This Thing Runs!

Care of:

Monterey Peninsula College Robotics Club

Instructor:

Jeremy Hertzeberg

Team Members:

Ross Williams

Grant McGregor

Lendz Elliott

Karl Pfeiffer

Zach Marshall

Chris Aramkul

Table Of Contents

Abstract:

This ROV was constructed to perform in the 2008 MATE International ROV Competition as an Explorer class Remotely Operated Vehicle. Designed and built by Zach Marshall, Grant McGregor, Karl Pfeiffer, Ross Williams, and Chris Aramkul, the ROV was conceptualized as a PVC and fiberglass constructed hydrodynamic unmanned aquatic vehicle built for efficient measuring of black smoker vents and freeing of OBS instrument packages that were trapped by underwater lava flow. Using a popular three motor model common in the industry, the ROV was circular and flat, with low drag, and maximum vulnerability. The body soon evolved into a eight motor powerhouse, enabling the efficient lifting of heavy weights and maneuverable enough to avoid to common obstacles in aquatic environments. Measuring eighty centimeters wide at the motors and twenty one centimeters tall, the team members prepared a fiberglass hull for the wet-type ROV by casting a mold of plastic, then fabricating a double fiberglass shell. This was bolted onto the ROV to create a shield for the inside motors. After extensive testing in and out of water, the ROV was given a camera, and sent on it's way to the competition!

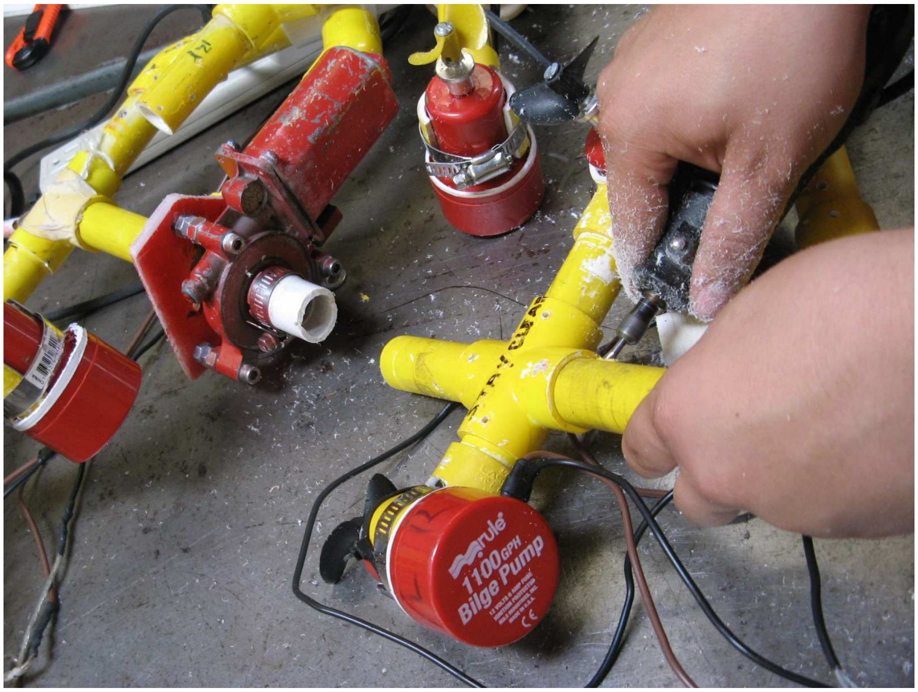


Figure 1

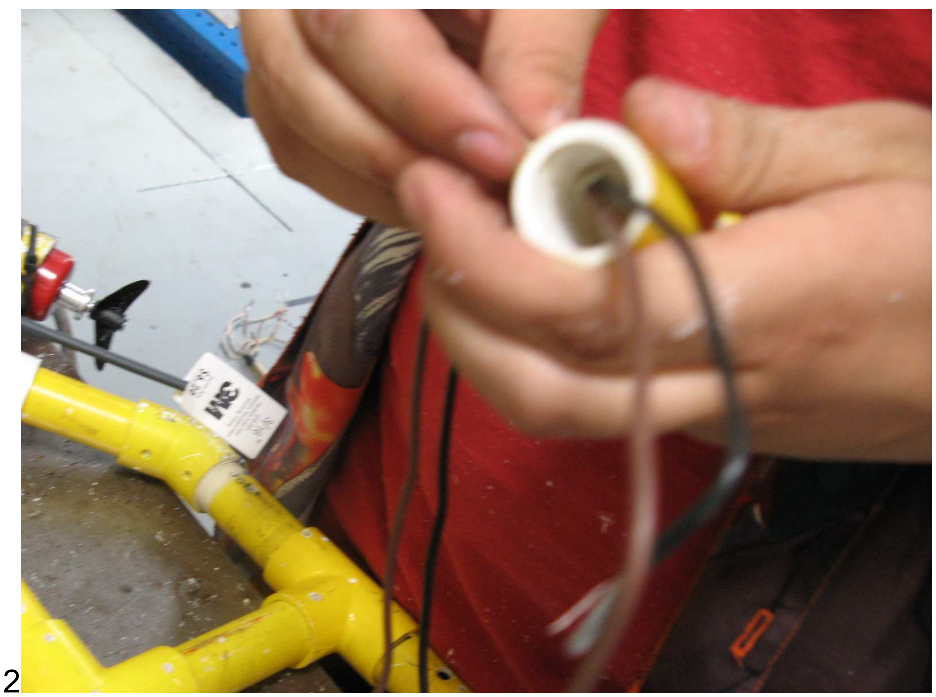


Figure 2

Figure 3

Figure4

Figure 5

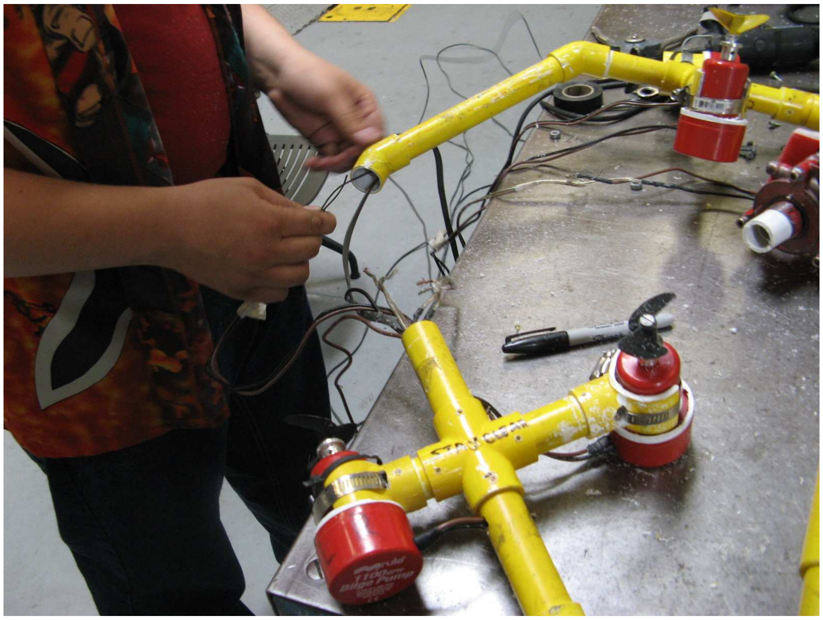


Figure 6

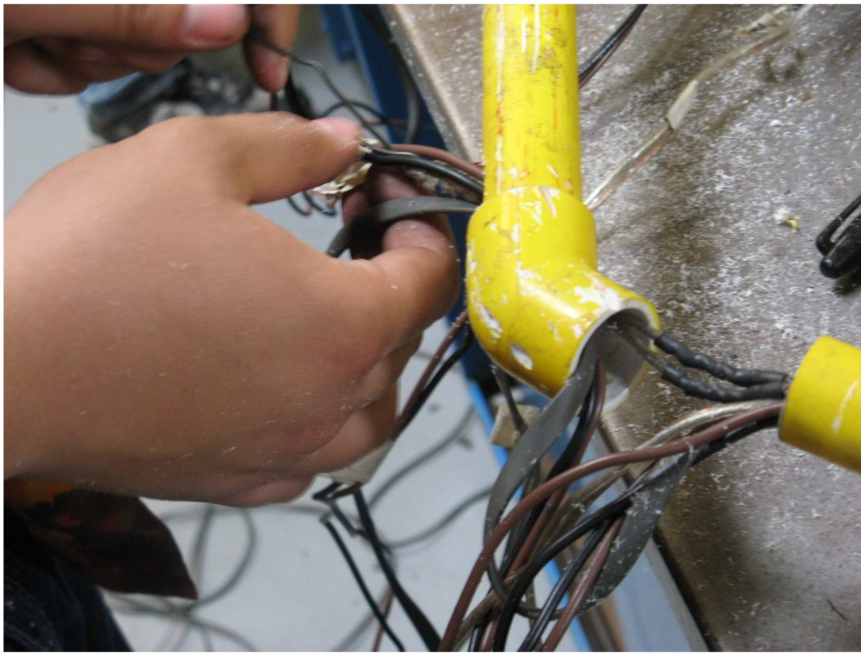


figure7

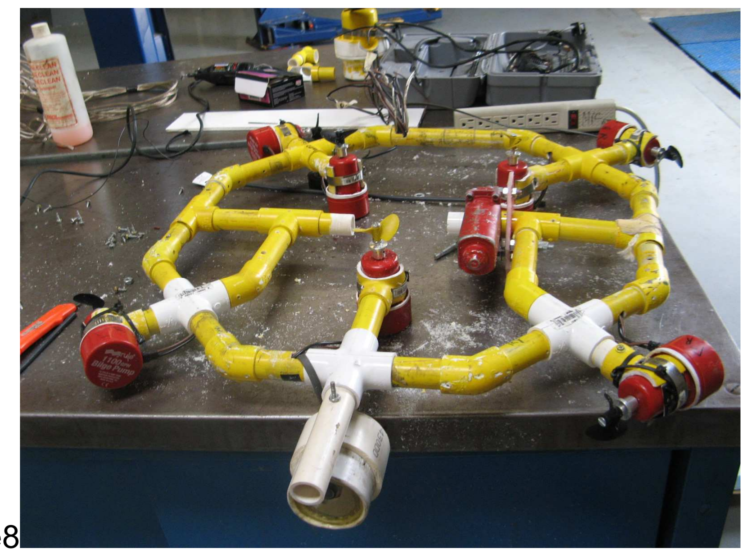


Figure8

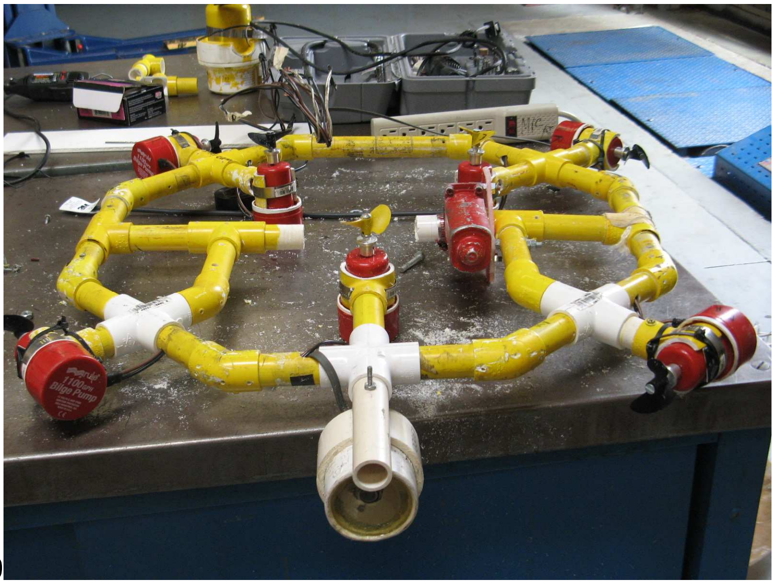


Figure9

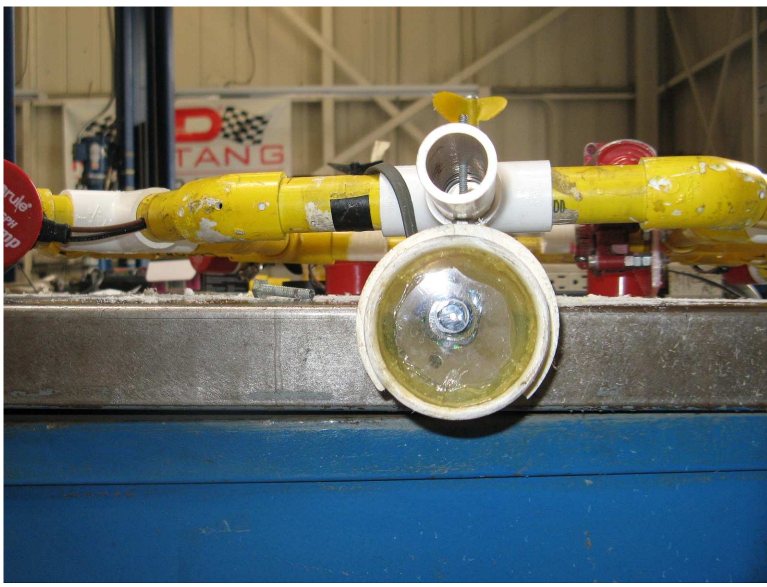
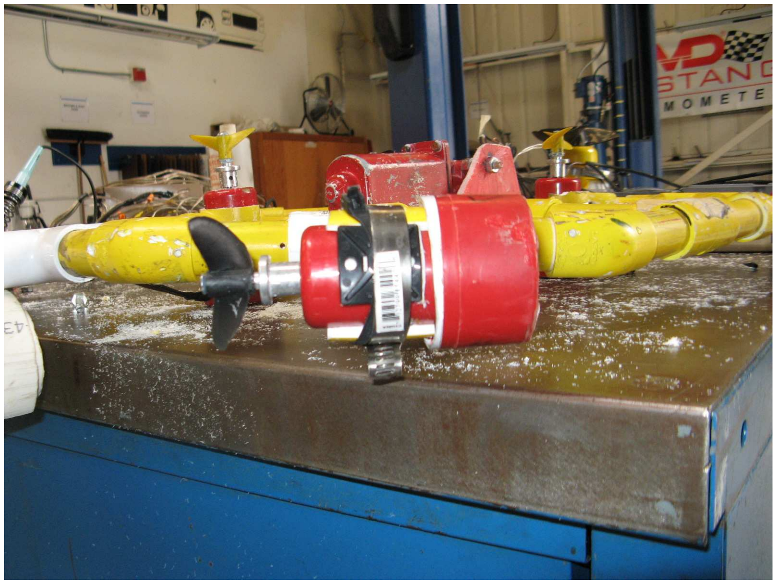


figure10

Figure11


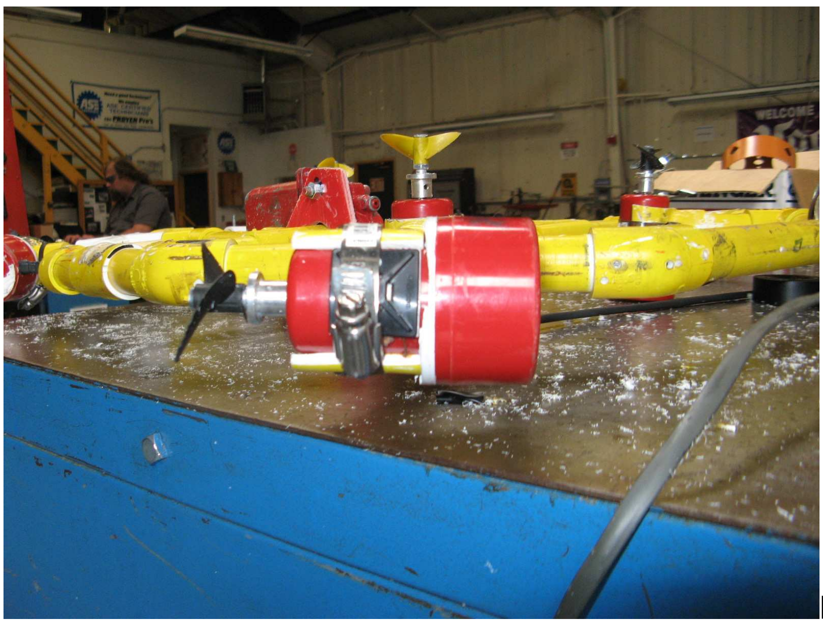
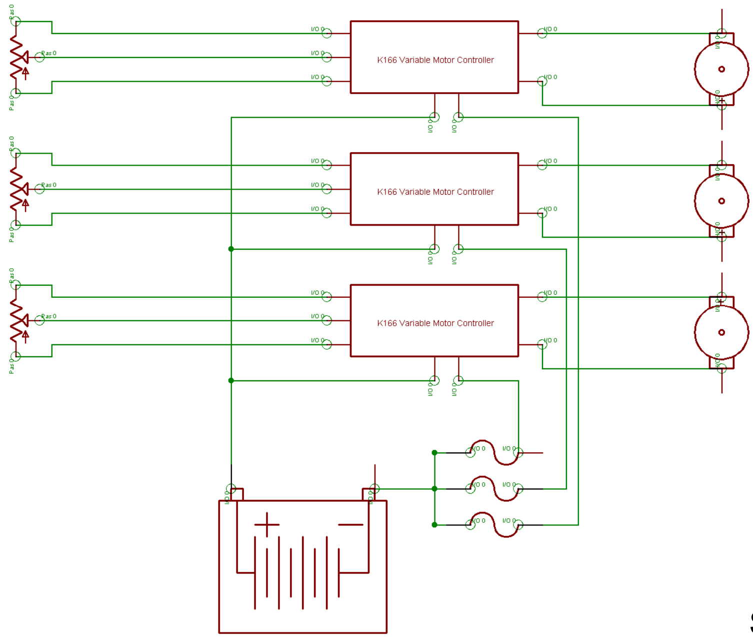


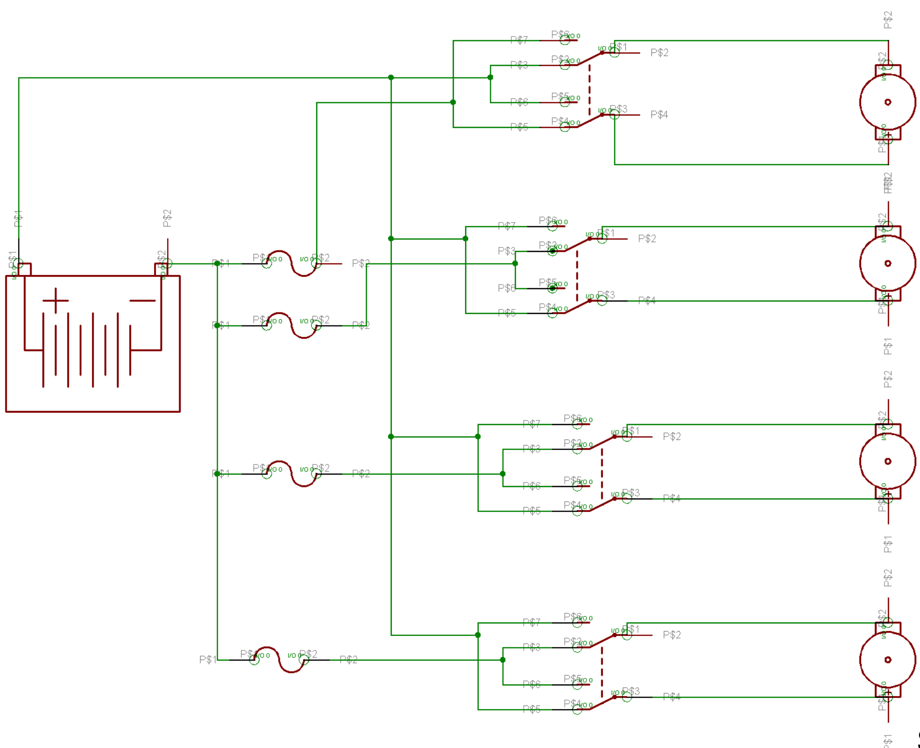
Figure12

Table1:

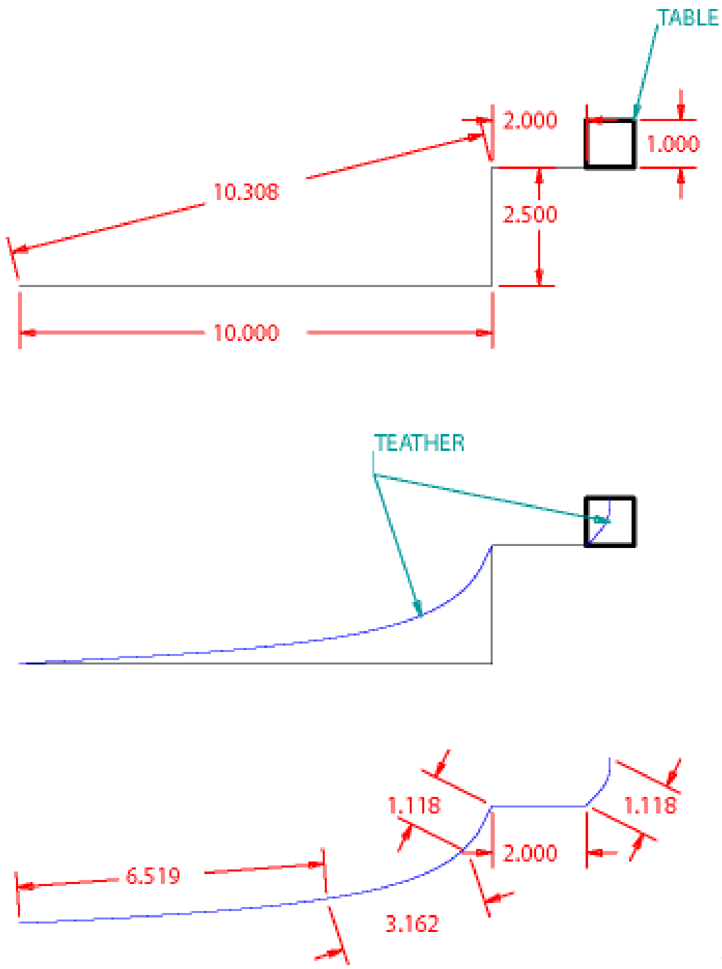
Straight PVC	salvage	34	
45 degree bend PVC	salvage	10	
T intersection PVC	salvage	13	
Cross intersection PVC	salvage	5	
Rule 1100 gph builg pumps	purchace	6	$35
a few 100 ft of stereo wire	salvage	420	
fiberglass cloth	purchace	4	$32
epoxy resin	purchace	1 gal	$55
1/2 inch screws	salvage	46	
1/4" 20 thread nuts	salvage	35	
1/4" 20 thread washers	salvage	18	
1/4" 20 thread rods	salvage	4	
belt clamps	purchace	8	$21
a camera and CAT5 wire	salvage	4	
DPDT switches	salvage	5	
k166 potentiometers motor conrollers	salvage	4	$26
rotory potentiomiters	salvage	3	
plastic box	salvage	2	
fuses	stock	12	
fuse holders	salvage	2	
Heat shrink tubing with meltable inner lining	purchace	2	$6.20
hole saw	purchace	1	$37
propellers	purchace	7	$21
propelller adapters	purchace	7	$35
misc hardware	purchace		$40
			$308



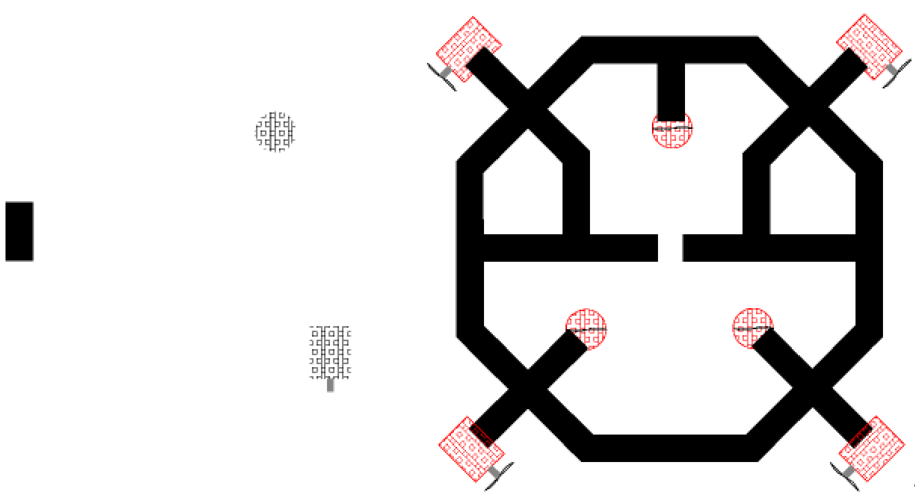
Schematic1



Schematic 2



Schematic3



Schematic4

Design rationale:

This machine was designed with the following purposes in mind:

1. The machine was designed to be operated through a remote command station aboard a theoretical vessel.

2. The machine was designed for deep-sea operations, specifically deep sea trenches formed by underwater volcanic activity.

3. The machine was designed to efficiently sample from around "black smokers", heated water vents found in deep-sea trenches.

4. The machine was designed to also take temperature measurements from the water around the "black smokers".

5. The machine was also designed to release trapped Ocean Bottom Seismometers from lava flow caused by underwater eruptions.

Typical Troubleshooting techniques:

For electrical components, the suspect problem element was isolated from the circuit and tested with a known working complementary system. Wires were tested with a working voltmeter, other components that require it with a working power source. Most of our systems were made modular, so that any broken components could be easily replaced.

For PVC or plastic systems, the protocol was much the same, except for the systems which require gluing; these are replaced with screws wherever possible.

Fiberglass

Materials used:

Snow dish (mold), Fiberglass (woven/unwoven), Protection (gloves/glasses/face masks), Epoxy resin, Bottle (a mixing container), Brushes (spreading epoxy), PVC (stirrer), Scissors, Petroleum jelly

Method:

First came the protection, since inhaling fiberglass is extremely bad since it will never leave the lungs and also it can damage cells. We mixed the resin with the epoxy so that the epoxy can then harden. We then coated the side of the snow dish that we were going to use in petroleum jelly so that the epoxy wouldn't form onto the mold itself. Follow # of times equal to the # of fiberglass layers. Following that, we coated the side with epoxy and attempted to spread it evenly about the inside of the mold. Then we laid the one sheet of the fiberglass onto the epoxy and spread it until the fiberglass had no more white spots, which were spots that the fiberglass didn't have enough epoxy. Then after the final layer was laid, a finishing layer of epoxy was laid to cover the fiberglass. Finally we brought the project inside and left it for the weekend.

Observations:

Using the inside of the mold allows smooth outside, and vice versa.

Using a brush that already was used in a first batch of epoxy Will increase the rate of hardening of the second batch.

If the epoxy was too thin to soak into the fiberglass, more epoxy would do the trick.

Epoxy works with resin to make plastic, which creates heat, and the more heat the faster it sets.

What we learned was that fiberglass is strong, though it gets ridged when it's mixed with epoxy and resin. Also that it will dull drill bits when drilling holes into the part.

One of the greatest challenges in making the ROV is trying to have Atmel's TinyAVR microcontrollers send and receive data from a pc. The decision to use AVRs was to expand the knowledge of different micro controllers besides pics and find a inexpensive way to control the movement and devices on the ROV. However, a number of factors slowed down or halted the process of developing a system that would have controlled the motors and several sensors via these AVRs. These problems were learning how to programming the AVR without a demo board, learning how to program without a book, and having getting the right parts to setup the AVR to operate not knowing if anything is broken.

The first hurtle was getting the AVR to be programmed, which involved reading multiple tutorials online, downloading a few programs, and buying a few parts like resistors, wires, parallel port cable, ect. The first major problem came when trying to upload data the micro controller from the parallel port. I first used WinAVR, (an open source group of tools to compile and upload data to AVR microcontollers), which couldn‘t upload the files in hex or binary to the AVR, so I used which didn’t work. It was only until an update to WinAVR that allowed the team to program the micro controller.

A second major hurtle is finding tutorials online on how to program the AVR in embedded C. The choice to use embedded C was due one the only person assigned the write code for the AVR was more familiar to the C environment. Yet, the choice to use online tutorials instead of ordering a book form online was use reduce the amount of cost to develop for the AVR, due to budget reasons. Though it didn’t halt the programming the AVR it did slow down creating code and understanding what the embedded C code did. After the competition is over and there is still money left in the budget, if possible a book on how to program the AVR in embedded C should be purchased incase future teams want to use the AVR micro controllers already purchased.

The biggest obstacle is trying to get a serial interface to send information to a AVR micro controller and have the AVR send information back to the pc. The first problem was getting a working USB to serial interface, which was problematic when the purchased USB to serial interface didn’t read as a communications port when connected the team programmer’s pc. So far a possible working USB to serial interface has been found, but hasn’t been tested to see if the interface is properly working. The other problem is not having information being sent to and from the AVR, this could be one of several factors. One is that there’s something is wrong with the MAX232, (converts the positive and negative voltage from the serial port on the pc to zero to five voltage on the micro controller), there’s a possibility that the MAX232 or how the chip is assembled is causing the problem. Two the AVR microcontroller isn't receiving or able to send code, what is known so far is the AVR hsan't picked up any code being sent from the PC, but the AVR completes other tasks while waiting for information. Finally, three is not knowing if the serial interface is working properly.   In order to find out if it is the serial interface, a feedback loop is needed to be made which hasn't been done as of yet.

Since the problem has not been found why AVRs aren't communicating, the method of controlling the motors currently is by using motor controllers that use potentiometers the determine the speed and direction of the motors. So far the motor controllers have been successful in controlling the motors.

Karl:

My fondest memory of the competition will always be looking back and realizing that most of the competition did not consist of thinking, but doing. Being a computer-bound fluorescent cave dweller, I found that it was more than an enjoyable experience was simply putting my hands on tools and pulling things apart, not to mention putting them together. I learned a lot about fluid dynamics, ROV design, detached and unmanned systems, electrical engineering, time management, but more importantly, I applied these skills while still in the process of learning them. The thrill of both building something, and proving to myself that I was able to us some of the things that I learned from countless hours of Wikipedia surfing was unforgettable.