Roboboat 2023: Technical Design Report UPRM Roboboat Team: CatamaRUM

José Yamil Soto Rivera
Team Captain
Mechanical Engineering
Mayagüez, Puerto Rico
jose.soto41@upr.edu

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

Yaidimar L. Pallens Rivera
Mechanical Lead
Mechanical Engineering
Mayagüez, Puerto Rico
yaidimar.pallens@upr.edu

Nomar Rodríguez Díaz
Electrical Lead
Electrical Engineering
Mayagüez, Puerto Rico
nomar.rodriguez@upr.edu

Josúe Rodríguez Mercado
Software Lead
Software Engineering
Mayagüez, Puerto Rico
josue.rodriguez38@upr.edu

Valeria Agosto Arroyo
Business Lead
Industrial Engineering
Mayagüez, Puerto Rico
valeria.agosto1@upr.edu

***Abstract*—The UPRM RoboBoat Team is introducing a new vehicle design for the 2023 competition: CatamaRUM. This report details the physical specifications of the Autonomous Surface Vehicle (ASV) and the electrical and software systems chosen to operate autonomously. The autonomy software was designed based on the competition strategy. The systems were designed based on the competition strategy detailed in this report.**

Keywords—*ASV, Catamaran, AI, Object Detection.*

I. Competition Strategy

After gaining experience from competing in person in 2019 and virtually in 2020 and 2021, the UPRM Roboboat Team decided to take on a new challenge: CatamaRUM. Since the team's inception in 2017, CatamaRUM is the second ASV that has been manufactured. Therefore, for this year, UPRM RoboBoat is highly motivated to achieve and reach as many new milestones as possible in the competition. The idea behind designing an ASV with a catamaran hull was to manufacture a more stable, lightweight, and compact ASV for this year's main goal, which is full autonomy. Given the main objective of achieving full autonomy during competition, the milestone for the 2023 International RoboBoat Competition is the completion of, at least, three of the following four tasks: Navigate the Panama Canal, Magellan's Route / Count the Manatees & Jellyfish, Northern Passage Challenge and the Beaching & Inspecting Turtle Nests. These tasks were selected because of their level of difficulty. Also, they were preliminarily developed but never tested for past competitions. First, CatamaRUM will complete the Navigate the Panama Canal task. Then, it will complete the Northern Passage Challenge. Finally, it will attempt the Magellan's Route and/or Beaching & Inspecting Turtle Nests. Depending on how much time is left on the run, CatamaRUM will attempt return to home as described by the Explore the Coral Reef task. The software division is prioritizing the software development based on task order as described above. Due to the lack of experience of autonomy on the team's history, it is important to accumulate as many points as possible in the rest of the categories. Consequently, the strategy is to also prioritize other aspects of the competition that are separate from the Autonomy Challenge.

When it comes to the tasks just mentioned, there is already a base code for the first two tasks. Priority is given to these, and they will be the first to be implemented and tested. After this, there will be a discussion on the others to decide which is going to be implemented or if both ca be done too. This depends on testing outcomes and the time available after the first two tasks have been completed.

II. ASV Design

A. Vehicle Design

The focus when designing CatamaRUM was to manufacture an ASV that was fully dismountable, compact, and lightweight. For this reason, the design consists of a 3-part assembly, which was manufactured using fiberglass and polyester resin with the addition of minor aluminum parts. Refer to Appendix A, Figure 1 for the CatamaRUM CAD design.

1) *Hull Geometry:* Research was conduted to find varying types of hull shapes to see which of these provided the best fit for the chosen criteria. (See Appendix A, Figure 2). The V-shape provides high stability in both calm and rough waters. Even though the selected shape does incur in a drag force penalty, having additional stability and maneuverability was paramount to the design. The specifications for each hull were set at 30 inches in length, with a width to height ratio of 8 to 6 inches to maximize volume displacement.

2) *Component Box:* The shape of the component box was designed to be completely removable while having a generous amount of space to work. The step-down on the front of the component box works as the base for the exterior components such as the light tower, kill switch and the Real Sense camera. Also, provides protection and neatness to the connections from the components to the component box. Mainly, the step allows the ASV to not exceed the height constraints while achieving a simple fiberglass manufacturing since it avoids complicated geometries. Final dimensions were determined to be: 20in x 26in x 9(6) in, where the 6in represent the height on the step-down of the component box. See Appendix A, Figure 3.

3) *Thruster Assembly*: At first, the thruster mounts were designed with several mounting locations to test for the best depth at which to place the thrusters. Experimentally, it was found that placing the thrusters at 4.5 inches below the hull resulted in the best performance for CatamaRUM. Since this is an outboard motor placed in the same position as fishing boats, yachts, speedboats, and others, there is greater information available that could be of use for improvements and ideas. Furthermore, outboard motors have proven to have great mobility because it is the best place to control the thrust vector. In addition, because it is easier to modify, it provides the ability to adjust the boat's trim angle, which can increase or decrease the ASV's performance. See Appendix A, Figure 4.

4) *Thermal Management System:* The heat generation by the electrical components inside the component box represented a major risk regarding the performance of the electrical system. Assuming a worst-case scenario, and steady state conditions, calculations were done by assuming that all energy supplied by the battery, would be dissipated as heat inside the component box. To effectively manage the generated heat, a fan was installed on both sides of the component box and a rear vent served as an intake to exhaust hot air. A temperature of 150 °F was defined as the maximum temperature inside the component box to ensure optimal performance. After calculations, it was concluded that an air flow of 30 CFM is necessary to avoid going over the maximum defined temperature.

B. Electrical System

The CatamaRUM operates by having 3 main components to achieve its AI operation. These components are a Jetson TX2, a Raspberry Pi 3 and different types of microcontrollers such as Arduinos. The Raspberry Pi will handle the AI which means that every signal and data that the AI will process will have to find its way to the Raspberry Pi. The Jetson TX2 will handle the computer vision aspect which in perspective has to work on the powering aspect of the Jetson. The Arduinos will handle the data recollection by implementing sensors and control the kill switches of the RoboBoat. Visit Appendix B, Figure 5 for power schematics.

1) Power System: The system is powered by a 11.1 V battery that will power the main processing unit (Jetson Tx2), the Raspberry Pi, and, along with a step down Converter, the Arduino Uno that's in charge of the various sensors and the light tower. The 11.1 V battery will also provide the power needed for the Arduino Uno to power the light tower. Alongside that system is a different circuit that uses the custom designed battery pack to power the thrusters. This circuit includes the required kill switches that would disconnect the thrusters.

a) Thruster Battery Pack: To power the Blue Robotics T200 thrusters, the system needs a battery pack that can handle some time in a full throttle position while also being able to power the boat during the duration of the challenges and gates. An adequate battery pack configuration would be 4S6P delivering 14.8V with a capacity of 15Ah see Apendix B, Figure 6.

b) System Battery Pack: The Team has acquired a 7,000 mAh 77.7Whr 11.1V Gens Ace Battery Pack that will power the 12V components through a connection terminal. Based on calculations, the system will consume 42.4 Whr during an hour of use with all components consuming max power. Therefore, the boat can operate at maximum power for 1.5 hours and still have a 15% of energy in the battery as a reserve. (see Apendix B, Figure 7).

c) Power Delivery: The Team has designed a Power delivery system that uses the system battery to power the 12V based elements. Firstly the Raspberry Pi will be powered thanks to its power adapter that connects directly to the XT60 port from the Gens Ace Battery, then a power distribution terminal will deliver 12V to the computers, microcontrollers and fans. Using a Buck converter the 5V systems, such as the antennas and the sensors, can be powered without the need of extra batteries.

2) Kill Switch: The team has set up two 30A 30V DC relays in-series through which the thruster current will pass. They're set up in such a way that if only one is deactivated no current passes through to the thrusters. Having the output port of one connected into the COM port of the next one secures that once one is deactivated neither will let energy flow. One of the relays is set up with the physical kill switch which (when pressed) will cut power from the relay setting it onto a normally closed position, not delivering power to the thrusters. The other relay will be able to be deactivated by a remote signal sent by the ground station if necessary. See Appendix B, Figure 8.

C. System

During the implementation of all software coding and decision making, the team decided to focus on: Navigate the Panama Canal, Magellan's Route / Count the Manatees & Jellyfish, Northern Passage Challenge and the Beaching & Inspecting Turtle Nests. These tasks were chosen because they would all run on the same object detection model and don't require additional hardware.

1) Vision Pipeline (See Appendix C, Figure 9 and Appendix C Figure 10): The first step in the ASV's decision-making process is collecting and processing the necessary data. The vision system is composed of a pipeline that receives images, passes them through an object detection model, and publishes messages containing the object's positions. The camera provides depth information as well as color information. The color information is essentially a regular image and is used for object detection. Once the objects are classified, the depth information is used to infer how distant they are. Since the decision trees work with GPS coordinates, it is necessary to convert the radial coordinates obtained to cartesian coordinates. Since the camera does not provide the angle, a function calculates it based on the position of the object within the image (Appendix C Figure 11).

a) Object Detetction: Object detection is handled by a YOLOv5 model. The biggest issue faced with this model is the low quality of the available data for training. To mitigate this problem, the team has created a synthetic dataset of around 1,500 images. These images were created from paper printouts of images of buoy images. This is a temporary solution, and the goal to have the ASV capture real images during operation that will serve as high quality training data.

2) Messaging: Due to its internal GPU, the TX2 runs the vision pipeline. Since the Raspberry Pi (RPi) is controlling the motors, it runs the AI decision trees. The gap is bridged using ZeroMQ (ZMQ). To compose and decompose messages on each end of the communication channel, Google's protocol buffers (protobuf) ensures message structure.

3) Ground Control Station: The goal is sending information wirelessly back and forth between the ground station and the boat. As messaging will be coded in Python, the idea was to develop a Python Real-Time Web Application so that data received and displayed updates constantly. First, the team designed and developed the interface's Front End and Back End. In this case it was decided to implement it with a web framework, Flask. The Ground Station will display some feedback in real time from the boat: location, distance, battery status, thrusters status and the current task being executed.

4) AI: Everything related the AI is processed by the raspberry pi. When working in AI the focus is to have an efficient workflow in order to comply with any time constraints.

a) Stage I. Understanding the task: To start off software members start to discuss the given task to clarify doubts and ensure that there is a clear understanding on what the goal is.

b) Stage II. Logistics and structure design: After everything is understood the next step is to create the descision tree. Here there is a step by step guide on all actions, descisions and considerations to make in the task. This graphical representation reduces future problems in implementation by giving a view into what functions will be needed and the conditions pertaining to each descision that will be made.

c) Stage III. Implementation: Right now the preferred programing language is Python for its lower learning curve. This is to ensure success within the timeframe available. During this phase, the team closely follows the trees and develop codes around them to reduce future problems during testing.

d) Stage IV. Testing: Before beginning in-water tests, the testing is done through unit tests as seen below in the experimental results and with the dronekit SITL simulated vehicle. With it, commands and basic scripts were tested to avoid accidents when going to in-water testing later. After enough proof and confidence of the codes performance was attained, the team passes on to water testing.

D. Embedded

Due to overheating and minor leakage issues experienced with RUMBA previously, research was conducted on a new Embedded system that will be used to monitor the temperature and water exposure inside the boat. This Embedded system consists of 3 temperature sensors and 4 leak sensors, and relays to control the light tower.

1) Raspberry Pi with Navio Hat: The Raspberry Pi will oversee the AI, while at the same time use the ArduPilot system which enables RC capabilities. The RC capabilities are implemented by using a Raspberry Navio which works as the previous component that the Rumba used which was the Pixhawk. This hat allows the use of ArduPilot connect to a computer with Ardupilot compatible software for the setup of the RC capabilities. By allowing the same system of the ArduPilot on the raspberry pi is a simple way to connect what was the Pixhawk to the AI since the Raspberry Pi. See Appendix D, Figure 12

2) Telemetry System: The telemetry system hast to emcompass the three main components the Raspberry Pi, Jetson TX2 and the Arduino Mega. The Raspberry Pi will be connected to the Arduino via it's USB ports , which will transmit the data obtained by the sensors to the Raspberry Pi where the team will be able to keep track of this information and make adjustments if necessary (see in Apendix D, Figure 13). The Raspberry Pi is also connected via its USB port to the TX and RX pins to the Jetson TX2 this way the camara data processed by the Jeston TX2 can help take decisions with the AI (see in Apendix D, Figure 14). The wireless killswitch will be implemented with arduino antennas. There will also be an antenna in the boat which will facilitate the data transfer between the vision and AI computers and which will also grant the team the ability to send live data to the ground station.

a) Temperatue Sensor: Implementation of the temperature sensors to measure the temperature conditions inside the boat at all times, more specificlly key components like the Jetson, the Raspberry pi and the ESC's. 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 of key components.

b) Leak Sensor: Implementation 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. The boat will have the sensors placed in the bottom of the boat in every corner.

3) Arduino Mega: The Arduino Mega oversees the sensors and actuators of the Roboboat. This way it is possible to achieve a connection between the AI and the current risk status of the Roboboat. This way it is possible to turn on the light tower when the Roboboat's mode is currently in execution.

4) Arduino nano: The arduino nano will be implemented for the wireless killswitch. One Arduino nano will act as a receiver ubicated on the boat, it will have a digital signal connected to the relay to open the circuit if the remote killswitch is pressed. A second Arduino Nano will be implemented as a transmitter in the form of a controller with a button. This button will send a signal via the antenna, which will be received by the Arduino Nano on the boat and turn off the thrusters.

III. Experimental Results

A. Weight

The weight was estimated in an excel spreadsheet by using density and volume for each main part. Initially it resulted in approximately 25 pounds. Although, when the manufacturing was finalized it resulted in 35 pounds. This represents a 40% margin of error on the weight calculations.

B. Network

a) Messaging: Tested messaging between ground control station and raspberry pi both via ethernet and antenna. This was done by running a test subscriber file on Raspberry pi and a test publisher file on ground station. It took a while to understand the network this messaging pattern was going to implement. The team learned about obtaining addresses that both publisher and subscriber's sockets had to bind and connect to respectively.

b) Connections: Altering Navio's file of configuration to modify default settings, the team was able to change telemetry settings and give ground station's IP address so that GCS software (Mission Planner and QGroundControl) could connect to Raspberry Pi. This was necessary for monitoring of the vehicle and test commands that were going to be used on the python script in the future.

c) Ground Station to Raspberry Pi with Python script: A test that was necessary was to be able to control and communicate with vehicle via Python script, so that AI and Image Processing algorithms can receive data of vehicle's behavior Dronekit, provided a set of APIs to help the team do this, providing functions to connect, arm, disarm, obtain boat location, distance between GCS and boat and movement commands.

C. Unit Testing

Functions used throughout the ASV's software systems are validated through unit testing. For example, for functions that convert between global coordinate systems, calculations were made by hand for a variety of cases. The functions are automatically tested against these cases and other edge cases to ensure their behavior is correct. Unit tests are implemented using Python's unittest library.

D. Other Tests

Automated tests were used to examine the runtime of the different components in the vision pipeline and ensure it remained reasonably low. Additionally, tester programs were implemented to facilitate manual testing.

I. ACKNOWLEDGEMENTS

The UPRM Roboboat Team would like to acknowledge first and foremost the team’s Software Division Leader from 2019-2021, Pedro Santiago Dominicci. With his technical help, the team improved the algorithms and gained necessary knowledge to persevere. Christian J. Lagares Nieves, PhD student, is also to be recognized for his contributions in setting up CUDA on the Jetson TX2 and providing helpful resources to navigate this area of knowledge. Furthermore, gratitude is extended to the team’s academic advisor Dr. Rúben Diaz who provides us with his expertise in Mechanical Engineering and with resources to help complete this project. Thank you to all the UPRM Roboboat Team members that have contributed since the team’s inception in 2017. With your hard work, support, resources and connections, the team has surpassed set goals and milestones. Finally, deep gratitude is extended to the sponsors and donors.

Rerefernces

[1] “Welcome to DroneKit-Python's documentation!¶,” *Welcome to DroneKit-Python's documentation!* [Online]. Available: https://dronekit-python.readthedocs.io/en/latest/. [Accessed: 2022].

[2] “Rover home¶,” *Rover Home - Rover documentation*. [Online]. Available: https://ardupilot.org/rover/index.html. [Accessed: 2022].

[3] “Pyrealsense2¶,” *pyrealsense2 - pyrealsense2 2.33.1 documentation*. [Online]. Available: https://intelrealsense.github.io/librealsense/python_docs/_generated/pyrealsense2.html. [Accessed: 2022].

[4] “Python,” *ZeroMQ*. [Online]. Available: https://zeromq.org/languages/python/. [Accessed: 2022].

[5] Ultralytics, “Ultralytics/yolov5: Yolov5 in PyTorch > ONNX > CoreML > TFLite,” *GitHub*. [Online]. Available: https://github.com/ultralytics/yolov5. [Accessed: 2022].

APPENDIX A

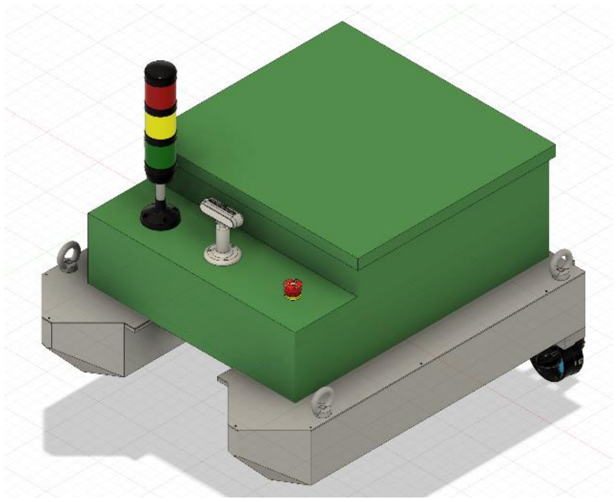

Figure 1. CatamaRUM assembly.

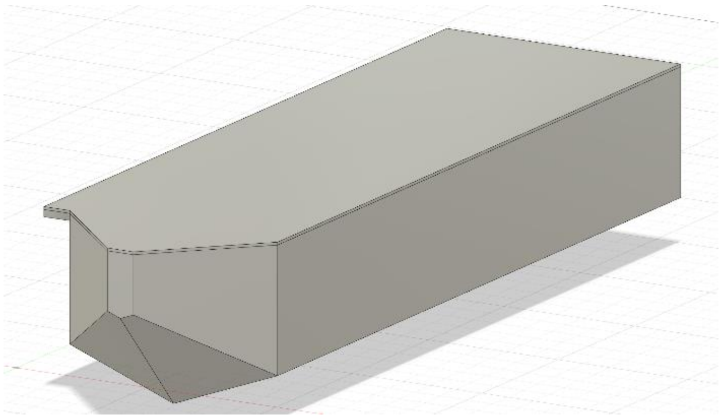

Figure 2. Right hull design.

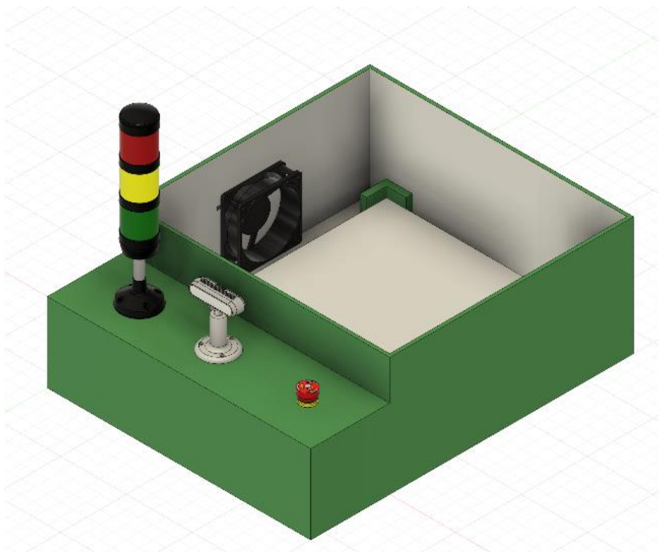

Figure 3. Component box.

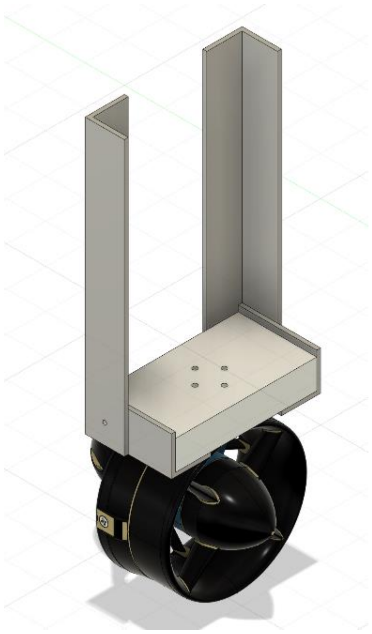

Figure 4. Thruster assembly.

APPENDIX B

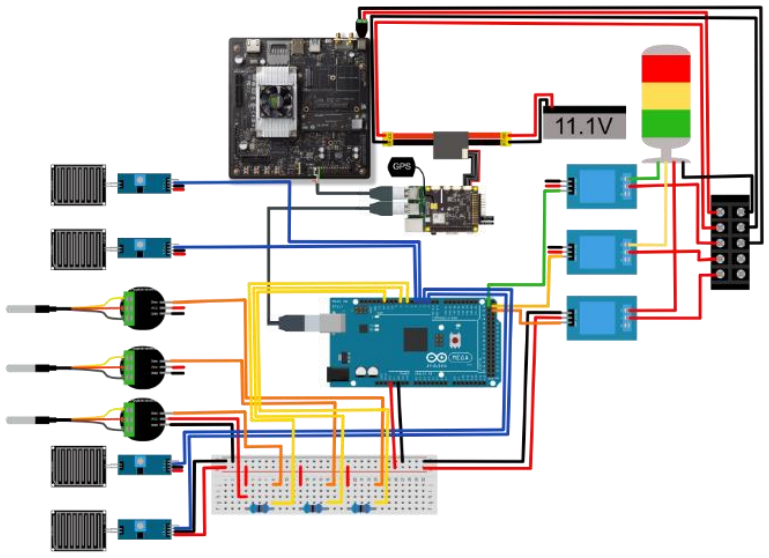

Figure 5. Power and data circuit diagram.

Name	Capacity	Voltage	Chemistry	Photo
Custom made Battery	15,000	14.8V	Lithium Polymer	

Figure 6. Thruster Custom made battery pack

Name	Capacity	Voltage	Chemistry	Photo
Gens Ace	7000 mAh	11.1V	Lithium Polymer	

Figure 7. System Battery pack

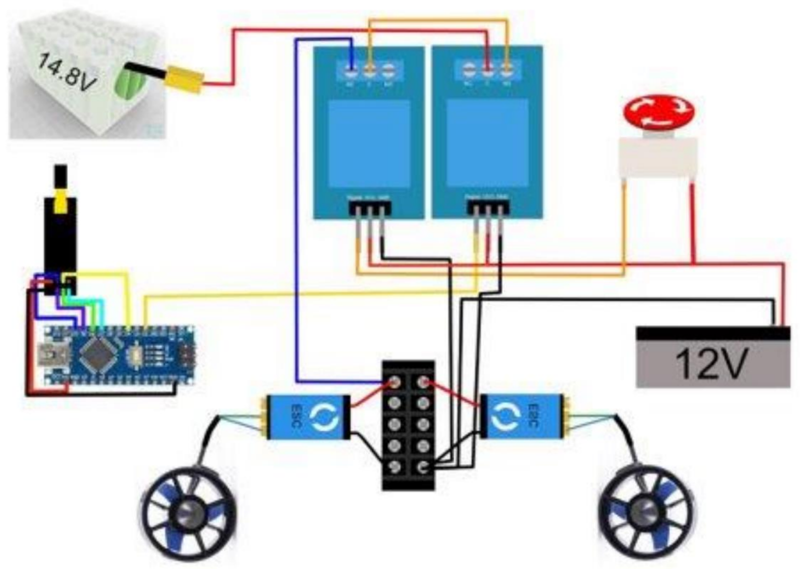

Figure 8. Thruster Kill switch circuit diagram.

APPENDIX C

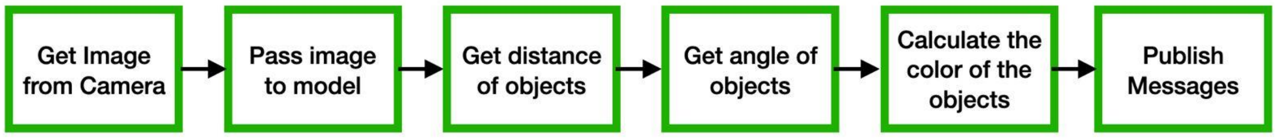

Figure 9. Vision pipeline command order.

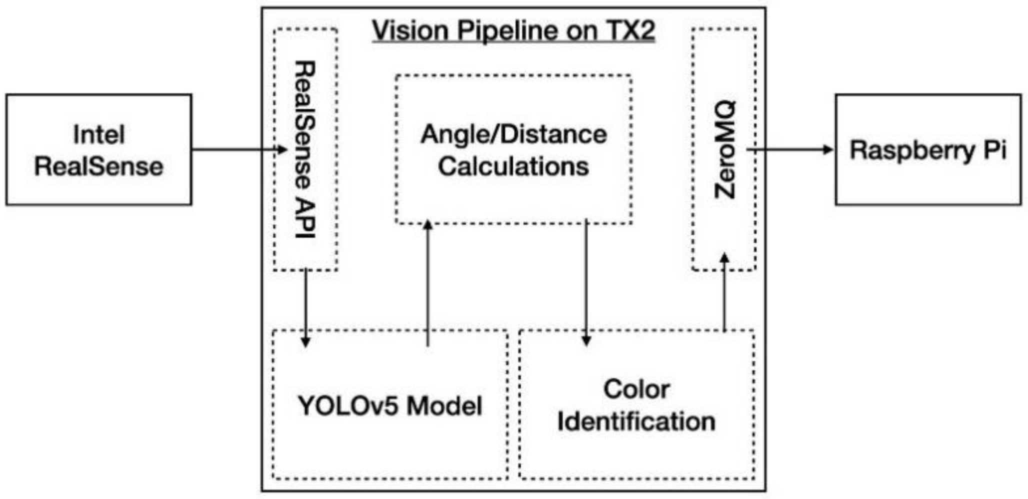

Figure 10: Vision Pipeline Interfaces Visualization

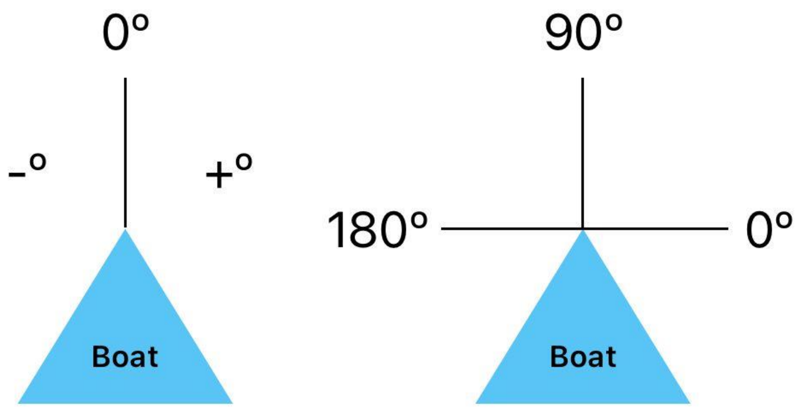

Figure 11. Camera’s angle measurement

APPENDIX D

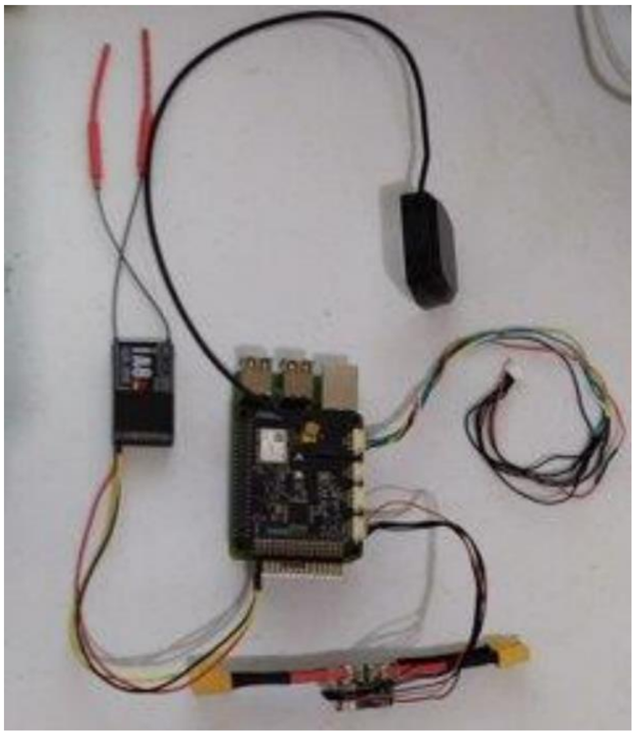
Figure 12. Raspberry Pi Navio Hat set up.

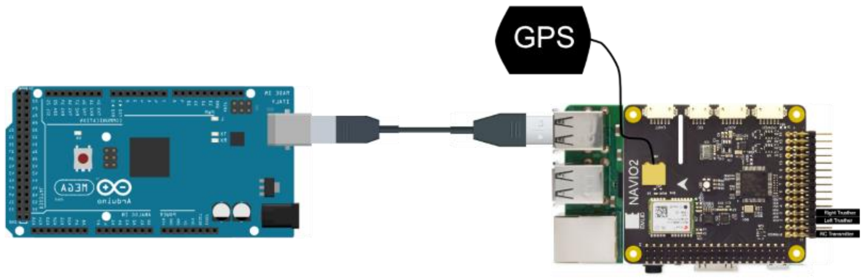

Figure 13. Arduino Mega and Raspberry Pi serial communication connection.

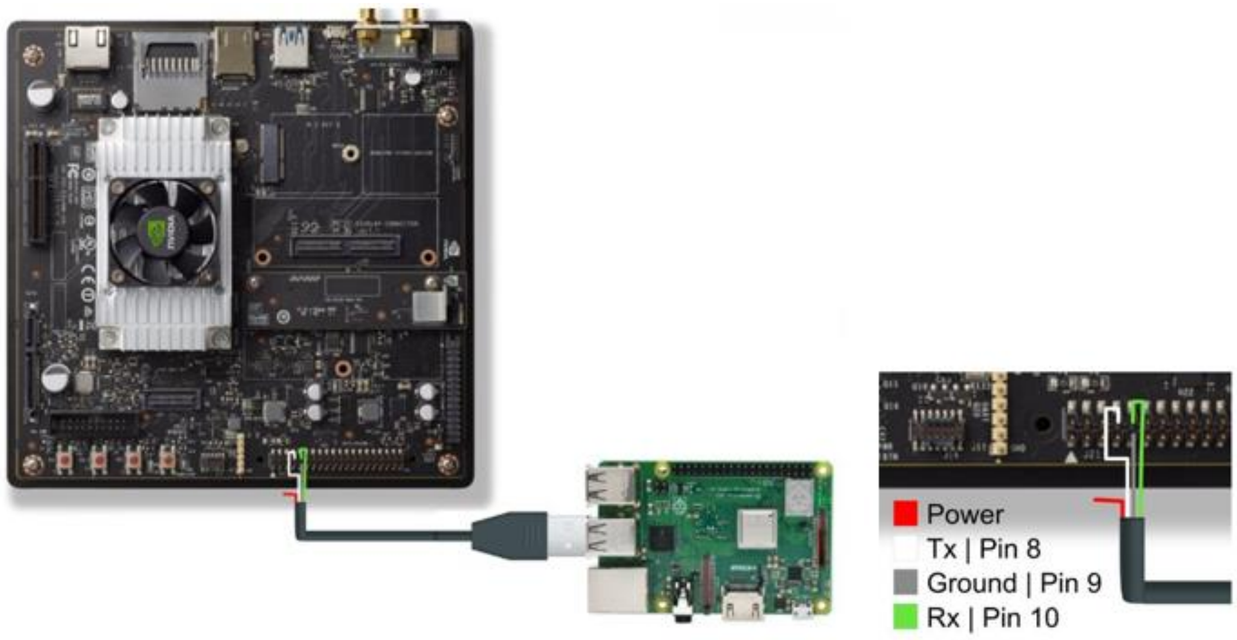

Figure 14. Jetson TX2 and Raspberry Pi 3 USB to UART connection.