.

Lastly, it is inefficient to communicate with an UAV directly from the radio transmitter of the user from a long distance. The radio transmitter of the user have limited distance range.

These facts motivated the thesis work and inspired to solve the issues.

1.3 Scope and Contribution

This research focuses on the communication method between a flight controller and the ground station. A ground station is a system that allows us to control the UAV remotely. Researchers have proposed many ideas on how to connect the drone with the ground stations but all have the same basic principle like using a radio transmitter and a receiver. But we are currently living in the era of internet. Over two third of the population of the world are connected to the internet [1]. So, it would be a great idea if we can connect our drones to the internet. And to communicate with the drone through internet this thesis is suggesting to use client-server model. Using this model widens the flexibility of controlling a drone and as client-server model works on TCP/IP protocol, it also secures the communication. Controlling a UAV using client-server model is not a new idea. But controlling a UAV using RFB protocol is completely a new idea. This thesis will show the possibility of creating such UAVs and the consequences of using such UAVs.

As mentioned before that making a drone is costly. So, researchers do not want to test and propose new ideas on drones communication process frequently. In this regard, this research will contribute in overcoming the obstacles in long distance communication between a drone and a ground station.

1.4 Objectives and Outcomes

The objectives of this thesis are as follows:

1. To design a drone structure that is capable of using RFB protocol.
2. To find a way to connect the drone to the cell tower.
3. To write a programme to control the drone using PWM (Pulse Width Modulation).
4. To compare this drone model with the traditional drone structure.
5. To compare the most common live streaming protocols with the RFB protocol as a live streaming protocol.

The Outcomes of this thesis are as follows:

1. A drone that is controllable using RFB protocol.
2. A communication method that is capable of communicating with a UAV from a long distance.
3. An algorithm to run a UAV using pulse width modulation, threading, multiprocessing and opencv.
4. Proof of efficiency and cost effectiveness.

1.5 Thesis Outline

The body of this thesis is structured around the four critical contributions outlined above. Chapter 2 provides an overviews and surveys of different types of protocols to control a drone like- TX protocol, RX protocol, ESC protocol and other communication methods with the drone. It gives a more detailed background on previous works on client-server based method of communication with the drone. Chapter 3 discusses the experimental methodology, discusses various properties of client-server model based communication with a drone and also provides proof that this technique is possible to implement. And it also includes the algorithms this thesis has designed, the flowchart and the circuit diagram. Chapter 4 presents the results of the practical implementation, comparison of this technique with the traditional techniques, debunking the myth of using an ESC and Chapter 5 describes how the contributions have been identified, quantified, and addressed the challenges of achieving high performance communication with a UAV from a long distance. It shows the cost comparison between the traditional way of making a quadcoter and using RFB protocol, and the limitations of this research work. This section will show It further identifies vital future directions for research. Last but not least, Chapter 6 concludes the thesis.

Chapter 2

Literature review

This chapter provides background information on the client-server model in drone controlling basics to place the research contributions in context. This chapter starts with a brief introduction to client-server model. Section 2.1 outlines the communication techniques in drones terminology. Section 2.2 provides detailed background on studies of drone communication techniques for long distance. Section 2.3 discusses the General Review and remaining challenges of Related Works.

2.1 Terminology

2.1.1 Client-Server Model

Client-server is a system where it has a request sender and a request accepter to share information **[2]**. Request sender can also work as a request accepter and request accepter can work as a request sender. It allows many users to have access to the same database at the same time **[3]**. The fact that client-server systems are used almost daily for various applications is why they have grown to be so popular. Some of the common protocols used by clients and servers to communicate with one another are: File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) and Hypertext Transfer Protocol (HTTP). Thus, client-server system can be defined as a software architecture made up of both the client and server **[2]**. There are a lot of advantages of client-server model but the main advantages of it, which are necessary for this thesis are as below-

1. Centralized system with all data in a single place.
2. Cost efficient requires less maintenance cost and data recovery is possible.
3. The capacity of the client and servers can be changed separately.
4. It splits the processing of application across multiple machines **[4]**.
5. It allows easier sharing of resources from client to servers.

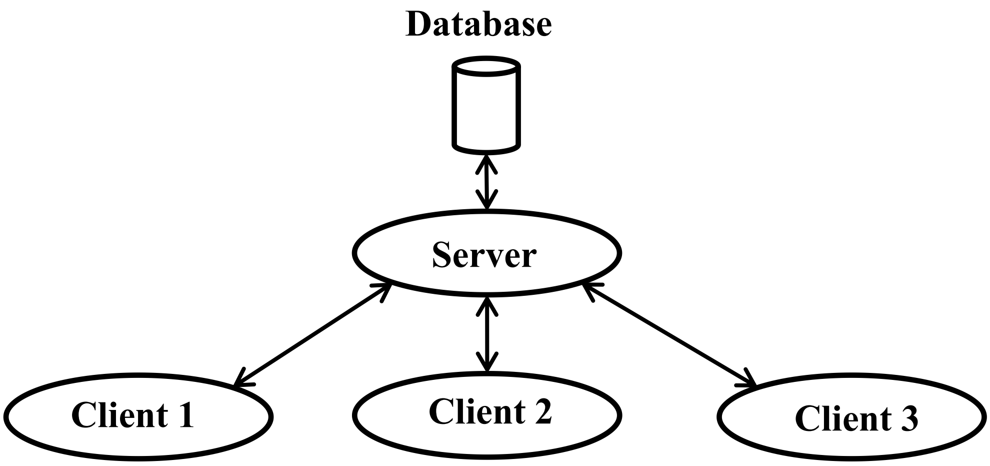


Figure 2.1.1: Interprocess communication among clients and server.

2.1.2 Drones Communication System (TX Protocols)

Communication protocol used in an UAV is divided into 3 groups-

TX Protocols– communication between radio transmitter and radio receiver
RX Protocols– communication between radio receiver and flight controller
ESC Protocols– communication between flight controller and electronic speed controllers (ESCs)

Each of these group has different kind of protocols set by different manufacturers. Some of the most commonly used protocols and necessary for this thesis are discussed below-

TX Protocols-

The communication between radio transmitter and receiver is wireless. Most radio transmitter manufacturers have their own proprietary TX protocols unless it's an open source radio system. Some of the common manufacturers TX protocols are-

Frsky’s TX Protocols

Manufacturer Frsky has two TX protocols named ACCST and ACCESS. ACCST protocol is for their relatively older series radio signal receivers like the model D8, D16, LR12 **[5]**. ACCESS is their latest protocol.

Spektrum’s TX Protocols

Manufacturer Spektrum has also two TX protocols named DSM2 and DSMX. DSM2 protocol is for their relatively older series radio signal receivers. DSMX is their latest protocol **[6]**.

The DSM2 signal has a reputation for withstanding noise, interference, and transmissions from other transmitters using the same frequency. At startup, it also locates a backup frequency in case the primary frequency fails. This significantly reduces the likelihood of signal loss, but if both channels stop working, you might still lose the connection.

DSMX was built upon and enhanced from DSM2, which also employs the same encoding method. The DSMX signal can switch to a different frequency channel in the event of a cut out within a few milliseconds. So, one wouldn't even notice the glitch.

2.1.3 Drones Communication System (RX Protocols)

The communication occurs in TX protocol is wireless but RX protocol is wired. It is the communication protocol between a radio signal receiver and a flight controller (FC). It is desirable that the protocol have low latency. Low latency means the drone will response quicker.

Some RX protocols are used universally by the different manufacturers but some are made specially for the manufacturers products. Here are some common RX protocols-

PWM- Pulse Width Modulation

PWM, which stands for pulse width modulation, combines the characteristics of both digital and analog signals. The average signal pulse lasts between 1000 and 2000 microseconds. This is the most typical, fundamental and universally used radio control protocol. The time before the flight controller was first created, the receivers used a standard PWM signal to directly control the ESC [7]. As this thesis have used python pulse width modulation, it is important to discuss the PWM in brief but which fulfills the thesis need.

In essence, pulse width modulation is a square wave with variable high and low times. We can say that it is a method of harmonically supplying current [8]. The following figure displays a fundamental PWM signal:

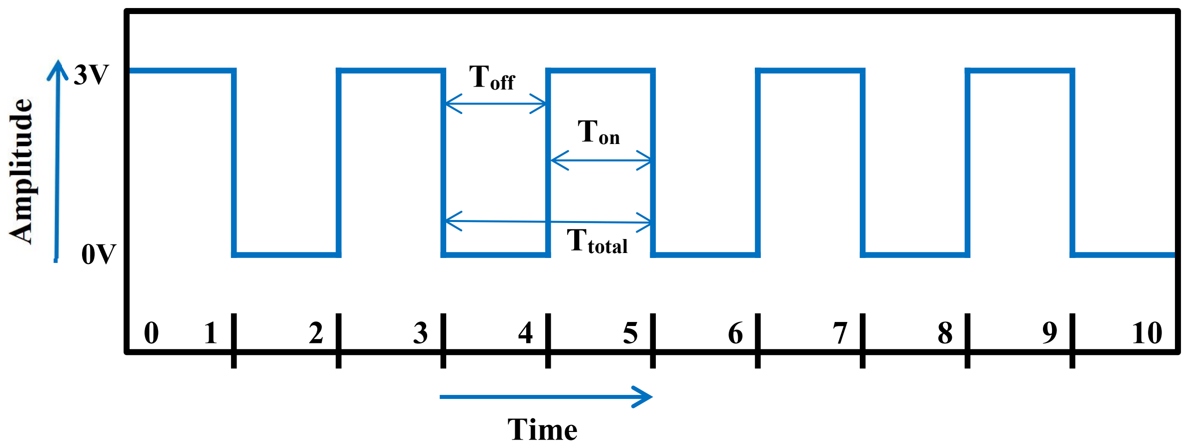


Figure 2.1.3: A basic PWM signal.

Several crucial PWM-related terms include:

On-Time (T_{on}) is when the signal is strong or the current is flowing.

Off-Time (T_{off}) is the period of time when there is no current or a weak signal.

Period (T_{total}) is shown as the PWM signal's on- and off-time added together.

$$T_{total} = T_{on} + T_{off} \qquad \text{(i)}$$

Duty Cycle (%) is the percentage of time that current is flowing during the PWM signal period **(T_{total})**. So,

$$\textbf{duty cycle} = T_{on} / T_{total} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \text{(ii)}$$

For example, in figure 2.1.3, duty cycle is 50%. It means, 50% time of the period (T_{total}), current is flowing. Change of duty cycle increases/decreases the time of flow of current and affect the output.

Frequency of PWM determines how quickly a PWM finishes a period **(T_{total})**. The formula below can be used to determine PWM's frequency:

$$\textbf{Frequency} = 1/\text{Time Period } (T_{total}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \text{(iii)}$$

PPM – Pulse Position Modulation

It is a widely used RX protocol. Other names for PPM include CPPM and PPMSUM. This is a succession of variously modulated PWM signals that are sent one after the other on the same wire **[9]**.

Serial Protocols

A serial protocol is a digital lossless protocol that supports multiple channels on just 3 wires (signal, power, and ground). Serial protocols, in contrast to PPM are entirely digital, consisting only of ones and zeros. As the name suggests, serial protocols require a serial port on the flight controller (also known as UART). Different UAV manufacturers uses different kinds of serial protocols. For example, SBUS or, serial BUS is commonly used by the manufacturer Futaba and the manufacturer FrSky. CRSF is developed by the manufacturer TBS for their Crossfire RC system. IBUS is used by the manufacturer flysky. Serial protocol XBUS is used by the manufacturer JR.

2.1.4 Drones Communication System (ESC Protocols)

Like RX protocols, ESC protocols are wired. It is the communication protocol between a flight controller (FC) and an electronic speed controller (ESC). The FC has to communicate to the ESC's at a much faster rate than to communicate to the radio receiver. It is because apart from taking commands from the pilot, the FCs get lots of data continuously from various sensors such as gyro and accelerometer at a much faster rate. Some of the well known ESC protocols are PWM, oneshot, multishot, proshot, dshot.

2.1.5 Electronic Speed Controller (ESC)

An electric propulsion system cannot function without the electronic speed controller (ESC). To create thrust, spacecraft use electric propulsion systems. They can be divided into two categories: steady (constant firing for a set period of time) and unsteady (pulsed firings building up to a desired impulse) **[10]**. Electronic speed controllers (ESC) enable a connection between the power source and the motor. The ESC controls the motor's power output and rotational speed in response to the operator's throttle commands. A brushless DC motor is typically driven by an ESC, which converts DC electricity from a battery to three-phase AC. This involves high frequency electronic switching, which could lead to sizable efficiency losses **[11]**. Here, the differences among a stepper motor, a brushed DC motor and a brushless DC (BLDC) motor should be clarified. All of them uses the same electromagnetic technique. The main difference is in the number of phases, Stepper motors generally uses higher number of phases to maximize holding torque. Brushed DC motors uses relatively lower number of phases. And BLDC motors uses lowest number of phases to generate smooth torque between steps.

2.1.6 Cell Phone Tower

Cell phone towers are buildings placed on particular plots of land with the purpose of housing wireless tenants. Cell towers are used by wireless tenants to roll out different technologies, such as mobile data, television, and radio, to subscriber bases. Typically, wireless carriers or, tower companies construct cell towers.

Cell towers are built to make sure that mobile signal is available even in remote places. Cell towers are tall structures that are typically built on high ground, such as hills or mountains. This is because they need to be in a high location to provide good coverage for a large area. Cell towers are also often built on tall buildings, such as skyscrapers. However, these structures are often subject to harsh environments.

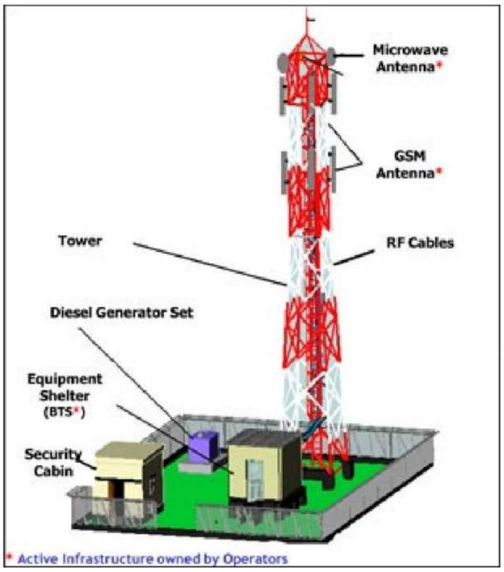


Figure 2.1.6: Different parts of a cell phone tower.

It is important to know that 'cell sites' (which are antennas placed on rooftop by the network providers) are different from the cell phone towers. there can be multiple cell sites (and multiple tenants) under any one tower **[12]**. Some important components of a cell site are- whip antenna, antenna array, port holes, panel/antenna, microwave dish.

Cell sites are structured in a manner that is essentially the same whether they are handling basic phone calls or 12Mbps WiMax data. In essence, a cellphone or modem radio communicates with the closest towers to say, "I'm here!" When a person makes a call or logs on, their phone then transmits a radio message that the antenna array then receives. The call is transmitted down a wire or fiberoptic line to the wireless access point, which is linked to a multi-port switch. Here is an important fact to mention, that the mobile phone users or, other end users do not notice is a lot of portion of this wireless communication is actually wired communication as the cell phone towers are connected through fiberoptics. This is done to make our communication more stable.

2.1.7 VHF radios

Very High Frequency (VHF) radio calls are the most typical form of communication in aviation. In simplest terms, the transmitting station sends a signal that is picked up by the receiving station after traveling in a straight line. As radio waves travel in straight lines, their range is constrained by the terrain they may encounter, such as hills and mountains. A VHF signal's range is determined by the height from which it is sent as well as the height of the receiving station. The distance will be relatively short if the sender and receiver are both on the ground. The signal's range is significantly increased if both stations are in the air. The distance between an aircraft and a ground-based station, such as Air Traffic Control, is in the middle. The following equation can be used to determine how far a VHF signal will travel:

VHF signal distance in nautical miles = 1.23 x (√ height of transmitting object in feet + √ height of receiving station in feet) ………………………………………….. (iv)

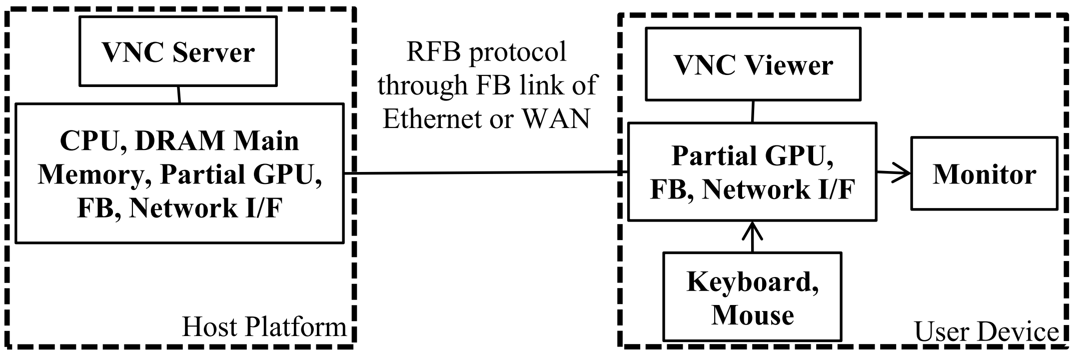
VHF radios are great when flying over areas where there are regular radio towers to send and receive signals. However, when flying over the oceans, VHF comms are not really an option. In this case, High Frequency (HF) radio signals help. They aren't as strong as VHF signals but are actually able to travel much farther. The radio environment changes as one rises higher above the ground. It might become more difficult to connect UAVs that are moving through the air using cellular networks **[13]**.

2.1.8 Autopilot

An autopilot is a device that autonomously directs ships, spacecraft, and aircraft in a predetermined direction.It may also be used to describe the automatic control that such a device offers. Although the terms "flight controller" and "autopilot" are frequently used synonymously, they can also mean different things. A complete system known as an autopilot enables your drone to fly to waypoints, modes, or missions on its own. The only thing that will keep the aircraft stable by adjusting the control surfaces is a flight controller. A flight controller is typically a part of an autopilot, but not all flight controllers have autopilot functionality. For example, Pixhawk FC is a type of autopilot system for drones and other vehicles **[14]**. It is based on the PX4 open-source software and hardware platform. It can perform various tasks such as GPS navigation, obstacle avoidance, and telemetry.

2.1.9 RFB protocol

The protocol used by Virtual Network Computing (VNC) and its offshoots is called RFB (Remote FrameBuffer). For remote access to graphical user interfaces (GUI), it is an open, basic protocol. It is applicable to all windowing systems and programs because it operates at the framebuffer level, which includes Microsoft Windows, macOS, Linux, Raspbian OS, and other operating systems. Here, it's important to note that RFB is a client-server protocol while VNC is platform-independent. As a result, multiple clients can simultaneously connect to a VNC server using the RBF protocol. As of right now, all VNC implementations are made up of two pieces of software: a VNC viewer (client) and a VNC server [15], as seen in figure 2.1.9.



Any PC, Server or Workstation — Any PC or Workstation

Figure 2.1.9: VNC and Remote-Desktop belong to the FB approach.

RFB is a ‘thin client’ protocol. The RFB protocol was designed with the goal of imposing the fewest possible client requirements. This makes it possible for clients to function on the broadest range of hardware while also simplifying the process of implementing a client. Additionally, the protocol makes the client stateless. The state of the user interface is preserved even if a client disconnects from one server and then reconnects to it.

RFB protocol has 3 stages- handshaking, initialization and normal operation. In the handshaking stage, the client and the server agree on the protocol version and security type. In the initialization stage, they exchange information about the framebuffer size, pixel format and name. In the normal operation stage, they send messages to update the framebuffer and handle user input. This protocol can work over any reliable transport, i.e., TCP/IP. It is based on the concept of a remote framebuffer, which is a representation of the pixels on the screen. The server sends updates to the client whenever the framebuffer changes, and the client sends back control events such as keyboard or, mouse input.

2.1.10 A UAV's Working Principle

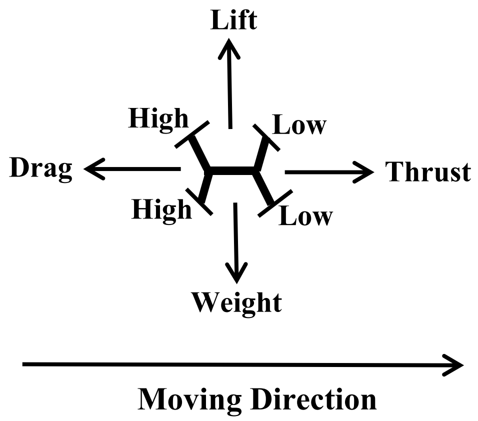


Figure 2.1.10: Four forces occur in UAVs **[16]**.

For a UAV to fly effectively and steadily, drone aerodynamics is essential. Designing and using drones for a variety of applications requires a fundamental understanding of the laws of lift, thrust, drag, and weight **[17]**. The drone's wings or rotors produce lift, which is an upward force that defies gravity. Wings on fixed-wing drones (drones with just one rotor) produce lift as they move through the air by creating a difference in pressure between their upper and lower surfaces. Multirotor drones, like quadcopters, generate lift through the rotation of their propellers, which force air downward while also creating an opposing force that lifts the drone.

On the other hand, thrust is the forward force that moves the drone through the air. Thrust is typically produced by a propulsion system in fixed-wing drones, such as a jet engine or motor-driven propeller. Multirotor drones use the same propellers that create lift to produce thrust by varying the speed of rotation of the propellers.

Drones experience drag and weight in addition to lift and thrust, which can have an impact on performance. Drag is the opposition to the drone's forward motion that it faces while flying through the air. Increasing efficiency and range requires minimizing drag. Drag can be reduced by using streamlined designs and smooth surfaces.

Weight is the downward-moving force of gravity. In order to maintain stable flight, the lift generated by the drone's wings or rotors must be greater than its weight. The device's overall weight is comprised of the drone, the payload (which may include cameras or sensors) and the battery's weight.

All the terms discussed here, can be putted in a table along with their main using purposes like the table 2.1.

Table 2.1: Terms those are discussed in this chapter.

Term	Main using purpose
Client-Server Model	Works with a request sender and a request accepter.
TX Protocols	Communication between radio transmitter (TX) and radio receiver (RX).
RX Protocols	Communication between radio receiver (RX) and flight controller (FC).
ESC Protocols	Communication between flight controller (FC) and ESC.
Electronic Speed Controller (ESC)	Electric propulsion systems are used in spacecrafts to generate thrust. ESCs are important part of it.
Cell Phone Tower	To make sure that mobile signal is available even in remote places.
VHF radios	Very High Frequency (VHF) radios are used to communicate with the aereal vehicles.
Autopilot	It provides automatic control to the UAVs.
RFB protocol	It is the protocol used in VNC for remote access to GUI.

2.2 Prior studies on drone communication techniques for long distance

UAVs have existed since the turn of the 20th century. They were initially used for military operations, but after around 2010, as electronic technology shrunk, became more affordable, and became more effective, the cost of cameras and sensors fell, and battery life increased, they started to be used more frequently. Once limited to using manned aircraft or satellites to observe the earth from above, scientists are now able to expand, develop, and improve their research thanks to drones.

But doing research on drones is still expensive due to it’s high equipment price. There is also the risk of losing a drone through accidents during research. And there is less amount of funding to do research in this area. That is why researchers are not eager to do research on this topic. There are many other factors that affect the amount of researches on drone technology, such as security, safety, regulation, privacy, and technical challenges. But in recent years, drone technologies become so much popular which draws the researchers attention.

Abhishek et al., 2020 [18], along with his team did a survey on current communication and networking technologies available for the UAVs. They explored techniques such as machine learning and path planning to enhance existing drone communication methods, encryption and optimization techniques for ensuring long–lasting and secure communications, as well as for power management. The intricate interplay between UAV, advanced cellular communication, and internet of things constitutes one of the focal points of this paper.

IOT services and protocols devised for the drone communication

Figure 2.2(i): A typical scenario of the internet as the medium for drone communication according to Abhisek et al., 2020 [18].

Yang et al., 2023 [19], worked on Graph Attention Network-Based Partially Observable Mean Field Multi-Agent Reinforcement Learning for UAV Swarms. By constructing the multi-UAV cooperation problem as a multi-agent system (MAS), the cooperative decision-making among UAVs can be realized by using multi-agent reinforcement learning (MARL). Following this model, the authors focused on developing partially observable MARL models that capture important information from local observations in order to select effective actions.

Kangunde et al., 2021 [20], reviewed drone control techniques available with respect to real-time and discusses the possible implementation of real-time flight control system to enhance drones performance. The authors of this paper proposed that, a UAV's operational environment is highly dynamic due to unpredictable changes in weather conditions affecting the air space. For drones to be reliable, their flight controllers must adapt to these environmental changes in real-time.

Arya et al., 2023 [21], designed and implemented a ground-to-UAV (G2U) communication network with an aim to minimize end-to-end transmission delay in the presence of interference in Space-Air-Ground Integrated Networks (SAGIN). They applied Fano's inequality to characterize the transmission delay and derived the tight upper bound for the G2U uplink channel's capacity in the presence of interference, noise, and potential jamming. A tight lower bound on the transmit power was also obtained as a function of the UAV's location information, subject to the reliability constraint and the maximum delay threshold. Additionally, they also proved that there exists an optimal height that minimizes the end-to-end transmission delay in the presence of interference.

Now, coming to the point of UAVs into cellular networks, there have been a lot of interests in incorporating a UAV communication system into the present and future cellular networks in recent years [18]. There have been numerous attempts to combine UAVs with cellular networks ever since the early 2000s. Wzorek et al., 2006 [22], demonstrated a test network built using GPRS technology between a UAV and a ground operator. The concept hasn't been further developed or made into a product, though, because of technological constraints. Their results showed that GSM network infrastructures provide a useful means for communicating with UAVs, especially in urban areas. Their research has a similarity with this thesis which tries to communicate with a UAV through cell towers but a lot of dissimilarities as well.

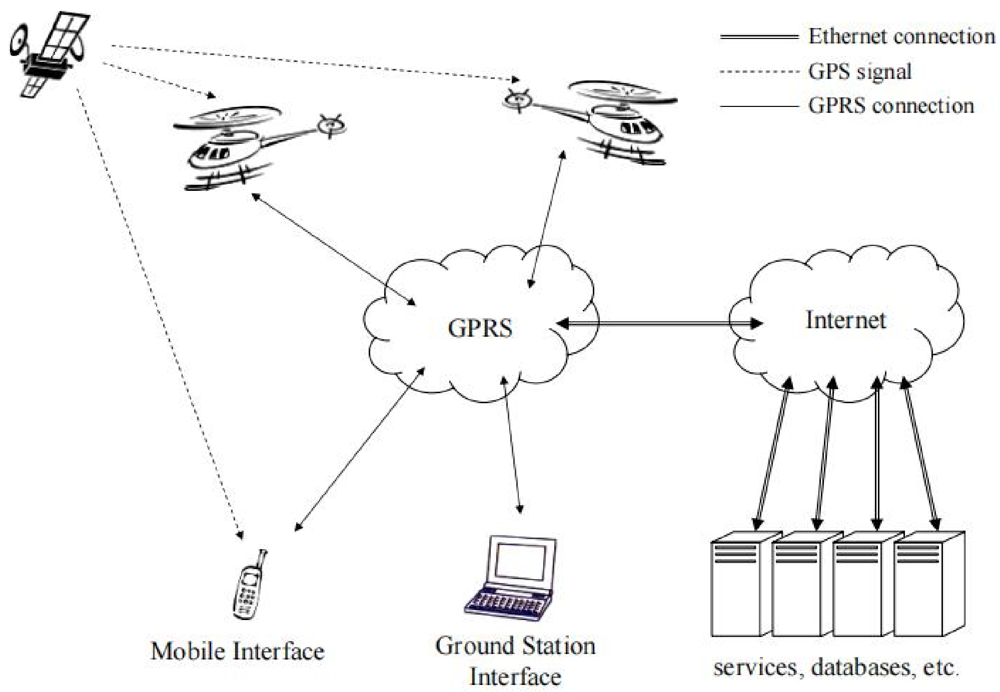


Figure 2.2(ii): Network configuration developed and used by Wzorek et al., 2006 [22].

First of all, they suggest that TCP/IP protocol has performance issue. That is why they used UDP protocol to communicate with the drone through GPRS internet. But UDP protocol is unsecured. This thesis uses client-server model to communicate with the drone which uses TCP/IP protocol.

And they had to use a lot of equipment to communicate with the drone through cell phone towers from the ground station which makes sense because the research was done during 2005-2006. At that moment technological advancement was not achieved like today. And those equipment were expensive as they tried to use the latest available technologies at that moment.

On the contrary, this thesis tries to make the total system so simple that any average person will able to understand the structure of the UAV that is used in this thesis and it's working principle (it will discussed in detail in the next chapter). This thesis is using mainly 4 types of equipment- a power source, a raspberry pi, ESCs and BLDC (brushless DC) motors. And the total system is cost effective as well. Moreover, they have implemented their idea on a helicopter with a 21 hp two-stroke engine but this thesis uses a quadcopter drone which uses 4 BLDC motors. The reason behind choosing a quadcopter is it is easier to control a quadcopter than a UAV with fixed-wing aircraft or, a bicopter or, a tricopter. And the reason behind not choosing more than 4 motored UAV is to keep the cost of the project bearable.

Lastly, A hybrid deliberative/reactive software architecture had been developed for their UAV platform and had also been used in a ground robot. The software implementation is based on CORBA (Common Object Request Broker Architecture). In contrust, this thesis has used a python code which has written for this thesis and has been installed in the drone and the ground station will only access that code through the internet (this thesis has used 4g), will run the code and will give user inputs. In this research, as a ground station any type of smart device or, desktop with internet connection can be used that supports client-server model. So, one of the key approach of this thesis is to create a totally programmable drone using client-server model.

There are some more approaches on interacting UAVs into cellular networks, i.e., Mingzhe et al., 2017 **[23]**, worked on the problem of proactive deployment of cache-enabled UAVs for optimizing the quality-of-experience (QoE) of wireless devices in a cloud radio access network (CRAN). The objective of this problem is to maximize the users QoE while minimizing the transmit power used by the UAVs. It is formulated as an optimization problem. An algorithm on the machine learning framework of conceptor-based echo state networks (ESNs) was put forth to address this issue.

Mozaffari et al., 2017 [24], investigated the efficient deployment and mobility of multiple UAVs, used as aerial base stations to collect data from ground Internet of Things (IoT) devices, was investigated. they proposed a model for jointly optimizing the three-dimensional (3D) placement and mobility of the UAVs, device-UAV association, and uplink power control to enable reliable uplink communications for IoT devices with a minimum total transmit power.

UAVs can be used as aerial base stations (BSs) to deliver broadband wireless connectivity during temporary events, at hotspot areas, or after disasters that may destroy existing communication infrastructure. Sudheesh et al., 2017 [25], analyzed the sum-rate for high altitude platform (HAP) drones with a multiple-antenna tethered balloon as a relay. They proposed a method to achieve maximum sum-rate of a HAP without acquaring channel state information (CSI).

One of the limitations in drone base stations implementation is the availability of reliable wireless backhaul link. Kalantari et al, 2017 [26] investigated how different types of wireless backhaul offering various data rates would affect the number of served users. They introduced two approaches, namely, network-centric and user-centric, and the optimal 3D backhaul-aware placement of a drone-BS was found for each approach.

Fouda et al., 2018 [27], worked on UAV-Based in-band Integrated Access and Backhaul for 5G Communications. They proposed a system that utilizes UAVs as drone BSs in an IAB (Integrated Access and Backhaul) scenario for 5G networks. They also proposed a model to find the optimal precoder design for BH (Backhaul) links, user-base station association, UAV 3D hovering locations and power allocation for forward link transmissions. They utilized this model to optimize the network performance from both large and small-scale perspectives.

Lokman et al., 2017 [28], worked on controlling UAVs using 5G MIMO systems. They proposed a special linear precoding scheme to enable the secondary user (SU) to exploit the primary user (PU) free eigenmodes.

Rupasinghe et al., 2017 [29], tried to identify optimum hovering locations for UAV-BSs equipped with multi-antenna arrays. They considered a scenario with two UAVs each with a single user attached and developed their proposed scheme to achieve angular domain user separation and signal-to-noise ratio (SNR) maximization at each user and evaluated the performance through simulation.

Yaliniz et al., 2016 **[30]**, highlighted the properties of the drone-cell placement problem and formulated it as a 3-D placement problem with the objective of maximizing the revenue of the network. Drone cells are UAVs equipped with base stations (BSs). Following some mathematical adjustments, they developed an equivalent mixed integer non-linear quadratically constrained optimization problem and suggested a computationally effective numerical solution.

It is noticed that a lot of these research on UAVs has done to enhance backhaul connectivity and providing extended coverage areas, i.e.- Abdel-Malek et al., 2018 **[31]**, Mozaffari et al., 2016 **[32]**, Mozaffari et al., 2019 **[33]**.

Zeng et al., 2018 **[34]** and Guang et al., **[35]**, discussed the advantages of cellular connection in a UAV. They discussed the advantages on the points like ubiquitous accessibility, enhanced performance, ease of monitoring and management, robust navigation and cost-effectiveness. They also discussed the complexities behind connecting a cellular network in a UAV. Guang et al., 2018 **[35]**, mentioned the point that Mobile networks stand ready nationwide to power flying drones beyond ground devices. They also created a prototype drone to check the signal strength with the cell tower at different height level by measuring RSRM (Received Signal Strength Indicator), SINR (signal-to-noise ratio) and latency performance.

A prototype LTE-UAV integrated network's field data was presented in 2016 by China Mobile Research Institute and Ericsson. In this prototype, they go into greater detail about the advantages that mobile technologies can bring to the drone ecosystem, list the essential capabilities that drone applications need, and examine the service needs of mobile networks. Further research was done on this plan, and the third-generation partnership project (3GPP) released several proposals that looked into the viability of using an LTE network for aerial vehicle service **[36]**.

Euler et al., 2019 **[13]**, shared some of their findings in mobility support for cellular connected UAVs by simulating mobility performance. They examined the effects on mobility performance of the corresponding changes in the radio environment with altitude. They also talked about potential improvements to the performance of aerial mobility.

A fundamental limitation of all these papers is we can not vizualise what is happening inside the UAV while in the sky although they used cellular network. Another limitation is that the authors of these papers were not eager to reduce the complexity of the total UAV setup and most of them used prebuild UAVs available in the market. An overview of these challenges is illustrated in table 2.2.

Table 2.2: Findings of different review papers.

Author	Topics Discussed and Identified Challenges
Wzorek et al., 2006 [22]	Used gprs technology using gsm network. But did not tested client-server model.
	Complex setup of the total system.
	Expensive
Mozaffari et al., 2017 [24]	Discussed how UAVs as base station can enable IoT devices to communicate energy efficiently.
Fouda et al., 2018 [27]	Discussed a technique to improve 5g communications using UAVs as base stations.
Zeng et al., 2018 [34]	Discussed the advantages of cellular connection in a UAV and basic structure of this communication.
	No prototype was built.
Guang et al., 2018 [35]	Used cellular connection.
	Built a prototype using data acquisition module, a flight control device, and a data processing server. However, did not discuss the total UAV structure and the equipment's using purposes.
Euler et al., 2019 [13]	Discussed the mobility support for cellular connected UAVs.
	Simulation based evaluation for mobility performance.

2.3 General Review and remaining challenges of Related Works

The critical challenges of previous works are as follows, which will be overcome by this thesis's proposed model.

- Previous primary researches have done using traditional flight controllers.
- These flight controllers are pre-designed for any special purposes like for cinema flying, for sport flying, or for autonomous flying.
- In a word, one can not customize it according to his/her need.
- Previous researchers mainly worked on effectively controlling a UAV rather than efficiently controlling it.
- Here, also no proper explains how their UAV was working in detail.
- Using traditional communication techniques (directly from the ground station to the UAV or, from the ground station to the UAV through cell towers) we can not visualize what is happening inside a UAV.
- Previous researchers did not do their research on the cost perspective.
- Those researchers did not discuss the possibility of controlling a drone world wide from anywhere in the world either efficiently or, effectively.

To address these gaps, this thesis proposes to use client-server based RFB protocol to communicate with the UAVs from the ground stations.

Chapter 3

Proposed System Based on RFB Protocol

This section presents the proposed system's architecture, methodologies, used equipment and cost, and algorithms that is used in this thesis.

3.1 Problem Formulation

To formulate the problem, lets assume a user has 2 UAVs of same model, made by the same manufacturer and the receivers in both the UAVs have a lot of channels. If two receivers having the same number of required channels are bound with one radio transmitter then both aircraft would experience the same control commands simultaneously from that transmitter.

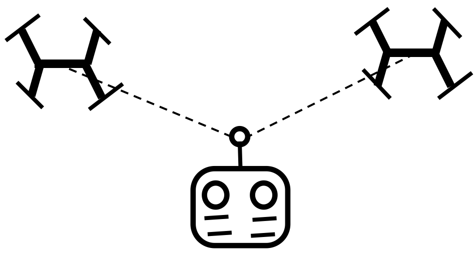


Figure 3.1: Controlling two UAVs with the same radio transmitter simultaneously.

It may cause unwanted collision with other objects. This is why it is illegal to fly or observe more than 1 aircraft by 1 operator simultaneously in the USA [37]. This situation does not only occur only for the 2 UAVs of same model and made by the same manufacturer rather it happens for more than 1 UAV, having the same number of required channels, and if they are bound with the same radio transmitter.

And last but not least, there is a range limit of TX protocols to establish communication in between the radio transmitter and the radio receiver. The transmitters that come with the drones generally operate at 2.4 GHz and have a maximum range of 1.6 km. However, the maximum value can be affected by things like obstacles or even noise blocking the signal. To deal with these situations, this article recommends using the client model to reduce conflict and expand control.

3.2 System Model

A client-server model simply generates two types of executable. The single server executable has no GUI (Graphical User Interface) but contains all of the drivers and device settings and the client executables contain the user interface that connects with the server. This enables an installation to have multiple control points for its equipment. Each client establishes a connection with the server, whose responsibility is to communicate with the controlled devices.

Numerous device protocols only permit one connection to the device, which is one of the technical reasons why client-server is so powerful. For example Serial protocol can only support one connection, or Ethernet devices support only one TCP/IP connection at a time. Before client-server, it was very difficult to scale a system beyond adding one or two UIs (User Interfaces). One can now have as many clients as they want connected to the same network. The server can be a physical, virtual, or cloud-based machine, and the client can be any device that has network access and can communicate with the server. Each client connects to the server using either TCP/IP or HTTP, which means one can extend the reach of control across the world if he/she wishes, simply by using the HTTP protocol and ensuring the server has a globally accessible (or routable) IP Address, and an internet connection. Figure 3.2(i) depicts the client-server model conceptually, illustrating the relationship between clients, servers, and devices.

To establish client-server model between two devices wirelessly, one needs to set up one device as a server and the other as a client, and needs to connect them using a wireless network protocol such as Wi-Fi or Bluetooth. To apply the client-server model in a UAV, same technique can be applied. In this case, the UAV itself should be used as a server and the ground station (GS) should be used as a client. The scenario should be similar like the figure 3.2(ii) below.

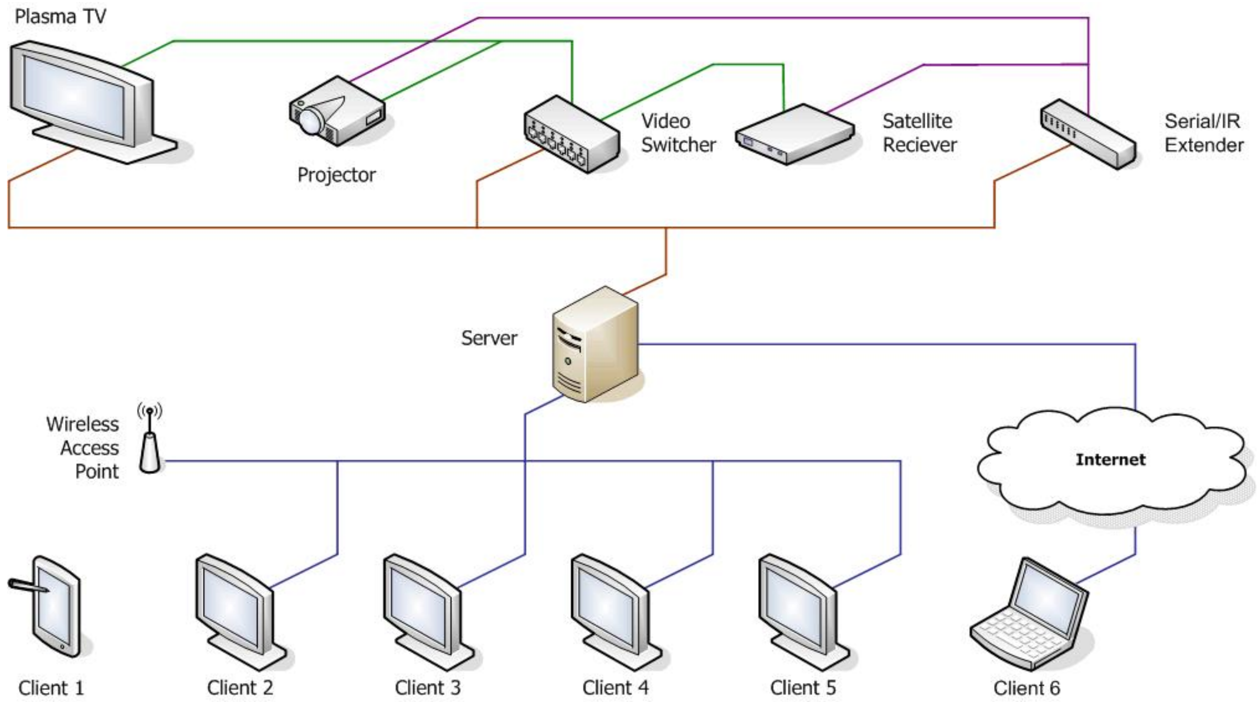


Figure 3.2(i): The client-server model.

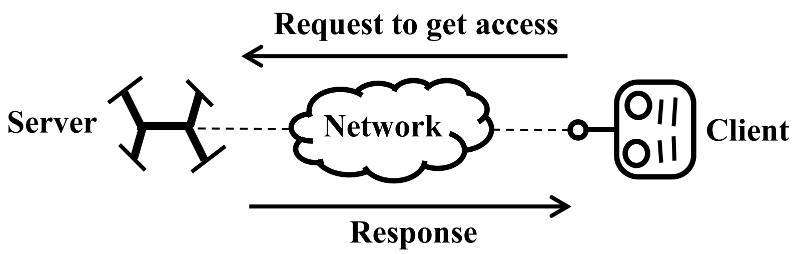


Figure 3.2(ii): Controlling a UAV using client-server model.

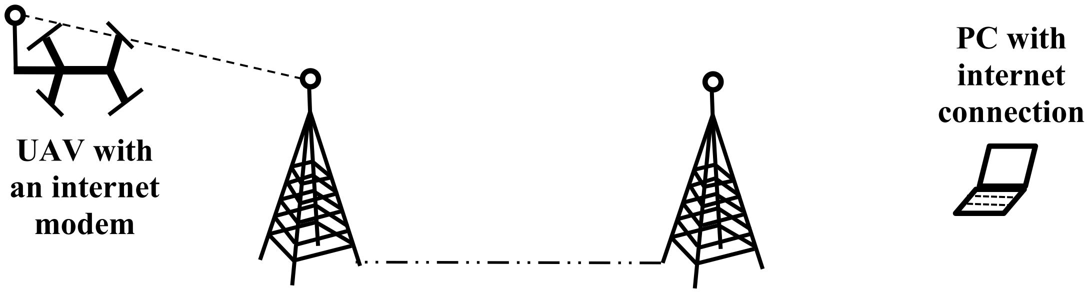


Figure 3.2(iii): Controlling a UAV using client-server model through the cell phone towers.

But this thesis will also try to control a UAV using client-server model through cell phone towers. Then the scenerio should be similar like the figure 3.2(iii). In the figure, it is noticed that cell towers communicate with each other through fiberoptic cables to make the total communication more stable.

3.3 Hardware Setup

This thesis will use a quadcopter as a UAV. All the equipment those will be used in this thesis are listed below-

- 4 BLDC motors with constant velocity (KV) of 380 KV. It is measured by the number of revolutions per minute (rpm) that a motor turns when 1 volt is applied with no load attached to that motor.
- 1 piece of 6S 60C 6000 mah lipo battery. So,
  - Nominal Battery Voltage is 22.2V.
  - Minimum Battery Voltage is 18.6V.
  - Maximum Battery Voltage is 25.2V.
  - Maximum Continuous Discharge rate is 360 amp.

  One can learn these simply by using LiPo Battery Power and Configuration Calculator **[38]**.
- 4 pieces of 7S ESCs.
- 4 pieces of 10 inches carbon fiber propellers.
- 1 piece raspberry pi 4B 8GB.
- 1 piece of lipo battery to raspberry pi charging converter to drop down the lipo battery voltage and power the raspberry pi.
- 1 piece of Huawei E3372 LTE USB stick
- 1 piece of f450 quadcopter frame.
- Some connecting wires and screws.
- And a computer with internet connection as a ground station.

To build any UAV, one needs to have some knowledge on electricity and electronics. Main critical components here are BLDC motors, ESCs and Lipo battery or power source. These are critical because one have to use the proper one that matches with the other 2 components.

At first, it is needed to be decided, with how much total weight the drone is going to take up. Then how many motors the owner wishes to use. The motors total weight load capacity can be calculated using the thrust rating of each motor. One can learn about the motor specification from the datasheet provided by the manufacturer like the datasheet of the motor used in this thesis can be found here **[39]**. The motor used in this thesis has a thrust rating of 1280 grams for a 1238 APC nylon propeller at 100% throttle. The number 1238 means propeller size is 12 inches and 3.8 inches is the pitch. Pitch is the drone propeller's theoretical traveling distance in one complete spin of 360° and throttle means speed. So, 1 motor can take up 1280 grams of weight which excludes it's own weight. Including it's own weight, the maximum weight that the motor can take up is 1280g + 58g = 1338 grams. Here, 58g is the weight of the motor itself. It is the calculation for 1 motor. So, if one uses 4 motors, then the total maximum weight that the motors can take up will be 1338g * 4 = 5352 grams. If one decreases the propeller size to 1 inch this maximum load to take up will also decrease. If he/she decreases the propeller size to 2 inches the maximum load to take up will decrease more. Material and shape of the propeller are responsible for the UAV's shaking and noise. Propellers are of 2 types. One type is for rotating clockwise and the other type is for rotating anti-clockwise. Both types are needed to be attached with the motors to spin with it to create upward force. Propeller power is calculated from the measured propeller torque, **Q** by **[40]**-

$$\mathbf{P = 2\pi nQ} \quad \text{……………………..} \quad \text{(v)}$$

And for an actuator disk model of a propeller that is not translating in the ambient air, the aerodynamic power $\mathbf{P_{aero}}$ required may be computed as **[41]**-

$$P_{aero} = \sqrt{\frac{f^3}{2\rho A_p}} \quad \text{…………………..} \quad \text{(vi)}$$

Where, $\boldsymbol{f}$ is the thrust produced, $\boldsymbol{\rho}$ is the density of the air, and $\mathbf{A_p}$ is the area swept by the rotor. This derivation is based on applying the conservation of mass and energy to a control volume containing the propeller and assumes that the flow is inviscid and incompressible.

After choosing desired motors and propellers according to the desired load to take up one has to choose proper power source for the motors. In the case of this thesis, it is lipo battery. Again, the datasheet provides the information about the voltage that is needed to run the motor, the amount of current (in ampere) that is needed at different throttle rates, and the maximum continuous current that the motor can handle **[39]**. In the case of this thesis, the chosen motor needs 6S lipo battery to run or, spin. So, the motor works properly by getting 18.6V - 25.2V current. The datasheet **[40]** is also telling that the maximum continuous current that the motor can handle is 20 amp but 7.2 amp current is needed to achieve 100% throttle. So, while choosing an ESC for each motor, these should be remembered. The ESC should be rated for higher amps than the motor's maximum current draw. A higher amp ESC will not ruin the motor, but a lower amp ESC may overheat and burn out. But to reduce the cost it is always better to match the ESC amp with the maximum continuous current that the motor can handle. In the case of this thesis it is 20 amp. Power of a lipo battery is measured by the following formula-

$$\mathbf{P = VI} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \qquad \text{(vii)}$$

Here, '**P**' is power, '**V**' is voltage, '**I**' is current. The voltage influences the power of the battery, and power has an impact on the RPM (Rotations Per Minute) of the motor, which means speed. Now, the chosen motor works properly with a 6s lipo battery. So, this thesis can choose an ESC that supports 6s lipo battery. But the practical experience during this thesis tells that the ESC can also burst in this case. The reason behind it is, the maximum voltage that a 6S lipo battery can provide is 25.2V and the nominal voltage of 7S lipo battery is 25.9V. So, 6S lipo battery can provide nearly the nominal voltage of a 7S ESC. Although, products from different manufacturers can vary but it is safe to use an ESC that supports atleast 1S more voltage. So, in the case of this thesis, the chosen ESC supports the voltage of a 7s lipo battery. And if one wants to use a 4 in 1 ESC for a quadcopter then he/she needs to choose it in such a way so that, the maximum current it can provide is 4 times the maximum continuous current that a motor can handle. So, in the case of this thesis it would be 7S 80 amp ESC if wanted. In case, one wants to use multiple lipo batteries or multiple power sources should follow the electricity laws for parallel or series circuit according to his/her need. In summary, parallel maintains the same voltage while increasing endurance; series increases voltage while maintaining the same endurance as a single battery. However, connecting two LiPo batteries with different number of cells in parallel is not advised. When two LiPo batteries are connected in parallel, even if there is a large voltage difference between them, the higher voltage LiPo will charge the lower voltage LiPo until their voltages are equal. It could turn destructive.

After choosing these 3 components properly, half of the work is done. Now, it is needed to put these components in a frame. A frame is important to apply the aerodynamic theories because it affects the drag and lift of the aircraft. This thesis uses an f450 quadcopter frame which can give enough space for 4 pieces of 10 inches propellers without colliding with each other. Aircraft operate in air which is mainly mixed up with nitrogen, oxygen. The behaviour of air depends on temperature, pressure, and density related to each other **[42]**. It can be described by the Ideal or Perfect Gas Equation of State:

$$\mathbf{P = \rho RT} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad \text{(viii)}$$

Here, **P** is the barometric or hydrostatic pressure, **ρ** is the density of air and **T** is the temperature in kelvin. **R** is the gas constant for air which is equal to 287.05 N-m/kg.kelvin. Anothor equation used in aircraft is-

$$\mathbf{\Delta P = -\rho g \Delta h} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad \text{(ix)}$$

This tells us how pressure changes as we move up or down through the atmosphere. Here, **Δh** is the change of height. Another equation that is used in aircraft is-

$$\mathbf{T_{alt} = T_{sea\ level} - Lh} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad \text{(x)}$$

Where 'L' is called the 'lapse rate'. A normal average lapse rate is 0.0065 kelvin/meter.

Now, differential form of the (vii) will be-

$$\mathbf{dP = -\rho g.dh}$$

$$\text{or, } \mathbf{dp/dh = -\rho g}$$

$$\text{or, } \mathbf{dp/dh = -(P/RT)g \quad [From\ (vi)]}$$

$$\text{or, } \mathbf{dp/P = -g.dh/RT}$$

$$\text{or, } \mathbf{dp/P = -g.dh/R(T_{sea\ level} - Lh) \quad [From\ (vii)]}$$

$$\text{So, } \mathbf{dp/P = g.dh/R(Lh - T_{sea\ level})} \qquad \ldots\ldots\ldots \qquad \text{(xi)}$$

Now, (xi) is a relationship with one variable (P) on the left and one variable (h) on the right. It can be integrated to give -

$$\mathbf{P_{alt}/P_{sea\ level} = [T_{alt}/T_{sea\ level}]^{g/LR}} \qquad \ldots\ldots\ldots\ldots \qquad \text{(xii)}$$

In a similar way, we can obtain a relationship to determine the density at any altitude -

$$\rho_{alt}/\rho_{sea\ level} = [T_{alt}/T_{sea\ level}]^{\{(g/LR) - 1\}} \quad \ldots\ldots\ldots\ldots\ldots \quad \text{(xiii)}$$

Now we have equations to find pressure, density, and temperature at any altitude until stratosphere. In stratosphere (which is situated up above 11,000 meter) remains constant. So,

$$T_{stratosphere} = 216.5 \text{ kelvin} = \text{Constant}$$

Since temperature is constant both pressure and density vary in the same manner. For stratosphere, (xii) and (xiii) become-

$$P_2/P_1 = \rho_2/\rho_1 = e^{g(h1-h2)/RT} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \text{(xiv)}$$

Here, e = 2.718. Another equation that is used in aircraft is -

$$\text{Power} = \text{(Thrust)(Velocity)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \text{(xv)}$$

The performance of an aircraft or propeller airplane is measured by power. Power and thrust are just two different ways of looking at aircraft propulsion and performance. They are directly related to each other in terms of speed. Both the thrust of the jet and the power of the propeller aircraft remain constant as the speed increases [42].

3.4 Software Setup

In the case of software perspective this thesis uses python code to directly control the BLDC motors from raspberry pi through ESCs using python PWM technique. It is a small sized code which contains just over 700 lines. Using this code this thesis will test the basic operations like up, down, forward, backward, right, left, right yaw and left yaw like the figure 3.4 below.

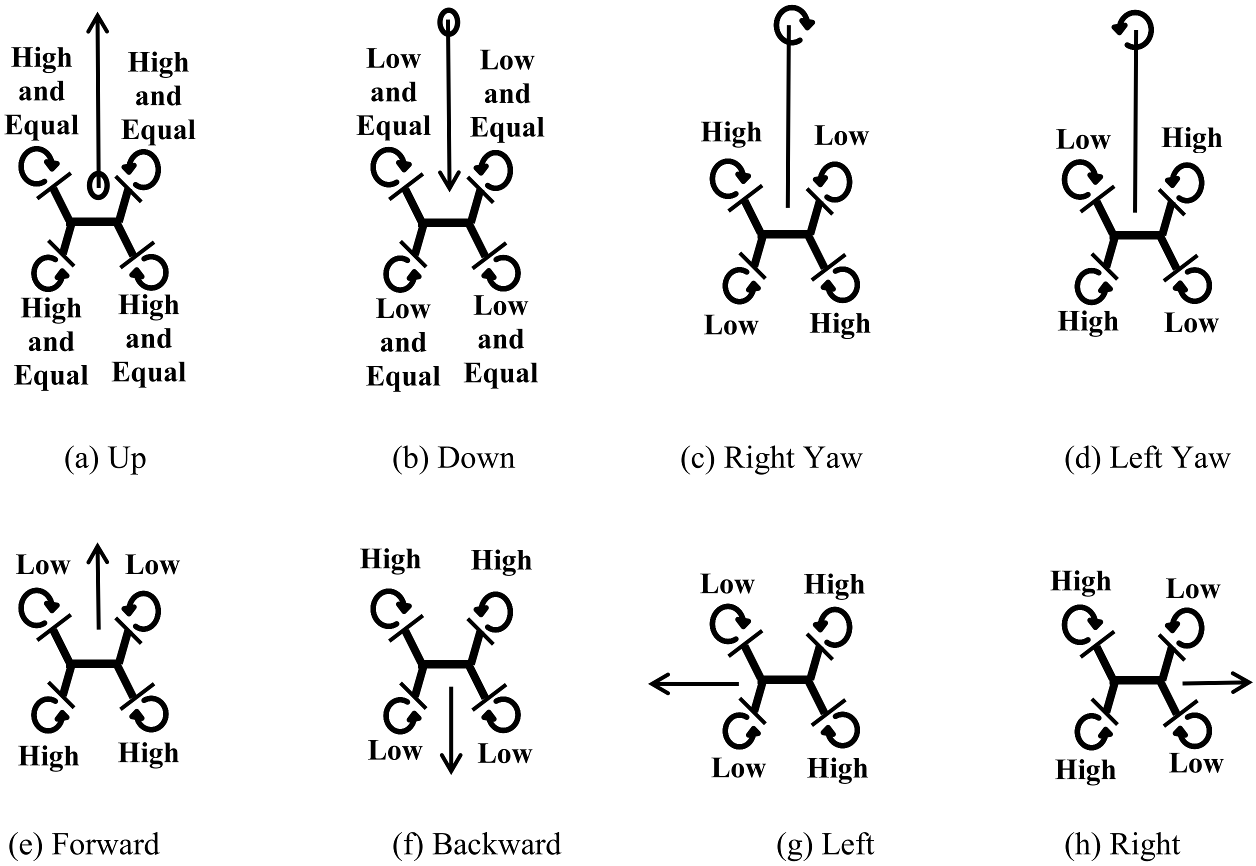


Figure 3.4: Basic operations of a quadcopter drone.

From the figure 3.4 we can see, in a quadcopter 2 diagonally positioned motors must spin in one direction and other 2 diagonally positioned motors must spin in the opposite direction. This is done to prevent unwanted yaw motion. When the total structure starts rotating, it is called yaw motion.

An important task in writing the code is to answer the question-

"What will happen if the communication between the cell towers and the internet modem in the raspberry pi lost?"

This thesis will run the code inside the raspberry pi and give the commands or user inputs from the ground station or through cell towers. So, if the connection lost the code will be still running. So, here the task is to automatically stop the running code and before stopping the code, it should land the UAV safely.

Fortunately, there is threading function in python. By using threading it is set in the code that if the user does not give any input within 15 seconds the code will call the landing function and then will exit the code. It works like multiprocessing. Concurrency is achieved by using threading and multiprocessing in Python. But the main difference between them is that threads run in the same memory location while processes have separate memory locations. Thread is lightweight and suitable for I/O bound tasks.

Python multiprocessing technique has also used to run the camera module simultaneously. And to run the camera from the code, openCV has used. To install openCV in an arm based processor is time consuming and difficult. One has to install a lot of required libraries **[43]**.

To run the total system, this thesis uses the algorithm 1 and flowchart 1. Here, it is important to stop the PWM function of all the 4 GPIO pins those have used and to clean up all the ports those were used after getting the exit or landing command. If these are not done, these affect the next time running the system.

In brief, the algorithm and flowchart work like this-

If the user give input anything from 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N' the algorithm will continue running. If the input is anything other than these keys (except the 'esc' key) or, there is no input within 15 seconds then the algorithm gets the instructions to land the UAV and exit running the algorithm. 'Esc' key will stop running the camera module at any time.

For the convenience of any future work the connected pins diagram of the raspberry pi is given in diagram 1. If a motor does not spin according to the direction, reconnecting the 3 wires of the motor to the ESC by rearranging, solve the problem.

So, it is possible to make and control a UAV using RFB protocol which this thesis will try to create and fly in the next section.

3.4.1 Algorithm

Algorithm 1: Spinning the BLDC motors.

Functions are ‘Down’, ‘Up’, ‘Forward’, ‘Left’, ‘Backward’, ‘Right’, ‘Right_yaw_motion’, ‘Left_yaw_motion’, ‘Landing’.

While input is **True** within a time period-

Camera ⟶ On

if input = ‘F’ **then**

Continue running the system with previous state.

else if input = ‘G’ **then**

call ⟶ Down

else if input = ‘H’ **then**

call ⟶ Up

else if input = ‘I’ **then**

call ⟶ Forward

else if input = ‘J’ **then**

call ⟶ Left

else if input = ‘K’ **then**

call ⟶ Backward

else if input = ‘L’ **then**

call ⟶ Right

else if input = ‘M’ **then**

call ⟶ Right_yaw_motion

else if input = ‘N’ **then**

call ⟶ Left_yaw_motion

else, exit ⟶ loop

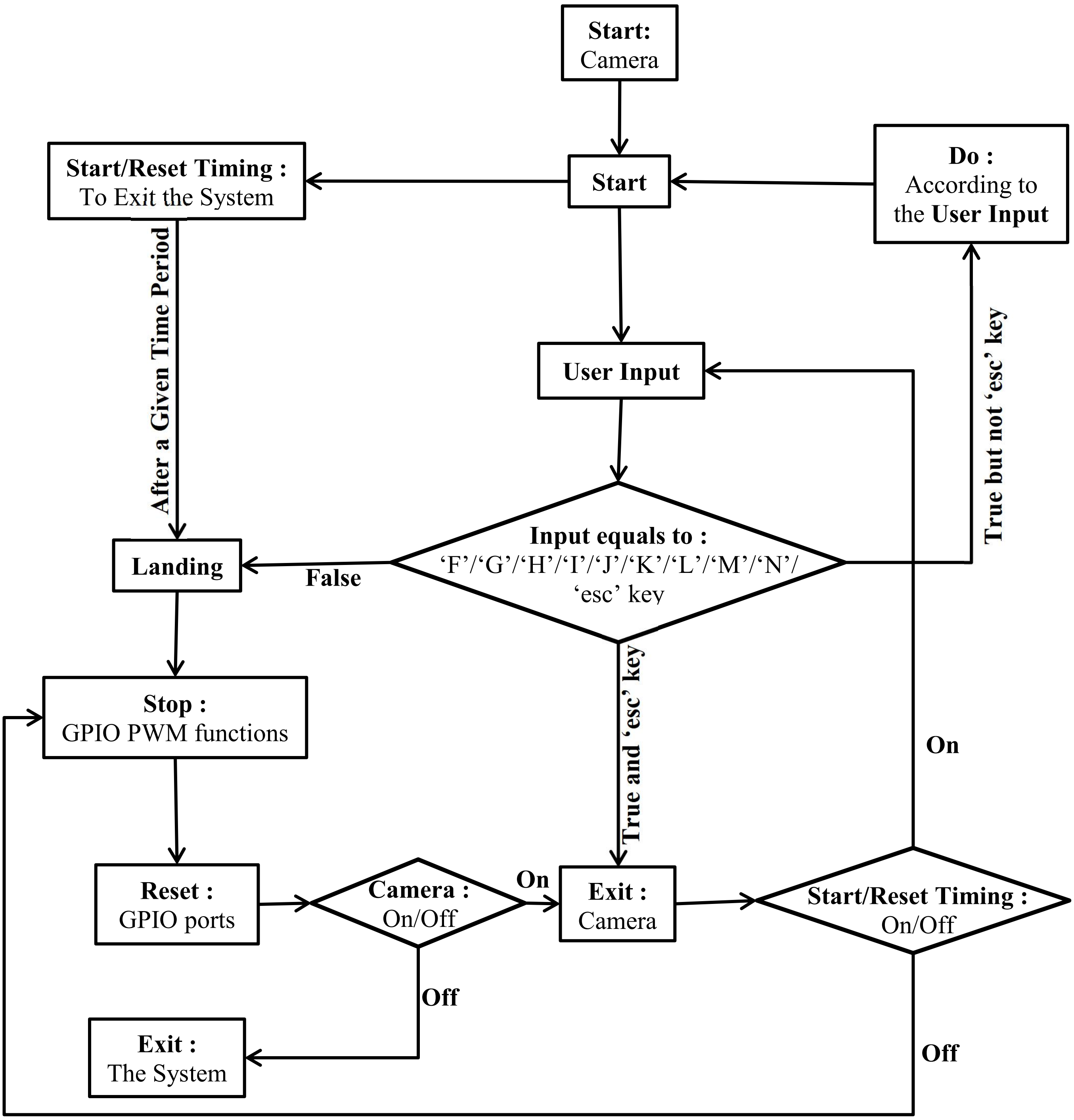
Landing the UAV

stop ⟶ GPIO PWM functions

cleanup ⟶ GPIO Ports

Camera ⟶ Off

3.5 Flowchart



Flowchart 1: Workflow of the total system.

3.6 Circuit Diagram

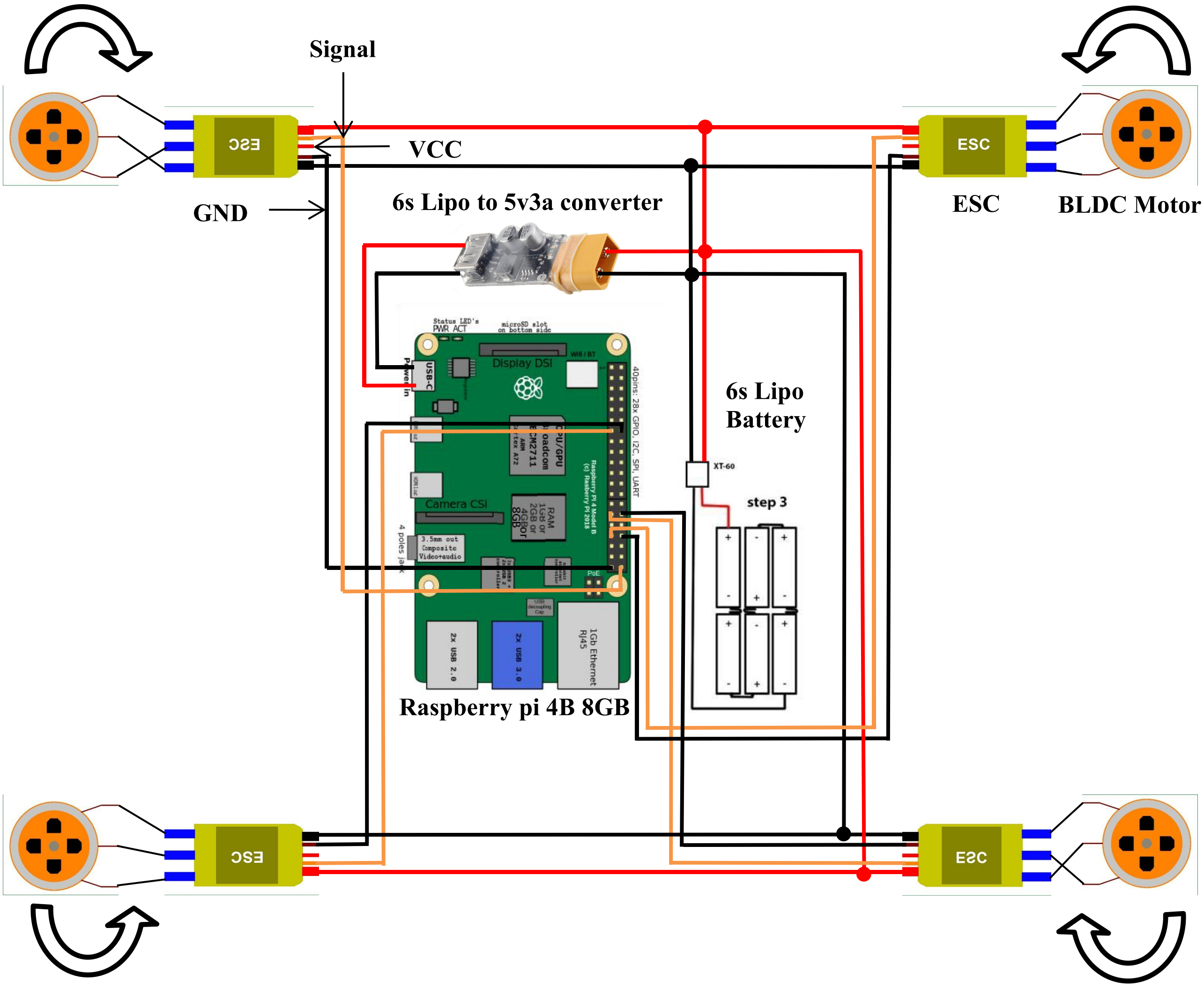


Diagram 1: Circuit Diagram of the total system.

Chapter 4

Field Experiment

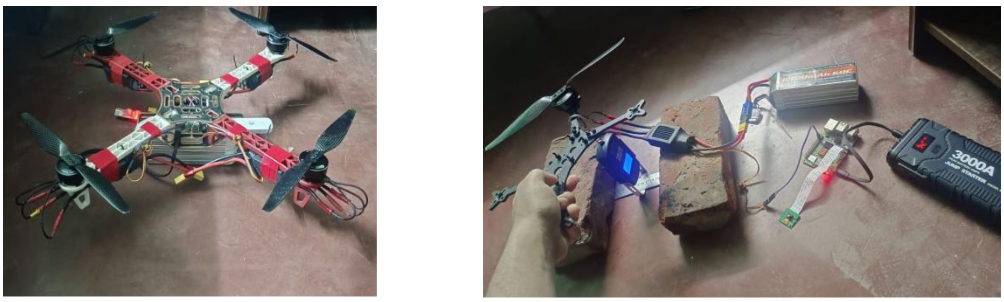
4.1 Setup Information

After setting up all the equipment and software, it is time to run the total system in an open area. It is important to collect all the necessary data for future need. Table 4.1 gives us information on weight of the total system and thrust produced by each of the four 10 inches propeller.

Table 4.1: System weight and thrust produced by the propeller.

Equipment	Information
Total system's weight (without battery)	1300 grams.
Battery weight	850 grams.
Maximum thrust given by each 10 inches propeller	700 grams.

(a) Top view (b) Front view (c) Backside view



(d) Side view (e) Thrust test

Photos 1: Total setup before flight.

Here, an important fact to mention is that while testing the thrust (photos 1(e)), the data that shows on the weighing scale does not include the weight of the motor itself and propeller, so these need to add with the weighing scale data.

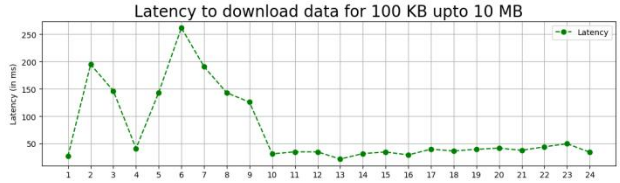
4.2 Internet Speed

To measure the internet speed is important in this type of research work. As mentioned before that the network behavior on the ground and up above a certain height are different. The result found after testing the internet speed on the ground and above 108 feet height are illustrated in the figure 4.2. The test was done using online testing site **[44]**. The most beneficiary side of this site is, it provides a csv file which can later be used to visualize the data according to the user's need. To test the internet speed it downloaded and uploaded different sized data from 100 KB to 10 MB consistently. The simulation below is illustrating the results where the left side graphs are showing the results on the ground (figure 4.2(i)) and right side graphs are showing the results on 108 feet height (figure 4.2(i)). All the results are measured in milliseconds (ms) except BPS (Bits Per Second). It is showing results in bits (in graph it is showing in exponential scale).

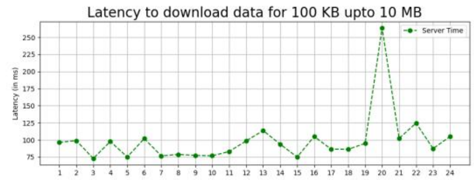
From the figure it can be seen that while downloading, latency is higher at 108 feet height, though at the beginning it was higher on the ground (figure 4.2(a)). At 108 feet height, latency is higher for big sized data like 10 MB. BPS was fluctuating throughout the testing while downloading(figure 4.2(b)). For small sized data to download BPS was very low at ground elevation. Duration to download was higher for large size data for both the heights (figure 4.2(c)). Server time is the round trip time (RTT) which is the time taken for a request to go from one source to one destination and then get back to the source with a response. It was nearly constant and smaller on the ground (figure 4.2(d)).

While uploading, it is noticed that latency was also lower on the ground (figure 4.2(e)) but for both the heights it was fluctuating. BPS was also higher and comparatively stable (figure 4.2(f)). Duration to upload on the ground was relatively stable but for larger sized data it was taking longer time (figure 4.2(g)(i)). On the other side, on the ground it was taking less amount of time but at the end there was a spike (figure 4.2(g)(ii)). And server time or RTT was also showing similar results at both elevations (figure 4.2(h)).

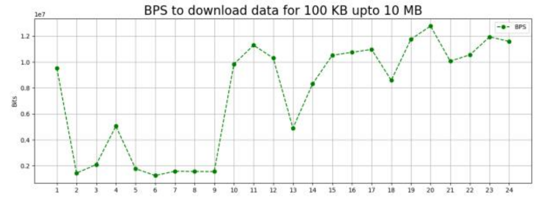
Overalall, on the ground the internet connection was more stable and internet speed was faster.



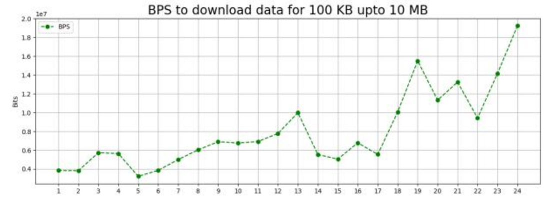
(a) (i) On the ground.



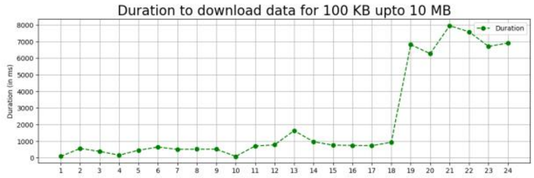
(a) (ii) At 108 feet elevation.



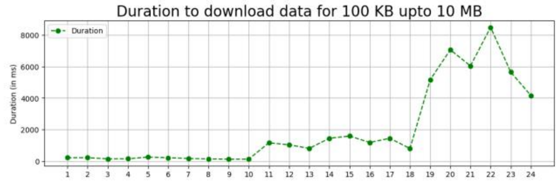
(b) (i) On the ground.



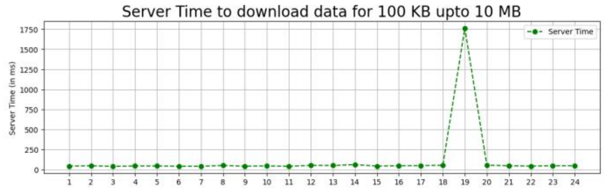
(b) (ii) At 108 feet elevation.



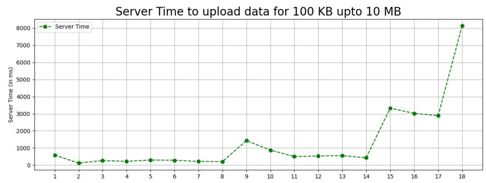
(c) (i) On the ground.



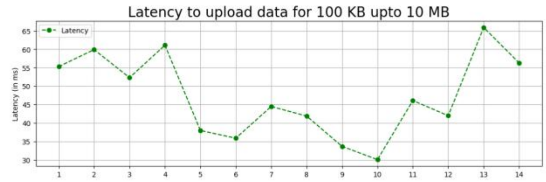
(c) (ii) At 108 feet elevation.



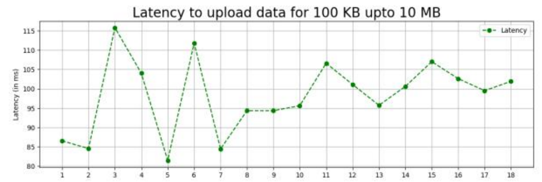
(d) (i) On the ground.



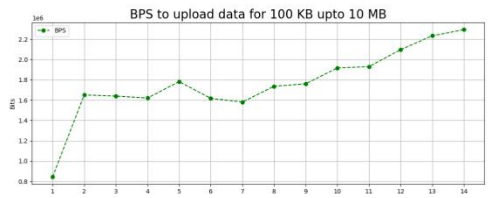
(d) (ii) At 108 feet elevation.



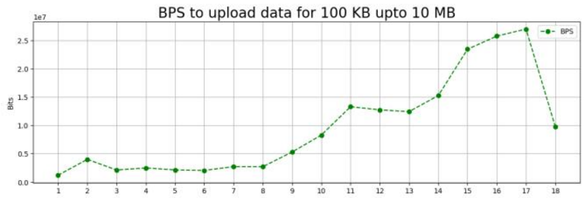
(e) (i) On the ground.



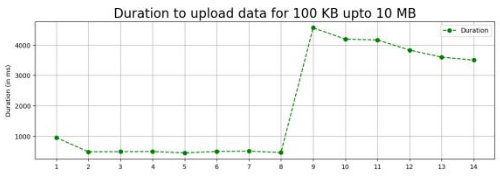
(e) (ii) At 108 feet elevation.



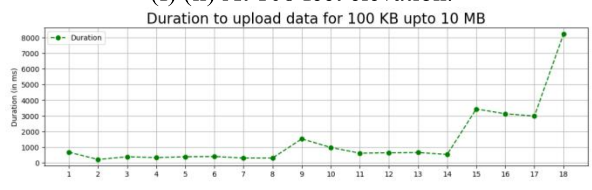
(f) (i) On the ground.



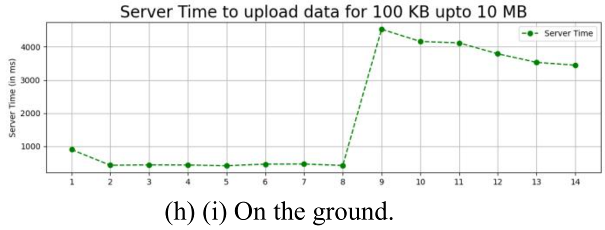
(f) (ii) At 108 feet elevation.



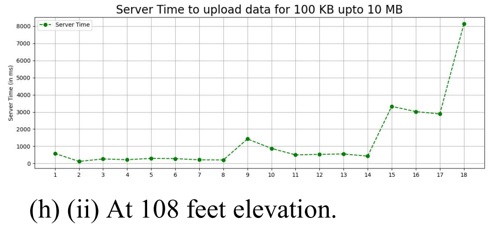
(g) (i) On the ground.



(g) (ii) At 108 feet elevation.



(h) (i) On the ground.



(h) (ii) At 108 feet elevation.

Figure 4.2: Internet Speed on the ground and at 108 feet elevation.

4.3 Latency Comparison Between RFB protocol and Other Types of Live Streaming Protocol

The term "latency" in live streaming describes the interval of time between the time a scene is captured by the camera and when it is shown to viewers. One should consider how the level of latency may impact his or her viewers when setting up a live stream. A lower latency is preferable when responding to comments and questions from viewers during live chat sessions. Keep in mind that viewers may notice more playback buffering at lower latency. Higher latency poses no issues if one doesn't interact with the audience. Three latency levels are available for live streams: normal, low, and ultra-low. Less than 10 seconds are the typical video latency for live streaming. In the case of using RFB protocol the latency is very low and visually there is no latency but considering the latency, there is not much buffering. The table 4.3 shows the latency in live chat using google meet, facebook live streaming, using RTSP (Real Time Streaming Protocol) protocol and using RFB protocol, using same setup for each protocol.

Table 4.3: Latency in different types of streaming protocols and using RFB Protocol.

Protocol Name	Latency
WebRTC (Web Real-Time Communication) protocol used by google meet	2 seconds
RTMPS which is a secure form of RTMP (Real Time Messaging Protocol) used by facebook live	5-10 seconds
RTSP (Real Time Streaming Protocol) [45]	5-10 seconds
RFB (Remote FrameBuffer) Protocol	Less than 1 second

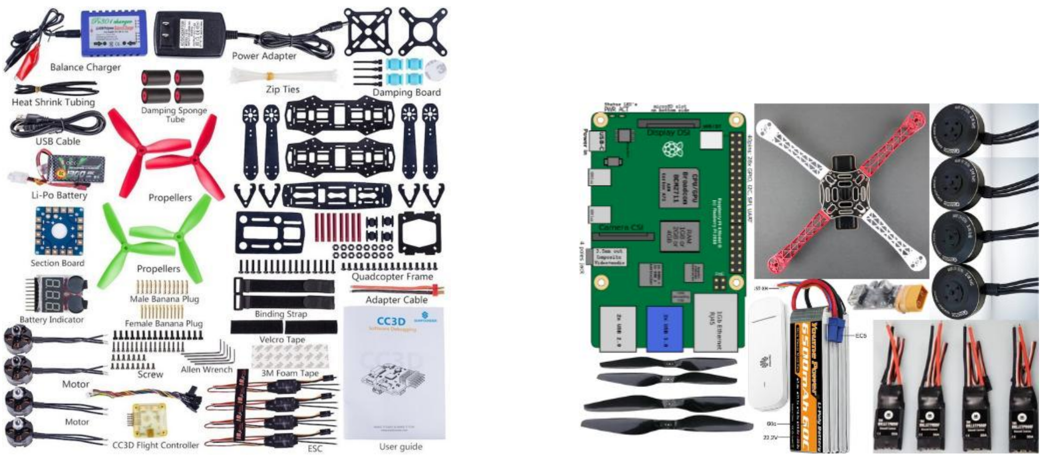
Chapter 5

Discussion

In this section, it will be discussed that how the goals and objectives of this thesis were achieved.

5.1 Simple System Structure

To make the entire UAV system understandable for the average person, is one of the objectives of this thesis. The RFB protocol was used to accomplish it. An upward force must be produced in order to fly a UAV. One method of applying aerodynamic theories is to spin a propeller. We need an energy source and electric motors to rotate the propellers. ESCs were used to regulate the flow of electricity from the energy source using an electronic propulsion system. A pi board was also employed to transmit the command to move in accordance with the user's instructions. The benefits of using a pi board instead of a flight controller are it has a user interface and it supports internet connection as well which helped to achieve another goal of this thesis, to control a UAV worldwide and to achieve the main goal of the thesis, to control a UAV using RFB protocol.



(a) Traditional method (b) Using RFB protocol

Figure 5.1: Equipment to build a quadcopter drone in traditional method and using RFB protocol.

Figure 5.1 shows the comparison of equipment needed between the traditional method and using the RFB protocol. In making a RFB protocol there is no complex connection and no need to calibrate the ESCs with the radio transmitter. The setup is like 'just plug and play'. There is no complex software setup as well. A person who knows how to write a code in python can easily customize the code according to his/her need. And the persons who does not know python, for them a software can be build which will be more user friendly.

It is possible to communicate with the raspberry pi with the SSH (Secure SHell) connection but in that case there will be no visual element and while live streaming through the pi camera using RTSP (Real-Time Streaming Protocol) there is a 5-10 seconds delay **[45]**. It is because once the stream which is received from the server to a Video Distribution Service (VDS), is assigned to process it. It adds delivery method, security features, scalability and compatibility. For example, HLS (HTTP Live Streaming) protocol delivers video segments in small chunks, which makes it scalable and compatible with most devices, but also introduces a latency of 5-20 seconds. In addition, multiple bitrates are output. So for the incoming stream, there is need to make multiple renditions so that viewers with varying internet speeds can watch the video without buffering or lagging. On the other hand, RFB protocol is optimized for low latency and high performance. It has provided a lot faster streaming than the other live streaming protocols during this thesis work (Table 4.2). That is a reason behind using RFB protocol instead of using other client-server based protocol in this thesis.

5.2 Cost Comparison

One of the goal of this thesis is to reduce the overall cost of making a customizable UAV. Using RFB protocol, there is no need to use a flight controller, power distribution board, transmitter, radio signal receiver. Only need to use an arm based microprocessor like raspberry pi with internet dongle to receive commands and showing the GUI from anywhere in the world. Raspberry pi also supports WiFi. So, one can also control the UAV without using internet dongle within the WiFi coverage area. Table 5.2 shows a comparative analysis on the cost of making a customizable quadcopter drone in traditional method and using RFB protocol. The table below shows the information for making same type of quadcopter drone in 2 ways. All the products prices are based on online prices and there was an intention to find the cheapest possible and secured way to buy the products.

Table 5.2: Cost comparison of making a quadcopter drone between traditional method and using RFB protocol.

In traditional method	Cost	Using RFB protocol	Cost
F450 Frame (1 piece)	$17 [46]	Same	$17
PDB (1 piece)	Already attached with the F450 Frame.		
7s Flight Controller (1 piece)	$46 [47]	Raspberry pi 4B 8GB (1 piece)	$87 [54]
Transmitter (1 piece) and Radio Receiver (1 piece)	$55 [48]	HUAWEI E3372 LTE Modem (1 piece)	$32 [55]
7s ESC (4 pieces)	$37 x 2 = $74 [49]	Same	$74
BLDC Motors (4 pieces)	$20 x 4 = $80 [50]	Same	$80
6s Lipo Battery (1 piece)	$64 [51]	Same	$64
FPV Camera (1 piece) and Video Transmitter (1 piece)	$19 [52]	5MP 1080P camera module for Raspberry pi 4B	$8 [56]
Video receiver Antenna (1 piece)	$32 [53]		
Total	**= $387**		**= $362**

Though the total cost for both the cases look similar but using RFB protocol it is actually much cheaper because rapberry pi and LTE modem are not only used for making FPV drone. They are regularly used at home and by the students for different kind of science projects. But one can not use the equipment those are needed to create a UAV in traditional method in such a way. So, if one has a raspberry pi and LTE modem at home then the total cost for making a drone using RFB protocol becomes $243. And last but not least, This thesis has used highest variant of raspberry pi which has 8gb RAM but it is totally okay if anyone uses 4gb variant. One important fact is, all the LTE modems may not be supported by the raspberry pi. [57] Here is a list of raspberry pi supported internet dongles.

Another fact is needed to mention here is that radio transmitter and radio receiver are not universal. So, while choosing traditional method one needs to buy a radio signal receiver that is compatible with the radio transmitter. There are some more things to remember while choosing traditional method, such as, FC and ESCs are needed to run using the same ESC protocol i.e., DShoot 600. So, it is important to make sure that both the FC and ESCs support the same ESC protocol.

There are some works that uses Navio2 autopilot to control the UAV using raspberry pi but in their approaches the complexity of making and controlling the drone remain the same. And it is expensive as well. The price of Navio2 autopilot kit is $238 [58].

5.3 Controlling a UAV Worldwide

An important objective of this thesis is to make a UAV controllable from anywhere in the world. That is why this thesis has used wireless broadband internet. One may fall in an awkward situation when entering into a new country while running the system. In that kind of situation if it is needed and one have necessary permission, he/she can use internet roaming service. It is a little expensive (not too much) but make the work done. As these thesis uses the RFB protocol through the internet. So, if the first person who is controlling the UAV, gives permission to access the UAV to any person(s) from another part of the world can control the UAV as his/her (their) wishes. But one needs to make sure that he/she trust that person(s). The original owner of the UAV can remove any user to access the UAV at anytime as he/she wishes and can change the password to access the system at any time.

5.4 Limitations

This thesis is unable to reduce the time from getting the signal by the raspberry pi to response according to the user input. It is not because of the limitation of the cell towers or LTE modem. It is due to the limitation of the python threading and multiprocessing. After starting a thread it is needed to sleep the running code for a given time period otherwise, the running thread stops immediately. And if it is tried to use python multiprocessing technique, it is not permitted to pass a process as a parameter inside another process. These make it difficult to response for the system as early as possible. Another important fact about the quadcopter of this thesis is if some how connection lost than the system will wait for a given time period to get the user input. After this time period the system will automatically land just under the last position. It is not set to land in a safer place. Hope, later research will solve the problem. But at present, to run such kind of UAV, it is recommended to fly the drone up over dry and plain places. Another limitation of this thesis is the internet speed testing which was tested at just 108 feet height. At this height the internet speed was slower than on the ground but the internet connection and it's speed should be tested at higher altitude on a UAV. This type of UAV which uses RFB protocol will not work where there is no cell phone tower. Satellite internet works with this type of UAV but satellite internet is expensive. So, in these type of areas only WiFi will work well.

5.5 Future Directions of Further Research

Future works can be done on the following areas on the UAV to control it using RFB protocol :

- RFB protocol is little slow on raspberry pi as a server which has 32 bits processor. Latest 64 bits version has released. That should be checked out. Real VNC works well but more research should be done.
- Software researchers can work on python threading to stop a thread by writing stop command and on python multiprocessing to pass a process inside another process as a parameter.
- To control a UAV using RFB protocol a software can be build which will be installed in the server device and a client can input command easily to move the UAV according to his/her need.

Chapter 6

Conclusion

To sum up this thesis believe drone technologies using RFB protocol has a future due to it's easy setup and wide control area. When a drone will be accessible from anywhere in the world, then one can visit a place without physically visiting there. Willing companies can provide such service to show the beautiful sights of their countries to the people of another countries at a cheap price. In this process many people will not visit a place frequently and the nature will be kept reserve. Manking a UAV is expensive and researchers are not willing to research on UAV because there is chance of crashing if something went wrong. Then the cost does not bring any profit. Major contributions of this thesis can be pointed out as below-

- Implementing RFB protocol in a UAV.
- Controlling a UAV worldwide through the cell phone towers.
- An easiest way to setup a UAV.
- An easiest way to setup the software and to run it.
- A cost effective way to make a UAV at home.
- A raspberry pi based drone with raspbian operating system.

Flying a UAV seems easy like only need to spin the propeller(s) but controlling it is difficult. Wright brothers first flied an aircraft and successfully landed. Before them a lot of attempts were occurred but all of them either died or badly injured due to crash or never came back. As mentioned before that there are some limitations of this research due to unavailable resources. But hope, future research will overcome these limitations. And proper funding will increase the amount of researches in this area.

Bibliography

[1] Internet users according to "Miniwatts Marketing Group".

https://www.internetworldstats.com/stats.htm

[2] [Kratky & Reichenberger et al., 2013]. Sascha Kratky and Christoph Reichenberger, edited by Richard Clark (2013). Client/Server Development based on the Apple Event Object Model. Atlanta.

http://preserve.mactech.com/articles/mactech/Vol.14/14.11/Client-ServerDevelopment/index.html

[3] [Shakhirat et al., 2014]. Haroon Shakirat Oluwatosin (2014). Client-Server Model. Published in - IOSR Journal of computer engineering (IOSR-JCE). Published on - February, 2014.

http://www.iosrjournals.org/iosr-jce/papers/Vol16-issue1/Version-9/J016195771.pdf

[4] A presentation on 3-Tier Architecture.

https://fdocuments.net/document/3-tier-architecture-558454cb58a35.html?page=1

[5] Frsky ACCST protocol's user manual.

https://www.frsky-rc.com/wp-content/uploads/2017/07/Manual/X8R.pdf

[6] Spektrum DSMX protocol.

https://www.spektrumrc.com/spm-bs-dsmx.html

[7] An artical on Electronic Speed Controller (ESC) by Tyto Robotics.

https://www.tytorobotics.com/blogs/articles/what-is-an-esc-how-does-an-esc-work

[8] [Holtz et al., 1992]. Joachim Holtz (1992). Pulsewidth modulation--A survey. Published in - IEEE Journal. Published on - June, 1992.

https://www.researchgate.net/publication/3536915_Pulsewidth_modulation--A_survey

[9] An artical on FPV protocols made by Oscar Liang.

https://oscarliang.com/rc-protocols/

[10] Spacecraft electric propulsion

https://en.wikipedia.org/wiki/Spacecraft_electric_propulsion

[11] [Andrew and Dries et al., 2017]. Andrew Gong and Dies Verstraete (2017). Experimental Testing of Electronic Speed Controllers for UAVs. Published in - Aerospace Research Central Journal. Published on - July, 2017.

https://www.researchgate.net/publication/318294444_Experimental_Testing_of_Electronic_Speed_Controllers_for_UAVs

[12] [Sneha et al., 2016]. Sneha.K, Ishwarya.N, Dharani.E and Uma Maheshwari (2016). An Overview of Cell Phone Towers. Published in - International Journal of Trend in Research and Development. Published on - September, 2016.

http://www.ijtrd.com/papers/IJTRD4244.pdf

[13] [Euler et al., 2019]. Sebastian Euler, Helka-Liina Maattanen, Xingqin Lin, Zhenhua Zou, Mattias Bergström, and Jonas Sedin Ericsson (2019). Mobility Support for Cellular Connected Unmanned Aerial Vehicles: Performance and Analysis. Published in - IEEE Journal. Published on - October, 2019.

https://arxiv.org/ftp/arxiv/papers/1804/1804.04523.pdf

[14] Pixhawk autopilot website.

https://pixhawk.org/

[15] [Tao et al., 2013]. Tao Lin, Kailun Zhou and Shuhui Wang (2013). Cloudlet–screen computing: a client–server architecture with top graphics performance. Published in - Inder Science Journal. Published on - May, 2013.

https://www.inderscienceonline.com/doi/abs/10.1504/IJAHUC.2013.054174

[16] [Dao et al., 2021]. Nhu-Ngoc Dao, Quoc-Viet Pham, Ngo Hoang Tu, Tran Thien Thanh, Vo Nguyen Quoc Bao, Demeke Shumeye Lakew, and Sungrae Cho (2021). Survey on Aerial Radio Access Networks: Toward a Comprehensive 6G Access Infrastructure. Published in - IEEE Journal. Published on - February, 2021.

https://arxiv.org/pdf/2102.07087v2.pdf

[17] An article on 'Understanding Drone Aerodynamics: The Key to Efficient and Stable Flight' by Dr. M. Subramanian.

https://www.linkedin.com/pulse/understanding-drone-aerodynamics-key-efficient-stable-#

[18] [Abhisek et al., 2020]. Abhishek Sharma, Pankhuri Vanjani, Nikhil Paliwal, Chathuranga M. Wijerathna Basnayaka, Dushantha Nalin K. Jayakody, Hwang-Cheng Wang, P. Muthuchidambaranathan (2020). Communication and Networking Technologies for UAVs: A Survey. Published in - Science Direct Journal. Published on - Octobor, 2020.

https://arxiv.org/pdf/2009.02280.pdf

[19] [Yang et al., 2023]. Min Yang, Guanjun Liu, Ziyuan Zhou, and Jiacun Wang (2023). Partially Observable Mean Field Multi-Agent Reinforcement Learning Based on Graph Attention Network for UAV Swarms. Published in - MDPI Journal. Published on - May, 2023.

https://www.mdpi.com/2504-446X/7/7/476

[20] [Kangunde et al., 2021]. Vemema Kangunde, Rodrigo S. Jamisola Jr. and Emmanuel K. Theophilus (2021). A review on drones controlled in real-time. Published in - Springer Link Journal. Published on - January, 2021.

https://link.springer.com/article/10.1007/s40435-020-00737-5

[21] [Arya et al., 2023]. Sudhanshu Arya, Jingda Yang, Ying Wang (2023). Towards the Designing of Low-Latency SAGIN: Ground-to-UAV Communications over Interference Channel. Published in - MDPI Journal. Published on - June, 2023.

https://www.mdpi.com/2504-446X/7/7/479

[22] [Wzorek et al., 2006]. Mariusz Wzorek, Patrick Doherty, David Landen (2006). GSM Technology as a Communication Media for an Autonomous Unmanned Aerial Vehicle. Published in - IDA Journal. Published on - April, 2006.

https://www.researchgate.net/publication/254847028_GSM_Technology_as_a_Communication_Media_for_an_Autonomous_Unmanned_Aerial_Vehicle

[23] [Mingzhe et al., 2017]. Mingzhe Chen, Mohammad Mozaffari, Walid Saad, Changchuan Yin, Mérouane Debbah, and Choong-Seon Hong (2017). Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience. Published in - IEEE Journal. Published on - March, 2017.

https://arxiv.org/pdf/1610.01585v1.pdf

[24] [Mozaffari et al., 2017]. Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Merouane Debbah (2017). Mobile Unmanned Aerial Vehicles (UAVs) for Energy-Efficient Internet of Things Communications. Published in - IEEE Journal. Published on - September, 2017.

https://arxiv.org/pdf/1703.05401.pdf

[25] [Sudhesh et al., 2017]. Sudheesh P. G, Mohammad Mozaffari, Maurizio Magarini, Walid Saad and P. Muthuchidambaranathan (2017). Sum-Rate Analysis for High Altitude Platform (HAP) Drones with Tethered Balloon Relay. Published in - IEEE Journal. Published on - December, 2017.

https://arxiv.org/pdf/1712.06583.pdf

[26] [Kalantari et al., 2017]. Elham Kalantari, Muhammad Zeeshan Shakir, Halim Yanikomeroglu, and Abbas Yongacoglu (2017). Backhaul-aware Robust 3D Drone Placement in 5G+ Wireless Networks. Published in - IEEE Journal. Published on - March, 2017.

https://arxiv.org/pdf/1702.08395.pdf

[27] [Fouda et al., 2018]. Abdurrahman Fouda, Ahmed S. Ibrahim, Ismail Guvenc and Monisha Ghosh (2018). UAV-Based In-Band Integrated Access and Backhaul for 5G Communications. published in - IEEE Journal. Published on - August, 2018.

https://arxiv.org/ftp/arxiv/papers/1807/1807.07230.pdf

[28] [Lokman et al., 2017]. Lokman Sboui, Hakim Ghazzai and Zouheir Rezki (2017). On the Throughput of Cognitive Radio MIMO Systems Assisted with UAV Relays. Published in - IEEE Journal. Published on - June, 2017.

https://www.researchgate.net/publication/318579011_On_the_throughput_of_cognitive_radio_MIMO_systems_assisted_with_UAV_relays

[29] [Rupasinghe et al., 2016]. Nadisanka Rupasinghe, Ahmed S Ibrahim and Ismail Guvenc (2016). Optimum Hovering Locations with Angular Domain User Separation for Cooperative UAV Networks. Published in - IEEE Journal. Published on - December, 2016.

https://www.researchgate.net/publication/313456453_Optimum_Hovering_Locations_with_Angular_Domain_User_Separation_for_Cooperative_UAV_Networks

[30] [Yalniz et al., 2016]. R. Irem Bor Yaliniz, Amr El-Keyi, and Halim Yanikomeroglu (2016). Efficient 3-D Placement of an Aerial Base Station in Next Generation Cellular Networks. Published in - IEEE Journal. Published on - February, 2016.

https://arxiv.org/pdf/1603.00300.pdf

[31] [Abel-Malek et al., 2018]. Mai A. Abdel-Malek, Ahmed S. Ibrahim, and Mohamed Mokhtar (2018). Optimum UAV Positioning for Better Coverage-Connectivity Tradeoff. Published in - IEEE Journal. Published on - February, 2018.

https://par.nsf.gov/servlets/purl/10041983

[32] [Mozaffari et al., 2016]. Mohammad Mozaffari, Walid Saad, Mehdi Bennis and Merouane Debbah (2016). Unmanned Aerial Vehicle with Underlaid Device-to-Device Communications: Performance and Tradeoffs. Published in - IEEE Journal. Published on - February, 2016.

https://arxiv.org/pdf/1509.01187.pdf

[33] [Mozaffari et al., 2019]. Mohammad Mozaffari, Walid Saad, Mehdi Bennis, Young-Han Nam, and Merouane Debbah (2019). A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems. Published in - IEEE Journal. Published on - March, 2019.

https://arxiv.org/pdf/1803.00680.pdf

[34] [Zeng et al., 2018]. Yong Zeng, Jiangbin Lyu and Rui Zhang (2018). Cellular-Connected UAV: Potentials, Challenges and Promising Technologies. Published in - IEEE Journal. Published on - September, 2018.

https://arxiv.org/pdf/1804.02217.pdf

[35] [Guang et al., 2018]. Guang Yang, Xingqin Lin, Yan Li, Hang Cui, Min Xu, Dan Wu, Henrik Rydén and Sakib Bin Redhwan (2018). A Telecom Perspective on the Internet of Drones: From LTE-Advanced to 5G. Published in - Arxiv Journal. Published on - March, 2018.

https://arxiv.org/ftp/arxiv/papers/1803/1803.11048.pdf

[36] [Muruganathan et al., 2021]. Siva D. Muruganathan, Xingqin Lin, Helka-Liina Määttänen, Zhenhua Zou, Wuri A. Hapsari and Shinpei Yasukawa (2021). An Overview of 3GPP Release-15 Study on Enhanced LTE Support for Connected Drones. Published in - IEEE Journal. Published on - December, 2021.

https://arxiv.org/vc/arxiv/papers/1805/1805.00826v1.pdf

[37] Small Unmanned Aircraft Systems (UAS) Regulations (Part 107). October, 2020.

https://www.faa.gov/newsroom/small-unmanned-aircraft-systems-uas-regulations-part-107

[38] RC LiPo Battery Power and Configuration Calculator.

https://www.radiocontrolinfo.com/information/rc-calculators/rc-lipo-battery-power-configuration-calculator/

[39] Datasheet of the BLDC motor that has used in this thesis.

http://www.qxmodel.net/index.php?m=Show&a=index&cid=172&id=125

[40] [Robert et al., 2017]. Robert Deters, Stefan Kleinke and, Michael S Selig (2017). Static Testing of Propulsion Elements for Small Multirotor Unmanned Aerial Vehicles. Published in - Aerospace Research Central Journal. Published on - June, 2017.

https://www.researchgate.net/publication/318169547_Static_Testing_of_Propulsion_Elements_for_Small_Multirotor_Unmanned_Aerial_Vehicles

[41] [Qin et al., 2023]. Titanyi Qin, Guangyu Zhang, Liying Yang and Yuqing He (2023). Research on the Endurance Optimisation of Multirotor UAVs for High-Altitude Environments. Published in - MDPI Journal. Published on - July, 2023.

https://www.mdpi.com/2504-446X/7/7/469

[42] Aerodynamics and Aircraft Performance, 3rd edition. Written by James F. Marchman. Published in 2004.

https://pressbooks.lib.vt.edu/aerodynamics/chapter/chapter-1/

[43] How to install openCV on raspberry pi 4.

https://www.qengineering.eu/install-opencv-4.2-on-raspberry-pi-4.html

[44] Internet speed test.

https://speed.cloudflare.com/

[45] A tutorial on 'How To Stream Live Video From Your Raspberry Pi Camera' by Les Pounder.

https://www.tomshardware.com/how-to/stream-live-video-raspberry-pi

[46] F450 Frame.

https://www.amazon.com/4-axle-Drone-Colored-Quadcopter-Integrated/dp/B0C91NQPPY

[47] 7S flight controller.

https://www.amazon.com/gp/product/B0C49V1YP6

[48] Transmitter and Radio Receiver.

https://www.amazon.com/Flysky-Transmitter-Multirotor-Helicopter-Quadcopter/dp/B07CXL9LCT

[49] 7S ESC.

https://www.amazon.com/Vaveren-2pcs-Controller-Brushless-Helicopter/dp/B0B86GBQH9

[50] BLDC motor.

https://www.aliexpress.com/item/1005005670219453.html

[51] 6S lipo battery.

https://www.ebay.com/itm/362763910374

[52] FPV Camera and Video Transmitter.

https://www.amazon.com/AKK-5-8GHz-Transmitter-600TVL-Antenna/dp/B06XV7G439

[53] Video receiver Antenna.

https://www.amazon.com/Receiver-Female-Plastic-Android-Monitor/dp/B07Q5MPC8V

[54] Raspberry pi, model-4B, RAM-8GB.

https://www.amazon.com/Raspberry-Pi-Computer-Suitable-Workstation/dp/B0899VXM8F

[55] HUAWEI E3372 LTE Modem.

https://www.aliexpress.com/item/1005005671553610.html

[56] 5MP 1080P camera module for Raspberry pi 4B.

https://www.amazon.com/Dorhea-Raspberry-Camera-Module-Sensor/dp/B07DNSS1L7

[57] Raspberry pi supported internet dongles.

https://elinux.org/RPi_VerifiedPeripherals#Internet

[58] Navio2 Autopilot Kit.

https://www.robotshop.com/products/navio2-autopilot-kit-raspberry-pi-2-3

Appendix

A1. Python Code to Run a UAV using RFB protocol

```
import os
import cv2
import RPi.GPIO as GPIO
from threading import Thread
from multiprocessing import Process
from time import time
from time import sleep

GPIO.setmode(GPIO.BCM)
GPIO.setup(21,GPIO.OUT)
GPIO.setup(13,GPIO.OUT)
GPIO.setup(5,GPIO.OUT)
GPIO.setup(27,GPIO.OUT)

r=GPIO.PWM(21,50)
r.start(0)
sleep(2)
r.ChangeDutyCycle(3)
sleep(2)

p=GPIO.PWM(13,50)
p.start(0)
sleep(2)
p.ChangeDutyCycle(3)
sleep(2)

q=GPIO.PWM(5,50)
q.start(0)
sleep(2)
q.ChangeDutyCycle(3)
sleep(2)

s=GPIO.PWM(27,50)
s.start(0)
sleep(2)
s.ChangeDutyCycle(3)
sleep(2)

i,j,k,l=3,3,3,3
```

```
def up(i,j,k,l):

    maximum=max(i,j,k,l)
    if maximum>10:
        maximum=10
    while not i==j==k==l:
        if i>maximum:
            p.ChangeDutyCycle(i)
            sleep(0.05)
            i -= 0.01
        if j>maximum:
            p.ChangeDutyCycle(j)
            sleep(0.05)
            j -= 0.01
        if k>maximum:
            p.ChangeDutyCycle(k)
            sleep(0.05)
            k -= 0.01
        if l>maximum:
            p.ChangeDutyCycle(l)
            sleep(0.05)
            l -= 0.01
        if i<maximum:
            p.ChangeDutyCycle(i)
            sleep(0.05)
            i += 0.01
        if j<maximum:
            p.ChangeDutyCycle(j)
            sleep(0.05)
            j += 0.01
        if k<maximum:
            p.ChangeDutyCycle(k)
            sleep(0.05)
            k += 0.01
        if l<maximum:
            p.ChangeDutyCycle(l)
            sleep(0.05)
            l += 0.01

    i,j,k,l =round(i*2)/2,round(j*2)/2,round(k*2)/2,round(l*2)/2
    w,x,y,z=i,j,k,l

    if w>=10 or x>=10 or y>=10 or z>=10:
        print("max speed")

    else:
        while w<=i+0.5 and x<=j+0.5 and y<=k+0.5 and z<=l+0.5:

            p.ChangeDutyCycle(w)
            sleep(0.05)
```

```
            w += 0.01

            q.ChangeDutyCycle(x)
            sleep(0.05)
            x += 0.01

            r.ChangeDutyCycle(y)
            sleep(0.05)
            y += 0.01

            s.ChangeDutyCycle(z)
            sleep(0.05)
            z += 0.01

    i,j,k,l=round (w*2)/2, round (x*2)/2, round (y*2)/2, round (z*2)/2

    return i,j,k,l

def down(i,j,k,l):

    maximum=max(i,j,k,l)
    if maximum>10:
        maximum=10
    while not i==j==k==l:
        if i>maximum:
            p.ChangeDutyCycle(i)
            sleep(0.05)
            i -= 0.01
        if j>maximum:
            p.ChangeDutyCycle(j)
            sleep(0.05)
            j -= 0.01
        if k>maximum:
            p.ChangeDutyCycle(k)
            sleep(0.05)
            k -= 0.01
        if l>maximum:
            p.ChangeDutyCycle(l)
            sleep(0.05)
            l -= 0.01
        if i<maximum:
            p.ChangeDutyCycle(i)
            sleep(0.05)
            i += 0.01
        if j<maximum:
            p.ChangeDutyCycle(j)
            sleep(0.05)
            j += 0.01
        if k<maximum:
            p.ChangeDutyCycle(k)
```

```
            sleep(0.05)
            k += 0.01
        if l<maximum:
            p.ChangeDutyCycle(l)
            sleep(0.05)
            l += 0.01

    i,j,k,l =round(i*2)/2,round(j*2)/2,round(k*2)/2,round(l*2)/2
    w,x,y,z=i,j,k,l

    if w<=3 or x<=3 or y<=3 or z<=3:
        print("lowest speed")

    else:
        while w>=i-0.5 and x>=j-0.5 and y>=k-0.5 and z>=l-0.5:

            p.ChangeDutyCycle(w)
            sleep(0.05)
            w -= 0.01

            q.ChangeDutyCycle(x)
            sleep(0.05)
            x -= 0.01

            r.ChangeDutyCycle(y)
            sleep(0.05)
            y -= 0.01

            s.ChangeDutyCycle(z)
            sleep(0.05)
            z -= 0.01

    i,j,k,l=round (w*2)/2, round (x*2)/2, round (y*2)/2, round (z*2)/2

    return i,j,k,l

def forward(i,j,k,l):

    maximum=max(i,j,k,l)
    if maximum>10:
        maximum=10
    while not i==j==k==l:
        if i>maximum:
            p.ChangeDutyCycle(i)
            sleep(0.05)
            i -= 0.01
        if j>maximum:
            p.ChangeDutyCycle(j)
            sleep(0.05)
            j -= 0.01
```

```
    if k>maximum:
        p.ChangeDutyCycle(k)
        sleep(0.05)
        k -= 0.01
    if l>maximum:
        p.ChangeDutyCycle(l)
        sleep(0.05)
        l -= 0.01
    if i<maximum:
        p.ChangeDutyCycle(i)
        sleep(0.05)
        i += 0.01
    if j<maximum:
        p.ChangeDutyCycle(j)
        sleep(0.05)
        j += 0.01
    if k<maximum:
        p.ChangeDutyCycle(k)
        sleep(0.05)
        k += 0.01
    if l<maximum:
        p.ChangeDutyCycle(l)
        sleep(0.05)
        l += 0.01

i,j,k,l =round(i*2)/2,round(j*2)/2,round(k*2)/2,round(l*2)/2
w,x,y,z=i,j,k,l

while w<=i+0.5 and x<=j+0.5 and y>=k-0.5 and z>=l-0.5:

    p.ChangeDutyCycle(w)
    sleep(0.05)
    w += 0.01

    q.ChangeDutyCycle(x)
    sleep(0.05)
    x += 0.01

    r.ChangeDutyCycle(y)
    sleep(0.05)
    y -= 0.01

    s.ChangeDutyCycle(z)
    sleep(0.05)
    z -= 0.01

while w>=i-0.5 and x>=j-0.5 and y<=k+0.5 and z<=l+0.5:

    p.ChangeDutyCycle(w)
    sleep(0.05)
```

```
        w -= 0.01

        q.ChangeDutyCycle(x)
        sleep(0.05)
        x -= 0.01

        r.ChangeDutyCycle(y)
        sleep(0.05)
        y += 0.01

        s.ChangeDutyCycle(z)
        sleep(0.05)
        z += 0.01

    i,j,k,l=round (w*2)/2, round (x*2)/2, round (y*2)/2, round (z*2)/2

    return i,j,k,l

def left(i,j,k,l):

    maximum=max(i,j,k,l)
    if maximum>10:
        maximum=10
    while not i==j==k==l:
        if i>maximum:
            p.ChangeDutyCycle(i)
            sleep(0.05)
            i -= 0.01
        if j>maximum:
            p.ChangeDutyCycle(j)
            sleep(0.05)
            j -= 0.01
        if k>maximum:
            p.ChangeDutyCycle(k)
            sleep(0.05)
            k -= 0.01
        if l>maximum:
            p.ChangeDutyCycle(l)
            sleep(0.05)
            l -= 0.01
        if i<maximum:
            p.ChangeDutyCycle(i)
            sleep(0.05)
            i += 0.01
        if j<maximum:
            p.ChangeDutyCycle(j)
            sleep(0.05)
            j += 0.01
        if k<maximum:
            p.ChangeDutyCycle(k)
```

```
        sleep(0.05)
        k += 0.01
    if l<maximum:
        p.ChangeDutyCycle(l)
        sleep(0.05)
        l += 0.01

i,j,k,l =round(i*2)/2,round(j*2)/2,round(k*2)/2,round(l*2)/2
w,x,y,z=i,j,k,l

while w<=i+0.5 and x>=j-0.5 and y>=k-0.5 and z<=l+0.5:

    p.ChangeDutyCycle(w)
    sleep(0.05)
    w += 0.01

    q.ChangeDutyCycle(x)
    sleep(0.05)
    x -= 0.01

    r.ChangeDutyCycle(y)
    sleep(0.05)
    y -= 0.01

    s.ChangeDutyCycle(z)
    sleep(0.05)
    z += 0.01

while w>=i-0.5 and x<=j+0.5 and y<=k+0.5 and z>=l-0.5:

    p.ChangeDutyCycle(w)
    sleep(0.05)
    w -= 0.01

    q.ChangeDutyCycle(x)
    sleep(0.05)
    x += 0.01

    r.ChangeDutyCycle(y)
    sleep(0.05)
    y += 0.01

    s.ChangeDutyCycle(z)
    sleep(0.05)
    z -= 0.01

i,j,k,l=round (w*2)/2, round (x*2)/2, round (y*2)/2, round (z*2)/2

return i,j,k,l
```

```
def backward(i,j,k,l):

    maximum=max(i,j,k,l)
    if maximum>10:
        maximum=10
    while not i==j==k==l:
        if i>maximum:
            p.ChangeDutyCycle(i)
            sleep(0.05)
            i -= 0.01
        if j>maximum:
            p.ChangeDutyCycle(j)
            sleep(0.05)
            j -= 0.01
        if k>maximum:
            p.ChangeDutyCycle(k)
            sleep(0.05)
            k -= 0.01
        if l>maximum:
            p.ChangeDutyCycle(l)
            sleep(0.05)
            l -= 0.01
        if i<maximum:
            p.ChangeDutyCycle(i)
            sleep(0.05)
            i += 0.01
        if j<maximum:
            p.ChangeDutyCycle(j)
            sleep(0.05)
            j += 0.01
        if k<maximum:
            p.ChangeDutyCycle(k)
            sleep(0.05)
            k += 0.01
        if l<maximum:
            p.ChangeDutyCycle(l)
            sleep(0.05)
            l += 0.01

    i,j,k,l =round(i*2)/2,round(j*2)/2,round(k*2)/2,round(l*2)/2
    w,x,y,z=i,j,k,l

    while w>=i-0.5 and x>=j-0.5 and y<=k+0.5 and z<=l+0.5:

        p.ChangeDutyCycle(w)
        sleep(0.05)
        w -= 0.01

        q.ChangeDutyCycle(x)
        sleep(0.05)
```

```
        x -= 0.01

        r.ChangeDutyCycle(y)
        sleep(0.05)
        y += 0.01

        s.ChangeDutyCycle(z)
        sleep(0.05)
        z += 0.01

    while w<=i+0.5 and x<=j+0.5 and y>=k-0.5 and z>=l-0.5:

        p.ChangeDutyCycle(w)
        sleep(0.05)
        w += 0.01

        q.ChangeDutyCycle(x)
        sleep(0.05)
        x += 0.01

        r.ChangeDutyCycle(y)
        sleep(0.05)
        y -= 0.01

        s.ChangeDutyCycle(z)
        sleep(0.05)
        z -= 0.01

    i,j,k,l=round (w*2)/2, round (x*2)/2, round (y*2)/2, round (z*2)/2

    return i,j,k,l

def right(i,j,k,l):

    maximum=max(i,j,k,l)
    if maximum>10:
        maximum=10
    while not i==j==k==l:
        if i>maximum:
            p.ChangeDutyCycle(i)
            sleep(0.05)
            i -= 0.01
        if j>maximum:
            p.ChangeDutyCycle(j)
            sleep(0.05)
            j -= 0.01
        if k>maximum:
            p.ChangeDutyCycle(k)
            sleep(0.05)
            k -= 0.01
```

```
    if l>maximum:
        p.ChangeDutyCycle(l)
        sleep(0.05)
        l -= 0.01
    if i<maximum:
        p.ChangeDutyCycle(i)
        sleep(0.05)
        i += 0.01
    if j<maximum:
        p.ChangeDutyCycle(j)
        sleep(0.05)
        j += 0.01
    if k<maximum:
        p.ChangeDutyCycle(k)
        sleep(0.05)
        k += 0.01
    if l<maximum:
        p.ChangeDutyCycle(l)
        sleep(0.05)
        l += 0.01

i,j,k,l =round(i*2)/2,round(j*2)/2,round(k*2)/2,round(l*2)/2
w,x,y,z=i,j,k,l

while w>=i-0.5 and x<=j+0.5 and y<=k+0.5 and z>=l-0.5:

    p.ChangeDutyCycle(w)
    sleep(0.05)
    w -= 0.01

    q.ChangeDutyCycle(x)
    sleep(0.05)
    x += 0.01

    r.ChangeDutyCycle(y)
    sleep(0.05)
    y += 0.01

    s.ChangeDutyCycle(z)
    sleep(0.05)
    z -= 0.01

while w<=i+0.5 and x>=j-0.5 and y>=k-0.5 and z<=l+0.5:

    p.ChangeDutyCycle(w)
    sleep(0.05)
    w += 0.01

    q.ChangeDutyCycle(x)
    sleep(0.05)
```

```
        x -= 0.01

        r.ChangeDutyCycle(y)
        sleep(0.05)
        y -= 0.01

        s.ChangeDutyCycle(z)
        sleep(0.05)
        z += 0.01

    i,j,k,l=round (w*2)/2, round (x*2)/2, round (y*2)/2, round (z*2)/2

    return i,j,k,l

def right_yaw(i,j,k,l):

    w,x,y,z=i,j,k,l

    if i>10 or j<=3 or k>10 or l<=3:
        print("highest right yaw speed.")
    else:
        while w<=i+0.5 and x>=j-0.5 and y<=k+0.5 and z>=l-0.5:

            p.ChangeDutyCycle(w)
            sleep(0.05)
            w += 0.01

            q.ChangeDutyCycle(x)
            sleep(0.05)
            x -= 0.01

            r.ChangeDutyCycle(y)
            sleep(0.05)
            y += 0.01

            s.ChangeDutyCycle(z)
            sleep(0.05)
            z -= 0.01

    i,j,k,l=round (w*2)/2, round (x*2)/2, round (y*2)/2, round (z*2)/2

    return i,j,k,l

def left_yaw(i,j,k,l):

    w,x,y,z=i,j,k,l

    if i<=3 or j>10 or k<=3 or l>10:
        print("highest left yaw speed.")
    else:
```

```
        while w>=i-0.5 and x<=j+0.5 and y>=k-0.5 and z<=l+0.5:

            p.ChangeDutyCycle(w)
            sleep(0.05)
            w -= 0.01

            q.ChangeDutyCycle(x)
            sleep(0.05)
            x += 0.01

            r.ChangeDutyCycle(y)
            sleep(0.05)
            y -= 0.01

            s.ChangeDutyCycle(z)
            sleep(0.05)
            z += 0.01

    i,j,k,l=round (w*2)/2, round (x*2)/2, round (y*2)/2, round (z*2)/2

    return i,j,k,l
def landing(i,j,k,l):

    maximum=max(i,j,k,l)
    if maximum>10:
        maximum=10
    while not i==j==k==l:
        if i>maximum:
            p.ChangeDutyCycle(i)
            sleep(0.05)
            i -= 0.01
        if j>maximum:
            p.ChangeDutyCycle(j)
            sleep(0.05)
            j -= 0.01
        if k>maximum:
            p.ChangeDutyCycle(k)
            sleep(0.05)
            k -= 0.01
        if l>maximum:
            p.ChangeDutyCycle(l)
            sleep(0.05)
            l -= 0.01
        if i<maximum:
            p.ChangeDutyCycle(i)
            sleep(0.05)
            i += 0.01
        if j<maximum:
            p.ChangeDutyCycle(j)
            sleep(0.05)
```

```
            j += 0.01
        if k<maximum:
            p.ChangeDutyCycle(k)
            sleep(0.05)
            k += 0.01
        if l<maximum:
            p.ChangeDutyCycle(l)
            sleep(0.05)
            l += 0.01

    i,j,k,l =round(i*2)/2,round(j*2)/2,round(k*2)/2,round(l*2)/2
    w,x,y,z=i,j,k,l

    while w>3 or x>3 or y>3 or z>3:

        p.ChangeDutyCycle(w)
        sleep(0.05)
        w -= 0.01

        q.ChangeDutyCycle(x)
        sleep(0.05)
        x -= 0.01

        r.ChangeDutyCycle(y)
        sleep(0.05)
        y -= 0.01

        s.ChangeDutyCycle(z)
        sleep(0.05)
        z -= 0.01

#---------Semi Main Code---------

def shutdown():
    p.stop()
    q.stop()
    r.stop()
    s.stop()
    GPIO.cleanup()

def video():
    cv2.namedWindow("preview")
    vc = cv2.VideoCapture(0)

    if vc.isOpened(): # try to get the first frame
        rval, frame = vc.read()
    else:
        rval = False

    while rval:
```

```
        cv2.imshow("preview", frame)
        rval, frame = vc.read()
        frame = cv2.flip (frame,0)
        key = cv2.waitKey(20)
        if key == 27: # exit on ESC
            shutdown()
            break

    vc.release()
    cv2.destroyWindow("preview")

def inp():
    global start_t, choice
    choice=None
    start_t=time()
    choice=input("G/H/I/J/K/L/M/N : ").capitalize()

def tim(*ijkl):
    ijkl=list(ijkl)
    i=ijkl[0]
    j=ijkl[1]
    k=ijkl[2]
    l=ijkl[3]
    sec=18
    passed = time()-start_t
    if passed>sec and choice==None:
        print("Please enter input within 15 seconds. Now landing....")
        landing(i,j,k,l)
        os._exit(1)

#---------Main Code---------

v=Process(target=video, name='video')
v.start()

while True:
    t1= Thread(target=inp)
    t1.start()
    sleep(20)
    t2=Thread(target=tim, args=(i,j,k,l))
    t2.start()

    if choice=='F':
        pass

    elif choice=='G':
        i,j,k,l=down(i,j,k,l)

    elif choice== 'H':
```

```
        i,j,k,l=up(i,j,k,l)

    elif choice=='I':
        i,j,k,l=forward(i,j,k,l)

    elif choice=='J':
        i,j,k,l=left(i,j,k,l)

    elif choice=='K':
        i,j,k,l=backward(i,j,k,l)

    elif choice =='L':
        i,j,k,l=right(i,j,k,l)

    elif choice=='M':
        i,j,k,l=right_yaw(i,j,k,l)

    elif choice=='N':
        i,j,k,l=left_yaw(i,j,k,l)

    else:
        landing(i,j,k,l)
        shutdown()
        break

if v.is_alive():
    v.terminate()
```