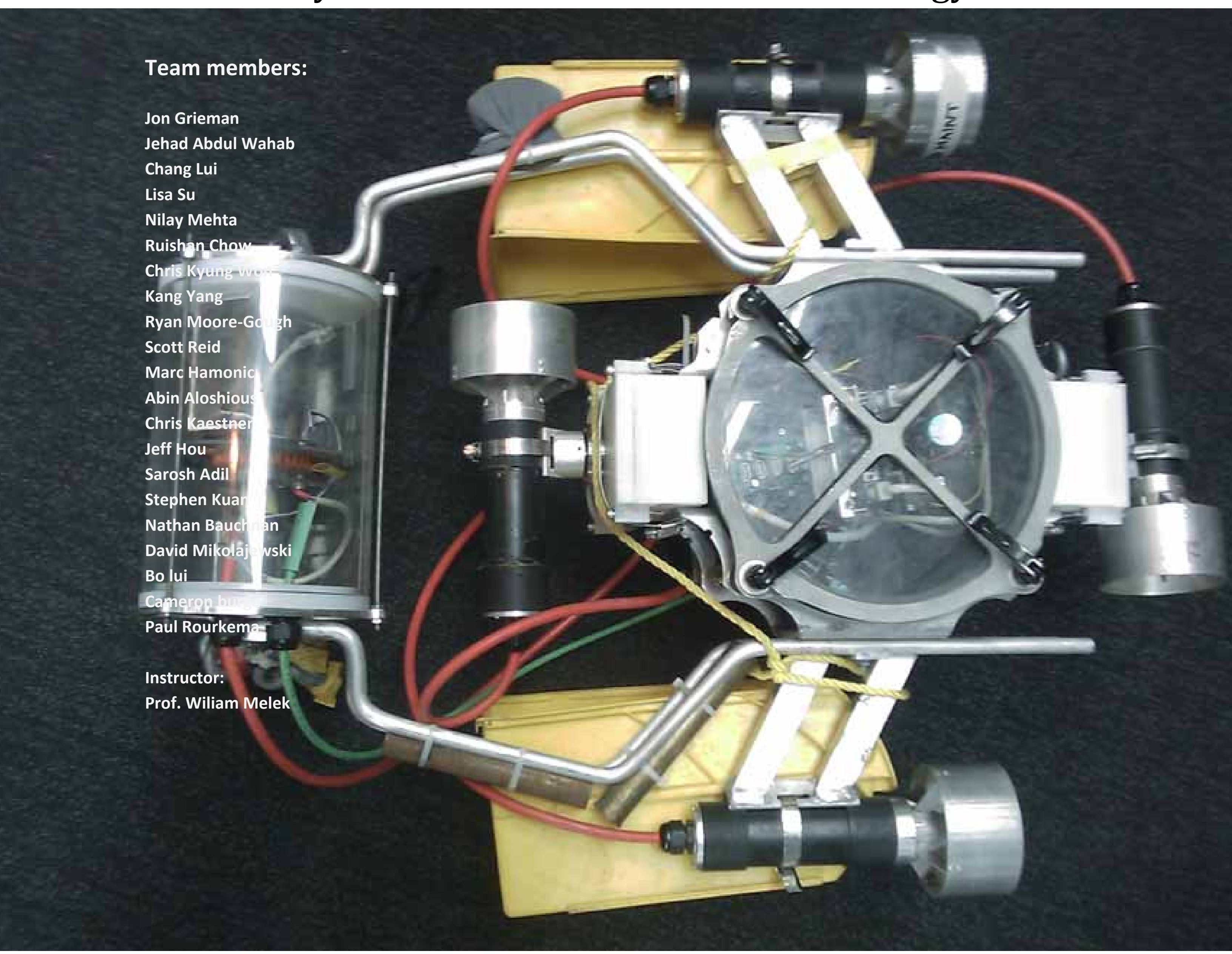
SUBrina

University of Waterloo Underwater Technology Team



Team members:

Jon Grieman
Jehad Abdul Wahab
Chang Lui
Lisa Su
Nilay Mehta
Ruishan Chow
Chris Kyung Won
Kang Yang
Ryan Moore-Gough
Scott Reid
Marc Hamonic
Abin Aloshious
Chris Kaestner
Jeff Hou
Sarosh Adil
Stephen Kuan
Nathan Bauchman
David Mikolajewski
Bo lui
Cameron burge
Paul Rourkema

Instructor:
Prof. Wiliam Melek

Table of Contents

List of Figures

Abstract

abstract

3.0 Introduction

The 2009 Marine Advanced Technology Education (MATE) competition aims to help students throughout secondary and post secondary institutions gain practical experience and develop problem solving, critical thinking and technical skills [1]. The University of Waterloo Underwater Technology Team (uw2tt) has prepared an entry for the 2009 MATE ROV competition named “SUBrina” hereby referring to “Submersible: Remote Intelligent Nautical Autonomy”. Our Remotely Operated Vehicle (ROV) aims to meet competition requirements, and prove able in completing mission tasks.

Problem Summary

The 2009 competition intends to be a simulative environment for manned submersible rescue, in which an ROV performs rescue of a manned submersible. A simulation structure built of milk crates and PVC / ABS plumbing pipe is required to be manipulated and observed by the competing ROV.

Brief Summary of Tasks

The ROV in question is required to perform several tasks to prove competency.

Task 1: Survey submarine for damage

Task 2: Transfer supply pods into submersible

Task 3: Replenish submersible’s air supply

Task 4: Mate with submersible and retrieve the crew

3.1 Problem Relevancy

In order to determine if the competition simulation is realistic, information regarding modern day submersible rescue systems must be investigated.

NATO Submarine Rescue System (NSRS)

The NSRS is a new European rescue system that is the equivalent of the United States Submarine Rescue Diving Recompression System. The system consists of a standard remote operated vehicle (IROV), a rescue submarine (SRV), a portable launch and recovery system (PLARS), and a transfer under pressure system (TUP) [2]. The entire system is fascinating because of the design philosophy behind it. Each subsystem is designed to be operated independently. Yet, systems are able to work in coordination with each other, depending on the situation of the vehicle in distress. In addition, the systems are designed to be easy to transport: the SRV and ROV can be easily transported in many common military transport aircraft as well as by boat and over land. Together with the United States

Navy's submarine rescue team, the two groups can provide complete worldwide rescue services within a matter of hours. The current systems that are used for this service have been enormously successful, and have been deployed in response to several international submarine accidents.

When designing our current ROV, we also tackled the transportation issue as a major design requirement. The 2006-2008 vehicle looked very sophisticated, but transporting the ROV without disassembling the entire thing was cumbersome and difficult. The new vehicle is designed to be taken apart and reassembled in a matter of minutes. We will be releasing a video of a complete assembly via our website soon: we estimate that with all the parts at hand, we can assemble our new vehicle in the space of 30 minutes or less.

Figure 1- Part of the NSRS[4]

The transfer under pressure system (TUP) is used for safely bringing the rescued submariners to the surface. To maintain seals at the huge depths that many submarines operate, the atmospheric pressure inside the vehicle is raised significantly. The problem is that in a high-pressure environment, more nitrogen becomes dissolved in the blood. When the pressure on a person's body rapidly decreases, the nitrogen is no longer as soluble, and so can form bubbles in a person's bloodstream. This condition is known as “the bends” [3] . The TUP is able to maintain the internal pressure at 6 Bar, and together with additional modules on board individual mariners can be brought back from significantly higher pressures. The most unique element of the TUP is the fact that it, like all other elements of the NSRS is designed to be transported and deployed anywhere in the world in under 72 hours. The entire system is sufficiently modular and self contained and may be airlifted to anywhere in the world, secured to a ship's deck an placed into commission in even the most severe conditions.

One of the key design aspects that is shared between our system and the NSRS is the focus on deployability and modularity. In contrast to our previous design which was bulky and difficult to transport, our new vehicle is smaller and lighter, in addition to being designed to be easy to break down if necessary for transport.

The entire system is extremely sophisticated, and uses some very advanced technologies. It can literally be said to be the Rolls Royce of submarine rescue systems since the system is currently being designed and manufactured under license by Rolls Royce for a consortium consisting of Norway, Britain and France.

It is evident that the competition tasks present themselves as reasonably scaled-down tasks which could be encountered upon an actual submersible rescue mission.

3.2 Design Rationale

The former uw2tt design focused on the implementation of innovative ideas. Due to the focus on innovation, the time available to design for ease of assembly and maintainability was minimized. The result was a highly innovative system which functioned poorly, and was unreliable in a competition setting.

Thus, the focus of the uw2tt 2009 design, "SUBrina", is to realize a long-term, maintainable system that is easy to upgrade. Modularity and ease of assembly was the focus of every tier of the design. The following goals were set for the 2009 system:

The Design is to be:

- a simplistic, commercial grade design
- meet all competition requirements
- upgradeable to a working class AUV
- durable, robust and easy to maintain
- prove practical in real world application

4.0 Preliminary Design

The greatest leaps in design are made in the preliminary stages. This is where constraints are set and where art meets function and mechanical mock ups.

4.1 Mechanical design concepts

One mechanical design concept preceded the final design, though the final design has evolved over the design cycle.

Design Draft One

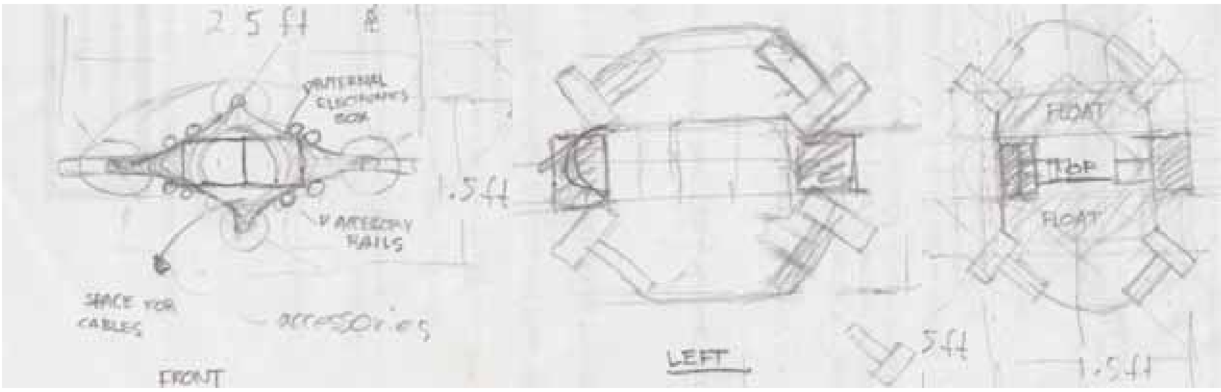


Figure 2- Design Draft One Sketch

The first concept of the 2009 design incorporated ideas from both the previous year's design, along with concepts from vehicles currently on the ROV market. The design was innovative since it used a two-plane vectored thrust configuration. Most ROVs use a thruster configuration named vectored thrust where thrusters (or propellers) are statically mounted at varying angles to the desired direction of motion. Such a design is evident on the SeaEye Falcon ROV [5]. Our design concept incorporated vectored thrust on the horizontal plane (top view) as well as the right plane (side view). From experience flying ROVs of various manufacturers, keeping the ROV level while grasping or carrying objects in a manipulator is a difficult task. When carrying a heavy object in the arm of an ROV, the pilot is usually forced to fly an ROV which is tilted forward. By incorporating vectored thrust on the right plane (side view) the pilot has the ability to level the ROV and thus be able to compensate for any load in a gripper.

Design Draft Two

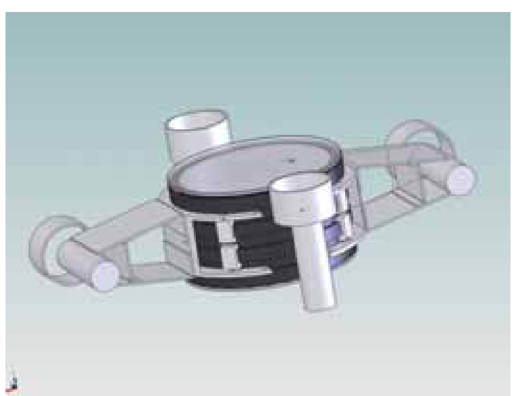


Figure 3- Design Draft 2 CAD sketch

Since the 2009 vehicle is designed to be an Autonomous Underwater Vehicle (AUV) as well as a Remotely Operated Vehicle (ROV), it must optimize its energy consumption due to the limited amount of onboard batteries it may be required to carry. Although vectored thrust is very effective with regards to vehicle stability, it wastes energy. Because the thrusters are mounted at varying angles to each other, there are usually components from each thruster that cancel each other. In order to suppress this property, a new thruster configuration was posed. The new thruster configuration uses only 4 thrusters, reducing power consumption. The new configuration incorporates two statically mounted forward facing thrusters, and two thrusters which can be rotated from the vertical to horizontal positions. This allows for the payload correction previously discussed, and is effective since the vehicle is able to move up and left simultaneously by directing the rotation of the two thrusters.

4.2 Thruster controller board

While mechanical design was under way, software and electrical design concepts were in the works. Of our electrical system, one of the more complicated boards to design was the thruster motor controller board. Due to the space requirements and desired features, it proved difficult to arrive at both the schematic, and component layout. Another wish was to re-use the thruster motor controller board to control the motors located within the manipulator. Several prototypes were constructed with limited success in the early design phases.

5.0 Final Design

The final design is a process of continuous improvement. We are constantly seeking ways of improving the design, optimizing and tweaking the subsystems. The following is a description of our design from the three disciplines which collaborated on the design of the vehicle.

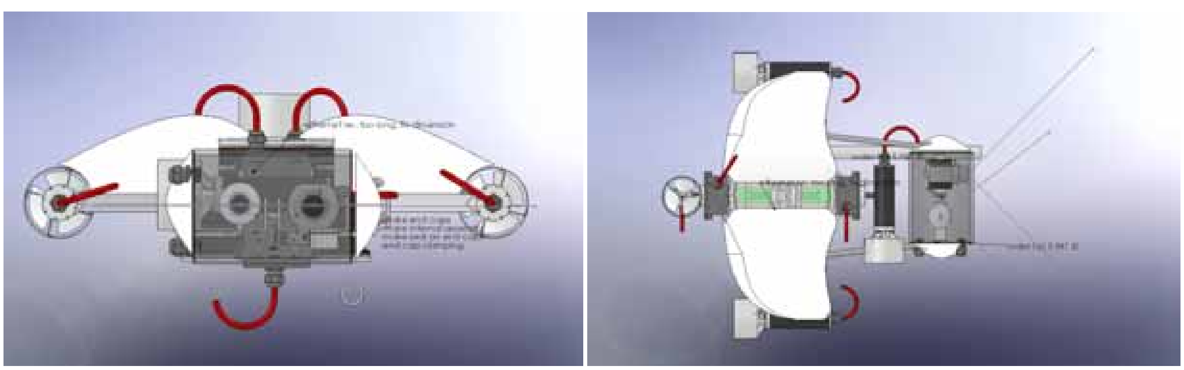


Figure 4- Final Design Front and Top View

5.1 Mechanical

The final design implements the various required components including thruster rotators, cameras and electronics arrangements. Several revisions of the final design have been made since the above images were captured.

Arm/Manipulator

Our manipulator builds on the design by Seabotix on their LBV ROV. Rather than purchase the LBV manipulator at its high price, we have assigned it as a 4th year design project. Our arm will have upgrades such as having quick-detachable grippers that can be swapped without tools. We are also targeting to complete an upgrade which incorporates a rotating wrist in the design. a rotating wrist allows the pilot to perform tasks which are usually difficult otherwise. Designing a wrist that can continuously rotate allows for tasks such as opening valves and even removing nuts and bolts in conjunction with a screw driver.

Thruster Rotators

The rotators are the devices that move the center thrusters from the horizontal to vertical position. The primary purpose of the rotator is to conserve power. Pointing the thruster in the direction of intended thrust does not cause canceled out thrust components as in vectored thrust. Initially it was believed that one of the rotator devices would have less than half of the mass of a thruster, which would further justify it as a device that would increase the power to weight ratio, when compared to the necessary quantity of thrusters needed to achieve vectored thrust. However, a rotator is significantly heavier than a thruster. Increased power efficiency is justifiable as a positive attribute of using rotators in our thruster configuration. DC stepper motors are implemented as brushless DC motors. Encoders allow for precise control of the thruster angle. The gearing within the rotator is set up so that the motors are not required to apply a holding torque in order to maintain a particular output shaft angle, reducing power consumption.

Thrusters

Due to the new nodal approach of our system, the thrusters needed to be stand-alone units which tie into the i2c Bus. Motor drivers are now implemented within the thruster. Having a motor driver within each thruster requires additional space within the thruster, and as such, the housings of the thrusters are extended. In order to reduce the weight and cost of the thruster, ABS was used as a material for the new thruster housings rather than Aluminum. This change in material is one of the reasons why the designed depth rating of our vehicle has been reduced from a potential 150 meters to a shallower 60 meters

Camera Housing

Our video and lighting equipment is housed within a transparent acrylic tube. Our previous vehicle implemented pan and tilt control of the video camera. Pan is less useful than tilt since it is seldom required that the vehicle remain stationary while a view towards the left or right is desired. Tilt is a useful feature to include, since having a view of the manipulator permanently within sight results in an obstruction in the field of view. Including tilt allows for removal of the manipulator from the field of view for ease of ROV flight. in addition, video correction can be implemented if the vehicle is required to tilt forward in order to pick up an object.

5.2 Electrical

One of the design goals this year was to keep power usage to a minimum. Efficiency was particularly important since in the future we would like to convert our vehicle to operate autonomously which will require the vehicle to run on batteries.

Last year the umbilical hampered the vehicles movement because of the large gauge wires used to carry power to the bottom side. In previous years the power carrying conductors were specified based on the 40 amp maximum allowable current draw. This year conservative power estimates were made for each of the devices on the bottom side and the respective powers at 48 volts were summed.

The calculations showed that, even with all of the motors stalled, our system would require less than half of the 1920 watts allowed. These power calculations justified the use of 14 gauge power conductors in our umbilical which will allow the 2009 vehicle to be much more agile. The 14 gauge conductors will allow our system to draw 20 amps which allows for a significant safety factor.

5.2.1 ROV (bottom side) Electronics

The bottom side electronics are dominated by the embedded computer and Ethernet switch, which form the core of the can electronics stack. The embedded computer is a VIA Epia PX10000, equipped with a C7 processor, allowing us to run standard x86 code. The computer, along with all other electronics loads, are powered by an isolated 48 V to 12 V converter, providing up to 10 A for the electronics loads. The embedded computer uses a custom built level shifter board to interface with the internal backplane and distribute I2C and 12 V power to the internal modules. The 7.5 V Ethernet switch supply also draws power off the internal backplane.

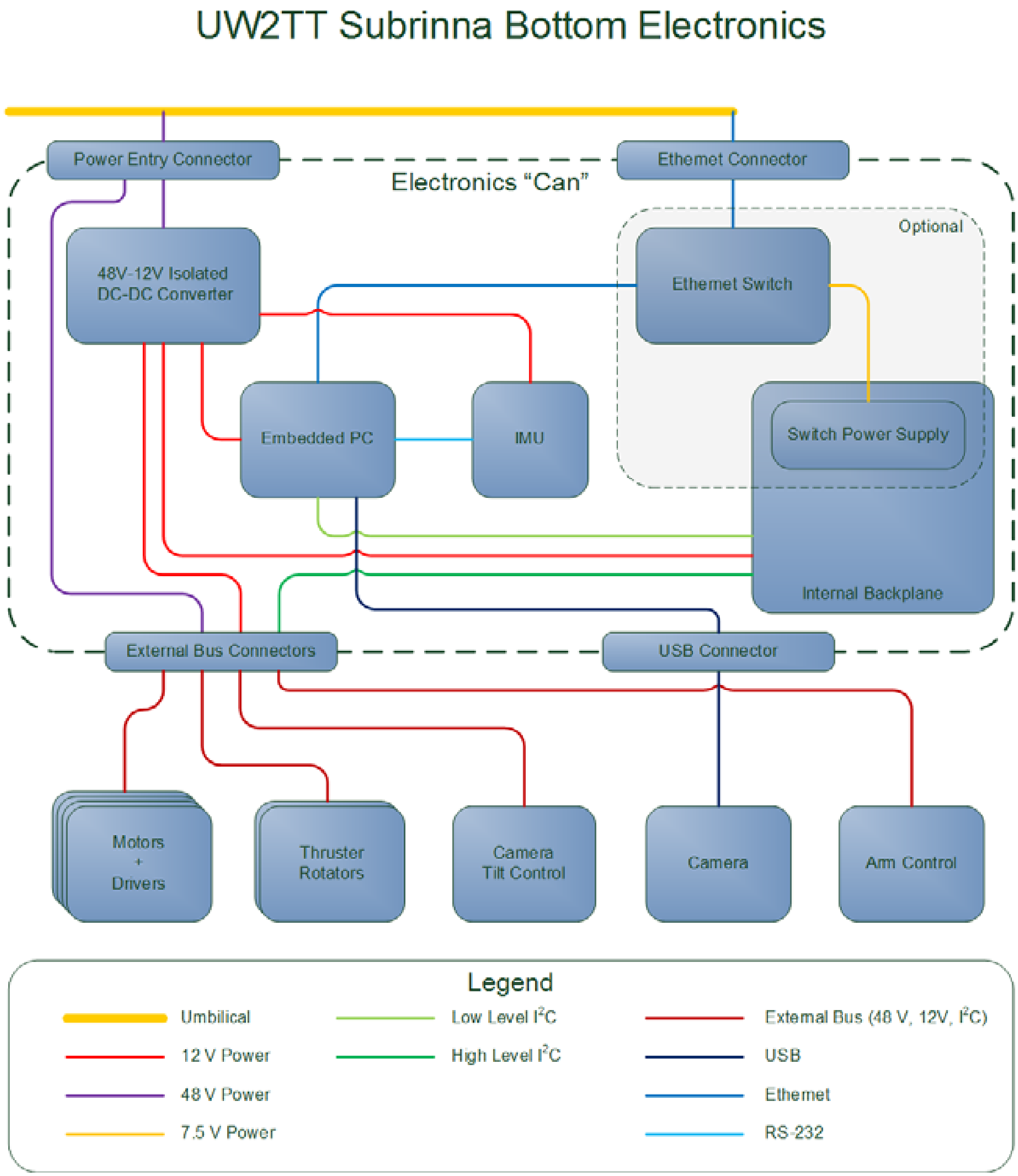


Figure 5- Bottom Side Electronics System

Originally we designed for an Ethernet Camera, which necessitated the Ethernet Switch. Due to our decision to use a USB camera, the Ethernet switch is optional and may be bypassed, but remains inside the vehicle in anticipation of additional networked hardware. The embedded computer also interfaces directly to an Inertial Measurement Unit (IMU), a

3DM-GX1, which provides navigation data and participates in the stabilization of the robot. The external bus carries a level shifted I2C bus as well as a 12 V supply for the electronics modules and a 48V supply for motors.

External Vehicle Network (ExtBus) Devices

On the external bus, the motor drivers are located inside each motor housing. This helps isolate the Electro-Magnetic Interference from each motor driver by moving high-voltage switching circuits as far away from communications as possible. These drivers are also designed to operate from a 48 V power supply but maintain the operation of the motor safely within the 24 V operating region through the use of hardware and software current limiters. This allows increased power distribution efficiency without resorting to bulky high current DC-DC converters. Each rotator housing contains a motor driver to control a stepper motor and collect encoder feedback for closed loop position control. Both motor drivers implement full isolation between the high power and control segments, which protects the vehicle's system control bus and limits the effect of a hardware malfunction.
The camera tilt control board controls a servo motor that provides tilt control for the camera. The board also delivers 15V power to allow for an Ethernet camera should it become necessary in the future.
The arm controller is designed along the same lines as out thruster motor drivers and provides full encoder feedback.

Thruster Motor Driver

The thruster motor driver incorporates closed loop feedback through an encoder and full isolation between the high and low power components. The driver is isolated using optocouplers for signal transfer and a DC-DC converter for power isolation. The board also shows I2C bus level shifting, which uses the Texas Instruments P82B96 level shifter chip, which allows the external bus to operate at 12V and also provides a near 6V of noise margin on the external bus, which helps prevent interference from causing communication discrepancies.

The board “Subrina Main Thrusters, Appendix A was the focus of much of our research and development work throughout the past year since it drives the primary thruster motors. This area provided a significant technical challenge, since we need it to provide reliable and safe control of the motors despite the lack of encoders for closed-loop control. The motors are designed for 24V operation, but we calculate that the supply voltage can vary between 36 and 48 V. In addition, physical layout constraints required that the board fit behind the motor at the back of the thruster housing. As a result, the design that evolved uses 2 stacked circuit boards to allow us to achieve component densities close to those obtainable from 4 layer boards without limiting the manufacturability and testability of the assemblies.

As with our other high power drivers, the board features full isolation between high and low voltage sections. To achieve this we use a combination of optical and magnetic isolation. Again, a standard optocoupler is used for the voltage and current feedback whereas, the high power section and the control signals are sent using an electromagnetic isolator. In addition, the magnetic coupler includes a small isolated DC-DC converter to provide the high voltage side of the circuit with 5V. To reduce the size of the controller as much as possible, we used surface-mount components exclusively. Though the extensive use of innovative components, surface mount technologies and careful module design we were able to fit this feature-rich motor controller into a 2-board stack with a diameter of less than 40 mm, and a height of less than 35mm.

User Interface Control Panel (top side) Electronics

The topside configuration is simplified from previous years. In order to minimize the size and mass of our umbilical we incorporated only those signals that are absolutely necessary. In addition, the laptop is the hub for all human interface controls at the surface. This eliminates the need for complex control circuitry or custom electromechanical control hardware at the surface. Control is provided through the umbilical's Ethernet cabling and connected to a wireless router to eliminate the need for a tethered control computer. In addition to providing the convenience of an un-tethered computer, the wireless connection serves to isolate the computer from the system, thereby protecting the computer and isolating it from water damage. The router also serves as a DHCP server, making configuration and setup simple. Currently our primary control inputs are a 3 axis joystick and a 6 axis industrial steering device that are interfaced to the laptop though a customized USB-serial adapter. Due to the common nature of the hardware, augmenting the controls with additional interface devices should be painless.

Safety

In normal use, conservative calculations show that our vehicle should never draw more than 15 amps from the 48V power source. We have installed a circuit breaker with a 15 A rating as our first line of protection. A mechanical master kill switch is the next safety device, allowing anyone to cut the power to the ROV in case of an accident. Finally a 40 amp fuse is provided as last line of defense, and is required for us to participate in the MATE ROV competition.

Power

The vehicle is designed to be extremely versatile and we have designed for a nominal supply voltage range between 24 and 48 volts. As such, we can operate from many different battery configurations as well as from 48V AC switching power supply. This means that we have a backup power source even in the case of a severely depleted battery pack

5.3 Software

Block-diagram or flow-chart of software in the ROV

SUBrina's software systems are separated into three main categories: Embedded software, server/controller software and client software. The diagram below shows the software and communication architecture used on our vehicle.

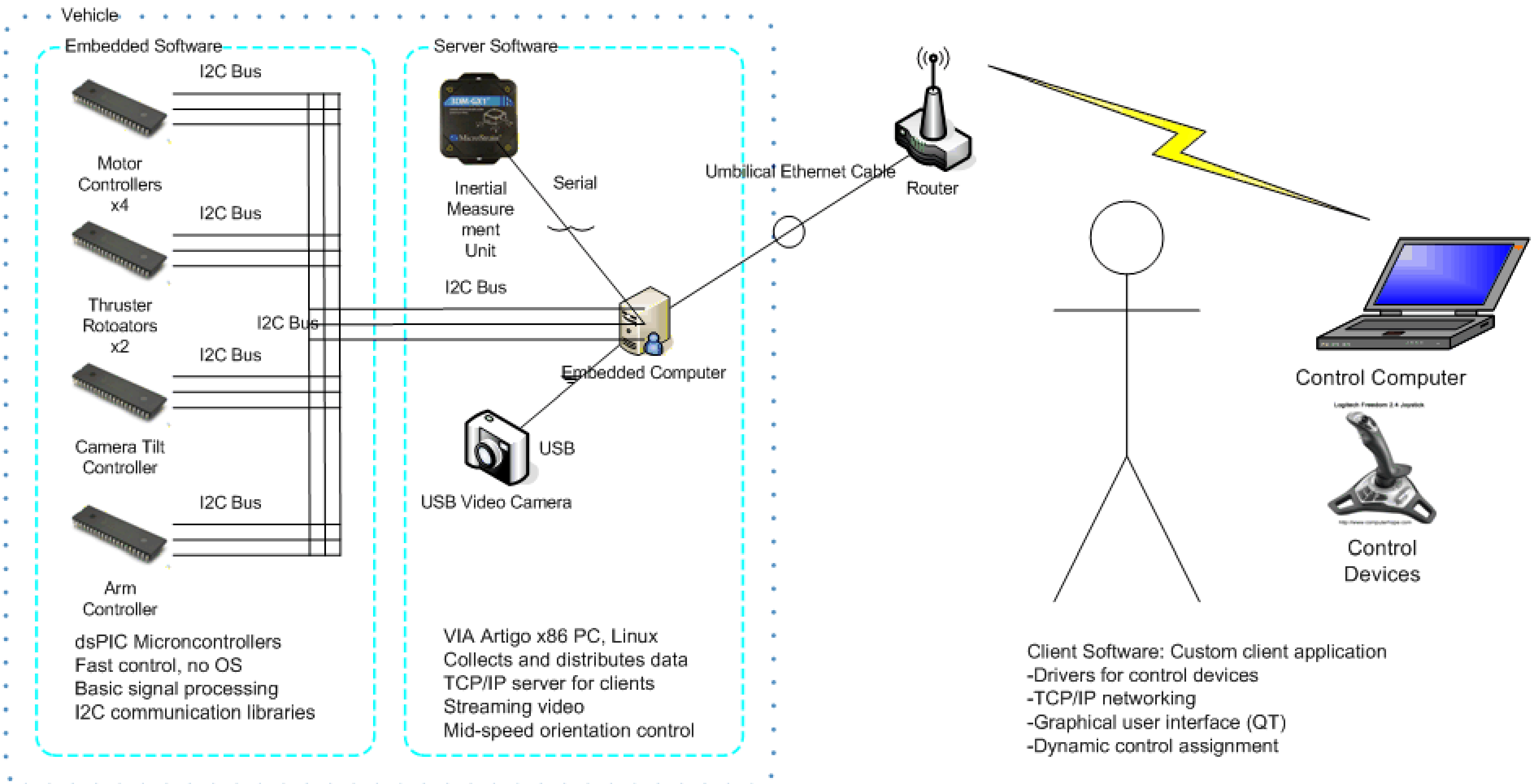


Figure 6- Software Flow Chart

Each controller board designed by the electrical team has a dsPIC microcontroller on board with an I2C interface [6] used to communicate with the embedded computer, a VIA Artigo Pico-ITX pc [7]. The embedded computer co-ordinates the entire system, collecting data from all boards and transmitting commands to the boards. It also streams near-real-time video up to the control computer over an Ethernet link. The control computer receives feedback from the vehicle and displays this for the operator while taking input from the control devices the operator uses to control the vehicle.

Embedded Software

The embedded software implements fast closed loop control of the thrusters and rotator boards. It also provides some pre-processing of sensor data. The on-board microcontrollers run at 1.87 MIPS or 7.37 MIPS allowing them to react to changing conditions on a 10us timescale. By using hardware interrupts and the microcontroller's I2C module we can ensure the communication does not interrupt the microcontroller's control tasks for more than around 10us while still allowing full communication rates between the microcontroller and embedded pc.

The use of embedded software and microcontrollers enables our vehicle to respond to incredibly fast phenomena that a normal computer would take many orders of magnitude longer to respond. (for comparison, the round trip time of a byte of data on a fast serial link is approximately 200us plus the additional overhead of the operating system's context switching - on the order of 1000us).

The embedded software also includes models for the motors it controls so that the motors can be controlled smoothly and so that we can achieve uniform thrust between thrusters.

All embedded software uses a custom communication protocol over I2C that allows a type of plug and play architecture and a strong 6 bit Cyclic Redundancy Check (CRC) to all messages. In most cases a conformant device can be added or removed while the system is running and the system will recognize that the device is new or that the device is missing. All that is required is that two devices do not share the same bus address (required by the I2C bus). These features allow the following scenarios to be realized:

- A device could be removed from the system, reprogrammed to provide different/new data and then re-connected and the system will interrogate the device to find out what data it accepts or provides.
- A faulty thruster can be replaced without re-compiling any code.
- A new sensor can be added to the system without the system needing to know about the sensor beforehand.

Server Software

The server software component of the system resides entirely on the embedded computer. It is responsible for the following functions:

- Being the I2C bus master device to transferring all sensor values from the boards, and sending all commands to the boards.
- Acquiring orientation information from the on-board Inertial Measurement Unit (IMU)
- Acquiring video from the camera
- Running a TCP/IP Ethernet service that the client computer can connect to in order to retrieve readings from the vehicle and to send commands to the vehicle. This also includes a separate TPC/IP service to do near-real-time video streaming.

The server software in this year's vehicle is based on last year's software with some major changes.

This year we support streaming video in the embedded computer. Last year's vehicle had an Ethernet video camera that did not go through the embedded computer. We found we could get similar latency and frame rates from a good USB camera and efficient software on the embedded computer as we could from a dedicated Ethernet camera at much lower cost. This meant we had to understand how to interface with a USB camera.

This year we also changed our inter-process communication strategy on the computer. Previously all communication was done through a TCP/IP connection to a single server. This meant we incurred the overhead of the entire TCP/IP stack for every bit of data we wanted to send/receive. This year we upgraded to using shared memory which made passing data between processes as inexpensive as a memory read or write. It also made our system tolerant to processes crashing. a process could crash and restart and the data it was working with would persist in the shared memory.

The revised TCP/IP server is enhanced to run as multiple processes - one process per connected client instead of running as a single process.

Client Software

The client software, written using the QT graphics toolkit for cross-platform compatibility, is responsible for the following items:

- Displaying status and sensor values from the vehicle
- Displaying near-real-time video on the Head Mounted Display (HMD)
- Processing the following inputs from control devices to control vehicle motion:

- o Joystick (USB)
- o Industrial steering Device (USB)
- o Inertial Measurement Unit (serial).

The client software is user friendly interface with customizable controls. A few unique features include:

- Head movement control of the camera tilt. The Operator can, by looking upwards and downwards cause the camera to follow their eyes by tilting up or down.
- Head movement control of the vehicle direction. By correlating the IMU angles from the HMD and the IMU in the vehicle one is able to control the direction of the vehicle using head gestures.

The client will have dynamically map-able thrusters so that a thruster can be moved or replaced. The operator will be able to re-assign the thruster position on the vehicle (along with any associated control information).

Many controls can be re-assigned to suit the preferences of the user, but the general software process is shown in the diagram below.

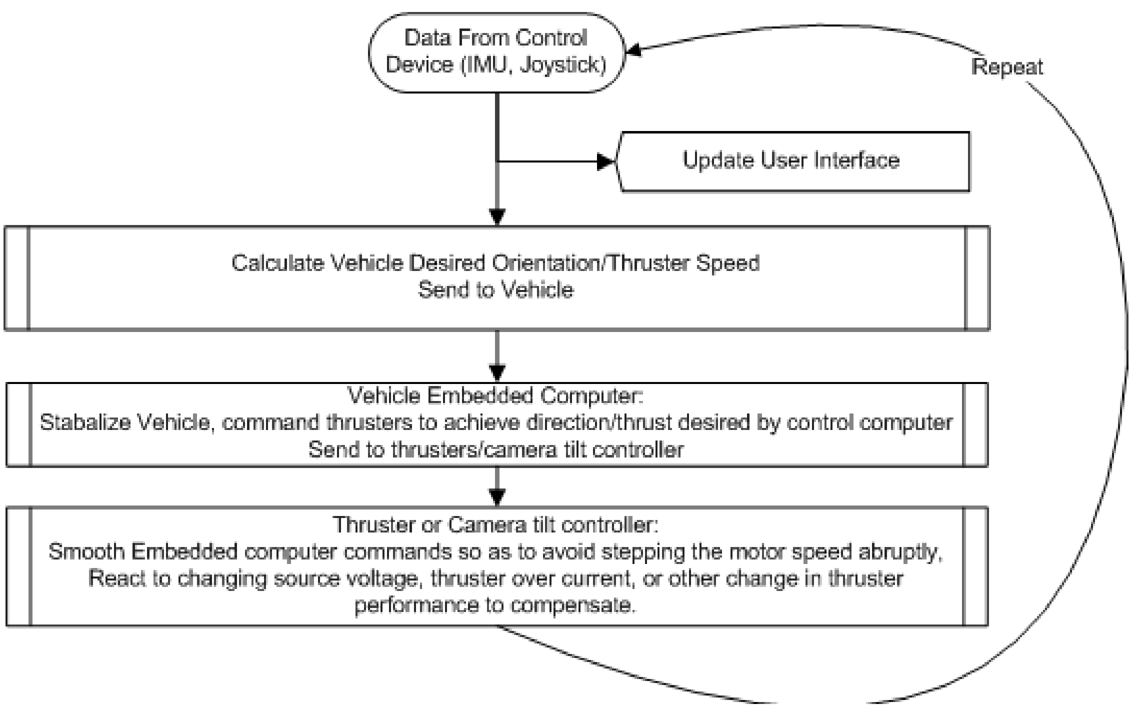


Figure 7- Basic User Control Loop

6.0 Analysis

Every project comes with its share of design challenges and road blocks. Here we describe some challenges which were faced and how they were resolved.

6.1 Finance

Analysis of our Financial Situation Proved to be reasonable, as shown in the table below. Furthur details are included in Appendix A

The table below summarizes our expenditures.

Transaction Report

From 01/08/2008 To 30/06/2009

Date		Description	Amount	
Electrical				
Total For Electrical			$1,472.00	
General				
Total For General			$9.30	
Mechanical				
Total For Mechanical			$1,341.54	
Software				
Total For Software			$1,108.93	
Upcoming Expenses				
Total For Upcoming Expenses			$3,900.00	
Expenses Grand Total			$7,831.77	
EngSoc Sponsorship				
	31/10/08	EngSoc A Sponsorship	$300.00	
	19/03/09	EngSoc B Sponsorship	$585.00	
Total For EngSoc Sponsorship			$885.00	
WEEF Sponsorship				
	24/11/08	Weef Sponsorship	$1,890.00	
	18/03/09	Weef Sponsorship	$1,545.00	
Total For WEEF Sponsorship			$3,435.00	
Sponsorship Grand Total				$4,320.00

We are re-using the following items:

Items used from last year's competition	Approximate Value
Inertial Measurement Unit x2	$1200 US
I-glasses Head Mounted Display	$900 US
Thrusters x4	$150.00

6.2 Challenges

I2C Clock Stretching - Technical Challenge

One feature of the I2C bus, as specified in the SMbus standard is the ability for a slave device (sensor board) to tell the master device (embedded pc) that the data is not ready yet and the master needs to wait while the slave prepares the data. This is a rather fundamental feature that allows the slave up to an extra 100µs to get the data ready. While this does not sound like a lot of time, one must consider that the slave has no advance warning that the master intends to read from it. In our system the master would start trying to pull data out of a device just under 15µs after it finished telling the device it wanted to read from it.

Shortly after we began testing our communications system we noticed that out embedded pc's hardware or hardware drivers did not support clock stretching, and would return an error if the slave said it needed a few microseconds more. We either had to try and find where the bug in the pc's hardware/drivers was or we had to get all our microcontrollers to completely avoid clock stretching. This meant all of the following had to happen in under 15us, without warning, each and every time:

- The microcontroller had to recognize that it was being addressed and recognize that the master was trying to read from it.
- The microcontroller had to enter our interrupt service routine and save all its working registers
- And our code had to come up with the data or information requested by the master

Our final solution performed significantly better than required to meet the goal. We reduced the second bullet item by using the microprocessor's shadow registers to save the working registers, reducing this part from ~15 assembly language instructions to ~1 assembly instruction. We then re-factored our code to pre-calculate the next byte of the response that we would likely have to give. This got us closer - to around 18µs or so. We then had two options: write the routines in pure assembly language or utilize the microcontroller's phase locked loop (PLL) to increase the clock speed by a factor of 4. We chose the second solution and achieved a response time of ~5µs. Success!

6.3 Trouble Shooting

Using Logic Analyzers to Debug Low Level Code

During the initial use of our first I2C device boards we encountered difficulties communicating with some of our boards. Purely from a software perspective, it was not entirely clear where the problems being encountered were originating. Both our microcontroller and embedded software appeared to be performing all the correct API calls. Without the use of a logic analyzer it would have been difficult or impossible to pinpoint the cause of the errors we were seeing. When writing code that operates on the edge of the boundary between hardware and software, the additional insight provided by a low level examination is extremely helpful.

Using the logic analyzer we were able to determine that the signaling on the bus appeared to be in specification up until an unexpected negative acknowledge of data following a clock stretch event. Because of our buffering scheme on the I2C bus one possibility that was raised involved a malfunctioning or improperly installed buffer chip not properly transmitting information in upstream direction. The larger number of channels available in a logic analyzer as compared to an oscilloscope, 16 versus 2 in our case, allowed us to probe out i2C bus at multiple locations along our signal chain and verify that our signal integrity was maintained. This same advantage is very useful in allowing the use of extra microcontroller I/O pins as trigger inputs for the logic analyzer in order to perform timing analysis and to determine the which side of our signal chain the unexpected negative acknowledge was coming from. Detailed analysis though a combination of hardware and software guided debugging eventually allowed us to isolate the error we were seeing to a bug in either our chipset's I2C hardware or the Linux kernel driver for the chipset. In either case we decided to implement a workaround rather than sink more time into isolating what was possibly a bug in silicon.

Without a logic analyzer this debugging process would have taken significantly longer to complete and would have required the use of multiple independent test instruments. The use of multiple instruments would not only have been time consuming, it would also have required that we gain temporary access to the instruments, which are typically expensive and fairly spread out between university laboratories. One feature that some logic analyzers provide that we were not able to employ in our case are built in protocol decoders, which would have helped accelerate our analysis of bus traffic by eliminating the time consuming and error prone manual decoding of traffic. Despite this, we found logic analyzers to be an essential tool in working with low level code.

7.0 Future Improvements

Thruster Motor Upgrades

One improvement we plan on investigating in the future is upgrading our thruster motors. There are two main areas of improvement we wish to investigate, the first being true encoder feedback and the second being switching from brushed to brushless DC motors. Although our current system already represents a significant advance over our previous designs, these enhancements would help further improve the agility and efficiency of our system.

Adding encoders to our motors will allow us to monitor the speed of the motors directly, rather than operating open loop or relying on indirect forms of feedback. This will allow us to exert more direct control over the thrust generated by our motors, making the vehicle easier to control, both under full manual control and when using computer algorithms to assist in stability control. This enhanced feedback will also be a significant advantage in our future efforts to equip our vehicle for autonomous operation.

The other major change we are investigating is to make use of brushless DC motors in the thrusters instead of our current brushed DC motors. Brushless motors will allow us to obtain higher power density and greater total output power. This will greatly help to improve our ability to maneuver, even in difficult conditions. In addition brushless motor drivers will allow us to operate at higher efficiencies than our current motors. Although we are already focusing on efficiency throughout our design, the efficiency gains made by using better motors will produce significant improvements that will enable autonomous operation from batteries.

Other Improvements

- video processing/machine vision assisted control of the vehicle
- Field Testing of our claimed depth rating
- autonomous navigation using way-points
- autonomous navigation using goal-setting and using path finding in a SLAM-type system
- Fuzzy logic controller for the overall vehicle control
- Investigation into optimal control of a dynamic MIMO system using a LQR controller

8.0 Conclusions

8.1.1 Team Reflections

David's Reflection:

2nd year Mechatronics engineering, Mechanical Leader / Team Leader

This year a distributed leadership approach was taken. Electrical, software, and mechanical teams cooperated and often overlapped their tasks. Though team responsibilities were shared, as the team leader and mechanical designer I often found it difficult to balance the two positions. Setting deadlines, machining parts, dealing with manufacturers and shipping taught me the importance of detailed design. From working with the other members of the team I learned the importance of clarity in communication. Designing the mechanical system from scratch was both time consuming and difficult. One of the most time consuming tasks was determining the scope of the project and how our design would meet all the requirements we had set. Our design needed to be compatible with both the MATE ROV competition, and the AUVSI restrictions. Deciding the target weight, dimensions, and the thruster configuration was difficult since they are the most open-ended aspects of the design. Several compromises in the design occurred along the way, such as the camera housing being nearly double the intended diameter due to the camera we initially chose. One challenge was how to mount the camera housing in front of the rotating thruster without significantly increasing the overall vehicle dimensions. Although the target dimensions of the ROV were not met, the purpose of the dimensions was met: to fit easily into the trunk of a car. This was one of the first design projects that allowed me to design a complete mechanical system from the ground up using CAD. while part of a FIRST team I was able to observe proper design techniques. While on a SKILLS robotics [9] team I was able to design a system on a trial and error basis, using CAD as a post project documentation tool. Being able to combine both previous experiences opened the door to all sorts of learning experiences I had not yet encountered. Representing the team and being responsible for meeting the qualifying deadline solidified my learned management skills. Pushing for deadlines significantly earlier than the actual deadline proved effective. although we needed to hurdle through an embarrassing 4 backup qualifying plans in order to demonstrate qualification, I gained insight that working out the kinks of a system always takes longer than is expected, even if a 2 times safety margin is used.

9.0 Lessons Learned

One lesson that was learned was the difficulty and importance of time and resource management. Despite being scheduled early, our qualification demonstration kept on getting pushed back further. By the time we found small electrical nuances, located all the last minute water leaks, etc. we qualified with only days to spare, rather than weeks as planned. we learned that things take much longer than expected, even when the expectations are very conservative. learning to get tasks completed early rather than waiting until the last minute is something that continues to emerge when we find ourselves completing tasks.

10.0 Acknowledgements

We would like to thank the following sponsors for their continued support:

Assembla

Asi-Group

Curtiss-Wright Controls

JetCut Main
OceanWorks
StrataSys
WEEF
University of Waterloo Engineering Machine Shop
Leoni
MicroStrain
Advanced Circuits
PCBexpress
SpaceControl
EngSoc
Dean of Engineering

11.0 References

[1]MATE ROV Competitions <http://www.marinetech.org/rov_competition/ >Retrieved 2009/05/27

[2] NATO Submarine Rescue System <http://www1.rolls-royce.com/marine/ downloads/submarine/broc_nato.pdf>Retrieved 2009/05/27

[3] Francis, T. James R.; D. J. Smith (1991). "Describing Decompression Illness.". 42nd Undersea and Hyperbaric Medical Society Workshop. UHMS Publication Number 79(DECO)5-15-91.. <http://archive.rubicon-foundation.org/4499> Retrieved on 2008-03-17.

[4] NATO Submarine <http://www.royalnavy.mod.uk/upload/img/FX08_0210_354.jpg> Retrieved 2009/05/27

[5]SeaEye Falcon ROV <http://www.seaeye.com/falcon.html> Retrieved 2009/05/27

[6] I2C Specification <http://www.nxp.com/acrobat_download/literature/9398/39340011.pdf> Retrieved 2009/05/27

[7] Artigo pico ITX PC<http://www.via.com.tw/en/products/embedded/artigo/a1000/index.jsp> Retrieved 2009/05/27

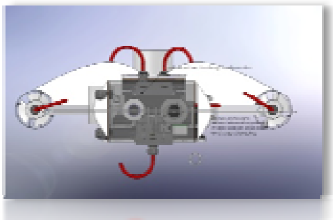
[8] "closed-loop feedback control system." Encyclopædia Britannica. 2009. Encyclopædia Britannica Online. 28 May. 2009 <http://www.britannica.com/EBchecked/topic/122157/closed-loop-feedback-control-system>

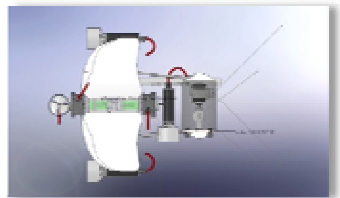
[9] Dmikolaj's 2006 Skills Involvment <http://mpceng.com/david/skills_2006.asp?go=team_involvment> Retrieved 2009/05/27

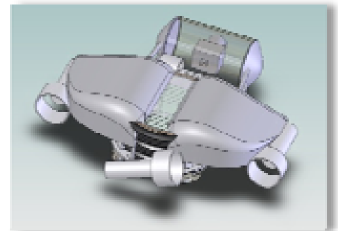
Appendix A

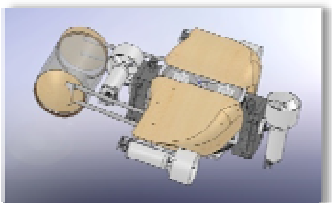
Summary of Design Concepts

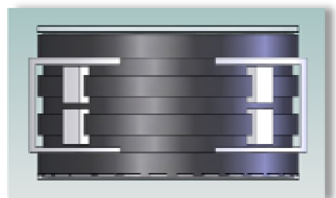
Images of Design Concepts

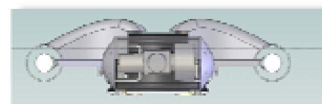


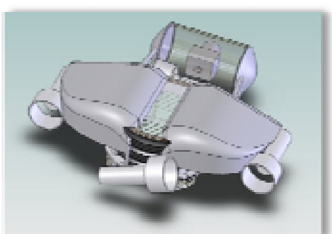


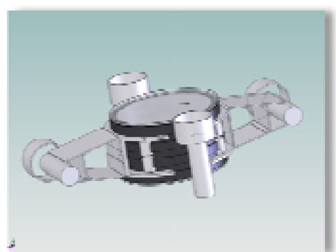


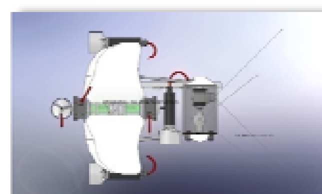


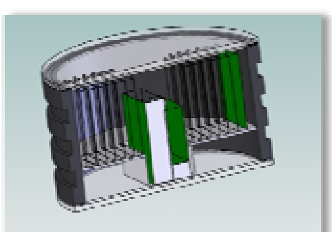












Additional Team Reflections

Nathan Buchanan's Reflection:

4th year Electrical Engineering/Software Option. 2008 Software Lead.

This year's software team's experience has been significantly different than last year's. We started out with a blank slate and designed the system we wanted instead of simply expanding on an existing system. This time we worked closely with the electrical and mechanical teams to ensure we ended up with a robust communication system. We got significant support from both electrical and mechanical teams where they used their expertise to add on critical items that were unknown to the software team (such as the I2C buffer chips, and sourcing shielded Ethernet cables).

The software team has added two new members: Saskana and Chris with another member, Pablo, leaving part way through the year to focus on staring up another team. Our focus has broadened as well. We are now involved in writing our own video streaming applications instead of using off the shelf ones. We have also extended into embedded programming to achieve better vehicle control, partially taking over this task from the electrical team.

Two main themes have been our focus this time around: feedback/performance and fault tolerance. We have been instrumenting each software system to provide as much information as possible about its state and the state of the software layers below it. For example, the GUI would indicate how reliable its connection to the embedded computer is, the embedded computer would indicate how reliable it's connection to the thruster boards is, and the thruster boards would indicate how often they had to restart (and for what reason: power loss, code error, etc) This information allows us to quickly pinpoint the location of problems.

The second focus was fault tolerance using the concept that a failure in one part of the system should not require the entire system to restart. We achieved this with our shared memory and multiple process architecture.

The biggest challenge our team has encountered was delays in getting operational components. For example, we waited on our camera for 4 months before deciding we had to change cameras a few weeks ago (as of this writing). We were partially impacted by a tendency to leave things for later as well.

Overall it has been a great experience and we have learnt much about different classes of software programming and real time requirements of control systems.

Cameron's Reflection:
Third year Electrical Engineering, Electrical lead

I joined the team near the end of my first year of studies at Waterloo. The first year I helped with all the little tasks that needed to be done and I saw the first vehicle come together. This year I took on the lead role for the electrical team and created some of my very own designs as part of a brand new vehicle.

This was my first real leadership role and I found the responsibilities and expectations that came with leading almost ten people quite daunting. It was very easy for me to spend all of my time organizing others and not have anytime left to do the technical work that I enjoy.

In future leadership roles I will need to better utilize time and project management tools such as a Gantt chart. The difficulties that I encountered through this leadership experience will make me a much better leader in the future. The best part of this experience was building some very interesting electrical systems with some very smart people.

Paul's Reflection:
First Year Computer Engineering: Paul Roukem. Electrical Co-Lead.

My experience joining the team was much different than that of many first year students. I had studied many of the technologies and programs we would be using independently before coming to university, but had not had a chance to significantly apply them. Because of my prior knowledge, I was able to jump in quickly on the technical aspects of the project, but I still had a lot to learn about how to communicate my ideas with other team members. I feel that working with the team has helped me grow significantly in terms of teamwork skills, both as a team member and as a leader.

One area that I found particularly challenging was working with other junior team members to review and enhance their circuit and board designs. This difficulty was only increased by the fact that I was working remotely since I was on a work-term when much of the board design work was being done. Ensuring that the boards we produced were functional while ensuring that other team members were not discouraged or offended by large volumes of feedback required careful attention to the way the feedback was delivered.

In the future I think that better management of our information and technical documents would be an asset, as would improved time management. Currently schedules are often set, but no work proceeds towards meeting them until near the deadline. This results in compromises being made, sometimes leading to other problems later on. I believe that using a wiki, rather than documents in our revision control system may help with our document control. I look forward to taking on more of a leadership position and working more with the team in the future.

Nick Ford's Reflection:
2009 Graduate from Electrical Engineering; Electrical Lead during 2007 and 2008 competition entries; General labourer

I have worked heavily on the last three year's ROV designs, and have operated as the Electrical Lead twice in the past, as well as something of a technical lead to pick up the slack when we were short of software and mechanical leaders. During the 2008 competition preparation, I became very frustrated at the amount of admin I ended up performing as a team leader. And as someone intensely technical I found it quite difficult to delegate my design tasks to other members in the team. However, there were exciting moments, I enjoyed seeing the overall system work, had the chance of leading groups of highly intelligent people, and would take the leadership positions again without hesitation. My decision to take less of a leadership role this past year was intended to allow other members to step forward and take over following my graduation, and I anticipated working on more of the design side again. I think I achieved that to a large extent.

As an electrical engineer, it is unusual that of the design work I did on this year's ROV, much of it was mechanical-related. Dave won't admit it, but I think he might have done a better job with some of my designs; I am finally on the other end of the delegating task and have had the chance to frustrate a team lead with my inexperience, and overdesign. I had a lot of fun designing the internal mechanism of the thruster rotators and learned a huge amount about pressure angles, pitch diameters, encoder design, stepper motor construction and how not to choose shaft seals. I am no mechanical engineer, but I have a very well-rounded education as a result of working with this team. If I had the opportunity of starting my university career over, I would still take an electrical engineering degree, but I would jump at the opportunity of becoming involved with a student team far earlier. I believe this team is very promising, has some outstanding current team members and I will actively encourage new students to get involved as early as possible during their degree.

IMU Positioning Feedback Improvement

The incorporation of an inertial measurement unit (IMU) into the control system is an improvement we are currently working on and hope to complete before competition. There are a number of areas where the use of an IMU will provide significant benefits. One form of integration we are investigating is stability and roll/ pitch control. This will help simplify our control system and reduce the workload of the ROV operator. Another type of integration that we hope to employ is direct orientation feedback to the operator. This should improve the situational awareness of the operator and allow them to reason better about the positioning and orientation of the vehicle.

For stability and pitch/roll control we hope to implement a control loop on our embedded computer, which will allow us to adapt to changes in the forces acting on the vehicle and maintain the desired orientation. Currently the operator is responsible for manually adjusting the power of each thruster to manually control the attitude of the vehicle. Ideally the operator will be able to specify the desired pitch angle one of our control interfaces and the control system will automatically compensate for buoyancy imbalances and thrust vectors that are not through the vehicles center of mass. This capability will free the operator from dynamically adjusting the thrust when changing from stationary to forward movement, allowing them to focus to the task at hand.

Orientation feedback should be less difficult to achieve, however presenting the orientation information in an intuitive manner still presents some challenges. One form of integration we are investigating is to use the artificial horizon system often seen in aircraft. This system has the advantage of being relatively easy to implement. A longer term option is to integrate orientation feedback into our video feedback, possibly in the form of heads up display type indicators for pitch and heading. Either of these options provide significant orientation and positioning feedback to the vehicle operator, although the heads up display option will make the information more intuitively available.

Other Trouble Shooting Techniques

- We designed a I2C bus enumerator that can be connected to any port on our I2C bus and will scan the bus to detect which device are on the bus or if a device is malfunctioning and preventing the bus from operating. This tool consists of a PIC microcontroller connected to a computer using a serial cable (to display results).
- In debugging out network and in our initial bring up of the I2C code on the microcontrollers and computer we used an oscilloscope to verify that the communication signals were clean and were at the proper voltage levels.
- Logic Analyzers were used to probe the running bus and examine the signal timing and discover associated problems (such as lack of support for clock stretching).
- When programming the embedded controllers we used flashing LEDs and eventually serial port output to ensure the code was doing what it needed to be doing.
- For the computer based programs, we primarily used printf statements, but occasionally ran the GDB debugger to examine more difficult errors.
- Finding mechanical leaks by blowing into a pipe and finding air bubbles, and using Dave's “patented” paper-towel method

Other Challenges

Indirect Motor Speed Sensing - Technical Challenge

Having designed the thrusters for our vehicle for the last two competitions, our team has gained significant experience in the design of propulsion technology for use in submersible vehicles. Our original thruster motor controllers were designed without feedback, and were based solely on the assumption that the PWM duty cycle would be proportional to the motor speed and thrust output. This proved to be too crude an approximation to be effective for control. The classical controls solution to this issue is to correct for the error with closed-loop feedback [8]. However, this change would require a significant investment into new motors fitted with rotary encoders; an expense that we could not justify.

To overcome this challenge, we spent several months simulating and testing alternate solutions to the control problem. Using classical controls theory for modeling the motor as well as cutting-edge signal manipulation techniques we produced an innovative motor controller that avoids the cost of additional motors and encoders.

We used the simplified model of a DC motor shown in Figure x. The speed of the DC motor is directly proportional to the Electro-Magnetic Field, and the supply current is proportional to the torque on the motor shaft. Using a resistor to model the non-idealities of the motor we verified the relationship between the motor's terminal voltage, supply current and speed. We eliminated the need for an encoder by calculating the motor speed from average terminal voltage and current measurements.

the design challenge became more complicated because we completely isolated the control logic from the power circuitry. By design, isolation barriers are non-linear, and so only suitable for digital signaling. Consequently we cannot transmit a raw analog signal reliably without some form of modulation. Using the novel technique of pulse density modulation, still being researched as a possible design for arbitrary precision analog-to-digital converters, we successfully transmitted a high-frequency analog signal to the control logic and maintained isolation.

As a result of the challenge faced, our controller turns a simple DC motor into a more versatile device that is not only able to achieve the same performance as a more expensive motor, but we have significantly more control over the isolation between logic and power, over-current and over-voltage protection, along with speed and torque control.

Detailed Financial Report

The table below summarizes our expenditures.

Transaction Report

From 01/08/2008 To 30/06/2009

Date	Description	Amount	
Electrical			
01/11/08	Prototype boards	$7.86	
29/12/08	Circuit Boards	$70.72	
30/12/08	Digikey Order	$117.75	
30/01/09	norvel IC chips	$77.00	
12/02/09	Digikey Order	$85.86	
28/02/09	Camera Tilt Board (90.04US)	$117.45	
28/02/09	IC shipping	$30.26	
27/03/09	Electrical Components	$125.60	
27/03/09	Thuster Prototype Board (102.71US)	$131.01	
30/03/09	sayal: copper clad board	$9.32	
06/04/09	Servo Motor for Camera Tilt Module	$59.94	
16/04/09	Ethernet Switched Power Supply Board (56.94US)	$70.53	
16/04/09	Rotator Controller Circuit Boards (66.00US)	$81.75	
20/04/09	Shielded Ethernet Cables (22.26+8.02)	$30.28	
24/04/09	Thruster Motor Control boards (78.04+49.01)	$127.05	
26/04/09	Thruster Motor Control Board Components	$291.21	
29/04/09	sayal: heatshrink	$14.41	
09/05/09	Canadian Tire fuse	$12.72	
16/05/09	Fuses	$11.28	
Total For Electrical			$1,472.00
General			
01/01/09	Poster Printing	$9.30	
Total For General			$9.30
Mechanical			
04/12/08	Manipulator Parts	$55.88	
12/12/08	Can Materials(61.16US)	$77.76	
23/12/08	Aluminum pipe	$60.47	
31/12/08	Digital Force sensor for thruster testing	$45.19	
13/03/09	mcPhail's: quick release clamps	$22.58	
01/04/09	Home Depot (grouped): bolts, nuts, threaded rods, etc (17.62+12.34+17.48+6.77+16.60)	$70.81	
01/04/09	Canadian Tire(grouped): nuts, bolts, washers, pipes, etc (9.45+7.64+23.14+4.95+3.59+10.12+2.66+14.50)	$76.05	
06/04/09	potting compound	$27.35	
14/04/09	mcPhail's: quick release clamps	$27.10	
23/04/09	Rotator Housing Plastic (36.16+33.90+16.95)	$87.01	

23/04/09	Ball Bearings for thruster rotators	$114.63	
27/04/09	sayal: spacers	$7.12	
01/05/09	Fastenal: Nuts, Bolts, connectors (42.48+9.35+9.76+8)	$69.59	
01/05/09	Connectors, camera housing assembly	$600.00	
Total For Mechanical			$1,341.54
Software			
19/08/08	Pic Programmers (Digi-key) and Breadboards (Sayal)	$171.73	
31/10/08	Embedded PC	$364.35	
13/11/08	Memory for Embedded PC	$48.57	
21/04/09	Joystick	$22.59	
12/05/09	Webcam	$141.22	
13/05/09	Embedded PC (backup)	$360.47	
Total For Software			$1,108.93
Upcoming Expenses			
	Thruster Board re-spin	$400.00	
	Trip to Competition, Competition Entry fees	$3,000.00	
	Mechanical Arm Re-design	$500.00	
Total For Upcoming Expenses			$3,900.00
Expenses Grand Total			$7,831.77

EngSoc Sponsorship			
31/10/08	EngSoc A Sponsorship	$300.00	
19/03/09	EngSoc B Sponsorship	$585.00	
Total For EngSoc Sponsorship			$885.00
WEEF Sponsorship			
24/11/08	Weef Sponsorship	$1,890.00	
18/03/09	Weef Sponsorship	$1,545.00	
Total For WEEF Sponsorship			$3,435.00
Sponsorship Grand Total			$4,320.00

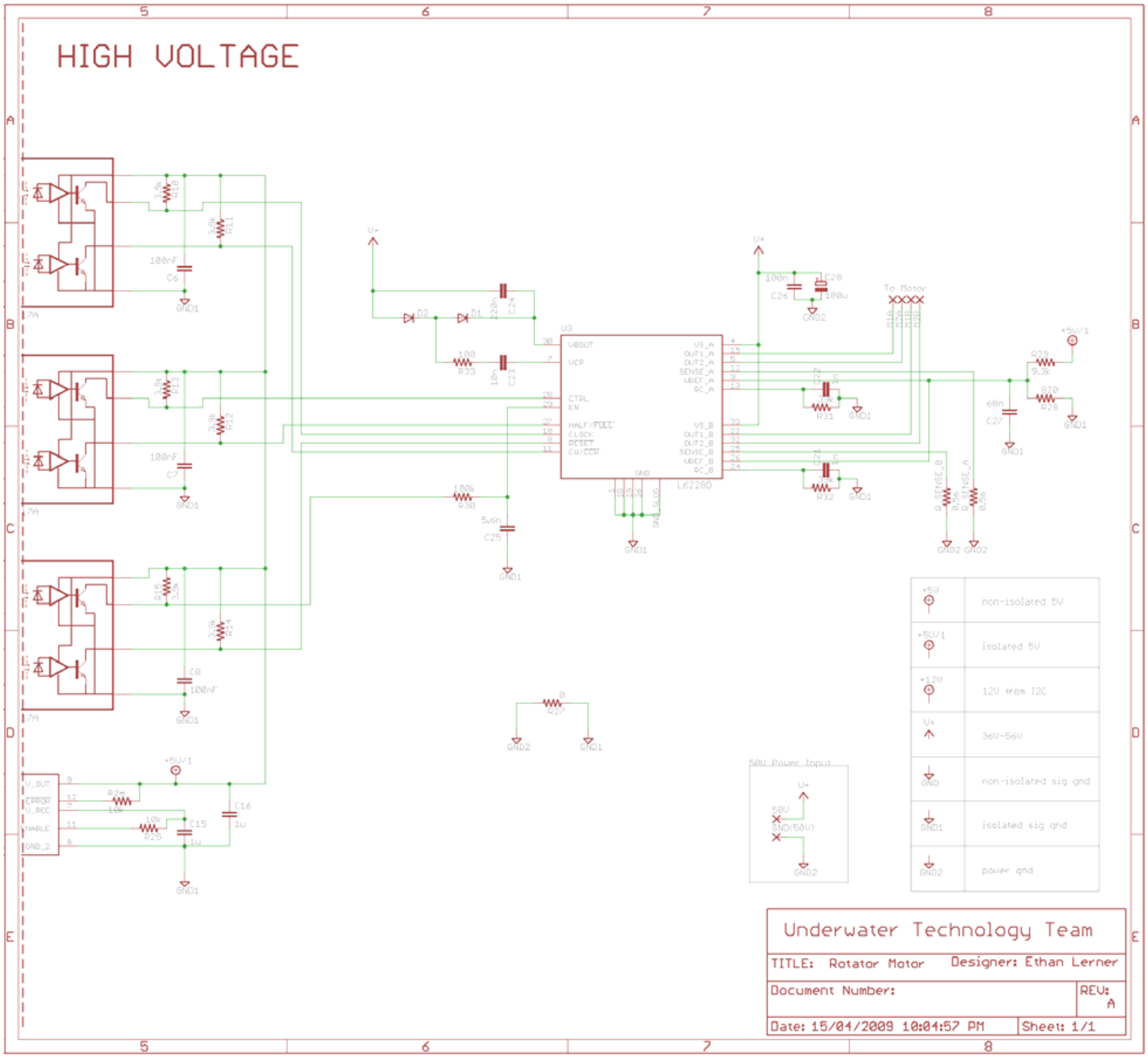
We are re-using the following items:

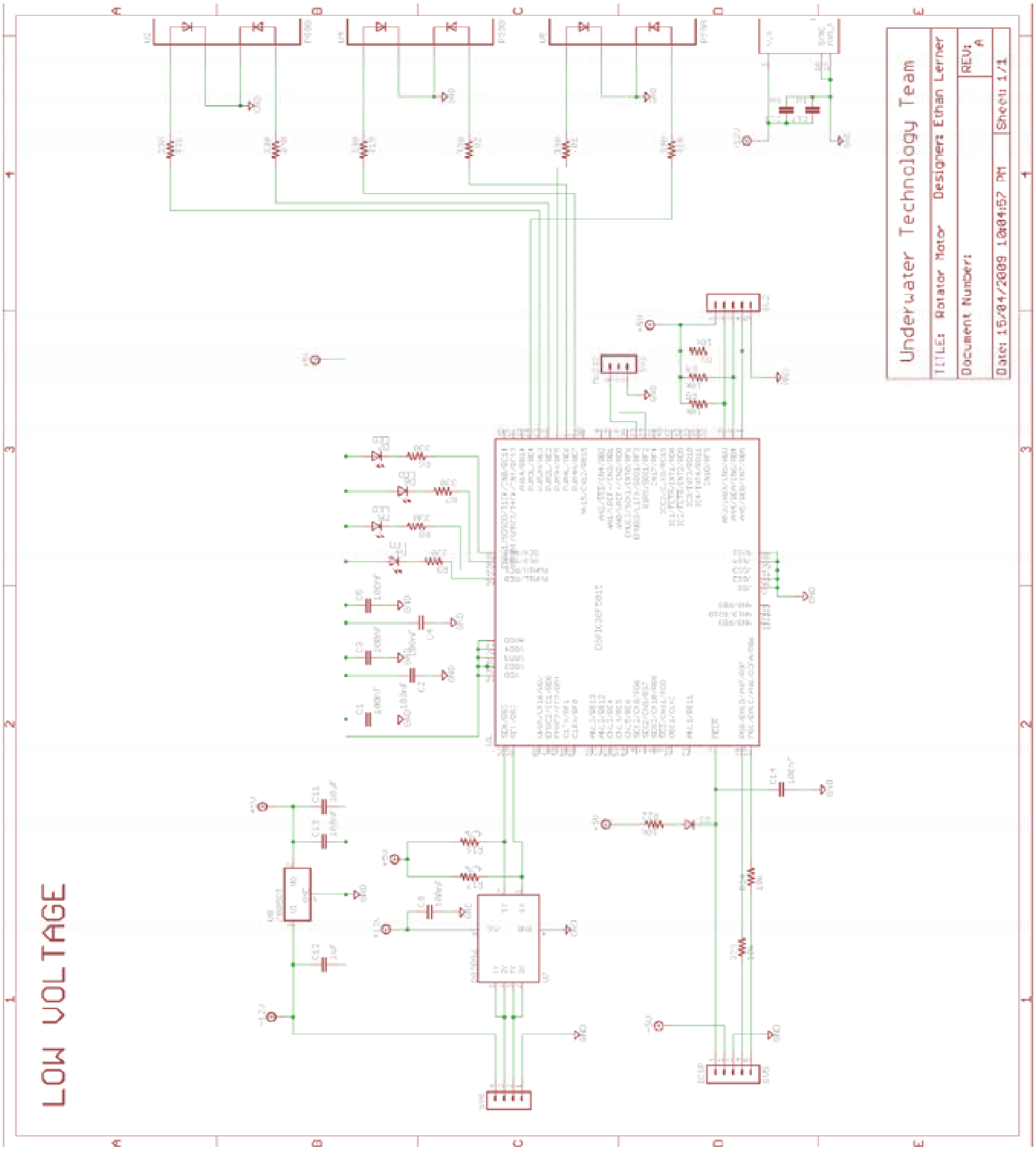
Items used from last year's competition	Approximate Value
Inertial Measurement Unit x2	$1200 US
I-glasses Head Mounted Display	$900 US
Thrusters x4	$150.00(really?)

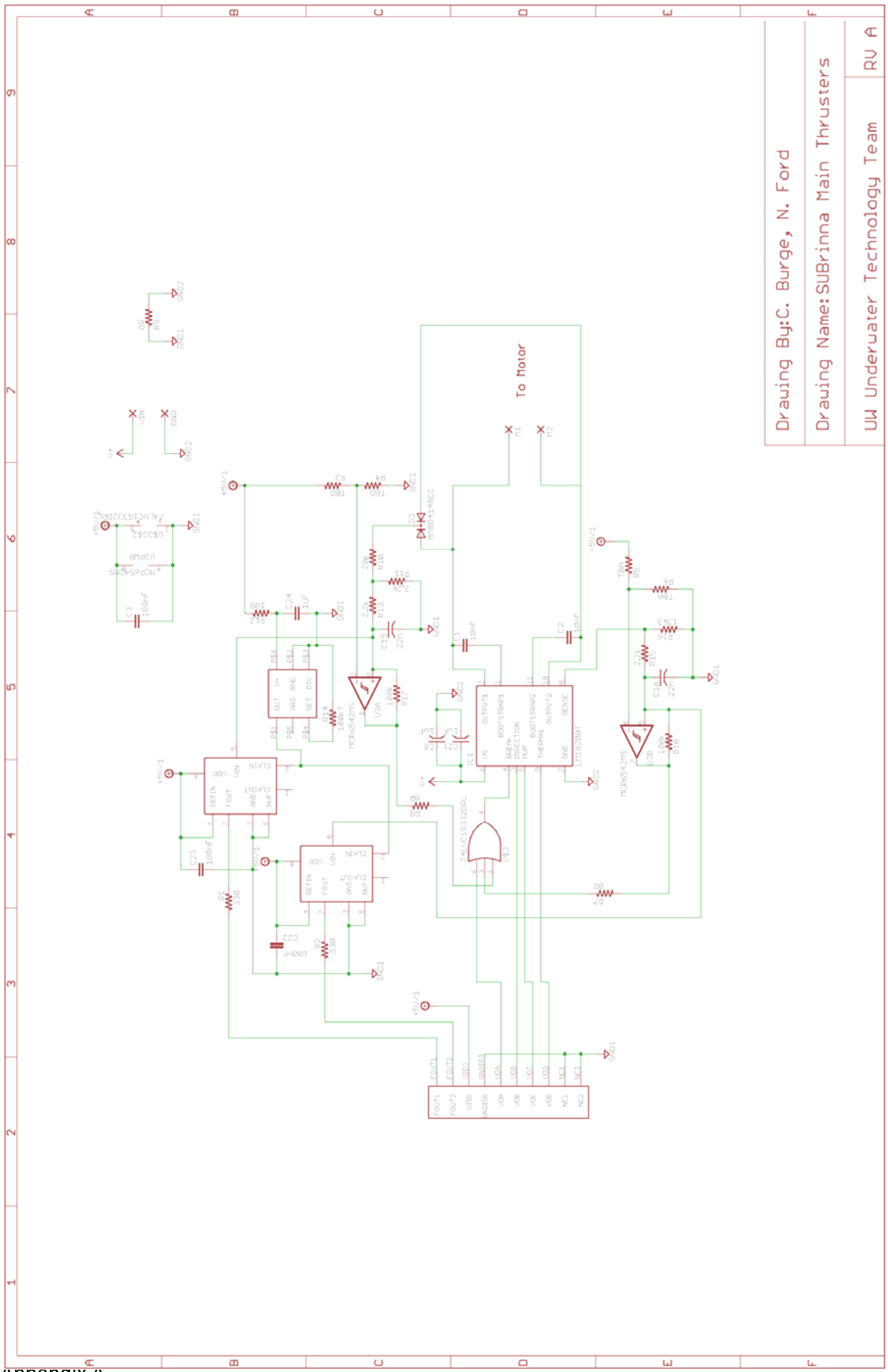
We are currently perusing additional sources of funding, that may include:

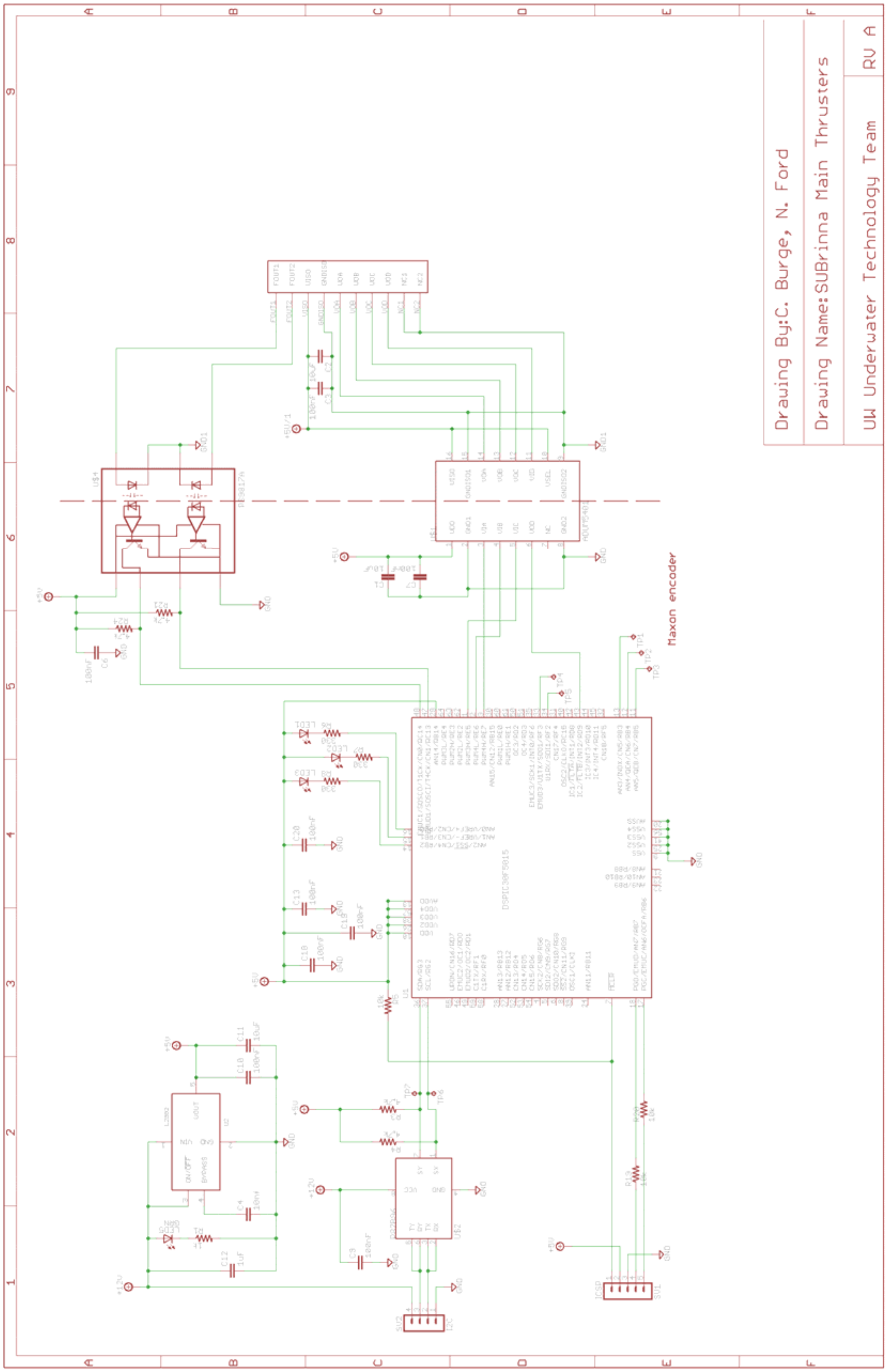
Potential Sponsors
ASI
OceanWorks
Engineering Dean

Additional Electrical Schematics




LOW VOLTAGE
Underwater Technology Team
TITLE: Rotator Motor
Designer: Ethan Lerner
Document Number:
REV: A
Date: 15/04/2009 10:04:57 PM
Sheet: 1/1


Drawing By:C. Burge, N. Ford
Drawing Name:SUBrinna Main Thrusters
UW Underwater Technology Team
RV A
To Motor


Drawing By:C. Burge, N. Ford
Drawing Name:SUBrinna Main Thrusters
UW Underwater Technology Team
RV A
Maxon encoder
DSPIC30F5015