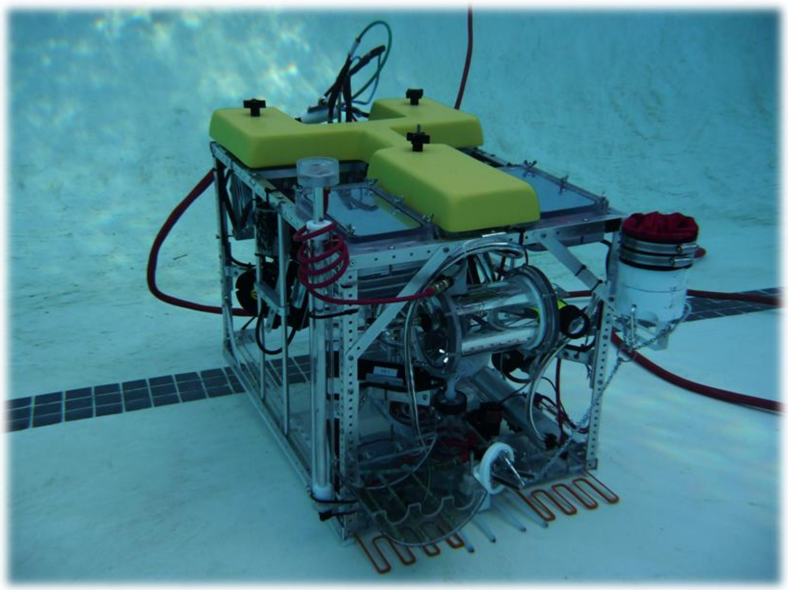
Presentation to MATE Center

ROVotics, A Division of Jesuit Robotics | Carmichael, CA
June 2012



Triton

Rolf Konstad, PE BSME	*Head Coach*	Jay Isaacs BSCS	*Mentor*
Joseph Griffin *Chief Executive Officer*	'12	Amirali Akhavi *Engineer*	'14
Evan Arnold *Chief Financial Officer, Pilot*	'12	Tyler Honnold *Engineer*	'14
Greg Marchese *Director of Engineering*	'12	Nolan Schneider *Engineer*	'14
Alan Luu *Website Administrator*	'12	Nick Sopwith *Programmer/Electronics*	'14
Spencer Breining-Aday *Head Machinist, Pilot*	'13	Andrew Standriff *Engineer*	'14
Chris Konstad *Head Programmer*	'13	Alex Aprea *Programmer/Composites*	'15
Drake Nylund *CADD*	'13	Jared Borg *CNC Operator*	'15
Jesse Tambornini Head of CADD, *CNC Specialist*	'13	Ryan Kenneally *Engineer/Composites*	'15
		Matthew Woollgar *Engineer/Composites*	'15

www.jesuitrobotics.org

Abstract

Triton is a work class remotely operated vehicle (ROV) designed to compete in the Marine Advanced Technology Education (MATE) Center International ROV Competition. Envisioned and created by ROVotics Underwater Solutions, an ROV production company that produces specialized vehicles for clients all over the world, *Triton* ultimately made its way from the drafting board to the pool deck through a structured build process that included several design reviews, prototypes, and revisions.

Triton utilizes an open frame design, common to many professional work class ROVs, providing flexibility for mounting accessories, access for field service, and minimal water resistance when maneuvering. Four of Triton's six thrusters are positioned at the frame's corners to allow for precise, vector-based control, and the two vertical thrusters are positioned inside the frame for protection. Triton has custom built tooling designed specifically to complete this year's MATE mission.

Designed with the customer in mind, *Triton*'s control system, which is based on ROVotics' proprietary C++ software in both the topside control system and embedded microcontrollers, allows smooth and precise control, monitoring of vehicle health via an effective graphic user interface (GUI), and ROV safety in the event of a communications interruption. Serviceability is facilitated via easy access electronics containers, modular electronics, and a quick release buoyancy compensator.

Originally designed in Solidworks, a computer aided design and drafting (CADD) program, and brought to life through mockups, manufacturing, and testing, *Triton* is a well-engineered product ready to meet the demands of this year's MATE competition. Testing on both the component level and full system level ensures that *Triton* is an effective machine, perfect for the survey and retrieval mission provided by MATE this year.

Table of Contents

ROVotics

ROVotics is a privately held corporation that specializes in designing and operating custom ROVs for deep-sea exploration and retrieval. Founded in 2003, ROVotics has successfully delivered six ROV systems to various clients worldwide. With the strength of the current ocean-exploration market, ROVotics expects to continue holding a strong position in its field.

Each fall, ROVotics recruits qualified employees eager to learn about and take part in the company's mission. Senior employees take on apprentices from the new hires so skills and knowledge can be carried on and developed. Refer to Figure 1 for the ROVotics corporate structure. This year, ROVotics screened over fifty individuals to fill six open positions.

ROVotics has served several clients in the past, all of whom have required the company to complete missions with the same dexterity and precision required in this year's MATE Center mission. Not only has ROVotics met all of the needs of its previous clients, it has also been featured on Good Day Sacramento and in the MAKE magazine for its success in survey and retrieval missions similar to the MATE mission this year. Working with these and other media contacts, ROVotics has increased public awareness of the applications of ROV technology.

Headquartered in Carmichael, CA, ROVotics currently employs seventeen people and has a long-standing tradition of quality, professionalism, and leadership. This year, we enhanced our engineering capability with the acquisition of a CNC mill, printed circuit board design and composites manufacturing. The company is dedicated to producing robust and reliable products able to withstand the harshest conditions in some of the most remote bodies of water on Earth.

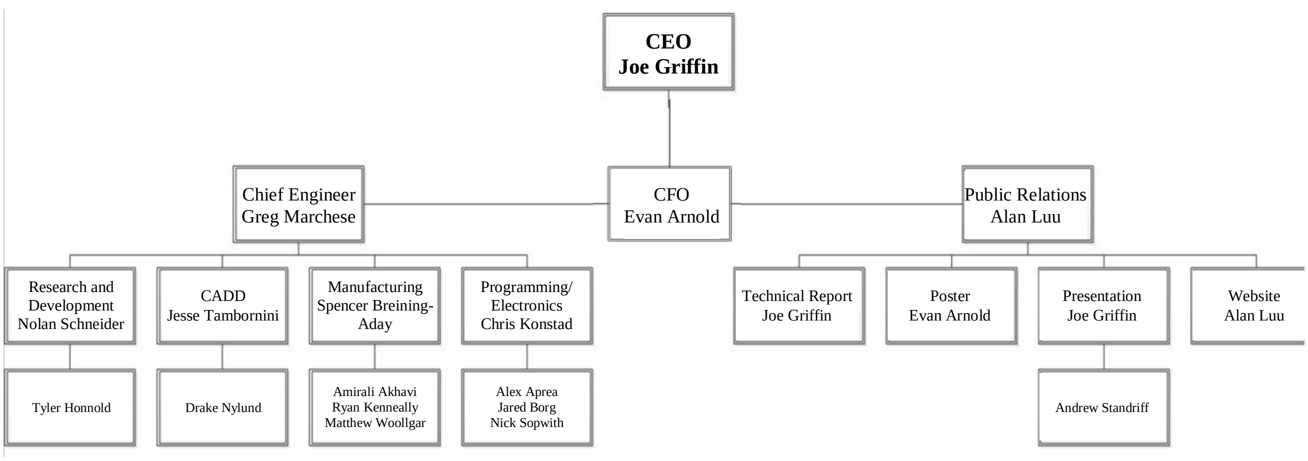


Figure 1: ROVotics Organization

Design Rationale

Frame

The frame, a rectangular aluminum skeleton with holes drilled at 1” (2.54 cm) intervals along every horizontal beam and at ½” (1.27 cm) intervals along every vertical beam, was initially modeled in SolidWorks® and then refined and constructed to fit the finalized mission accessories. The dimensions of the frame were designed in inches so the number of holes could be adjusted easily during the design phase. *Triton* is 17.5” (44.5 cm) wide by 25” (63.5 cm) long by 17.25” (43.8 cm) tall. The frame has high density polyethylene (HDPE) skids along the lengths of the lower left and lower right beams to prevent damage to the testing facility. The horizontal rear beams are shifted for the rear electronics container and skids. The front mission camera is supported by a third beam across the top and a smaller beam that connects it to the front. To make the frame more rigid, diagonal braces were secured across the top corners of the front face. A CADD diagram of the frame is provided in Figure 2.

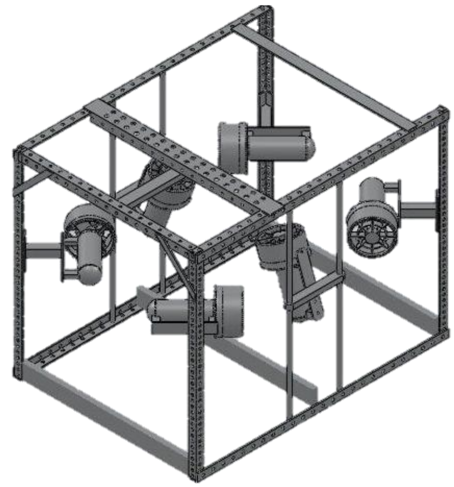


Figure 2: Triton *Frame*

Thrusters

Triton uses six Seabotix™ thrusters chosen for their size, power, and reliability. Four thrusters are angled across each of the frame’s four corners (Figure 2) to enable vector control, giving *Triton* the ability to translate and rotate simultaneously in any direction in the XY plane. The software for vector control was developed by ROVotics (see Appendix B) and provides the ROV with over 2.5 times the thrust of a single thruster when moving horizontally. The thrust profile of a single Seabotix™ thruster is given in Figure 3. The two remaining vertical thrusters are positioned at angles tilted inward above either side of the ROV’s center of mass to prevent downward flow from impinging on the accessories and are controlled by a separate switch on the pilot’s joystick.

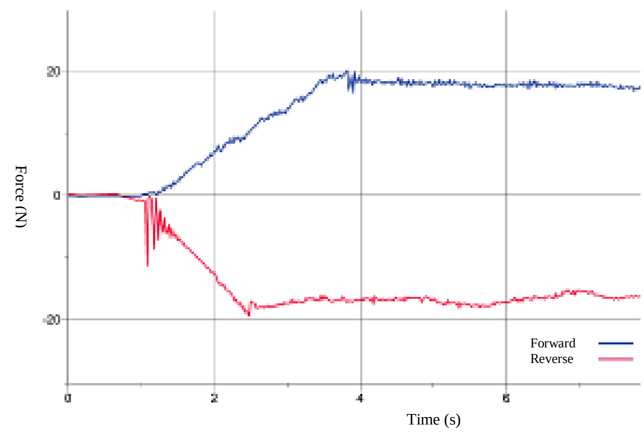


Figure 3: Seabotix™ Thrust Profile

Buoyancy

Triton is equipped with a composite foam block cut to balance the ROV and to a volume calculated to bring it to neutral buoyancy. This block, covered in fiberglass to prevent waterlogging or implosion, is attached to the top of the ROV with knobs for quick, tool-free removal for servicing. The volume of the block was found by dividing the weight of the ROV in water by the density of water and acceleration of gravity, resulting in a volume of 0.15 m3.

The forward section of the block is shaped to allow access to the forward electronic pods. The narrow middle section allows unhindered flow from the vertical thrusters. The split in the rear section allows the team to access the tether connectors on the rear electronics container. The volumes of the front and rear sections are adjusted to keep the center of buoyancy above the center of mass. Figure 4 provides a CADD image of the buoyancy unit without the bolts on the front and rear sections.

Figure 4: CADD Image of Buoyancy Unit

Pneumatics Systems

Triton is equipped with a pneumatics system to inflate the lift bag, transplant the coral, and separate the electronic compass from the ROV frame, this system is run off a 276 kpa air supply controlled with two solenoid valves that allow flow when energized. One valve controls air flow to the lift bag device, and the other sends air to the digital compass housing, lifting the compass away from the metal frame that would interfere with the orientation reading. The second valve also sends air to the coral collector to lift the coral out of the basket when the ROV is above a designated grid square. Figure 5 provides a diagram of the entire pneumatics system.

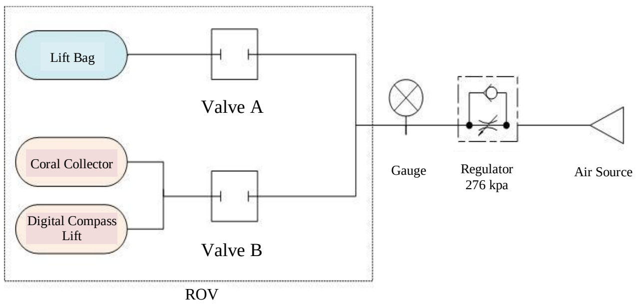


Figure 5: Pneumatics System

Electronics Containers

In previous years, ROVotics has used one large electronics container mounted at the center of the top of the ROV to provide buoyancy and stability. This year, however, ROVotics has separated the electronics into three containers to keep electrical interference from the high-voltage relays and electronic speed controllers (ESC) from disturbing the low-voltage microprocessors. Splitting the electronics containers also allows for more stable, centralized buoyancy and flexibility in the exact location of the containers, rather than directly above the center of mass.

The tether, thrusters, and all accessories except for sensors are connected to the rear electronics container using SubConn® underwater connectors. In total, the container houses six brushed speed controllers, a video balun, five control relays, two 48V to 24V DC-DC converters, and a 24V to 12V battery eliminating circuit (BEC) voltage regulator. The two front containers house only Arduino Ethernet microprocessors. The sensors connect to the right front container, which houses ROV2, the sensor Arduino. ROV1, for thrusters and servos, is housed in the left container with ROV3, the relay and LED microprocessor.

Electronics

Processing

The ROV has three onboard Arduino microcontrollers with Ethernet network interfaces for system control and telemetry. Having three onboard processors offers more I/O pins and the ability to execute several tasks simultaneously. Critical motor control functions can also be separated from sensor readings and accessory control, improving reliability and safety. Each Arduino connects to a custom-built printed circuit board (PCB) that breaks out all of the pins to locking connectors and reduces the volume of required wiring, making maintenance much easier and reducing the risk of loose wires. The design of the PCB is provided in Figure 6.

ROV1 controls the motors and servos with pulse width modulation (PWM). ROV2 reads and returns the data from the onboard sensors. ROV3 controls relays and RGB LEDs. The LED color indicates the status of the ROV's communication. All three Arduinos communicate with the topside application via user datagram protocol (UDP) broadcasting. UDP broadcasting keeps the control system simple and robust. Finally, the system requires no end point IP address, so audience members with the custom ROVotics applications for Windows, iOS, and Android can view telemetry information and the status of the ROV's main systems as listeners on the network.

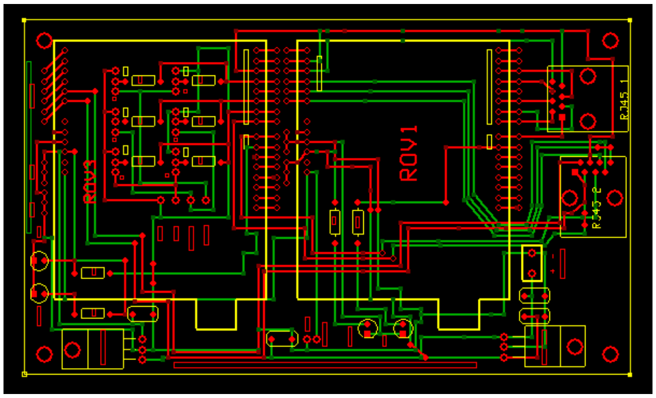


Figure 6: Onboard PCB Design for Arduino Connections

Electronic Speed Controllers

The brushed Seabotix™ motors are controlled by ROV1 via Sabertooth dual ESCs. The ESCs are powered by a 24V bus and interface with the Arduino via an analog signal. Arduino has a servo library that uses PWM to send predetermined commands to the thrusters.

Programming

Source Code Management

ROVotics implemented a new source control system this year using an open source application ("Git"), which allows collaborative software development among multiple employees while carefully managing and tracking the code development process. This allowed the team to utilize the skills of several engineers to develop the software for the topside code and 4 embedded controllers (3 in the ROV, 1 in the TCU) in parallel. Also, to increase the readability, modularity, and integrity, many blocks of code were split into sub-routines or libraries to provide abstraction for easier understanding and maintenance.

Bottom-Side Code

The bottom side software, like all software, has a specific flow of functions from which it cannot deviate. After the Arduinos complete their respective "setup()" functions, which initialize all objects and variables, each one begins its individual "loop()" function. The loop serves as the main body of the code, in which other functions are called. In the loop, each Arduino checks for UDP communication and reads any packet it finds.

The loop for ROV1, which controls motors and servos, first checks the values to be within safety limits, then outputs safe values to the motors and servos. For added safety in the event of lost communications, ROV1 will shut down all motors if no valid packet is received in the past 1.5 seconds.

The ROV2 loop, which interprets and returns sensor values, detects and zeros out available sensors as part of its "setup()" function. It then sends values up the tether to be checked by the topside software to see if they make sense. This ensures that the pilot does not receive nonsensical data either from instantaneous outliers or incorrectly calibrated sensors.

ROV3 controls relay signals and RGB LEDs. The ROV3 loop receives relay values and RGB values, if applicable, and then outputs corresponding values to the relays and LEDs. In the event of lost communications, as with ROV1, ROV3 will shut down all accessories if no valid packet was received in the past 1.5 seconds. Appendix C contains flowcharts for all three Arduinos.

Vector Thrust Control
To maximize *Triton*'s maneuverability, the control software utilizes vector thrust control, which converts a joystick value into a vector whose components are modified for 45° angles and distributed to the horizontal thrusters. ROVotics' topside human interface controller is a joystick with vertical (z) and horizontal (x, y, and w-rotational) control. The horizontal control input signals must be converted to motor values for each thruster for synchronized movement. A null zone prevents slight joystick deviations from maneuvering the ROV. The horizontal x-y input must be incorporated with the rotational w input to create a smooth integrated movement. The development of the control equations can be found in Appendix B.

Top-Side Code
Triton is controlled via a poolside laptop running a custom C++ application, "Catalina", written in the QT framework. The goal during the development of Catalina was to lead the pilot to feel like he has direct control of the ROV, with no software bridge. This feel is simulated by our vector drive control system, by which we can move the ROV in any direction the joystick can move. Catalina and all the onboard Arduinos send a minimum of 20 packets per second, allowing the ROV to react as fast as its pilot, for smooth control. In addition, to aid the pilot in keeping the ROV on target while scanning the ship, two proportional-integral-derivative (PID) algorithms maintain the ROV's depth and heading. The depth PID algorithm, in combination with the depth sensor, can keep the ROV at a specified depth within about two centimeters of accuracy. The pilot can shift the ROV up or down by one centimeter if he is off target. The heading PID algorithm takes the value of the digital compass as an input and rotates the ROV to keep the heading on target. It also disables the joystick's rotational input so that the pilot cannot accidentally rotate the ROV off target. By holding the depth and the heading, the pilot is only concerned with translational movement in the XY plane, which is handled by our pilot with his vectored thrust system. Appendix C contains a flowchart for Catalina and a logical connection diagram for the entire *Triton* software system.

Graphical User Interface
Catalina has a GUI to display information from the ROV and help programmers with debugging. Refer to Figure 7 for a screenshot of the Catalina GUI. Catalina itself takes input from a USB joystick and the GUI to control the ROV. It sends and receives packetized data through a custom UDP.

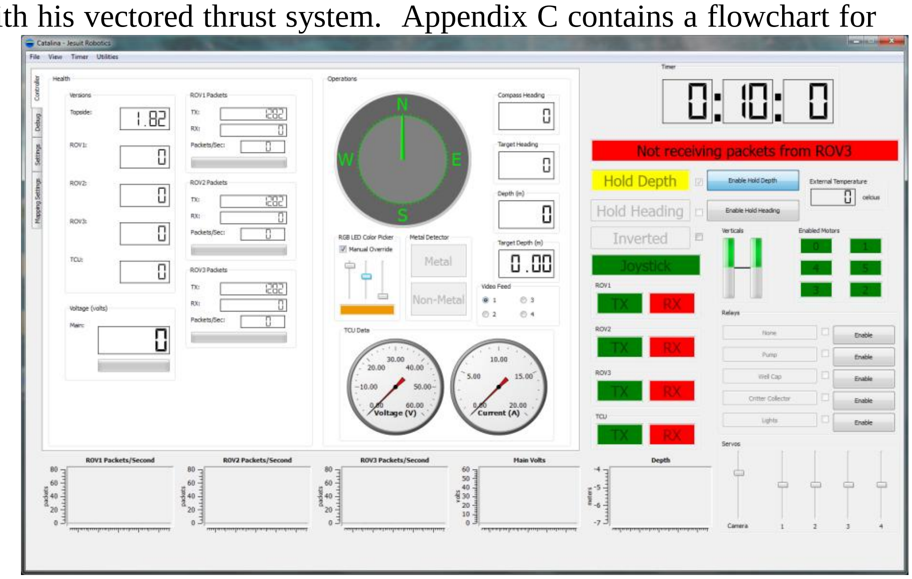


Figure 7: Catalina GUI

Catalina was developed from scratch to be robust and tolerant of user error. It alerts the user with red lights if any one of the three Arduinos or the joystick stops communicating. An explanation of the error is then displayed in a text box at the top of the screen. The error system is designed to cycle through the displayed errors so that no one error can hide another. The simplicity of the communication protocol prevents disturbances, except in the case of a physical disconnection. Catalina broadcasts UDP packets to different ports to reduce the amount of information needed about the end points and minimize communication errors.

Tether/Tether Control Unit

The heart of *Triton*'s Command, Control, and Communication (C3) network is the tether control unit (TCU) at the surface. The TCU is connected directly to the control laptop and all monitors to receive electrical connections from the tether at the surface. A physical connections diagram of the entire *Triton* system is provided in Figure 8. In order to activate the ROV, a 20A circuit breaker must be switched on as well as our main power switch. Voltage and current meters allow the co-pilot to easily see how much power the ROV is using. Each of the video signals is split into two signals with one going to a video multiplexer, a custom-designed and built quad video switching circuit board. The design for this board was found in a DIY article in Electronics Now Magazine (Ref 5) and modified to

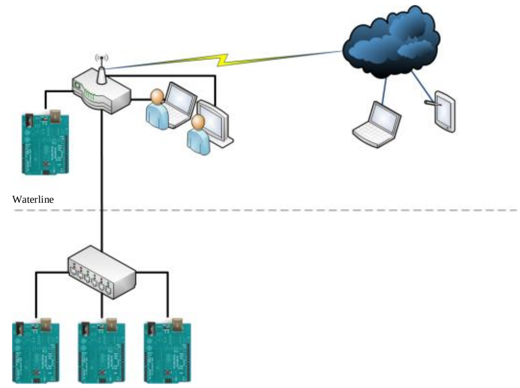


Figure 8: Full-system Connection Diagram

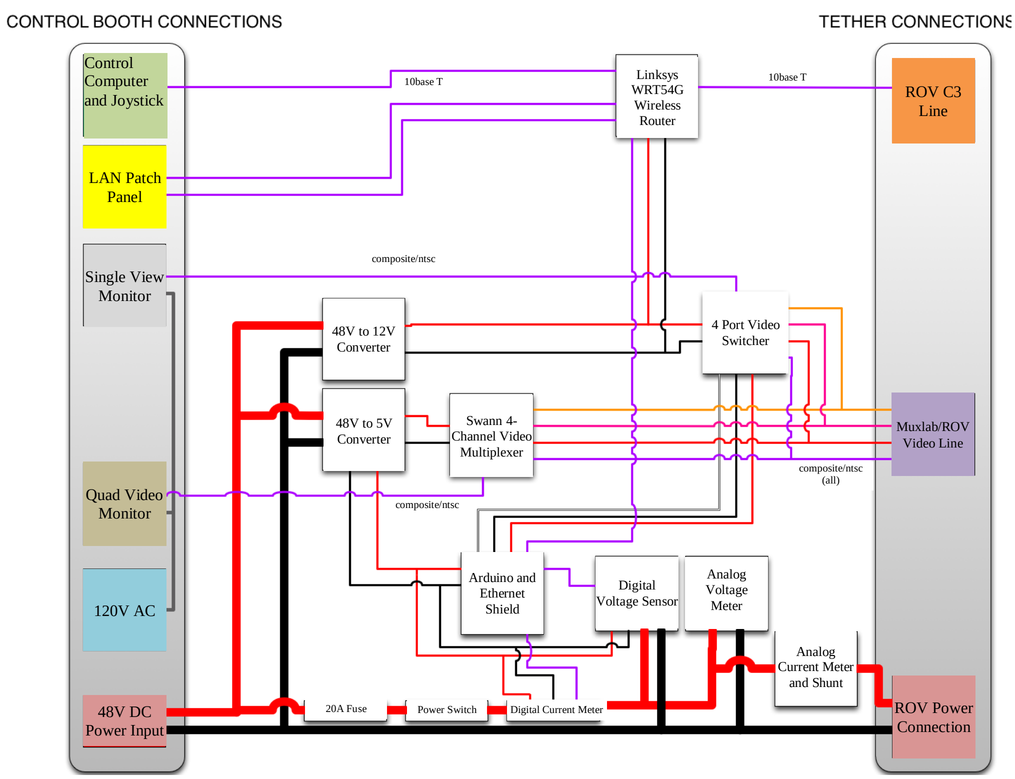


Figure 9: TCU Block Diagram

fit the DC power supply. Custom PCBs were designed and built using ExpressPCB software. Figure 9 provides a block diagram of the TCU, and Figure 10 provides the design of the PCB. A schematic of the PCB is available in Appendix A. The multiplexer allows the pilot to see the video feed from all four cameras on the ROV on a single monitor, and the switcher allows him to enlarge one specific feed. Figure 11 shows the control panel on the TCU.

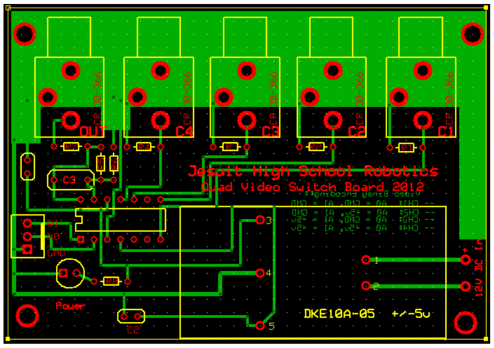


Figure 10: Video Multiplexer PCB Design

The tether consists of two CAT5 data lines, one 12AWG DC power line, and one 1/8” (3.175 mm) pneumatics line rated to 276 kpa. One CAT 5 Ethernet cable carries data packets to and from the Arduino microprocessors onboard *Triton*. The second transmits NTSC/Composite video with video baluns at each end. The power line is certified to carry 48V at 40A to *Triton* to allow for a power overload of double the normal current. Onboard voltage regulators supply 24V, 12V, and 5V power to *Triton*’s subsystems.

Figure 11: TCU Front Panel

Video System

Triton’s video system design is based on high resolution, wide field of view cameras, allowing the pilot to see the entire operating area without moving the ROV. All video cameras have 640 to 780 lines of resolution providing nearly twice the resolution of a more common 480-line system. This was achieved with two wide-angle cameras facing the front and back of the ROV, respectively. The front camera is mounted on a servo in the clear acrylic tube and can be tilted through 180° of vertical rotation. With a 140° field of view, no pan function was necessary. The rear electronics container houses a fixed camera with a 160° field of view. These serve as our main drive cameras. The proprietary housing system also contains two mission-specific cameras. The feeds from these four cameras are sent to the surface via a video balun, which extends the distance for NTSC video signaling over a CAT5 Ethernet cable. Topside, the Ethernet cable connects to the TCU video balun, splitting the feeds into four individual cables.

Safety

Safety is a major concern of ROVotics, and several steps are taken to ensure that incidents are avoided or minimized. At the beginning of the year, the company holds a training day to give a safety presentation, and senior employees give hands-on training to new hires. Additionally, during testing, a fuse is always connected to the ROV's electrical source to prevent overload. All electrical equipment is powered through a ground fault interrupter to prevent any damage. Before the ROV is deployed, the deck crew reviews a checklist to avoid overlooking important pre-launch details. Deck crew members are required to follow certain safety procedures and verbal communication protocols, such as calling out when certain electrical or pneumatic connections are being completed or activated. Finally, all thrusters are shrouded to reduce the possibility of objects being caught in the propellers.

Triton itself is protected with shutoff states integrated into the electronics and software that will trigger under predetermined circumstances. The topside software will send neutral motor values to the ROV if the joystick is disconnected so the ROV will not self-navigate in a random direction when communication resumes. Onboard microcontrollers are programmed to shut down all thrusters and accessories if communication is lost with the topside. ROVotics code is designed around maintaining the ability to disconnect from *Triton* at any time to correct and restart communication.

Mission-Specific Accessories

Linear Measurement System (LMS)

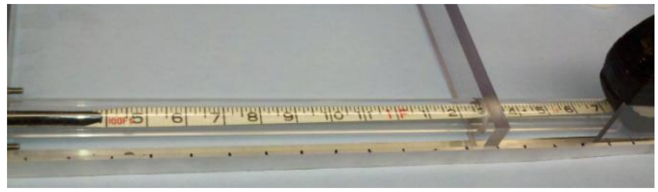


Figure 12: LMS

The boat length is measured with a simple tape measure that runs down an acrylic tube (Figure 12). A camera shows the measurement through the tube. Acrylic, which has an index of refraction very close to that of water, will not distort the camera's view of the tape measure, allowing the pilot to easily read the measurement from the camera feed. A tension reel ensures the tape measure will be taut during the measurement.

Scanning Beam

To assist the pilot with scanning the ship, *Triton* is equipped with an illuminator beam that shines at 90 lm (Figure 13). The beam uses a tactical LED flashlight. The flashlight draws 200mA at 5V, and can be powered by either the 12V or 24V bus on *Triton*; an internal voltage regulator and current limiting resistor replace the internal battery supply.

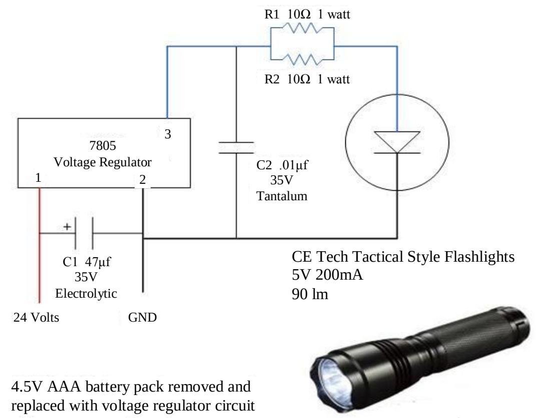


Figure 13: Scanning Beam Diagram

Coral Payload

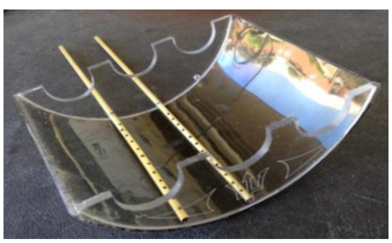


Figure 14: Coral Payload

For the coral removal, the ROV is outfitted with a curved polycarbonate rectangle with semicircles glued to the curved edges, as shown in Figure 14. The slots in the end semicircles are designed to catch onto the base of the coral and separate them from the hull of the ship. Once removed, the coral can be lifted out at any time with a burst of air from tubes connected to pneumatic lines. Air bubbles will adhere to the porous surface of the coral, gently lifting it out of the basket.

Fuel Oil Retrieval System (FORS)

Triton's FORS is a single 4.5" diameter cylindrical tank in a closed loop system that contains salt water on the outflow side of a diaphragm with an O-ring seal that moves across the tank and creates a pressure differential to push the fuel out of the tank on the ship. Figure 15 provides a diagram of the FORS loop. Compression seals on both ports of the

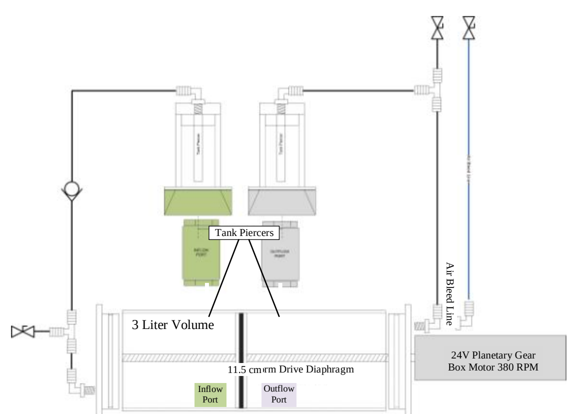


Figure 15: FORS Diagram

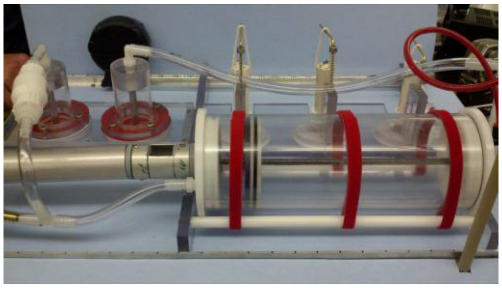


Figure 16: FORS

FORS maintain this pressure differential in the loop. The liquid in the tank flows through the inflow port of the collector and is pushed by the saltwater entering the tank through the FORS outflow port. A 380 RPM motor rotating a jack screw slowly pushes the plate across the tank. Ball valves in the inflow and outflow channels allow the crew to fill the fuel reservoir prior to the mission and to drain excess fluid afterward. At full rotational speed, the plate moves at 1 cm per second. With a tank of radius 5.4 cm, the maximum possible flow rate is approximately 93 cm^3 per second, completing the removal in 21.5 seconds. In line check valves prevent back flow of retrieved fuel once FORS has been disconnected from the fuel tank. To prevent dilution and potential backflow, the sample will be pumped very slowly, avoiding any turbulence and taking advantage of the difference in fluid densities. *Triton*'s FORS can retrieve a total of 2 L of oil. Figure 16 provides an image of *Triton*'s FORS.

Metal Detector

The metal detector tool is an integrated Frequency Beeping Oscillator (FBO) modified for undersea operation and *Triton*'s C3 network. The battery was replaced with a dedicated 9V voltage regulator and a LED replaced the detection buzzer. A signal line was taken from the positive side of the LED and connected to the analogue input port of one of the Arduinos. An active ferrous metal sample will be visually apparent on the video feed because an LED will be activated and metal signal strength will be displayed on the pilot's ROV GUI interface. The FBO detects metal by sending out radio waves that return at a higher frequency after being deflected by metal, causing a spike in frequency recorded by the metal detector and activating the LED and ROV GUI interface. Refer to Figure 17 for a picture of the metal detector mounted on the ROV.

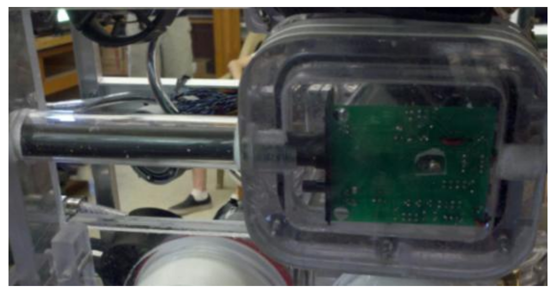


Figure 17: Metal Detector

Lift Bag

The mast will be moved with a lift bag chained to a self-locking gate latch mechanism. To move the mast, the ROV connects the lift bag and inflates it to bring the mast to neutral buoyancy, so it can be easily pushed away from the ship. Initially, the bag will be empty and packed into a short length of pipe to minimize drag. After attaching the bag to the mast, air pumped down the pneumatics line in the tether inflates the bag. Since the mast weighs between 50 N and 75 N in water, we designed our lift bag to displace about 8,100 cc, offering 80 N of buoyant force. The lift bag is also fitted with a valve to bleed air and slowly lower the mast when the air supply is deactivated. The valve opens automatically when the air feed is disconnected, and the mast and bag slowly sink to the designated area. Figure 18 provides a picture of the lift bag.

Figure 18: Lift Bag

Mission Objectives

Surveying the wreck site

Measuring the length of the wreck
A tape measure with a hook on the bottom right rail of the ROV is scraped against the back rail of the boat. A camera aimed at the tube containing the first two feet of the tape measure shows the pilot the length of the boat when he maneuvers to the far end, then back up using inverse control logic and rear camera to retrieve.

Determining the orientation of the ship on the seafloor
The orientation measurement requires that the ROV not rotate at all. To achieve this, an electronic compass read by an orientation feature of the software allows *Triton* to maintain the same orientation as the boat for measurement. The ROV needs only visually align with the boat.

Creating a map of the wreck site
A map of the site will be drawn based on the length and orientation data provided by the earlier two measurements by the mission manager on the deck crew. The mission manager will also be able to use the pilot's video feed to determine how to sketch the wreck and where to place the debris piles.

Determining if debris piles are metal or non-metal
To complete this task, we mounted a modified FBO metal detector on the front of the ROV, far from any metal components to prevent unwanted noise. This metal detector activates an LED and a message in the GUI. The pilot then maneuvers the ROV to each debris pile and waves the antenna over the pile. Upon activation of the GUI or LED, the co-pilot marks the pile on the map.

Scanning the shipwreck with sonar
This task entails "scanning" the ship at three target locations, where we need to maintain vision of an entire ring inside the target, forcing the ROV to be perpendicular to the hull for 10 seconds. Any water current or leftover momentum may cause a slight shift, resetting the clock and wasting time. A depth sensor allows the software to maintain the ROV's depth and the heading hold feature ensures that the ROV remains perpendicular to the hull.

Removing fuel oil from the shipwreck

Transporting and attaching a lift bag to a fallen mast
When the mission starts, the lift bag is packed into a short tube out of which it inflates to lift the mast. The bag is connected to the U-bolt by a gate latch chained to the lift bag, which lifts the latch off the fork on which it rests during the ROV launch.

Inflating the lift bag and removing the fallen mast from the worksite
The bag is inflated with air from a pneumatic line mounted beneath it to make the mast positively buoyant for a short time, long enough for the ROV to move it to a designated area. Once the air flow stops, the bag leaks air and the mast slowly sinks back to the floor.

Removing endangered encrusting coral from the ship's hull
The coral is removed with a basket with semicircles cut out of the front edge that match the diameter of the coral base. The pilot maneuvers the payload container under the coral and removes it.

Transplanting the coral
Two perforated tubes in the bottom of the coral basket release a burst of air that lifts the coral out of the payload, allowing the ROV to back away and let the coral sink into the appropriate grid square.

Using two simulated sensors, determine if fuel oil remains inside the fuel tank
The ROV is equipped with two devices to determine if fuel remains in the ship's tank: a neutron backscatter device and an ultrasonic thickness gauge. The gauge sends high-frequency sound waves through the hull and calculates its thickness based on the time taken to return. These readings are then used to calibrate the neutron backscatter device. The device detects the presence of "thermal neutrons," created when high-speed neutrons collide with hydrogen; substances with high hydrogen concentrations have more thermal neutrons. The backscatter device counts the number of thermal neutrons and determines whether or not the tank contains fuel oil.

Simulating drilling two holes into the hull and underlying fuel tank by penetrating a layer of petroleum jelly
Should the fuel tank contain oil, the ROV is equipped with a pneumatic fuel pump, which can simultaneously extract the fuel from the tank and refill it with seawater. Each port on the pump is equipped with a stainless steel tube extending down the middle, which is used to both puncture the fuel tank's seal and reach inside the tank to pump the fuel out. The motor dislodges excess petroleum jelly in the tube, allowing for easy access and quick recovery of the oil in the fuel tank.

Removing fuel oil from within the tank and replacing it with simulated seawater
The ROV first slides two tubes with compression seal diaphragms over the inflow and outflow ports of the fuel tank. A motor rotates a jack screw and pushes a plate along the length of the cylindrical tank, originally full of seawater. The moving plate creates a high pressure environment on the seawater side and leaves a low pressure environment for the oil. Check valves on the inflow and outflow tubes ensure the fluids will only flow in the correct direction.

Resealing the drill holes with a simulated magnetic patch
A piece of polycarbonate on the bottom of the ROV serves as the magnetic patch deployment system. Two thin brass rods bent at 90° angles extend straight up into the body of the ROV, with the string on the end of the patch looped over the bent end. Springs force the rods up, using the friction of the patch and polycarbonate to keep the patch from sliding out of a slot cut into the plastic sheet until the patch is placed on the oil tank port.

Competition Logistics

Build Schedule

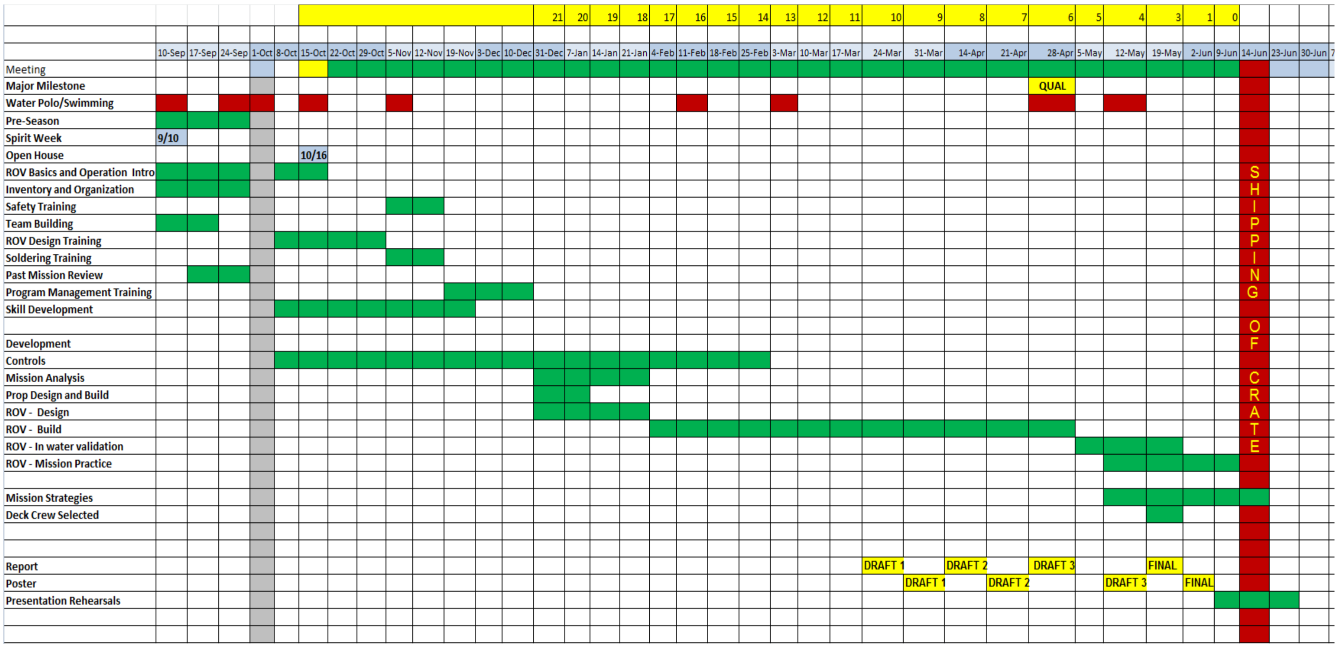


Figure 19: Gantt Chart

With workdays limited to 7-hour Saturday shifts, the team needed to adhere to a strict build schedule to which to adhere. A Gantt chart, provided in Figure 19 above, was developed based on the dates of the competition details reveal, qualifier, and competition. The Gantt chart covered deadlines in the following four categories: pre-season preparation, brainstorming, building, and testing. Before the mission details were released, we prepared by organizing the lab, training recruits, and fabricating non-mission specific components. After the mission was released, we planned project development with concrete dates. In addition to these time constraints, we developed a presentation for Underwater Intervention in New Orleans, delivered to ROV industry professionals. Preparation for this presentation, which earned us several offers of inventory donations, had to be planned into our preseason as well.

Budget

We had to be careful with our limited financial resources. At the beginning of the season we budgeted all our expenses and adhered to the plan as much as possible. The team would not run any fundraisers, so the team expenses needed to be covered solely by donations. We tracked our expenses closely and ensured no extraneous costs were incurred. Table 1 provides all major incomes for the team, and Table 2 shows all major expenses. The cost of the ROV alone, including all prototypes, was $3,880.00.

Description	Amount
Jesuit High School	$14,000
QSP Magazine Sales	$3,000
Dues (18 members x $100)	$1,800
PACE Auction	$700
Zinn Family Donation	$1,000
MATE Explorer Class Overall Winner Award	$3,000
	23,500

Table 1: Revenues

Description	Amount
CNC Mill	$2850.00
Research, Development, and Demonstration	$940.00
Bulk materials	$800.00
Electronics	$930.00
Tether and TCU	$740.00
Cameras	$70.00
Other ROV expenses	$400.00
Lab Equipment (Tools and Replacements)	$300.00
Shirts	$200.00
Prop Construction	$500.00
Traveling, Lodging, Shipping, MATE Fee	$13,025.00
	$20,755.00

Table 2: Expenses

Conclusion

Challenges

In previous years, ROVotics has been led by a single team member who has taken on alone the tasks of overseeing technical writing, manufacturing, and budgeting. While he was able to divide these tasks among the rest of the team, the responsibility for the quality of the ROV, technical report, poster, and presentation always fell on the same person, who would spend the hours before the deadline for the technical report writing frantically and trying to find the extra character in the code. Rather than force one person to suffer this ordeal, the fourth year members agreed before the mission reveal to divide the workload. Greg Marchese became the engineering director and oversees the production of the ROV itself. Greg's work has made *Triton* a powerful, effective machine. Evan Arnold is the operations director, in charge of running shipping, ordering parts, and handling logistics for the team. Evan's efforts made ROVotics productive and efficient every workday. Joe Griffin took on the responsibility for the technical report, presentation, poster, and press release, as well as the role of competition manager. Dividing the administrative tasks in this way allows the ROVotics administration to produce an excellent product, competitive at the highest level, on time and under budget and market it as a high-quality product from a professional company.

One of the greatest challenges each year is manufacturing watertight enclosures for our critical electronic components. This year we experienced failure due to design errors and manufacturing quality. In designing a casing for the front camera, we chose an acrylic tube with an O-ring based bayonet seal on the outside of two identical end caps. The two end caps were turned on a lathe and the O-ring grooves carefully cut into the aluminum. Preliminary testing showed good results with a tight fit and what appeared to be solid contact along the O-ring. On further in-pool testing however, the housing leaked. The leak was caused by discrepancy in the interior diameter of the commercially purchased acrylic tube in addition to slight lateral grooves along the bottom of the O-ring groove. These grooves were created by inevitable tool "chatter" in the lathing process. We modified the design to use a face seal, which could be manufactured easily on a CNC mill and bypasses the problems both in our manufacturing processes and in that of the acrylic tube's manufacturer. The new design was and is successful in all testing.

Troubleshooting Techniques

Electronic PODs were designed with face seals with required custom O-rings from raw O-ring material. Completed PODs were submerged, pressurized to 12 PSI, and monitored for small bubble leaks. Without a pressurizing testing tank this reverse approach, although not ideally matching the design characters of the PODs, proved to be a very effective way of identifying manufacturing and quality issues. From this testing process we learned that O-ring grooves must be clean and free of defects, such as the tool chatter shown in Figure 20, to provide a good seal. When joining raw O-ring material, the material needs to be cut with a new, sharp blade and carefully joined with cynoacalate (CA) glue. After the gluing process, the glued joint can be inspected by rolling it under a piece of

Figure 20: Failed O-ring Seal

clear polycarbonate. If the cut is clean, both ends of the connections will clearly press against the polycarbonate sheet. Extra CA glue on the surface of the O-ring will cause a failure in the seal, but can easily be cleaned up with 1000 grit sand paper. When properly built, the O-ring should appear as a polished, continuous black line against the clear polycarbonate sheet.

Lessons Learned and Skills Gained

Every year, one of ROVotics' most important goals is to enact its corporate succession plan, which consists of seasoned employees passing on their knowledge to new trainees so that the company will continue to thrive in the future. Our new hires, for example, learned skills such as how to solder and de-solder, and how to use the different machines in the lab. Our seasoned employees also learned new skills. A few have mastered the art of machining parts on the CNC mill and can accurately make any part designed by the CADD department. The CADD department spent the beginning of the year learning about tolerances of parts and setting a standardized chart so all parts will fit together and have room for error. One of the primary goals of the design department was to standardize the hardware and raw materials used in the construction of *Triton*. Using only materials from a predetermined list allowed us to buy material in bulk, saving time and money. All fastening hardware is 8-32, 6-32 and ¼-20, and the limited availability of metric fasting hardware pushed us to maintain SAE standards for fastening. All other measurements and ROV specifications are provided in the MKS system to ensure seamless communication between CADD and fabrication. *Triton*'s frame is built from 3 basic stocks of aluminum material and 3 thicknesses of clear polycarbonate. Polycarbonate was specifically selected over acrylic for its flexibility, durability, and machinability. The ROVotics also learned how to utilize PID algorithms in writing code functions to maintain depth, along with using PCB design software to reduce wiring volume and save space in the electronics containers.

In addition to learning specific skills, the employees of ROVotics learned about the entire engineering process—seeing a project through from the initial concept stage to final testing and production. This included setting and maintaining a schedule. Members had to plan into the future and work together. They had to know what parts they needed for subsequent work days in order to maintain this schedule. This also meant that different departments had to work together when changes were made. For example, the electronics containers were designed for one type of speed controller, but after they were changed by the electronics department, the containers needed to be enlarged. These new skills show that ROVotics has a bright future ahead of it.

Future Improvements

Future sustainability has always been a primary goal of ROVotics. To create this environment, towards the conclusion of the season we brainstorm ways to ensure our team is in the best possible position for the following competition and for years to come. This year, we have proposed several developments that we believe will accomplish that goal.

Research and development is a key component of ROVotics' design process. To facilitate faster and more fruitful research in the future, we decided to maintain the post-competition ROV for design testing in the early stages of the following season. This scaffolding will allow members to quickly beta test possible developments, leaving more time for revision and refinement. Having a fully functional ROV will also provide a preseason opportunity to train new pilots and familiarize new members with deck operations.

Another improvement ROVotics is looking to make is in efficiency. In the past, we have used six brushed DC motors that have provided ample thrust at a reasonable power budget. However, next year, Jesuit Robotics plans to invest in Seabotix's new, brushless DC motors. These new thrusters will produce 100% more thrust while using significantly less power than this year's brushed motors, freeing up current for other accessories or instruments.

Jesuit Robotics' final improvement caters to efficient operation and aesthetics. In previous years, telemetry data and readings were displayed on the co-pilot's computer screen away from the pilot. Next year, in order to provide as much useful information to the pilot as possible, we plan to implement a heads-up display (HUD) on the video feed. By displaying sensory data such as temperature readings, compass headings, and depth in a neat configuration on the main video feed, the pilot will be able to make more informed decisions and ease the act of recording data required by the mission.

Senior Reflections

Alan Luu

Thanks to all the students, parents, coaches, and especially MATE for making the robotics program at Jesuit possible. My time on the team taught me more than I could ever imagine, from machining parts to managing a website. The most valuable experience was creating a feat of engineering and competing at international competitions with friends who also shared my passion for engineering. I'll major in physics at UC Berkeley, but I'll always keep the knowledge and fond memories of MATE.

Greg Marchese

Thanks to all of the students, parents, coaches, and the MATE center for creating, and ensuring that the robotics program and the MATE competition exist. As a fourth year member, I've been given the opportunity and motivation to further a childhood dream into something that will remain a lifelong passion. With the skill set I have developed, I have ensured a successful future, and plan to stay within the robotics field; I plan to major in biomedical engineering at UCSD. As the lead programmer and co-pilot on a robot that brought home an international title, I am certain that I will dedicate my life to engineering. As I move on to other types of engineering, I will always keep my robotics memories.

Joe Griffin

MATE has been the most powerful learning experience of my high school years. Through MATE, I've had the opportunity to give several technical presentations for corporate level ROV engineers, been interviewed on Good Day Sacramento about our ROV, been able to say on several of my college applications that my team beat their own students, and won a world-class underwater robotics competition against college-level teams. Aside from these, I've learned technical writing, machining, and computerized drafting skills and have started to learn about programming and electronics. MATE has taught me invaluable lessons that I will use heavily as an electrical engineering major at MIT.

Evan Arnold

Jesuit Robotics has taught me invaluable lessons. The opportunity to supplement my high school education with the MATE competition has been an experience for which I'm incredibly grateful. I'm thankful to the MATE Center for the opportunity to compete with universities internationally. I'll take my lessons from MATE to Stanford University and major in Cognitive Neuroscience.

Acknowledgements

None of this would be possible without the help of companies, our school, and different individuals. We would like to thank the following people and organizations for their support in this endeavor:

MATE—Sponsoring Underwater ROV Competition
Jesuit High School—Monetary Stipend
Mr. Rolf Konstad—Head coach who gave up many weekends to work with us
Mr. Jay Isaacs—Mentor for seven years
Mrs. Celine Isaacs – For weekly shopping trips, sewing projects for team spirit
Mr. Peter Brown—Assistant Coach for the first half of the season
Ms. Julia Yang—Faculty Liaison
Ms. Lyndi Marchese—Travel logistics coordination
Ms. Kim Arnold – Shipping logistics coordination
Mr. Mark Standriff—Presentation Coach
Mr. Brian Honnold—Parent Volunteer who helped in the lab
Mrs. Laurie Sopwith – Coordination of lunches and snacks
Mr. Jim Claybrook, "Weldmasters"—complimentary welding of boss onto enclosure
Helen and Neil Zinn — for monetary donation
Lund Family—used computer for controlling our CNC mill
Aprea Family—used computer for a CADD station
Seabotix™—Discount on thrusters
SubConn® Corp.—Underwater Connectors
SolidWorks®—SolidWorks® CADD Program
Fisheye Scuba—Discounted Scuba Tank Rental
Parents—Great food for all those long work days and emotional support!

References

1. "Mission & Specs." *ROV Education and Resources*. MATE Center, 2011. Web. 28 Apr. 2012. <http://www.materover.org/main/index.php?option=com_content&view=article&id=140&Itemid=199>.
2. Moore, Steve W., Dr., et al., eds. *UNDERWATER ROBOTICS: Science, Design & Fabrication*. Vol. 1. N.p.: Marine Advanced Technology Education, 2010. Print.
3. Parker Hannifin Corporation. "Parker O-Ring Handbook." 2007. PDF file.
4. Pohanish, Richard P., ed. *Machinery's Handbook Pocket Companion*. Vol. 1. New York: Industrial Press, 2008. Print.
5. Montegari, Frank. "Build a Video Switcher." *Electronics Now* September 1997. Magazine.

Appendix A: Electronics Diagrams

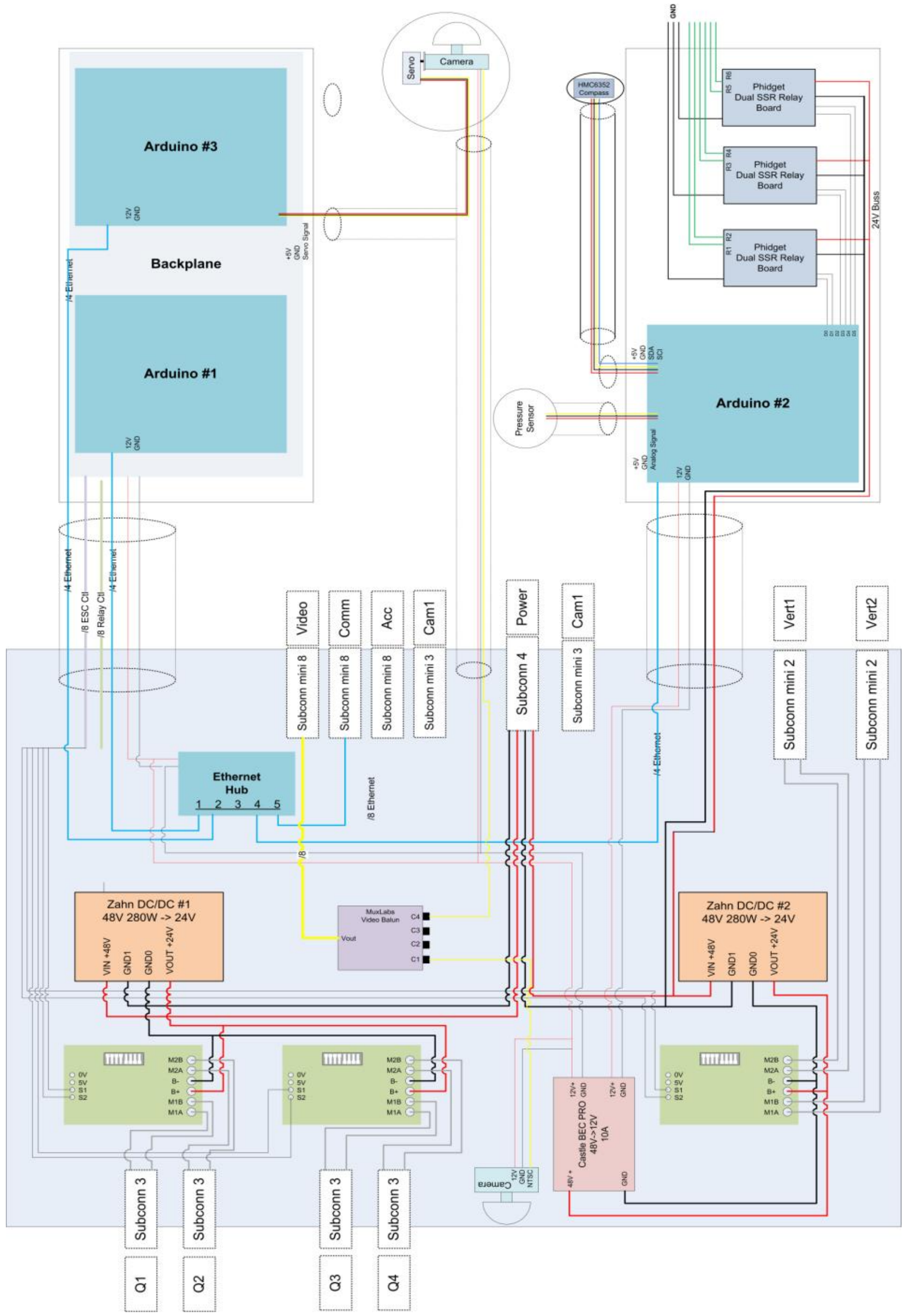


Figure A-1: Onboard Connections Diagram

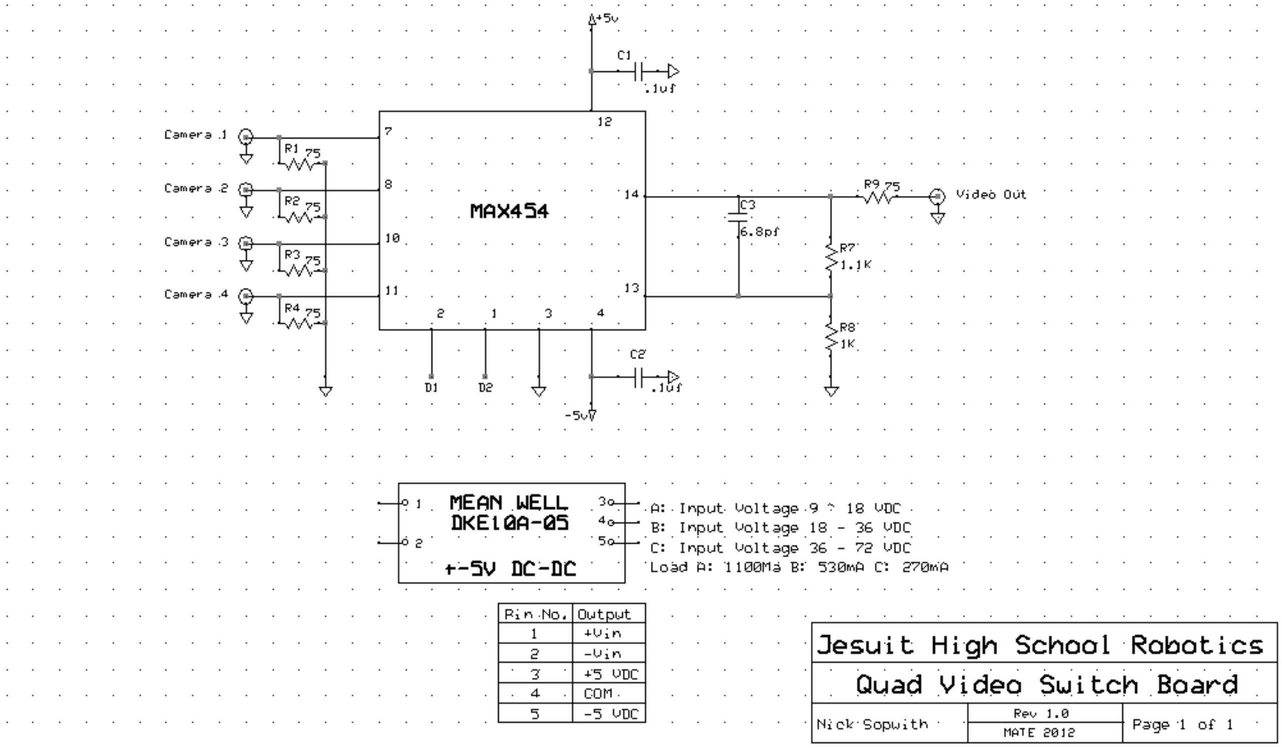


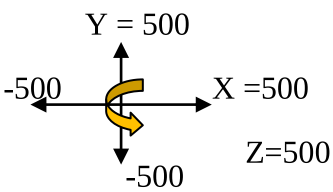
Figure A-2: Multiplexer PCB Schematic

Appendix B: Calculations for Vector Thrust Control

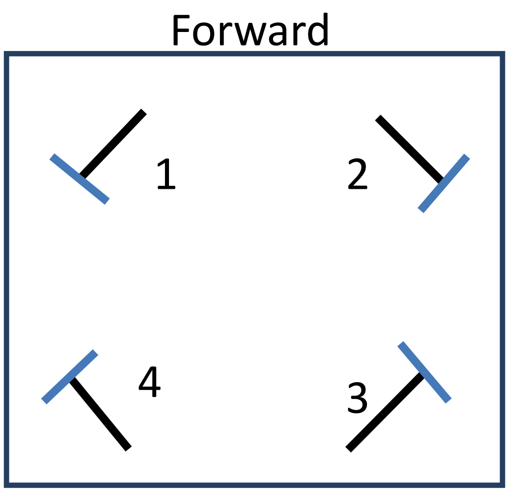
The strategy for this year's vector thrust system was to use the superposition method to determine the contribution of each x, y, z of the joystick to each motor independently, then normalize the resulting motor vectors to match the magnitude of the joystick vector. This method should greatly simplify the math. Results should be a seamless control regardless of direction or mixing and allow full power in any direction.

Theory

The joystick provides input for forward, lateral, and rotation



The motors are assumed as follows, where the positive direction is defined as water being ejected by the motor from the impeller over the motor casing, resulting in thrust where the impeller is forward.



First, built a table that shows the impact of a full scale deflection of the joy stick and impact on the motor command

	Joy stick input (full deflection)			Motor Command, % of full power			
Vector	X	Y	Z	Motor1	Motor2	Motor3	Motor4
Forward	0	500	0	-100	-100	100	100
Backward	0	-500	0	100	100	-100	-100
Trans R	500	0	0	-100	100	100	-100
Trans L	-500	0	0	100	-100	-100	100
Rot L	0	0	500	100	-100	100	-100
Rot R	0	0	-500	-100	100	-100	100

Or, another way to look at is how each motor direction relates to the stick direction

	X	Y	Z
Motor1	Inverse	Inverse	Same
Motor2	Same	Inverse	Inverse
Motor3	Same	Same	Same
Motor4	Inverse	Same	Inverse

Now, knowing how each motor contributes to a basic motion, more complex motions can be calculated simply by adding up the contribution of each direction. For example, to go diagonal (forward, left), the "Forward" and "Trans L" motor commands are added for each motor:

Motor 1 = -100 (F) + 100 (L) = 0

Motor 2 = -100 (F) - 100 (L) = -200

Motor 3 = 0

Motor 4 = 200

From the ROV diagram, the results make sense in terms of direction, but the magnitude is too large; motors cannot be run at 200%. Therefore, values must be "normalized", where all the motors are scaled to fall with their capability range. The highest % motor should match the highest % of full scale on the stick.

For each motor i,

Motor(i) = Motor(i) * (% highest stick deflection)/(% highest motor)

Example code for this calculation would appear as the following:

```
JoyStickRead(x, y, z)

        if (x< deadzone),
                x_percent = 0
        else
                x_percent = (100/(500-deadzone))*(x - deadzone); // find percent deflection with deadzone
        endif

        ...this process is repeated exactly for y and z...

        max_input = max (abs(x_percent), abs(y_percent), abs(z_percent));   //note highest deflection

MotorCalcs()   //
        motor1 = - y_percent  - x_percent + zpercent; //based on look up table above
        motor2 = - y_percent  + x_percent - zpercent
        motor3 = + y_percent  + x_percent + zpercent
        motor4 = + y_percent  - x_percent - zpercent
        max_motor = max (abs(motor1),abs(motor2),abs( motor3), abs(motor4))
Normalize ()  //scale motors so none receive command >100%
        motor1 = motor1 * (max_input/max_motor)
        motor2 = motor2 * (max_input/max_motor)
        motor3 = motor3 * (max_input/max_motor)
        motor4 = motor4 * (max_input/max_motor)

MotorCommand()    //generate motor command for packet
        motor1_command = 90 + (maxtrim/100)*motor1
        ....and so on
```

Appendix C: Program Flowcharts

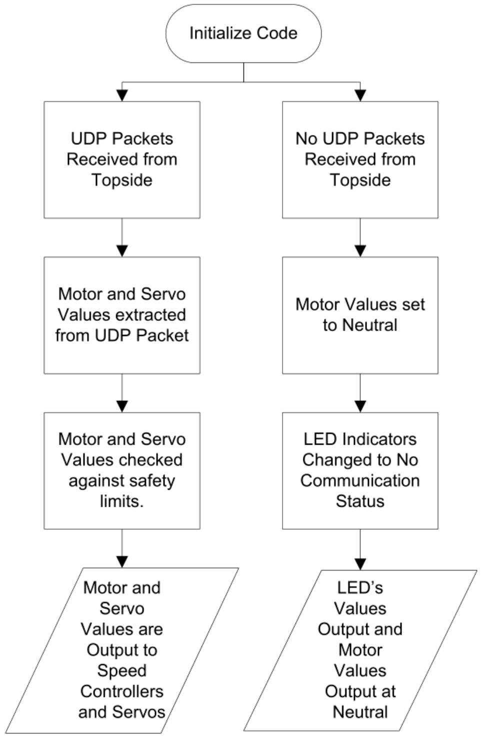


Figure C-1: ROV1 Flowchart

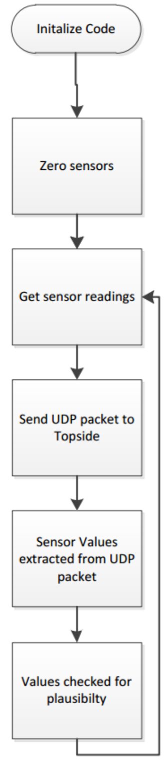


Figure C-2: ROV2 Flowchart

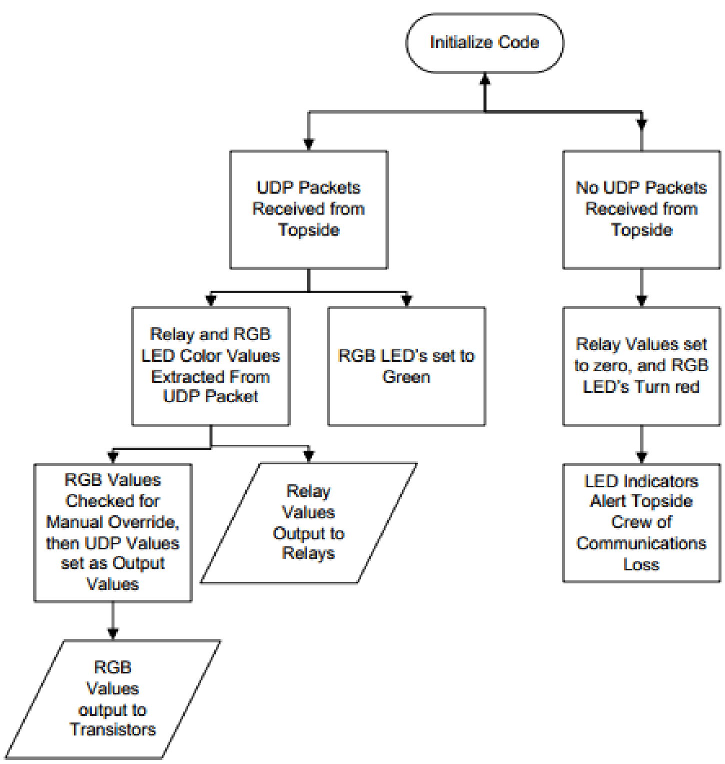


Figure C-3: ROV3 Flowchart

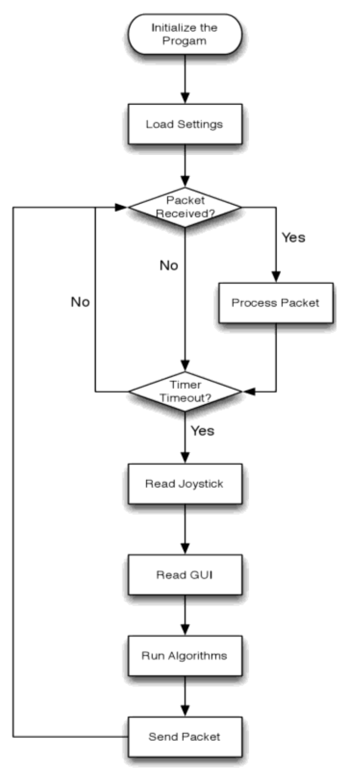


Figure C-4: Catalina Flowchart

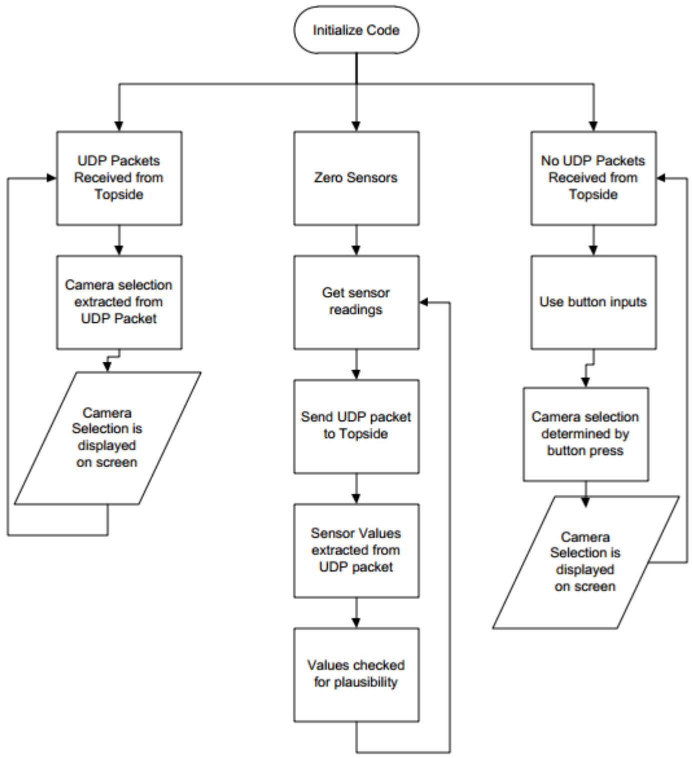


Figure C-5: TCU Software Flowchart

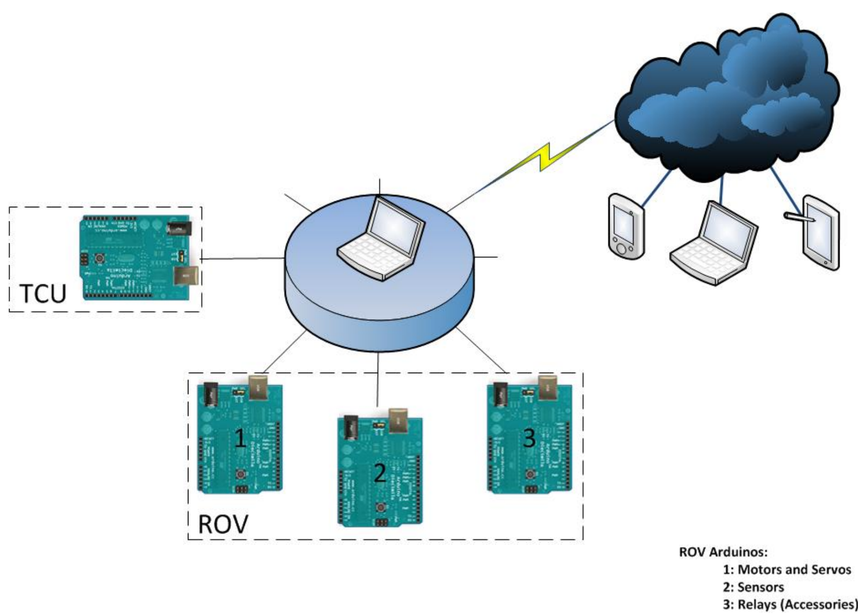


Figure C-6: Logical Connection Diagram for Entire System