Table of Contents

Executive Summary

The objective of this mission was to design one of five OSV's each tasked with exploring different aspects of a remote island that an airplane had crashed on. Our team's goal was to design the most efficient OSV that could complete the mission in a limited amount of time on a constrained budget. To be viable the OSV had to be capable of navigating on sand and over a rocky terrain, while avoiding obstacles in order to navigate to a predetermined mission site. Once there our OSV had to complete its mission specific tasks. The basic requirements specific to our mission were as follows: the OSV had to navigate within 250 mm of a pool of water, test the quality of the water and determine whether it is fresh, salt, or polluted water, and then transmit that data to the control center. Our OSV was also tasked with a set of advanced requirements which included: transmitting the depth of the pool to within 4mm and collecting a sample of water between 65-75mL. All of these tasks had to be completed within the time frame of five minutes.

Our OSV was designed to efficiently complete all mission requirements while remaining light weight, under 3kg, and cost effective. The chassis of the OSV was made of plywood which is lightweight and strong. The OSV drove using 4 large tires made of rubber with grooved treads to allow for maximum traction on sand and rocky terrain. Each wheel was powered by its own motor creating four drive wheels to ensure that the OSV had maximum maneuverability and power to get over the different terrain. The four motors, Arduino circuit board with motor shield, and other sensors were powered by a 12V NiMH battery. The Arduino circuit board controlled all vehicle movements and reactions through code. The navigation of the OSV used two ultrasonic sensors to detect obstacles in its path, while using an RF module to communicate the OSV's position to the overhead vision system. The quality of the water was determined using a conductivity sensor. The advanced objectives were completed using a water pump and an ultrasonic sensor based algorithm to test the depth and remove the specified volume of water. The conductivity sensor and pump tubing were lowered into the pool using a pulley system that was controlled by two servo motors. The components of the OSV allowed it to properly function and achieve both basic and mission specific requirements.

Introduction

Team Hydration Nation was tasked by an engineering firm to design an OSV to survey a remote island in the Pacific Ocean. This was necessary with the finding of airplane debris from missing flight MH370 on the island, and the media wanted to know if there were any survivors. Team Hydration Nation competed with other teams to design an OSV, with the specific task of landing on the island and autonomously navigating towards a sample of water designated by a command center in the United States. When it reached the mission site, the OSV tested and attempted to transmit whether the water was polluted, salty, or freshwater. This gave us valuable insight as to whether there were any possible survivors of the flight, and prompted officials to go to the remote island and conduct a search. The engineering firm had given our team a plethora of constraints. When considering the structure of the OSV, we had to keep the mass of the OSV to under three kilograms, fit it in a 350mm by 350mm space, and if we had used any kits, they could only satisfy up to two of the following sections: motors, wheels, chassis, suspension, transmission and motor controllers. When looking at power, the firm gave our team the following criteria: we could not use any lead acid or lithium batteries, the OSV had to run autonomously for at least 10 minutes, and combustion engines are prohibited. The power criteria were due to safety regulations, as well as the firm not wanting anything to catch on fire and compromise the mission. For navigation, the firm asked that our team used an Arduino microcontroller to control the various motors and sensors, as well as the wheels. The firm also asked that we have the OSV operate autonomously, and navigate to within 250 mm of the mission site. We had to navigate using the vision system which was located in the engineering firm’s headquarters. Our OSV, as Image 1 shows, was constructed out of a multitude of materials. We used a plywood chassis, and wheels exported from China. We 3D printed many of the necessary pieces to construct the OSV, such as the motor mounts and the Arduino holder. Wiring was done through two breadboard, an L298N h-bridge module, and an Arduino. We bought three ultrasonic sensors, a water pump, and made a conductivity sensor.These were attached to the OSV and connected to the Arduino so we could deploy them at precise times so they could perform their specific functions.

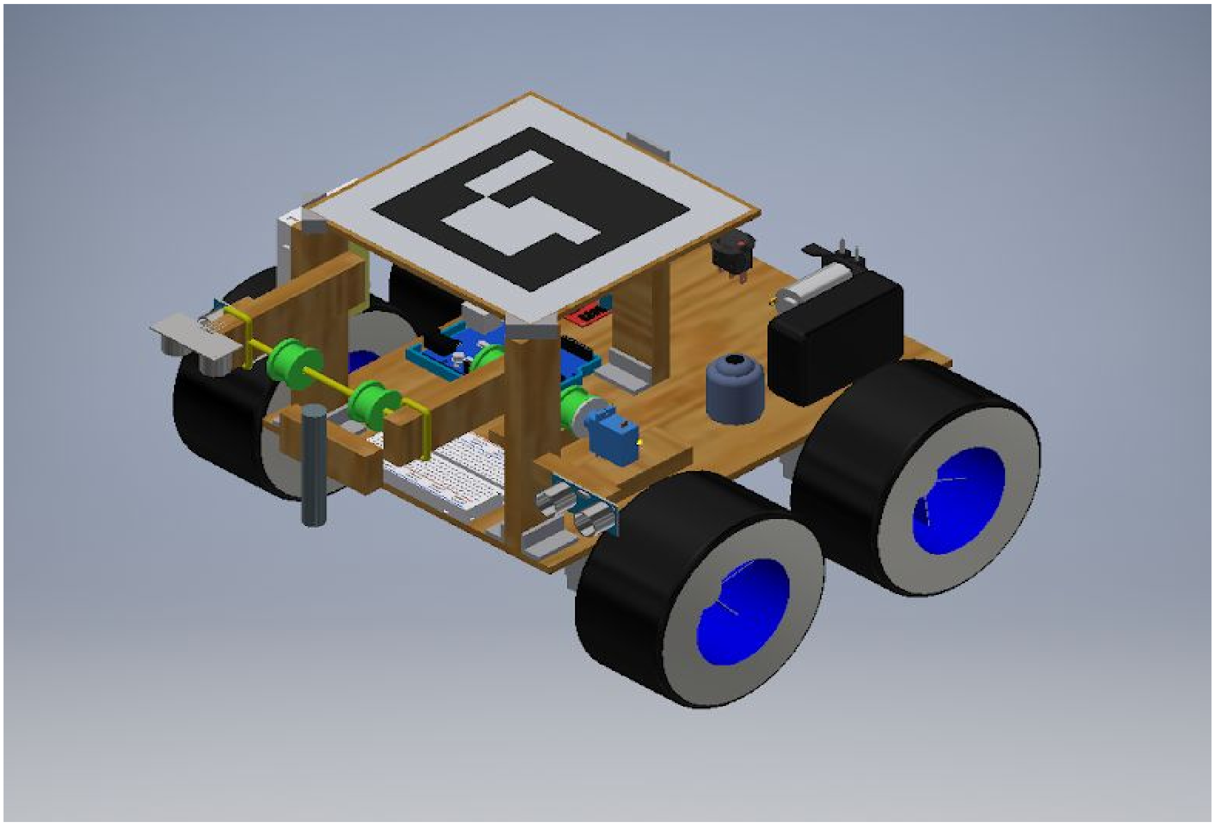

Image 1: 3D CAD Drawing of our OSV

Preliminary Design Shortcomings

As constructed our OSV there were many alterations from our original design since certain components were replaced with more effective ones to do the job, and certain conflicts caused other parts to be left out of the final OSV. Our idea to incorporate the forklift to drop our sensors into the pool of water was changed to a pulley system with two servo motors. We decided to use the pulley system instead of the forklift because it kept the sensors above the OSV and we would not have to worry about it causing any issues over the rocky terrain. Also, the forklift was over complicated with the 3D printing; whereas, the pulley was much simpler and more effective. Another shortcoming from our preliminary design was using the conductivity sensor to replace the pH sensor. We used the conductivity sensor because it was cheap to make, and it was an easier way to figure out if the water was polluted, fresh or salt. Furthermore, we replaced the water level sensor with an ultrasonic sensor based algorithm for water depth measuring, since the returned value of algorithm was more precise and more consistent. Lastly, towards the end of the construction of our OSV we had an issue with our motor shield, and had to replace the arduino, kill switch, and motor shield which caused us to run out of time to integrate the water pump. We were more worried about getting our OSV to drive to the destination and drop the pulley system into the water rather than having a water pump on an OSV that does not work.

Design Details

<u>Structure</u>

When the structure of the over sand vehicle was designed (OSV), our team kept in mind the type of environment we would be dealing with. Since the OSV had to traverse over rocky terrain, we favored a chassis with a high clearance, so it was not caught on rocks or debris. Additionally, the chassis was rectangular with the dimensions 200mm x 280mm made of plywood since it was relatively lightweight, yet sturdy enough to provide a decent structure for the OSV. We attached our four motor housing parts below the chassis so that our wheels were directly across from one another, and had enough space between each other on the same side. Each motor house was gorilla glued to the bottom of the chassis board with the use of clamps. The motors were four Uxcell DC 12V 50RPM Gear Motor (Image 2). These motors provided ample torque (discussed in propulsion section) to drive the OSV through/over any obstacle in its path.

Image 2: Driving Motor

The OSV has four motors, each attached to a wheel. Hence, we had a four wheel drive system. Having a four wheel drive system enhanced the OSV by allowing it to overcome obstacles with ease. When traversing over rocky terrain, a two wheel drive system may have had one or both of its wheels off of the ground temporarily. This could have caused issues related to navigation and even simply crossing the rocky region. Therefore, by going with a four wheel drive system we greatly increased our chances of crossing the rocky terrain and made it to the mission site. The type of wheels we used were Dagu Wild Thumper Wheel (Image 3). They had a diameter of 120 mm (plenty high to clear obstacles) and a width of 60 mm. The material our wheels were made of was a soft rubber, which was spiked for additional traction, and a plastic rim. All together, the wheels and motors weighed almost 1.6 kilograms, which was already at half of the maximum weight. Therefore, we had to be conservative from this point onwards in our design.

Image 3: 120x60mm Wheel

On top of the chassis, we mounted three wooden columns (two on the front and one on the back of the OSV), with 3D printed pieces on top, in order to hold the marker used to track the OSV. Since the wooden columns we attached to the chassis stood upright, we 3D printed “L” shaped pieces to screw and brace the columns. When deciding how to hang the pulley, we decided to attach two horizontal wooden pieces to the already built mounted columns facing the front of the OSV. On the two extended wooden pieces, we strung two pipe cleaners with two 3D printed spindles. This allowed the pulley to hang in front of the OSV and over the water pool. To power the pulley system we created, two servos were used which each controlled one string. The two strings led to the same mass hanging on the pulley; however, by doing this we divided the load between our two servos. This worked well since the two servos also provided more stability to the pulley system. The conductivity sensor hung off the front attached to a wooden plank that the string was tied to. Our tubing for the pump was also supposed to be attached, but our motor shield burnt out a week before deadline, so we couldn’t integrate the pump to our system. The overall design and build of the pulley system was sufficient in order to complete the mission; however, we had difficulties with the programming aspect of the servos.

This table on the next page (Table 1) has listed out all building components of our OSV and their mass, and it has calculated the total estimated mass.

Mass Estimation			
Item	**Mass (kg)**	**Item**	**Mass (kg)**
Wheel (4)	0.78	Chassis Board (1)	0.28
DC Motor(4)	0.8	Arduino Holder (1)	0.01
Wheel Adapter (4)	0.052	Ultrasonic Sensor (3)	0.042
Motor House (4)	0.092	Home Made EC Sensor (1)	0.006
Battery Pack (1)	0.28	Continuous Servo (2)	0.018
Switch (1)	0.0045	Pulley (4)	0.024
Solderless BreadBoard (2)	0.0778	Pulley RFholder (2)	0.09
Tamiya Connectors (1)	0.045	Ultrasonic holder (2)	0.006
Arduino UNO R3 (1)	0.025	RFholder (1)	0.026
H-Bridge Module (1)	0.033	Servo Holder (2)	0.02
Other	0.02	**Total**	2.7313

Table 1: Mass Estimation of the OSV

Following engineering drawing (Image 4) demonstrates the overall width, depth, and height of our OSV.

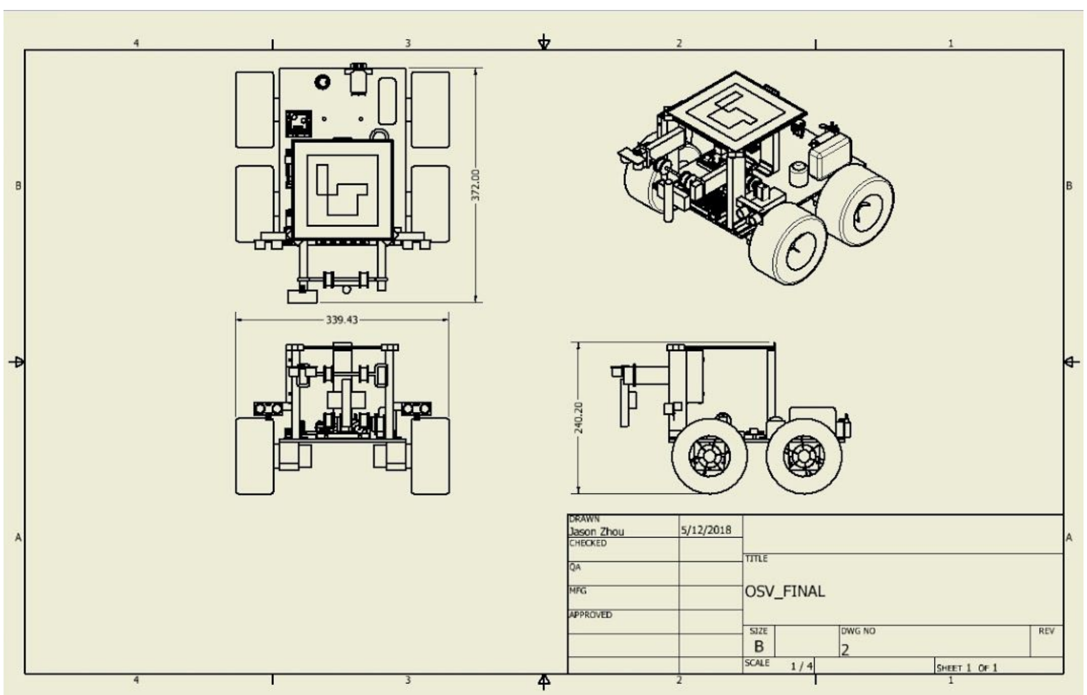

Image 4: Technical Engineering Drawing #2, including side, top, and front views

Following engineering drawing (Image 5) is the exploded view of our OSV. It shows how our OSV was assembled, and which pieces are connected. It also demonstrates the position and orientation of each component on our OSV.

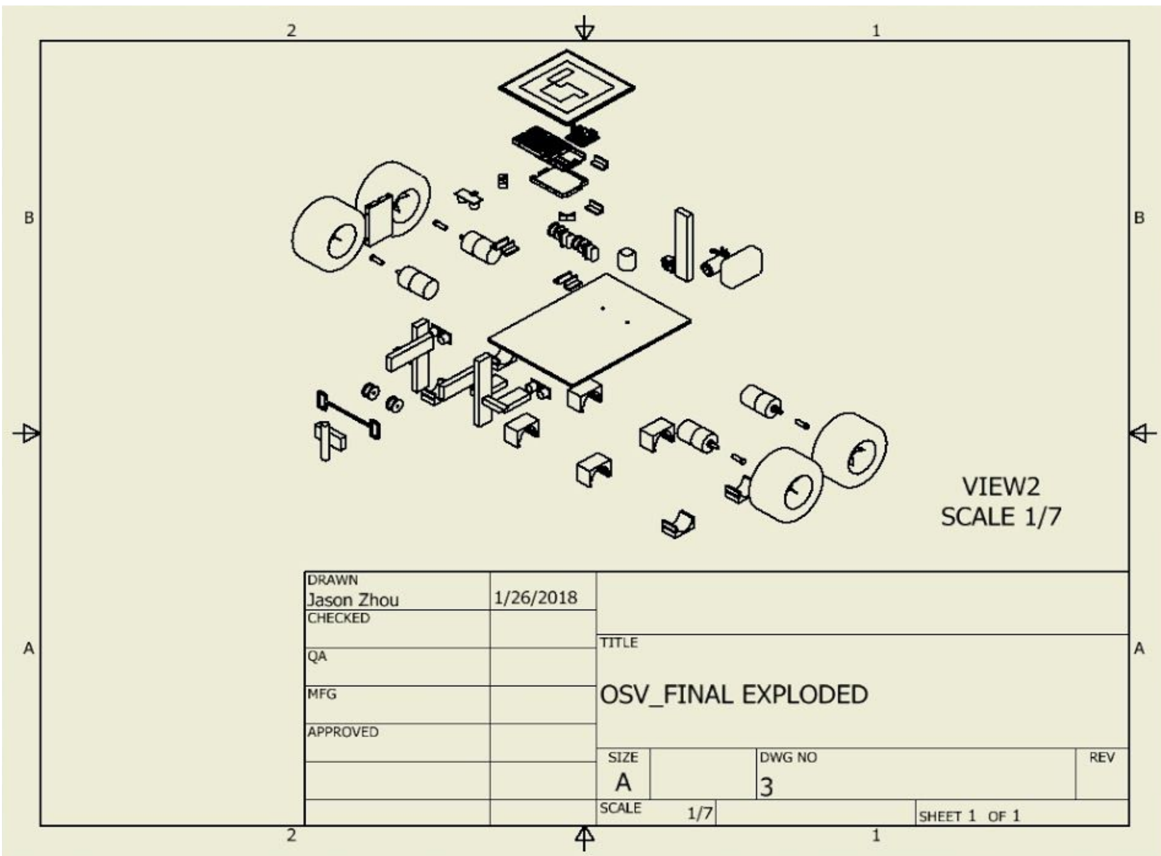

Image 5: Exploded view of OSV

Propulsion

We used four motors and four wheels on our OSV because we felt this would optimize the turning radius of the OSV. The OSV turned by connecting the motors on each side in parallel with a L298n bridge which connected the wheels on each side to go forwards or backwards at the same time. We used the arduino code so that each side spin in opposite directions to turn. For example, to turn to the right the wheels on the right side of the OSV reversed and the wheels on the left side of the OSV went forward. Then when the OSV was pointed in the correct direction all four wheels began to drive forward again. We chose large wheels on our OSV which helped to get over the rocky terrain, the wheels we chose had a diameter of 120 millimeters and a width of 60 millimeters. The motors that were chosen had a stall torque of 0.39 newton meter and a no load speed of 50 rotations per minute. The calculations below found the forces of rolling resistance and traction, torque of each wheel, angular speed of each wheel, and the linear speed of the OSV.

$$F_{N\ each\ wheel} = \frac{F_g}{4} = \frac{2.815kg(9.81\ ^{m}/_{s^2})}{4} = 6.91\ N$$

$$C_{RR} = \ [3.33\frac{cm^3}{N} * \frac{F_N}{wd^2}]^{1/3} = [3.33\frac{cm^3}{N} * \frac{6.91\ N}{6cm * 144cm^2}]^{1/3} = .299$$

$$F_{RR\ each\ wheel} = F_N C_{RR} = 6.91N(.299) = 2.07\ N$$

$$F_{T\ each\ wheel} = F_{RR\ each\ wheel} = 2.07\ N$$

$$\tau_{each\ wheel} = F_{T\ each\ wheel} * R = (2.07N)(.06m) = \ .124\ Nm$$

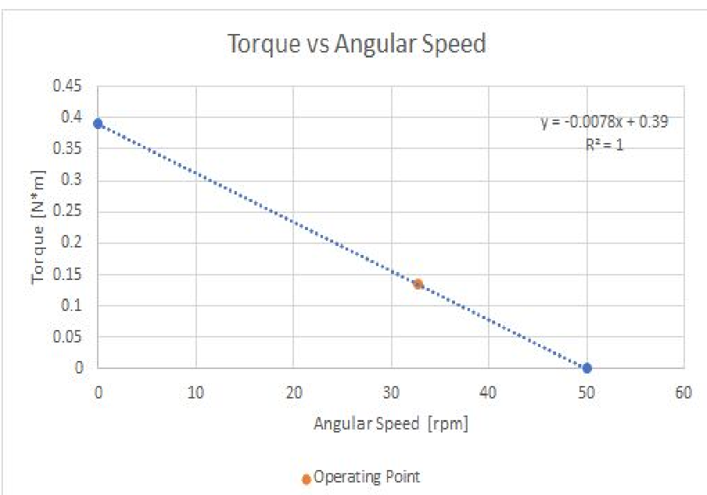

Image 6: Graph of Angular Speed vs Torque, including OSV’s operating point

$\tau = -.0078\omega + .39\ Nm$

$.124\ Nm = -.0078\omega_{operating} + .39\ Nm$

$\omega_{operating} = 34.1\ rpm$ (as shown in Table 1)

Angular Speed Conversion (rpm to rad/s):

$$34.1\ rpm = 34.1\ rpm\left(\frac{2\pi}{revolution}\right)\left(\frac{1\ min}{60\ sec}\right) = 3.57\frac{rad}{s}$$

Linear speed Calculation:

$$v = \omega r = 3.57\frac{rad}{s}(.06m) = .214\frac{m}{s}$$

We find that the operating torque is about [(0.124/0.39)*100= 31.8%] 31.8 percent of the stall torque, and an efficient operating torque should be around twenty to fifty percent of the stall torque which we are within. These motors draw a decent amount of torque but it is not too great to be an issue. Our operating angular speed is [(34.1)/(50)*100=68.2%] 68.2 percent of the no load angular speed which is acceptable. The linear speed at 0.214 meters per second is a little fast, but we were able to manipulate it through the arduino code, and it was not an issue in our mission.

OSV Mission

Our OSV had attempted to complete two major objectives and one advanced objective. The first major objective of the OSV was to autonomously navigate to within 250 mm of the edge of water pool, known as the mission site. The second mission was that the OSV must be able to measure and transmit the correct water type back. The water sample in the pool could be one of the following types: freshwater, saltwater, or polluted water. For the advanced objective, the OSV had to measure and transmit back the depth of the water in pool within 4 mm.

To accomplish the navigation mission, we had to know the mission site's and our OSV's current coordinates at any immediates. Therefore, we used an APC220 Radio Communication Module to receive the coordinates from an existing vision system that overlooks the arena. Then, we used an L298N H-bridge module to control the locomotions of our OSV and two HC-SR04 ultrasonic modules to detect obstacles in front of it. Finally, our programming subteam had integrated collected data and implemented a navigation algorithm that would let our OSV autonomously driving to the mission site without getting into any obstacles.

After it arrived at the mission site, our OSV was designed to lower an electrical conductivity sensor using its pulley system. The pulley system was powered by two FS90R continuous servo motors, which each servo controls one string that is attached to the hanging object. Once the electrical conductivity sensor was lowered into the pool, the servos stopped and the sensor started to get readings from the pool. Water type was determined by the returned values (We had separately completed electrical conductivity sensor tests for different water samples before it was attached onto the OSV). Image 7 demonstrates how our pulley system was built.

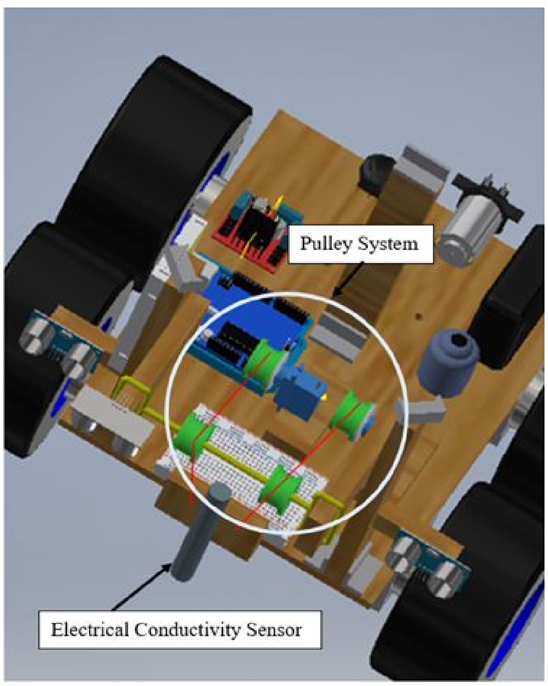

Image 7: Water Sample Mission

At the same time the OSV started lowering the electrical conductivity sensor, the third ultrasonic sensor that was fixed on the top arm started to get readings. As a part of our water depth calculation algorithm, the returned value, which is the distance from the ultrasonic sensor to the surface of water, was subtracted from the distance from the ultrasonic sensor to ground, an preset value, to get the water depth. Image 8 demonstrates the arrived OSV and our water depth arithmetic.

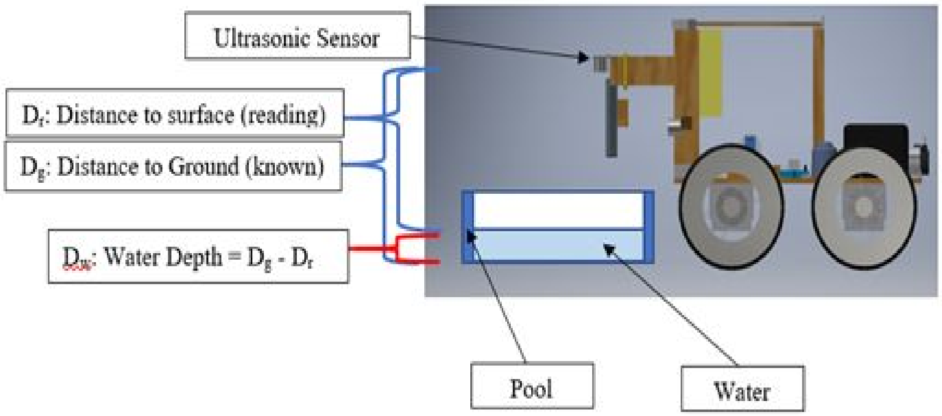

Image 8: Water Depth Algorithm

We originally planned to accomplish the advanced mission of collecting water sample. However, because we ran into a series of technical problems one week before the deadline, we decided to ignore it.

<u>Power</u>

In order to control the OSV, we had to power both propulsion and electronic control systems. The OSV propulsion system is controlled by a L298N H-bridge module and an Arduino microcontroller. Rest of the electrical components are controlled and powered by the Arduino microcontroller.

The Arduino and H-bridge were simultaneously powered by the 12V battery, as is shown in Image 9. The battery powers the H-bridge and Arduino via power terminal and Vin pin. Driving motors are powered by the H-bridge. The rest of the electrical components are supplied by the 5V output pin of the Arduino. A 5V

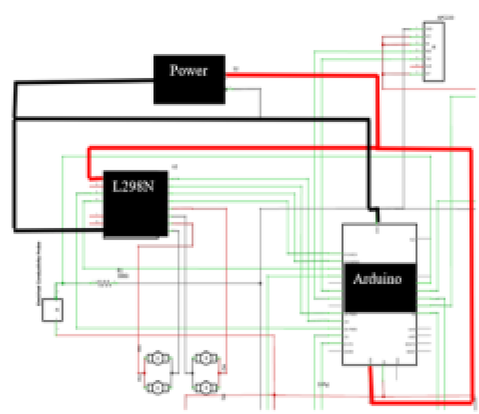

Image 9: Power Modulation

voltage regulator of the Arduino has modulated the 12V input power to a 5V output power.

Considering our OSV situation and lab hours, we decided to use a 12V 2000mAh NiMH battery. It has a high charge capacity and is environmentally friendly.

To estimate the run time of the OSV, we first figured out the operating current of each DC motor. We used this current vs. angular velocity graph to determine the operating current.

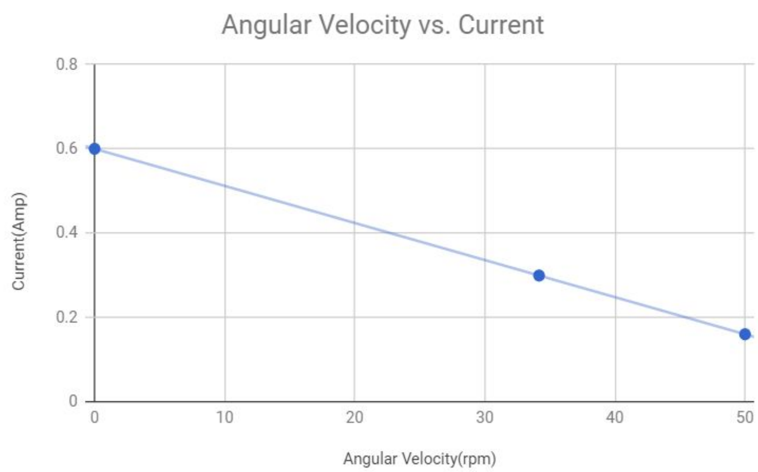

Image 10: Angular Velocity vs. Current

Our OSV's operating point is at around 34.1 rpm. By substituting that value into the line of best fit equation, we got that the operating current of each motor was .3 A, as is shown in Image 10.

Other electrical components, including two servos and three ultrasonic sensors, only draw 6 - 200 mA each to operate, depending what module it is . We assumed that they used a total of 500 mA while the OSV was operating.

To find the estimated run-time of the battery, we divided the battery capacity by the total current draw while operating. Which comes to the following equation:

$$T(hr) = \frac{Q(mAh)}{I(mA)} = \frac{2000mAh}{4 * 300mA + 500mAh} = 1.18hr$$

The estimated run-time of our OSV is about one hour and 11 minutes.

During the actual test, every electronic component except the motors performed as we expected. The motors were running slower than we thought. Later, we did research on why that phenomenon happened. It happened because the H-bridge module had made a voltage drop of about 2V. Therefore, even though we used a 12V battery to power the module, the input voltage for the motor would be only about 10V, which means we could not run the 12V DC motor at its maximum speed.

Circuit

We designed our circuit to put our Arduino Uno and H-bridge module in parallel, so they could be powered by the 12V battery at the same time. We originally planned on stacking a motor shield directly on top of the Arduino. However, we ran into trouble when our original electrical components short circuited and we had to replace almost all of the circuitry. Due to this, our final design includes a L298N H-bridge module instead, which is controlled by six digital pins, including two PWM pins on the Arduino, and outputs to the four motors we use. Pins 13, 12, 11 (PWM), 5, 4, and 3 were all used to control the L298 H-bridge. Each motor port controls two motors that are on the same side of the OSV and are wired in parallel. We also use three ultrasonic sensors for navigational purposes and to measure the water level, and these sensors are controlled by digital pins on the arduino. Digital pins 1, 2, 14, 15, 16, and 17 were used to control the ultrasonic sensor. We use a 300 ohm resistor in series with the conductivity sensor to reduce the possibility if the sensor breaking due to an overload of current. Reading returned from the electrical conductivity sensor was recorded by analog pin A4. All of the modules we used except the four motors and the L298N H-bridge are connected in parallel and are supplied by the 5V output pin of the Arduino UNO. In addition, we used a kill switch in the circuit as an extra precaution. The whole schematic(Image 11) is shown below.

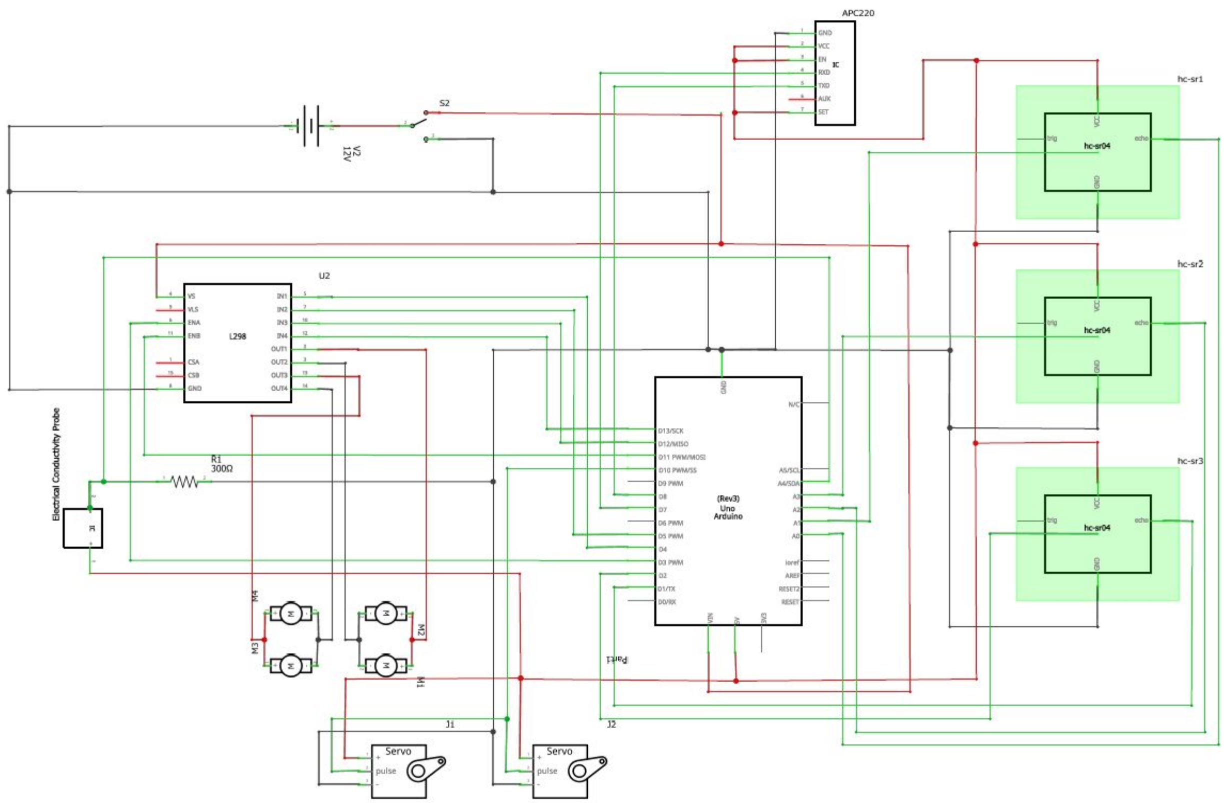

Image 11: Full Circuit Schematic of OSV Components, Arduino and H-bridges

Sensors and Actuators

Two ultrasonic sensors(Image 12) were placed on the front corners of the OSV to detect obstacles in the front and slightly to the sides of the OSV. The ultrasonic sensors were integrated onto the OSV and used through a method that sent out a ping and measured the time it takes to return. This time was converted to the OSV's distance from the detected object. If the distance that came back was within a predetermined range then differential steering will be activated to avoid the obstacles.A third ultrasonic sensor was placed off on the top arm of the OSV facing downwards. This sensor used a similar method to measure the distance from the sensor to water in the pool. This value will then be subtracted from the distance to the bottom of the pool to return in mm the water level. In the preliminary

Image 12: Ultrasonic Sensor

design a water level sensor was going to be used to detect the water depth, but it was found that ultrasonic sensors provided a more consistent measurement. The water level sensor was not calibrated to the precision needed for the advanced requirements and was not feasible.

The APC220 Radio Communication Module (Image 13) has transmitted and received signals between the OSV and mission control. The antenna has a wide range; therefore, it did not matter where specifically the transceiver is placed. Ultimately it was placed on the second bread board mounted on the post of the OSV near the front. The transceiver modules were already implemented, and we simply had to use pre-written messages in order to communicate with the command module.

Image 13: APC220 Communication Module

The H-bridge (Image 14) is an electronic circuit that connects all four motors at a central point that then connects to the Arduino. The H-bridge was placed centrally off to one side of the OSV so that the all four motors could reach it as well as the Arduino. The H-bridge needed be connected to the Arduino via six input pins. Motors on the same side of the OSV will be wired in parallel; therefore, they will move forward and backward together when the correct switches are closed, which allowed differential steering. A motor shield was our prefered method for differential steering, but before the competition our motor shield burnt out and a new one could not be ordered in time.

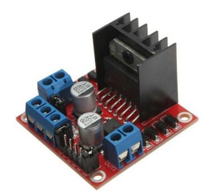

Image 14: L298N H-Bridge

The electrical conductivity sensor (Image 15) is made using nichrome wire soldered to insulated copper wire. The exposed nichrome wire has been attached to a plastic pen body with electrical tape, leaving a small part of both wires exposed to complete the circuit. The copper wires will then be attached to the breadboard and ultimately the value will be transmitted through an Arduino analog pin. If the output value was above 200 then the method returned a statement to the command center stating the water quality was salty, if the output was

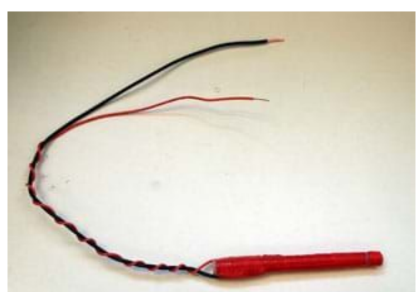

Image 15: Electrical Conductivity Sensor

between 35 and 200 it returned polluted water, and if the output was lower than 35 it returned fresh water. Initially a pH sensor was going to be used to determine if the water was tested was polluted because it was assumed that the acid would not cause a significant change in conductivity. After initial testing it was found that the conductivity sensor could in fact detect the addition of H+ ions rendering the pH sensor unnecessary.

A water pump(Image 16) was attached to plastic tubing and placed on the back corner of the OSV. One end of the tubing was attached to the pulley so it could be lowered into the water. The other end of the tubing goes into the plastic bottle placed on the back corner of the OSV. The water pump was attached to the Arduino by an input pin so that the water pump could be turned on and off for a set amount of time. The time that the pump is on was determined by the flow rate, 2 ml/sec, to get the desired volume.

Image 16: Water Pump

Two FS90R servo motors (Image 17) were attached to the chassis in the center with screws. Attached to these motors were 3D printed spindles to hold the string. The purpose of the servo motors were to effectively raise the pulley system containing the conductivity sensor in and out of the water. The two servos ran in parallel at the same speed so as to evenly lower the pulley. Both servos were connected to the arduino by three digital pins. The motors were controlled by code that would turn them on for approximately one second once the OSV was over the water pool.

Image 17: Servo

Control Algorithm

Our control algorithm consisted of two different parts: the first navigated the OSV to the destination (Image 18) and the second (Image 19) which activated our pulley system which would allow us to complete our mission requirements. The first part of the algorithm connects with the visual system to receive the coordinates for the destination and the OSV’s current location and direction, and then has the OSV begin to move towards the destination. As it moves it is constantly updating its current location and checking for obstacles in its path. If the OSV sees any obstacles it tries to maneuver around the obstacle. After the OSV moves around an obstacle it will reorient itself towards the objective and attempt to continue navigating towards this objective. This causes the OSV to move towards the obstacle while avoiding running into any obstacle until the OSV arrives at the objective. Once the OSV has arrived at the destination it will activate the servo motor to lower our pulley system, and then reads the pin inputs, which will detect the values we need to complete our missions from the sensors.

We changed the design of our control algorithm in our final implementation by changing how the OSV navigated around obstacles so that it would not bump into these obstacles unintentionally and changed how the OSV approached its destination to avoid bumping into the water pool at the destination. Instead of having the OSV just turn away from the obstacle and moving away, we had the OSV turn away from the obstacle, move forward, and then turn back towards the original obstacle to see if the obstacle is still too close to the OSV. If it is, then the OSV will turn even further away and move further away from the obstacle to assure that the OSV does not unintentionally bump into these obstacles. To avoid bumping into the water tank present at our destination we had the OSV move more slowly as it moved closer to the water pool to ensure that the we arrived within 250mm but also not overshoot and move into the water pool. Flow charts on the next two pages demonstrate both our control algorithms, navigation and mission.

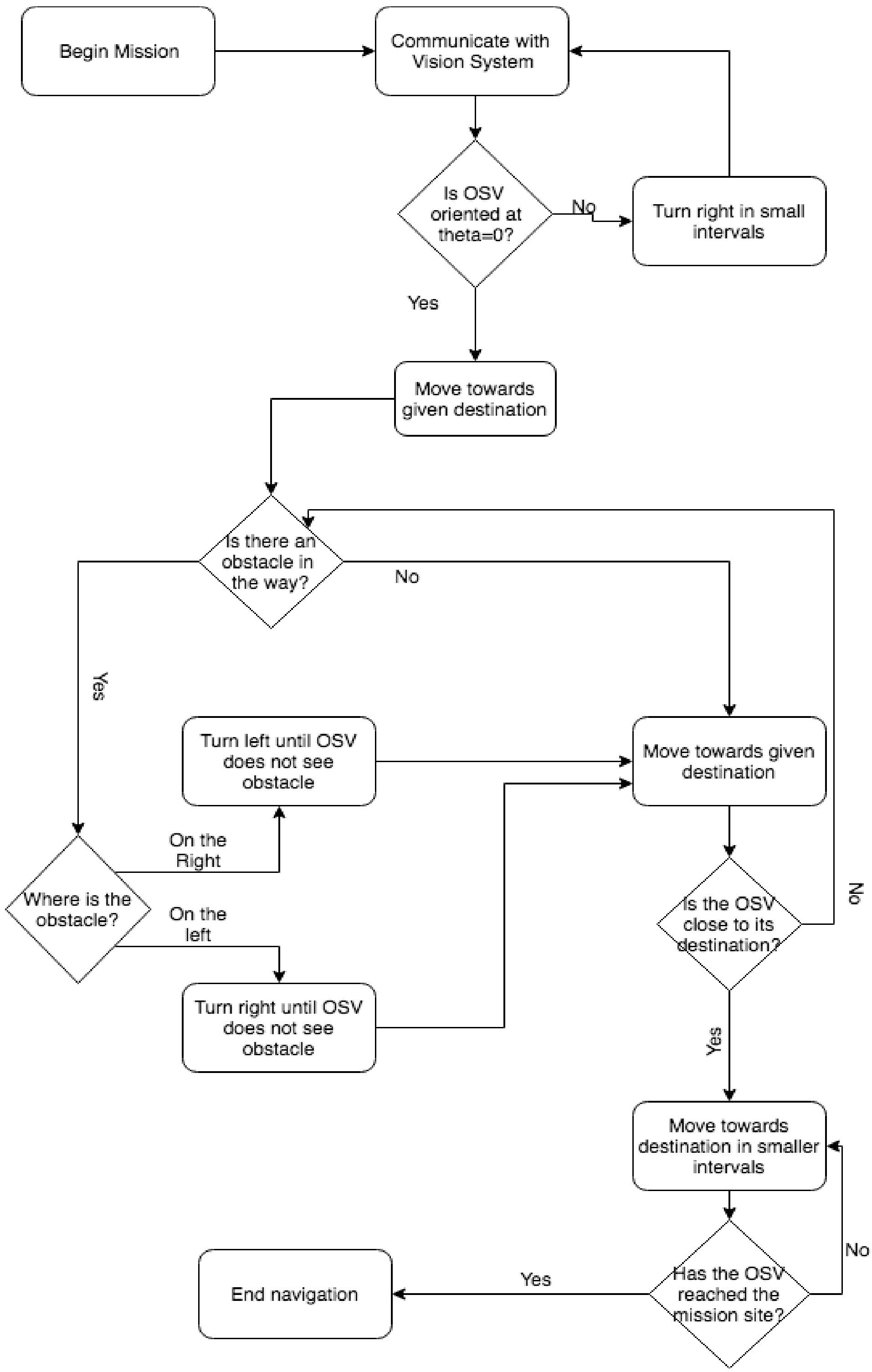

Image 18: Navigation Algorithm

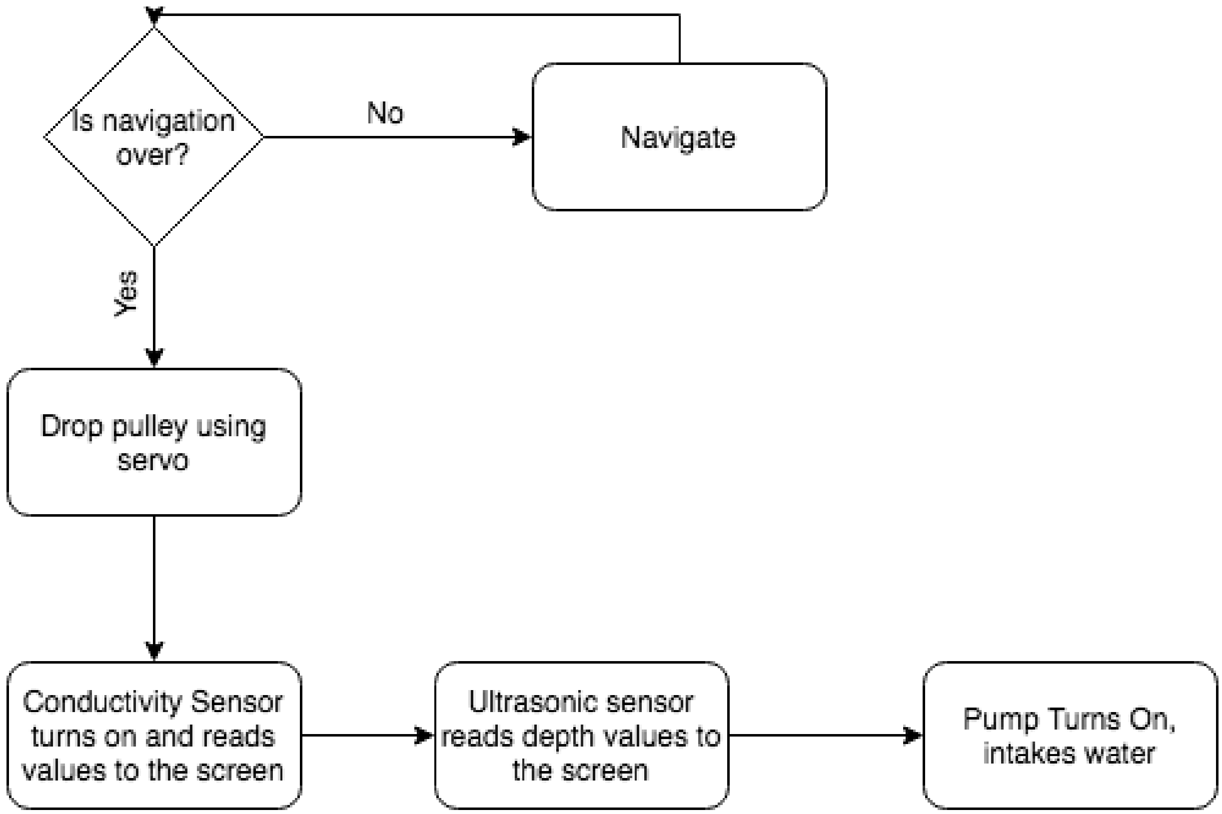

Image 19: Control Algorithm for post-navigation maneuvers using the servo and conductivity sensor

Final Design Drawings

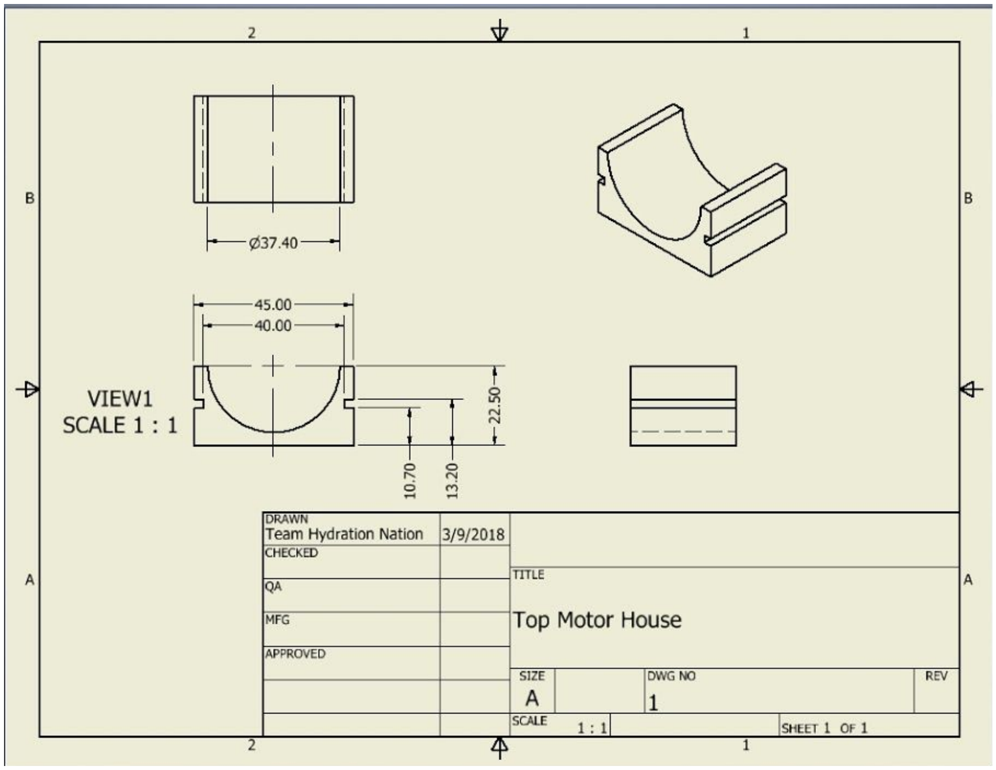

Image 20: Top Motor House

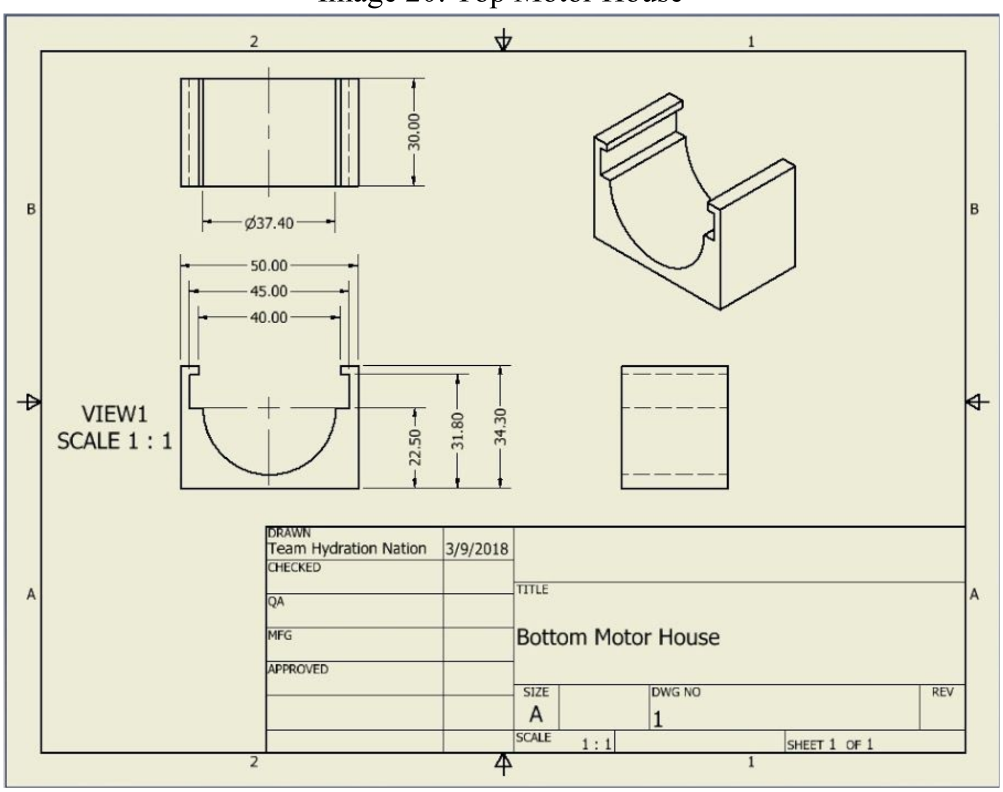

Image 21: Bottom Motor House

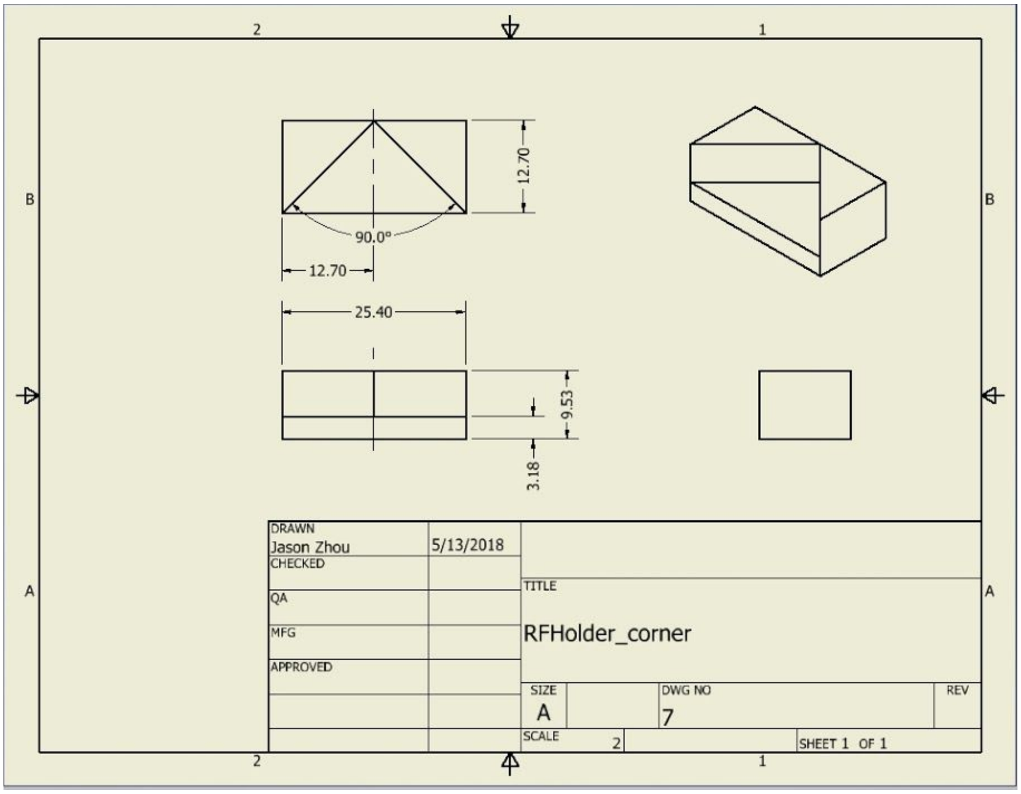

Image 22: RFHolder_corner

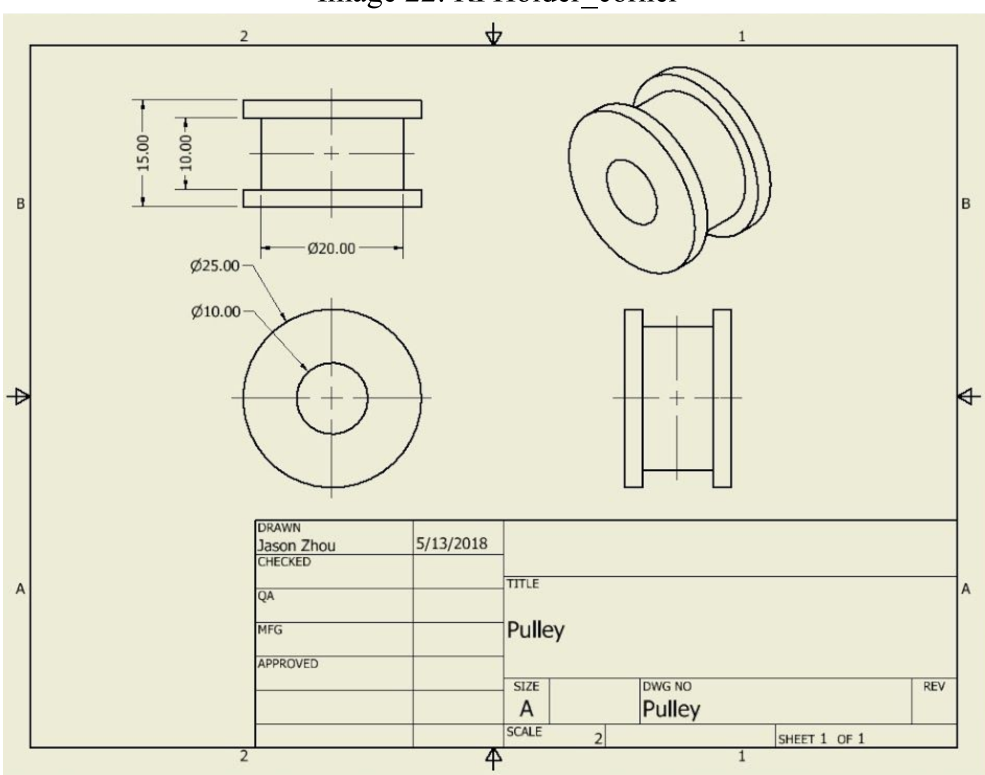

Image 23: Pulley

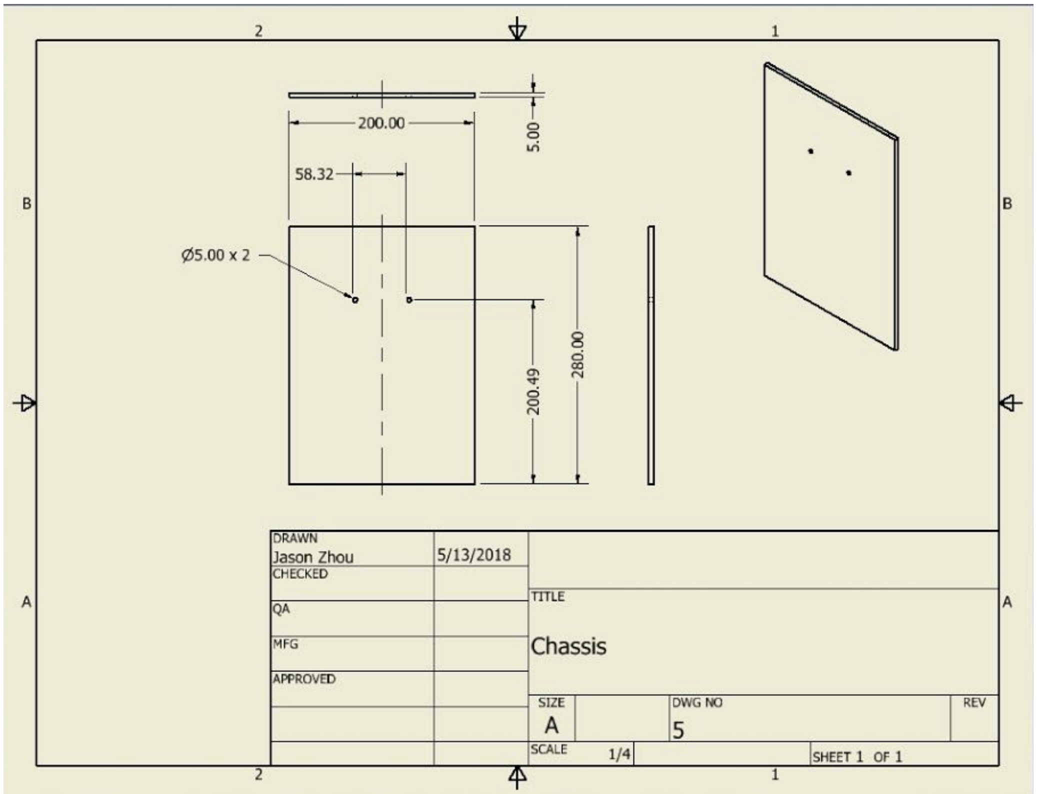

Image 24: Chassis

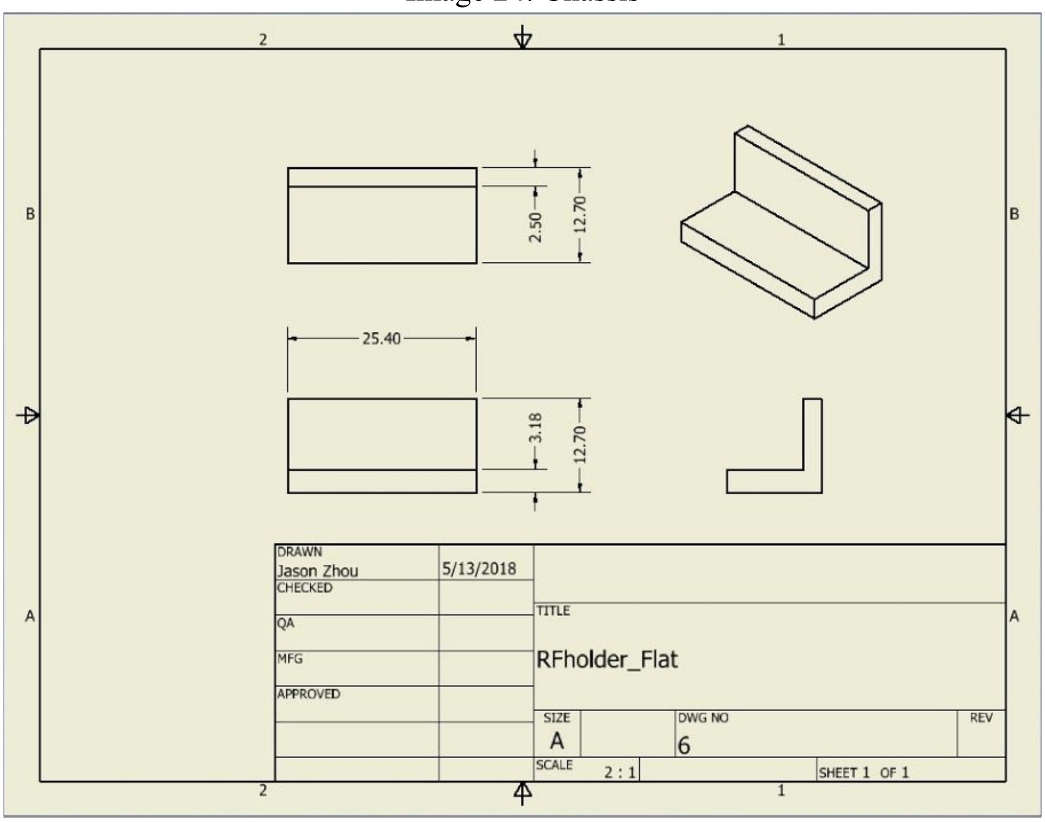

Image 25: RFholder_Flat

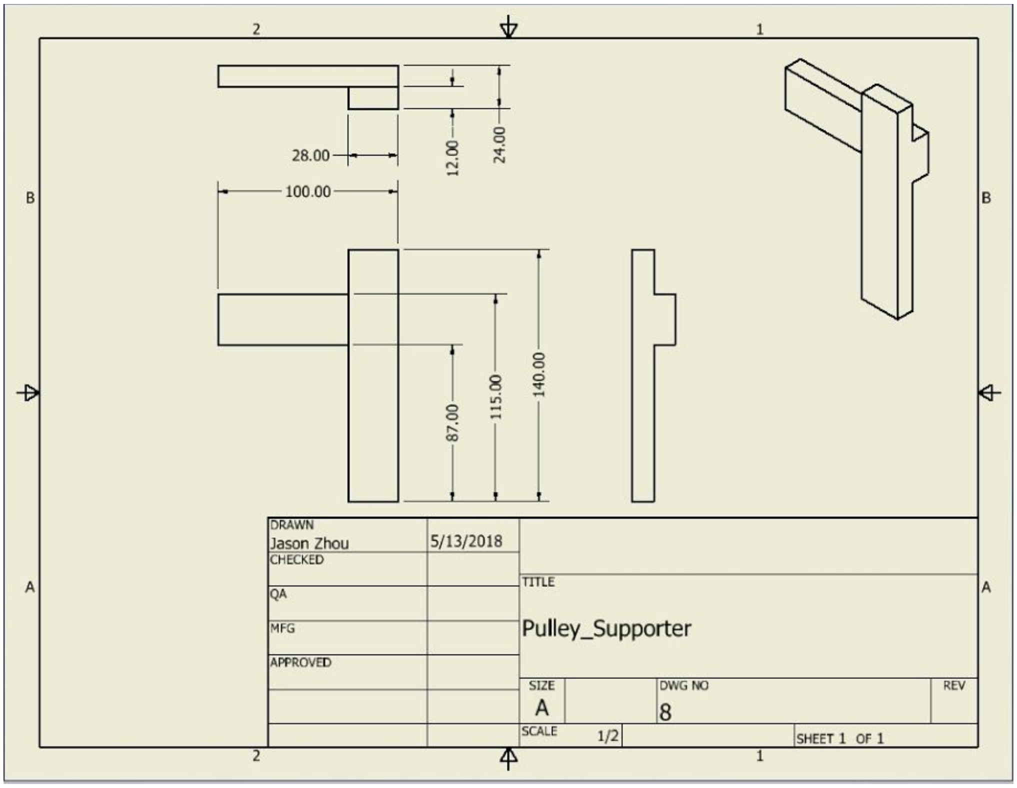

Image 26: Pulley_Supporter

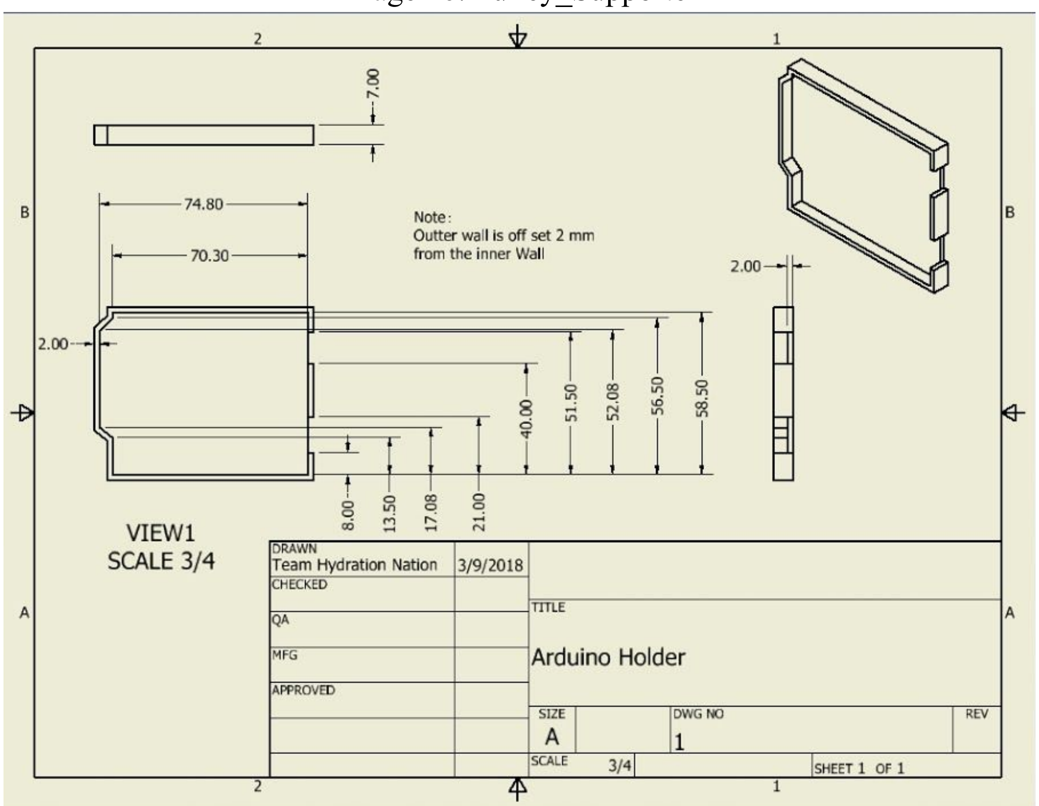

Image 27: Arduino Holder

Construction Details

We began by buying plywood for our chassis. This plywood was then cut to the 200mm x 280mm dimensions we had designed for our OSV. Once the chassis had arrived, we 3D-printed motor mounts and an Arduino holder. We had to used AutoDesk Inventor 2017 to CAD these motor mounts and Arduino holder, and once we had done so, 3D-printing them took approximately 2 hours on the 3D-printers provided. We then attached the motors to the motor mounts using epoxy, and the motor mounts to the chassis on all four sides, using wood glue. The wheels, pre-ordered, were then attached to the motors.

After the basic chassis and wheels were finished, we attached the Arduino holder to the top of the chassis, along with three wood beams, which would act as both a holder for the tracker and the front two would act as support beams for our pulley. The pulley consisted of a small rectangular wooden piece, to which the tubing for the pump and the conductivity sensor were attached with zip ties. The conductivity sensor was constructed by attaching two wires two a hollowed-out pen Two separate strings held the pulley in place, each one running up to a servo. These strings were supported by a pipe cleaner that ran across the two support beams and allowed the pulley to move up and down. This pulley system was designed so that the servos would rotate and lower the pulley into the water so that the sensor could collect data and the pump could collect a sample of the water.

Once this was completed, we added a breadboard next to the Arduino holder and one on the right support beam to ensure wiring which was not too messy. We then attached the designed pulley to the two front support beams, and also attached the two ultrasonic sensors to the support beams, allowing for the OSV to “see” obstacles a little higher. An additional ultrasonic sensor was attached to the top of one of the support beams, facing downward, designed to measure water depth. A peristaltic pump was mounted on the rear of the OSV, and the tubing was routed from the pulley to the pump. This system would pump our water sample into a closed container constructed from a modified water bottle.

Image 28 on the next page shows a rear of our built OSV.

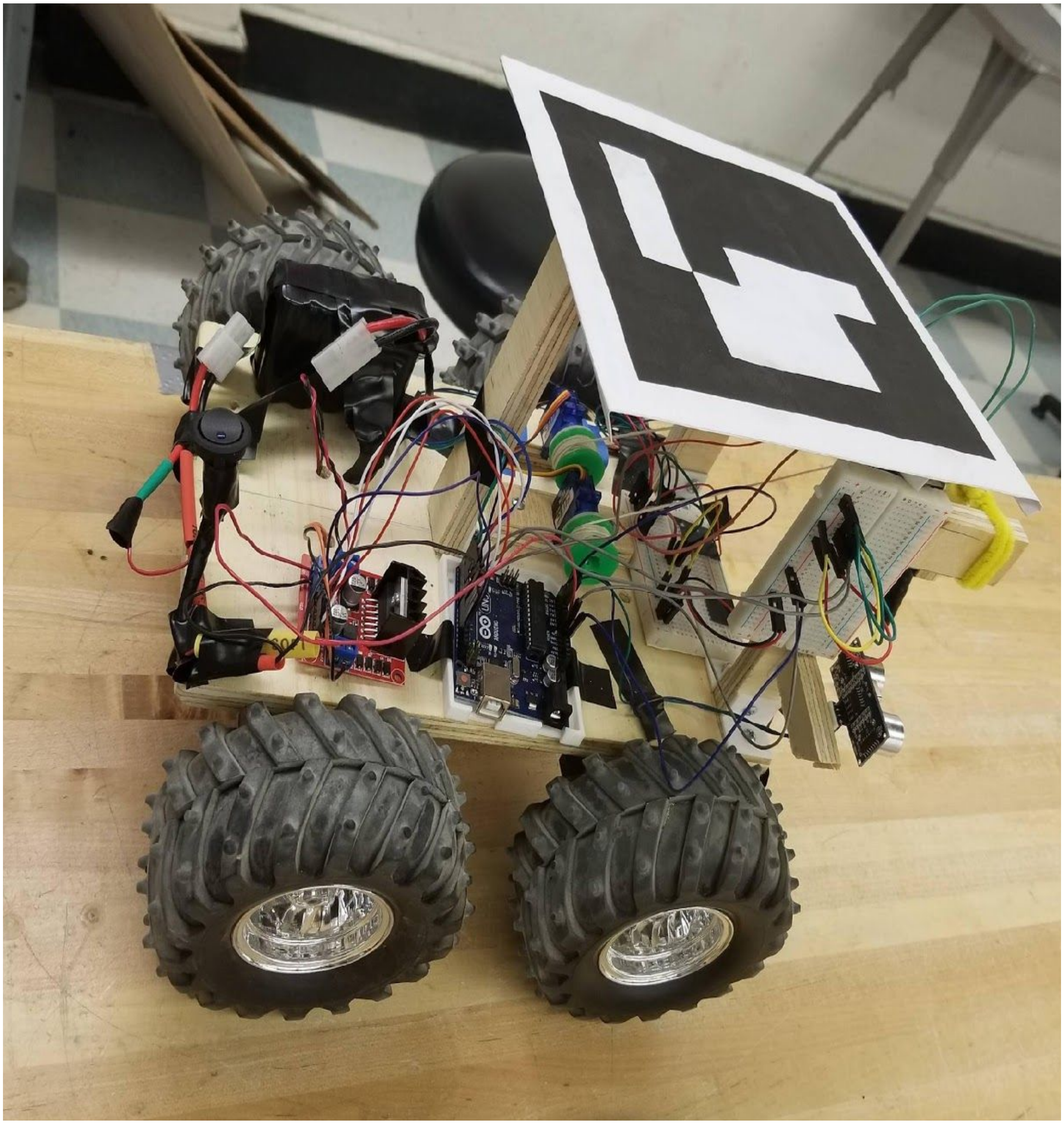

Image 28: Rear view of OSV

Bill of Materials

The bill of materials is a list of raw materials, sub-assemblies, sub-components, parts and quantities of each needed to manufacture our final OSV. In our bill of materials, mass and cost of each components are listed, as well as manufacturer, vendor, and model number.

Bill of Materials					
Item	**Mass(kg)**	**Cost($)**	**Manufacturer**	**Vendor**	**Model**
Propulsion					
Wheel (4)	0.780	42.35	Dagu Electronics	Pololu	N/A
DC Motor(4)	0.800	59.96	Uxcell	Uxcell	a17092200ux0456
Wheel Adapter (4)	0.052	23.28	Uxcell	Uxcell	a14012200ux0032
Motor House (4)	0.092	2.30	UMD Keystone	UMD Keystone	N/A
Power					
Battery Pack (1)	0.280	23.92	Tenergy	Tenergy	N/A
Switch (1)	0.005	1.50	HOTSYSTEM	HOTSYSTEM	KN3(D)-101
BreadBoard (2)	0.078	10.00	UMD Keystone Lab	UMD Keystone Lab	HS-CP-19x5-1
Connecters (1)	0.045	7.48	Venom RC	Venom RC	N/A
Control					
Arduino UNO R3 (1)	0.025	22.00	Arduino	Arduino	A000066
H-Bridge (1)	0.033	6.89	Qunqi	Qunqi	L298N
Chassis Board (1)	0.280	7.21	UMD Keystone Lab	UMD Keystone Lab	N/A
Arduino Holder (1)	0.010	0.25	UMD Keystone Lab	UMD Keystone Lab	N/A
Sensor / Actuator					
Ultrasonic Sensor (3)	0.042	5.87	ElecRight	ElecRight	HC-SR04
EC Sensor (1)	0.006	3.00	UMD Keystone Lab	UMD Keystone Lab	N/A
Servo Motor (2)	0.018	14.99	Feetech	RCMall	FS90R

Pulley (4)	0.024	0.60	UMD Keystone Lab	UMD Keystone Lab	N/A
Pulley RFholder (2)	0.090	0.00	UMD Keystone Lab	UMD Keystone Lab	N/A
Ultrasonic Holder (2)	0.006	0.00	UMD Keystone Lab	UMD Keystone Lab	N/A
RFholder (1)	0.026	0.00	UMD Keystone Lab	UMD Keystone Lab	N/A
Servo Holder (2)	0.020	0.00	UMD Keystone Lab	UMD Keystone Lab	N/A
Screws & Small 3D	0.020	7.99	UMD Keystone Lab	UMD Keystone Lab	N/A
Total	2.731	239.59			

Table 2: Bill of Materials

*Cost for using tools in the Keystone lab is not counted toward the final cost.

Product Performance and Evaluation

During our first OSV run, our OSV failed to get any points. This was because for an unknown reason our front right and back left wheels were stuck, and did not move, inhibiting OSV navigation. After the failure of our first run, we made no changes to our OSV except for removing the servo code from our code we uploaded to the OSV. During our second OSV run, our OSV navigated perfectly through rocks and came right up to the water pool before stopping. We acquired twenty points by successfully completing navigation. The servo code could have been interfering with our navigation code even though they were not in the same loop, which is why the first run did not work.

If we had had more time, we would have been able to correctly attach and utilize the servo, as it worked by itself, but when attached to the OSV, it did not work as planned. In addition, another issue we could have fixed given more time was to have a slightly bigger chassis. While this was not a big issue, it was a little difficult to attach everything onto the small chassis. We had to attach a breadboard to the side of one of our support beams for the pulley due to lack of space, and if we had more space, this could have been avoided, as it was difficult to keep the wires away from the pulley.

Moreover, we should have redesigned our motor houses at the beginning. We currently glued our motor houses to the chassis and motors to the motor houses, so they wouldn't shift position in the motor house after the OSV making many movements. However, we realized that was a failed design, because motors are not removable anymore. What if one motor is broken or the solder disconnects? Fortunately, they didn't happen. We should have either designed or purchased mounting bracket holders and screwed them to the chassis, which would still keep the motors in position, but now they become removable and adjustable. In a real incident, this design could be very crucial.

Finally, one other thing we could have done was to place the ultrasonic sensors in front of our wheels. In the current configuration, they were on the side of the wheels and pointed straight. This led to the OSV not being able to see obstacles directly in the center of our path. For the future, we could have attached beams in front of our wheels so the ultrasonic sensors could be place on them and then they would be able to see objects directly in front of them. We could also have attached more ultrasonic sensors to ensure more precision in navigating. Overall, we were happy with our design and product, but with more time we would have worked more on the pulley and attempted to fully finish it.

Lessons Learned

Throughout the entirety of the OSV mission, our team came across a multitude of setbacks, and had to deal with them swiftly and efficiently. Among these difficulties were our battery being stolen, our motor shield breaking, and the mission system being inoperable at times. We quickly ordered a new battery, got an H-bridge so our OSV could at least navigate, and went in during the night to work on our OSV if the mission system did not function properly during class. From these difficulties, some of the biggest lessons learned were to be flexible, understanding, and hardworking. From the beginning, this OSV was not going to be something that was easily built. It required extensive knowledge of circuitry, programming, propulsion techniques, and general building techniques. Our entire team had to be very hardworking from the start and adhere strictly to deadlines, otherwise we would not have completed this project. Due to the fact our motor shield broke two classes before the final testing, we could not properly and safely attach the servo with the conductivity sensor to our OSV, so, next time, we would attempt to do that with a functioning motor shield. In addition, we had to be flexible. When the motor shield broke, we had to quickly think of a solution. The team decided to get an H bridge, as it was quick to find and our only viable option at the time. This turned out well, because our navigation ended up running perfectly, so instead of a zero, we got some points on our runs. Finally, the biggest lesson learned was to be understanding of everyone involved in the project. Whether it be a team member being sick or late, or a TF or Professor Roshwalb not having enough time to see us at the moment, we had to understand that this is a stressful project, issues will arise, but if we were patient and worked hard, all issues would be resolved, and the team would accomplish the task we set out to accomplish.