PROJECT REPORT

on

UAV BASED ATMOSPHERIC PROFILER

Submitted by

TIMOTHY SIMON THOMAS

AARON RENNY

JOSEPH JOJO A

MUHAMMED HAFIS NK

K NAVANEETH

in partial fulfillment of requirement for the award of the degree

of

BACHELOR OF TECHNOLOGY
in
ELECTRONICS AND COMMUNICATION

DIVISION OF ELECTRONICS ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
KOCHI-682022

MAY 2021

DIVISION OF ELECTRONICS ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
KOCHI-682022

CERTIFICATE

Certified that the report entitled ***"UAV BASED ATMOSPHERIC PROFILER"*** *is a bonafide work of* ***TIMOTHY SIMON THOMAS, AARON RENNY, JOSEPH JOJO A, MUHAMMED HAFIS NK, K NAVANEETH*** *towards the partial fulfillment for the award of the degree of B.Tech in Electronics and Communication of Cochin University of Science and Technology, Kochi-682022.*

Project Guide	**Project Coordinator**	**Head of the Division**
Dr. Rekha James	Dr. Binu Paul	Dr. Babita Roslind Jose

ACKNOWLEDGEMENT

We extend our sincere gratitude to our respected Head of Division Dr. Babitha Roslind Jose. We are highly indebted to our guide Dr. Rekha.K.James for her guidance and constant supervision, as well as for providing necessary information regarding the various topics related to the project and guiding us through it . We would like to express our gratitude towards various faculties at the Division of Electronics and Communication, School of Engineering for the knowledge and encouragement they have imparted which made the successful completion of this project possible. We also express our sincere thanks to all the faculty members of the Division of Electronics and Communication Engineering for their kind cooperation and valuable suggestions during the period of this work. We take this opportunity to thank all our friends who helped us throughout this work and for their patient discussion and suggestion and for their timely aid. Finally, we take the opportunity to thank our parents for their blessings and suitable help. Above all, we thank God Almighty for his abundant blessings, without his blessings we would not have been able to complete this venture.

Thank you.

ABSTRACT

In this modern era of technology Unmanned Aerial Vehicles (UAVs) have found their use in a variety of applications ranging from agriculture to military surveillance. Due to the ease of control of these drones and the ability to manufacture them to fit various applications, they have emerged to become a dynamically expanding field of technology. The flight controllers are the brains of multirotor aircraft. They consist of circuits which are used to control the aircraft. This flight controller can be designed in various ways in order to execute the desired operations. This gives robustness to UAVs. This report discusses how to make a flight controller for weather data collection , its applications and major challenges.

CONTENTS

1. **Introduction** 1
   1.1 Multirotor aircraft in data collection 1

2. Principle of operation **3**
   2.1 Input capture mode 3
   2.2 Receiver - MCU interconnection 4
   2.3 ESC pulse generation from STM-32 5
   2.4 Complementary filter 5
   2.5 PID controller 8
   2.6 PID tuning 10
   2.7 Propeller balancing 11
   2.8 IMU calibration 11

3. Implementation **13**
   3.1 Prototyping 13
   3.2 Software development 14
   3.3 PCB design 14
   3.4 Telemetry module 17

4. Components **19**
   4.1 Microcontroller 19
   4.2 MPU-6050 20
   4.3 MS5611 21
   4.4 LoRa 21
   4.5 ATMEGA328-PU 22
   4.6 Sensor - DHT11 22
   4.7 Sensor - UBLOX NEO 6M GPS module 23

4.8 Power distribution 24
4.9 Drone frame 25
4.10 Transmitter 25
4.11 BLDC motor 26
4.12 ESC 26
5. Flight testing 28
6.Challenges 30
6.1 Mechanical noise 30
6.2 Temperature 30
6.3 EMI 30
6.4 Electrical noise 30
6.5 Oscillations during flight 30
7.Conclusion 31
8.References 32
8.List of figures 33

CHAPTER 1

INTRODUCTION

A Flight controller is the brain of a drone. It's a circuit board with a range of sensors that detect movement of the drone. Its main function is to direct the RPM of each motor in response to input. Simply applying a similar input to the 4-motors of a quadcopter will not cause the aircraft to fly, it will simply topple and crash. Commands from the pilot for the multi-rotor to move are fed into the flight controller, which determines how to manipulate the motors. All flight controllers consist of basic sensors such as gyroscope and accelerometer, barometric sensor and compass.

There are three key function tha a flight controller perform in a drone

1. Perception

   The inertial measurement unit and some sensors in the flight controller are used to detect the height, orientation and obstacle distances. Flight controller filters some of these informations and operates on others for a more precise operation
2. Controlling besides sensing the flight controller uses the data from the sensors to control the drone. The drone can rotate, accelerate by creating a speed difference between each motor.
3. Communication

   Communication is a key feature of the flight controller. The information given out by the flight controller must be understood by the human pilot. Another important thing to be communicated is the battery level. With the autonomous features, nowadays flight computers need to communicate with other systems to know its flight path and destination.

1.1 MULTIROTOR AIRCRAFT IN DATA COLLECTION

As a basic definition, weather is the state of the atmosphere. Most weather occurs in the troposphere, or the lowest layer of the atmosphere. Weather is made up of multiple parameters, including air temperature, atmospheric pressure, humidity, and wind. Wind speed describes how fast the air is moving past a certain point. This may be averaged over a given unit of time, such as miles per hour, or an instantaneous speed, which is reported as a peak wind speed, wind gust

or squall. Wind speed and direction are important for monitoring and predicting weather patterns and global climate.

Small drones (multi-copters) have the potential to deliver valuable data for atmospheric research. They are especially useful for collecting wind speed , temperature and humidity. What makes them particularly useful is their vertical take-off and landing, they are also easy to operate, as they are equipped with autopilot systems. Moreover, drones do not produce pollution, typical for combustion engines.

Due to these desired properties, multicopters have been used to monitor and collect weather data.

CHAPTER 2

PRINCIPLE OF OPERATION

2.1 INPUT CAPTURE MODE

Multicopters are very unstable by nature. So an enormous number of corrections must be done for a stable flight. This quadcopter is designed for a refresh rate of 250 hz(i.e., ESCs are corrected for every 4ms). ESCs are controlled using pulses that vary between 1000us to 2000us. In order to detect this pulse width from the transmitter input capture mode is performed by the stm32 microcontroller. Pulse width is obtained by subtracting the begin time (tbegin)and end time(tend) of the pulse.

$t_{pulse} = t_{begin} - t_{end}$.

This is the timer section of the stm32 microcontroller.

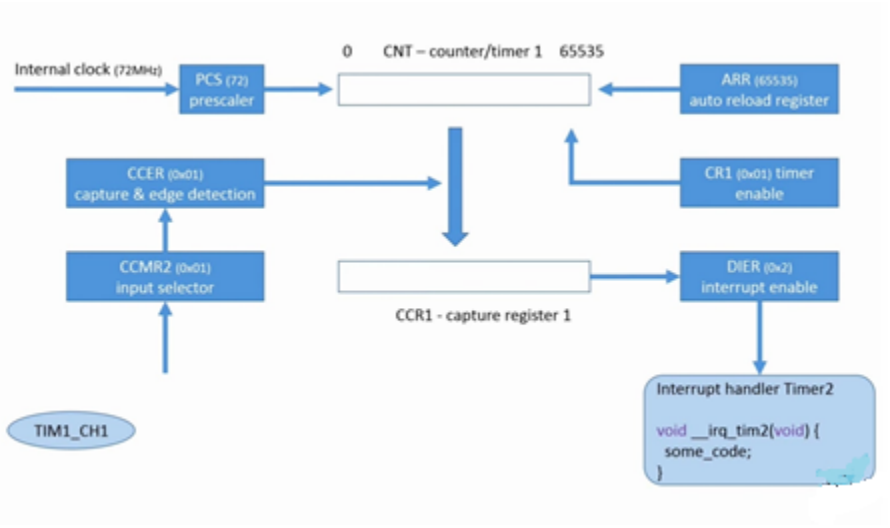


Fig 2.1 Timer section of STM32f103

There is a 16 bit counter which is connected to an Auto Reload Register. This counter counts from 0 to FFFF(or the count loaded) and when the count is over it gets auto reloaded. This counter is not affected by any interrupts. The clock pulse for the counter is generated by the main system clock which runs at 72Mhz frequency. Before giving the clock the system clock is pre scaled to 1 Mhz by the PSC block as a result the clock signal increments every 1us.

The receiver is connected to the TIM1_CH1 , the rising edge of the pulse is detected by the Capture Compare Enable Register (CCER)block and the timer value is stored at the capture register. After capturing, an interrupt is generated and the handler subroutine is executed. In the

handler subroutine the pulse width is calculated. On detecting a rising edge the capture time is stored in a variable and CCER is set to detect the falling edge. The CCER is set here because it can detect only one type of edge at a time. After detecting the falling edge the current capture time is subtracted from the variable where the capture time of rising edge is stored and this will give the pulse width and the CCER set back again to detect the rising edge.

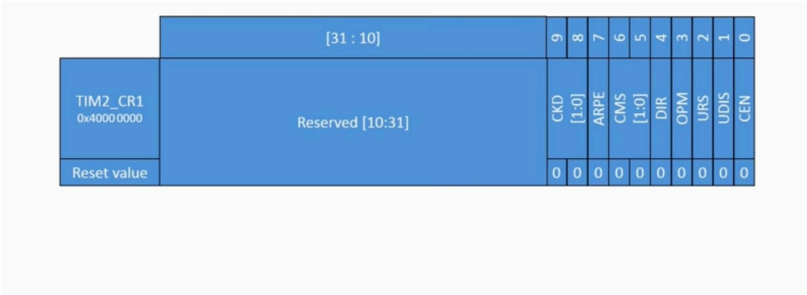


Fig 2.2 32 bit Timer2 register

In order to enable this counter we need to set the timer register. Here we use timer2 register which is a 32 bit log register This is how the input capture mode works

2.2 RECEIVER - MCU INTERCONNECTION

The FS-i6 receiver uses two possible encoding techniques to communicate data from the transmitter, namely PWM and PPM. Flight controllers use either one or both as inputs, conventional circuits tend to utilize PWM owing to its simplicity. The PWM route however comes with an overhead in the number of PCB traces required. A six channel would require a minimum of 8 traces to communicate via PWM, while also utilizing a larger number of GPIO pins of the MCU. The PPM route was thus opted for as only a total of three traces and a single MCU pin are utilized for the implementation.

An appropriate program to decode the encoded signal is required to interpret the receiver inputs. The interrupt handler linked with the timer input pin executes the decoding into six separate channels. Below given is the figure detailing the encoding of the 6 receiver channels(servo channels).

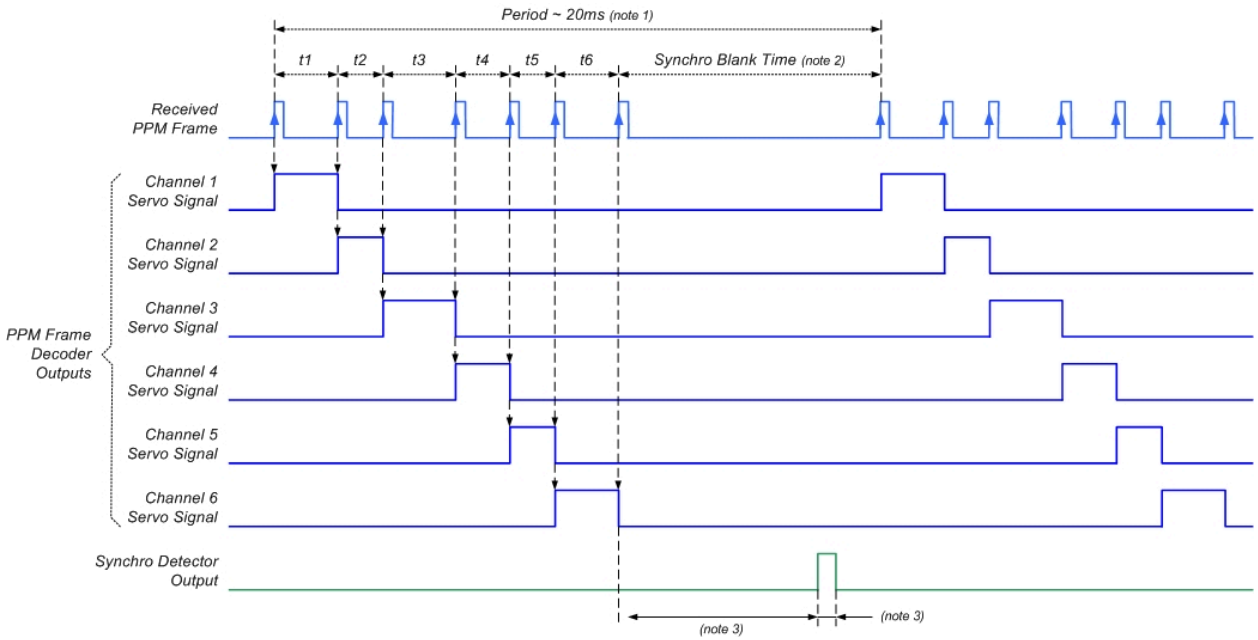


Fig 2.3 PPM Frame for 6 channel receiver

2.3 ESC PULSE GENERATION FROM STM-32

Input capture mode stores the pulse width of the signal in the CCR1 register. The output to be fed to the ESC is generated with the help of this stored value. The STM-32 has a 0-5000 auto reload counter CNT. Since the working frequency is 72 Mhz, each increment takes 1microseconds. Hence, for the counter to be reloaded, it takes 5000 microseconds. The pulse width varies from 1000 to 2000 microseconds and the refresh rate is 250 Mhz. These values come well within the range of the timer.

In order to generate the required output, the CCR1 register is compared to the CNT timer. If CCR1 is less than the timer value, the output remains high. When the timer value surpasses the register value, the output is switched to low. In this way, the pulse of required width can be supplied to the ESC. PB6 is configured as the PPM pin to act as the output.

2.4 COMPLEMENTARY FILTER

Complementary filter is a stabilisation technique used to ensure smooth working of PID controllers. The main function of the filter is to take the angular displacement values from the gyroscope and accelerometer and determine the stable position of the quadcopter.

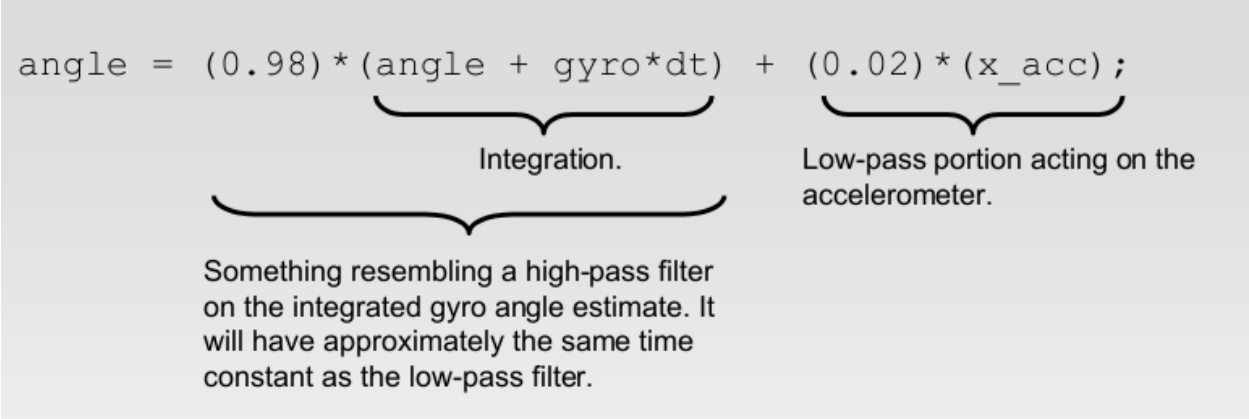


Fig 2.4 Implementation of complementary filter in code

Angular displacement can be obtained in two ways. One is from the accelerometer which gives the value of angular displacement directly. Another is from the gyroscope. The gyro gives the value of the angular velocity, which when integrated, gives the value of the angular displacement. The time period over which this integration will take place is given by the time taken for the loop which does the calculation to run.

There are disadvantages for both of these methods. The angular displacement from the gyro is obtained through integration. Hence, there is always an integral constant getting added to the final value. This leads to an error which is cumulative in nature and for small values of angular displacement, the error can increase drastically with increase in time period. Hence, this method is faulty in long intervals of time.

The accelerometer is a highly sensitive device. Hence, even a small jerk from the propellers of the quadcopter can register as valid readings in an accelerometer angular displacement value. Hence, it is faulty in short periods of time. But on taking the average value over a long period of time, it is possible to get an accurate reading.

From the above, it can be inferred that the high frequency components of the accelerometer are faulty and the low frequency components of the gyroscope also represent errors. In order to obtain an accurate value over all periods of time, we set up a complementary filter. The low frequency components of the gyroscope and high frequency components of the accelerometer are discarded. This is done by using the weighted mean equation.

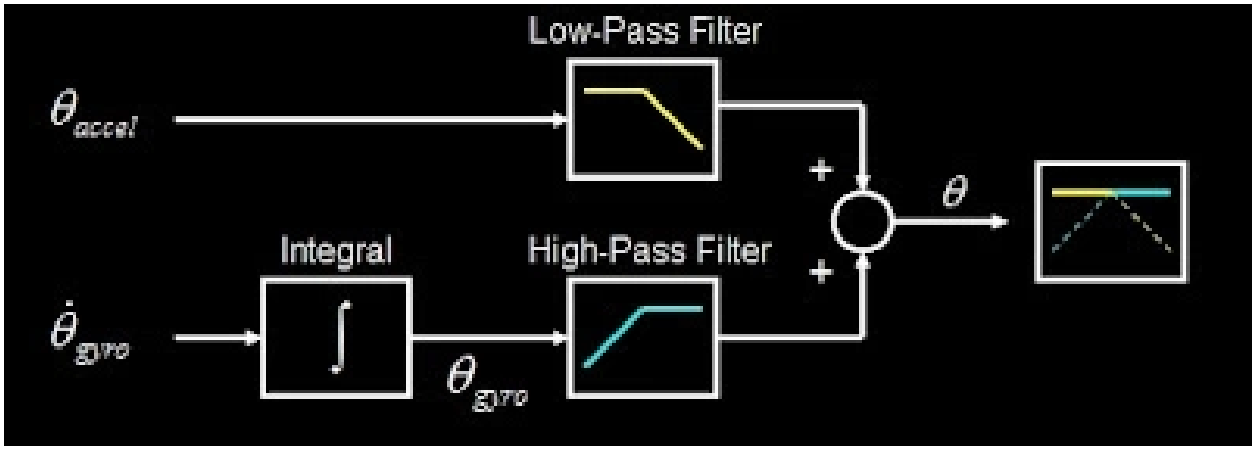


Fig 2.5 Block Diagram of Complementary Filter

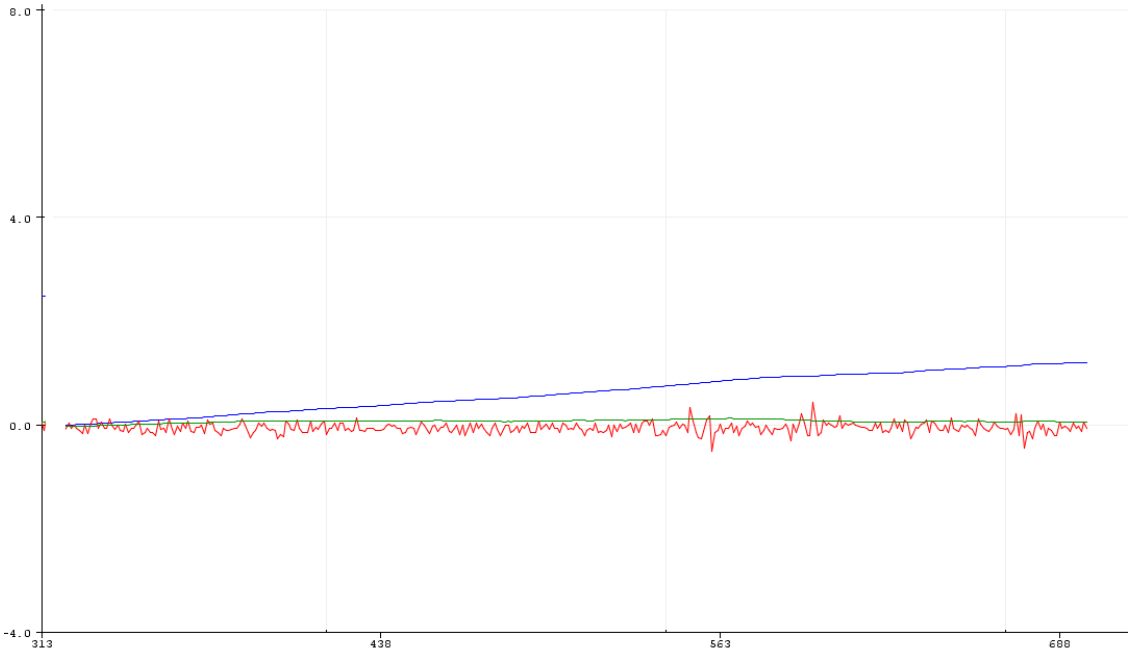


Fig 2.6 Complementary filter data obtained from the IMU

The above graph is obtained from the IMU of the implemented flight controller, the blue line represents the angle as estimated by the gyroscope. The red plot is the angle as indicated by the accelerometer. The angle as estimated by the complementary filter is given in green. It can be observed that the gyroscope angle tends to deviate with time due to the error in summation. The accelerometer data is extremely noisy, the output of the complementary filter is as expected, and is the closest to the true value 0. The above experiment is conducted with the quadcopter kept level with no induced disturbances.

Hence, both the values are combined optimally to get an accurate reading of angular displacement. This helps the quadcopter to maintain balance and stabilise itself.

2.5 PID CONTROLLER

The response of the UAV to the inputs from the IMU are primarily governed by the PID controller. The PID algorithm forms the heart of the flight controller, it enables the aircraft to maintain level flight. In the absence of a controller the aircraft would topple due to uneven thrust from the motors. The controller compensates for variations in CG position, environmental disturbances and user inputs. PID control uses a closed-loop control feedback to keep the actual output from a process as close to the target or setpoint output as possible. In this application the setpoint is the desired angle of the multirotor, while the feedback is obtained from the IMU measurements. The controller provides a control signal that governs the individual RPMs of the motors. This ensures that the quadcopter can achieve level flight.

PID controller consists of three terms, namely proportional, integral, and derivative control. The combined operation of these three controllers gives a control strategy for process control

A feedback signal from the process plant is compared with a set point (desired angle) and the corresponding error signal e(t) is fed to the PID algorithm. According to the proportional, integral, and derivative control calculations in the algorithm, the controller produces a combined response or controlled output which is applied to the plant (quadcopter).

The working principle behind a PID controller is that the proportional, integral and derivative terms must be individually adjusted or "tuned." Based on the difference between these values a correction factor is calculated and applied to the output. For example, if the UAV has a bank to the right, the IMU will measure the deviation from the setpoint and an error signal which results in the speeding up of the motors on the right hand side will be sent. Here are the three steps in the PID algorithm:

1. **Proportional tuning** involves correcting a target proportional to the difference. Thus, the target value is never achieved because as the difference approaches zero, so too does the applied correction.
2. **Integral tuning** attempts to remedy this by effectively cumulating the error result from the "P" action to increase the correction factor. For example, if the oven remained below temperature, “I” would act to increase the head delivered. However, rather than stop heating when the target is reached, "I" attempts to drive the cumulative error to zero, resulting in an overshoot.
3. **Derivative tuning** attempts to minimize this overshoot by slowing the correction factor applied as the target is approached.

Mathematically the algorithm can be expressed as below. The below block diagram describes the entire control system.

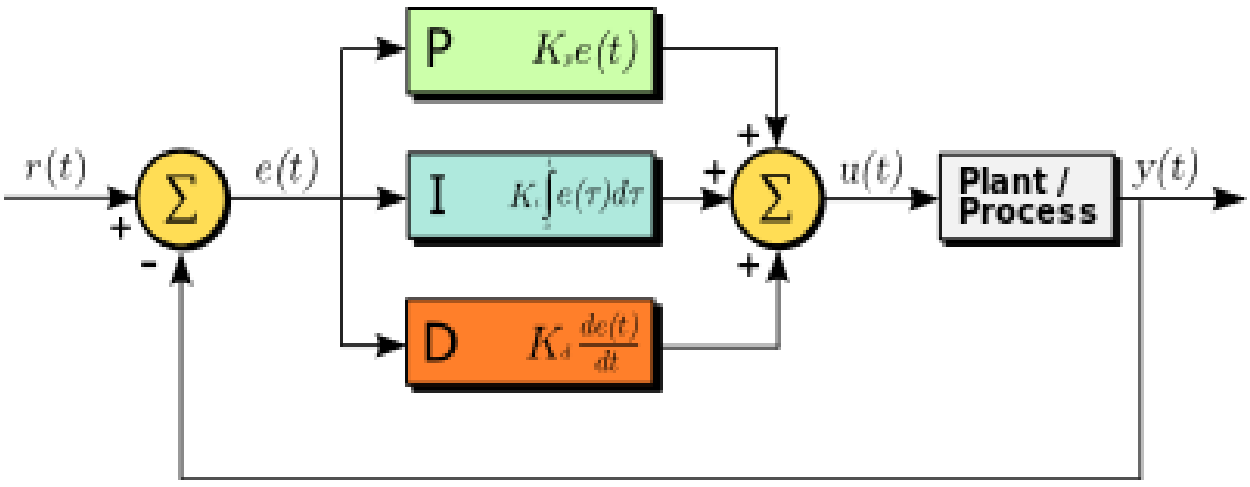


Fig 2.7 PID controller block diagram

The effect of the various gains are visible in the step response of an arbitrary system given below.

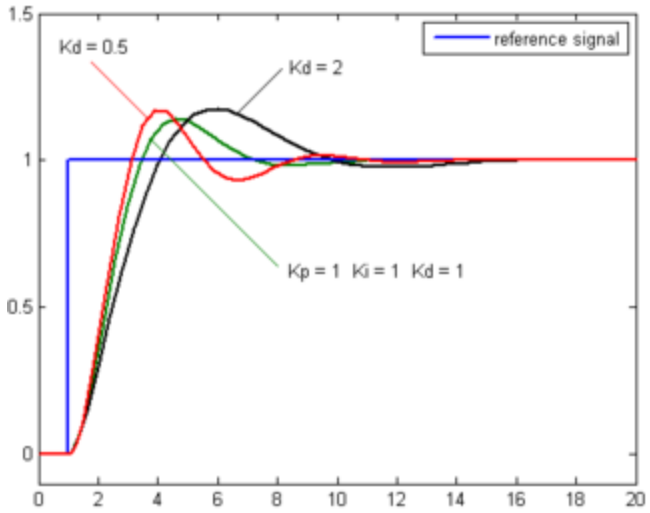


Fig 2.8 Step response of a generic PID controller

As evident the individual gains vary from system to system, for the UAV the gains have to be tuned manually as most simulations yield highly un-realistic gain estimates. The individual gains are highly specific to the system under consideration.

2.6 PID TUNING

Tuning of the proportional, differential, integral gains is done by observing the response of the UAV to changes in the gains. The UAV is first held firmy, with the motors armed. Practically it was found that the below method proved most accurate.

1) All gains are set to zero.
2) The proportional gain and differential gains on the yaw axis are increased, but kept at a minimum to counteract the propellor induced Yaw.
3) The proportional gain(Kp) of the controller for both the roll and pitch axes are increased. After each increment the response of the UAV at approximately 25% of max throttle is observed.
4) The proportional gain is increased until the UAV begins stable oscillations about the mean position, in increments of 0.5. Kp is then set at 10% below the gain at which the oscillations first occurred.

5) All gains are set to zero.
6) The differential gain(Kd) of the controller for both the roll and pitch axes are increased in increments of 0.2. After each increment the response of the UAV at approximately 25% of max throttle is observed.
7) The differential gain is increased until the UAV begins to violently vibrate. Kd is then set at 10% below the gain at which the vibrations occurred.(It should be noted that vibrations induced by the motor can feedforward and lead to uncontrolled vibrations if the gains are set too high.
8) The proportional and differential gains are set to the obtained values and the integral gain is increased in increments if 0.0005, until the multirotor becomes unstable. The obtained value of Ki is the final integral gain.

The aircraft can now be flightested with caution. Slight modifications to the obtained gains may be made to improve flight characteristics.

2.7 PROPELLER BALANCING

Propeller balancing is important because unbalanced propellers cause vibrations. This will make the gyroscope and accelerometer to produce noise which will make the motors to jolt. Each propeller needs to be balanced separately. For balancing, hold the end of the motor that is needed to be calibrated. Disconnect the other motors and give the throttle. Some values will be displayed on the serial monitor. These are the vibrations detected by the accelerometer. Now to reduce these vibrations stop the motors and stick a small piece of tape on the propeller. And again give the throttle. Now check whether the values are reduced and the serial monitor. Keep doing this until the motor and propeller runs smooth. Repeat these steps for other propellers.

2.8 IMU CALIBRATION

The gyro and accelerometer is needed to be calibrated for perfect angle measurements. Inorder to calibrate,the quadcopter should be in still position. Place the quadcopter on a leveled surface and upload the gyro and accelerometer codes. A set of values will be printed on the screen; these are the offset values for the gyro and accelerometer. These values are used to calculate the yaw, pitch and roll angles of the drone. The gyro and accelerometer must be calibrated after each crash or if

the temperature varies +/-5. The algorithm measures and computes the average of over 2000 samples of data from the accelerometer and gyroscope and uses this obtained value as the reference for the setpoint.

The gyroscopes calibration is dependent on the ambient temperature,These changes can be significant enough to cause unnecessarily drifting of the quadcopter.The flight controller thus calibrates the gyroscope before each new session. A manual override can be used to save time, however this is not recommended.

CHAPTER 3

IMPLEMENTATION

3.1 PROTOTYPING

The first step in the design process was to develop a schematic for prototyping. A subset of the final schematic with the bare essentials was initially chosen. The design was then developed in segments to develop software for each region. Such a method also paved the way for easier debugging and fault detection. The individual units were first tested on a breadboard to ensure that the individual units performed as expected.

Once all components were satisfactorily tested on a breadboard with the MCU the next phase of development began. The individual components were then integrated into their final form on a dot PCB (prototyping board). A Dot PCB allowed for firm connections that were potentially resistant to the vibrations of flight testing. A breadboard would be highly unsuitable for such testing.

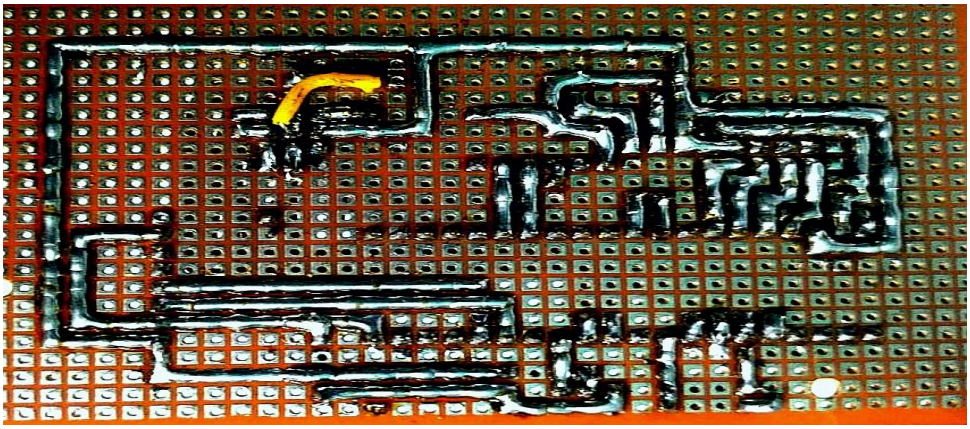


Fig 3.1 Bottom Layer of DOT PCB

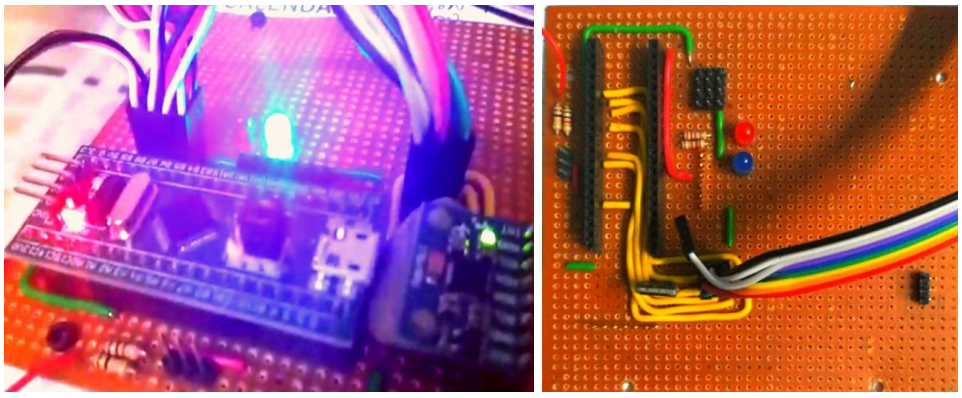


Fig 3.2 Populated Dot PCB

Once our primary hardware was validated the next step was to develop the arduino software.

3.2 SOFTWARE DEVELOPMENT

The software development was to be done on either of two contesting platforms namely STM32Cube IDE or the Arduino IDE. The STM32Cube IDE provided support for Embedded C and C++, providing a very low level programming approach. The project's unique speed requirement called for low level programming. However working on embedded C can be a very challenging task as most drivers and handlers need to be developed. This can prove to be a mammoth task for a flight control program. The Arduino IDE however comes with a lot of baggage as most of the inbuilt functions come with many lines of code that needlessly increase latency. The Arduino C/C++ language has much better abstraction and readability.

A tradeoff was thus made and the software was developed on the Arduino IDE by using port manipulation techniques to reduce the latency caused by inbuilt function. The use of port manipulation and register manipulation significantly reduces latency. The STM32 add-on for Arduino developed by RogerClark was used and the program was developed in Arduino. The low level approach undertaken required fundamental understanding of the STM32F103 MCU architecture.

3.3 PCB DESIGN

The first step in the design process if to design the schematic taking into account all the design considerations and requirements. Below given is the schematic used for the design of the primary flight control unit, The distribution of power, reduction of EMI etc are also taken into consideration at this stage. The PCB is designed such that the flight controller can be upgraded without changes in hardware. Additional connectors for GPS, Compass, barometer, SDcard etc are added to pave way for future development on the flight controller without changes in hardware. A software patch would thus be sufficient to ensure a fully autonomous multirotor.

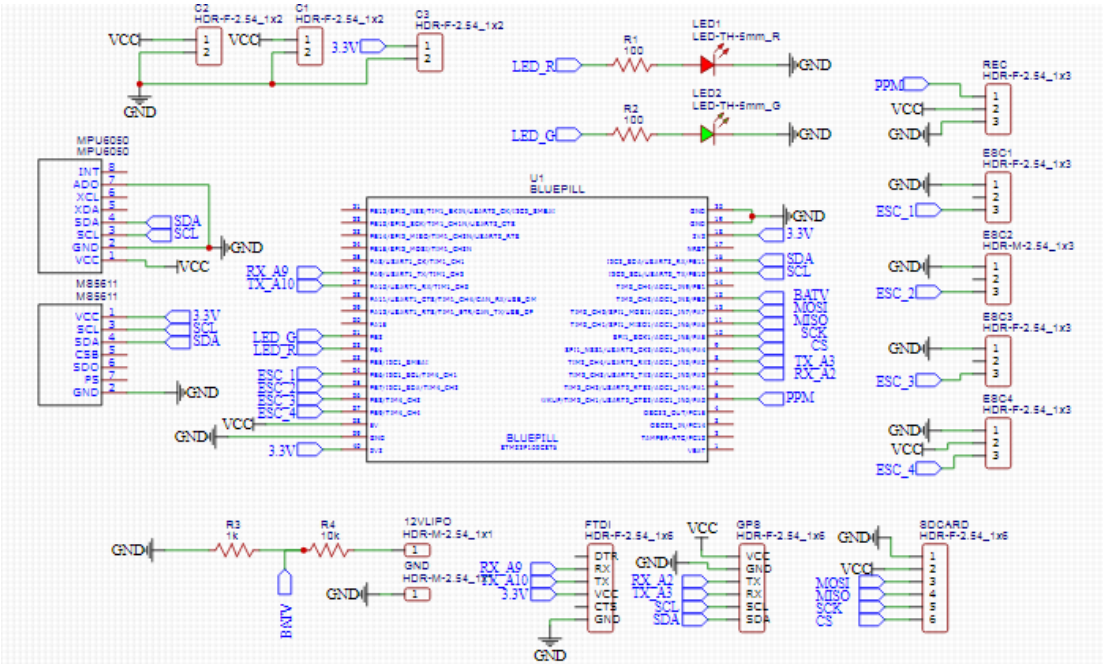
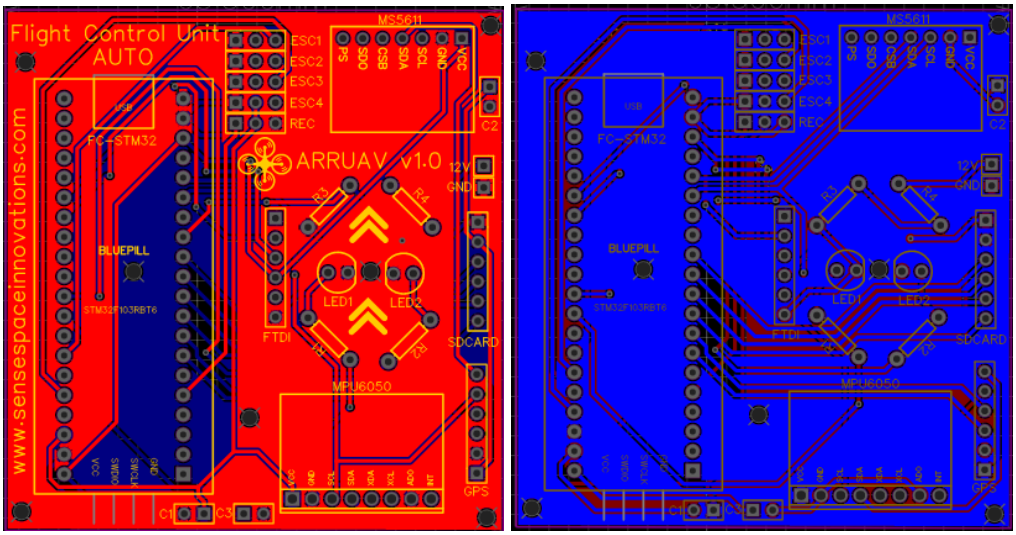


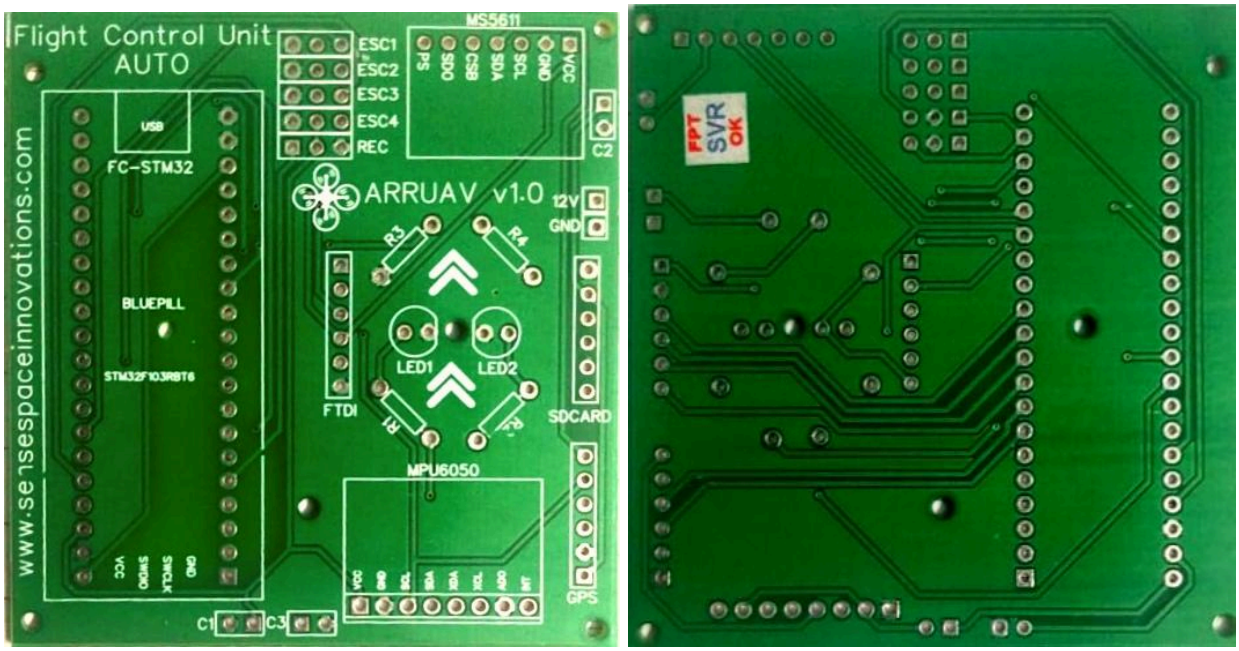
Fig 3.3 Schematic of the flight control unit

The PCB Layout was then designed in EasyEDA with dimensions 63.724mm x 66.989 mm. The PCB traces were designed to carry a maximum of 5V 1.3A under standard conditions. This yielded a trace width of 0.6mm. The place and routing was then done and required two layers to converge. 2mm holes were placed to mount the flight controller to the main drone frame. Ports of LOW ESR capacitances are provided to facilitate mitigation to electronic noise generated by the high frequency switching of the ESC. Two copper areas connecting the GND and VCC nets are placed on the Top and bottom layers respectively.



a) Top Layer **b) Bottom Layer**

Fig 3.4 PCB Layout



a)**Top Layer** **b) Bottom Layer**

Fig 3.5 Printed PCB

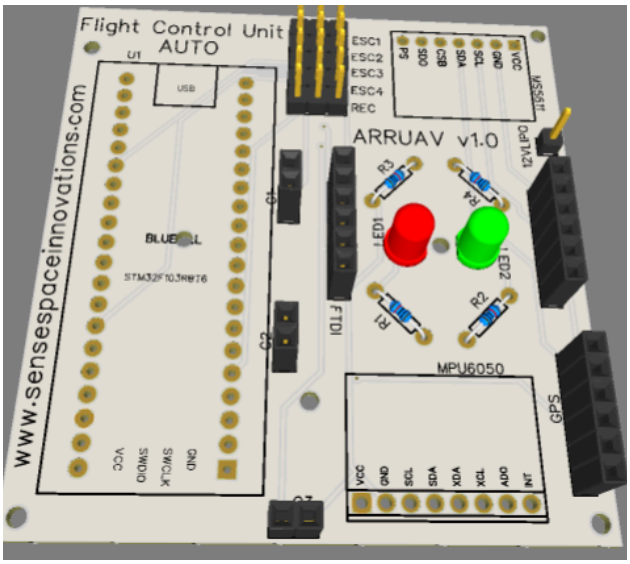


Fig 3.6 3D Software render of PCB

Below shown is the entirely populated board. The entire PCB was mounted on a vibration damper to reduce the mechanical vibrations incident on the IMU.

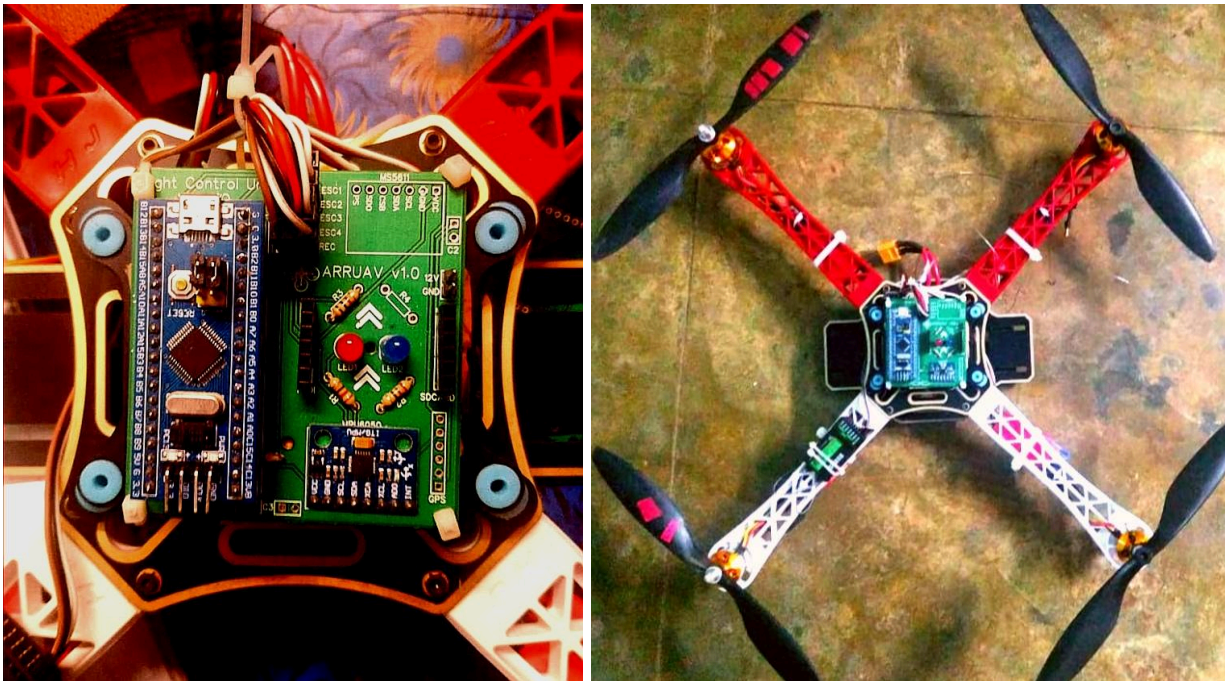


Fig 3.7 Avionics bay of the UAV

3.4 TELEMETRY MODULE

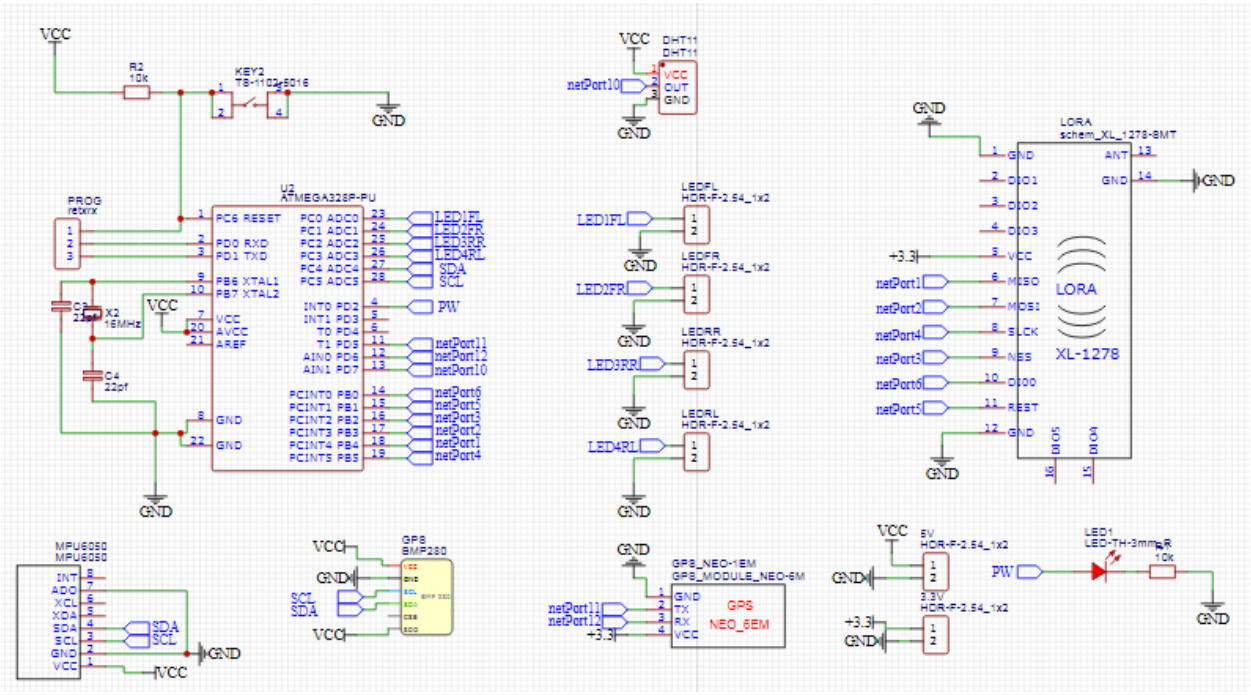
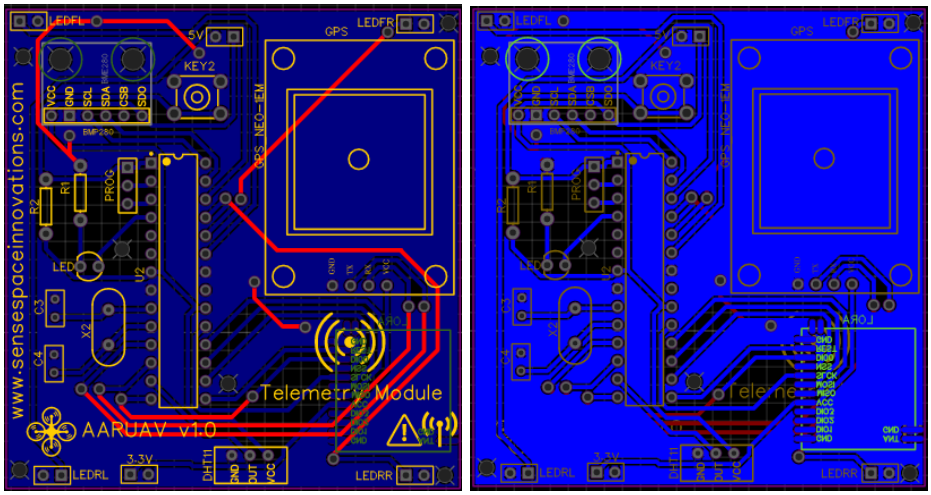


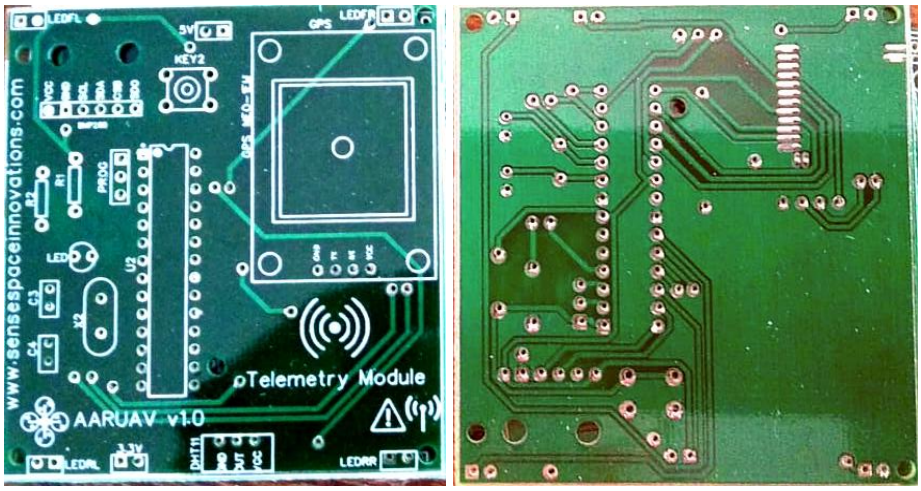
Fig 3.8 Schematic of the telemetry module

The PCB Layout was then designed in EasyEDA with dimensions 63.724mm x 66.989 mm exactly the same as the Primary Flight Controller to facilitate stacking. The PCB traces were designed to carry a maximum of 5V 1.3A under standard conditions. This yielded a trace width of 0.6mm. The place and routing was then done and required two layers to converge. 2mm holes were placed to mount the flight controller to the main drone frame. The traces are routed with 45° bends to reduce EM radiation from sharp bends, Length of HF signals have been optimised to minimise attenuation. A copper area connecting the GND pads is placed on the bottom layer. Copper on the top layer was avoided due to fabrication constraints.



a) Top Layer **b) Bottom Layer**

Fig 3.9



c) Printed PCB Top Layer **d) Printed PCB Bottom Layer**

CHAPTER 4

COMPONENTS

4.1 MICROCONTROLLER

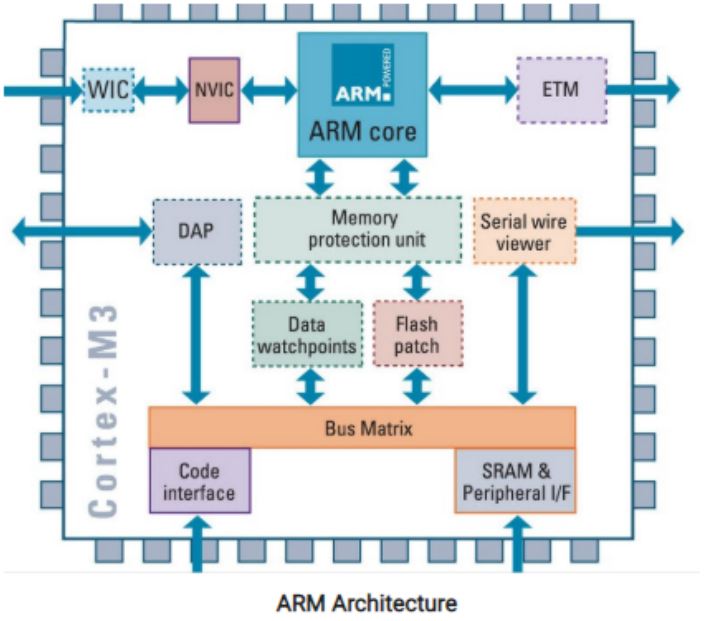


Fig 4.1 Architecture of STM32f103

The above shown is the basic block diagram of the stm32 m3 cortex. **STM32** is a family of 32-bit microcontroller integrated circuits by STMicroelectronics. Internally, each microcontroller consists of the processor core, static RAM, flash memory, debugging interface, and various peripherals.

The STM32F307xx family incorporates high-speed embedded memories (Flash memory up to 1 Mbyte, up to 192 Kbytes of SRAM), up to 4 Kbytes of backup SRAM, and an extensive range of enhanced I/Os and peripherals connected to two APB buses, three AHB buses and a 32-bit multi-AHB bus matrix.

This family of microprocessors include three 12-bit ADCs, two DACs, a low-power RTC, twelve general-purpose 16-bit timers including two PWM timers for motor control, two general-purpose

32-bit timers. a true random number generator (RNG). They also feature standard and advanced communication interfaces.

These communication interfaces include

- Up to three I2Cs
- Three SPIs, two I2Ss full duplex..
- Four USARTs plus two UARTs •
- An USB OTG full-speed and a USB OTG high-speed with full-speed capability (with the ULPI),
- Two CANs protocol.
- Ethernet and the camera interface available.

4.2 MPU6050

MPU6050 sensor module is a complete 6-axis Motion Tracking Device. It combines a 3-axis Gyroscope, 3-axis Accelerometer and Digital Motion Processor all in a small package. It uses an I2C bus for communication. It has an Auxiliary I2C bus to communicate with other sensor devices like 3-axis Magnetometer, Pressure sensor etc.

3-Axis Gyroscope

The MPU6050 consists of a 3-axis Gyroscope with Micro Electro Mechanical System(MEMS) technology. It is used to detect rotational velocity along the X, Y, Z axes as shown in below figure.

3-Axis Accelerometer

The MPU6050 consists of a 3-axis Accelerometer with Micro Electro Mechanical (MEMs) technology. It is used to detect angle of tilt or inclination along the X, Y and Z axes as shown in below figure.

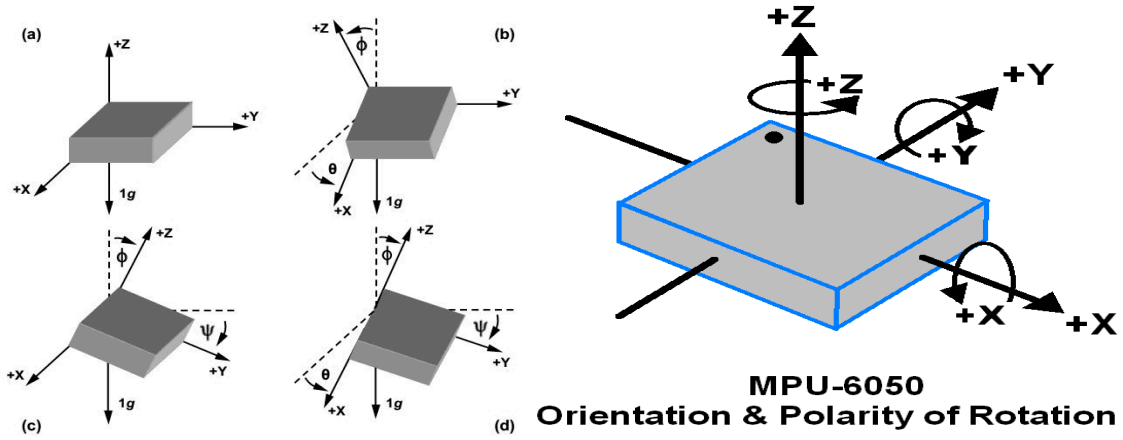


Fig 4.2 Orientation and Polarity of rotation.

4.3 MS5611

The sensor module includes a high linearity pressure sensor and an ultra-low power 24 bit ADC with internal factory calibrated coefficients.This barometric pressure sensor is optimized for altimeters and variometers with an altitude resolution of 10 cm. It provides a precise digital 24 Bit pressure and temperature value and different operation modes that allow the user to optimize for conversion speed and current consumption. A high resolution temperature output allows the implementation of an altimeter/thermometer function without any additional sensor. The MS5611-01BA can be interfaced to virtually any microcontroller.

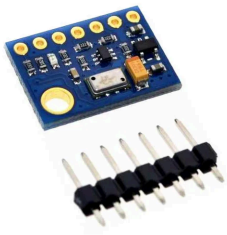


Fig 4.3 MS5611 Module

4.4 LoRa

It is a transceiver module with the operating frequencies between 137Mhz to 525Mhz. The module operates between1.8-3.6V with a high power output of 20dBm . Major advantage is that

it consumes very low power and is economical. It uses Chirp Spread Spectrum for modulation and uses I2C/SPI bus for communication with other modules.

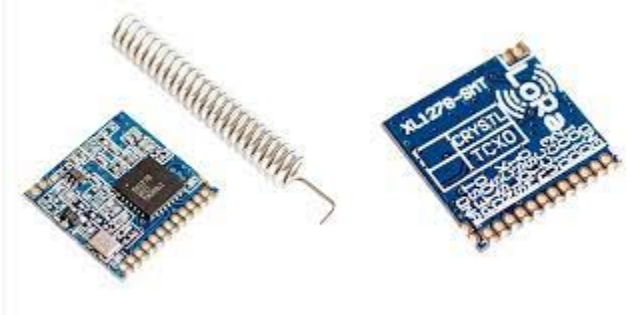


Fig 4.4 LoRa SX1278 module for communication

4.5 ATMEGA328P

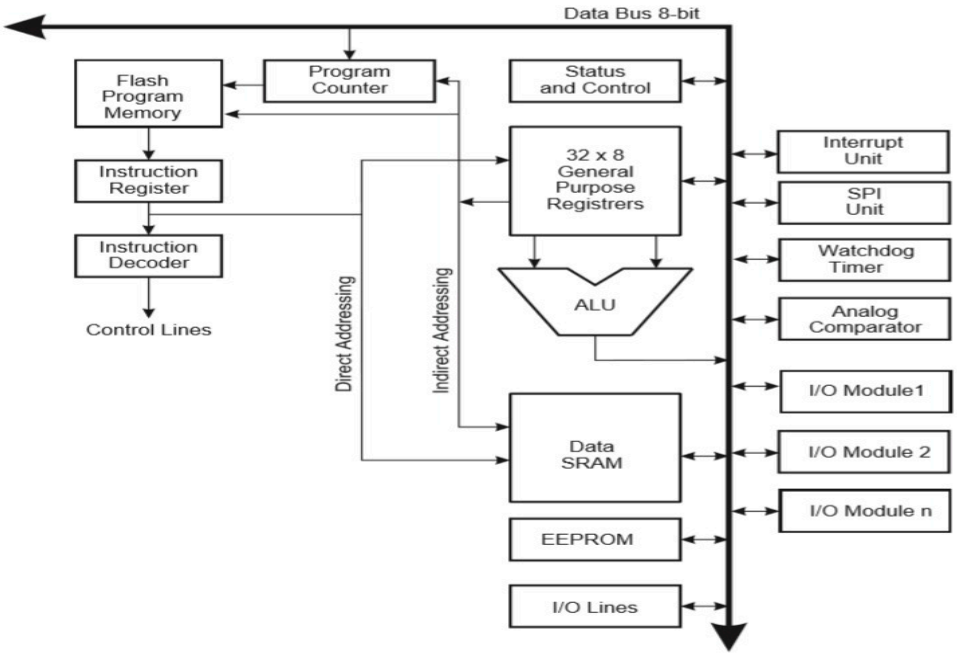


Fig 4.5 Block level Architecture of Atmega328PU

Atmega 328 PU is more easy to operate. It is a high-performance 8 bit microcontroller. It has a 32 KB ISP Flash memory with read-while-write capabilities, 1 KB EEPROM, 2 KB SRAM. it supports SPI,I2C and UART communication. The device operates between 1.8-5.5 volts.

4.6 SENSOR - DHT11

The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on

the data pin (no analog input pins needed). It's fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using the library, sensor readings can be up to 2 seconds old.

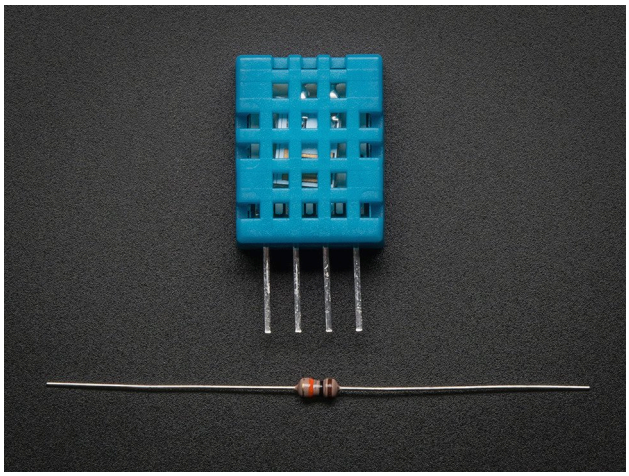


Fig 4.6 DHT11-Humidity sensor

4.7 UBLOX NEO 6M GPS module

The NEO-6M GPS module is a well-performing complete GPS receiver with a built-in 25 x 25 x 4mm ceramic antenna, which provides a strong satellite search capability. With the power and signal indicators, you can monitor the status of the module. The data backup battery helps the module save the data when the main power is shut down accidentally. Its 3mm mounting holes can ensure easy assembly on the aircraft, which thus can fly steadily at a fixed position, return to home automatically, and automatic waypoint flying, etc.

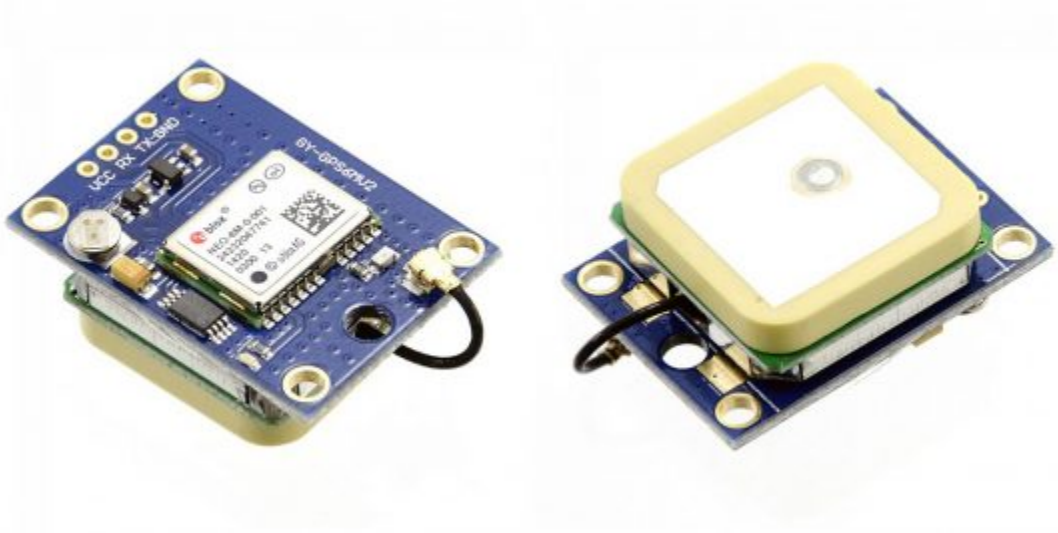


Fig 4.7 UBLOX NEO 6M GPS module GPS Sensor

Features

1) A complete GPS module with an active antenna integrated, and a built-in EEPROM to save 40 configuration parameter data.

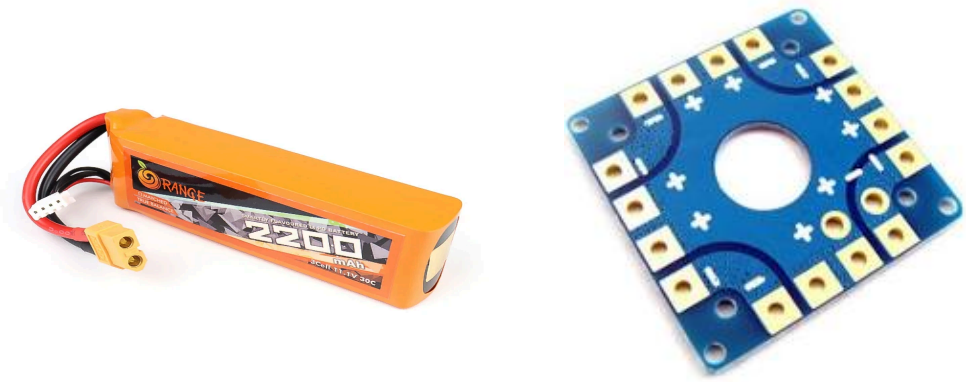
2) Built-in 25 x 25 x 4mm ceramic active antenna provides strong satellite search capability.

3) Equipped with power and signal indicator lights and data backup battery.

4) Power supply: 3-5V; Default baud rate: 9600bps.

5) Interface: RS232 TTL

4.8 POWER DISTRIBUTION



a) Lipo battery **b) Power distribution board**

Fig 4.8 Power Distribution

LiPo batteries are used to power up the ESC to drive the motors of the drone. LiPo batteries offer the advantage of high energy density in relation to the drone size and weight, with a higher voltage per cell. The Battery Eliminator Circuit of the ESC is used to operate the flight controller and the telemetry module. A 3s 2400mah LiPo is used to get a flight time of about 10 min.

4.9 DRONE FRAME

We have used the F450 for the drone frame. This gives the structure for the drone and also houses the flight controller, batteries, esc and motors. It is a 450mm quad frame built from quality materials. The main frame is glass fiber while the arms are constructed from ultra durable polyamide nylon.

Fig 4.9 FS450 Drone Frame

4.10 TRANSMITTER

A Drone Radio Transmitter is an electronic device that uses radio signals to transmit commands wirelessly via a set radio frequency over to the Radio Receiver, which is connected to the drone being remotely controlled. In other words, it's the device that translates the pilot's commands into the movement of the multirotor.

1. **Roll**. Moves the drone left or right in the air, literally "rolling" the drone.

2. **Pitch**. Tilts the drone forward or backward.

3. **Yaw**. Rotates the drone clockwise or counterclockwise, allowing you to make circles or patterns in the air.

4. **Throttle**. Controls the amount of power sent to the drone, which makes the drone go faster or slower.

Fig 4.10 FS-i6 Transmitter

4.11 BLDC MOTOR

This is a A2212 brushless outrunner motor specifically made to power Quadcopters and Multirotors. It is a 1000kV motor. It provides high performance, super power and brilliant efficiency. These motors are perfect for medium size quadcopters with 8 inch to 10 inch propellers.

Fig 4.11 1400KV BLDC motor

4.12 ESC

An **electronic speed control** (**ESC**) is an electronic circuit that controls and regulates the speed of an electric motor. It may also provide reversing of the motor and dynamic braking. Miniature electronic speed controls are used in electrically powered radio controlled models

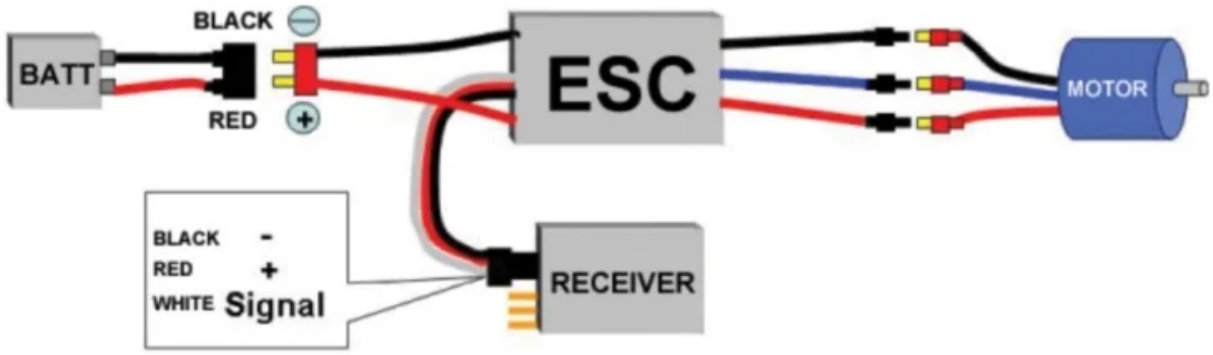


Fig 4.12 BLDC - ESC Block diagram

The above diagram shows how an esc can be used in a circuit. A signal from the flight **controller** causes the **ESC** to raise or lower the voltage to the motor as required, thus changing the **speed** of the propeller. An ESC controls the brushless motor movement or speed by activating the appropriate MOSFETs to create the rotating magnetic field so that the motor rotates. The higher the frequency or the quicker the ESC goes through the 6 intervals, the higher the speed of the motor will be

CHAPTER 5

FLIGHT TESTING

The Flight controller with the F450 frame was flight tested multiple times, in accordance with regulations enforced by DGCA. The aircraft flew stable in moderate winds and was tested in varying atmospheric conditions. The aircraft reached an approximate altitude of over (well below it's ceiling)LOS.

Fig 5.1 Images of the UAV in flight in carrying weather conditions

Fig 5.2 Aerial view from the quadcopter

CHAPTER 6
CHALLENGES

6.1 Mechanical noise: An unbalanced propeller refers to a propellor whose centre of gravity if slightly offset from the axis of rotation. Unbalanced propellers will make the accelerometer and gyroscope to produce noise that causes the motors to jerk. A shaky motor will affect the stable flight of the drone. This is reduced by proper balancing.

6.2 Temperature: the calibration of gyro is significantly affected by the temperature. For each +/- 5^{o} variation in temperature, the offset values must change.

6.3 EMI : While designing the pcb ,EMI radiation are needed to be considered. This affects the overall performance of the circuit. To minimise this effect the traces are routed with 45^{o} bends instead of sharp 90^{o} bends.

6.4 Electrical noise: This noise is generated when the UAV motors are driven. Many amps of current are used to drive the motors. This causes voltage spikes on ground and power lines of the circuit. Therefore, spikes appear on smooth transitions of the UAV. This can be registered as false readings in the accelerometer and gyroscope. To minimise this effect, high esr(equivalent series resistance) capacitors are used which filters out all the abrupt voltage spikes.

6.5 Oscillations during flight: Improper tuning of P, I and D gains will result in the destabilisation of the drone. This causes oscillations during flight time and thereby resulting in an unstable flight. This can be reduced by calibration of the P, I and D values before each individual flight. Although the perfect values can never be obtained, multiple calibrations can lead to a stable flight.

CHAPTER 7
CONCLUSION

Successfully designed and implemented airworthy Avionics for quadcopter flight control, specific to the application of profiling the atmosphere. The STM32 F1 series based flight controller is a cheap alternative to commercially available flight controllers while also allowing for the access of internal signals. These internal signals can be used for further data analysis. The aircraft was tested in different atmospheric conditions and the overall response was found to be satisfactory. The telemetry module transmits data to the ground station, which can be received on any PC.

References

1. Ross t. Palomaki, Nathan T. Rose, Michael Van Den Bossche, Thomas J. Sherman, and Stephan F. J. De Wekker, "*Wind Estimation in the Lower Atmosphere Using Multirotor Aircraft"*, Vol.34, Issue.5, March 2017 [Accessed September 28 2020]
2. T. A. Mai, T. S. Dang, D. N. Anisimov and E. Fedorova, "Fuzzy-PID Controller for Two Wheels Balancing Robot Based on STM32 Microcontroller," *2019 International Conference on Engineering Technologies and Computer Science (EnT)*, 2019, pp. 20-24, doi: 10.1109/EnT.2019.00009.
3. B. Sumantri, N. Tamami, Y. B. Nuraga and B. Kurniawan, "Development of a Low-Cost Embedded Flight Controller for Quadcopter," *2020 International Electronics Symposium (IES)*, 2020, pp. 233-238, doi: 10.1109/IES50839.2020.9231564.]
4. K. Priandana, M. Hazim, Wulandari and B. Kusumoputro, "Development of Autonomous UAV Quadcopters using Pixhawk Controller and Its Flight Data Acquisition," *2020 International Conference on Computer Science and Its Application in Agriculture (ICOSICA)*, 2020, pp. 1-6, doi: 10.1109/ICOSICA49951.2020.9243289.
5. M. Waliszkiewicz, K. Wojtowicz and Z. Rochala, "Experimental method of controller tuning for quadcopters," *2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2019, pp. 165-170, doi: 10.1109/MetroAeroSpace.2019.8869573.

URLs

1) https://github.com/rogerclarkmelbourne
2) https://www.udemy.com/course/microcontroller-embedded-c-programming/
3) https://www.udemy.com/course/mastering-microcontroller-with-peripheral-driver-development/
4) https://www.udemy.com/course/microcontroller-programming-stm32-timers-pwm-can-bus-protocol/
5) https://youtu.be/AN3yxIBAxTA
6) https://electronoobs.com/eng_robotica_tut6.php

LIST OF FIGURES

1. Fig 2.1 Timer section of STM32f103 3
2. Fig 2.2 32 bit Timer2 register 4
3. Fig 2.3 PPM Frame for 6 channel receiver 5
4. Fig 2.4 Implementation of complementary filter in code 6
5. Fig 2.5 Block Diagram of Complementary Filter 7
6. Fig 2.6 Complementary filter data obtained from the IMU 7
7. Fig 2.7 PID controller block diagram 9
8. Fig 2.8 Step response of a generic PID controller 10
9. Fig 3.1 Bottom Layer of DOT PCB 13
10. Fig 3.2 Top Layer of DOT PCB 13
11. Fig 3.3 Schematic of the flight control unit 15
12. Fig 3.4 PCB Layout 15
13. Fig 3.5 Printed PCB 16
14. Fig 3.6 3D Software render of PCB 16
15. Fig 3.7 Avionics bay of the Drone 17
16. Fig 3.8 Schematic of the telemetry module 17
17. Fig 3.9 18
18. Fig 4.1 Architecture of STM32f103 19
19. Fig 4.2 Orientation and Polarity of rotation 21
20. Fig 4.3 MS5611 Module 21
21. Fig 4.4 Lora SX 1278 module for communication 22
22. Fig 4.5 Block level Architecture of Atmega 328P 22
23. Fig 4.6 DHT11-Humidity sensor 23
24. Fig 4.7 UBLOX NEO 6M GPS module GPS Sensor 23
25. Fig 4.8 Power Distribution 24
26. Fig 4.9 F$450 Drone Frame 25
27. Fig 4.10 FS-i6 Transmitter 26
28. Fig 4.11 1400KV BLDC motor 26
29. Fig 4.12 BLDC - ESC Block diagram 27

30. Fig 5.1 Images of the UAV in flight in carrying weather conditions 28
31. Fig 5.2 Aerial view from the quadcopter 29