MARMARA UNIVERSITY
FACULTY OF ENGINEERING

FLIGHT CONTROLLER DESIGN FOR QUADCOPTERS

Osman BÜYÜKKAĞNICI – Ali Eren TÜRK

GRADUATION PROJECT REPORT

Department of Electrical and Electronics Engineering

Supervisor

Doç. Dr. Fulya ÇALLIALP

ISTANBUL, 2023

MARMARA UNIVERSITY
FACULTY OF ENGINEERING

FLIGHT CONTROLLER DESIGN FOR QUADCOPTERS

by

Osman BÜYÜKKAĞNICI – Ali Eren TÜRK

January 20, 2023, Istanbul

SUMBITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF

BACHELOR OF SCIENCE

AT

MARMARA UNIVERSITY

Signature of Author(s) ....

Department of Electrical and Electronics Engineering

Certified By Doç. Dr. Fulya ÇALLIALP

Project Supervisor, Department of Electrical and Electronics Engineering

Accepted By ....

Head of the Department of Electrical and Electronics Engineering

ACKNOWLEDGEMENTS

We want to thank our supervisor Doç. Dr. Fulya ÇALLIALP for guiding us throughout our university life and for his support and this research. We also thank Dr. Alper ŞİŞMAN who let us use his laboratory and stuff.

January, 2023

Ali Eren TÜRK – Osman BÜYÜKKAĞNICI

TABLE OF CONTENTS

ABSTRACT

The use of Unmanned Aerial Vehicles is increasing in the world and in our country. They are used for various applications both in the military and civil areas. UAVs eliminate the risk of human loss in the defence field. In the project, the aim is to design a flight control card for a Drone, which is a multi-propeller type of Unmanned Aerial Vehicles.

Our flight controller board contains original and open-source software that is accessible and more understandable. The working principles of an important part cannot be understood by the designer/manufacturer of the drone. In our study, it is aimed to provide knowledge of the software and electronic circuits in the background by preventing for this problem.

LIST OF SYMBOLS

ABBREVIATIONS

PWM: pulse width modulation

MCU: micro controller unit

I2C: inter integrated circuit

UAV: unmanned aerial vehicle

PID: proportional integral derivative

PCB: printed circuit board

ESC: electronic speed controller

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Unmanned aerial vehicles have succeeded to take their place for daily life in many areas, by facilitating people's lives and contributing to the defense industries and economies of countries. The investments of countries in the field of unmanned aerial vehicles continue at an accelerating rate. USA, Israel and China, have come to the fore in recent years. Our state is also strong like them. In this sense, we wanted to carry out our project in order to support production studies by creating an easily accessible, understandable open source on topics such as how to use the necessary electronic equipment together, how to develop the software and control algorithm for our country.

Usually, the relevant people buy the control card, and the working principles of an important part cannot be understood by the designer/manufacturer of the drone. In general, Pixhawk brand control cards are used with their own ready software. With our study, it is aimed to prevent this situation and to provide knowledge of the software and electronic circuits in the background.

Our board, acceleration sensor, motors and drivers, remote control receiver, etc. required for the quadcopter to fly will ensure all components work as a whole and enable us to fly our aircraft with a remote control.

Our flight control card, which will be designed as a result of the project, will contain an STM32 MCU with ARM M4 architecture. With this project, it is aimed to create a unique flight control algorithm, to pave the way for student activities in this field, to make these works more understandable, and to make things easier by providing readable and accessible open sources.

1.1. Thesis Content

The content should include the following sections:

- How the transmitter and receiver communicate?
- How to read angle values from the MPU6050 sensor?
- How are the motors served by ESC?
- Overall circuit diagram

2. RESEARCH OBJECTIVE

Technically, ARM M4-based MCU is programmed in our control card that is for manual flight. The goal of our project is to use a drone with a national and original flight control algorithm in an understandable way.

To understand how the relevant sensors do, modules and all electronic parts work together, we have to ensure:

1- To communicate between the control receiver and the microcontroller,

2- Reading data from accelerometer and barometer sensors,

3- According to the commands received from the remote controller, the sensor data is passed through the PID algorithm that we will create.

4- Integrating the codes into the FreeRtos operating system,

5- Designing the PCB of the control card where the ESC motor drivers, sensors and MCU will communicate and work together.

and produce,

6- To complete the project by feeding the control card and drivers with the battery.

The main purpose of the project; to decrease usage of foreign-sourced ready-made control cards.

First of all, we aim to keep the money in the country and then increase the export income through this project.

3. RELATED LITERATURE

Drones are used for different purposes in the military field, such as mine detection, enemy detection. In addition, they are used in various sectors from agriculture to photography, cargo transportation to health in a civilian sense.

While the use of drones has increased professionally, it has also reached widespread use as a hobby. As a result of our observations, when people interested in aviation want to develop their own drones, they actually buy control cards, the contents of which they do not understand, which are like a closed box for the user. They also participate in the unmanned aerial vehicles competitions organized by TUBITAK by using these ready-made control cards. This situation greatly reduces the original value of most of the projects participating in the competition and causes people not to have access to the information that is the cornerstone of their work. In addition, although some cards allow access to the software inside, these codes are insufficient because they are not in a very comfortable state to understand. On the other hand, although there are websites that develop such projects with Arduino and clearly explain the algorithms, the professionalism and industrial aspects of such projects are insufficient because they are made with Arduino.

Mr. Joop Brookking has a project called YMFC-Al which is an Arduino uno based quadcopter. We benefit from his documents about PID controlling.[1]

Mr. Joop Brookking has also a project called YMFC-32 which is an STM32F103 based flight controller, but its codes are written in Arduino platform the only different thing is the microcontroller. We will do this project with higher performance MCU than STM32F103.[9]

Mr. Berkay has a YouTube channel called “Berkay ile elektronik” in this channel he tells how to code Arduino to build a quadcopter. Actually, he just translated the things that Joop Brookking did to Turkish, but we also benefit from him as well. [10]

By Shenyang University, a study is done, and its essay is published on IEEE website which is about designing PID controller for flight controller and choosing reasonable parameters, it is a good source for us to build an algorithm for PID.[11]

We have started this project, which is a solution to these problems encountered by creating a flight control board with a processor that will run the flight control software more efficiently, unlike those made with Arduino. Different from the closed source ready-made control cards, we will develop a product where people can easily understand its hardware and software.

4. DESIGN

4.1. Realistic constraints and conditions

4.1.1 Environmental Issues

While we are making flight test to observe the stability of the drone, although taking precautions such as fail-safe feature and selecting suitable place for flight there may be unexpected events. Drone may fall during flight and can cause damage on objects around flight area.

4.1.2 Health and Safety

First, since the motors that rotate the propeller are high speed and the propeller is sharp, you should stay away from the drone during the movement. It should not be approached without making sure that the motors have come to a complete stop. A protective mold will be printed on the outside to prevent the electronic board from getting wet or damaged. Drones cannot be flown in public areas. It can only be used in places designated by the state for UAVs and with permission.

4.1.3 Sustainability

Designing a national hardware that contains open source and understandable software for rapidly developing UAV, will contribute to the next studies. It shows that is a sustainable academic study.

4.1.4 Manufacturability

In the prototype production of our project, a PCB with sensors such as MPU6050, barometer, rf receiver will be designed. This PCB is compatible with STM32F407 MCU and they will work together. In terms of cost, it is not more expensive than the cards purchased with ready-made software. Because the actual value originates from the software embedded in it.

4.1.5 Social, Economic, Political Issues

Our country, in the field of unmanned aerial vehicles, made progress. In order to make these progresses further designing local control cards are important to reduce costs. If the flight control board is supported, it will be source for the next studies about UAVs.

4.2. Cost of the design

Products	Price
STM32F407 Development Kit	700 TL
MPU6050 Sensor	45 TL
Li-Po Battery	1200 TL
Electronic Speed Controller x4	600 TL
Brushless Motor x4	600 TL
Remote Controller	1150 TL
Power Distribution Board	115 TL
Drone Frame and Propellers	250 TL

Table 1.1

We bought all the material listed above except the development kit; it costs approximately 4660 Tl. We already have development kit. For battery we need a charger, but we will not buy, it instead we will use battery charger of a UAV team in our university.

4.3. Engineering Standards

4.3.1 Communication Standards

I2C

The I2C communication protocol is used in this project. Between the MPU6050 and the microcontroller communication is set by I2C (inter integrated circuit) in order to take raw data about the acceleration and gyro value of the drone.

This protocol uses 2 pins which are SDA (for data transmission) and SCL (for clock). It allows multiple slave devices communicate with multiple masters. The clock signal is always generated by master device. Let's have a look at the message frame of this protocol.[2]

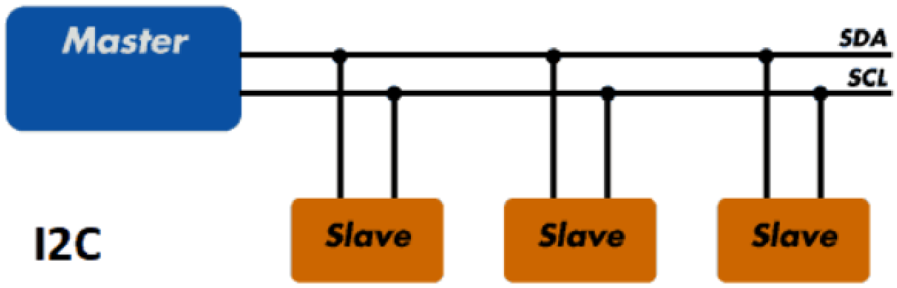

Figure 1

There two types of frames:

- First one is the address information of the devices which will receive the message.
- Second one is data frame to be sent from the master or slave.

After the SCL line is pulled low, data is places on SDA line and after the SDA line is pulled high data is sampled by the receiver device.

Start Condition:

To start the address frame, master device pulls SCL high and pull SDA low. This process warns all the slave devices about the communication is starting.

Address Frame:

Address frame is the first frame in this I2C sequence. Firstly, the address information (7 bit) is sent by SDA line then 1 bit for indicating Read (0) or Write (1) is sent. The nineth bit of the frame is NACK/ACK bit. This bit exists in both address frame and data frame. After sending the first 8-bit, receiver takes control of SDA line if the received byte is valid for receiver, it sends ACK, else it sends NACK bit.

Data Frame:

After sending the address frame, data is started to send. While the master device is sending clock signal, according to read/write bit master or slave puts data on SDA line.

Stop Condition:

After sending all frames, master device creates a stop condition. Stop condition occurs pulling SDA up when SCL is high. In normal data sending process, to avoid misleading of stop condition the value on SDA should not be changed when SCL is high.

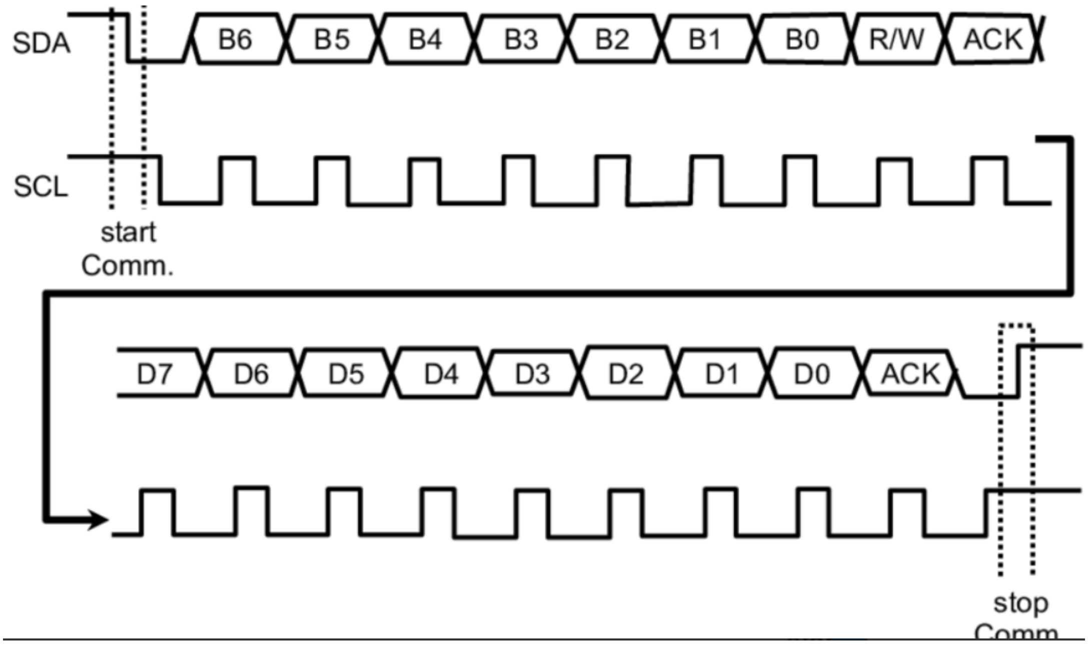

Figure 2

PWM

PWM sets the opening and closing times of waves produced in a system. Thus, the power to be supplied to the system can be adjusted. PWM is sometimes obtained with microcontrollers as well as obtained in analog form. The PWM signal is made by the switching method.

It must be known how long the PWM signal pulses pull the wave to "high" level and how long to "low" level in a period of the wave. The time that the wave stays at the "high" level in the

PWM signal is the pulse width, the percentage of this pulse width to a period is called the "Duty Cycle". The output changes between 100% and 0% as a duty cycle for a period.

The image below explains:

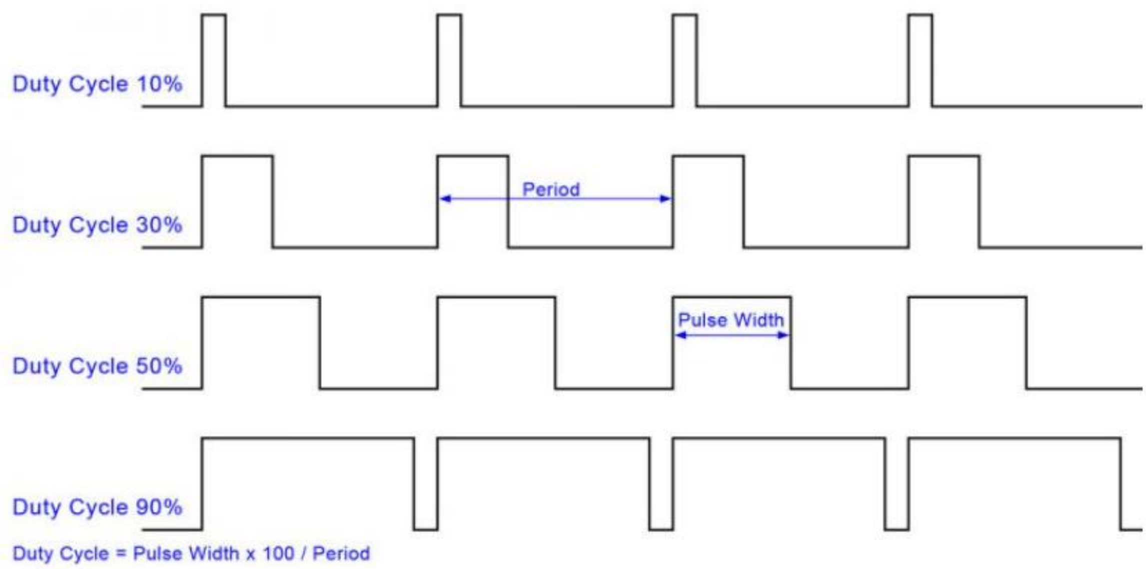

Figure 3

According to the pulse width in the PWM signal, we get results from the output as if we had an analog signal. However, if we measure it with an oscilloscope, we see that this analog-like signal is a digital signal on the screen. Various applications are available with PWM. The brightness of a led and the speed of a DC motor can be adjusted with a MOSFET in this way. Since we will adjust the speed controls of brushless motors according to the commands coming from the remote controller in our project, we do the motor driving processes by generating PWM signals.[3]

4.3.2 Software Standards

In embedded C programming language is the most common one because of its pointer mechanism. We can reach different bit position of register content of microcontrollers via pointers in C programming language.

We are using HAL library to program our chip. Which is released by ST company. As name indicates it is a hardware abstraction layer it allows us to use the hardware with peripheral functions by abstracting us from the details of the hardware. Apart from that, it supports many products, so it is easy to switch between different microcontrollers.

4.4. Details of the design

We are using STM32F407vg board to develop software because it is high performance device and if we want to add some extra features such as autonomous flight in future, this MCU will be sufficient for us. It’s core operating at frequency max 168 MHz This feature provides speed for processing data. It also has floating point unit; with floating point unit we can get more accurate results. Apart from that it has 192 Kbytes of SRAM it is enough for us because we are not collecting data other than MPU6050 and remote controller.

Power consumption is crucial in UAVs because flight duration is directly related with battery life. Therefore, low power modes of our microcontroller can be used to decrease power consumption. There are three modes of operation sleep, stop and standby modes.

We will use timers for two different purposes. One of them is detecting PWM pulse width from receiver and other one is to create PWM signal for motors. There 17 timers in chip which is good enough for us.

I2C protocol is used for communicating with MPU6050. This protocol has 2 different speed 100 kbps and 400 kbps the MCU supports both speed we will use 400 kbps speed to get better results.

As software tool we are using STM32 Cube IDE which allow us to make all the configuration like selecting I2C speed, clock frequency, assigning pins etc. on a GUI. So, we don’t have to write all codes about configuration it is done by IDE automatically.

The other tool is STM studio, we are importing globally defined variables from .elf file of project into this program then when we run the program and program sees the variables via ST-Link SWD interface, so we can watch the values of imported variables.

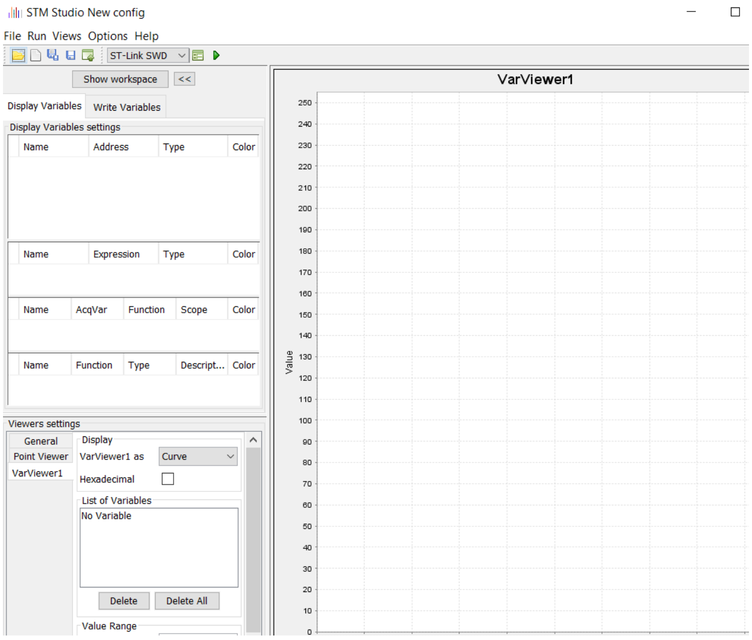

Figure 4

Let's have look at how did we choose battery and motor. Two points that we should pay attention to before making the choices of our battery and motor. These were the thrust and the hover duration. Thrust is simply the lifting force applied by our propellers to the Drone.

As a result of our research, the thrust to be received from the engines should be twice the weight of the drone. When choosing an engine and propeller, it should be treated to increase the thrust. For hover time, the battery and motors to be selected will be important. when we want to have hover for 10 minutes, battery should be selected based on followings:

The expected weight of our vehicle is 1220 grams. Therefore, the thrust we need to get from the engines is at least 2440 grams. We can get a maximum of 860 grams of thrust from our A2212 model engine. A total of four engines can receive thrust capable of carrying 4 x 860 = 3440 grams. Our engines are the minimum required it gives thrust that can carry 1000 grams more mass than thrust. Thanks to this excess, useful loads according to the purpose it can also be moved.[6],[7],[8]

İf we have a battery which is 4000mAh and 3S.

- The current supplied by battery in 1 minute $4Ah * \frac{60dk}{1h} = 240Adk$

- Using battery at full capacity reduces its life so assume we use its %80 ,240%80=192Adk
- Dividing the total weight of the drone we learn how much minimum thrust we must get from 1 motor (304 gr)
- According to data sheet of motor, to get 304 gr thrust motor needs approximately 4.5A current.
- Total required current is 18 A.
- Hover duration is $\frac{192Adk}{18A} = 10.67\ dk$

5. METHODS

Remote controller and Microcontroller communication

We are using FlySky fs 6 model transmitter and FS i6b model receiver. On the receiver side we are using PWM signals to give information to microcontroller. Receiver sends 50 Hz signal with its duty cycle changes from 1000 ms to 2000 Ms. Receiver supports up to 6 channel we will use 4 of them for pitch, roll, yaw, arm.

In the microcontroller side, one of the timers is used in input capture mode (feed by internal clock) to detect the width of PWM pulse sent by receiver.

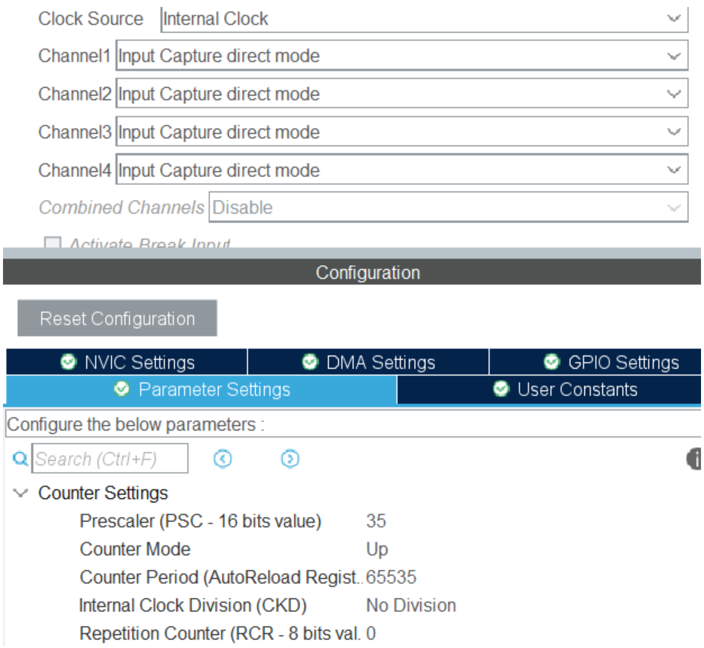

Figure 5

And the interrupt for input capture callback is activated from NVIC settings

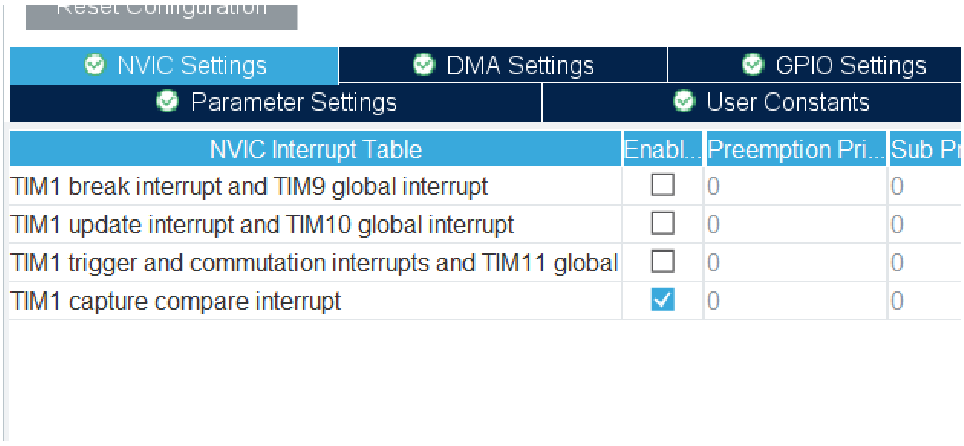

Figure 6

At first, channels triggered by rising edge of the input to the related pins and interrupt service routine runs the HAL_TIM_IC_CaptureCallback() .

```
65 void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
66 {
67     if(htim->Instance == TIM1)
68     {
69         switch(htim->Channel)
70         {
71         case HAL_TIM_ACTIVE_CHANNEL_1:
72             if((TIM1->CCER & TIM_CCER_CC1P)==0)
73             {
74                 ch1_rising = TIM1->CCR1;
75                 TIM1->CCER |= TIM_CCER_CC1P;
76             }
77
78             else
79             {
80                 ch1_falling = TIM1->CCR1;
81                 pre_ch1 = ch1_falling - ch1_rising;
82                 if(pre_ch1 < 0)pre_ch1 += 0xFFFF;
83                 if(pre_ch1 < 4000 && pre_ch1 > 2000)ch1=pre_ch1;
84                 TIM1->CCER &= ~TIM_CCER_CC1P;
85
86             }
87             break;
```

Figure 7

In the 67. Line software checks whether interrupt caused by TIM1 or not. Then program goes to related channel (in line 71)

In 72 nd line it checks whether triggering is due to rising edge or not if it is ch1_rising variable takes the value of counting register (line 74) then change the polarity selection to falling edge (line 75)

Now polarity is falling edge when pulse down to zero related channel is triggered again and ch1_falling variable takes the value of county register then (ch1_falling - ch1_falling) calculated if it is negative add 65535 to the result. And lastly change the polarity to rising edge (line 84) .[4]

Reading Data from MPU6050

We are reading data from MPU6050 via I2C protocol. This sensor provides gyro and accelerometer values as a raw values .It means that we have to process these data to get meaningful data.

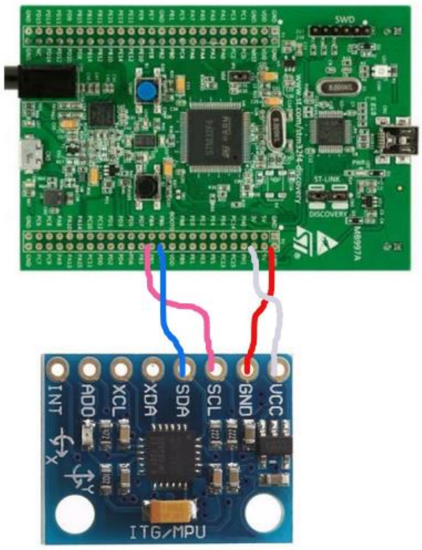

Figure 8

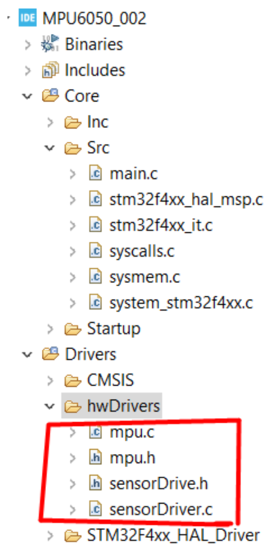

Figure 9

We create library for sensor. This library contains all register addresses and requires functions for getting data from the sensor such that.

These codes are from application layer (main.c) firstly we initialize sensor with gyro sensitivity of 500deg/sec and accelerometer sensitivity of 8g.Then offset values are calculated ,while this values are calculating sensor located in a smooth place to get better result .In while loop we have a function called mpu_Update() this function passes the raw data from a filter and gives pitch and roll angles of the device.[5]

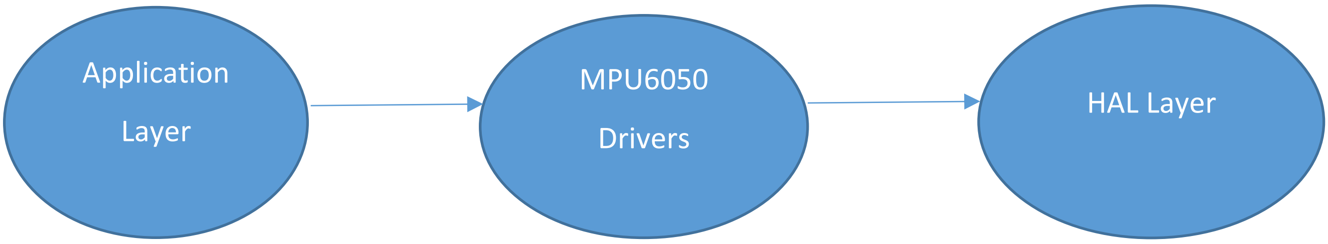

Figure 10

```
MPU6050_initialize(&pSensor,FS_500,AFS_8G);
CalcOffSet(&pSensor);
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
    mpu_Update(&pSensor);

}
/* USER CODE END 3 */
}
```

Figure 11

These codes are from application layer (main.c) firstly we initialize sensor with gyro sensitivity of 500deg/sec and accelerometer sensitivity of 8g.Then offset values are calculated ,while this values are calculating sensor located in a smooth place to get better result .In while loop we have a function called mpu_Update() this function passes the raw data from a filter and gives pitch and roll angles of the device.

When this sensor is worked under vibrations, the data we get is so noisy that PID code is generating wrong power for the motors. To avoid this issue, we applied filter to sensor data.

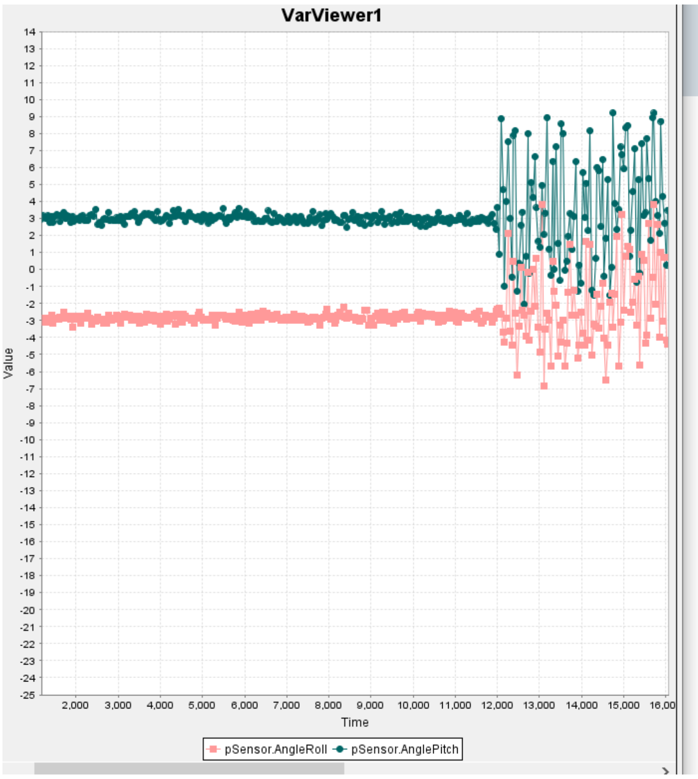

Figure 12

Figure 13

As you can see, in figure 12th millisecond motors are started to tun so MPU6050 data are so noisy.

In figure 13 the black datas are filtered data, it is obvious that filtering process eliminated most of the noise.

Driving Motors

We use ESC to serve motors. There are three sets of cables coming out of the ESC; these are +5V and ground signal line. This feature of the ESC is called the Battery Eliminator Circuit, and this feature eliminates the need for a separate battery for the microcontroller. With this, the ESC provides regulated 5V.ESC works in 50 Hz like servo motors. So, when serving motors, we should produce PWM signal whose duty is in the range of 1000 to 2000 ms.

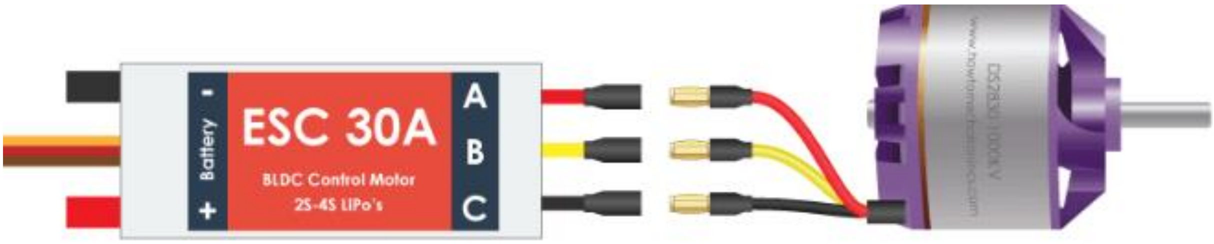

Figure 14

```
  HAL_TIM_PWM_Start(&htim8, TIM_CHANNEL_1);
  HAL_TIM_IC_Start_IT(&htim1, TIM_CHANNEL_1);
  //__HAL_TIM_SET_COMPARE(&htim8,TIM_CHANNEL_1,0);
  //HAL_Delay(3000);
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */

      __HAL_TIM_SET_COMPARE(&htim8,TIM_CHANNEL_1,newch1);


  }
  /* USER CODE END 3 */
}
```

Figure 15

First, we start two timers one of them to produce PWM signal and other one is for detecting signals from receiver. Then in the while loop we continuously writing the value which is sent by transmitter as width of the PWM signal. Because we are using interrupt for reading receiver, we don't call any function about receiver in while loop.

PID
Proportional, Integral, and Derivative is referred to as PID. It is a control method frequently used to stabilize and control the motion of drones and other autonomous devices. Based on the discrepancy between the desired setpoint and the actual measurement, the PID controller modifies the system's output. A drone's attitude (roll, pitch, and yaw) and altitude (vertical position) can be managed by a PID controller.

We create our PID algorithm according to following scheme.

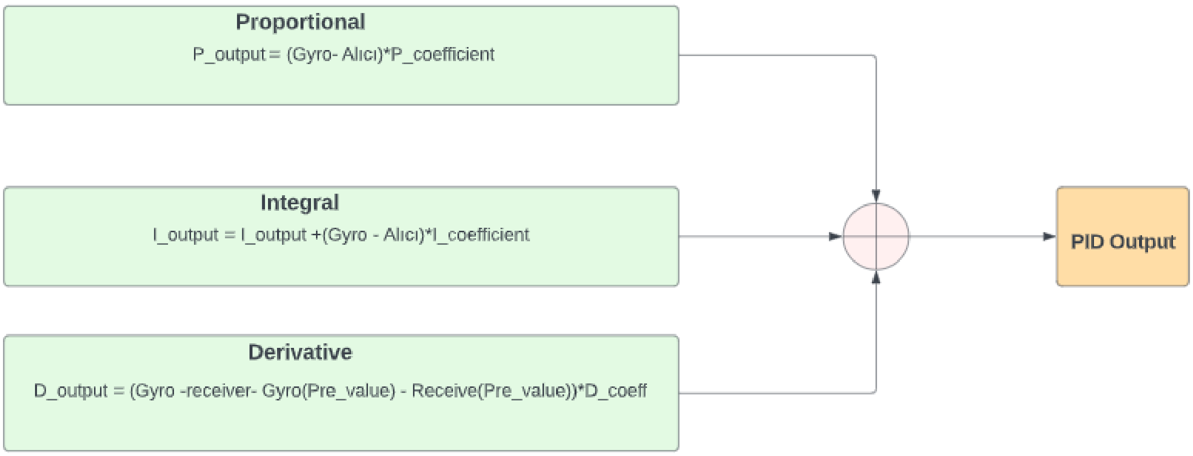

Figure 16

To provide steady and responsive control, PID tuning for a drone entail changing the proportional, integral, and derivative gains. A general method for fine-tuning the PID controller for a drone is as follow:

Start with modest gains: Set all the gains (P gain, I gain, and D gain) to modest or even zero amounts at first. This will make sure that the drone responds smoothly and steadily at first, but it can take a while for it to reach the required setpoint.
Adjust the proportional gain (P gain) by progressively raising it while monitoring the drone's response. The drone will react to errors more aggressively and get to the setpoint more quickly with a higher P gain. However, if it is increased excessively, oscillations and overshooting may occur.
Tune the integral gain (I gain): The I gain aids in the correction of biases and steady-state errors. It incorporates the overtime accumulated mistake. To adjust the I gain: Gradually increase the I gain while monitoring the drone's behavior. Continue raising the I gain if you see a noticeable decrease in steady-state errors until the response becomes too sluggish or unreliable. Reduce the I gain somewhat to strike a balance between error correction and system stability whenever instability or a sluggish response is noticed. Adjust the derivative gain (D gain) to attenuate the response, lessen overshoot, and eliminate oscillations.

The D gain can be tuned as follows: As you watch the drone's response, raise the D gain. Reduce the D gain if the response starts to become excessively sensitive or shows high-frequency oscillations. Usually, the D gain is less than the P gain and the I gain. Iterative improvement: After fine-tuning each gain separately, go back and adjust them all at once. While keeping an eye on the drone, make little modifications to each gain. Finding the ideal balance of responsiveness, stability, and error correction is made easier by this iterative approach.

Once the PID gains have been adjusted, evaluate the drone's performance under various flight scenarios, such as hovering, challenging maneuvers, or altitude changes. Keep an eye on the answer and adjust as required.

PCB

For electronic circuits, designing a printed circuit board (PCB) has a few benefits over jumper wiring. Following are some advantages of PCB design:

Enhanced Reliability: When compared to jumper wiring, PCBs offer a more dependable and durable option. They are made to manage high-frequency transmissions, reduce noise, and lessen the possibility of signal interference or component crosstalk. Additionally, PCBs provide greater resistance to mechanical stress and vibration, ensuring long-term dependability in a variety of applications.

Compact and Space-Efficient: PCBs enable circuit designs that are both compact and space-efficient. PCBs minimize the need for long wires or jumper cables that can take up a lot of room in the circuit by combining components and their connections on a single board.

Better Signal Integrity: PCBs are built with regulated impedance traces and appropriate ground planes, which maximize signal integrity and reduce signal loss or distortion. This is crucial for delicate analog circuits or high-speed digital circuits where signal quality is paramount. It might be difficult to maintain constant impedance and minimize signal deterioration with jumper wiring.

Ease of Manufacturing: Manufacturing is simple since PCBs can be produced utilizing automated techniques, which improves the effectiveness and efficiency of mass production. Once the PCB design is complete, it is simple to copy and maintain consistency, which minimizes human error and guarantees high quality. Jumper wiring, on the other hand, frequently involves human effort, increasing the chance of errors and differences in the finished output.

Using the KiCad software, we designed the PCB layout based on the schematic. Then components are placed, and traces are routed to establish electrical connections.

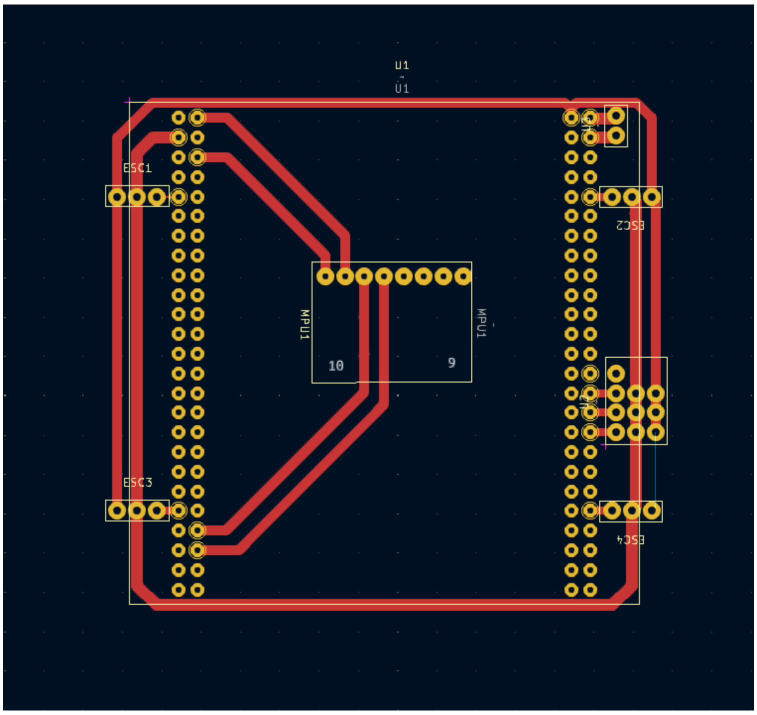

Figure 17

6. RESULTS AND DISCUSSION

We initially started with research on the principle of drone flight.
We decided what the peripherals would be, with the STM32F407 MCU in the center. As a result, we created the following scheme.

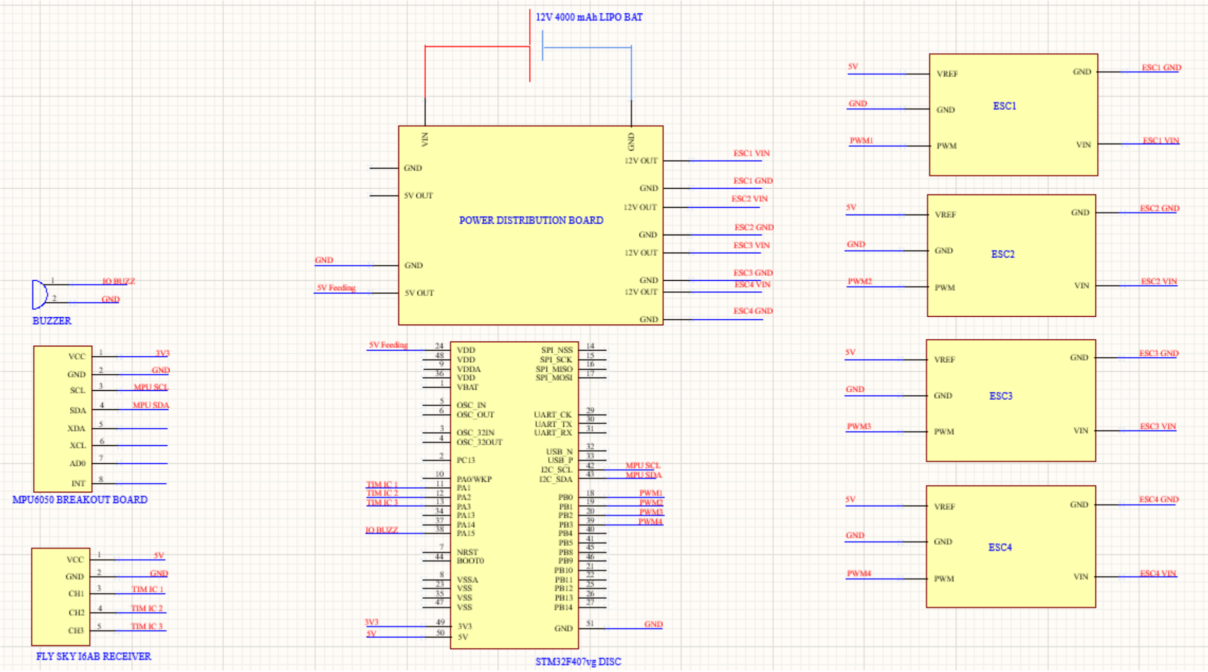

Figure 18

In this direction, we had to choose materials and shop. Before shopping, we calculated the thrust required by the drone according to the time that it will fly and material weights. After completing the material shopping, we started to work in the following order.

1-) Receiving data from the remote controller:
We calculated the duty cycle according to the commands on 50Hz frequency signals from our remote. With this, we examined the graphics in the simulation before driving motors.

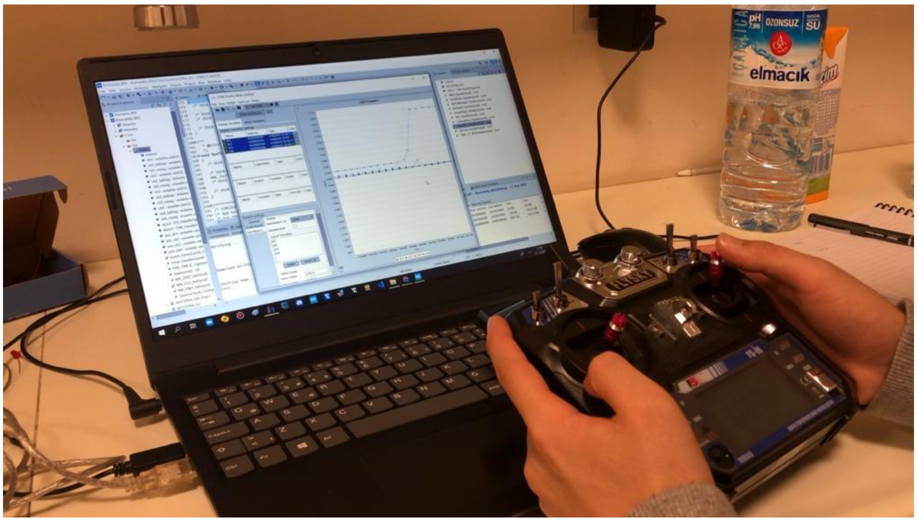

Figure 19

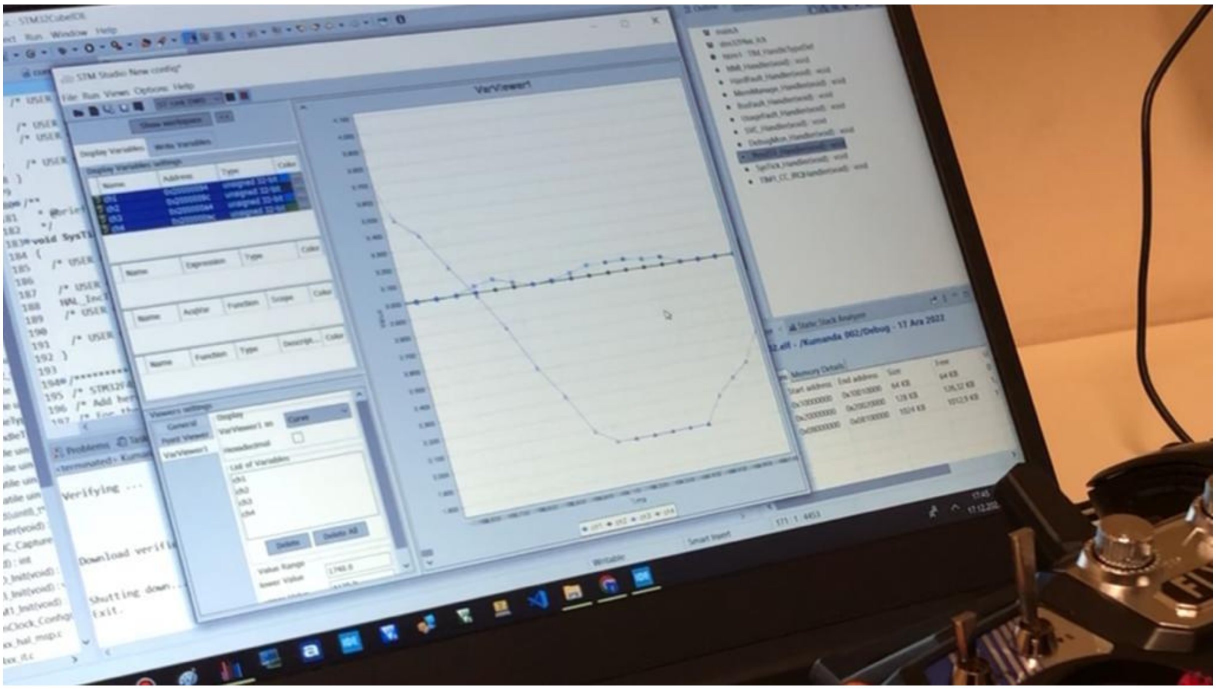

Figure 20

2-) We worked with the motor with the commands from the remote. We fed the ESC with the battery. We connected the three wires of the ESC to the motor. We rotated the motor according to the remote-control data with STM via the PWM data pin of the motor.

3-) After learning how to use the MPU6050, we started working. We needed to know the I2C protocol. Then the necessary connections were made with STM32. We performed data reading from MPU6050. We watched this data on STM Studio with the help of graphics.

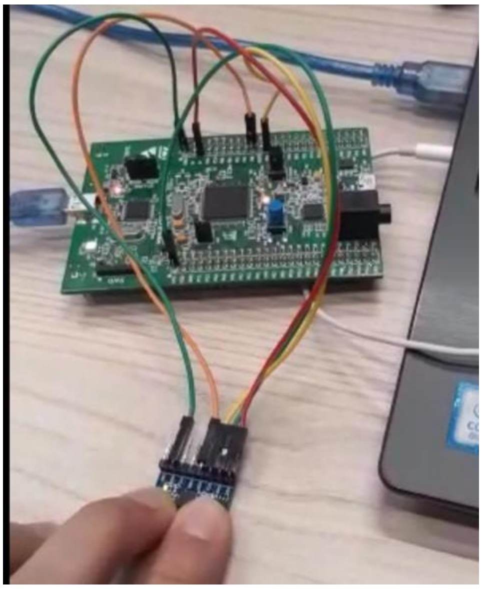

Figure 21

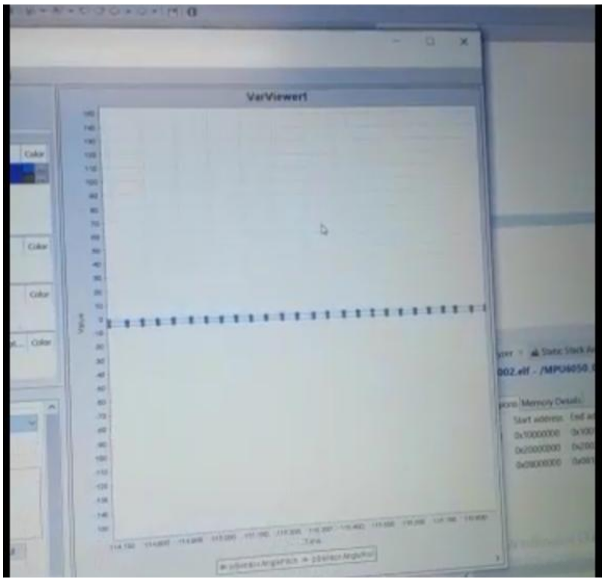

Figure 22

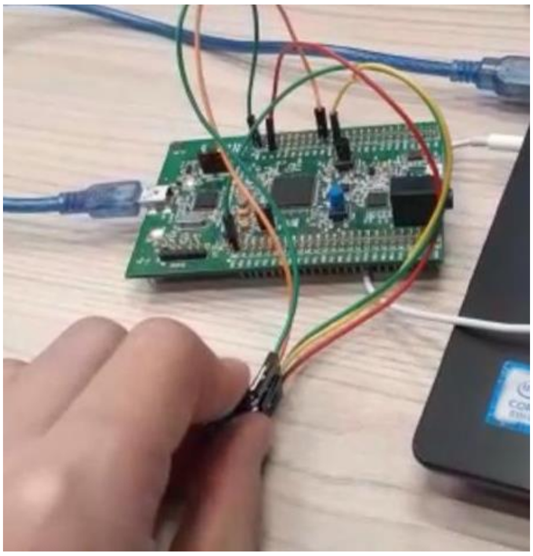

Figure 21

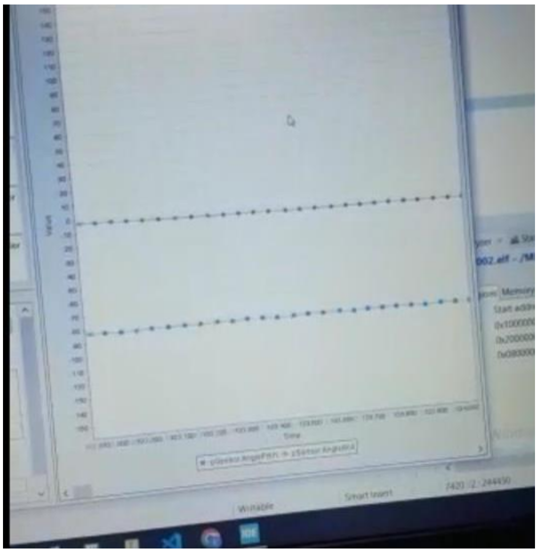

Figure 24

4-) Lastly, we have tuned the PID algorithm via giving different values to PID constant and observe the stability of the drone by two ropes. We have reached the suitable constant at the end, so it was ready to fly without rope.

In figure 25 we have tested stability of take-off, and in figure 26 stability of pitch movement is tested.

Figure 22

Figure 23

7. CONCLUSION

With the popularization of the use of Unmanned Aerial Vehicles, the budget and time allocated to this area by states and companies for various applications that we mentioned before have also increased. Unmanned aerial vehicles.
have become the focus of attention of people, especially engineering students, as it facilitates people's lives from the field of health to hobby activities.

In line with all these developments, it is important for the future of our country that our students have knowledge and experience in this field. In future studies, we want to set a good example that will enable students to have information about important engineering subjects. The main ones of these topics can be listed as PID control algorithms, embedded programming with STM32 MCU using C and C++, understanding communication protocols, working with brushless motors and necessary sensors, and being familiar with the PWM signal. Researching and doing all these is very beneficial for us to increase our knowledge and experience before graduating. By researching and reinforcing the knowledge we have learned in the previous lessons, we could apply it.

As a result of our studies we have completed the %95 of our project the only missing part is flying without rope and let it go up to higher altitudes .Our development board was burned while we were preparing it for fly .Then I borrowed a board from Mr. Cem ÜNSALAN for a week but at this time while we were soldering components to board ,again the board was damaged .Therefore we bought a new board and gave it to Mr. Cem. Due to economic issues we had to stop the project for a while until we get a job. We are going to continue this project by adding autonomous drive feature to it.

REFERENCES

[1] J. M. Brokking, "Project YMFC-AL - The Arduino auto-level quadcopter," *Brokking.net*. [Online]. Available: http://www.brokking.net/ymfc-al_main.html. [Accessed: 23-Jan-2023].

[2] "Arduino I2C Haberleşme Nedir?," *Elektrikport.com*. [Online]. Available: https://www.elektrikport.com/universite/arduino-i2c-haberlesme-nedir/23313. [Accessed: 23-Jan-2023].

[3] "Pulse Width Modulation," in *Analog Interfacing to Embedded Microprocessor Systems*, Elsevier, 2004, pp. 287–297.

[4] "Enescicek.com," *Enescicek.com*. [Online]. Available: https://www.enescicek.com/page/2/. [Accessed: 23-Jan-2023].

[5] C. Taşdemir, *Ctasdemir - overview*. .

[6] D. Yapımı, "Verimlilik Nedir? Drone için Motor Seçimi - 1," 16-Oct-2017. [Online]. Available: https://www.youtube.com/watch?v=zuXdCK0hJCE. [Accessed: 23-Jan-2023].

[7] D. Yapımı, "KV Nedir? Drone için Motor Seçimi - 2," 26-Oct-2017. [Online]. Available: https://www.youtube.com/watch?v=_VPCOE3iTcM. [Accessed: 23-Jan-2023].

[8] Rota, "RC uçak Elektroniği anlatımı," 02-Oct-2017. [Online]. Available: https://www.youtube.com/watch?v=aThEu4dhi6M. [Accessed: 23-Jan-2023].

[9] J. M. Brokking, "Project YMFC-32 - The STM32 quadcopter," *Brokking.net*. [Online]. Available: http://www.brokking.net/ymfc-32_main.html. [Accessed: 23-Jan-2023].

[10] B. Elektronik, "Arduino drone programlama." [Online]. Available: https://www.youtube.com/playlist?list=PLtQlkJsohc9c_IXv-PY0WzH45f750nHhu. [Accessed: 23-Jan-2023].

[11] F. Lin, H. Duan, and X. Qu, "PID control strategy for UAV flight control system based on improved genetic algorithm optimization," in *The 26th Chinese Control and Decision Conference (2014 CCDC)*, 2014, pp. 92–97.

APPENDICES

Appendix A

C Code For Reading Command from Receiver in the Following Link:

https://github.com/alerntrk/BitirmeProjesi/tree/main/Kumanda_002

C Code For Getting Pitch and Roll Values from MPU6050 in the Following Link:

https://github.com/alerntrk/BitirmeProjesi/tree/main/MPU6050_002

C Code For Driving Brushless Motor in the Following Link:

https://github.com/alerntrk/BitirmeProjesi/tree/main/Motor_001