Pamukkale University

Engineering Faculty

Computer Engineering Department

<u>Bachelor's Thesis</u>

RangeFinder Quadcopter

14253805 **Rozan MUSTAFA**

13253813 **Ömer Mustafa OĞUZOĞLU**

Thesis Advisor Dr. Research Assistant Gökçen YILMAZ

June 2019

DENİZLİ

Bachelor's Thesis

Final Report

RangeFinder Quadcopter

Rozan MUSTAFA

Ömer Mustafa OĞUZOĞLU

PAMUKKALE UNİVERSİTY

The aim of this thesis is to design and build an anti-crash quadcopter from scratch intended for self-development purposes in the area of controlling systems. A secondary goal of this project is to use this platform for future innovative projects that could include stabilization, image processing and Artificial Intelligence.

Supervised by : Dr. R. A **Gökçen YILMAZ**

Status : Public

Date : 31.05.2019

GRADUATE THESIS APPROVAL FORM

This thesis named **“RangeFinder Quadcopter”** prepared by Rozan MUSTAFA and Ömer Mustafa OĞUZOĞLU, supervised by Dr. R. A. Gökçen YILMAZ was viewed and accepted as a bachelor thesis in terms of its scope and quality.

Dr. R. A. Gökçen YILMAZ

Supervisor

…………….	……………..
Jury Member	Jury Member

Approved by the decision of the Board of Directors at Pamukkale University Engineering Faculty Department of Computer Engineering dated …../…../……… and numbered ………….

Prof. Dr. Sezai TOKAT

Head of Department

We declare that the scientific ethics and academic rules have been carefully observed in the design, preparation, execution, research and analysis of this thesis. The findings, data and materials which are not the primary product of this study are cited in accordance with the scientific ethics and are attributed to the studies cited.

Rozan MUSTAFA	Ömer Mustafa OĞUZOĞLU
………………….	…………………………..

Abstract

The popularity of the quadcopters is increasing as the sensors and control systems are becoming more advanced and less expensive. There are many commercial quadcopters available on the market today, but they are often hard to configure and comprehend. The time required to grasp the existing systems, could be spent designing better solutions.

This project aims to use understandable system descriptions and sensor models as a basis to design configurable estimators and controllers, and to build a quadcopter well suited for educational purposes; as well as aiding to more advanced control in the future.

The system consists of several components for necessary sensor input, a radio transmitter and an Arduino microcontroller. All filtering of signals, estimation of system states, calculation of control inputs and communication handling is done on the microcontroller.

A Quadcopter has been built that can be operated by radio frequency controller and send live audio-visual feedback. The developed Quadcopter control system has been simulated in iNavflight.

The simulation shows a very stable operation and control of the developed Quadcopter. Microcontroller based drone control system has also been developed where a RF transmitter and receiver operating in the frequency of 2.4 GHz are used for remote operation for the Quadcopter.

This paper describes modelling, estimation, and control of the horizontal translational motion of an open-source and cost-effective quadcopter. I determine the dynamics of its roll and pitch attitude controller, system latencies, and the units associated with the values exchanged with the vehicle over its serial port.

The final quadcopter concept is well suited for further experimental work.

Keywords: Drone, Quadcopter, Flight Controller, Naze32, iNavflight, Obstacle Avoidance Transmitter, Receiver, LoRa, Sonar

Acknowledgements

We would like to thank those who helped during our thesis work. Without their support, we could have never accomplished this work.

We take this special occasion to thank our parents. We dedicate this work to them. It would have been simply impossible to start, continue and complete without the support of our families who, unconditionally provided the resources to us. We are eternally indebted to our supervisor Dr. R. A Gökçen YILMAZ for all the help, invaluable guidance and generous support throughout our thesis work. We have been very fortunate to be associated with such a kind and good person and it would take more than a few words to express our sincere gratitude.

We also like to thank Prof. De. Sezai TOKAT and Lect. Dr. Gökhan Uçkan for their enlightening suggestions and advices. Their professionalism, guidance, dedication and inspirations will always serve us as an example in our professional life.

Table of Contents

Terms and Abbreviations

UAV: Unmanned Aerial Vehicle

MTOW: Maximum TakeOff Weight

GPS: Global Positioning System

DIY: Do It Yourself

LoRa: Long-Range Communication Protocol

PWM: Pulse Width Modulation

ESC: Electronic Speed Control

MEMS: Micro Electrical Mechanical System

PDB: Power Distribution Board

Rx: Receiver

RPM: Rounds Per Minute

FC: Flight Controller

Index of Tables

Index of Figures

This page left intentionally blank

Chapter 1

Introduction

This chapter familiarizes the reader with the emerging field of unmanned vehicles through the reviewing of their history and their implications at the current legislative and socioeconomic levels. Furthermore, the personal motivation which was triggered by the previously mentioned factors is presented. Moreover, in an attempt to ease the bachelor thesis reading, the structure along with the objectives to be achieved are introduced.

1.1.History

Although the popular belief in which the unmanned aerial vehicles are a relatively recent invention dating back to at most thirty years, these vehicles go back to ancient times. The oldest registered data date from 1849 when the Austrian army used balloons armed with explosives against the city of Venezia. Even though, nowadays balloons are left out of the current UAV category, they were close to the concept of an unmanned platform carrying a payload. As it can be deduced from the previous historical fact, technology and more specifically drones emerged to address a need of improvement, to meet the current needs and sometimes to surprise and conquer the contrary. Moreover, these vehicles quickly developed during war periods, in which instilling the contrary was a must.

During the First World War, the first pilotless aircraft were built and the radio control techniques were mastered to the point at which they were able to carry and guide missiles to the desired aims. Even though remotely piloted aircraft were also included as part of military weapons during the Second World War, it was not until the 90's when they reached their peak.

During this period, unmanned aerial vehicles, which had been perfected during the previous decade, were used as tools to watch and film the targets and to ultimately neutralize them. Consequently, the inclusion of these remotely piloted vehicles in war times has become vital, Iraq war, Syria war... However, despite their clear military purpose in the olden times, since 2005 the world is starting to experience a thrilling growth in the use of Unmanned Aircraft Systems as the technology with the potential to make possible new civilian applications. As a consequence, their application has been widened and they are playing an important role on missions with very different scopes as the ones shown in Table 1: [1]

Table 1: Unmanned Aerial Vehicles Current Applications

Scope	Mission
Security-Related	Homeland security
Safety-Related	Civil Defense Fire Fighters
European Union	Border Guards Police
Commercial Air Transport	Scheduled Air Service
General Aviation	Pleasure Corporate Operations Flight Training
Aerial Work	Advertising Survey and Mapping Photography Search and Rescue

1.2.Socioeconomic Framework

Due to the unstoppable inclusion of unmanned vehicles in quotidian activities at both governmental and non-governmental level, this industry has earned a positive image over the years and so far, it has the sympathy and attention of the public.

Regarding its market evolution, experts coincide that the current trend will not vary in the years to come, meaning that it will be still leaded by USA and followed by Europe and Asia. However, this situation may not be as durable as expected, countries where this market is starting to take off nowadays are believed to take over thanks to their untapped potential. As a consequence of this believed growth, a lot of new jobs will be created, and the success of the market will depend on regulations to maintain acceptable market conditions. According to the new research published by QRResearch, The Unmanned Aerial Vehicle (UAV) market was valued at 11.7 Billion US$ in 2018 and is projected to reach 31.7 Billion US$ by 2025, at a CAGR of 13.3% during the forecast period 2019 to 2025.[2]

1.3.Legal Framework

Due to this dramatic increase in their usage many organizations at both international and national level have been obliged to step forward to understand, define and ultimately reflect them in their legislation.

Regarding the Spanish legislation [3] , in order of defining the requirements that each vehicle has to comply with to be allowed to fly, these vehicles are divided in two main categories depending on their weights as follows:

1. MTOW below 25 kilograms: they are allowed to operate on the outskirts of cities and inhabited places at a maximum height of 120 meters and within the pilot's visual field.
2. MTOW over 25 kilograms: these kinds of vehicles are used in fire extinguishing, rescue and search operations.

Despite the legislation introduced by every State within their territories it is indispensable that all the countries come together to elaborate a global and variant legislation to accommodate the UAS requirements in the years to come.

1.4.Motivation

Quadcopter control is a fundamentally difficult and interesting problem. With six degrees of freedom and only four independent inputs, the quadcopter is underactuated and the resulting dynamics are highly nonlinear. Unlike ground vehicles, aerial vehicles have very little friction to prevent their motion, therefore they must provide their own damping in order to stop moving and remain stable. These factors make the stabilization of the quadcopter a fascinating control problem. The span and complexity make this project an extensive learning platform.

Our motivation lies in the design of our own solutions to the many challenges faced during the development of a quadcopter platform. Time required to grasp the existing commercial quadcopter systems, could be better spent designing our own solution. Another key motivational factor is the learning outcome gained during the execution of this project. Throughout this paper we will therefore evaluate various designs, implement and compare the results in order to conclude which of the solutions that has the best performance. Furthermore, this project can be used by future student who wish to work with quadcopters. Our greatest motivation for this project is to learn the many various aspects regarding a project of this size, and to improve our advanced problem-solving capabilities.

1.5.Goals

In this project the main objective is to predict the behavior of a quadcopter fully built and configured from scratch. However, this main objective implies some other ones to be accomplished, stated hereinafter:

1. Selection of the different parts that will form the vehicle, considering several alternatives and reasoning the selection of each element that finally composes the setup.
2. Start up and refinement of every component, such as calibrating the electronic speed controllers and the processor.
3. Stabilization of the vehicle through the tuning of its controller.
4. Performing an anti-crash approach using ultrasonic sensors.
5. Performing Return-Home model using GPS sensors.
6. Implementation of the vehicle’s dynamics and mechanics in iNavflight to predict its theoretical behavior.
7. Analysis and comprehension of the practical and theoretical results, with detailed reasoning and explanation of the observed consequences experienced by its exposure to different inputs.
8. Based on obtained results, suggest new ways to research into this field more deeply

1.6.Methodology

In order to achieve the previously mentioned objectives, the methodology that will be followed consists in first doing an extensive market analysis that will drive the vehicle’s selection. For that instance, an expert within the field will be contacted for advice. Since there are not many written references on how to mount quadcopters, mainly DIY blogs will be used in the documentation process before starting the assembly. In order to stabilize the vehicle, it will be flown with different controller values until an almost no-perturbations flight is achieved.
For that purpose, the SP Racing F3 Flight Controller will be used along with iNavFlight.

Chapter 2

Structure

In this chapter, under the state of art the different categories in which these vehicles are grouped is presented and the different existing vehicles are introduced at both hardware and software levels. This section leads to an extensive reasoning of the elements that are needed to build our own vehicle and it drives the components election.

2.1.State of the art

The selection of a vehicle is based on the objective and mission it has to fulfill and this choice will drive the drone dimensions and the on-board integrated technology. For instance, the number and size of the batteries will constrain the level of autonomy of the vehicle and the frequency of the controller will drive the maximum range of the drone.
In line with this project and according to the main purpose of the vehicle, inspection, that only requires it to carry a sonar, and later a GPS, the category that better fits it is Quadcopter.

- *Quadcopter*: these vehicles possess 4 rotors for lifting and propelling and their movement is controlled by varying the relative thrust of each rotor. In order to avoid them from spinning around themselves two opposite lying rotors spin in one direction, while the other two opposite lying rotors spin in the opposite direction.
- *Hexacopter:* the most notable difference at the construction level between these vehicles and the previous category is the number of propellers that they include. Moreover, at the performance level their stability is better and even though they lost one engine they could still land.
- *Octocopter*: they are propelled by eight rotors and they operate in a similar way as they previous cases. As a matter of fact, flight could continue after the loss of either one engine or two on opposite sides. They are mainly used for transportation of heavy objects due to their great stability.

Figure 1: Types of Multirotors

2.2.Hardware

All of the hardware components need to be compatible with each other. The battery needs to be able to output enough current for the electronic speed controllers and motors. A circuit board and power distribution board are needed to provide a supply voltage for all of the components. The microprocessor needs to be able to communicate with the sensors and LoRa module and output a PWM signal for the ESCs. The flight controller's update frequency should be high enough to control the quadcopter in a smooth manner.

Each part of will studied separately.

2.3.Software

In pursuance of obtaining a total controlling of the aerial vehicle capable of accomplishing the given mission, a complete symbiosis between the hardware and the software is needed. This important union leads to the so-called flight controllers which can be said to be the “neuralgic center” of the vehicle and the ones in charge of executing different orders. Their functioning combines the usage of both an algorithm or code integrated into a board to interpret and send orders, and a physical support to transfer these commands to the rest of the vehicle through their output PINS.

The flight controller will be implemented on Arduino systems. It should be able to communicate with the joystick using LoRa technology, read sensor measurements to estimate the current orientation and control the speed of the motors.
The flight controller implements a self-leveling flight mode. The user should be able to configure PID controller terms and other parameters with the joystick. The flight controller needs to communicate with the sensors and LoRa module using a Serial Peripheral Interface bus. The LoRa module needs to support the legal hobbyist frequencies used in Turkey.

2.4.Selection Process

2.4.1. Specifications

Before entering into a detailed discussion about which components are selected and why it is believed that they are the most suitable for this project, it is highly desired to recall the mission that needs to be undertaken.

"<u>Performing an anti-crash approach using ultrasonic sensors</u>"

In order to fulfill this objective a set of requirements based on a thesis presented by Román Esteban Hofer[4] is proposed.

Weight: it must not exceed 2kg.

Propulsion system: the maximum thrust per motor has to be of 1 kg.

Flight controller: it has to include an inertial measurement unit (IMU), a gyro stabilization, accelerometers, GPS, barometer and magnetometer

Transmitter and receiver: it must have at least four channels

2.4.2. Type of vehicles

Knowing that the vehicle has to be compatible with any of the mentioned boards and before looking into any hardware components (Sub-Section 2.2) such as the vehicle frame or the controller, a type of vehicle has to be chosen. For this purpose, the most common vehicle configurations are analyzed and compared in Table 2.

Table 2: Comparison of the different vehicle configurations

Characteristics	Quadcopter	Hexacopter	Octocopter
Overall price	Low	Medium	High
Replacement	Low	Medium	High
Maneuverability	High	Medium	Low
Landing after loss of one engine	No	Yes	Yes
Payload that can be carried	Low	Medium	Large
Size	Small	Medium	Large

Considering that the device will be flown in a controlled and hazard-free place it is assumed that no engine loss will occur, and high maneuverability will be preferred. Besides this, a quadcopter is determined to be the best option among other configurations for its low overall price.

2.4.3. Flight Controller

To help decide which flight controller is better to use, a table gathering all the notable specifications has been built:

Table 3: Comparison of the flight controllers

	SP Racing F3	Matek F405-CTR	Naze32	Airbot OMNINXT F7
Processor	MPU6050+F3	MPU6000+F405	MPU+F103	MPU6000+F7
iNavflight Firmware	Yes	Yes	Yes	Yes
Radio Protocols	Frsky Rx SBUS + PPM	Frsky Rx SBUS + PPM	Frsky Rx SBUS	Frsky Rx SBUS + PPM
Weight	6g	10g	5.3g	12g
OSD	Yes	Yes	Yes	Yes
Built-in PBD	No	No	No	Yes
Price	12 -20 $	35 – 45 $	7 – 18 $	91 – 96

Given this context and due to the nature of this project, which implies great tracking and intervention in the operative system, it is concluded that the most suitable option is SP Racing F3.

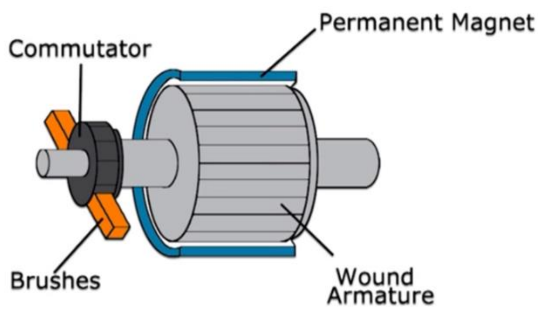
2.4.3. Motors

The motors election is very important as they need to be powerful enough to lift the whole drone. There are two main categories: brushed motors and brushless motors:

- *Brushed DC Motors*

Brush dc motors are one of the simplest types of motors, a typical brush dc motor consists of brushes, commutator, permanent magnet, wound armature.

The brushes charge the commutator inversely in polarity to the permanent magnet, in turn causing the armature to rotate. The rotation's direction, clockwise and/or counterclockwise, can be reversed easily by reversing the polarity of the brushes, reversing the leads on the battery.



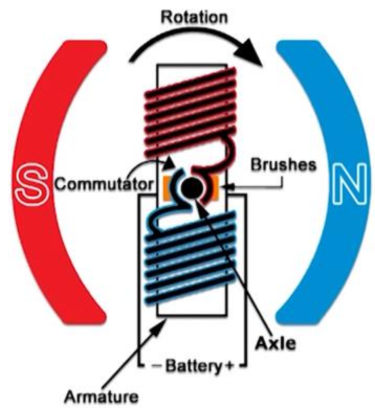


Figure 2: Brushed Motors

- *Brushless DC Motors:*

In terms of differences, the name is revealed obviously. BLDC motors lack of brushes, connections and commutator. In place of these, the motor employs control circuitry. To detect where the rotor is at certain times, BLDC motors employ, along with controllers, rotary encoders or a Hall sensor.

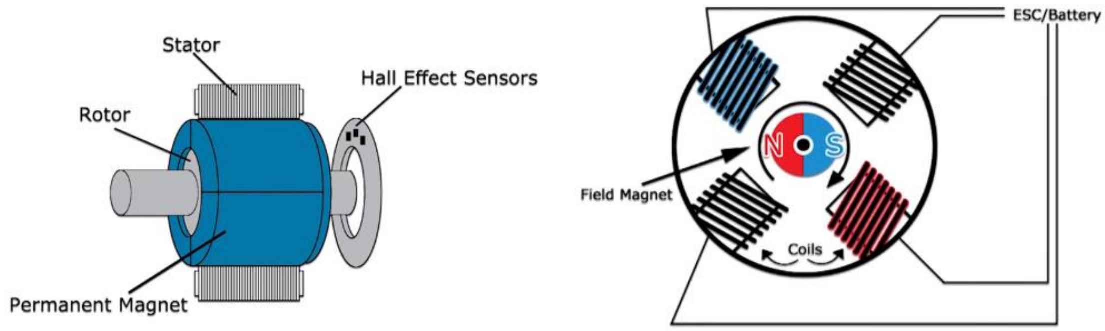


Figure 3: Brushless Motors

The brushless motors beat brush motors in the speed/torque tradeoff with their ability to maintain or increase torque at various speeds. Importantly, there's no power loss across brushes, making the components significantly more efficient. Better comparison is presented in table 4:

Table 4: Comparison of DC Motors

Feature	Brushless DC Motors	Brushed DC Motors
Communication	Electronic communication based on Hall Position Sensors	Brushed communication
Maintenance	Less or no maintenance	Periodic maintenance
Life	Longer	Shorter
Speed/Torque	Enable operation on all speeds with rated load	At higher speeds, brush friction increases and reduces torque
Efficiency	High	Moderate
Speed Range	Higher – no mechanical limitation due to contact	Lower – mechanical limitations due to brushes
Noise	Low	High

From Table 4, it can be summed up that the number of advantages of the brushless motors is greater than the ones of the brushed motors, hence the usage of brushless motors even though they can be a bit more expensive is convenient.

2.4.5. Electronic Speed Controller

These devices are electronic circuits that essentially generate a three-phase electric source of energy. Their purpose is to vary the brushless motor's speed in a radio-controlled model. A regular ESC used for aeromodelling has five wires and three connectors used for different tasks:

- *Powering the ESCs*: two cables are connected to the battery and give enough energy to the ESCs to power them up.
- *Establishing the motors spin direction*: it consists in three connectors hooked to the three motor's wires by a three-phase connection. The rotation direction will be determined by the order in which the wires are connected. In order to maintain a net aerodynamic torque and an angular acceleration equal to zero, two of the motors have to spin clockwise while the other do it counterclockwise.
- *Connecting the ESCs and the motors to the controller:* three cables (red, white and black) are connected to the microcontroller's PINs. In the simplest configuration, these cables will be used to both power and transmit the orders from the board to the motors. But if a power modulation was used, the board would receive the energy directly from the battery, only the white cable (the controller one) would be necessary.

In this case, since four motors will spin at different velocities four ESC's will be needed. It is recommended to use around 20 AMp- ESC's for quadrotors of around 1kg. Since this vehicle is believed to develop more complex missions in the future 30mAh ESCs are used instead.

2.4.5. Blades

A motor by itself is not able to specify the amount of thrust that it will generate. For that reason propellers need to be included. In order to select a suitable size for them, it must be taken into account that the torque constant and the voltage constant are related through a product that is kept constant for every motor:

$$Kv \,.\, Kt = 1352 \qquad \text{constant for every motor} \quad \ldots\ldots 1$$

As it can be seen from the formula Kv and Kt are inversely related, if one decreases the other one increases meaning that a small propeller would allow for greater speeds. Following the current's market trend 5 inches diameter blades will be used.

Note: Bigger propellers produces lower speed but greater stability, However, it will need bigger frame and low voltage expensive brushless motors.

Once the size is figured out it is necessary to determine the material. Carbon Fiber, wood and plastic seem to be the most popular in the market7 and are analyzed to determine the one that better fits our design.

Table 5: Comparison of different blade materials

Features	**Carbon Fiber**	**Wood**	**Plastic**
Vibration	Low	Low	Low
Weight	Light	Heavy	Medium
Price	High	High	Low
Thrust	Medium	Low	High

From this table it can be seen that plastic propellers are the ones that better fit this project needs as its flexibility would allow them to absorb a shock and take up more easily any sudden changes in direction. On the contrary, if a carbon fiber one impacted, the motor bearing would take most of the impact. Regarding the wood propellers, they do not work well for acrobatics or stunt flying because of the increase in rotation momentum due to their heavier weight. As mentioned before, 5 inch diameter blades are selected and from the materials comparison table it is determined that the best the material is plastic.

2.4.7. Receiver and Transmitter

Nowadays the most used system to establish a communication with the vehicle is the radio control. Its great advantage is its continuous updates thanks to the expansion of the aeromodelism world. However, its usage encounters a main limitation due to the fact that more than one user could be using the same frequency. Despite the possible appearance of interferences, it is decided to choose a system based on radio control because the vehicle will be flying over a territory. In order to choose a model and consequently a manufacturer, the number of channels needs to be taken into account. Each channel allows one individual action to happen and four is the minimum number for quadcopters, i.e. one for throttle, one for turning right and left, one for pitching forward and backward and one for rolling left and right. Since four channels is the minimum for quadcopters and in the future extra sensors may be added to the vehicle, it is decided to choose a 16-channel transmitter. Receivers and transmitters are not required to be from the same manufacturer as they come with a binding mechanism that allows the communication with each other. Despite this, it is preferred that both come from the same fabricator to avoid future problems.

Figure 4: Transmitter and Receiver

This page left intentionally blank

Chapter 3

Principles of Quadcopter work

In this chapter we are going do discuss how does a quadcopter hover or fly in any direction, lift or descend at a moments touch on the remote controller stick. Drones can also fly autonomously through programmed software and fly in any direction going from point to point. So let's look at the quadcopter technology, which makes this possible.

Before explaining how a quadcopter or any multi-rotator works there are some important concepts that we need to define – pitch, yaw and roll. These terms refer to the three dimensions that an aircraft in flight is free to move it.

- **Pitch** – Pitch refers to the nose of the aircraft going up or down. You could think of it as climbing or diving.
- **Yaw** – Yaw on the other hand refers to the nose of the aircraft turning left or right. You could simply think of this as turning.
- **Roll** – To understand roll think of an axis running from the front to the back of the aircraft. When an aircraft rolls it is turning on this access. You can also think of roll as a tilt.
- **Altitude** – Altitude is a term that we are probably familiar with – it simply describes going up or down. A quadcopter can either hover or adjust its altitude by applying equal thrust to all four rotors.

6.1.Physics behind Quadcopters

For a quadcopter to rise into the air, a force must be created, which equals or exceeds the force of gravity. This is the basic idea behind aircraft lift, which comes down to controlling the upward and downward force.
Now, quadcopters use motor design and propeller direction for propulsion to basically control the force of gravity against the quadcopter.
The spinning of the quadcopter propeller blades push air down. All forces come in pairs (Newtons Third Law), which means for every action force there is an equal (in size) and opposite (in direction) reaction force. Therefore, as the rotor pushes down on the air, the air pushes up on the rotor. The faster the rotors spin, the greater the lift and vice-versa.
The Rotors act as wings. They generate thrust by rotating at Fast speeds, which pulls the air downwards and keeps the quad in the air.

- The Thrust cancels out the acting weight and the quad hovers.
- A directional Thrust causes the quad to move in that direction.
- Or a decrease in Thrust overall causes the Drone to lose height.

The setups for Flying is simple:

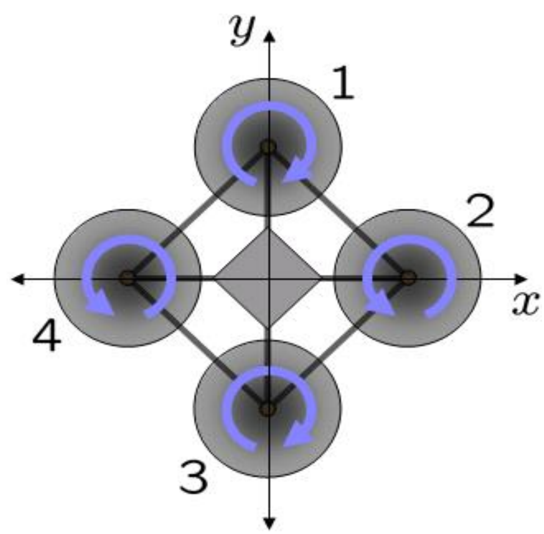
1. adjacent motors spin in the opposite direction.
2. Two opposite motors spin in the same direction.

Why?

3. Physics says to be in stability the net forces acting on a body should be zero.
4. So, if all the rotors were to spin in the same direction, it would result in a net Torque causing the complete Quad to rotate around itself.

6.2.Motors

Quadcopters make use of 4 Motors. Two of these motors spin clockwise while the other two spin counterclockwise. Motors on the same axis spin in the same direction, as illustrated here.



6.2.1. Directions

To adjust its **Yaw**, or make it turn left or right, the quadcopter applies more thrust to one set of motors. For example, a quadcopter may apply more thrust to the two motors that spin clockwise to make a turn.

Pitch and roll on the other hand are adjusted by apply more thrust on one rotor and less to the other opposing rotor. For example, the quadcopter can adjust its pitch by applying more thrust to the clockwise spinning motor in the front and less thrust to the clockwise spinning motor directly opposite in the back.

One thing that makes quadcopters easy to fly is that you don't have to adjust the motor speeds manually as this is where the controller and electronics come in. Ww'll discuss the electronics in a moment but first let's examine the motors.

The figure below explains how all directions can be obtained

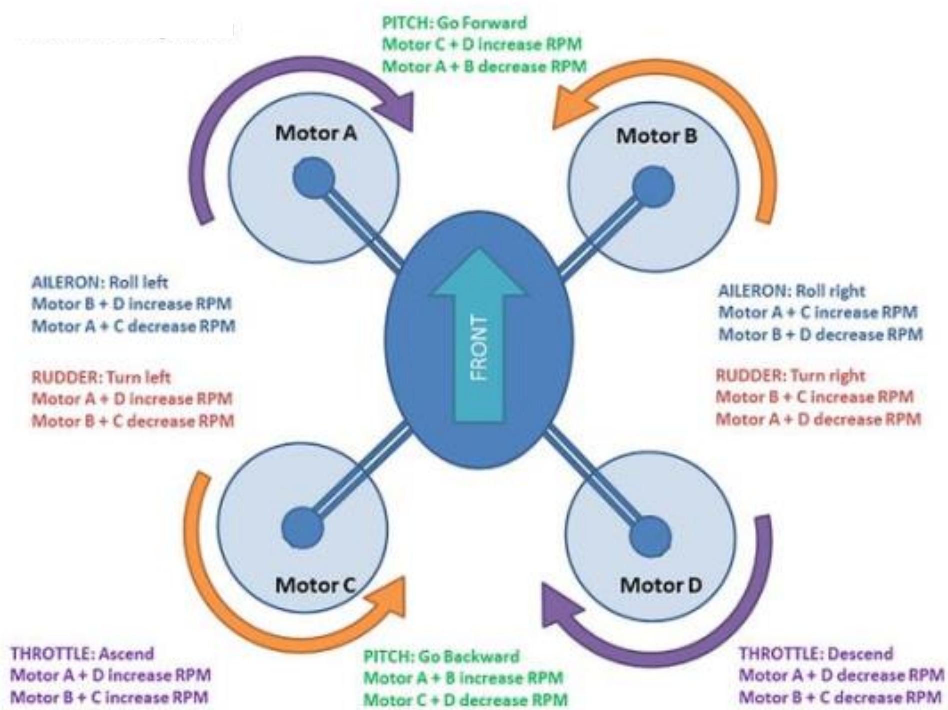


Figure 6: Quadcopter Directions Principles

6.2.2. Torque

A motor with lower Kv rating can produce higher torque and is able to spin a bigger propeller than a motor with higher Kv rating. Racing size quadcopters use small propellers, motors with higher Kv rating and therefore they maneuver faster but can lift less mass than heavy drones. Kv is equivalent to rpm/V, that is rounds per minute per voltage when there is no load attached to the motor. The more input voltage is supplied to the motor the faster it spins.

SP Racing F3 motors are used in this project. Their Kv rating is 2300. Drawing full voltage output from a 11.1 V battery results into 25 530 rpm with no load attached. In fact, many factors affect the test result, such as battery discharge rating, motor windings, propeller loading and motor efficiency etc.

$$\text{RPM} = \text{Motor KV} \cdot \text{Battery Voltage} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2$$

Let's now talk about Center of Gravity, have you ever tried to balance an object on the tip of your finger? That's a center of gravity!

Fun Fact: Your center of gravity is your belly button! The quads is, predictably, near its center.

Torque (τ) is a twisting force, we use it when we do things like turn a door-knob handle, use a wrench, or pedal a bicycle. Now, the torque produced by a force equals the magnitude of the *force* (F) times the *radius* (r), or distance from the center of gravity.

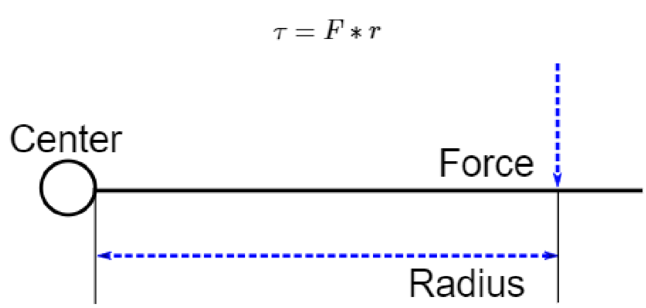


Figure 7: Diagram of Torque

So, what does torque have to do with the yaw of the quadcopter? Well, it turns out, a heck of a lot! Remember how the motors on our quadcopter spin?

Now, consider the torque generated by the motors.

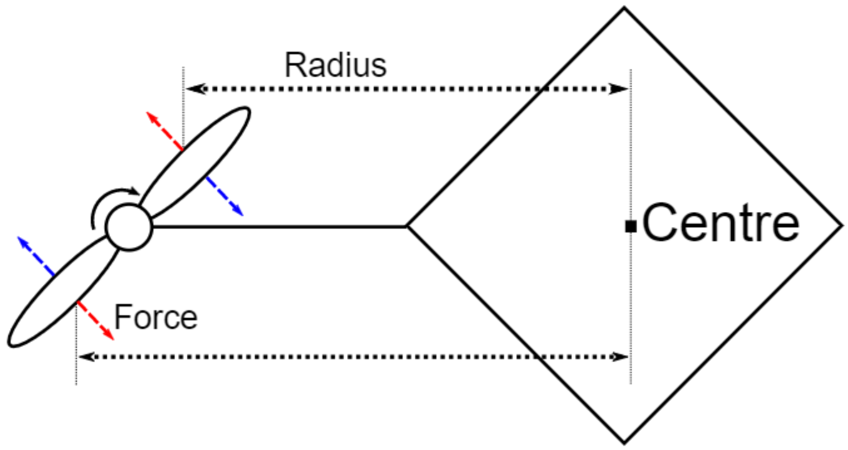


Figure 8: Torque diagram for a motor

In the diagram, *blue* arrows represent the force of the motor pushing on the air, while *red* arrows represent the force of the air pushing back on the motors.

"Every action force has an equal and opposite reaction force." Sir Isaac Newton

Note that the force is drawn at the center of the propeller. In this example we're using the average distance and force to simplify calculations. Feel encouraged to follow along and use calculus concepts (representing the force as a continuous amount), the math still works out!

In our example we're only showing one motor, but recalling that two motors almost shut off while yawing, and the other motor is generating the same directional forces, it is safe to do this.

Calculating the net maximum torque in this diagram can be done using the following equation:

$$T = F(r1) - F(r2) = F(r1 - r2) \quad \ldots\ldots\ldots\ldots 3$$

Where r1 is the bottom (larger) radius, and r2 is the top (smaller) radius. The key takeaway here is for each propeller **despite the forces being equal, the radii are not** and because of that, τ is a positive value.

What we end up with is a small amount of torque being created by each of the two active motors, these two amounts are in the same direction, and can be added together!

The combination of these two torques is enough to cause the quadcopter to yaw (twist) in the air[5].

6.2.3. Thrust

In a 3-phase brushless DC motor the electronic speed controller controls the motor by rapidly switching the current input for each of the coils. This creates a rotating magnetic field around the coil and spins the magnet. The magnet is attached to a shaft and causes it to spin. A propeller is attached to the same shaft and the thrust generated by the spinning propeller lifts the quadcopter in the air.

It can be shown that a propeller's thrust can be approximated by equation the following equation where *Ti* is a force vector for motor index *i*, where *k* is a scaling constant, and where ω is the angular velocity of the propeller [6]

$$Ti = k \,.\, \omega^2 \quad \ldots\ldots\ldots\ldots 4$$

In the table below, the thrust is calculated for the motor and blades used in this project, to not forget

- Motor: Emax RS2205 2300KV
- Blades : 5"inch

Table 6: Thrust calculations of the motor used in this project

Current (A)	**Thrust (G)**	**Power (W)**	**Efficiency (G/W)**	**Speed (RPM)**
1	62	12.00	5.17	64000
3	162	36.00	4.50	10080
5	236	60.00	3.93	12070
7	311	84.00	3.70	13730
9.1	374	109.20	3.42	15100
11	439	132.00	3.33	16320
13	490	156.00	3.14	17350
15.3	548	186.60	2.98	18350
17.3	611	207.60	2.94	19210
20.7	712	248.40	2.87	20080

Note: The thrust rating of a motor is specified in conjunction with a propeller size. This is an important specification that requires knowing the total weight of the quadcopter, keep in mind that this weight includes the weight of the motors themselves.

The basic rule of thumb is that the motor and propeller combination should be able to generate twice the weight of the craft in thrust. So with a quadcopter having four Motors this means that each individual motor should be able to provide thrust equal to half the weight of the entire quadcopter. In other words a 1 kilogram quadcopter requires four motors, each capable of at least half a kilogram of thrust.

The overall weight of the quadcopter built in this project is about 650g, however, each motor -as we can notice from Table 6- can generate not just the half, but the whole weight, so we are safe so far.

6.2.4. Payload Capacity

Payload Capacity = (A * B * D) – C ……………………………………………………………………….5

A = Motor Max Thrust

B = Number of motors

C = The weight of the craft itself

D = Hover throttle %

Keep in mind that for an agile aircraft we generally want it to hover at 50% throttle or lower.

Our quadcopter payload capacity on 50% throttle = (712 * 4 * 0.5) – 650 = 774 g

6.3.Orientation Sensors

In order to meet the control goals of the quadcopter, estimation of the quadcopter's orientation is required. With modern advancements in electronics, determining orientation can be done inexpensively, efficiently, and quickly with micro electrical mechanical system (MEMS) based sensors.

Different types of sensors are required to estimate the quadcopter's orientation. The Horizon mode we are using in this project requires an accelerometer, gyroscope and magnetometer. Other sensors can be used to provide various types of measurements such as atmospheric pressure or distance which can be used to maintain the altitude.

6.3.1. Accelerometer

Accelerometer provides linear acceleration on three axes in g. Value of g is 9.81 m/s^2 and it is equivalent to acceleration caused by earth's gravitational force. Due to the gravitational force the accelerometer measures 1 g when it is not moving. If the orientation of the sensor is neutral, 1 g is measured on Z-axis. Once it rotates around X or Y-axis, acceleration can be measured also on other axes. From this data it is possible to calculate the accelerometer's pitch and roll – the rotation around the X and Y axes.[7]

Rotation around an axis that is pointing to the same direction as the gravitational force vector cannot be measured. For example, in the neutral orientation the Z-axis is pointing to the same direction as the gravitational force vector. Rotation around the Z-axis does not affect the acceleration measured on X or Y axes and therefore it results to 1 g on Z-axis

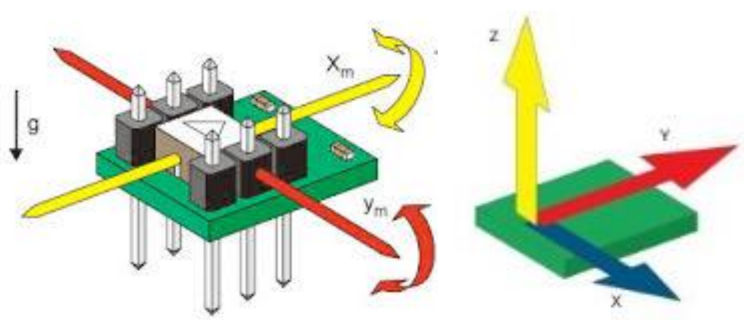


Figure 9: Accelerometer

The quadcopter's orientation can be calculated from the accelerometer data using the following equations:

Pitch = atan(accelerometer.Y, accelerometer.Z) * 180 / Pi 6

Roll = atan(accelerometer.X, accelerometer.Y) * 180 / Pi 7

Rotation around the X-axis is called the pitch and it can be measured on the Yaxis. Rotation around the Y-axis can be measured on the X-axis and it is called the roll. Both values are converted from radians to degrees. Accelerometer is easily affected by noise, such as the quadcopter's motors causing the frame to shake. Therefore it is important that the accelerometer is installed on a noise reduction material such as rubber, and that it is used together with a gyroscope and a complementary filter.

6.3.2. Gyroscope

The gyroscope measures angular velocity on three axes in dps (degrees per second) [7]. The quadcopter's orientation cannot be calculated from a single sensor measurement like it can be with the accelerometer. When the quadcopter has no angular velocity on an axis, the gyroscope measures 0 dps. When the quadcopter begins to rotate around an axis, the gyroscope will measure the angular velocity and it will be integrated by the microcontroller to estimate the quadcopter's orientation. [7]

pitch = pitch + gyroscope.X * deltaTime 8
roll = roll + gyroscope.Y * deltaTime 9
yaw = yaw + gyroscope.Z * deltaTime 10

This way the gyroscope can be used to measure the offset from its original rotation around an axis. Theoretically the original rotation is 0 degrees pitch and roll when the quadcopter is placed on a flat surface. Delta time is used as a multiplier because the quadcopter's flight controller loop executes n amount of times in a second, and the measured unit is degrees per second. The faster the angular velocity can be measured and integrated the more accurate the result is. However there will be always minor inaccuracies due to the sensor's refresh rate, measurement accuracy and floating-point operations. This will result into drift which can be observed when the quadcopter rotates away from the original rotation and returns back to the original rotation - the calculated orientation will have some offset from the original position. When the integration is run for a longer time these inaccuracies can increase and decrease.

The gyroscope is not prone to noise caused by oscillation of the quadcopter's frame and, therefore, it is used to compensate the noisy measurements of the accelerometer.

3.3.3. Magnetometer

A magnetometer measures the earth's magnetic field in G (gauss) and it is used to calculate the quadcopter's geographical heading. The magnetometer readings are converted into polar coordinates which behave similar to a compass. The polar coordinates measure rotation around the quadcopter's Z-axis in 360 degrees, where north is 0 degrees and south is 180 degrees. The magnetometer helps to compensate the gyroscope's drift when calculating yaw for the quadcopter.[7]

3.3.4. Barometer

A barometer measures atmospheric pressure in hPa (hectopascal). When the barometer is properly calibrated and has an accurate resolution it can measure even small changes in the atmospheric pressure when quadcopter's altitude changes. The barometer is used to maintain the quadcopter's altitude.

However, changing weather conditions can change the atmospheric pressure drastically and in that case the barometer's measurement will drift from the original altitude and maintaining the altitude is not accurate. The barometer can provide accurate altitude information only short term. A rangefinder would provide much more accurate measurement of the quadcopter's altitude relative to the ground level. It can measure distance to the ground and therefore it can be used to adjust altitude when the terrain is not flat.

This page left intentionally blank

Chapter 4

Assembly and installation

In this chapter the whole process of building the quadcopter will be gone through. For that instance, the parts that came within the kit will be used along with a few additional components that were not part of it. Since no information or assembly guide was provided with the pack, a whole set of written instructions along with figures has been written to allow a future student to continue this work.

In order to make this process more understandable it will be split in two sections: at the hardware part and at the software level.

4.1. Hardware

First of all, it is highly recommended to cross-check all the material in the kit against the table below and contact the seller in case anything is missing. All the pack contents can be found in the table below:

Table 7: Kit components

Pieces	**Quantity**
Upper frame plate	1
Upper frame plate	1
Arms	4
M2,5*6 Mounting screws	24
M3*8 Mounting screws	16
RS2205 Brushless Motors 2300kv	4
SP Racing F3 Flight Controller	1
3S 30A Brushless ESCs	4
5 inch Propellers (ccw)	2
5 inch Propellers (cw)	2
Transmitter (6 channels at least)	1
Receiver	1
3S 1300-2600mAh LiPo battery	1
AC/DC balance charger	1
Ultrasonic sonar sensor	1
PDB	1

4.2. SP Racing F3

The SP Racing F3 Flight Controller is built for FPV Racing. It is designed to use the latest and greatest hardware while keeping the board cost to a minimum, at the end of the day when pushing your FPV racer to the max you will eventually crash, so having an affordable flight controller with incredible performance means that you can push a little harder, or try something new without breaking the bank if you crash.

4.2.1. SP Racing F3 Key Features

- Micro SD card reader – can record flight data logs easily onto and SD card (not limited to 128mb like on other flight controllers)
- Stackable – Can mount onto compatible PDB's or OSD's with just 2 through hole pins in each corner.
- Next Generation CPU and Sensors – Using the ARM Cortex-M4 72Mhz CPU along with MPU9250 connected directly via SPI (8Khz sensor update rate), to ensure processing of more sensor data than ever before.

4.2.2. SP Racing F3 Pinout

The UART2 port is primarily used for R/C input, be it PPM, SBUS or Spektrum, there is a variety of connectors for each. If you connect your R/C receiver via PPM then you can still use the UART2 out to send telemetry info back to your receiver (such as an X4R, or D4R). However, if using spektrum or and SBUS receiver then the UART2 out cannot be used.

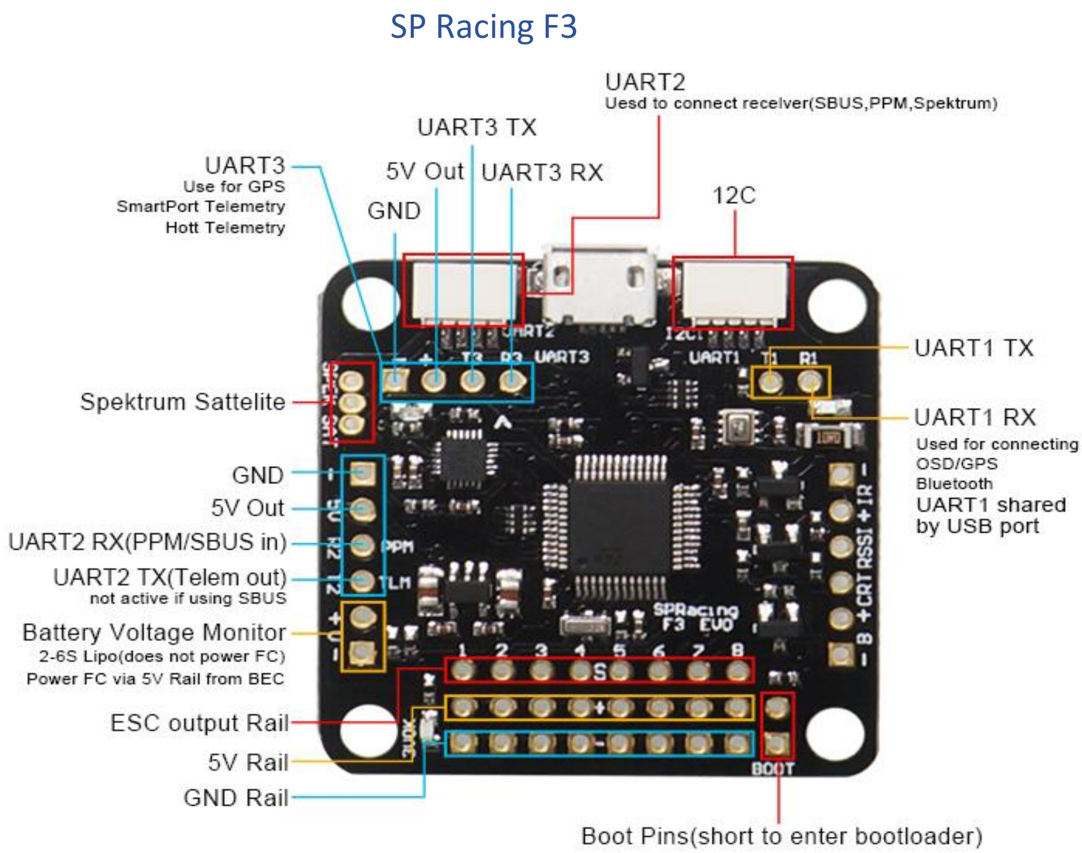


Figure 10: SP Racing F3 Pinout

4.3. Connection Diagram

The diagram below is a typical connection for a FPV racing quadcopter. There are some things that you can do differently, but this is just as a guide if you have no idea on what to do. This diagram connects the receiver in PPM mode (so that you can still use the same UART port for telemetry out)

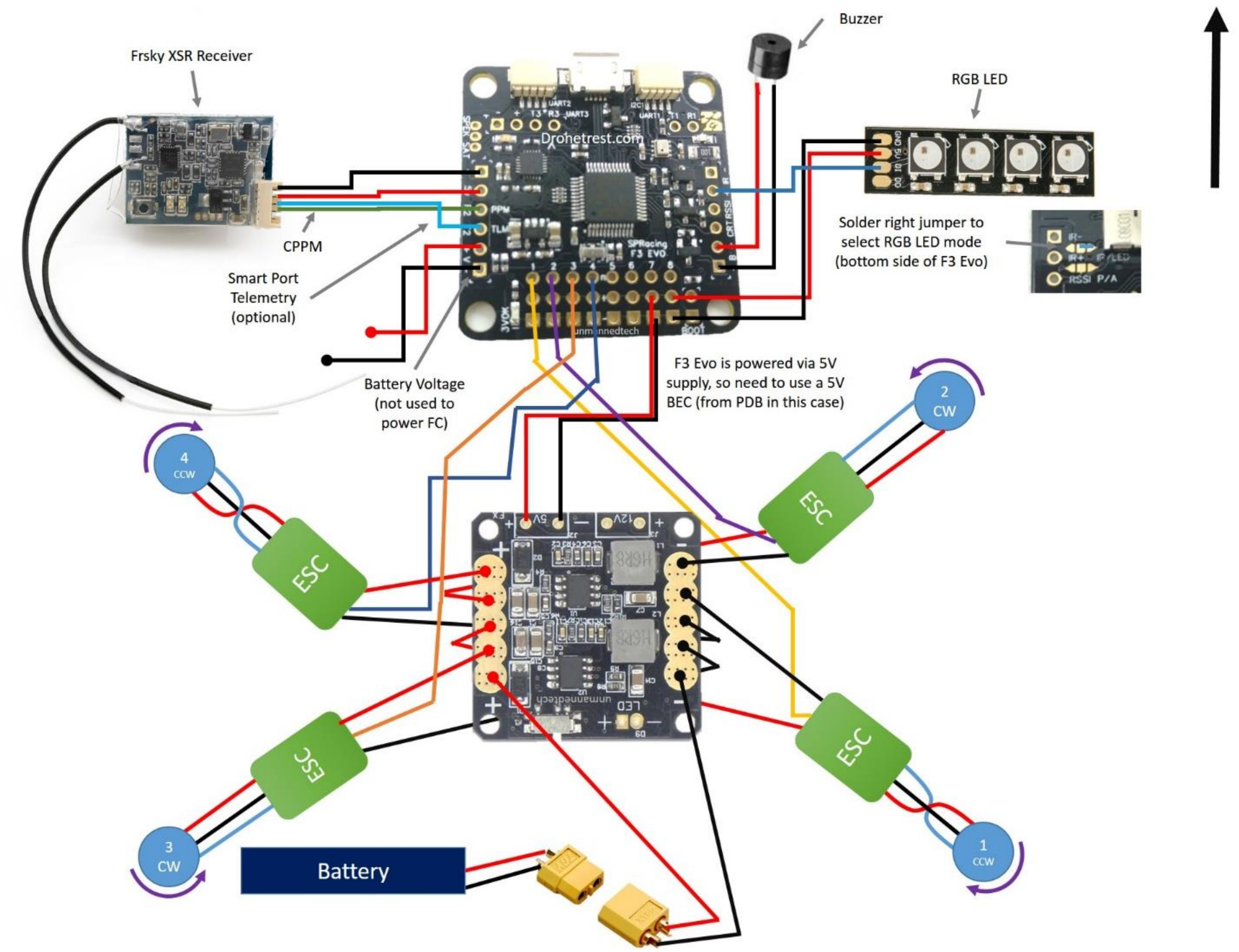


Figure 11: Quadcopter Connection Diagram

4.4. iNavflight

iNavflight is a fork of Cleanflight which is designed to make the best of GPS and navigation capabilities. Cleanflight and BetaFlight don't have the best GPS capabilities – so some developers decided to fork Cleanflight, keep the same interface, and adapt the code to have navigation. (GPS is going to be the main objective in the future works for our project)

Since it's a fork of Cleanflight, iNavflight works on most common flight control boards. Moreover, it be can be used it for multirotors, flying wings, and planes.

iNav supports a lot of flight modes: ALT hold, GPS hold, magnetic hold (quad will respond to stick inputs according to original orientation), Waypoints, and Return To Home.

4.4.1. Flashing

For configuring and flashing iNavflight, we need the iNavflight configurator, which has a chrome extension, or may it can be installed as independently software.

Note: The VCP drivers should be installed in the computer to be able to talk to the flight controller.

To flash iNavflight, we connect flight controller to the computer via USB while holding down the BOOT button (or shorting the BOOT jumper).

When successfully connected, the Configurator will recognize a device in DFU mode – which will be reflected in the port selection tab at the top.

Next, we click on the Firmware Flasher tab and select the correct flight controller board (SP Racing F3 for this project) and the latest release of the firmware, "Full Chip Erase" should be selected and then we click Load Firmware Online.

Once the firmware is downloaded, the Flash option will become selectable. We click it, and wait for flashing to complete.

Then if all goes well, we will see a message saying:

PROGRAMMING SUCCESSFUL

At this point, we unplug the flight controller, and plug it in again. Now, we can connect to the board and see settings.

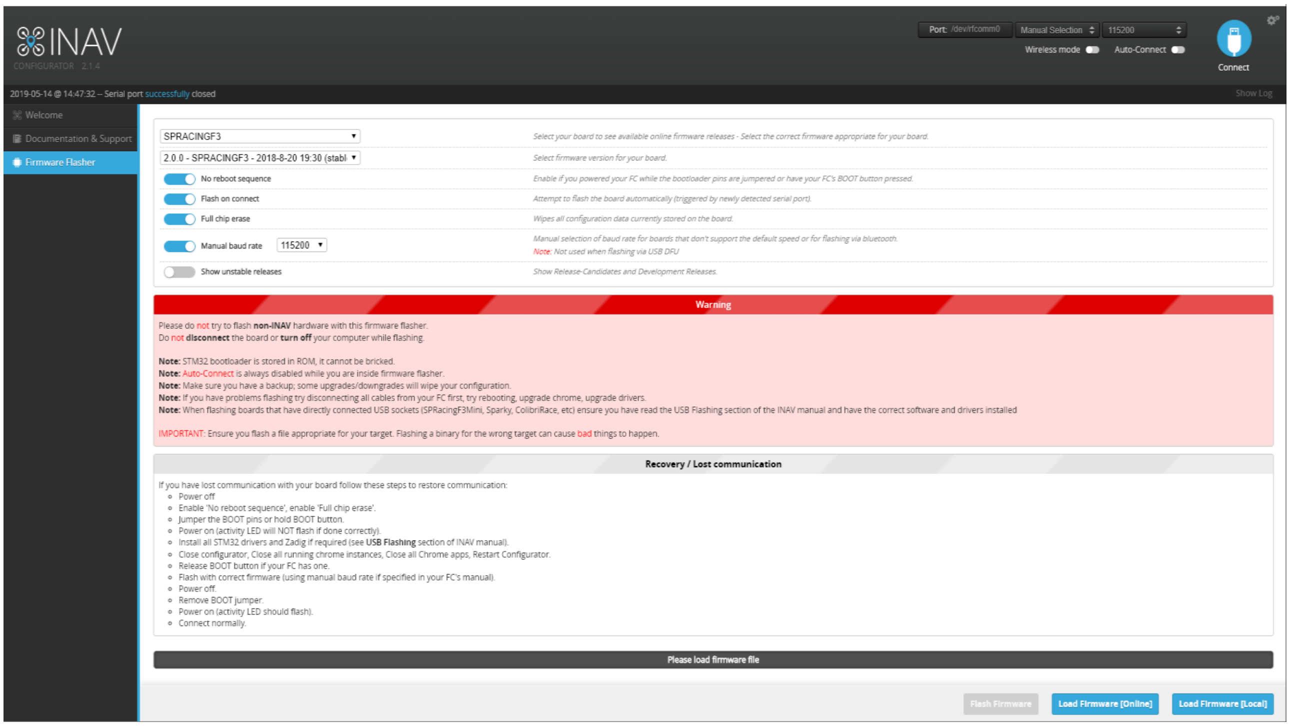


Figure 12: Flashing Flight Controller in iNav

4.4.2. Presets

Because iNav can run very wide spectrum of aircrafts, it suffers a penalty of very general default values. Different frame sizes, propellers, multirotors, airplanes, flying wings. It is impossible to have one set of default values that will make all those different UAVs fly good right after flashing. And it’s not only about PIDs. Also, rates, filters and other settings differs between them. Good default tune for 5″ racer will not work on 1.2m flying wing or 1.6kg quadcopter on 12″ propellers.

To solve this problem, iNav Configurator 1.6 introduces new tab:

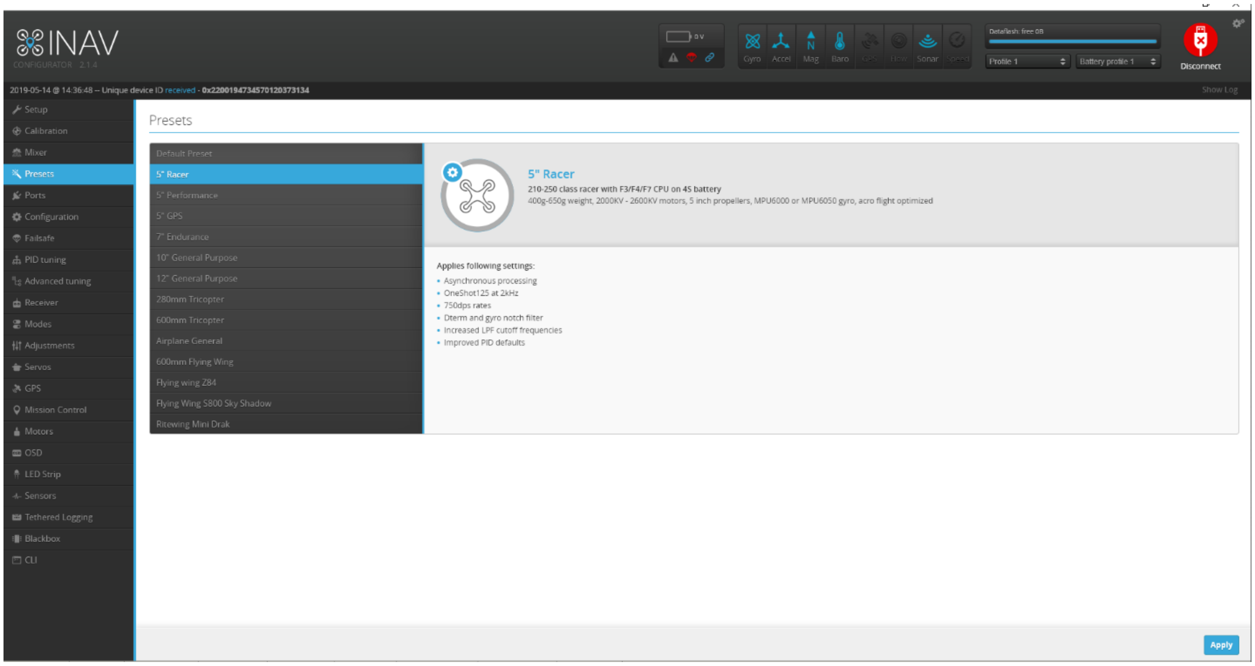


Figure 13: iNav- Presets

Presets tab allows to apply default settings designed for a specific type of UAV:

- 5″ Racer Quadcopter
- 10″ Propeller Multirotor
- 12″ Propeller Multirotor
- Generic airplane
- 600mm Flying Wing

Presets do not overwrite all settings. While filters or PIDs will change from preset to preset, Failsafe, Modes or Ports are never altered.

For this project, the best choice that is capable with our objectives is 5’’ Racer

4.4.3. Ports

The ports tab is quite critical to make sure that the setup is running correctly!

Here, you'll assign the peripherals to whatever UARTs we soldered them up to. (Please see the connection diagram).

We'll have one UART for Serial RX(we're using SBUS, and there's no reason you should be using anything else!)

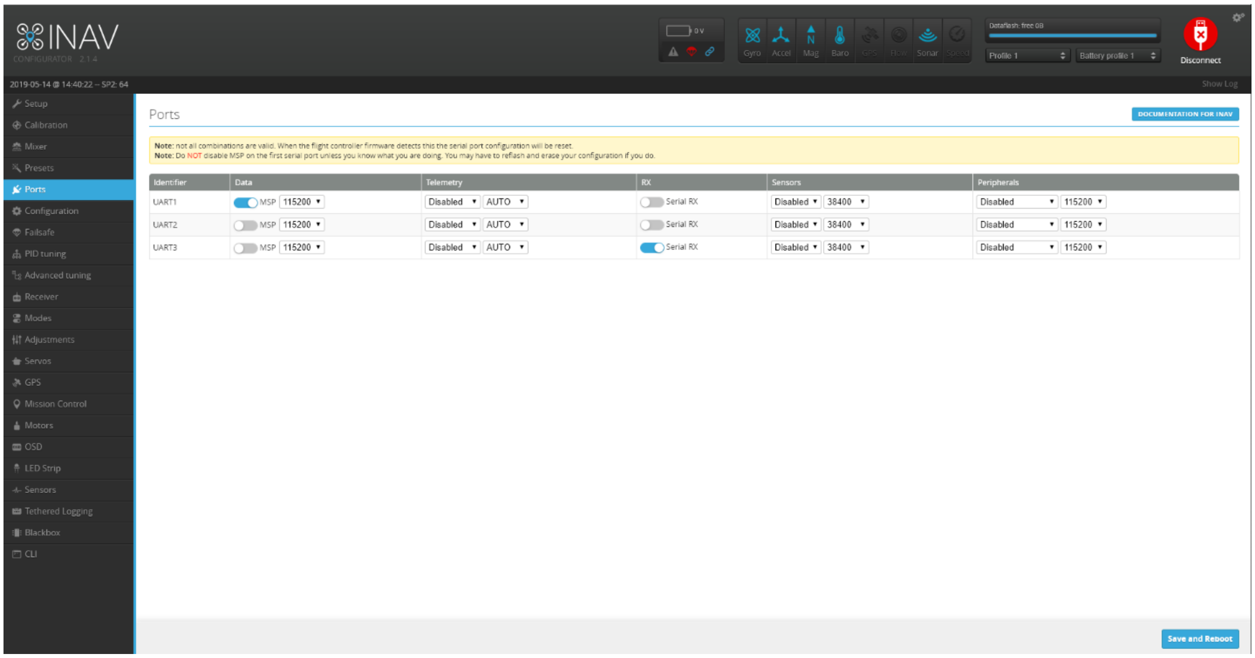


Figure 14: iNav– Ports

4.4.4. Configuration

In the Configuration tab, the first screen is the mixer – since we used a Preset from earlier, this is already correct.

Under the Mixer, we have the Sensors tab. Here, we'll have to select all of our peripherals. Most likely, the firmware will already pick them up – but if any of the sensors are showing up red at the top of the screen, we may have to do a bit of process of elimination to get the right one. Most likely just selecting AUTO from each menu will do the trick.

For a multirotor, we'll only need the Accelerometer, Magnetometer, and Barometer. We select auto for each, hit save and reboot, and then see the red icons magically turn blue.

Under ESC Motor Features, the first toggle says "Enable motor and servo output" – unless this toggle is checked, the motors will not spin up! This is fine for now, but make sure you enable this before your final setup and maiden flight.

We can choose our ESC protocol(go for Multishot if your ESCs support it – BLHeli's ESCs certainly do), change the refresh rates, and change your other settings.

(We are already using BLHeli's ESCs)

Finally, under system configuration, you can adjust the cutoff frequency and adjust looptime.

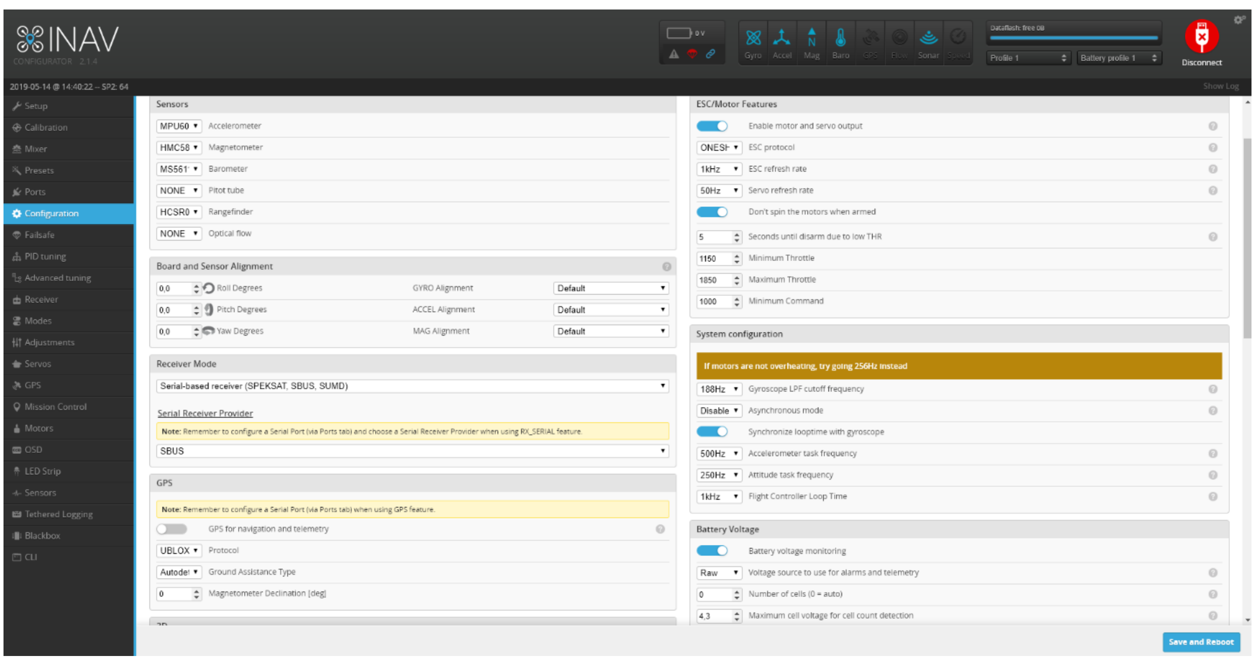


Figure 15: iNav– Configuration

We also can adjust the board and sensor alignment. We need to do this *if* the board is not mounted with the arrow facing forward. If it isn't facing forward, just adjust the YAW degrees to match the setting.

Note: You may have to touch MAG alignment after calibrating the compass, but we'll come back to that during our final calibration and setup.

Under that, there is the Receiver tab. We select the type of receiver we are using, and the correct protocol (Serial and SBUS for our project)

Provided you are using a GPS module, enable the toggle, and select the correct protocol. Most common protocols are UBLOX, but check your GPS module to make sure.

On the right of the screen, we can adjust battery voltage and current monitoring.

Under other features, we enable the features we need.

- telemetry (select if your receiver supports it, ours do)
- Airmode (as a safeguard in case you cut the throttle by mistake)
- Sonar (if you have)

4.4.5. Failsafe

The next tab is the Failsafe tab. This is CRITICAL! Make sure you set this up. You have four choices in failsafe.

- Drop: Your copter kills all inputs and plummets to the ground
- Land: Your copter will stabilize itself and do a controlled descent towards the ground (you'll have to configure the throttle value and delay for motors off)
- RTH: Your copter will come back to the spot it was armed in (provided you had a good GPS fix)
- Do nothing: Self-explanatory, your copter will continue flying in the direction it was going in until it crashes into something or the battery runs out! (not recommended).

Ideally, you'll want to choose RTH – return to home – as that's the whole point of using iNav! However, if you're setting this up on a race quad or a smaller build with tighter battery capacity, you may wish to employ land or drop.

Here, we choose Land until we configure the GPS later in the future work.

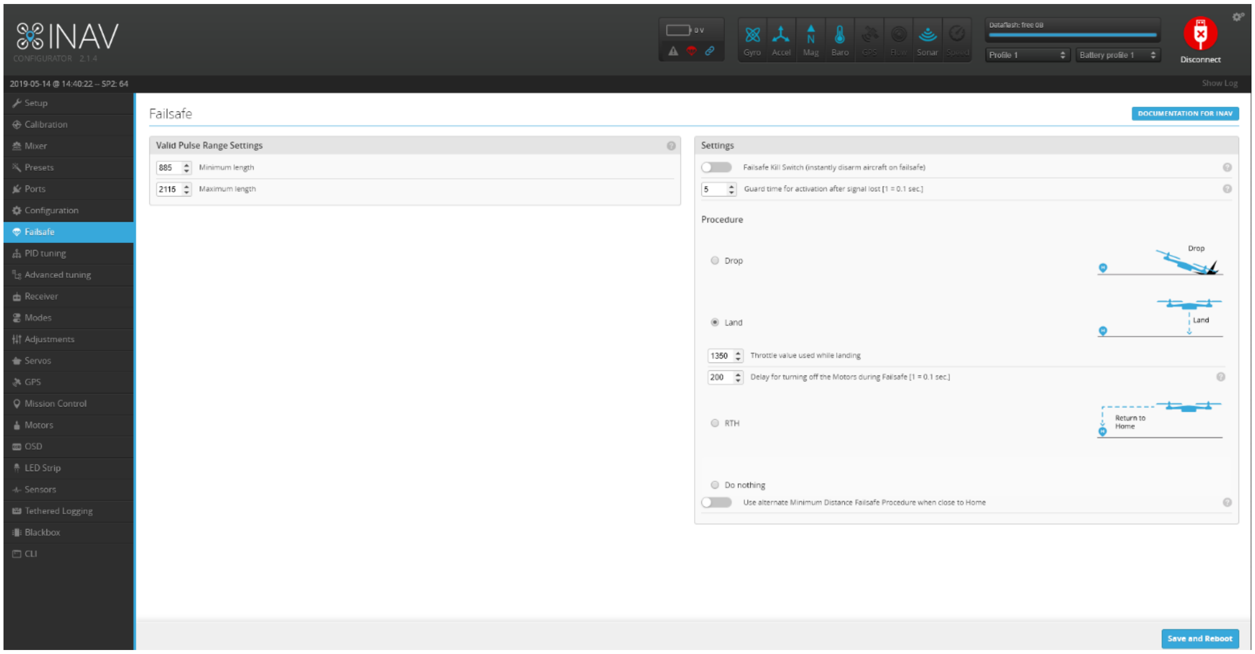


Figure 16: iNav- Failsafe

4.4.6. Advanced tuning

The advanced tuning tab is for changing more specific GPS and navigational settings.

Under basic navigation settings, we can adjust the settings to change the behavior of our copter in the navigation modes – but we've left mine at stock right now.

The only thing we'll want to change is the Hover Throttle value which we'll have to determine by flying the copter without any nav modes turned on and see what throttle is required to maintain a hover. Then enter that number (even rough estimate is fine), save, and test the ALT hold mode to see if the copter can maintain altitude or not.

Below that, we have RTH and landing settings – these are also quite self-explanatory, and we have left mine at stock for now. In RTH, the copter will first climb, turn around to face the home direction, fly back, turn into original orientation, and then land.

This is with default settings – you don't really need to change anything on this screen for the time being.

4.4.7. Receiver

The receiver tab is pretty much the same as the receiver tab in CF and BF. Provided you've bound your transmitter and receiver, and configured your receiver correctly in Ports and Configuration, you should be able to see the bars moving according to your transmitter.

Please note that the channel map here is a bit different: AETR5678 means channels 1-4 are AETR, and the other 4 channels are 5678.

So, if you are using a Taranis (as we do), you would set it to TAER5678.

If you're unable to change the channel map, your receiver probably doesn't work with iNav…yet.

So, make sure if your receiver work with iNav.

4.4.8. The Modes

The next tab to set up is the Modes tab. Here we'll assign all of our flight modes to switches on the transmitter.

We would recommend having at least two, here, four switches configured so we can utilize all of the flight modes correctly.

We'll take you through our recommended setup, in which we'll only talk about the modes that we use – but you're more than welcome to refer to the documentation to read up on the other flight modes, too.[8]

Here's how we have set our quadcopter up:

- **Angle**: In this auto-leveled mode the roll and pitch channels control the angle between the relevant axis and the vertical, achieving leveled flight just by leaving the sticks centered. Maximum banking angle is limited by max_angle_inclination_rll and max_angle_inclination_pit
- **Horizon**: This hybrid mode works exactly like the previous ANGLE mode with centered roll and pitch sticks (thus enabling auto-leveled flight), then gradually behaves more and more like the default RATE mode as the sticks are moved away from the center position. Which means it has no limitation on banking angle and can do flips.
- **Althold**: When activated, the aircraft maintains its actual altitude unless changed by manual throttle input. Throttle input indicates climb or sink up to a predetermined maximum rate (see CLI variables). Using ALTHOLD with a multicopter, you need a barometer.
- **Surface**: Enable terrain following when you have a rangefinder enabled

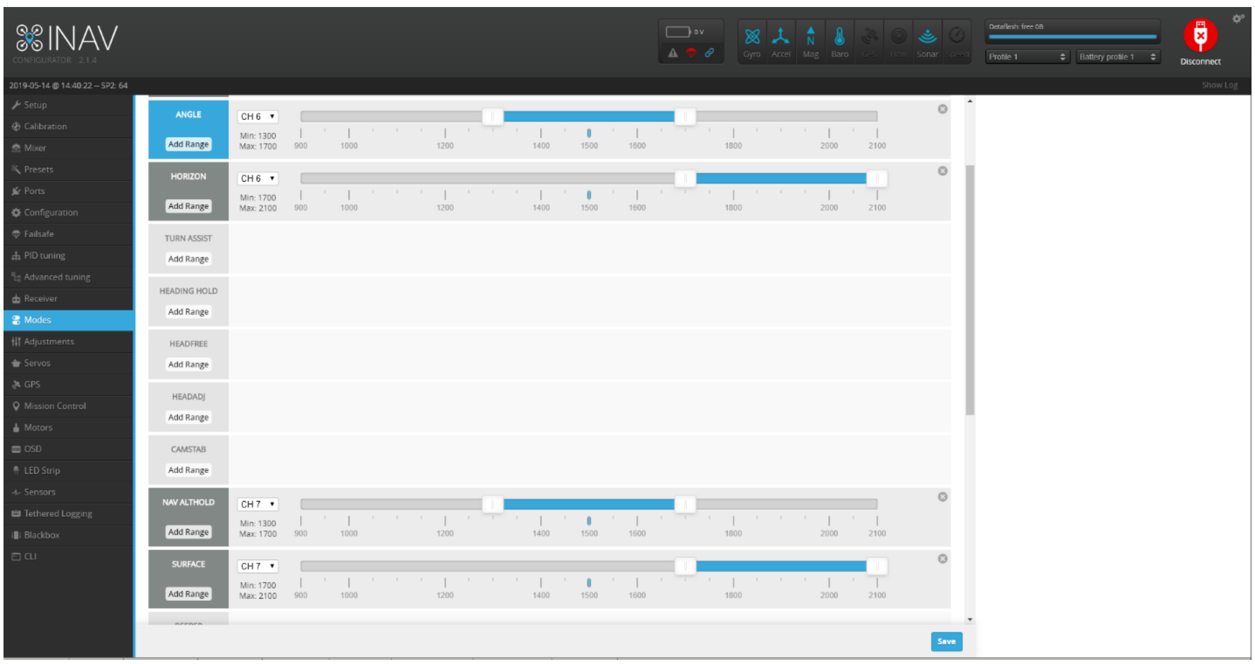


Figure 17: Flight modes used in the project

4.4.9. Motors

The motors tab is where we'll calibrate our ESCs, find min_throttle, and check that the motor directions are okay.

To calibrate ESCs, make sure the propellers are off (important for safety), flick on the "I understand" toggle, raise Master to full value, and plug in your battery.

The ESCs will go through their tones. Once the tones are done, lower master to zero, and the ESCs will finish their tones. Now unplug, plug in again, and raise Master very slowly until the motors are spinning comfortably.

Make a note of the value this is at.

You'll also want to make sure the motors are spinning correctly – if they're not, head over to the BLHeli configurator and change the motor orientations until it's correct.

Now head back to the Configuration tab, and change Min Throttle to about 20 points greater than the value you noted earlier. Hit Save and Reboot.

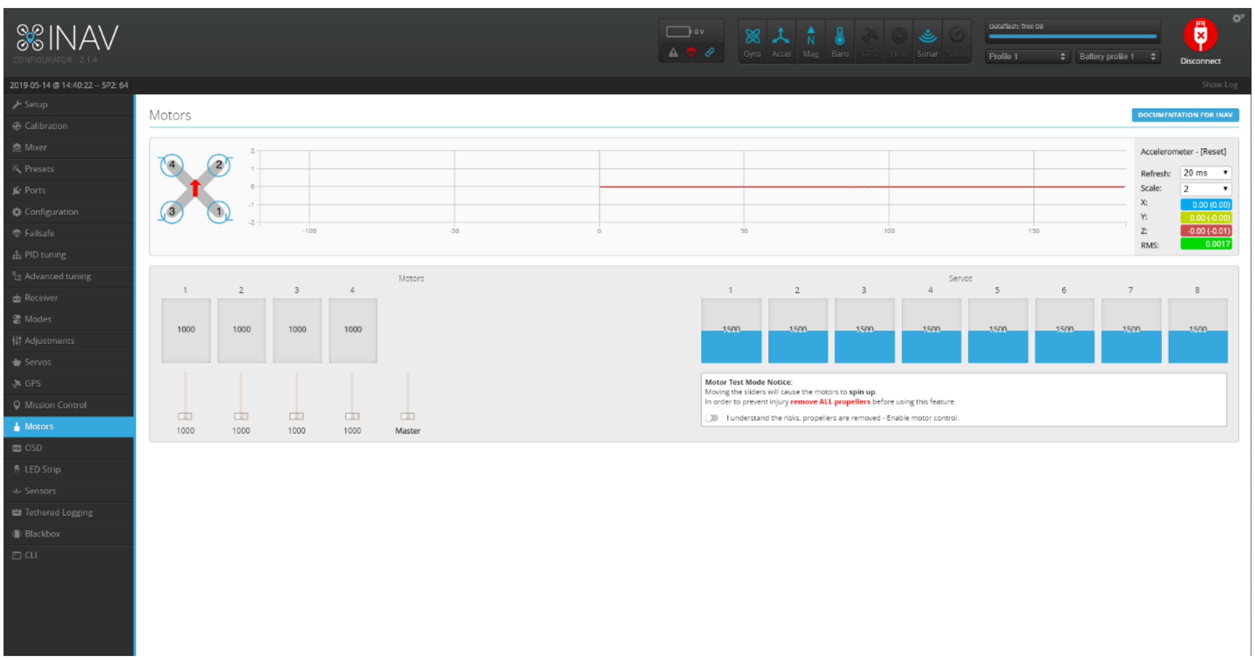


Figure 18: iNav- Motors

4.4.10. Calibrating Sensors

Before we can start flying, we must do two vital calibrations: the accelerometer calibration and the magnetometer calibration.

4.4.10.1. Accelerometer calibration

To calibrate the accelerometer in iNav, you have to go through a six step process.

1. To start, keep your copter flat on the ground, and press the "Calibrate Accelerometer" button on the Setup screen. Once it's done, there will be a SUCCESS message at the top of the screen in green text.
2. Now flip the copter upside down and press the "Calibrate Accelerometer" button again. Wait for success confirmation.
3. Now hold the copter perpendicular to the ground with the front of the copter facing right. Click the button, wait for confirmation.
4. Continuing to hold it perpendicular, now face the copter straight towards the sky. Click the button, wait for confirmation.
5. Now face the copter towards the left, continuing to hold it perpendicular to the ground. Click the button, wait for confirmation.
6. Finally, keeping it perpendicular, face the copter to the ground, click the button, wait for confirmation.

After these 6 steps are completed successfully, calibration is done. Don't worry if you don't get the position of the copter exact each time – just get it as close as possible to 90 degrees from the last position.

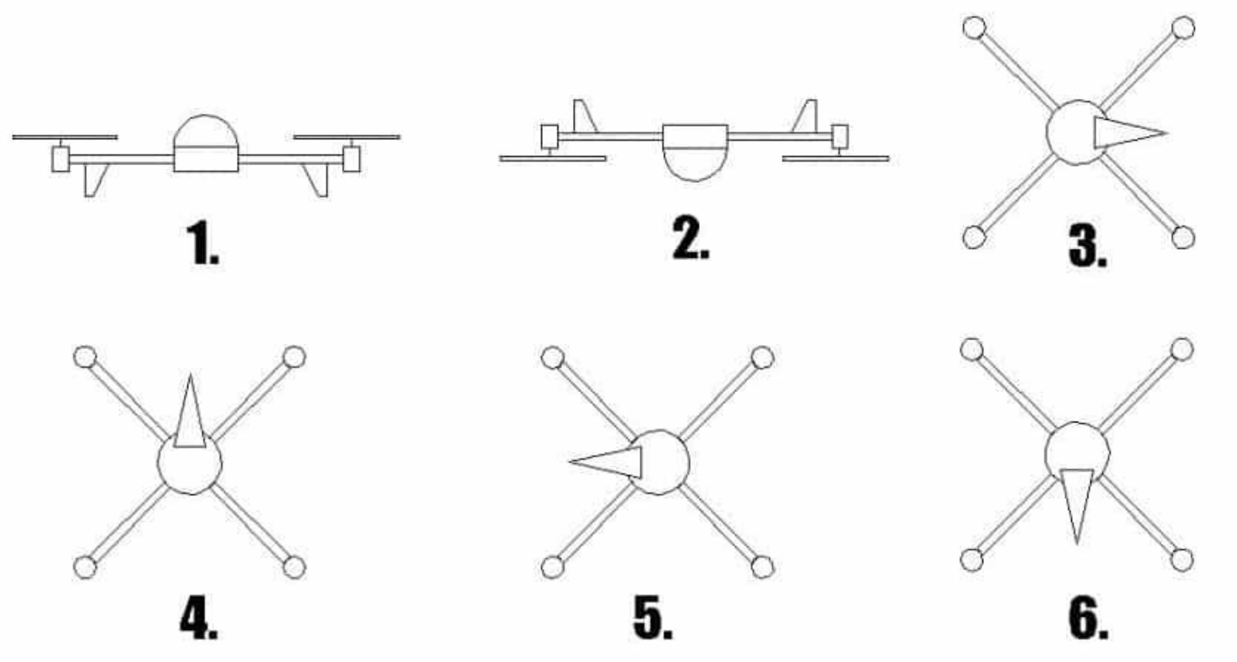


Figure 19: Quadcopter Accelerometer Calibration

To confirm, head to the CLI, type *dump*

And check if the values accgain_x, accgain_y and accgain_z are NOT 4096

Additionally, the values acczero_x, acczero_y, acczero_z are NOT 0.

If either of those are not true, you'll have to repeat the calibration process.

4.4.10.2 Magnetometer calibration

To calibrate the magnetometer, click the Calibrate magnetometer button, and within 30 seconds, rotate the copter 360 degrees along all 3 axes. It doesn’t have to be exact!

Just make sure you are doing this far away from magnetic interference (large metal objects, high levels of electric current).

To confirm, head over to the CLI, type *dump*

And look for the values magzero_x, magzero_y and magzero_z are no longer “0”

If that is the case, you’re good to go! In the Setup screen, just check what the “Heading” of the copter is and if it is in fact the direction it’s pointing in!

0 degrees is N

90 degrees is E

180 degrees is S

and 270 degrees is W

4.5.UltraSonic Sonar Sensor

Ultrasonic distance sensor determines the distance to an object by measuring the time taken by the sound to reflect back from that object. The frequency of the sound is somewhere in the range of ultrasound, this ensures more concentrated direction of the sound wave because sound at higher frequency dissipates less in the environment. A typical ultrasonic distance sensor consists of two membranes. One membrane produces sound, another catches reflected echo. Basically, they are speaker and microphone.

4.5.1. How does it work?

The ultrasonic sensor uses sonar to determine the distance to an object. Here's what happens:

1. The transmitter (trig pin) sends a signal: a high-frequency sound.
2. When the signal finds an object, it is reflected and…
3. … the transmitter (echo pin) receives it.
4. The time between the transmission and reception of the signal allows us to know the distance to an object. This is possible because we know the sound's velocity in the air

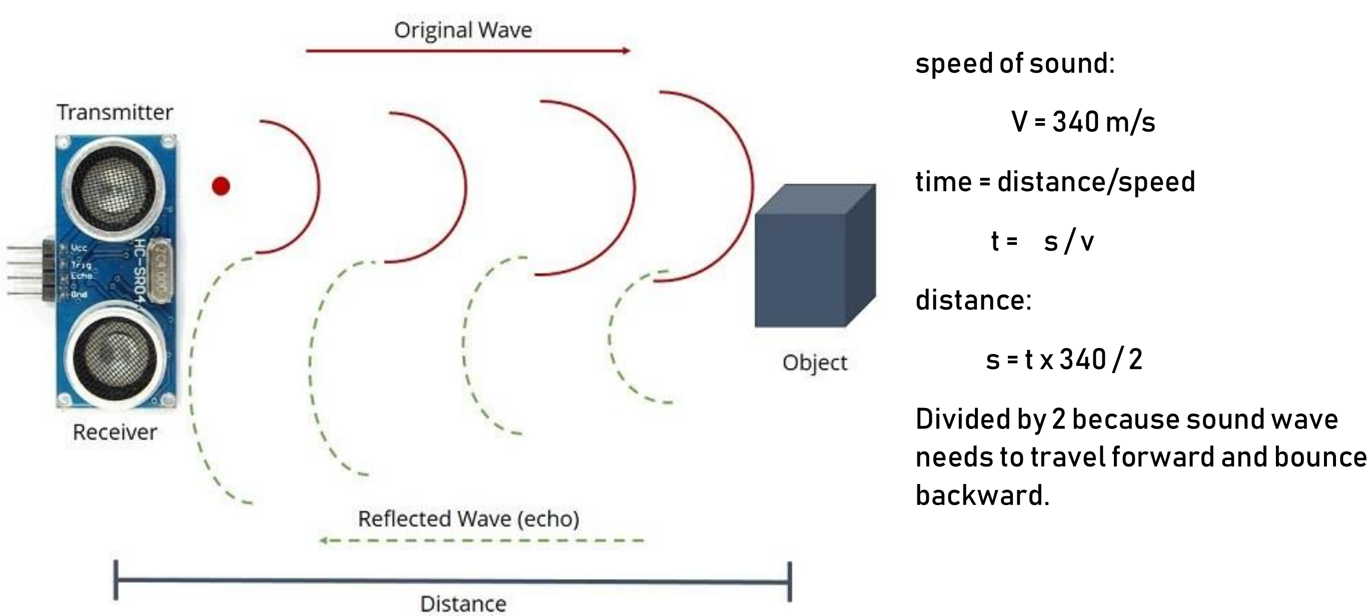


Figure 20: Principle of Ultrasonic Sensor

4.5.2. Connection to SP Racing F3

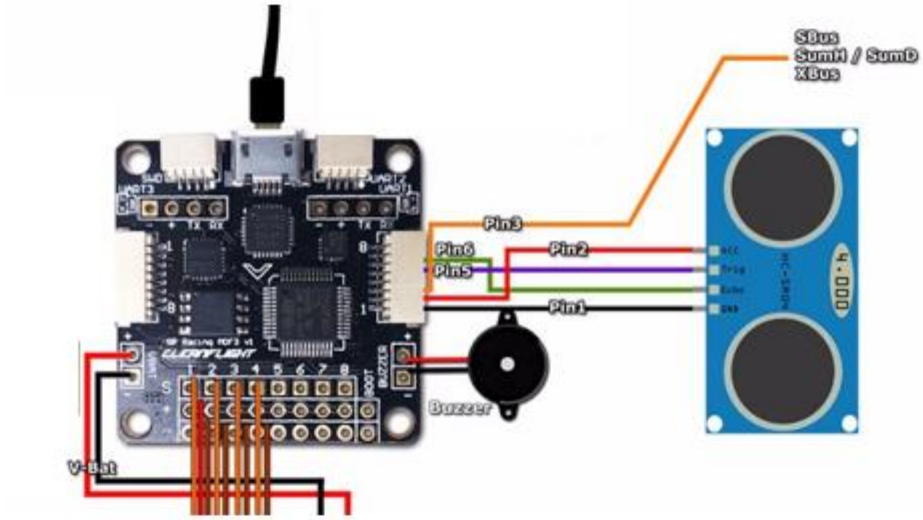


Figure 21: Sonar to SP Racing F3

4.5.3. Sonar configuration in iNav

Please remember that we have already enabled Sonar in configuration tab under "other features" section.

Things to consider:

Since FC now has a way to measure its altitude above ground, altitude hold Sonar mode will be available. In theory, it should work very good and allow altitude stabilization with centimeter accuracy. Unfortunately, this is not that simple, and there are a few reasons for that:

1. In perfect conditions, sonar can measure distance up to 4 meters. Conditions onboard drone are not perfect. Disturbed air and noise can lower this range to less than 1 meter
2. Measurement cone is only about 45 degrees. That means, if surface (ground) is tilted above 22.5 degrees, measurement will probably fail
3. Accuracy and range above grass or any not very solid surface (like concrete or asphalt) will be greatly reduced.
4. HC-SR04 **- DO NOT USE** HC-SR04 while most popular is not suited to use in noise reach environments (like multirotors). It is proven that this sonar rangefinder start to report random values already at 1.5m above solid concrete surface. Reported altitude is valid up to only 75cm above concrete. **DO NOT USE**

5. US-100 in trigger-echo mode - Can be used as direct replacement of HC-SR04 when rangefinder_hardware is set to HCSR04. Useful up to 2m over concrete and correctly reporting out of range when out of range

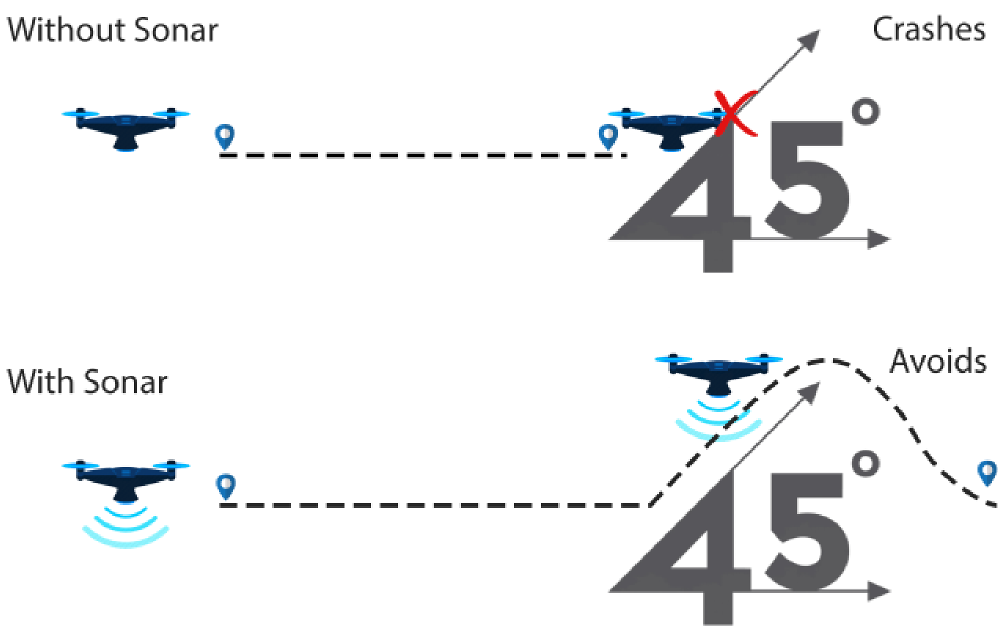


Figure 22: Sensors

Chapter 5

CLI Commands

In order to start using CLI you have to connect to your flight controller with configurator software. Click on the "CLI" tab – this will enable flight controllers CLI mode. Write the command in the input field and hit the [Enter] to execute it.

You need to execute save command if you want all the commands and changes to be saved into Flight Controllers memory. Saving will reboot the flight controller. Close the CLI tab or send the exit command to exit the CLI mode without saving.

5.1. Dump or Diff

dump command dumps all the setting of the flight controller.
diff command dumps only the settings that are different from the defaults.
diff all command dumps only differences from the defaults but also dumps all the PID profiles.

It is highly recommended to use **diff** instead of the **dump** as it results in the significant smaller settings list and only the ones that were changed or differ from the defaults.

5.2. Backup/Restore settings

Sometimes it is useful to save the Flight Controller settings and restore them in case we need. These cases include FC firmware upgrade, restoring the FC to the point where it worked.

Backuping and restoring is very simple task.

Backup: Go to CLI, enter **diff** (or **diff all** if you have multiple PID profiles) and copy the provided text to anywhere just to save it for later use.

Restore: Go to CLI, paste the saved settings text to the command input field, hit **[Enter]**. Your FC settings are restored. Don't forget to save the settings.

5.3. CLI commands list

Table 8: CLI commands list

Command	Description
1wire <esc>	passthrough 1wire to the specified esc
Adjrange	show/set adjustment ranges settings
Aux	show/set aux settings
Color	configure colors
Defaults	reset controller to defaults and reboot
Feature	list or -val or val
Get	get the value of the specified variable
Gpspassthrough	passthrough gps to serial
Help	lists all commands
Led	configure leds
Map	mapping of rc channel order
Mixer	mixer name or list
mode_color	configure mode colors
Motor	get/set motor output value
play_sound	index, or none for next
Profile	index (0 to 2)
Rateprofile	index (0 to 2)
Rxrange	configure rx channel ranges (end-points)
Rxfail	show/set rx failsafe settings
Save	save settings and reboot
Serialpassthrough	serial passthrough mode, reset board to exit
Set	set the the given value to the variable. (name=value or blank or * for list)
Status	show system status
Version	show version
Serial	configure serial ports
Servo	configure servos
sd_info	sdcard info
Tasks	show task stats
Mmix	design custom motor mixer
Smix	design custom servo mixer

Note that specific command list depends on the type of the firmware (BetaFlight, Butterflight, iNav) and firmware version you are using. For exact list of the commands type **help** in CLI mode.

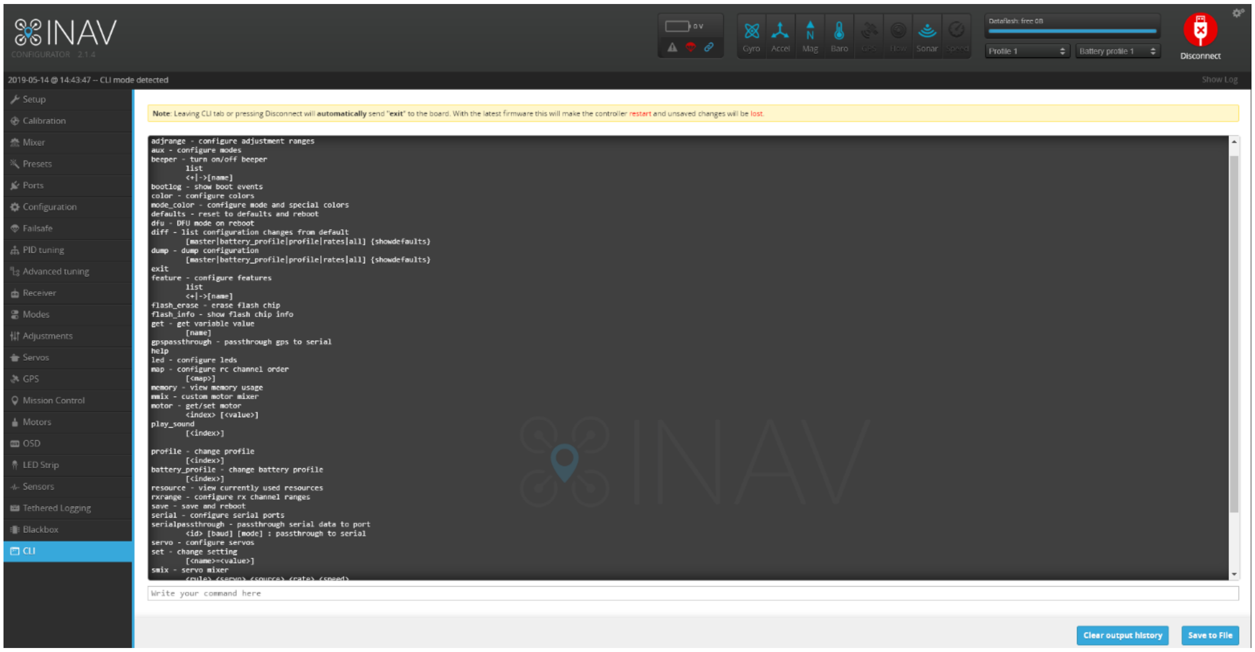


Figure 23: iNav CLI

This page left intentionally blank

Chapter 6

Discussion and Results

Throughout this paper, we have developed a quadcopter platform from scratch, including; system modeling, state estimation, control design, communication handling and implementation of a sonar sensor.

Regarding the results presented in this bachelor thesis, several conclusions can be inferred.

The primary goal was to find a vehicle that along with the appropriated components would best fit the mission requirements. This was translated into the selection of a robust DIY quadcopter with an open source processor that allowed us to introduce variants and improvements of the quadcopter behavior using iNavflight.

Regarding the stabilization of the vehicle through the tuning of its controller, important improvements were introduced at the point of making possible a hover flight. Even though it was tried to be as accurate as possible, the tuning method is not infallible due to the trial and error nature of the process.

Furthermore, with respect to the system used to fly the quadcopter indoors, it is seen that the tension of the wires contributes to great disturbances that unbalance the vehicle, but its stability is much better outdoors at high speeds.

Finally, as can be seen, the assigned goals were achieved to some extent. We successfully analyzed the existing hardware and built a quadcopter form scratch and derived a model that can avoid crashes, and thus, prepared the quadcopter for the future work.

Chapter 7

Future Work

The quadcopter possesses countless possibilities when it comes to functional capabilities. Some of the expansions we would consider to implement in the future, can be summarized as:

- One of the biggest steps we will do is getting a completely stabilized flight that will allow the Quadcopter to hover untethered in a single location. This should be done by using AutoTune mode that attempts to tune automatically the PID parameters of the vehicle in order to provide with the highest response without overshoot.
- Adding a camera and develop a target tracking algorithm using image processing techniques.
- Tracking of target equipped with a GPS device.
- After implanting the GPS device, we will work to make the quadcopter to reach a target location and then return home RTH.
- The initial goal of the future work is to develop the controller for autonomous flying of quadcopter along with the features mentioned above.

Chapter 8

References

[1] "Armed Quadrotors Are Coming". *Popular Mechanics*. Retrieved 29 December 2014.

[2] Global Unmanned Aerial Vehicle (UAV) Market Insights, Forecast to 2025 – QYResearch Jan, 2019

[3] Boletín Oficial del Estado 17.10.2014, number 252

[4] Román Estebn Hofer. Preliminary design of an unmanned aerial vehicle with application to photovoltaic farm inspection. 2014/2015

[5] Stafford, Jesse (Spring 2014). "How a Quadcopter works | Clay Allen". University of Alaska, Fairbanks. Retrieved 2015-01-20.

[6] A. Gibiansky, "Quadcopter Dynamics and Simulation", 2012

[7] iNEMO inertial module: 3D accelerometer, 3D gyroscope, 3D magnetometer – LSM9DS1 STMicroelectronics

[8] iNavFlight Documentation https://github.com/iNavFlight/inav/wiki

Chapter 9

Resume

- Rozan MUSTAFA

  Birth Date: 01 June 1992

  Address: Asmalievler Mah. 6612.Sok, 20160 Pamukkale/Denizli

  E-posta: rozanalmustafa@gmail.com

- Ömer Mustafa OĞUZOĞLU

  Birth Date: 05 July 1992

  Address: Asmalievler Mah. 6662 sokak no 31/1 Pamukkale/Denizli

  E-posta: omer1991mustafa@gmail.com

-