Creation of Drone: Final Project Portfolio

STEM 382: Industrial Design

4/25/2022

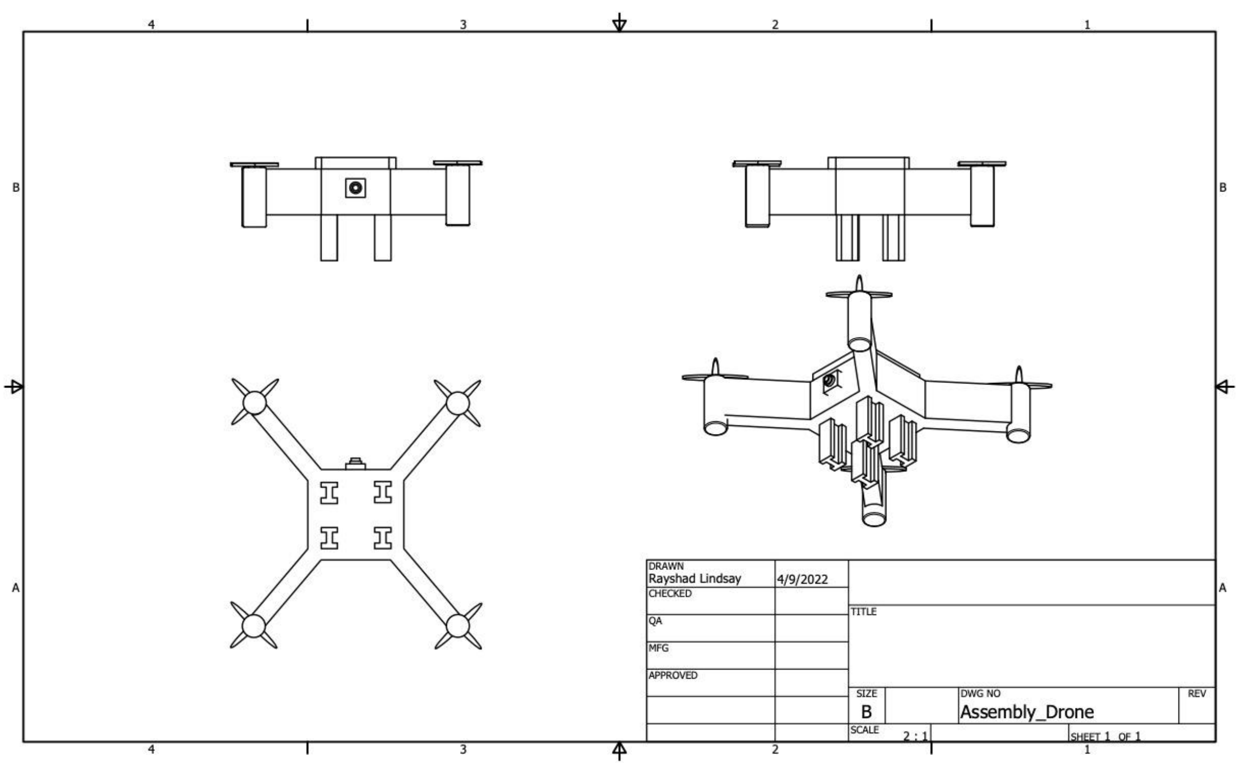
Rayshad Lindsay

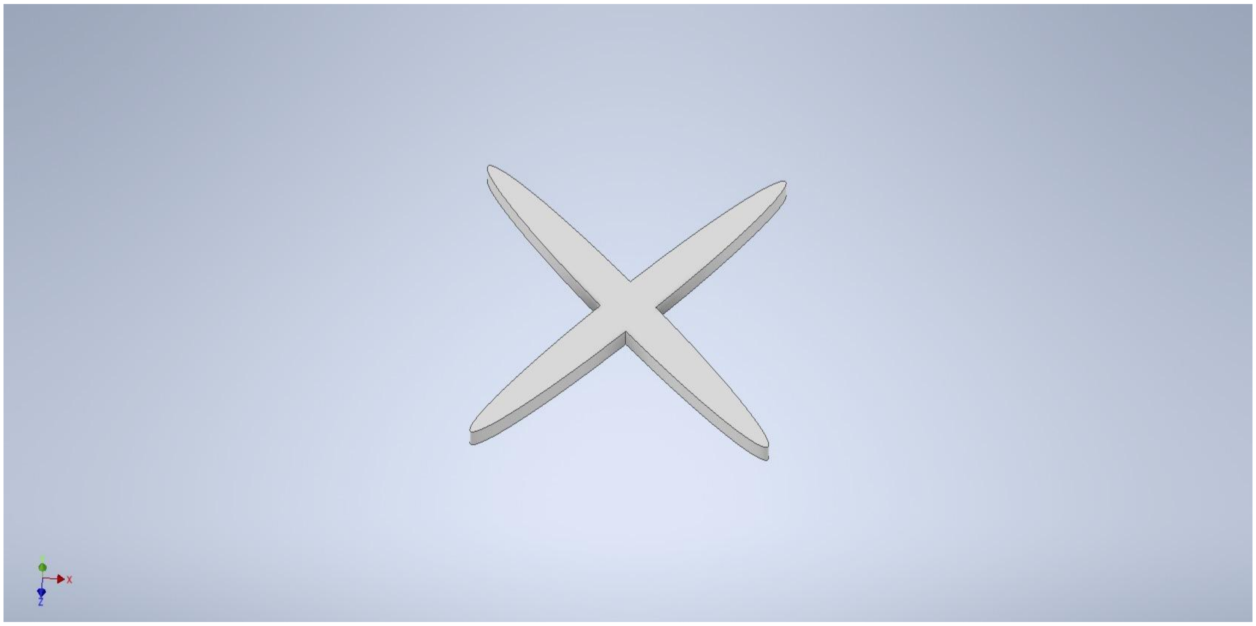
UIN: 01601201

Old Dominion University

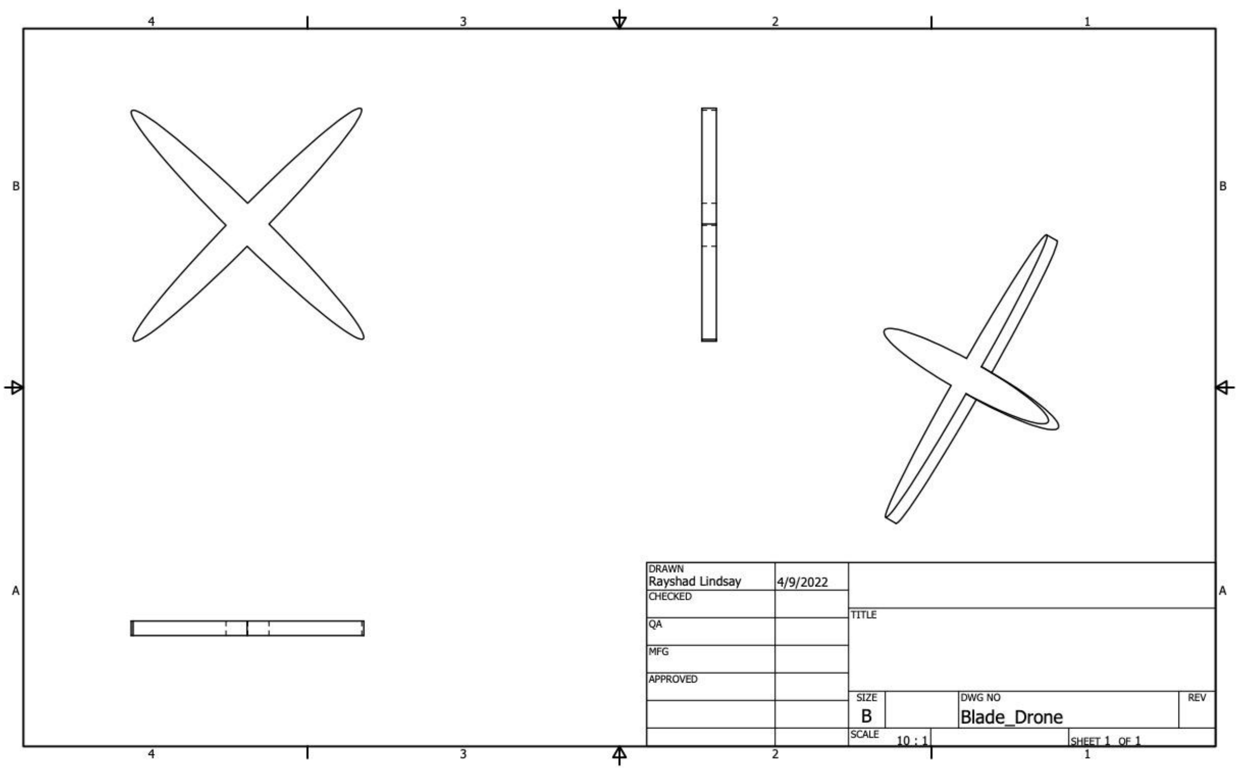
Table of Contents

Lists of Figures

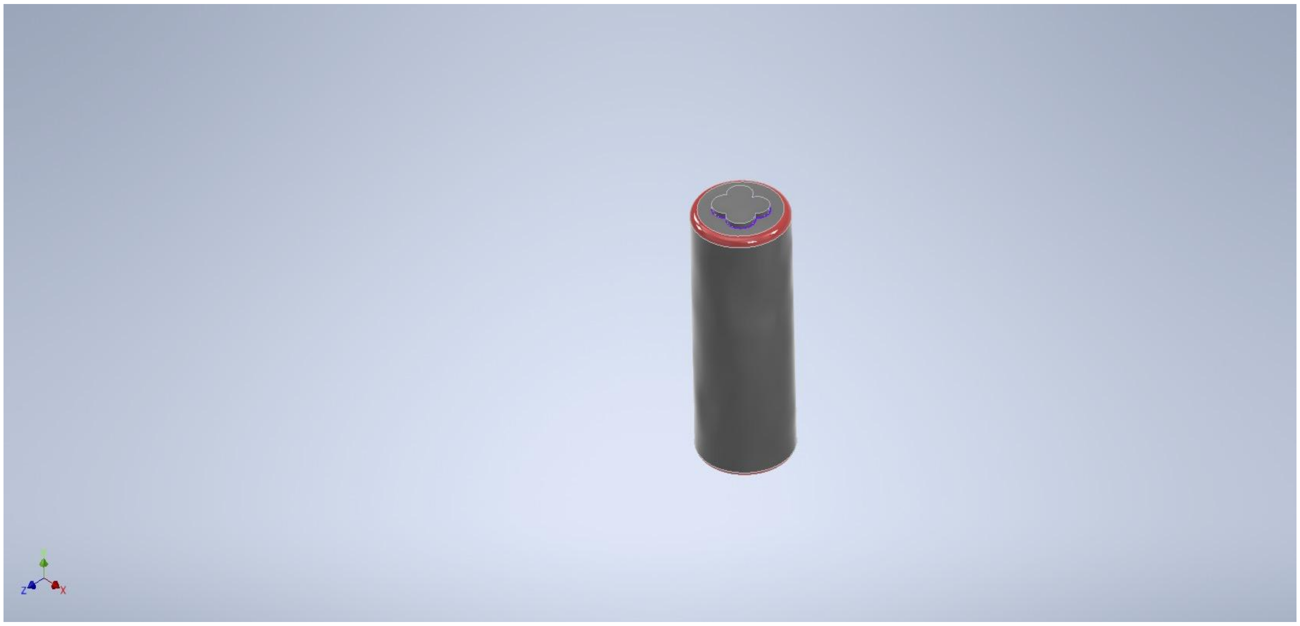




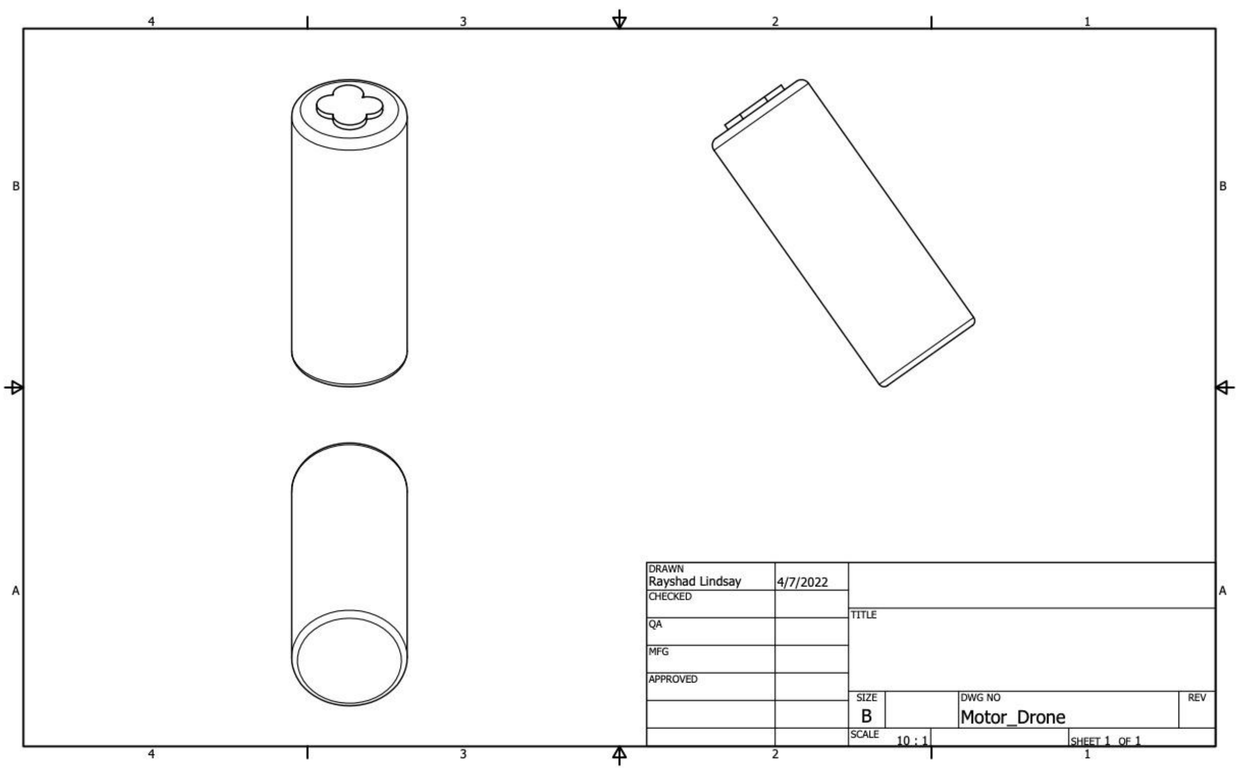
X
Z



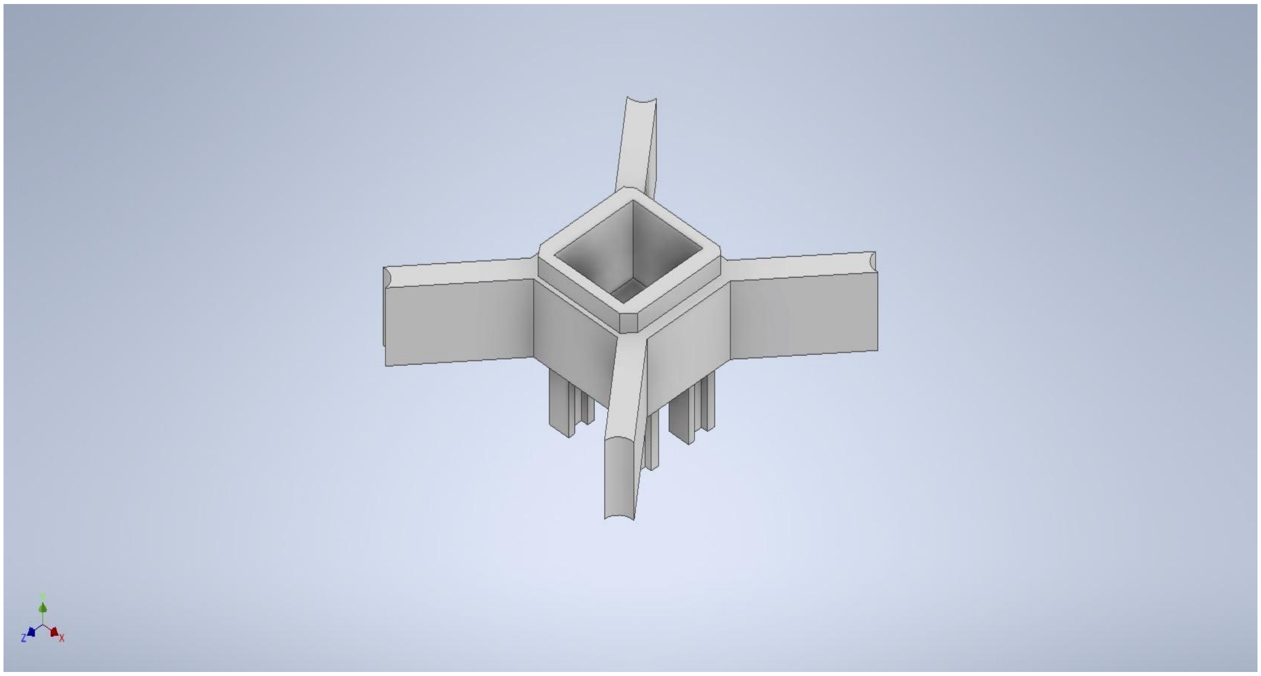
4
3
2
1
B
A
DRAWN
Rayshad Lindsay
4/9/2022
CHECKED
QA
MFG
APPROVED
TITLE
SIZE
B
DWG NO
Blade_Drone
REV
SCALE
10 : 1
SHEET 1 OF 1

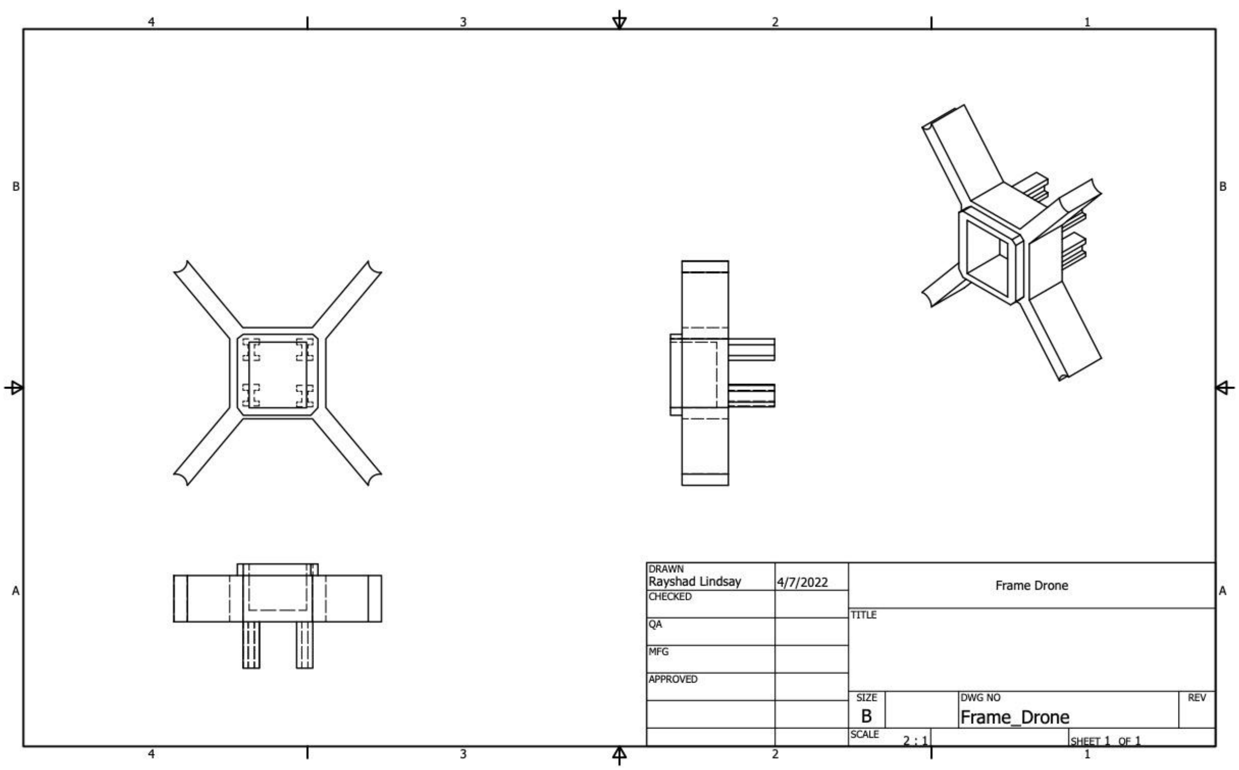


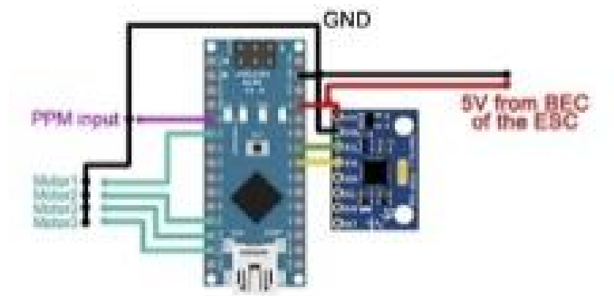
Y
Z
X

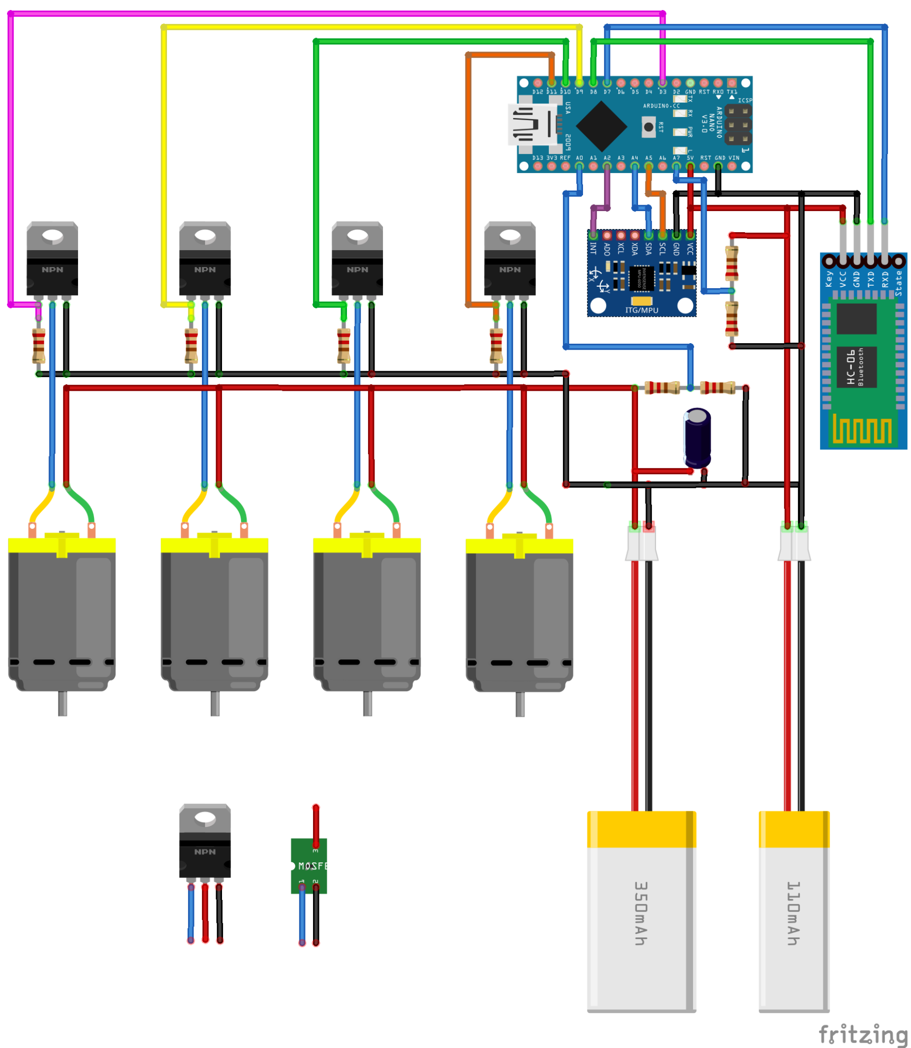


4
3
2
1
B
A
DRAWN
Rayshad Lindsay
4/7/2022
CHECKED
QA
MFG
APPROVED
TITLE
SIZE
B
DWG NO
Motor_Drone
REV
SCALE
10 : 1
SHEET 1 OF 1




DRAWN
Rayshad Lindsay
4/7/2022
CHECKED
QA
MFG
APPROVED
Frame Drone
TITLE
SIZE
B
DWG NO
Frame_Drone
REV
SCALE
2 : 1
SHEET 1 OF 1


GND
PPM input
5V from BEC
of the ESC


NPN
HC-06
350mAh
110mAh
fritzing

List of Tables

Frame Section Matrix Table

Supplier	Size	Design	Weight	Hardware	Extras	Score
NanoLongRange	135.0 x 128.5 mm	2 Parts	100 gram	M2 screws, glue, electrical tape	Camera, wire design, parts list	2
T4 QuadCopter	152.4 mm x 152.4 mm	4 parts	937 gram	8 M3 x 40mm+ bolts (arms to body) 8 M3 x ~5mm bolts (motors)	None	3
FPV quad	69.9 x 69.9 mm	3 parts	40 gram	None- Snap design	Camera and wiring design	1
PL1Q Vampire	104.9 x 104.9 mm	5 parts	231.6 gram	None- Snap design	Camera and wiring design	4

Brand	Mass	Ranked	Dimension	Ranked	Voltage	Ranked	Current	Ranked	C rating	Ranked	Cost	Ranked	Total
Walkera V120 and X100 helicopters	25 g	4	68 x 7 x 25mm	2	3.7 V	5	950mAh	2	25-50 C	4	$2.99	3	20
1/18 Brushless 4WD Stadium Truck	76 g	1	90 x 30 x 14mm	1	7.4 V	4	1300mAh	3	20-30 C	1	$8.99	1	11
Lipo Pack w/ JST Connectors	49 g	2	60 x 31 x 16mm	3	7.4 V	4	850mAh	4	25-40 C	3	$6.23	2	18
Lipo Pack w/JST-PH	20 g	5	53 x 31 x 7mm	5	3.8 V	7	650mAh	6	70-140 C	7	$4.60	6	37
Lipo Pack w/JST-PH	16 g	6	66 x 18 x 7.5mm	4	3.8 V	7	550mAh	7	70-140 C	7	$4.86	7	38
Lipo Pack w/XT30	33 g	3	47 x 26 x 14mm	6	7.4 V	4	450mAh	5	45-90 C	5	$5.44	4	27
Lipo Battery (Losi Mini Compatible)	9 g	7	34 x 20 x 7.5mm	7	3.7 V	5	300mAh	1	20-40 C	2	$3.49	5	27

Battery matrix

Abstract

The primary purpose of this class project is to create a product drone using cheap and small quality materials. We also planned to create unique features such as less impact on the drone, using any phone software apps as the wireless controller, and installing a camera. However, due to much time consumption, technical issues, and inexperience in solving error coding, we could not be able to make a complete product drone. Further development on this project will continue at a later date.

Introduction

The project is creating a customized product drone using the cheapest and most available materials. The project aims to install a camera on the drone as one of our drone's features. The next is to connect the drone with a wireless phone as a controller. The other goal is wiring the motor to the mainframe of the drone.

Background

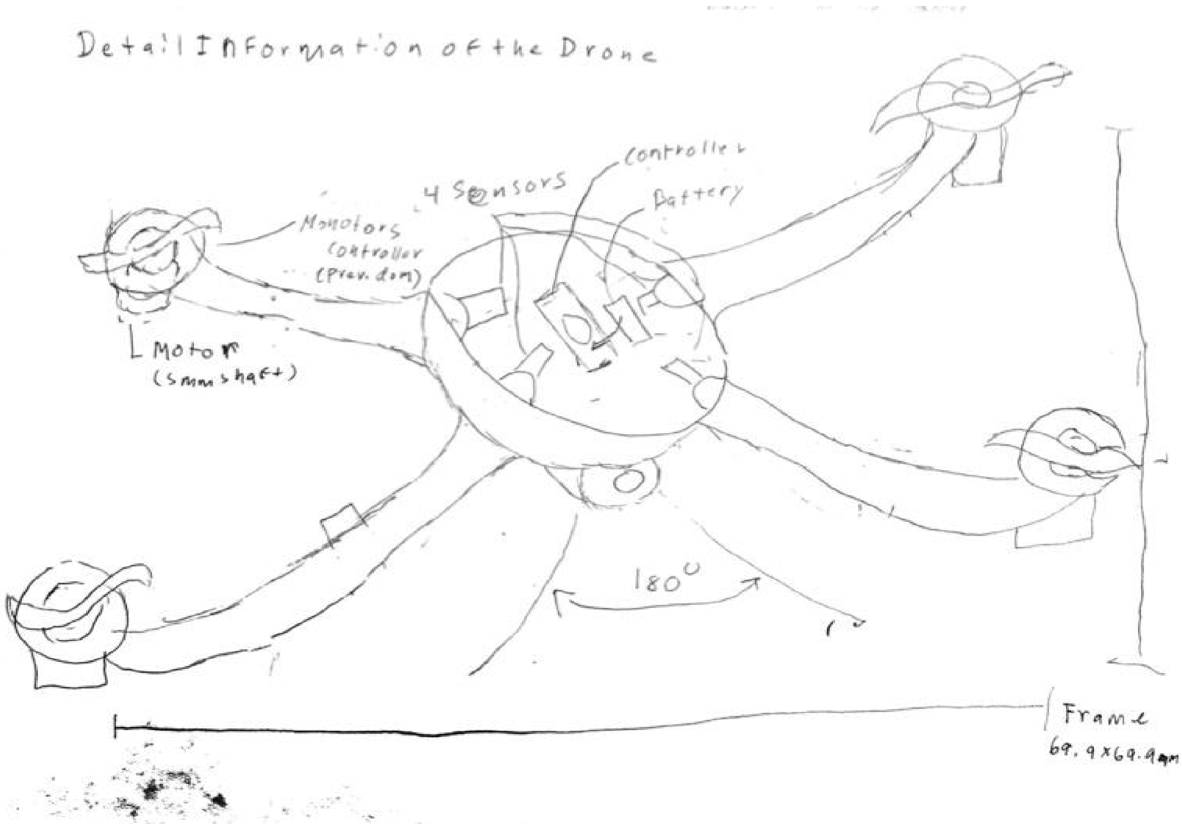
Population target and environment

The ideal focus of this project is to make the drone pleasant and well constructed for people to buy this in a rare and processed way. The target audiences are Aerospace and Mechanical engineering students, professors, and business companies for research purposes or interest in drones. Other audiences include STEM scientists and innovators, Drone Infusicists, and possibly military forces interested in this drone.

Brainstorming an ideal drone

During the second week of class, we work on deciding on our very own product from varieties of options and ideas in the fields of STEM. Most of the class has been designing a prototype drone using cheaper products in and outside the classroom. After deciding, we explore different drones on websites to determine which type of drone is perfect for our product.

Ideal Sketch



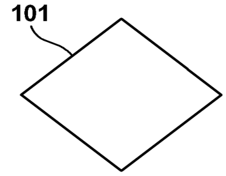
Constraints and design specifications

The drone also has features and components to the drone:

- Motors going on top of the motor to avoid crash impact damages or midair collisions
- 4 sensors on the same plane as the nano battery in case of any high heat performance when starting the drone
- Open excellent sensors for battery functions
- Easy to remove the battery
- Accessible charger

Prior to work

FIG. 1



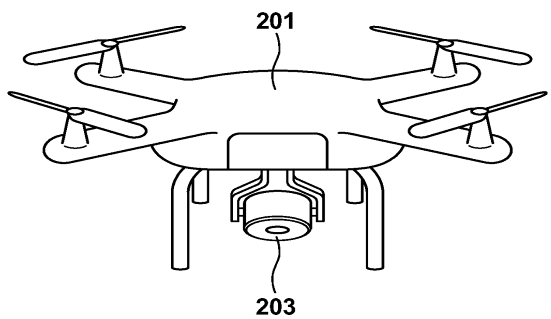


FIG. 2

Background Knowledge

In 1849, drones were once used as aerial military weapons when Austria assaulted Venice, containing packets of destruction ("Complete Evolution & History of Drones: From the 1800s to 2022"). From this ideal aircraft, an engineer named Nikola Tesla invented an aerial vehicle that controls remotely using a wireless beam transporting information and pictures from a short distance in 1898 (Nye).

During the early 1900s, drones became more popular, including toy and military drones, when the first director of Royal Aircraft in 1916 created a radio-controlled electric drone containing a gyroscope to the control system that ignites explosives (Nye). As the first drone developed, a new type of drone known as the 'Fairey Rotors' enables the drone to fly themselves using gyroscopes to help take control of the flight.

In 1918, a new low-cost radio-controlled drone aircraft called the De Havilland DH.82B Queen Bee aircraft. The purpose of this aircraft is as target practice for pilots in training (Vyas et al.). In 1936, the commander of the US Navy, William H. Standley, admired Queen Bee's advanced radio control and began to develop a system similar to the aircraft for combat purposes (Vyas et al.).

In the 1980s, drones became helpful for filming movie scenes and surveillance with a camera installed within the drone's system ("Complete Evolution & History of Drones: From 1800s to 2022"). By the 1990s, drones quickly became helpful for the CIA as spy planes allowed small-scale drones with high-resolution cameras. Today, drones can be used for aerial mapping, inspecting power lines, and delivering goods (“Complete Evolution & History of Drones: From the 1800s to 2022”).

Methodology

Description of Materials

- Arduino
- 4 brush motors
- Frame
- Small Drone Battery
- Speed Controller
- Wi-Fi signal
- Flight control board
- Propeller
- Sensor camera
- Remote XY (Software application for a wireless controller)

Descriptive Process

1. Determine the structure of our drone
2. Searching the type of drone for our product
3. Constructing size for our drone
4. The listing requires parts needed to construct our product
5. Searching each of the sizes of our parts to fit within our drone
6. Forming a matrix to determine the right part for our drone (i.e., size, cost, and compatibility)

7. Creating the blades for the drone using light-thin plastic
8. Constructing idea for the position of the drone's motors and propellers
9. Construct the type of controller that stabilizes the drone while it is in the air
10. Mount the motor controller
11. Determining the flight control of the drone controller.
12. Attach the correct size of the battery drone to the drone
13. Attach the main lead of the battery to the controller
14. Attaching each blade to the 4 four different brush motors
15. The idea of attaching a camera to our drone
16. Type of camera used to install our camera
17. Creating the Microbatic code for MOAB
18. Creating controller code (working code) for the controller using an IR Sensor Arduino code
19. Constructing a code for our drone control
20. Searching for the correct sensor for our drone control
21. Connect the drone to the sensors containing Wi-Fi
22. Installing a Bluetooth app on our phone

The rationale for each decision

The major problem with this project is determining the type of required materials to operate, such as capabilities and functions. When determining the motor, we have difficulty operating the drone, which could cause the motor to break due to the battery size and capacity. We used the matrix sheets ranking the second-highest choice battery in some cases. If the second choice fails, we need to search for another type of battery that fits the motors' requirements.

Alternate Solution

In our project, we were able to get the necessary parts for our drone by splitting it into various groups, making it quicker and easier to start the project. We were also able to make the blades for each drone motor in the correct sizes. We also have a wiring diagram with instructions on connecting the nano batteries with the motors and Adriano. The IR sensors were able to be placed into the drone, connecting it to the frame. We also found the camera placement and installation to it using the Wi-Fi UFO app as remote control via phone.

The major problem of this project was connecting the sensor that features a Bluetooth interface to the Arduino motherboard, connecting the phone as a wireless controller for the drone. When putting the sensors in the correct spot into the Arduino, we could not be able to get the phone app called the Remote XY wireless controller connected to the Bluetooth interface. We believed that the cause of this problem was that the coding that the Arduino we use is not the right type for the coding, and so we looked up other codes to help fix the situation. We also have a different look into other types of Bluetooth interface apps connecting to the Arduino.

The ideal cost of this drone should be downright cheap parts and materials capable of flying in the air without high expenses. The drone should be able to fly above 120 ft in the air away from the wireless controller with a camera visible to see in a 120 angle view when seeing into the phone app.

The technical assessments ensure that the drone's operational controls are straightforward. The first test to determine the drone's operation sequence is secure by re-insert the battery to the frames and then to the motors making sure that it is functional for air motion. The next step is to determine the coding runs the Bluetooth module sensor for the phone app able to connect with the Arduino by putting the wires in the correct spots.

The optimization techniques used for this project are minimizing the cost of the drone's material and focusing on their good and bad qualities. Although the quality of each material to assemble the drone is small, we can make and buy it at a low price cable of flying as much as high-cost drones.

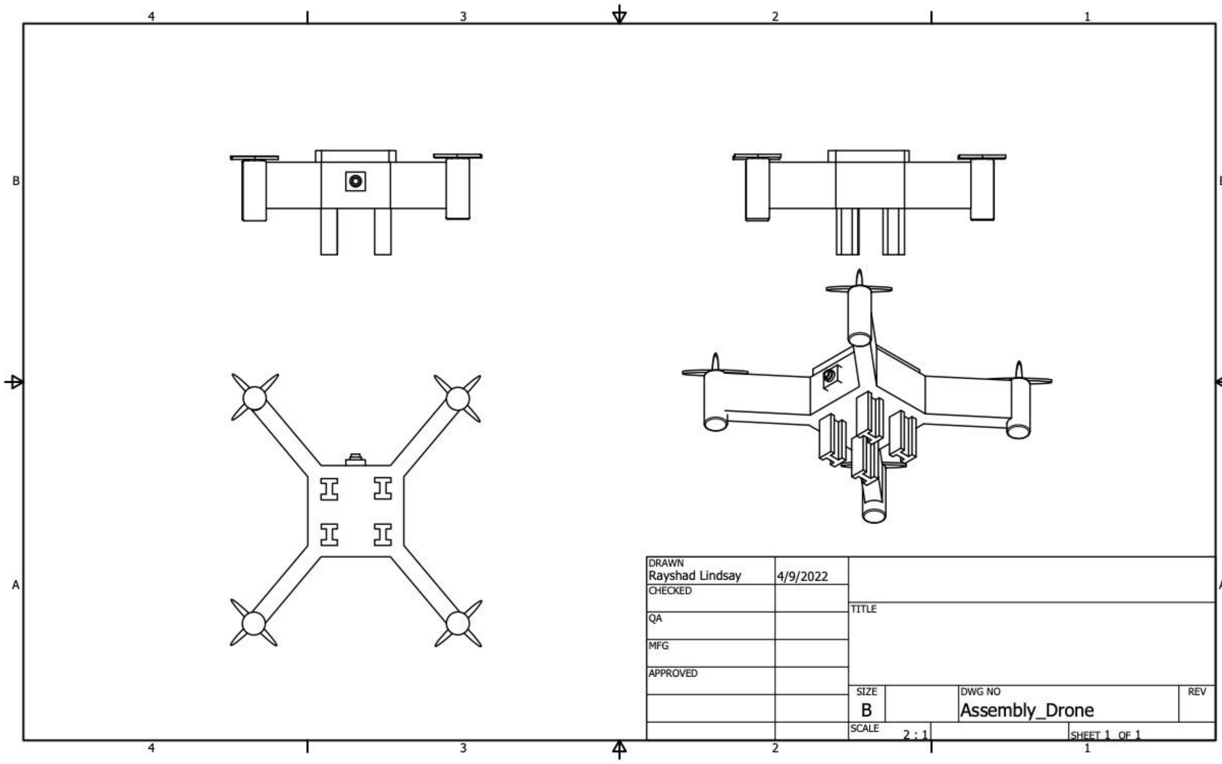
Final Design Solution

List of Decisions and justifications

- The frame length is 3x3 inches, a small Nano Long Range that consists of two parts; the top (containing arms and the majority of the frame) and the bottom (holding all the components).
- The bottom of the frame contains all components merge and linked together.
- The battery that connects the frame should be a small turning-nano and weightless for the bottom frame to carry the components.
- A high powerful camera was implemented to connect the Arduino with an ideal code capable of recording and saving videos on an SD card.
- The resource for deciding the code for our drone is the GitHub, consists with varieties of codes to help operate the Arduino using a wireless communication phone.

- We made special arrangements for a code to aid in adding the avoidance sensors to the sensors for better communication.
- Instead of using the Bluetooth interface, we were hoping to use a code for both the receiver and transmitter. Yet, it did not work.
- The ideal app for the Arduino Bluetooth controller is the Remote XY.
- The type of motors connected to our drone are brushless motors.

Detailed Drawing of Product



List of materials

- NewBeeDrone BeeEye Nano 600TVL FPV Camera
- Turnigy nano-tech 600mah 1S 35~70C Lipo battery (Nine Eagles Solo Pro 328, Eflite MQX, 120SR)
- Arduino Camera Sensor Module Framecapture
- DSD TECH HC-05 Bluetooth Serial Pass-through Module Wireless Serial Communication with Button for Arduino
- Pluto Controller
- Gy-521 MPU-6050 MPU6050 Module 3 Axis Analog Gyro Sensors+ 3 Axis Accelerometer Module
- 4 plastic blades Propellors
- Barebones code framework
- 3D printer
- Nano Long-Range Frame
- 3 Arduino Every
- 5 Gikfun Obstacle avoidance IR Infrared Sensor Module Reflective Photoelectric Light Intensity DIY Kit for Arduino UNO
- WYPH Mini Nano V3.0 Module ATmega328P 5V 16MHz CH340G Chip Microcontroller Development Board for Arduino Without USB Cable

Technical Data

Controller Code: IR sensors Arduino code

```
int LEDpin = 13;
int obstaclePin = 10;
int hasObstacle = LOW;  // LOW MEANS NO OBSTACLE

void setup() {
  pinMode(LEDpin, OUTPUT);
  pinMode(obstaclePin, INPUT);
  Serial.begin(9600);
}

void loop() {
  hasObstacle = digitalRead(obstaclePin);

  if (hasObstacle == HIGH) {
    Serial.println("Stop something is ahead!!");
    digitalWrite(LEDpin, HIGH);
  }
  else {
    Serial.println("Path is clear");
    digitalWrite(LEDpin, LOW);
  }
  delay(200);
}
```

Code for avoidance sensors

```
int irsensor= 5; //out on the module is connected to arduino's pin 5

int sensorvalue; // variable to store ir sensor reading

void setup()                    // run once, when the sketch starts

{

  Serial.begin(9600); //opens the serial port to communicate at 9600 bits per second

  pinMode(irsensor,INPUT); //irsensor is declared as INPUT

}

void loop()

{

 sensorvalue=digitalRead(irsensor);// gets sensor reading an stores in sensorvalue variable.

 Serial.print("Sensor =="); // prints Distance == on a row in the serial monitor

 Serial.println(sensorvalue); //prints the values stored in distance

if (sensorvalue==1){

Serial.println(" No obstacle");

digitalWrite(13,LOW);

}

else

{

Serial.println(" Obstacle ");

digitalWrite(13,HIGH);

}


}
```

Nano Arduino fight controller

Product Weight (Lbs):	0.04
Analog Input Channels:	12
Assembly:	Projects may require assembly
Clock Speed:	16 MHz
DC Current for 3.3V Pin:	50 mA
DC Current per I/O Pin:	40 mA
Digital I/O Pins:	20
EEPROM:	1 KB (ATmega32u4)
Flash Memory:	32 KB (ATmega32u4) 4 KB used by bootloader
Input Voltage (Limit):	6 - 20V
Input Voltage (Recommended):	7 - 12V
Length:	2 3/4"
Microcontroller:	ATmega32u4
Operating Voltage:	5V
Overall Width:	2"
PWM Channels:	7
SRAM:	2.5 KB (ATmega32u4)
Warranty:	1-year limited
Shipping Method:	UPS / FedEx

Gyro Sensor

- MPU-6050 MPU6050 6-axis Accelerometer Gyroscope Sensor
- Communication mode : standard IIC communication protocol
- Chip built-in 16bit AD converter, 16bit data output
- Gyroscopes range: +/- 250 500 1000 2000 degree/sec
- Acceleration range : ±2 ±4 ±8 ±16g

IR Sensor Obstacle avoidance (3 each at 120 Degrees) (4 each minimum of 90 degrees)

- Material: FR4
- Length: 45mm
- Width: 15mm
- Height: 10mm
- Weight: 4.20g
- Operating Voltage: 5v DC

Wi-Fi & Bluetooth

Brand	HiLetgo
Connectivity Technology	Bluetooth, Wi-Fi
Wireless Communication Standard	Bluetooth
CPU Model	Tegra
CPU Speed	2.4 GHz

- 2.4GHz Dual Mode WiFi + Bluetooth Development Board
- Ultra-Low power consumption, works perfectly with the Arduino IDE
- Support LWIP protocol, Freertos
- SupportThree Modes: AP, STA, and AP+STA
- ESP32 is a safe, reliable, and scalable to a variety of applications

Safety information and Operation manual

Beware that this drone is constructed with the cheapest products possible. The drone experienced significant and minor technical problems, including an electrical shortage, Wi-Fi connection loss, and calibration malefaction. Make sure to read the instructions carefully before starting the drone.

1. Before operating the drone, check to see that the system is working correctly.
2. Re-installed the battery into the center of the drone attached to the bottom frame, then repeat step 1.
3. Check that the IR sensor is connected to the Arduino tightly, with a color light appearing in the IR sensor, indicating that the sensor is connecting to a wireless phone.
4. Installed, the blades attach properly to each motor and ensure they can turn.
5. Once everything is in place, open the Remote XY app on your phone if you do not have it installed before starting the drone.
6. Connect the drone with your phone and check it is connection to the drone. Repeat step 3

Evaluation of the final solution

We were able to create the drone with the necessary parts to operate. The problem is that when testing the drone mainframe control using the Bluetooth interface app and IR sensors connecting to the Arduino, the IR sensor did not respond using the code. We could not figure out the source of the problem, either the code itself or the incorrect type of USB sensor to operate the drone.

Conclusion

There is a way to create a low-cost product drone using parts and materials that are entirely cheap and capable of flying, rather than buying a costly drone. However, we could not complete the entire drone in time and test its flight control due to technical difficulties when running the Arduino and IR sensors. This final design has many possible risks, including loss of control, landing impacts, radio interference, and electrical hazards when operating the drone. The project will continue to work with more people majoring in computer science, electrical engineering, or computer engineering who are experts and have experience in system installation.

Appendices

1. “Complete Evolution & History of Drones: From 1800s to 2022.” *Propel RC*, 31 December 2021, https://www.propelrc.com/history-of-drones/. Accessed 16 February 2022.
2. Vyas, Kashyap, et al. “A Brief History of Drones: The Remote Controlled Unmanned Aerial Vehicles (UAVs).” *Interesting Engineering*, 29 June 2020, https://interestingengineering.com/a-brief-history-of-drones-the-remote-controlled-unmanned-aerial-vehicles-uavs. Accessed 16 February 2022.
3. Nye, Logan. “This famous inventor designed drones before World War I.” *We Are The Mighty*, 4 May 2021, https://www.wearethemighty.com/mighty-history/tesla-designed-drones/. Accessed 16 February 2022.

All thanks to the rest of the team in the STEM 382 Industrial design course of 2022.