MHS Technical Design Report 2023

Rebekah Lara, Minh Nguyen, Benjamin Sitdham, Aspen Bedo, Aidan Nguyen, Sierra Nguyen, Sebastian Rodriguez, Emilio Reza, Karmen Chandler, Jerry Minh Tram, Dr. Chris Dale McMurrough

Abstract

The following Design Report serves as a record of the first Martin High School Roboboat team's design, thought-process, organization, and methods as we pursue Robonation's 2023 Roboboat Competition. This document is designed to be referred back to in the future for tips, changes in process, or as a guideline. Includes documentation on the first usage and implementation of buoyancy design, Depth Sensors, etc. The goal of this report is to serve as a framework for a club that consistently changes members.

I. INTRODUCTION

As a pioneer team in our state and high school for the RoboBoat competition, we decided to prioritize familiarizing ourselves with the competition and engineering process. As this will be the first time the team competes, we decided to focus on developing new skills – such as combining advanced hardware with unfamiliar software. A large portion of producing the ideal RoboBoat goes beyond the current capabilities of high school physical and intellectual resources. This mandated a plan of action with the ultimate goal of meeting the qualification guidelines for the 2023 Roboboat competition. As such, MHS Roboboat (SEALS) aims to undertake the necessary responsibilities to lay the foundation, both as competitors and for future years' teams to build off of.

II. COMPETITION STRATEGY

A. Course Approach

As RoboBoat 2023 will be our team's first experience at the competition, our priority is to meet the qualifications of the competition. This includes testing to ensure the boat is a water-safe vessel and possesses basic motor and sensor configuration to navigate the starting gates of the field. If the initial goal is met, the team's next iteration in development will involve passing the Magellean's Route and the Northern Passage challenge.

B. Time and Members Management

Prior to kicking off the school year, we networked with Jerry Tram and Dr. McMurrough at the University of Texas at Arlington for access to high quality lab space with resources such as 3D printers, laser cutting machines, and a safe area for work.

In addition, these mentors advised the various necessary components and tools required for autonomous boat function, including but not limited to: Cube Orange and Here3; and mentoring on difficult materials such as carbon fiber and fiberglass. This access to more resources and a professional perspective enabled the team to manufacture the boat utilizing materials and an understanding of ASV's not offered by our high school.

When the school year began with incoming potential recruits, our officer team outreached to become a public club amongst the Martin High School student body and delegated our responsibilities between our officers and members, creating three main teams: Engineering, Computer Science, and Marketing.

C. *Design Strategy*

Our approach to design prioritized simplicity, ease of access, and ease of adoption. This meant using materials that not only can be easily changed for edits or additions, but also easily obtained and manufactured without need for costly equipment or orders. For example, testing utilized the special MHS "*TupperBoat"*. (see section III "Testing Strategy" for more information, fig. 4).

This focus on learning the fundamentals of boat and robot design allowed for the team, who were highly inexperienced and lacked knowledge of these advanced systems and concepts, to properly prioritize objectives. The team committed to a dynamic learning process, to easily switch around plans and test out various solutions to the competition theme without restarting from scratch or an unobtainable venture. This iterative design process ensured that a boat would be ready for competition regardless of overambition, unfortunate circumstances, or a lack of resources.

III. DESIGN CREATIVITY

A. *Hardware*

For this challenge, we drew reference from past challengers and based our boat design on a catamaran style boat, two parallel hulls with a center connecting them. We chose this design because of its stability and agility, with the peaks of the hulls allowing for easier and faster movement in the water. This design also reduces the chance of a catastrophic leak that could possibly harm the electronics.

To manufacture the hulls, the engineering team made two cardboard hull molds to later lay fiberglass mat and resin to finish. Our infamous "*TupperBoat*" is placed in between the hulls which are then secured underneath with U-brackets to ensure that the center does not move around. It is crucial that the hardware that is in the container does not move around or has minimum movement to ensure precision with coordinates and calibrations, so we added an acrylic organizer.

For thrusters, we decided to use the basic electronic speed controllers (ESC) thrusters from Blue Robotics controlled by an Arduino UNO. As far as placement, the team decided to place the thrusters on the interior side of each hull, which limits the possibility for debris to clog them up while still allowing them to provide full thrust. A disadvantage of our thruster configuration is that the boat must be a certain weight to keep the thrusters submerged, which limits the top speed of our boat.

Our main computer was previously a board with an ARM processor. Our design revolved around using the board for vision processing and navigation; however, it was not suitable for this task, as it had limited RAM, processing power, and storage, paired with no precompiled binaries. We decided to use a Firefly AOI, which has an x86 processor. While the power consumption is higher, we're able to process more and use existing program binaries.

Our initial design revolved around using an Intel depth camera for buoy detection in small areas in front of the boat, and a Velodyne VLP-16 LiDAR module for wider range vision, but with less precision. The depth camera would be placed on the front end of the boat and the LiDAR would be elevated on the boat for optimal vision. In addition to these external components, the internal hardware essentials are neatly organized in an acrylic organizer (fig. 1) that was laser cut for cleanliness and easy access. This was placed in the container at the center of our boat in between the hulls.

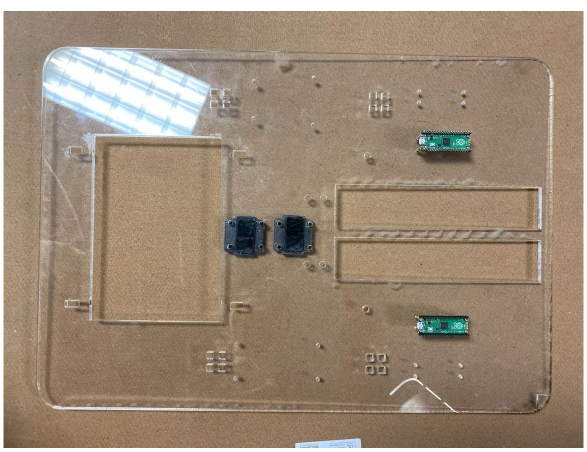

fig. 1 acrylic organizer

B. Software

The two most integral parts of our boat are the Cube Orange and the mini-PC. The mini-PC contains the processing power we need and sends commands and instructions to the Cube Orange using the MavSDK Python library. The majority of the software to detect the buoys – the info is gathered using the Intel Depth Sensor, both color and depth data – and to pass the navigation channel are written in Python on this computer using the PyRealSense library. Additionally, software is written on the Raspberry Pi Picos to monitor the temperature of the batteries. If they get too hot, then it sends a signal to kill the boat.

Instead of the typical GPS sensor, which has a 4.9m precision on location, we used the Real-Time Kinetic sensor (RTK), which gives centimeter-level precision. By attaching the RTK base station to our RoboBoat trailer, it gives our boat accurate positional data to navigate around the Florida course.

The Cube Orange is in charge of aggregating data from the RTK GPS sensor, and additional data from its built-in sensors (heading, speed & acceleration, etc…) to tell the motors where to go and when to stop and such.

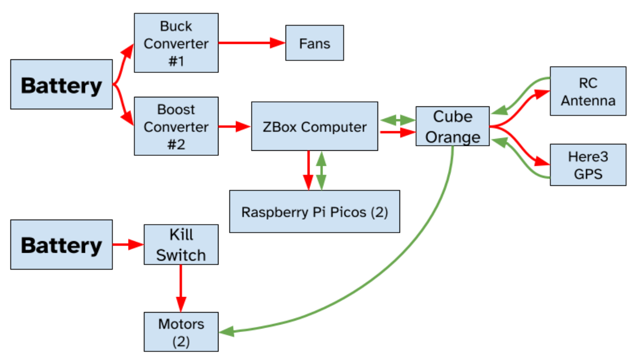

fig. 2 - connection diagram

Using Mission Planner, a ground control station software for the Cube Orange, the team installed the Ardupilot firmware to program the Cube Orange. Functionally, the flying support of Mission Planner can be repurposed for piloting the boat. Our team decided that it would be best to use Python language to program the boat, and used various helper libraries such as MavSDK to tell the boat where and how to navigate.

Our team also created a Python program in order to detect buoys on water using the Tensorflow Lite AI library. In order to train our model we took buoys of varying colors, angles, and distances to virtually feed the model. To get more image data, we also provided the AI with 3D models for more reliable recognition.

fig. 3 - test buoys

IV. TESTING STRATEGY

We purchased buoys in hopes of being able to test the boat physically on water to see if it can pass the starting gates.

Software testing is done through the use of a Gazebo simulation run in a docker container. The boat's design is ported over as a Gazebo model, and from there we're able to program and test to see if the boat can pass the starting gates via software before the physical boat is built.

Testing utilized the special MHS "*TupperBoat"* prototype: the circuits and motors of the boat inside a large tupperware container (fig 4). Since it is both buoyant and resistant to water, this creative idea allowed our programmers to test the boat prior to completion of a fully capable boat and opened up more possibilities in scheduling.

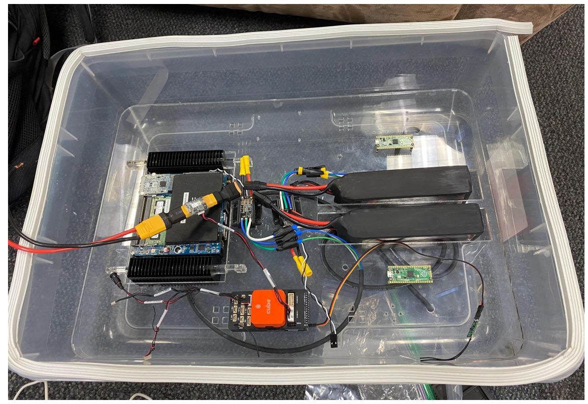

fig. 4 - *TupperBoat*

This allowed for the Engineering and Computer Science teams to act independently from one another, not limited by the other's progress. This also provided a bare minimum, last resort alternative for a competition boat in case of engineering malfunction or time constraints as it would meet our minimum requirements for a boat submission to the official competition.

Testing of the sensors and programming occurred at the local *Lake Arlington* to simulate the conditions of competition. This involved both transporting computers, sensors, the *"Tupperware"*, and buoys frequently over the course of software testing. These tests included Depth Sensors recognizing buoys, troubleshooting directional motor control based upon sensor inputs, properly transferring programming information to the robot, electrical component connectivity tests, etc.

V. ACKNOWLEDGEMENTS

Many thanks to Dr. Chris Dale McMurrough for his technical guidance and support for our ambitious students, as well as the

Engineering Department at the University of Texas at Arlington, who allowed our team to complete research and workshops in their lab spaces. We would also like to extend our gratitude to Jerry Tram who provided our team with various resources regarding the software of our project, constructive comments, and suggestions. In addition, James Hovey, Laura Ebanks, and Jason Forsythe were our teacher sponsors for the 2022-2023 competition year and have continuously supported the team by sacrificing their time supervising our meetings and handling financial aspects of the team. Lastly, General Motors has generously funded the school for this project ultimately making it more feasible.

Appendix A: Component List

Component	Vendor	Model/Type	Specs	Custom/ Purchased	Cost	Year of Purchase
ASV Hull Form/Platform	Home Depot/Own Design	Plastic Container	link	Purchased	9.98	2022
Waterproof Connectors	Lowes	Silicone Sealant	link	Purchased	10.98	2022
Propulsion	BlueRobotics	T200	link	Purchased	200.00	2022
Power System	HobbyKing	Lithium-polymer battery	link	Purchased	108.97	2022
Motor Controls	BlueRobotics	Basic ESC	link	Purchased	36.00	2022
CPU	Zotac	ZBOX CI660 nano	link	Purchased	945.00	2023
Teleoperation	Amazon	Taranis Q X7	link	Purchased	137.99	2022
Compass	CubePilot	Here3	link	Purchased	290.00	2023
Inertial Measurement Unit (IMU	CubePilot	Here3	link	Purchased	290.00	2023
Camera(s)	Intel	Intel RealSense Depth D435	link	Purchased	314.00	2022
Doppler Velocity Logger (DVL)	N/A	N/A	N/A	N/A	N/A	N/A
Hydrophones	N/A	N/A	N/A	N/A	N/A	N/A
Algorithms	N/A	N/A	N/A	N/A	N/A	N/A
Vision	Intel	Intel RealSense Depth D435	link	Purchased	314.00	2022
Localization and Mapping	CubePilot	Cube Orange	link	Purchased	485.00	2022

PyRealsense	Intel		link	N/A	N/A	2023
MavSDK	DroneCode		link	N/A	N/A	2023
Gazebo Simulator	Open Robotics	N/A	link	N/A	N/A	2023
ROS	Open Robotics	N/A	link	N/A	N/A	2022
Docker	Docker	N/A	link	N/A	N/A	2022
MicroPython	Raspberry	N/A	link	N/A	N/A	2023
QGroundControl	DroneCode	N/A	link	N/A	N/A	2023
Mission Planner	Ardupilot	N/A	link	N/A	N/A	2022