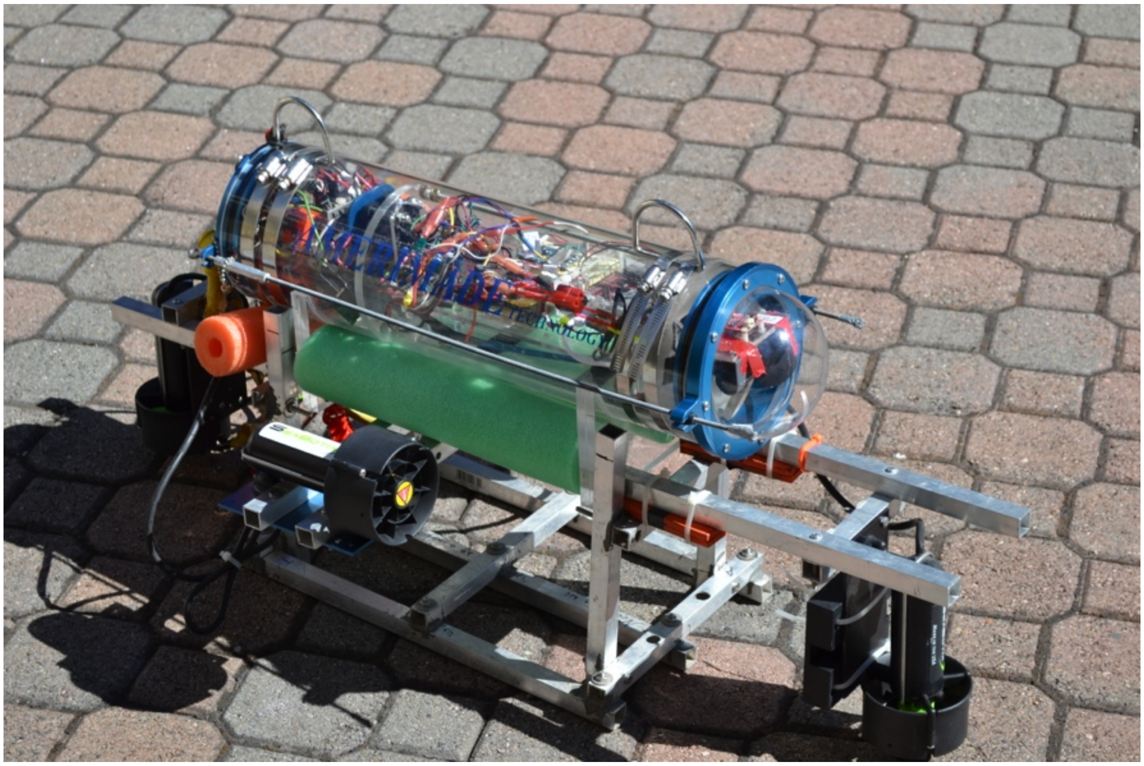
Amador Valley High School

Barracuda Mark XI (2012)



Vignesh Kalidas, Kevin Chen, Pierson Conners, Ashutosh Shiledar, Enrica Beltran, Jeff Chen, Siddhartha Desai, Jack Herbert, Sridivya Kondapalli, Alex Korshak, Brandon Lopez, Yosef Mirsky, Adit Sinha, Swayam Suri, Glynn Williams, Angus Wu

Abstract

The Amador Valley High School Robotics Team has constructed an autonomous underwater vehicle (AUV) named Barracuda Mark XI for the 15th annual AUVSI RoboSub competition.

The internal components of Barracuda are housed within a single 6-inch (15-cm) diameter acrylic tube. The vehicle is propelled by two laterally mounted SeaBotix thrusters controlling velocity and heading, and two vertically mounted SeaBotix thrusters controlling pitch and depth. A pressure sensor, inertial measurement unit (IMU), two cameras, and a hydrophone array provide navigation and orientation data to the main computer. The main computer is a BeagleBoard, which autonomously controls the vehicle and the peripheral computers. The main computer focuses on high-level mission control and image-processing tasks, while an mbed, a flexible prototyping board for microcontrollers, provides low-level control over the thrusters and acts as a liaison with the various sensors.

This year, the team focused on pioneering new peripherals and streamlining the Barracuda's internals, while educating its many new members on the side. The mechanics subdivision spearheaded many peripheral development projects, including a torpedo launcher and dropper. Electrical streamlined the Barracuda's electrical subsystems, and introduced a replacement for the control board: the mbed. Software wrote the firmware for the mbed, and replaced AVI, the old main-computer software, with a new, more flexible program, Extensible Vehicular Automaton (EVA).

The Team

The Amador Valley High School Robotics Team, AVBotz, was founded in 1999 by a group of precocious students who were curious about robotics. Since then, the team has grown and evolved: the current team has over twenty-five contributing members, and has a large population of talented underclassmen. Apart from striving to build the best submarine at RoboSub, another one of the team's goals is to introduce the different facets of engineering to students in an engaging way. The team prioritizes teaching new members about the submarine's systems. It also participates in community outreach by guiding robotics club members at the local Harvest Park Middle School as they build their autonomous LEGO robots.

The RoboSub 2012 Mission ("Ides of TRANSDEC")

Each year's RoboSub mission is designed to simulate tasks that autonomous underwater vehicles face in the real world, such as motor control, computer vision, and passive sonar. The San Diego, CA TRANSDEC facility, which contains a naval testing pool designed to be anechoic, has hosted the competition in recent years.

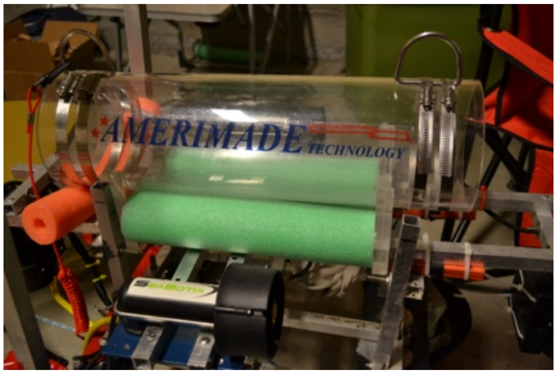
Each vehicle is allotted fifteen minutes to complete the mission. Points are awarded upon successful completion of tasks. In addition, bonus points are awarded based on unused time and weight below the weight limit.

The tasks in this year's mission are as follows. First, the submarine must pass through an underwater validation gate in order to demonstrate autonomy. The submarine may then proceed to a series of tasks, which it can locate by following an orange path on the floor of the pool with computer-vision algorithms. Then, the AUV must recognize and strike two specific colored buoys out of three. Next, the AUV must pass above a green PVC structure. The AUV then must drop two markers into two designated bins out of four. Each bin is decorated with a different image, so the AUV must recognize the correct shape to complete this task. Afterwards, the submarine must approach a 2×2 array of four windows — one side colored red, the other side blue — and shoot torpedoes through a window of the specified color. Next, the AUV must complete an autonomous-manipulation task that involves picking up two red cylinders and then releasing them. Finally, the AUV must navigate to an acoustic pinger to secure a PVC-pipe structure near the pinger, and surface within a octagon floating on the surface.

Additional mission details are available on the RoboSub website, <http://robosub.org/>.

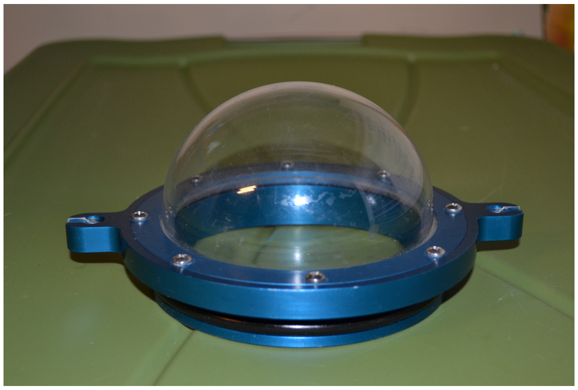
Hardware

The Barracuda Mark XI consists of a hull, a frame, engines, and several other modular components. All external components are attached by screws to the frame, which is made of two aluminum square rods running parallel to the tube. The ability to attach components directly onto the frame with any size of screw gives the team greater flexibility in rapidly attaching new components without relying on separate attachment pieces, like ring clamps.



Tube and Frame

The hull of the Barracuda is a cast acrylic tube. The tube has a 6-inch (15-cm) outer diameter, and is $\frac{3}{16}$ inches (0.5 cm) thick and 20 inches (51 cm) long. The frame is comprised of two parallel square aluminum rods that are $\frac{1}{16}$ inches (0.16 cm) thick and are $\frac{3}{4}$ inches (2 cm) wide. These run along to the bottom of the tube, fixed to the hull through a series of acrylic cradles cemented to the tube. The cradles have holes through which the frame rods are attached via a series of pins. Two metal D-rings are attached to the top of the tube for deployment and weight measurement. Both ends of the tube have anodized aluminum end caps, which are secured to each other. This design alleviates the strain on the tube to extend the life of the tube. The end caps are equipped with compression O-rings that create a watertight seal.

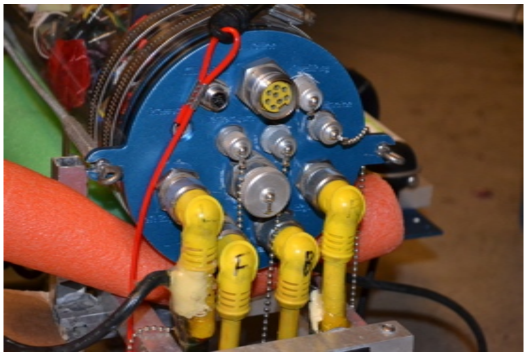


Dome (Front) End Cap

The front end cap contains a clear cast-acrylic dome in the shape of a hemisphere. We chose the acrylic material for its transparency and its durability: the exceptional visibility it provides is necessary for Barracuda's cameras to take clear photos, and its dome shape gives the cameras an increased field of vision. The flange of the dome is fixed between the front end cap and another metal ring. This assembly is secured by eight screws and made watertight by an O-ring.

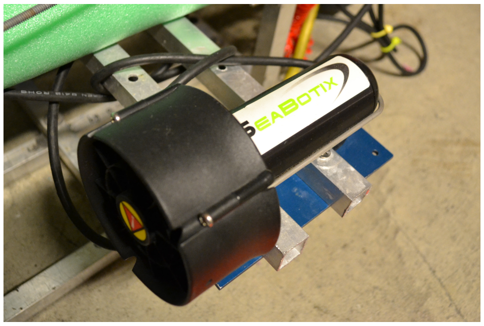
Connector (Rear) End Cap

All through-hull connectors are made on the rear end cap and use Brad Harrison Waterproof connectors. These connectors are IP67 rated and can handle up to 1000 PSI, which is the equivalent of about 231 feet (70 m) of water. The ability to remove and reattach these connectors rapidly without disrupting the watertight seal of the hull adds to the modularity of Barracuda's design.



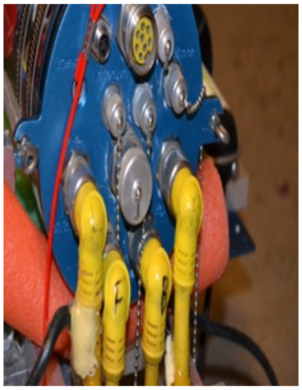
Cooling Systems

Care has been taken to select components that are power efficient: Barracuda requires no cooling systems for a majority of its components. However, the Novak Super Rooster motor controllers generate significant heat due to the high currents they switch. Therefore, they are mounted on the thermally conductive connector end cap, which doubles as the heatsink in a passive-cooling system by dissipating heat into the water.



Thrusters

Seabotix BTD150 thrusters with built-in shrouds provide Barracuda with horizontal and vertical thrust. Power consumption is quite low: each thruster draws only 3.3 amps at full power. This limits the amount of magnetic interference experienced by the compass and increases Barracuda's battery life. The vertical thrusters control depth and pitch, and are mounted on aluminum bars affixed perpendicularly to the frame. The horizontal thrusters are mounted on an aluminum plate on the horizontal engine bar running perpendicular to the tube. They control the heading and velocity of Barracuda, and allow both forward and backward motion.

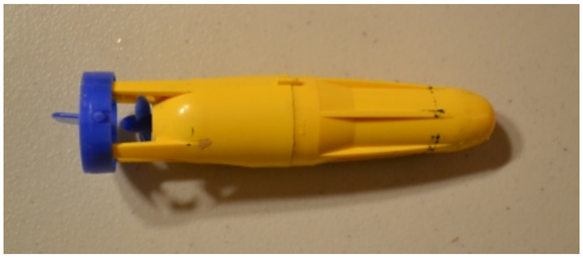


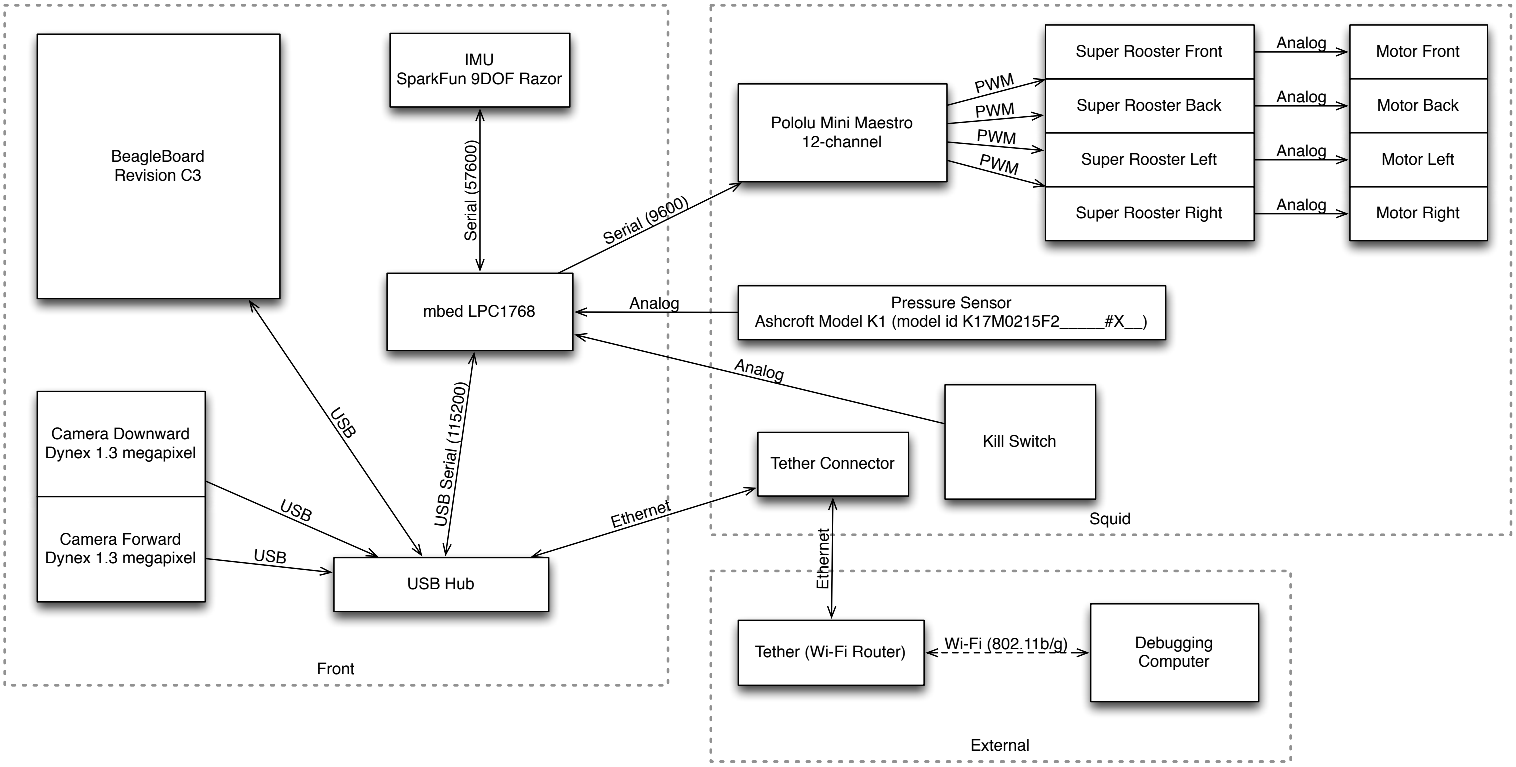
Markers and Dropper

The markers are repurposed spent 12-g (0.4-ounce) carbon dioxide canisters with "AV" engraved onto them. They stay in a PVC tube mounted on the side of the horizontal engine bar. A small waterproof servo motor with an aluminum arm is located to the side of this tube; its blade prevents the markers from falling out of the tube. The servo will turn when the correct location is detected, releasing both of the markers.

Torpedoes and Torpedo Launcher

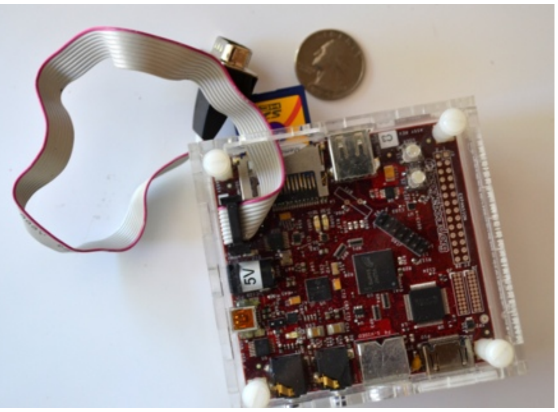
The torpedoes are waterproof torpedo motors which consist of a plastic shell, battery, and a small electric motor sealed with a gasket and O-ring system. Around this is a low-density polymer (not shown) that brings the density of the torpedo near that of water. This provides an approximately flat trajectory. These torpedoes are turned on and placed in a 1.5-inch (3.8-cm) wide acrylic tube. A plate attached to a servo motor prevents the activated torpedoes from leaving the tube until they are launched by turning the servo motor.





Electrical Subsystems

The main tube is partitioned horizontally into two levels by an acrylic rackboard, which extends from the dome to the rear end cap. The BeagleBoard, USB hub, control board, hydrophone board, and power supply reside on the upper level, while the batteries are located in the lower level. Elcon Rack Connectors allow the rackboard and rear end cap to attach securely, enabling power and data transfer to the end cap and to any external peripherals or debugging computers. This design also allows team members to remove the rackboard from the submarine and work on it independently of the other components through the use of a cradle, which is a U-shaped acrylic tube that provides structural support for the rackboard's contents while it is being serviced.



Main Computer (BeagleBoard)

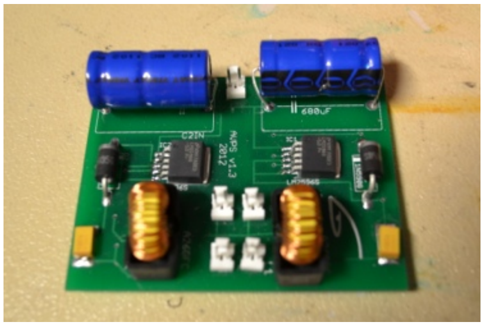
The sub's main computer is a BeagleBoard Revision C3, a low-cost, single-board computer. The BeagleBoard runs Ångström Linux, a stripped-down distribution of Linux designed to be run on embedded devices. The CPU of the BeagleBoard is a Texas Instruments OMAP3530 processor running at 600 MHz, and the same area of the board also contains 128 MB of LPDDR RAM in a package-on-package configuration.

An advantage of the BeagleBoard is that it is efficient: it provides smartphone-level performance while consuming less than 5 watts of power and dissipating negligible amounts of heat. This also allows it to run off of the main batteries without significantly decreasing Barracuda's battery life.

The boot drive for the BeagleBoard, which contains the operating system and the mission code, is located on a 4-GB SD card. The remainder of the input and output is conducted through a single USB port, connected to a 4-port USB hub that also contains an Ethernet port. Debugging and troubleshooting can be easily done over the network through remote terminal (SSH).

Batteries

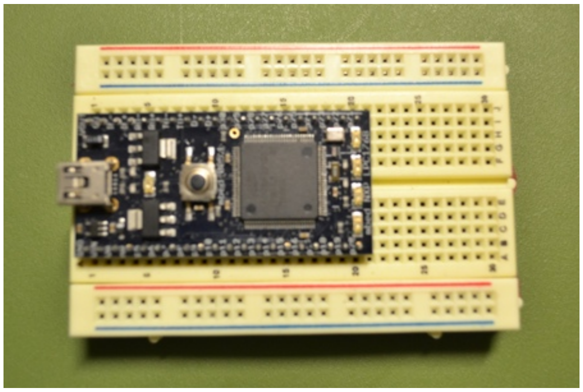
The batteries, situated in the lower level of the rackboard, supply power to Barracuda's thrusters and internal electronics. Barracuda uses two 4-cell, 14.8-volt lithium-polymer battery packs with a combined capacity of about 12 Ah, a value that varies slightly depending on which set of cells is being used. Because of the low power consumption of its components, Barracuda can achieve a run time of 2 hours while in full operation, but it typically achieves a much longer time because the motors do not run continuously. Anderson Power Pole connectors make the battery packs hot swappable, so that the main computer and its peripherals do not need to be powered down when replacing depleted batteries. The battery packs contain protection circuitry which prevents over-discharging and over-charging of the packs.



Power Supply

While the thrusters run on 14.8 volts, the rest of Barracuda's electrical systems require a 5-volt source to function properly. To lower the battery's high voltage to one that is more tolerable, Barracuda uses a power supply based on National Semiconductor's LM2596S Simple Switcher power converter. The power supply uses two of these ICs in parallel, giving it an output of 5 volts, with a maximum current of 6 amps. In addition, the board contains multiple output connectors, allowing for connecting and disconnecting of components and additional headroom to add future components.

Control Board (mbed LPC1768)

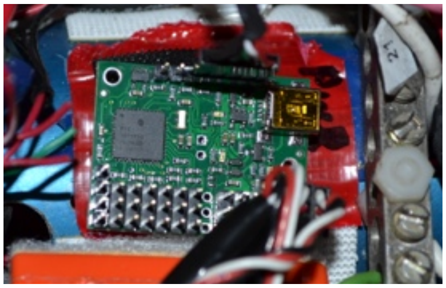


To enable the BeagleBoard to focus on high-level mission and image-processing tasks, Barracuda uses a control board to handle low-level tasks. This year, the team decided to replace the previous year's custom, dsPIC33FJ128GP804-based control board with the mbed LPC1768, a microcontroller development board designed for flexible and rapid prototyping.

The mbed is based around an ARM Cortex-M3 running at 96 MHz. It has a 512-KB flash disk for program storage, and 32 KB of user RAM. It also supports a variety of I/O. At its peak, it consumes only 0.6 watts of power.

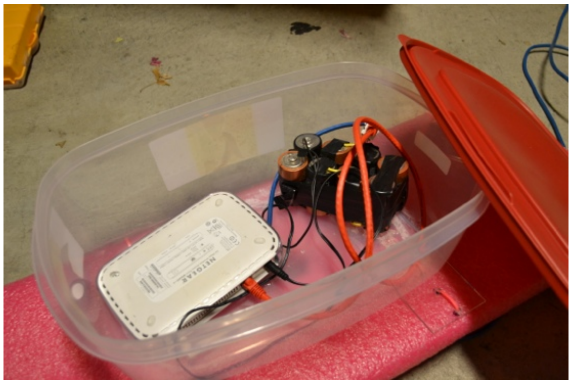
The board uses a set of serial lines to monitor the output of the navigation sensors and uses its built-in analog-to-digital converters to monitor the voltages of the pressure sensor and the kill switch. By appearing as a serial port, it then delivers this information to and receives control commands from the mission software running on the main computer. Several Proportional–Integral–Derivative (PID) loops receive input from the control commands and calculate adjustments to maintain the desired heading, depth, and pitch of the submarine. Finally, to move the motors, the board sends commands to the serial-to-servo board over another set of serial lines.

The mbed offers several advantages over the dsPIC33FJ family. First, it has a well-documented API that provides abstractions of esoteric registers used to access the hardware of the ARM processor. Second, the mbed cloud-based development environment allows team members to edit and compile code from a web browser on any computer of any platform. This eliminates the need for members to install software, facilitating member education by making it easier to begin developing for the mbed. Last, the mbed Controller provides easy, drag-and-drop flashing of programs onto the ARM Cortex by appearing as a USB external drive. This feature also allows flashing of the control board over the network while the AUV is still in the water, reducing the length of development cycles.



Serial to Servo Board (Pololu Mini Maestro 12-channel)

The Mini Maestro is a 12-channel serial-to-servo board manufactured by Pololu. Mounted in the rear end cap, the board generates PWM signals with 4 of its 12 channels to operate the Novak Super Rooster controllers in response to commands from the Control Board's serial module. The Mini Maestro also has 8 KB of flash memory to store scripts for hard-coding PWM signals; however, this functionality is not currently used by the AUV.

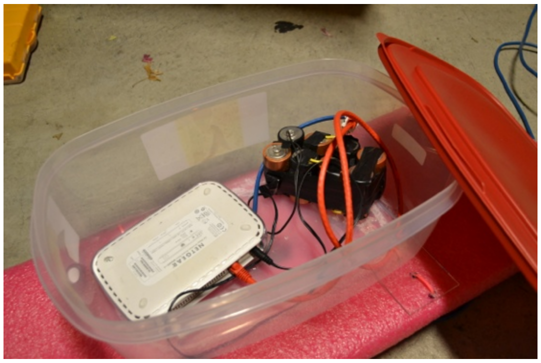


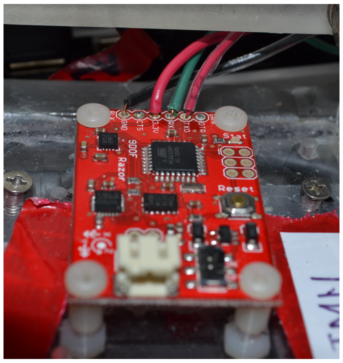
Novak Super Rooster

Barracuda uses the Novak Super Roosters reversible speed controllers, which are rated for 150 amps of current. The roosters are based on a reliable H-Bridge and are controlled using RC pulses. They were originally built for RC cars; however, these are more than adequate for Barracuda's thrusters. Again, the high-current Super Roosters maintain a tolerable temperature because they dissipate heat into the water via the connector end cap, which acts as a heatsink.

Tether and Communications

A communications link to the main computer is required to facilitate testing of the vehicle. This is provided by a wireless access point encased within a plastic food container. The entire assembly is attached to the vehicle via an Ethernet cable, which drags it along in the water. The main computer is connected to the tether via the Ethernet port on the USB hub. Through this network connection, team members can not only monitor and control the submarine, but also alter the mission code and even flash the control board. Furthermore, the tether, when connected, enables a remote engine kill that replicates the functionality of the kill switch. Finally, the tether enables manual control of the sub from the club's iNav application running on an iOS device, for demonstration and debugging purposes.

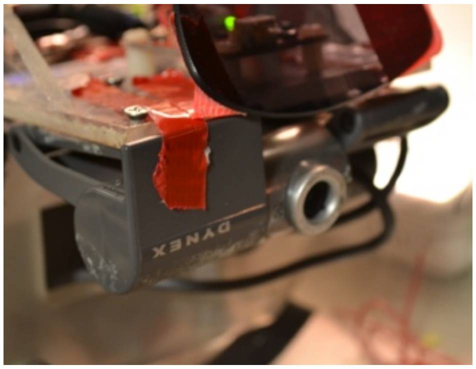




Navigation Sensors (Sparkfun 9DOF Razor IMU)

The Barracuda queries a SparkFun 9DOF Razor Inertial Measurement Unit (IMU), which contains a three-axis accelerometer (Analog Devices ADXL345), a three-axis gyroscope (InvenSense ITG-3200), and a three-axis magnetometer (Honeywell HMC5883L).

The magnetometer can be used to measure absolute heading based on the Earth's magnetic field. However, the presence of constantly changing soft-iron magnetic distortion caused by switching current in the motors required another way to determine heading, which is provided by the accelerometer and gyroscope. This allows the submarine to determine its relative orientation even in the presence of fluctuating magnetic fields.

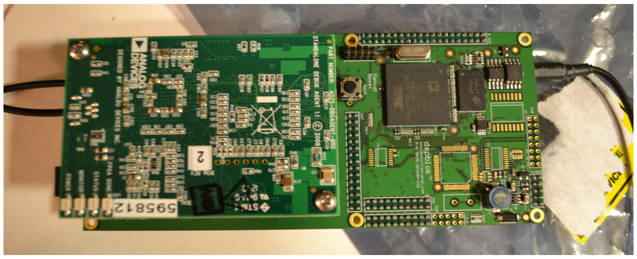


Camera

The Barracuda uses two 1.3-megapixel Dynex webcams. Using webcams brings several advantages: webcams (1) do not need separate analog-to-digital converters for video signal, (2) are compact, offering greater design flexibility for the AUV, and (3) are relatively inexpensive and therefore more replaceable.

The Barracuda's webcams communicate over a USB 2.0 bus with the computer, which reads and processes the video feed. The forward-facing camera detects the start gate, buoys, obstacle course, and circular cutouts, while the downward-facing camera detects the path and the bins.

Passive Sonar



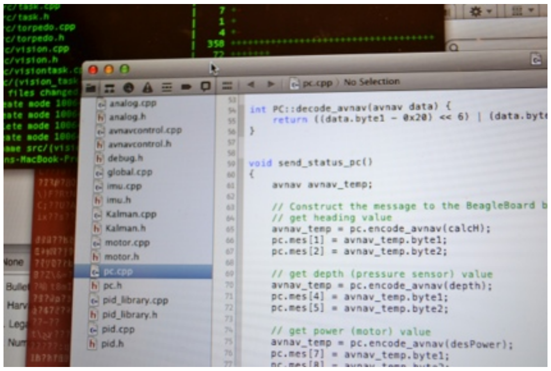
Last year, we began the process of redesigning and implementing our passive hydrophone sonar system. The hydrophone system consists of an array of four RESON TC4013 Miniature Reference Omni-directional hydrophones. The piezoelectric sensor element contained inside the rubber nibble produces an electrical signal on the order of approximately 40 mV, which propagates up the coaxial cable. The electrical signal is amplified using an L711JN operational amplifier (op-amp) to satisfy the 1.8-V minimum for the dspblok ad96k42 audio I/O board. After the dspblok receives the analog signal, it filters the signal and converts it from analog to digital. The digital representation of the signal is then passed to the ADSP-21469 processor on the dspblok 21469, where it is processed to determine the time difference on arrival (TDOA) for each signal received. The TDOA is then transmitted via an RS-232 connection to the BeagleBoard for additional processing. As we continue to improve the hydrophone system, our goal is to move from TDOA analysis to a more robust phase difference on arrival method, which will improve the system accuracy and reduce the sensor spacing requirement from approximately 12 square inches (30 square cm) to less than 4 square inches (10 square cm).

Software

The BeagleBoard runs the Extensible Vehicular Automaton software, which carries out the high-level management of the sub by reading input from configuration files, cameras, and passive sonar board, and by communicating with the control board, which is responsible for low-level tasks. EVA also contains many intelligent algorithms for target location, color identification, and edge detection. In the past two years, the BeagleBoard has run iterations of Autonomous Vehicle Intelligence (AVI). This year, the software team developed EVA because AVI was not flexible enough to accommodate the expansion they had in mind.

Development Strategy

To manage complex software projects with many collaborators, we use GitHub, a software-as-a-service (SaaS) website that specializes in hosting source-code repositories in the Git format. There are many advantages of such a version-control system. First, each developer given the ability to complete work asynchronously and without an Internet connection, leading to more flexibility and faster development. Second, the code is backed up on external servers and on each team member's computer, and the underlying version control system, Git, allows the team to revert the source code to older states that are known to be functional. The combination of these two features encourages team members to experiment freely when adding new features. Last, GitHub's Pull Request feature encourages discussion of software development, which is then archived. We hope that the records generated by Pull Requests will enable future club members to understand the reasoning behind implementations used in the code.



```
int PC::decode_avnav(avnav data) {
    return ((data.byte1 - 0x20) << 6) | (data.byt
}


void send_status_pc()
{
    avnav avnav_temp;

    // Construct the message to the BeagleBoard b
    // get heading value
    avnav_temp = pc.encode_avnav(calcH);
    pc.mes[1] = avnav_temp.byte1;
    pc.mes[2] = avnav_temp.byte2;

    // get depth (pressure sensor) value
    avnav_temp = pc.encode_avnav(depth);
    pc.mes[4] = avnav_temp.byte1;
    pc.mes[5] = avnav_temp.byte2;

    // get power (motor) value
    avnav_temp = pc.encode_avnav(desPower);
    pc.mes[7] = avnav_temp.byte1;
```

Class Design

One of the biggest problems with AVI was that it was not modular enough: even small changes required large refactoring of code. EVA attempts to fix this by keeping a single mission controller class, which encapsulates all positional variables, like AVI. Unlike AVI, which handled all navigation in the main class and vision in separate classes, EVA allows each class to run the sub when invoked. Since each class is self-contained, this design is highly modular.

Feedback Control

Used indirectly by almost every high-level control process on the vehicle, the feedback control allows the vehicle to adapt itself to changing external situations. This process implements a general Proportional–Integral–Derivative control loop, which can be run on any number of feedback inputs. Every feedback mechanism used by the system simply has to supply a configuration, an input value, and a set point. The feedback control process then produces a value from –1 to 1, indicating to the vehicle how it needs to change to reach the set point. The feedback control is used to control not only heading, depth, and pitch, which are handled by the control board, but also for some of the high-level navigation tasks. The control loop itself includes several features, including derivative filtration (important for sensors that have noise spikes occasionally) and mechanisms to prevent integral windup.

Speed Control

Upon receiving commands from the BeagleBoard, the control board sends motor commands to the serial to servo board, which drive the Novak speed controllers and the dropper.

Orientation Measurement

The control system combines the high-bias but low-noise integrated rotation data from the IMU's gyroscopes and the low-bias, high-noise data from the accelerometers and magnetometers by using several Kalman filters, which estimate the current orientation based on the previous known position of the AUV, the current sensor measurements, and dynamics of the AUV. These data are communicated with the BeagleBoard, which in turn sends the desired orientation.

Mission Control

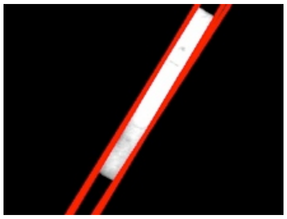
The mission control class contains the information on the state and current objective for the completion of the mission. Mission state is coordinated using shared variables, which other classes check to determine their behavior. Once the controller has determined that one portion of the mission has been completed, it updates the mission state variable, and the other classes adapt their behavior to the new mission state. Due to the increased flexibility of EVA over AVI, task execution is not required to be linear; failing to complete a task will not require a complete mission restart.

Optical Analysis

The BeagleBoard reads video streams from the two Dynex webcams using Video4Linux 2. But to make the navigation processes possible, it is necessary to have robust and effective optical analysis algorithms in place. In developing vision filters, the team uses the Mathematica software to model the filters and tweak settings. Then, these custom filters are implemented with the OpenCV library's image-handling functions. This allows the team to develop highly efficient optical processing algorithms quickly and easily.

Beginning the Mission

Once EVA is executed, it waits for the kill switch to be inserted. After insertion, the vehicle dives to mission depth and continues straight through the validation gate. During this time the optical processing code begins to search for the buoys. Based on the order of colors specified in the configuration file, it checks the forward camera for the presence of a circle of the first color within the image. The center of this circle and the center's distance from the center of the field of view are computed. Any deviation causes a command to be sent to the control board to correct for the deviation. When a successful collision is detected (signaled by the sudden disappearance of the enlarged buoy), the submarine moves backward and targets the buoy of the second color in the same way.



Following the Path

With a second web camera facing directly downwards, the BeagleBoard is able to switch between input sources to view the orange path that outlines the course. Similar to the buoy algorithm, EVA's path algorithm quickly filters out the orange color of the path. Then, it applies a Hough transform.

The goal of the Hough transform is to find a geometric description of lines within images, including disjoint (dashed) lines, which makes it ideal for recognizing the edges produced by the target. The ideal output of a Hough transform on the box target image is a grid of 2 lines overlapping 2 other lines at right angles.

From the Hough transform's output, EVA can calculate the difference between the slope of the path and that of a vertical line. The vision processing unit can then adjust the desired heading.

Edge Detection

The first step in finding the bins is finding the edges. First, to cut down on noise, the image is normalized. Then, EVA uses a Canny filter to find the edges. The results are run through an intensity threshold, which provides a black-and-white image. The binary nature of the image makes it more suitable for the Hough transform and less CPU-intensive to process.

Firing Torpedoes into Caesar

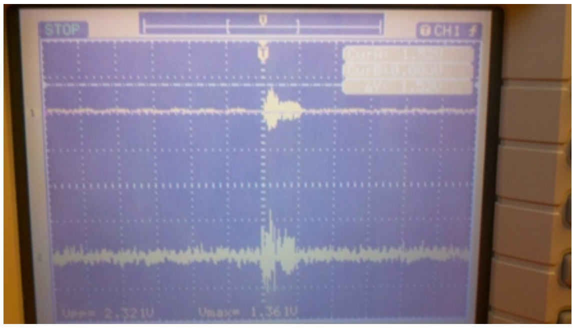
The vehicle resumes processing forward vision once the markers have been dropped. It then searches for an intense red object, which would indicate the presence of the cutouts. Once the cutout is found, the vehicle slows down and orients itself in front of the cutout of a specified color. The vehicle uses the size and distortion of the cutout to determine its relative placement and orientation. Once it is properly aligned, the vehicle will fire its torpedoes.

Dropping Markers in the Bins

While continuing to travel straight, the vehicle searches for the orange path with the downward-facing camera and the target bins. When one of the bins is located, the vehicle will attempt to station itself directly over the center of the target, using the center of "mass" image-processing algorithm. Once the vehicle is positioned properly over the center of the bin, it drops both markers and proceeds toward the window.

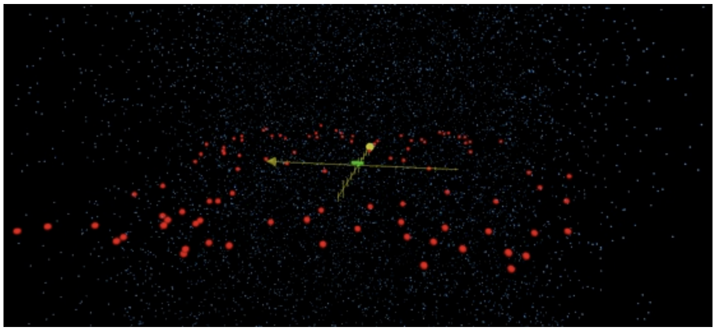
Manipulation Task

After dropping the markers in bins, the vehicle will travel to the center of the pool. The presence of a yellow object denotes the board, and the orientation of the cylinders will be found by measuring the intensity of the color red. A probe is then used to manipulate each cylinder off its holder.



Navigating to the Laurel Wreath

After completing the marker drop, the computer signals the hydrophone board to begin analysis of the audio data. The dspblok polls the hydrophones through the analog-to-digital converter at a constant interval, and fills a buffer. Once the buffer is full, it runs algorithms on the data in the buffer to filter out the noise, check for erroneous readings, and finally, check if a ping was received. If a ping was received, the board sends the time difference of the pings to the BeagleBoard via RS-232. The BeagleBoard uses these data to calculate a vector of the pinger's relative location and then navigate to the pinger.



Retrieving the Wreath and Surfacing in the Palace

Once the vehicle is positioned above the pinger, signified by a directional vector with very small X, Y, and Z components, it secures the structure using its passive grabber by pitching forward and then backward. After the structure is secured, the vehicle will then carry it to the other octagon, navigating again using the hydrophones. Once there, it will rise until the pressure sensor detects that the AUV is at the surface, pitch forward to drop the structure, and shut off the motors to complete the mission.

Conclusion

The advancements made this year will allow our AUV, Barracuda Mark XI, to face the new challenges presented by the 15th AUVSI RoboSub Competition. We hope that these changes will make Barracuda a much more competitive entry, and the transition of this year's team to the next a much smoother process.

Acknowledgements

Several individuals have provided us with their guidance throughout our project. We would like to thank them because they have helped the team succeed: Joel and Tim Soppet, Karl Schulze, Mark Pereira, and Chris LaFlash offered their advice when the team encountered persistent technical issues. We thank our advisor, Bree Barnett Dreyfuss, for her regular discussions about the administrative side of the team. We also thank the Kalidas family for use of their garage and pool, and all of our parents for their unwavering support of our project. Finally, we thank our sponsors: Amerimade Technology, FrontRange Solutions, Hans Hansen, Karl & Melonie Schulze, mbed, and UPrinting.com.