People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering

Department of Electronics

Final Year Project Report Presented in Partial Fulfilment of
The Requirements for the Degree of

MASTER

In **Electrical and Electronic Engineering**

Option: **Computer Engineering**

Title:

Design & Implementation of a Quadcopter with Image Reconstruction and Object Detection

Presented by:

- **BELADJERI Mohammed Ilyes**
- **KHADRAOUI Souleymen**

Supervisor:
Mr. Ad MOHAMMED-SAHNOUN
Co-Supervisor:
Dr. Dalila CHERIFI

Registration Number:...../2016

To our families, and friends…

Acknowledgements

First of all we would like to praise almighty ALLAH for giving us the power to realize this project.

We would also thank our supervisor Mr. A.Mohammed SAHNOUN and co-supervisor Ms. Dalila Cherifi who supported us, and helped us during the different phases of the project.

Many thanks are given to all the professors and workers at the IGEE for guiding us during our cursus.

We would like to express our gratitude to Mr. Mohammed Della Krachai from the University of Science and Technology Oran who helped us with all the necessary materials to achieve this project, without forgetting Mr. Mohammed Embarek who gave us the propeller balancer.

Special Thanks to Badis, Malek, Tamila and Feriel who helped us preparing and filming the final test.

At last, and not least, we would like to thank all our families' members and friends; they have been a source of inspiration and encouragement for years.

No acknowledgement would be complete without expressing our appreciation and thankfulness for the maintenance laboratory workers for their support and help.

Abstract

Aerial robotics is seen to be one of the most dominant aspects in several areas in the last few years, and one special type of the flying robots are quadcopters, which are of an important use nowadays to ease the way of our life.

This project is about building a quadcopter, and implementing a control algorithm to stabilize the system, the second aspect is to develop a ground control station equipped with two image processing applications using Emgu.CV.

The GCS communicates with the quadcopter in order to take aerial pictures and perform image reconstruction from the captured ones, the second application is to detect a specific object in the reconstructed image or from any other captured scene.

Table of Content

List of Figures

Chapter I

Chapter II

Chapter III

List of Tables

Nomenclature

–	3D	Three Dimensions
–	BEC	Battery Eliminator Circuit
–	BLDC	Brushless Direct Current
–	CPU	Central Processing Unit
–	DC	Direct Current
–	DOF	Degrees Of Freedom
–	DoG	Difference Of Gaussian
–	ECM	Electronically Commutated Motors
–	ESC	Electronic Speed Controller
–	GCS	Ground Control Station
–	GFSK	Gaussian Frequency Shift Keying
–	GPS	Global Positioning System
–	I2C	Inter-Integrated Circuit
–	ICSP	In-Circuit Serial Programming
–	IDE	Integrated Development Environment
–	IMU	Inertial Measurement Unit
–	LiPo	Lithium Polymer
–	MEMS	Micro Electromechanical Sensors
–	PID	Proportional, Integral, Derivative
–	PWM	Pulse Width Modulation
–	RANSAC	Random Sample Consensus
–	RC	Radio Control
–	RPM	Revolution Per Minute
–	SIFT	Scale Invariant Feature Transform
–	SURF	Speeded Up Robust Feature
–	UART	Universal Asynchronous Receiver/Transmitter
–	UAV	Unmanned Aerial Vehicle
–	USB	Universal Serial Bus
–	VTOL	Vertical Take Off & Landing

Introduction

The development of autonomous flying robots have increased in the last years because of the increasing amount of applications for these unmanned aerial vehicle (UAV) in many fields such as, surveillance, reconnaissance missions, aerial photography and video, Emergency procedures in firefighting, and many others that are emerging. In all cases, the aim of this developments in the field is the necessity to replace human intervention.

In modern times, there is a wide variety of aircrafts used for UAVs, like fixed-wing airplanes, airships, helicopters and quadcopters. Nevertheless, the platforms that have caught a lot of attention for development projects are multirotor helicopters, in particular quadrotors, because these flying robots have a lot of advantages with respect to other aircrafts. Quadcopters have a simple structure and a great flight capacity and maneuverability, such as vertical take-off and landing (VTOL) and they can achieve a hovering flight.

The quad rotor is an aerial vehicle with four arms whose motion is based on the speed of four motors. Due to its ease of maintenance, high maneuverability, vertical takeoff and landing capabilities (VTOL), etc., it is being increasingly used in reconnaissance, mapping and real time photography. The constraint with the quad rotor is the high degree of control required for maintaining the stability of the system and the real time relation to its applications.

The quadcopter is an inherently unstable system. There are six degrees of freedom – translational (X, Y and Z) and rotational ($\varphi, \theta \; and \; \phi$) parameters. These are being controlled by 4 actuating signals. The x and y axis translational motion are coupled with the roll and pitch. Thus we need to constantly monitor the state of the system, and give appropriate control signals to the motors. The variation in speeds of the motors based on these signals will help stabilize the system.

Large area mapping and reconnaissance techniques have been limited, resource consuming and high-risk jobs for humans. The development of quadcopters that can achieve these missions will be time and cost effective.

The aim of this project is the development and implementation of a stabilized quadcopter using PID controllers, the UAV will achieve reconnaissance flights where it takes aerial images, along with a ground control station (GCS) that is developed to do processing of the images captured, the GCS is equipped with two main image processing applications which are the aim of the quadcopter flights, the first one is image reconstruction from different captured photos using Automatic Panorama Image Stitching Algorithm, the second application is the object detection in a certain scene or from the reconstructed picture with invariant features using the Speeded Up Robust Feature (SURF) Algorithm, alongside the applications the quadcopter can be controlled directly from the GCS using a joystick for manual flights or GPS waypoints navigation for autonomous flights in order to take aerial pictures in a predefined coordinates.

Chapter I

Modeling & Controller Design

This chapter discusses the mathematics and the theory behind the quadcopter kinematics and dynamics, in addition to the controller design that stabilizes the quadcopter and allow its maneuvers.

I.1 Introduction

Controller design requires prior modeling of the system to know its behavior. Quadrotors have their four propellers placed on the ends of a cross-like structure. To maintain the balance of the overall torque, one pair of rotors spins in a clockwise direction while the remaining pair spins in a counter-clockwise direction. The speed of every rotor is controlled independently to generate the thrust and torque to move the aircraft.

I.2 Modeling

Before jumping into the mathematical model equations, certain assumptions should be defined in order to go further in the theory:

- The quadcopter structure is considered as a rigid body [1].
- The quadcopter structure is symmetric [1].

Another important aspect of the mathematical model is the coordinate system used which will vary whether a plus ("+") or "X" configuration is presented (Figure I.1). The chosen model and conventions will become very important as we work through the model.

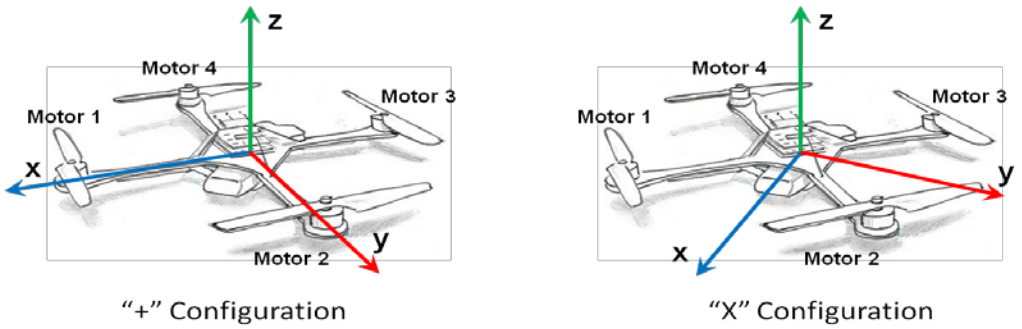


Figure I.1: *X and + configuration [2]*

To start deriving quadcopter dynamics, introducing the two coordinate systems in which will operate. The inertial frame is defined by the ground, with gravity pointing in the negative z direction. The body frame is defined by the orientation of the quadcopter, with the rotor axes pointing in the positive z direction and the x and y directions pointing as the "X" configuration which will be used in the whole project.

I.2.1 Kinematics

To study the kinematics of the quadcopter, the illustration of the structure, body frame and inertial frame (Figure I.2) must be presented including the corresponding Roll, Pitch, Yaw Angles and the angular velocities p, q and r.

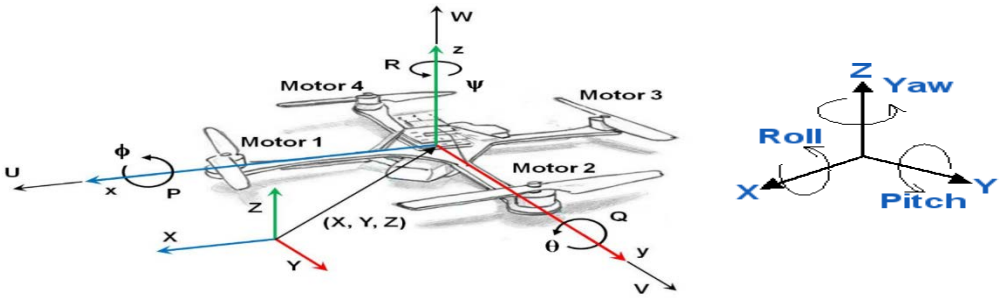


Figure I.1: *The inertial and body frames of a quadcopter [2]*

The absolute linear position of the quadcopter is defined in the inertial frame **x, y, z** axes with **ξ**. The angular position is defined in the inertial frame with three Euler angles **η**. Pitch angle **θ** determines the rotation of the quadcopter around the y-axis. Roll angle **φ** determines the rotation around the x-axis and yaw angle **ψ** around the z-axis, the angular velocity in body-fixed frame is defined by the vector $\boldsymbol{\omega_B}$, and the linear velocity in body-fixed frame is defined by the vector $\mathbf{V_B}$.

$$\xi = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \ \eta = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}, \quad V_B = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \ \omega_B = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

It's very convenient to have both coordinate systems in order to control the quadcopter for that reason it is required to do transformation from body frame to inertial frame and vice versa.

The rotation matrix R is defined for body frame to inertial frame transformation which is the result of the rotation over x, y and z respectively [3].

$$R = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\varphi - S_\psi C_\theta & C_\psi S_\theta C_\varphi + S_\psi S_\varphi \\ S_\psi C_\theta & S_\psi S_\theta S_\varphi + C_\psi C_\theta & S_\psi S_\theta C_\varphi - C_\psi S_\varphi \\ -S_\theta & C_\theta S_\varphi & C_\theta C_\varphi \end{bmatrix} \text{ Where: } \begin{matrix} C_x = Cos(x) \\ S_x = Sin(x) \end{matrix}$$

The matrix R is orthogonal and $R^{-1} = R^T$.

The first set of equation (Eq I.1) [4] is describing the change of position according to quadrotor's attitude in its velocity measured in the body frame

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} = R^{-1} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \qquad \text{I.1}$$

The transformation between angular rates in Earth-fixed frame to body-fixed frame is given by (Eq I.2) [4].

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = E \begin{bmatrix} \dot{\varphi}(t) \\ \dot{\theta}(t) \\ \dot{\psi}(t) \end{bmatrix} \text{ Where } E = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\varphi & sin\varphi cos\theta \\ 0 & -sin\varphi & cos\varphi cos\theta \end{bmatrix} \qquad \text{I.2}$$

I.2.2 Dynamics

The dynamics of the quadcopter are represented by all the forces and torques applied to it and the inertia of the structure itself, also it is more convenient to formulate all the equations of motion in the body-fixed frame because the inertia matrix is time-invariant, the advantage of symmetry in the frame, the measurements are done onboard and finally the control forces are given in the body-fixed frame.

Since the quadcopter is a 6 DOF rigid-body then the dynamics are described by Newton-Euler equation (Eq I.3) [5] which is similar to newton's second law

$$\begin{bmatrix} mI_3 & 0_3 \\ 0_3 & I \end{bmatrix} \begin{bmatrix} \dot{V}_B \\ \dot{\omega}_B \end{bmatrix} + \begin{bmatrix} \omega_B \times mV_B \\ \omega_B \times I\omega_B \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix} \quad \text{I.3}$$

Where:

- "m" is the mass of the quadcopter.
- "I" is the inertia matrix of the quadcopter, given by (Eq I.4) [4].

$$\text{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad \text{I.4}$$

- "$I_{3\times3}$" is the identity matrix.
- "F" is the total force.
- "τ" is the total torque.

The torque equations of the roll (Eq I.5), pitch (Eq I.6) and yaw (Eq I.7) for this project are derived as follow:

$$\tau_\varphi = L * (-f_1 + f_2 + f_3 - f_4) = LC_T * (-\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2) \quad \text{I.5}$$

$$\tau_\theta = L * (-f_1 - f_2 + f_3 + f_4) = LC_T * (-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2) \quad \text{I.6}$$

$$\tau_\psi = (-\tau_1 + \tau_2 - \tau_3 + \tau_4) = C_\tau * (-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \quad \text{I.7}$$

Thus the total torque is $\tau = \begin{bmatrix} LC_T * (-\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2) \\ LC_T * (-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2) \\ C_\tau * (-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \end{bmatrix}$

- "ω_i" is the angular velocity of the rotor "i".
- "L" is the length of the arm (from center of mass to the propeller).
- "C_τ" is the torque coefficient of the system motor/propeller.
- "C_T" is the thrust coefficient of the system motor/propeller.

The total force F includes the thrust of the motors F_b that is behind all the quadcopter maneuvers and it is derived as (Eq I.8):

$$F = F_b - mg\begin{bmatrix}0\\0\\1\end{bmatrix} \quad \text{I.8}$$

Where $F_b = \begin{bmatrix}0\\0\\f\end{bmatrix}$, and $f = \sum_{i=1}^{4} f_i$

- The motor thrust is defined by $f_i = C_T * {\omega_i}^2$
- The motor torque is defined by $\tau_i = C_\tau * \omega_i^2$

At the end, the relationship between propeller speeds and the generated thrusts and moments is summarized by (Eq I.9):

$$\begin{bmatrix}f\\ \tau_\varphi\\ \tau_\theta\\ \tau_\psi\end{bmatrix} = \begin{bmatrix}C_T & C_T & C_T & C_T\\ -L*C_T & L*C_T & L*C_T & -L*C_T\\ -L*C_T & -L*C_T & L*C_T & L*C_T\\ -C_\tau & C_\tau & -C_\tau & C_\tau\end{bmatrix}\begin{bmatrix}{\omega_1}^2\\ {\omega_2}^2\\ {\omega_3}^2\\ {\omega_4}^2\end{bmatrix} \quad \text{I.9}$$

I.3 Controllers Design

The controller design is a large field itself, and the choice of controller for any system is a real struggle, after researches we decided to go with the PID controller because it was covered in our curriculum, and also because it is a successful one when it comes to quadcopter control, before diving into the controller design for our quadrotor we will discuss the theory behind PID controllers briefly.

I.3.1 PID Controller

PID stands for Proportional, Integral and Derivative, it is a control loop feedback mechanism controller (Figure I.3) commonly used in industrial control systems, and it continuously calculates the error value as the difference between a desired set-point and a measured process variable. The controller attempts to minimize the error over time by adjustment of a control variable [6].

In robotics, PID technique represents the basics of control even though a lot of different algorithms such as Back Stepping Techniques, Sliding mode Control, Adaptive Control...Etc., these algorithms provide better performance than PID, but this last structure has been chosen to be implemented in this project for the following reasons:

- Simple structure.
- Good performance for several processes.
- Tunable even without a specific model of the controlled system.

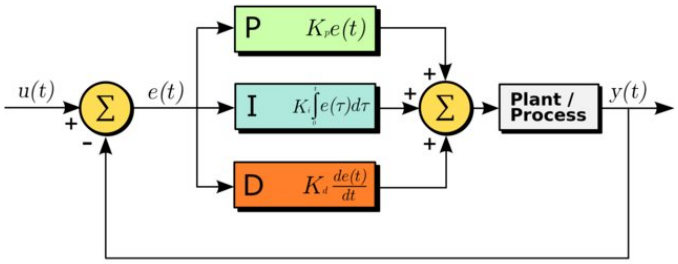


Figure I.3: *PID Structure Diagram [6]*

A proportional controller (Kp) will have the effect of reducing the rise time and will reduce but never eliminate the steady-state error. The proportional component depends only on the difference between the set point and the process variable. This difference is referred to as the Error term. The proportional gain (Kp) determines the ratio of output response to the error signal. An integral control (Ki) will have the effect of eliminating the steady-state error for a constant or step input, but it may make the transient response slower. The integral component sums the error term over time. The result is that even a small error term will cause the integral component to increase slowly. The integral response will continually increase over time unless the error is zero, so the effect is to drive the Steady-State error to zero. A derivative control (Kd) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response [7].

I.3.2 PID Controller Design

In order to control a quadcopter we need to control its position thus controlling the main angles of rotation for the quadcopter (Roll, Pitch and Yaw), but this will be difficult to stabilize, for this reason the use of double loop control is better, a minor loop that controls the rate of change for the angles and a major loop that control the angles of rotation, from this point we need to design three double loop controllers, one for each angle of rotation and its angular rate, the roll, pitch and yaw controllers are shown in figures I.4 - I.5 - I.6 respectively.

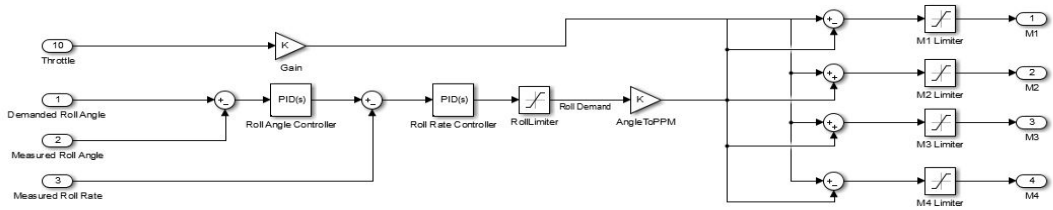


Figure I.4: *Roll Controller*

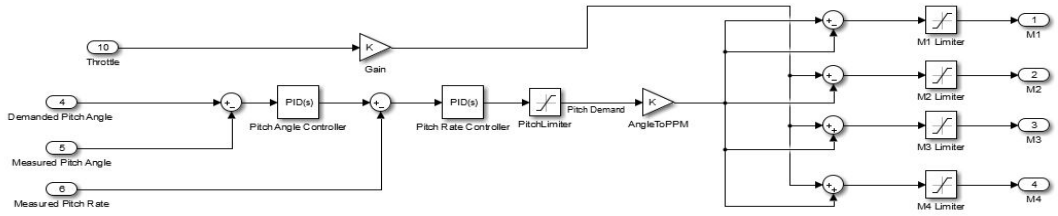


Figure I.5: *Pitch Controller*

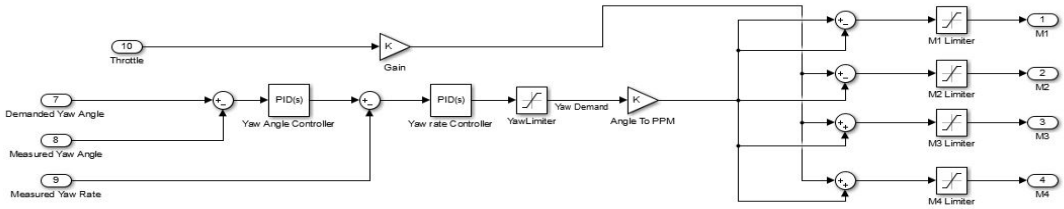


Figure I.6: *Yaw Controller*

After designing the controllers, the struggle was to find the appropriate PID gains K_p, K_i and K_d for each one, trying to tune all the parameters at the same time experimentally is difficult when the system is over parametrized, the most suitable method was to use the Ziegler-Nichols Method [33], where we found accurate results from the parameters derived.

There is another controller that has been designed for the purpose of altitude control which require the use of an ultrasonic sensor and barometer, this one is used to limit the quadcopter from losing or gaining altitude due to wind or aerodynamic phenomena and also to map the motor forces to the required altitude.

The use of two sensors is deployed because the barometer is inaccurate at low altitude, so we use the ultrasonic sensor below 5 meters and the barometer above 5 meters, the designed controller is shown in figure I.7.

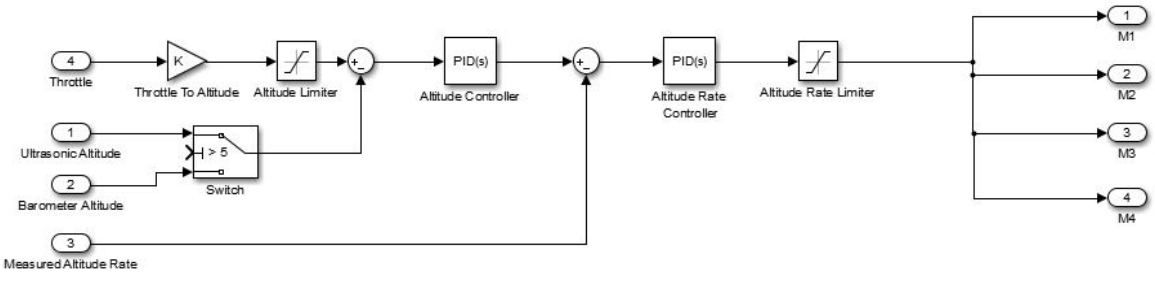


Figure I.7: *Altitude Controller*

I.4 Summary

In this chapter the theory behind the quadcopter manoeuvers has been discussed and the controllers responsible of stabilizing the quadcopters has been designed to be implemented.

Chapter II

The Quadcopter Implementation

In this chapter the implementation of the quadcopter is discussed from the hardware to the software then the ground control station is introduced to complete the overall system.

II.1 Introduction

The development and implementation of quadcopters require many steps to achieve the final project, the first thing to start with was to define the requirements of the quadrotor then list all the appropriate parts that meet these requirements based on the calculations performed, in addition to the quadcopter a ground control station is developed to perform the different required tasks.

II.1.1 Requirements

- The quadcopter needs to be medium size, lightweight and rigid.
- The motors should lift twice the weight of the full load quadcopter.
- The quadcopter frame should absorb vibrations.
- Considerable flight time in order to achieve the required application.
- The quadcopter should have a reliable communication with the ground station.

II.1.2 System Overview

In order to achieve the above requirements, the constructed block diagram system gives the global overview and show the necessary parts and there interrelation (Figure II.1).

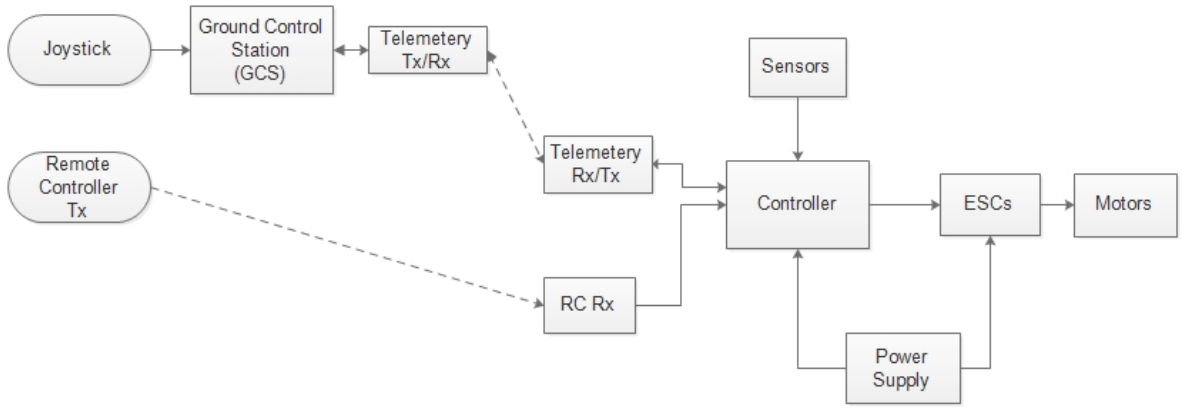


Figure II.1: *System Overview*

The implementation of the overall system mentioned above will be divided into two parts, the first part will discuss the mechanical construction of the quadcopter the second part is about the electrical construction which include the software implementation and the algorithm deployed in the system.

II.2 Mechanical Construction

The mechanical construction of the quadcopter is simple, which is one of the many advantages that made them so popular, but it is necessary to know the material and the geometrics of the design used, here the frame and the propellers characteristics will be covered.

II.2.1 The Frame

The quadcopter frame has four arms, each connected to one motor. The front of the UAV tends to be between two arms (X configuration), its size can vary from the size of palm to the size of truck, in our case we use the common size for hobbyist which is easy to stabilize with the motors available in the market.

The frame can be made from different types of materials wood, foam, plastic, carbon fiber…etc., each of these materials has its own advantages and disadvantages, and the selection can be made mostly about two features weight and solidity, we have chosen the fiber glass frame (Figure II.2) which tends to have medium values for the mentioned features and it serves the purpose very well.

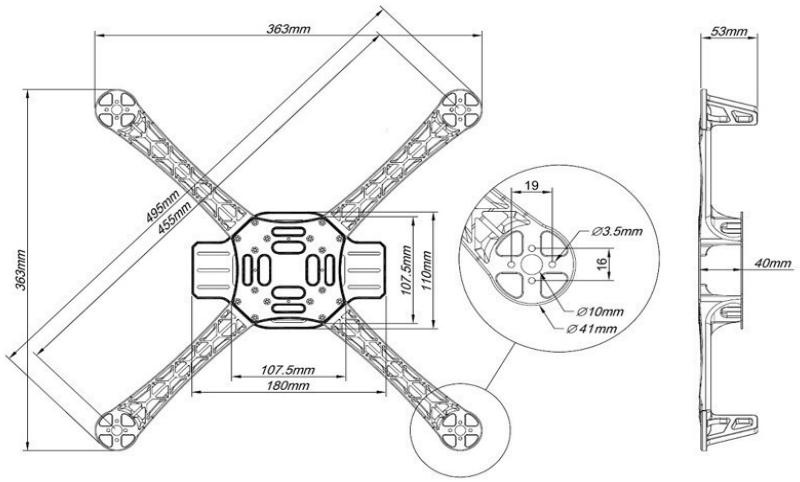


Figure II.2: *The Quadcopter Frame [8]*

II.2.2 The Propellers

Without getting into the aerodynamic theory, a propeller is the type of fan that transmits power by converting rotational motion into thrust that lifts our quadcopter, and the choice of the propeller can be difficult to decide because it has many characteristics in different materials (plastic, fiber carbon), and it depends on the motor specifications.

The propellers of the quadcopter come in pairs, one pair gives thrust when rotating clockwise, and another pair that gives thrust when rotating counter-clockwise, and they are characterized by number of blades, diameter (Figure II.3) and the pitch (Figure II.4).

The thrust produced by a propeller depends on the density of the air, on the propeller's RPM, its diameter, the shape, the area of the blades and on the pitch.

The most common used number of blades in quadcopters propellers is two, and the diameter (figure II.3) define its inertia, less inertia means easier to speed up and slow down, so the choice is really application specific.

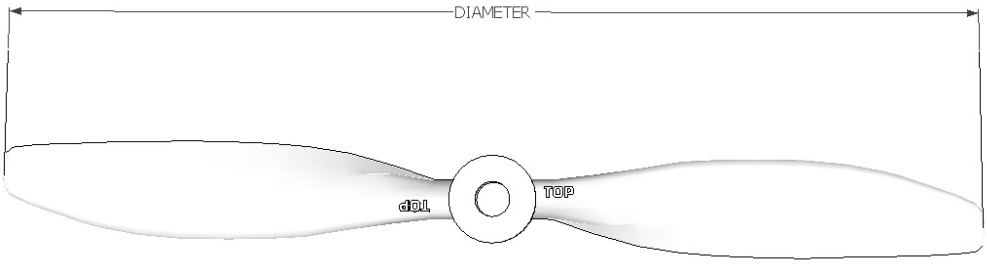


Figure II.3: *Propeller Diameter [9]*

A propeller's efficiency relates to the angle of attack which is defined as the blade pitch (Figure II.4), most well-designed propellers have an efficiency of 80%, unfortunately the ones available were off balance, so we had to balance them again [Appendix A].

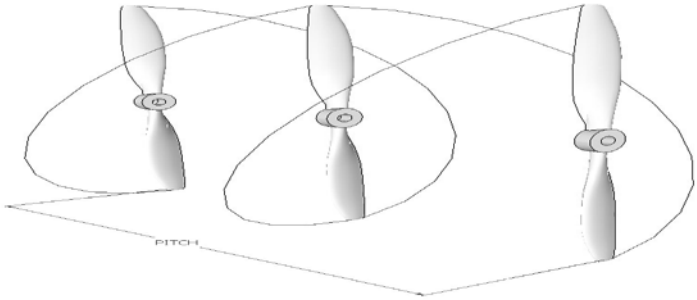


Figure II.4: *Propeller Pitch (Distance per one revolution) [9]*

In this project the propellers used are with 25.4cm diameter and 11.43cm pitch which fits with the characteristics of the motors chosen.

II.3 Electrical Construction

The electrical and electronic construction of the quadcopter is quite tricky but with the available integrated circuits of the modules like the GY-88 which contains three sensors and the ESC that can generate the required signals to drive the BLDC motor we can easily combine the modules and interface them with the microcontroller.

II.3.1 The Input System

The quadcopter takes inputs from three main parts, Sensors that are onboard, Radio Control system used by the pilot to move manoeuver the quadcopter and the telemetry to receive control data from the GCS also the Camera is deployed to take the pictures is considered as an input system.

II.3.1.1 Sensors

The quadcopter uses many different sensors, these are considered the most important components in stability and control of the quadrotor, so the choice is critical, and the sensors used in the project are Inertial Measurement Unit (IMU), Ultrasonic Sensor and the GPS sensor and the.

A. Inertial Measurement Unit (IMU)

An inertial measurement unit is an electronic device that measures and reports a body's specific force, angular rate, and sometimes the magnetic field surrounding the body, using a combination of accelerometers and gyroscopes, sometimes also magnetometers and barometer, normally these are three-axis Micro-Electro-Mechanical Systems (MEMS) [10].

- **Accelerometer**

Accelerometers measure linear acceleration in up to three. The units are normally in gravity (g) which is 9.81 meters per second per second. The output of an accelerometer can be integrated twice to give a position, though it is very unstable because it is sensitive to noise it can detect any small amount of force. A very important characteristic of three axis accelerometers is that they detect gravity, and as such, can know which direction is down.

- **Gyroscope**

A gyroscope measures the rate of angular change in up to three angular axes. The units are often degrees per second. Note that a gyroscope does not measure absolute angles directly, but you can integrate to get the angle, just like an accelerometer, it is subject to drift over time due to discrete integration.

- **Magnetometer**

An electronic magnetic compass or simply magnetometer is able to measure the earth's magnetic field and used it to determine the drone's compass direction (with respect to magnetic north). This sensor is almost always present if the system has GPS input and it is available in one to three axes.

- **Barometer**

Since atmospheric pressure changes the farther away you are from sea level, a pressure sensor can be used to give you a pretty accurate reading for the UAV's height. Most flight controllers take input from both the pressure sensor and GPS altitude to calculate a more accurate height above sea level.

B. GY-88 10DOF IMU

The implementation of the quadcopter was done using the module GY-88 10DOF IMU (Figure II.5) which includes a three axis accelerometer, three axis gyroscope, three axis magnetometer and a barometer.

This module is interfaced via I2C protocol [27], [28] where via programming it can output all the values needed to control the quadcopter.

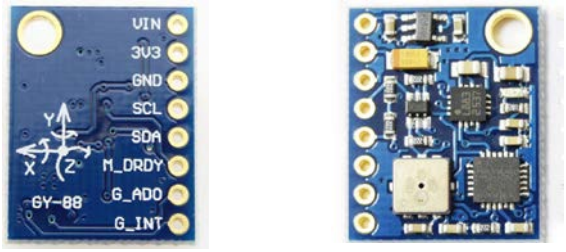


Figure II.5: *GY-88 10DOF IMU*

The module combines the four different sensors listed in the following:

- MPU6050: is the motion processing unit that combines the accelerometer and the gyroscope in one electronic chip, the raw data of the sensors are accessed via I2C.
- HMC5883L: is the three axis magnetometer.
- BMP085: is the barometer.

The internal schematic of the GY-88 module is presented in figure II.6.

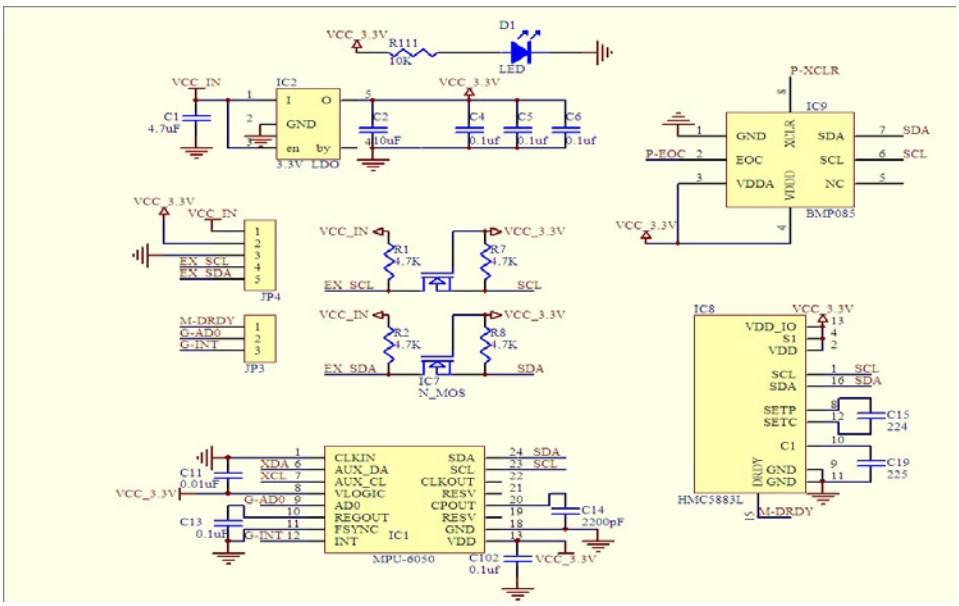
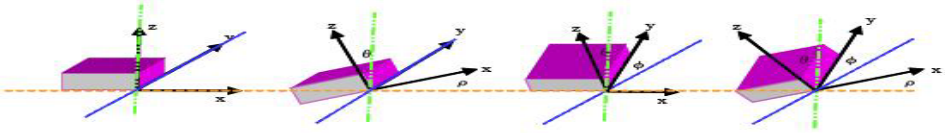


Figure II.6: *GY-88 Internal Schematic [11]*

The most important sensor that stabilizes the quadcopter is the motion processing unit which includes the accelerometer and the gyroscope on the MPU6050 but these two sensors need more processing and filtering in order to give an accurate angle of rotation for the quadcopter.

After reading the digital data from the accelerometer the datasheet is used to convert the data to meaningful values which are the force vectors exerted on the accelerometer, then 3D geometry projections will determine the rotation angles (Figure II.7).



Roll Angle $$\rho = \arctan\left(\frac{A_X}{\sqrt{A_Y^2 + A_Z^2}}\right)$$

Pitch Angle $$\phi = \arctan\left(\frac{A_Y}{\sqrt{A_X^2 + A_Z^2}}\right)$$

Yaw Angle $$\theta = \arctan\left(\frac{\sqrt{A_X^2 + A_Y^2}}{A_Z}\right)$$

Figure II.7: *Angle Calculation from the Accelerometer [12]*

For the gyroscope the digital data is converted to angular velocity, then integration over time is performed to find the angles of rotation for each axis (Figure II.8).

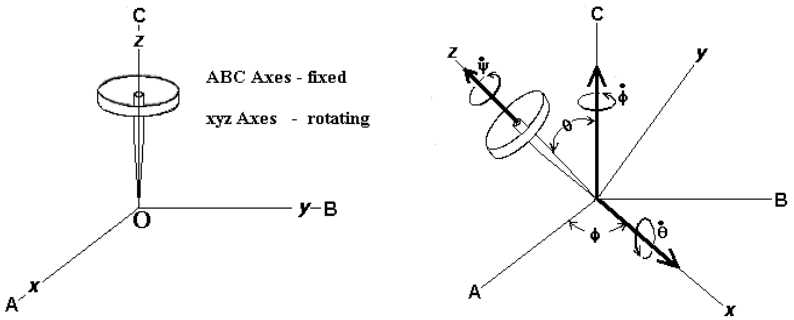


Figure II.8: *Angles from Gyroscope [13]*

Since the accelerometer is very unstable and sensitive and the gyroscope tends to drift over time, fusion sensor method and filtering are used in order to combine both angles from the gyro and accelerometer, this will result in very accurate angle. There are several methods of filtering, the most common ones are Kalman Filter and the complementary filter which is the one used with the sensors onboard.

The complementary filter uses low-pass filter for the accelerometer and high-pass filter for the gyroscope, this way all the frequencies are available at all time [14], and figure II.9 shows the general block diagram of the filter.

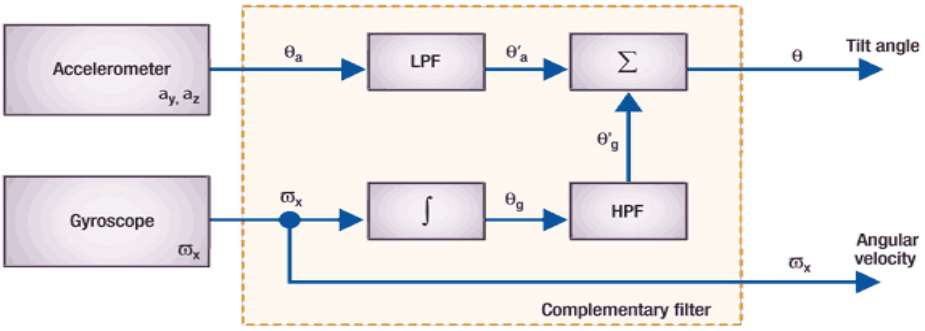


Figure II.9: *Complementary filter [15]*

The reason complementary filter is used instead of Kalman was because of the simplicity to understand the theory and it is programmable into the microcontroller, besides Kalman filter requires more processing power than the complementary filter which will give more CPU cycles to the control algorithm.

The equation below (Eq II.1) [14] shows the implementation of the complementary filter in the flight controller, while figure II.10 illustrates the angles recorded from the accelerometer and gyroscope after interfacing the IMU with the microcontroller and the filtered angle after programming the complementary filter.

$$FiltredAngle = (1 - \alpha) * (FilteredAngle + GyroRate * dt) + \alpha * AccelAngle \quad \text{II.1}$$

Where α is defined by the equation (Eq II.2) $\alpha = \frac{\tau}{\tau - dt}$ II.2

Such that:

- **τ** is the time constant (How fast the readings will respond).
- **dt** is the sampling time.

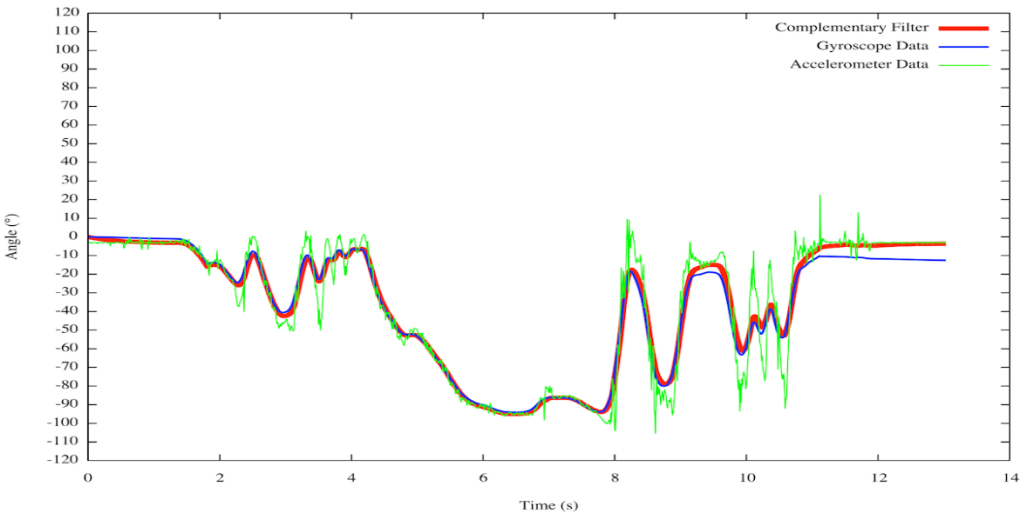


Figure II.10: *Filtered Angle by Complementary Filter*

C. Ultrasonic

The barometer is very inaccurate in altitude less than 10 meter due to humidity and ground air turbulence, for this reason the ultrasonic sensor (HC-SR04) (Figure II.11) is introduced to calculate the low altitude accurately.

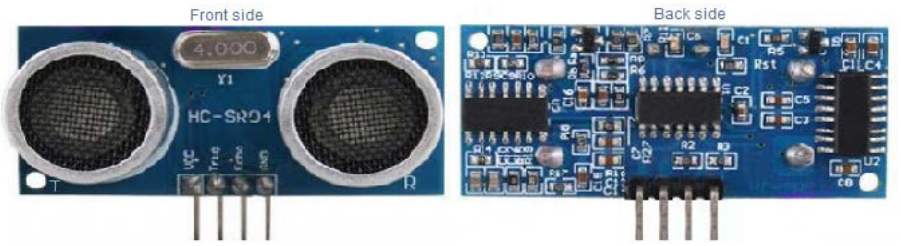


Figure II.11: *HC-SR04 Ultrasonic Sensors*

The ultrasonic sensor sends a trigger pulse of 10µs then the echo pin will receive the pulse (Figure II.12), the distance is then calculated from the time it takes for the pulse to leave and return to the sensor using the equation below (Eq II.3).

$$Distance(m) = \frac{SoundSpeed\left(\frac{m}{s}\right) * EchoPulseDuration(s)}{2} \qquad \text{II.3}$$

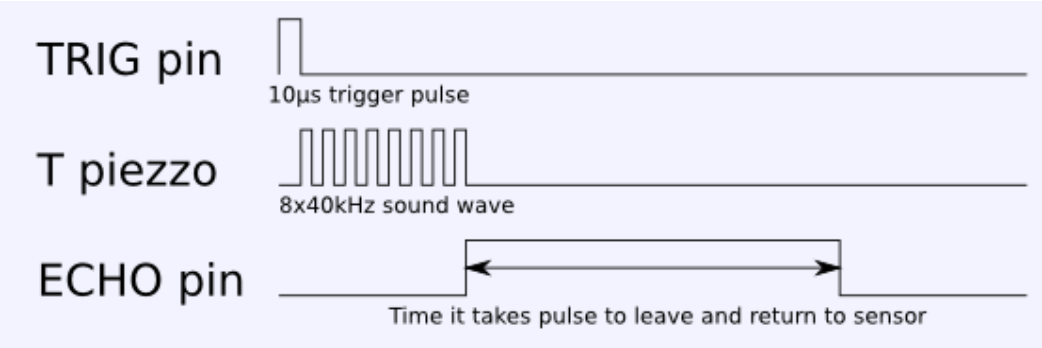


Figure II.12: *Trigger & Echo of Ultrasonic Timing Chart [16]*

D. GPS Sensor

Global Positioning Systems (GPS) [17] uses the signals sent by a number of satellites in orbit around the earth in order to determine the specific geographic location (latitude and longitude). In order to get an accurate GPS lock, the GPS chip should receive data from multiple satellites, usually the GPS is used when we have autonomous navigation.

The GPS used in the quadcopter is the *Ublox NEO-7M* (Figure II.14) which uses UART protocol [29], [30] to communicate with the microcontroller, the schematic of the GPS Module is shown in figure II.13.

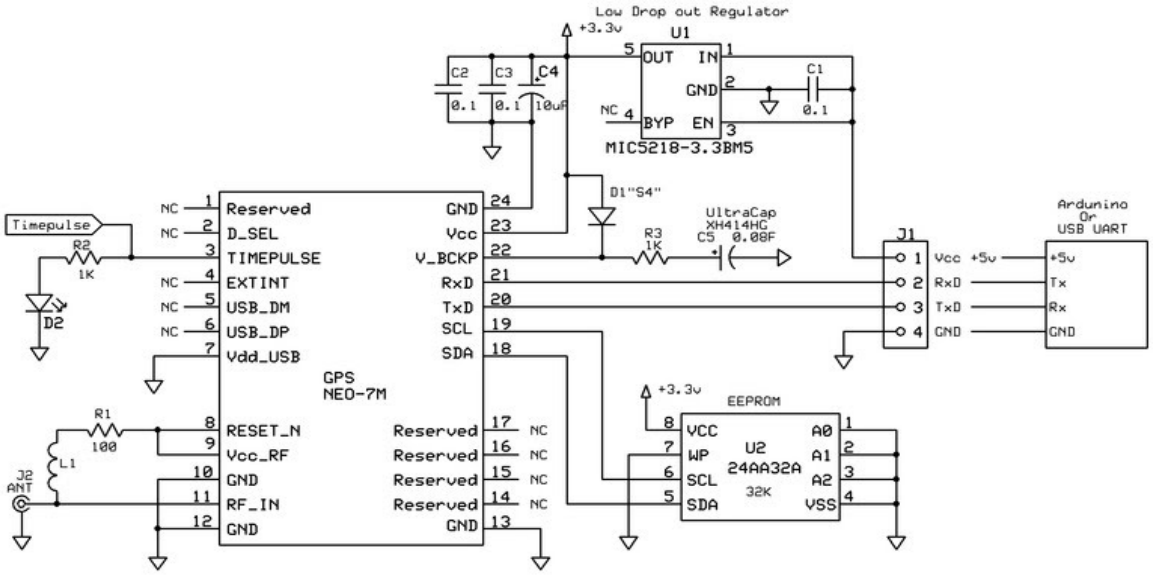


Figure II.13: *Internal Schematic Of the GPS Module [18]*

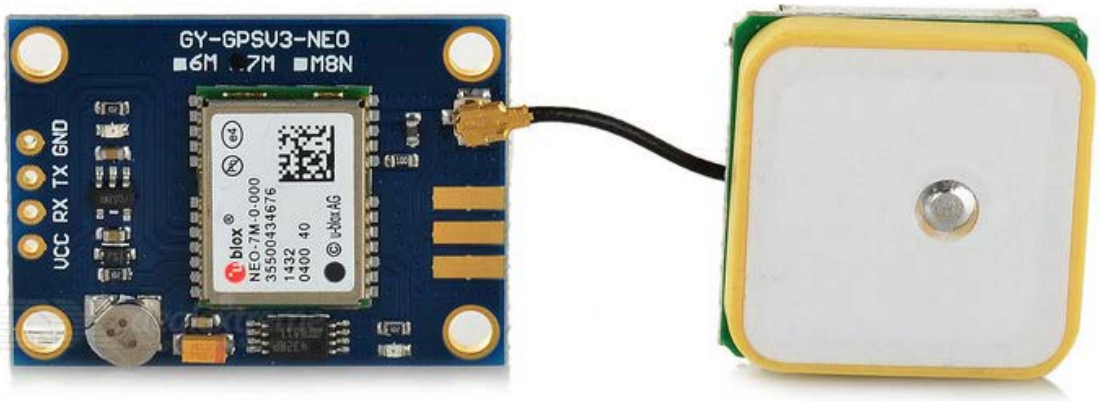


Figure II.14: *GPS Module NEO-7M*

II.3.1.2 Radio Control

Radio Control (RC) communication involves a hand-held RC transmitter and RC receiver. For UAVs, we need a minimum four channels to control the quadcopter:

- Pitch (which translates to forward / backward motion).
- Elevation (closer to or farther away from the ground).
- Yaw (rotating clockwise or counter-clockwise).
- Roll (to move left and right).

Most quadcopters involve handheld control rather than auto navigation alone, even for advanced applications professionals use both techniques in order to override the auto navigation at any moment.

The RC used with the quadcopter is the FlySky fs-t6 (Figure II.15) and it has 6 channels, thus two more free channels to exploit for other features such as automatic landing or takeoff, the RC uses the frequency 2.4GHz with GFSK [31] modulation.

Figure II.15: *FlySky FS-T6 Radio Control (Reciever and Transmitter) [19]*

The receiver of the RC was programmed with the microcontroller to read the signals from the transmitter over interrupt only in order to free the CPU while no change on the signal is happening.

II.3.1.3 Telemetry

Telemetry is an automated wireless communication process by which measurements and other data are collected at remote or inaccessible points and are subsequently transmitted to receiving equipment for monitoring [20], it is used also to transmit data to the remote point. The telemetry is used in order to monitor the quadcopter motion, altitude, GPS navigation and also to transmit the GPS coordinates for waypoint navigation or to use the joystick to control the quadcopter from the ground station directly, the module used is the RCTimer Telemetry that operates at 915MHz which provide a full-duplex link with Serial and USB Serial (Figure II.16).

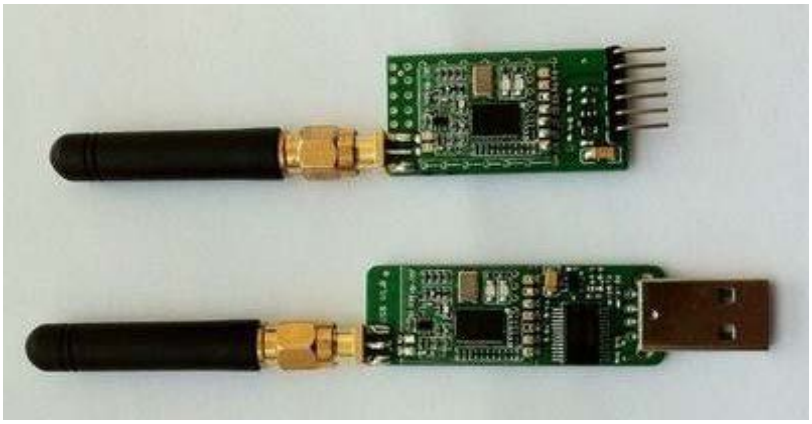


Figure II.16: *RCTimer 915MHz Telemetry*

II.3.1.4 Camera

The camera is considered as a sensor and an input to the ground station, because it is used to transmit live streaming video to the ground station and take the aerial images when needed. In order to have a good aerial shots, a good camera should be mounted to the quadcopter, in this project a small one with WiFi protocol (Figure II.17) is used in order to keep the weight of the quadrotor as light as possible and allow the data transmission to the GCS.

Figure II.17: *ActionCam W9 with WiFi*

II.3.2 The Output System

The output system is considered as the actuators that perform the flights and maneuvers of the quadcopter, this system is composed of two parts the propulsion system which combine the four brushless motors and there appropriate propellers the second part is the electronic speed controller which is the module responsible of driving the motors with the necessary electrical signals.

II.3.2.1 Brushless Motor

When it comes to propulsion and lifting the quadcopter, the motors used need to be well chosen in order to meet the requirements, for quadcopters we use only brushless motors due to their ability of generating more torque rather than brushed motors.

Brushless DC motor (BLDC) also known as electronically commutated motors (ECM) are synchronous motors that are powered by a DC electric source via an integrated inverter, which produces a three phase AC electric signal to drive the motor, for the quadcopters there are two types of BLDCs inrunners and outrunners.

The inrunner BLDC have higher RPM but less torque and the rotational core is contained within the motor same as a brushed DC motor. The outrunner BLDC is the most common one for quadcopters because they produce high torque which is required to lift the quadrotor, the principle of outrunner is that the outer shell that contains the magnets is the one that rotates (Figure II.18).

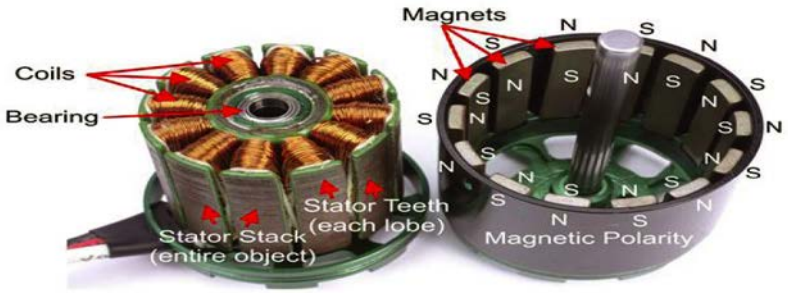


Figure II.18: *Outrunner Brushless Motor [21]*

The Brushless motor specifications comes like the format shown in Figure II.19.

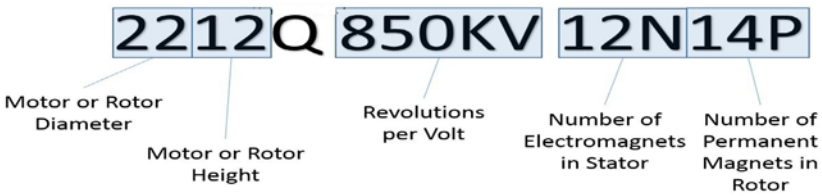


Figure II.19: *Brushless Motor Specifications Example*

The Kv number is the most important characteristic which define the number of revolutions per minute per volt; a 1000Kv BLDC can rotate at 1000 RPM per 1 volt, 10000RPM per 10 volts and so on.

As a rule of thumb the four motors on the quadcopter should lift twice the weight of the quadcopter this means that it should hover at half throttle, our quadcopter weighs 1Kg with all the components onboard, the motors must produce a 2Kg thrust, thus each motor should produce 500g of thrust with a given propeller specifications, the combination of propulsion system (motor & propeller) are given with the specifications of the motors [Appendix B].

After the calculations made for the quadcopter which includes the mass of the quadcopter and the type of flights that we want to have, the most appropriate brushless motor that suits the requirements in addition to the available propellers (10”x4.5”) is the one with 1000kv, the available motors were 980Kv.

II.3.2.2 Electronic Speed Controller

An Electronic Speed Controller (ESC) is the electronic circuit that allows the flight to control the speed and direction of a motor. The ESC must be able to handle the maximum current which the motor might consume, and be able to provide it at the right voltage. Most ESCs used in the hobby industry only allow the motor to rotate in one direction, though with the right firmware, they can operate in both directions [9].

The ESC provides also a 5V with the Battery Eliminator Circuit (BEC) [32] which can be used to power the other onboard components of the quadcopter, it is also characterized by the max current rating that can handle and usually this is defined by the max amps that the motor can draw, The ESC itself is quite a large circuit with microcontroller inside and inverters that control the generation of the three phase signals for the brushless motor, a simple view of the ESC is given in Figure II.20 and a full schematic is in [Appendix C].

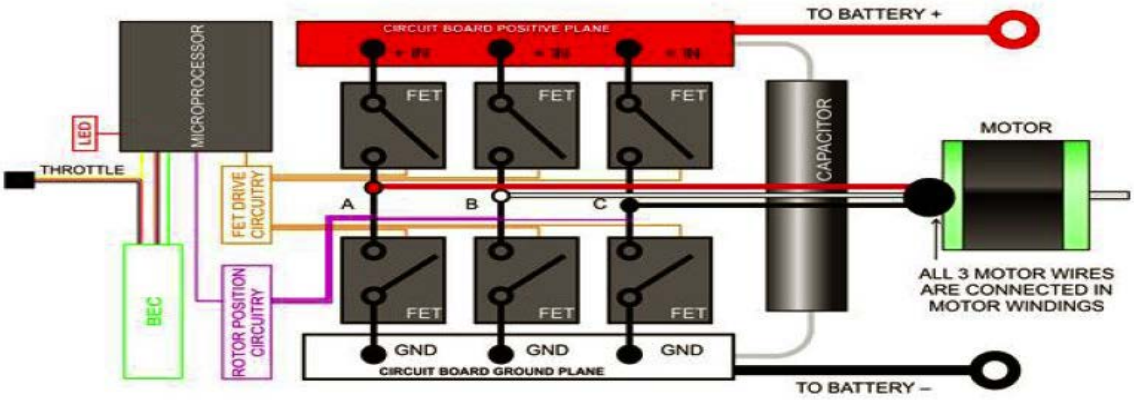


Figure II.20: *Simplified ESC Circuit [21]*

The ESC used in the quadcopter is the Mystery 30A with 5V BEC (Figure II.21), 30A is the max current rate that can support, since the max current draw in the motors available is 14.5A, so this is more than enough, and a full specification is in [Appendix C].

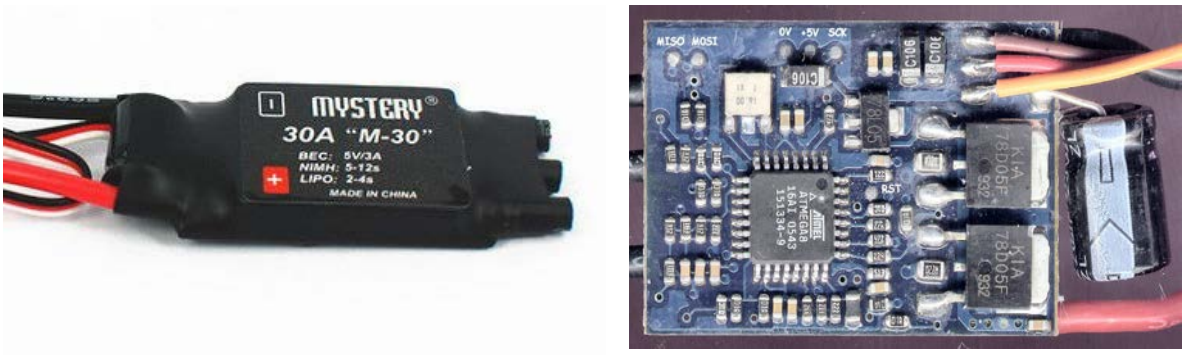


Figure II.21: *The ESC Mystery 30A*

II.3.3 Battery

The power in quadcopter is very important to consider because it's going to decide the amount of fly time, so the consumption should be minimized as maximum as possible, the battery will be chosen depending on this.

Batteries used in UAVs are now almost exclusively Lithium polymer (LiPo), they offer high capacity with low weight, and high discharge rates. The downsides are their comparatively higher cost and continued safety issues.

The battery is characterized by the voltage, the capacity and the discharge rate.

- Voltage depends on how many cells are in the battery, each cell is 3.7V and this is mentioned in the battery label as 1S, 2S, 3S and 4S, in quadcopters we use 3S batteries which means 11.1V, this fits the requirements of the motors and ESCs used.
- Capacity will define the flight time of the quadcopter, it is measured by amp-hours (Ah), the higher capacity the longer the flight time but also heavier pack which will be a drawback at certain point.
- Discharge rate in LiPo batteries is measured in C where 1C is the capacity of the battery, and it defines the maximum current that can be drawn from the battery, a battery with 30C of capacity 5000mAh can supply 30*5Ah = 150A but this will be for small amount of time, this discharge rate is needed because the motors in quadcopter consume high current.

The choice of the battery is made after the calculations of the power consumption of the quadrotor and the desired amount of flight time but in our case we have already a battery of 2200mAh, with this one we have found unfortunately short flight time with our system.

The motors in the quadcopter consumes the biggest portion of power in the system, from the specifications of our motor the current draw at 60% of throttle (which is the required for hovering) is 9.6A, so four motors draw 9.6*4=38.4A giving a 1.6A for the rest of the electronic components onboard, we result in 40A total, convert this to capacity we get 40*1000/60= 666mah consumed per minute. So the battery of 2200mah can give approximately 3.3 minutes of flight time at 60% of total power unfortunately.

II.4 Flight Controller

The flight controller is the brain behind all the quadcopter stability and maneuvers, it should achieve very fast computations in order to keep the quadcopter stabilized, and in our project we are using the Arduino Mega board due to its simplicity in reprogramming through USB interface also the availability of its free IDE.

II.4.1 Arduino Mega Board

The Arduino Mega 2560 [Appendix D] (Figure II.22) is an 8bit microcontroller board based on the ATmega2560 [Appendix D]. It has 54 digital I/O pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller [22].

Figure II.22: *Arduino Mega Board*

The Arduino simplifies also the circuitry of the whole system where the pin connection can be made easy with the different sensors and motors.

II.4.2 The Overall System Circuitry

After discussing all the necessary hardware parts from the sensor to the actuators to the microcontroller all these together make the quadcopter fly, the system circuitry that combine them together is shown in figure II.24.

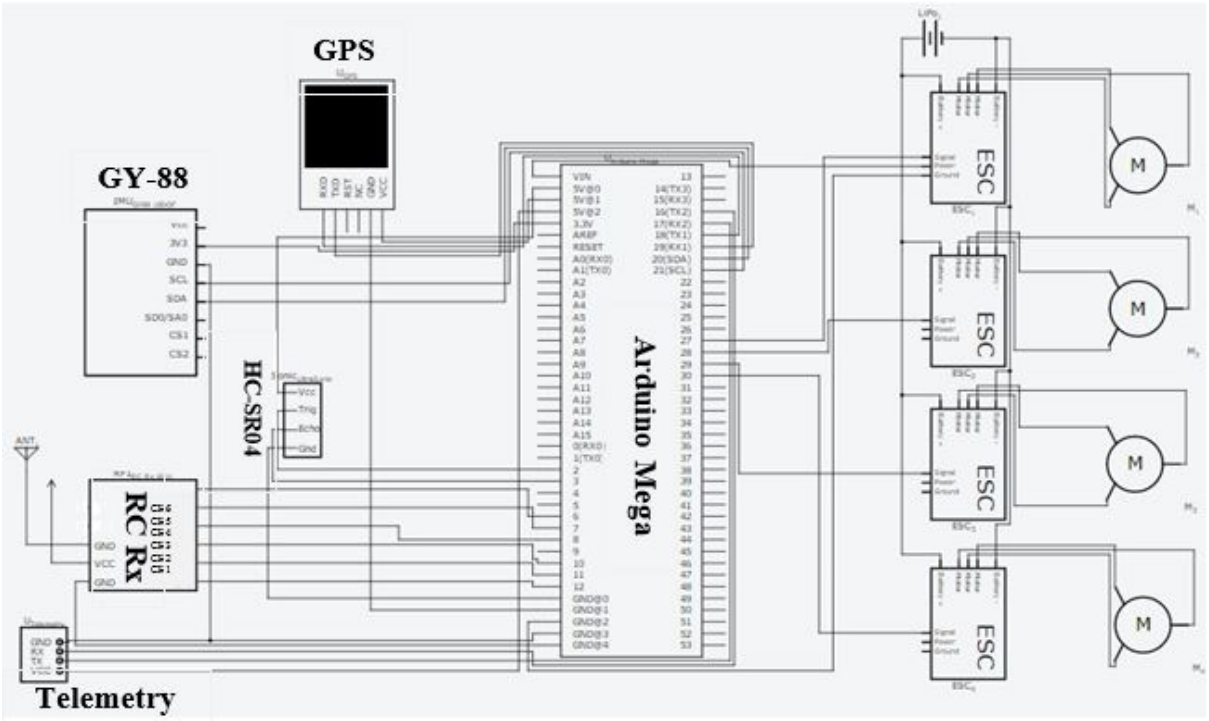


Figure II.24: *Overall System Circuitry Using 123d Circuit Design*

II.5 Software

The Arduino itself cannot do anything, we need to program all the sensors, ESC signals, and RC Signals and PID controllers and combine everything together in a well-defined algorithm that will create the overall Flight Controller.

The Algorithm of the flight controller is large, a global view flowchart (Figure II.23) is used to describe just the behavior of the system and its communication with the ground station.

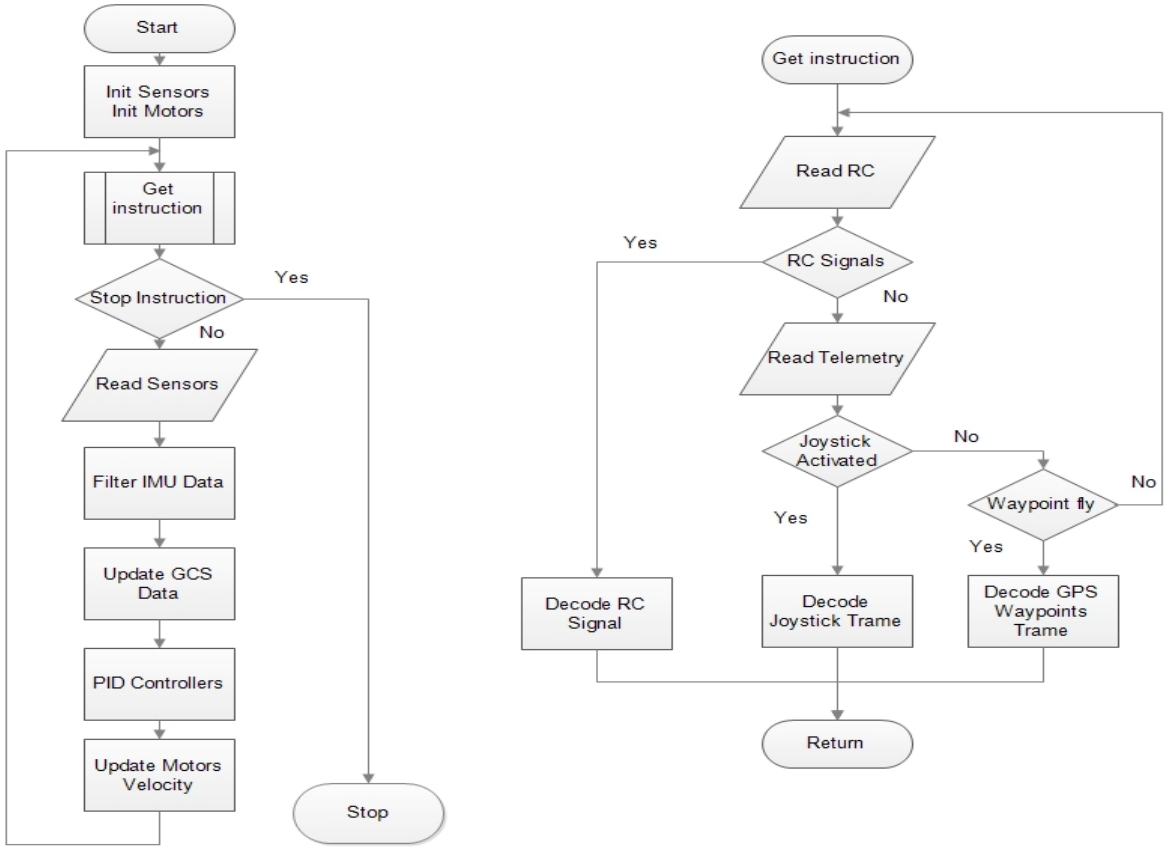


Figure II.23: *Flowchart for the Quadcopter System Using Edraw Max*

The flowchart shows the basic steps of the program that control the quadcopter from initialization to updating motors speed where the sub process “get instruction” has three priority levels, the highest priority is given to the RC and the lowest to the GPS waypoints this is a security measure in order to override to the manual control whenever the user wants to.

II.6 Quadcopter Full Specifications

After implementing all the quadcopter hardware and uploaded the appropriate software to the flight controller, the summary in table II.1 gives the full specifications and figure II.25 shows the real picture of the quadcopter.

Table II.1: *Quadcopter Full Specifications*

Component	Characteristics	Weight (g)
Frame	X configuration	340
GY-88	Acc, Gyro, Baro, Magneto	5
Ultrasonic	5m max range	5
GPS	Ublox NEO-7M	7
Flight controller	Arduino Mega	20
Telemetry	5Km Range	10
RC Receiver	1Km range	5
Propellers x 4	10”x4.5”	12x4
Motors x 4	980Kv	50x4
ESC x 4	30A	25x4
Battery	2200mAh	188
Payload (Camera)	ActionCam W9	50

The most important characteristics of the quadcopter are: the total mass which is 978g, the flight time of the quadcopter is approximately 4 minutes and finally the maximum piloting range is 1km with the RC and 5km with joystick.

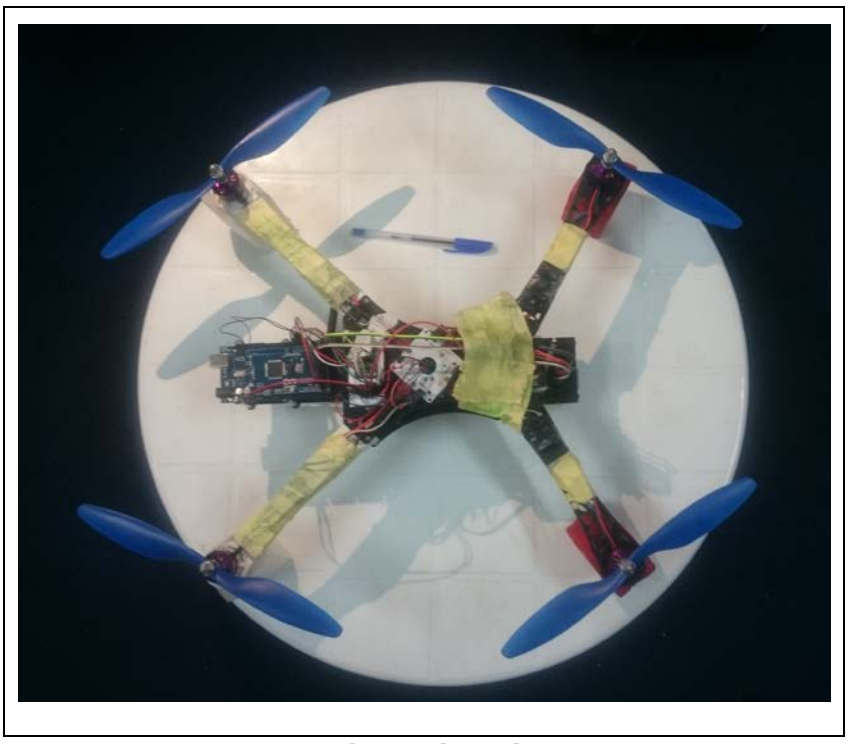


Figure II.25: *The Final Quadcopter Design*

II.7 Ground Control Station (GCS)

The Ground Control Station (GCS) is a land- or sea-based control centre that provides an interface and facilitates the human control of unmanned vehicles which can be used to control and monitor single or multiple UAV missions from take-off to landing at the same time. It also provides effective control of UAV payloads and allows sensor data transmitted from the UAV to be monitored [24].

The GCS can be deployed in a variety of different hardware configurations according to the needs: transportable desk mounted, portable or in a rugged standalone version for use out in the field.

The application interface is designed to be user-friendly and to have two main image processing application, high resolution image reconstruction that build a whole overview of a targeted area and object recognition that is able to reveal a pre-defined specific object approximate position on the swept zone.

All the parts that construct the GCS are developed with multithreading [23] in order to optimize the work and give concurrency between the different applications.

II.7.1 Ground Station Configuration

- **Software configuration**

In this work the software is composed of several parts shown in figure II.26.

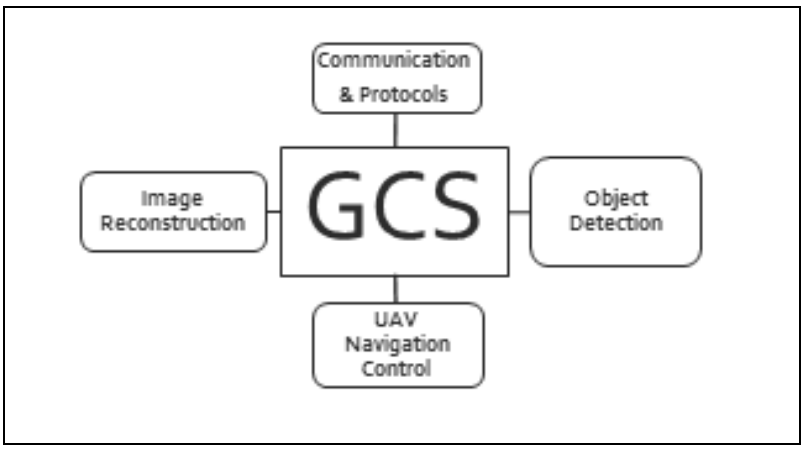


Figure II.26: *Ground Station Control Overview*

- **Hardware configuration**
  - Computer
  - Joystick
  - Rx/Tx Telemetry Module.

II.7.2 Communication & Protocols

In order for the UAV to work properly it must communicate with its Ground Control Station (GCS) in a real time fashion, where all the sensors must report the status of the flying machine to the monitor within the ground control station and this last should respond to the those data and must be able to handle that and make a full clear user friendly report to the operator.

As well the UAV must catch out the commands emitted by the ground station and process the data, so the communication method that is uses the Arduino on the UAV board that continuously streams data packets to the ground station via the Telemetry module.

In the other hand the camera on board continuously streams video to the ground station via the Real Time Streaming Protocol (RSTP) [25], in order to separate the load of communication on the channels and allow image acquisition.

II.7.2.1 Control Protocol

The control protocol from the GCS to the UAV is devide into two modes the manual mode where the control is done via the joystick, the second one is automatic where the flight is done via GPS waypoints navigation, the packet model used in the protocol is in figure II.27.

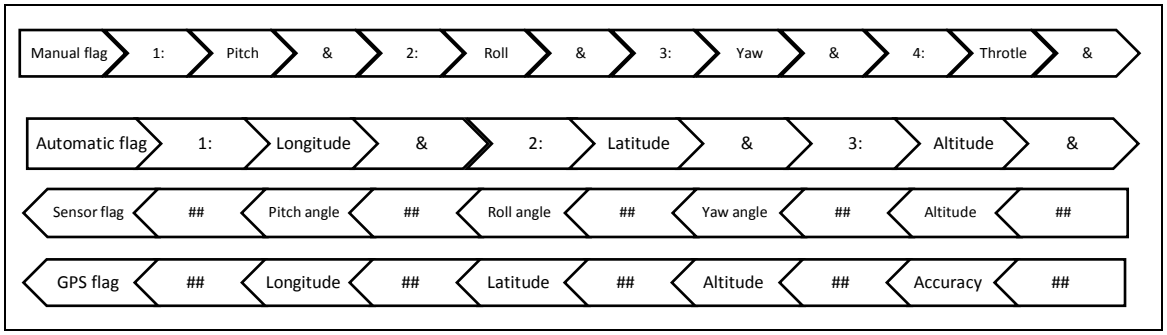


Figure II.27: *Rx and Tx Control Protocol*

II.7.2.2 Image Acquisition Protocol

The UAV requires a real time image acquisition in order to do the image processing in the GCS, the protocol used is the RTSP [26] to stream video from the camera onboard to the GCS and take images when needed. The reason this protocol is used because it operates over the WiFi channel which gives a good data transmission speed compared to the wireless serial telemetry module.

II.7.2.3 UAV Navigation from GCS

This work consists of the Quadrotor controlled from ground station, where it is possible to have two types of navigation modes automatic navigation using GPS coordinates and manual control using joystick.

It is necessary for the UAV automatic flight mode to localise itself in its outdoor environment, as well as it is important to follow the pre-registered path, for that reason a continuous communication with the ground station is employed so such that the UAV position is tracked from its starting location through every key location on the path. All the key locations are stored in a database, and the ground station transmit each time the next position where the quadcopter should be, if the UAV succeed reaching the desired position, the GCS transmits the next position coordinates, else it can be turned to the manual flight mode, a full view of the GCS interface is shown in figure II.28.

In order to get a better fly performance and endurance the UAV path must be typical, such that the drone should pass throw the different locations where it supposed to fly with the shortest path with taking in consideration all the points in the database.

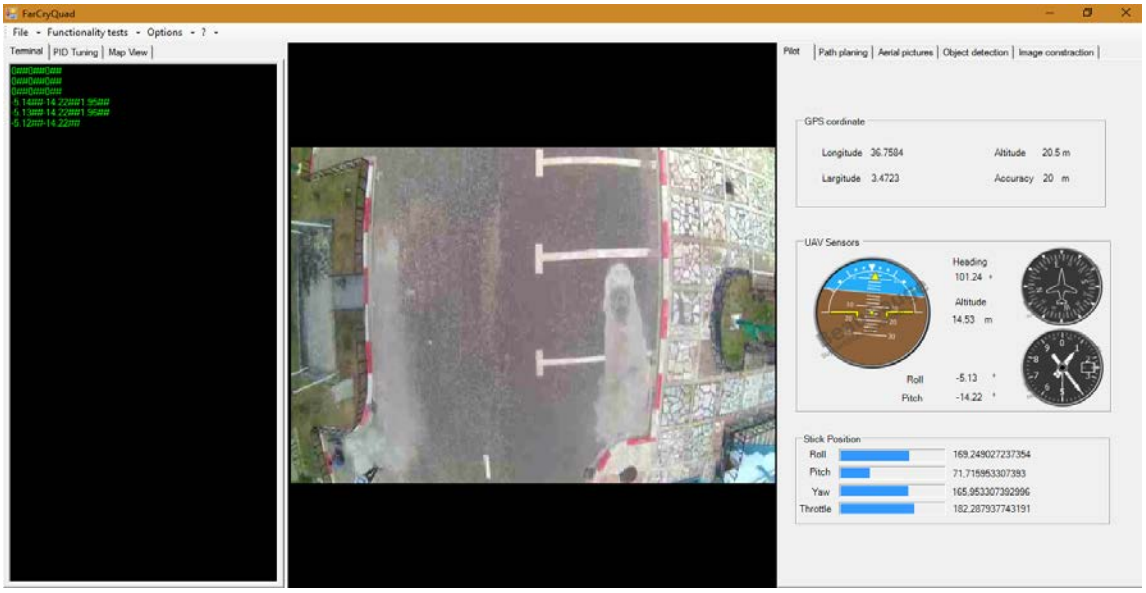


Figure II.28: GCS Interface

II.8 Summary

In this chapter we covered the hardware and software implementation of the quadcopter in the first part we combined all the electronic and mechanical components together and programmed the algorithm that allow the quadrotor to fly stable, in the second part we developed a ground control station in order to monitor and control the UAV to perform the image processing applications that will be covered in the next chapters.

Chapter III

Image Reconstruction

This chapter goes with the image reconstruction, based on the Quadrotor image acquisition. The output image is rebuilt after identification of the feature in a set of images, then the process goes throw the image matching algorithms to the final panorama construction.

III.1 Introduction

The quadcopter discussed in the previous chapters is used to achieve high resolution image reconstruction technique, the main problem with this feature is the automation of the process, several methods have been used before for this issue where those approaches have used human input or restrictions on the image sequence in order to establish matching images., while the need in this project is to implement an algorithm that can handle the main problematic of the image reconstruction, the Automatic Panoramic Image Stitching using Invariant Features algorithm is delegated for this topic.

The Automatic Panoramic Image Stitching using Invariant Features method proposed by Matthew Brown and David G. Lowe [34] is insensitive to the ordering, orientation, scale and illumination of the input images. It is also insensitive to noise images that are not part of a panorama, and can recognise multiple panoramas in an unordered image dataset.

The automatic panoramic image stitching using invariant features method include six successive steps to reconstruct the final panoramic image, feature matching, Image matching, bundle adjustment, automatic panorama straightening, gain compensation and finally multi-band blending.

III.2 Image Acquisition

The image acquisition is done with the camera onboard of the UAV. The aerial shots must have an overlapping between them, such that some features should appear in more than a single capture, so that the reconstruction can take a place. For this reason, the GPS is selected for our application.

The quadcopter hovers above specific points in order to make the required visualization of the scenes to take the captures that will form a final panorama.

The camera carried by the quadrotor streams real time view to the ground control station where the functionalities of the camera are controlled by. The GCS transmits the operator requests to the camera, then this last will respond with the suitable action (Pause, Stop or Capture). The GCS triggers the capture on the specific location, stored on the database as GPS waypoint however the GPS have some drawbacks on the image capturing due to its accuracy discussed on the next part.

III.2.1 GPS System for Image Capturing

GPS uses a one-way ranging technique from GPS satellites, which are also broadcasting their estimated positions. Using triangulation techniques, the location of the receiver can be retrieved.

Latitude, longitude, altitude and a correction to the user's clock are determined using the satellites in an appropriate geometry. The GPS receiver coupled with the computer receiver returns elevation angle and azimuth angle between the user and the satellite.

The use of GPS in the application is to give the exact coordinates to the quadcopter where it must trigger an image capture for further processing, but the mechanism that allows the quadrotor to get its position is with accuracy of 24 to 6 meters (figure III.1), which is critical to the aerial imaging. The drawback of this technique is the GPS error accumulation, and the unpredictable variations.

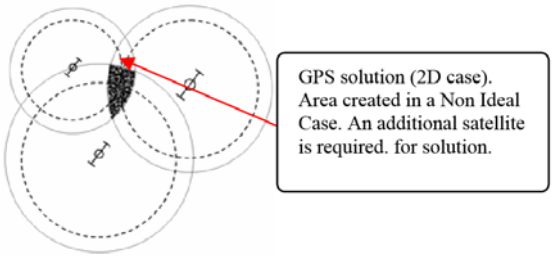


Figure III.1: *Ideal and Non Ideal Traingulation Accuracy*

The GPS error affects directly the image overlapping since any error within the position of the perspective leads to an undesirable shift in the image captured from the UAV.

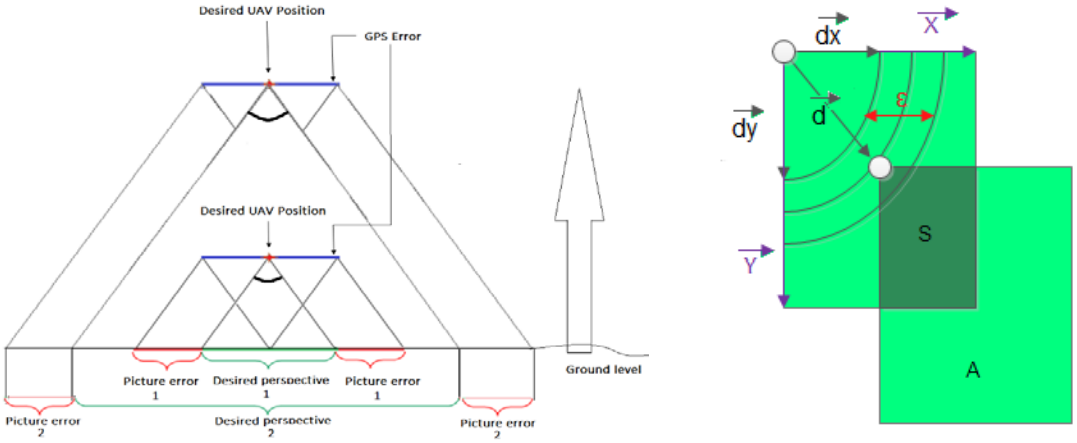


Figure III.2: *Impact of altitude over the GPS error & Image Perspective*

In order to solve the problem of the GPS error over the image capturing, it is necessary to use high altitude for the navigation and area swapping.

The figure III.2 shows the overlapping between two successive images where “A” is the area covered by the camera perspective at a constant altitude and “S” is the overlapping between the two scenes, the displacement vector “d” plus the GPS error “ε” ($d + \frac{1}{2}\varepsilon$) should not exceed a certain value "$d + \varepsilon$" which is the real displacement from the first coordinate to the second exact coordinate. While the desire perspective named “A” is relative to the altitude and the camera lens characteristics.

III.3 Feature Matching

The first step in the panoramic reconstruction algorithm is to extract and match Scale Invariant Feature Transform (SIFT) [35] [36] (Figure III.3) features between all the images. SIFT features are located at scale-space maxima/minima of a difference of Gaussian function [35]. At each feature location, a characteristic scale and orientation is established. This gives a similarity-invariant frame in which to make measurements (Figure III.4).

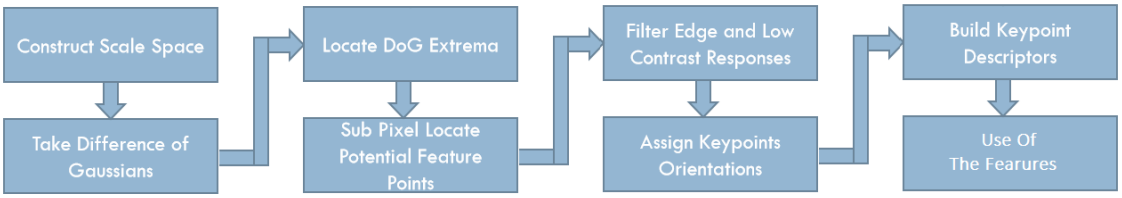


Figure III.3: *SIFT Algorithm Block Diagram*

Figure III.4: *Extracted Sift Key points from an Image taken at IGEE -Boumerdes*

III.4 Image Matching

At this level the goal is to find all corresponding matching points between the input image set (Figure III.5) which leads to the identification of the overlapping. Connected images with a successful matching will later become panoramas.

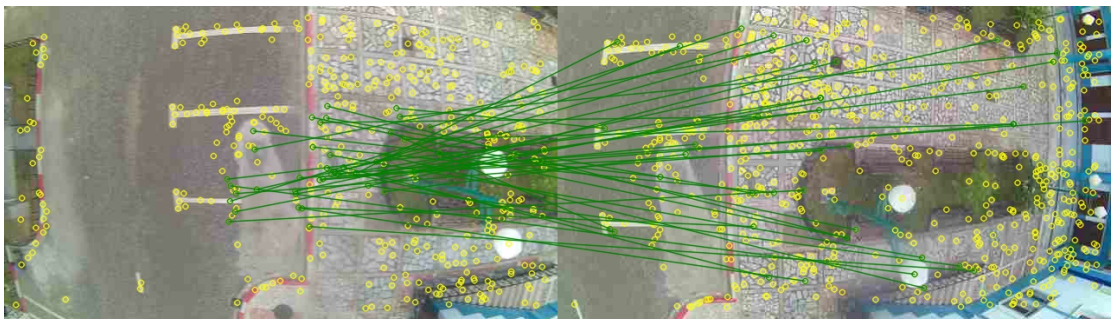


Figure III.5: *Image Matching Example*

Since each image can have more than one match with the rest of the input data set, the selection of the best corresponding match become a serious issue. For this reason, it is only necessary to match each image to a small number of overlapping images in order to get a good results for the image geometry.

The feature matching step identify the images that have a large number of matches between them. A constant number images is defined, identify by the greatest number of feature matches to the current image, as potential image matches.

First step of selection is the use of Random Sample Consensus (RANSAC) algorithm [37] to select a set of inliers that are compatible with a homography [38] between the images. Next the probabilistic model [34] is used to verify the match. The RANSAC and probabilistic model are applied iteratively for images with the most features matching. The Figure III.6 shows RANSAC inliers in blue on two images with overlapping between them where the yellow are the Features extracted with the SIFT.

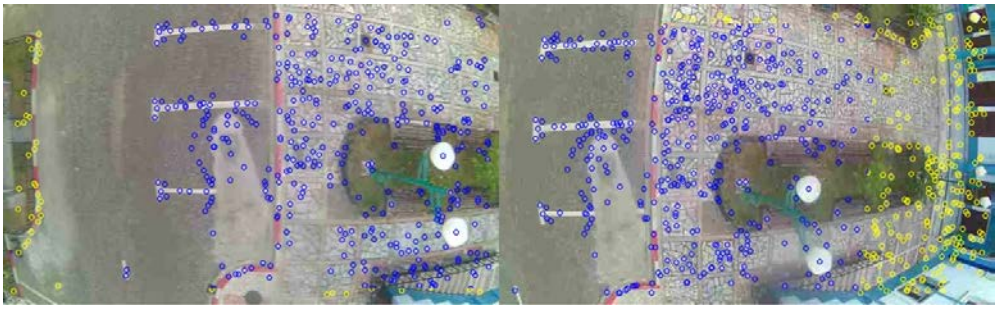


Figure III.6: *RANSAC Inliers & SIFT Features*

III.5 Bundle Adjustment

Given a set of geometrically consistent matches between the images, we use bundle adjustment [34] to solve for all of the camera parameters jointly. This is an essential step as concatenation of pairwise homography would cause accumulated errors and disregard multiple constraints between images. Images are added to the bundle adjuster one by one, with the best matching image being added at each step. The new image is initialised with the same rotation and focal length as the image to which it best matches. Then the parameters are updated using Levenberg-Marquardt Algorithm [39].

Figure III.7 shows the pairwise images with homography between them and figure III.8 shows the final result after bundle adjustment.

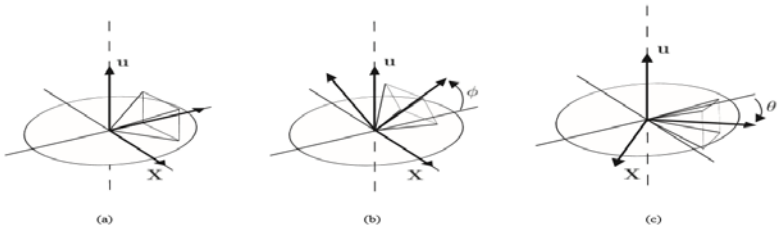
Figure III.7: *The Matching and Homography Between Two Images*

Figure III.8: *Final Result after Bundle Adjustment*

III.6 Automatic Panorama Straightening

Image reconstruction from feature matching to the bundle adjustment can give a relative rotation between the camera's views, which causes a wavy effect in the output panorama (Figure III.9), which is undesired in the final result, the correction of the wavy output and automatically straighten the panorama by making use of a heuristic about the way panoramic images are taken [34].

The idea is that it is rare to twist the camera relative to the horizon, so the camera $\vec{X}$ vectors (horizontal axis) typically lie in a plane (figure III.9). By finding the null vector of the covariance matrix of the camera $\vec{X}$ vectors, we can find the $\vec{u}$ vector (normal to the plane containing the camera centre and the horizon) with the function $(\sum_{i=0}^{n} X_i X_i^T)u = 0$ [34].



Good Alignment Tilt Rotation

Figure III.9: *Finding the Vector $\vec{u}$ [1]*

Applying a global rotation such that the $\vec{u}$ vector is vertical removes the wavy effect from the output panorama as shown in (figure III.10).

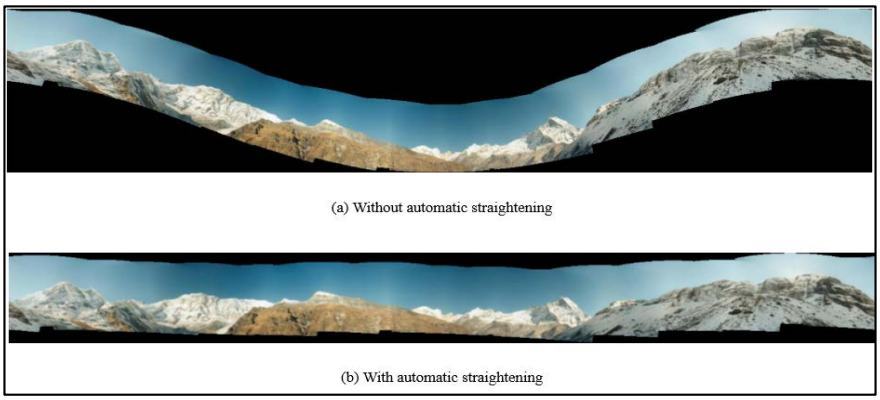


Figure III.10: *Output Panorama With & Without Automatic Straightening [34]*

III.7 Gain Compensation

The previous descriptions deals with the methods for computing the geometric parameters (orientation and focal length). Here the photometric parameter is introduced, namely the overall gain between images. This is set up in a similar manner, with an error function defined over all images. The error function is the sum of gain normalised intensity errors for all overlapping pixels [34] (Figure III.11).

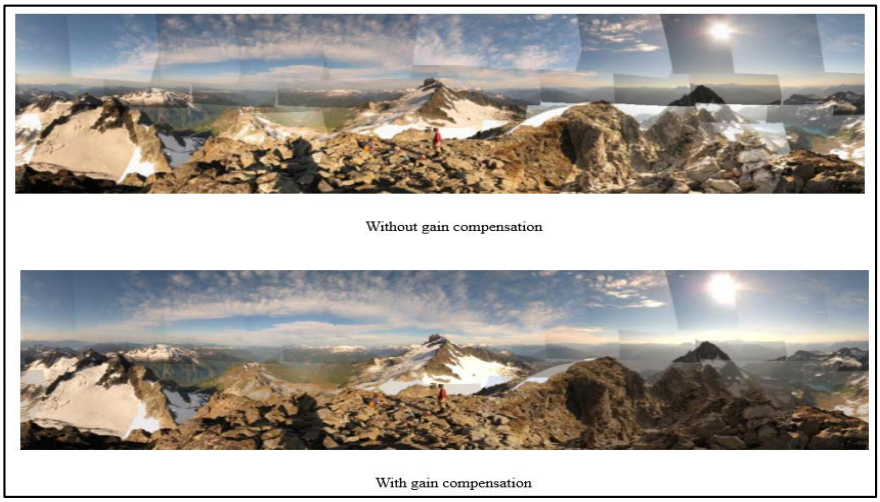


Figure III.11: *Image Reconstruction With and Without Gain Compensation.* [34]

III.8 Multi-Band Blending

Ideally each sample (pixel) along a ray would have the same intensity in every image that it intersects, but in reality this is not the case. Even after gain compensation some image edges are still visible due to a number of unmodelled effects, such as vignetting (intensity decreases on the edge of the image), parallax effects due to unwanted motion of the optical center, misconstruction errors due to mismodelling of the camera, radial distortion and so on. Because of this a good blending strategy is important. Figure III.12 shows the effect of Gain Compensation and Multi-Band Blending.

Figure III.12: *Image Reconstruction with Gain Compensation & Multi-Band Blending* [34]

III.9 Algorithm Implementation

The algorithm of the automatic panoramic image stitching using invariant features was done on the ground station level, the flowchart in Figure III.13 illustrates its implementation.

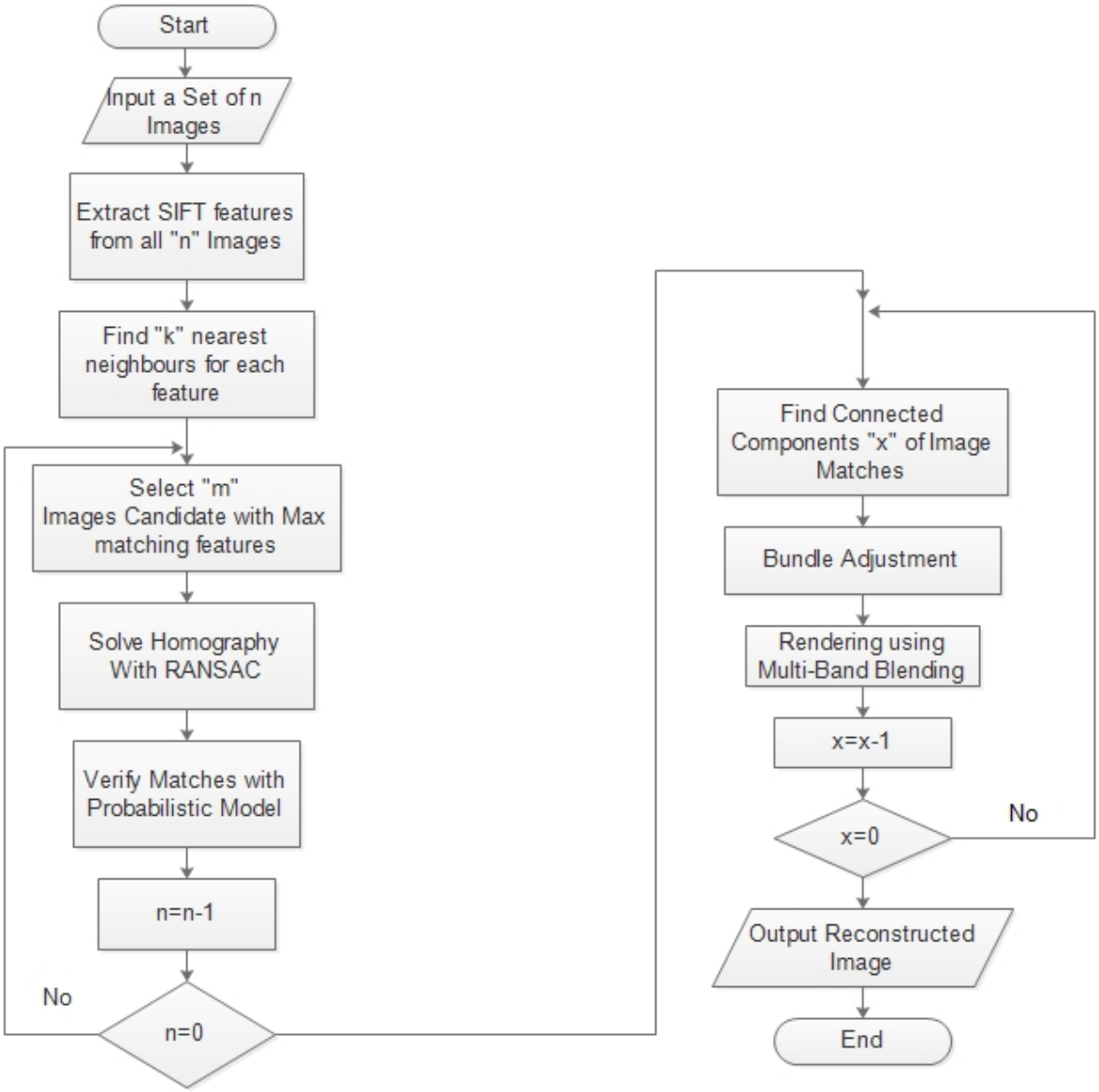


Figure III.13: *Automatic panoramic image stitching using invariant features algorithm*

III.10 Results and Discussion of the Automatic Panorama Image Stitching Algorithm

After the implementation of the algorithm and taking different aerial images in different areas with the quadcopter, the results in figure III.15 shows the reconstructed image from the four images in figure III.14 taken at IGEE.

Figure III.14: *Images Taken at IGEE -Boumerdes*

Figure III.15: *Reconstructed Images from IGEE –Boumerdes Aerial View*

Figure III.17 shows the reconstructed image from five images (Figure III.16) taken at Bouhri Boualam Campus Stadium ex-INH, Boumerdes.

Figure III.16: *Images Taken at ex-INH Campus Stadium,Boumerdes*

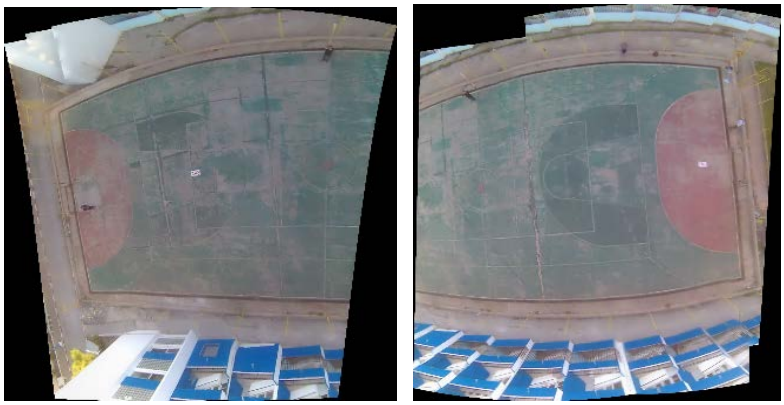


Figure III.17: *Reconstructed Images from ex-INH Campus Stadium Aerial View*

The results in figure III.15 had a better reconstruction comparing with the results of the figure III.17 because the flight performed in the first one is more stable in piloting while the flight on the second scene was performed with high velocity this creates two origins for the camera perspective. The final reconstruction have more visible distortion when coupling the two origins perspectives as shown in the figure III.18.

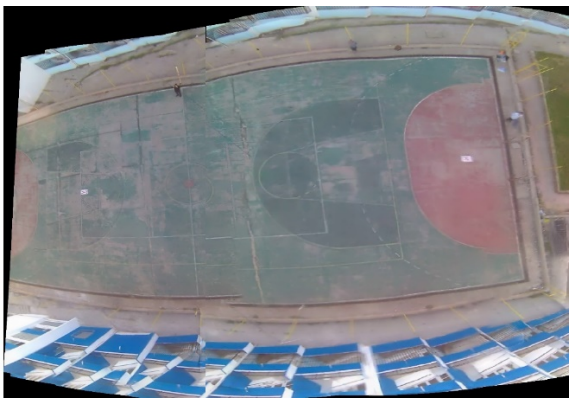
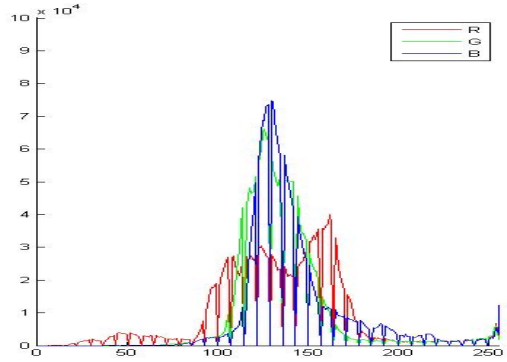
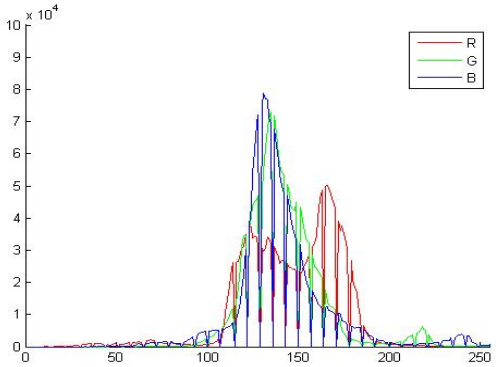
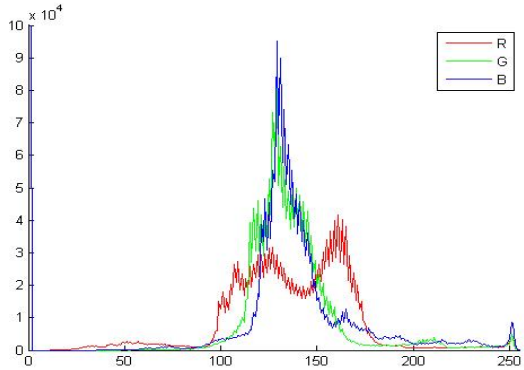


Figure III.18: *Final reconstructed Images from e-INH Campus Stadium Aerial View*

The fisheye lens distortion [40] on the camera used affect the input images thus affects the output slightly. In order to assess well the difference existing between the stitched images and the concatenation of its inputs, we have used tri-color image histograms to perform a thorough comparison. The table III.1 highlights the stitching characteristics of two images (P1) and (P2), where the result is revealed in the picture (P3). The histogram of the three pictures is named (H1), (H2) and (H3) respectively. From the same table each image is associated with its size and number of pixels within. The histograms on the Table III.1 as well as the number of pixel indicate a successful stitching .Where the values on the histogram H3 fits exactly with the value of the H2 and H1. So the shape of the plots on the histograms are conserved as well as the brightness of the scene is preserved. A quick variation appears on the plots of the histogram H3 with is translation of the slight blurring on the result image.

Table III.1: Histogram and Characteristics Table of Image Reconstruction

Histogram	Picture	Size	Pixels
 (H1)	 (P1)	1090 x 1920	2,092,800
 (H2)	 (P2)	1090 x 1920	2,092,800
 (H3)	 (P3)	1184 x1804	2,135,936

III.11 Summary

This chapter covered the problematic of the image reconstruction in automatic manner , with a UAV image acquisition , the Automatic Panoramic Image Stitching algorithm shows a robustness over the different transformations (Rotation , Scale) and the camera optical characteristics (Fisheye distortion , Brightness) as well as the optical origin.

Chapter VI

Object Detection

This chapter comes to highlight the object detection, which is based on the invariant feature extraction of both the scene and the target. This chapter comes as an extension of the image reconstruction. This chapter also experiment the detection with different image perspectives.

IV.1 Introduction

Object detection is the process of finding instances of real-world objects such as faces, cars, and buildings in images or scenes. Object detection algorithms typically use extracted features and learning algorithms to recognize instances of an object category. It is commonly used in applications such as image retrieval, security, surveillance, and automated vehicle systems. Many solutions have been proposed for this issue, this work focus on the SURF algorithm [1] that best fits the requirement of the object detection from aerial perspective (Rotation and Scalability) as well as the computations time consumption.

The SURF (Speed up Robust Feature), feature based algorithm is a scale and rotation invariant detector and descriptor. This detection should ideally be possible when the image shows the object with different transformations, mainly scale and rotation, or when parts of the object are locked. The processes can be divided into three global steps detection, description and matching.

IV.2 Detection and Description

Detection automatically identify interesting features, interest points this must be done robustly. The same feature should always be detected regardless of viewpoint.

The SIFT algorithm is a successful approach to feature detection introduced by Lowe [2]. The SURF-algorithm [1] is based on the same principles and steps, but it utilizes a different scheme and it should provide better and faster results. In order to detect feature points in a scale invariant manner, SIFT uses a cascading filtering approach. Where the Difference of Gaussians (DoG) [3], is calculated on progressively downscaled images. In general the technique to achieve scale invariance is to examine the image at different scales, scale space, using Gaussian kernels. Both SIFT and SURF divides the scale space into levels and octaves. An octave corresponds to a doubling of σ, and the octave is divided into uniformly spaced levels where σ is the Gaussian kernel variance (Figure IV.1).

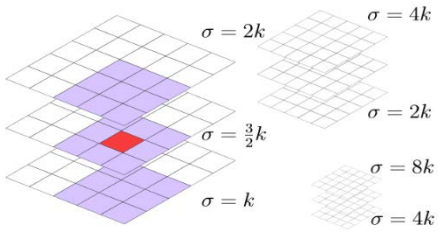


Figure IV.1: *Gaussian Scale Space Pyramid* [1]

Both approaches SIFT and SURF builds a pyramid of response maps, with different levels within octaves. A response map is the result of an operation on the image. The interest points are the points that are the extrema among eight neighbors in the current level and its 2×9 in the level below and above. This is a non-neighbors maximum suppression in a 3×3×3 neighborhood, the relation between levels, octaves and neighborhood is illustrated (Figure IV.1). SURF uses the Hessian matrix because of its lesser computation time and good accuracy. Here the reliability is on the Hessian matrix for determining both the scale and the location of the interest point. This algorithm is also based on scale space theory. Here, the local maxima are estimated using Hessian matrix "**H**". Given a point "**x**", in an image "**I**", the Hessian matrix H (x, σ) (Eq IV.1) [1] in "x" at scale "σ" is defined as follow

$$H(\mathrm{x},\sigma) = \begin{bmatrix} L_{xx}(\mathrm{x},\sigma) & L_{xy(\mathrm{x},\sigma)} \\ L_{xy(\mathrm{x},\sigma)} & L_{yy(\mathrm{x},\sigma)} \end{bmatrix} \quad \text{IV.1}$$

Where:

$$L_{xx}(\mathrm{x},\sigma) = \mathrm{I}(\mathrm{x}) * \frac{\partial^2}{\partial \mathrm{x}^2}\mathrm{g}(\sigma) \quad \text{IV.2}$$

$$L_{xy}(\mathrm{x},\sigma) = \mathrm{I}(\mathrm{x}) * \frac{\partial^2}{\partial \mathrm{xy}}\mathrm{g}(\sigma) \quad \text{IV.3}$$

$L_{xx}(\mathrm{x},\sigma)$ (Eq IV.2) [1] and $L_{xx}(\mathrm{x},\sigma)$ (Eq IV.3) [1] are the convolution of the image with the second derivative of the Gaussian. The heart of the SURF detection is non-maximal-suppression of the determinants of the hessian matrices. The convolutions is very costly to calculate and it is approximated and speeded-up with the use of integral images. An Integral image I(x) (Eq IV.4) [1] is an image where each point x stores the sum of all pixels in a rectangular area between origin and x .

$$I(x) = \sum_{i=0}^{i\leq x}\sum_{j=0}^{j\leq y} I(x,y) \quad \text{IV.4}$$

To enhance the computing speed, the box filter approximation is taken instead of Gaussian filter. The multi-directional box filters are shown in figure IV.2.The determinant of Hessian matrix, ΔH (Eq IV.5) can be reduced to:

$$\Delta\mathrm{H} = D_{xx}\,D_{yy} - (\,\omega D_{xy})^2 \quad \text{IV.5}$$

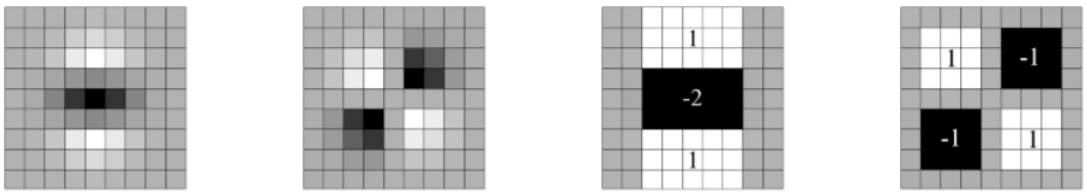


Figure IV.2: *The discretized and cropped Gaussian second order partial derivatives filters* [1]

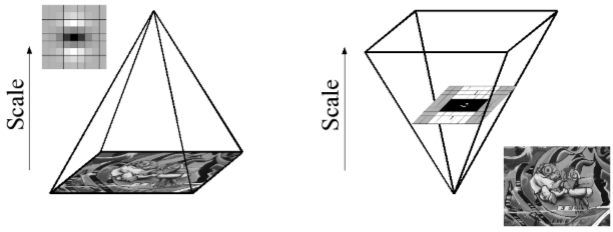
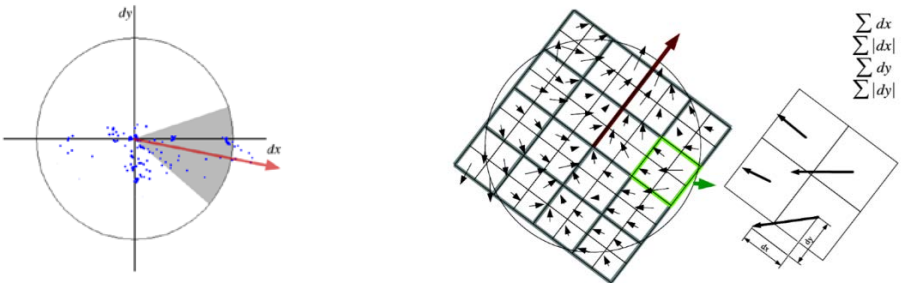


Figure IV.3: *Iterative image size reduction and the use of integral images.*[1]

The illustrated kernels corresponds to an σ of 1.2 and are the lowest scale that the SURF algorithm can handle. When using the approximated kernels to calculate the determinant of the Hessian matrix - we have to weight it with ω in Eq IV.5, this is to assure the energy conservation for the Gaussians. The ω term is theoretically sensitive to scale but it can be kept constant at 0.9 [1].Where ω is threshold is set for non-maxima suppression to detect the extreme points. The stable feature points are chosen by comparing with the neighboring values by the interpolation operation in scale space (figure IV.3). Gaussian weighting coefficients are merged with Haar wavelet responses [1] to extract the interest points. The Haar wavelet responses in vertical direction (dy) and in horizontal direction (dx) are summed up along with the absolute value of the response as $v = \{\sum dx, |\sum dx|, \sum dy, |\sum dy|\}$ [1].

The figure IV.4 shows keypoints vector orientation and scalar assignment.



Orientation assignment Description vector assignment

Figure IV.4*: SURF Feature Descriptor* [1]

IV.3 Matching

The SURF algorithm uses the sign of Laplacian for underlying interest point, it adds no computation cost since it is already computed during the detection.

$$\nabla_2 L = tr(H) = Lxx(x,\sigma) + Lyy(x,\sigma) \qquad \text{IV.6}$$

The Laplacian is the trace of the hessian matrix (Eq IV.6) [1] and when calculating the determinant of the hessian matrix these values are available. It is a matter of storing the sign. The reason to store the sign of the Laplacian is that distinguishes between bright blobs on dark backgrounds and vice versa. It is only necessary to compare the full descriptor vectors if they have the same sign, which can lower the computational cost of matching.

IV.4 Surf Algorithm Implementation

The algorithm of the SURF was implemented on the ground station, the diagram in Figure IV.5 illustrates its main blocks.

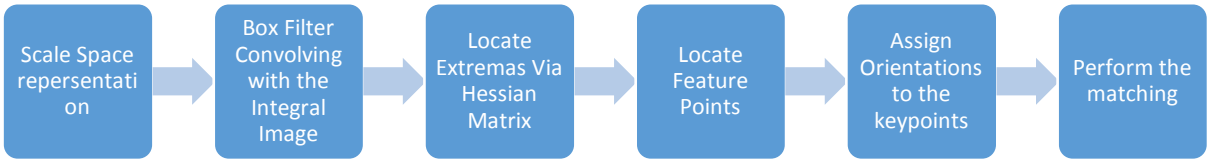


Figure IV.5: *Automatic Panorama Stitching Algorithm*

In the previous parts the theoretical background for the SURF algorithm is revealed, in this segment a details of the implementation made is presented. The implementation uses the Emgu.CV framework, which is a VB.net framework for computer vision and interpretation of the Open.CV for the windows operating system. It ships with it's the implementation of SURF and SIFT and several other computer vision algorithms. It was chosen as it provides a good low level routines for working with images and easy loading and saving of different image formats. The focus when implementing has been on getting to a state where it was possible to identify objects with aerial captures.

IV.5 Results and Discussion of the SURF Algorithm

After the implementation of the SURF algorithm and taking different aerial images, the test was performed by grabbing an object from an aerial view shown on figure IV.6 and perform a detection with the SURF algorithm implemented within our application. The figure IV.7 shows the object detection performed from different camera perspective of the target, the images have been taken at Bouhri Boualam Campus Stadium ex-INH -Boumerdes, where the different transformation views of the images are natural.

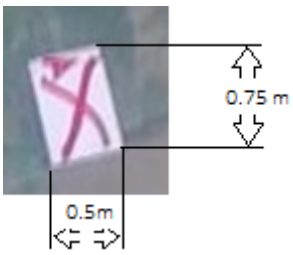
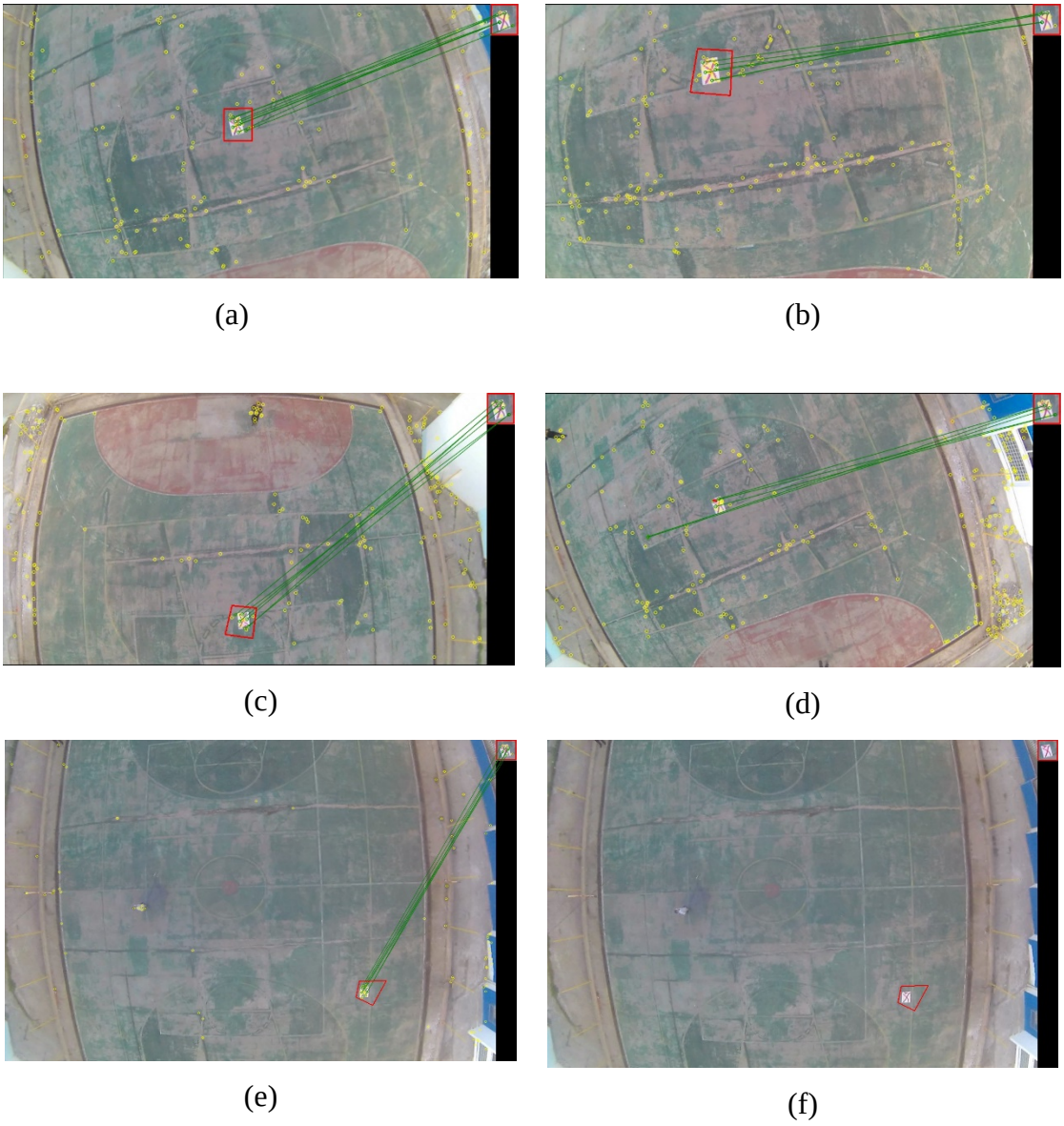


Figure IV.6: *The Object Picture*

First a target is locked from the picture (a) which shows the ideal match. (b) represents the target identification with a slight shift of the object to the left, where in (c) a successful identification is performed with a rotation. After increasing the altitude the algorithm still able to localize the target as shown in (d). For more credibility the target has been moved to another location on the scene and the system positively recognize it in (e) and (f).



(a) (b)

(c) (d)

(e) (f)

Figure IV.7: *Results of the Object Detection*

An extra test of detection have been performed, the object in this investigation is located on stitched scene as shown later on the figure IV.8; where the object is viewed clearly in (a). (b) shows the recognition made with the SURF (The yellow marks represent the SURF key points) and (c) illustrates the original stitched picture.

The table IV.1 shows more details about the detections tests located in figure IV.7.

Table IV.1: *Object detection using the SURF characteristics*

Picture	Real target position	Detected target position	Number of features		Matching time
			Detected	Matched	
(a)	X=490.00 ;Y=255.00	X=493.67 ; Y=253.15	168	9	0.1347474 s
(b)	X=347.00; Y=139.00	X=350,78; Y=142,02	214	4	0.192.8479 s
(c)	X=516.00,Y=492.00	X=504,62;Y=482,873	192	6	0.1838569 s
(d)	X=362.00;Y=231.00	X=354,11; Y=226,01	289	6	0.1978807s
(e)	X=978.00,Y=454.00	X= 938,15; Y=373,85	98	5	0.2725167 s

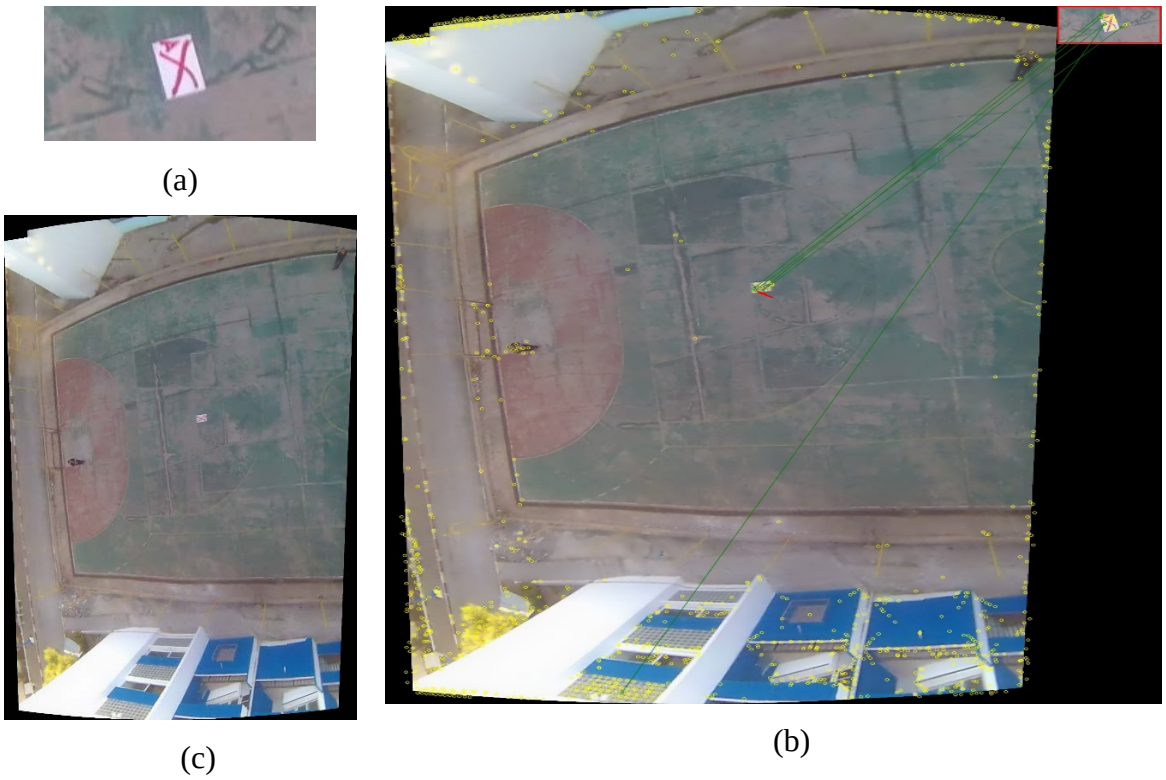


Figure IV.8*: Results of the Object Detection on Stitched Scene*

The results on figure IV.8 are promising as the detection shows positive results over stitching, rotation and scalability.

IV.6 Summary

In this chapter the SURF algorithm is experimented on the images taken by the quadrotor under different conditions, where the SURF have demonstrates its powerful performance as a detector. Under the diverse conditions the algorithm is still getting to the desired object.

Conclusion

In this project we explored a portion for the use of flying robots with mapping and reconnaissance techniques however this needs a lot of fining in order to achieve a fully autonomous system for the applications discussed.

To implement the quadcopter we needed to understand its concept of operation theoretically so we passed by the modeling and controller design methods where we covered the mathematical theory behind the quadrotor kinematics and dynamics, this led to the controller design part where we used the PID techniques to achieve the stability of the 6 DOF of the system.

After discussing the mathematical modelling we interfaced all the sensors and actuators parts with the microcontroller, then we designed the control algorithm that is responsible for stabilizing and maneuvering the quadcopter from hovering to rotations and translations, in addition we developed a ground control station to communicate with the UAV either for manual or automatic piloting or retrieving the pictures captured by the camera onboard.

The images taken by the camera on specific coordinates are processed in the ground station depending on the application selected either image reconstruction using automatic image panorama stitching algorithm which will create a full image from the different captured ones, or the second application which is object detection where it will detect a specific target in a certain area swapped by the quadcopter.

The contribution of this project was to create a full system that consists of a quadcopter with its ground control station that can perform image reconstruction and object detection, this full system can minimize the risks of human being intervening in such missions, it will also minimize the time and cost for these tasks.

A future work to this project can consists of designing a better performance controllers like the Back-stepping, also implementing a reliable communication channels with the ground control station using Xbee modules, and finally adding depth cameras or laser scanners to create a 3D reconstruction environment application.

References

[1] V. Kumar, "Robotics: Aerial Robotics," Coursera, September 2015
Available: https://www.coursera.org/learn/robotics-flight

[2] D. Hartman, K. Landis, M. Mehrer, S. Moreno, J. Kim,
"Quadcopter Dynamic Modeling and Simulation", Master's Report, September 2014.

[3] Jürgen Sturm, Daniel Cremers, "Autonomous Navigation for Flying Robots" Edx, October 2015. Available: https://courses.edx.org/courses/course-v1:TUMx+AUTONAVx+2T2015/info.

[4] T. Jiřinec, "Stabilization and control of unmanned quadcopter", Master's Report,
Czech Technical University in Prague, Mai 2011.

[5] T. Bresciani, "Modeling, Identification and Control of Quadcopter Helicopter", Master's Report, Lund University October 2008.

[6] Araki M, "PID Controller," [Online], Accessed April 2016
Available: http://www.eolss.net/ebooks/Sample%20Chapters/C18/E6-43-03-03.pdf

[7] V.P. Kodgirwar, Vivek Kumar, Manish Shegokar, SushantSawant, "Design of Control System for Quadcopter using Complementary Filter and PID Controller," International Journal of Engineering Research & Technology (IJERT), no. 4, April - 2014.

[8] "Quadcopter DJI-F450 Frame," [Online], Accessed April 2016.
Available: http://www.aliexpress.com/item/HJ450-4-Axis-Multi-Wheel-Frame-Strong-Smooth-KK-MK-MWC-Quadcopter-Kit-For-DJI-F450/32238288681.html.

[9] "How to Make a Drone / UAV – Lesson 3: Propulsion," [Online], Accessed Feb 2016
Available: http://www.robotshop.com/blog/en/make-uav-lesson-3-propulsion-14785.

[10] "Inertial Measurement Unit," [Online], Accessed Feb 2016
Available: https://en.wikipedia.org/wiki/Inertial_measurement_unit.

[11] "GY-88 DataSheet"

[12] "Using the Accelerometer," [Online], accessed Feb 2016
Available: http://husstechlabs.com/projects/atb1/using-the-accelerometer/.

[13] "Gyroscopes - Everything You need to know," [Online], accessed Feb 2016
Available: http://www.gyroscopes.org/math2.asp.

[14] S. Colton, "The Balance Filter", June 2007, accessed Mars 2016
Available: http://d1.amobbs.com/bbs_upload782111/files_44/ourdev_665531S2JZG6.pdf

[15] Jay Esfandyari, Roberto De Nuccio, Gang Xu,
"Solutions for MEMS sensor fusion,", accessed Mars 2016
Available: http://ca.mouser.com/applications/sensor_solutions_mems/.

[16] "The HC-SR04 Ultrasonic Sensor," [Online], Accessed April 2016
Available: http://www.ezdenki.com/ultrasonic.php.

[17] "Global Positioning System," Wikipedia, [Online], Accessed April 2016
Available: https://en.wikipedia.org/wiki/Global_Positioning_System.

[18] Ublox, "Ublox Neo-7N Datasheet".

[19] FlySky, "FlySky FS-T6 Datasheet".

[20] TechTarget, "What is Telemetry," [Online], Accessed April 2016
Available: http://whatis.techtarget.com/definition/telemetry

[21] G. Covey', "Brushless Basics," [Online], Accessed Feb 2016
Available:http://www.rcuniverse.com/magazine/article_display.cfm?article_id=1344.

[22] Arduino, "Arduino Mega," [Online], Accessed Feb 2016
Available: https://www.arduino.cc/en/Main/arduinoBoardMega.

[23] Wikipedia, "Multithreading," [Online], Accessed April 2016
Available: https://en.wikipedia.org/wiki/Multithreading_(computer_architecture).

[24] Wikipedia, "Ground control station," [Online], Accessed April 2016
Available: https://en.wikipedia.org/wiki/Ground_control_station.

[25] Wikipedia, "Real Time Streaming Protocol," [Online], Accessed Mars 2016
Available: https://en.wikipedia.org/wiki/Real_Time_Streaming_Protocol.

[26] Arjan Durressi, Raj Jain, 2005
"RTP, RTCP and RTSP -Internet Protocols for Real Time Multimedia Communication".

[27] Arduino, "Wire Library," Arduino, [Online], Accessed Feb 2016
Available: https://www.arduino.cc/en/Reference/Wire.

[28] "I2C Tutorial," [Online], Accessed Feb 2016
Available: http://www.robot-electronics.co.uk/i2c-tutorial.

[29] Sparkfun, "Serial Communication," [Online], Accessed Feb 2016
Available: https://learn.sparkfun.com/tutorials/serial-communication.

[30] Arduino, "Serial," [Online], Accessed Feb 2016
Available: https://www.arduino.cc/en/Reference/Serial.

[31] Wikipedia, "Gaussian Frequency-Shift Keying," [Online], Accessed April 2016

Available: https://en.wikipedia.org/wiki/Frequency-shift_keying#Gaussian_frequency-shift_keying.

[32] D. Engineering, "BEC," [Online], Accessed Mars 2016
Available: https://www.dimensionengineering.com/info/bec.

[33] U. Holmberg, "PID control - Simple Tuning Methods," 2010.

[34] M. B. a. D. G. Lowe, "Automatic Panoramic Image Stitching using Invariant Features," Department of Computer Science, University of British Columbia, Vancouver, Canada, International Jornal of computer vision , Aug 2007.

[35] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints".International Jornal of computer vision , Nov 2004.

[36] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," Proc. of the International Conference on Computer Vision, Sept 1999.

[37] K. G. Derpanis, "Overview of the RANSAC Algorithm," kosta@cs.yorku.ca, Version 1.2, May 13, 2010..

[38] D. Kriegman, "HomographyEstimation," CSE252A, Winter2007.

[39] H. P.Gavin, "The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems," Department of Civil and Environmental Engineering Duke University, May 4, 2016.

[40] D. H. T. T. William T.C. Neale, "Photogrammetric Measurement Error Associated with Lens Distortion," Kineticorp, LLC, 2011.

[41] A. E. T. T. a. L. V. G. Herbert Bay, "Speeded-UpRobustFeatures(SURF)," Elsevier , 10 September 2008.

[42] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," Proc. of the International Conference on Computer Vision, Sept, 1999.

[43] R. Wang, "Difference of Gaussian (DoG)," Oct 2014. [Online].
Available: http://fourier.eng.hmc.edu/e161/lectures/gradient/node9.html.

Appendix A: Propeller Balancing

Propellers are the ones responsible for the propulsion system due to their ability to convert a rotational movement to a lifting power, but they are very sensitive and must be balanced, because if they are not, a series of vibrations will be generated when rotating which will transformed all over the body frame, thus the accelerometer will have inaccurate reading in addition, we will notice that the quadcopter is drifting to the side of the unbalanced propeller.

Even though a new pack of propellers come unbalanced, thus we need to check one by one for the defective ones. The easy way to balance propellers is to add a small piece of tape into the unbalanced side and check again.

Figure A.1 shows the unbalanced propeller using simple mechanism.

Figure A.1: *Unbalanced Propeller*

After adding the necessary amount of tape into the bottom side of the propeller, we got it balanced (Figure A.2).

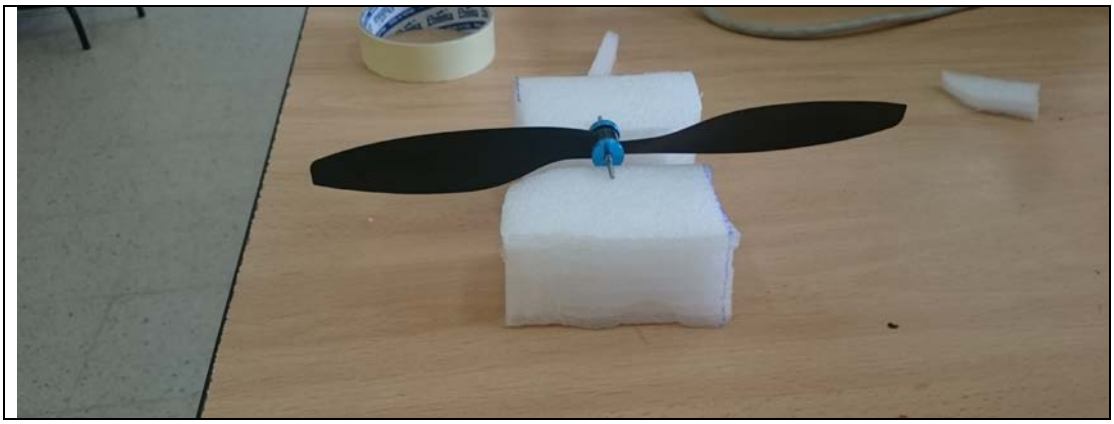


Figure A.2*: Balanced Propeller*

Appendix B: BLDC Motor Specifications

The motors used in this project are 980KV, a full specification with the appropriate propellers and the thrust generated by the propulsion system (motor and propeller) is given in figure B.1.

Motor: A2212 KV:980

Technical Datas		Recommended Prop(inch)			
KV	980	Standard	2s-1147/1155	Max thrust	2s-9050
Configu-ration	12N14P		3s-7035/8040		3s-8043
Stator Diameter	22mm				
STator Length	12mm				
Shaft Diameter	3mm				
Motor Dimension(Dia.*Len)	Φ27×30mm				
Weight(g)	50				
Idle current(10)@10v(A)	0.5				
No.of Cells(Lipo)	2-3S				
Max Continuous current(A)180S	14.5				
Max Continuous Power(W)180S	160				
Max. efficiency current	(4-9A)>80%				
internal resistance	120mΩ				

Tested with Angel 20A ESC

Prop	Volts (V)	Amps (A)	Watts (W)	Thrust (g)	Efficiency (g/W)			
9x4.7	7	4.4	30.8	360	11.69			
	8.5	5.7	48.45	510	10.53			
	10	7.3	73	610	8.36			
	11	8.3	91.3	720	7.89			
10x4.7	7	7.5	52.5	540	10.29			
	8.5	9.6	81.6	660	8.09			
	10	11.2	112	760	6.79			
	11	12.4	136.4	820	6.01			

Figure B.1: *A2212 Motor Specifications*

Appendix C: Electronic Speed Controller (ESC)

The ESC as mentioned in chapter two is the electronic module behind driving the motors with the appropriate signals, a full specification of the one used in this project is stated in table C.1 and a full schematic describing the internal circuitry is in figure C.1.

Table C.1: *ESC Mystery 30A Specifications*

Brand	Mystery
Output	Continuous 30A, burst 40A up to 10 Secs.
Input voltage	2-4 cells lithium battery 5-12 cells NiCd/NIMh battery
BEC	2A / 5V (Linear mode).
Max speed	210,000rpm for 2 poles BLM, 70,000rpm for 6 poles BLM, 35,000rpm for 12 poles BLM
Size	45 * 24 * 11mm
Weight	25g

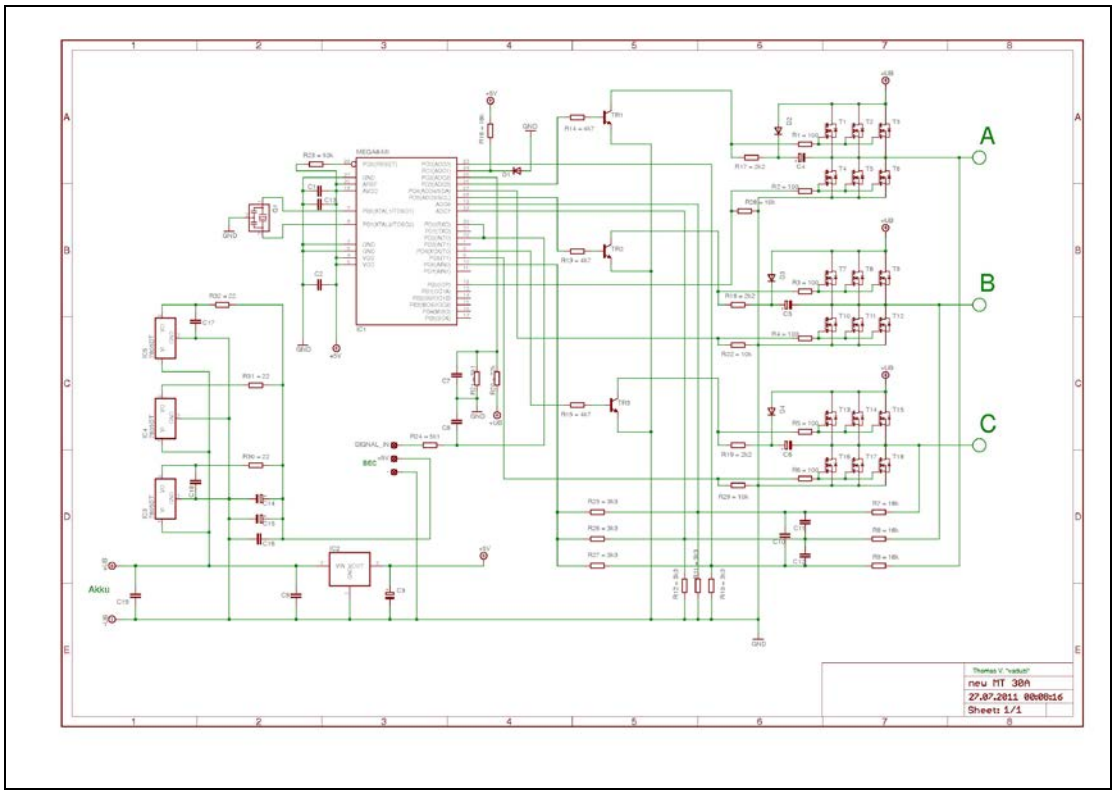


Figure C.1: *ESC Mystery 30A Full Schematic*

Appendix D: Arduino Mega Board and ATmega2560 Microcontroller

The Arduino Mega board is the one used in order to implement the flight controller of the quadcopter, it is based on the ATmega2560 microcontroller which has enough interrupts and serial communication ports, this fits with the requirements of our project, a full specifications of the board are presented in table D.1 and a schematic of the microcontroller is in figure D.1.

Table D.1: *Arduino Mega Board 2560 Technical Specifications*

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	37 g

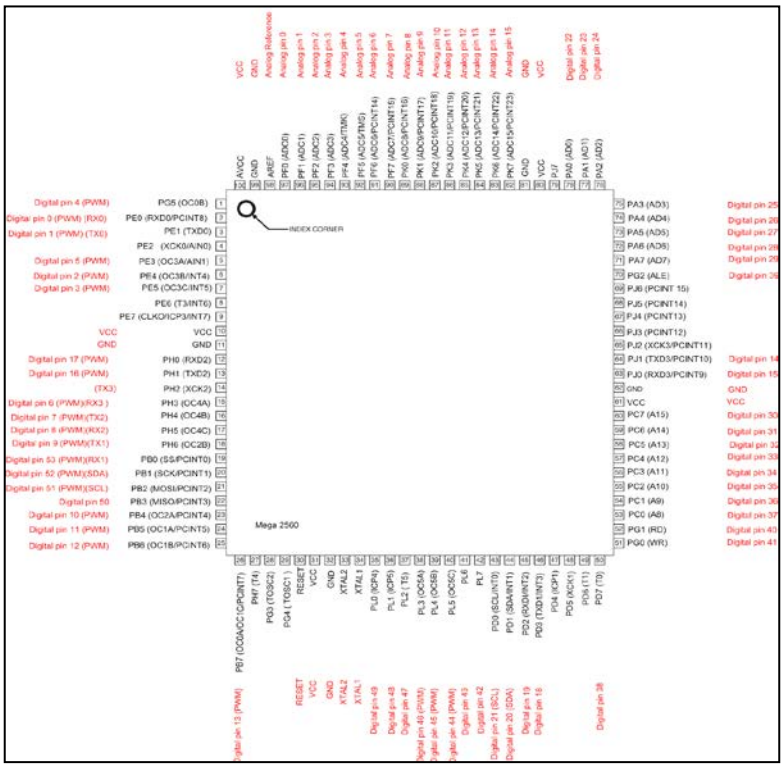


Figure D.1: *ATmega 2560 Pinmap to Arduino Mega*