PEOPLE’S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
BELHADJ BOUCHAIB UNIVERSITY-AIN TEMOUCHENT
Faculty of electrical Engineering
Electronics Department

THESIS

To obtain the degree of

ACADEMIC MASTERs

Domain: SCIENCES & TECHNOLOGIES
Field: ELECTRONIC
Option: INSTRUMENTATION

Theme

Design, analysis and fabrication of "quadcopter, Radio controller" Based on Arduino board

Presented by

Mr. BENAHMEDI Lakhdar **Mr. AMMAR Abdelkarim**

At / 09 /2021 with the board of examinators:

Dr. S. BENTAIEB	M.C.B	UAT.B.B (Ain Temouchent)	President
Dr. C. BEMMOUSSET	M.C.B	UAT.B.B (Ain Temouchent)	Examiner
Dr. A. BENZINA	M.C.B	UAT.B. B (Ain Temouchent)	Thesis Advisor

University year 2020 / 2021

TABLE OF CONTENTS

Table of Contents

ABSTRACT……..........…………………………………………………………………… i

NOMENCLATURE ……..………………………………………………………………… ii

LIST OF FIGURES ..………………………………………….……………………… iv

LIST OF TABLES ………………………………………………………..………………... vi

GENERALI INTRODUCTION ………………………………...………………………… 1

I. LITERATURE REVIEW:

I.1. INTRODUCTION……………………………………………………….………………. 2

I.2. DEFINITION …………………………...………………………………………….……… 2

I.2.1. QUADCOPTER 2

I.2.2. AGRICULTURAL QUADCOPTER ……………………………………………….…..... 3

I.3. HISTORICAL BACKGROUND AND TRENDS IN AGRICULTURAL AVIATION ....…………....... 3

I.4.BACKGROUND STUDIES ………….……………………………………………….…..... 4

I.5.SCOPE OF THESIS.. 5

I.6.CONCLUSION ……………………………………………………………...………… 5

II. Vehicle Dynamic :

II.1. INTRODUCTION……………………………………...……………………………… 6

II.2. SIX-DEGREES OF FREEDOM (6DOF) ………………………………………….............. 6

II.3. ORIENTATION AND FRAMES OF REFERENCES ………………….……………………….. 7

II.4. EULER ANGELS ……………..……………………………………………………...... 7

II.5. EQUATIONS OF MOTION ……………………………………………………………… 9

II.5.1. VARIABLES ………………………………………………………………….......... 9

II.5.2. INERTIAL MOTION IN BODY FRAME …………………….…………………………. 10

II.5.3. THE CHAIN RULE ………………………………………..……………………. 10

II.6. NEWTON'S 2ND LAW OF MOTION ……………………………………………….......... 11

II.6.1. ETERNAL FORCES AND MOMENTS …………………………..………………… 12

II.6.2. TRUST …………………………………………………………………………… 12
II.6.3. GRAVITY IN BODY FRAME …………………………………………………… 13
II.6.4. MOMENTS OF INERTIA ………………………………………………………… 14
II.6.5. LINEAR ACCELERATION AND MOTION ……………………………………… 14
II.6.6. ROTATIONAL ACCELERATION ……………………………………………….. 15
II.6.7. ANGULAR VELOCITY AND EULER ANGLES………………………………… 15
II.6.8. INERTIAL COORDINATE ……………………………………………………… 16
II.5.9. NON-LINEAR EQUATIONS …………………………………………………… 16
II.6. CONCLUSION……………………………………………………………………… 17

III. COMPONENTS

III.1. INTRODUCTION…………………………………………………………………….. 18
III.2. COMPONENTS……………………………………………………………………… 18
III.2.1. QUADCOPTER COMPONENTS ………………………………………………... 18
III.2.2 1. Frame ………………………………………………………………………… 18
III.2.1.2. PROPELLERS ………………………………………………………………… 18
III.2.1.3. BRUSHLESS MOTORS ……………………………………………………… 19
III.2.1.4. ELECTRONIC SPEED CONTROLLER (ESC)……………………………….. 20
III.2.1.5. BATTERY…………………………………………………………………….. 21
III.2.1.6. TRANSMITTER ……………………………………………………………… 22
III.2.1.6.1. ARDUINO………………………………………………………....... 23
III.2.1.6.2. BATTERY MANAGEMENT SYSTEM (BMS)... 26
III.2.1.6.3. DC CONNECTOR …………………………………………………..… 27
III.2.1.6.4. BUCK/BOOST CONVERTER…………………………………………….. 27
III.2.1.6.5. BATTERY …………………………………………………………… 29
III.2.1.6.6. JOYSTICK …………………………………………………………… 29
III.2.1.6.7. TOGGLE SWITCH …………………………………………………… 30
III.2.1.6.8. I2C OLED SCREEN ………………………………………………… 31
III.2.1.6.9. RADIO MODULE …………………………………………………… 32

III.2.1.6.10. Push button …………………………………………………………………… **33**
III.2.1.6.11. RESISTORS ……………………….…………………………………… **34**
III.2.1.7. Receiver ……………………………………………………………..……… **34**
III.2.1.7.1. Regulator ………………………………….…………………………………… **35**
III.2.1.7.2. Prototype PCB ……………………………………………….………………… **35**
III.2.1.7.3. capacitor ……………………….……………………….…………………… **36**
III.2.1.8. Flight Controller ……………………………………………….…………………… **37**
III.2.1.8.1. Gyroscope ……………………………………..…………………… **37**
III.2.1.9. Spraying System…………..…………………………………………………… **38**
III.2.1.9.1. Water Pump……………………………………………..…………………… **38**
III.2.1.9.2. Servo Motor…………………………………………………………………… **39**
III.3. Software …………………………………………………………………………………… **39**
III.3.1. Arduino IDE ………………………………………………………………………… **39**
III.3.2 Fritzing………………………………………………………………………………… **41**
III.3.3. Graphical configurator “Multiwii” ……………………...………………… **42**
III.4. Conclusion……………………..……………………………………………………….. **44**

III. Designs and Realization of the Quadcopter

IV.1. Introduction…………………..………………………………………………………….. **45**
IV.2. Transmitter ……………………..……………………………………………………….. **45**
IV.3. Receiver ……………………..……………………………………………………….. **46**
IV.4. Quadcopter ……………………..……..…………………………………………….. **47**
IV.5. Spraying system:……………………...……..………….……………………..…… **50**
IV.5. Conclusion……………………..……………………………………………………….. **51**

GENERAL CONCLUSION …………………………………………………………………… **52**

Appendix

Appendix a……………………………………………………………………………….. **53**
Appendix b……………………………………………………………………………….. **56**

NOMENCLATURE

Nomenclature

PID – Proportional, Integral, Derivative

UAV – Unmanned Aerial Vehicle

ESC – Electronic Speed Controller

PWM – Pulse Width Modulation

PPM – Pulse Position

Modulation RPM – Rotations Per Minute

Hz – Hertz

MHz – Megahertz

LQR – Linear Quadratic Regulator

DC – Direct Current

ID – Identification

IMU – Inertial Measurement Unit

LiPo – Lithium Polymer

NED – North East Down

GPS – Global Positioning System

BLDC – Brushless DC

ESP – Espressif System

TCP – Transmission Control Protocol

IP – Internet Protocol

I^2C – Inter-integrated circuit

SCL – Serial Clock

SDA – Serial Data

VOTL – Vertical Take-off and Landing

MCU – Microcontroller Unit

FLA – Fast Lightweight Autonomy

KV – Constant Velocity

MEMS – Micro Electromechanical Sensor

3-D – Three Dimensions

6-DOF – Six-degrees of Freedom

SRAM - Static Random-Access Memory

AVR - Alf and Vegard's RISC processor

RISC - Reduced Instruction Set Computer

LIST OF FIGURES

LIST OF FIGURES

Figure I.1 Agricultural Drone………………………………………………..……………. 3

Figure I.2 Yamaha RMAX ……………………………………………………...………… 4

Figure II.1 Six-Degrees of Freedom …………………………………………………….. 6

Figure II.2 Inertial and Body Frames of References …………………………………...... 7

Figure II.3 Euler Angles ……………………………………………...………………….. 8

Figure II.4 Propellers Distance from center of gravity (CG) ……………………………… 12

Figure II.5 Motors rotation direction for yaw moment ……………………………......... 13

Figure III.1 Propellers …………………………………………………………………. 19

Figure III.2 Brushless Motor A2212/13T ………………………………………......... 20

Figure III.3 Electronic Speed Controller (ESC) 20A ……………………………….......... 21

Figure III.4 ZDF LiPo Battery 3300Mah ……………………...……………………………. 21

Figure III.5 Different types of Arduino boards ……………………...……………………… 24

Figure III.6 Presentation of the Arduino UNO board ………….………………………… 25

Figure III.7 Presentation of the Arduino NANO board ………………………………......... 25

Figure III.8 Battery Management System BMS 3S 20A.………….................................... 26

Figure III.9 DC Connector 2.1x5.5 mm ………………………………..………………… 27

Figure III.10 Boost converter LM2596S and MP2307 ……………………………………… 28

Figure III.11 18650 Lithium Battery…………………………...…………………………... 29

Figure III.12 Joystick ……………………………………………………….……….…...... 30

Figure III.13 Toggle Switch…………………………..…………………………………… 30

Figure III.14 Oled screen 0.96” ……………………………………………………….......... 31

Figure III.15 NRF24|01 + PA ……………………………..……………………………… 32

Figure III.16 NRF24|01……………………………..……………………………..……........ 33

Figure III.17 Push Button 6x6x5 mm ………………………………………………………… 33

Figure III.18 Resistors……………………………..…………………….…………………... 34

Figure III.19 AMS1117 3.3V Regulator ……………………...………………………………… 35

Figure.III.20 Prototype PCB…………………………………………………….…………. 36

Figure III.21 Aluminum electrolytic capacitor ……………………………………………........ 36

Figure III.22 MPU6050 Gyroscope ………………………………………………………….. 37

Figure III.23 DC 3v 6v Mini Micro Submersible Water Pump…………………………………. 38

Figure III.24 two channel relay module……………………………………………………….. 39

Figure III.25 Arduino IDE software…………………………………………...……………... 40

Figure III.26 Fritzing software………………………………………………………………... 41

Figure III.27 Multiwii software…………………………………………….…..………….. 42

Figure III.28 Monitor space of Multiwii software………………………………………………. 43

Figure III.29 Control space of Multiwii software………………………………………..…… 43

Figure IV.1 Transmitter schematic…………………………………………………………………… 45

Figure IV.2 Transmitter……………………………………………………………….…………. 46

Figure IV.3 Receiver schematic………………………………………………………………… 46

Figure IV.4 Receiver…………………………………………………….……………………… 47

Figure IV.5 Flight Controller schematic………………………………………………………… 47

Figure IV.6 Flight Controller ……………………………………………………………………… 48

Figure IV.7 Quadcopter schematic………………………………………………………………… 49

Figure IV.8 Quadcopter……………………………………………………..………………… 50

Figure IV.9 Quadcopter…………………………………………………….………………… 51

LIST OF TABLES

LIST OF TABLES

Tableau II.1 Variables **9**

Tableau III.1 Quadcopter Components **18**

Tableau III.2 Transmitter Components. **22**

Tableau III.3 Different features of the Arduino family. **24**

Tableau III.4 Receiver Components. **34**

Tableau III.5 Flight Controller Components **37**

Tableau III.6 Spraying System Components **38**

GENERAL INTRODUCTION

Research and development of unmanned aerial vehicle (UAV) and micro aerial vehicle (MAV) are getting high encouragement nowadays since the application of UAV and MAV can apply to a variety of areas such as rescue mission, military, film making, agriculture, and others. In U.S. Coast Guard maritime search and rescue mission, UAV that attached with infrared cameras assist the mission to search the target [1].

Quadcopter or quadrotor aircraft is one of the UAVs that are major focuses of active researches in recent years. Compare to a terrestrial mobile robot that often possible to limit the model to kinematics, Quadcopter required dynamics to account for gravity effect and aerodynamic forces [2].

Quadcopter operated by a thrust that produces by four motors that are attached to their body. It has four input forces and six output states (x, y, z, θ, ψ, ω) and it is an under-actuated system since this enables Quadcopter to carry more load [3].

In our research we focus on new type of structure based on the unmanned aerial vehicle which can be put into a control loop for agricultural purposes to spray chemicals over crops.

Our aim in this project is to focus on solving several problems in agriculture, of course by using our knowledge in electronics engineering. Our first motivation is creating a project could help farmers spray their chemical products on huge field saving time and workers effort by using agricultural drone. Our 2nd motivation is to reduce the cost of this the agricultural drone by building our own drone, transmitter, receiver, flight controller, and spraying system.

CHAPTRE I

LITERATURE REVIEW

I.1. Introduction:

This chapter illustrates a general overview of the scope and the aim behind writing this thesis. A brief quadcopter background studies, problem statement, motivation, and objectives are introduced and then the layout of the thesis is presented.

I.2. Definition:

1. Quadcopter:

A quadcopter also known as a quadrotor is a multi-rotor unmanned aerial vehicle (UAV). Quadcopters fall in the category of vertical take-off and landing (VTOL) UAVs. A quadcopter has four rotors in square formation at the equal distance from the center of mass of the vehicle. The speed of the rotors is manipulated to perform different maneuvers, hovering, take-off and landing.

Before GPS and the internet, drones were only available for military use, but with continuous development in UAV technology and exceptional growth rate in the previous ten years, drones have become very popular among the civil sector. With the growth in popularity drones market was valued at 18.14 billion USD and it is expected to reach 52.30 billion USD by 2025

The advancement and introduction of impressive technology in drones has developed new fields of applications for it. Today drones are being used in several areas for various purposes. Stated below are some of the common areas of drones applications.

- Aerial photography
- Search and rescue
- Agriculture
- Shipping and delivery
- Engineering applications
- 3D mapping
- Research and science
- Aerial surveillance
- Mineral’s exploration
- Military use, etc.

2. Agricultural Quadcopter:

An agricultural Quadcopter is an UAVs used to help optimize and facilitate agriculture operations, increase crop production, and monitor crop growth. Sensors and digital imaging capabilities can give farmers a richer picture of their fields. Using a drone for agriculture and collecting information from it can be useful in improving crop yields and farm efficiency.

The aerial view provided by a drone can reveal many issues such as irrigation problems, soil variation, and pest and fungal infestations. Multispectral images show a near-infrared view as well as a visual spectrum view. The combination shows the farmer the differences between healthy and unhealthy plants, a difference not always clearly visible to the human eye. Thus, these views can assist in assessing crop growth and production. Crops can be surveyed at any time using agricultural drones, allowing for rapid identification of problems.

Figure I.1. Agricultural Drone [W1]

I.3. Historical Background and Trends in Agricultural Aviation:

Multiple unreferenced sources state that the first known aerial application of agricultural materials was in 1906 by John Clervaux Chaytor, who spread seeds over a swamped valley floor in Wairau (New Zealand) on the family farm “marshlands” by using a hot air balloon with mobile tethers. Soon after, in 1921, a plane was used in crop dusting by the US Agriculture Department and the US Army Signal Corps research station in Ohio [4].

The advantages of using aerial services in agriculture favored the extension to other tasks such as top dressing, the application of fertilizers over farmland from the air in 1940.

Nevertheless, physical treatments were not the unique use of aerial vehicles in agriculture. Thus, there are works documented in 1930 and applications since 1950s on the use of infrared aerial photography to detect loss of vigor in wheat and other small grains due to several diseases by using planes [5].

Unlike commercial services that used fix wing solutions for crop dusting, Yamaha developed probably the first Unmanned Aerial Vehicle (UAV) applied to agriculture in 1997 by using a rotary wing aircraft **(Figure I.2**). Using helicopters showed big advantages in field spraying due to their high maneuverability, reduced speed and velocity and the positive impact of the airflow from the rotor in spraying tasks. This was possible thanks to the applications of novel fuzzy control techniques by Professor M. Sugeno for controlling unstable systems such as helicopters [6].

Figure I.2. Yamaha RMAX [W2]

I.4. Background Studies:

Quadcopter as a testing platform for different control strategies attracts people related to control engineering. A lot of research is happening on quadcopters to achieve desired maneuverability and hovering while using various control strategies such as Linear Fuzzy Logic, Quadratic Regulator (LQR) control, Adaptive control, Predictive control, Robust control, etc.

In Munich Technical University Computer Vision Group is researching an autonomous quadcopter. The autonomous quadcopter will be able to, localize 7 and navigate in the 3D environment. They are interested in using RGBD cameras and monocular stereo as the main sensor

A team of researchers is working on the Fast Lightweight Autonomy program in DARPA. The purpose of the researchers is to enable quadcopters to fly through obstacles and buildings at a speed of 20 m/s. The drone would be able to use an onboard camera and sensor as eyes to see around. An advanced algorithm will be developed which will use data from camera and sensors to decide without human interference.

In Middle East Technical University altitude control was developed by using Linear Quadratic Regulator “LQR”

I.5. Scope of Thesis:

The objective of the project is to utilize the existing material to understand the dynamic equations and behavior of quadcopters. Depending on the dynamic equations of the quadcopter a Proportional, Integral, and Derivative (PID) based MATLAB/Simulink control system will be designed and implemented to achieve control of the quadcopter. The designed controller will be able to control the attitude of the vehicle (Roll, Pitch, and Yaw). This paper will explain the PID controller tuning process and integration of the designed controller with real hardware in detail. The project is primarily focused on the PID controller, other control strategies are not explained in this project.

Hardware board, Sensors, Electrical and Mechanical components will be selected, and their general working and compatibility to each other will be discussed for implementation of the controller on the actual system. However, the components manufacturing process, materials, and detailed work are not concerned with the purpose of the project.

I.6. Conclusion:

Drones have already vastly altered the agricultural industry and will continue to grow in the coming years. While drone use is becoming more useful to small farmers, there is still a way to go before they become part of every farmer’s equipment roster and there still several challenges and hurdles in these early stages of research and development with the Artificial intelligence domain. [7]

CHAPTRE II

VEHICLE DYNAMIC

II.1. Introduction :

This chapter introduces a theoretical background of the quadcopter concept, physics of flight, structure, and design configurations. Then numerous papers and theses dealing with the problem of modeling, controlling, and implementing the quadcopter are discussed briefly. Finally, the quadcopter mathematical model is shown with the details of reference frames, quadcopter kinematics, dynamics, and aerodynamic effects.

II.2. Vehicle Dynamic :

It is very important to have the concept of 6 degrees of freedom "6DOF" to describe the motion of the quadcopter in terms of equations. The 6DOF concept gives the position and the orientation of the vehicle in 3 dimensions (3D). The vehicle's dynamic equations simply represent the change in orientation and position over time.

We assume that in our case the earth is flat, the force of gravity is constant, the center of mass is equal to the center of gravity, and the quadcopter structure is rigid.

1. Six-Degrees of Freedom (6DOF):

The position of the vehicle can be described easily by three coordinates in space, but to achieve control on the vehicle we must be aware of the orientation of the vehicle in space as well. To describe the dynamics (Position vs time and Orientation vs time) in space we need to describe the position of all points on the vehicles body, which can be done with six Coordinates (Two Frames of References), originating the concept of six-degrees of freedom (6DOF) (**Figure II.1**).

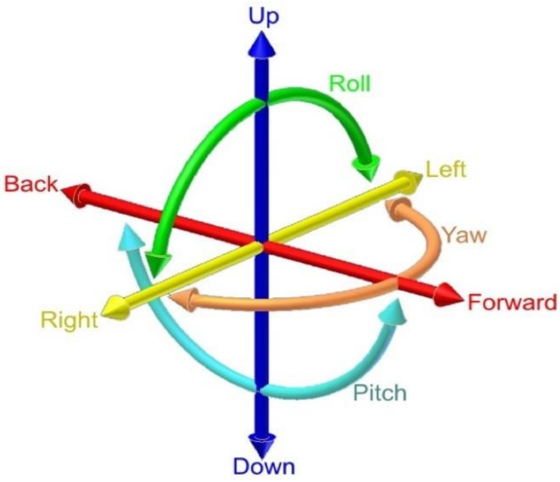


Figure II.1. Six-Degrees of Freedom [W3]

2. Orientation and Frames of References:

The position of the vehicle can be described by x, y, and z coordinates, giving the respective distance from an origin fixed to the earth known as the inertial frame of reference, and coordinates are usually in the cardinal directions North, East, and Down. To represent the orientation or attitude of the vehicle φ, θ, and ψ angles are used with respect to the body frame reference which is a coordinate system with its origin at the body center of gravity (Cg). **Figure II.2** is the representation of the frames of reference for visualization and better understanding.

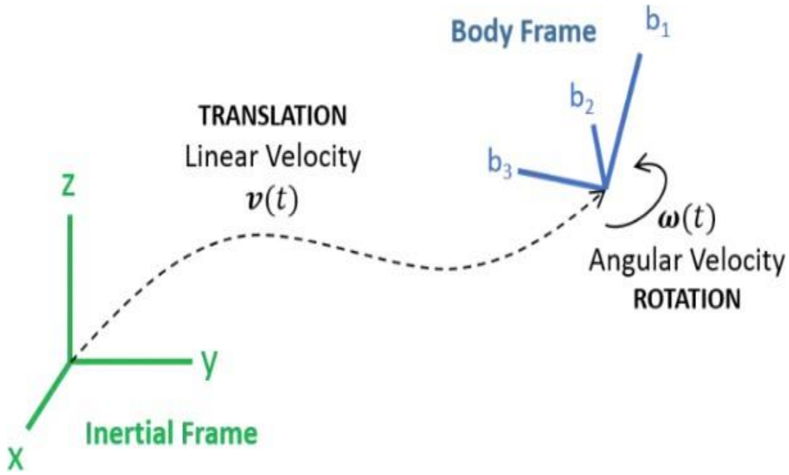


Figure II.2. Inertial and Body Frames of References [W4]

3. Euler Angels:

For this project, we are mainly focusing on the attitude control of the quadcopter. There are some ways to get the orientation of the vehicle, for example, Trigonometric functions, Euler angles, and Quatrains. For the sake of simplicity, we will use Euler angles even though there are some singularity issues. By Euler angles, we can find the final orientation of the vehicle concerning the body frame by using an inertial to body frame transformation matrix. It is convenient to use matrix multiplication for vector transformation. Let us say (x, y, and z) is an inertial frame of reference, and (b1, b2, and b3) is a body frame of reference. Rotation around one of the body frame axis results in the displacement of the other two body frame axes concerning inertial frame axes.

At the same time, the rotation axis remains parallel to the corresponding inertial axis. Rotated body frame axes can be represented as inertial trigonometric function equations. Trigonometric equations can be later expressed in matrix forms. Similarly, we can get two other matrices by rotating the other two body frame axes. Figure 3 shows the rotation of all three-body frame axes b1, b2, and b3 concerning corresponding inertial frame axes, inertial trigonometric function equations, and their matrix representations

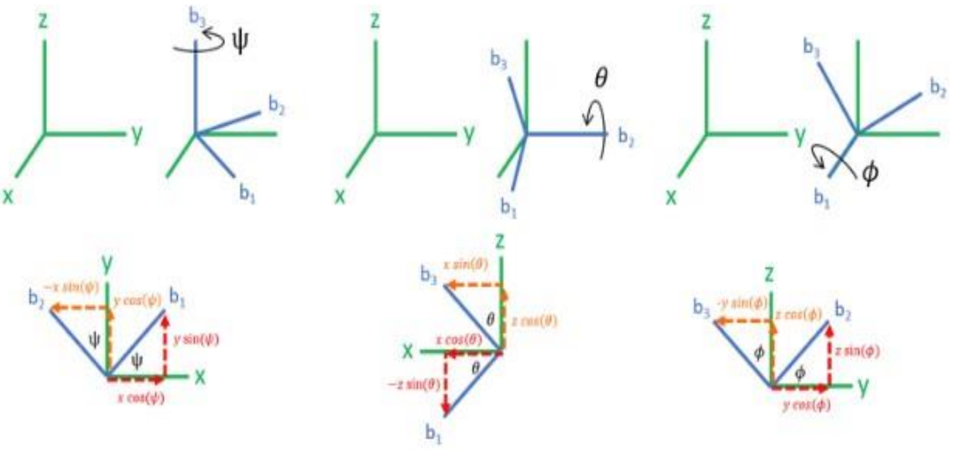


Figure II.3. Euler Angles [W4]

$$\boldsymbol{b_1 = x\cos(\Psi) + y\sin(\Psi)} \qquad \boldsymbol{b_1 = x\cos(\theta) - z\sin(\theta)} \qquad \boldsymbol{b_1 = x}$$

$$\boldsymbol{b_2 = -x\sin(\Psi) + y\cos(\Psi)} \qquad \boldsymbol{b_2 = y} \qquad \boldsymbol{b_2 = y\cos(\varphi) + z\sin(\varphi)}$$

$$\boldsymbol{b_3 = z} \qquad \boldsymbol{b_3 = x\sin(\theta) + z\cos(\theta)} \qquad \boldsymbol{b_3 = -y\sin(\varphi) + z\cos(\varphi)}$$

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} \cos(\Psi) & \sin(\Psi) & 0 \\ -\sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & \sin(\varphi) \\ 0 & -\sin(\varphi) & \cos(\varphi) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Orderly combination of these three matrices (**Figure II.3**) gives the inertial to body frame transformation matrix. Order of the operations is really important

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & \sin(\varphi) \\ 0 & -\sin(\varphi) & \cos(\varphi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\Psi) & \sin(\Psi) & 0 \\ -\sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (II.1)$$

All three matrices representing the trigonometric equations are multiplied to get a single transformation matrix C_n^b shown in equation 2 which will be used for inertial to body frame transformation.

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\Psi) & \cos(\theta)\sin(\Psi) & -\sin(\theta) \\ \cos(\varphi)\sin(\Psi) + \cos(\Psi)\sin(\theta)\sin(\varphi) & \cos(\Psi)\cos(\varphi) + \sin(\theta)\sin(\varphi)\sin(\Psi) & \cos(\theta)\sin(\varphi) \\ \sin(\Psi)\sin(\varphi) + \cos(\Psi)\cos(\varphi)\sin(\theta) & -\sin(\varphi)\cos(\Psi) + \cos(\varphi)\sin(\theta)\sin(\Psi) & \cos(\theta)\cos(\varphi) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (II.2)$$

For transformation from body to inertial frame theoretically the inverse of the transformation matrix C_b^n is used. As transformation matrices are orthonormal, we can simply use the transpose of the transformation matrix Rt=R−1. Equation 3 shows transpose of the equation 2 matrix

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\Psi) & -\cos(\varphi)\sin(\Psi)+\cos(\Psi)\sin(\theta)\sin(\varphi) & \sin(\Psi)\sin(\varphi)+\cos(\Psi)\cos(\varphi)\sin(\theta) \\ \cos(\theta)\sin(\Psi) & \cos(\Psi)\cos(\varphi)+\sin(\theta)\sin(\varphi)\sin(\Psi) & -\sin(\varphi)\cos(\Psi)+\cos(\varphi)\sin(\theta)\sin(\Psi) \\ -\sin(\theta) & \cos(\theta)\sin(\varphi) & \cos(\theta)\cos(\varphi) \end{bmatrix} \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} \quad (II.3)$$

Now using the equations 2 and 3, we can find the orientation of the vehicle with respect to body frame and inertial frame. [9]

4. Equations of Motion:

4.1. Variables:

Now we can define the position and orientation of the vehicle in space. To get equations of motion of the vehicle we need variables (Table 1) to represent the linear and angular velocity of the quadcopter. Using the right-hand rule and old convention referring X, Y, and Z to North, East, and Down respectively, we can derive the equations of dynamics.

Linear velocity variables	
VR	Linear velocity in body frame
u	Longitudinal velocity
v	Lateral velocity
w	Normal velocity
Rotational velocity variables	
$\mathcal{W}$R	Rotational velocity in body frame
p	Roll rate
q	Pitch rate
r	Yaw rate
Forces	
F_X	Force in X direction
F_Y	Force in Y direction

F_Z	Force in Z direction
Euler angles	
ϕ	Roll angle
θ	Pitch angle
Ψ	Yaw angle
Moments or Torques	
L	Rotational moment along x axis
M	Rotational moment along y axis
N	Rotational moment along Z axis

Table II.1. Variables

4.2.Inertial Motion in Body Frame:

As we have assumed that the vehicle body is rigid, there is no movement in the body parts regarding the body frame. Our sensors and propellers are attached to the body, so we want to derive equations of motion in the body frame coordinates independent of an inertial frame. These equations could be used later with any inertial starting point.

4.3.The Chain Rule:

The chain rule represents the inertial motion in the body frame and simultaneously gives the mathematical representation. We take the derivative of inertial frame vectors to represent them in the body frame. The chain rule of derivation gives the derivative of both the change because of the time derivative of the vector within the coordinate frame, as well as the time derivative of the coordinate frame rotation. Where b_1, b_2, and b_3 in equation **II.4** are unit vectors associated with the body frame to represent the inertial frame velocity in the body frame.

$$\mathbf{v^b} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \end{bmatrix}^{\mathbf{b}} = \mathbf{ub_1} + \mathbf{vb_2} + \mathbf{wb_3} \qquad (\boldsymbol{II.4})$$

Applying the Chain rule on equation 4 we get equation 5.

$$\left(\frac{dV}{dx}\right)_{inertial} = \left(\frac{du}{dt}b_1 + \frac{dv}{dt}b_2 + \frac{dw}{dt}b_3\right) + \left(u\frac{db_1}{dt} + v\frac{db_2}{dt} + w\frac{db_3}{dt}\right) \qquad (II.5)$$

In equation 5 above, the first set of parentheses represent inertial velocity in the body coordinates. The second set represents velocity change due to coordinate frame rotation. Equation **II.5** above can be written as equation **II.6** using the Coriolis Theorem, which is a cross product of angular velocity with velocity vector representing frame rotation.

$$\dot{v}_{inertial} = \dot{v}^b + w_n^b \times v^b \qquad (II.6)$$

Further it can be represented as a skew symmetric matrix as shown in the (equation **II.7**.)

$$w \times v = [w \times]v = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_z & 0 \end{bmatrix} v \qquad (II.7)$$

5. Newton’s 2nd Law of Motion:

So far, we can switch between the inertial and body frame and can get inertial motion in the body frame of reference. To derive linear and rotational equations of motion we use Newton's Second Law of motion summarized as force (F) equals the change in momentum (P) or mass (m) times acceleration (a) or derivative of velocity (v), represented in equation **II.8**.

$$F = \frac{dP}{dt} = m\frac{dv}{dt} = ma \qquad (II.8)$$

The equation **II.8** giving linear momentum, can be used to get angular momentum (Equation **II.9**) by multiplying with a position vector ($\vec{r}$)

$$H = \vec{r} \times F = \vec{r} \times \ m\frac{dv}{dt} \qquad (II.9)$$

Further simplified as the moment (M) equals the change in angular momentum (H) which can be replaced with the product of moment of inertia (I) for continuously mass distributed objects and angular acceleration (Ω), where *Ω* is the derivative of angular velocity (ω). Equation **II.9** becomes as follows (Equation **II.10**).

$$M = \frac{dH}{dt} = I\frac{d\omega}{dt} = I\Omega \qquad (II.10)$$

We will derive linear and rotational equations separately.

5.1.External Forces and Moments:

For translation equations of motion, we need to know the external forces and moments acting on the body of the vehicle

5.2.Thrust:

The thrust is produced by the blades of the quad-copter acts perpendicular to the vehicle and moments acts at the center of gravity of the vehicle. **Figure II.4** shows the distance of propellers from the center of gravity (CG) of the vehicle. These distances are later used to calculate the moment around the center of gravity of the vehicle.

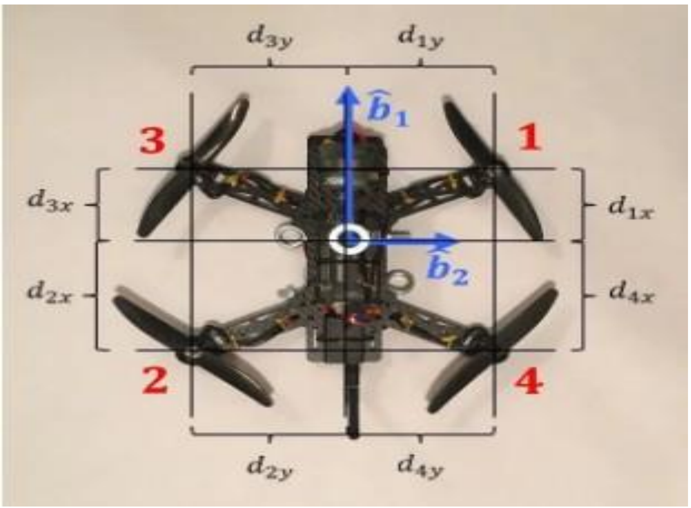


Figure II.4. Propellers Distance from center of gravity (CG) [W5]

Following equation **II.11** represent the thrust equation of the vehicle.

$$\begin{bmatrix} \mathbf{F_x} \\ \mathbf{F_y} \\ \mathbf{F_z} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{-F1 - F2 - F3 - F4} \end{bmatrix} \qquad (\boldsymbol{II.11})$$

Moments are simple product of forces and distance from the center of gravity around all three axes of the body frame.

Equation **II.12** shows moment along X axis of the vehicle.

$$\mathbf{L = F_1 d_{1y} - F_2 d_{2y} - F_3 d_{3y} + F_4 d_{4y}} \qquad (\boldsymbol{II.12})$$

Equation **II.13** shows moment along Z axis of the vehicle.

$$\mathbf{M = -F_1 d_{1x} + F_2 d_{2x} - F_3 d_{3x} + F_4 d_{4x}} \qquad (\boldsymbol{II.13})$$

Yaw is produced in quadcopter by the rotation of the propellers. Two propellers rotate clockwise and the other two rotate counterclockwise to balance the yaw moment (**Figure II.5**). Equation **II.14** shows how to calculate the yaw moment of the quadcopter.

$$\mathbf{N = -T(F_1 d_{1x} d_{1x}) - T(F_2 d_{2x} d_{2x}) + T(F_3 d_{3x} d_{3x}) + T(F_4 d_{4x} d_{4x})} \qquad (\boldsymbol{II.14})$$

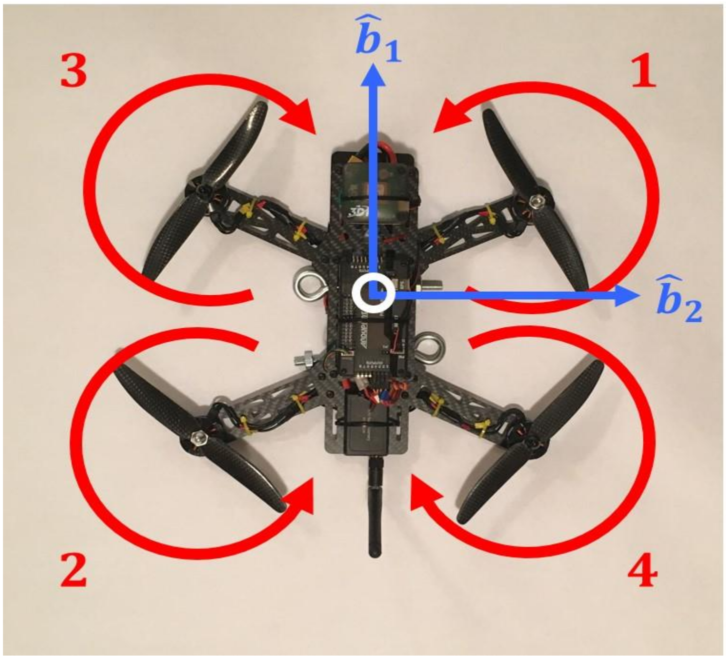


Figure II.5. Motors rotation direction for yaw moment [W5]

5.3.Gravity in Body Frame:

The force of gravity acting on the vehicle can be represented in the body frame using Euler angles. The transformation matrix C_n^b from equation **II.2** can be used for this purpose. Where C_n^b is an inertial to body frame transformation matrix, F_g^n and C_g^b represents the force of gravity in inertial and body frames of references respectively. Equation **II.15** shows the force of gravity in an inertial frame of reference.

$$\mathbf{F_g^n} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{mg} \end{bmatrix} \qquad (\boldsymbol{II.15})$$

Equation **II.15** is multiplied with inertial to body frame transformation matrix C_n^b to get the force of gravity in the body frame of reference (Equation **II.16**).

$$\mathbf{F_g^b = C_n^b F_g^n}$$

$$= \begin{bmatrix} \cos(\theta)\cos(\Psi) & \cos(\theta)\sin(\Psi) & -\sin(\theta) \\ -\cos(\varphi)\sin(\Psi)+\cos(\Psi)\sin(\theta)\sin(\varphi) & \cos(\Psi)\cos(\varphi)+\sin(\theta)\sin(\varphi)\sin(\Psi) & \cos(\theta)\sin(\varphi) \\ \sin(\Psi)\sin(\varphi)+\cos(\Psi)\cos(\varphi)\sin(\theta) & -\sin(\varphi)\cos(\Psi)+\cos(\varphi)\sin(\theta)\sin(\Psi) & \cos(\theta)\cos(\varphi) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (II.16)$$

After multiplication, equation **II.16** simplifies as equation **II.17**.

$$\mathbf{F_g^b} = \begin{bmatrix} -mg\sin(\theta) \\ mg\sin(\varphi)\cos(\theta) \\ mg\cos(mg\cos(\varphi)\cos(\theta)) \end{bmatrix} \qquad (II.17)$$

5.4.Moments of Inertia:

The moment of inertia gives the amount of moment needed to rotate a still object and the moment needed to stop a rotating object. We need a moment around all three axes of the vehicle, so we will present it in matrix form (Equation **II.18**). The moment of inertia is the square of the distance from the center of mass of the body.

$$\mathbf{I} = \begin{bmatrix} I_{XX} & -I_{YX} & -I_{XZ} \\ -I_{XY} & I_{YY} & -I_{YZ} \\ -I_{XZ} & -I_{YZ} & I_{ZZ} \end{bmatrix} \qquad (II.18)$$

For a symmetrical body the moment of inertia on opposite sides of the vehicle cancels each other. We have assumed our vehicle is symmetrical, so our inertia matrix will be simplified as follows (Equation **II.19**).

$$\mathbf{I} = \begin{bmatrix} I_{XX} & 0 & 0 \\ 0 & I_{YY} & 0 \\ 0 & 0 & I_{ZZ} \end{bmatrix} \qquad (II.19)$$

5.5.Linear Acceleration and Motion:

Using the Coriolis theorem Equation **II.6**, we can convert inertial acceleration into a rotating body frame. Equation **II.20** shows the inertial acceleration conversion into body frame acceleration.

$$\dot{V}_b = \overset{\cdot}{\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix}}^b + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}^b = \begin{bmatrix} \dot{u} & qw & -rv \\ \dot{v} & ru & -pw \\ \dot{w} & pv & -qu \end{bmatrix} \qquad (II.20)$$

By putting forces and acceleration in Newton’s 2nd law F=ma we can get linear motion (Equations **II.21**).

$$\begin{bmatrix} -\mathbf{mg\,sin\,(\theta)} \\ \mathbf{mg\,sin(\varphi)\,cos\,(\theta)} \\ -\mathbf{F_1}-\mathbf{F_2}-\mathbf{F_3}-\mathbf{F_4mg\,cos\,(mg\,cos(\varphi)\,cos(\theta))} \end{bmatrix} = \mathbf{m}\begin{bmatrix} \dot{\mathbf{u}} & \mathbf{qw} & -\mathbf{rv} \\ \dot{\mathbf{v}} & \mathbf{ru} & -\mathbf{pw} \\ \dot{\mathbf{w}} & \mathbf{pv} & -\mathbf{qu} \end{bmatrix} \qquad (II.21)$$

5.6.Rotational Acceleration:

Using Coriolis Theorem (Equation **II.6**) again as above in equation **II.20**, we can get rotational acceleration (Equation **II.22**), the difference is that here we use the angular velocity instead of linear velocity.

$$\begin{bmatrix} \mathbf{L} \\ \mathbf{M} \\ \mathbf{N} \end{bmatrix} = \begin{bmatrix} \mathbf{I_{XX}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I_{YY}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I_{ZZ}} \end{bmatrix}\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{q}} \\ \dot{\mathbf{q}} \end{bmatrix}^{\mathbf{b}} + \begin{bmatrix} \mathbf{0} & -\mathbf{r} & \mathbf{q} \\ \mathbf{r} & \mathbf{0} & -\mathbf{p} \\ -\mathbf{q} & \mathbf{p} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{I_{XX}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I_{YY}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I_{ZZ}} \end{bmatrix}\begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \mathbf{r} \end{bmatrix} \qquad (II.22)$$

After addition and multiplication, equation **II.22**simplifies as equation **II.23**.

$$\begin{bmatrix} \mathbf{L} \\ \mathbf{M} \\ \mathbf{N} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{p}}\mathbf{I_{XX}} \\ \dot{\mathbf{q}}\mathbf{I_{YY}} \\ \dot{\mathbf{q}}\mathbf{I_{ZZ}} \end{bmatrix} + \begin{bmatrix} -\mathbf{I_{YY}qr} + \mathbf{I_{ZZ}qr} \\ \mathbf{I_{XX}pr} - \mathbf{I_{ZZ}pr} \\ -\mathbf{I_{XX}pq} + \mathbf{I_{YY}pq} \end{bmatrix} \qquad (II.23)$$

5.7.Angular Velocity and Euler Angles:

We know, how the angular velocities p, q, and r are changing over time and a gyro will give us these values, but these are not the same as Euler angles ϕ, θ and Ψ. Angular velocity is the rate of change of angles with the body axis, while Euler angles are rotation in their frame of reference. Using the coordinate transformation, angular velocity can be represented as the Euler angle derivative (Equation **II.24**).

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \mathbf{r} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{cos\,(\varphi)} & \mathbf{sin\,(\varphi)} \\ \mathbf{0} & \mathbf{sin\,(\varphi)} & \mathbf{cos\,(\varphi)} \end{bmatrix}\begin{bmatrix} \mathbf{cos\,(\theta)} & \mathbf{0} & -\mathbf{sin\,(\theta)} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{sin\,(\theta)} & \mathbf{0} & \mathbf{cos\,(\theta)} \end{bmatrix}\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \dot{\Psi} \end{bmatrix} + \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{cos\,(\varphi)} & \mathbf{sin\,(\varphi)} \\ \mathbf{0} & -\mathbf{sin\,(\varphi)} & \mathbf{cos\,(\varphi)} \end{bmatrix}\begin{bmatrix} \mathbf{0} \\ \dot{\theta} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \dot{\varphi} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \; (II.24)$$

Equation **II.24**simplifies as Equation **II.25**.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} -\dot{\Psi} sin\,(\theta) \\ \dot{\Psi}\,cos(\theta)\,sin\,(\varphi) \\ \dot{\Psi}\,cos(\varphi)\,cos\,(\theta) \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} cos\,(\varphi) \\ \dot{\theta} sin\,(\varphi) \end{bmatrix} + \begin{bmatrix} \dot{\varphi} \\ 0 \\ 0 \end{bmatrix} \qquad (II.25)$$

Finally, the equations look as follows

$$p = \dot{\varphi} - \dot{\Psi} si\,n(\theta) \qquad (II.26)$$

$$q = \dot{\theta}\,cos(\varphi) + \dot{\Psi}\,cos(\theta)\,si\,n(\varphi) \qquad (II.27)$$

$$r = \dot{\Psi}\,cos(\varphi)\,cos(\theta) + \dot{\theta} si\,n(\varphi) \qquad (II.28)$$

5.8.Inertial Coordinate:

To calculate the position of the vehicle in an inertial frame, we simply use the Euler angle transformation matrix C_b^n (Equation **II.3**) to convert body frame velocities to an inertial coordinate position (Equation **II.29**). Where $\dot{x}^E$, $\dot{y}^E$, and $\dot{z}^E$represents the first derivative of X, Y, and Z positions of the vehicle in an inertial frame.

$$\begin{bmatrix} \dot{x}^E \\ \dot{y}^E \\ -\dot{z}^E \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\Psi) & -\cos(\varphi)\sin(\Psi)+\cos(\Psi)\sin(\theta)\sin(\varphi) & \sin(\Psi)\sin(\varphi)+\cos(\Psi)\cos(\varphi)\sin(\theta) \\ \cos(\theta)\sin(\Psi) & \cos(\Psi)\cos(\varphi)+\sin(\theta)\sin(\varphi)\sin(\Psi) & -\sin(\varphi)\cos(\Psi)+\cos(\varphi)\sin(\theta)\sin(\Psi) \\ -\sin(\theta) & \cos(\theta)\sin(\varphi) & \cos(\theta)\cos(\varphi) \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}^b \quad (II.29)$$

5.9.Non Linear Equations :

By simple mathematical operations on the matrices above, we can rewrite the equations in non-linear form and can modify them as we want. Equations are represented below.

$$\dot{u} = -mg\sin(\theta) + rv - qw \qquad (II.30)$$

$$\dot{v} = -mg\sin(\varphi)\cos(\theta) - ru + qw \qquad (II.31)$$

$$\dot{w} = \frac{1}{m}(-F_Z) + mg\cos(\varphi)\cos(\theta) - qu - pv \qquad (II.32)$$

$$\dot{p} = \frac{1}{I_{xx}}(L + (I_{YY} - I_{ZZ})qr) \qquad (II.33)$$

$$\dot{q} = \frac{1}{I_{YY}}(M + (I_{ZZ} - I_{XX})pr) \qquad (II.34)$$

$$\dot{r} = \frac{1}{I_{zz}}(N + (I_{XX} - I_{YY})pq) \qquad (II.35)$$

$$\dot{\varphi} = \left(p + \left(q\sin(\varphi) - r\cos(\varphi)\right)\tan(\theta)\right) \qquad (II.36)$$

$$\dot{\theta} = q\cos(\varphi) - r\sin(\varphi) \qquad (II.37)$$

$$\dot{\Psi} = \left(q\sin(\varphi) - r\cos(\varphi)\right)\sec\theta \qquad (II.38)$$

$$\dot{x}^E = \left(q\sin(\varphi) - r\cos(\varphi)\right)\sec\theta \qquad (II.39)$$

Non-Linear equations represent the dynamic behavior of the quadcopter, but we still need to represent the equations including motors rpm, propellers thrust, the mass of the quadcopter, torque functions, and inertial variables values. These calculations and variable values can be found in Appendix 1(A1, A2, A3, A4, A5, and A6).

II.3. Conclusion :

On this chapter we studied the mathematical modelling and control of a quadcopter. The mathematical model of quadcopter dynamics was presented and the differential equations were derived from the Newton-Euler and the Euler-Lagrange equations.

CHAPTRE III
COMPONENTS

III.1. Introduction :

This chapter presents the hardware implementation of the quadcopter which provides a means of verifying the mathematical model and controller designed earlier in the previous chapter. Not all of the simulated scenarios were implemented on real hardware due to both the lack of time and some sensor problems encountered.

III.2. Components (Hardware) :

1. Quadcopter Components :

The components (**Table III.1**) are selected considering the performance and compatibility with other selected components. The most suitable components are selected online considering the application and budget of the project to build the actual model.

Frame	Wood Quadcopter Frame (1)
Propellers/Rotors	9x4.5 Propellers (4)
Motors	A2212/13t 1000kv motors (4)
ESCs	30A ESCs (4)
Battery	3S LiPo Battery (1)
Transmitter	Homemade Transmitter (1)
Receiver	Homemade 6S Channel Receiver (1)
Flight Controller	Homemade Flight Controller (1)

Tables III.1: Quadcopter Components.

1.1.Frame (Quadcopter Body):

The frame will be the physical support for the drone and to which all the other elements will be attached. Since there are numerous possibilities, in an attempt to narrow the quest, the choice will mainly depend on its size and material because they are the main parameters that drive the drone’s weight and more specifically its performance.

1.2.Propellers (Rotors) :

A propeller converts rotary motion from our engine (brushless motor) into a swirling slipstream which pushes the propeller forwards or backwards

Figure III.1 Propellers

1.2.1. Features [W6]:

- Material: ABS
- Length: 15.3 cm/6.0"
- Hub ID dia: 5mm
- Hub OD dia: 13.5mm
- Hub thickness: 6mm
- Color: black

1.3.Brushless Motors :

Brushless DC Motors (aka BLDC motors) are synchronous motors powered by DC electricity via an inverter or switching power supply that produces an AC or bi-directional electric current to drive each phase of the motor via a closed-loop controller. In this context, alternating current does not imply but does include a sinusoidal waveform, with minimal restriction on waveform; it must be periodic, and its frequency will determine motor rpm, and the waveform does affect how smooth the generated torque is as well as the motor's efficiency at transforming electrical to mechanical energy.[8]

Figure III.2 Brushless Motor A2212/13T

1.3.1. Features [W7]:

- KV: 1000
- Max Efficiency: 80%
- Max Efficiency Current: 4-10A (>75%)
- Current Capacity: 12A/60s
- No Load Current @ 10V: 0.5A
- No. Of Cells: 2-3 Li-Poly
- Motor Dimensions: Φ27.5 x 30mm
- Shaft Diameter: Φ3.17mm
- Weight: 47g

1.4.Electronic Speed Controller (ESC):

An electronic speed controller or ESC is a device installed to a remote-controlled electrical model to vary the motor's speed and direction.

We have used 30A electronic speed controllers to control each brushless motors in this experiment which can constantly supply required current to drive brushless motors. It has following specifications

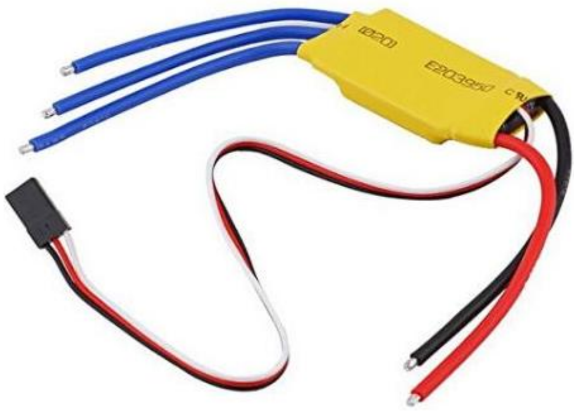


Figure III.3 Electronic Speed Controller (ESC) 20A

1.4.1. Features [W8]:

- Weight: 25g
- Dimensions:45 x 24 x 11mm
- Firmware:
- Power input: 5.6V - 16.8V (2-3 cells Li-Poly, OR 5-12 cells Ni-MH Ni-MH / Ni-Cd battery)
- BEC: 2A
- Constant current: 30A (Max 40A less than 10 seconds)

1.5.**Battery**:

As the brushless motor we have used in this experiment needs high amount of current so we have used 3300mAh 11.1V 3 cell Li-Po (Lithium Polymer) battery. It can provide approximately 3A current constantly.

Figure III.4 ZDF LiPo Battery 3300Mah

1.5.1. Features [W9]:

- Brand: ZDF
- Capacity: 3300mAh
- Continuous discharge rate: 30C
- Burst Rate: 60C
- Voltage per cell: 3.7V
- Max voltage per cell: 4.2V
- Max pack voltage: 12.6V
- Cells: 3S
- Suggest charge rate: 1C
- Silicone wire: 14awg

1.6.Transmitter:

Component	Features
Battery Management System (BMS)	3S BMS 20A
DC Connector	12V 3A 5.5 x 2.1mm
Buck/Boost Converter	LM2596S and MP2307
Battery	18650 Li-ion Lithium Battery
Joystick	JH-D202X-R2 / R4 5K 2D
toggle switch	Mini MTS-102 3-Pin
i2c OLED screen	OLED 128X64 I2C SSD1306
Radio module	NRF24+PA
push button	6mm X 6mm X 5mm Push button
RESISTORS	1K, 2K, 2.2K, 3.3K, 4.7K, 5.1K, 6.8K, 10K, 20K
Arduino	Arduino Nano

Tables III.2: Transmitter Components.

1.6.1. Arduino:

1.6.1.1.Definition:

Arduino board is an open-source platform used to make electronics projects. It consists of both a microcontroller and a part of the software or Integrated Development Environment (IDE) that runs on your PC, used to write & upload computer code to the physical board. The platform of an Arduino has become very famous with designers or students just starting out with electronics, and for an excellent cause. [W10]

1.6.1.2.Application:

The Arduino allows us to make much project, the applications are in almost any and every field out there, ranging from biology and chemistry to manufacturing and advertising, some examples:

- Traffic Light Count Down Timer
- Parking Lot Counter
- Weighing Machines
- Medical Instrument
- Emergency Light for Railways

1.6.1.3. Different Types of Arduino Boards:

The figure. II.1 presents different types of Arduino boards, there are different in size, shape and possibilities of connection

Figure III.5 Different types of Arduino boards [9]

We will cite the different features of the Arduino family, from these we have chosen the four most big boards (UNO, MEGA, Leonardo, Due).

Arduino Board	Processor	Memory	Digital I/O	Analogue I/O
Arduino Uno	16Mhz ATmega328	2KB SRAM, 32KB flash	14	6 input, 0 output
Arduino Mega	16MHz ATmega2560	8KB SRAM, 256KB flash	54	16 input, 0 output
Arduino Due	84MHz AT91SAM3X8E	96KB SRAM, 512KB flash	54	12 input, 2 output
Arduino Leonardo	16MHz ATmega32u4	2.5KB SRAM, 32KB flash	20	12 input, 0 output

Tables III.3: Different features of the Arduino family [W11].

1.6.1.4.Presentation of the Arduino UNO board:

The Uno is a huge option for your initial Arduino. This Arduino board depends on an ATmega328P based microcontroller. As compared with other types of Arduino boards, it is very simple to use like the Arduino Mega type board. It consists of 14-digital I/O pins, where

6-pins can be used as PWM (pulse width modulation outputs), 6-analog inputs, a reset button, a power jack, a USB connection, an In-Circuit Serial Programming header (ICSP), etc.

It includes everything required to hold up the microcontroller; simply attach it to a PC with the help of a USB cable and give the supply to get started with an AC-to-DC adapter or battery. [10]

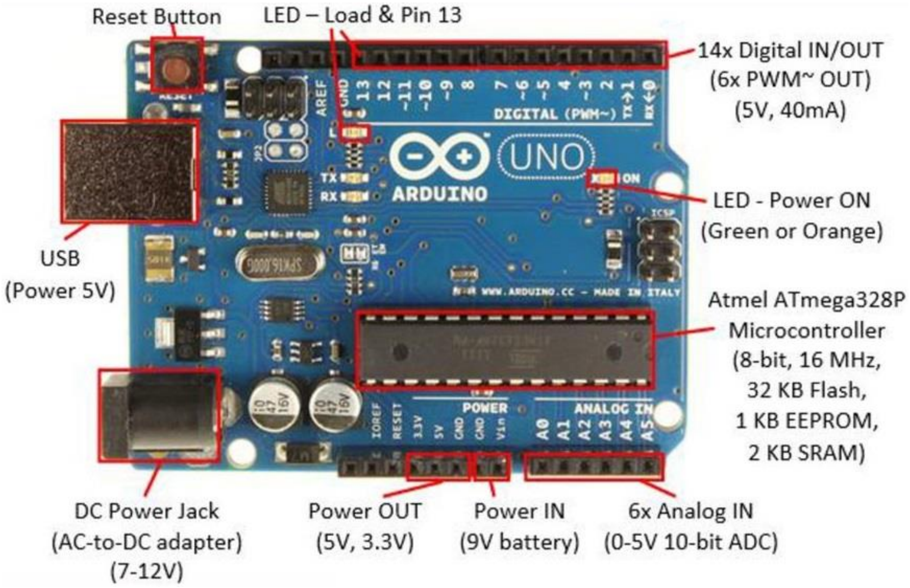


Figure III.6 Presentation of the Arduino UNO board [9]

1.6.1.5.Presentation of the Arduino NANO board:

This is a small board based on the microcontrollers like ATmega328P otherwise ATmega628 but the connection of this board is the same as to the Arduino UNO board. This kind of microcontroller board is very small in size, sustainable, flexible, and reliable.

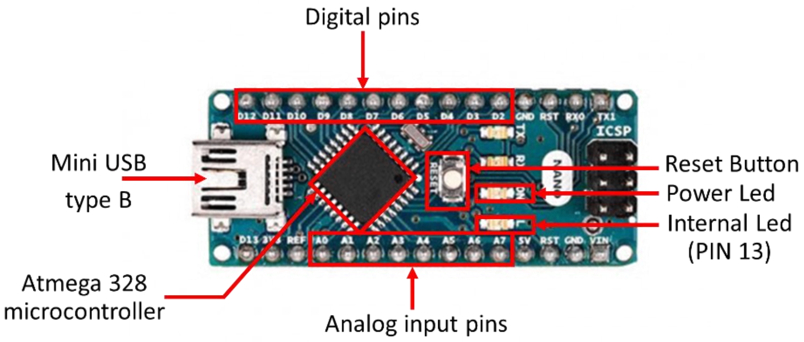


Figure III.7 Presentation of the Arduino NANO board [W12]

We chose the Arduino UNO board and NANO board in our realization because it has several advantages that suit us in our project

- Size and weight of the boards
- The cost of the boards
- Number of pins
- Easy to connect the modules with it
- Easy to passe code in it

1.6.2. Battery Management System (BMS):

A Battery Management System, or BMS for short, is used to protect your battery during charging and discharging.

During charging, the BMS will monitor the voltage of all of your cells and balance the cell groups to ensure they are charged equally.

During discharging, the BMS will monitor the voltage of all of the cells, as well as the entire pack voltage and the discharge current. If the BMS determines that any of its preset limits have been passed, such as the battery draining too low or an unsafe amount of current being pulled from the battery, the BMS will cut power to the battery to protect it from damage. [W13]

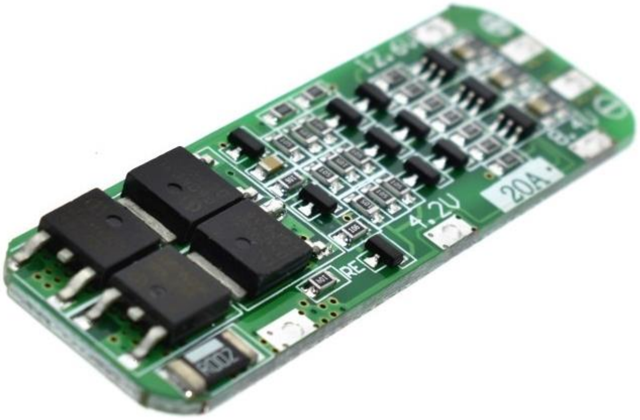


Figure III.8 Battery Management System BMS 3S 20A

1.6.2.1.Features: [W14]

- Number of cells in series: 3
- Maximum rated charge/discharge current: 20 A
- Charging voltage: 12.6 V
- Operating temperature: -4ºF to 122ºF (-20ºC to +50ºC)
- Board dimensions: 59*20*3.4mm (4.3g)

1.6.3. DC Connector:

A DC jack is a component used in many electronic devices that allows a steady power source to be plugged in. Though electronics require direct current (DC) power, alternating current (AC) is the type of electricity supplied to and available in household wall sockets, mainly because of its ability to be delivered over long distances without losing strength. Therefore, with most electronics, an AC adapter connected to a DC jack is necessary to supply power in a usable way. [W15]

Figure III.9 DC Connector 2.1x5.5 mm

1.6.3.1.Features:

- Pin counts: 3
- Withstand voltage: AC 500V
- Color: black
- Inner diameter: 2.1 mm
- Outer diameter: 5.5 mm
- Material: plastic and metal

1.6.4. DC to DC Boost Converter:

A DC-to-DC converter is an electronic circuit or electromechanical device that converts a source of direct current (DC) from one voltage level to another. It is a type of electric power converter. Power levels range from very low (small batteries) to very high (high-voltage power transmission). [W16]

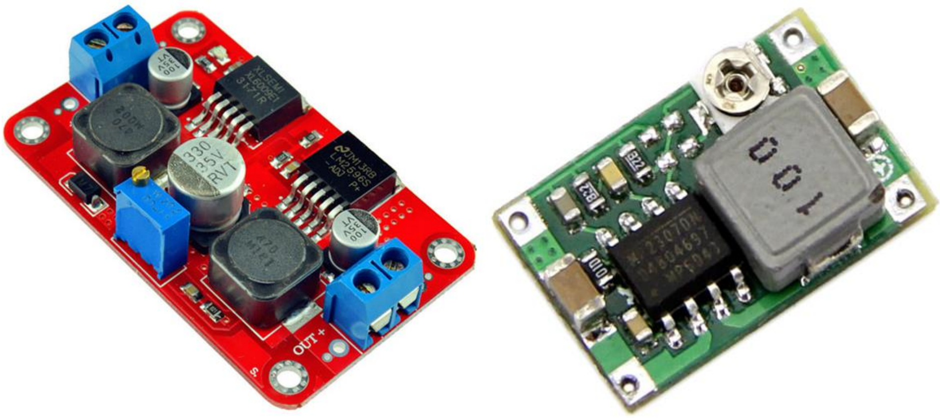


Figure III.10 Boost converter LM2596S and MP2307

1.6.4.1.Features (XL6009 LM2596S):

- Input: 3.5-28V.
- Output: 1.25V-26V.
- Output voltage independent of input voltage (3.5V-28V input, any value output 1.25V-26V).
- Maximum input current 3A.
- Maximum output 3A

1.6.4.2.Features (MP2307):

- Input voltage: DC 4.75V-23V
- Output voltage: DC 1.0V-17V (Adjustable, Output < Input)
- Output current: Rated current 1.8A (3A MAX, cannot be prolonged)
- Conversion efficiency:96%(highest)
- Switching Frequency: 340KHz.
- Output ripple:30mV(no-load)
- Load regulation: ±0.5%
- Voltage regulation: ±2.5%
- Operating Temperature: Industrial (-40℃ ~ +85℃)

1.6.5. 18650 Li-ion Lithium Battery:

18650 batteries are lithium-ion batteries. They get their name from their size: 18mm by 65mm. These batteries not only used in flashlights, but also in: power tools, electric vehicles, vaporizers, cameras, laptops, and more… [w17]

Figure III.11 18650 Lithium Battery

1.6.5.1.Features:

- Size: 6.5 cm
- Voltage: 3.7V
- Capacity: 7800Mah
- Battery life: More than 500 times

1.6.6. Joystick:

A joystick is an input device commonly used to control video games. Joysticks consist of a base and a stick that can be moved in any direction. The stick can be moved slowly or quickly and in different amounts. Some joysticks have sticks that can also be rotated to the left or right. Because of the flexible movements a joystick allows, it can provide much greater control than the keys on a keyboard. [W18]

Figure III.12 Joystick

1.6.6.1.Features: [W19]

- Color: Black
- Resistance Value: Standard 5KΩ
- Resistance Tolerance: ±20%
- Independent Linearity: ±1%
- Temperature Coefficient Resistance: ±400 ppm/°C
- Output Smoothness: 0.5% maximum
- X& Y Axis Electricity Corner: ±25°
- Z Axis Electricity Corner: ±45°

1.6.7. Toggle switch:

A switch that uses a toggle joint with a spring to open or close an electric circuit as an attached lever is pushed through a small arc. [W20]

Figure III.13 Toggle Switch

1.6.7.1.Features: [W21]

- Pin number:3-Pin
- Position: 2 (On-On) 3ON-OFF-ON
- Mounting hole:1/4"(6mm)
- Size:33 x 13 mm

1.6.8. Oled Screen:

The acronym ‘OLED’ stands for Organic Light-Emitting Diode - a technology that uses LEDs in which the light is produced by organic molecules. These organic LEDs are used to create what are considered to be the world’s best display panels.

OLED displays are made by placing a series of organic thin films between two conductors. When an electrical current is applied, a bright light is emitted. A simple design - which brings with it many advantages over other display technologies. [W22]

Figure III.14 Oled screen 0.96”

1.6.8.1.Features:

- Interface type: IIC interface
- Pin definition: GND, VDD, SCK, SDA
- GND (OLED power supply)
- VDD (OLED 3.3~5V)
- SCK (OLED IIC Clock line)
- SDA (OLED IIC Cable)

1.6.9. Radio module NRF24|01:

RF24L01 is a single chip radio transceiver for the world wide 2.4 - 2.5 GHz ISM band. The transceiver consists of a fully integrated frequency synthesizer, a power amplifier, a crystal oscillator, a demodulator, modulator and Enhanced ShockBurst protocol engine. Output power, frequency channels, and protocol setup are easily programmable through a SPI interface. Current consumption is very low, only 9.0mA at an output power of -6dBm and 12.3mA in RX mode. Built-in Power Down and Standby modes makes power saving easily realizable. [W23]

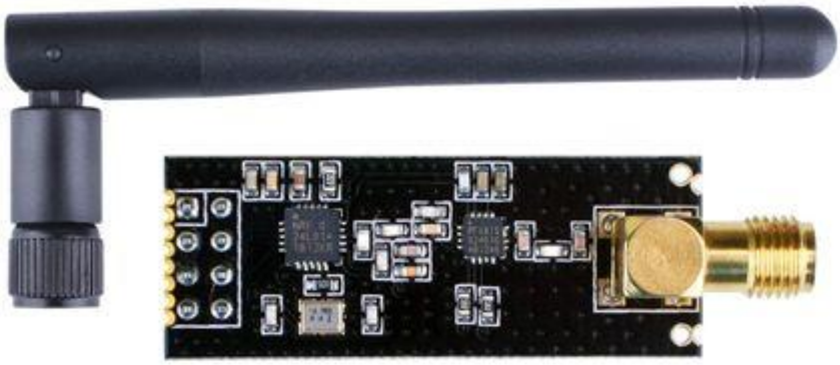


Figure III.15 NRF24|01 + PA

1.6.9.1.Features (NRF24|01+PA):

- Voltage: 3-3.6V
- Max output power: +20 dBm
- Working current in transmit mode: 115mA
- Working current in receiver mode: 45mA
- Current in mode: 4.2uA
- Operating temperature: -20-70 degree
- Receiver sensitivity: -92dBm in 2Mbps mode, -95dBm in 1Mbps mode, -104dBm in 250kbps mode
- PA growth: 20dB
- LAN growth: 10dB
- LAN noise figure: 2.6dB
- Antenna growth: 2dBI
- 2M rate: 520m

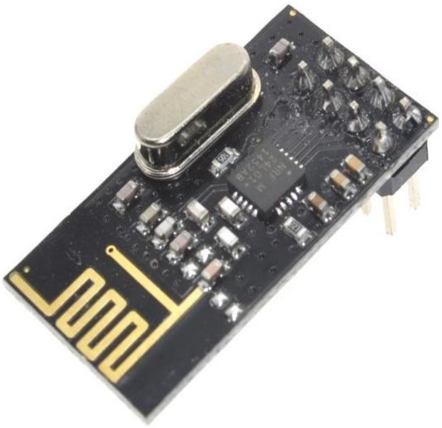


Figure III.16 NRF24|01

1.6.9.2.Features (NRF24|01):

- low operating voltage: 1.9 to 3.6V
- high-rate: 2Mbps
- Multi-frequency points: 125 frequency points
- ultra-compact: built-in 2.4GHz antenna compact 15x29mm

1.6.10. Push button:

A push-button is a simple switch mechanism to control some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal. [W24]

Figure III.17 Push Button 6x6x5 mm

1.6.10.1. Features:

- Size: 6x6x5mm.
- Temperature: -30~+70Centigrade
- Withstand Voltage:AC250V

- Rated Load: DC12V 50mA.
- Contact Resistance: ≤0.03Ω
- Insulation Resistance: ≥100MΩ

1.6.11. RESISTORS:

Resistors are electronic components which have a specific, never-changing electrical resistance. The resistor's resistance limits the flow of electrons through a circuit.

They are passive components, meaning they only consume power (and can't generate it). Resistors are usually added to circuits where they complement active components like op-amps, microcontrollers, and other integrated circuits. Commonly resistors are used to limit current, divide voltages, and pull-up I/O lines. [W25]

Figure III.18 RESISTORS

1.6.11.1. Features:

- Reference: h849.
- Voltage: 230V.
- Power: 40W.

1.7.Receiver:

Component	Features
Radio module	NRF24
Regulator	AMS1117 3.3V

Pins	Male pins
Prototype PCB	Double Side Copper prototype BCB 9*7 cm
capacitor	Electrolytic Capacitor 10uF
Arduino	Arduino Nano

Tables III.4: Receiver Components.

1.7.1. AMS1117 3.3V regulator:

The AMS1117 series of adjustable and fixed voltage regulators are designed to provide up to1A output current and to operate down to 1V input-to-output differential. The dropout voltage of the device is guaranteed maximum 1.3V, decreasing at lower load currents. [W26]

Figure III.19 AMS1117 3.3V Regulator

1.7.1.1.Features:

- Output Current of 1A
- Operates Down to 1V Dropout
- Line Regulation: 0.2% Max.
- Load Regulation: 0.4% Max.

1.7.2. Prototype PCB:

PCB prototypes are early samples of products built with the sole purpose of testing design ideas to see if they work. Although most prototypes, in general, are made to test basic user functionality, engineers require somewhat, if not entirely, functional PCB prototypes to check the complete functionality of designs. [W27]

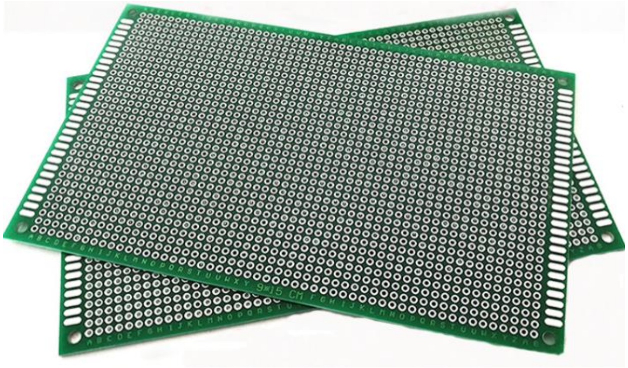


Figure III.20 Prototype PCB

1.7.2.1.Features:

- Material: double-Side Copper
- Size: 9x7 cm

1.7.3. Aluminum electrolytic capacitor:

Aluminum capacitors are polarized electrolytic capacitors whose anode electrode (+) is made of a pure aluminum foil with an etched surface. The aluminum forms a very thin insulating layer of aluminum oxide by anodization that acts as the dielectric of the capacitor. A non-solid electrolyte covers the rough surface of the oxide layer, serving in principle as the second electrode (cathode) (-) of the capacitor. A second aluminum foil called “cathode foil” contacts the electrolyte and serves as the electrical connection to the negative terminal of the capacitor. [W28]

Figure III.21 Aluminum electrolytic capacitor

1.7.3.1.Features:

- Capacitance: 10 µF

- Tolerance: ±20%
- Lifetime @ Temp: 2000 Hrs @ 85°C
- Operating Temperature: -40°C ~ 85°C

1.8.Flight Controller:

Component	Features
Arduino	Arduino Uno
Gyroscope	MPU6050

Tables III.5: Flight Controller Components.

1.8.1. Gyroscope:

The MPU6050 contains both a 3-Axis Gyroscope and a 3-Axis accelerometer allowing measurements of both independently, but all based around the same axes, thus eliminating the problems of cross-axis errors when using separate devices. [11]

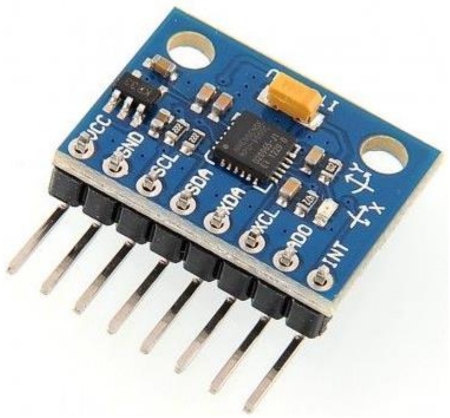


Figure III.22 MPU6050 Gyroscope

1.8.1.1.Features:

- Power supply: 3v-5V power
- Gyroscope range: + 250 500 1000 2000 ° / s
- Acceleration range: ± 2 ± 4 ± 8 ± 16 g
- Communication: standard IIC communication agreement
- Chip built-in 16-bit AD converter, 16 bits of data output
- Pin spacing 2.54 mm

1.9.Spraying System:

Component	Features
Water Pump	DC 3v 6v Mini Micro Submersible Water Pump
Channel relay	SainSmart 2-Channel Relay Module
Water Sprayer	3” Water Sprayer
Plastic Pipe	8mm plastic pipe

Tables III.6: Spraying System Components.

1.9.1. Water Pump:

A mini submersible water pump is a centrifugal water pump, which means that it uses a motor to push water outwards by power an impeller that is designed to rotate and push the water out.

Figure III.23 DC 3v 6v Mini Micro Submersible Water Pump.

1.9.1.1.Features: [W29]

- DC Voltage: 2.5-6V
- Maximum lift: 40-110cm / 15.75"-43.4"
- Flow rate: 80-120L/H
- Outside diameter of water outlet: 7.5mm / 0.3"
- Inside diameter of water outlet: 5mm / 0.2"
- Diameter: Approx. 24mm / 0.95"

- Length: Approx. 45mm / 1.8"
- Height: Approx. 30mm / 1.2"

Material: engineering plastic

Driving mode: brushless dc design, magnetic driving

1.9.2. Channel Relay Module:

This module contains two relays that are electrically isolated from the controlling input. The relays can be used to switch higher voltage and current loads than a microcontroller can traditionally accomplish. [W30]

Figure III.24. Two channel Relay Module

1.9.2.1.Features:

- Powered from 5V
- 2 channels
- Can be used as Normally Open (NO) or Normally Closed (NC)
- Optically isolated inputs

III.3. Software :

1. Arduino IDE:

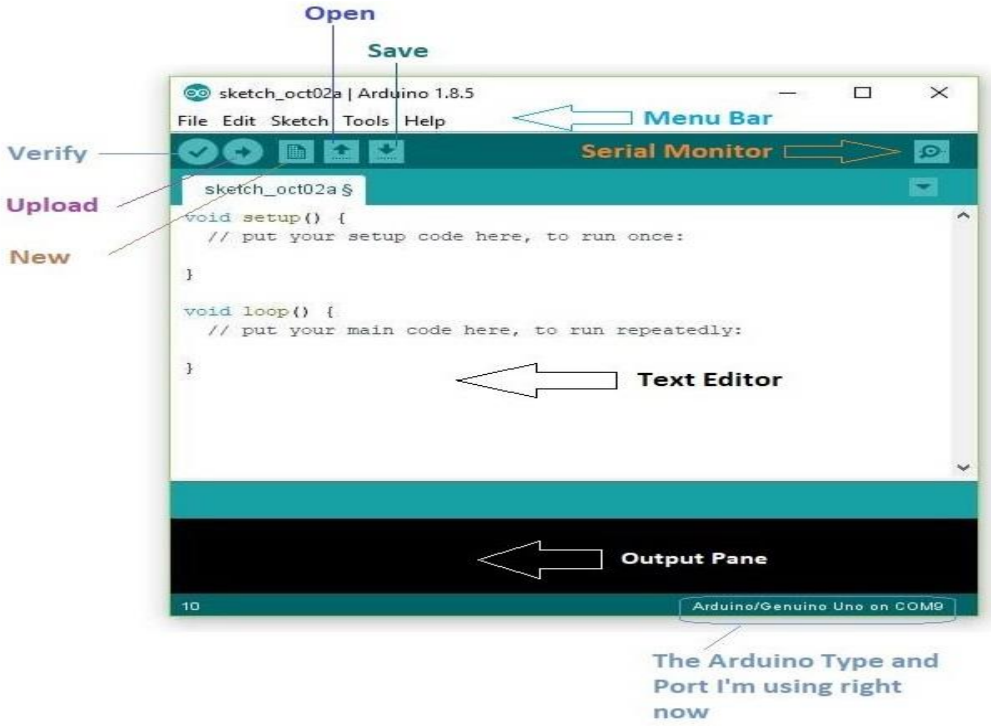
1.1.Introduction to Arduino IDE:

Arduino IDE is open-source software that is used for writing and compiling the code into the Arduino boards.

this software making the compilation of code too easy that even a common person with no prior technical knowledge can use it easily.

1.1.1. Menu Bar:

- File: you can open a new window for writing the code or open an existing one. The following table shows the number of further subdivisions the file option is categorized into.
- Edit: used for copying and pasting the code with further modification for font
- Sketch: for compiling and programming
- Tools: mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller.
- Help: In case you are feeling skeptical about software, complete help is available from getting started to troubleshooting.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure III.25 Arduino IDE software

The Six Buttons appearing under the Menu tab are connected with the running program as follows.

- Verify: used to verify the code. Click this once you have written your code.
- Upload: upload and transfer the required code to the Arduino board.
- New: is used for creating a new file.
- Open: is reserved for opening an existing Arduino project.
- Save: is used to save the current running code.

- Serial Monitor: a separate pop-up window that acts as an independent terminal and plays a vital role for sending and receiving the Serial Data.

1.1.2. Text Editor:

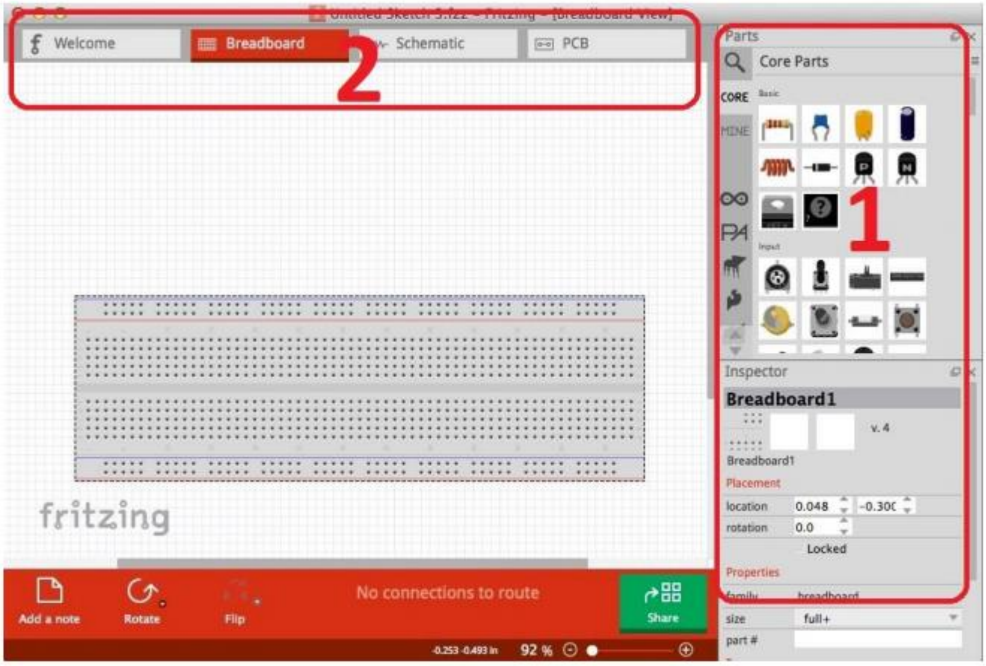
The main screen below the Menu bard is known as a simple text editor used for writing the required code.

1.1.3. Output Pane:

The bottom of the main screen is described as an Output Pane that mainly highlights the compilation status of the running code: the memory used by the code, and errors that occurred in the program. You need to fix those errors before you intend to upload the hex file into your Arduino Module.

2. Fritzing

We used this software just for the assembly of our electronic circuit.



Breadboard1
Placement
location 0.048 -0.300
rotation 0.0
Locked
Properties
family breadboard
size full+
part #

Figure III.26 Fritzing software

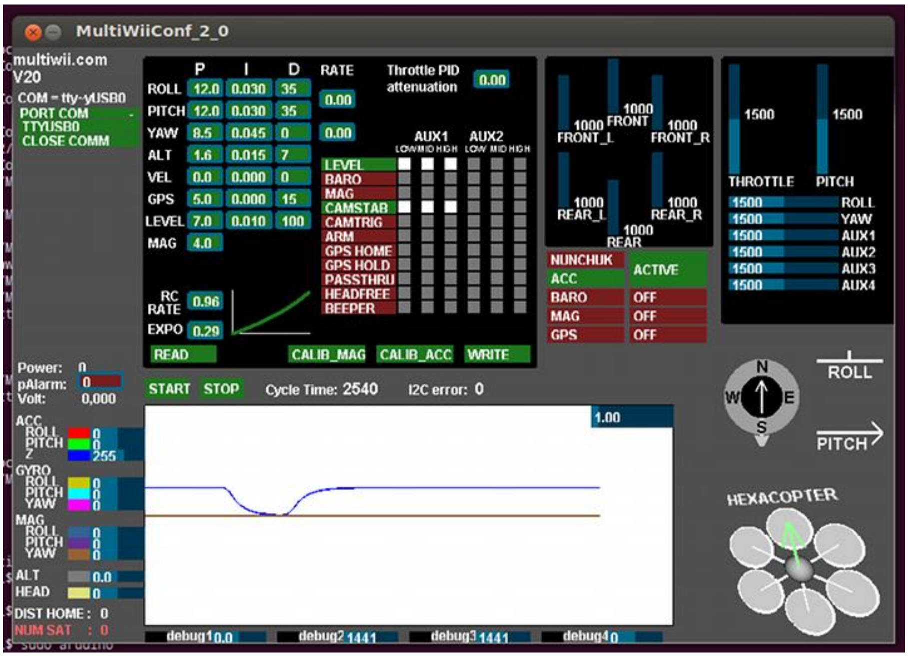
Figure shows the interface of the Fritzing software where it is made up of 2 sections:

- **Section 1:** List of components.
- **Section 2:** Toolbar composed of:
  - Breadboard

- Schematic
- PCB
- Code

3. Graphical configurator “Multiwii”

The graphical configurator is written in processing and is a cross-platform Java application, currently compiled for Linux, Windows, and macOS operating systems 32- and 64-bits architectures.



	P	I	D	RATE
ROLL	12.0	0.030	35	0.00
PITCH	12.0	0.030	35	
YAW	8.5	0.045	0	0.00
ALT	1.6	0.015	7	
VEL	0.0	0.000	0	
GPS	5.0	0.000	15	
LEVEL	7.0	0.010	100	
MAG	4.0			

Throttle PID attenuation 0.00

RC RATE 0.96

EXPO 0.29

Cycle Time: 2540

I2C error: 0

Power: 0

pAlarm: 0

Volt: 0,000

DIST HOME : 0

NUM SAT : 0

Figure III.27 Multiwii software

The appearance is a little scary, but often you do not have to climb here. Launch the configurator after inserting the flight controller into the computer. In the upper left corner, select the controller port and press the START button - the curves on the graph and the numbers next to it should run (ideally, everything should be smooth).

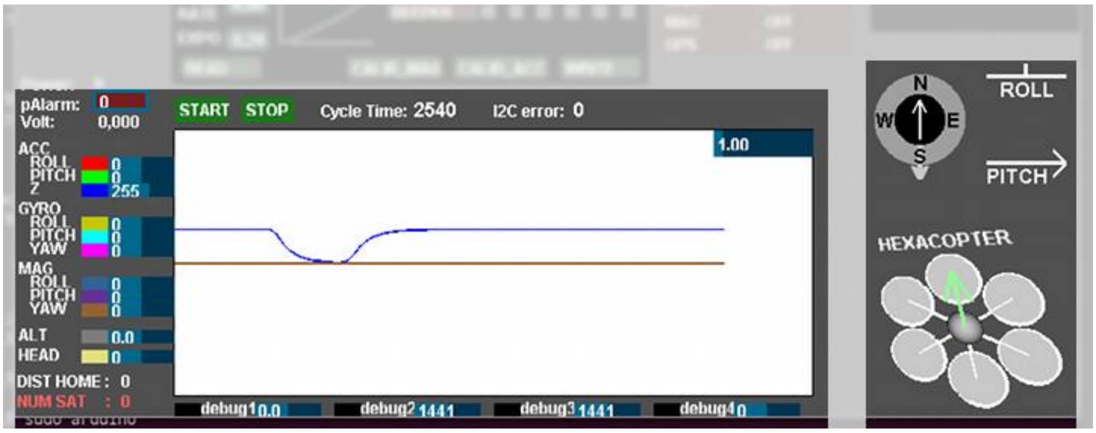
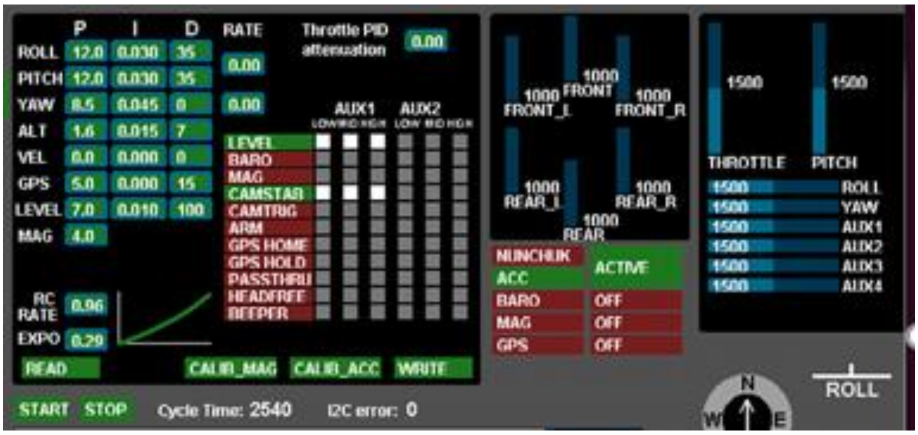


Figure III.28 Monitor space of Multiwii software

To the right of the graph is a three-dimensional model of your copter, as well as compass and tilt indicators. If the device is parallel to the horizon, and the PITCH and ROLL in the picture are tilted, press the CALIB_ACC button so that the controller remembers this position and always strives for it.



	P	I	D	RATE
ROLL	12.0	0.030	35	0.00
PITCH	12.0	0.030	35	
YAW	8.5	0.045	0	0.00
ALT	1.6	0.015	7	
VEL	0.0	0.000	0	
GPS	5.0	0.000	15	
LEVEL	7.0	0.010	100	
MAG	4.0			

Throttle PID attenuation 0.00
RC RATE 0.96
EXPO 0.29
Cycle Time: 2540
I2C error: 0

Figure III.29 Control space of Multiwii software

Above the graph are the settings of the PID controller, mode activation, and some status data. To see the values, you need to press the READ button, to change them, click on the window with a number and move the mouse, and to save - press WRITE. Never change several PID values of the examiner at one time.

III.4. Conclusion:

Throughout this chapter, we have presented in detail the electronic components that we need to complete our project. And in the next chapter, we will see how our system works.

CHAPTRE IV

DESIGNS AND REALIZATION OF THE QUADCOPTER

IV.1. Introduction:

Design and implementation are parts of the most important parts in our work that gives the final image of our thesis. So, in this chapter, we will talk about 4 steps: the design and implementation of the transmitter, Design and implementation of the receiver, Design and implementation of the quadcopter and finally Design and implementation of the spraying system.

IV.2. Transmitter:

Below we have the schematic for this project with all the connections and components values. Before getting start, we need to set the buck converters to 12.6V and 3.3V before we connect them to the circuit. That's very important. For the transmitter, we need an NRF24 module with a power amplified antenna to get more range.

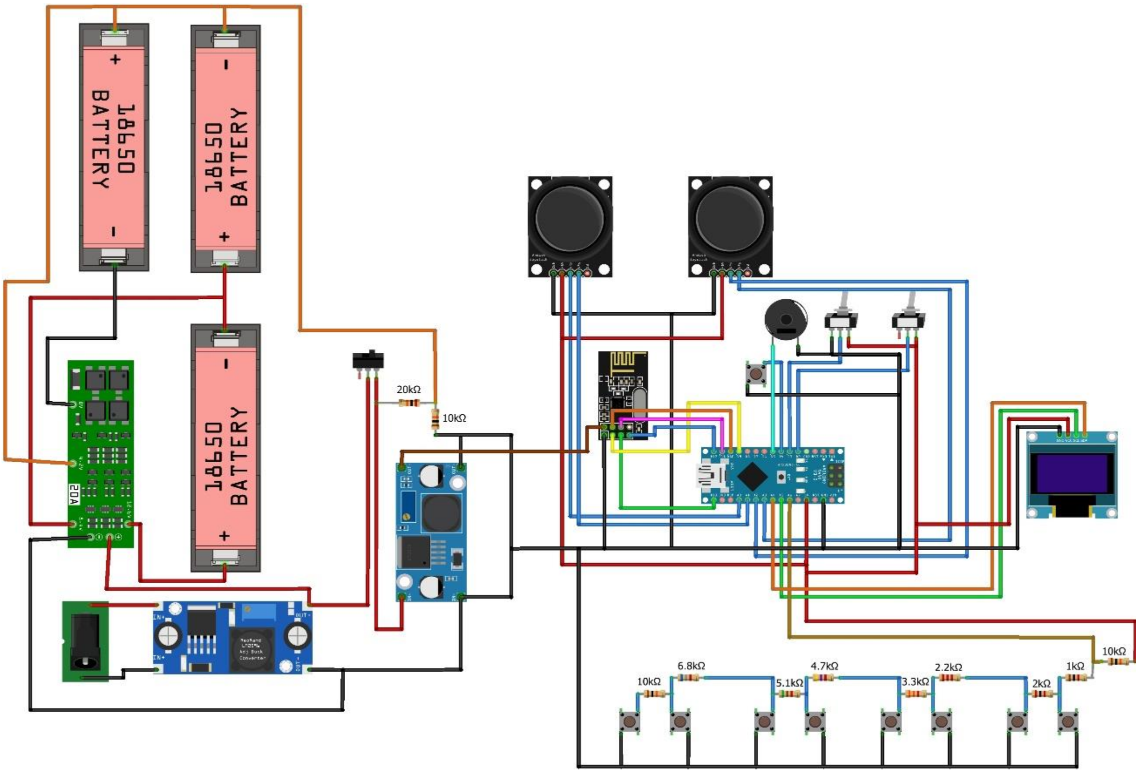


Figure IV.1. Transmitter schematic

First, we connect in series the batteries and collect them to the back part of the radio controller. Then we solder wires from the DC jack to the big boost/buck converter and we plug it in the main adaptor. Now we connect the batteries to the 3S BMS and the output from the buck converter to the BMS input/output.

The second point, is to add all the small push buttons.

Otherwise, we add wires to the potentiometers of the joysticks for GND, 5V, and signal, then the toggle switches on the top side as well as the Arduino NANO.

The most important is to set the small buck converter to 3.3V then connect it to the NRF24 module.

Finally, we set all wires to the Arduino as shown in the schematic. and add the i2c OLED display as well. The Last point is to program the Arduino board (the code is in the Appendices).

IV.3. Receiver:

First, we have to create our radio receiver that will receive for 4 values from the radio controller transmitter which are the throttle, yaw, pitch, and roll values. That’s why we will use an “Arduino NANO, the NRF24 radio module, the AMS1117 3.3 voltage regulator, a drilled PCB” already cited on chapter 3, and some capacitors. The Figure 3 shown the basic schematic of the radio receiver.

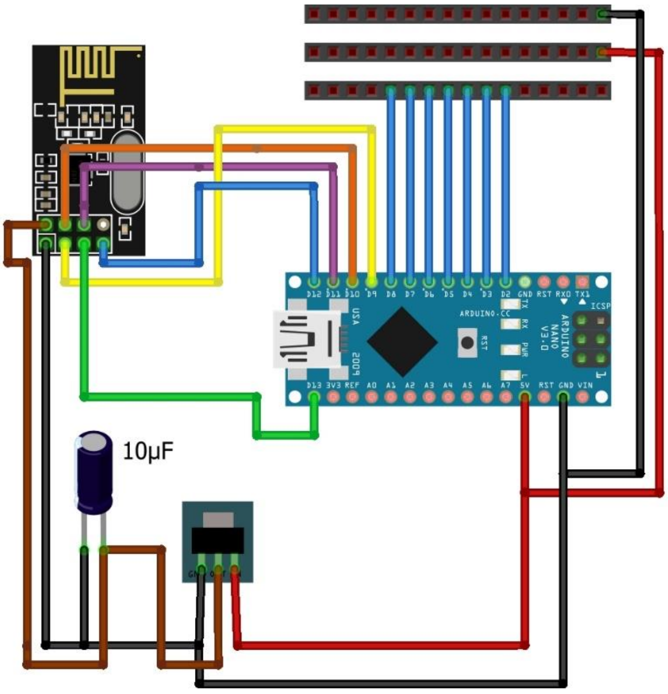


Figure IV.3. Receiver schematic

We connect the NRF24 module to the Arduino. We know that the NRF24 module works with a 3.3V supply so we need an external 3.3 voltage regulator to supply a higher current to the module, the final receiver board presented in the Figure 4.

Now we have to program this receiver (the used code in the Appendices)

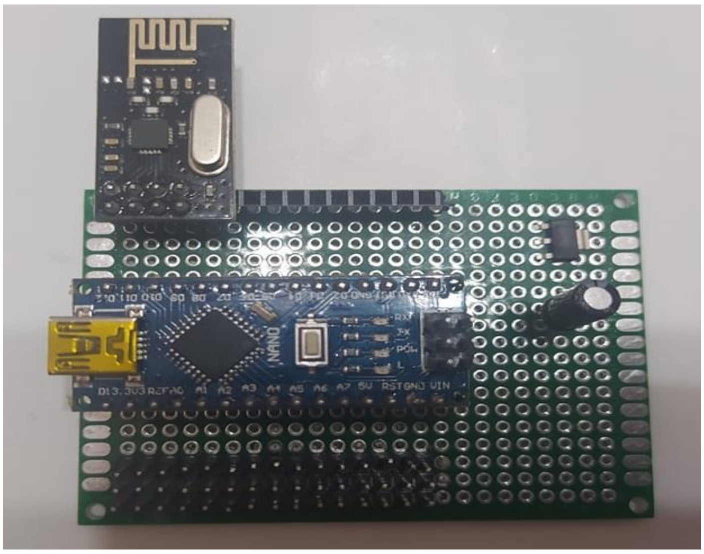


Figure IV.4. Receiver schematic

IV.4. Quadcopter:

1. Flight controller and the Quadcopter Schematic:

We will use the MPU6050 gyro and accelerometer module. It has an i2c communication because it only needs 2 pins from the Arduino, clock, and data pins. Also, we can connect multiple i2c modules to the same port. The schematic for PWM receiver in Figure 5.

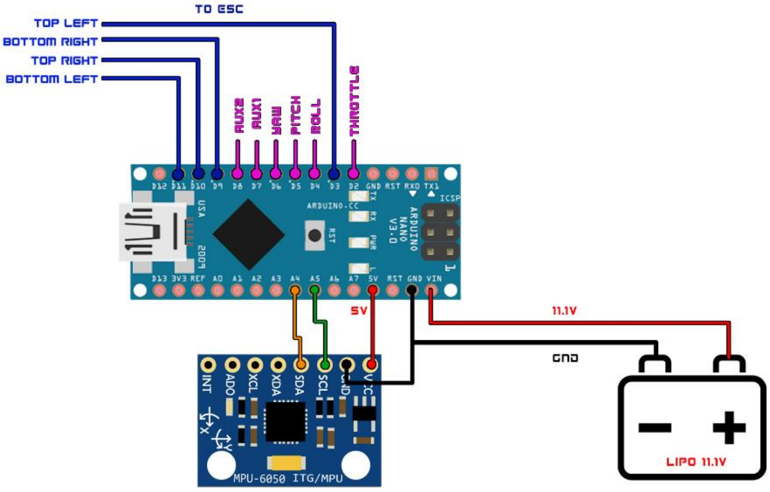


Figure IV.5. Flight Controller schematic

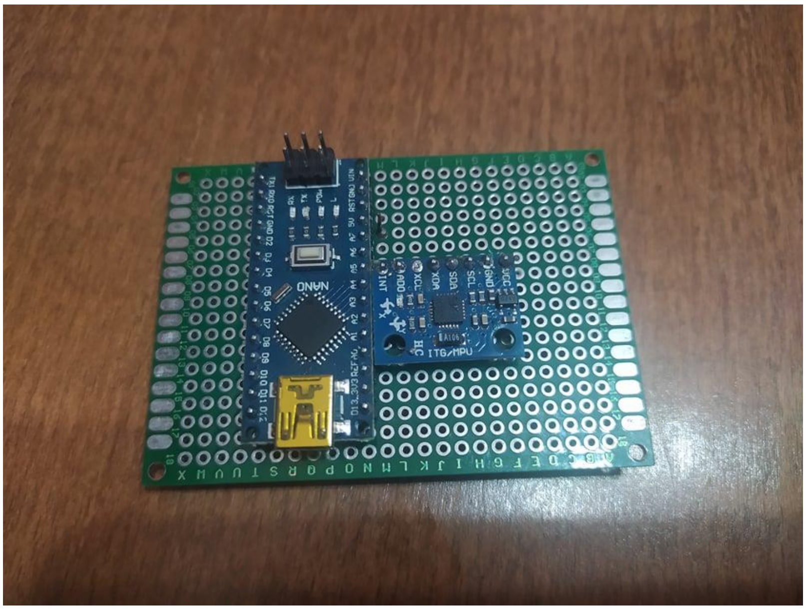


Figure IV.6. Flight Controller

With this configuration our quadcopter would be able to fly without problems, we can add more modules like a magnetometer or a barometer but with a gyroscope and accelerometer, it's enough for our quadcopter.

First, we connect pins D3, D9, D10, and D11 to each of the four motors ESCs. Each ESC also needs to share ground with the Arduino in order to understand the PMW signals.

Next, we use the Li-Po battery of the quadcopter to supply the flight controller. We will connect both ground and the 11 V directly to the Vin pin of the Arduino NANO board because the board already has 5 and 3.3, voltage regulators.

The 3rd step is to supply 5 volts to the MPU6050 module and ground as well. And connect the SDA to (A4) and SCL to (A5) Arduino analog pins and we are done. we can see that the IMU MPU6050 also has the "X" and "Y" axis on it. Place the module as centered as it can on the drone and respecting that axis.

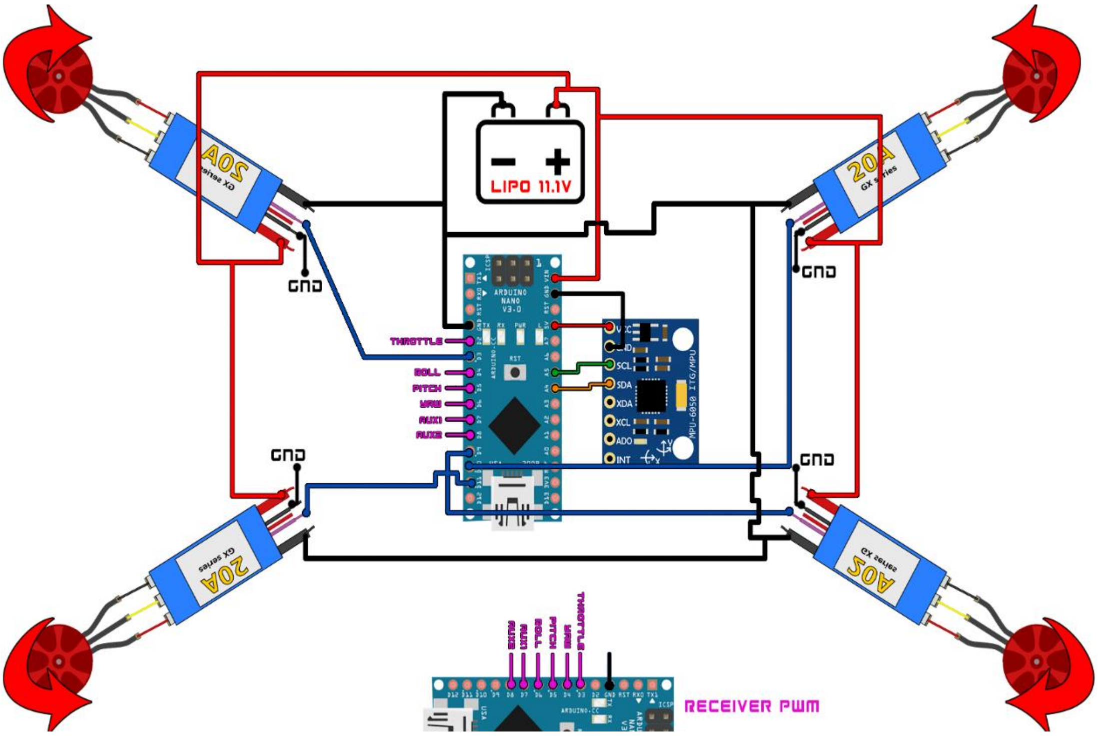


Figure IV.7. Quadcopter schematic

This is sort of a full schematic of the flight controller. We can see the receiver with the 6 channels, the 4 ESCs, and brushless motors. Everything has to share the same ground. Supply 11.1 volts to each ESC and to the flight controller Arduino. Connect the i2c MPU6050 module and place it in the middle of the drone.

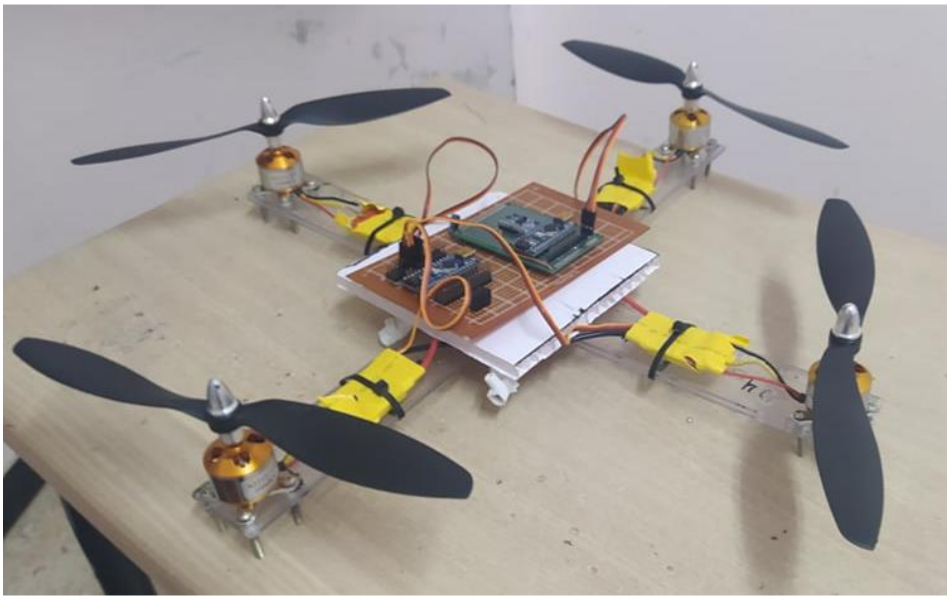


Figure IV.8. Quadcopter

2. Flight controller and the Quadcopter configuration:

To program our code, we'll use a platform that we will use is called Multiwii. The flight controller should read the accelerations and gyroscope data from the MPU module. Next, it should calculate the real angle of the quadcopter and using a PID operation to control the 4 motors and move the drone in the desired direction.

IV.5. Spraying system:

The performed circuit is simple to minimize the overall footprint and decrease potential malfunction. Using an Arduino board, we will control the LED and the 12DC water pump function. They will both be on individual schedules. While in space chemical products must be rationed therefore, we need to build a system capable of releasing only the required amount of chemical product for each plant.

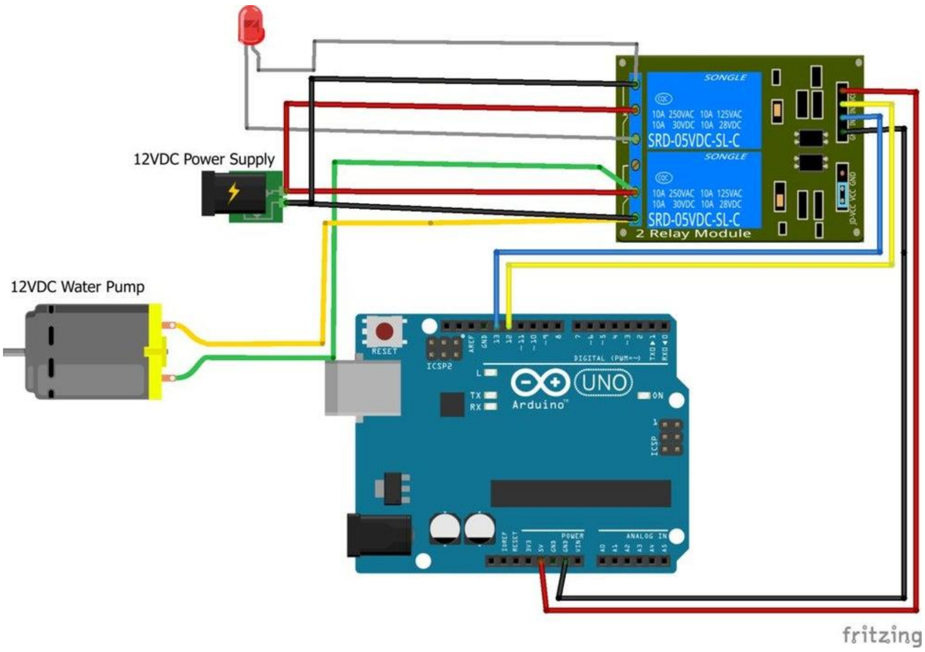


Figure IV.9. Spraying system

- Pinout
- Arduino Board ➔Two Channel Relay
  - 5v to VCC
  - GND to GND
  - Pin 12 to IN2
  - Pin 13 to IN1

Since there are so many available pins in the flight controller board, we could incorporate the spraying system with the flight controller in the same Arduino uno board.

IV.6. Conclusion:

Our research work yielded a successful development of Arduino based Quadcopter at a cheaper and affordable amount. The quadcopter can be easily made from shelf components. It can be used as a low-cost choice for various applications which includes pesticide sprinkling, end-to-end delivery within the transmitter’s RF range, surveillance in defense and other sensitive places like nation border, mapping through remote sensing, etc., with a very high level of precision.

GENERAL CONCLUSION

The first main purposes of this thesis are to build an accurate and reliable simulation representing the real physical system, develop a control algorithm to stabilize all its states, and implement the controller in the actual quadcopter to verify its performance on the real system.

Firstly, dynamics and kinematics of quadcopter aircraft were investigated and a mathematical non-linear model was acquired using classical mechanics. Major aerodynamics effects were taken into consideration, such as thrust, hub, and drag, while gyroscope and ground effects were neglected due to their small contribution to the system.

To build an accurate model, experimental identification of system parameters was performed. Those experiments include motor model, inertia calculations, quadcopter mass estimation, hub torque, thrust, and drag coefficients. Derived dynamics equations and experimental data obtained. Then, significant improvements, which are the hovering condition, the effect of roll and pitch on thrust, and the effect of altitude signal on the translation of the position to attitude, were applied to the simulation environment during model testing and analysis. These improvements were not introduced in previous theses.

A linearized model of a quadcopter system was obtained to use linear control techniques in system analysis and controller design. Improved PID control techniques were applied to stabilize the quadcopter, track the reference input signal and reject small disturbances (light wind). The response of the nonlinear model operating within the linear region was compared to the linear model response.

The 2nd main valuable output of this thesis is to build an agricultural system that uses for spray chemicals over crops, that system is controlled at a distance using our transmitter. This idea can be developed in our future use by adding more sensors to our project like a camera that makes us capable fly a drone over the crops and identify issues in a specific area and take the necessary actions to correct the problem.

BIBLIOGRAPHY

Bibliography

[1] Allison Ryan and J. Karl Hedrick (2005). “A mode-switching path planner for UAV-assisted search and rescue.” 44th IEEE Conference on Decision and Control, and the European Control Conference 2005.

[2] Atheer L. Salih, M. Moghavvemil, Haider A. F. Mohamed and Khalaf Sallom Gaeid (2010). “Flight PID controller design for a UAV Quadcopter.” Scientific Research and Essays Vol. 5(23), pp. 3660-3667, 2010.

[3] A. Zul Azfar and D. Hazry (2011). “Simple GUI Design for Monitoring of a Remotely Operated Quadcopter Unmanned Aerial Vehicle.” 2011 IEEE 7th International Colloquium on Signal Processing and its Applications.

[4] Johnson, A.A. McCook Field 1917–1927; Landfall Press: Dayton, OH, USA, 2002; pp. 190–191; ISBN 0-913428-84-1.

[5] Colwell, R.N. Determining the prevalence of certain cereal crop diseases by means of aerial photography. Hilgardia 1956, 26,223–286.

[6] Sugeno, M.; Hirano, I.; Kotsu, S. Development of an intelligent unmanned helicopter. In Proceedings of the 1995 IEEE International Conference on Fuzzy Systems, Yokohama, Japan, 14–20 March 1995; Volume 5, pp. 33–34.

[7] Gerard Sylvester; E-AGRICULTURE IN ACTION: DRONES FOR AGRICULTURE; Published by Food and Agriculture Organization of the United Nations and International Telecommunication Union ;Bangkok, 2018

[8] Ashwin Bhat; Gururaj Joshi; Nirmal Hegde; WIRELESS REMOTE CONTROLLED UTILITY DRONE; Autonomous Institute under VTU, Belgaum; Jan - May 2017; pp 7

[**9**] ADDOU Asmaa, BAHOUS Nawel « Réalisation d’une maison intelligente à base d’Arduino » Mémoire de Master, université BELHADJ Bouchaib, département génie électrique soutenu 2019/2020. « translated »

[10] Vibha Kishor; Swati Singh; Design and Development of Arduino Uno based Quadcopter; Meerut Institute of Technology; 2017, pp 15-16

[11] Angshuman Bhattacharjee; Arghya Hazra; Suvam Kumar Sar; QUADCOPTER CONTROL USING ARDUINO MICROCONTROLLER; RCC Institute of Information Technology, Kolkata; MAY-2018, pp 9

WEBOGRAPHY

Webography

[W1] https://www.wikiwand.com/en/Agricultural_drone (accessed on 22/03/2021).

[W2] https://www.yamahamotorsports.com/motorsports/pages/precision-agriculture-rmax (accessed on 26/03/2021).

[W3] https://en.wikipedia.org/wiki/Six_degrees_of_freedom (accessed on 06/04/2021).

[W4] https://charlestytler.com/modeling-vehicle-dynamics-euler-angles/ (accessed on 10/04/2021).

[W5] https://charlestytler.com/quadcopter-equations-motion/ (accessed on 10/04/2021).

[W6] https://www.amazon.in/Kewholesale-Propellers-9X4-5-Glass-Fiber/dp/B07YXSFY3Z (accessed on 14/02/2021).

[W7] https://www.rhydolabz.com/documents/26/BLDC_A2212_13T.pdf (accessed on 14/02/2021).

[W8] https://www.optimusdigital.ro/index.php?controller=attachment&id_attachment=451 (accessed on 14/02/2021).

[W9] https://www.amazon.com/ZDF-8000mAh-14-8V-LiPo-Battery/dp/B07H3B45CY (accessed on 15/02/2021).

[W10] https://www.arduino.cc/en/guide/introduction (accessed on 15/02/2021).

[W11] https://www.elprocus.com/different-types-of-arduino-boards/ (accessed on 16/02/2021).

[W12] https://www.javatpoint.com/arduino-nano-pinout (accessed on 16/02/2021).

[W13] https://vruzend.com/product/3s-battery-management-system-bms/ (accessed on 16/02/2021).

[W14] http://www.mantech.co.za/Datasheets/Products/BMS-20A-3S-S_SGT.pdf (accessed on 16/02/2021).

[W15] https://www.infobloom.com/what-is-a-dc-jack.htm (accessed on 17/02/2021).

[W16] https://ctparts.com/dc-dc-converters (accessed on 17/02/2021).

[W17] https://www.fenix-store.com/blog/the-story-behind-18650-batteries/ (accessed on 17/03/2021).

[W18] https://techterms.com/definition/joystick (accessed on 18/02/2021).

[W19] https://es.aliexpress.com/item/32851202122.html?spm=a2g0s.9042311.0.0.3da263c0sUibG5 (accessed on 18/02/2021).

[W20] https://www.yourdictionary.com/toggle-switch (accessed on 19/02/2021).

[W21] https://www.cytron.io/p-3-pins-toggle-switch (accessed on 21/02/2021).

[W22] https://www.ossila.com/pages/what-is-an-oled (accessed on 22/02/2021).

[W23] https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf (accessed on 22/02/2021).

[W24] http://www.farnell.com/datasheets/2343587.pdf (accessed on 22/02/2021).

[W25] https://learn.sparkfun.com/tutorials/resistors/resistor-basics (accessed on 23/02/2021).

[W26] https://datasheetgo.com/ams1117-datasheet-voltage-regulator/#:~:text=The%20AMS1117%20series%20of%20adjustable,the%20reference%20voltage%20to%201.5%25. (accessed on 22/02/2021).

[W27] https://www.pcbbuy.com/news/What-is-PCB-prototype.html (accessed on 28/02/2021).

[W28] https://academic.microsoft.com/topic/2776321160 (accessed on 03/03/2021).

[W29]https://www.aliexpress.com/item/32691140510.html?spm=a2g0o.productlist.0.0.51f275212e3CpL&algo_pvid=b2098159-d0bf-4172-a6f6-fac4a68491a0&algo_exp_id=b2098159-d0bf-4172-a6f6-fac4a68491a0-5 (accessed on 19/06/2021).

[W30] https://eecs.oregonstate.edu/tekbots/modules/2chrelay (accessed on 22/06/2021).

[W31] http://www.multiwii.com/software (accessed on 13/05/2021).

APPENDIX A

A.1. Inertia of the Vehicle:

Moment of inertia estimation is very important to control the vehicle it limits the amount of torque to produce angular acceleration around an axis. Moment of inertia can be estimated in various ways. One way is to calculate the moment of inertia of each component separately and then add the moment of inertia of components to get moment of inertia for X, Y and Z orientation of the vehicle. Here we have used a rather simple approach we know the mass of the vehicle and we have assumed that the vehicle is rigid and uniform so we can simply distribute the mass equally for each arm of the quadcopter to find moment of inertia of each arm which can be added accordingly to find moment of inertia for X, Y and Z orientation of the quadcopter.

A.2. Yaw Moment of Inertia:

The moment of inertia in the Z direction can also be called a yaw moment of inertia as it will resist the yaw moment. It is simply the sum of the moment of inertia of all four arms of the quadcopter

- Mass of the vehicle =M=0.9928kg
- Length of the arm=L=0.24m

Moment of inertia=I_{ZZ}

$$I_{zz} = \left(\frac{1}{3} * ml^2\right) \qquad (\boldsymbol{A.1})$$

$$I_{zz} = \left(\frac{1}{3} * \frac{0.992B}{4} * 0.24^2\right) * 4 \qquad (\boldsymbol{A.2})$$

$$I_{zz} = 0.019kgm^2 \qquad (\boldsymbol{A.3})$$

A.2.2. Roll and Pitch Moment of Inertia:

Moment of inertia in X and Y direction is the same as the vehicle is symmetrical It can be found using the same equation as above the only thing that changed in this is the length of the arm instead of using the length of the arm we use the perpendicular distance of the motors from the center of the vehicle.

$$I_{xx} = I_{yy} = 0.00963kgm^2 \qquad (\boldsymbol{A.4})$$

A.3 Propeller Moment of Inertia:

Propeller’s moment of inertia is taken from Tomas Jiinec’s master thesis. As he has calculated the moment of inertia of the same propellers 10x4 as we are using in our project. He has estimated the moment of inertia by dividing the propeller into small sections and then calculating the moment of inertia by using the parallel axis theorem

$$b = 0.04439 kgm^2 \qquad (\boldsymbol{A.5})$$

A.4 Thrust Coefficient:

Thrust coefficient depends on the weight of the vehicle, motor RPM, and propeller pitch and diameter. The thrust coefficient can be obtained by analyzing the thrust value at different RPMs of the motor. It is simpler to calculate the thrust coefficient when the quadcopter is hovering. When the quadcopter is hovering the thrust of all four rotors is equal to the weight of the vehicle. So, the thrust of one rotor is a quarter of the weight of the quadcopter

Thrust=thrust coefficient x motor rotation speed

$$b = \frac{\frac{mg}{4}}{omega^2} \qquad (\boldsymbol{A.6})$$

As we do not have the omega value at hovering for our vehicle, so we have estimated the omega value from a model developed for another quadcopter. Which same propellers are used as in our project with different weights of the vehicle. The estimated omega for our rotor is around 260.

$$b = \frac{\frac{0.9928 * 9.8}{4}}{260^2} \qquad (\boldsymbol{A.7})$$

$$b = 3.59 * 10^{-5} kgm \qquad (\boldsymbol{A.8})$$

A.5 Drag Torque Coefficient:

Drag coefficient depends on the Inertia of the vehicle, motor RPM, and propeller's pitch and diameter. The drag coefficient is estimated as a percentage of the drag coefficient of another model with the same propellers but a different moment of inertia.

$$d = 2.0810 * 10^{-6}\frac{kgm^2}{S^2} \qquad (\boldsymbol{A.9})$$

A.6 Air Resistance:

Measuring the air resistance is extremely difficult so a simple estimated value is used which do not affect the behavior of the quad significantly in closed environment.

$$d = 0.1\frac{kg}{S^2} \qquad (\boldsymbol{A.10})$$

APPENDIX B

B.1. Flight controller and Quadcopter code :

Multiwii platform flight controller with PPM signal [W31]

B.2. Transmitter code:

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
const uint64_t pipeOut = 0xE8E8F0F0E1LL; //IMPORTANT: The same as in the receiver
RF24 radio(9, 10);
struct MyData {
  byte throttle;
  byte yaw;
  byte pitch;
  byte roll;
  byte AUX1;
  byte AUX2;
};
MyData data;
void resetData()
{
  data.throttle = 0;
  data.yaw = 127;
  data.pitch = 127;
  data.roll = 127;
```

```
  data.AUX1 = 0;

  data.AUX2 = 0;

}

void setup()

{

  //Start everything up

  radio.begin();

  radio.setAutoAck(false);

  radio.setDataRate(RF24_250KBPS);

  radio.openWritingPipe(pipeOut);

  resetData();

}

int mapJoystickValues(int val, int lower, int middle, int upper, bool reverse)

{

  val = constrain(val, lower, upper);

  if ( val < middle )

    val = map(val, lower, middle, 0, 128);

  else

    val = map(val, middle, upper, 128, 255);

  return ( reverse ? 255 - val : val );

}

void loop()

{
```

```
  // The calibration numbers used here should be measured
  // for your joysticks till they send the correct values.
  data.throttle = mapJoystickValues( analogRead(A0), 13, 524, 1015, true );
  data.yaw      = mapJoystickValues( analogRead(A1),  1, 505, 1020, true );
  data.pitch    = mapJoystickValues( analogRead(A2), 12, 544, 1021, true );
  data.roll     = mapJoystickValues( analogRead(A3), 34, 522, 1020, true );
  data.AUX1     = digitalRead(4); //The 2 toggle switches
  data.AUX2     = digitalRead(5);
  radio.write(&data, sizeof(MyData));
}
```

B.3. Receiver code:

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#define channel_number 6
#define sigPin 2
#define PPM_FrLen 27000
#define PPM_PulseLen 400
int ppm[channel_number];
const uint64_t pipeIn =  0xE8E8F0F0E1LL;
RF24 radio(9, 10);
struct MyData {
  byte throttle;
```

```
  byte yaw;

  byte pitch;

  byte roll;

  byte AUX1;

  byte AUX2;

};

MyData data;

void resetData()

{

  data.throttle = 0;

  data.yaw = 127;

  data.pitch = 127;

  data.roll = 127;

  data.AUX1 = 0;

  data.AUX2= 0;

  setPPMValuesFromData();

}

void setPPMValuesFromData()

{

  ppm[0] = map(data.throttle, 0, 255, 1000, 2000);

  ppm[1] = map(data.yaw,      0, 255, 1000, 2000);

  ppm[2] = map(data.pitch,    0, 255, 1000, 2000);

  ppm[3] = map(data.roll,     0, 255, 1000, 2000);
```

```
  ppm[4] = map(data.AUX1,     0, 1, 1000, 2000);

  ppm[5] = map(data.AUX2,     0, 1, 1000, 2000);

  }

void setupPPM() {

  pinMode(sigPin, OUTPUT);

  digitalWrite(sigPin, 0);

  cli();

  TCCR1A = 0;

  TCCR1B = 0;

  OCR1A = 100;

first interrupt)

  TCCR1B |= (1 << WGM12);

  TCCR1B |= (1 << CS11);

  TIMSK1 |= (1 << OCIE1A);

  sei();

}

void setup()

{

  resetData();

  setupPPM();

  radio.begin();

  radio.setDataRate(RF24_250KBPS);

  radio.setAutoAck(false);
```

```
  radio.openReadingPipe(1,pipeIn);
  radio.startListening();
}
unsigned long lastRecvTime = 0;
void recvData()
{
  while ( radio.available() ) {
    radio.read(&data, sizeof(MyData));
    lastRecvTime = millis();
  }
}
void loop()
{
  recvData();
  unsigned long now = millis();
  if ( now - lastRecvTime > 1000 ) {
    // signal lost?
    resetData();
  }
  setPPMValuesFromData();
}
```

```
#error Delete this line befor you cahnge the value (clockMultiplier) below

#define clockMultiplier 2

ISR(TIMER1_COMPA_vect){

  static boolean state = true;

  TCNT1 = 0;

  if ( state ) {

    PORTD = PORTD & ~B00000100;

digitalWrite(sigPin,0)

    OCR1A = PPM_PulseLen * clockMultiplier;

    state = false;

  }

  else {

    static byte cur_chan_numb;

    static unsigned int calc_rest;

    PORTD = PORTD | B00000100;

digitalWrite(sigPin,1)

    state = true;

    if(cur_chan_numb >= channel_number) {

      cur_chan_numb = 0;

      calc_rest += PPM_PulseLen;

      OCR1A = (PPM_FrLen - calc_rest) * clockMultiplier;

      calc_rest = 0;

    }
```

```
    else {
      OCR1A = (ppm[cur_chan_numb] - PPM_PulseLen) * clockMultiplier;
      calc_rest += ppm[cur_chan_numb];
      cur_chan_numb++;
    }
  }
}
```

B.3. Spraying system code:

```
const int LED = 12;
const int WATERPUMP = 13;
const long onDurationLED = 43200000; //On time for LEDs
const long offDurationLED = 28800000; //Off time for LED
const long onDurationWATER = 20000;//On time for waterpump
const long offDurationWATER = 36000000;//Off time for waterpump
int LEDstate =LOW;//initial state for LED is ON
int WATERstate =LOW;//initial state for waterpump is ON
```

ملخص:

هذا البحث يعرض المنهجية الشاملة فيما يخص تصميم و تطبيق نظام تحكم لطائرة ذات أربعة محركات دوارة و ستة درجات حرية تدعى "الكوادكوبتر" مع اظافة تحسينات اليها لجعلها ملائمة للعمل في ميدان الفلاحة. أولا تم تحليل جميع القوى الديناميكية الهوائية, ثم اشتقاق النموذج الرياضي ثم تحليل جميع القوى و تأثيرا اللاخطي اعتمادا على طريقة نيوتن-اويلر. ثانيا قمنا بتصميم وتطوير جهاز تحكم ومستقبل للتحكم بحركة الطائرة يدويا قمنا ايضا بتصميم متحكم بالطيران. ثلاثا تم تركيب الطائرة و طبق عليها نظام التحكم المصمم مسبقا على لوحة متحكم الطيران. اختبر نظام التحكم فيما يخص االستقرار و المتابعة بواسطة منصة اختبار ذات ثلاث درجات حرية. رابعا تم تصميم وتركيب نظام رش للمواد الشيماوية يتم التحكم فيه عن بعد بواسطة جهاز التحكم مصمم مسبقا

Abstract:

This thesis presents an approach concerning the design and implementation of a control system for six degrees of freedom (6DOF) add to four rotors unmanned aerial vehicle (UAV), known as a quadcopter.

At First, we analyze the different forces acting on the system and aerodynamic effects. Then, a detailed mathematical nonlinear model of the quadcopter be formulated based on Newton-Euler equations. Secondly, we designed moreover developed a transmitter, receiver, and flight controller. On the other hand, the quadcopter already installed, we applied the control system to the flight controller board, the performances of the controller regarding the stability and the tracking references; were tested throughout three degrees of freedom experimental configuration. Finally, a remote-controlled system for a spraying chemical product was designed and installed to the quadcopter.

Résumé :

Ce mémoire présente une approche a la conception et la mise en œuvre d'un système de contrôle à six degrés de liberté (6DOF) ajoutés à quatre rotors pour véhicule aérien sans pilote (UAV), appelé quadricoptère.

En premier lieu, nous analysons les différentes forces agissant sur le système et les effets aérodynamiques. Ensuite, un modèle mathématique non linéaire détaillé du quadricoptère sera formulé sur la base des équations de Newton-Euler. Deuxièmement, nous avons conçu et développé un émetteur, un récepteur et un contrôleur de vol. Autrement, le quadricoptère déjà installé, d’où on applique le système de contrôle à la carte du contrôleur de vol, les performances du contrôleur concernant la stabilité et les références de suivi ; ont été testés dans une configuration expérimentale à trois degrés de liberté. Enfin, un système télécommandé de pulvérisation de produit chimique a été conçu et installé sur le quadricoptère.