

# AUVSI RoboSub Team 24

Senior Design Final Report – April 2012

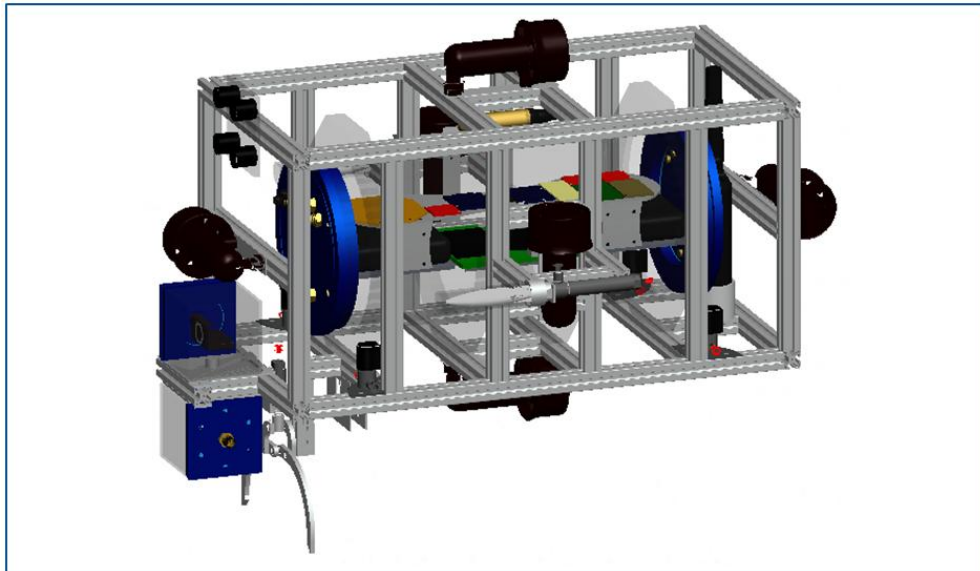
*By*

Eric Sloan, Kashief Moody, Tra Hunter

*Department of Mechanical Engineering, FAMU-FSU College of Engineering*

Antony Jepson, Hang Zhang, Ryan Kopinsky

*Department of Electrical and Computer Engineering, FAMU-FSU College of Engineering*



*2011-2012 FAMU- FSU College of Engineering Senior Design Team 24 Project Sponsors*



Department of Mechanical Engineering

FAMU-FSU College of Engineering

2525 Pottsdamer St, Tallahassee, FL 32310

## Abstract

Team Robosub—a multidisciplinary team of mechanical, electrical, and computer engineers—has designed and built an autonomous underwater vehicle (AUV) to compete in the 15<sup>th</sup> Annual Association for Unmanned Vehicle Systems International (AUVSI) *Ides of Transdec* Robosub Competition in San Diego, CA in July 2012. The vehicle will represent the FAMU-FSU College of Engineering as a first-year entry against multiple established, well-funded, veteran teams such as perennial powerhouses ETS Team SONIA (Canada) and Cornell University's CUAUV team. The vehicle is mechanically and electrically complete and fulfills the requirements set forth by the competition rules committee.

The competition is to be held in a 16ft-deep section of the SSC Pacific Transdec Facility—an anechoic saltwater pool off the Pacific coast, which will contain a series of obstacles that the AUV must autonomously attempt to complete in a 15 minute time frame. These tasks include passing through a gate, dropping markers in designated bins, firing torpedoes through designated PVC cut-outs, triangulating and navigating to the location of an acoustic pinger, recovering a PVC wreath, surfacing within a specified octagonal region, and replacing the wreath to either its original location or to a secondary location guarded by a second pinger. Furthermore, the vehicle must navigate to each mission task by following a segmented PVC path—all while maintaining proper stability and control.

In order to complete these tasks, the AUV is equipped with several peripheral mechanical subsystems including thrusters for propulsion, a marker dropper, a compressed air distribution system, pneumatically-actuated torpedo launchers, and a pneumatically-actuated grasp/release mechanism (claw). The AUV further features several sensory devices including a sealed gauge submersible pressure transducer to monitor the vehicle's depth, web cameras (computer vision), an array of passive hydrophones, and an inertial measurement unit (IMU) for navigation and stability control. In addition, the vehicle contains three watertight acrylic enclosures—the main hull which contains the PC, microcontrollers, motor drivers, voltage regulator boards, and interface circuits, and the two camera enclosures which each house either the front-facing or down-facing camera. Communication between the peripheral components (arms, legs, eyes, and ears) and the electronics (brain) inside the main hull is achieved via the use of high-quality, reliable underwater wet-mate micro connectors. Lastly, the vehicle is equipped with a remote toggle kill switch which features an elegant watertight rubber seal; actuation of the kill switch not only shuts on/off all the electrical devices inside the AUV, but also initiates the mission program.

The AUV was designed almost completely from scratch—only inheriting four thrusters, two hydrophones, a partially-completed marker dropper mechanism, and some pieces of 80/20 *Inch Solid* extruded aluminum framing from the previous year's team. Provided an initial budget of \$7,433, followed by \$4,000 in further sponsorships from the ECE Department at the FAMU-FSU College of Engineering and the Naval Engineering Education Center (NEEC), the vehicle was carefully designed with themes of efficiency, cost-effectiveness, symmetry, density, weight, balance, ergonomics, and aesthetics in mind. It was desired to have a modular design that was both naturally stable and balanced via

symmetry and mass properties, but also agile and versatile. Furthermore, the vehicle needed to be less than 110 lbs to maintain eligibility (although a lighter vehicle earns more points and was thus desired), and be slightly positively buoyant in salt water so that it would naturally surface upon actuation of the kill switch to shut down the system. Aesthetics and elegance were also considered in order to yield an appealing foundation for sponsorship logos, a clean and organized design, and also to earn points from the competition judges since the completion of tasks in the underwater obstacle course is not the sole source of points.

Engineering design tools such as Pro/Engineer, ADAMS, P-Spice, DipTrace PCB Design, were all used in the formulation of the AUV design to develop 3D models, monitor system mass properties, generate mechanical and electrical simulations, etc. The following report thoroughly, yet concisely, documents the developmental process of the AUV from concept generation and selection to manufacturing and assembly of all the components of the design, as well as their integration into a single coherent system.

The mechanically completed AUV weighs  $\approx 87.7$  lbs, has a density of  $\approx 0.036$  lb/in<sup>3</sup>, maintains stable natural roll and pitch angles of zero degrees, and is capable of accurately and powerfully launching torpedoes, dropping markers upon command, grasping/releasing upon command, and detecting the gate and the PVC path segments. Integration of the pressure transducer and IMU into the guidance system is the next goal for the senior design team, enabling the control system for navigation and stability to be derived, and the competition task algorithms to be debugged in time for the competition. Overall, the outlook is optimistic, and the design team is committed to finishing the project strong and representing the FAMU-FSU College of Engineering well in the competition as first-time competitors.

## Contents

Abstract .....	2
Introduction .....	7
Problem Description .....	7
<i>Goal Statement</i> .....	7
<i>Objectives</i> .....	7
<i>Operating Environment</i> .....	8
<i>Constraints</i> .....	9
Functional Diagram .....	9
House of Quality .....	10
Project Plan .....	10
Concept Generation .....	11
Hull / Frame .....	11
Interior Hull Design .....	13
Electronics .....	13
<i>Main Controller Unit</i> .....	13
<i>Power Supply</i> .....	16
<i>Computer Vision</i> .....	16
<i>Guidance system</i> .....	18
<i>Inertial measurement unit (IMU)</i> .....	18
<i>Pressure Transducer</i> .....	18
<i>Hydrophones</i> .....	19
Vehicle Propulsion System .....	20
Mechanical Subsystems .....	20
<i>Grasp/Release Mechanism</i> .....	20
<i>Marker Dropper</i> .....	22
<i>Torpedo Launchers</i> .....	24
Final Concept .....	26
Hull / Frame .....	28
Interior Hull Design .....	31
Camera Enclosures .....	33
Vehicle Propulsion .....	34
Mechanical Subsystems .....	36
<i>Compressed Air Distribution System</i> .....	36
<i>Grasp / Release Mechanism</i> .....	39
<i>Marker Dropper</i> .....	41
<i>Torpedo Launchers</i> .....	42
Electrical System .....	43
<i>Voltage Regulator Boards</i> .....	45
Software System .....	48
<i>Operating System</i> .....	48
<i>Software Interface and Design</i> .....	48
<i>Mission Control</i> .....	50
Guidance System .....	53
Computer Vision .....	54
Inertial Measurement Unit .....	55
Hydrophone Interface Circuit (Proposed Alternate Design) .....	55
Engineering Economics .....	59
Results and Discussion .....	61

Environment, Health, and Safety .....	73
Conclusions.....	73
Acknowledgements.....	73
Appendix A – Data Sheets.....	75
Appendix B – IMU Code.....	77
Appendix C – Preliminary Thruster Control Programs .....	82
Engineering Drawings .....	96
References.....	142
Biographical Sketches.....	143

Figure 1: SSC Pacific Transdec Facility in San Diego, CA .....	8
Figure 2: IMCL Submersible Pressure Transducer .....	19
Figure 3: Pro/Engineer Model of the Final AUV Design - Angled View.....	26
Figure 4: Pro/Engineer Model of the Final AUV Design - Side View .....	26
Figure 5: Pro/Engineer Model of the Final AUV Design - Front .....	27
Figure 6: Pro/Engineer Model of the Final AUV Design - Top View .....	27
Figure 7: Photograph of the Assembled AUV - Angled View .....	28
Figure 8: Close-up view of the Hull / Frame .....	29
Figure 9: Close-up Exploded View of the Hull End Caps .....	30
Figure 10: Electronics Platform (IMU—Center—Not Shown).....	31
Figure 11: Close-Up View of the Interior Hull Layout .....	32
Figure 12: Close-Up View of the Camera Enclosures.....	34
Figure 13: Wiring Schematic for Thrusters .....	35
Figure 14: Close-Up View of the Compressed Air Tank and Regulators .....	37
Figure 15: Compressed Air Distribution System Overview .....	38
Figure 16: Close-Up View of Grasp / Release Mechanism.....	39
Figure 17: Dynamic Simulation of Grasp/Release Mechanism Using Pro/Engineer - Mechanism.....	39
Figure 18: Right Jaw Angle of Rotation – ADAMS Simulation.....	40
Figure 19: Left Jaw Angle of Rotation – ADAMS Simulation.....	40
Figure 20: Close-Up View of Marker Dropper.....	41
Figure 21: Close-Up View of Torpedo Launcher .....	43
Figure 22: Lithium-Ion Battery Pack Used on AUV .....	43
Figure 23: Electrical Circuit Diagram Overview for the AUV.....	45
Figure 24: Circuit diagram of LM22677 Voltage Regulator for the Zotac PC and IMCL Submersible Pressure Transducer .	46
Figure 25: Startup Simulation of LM22677 Voltage Regulator for Zotac PC and IMCL Submersible Pressure Transducer .	46
Figure 26: Circuit Diagram of LM22676 Voltage Regulator for the Arduino Uno Boards and Hydrophone Interface Circuit .....	47
Figure 27: Startup Simulation of LM22676 Voltage Regulator for the Arduino Uno Boards .....	47
Figure 28: Photos of the LM22677 VRB (Left) and LM22676 VRB (Right) .....	48
Figure 29: Graphical illustration of the Software Communication Interface and Structure .....	49
Figure 30: Mission control program fork illustration.....	49
Figure 31: Graphical Illustration of the Gate Passing Algorithm.....	52
Figure 32: Gate Passing Flow Chart.....	52
Figure 33: Gladiator Ring Task Flow Chart .....	53
Figure 34: Block Diagram of the Guidance System .....	54

Figure 35: Phidget-Spatial 3/3/3 Inertial Measurement Unit .....	55
Figure 36: Proposed Hydrophone Interface Circuit Design Schematic.....	58
Table 1: Hull/Frame Concept Descriptions .....	11
Table 2: Hull/Frame Decision Matrix .....	12
Table 3: Grasp/Release Mechanism Decision Matrix .....	21
Table 4: Marker Dropper Decision Matrix .....	23
Table 5: Torpedo Launcher Decision Matrix .....	25
Table 6: Current Consumption of Electrical Devices.....	44
Table 7: Voltage Regulator Board Specifications .....	45
Table 8: Component Selection for Proposed Alternative Hydrophone Interface Circuit Design .....	57
Table 9: Team Robosub 2011-2012 Project Budget .....	59
Table 10: Sponsorship Funds .....	60

## Introduction

The FAMU-FSU College of Engineering 2012 Senior Design Team 24 (ME) / Team 4 (ECE) is a multidisciplinary group comprised of three electrical/computer engineers and three mechanical engineers who were asked to design and build an autonomous underwater vehicle (AUV) as a competition-based project. The design team was initially allocated \$7,433 of available funds, and only inherited a shallow selection of potential equipment. Each team member was assigned a vital role in the project, including the following administrative/leadership duties:

- Eric Sloan – Mechanical Engineering Project Manager
- Antony Jepson – Electrical/Computer Engineering Project Manager
- Tra Hunter – Lead Treasurer
- Ryan Kopinsky – Lead Secretary
- Hang Zhang – Assistant Treasurer
- Kashief Moody – Assistant Secretary

## Problem Description

### *Goal Statement*

The goals of the Association for Unmanned Vehicle Systems International (AUVSI) student competitions are to provide opportunities for students to experience the challenges of system engineering, to develop skill in accomplishing realistic missions with autonomous vehicles and to foster relationships between young engineers and the organizations developing and producing autonomous vehicle technologies.

### *Objectives*

The AUVSI final mission statement asks competing teams to design and build an autonomous underwater vehicle (AUV) capable of completing the following underwater tasks in the 15<sup>th</sup> Annual *Ides of Transdec* AUVSI Robosub Competition:

**[1] Gate** Pass through the center of a PVC gate

**[2] Path** Navigate from task to task by following a segmented orange PVC path

**[3] Training** Strike two of three buoys in a designated order prior to advancing to the next task—each buoy will be a different color (i.e. green, red, yellow)

**[4] Obstacle Course** Traverse through a PVC window at a particular height (i.e. the top of the “box” is imaginary in that it is not designated by a physical PVC structure)

**[5] Gladiator Ring** Drop two markers into two of four explicit bins (black with white borders), designated by a red silhouette of either a sword, shield, trident, or net.

**[6] Et Tu Brute (Kill Caesar)** Independently launch a torpedo through two of four explicit PVC cutouts designated by size (i.e. 7" diameter circle or 14" diameter circle) and color (i.e. red or blue)

**[7] Laurel Wreath and Emperor's Palace** Triangulate and navigate to the location of an acoustic pinger (operating at a designated frequency between 22 kHz and 30 kHz), secure a PVC laurel wreath object, surface within a designated octagonal region, and submerge to return and release the object either to its original location, or to a secondary location designated by an alternate pinger

### *Operating Environment*

The AUV will be operating at the SSC SD TRANSDEC Facility, which houses an anechoic saltwater pool. The facility is located in San Diego, CA, and the water is obtained directly from the Pacific Ocean. The underwater obstacle course will be arranged in a region with a maximum depth of 16 ft. The temperature of the water is expected to be between 70 – 75 °F, with calm winds/currents. The practice facility that will be utilized to test the components of the AUV, as well as the end product, will be the nearby FSU Morcom Aquatics Center, which has granted the design team permission to set up replicated obstacle course environments and test the AUV at depths of up to 17 ft.



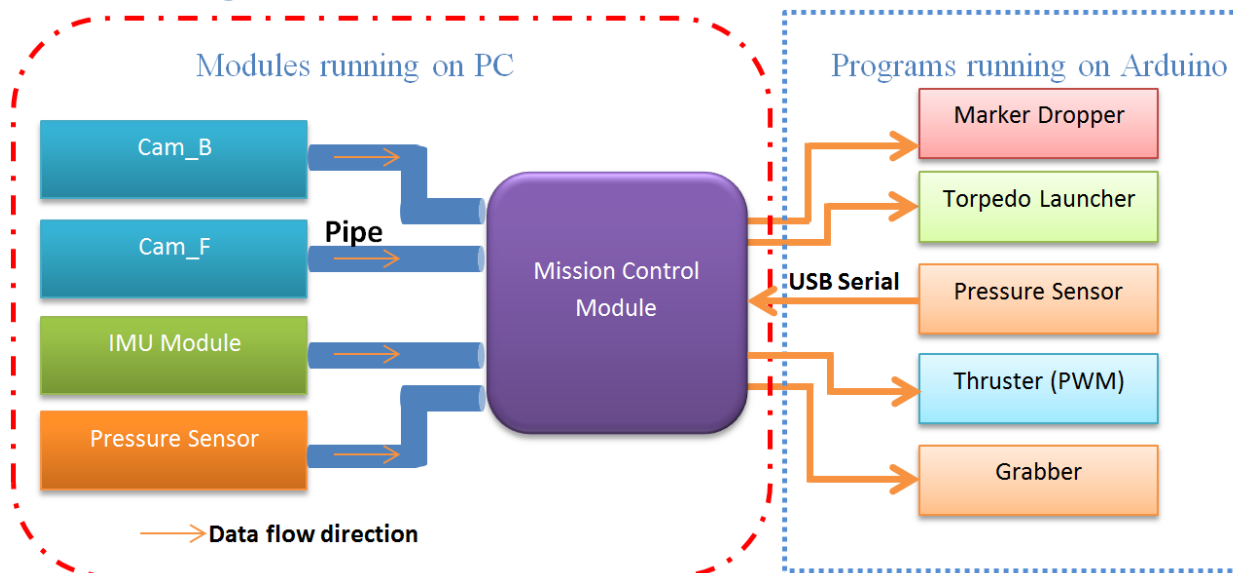
Figure 1: SSC Pacific Transdec Facility in San Diego, CA



## Constraints

- 1) The AUV must be less than 6 ft x 3 ft x 3 ft in size
- 2) The vehicle must be less than 110 pounds (under 84 pounds to avoid a point penalty)
- 3) The device has 15 minutes to complete as many tasks as it can
- 4) Total current budget of \$11,433 must not be exceeded
- 5) The markers and torpedoes on the AUV must not exceed 2.0 in x 2.0 in x 6.0 in and must weigh no more than 2.0 lbs in air
- 6) The torpedoes must travel at a “safe” speed, so as to not induce a bruise upon accidental impact with one of the divers.
- 7) The battery must not have an open source voltage exceeding 60 VDC
- 8) The vehicle must be buoyant by at least one half of one percent (0.5 %) of their mass when they have been shut off via the kill switch

## Functional Diagram



		AUV Functional Capabilities															
		Density	Weight	Project cost	BTD150 Thrusters (6)	Sensortech SQ26 Hydrophones (4)	Phidget Spatial 3/3/3 IMU	Logitech C615 web Camera (x2)	Lithium-ion Battery (x2)	Compressed Air Distribution System	Programming	IMCL Submersible Pressure Transducer	Grasp/Release Mechanism	Torpedo Launcher	Marker Dropper	Safety	
<b>Competition Requirements</b>	Weight < 110 lbs	x	x														
	Size ≤ 6ft x 3ft x 3ft	x	x														
	<i>Et Tu Brute (Kill Caesar)</i>							x		x	x			x		x	
	<i>Gladiator Bins</i>							x			x				x		
	<i>Path (Navigation/ Stability Control)</i>					x	x	x			x	x					
	<i>Laurel Wreath (Recover Wreath)</i>					x		x		x	x		x			x	
	<i>Emperor's Palace (Surface Inside Octagon)</i>				x		x				x	x					
	Cost ≤ \$7,433			x													
	Operate for 15 Minutes				x			x	x		x						
	Buoyancy ≤ 0.5% Mass	x	x														
	Open Circuit Voltage ≤ 60 VDC								x								x
	Units	lbs/in <sup>3</sup>	lbs	USD (\$)													
Value	0.036	87.7	8,408														

### Project Plan

Our approach to this complex task was to initially spend a significant amount of time researching successful themes evident among the veteran competing teams, gain insight from the provided AUVSI competition primer document, and get off to a fast start in the development of the AUV design due to the anticipated magnitude of the project. The majority of the Fall 2011 semester was dedicated to concept generation and design of the AUV and its subsystems, however, a significant amount of parts were ordered and manufactured in November and December as well. Spring

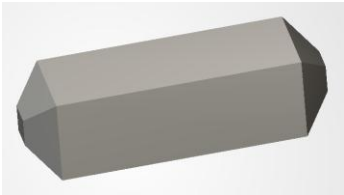
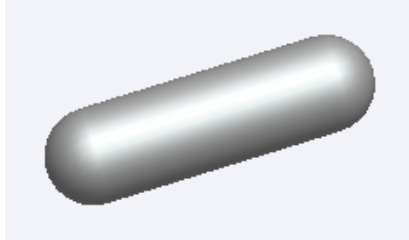
semester was mainly devoted to the continued ordering and manufacturing of components and subsystems of the AUV. However, the design phase still continued as adaptations to the design, as well as the detailed design of some of the remaining mechanical subsystems were completed as well. Obstacles had to be overcome, and bold decisions had to be made in order to yield a quality product. It was considered critical to ensure that the mechanical features of the design would also accommodate the needs and requests of the electrical and computer engineers. Furthermore, it was important to not only maintain impeccable organization and consistently demanding work pace, but also to prioritize aspects of the design that were coupled to the progress of other features of the design—mechanical, electrical, or computer-related. A link to the two-semester Gantt Chart is provided below for convenience:

[http://www.eng.fsu.edu/~jepsoan/robosub\\_gantt\\_chart.html](http://www.eng.fsu.edu/~jepsoan/robosub_gantt_chart.html)

## Concept Generation

### Hull / Frame

Table 1: Hull/Frame Concept Descriptions

	Shape		Material	
	Frame	Hull	Frame	Hull
Concept 1	None		None	Aluminum 6061/Acrylic
Concept 2	None		None	Acrylic
Concept 3	None		None	Aluminum

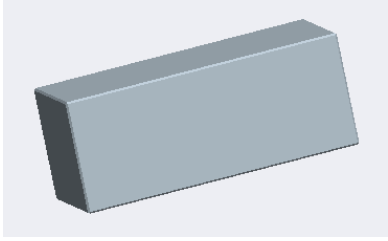
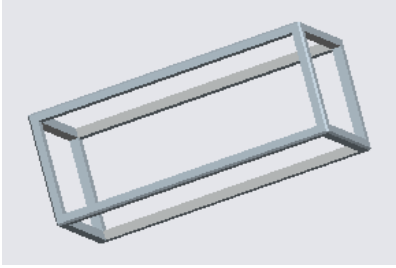
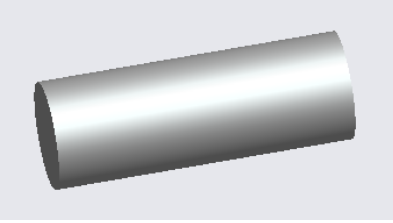

				6061/Acrylic
Concept 4			80/20 Extruded Aluminum	Acrylic
Concept 5	None		None	Carbon Fiber

Table 2: Hull/Frame Decision Matrix

Design Criteria	Ease of Manufacture		Waterproof Reliability		Hydrodynamics (Drag Force)		Weight		Versatility/Ease of Peripheral Subsystem Attachment		Cost		Sum
Weighting Factor	0.25		0.20		0.10		0.15		0.20		0.10		1.00
Concept 1	6	1.50	5	1.00	8	0.80	5	0.75	7	1.40	7	0.70	6.15
Concept 2	3	0.75	7	1.40	9	0.90	7	1.05	2	0.40	4	0.40	4.90
Concept 3	8	2.00	6	1.20	3	0.30	6	0.90	10	2.00	8	0.80	7.20
Concept 4	7	1.75	8	1.60	6	0.60	7	1.05	9	1.80	6	0.60	7.40
Concept 5	2	0.50	6	1.20	10	1.00	9	1.35	1	0.20	4	0.30	4.65

In reference to the above decision matrix, ease of manufacture, waterproof reliability, and versatility/ease of peripheral subsystem attachment were deemed the most critical determining factors in the selection process of the hull/frame. Furthermore, considerations in regards to the visibility of the electronics from an external perspective, natural stability of the structure, and ease of implementing access to the interior of the hull were also given. Thus, following analysis, it was decided that concept 4 provided the best design with which to move forward in the project.

## **Interior Hull Design**

Various concepts were considered for the layout of the interior of the pressure vessel. Driven by the need to securely support the electronics (i.e. lithium-ion battery packs, microcontrollers, interface circuits, voltage regulator boards, motor drivers, and inertial measurement unit), as well as to efficiently and effectively dissipate heat away from the electronics, it was determined that a thermally conductive, flat platform would need to rest inside the hull. Since the hull was decided to be made out of a thermal insulator in acrylic, the goal was set to direct the heat toward the aluminum end caps of the pressure vessel via this thermally conductive plate. However, in order to transfer heat effectively to the end caps—the only viable outlet to the cooler surrounding environment—the thermally conductive plate would need to share surface area with the end caps. It was determined that the most logical way to achieve this would be to add slots to the custom-designed end caps, thus providing surface area for the ends of the plate to rest and continue to efficiently transfer heat via conduction. In order to provide support for this flat plate, it was determined that acrylic would be the best candidate because due to its thermal insulation properties, it could also serve to prevent heat dissipation into the internal surroundings of the pressure vessel, and thus increase the efficiency of the conductive network. Due to cost, thermal conductivity, and weight considerations, it was determined that Aluminum Alloy 6061 would be the best material for the job.

## **Electronics**

### ***Main Controller Unit***

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 central “brain” of the AUV, the main control unit must be composed of a powerful microprocessor. Since image processing and sensor controls involve numerous amounts of data processing and storage, a large onboard memory for fast transferring of data becomes an important factor in choosing the main control unit for the AUV. Other important factors in selecting the most appropriate main controller unit included the cost, I/O functionality and compatibility, power consumption and dissipation, and the compatibility of the software system with sensors. Here are three concepts or options that were explored and discussed:

#### **Option 1: Use the existing BeagleBoard**

The first option was to use a BeagleBoard (REV B7) that was inherited from last year's project (donated by ARM). The BeagleBoard features a 600MHz ARM OMAP3530 processor and a separate 430MHz DSP processor. The beagle board also has 128MB DDR RAM.

**Benefits:**

- Economic: no cost (donated)
- Readiness: Software system is already setup and ready to run programs
- Peripherals availability and accessibility: most accessories are and we can get started programing on this board immediately
- Low power consumption: The total power required by the system is less than 5W

**Drawbacks:**

- Performance: slow clock rate of microprocessor and Low MIPS performance (< 1400MIPS)
- Memory: small onboard memory and slow memory speed. The board only has 128MB RAM. With Angstrom Linux running, only 80MB memory is left. The memory bus runs at 166MHz
- I/O constraints: No USB connector onboard due the PCB defect in this version of BeagleBoard. A microUSB connector onboard requires a miniA type connector to function as a USB host
- Defect: Power jack is in a poor condition. It is very easy to lose external power. This problem could be fixed but might require some effort.

**Option 2: Purchase an Intel x86-based PC**

The second option is to purchase an Intel x86 based PC. An Intel x86 based microprocessor could deliver us abundant computing power for all the functionalities that the AUV needs to perform. The components we are looking at are Intel Core i3 2.6GHz LGA 1155 35W dual core processor, 3GB of DDR3 RAM, and a 32GB Solid State Drive for storage.

**Benefits:**

- Performance: Very high performance compared to the BeagleBoard we currently have. Multicore and the Hyperthread technology enable multithread parallel computing, which further improve the system performance
- Software system: With this Intel PC onboard, the AUV could use full featured operating system such as Ubuntu, Debian or Windows OS instead of embedded operating systems. The full support for compilers such as GCC and G++ and the familiar programming environment could make the programming process more efficient

- Computer vision support: For computer vision and image processing, OpenCV will be used since it provides lots of libraries and functions which could make the most challenging problem easier. Since the OpenCV optimized for the Intel architecture, using an Intel platform will boost the performance of the imaging processing.

Drawbacks:

- Power consumption: The CPU itself has a TDP (Thermal Design Power) of 35W. The total power required by the computer system was estimated to be about 104W. The high power consumption requires the AUV to efficiently transfer heat from the enclosed pressure chamber to the surroundings, so an additional cooling device or heat sink might be required.
- Cost: The total cost of the system was estimated to be more than \$300.
- Communication overhead: Since the project is sponsored by ARM, the BeagleBoard was required to be used. Therefore communication between PC and the BeagleBoard became an overhead. If a PC were to be used, the BeagleBoard would not necessary for other functions due to the fact that a single PC would powerful enough to handle all the tasks.

### **Option 3: Purchase the latest BeagleBoard model (i.e. BeagleBoard-xM)**

The third option was to update the current BeagleBoard to a newer version. The original BeagleBoard was released in 2008. And the newest version BeagleBoard-xM was released in 2010. The new version features a faster Texas Instruments Cortex A8 1GHz processor, separate 800MHz DSP, and a 512MB DDR memory.

Benefits:

- Performance: Compared to the original BeagleBoard, the BeagleBoard-xM contains a much more powerful Cortex A8 1GHz processor which can achieve 2000 Dhrystone<sup>1</sup> MIPS and is about 1.5 times faster than original BeagleBoard. 800MHz DSP and the 512 MB RAM will significantly boost the performance of image processing compared to the original BeagleBoard.
- Low cost: This new version of BeagleBoard costs about \$150, which is about half the price of the Intel PC.
- I/O Ports: The BeagleBoard-xM has onboard USB hub which provides 4 connectors. Therefore the need for additional USB hub is eliminated. The USB ports provide full USB host and OTG supports with a data transfer rate of 480Mbps. A DB9 connector is provided for the RS232 port. And a full functional Ethernet port is provided.
- Backward compatibility: Most cables, peripherals, accessories and software used for the original BeagleBoard are compatible with the newer version.

---

<sup>1</sup> A synthetic computing benchmark program developed in 1984 by Reinhold P. Weicker intended to be representative of system (integer) programming

- Resources: The BeagleBoard-xM has a very broad online community. A lot of open source projects that were done on the BeagleBoard-xM could help us to design our programs faster and more efficient.
- Low power consumption: The total power required by the system is less than 5W

Drawbacks:

- Processing Power: There is a potential that the BeagleBoard-xM will be unable to handle the relatively large processing demand of the computer vision algorithms.

After in-depth research and discussion on all three possible solutions according to their benefits and drawbacks, our team decided to use the Beagleboard-xM system, which theoretically had high performance, low power consumption, low cost and abundant available resources. The old BeagleBoard was to serve as a backup unit. However, upon testing, it was discovered that the processing power of the BeagleBoard-xM system was insufficient to process or handle the image processing requirements of even a simple computer vision program which receives input information from the web cameras. As a result, an alteration was made to the design in order to accommodate a Zotac mini-itx PC instead. This PC was tested and verified to be able to successfully handle the computer vision algorithms to our specifications.

*Power Supply*

Several options were available to use as power supplies for the components of the AUV. Lead-acid batteries, Nickel Metal Hydride (NiMH) batteries, and Lithium-Ion Batteries were deemed the most viable options to consider. Due to their high energy density and relatively low cost, it was decided that Lithium-Ion batteries would be the best option to use in the design of the AUV. Lead-acid batteries, while extremely robust, have a low energy density and would add too much unnecessary weight and density to the system. Furthermore, since the batteries were to be stored in an enclosed environment, the robustness of the lead-acid batteries held relatively low value in our decision-making process. Comparable NiMH batteries could not supply the desired sustained power supply that the inherited high-quality rechargeable lithium-ion batteries could.

*Computer Vision*

When selecting the cameras for the Computer Vision module, the cameras have to satisfy a certain set of requirements in order for them to be suitable for the system. In table 1 three potential cameras have been listed and will be compared in order to select the best camera for the project.

*Table 3: Camera Selection*



Feature	Logitech QuickCam Pro 4000	Logitech WebCam Pro 9000	Logitech WebCam C615
Driver	Philips USB Webcam (PWC)	USB Video Class (UVC)	USB Video Class (UVC)
Number of Pixels	1.3MP	2MP	2MP
Max Resolution	1280 x 960	1600 x 1200	1920 x 1080
Auto-Focus	No	Yes	Yes
Auto-Light Adjust	No	Yes	Yes
USB Interface	1.1	2.0	2.0
Tri-pod Mount	No	No	Yes
Price	Free (inherited hardware)	\$55	\$55

In order for the cameras to interface with the OpenCV libraries, it was important for the webcams to be USB Video Class (UVC) compatible cameras. The Logitech QuickCam Pro 4000 did not satisfy this requirement and was, therefore, not suitable for the implementation in the Computer Vision module. The number of Pixels and maximum resolution are also important features since more detailed images would result in better accuracy for the algorithms. Both the Webcam Pro 9000 and Webcam C615 had satisfactory specs in regards to pixels and resolution. Auto-Focus was a definite must since this would result in a sharp and detailed image, which is a requirement for image processing. Furthermore, due to the fact that the image processing is done underwater, lighting conditions vary greatly; it was, therefore, important that the camera automatically adjusts to the light settings. Auto-Light Adjust would result in better pictures in low or harsh lighting conditions. USB 2.0 is needed for interfacing with the camera since this enables faster transmission rates. Both the WebCam Pro 9000 and WebCam C615 satisfied the requirements for Auto-Focus, Auto-Light Adjust and the USB interface. Since the price of these two cameras was the same, the only factor that differed was the tri-pod mounting capabilities of the WebCam C615. The tri-pod mounting capabilities were determining to give the team more flexibility in regards to mounting of the web cameras in their enclosures. Considering all the advantages, it was decided that the Logitech Webcam C615 would be used as the cameras for the Computer Vision module.

## *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) is the result. An IMU 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.

## *Inertial measurement unit (IMU)*

An IGS is programmed with an initial input and then uses its IMU to update its position relative to that initial position. This initial input can be provided by many sources such as a Global Position System (GPS), human input, or input from other sensory devices. For the purposes of this investigation and because wireless communication was prohibited to and from the Robosub (per the competition guidelines) and the fact that GPS does not work underwater, concept generation was restricted to IMUs that do not include GPS sensors. Following considerable research into various IMU options, it was determined that the inherited PhidgetSpatial 3/3/3 IMU would not only be the best option due to its compact size, incurred monetary savings, and compatibility with multiple common microcontrollers including Arduino Uno boards, but most importantly because of the plethora of resources available for this particular product in regards to sample code, programming, and technical information.

## *Pressure Transducer*

Since the Robosub will be underwater, an additional level of measurement is required to create a comprehensive IGS—depth. There are various methods of measuring depth:

- Sending an acoustic signal towards the floor and measuring the amount of time it takes to receive a response (Ultrasonic Sensor)
- Combining information from the IMU and the Robosub's initial position to calculate the vehicular depth in the water
- Integrating a submersible pressure transducer (either gauge, absolute, or sealed gauge)

From the options listed, using a submersible pressure transducer was deemed the best option. Using only the IMU to determine depth is tricky because it requires high frequency and high resolution polling of both the accelerometer and gyroscope, and could the introduction of drift could lead to significant error throughout the mission. A sealed gauge IMCL submersible pressure transducer with a marine bronze housing, polyurethane (PUR) cable, and a 0 – 5 VDC output in the corresponding range of 0 mWG – 10mWG (meters of water gauge). Sealed gauge was chosen over gauge because

the gauge transducer contains a vent cable that must be routed to a dry box—this requirement clearly was insufficient for our application. Furthermore, the absolute option, while not containing an inconvenient vent cable, would have yielded an output of 2.5 V at atmospheric pressure, and 5 V at 10 mWG (i.e. about 30 feet), thus reducing the accuracy of the device for the given operating range of 0 mWG to about 5 mWG. The sealed gauge option contained all the benefits of the absolute option, but also provided the desired 0 – 5 V output within the range of 0 mWG – 10 mWG, making it ideal for implementation on our AUV.



**Figure 2: IMCL Submersible Pressure Transducer**

The PUR cable and marine bronze (over stainless steel) housing were chosen for their great corrosion resistant properties—particularly in salt water. In addition, the 0 – 5 VDC output option was selected over the standard 4 mA – 20 mA output option because it was desired to send the output readings of the transducer directly to an A/D converter, which requires a 0 – 5 V analog input. Furthermore, due to the relatively short cable length in our application, the potential increase in noise and thus reduction in accuracy of the device that might result from transmitting a voltage signal versus a current signal, was not a concern—plus it contributed to the aesthetic appeal of the design as well.

### ***Hydrophones***

One of the competition tasks requires that the AUV be capable of detecting and triangulating the location of an acoustic pinger located in the saltwater pool. Following extensive research—particularly of veteran teams’ successful AUV designs—it was determined that the implementation of an array of hydrophones (essentially submersible passive microphones) would clearly be the best approach. By registering time delays between signals and applying basic trigonometric algorithms, the heading of the pinger could be determined. The piezoelectric hydrophones selected needed to be able to pass an acoustic signal within the pinger frequency range of 22 kHz and 30 kHz, as specified in the

final competition rules. Furthermore, cost, size, noise cancellation, sensitivity, and ease of integration with custom preamplifiers (purchasing preamplifiers or hydrophones with preamplifiers would have cost at least \$1,000 more) were deemed critical factors.

Despite their high quality and appropriate physical characteristics, the obscenely high cost of the Reson TC-4013 hydrophones (i.e. \$1,400 each) forced them to be absolved as a viable candidate for implementation on the AUV. Ultimately, it was experimentally determined that the two inherited hydrophones from last year's team (Sensortech SQ26-01), which were specified to only register input frequencies up to 28 kHz, were capable of responding to frequencies up to about 35 kHz. The compatibility of these hydrophones with our proposed preamplifier circuit, their compact size, their ability to be conveniently mounted on the vehicle in the desired location (i.e. the front of the AUV—away from visual, acoustic, or magnetic interference), and the vast savings that would incur as a result of only having to purchase two more of these inexpensive, yet reliable and applicable hydrophones (\$200 each after a \$20 student discount per hydrophone) made these SQ26-01 towed array hydrophones the clear choice for implementation on our AUV.

## Vehicle Propulsion System

Research indicated that thrusters were used by every single team in competitions past, and were the obvious choice for the vehicle propulsion system. They are comprised of brushed DC motors and propellers, and are able to equally rotate both clockwise and counterclockwise. Thus, they can provide equal force in both directions. They are also easily controllable by a microcontroller, and their speed, and thus thrust force can be adjusted simply by varying an input PWM signal. This characteristic would provide dynamic versatility for the vehicle. In addition, their integration with the commonly-used L298 Dual H-Bridge Motor Drivers made thrusters an attractive option. Furthermore, since the 2011 FAMU-FSU College of Engineering RoboSub design team handed down four functioning quality thrusters valued at over \$2,000, there was yet another incentive to use these established, reliable, watertight devices from a trusted supplier in SeaBotix for vehicle propulsion and general maneuverability. Two more SeaBotix BTD150 thrusters were purchased, however, so that the vehicle could adorn three pairs of thusters—one thruster along the centerline of each of the six faces of the symmetrical AUV design in order to enable translation and rotation along and about each of the major axis.

## Mechanical Subsystems

### *Grasp/Release Mechanism*

The *Laurel Wreath and Emperor's Palace* task requires the AUV to be able to grab and release a circular PVC wreath upon command. Thus, a grasp/release mechanism needed to be designed and constructed. Four viable concepts for this mechanical subsystem were derived.

**Concept 1** was to use pneumatics to actuate a mirrored three-bar linkage grasping mechanism—high-pressure gas would be released upon command (solenoid valve actuation) into a single-acting air cylinder with a built-in spring return in order to drive two jaws to close around the object. When the object was to be released, the first valve would be actuated to close and a second solenoid valve located between the entrance to the air cylinder and the first valve would be actuated to open, thus releasing the pressure inside the chamber and allowing the spring return to retract the piston/claw and release the circular PVC wreath.

**Concept 2** is similar to concept 1 in that it also incorporates the use of an air cylinder driven by the actuated opening of a solenoid valve and subsequent release of compressed gas. However in this design, rather than inducing rotational motion of the actuated claw, the thrust of the air cylinder piston would induce translation of a c-shaped arm about, compressing it into a complementary c-shaped arm that would be rigidly fixed to the frame.

**Concept 3** was to implement a submersible electric linear actuator in order push the actuated c-shaped arm horizontally into the PVC object and against a similar complimentary, stationary claw. Concept 3 was essentially the integration of a linear motor into Concept 2, in place of the pneumatic device and solenoid valves.

**Concept 4** was to implement an electric linear actuator in order rotate the same mirrored 3-bar slider-crank linkage proposed in Concept 1 around the PVC object. The jaws could then release the object by simply actuation the linear motor in the opposite direction. Concept 4 was essentially the integration of a linear motor to Concept 2, in place of the air cylinder, solenoid valves, compressed air tank, gas lines, etc.

**Table 3: Grasp/Release Mechanism Decision Matrix**

Design Criteria	Ease of Manufacture/ Implementation		Reliability/ Durability		Weight		Grasp Force Per Unit Power Consumption		Cost		Sum
	Weighting Factor										
Concept 1	4	1.50	6	2.10	7	0.70	8	1.80	8	0.80	6.10
Concept 2	5	1.20	7	1.80	7	0.70	9	1.60	8	0.80	6.90
Concept 3	7	2.10	8	2.40	6	0.60	7	1.40	1	0.10	6.60
Concept 4	6	1.80	7	2.10	6	0.60	6	1.20	1	0.10	5.80

In review of the above decision matrix, ease of manufacture/implementation and reliability/durability were deemed the most important factors in the selection process for the grasp/release mechanism design. Concept 2 was deemed the best option due to its relatively high reliability and theoretical consistency, low weight, low cost, and significantly greater ratio of grasping range versus air cylinder displacement (and thus size and cost). Although the integration of automated submersible solenoid valves was thought to provide a challenge, the design team deemed it sufficiently viable and a necessary risk. Furthermore, the decision to thus integrate a compressed air distribution system opened the door for implementation in the design and operation of other mechanical subsystems as well (see torpedo launchers)—thus increasing the value of this decision.

### ***Marker Dropper***

The *Gladiator Ring* mission task requires the AUV to be able to drop markers in specific bins upon command during its mission. Thus, a controllable marker dropper mechanism needed to be designed. After extensive research and brainstorming, four viable concepts were developed for this mechanical subsystem, and are described below:

**Concept 1** involved a slight modification of the 2011 FAMU-FSU RoboSub team’s design. In this design, two markers (0.5-inch diameter steel balls) would be stored in a parabolic channel made out of Aluminum 6061—one on either side of the downward-facing servo. Upon command, the servo motor would be actuated to move to one of two set release positions (i.e. angle left or angle right), causing the servo to rotate out of the way and release one of the two markers. After returning to its initial position, the servo could then rotate in the opposite direction upon command, thus releasing the other marker into the subsequent desired bin. In order to ensure that each of the markers would drop vertically rather than at a slight angle (as occurs in the current design), however, a vertically oriented funnel was considered to be added to the outlet of the device as a guide for each of the steel balls upon release if deemed necessary. Furthermore, this design would require the custom waterproofing of the servo motor since it is clearly not rated for sustained submersion beyond depths of 3 ft—the IP67 rating would be insufficient for our application. In addition, a new, compatible mount would also need to be derived in order to make this design compatible with the desired attachment location on the AUV.

**Concept 2** involved the use of a pre-pressurized pneumatic air cylinder, a thin rectangular plate, and individual storage compartments for the markers. In this design, each of the two steel balls would be surrounded by an individual cylindrical enclosure, which would be attached to the bottom of the vehicle’s frame. Both markers would rest on a thin rectangular plate that was fixed to the piston of an air cylinder so that when one of the two solenoid valves is actuated to open, the compressed gas would be released into the surrounding water, the spring inside the single-acting air cylinder would retract the piston/rectangular plate, and each marker would thus be independently released into the

appropriate bin. The pressure could be released in stages via the timed, multi-stage opening and closing of the pressure release valve (similar to the second valve in the design of the grasp/release mechanism).

**Concept 3** was to implement a servo motor with a thin, cylindrical, horizontally-oriented disk attached to the end of its shaft. The disk would have a circular cutout with a diameter approximately 25% larger than that of the steel balls. The markers would rest above the disk, surrounded by individual cylindrical barriers similar to those in concept 2. The actuation of the servo motor would cause rotation of the disk, allowing the cutout to lie directly underneath one of the two markers, thus allowing it to drop into the designated bin. After returning to its initial position, the servo would then be able to rotate the same amount in the opposition direction in order to allow the other marker to drop through this gap upon command.

**Concept 4** required the use of two electromagnets to hold onto each of the two markers during the mission. Once the target (i.e. the proper bin) was detected and reached, one of the electromagnets would shut off, thus allowing the release of the respective steel ball. Each submersible electromagnet would be independently controlled.

**Table 4: Marker Dropper Decision Matrix**

Design Criteria	Ease of Manufacture/ Implementation		Reliability/ Durability		Weight		Power Consumption		Cost		Sum
Weighting Factor	0.35		0.30		0.10		0.15		0.10		1.00
Concept 1	9	3.15	8	2.70	6	0.60	9	1.35	10	1.00	8.50
Concept 2	5	1.75	7	2.10	7	0.70	7	1.05	7	0.70	6.30
Concept 3	9	3.15	7	2.40	9	0.90	9	1.35	8	0.80	8.30
Concept 4	4	1.40	7	2.10	8	0.80	4	0.60	4	0.40	5.30

Upon analyzing the decision matrix, it was determined that the best approach was to make slight modifications to the design developed by the 2011 FAMU-FSU RoboSub team. Although concept 3 may have yielded a slightly better design, the benefits gained would likely not outweigh the time and energy saved by settling for an established mechanical subsystem that was near complete functionality. The pneumatic air cylinder required for concept 2 would have yielded a more complicated and potentially less reliable design that would be more difficult to implement than those that utilized a servo motor as the actuator instead. Furthermore, the difficulty of manufacture, relatively large power consumption, and relatively high cost involved in the production of concept 4 yielded it as a poor option.

## *Torpedo Launchers*

The *Et Tu Brute* mission task implies that the AUV must also be capable of independently shooting two torpedoes through circular PVC cut-outs—each with a different size and color. Thus, the design team decided to construct two identical torpedo launchers and place both of these mechanical subsystems on the horizontal neutral axis on either side of the vehicle to maintain symmetry, balance, and stability. Three viable design concepts were generated following extensive research of previous years' successful competition designs, as well as original brainstorming. Descriptions of each of these concepts can be found below:

**Concept 1** involved the design of self-propelled torpedoes. An on-board power supply and motor would drive the rotation of attached propellers, thus driving the torpedo out of the barrel via thrust. The general shape of the torpedoes would be similar to an elongated blimp (i.e. a prolate spheroid), with a flat bottom, in order to yield a hydrodynamic form and reduced disturbance during launch.

**Concept 2** utilized a pneumatic system, which would enable shared use of the compressed air distribution system that was proposed for the grasp/release mechanism. Two solenoid valves would individually control the flow of compressed gas to each of the two torpedo launchers. Each torpedo launcher would contain a double-acting air cylinder and a surrounding barrel to house the ABS plastic, 3D-printed torpedoes. It was determined that if this route was to be selected, the torpedoes could be modified to increase their density to that of saltwater (in order to yield a straight flight path) while maintaining balance and a hydrodynamic external form. Upon command, one of the two solenoid valves would be electrically actuated to open, thus releasing the high pressure gas from the regulator, through gas lines, and into the respective air cylinder, resulting in the rapid extension of the piston, and thus the propulsion of the torpedo. Immediately thereafter, the entry valve would be electronically actuated to close. Each torpedo could be actuated independently via the independent control of each solenoid valve.

**Concept 3** implemented the use of a spring-loaded device. This design was similar to both concepts 1 and 2 in regards to the shape of the torpedoes and use of cylindrical barrels as guides. However, this concept would have incorporated a pre-loaded spring in each of the torpedo launchers, against which one of the torpedoes would rest. A stepper motor or servo motor would be used to trigger the release of the spring, and thus stored energy, into the torpedo.



**Table 5: Torpedo Launcher Decision Matrix**

Design Criteria	Versatility/Safety Regulation		Reliability		Ease of Manufacture/Implementation		Cost		Weight		Power Requirement		Sum
Weighting Factor	0.25		0.25		0.15		0.10		0.15		0.10		1.00
Concept 1	5	1.25	5	1.25	3	0.45	6	0.60	5	0.75	7	0.70	5.00
Concept 2	8	2.00	8	2.00	6	0.90	8	0.80	7	1.05	9	0.90	7.65
Concept 3	8	2.00	7	1.75	5	0.75	8	0.80	8	1.20	5	0.50	7.00

In reference to the above decision matrix, the design team concluded that the most important design criteria to consider in the design of this mechanical subsystem were the safety and reliability of the mechanism; according to the competition committee, the torpedoes should not be capable of causing bodily harm upon accidental impact. The safety of the mechanism would be ensured via the control of the regulator compressed air pressure, and thus the launch velocity and accuracy of the subsystem. The reliability of the device and ease of manufacture/implementation were also given strong consideration in the decision process, as high scores in these factors would likely yield the efficient production of a consistently executable device. Due to these factors, along with the strong possibility of using the same compressed air distribution system to power the grasp/release mechanism as well, Concept 2 was deemed the best design concept and was selected to be carried out in the overall design of the AUV.

## Final Concept

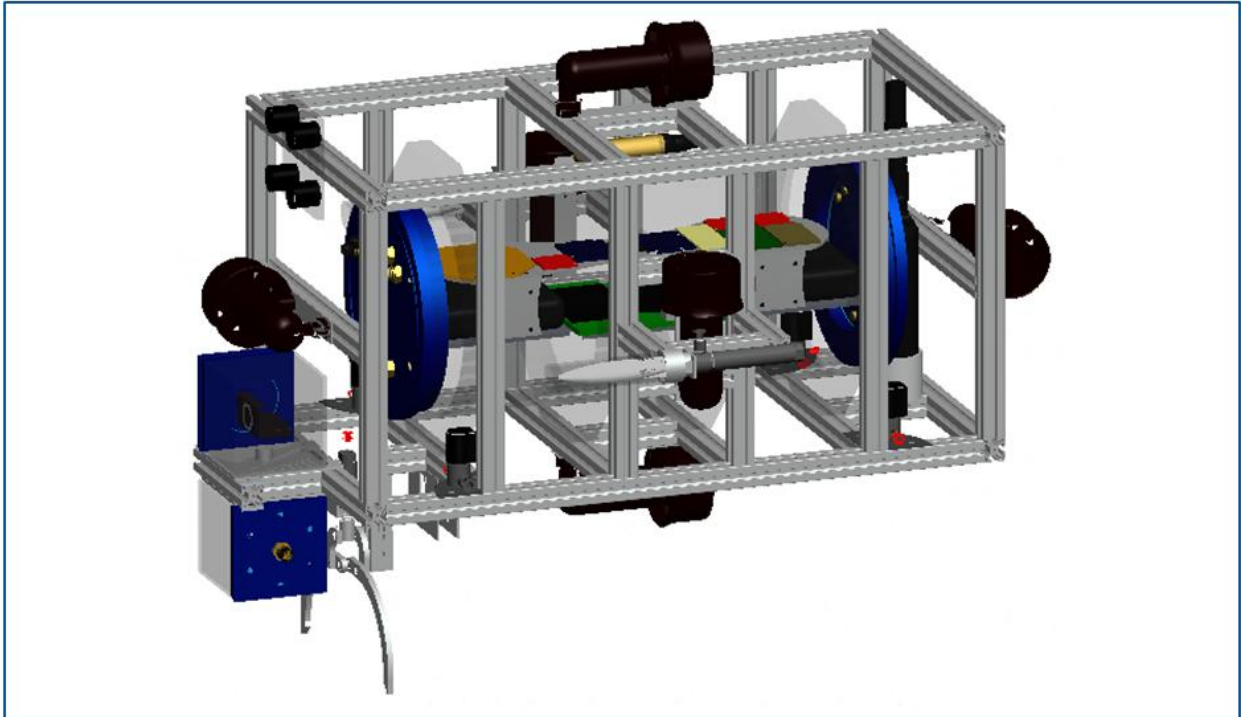


Figure 3: Pro/Engineer Model of the Final AUV Design - Angled View

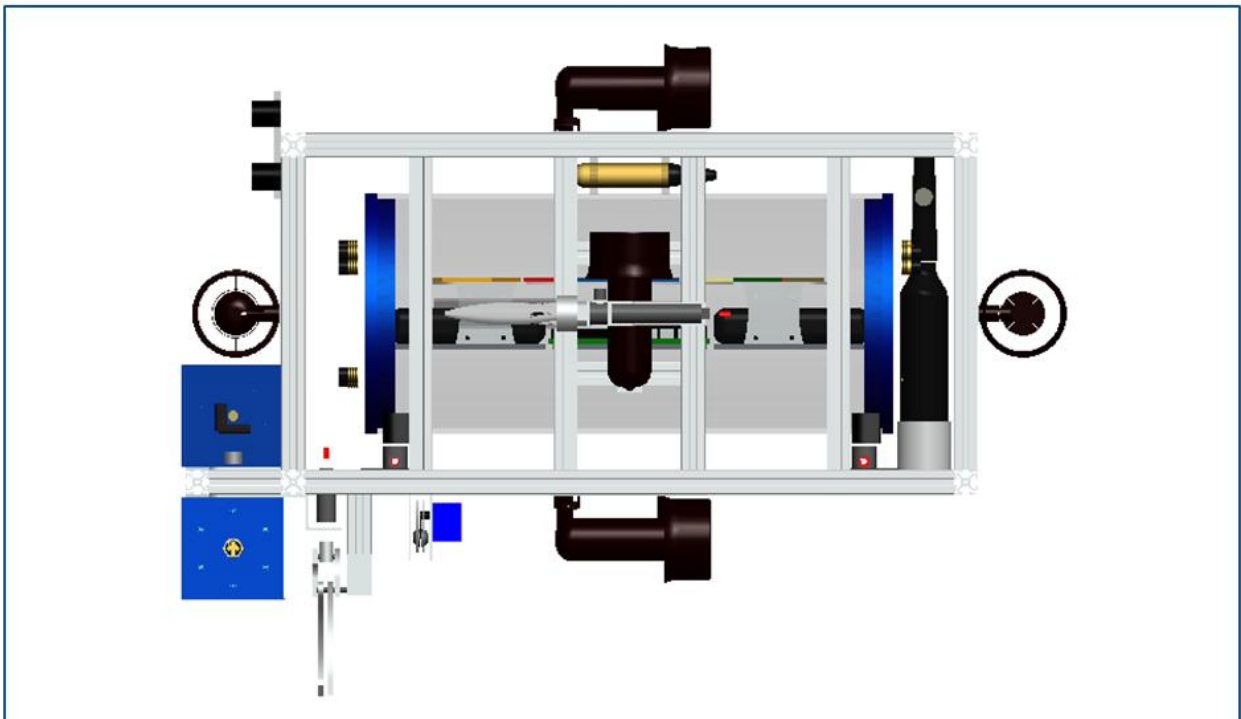


Figure 4: Pro/Engineer Model of the Final AUV Design - Side View

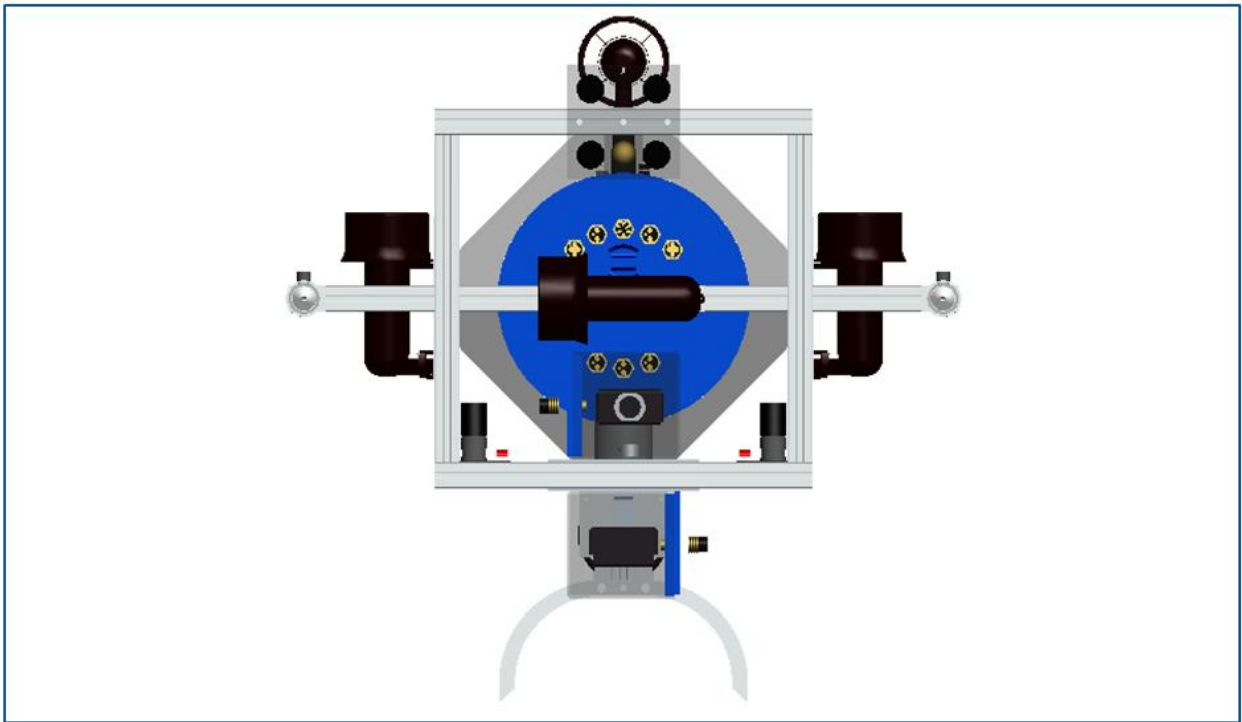


Figure 5: Pro/Engineer Model of the Final AUV Design - Front

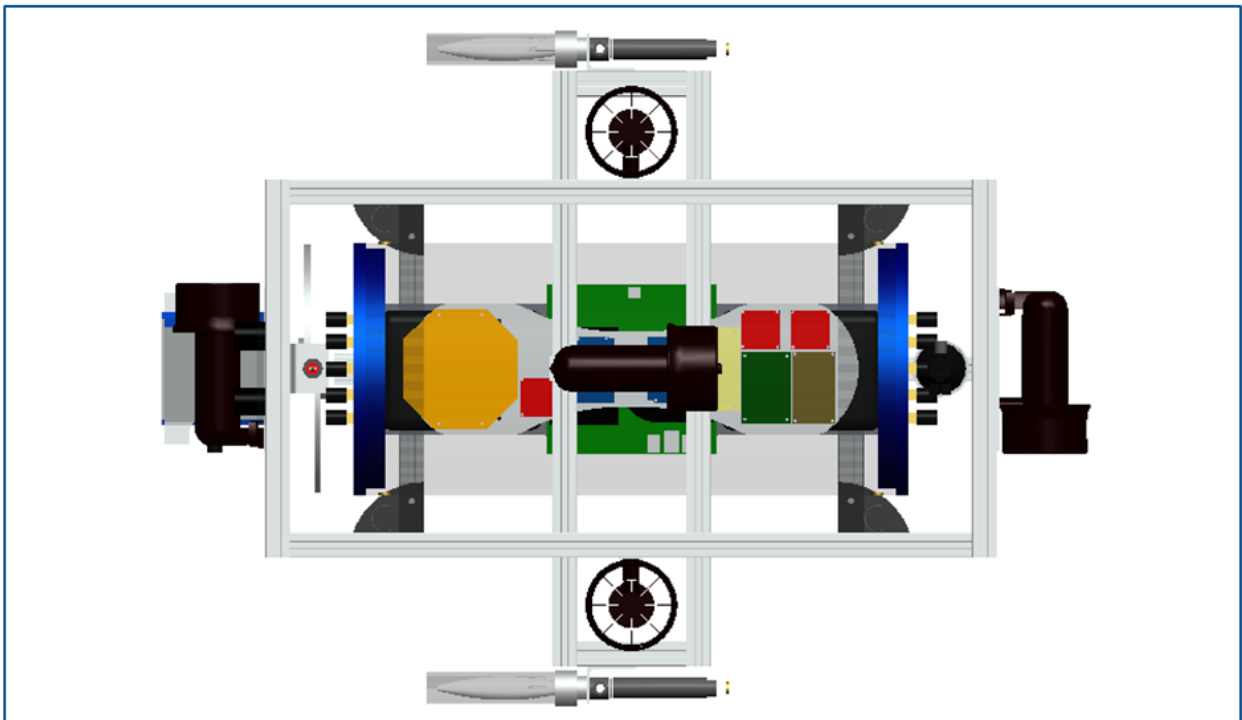


Figure 6: Pro/Engineer Model of the Final AUV Design - Top View

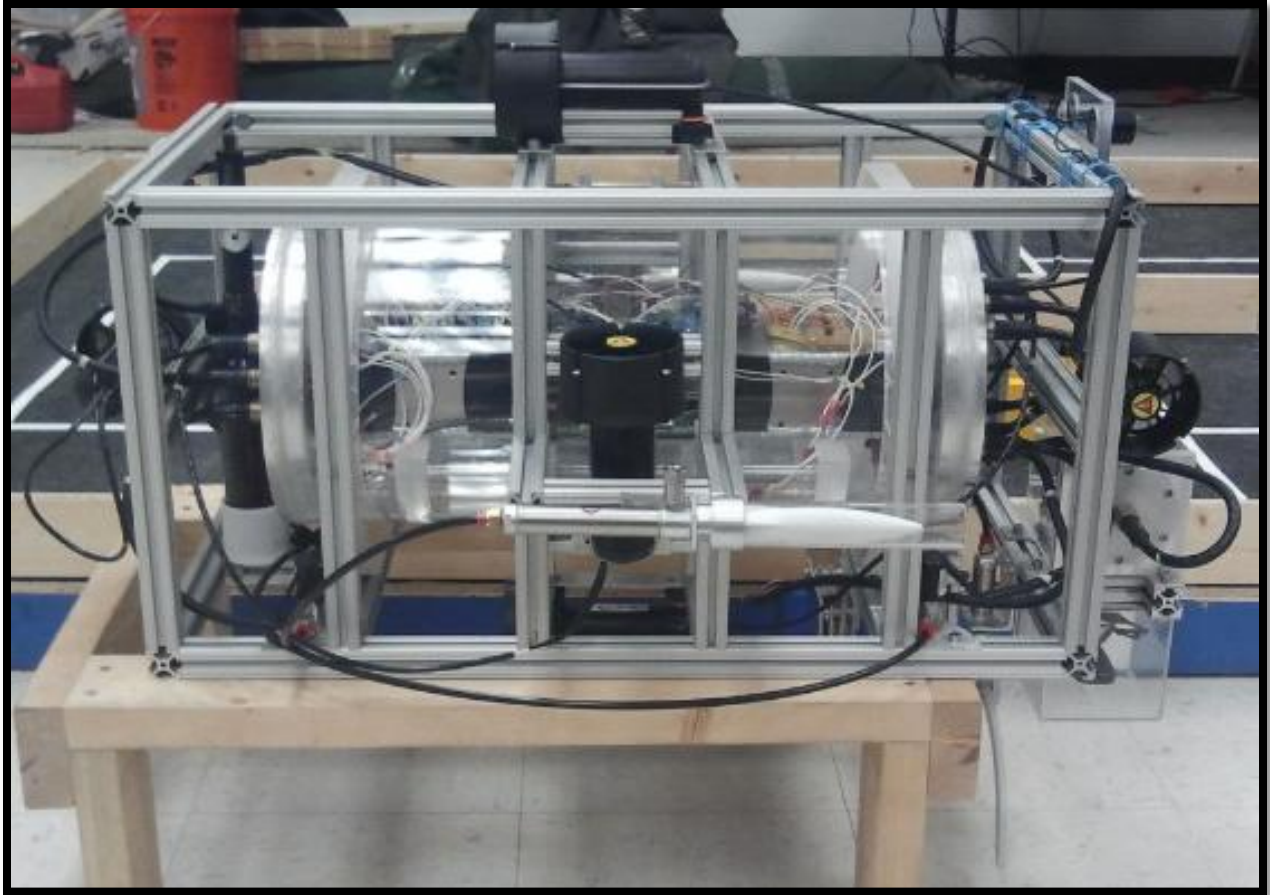


Figure 7: Photograph of the Assembled AUV - Angled View

### 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 seems). 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  $\frac{1}{4}$ " - 20 bolts are used to attach subsystem mounts/platforms to the frame at various locations.

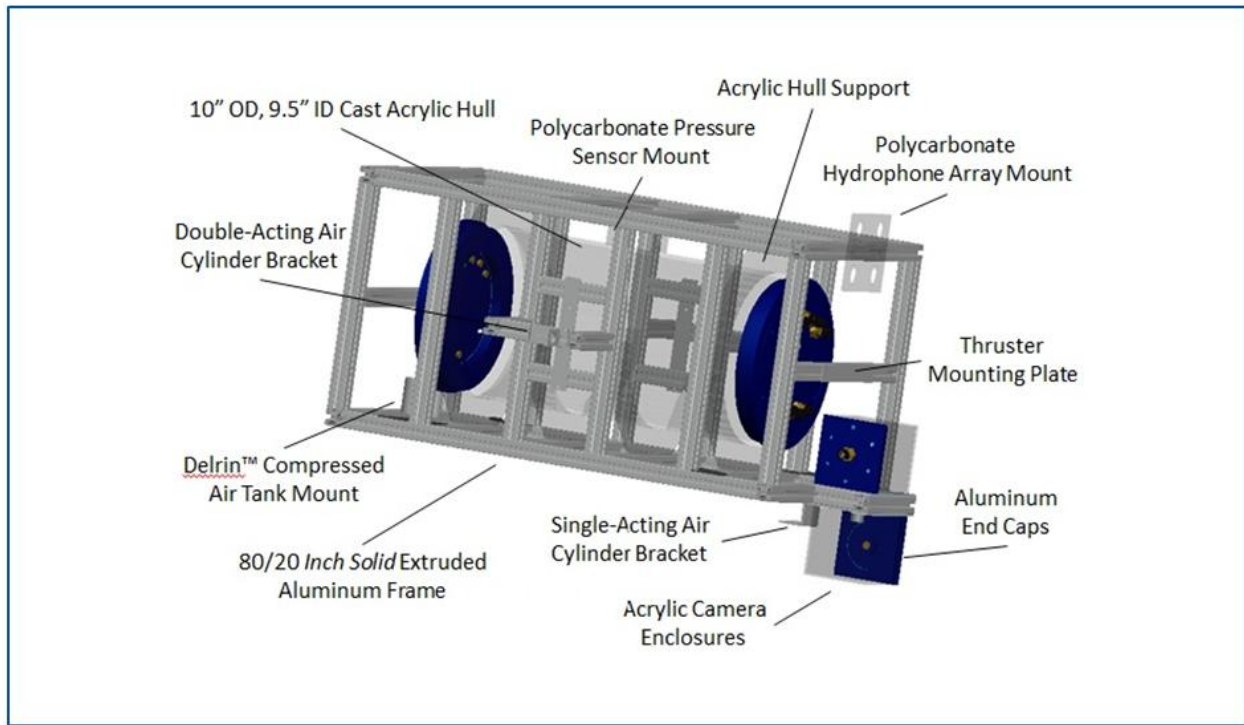


Figure 8: Close-up view of the Hull / Frame

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 serves to house the electronics in a secure, watertight enclosure. The ¼" thick walls of the precisely manufactured, cast acrylic tube yield it capable of withstanding the relatively moderate hydrodynamic and hydrostatic pressure that it will encounter at its maximum depth of about 13 ft in salt water (the tasks are elevated a couple feet from the 16 ft-deep floor) without any measureable deflection or deformation. Furthermore, the clear acrylic material has a low density (i.e. only slightly greater than salt water), has proven applications in similar environments (e.g. used for walls of aquariums, as well as in other successful AUV designs), provides enough positive buoyancy (mainly due to the trapped air inside the enclosed volume) to counteract the denser surrounding components of the AUV, and allows the electronics to be seen from the exterior of the system—an aesthetic bonus, as well as a prime location for sponsorship logos.

A two-piece end cap design was implemented for both the main hull and the camera enclosures. The outside part of each end cap is rigidly attached to the interior surface and smooth ends of the acrylic hull via marine-grade, 100% silicone caulking. This fixed piece contains an open center with a lip containing six threaded holes, 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 logically featuring a square shape rather than circular.

Submersible SEACON™ All-Wet and Micro Wet-Con connectors are 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). The implementation of tri-split contact configurations for the thruster connectors rather than individual connectors considerably reduced the amount of holes that needed to be drilled into each end cap. This design decision resulted in more reliable water-tight seals, as well as a more compact and direct internal wiring scheme. The female SEACON™ connectors simply screw directly into the corresponding end caps, with Loctite™ thread sealant added to secure the female connectors permanently in place and prevent any potential leakage through the threads. The male SEACON™ connectors/cables plug directly into the female sockets, and the electrical leads of these cables are soldered to the leads of their complementary peripheral electrical subsystem (e.g. solenoid valves, thrusters, pressure transducer, etc.). Heat shrink tubing, in conjunction with Plasti-Dip™ liquid rubber are used to seal and protect the exposed wires at these interfaces from water intrusion when the AUV is submerged.

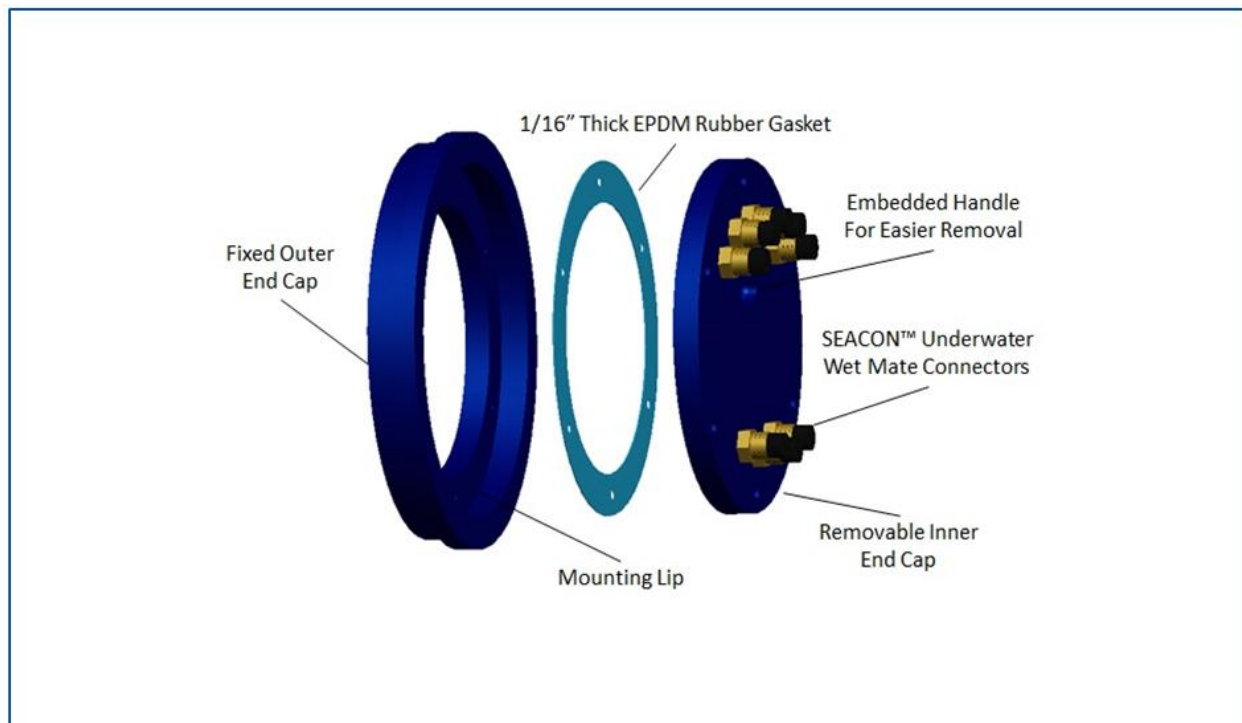


Figure 9: Close-up Exploded View of the Hull End Caps

## Interior Hull Design

The interior of the hull contains a 3/16"-thick aluminum electronics platform which supports the two 14.8 V lithium-ion batteries (the batteries each output 16.5 V in practice), two Arduino Uno Boards, a Zotac PC, three L298 dual h-bridge motor drivers, an inertial measurement unit, a solenoid valve interface circuit (i.e. four identical low-side drive BJT circuits), a custom preamplifier interface circuit for the hydrophones, two voltage regulator boards, and thermally-conductive aluminum plates which connect the heat dissipation plate from each motor driver directly to the top surface of the aluminum electronics platform. The use of the compact L298 motor drivers (i.e. two on each chip) ensures that only a power and ground wire (i.e. two-pin cable) are needed 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. 100% silicone caulking was used to fix the curved surface of each of these cut-outs to the interior surface of the hull. 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.

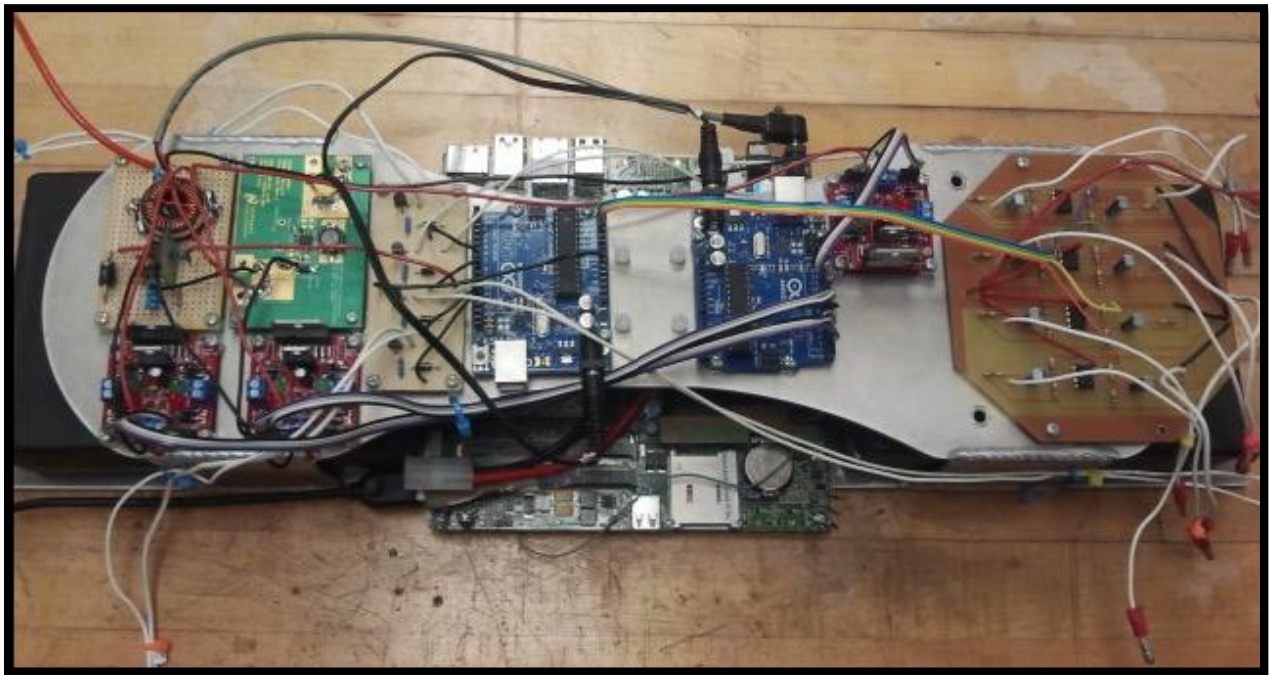


Figure 10: Electronics Platform (IMU—Center—Not Shown)

The aluminum platform has a 3/16" thick aluminum base to provide rigidity, and contains an upper level which is welded to four vertical supports. The decision to weld the electronics platform into a single entity rather than use an alternative, less permanent attachment method such as angle brackets, nuts and bolts has proved wise and has yielded a far more durable structure. The electronics platform is easily removable, providing relatively convenient access to the electronics.

Threaded nylon standoffs are screwed into the electronics platform and serve to support the interface circuits, microcontrollers, IMU, and PC while also ensuring these components don't come into contact with the aluminum platform and introduce the significant potential for short circuits. The highly dense lithium-ion batteries are placed on either end of the electronics platform in order to maintain symmetry and balance in the AUV, and also to make it less of a hassle to recharge the batteries on occasion. The internal SEACON™ wires were cut relatively short, and each contain quick-disconnect adapters. This enables the internal electronics to be detached from the end caps after simply unscrewing the hull end caps and retracting them slightly, rather than having to reach well inside the hull in order to detach all the connected wires.

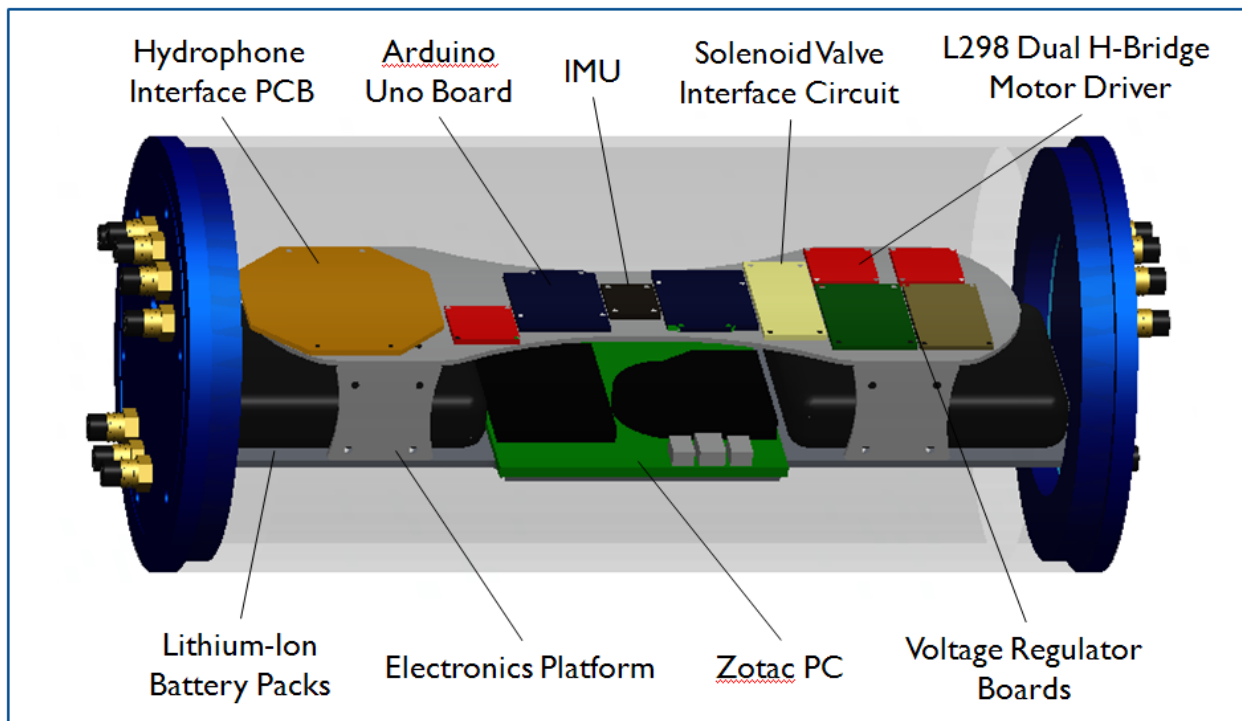


Figure 11: Close-Up View of the Interior Hull Layout

In keeping with the theme of wire management inside the hull, colored zip ties, colored ribbon cable (used for signal lines between the Arduino Uno boards and the L298 Dual H-Bridge motor drivers, as well as the in initial attempt at a hydrophones interface PCB), and numbered labels all contribute the organization of complex network of wires in such a confined space. Furthermore, since the Zotac PC was a late addition to the design, the following design modifications were made in order to enable its implementation—the fixed part of the rear end cap was carefully and laboriously carved away, and three unused ports were de-soldered from the Zotac PC in order to allow it to physically fit inside the hull. Furthermore, custom right-angle USB cables were fortunately discovered and implemented in order to be able to access the essential ports on the sides of the motherboard.



## Camera Enclosures

The AUV features two camera enclosures located along the midline of the front face of the vehicle. While the original 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 them 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") ensured the avoidance of potential leakage along the edges of each of the faces—a common and frustrating occurrence in the original custom enclosures. In addition, the dimensions of the acrylic boxes used were so similar to the initial custom enclosures that the removable inner end caps from the original camera enclosures did not have to be remanufactured as they were directly compatible with the revised design. Furthermore, the overall design of the vehicle did not need to be adapted whatsoever due to the decoupled nature of this design modification and the modularity of the AUV design.

The remodeled enclosures feature a thin aluminum plate fixed to either side of the T-slotted frame supports in order to provide a clean, smooth mating surface for the optically transparent acrylic enclosures. High-strength DUCO™ cement is used to non-intrusively bond the top/bottom acrylic face of each of the enclosures to its respective aluminum base, thus reducing the risk of leakage (e.g. versus a nut, bolt, and sealing washer attachment technique) and increasing their robustness and reliability. The top enclosure houses the front-facing Logitech C615 web camera, and the bottom enclosure houses the bottom-facing Logitech C615 web camera. This arrangement provides an unobstructed view for both web cameras and yields a more sensible positioning of the cameras to assuage the potential programming difficulty that might result from the cameras being offset from the center of the vehicle, or being located at severely different locations along the frame.

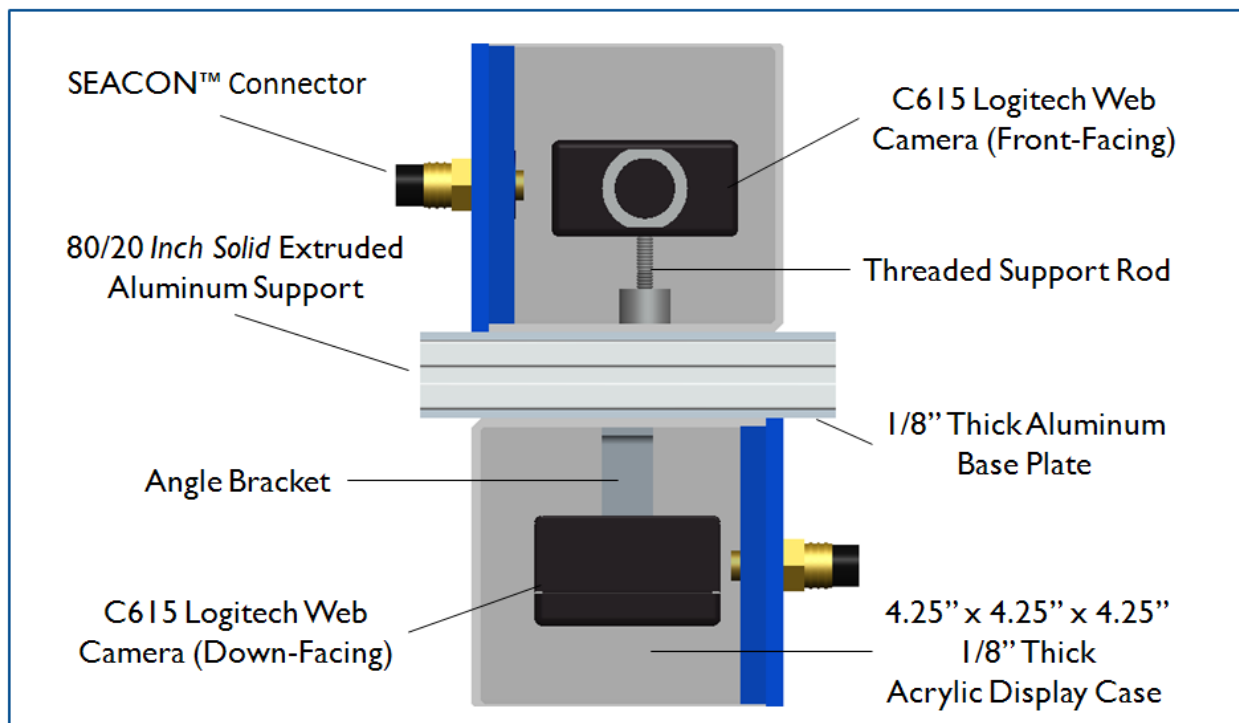


Figure 12: Close-Up View of the Camera Enclosures

## Vehicle Propulsion

Six thrusters are implemented on the AUV in order to propel the vehicle and provide stability, depth, and velocity control throughout the mission. Four SeaBotix SBT150 thrusters and two SeaBotix BTD150 thrusters are 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<sup>2</sup> have the same exact dimensions and weight since the I<sup>2</sup>C controllers in the SBT thrusters were extracted. Thus, each thruster operates via L298 Dual H-Bridge Motor Drivers located inside the main hull on the electronics platform. The data and clock ports on the SBT150 thrusters are not used as they are unnecessary for this implementation. Furthermore, the cables for the SBT150 thrusters were re-routed and potted (using waterproof epoxy, followed by Plasti-Dip™ liquid rubber for an insurance/aesthetic coating at the interface) in order to be able to mount them while still maintaining functionality—in fact the newer BTD150 thrusters already incorporated this potting technique as well. A single thruster is attached, via 1/16"-thick mounting plates, along one of the two centerlines of each face of the open, rectangular 80/20 frame in order to absolve any undesired torque on the system during operation, as well as to minimize unnecessary weight. The orientation and location of the thrusters enable the AUV to translate and rotate along and about each of the three major axes, resulting in an agile, easily maneuverable vehicle. The bidirectional nature of the thrusters via the simple software-enforced alteration of the motor direction is another convenient feature.

<sup>2</sup> Data sheets for the thrusters can be found in Appendix A

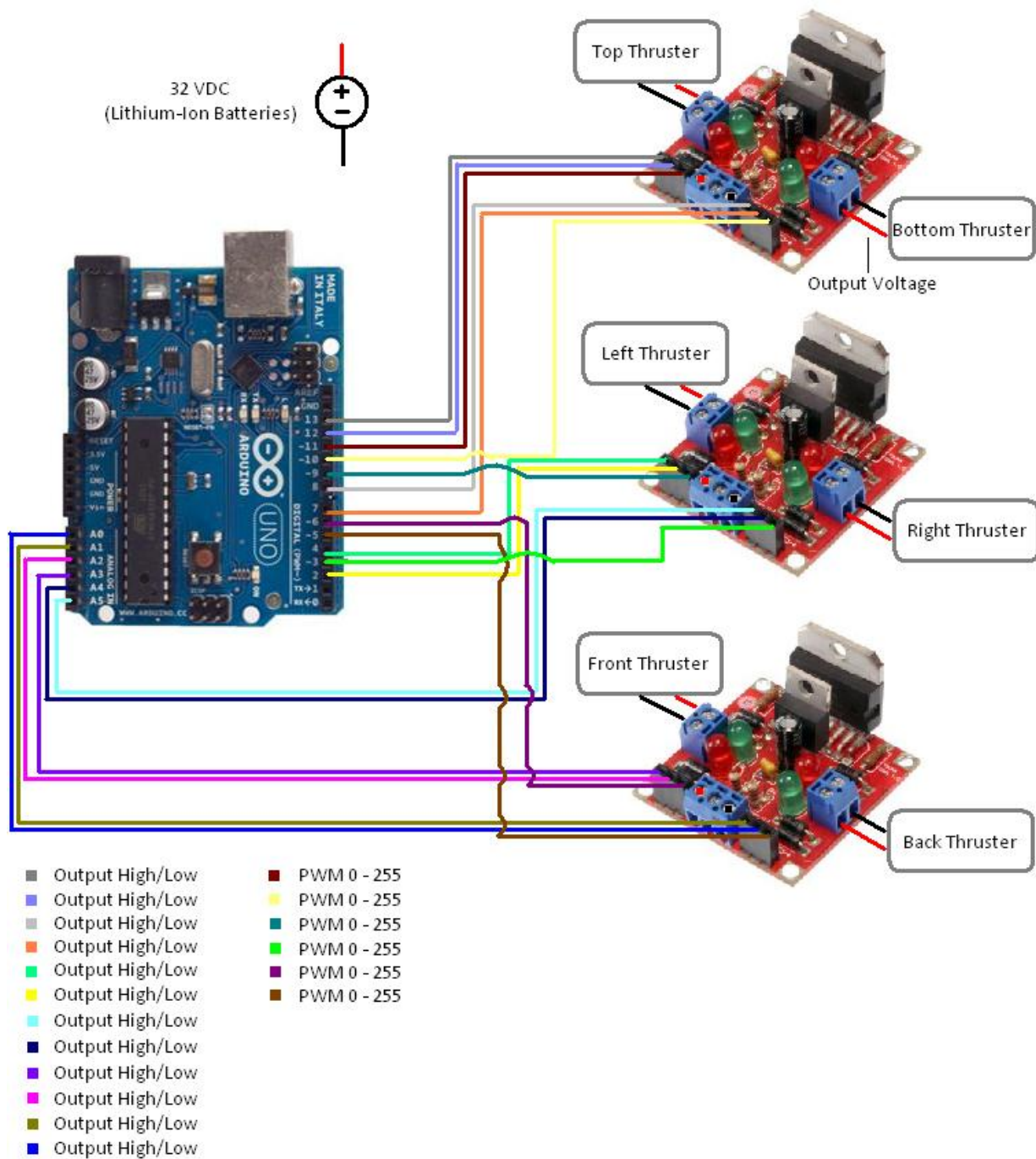


Figure 13: Wiring Schematic for Thrusters

The thrusters will receive PWM commands from the Arduino Board in response to interpreted IMU readings (i.e. PID closed-loop controllers), thus controlling the voltage/speed (and thrust force) of each thruster in order to obtain the desired position, velocity, and/or orientation. Considering the density of the vehicle relative to the density of salt water,

and applying Equation 1 below, only about 0.75 – 1.5 lbs of thrust force is required from the side thrusters in order to maintain a constant depth (i.e. zero velocity and zero acceleration).

This yields 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 nominal travel speed of the vehicle (i.e. about 2 – 2.5 ft/sec), only minimal thrust force is required from the top and bottom thrusters in order to maintain a constant longitudinal velocity. Provided the actual 33 V output from connecting both of the lithium-ion battery packs in series, combined with manageable current draw from the thrusters (i.e. no more than 1.5 A per thruster in bursts), there should be no problem supplying ample power to the thrusters during the mission.

## **Mechanical Subsystems**

### ***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 major components of the pneumatic system are the 13 cc compressed air tank, tank regulator, secondary/low pressure custom regulator, four submersible stainless steel solenoid valves, a network of ¼" OD nylon tubing, and multiple 1/8" NPT - ¼" 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<sub>2</sub> 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<sub>2</sub>, the greater availability of compressed air versus CO<sub>2</sub> (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<sub>2</sub> 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<sub>2</sub> 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 (in actuality, as little as 4 V are required for actuation, but 9 V are supplied due to the electrical circuit design), 1/8" NPT adapters, and are 1" in diameter x 2.5" in height. The solenoid valves are each connected to identical low-side drive BJT interface circuits in order to enable simple actuation via one of the Arduino Uno microcontrollers. The network of nylon, 250 psi-rated tubing, in combination with a quad- and tri- tube junction, enables 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 one-way 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 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:

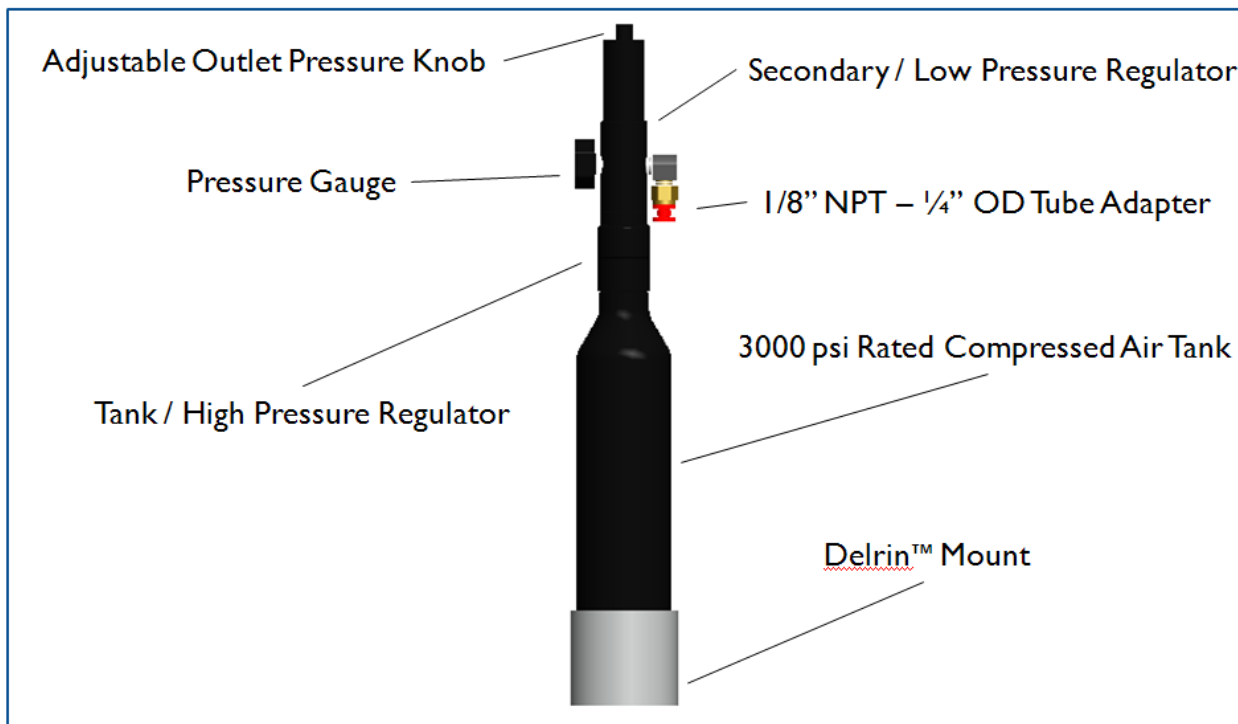


Figure 14: Close-Up View of the Compressed Air Tank and Regulators

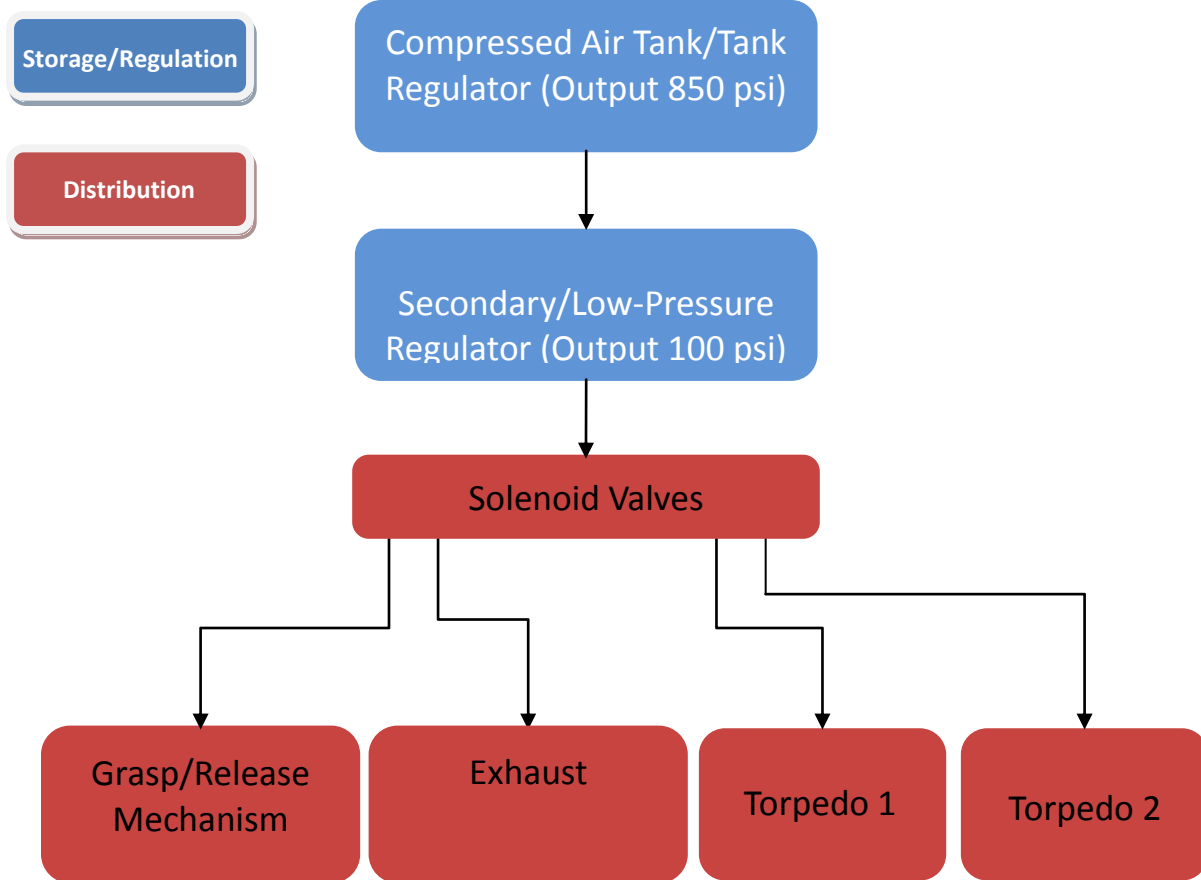


Figure 15: Compressed Air Distribution System Overview

## Grasp / Release Mechanism

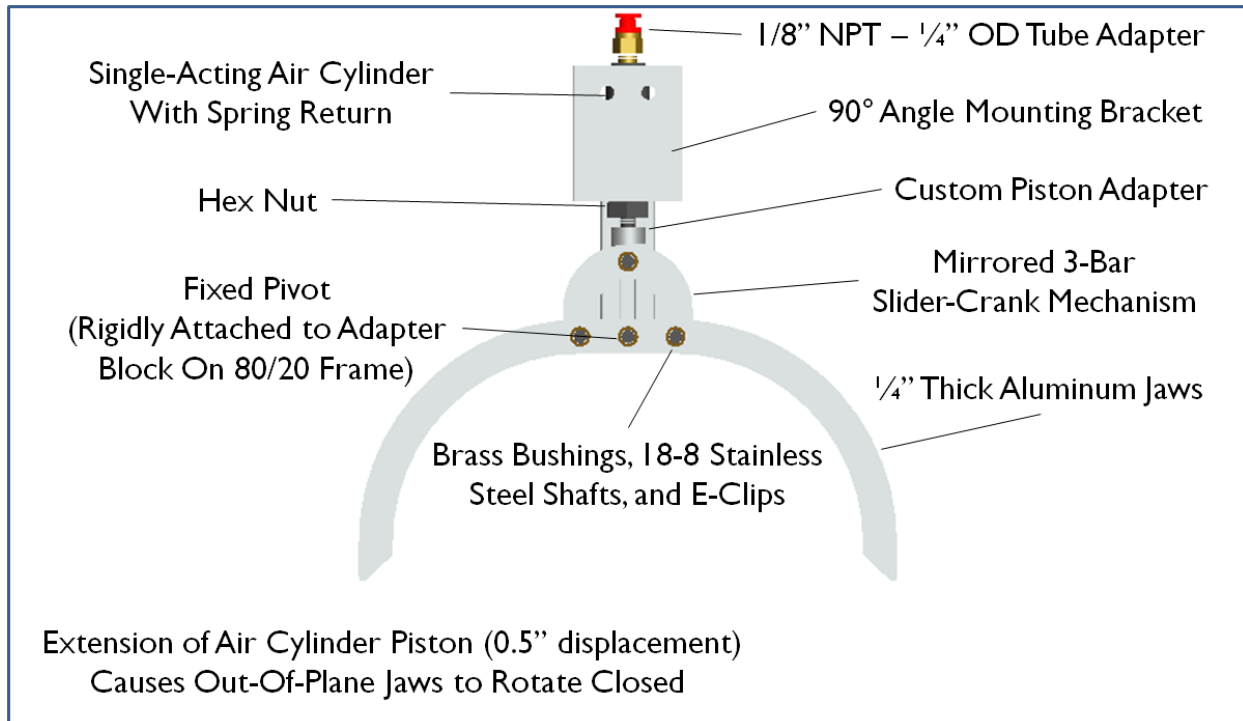


Figure 16: Close-Up View of Grasp / Release Mechanism

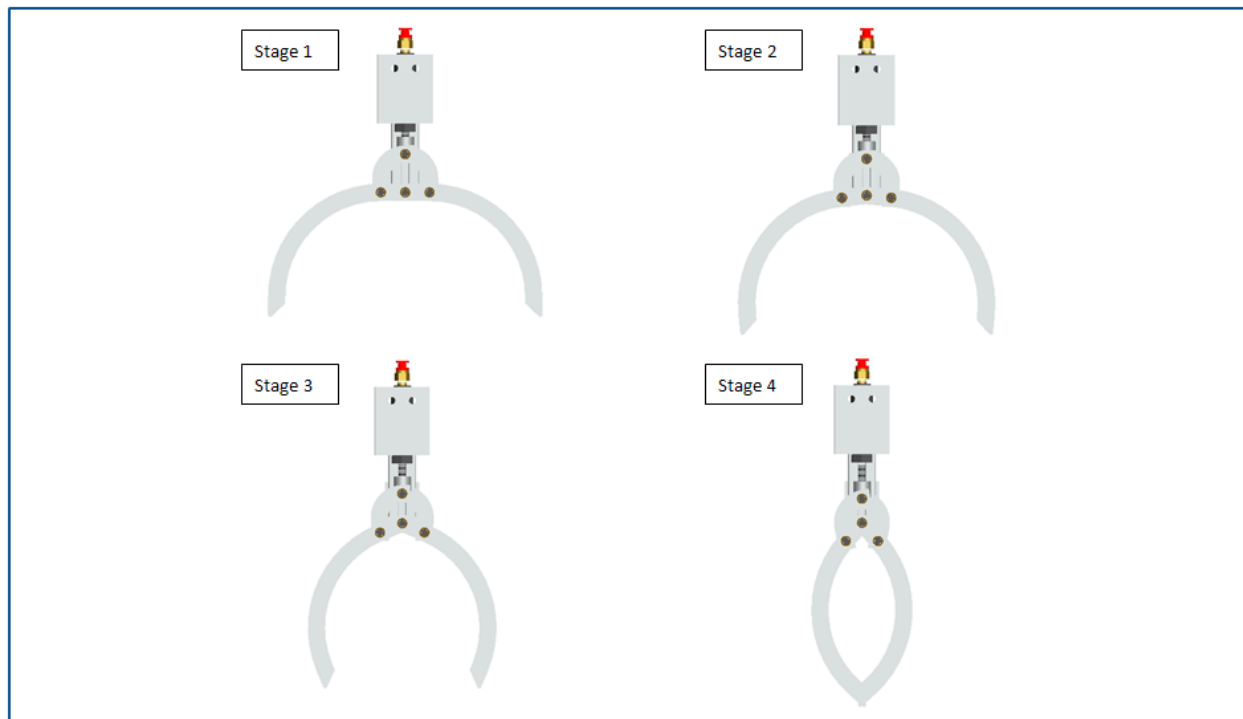
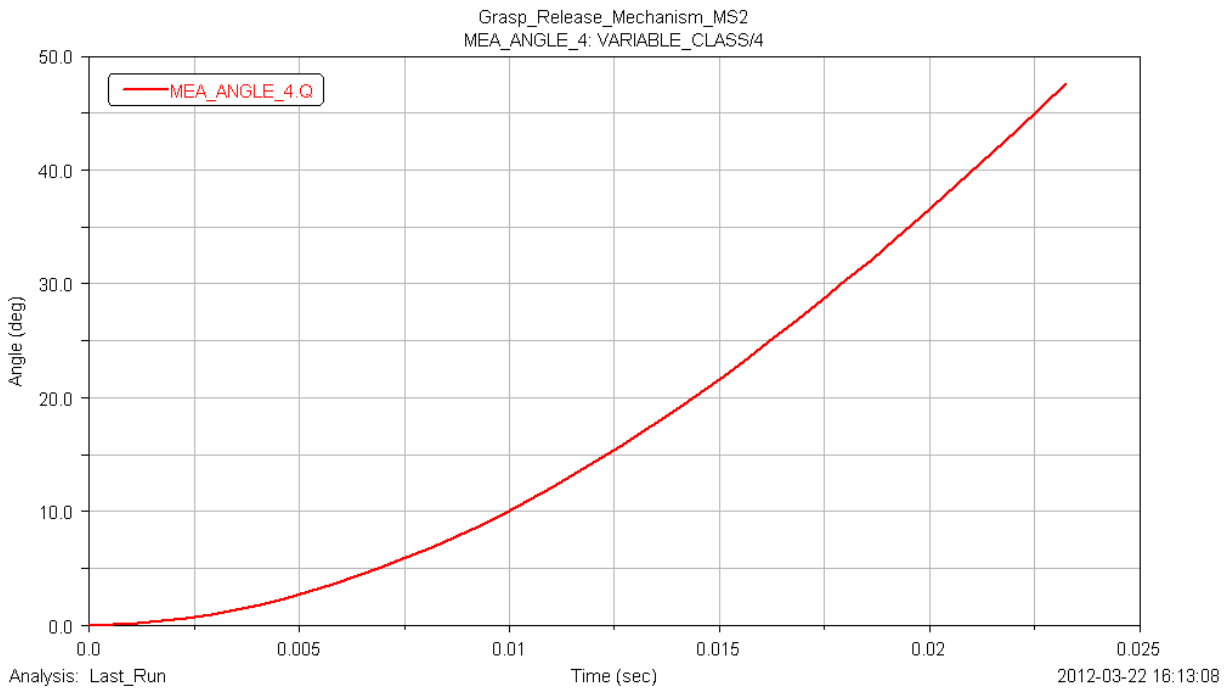
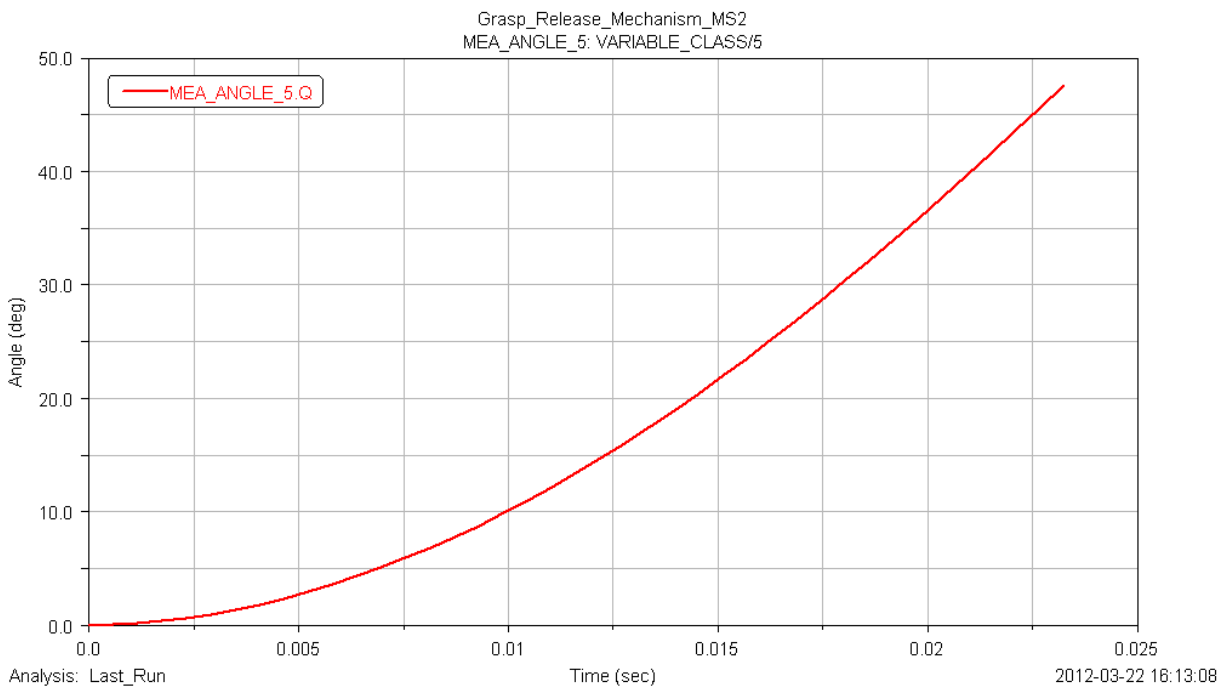


Figure 17: Dynamic Simulation of Grasp/Release Mechanism Using Pro/Engineer - Mechanism



**Figure 18: Right Jaw Angle of Rotation – ADAMS Simulation**



**Figure 19: Left Jaw Angle of Rotation – ADAMS Simulation**



## 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.

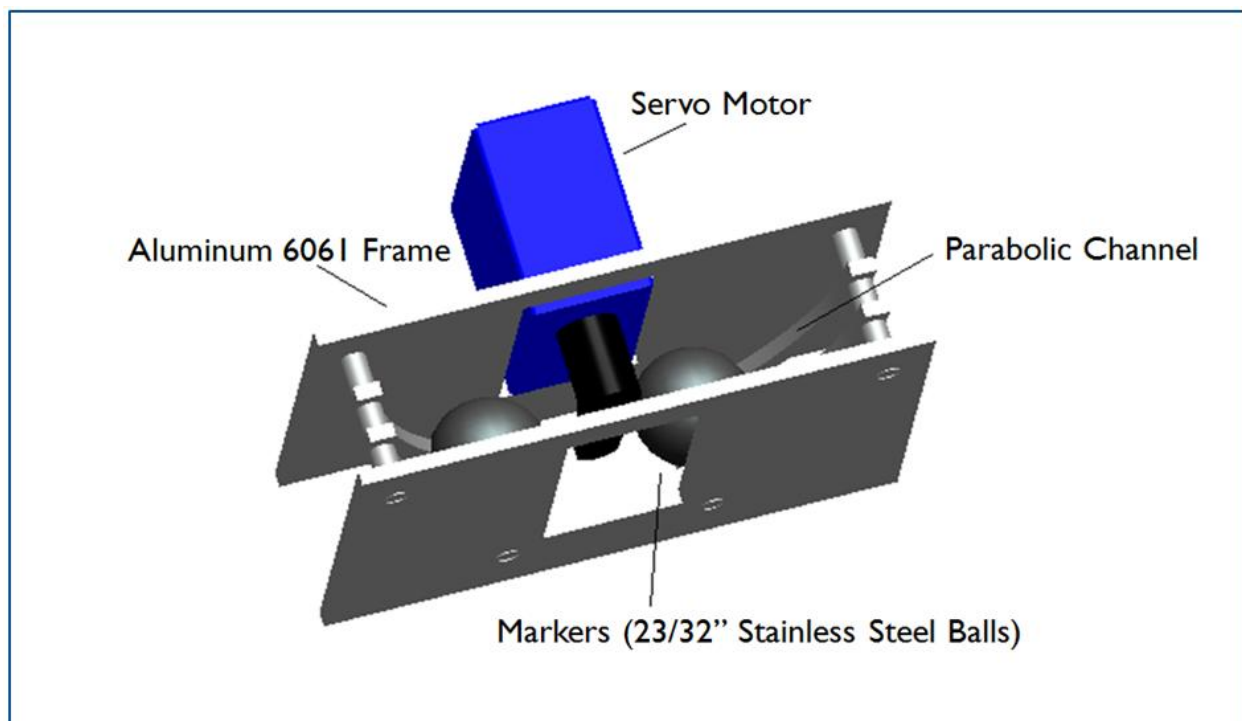


Figure 20: Close-Up View of Marker Dropper

## *Torpedo Launchers*

Similar to the grasp/release mechanism, the torpedo launchers 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 ABS Plastic 3D-printed torpedoes 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 cast acrylic guide cannons is placed on the horizontal neutral axis on opposite sides of the vehicle. The cannons have a cylindrical shape with an outside diameter of 1.25" and an inside diameter of 1.00" (i.e. 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 more accurate launch. The torpedo launchers are controlled by two independent solenoid valves, allowing the torpedoes to be fired individually.

Since the 3D printer could not 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 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 magnitude. 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 torpedo's 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 is able to 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 and preliminary tests, compressed air at 100 psi provides about 35 pounds of launch force, yielding an appropriate initial launch velocity for the torpedoes.

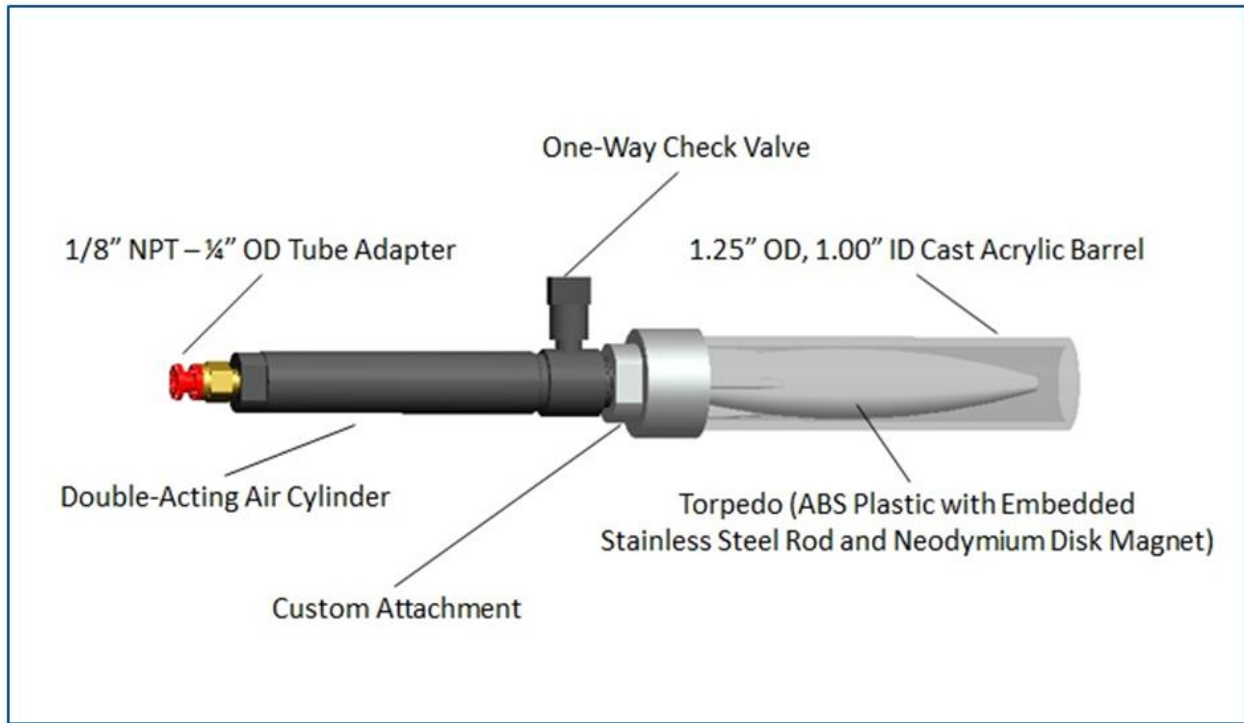


Figure 21: Close-Up View of Torpedo Launcher

### Electrical System

The AUV is powered by two identical Lithium-Ion batteries. Each is rated at 14.8 V (although their typical output is 16.5 V) and their net capacity is 20 Ah or 296 Wh. The maximum discharging rate of the batteries is 30 Amps. In addition, each battery requires ten hours to become fully charged.



Figure 22: Lithium-Ion Battery Pack Used on AUV

<b>Voltage</b>	Voltage: 14.8V (working) 16.8V (peak) 11.0V (cut-off)
<b>Capacity Prewired</b>	20 Ah or 296 Wh Charge / Discharge Terminal: 6.0" 14AWG Standard male Tamiya connector
<b>Max Discharging Rate</b>	30 Amp limited by PCM
<b>Dimensions (L x W x H)</b>	166mm (6.6") x 125mm (4.9") x 54mm (2.1")
<b>Weight</b>	4.0 lb (1810g)

The electrical components on the AUV are estimated to draw a combined average sustained current of  $\approx 4.1$  A. Therefore, applying the capacity specifications of the lithium-ion batteries, the AUV should be able to operate continuously for about 5 hrs. During the competition, the AUV is required to operate for less than twenty minutes. Therefore, these two batteries should be able to provide more than enough power for our AUV. Furthermore, their large capacity should serve well during the testing/debugging stage of the vehicle's development (mainly in May and June).

<b>Component</b>	<b>Maximum Current Consumption</b>	<b>Typical Current Consumption</b>
<b>Motor Drivers</b>	$\approx 12$ A	$\approx 2.25$ A
<b>Zotac PC Board</b>	$\approx 3.5$ A	$\approx 1.25$ A
<b>Arduino Board 1 – Thrusters</b>	$\approx 0.75$ A	$\approx 0.40$ A
<b>Arduino Board – Mechanical Subsystems, Pressure Transducer, and Hydrophones</b>	$\approx 1.0$ A	$\approx 0.20$ A
<b>IMU</b>	$\approx 75$ mA	$\approx 60$ mA
<b>TOTAL</b>	$\approx 17.3$ A	$\approx 4.1$ A

**Table 6: Current Consumption of Electrical Devices**

The AUV has various electrical components which require different DC input voltages and current supplies. Therefore, two customized voltage regulation boards were designed and simulated using the WEBENCH® Power Architect from TI to receive input power from the two lithium-ion batteries in series and regulate the output voltage to levels suitable for each component. Switching regulators are used in order to yield highly efficient (i.e.  $\approx 97\%$ ) voltage regulator circuits that contribute relatively insignificantly to the heat generation inside the enclosed hull. A schematic of the electrical system is illustrated below:

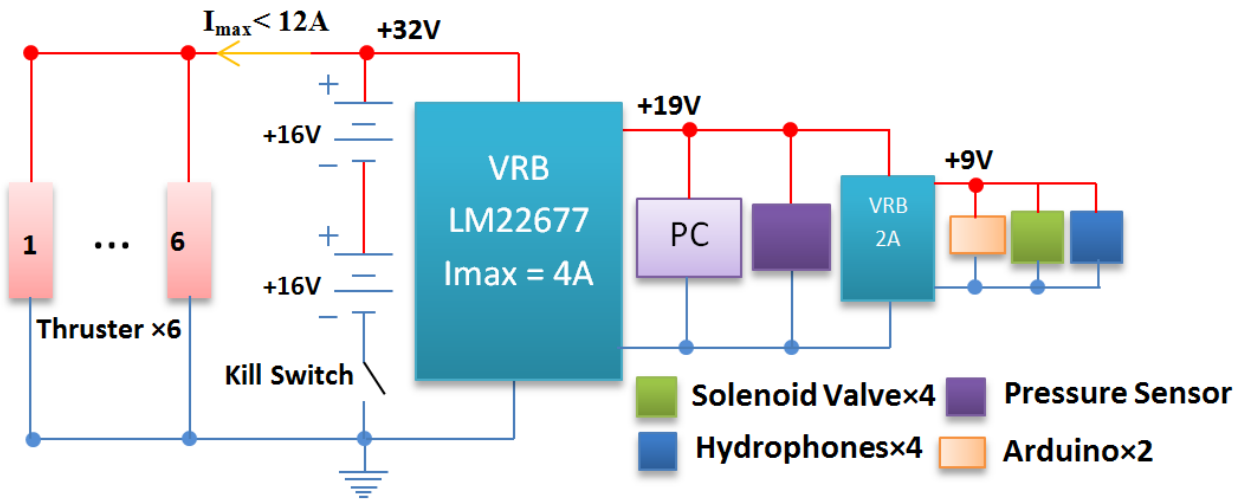


Figure 23: Electrical Circuit Diagram Overview for the AUV

### Voltage Regulator Boards

The AUV contains various electrical components which require different DC input voltages and currents. Therefore, two customized voltage regulation boards were designed to take in the power from the batteries (DC) and regulate correct DC output voltage for each component. For instance, the LM22677 voltage regulator board provides a steady 19 VDC to the Zotac PC and pressure transducer. Furthermore, the LM22676 regulator, which uses the output of the 19 V regulator board as its input in order to yield a more efficient and direct electrical circuit and physical wiring scheme, provides 9 V to the two Arduino Uno boards. The regulator boards were designed and simulated using the WEBENCH® Power Architect from TI, and the regulator circuit diagrams are shown below:

Voltage Regulator Board Summary	
Total System Efficiency	96.715 %
Total System Cost	\$48
Total System Power Dissipation	< 4W

Table 7: Voltage Regulator Board Specifications

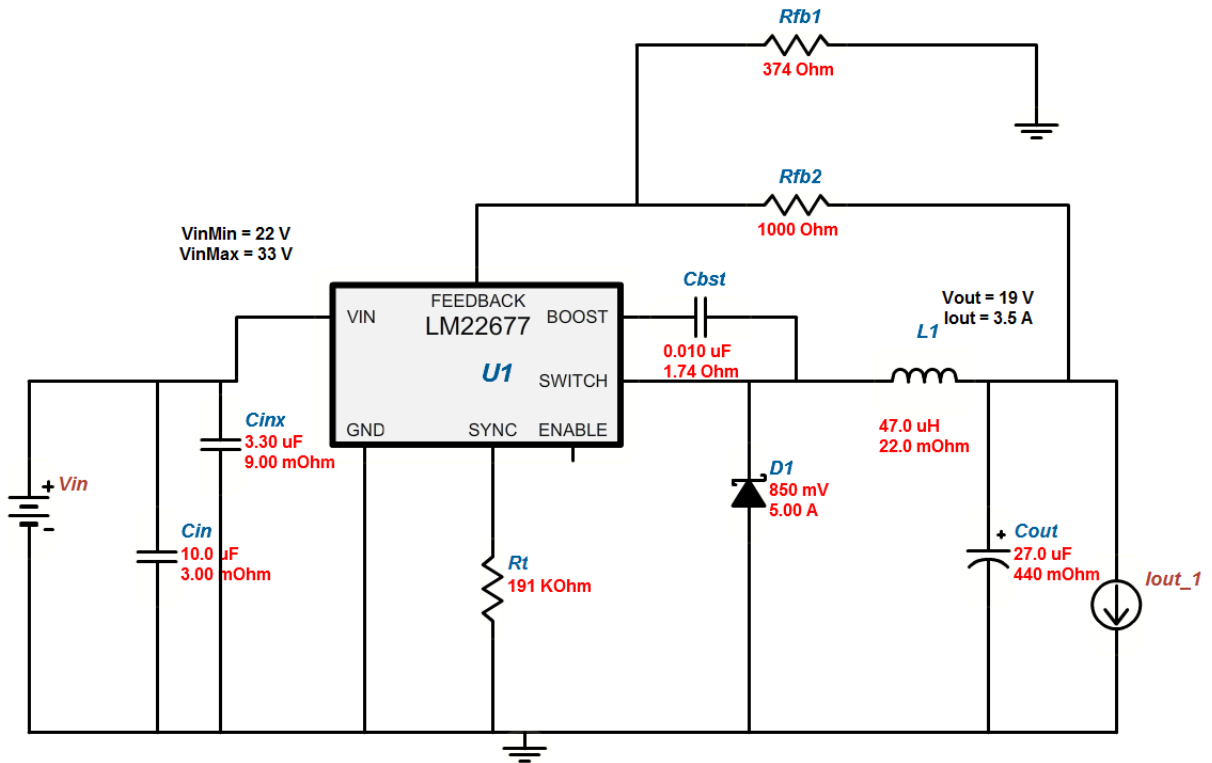


Figure 24: Circuit diagram of LM22677 Voltage Regulator for the Zotac PC and IMCL Submersible Pressure Transducer

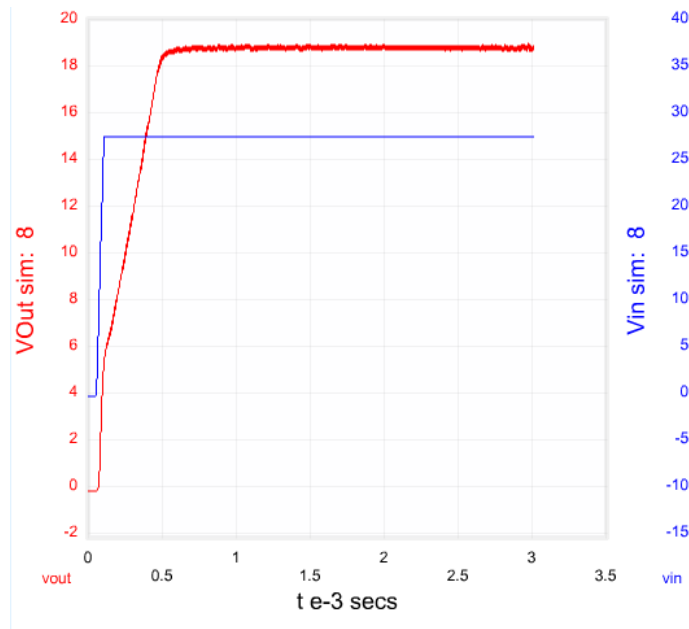


Figure 25: Startup Simulation of LM22677 Voltage Regulator for Zotac PC and IMCL Submersible Pressure Transducer

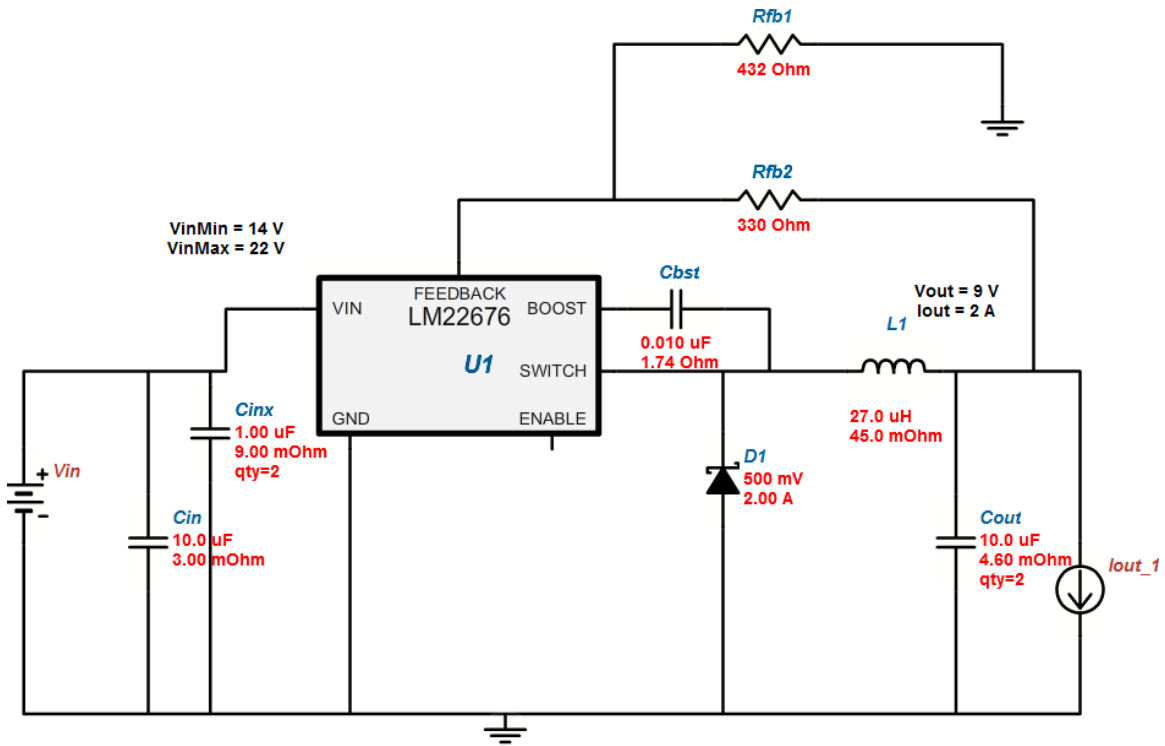


Figure 26: Circuit Diagram of LM22676 Voltage Regulator for the Arduino Uno Boards and Hydrophone Interface Circuit

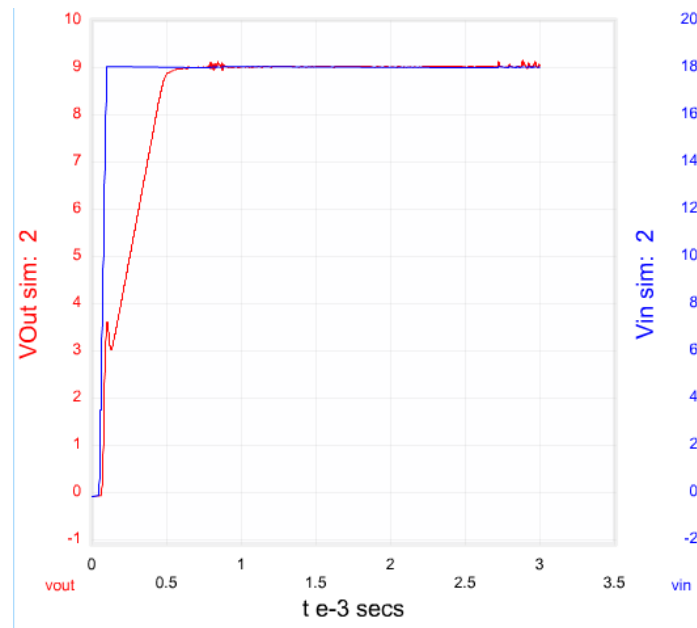


Figure 27: Startup Simulation of LM22676 Voltage Regulator for the Arduino Uno Boards

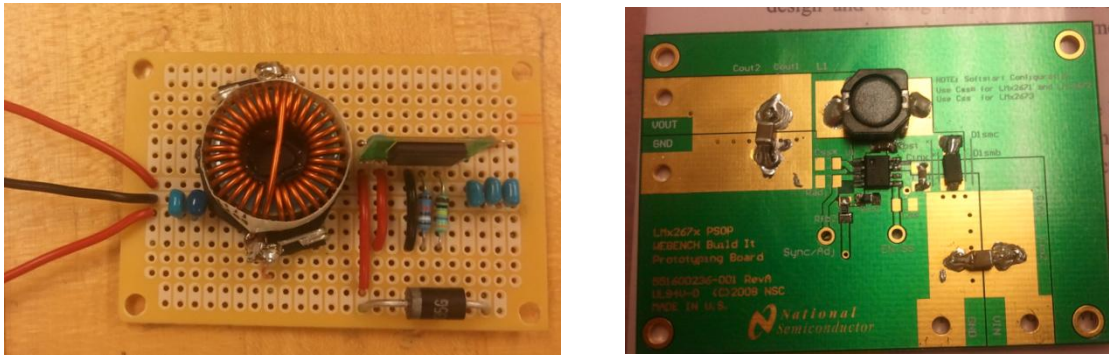


Figure 28: Photos of the LM22677 VRB (Left) and LM22676 VRB (Right)

## Software System

### *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 Ubuntu Linux operating system running 2.6.35 kernel. The Ubuntu Linux is one of the most popular open source Linux distributions in embedded systems and robotic control. The operating system can provide an easy to use platform for our software development. Text editors such as VIM could be used so that we can develop programs directly on the AUV. To automatically start all our programs once the vehicle is powered on, or the kill switch is turned on, we could just simply turn off password login and copy the compiled source code to the `/etc/xdg/autostart` Folder. When the AUV boots up, all the programs will run automatically.

### *Software Interface and Design*

All our programs will be primarily written in C and C++. The OpenCV will be written in C++. And the rest will be in C. The AUV will have a master program which is the mission controller. This is the first program that the AUV will run once it is 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, and the IMU module. The two cameras and IMU will directly communicate with the mission control program through USB interface. 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 mostly unidirectional, our software communication interface between the mission control process and any of the sensor module will be the unidirectional pipe. The pipe IPC is easier to implement compared to other methods. The graph illustration is shown below.



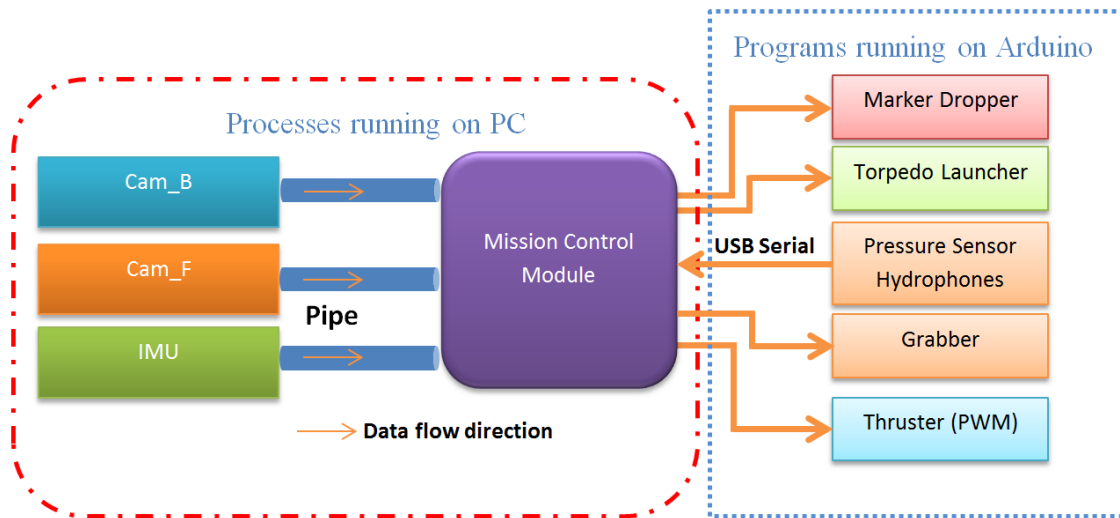


Figure 29: Graphical illustration of the Software Communication Interface and Structure

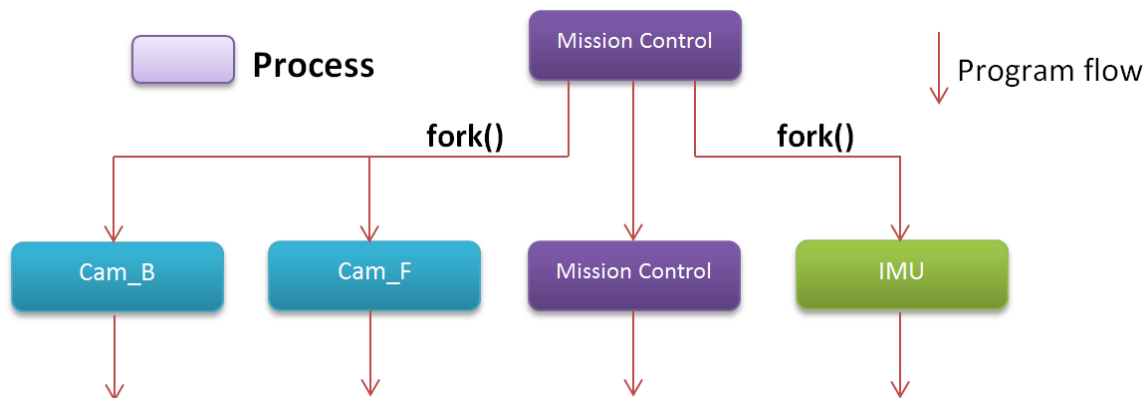


Figure 30: Mission control program fork illustration

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 Zotac PC 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 Zotac PC and the Arduino board uses Linux USB serial communication. The Arduino board will appear as a device named `"/dev/ttyACM0"` in the Ubuntu operating system. 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 programs are attached in the Appendix.

## *Mission Control*

### *Introduction*

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.

### *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.

- I. Initialization
  1. Fork processes for different modules
  2. Test communications between controller and modules
  
- II. Get input from pressure sensor and adjust the vehicle to the desired depth
  1. about 3.5ft deep(3-4 feet range)
  2. Balance the vehicle

### III. Detect Gate Position

1. Needs to adjust the vehicle so that it is perpendicular to the gate
2. Algorithm:
  - (a) 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
  - (b) Needs to measure 5 points. Four corners and one middle point of the rectangle.
  - (c) Calculate the length of the top bar (distance between top 2 points in image, blue line below) and the vertical segment (yellow line).
  - (d) 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  $5/3$ .
  - (e) 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.
  - (f) Check the angle again to make sure the vehicle is perpendicular to the gate.
  - (g) Once the vehicle is appropriately adjusted, move the vehicle forward and pass through the gate.
  - (h) Graph Illustration:

### IV. Feedback of Gate Passing

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.

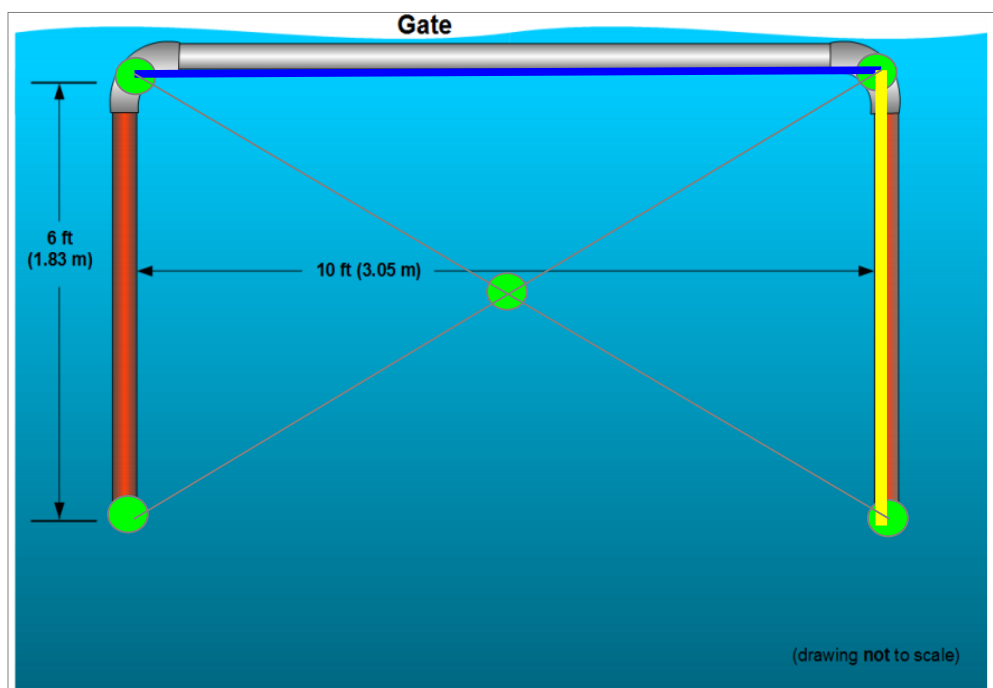


Figure 31: Graphical Illustration of the Gate Passing Algorithm

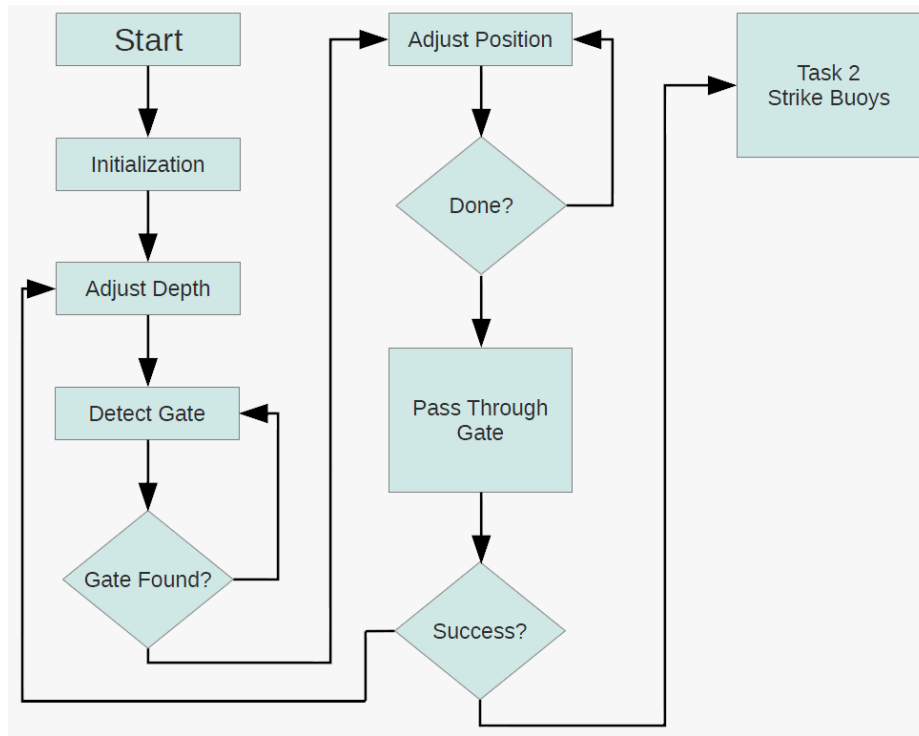


Figure 32: Gate Passing Flow Chart

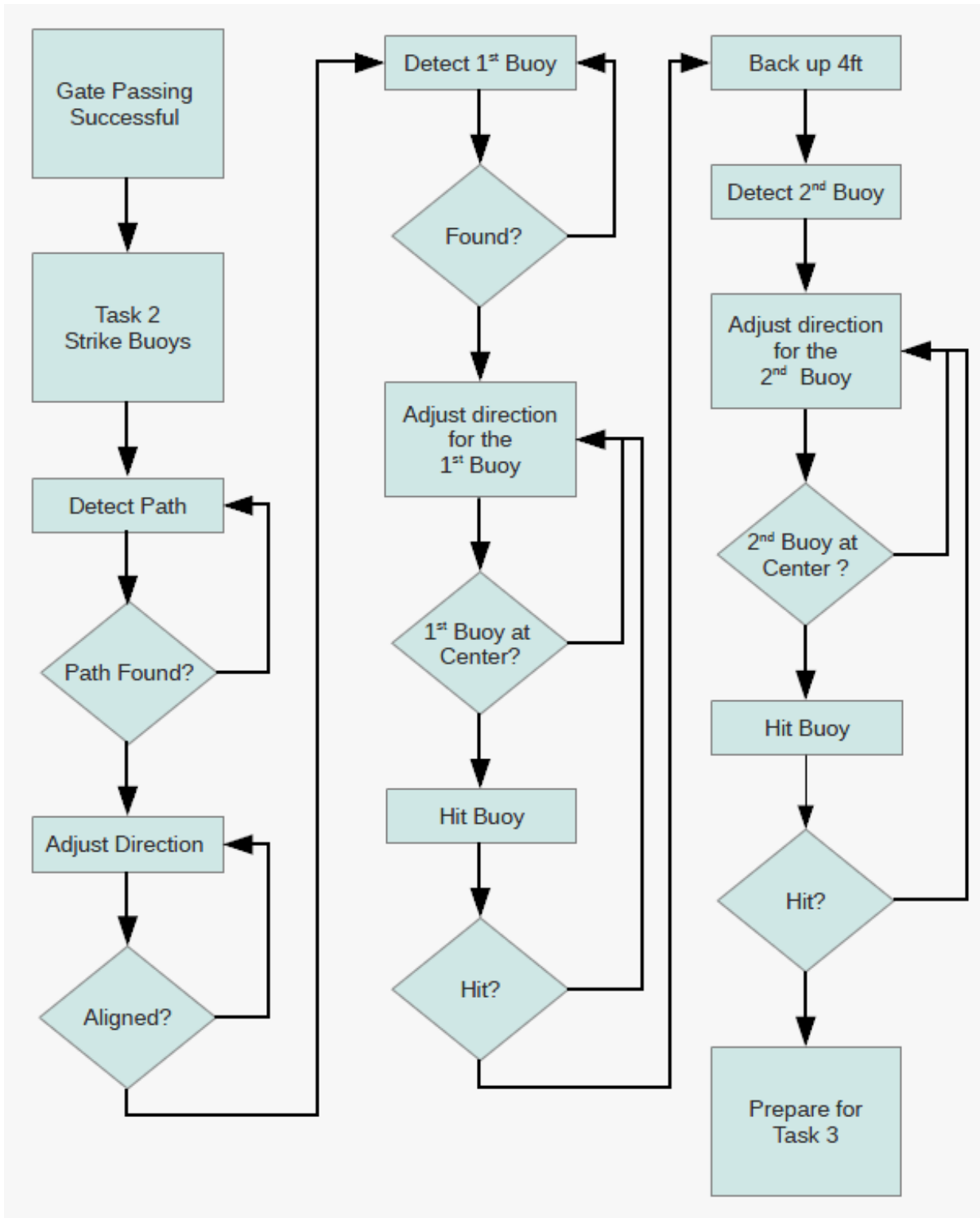


Figure 33: Gladiator Ring Task Flow Chart

### 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.

Furthermore, a complete guidance system does not only contain information about the inertial frame of reference. Information about depth (obtained via a pressure sensor) and velocity (obtainable via the IMU) also contribute to a complete system. For the purposes of the AUVSI Robosub competition, an additional component must be added to obtain this complete IGS: an acoustic location detection system. This is realized through the use of a 2 x 2 hydrophone array – three for the attitude of a pinger and one which serves as a reference. The conception, design, and implementation of this system are discussed below.

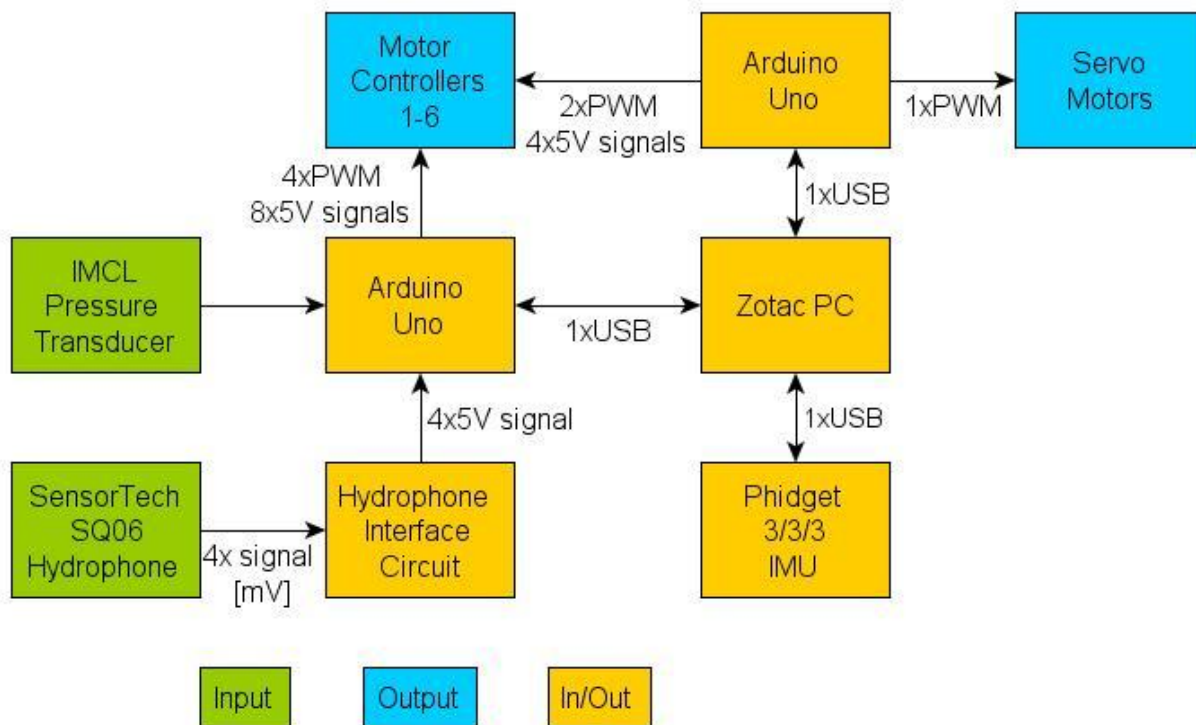


Figure 34: Block Diagram of the Guidance System

Evidently, the design constraints for a guidance system are system specific. For this competition, the Robosub had to be capable of moving autonomously which requires that it know about its environment and its vehicular heading.

### Computer Vision

The computer vision algorithms for gate detection and path detection have successfully been developed via the implementation of OpenCV, the Zotac PC board, and the Logitech C615 Web Cameras. These programs have been successfully tested in an underwater environment with the cameras installed inside their respective watertight

enclosure—the optical clarity has provided for quality image processing (i.e. 7 – 8 frames per second) and characterization accuracy. Integration of these algorithms with the guidance system still needs to be achieved prior to the July completion.

### **Inertial Measurement Unit**

Selecting an IMU was trivial: the Phidgets Spatial 3/3/3 (3-axis {gyroscope, magnetometer, accelerometer}) was reused from last year’s design. This IMU features high quality sensors that can measure both static and dynamic acceleration on three axes, magnetic field on three axes, and also angular rotation on three axes, totally 9 degrees of freedom. As most IMUs used on land-bearing projects have only 6 degrees of freedom, the additional 3 degrees allows for added flexibility when measuring an accurate heading underwater.



**Figure 35: Phidget-Spatial 3/3/3 Inertial Measurement Unit**

Programming the IMU is an interesting endeavor due to the relatively short-term inaccuracies of the accelerometer and the relatively long-term inaccuracies of the gyroscope. By combining the two data inputs using a Kalman filter, a more accurate overall measurement can be determined. The heavily commented code is listed in the Appendix A – IMU Code.

### **Hydrophone Interface Circuit (Proposed Alternate Design)**

The hydrophone interface circuit was subject to many constraints including time, size, accuracy, and noise-reduction capabilities. Because a majority of the hydrophone design was not completed until mid-way through the project, the components needed to fit in the remaining space on the electronics platform.

The *Laurel Wreath and Emperor’s Palace* task of the competition contains a pinger that emits a 2-second square-wave at a frequency between 22 -30 kHz. The exact pinger frequency is not revealed until the competition begins so a simple software adjustment is a must for this task. Furthermore, this design must be sensitive because acutely locating the pinger requires the augmentation of information from the hydrophones. Given these parameters, the following circuit was designed:

- 4 x National Instrument’s LMF100 high performance dual switched capacitor filter

- 1 x National Instrument's LM837 low-noise quad operational amplifier
- 1 x CM402 65ns low voltage comparator

These elements were connected with various voltages lines as shown in the figure below. The LMF100 works in four modes: band-pass, high-pass, low-pass, and notch modes. As each output filter is a 2<sup>nd</sup>-order filter, a 4<sup>th</sup>-order band-pass filter with an adjustable center frequency was attained by staggering the output from the first band-pass filter into the second bandpass filter. The output from the second band-pass filter was then sent to the op-amp for amplification. The bandwidth of the filter was fixed at 1 kHz with a minimum quality factor of 22, using the equation 1 below:

$$Q = \frac{f_m}{f_2 - f_1} = \frac{22\text{kHz}}{1\text{kHz}} = 22$$

**Equation 1: Quality Factor Determination for the Hydrophones Interface Circuit Design (Part 1)**

Initially, designing this filter proved extremely difficult due to the lack of experience with filter design. However, a combination of using the following formulas from the Band-pass Filter Design section of the "Op Amps for Everyone" and excerpts from the APLS Semester Project at Duke University was immensely helpful:

$$\alpha^2 + \left[ \frac{\alpha(\Delta\Omega)a_1}{b_1(1 + \alpha^2)} \right] + \frac{1}{\alpha^2} - 2 - \frac{(\Delta\Omega)^2}{b_1} = 0$$

$$\Delta\Omega = \frac{1}{Q} = 0.45$$

$$a_1 = \sqrt{2}$$

$$b_1 = 1.000$$

**Equation 2: Quality Factor Determination for the Hydrophones Interface Circuit Design (Part 1)**

As mentioned, two band-pass filters are staggered to create a fourth-order filter. For this to work, the two second-order filters need to have a center frequency slightly above and slightly below the required center frequency of the entire filter. In this case,  $\alpha$  represents the factor applied to the required midpoint frequency.

Successive Newtonian approximations resulted in an evaluation of  $\alpha \approx 1.0162$ . Using this value, the expected quality factor of each filter and their associated mid-point gain is given as follows:

$$Q_i = Q \frac{(1 + \alpha^2)b_1}{a_1\alpha} = 28.288$$



$$A_{mi} = \frac{Q_i}{Q} \sqrt{\frac{A_m}{b_1}} = \frac{28.288}{22} \sqrt{\frac{1}{1.000}} = 1.286$$

**Equation 3: Quality Factor and Mid-Point Gain Determination for Proposed Hydrophones Interface Circuit Design**

Evidently, the chosen gain ( ) for this portion of the hydrophone circuit was 1 because a separate amplification portion was designed. These parameters given us a mid-point frequency of  $f_{m2} = \alpha f_m = 22.356\text{kHz}$  for the first band-pass filter and  $f_{m2} = \frac{f_m}{\alpha} = 21.649\text{kHz}$  for the second bandpass filter.

With the theory complete, the next step was to design the circuit. The LMF100 data manual came with many example circuit and resistor combinations to create the desired quality factor and center frequency. One particular mode of interest is mode 3 which can be used to create highpass, lowpass, and bandpass outputs. The center frequency is given by the following equation:

$$f_o = \frac{f_{clk}}{100} \times \sqrt{\frac{R_2}{R_4}}$$

**Equation 4: Center Frequency Determination for the Proposed Hydrophone Interface Circuit Design**

and the quality factor of the complex pole pair is  $Q = \sqrt{\frac{R_2}{R_4}} \times \frac{R_3}{R_2}$ . Finally, the bandpass gain at is

$H_{OBP} = -\frac{R_3}{R_1}$ . By carefully selecting the resistor ratios, the following values were attained for the circuit. Note that two circuits were constructed, one for each 2<sup>nd</sup>-order bandpass filter.

Resistor	Circuit 1 (kΩ)	Circuit 2 (kΩ)
R1	77.76	77.76
R2	3.6	3.49
R3	100	100
R4	3.49	3.6

Circuit Value	Circuit 1	Circuit 2
Q	28.212	28.212
f0	fclk * 0.010156	fclk * 0.0098460
Bandpass gain	1.286	1.286

**Table 8: Component Selection for Proposed Alternative Hydrophone Interface Circuit Design**

As expected, the f0 values for Circuit 1 and Circuit 2 are slightly above and below each other due to the previously mentioned staggering.

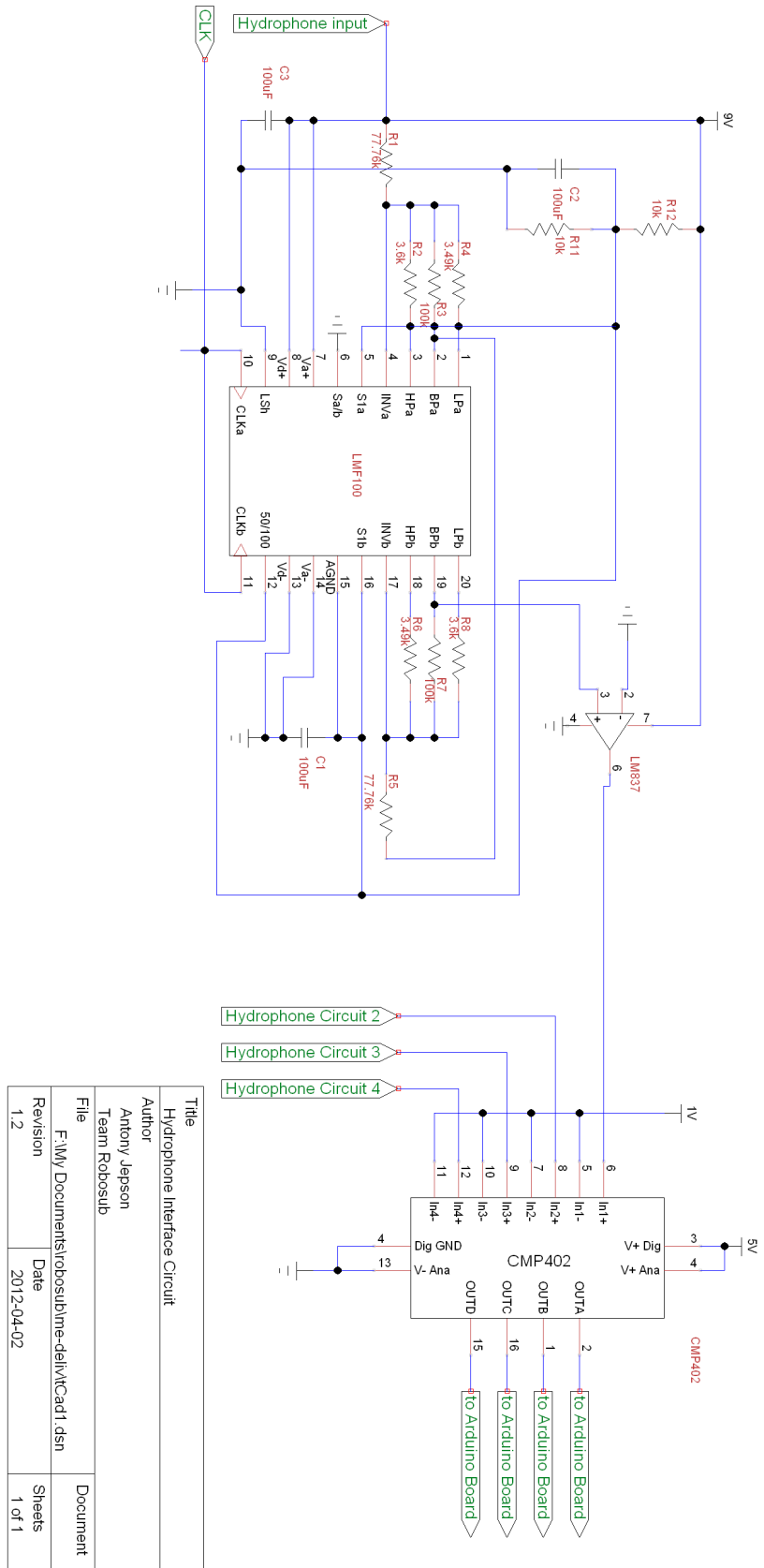


Figure 36: Proposed Hydrophone Interface Circuit Design Schematic

Table 9: Team Robosub 2011-2012 Project Budget

System Component	Price
Acrylic (Hull, Torpedo Cannons)	\$340.19
BTD150 Thrusters (x2)	\$1,005.18
Frame (80/20 Inch Solid Extruded Aluminum, Anchor Fasteners)	\$549.91
SEACON™ Wet-Mate Micro Connectors	\$1,374.47
Compressed Air Distribution System	\$396.26
SQ26-01 Hydrophones (x2)	\$410.75
Microcontrollers, Interface Circuit Components, Motor Drivers, Cables	\$852.96
Logitech C615 Web Cameras (x2)	\$108.49
IMCL Submersible Pressure Transducer	\$456.18
Zotac PC	\$293.98
Camera Enclosure Display Cases	\$52.50
Obstacle Course Materials	\$146.84
Traxxas 2056 Servo Motor	\$37.98
100% Silicone Sealant, Molykote™ Silicone Grease, Mineral Oil, Epoxy	\$107.13
Washers, Screws, Rods, Bolts, Etc.	\$103.01
Raw Materials (Aluminum, Acrylic, Etc.)	\$1,623.72
Laser Pointer	\$48.36
Competition Fee	\$500.00
<b>Total Expenditures for project</b>	<b>\$8,407.91</b>

A lot of progress was made on this project this year. With a lot of progress, come a lot of expenditures. The above table displays all of the costs that went into the project. Many discounts were provided along the way from various vendors to keep the overall expenses lower. One important expense was the acrylic tube used as the pressure vessel to house all of the electronics as well as the cannons for the torpedo launchers. A huge benefit gained from the previous year was the four SBT150 thrusters saving the team this year roughly two thousand dollars. The design this year did require two additional thrusters, so the BTD150 thrusters, which are nearly exact to the SBT150 thrusters, were chosen. The design process did begin early before the requirements and rules were post so the design needed to implement a frame that is versatile and could add components as necessary. 80/20 extruded aluminum was chosen specifically for these reasons.

Keeping the hull water tight and running the wires inside was another challenge so the SEACON Wet-Mate micro connectors worked perfectly for this yet the price reflects the importance of the product. Two of the peripheral subsystems were decided to use compressed air as the power source so nearly four hundred dollars was dedicated to the prototype/testing and final compressed air distribution system. Other salvaged parts from last year’s project include two hydrophones. This year’s design required an array of four, hence two additional needed to be purchased. In dealing with an autonomous vehicle the circuitry is very complex and requires many circuits working together. Many circuits were made in house to save money as well as purchasing kits and assembling in house. Others were purchased from vendors and required replacements due to electrical issues upon testing the vehicle. Several Arduino Uno boards as well as L298 motor drivers had to be replaced. The pressure transducer is used for determining depth and is a very specialized piece of equipment. The price reflected incorporates shipping from the United Kingdom as well. The cameras used for computer vision were found through Amazon to save money and still receive a high quality item that can operate effectively throughout the competition. The initial BeagleBoard-xM that was provided from the previous year turned out to not be sufficient for the tasks at hand so an upgrade in the form of a Zotac PC was purchased. The Zotac can handle the mission control with no problems so this was a requirement for the project. Raw materials became the largest purchase for this project due to the relevant sizes of material needed. All machining/fabricating/assembling was done in house to save money and time. Left over stocks will/have been donated to the machine shop for all of their contributions to the project. The final expenditure was a five hundred dollar entrance fee. This fee is fully refundable before May 1, 2012 and 50% refundable before June 1, 2012.

**Table 10: Sponsorship Funds**

<b>Provider</b>	<b>Amount</b>
<b>Harris</b>	\$3,000.00
<b>FAMU/FSU</b>	\$2,433.00
<b>Boeing</b>	\$2,000.00
<b>NEEC</b>	\$2,000.00
<b>ECE Department</b>	\$2,000.00
<b>Total</b>	<b>\$11,433.00</b>

## Results and Discussion

Team ROBOSUB  
FAMU-FSU College of Engineering

Form for Report of Test and Maintenance

### Title: Watertight Test – Hull

<b>Date (YYYY-MM-DD)</b>	2012-01-25	<b>Component</b>
<b>Time (HH:MM)</b>	17:30	<input checked="" type="checkbox"/> Hull / Frame
<b>Author</b>	Eric Sloan, Tra Hunter, Kahief Moody	<input checked="" type="checkbox"/> Interior Hull
<b>Location</b>	FSU Morcom Aquatics Center	<input type="checkbox"/> Electronics
<b>Test No</b>	1	<input type="checkbox"/> Electrical System
		<input type="checkbox"/> Mechanical Subsystem:
		<input type="checkbox"/>
		<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
		Test Result:

#### 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
- Person to retrieve AUV in the pool
- Hull (attached to AUV frame)
- 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 and supported by one of the team members. 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 at this point in the manufacturing stage, the vehicle was balanced (although the hull need to be rotated very slightly in order to completely optimize the naturally stable roll angle), and the hull proved to be completely water tight.

**Comments**

Multiple tests thereafter, including several with electrical components installed inside the hull, have continued to verify these findings of a completely watertight hull and balanced system.

<b>Team ROBOSUB</b> FAMU-FSU College of Engineering	<b>Form for Report of Test and Maintenance</b>
--	--

**Title: Watertight Test – Camera Enclosures**

<b>Date (YYYY-MM-DD)</b>	2012-01-26	<b>Component</b>
<b>Time (HH:MM)</b>	19:30	<input checked="" type="checkbox"/> Hull / Frame
<b>Author</b>	Eric Sloan, Tra Hunter, Kashief Moody, Ryan Kopinsky	<input type="checkbox"/> Interior Hull
<b>Location</b>	FSU Morcom Aquatics Center	<input type="checkbox"/> Electronics
<b>Test No</b>	1	<input type="checkbox"/> Electrical System
		<input type="checkbox"/> Mechanical Subsystem: Torpedo Launch

### **Objective**

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

### **Equipment**

- Pool
- Camera Enclosures
- Stop watch

### **Process**

The redesigned camera enclosures were completely sealed off (i.e. caulking was applied and allowed time to sufficiently set, and the removable end caps were firmly installed), brought to the FSU Morcom Aquatics Center and submerged for a time period of 5 – 10 minutes. The enclosures were then taken out of the water and assessed for signs of leakage

### **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 5 – 10 minute waiting period, the interior of both of the enclosures were completely dry—an indication that the Logitech C615 Web Cameras were ready for installation in their respective enclosure.

### **Comments**

The remodeled camera enclosure design and manufacture was executed properly. Several tests thereafter, including with the Logitech C615 Web Cameras securely and permanently installed, have continued to verify the results of this successful test.

**Title: Torpedo Performance Test**

<b>Date (YYYY-MM-DD)</b>	2012-01-30	<b>Component</b>
<b>Time (HH:MM)</b>	21:30	<input type="checkbox"/> Hull / Frame
<b>Author</b>	Eric Sloan	<input type="checkbox"/> Interior Hull
<b>Location</b>	Pool Outside Apartment	<input type="checkbox"/> Electronics
<b>Test No</b>	2	<input type="checkbox"/> Electrical System
		<input checked="" type="checkbox"/> Mechanical Subsystem: Torpedo Launcher
		Test Result: <input checked="" type="checkbox"/> Pass <input type="checkbox"/> 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.

<b>Team ROBOSUB</b> FAMU-FSU College of Engineering	<b>Form for Report of Test and Maintenance</b>
--	--

## Title: Marker Dropper Test – Air

<b>Date (YYYY-MM-DD)</b>	2012-12-07	<b>Component</b>
<b>Time (HH:MM)</b>	14:30	<input type="checkbox"/> Hull / Frame
<b>Author</b>	Eric Sloan	<input type="checkbox"/> Interior Hull
<b>Location</b>	Mechatronics Lab, FAMU-FSU College of Engineering, Room B324	<input type="checkbox"/> Electronics
<b>Test No</b>	2	<input type="checkbox"/> Electrical System
		<input checked="" type="checkbox"/> Mechanical Subsystem: Marker Dropper
		Test Result: <input checked="" type="checkbox"/> Pass <input type="checkbox"/> 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 ( $\approx 8V$ )
- 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 program worked successfully, and the servo motor rotated to the calibrated release positions, thus allowing the steel balls to drop from the device upon command.

## **Comments**

The servo motor was also waterproofed thereafter, and verified to still function properly. Integration with the Arduino Board, however, has yet to be done. This should be a seamless transition, though.

## Title: Torpedo Launchers Test – Air

<b>Date (YYYY-MM-DD)</b>	2012-01-31	<b>Component</b>
<b>Time (HH:MM)</b>	17:30	<input type="checkbox"/> Hull / Frame
<b>Author</b>	Eric Sloan, Tra Hunter	<input type="checkbox"/> Interior Hull
<b>Location</b>	Senior Design Lab	<input type="checkbox"/> Electronics
<b>Test No</b>	1	<input type="checkbox"/> Electrical System
		<input checked="" type="checkbox"/> Mechanical Subsystem: Torpedo Launcher
<b>Test Result:</b> <input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail		

### 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.

<b>Team ROBOSUB</b> FAMU-FSU College of Engineering	<b>Form for Report of Test and Maintenance</b>
--	--

## Title: Torpedo Launcher Test – Water

<b>Date (YYYY-MM-DD)</b>	2012-03-30	<b>Component</b>
<b>Time (HH:MM)</b>	11:30	<input type="checkbox"/> Hull / Frame
<b>Author</b>	Eric Sloan, Tra Hunter, Ryan Kopinsky	<input type="checkbox"/> Interior Hull
<b>Location</b>	FSU Morcom Aquatics Center	<input type="checkbox"/> Electronics
<b>Test No</b>	1	<input type="checkbox"/> Electrical System
		<input checked="" type="checkbox"/> Mechanical Subsystem: Torpedo Launchers
		Test Result: <input type="checkbox"/> Pass <input checked="" type="checkbox"/> Fail

## **Objective**

Test the torpedo launchers underwater via the integration of the compressed air distribution system (includes implementation of the Arduino Board and solenoid valve interface circuit)

## **Equipment**

- Solenoid Valve Interface Circuit
- Compressed Air Tank
- Tank Regulator
- Pressure Gauge(s)
- Secondary/Low Pressure Regulator
- Solenoid Valves
- Nylon Gas Lines + Adapters + Check Valves
- Arduino Board
- Voltage Regulator Board
- Lithium-Ion Batteries
- Kill Switch
- Torpedo Launchers
- AUV

## **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 (by 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. A code was uploaded to the Arduino Uno Board responsible for actuating the mechanical subsystems (including the grasp/release mechanism). The AUV was transported to the FSU Morcom Aquatics Center, placed in the water, and the kill switch was triggered, thus closing the electrical circuit and initiating the program. The program sequentially actuated each of the four solenoid valves, thus activating both of the torpedo launchers as well as the grasp/release mechanism.

## **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 and direct flight underwater for several feet prior to coming to a stop.

**Success Criteria**

The torpedoes will be individually fired at a safe, yet sufficiently high speed, and in a straight path for a reasonable distance.

**Actual Results**

Although the solenoid valves that control the flow of compressed air to the torpedo launcher air cylinders were properly actuated, high pressure build-up due to incurred viscous drag between the torpedoes and the acrylic guide cannons resulted in a failure to launch.

**Comments**

A simple, yet effective, solution to this issue has been derived—a pattern of holes will be drilled into the cast acrylic guide barrels (particularly toward the base) in order to enable water to better flow around the torpedoes during launch. This adjustment will be made shortly and should yield a significant reduction in both friction and pressure drag, resulting in successful future operation.

<b>Team ROBOSUB</b> FAMU-FSU College of Engineering	<b>Form for Report of Test and Maintenance</b>
--	--

**Title: Grasp/Release Mechanism Test – Water**

<b>Date (YYYY-MM-DD)</b>	2012-03-30	<b>Component</b>
<b>Time (HH:MM)</b>	11:30	<input type="checkbox"/> Hull / Frame
<b>Author</b>	Eric Sloan, Tra Hunter, Ryan Kopinsky	<input type="checkbox"/> Interior Hull
<b>Location</b>	FSU Morcom Aquatics Center	<input type="checkbox"/> Electronics
<b>Test No</b>	1	<input type="checkbox"/> Electrical System
		<input checked="" type="checkbox"/> Mechanical Subsystem: Grasp/Release Mechanism
		Test Result: <input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail

## Objective

Test the grasp/release mechanism underwater via the integration of the compressed air distribution system (includes implementation of the Arduino Board and solenoid valve interface circuit)

## Equipment

- Solenoid Valve Interface Circuit
- Compressed Air Tank
- Tank Regulator
- Pressure Gauge(s)
- Secondary/Low Pressure Regulator
- Solenoid Valves
- Nylon Gas Lines + Adapters + Check Valves
- Arduino Board
- Voltage Regulator Board
- Lithium-Ion Batteries
- Kill Switch
- Grasp/Release Mechanism
- AUV

## 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 (by 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. A code was uploaded to the Arduino Uno Board responsible for actuating the mechanical subsystems (including the grasp/release mechanism). The AUV was transported to the FSU Morcom Aquatics Center, placed in the water, and the kill switch was triggered, thus closing the electrical circuit and initiating the program. The program sequentially actuated each of the four solenoid valves, thus activating both of the torpedo launchers as well as the grasp/release mechanism. The grasp/release mechanism behaved exactly as designed and simulated, closing to the projected position at a fast, yet controlled speed, and opening properly once the compressed air was purged from the system.

### **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.



## Environment, Health, and Safety

When dealing with a vehicle that is designed to operate underwater, environmental, health, and safety issues have to be taken into account. In the sealing of the hull and camera enclosures, only non-toxic silicone was used to provide reassurance for environmental safety issues. For the propulsion system of the vehicle, each thruster has a safety shroud as required in the competition rules for safe operation around divers. This is a major concern for the competition since divers will accompany the sub in the pool. For the *Et Tu Brute* (kill Caesar) part of the competition, the torpedoes had to be fired in air first to establish that the velocity in water would not be enough to cause injury to the divers. The rules state that the torpedo must not fire at a rate that would cause a bruise to a human. The main safety feature stressed by the competition board is the presence of an operational kill switch. The kill switch in this design not only shuts off the thrusters, but the entire vehicle. The strategic placement of the kill switch on the vehicle makes it easy to access during a predicament.

## Conclusions

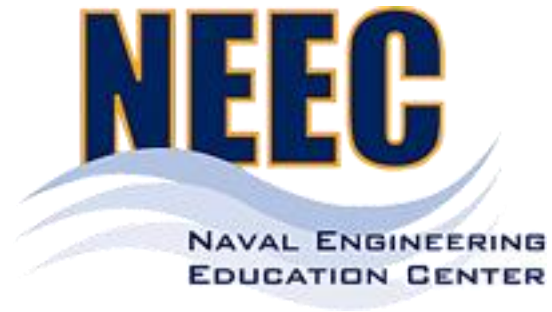
The AUV is mechanically complete (aside from a couple minor additions—aluminum plates to attach to the front of the vehicle along the sides in order to obtain optimal density and achieve a natural pitch angle of zero, thermally-conductive heat dissipation plates which will screw into the heat sinks on the motor drivers and directly contact the electronics platform, and optimization of the cast acrylic torpedo barrels by drilling a pattern of holes in order to reduce the viscous and pressure drag on the torpedoes during launch—all of which will be completed in the upcoming days). Furthermore, the vehicle is nearing electrical completion with the exception of the hydrophones interface circuit.

## Acknowledgements

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

Dr. Michael Frank, Michael Greenleaf, Dr. Bruce Harvey, Dr. Zohrob Hovsopian, Dr. Srinivas Kosaraju, Dr. Mattheiu Dalban-Canassy, Dr. Chiang Shih, Dr. Uwe Meyer-Baese, Dr. Ming Yu

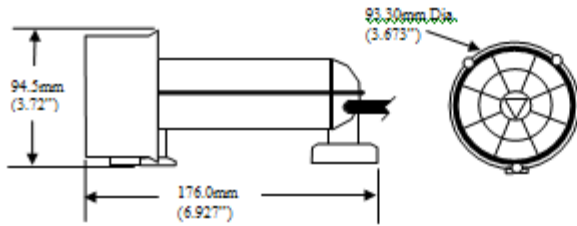
Further gratitude goes out to our sponsors for their generous financial contributions:



# Appendix A - Data Sheets



## DATA SHEET BTD150 BRUSHED DC THRUSTER



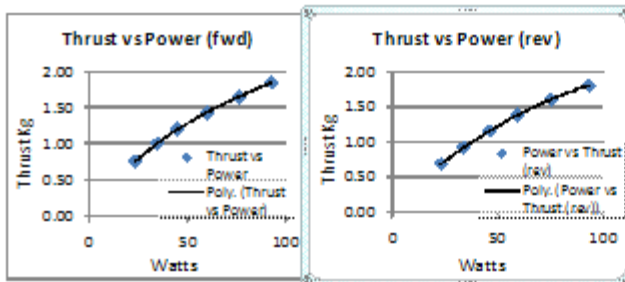
### SPECIFICATIONS:

VOLTAGE NOMINAL..... 12VDC  
 MAX CURRENT  
 OF MOTOR WINDINGS..... \*4.0AMPS  
 (a current regulator is highly recommended)

WEIGHT:  
 Dry..... 718grams  
 Wet..... 413grams (in fresh water)

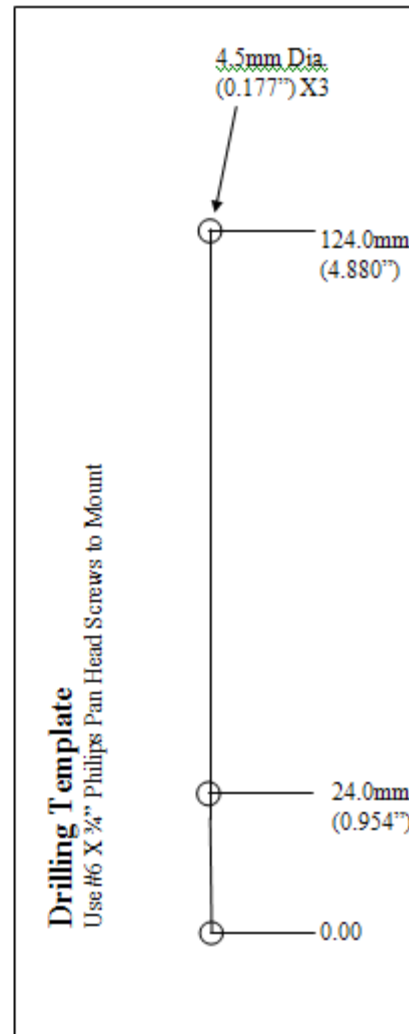
WIRING CONFIGURATION:  
 Red..... Positive VDC  
 Black..... Negative VDC

SEABOTIX BTD150 DATA



Direction	Voltage (DC)	Current Draw (A)	Thrust (kg)	Power (W)
Forward	12.0	1.91	0.75	22.92
	14.0	2.45	1.00	34.3
	16.0	2.78	1.20	44.48
	18.0	3.32	1.43	59.76
	20.0	3.78	1.65	75.6
Reverse	22.0	4.20	1.85	92.4
	12.0	1.92	0.68	23.04
	14.0	2.42	0.92	33.88
	16.0	2.89	1.15	46.24
	18.0	3.30	1.38	59.4
	20.0	3.75	1.60	75
	22.0	4.23	1.80	93.06

\* Higher currents (up to 6A) may be used, but will drastically shorten the life of the thruster if ran continuously in this mode. High current bursts of 1 minute or less can be performed as long as the running average current is kept at 4.0 Amps MAX to prevent the motor windings from building up excessive heat



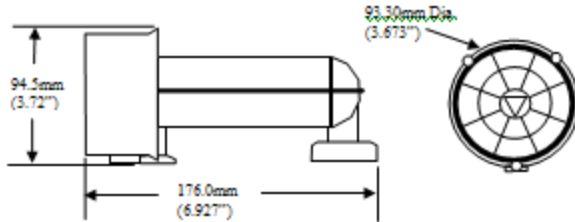
### Drilling Template

Use #6 X 3/4" Philips Pan Head Screws to Mount

SEABOTIX, 2877 Historic Decatur Road, Suite 100, San Diego, CA, 92106, USA (619) 450-4000  
 Specifications Subject to Change without Prior Notice 30 July 2010



## DATA SHEET SBT150 SPARE BRUSHED THRUSTER



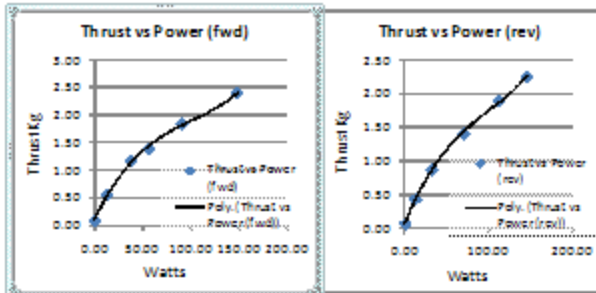
### SPECIFICATIONS:

VOLTAGE NOMINAL ..... 28VDC  
 MAX CURRENT  
 OF MOTOR WINDINGS ..... \*5.0AMPS  
 (\* current regulator is highly recommended)

WEIGHT:  
 Dry ..... 686grams  
 Wet ..... 393grams (in fresh water)

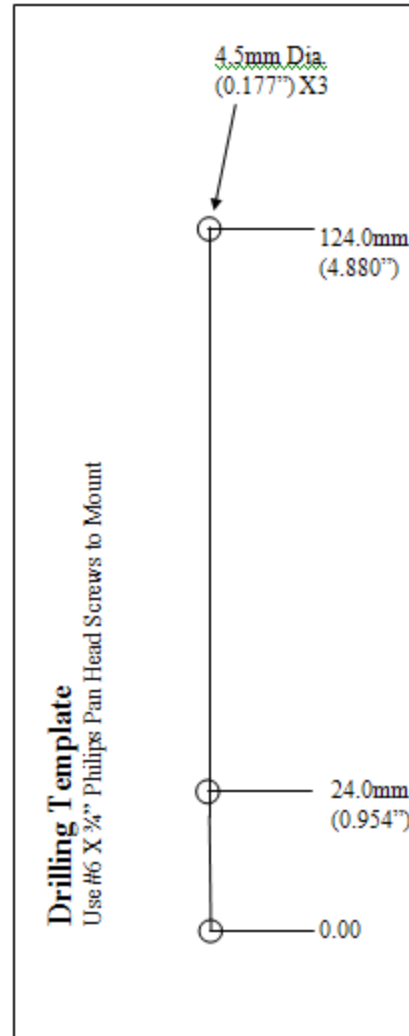
WIRING CONFIGURATION:  
 White ..... Clock  
 Black ..... Ground  
 Green/Yel. .... Data  
 Red ..... 28VDC

SEABOTIX SBT150 DATA



Direction	Voltage(DC)	Current Draw (A)	Thrust (kg)	Power (W)
Forward	28	0.000	0.09	0.00
	28	0.420	0.55	11.80
	28	1.320	1.17	37.10
	28	2.000	1.38	56.20
	28	3.240	1.84	90.70
Reverse	28	0.000	0.07	0.00
	28	0.440	0.43	12.44
	28	1.120	0.87	31.42
	28	2.500	1.40	70.20
	28	4.000	1.89	111.90
	28	5.210	2.25	145.90

\* Higher currents (up to 6A) may be used, but will drastically shorten the life of the thruster if ran continuously in this mode. High current bursts of 1 minute or less can be performed as long as the running average current is kept at 4.0 Amps MAX to prevent the motor windings from building up excessive heat



**Drilling T-plate**  
 Use #6 X 3/4" Philips Pan Head Screws to Mount


**FEATURES**

- Low cost
- Rugged
- Good depth capability

**APPLICATIONS**

- General-purpose research
- Towed arrays

**OVERVIEW**

The SQ26 is a general-purpose low-cost hydrophone. It has good sensitivity, wide bandwidth and good stability. Custom configurations of these hydrophones are also available. For additional data on frequency response or outline drawings, please call our [technical support](#).

All parameters measured after hydrophones have been subjected to pressures of 200 bar. The polyurethane-encapsulated hydrophone will withstand continuous immersion in isoparaffinic hydrocarbon fluids and sea water.

**SPECIFICATIONS**

**Voltage sensitivity:**  $-193.5 \pm 1.0$  dBV re  $1 \mu\text{Pa}$  @  $20^\circ\text{C}$ , 20 V/bar

**Charge sensitivity:** 24 nC/bar

**Capacitance:**  $1.4 \text{ nF} \pm 10\%$  @  $20^\circ\text{C}$

**Capacitance variation with temperature:** 0.33% increase per  $^\circ\text{C}$

**Capacitance variation with depth:** 7% loss per 1,000 m (3,300 ft)

**Operating depth:** down to 2,000 m

**Frequency response:** flat from 1 Hz to 28,000 Hz

**Acceleration sensitivity:**  $< 0.2$  mbar/g when properly mounted

**Diameter:** 28.7 mm (1.13")

**Length:** 30.5 mm (1.2") max.

**Mass:** 16 grams (0.56 oz)

**Electrical leads:** 30cm (12") twisted pair with shielded polyurethane jacket

**Electrical insulation:**  $> 500$  M Ohms

**Water blocked leads:** Yes

**Notes:** Comes standard with integral shield in a low noise configuration



20 Stewart Rd, Collingwood, ON, Canada L9Y 4K1  
 Tel: (705) 444-1440 Fax: (705) 444-6787  
[www.sensortech.ca](http://www.sensortech.ca) [techsupport@sensortech.ca](mailto:techsupport@sensortech.ca)

[EMAIL](#) [HOME](#) [◀BACK](#) [NEXT▶](#)

## Appendix B - IMU Code

```
// Orientation code for IMU Kalman filter
// 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 square(x) ((x) * (x))
#define PI 3.14159265358979f
```

```
/* According to the product manual, if the spatial is placed perpendicular to
 * the ground on the listed side, 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}
*
* To adjust for a 90-degree clockwise rotation the following transformation
* matrix will be applied to output.
*
* [x'] = [ 0 1][x]
* [y'] = [-1 0][y]
*
* TODO
* - determine velocity (new file)
* - determine heading using kalman filter
*/

void normalize_1x3_vec(double vec[3]) {
    double r = sqrt(square(vec[0]) + square(vec[1]) + square(vec[2]));
    vec[0] /= r;
    vec[1] /= r;
    vec[2] /= r;
}

// Assumes that only one spatial is connected.
int CCONV SpatialDataHandler(CPhidgetSpatialHandle spatial, void *userptr,
CPhidgetSpatial_SpatialEventDataHandle *data, int count) {
    /**** SETUP ****/
    static const double accel_max = 5.0; // (g m/s)
    static const double accel_min = -5.0; // (g m/s)
    static const double ang_rate_max = 400; // (deg/s)
    static const double ang_rate_min = -400; // (deg/s)
    double acc[3]; // vector: Convenience array containing acceleration values.
    double acc_prev[3]; // vector: Contains previous acceleration values.
    double acc_norm[3]; // vector: Normalized acceleration values using acc[3] / Racc
    double acc_est[3]; // vector: Estimated acceleration values for Kalman filter
    double Axz, Ayz; // angles as discussed in notation
    double ang_rate[3]; // Convenience array containing gyroscope values.
    double ang_rate_est[3]; // Contains previous angular rate.
    double Racc; // Racc is the R acceleration vector: R^2 = (R0)^2 + (R1)^2 + (R2)^2
    double prev_acc_est[3]; // Previous estimated acceleration values for Kalman filter
        static double timestamp_us = 0.0; // Seconds since last Spatial event
    int firstSample = 1; // Used to prime Kalman filter.
    int ftmp = 0.0; // Temporary float value used for calculations.

    // Timestamp (s): Used for integration of readings.
    timestamp_us += data[0]->timestamp.microseconds * 1e-6f;

    // Accelerometer (g's): based on a left handed system with thumb pointing
    // down.
    acc[0] = data[0]->acceleration[0];
    acc[1] = data[0]->acceleration[1];
    acc[2] = data[0]->acceleration[2];

    // Gyroscope (deg/s): has a different coordinate system than the accelerometer and
    // needs to be adjusted.
    // 0: pitch (1, 2 plane)
    // 1: roll (0, 2 plane)
    // 2: yaw (0, 1, plane)
    ang_rate[0] = data[0]->angularRate[0];
    ang_rate[1] = data[0]->angularRate[1];

```

```

ang_rate[2] = data[0]->angularRate[2];

// Adjust gyroscope coordinate system to match accelerometer system assuming
// that anti-clockwise rotation is positive.
ang_rate[1] = -ang_rate[1];
ang_rate[0] = -ang_rate[0];

// Remove suspect values greater than min/max.
if ((acc[0] > accel_max) | acc[0] < accel_min) |
    ((acc[1] > accel_max) | acc[1] < accel_min) |
    ((acc[2] > accel_max) | acc[2] < accel_min))
    printf("Err: Acceleration is suspect (> 5g's or < -5g's)\n");
if ((ang_rate[0] > ang_rate_max) | ang_rate[0] < ang_rate_min) |
    ((ang_rate[1] > ang_rate_max) | ang_rate[1] < ang_rate_min) |
    ((ang_rate[2] > ang_rate_max) | ang_rate[2] < ang_rate_min))
    printf("Err: Angular Rate is suspect (> 400deg/s or < 400deg/s)\n");

/**** COMPUTATION *****/
// Normalize the acceleration vector
Racc = sqrt(square(acc[0]) + square(acc[1]) + square(acc[2]));
acc_norm[0] = acc[0] / Racc;
    acc_norm[1] = acc[1] / Racc;
    acc_norm[2] = acc[2] / Racc;

/*
Apply Kalman filter. This works by estimating the next value received by
the accelerometer and correcting the guess using gyroscope data and the
previous guess. The resultant estimate is used as the assumed position of
the AUV.

The Discrete Kalman Filter
Eq.:  $x(k) = F(k) x(k - 1) + B(k) u(k) + w(k)$ 
-  $F(k)$ : ( $[n \times n]$  matrix) the "state-transition model". Relates current
  step with previous step in absense of a driving function or noise.
-  $B(k)$ : ( $[n \times 1]$  matrix) control input to state  $x(k)$ .
-  $x(k)$ : ( $[n]$  vector) the current state.
-  $u(k)$ : ( $[1]$  vector) known input to the system.
-  $w(k)$ : process noise

Eq.:  $z(k) = H(k) x(k) + v(k)$ 
-  $z(k)$ : ( $[m]$  vector) measurement
-  $H(k)$ : ( $[m \times n]$  matrix) relates the state to the measurement  $z(k)$ .
-  $x(k)$ : ( $[n]$  vector) the current state.
-  $v(k)$ : measurement noise

 $w(k)$  and  $v(k)$  are assumed to be independent of each other, white, and with
normal propability distributions:

Eq.:  $p(w) \sim N(0, Q)$ ,  $Q$  = process noise covariance
Eq.:  $p(v) \sim N(0, R)$ ,  $R$  = measurement noise covariance

These covariances are assumed to be constant as well as matrix  $H$ .

Application
The state of the Kalman filter is represented by the  $x(k)|k$ , estimate of
 $x(k)$  @ time  $k$  and  $P(k)$ , the error covariance matrix.

Applying the filter consists of two stages: (1) predict the next state and
the estimated covariance and (2) updating the state estimate and covariance
estimate.

```

```

Notation
R: acceleration vector
Ra: a component of acceleration vector
Rab: Proj of R vector on a-b plane.
Aab: Angle on a-b plane between the b-axis and the Rab vector.
*/

/* Stage 1: Predict the next state.
Axz(k)          = atan2(Rx,Rz)
Axz(k-1)        = atan2(RxEst(k-1), RzEst(k-1))
RateAxz(k)      = ang_rate[1];
RateAxz(k-1)    = prev_ang_rate[1];
RateAxzAvg      = RateAxz(
Axz(k)          = Axz(k-1) + RateAxzAvg * dT
*/

// Accept initial accelerometer readings.
if (firstSample == 1) {
    acc_est[0] = acc_norm[0];
    acc_est[1] = acc_norm[1];
    acc_est[2] = acc_norm[2];
    firstSample = 0;
}

// TODO Remove small Rz

// Get angles
Axz = atan2(acc_est[0], acc_est[2]) * 180 / PI;
Axz += ang_rate[1] * timestamp_us;
Ayz = atan2(acc_est[1], acc_est[2]) * 180 / PI;
Ayz += ang_rate[0] * timestamp_us;

int signGyro = ( cos(ang_rate[0] * PI / 180) >= 0) ? 1 : -1;

// Determine estimated gyro values
ang_rate_est[0] = sin(Axz * PI / 180);
ang_rate_est[0] /= sqrt(1 + square(cos(Axz * PI / 180)) * square(tan(Ayz * PI / 180)));
ang_rate_est[1] = sin(Ayz * PI / 180);
ang_rate_est[1] /= sqrt(1 + square(cos(Ayz * PI / 180)) * square(tan(Axz * PI / 180)));
ang_rate_est[2] = signGyro * sqrt(1 - square(ang_rate_est[0]) -
square(ang_rate_est[1]));

// Combine readings
acc_est[0] = (acc_norm[0] + 10 * ang_rate_est[0]) / (1 + 10);
acc_est[1] = (acc_norm[1] + 10 * ang_rate_est[1]) / (1 + 10);
acc_est[2] = (acc_norm[2] + 10 * ang_rate_est[2]) / (1 + 10);

normalize_1x3_vec(acc_est);

// component vector angles
printf("%6f %6f %6f\n", timestamp_us, acc[0], acc_est[0]);
//printf("accel(%d)      %6f %6f %6f\n", count, acc[0], acc[1], acc[2]);
//printf("accel_est(%d) %6f %6f %6f\n", count, acc_est[0], acc_est[1], acc_est[2]);
//printf("ang_rate(%d)  %6f %6f %6f\n", count, ang_rate[0], ang_rate[1],
ang_rate[2]);
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[]) {

    spatial_orientation();
    return 0;
}

/* ex: set ts=2 tw=80 shiftwidth=2 autoindent cindent expandtab: */

```

## Appendix C – Preliminary Thruster Control Programs

```
1 /* This is the mission control Thruster Testing program of RoboSUB 2012
2 *
3 * FOR Thruster testing program
4 *
5 * Author: Hang Zhang
6 *
7 * Thrusters Position PWM PIN IO PIN
8 * T1 Left 5 7
9 * T2 Right 6 8
10 * T3 Front 9 12
11 * T4 Back 10 13
12 * T5 Top 3 4
13 * T6 Bottom 11 2
14 */
15
16 #include "arduino.h"
17 #include "thruster.h"
18 #include <sys/wait.h>
19 #include <stdio.h>
20 #include <stdlib.h>
21 #include <unistd.h>
22 #include <string.h>
23 #include <sys/types.h>
24 #include <sys/time.h>
25 #include <math.h>
26 #include <string.h>
27
28 const char * SERIAL_PORT = "/dev/ttyACM0";
29 const int BAUD_R = 9600;
30
31 const unsigned int INIT_TIME = 15; // 15s
32 const unsigned int DELAY5S = 5; // 5s delay
33 const unsigned int DELAY1S = 1; // 1s delay
34
35 //-----//
36 int PWM_fd = 0; // fd for PWM controller interface
37
38 int main(int argc, char* argv[])
39 {
40 int speed = 0;
41
42 //-- PWM_Controller Interface initializaiton --//
43 PWM_fd = serialport_init(SERIAL_PORT,BAUD_R);
44 if(PWM_fd==-1) //exit(0);
45 printf("Arduino board is not connected!\n");
46 //-----//
47 speed = 30;
48 AdjustDepth(PWM_fd, speed);
49 sleep(INIT_TIME); // 15s delay;
50
51
52 speed = 50;
```

```

53 MoveForward(PWM_fd, speed);
54 KeepMoving(2*DELAY5S); // keepmoving for 10s
55 Stop(PWM_fd,DELAY5S);
56
57 speed = 30;
58 MakeTurn(PWM_fd, "left",speed);
59 KeepMoving(3*DELAY1S);
60 Stop(PWM_fd,DELAY5S); // stop for 5s
61
62 speed = 100;
63 MoveForward(PWM_fd, speed);
64 KeepMoving(2*DELAY5S);
65 Stop(PWM_fd,DELAY5S);
66
67 speed = 30;
68 MakeTurn(PWM_fd,"right",speed);
69 KeepMoving(3*DELAY1S);
70 Stop(PWM_fd,DELAY5S);
71
72 speed = 30;
73 MoveBackward(PWM_fd, speed);
74 KeepMoving(2*DELAY5S);
75 Stop(PWM_fd,DELAY5S);
76
77 speed = 50;
78 ShiftLeft(PWM_fd,speed);
79 KeepMoving(2*DELAY5S);
80 Stop(PWM_fd,DELAY5S);
81
82 ShiftRight(PWM_fd,speed);
83 KeepMoving(2*DELAY5S);
84 Stop(PWM_fd,DELAY5S);
85
86 speed = 30;
87 Spin(PWM_fd, speed);
88 KeepMoving(2*DELAY5S);
89 Stop(PWM_fd,DELAY5S);
90
91 FloatVehicle(PWM_fd);
92
93 return 0;
94
95 }
96

```

```

1 #ifndef THRUSTER_H_
2 #define THRUSTER_H_
3 /*
4 * Header file for Thruster control program
5 * Function declarations
6 * To be used by Mission Control
7 *
8 * Author: Hang Zhang
9 *
10 *
11 */
12 #include <string.h>
13
14 void AdjustDepth(int,int);
15 void MoveForward(int,int);
16 void MakeTurn(int,const char *, int);
17 void MoveBackward(int,int);
18 void ShiftLeft(int,int);
19 void ShiftRight(int,int);
20 void Stop(int,unsigned int);
21 void Spin(int,int);
22 void KeepMoving(unsigned int);
23 void FloatVehicle(int);
24 void GetCMD(char *, int, char);
25 void printCMD(char *);
26
27 #endif
28

```

```

1 #ifndef THRUSTER_C_
2 #define THRUSTER_C_
3 /*
4 * Thruster control program
5 * To be used by Mission Control
6 *
7 * Author: Hang Zhang
8 *
9 *
10 */
11 #include "thruster.h"
12 #include "arduino.h"
13 #include <unistd.h>
14 #include <stdlib.h>
15 #include <string.h>
16 #include <stdio.h>
17
18 static const char T_PWM = '0';
19 static const char BOT_PWM = '1';
20 static const char L_PWM = '2';
21 static const char R_PWM = '3';
22 static const char F_PWM = '4';
23 static const char BAC_PWM = '5';
24 static const int dt = 100000; // ramping delay
25

```

```

26 //=====//
27 // Set the left and right motor to adjust//
28 // depth //
29 //=====//
30 void AdjustDepth(int PWM_fd, int spd)
31 {
32 printf("adjusting depth\n");
33 char cmd[5] = {'\0'};
34 cmd[0] = L_PWM;
35 cmd[1] = (spd/10) + '0';
36 cmd[2] = (spd%10) + '0';
37 cmd[3] = 'f';
38 serialport_write(PWM_fd,cmd);
39 usleep(10); // wait for 10 us
40 cmd[0] = R_PWM;
41 // printCMD(cmd);
42 serialport_write(PWM_fd,cmd);
43 }
44
45 void MoveForward(int PWM_fd,int spd){
46 int s = 10;
47 spd = 150;
48 char cmd[6] = {'\0'};
49 printf("moving forward\n");
50
51 for(int i=s; i< spd;i++){ // ramp up speed
52 s++;
53 GetCMD(cmd,s,'f');
54 cmd[0] = T_PWM;
55 serialport_write(PWM_fd,cmd);
56 cmd[0] = BOT_PWM;
57 serialport_write(PWM_fd,cmd);
58 usleep(dt);
59 }
60 }
61
62 void MakeTurn(int PWM_fd,const char * dir, int spd)
63 {
64
65 printf("turning %s\n",dir);
66 int s = 10;
67 char cmd[6] = {'\0'};
68 char d;
69 if(strcmp(dir,"left") == 0)
70 d = 'f';
71 else
72 d = 'b';
73 for(int i=s; i< spd;i++){ //ramp up speed
74 s++;
75 GetCMD(cmd,s,d);
76 cmd[0] = F_PWM;
77 serialport_write(PWM_fd,cmd);
78 cmd[0] = BAC_PWM;
79 serialport_write(PWM_fd,cmd);
80 usleep(dt);
81 }
82 }

```

```

83
84 void MoveBackward(int PWM_fd,int spd)
85 {
86 int s = 10;
87 char cmd[6] = {'\0'};
88 printf("moving backward\n");
89
90 for(int i=s; i< spd;i++){
91 s++;
92 GetCMD(cmd,s,'b');
93 cmd[0] = T_PWM;
94 serialport_write(PWM_fd,cmd);
95 cmd[0] = BOT_PWM;
96 serialport_write(PWM_fd,cmd);
97 usleep(dt); // wait for 10 us
98 }
99 }
100 void Stop(int PWM_fd, unsigned int delay)
101 {
102 printf("vehicle stopped\n");
103 char cmd[4] = {'\0'};
104 cmd[0] = T_PWM;
105 cmd[1] = '0';
106 cmd[2] = 's';
107 serialport_write(PWM_fd,cmd);
108 cmd[0] = BOT_PWM;
109 serialport_write(PWM_fd,cmd);
110 cmd[0] = F_PWM;
111 serialport_write(PWM_fd,cmd);
112 cmd[0] = BAC_PWM;
113 serialport_write(PWM_fd,cmd);
114 sleep(delay);
115 }
116
117 void Spin(int PWM_fd,int spd)
118 { // speed for spin must less than 100
119 char cmd[5] = {'\0'};
120 cmd[0] = F_PWM;
121 cmd[1] = (spd/10) + '0';
122 cmd[2] = (spd%10)+'0';
123 cmd[3] = 'f';
124 serialport_write(PWM_fd,cmd);
125 cmd[0] = BAC_PWM;
126 cmd[3] = 'b';
127 serialport_write(PWM_fd,cmd);
128 printf("vehicle spinning\n");
129 }
130
131 void ShiftLeft(int PWM_fd,int spd)
132 { int s = 10;
133 char cmd[6] = {'\0'};
134
135 for(int i=s; i< spd;i++){
136 s++;
137 GetCMD(cmd,s,'f');
138 cmd[0] = F_PWM;
139 serialport_write(PWM_fd,cmd);

```

```

140 cmd[0] = BAC_PWM;
141 serialport_write(PWM_fd,cmd);
142 usleep(dt);
143 }
144
145 printf("shifting left\n");
146 }
147 void ShiftRight(int PWM_fd,int spd)
148 { int s = 10;
149 char cmd[6] = {'\0'};
150
151 for(int i=s; i< spd;i++){
152 s++;
153 GetCMD(cmd,s,'b');
154 cmd[0] = F_PWM;
155 serialport_write(PWM_fd,cmd);
156 cmd[0] = BAC_PWM;
157 serialport_write(PWM_fd,cmd);
158 usleep(dt);
159 }
160
161 }
162
163 void FloatVehicle(int PWM_fd)
164 {
165 char cmd[4] = {'\0'};
166 cmd[0] = L_PWM;
167 cmd[1] = '0';
168 cmd[2] = 's';
169 serialport_write(PWM_fd,cmd);
170 cmd[0] = R_PWM;
171 serialport_write(PWM_fd,cmd);
172 printf("Floating the vehicle\n");
173 }
174
175 void KeepMoving(unsigned int delay)
176 {
177 printf("keep moving\n");
178 sleep(delay);
179 }
180 //=====//
181 // Convert integer to char stream //
182 //=====//
183 void GetCMD(char *cmd, int s, char dir ){
184
185 if(s < 100){
186 cmd[1] = (s/10)+'0';
187 cmd[2] = (s%10)+'0';
188 cmd[3] = dir;
189 }
190 else
191 {
192 cmd[1] = (s/100) + '0';
193 cmd[2] = ( (s%100) / 10 ) + '0';
194 cmd[3] = (s%10) + '0';
195 cmd[4] = dir;
196 }

```

```

197 }
198
199 void printCMD(char * cmd)
200 {
201 printf("cmd: %s\n",cmd);
202 }
203 #endif
204

```

```

1 #ifndef ARDUINO_H_
2 #define ARDUINO_H_
3 // header file for serial port communication between
4 // Arduino boards and mission control
5 #include <stdint.h>
6
7 int serialport_init(const char*, int);
8
9 int serialport_writebyte(int,uint8_t);
10
11 int serialport_write(int, const char*);
12
13 int serialport_read_until(int,char*, char);
14
15 int Print_Cont_Menu();
16
17 #endif
18

```

```

1 #ifndef ARDUINO_C_
2 #define ARDUINO_C_
3
4 /* RoboSub MC Arduino Interface */
5 *=====*
6 * *
7 * *
8 * Interface w/ Thruster Controller: *
9 * Send: id+spd+dir eg: '31271' *
10 * id: 3->T/B || 5->Left || 6->Right || 9->Front || *
11 * 10->Back
12 * spd: int; 0<= spd <= 255
13 * dir: char; 'f' || 'b'
14 *
15 */
16
17 #include "arduino.h"
18 #include <stdio.h> /* Standard input/output definitions */
19 #include <stdlib.h>
20 #include <stdint.h> /* Standard types */
21 #include <string.h> /* String function definitions */
22 #include <unistd.h> /* UNIX standard function definitions */
23 #include <fcntl.h> /* File control definitions */
24 #include <errno.h> /* Error number definitions */
25 #include <termios.h> /* POSIX terminal control definitions */

```



```

26 #include <sys/ioctl.h>
27 #include <getopt.h>
28
29
30 int serialport_init(const char* serialport, int baud)
31 {
32     struct termios toptions;
33     int fd;
34
35     //fprintf(stderr,"init_serialport: opening port %s @ %d bps\n",
36     // serialport,baud);
37
38     fd = open(serialport, O_RDWR | O_NOCTTY | O_NDELAY);
39     if (fd == -1) {
40         perror("init_serialport: Unable to open port ");
41         return -1;
42     }
43
44     if (tcgetattr(fd, &toptions) < 0) {
45         perror("init_serialport: Couldn't get term attributes");
46         return -1;
47     }
48     speed_t brate = baud;
49     switch(baud) {
50     case 4800: brate=B4800; break;
51     case 9600: brate=B9600; break;
52     #ifdef B14400 // not defined in termios.h
53     case 14400: brate=B14400; break;
54     #endif
55     case 19200: brate=B19200; break;
56     #ifdef B28800 // not defined in termios.h
57     case 28800: brate=B28800; break;
58     #endif
59     case 38400: brate=B38400; break;
60     case 57600: brate=B57600; break;
61     case 115200: brate=B115200; break;
62     }
63     cfsetispeed(&toptions, brate);
64     cfsetospeed(&toptions, brate);
65     // 8N1
66     toptions.c_cflag &= ~PARENB;
67     toptions.c_cflag &= ~CSTOPB;
68     toptions.c_cflag &= ~CSIZE;
69     toptions.c_cflag |= CS8;
70     // no flow control
71     toptions.c_cflag &= ~CRTSCTS;
72
73     toptions.c_cflag |= CREAD | CLOCAL; // turn on READ & ignore ctrl lines
74     toptions.c_iflag &= ~(IXON | IXOFF | IXANY); // turn off s/w flow ctrl
75
76     toptions.c_lflag &= ~(ICANON | ECHO | ECHOE | ISIG); // make raw
77     toptions.c_oflag &= ~OPOST; // make raw
78
79     toptions.c_cc[VMIN] = 0;
80     toptions.c_cc[VTIME] = 20;
81
82     if( tcsetattr(fd, TCSANOW, &toptions) < 0) {

```

```

83 perror("init_serialport: Couldn't set term attributes");
84 return -1;
85 }
86
87 return fd;
88 }
89 // write a single byte, used for Arduino board 2
90 int serialport_writebyte( int fd, uint8_t b)
91 {
92 int n = write(fd,&b,1);
93 if( n!=1)
94 return -1;
95 return 0;
96 }
97
98 // send a character stream to the arduino board 1, control PWMs
99 int serialport_write(int fd, const char* str)
100 {
101 int len = strlen(str);
102 int n = write(fd, str, len);
103 if( n!=len )
104 return -1;
105 return 0;
106 }
107 // read data stream from Arduino board 2, for pressure sensor
108 // and hydrophone reading
109 int serialport_read_until(int fd, char* buf, char until)
110 {
111 char b[1];
112 int i=0;
113 do {
114 int n = read(fd, b, 1); // read a char at a time
115 if( n==-1) return -1; // couldn't read
116 if( n==0 ) {
117 usleep( 10 * 1000 ); // wait 10 msec try again
118 continue;
119 }
120 buf[i] = b[0]; i++;
121 } while( b[0] != until );
122
123 buf[i] = 0; // null terminate the string
124 return 0;
125 }
126
127 /*
128 int Print_Cont_Menu(){
129 printf("Please make a choice: \n");
130 printf("1.");
131 }
132 */
133
134 /*
135 void main(int argc, char * argv[]){
136 int fd = 0;
137 int choice;
138 char buf[128]; // maximum buffer of Arduino is 128bytes
139 int rc;

```

```

140
141
142 fd = serialport_init(SERIAL_PORT,BAUD_R);
143 if(fd==-1) exit(0);
144 printf("*****RoboSUB Thruster Controller*****\n");
145 printf("\nInstruction: enter 0-255 as speed\nfollowed by 'l' or 'r' as
rotation
direction\n ");
146
147
148 while(1){
149 printf("\n\nEnter the speed and direction: ");
150 scanf("%s",buf);
151 rc = serialport_write(fd,buf);
152 if(rc==-1) break;
153 }
154 }
155
156 */
157
158 #endif
159

```

```

1 /*
2 *PWM(Thruster) Controller Program
3 *running on Arduino board
4 *
5 *Author: Hang Zhang
6 *
7 *Setting a duty cycle of 255 sets the related pin to a constant +5V,
8 *while a duty cycle of 0 sets the related pin to a constant 0V
9 *
10 *PWM frequencies are tied together in pairs of pins. If one in a
11 *pair is changed, the other is also changed to match:
12 * - Pins 5 and 6 are paired on timer0
13 * - Pins 9 and 10 are paired on timer1
14 * - Pins 3 and 11 are paired on timer2
15
16 Thrusters Position PWM PIN IO PIN
17 T1 Left 5 7 8
18 T2 Right 6 11 12
19 T3 Front 9 A0 A1
20 T4 Back 10 A2 A3
21 T5 Top 3 2 4
22 T6 Bottom 11 A4 A5
23 */
24
25 #define T_PWM 3 // Top and Bottom PWM ---- Propulsion
26 #define BOT_PWM 11
27 #define L_PWM 5 // Left PWM ---- depth
28 #define R_PWM 6 // Right PWM ---- depth
29 #define F_PWM 9 //Front Thruster ---- Orientati
30 #define BAC_PWM 10 // Back PWM ---- Orientation
31
32 #define T_DIR_F 2

```

```

33 #define BOT_DIR_F A4
34 #define L_DIR_F 7
35 #define R_DIR_F 11
36 #define F_DIR_F A0
37 #define BAC_DIR_F A2
38
39 #define T_DIR_B 4
40 #define BOT_DIR_B A5
41 #define L_DIR_B 8
42 #define R_DIR_B 12
43 #define F_DIR_B A1
44 #define BAC_DIR_B A3
45
46
47 int int_array[] = {0,0,0,0,0}; // int_array[0] = id;
48 int tid; // tid = int_array[0]; The Thruster ID
49 int cnt = 0;
50 char dir; // direction of the thruster 'f' or 'b'
51 int io_pin_f; // direction IO pin ID
52 int io_pin_b;
53 int spd; // speed of the thrusters
54 int pwm;
55
56 void setup()
57 {
58   Serial.begin(9600);
59   pinMode(T_PWM, OUTPUT);
60   pinMode(L_PWM, OUTPUT);
61   pinMode(R_PWM, OUTPUT);
62   pinMode(F_PWM, OUTPUT);
63   pinMode(BOT_PWM, OUTPUT);
64   pinMode(BAC_PWM, OUTPUT);
65
66   pinMode(T_DIR_F, OUTPUT);
67   pinMode(BOT_DIR_F, OUTPUT);
68   pinMode(L_DIR_F, OUTPUT);
69   pinMode(R_DIR_F, OUTPUT);
70   pinMode(F_DIR_F, OUTPUT);
71   pinMode(BOT_DIR_F, OUTPUT);
72
73   pinMode(T_DIR_B, OUTPUT);
74   pinMode(BOT_DIR_B, OUTPUT);
75   pinMode(L_DIR_B, OUTPUT);
76   pinMode(R_DIR_B, OUTPUT);
77   pinMode(F_DIR_B, OUTPUT);
78   pinMode(BOT_DIR_B, OUTPUT);
79
80   while(Serial.available()) // clear buffer
81     Serial.read();
82
83 }
84
85 // main function
86 void loop()
87 {
88   read_speed();
89   run();

```

```

90 }
91
92 //-----//
93 // This function will wait for the mission //
94 // control program to send the command to //
95 // the thrusters. //
96 // Command will be stored in an array called //
97 // int_array[] of size 5 //
98 // int_array[0]: thruster id //
99 // int_array[1-3]: Speed level //
100 // int_array[4]: direction 'f' || 'b' //
101 //-----//
102 int read_speed()
103 {
104
105 cnt = 0;
106
107 Serial.println("The thruster is waiting for command");
108
109 while(Serial.available() == 0)
110 {
111 delay(10); // wait for 10ms and try to read from serial port again
112 }
113
114 while( Serial.available() > 0 && cnt < 5)
115 {
116 int_array[cnt++]= Serial.read(); // read the serial port
117 delay(100);
118 }
119
120 Serial.println("Command received!");
121
122 if(cnt != 0){
123 tid = int_array[0];
124 if(cnt == 3){
125 spd = int_array[1]-'0';
126 dir = int_array[2];
127 }
128 else if(cnt==4){
129 spd = 10*(int_array[1]-'0') + (int_array[2]-'0');
130 dir = int_array[3];
131 }
132 else if(cnt == 5){
133 spd = 100*(int_array[1]-'0') + 10*(int_array[2]-'0') +(int_array[3]-'0');
134 dir = int_array[4];
135 }
136 else
137 Serial.println("ERROR");
138 //-----//
139 Serial.print("Duty Cycle of the PWM is set to: ");
140 Serial.println(spd,DEC);
141 Serial.print("Direction is set to: ");
142 Serial.println(dir);
143 }
144
145 }
146

```

```

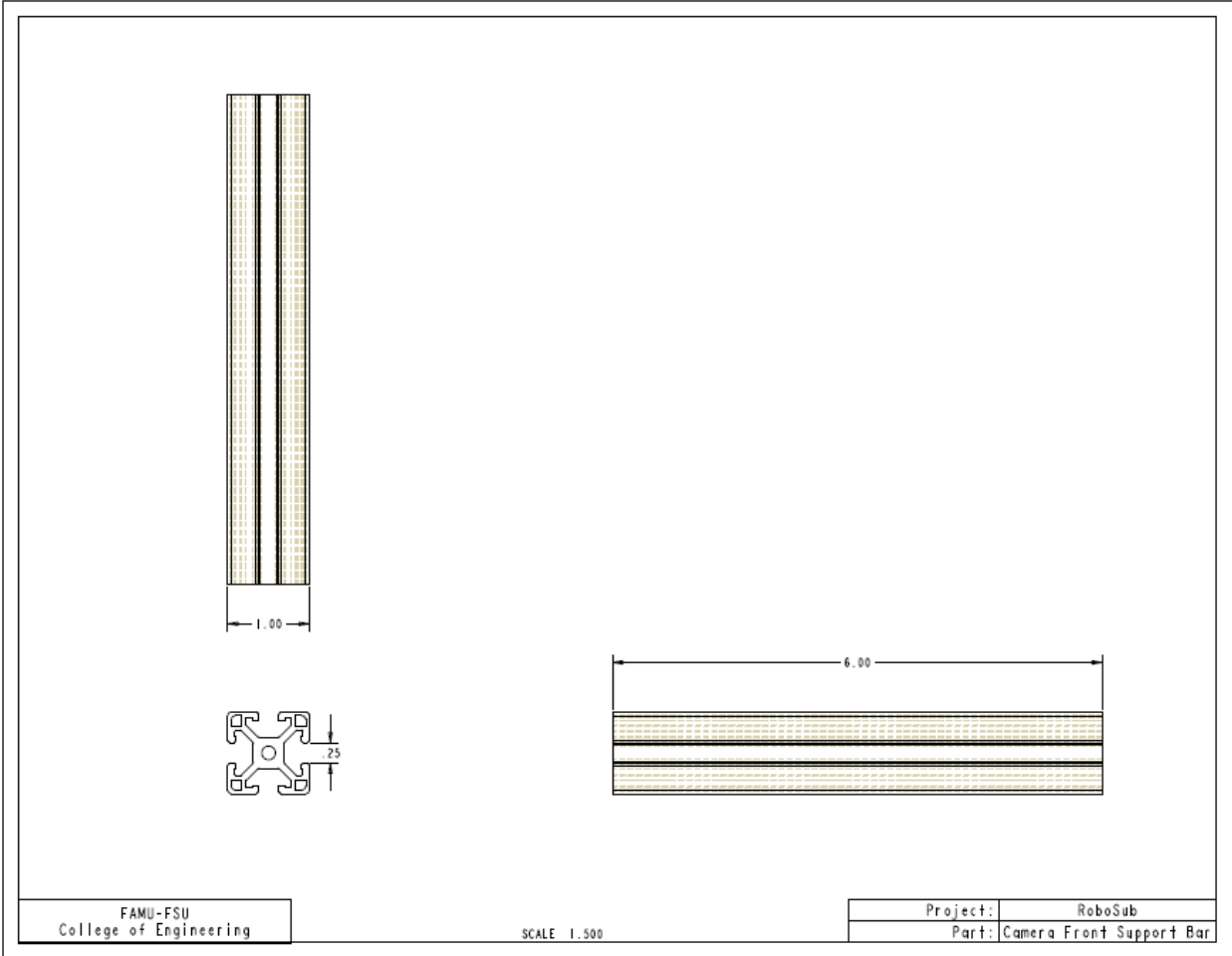
147 //-----//
148 // This function will set the thruster to //
149 // rotate at the specified speed and direc- //
150 // tion according to the mission control //
151 // program. //
152 //-----//
153 void run()
154 {
155   switch (tid)
156   {
157     case 0 :
158       pwm = T_PWM;
159       io_pin_f = T_DIR_F;
160       io_pin_b = T_DIR_B;
161       break;
162     case 1 :
163       pwm = BOT_PWM;
164       io_pin_f = BOT_DIR_F;
165       io_pin_b = BOT_DIR_B;
166       break;
167     case 2 :
168       pwm = L_PWM;
169       io_pin_f = L_DIR_F;
170       io_pin_b = L_DIR_B;
171       break;
172
173     case 3 :
174       pwm = R_PWM;
175       io_pin_f = R_DIR_F;
176       io_pin_b = R_DIR_B;
177       break;
178
179     case 4 :
180       pwm = F_PWM;
181       io_pin_f = F_DIR_F;
182       io_pin_b = F_DIR_B;
183       break;
184     case 5 :
185       pwm = BAC_PWM;
186       io_pin_f = BAC_DIR_F;
187       io_pin_b = BAC_DIR_B;
188       break;
189     default :
190       break;
191   }
192
193   if(dir == 'f'){
194     digitalWrite(io_pin_f,HIGH);
195     digitalWrite(io_pin_b,LOW);
196     analogWrite(pwm,spd);
197   }
198   else if(dir == 'b'){
199     digitalWrite(io_pin_f,LOW);
200     digitalWrite(io_pin_b,HIGH);
201     analogWrite(pwm,spd);
202   }
203   else{ // 's' Break the motor;

```

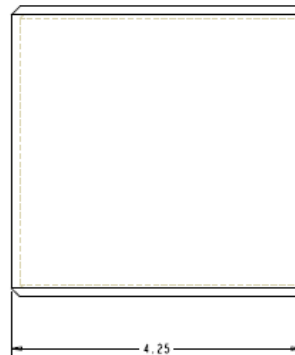
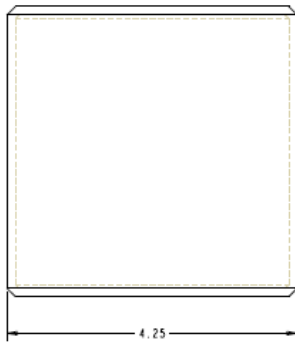
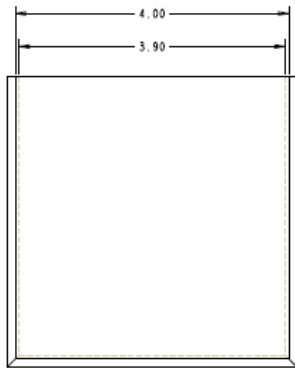
```
204 digitalWrite(pwm,HIGH); // EN to HIGH
205 digitalWrite(io_pin_f,LOW);
206 digitalWrite(io_pin_f,LOW);
207 }
208
209 /*
210 Serial.print("Thruster is now rotating to the LEFT at speed: ");
211 Serial.println(sp);
212 */
213 }
214 // =====END of PWM controller=====//
215
```

```
1 # This is the Makefile utility for Robosub 2012
2 # Author: Hang Zhang
3 CFLAGS = -std=gnu99 -Wall
4
5 run: main.o arduino.o thruster.o
6 gcc thruster.o arduino.o main.o -o main
7
8 main.o: main.c
9 gcc -c main.c $(CFLAGS)
10
11 thruster.o: thruster.c
12 gcc -c thruster.c $(CFLAGS)
13
14 arduino.o: arduino.c
15 gcc -c arduino.c $(CFLAGS)
16
17 clean:
18 rm *.o main
```

# Engineering Drawings





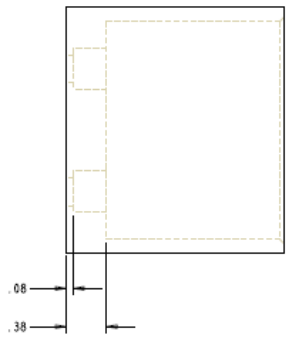
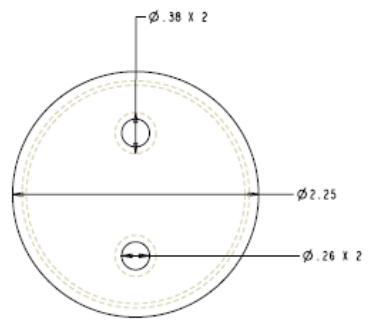
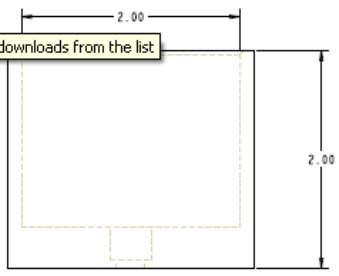


FAMU-FSU  
College of Engineering

SCALE: 1:250

Project:	RoboSub
Part:	Camera Enclosure

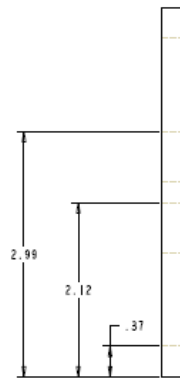
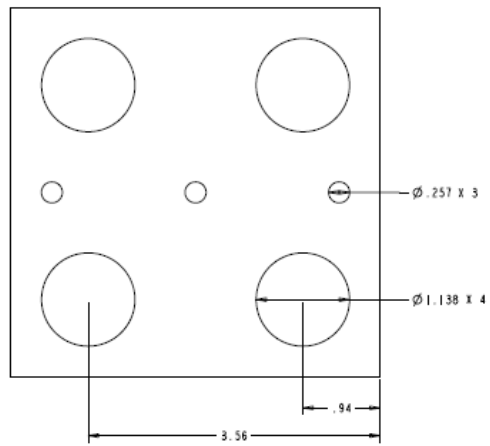
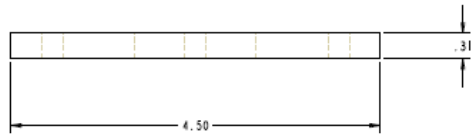
leted, canceled, and failed downloads from the list



FAMU-FSU  
College of Engineering

SCALE 2.000

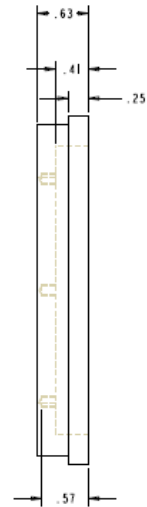
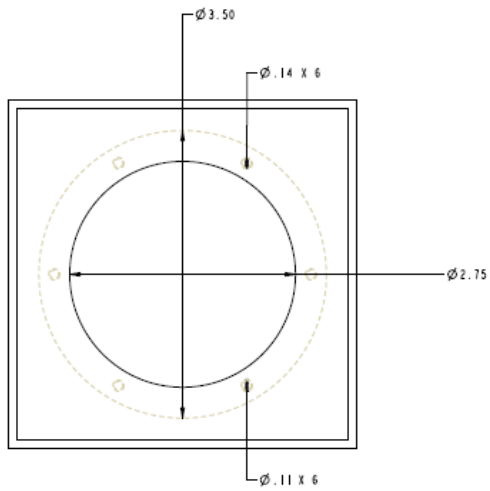
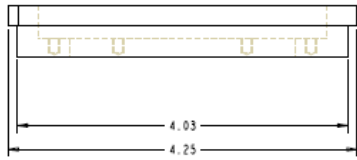
Project:	RoboSub
Part:	Air Cylinder Mount



FAMU-FSU  
College of Engineering

SCALE 1:500

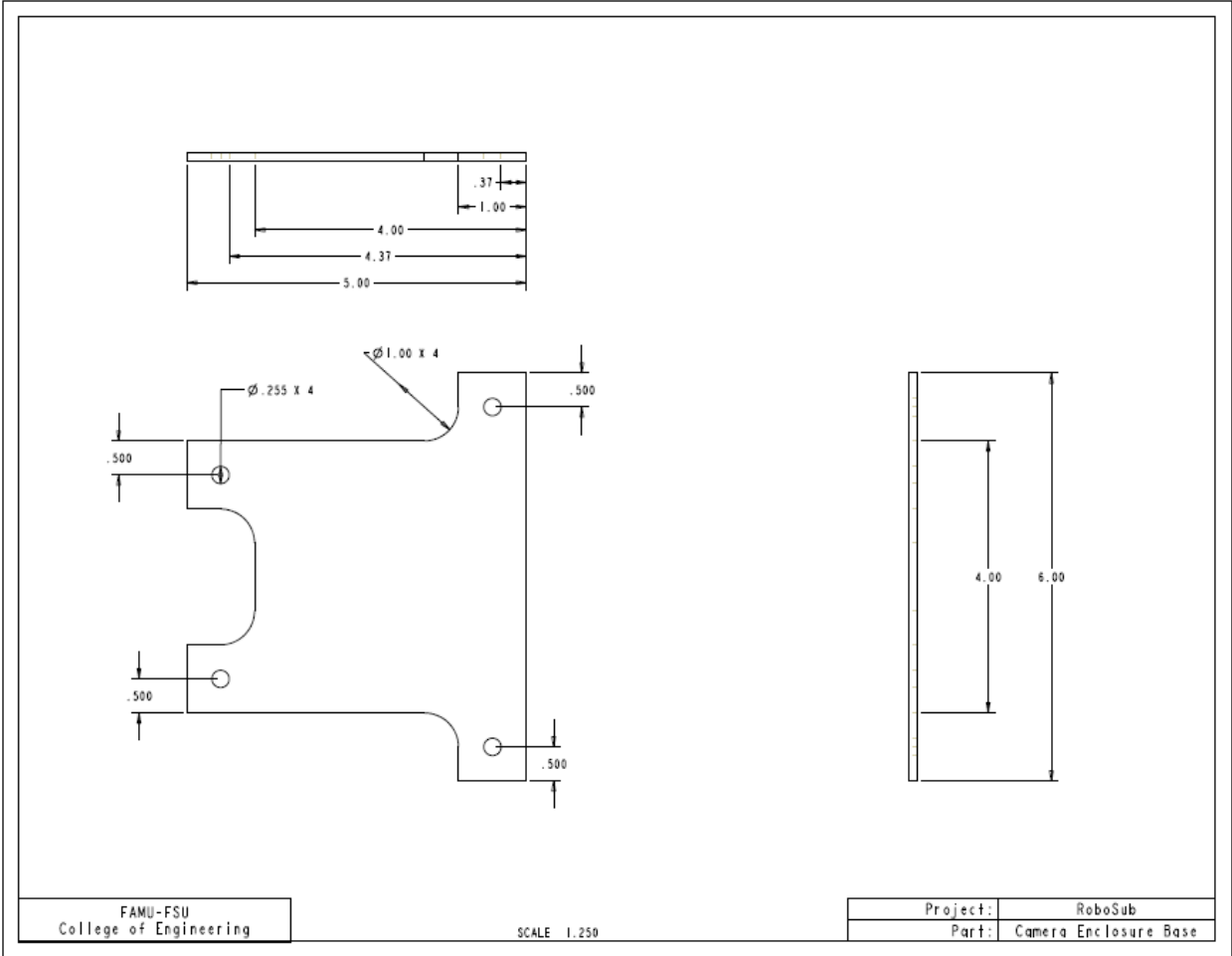
Project:	RoboSub
Part:	Hydrophone Mount

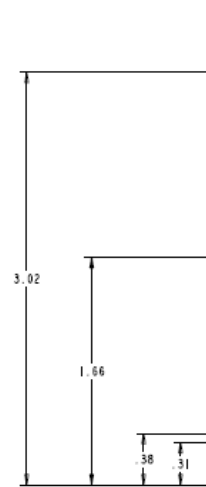
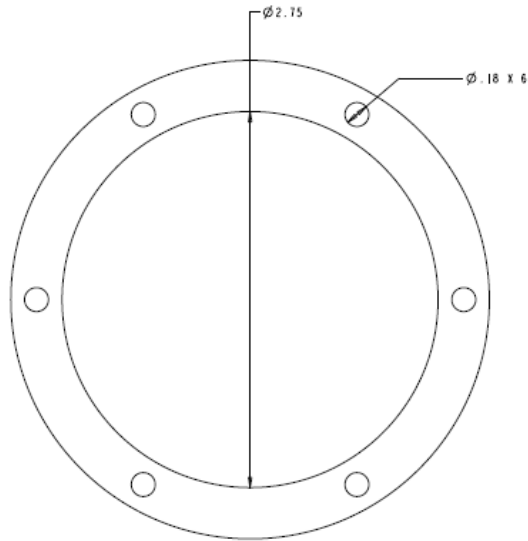
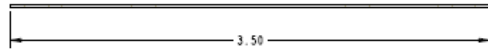


FAMU-FSU  
College of Engineering

SCALE 1:500

Project:	RoboSub
Part:	Camera Enclosure Cap

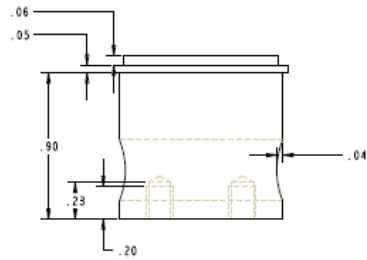
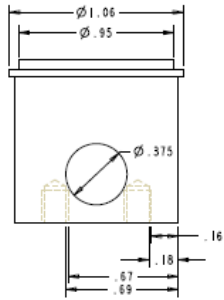
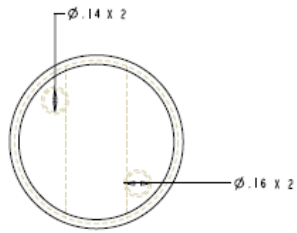




FAMU-FSU  
College of Engineering

SCALE 2.500

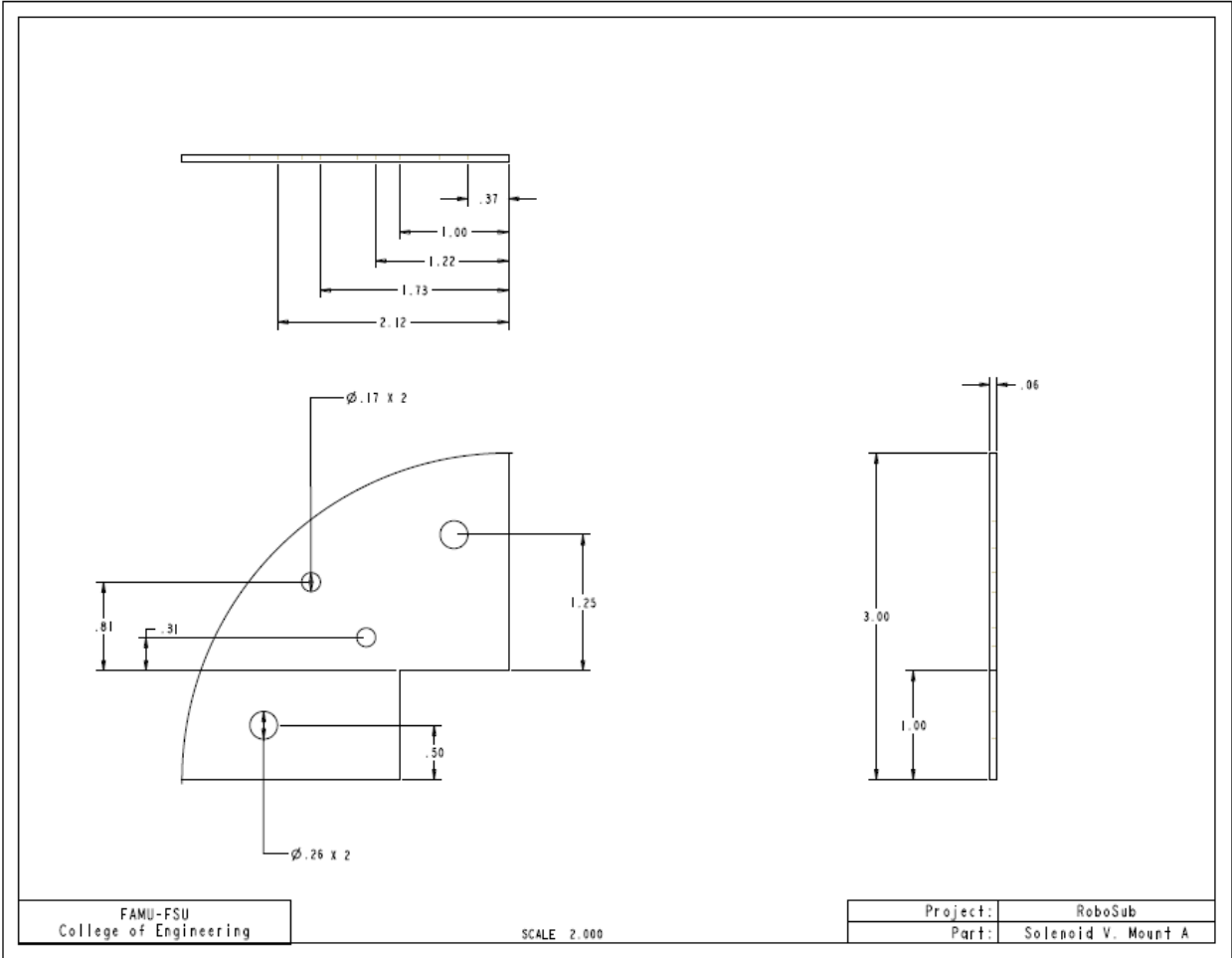
Project:	RoboSub
Part:	Camera Enc. Gasket



FAMU-FSU  
College of Engineering

SCALE 3.000

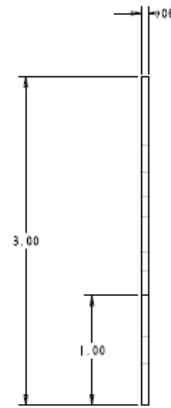
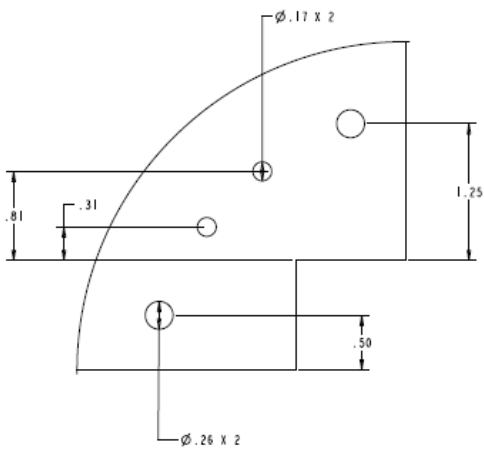
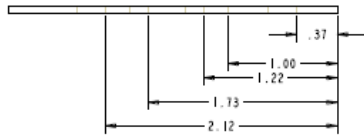
Project:	RoboSub
Part:	Solenoid Valve



FAMU-FSU  
College of Engineering

SCALE 2.000

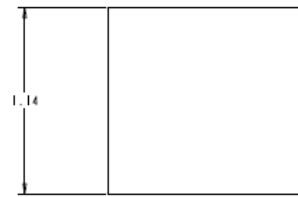
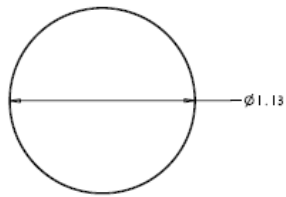
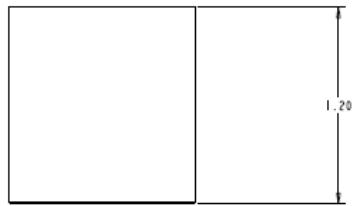




FAMU-FSU  
College of Engineering

SCALE 2.000

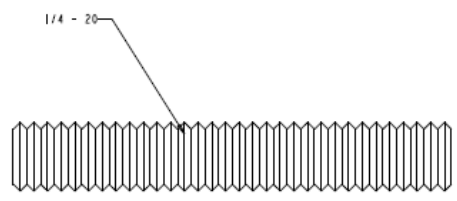
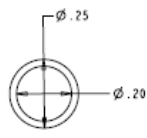
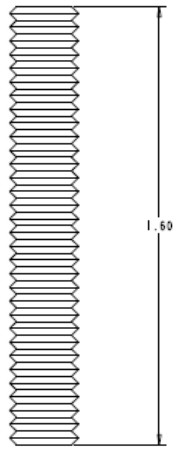
Project:	RoboSub
Part:	Solenoid V. Mount B



FAMU-FSU  
College of Engineering

SCALE 3.000

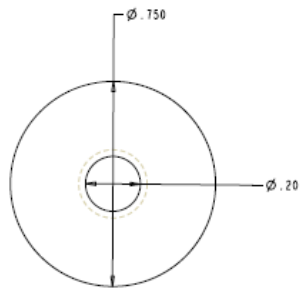
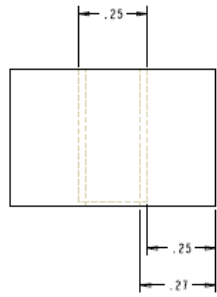
Project:	RoboSub
Part:	SQ26-01 Hydrophone



FAMU-FSU  
College of Engineering

SCALE 5.000

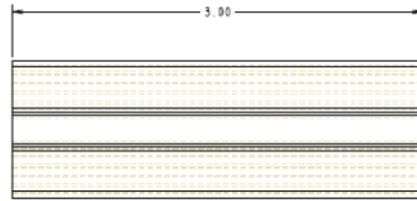
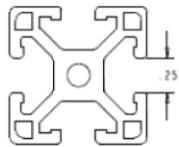
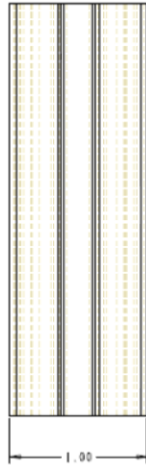
Project:	RoboSub
Part:	Camera Enc. Threaded Rod



FAMU-FSU  
College of Engineering

SCALE 5.000

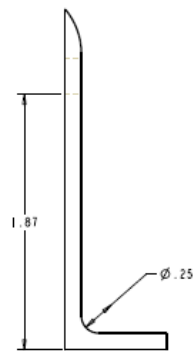
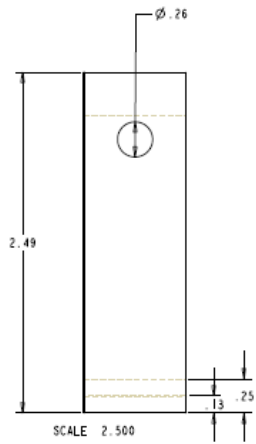
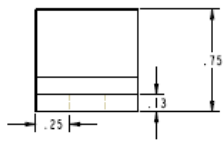
Project:	RoboSub
Part:	Threded Rod Mount



FAMU-FSU  
College of Engineering

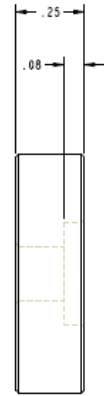
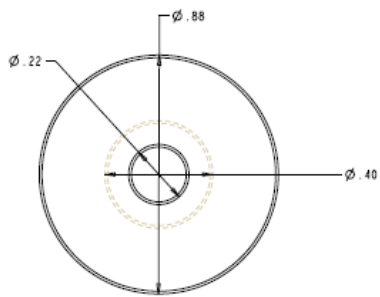
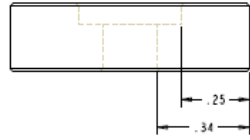
SCALE 2.500

Project:	RoboSub
Part:	Camera Side Support Bar



FAMU-FSU  
College of Engineering

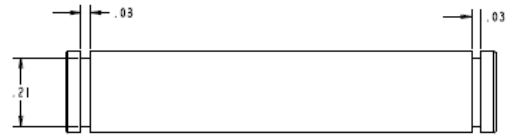
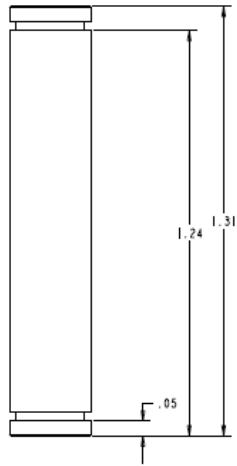
Project:	RoboSub
Part:	Camera Angle Bracket



FAMU-FSU  
College of Engineering

SCALE 5.000

Project:	RoboSub
Part:	Cylindrical Disk ATT.

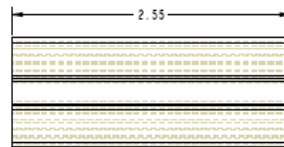
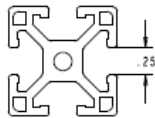
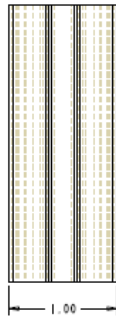


FAMU-FSU  
College of Engineering

SCALE 6.000

Project:	RoboSub
Part:	Dowell Pin 2

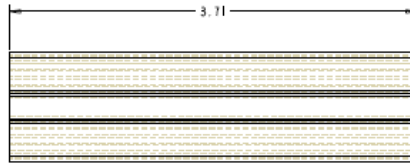
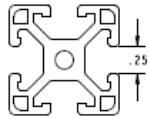
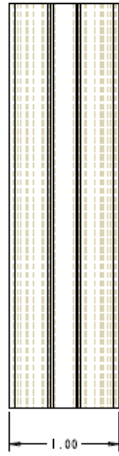




FAMU-FSU  
College of Engineering

SCALE 2.000

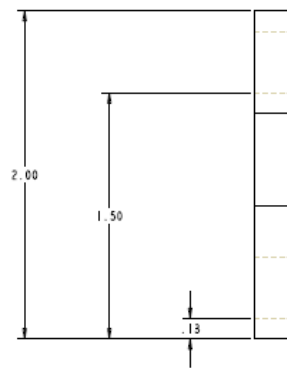
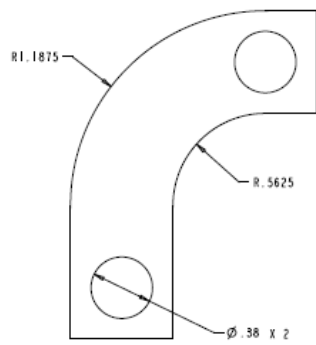
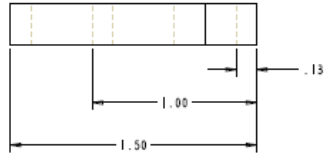
Project:	RoboSub
Part:	Grasp Rel. Mech. Top Support Bar



FAMU-FSU  
College of Engineering

SCALE 2.000

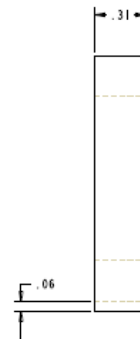
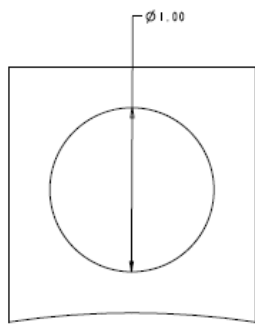
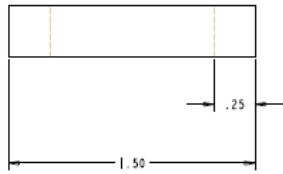
Project:	RoboSub
Part:	Grasp Rel. Mech. Long Support Bar



FAMU-FSU  
College of Engineering

SCALE 3.000

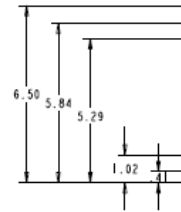
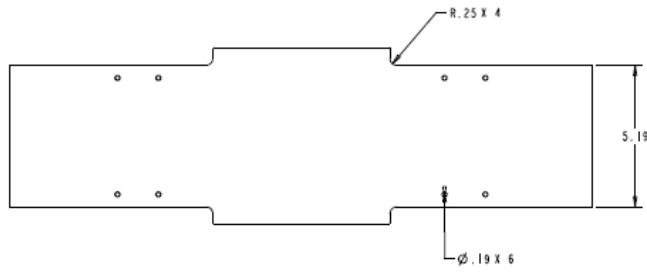
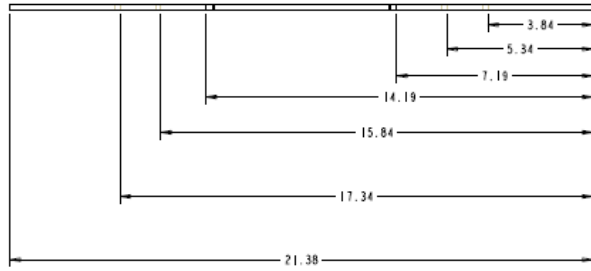
Project:	RoboSub
Part:	Grasp Rel. Mech. Claw



FAMU-FSU  
College of Engineering

SCALE 3.000

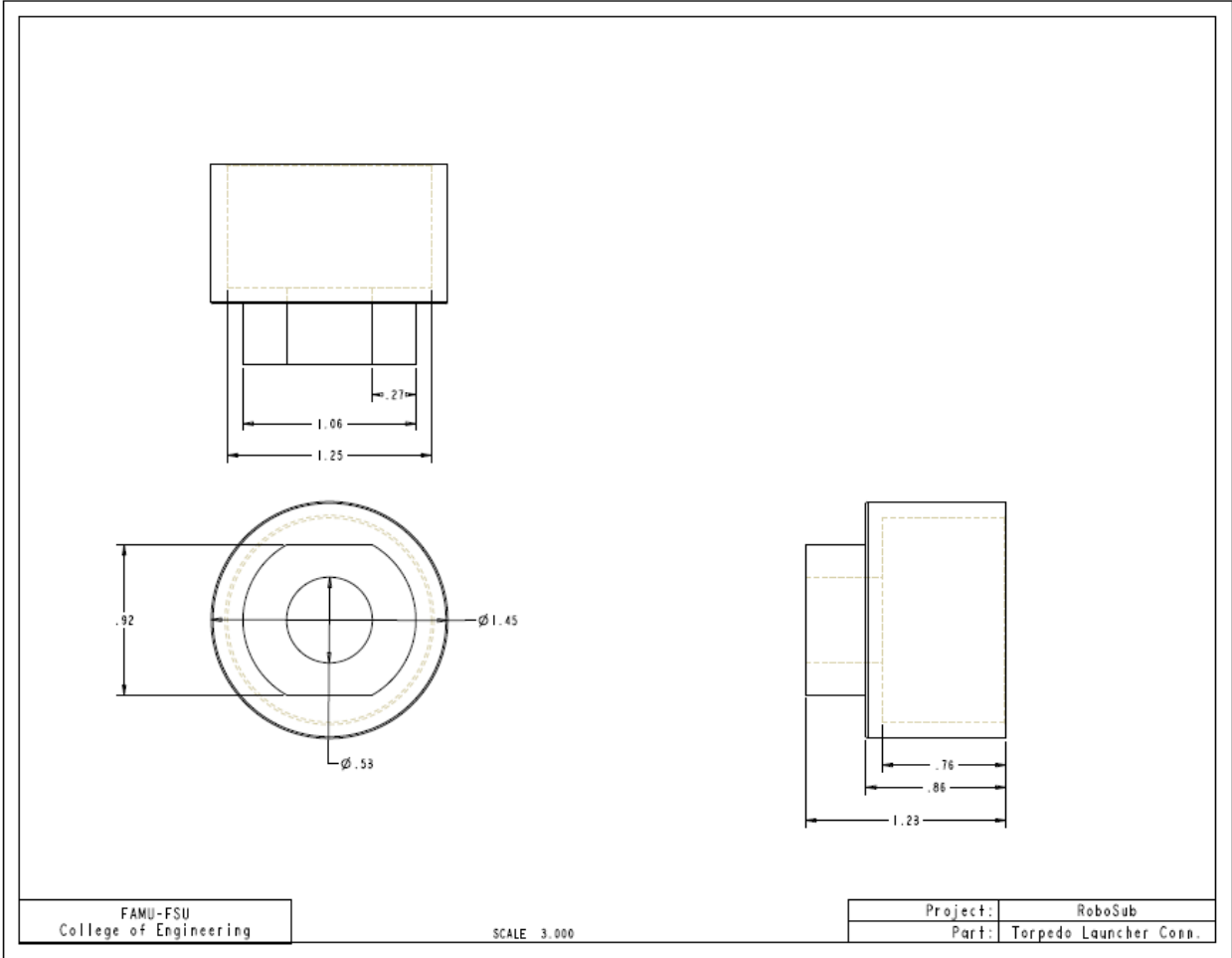
Project:	RoboSub
Part:	Pressure Sensor Mount



FAMU-FSU  
College of Engineering

SCALE 0.500

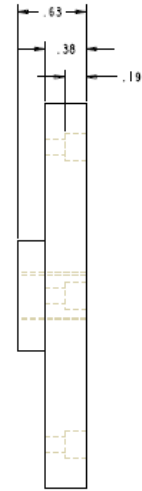
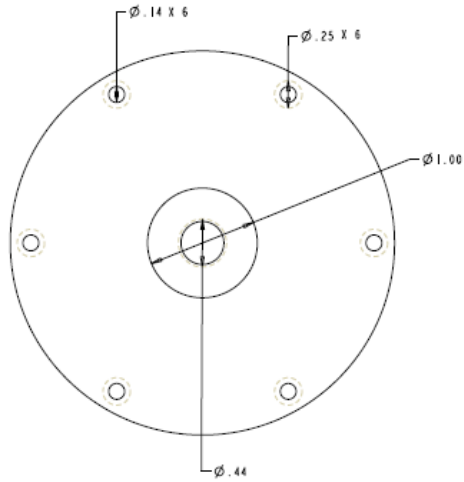
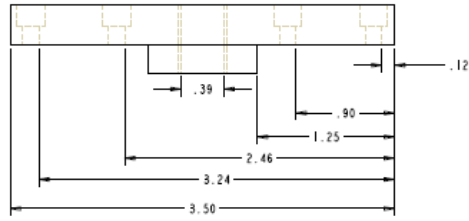
Project:	RoboSub
Part:	Bottom Electronics Rack



FAMU-FSU  
College of Engineering

SCALE 3.000

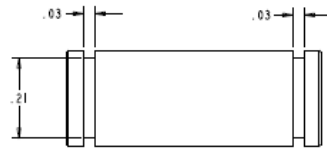
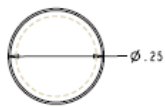
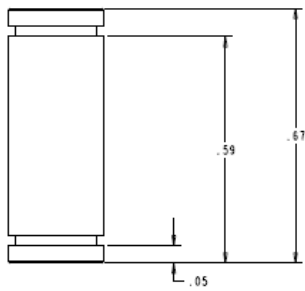
Project:	RoboSub
Part:	Torpedo Launcher Conn.



FAMU-FSU  
College of Engineering

SCALE 2.000

Project:	RoboSub
Part:	Camera Inclosure Inside

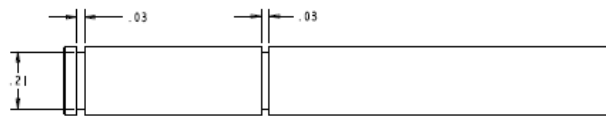
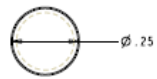
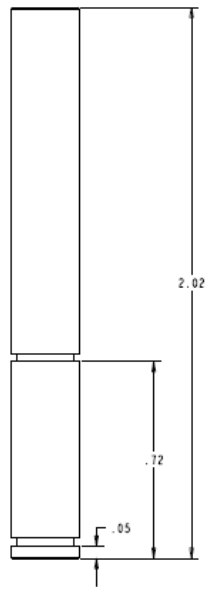


FAMU-FSU  
College of Engineering

SCALE 7.000

Project:	RoboSub
Part:	Dowell Pin 3

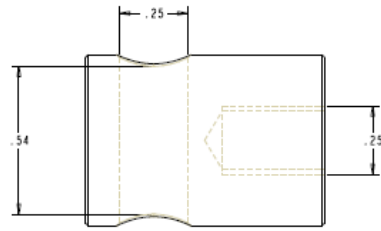
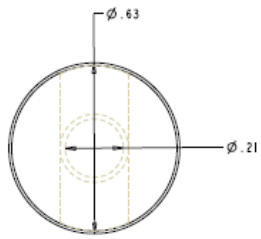
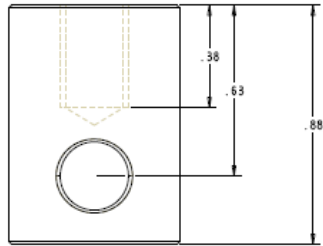




FAMU-FSU  
College of Engineering

SCALE 5.000

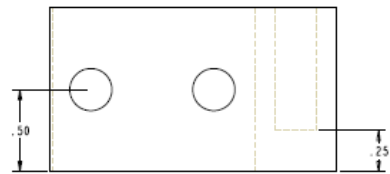
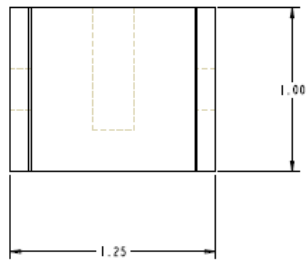
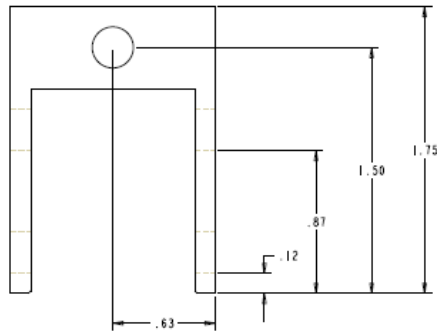
Project:	RoboSub
Part:	Dowel Pin 1



FAMU-FSU  
College of Engineering

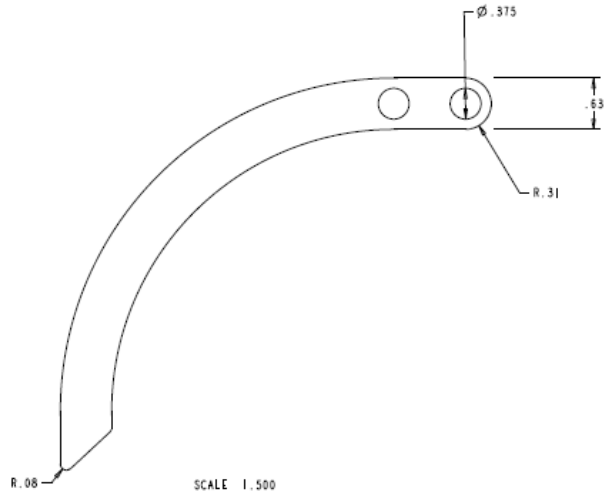
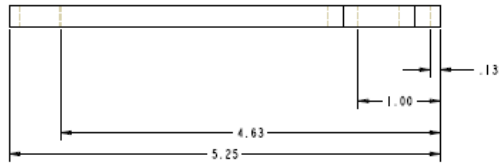
SCALE 5.000

Project:	RoboSub
Part:	Grasp Rel. Mech. Adpt.



FAMU-FSU  
College of Engineering

Project:	RoboSub
Part:	Grab Rel. Mech. Block

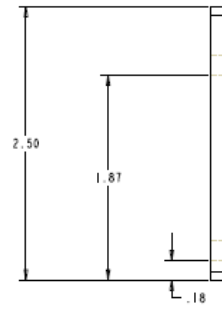
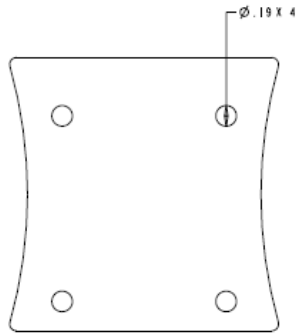
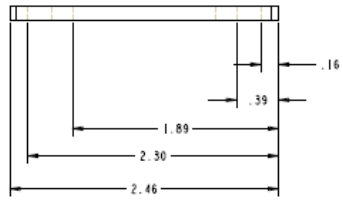


SCALE 1.500



FAMU-FSU  
College of Engineering

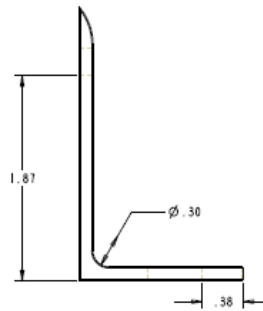
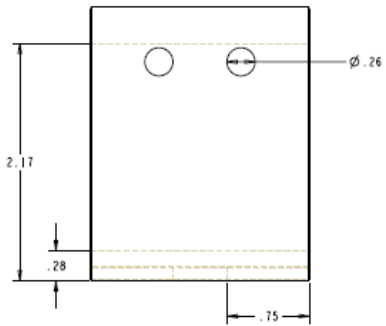
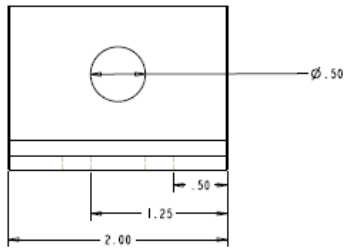
Project:	RoboSub
Part:	Grasp Rel. Mech. Claw 2



FAMU-FSU  
College of Engineering

SCALE 2.000

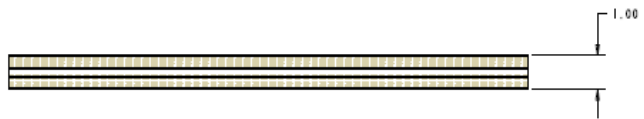
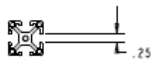
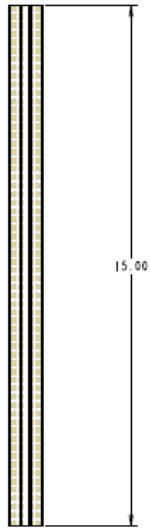
Project:	RoboSub
Part:	Side Support Electronics Rack



FAMU-FSU  
College of Engineering

SCALE 2.000

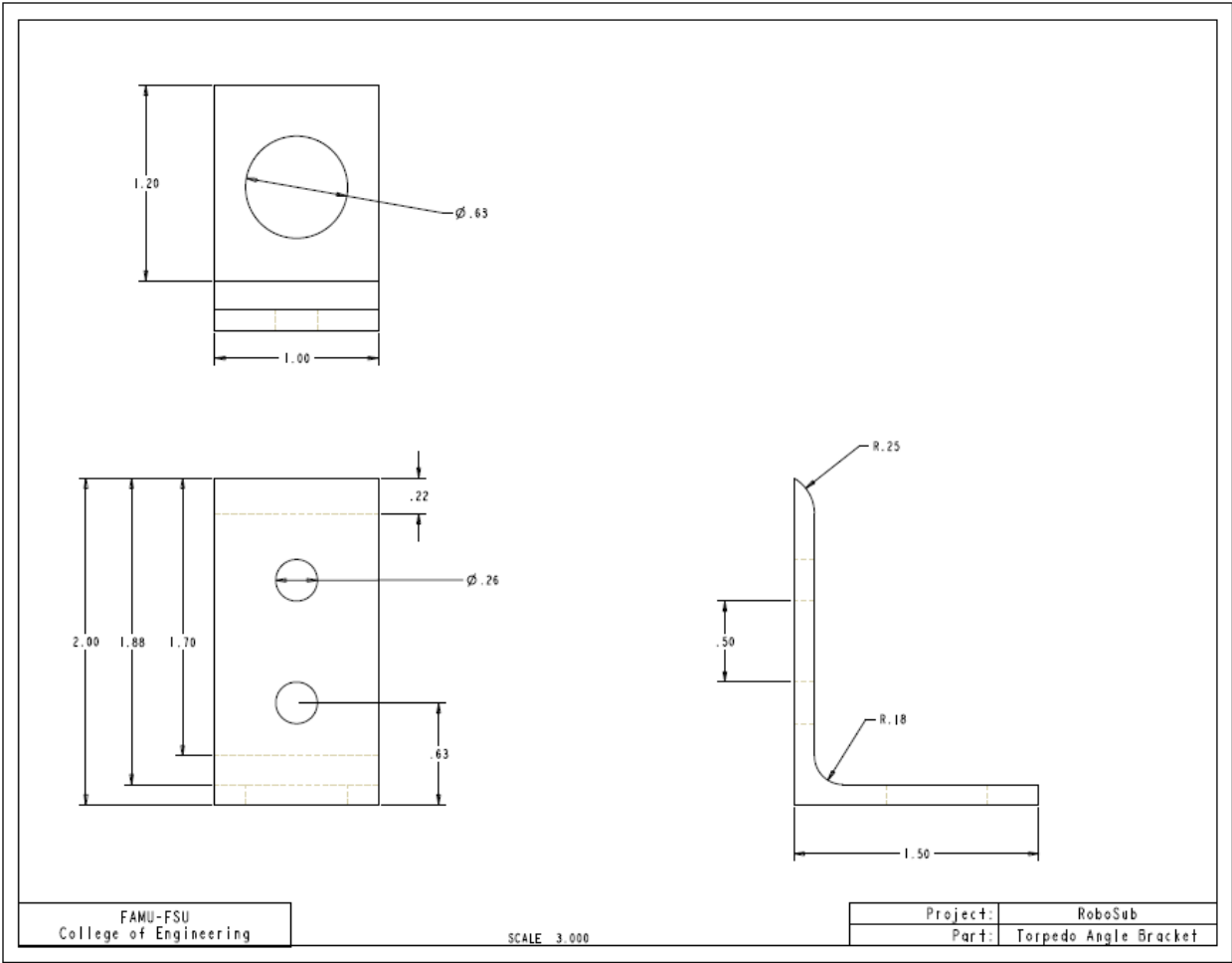
Project:	RoboSub
Part:	Air Cyl. Bracket



FAMU-FSU  
College of Engineering

SCALE 0.633

Project:	RoboSub
Part:	Frame End Piece

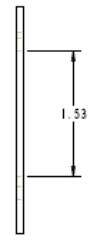
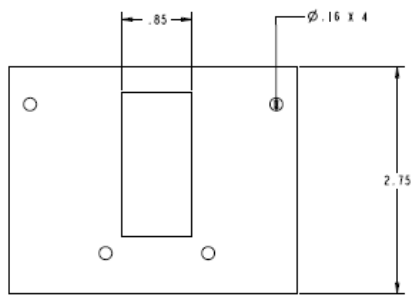
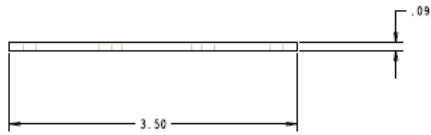


FAMU-FSU  
College of Engineering

SCALE 3.000

Project:	RoboSub
Part:	Torpedo Angle Bracket

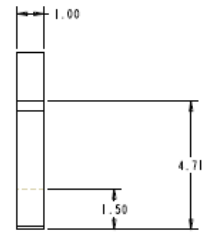
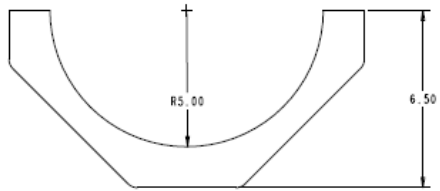
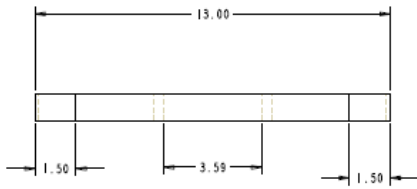




FAMU-FSU  
College of Engineering

SCALE 1.500

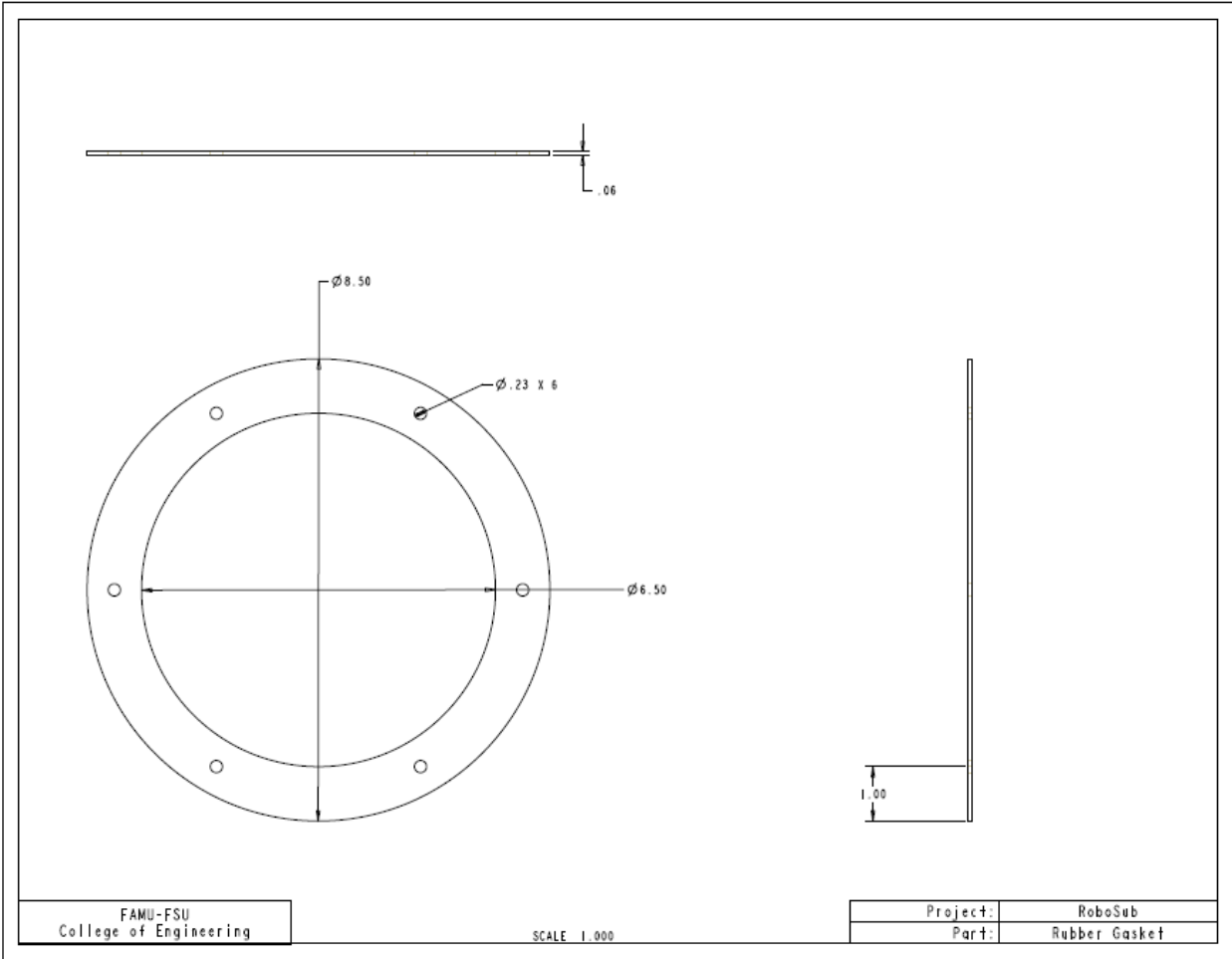
Project:	RoboSub
Part:	Marker Dropper Plate

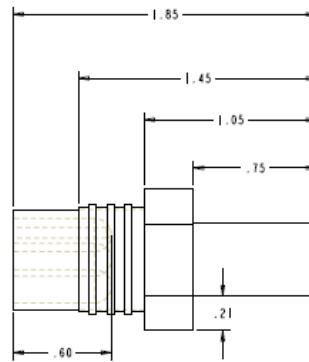
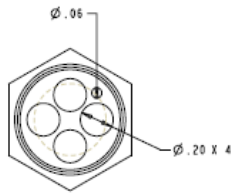
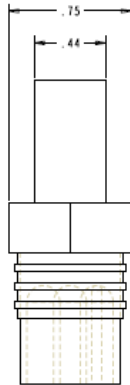


FAMU-FSU  
College of Engineering

SCALE 0.500

Project:	RoboSub
Part:	Pressvessel Support

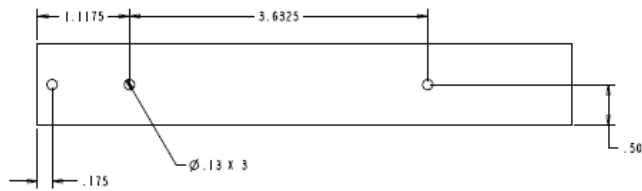
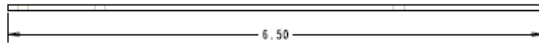




FAMU-FSU  
College of Engineering

SCALE 3.000

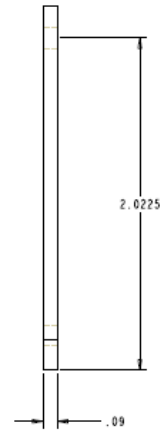
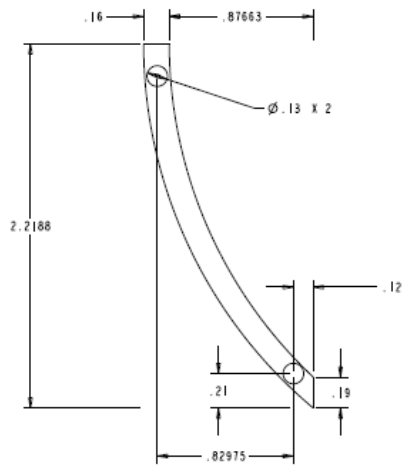
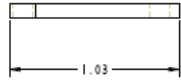
Project:	RoboSub
Part:	Seqcon 4 Pins



FAMU-FSU  
College of Engineering

SCALE 1:500

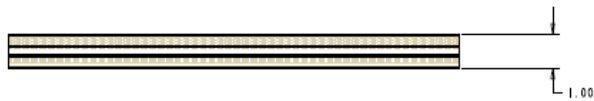
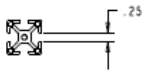
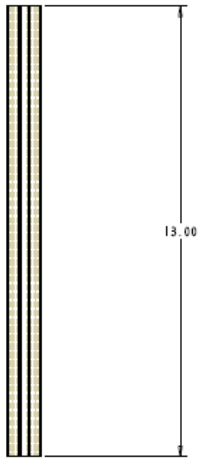
Project:	RoboSub
Part:	Thruster Mounting Plate



FAMU-FSU  
College of Engineering

SCALE 3.000

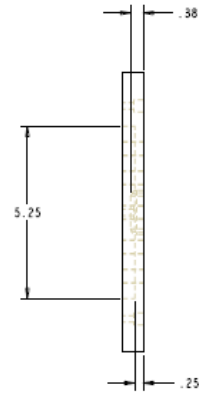
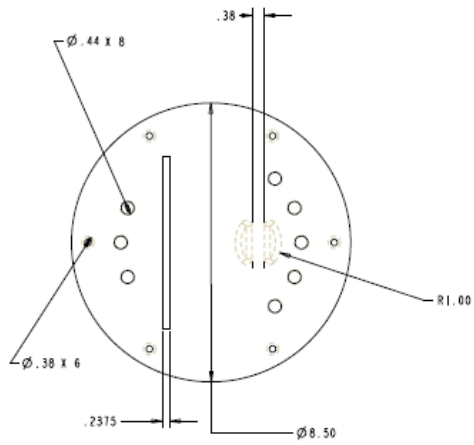
Project:	RoboSub
Part:	Marker Dropper Channel



FAMU-FSU  
College of Engineering

SCALE 0.633

Project:	RoboSub
Part:	Frame Small End piece

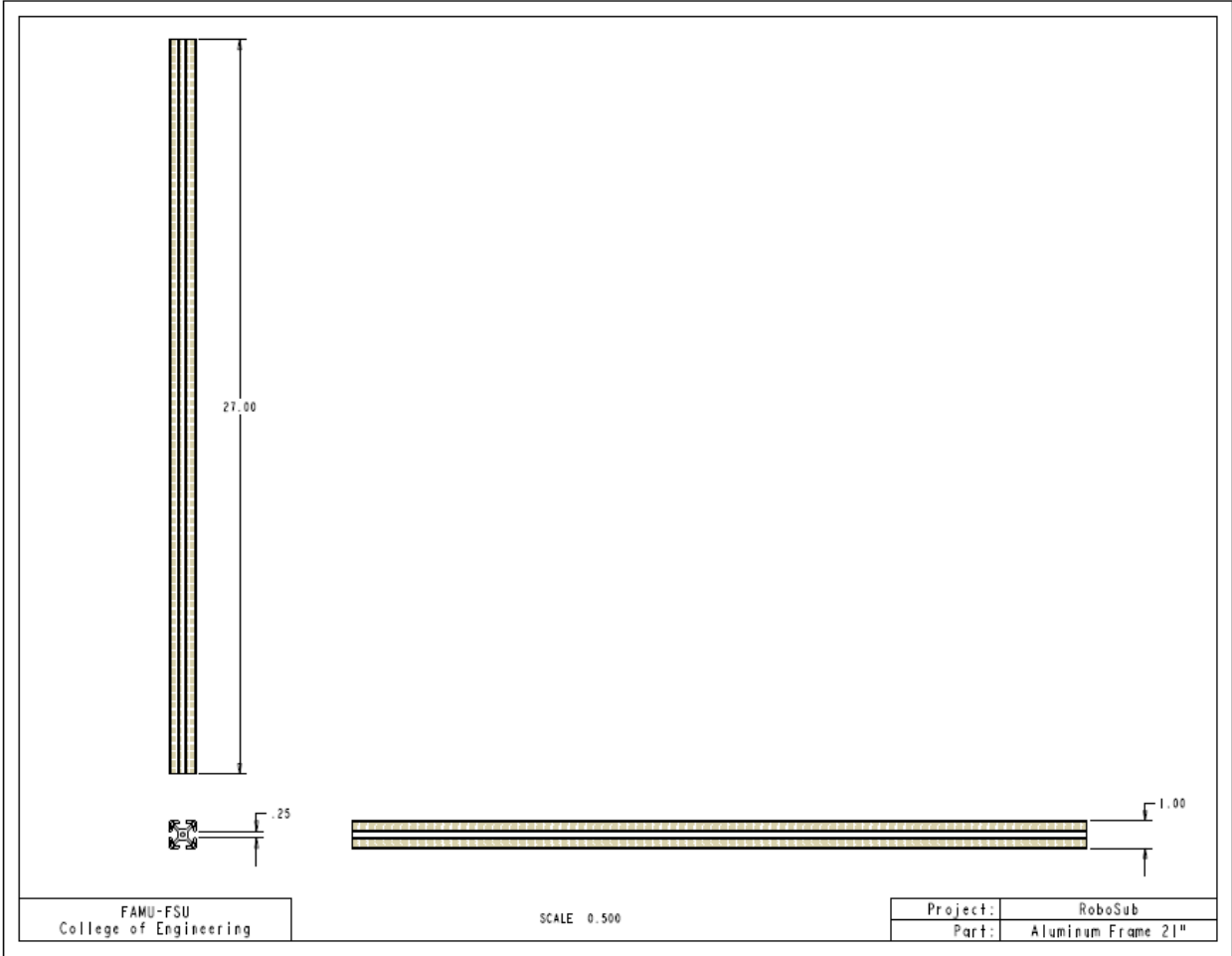


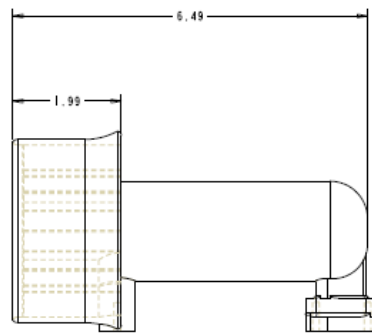
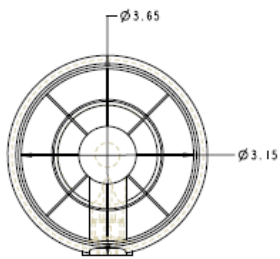
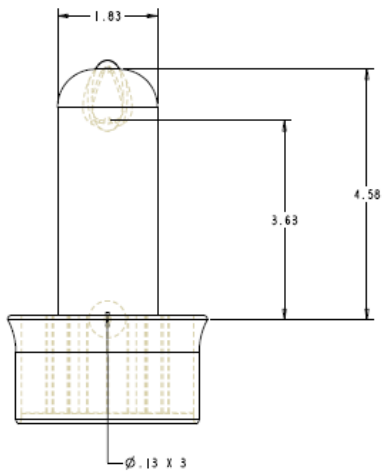
FAMU-FSU  
College of Engineering

SCALE 0.600

Project:	RoboSub
Part:	Back Inside End Cap



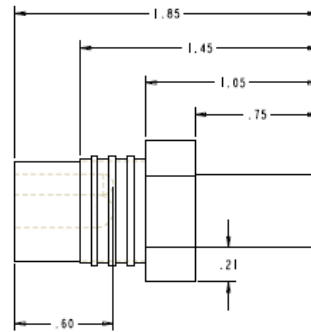
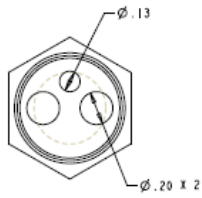
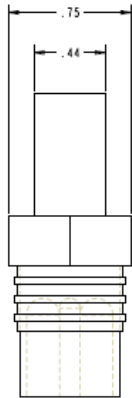




FAMU-FSU  
College of Engineering

SCALE 1.000

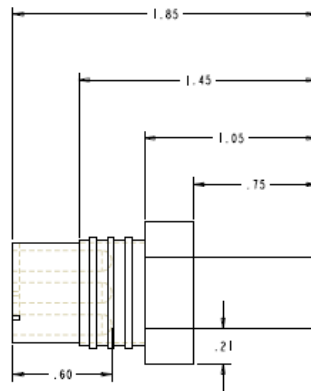
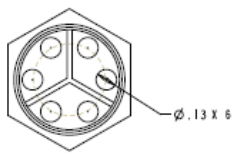
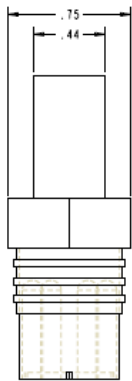
Project:	RoboSub
Part:	Thruster



FAMU-FSU  
College of Engineering

SCALE 3.000

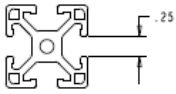
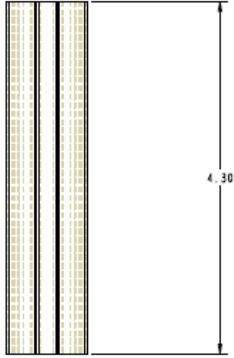
Project:	RoboSub
Part:	Seacon 2 Pins



FAMU-FSU  
College of Engineering

SCALE 3.000

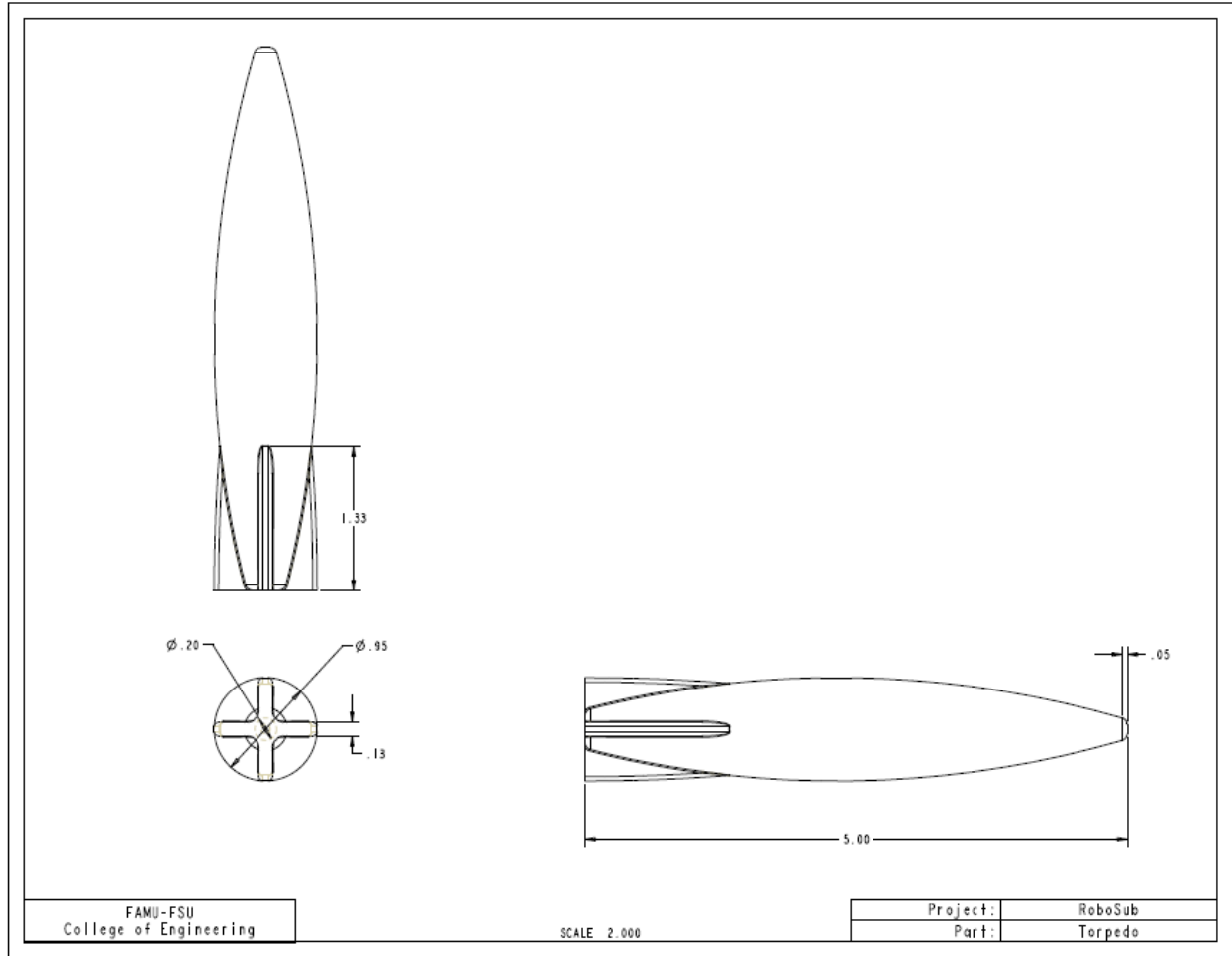
Project:	RoboSub
Part:	Seacon 2 X 6



FAMU-FSU  
College of Engineering

SCALE 1.500

Project:	RoboSub
Part:	Thruster Support Bar



## References

[1] Texas Instruments, "Active Filter Design Techniques" in Op Amps for Everyone, Ch. 16

<<http://focus.ti.com/lit/ml/sloa088/sloa088.pdf>>.

[2] Waterman, Andrew et. al, "APLS – An Acoustic Pinger Location System for Autonomous Underwater Vehicles" (Bachelor's diss., Duke University, 2006). Available at

<[www.eecs.berkeley.edu/~waterman/papers/apls.pdf](http://www.eecs.berkeley.edu/~waterman/papers/apls.pdf)>.

## Biographical Sketches



My name is Tra Hunter, and I am graduating this spring from Florida State University with a Mechanical Engineering degree. I am a Tallahassee native and plan to remain in Tallahassee after graduation. I have worked at the National High Magnetic Field Laboratory for two and half years doing cryogen fills on some of the largest superconducting magnets in the world. Last summer I had the chance to work for a local engineering company who is contributing to the ITER project, which is constructing a large scale fusion reactor in Southern France. Outside of school and work, I enjoy playing sports, working on just about anything, and hanging with friends and family.



My name is Hang Zhang, a senior FSU Computer Engineering student. I will graduate in Spring 2012, and will go to graduate school to pursue a PhD degree in Computer Architecture. I'm interested in programming microcontrollers, micro architecture designs, and high performance and parallel computing. I conducted research on heterogeneous computing and own one patent in high performance material application. My primary responsibilities in the Robosub project include overall software structure design, mission control, and electrical system design.



My name is Eric Sloan, a senior Mechanical Engineering student at Florida State University. I am from Sarasota, FL and look forward to graduating this spring. I will be attending graduate school in the Fall in pursuit of a Masters Degree in Mechanical Engineering with an emphasis on intelligent/mechatronic systems. Outside of academics, I enjoy playing a variety of sports including golf, football, ping pong, bowling, pool, and more—although golf is unequivocally my extracurricular passion.



Welcome to the world of Kashief Simon Moody. I'm from Washington D.C. and I'm a highly motivated senior studying Mechanical Engineering at Florida A & M University. My philosophy in life is play hard,



work harder and put your heart into your endeavors and success will follow. Some of my interests outside of engineering are sports (Basketball, Soccer, Football and Hockey), movies (Mystery, Suspense, Horror and Action), music (Jazz, Hip – Hop and R&B) and community service. I will be graduating spring 2012, prepared to follow my dreams; seeking a position in a company where I can offer my current skills and education as a service, and experience growth and development through the challenges offered by the environment. Also upon graduating, I want give back to community where I grew up; not just money, buy time. In addition I want to start a family and take care of my existing family.



My name is Antony Jepson, and I am a graduating computer engineering senior who attends Florida A&M University – Florida State University College of Engineering. I will be graduating in 2012, shortly before the neutrino apocalypse. I have worked with many organizations and corporations including the STEM Learning Community, the Astro-physics Cosmic Radiation Detector Research and Development Laboratory, Apple, and Microsoft. I enjoy working with and programming micro-controllers. When I'm not pecking at the keyboard, I enjoy spending my time playing sports such as track/field, tennis, and football. Additionally, I relish eating delicious food and hanging out with my friends.