

A Project Report

On

**SURVEILLANCE QUADCOPTER WITH FACIAL RECOGNITION USING
4G COMMUNICATION**

Submitted in partial fulfilment of the requirements for the award of the degree of

**BACHELOR OF ENGINEERING
IN
ELECTRICAL AND ELECTRONICS ENGINEERING
BY**

V.NAGA VENKATA SAI :	160114734036
G.SHREYA REDDY :	160114734012
B.HARISH :	160114734024

Under the esteemed guidance of

Dr. P. KOUSTHUBA
Asst. Professor
Department of EEE, CBIT,
Hyderabad.



Department of Electrical and Electronics Engineering
Chaitanya Bharathi Institute of Technology(A)
(Affiliated to Osmania University)

Gandipet, Hyderabad-
500075
2017- 2018



Chaitanya Bharathi Institute of Technology

(Affiliated to Osmania University : accredited by NBA (AICTE), ISO Certified 9001:2000)

Chaitanya Bharathi P.O., Gandipet, Hyderabad - 500 075.

e.mail: ircrcbit@yahoo.com; Website : www.cbbit.ac.in

☎ (O) 08413-233356; 958413-233356; 958413-233865; Fax : 958413 - 232099



CERTIFICATE

This is to certify that the project work entitled "**SURVEILLANCE QUADCOPTER
WITH FACIAL RECOGNITION USING 4G COMMUNICATION**" is a
bonafidework carried out by

V.NAGA VENKATA SAI :	160114734036
G.SHREYA REDDY :	160114734012
B.HARISH :	160114734024

In partial fulfilment of the requirement for the degree **BACHELOR OF
ENGINEERING** in **ELECTRICAL & ELECTRONICS ENGINEERING** by the
Chaitanya Bharathi Institute of Technology, Osmania University, Hyderabad during the
academic year 2017-2018.

The results embedded in this report have not been submitted to any other university or
institution for the award of any degree or diploma.

DECLARATION

We, Varanasi Naga Venkata sai, Boya Harish, and G.Shreya Reddy students of Bachelor of Technology under Department of Electrical & Electronics Engineering of Chaitanya Bharathi Institute of Technology, Gandipet, hereby declare that all the information furnished in this capstone project report is based on our own intensive research and is genuine.

BATCH NO:14

G.Shreya Reddy :160114734012

Boya Harish :160114734024

Varanasi Naga Venkata Sai :160114734036

ACKNOWLEDGMENT

We have made a lot of efforts in making this project. However, it would not have been possible without the kind support of the fellow students and teachers.

We would like to extend my sincere thanks to all of them. We highly pay our gratitude for the unconditional support Dr.P.Kowstubha for her guidance and constant supervision as well as for providing necessary information regarding the project & also for her support in completing our project.

We are grateful to Dr.P.V. Prasad, HOD of the Department of Electrical And Electronics engineering and Dr.P.V. Balasubrahmanyam for permitting us to make use of the facilities available in the department to carry out the project successfully .

We would like to express our gratitude towards our parents & member of EEE department for their kind co-operation and encouragement which help us in completion of this project.

We also would like to thank the Chaitanya Bharathi Institute Of Technology for providing the support of their fully functional labs.

ABSTRACT

Quadcopter can achieve vertical flight in a stable manner and be used to monitor or collect data in a specific region such as mapping terrains. Technological advances have reduced the cost and increase the performance of the low power microcontrollers that allowed the general public to develop their own Quadcopter.

The goal of this project is to build, modify, and make improvements in Quadcopter design to obtain stable flight, 4G video streaming, Facial recognition . The project used a Quadcopter that included a frame, motors, electronic speed controllers, raspberry pi3, pi-camera, navio sensor board. Batteries, a transmitter, a receiver, and Pi-camera were interfaced with the Quadcopter's frame. Individual components were tested and verified to work properly. This report also described the auto-commands, live video streaming and facial recognition that was implemented. All the goals in this project have been achieved, resulting in a stable and manoeuvrable Quadcopter flight.

TABLE OF CONTENTS

Contents	Page.No
Chapter:1 Introduction	
1.1 History	12
1.1.2 Recent Development	13
1.2 Objective	10
1.3 Importance Of The Project	11
1.4 Organization Of The Project	11
1.5 Summary	
Chapter: 2 Components Used In The Project	
2.1 Working Principle	14
2.1.1 PID	15
2.1.2 Pulse Width Modulation	16
2.1.3 MAV link	17
2.1.4 S. bus	17
2.2 Hardware And Software	17
2.2.1 Frame	17
2.2.2 Selected Frame	18
2.2.3 Propeller	18
2.2.4 BLDC Motors	18
2.2.5 Electronic Speed Controllers	20
2.2.6 Power Supply	22
2.2.7 Controller	23
2.2.8 Ardu Pilot Software	26
2.2.9 Calibration	28
2.2.10 RC Transmitter And Receiver	28
2.3 Flight Modes	30
2.4 Summary	
Chapter 3: Working Principle of quadcopter	
3.1 Principle of Operation	40
3.2 Processing Of Receiving Signal	43
3.3 Calculations	
3.4 Calibrated Values	
3.5 Summary	
Chapter 4 : Results	
6.1 Results	
Chapter 5: Conclusions And Future Scope	
6.1 Conclusions	46
6.1 Future Scope	46

Apendix 1

Apendix 2

LIST OF FIGURES

Figure. No	Figure name	Page. No
Figure 2.1	Connection diagram	13
Figure 2.2	Telemetry Diagram	14
Figure 2.3	BLDC Motor	15
Figure 2.4	Simplified ESC	19
Figure 2.5	ESC Connections	20
Figure 2.6	Navio Sensors	24
Figure 2.7	Ardupilot Flow Diagram	25
Figure 2.8	RC Transmitter And Receiver	28
Figure 2.9	Flight Modes	30
Figure 3.1	Take Off Motion	39
Figure 3.2	Landing Motion	39
Figure 3.3	Forward Motion	40
Figure 3.4	Backward Motion	40
Figure 3.5	Right Motion	40
Figure 3.6	Left Motion	40
Figure 3.7	Pitch Direction	41
Figure 3.8	Roll Direction	41
Figure 3.9	Yaw Direction	41
Figure 3.10	Pulse Width Modulation	42
Figure 3.11	Pulse Position Modulation	43
Figure 3.12	PWM To PPM Modulation	43
Figure 5.1	Our quadcopter	46

LIST OF TABLES

Table. No	Table Name	Page. No
Table 3.1	Weight Measurement	16

CHAPTER 1: INTRODUCTION

Over the last few years we have seen a massive growth in the manufacture and sales of remote control airborne vehicles known as Quadcopters. These Unmanned Aerial Vehicles have four arms and fixed pitch propellers which are set in an X or + configuration with X being the preferred configuration. They are sometimes referred to as Drones, Quadrotors or Quadrocopters. In the standard format two propellers will spin in a clockwise direction with the other two spinning in an anticlockwise direction allowing the craft to vertically ascend, hover in the air and fly in a designated direction. By adding four motors and four propellers to a lightweight frame constructed of light wood, carbon fiber, or fiber glass then connecting it to a remote control transmitter via a small control board fitted with a gyroscopic stabilization system and connected to a Lipo battery these craft were relatively simple to construct. Experimentation has led to the configuration of variations of the Quadcopter by using different amounts of arms we have seen Tricopters, Hex copters and Octocopters (with eight arms). Other configurations include a Vtail and an H frame variation.

The rapid advances in computing power, the efficiency of the coreless or brushless motors, smaller microprocessors the development of batteries and gyroscopic and accelerometer technology has all led to a proliferation of Quadcopter designs. The first Quadcopters were not designed for acrobatic flight as the development was concentrated on simple stable flight patterns but now this has all changed. Micro and even Nano Quadcopters are being produced mainly in China that can perform intricate aerobatic moves, flips and barrel rolls that years ago would have been unthinkable. In the early days of flight, quadcopters (then referred to as 'quadrotors') were seen as possible solutions to some of the persistent problems in vertical flight; torque-induced control issues (as well as efficiency issues originating from the tail rotor, which generates no useful lift) can be eliminated by counter-rotation and the relatively short blades are much easier to construct.

1.1 Literature survey

The quad rotor project required extensive research into similar systems. By reviewing others work, we used this insight to develop our system. To this end, research papers from various quad rotor groups were used as guides in the early development of the dynamics and control theory. Quad rotor platforms used in research remain somewhat the same, having four electric motors pointed vertically upwards and equally spaced in a square fashion. However, there were some groups whom designed their own platforms, whereas commercial models available to the consumer were the Dragan Flyer, the X-UFO and the MD4-200.

1.1.1 History:

1898:

At an exhibition at Madison Square Garden, inventor Nikola Tesla wows audiences with a small unmanned boat that appears to change direction on verbal command. (He's actually using radio frequencies to switch motors on and off).

1940:

Actor and hobbyist Reginald Denny sells 15,000 radio-controlled target drones to the U.S. military to train anti-aircraft gunners for World War II.

1943:

The German military debuts the FX-1400, or "the Fritz X," a 2,300-pound bomb with four small wings and a radio controller. A breakthrough for guided aerial weapons, it also was the first remotely controlled munition put into operational use.

1944:

Photographer David Conover snaps images of Norma Jeane Dougherty—soon to be known as Marilyn Monroe—working at a wartime assembly plant. On her workbench: a half-assembled drone.

1960s:

The size and cost of transistors plummet, ushering in an age of radio-controlled products and a generation of tinkerers and enthusiasts.

1982:

At the outbreak of hostilities with Syria, Israel outwits Soviet anti-aircraft technology by using a swarm of unmanned aircraft to draw it into revealing its location.

1995:

The General Atomics MQ-1 Predator UAV enters service. In the decade following 9/11, the Predator will become the public face of drones.

2010:

The Parrot AR Drone, a smartphone-controlled quadcopter for consumers, is introduced at the Consumer Electronics Show in Las Vegas.

2012:

Congress requires the FAA to integrate small drones into the national airspace by 2015.

2013:

On 60 Minutes, Amazon CEO Jeff Bezos unveils an audacious plan: delivery drones and a future service called Prime Air.

2014:

As the FAA grants an exemption to film and TV production companies for drone use, producer Jerry Bruckheimer reveals that actor Tom Cruise will star in Top Gun 2 alongside drones.

1.1.2 Recent development:

In the last few decades, small-scale unmanned aerial vehicles have been used for many applications. The need for aircraft with greater maneuverability and hovering ability has led to a rise in quadcopter research. The four-rotor design allows quadcopters to be relatively simple in design yet highly reliable and maneuverable. Research is continuing to increase the abilities of quadcopters by making advances in multi-craft communication, environment exploration, and maneuverability. If these developing

qualities can be combined, quadcopters would be capable of advanced autonomous missions that are currently not possible with other vehicles.

Some current programs include:

- The Bell Boeing Quad Tilt Rotor concept takes the fixed quadcopter concept further by combining it with the tilt rotor concept for a proposed C-130 sized military transport.
- AeroQuad and ArduCopter are open-source hardware and software projects based on Arduino for the DIY construction of quadcopters.
- Nixie is a small camera-equipped drone that can be worn as a wrist band.
- Airbus is developing a battery-powered quadcopter to act as an urban air taxi, at first with a pilot but potentially autonomous in the future.

1.2 Objective

The goal of our project is to design, implement, and test a stable flying Quadcopter. Our plan was to choose an existing Quadcopter kit and add the required components to give the Quadcopter the capabilities to gather and log data autonomously. Using 4G modem live video streaming is established and can be accessed from any device with internet connectivity. Using Dlib's machine learning API to achieve facial recognition i.e., detecting and identifying faces. If this goal is accomplished, our team would also like to add a feature to the quadcopter camera that makes it to focus on a particular face as required. The final Quadcopter design had to meet the following specifications:

1. The Quadcopter must be capable of flying and landing in stable manner.
2. The Quadcopter must be capable of sending the data through 4g network.
3. The Quadcopter must be capable to storing and logging data.
4. The Quadcopter must be able to perform the following commands:
 - Stable flight control
 - 4G video streaming

- Hold position

5. The Quadcopter camera must achieve facial recognition.

Stages of project

Stage 1: Design and construct a mechanically balance robust quadcopter frame

Stage 2: Install controllers into the frame and stabilize flight with wi-fi.

Stage 3: Extend the surveillance with 4G modem with less latency video streaming.

Stage 4: Add the facial recognition feature to the quadcopter.

1.3 Importance of the project

Personal Drones have been all the rage for the past few years, as toys, and primarily as new devices for capturing amazing aerial photography. As the technology has matured and become more mainstream, a number of practical and very interesting uses of Drone technology have emerged. In the past few months we have seen some amazing developments in the flying drone industry. Amazon has announced a service, which will deliver your orders right to your door, and 3D Robotics, a commercial drone maker, has received \$36 million in funding. The future of drones flying around everywhere is coming closer and closer to us.

1.4 Organization of the report:

Chapter 1: In chapter 1 introduction to the quadcopters is given. In this chapter literature survey, objective and importance of the project are discussed.

Chapter 2: In this chapter Controlling the quadcopter for stable flight and how power is distributed for each component is discussed.

Chapter 3: In this chapter .

Chapter 4: In this chapter results, conclusion and future scope are discussed.

Summary of the Chapter 1:

In this chapter introduction to the quadcopter technology is discussed initially. Then the main objectives and the importance of this project are discussed. Finally a quick explanation of organization of the report is given.

A Project Report

On

**SURVEILLANCE QUADCOPTER WITH FACIAL RECOGNITION USING
4G COMMUNICATION**

Submitted in partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF ENGINEERING
IN
ELECTRICAL AND ELECTRONICS ENGINEERING
BY**

V.NAGA VENKATA SAI :	160114734036
G.SHREYA REDDY :	160114734012
B.HARISH :	160114734024

Under the esteemed guidance of

**Dr. P. KOUSTHUBA
Asst. Professor**

**Department of EEE, CBIT,
Hyderabad.**




**Department of Electrical and Electronics Engineering
Chaitanya Bharathi Institute of Technology(A)
(Affiliated to Osmania University)**

**Gandipet, Hyderabad-
500075
2017- 2018**



Chaitanya Bharathi Institute of Technology
 (Affiliated to Osmania University : accredited by NBA (AICTE), ISO Certified 9001:2000)
Chaitanya Bharathi P.O., Gandipet, Hyderabad - 500 075.
 e.mail: ircrcbit@yahoo.com; Website : www.cbti.ac.in
 ☎ (O) 08413-233356; 958413-233356; 958413-233865; Fax : 958413 - 232099



CERTIFICATE

This is to certify that the project work entitled "**SURVEILLANCE
QUADCOPTER WITH FACIAL RECOGNITION USING 4G
COMMUNICATION**" is a bonafidework carried out by

V.NAGA VENKATA SAI :	160114734036
G.SHREYA REDDY :	160114734012
B.HARISH :	160114734024

In partial fulfilment of the requirement for the degree **BACHELOR OF ENGINEERING** in **ELECTRICAL & ELECTRONICS ENGINEERING** by the Chaitanya Bharathi Institute of Technology, Osmania University, Hyderabad during the academic year 2017-2018.

The results embedded in this report have not been submitted to any other university or institution for the award of any degree or diploma.

DECLARATION

We, Varanasi Naga Venkata sai, Boya Harish, and G.Shreya Reddy students of Bachelor of Technology under Department of Electrical & electronics of Chaitanya Bharathi Institute Of Technology, Gandipet, hereby declare that all the information furnished in this capstone project report is based on our own intensive research and is genuine.

BATCH NO:14

G.Shreya Reddy :160114734012

Boya Harish :160114734024

Varanasi Naga Venkata Sai :160114734036

ACKNOWLEDGMENT

We have made a lot of efforts in making this project. However, it would not have been possible without the kind support of the fellow students and teachers.

We would like to extend my sincere thanks to all of them. We highly pay our gratitude for the unconditional support Dr.P.Kowstubha for her guidance and constant supervision as well as for providing necessary information regarding the project & also for her support in completing our project.

We are grateful to Dr.P.V. Prasad, HOD of the Department of Electrical And Electronics engineering and Dr.P.V. Balasubrahmanyam for permitting us to make use of the facilities available in the department to carry out the project successfully .

We would like to express our gratitude towards our parents & member of EEE department for their kind co-operation and encouragement which help us in completion of this project.

We also would like to thank the Chaitanya Bharathi Institute Of Technology for providing the support of their fully functional labs.

ABSTRACT

Quadcopter can achieve vertical flight in a stable manner and be used to monitor or collect data in a specific region such as mapping terrains. Technological advances have reduced the cost and increase the performance of the low power microcontrollers that allowed the general public to develop their own Quadcopter.

The goal of this project is to build, modify, and make improvements in Quadcopter design to obtain stable flight, 4G video streaming, Facial recognition . The project used a Quadcopter that included a frame, motors, electronic speed controllers, raspberry pi3, pi-camera, navio sensor board. Batteries, a transmitter, a receiver, and Pi-camera were interfaced with the Quadcopter's frame. Individual components were tested and verified to work properly. This report also described the auto-commands, live video streaming and facial recognition that was implemented. All the goals in this project have been achieved, resulting in a stable and manoeuvrable Quadcopter flight.

TABLE OF CONTENTS

Contents	Page.No
Chapter:1 Introduction	
1.1 History	12
1.1.2 Recent Development	13
1.2 Objective	10
1.3 Importance Of The Project	11
1.4 Organization Of The Project	11
1.5 Summary	
Chapter: 2 Components Used In The Project	
2.1 Working Principle	14
2.1.1 PID	15
2.1.2 Pulse Width Modulation	16
2.1.3 MAV link	17
2.1.4 S. bus	17
2.2 Hardware And Software	17
2.2.1 Frame	17
2.2.2 Selected Frame	18
2.2.3 Propeller	18
2.2.4 BLDC Motors	18
2.2.5 Electronic Speed Controllers	20
2.2.6 Power Supply	22
2.2.7 Controller	23
2.2.8 Ardu Pilot Software	26
2.2.9 Calibration	28
2.2.10 RC Transmitter And Receiver	28
2.3 Flight Modes	30
2.4 Summary	
Chapter 3: Working Principle of quadcopter	
3.1 Principle of Operation	40
3.2 Processing Of Receiving Signal	43
3.3 Calculations	
3.4 Calibrated Values	
3.5 Summary	
Chapter 4 : Results	
6.1 Results	

Chapter 5: Conclusions And Future Scope

6.1	Conclusions	46
6.1	Future Scope	46
6.2	References	47

Apendix 1**Apendix 2**

LIST OF FIGURES

Figure. No	Figure name	Page. No
Figure 2.1	Connection diagram	13
Figure 2.2	Telemetry Diagram	14
Figure 2.3	BLDC Motor	15
Figure 2.4	Simplified ESC	19
Figure 2.5	ESC Connections	20
Figure 2.6	Navio Sensors	24
Figure 2.7	Ardupilot Flow Diagram	25
Figure 2.8	RC Transmitter And Receiver	28
Figure 2.9	Flight Modes	30
Figure 3.1	Take Off Motion	39
Figure 3.2	Landing Motion	39
Figure 3.3	Forward Motion	40
Figure 3.4	Backward Motion	40
Figure 3.5	Right Motion	40
Figure 3.6	Left Motion	40
Figure 3.7	Pitch Direction	41
Figure 3.8	Roll Direction	41
Figure 3.9	Yaw Direction	41
Figure 3.10	Pulse Width Modulation	42
Figure 3.11	Pulse Position Modulation	43
Figure 3.12	PWM To PPM Modulation	43
Figure 5.1	Our quadcopter	46

LIST OF TABLES

Table. No	Table Name	Page. No
Table 3.1	Weight Measurement	16

CHAPTER 1: INTRODUCTION

Over the last few years we have seen a massive growth in the manufacture and sales of remote control airborne vehicles known as Quadcopters. These Unmanned Aerial Vehicles have four arms and fixed pitch propellers which are set in an X or + configuration with X being the preferred configuration. They are sometimes referred to as Drones, Quadrotors or Quadrocopters. In the standard format two propellers will spin in a clockwise direction with the other two spinning in an anticlockwise direction allowing the craft to vertically ascend, hover in the air and fly in a designated direction. By adding four motors and four propellers to a lightweight frame constructed of light wood, carbon fiber, or fiber glass then connecting it to a remote control transmitter via a small control board fitted with a gyroscopic stabilization system and connected to a Lipo battery these craft were relatively simple to construct. Experimentation has led to the configuration of variations of the Quadcopter by using different amounts of arms we have seen Tricopters, Hex copters and Octocopters (with eight arms). Other configurations include a Vtail and an H frame variation.

The rapid advances in computing power, the efficiency of the coreless or brushless motors, smaller microprocessors the development of batteries and gyroscopic and accelerometer technology has all led to a proliferation of Quadcopter designs. The first Quadcopters were not designed for acrobatic flight as the development was concentrated on simple stable flight patterns but now this has all changed. Micro and even Nano Quadcopters are being produced mainly in China that can perform intricate aerobatic moves, flips and barrel rolls that years ago would have been unthinkable. In the early days of flight, quadcopters (then referred to as 'quadrotors') were seen as possible solutions to some of the persistent problems in vertical flight; torque-induced control issues (as well as efficiency issues originating from the tail rotor, which generates no useful lift) can be eliminated by counter-rotation and the relatively short blades are much easier to construct.

1.1 Literature survey

The quad rotor project required extensive research into similar systems. By reviewing others work, we used this insight to develop our system. To this end, research papers from various quad rotor groups were used as guides in the early development of the dynamics and control theory. Quad rotor platforms used in research remain somewhat the same, having four electric motors pointed vertically upwards and equally spaced in a square fashion. However, there were some groups whom designed their own platforms, whereas commercial models available to the consumer were the Dragan Flyer, the X-UFO and the MD4-200.

1.1.2 History:

1898:

At an exhibition at Madison Square Garden, inventor Nikola Tesla wows audiences with a small unmanned boat that appears to change direction on verbal command. (He's actually using radio frequencies to switch motors on and off).

1940:

Actor and hobbyist Reginald Denny sells 15,000 radio-controlled target drones to the U.S. military to train anti-aircraft gunners for World War II.

1943:

The German military debuts the FX-1400, or "the Fritz X," a 2,300-pound bomb with four small wings and a radio controller. A breakthrough for guided aerial weapons, it also was the first remotely controlled munition put into operational use.

1944:

Photographer David Conover snaps images of Norma Jeane Dougherty—soon to be known as Marilyn Monroe—working at a wartime assembly plant. On her workbench: a half-assembled drone.

1960s:

The size and cost of transistors plummet, ushering in an age of radio-controlled products and a generation of tinkerers and enthusiasts.

1982:

At the outbreak of hostilities with Syria, Israel outwits Soviet anti-aircraft technology by using a swarm of unmanned aircraft to draw it into revealing its location.

1995:

The General Atomics MQ-1 Predator UAV enters service. In the decade following 9/11, the Predator will become the public face of drones.

2010:

The Parrot AR Drone, a smartphone-controlled quadcopter for consumers, is introduced at the Consumer Electronics Show in Las Vegas.

2012:

Congress requires the FAA to integrate small drones into the national airspace by 2015.

2013:

On 60 Minutes, Amazon CEO Jeff Bezos unveils an audacious plan: delivery drones and a future service called Prime Air.

2014:

As the FAA grants an exemption to film and TV production companies for drone use, producer Jerry Bruckheimer reveals that actor Tom Cruise will star in Top Gun 2 alongside drones.

1.1.2 Recent development:

In the last few decades, small-scale unmanned aerial vehicles have been used for many applications. The need for aircraft with greater maneuverability and hovering ability has led to a rise in quadcopter research. The four-rotor design allows quadcopters to be relatively simple in design yet highly reliable and maneuverable. Research is continuing to increase the abilities of quadcopters by making advances in multi-craft communication, environment exploration, and maneuverability. If these developing qualities can be combined, quadcopters would be capable of advanced autonomous missions that are currently not possible with other vehicles.

Some current programs include:

- The Bell Boeing Quad Tilt Rotor concept takes the fixed quadcopter concept further by combining it with the tilt rotor concept for a proposed C-130 sized military transport.
- AeroQuad and ArduCopter are open-source hardware and software projects based on Arduino for the DIY construction of quadcopters.
- Nixie is a small camera-equipped drone that can be worn as a wrist band.
- Airbus is developing a battery-powered quadcopter to act as an urban air taxi, at first with a pilot but potentially autonomous in the future.

1.2 Objective

The goal of our project is to design, implement, and test a stable flying Quadcopter. Our plan was to choose an existing Quadcopter kit and add the required components to give the Quadcopter the capabilities to gather and log data autonomously. Using 4G modem live video streaming is established and can be accessed from any device with internet connectivity. Using Dlib's machine learning API to achieve facial recognition i.e., detecting and identifying faces. If this goal is accomplished, our team would also like to add a feature to the quadcopter camera that

makes it to focus on a particular face as required. The final Quadcopter design had to meet the following specifications:

1. The Quadcopter must be capable of flying and landing in stable manner.
2. The Quadcopter must be capable of sending the data through 4g network.
3. The Quadcopter must be capable to storing and logging data.
4. The Quadcopter must be able to perform the following commands:
 - Stable flight control
 - 4G video streaming
 - Hold position
5. The Quadcopter camera must achieve facial recognition.

Stages of project

Stage 1: Design and construct a mechanically balance robust quadcopter frame

Stage 2: Install controllers into the frame and stabilize flight with wi-fi.

Stage 3: Extend the surveillance with 4G modem with less latency video streaming.

Stage 4: Add the facial recognition feature to the quadcopter.

1.3 Importance of the project

Personal Drones have been all the rage for the past few years, as toys, and primarily as new devices for capturing amazing aerial photography. As the technology has matured and become more mainstream, a number of practical and very interesting uses of Drone technology have emerged. In the past few months we have seen some amazing developments in the flying drone industry. Amazon has announced a service, which will deliver your orders right to your door, and 3D Robotics, a commercial

drone maker, has received \$36 million in funding. The future of drones flying around everywhere is coming closer and closer to us.

1.4 Organization of the report:

Chapter 1: In chapter 1 introduction to the quadcopters is given. In this chapter literature survey, objective and importance of the project are discussed.

Chapter 2: In this chapter controlling of quadcopter to achieve stable and controlled flight and how the power is distributed among components used are discussed.

Chapter 3: In this chapter how IoT services and other services are used to achieve telemetry, video-streaming and facial recognition are discussed

Chapter 4: In this chapter results, conclusion and future scope are discussed.

Summary of the Chapter 1:

In this chapter introduction to the quadcopter technology is discussed initially. Then the main objectives and the importance of this project are discussed. Finally a quick explanation of organization of the report is given.

CHAPTER 2: CONTROLLING AND POWER DISTRIBUTION

2.1 Principle of operation & control:

2.1.1 Principle of operation

A propeller is a type of fan that transmits power by converting motion into thrust. Propeller dynamics can be modelled by both Bernoulli's principle and Newton's third law. Bernoulli's principle states that for an inviscid flow, an increase in the speed of the fluid occurs simultaneously with a decrease in pressure or a decrease in the fluid's potential energy. Newton's third law states that every action has an equal and opposite reaction. An aero foil of a propeller is shaped so that air flows faster over the top than under the bottom. There is, therefore, a greater pressure below the aero foil than above it. This difference in pressure produces the lift. Lift coefficient is a dimensionless coefficient that relates the lift generated by an aerodynamic body such as a wing or complete aircraft, the dynamic pressure of the fluid flow around the body, and a reference area associated with the body.

2.1.2 Controlling principle

Quadcopter can be described as a small vehicle with four propellers attached to the root located at the cross frame. This aim for fixed rotors is used to control the vehicle motion. The speeds of these four rotors are independent. By independent pitch, roll and yaw attitude of the vehicle can be controlled easily. Pitch, roll and yaw attitude of Quadcopter. Hovering or static position: The hovering or static position of the Quadcopter is done by two pairs of rotors, by rotating in clockwise or counter-clockwise respectively with the same speed. By two rotors rotating in clockwise and counter-clockwise position, the total sum of reaction torque is zero and this allows the Quadcopter to be in a hovering position. Forward and backward motion Forward (backward) motion is controlled by increasing (decreasing) speed of rear (front) rotor. Decreasing (increasing) rear (front) rotor's speed simultaneously will affect the pitch angle of the Quadcopter. Left and right motion for left and right motion, it can be controlled by changing the yaw angle of the Quadcopter. Yaw angle can be controlled

by increasing (decreasing) counter-clockwise rotors speed while decreasing (increasing) clockwise rotor speed.

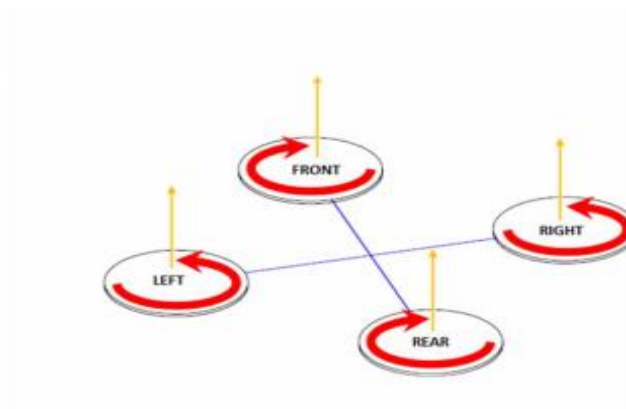


Figure 2.1 Take off Motion

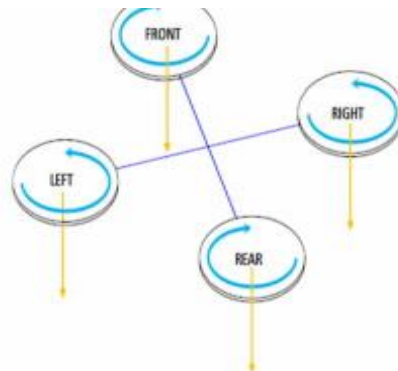


Figure 2.2 Landing Motion

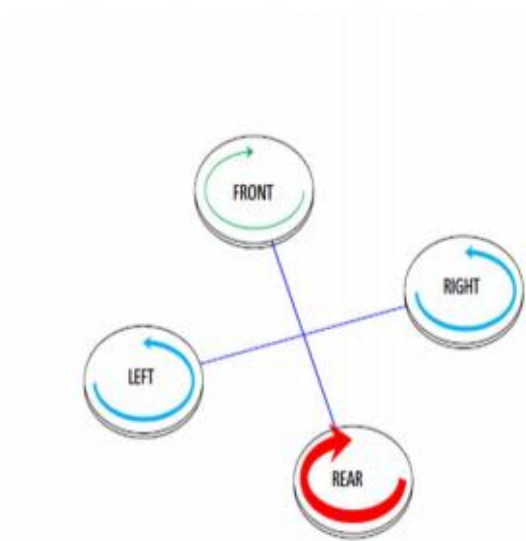
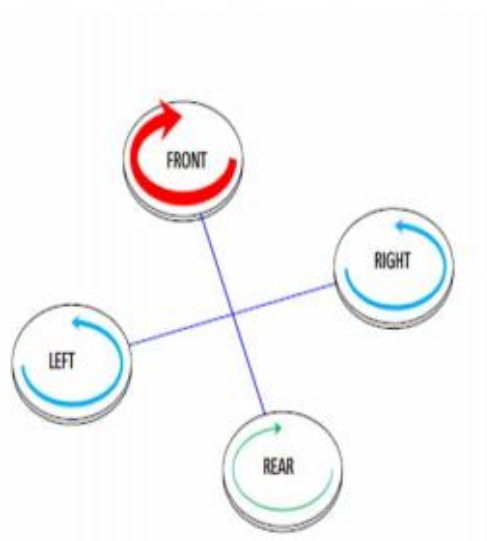
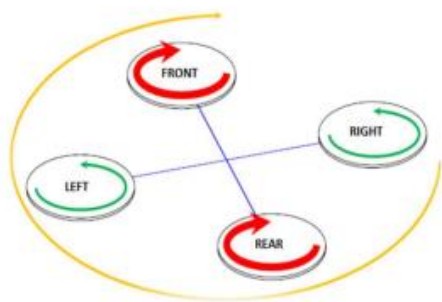


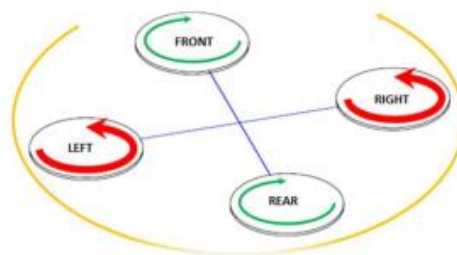
Figure 3.3 Forward Motion Figure



3.4 Back ward Motion Figure



3.5 Right Motion Figure



3.6 Left Motion

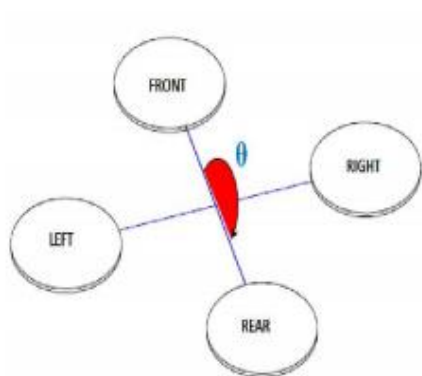


Figure 3.7 : pitch direction

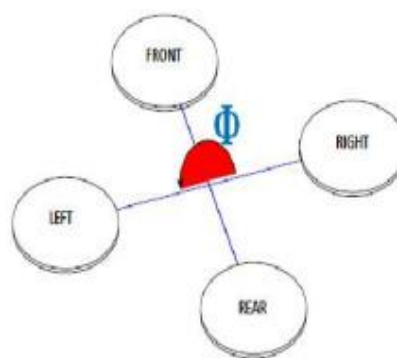


Figure 3.8 : roll direction

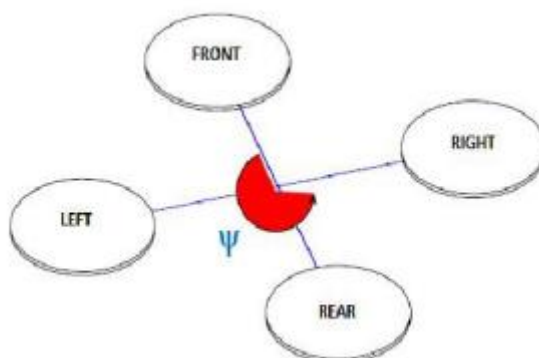


Figure 3.9 Yaw direction

The choice of the various hardware components must be based on the aircraft of total weight. I compiled a table to calculate the approximate weight (Table 2.1).

Table 2.1. Approximate weight of equipment

COMPONENT	WEIGHT
Raspberry Pi	45g
Navio 2	100g
Frame	480g
Engine speed controllers	190g
battery	270g
Propellers	30g
Bldc motors	300g
Rpi camera	15g
total	1747g

According to the weight and frame, it was necessary to choose suitable engines, ESC (Electronic Speed Controller) and propellers.

2.2 Signal processing:

Controlling the movement of quadcopter is done by sending signal from radio transmitter which is then processed by the flight controller (Navio2) using PID controller and output signal is sent to ESC(Electronic speed controller) which controls speed of BLDC motors.

2.2.1 Principle of signal processing

Modulation Technique in Quadcopter: Pulse width modulation is a way of simulating an analog output by varying HIGH and LOW signals at intervals proportional to the value. Width of each pulse varies according to the amplitude of the analog signal.

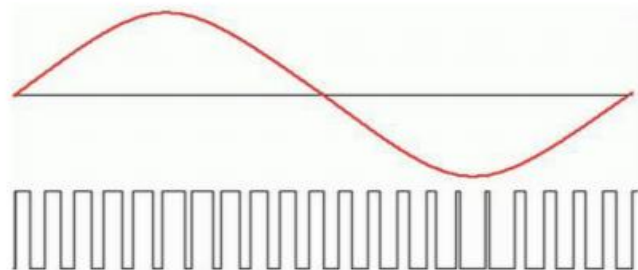


Figure 5.10 Pulse width Modulation

Pulse Position Modulation Pulse- position modulation is a form of signal modulation in which M message bits are encoded by transmitting a single pulse in one of $2M$ possible time-shifts. This is repeated every T seconds, such that the transmitted bit rate is M/T bits per second.

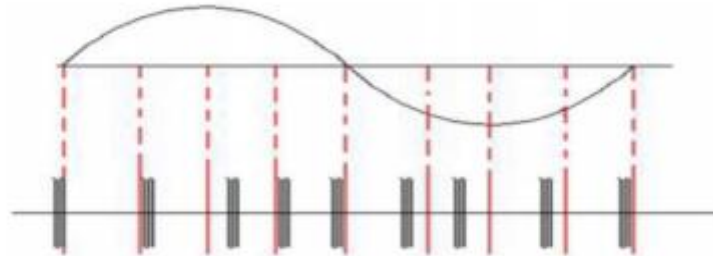


Figure 5.11 Pulse Position Modulation

PWM to PPM conversion

PWM refers to a pulse width modulation signal, where the width of each pulse changes according to the amplitude of an analog signal. PPM on the other hand refers to a pulse position modulation signal, where the width of each pulse remains the same, but each pulse is displaced by a certain position based on the analog signal amplitude. The basic need for conversion of a PWM signal received from a transmitter into a PPM signal arises due to the fact that the main controller board used (Ardu IMU) on a Quadcopter can't process a PWM signal and hence a converter is required to convert a PWM signal to a PPM signal.

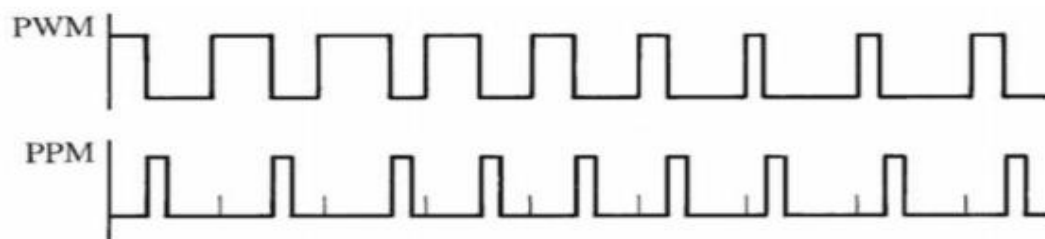


Figure 5.12: PWM to PPM conversion

PID (Proportional-Integral-Derivative) system Of Quadcopter

The PID method is used to continuously stabilize the aircraft. PID is a feedback control system that receives data from sensors and receives an output signal summing up the proportional (P), integral (I) and differential (D) component .

According to the total output signal is adjusted process to achieve the desired direction.

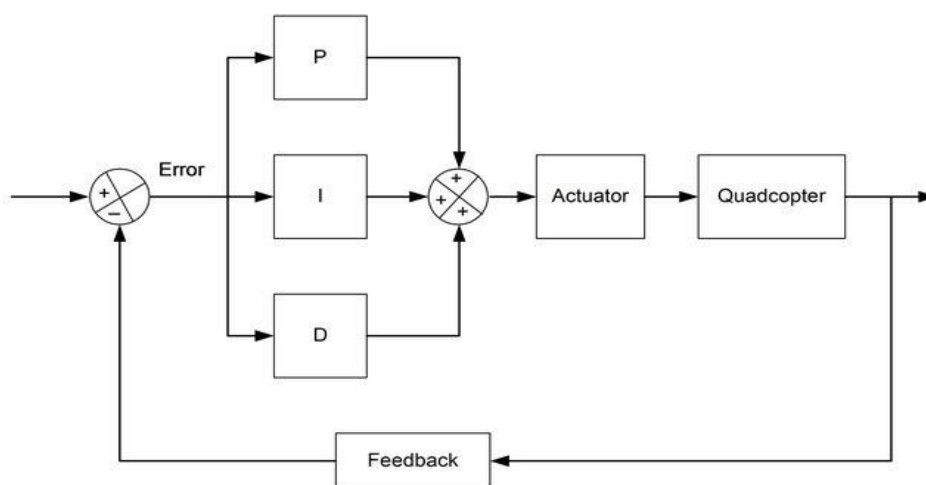


Figure 2.3: PID

The connections of the quadcopter are shown in (figures 2.1) respectively.

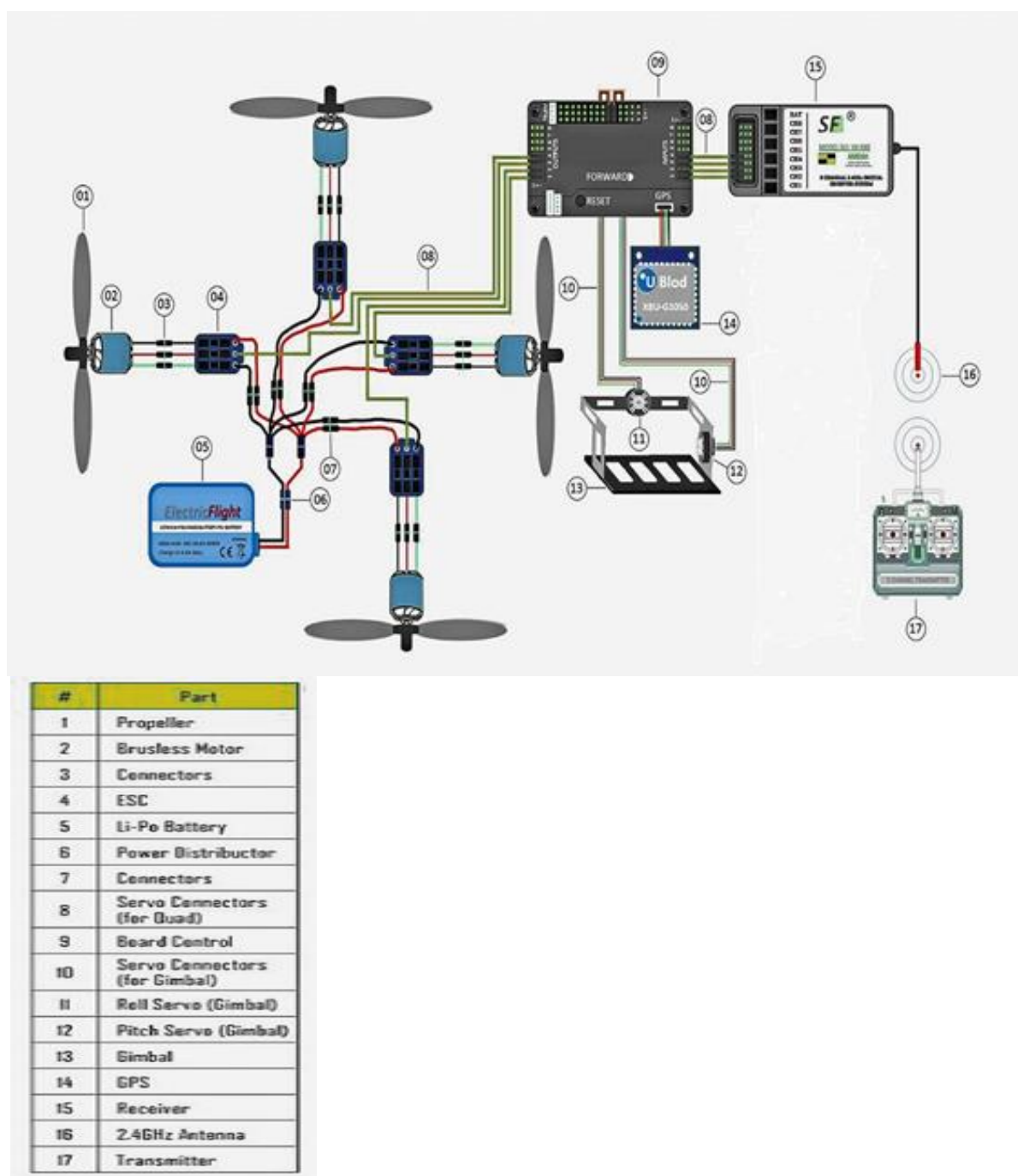


Figure 2.1: connection diagram of quadcopter

RC transmitter and receiver Of Quadcopter

The remote control of the aircraft must be equipped with an RC remote control, i.e. a transmitter and a receiver. The RC transmitter is used by FrSkyTaranis X9D Plus and the receiver FrSkyX8R. The communication takes place at 2.4 GHz frequency. The receiver is connected to the Navio2 peak band RC input via PPM.

Encoder, the input may be either S.Bus or PPM signal. FrSky receiver uses 6-channel PWM for communication with the controller. PPM Encoder is additionally attached to receiver to convert 6- channel PWM to PPM.



Figure 2.8: RC transmitter and receiver

When connecting the transmitter and the receiver, a unique connection is created between them: one receiver is connected to one specific model of the transmitter. One sender can be connected to several The receiver, one transmitter, can be used with several different aircraft (not at the same time), but one receiver can be connected with only one remote control. So even if there is another identical sender and receiver on the same frequency, there will be no conflict

Up to 6 channels can be used on a single receiver and transmitter, but ArduCopter The software supports up to 12 channels. The first 4 channels are joystick channels, respectively Roll (side to side movement), Pitch, Throttle, and Yaw (rotational movement).The fifth channel is for flight modes. Mission Planner lets you assign 6 different senders flight mode The remaining channels can be used, for example, to activate additional sensors or inactivating the camera, activating the camera, storing the location, making a splash

Engine Speed Controllers (ESC)

The ESCs adjust the rotational direction and speed of the motors by adjusting the voltage accordingly commands received from the controller, which changes the engine speed. The ESC is powered by PDB and its three cables are connected to the engine stator coils for which ESC transmits three-phase pulsed pulses. Flow stator winding In the process of conducting, a magnetic field emerges that generates a rotor when pulling permanent magnets rotation Thus, the rotational direction of the motors

can also be changed in the order of the wires exchanging. The ESC controls the engine in accordance with instructions from the aircraft controller. Made the case communicates with the controller using the PWM protocol. The protocol depends on the ESC the firmware on the microcontroller, and the speed controllers used in the job to do this.(Figure 3.4) shows internal structure of ESC.

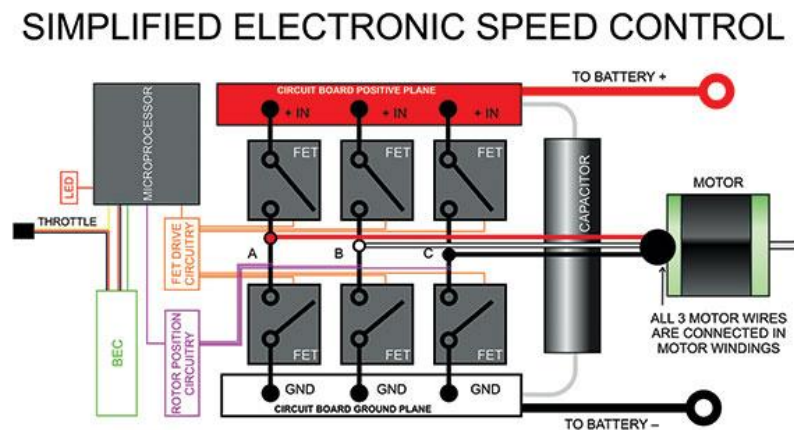


Figure 2.4:simplified ESC

BLDC Motors

Most of the UAV's engines today are brushless engines, which are compared to brushes, more durable and more efficient. Brushless motor (Figure 3.3) consists of stationary winding static and permanent magnet rotary rotors, which placed in a magnetic field. There are two types of brushless motors: inrunner and outrunner. The type of inrunner means that the rotating part of the engine is located inside and for the outrunner engine, the rotating part is located outside. Aircraft are used as a rule, outrunne rmotors Turnigy A2212/13T 1000KV brushless is selected due to the possibilities and the time limit outrunner motors.



Figure 2.3:BLDC motor

Navio2 controller and ArduPilot software

The Emlid Limited Navio2 is a Raspberry Pi expansion board that is equipped with sensors required for autonomous flight. Navio2 binds the aircraft's hardware collection and sensor data with Raspberry Pi and gives Raspberry Pi 5V power supply.

The Navio2 disc is equipped with a wide range of sensors for autonomous flying. In order to the location of the aircraft can be determined as precisely as possible, the data of these sensors must be provided to process This is the use of ArduPilot software. Ardupilot is an open source Autopilot software, with a variety of versions to use multi-engine (ArduCopter), airplanes (ArduPlane), ground vehicles(ArduRover) and watercraft (ArduSub). The main programming language is Python. In this work, it is a multi-rotor, so the used version is ArduCopter. The ArduCopter code uses the processed data from different directories. Directories includes sensor drivers (AP_Inertial Sensor, AP_Baro, AP_GPS), position Calculation (AP_AHRS), PID controller (AC_PosControl, AC_AttitudeControl.

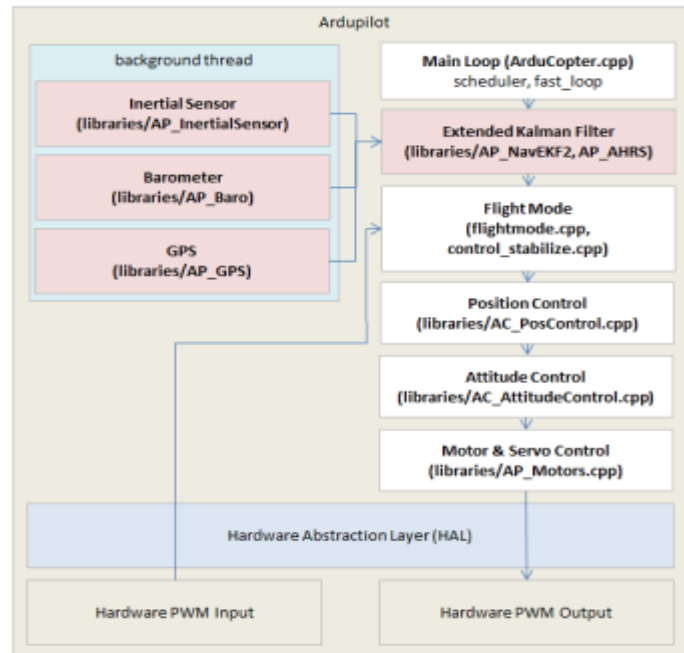


Figure 2.7: ardupilot flow diagram

Navio2 sensors

IMU is an inertia sensor that processes Mahony AHRS from three sensors (Attitude and Heading Reference System) algorithm. Navio2 IMU contains three MEMS (Micro electromechanical Systems) sensor: gyroscope, accelerometer and Magnetic sensor (compass) that perceives straight and rotary motion towards x, y and z axis. In order to increase accuracy, there are two IMUs in front of Navio2-el. Figure 8. Navio2 connections. GNSS (Global Navigation Satellite System, or global) satellite navigation system.

The gyroscope perceives angular acceleration due to the Coriolis vibration effect three axes. For an acceleration, a corresponding signal is generated and transmitted digitalized processor. The accelerometer measures linear acceleration around three axes. The sensor measures the compressive force due to acceleration for each axle. The pressure on the sensor test mass causes a change in capacitance and corresponds to the capacitance the digitalized electrical signal is transmitted to the processor. Magnetic feeder is based on Hall effect and measures the magnetic field of the axes, depending on the strength of the magnetic field course. The barometer measures altitude according to air pressure changes. Measurement works pomegranate When pressure is pressed on the piezo, this impedance increases and accordingly An electrical signal is generated for the change. The U-blox GNSS module is connected

to GPS (USA), GLONASS (Russia) and Beidou (China) satellite navigation systems. Uses the SPI protocol for data transfer. Locate and configure the data according to the controller commands.

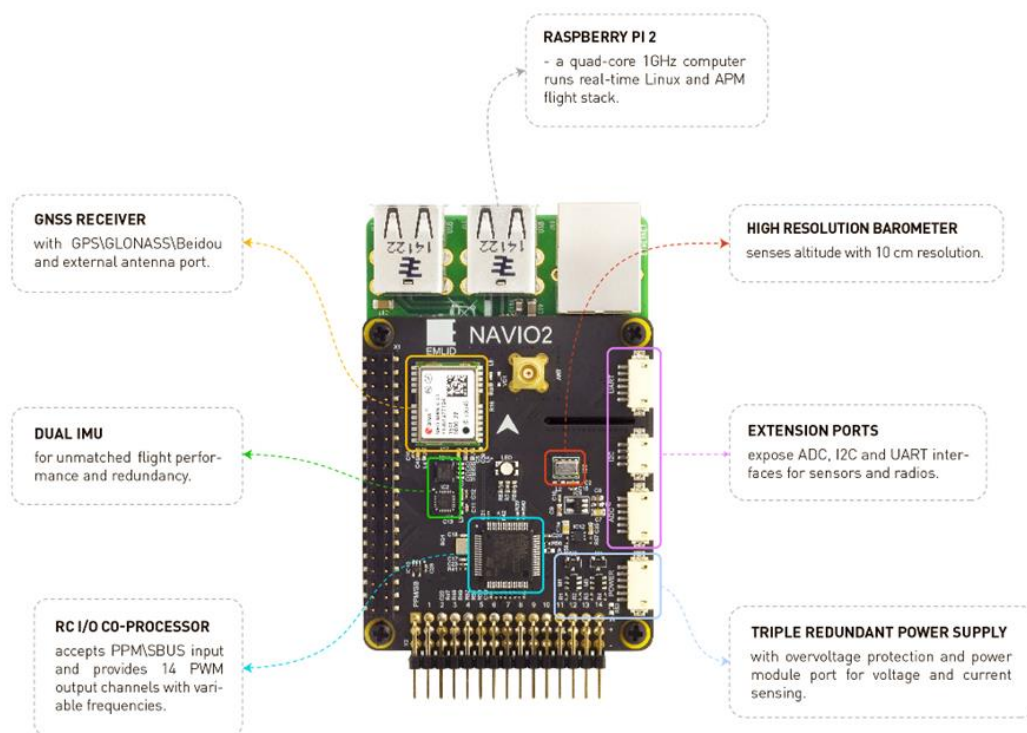


Figure 2.6:Navio sensors

2.3 Power distribution:

Whole quadcopter setup is powered by single Li-po battery with different components using different voltages.

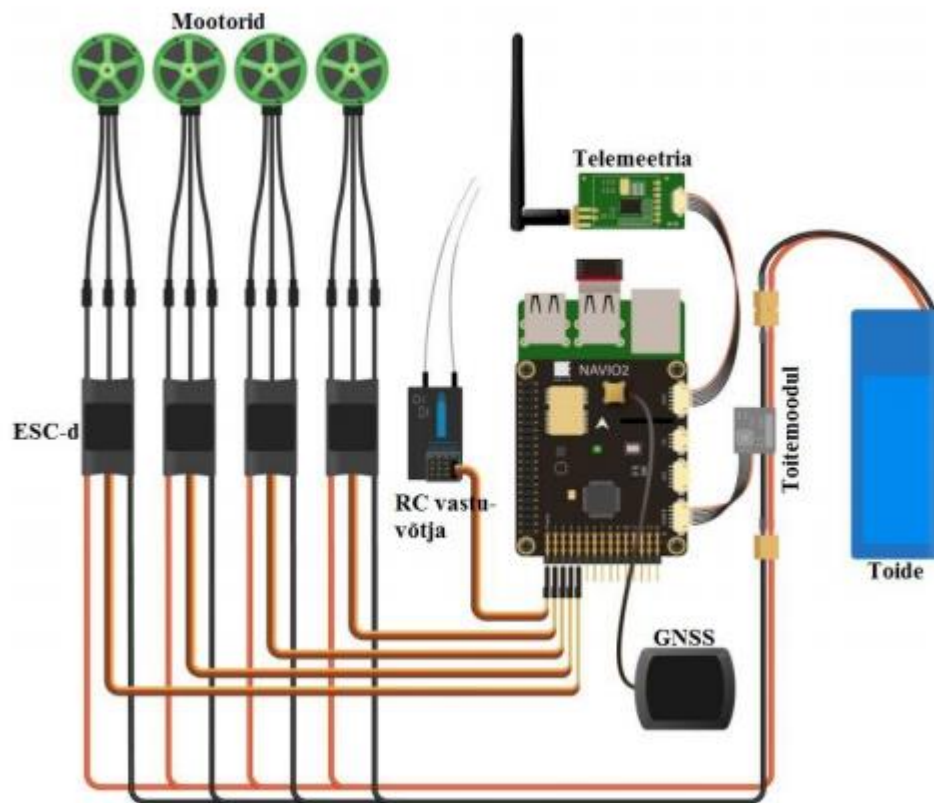


Figure 2.5 : Power distribution

Battery

In this work one lithium polymer (LiPo) Foxy 2200mAh / 11.1V 25 have been used. This is a 3S battery, which means there are three 3.7V (rated voltage) batteries in the battery sequences connected, i.e. total 11.1V. When the battery is fully charged, there is a total voltage 12.6V or 4.2V battery. For the controller, the voltage (5V) changes to the power module. The 2200mAh is a capacitance, which depends on how long the battery is held (how much does the battery charge can fit) mAh is a milli-ampere hour and the number shows how much current you can consume the battery getting cleared within one hour. In this case, then 2.2 hours per hour, but the number

is relative. 25C indicates how much the battery can discharge. In the case of 80 sharp increases (10s). For a given battery, then 25C * 2.2A or 55A on the battery. The voltage of the battery depends on the engine speed, because it is in direct relation with the KV. Made In case 11.1V * 1000KV is 11100RPM per Volt. Because the suggested KV size was actually smaller, this may mean that the motors consume too much current, but this is again probable only with full-time driving.

Monitor Battery Status

FrSky FLVSS Voltage Sensor has been used to monitor a single battery condition. The Ping Sensor is connected to a single battery balance port and the FrSky RC receiver Smart the port.

The sensor has an OLED display, which shows the voltage of all three batteries connected to the battery separately (with 3s battery). The Pulto shows the total voltage of LiPo and the tension of all three batteries separately. You can also show the battery with the least voltage or, for example, the highest and lowest voltage battery difference.

If the voltage level drops below a certain limit, the controller sends a continuous loud alarm. Boundary It can be determined by itself, with this controller model it is 2.7-3.8V per battery. The limit is set 3.1V, which is already a very critical limit, but for example 3.5V, the alarm sounds like a sound because the sudden increase in load temporarily pulls the voltage down to about 3.2V after In addition, the tension and power consumption are also displayed in the Mission Planner. Mission Planner displays the total voltage and informs the user if the battery voltage has fallen below a certain level. That's it also, the user determines the user according to his wishes. The software can display this information thanks to controller power module. The Mission Planner software must be enabled to display the information monitor the battery and set the module and controller type.

Navio2

Navio2 uses 5V input but battery gives a 11.1V output. Therefore Power module is used which provides constant 5V supply to controller board. Navio2 can be powered from three different ways(power module port, from raspberry pi, Voltage from BEC in ESC), this special feature is called triple-redundant power supply.

ESC

This ESC is equipped with the BEC (Battery Elimination Circuit). BEC is nothing but step down converter. BEC adapts power supply to the consumer suitable. In this work, the power supply is 11.1V and the controller is suitable for 5V. BEC supplies power to the other electronic devices (RC Receiver) present on drone without any requirement of additional battery.

The aircraft controller is powered by a separate port, but the BEC is required because if necessary, they will receive power supplies via ESCs. ESCs also provide back-up power to the controller.

Summary of the Chapter 2:

In this chapter Main principle for stable flight control is discussed initially. Then how the signal sent from the transmitter is processed for controlling drone movement are discussed. Finally power distribution and roles of each hardware component used are given.

CHAPTER 3: TELEMETRY AND PROGRAMMING

3.1: Ardupilot Manager (APM) & Mission Planner:

Using APM as software suite and Mission planner as ground control station communication with the quadcopter is achieved. Connection is established using TCP Connection and MAV-Link protocol.

APM (ArduPilot Manager)

ArduPilot is an open source, unmanned vehicle Autopilot Software Suite, capable of controlling autonomous vehicles. The ArduPilot software suite consists of navigation software (typically referred to as firmware when it is compiled to binary form for microcontroller hardware targets) running on the vehicle (either Copter, Plane, rover, or Sub), along with ground station controlling software including Mission Planner, APM Planner, QGroundControl, MavProxy, Tower and others.

ArduPilot source code is stored and managed on GitHub, and as of early 2017 has been forked by more than 5,000 GitHub users. The source tree includes approximately 700,000 lines of primarily C++ code, originating from 25,000 patches with 300+ contributors, and has been forked over 5,000 times

Emlid Raspbian has preinstalled ArduPilot. It includes all vehicles and is based on the most stable branch available.

Navio2 connections of Quadcopter

Extensions include Navio2-U UART, I2C, and ADC protocol interfaces with analogue sensors. (ADC), for example, an external compass (I2C) and telemetry (UART). Long band channels 1-14 is a PWM output that is connected to an ESC and can be connected, for example servo Because the servo does not power through the battery, the power strip provides power to the ESC. Additionally, the PPM / S.Bus Signal Input Peak Band is used to decode the RC input. Here this is the job of S.Bus. Power is supplied through Navio2. Navio2 is powered by a power module POWER port. The module provides the correct voltage and current to Navio2.

Mission Planner software

- Allows the aircraft to set up a wide range of hardware and parameters tuning
- Plan and save an autonomous flight
- Displays real-time flight information

MAV Link (Micro Air Vehicle Link)

MAVL is a protocol for two-way data transmission between the GCS and the aircraft. The MAVLink message is a byte stream that the encoder encodes and receives the decoder. The message corresponds to a specific command and is identified through the message ID. GCS takes receive flight data from the autopilot, which is displayed on the screen. The commands are of three types :

- Navigational commands control the movement of the aircraft (take-off, movement between designated destinations, elevation change, landing)
- The "Do" or "Dont" commands for additional functions, do not affect the aircraft position. The commands can be used, for example, to start the camera at the desired moment.
- The conditional command sets the condition for the "Do" command to be executed. For example, an aircraft must Be at a certain height or distance before completing the "Do" command. The MAVLink message consists of a header, a payload, and a check sum

The GCS sends an "heartbeat" message to the aircraft every 1 second to identify the existence of a connection between the GCS and the aircraft. Connection not activated Failsafe mode, in which the helicopter landed, continues to mission or turns the starting point. When encoding, the data packet's data structure, ie the type of information (info position, GPS, RC (Radio Control) channels, etc.) and an error correction is added bytes When decoding, the software first identifies the "System ID" and "Component ID". Id is available every system that uses MAVLink. GCS is usually 255 and aircraft 1. Then the "Message ID" is encoded. Next, the software separates the content from the message (payload) and transmits it according to the type of information. MAVLink checks for the compatibility of the length of the package and the checksum. If they do not flip the package is invalid and removed. For this reason, the telemetry data transmission is an IP connection 115200 bps instead of 57600 bps, the lower the speed the probability is less.

Calibration of Quadcopter

You must first select the frame type in the program. Then you can start the compass, RC, accelerometer and ESC calibration. The controller is a computer for calibration connected USB cable. The exact compass configuration is important because it is the main source of information for the aircraft. To calibrate the compass, each aircraft should be routed towards the ground. Moving aircraft around the axes, the sensor collects data points in positions and in the magnetic field strength, and can be correctly orientated in the magnetic field in the earth.

To calibrate the remote control, the control panel and switches must be moved in all possible directions in order to save their minimum and maximum positions. The calibration of the accelerometer is similar to the calibration of the compass: an aircraft must be placed on the plane, on the left and on the right side, on the front and back ends, and on the back. After calibration, the accelerometer should be in the position of the aircraft correctly identify the horizon.

To calibrate the ESCs, use the joystick, in ESC calibration mode record maximum and minimum positions of the lever to ensure the correct maximum and minimum position for determining the engine speed.

Failsafe

If the connection between the transmitter and receiver is interrupted, the failsafe mode is activated. Made the aircraft is assigned a No Pulse mode, that is, between the sender and the receiver When the connection is interrupted, the receiver terminates the signal transmission on all channels and The controller activates Land flight mode, which means the helicopter will fly back to where it is flying got up.

Instead of Land, you can, for example, put the landing mode, that is, a connection when the helicopter lands. You can also set up an order for resuming a mission if the helicopter was in Auto mode when the connection was interrupted. RC connection interrupted in the case, however, the most reasonable choice seems RTL. You can set Mission in modes through the planner.

Raspberry Pi Confection

You must first install Raspberry Pi on your operating system. It is used for this purpose micro-SD card. Modified “real time” emblems for Navio2 The raspbian operating system, because compared to the default Raspbian is with it Raspberry Pi response time with a smaller reference, because it allows almost the entire core of Linux give privileges to higher priority sections within the code. EmlidRaspbian is on pre-installed Ardupilot software.

After installing the operating system, Raspberry Pi must be connected to the Internet Modify the wpa_supplicant.conf file on it, specifying the one currently in use wireless local area network SSID and password. An Ethernet cable can also be used if necessary. For editing, there are two ways to: Use the graphic environment for

raspberryPi connection to the monitor via an HDMI cable or through the SSH network protocol. SSH the protocol creates a secure channel for transmitting data between the client and the server. Using a monitor with Raspberry Pi, you need to connect another USB keyboard and mouse, but unfortunately, neither was used, and therefore we used SSH.

3.2 IoT Service for SSH & Video streaming:

Block diagram

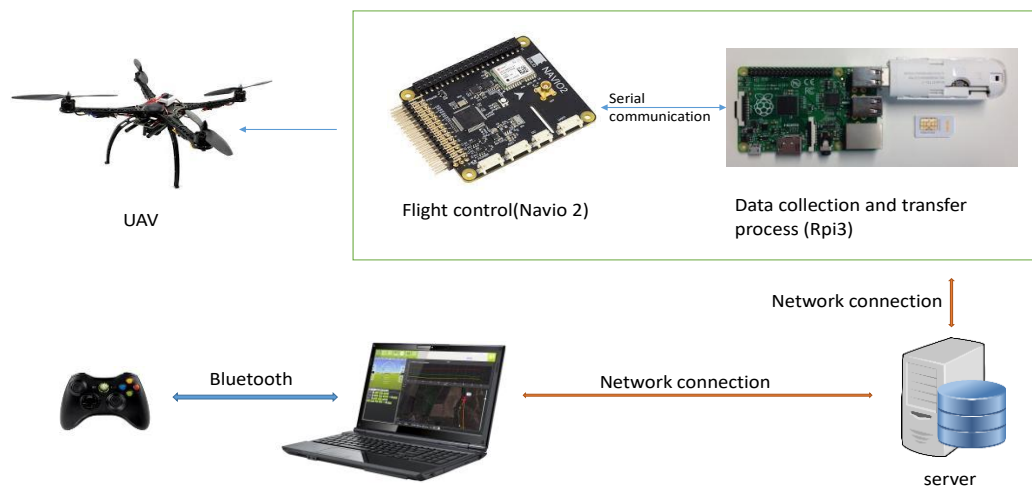


Figure 3. :Telemetry diagram

3.2.1 Weaved Service for SSH

We need to convert our Rpi into an IoT device to establish connection to it for running our program remotely using 4G. When we try to establish IoT platform independently we're stuck serving web pages and data on our local network. Getting information from our Pi on your phone, or while you're at work or school, it's nearly impossible.

There are a lot of ways to work our way out to the internet. They're often hard, ISP's block ports, you need to setup port forwarding, you might violate your terms of service with your ISP, and you might need to customize your router.

The easiest way we've seen to open up your Raspberry Pi as an Internet of Things device is to use the service Weaved. Weaved provides an IoT (Internet of Things) Kit for the Raspberry Pi. The kit provides really simple tools for connecting your Pi to the cloud, receiving notifications, and turning your Pi into an Internet of Things Kit.

They offer a few services that let you connect from a distance:
SSH - You can login to your Pi from anywhere over SSH.

We have used this to connect to Rpi using SSH and Bitvise as the client software on PC. The main purpose of using Weaved is to run 3 main tasks (getting sensor data, video-streaming & facial recognition). Once the connection is established with Rpi though SSH programs for getting sensor data, video-streaming and facial recognition are executed directly from the terminal window provided by Bitvise client on PC/Mobile with internet.

Face-recognition:

Face recognition identifies persons on face images or video frames. In a nutshell, a face recognition system extracts features from an input face image and compares them to the features of labeled faces in a database. Comparison is based on a feature similarity metric and the label of the most similar database entry is used to label the input image. If the similarity value is below a certain threshold the input image is labeled as *unknown*. Comparing two face images to determine if they show the same person is known as face verification.

We have used dlib's facial-recognition API for our application. It uses a deep convolutional neural network (CNN) to extract features from input images. In a nutshell CNN architecture and how the model can be trained is as follows:

- Detect, transform, and crop faces on input images. This ensures that faces are aligned before feeding them into the CNN. This preprocessing step is very important for the performance of the neural network.
- Use the CNN to extract 128-dimensional representations, or *embeddings*, of faces from the aligned input images. In embedding space, Euclidean distance directly corresponds to a measure of face similarity.

This API can do things like

- Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library.
- Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.
- This also provides a simple `face_recognition` command line tool that lets you do face recognition on a folder of images from the command line!

A python program is written to detect number of faces, name the known faces and display information on terminal window. Additionally using SMTP Server we have added code to python program to generate An E-Mail with the information about Face detected.

3.2.2 Dataplicity IoT Service for Video Streaming:

Video-streaming through internet was one of the tough goals of this project. MJPG-Streaming through Dataplicity IoT service is the technology we used to achieve this goal. MJPG is nothing but motion JPG which sends series of images as mentioned by the user which in turn appears as a video streaming. In the command line we give the following constraints : frame rate ,resolution.

Dataplicity provides secure login through internet through which live-video streaming could be achieved from any device with internet connectivity.

Summary of the Chapter 2:

In this chapter Main principle for stable flight control is discussed initially. Then how the signal sent from the transmitter is processed for controlling drone movement are discussed. Finally power distribution and roles of each hardware component used are given.

CHAPTER 5: RESULTS

- Stable flight control & telemetry over wi-fi range using TCP has been achieved.
- ESC and IMU sensors were calibrated properly which was reflected in stable flight control.
- Failsafe works fine when receiver loses connection with the transmitter the drone slowly landed on ground without crashing.
- Successfully achieved the IoT service connections – Weaved & dataplicity for Raspberry Pi.
- SSH connection to drone's Raspberry pi was established over 4G network. Through which custom python programmes and command lines can be sent to Rpi and returned values can be seen through Terminal.
- On board sensor values are obtained using a python program without any errors.
- Video-streaming with less latency was obtained with dataplicity service v4l2 driver.
- Python program for running facial recognition and sending email alerts worked without any errors.



Figure 5.1 : Final quadcopter

CHAPTER 5: CONCLUSIONS AND FUTURE SCOPE

- The integration of Linux based raspberry pi to the quadcopter opens up opportunity of adding modern MEMS based sensors to the quadcopter with ease.
- Due to the high processing power of raspberry pi high performance jobs like 4G telemetry, machine learning could be achieved in the future
- 3d scanning of the objects could also be achieved thus drone can be converted into intelligent devices
- By adding GPS to the drone particular location of the drone can be obtained remotely and additional flight modes like position hold and loiter can be achieved using mission planner.

REFERENCES

- [1]. Hoffmann, G.M.; Rajnarayan, D.G., Waslander, S.L., Dostal, D., Jang, J.S., and Tomlin, C.J. (November 2004). “The Stanford Test bed of Autonomous Rotorcraft for Multi Agent Control (STARMAC)”
- [2]. Leishman, J.G. (2000). Principles of Helicopter Aerodynamics. New York, NY: Cambridge University Press.
- [3]. Anderson, S.B. (1997). “Historical Overview of V/STOL Aircraft Technology .NASA Technical Memorandum 81280.
- [4]. Pounds, P.; Mahony, R., Corke, P. (December 2006). “Modelling and Control of a QuadRotor Robot”.
- [5]. Hoffman, G.; Huang, H., Waslander, S.L., Tomlin, C.J. (20–23 August 2007).”Quadcopter Helicopter Flight Dynamics and Control: Theory and Experiment”

WEBSITES

- [6] <http://www.as.northropgrumman.com/products/ghrq4a/index.html>
- [7] <http://www.ga-asi.com/products/aircraft/predator.php>
- [8] <http://www.sciencedaily.com/releases/2011/07/110701203725.html>
- [9] <http://www.uavforge.net/>
- [10] <http://diydrones.com/>
- [11] http://dlnmh9ip6v2uc.cloudfront.net/images/products/10061-01b_i_ma.jpg
- [12] <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rfmodules/pointmultipoint-rfmodules/xbee-pro-xsc#overview>

APENDIX 1: CODE GENERATION OF QUADCOPTER

Setting up navio and raspbian

Downloading configured Raspbian image

Navio requires a preconfigured Raspbian to run. We provide a unified SD card image for Raspberry Pi 2 and 3. The OS is headless, i.e. it comes without GUI as it is not required for drone applications.

Configuring Wi-Fi access

```
network={
ssid="yourssid"
psk="yourpasskey"
key_mgmt=WPA-PSK
}
```

Choosing a vehicle, version and board

```
sudoemlidtoolardupilot
```

Specifying launching options

Open the file:

```
pi@navio: ~ $ sudonano /etc/default/arducopter
```

Here you can specify IP of your ground station.

```
TELEM1="-A udp:127.0.0.1:14550"
#TELEM2="-C /dev/ttyAMA0"
```

```
# Options to pass to ArduPilot
ARDUPILOT_OPTS="$TELEM1 $TELEM2"
```

-A is a console switch (usually this is a Wi-Fi link)

-C is a telemetry switch

Usually this is either /dev/ttyAMA0 – UART connector on your Navio
or /dev/ttyUSB0 if you're using a serial to USB convertor

-B or -E is used to specify non default GPS

All lines marked '#' are comments and have no effect.

For example, you'll need to modify TELEM1 to point to your IP like this:

```
TELEM1="-A udp:192.168.1.2:14550"
```

Where 192.168.1.2 is the IP address of the device with the Ground Control Station – your laptop, smartphone etc.

- -A – serial 0 (always console; default baud rate 115200)
- -C – serial 1 (normally telemetry 1; default baud rate 57600)
3DR Radios are configured for 57600 by default, so the simplest way to connect over them is to run with –C option.
- -D – serial 2 (normally telemetry 2; default baud rate 57600)
- -B – serial 3 (normally 1st GPS; default baud rate 38400)
- -E – serial 4 (normally 2st GPS; default baud rate 38400)
- -F – serial 5

Starting arducopter

```
pi@navio: ~ $ sudo systemctl start arducopter
```

Stopping arducopter

```
pi@navio: ~ $ sudo systemctl stop arducopter
```

Autostarting on boot

To automatically start ArduPilot on boot you need to enable arducopter:

```
pi@navio: ~ $ sudo systemctl enable arducopter
```

To disable the autostart:

```
pi@navio: ~ $ sudo systemctl disable arducopter
```

You can check if ArduPilot is already enabled or not:

```
pi@navio: ~ $ systemctl is-enabled arducopter
```

weaved service for SSH to Rpi

Install the weaved connectd package

First, get the latest repositories:

```
sudo apt-get update
```

Next, download the remot3.it weaved connectd package:

```
sudo apt-get install weavedconnectd
```

```

pi@raspberrypi-3:~$ sudo apt-get install weavedconnectd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  weavedconnectd
0 upgraded, 1 newly installed, 0 to remove and 176 not upgraded.
2 not fully installed or removed.
Need to get 110 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 http://archive.raspberrypi.org/debian/ jessie/main weavedconnectd armhf 1.3-07v [110 kB]
Fetched 110 kB in 0s (141 kB/s)
Selecting previously unselected package weavedconnectd.
(Reading database ... 122735 files and directories currently installed.)
Preparing to unpack .../weavedconnectd_1.3-07v_armhf.deb ...
Unpacking weavedconnectd (1.3-07v) ...
Processing triggers for systemd (215-17+deb8u5) ...
Setting up nginx-full (1.6.2-5+deb8u4) ...
Setting up nginx (1.6.2-5+deb8u4) ...
Setting up weavedconnectd (1.3-07v) ...

=====
Available commands:

sudo weavedinstaller - configures your device's tcp services for remote access

=====
Processing triggers for systemd (215-17+deb8u5) ...
pi@raspberrypi-3:~$

```

Figure :Weaved installed

Run weaved installer to configure remot3.it service attachments

running

sudo weavedinstaller

to launch the installer.

```

***** Sign In Menu *****

1) Sign in to your existing remot3.it account
2) Request a code for a new remot3.it account
3) Enter a verification code received in e-mail
4) Exit

*****

Use your Weaved account with remot3.it!
*****

Please select from the above options (1-4):

```

Figure : After initialization weaved

select option 1, then log in..

Next, as there are no services yet installed, you will be asked to enter the Device Name. Valid characters include numbers, letters, space, underscore, and dash.

```

weavedconnectd version 1.3-07v is currently installed.
1.3-07v is the most recent version of pi_installer
.....

Enter a name for your device (e.g. my_Pi_001).

The Device Name identifies your device in the remot3.it portal.
Your services will be grouped under the Device Name.

Only letters, numbers, underscore, space and dash are allowed.

My-Device-01_

```

Figure :Initial view of installation

This is the initial view of the Installed remot3.it Services Menu. All that you have at this point is the Device Name.

```

===== Installed remot3.it Services =====
Protocol      Port      Service      remot3.it Service Name
-----
Device Name: My-Device-01
=====

***** Main Menu *****

1) Attach/reinstall remot3.it to a Service
2) Remove remot3.it attachment from a Service
3) Remove all remot3.it attachments, then exit
4) Exit

*****

Please select from the above options (1-4):

```

Figure : Main menu of weaved

```

***** Protocol Selection Menu *****

1) SSH on port 22
2) Web (HTTP) on port 80
3) VNC on port 5901
4) Custom (TCP)
5) Return to previous menu

*****

You can change the port value during install

*****

Please select from the above options (1-5):
_

```

Figure : Protocol selection menu

Select the type of service you are connecting to. Options 1 through 3 are predefined for the most common use cases. You can choose a different port during installation of options 1 through 3. Select option 4 to configure remot3.it for other use cases.

Installing a remot3.it attachment for SSH

Select 1 at the Protocol Selection Menu to configure an attachment for SSH.

```
Please select from the above options (1-5):
1
You have selected: 1.

The default port for SSH is 22.

Would you like to continue with the default port assignment? [y/n] y
We will attach a remot3.it connection to the following service:

Protocol: ssh
Port #: 22
.....

Enter a name for this remot3.it service (e.g. SSH-Pi).
This name will be shown in your remot3.it Service List.

Only letters, numbers, underscore, space and dash are allowed.
Pi-SSH_
```

Figure : configure attachment

Next, you will be returned to the **Installed remot3.it Services** menu. Now your attachment to the ssh service is shown.

We have successfully installed the remot3.it service connection for SSH. In the “Service” column you will see the name of the tcp listener program (usually called a “server”) which was found on the specified port.

Learn more about using remot3.it and SSH [here](#).

Dataplicity for pi camera web-service

The technology that we've used to achieve that is an MJPG-streamer application and a Raspberry Pi camera.

Enable camera

```
sudo raspi-config
```

Test the camera

```
raspistill -o /home/pi/image.jpg
```

Rpi camera as USB video device

```
sudo modprobe bcm2835-v4l2
```

Verify USB video device is available

```
ls /dev | grep vid
```

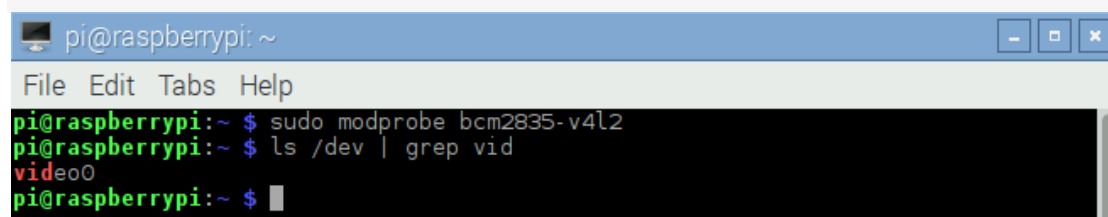


Figure : Verifying USB Video

Installing dependencies for MJPG-streamer

```
sudo apt-get install libjpeg8-dev imagemagick libv4l-dev
```

MJPG Server run command

```
sudo ./mjpg_streamer -I"./input_uvc.so -f 10 -r 640x320 -n -y" -O"./output_http.so -w ./www -p 80"
```

The **MJPG-streamer** itself contains couple of options. In the above command that we used to run the server we've set two options:

- “-I” is an option that takes path to an input plugin and the parameters to it.
- “-O” is an option that takes path to an output plugin and the parameters to it.

The parameters that we've used for the input plugin are:

- “-f” framerate– this sets how many frames are captured per second.
- “-r” resolution – this sets the size for the image we’re capturing
- “-n” This option suppresses the errors that show up when the command is ran. These errors show because our camera can’t be physically controlled. The errors have no impact on the streaming functionality but it’s nice to run the server without seeing them.
- “-y” Specifies YUV format for the output image.

The parameters that output option takes that we’re using are:

- “-w” path to folder where the website content is served from.
- “-p” port on which the website and stream is served on.

Wormhole URL with video stream in HTML

https://<YOUR_ID>.dataplicity.io/stream_simple.html

Python program

```
import spidev
import time
import argparse
import sys
import navio.mpu9250
import navio.util

import sys, time
import time
import sys, time

import navio.rcinput
import navio.util

import navio.ms5611
import navio.util

import navio.adc
import navio.util
parser = argparse.ArgumentParser()
parser.add_argument("-i", help = "Sensor selection: -i [sensor name].
Sensors names: mpu is MPU9250, lsm is LSM9DS1")
baro = navio.ms5611.MS5611()
baro.initialize()

if len(sys.argv) == 1:
    print "Enter parameter"
    parser.print_help()
    sys.exit(1)
```

```

elif len(sys.argv) == 2:
    sys.exit("Enter sensor name: mpu or lsm")

args = parser.parse_args()

if args.i == 'mpu':
    print "Selected: MPU9250"
    imu = navio.mpu9250.MPU9250()
elif args.i == 'lsm':
    print "Selected: LSM9DS1"
    imu = navio.lsm9ds1.LSM9DS1()
else:
    print "Wrong sensor name. Select: mpu or lsm"
    sys.exit(1)

if imu.testConnection():
    print "Connection established: True"
else:
    sys.exit("Connection established: False")

imu.initialize()

time.sleep(1)

adc = navio.adc.ADC()
results = [0] * adc.channel_count
rcin = navio.rcinput.RCInput()

while (True):
    s = ''
    for i in range (0, adc.channel_count):
        results[i] = adc.read(i)
        s += 'A{0}: {1:6.4f}V '.format(i, results[i] / 1000)
    print(s)
    # imu.read_all()
    # imu.read_gyro()
    # imu.read_acc()
    # imu.read_temp()
    # imu.read_mag()

    # print "Accelerometer: ", imu.accelerometer_data
    # print "Gyroscope:      ", imu.gyroscope_data
    # print "Temperature:    ", imu.temperature
    # print "Magnetometer:   ", imu.magnetometer_data

    # time.sleep(0.1)

    m9a, m9g, m9m = imu.getMotion9()

    print "Acc:", "{:+7.3f}".format(m9a[0]),
    "{:+7.3f}".format(m9a[1]), "{:+7.3f}".format(m9a[2]),
    print " Gyr:", "{:+8.3f}".format(m9g[0]),
    "{:+8.3f}".format(m9g[1]), "{:+8.3f}".format(m9g[2]),
    print " Mag:", "{:+7.3f}".format(m9m[0]),
    "{:+7.3f}".format(m9m[1]), "{:+7.3f}".format(m9m[2])
    baro.refreshPressure()

```



```

        time.sleep(0.01) # Waiting for pressure data ready 10ms
        baro.readPressure()

        baro.refreshTemperature()
        time.sleep(0.01) # Waiting for temperature data ready 10ms
        baro.readTemperature()

        baro.calculatePressureAndTemperature()

        print "Temperature(C): %.6f" % (baro.TEMP),
        "Pressure(millibar): %.6f" % (baro.PRES)

        period = rcin.read(2)
        print period
        time.sleep(1)

    time.sleep(5)

```

FACE RECOGNITION:

Install dlib and face_recognition

Install the picamera python library with array support (if you are using a camera):

```

sudo apt-get install python3-picamera
sudo pip3 install --upgrade picamera[array]

```

Temporarily enable a larger swap file size (so the dlib compile won't fail due to limited memory):

```

sudo nano /etc/dphys-swapfile

< change CONF_SWAPSIZE=100 to CONF_SWAPSIZE=1024 and save / exit nano >

sudo /etc/init.d/dphys-swapfile restart

```

Download and install dlib v19.6:

```

mkdir -p dlib
git clone -b 'v19.6' --single-branch https://github.com/davisking/dlib.git
cd ./dlib
sudo python3 setup.py install --compiler-flags "-mfpu=neon"

```

Install face_recognition:

```

sudo pip3 install face_recognition

```

Revert the swap file size change now that dlib is installed:

```

sudo nano /etc/dphys-swapfile

< change CONF_SWAPSIZE=1024 to CONF_SWAPSIZE=100 and save / exit nano >

sudo /etc/init.d/dphys-swapfile restart

```

Download the face recognition code examples:

```
git clone --single-branch https://github.com/ageitgey/face_recognition.git
cd ./face_recognition/examples
python3 facerec_on_raspberry_pi.py
```

python program:

```
import face_recognition
import picamera
import numpy as np
import smtplib

smtpUser = 'from_name@gmail.com'
smtpPass = 'password'

toAdd = 'to_name@gmail.com'
fromAdd = smtpUser

subject = 'Face detected'
header = 'to:' + toAdd + '\n' + 'From: ' + fromAdd + '\n' + 'subject: ' + subject
body = 'Drone has detected Barak Obama'

camera = picamera.PiCamera()
camera.resolution = (320, 240)
output = np.empty((240, 320, 3), dtype=np.uint8)

print("Loading known face image(s)")
obama_image = face_recognition.load_image_file("obama_small.jpg")
obama_face_encoding = face_recognition.face_encodings(obama_image)[0]

face_locations = []
face_encodings = []
while True:
    print("Capturing image.")
    camera.capture(output, format="rgb")

    face_locations = face_recognition.face_locations(output)
    print("Found {} faces in image.".format(len(face_locations)))
    face_encodings = face_recognition.face_encodings(output,
face_locations)

    for face_encoding in face_encodings
        match = face_recognition.compare_faces([obama_face_encoding],
face_encoding)
        name = "<Unknown Person>"

        if match[0]:
            name = "Barack Obama"
            s=smtplib.SMTP('smtp.gmail.com',587)
            s.echlo()
            s.starttls()
            s.echlo()
            s.login(smtpUser, smtpPass)
            s.sendmail(fromAdd, toAdd, header + '\n' + body)
            s.quit()

    print("I see someone named {}".format(name))
```

APPENDIX 2: HARDWARE COMPONENTS SPECIFICATIONS

BLDC MOTOR:

Basic parameters:

- RPM: 1000KV
- Power: 210W
- Maximum current: 21A
- ESCs: 30A

ELECTRONIC SPEED CONTROLLER(ESC'S):

The aircraft has HobbyKing 20A 3A UBEC speed controllers.

Essential parameters :

- Continuous current: 20A
- Freshwater: 25A
- BEC (Battery Eliminator Circuit): 5V / 3A

Engine parameters indicated that the ESC could be 30A, but it was not get it now Engines use a marked 21A current only on full-time driving, most it's definitely less than that. For security reasons, longer term should be avoided full flying.

Programmable parameters:

- Battery type: LiPo / NiXX
- Brake: On / Off – When the braking is on, the engine will stop its work abruptly, otherwise it will do so smoothly.
- Voltage protection: Low / Medium / High – How much percentage of voltage can be consumed? Before the protection mode is activated in the speed controllers. Low 85%, average 65%, high 50%.

- Protective mode: Decrease power / Stop operation – The predetermined voltage is critical At the limit, the power is reduced or the work is interrupted.
- Timing: Automatic / High / Low – Appropriate timing for starting engine operation.
- Start: Fast / Normal / Smooth – the nature of the engine start up.
- PWM frequency: 8kHz / 16kHz – communicates with the ESC controller at the frequency. Multirootors usually have 8kHz.
- Helicopter mode: Off / 5s / 15s – if it's a helicopter.

BATTERY:

2200mAh 11.1V 25C 3S battery was used.

Navio2 controller:

Navio 2 is equipped with the following sensors :

- MPU9250 9DOF IMU
- LSM9DS1 9DOF IMU MS5611 Barometer
- U-blox M8N Glonass / GPS / Beidou

RC TRANSMITTER AND RECEIVER:

FrSky 2.4GHz receiver transmitter with 6-channel PWM communication was used.