



Team: Mark Carlin, William Carrington, Michael Plante, Alex Raymond, Raul Rodriguez, Randy Rosales, Daniel Schlieter, Ian Williams, Collin Wells
Advisor: Dr. Edward Esposito

INTRODUCTION

The University of Texas at Dallas' 2007 autonomous underwater vehicle exemplifies the integration of an all-encompassing rover solution fully capable of completing all tasks required at the 2007 AUVSI and ONR's International Autonomous Underwater Vehicle Competition. 2007's vehicle design incorporates the benefits of a successful single-hull mechanical structure with highly integrated subsystems that culminate into a design that successfully fulfills the requirements of this year's competition.

STRATEGY & ARCHITECTURE

Mission strategy

This year's vehicle has a mission control system and sensor subsystems tailored for the pirating tasks required during competition. The vehicle's mission control system will incorporate the capabilities of all sensor subsystems into a decision-making algorithm designed for the highest probability of task and mission success.

Our vehicle will begin its voyage at the launch point and pass through the starting gate submerged, completing the first mission task. The vehicle structure facilitates smooth and steady movement with excellent motor control. Securing a bearing will be achieved by applying a discrete controller to the onboard inertial measurement unit.

The vehicle will then use two sensor subsystems to approach the eerily red glowing beacon. Built specifically for this task, the light detection subsystem consists of multiple photodiodes arranged to determine the direction and modulation frequency of the beacon. The forward-looking camera, a portion of the overall vision subsystem, will be used in this task to assist in a successful dock with the eerie beacon.

Navigating the ominous waters surrounding the infamous “dashed line” will be achieved solely by the camera vision system. Both camera systems will distinguish the “dashed line” from the surrounding environment by tracking image features dominated by the orange hue component of the line. The forward-looking camera will be used to maneuver the vehicle on top of the dashed line, while the downward-looking camera will be used exclusively to track the lines and to position the vehicle directly along their path. While traveling along the dashed line, our vehicle will successfully fire a marker into the first target bin of the Davy Jones Tribute. The inner and outer portions of the target bin will be identified using the downward-looking camera. Also, while traveling along the dashed line, our vehicle will use the forward-looking camera and the photodiode sensor subsystems to force the red-eyed sea beastie into firing range. We will ram her to uncover the second target bin and blast off a second marker into the cold PVC heart of the Davy Jones tribute.

The final test of our vehicle’s pirating potential will be the plunder of the lost “Treasure X” from the depths surrounding the acoustic pinger. The primary sensor subsystem for this task will be SONAR. This subsystem consists of three passive hydrophone sensors capable of determining pinger bearing and elevation from the vehicle using time-of-arrival differences. Once the vehicle has centered over the pinger, our grabber claw of carnage will seize the X out of its peaceful slumber. The vehicle will emerge from the chlorinated abyss within the surface octagon, victorious in its pirating endeavor.

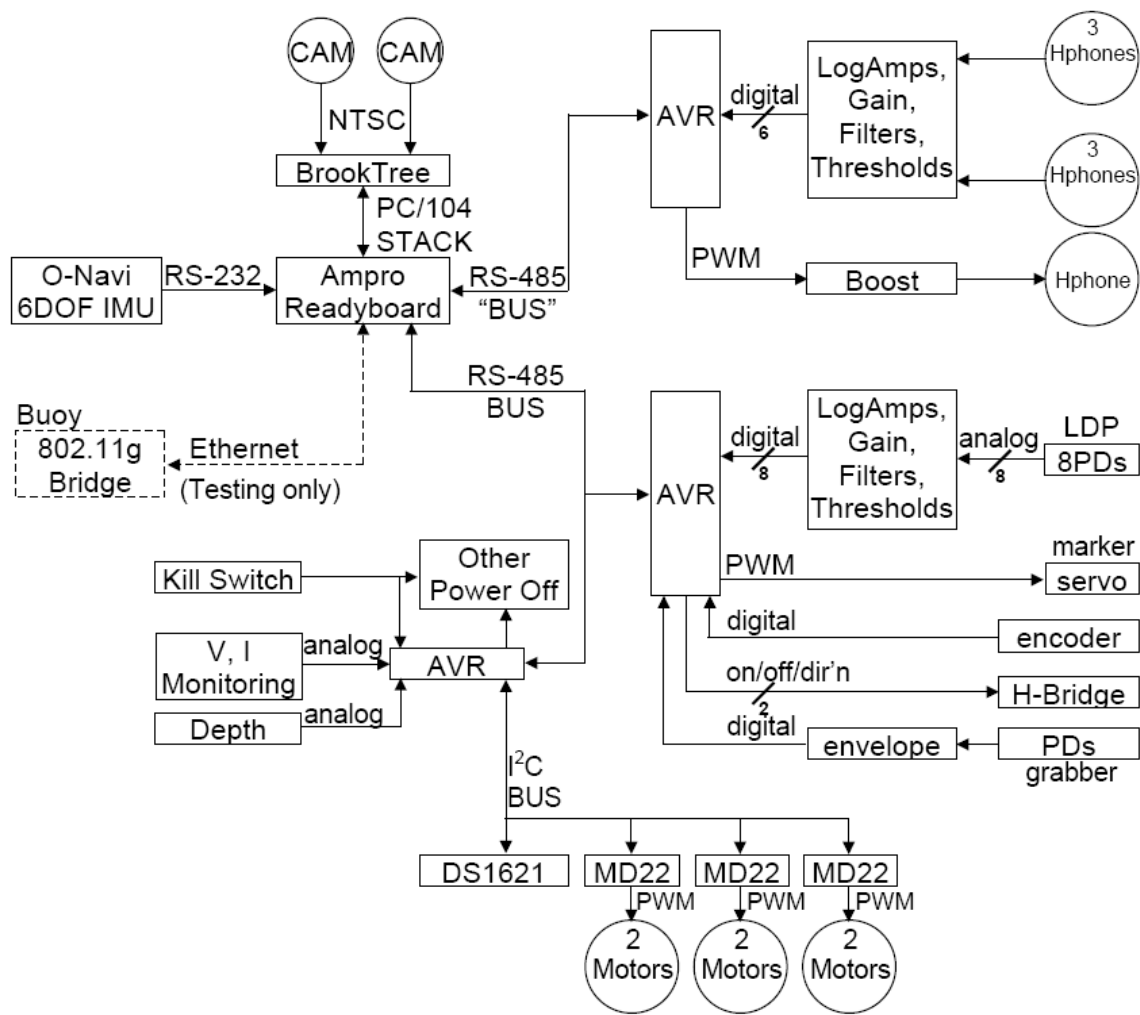
Vehicle Design

This year’s vehicle will follow on the success of the single-hull design developed for last year’s competition. The vehicle uses a single sealed tubular hull to contain all necessary electronics and battery systems. All connections to external actuators and sensors are made through the vehicle’s faceplate that incorporates Bulgin IP68-rated connectors. The hull, motors, and sensors are mounted to an MK Automation extruded-aluminum frame.

The vehicle uses COTS high-current lithium-ion battery technology to enable both extended operation during development and rapid recharging, giving virtually continuous operation. A battery tray steadies the batteries inside the hull and enables rapid battery replacement.

System architecture

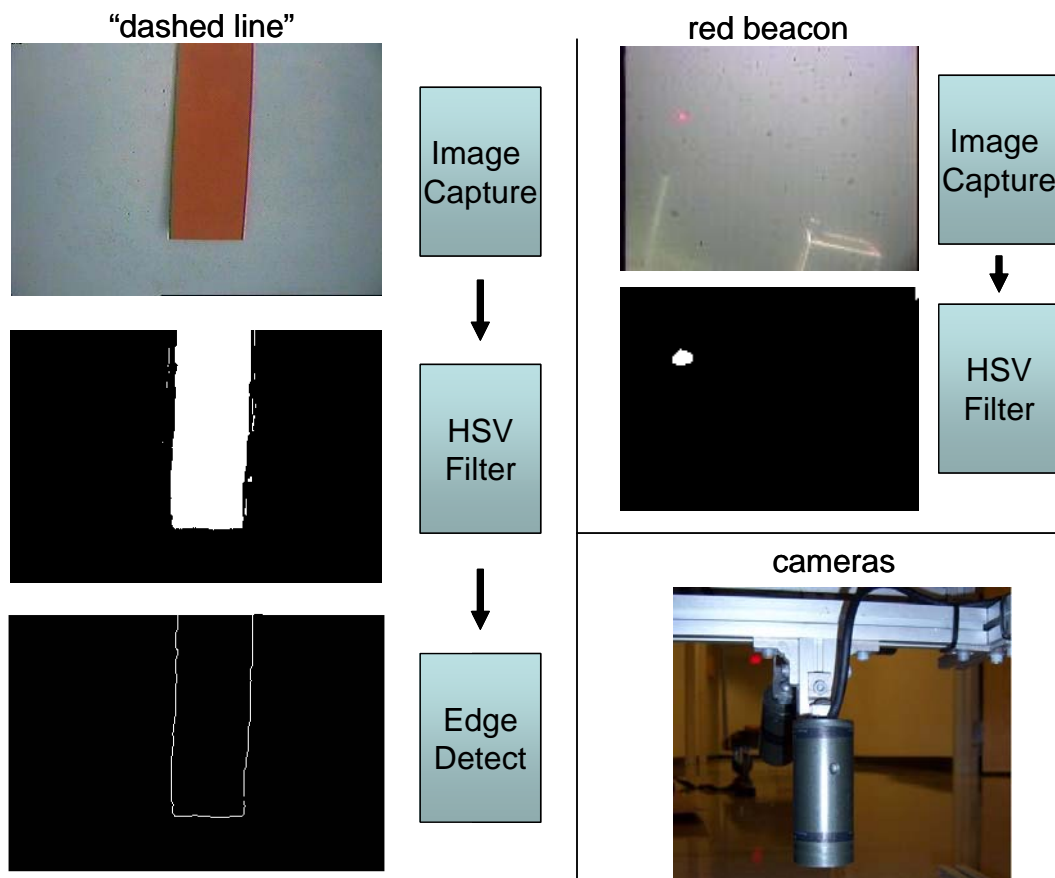
This year's vehicle is organized into a three-level hierarchy consisting of the small form-factor (SFF) PC, subsystem boards, and sensing/actuating devices. This hierarchical design allows each microcontroller/device subsystem to be modular and therefore replaceable with little or no modification to the core CPU. The vehicle builds off existing 2006 systems and integrates new systems in order to achieve this year's goals. The SFF PC runs the high-level control and vision software. The SFF PC uses the STX/ETX packet protocol to communicate over the RS-485 buses with the microcontrollers. The microcontrollers communicate with the sensors/actuators, including the hydrophones and grabber arm, which will be explained in more detail throughout this document. Below is a diagram of the entire system architecture. The three-level hierarchy is illustrated with the top tier being the Ampro Readyboard PC, the second level subsystems controlled by AVR microcontroller boards and the bottom level by the sensors/actuators.



SUBSYSTEMS

Vision

The vision subsystem consists of two underwater NTSC cameras, a PC-104 multi-channel capture card, and image processing algorithms implemented exclusively within PC software. There are two cameras, one facing downward, and one facing forward roughly 45° below the horizontal. Both camera image streams transform the captured image into the Hue-Saturation-Value (HSV) color space. For the dashed line and target bins, the software detects edges and fits each contour to a Bivariate Gaussian to determine the direction and position of the object of interest. Shown below are vision processing steps for identifying the dashed line and the red beacon.



A more detailed description of the image processing algorithms used to identify and track the dashed line and the red beacon are presented in the SOFTWARE section.

SONAR

The SONAR subsystem will build primarily off 2006's successes. The sea siren pinger detection architecture includes three passive hydrophones, front-end analog gain/filtering/thresholding stages, measurement on an AVR, and bearing/elevation calculations on the PC. The hydrophones are arranged in a horizontal plane, and the time differences define two hyperboloids of two sheets. The intersection of the asymptotes indicates the two possible 3D directions to the beacon, and the depth sensor resolves the ambiguity.

Light Pod

The photodiodes mounted on the pods detect the incoming light from the light buoys. The circuit utilizes logarithmic amplifiers and analog filters to gain, filter, and threshold the incoming signals into digital signals. Each input feeds into an amplifier that logarithmically gains the incoming signal based on a tunable reference current. The signal is then filtered by a third order active bandpass filter that has a bandwidth from 2 kHz to 6 kHz. The signal is then gained to saturate the operational amplifiers, sent through an envelope detector, and then compared to a threshold voltage with an operational amplifier. The software polls the thresholded outputs and determines the direction to the buoy based on which channels are active.

Inertial Measurement Unit (IMU)

The primary sensor for vehicle motion control is the Falcon GX/4 MEMS Inertial Measurement Unit. This IMU has six degrees of freedom and will explicitly measure yaw rate as a feedback parameter to vehicle control algorithms. The vehicle Proportional–Integral–Derivative (PID) controller will provide robust relative bearing tracking capability utilizing the IMU.

Depth Sensor

This year's vehicle employs a high-performance commercial depth sensor, the GE PTX 1830. Encased in a submersible titanium shell, the GE PTX 1830 can accurately measure pressures up to 900 psia. A tuned differential amplifier scales the depth sensor's output to 0–4.7 V, in keeping with the operating range of 0–38 ft. A microcontroller reads the analog output through an A/D pin. The depth sensor is the primary input to the depth PID controller.



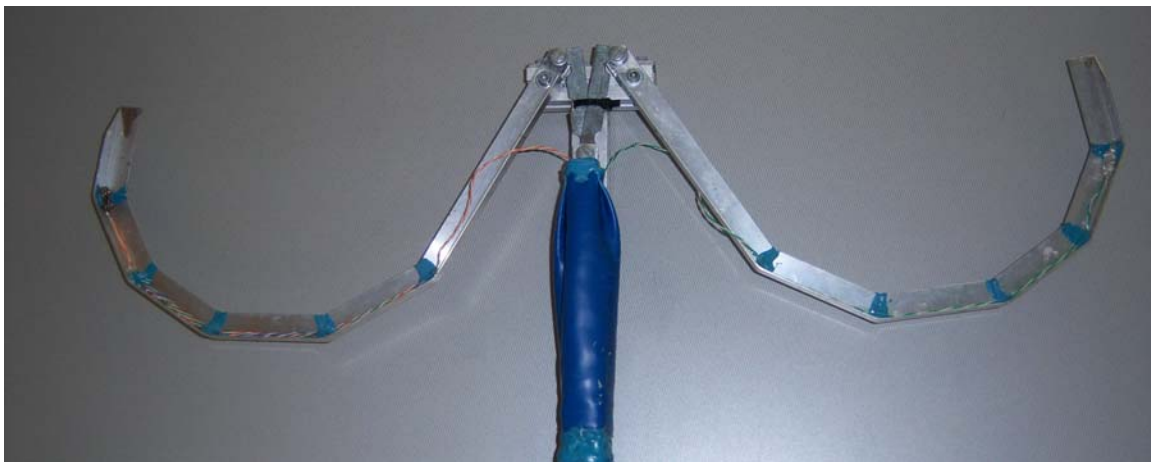
Marker Dropper

The marker dropper subsystem has a sliding plate with rare-earth magnets that retain the markers until the system chooses to drop them into the desired bins. A Hitec HSR-5995 digital servomechanism, driven by a serial board, controls the position of the plate and enables the system to drop each marker sequentially. Each marker can be dropped by commanding the servo to assume a particular rotation angle. This system enables the vehicle to carry markers without consuming additional power or dropping markers in case of an emergency shutdown.

Grabber Arm

This year a custom grabber arm will accomplish the task of excavating the “X” from above the sonar pinger. The grabber arm consists of a linear actuator and two identical semi-circular claws with crossbar pivots mounted on extruded aluminum. The linear actuator is a motor that turns a screw, providing linear motion on one axis. The claws are mounted onto the aluminum crossbar to allow motion in a single plane perpendicular to the length of the vehicle. The linear actuator controls the claws to grab the object.

An optical encoder attached to the motor’s axle feeds output into a microcontroller to track position via software. On the end of the claws, a detection circuit determines whether the “X” has passed into the grasp of the grabber arm by using an emitter/receiver circuit made out of a photodiode, a group of LEDs, and modulation circuitry. When attempting to pillage the X-booty, the actuator moves into a known position that aligns the photodiode and LEDs, providing a line-of-sight corridor that the microcontroller constantly monitors. The vehicle begins descending to the plane of the “X” until the booty breaks the beam between the photodiodes and the LEDs. The microcontroller notices the disconnection and sends a signal to the claws to grab the booty. The grabber arm will remain closed, grasping the “X” until the vehicle surfaces.



Motor Control System

The vehicle includes three Devantech MD22 dual-channel motor controllers, each commanding a pair of thrusters. To ensure safe and reliable operation, the boards interface with a microcontroller responsible for commanding them and a power board with the ability to terminate motor power via the kill switch. The system allows either the microcontroller software or the hardware kill switch to terminate motor function, ensuring that an operator can halt operation in or out of the water. Furthermore, the system includes a watchdog timer that halts the motors in the absence of a periodic reset from an external command source. This ensures that any malfunction in the autonomous control software will not endanger the safety of the vessel, crew, or divers nearby.

Thrusters

The vehicle uses six SeaBotix brushed thrusters mounted in pairs along the three primary axes providing depth and planar vehicle control with up to four pounds of thrust. Two thrusters are mounted vertically at the fore and aft in order to submerge the vehicle, while the remaining four are mounted horizontally to provide yaw, drive and strafe control. This arrangement achieves maximum stability while presenting a robust control model to the vehicle control systems, enabling rapid development of precise motion control algorithms.



Thermometer

The vehicle's hull contains several high current devices that dissipate heat; thus, a DS75 thermometer monitors temperature inside the hull. The DS75 has appropriate resolution, supports rapid temperature polling, and communicates over the I²C bus with the microcontroller. The PC will monitor heat and shut down non-critical systems when the temperature exceeds the threshold.

SYSTEM BOARDS

Small Form-Factor PC

The small form-factor (SFF) PC, an Ampro ReadyBoard 800, is the captain of the vessel, receiving subsystem data and issuing all orders. The ReadyBoard 800 utilizes a 650 MHz ultra-low voltage Celeron processor, an Ethernet port, PC-104 add-on card support and serial ports with RS-232 and RS-485. This system runs the mission software, navigation software, and vision processing software using a custom-configured Gentoo Linux installation. A laptop hard drive is connected to the on-board mini-IDE

port. The subsystem microcontrollers use RS-485 to communicate with the PC. For testing, a floating communications buoy is connected through Ethernet while the vehicle is in the water.

Microcontroller Boards

The microcontroller boards interface with the sensors/actuators and the PC by running low-level software that controls the subsystems. These boards communicate with the PC using full-duplex RS-485 and with the subsystems using I²C, A/D conversion, PWM I/O, general purpose I/O, SPI, and JTAG interfaces. We are using two versions of the microcontroller boards: one with an Atmel AT90CAN128 microcontroller and the other using the ATmega2560 microcontroller. The AT90CAN128 has two hardware UARTs, 128k of flash memory, and was applied successfully last year. The ATmega2560 was integrated because of its greater peripheral support.

Power Management

The crew selected Texas Instruments high-efficiency switching regulators to provide low-ripple power to the CPU and other vehicle systems. The vehicle uses three regulators: one providing 5V 4A power to the CPU, a similar one providing 6A to the other electronics and a third providing 9V 1.5A power to the cameras. At the specified loads, the regulators operate at approximately 90% efficiency providing run times of up to two hours.

The central power regulation circuits on the power board feed power to all the electronics. Each regulator's inhibit line connects to one of the microcontroller boards running power management software to control the on/off state of each rail of power. This board is connected to an independent bus and is thus capable of controlling power to the vehicle network and the camera system. Furthermore, voltage divider networks provide compatible voltages to the A/D pins on the serial board, allowing the CPU to monitor the state of all batteries. This is also advantageous during testing, as components can be powered down remotely when not in use in order to conserve power and extend development time in the water.

SOFTWARE

Player

Player is an open-source software framework that provides an abstraction layer between software components, facilitates simulation with few modifications, and allows rapid testing via a number of canned executables. Among other things, Player allows us to communicate in a hardware-independent fashion, manage the mission in a flexible manner, and work in conjunction with one of two simulators, Stage or Gazebo. Player

itself implements a client/server architecture, where the server consists of a collection of drivers, each of which provides several predefined interfaces back to Player. A driver is responsible for communicating with the physical hardware, while the interface describes some abstract piece of information, such as an estimate of velocity or a command to move a servo. The pairing of one or more interfaces with a driver gives meaning to the information each interface holds; this pairing is done in a plaintext configuration file. The client, on the other hand, generally contains the high-level control algorithms for the vehicle. The drivers, clients, and sometimes interfaces must generally be rewritten for each new seadog, which the crew handled in implementing the software system for the 2007 vehicle. The Player package then manages dependencies between drivers, marshals the “interfaces” over TCP/IP, parses configuration files, and performs a variety of other miscellaneous functions. As an example, UTD’s software utilizes Player’s dependencies for simple control systems, such as state estimation and PID control: a PID controller might rely on the motors and the state estimator, while the state estimator might rely on a variety of sensors, such as an IMU, a compass, or a depth sensor.

By using freely modifiable software, UTD eliminated a large percentage of the necessary development and testing time, enhanced flexibility as far as where the components can run, and organized the software architecture in a logical manner that promotes clear program design and reuse. An unfortunate consequence of this method, however, is that a good deal of tweaking is required; in particular, the simulators are primarily targeted at land-based vehicles, and the team did not have sufficient time to modify the Gazebo simulator to a point where it could return realistic sensor data in a simulated underwater environment. Nevertheless, this approach reduced testing time, resulting in far fewer places to search for bugs, given the rigorous testing a well-established project like Player has already received at the hands of amateurs worldwide.

So far, the team has written quite a few drivers, which have been grouped by the bus to which they need access. Specifically, the team implemented a driver that manages the device(s) on each RS-485 bus, a driver that manages traffic on the IMU’s dedicated connection, and a set of drivers that perform relatively constant control system functions. One AVR handles the start switch, power bus inhibit lines, motors, depth sensor, thermometer, and power level monitor. Another AVR handles light detection, the marker dropper, and the grabber mechanism. The last AVR handles the precise SONAR timing. Vision is currently being integrated in a non-standard manner into the Player server. Development of a client has been hindered by design dependencies on hardware interfaces, though it will be ready in fairly short order, thanks to the extensive work already performed on hardware abstraction. The resulting server is highly configurable, allowing permutations of motors, arbitrary mounting orientations for the IMU, and an ad hoc selection of required devices.

Vision Processing

The vision processing algorithms take as input the two image streams captured from the downward- and forward-looking cameras as discussed in the SUBSYSTEMS/VISION subsection. These programs, executed entirely on the PC, use a modular framework with well-defined input and output formats to facilitate experimentation and tuning of the vision algorithms. The pixels of one of the two image streams are converted from the NTSC cameras in Red-Green-Blue (RGB) color space to the Hue-Saturation-Value (HSV) color space. The stream then passes the transformed image through a 9x9 mode filter to remove spurious matches. This operation is performed on each image for empirically-derived HSV values for orange, red, white, and black. Four binary images are created with this information. In order to detect and track the red beacon, the mean pixel coordinates of the red component are relayed to mission control.

In order to track the pipeline and bins, a further stage of processing is required. For each potential object in the appropriate bitmaps, the edge points are extracted into lists of vertex coordinates. Henceforth, the algorithm operates on the image as lists of points rather than a raster of pixels. The pixels on the edges of the pipeline are modeled as Bivariate Gaussian variables, and their covariance ellipses are calculated. Several additional heuristics are then applied to determine the plausibility of the targets; ones that fail are excluded from further consideration. The algorithm finally reports the major axis of each ellipse to the control system as the location and orientation of the object to track.

CONCLUSION

The University of Texas at Dallas' Autonomous Underwater Vehicle was built to succeed in this year's AUVSI and ONR competition. We have used last year's successes including the single-hull design and SONAR subsystem performance to develop and integrate new subsystems, most notably Vision Processing. 2007's AUV will successfully complete this perilous pirating mission.