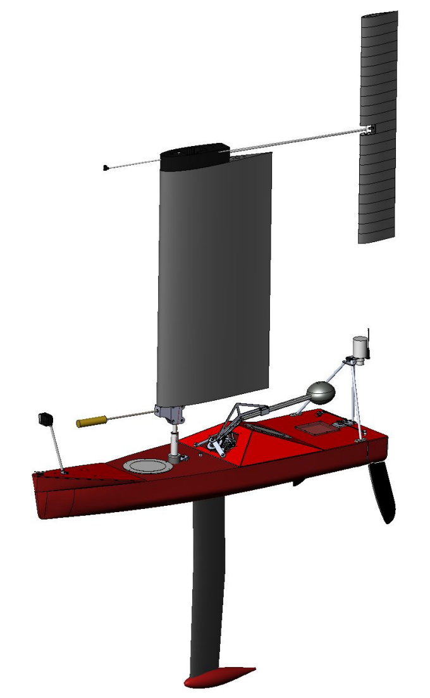
Sailbot

Connor Burri, Nickolas Eusman, Nathan Jackson, Michael Laks,
Colin Scholler, Abhijay Thammana, Luke Zebrowski
Advisors: Professor William Michalson (RBE), Professor Kenneth Stafford (RBE)



Abstract

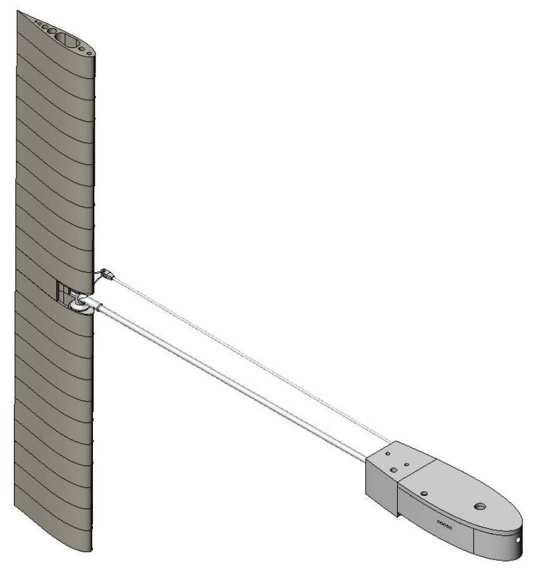
The objective of this project is to improve upon prior implementations of the robotic sailboat known as “The Great Awake”. By further developing the mechanical, electrical, and software systems put in place by prior teams, this team plans to increase autonomy and reliability throughout the entirety of the Sailbot. Our main focus for this iteration of the project is to implement a robust, fully autonomous sailing apparatus, and enable future teams to more easily continue improving the robot.

Objectives

1. Allow for a human operator to use full RC control to navigate a closed course.
2. Autonomously navigate between two points placed a maximum of 50 meters apart with an accuracy of within 3 meters of the target point.
3. Autonomously round buoys.
4. Continuously sail a predetermined closed pattern for a minimum of six hours.
5. Sail a search pattern within a predetermined 50 meter diameter circle in order to locate a randomly placed buoy, then signal the GPS coordinates of the buoy to shore.

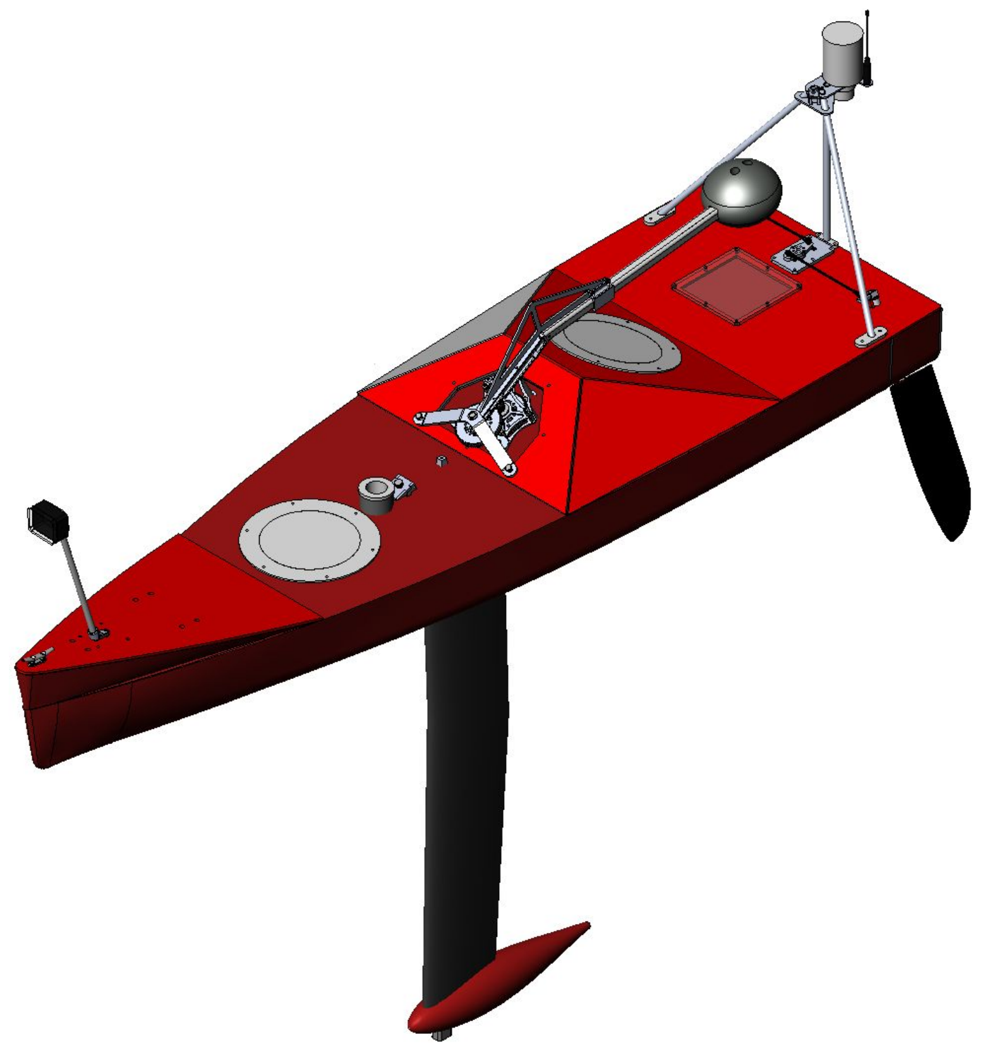
Trim Tab

The trim tab sits midway up the mainsail, and has the following features: a servo to control its angle, a relative wind indicator for finding max lift, a MKR Wifi 1010 microcontroller to control the systems, and a splash proof enclosure. The trim tab is responsible for generating the lift necessary to propel the boat.



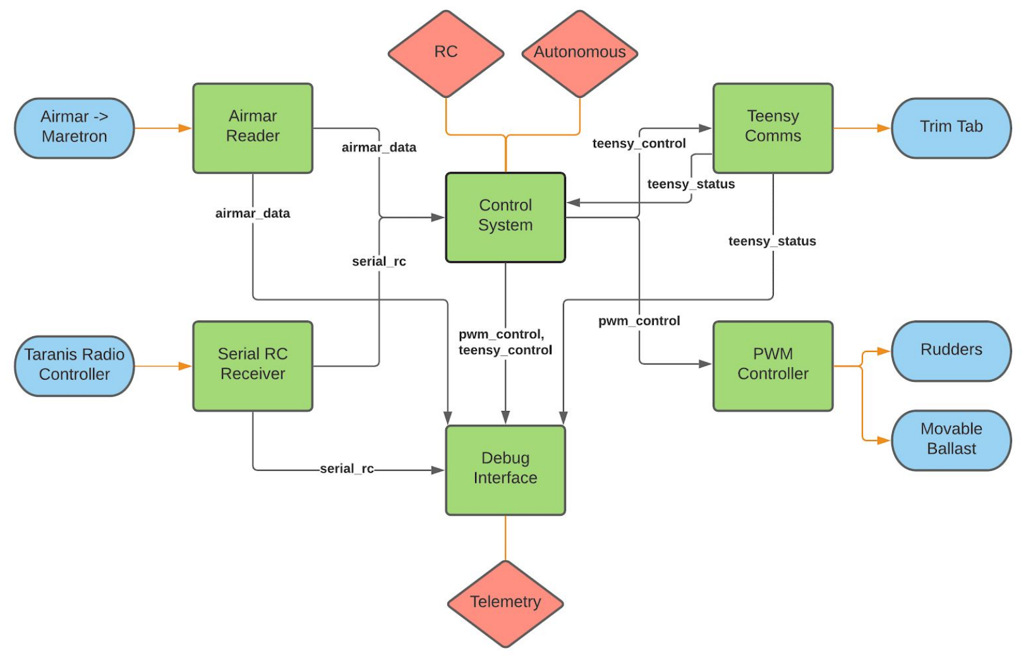
Hull

The hull consists of multiple mechanical components including the keel, ballast, rudders, and the hull itself. The onboard computer was switched to a Jetson Nano this iteration. This board handles all processing and control for the hull systems and communicates with the trim-tab Nano through a WiFi connection. 2 LiPo batteries were used for power as they would be light weight while still providing enough power to endure a 6 hour run.



Software

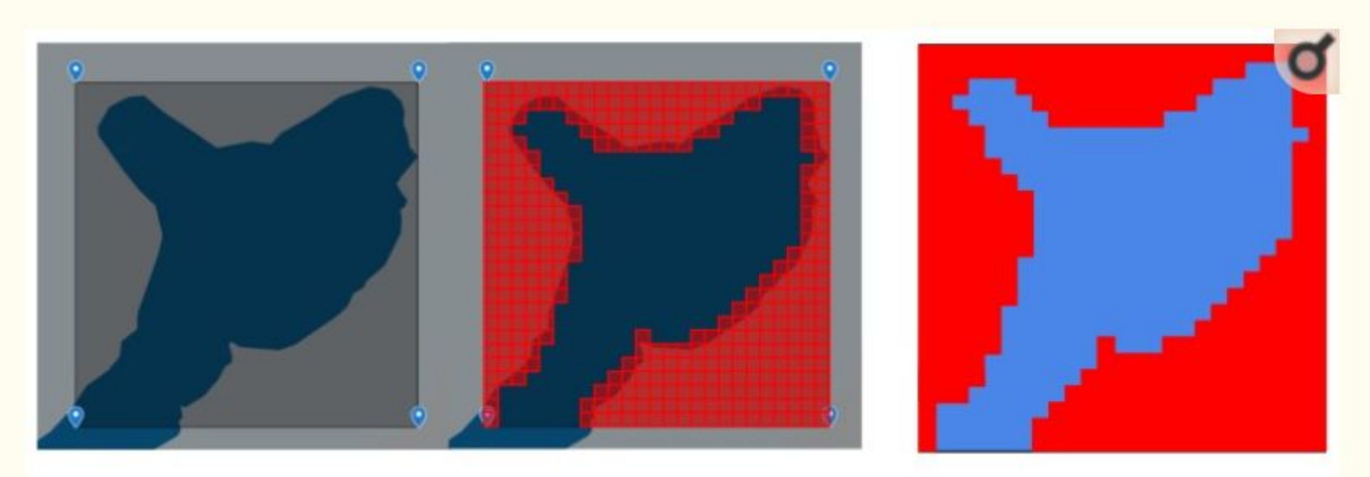
The software for Sailbot is written in ROS2 and consists of six nodes: airmar_reader, pwm_controller, serial_rc_receiver, control_system, teensy_comms, and debug_interface.



The software of the boat consists of three modes of operation:
1. Full RC
2. Semi-autonomous (ballast and trim tab are automated)
3. Fully autonomous.

Algorithms

The sailing algorithm is broken up into three parts:
- A high level pathing algorithm
- A low level point to point (P2P) travel algorithm
- A ballast control algorithm



High Level Pathing:
Divides the sailing area into a grid of squares with the edges of each node representing the possible directions of sail. An A* algorithm is used to pathfind from the given GPS coordinates to the next to create a path around the course.

P2P:
Takes in inputs about the boat and wind status to set the trim tab and rudders appropriately. If the wind changes by more than 20 degrees for a period of 20 seconds, the A* algorithm will recalculate before continuing P2P

Ballast Control:
Keeps the roll angle at 20 degrees leeward to minimize hull drag while maximizing rudder authority. The algorithm continually checks the boat’s current roll and adjusts the ballast to maintain ideal trim within a margin of ±5°.

Conclusion

The goal of having full RC control was met and exceeded with the operator being able to use fully manual and semi-autonomous control. While the algorithm for autonomous navigation is completed, it lacks full testing. Computer vision was also not implemented, making autonomously rounding buoys and performing search and rescue not currently feasible. For future work on sailboat, the main goal should be implementing computer vision using the available ZED 2 camera. Refinement of the algorithms we programmed will improve speed and reliability of the boat. And new connection protocols for the trim tab should be explored, as the current wireless connection suffers from latency.