UPRM Roboboat Team: RUM-BA 3.0

Geovanna Goycochea Nazario
Team Captain
Mechanical Engineering
Mayagüez, Puerto Rico
geovanna.goycochea@upr.edu

Luis A. Vega Cordero
Team Co-Captain
Computer Engineering
Mayagüez, Puerto Rico
luis.vega21@upr.edu

Naidirah T. Barbot Cortés
Mechanical Lead
Mechanical Engineering
Mayagüez, Puerto Rico
naidirah.barbot1@upr.edu

Kaysie A. Cardona Pérez
Electrical Lead
Electrical Engineering
Mayagüez, Puerto Rico
kaysie.cardona@upr.edu

William Nogueras Vázquez
Drone Lead
Electrical Engineering
Mayagüez, Puerto Rico
william.nogueras@upr.edu

***Abstract*—The UPRM Roboboat Team decided that the original Autonomous Surface Vehicle – RUMBA, would be returning to the 2021 Roboboat competition. A detailed description of the competition strategy, design creativity and experimental results of the improved RUMBA model are documented and presented in this paper. A variety of innovative improvements were implemented to the ASV while taking into consideration the results of last year's competition. These adjustments include: fixing the hull to reduce lift forces, altering the top lid of the vehicle to reduce weight, the application of a thermal management to reduce overheating, the improvement of thruster efficiency, and the incorporation of simulations to test the vehicles coding. Identical to last year, the ASV will rely on the recreation of objects detected in a virtual mapping generated by the camera alongside ultrasonic sensors. The base communication system is designed using ROS, and the embedded system consists of a Pixhawk 4 controller that has Ardupilot connected to a TX2 for decisions and commands. Due to COVID restrictions, some of these improvements could not be manufactured or tested.**

Keywords— *AI, ASV, ROS, Ardupilot.*

I. COMPETITION STRATEGY

Due to the ongoing COVID-19 pandemic, which brought forth unprecedented changes like virtual classes, curfew and physical distancing, the team was restricted from meeting in the workshop most of the past year. The team decided to adopt a minimalist approach to this year's competition strategy, which is to minimize the amount of tech and manufacturing that had to be completed, while maximizing the performance. As such it was decided to utilize the previous boat the RUMBA but made alterations to it, to comply with improvements that the team has set out to accomplish since our first competition. The focus was on completing four of the six tasks, those being the mandatory navigation channel, speed gate, object delivery and obstacle field. The tasks of the mandatory navigation channel, speed gate and obstacle field were selected because they could all be performed with the current camera and sensor configuration. The object delivery task was selected to be performed because of the progress in design and manufacturing that the drone division has established. The team's objective is to perform these four tasks with satisfactory results.

A. Mandatory Navigation Channel

The Mandatory Navigation Channel was a priority since it is a mandatory task. RUM-BA 3.0 will determine the position of the buoys by using the camera. This will be used as the starting point. The ASV will go from one waypoint to the other using the You Only Look Once (YOLO) real-time detection system to detect the next gateway buoys and Stereo Depth to confirm midpoint positioning to pass through them. In case it gets near a buoy that is out of its Field of View (FOV), the ultrasonic sensors will help in the redirecting of the vehicle, avoiding the object.

B. Speed Gate

Comparable to the methodology used for the Mandatory Navigation Channel, RUM-BA 3.0 will use the camera and GPS to determine the position of the starting gateway. After going through, the vehicle will use 75% of the thrust to reach the blue buoy as fast as possible without losing vision of it. Then it will go around it and come back to the starting waypoint.

C. Obstacle Field

For this task, RUM-BA 3.0 will rely on the virtual map that was generated using data from the camera and the ultrasonic sensors, which are positioned in five locations around the vehicle. This combination performs obstacle avoidance considering all the buoys detected and allows safe navigation through the empty paths of the course.

D. Object Delivery

For this task, the drone division designed and manufactured our own autonomous drone, which will utilize an object collection mechanism to pick up and transport the objects towards the four corners of the platform. Specific details about Drone development and of the object collection mechanism are discussed further along the report.

II. DESIGN CREATIVITY

A. Vehicle Design

Assuming that the vehicle would solely be exposed to calm waters [1], all the analysis and design considerations were obtained by taking as a reference the Reed Canal Park Lake in South Daytona, Florida where the competition would have taken place.

1) Bow Level Correction: Given that during the last on-site RoboBoat competition RUMBA had faced problems concerning bow level, a slight curvature was added to the front of the otherwise completely flat hull. Vertical slices of 1/4 – inch PVC material were arranged to bring the boat's bow to a peak. The PVC pieces were then covered and shaped with an all-purpose putty material and later sanded down to the desired shape, as shown in Figure 1. Once this modification was applied, it was clear that it reduced water drag resistance and lift force; these observations were then confirmed through ANSYS simulations as shown in Figure 2. With this stability issue resolved, the ASV also experienced a correction with front camera visibility, along with improved maneuverability and agility.

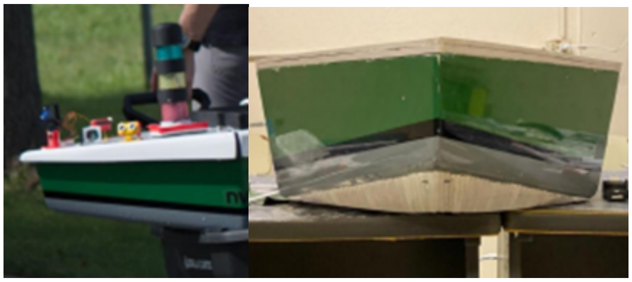


Figure 1. Comparison of the previous flat bow and the new rounded bow design.

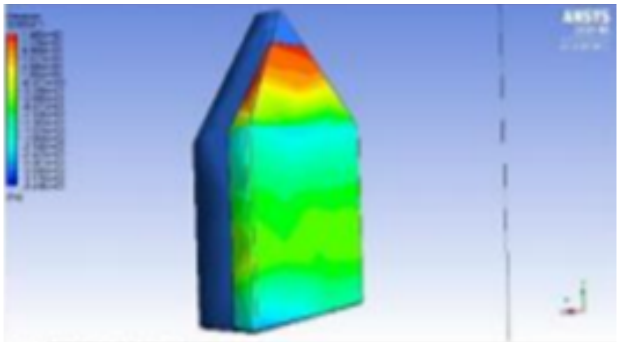


Figure 2. CFD analysis done on RUMBA 3.0 hull.

2) Thruster mount redesign: With the objective of reducing the drag force afflicting boat performance, the former thrust extensions were redesigned. Previously, the thruster mounts were geometrically square, wooden extensions which would exert great force against the water because of their large surface area. In response, "C" shaped mounts were designed, as shown in Figure 3. This design would reduce the overall submerged surface area without having to compromise the attachment's stability. Given the current situation regarding the 2020-2021 pandemic, these thruster mounts were designed and tested, but not manufactured.

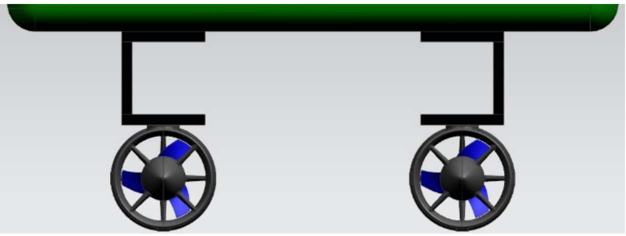


Figure 3. "C" shaped mount design.

3) Top lid redesign: The ASV's top lid was replaced in its entirety. Currently, the top consists of a lid lined with sealing foam material and operated with a pressurized lock, which should avoid any stray water almost in its entirety. As a bonus feature, the lid's reduced size not only provides a 1.5-inch clearance on each side for sensors, but also allows for a shorter cable distribution.

4) Thermal Management System: Due to high temperatures measured during previous tests, a thermal management system was designed to maintain the electrical components inside RUMBA at nominal temperatures. Although not manufactured, the team designed intake and exhaust vents, as shown in Figures 3 and 4, which will be utilized to increase airflow inside RUMBA by creating forced convection with the electrical components and thus lowering the general temperature. Both the intake and the exhaust will possess a fan to force the airflow. Safety measures were taken into account during the design of the cooling system to prevent any water from infiltrating into the ASV by utilizing a mesh that prevents liquid fluids from entering, which are commonly used in boats.

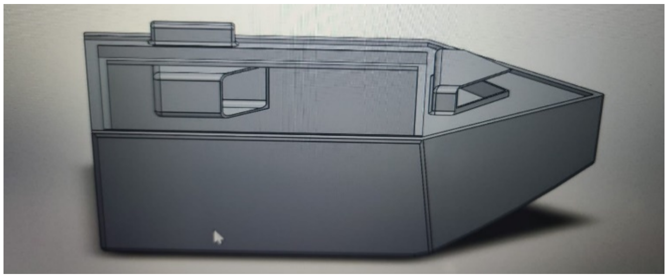


Figure 3. Side view of the cooling system proposal for the interior platform.

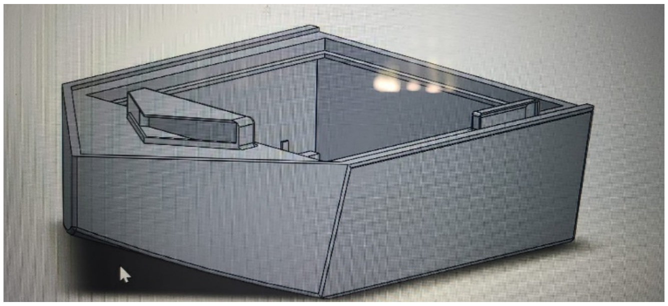


Figure 4. Front view of the cooling system proposal for the interior platform.

B. Electrical

1) Power System: Following the 2020 Roboboat competition, it was decided to change the Turnigy 10000mAh 4S 15C Lipo Batteries that were previously used to power the ASV. The main reasons as to why these batteries were changed are the following:

a) Battery Voltage: The Turnigy battery arrangement provided 14.8 Volts, which did not comply with the optimum voltage required by the Blue Robotics T200 thrusters. Nominal operation of these thrusters at 16 Volts is recommended for the best balance of thrust and efficiency.

b) Swollen Batteries: Multiple batteries had become swollen. This would result expensive for the team due to the cost of buying new batteries each time they became swollen. Additionally, the use of swollen batteries is dangerous and could put at risk the safety of the team members.

The advantages and disadvantages of different battery formats, chemistries and sizes were researched. After extensive analysis of the cylindrical, pouch and prismatic cell formats, the cylindrical cell format was selected. After analysis lithium-, lead-, and nickel-based battery chemistries, it was concluded that the best option was to select lithium-ion batteries. This battery format and chemistry were chosen due to their vast use in electric vehicle applications. There are also innumerable amounts of videos, books, websites, articles, and other resources that discuss the construction of cylindrical lithium-ion battery packs which were used as guides in the battery pack development process.

After analysis of the different cell sizes, the 21700 battery size was selected instead of the commonly used 18650 battery size because they provided a higher capacity rating with the least amount of batteries required for the optimal voltage and current. This would result in a more lightweight and economic battery pack.

The cylindrical lithium-ion batteries selected were the Molicel 21700 P42A batteries based on its positive reviews, economic price and high capacity rating. Each Molicel battery consist of a 3.6 Volt nominal voltage, a 4200mAh nominal capacity, a 45A continuous discharge rating, and an approximate weight of 67.8g. A 5S5P (5 in series and 5 in parallel) battery pack was developed using these Molicel batteries, which resulted in a nominal voltage of 18 Volts, a capacity of 21.0 Ah and a weight reduction of 1,113g when compared to the previous batteries used. These values were obtained using the following calculations:

a) *Voltage:* 5 batteries in series × 3.6V nominal voltage = 18 Volts

b) *Capacity:* 5 batteries in parallel × 4200mAh nominal capacity = 21,000mAh

c) *Turnigy batteries total weight:* 3 batteries in series × 936g each battery = 2,808g

d) *Molicel batteries total weight:* 5 batteries in series × 5 batteries in parallel × 67.8g each battery = 1,695g

e) *Weight reduction:* 2,808g – 1,695g = 1,113g

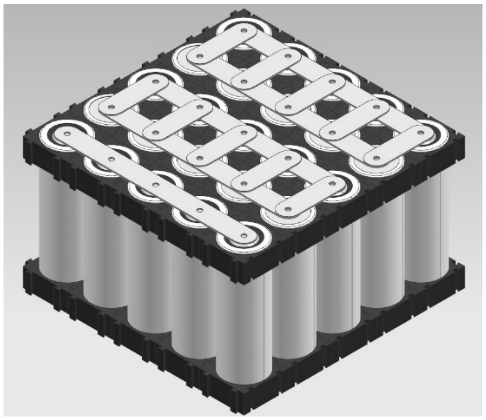


Figure 4. Isometric view of the 5S5P battery pack including series and parallel connections.

The battery pack could not be manufactured due to COVID restrictions and procurement issues. The battery shipment was canceled due to Federal Aviation Regulations.

2) Pegboard Design: A pegboard consists of a constant pattern of small holes. These holes can be used to support and fix in place various items. The electrical component board used previously was substituted with a pegboard platform. 3D printed holders that fit the pattern of the pegboard were designed for each electrical component. These holders secure the components in place as well as allow the flexible repositioning of every component if needed. Securing the components in place will allow the team to detach the board with the components from the boat and avoid the agglomeration of team members working on different aspects of the ASV at the same time. This design will also allow the interchangeability of the board between the current RUMBA boat and the new boat that will be used for the 2022 competition. This idea was inspired the 2019 Roboboat of the Georgia Institute of Technology Roboboat Team[2].

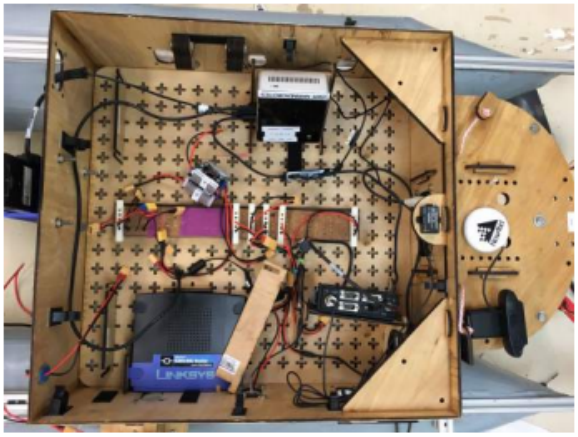


Figure 5. Pegboard Design Inspiration [2].

3) Telemetry System: Due to overheating and minor leakage issues experienced with RUMBA previously, research was conducted on a new telemetry system that will be used to monitor the temperature and water exposure inside the boat. This telemetry system consists of a temperature sensor and a leak sensor. The telemetry system will be connected to the Raspberry Pi 3, which will transmit the data obtained by the sensors to the Robotic Operating System (ROS) where the team will be able to keep track of this information and make adjustments if necessary.

a) *Temperatue Sensor:* Due to the overheating issues experienced, it was decided to implement a temperature sensor to measure the temperature conditions inside the boat at all times. This will allow the team to keep track of the temperature from a distance, without the need of stopping the boat and measuring it manually or exposing the electrical components to high temperatures which could decrese their longevity and reliability. The temperature sensor selected is the Waterproof DS18B20.

b) *Leak Sensor:* It was decided to implemet a leak sensor that will detect if there is a leakage inside the boat. In previous tests, the boat has experienced minor leakage issues yet nothing that could affect the conditions of the boat. However, a leakage in the boat could cause damage to the electrical components and surrounding, thus the decision to implenent a leak sensor was justified. It was decided to implement the LM393 FC-37 Sensor.

C. System

One of the main strategies in the design was to be able to modularize and divide the subsystem in such a way that the development and testing is not dependent on other subsystems. Considering our experience in last year competition, and the communication difficulties between the vehicle and the base, it was determined to start with a redesign of the full network. A ubiquity antenna was added to optimize connection and the final design of the network can be seen on Appendix A, Figure 7. Once the network was defined, the team proceeded to optimize the subsystems even further. ROS (Robotic Operating System) was kept as the foundation of the vehicle since it still follows the initial conditions of modularity that is being considered in the design. The main software subsystem is composed of perception, actions, and Artificial Intelligence (AI), and will be further explained.

1) Perception: A separated module was created to a conglomerate of all the data received from the different sensors to create a cleaner representation of the world. By separating this component, that is normally in conjunction with the AI, it can be tested separated from such. In other words, the dependency of our main AI algorithm from requiring real data for testing was eliminated. Instead of collecting or creating all the data from the sensors to test the AI subsystem, the team simply created the outputs of this module which are very easy to validate.

2) Actions: Another subsystem extracted from the regular AI module is the actions. Deliverables were made into a list of the necessary actions that the system needs to perform to complete all the tasks and created an API around it to be called by the AI system. By being their own module, the need of the AI having a physical system to test was eliminated and it became easier to debug.

3) AI: The AI module is essentially a group of decision trees. Each decision tree represents the different decision and actions to be taken during a task. These trees are then put into a queue depending on which task it to be performed first, giving us flexibility on how we want to approach the competition. In other words, the AI system listens to the perception subsystem to decide, and it calls the Action subsystem to make the actions. This system is broken up into two sections: ai_node and ai_tasks. In ai_tasks we can find the implementation of each of the AI trees. While ai_node contains configuration files and methods. As you may know, the most important part of the artificial intelligence section are the AI trees. Each of them has the objective of carrying out a series of steps to solve the tasks given in the competition. The process of creating an AI tree consists of 4 stages:

a) Stage I. Understanding the task: This stage seeks to ensure a clear understanding of the task that the artificial intelligence tree must do. For this, the rules of the competition are used, which gives us a visual and written idea of what the boat should do in each task.

b) Stage II. Logistics and structure design: At this stage, the design of the logistics and structure that the tree will have begins to be defined. To do this, a diagram is made, like the one in Figure 6, which details all the steps of what the tree must do to complete the task from start to finish. These diagrams should have a clear explanation of the methods and processes necessary for the task, so that those who see the diagram are clear about what the tree should do.

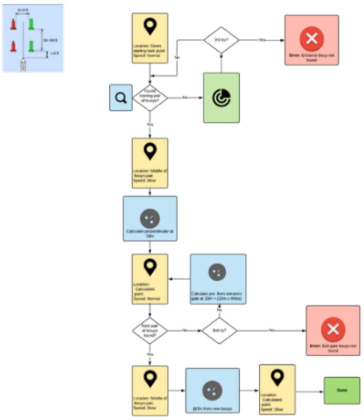


Figure 6. Mandatory Navigation structure and logic diagram.

c) Stage III. Implementation: This stage begins by reviewing and understanding the entire design of the tree that is going to be implemented. Then a tree diagram is made, like the one shown in Figure 7, which is similar to the real structure of what will be implemented. Each component that carries a series of conditions or processes has a pseudo-code written to make it easier to implement later. To implement this, the Toyota Research Institute library, named "task_behavior_engine", is used. This library is a behavior tree-based task engine written in Python. Before starting the implementation of the AI tree, it is verified that the necessary methods and functions are already implemented and that they work correctly. Once this is verified, the tree is created with the format provided by the Toyota library.

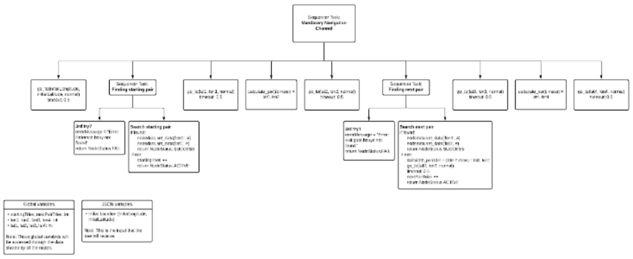


Figure 7. Mandatory Navigation Tree.

d) Stage IV. Testing: At this stage, the Google Maps application is used to perform a series of tests to ensure that what was implemented is working correctly. For this, some locations are determined to represent the buoys that will be in the task channel and another location for the initial location for the boat; an example of this can be seen in the Figure 8 where the boat is the red mark and the buoys the blue marks. Then this data is added as input to the AI tree, and it is run resulting in all the locations that the boat went to. Finally, those locations that were obtained as a result are taken and visualized in Google maps to see if it works correctly.

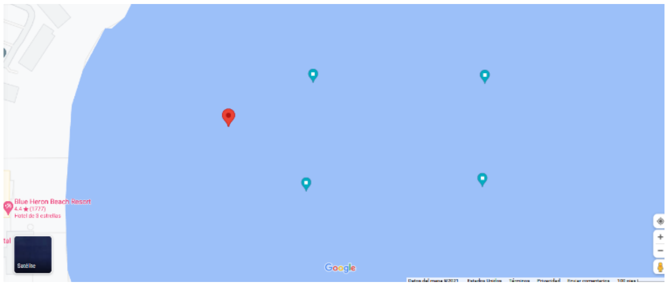


Figure 8. Mandatory Navigation Testing Map.

D. Vision

The vision system consists of a dual camera that is used to apply a SLAM with Stereo Depth to generate point cloud data from the visible surroundings. A clustering algorithm, HDBScan, is then used to identify object due to a high point density in a specific location. These detected objects are passed to the mapper with the corresponding location to position them in a virtual map appropriately.

E. Embedded

The embedded system consists of a Pixhawk 4 controller that has Ardupilot installed to be capable of autonomously move the boat to the different positions required. A TX2 is connected to it to take all the decisions necessary and to then give the correct commands to the Pixhawk 4 to acquire the correct positions. A Raspberry Pi is connected to the Pixhawk 4 and to all the ultrasonic sensors to implement a layer of object avoidance. Also connected to the Pixhawk 4 is the GPS, which is utilized to know the positions of the boat, and the radio receiver to receive manual inputs from a radio controller.

F. Simulation

Unity Real-Time Development Platform was used to simulate the tasks to scale as specified by the latest competition rules. This is done with the purpose of creating a user interface where the team can monitor the boat's interaction with the objects in the environment, that is buoys of different models, shapes, and sizes as specified and observe its telemetry data. Furthermore, the simulation generates the objects detected in the boats path as well as a boat model to convey an atmosphere as similar as possible to the surroundings of the real boat.

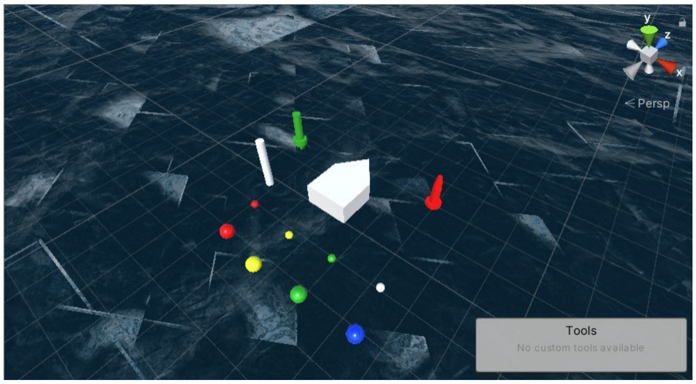
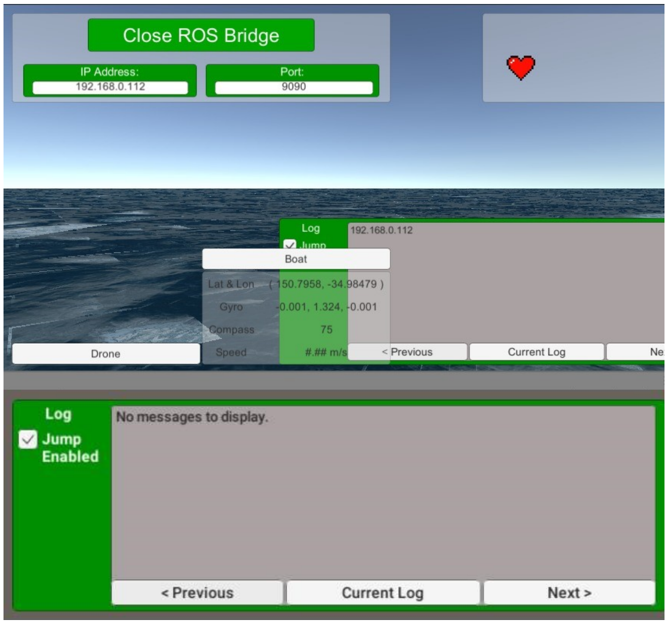


Figure 9. Boat and buoy models in the Unity Real-Time Development Platform to scale as specified in the current competition rules.

Figure 9 contains all the models that are currently to scale, this includes the boat and the buoys.Having the models to scale enables us to make better predictions of how the boat will go through the tasks in the real scenario by testing the artificial intelligence components of the boat.



IP Address: 192.168.0.112

Port: 9090

192.168.0.112

Lat & Lon (150.7958, -34.98479)

Gyro -0.001, 1.324, -0.001

Compass 75

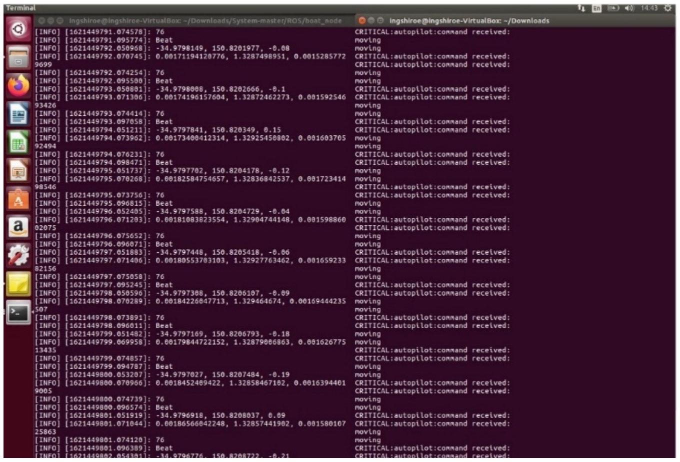
Speed #.## m/s

No messages to display.

Figure 10. User interface in the Unity Real-Time Development Platform to connect to the boat and monitor its telemetry data.

The user interface (UI) will be used to connect to the boat and monitor its telemetry data while it navigates the tasks in the simulated environment or the generated environment from the real task. The UI will be expanded to monitor the drone as well. The interface includes a message log which will display data related to the AI actions the boat takes.

As of now the UI is used to connect to a simulated autonomous vehicle generated using ROS in a virtual machine as well as Ardupilot. Figure 11 shows the simulated boat's telemetry data which Is sent to Unity using the ROS Bridge API. This data is displayed in the user interface, additionally it controls the boats behavior, such as moving locations and rotation.



```
[INFO] [1621449791.074578]: 76
[INFO] [1621449791.095774]: Beat
[INFO] [1621449792.050968]: -34.9798149, 150.8201977, -0.08
[INFO] [1621449792.070745]: 0.00171194120776, 1.3287498951, 0.0015285772
9699
[INFO] [1621449792.074254]: 76
[INFO] [1621449792.095500]: Beat
[INFO] [1621449793.050801]: -34.9798008, 150.8202666, -0.1
[INFO] [1621449793.071306]: 0.00174196157604, 1.32872462273, 0.001592546
93426
[INFO] [1621449793.074414]: 76
[INFO] [1621449793.097058]: Beat
[INFO] [1621449794.051211]: -34.9797841, 150.820349, 0.15
[INFO] [1621449794.073962]: 0.00173400412314, 1.32925450802, 0.001603705
92494
[INFO] [1621449794.076231]: 76
[INFO] [1621449794.098471]: Beat
[INFO] [1621449795.051737]: -34.9797702, 150.8204178, -0.12
[INFO] [1621449795.070268]: 0.00182584754657, 1.32836842537, 0.001723414
98546
[INFO] [1621449795.073756]: 76
[INFO] [1621449795.096815]: Beat
[INFO] [1621449796.052405]: -34.9797588, 150.8204729, -0.04
[INFO] [1621449796.071203]: 0.00181083823554, 1.32904744148, 0.001598860
02075
[INFO] [1621449796.075652]: 76
[INFO] [1621449796.096071]: Beat
[INFO] [1621449797.051883]: -34.9797448, 150.8205418, -0.06
[INFO] [1621449797.071406]: 0.00180553703103, 1.32927763462, 0.001659233
82156
[INFO] [1621449797.075058]: 76
[INFO] [1621449797.095245]: Beat
[INFO] [1621449798.050596]: -34.9797308, 150.8206107, -0.09
[INFO] [1621449798.070289]: 0.00184226047713, 1.329464674, 0.00169444235
507
[INFO] [1621449798.073891]: 76
[INFO] [1621449798.096011]: Beat
[INFO] [1621449799.051482]: -34.9797169, 150.8206793, -0.18
[INFO] [1621449799.069958]: 0.00179844722152, 1.32879006863, 0.001626775
13435
[INFO] [1621449799.074857]: 76
[INFO] [1621449799.094787]: Beat
[INFO] [1621449800.053207]: -34.9797027, 150.8207484, -0.19
[INFO] [1621449800.070966]: 0.0018452409422, 1.32858467102, 0.0016394401
9005
[INFO] [1621449800.074739]: 76
[INFO] [1621449800.096574]: Beat
[INFO] [1621449801.051919]: -34.9796918, 150.8208037, 0.09
[INFO] [1621449801.071044]: 0.00186566042248, 1.32857441902, 0.001580107
25863
[INFO] [1621449801.074120]: 76
[INFO] [1621449801.096389]: Beat
[INFO] [1621449802.054301]: -34.9796776, 150.8208722, -0.21
```

```
CRITICAL:autopilot:command received:
moving
moving
CRITICAL:autopilot:command received:
CRITICAL:autopilot:command received:
moving
moving
CRITICAL:autopilot:command received:
CRITICAL:autopilot:command received:
moving
moving
CRITICAL:autopilot:command received:
CRITICAL:autopilot:command received:
moving
moving
CRITICAL:autopilot:command received:
CRITICAL:autopilot:command received:
moving
moving
CRITICAL:autopilot:command received:
CRITICAL:autopilot:command received:
moving
moving
CRITICAL:autopilot:command received:
CRITICAL:autopilot:command received:
moving
moving
CRITICAL:autopilot:command received:
CRITICAL:autopilot:command received:
moving
moving
CRITICAL:autopilot:command received:
CRITICAL:autopilot:command received:
moving
moving
CRITICAL:autopilot:command received:
CRITICAL:autopilot:command received:
moving
moving
CRITICAL:autopilot:command received:
CRITICAL:autopilot:command received:
moving
moving
CRITICAL:autopilot:command received:
CRITICAL:autopilot:command received:
moving
moving
CRITICAL:autopilot:command received:
CRITICAL:autopilot:command received:
```

Figure 11. Boat's generated telemetry data in virtual machine using ROS.

G. Unmanned Aerial Vehicle (UAV)

The UAV vehicle is set for the "Object Delivery" challenge. This challenge is described as the delivery of four objects into delivery points located at each corner of a target area. The RUMBA UAV design is expected to be able to deploy from the ASV, fly to the target area, and deliver the objects to the center of the targeted area. The UAV was constructed using a DJI Flame Wheel F450 Airframe which was modified to include sensors, a battery, and a GPS. A "pick-up" mechanism for the collection of the four objects was researched, designed, and analyzed. However, due to COVID limitations, this mechanism could not be manufactured and integrated.

1) Weight Analysis: To confirm that the motors selected could launch the UAV from the boat, a weight analysis was conducted. For the total weight calculation 8 out of 10 components were used. The Turnigy 9×9 Ch Mode 2 Transmitter (control) was left out. The frame specifications are:

a) *iA8 Receiver:* 0.0286601 lb.

b) *Pixhawk 4 Autopilot:* 0.03483304 lb.

c) *Pixhawk 4 GPS Module:* 0.0705479 lb.

d) *Pixhawk 4 Power Module:* 0.0793664 lb.

e) *Telemetry Radio 915MHz:* 0.28125 lb.

f) *Motors:* 4×33.16oz. =132.64 oz.=8.29 lb

g) *Battery:* 0.901691 lb.

h) *Airframe:* 0.621704 lb.

After adding together the weight of all the components, the total weight resulted in: 2.72lb, hence the motors can lift the drone.

2) Buoyancy Analysis: The reasoning behind using the safety factor of the UAV to calculate buoyancy is that the dimensions of the components are cuboid, meaning that they do not account for the curves in the surface of that component. The UAV has many curved surfaces, and if the volume of the drone is calculated as a cuboid, the volume obtained will be higher than its real volume. When calculating the density of the drone, a lower density could be obtained since density and volume are inversely proportional with constant weight. Even so, a lot of the components do have a cuboid shape, hence the safety factor would not be an accurate representation, and curved surfaces are not usually spherical for the components. However, as this is a safety factor, it is always best to err on the side of caution.

For floating bodies, the weight of the entire body must be equal to the buoyant force, which is the weight of the fluid whose volume is equal to the volume of the submerged portion of the floating body. The equation is as follows:

$$F_B = W \rightarrow \rho_f g V_{total} \rightarrow \frac{V_{sub}}{V_{total}} = \frac{\rho_{avg,body}}{\rho_f}$$

The safety factor is the ratio between the volume of a cube and a sphere and is cal

a) Volume of cube: $l \times w \times h$=s3

b) Volume of sphere: 43r3=43s23=16s3

c) Cube to sphere volume ratio: s316s3=1.910

d) Weight for current design: ~850g

e) Volume required with SF:.~1650cm^3

f) PLA density: 1.24 g/cm^3

The main idea for now is using foam noodles of Polyethylene foam. It advantages are:

- Various densities: 1.2lb/ft^3, 1.7lb/ft^3, 2.2lb/ft^3…
- Inexpensive, should be buoyant enough, easy to mold to our needs.

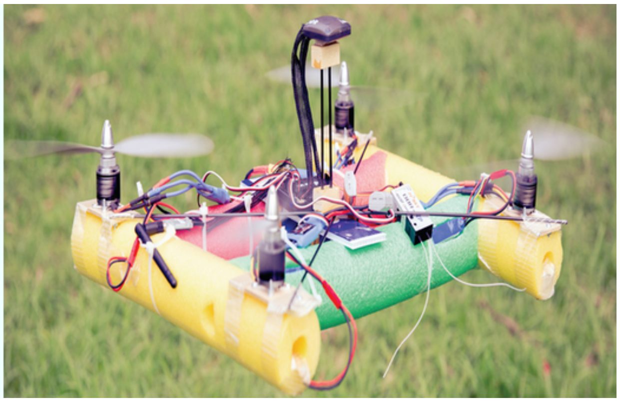


Figure 12. Foam noodles idea inspiration.

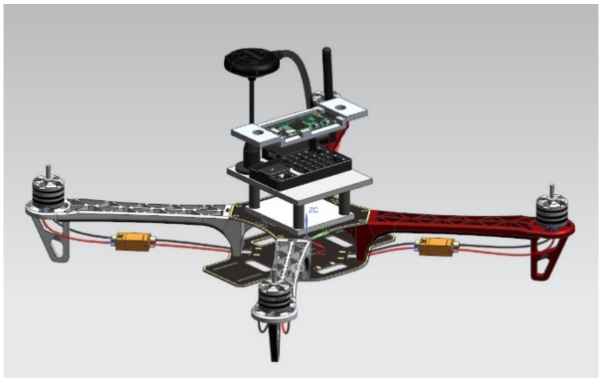


Figure 13. Isometric view of the final CAD design with components attached.

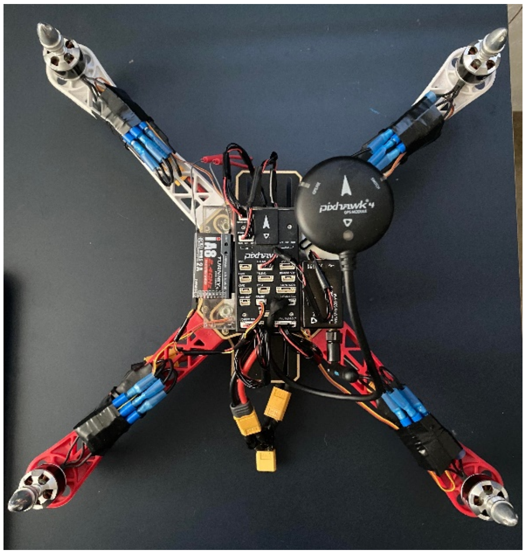


Figure 15. Top view of the final UAV prototype after wiring fix.

3) Pick Up Mechanism: The gears would be connected to a DC motor capable of moving both arms of the mechanism in order to expand and contract them easily to carry out the task of collecting objects. When performing a failure analysis and determining that a motor is not sufficient or safe for the mechanism, the arms of the mechanism would be independently connected to one DC motor each. To perform the object collection motion, the UAV must land on the object and, after a few seconds, the movement of the mechanism will begin by means of software programming and the use of a distance sensor. To prevent the mechanism from moving during landing on the ASV or during mid-flight, the UAV must be programmed so that when the sensor detects an object at a closer distance than the ground, the mechanism should expand and collect the object. If the sensor detects distances equal to or greater than that from the sensor to the ground, the mechanism will not move.

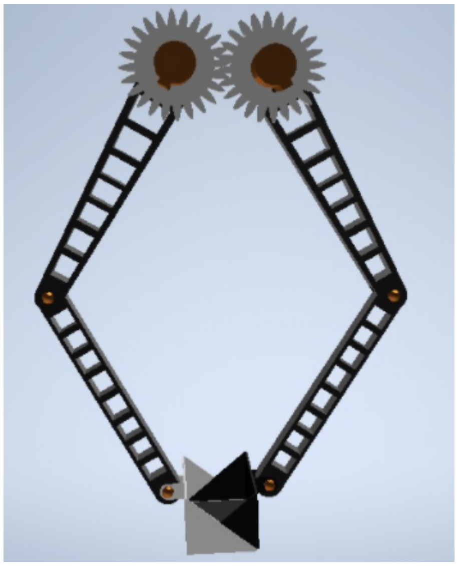


Figure 16. Position of the mechanism during the collection of objects.

Figure 16 demonstrates how the mechanism would look like when picking up a small object. The object could have any type of geometry; if the object was a round, for example, there wouldn't be any problems if it rolls since the walls of the mechanism would not allow the object to fall to the ground.

If the object is large, the mechanism is capable of expanding in order to carry out the task. In theory, the mechanism should be able to pick up larger objects and press them against its walls, opening more than presented in Figure 18, in order to avoid their fall and that the size of the object is not a limitation.

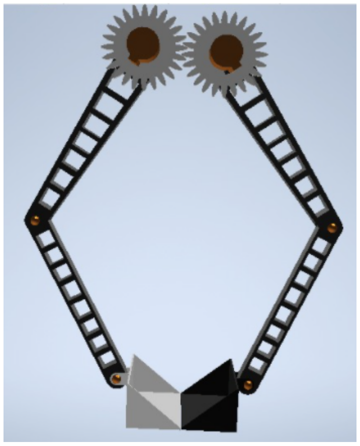


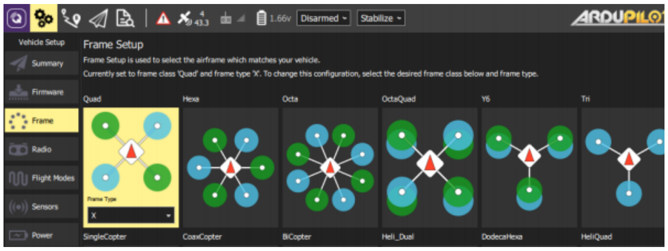
Figure 18. Mechanism with wider collection area.

4) Software: As a starting point, the team started working on fixing some of the wiring, electrical components and physical aspects of the prototype left by the group of last semesters. Some of the issues were that some wiring was not joined in a good way, and this was causing some lack of current trough the different motor and the PWM controller. After fixing these electrical problems, the team proceeded to begin with the implementation of the ArduPilot firmware using QGround Control to install the firmware and set up the airframe used for the ASV. After completing the setup, the recalibration of the different components of the UAV began. The calibrations were completed for the R/C control, GPS module, accelerometer, gyroscope, and compass of the drone.

The following figures present the tasks that were completed and a brief description of each one:

```
QGroundControl can upgrade the firmware on Pixhawk devices, SiK Radios and PX4 Flow Smart Cameras.
Plug in your device via USB to start firmware upgrade.
Found device: Pixhawk
Connected to bootloader:
Version: 4
Board ID: 9
Flash size: 2080768
Downloading firmware...
From: http://px4-travis.s3.amazonaws.com/Firmware/stable/px4fmu-v2_default.px4
Download complete
MAV_AUTOPILOT = 12
Successfully decompressed parameter_xml
Successfully decompressed airframe_xml
Successfully decompressed image
Erasing previous program...
Erase complete
Programming new version...
Program complete
Verifying program...
Verify complete
Upgrade complete
------------------------------------------
```

Figure 19. Update to the lastest ArduPilot firmware to the Pixhawk of the Drone using QGround Control.



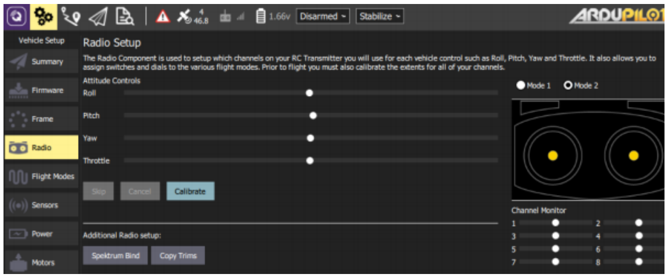
Frame Setup

Frame Setup is used to select the airframe which matches your vehicle.

Currently set to frame class 'Quad' and frame type 'X'. To change this configuration, select the desired frame class below and frame type.

Frame Type: X

Figure 20. Setting up the Airframe of the drone to the class of Quadcopter with X type frame.



Radio Setup

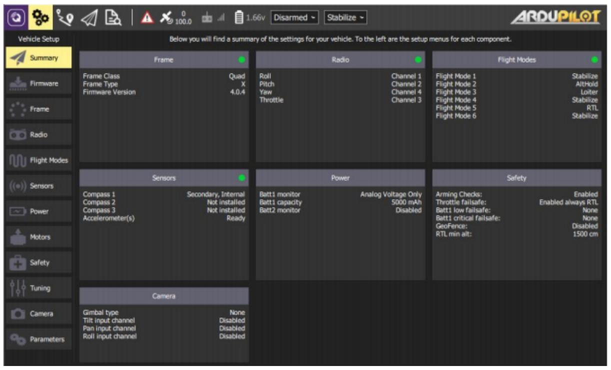
The Radio Component is used to setup which channels on your RC Transmitter you will use for each vehicle control such as Roll, Pitch, Yaw and Throttle. It also allows you to assign switches and dials to the various flight modes. Prior to flight you must also calibrate the extents for all of your channels.

Attitude Controls: Roll, Pitch, Yaw, Throttle

○ Mode 1 ● Mode 2

Additional Radio setup:

Figure 17. R/C Control and transmitter setup and calibration for vehicle control of Roll, Pitch, Yaw and Throttle.



Below you will find a summary of the settings for your vehicle. To the left are the setup menus for each component.

Frame
Frame Class: Quad
Frame Type: X
Firmware Version: 4.0.4

Radio
Roll: Channel 1
Pitch: Channel 2
Yaw: Channel 4
Throttle: Channel 3

Flight Modes
Flight Mode 1: Stabilize
Flight Mode 2: AltHold
Flight Mode 3: Loiter
Flight Mode 4: Stabilize
Flight Mode 5: RTL
Flight Mode 6: Stabilize

Sensors
Compass 1: Secondary, Internal
Compass 2: Not installed
Compass 3: Not installed
Accelerometer(s): Ready

Power
Batt1 monitor: Analog Voltage Only
Batt1 capacity: 5000 mAh
Batt2 monitor: Disabled

Safety
Arming Checks: Enabled
Throttle failsafe: Enabled always RTL
Batt1 low failsafe: None
Batt1 critical failsafe: None
GeoFence: Disabled
RTL min alt: 1500 cm

Camera
Gimbal type: None
Tilt input channel: Disabled
Pan input channel: Disabled
Roll input channel: Disabled

Figure 21. Summary of the calibrations made for each component. Green dot represents calibrations done correctly and the unmarked ones are the missing calibrations.

A draft of the set up for the autonomous control of the UAV was completed. A Raspberry Pi Microcontroller was implemented to send the commands for the directions and actions of the UAV. The Raspberry Pi Micocontroller setup can be observed in Figure 22.

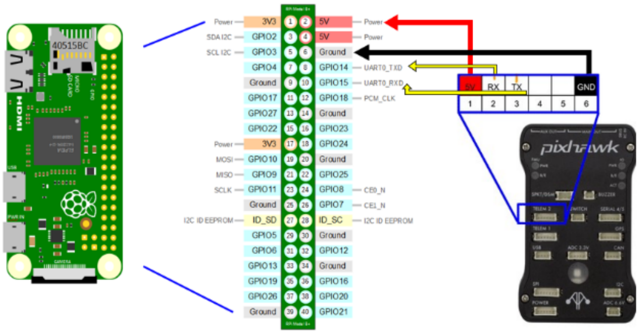


Figure 22. Raspberry pi set up with control module.

III. EXPERIMENTAL RESULTS

After modifications were done to the hull and overall design of the vehicle, the weight was reduced by 40% which is more than expected. Computational Fluid Dynamics (CFD) analysis was performed to evaluate the overall drag and lift forces being exerted on the hull. Such analysis demonstrates a reduction in both forces and overall better performance.

Due to the COVID 19 pandemic, the in-water tests performed were not performed. Instead, simulation tests were performed with the use of ROS and Unity. The reason for Unity instead of using already available simulated testing software Gazebo is that the user interface was made in Unity, making the implementation simulation for testing relatively easy with the nature of ROS. Additionally, most of the team members were using Virtual Environments to be able to use ROS, which are not ideal for performing simulated test, but since Unity runs on their native OS, the computing issue was solved. The initial results from the simulated test were promising although due to issues out of the team's control, a successful simulated task run has not been able to be completed. YOLO is still the main object recognition software used with the NVIDIA TX2 board which can run such. To train it for the real competition objects, data from the forums and last year's competition was collected to then perform data augmentation techniques to train a model that would recognize these objects.

ACKNOWLEDGMENTS

The UPRM Roboboat Team thanks the Association of Females in Mechanical Engineering for providing the tools needed to establish the team and for their effort to promote gender equality within engineering. This UPRM Roboboat Team is sponsored by the companies Lockheed Martin, General Motors, Boeing, NVIDIA, AMGEN, Procter & Gamble, SolidWorks, MathWorks and Blue Robotics. The team is grateful for the alumni that helped provide these sponsorships.

REFERENCES

[1] "Boat Pennsylvania Course." Hull Types andHow They Operate | | Boat Ed.com™,www.boat-ed.com/pennsylvania/studyGuide/Hull-Types-and-How-They-Operate/101039_101039011/.

[2] V. Ruia, R. Kuramshin, D. Foreman, S. Fish, E. Fu and P. Meyer, "Roboboat 2019: Technical Design Report," Georgia Institute of Technology, Atlanta, 2019.

APENDIX A: COMPONENT SPECIFICATIONS.

TABLE I
COMPONENT SPECIFICATIONS

Component	Vendor	Model/Type	Specs	Cost (if new)	Status
ASV Hull form/platform	Home Depot	Modified PVC	4’ x 8’	$89	Installed
Propulsion	Blue Robotics	T200	https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster/	N/A	Installed
Power System	Molicel	INR-21700-P42A	https://www.imrbatteries.com/content/molicel_p42a.pdf	$174.75	Selected
Temperature Sensor	Low Voltage Labs	DS18B20	https://www.terraelectronica.ru/pdf/show?pdf_file=%2Fz%2FDatasheet%2F1%2F1420644897.pdf	$8.99	Purchased
Leak Sensor	Acrobotic	LM393 FC-37	https://urolakostapk.files.wordpress.com/2016/10/yl-83-rain-detector-datasheet_low.pdf	$14.00	Purchased
Voltage Regulator	eBoot	LM2596	Input: DC 3 V to 40 V (input voltage must be 1.5 V higher than the output voltage, no boost) Output: DC 1.5 V to 35 V voltage is continuously adjustable, maximum output current is 3 A	$10.95	Purchased
Motor controls	Blue Robotics	Basic ESC	https://bluerobotics.com/store/thrusters/speed-controllers/besc30-r3/	N/A	Installed
CPU	Nvidia	Jetson TX2	https://developer.nvidia.com/embedded/jetson-tx2	$600	Installed
Compass	FPVDrone	Ublox M8N	Ublox Neo-M8N module Industry leading –167 dBm navigation sensitivity Navigation update rate up to 10 Hz Cold starts: 26s LNA MAX2659ELT+ 10• 25 x 25 x 4 mm ceramic patch antenna Rechargeable 3V lithium backup battery Low noise 3.3V regulator Power and fix indicator LEDs Protective case 30cm Pixhawk2.4 compatible 6-pin and APM compatible 5-pin 2 types of cable included Diameter 60mm total size, 32 grams with case	$27.89	Installed

Camera(s)	SVPRO	SV-960P2CAM-V90	1.3MegaPixels, Max. Resolution2560(H)X960(V) Sensor 1/3-inchOV9750 Pixel Size 4860µm x 3660 µm Resolution & frame MJPEG:2560X960@60fps/2560X720@60fps/1280X480@60fps /640X240@60fps Sensitivity 3.7V/lux-sec@550nm Mini illumination 0. 1lux Shutter Type Electronic rolling shutter /Frame exposure Connecting Port type USB2.0 HighSpeedFree Drive Protocol USB Video Class（UVC） Support OTG Protocol USB2.0 OTGAEC Support, AEB SupportAdjustable parameters Brightness,Contrast, Saturation, Hue, Sharpness,Gamma, Gain, White balance, BacklightContrast, Exposure Lens Parameter: M9 Lens HOV 90-degree, optional M12 no distortionlens,wide angle 180degree lens USB Interface Micro USB Operating Voltage DC5V/ OperatingCurrent160Ma~220Ma Working temperature -10～70°C/ Storagetemperature -20～85°C Board size / Weight 80X16.5mm, about30g Cable Standard 1M	$81.99	Installed
Aerial vehicle Airfame	DJI	Flame Wheel 450	https://www.dji.com/pr/flame-wheel-arf/spec	$62.00	Installed
Aerial vehicle platform	SYMIK	LP500E	https://www.amazon.com/dp/B087854R92/ref=cm_sw_r_wa_api_glti_CBEZ3CPD0GGMS5W30MTS	$14.99	Selected
Motor and propellers	EMAX	CF2822	http://www.yampe.com/images/pdf/4411.pdf	$31.56	Installed
Motor controls	HobbyKing	Turnigy 9×9Ch Mode 2	https://hobbyking.com/en_us/turnigy-9x-9ch-mode-2-transmitter-w-module-ia8-receiver-afhds-2a-system.html	N/A	Installed
CPU	PX4 Autopilot & Holybro	Pixhawk 4	https://docs.px4.io/master/en/flight_controller/pixhawk4.html	$244.95	Installed

Autopilot	ArduPilot	Mission Planner-1.3.7.4	https://ardupilot.org/planner/docs/mission-planner-features.html	Open Source	Installed
Team Size (number of people)	Undergrads	Electrical, Mechanical, Computer, Software, Business, Accounting, and Industrial Engineering	30 Members	N/A	N/A
Programming Language(s)	Python Microsoft	Python 3.9.5 C# 6.0	https://docs.python.org/3/ https://docs.microsoft.com/en-us/dotnet/csharp/	Open Source Free to use	Installed