Fiducial Pattern Tracking Drone

ECE 445 Spring 2021

Design Review

Team #59
Angelos Guan, Muhammad Umer Belagam, Alexander Amiel

TA: Chaitanya Sindagi

Table of Contents

1 Introduction
1.1 Problem and Solution Overview 4
1.2 Visual Aid 5
1.3 High Level Requirements 5
1.3.1 5
1.3.2 5
1.3.3 5

2 Design 6
2.1 Block Diagrams 6
2.1.1 Overall Block Diagram 6
2.2.2 Fiducial and Camera Module Block Diagram 7
2.2 Schematic 8
2.3 Subsystem 9
2.3.1 Power Supply 9
2.3.1.1 Li-ion battery 9
2.3.1.2: Li-ion discharging and voltage control circuit 10
2.3.2 Raspberry Pi/Camera Unit 10
2.3.2.1 Raspberry Pi Camera Module 10
2.3.2.2 Raspberry Pi Zero 10
2.3.3 Fiducial Pattern 12
2.3.4 Flight Controller/Control Unit 12
2.3.4.1 Microcontroller 12
2.3.4.2 Inertial Measurement Unit (IMU) 13
2.3.4.3 Pressure Sensor 14
2.3.4.4 Ultrasonic Sensor 17
2.3.5 Motor Unit 18
2.3.5.1 ESC 18
2.3.5.2 Motors 18
2.3.6 User Interface 18
2.3.6.1 LEDs 18
2.3.6.2 Buzzer 19
2.4 Tolerance Analysis 19

3 Ethics and Safety 20

4 Cost 20
4.1 Cost Analysis 20

4.1.1 Labor 20
4.1.2 Parts 21
4.1.3 Sum of Total Cost Analysis 22
4.2 Schedule 22

References **23**

1. Introduction

1.1 Problem and Solution Overview

The goal of our project is to develop an equipment-free hand-operated drone guided by a fiducial pattern. A fiducial pattern is like a QR code pattern except that it is not as complex, yet it still has many sequences that can be generated[1]. The drone is able to detect and track the pattern, maintaining a set distance from it at all times.

Our project seeks to provide an alternative solution to the problem of controlling the drone without a physical controller. Currently, the most popular way of doing so is using mobile phones to keep track of location via bluetooth and monitoring the drone's perspective vision via camera. Such an approach is limited by the range of electromagnetic radiation and susceptible to interface noise. Events such as the phone dying and getting a call or notification can cause the drone piloting app to exit the camera mode and trigger unwanted behaviors for the drone. Another problem with this approach is that downloading an app is required in order to control the drone, and the flying stability and controllability may be dependent on the smartphone running the app. Furthermore, app-controlled drones are usually expensive since they require a bluetooth module and high-quality camera.

Other alternative ways to pilot a drone without physical controllers are through the use of hand gestures (DJI's spark and Samsung)[2] [9] and through the use of a muscular-control system (MIT's artificial intelligence lab)[3]. These methods also encounter many limitations. The problem with hand gestured controlled drones is that it may detect someone else's hand which may result in undesired behaviors. The limitations on vision-based gesture recognition such as poor performance due to optical noises also apply to this approach as discussed in the paper[9]. In MIT's muscular-based control approach, problems arise when the person piloting the drone has a muscular breakdown or accidentally squeezes too hard or too soft.

Our solution provides a new way to pilot the drone by detecting a given fiducial pattern and maintaining a distance from it. The user would be holding the fiducial pattern and guide the drone by moving the fiducial pattern to the desired position. If no pattern is detected, the drone would signal the user through a buzzer and LED. Our method provides a controller-free approach to reliably pilot a drone while solving the problem of confusing other people's hands as user input.

We will be powering the drone with a Li-ion rechargeable battery attached to a voltage control unit. We will use a raspberry pi connected to a camera module to recognize the pattern, and send goal position extracted from the camera data from raspberry pi to the stm32 microcontroller so that the drone can maintain a certain distance to the fiducial pattern. The stm32 microcontroller will be responsible for processing sensor data and running the flight control loop.

1.2 Visual Aid

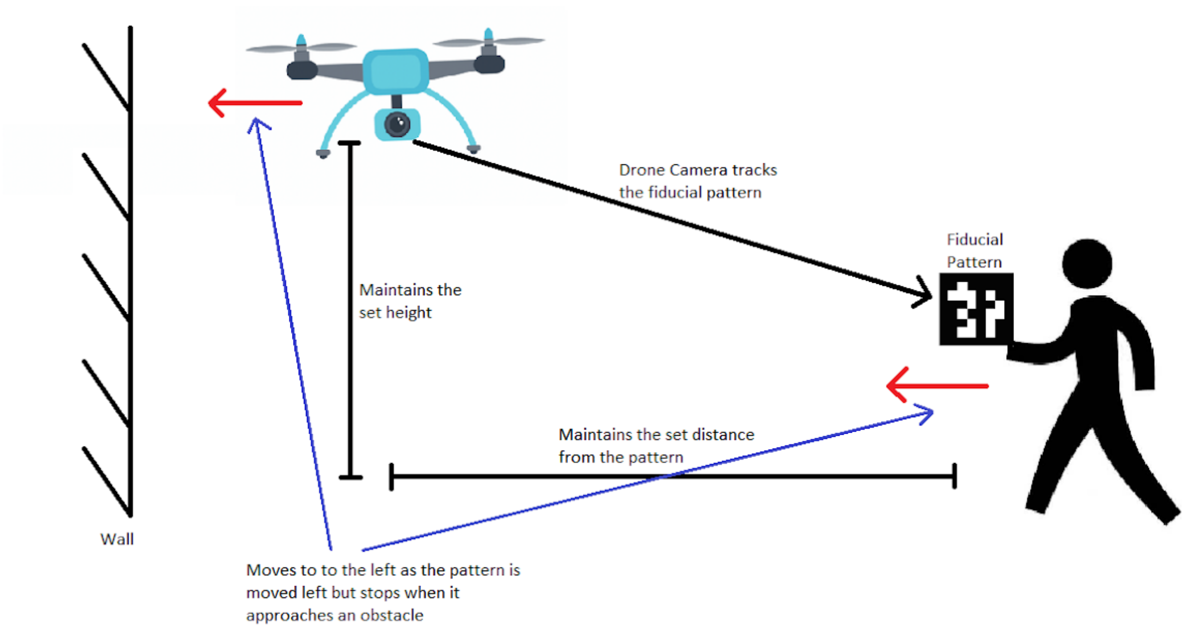


Figure 1: Visual representation of the drone recognizing the fiducial pattern and staying a fixed distance from the pattern

1.3 High-level requirements list

1.3.1: Our project must be able to use a camera mounted on the drone to detect and track a fixed sized fiducial pattern from 1-2m distance and estimate the real-world position coordinate of the detected pattern within 10% mean squared error.

1.3.2: The drone must be able to fly with an upright orientation and maintain a 1.5m distance to the detected fiducial pattern within 15% steady-state error and less than 5s response latency. If a fiducial pattern is not detected, the drone should maintain its current position and ring the buzzer and light up the LED to notify the user.

1.3.3: The drone must be able to halt in place if there is any obstacle detected by the ultrasonic sensors in 0.2m range along the horizontal or vertical axes.

2. Design

2.1.1 Overall Block Diagram

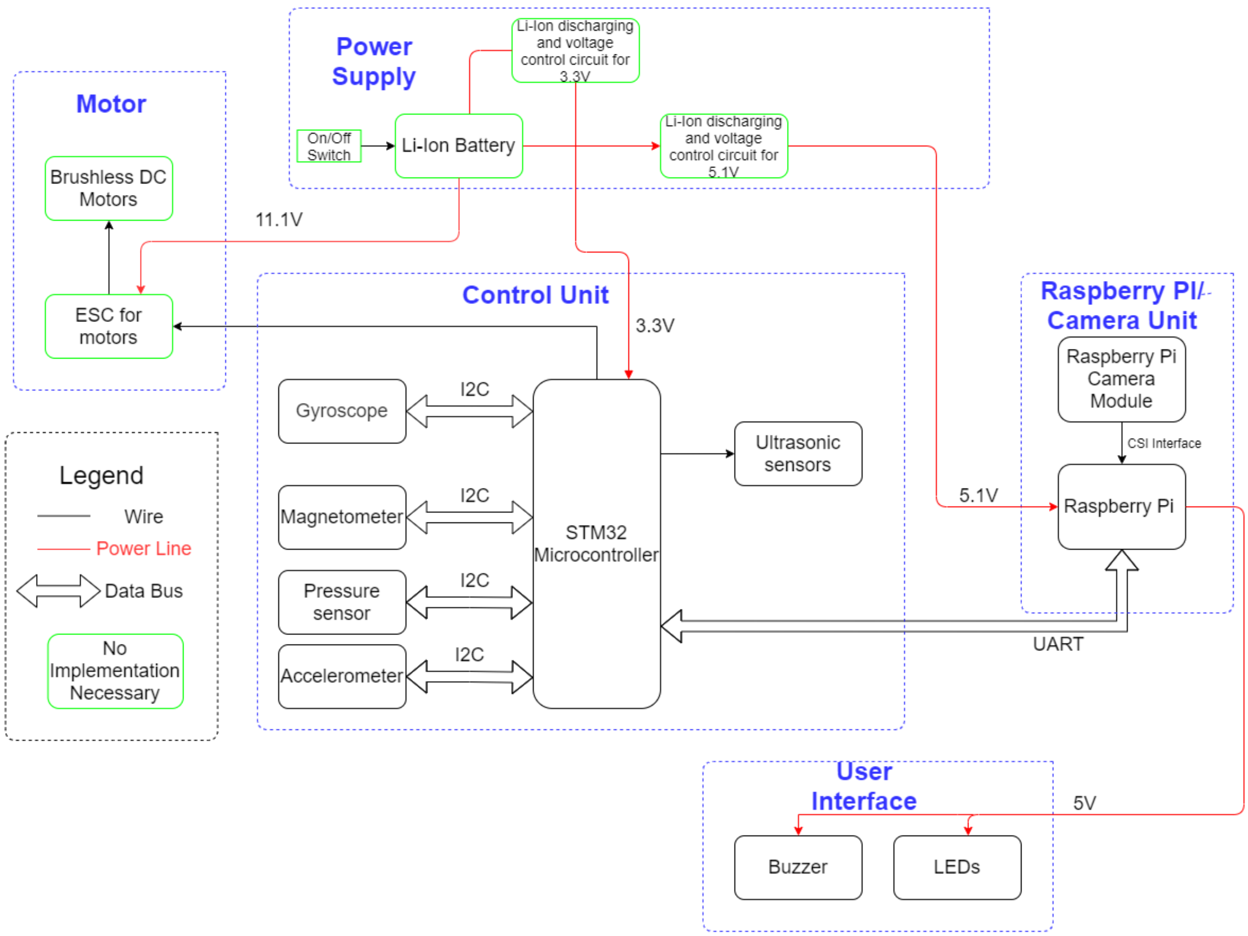


Figure 2: Block diagram for the interface between each subsystem

2.1.2 Fiducial and Camera Module Block Diagram

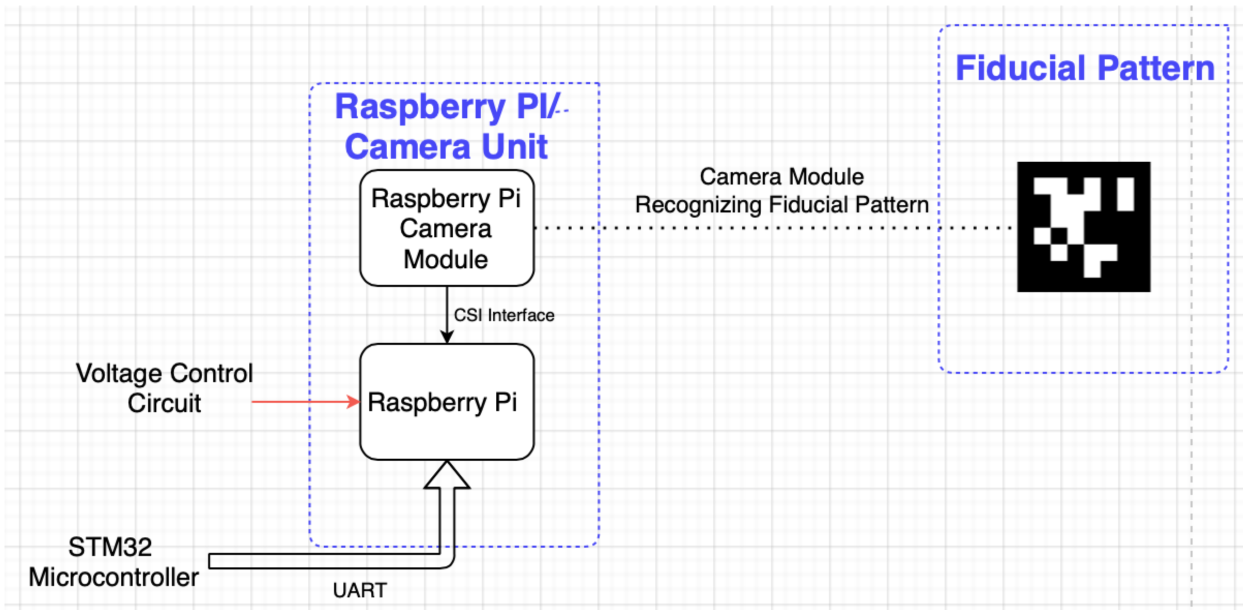
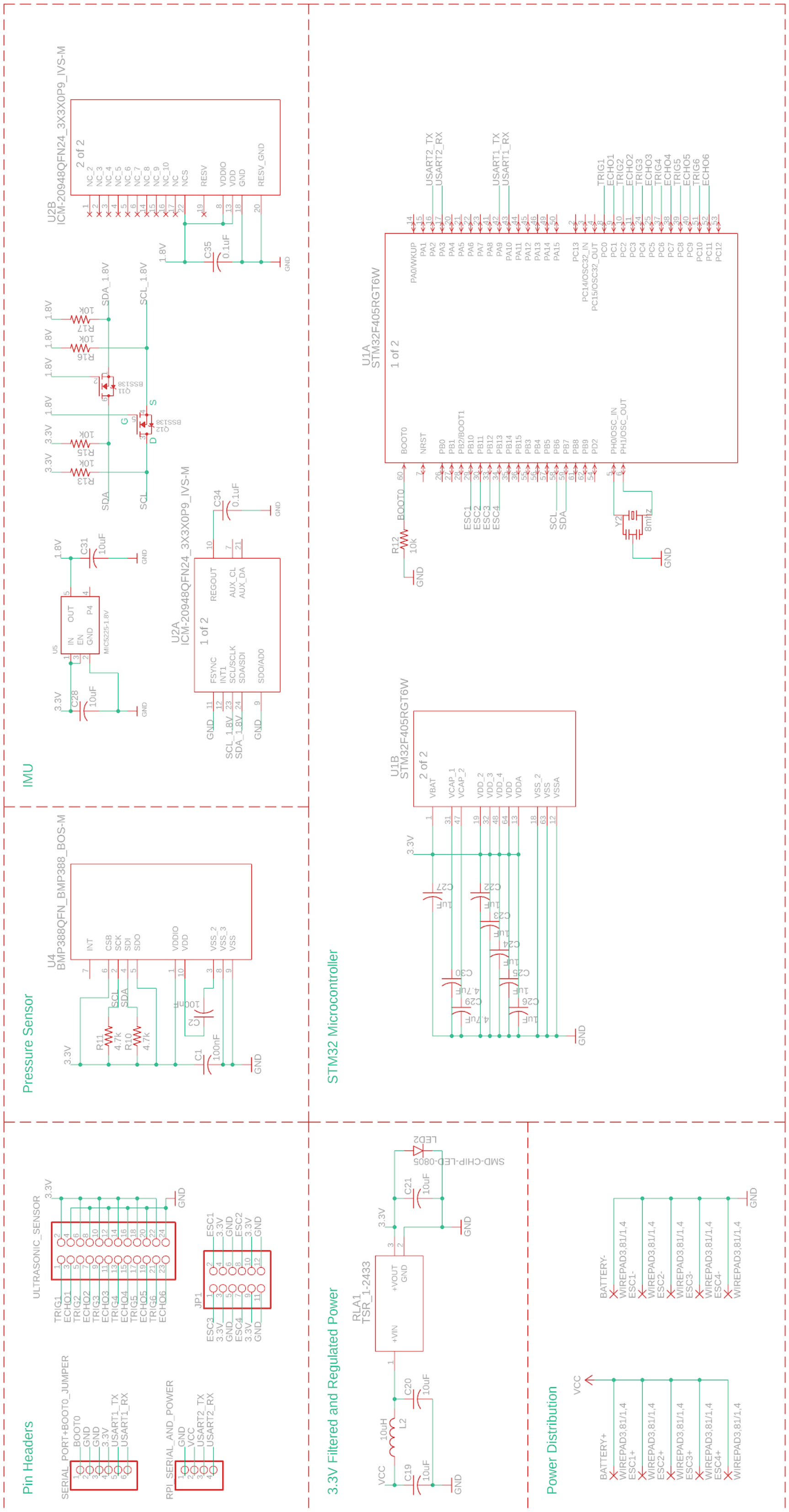


Figure 3: Block diagram for raspberry pi camera module recognizing the fiducial pattern

2.2 Schematic



2.3 Subsystem

2.3.1 Power Supply

The power supply is responsible for keeping the drone operational at all times. The power supply voltage control is responsible for a regulated 3.3V output to the control unit and 5.1V to the raspberry pi. It has an over discharge protection for the li-ion batteries and it powers the ESC with the protected output. We won't be building the power supply or motor from scratch, rather we are buying an already built power supply and ESC motor.

Requirements	**Verification**
Supply 3.3V ± 5% to the microcontroller and ESC motor unit	Use an oscilloscope to measure the output of the voltage control circuit to ensure it is supplying within 5% of the 3.3V
Supply 5.1V ± 5% to the raspberry pi to ensure it is functional	Use an oscilloscope to measure the output of the voltage control circuit to ensure it is supplying within 5% of the 5.1V

2.3.1.1 Li-ion Battery

A 3S1P Li-ion battery pack with a nominal voltage of 11.1V and minimum and maximum voltage of 9.6V and 12.6V respectively. This is used to directly power the ESC. It also goes through two voltage control circuits: 1 step down regulator to 3.3V to power the microcontroller, and one step down regulator to 5.1V to power the Raspberry Pi zero. We are buying this battery pack.

Requirement	**Verification**
Be able to supply a continuous 20A with a peak of 35A.	Use an electronic load to ensure that 20A is able to be drawn from the batteries without any of the cells voltage dropping below 3.2V
Ensure that the batteries are within the optimal temperature range of 15°C to 30°C to not risk the potential hazard of a fire/explosion[4].	Use a thermocouple temperature sensor taped to the battery to ensure it is within the optimal temperature range under

2.3.1.2 Li-ion discharging and voltage control circuit

Overdischarge and short circuit protection circuit for the li-ion batteries to prevent unsafe conditions. Two buck convertors with 3.3V±5% and 5V±5% output to step down the battery voltage for powering the control unit and Raspberry Pi. The 3.3V buck converter should be able to output 1A and the 5V convertor should be able to output 3A at 80%+ efficiency.

Requirements	**Verification**
Voltage control circuit must step down the 12.6V from the 3S Li-ion batteries at max capacity to supply 3.3V ± 5% to the microcontroller and ESC motors and 5.1V ± 5% to the Raspberry Pi	Use the oscilloscope to ensure the voltage control circuit is supplying the correct voltages to each respective subsystem
Must be able to output at 1A for 3.3V and 3A for 5V at 80% efficiency.	Use an electronic load to draw 1A from the 3.3V and 3A from 5V output of the voltage regulators and measure the current draw and voltage on the inputs to calculate power and make sure efficiency is met. $Efficiency = \frac{Input\ Current * Input\ Voltage}{0.5A * (3.3V\ or\ 5V)}$

2.3.2 Raspberry Pi/Camera Unit

The raspberry pi/camera unit recognizes the fiducial pattern and prepares data to be sent to the stm32 microcontroller through UART RX/TX ports. This is enabled through the raspberry pi's openCV software library to create our own library for detecting the fiducial pattern.

2.3.2.1 Raspberry Pi Camera Module

The camera module is connected to the raspberry pi through the CSI interface. It is responsible for capturing the fiducial pattern the user shows to the camera. The camera module is light, 3 grams, and can support resolution up to 1080p30, 720p60 and 640x480p90.

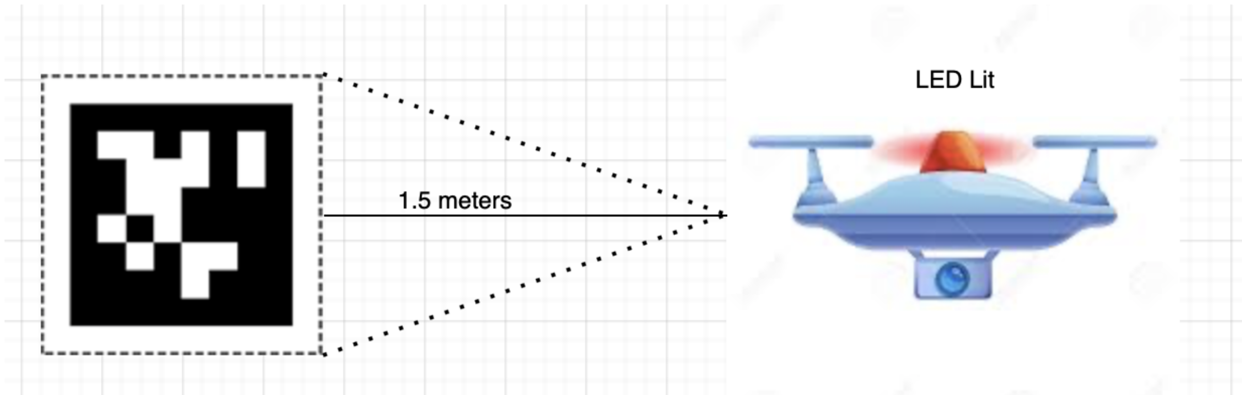
2.3.2.2 Raspberry Pi Zero

The raspberry pi zero is responsible for taking the fiducial pattern, recognized from the camera module connected through the CSI interface, and sending data through the UART to the stm32 microcontroller. Each fiducial pattern encodes a different function that the raspberry pi has to recognize. The stm32 microcontroller is signaled to make the drone do a specific action.

Requirements	Verification
Must have the CSI interface to connect with the camera module	Confirm that the raspberry pi zero has the CSI port on the board
Must be able to communicate with the camera module to recognize the fiducial pattern from 0-1.5 ± 0.3 meters away	Run AprilTag in realtime to recognize the fiducial pattern
Must be able to interface with the STM32 microcontroller through the UART and signal the microcontroller to have the drone follow the fiducial pattern from 1.5 ± meters away within 1s latency	Send signal from raspberry pi to stm32 to light up an led and measure the response time.
Must be able to light up the LEDs from the raspberry pi's power line of 5V.	Use an oscilloscope to ensure 5V is being supplied to the LED

Figure 4: Raspberry Pi zero connected to the camera module

We will be using the fiducial pattern detection package AprilTag because it is sufficiently invariant to rotation and scale. This software package can be downloaded, run in python, and implemented in openCV. We specify a source image through an AprilTag Family, a specific fiducial pattern, and the camera can detect a set of corners from the fiducial pattern to recognize the image. Once the image is detected, a red LED will be displayed.



Visual Aid: Red LED Lights up when the drone detects the fiducial pattern

2.3.3 Fiducial Pattern

A 100x100 mm pattern of black and white squares on a piece of paper to uniquely identify as a marker. The fiducial pattern will be recognized by the raspberry pi camera module using the technique we described above.

Requirement	**Verification**
Must be within 1.5 ± 0.3 meters of the camera module, and oriented 0°±15° in all dimensions where 0° would be when it is perfectly parallel to the plane of the camera lens and has no rotation in that plane.	Through the use of LEDs, they will verify the user when the fiducial pattern is recognized by flashing 1 red LED.

2.3.4 Flight Controller/Control Unit

2.3.4.1 Microcontroller

We will be using the STM32F405RGT6W microcontroller which is based on the Arm Cortex M4 core. This chip would be responsible for processing the data from the various sensors and use a PID algorithm to send the appropriate commands to the ESC for the motors. It must have I²C peripherals in order to read the sensor data. It must also take commands from the raspberry pi through the UART interface when the raspberry pi signals the microcontroller that the fiducial pattern has been read.

Requirement	Verification
Must have FPU (floating point unit) to make fast calculations on the sensor data to go through the PID control loop and send commands to the ESC at 16kHz[7]. So the clock speed must be at least 16k * the number of machine instructions for the PID loop.	Toggle a pin on the microcontroller after each PID loop and measure the time it takes to switch states using an oscilloscope and verify that it's 1/16kHz = 62.5us

2.3.4.2 Inertial Measurement Unit (IMU)

IMU is a combination of 3-axis gyroscope, accelerometer, and magnetometer that measures the angular velocity and the acceleration experienced by the drone during flight through the I2C interface with the microcontroller. IMU is used to figure out the drone's orientation with respect to the gravity vector and the change in velocity when it is moving for the purpose of navigation. We will be using ICM-20948 for this module.

We will need to detect sufficient ranges of ± 2g for the accelerometer. The available ranges for the gyroscope is between 250-2000 degrees/sec. The output data will be digitized into 16 bits and is fed into the microcontroller via the I2C communication port. A low pass filter can be programmed with a range from 5.7 to 197 Hz. The VDDIO operating range is between 1.71V to 1.95V. Pin connection is shown in table 1.

PIN NUMBER	PIN NAME	PIN DESCRIPTION
7	AUX_CL	I^2C Master serial clock, for connecting to external sensors
8	VDDIO	Digital I/O supply voltage
9	AD0 / SDO	I^2C Slave Address LSB (AD0); SPI serial data output (SDO)
10	REGOUT	Regulator filter capacitor connection
11	FSYNC	Frame synchronization digital input. Connect to GND if unused
12	INT1	Interrupt 1
13	VDD	Power supply voltage
18	GND	Power supply ground
19	RESV	Reserved. Do not connect.
20	RESV	Reserved. Connect to GND.
21	AUX_DA	I^2C master serial data, for connecting to external sensors
22	nCS	Chip select (SPI mode only)
23	SCL / SCLK	I^2C serial clock (SCL); SPI serial clock (SCLK)
24	SDA / SDI	I^2C serial data (SDA); SPI serial data input (SDI)
1 – 6, 14 - 17	NC	Do not connect

Table 1: Pin Output Diagram

In the figure below, the IMU acts as an I2C master to the external sensor. The bypass multiplexer connects the system processor I2C bus pins 23 and 24 directly to the auxiliary sensor I2C pins 7 and 21 to manage configurations of the external sensor. The configuration of this is shown below:

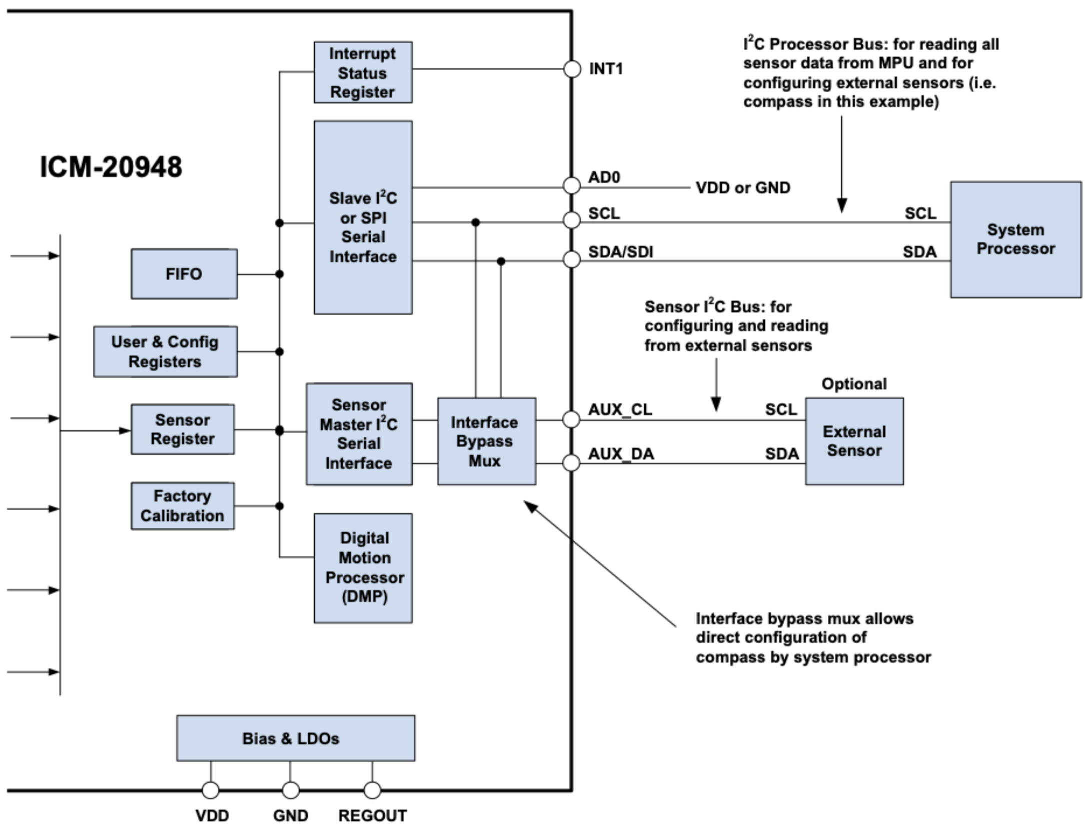


Diagram 1: I2C Serial Communication Interface

Requirement	Verification
Must be able to detect rotational movement at max 1000 degrees per second with an accuracy of 15%.	Ensure that the measurement being collected by the software from the sensor data acquired

2.3.4.3 Pressure sensor

Pressure sensor is responsible for measuring the atmospheric pressure the drone experiences at its current position. This will be used to determine what altitude the drone is at so it can maintain its distance from the ground. We will be using BMP388 which offers accurate measurement of barometric pressure that provides altitude data for flight stability and landing. It sends data through the I2C communication port to the microcontroller.

Important Specifications:

- Power consumption of 3.4uA at 1Hz.
- VDD supply voltage of 1.65V to 3.6V.
- Relative accuracy Pressure: 900-1100 hPa ± 0.08^2 hPa (equivalent to 0.66 meters)
- Start up time to first communication: 2 ms when voltage is greater than 1.58V
- Sampling rate: 200 Hz

A description of the measurement cycle is shown below. Oversampling isn't required, and the optional IIR filters are used to remove short term fluctuations in pressure (slamming a door):

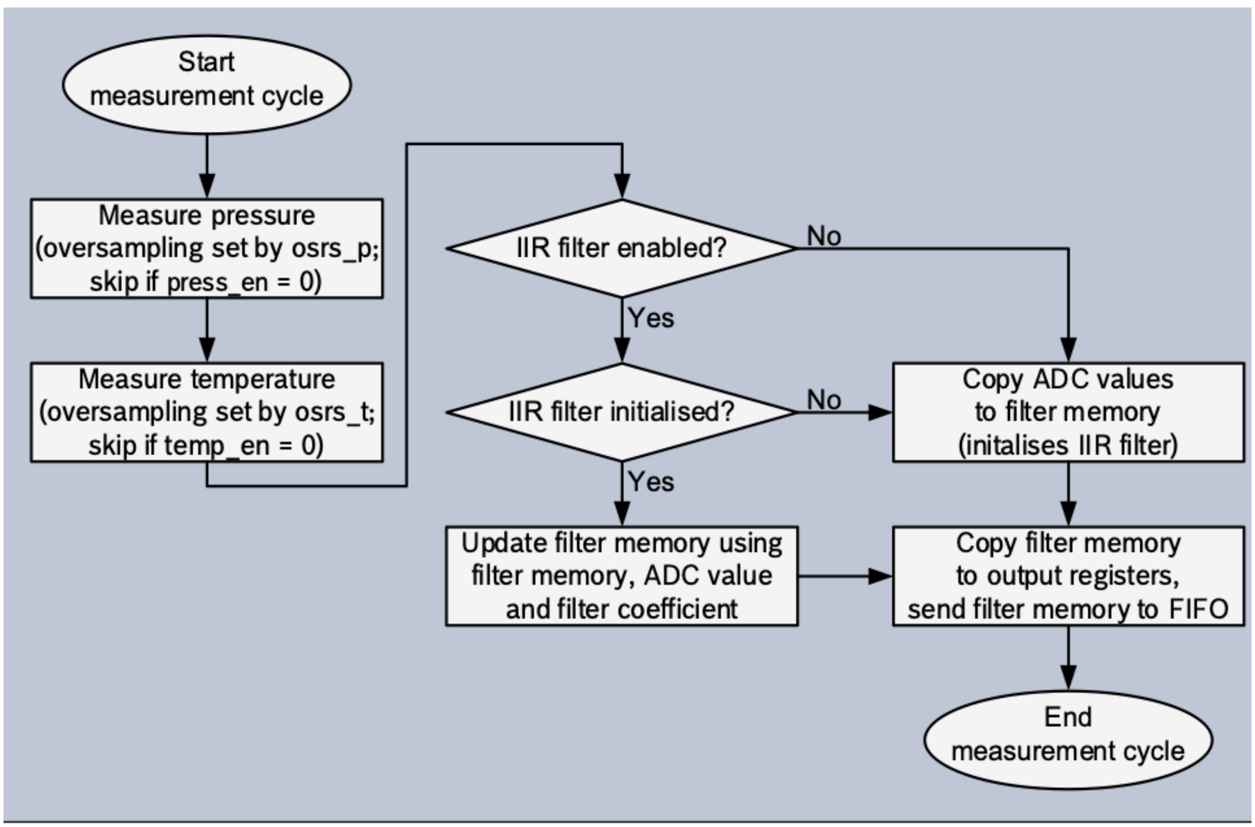


Diagram 2: Measurement Cycle

The internal IIR filter can effectively reduce the bandwidth of the output signals. The output of the next measurement step is shown below:

$$data_{filtered} = \frac{data_{filtered\_old} * filter\_coefficient + data_{ADC}}{filter\_coefficient + 1}$$

Equation 1: equation for next measurement step

The data filtered_old is the data coming from the previous acquisition, and the data_ADC is the data coming from the ADC before IIR filtering. The next value will pass through the filter and be the initial memory value for the filter. IIR filter is reset if the temperature measurement is disabled or there is a transition from sleep to normal mode. After enabling pressure or temperature measurement, the filtering will start, thus the next incoming value will pass unfiltered and be the initial value of the IIR filter. The step response (e.g. response to in sudden change in height) of different filter settings is shown below:

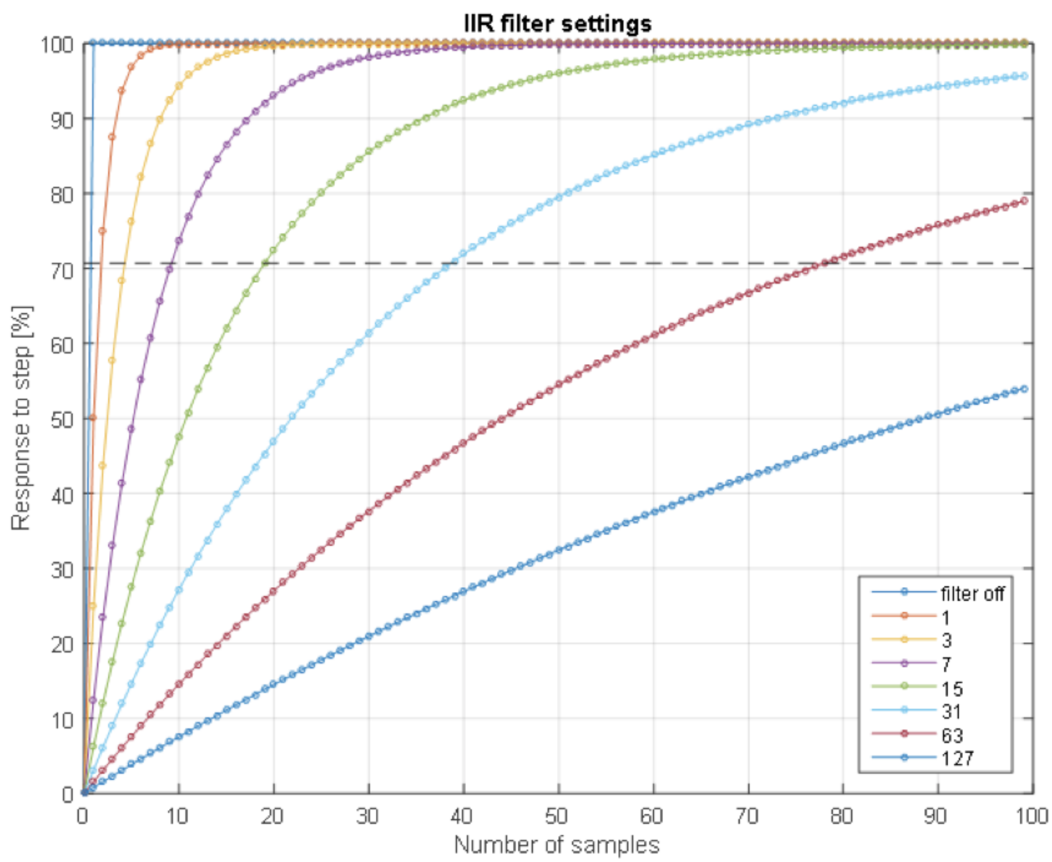


Figure 5: step response % vs Number of Samples due to different filtering settings

Measurement Times are critical so that we can communicate to the microcontroller. Below is a table referencing measurement time vs measurement rate at different oversampling settings:

Oversampling setting	Pressure oversampling	Temperature oversampling	Measurement time[ms]		Measurement rate [Hz]	
			Typ	Max	Typ	Min
Ultra low power	×1	×1	4.82	5.70	207.08	175.39
Low power	×2	×1	6.84	7.96	146.00	125.56
Standard resolution	×4	×1	10.88	12.48	91.83	80.07
High resolution	×8	×1	18.69	21.53	52.71	46.42
Ultra high resolution	×16	×2	37.14	41.89	26.91	23.86
highest resolution	×32	×2	69.46	78.09	14.39	12.80

Table 2: Measurement Time vs Measurement Rate

Requirement	**Verification**
Be able to detect the ground within 0.6m ± 0.2m	Hold the sensor above a piece of wood and read the outputted data.

2.3.4.4 Ultrasonic Sensor

Ultrasonic sensors are used for detecting any obstacle in 6 horizontal and vertical directions. These sensors will be responsible for detecting if the drone is going to hit any obstacle in its way. We will be using 6 GY-US42 sensors for this purpose.

It has a working voltage of 3-5V, and it can directly connect to the microcontroller. It has a 15Hz frequency range and a resolution of 1 cm. It has a measuring range of 20cm to 720cm

Connection configurations Pinout to the microcontroller:

- “CR” pin is either the SCL (I2C mode), Serial RX, or Trigger input
- “DT” pin is either the SDA (I2C mode), Serail TX, or Echo output

Requirement	**Verification**
Must be able to detect obstacles in 1-4m range.	Gradually move and object with a range slightly wider than its detection range and check the reading. Do this every 0.5 meters from 0 to 5 meters.

2.3.5 Motor Unit

2.3.5.1 ESC

Used to control the speed of the motors to maneuver the drone. It is powered directly from the batteries, supplied 9.6V to 12.6V. It can draw 20A continuous and peak 35A current. We are buying this part. We will use an ESC with the BLHeli firmware and it will be an “Opto” type ESC and not the “BEC” (battery elimination circuit) type because we have a separate power supply.

Requirement	
Be able to be supplied 11.1V directly from the battery.	Use the oscilloscope to ensure the batteries are supplying the correct voltages to the ESC
Be able to run at continuous 20A and 30A peak current.	Use a bench power supply and ammeter to ensure the desired current is drawn without overheating and test the correct output is produced using an oscilloscope.

2.3.5.2 Motors

We will be using brushless dc motors because of their higher efficiency compared to brushed DC motors as we have a limited battery capacity we need to maximize flight time. We will be purchasing this part and it has to be of the 2212 size to fit in the drone frame we are using.

2.3.6 User Interface

2.3.6.1 LEDs

We will be using three LEDs to indicate the status of the drone including fiducial pattern not detected, battery low, when the user is guiding the drone into or too close to any obstacle. This is powered by the raspberry pi zero 5V pin.

Requirement	**Verification**
Know if the fiducial pattern is recognized: The microcontroller and raspberry pi must be integrated to enable status LED	Red LED is lit up when the fiducial pattern is recognized by the camera module
If the battery is low: the microcontroller and power supply must be integrated to enable	Green LED is lit up when the battery is low

battery low LED	
If there is an obstacle: The microcontroller and raspberry pi must be integrated to enable status LED	Yellow LED is lit up when there is an obstacle in the way of the goal position that the user indicates

2.3.6.2 Buzzer

We will be using a buzzer to warn the user when the fiducial pattern is not detected. Drive a magnetic buzzer at a voltage of 5V from the raspberry pi zero to the buzzer. The buzzer's sound output is 85dB.

Requirement	**Verification**
Know if the pattern isn't recognized: the microcontroller and raspberry pi must be integrated to enable the buzzer going off	Hear the buzzer going off when the pattern isn't recognized, and it stops when it is recognized

2.4 Tolerance Analysis

The critical part of the project is the flight controller and pressure sensor. Pressure sensor provides pressure data that is used to calculate the altitude of the drone. The output of the pressure sensor is essential for controlling the height of the drone. As shown in Equation 2, PID Flight controller would send corresponding output signal to the ESC according to the desired positions calculated from fiducial pattern detection algorithm on raspberry pi and the estimated pose and position from accelerator, gyroscope, magnetometer and compass.

$$u(t) = k_p e(t) + k_i \int e(t) + k_d \frac{d}{dt} e(t)$$

$$e(t) = x_d(t) - x_o(t)$$

Equation 2: equation for PID controller

$e(t)$ in the equation 2 is the difference between the desired output provided by raspberry pi via fiducial pattern detection algorithm and the actual output estimated by the sensor. The fiducial pattern recognition algorithm we are using is AprilTag and the error in position has a linear correlation to the distance from the pattern to the camera. In our use case we wish to keep the drone in 1.5m distance to the fiducial pattern. The estimation accuracy for the position from AprilTag package in this case is about 0.03m.[10] and the error of actual output is dependent on the pressure sensor reading. The sensor we are using is BMP388, it has an output accuracy about ±8Pa, which is equivalent to about ±0.5m difference in altitude. The total error would be within 0.5m range.

3. Ethics and Safety

Since the goal of our project is to implement a new approach for the user to control the drone, we assume that the action of the drone under ethical evaluation is based on the action of the person controlling the drone. The most relevant codes of ethics for this project are IEEE Code of Ethics #1 and #9[5], which stresses the importance of ensuring the safety of the public and their properties. Both the drone and its operator must follow the IEEE Code of Ethics and develop safety protocols in case of failures such as communication failure, drone is out of battery or controller malfunction. For the purpose of this project, we assume that the usage of the drone is recreational. Thus, under Federal Aviation Administration's (FAA) policies on Recreational Flyers[6], our project can only be flown at or below 400 feet above the ground when in uncontrolled airspace. In addition, under university policy Fo-05, in order to test our project on campus, we must obtain approval from the Division of Public Safety by submitting a request to dpscomments@illinois.edu[7]. If the aerial testing happens inside campus property, we must obtain approval from Code Compliance & Fire Safety as well as from faculty.

Usage of Li-Ion batteries also raises several safety concerns[8]. First, they need to be stored at room temperature and not under direct sunlight or other heat sources. Second, Li-Ion battery cells should never be stored fully charged. Third, a battery protection circuit is required to control overvoltage, undervoltage and current surges.

Due to the high current applied to the motors, we also need to follow the safety rules regarding lab fire safety and electrical safety.

4. Cost

4.1 Cost Analysis

4.1.1 Labor

For recent graduate students from ECE from the University of Illinois, the typical hourly pay is around $44. Each of us will be working 8 hours a week for 12 weeks. Thus, the total labor cost of the development of the product will be:

3 students * $44/hour * 8 hours/week * 12 weeks * 2.5 (overhead included) = $31,680

4.1.2 Parts

The cost for all the parts we will be buying for the product and testing equipment is shown in the table below:

Description (Part #)	**Vendor**	**Manufacturer**	**Quantity**	**Cost**
Raspberry Pi 0	Adafruit	Raspberry Pi Foundation	1	$5.00
Raspberry Pi Camera Module	Amazon	Raspberry Pi Foundation	1	$24.99
Raspberry Pi Camera Cable	Adafruit	Raspberry Pi	1	$6.00
STM32F405RGT6W Microcontroller	Mouser	STMicroelectronics		$12.44
Accelerometer, Gyroscope, and magnetometer (ICM-20948)	Mouser	TDK	1	$5.91
Pressure Sensor (BMP388)	Mouser	Bosch Sensortec	1	$3.20
Crazepony F450-V2 Drone Frame Kit 4-Axis Airframe 450mm Quadcopter Frame Kit with Landing Skid Gear	Amazon	Crazepony	1	$19.99
2212 920kV Brushless DC motors + SimonK 30A ESC	Amazon	Hobbypower	4	$55.99
Fan blades	Amazon	RAYCorp	8	$10.15
3S Lipo Battery	Amazon	Zeee	1	$33.99
TSR12433 3.3V Buck Converter Breakout - 3.3V Output 1A Max	Adafruit	Adafruit	1	$4.95
UBEC DC/DC Step-Down (Buck) Converter - 5V @ 3A output	Adafruit	Adafruit	1	$9.95
LEDs	Adafruit	Adafruit	3	$0.96
Buzzer (SD1614T5-B5ME)	Mouser	CUI	1	$1.25

Miscellaneous capacitors and resistors	Mouser	Mouser	-	$10

4.1.3 Sum

Labor + Parts = $31,680 + $194.77 = $31874.77

4.2 Schedule

Week	**Task**	**Responsibility**
3/1	1. **Design Document due** 2. Design document check 3. Design review sign up 4. Finalize parts to buy	All
3/8	1. **Design review** 2. Buy all parts 3. Ensure all parts from machine shop are bought if needed	1. All 2. Alexander 3. Umer
3/15	1. Study data sheets of sensors 2. **Simulation assignment due** 3. **Soldering assignment due** 4. **First round PCB orders due and audit needs to be passed**	1. Angelos 2. All 3. All 4. All
3/22	1. **Write code for flight controller**	1. All
3/29	1. Get the fiducial pattern detection working 2. Test all the sensors 3. Prepare final paper	1. Angelos 2. Umer 3. Alexander
4/5	1. **Individual progress reports due** 2. **Get Flight controller working**	2. All
4/12	1. Prepare for mock demo	1. All
4/19	1. **Mock demo**	1. All

4/26	1. Demonstration 2. Mock presentation	1. All
5/3	1. **Final paper due** 2. Final presentation 3. Lab checkout and notebooks due	1. All

All of the work we do together either in person or through zoom, so that is why we write all in most of the responsibility sections.

References

[1] Magid, E., 2018. *ARTag, AprilTag and CALTag Fiducial Marker Systems: Comparison in a Presence of Partial Marker Occlusion and Rotation*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/318871910_ARTag_AprilTag_and_CALTag_Fiducial_Marker_Systems_Comparison_in_a_Presence_of_Partial_Marker_Occlusion_and_Rotation [Accessed 4 March 2021].

[2] J. McCorvey, "DJI's Spark Is The First Drone You Can Control With Hand Gestures", *Fast Company*, 2021. [Online]. Available: https://www.fastcompany.com/40525592/djis-spark-is-the-first-drone-you-can-control-with-hand-gestures [Accessed: 04- Mar- 2021].

[3] D. Etherington, "TechCrunch is now a part of Verizon Media", *Techcrunch.com*, 2021. [Online]. Available: https://techcrunch.com/2020/04/27/mit-muscle-control-system-for-drones-lets-a-pilot-use-gestures-for-accurate-and-specific-navigation/ [Accessed: 04- Mar- 2021].

[4] S. Ma, "Temperature effect and thermal impact in lithium-ion batteries: A review", *Sciencedirect.com*, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1002007118307536 [Accessed: 04- Mar- 2021].

[5] "IEEE Code of Ethics", *Ieee.org*, 2021. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed: 04- Mar- 2021].

[6] "Recreational Flyers & Modeler Community-Based Organizations", *faa.gov*, 2021. [Online]. Available: https://www.faa.gov/uas/recreational_fliers/ [Accessed: 04- Mar- 2021].
[7] "Aerial Activities Over, On, or In Campus Property – Campus Administrative Manual", *Cam.illinois.edu*, 2021. [Online]. Available: https://cam.illinois.edu/policies/fo-05/ [Accessed: 04- Mar- 2021].

[8] "Lithium-Ion & Lithium Polymer Cells and Batteries Safety Precautions", *Ultralifecorporation.com*, 2021. [Online]. Available:

https://www.ultralifecorporation.com/PrivateDocuments/BR_LiIon_Safety_Precautions.pdf [Accessed: 04-Mar- 2021].

[9] K. Natarajan, T. D. Nguyen and M. Mete, "Hand Gesture Controlled Drones: An Open Source Library," *2018 1st International Conference on Data Intelligence and Security (ICDIS)*, South Padre Island, TX, USA, 2018, pp. 168-175, doi: 10.1109/ICDIS.2018.00035.
[10] K. Pentenrieder. Analysis of tracking accuracy for singlecamera square-marker-based tracking.