



Cedarville University RoboBoat 2013



**Jimmy Smyly (Team Leader), Kyle Girtz, Jonathan Power,
Michael Comparetto, Josiah Franklin, David Moeller
Team Advisor: Dr. Timothy Tuinstra**

ABSTRACT

The purpose of the Cedarville RoboBoat team was to build on the successes of last year's team in preparation for the 6th Annual International AUVSI RoboBoat Competition. We reused the platform, thrusters, control hardware, and camera from last year's boat to maximize the boat's reliability and functionality. We improved the boat's vision and GPS navigation capabilities, wireless boat-to-shore network, and power systems. The core of the boat's intelligence is a single laptop running MATLAB software that segments the vision from a webcam. The design features the ability to autonomously navigate a course of color-coded buoys and complete various shore-based challenges. Among these challenges are the abilities to send GPS coordinates of a location back to a home network and retrieve an object from shore with a separate autonomous subsystem.

1. PLATFORM

1.1 pontoons

Our platform performed well last year at competition and we continue to use it successfully. The pontoon style platform consists of two Echo Fisher pontoons from Venture Outdoors and offers maximum stability on the water. These pontoons are composed of linear, low-density polyethylene (LLDP), which is much lighter than PVC. Each pontoon is 48” long by 12” wide by 10” high and weighs only 11 pounds. Originally the pontoons were designed to be a fishing kick-boat with a seat between the pontoons. This seat caused the platform to exceed the competition width limit. By removing a section from the middle of the seat and plastic welding the seat back together, the boat was made to fit within the three-foot limit.

1.2 Propulsion

The propulsion system consists of the same two Seabotix BTD-150 thrusters used last year. These thrusters were chosen because they are extremely small (6.927”x3.72”x3.673”), lightweight (1.58 *lbs* each), and easy to interface. At the maximum voltage, the thrusters can each produce 4.07 *lbs* of thrust per motor. The thrusters are brushed DC motors enclosed in a watertight housing. Each thruster is mounted directly underneath the center of each pontoon. This design makes zero-radius turns possible using differential steering, a key element navigating narrow spaces.

1.3 Electronics Enclosure

Our team also chose to reuse the electronics enclosure from last year. The Nanuk 945 protective case was chosen because it is relatively light and the interior dimensions are large enough for all of our electronics. However, the Nanuk 945 does not have any output connectors through which wires could be routed. Therefore, simple 90-degree elbow plastic conduits (with threaded ends) were installed. The diameter of conduits is just wide enough to fit USB plugs. Drilling into the box and combining the threaded ends with O-rings created a water-resistant connection. It would be very difficult for water to get into the enclosure, as it would have to go directly up a small elbow opening that is not directly over the water.

2. ELECTRONICS

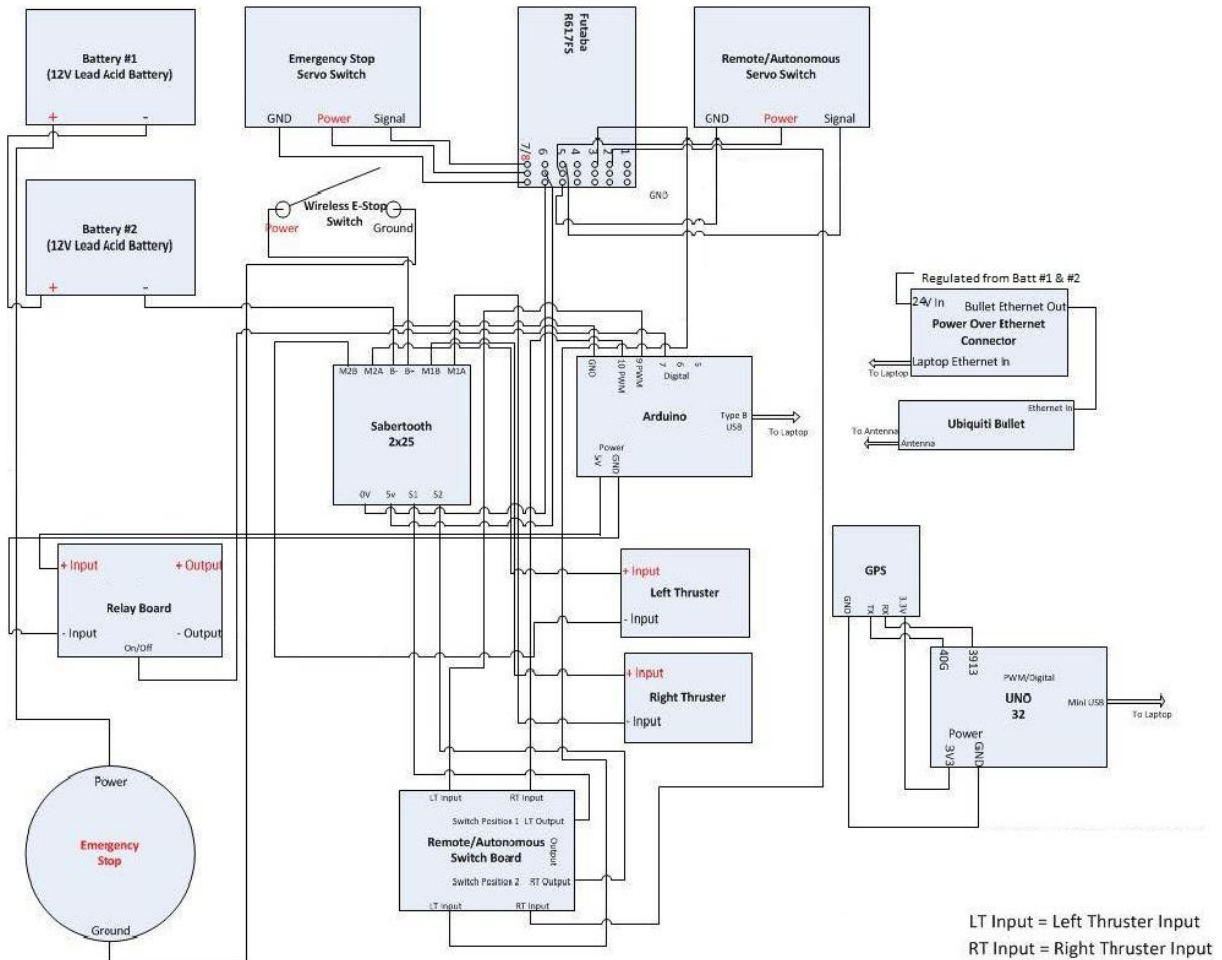


Figure 1. Electronics System Diagram

2.1 Global Positioning System (GPS)

Our GPS unit is an LS20031 receiver. This receiver was chosen last year. It is a Wide Area Augmentation System (WAAS) receiver with an accuracy of ± 3 m (9.84 ft). This unit was chosen for its simplicity, in addition to providing sufficient precision. This year we decided to add a GPS to the boat. With two units we hope to be able to use an averaging algorithm to increase our precision.

2.2 Compass

The Devantech CMPS03 Magnetic Compass Module has 0.1 degree of resolution and 3-4 degrees of accuracy. The headings are accessed via I2C; the device is connected to the laptop by a USB port through an I2C to USB module. The Devantech compass' 19200 baud rate

allows reception of data from the compass in real time, making it useful for navigation. The compass is a key part of our design; all of our navigation is based off the boat's ability to follow a given compass heading.

2.3 Webcams

The Microsoft Lifecam Studio Webcam was chosen because it is very easy to interface with Matlab through USB. This webcam also has many RGB resolution options. We determined a resolution of 160x120 was sufficient to achieve the needed resolution without sacrificing processing time. The white balance, exposure, hue, and saturation can all be manually set using one line commands in Matlab which is helpful in varying lighting conditions. Our camera is installed tilted down to reduce visual noise on shore and has a polarizer to reduce reflections from the water's surface.

2.4 Laptop (Matlab)

The Dell Inspiron N4030 laptop is the main processor for our system. The laptop runs Matlab and performs all of the vision and data processing. Programming in Matlab, with its built-in Image Processing Toolbox, was much simpler than using a less abstract language like C. In Matlab, we are able to interface with our microcontroller, compass, GPS, webcam, and network antenna simultaneously while concurrently doing all image processing and navigation computations in a single program.

2.5 Microcontroller

The laptop provides data to control the thrusters. However, intermediate electronics are required. The Arduino microcontroller converts numerical values to pulse width modulation (PWM) signals for the motor controller. An Arduino was chosen because it is very common and has open source software. Forums and help sections make programming the microcontroller very straightforward. The Arduino also hooks directly to a virtual COM port on the laptop via USB.

2.6 Motor Controller

The Sabertooth 2x25 motor controller has two output channels capable of sourcing up to 25A. Another key feature of this choice is its ability to drive both thrusters independently, allowing for differential steering. This motor controller met the voltage and current specifications, contains built-in overcurrent and thermal protection, and is familiar to our engineering faculty. To switch between autonomous and RC mode a motor controller was

needed that could also work with the remote receiver. The Sabertooth has a 5V power supply capable of powering the remote control receiver. A servo throws a switch controlled by a channel on the remote control so the input to the motor controller can alternate between autonomous and remote modes.

2.7 Remote Control

Last year, a Futaba 7C 2.4 GHz remote control was chosen. We continue to use this remote due to its spread-spectrum technology. This feature makes the remote very reliable. Even in highly networked areas, like the competition conditions, we are able to connect easily and maintain a solid connection.

3. TENNIS BALL RETRIEVER (CATCH THE BALL CHALLENGE)

3.1 Rover

Our rover for the autonomous object retrieval challenge has been completely redesigned since last year's competition. The new rover, shown in Figure 2, features a redesigned printed circuit board, chassis and object retrieval mechanism. An improvement over last year's rover is the new tracked chassis; this provides much better traction on rough surfaces like the dock. The new printed circuit board boasts a variety of improvements, including the implementation of two-way communication with an instance of Matlab on the boat laptop via RS232 transceivers. Based on the preliminary rules for the 2013 competition, the retrieval mechanism is designed for retrieving a tennis ball and features Velcro pads to provide extra adhesion, a grabber servo to prevent the ball from falling out as it is being lifted, a high-torque servo to lift the ball, and a digital sensor to verify successful retrieval. After retrieval, the ball is lifted on top of the rover to prevent it from rolling out. The brain of the rover control circuitry is still a PIC18F45K22 microcontroller, which features more than enough analog, digital, and pulse width modulated I/O for our needs.

Our deployment system is new as well. The crane we attempted to use last year wasn't powerful enough to deploy the new, heavier rover, so a ramp was constructed to get the rover onto the dock, as shown in Figure 2. After the boat makes contact with the dock, it will signal the rover via the RS232 transceiver to start. The rover will then push the ramp down and continue onto the dock. If the ramp deploys and the rover does not detect solid ground on the other side, it will backup, stop, and signal the boat to abort the challenge attempt.

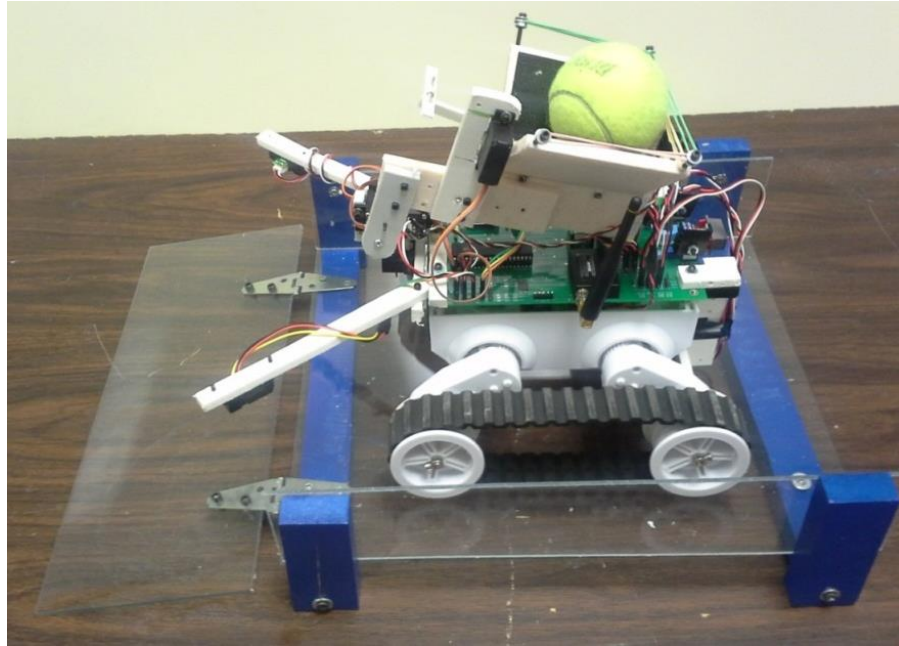


Figure 2. Tennis Ball Retriever and Deployment Ramp

3.2 Rover Search Algorithm

The redesigned rover uses an array of distance sensors, shown in Figure 3, to find the object and avoid driving off the dock. As in last year's rover design, two distance sensors mounted at the ends of "antennae" are used to detect the edge of the dock. If the rover nears an edge, it will back up and rotate before continuing its search.

To find the object, the rover uses three distance sensors mounted beneath the chassis. These sensors are also used to home in on the ball so the rover can position itself for retrieval. An additional sensor mounted above these three sensors determines if an object, such as the boat or dock poles, is too tall to be the ball in which case the rover will ignore it. This search algorithm has been demonstrated successfully.

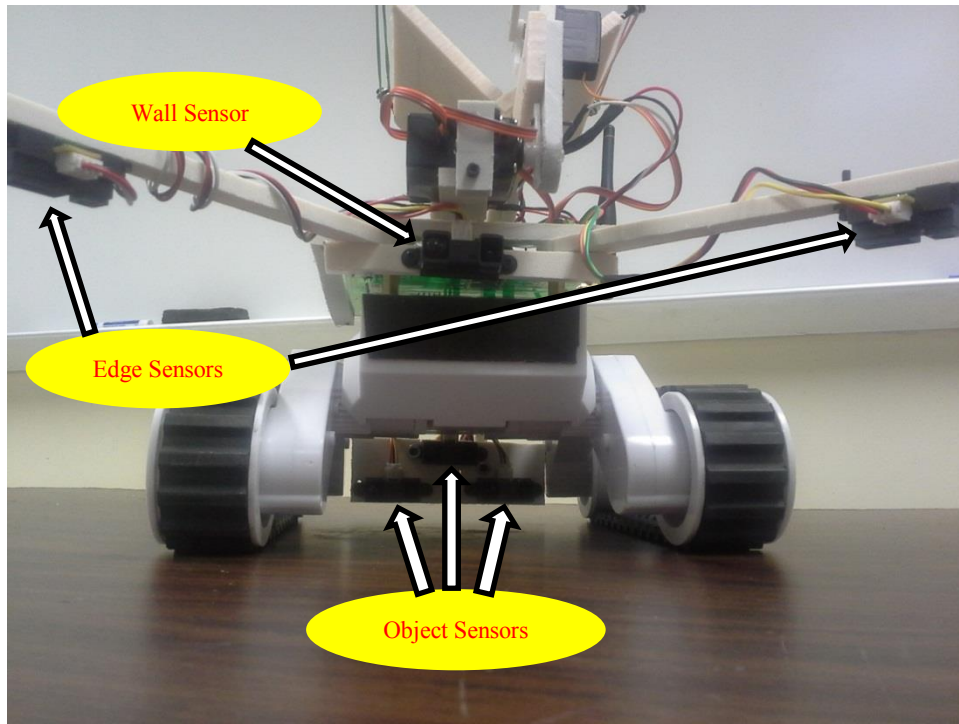


Figure 3. Distance Sensors

4. VISION

Because our boat sees its surroundings with only a single webcam, an algorithm is needed that can effectively and consistently identify red, green, yellow, and blue buoys in an image. Last year, video frames were processed in the HSV (hue, saturation, and value) color space. In this color space only two dimensions were needed to create thresholds between each color. Unfortunately, this approach made it very difficult to be consistent in changing light conditions. This year we took another look at the RGB space because we believed that a more sophisticated algorithm might work in this space better than the HSV space. It is also much faster to process because we do not have to convert between spaces. Instead of looking for simple thresholds on each dimension, we used a relative approach that segmented colors based on the differences between the dimensions. Effectively, this allowed us to have angled planes in the RGB space rather than planes only parallel to the axes. This idea was expanded into RGB planes of any kind. Our current segmentation algorithm allows us to consistently identify red, green, yellow, blue, white, and purple objects in bright or shady conditions. These objects are then filtered into large blobs and located within the image. The coordinates of the centroid of each identified blob of each color is passed to our navigation algorithm.

5. NAVIGATION

5.1 PD Controller

Because our thrusters generate different output thrusts given the same input voltage and because wind and other disturbances can throw the boat off course, a controller was designed to compensate for these problems. This controller will allow the boat to follow a given heading regardless of physical issues by giving a feedback from the compass. To create this controller, the left thruster is set to a specified “forward” value while the right thruster is set to the same value plus or minus proportional and derivative quantities. A proportional controller adds some multiple of the current versus desired heading error. However, a simple proportional controller causes the boat to oscillate considerably, taking longer than a minute to settle out. To remove the oscillations, a PD controller is implemented. It adds an approximation of the derivative of the error multiplied by a constant to the thruster value as well as the proportional value discussed earlier. Using the PD controller, an extremely reliable compass following algorithm has been achieved.

5.2 Speed Gates

To navigate the speed gates at the beginning of the course, heading information from the compass is used to navigate the speed gates in minimal time without error. A Matlab function was developed that causes the boat to go in the direction of a given compass heading using the PD controller previously discussed. Using this function, the boat is simply pointed in the correct direction initially and follows that heading for a given amount of time before entering channel navigation mode. If this occurs too early and the boat detects the speed gates, the channel navigation algorithm will navigate the remaining speed gates appropriately.

5.3 Buoy Channel

To navigate the channel of red, green, and yellow buoys our webcam and compass are used in tandem. The segmented buoy locations (discussed above) are used to make logical decisions to navigate the channel. The algorithm determines the most relevant “red” buoy (a red buoy, yellow buoy on the left, the left edge of the frame, or a green buoy on the return trip) and “green” buoy (a green buoy, yellow buoy on the right, the right edge of the frame, or a red buoy on the return trip) and attempts to drive the boat towards the midpoint of these

two objects using a proportional correction to our desired heading. This heading is then followed with the PD controller discussed above.

5.4 Challenge Navigation

In order to navigate to challenges at the end of the channel, a sequence of stored GPS waypoints is used. These waypoints will be the location of each challenge given to us or determined before competition. Based on the boat's current location and next desired waypoint, the heading and distance can be calculated and passed to the PD controller. The boat will use the GPS unit's margin of error (3 m) to determine if it has reached the destination. Upon arrival, it will call the appropriate challenge function. Once completed, the boat will move onto the next preset waypoint and complete another challenge. After all desired challenges are completed the boat will navigate back to the waypoint representing the blue buoy to return through the channel.

6. CONCLUSION

The last couple of years that our team has participated in the competition we have had some problems with organization and robustness. This year, one of our priorities was to make all of our subsystems more robust, accurate, and reliable, particularly the software. To this end we believe that our vision and hardware will perform much better than previously. This will leave us free to focus on GPS and channel navigation logic. We would also like to attempt some of the challenges. We hope to be able to finish the channel, navigate to at least one challenge, possibly attempt it, find the channel again, and try to return to the dock.