

University of West Florida Water Team

Journal Papers

Members:

Colby Cohron
Daniel Gray
Jonathan Herrero
Erik Estess
Perry Journey

Abstract

The mission of the water team is to create an autonomous water vessel able to perform an array of tasks given by the Association for Unmanned Vehicle Systems International (AUSVI) for International RoboBoat Competition. As these events change yearly the team must adapt and come up with new solutions for each task.

There are many aspects of the boat the team has had to plan and design. These parts mainly consist of the hull, electronics, and programming. Each of these sections is crucial for the boat to move and function autonomously. Through extensive testing and troubleshooting we got the final result for competition. And to plan and design these aspects members of the team have learned how to coordinate and cooperate to reach a long term goal.

Hull



Figure 1.

The hull is the crucial part to any water vessel. In the case of our team we aimed to design a hull that could be built upon but also stayed within the constraints of the competition rules. The hull of our boat was actually used as the hood of a Yamaha jet ski, as seen in figure 1 above. The hull was then modified by constructing a mount for our main 15V trolling motor. Side mounts were then fabricated to hold the small bilge pumps used to steer the boat.

In the case of our hull there was a lot of un-needed empty space in the bottom, so the team decided to fill this section with expandable foam. This foam helped to add stability to the boat and allowed the team to make a large platform to secure electronics too.

Electronics

The electronics systems of our boat have many different parts. First off we have 4 Lithium-Polymer batteries powering the whole boat. With these batteries the boat is capable of running for 1.5 hours at full capability. These batteries supply the needed power to the boat without compromising weight.



Figure 2.

The next aspect of the electronics is the computer. We use a FitPC3, it has an AMD APU with 4 GBs of RAM and a 500 GB hard drive. This computer allows us to have a lot of computing in a compact space. The computer's job is to analyze the different colored buoys using our 1080P camera mounted to the front of the boat. The computer



Figure 3.

The computer talks to our next key component the Arduino Mega. The Arduino takes directional inputs from the computer and RC inputs if needed. We decided to go with the Arduino Mega over the Arduino Uno because it has more IO for our use. The Arduino communicates with the motor controllers telling them how to react.

Low pass filter for using PWM in Analog Mode

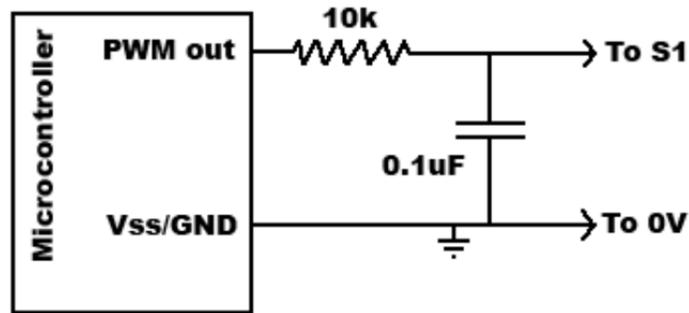


Figure 4.

The outputs of the Arduino are then sent into a low-pass filter that transforms the output of the Arduino so it can be used by the motor controllers. A diagram of the low pass filter is shown above. This filter transforms our PWM signal into a triangular waveform the motor controllers use to determine left or right movement.

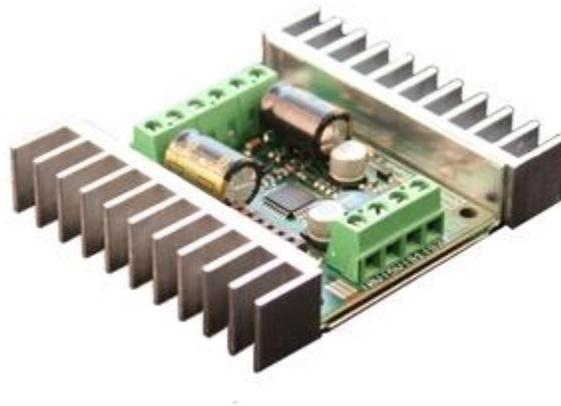


Figure 5.

The next part of the electronics is the motor controllers. We use 2 Sabertooth 2x12's to control our 3 motors on the boat. A Sabertooth 2x12, shown in figure 5 above, motor controller can handle up to one 25A motor or two 12A motors on the same controller. In our configuration we have a single Sabertooth controlling our main 15V/15A motor for the boat. We then have the other Sabertooth controlling the two steering motors.

corresponds to 'Turn Left', and 'R' corresponds to 'Turn Right'), and are displayed as words in the terminal. This assists the user in observing the commands that are being sent to the microcontroller. A direction command is sent only if red and green are detected. If either color is not detected in the frame, then 'S' (Stop) is sent to cut all motor power. This entire code sequence can be adapted to any buoy color scheme, in our case we will be using this method to navigate through the course of black and yellow buoys. And to navigate through the green and red starting gates.

The microcontroller is configured to output signals to the motor controllers. Since each direction command sent from the computer is unique, the microcontroller is programmed to interpret every instruction and act accordingly. Aside from the serial connection, the microcontroller has two additional inputs: two separate channel inputs from an RC receiver. In the event of an emergency, the microcontroller is configured to pass RC commands to the motor controllers. When the program initializes, the microcontroller acts based on the instructions sent from the computer. Full RC control can be given to the operator at any time by turning on the transmitter.