FAMU-FSU College of Engineering
Department of Electrical and Computer Engineering
EEL4914C / 4915C – ECE Senior Design Project II

Detailed Design Review and Test Plan

Robosub: Autonomous Underwater Vehicle

Team Members:
Tra Hunter
Antony Jepson
Ryan Kopinsky
Kashief Moody
Eric Sloan
Hang Zhang

Supervisors:
Dr. Michael Frank
Dr. Bruce Harvey
Dr. Zohrob Hovsapian
Dr. Srinivas Kosaraju
Michael Greenleaf
Dr. Chiang Shih

Submitted in partial fulfillment of the requirements for
EEL4911C – ECE Senior Design Project I

February 2, 2012

1 Executive Summary

Team Robosub is producing an Autonomous Underwater Vehicle, affectionately called a Robosub, to represent the Florida A&M University – Florida State University College of Engineering (FAMU-FSU CoE) and compete at the Association for Unmanned Vehicle Systems International (AUVSI) Robosub competition. This system is being developed according to the guidelines published by the competition ruling committee with an anticipated completion date of April 2012; the competition takes place in San Diego in July 2012.

Currently, the system consists of five major components: the Hull / Frame, supporting the peripheral sub-systems, the Interior Hull, housing the electronics, the Electronics, containing the decision making portion of the Robosub, the Mechanical Sub-systems, such as the torpedo launcher and marker dropper, and the Electrical System, controlling power expenditure of the all systems on the Robosub. These systems are used in tandem to create an autonomous underwater vehicle (AUV) capable of completing all the tasks required by the competition for the 15-minute duration of the mission.

Team Robosubs AUV is designed with modularity and cost-effectiveness in mind and will achieve the competition tasks by employing an arrangement of thrusters (for general maneuverability), a torpedo launcher (to fire torpedoes through PVC cutouts), a visual system (for object, shape, and color recognition), a grabber / dropper (to grasp and release objects), an array of hydrophones (for acoustic pinger triangulation), a remote kill switch (to terminate vehicle operation), and an inertial measurement unit (to capture orientation and acceleration of the vehicle).

Engagement of this inter-disciplinary design will be completed by six highly qualified engineers from both the Electrical and Computer department and the Mechanical Engineering department at the FAMU-FSU College of Engineering. Team Robosub endeavors to deliver a cost-effective and efficient design that not only meets but exceeds the competition requirements and is firmly committed to completing the project within the aforementioned timeframe.

Currently, the main hull, torpedo launchers, torpedoes, compressed air distribution system, frame, camera enclosures (version 1), marker dropper, multi-level electronics rack, and the vast array of mounts for the peripheral subsystems have all been carefully manufactured and assembled. Furthermore, the control units, motor drivers, inertial measurement unit, pressure transducer, hydrophones, and SEACON underwater wet mate connectors have either been ordered or received. All the various smaller components have also been either ordered or obtained, so not further orders (or at least any further significant orders) are expected to remain. The vehicle is on pace to be mechanically and electrically (including interface circuits) completed by spring break due to the ardent work and organization of the mechanical engineers, ideally providing the design team with at least half of March and all of April to focus solely on programming and debugging the vehicle to navigate through, and complete the tasks encountered in a replicated mission course at the FSU Morcom Aquatics Center. Pursuit of further sponsorships will be conducted in late March and April in order to obtain the necessary funding for travel, lodging, and shipping of the vehicle to the competition in San Diego. A new sponsorship has recently been obtained, setting the team on the path to meeting this goal. Overall, the outlook is optimistic, and this design team is committed to finishing this project strong and representing the FAMU-FSU College of Engineering well in the competition as first-time competitors.

Contents

List of Figures

Listings

2 Introduction

2.1 Acknowledgements

Team Robosub would like to thank the following individuals for suggestions, corrections, and material that were used to further development of the project:

Dr. Michael Frank, Michael Greenleaf, Dr. Bruce Harvey, Dr. Zohrob Hovsapian, Dr. Srinivas Kosaraju, Dr. Chiang Shih, Dr. Uwe Meyer-Baese, Dr. Ming Yu

Further accolades go out to:

Harris Corporation for their generous contribution to the project, the Florida A&M University – Florida State University College of Engineering for their investment in the project, and the Navy Engineering Education Center for their support in helping us travel to the competition.

2.2 Problem Statement

2.2.1 General Problem Approach

The Association for Unmanned Vehicle Systems International (AUVSI) preliminary mission statement asks competing teams to demonstrate autonomy by completing underwater tasks:

Gate Pass through a gate.

Buoys Strike two of three buoys of particular colors.

Box Crossing Navigate through a PVC[1] rectangle (does not contain bottom side).

Drop-in Bin Drop two markers into two explicit bins (out of four).

Torpedo Launch two torpedoes through a PVC structure.

Surface and Recover Surface under an octagonal region guided by an acoustic signal. Next, collect another object and surface within a second octagonal region. Finally, drop the marker.

2.2.2 General Solution Approach

Team Robosub aims to solve these problems by building a system that contains: vision systems, for navigation and identification of the borders of objects; electrical system, for delivering power to the various systems; peripheries, such as hydrophones, grabbers, and droppers, in order to complete the competition tasks; a robush inner hull to house the electronics; and a strong exterior frame upon which the components are secured.

2.3 Operating Environment

The competition takes place inside an anechoic[2] saltwater pool located at the SPAWAR[3] Systems Center TRANSDEC[4] Facility in San Diego, California. This pool contains ocean water and has a maximum depth of 16ft (4.88m) with a temperature range of 70–75°F (21–24°C).

Local testing will take place at the FSU Morcom Aquatics Center, located near the College of Engineering.

2.4 Intended Use(s) and Intended User(s)

The designed and constructed autonomous underwater vehicle is intended for the specific use of competing in the RoboSub Competition in San Diego. However, the end product and experiences gained will provide insight into the design of autonomous submarines for potential use in the Navy or other real-world applications, and will also provide the design team with further experience in regards to the proper execution of an engineering project from beginning to end.

While the intended users of this AUV are the design team, future potential users of the device that the team derives, or particular design features of the device, are people engaging in rescue operations and underwater marine researchers.

[1] Polyvinyl chloride
[2] not producing echoes
[3] Space and Naval Warfare
[4] Transducer Evaluation Center

2.5 Assumptions and Limitations

Assumptions are items that are taken for granted in the product. *Limitations* are the factors over which the engineer has no control.

- Assumptions
  - The competitions takes place in a salt-water ancheoic pool.
  - The device will be powered by a battery.
  - Batteries do not overheat and damage themselves or the vehicle.
  - Assigned tasks do not change when the final rules / competition guidelines are released.
- Limitations
  - Frequency response of the hydrophones, rated for 1 to 28,000Hz.
  - Lighting of the pool.
  - Time to complete construction of the vehicle and programming the vehicle: competition is in July 2012.
  - Engineer availability after graduation (all members are graduating seniors).
  - Life cycle of battery.
  - Thruster pairs (e.g. top–bottom) do not draw the same amount of current.
  - Vehicle weight must be no more than 84 pounds to avoid penalty.
  - Budget, pending additional sponsorships.
  - Battery open-circuit voltage not greater than 60 VDC.

2.6 Expected End Product and Other Deliverables

By July 2012, Team Robosub will deliver a complete autonomous underwater vehicle capable of completing the tasks listed by the competition ruling committee. This Robosub will be able to autonomously navigate underwater using a combination of a main control unit, computer vision system, and a guidance system.

Additionally, the AUV will be augmented with periphery systems that enable the AUV to complete the competition tasks, such as grasping and releasing a laurel wreath, launching a torpedo, and locating a pinger in the water. The competition also calls for a video and paper that details the teams submission. This and the other mentioned deliverables have an expected delivery date of April 2011 (before the term ends).

Furthermore, this system contains a full-fullytional propulsion system which is capable of moving the AUV underwater at a specified path and detection. It also contains an electrical system that powers the AUV with enough power, a stable current and voltage to ensure reliably component actuation.

Finally, a mission control unit will be implemented that directs the AUV to complete each task succesfully, manage communication between different modules, and autonomously control the AUV underwater. between

See the schedule for more detailed milestone dates.

3 System Design

3.1 Overview of the System

The AUV has an open frame that supports peripheral subsystems, such as the grasp / release mechanism, and a centrally-located, water-tight hull that houses the electronics, such as the BeagleBoard-xM. The design is almost completely symmetrical in order to produce a more robust vehicle that is not only less susceptible to disturbance forces but is also easier to stabilize and maneuver.

3.2 Major Components of the System

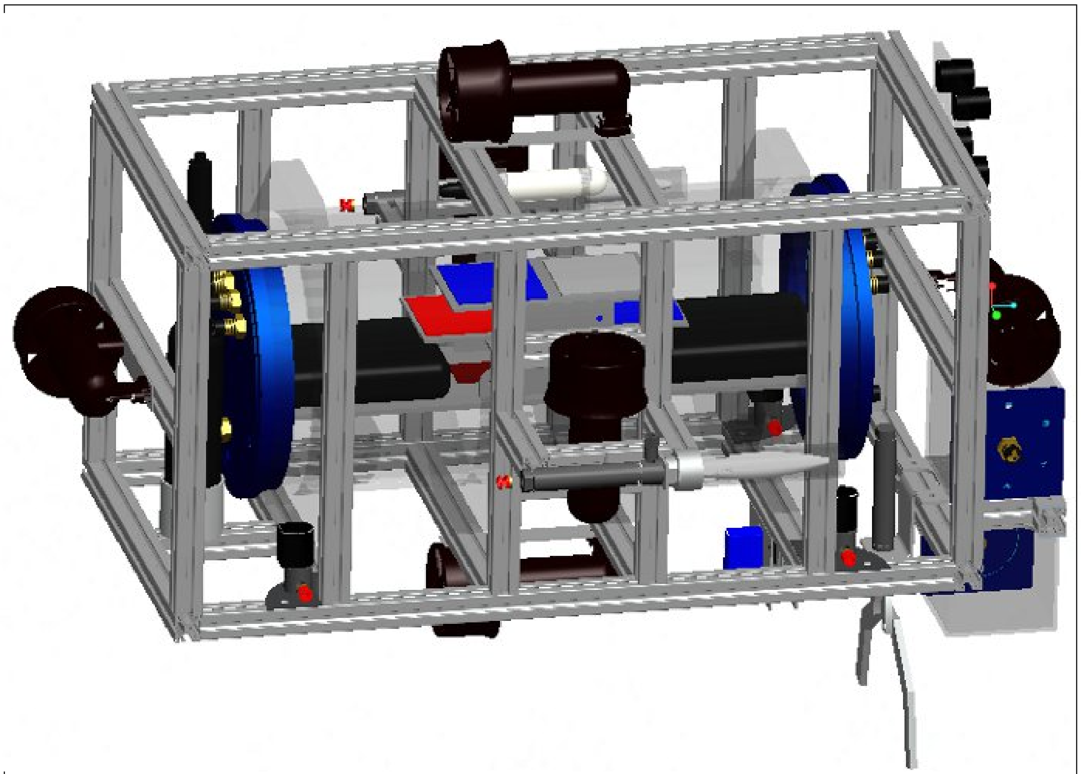


Figure 1: Pro / E Rendering of AUV

3.2.1 Hull / Frame

A versatile, rectangular 80 / 20 T-Slotted aluminum frame is used to support the peripheral subsystems (detailed below), the acrylic hull, and the acrylic, optically transparent camera enclosureseach housing either a forward-facing or downward-facing Logitech C615 web camera. The frame has all the necessary custom attachments for each of the external components of the AUV and provides an open canvas for various subsystem mounts. A cast acrylic tube with custom aluminum end caps is used for the hull. It is located at the center of the frame, houses the power supplies and electronics, and serves as the heart of the AUV while providing necessary buoyancy to the vehicle due to the enclosed air.

Components

- 80 / 20 T-slotted aluminum

3.2.2 Interior Hull

The interior of the hull supports the two lithium-ion battery packs, as well as the Arduino Board, BeagleBoard-xM, inertial measurement unit (IMU), three L298 dual h-bridge motor drivers (for the thrusters), and interface circuits for the solenoid valves and hydrophones (mentioned below). It has simultaneously been designed to efficiently and effectively dissipate heat away from the electronics and into the surrounding salt water environment via the integration of a thermally-conductive platform which will serve to create a conductive network directly from the electronics to the aluminum end caps in contact with the cooler flowing water.

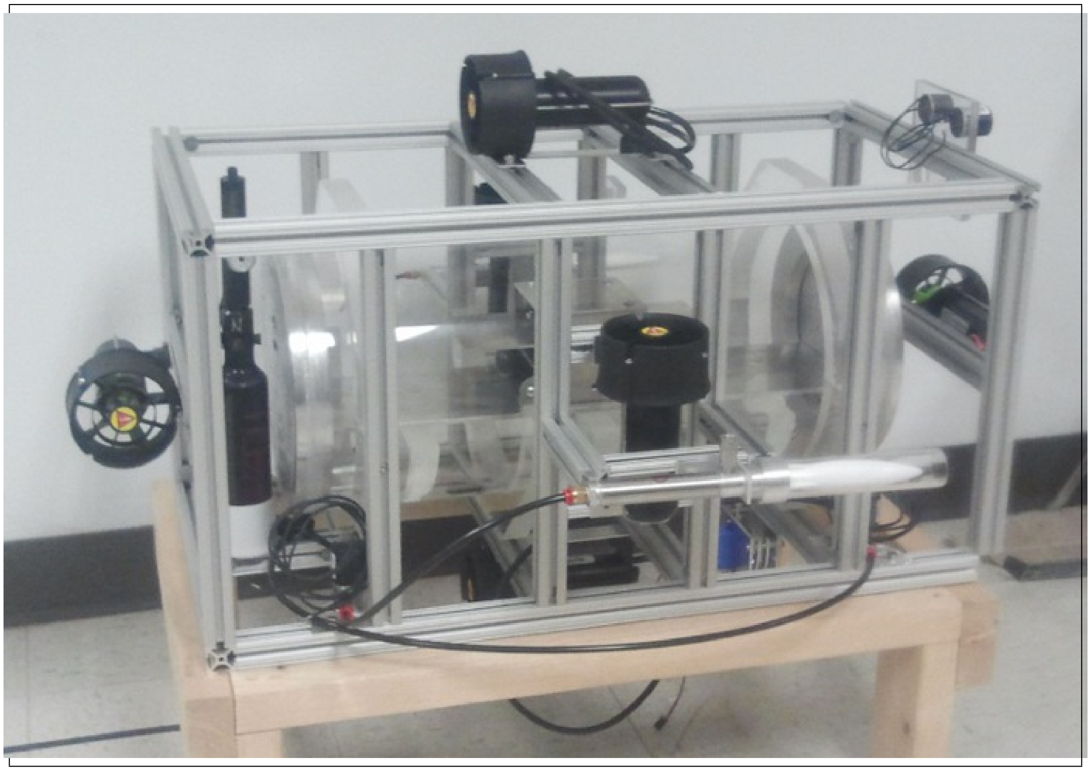


Figure 2: Angle view of AUV

3.2.3 Electronics

Main Controller Unit (MCU) Regarded as the "brain" of the AUV, the main controller unit coordinates the multiple tasks required by the competition. It communicates with the guidance system and received input from the computer vision system to make sure that tasks are completed. It monitors the vehicle state in the Mission Control program.

Components

- 1x Beagleboard-xM

Computer Vision (CV) Regarded as the "eyes" of the AUV, the Computer Vision module provides the Main Controller with information regarding the navigation of the AUV and the tasks in the obstacle course.

Components

- 2x Logitech C615 Webcam
- 2x USB 2 Serial Converter

Guidance System Regarded as the "senses" of the AUV, the guidance system monitors the vehicles orientation, depth, acceleration and reports it to the various subsystems of the AUV. It also controls the thrusters by using the on-board pulse width modulator.

Components

- 1x Arduino Uno Board
- 1x IMCL Low Cost Submersible Pressure Sensor
- 1x Phidget Spatial 3 / 3 / 3
- 4x Sensortech SQ26-01 Hydrophone

3.2.4 Electrical System

Voltage Regulator Board A DC to DC converter that converts the 14.8V battery input to 5V or 3.3V for the Beagleboard and the Arduino board.

Components

- 1x LMZ12002 switching voltage regulator
- 1x LMLM3150 switching voltage regulator
- 1x Voltage Regulator Board (in-house)

Power Supplies Two 14.8V Lithium-ion polymer batteries provide the power for the entire electrical system of the AUV.

Components

- 2x Lithium-ion polymer batteries

Kill Switch A waterproof kill switch will be implemented to shut off the power supply to the AUV upon actuation, thus causing the vehicle to naturally rise to the surface in case of a program error or need to restart the program. A waterproof (marine-grade) mechanical switch might be used and fixed to a corner of the frame–away from the thrusters and mechanical subsystems. However, this has yet to be finalized, and further research on this component needs to be done and a kill switch implemented (either electrical or mechanical) prior to any underwater general maneuverability tests.

3.2.5 Mechanical Subsystems

Vehicle Propulsion Thrusters will be strategically located on the AUV to provide general maneuverability along and about each axis. These thrusters will be able to intake PWM signal, thus providing the ability to implement stability, velocity, position, orientation, and depth control algorithms with PID closed-loop controllers which will use the individual thrusters or pairs of thrusters as the sole outputs in order to obtain a desired position, orientation, speed, etc. of the vehicle. The thrusters are highly energy dense, simple to implement, and easy to mount to the vehicle. Thrusters have also unanimously selected for use by prior competing Robosub teams for these reasons. Components

-

Compressed Air Distribution System The pneumatic system will initiate mechanical motion in the grasp/release mechanism and the torpedo launcher subsystems. The current/final design of the compressed air distribution system uses a compact compressed air paintball tank, a tank regulator, a secondary/low pressure regulator, submersible solenoid valves, and a compact network of nylon tubing/gas lines. Compressed air was recently selected as the new gas of choice following further extensive research which indicated its greater reliability, ease of pressure control, increased refill availability, and equally compact tank sizes. The compressed air tank is directly connected to tank regulator, which reduces the air pressure down to about 850 psi. The custom low pressure regulator is directly attached to the outlet of the tank regular, thus reducing the air pressure to the desired operational pressure of about 100 psi. This outlet pressure can be easily adjusted to the optimal value by simply rotating a hexagonal set screw/knob. The submersible solenoid valves enable the flow of the compressed air through the gas lines and to the air cylinders of the respective mechanical systems (more specifically, the torpedo launchers and the grasp/release mechanism) to be controlled via electrical actuation, and thus a microcontroller.
Components

- No items

Grasp / Release Mechanism The grasp/release mechanism will serve to grasp the rescue object (i.e. a laurel wreath) above the first located pinger during the competition and then release the object once the vehicle has located the second pinger and surfaced. It will be connected to the gas distribution system and will grasp or release the object upon actuation of the respective pair of solenoid valves via the microcontroller. A built-in spring inside the single-acting air cylinder which will actuate the grasping motion will provide a smooth grasping motion, as well as a quick release of the object after the compressed air has been purged from the device via a one-way check valve upon command.
Components

- No items

Marker Dropper The marker dropper subsystem was inherited from FAMU-FSU 2010 Robosub team. An aluminum housing containing parabolic channels will support the markers (miniature stainless steel balls) and a servomotor, which will be used to control the release of these markers. The mechanism will be placed towards the front of the vehicle, behind the cameras. The servo arm will prevent the markers from dropping prematurely, and upon actuation via the microcontroller, it will rotate in either direction, thus releasing each marker individually upon command. This mechanism will serve to fulfill the task that requires the AUV to drop markers in designated drop-in bins.
Components

- No items

Torpedo Launcher The torpedo launchers will utilize mechanical motion initiated by a pneumatic system to perform a desired task. The key components of the system are the 3D-printed ABS plastic torpedo with an embedded stainless steel rod and neodymium magnet (to obtain the desired balance, density, and magnetic force), the cylindrical acrylic barrel, the double-acting air cylinder (i.e. no spring return), the air cylinder mount, the one-way check valve at the exhaust of the air cylinder, and the cylindrical piston attachment containing a complementary embedded neodymium magnet to hold the torpedo in the proper position during the mission prior to launch. Successfully completing the task calls for two torpedoes; therefore, identical torpedo launchers have been developed and are located on opposite sides of the vehicle for symmetry and stability. Each cylindrical acrylic barrel has a diameter slightly larger than the maximum diameter of the torpedoes, thus providing a low-friction guide to increase launch accuracy and precision. Each torpedo launcher will be controlled by two independent solenoid valves, allowing for the torpedoes to be fired individually.
Components

- No items

3.3 Performance Assessment

Req. ID	Capability Definition	Section
RC2.2.1	The vehicle must operate autonomously (no external / remote control).	3.1: Mission Control
RC2.2.2	The AUV, and any parts connected to it, must submerge and remain submerged once the vehicle has embarked on its mission.	3.5: Hull / Frame
RC2.2.3	All electronics must be preserved in a waterproof environment.	3.5: Hull / Frame
RC2.2.4	The AUV must have a remote kill switch (in case of an emergency) which, when activated, causes the vehicle to rise to the surface of the water.	Work in progress.
RC2.2.6	The device should have onboard subsystems which enable the AUV to successfully complete the course tasks.	3.1: Mission Control
RC2.2.6.1	[Gate] The AUV should pass through the gate.	3.7: Vehicle Propulsion
RC2.2.6.2	[Buoys] The AUV should strike two of the three buoys (Red, Green, and Yellow) in the given order.	3.8.4:Torpedo Launcher
RC2.2.6.3	[Box Crossing] The AUV should navigate through a box defined by PVC and imaginary sides (i.e. not all sides have physical boundaries).	3.3: Computer Vision
RC2.2.6.4	[Drop-in-bin] The AUV should drop two markers in the correct bins (four total bins). Each bin will have a distinct symbol or object which will need to be sensed and deciphered.	3.8.3: Marker Dropper
RC2.2.6.5	[Torpedo] The AUV will need to fire two torpedoes (at a "safe" speed) through certain cut-outs of a PVC structure.	3.8.4: Torpedo Launcher
RC2.2.6.6	[Surface-and-Recover] Guided by a specific acoustic ping signal, the AUV must position itself under a designated octagonal region on the surface of the water. After the vehicle has completely surfaced within this designated region, the AUV must successfully recover a specified object. Thereafter, the AUV must navigate to the second octagon. After the vehicle has completely surfaced within the second designated octagonal region, the AUV must release the object.	3.4.5: Hydrophone

Table 1: Performance Assessment

3.4 Design Process

Major decisions made

-
-
-
-

3.5 Overall Risk Assessment

Development of AUV's components is moving on schedule. By far the largest risk to the project is that the engineers will graduate before it is completete. All of the engineers working on the product aim to graduate this semester and, assuming that all classes are passed, will do so. Therefore, to mitigate this risk, development of the AUV has been placed on a rapid schedule with most components aimed to be integrated into the complete system by the end of spring break. This includes: the hydrophone mounting array, mounting the pressure transducer, completed (or well progressed) programming of the thrusters, workable computer vision system, and fully functionally heading / orientation detection with the use of the intertial measurement unit.

4 Design of Major Components

The design of most major components complete, this section has been updated to include recent pictures of the components as well as code samples. Longer code segments will be placed in the appendix.

This section will demonstrate the status of each component, the design process used, the ability of the component to meet the requirements, and the current risks.

4.1 Mission Control

4.1.1 BeagleBoard-xM

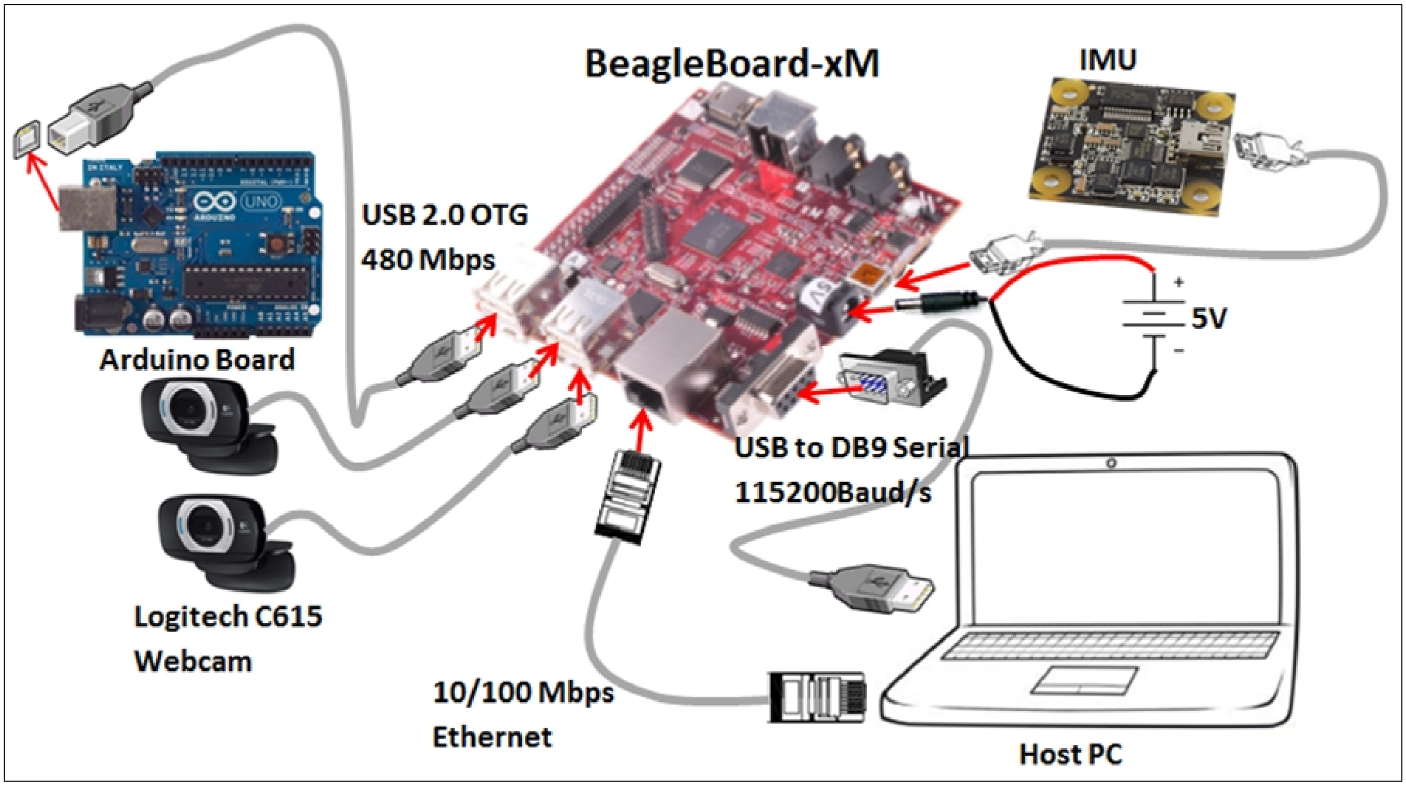


Figure 3: MCU Interface and Connection Diagram

The complexity of the mission requires the AUV to be capable of coordinating multiple tasks concurrently and process large amount of data from all the sensors. As a mission controller and the main "brain" of the AUV, the main control unit must be composed of a powerful microprocessor with low power consumption. In this case, our team decided to use the Beagleboard-xM. The main features of the Beagleboard-xM are listed in Appendix. It contains a TI cortex A8 1GHz processor, an 800MHz DSP, and 512MB DDR memory. And the power consumption is less than 15 Watts

The microprocessor, memory and the DSP is located at the center of the Beagleboard. The microprocessor is a 1GHz ARM Cortex-A8 processor and an 800MHz C64x+ digital signal processor which provide more than 2,000 Dhrystone MIPS and up to 20 Million polygons per sec graphic. The 1GHz ARM processor will be used to perform the mission control, calculation of data from sensors, and sending out commands to components such as thrusters. The C64x+ digital signal processor is capable of HD video and image processing and will handle all the image processing tasks from two onboard cameras on our AUV.

The Beagleboard-xM will be run on Ängstrom Linux, with Linux kernel 2.6.29. Ängstrom is a complete Linux distribution: includes the kernel, a base file system, basic tools, and a package manager to install software from a repository. It is optimized for low-power microcontrollers like the Beagleboard-xM. To run our programs on Ängstrm Linux on Beagleboard, a toolchain is needed. OpenEmbeded and CodeSourcery provides the cross compiling toolchains for Beagleboard. We will use the CodeSourcery Lite Edition in our project since it is free, easy to configure, and provides all the functions needed. The CodeSourcery cross compiling environment is installed on the host computer.

4.2 Electrical System

The electrical system of the AUV contains two lithium-ion polymer batteries which supply the power for the entire system, a voltage regulator board which step down the 28.6V voltage to desired voltages for each component such as 5V for the

Beagleboard and 9V for the Arduino boards.

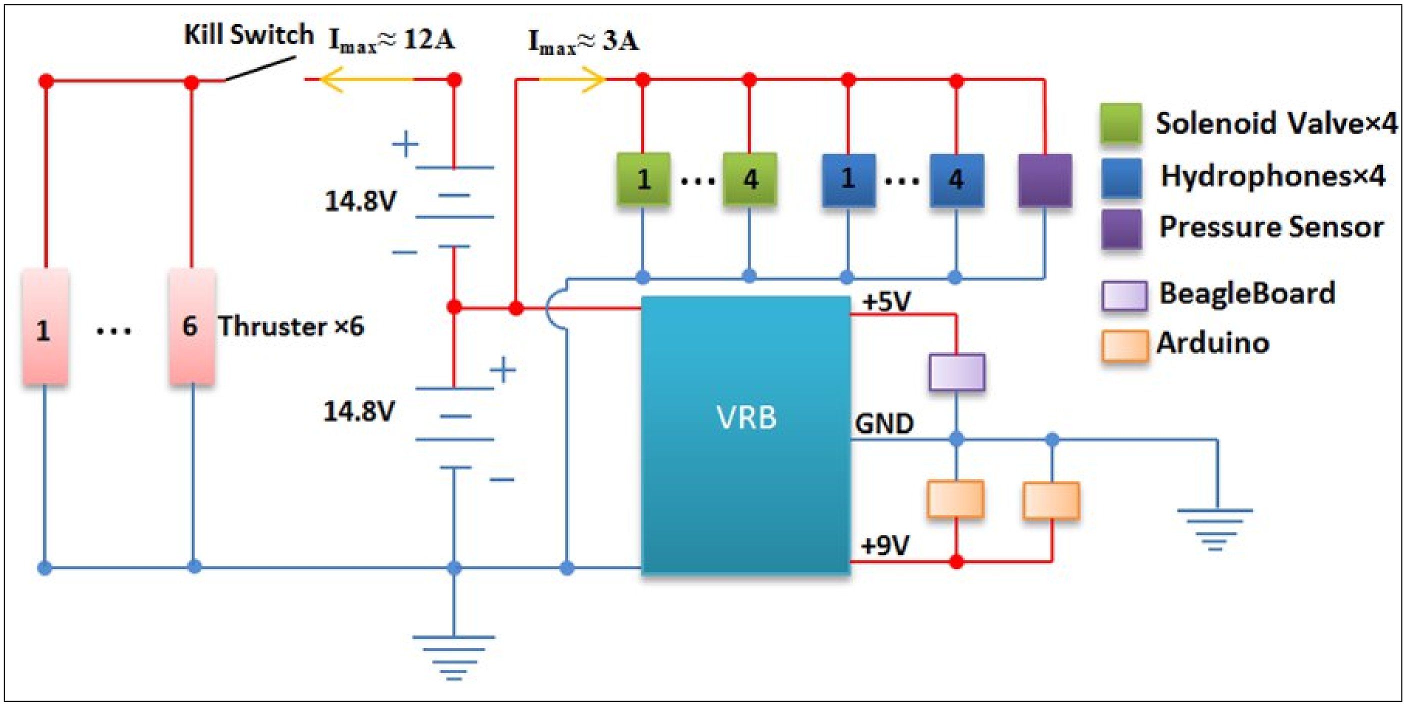


Figure 4: Electrical System Diagram

The AUV will have four Solenoid Valves for one grasper, one marker dropper and two torpedo launchers, four hydrophones, and one pressure sensor. These components will be able to operate at around 14V voltage. The voltage regulator board will Step down from +14.8V DC to +5V and +9V DC output. The +5V output is for the main controller: Beagleboard XM. And the 9V will be used to power the Arduino board. There is an onboard voltage regulator on the Arduino board which produces +3.3V and therefore can be used to power the IMU. There six thrusters in the system will be powered by three 2 - channel L298 Motor Driver. The motor driver takes maximum voltage of 50V. The voltages coming out of the motor driver will be less than 28.6V depending on the thruster loads. It is tested that at 19V input to the motor driver, the output voltage to the thrusters is measured to be about 8.3V at maximum speed (ignoring drag force). The detailed connections between thrusters and motor drivers are shown in the propulsion system section.

4.2.1 Power Supply

The AUV will be powered by two identical Li-ion polymer batteries. Each is rated 14.8V and the capacity is 20Ah or 296 Wh. The maximum discharging rate of the battery is 30Amps. Each battery requires ten hours to be fully charged.

Figure 5: Li-Ion Battery

Battery Characteristics

- Voltage: (14.8V working, 16.8V peak, 11.0V cut-off)
- Capacity: (20Ah / 296 Wh)
- Connector: 6.0" 14AWG Standard male Tamiya connector

- Max Discharging Rate: 30A limited by PCM
- Dimensions: 166mm x 125mm x 54mm
- Weight: 1.81kg

The electrical components on the AUV are estimated to draw a maximum current of 19A together. Therefore, with the two batteries, the AUV can operate continuously for about two hour. During the competition, the AUV is required to operate for less than fifteen minutes. Therefore, these two batteries can provide more than enough power for our AUV.

The table below shows the maximum current estimation. Note that the actual current (or average total current) is much less than the maximum value.

Component	**Max Current (A)**
6 Thrusters	$\simeq$ 15A
Beagleboard-xM	$\simeq$ 1.5A
2 Arduino Boards	$\simeq$ 1.5A
Hydrophone, IMU, Pressure Sensor	$\simeq$ 1A
Total	$\simeq$ 19A

Table 2: Maximum Current Summary

4.2.2 Voltage Regulator Board

The AUV has different electrical components which require different DC input voltages and currents. Therefore, we propose of designing a customized voltage regulation board which take in the power from the batteries (DC) and regulate correct DC output voltage for each component. For instance, the voltage regulator board should provide a steady +5V voltage to the Beagleboard-XM, and +9V to the Arduino board. The regulator board is designed and simulated using the WEBENCH Power Architect from TI. The block diagram and detailed regulator circuit diagram are shown below.

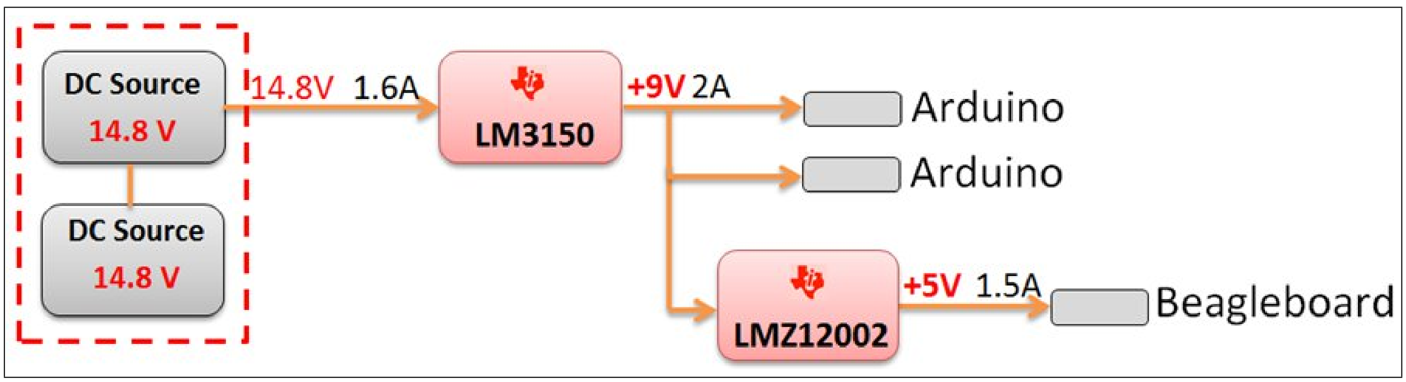


Figure 6: Block diagram of the Voltage Regulator Board

Voltage Regulator Board Summary

- Total System Efficiency: 96.715
- Total System Cost: $27.18
- Total System Power Dissipation: 0.8661mW

These power regulators can achieve very high efficiency which reaches to about 96%. This is desired in our AUV design because we would like to minimize the heat dissipation in the water tight hull. Since the power dissipation is less than 1W, no heat sink will be needed for the VRB.

4.3 Software System

4.3.1 Operating System

The complexity of the AUV requires the software to be able to provide enough functions, libraries, and kernel modules for communications and controls. The AUV will utilize the Angstrom Linux operating system running 2.6.32 kernel. The Angstrom Linux is optimized OS especially for embedded system. It has wide community support on Beagleboard which is the AUVs computer. The basic structure of the software hierarchy is shown in the picture below.

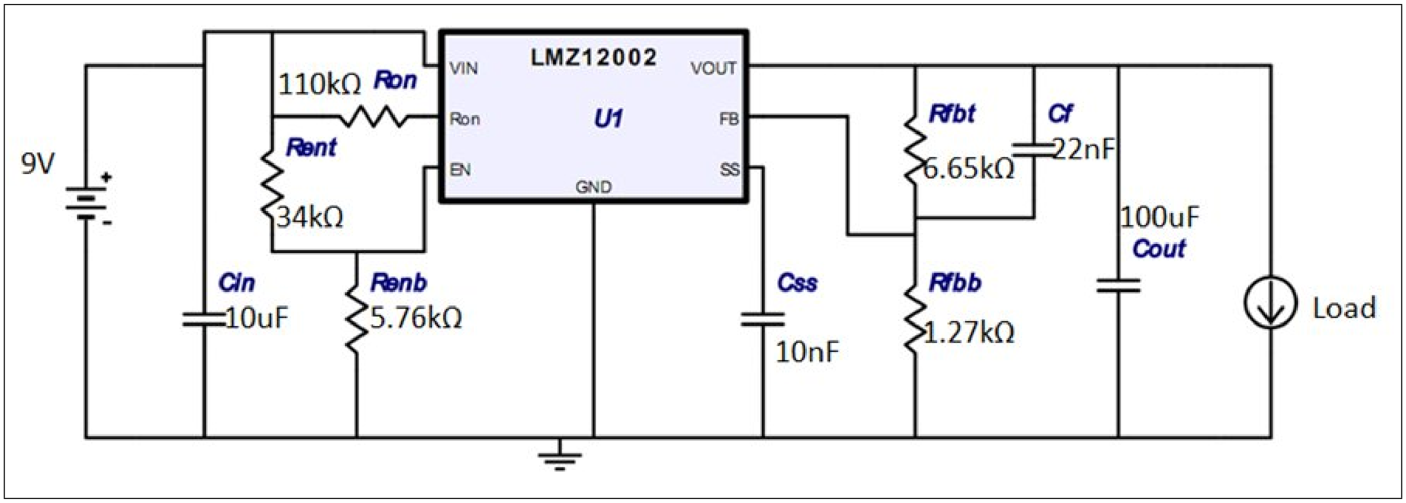


Figure 7: Circuit diagram of LMZ12002 Voltage Regulator

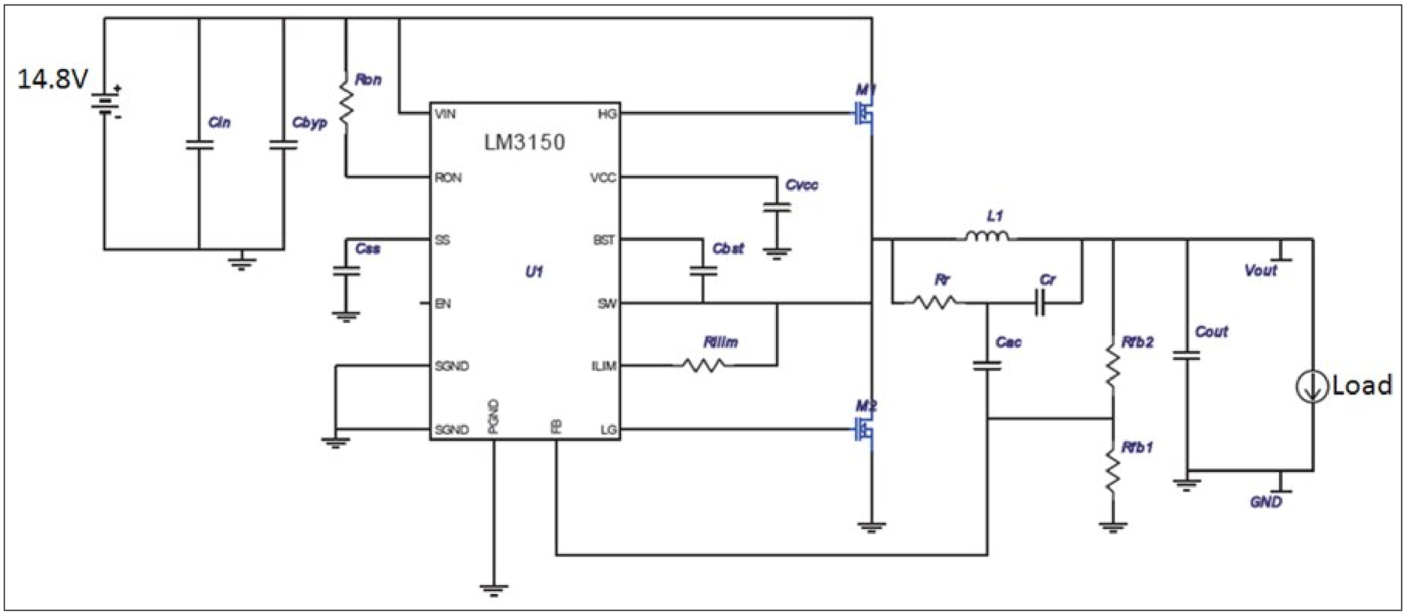


Figure 8: Circuit diagram of LM3150 Voltage Regulator

4.3.2 Software Interface and Design

All programs shall be written in C language. The AUV will have a master program which is the mission controller. This is the first program that the AUV will run once turned on. It will run in user space as a single process. Immediately after the mission controller process starts, it will spawn multiple independent child processes for other software modules including the two OpenCV modules for cameras, IMU module, pressure sensor module, and the module for hydrophones. The cameras, IMU, pressure sensor and hydrophones will directly communicate with the master program. Since these modules are spawned within the master program process, Linux inter process communication can be utilized. Different Linux inter process communication methods include: unidirectional pipe, FIFO, message queue, bidirectional pipe, shared memory, and sockets. Since, the communication between these sensors and the master program are unidirectional, our software communication interface between the mission control process and any of the sensor module will be the unidirectional pipe which is easier to implement compared to other methods. The graph illustration is shown below.

The Cam_B module stands for the bottom camera whose main task is to detect the path, and the Cam_F module is the front facing camera for object detection. Programs running on the Arduino board include PWM controller which controls the speed and direction of the thrusters, and the program to activate marker dropper, torpedo launcher and also the grasper. The communication between modules on the Beagleboard uses unidirectional pipes based on modifying Linux file descriptors. Example calls are `popen()` and `pclose()` functions which automatically fork or close a child process and setup the pipes. The communication between the Beagleboard and the Arduino board uses Linux USB serial communication. The Arduino board will appear as a device named `/dev/ttyACM0` in the OS. Therefore, by using Linux `open()`, and `close()` command, we can set up communication between the master mission control program and the programs on the Arduino board. The

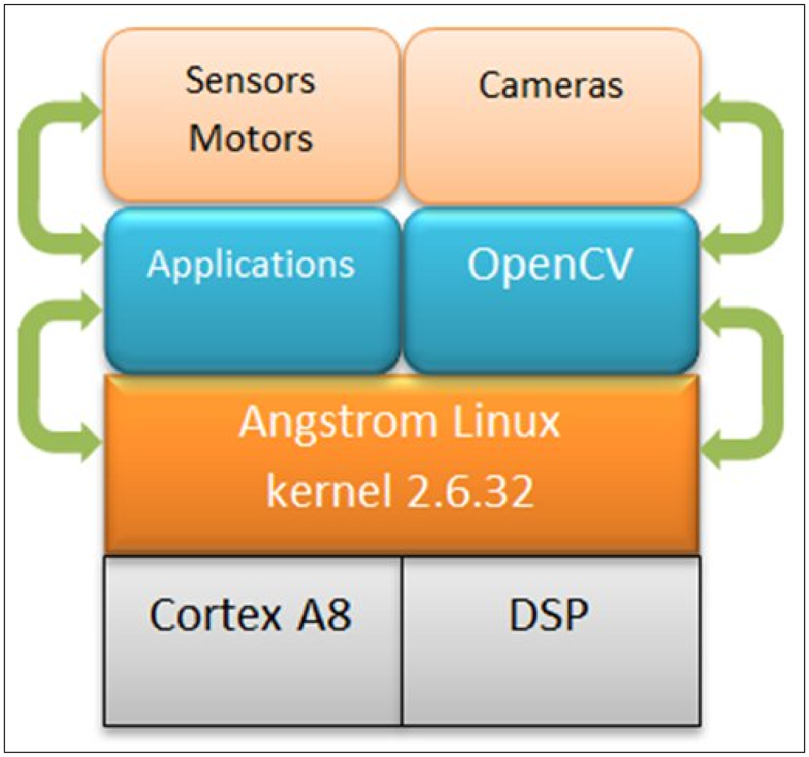


Figure 9: Software hierarchy of AUV

programs are attached in the Appendix.

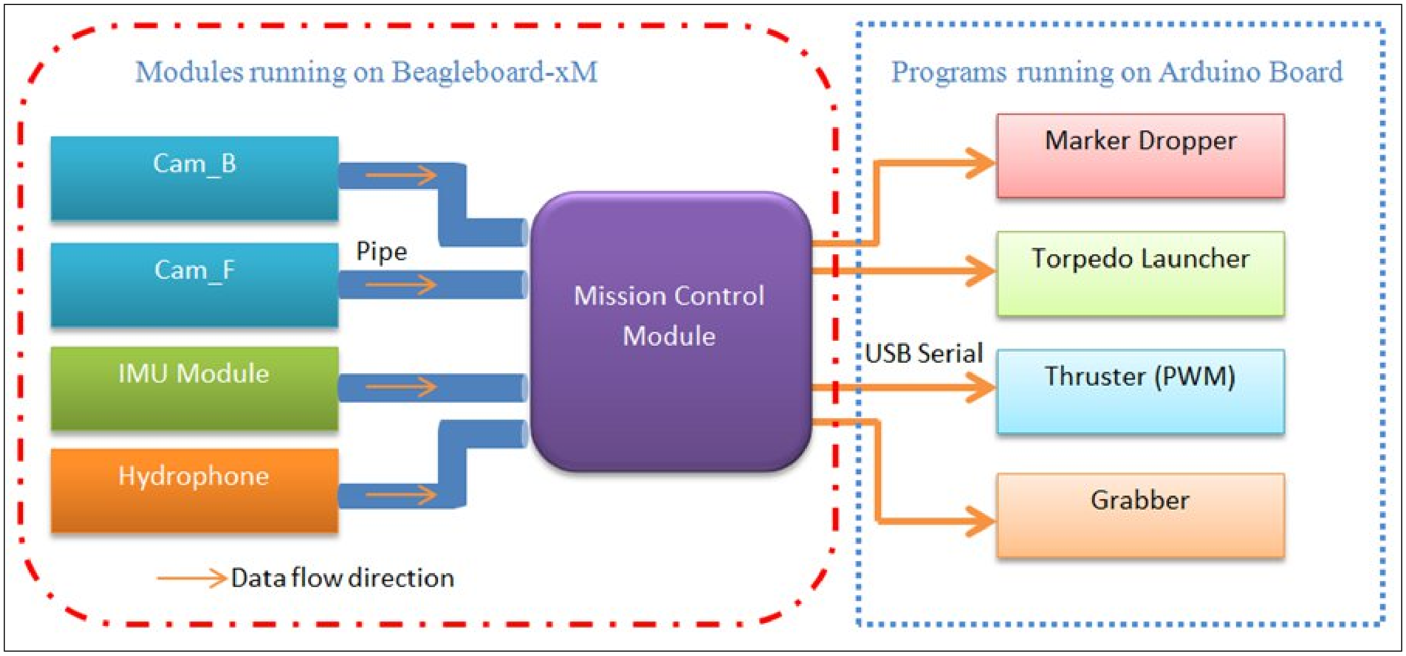


Figure 10: Illustration of Software Communication Interface and Structure

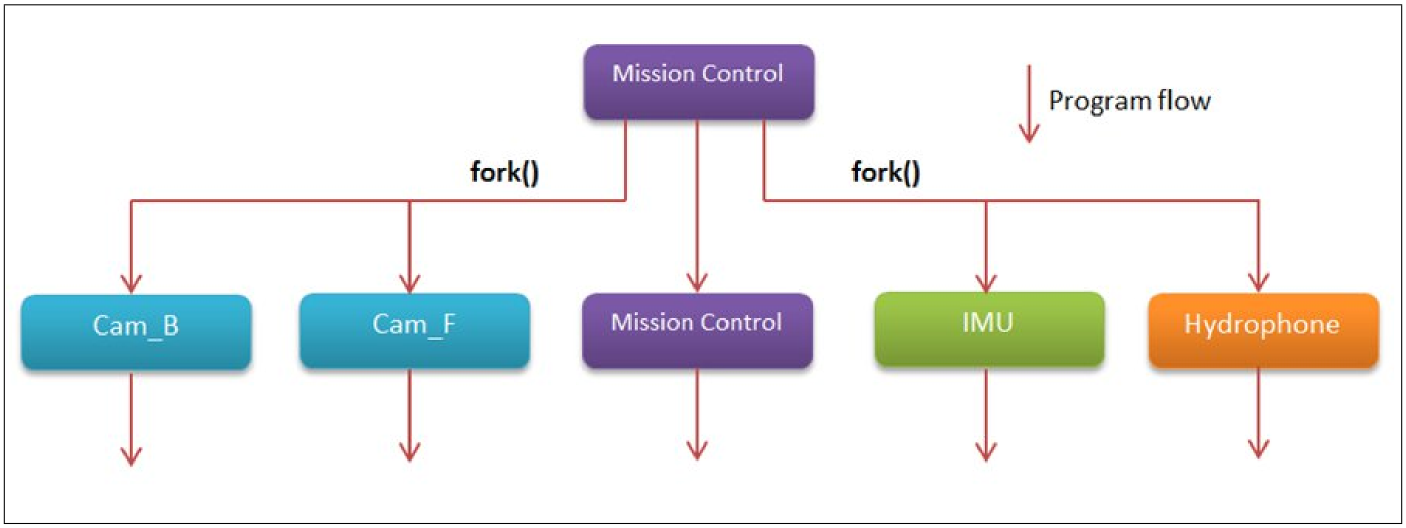


Figure 11: Mission Control Program Form Illustration

4.4 Mission Control

During the Competition, the AUV is required to complete a series of missions. As a summary of the missions, the AUV needs to pass an underwater gate first, strike two of the three buoys, navigate through a box (obstacle course), drop two markers into specified bins, fire two torpedoes, and finally grab and then release certain object after successful detected the position of the pinger. In order to complete these tasks successfully, a software mission control system is necessary. After the AUV passed through the gate, the AUV can complete the tasks in any order. However, to simplify and avoid some control overhead, we will design our AUV to complete the tasks in the order stated above. The mission control system will control the states of the AUV during each mission, coordinate all components in the system, and handle the proper transitions between each task. The mission control system should also have error detection and self-correction capability in the software layer to proper handle as many exceptions during the operation as possible. For example, if the AUV cannot find the underwater gate after a timeout, the mission control system should not only adjust its direction, but also its height and orientation to find the correct position of the gate as soon as possible. And after a second timeout, the mission control system should expand its search area. However, the exact implementation will be very complicated due to the complexity of the system and the proper cooperation required among all the sensors and motors. In this detailed system design review, some algorithms will be presented as well as the state diagram, the flow chart.

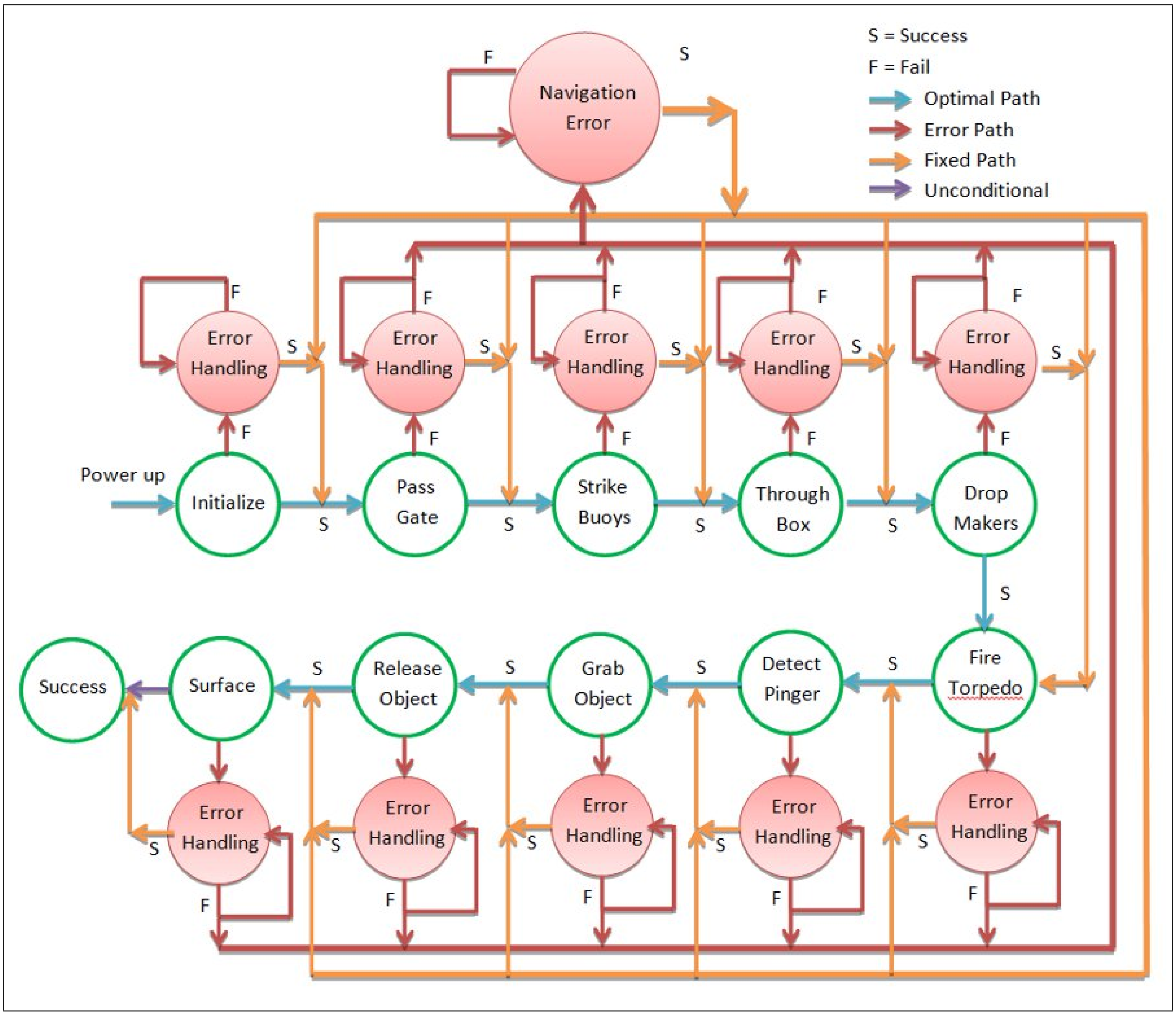


Figure 12: Mission Control State Diagram

As shown in the diagram, each state has an own error handling module, which deals with all potential errors other than the navigation error. These errors include communication error between the mission control and sensors, operation timed out, memory operation violation, data misinterpretation, and etc. However, navigation error can happen at all stages except the initialization stage, the surface stage and the success stage. Therefore, when navigation error occurs, the mission control will call the Navigation Error module to get the correct path tracking and object detection.

4.4.1 Algorithms

The details of the algorithm for passing through the underwater gate are presented here. It is the first and mandatory task during the competition. If the vehicle did not pass through the gate successfully, it will not be able to continue the competition. Therefore, it is necessary that we develop a good algorithm that will ensure that the vehicle can pass through the gate successfully. One challenge for the first task is to measure the distance from the vehicle to the gate, which is very helpful for accurate positioning of the vehicle. Since we do not have any equipment that could measure the distance directly, the algorithm does not rely on the knowledge of the distance between the vehicle and the gate. The algorithm assumes that the vehicle is balanced and positioned approximately at half height of the gate and its headings are arbitrary.

1. Initialization
   (a) Fork processes for different modules
   (b) Test communications between controller and modules
2. Poll input from pressure sensor and adjust vehicle to desired depth
   (a) About 3.5ft deep(3-4 feet range)
   (b) Balance the vehicle
3. Detect Gate Position
   (a) Needs to adjust the vehicle so that it is perpendicular to the gate.
   (b) Algorithm
      i. The length of the vertical segment can be assumed to be constant once the AUV is balanced. Therefore the ratio between the length of the top bar and the length of the vertical segment can be used to measure the angle between the gate and the AUV
      ii. Needs to measure 5 points. Four corners and one middle point of the rectangle.
      iii. Calculate the length of the top bar (distance between top 2 points in image, blue line below) and the vertical segment (yellow line).
      iv. If the vehicle is heading straight toward the gate, the ratio of the lengths between the top bar and the vertical segment is about 5/3. If the ratio in the image is ¡ 5/3, there is an angle between AUV and the gate. Rotate vehicle to the right by a small amount, stop the vehicle, and calculate the ratio again, if the ratio is smaller than the original, rotate left. Otherwise rotating right, until the ratio is very close to 5/3.
      v. Once the vehicle is perpendicular to the Gate, locate the middle point of the four corners in the image. Adjust the vehicle to align with the middle point by moving the vehicle left and right.
      vi. Check the angle again to make sure the vehicle is perpendicular to the gate.
      vii. Once the vehicle is appropriately adjusted, move the vehicle forward and pass through the gate.
4. Feedback of gate passing
   (a) Once four points all goes out of the image and the bottom camera detects the path segment. Then, the AUV has successfully passed through the gate.

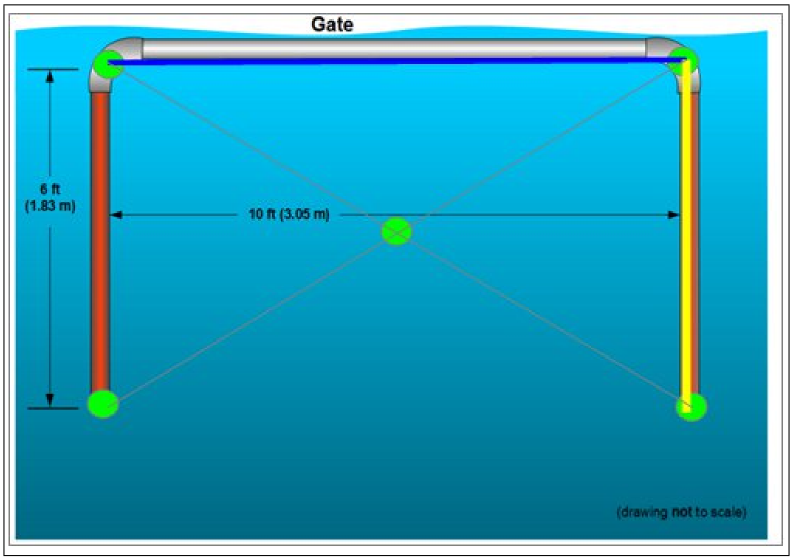


Figure 13: Illustration of the Gate Passing

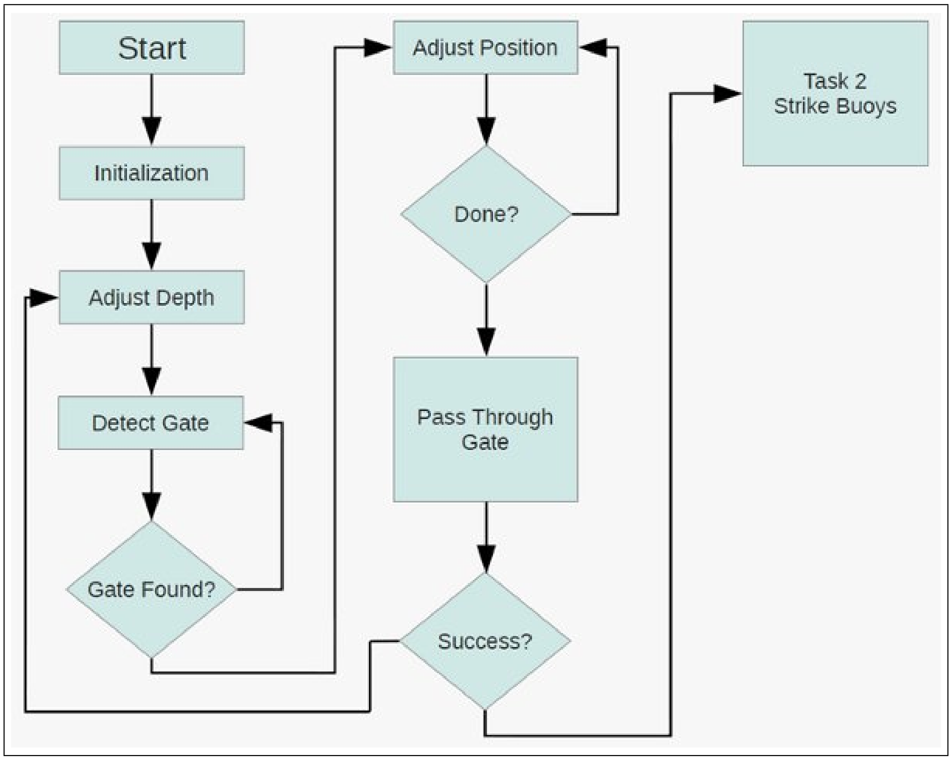


Figure 14: Gate Passing Flow Chart

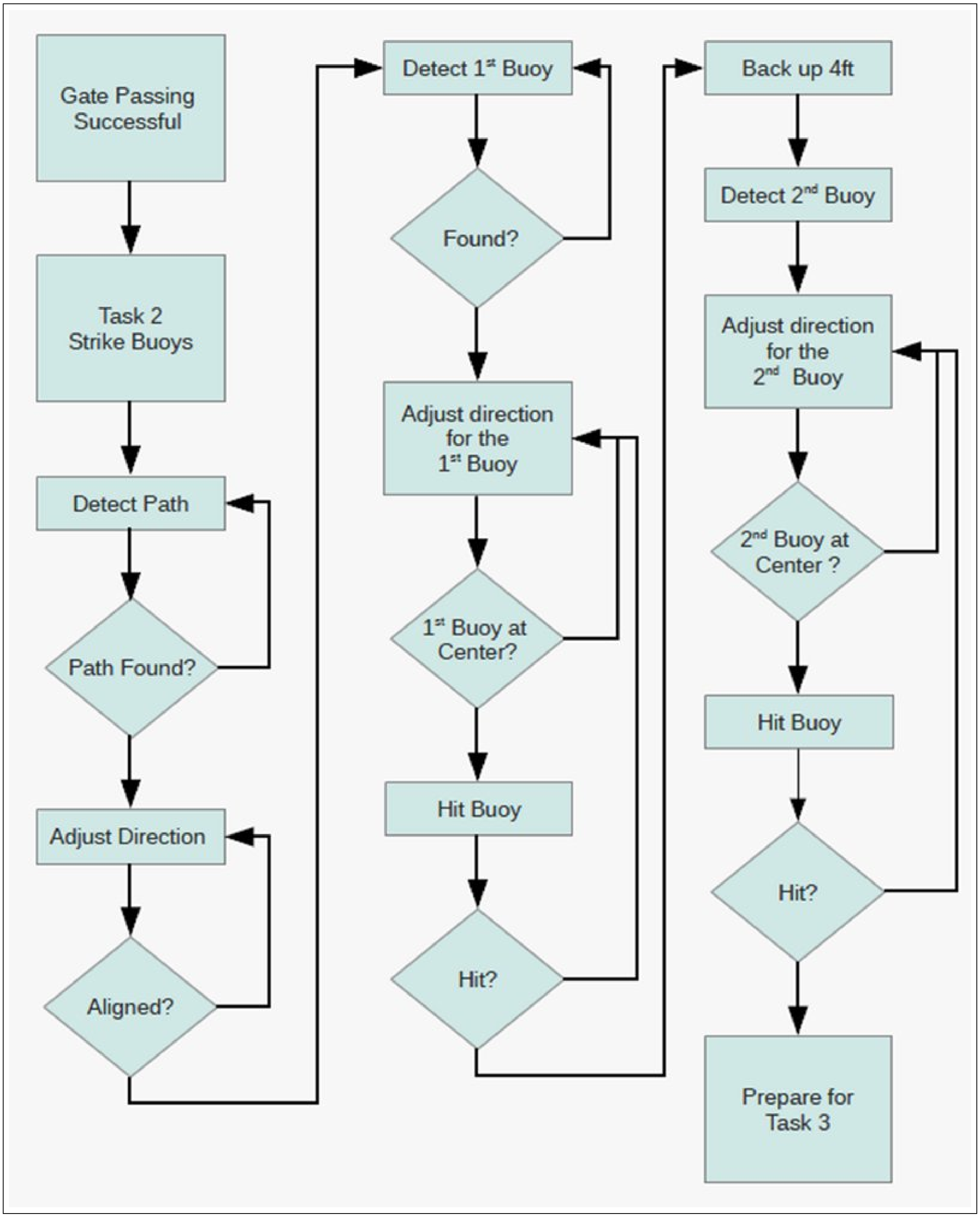


Figure 15: Strike Buoy Flow Chart

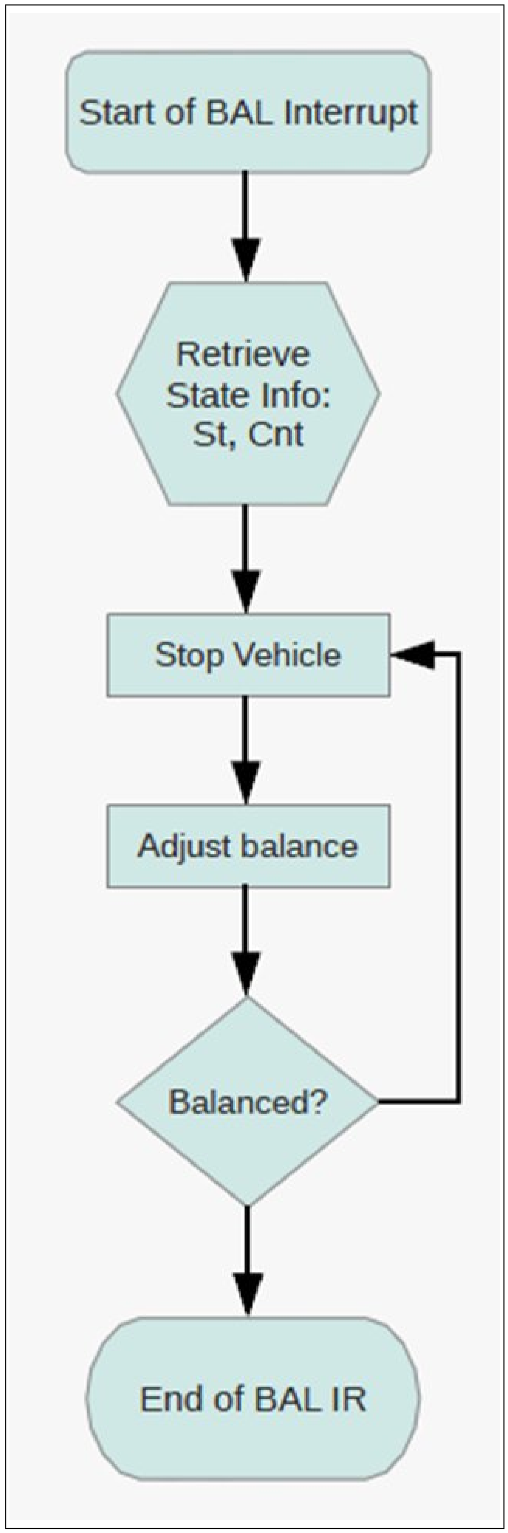


Figure 16: Balance Interrupt Flow Chart

4.5 Computer Vision

4.5.1 Main Module

The Computer Vision module serves as the "eyes" of the RoboSub. The module is responsible for task management and navigation through the obstacle course. The Computer Vision module consists of three systems: General Image Processing, Navigation System and Task Management System.

The Computer Vision (CV) module provides the AUV with path and task information. The OpenCV library takes care of the image processing on the BeagleBoard xM while two Logitech C615 cameras provide the module with the image feed.

Requirements

- Identify the path for guidance through the obstacle course.
- Identify the tasks in the obstacle course.

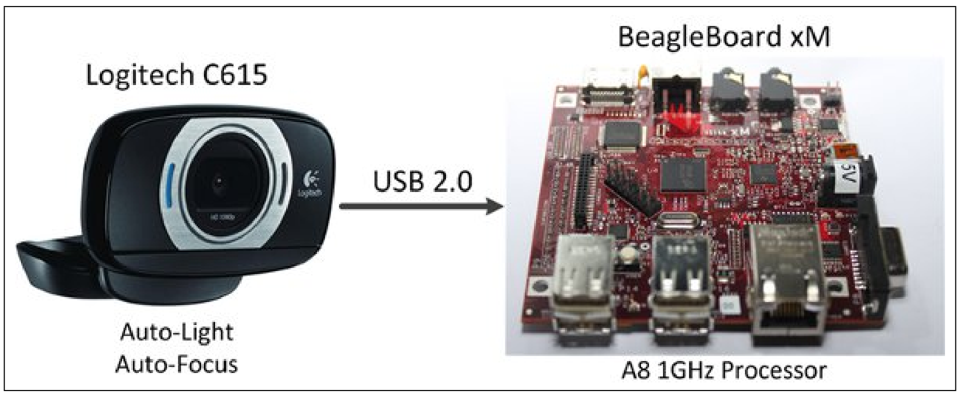


Figure 17: Hardware Setup for the Computer Vision Module

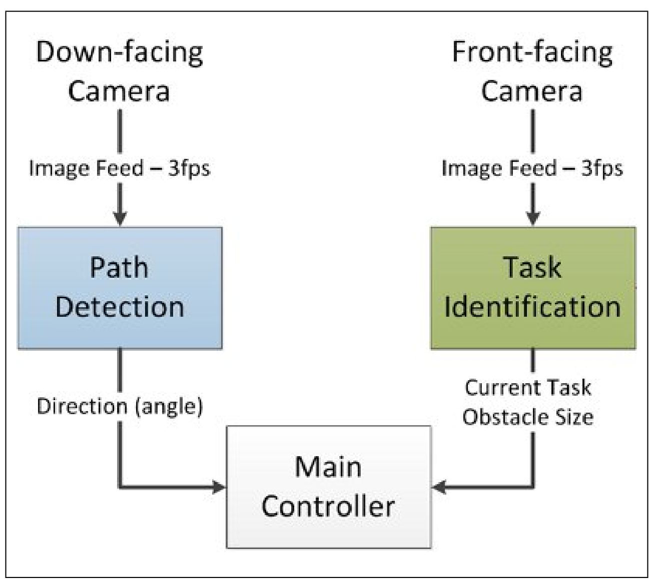


Figure 18: Software Design of the Computer Vision Module

Diagrams

Risk Assessment

4.5.2 Path Detection

Requirements

- Detects the path that will guide the AUV through the obstacle course.

Risk	Probability	Severity	Mitigation Strategy
Camera Failure	Very Low	Catastrophic	Backup Camera
Splicing Distortion	Low	Severe	Minimize Distortion
Incorrect Color Classification	Low	Moderate	Extensive Testing

Table 3: Risk Assessment

Design Parameters

- Detects orange PVC segments that represent the obstacle path.

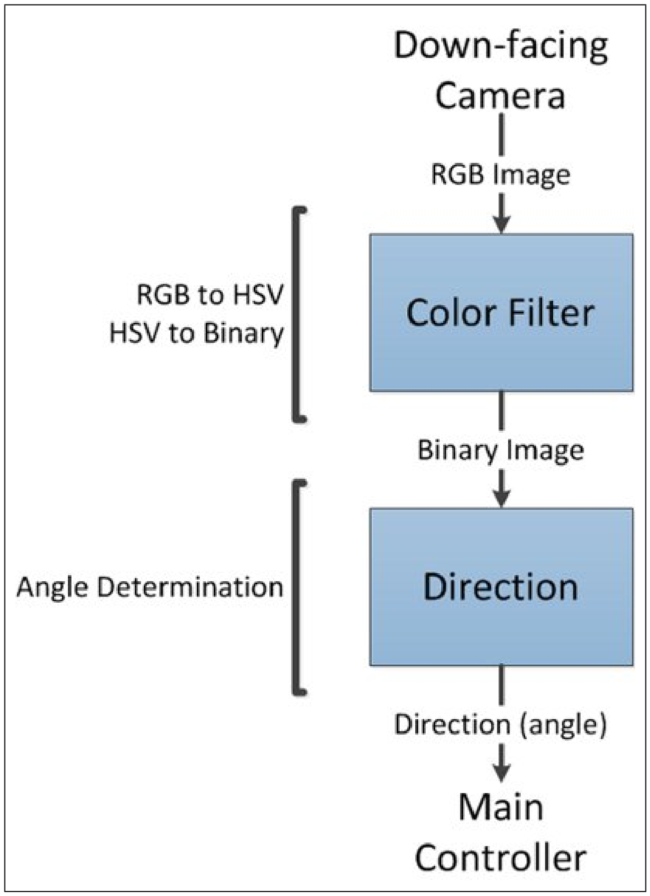


Figure 19: Software Design for the Path Detection Module

Diagrams

Images

Risk	Probability	Severity	Mitigation Strategy
Incorrect Path Detection	Low	Severe	Extensive Testing

Table 4: Risk Assessment

Risk Assessment In addition to the old risk, a NEW risk has been discovered. After some testing of the camera calibration code (see code section), the possibility of not having enough depth resolution was discovered. Since the path segments are fixed on the bottom of the pool, the camera should be able to see far enough to accurately classify the path. There is moderate probability with severe consequences. The mitigation strategy for this risk would be to buy new cameras with a better "image depth."

Code Contained in listings 3 and **??**.

Figure 20: Front View of Logitech C615 inside Camera Enclosure

4.5.3 Task Identification

Requirements

- Identify each task in the obstacle course.

Design Parameters

- Identifies shapes, sizes and colors to determine the task at hand.

Diagrams

Risk	Probability	Severity	Mitigation Strategy
Incorrect Size Determination	Low	Moderate	Extensive Testing
Incorrect Shape Detection	Low	Severe	Extensive Testing
Incorrect Task Identification	Low	Severe	Extensive Testing

Table 5: Risk Assessment

Risk Assessment

Code Pseudo-code since development of this module has yet to start, however, the color filter is also used in path detection. So the only remaining code is the shape detection.

4.5.4 Computer Vision Test Plan

Hardware During the initial phase of testing, the hardware needs to be configured and tested. The BeagleBoard and Logitech webcams are tested during this phase.

1. Beagleboard
   (a) OpenCV has to be installed and configured on the BeagleBoard. Initial testing is required to ensure that all computer vision algorithms will run smoothly on the board.
2. Logitech C615

Figure 21: Endcap View of Logitech C615 inside Camera Enclosure

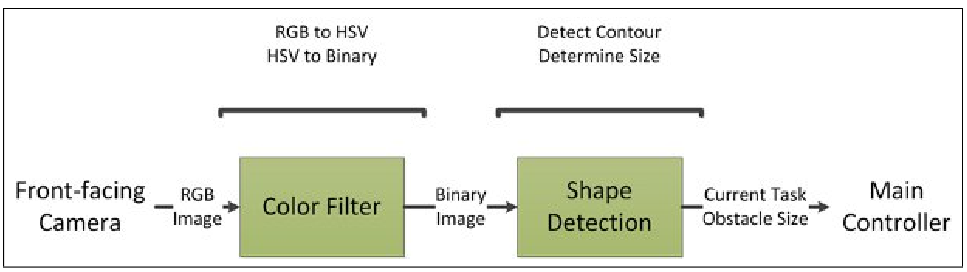


Figure 22: Software Design of the Task Identification Module

(a) The webcams have to be calibrated to ensure that the best possible images are obtained for processing. During this phase, the cameras will also be mounted in the best possible way.

3. Camera Enclosures

(a) The Mechanical Engineers will make sure that the camera enclosures are watertight; however, before submerging the enclosures with the cameras, the Computer Vision Engineer will make sure that the enclosures are indeed watertight.

Path Detection The second phase of testing will focus on path detection for the AUV. Tests will be conducted in the lab (dry-test) as well as in the pool (wet-test). During the later stages of this phase, the test environment will be modified to resemble the competition environment as much as possible.

1. Color Filter

(a) The Color Filter will undergo extensive testing in order to make it robust and ideal for the competition environment. The TRANSDEC pool (competition pool) will have sediments on the pool floor that will complicate color filtering. Furthermore, lighting greatly affects the color classification process

2. Direction

(a) The Direction module has to be optimized in order for it to produce reliable direction information. The module will also be tested for sharp turns ($> 90°$).

Task Identification Tests for the Task Identification module will be conducted during the third phase of testing. Since there are several tasks to be completed, the focus will be on the tasks that are the easiest to successfully complete. The AUV will have to pass the Gate task before attempting any others tasks. The first couple of tests will, therefore, focus on the Gate.

1. Color Filter
   (a) Unlike the Color Filter for Path Detection, this filter has to classify several colors. This module will, therefore, require additional tests to ensure the correct identification of colors.
2. Shape Detection
   (a) The module will be tested for correct shape and size identification. Size comparison tests will be important for the Torpedo task.

System Performance and Integration (SPI) This part of the test plan will occur concurrently with the aforementioned three phases of the test plan. As soon as a component passes all its tests, it moves on to SPI testing. For example, after phase two (Path Detection) is completed, the Path Detection and Navigation SPI testing will coincide with phase three (Task Identification) testing. This method of testing will ensure that newly completed modules successfully integrate with the system.

1. Path Detection and Navigation
   (a) Dry Test
   (b) Wet Test
2. Task Identification and Mission Control
   (a) Dry Test (each task)
   (b) Wet Test (each task)
3. Completed AUV (incl. Path Detection and Task Identification)

4.6 Guidance System

Successful autonomous operation requires detailed environmental awareness. Devices such as gyroscopes (to measure orientation), accelerometers (to measure acceleration), and magnetometers (to measure direction) contribute to the positional component of environmental awareness. When these devices work in tandem an inertial measurement system (IMU) results which can be used in an inertial guidance system (IGS) to precisely track vehicular heading and contribute to the vehicle's internal model of its location.

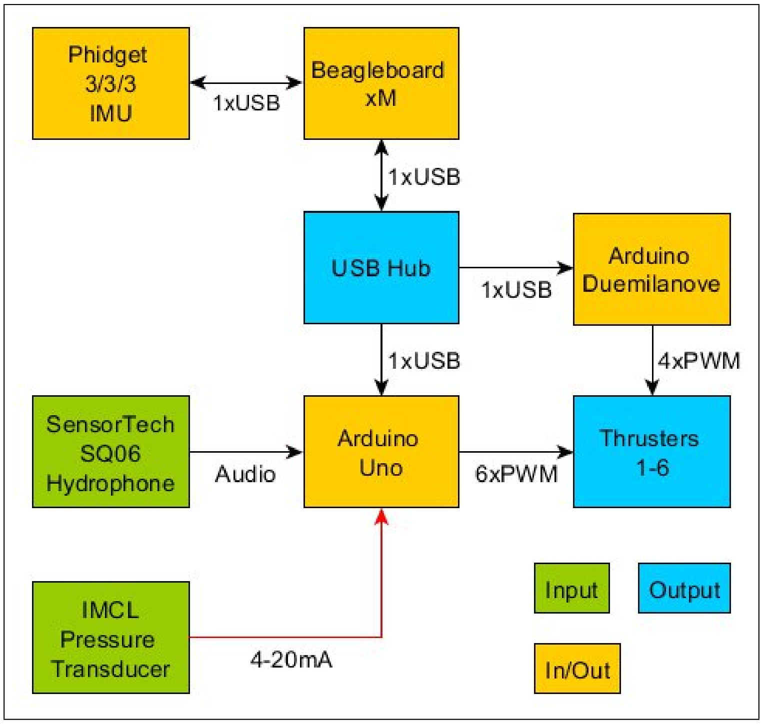


Figure 23: Guidance System Block Diagram

4.6.1 Software System

The software system is essentially a finite state machine that monitors the inputs from the sensors, consolidates them, and forwards them to the Beagleboard for post-processing. The list below contains the States used for the FSM.

1. Calibration
   (a) Take an initial reading of the environment to calibrate the sensors.
   (b) If readings failed, return to State 1; otherwise, continue to State 2.
2. Measurement
   (a) Take readings from all the attached sensors.
   (b) If readings failed, return to State 2; otherwise, continue to State 3.
3. Calculation
   (a) Use readings to determine vehicular heading, depth, and orientation.
   (b) If determined heading, depth, or orientation seems incorrect / incomplete, return to Stage 2; otherwise, continue to Stage 4.
4. Consolidation
   (a) Check the sentOnce flag
      i. If sentOnce = '0', send the data to the Beagleboard and wait for ACK.
      ii. If sentOnce = '1', send the data to the Beagleboard without waiting for ACK, clear the sentOnce flag, and continue to Stage 5.

(b) If no ACK received, set sentOnce flag to '1' and return to Stage 4. Furthermore, begin a new FSM continuing from Stage 2.

5. Termination

   (a) Clean up the FSM by removing any network connections.

   (b) If a new FSM is not already in place, create a new one continuing from Stage 2 (i.e., already calibrated).

4.6.2 Inertial Measurement Unit

Phidget 3/3/3 The Inertial Measurement Unit is paramount to correctly determining the position of the vehicle during the course of the competition. This IMU allows for 3-axis measurement of acceleration, orientation, and magnetic vectors. It is fed an initial input that is then referenced throughout the course of the competition to calculate vehicle position relative to that position. Because this IMU, left over from the previous years project, uses a microUSB connector, it will be connected to the Beagleboard and any required calculations will be performed on it. Even though the Beagleboard is performing the calculations for the IMU and not the Arduino board, it is still considered part of the guidance system.

Code Currently, the Phidget is capable of detecting angular orientation on all three axes. The X and Y axes have an angular range of $0 \leq \theta \leq 360°$ while the Z axis only requires a range of $0 \leq \theta \leq 180°$. See listing 1 to analyze the code. Continued testing is dependent on getting the submarine in the water.

Interactions with the Phidget is shown in the testing report.

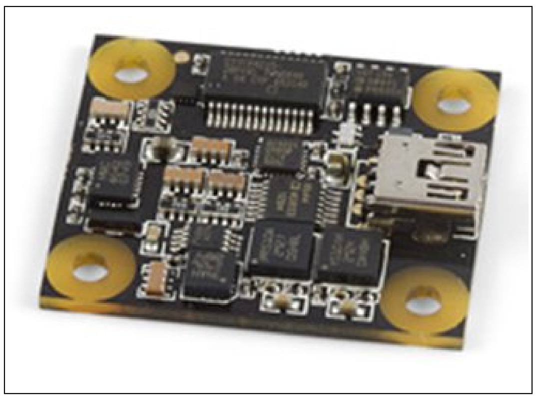


Figure 24: Phidget 3/3/3 IMU

4.6.3 Submersible Pressure Transducer

IMCL Low Cost Submersible Pressure Sensor Since the Robosub will be underwater, an additional level of measurement is required to create a comprehensive guidance system: depth. The IMCL Low Cost Submersible Pressure Sensor is designed for use in continuous submersion in liquids such as water. It has a ceramic sensor, reducing corrosion, and a stainless steel diaphragm for use in aggressive environments. This pressure transducer was selected after careful analysis of the alternatives and was the most cost effective option are this analysis was completed.

The pressure transducer will be read during Stage 1 and Stage 2 of the Software System Cycle. The measurement will be delivered by a current ranging between 4-20mA and will be sent directly to an A/D convertor on the Arduino board and quantized according to the accuracy of the model. The nominal pressure for the gauge is 0 - 10mWG (about 30 feet deep).

This component has not arrived yet (currently in shipment).

4.6.4 Hydrophone

Sensortech SQ26 Hydrophones One of the competition tasks require that the AUV be capable of detecting a pinger located in the salt-water pool. The best way to detect the sounds emitted by the frequency is by the use of an underwater microphone called a hydrophone. This hydrophone should be capable of detecting a range of frequencies. Through the use of triangulation and multiple hydrophones, the heading of the pinger can be determined by using basic geometry and trigonometry.

The Sensortech SQ26, like the IMCL Submersible Pressure Sensor, can withstand immersion in liquids such as water. However, unlike the Pressure Sensor, it can only stay underwater for a relatively short period of time a day before it has to be dried completely. With a frequency response of 1Hz to 28,000Hz (and experimentally tested up to 31kHz), this device

Figure 25: IMCL Low Cost Submersible Pressure Sensor

will be sensitive enough to locate the pinger underwater. The hydrophone will be read during Stage 1 and Stage 2 of the Software System Cycle.

The hydrophone will be mounted in the 2x2 acrylic mount, shown below. Three of the hydrophones will be used for triangulation of the acoustic pinger while the last will serve as a reference hydrophone. Once the competition fee is submitted (pending confirmation of quorum engineer attendance), the team becomes eligible to loan a transducer to test the hydrophone. Until this fee is submitted, the reference hydrophone will be used to test the other three.

The team currently possesses two hydrophones – two more are being shipped. The delay was caused by difficulties in finding a supplier that provided the now-deprecated SQ26 model used on this project. Once these arrive, testing will continue.

Figure 26: SQ26 Hydrophone

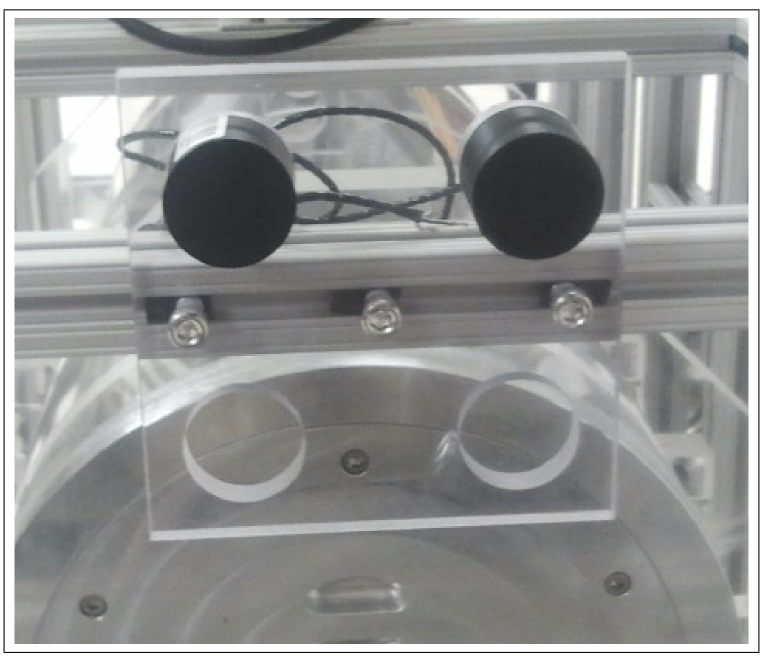


Figure 27: 2x2 Hydrophone Mount

4.7 Mechanical Systems

4.7.1 Hull / Frame

The frame is shaped like a rectangular prism and is constructed of 80 / 20 Inch Solid extruded aluminum due to its supreme versatility, ease of manufacture, and ease of assembly (although nothing is as easy as it is conveyed). Zinc anchor fasteners are used to secure each of the bars of the frame to one another at junctions, and t-slotted nuts in conjunction with $\frac{1}{4}$ inch – 20 bolts are used to attach subsystem mounts / platforms to the frame at various locations. The camera enclosure supports were modified to provide a single support located at the centerline of the front face of the vehicle that will house a camera enclosure on both the top (for the front-facing camera) and bottom (for the bottom-facing camera). A thin aluminum plate is fixed to either side of the T-slotted frame supports to provide a clean mating surface for the acrylic enclosures.

This modification provides a cleaner view for the bottom-facing camera, and yields a more sensible positioning of the cameras to assuage the potential programming difficulty that would result from the cameras being offset from the center of the vehicle, or being located at severely different locations along the frame. It should be mentioned here that while the camera enclosures were finally completed and verified to be watertight, they were clearly the one messy component of the system. Issues with water leakage that led to non-ideal measures in order to seal resulted in a less than pleasing appearance and moderate concerns about the reliability of the camera enclosures–particularly at operating depths of up to 16 ft. Thus, following brief yet extensive research, it was discovered that the same supplier of the cast acrylic tubes for both the main hull and the torpedo launcher cannons also sold acrylic display cases, including one with very similar length, width, and height dimensions, and the same 1/8" thickness.

A bold decision was made to quickly redesign the camera enclosures using these open-faced enclosures as the foundation, and utilizing all the lessons learned from the initial design in order to yield a far cleaner, far more efficient, and far more reliable set of enclosures to be permanently used on the AUV. The pre-made acrylic boxes (4" x 4" x 4") will ensure the avoidance of potential leakage along the edges of each of the facesa common and frustrating occurrence in the original custom enclosures. In addition, the dimensions of the acrylic boxes that will be used as the foundation will be so similar to the initial custom enclosures that the removable inner end caps from the original camera enclosures will not have to be remanufactured as they will be directly compatible with the revised design, and the overall design of the vehicle will not need to be adapted whatsoever. Furthermore, a non-intrusive, simple mounting technique has been derived, and the vertical location of the web cameras inside each of the enclosures will be easily modifiable by simply rotating the threaded mounting rod on which the cameras will reside. Lastly, the new camera enclosures can and will be completed the day after the acrylic boxes arrive, yielding a net turnaround of only about 10 days.

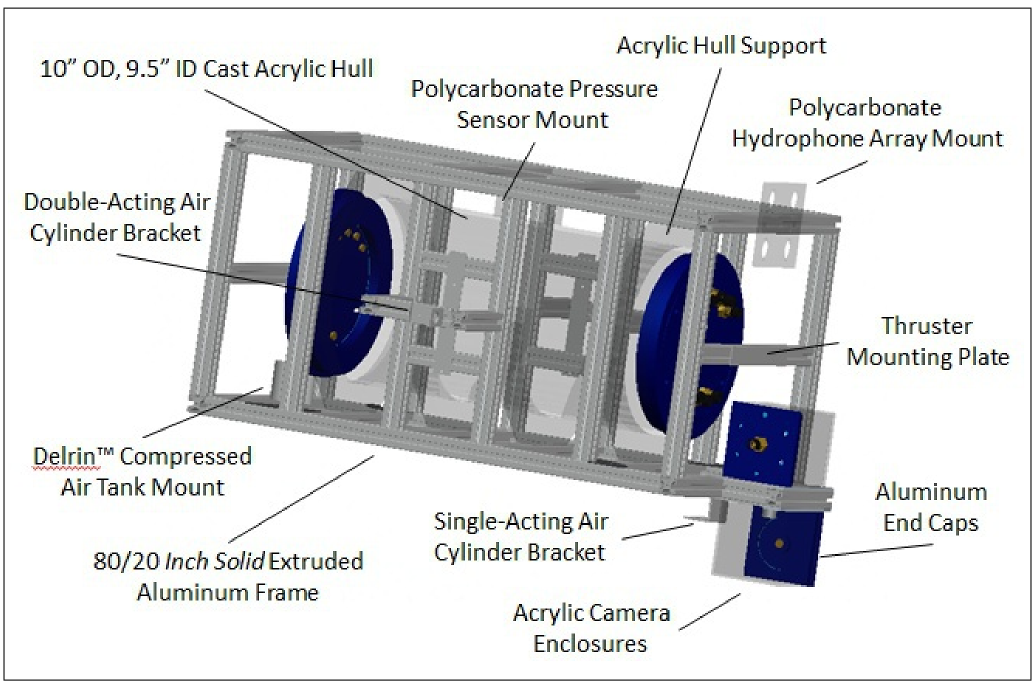


Figure 28: Close-up view of the Hull / Frame.

The hull/pressure vessel is cylindrical in form, and is made out of cast acrylic. It has an outside diameter of 10", an inside diameter of 9.5", a length of 21", and will serve to house the electronics. The $\frac{1}{4}$ inch thick walls of the acrylic tube will yield it capable of withstanding the relatively insubstantial hydrodynamic and hydrostatic pressure that it will encounter at its maximum depth of 16 ft in salt water without any measureable deflection or deformation. Furthermore, the clear acrylic material has a low density (only slightly greater than salt water), has proven applications in similar environments (e.g. used for walls of aquariums), provides enough positive buoyancy to counteract the denser surrounding components of the AUV, and will allow the electronics to be seen from the exterior of the system–an aesthetic bonus, as well as a desired feature for our ARM sponsor.

A two-piece end cap design has been implemented for both the main hull and the camera enclosures. The outside part of each end cap is rigidly attached to the inside and ends of the acrylic hull via marine-grade caulking. This fixed piece contains an open center with a lip containing six threaded holes thus resulting in a ring-like structure. Another solid aluminum, removable, circular cap is placed on top of this lip and screws into these threaded holes, with a 1/16"-thick EPDM rubber gasket serving as the intermediary at the interface between both the fixed and removable components of each end cap. This material is often used for gaskets and has the proper compliance and weather-resistant properties required from such a component. These gaskets are adhered to the lip of the fixed component of the end caps via a tremendously strong gasket adhesive developed by 3M, and compress as the removable inside part of each end cap is screwed into the complementary fixed outer aluminum ring, thus creating a secure watertight seal. This end cap design has also been implemented on the camera enclosures–although on a proportionately smaller scale, and with the outer end cap square in shape rather than circular.

Submersible SEACON All-Wet and Micro Wet-Con connectors will be attached to each of these end caps (one on each of the camera enclosure end caps, and several connectors on each of the two hull end caps). By using tri-split contact configurations for the thruster connectors, the amount of holes that need to be drilled into each end cap will be significantly reduced, resulting in more reliable water-tight seals, as well as a more compact and direct wiring scheme. The female SEACON connectors will simply screw directly into the corresponding end caps (with the aid of Loctite thread sealant to secure the female connectors in place and prevent any potential leakage through the threads). The male SEACON connectors/cables will plug directly into the female sockets, and the electrical leads of these cables will be soldered to the leads of their complementary peripheral electrical subsystem (e.g. solenoid valves, thrusters, pressure transducer, etc.). Heat shrink tubing with an adhesive inner lining, in conjunction with Plasti-Dip liquid rubber, will be used to protect and seal the exposed wires from the water intrusion.

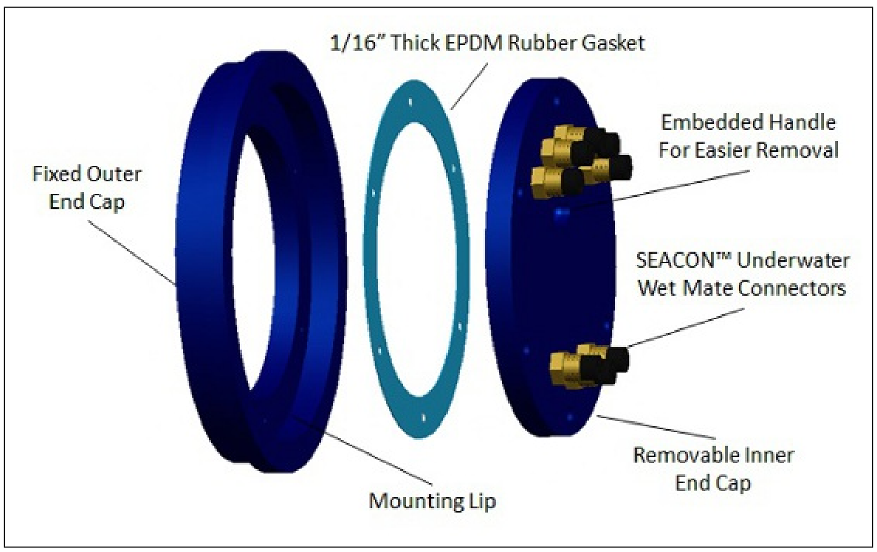


Figure 29: Close-up view of the Hull End Caps. The thickness of the end caps has been reduced in the current design in order to reduce weight and anticipatorily reduce the density of the AUV.

4.7.2 Interior Hull Design

The interior of the hull contains a 3/16"-thick aluminum sheet which will support the two 14.8 V lithium-ion batteries, the Arduino Board, Beagleboar-xM, three L298 dual h-bridge motor drivers, inertial measurement unit, and interface circuits for the solenoid valves (i.e. low-side drive BJT) and hydrophones (amplification and filtering). The use of the compact L298 motor drivers (two on each chip) ensures that only a power and ground wire (i.e. two-pin cable) will need to connect from the respective SEACON" connectors on the end caps to each of the thrusters. The aluminum 6061 platform rests on a bed of three 1"-thick, watermelon-shaped acrylic cut-outs which match the curvature of the inside of the hull. Caulking has been used to fix the curved surface of each of these cut-outs to the interior surface of the pressure vessel. These acrylic supports not only serve to create a flat resting surface for the aluminum plate, and thus the electronics and lithium-ion battery packs, but also provide insulation so that heat will be dissipated from the electronics to the exterior of the device more efficiently as it travels through the aluminum plate and aluminum end caps via conduction.

The 3/16" – thick aluminum platform is easily removable to provide convenient access to the electronics. Since the batteries are expected to generate the most heat, they will be placed at either end of the aluminum plateclosest to the aluminum end caps, and thus the external surroundings. This positioning also helps maintain symmetry and balance. Similarly, the motor

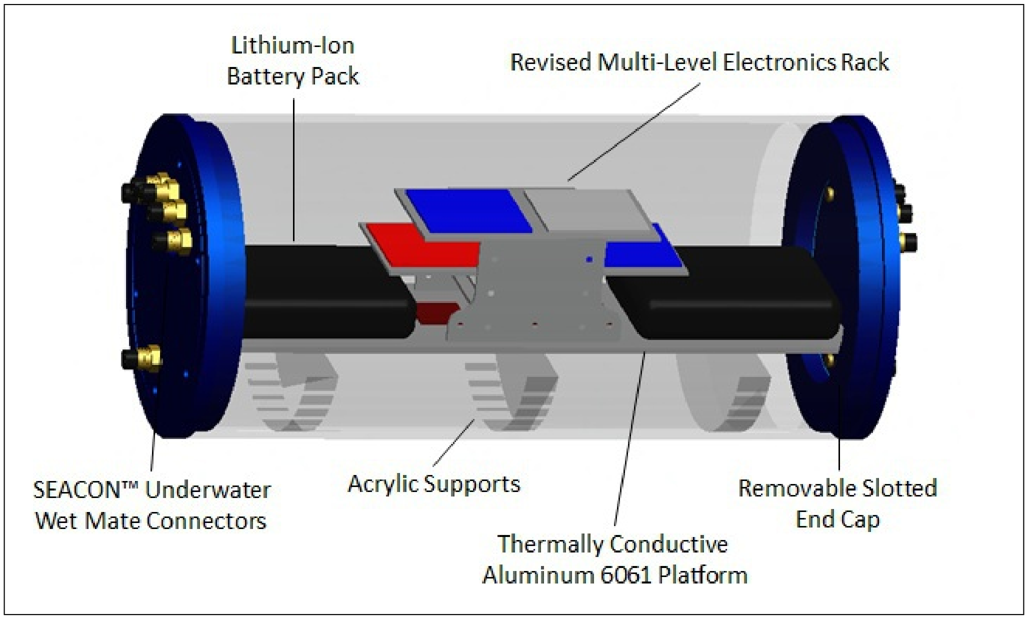


Figure 30: Close-Up View of the Interior Hull.

Figure 31: L298 Dual H-Bridge Motor Driver for the Thrusters

drivers are expected to be another significant source of heat generation, and thus will be located at the bottom of a multi-level electronics rack–directly on the aluminum platformin order to facilitate fast heat transfer to the exterior. The condensed multi-level rack will be located at the center of the hull. This structure will have three sets of horizontal racks which will support the aforementioned electronics. The circuit boards will simply screw into their respective rack, supported by spacers in order to prevent the circuit boards from coming into direct contact with the aluminum and thus risking a short circuit. The multi-level electronics rack itself will be constructed using several small 90°-angle brackets. The water jet was used (as it has been used to manufacture several other small components of the AUV) to fabricate the faces of the electronics rack. The second rack from the top (which will contain the Arduino Board, Beagleboard-xM, and IMU) will be extended laterally (i.e. partially cantilevered over each of the batteries) to accommodate the addition of the interface circuits and motor drivers to the initial conceptual design.

4.7.3 Vehicle Propulsion

Six thrusters will be integrated into the design of the AUV in order to propel the vehicle and provide general maneuverability under water. Four SeaBotix SBT150 thrusters and two SeaBotix BTD150 thrusters will be used for their proven quality, relatively large energy density (see Appendix A), and because inherited ownership of the SBT150 thrusters was already granted. The SBT150 and BTD150 thrusters have the same exact dimensions and weight since the I2C controllers in the SBT thrusters were extracted. Thus, each thruster will have an H-bridge motor controller installed in their place. The data and clock ports on the SBT150 thrusters will not be used. A single thruster will be placed along one of the two centerlines of each face of the open, rectangular frame in order to absolve any undesired torque on the system during operation. Each thruster will be oriented in such a way to provide the ability for three-axis translation and three-axis rotation, resulting in an agile, easily maneuverable vehicle. The bidirectional nature of the thrusters via the simple alteration of the motor direction is another convenient feature.

The six thrusters will be powered through three 2-channel L298 motor drivers. Therefore, each motor driver will connect to a pair of thrusters. The thrusters are paired according to their functions and positions. They are paired as: top and bottom (forward and backward), left and right (depth control), and front and back (rotation). The detailed connections are shown below.

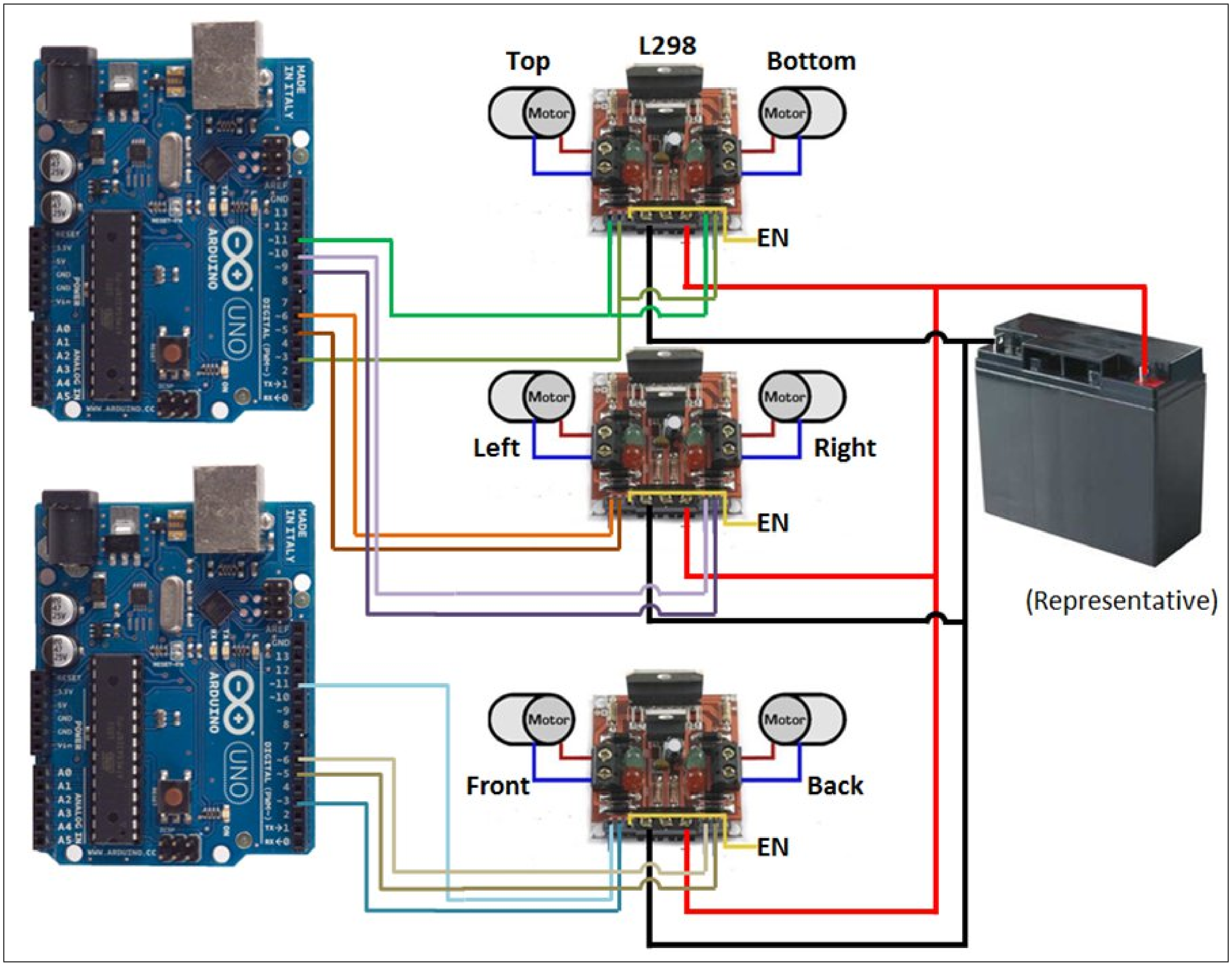


Figure 32: Propulsion System Overall Connection Diagram

The thrusters' speeds are controlled via PWM signals from the Arduino board. Each Arduino board contains 6 PWM channels on pin 3, 5, 6, 9, 10, and 11. For the top and bottom thrusters which are responsible for the forward and backward movement of the vehicle, they will be controlled by the same channels. In this way, it will be much easier to control the movement of the vehicle and reduce the possibility of out balancing the vehicle during any forward and backward movements. Each thruster will be controlled by two PWM channels to enable the bidirectional rotation. For example, as shown in the picture above, the 'Left' Thruster's speed and direction will be controlled by channel 5 and 6. Since we have a total of six thrusters, a total of 11 PWM channels are needed. Hence, we will use two Arduino boards in the AUV. The program that controls the PWM channels is attached in the Appendix. This program shall take input from the mission control program through USB serial connection and adjust the speed of motor accordingly.

Due to the aforementioned projected density of the vehicle relative to the density of salt water, and applying Equation 1 below, only 0.75 to 1.5 lbs of thrust force is expected to be required from the side thrusters in order to maintain a constant depth (i.e. zero velocity and zero acceleration).

$$F_{\text{Side Thrusters Total}} = \rho_{\text{Salt Water}} V_{\text{AUV}} g - \rho_{\text{AUV}} V_{\text{AUV}} g \tag{1}$$

This will yield minimal continuous power consumption from these thrusters. Similarly, since the pressure and frictional drag forces on the vehicle are expected to be relatively low at the expected nominal travel speed of the vehicle (i.e. about 2 to 2.5 ft/sec), only minimal thrust force will be required from the top and bottom thrusters in order to maintain a constant longitudinal velocity. Furthermore, each of the thrusters has a built-in voltage regulator at approximately 19.1 V, and thus will maintain a constant voltage as long as the supply is greater than this value and less than about 30 V. Provided the expected 28.6 V from connecting both of the lithium-ion battery packs in series, and manageable current draw from the thrusters, there should be no problem supplying ample power to the thrusters. The thrusters will be mounted to the vehicle using 1/16" thick aluminum mounting plates which have also already been manufactured and assembled. These compact designs allow the thrusters to be rigidly mounted to the frame without introducing unnecessary weight and density to the vehicle.

4.7.4 Compressed Air Distribution System

The compressed air distribution system stores and distributes pressure-regulated air to the grasp/release mechanism, as well as the torpedo launchers; actuation is initiated upon command from the main control unit. The distributed compressed air will cause the air cylinder pistons to extend (approximately 2.5"), thus thrusting the torpedoes forward. The major components of the pneumatic system are the compressed air tank, tank regulator, secondary/low pressure custom regulator, four submersible stainless steel solenoid valves, a network of $\frac{1}{4}$" OD nylon tubing, and multiple 1/8" NPT - $\frac{1}{4}$" OD tube adapters (the decision was made to use larger diameter tubing than in the initial design because it was deemed that it would be better to have a greater volume or reservoir of compressed air flowing through the gas lines to the mechanical subsystems when actuated). Originally, CO_2 was to be used as the gas of choice. However, following extensive research, the decision was made to instead use compressed air due to ability to regulate compressed air much better than CO_2, the greater availability of compressed air versus CO_2 (in order to refill the tank with highly compressed air (i.e. about 3,000 psi), the equal compactness of small paintball compressed air tanks to small paintball CO_2 tanks, and the ability to find a compatible low-pressure secondary regulator for a compressed air system versus the lack of such a regulator or series of adapters to enable the pressure from a paintball CO_2 tank to be sufficiently reduced to the desired operating pressure of approximately 100 psi. The culmination of this decision has resulted in a clean, compact, functional compressed air distribution system.

The AUV contains a total of four submersible stainless steel solenoid valves – one for each of the double acting air cylinders for the torpedo launchers, and two for the grasp/release mechanism. The solenoid valves were specified to require 12 VDC and approximately 0.583 A to open, 1/8" NPT adapters, and are 1" in diameter x 2.5" in height. The solenoid valves will each be connected to identical low-side drive BJT interface circuits to enable simple actuation via one of the microcontrollers. The network of nylon, 250 psi-rated tubing, in combination with a quad- and tri- tube junction, will enable the gas to be properly distributed to either of the torpedo launchers or the grasp/release mechanism upon command. The gas lines connect from the exit of the quad- junction to three of the four solenoid valves – one which leads to the inlet of the left torpedo launcher air cylinder, one which leads to the inlet of the right torpedo launcher air cylinder, and one which leads to the tri- junction, which routes the gas either to the inlet of the single-acting air cylinder of the grasp/release mechanism, or the inlet of the exit solenoid valve, which when actuated releases the stored compressed air into the environment, thus allowing the pressure inside this air cylinder to neutralize (i.e. purge from the system) and the internal spring to retract, thus retracting the jaws of the grasp/release mechanism. Three check valves are also integrated into the compressed air distribution system and are located at the exhausts of the double-acting air cylinders of the torpedo launchers, and at the outlet of the aforementioned exit solenoid valve of the grasp/release mechanism. These one-way check valves will serve to prevent water from flowing into the air cylinders and potentially into the gas lines, while not restricting the flow of gas out of the cylinder or solenoid valve upon actuation of these devices. A brief diagram of the compressed air distribution system can be seen below:

In the above figure, the blue blocks correspond to the storage and regulation of the compressed air and the red blocks correspond to its distribution. The network of nylon tubing is represented by the direction arrows leading from the block labeled solenoid valves to the blocks for grasp/release arms 1 and 2 and torpedoes 1 and 2.

4.7.5 Grasp / Release Mechanism

Mechanical motion will be actuated in the grasp/release mechanism by the pneumatic system. Revisions have been made to the original grasp/release mechanism, which will now feature one vertically downward-facing single-acting (i.e. contains an internal spring return) air cylinder located directly behind the camera enclosures instead of two simultaneously actuated air cylinders. This revision was made after discovering that the rescue object this year will be a laurel wreath–a naturally stable shape requiring only one set of jaws to securely grasp, unlike the previous years long PVC cylinder. A c-shaped pair of jaws will be attached to the threaded rod protruding from the single-acting air cylinder piston. The grasping claw will be formed by joining the two jaws at a central pivot via a pin-hole method; each jaw will be made of 0.25"-thick aluminum, and will be actuated to open or close via an adjustable set of arms that will induce rotation upon extension or retraction of the air cylinder piston. Proper operation of the jaws requires two submersible, stainless steel solenoid valves–one to close the jaws and grasp the laurel wreath, and the other to open the jaws (by releasing stored compressed air into the environment) and release the rescue object upon command. The air cylinders are single-acting and will attach along the center line on either side of the vehicle. Introducing this design revision removes unneeded mass from the vehicle. The grasp region of

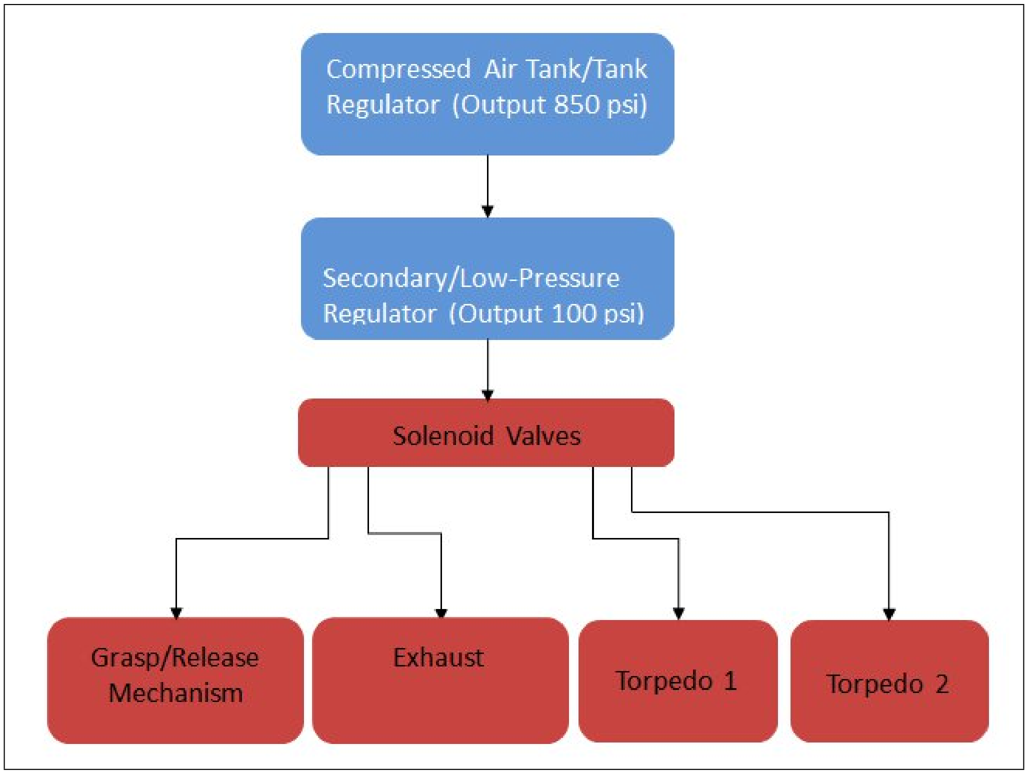


Figure 33: Compressed Air Distribution System Overview

the mechanism is expected to be approximately 9 – 10", thus providing a substantial accuracy tolerance. Furthermore, by simply shifting the location at which each of the arms contact or pivot the respective jaw, the amount of grasp force can be adjusted to compensate for a scenario in which the supply force is too large due to a necessarily greater regulated compressed air pressure for the torpedo launchers.

4.7.6 Marker Dropper

The design team has opted to use the marker dropper design created by the mechanical engineers on the 2010-2011 FAMU-FSU RoboSub team due to its simplicity and effectiveness. The mechanical subsystem is made out of aluminum 6061 and contains a parabolic track on which rests the two steel balls. The parabolic track is bound on both sides by aluminum walls in order to prevent the markers from accidentally falling off the device, as well as any undesired motion. Furthermore, there is a servo that is oriented vertically downward, located directly between each of the two markers. Upon command, the servomotor induces rotation to a desired angle, thus allowing the release of one of the two steel balls. After returning to its initial orientation, the servo can then be autonomously commanded to rotate to the same angle in the opposite direction in order to allow the other steel ball to drop into the desired bin. The servomotor will be controlled by one of the two control units located inside the pressure vessel. The wires that will need to run from the servo motor the microcontroller will be protected via thin tubing that will run to a connector on the nearest end cap. The original mount has been disassembled and a new, more appropriate mount has been installed to provide simple attachment to the bottom of the frame while maintaining easy access to the top of the device so that the markers can be conveniently reloaded.

4.7.7 Torpedo Launcher

Similar to the grasp/release mechanism, the torpedo launchers will incorporate a pneumatic system to perform the desired task. The key components of the system are the cylindrical acrylic guide barrel, cylindrical disk attachment with embedded neodymium magnet, double acting air cylinder, air cylinder mount, and the torpedo with an embedded 3/16" diameter stainless steel rod and 0.236" diameter x 0.079" neodymium magnet to obtain the desired density, balance, and magnetic attractive force from each of the torpedoes. Each of the two cannons is placed on the horizontal neutral axis on opposite sides of the vehicle. The cannons have a cylindrical shape with an inside diameter slightly larger than the maximum diameter of the torpedoes. This has been done to restrict the amount of relative motion between the torpedo and cannon walls, providing a theoretically more accurate launch. The cannons will be controlled by two independent solenoid valves (as aforementioned), allowing for the torpedoes to be fired individually. Since the 3D printer cannot directly manufacture the ABS plastic torpedoes to the desired density of saltwater, a stainless steel rod (as aforementioned above) had to be carefully inserted into the torpedoes to adjust their density. Calculations were performed using MathCad in order to derive the proper

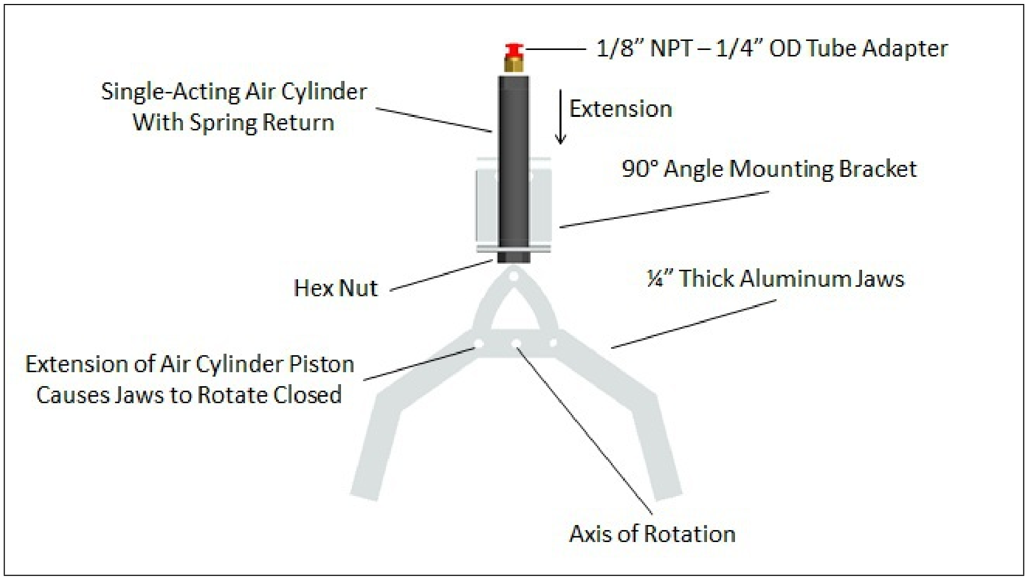


Figure 34: Close-Up View of Grasp / Release Mechanism. While this illustration conveys two simultaneously-actuated jaws, only one device will be utilized in the design. The mount will also be greatly simplified to mirror the efficient revision to the double-acting air cylinder mounts.

length of stainless steel rod that needed to be inserted into each of the torpedoes, as well as the proper depth at which the rods needed to be inserted in order to maintain proper balance of the torpedoes, and thus hydrodynamic flight through the water. Furthermore, since the neodymium magnets were slightly too strong (despite being very small), it was determined that embedding the torpedo magnet under 1/8" of epoxy at the base would reduce the magnetic attractive force between each of the torpedoes and the respective neodymium magnets on the disk attachment to the threaded rod extension from the air cylinder pistons to the desired value. Another benefit of using the neodymium magnets versus bar magnets (as was done in the initial version) is that the neodymium magnets have a north face and a south face, providing equal attractive force, and thus consistent launches, regardless of the roll orientation of the torpedo inside the acrylic cannons. Conversely, the bar magnets had both a north and south pole on each face, thus causing the attractive magnetic force between the base of the original torpedoes and the cylindrical disk attachment to be a strong function of the torpedos roll orientation; this would have potentially resulted in a lack of sufficient launch consistency and a lack of robustness of this subsystem design. Another important design criterion regarding the torpedo launchers was the safety of the mechanism; according to the rules, the mechanism should not be capable of causing bodily harm. So, in order to ensure safety of these identical mechanical subsystems, the launch velocity of the torpedoes will be controlled via simple adjustment of the low-pressure regulator until the optimal conditions are obtained. However, based off of theoretical calculations as well as data from the supplier, compressed air at 100 psi should provide about 35 pounds of launch force, yielding an appropriate initial launch velocity for the torpedoes.

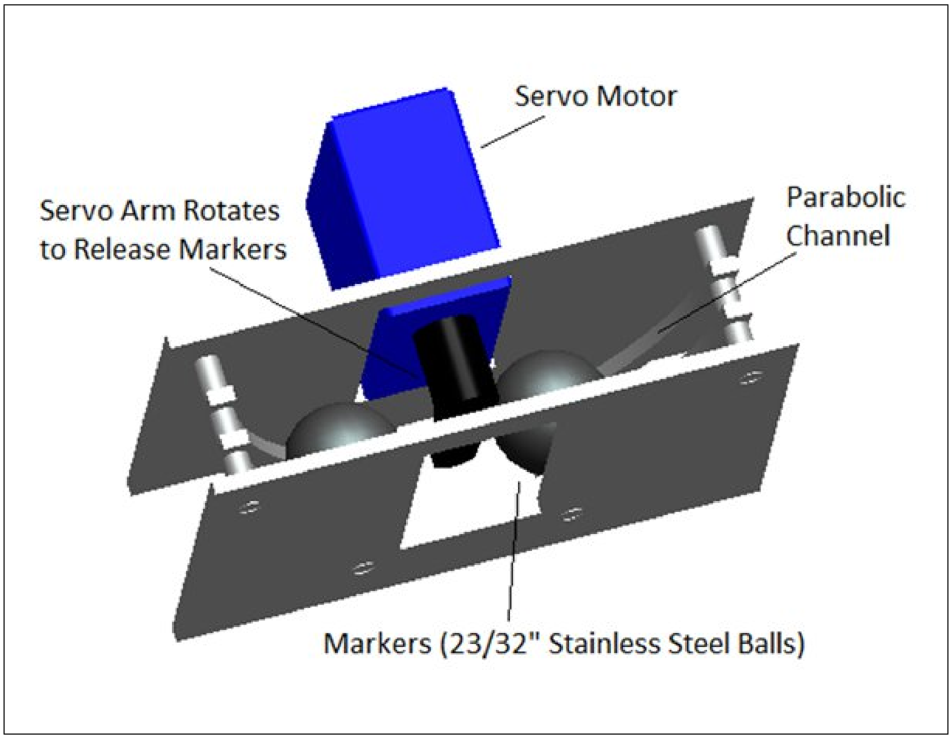


Figure 35: Close-Up View of Marker Dropper

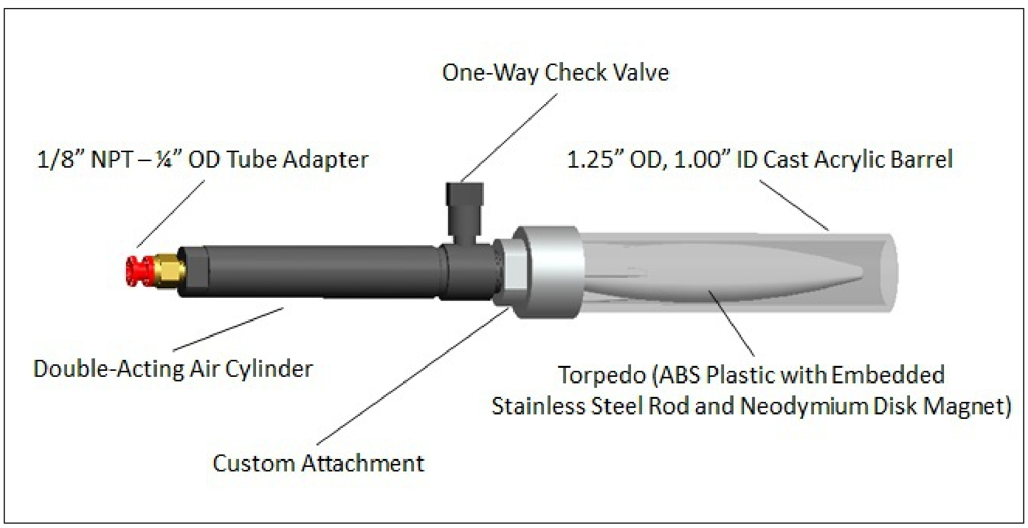


Figure 36: Close-Up View of Pro / E Rendering of Torpedo Launcher.

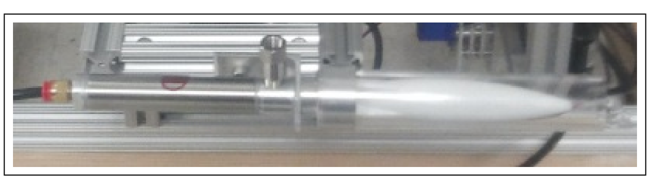


Figure 37: Close-Up View of Actual Torpedo Launcher.

5 Test Plan

5.1 System and Integration Test Plan

This section will be updated in time for the final milestone as only unit tests are being conducted at the moment.

5.1.1 Computer Vision

5.1.2 Electrical System

Subsystem

Battery Functional Test The functional test of the batteries is to ensure that the batteries will produce enough voltage and currents for the AUV. The working voltage for the AUV is required to be in the 24-32V DC range. The minimum supply voltage of the batteries should be determined. According the specifics sheet, the batteries will be cut off at 12V. This cutoff voltage will be tested.

Testing equipment: Multimeters, batteries, wires.

Testing setup

1. When battery is fully charged, measure the voltage of the battery
2. Discharge the battery during other tests, for example: testing thrusters, until the batteries do not provide enough current to drive all six thrusters. Measure the voltage at this time.

Connection Since a lot of wiring and soldering are required when integrate the entire electrical system, the electrical connections between each component need to be tested. This test can be done gradually when adding components to the system. For example, the connections between the batteries and the motor drivers can be tested first to make sure the connection is good and will not become loose easily once connected. After that we can add other components such as thrusters and voltage regulator boards to the system to make sure all connections are good. In this way we can minimize the risk of having bad connections during the competition which will probably make our AUV fail the competition.

Testing equipment: Multimeters
Testing setup:

1. Use multimeters to measure the voltage across each connection point. And use Ampere meter to measure the current go through the circuit to check if the connections are good.

Voltage Regulator Board Two voltage regulator circuits will be constructed for the AUV. One provides +5V DC for the Beagleboard, and the other one provides a +9V DC for the two Arduino boards. It needs to be verified that the VRB does provide enough current and voltage to the loads. The goal of this test is to verify the functionality of the VRB and ensure the VRB can be utilized in the system.

Testing equipment: Multimeters, variable power supplies, wires.

Testing setup:
A variable power supply can be used as the power source for the VRB to simulate the voltage drop across batteries. At first the zero load voltage from the VRB shall be tested to ensure it will not produce any voltage that would exceeds the limit of the load voltage. Then during the testing, the VRB should connect to all the loads as appropriate, and all the loads should be turned on. The input voltage should be set to maximum 16V at first, and the voltage across at the load shall be measured. If the VRB do not provide appropriate output, then further modifications or redesign of the board may be required.

Integration The integration test of the electrical system shall be conducted when all electrical components are finally assembled in the system. This test is scheduled to be in the March, the time when we starts to put together all necessary electrical components. The goal of this test is finally ensure the electrical system will be stable enough with adequate amount of power for the AUV to compete in competition. The AUV cannot attend the competition until this test is passed successfully. If the test fails at any point, we need to fix the system as soon as possible and try to avoid any damages to the electrical components on board.

Test ID	Test Description	Number of Attempts	Date (Last Attempt)	Pass / Fail
		Hardware		
CV-1	BeagleBoard xM w/ OpenCV	1	01/27/12	Fail
CV-2	Camera Calibration	1	01/29/12	Fail
CV-3	Camera Mounting	1	01/29/12	Pass
CV-4	Camera Enclosures Watertight	0	N/A	N/A
		Path Detection		
CV-5	Color Filter - Dry	0	N/A	N/A
CV-6	Color Filter - Wet	0	N/A	N/A
CV-7	Color Filter - Competition	0	N/A	N/A
CV-8	Direction - Straight Path Segment	0	N/A	N/A
CV-9	Direction - Angle between Path Segments ($> 90°$)	0	N/A	N/A
CV-10	Path Detection - Dry	0	N/A	N/A
CV-11	Path Detection - Wet	0	N/A	N/A
CV-12	Path Detection - Competition	0	N/A	N/A
		Task Identification		
CV-13	Color Filter - Dry	0	N/A	N/A
CV-14	Color Filter - Wet	0	N/A	N/A
CV-15	Color Filter - Competition	0	N/A	N/A
CV-16	Shape Detection - Dry	0	N/A	N/A
CV-17	Shape Detection - Wet	0	N/A	N/A
CV-18	Gate Task	0	N/A	N/A
CV-19	Buoys Task	0	N/A	N/A
CV-20	Box-Passing Task	0	N/A	N/A
CV-21	Fire-Torpedo Task	0	N/A	N/A
CV-22	Drop-in-Bin Task	0	N/A	N/A
CV-23	Surface-and-Recover Task	0	N/A	N/A
		System Performance and Integration		
CV-24	Path Detection and Navigation - Dry	0	N/A	N/A
CV-25	Path Detection and Navigation - Wet	0	N/A	N/A
CV-26	Task Identification and Mission Control - Dry	0	N/A	N/A
CV-27	Task Identification and Mission Control - Wet	0	N/A	N/A
CV-28	Completed AUV - Performance	0	N/A	N/A
CV-29	Completed AUV - Accuracy	0	N/A	N/A
CV-30	Completed AUV - Competition Environment	0	N/A	N/A

Table 6: Test Plans for Computer Vision

5.1.3 Mission Control

Very extensive testing shall be conducted for the mission control system to ensure that the AUV will be capable of completing each individual task during the competition. We will set up a testing environment similar to the competition site. For example, the first task of the competition is to pass through an underwater gate. In order to test this single task, an gate with the same dimensions as the specification and dimensions as the one during the competition will be constructed and placed under water in the pool of the Aquatic Center. We will test the algorithms for passing through the gate with different entry angles, different orientation and initial speed of the AUV, with or without influence of external water current, and other parameters.

Testing Equipment: AUV, aquatic center, PVC pipes, buoys, pingers

Testing setup:
A simulation of the competition site shall be constructed in the Aquatic center after we have fully assembled our AUV. Each task will have its components set up appropriately. For example, during the buoy testing, the buoys should be tied under the water at the same depth as the competition site, and the buoys shall be colored accordingly as well.

5.1.4 Propulsion System

The propulsion system shall be tested to ensure that all thrusters will cooperate with each other and allows the AUV to move in any direction that is necessary. The propulsion system should be able to maintain the AUV at certain depth, rotate the vehicle, move the vehicle forward or backward, move left or right, and finally stabilize the AUV dynamically. Each thruster's speed and direction will be controlled via PWMs generated on Arduino boards. The software part will be tested and debugged in the lab. After the propulsion system is assembled on the vehicle, we will set different paths and try to let our vehicle follow these paths. Tests include but not limited to: 1. Functional test on moving forward and backward. 2. Functional test on rotating the vehicle to both left and right. 3. Functional test on maintaining the vehicle at specified depth. 4. Integration test when all thrusters work together to finish predefined paths.

5.1.5 Mechanical System Test Schedule

5.2 Test Reports for Major Components

5.2.1 Camera Enclosures

Test ID	Test Description	Number of Attempts	Date (Last Attempt)	Pass / Fail
ME-1.1	Watertight Test - Hull	1	01/18/12	Pass
ME-1.2	Watertight Test - Camera Enclosures (Version 1)	3	01/18/12	Pass
ME-1.3	Watertight Test - Camera Enclosures (Version 2)	0	01/26/12	N/A
ME-2.1	Torpedo Test (Version 1)	1	01/18/12	Fail
ME-2.2	Torpedo Test (Version 2)	1	01/30/12	Pass
ME-3.1	Torpedo Launchers - Air Test	1	01/31/12	Pass
ME-3.2	Torpedo Launchers - Water Test	0	N/A	N/A
ME-4	Compressed Air			
ME-5	Distribution System	1	01/31/12	Pass
ME-6	Solenoid Valves - Direct Actuation	1	01/20/12	Pass
ME-7.1	Completed Vehicle Weight	0	N/A	N/A
ME-7.2	Completed Vehicle			
ME-7.3	Completed Vehicle			
ME-8	Density	0	N/A	N/A
ME-9	Balance	0	N/A	N/A
ME-10.1	Grasp / Release Mechanism - Air Test	0	N/A	N/A
ME-10.2	Grasp / Release Mechanism - Water Test	0	N/A	N/A
ME-11.1	Marker Dropper/Servo Motor - Air Test	1	12/07/11	Fail
ME-11.2	Marker Dropper - Water Test	0	N/A	N/A
ME-12.1	Interface Circuit Test - Solenoid Valves	0	N/A	N/A
ME-12.2	Interface Circuit Test - Hydrophones	0	N/A	N/A

Table 7: Test Plans for Mechanical Engineering

Team ROBOSUB FAMU-FSU College of Engineering	**Form for Report of Test and Maintenance**

Title: Camera Enclosure Watertight Test

Date (YYYY-MM-DD)	2012-01-18
Time (HH:MM)	17:30
Author	Eric Sloan, Tra Hunter, Kashief Moody
Location	FSU Morcom Aquatics Center
Test No	1

	Component
■	Hull / Frame: Camera Enclosure
☐	Interior Hull
☐	Electronics
☐	Electrical System
☐	Mechanical Subsystem: ____________________
	Test Result: ☐ Pass ■ Fail

Objective

Verify that the camera enclosures are properly sealed from the surrounding water upon submersion.

Equipment

- Pool
- Rope
- Camera Enclosures
- Stop watch

Process

The camera enclosures were tested along with the hull. The watertight test was conducted after the applied 100% silicone sealant had been provided ample time to set. The vehicle was placed in the water with the rope attached to it. The vehicle was allowed to stabilize itself after its substantial positive buoyancy was verified. Thereafter, the vehicle was forced to submerge and held underwater for an extended period of time.

Anticipated Results

The camera enclosures should be completely dry on the inside once they resurface.

Success Criteria

The enclosures will be considered properly sealed if after the 10 minute waiting period has elapsed, their interior is completely dry.

Actual Results

The test was a disappointing failure. At the end of the 10 minute waiting period, the bottom camera enclosure had almost completely flooded, and the top camera enclosure was partially flooded. This exposed clear leakage issues that needed to be addressed.

Comments

The camera enclosures were mated to their end caps using silicon sealant as with the hull. However, unlike the cast acrylic hull, the acrylic camera enclosure boxes were custom built by adhering each of the faces of the box along the mating edges. The implementation of a non-optimal construction methodology in conjunction with the required penetration of the enclosures with two bolts and a threaded rod to mount the cameras (although neoprene sealing washers were used to prevent water intrusion), culminated in the flooding of the enclosures. By observing the locations of the bubbles emitting from the enclosures during submersion, the origin of the sealing failure (and thus water penetration) were detected, and it was decided to simply add a careful layer of caulking at these locations, as well as to all the edges of the enclosures as a precaution. This was very difficult to do at this stage, however—particularly to apply caulking to the inside of the enclosures given their small interior and the non-removable, obstructive threaded rod in the way.

Team ROBOSUB FAMU-FSU College of Engineering	Form for Report of Test and Maintenance

Title: Watertight Test

Date (YYYY-MM-DD)	2012-01-23
Time (HH:MM)	18:30
Author	Eric Sloan
Location	Pool Outside Apartment
Test No	2

	Component
■	Hull / Frame: Camera Enclosure
☐	Interior Hull
☐	Electronics
☐	Electrical System
☐	Mechanical Subsystem: ____________________
	Test Result: ☐ Pass ■ Fail

Objective
Verify that the camera enclosures are properly sealed from the surrounding water upon submersion.

Equipment
- Pool
- Camera Enclosures
- Stop watch

Process
The camera enclosures were tested separately from the hull this time around. After an unsuccessful first attempt, the enclosures were dried and resealed with additional caulking (as aforementioned in the conclusion of the initial test). The watertight test was conducted again once the sealant had properly set. After tightly closing the end caps, the enclosures were forced to submerge in a pool and held in place for an extended period of time.

Anticipated Results
The camera enclosures should be completely dry on the inside once they resurface.

Success Criteria
The camera enclosures will be considered properly sealed if after the 10 minute waiting period has elapsed, their interior is completely dry.

Actual Results
As with the first test, the second test was also a failure. At the end of the ten minute waiting period, the leakage was substantially reduced, but there still remained significant water intrusion (identified to be mostly through the bolt holes in the bases).

Comments
This test result directed the attention to thoroughly sealing the base of the enclosures—particularly the bolt holes, since the sealing washers were apparently insufficient. It has been concluded that a more severe approach needed to be derived in order to provide a reliable seal at the bases, so 3/8" EPDM rubber will be ordered and water cut to be adhered to each of the bases via gasket adhesive, enabling the heads of the bolts, and the threaded rod to protrude through properly-located holes in the rubber. This will provide cavities in which to apply the less viscous (than silicone) epoxy, and also yield a clean surface with only four clear edges of the top surfaces to caulk sealed. Hopefully this will ultimately solve the issue of flooding.

Team ROBOSUB FAMU-FSU College of Engineering	Form for Report of Test and Maintenance

Title: Watertight Test

Date (YYYY-MM-DD)	2012-01-26
Time (HH:MM)	19:30
Author	Eric Sloan
Location	Pool Outside Apartment
Test No	3

	Component
■	Hull / Frame: Camera Enclosure
☐	Interior Hull
☐	Electronics
☐	Electrical System
☐	Mechanical Subsystem: ____________________
	Test Result: ■ Pass ☐ Fail

Objective

Verify that the camera enclosures are properly sealed from the surrounding water upon submersion.

Equipment

- Pool
- Camera Enclosures
- Stop watch

Process

After an unsuccessful second attempt, the enclosures were dried and resealed with additional caulking, and the EPDM padding design was successfully implemented (although it was concededly messy due to the limited working space and properties of the adhesives and sealant). The watertight test was conducted again once the sealant and epoxy had properly set. After tightly closing the end caps, the enclosures were forced to submerge in a pool and held in place for an extended period of time.

Anticipated Results

The camera enclosures should be completely dry on the inside once they resurface.

Success Criteria

The enclosures will be considered properly sealed if after the 10 minute waiting period has elapsed, their interior is completely dry.

Actual Results

The test was a success. After the 10 minute waiting period, the interior of the enclosures were completely dry, except for a few negligible drops in the bottom enclosure.

Comments

Despite a relatively successful test, the messiness of the final product, multitude of sealed interfaces with the potential for leakage, lack of versatility of the design (e.g. the ability to adjust the height at which the cameras are mounted), as well as the successive initial failed tests, has brought about concern about the reliability and aesthetic appeal of these enclosures. A backup or alternative option will be pursued with urgency. However, at the very least, these enclosures should suffice for preliminary computer vision testing/training.

5.2.2 Compressed Air Distribution System

Team ROBOSUB FAMU-FSU College of Engineering	Form for Report of Test and Maintenance

Title: Compressed Air Distribution System

Date (YYYY-MM-DD)	2012-01-31
Time (HH:MM)	17:30
Author	Eric Sloan, Tra Hunter
Location	Senior Design Room
Test No	1

	Component
☐	Hull / Frame
☐	Interior Hull
☐	Electronics
☐	Electrical System
■	Mechanical Subsystem: Compressed Air Distribution System

Test Result: ■ Pass ☐ Fail

Objective

Check the compressed air distribution system for the proper regulated pressure, no leakage of air, and integration with the solenoid valves and air cylinders

Equipment

- Compressed Air Tank
- Tank/High Pressure Regulator
- Secondary/Low Pressure Regulator
- Nylon 1/4 " OD, 0.170" ID Gas Lines (rated up to 250 psi)
- Four Solenoid Valves
- 1/8" NPT – ¼" OD tubing push-to-connect instant fittings/adapters
- Check valve
- Power supply

Process

Directly connected to the tank is the tank/high pressure regulator, which reduces the 3,000 psi air from the tank to approximately 850 psi. Attached directly to the outlet of the tank regulator is the secondary/low pressure regulator, which drops the 850 psi air from the tank regulator to approximately 100 psi (this is adjustable). The nylon gas lines are then connected to the outlet of the low pressure regulator (via an 1/8" NPT – ¼" OD tubing push-to-connect instant fitting/adapter), which then feeds into the quad junction/splitter which sends the gas directly to three of the four solenoid valves (i.e. left torpedo launcher double-acting air cylinder, right torpedo launcher double-acting air cylinder, and grasp release mechanism). The circuit is completed with the tubing from the exit of the third (grasp release mechanism) solenoid valve feeding into a tri junction/splitter, which outlets to the fourth/exit solenoid valve, as well as directly to the inlet of the single-acting air cylinder for the grasp/release mechanism. Finally, the outlet of the fourth/exit solenoid valve has a check valve installed, which will be used to prevent water from flowing into the gas lines while the compressed air is purged from the single-acting air cylinder, ultimately enabling the retraction of the grasp/release mechanism jaws. After regulating the outlet pressure of the low pressure regulator to 100 psi and turning on the power supply, the solenoid valves were directly actuated, enabling the individual extension of the double-acting air cylinders. The single acting air cylinder will be tested shortly.

Anticipated Results

The compressed air tank should supply enough gas, and at the proper regulated pressure (with no leakage) in order to actuate the air cylinders. The gas lines should also be capable of releasing the internal stored air pressure in the gas lines by simply actuating the opening of the exit solenoid valve to purge the gas from the system.

Success Criteria

The compressed air tank and complementary regulators must deliver the gas to the lines at the desired pressure, and there must not any air leakage of air through the gas lines or adapters (thus verifying the sustainability of compressed air in the gas lines). Furthermore, a large enough volume of air must be supplied through the lines in order to properly actuate the air cylinders, thus thrusting the pistons forward at the specified force of around 35 lb. In order to do this, the solenoid valves must properly open when provided 4V – 5V, thus releasing the compressed gas to the respective air cylinder inlet.

Actual Results

The compressed air distribution system worked as expected. The low pressure regulator was capable of delivering the needed output pressure of 100 psi, and was easily adjustable using a hex key as designed. In addition, the nylon tubing and the multiple instant-connect adapters showed no signs of gas leakage and provided a large enough volume of gas to properly actuate each of the double-acting air cylinders. Due to some preliminary mishaps in properly installing the filled compressed air tank, however, the tank had lost 2000 psi of air pressure by the time all the testing was completed. This will easily be avoided in the future now that the proper installation techniques have been established. The tank will once more be refilled at a filling station (scuba diving shop), and should be sufficient to last the entirety of the testing phase at the very least. The tank will likely be topped off once again prior to the competition as well.

Comments

The air tank needs to be refilled prior to testing the grasp/release mechanism in the coming weeks. This is a non-issue as it costs only $1 to have it filled and the location is relatively convenient.

5.2.3 Hydrophones

Team ROBOSUB FAMU-FSU College of Engineering	Form for Report of Test and Maintenance

Title: Hydrophone Frequency Test

Date (YYYY-MM-DD)	2012-01-20
Time (HH:MM)	17:30
Author	Antony Jepson
Location	Senior Design Room
Test No	1

	Component
☐	Hull / Frame
☐	Interior Hull
■	Electronics
☐	Electrical System
☐	Mechanical Subsystem: ____________________
	Test Result: ■ Pass ☐ Fail

Objective

What is the goal of the test?

The goal of this test was to determine if the hydrophones available for this project, rated for frequencies up to 28kHz, could support frequencies up and to 30kHz without a substantial drop off.

Equipment

List equipment used to complete the test.

1x SQ26 hydrophones
1x Speaker
1x Tektronix oscilloscope
1x Tektronix signal generator

Process

List steps taken to complete the test.

1. Connect speaker to signal generator.
2. Ramp up signal generator in 1kHz increments from 1kHz to 33kHz.
3. Record frequency response from hydrophone on Tektronix oscilloscope.

Anticipated Results

What is the expected outcome?

The signal will be measured successfully at 30khz.

Success Criteria

Which events determine a successful test?

If frequencies including and above 30kHz are reliably and reproducibly detected.

Actual Results

What were the results of the test? Is another test required?

The test was successful. The hydrophone detected a signal up to 37kHz without a reasonable dropoff.

The frequencies reads: 37.3903kHz.

5.2.4 Torpedoes

Team ROBOSUB FAMU-FSU College of Engineering	**Form for Report of Test and Maintenance**

Title: Torpedo Performance

Date (YYYY-MM-DD)	2012-01-30
Time (HH:MM)	21:30
Author	Eric Sloan
Location	Pool Outside Apartment
Test No	2

	Component
☐	Hull / Frame
☐	Interior Hull
☐	Electronics
☐	Electrical System
■	Mechanical Subsystem: Torpedo Launcher
	Test Result: ■ Pass ☐ Fail

Objective

Test the torpedoes' hydrodynamics, density, and balance in an underwater setting.

Equipment

- Torpedoes (Version 2)
- Water

Process

The torpedoes were placed in the water and released horizontally. This was done in order to view the balance and relative density of the torpedoes. Next, the torpedoes were launched by hand, horizontally through the water to qualitatively (and somewhat primitively) check for their flight characteristics.

Anticipated Results

The torpedoes should be slightly negatively buoyant, and should thus slowly sink toward the floor of the pool. Furthermore, the pitch angle of the torpedoes should approach zero degrees when released horizontally in the water, indicating properly balanced torpedoes (and better hydrodynamic characteristics). Furthermore, the torpedoes are expected to glide straight through the water without much deviation from the initial launch direction, and should travel a reasonable distance for the provided input force prior to coming to a halt.

Success Criteria

The torpedoes need to be balanced, have the proper density (at least passing the eye test), and demonstrate streamline capabilities in order to be ready to be fired accurately underwater using the torpedo launchers.

Actual Results

The revised torpedoes demonstrated proper balance, proper density, and great streamline capabilities in the fresh water pool. Furthermore, the attractive force between the embedded neodymium magnets and neodymium disk magnets on the cylindrical piston attachments to the double-acting air cylinders was optimal. Thus, it has been determined that the torpedoes are ready for direct, final integration onto the vehicle.

Comments

The torpedoes now need to be actuated from the torpedo launchers both in air and underwater in order to confirm their successful integration into the broader system.

Team ROBOSUB FAMU-FSU College of Engineering	Form for Report of Test and Maintenance

Title: Torpedo Launchers – Air Test

Date (YYYY-MM-DD)	2012-01-31
Time (HH:MM)	17:30
Author	Eric Sloan, Tra Hunter
Location	Senior Design Lab
Test No	1

	Component
☐	Hull / Frame
☐	Interior Hull
☐	Electronics
☐	Electrical System
■	Mechanical Subsystem: Torpedo Launcher
	Test Result: ■ Pass ☐ Fail

Objective

Test the torpedo launchers via the integration of the new torpedoes, solenoid valves, and the compressed air distribution system in an air environment.

Equipment

- Torpedo Launchers
- Compressed Air Tank
- Tank Regulator
- Pressure Gauge(s)
- Secondary/Low Pressure Regulator
- Solenoid Valves
- Nylon Gas Lines + Adapters + Check Valves
- Torpedoes
- DC Power Supply

Process

The compressed air tank was filled to 3,000 psi. After properly connecting the low pressure regulator to the tank regulator as well as to the gas lines, the low pressure regulator was adjusted (via rotating the set screw/knob at the top of the device and verifying the outlet pressure via the attached gauge) to drop the pressure from the tank regulator from 850 psi to 100 psi. Since the gas distribution circuit was entirely complete by the time of this test, no further connections needed to be made. The torpedoes were loaded into their respective cast acrylic cannons, and all gas lines, adapters were ensured to be secure and that there were no leaks. Then, a cushioned box was placed in front of the torpedo launcher to be tested to ensure a relatively soft impact. Thereafter, the power supply was turned on and the leads were attached to those of the solenoid valve controlling the direct flow the inlet of the double-acting air cylinder for one of the torpedo launchers. The voltage was then progressively increased until the 4V – 5V actuation threshold was reach, at which point the torpedo launcher was actuated, and the torpedo took flight toward the cushioned box. This process was repeated for the other torpedo launcher as well.

Anticipated Results

The torpedo launcher air cylinder piston should fully extend (about 2.5”) once the pressure-regulated compressed air is allowed to flow past the actuated solenoid valve and into the double-acting air cylinder. The torpedo should then detach from the neodymium magnet on the piston attachment, and exit the barrel at a reasonably high velocity, taking a straight, direct flight into the cushion box several feet down range.

Success Criteria

The torpedoes will be individually fired at a safe, yet sufficiently high speed, and in a straight path toward the target.

Actual Results

The test was a success, as the torpedoes were accurately launched at an apparently sufficiently high speed and at a straight path toward the target (i.e. cushioned box down range). After further review of the video footage, analysis confirmed what was witnessed live, which was that the torpedoes displayed great aerodynamics, and also traveled about 8 – 10 feet in flight at the 100 psi setting.

Comments

The torpedo launcher was a success in lab/air setting. The final test will be conducted underwater when the vehicle has been completed. During this test, the hydrodynamics of the torpedoes via actuation of the torpedo launchers will be assessed, and the horizontal distance traveled at a 100 psi setting will be determined. Through this test, the optimal working pressure of the compressed air will also be concluded, although it appears that 100 psi should be close to ideal. Furthermore, potential leakage of water into the gas lines will be assessed, although this is not anticipated to be of concern due to the integrity of the adapters and the lack of any leakage of compressed air through the gas lines during the in-air test.

Team ROBOSUB FAMU-FSU College of Engineering	Form for Report of Test and Maintenance

Title: Torpedo Performance

Date (YYYY-MM-DD)	2012-01-20
Time (HH:MM)	17:30
Author	Eric Sloan, Tra Hunter, Kashief Moody
Location	FSU Morcom Aquatic Center
Test No	1

	Component
☐	Hull / Frame
☐	Interior Hull
☐	Electronics
☐	Electrical System
■	Mechanical Subsystem: Torpedo Launcher
	Test Result: ☐ Pass ■ Fail

Objective

Test the torpedoes' hydrodynamics, density, and balance in an underwater setting.

Equipment

- Torpedoes
- Pool

Process

The torpedoes were placed in the water and released horizontally. This was done in order to view the balance and relative density of the torpedoes. Next, the torpedoes were launched by hand, horizontally through the water to qualitatively (and somewhat primitively) check for their flight characteristics.

Anticipated Results

The torpedoes should be slightly negatively buoyant, and should thus slowly sink toward the floor of the pool. Furthermore, the pitch angle of the torpedoes should approach zero degrees when released horizontally in the water, indicating properly balanced torpedoes (and better hydrodynamic characteristics). Furthermore, the torpedoes are expected to glide straight through the water without much deviation from the initial launch direction, and should travel a reasonable distance for the provided input force prior to coming to a halt.

Success Criteria

The torpedoes need to be balanced, have the proper density (at least passing the eye test), and demonstrate streamline capabilities in order to be ready to be fired accurately underwater using the torpedo launchers.

Actual Results

The tests demonstrated that the torpedoes are slightly negatively buoyant, which passes this requirement. However, the balance test did not pass was deemed a mild failure due to the apparent miscalculation from neglecting the addition of the bar magnet to the base of each of the torpedoes. The streamline test proved to be difficult to judge and somewhat inconclusive, but it was by no means a clear success as direct, straight, sustained underwater flight was not consistently attained.

Comments

New torpedoes will be developed to accommodate for the slight error(s) made in these initial versions. The updates include further smoothing the surface of the torpedoes by finely sanding them, applying a thicker layer of spray paint to ensure an even smoother finish (to reduce frictional drag during flight), embedding them with miniature neodymium magnets instead of adhering bar magnets to the base of the torpedoes (to make the torpedoes more compact and guarantee a more consistent launch alignment), and factoring in the mass properties of the magnets during the revised calculations to determine the proper length of the stainless steel rod inserts, as well as the proper drilling depth and displacement between the torpedo magnets and disk attachment magnets (neodymium as well) in order ensure the proper density and balance of the torpedoes, as well as to obtain the desired magnetic attractive force to prevent premature release of the torpedoes while not significantly inhibiting the launch velocity of the torpedoes at a given compressed air pressure. The revised versions are expected to perform exceptionally well.

5.2.5 Servomotor

Team ROBOSUB FAMU-FSU College of Engineering	Form for Report of Test and Maintenance

Title: Servomotor Test

Date (YYYY-MM-DD)	2011-12-07
Time (HH:MM)	14:30
Author	Eric Sloan, Tra Hunter
Location	Mechatronics Lab, College of Engineering, Room B324
Test No	1

	Component
☐	Hull / Frame
☐	Interior Hull
☐	Electronics
☐	Electrical System
■	Mechanical Subsystem: Marker Dropper
	Test Result: ☐ Pass ■ Fail

Objective
Successfully command the servomotor to rotate to specified angles relative to its initial position, thus simulating the release of the markers (i.e. stainless steel spheres).

Equipment

- **Freescale Dragon Board**
- **Bread Board**
- **Fully Charged NiMH Battery (≈ 8V)**
- **Code Warrior (Programming Language – C)**

Process
Code Warrior was used to develop a simple program to test the proper actuation of the servo motor by simply pressing one of four SW push buttons. The power and ground leads of the servo motor were hooked up to the power supply via a bread board, and the PWM enable wire was routed to the Dragon Board. Furthermore, a common ground was established between the bread board and the Dragon Board. The basic code was then debugged and compiled.

Anticipated Results
When one of the SW push buttons is individually pressed, the servo motor should rotate in one direction to a specified location/angle, allowing one of the markers to fall through the exposed gap below the servo arm. When the other activated push button is pressed, the servo arm should rotate in the opposite direction to allow the other marker to be released as well.

Success Criteria
The servo arm must successfully rotate to the provided locations, thus releasing each of the two markers after each of the buttons is pushed.

Actual Results
The servomotor failed to respond to the pushbutton commands. However, after replacing the original servo motor with another servo motor in the lab, the new servo motor responded as expected (with the same code). Thus, it has been concluded that the current servo motor is malfunctioning/dead, and a new one needs to be purchased (≈ $20).

Comments
The new servo motor has been purchased, and upon arrival, it will be tested with the same code. If there is still an issue, the product specifications and the code will be carefully reviewed for potential sources of the servo inactivity. It is strongly believed, however, that the error was in the hardware.

5.2.6 Arduino Interface

5.2.7 AUV Hull

Team ROBOSUB FAMU-FSU College of Engineering	Form for Report of Test and Maintenance

Title: Watertight Test

Date (YYYY-MM-DD)	2012-01-18
Time (HH:MM)	17:30
Author	Eric Sloan, Tra Hunter, Kashief Moody
Location	FSU Morcom Aquatics Center
Test No	1

	Component
■	Hull / Frame
■	Interior Hull
☐	Electronics
☐	Electrical System
☐	Mechanical Subsystem: ____________________
	Test Result: ■ Pass ☐ Fail

Objective

Verify that the hull of the vehicle is properly sealed from the surrounding water when submerged, and simultaneously assess the balance and density of the current system.

Equipment

- Pool
- Rope
- Hull
- Stop watch

Process

When constructed, the hull was mated with the end caps using 100% silicon sealant. In order to complete the seal, additional sealant was placed around the outer ring of the end caps. The watertight test was conducted once the sealant had been given ample time to set. The vehicle was placed in the water with the rope attached to it. The vehicle was allowed to stabilize itself after its positive buoyancy was verified. The current balance of the system was also assessed during this process. Thereafter, the vehicle was forced to submerge and held underwater for an extended period of time.

Anticipated Results

The vehicle is expected to be properly balanced in reference to a stable equilibrium roll angle of zero degrees. The vehicle should also be substantially positively buoyant at this point since several components of the AUV were not attached or installed during this test. Upon resurfacing, it is expected that the hull be completely dry on the inside, thus indicating that the electronics will be well protected and are safe to install.

Success Criteria

A properly sealed hull will result in a completely dry interior hull. The hull must stay fully submerged underwater for 10 minutes.

Actual Results

The test was successful in that the vehicle was, in fact, substantially positively buoyant, the vehicle was balanced (although the hull need to be rotated a very slight angle in order to completely optimize the naturally stable roll angle), and the hull proved to be water tight.

Comments

The hull was tested along with the camera enclosures, and the end caps of each of these enclosures will receive an additional clean bead of provisionary caulking at the aluminum – acrylic interfaces prior to testing the completed vehicle underwater.

5.2.8 Computer Vision

Team ROBOSUB FAMU-FSU College of Engineering	Form for Report of Test and Maintenance

Title: Run OpenCV on the BeagleBoard xM

Date (YYYY-MM-DD)	2012-01-27
Time (HH:MM)	18:25
Author	Ryan Kopinsky
Location	Home
Test No	1

	Component
☐	Hull / Frame
☐	Interior Hull
■	Electronics
☐	Electrical System
☐	Mechanical Subsystem: ____________________
	Test Result: ☐ Pass ■ Fail

Objective

Successfully run an OpenCV test program on the BeagleBoard xM.

Equipment

- Logitech C615 Webcam
- BeagleBoard xM

Process

Install OpenCV libraries and compile a test program to run on the BeagleBoard xm. The UCV driver for the webcam also has to be installed on Angstrom (Linux on the BeagleBoard xM).

Anticipated Results

Successfully run an OpenCV test program on the BeagleBoard xM. The test program will access the webcam as well.

Success Criteria

Processed image frames need to be saved to memory on the BeagleBoard xM.

Actual Results

OpenCV libraries were successfully installed on the BeagleBoard. The test program was also successfully compiled; however, after installing the UCV driver for the webcam, a kernel panic occurred. The BeagleBoard freezes at startup and the team is in the process of restoring the system. A follow-up test is required.

Comments

Team ROBOSUB FAMU-FSU College of Engineering	**Form for Report of Test and Maintenance**

Title: Camera (Down-Facing) Calibration

Date (YYYY-MM-DD)	2012-01-29
Time (HH:MM)	16:30
Author	Ryan Kopinsky
Location	Home
Test No	1

	Component
☐	Hull / Frame
☐	Interior Hull
■	Electronics
☐	Electrical System
☐	Mechanical Subsystem: ____________________
	Test Result: ☐ Pass ■ Fail

Objective

Configure the camera for low lighting and underwater conditions.

Equipment

- Camera Enclosure
- Logitech C615

Process

The camera has to be tested for low lighting and underwater conditions. This will ensure that image processing will be successful at the competition. The camera is tested in a hot tub, a pool and a simulated-competition environment (pool with sediments and low-light).

Anticipated Results

The camera will identify colors in low-lighting and underwater (competition) conditions.

Success Criteria

The camera needs to identify red (or any other color) in an environment that is very similar to the competition pool.

Actual Results

The camera is mounted in a way such that it is down-facing at a tilted angle (it is looking forward). The control algorithm needs to be aware of the fact that the camera looks at what is ahead of the vehicle, not right below it.
Isolating the color red in the hot tub was successful; however, the conditions were too controlled to simulate the competition environment. A follow-up test for competition conditions is required.

Red was isolated for the following HSV values:

- Hue: min = 28, max = 41
- Saturation: min = 82, max = 174
- Value: min = 185, max = 256

Comments

A test environment needs to be constructed to resemble competition conditions. This will be done ASAP. The follow-up test will follow shortly after.

New Risk: the camera might not have enough depth to see details from a distance.

Team ROBOSUB FAMU-FSU College of Engineering	**Form for Report of Test and Maintenance**

Title: Camera (Down-Facing) Mounting

Date (YYYY-MM-DD)	2012-01-29
Time (HH:MM)	15:52
Author	Ryan Kopinsky
Location	Home
Test No	1

	Component
☐	Hull / Frame
☐	Interior Hull
■	Electronics
☐	Electrical System
☐	Mechanical Subsystem: ____________________
	Test Result: ■ Pass ☐ Fail

Objective
Determine the optimal mounting position in the camera enclosure for the down-facing camera.

Equipment

- Camera Enclosure
- Logitech C615 Webcam

Process
The camera needs 3 ½ turns to be fixed on the screw thread. One therefore needs to start at the right position for the camera to face in the right direction.

Anticipated Results
The camera will be fixed in one position facing the right direction.

Success Criteria
Once the camera is mounted, it should not move. After calibration is complete, the camera should stay in that specific configuration. Once the end-caps are closed, the camera will not be easily accessible.

Actual Results
Mounting of the camera is fairly difficult due to the limited space in the camera enclosures; however, after two attempts, the camera was mounted in the optimal position.

Comments
For future designs, the camera should be mounted on a bracket. The bracket can then be fixed in any desired position in the camera enclosure.

5.2.9 Mission Control

Team ROBOSUB FAMU-FSU College of Engineering	**Form for Report of Test and Maintenance**

Title: Communication between Host and Arduino board

Date (YYYY-MM-DD)	2012-01-16
Time (HH:MM)	1:00pm
Author	Hang Zhang
Location	Senior Design Lab
Test No	1

	Component
☐	Hull / Frame
☐	Interior Hull
☐	Electronics
x	Electrical System
☐	Mechanical Subsystem: ____________________
	Test Result: X Pass ☐ Fail

Objective

Enable the communication between the Arduino board and the host, so that the speed of thrusters can be controlled by host computer.

Equipment

Arduino board;
Thruster;
L298 Motor Driver;
Power Supply;

Process

1. Connect the thrusters to motor driver, connect the motor driver to arduino board through PWM pins, and connect Arduino board to host through USB2.0
2. Set up the programs on both the host computer and the Arduino
3. Enter user input on the host computer
4. Monitor the feedback of the speed from the Arduino board and notice the speed of the Thrusters.

Anticipated Results

The thrusters shall rotate at the speed and direction specified by the host program.

Success Criteria

The thrusters shall rotate at the speed and direction specified by the host program.

Actual Results

The thruster can rotate at both direction and any speed according the user input at the host computer. The test is very successful.

Comments

The communication between the Arduino board and the host computer are successful.

Team ROBOSUB FAMU-FSU College of Engineering	**Form for Report of Test and Maintenance**

Title: Inter Process Communication

Date (YYYY-MM-DD)	2012-01-24
Time (HH:MM)	3:00pm
Author	Hang Zhang
Location	Senior Design Lab
Test No	3

	Component
☐	Hull / Frame
☐	Interior Hull
■	Electronics
☐	Electrical System
☐	Mechanical Subsystem: ____________________
	Test Result: ☐ Pass ☐ Fail

Objective

Test the message passing interface between different processes

Equipment

Host computer

Process

1. *Prepare for the main program which shall fork different test programs*
2. *Test program should send data to the main process*
3. *The main process than print data to the standard console*

Anticipated Results

Message from the test programs should be successfully printed to the console by the main program.

Success Criteria

Message from the test programs should be successfully printed to the console by the main program.

Actual Results

Message successfully delivered between several tests programs and the main program.

Comments

None

Team ROBOSUB FAMU-FSU College of Engineering	**Form for Report of Test and Maintenance**

Title: Thruster Controlled by Same PWM Channel Set

Date (YYYY-MM-DD)	2012-01-21
Time (HH:MM)	3:00pm
Author	Hang Zhang
Location	Senior Design Lab
Test No	2

	Component
☐	Hull / Frame
☐	Interior Hull
☐	Electronics
x	Electrical System
☐	Mechanical Subsystem: ____________________
	Test Result: x Pass ☐ Fail

Objective

To measure the current difference when two thrusters are controlled by the same PWM signals and verify if it is possible to be implemented in this way.

Equipment

Power supply;
Two Digital multimeters, one for currents and one for voltages;
Two thrusters;
One L298 motor driver;

Process

PWM channel 3 and 11 are used which are based on the same timer. The motor controller now propels two thrusters. A power supply is set to be 19V. The arduino board is connected to the host using USB. Speed and rotation information are sent to PWM controller on the Arduino board from the host program.

Anticipated Results

Current draw and voltage drop should be equal for both thrusters.

Success Criteria

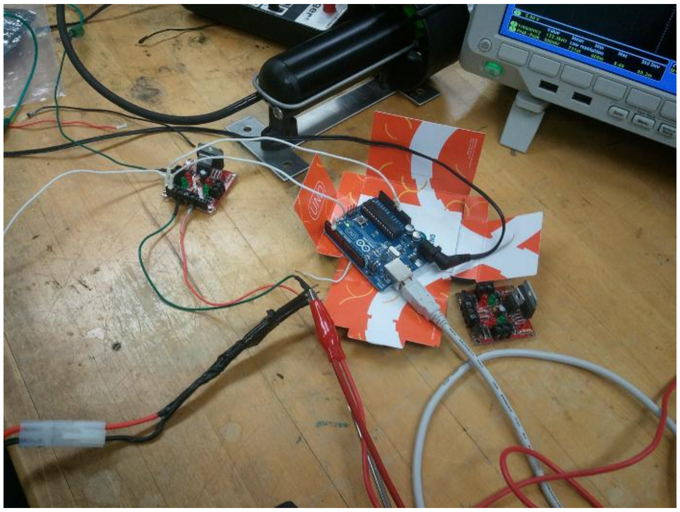
Measured Current and Voltage values are almost equal for both thrusters, very small discrepancies can be tolerated.

Actual Results

	T1(I1,I2) A	T2(I3,I4) A	Voltage(ac) V
19V speed: 127			
T1 OFF	0	1.24	8.34
T2 OFF	1.24	0	8.36
Both ON:	1.240	1.232	8.70(T1) 8.69(T2)
19V speed: 50			
T1 OFF	0	0.786	6.67
T2 OFF	0.80	0	6.69
Both ON:	0.75	0.75	7.28(T1) 7.22(T2)
19V speed: 20			
T1 OFF	0	0.363	4.57
T2 OFF	0.387	0	4.60
Both ON:	0.380	0.362	4.59(T1) 4.55(T2)

Comments

1. Batteries connected in series provides 29.6V. Should be less than 19V coming out of the motor controller, which means no voltage regulators are needed.
2. Very small current difference between two thrusters when controlled by the same PWM channel. Therefore, two thrusters could be controlled by one set of PWM channel.For example: Top and Bottom Thrusters.
3. Have two thrusters controlled by one set of channel may result in difficulty of balancing the vehicle. Further tests needs to be carried out.



5.2.10 Solenoid Valve

Team ROBOSUB FAMU-FSU College of Engineering	**Form for Report of Test and Maintenance**

Title: Solenoid Valve Test

Date (YYYY-MM-DD)	2012-01-20
Time (HH:MM)	16:25
Author	Eric Sloan, Tra Hunter
Location	Senior Design Lab
Test No	1

	Component
☐	Hull / Frame
☐	Interior Hull
■	Electronics
☐	Electrical System
☐	Mechanical Subsystem: ____________________
	Test Result: ■Pass ☐ Fail

Objective

Actuate the solenoid valves to open using a direct power supply.

Equipment

- **Solenoid Valves (quantity 4)**
- **Tektronix DDM4050 Power Source**
- **Electrical Leads (Positive and Negative)**

Process

The power supply was initially set up, turned on, and adjusted to 0 VDC. The positive and negative leads from the power supply were then connected to the corresponding leads of one of the four solenoid valves. The voltage was then progressively increased in increments of 1V until the solenoid valve was actuated (a "click" can be clearly heard when the power (i.e. voltage) threshold has been surpassed). The voltage was driven up to the suggested operative voltage of 12 VDC (at about 0.583 A), and then was cycled back to 0 VDC in order to deactivate the solenoid valve. This process was done for each of the four solenoid valves, including the exit valve for the grasp/release mechanism which contains a one-way check valve.

Anticipated Results

It's anticipated that the solenoid valves will spring open at 12 VDC as projected by the product specifications. Furthermore, it is expected that a provided voltage below this threshold will cause the solenoid valves to deactivate and return to their naturally closed positions.

Success Criteria

The test will be considered a success if the solenoid valves are actuated open at a given voltage, particularly at or below 12 VDC, and return to their closed position when the provided voltage is reduced below that threshold.

Actual Results

The test was a success, but was also surprising. While gradually increasing the supply voltage, each of the solenoid valves opened at around 4V – 5V, rather than the expected 12 V. Finally as the voltage was gradually decreased, the solenoid valves predictably closed as the input voltage was reduced below the aforementioned threshold.

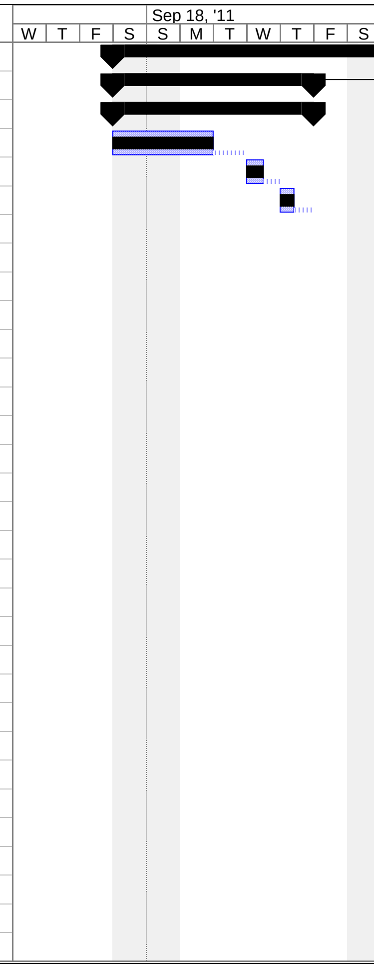
Comments

These tests indicate that the solenoid valves can be supplied with a less than anticipated voltage if desired. However, if it is more convenient to drive them at a voltage level somewhere between 4V – 12V, that will work equally as well. This will be discussed and finalized prior to the derivation of the PCB versions of the low-side drive BJT interface circuits which will be ultimately used to drive these devices (and also potentially the servo motor for the marker dropper mechanism).

5.3 Summary of Test Plan Status

6 Schedule

ID	Task Name	Start	Finish	Resource Names
1	**Autonomous Underwater Vehicle (AUVSI RoboSub Competition)**	**Fri 9/16/11**	**Fri 4/27/12**	
2	**Ramp-up (Analysis/Synthesis)**	**Fri 9/16/11**	**Thu 9/22/11**	
3	**Needs Analysis and Specification**	**Fri 9/16/11**	**Thu 9/22/11**	
4	Individual team composition	Fri 9/16/11	Tue 9/20/11	
5	Combine submissions	Tue 9/20/11	Wed 9/21/11	
6	Submit combined needs analysis	Wed 9/21/11	Thu 9/22/11	
7	**System Design (Development)**	**Fri 9/30/11**	**Wed 3/28/12**	
8	Professional engineering assignment	Mon 10/31/11	Mon 10/31/11	
9	**Project proposal**	**Fri 9/30/11**	**Thu 10/20/11**	
10	Brainstorming	Fri 9/30/11	Fri 10/7/11	
11	Project executive summary	Mon 10/17/11	Wed 10/19/11	
12	Introduction	Sat 10/8/11	Fri 10/14/11	
13	Proposed design (block diagram)	Fri 9/30/11	Fri 10/7/11	
14	Statement of work	Sat 10/8/11	Fri 10/14/11	
15	Risk assessment	Sat 10/8/11	Fri 10/14/11	
16	Qualifications and responsibilities of the team	Fri 9/30/11	Fri 10/7/11	
17	Schedule	Mon 10/17/11	Wed 10/19/11	
18	Budget estimate	Sat 10/8/11	Fri 10/14/11	
19	Deliverables	Sat 10/8/11	Fri 10/14/11	
20	Submission	Thu 10/20/11	Thu 10/20/11	
21	Project management	Thu 10/20/11	Fri 11/4/11	
22	**Complete the Hull and Frame of the AUV**	**Thu 10/20/11**	**Fri 12/16/11**	
23	Develop a Pro/Engineer Model of the Finalized Hull and Frame D	Thu 10/20/11	Fri 11/4/11	Eric Sloan
24	Order Components for Hull and Frame	Mon 11/14/11	Tue 11/15/11	Eric Sloan
25	Manufacture and Assemble the Hull and Frame	Thu 11/17/11	Tue 12/6/11	Eric Sloan
26	80/20 T-Slotted Frame	Thu 11/17/11	Fri 11/18/11	Eric Sloan
27	Hull Supports	Fri 11/25/11	Fri 11/25/11	Eric Sloan
28	Acrylic Hull	Fri 11/25/11	Sun 11/27/11	Eric Sloan
29	Aluminum End Caps	Mon 11/28/11	Mon 11/28/11	Eric Sloan
30	Camera Enclosures	Mon 11/28/11	Tue 11/29/11	Eric Sloan
31	SEACON Connectors	Wed 11/30/11	Thu 12/1/11	Eric Sloan
32	Test for Watertight Integrity	Fri 12/16/11	Fri 12/16/11	All ME



Project: Updated_Schedule_11_29_20
Date: Wed 2/13/02

Task	Milestone	External Tasks
Split	Summary	External Milestone
Progress	Project Summary	Deadline

ID	Task Name	Start	Finish	Resource Names
33	**Develop the Computer Vision System**	**Fri 10/28/11**	**Wed 3/28/12**	
34	Develop the Pre-Processing Module	Fri 10/28/11	Wed 11/2/11	Ryan Kopinsky
35	Design a Color Filter Module	Fri 11/11/11	Wed 11/16/11	Ryan Kopinsky
36	Design the Path Detection Module	Fri 11/25/11	Wed 11/30/11	Ryan Kopinsky
37	Design the Size Detection Module	Fri 1/13/12	Wed 1/18/12	Ryan Kopinsky
38	Design the Navigation Module	Thu 2/2/12	Tue 2/14/12	Ryan Kopinsky
39	Design the Shape Detection Module	Fri 2/24/12	Wed 3/14/12	Ryan Kopinsky
40	Design the Task Control Module	Tue 3/6/12	Wed 3/28/12	Ryan Kopinsky
41	**Develop the Guidance System**	**Thu 11/3/11**	**Thu 3/15/12**	
42	Develop a system to capture IMU data	Thu 11/3/11	Wed 11/9/11	Antony Jepson
43	Design a system to capture depth sensor data	Thu 2/16/12	Thu 3/1/12	Antony Jepson
44	Design a system to capture AUV heading and locate pinger	Thu 3/1/12	Thu 3/15/12	All ECE
45	**Develop Software to Control the Thrusters**	**Mon 11/14/11**	**Fri 3/23/12**	
46	Develop Software to Control the Thrusters	Thu 12/22/11	Thu 3/22/12	Hang Zhang,Eric Sloa
47	Order SeaBotix BTD150 Thrusters	Mon 11/14/11	Tue 11/15/11	Eric Sloan
48	Attach Thrusters to Frame of AUV and Complete General Maneu	Mon 12/19/11	Fri 3/23/12	Eric Sloan
49	**Complete the Compressed Air Distibution System for the AUV**	**Thu 10/20/11**	**Wed 2/15/12**	
50	Select a Compressed Air tank	Fri 1/20/12	Mon 1/23/12	All ME
51	Select a Pressure Regulator	Fri 1/20/12	Mon 1/23/12	All ME
52	Select Solenoid Valves and Pressure Lines/Tubing	Thu 10/20/11	Mon 10/24/11	All ME
53	Order Components for the Compressed Air Distribution System	Tue 1/24/12	Tue 1/24/12	All ME
54	Design Mounts for Solenoid Valves	Mon 1/9/12	Thu 1/12/12	Tra Hunter,Kashief Mc
55	Manufacture and Assemble the Solenoid Valve Mounts	Fri 1/13/12	Wed 1/18/12	Tra Hunter,Kashief Mc
56	Test Proper Functionality of the Compressed Air Distribution Syst	Mon 1/30/12	Tue 1/31/12	Tra Hunter,Kashief Mc
57	Install the Compressed Air Distribution System on the AUV and In	Thu 1/12/12	Wed 2/15/12	Tra Hunter,Kashief Mc
58	**Complete the Grasp/Release Mechanism for the AUV**	**Thu 10/27/11**	**Fri 3/2/12**	
59	Develop a Pro/Engineer model of the finalized Grasp/Release Me	Thu 10/27/11	Wed 11/30/11	Eric Sloan,Tra Hunter
60	Order Components for the Grasp/Release Mechanism	Thu 12/1/11	Fri 12/2/11	Eric Sloan,Tra Hunter
61	Manufacture and Assemble the Grasp/Release Mechanism	Mon 2/20/12	Thu 3/1/12	Eric Sloan,Tra Hunter
62	Test Proper Functionality of the Grasp/Release Mechanism	Thu 3/1/12	Fri 3/2/12	Eric Sloan,Tra Hunter
63	Install the Grasp/Release Mechanism on the AUV and Integrate w	Thu 2/2/12	Thu 2/2/12	Eric Sloan,Tra Hunter
64	**Complete the Torpedo Launchers for the AUV**	**Thu 10/27/11**	**Mon 2/27/12**	

Sep 18, '11

W T F S S M T W T F S

Project: Updated_Schedule_11_29_2(
Date: Wed 2/13/02

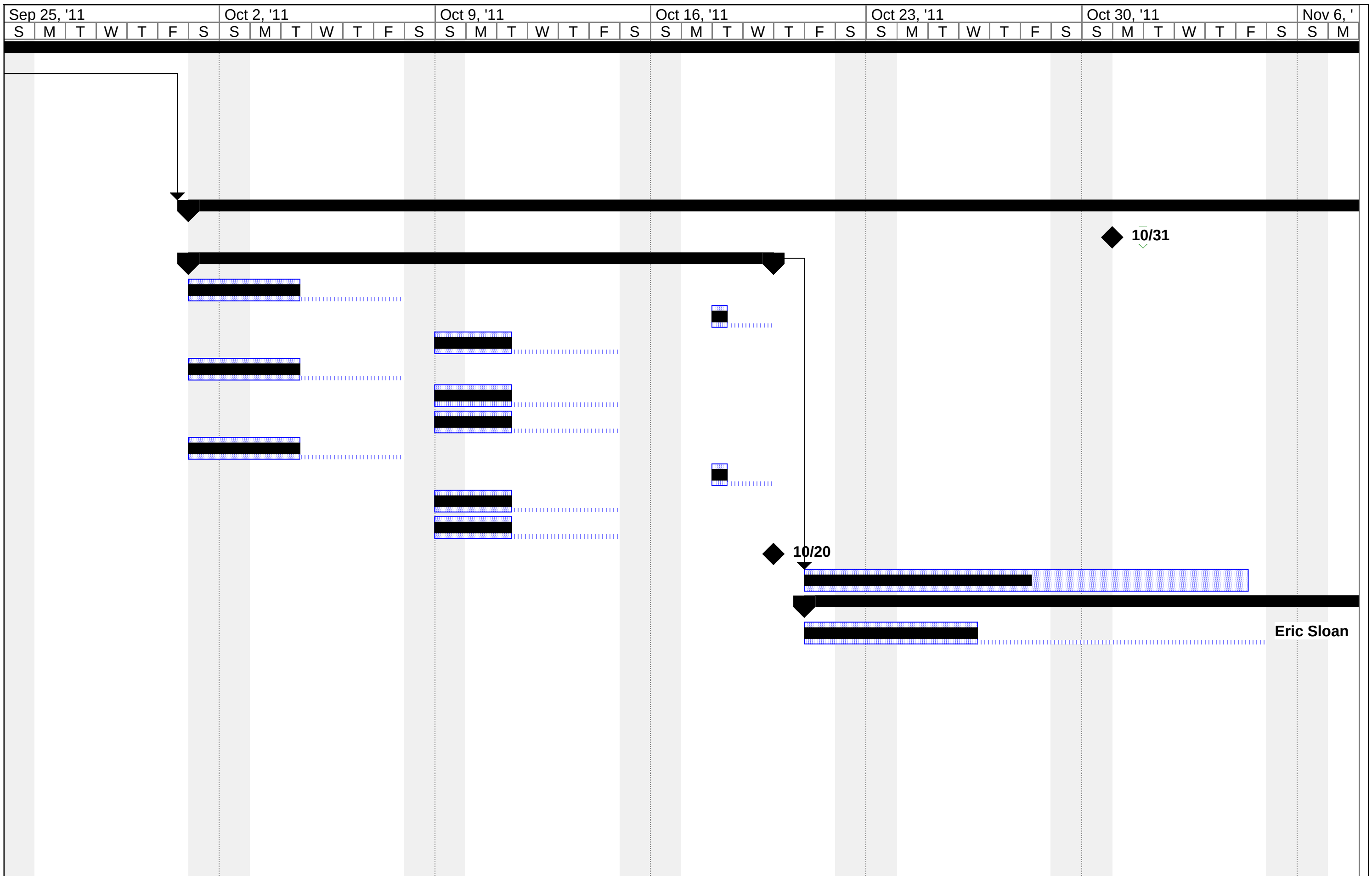
Task | Milestone | External Tasks
Split | Summary | External Milestone
Progress | Project Summary | Deadline

ID	Task Name	Start	Finish	Resource Names
65	Develop a Pro/Engineer model of the finalized Torpedo Launcher	Thu 10/27/11	Wed 11/30/11	Eric Sloan
66	Order Components for the Torpedo Launchers	Mon 11/14/11	Tue 11/15/11	Eric Sloan
67	Manufacture and Assemble the Torpedo Launchers	Wed 1/4/12	Mon 1/30/12	Kashief Moody
68	Test Proper Functionality of the Torpedo Launchers	Mon 1/30/12	Tue 1/31/12	All ME
69	Install the Torpedo Launchers on the AUV and Integrate with the I	Wed 1/11/12	Mon 2/27/12	Eric Sloan
70	Install Marker Dropper on AUV and Integrate with the Electronics	Thu 2/9/12	Mon 2/27/12	Eric Sloan
71	System Level (Conceptual) Design Review	Thu 11/17/11	Thu 11/17/11	
72	System Testing/Verification	Fri 1/13/12	Fri 4/27/12	
73	Documentation and Review	Fri 1/13/12	Fri 4/27/12	

Sep 18, '11: W T F S S M T W T F S

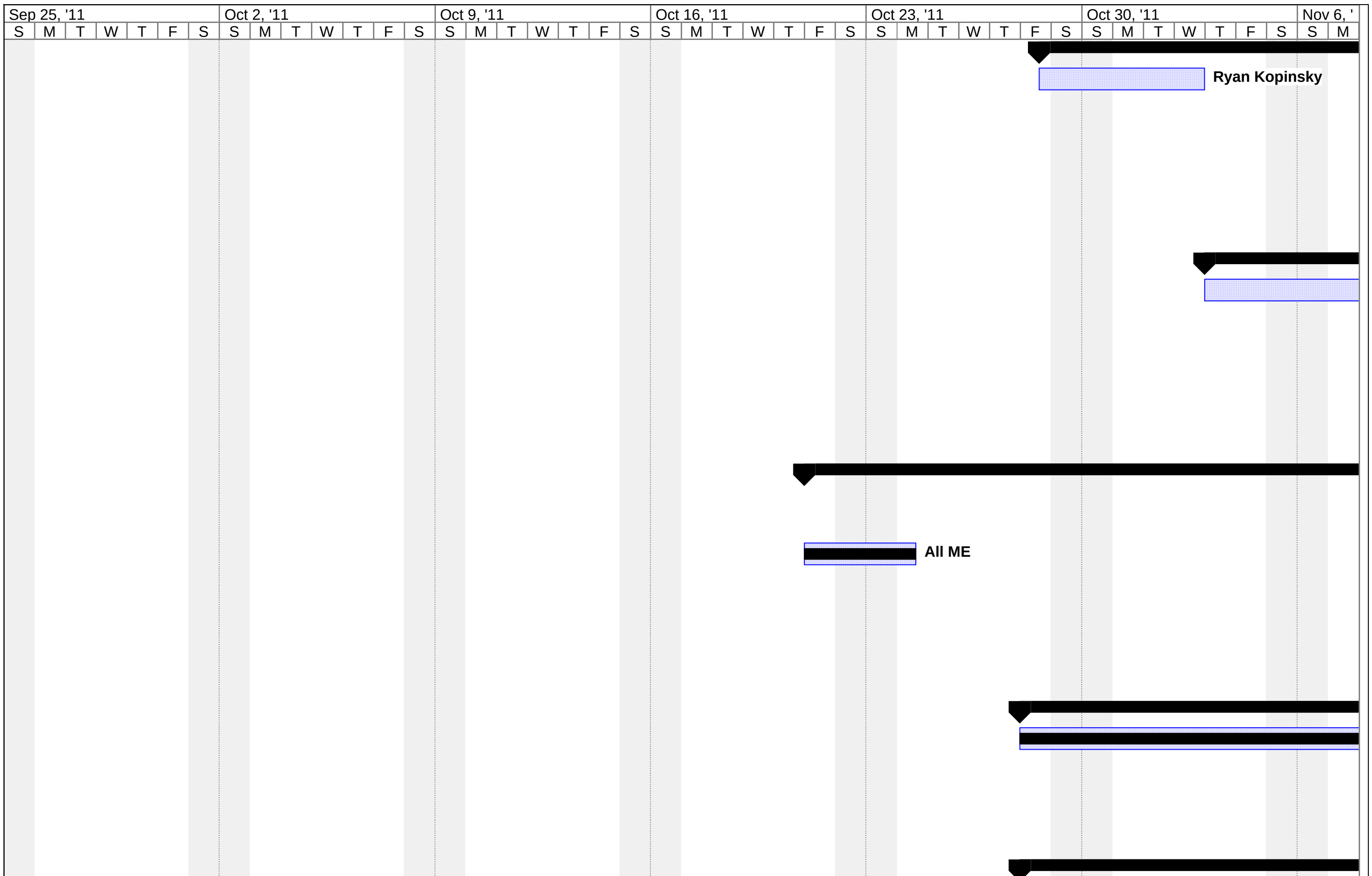
Project: Updated_Schedule_11_29_20
Date: Wed 2/13/02

Task | Milestone | External Tasks
Split | Summary | External Milestone
Progress | Project Summary | Deadline



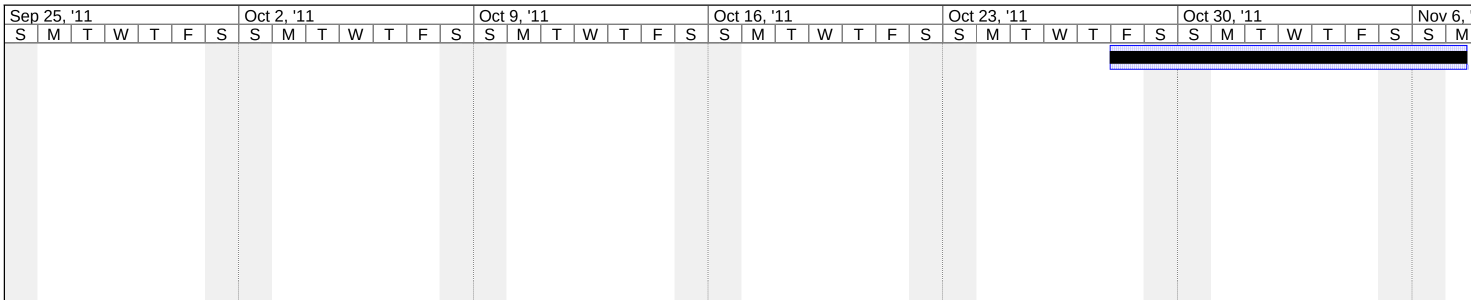
Project: Updated_Schedule_11_29_2(
Date: Wed 2/13/02

Task		Milestone		External Tasks	
Split		Summary		External Milestone	
Progress		Project Summary		Deadline	

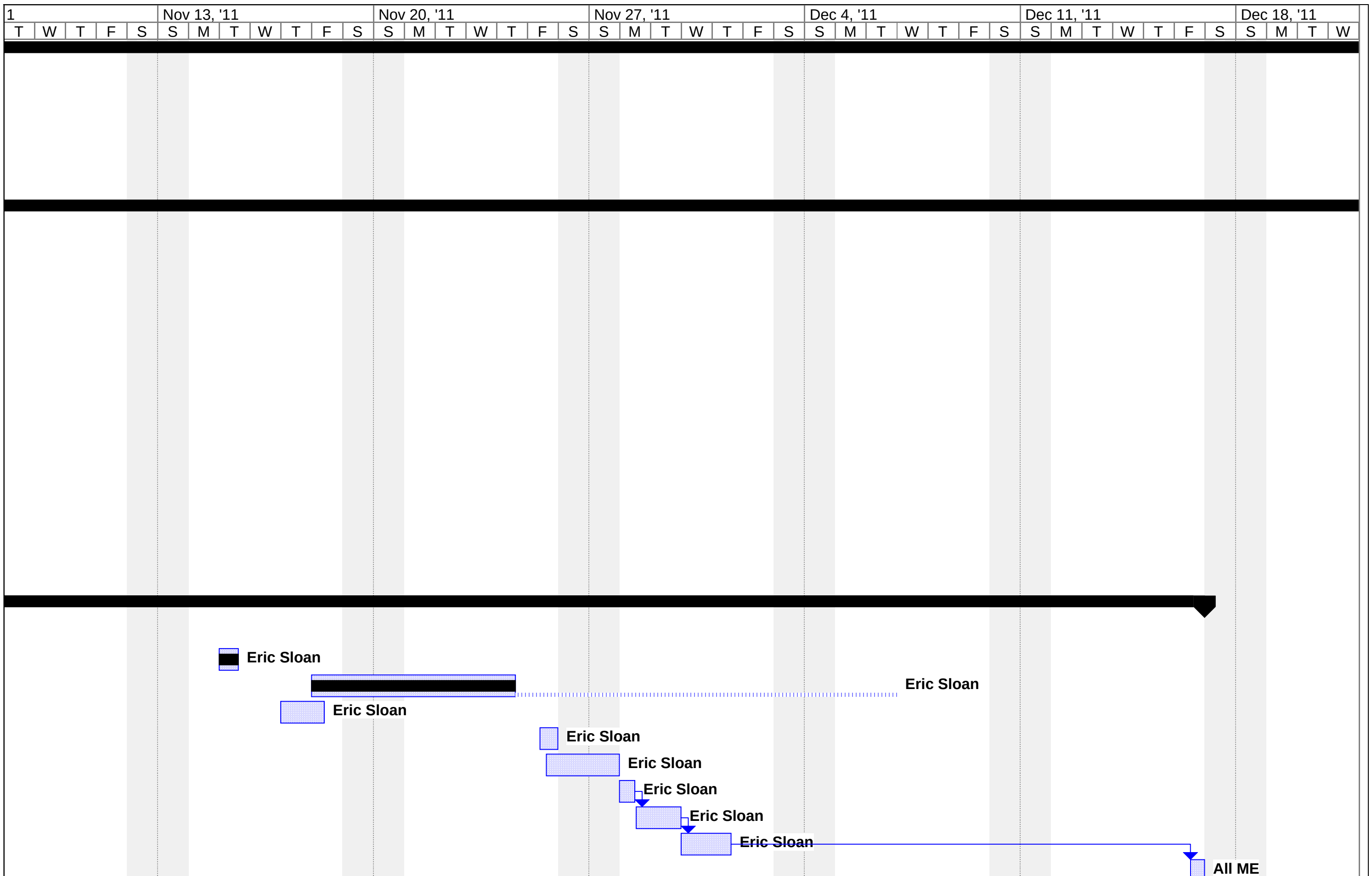


Project: Updated_Schedule_11_29_20
Date: Wed 2/13/02

Task		Milestone		External Tasks	
Split		Summary		External Milestone	
Progress		Project Summary		Deadline	

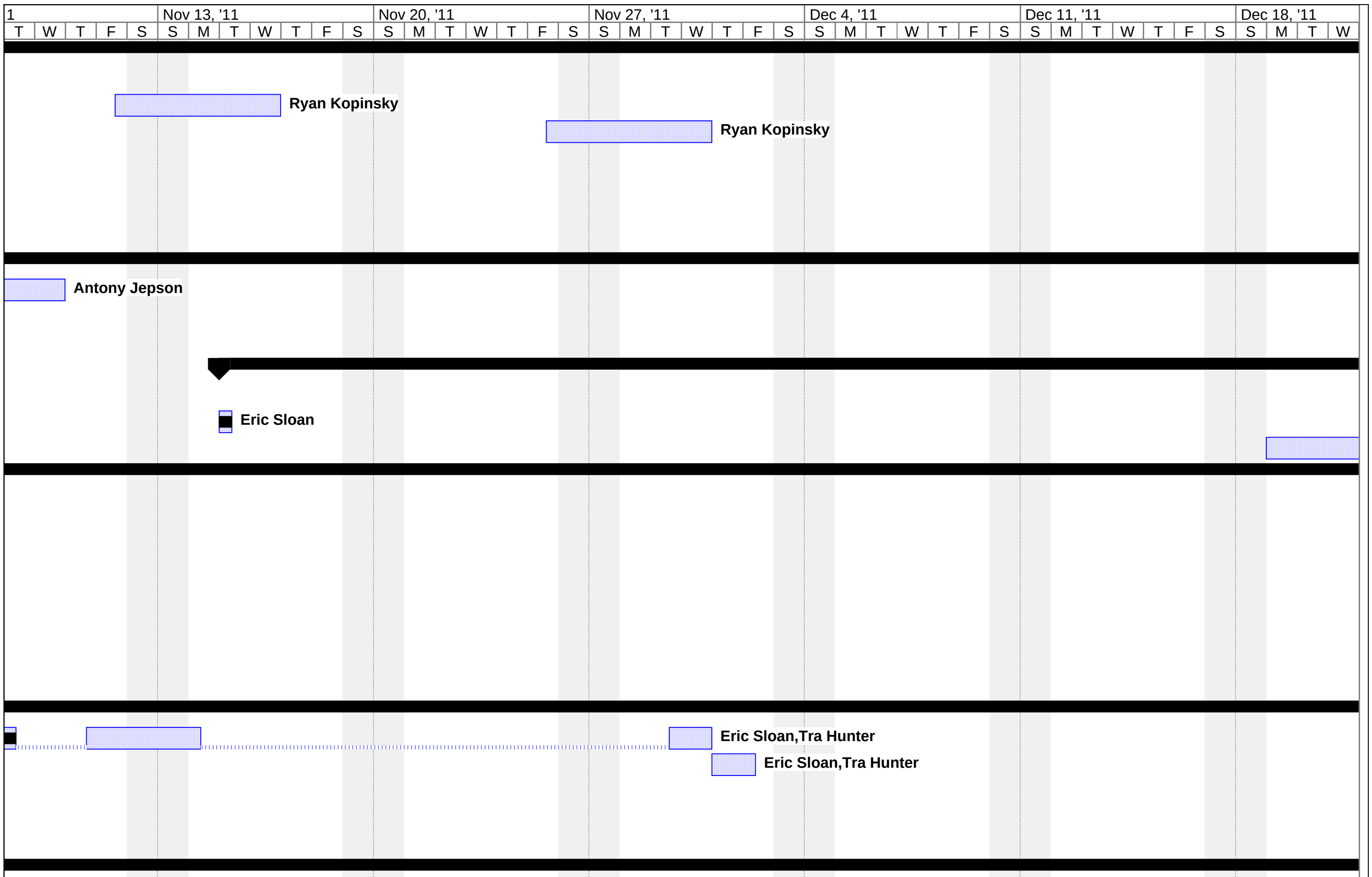


Project: Updated_Schedule_11_29_2(Date: Wed 2/13/02	Task		Milestone		External Tasks
	Split		Summary		External Milestone
	Progress		Project Summary		Deadline

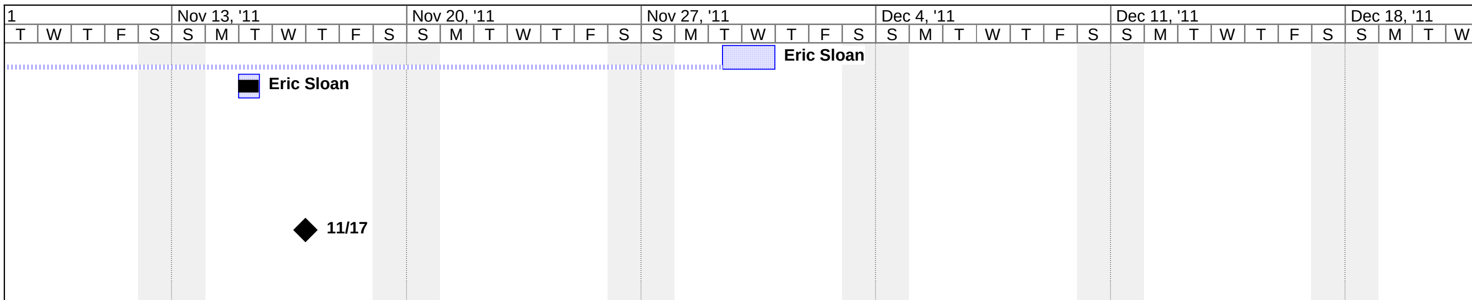


Project: Updated_Schedule_11_29_2(
Date: Wed 2/13/02

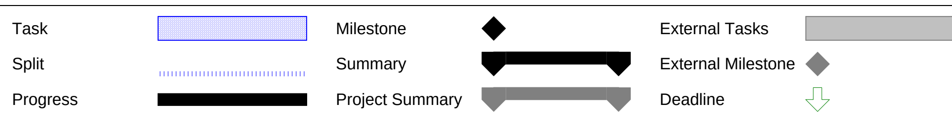
Task		Milestone		External Tasks	
Split		Summary		External Milestone	
Progress		Project Summary		Deadline	



Project: Updated_Schedule_11_29_2(Date: Wed 2/13/02	Task	Milestone		External Tasks	
	Split	Summary		External Milestone	
	Progress	Project Summary		Deadline	



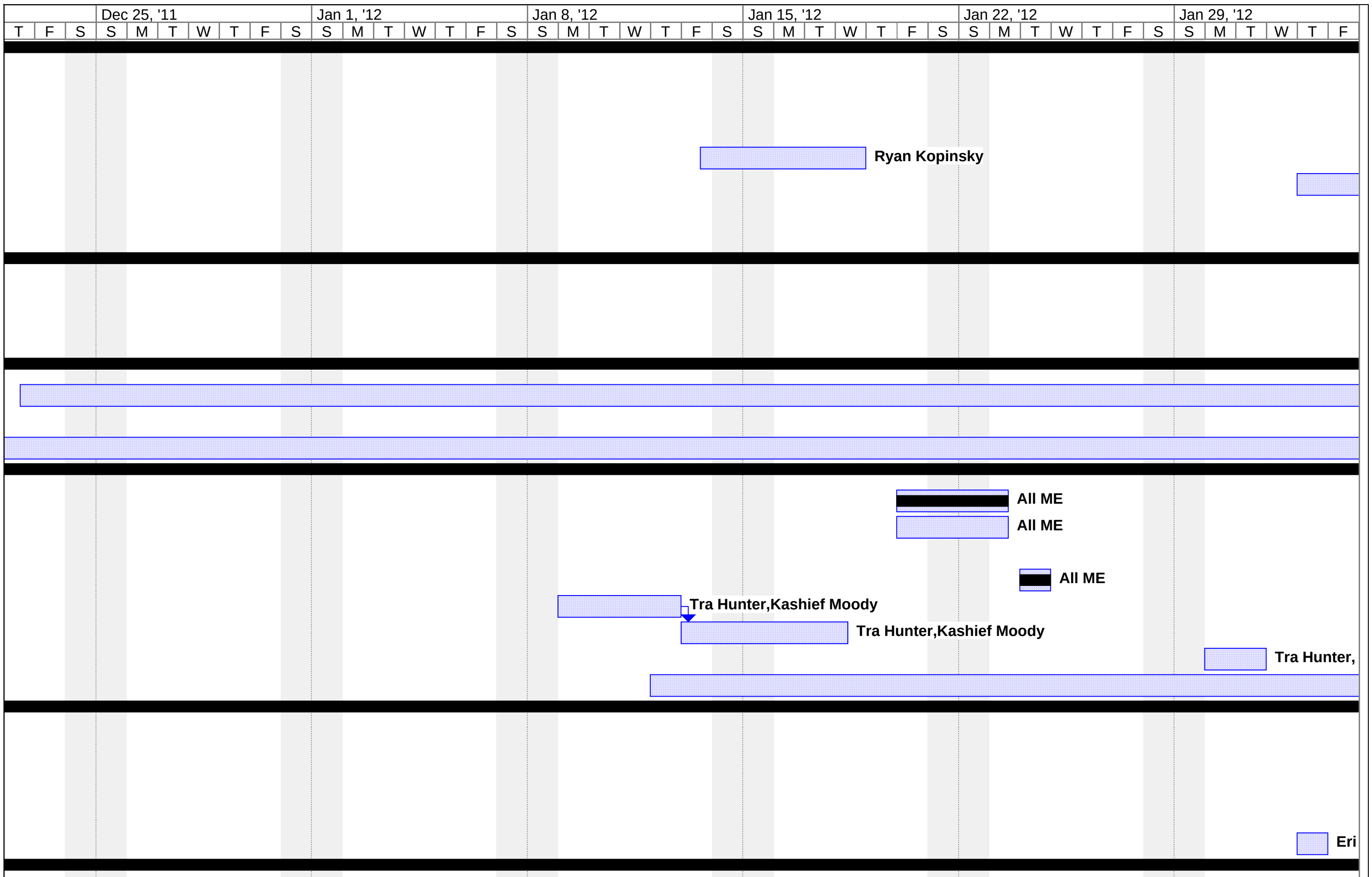
Project: Updated_Schedule_11_29_2(
Date: Wed 2/13/02



	Dec 25, '11	Jan 1, '12	Jan 8, '12	Jan 15, '12	Jan 22, '12	Jan 29, '12
T F S	S M T W T F S	S M T W T F S	S M T W T F S	S M T W T F S	S M T W T F S	S M T W T F

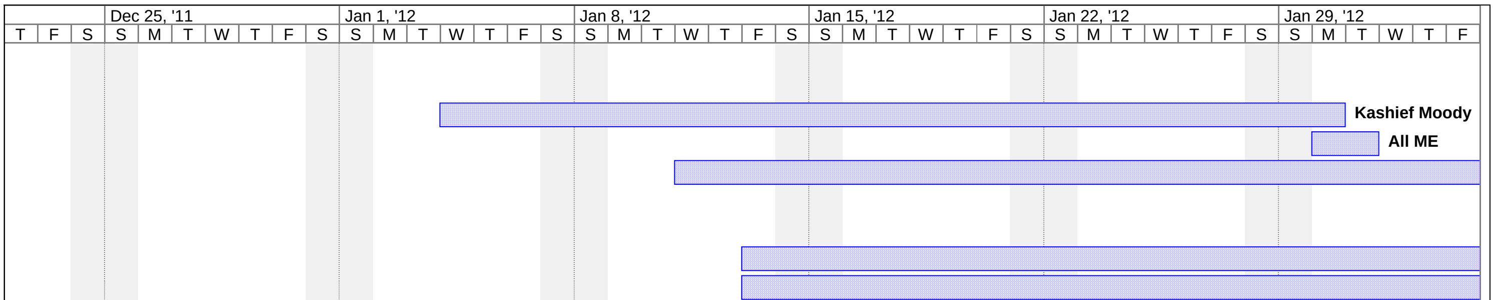
Project: Updated_Schedule_11_29_2(
Date: Wed 2/13/02

Task		Milestone		External Tasks	
Split		Summary		External Milestone	
Progress		Project Summary		Deadline	



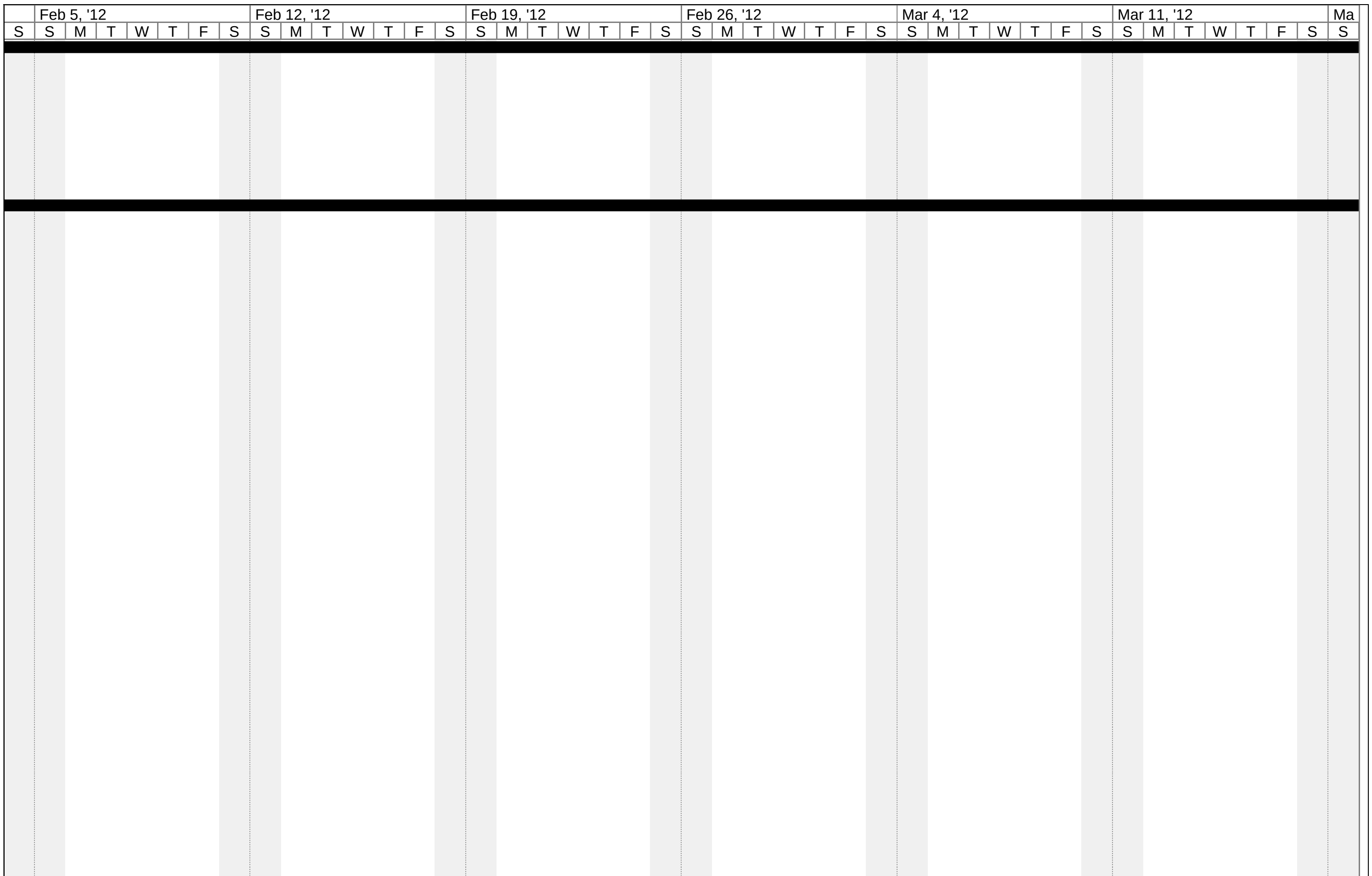
Project: Updated_Schedule_11_29_20
Date: Wed 2/13/02

Task		Milestone		External Tasks	
Split		Summary		External Milestone	
Progress		Project Summary		Deadline	



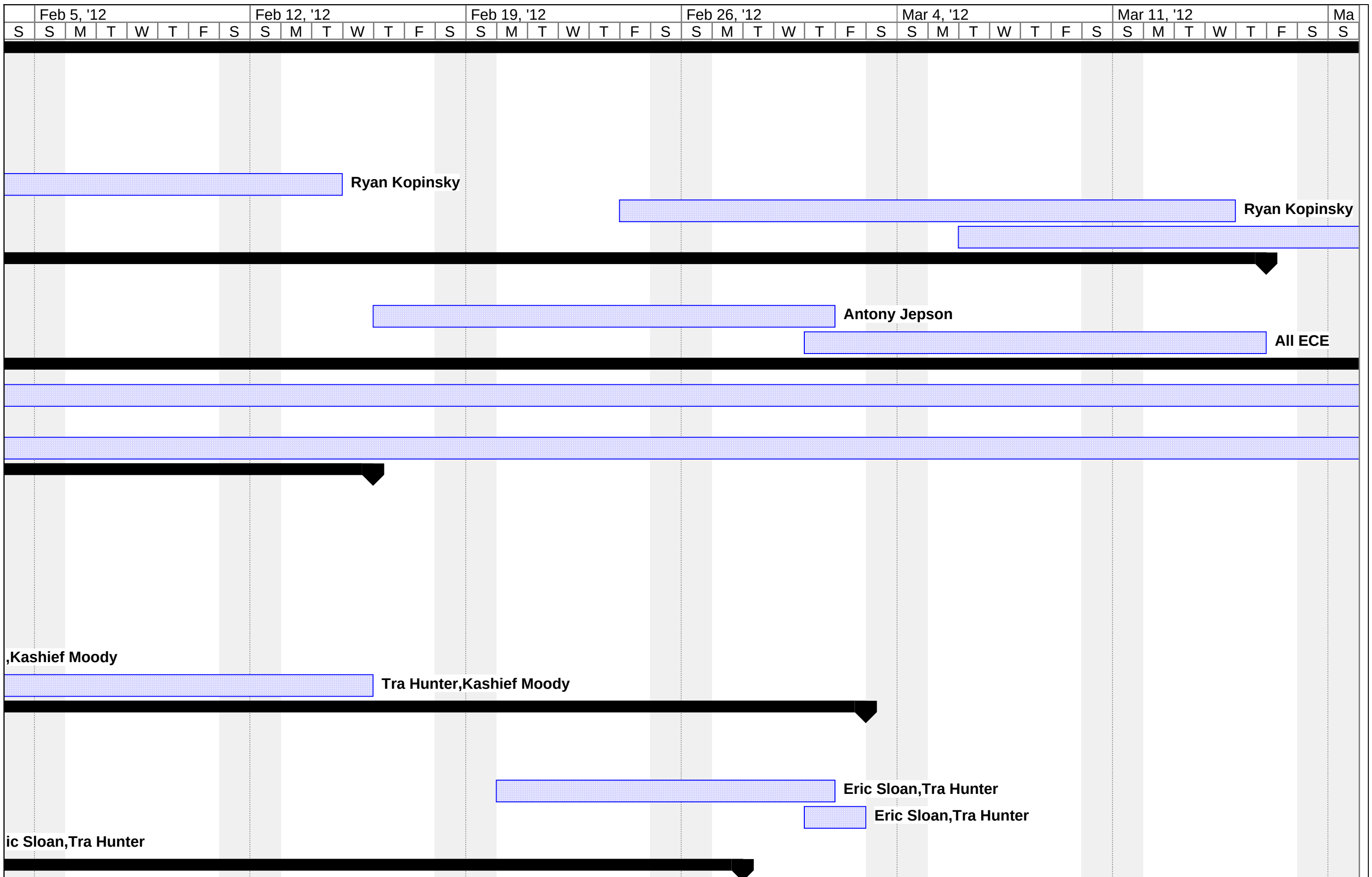
Project: Updated_Schedule_11_29_2(
Date: Wed 2/13/02

Task		Milestone		External Tasks	
Split		Summary		External Milestone	
Progress		Project Summary		Deadline	

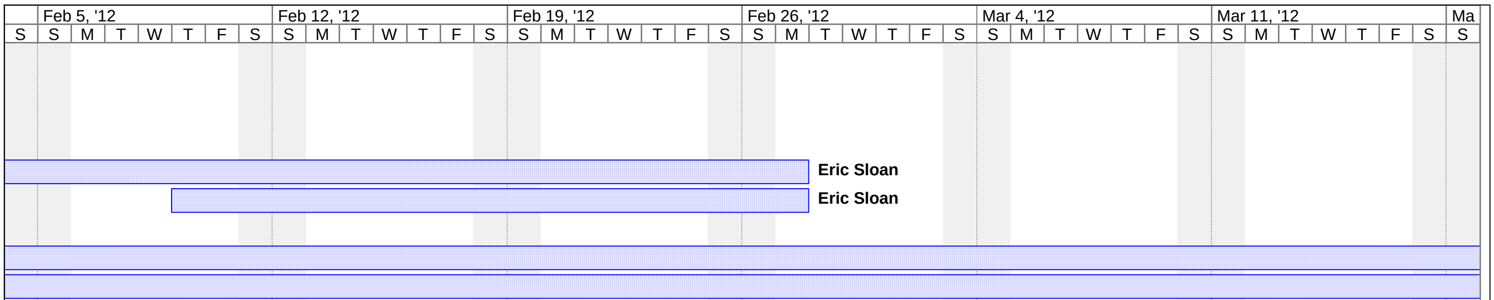


Project: Updated_Schedule_11_29_2(
Date: Wed 2/13/02

Task		Milestone	◆	External Tasks	
Split		Summary		External Milestone	◆
Progress		Project Summary		Deadline	⇩

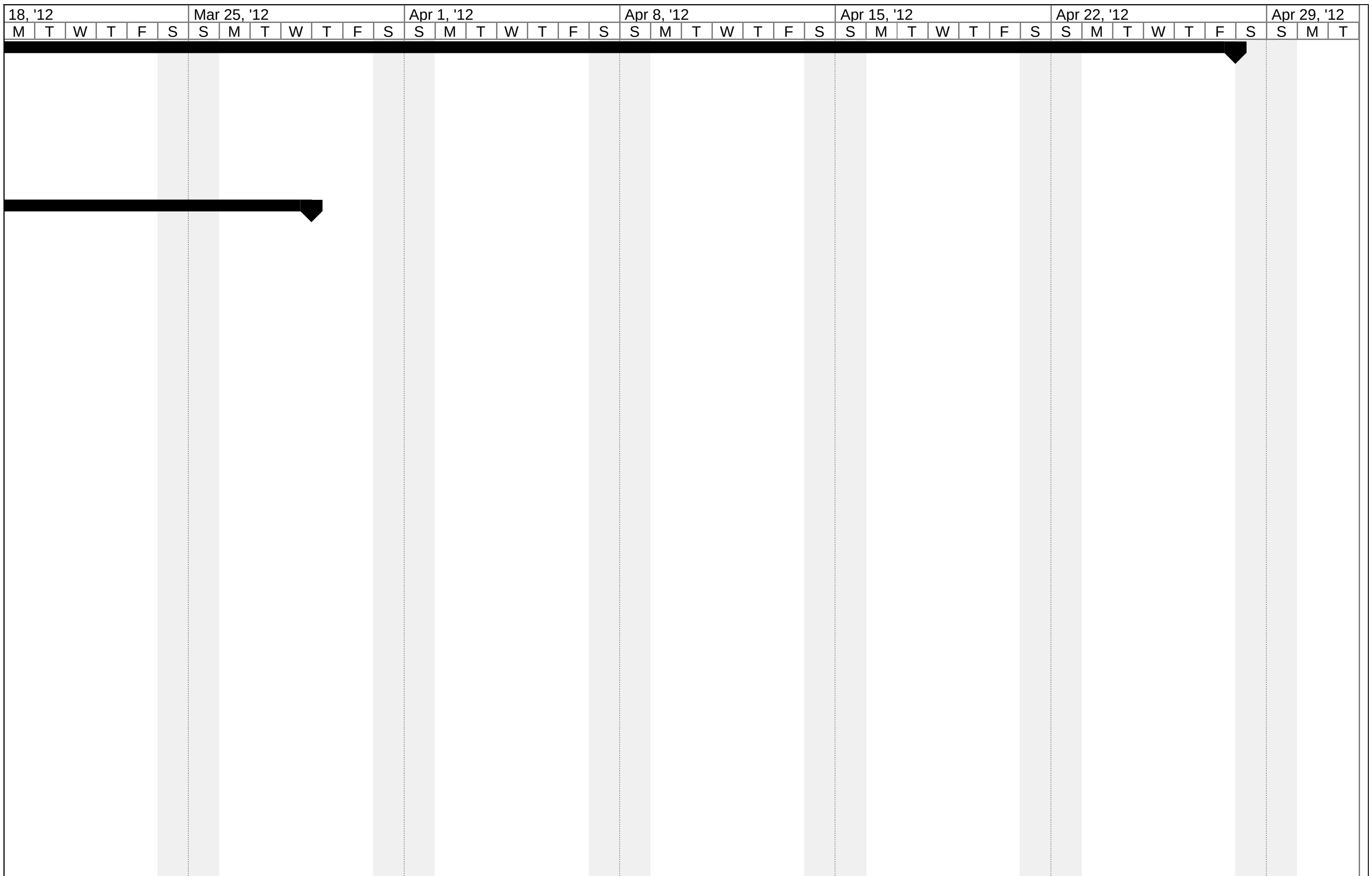


Project: Updated_Schedule_11_29_2(Date: Wed 2/13/02	Task	Milestone		External Tasks	
	Split	Summary		External Milestone	
	Progress	Project Summary		Deadline	



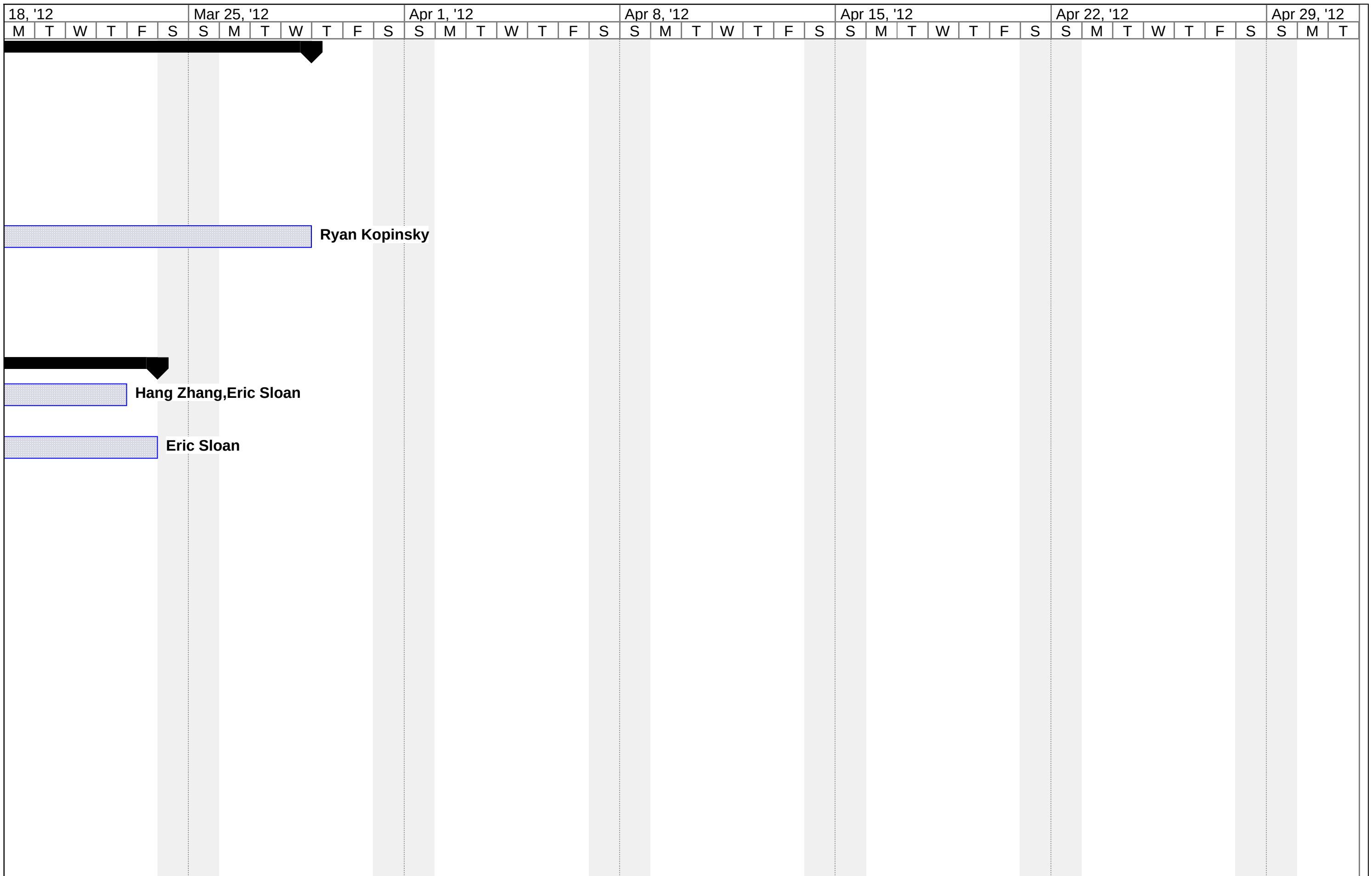
Project: Updated_Schedule_11_29_2(
Date: Wed 2/13/02

Task		Milestone		External Tasks	
Split		Summary		External Milestone	
Progress		Project Summary		Deadline	



Project: Updated_Schedule_11_29_20
Date: Wed 2/13/02

Task		Milestone		External Tasks	
Split		Summary		External Milestone	
Progress		Project Summary		Deadline	



Project: Updated_Schedule_11_29_20 Date: Wed 2/13/02	Task		Milestone		External Tasks
	Split		Summary		External Milestone
	Progress		Project Summary		Deadline

18, '12						Mar 25, '12							Apr 1, '12							Apr 8, '12							Apr 15, '12							Apr 22, '12							Apr 29, '12		
M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T

Project: Updated_Schedule_11_29_2(
Date: Wed 2/13/02

Task		Milestone		External Tasks	
Split		Summary		External Milestone	
Progress		Project Summary		Deadline	

7 Budget Estimate

Category	Details	Amount ($)
Fall Expenditures		4,479.36
Spring Expenditures	Beagleboard-xM	220.00
	Compressed Air Tank, Regulators, Gas Lines	313.71
	SQ26-01 Hydrophones (2)	415.00
	Redesigned Camera Enclosures	$52.50
	IMCL Submersible Pressure Transducer	407.47
	Servo Motor	25.00
	Miscellaneous (Nuts, Bolts, Adapters, Adhesives, Sealants, Raw Materials)	370.14
Remaining Expenditures		150.00
Travel/Shipping/Lodging Expenditures	Freight	300.00
	Flight	3,000.00
	Hotel (2 rooms, 6 nights)	2,000.00
	Food ($30/day/pp)	1,200.00
	Miscellanous	200.00
Competition Fee		500.00
Total		**13,633.18**
Current Budget		$9,433.00
Remaining Balance		-$4,200.18

8 Risks

This section has been largely condensed due to improved knowledge about AUV components. Only the main risks will be listed.

8.1 Electrical

8.1.1 Electronics overheat due to insufficient heat dissipation system

Risk	Electronics overheat due to insufficient heat dissipation system
Probability	Low
Consequence	Moderate
Strategy	Install a battery-powered fan inside the hull in order to circulate the heat away from the electronics and into the surrounding air inside the hull. The fan would induce forced convection, and provide the necessary heat extraction from the electronics.

Description

The electronics could potentially overheat if the heat dissipation system (i.e. conduction through the aluminum platform, thin aluminum end cap walls, and into the surrounding environment) is insufficient.

Probability: **Low**

Since the heat generation from the electronics is expected to be low however—particularly due to a relatively low power demand from the thrusters, the relatively large convection coefficient of flowing water against the end caps, and the thin wall through which the heat will need to conduct in order to escape the hull and reach the external environment—the probability of a fan being required is relatively low.

Consequences: **Moderate**

While overheating could cause the electronics to malfunction or burn out, and would thus provide a moderate consequence, a simple solution is available.

Strategy

Should the aluminum platform inside the hull prove unsuccessful in efficiently and effectively dissipating heat away from the electronics and into the surrounding salt water environment via conduction, a battery powered fan will be installed inside the hull at one end, and will serve to induce forced convection and circulate heat away from the electronics.

Test Plan

Thermocouples or thermistor-integrated wheat stone bridges will potentially be able to be used in order to determine the temperature at various locations inside the hull during an underwater test or operating level. There may be a way the design team can have the temperature readings captured and stored as a .txt file, so that a plot of temperature versus time can be obtained and analyzed following an underwater test. If this proves to be unsuccessful or excessively challenging, a theoretical FEM model of the closed system will be derived based on the theoretical average heat generation rates from each of the electrical components (mainly a function of the average required thrust from each thruster over a period of time).

8.1.2 Software Bugs May Cause Operational Failure of Some Tasks

Risk	Software bugs may cause operation failure of some tasks
Probability	Very High
Consequence	Severe
Strategy	1.Careful design of the software system 2.Extensive debugging and testing 3.Simulate the competition environment during tests

Description

The mission control software system is a large program that coordinates all the components onboard and determines what the AUV should perform at each stage. This program will be designed and written by our team. Bugs are inevitable in this large and complex program. Some bugs may be minor, but some may lead to operation failure during performing some tasks. For instance, a communication error between the camera and the mission control system may lead to an incorrect path for the AUV.

Probability: **Very High**

The probability of this risk is very high due to the fact that software bugs are almost inevitable in such a complex program.

Consequences: **Severe**

The consequences of the software system bugs varies, some bugs maybe minor, some maybe moderate, and some can be severe, but none of the bugs should produce catastrophic consequence which disables the entire system of the AUV (horrible design of the control system or hardware failure). Severe consequence may occur when some bugs cause failure in performing certain tasks.

Strategy

1. Careful design of the software system. During design stage, the mission control system should be very carefully designed to minimize the probability of severe bugs.
2. Extensive debugging and testing. At developing stage, debugging and testing should be conducted as much as possible. Testbench could be designed to conduct tests in the lab.
3. Simulate the competition environment during tests. After significant amount of testing in lab, the system will be put into a simulated competition environment to test possible bugs and functionalities.

8.1.3 Path Detection Failure

Risk	Failure to detect the direction of the path
Probability	Moderate
Consequence	Catastrophic
Strategy	1. Extensively test and debug code in order to minimize failure 2. Develop an algorithm to get back on track in the case of failure

Description

The Path Detection module may fail to detect the correct direction of the path (segments).

Probability: **Moderate**

The probability of failure is moderate considering the complexity of this module. This module is most likely the most complex software module in the Computer Vision system and will require a lot of effort to complete.

Consequences: **Catastrophic**

The consequence of failure of the Path Detection module is catastrophic. If the module fails to detect the correct direction of the path, the vehicle will go off-course and will most likely struggle to come back on track. The completion of the obstacle course heavily relies on this module and failure is, therefore, catastrophic.

Strategy

1. In order to avoid getting off-track, the Path Detection module will be extensively tested and debugged. The software engineers will spend a great deal of time on this module to ensure accurate tracking of the path segments.
2. If the RoboSub goes off-track due to failure of the Path Detection module, a custom algorithm will need to kick in to bring the RoboSub back on track. The custom algorithm will be part of the Path Detection module.

8.2 Mechanical

8.2.1 Vehicle density greater or less than optimal target density

Risk	Vehicle density greater of less than optimal target density
Probability	Low
Consequence	Moderate
Strategy	Symmetrically add material of greater or less density than the vehicle's target density to either side of the bottom (greater density material) or bottom (lower density material) of the AUV until the nominal system density and balance—particularly the neutralization of the pitch angle—has been obtained.

Description
The density of the completed AUV might potentially be a considerable amount greater or less than the desired target density of the system.

Probability: **Low**
Due to the very detailed Pro/Engineer model which contains accurate density values for each of the carefully dimensioned parts, the projected system density, weight, center of mass, and inertia values should yield accurate and trustworthy projections. Thus, the probability of the end product density not being near the nominal value is relatively low.

Consequences: **Moderate**
The consequence of not having the proper system density is deemed moderate because it would either yield an increased demand from the thrusters, or worse, a vehicle that would naturally sink when the kill switch is activated and the thrusters are shut off.

Strategy
If the vehicle density ends up being greater or less than the optimal target density to a significant extent, material of greater or less density than this target density will be added to either side of the bottom or top of the AUV, depending on the density of the added ballasting material, until the nominal system density and balance has been obtained. Symmetry and balance will be maintained through this process.

Test Plan
While the detailed Pro/Engineer calculations should provide an accurate theoretical projected system density should the design be followed as carefully planned, the actual system density will be derived using one or both of the following methods:

1) Weigh the vehicle on a scale, and calculate the density based off the known volume of each of the components of the vehicle (which will all eventually be implemented into the Pro/Engineer model).

2) Perform a water test where the vertical side thrusters would be progressively given a greater PWM duty until neutral buoyancy is achieved. Then, derive the density based off the corresponding thrust and the known density of the pool water in which the vehicle will be tested.

This experimentally derived density calculation will assist the design team in determining whether the addition of "dummy mass" is necessary, and if so, the quantity (in volume or mass) that is required.

8.2.2 Hull Leakage Post-SEACON Connector Implementation

Risk	The hull experiences leakage after the integration of the SEACON™ connectors and kill switch
Probability	Low
Consequence	Severe
Strategy	1. Apply Loctite thread sealant to the threads of all the SEACON™ connectors, as well as the kill switch. 2. If necessary, carefully apply 100% silicone caulking to the interface between the end caps and the connectors in order to yield a reliable seal.

Description

The hull could potentially leak following the addition of the SEACON™ connectors and kill switch, thus posing a threat to the integrity of the lithium-ion batteries and electronics.

Probability: **Low**

The SEACON™ connectors (i.e. female plugs) and mechanical kill switch will be tightly screwed into their corresponding tapped holes in the removable end caps of both the hull and camera enclosures. The addition of thread sealant should enable these connectors to remain tightly screwed throughout the lifespan of the AUV, while also serving to prevent water from penetrating through the threads and into the hull. Furthermore, these types of underwater wet mate connectors have been used successfully in competitions past and thus provide a sense of reassurance and confidence moving forward.

Consequences: **Severe**

The consequence of the watertight nature of the hull being compromised during an underwater test is severe. If water were to enter the hull with the electronics and battery packs inside as well, the circuits could short, sparks could fly, and the electronics could be destroyed. For these reasons, another water tight test will be performed on both the hull and camera enclosures following the addition of these components prior to permanently installing the electronics inside these enclosures.

Strategy

1. Apply Loctite thread sealant to the threads of all the SEACON™ connectors, as well as the kill switch.
2. If necessary, carefully apply 100% silicone caulking to the interface between the end caps and the connectors in order to yield a reliable seal.

8.2.3 Revised Camera Enclosure Leakage

Risk	Redesigned Camera Enclosures Experience Leakage
Probability	Low
Consequence	Severe
Strategy	1. Carefully and cleanly apply 100% silicone caulking to the interfaces between the fixed outer end caps of the revised camera enclosures and the acrylic box to ensure a clean, reliable water tight seal.

Description

The remodeled camera enclosures could experience leakage when submerged—especially at significant depths.

Probability: **Low**

The probability of the redesigned camera enclosures leaking is low—significantly lower than the original design. Several lessons were learned from mistakes made during the development of the original camera enclosures, and these factored into key design changes made for the new enclosures. The reliability of the new enclosures should be substantially increased for the following three reasons in particular:

1) The new enclosures will feature pre-made acrylic boxes (display cases), enabling the avoidance of leakage through the edges of the enclosures—a common theme in the original design. Thus, the only region that will need to be properly sealed will be the four edges where the updated outer camera enclosure end caps will be rigidly attached at the open face of the acrylic boxes.
2) Experience in proper caulking mechanics has been gained, and thus, the result should be much cleaner silicone beads and more reliable seals.
3) The remodeled enclosures feature no penetration of the acrylic faces whatsoever. The mounting will be done solely using adhesive, and the camera mounts will be installed completely inside each of the respective enclosures—the net result being no bolt holes or threads through which water could leak (even past neoprene sealing washers as used in the original design).

Consequences: **Severe**

Leakage of the camera enclosures could compromise the integrity of the Logitech C615 web cameras, and thus could absolve the entire computer vision system—a critical aspect of the AUV's navigation system.

Strategy

Carefully and cleanly apply 100% silicone caulking to the interfaces between the fixed outer end caps of the revised camera enclosures and the acrylic box to ensure a clean, reliable water tight seal.

8.3 Budget

Risk	Underestimate of budget which results in sufficient fund
Probability	Moderate
Consequence	Severe
Strategy	1. Carefully estimate our budget 2. Avoid unnecessary purchases 3. Seek additional sponsorship

Description

An under estimation of the team's budget can significantly delay our design process. So far, our team has a total of $9,433.00 funds available, thanks to sponsorship by Dr. Shih and the NEEC. However our budget exceeds the total amount of funds available after including travel expenses. And we are currently at around $4,200.18 short if the travel expense is estimated to be around $6,500. Therefore, our team must seek for additional sponsorship.

Probability: **Moderate**

The risk of having an under estimation of team's budget is moderate since the prices for most expensive parts we need to purchase such as thrusters and hydrophones are known. The probability of a sudden increase of prices for these parts before we order them is low. Under estimation of the travel expense can be high due to the fact that air tickect price, hotel price changes very often.

Consequences: **Severe**

The consequence of underestimate our budget is severe. If the budget is under estimated, we will have to spend additional time (which may not be in the plan) to seek for sponsiship. And this may significantly delay our design process. The underestimation may also result in having insufficient funds to purchase necessary parts, which will pose a big risk of not being able to complete the design.

Strategy

1. To minimize the risk of having an understimation of buget, we need to carefully estimate all the expenses required by this project. When estimating the budget, a small amount of safe budget should be included for emergencies.
2. For every purchase, we need to have a concious choice of what purchase is really necessary so that we can avoid unecessary purchases. And our team decides to only purchase the parts that are necessary to our design at current stage. Therefore,we will not purchase the parts that may be needed in the future at this moment. In this way, if the design is modified, we will not have to risk that certain parts purchased may not be needed in the future.
3. Since our team already have a negative budget, we will continously seek for sponsorship. We are currently seeking sponsorship from ARM to ask for a donation on the Beagleboard-xM.

8.4 Summary of Risk Status

Presently, most risks have concrete mitigation statements. The largest threat to the successful completion of the project is modification to the competition rules later on in this semester. The final competition rules will be soon be published so this threat is quickly diminishing as time moves forward (as large change close to the competition will affect all the teams).

Team Robosub has focused on making a venerable and modular AUV that can address most of the tasks previously undertaken at older competitions. However, if a major change is announced with the 2012 Competition, the team will need to back-pedal to address the change.

Team Robosub is certain that the competition ruling committee is aware of this, given that interest request forms were available back in October. The team is confident that the modular design presented in this document thoroughly mitigates this issue and is adaptable to most changes that will be required. The only limitation with this stance is the budget. Components are expensive; to mitigate this issue, we will continue to seek outside funding and sponsorship.

A large quantity of time has been invested in this project to reduce risk. Awareness is the largest component in mitigating risk and by identifying all the major risks with the design of the AUV, Team Robosub is confident that most risks can be avoided.

9 Conclusion

Team Robosub consists of six solid engineers from the ECE and ME departments at the FAMU-FSU College of Engineering. This team aims to compete at the Association for Unmanned Vehicle Systems International Robosub competition in July 2012. The goal of this competition is to further the exploration of engineering fields typically not explored at University (or even in high school). This team is confident that the provisions listed in this document have been completed with 100% dedication and vigor.

Slight design modifications may take place during the construction to overcome unforeseen obstacles. Most of the information not listed in this document will be derived via experimental testing and these results create a feedback loop that will be used to continually improve the design. Detailed density, center of mass, weight, inertia, and power consumptions calculations enforce the decisions made by the team.

The submission of this report marks the completion of the initial design process and the beginning of construction – a milestone in the Teams project roadmap. The design team endeavors to complete the design successfully by March 2012 and will secure additional funding to enable the team to travel to the competition in San Diego, CA.

10 Appendix

10.1 Code

Listing 1: Phidget Orientation Detection

```
// Orientation
// Author: Antony Jepson
// This program determines the orientation of the spatial relative to gravity.
// It uses routines from the Spatial-sample.c program included with the Phidget
// example tarball.

#include <stdio.h>
#include <math.h>
#include <phidget21.h>
#include "./orientation.h"

#define r2(x) ((x) * (x))

/* According to the product manual, if the spatial is placed perpendicular to
 * the ground, then the following combinations of acceleration readings
 * determine the orientation, given that the USB port faces the back of the
 * spatial.
 * Direction -- Axis {0, 1, 2}
 * Left      -- {+1, 0, 0}
 * Front     -- {0, +1, 0}
 * Flat      -- {0, 0, +1}
 * Flat-Back -- {0, 0, -1}
 * Back      -- {0, -1, 0}
 * Right     -- {-1, 0, 0}
 *
 * Moving the spatial
 * -
 * -
 *
 *
*/


// Assumes that only one spatial is connected.
int CCONV SpatialDataHandler(CPhidgetSpatialHandle spatial, void *userptr,
   CPhidgetSpatial_SpatialEventDataHandle *data, int count) {

        double accel_x = data[0]->acceleration[0];
        double accel_y = data[0]->acceleration[1];
        double accel_z = data[0]->acceleration[2];

        double rx = accel_x;
        double ry = accel_y;
        double rz = accel_z;
  double r  = sqrt(r2(rx) + r2(ry) + r2(rz));

        // component vector angles

        double rad_arx, rad_ary, rad_arz;

        if (rz < 0) {
                rad_arx = 2 * M_PI - acosl(rx/r);
                rad_ary = 2 * M_PI - acosl(ry/r);
        } else {
                rad_arx = acosl(rx/r);
```

```
                rad_ary = acosl(ry/r);
        }

  rad_arz = acosl(rz/r);

  double deg_arx = rad_arx * 180 / M_PI;
  double deg_ary = rad_ary * 180 / M_PI;
  double deg_arz = rad_arz * 180 / M_PI;

        //printf("accel        %6f   %6f   %6f\n", accel_x, accel_y, accel_z);
        printf("vec:          %6f, %6f, %6f, %6f\n", rx, ry, rz, r);
        printf("rad_ang_vec: %6f, %6f, %6f\n", rad_arx, rad_ary, rad_arz);
        printf("deg_ang_vec: %6f, %6f, %6f\n", deg_arx, deg_ary, deg_arz);


        // Simple formula 2. Place values in a range.
        if (    is_within_range(accel_x,  tol_accel_0, 1) &&
                is_within_range(accel_y,  tol_accel_1, 0) &&
                is_within_range(accel_z,  tol_accel_2, 0))
                printf("Orient> Left\n");

        else if (is_within_range(accel_x, tol_accel_0, 0) &&
                is_within_range(accel_y,  tol_accel_1, 1) &&
                is_within_range(accel_z,  tol_accel_2, 0)) printf("Orient> Front\n");

        else if (is_within_range(accel_x, tol_accel_0, 0) &&
                is_within_range(accel_y,  tol_accel_1, 0) &&
                is_within_range(accel_z,  tol_accel_2, 1)) printf("Orient> Right side up\n
                   ");

        else if (is_within_range(accel_x, tol_accel_0, 0) &&
                is_within_range(accel_y,  tol_accel_1, 0) &&
                is_within_range(accel_z,  tol_accel_2, -1)) printf("Orient> Upside down\n"
                   );

        else if (is_within_range(accel_x, tol_accel_0, 0) &&
                is_within_range(accel_y,  tol_accel_1, -1) &&
                is_within_range(accel_z,  tol_accel_2, 0)) printf("Orient> Back\n");

        else if (is_within_range(accel_x, tol_accel_0, -1) &&
                is_within_range(accel_y,  tol_accel_1, 0) &&
                is_within_range(accel_z,  tol_accel_2, 0)) printf("Orient> Right\n");
        else
                printf("Orient> Unknown\n");


        return 0;
}

/*
        printf("Number of Data Packets in this event: %d\n", count);
        for(i = 0; i < count; i++) {
                printf("=== Data Set: %d ===\n", i);
                printf("Acceleration> x: %6f   y: %6f   x: %6f\n", data[i]->acceleration[0],
                    data[i]->acceleration[1], data[i]->acceleration[2]);
                printf("Angular Rate> x: %6f   y: %6f   x: %6f\n", data[i]->angularRate[0],
                   data[i]->angularRate[1], data[i]->angularRate[2]);
                printf("Magnetic Field> x: %6f   y: %6f   x: %6f\n", data[i]->magneticField
                   [0], data[i]->magneticField[1], data[i]->magneticField[2]);
                printf("Timestamp> seconds: %d -- microseconds: %d\n", data[i]->timestamp.
```

```
                    seconds, data[i]->timestamp.microseconds);
        }
        return 0;
*/
//}

int spatial_orientation() {
        // see spatial_simple() for comments
        int result;
        const char *err;

        CPhidgetSpatialHandle spatial = 0;
        CPhidgetSpatial_create(&spatial);
        CPhidget_set_OnAttach_Handler((CPhidgetHandle)spatial, AttachHandler, NULL);
        CPhidget_set_OnDetach_Handler((CPhidgetHandle)spatial, DetachHandler, NULL);
        CPhidget_set_OnError_Handler((CPhidgetHandle)spatial, ErrorHandler, NULL);
        CPhidgetSpatial_set_OnSpatialData_Handler(spatial, SpatialDataHandler, NULL);
        CPhidget_open((CPhidgetHandle)spatial, -1);

        printf("Waiting for spatial to be attached.... \n");
        if((result = CPhidget_waitForAttachment((CPhidgetHandle)spatial, 10000)))
        {
                CPhidget_getErrorDescription(result, &err);
                printf("Problem waiting for attachment: %s\n", err);
                return 0;
        }

        //Set the data rate for the spatial events
        CPhidgetSpatial_setDataRate(spatial, 200);

        //run until user input is read
        printf("Press any key to end\n");
        printf("Zeroing gyro.\n");
        CPhidgetSpatial_zeroGyro(spatial);
        getchar();

        //since user input has been read, this is a signal to terminate the program so we
           will close the phidget and delete the object we created
        printf("Closing...\n");
        CPhidget_close((CPhidgetHandle)spatial);
        CPhidget_delete((CPhidgetHandle)spatial);

        return 0;
}

//main entry point to the program
int main(int argc, char* argv[]) {
        //all done, exit
        spatial_orientation();
        return 0;
}
```

Listing 2: Phidget "orientation.h" include

```
// CONFIGURATION

#include <stdbool.h>

// Tolerance for changes in acceleration (in gees) before updating the orientation.
static const double tol_accel_0 = 0.2;
static const double tol_accel_1 = 0.2;
```

```c
static const double tol_accel_2 = 0.2;

bool is_within_range(double value, double tolerance, double offset) {
        //printf("%6f, %6f, %6f\n", value, tolerance, offset);

        if ((offset - tolerance) <= value && (value < (offset + tolerance))) {
                //printf("1 true: %6f, %6f, %6f\n", offset - tolerance, value, offset + tolerance);
                return true;
        } else {
                //printf("1 false: %6f, %6f, %6f\n", offset - tolerance, value, offset + tolerance);
                return false;
        }
}

// GENERIC FUNCTIONS

//callback that will run if the Spatial is attached to the computer
int CCONV AttachHandler(CPhidgetHandle spatial, void *userptr) {
        int serialNo;
        CPhidget_getSerialNumber(spatial, &serialNo);
        printf("Spatial %10d attached!", serialNo);

        return 0;
}

//callback that will run if the Spatial is detached from the computer
int CCONV DetachHandler(CPhidgetHandle spatial, void *userptr) {
        int serialNo;
        CPhidget_getSerialNumber(spatial, &serialNo);
        printf("Spatial %10d detached! \n", serialNo);

        return 0;
}

//callback that will run if the Spatial generates an error
int CCONV ErrorHandler(CPhidgetHandle spatial, void *userptr, int ErrorCode, const char *unknown) {
        printf("Error handled. %d - %s \n", ErrorCode, unknown);
        return 0;
}

//Display the properties of the attached phidget to the screen.
//We will be displaying the name, serial number, version of the attached
//device, the number of accelerometer, gyro, and compass Axes, and the current
//data rate of the attached Spatial.
int display_properties(CPhidgetHandle phid) {
        int serialNo, version;
        const char* ptr;
        int numAccelAxes, numGyroAxes, numCompassAxes, dataRateMax, dataRateMin;

        CPhidget_getDeviceType(phid, &ptr);
        CPhidget_getSerialNumber(phid, &serialNo);
        CPhidget_getDeviceVersion(phid, &version);
        CPhidgetSpatial_getAccelerationAxisCount((CPhidgetSpatialHandle)phid, &numAccelAxes);
        CPhidgetSpatial_getGyroAxisCount((CPhidgetSpatialHandle)phid, &numGyroAxes);
        CPhidgetSpatial_getCompassAxisCount((CPhidgetSpatialHandle)phid, &numCompassAxes);
        CPhidgetSpatial_getDataRateMax((CPhidgetSpatialHandle)phid, &dataRateMax);
```

```
        CPhidgetSpatial_getDataRateMin((CPhidgetSpatialHandle)phid, &dataRateMin);

        printf("%s\n", ptr);
        printf("Serial Number: %10d\nVersion: %8d\n", serialNo, version);
        printf("Number of Accel Axes: %i\n", numAccelAxes);
        printf("Number of Gyro Axes: %i\n", numGyroAxes);
        printf("Number of Compass Axes: %i\n", numCompassAxes);
        printf("datarate> Max: %d  Min: %d\n", dataRateMax, dataRateMin);

        return 0;
}
```

Listing 3: Code for Path Detection

```
#include <cvaux.h>
#include <highgui.h>
#include <cxcore.h>
#include <stdio.h>

//to compile:
//gcc `pkg-config opencv --cflags --libs` -o <exec_name> <file.c>

int main (int argc, char* argv[])
{
        //Default capture size - 640 x 480
        CvSize size = cvSize(640,480);

        //Open capture device. 0 is /dev/video0, 1 is /dev/video1 etc...
        CvCapture* capture = cvCaptureFromCAM( 1 );
        if( !capture )
        {
                fprintf(stderr, "ERROR: capture is NULL\n");
                getchar();
                return -1;
        }

        //Create a window in which the captured images will be presented
        cvNamedWindow( "Camera", CV_WINDOW_AUTOSIZE );

        while(1)
        {
                //Get one frame
                IplImage* frame = cvQueryFrame (capture);

                if( !frame )
                {
                        fprintf(stderr, "ERROR: frame is null... \n");
                        getchar();
                        break;
                }

                cvShowImage( "Camera", frame );

                if( (cvWaitKey(10) & 255) == 27 ) break;
        }
        cvReleaseCapture( &capture );
        cvDestroyWindow( "mywindow" );
        return 0;
}
```

Listing 4: Processing Image Frames in Path Detection Module

```
//Referenced: http://robomcgill.git.sourceforge.net/git/gitweb.cgi?p=robomcgill/robomcgill
    ;a=blob_plain;f=vision/vision.cc;hb=ff2b309901eefa7ad921df4a98cd99d71f9becd5

#include <cvaux.h>
#include <highgui.h>
#include <cxcore.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include <math.h>
#include <float.h>
//#include <limits>
#include <time.h>
#include <ctype.h>

//to compile:
//gcc `pkg-config opencv --cflags --libs` -o <exec_name> <file.c>

typedef struct retValue {
        double x;
        double y;
} point;

typedef struct elipse {
        float angle;
        int total;
        CvPoint center;
        CvSize size;
} elipse;

int main (int argc, char* argv[])
{
        //Default capture size - 640 x 480
        CvSize size = cvSize(640,480);

        //Open capture device. 0 is /dev/video0, 1 is /dev/video1 etc...
        CvCapture* capture = cvCaptureFromCAM( 1 );
        if( !capture )
        {
                fprintf(stderr, "ERROR: capture is NULL\n");
                getchar();
                return -1;
        }

        //Create a window in which the captured images will be presented
        cvNamedWindow( "Camera", CV_WINDOW_AUTOSIZE );
        cvNamedWindow( "HSV", CV_WINDOW_AUTOSIZE );
        cvNamedWindow( "PathDetection", CV_WINDOW_AUTOSIZE );

        //Detect the color red
        //CvScalar hsv_min = cvScalar(150, 84, 130, 0);
        //CvScalar hsv_max = cvScalar(358, 256, 255, 0);
        //hsv values to detect red in hot tub - path detect does not work
        CvScalar hsv_min = cvScalar(28, 82, 185, 0);
        CvScalar hsv_max = cvScalar(41, 174, 256, 0);

        IplImage* hsv_frame = cvCreateImage(size, IPL_DEPTH_8U, 3);
        IplImage* thresholded = cvCreateImage(size, IPL_DEPTH_8U, 1);
```

```
    CvMemStorage* storage;

    //initialize font and add text
CvFont font;
cvInitFont(&font, CV_FONT_HERSHEY_SIMPLEX, 1.0, 1.0, 0, 1, CV_AA);
char *info;

    while(1)
    {
            //Get one frame
            IplImage* frame = cvQueryFrame (capture);

            if( !frame )
            {
                    fprintf(stderr, "ERROR: frame is null... \n");
                    getchar();
                    break;
            }

            //Convert color space to HSV as it is much easier to filter colors in the
               HSV color-space
            cvCvtColor(frame, hsv_frame, CV_BGR2HSV);
            //Filter out which colors are out of range
            cvInRangeS(hsv_frame, hsv_min, hsv_max, thresholded);

            //Play with threshold values
            int hlowerpipe = 127, hupperpipe = 45, sthresholdpipe = 86, erodepipe = 3,
                dilatepipe = 6; //for elipse
            int i, j, k; //for iterations

            int heighthsv, widthhsv, stephsv, channelshsv;
            int heightmono, widthmono, stepmono, channelsmono;
            uchar *datahsv, *datamono;

            heighthsv = hsv_frame->height;
            widthhsv = hsv_frame->width;
            stephsv = hsv_frame->widthStep;
            channelshsv = hsv_frame->nChannels;
            datahsv = (uchar *) hsv_frame->imageData;

            //change mono naming convention
            heightmono = thresholded->height;
            widthmono = thresholded->width;
            stepmono = thresholded->widthStep;
            channelsmono = thresholded->nChannels;
            datamono = (uchar *) thresholded->imageData;

            for (i = 0; i < (heighthsv); i++) {
            for (j = 0; j < (widthhsv); j++) {
                    if ((datahsv[(i) * stephsv + j * channelshsv] <= hlowerpipe)&& (
                       datahsv[(i) * stephsv + j * channelshsv] >= hupperpipe)) {
                                    if ((datahsv[(i) * stephsv + j * (channelshsv) +
                                       1])> sthresholdpipe) {
                                            datamono[i * stepmono + j * channelsmono]
                                               = 255;
                                    } else
                                      /*A very simple concept with the loops here if
                                         the hue values are in the aforementioned
                                         range and the
```

```
                                    threshold is met then logic one else logic
                                        zero*/

                            datamono[i * stepmono + j * channelsmono] = 0;
        }
}
}

//crosscheck all are either black or white
for (i = 0; i < (heighthsv); i++) {
for (j = 0; j < (widthhsv); j++) {
        if (!(datamono[i * stepmono + j * channelsmono] == 0 || datamono[i
            * stepmono + j * channelsmono] == 255))
                            datamono[i * stepmono + j * channelsmono] = 0;
}
}

//The path is detected as noise, be careful with filtering
// get rid of noise
cvErode(thresholded, thresholded, 0, erodepipe);
cvDilate(thresholded, thresholded, 0, dilatepipe);

CvSeq* contour;

// Create dynamic structure and sequence.
storage = cvCreateMemStorage(0);
contour = cvCreateSeq(CV_SEQ_ELTYPE_POINT, sizeof(CvSeq), sizeof(CvPoint),
        storage);

// Find all contours.
cvFindContours(thresholded, storage, &contour, sizeof(CvContour),
        CV_RETR_LIST, CV_CHAIN_APPROX_NONE, cvPoint(0, 0));

elipse retElipse;
retElipse.total = 0;
retElipse.angle = -1;

// This cycle draw all contours and approximate it by ellipses.
for (; contour; contour = contour->h_next) {

int count = contour->total; // This is number point in contour

CvBox2D box;

// Number point must be more than or equal to 6 (for cvFitEllipse_32f).
if (count < 6)
        continue;

        CvMat* points_f = cvCreateMat(1, count, CV_32FC2 );
        CvMat points_i = cvMat(1, count, CV_32SC2, points_f->data.ptr);
        cvCvtSeqToArray(contour, points_f->data.ptr, CV_WHOLE_SEQ );
        cvConvert( &points_i, points_f );

        // Fits ellipse to current contour.
        box = cvFitEllipse2(points_f);



        // Convert ellipse data from float to integer representation.
```

```
            if (count > retElipse.total) {

                    retElipse.total = count;
                    retElipse.angle = box.angle;
                    retElipse.center = cvPointFrom32f(box.center);
                    retElipse.size.width = cvRound(box.size.width * 0.5);
                    retElipse.size.height = cvRound(box.size.height * 0.5);
            }

            cvReleaseMat(&points_f);

            }

            if (retElipse.angle != -1) {
            cvLine(thresholded, cvPoint(thresholded->width / 2, thresholded->height
                        / 2), cvPoint((thresholded->width / 2) + (100 * (sin(
                                                  (-retElipse.angle) * 3.14159265 / 180))),
                                                     (thresholded->height
                                                                  / 2) + (100 * (cos(-
                                                                     retElipse.angle *
                                                                     3.14159265 / 180)))),
                                                                         CV_RGB(0,0,255),
                                                                            5);

            cvCircle(thresholded, cvPoint(thresholded->width / 2, thresholded->height
                              / 2), 10, CV_RGB(0,0,255), 10);

            // Draw ellipse.
            cvEllipse(thresholded, retElipse.center, retElipse.size,
                    -retElipse.angle, 0, 360, CV_RGB(255,0,255), 1, CV_AA, 0);

            cvCircle(thresholded, retElipse.center, 20, CV_RGB(0,255,0), 3);

            }



            asprintf(&info, "Angle = %f", 270-retElipse.angle);
            cvPutText(thresholded, info , cvPoint(10, 130), &font, cvScalar(255, 255,
               255, 0));

            cvShowImage( "Camera", frame );
            cvShowImage( "HSV", hsv_frame );
            cvShowImage( "PathDetection", thresholded );


            if( (cvWaitKey(10) & 255) == 27 ) break;
        }
        cvReleaseMemStorage(&storage);
        cvReleaseCapture( &capture );
        cvDestroyWindow( "mywindow" );
        return 0;
}
```