1. *Abstract*

RUMarino started with an agenda more ambitious than ever for this year's competition following the boost in morale and productivity last year's great performance in the RoboSub Competition gave our members. Despite having to deal with a Hurricane, and its prolonged aftermath, the members found a way to thrive in these harsh conditions. At the beginning of the academic year a completely new design of the AUV was created and was being developed, but due to a huge change in circumstances, the team adapted to the situation by deciding to modify our previous AUV model. It had many improvements such as in the mechanical structures, electrical & embedded systems, and software architecture for an even more efficient AUV.

2. *Competition Strategy*

This year's competition strategy was to tackle the same missions with a more robust and effective AUV. This would allow the team to focus on constructing a properly functioning platform that could be later expanded to accomplish more missions. Working on the same missions also allowed the team to show new members the previous system and the errors that were being corrected.

Since last year was the first time RUMarino competed in the RoboSub Competition, the team learned a lot from the experience. Every working team was inspired to implement their fresh ideas as best they could. This was done with certain key quality metrics that the team wanted to achieve; these include: usability, maintainability, reliability, modularity, and expandability. Another goal was to make sure that the team would be able to implement the design in time for testing before the competition.

In general, the Autonomous Architecture and Computer Vision teams upgraded their in-house code by modularizing it so that it could retrofit a node based framework. While they adapted to this year's new missions, the teams took their existing architecture and completely overhauled it by upgrading to the Robotic Operating System (ROS) framework. These changes were meant to enhance the usability and expandability of the system so that in the future most of the code can be reused [1].

The Embedded Systems Team worked with upgrading the existing system while looking out for possible sensing solutions that could work cost effectively within the team's allotted budget. They also upgraded the sensors and actuators so that these could run with the new architecture designed by the Software Development Division's working teams.

The Mechanical Structures Division and Electrical Systems Divisions had a new drastic perspective because of last year's competition. They wanted to design and implement a new mechanical structure from scratch. This would contain a more robust wiring system that would permit easy access to electrical components whilst improving upon previous areas that caused difficulty. These structures would be designed in such a manner that they would be able to keep up with the rapidly developing nature of the team; thus, serving us for many more years.

The team also noticed that it suffered a lot due to general problems with organization, such as not having documentation for recurring tasks like testing the mechanical structure and vision algorithms. This was a potential hazard and slowed us down, taking into account this year's membership growth of 100%. Due to the obvious need for a more organized team, a new division was added: Operations Management. This division is meant to alleviate many of the administrative tasks that hinder the team leaders from focusing on technical development. Thus, the Operations Management Division focuses on the optimization of workspaces and processes, documentation, locating potential sponsors, and making sure the knowledge from previous members is properly passed on to the new ones.

The team had to face immense struggles due to Hurricane Maria that ravaged our home island. This lead to a lot of changes in the plans we had made. Even though the vision for the teams held largely the same, compromises had to be made. We lost our workplace, we lost more than two months of development time, shipping components to the island took at least five weeks to arrive, and we had to balance a semester's worth of class in three months. However, the team members adapted, and this didn't stop us from being ready to this year's competition.

3. *Design Creativity*

The design creativity took flight after the end of the first competition. During the members' time in San Diego, they were exposed to a great amount of new ideas from talking to other team's members and were eager to work them in our future design.

The biggest upgrade this year was set to be the mechanical structures. A new dual hull design, vectored thrusters, better connector placement, and more was to be put in place to make the system more reliable, usable and upgradable. However, due to the setbacks, it could not obtain the necessary materials to manufacture it, and had to adapt. To accomplish this, the team found a way to preserve the core functional points of the new design and implement them reusing most of last year's components.

The use of rapid prototyping helped to keep the structural design simple and creative, while being able to make quick changes when they were necessary. Using the same core design of last year's competition, the AUV was developed as compact as possible, and this time, as light a posible. The 3D printed components are now the main structural element, and four of the aluminum extrusions from Proteus 2.0, are used to add support and stability. These components also add to the aesthetics of the AUV, projecting a new style and a uniform look, thus Proteus 3.0 is born (seen in ***Figure 1***).

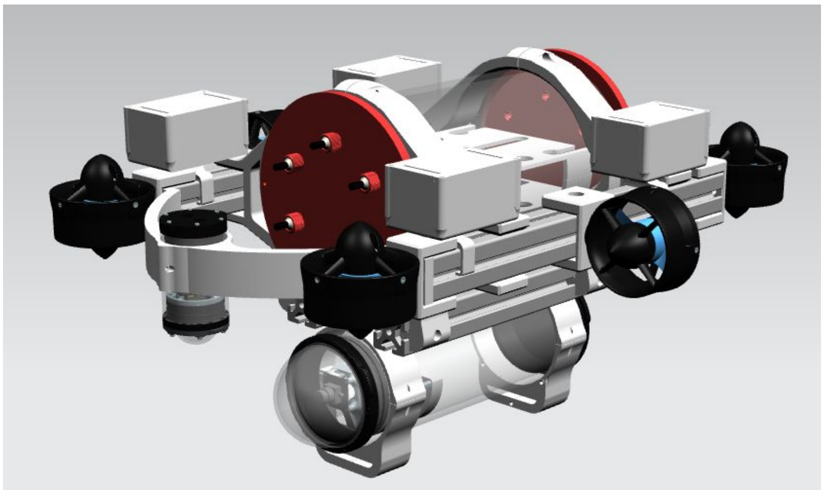

Figure 1: Proteus 3.0

The Electrical Systems Division redesigned the electrical system of the sub to make it more robust, modular, and reliable. The division designed Printed Circuit Boards (PCB) to facilitate the connections between the Arduino and the Electronic Speed Controllers (ESC), and designed another PCB to turn off the motors without cutting power to the electronics. This enables the Software Development Division to continue working in the case that the AUV malfunctions while doing one of the missions. The division also moved the batteries to their own enclosure, complimented by LED voltmeters (*Figure 2*), to facilitate the assembly, as well as monitoring, and recharging process.

Figure 2: Battery Cabin

The high-level code was designed considering the Autonomous Architecture team diverse levels of skill. The team opted to code in Python. This decision arises since Python offers itself as a versatile programming language, providing a simpler learning curve, in comparison to other high-level languages. However, it is still powerful enough to encapsulate the overall necessities needed from the project. Its relative simplicity allows for novice programmers to learn and contribute to the software development, whereas the more senior programmers may benefit from its versatility and be able to produce a more complex code. This allowed the team to add in object-oriented code and designs.

Last year, most of the code was done "in-house". This meant the team members were developing by themselves software that was already available, which slowed progress down. Therefore, the Software Development Division saw the benefits of implementing the popular ROS framework and decided to adopt it. With the adoption of ROS, a new software architecture was made to take advantage of all the functionality that ROS provides [2]. Given that ROS has a relatively high learning curve it took the team some time to get into its mindset, as a result the development of the new architecture took more time than expected. The new architecture was more autonomous in its core, since the architecture starts to be able to account for abnormal function situations, such as not finding an obstacle or getting lost, something that last year's architecture was unable to do. A layer of abstraction was also added to this new architecture which allowed the Mission Code and the Vision Code to work seamlessly with ROS without the team members of those sub teams having to learn about the intricacies of the communication between the two. ROS also permitted to run the controllers, which were originally run on microcontrollers on the main computer. This meant that the microcontrollers would serve as hardware interface boards.

The Autonomous Architecture team also had to adapt to the changing environment of other teams, since changes in their design affects how the missions can play out. This means that they had to be watchful for these changes.

One of the main challenges faced when designing the AUV's vision system was the lack of depth perception because of the use of a single camera as our front and bottom facing inputs [3]. The team worked around this by implementing a feedback loop with the AUV's mission logic controller that made use of the region of interest's centroid coordinate to align with said object. To achieve this, a reasonable set of bounding coordinates was determined through trial and error, where the bounding area was not too big or too small, making the system both precise and stable [4].

Members of the team also had some sparks of inspiration and because of the competition a few members are working on developing novel positioning systems for AUVs using machine learning.

4. *Experimental Results*

Due to many setbacks, the period allocated for testing had to be delayed. It was decided that if we were to compete, we had to be able to at least complete the pre-qualifying mission before the Robosub inscription end date. Even with a preliminary build of the AUV the team was able to complete this ultimatum it set forth. Now that the semester is over more time is being dedicated to quickly implement and test our system.

As of the writing of this paper, the only thoroughly tested code is the embedded systems hardware interface using an arduino with ROS. This means that the communications to the motors and the pressure sensors was tested to function properly.

Currently, the updated mechanical structure is passing its final tests. The new dual hull system caused issues with the stability of the system, since its center of mass and center of buoyancy were too close together. Hence additional testing was needed in order to make sure the system was mechanically stable. After these tests, the controllers will be tested to ensure proper calibration.

In parallel, the Embedded Systems Team is testing and debugging the ROS implementation of the mission controller. This controls the flow of the missions and allows one to make alterations. In the future to easily add and run specific missions on the fly. This is passing preliminary integration tests. Once they pass and the mechanical structure is ready, system wide integration and pool mission test will be done.

5. *Acknowledgements*

After such an adverse academic year due to natural disasters and their consequences, RUMarino would like to thank all of its team members and the faculty members of the Engineering Department of the University of Puerto Rico for their unwavering support. There is a special acknowledgement to all our sponsors who have helped guide us and challenge us to become better; RUMarino would not be here today without their help. From small to large, the team appreciates all the contributions and donations that have led to the team's participation in the upcoming RoboSub 2018 competition. The team would also like to thank all the students who researched in collaboration with the Industrial Affiliate Program (IAP), and excelled within the development of the AUV. The passion and knowledge passed on by these students have left a legacy within RUMarino.

6. *References*

1. Cacace, J. and Joseph, L. (2018). *Mastering ROS for Robotics Programming, Second Edition*. Birmingham: Packt Publishing.
2. Martín Gómez, David & Marín, Pablo & Hussein, Ahmed & de la Escalera, Arturo & Armingol, J.M.. (2016). ROS-based Architecture for Autonomous Intelligent Campus Automobile (iCab). 257-272.
3. Bertozzi, M., Broggi, A. and Fascioli, A. (2000). Vision-based intelligent vehicles: State of the art and perspectives. Robotics and Autonomous Systems, 32(1), pp.1-16.
4. Docs.opencv.org. (2018). OpenCV: Optical Flow. [online] Available at: https://docs.opencv.org/3.4/d7/d8b/tutorial_py_lucas_kanade.html [Accessed 3 Jul. 2018].

Appendix A: Component Specifications

Component	**Vendor**	**Model/Type**	**Specs**	**Cost (if new)**
Frame	McMaster-Carr	T-slot Extrusions	Material: Aluminum 6061 Size: 40 mm	
	Hatchbox	PLA Filament	Diameter: 1.75 mm	$22
Waterproof Housing	BlueRobotics	Watertight Enclosure: Acrylic Tube (4" & 2")	Max Depth: 330 ft ID: 4in & 2in OD: 4.5in & 2.25in Length: 13.15in	4" Series: $183 2" Series: $107
	CrustCrawler Robotics	WaterProof Vessel	Depth Rating: 150 ft ID: 6.5 in OD: 7 in Length: 12.75 in	
Waterproof Connectors	BlueRobotics	Cable Penetrators for 6mm and 8mm Cables	Bolt Threading: M10	
Thrusters	BlueRobotics	T100 Thruster	Max Thrust – Forward: 5.2lbf Operating Voltage: 12V Max Power: 130W Diameter: 3.8 in	$119.00
Motor Control	BlueRobotics	Basic ESC	Voltage: 7-26 V Current: 30 Amps Signal: Pulse-width (PWM) Max Reverse: 1100 µs Stopped: 1500 µs Max Forward: 1900 µs Deadband: 1475-1525 µs	$25.00
High Level Control				
Battery	Venom Power	LiPo 3 Cell Batteries	Capacity: 5000 mAh Voltage: 11.1	

			V	
CPU	A57	Cortex	j	
Internal Comm Network				
External Comm Interface				
Programming Language 1	Python 2/3			
Programming Language 2	C++/Arduino			
Inertial Measurement Unit (IMU)	VectorNav	VN-100	3-axis accelerometers, 3-axis gyros, 3-axis magnetometers, and a 32-bit processor.	
Cameras	BlueRobotics	Low-Light HD USB Camera	Field of View (Horizontal): 80° Field of View (Vertical): 64°	$89.00