GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

SENSOR PLATFORM FOR ALGAL GROWTH AND WATER TEMPERATURE MONITORING

Autor: María Luz Castilla Mena

Director: Enrique Gutiérrez-Wing

Madrid

Junio de 2020

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

SENSOR PLATFORM FOR ALGAL GROWTH AND WATER TEMPERATURE MONITORING

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2019-2020 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: María Luz Castilla Mena Fecha: 02/ 06/ 2020

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

ESGWing.

Fdo.: Dr. Enrique Simón Gutiérrez Wing Fecha: 03/JUN/2020.

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

SENSOR PLATFORM FOR ALGAL GROWTH AND WATER TEMPERATURE MONITORING

Autor: María Luz Castilla Mena

Director: Enrique Gutiérrez-Wing

Madrid

Junio de 2020

PLATAFORMA DE SENSORES PARA LA VIGILANCIA DEL CRECIMIENTO DE ALGAS Y LA TEMPERATURA DEL AGUA

Autor: Castilla Mena, María Luz.

Director: Gutiérrez-Wing, Enrique.

Entidad Colaboradora: Boston University.

RESUMEN DEL PROYECTO

<u>Introducción</u>

Planteamiento del Problema

Los depósitos de agua se utilizan para almacenar agua limpia. En estos depósitos de agua potable, las floraciones de algas pueden significar el crecimiento de cianobacterias (Figura 1, [SANC19]), que liberan toxinas nocivas que no solo son perjudiciales para la vida acuática sino también para la salud pública. La detección temprana de estas toxinas (mediante la detección de cianobacterias) es clave para controlar y detener el rápido crecimiento de estas bacterias, evitando así la contaminación del agua. La medición de parámetros, como la temperatura del agua a distintas profundidades, permite conocer la concentración de bacterias en ese depósito en particular. Sin embargo, los actuales procesos de medición son laboriosos, inconvenientes y limitados, ya que los parámetros del agua solo pueden ser medidos manualmente a lo largo del borde del depósito.

Figura (1). *Depósito de agua contaminado por cianobacterias.*

Estado de la Técnica

El cliente del equipo, la Dra. María Teresa Gutiérrez-Wing, espera que el equipo construya un dispositivo que mejore el alcance y la cantidad de recolección de datos para ayudar a detectar el crecimiento temprano de las bacterias antes de que se concentren demasiado para permitir el tratamiento temprano del agua. El objetivo del grupo es diseñar un prototipo de plataforma acuática móvil que se traslade por control remoto a un lugar deseado del depósito de agua, donde despliegue un sensor a una profundidad especificada por el usuario, y que registre y transmite los datos.

Objetivos del Proyecto

El objetivo de este proyecto es mejorar los métodos actuales de medición de los parámetros del agua con el fin de detectar toxinas nocivas en lagos y embalses. Para ello, el cliente mencionó varios requisitos que este dispositivo debe cumplir para funcionar con éxito. El equipo debe considerar todos estos requisitos para llevar a cabo el proyecto con éxito y así poder construir un dispositivo que mejore el alcance de recolección de datos del cliente.

Metodología de trabajo y resultados

El tamaño y el diseño de la plataforma se eligió basándose en cálculos de flotación aproximando el peso de los componentes internos que va a llevar la plataforma. Además, se hizo una rampa en la parte delantera de la plataforma y se añadieron rodillos para facilitar el despliegue del sensor. Se fabricó una pieza acrílica en la parte superior de la plataforma para asegurar adecuadamente los componentes internos a ella. También se diseñó una cubierta para la plataforma para proteger estos componentes del agua.

Tras muchas iteraciones, el mecanismo de despliegue final del sensor ha sido capaz de desplegar con seguridad el sensor hasta una profundidad total de 5 pies a través del control remoto con una precisión de 1.5 pulgadas. El mecanismo final fue el resultado de varios métodos probados con el primer prototipo de prueba, y se añadieron características como la capacidad de intercambiar cables y una robusta cubierta con rodillos para evitar el desenrollamiento del cable en el diseño del CAD para futuras mejoras.

Además, se consideraron dos sistemas de propulsión: las bombas hidráulicas y las hélices. Las bombas hidráulicas se probaron y se concluyó que eran un método de propulsión fiable pero ineficiente. Las hélices se consideraron como método de propulsión secundario, pero no se probaron físicamente. En su lugar, se realizaron simulaciones de flujo en SolidWorks para ambos métodos de propulsión con el fin de determinar la fuerza de arrastre producida por el agua en la plataforma y la velocidad terminal de operación, con el fin de comprobar si se cumplían los requisitos de tiempo mínimo de funcionamiento y de distancia mínima recorrida. Considerando el recorrido típico de un día de la plataforma, se realizaron cálculos de energía utilizando las velocidades terminales para determinar las limitaciones de la batería a la hora de alimentar los sistemas de propulsión. Con los cálculos realizados se determinó que el sistema de energía sin que el panel solar cargue la batería puede hacer funcionar las bombas hidráulicas durante 6.48 horas, mientras que puede hacer funcionar las hélices durante 3.6 horas. Recargando la batería con el panel solar, el sistema de propulsión con hélices debería ser capaz de cumplir con el mínimo tiempo de funcionamiento. Con las bombas hidráulicas la plataforma puede recorrer una distancia de hasta 0.76 millas/día y con las hélices 3.46 millas/día.

En cuando a la capacidad de transmisión inalámbrica, se realizaron pruebas de distancia con 2 transceptores Arduino XBee Pro para determinar el alcance de la comunicación bidireccional y se determinó que la transmisión era capaz de alcanzar hasta 320 m. Además, el sistema podía transmitir hasta una lectura cada 30 segundos (1 lectura/30 seg) y guardar los datos en una tarjeta SD en tiempo real.

Todos estos resultados se muestran en la tabla siguiente (Tabla 1).

Parámetros	*Especificaciones técnicas*	*Especificaciones finales*	*¿Se han cumplido los requisitos?*
Plataforma	Relativamente pequeño; impermeable; robusto	33x18x5.4 pulgadas, impermeable	Cumplido
Propulsión	Mínima distancia recorrida: 2 millas/día	Bombas hidráulicas: 0.76 millas/día (con recarga)	No cumplido
		Hélices: 3.46 millas/día (con recarga)	Cumplido por encima de lo esperado
Mecanismo de despliegue del sensor	Profundidad requerida: 5 ft	5 pies	Cumplido
Energía	Tiempo mínimo de funcionamiento: 6 horas	Bombas hidráulicas: 6.48 horas (sin recarga) Bombas hidráulicas: 11.48 horas (con recarga)	Cumplido por encima de lo esperado
		Hélices: 3.6 horas (sin recarga)	No cumplido
		Hélices: 8.6 horas (con recarga)	Cumplido
	Preferible componente de energía renovable	Panel solar incluido	Cumplido
Transmisión de datos	Mínimo rango de transmisión: 300 m	320 m	Cumplido por encima de lo esperado
	Mínima velocidad de transmisión de datos: 1 lectura/ 5 min	1 lectura/ 30 seg	Cumplido por encima de lo esperado
Control	Preferible dispositivo autónomo, aceptable control remoto	Control remoto	Cumplido

Tabla (1). *Resultados finales.*

Conclusiones

El diseño final consiste en una plataforma móvil que alberga un mecanismo de despliegue de un sensor para bajar y subir el sensor de una manera fiable, un sistema de control remoto que controla tanto el sistema de propulsión como el mecanismo de despliegue del sensor, y un sistema de recarga de baterías alimentado por energía solar. Con el controlador, el usuario puede controlar la velocidad y la dirección de la plataforma, así como a qué profundidad se puede desplegar el sensor. Un panel solar montado en la parte superior de la cubierta recarga las baterías a bordo y permite al cliente utilizar el dispositivo durante largos periodos de tiempo. El diseño final (Figura 2) incorpora el sistema de bombas hidráulicas ya que las hélices no se pudieron probar físicamente.

La Figura 3 muestra una imagen de 3 vistas del diseño final. Este diseño ha preparado el camino para la creación de un dispositivo totalmente autónomo, alimentado con energía renovable en el futuro.

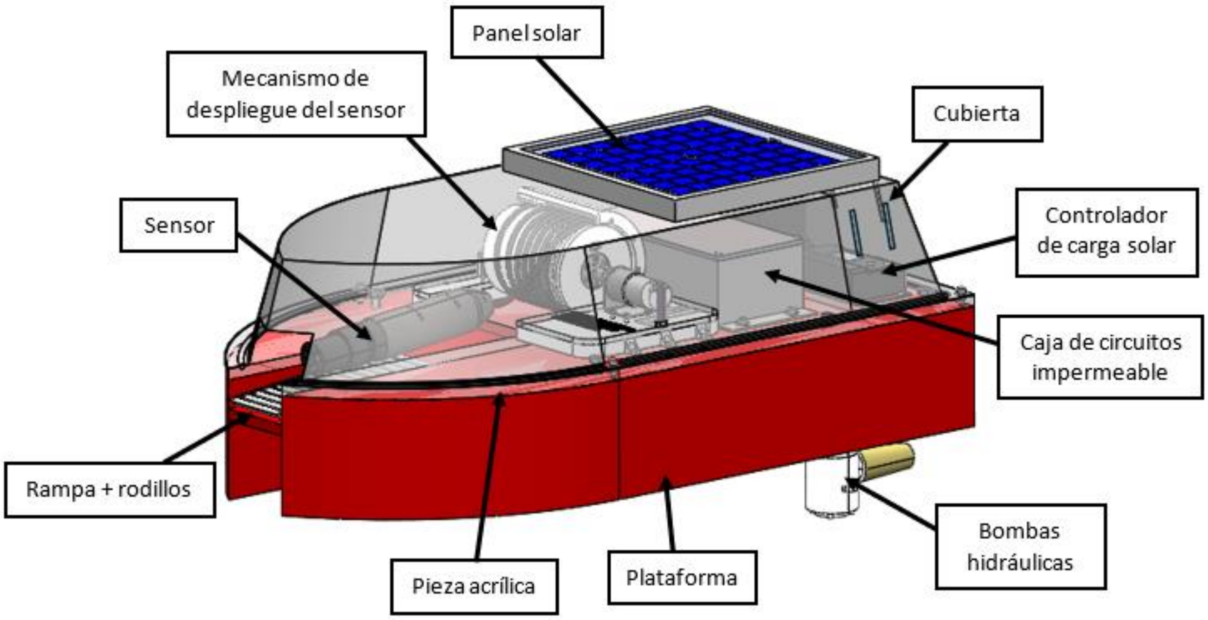


Figura (2). *Diseño final de la plataforma.*

Figura (3). *Imagen de 3 vistas del diseño final.*

SENSOR PLATFORM FOR ALGAL GROWTH AND WATER TEMPERATURE MONITORING

Author: Castilla Mena, María Luz.

Supervisor: Gutiérrez-Wing, Enrique.

Collaborating Entity: Boston University.

PROJECT SUMMARY

<u>Introduction</u>

Problem Statement

Water reservoirs are used to store clean water. In these drinking water reservoirs, algal blooms can signify cyanobacterial growth (Figure 4, [SANC19]), which release harmful toxins that are not only detrimental to aquatic life but also to public health. Early detection of these toxins (through the detection of cyanobacteria) is key to controlling and stopping the rapid growth of these cyanobacteria, thus preventing water contamination. Measuring parameters, such as temperature of the water at varying depths gives insight into the concentration of bacteria in that particular reservoir. However, current testing processes are labor intensive, inconvenient, and limiting as data can only be measured manually along the edges of reservoirs.

Figure (4). *Cyanobacterial water reservoir.*

State of play

The team's customer, Dr. María Teresa Gutiérrez-Wing, expects the team to create a device to improve the range and quantity of data collection to help detect early bacteria growth before it gets too concentrated to allow early treatment of the water. The group objective is to design and test a prototype mobile, aquatic platform that will be moved via remote-control to a desired location in the water reservoir, where it will deploy a sensor to a depth specified by the operator, and record and transmit data.

Project Goals

The goal of this project is to improve upon current methods that measure water parameters for the purpose of detecting harmful toxins in lakes and reservoirs. Therefore, the customer outlined several requirements the device must meet in order to successfully function. The team must consider all of these requirements in order to carry out the project successfully and build a device that will improve the customer´s scope of data collection.

Work methodology and Results

The platform size and design were chosen based on approximate weight calculations for the internal components of the device. Also, a ramp was carved at the front and a series of rollers were added to facilitate the deployment of the sensor. An acrylic piece was fashioned on top of the platform in order to properly secure these internal components to the platform. Additionally, a cover was designed for the platform in order to protect these main components from the water.

After many iterations, the final sensor deployment mechanism was capable of safely deploying a sensor to the full 5 ft depth via remote control with a 1.5 in accuracy. The final mechanism was a result of tried and tested methods with the first testing model, and features such as the ability to interchange cables and a sturdy rollery cover to protect against the cable unspooling were added in the CAD model for future improvements.

Furthermore, two propulsion systems were considered: hydro pumps and propellers. The hydro pumps were tested, and they were concluded to be a reliable but inefficient propulsion method. Propellers were considered as the secondary propulsion method but were not physically tested. Instead, SolidWorks Flow Simulations were conducted for both propulsion systems to determine the drag produced by the platform and the subsequent terminal velocity of operation, so that it can be checked if it met the requirements. Considering a typical run that the platform completes in a day, power calculations were conducted using these terminal velocities to determine limitations with the solar charging circuit and total run time. The calculations found that the power system may run the hydro pump propulsion system for 6.48 hours and the propeller system for 3.6 hours without the solar panel charging the battery. With recharging, the propeller system would also be able to meet the metric. Additionally, the hydro pumps could travel a distance up to 0.76 mi/day and the propellers 3.46 mi/day.

In terms of the wireless transmission capability, distance tests were conducted with the two Arduino XBee Pro transceivers in order to determine the range of bi-directional communication and it was determined that transmission was capable up to 320 m. Moreover, the system could transmit up to 1 rdg/30 s and save data to an SD card in real time.

In the following table all these results are shown (Table 2).

Parameter	***Engineering Specification***	***Final Specs***	***Did we meet spec?***
Platform	Relatively small; waterproof; sturdy	33 in by 18 in by 5.4 in, waterproof	Met Spec
Propulsion	Minimum travel: 2 mi/day	Hydro Pumps: 0.76 mi/day (w/ recharging)	Below Spec
		Propellers: 3.46 mi/day (w/ recharging)	Exceeds Spec
Sensor Deployment	Required depth: 5 ft	5 ft	Met Spec
Power	Minimum run time: 6 hrs	Hydro Pumps: 6.48 hrs (w/o recharging) Hydro Pumps: 11.48 hrs (w/ recharging)	Exceeds Spec
		Propellers: 3.6 hrs (w/o recharging)	Below Spec
		Propellers: 8.6 hrs (w/ recharging)	Met Spec
	Renewable energy component preferred	Solar Panel Included	Met Spec
Data Transmission	Minimum distance: 300 m	320 m	Exceeds Spec
	Minimum transmission: 1 rdg/ 5 min	1 rdg/ 30 sec	Exceeds Spec
Control	Autonomous preferred, remote control acceptable	Remote Controlled	Met Spec

Table (2). *Final results.*

Conclusions

The final design consists of a structural mobile platform that houses a sensor deployment mechanism to reliably lower and retrieve a sensor, a remote-control system that controls the propulsion and sensor deployment mechanism, and a solar-powered battery recharging system. With the controller, the user can control the speed and direction of the platform as well as to what depth the sensor will be deployed. A solar panel mounted on top of the cover recharges the batteries onboard and enables the customer to use the platform for extended periods of time. The final design (Figure 5) incorporates the hydro pump system since the propellers could not be physically tested. This design has paved the way for the creation of a fully autonomous, renewable-energy powered vessel in the future. Figure 6 shows a 3-view image of the final design.

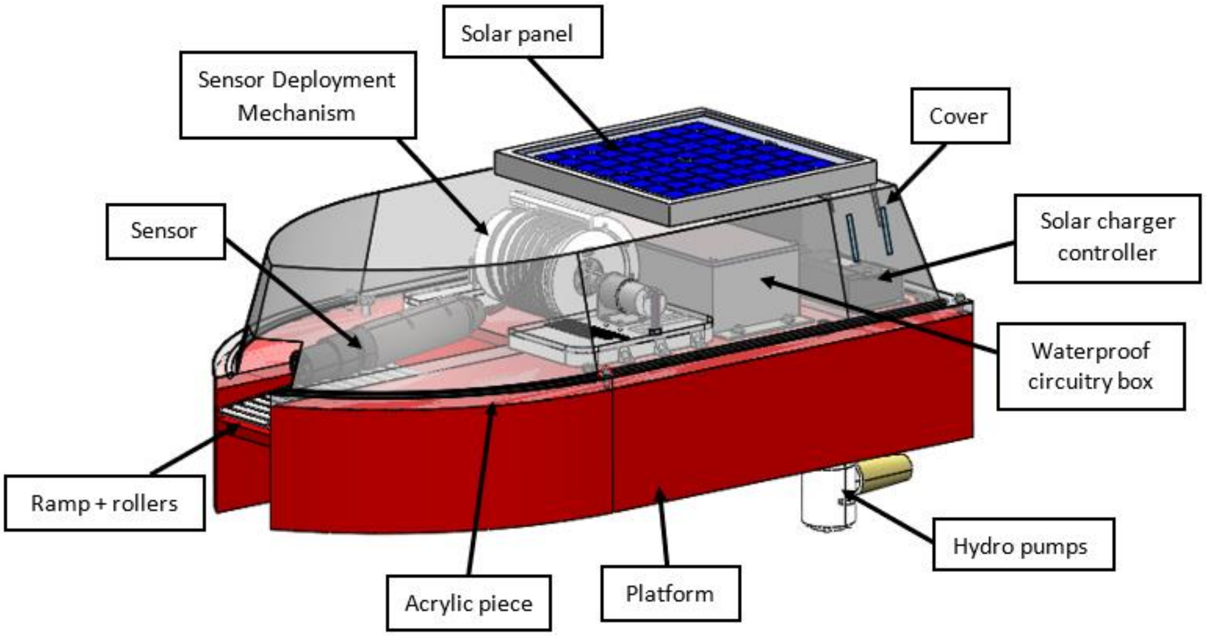


Figure (5). *Final design.*

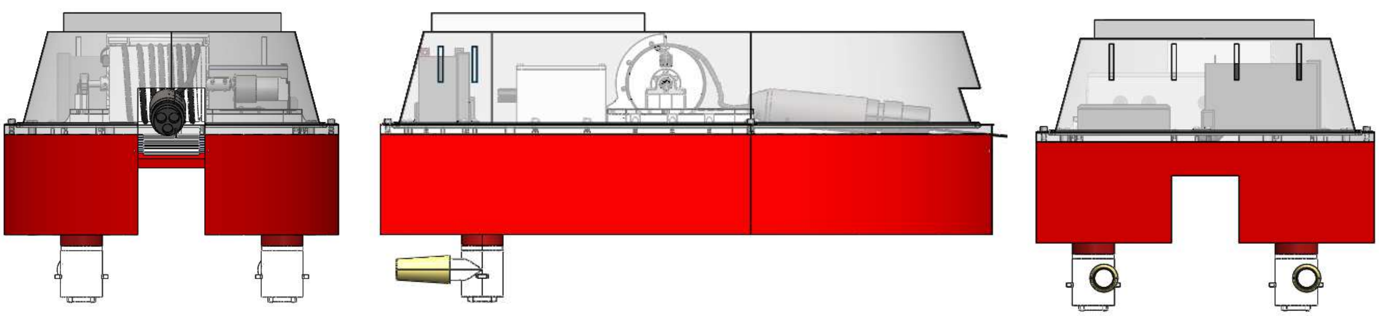


Figure (6). *Final design 3-view.*

Content

Figures

Tables

Graphs

DOCUMENT NO.1:

MEMORY

2. Acknowledgements

First and foremost, I would like to thank my team: Aleksandra Serafin, Elisabeth Hernández, Sydney Hall, and Justin González (Figure 7). They have been great partners, and thanks to our teamwork we have been able to carry out this project succesfully.

Figure (7). *From left to right: Elisabeth Hernández, Sydney Hall, María Luz Castilla Mena, Justin González, Aleksandra Serafin.*

Furthermore, I would like to thank our advisor Professor Gutiérrez for his invaluable advising and support throughout the course of this project. I would also like to thank Professor Hauser for his input and guidance. Thank you for taking the time to teach us new things, lending us important equipment and materials, and always offering help when we needed it the most. Without this support, the team would not have been able to advance as much as we have on this project.

I would also like to thank our customer, Dr. María Teresa Gutiérrez-Wing, for giving us the opportunity to work on this exciting and novel project. Additionally, I would like to thank Professor Thornton for offering us manufacturing advice, Professor Basu for energy assistance, and Cliff Merritt for allowing us to use the FitRec pool to test our initial sensor deployment design. Last, but not least, I would like to thank all the staff at the BU College of Engineering Product Innovation Center (EPIC) for both their help and support.

3. Introduction

Water reservoirs are used to store clean water for drinking, and watering crops. In these water reservoirs, algae growth can occur. Algal blooms can signify cyanobacteria growth which produces toxins that contaminate the water supply. These toxins can cause serious illness in humans consuming the water. These bacteria grow due to some factors, such as stagnant water, high temperatures, and nutrient runoff. In order to detect if there is any possibility of toxin development, water samples are taken to easily measure parameters such as temperature, level of oxygen, salinity, or nutrients. Currently, the customer tests water parameters manually, which wastes time and energy and limits the distance into the reservoir that samples can be taken.

These are the main reasons why the idea of this project was born. The customer expected the team to create a device to improve data collection of water parameters, in order to help detect early bacteria growth before it gets too concentrated to allow early treatment of the water.

The customer is Dr. Maria Teresa Gutierrez-Wing. She works in Louisiana State University (LSU) as an Assistant Research Professor in the Sea Grant College Program and an adjunct in LSU Department of Renewable Natural Resources. Her area of expertise covers the areas of water and wastewater, alternative materials, or microalgae production systems, among others. Before she started working at LSU, she owned a business in Mexico focused on water treatment of the aquaculture industry [AGGR18].

Her main request for the team is to build a mobile, aquatic device that can move to a desired location of the water reservoir, that will house and deploy a sensor to test water parameters, and it will collect this data into an SD card or either can be sent through wireless transmission. With this project proposal, her main objective will be achieved, as she will be able to monitor the whole water reservoir and receive instantaneous data of water parameters and be informed about the quality of the water at every moment. In the event the data indicates that there may be bacteria in the water, the customer will be prepared and will act right on time to treat the water before it gets contaminated. However, the customer has several requirements the device has to meet. In order to successfully accomplish the project, the team has to take into account all of these requirements.

First, all of these requirements will be explained, so that it is clear which objectives does the team have to accomplish when building the device. With these customer requirements in mind, the team will associate them with its respective engineering specifications the design must satisfy.

Then, the context of the project will be analyzed to understand the importance this project has in maintaining water reservoirs clean. Research will be done about the main aspects of the device: its structure, how it will move through the water, how it will deploy the sensor to measure water parameters, how it will be powered and how it will record and transmit data.

After doing the necessary research, the team realized which were the critical aspects of the project, the ones the team should first start working on. Therefore, the initial conceptual designs about the sensor deployment mechanism and the remote-control system of the platform will be explained, and some general brainstorming on how the device could look like will be shown.

Once explained the general brainstorming and the conceptual designs of the critical aspects of the device, it will be proceeded to describe the final design. First, it will be described how this device will be used by the customer in the water reservoir and all the components it consists of, which can be grouped into four main subsystems which will be developed in detail in the following sections.

Next, it will be analyzed if the requirements are met. For that, prototype evaluations will be shown, which include a power study, power calculations, a sensor deployment testing and wireless range testing.

Finally, a cost analysis section will be shown, along with a discussion section in which the final overviews of each subsystem is analyzed, and a conclusion section with future work improvements.

4. Requirements and Engineering Specifications

As mentioned before, the customer requires a means to record data remotely in order to facilitate the early detection of bacterial growth in a drinking water reservoir. The customer outlined requirements the device must meet to successfully function. The team must consider all these requirements in order to carry out the project successfully and build a device that will substantially improve the customer's scope of data collection.

4.1. Platform

The first task was to build a platform that will move through the water reservoir. The platform must be waterproof, and if designed to float it should be unflippable, as it houses components that cannot get wet. With respect to the platform's dimensions and weight considerations, it should not be too obtrusive, but it must be sturdy enough to withstand the wind and waves that may exist in the water reservoir. The customer's suggestion was "the smaller and lighter, the better". Also, the customer recommended that the device should be protected from birds that may be present in the reservoir.

4.2. Material Selection

Regarding the materials that could be used to build the platform, the customer restricted the use of heavy metals and any other hazardous materials since the platform may be partially submerged in a clean drinking water reservoir. Regulations would dictate which materials would be the most suitable and safe for the design. However, it was important that whatever material the team chose, the platform had to withstand the impact of the waves.

4.3. Propulsion

The platform was not required to move very fast but must be able to travel a minimum of 2 miles a day (2 mi/day). This does not encompass continuous movement of the platform though. It must be able to follow a typical run time in a day, that means moving to a desired spot in the reservoir where it will stop to take measurements and then move to a different spot and so on, covering 2 miles throughout the day.

4.4. Sensor Deployment Mechanism

Another essential part of the project is the design of the sensor deployment mechanism. The mechanism that the team will design must deploy a sensor into the water up to a depth of 5 ft below the surface. This is the zone where the algae grow and where the toxins tend to concentrate first. The customer uses several sensors to measure different water parameters, such as temperature, salinity, acidity, and oxygen levels. The initial request for the team was to focus on the temperature sensor and design a system to read its measurements. It is also important for the sensor to be protected from any potential damage such as water damage while on the platform.

4.5. Power

In terms of power, the device should be able to run continuously for 6 hrs. However, this does not mean it has to be in continuous movement for 6 hours, but it must be able to do 6 hours of data collection where the device would move to a desired spot and then stop and record data and so on, following the typical run time for the platform in a day. Additionally, the customer suggested adding a renewable energy component to the platform to recharge the onboard batteries to elongate time of data collection and to have a more sustainable option for battery charging. As stated before, the customer wanted the device as light as possible so adding a recharging component would lessen the need for large and heavy batteries.

4.6. Data Transmission

Furthermore, the device should be able to both store and transmit the data collected from the sensor. The data should be stored in an SD card as well as wirelessly transmitted to the user. The customer would be satisfied with 300 m for this prototype as this will cover the most critical parts of the water reservoir but would ideally prefer a larger range. Additionally, the device should transmit/receive one reading every 5 minutes (1 rdg/5 min).

4.7. Control

Finally, the customer suggested building an autonomous device. The customer wanted a platform that could be pre-programmed with a movement route and sensor deployment protocol to allow for operator-free testing. As a secondary option, a remote control could be used to allow the user to control the platform's movement and sensor deployment mechanism.

With these customer requirements in mind, the team made a series of engineering specifications that the design must satisfy to be successful Table 3.

Parameter	***Engineering Specification***
Platform	Relatively small; waterproof; sturdy
Propulsion	Minimum travel: 2 mi/day
Sensor Deployment	Required depth: 5 ft
Power	Minimum run time: 6 hours Renewable energy component preferred
Data Transmission	Minimum distance: 300 m Minimum transmission: 1 reading/ 5 min
Control	Autonomous preferred, remote control acceptable

Table (3). *Engineering specifications corresponding to each parameter of the project.*

5. Background

The importance of water as a resource is often overlooked by many since it is always readily available and generally safe to consume. However, there are many countries facing water scarcity, and human activity has increased fresh water contamination. Worldwide, one in three individuals are affected by water scarcity, and at least 10% of the human population consumes food that is irrigated with wastewater containing dangerous chemicals or harmful bacteria [HERO19]. Because of this unfortunate reality, it is essential to engineer solutions for the betterment of mankind. With that in mind, the objective for this capstone is to prototype a platform for measuring water parameters.

Although there is no defined method for doing so across the globe, it is necessary to start with a small-scale issue close to home. To illustrate, in the state of Louisiana, over one-third of the land is covered by water resources [RUTH15]. All this water is utilized for recreation, human consumption, crop irrigation, and storage. As mentioned earlier, water is an incredibly important and versatile resource. It is necessary to monitor and protect water resources from any depletion or contamination for the betterment of those who depend on it.

A body of water contaminated by nutrient runoff, which is commonly a result of excessive fertilizer usage by humans in coastal regions, will saturate the water's nitrogen and phosphorus levels [NOAA19]. If a body of stagnant water contains high levels of nutrients, cyanobacteria will quickly grow and degrade the water quality [TCHO12]. If left unchecked, cyanobacteria pose a big problem in reservoirs because it damages the local ecosystem, disrupts water purification processes, and produces toxins that affect humans [EPA_17]. Now, the client, Dr. María Teresa Gutiérrez-Wing, focuses on water and wastewater treatment, and she requires an efficient method to monitor Louisiana reservoirs.

Dr. Gutiérrez-Wing currently performs tests on water samples that are collected near the shore. Although she is able to verify the presence of cyanobacteria, her samples are inaccurate representations of the ecology in the center of reservoirs. Also, she would like to retrieve more data at a faster rate for different depths of water over a period of at least 6 hours. In order to achieve this, extensive research was done for potential platform designs, propulsion methods, and power methods. But first, it was necessary to research current solutions to similar problems.

5.1. Benchmarking

Before concept selection, it was necessary to research already existing solutions to similar problems to formulate a well-conceived design of a prototype. With this research, it was determined that the most similar existing invention was the Datamaran pictured in Figure 8 [AMS_19].

Figure (8). *Datamaran Mark 8 in operation.*

The Datamaran is a wave-piercing, catamaran-style, autonomous vehicle that relies on solar and wind power to monitor oceans. The platform houses multiple sensors and uses artificial intelligence to navigate. Aside from this, it is capable of self-righting after being flipped over. Arguably, its most impressive feature is being able to adjust to wind and sea currents while being commanded to move to different locations from the shore [AMS_19]. After learning about the Datamaran, the team began researching the individual components that go into building this type of device.

5.2. Hull Research

To start, many different types of platform designs were researched. The possible designs for platforms were a dual hull catamaran (Figure 9, [PB__16]), a hydroplane hull (Figure 10, [JOYC19]), and a monohull (Figure 11, [AMIT19]). Each type has its own advantages. To illustrate, catamarans have excellent stability in rough water and strong winds [MINN19]. Moreover, the hydroplane hulls excel at high speed because its streamlined body greatly reduces drag [PIKE11]. Finally, monohulls are most agile, best in turbulent water, mostly self-righting and easily move in any direction [PARK18].

Figure (9). *Catamaran hull.*

Figure (10). *Hydroplane hull.*

Figure (11). *Monohull.*

Another aspect of the hull to consider was its material composition. Some materials of interest for the hull were: styrofoam, wood, steel, titanium, aluminum, and fiberglass. One important parameter to consider for the hull material was density. For a material to float on a fluid, it must be less dense than that fluid so it will have a larger buoyant force. For this project, using metals was infeasible due to oxidation, cost, and limited manufacturing techniques. Also, some heavy metals could leach into the water, and this is an environmental safety concern [OLEA09], [TCHO12]. Composites like wood, styrofoam, and fiberglass will float, but they are prone to water damage so they would need to be waterproofed. Selecting the proper material is crucial, as it will determine the platform's performance.

5.3. Propulsion Research

The next research topic explored was propulsion methods. Modern aquatic platforms utilize various methods to move like propellers and pumps, but other methods were also explored. To start, aquatic thrusters were considered due to their high power-to-weight ratio [WIKI19]. Although thrusters can be compatible with Arduino, they were the most expensive form of propulsion. The ones considered for this platform cost over $200 and are pictured below (Figure 12, [BR__19]).

Figure (12). *T200 Thruster.*

It was ultimately determined that thrusters were too expensive. Furthermore, paddle wheels were considered, but they required intricate manufacturing even though they were about 90% efficient [MORT__]. A fish fin (Figure 13, [MASO15]) was explored as a potential alternative, but it was also considered not practical due to inefficiencies and intricate design [YANG70].

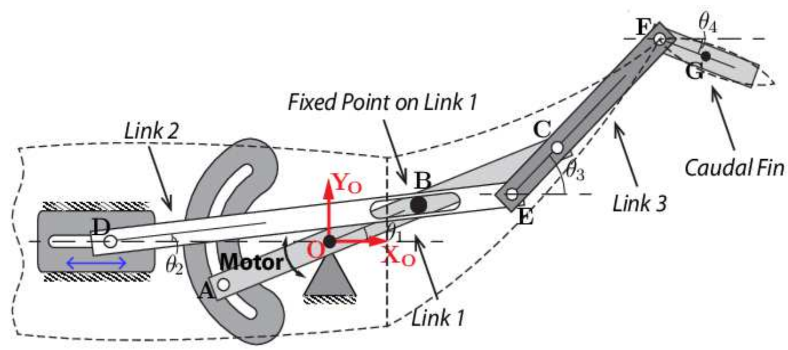


Figure (13). *Fin mechanism.*

Although they waste a lot of energy, pumps and propellers were deemed to be the most feasible modes of thrust for the scope of this project. Most pumps and propellers are advantageous due to their compact size, ease of operation, and availability [WIKI20].

5.4. Sensor Deployment Research

Apart from the chosen propulsion method, it was also necessary to investigate possible sensor deployment techniques. For this, telescoping poles and pulleys were considered. Between the two options, it was determined that telescoping poles were inferior to a pulley mechanism. Due to its linear motion, it would be difficult to create a sturdy telescoping pole that would reach the 5-foot-depth requirement. It was also found that a pulley could be combined with a reel and motor to ensure a secure and controlled deployment of a sensor attached to a wire. In the end, the pulley mechanism was favored due to its simplicity and reliability.

5.5. Energy Research

Finally, sustainable means of properly powering the device were explored. Because the platform was going to be outside during its operation, it was only logical to research solar, wind, and wave energy harvesting methods. Also, implementing a renewable source of energy would prolong the platform's operation time.

To start, solar panels can be monocrystalline or amorphous (Figure 14). Monocrystalline panels ([THD_19]) are more powerful than their amorphous thin-film counterparts, but amorphous panels are more efficient in low-lighting [SST_19]. It was also determined that most compact solar panels for microgrid applications are designed to charge 12-volt DC batteries. For optimal power generation, solar panels must be perpendicular to solar rays. It is also recommended to avoid shadows, as shadows can drop power generation by at least 50% [BURD20]. Due to the variable nature of solar power, a solar charge controller would also be needed in order to protect the battery from power surges [ASIR19].

Figure (14). *Monocrystalline (left) and amorphous thin-film (right) solar panels.*

Other forms of energy generation that were looked into were wind and wave, but these forms of generation were impractical. Mini wind turbines were too large for a small scale platform and wave energy would not be present in relatively calm reservoirs. Solar power appeared to be the ideal energy generation method.

After looking into energy harvesting methods, multiple batteries of varying capacities and compositions were researched. To maintain the platform's longevity, the selected battery must be rechargeable, safe, light-weight, energy dense, and reasonably priced. Three batteries of interest for solar applications are lithium-ion, lead-acid, and saltwater. Although lithium-ion is the lightest, most compact, and most energy dense, they tend to also be the most expensive battery technology. Saltwater batteries are a new, relatively untested battery technology. They are easily recyclable and environmentally safe due to the lack of heavy metals. Finally, lead-acid batteries contain heavy metals, but they are cheap and capable of storing lots of energy [ES__20]. In brief, the most efficient battery technology is very expensive and so it was unattainable for this project. However, some battery types like lead-acid are sufficient to theoretically meet the requirements set forth by the client.

6. Conceptual Design

6.1. Functional Decomposition

For initial concept designs, the group decided to focus on the platform control system and sensor deployment mechanism after determining these were two critical aspects of the project. Additionally, the group brainstormed different propulsion techniques and decided it would be best to have a very efficient, slow moving mechanism. The group considered monofin and paddle wheel designs which were modeled on CAD as alternatives to more conventional propulsion methods like hydro pumps and propellers. The goal for the initial conceptual design was to create an initial remote-controlled platform as well as design a sensor deployment mechanism.

In the table below (Table 4), the means considered for each function are stated and the methods utilized in the final design are highlighted. The main functions of the platform are propulsion, sensor deployment and data collection. Additionally, a renewable method to power the platform was considered. The platform would also have to be self-righting to ensure the data collection would not be interrupted by a wave overturning the platform.

Functions	**Means**				
Sensor Deployment	Pulley	Retractable Pole			
Propulsion	Electric Thruster	Paddle-Wheel	Propeller	Monofin	Air Boat
Vehicle	Floating (Mono/Dual)	Submarine	Drone		
Self-Righting	Weighted Hull/ Buoyant Top	Balloons	Water draining/filling system		
Renewable Power	Solar	Wind	Wave		
Data Collection	SD Card	Transmit to shore			

Table (4). *Function-Means Chart.*

6.2. Concept Selection

In the Pugh chart below (Table 5), some concepts the team considered when designing the platform are shown.

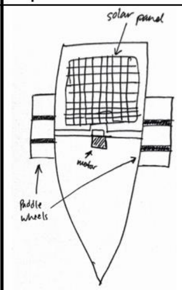
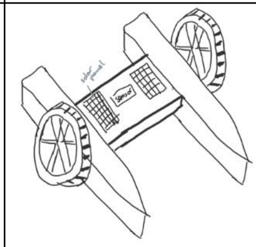
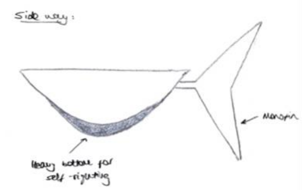
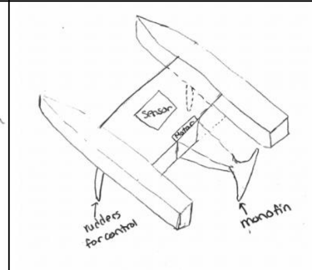
		Design 1 Monohull with paddle wheels	**Design 2** Catamaran with paddle wheels	**Design 3** Monohull with monofin	**Design 4** Catamaran with monofin
Design Criteria	**Weight**				
Self-Righting	2	+	0	+	0
Adequate propulsion	1	0	0	+	+
Space for sensor deployment	2	+	+	+	+
Space for battery and motors	2	+	+	+	+
+		6	4	7	5
0		1	2	-	1
-		-	-	-	-
Total Points		6	4	7	5

Table (5). *Pugh Chart of initial designs considered.*

Based on the concept selections, the goals for the feasibility demonstrations were to build and test a remote control system using foam as the material for a simple platform, and to build and test a sensor mechanism using the concept of a threaded spool.

After analyzing the fin and the paddle wheels as alternative propulsion methods, the team concluded they were not the most suitable options for this project. While the fin was an interesting original idea, it required extensive testing in order to build a suitable design. The team preferred to use its valuable time improving the sensor deployment mechanism or developing a better remote-controlled system. An idea of how the fin mechanism would have worked if it had been employed as a propulsion method is shown in Figure 15.

Regarding the paddle wheels, in order to correctly work, they need to meet a series of requirements. Paddle wheels require a large blade area to have a very low slip, and each blade should be at least one third of the length of the boat with an arc of immersion of less than 60 degrees for an efficient operation. In addition, at least one paddle must always be in the water. The wheels must have a large diameter, and ideally the peripheral speed should be 10% faster than the boat speed for the paddle wheel to be 90% efficient. As this results in a massive wheel relative to the boat, a wheel of practical proportions would be 75% efficient [BDN_08]. Most of these requirements need extensive study with the final dimensions of the platform. The team would have had to play with the size of the wheels for them to suit the boat dimensions as well as to determine the power required for various combinations of diameter and width. Figure 16 shows a first CAD prototype of a paddle wheel adjusted to initial estimations of the platform's dimensions. For these reasons, it was decided that the team would move forward with simpler propulsion methods such as hydro pumps and propellers.

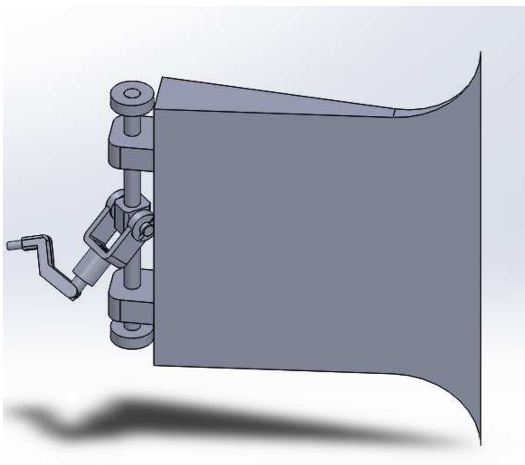


Figure (15). *Monofin CAD design.*

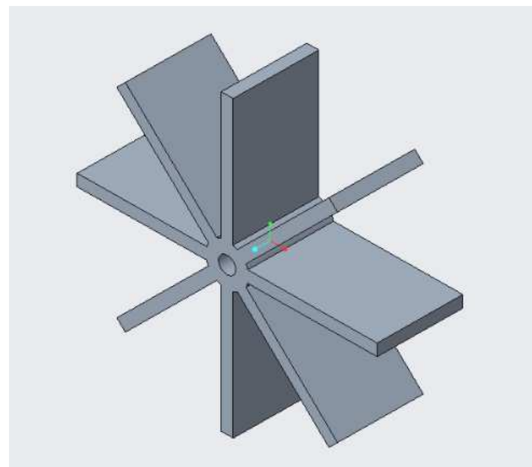


Figure (16). *Paddle wheel CAD design.*

Additionally, ways to power the platform had to be taken into consideration. A sealed lead-acid battery was determined to be the best match for this project considering factors such as cost, weight, energy density, lifetime, and safety, as shown in Table 6.

Type	Cost	Energy Density	Weight	Lifetime	Safety
Li-Ion	1	3	3	3	1
Ni-MH	2	2	2	3	2
Lead-Acid (sealed)	3	1	1	1	3

Table (6). *Battery Characteristics Comparison.*

6.3. Proof of Concept

6.3.1. Remote-controlled platform

To start the remote-controlled platform design, the group had to build a preliminary prototype made of foam and propelled by two hydro pumps—one at each side of the platform, as shown in Figure 17. Based on this prototype, the group was able to begin a preliminary bill of materials, test a remote-control system, and get a general idea of dimensions.

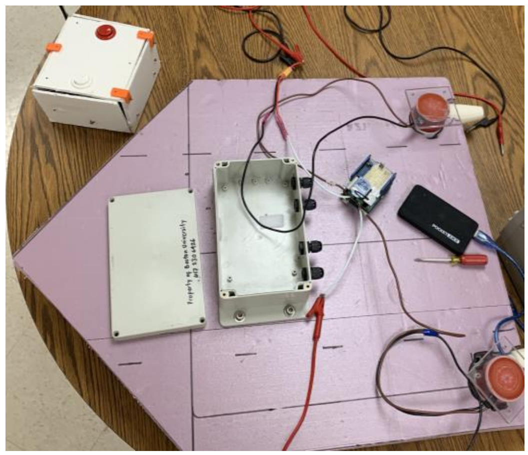


Figure (17). *First prototype of the platform.*

The remote control that the group designed works the following way: it has two buttons that act as switches for each of the hydro pumps—controlling when and how long they will run as well as the direction of the platform. The circuitry used to control the platform is shown below in Figures 18 and 19.

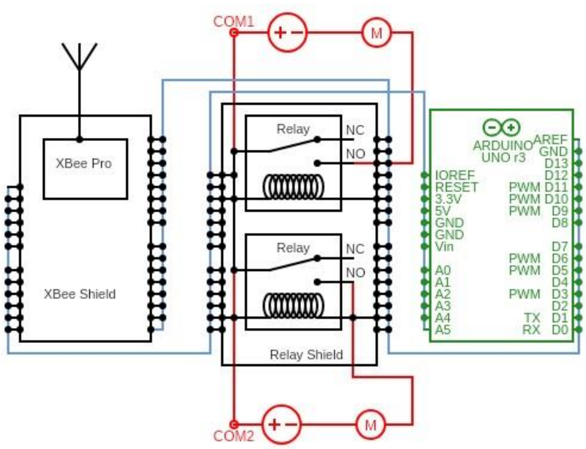


Figure (18). *Circuit used on the prototype platform.*

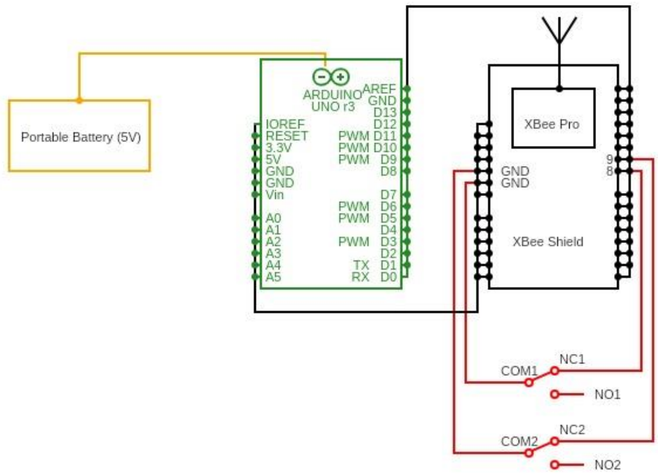


Figure (19). *Circuit used for the remote control.*

The controller's purpose is to control the platform's direction with its 2 manual relay switches and XBee shield. When a switch is manually pressed, it sends a signal from the XBee Pro in the controller to the XBee Pro on the platform. The platform circuit then responds by turning on the corresponding hydro pump. Each Arduino is connected in parallel with an XBee Shield and XBee Pro, and both Arduinos are programmed before the platform's operation. The controller Arduino is responsible for sending the signal through the XBee Pro, and the platform Arduino is programmed to turn on the hydro pumps by controlling the relay switches. Essentially, the XBee Pros are crucial to this circuit since all control signals travel between them.

6.3.2. Sensor Deployment

An important aspect of the project is to create a mechanism to hold and deploy a sensor that will detect parameters of water quality at various depths in a reservoir. To deploy the sensor up to 5 ft of water, a design based on the idea of a fishing reel was formulated. This mechanism would consist of a spinning reel that moves with a roller framework in order to keep the wire untangled, as shown in Figure 20. The outer framework; however, ended up not being needed and was not used in testing. A grooved spinning reel was added to hold at least a 0.25 in thick cable.

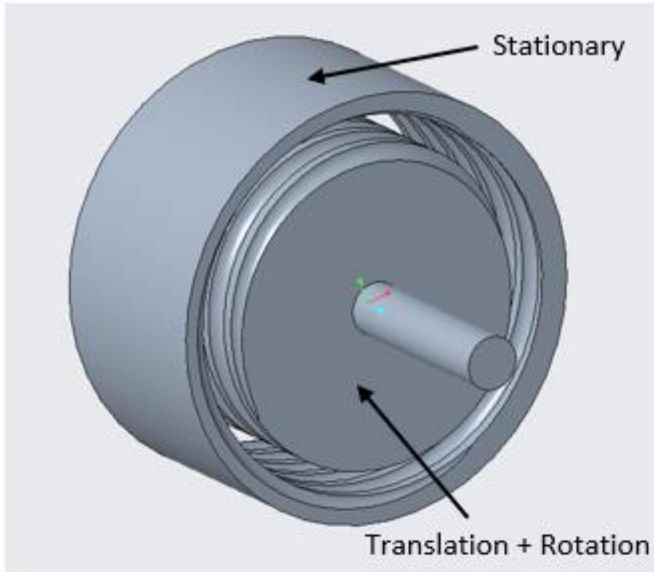


Figure (20). *First sensor deployment idea.*

The first sensor deployment design consisted of a cylinder with outer grooves and a cylinder with inner grooves. The idea behind this design is that the cable will fit between the grooves and while the outer cylinder is held stationary the inner screw will rotate and push the cable outward. However, using a 3D-printed small scale version, it was determined that the friction was too high to reliably force the cable out of the shell. Figure 21 shows a preliminary sketch of the sensor mechanism.

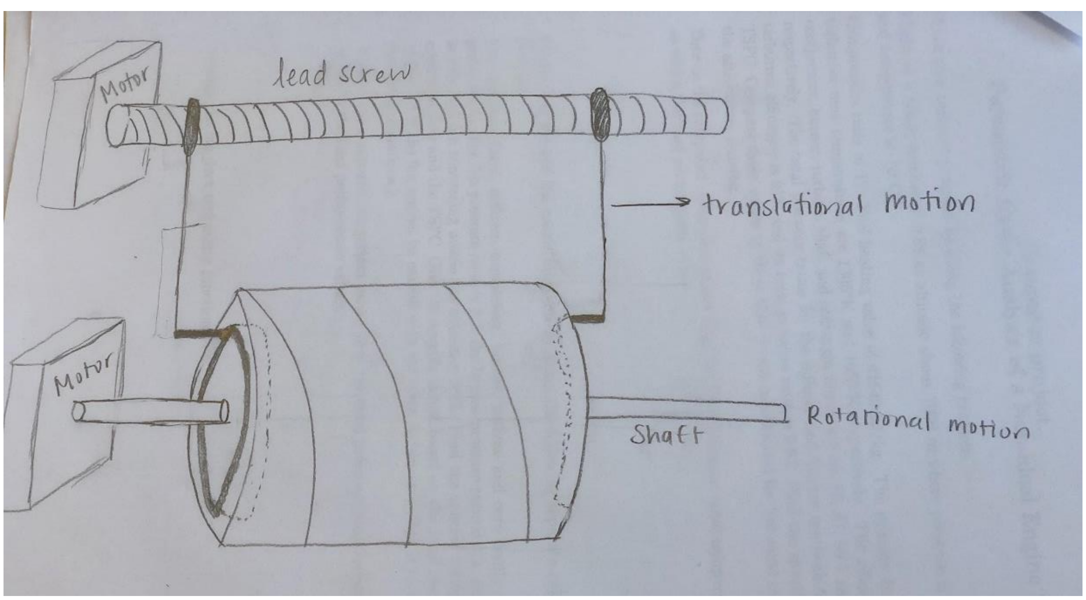


Figure (21). *Sensor deployment sketch.*

Initially, both motors remain stationary. However, upon further thought, the group decided to change the motor that is attached to the grooved screw as part of the moving system controlled by the linear drive. A preliminary system was then built using a linear drive fitted with 3D printed parts as can be seen in Figure 22. For proof of concept, a simple lead screw was used, which translated a black piece of acrylic, this piece of acrylic had another shaft mounted on it. A large screw is attached to this shaft and functions as a pulley when turned by the motor-controlled shaft. The grooves in the screw allow the cable to unwind and rewind without tangling. While the cable unwinds, the linear drive moves forward, allowing the wire to always deploy from one point. For testing purposes, the second motor was not attached to the acrylic because it was tested by hand.

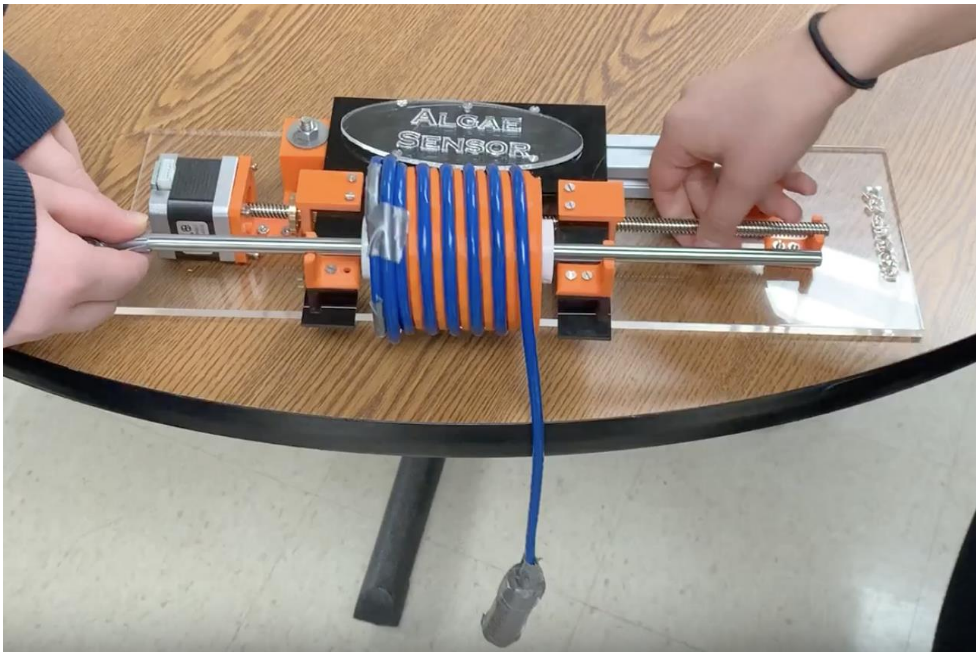


Figure (22). *Final conceptual design of sensor deployment mechanism.*

As shown in the previous figure, the linear drive is turned as the shaft with the attached screw is turned. This design results in the sensor unwinding and rewinding at the same point.

6.4. Concept Conclusions

In conclusion, the initial platform created fall semester enabled the group to test the remote-control capability, to understand proper platform dimensions, and to create a preliminary bill of materials. The group decided that it was easier to start with a user-dependent program than an autonomous, programmable platform. For the sensor deployment mechanism, the group chose a pulley with screw and linear drive system to deploy the sensor at one point.

The early platform circuitry demonstrates the feasibility of using a remote-controlled system and two hydro pumps to control motion and direction. The controller system works correctly when the user presses the buttons from across the room. One advantage of this design is using the hydro pumps as directional control while one disadvantage is how cumbersome it is to encase all the necessary wiring.

Although the controller successfully controlled the hydro pumps, there was a clear delay in the signal. The hydro pumps turned on/off a couple seconds after pressing the buttons. With this in mind, the group needed to test the distance limitations of the Xbee Pro to ensure that the remote control works for distances of at least 300 m to satisfy the customer requirement. Also, renewable methods for recharging the battery still needed to be incorporated to ensure that the platform operates for the full 6 hours that the customer requires.

The sensor deployment design demonstrates the ability to deploy a sensor with a weighted casing, without the cable tangling or overlapping. The group can conclude from testing that this design, with some added components such as motors and rollers to guide the cable, will function as needed. Additionally, the dimensions of the system need to be reevaluated and optimized for the final design.

7. The Design

7.1. Operation of the Device

The mobile platform is a multi-functional design that is able to remotely house, transport, and deploy a sensor at a specified location and to a specified depth. Data is recorded by the sensor and is both stored locally in an SD card and wirelessly transmitted to a remote location. The full platform can be seen in Figure 23. The drawing of the full assembly can be seen in Appendix 13.6.1.

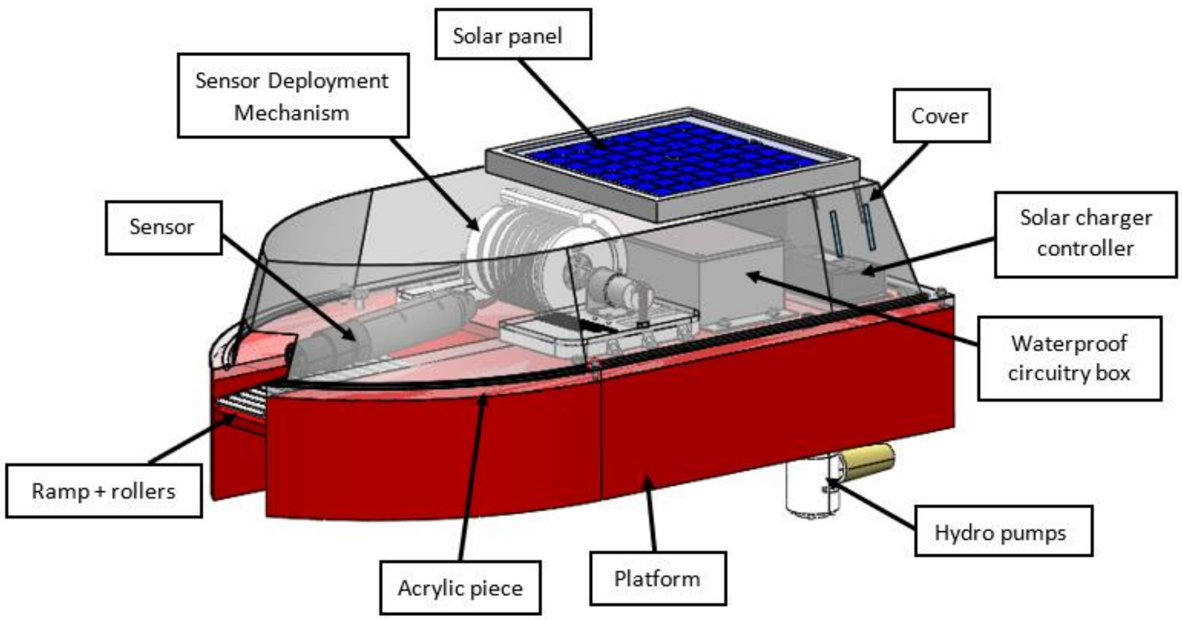


Figure (23). *Final prototype of the platform with all components.*

To paint a picture of how the mobile platform would ideally be used: first, the operator must charge the on-board batteries before plugging them back into the circuitry. Second, the desired sensor would be properly plugged into the cabling of the sensor deployment mechanism and screwed into the sensor casing. The operator should make sure that the remote control and on-board Arduino are turned on. Finally, the solar panel cable will be plugged into the circuit and the cover placed on top of the platform. The foam base is the main structural support of the mobile platform that holds and houses all the components of the whole system. The cover primarily protects the internal components of the design from the water, and also it is used to hold the solar panel in full view of the sun.

Next, the platform can be placed into the water, ensuring that the hydro pumps or propellers are completely submerged in the water. The operator can then use the remote control to turn on the propulsion system and use the controls to move to the desired location. When the platform has arrived at the location, the operator can activate the sensor deployment mechanism. With the remote control, the operator can deploy the sensor to a depth up to 5 feet. The sensor will take the data and the Arduino will store this data in an SD card and transmit a subset of this data directly to the operator.

The platform can move to different locations in the reservoir with the propulsion system. The solar panel recharges the battery that powers the propulsion to ensure the desired duration of data collection. Once the operator is done taking data, the platform can be maneuvered back to the shore and the operator can take it out of the reservoir.

Figure 24 depicts the four main subsystems of the platform: structural, mechanical, electronics, and power.

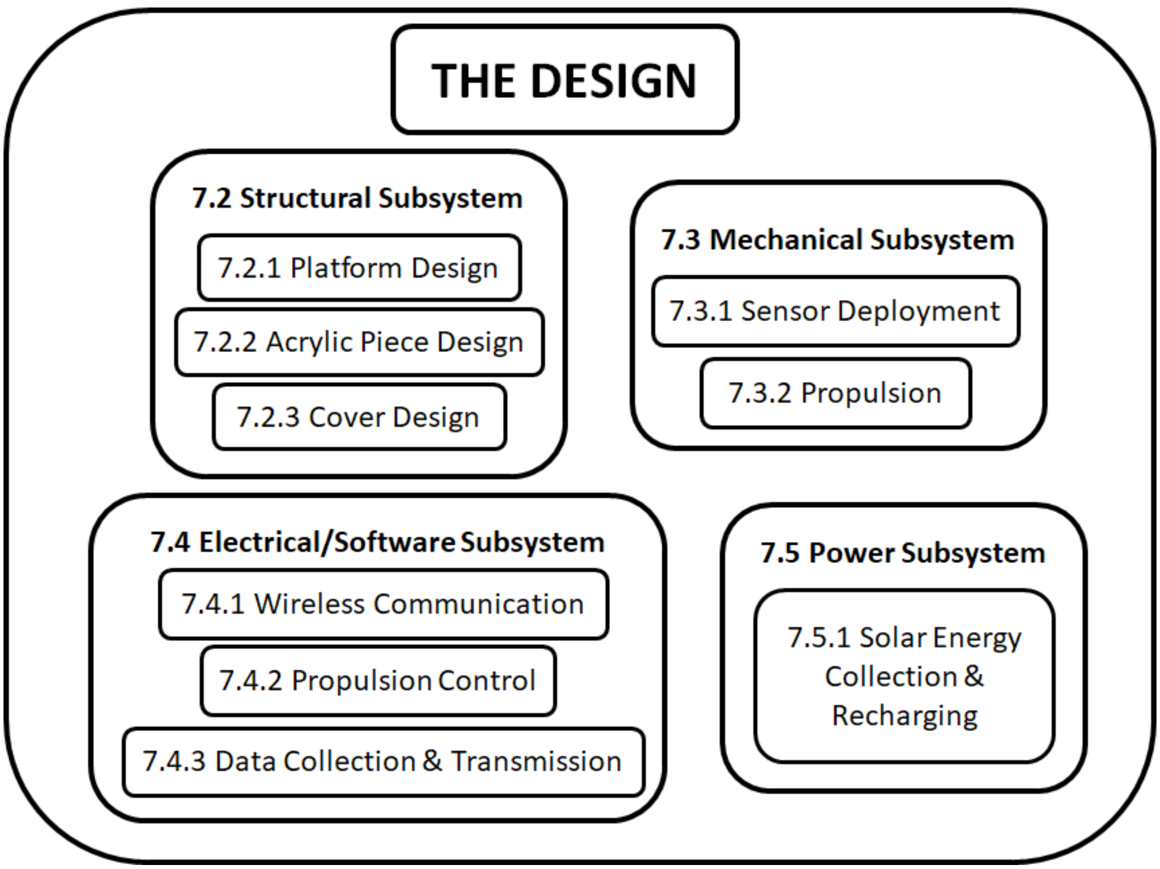


Figure (24). *Main subsystems of the platform.*

7.2. The Structural Subsystem

The structural components of the design include the platform, acrylic securing piece, and enclosing cover.

7.2.1. The Platform

The customer's requirements for the platform were to design it to be as light and small as possible. The design also must be able to withstand the waves caused by the wind in the water reservoir.

In order to make an accurate design of the platform, the team first had to approximate the weight of the components the final device would be carrying. As mentioned in previous sections, the device has to carry the sensor and sensor deployment mechanism, the waterproof box carrying the electronic system of the deployment mechanism, and the electric circuit that powers the propulsion system, consisting of the solar panel, the solar charger controller and the rechargeable battery. The platform also has to hold the acrylic securing piece and cover.

By estimating the weight of each component, the total weight the platform would be carrying would be around 23 lbs. A safety factor of 1.5 was applied to this weight when making the weight calculations to compensate for any unforeseen added weight and ensure that it floats when carrying all the desired components. Therefore, it was estimated that the platform will have to carry 35 lbs.

An important factor that determines how much weight the device can support is the material from which the platform is built. With research, the team concluded that the best material to build the platform with was waterproof foam. The Extruded Polystyrene (XPS) is a closed-cell rigid foam that prevents waterlogging. Its most remarkable properties are its low density compared to water, which makes it lightweight and able to float, and its relatively easy machinability.

Furthermore, based on the size dimensions of the sensor deployment mechanism, the waterproof box, the rechargeable battery, and after considering several other estimates, the team decided to draft initial platform dimensions of 33 in long by 18 in wide. After applying the buoyancy calculations, these initial dimensions were finalized. The thickness was determined by the decision to adhere three layers of XPS in order to make a reasonably high platform and thus further prevent water from getting inside the platform. Each layer of foam was 1.8 in thick, therefore the thickness of the platform with three layers of foam was 5.4 in. The base of the platform would be built with one layer of foam so 1.8 in thick and the hulls would be formed by two layers, therefore 3.6 in thick. For the bottom of the platform, it was considered monohull and dual hull designs but ultimately the team chose a dual hull design in order to ensure maximum stability of the platform when collecting data. Each hull is 33 in long by 7.2 in wide by 3.6 in thick. Note: The foamboard adhesive (Loctite PL 330 Foamboard VOC Latex Construction Adhesive) used to adhere the 3 layers of foam was described as water resistant, not waterproof and since this platform was meant to be immersed in water, the platform was fully painted red to ensure the adhesive was not in direct contact with water. Figure 25 depicts these dimensions, and Figure 26 shows the built platform.

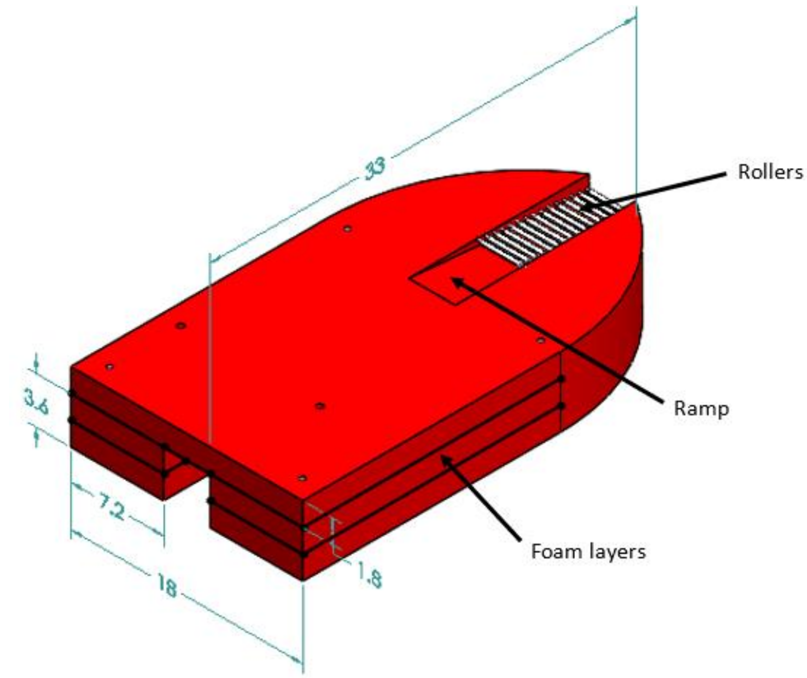


Figure (25). *Platform's dimensions.*

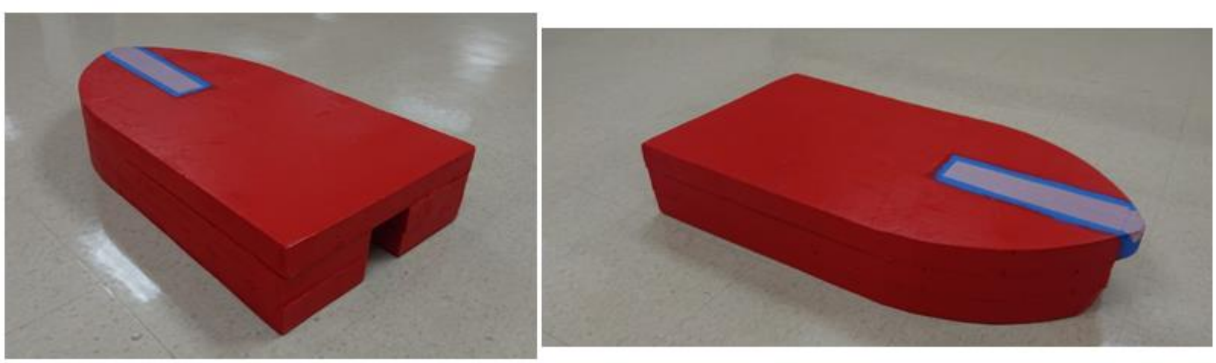


Figure (26). *Built platform.*

With these dimensions in mind, as mentioned above, the team proceeded to calculate the buoyant force to ensure that the platform would float when carrying the desired weight. In order for the platform to float, the buoyant force must be greater than the weight of the whole device. All the calculations to prove the following information are shown in the Appendix 13.3.2. First, the maximum weight the platform can carry without sinking in the worst case scenario was calculated. This occurs when the hulls are completely submerged in water and there is risk that the components can get wet, this corresponds to 3.6 in of submersion. For this case, the maximum weight carried by the platform should not be greater than 50 lbs. As the platform should carry 35 lbs, it has been proven that the platform will float without the hulls being completely submerged in water. When carrying the desired weight of 35 lbs, the hulls will be submerged 2.6 in in the water. That is 72.22% of the total thickness of the hull will be submerged in water. The team considered changing the dimensions for a shorter height of the platform but it was decided that it was better to keep these dimensions in case more components were to be added to the platform in the future, adding more weight to the design.

Another platform design considered was to shape a fiberglass hull and apply a waterproof treatment. While this design was more sophisticated than foam forming, the fiberglass and waterproofing treatment would have a steep learning curve. This design did not seem feasible for our short timeline.

7.2.1.1. The Ramp and Rollers

As can be seen in Figure 25, a shallow ramp was carved out of the bow of the platform where the sensor will be housed and deployed, and a series of rollers were added to facilitate deployment. This allows the sensor to be smoothly released off the platform and reeled back onto the platform without damage. Figure 27 shows a more detailed image of how the ramp looks and Figure 28 shows how the rollers were assembled.

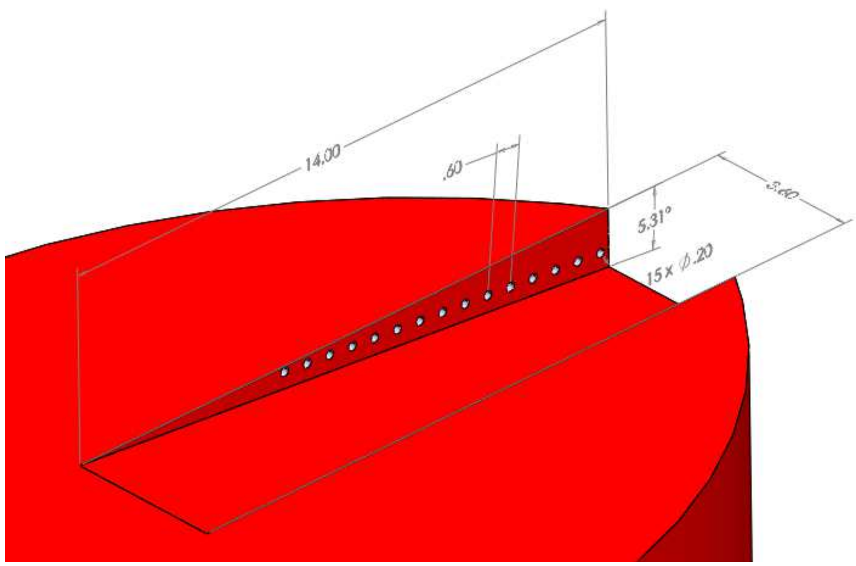


Figure (27). *Detailed view of the ramp and the holes carved for the rollers.*

The customer's sensor has dimensions 13.5 in long and 1.8 in diameter and weighs 1.26 lbs (Figure 33). Therefore, the ramp had to be at least 13.5 in long in order to contain the sensor. Final dimensions of the ramp included a 14 in length and a 5.3 degree slope. On either side of the ramp, 15 holes of 0.2 in diameter were carved out in order to mount the shaft of the rollers; the rollers are separated 0.6 in apart.

Each of the 15 rollers had identical dimensioning as depicted in Figure 28. As the ramp is not very steep, the dimensions of the rollers could not be too large. Each roller consists of a shaft and an outer cylinder. The shaft has a 0.2 in diameter and it is 4 in long, while the cylinder has a 0.35 in outer diameter, is 0.05 in thick and is 3 in long. For a complete drawing see Appendix 13.6.5. The shaft can be easily bought in any hardware store, whereas the cylinder would need to be machined to size out of HDPE, PVC, or equivalent.

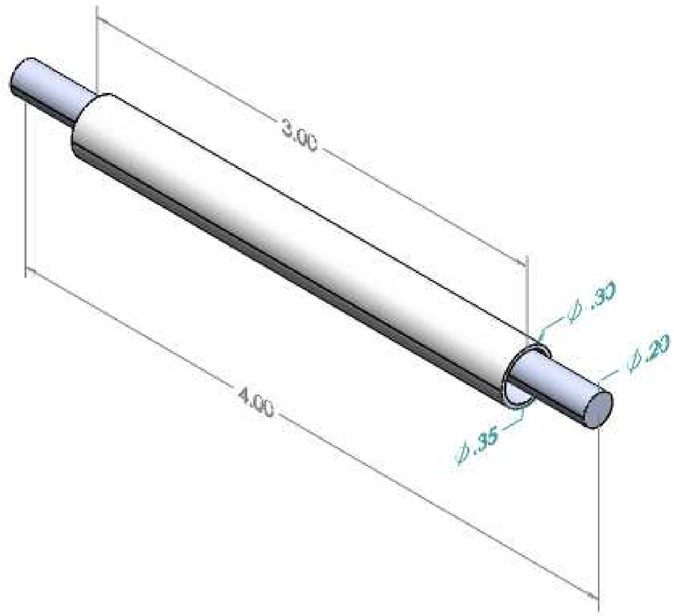


Figure (28). *Dimensions of the rollers.*

To conclude, the platform was constructed out of three Extruded Polystyrene (XPS) rigid waterproof foam sheets, adhered together, and then carved to shape. The rounded bow shape and hulls were carved with a band saw. A shallow ramp was carved out of the bow of the platform where the sensor will be housed and deployed, and a series of rollers were added to facilitate deployment.

Depending on the propulsion method used, the platform will have different features in order to be properly adapted to each propulsion system. The platform created for the hydro pumps has the hydro pump securing square cutouts on the bottom of the hulls as well as a wiring hole extending through the whole platform. This drawing can be seen in Appendix 13.6.6. If using propellers, the platform to be used does not have these cutouts, this design is shown in the drawing in the Appendix 13.6.10.

7.2.2. The Acrylic Piece

The acrylic securing piece was created as a means to attach the internal components of the design to the platform and sits just on top of the platform. It has the same basic design as the platform but only a 0.5 in thickness. Acrylic was decided to be relatively low cost, easy to find, easy to shape, and relatively low weight. The acrylic piece is attached to the platform via screws. This piece was used in order to fasten the internal components securely to the platform, Figure 29 depicts the many holes used for attaching these internal components. These internal components include the sensor deployment mechanism, the waterproof electrical box, the solar charger controller, and the battery.

Another function of the acrylic piece was for waterproofing. Around the edge of the acrylic is a small indented grove that will be used to hold a flexible material for sealing. Ideally, stock O-ring material would be adhered to this groove and would act as a seal when the cover is placed on top of it. The four holes on the outside of this groove are tapped holes made to secure the cover to the acrylic piece and to maintain the sealing. Figure 29 shows the acrylic piece used when using hydro pumps, as it has an additional two holes for the hydro pumps wires. Appendix 13.6.7 contains the drawing of this piece.

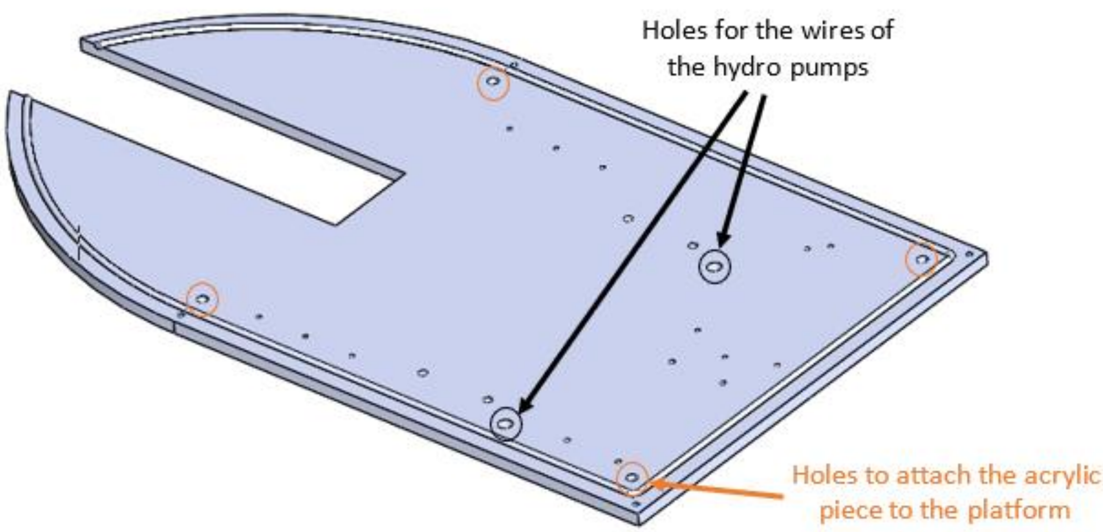


Figure (29). *Acrylic piece when using hydro pumps as the propulsion method.*

In the case of using propellers as the propulsion method of the platforms, the acrylic piece should have different features in order to properly mount the propeller system, as shown in the drawing in Appendix 13.6.11.

7.2.3. The Cover

The cover piece was proposed in order to protect the main design components on top of the platform from the water. The cover was created on CAD with the machining technique, vacuum thermoforming, in mind. With some research, it was determined that this technique would work well with 0.04 in thick clear plastic, PETG. Depending on the machine used for this technique, a different material or material thickness may be chosen. With vacuum thermoforming, a sheet of thermoplastic is heated and then suctioned into a mold to form the shape. There is a 15 degree draft angle on all sides of the cover to ensure that the cover piece forms well.

An important component for vacuum thermoforming is the mold. The mold is created in order for the thermoforming technique to make the desired shape. The mold is placed on the vacuum forming machine and when the vacuum is turned on, the plastic will be suctioned into the mold to form this shape. In order to produce a well fit cover, a female mold was proposed. The female-type mold will form the cover upside down, with the top of the cover suctioned into the bottom of the mold and the sides shaped last. Figure 30 shows an example of a male and female mold. Figure 31 shows the CAD model of the mold used to make the cover. Appendix 13.6.4 shows the CAD drawing for the mold created for this cover design.

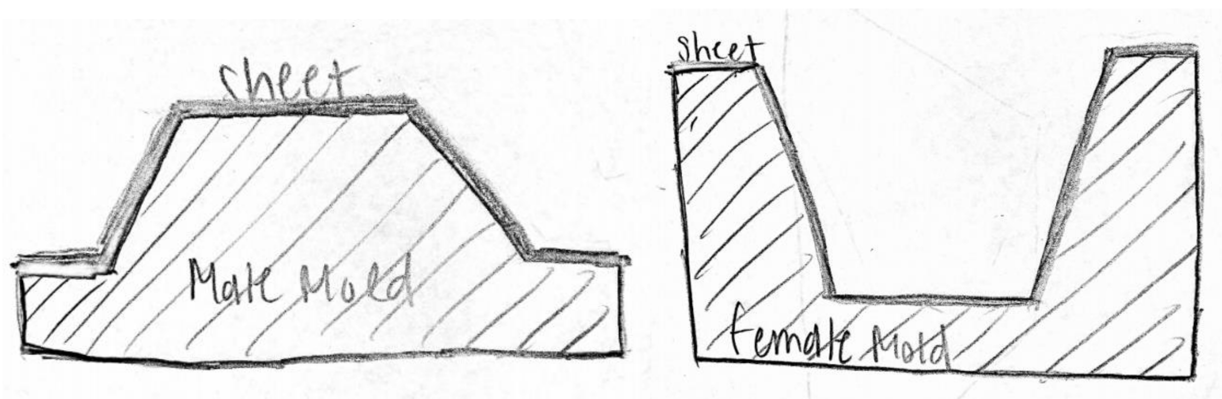


Figure (30). *Male mold (left) vs. female mold (right) used in vacuum thermoforming.*

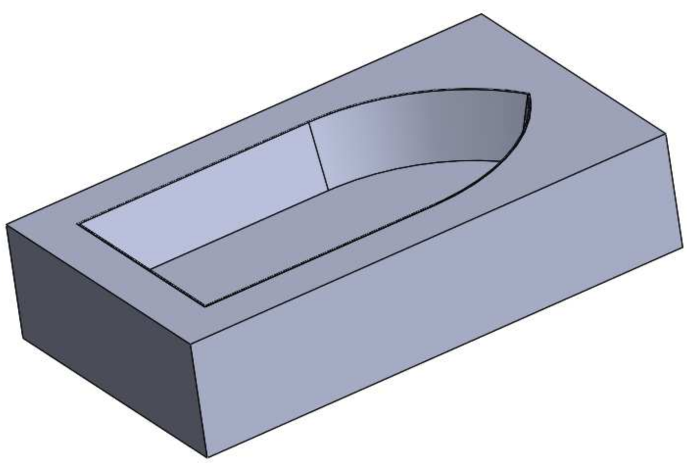


Figure (31). *Female mold used to make the cover.*

As seen in Figure 32, the cover has two divided sections within it. This partition must be added after the thermoforming technique. The front portion of the cover is where the sensor, sensor deployment mechanism, and electrical box are housed. In this portion, there is a small hole in the front where the sensor can be deployed from but is otherwise sealed. In the back portion of the cover there are small slits cut out. These were made in order to allow for cooling of the hot components of the device such as the large battery that powers the propulsion system. These holes were positioned towards the top of the cover to try to mitigate any splashing water from getting inside the cover. In the top of the cover, there is a small hole created for the wiring of the solar panel to the internal components. Appendix 13.6.3 shows the full drawing of the CAD cover.

Additionally, the cover has a waterproof seal. As mentioned in the acrylic securing piece section, a flexible tubing will be squished between the cover and acrylic piece when the cover is screwed into the acrylic to form a watertight seal.

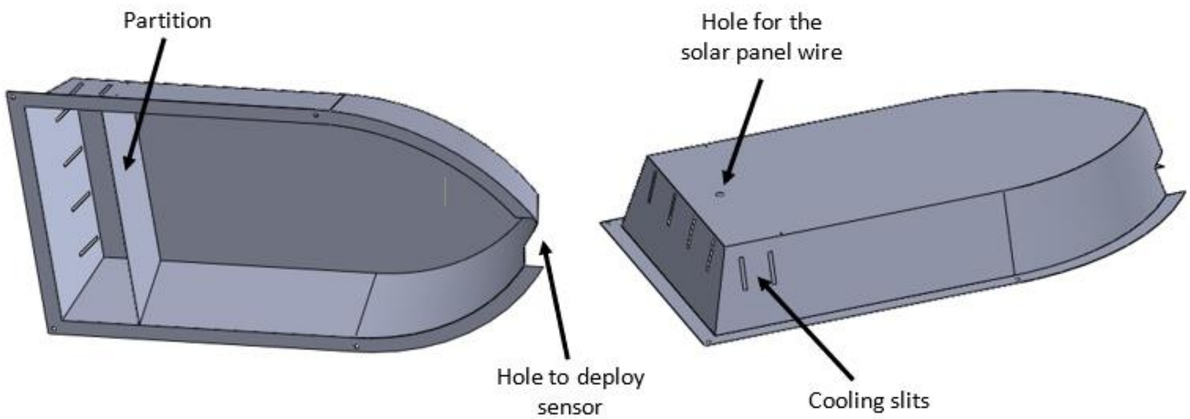


Figure (32). *Different views of the cover.*

7.3. Mechanical Subsystems

7.3.1. Sensor Deployment Mechanism

7.3.1.1. The Sensor

Figure (33). *Thermocouple sensor vs. its CAD model.*

The sensor depicted on the left in Figure 33 is the temperature sensor (thermocouple) that was given as the sensor to use in the deployment mechanism by the customer. The sensor is 13.5 in long and has a diameter of 1.8 in, it also weighs 1.26 lbs. In order to ensure proper waterproofing of the wire connection at the end of the sensor, a casing was made. This casing is depicted in Figure 34. This casing surrounds the base of the sensor and cable and uses screws to create an airtight closure. The weight of the casing also weighs down the sensor so that it will sink more when submerged into water.

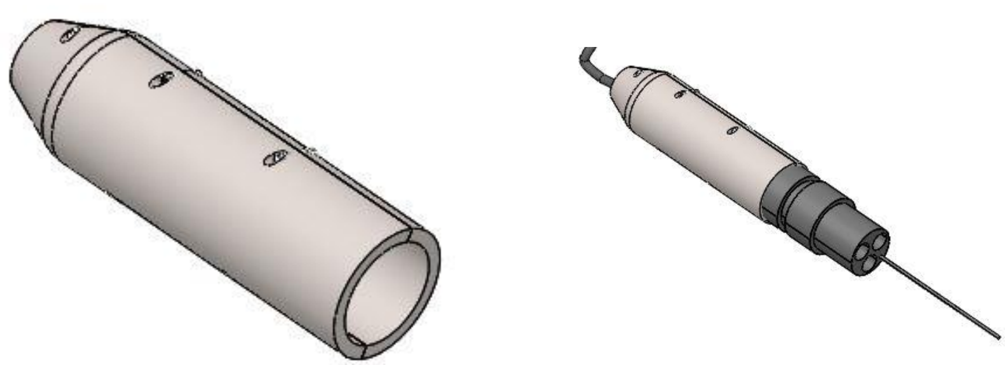


Figure (34). *CAD model of thermocouple casing.*

7.3.1.2. Iteration 1: Testing Model

To optimize the first model used for feasibility demonstration, the conceptual design was simplified to only have one motor controlling the rotation of the threaded spool. Whereas the feasibility design aimed to have the cable release and reel in at the same location, it was found that the cable displaced less than an inch horizontally as it was reeled back into the spool. Therefore, for 5 ft of cable and a spool length of 4.5 in, the horizontal motion of the spool was determined to be unnecessary. This test model, as shown in the figure below (Figure 35), allows only for rotation of the threaded spool in order to wind and unwind the cable without the cable overlapping or tangling on the spool. Figure 36 shows another angle of the testing model, with the wires leading out of the bracket in full view.

In order to choose the appropriate motor for the mechanism, it was known that the motor would need to support at least three pounds as well as have a high torque at low speeds. A DC motor replaced the previous stepper motor because in a DC motor the internal friction locks the position once it is deployed to a certain depth and no more power is running through, meaning that the cable will not slip further down. A stepper motor would need constant power to hold the position and would overheat with larger loads.

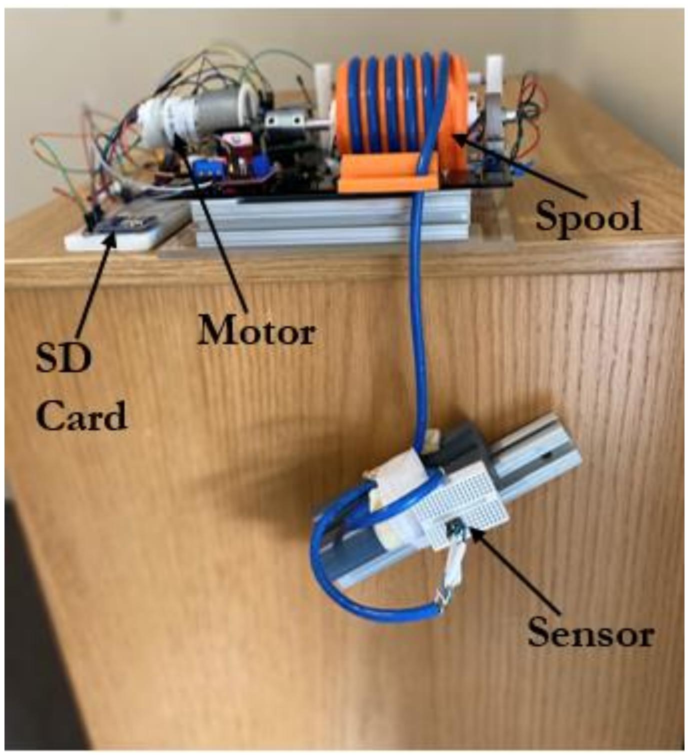


Figure (35). *Iteration 1 of the sensor deployment testing model.*

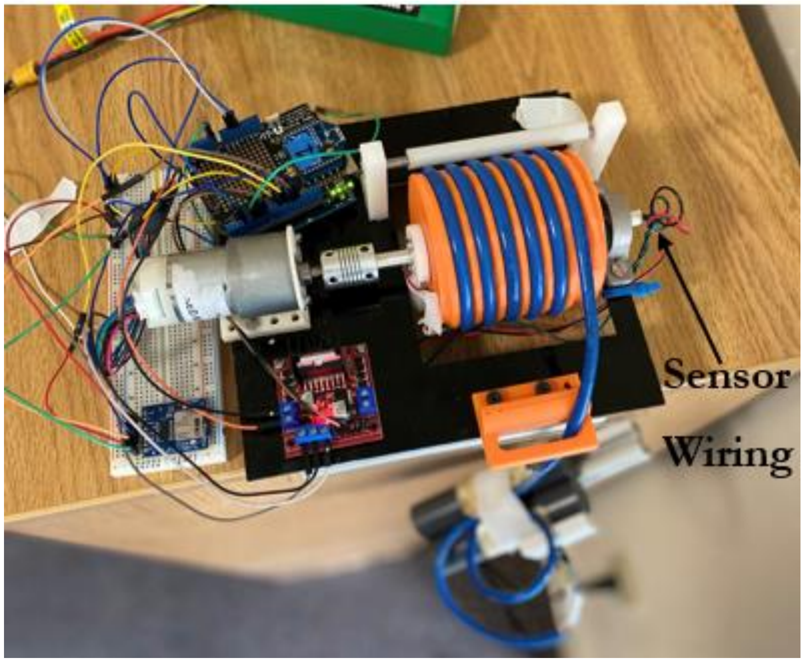


Figure (36). *Top view of Iteration 1 of the sensor deployment testing model.*

The construction of this model used parts that could be easily replaced, such as the acrylic and aluminum extrusions, to be able to test different configurations of overall height and spool dimensions without having to re-design the overall mechanism. In addition, many components, such as the threaded spool, were manufactured using 3D printers and therefore reduced cost and allowed for more versatility of the design.

An addition to this design that was not present for the feasibility design is the inclusion of cabling from the sensor to the Arduino board located outside of the mechanism. While alternatives such as winding excess cabling leading from the threaded spool to a custom-made pulley were explored, the decision for the wiring to go through the threaded spool into the main aluminum shaft was chosen due to its compactness, simplicity, and ease of interchanging different cables. Since this model was designated for testing, the wire routing was specific to this cable however the next iteration solved the issue of interchanging different cables onto the threaded spool.

Another design decision that arose from testing was to add a roller on the side of the spool in order to counteract the cable unspooling when submerged in water. For testing purposes, a weight was added in place of the sensor; fortunately, it was found that adding the extra weight solved the problem of the cable unwinding itself when it reached a depth of four feet. The testing videos are found in this following link: Sensor Deployment Testing Videos . With the roller in place, the cable did not unspool from its grooves as the spool rotated.

7.3.1.3. Iteration 2: CAD Model

The design changes that were made to the testing model were to improve the deployment mechanism aesthetically as well as to create a method to have interchangeable cables without having to change the wiring that leads to the Arduino controller. Below is a 3 View CAD model of the final deployment mechanism (Figure 37):

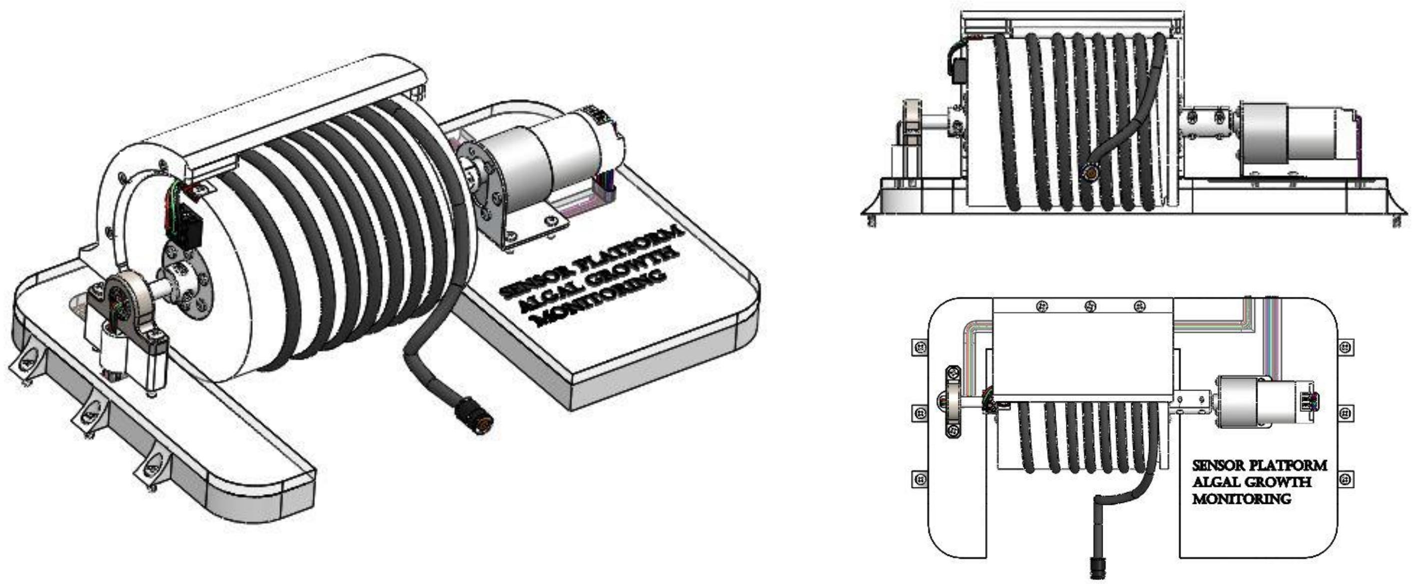


Figure (37). *3-view CAD model of Iteration 2 of the sensor deployment mechanism.*

Compared to the first iteration, several components had been re-sized such as the threaded spool, while other components such as the aluminum extrusions were taken out completely. The threaded spool had minor changes including increasing the cable length on the spool so that five feet of cable are fully submerged in the water when the second thread of the cable is off of the spool, as compared to the first design where the full length of the cable had to leave the spool in order to unwind five feet. This added length would lessen the strain of the weight on the end of the cable that is fixed to the spool, so the tension does not tug the cable out of the wire clip.

One structural issue that arose with the first iteration was bending due to the thin acrylic used to hold the deployment and so the aluminum extrusion was replaced with 0.4 in thick HDPE to prevent the acrylic bending. However, if it is desired to change the height of the mechanism, the HPDE can be switched out for something of a different thickness. This allows for simple interchangeability because all the components are fastened to the acrylic and not the HDPE. Interchangeability is necessary to account for because this deployment mechanism would ideally function for a variety of sensors of different sizes.

In the figure below (Figure 38), another 3 View is portrayed to highlight the wiring of the sensor connections from the cable end to the wire clip, terminal block and finally to the Arduino:

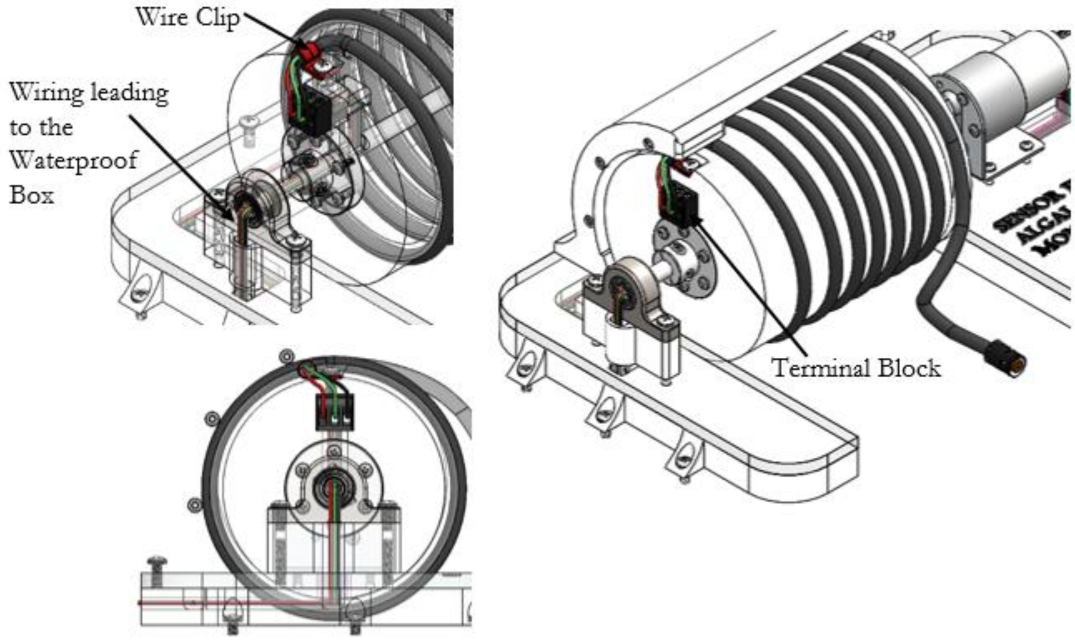


Figure (38). *3-view of the sensor deployment wiring scheme.*

Figure 38 shows that the wire clip fastens to the cable by gripping it using the tension of the screw that is threaded into the threaded spool. The wire clip is also permanently attached to the spool. This allows the user to use interchangeable cabling by simply unfastening the screw, placing a new cable, and fasting the screw again. From the wire clip, two or three of the inner wires will be directed into the black terminal block. The other end of the terminal block will have three permanent wires that will be routed through a hole in the spool into the aluminum shaft and outwards past the bearing to finally lead into the Arduino. This is shown in Figure 38 in the bottom left image. These permanent wires will not need to be switched when a different sensor is used because they are merely connectors to simplify wiring. The only function that the user performs is screwing the cable into the wire clip and the three inner cables into the terminal block.

7.3.2. Propulsion System Mechanics

7.3.2.1. Hydro Pumps

The primary propulsion method that the team started with were the hydro pumps. Although hydro pumps are not traditionally used as a propulsion method, they were readily available for preliminary testing to use for both thrust and directional control.

As tested in the conceptual design of the remote-controlled system, turning on both motors will propel the mobile platform forward and when one hydro pump is turned on and the other is off, a right or left movement can be achieved. This method was simpler than adding a rudder for direction control. However, this method does not allow for fine directional control as this depends on the delay of the control signal. Additionally, the hydro pumps were previously fitted with a nozzle to focus motion and increase thrust.

Figure 39 shows the hydro pumps the team used. These hydro pumps were electric bilge pumps, a type of centrifugal pump. Centrifugal pumps are used to induce flow or to raise a liquid from a lower to a higher level. It uses an impeller to move fluids, in this case water, by using centrifugal force. When the impeller rotates, the rotational energy provided by the motor is converted into energy in moving a fluid. They are generally designed for relatively low viscosity liquids, such as water. In order for the hydro pumps to work properly, it is important that they are completely immersed in water [PZE_20]. These hydro pumps were selected due to the fact that they were waterproof, easily available, relatively low cost, and robust.

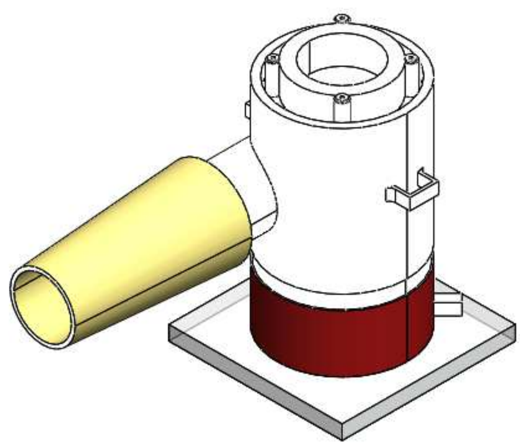
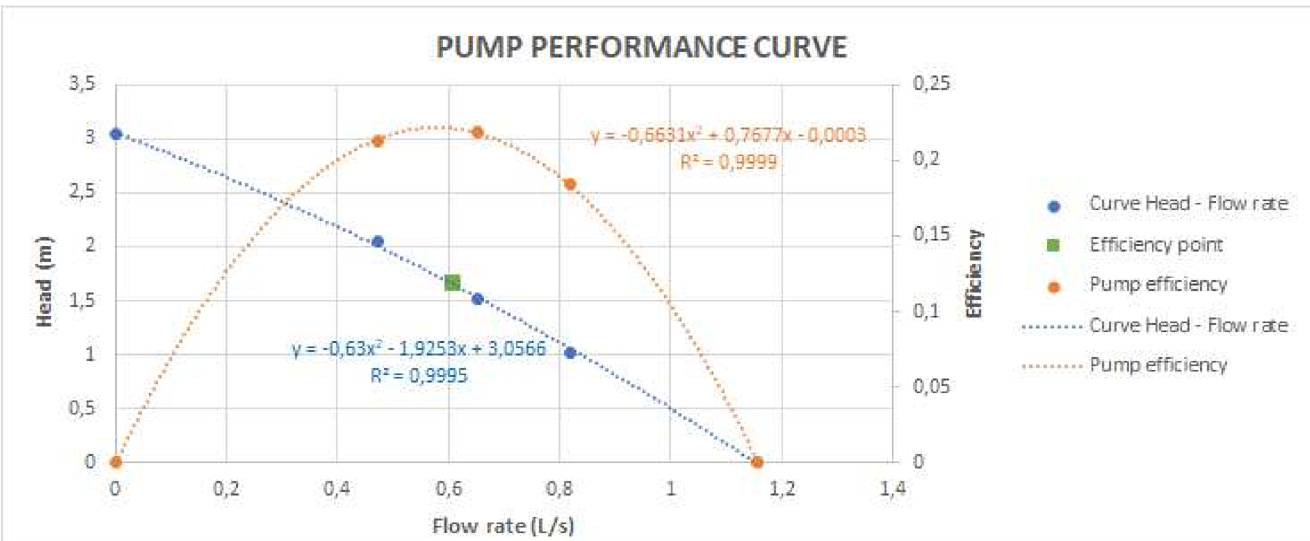


Figure (39). *CAD model of the hydro pumps used as the primary propulsion system.*

An important goal for the team was to estimate the speed the platform could reach by using these hydro pumps as the propulsion system. This speed would help determine whether or not these hydro pumps could achieve the engineering specification of minimum distance traveled per day. Further, these speeds would be incorporated in the power study (Section 8.2) to test whether the battery could power the hydro pumps/propellers for the 6 hours minimum activity requirement without recharging. For this, the team first had to do some research of the level of performance of these specific hydro pumps [ZORO20]. A pump's performance curve is essentially data about its ability to produce flow against a certain head pressure [AHMA19]. The performance curve for this type of hydro pump is shown in the following graph, with its respective efficiency curve (Graph 1).



Graph (1). *Pump's performance curve (Head pressure vs. Flow rate), with its respective efficiency curve (Efficiency vs. Flow Rate).*

In the previous graph, the blue curve represents the head pressure of the hydro pump with respect to the flow rate coming out of the discharge nozzle, whereas the orange line represents the efficiency of the hydro pump according to the different flow rates. The green point represents the point in which the hydro pump acts in its most efficient way, that is, with a flow rate of around 0.6 L/s. The calculations of how both curves were obtained are shown in Appendix 13.3.1.

In the design, the hydro pumps are fitted to the back and bottom of the platform (Figure 40). The blue arrows indicate where the water is sucked in by the impeller, whereas the green arrows indicate where it exits.

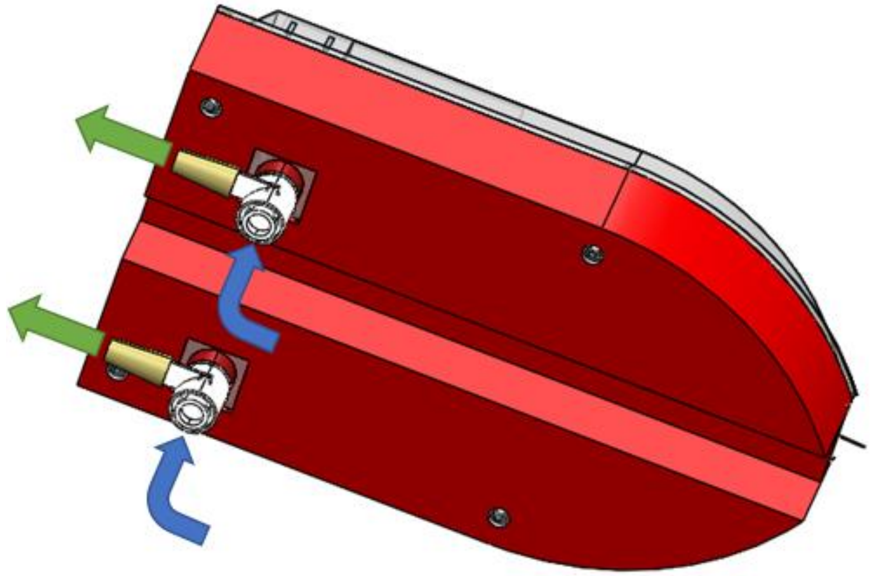


Figure (40). *Hydro pumps attached to the bottom of the hulls of the platform.*

7.3.2.2. Propellers

Although the hydro pumps do propel the platform, they are not the most efficient way of propelling it. A propeller is essentially a spiral screw with multiple leads. These leads correspond to each propeller blade. The propeller takes rotational energy and converts it to linear thrust. For each turn of the propeller blade, the platform will move the "height" of the propeller minus some slip due to drag. Figure 41 depicts the propeller used in the final CAD model of the full mobile platform taken from GrabCAD [GCC_19]. This propeller is very similar to the propeller found online and used for the calculations, but not identical.

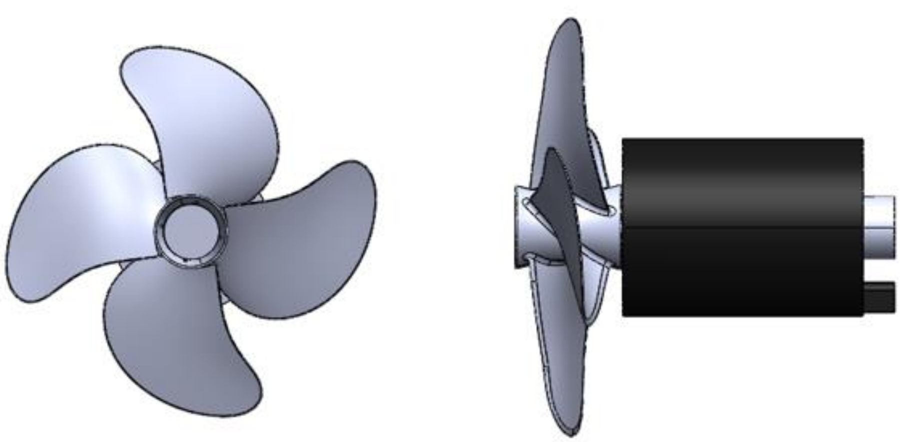


Figure (41). *Sample propeller and motor system CAD.*

Most boats use propellers and as found in research, propellers are efficient at low speeds. The propeller used for calculations has a 0.06 meter diameter and 0.0135 meter height. Like the hydro pumps, the propellers are mounted to the back and bottom of the mobile platform and take power from the renewably charged battery. Figure 42 shows a CAD model of the propeller mount used to attach the propellers to the platform. A complete drawing of the propeller mount is shown in the Appendix 13.6.9.

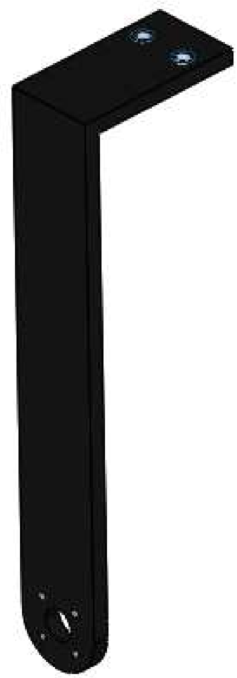


Figure (42). *CAD propeller mount.*

The full propeller assembly includes the two propellers, each with its own motor, propeller mounts, and fastening screws. This can be seen in Figure 43 and in more detail in Figure 44. The propeller mount allows the propeller to be placed at the bottom of the platform and thus fully submerged in the water. The complete assembly of the platform using propellers as the propulsion method is shown in the Appendix 13.6.8.

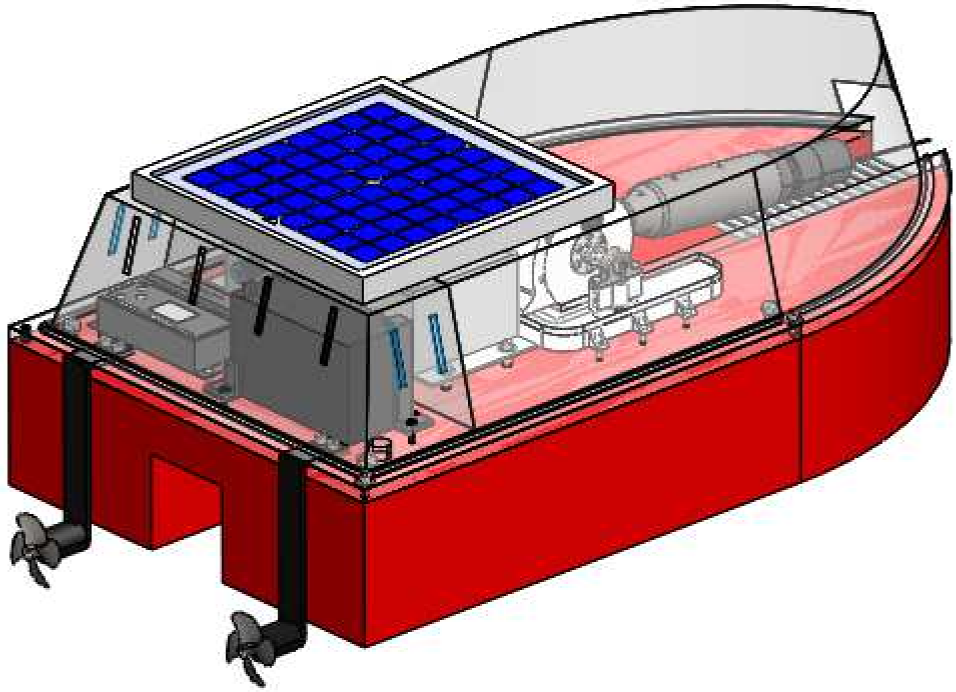


Figure (43). *Full assembly with propellers as the propulsion system.*

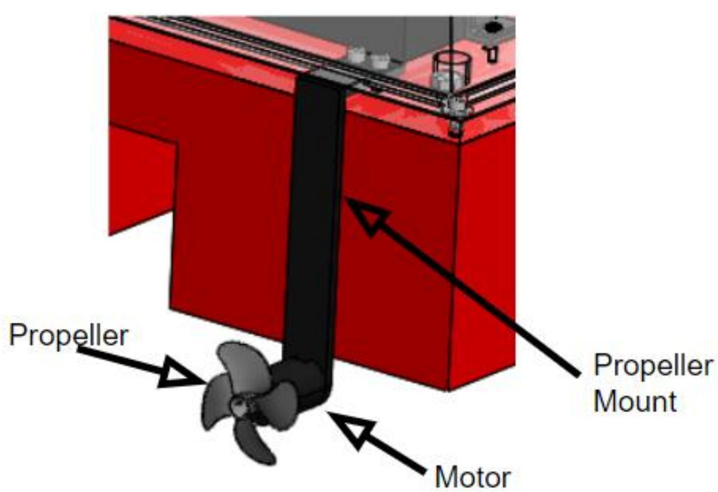


Figure (44). *Propellers close up.*

7.4. Electronics Subsystem

The electronic subsystem of the mobile platform is controlled using a user-operated remote controller, pictured in Figure 45. The system is capable of processing signals bi-directionally via wireless transceivers and instrumenting the manipulation of these signals in order to control either the platform propulsion, or the data collection and transmission of the onboard sensor.

This remote-control system is accomplished using the XBee Pro S1 wireless transceivers in conjunction with an Arduino. Full electrical schematics are available in the Appendix 13.4, as well as the Arduino codes for both the controller and the platform in Appendix 13.5. The figure below (Figure 45), is a prototype Controller Board modeled in SolidWorks CAD that contains all the electrical components used for wireless control, and was designed to be constructed from machined HDPE plastic and acrylic for the back of the Controller.

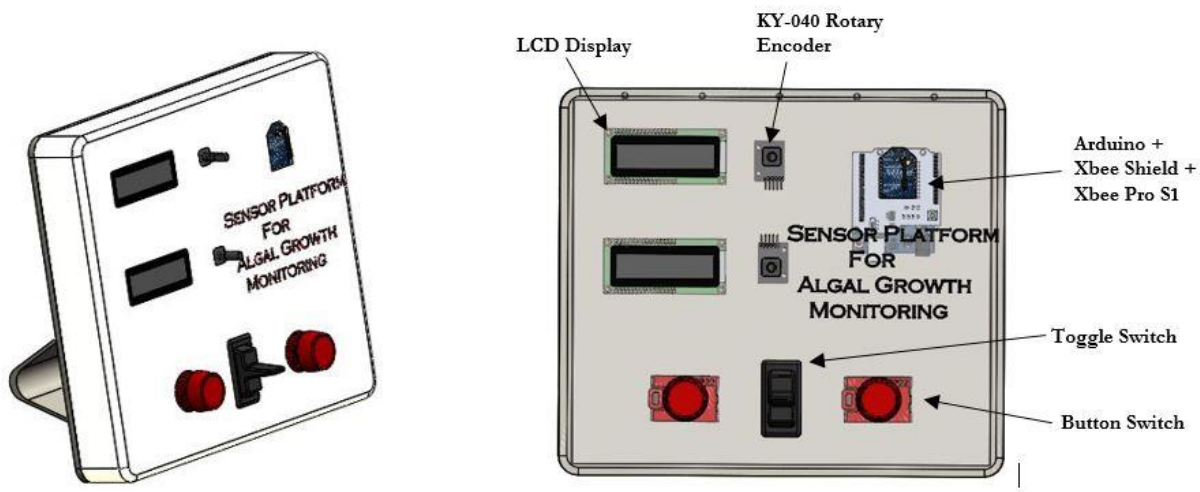


Figure (45). *Prototype Controller Board for remote control.*

7.4.1. Wireless Communication

The XBee Pro S1 modules are a product that supports wireless communication protocol, and in the electronics design they use the standard protocol ZigBee for wireless networking. These RF modules are capable of sending wireless commands through a serial communication protocol that can be readily implemented in Arduino. Their performance varies based on the transmitting power, orientation of the transmitter and receiver, and interferences in the area among other factors. According to the data sheet, these modules should ideally be able to transmit data up to a mile in best conditions i.e. low noise areas and a direct line of sight. In Section 8.4, evaluations of range testing are summarized.

Another module that was considered to be an alternative to the XBee Pro transceivers were the Lora Reyax Rylr896 modules. These modules function in the same way as the XBee Pro modules, and the same serial communication protocols are used. These modules were chosen to be compared due to their claim of a very long range of data transmission, over two and a half kilometers. These modules are evaluated in Section 8.4.

To meet the requirement of transmitting data to the controller, a bi-directional point to point communication was developed using serial communication functions in Arduino. Serial communication is a transfer of information, stored in bytes, from one serial device to another. On the Arduino UNO Board, the hardware serial is used to print these data bytes onto a computer serial port, while the software serial is used to send data to another serial device with the same initialized serial pins. For this design, the serial devices are the XBee Pro modules who transmit serial data to each other so long as the designated Receive and Transmit pins are the same as the software serial pins. The figure below (Figure 46) demonstrates this use of serial communication to send data over two devices.

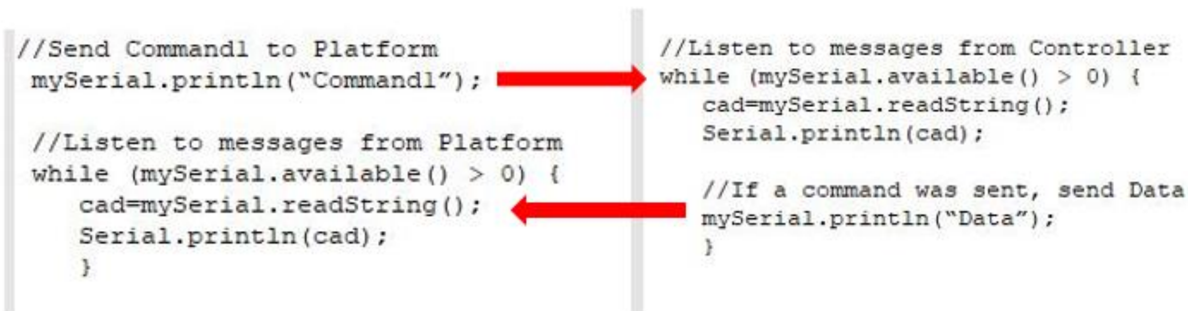


Figure (46). *Example schematic of serial communication between modules.*

Reliable two-way communication is important for the platform because it allows for direct feedback to ensure that the signal was properly received as well as instrumented. It is the case, especially with the XBee's, that the farther the two modules are the more likely that not every signal will be received by the other module. In fact, when rapidly sending commands to one of the modules, the serial communication cannot read the data as fast as it is receiving it and therefore some commands will be lost. This is due to the limited speed at which the module can process incoming data and the limited amount of memory allocated to store incoming data. Although this can be accounted for in the code by designating intervals in which serial communication can be received, it was determined that these delays also cause issues in bi-directional serial control. Figure 47 represents a schematic of the information sent between the controller and platform.

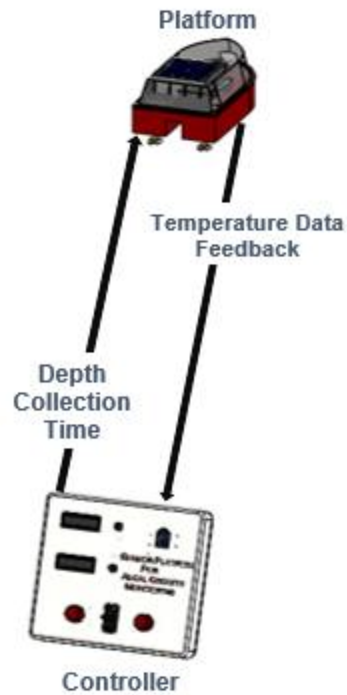


Figure (47). *Information sent between the controller and the platform.*

To combat this, a feedback system will ensure that every signal sent from the controller to the platform will trigger a response back from the platform, depending on what command was sent. The platform will then send updates in intervals of ten seconds as it is instrumenting the task.

For example, if the user sends a command to deploy the temperature sensor to a depth of three feet, and record data for five minutes, the platform will send one signal indicating the cable is deploying, then another signal when the temperature sensor begins recording data. Finally, for the amount of data storing time, the platform will transmit the temperature every thirty seconds. This interval is adjustable in the hardwired code, and can be at a minimum of two and a half seconds. In the case that the signal to the platform is lost, the platform will not be able to transmit any response and the controller serial will time-out, indicating to the user that they must check the signal transmission.

7.4.2. Propulsion Control

Referring to the feasibility demonstration of the remote-control system, the propulsion was controlled with two buttons and depending on how they were pressed the platform would move forward or turn. However, when tested in a small lake, the buttons had to be continuously pressed and this continuous transfer of signal resulted in signal loss, creating unpredictable platform maneuvers. This is portrayed in testing video at the following link : Hydro pumps testing videos.

To improve upon these controls a toggle switch was added so that the upper position turned both hydro pumps on while the lower position turned both hydro pumps off. To control direction two buttons were added however a single click for each button would send a command to turn one hydro pump off until that button was clicked again. This solution, although not able to be tested due to social distancing restrictions, would circumvent the need to continuously send over the serial monitor and limit instances of signal loss. For the addition of propellers, where speed would be controlled, a rotary dial would be used to control the speed where 100% speed would equate to full voltage sent to the propellers.

7.4.3. Data Collection and Transmission

Another one of the aforementioned rotary dials plays an important, multi-functional part in the sensor deployment mechanism. The rotary dial is responsible for all the user input by controlling an LCD menu where the user chooses options and parameters based on the encoder position and button press. However, it should be noted that these KY-040 rotary encoder dials are among the cheaper electronics that are widely available, and due to this an instrumentation error called switch debouncing occurs. A switch debounce occurs due to the physical parts of the sensor itself causing it to register a single tick mark as multiple ticks. This is due to the state change of the CLK and DATA pins on the encoder being measured from changes in metal contacts inside the encoder. Ideally the contact would be instantaneous resulting in one electrical transition however the non-ideal behavior of the metal contacts causes multiple transitions to be read from a single tick. Therefore, when the user turns the dial one tick, the Arduino will detect two or more state changes. Complications that can arise from debounce are unreliable data outputs when rotating the dial at a moderate pace and the moderate occurrences when the switch detects an incorrect direction change.

Although there are more sophisticated instruments available, or alternatives such as potentiometers, an elegant solution consists of using Arduino interrupt pins along with a written debounce function in order to interpret the rotations. Using this approach, the rotary dial accurately measures the tick marks even when rotating the dial very quickly. Therefore, it can reliably be used to choose the depth that the sensor will deploy to without the possibility of an incorrect reading. The decision to use this debounce function instead of using another more reliable button or switch was made because this dial is very user-friendly, goes 360 degrees, and the use of ticks lets the user choose exact values by setting increments for each tick mark. The switch-debounce function reads the raw data from the rotary encoder pins, and only looks for the correct pattern of binary data. For example, before using the switch debounce, a clockwise rotation of two tick marks would be (01 10 11 01 00 10). After implementing the function, the correct output would be (01 01).

The user interface on the controller for the sensor deployment consists of only this rotary encoder and the LCD display, as seen in Figure 48. On this interface the user can select either an *automatic* option that measures the temperature at a pre-written, hardwired pattern, or a *manual* option in which the user rotates the dial to choose a depth and sends that command when pressing the push button. Another option to view the temperature at a given depth but not collect any data is the *Sweep Temperature* option. The other two options are *Test Signal*, which tests the wireless transmission and *Home*, which brings the sensor to the initial position on the platform. The platform will then interpret the commands sent by the controller and deploy the sensor to that depth, as well as store temperature data into the SD card for the specified amount of time. The SD card is capable of storing up to 7 GB of storage, and is retrieved when the boat is taken out of the testing environment. While the temperature data is stored onto the SD card at a rate of one measurement per millisecond, the temperature at every thirty seconds will be transmitted back to the controller. This time interval is hardwired into the Arduino program and can be lowered to two and a half seconds, as previously mentioned. Figure 49 describes each menu option in detail and the receiving information resulting from these commands.

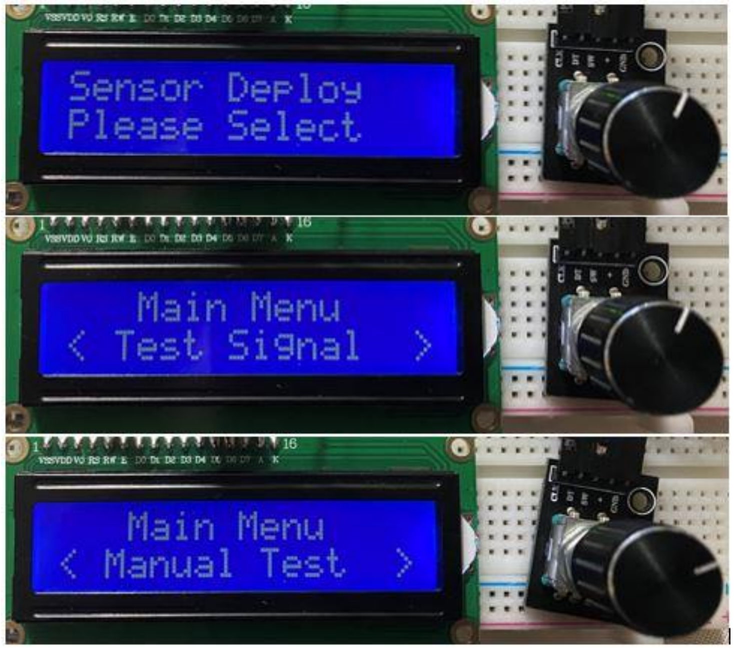


Figure (48). *Examples of the LCD Display on the Controller.*

Another feature on the LCD displays will update the position of the platform when called upon. However, it is important to note that the GPS used for testing, was a low-cost, low-functioning device. It will only transmit data when it fixes onto a satellite, which could take up to three minutes and is dependent upon a clear view to the sky with no obstructions or clouds. This feature is displayed on the LCD for the propulsion control.

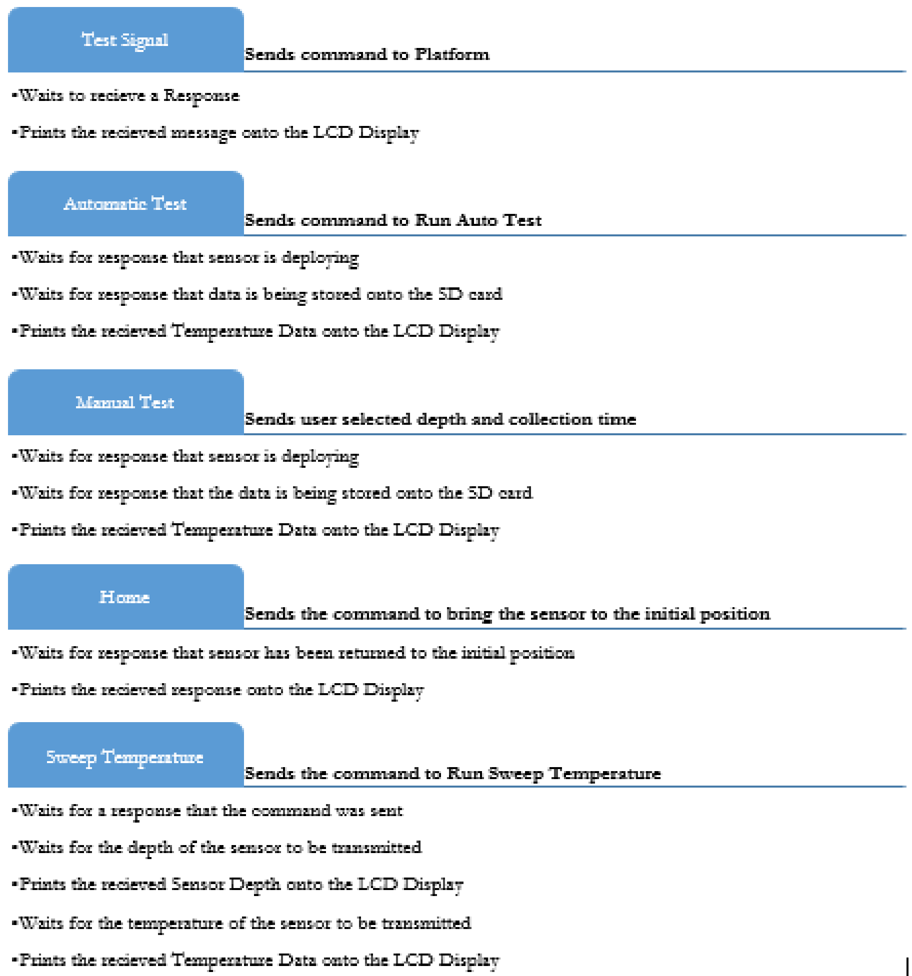


Figure (49). *Schematic of the functions for each of the LCD Menu Display Options.*

Another component of the Arduino platform code (Appendix 13.5) is the operation used to maintain the sensor deployment motor at a speed of 15 RPM. Using an attached quadrature encoder on the motor itself, the RPM is measured by counting the "pulses" and measuring the time the motor has run. These pulses represent how many times the motor spins through a magnetic Hall Effect sensor that comprises the quadrature encoder. The motor in the design has seven pulses in one rotation, and so dividing the total amount of pulses by seven will give the rotations of the motor. To measure these pulses, an interrupt routine is included in the code. Whenever the Arduino detects a change on the two pins of the quadrature encoder, the interrupt routine will ensure the Arduino responds quickly to the time sensitive state changes.

The RPM is then read into a PID (Proportional, Integral, and Derivative) function that calculates the output voltage necessary to have the motor run at a target speed of 13 RPM. With no load on the motor, this RPM corresponds to a voltage output of 10 V, and an Arduino PWM signal of 170. While the power supply is 15 V, the max speed the motor can run is 21 RPM while the minimum is 7 Volts for a speed of 9 RPM or PWM of 130. Therefore, the PID will begin with a PWM of 170 and increase or decrease this value as the sensor unwinds into the water, in order to maintain a speed near 13 RPM. With this protocol in place and taking into account the low speed of the motor, the time it would take for the motor to reach a specified depth could be calculated using the diameter of the threaded spool.

All of these components are located in a waterproof box (Figure 50) that would be attached to the acrylic securing piece. Whereas the four main gasket holes are reserved for wiring to the propulsion, extra drilled holes would have to be made in order to guide the wires from the sensor deployment into this waterproof box.

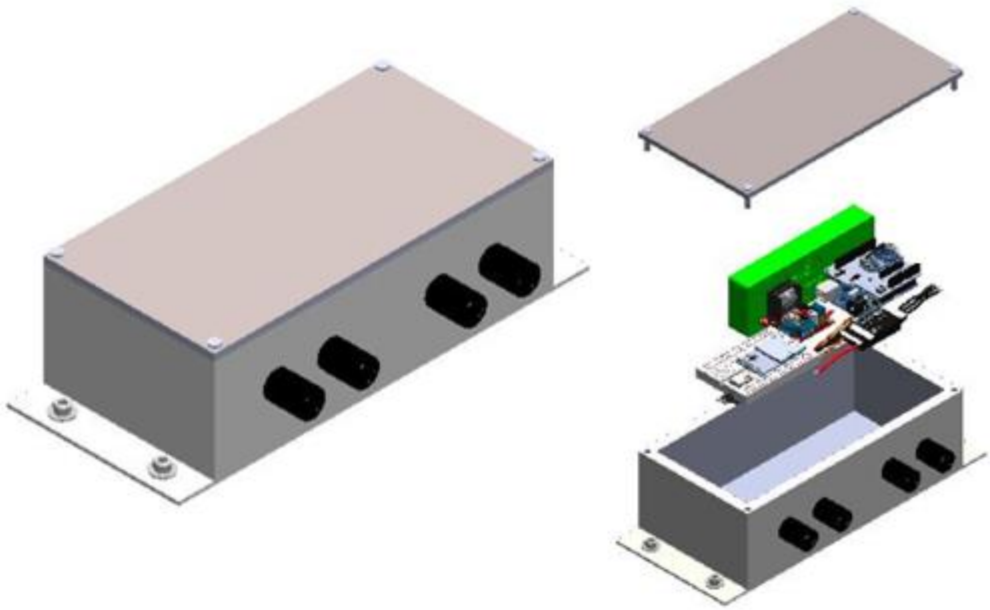


Figure (50). *Exploded view of the waterproof box used to house the electronics.*

7.5. Power Subsystem

The platform's power system consists of an ACOPOWER 10 Watt monocrystalline solar panel, PowMr solar controller, Battery Mart lead-acid battery, Adafruit XBee, Adafruit XBee shield, Arduino Relay shield, Arduino, and two motors as shown below (Figure 51):

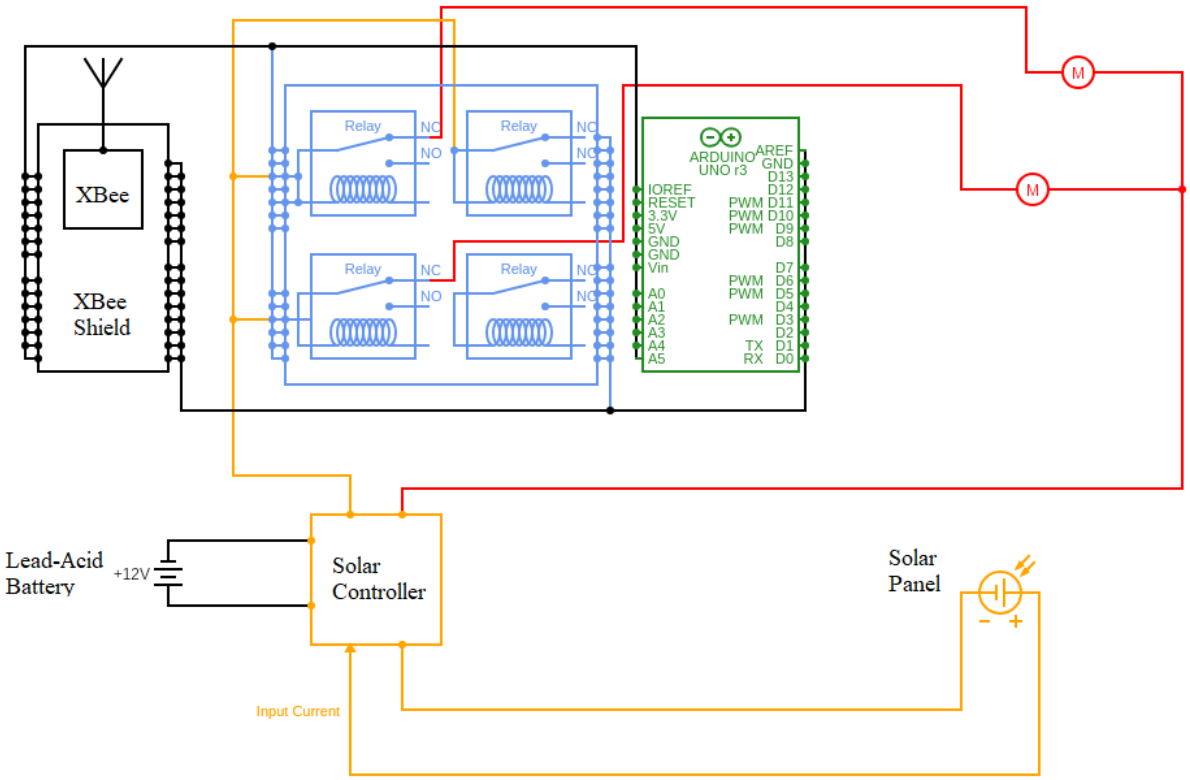


Figure (51). *Circuit diagram of power subsystem.*

The lead-acid battery is the main driver of the circuit while the platform is in operation. The battery supplies power to the propulsion system, which consists of each hydro pump being connected in parallel with respect to each other. The Arduino is supplied with 5 V from a separate battery within the platform's waterproof box and is responsible for controlling the Relays via code. Each hydro pump is connected to its respective Relay switch—doing this enables steering control of the platform's motion (Figure 52). All microcontroller communications happen between the XBees on the platform and the handheld controller.

At 12 V, both hydro pumps draw 7.04 A (3.52 A for each pump), so the battery will fully discharge after 58.3 min without having the solar panel for recharging. This was calculated using Peukert's Law (included in the Appendix 13.3.5). This calculation was done with the assumption that this lead-acid battery has a Peukert constant of 1.1. It was also assumed that each hydro pump will draw the same amount of current and that all other microcontroller currents were negligible.

When the battery begins discharging, its voltage will decrease. Currently, when the battery's voltage reaches 10.8V, the solar controller will begin charging the battery with current coming from the solar panel. The recharge voltage can be adjusted from within the controller settings. In its current setting, while the battery is being charged, the controller will cut power from the load in order to protect the electronics from any spontaneous power surges. This solar controller has three charge phases: bulk, absorption, and float. During the bulk phase, the battery's voltage is increased to the bulk voltage, and the battery draws the maximum current. Next, the bulk phase is maintained while the battery charges, and the current begins to decrease until charging is complete. Finally, during the float phase, the battery's voltage is brought down to normal operating levels until the next charge cycle. Battery charge settings can also be adjusted, and each setting affects the battery differently [JOHN__].

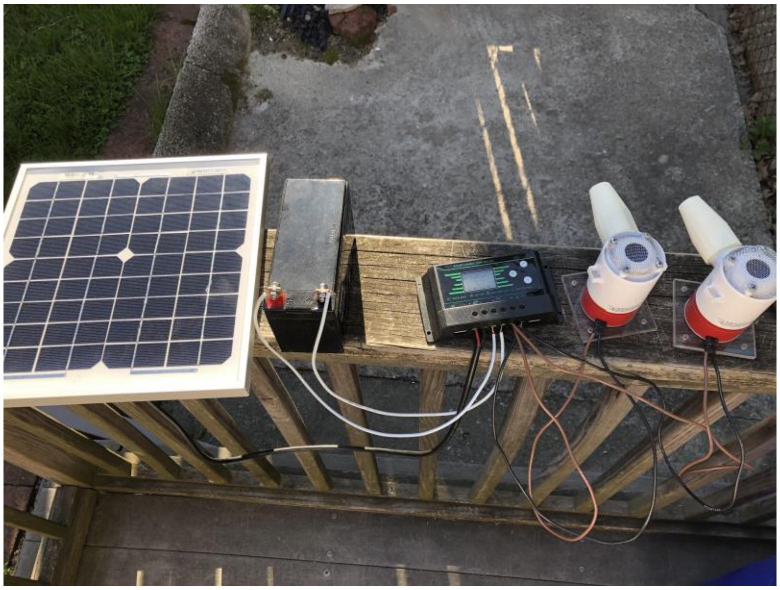


Figure (52). *Solar circuit outside of the platform.*

Since the platform was designed with sustainability in mind, the solar panel within the circuit allows the platform to utilize renewable energy. Unfortunately, the sun does not shine every day, so this limits the platform's usability to sunny days.

The solar controller was added to control when the battery will draw power to recharge from the solar panel and to protect the battery from overcharging. The solar controller acts as a medium between solar power and battery power. To illustrate, the solar controller allows the platform to operate even on overcast days by using the stored energy in the battery. Additionally, the solar controller allows the hydro pumps to pull power from the battery while simultaneously allowing the solar panel to charge the battery. Once the battery falls below 10.8 volts the solar controller will allow the solar panel to supply charge to the battery. The controller also will stop the flow of power from the panel to the battery once the battery is at full capacity to protect from overcharging. Because of these features, the circuit's solar controller is the most valuable component.

The initial platform circuit consisted of each hydro pump being in series with its own battery. In the current circuit design, the hydro pumps are connected in parallel to each other. This will allow the hydro pumps to draw the same amount of current, which was a crucial assumption when making power study calculations (seen in Appendix 13.3.5). They are arranged this way so that each one can have the same voltage input. This ensures that each hydro pump consumes an equal amount of power from the battery while allowing them to operate independently. This also eliminates the need for a second battery, so the latest circuit configuration is more space-efficient. The solar panel is able to recharge the battery quicker than power discharge of the battery from the pumps. Therefore, as long as there is sunlight the battery would work endlessly (Appendix 13.3.6).

8. Prototype Evaluations and Results

The main parameters that will be used to evaluate the final design created are the following. The mobile platform must travel a minimum of 2 mi/day and be able to run continuously for 6 hours following a typical run time of the platform in a day. Additionally, the platform must house and deploy the sensor to different depths up to 5 ft. Finally, as the sensor records the relevant data, this data must wirelessly transmit to a location at least 300 m away. The system should also be able to record at least 1 rdg/5 mins. These parameters were determined by evaluating the customer's expectations for the project.

8.1. Pump Current Testing

In order to make proper battery life and platform run time estimations, the hydro pumps needed to be tested to measure their current draw. For this test, one hydro pump was submerged and powered with a DC generator. The voltage and current were then slowly increased to find the minimum operating conditions and the conditions at the battery voltage (12.1 volts). The results are as follows (Table 7):

Voltage (V)	**Current (A)**
2.1	0.48
12.1	3.52

Table (7). *Voltage and current of the hydro pump.*

Because the hydro pumps will be connected in parallel, the currents used in calculations are double the measured values.

8.2. Power Subsystem Study

The power requirement set for the platform was that the platform must have a minimum run time of 6 hours. Further, the propulsion system must handle a minimum distance traveled of 2 mi/day. In order to achieve this propulsion requirement, both the propulsion system and power subsystem must work together. With this in mind, SolidWorks Flow Simulations were made for both of these propulsion systems in order to determine the terminal velocity of the platform achieved by each system. With these terminal velocity values, the amount of power consumed by each propulsion system can be analyzed to determine whether these requirements will be met.

8.2.1. Simulations

In order to discover whether or not the hydro pumps and motor-propeller systems that were chosen would be capable enough to propel the platform to the desired distance per day, a SolidWorks Flow Simulation was conducted.

8.2.1.1. Procedure

These simulations consisted of finding the terminal velocity of the platform; in other words, the speed of the platform when the drag force equals the thrust force. The drag force is the force acting opposite to the relative motion of the device moving with respect to the water, whereas the thrust force is the force caused by the propulsion system to get the device moving.

With the simulations the team will recreate the speed of the platform relative to the flow. In a reservoir, the platform will move in stationary water; in the simulation the water moves past a stationary platform. These two scenarios are equivalent because the relevant fluid forces depend on the relative velocity between the platform and the water, which is the same in both cases.

With this estimated velocity, the simulation was run to obtain the drag force. This drag force is to be compared to the thrust force made by the propulsion system at the estimated velocity. After a number of iterations, the study came to an end when the drag force and thrust force were the same, which means that the terminal velocity of the platform had been reached.

This procedure is applied in the simulations of both methods of propulsion: hydro pumps and propellers. However, what differs from one propulsion system to another is the way the thrust force is initially calculated.

8.2.1.2. Hydro Pump Simulation

With the estimated velocity of the device, the speed at which the flow comes out of the hydro pumps can be computed, and therefore the outlet pressure in the nozzle (with the pump performance curve, Graph 1). With the outlet pressure, the thrust force generated by the hydro pumps can be obtained.

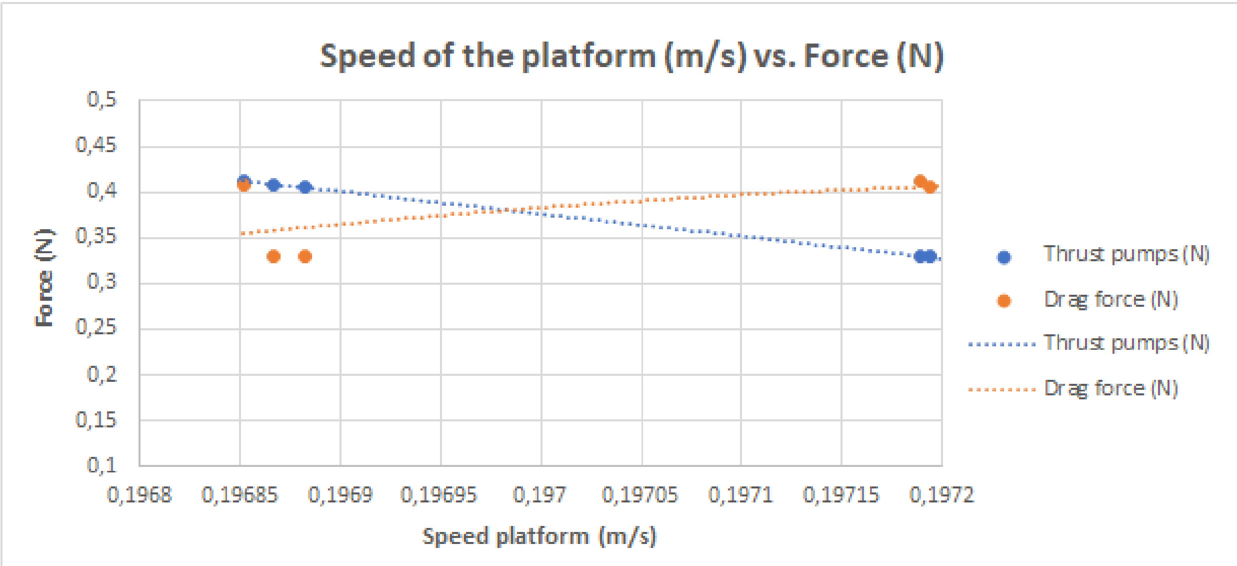
If the drag force obtained in the simulation was higher than the thrust force, the flow rate was decreased and another simulation was started. If the drag force obtained was lower than the thrust force, the flow rate was increased and another simulation started. The process will end when both forces are equal. To better understand the whole process, the calculations are shown in the Appendix 13.3.3.

The number of simulations the team had to complete in order to obtain the terminal velocity of the device is shown below (Table 8).

Iteration	v flow = v platform (m/s)	Fdrag (N)	v pump (m/s)	F thrust pumps (N)	P (Pa)	H (m)	Q (m^3/s)
1	0,1052	0,1301	0,9523	20,646	16096,78	1,6458	0,0006107
2	0,198	0,329	1,7927	0,1301	101,434	0,01037	0,0011497
3	0,1972	0,405	1,7854	0,329	256,509	0,02623	0,00114499
4	0,1968	0,3298	1,7826	0,405	315,763	0,0323	0,00114318
5	0,1972	0,4125	1,78537	0,3298	257,1327	0,0263	0,00114496
6	0,1968	0,4088	1,7823	0,4125	321,598	0,03288	0,00114300

Table (8). *Results obtained from the pump simulation to compute the terminal velocity of our device.*

With these results, a graph representing both forces with respect to the speed of the platform was made, and it is shown below (Graph 2):



Graph (2). *Drag and Thrust Force vs. Speed of the platform.*

As seen on the previous graph, the blue curve shows the thrust force generated by the hydro pumps with respect to the speed of the platform and the orange curve represents the drag force with respect to the speed of the platform. The point in which both curves intersect is the point in which the values of both forces are the same. In other words, that is the point in which the platform reaches its terminal velocity. The terminal velocity of the platform is 0.197 m/s when using the hydro pumps as the propulsion system, that is 0.44 mi/hr.

8.2.1.3. Propeller Simulation

For the propeller simulation, first the maximum thrust was calculated using the propeller-motor system specifications. This maximum thrust value was 21.9 N, Appendix 13.3.4.1. As explained above, a velocity flow value was inputted into the simulation settings and the generated force of drag on the platform was outputted. In Table 9, the third column are the inputted flow velocity values and the fourth column are the subsequent drag force values. The simulation was run until the drag force and maximum thrust were sufficiently close. This terminal velocity was 1.2 m/s. This is an approximately 84% larger terminal velocity than the hydro pumps can perform.

Trial	Motor Max Force [N]	Velocity of the Boat [m/s]	Simulation Drag Force [N]	F_Difference Motor	Coefficient of Drag
1	21.9	7.5598	632	610.1	0.44225604
2	21.9	5	316	294.1	0.50550375
3	21.9	3	131	109.1	0.58211139
4	21.9	3.25	135	113.1	0.51114528
5	21.9	3.5	175	153.1	0.57131979
6	21.9	3.4	150	128.1	0.51893234
7	21.9	3.35	163	141.1	0.58086512
8	21.9	3.3	163	141.1	0.59860044
9	21.9	3.2	149	127.1	0.58192045
10	21.9	1	14	-7.9	0.5598934
11	21.9	1.5	30	8.1	0.53323181
12	21.9	1.1	15.9	-6	0.52551978
13	21.9	1.2	18.1	-3.8	0.50268207

Table (9). *Propeller simulations trials.*

One factor that was worth analyzing with the propeller simulation results was the coefficient of drag. The rightmost column of Table 9 shows the coefficient of drag for the design at each flow velocity. At our terminal velocity, the coefficient of drag is 0.5. This is comparable to a conical frontal area. To view a sample calculation of the coefficient of drag see Appendix 13.3.4.2. Note that these coefficients of drag calculations are only an approximation, as the frontal area of the platform was simplified for the calculation.

In conclusion, the terminal velocity that the platform could maintain with the hydro pumps as the propulsion system was 0.19 m/s (0.44 mi/hr) and for the motor-propeller system was 1.2 m/s (2.68 mi/hr). This information is crucial for performing the power subsystem study according to which propulsion system the team chooses (Appendix 13.3.6).

8.2.2. Power Calculations

As mentioned previously, the mobile platform should run a minimum of 6 hours. Based on these terminal velocity values found, the continuous run time at maximum speed for the hydro pumps and propellers without recharging was calculated. The hydro pumps, rated at 12 volts and a current of 7.04 amps, were calculated to have a continuous run time of 58 minutes. Meanwhile the propellers, each rated at 12 volts and a current of 6 amps, were calculated to have a continuous run time of 32 minutes (Appendix 13.3.6.1). In order to ensure the minimum 6 hour run time, the team decided to add a solar energy recharging component.

One important aspect to consider before doing the time calculations including the recharging on the batteries is the actual run time. It is known that realistically the platform would not run continuously at max speed for 6 hrs. Instead, based on the customer suggestions, a hypothetical format of a typical day of recording data was created. This format “typical run time” included the process of data collection where the device would move for 1 min and then stop and record data for 6 min and so on. With this more realistic run time, the hydro pumps and propeller system could run for 6.48 hours and 3.6 hours respectively (Appendix 13.3.6). It is clear that the propeller system requires more power to run for the required 6 hours, in this scenario only the hydro pump is able to run for the desired period of time. Again, this is why the final design included a solar energy recharging capability to full-fill this 6 hour run time. The solar panel is capable of recharging the battery at a fast enough rate that it could run endlessly as long as the sun is present. (Appendix 13.3.6)

The second requirement that was dependent on the power subsystem and propulsion system was the minimum distance traveled per day. In order to determine this parameter, the “typical run time” was again used to determine the maximum distance traveled over a 6 hour period and then augmented for a full day. The calculations are depicted in (Appendix 13.3.6). The hydro pumps would travel 0.76 miles during the platform’s operation time in a day and the propellers would travel 3.46 miles. It is apparent that the propellers are capable of making the 2 mi/day metric while the hydro pumps are not.

8.3. Sensor Deployment Testing

Another requirement given to the team for the final design was for the sensor to deploy to different depths up to five feet. This depth was accomplished for both iterations of the sensor deployment method (Section 7.3).

During the construction of iteration 1, two tests were conducted to determine how well the cable descended into water. The first test consisted of deploying the sensor into a foot and a half bucket of water in EPIC as shown below (Figure 53). The primary goal of this test was to ensure the cable did not slip on the threaded spool, and to test the effectiveness of the roller behind the threaded spool. A video of this test is available in the team’s drive folder.

Figure (53). *First Test for the Iteration 1 Sensor Deployment Mechanism.*

It was determined that the spool was an effective method of deploying the sensor, and that the roller did assist in keeping the cable on the spool when the cable unreeled.

To simulate the environment the sensor would deploy in, the mechanism was tested in a larger body of water. This second test was performed in the FitRec pool to primarily observe the cable dispensing (Figure 54).

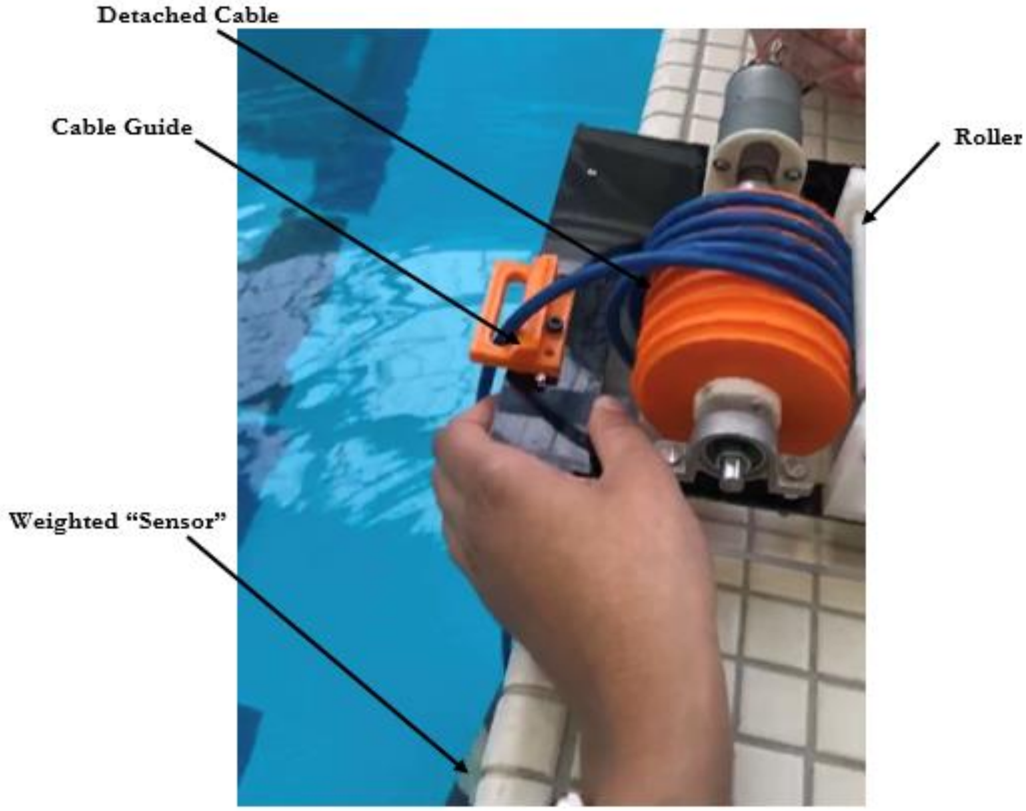


Figure (54). *Testing Iteration 1 in the FitRec Pool.*

As seen in figure above, the cable began to separate from the threaded spool while the sensor was deploying. When the sensor was reeled back onto the spool, it was without incident. This occurred because gravity was acting against the sensor as it was reeled up and so the sensor was pulled downwards, causing the cable to stay on the spool. When the cable was released into the water, there was not enough force on the sensor due to the force of buoyancy from the water and so the cable did not have enough tension to keep it on the spool. From this test it was determined that a sensor casing was necessary to weigh down the sensor as well as adding rollers in more locations around the threaded spool.

Another test conducted with the sensor deployment mechanism involved measuring how accurately the sensor deployed to a given depth. This test was conducted after the bi-directional communication was developed. With this code, the motor should ideally be maintained at a speed of 13 rpm; however, due to the instability of the mechanical structure and errors in the PID tuning, it was expected that an error in displacement would occur. The figure below (Figure 55) shows the set up for the experimental test.

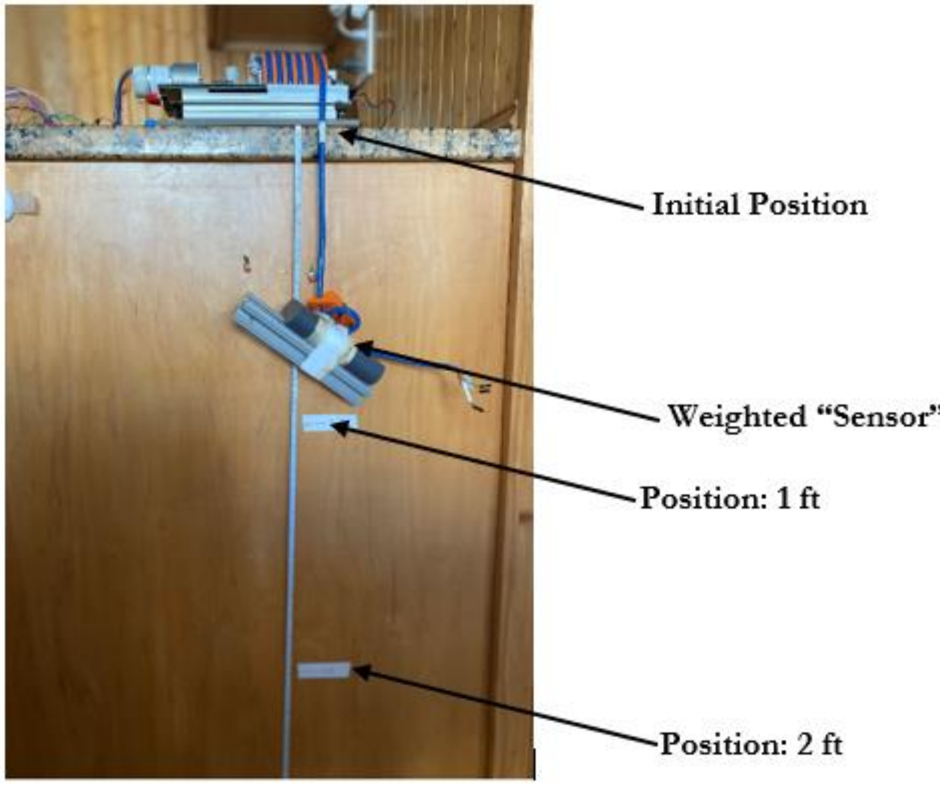


Figure (55). *Measuring the depth accuracy of the Sensor Deployment.*

To determine the error between the actual depth reached by the sensor and the user-inputted depth, a ruler was taped along a wall while commands were sent to lower and raise the sensor. The cable depth was measured from the top of the ruler to the top of the white piece of tape on the cable. The two white pieces of tape on the wall indicate 1 ft. and 2 ft. The initial position of the cable was set so that the top of the piece of tape on the cable was aligned with the top of the ruler.

To test the downward deployment, the cable began at the initial position and traveled to the user specified depth of 1 ft for each trial. To test the upward ascension, the cable began at the specified distance of 1 ft and traveled upward for the given increment of 1 ft, three times. This procedure was repeated for a depth of 2 ft. The results of the measured distances are summarized in the table below (Table 10 and Table 11):

Depth (in)	Trial 1	Trial 2	Trial 3
12	11	11.5	12.5
24	23.5	24.5	23

Table (10). *Upward Motion.*

Depth (in)	Trial 1 (in)	Trial 2 (in)	Trial 3 (in)
12	13	15	13
24	25.5	25	25.75

Table (11). *Downward Deployment.*

As predicted, the cable depth differed from the user inputted depth by an average of an inch and a half. As seen in the tables, the error was larger when the sensor moved downwards and less when the sensor moved upwards. It is hypothesized that this difference in errors between upward and download deployment has to do with the force of gravity. In downward deployment, gravity is working with the motor and in upwards deployment, gravity is working against the motor. An attempt to correct this error in the PID controller was to send a higher starting voltage when the sensor ascended and a lower starting voltage when the sensor deployed.

8.4. Wireless Range Testing

To evaluate the capability of the wireless transmissions, the XBee Pro modules were compared to an alternative transceiver - Lora Reyax Rylr896- to determine which module could more reliably transmit data for a longer range (Figure 56). Both modules were tested on the same days so that the weather conditions would be a constant in evaluating their performance. The testing environment was a mostly flat beach where a direct line of sight was available for approximately half a mile. Schematics and the Arduino Codes used for testing both modules are found in the Appendices 13.4 and 13.5, respectively.

Figure (56). *Picture of both the Lora Module (Left) and XBee Module (Right).*

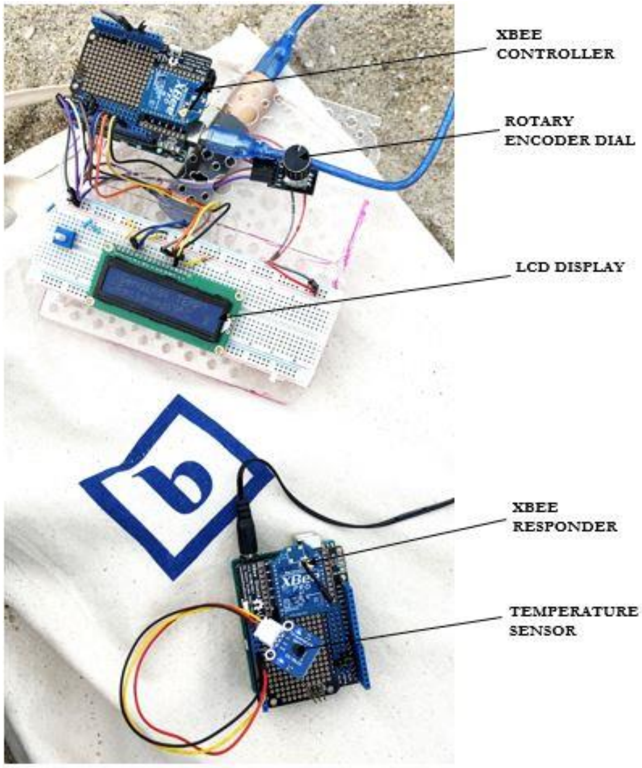


Figure (57) a

Figure (57) b

Figure (57). *Set up for the range testing of the XBee Pro Modules.*

First, the viable range for the XBee communication was determined. To do this, a shorter two-way communication protocol was written where the user could only send a signal to the responder Arduino and this Arduino would transmit back the temperature at that point in time. As can be seen on the LCD display in Figure 57b, the temperature was still received from a great distance. However, on both of the testing days the maximum range reached was 320 m. Beyond this point, some signals from the controller to the responder were sent however no temperature data was able to be relayed back.

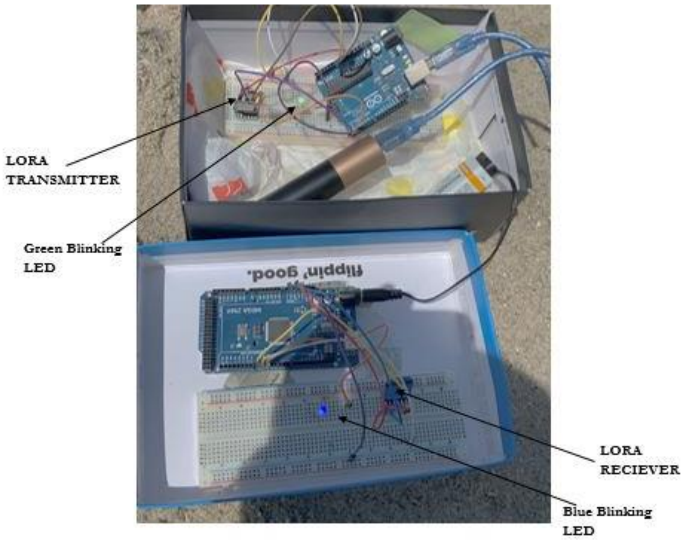


Figure (58) a

Figure (58) b

Figure (58). *Set up for range testing for Lora Modules.*

The alternative Lora Reyax Rylr896 transceiver module was also explored, pictured in Figure 58. These modules are advertised as being long range and high having interference immunity while maintaining the same low cost as the XBee's. Unfortunately, the bi-directional communication established was not reliable enough for testing and so, for these modules, a one-way communication protocol was implemented. The Lora transmitter would send a recurring message every few seconds (indicated by the blinking green LED), and if the Lora Receiver obtained the message successfully the blue LED light would turn on for half a second. On both days of testing, the Lora module was able to still receive signals up to 1.3 miles in a relatively urban area. It should be noted that this may not be the full working range due to the environment in which it was tested. After about half a mile, there was no more direct line of sight because of the terrain and due to the circumstances, the team was unable to test either module in an environment that most closely resembled the client's testing waters. Another feature that was not tested was bi-directional communication, which could also affect the range of signal transmission.

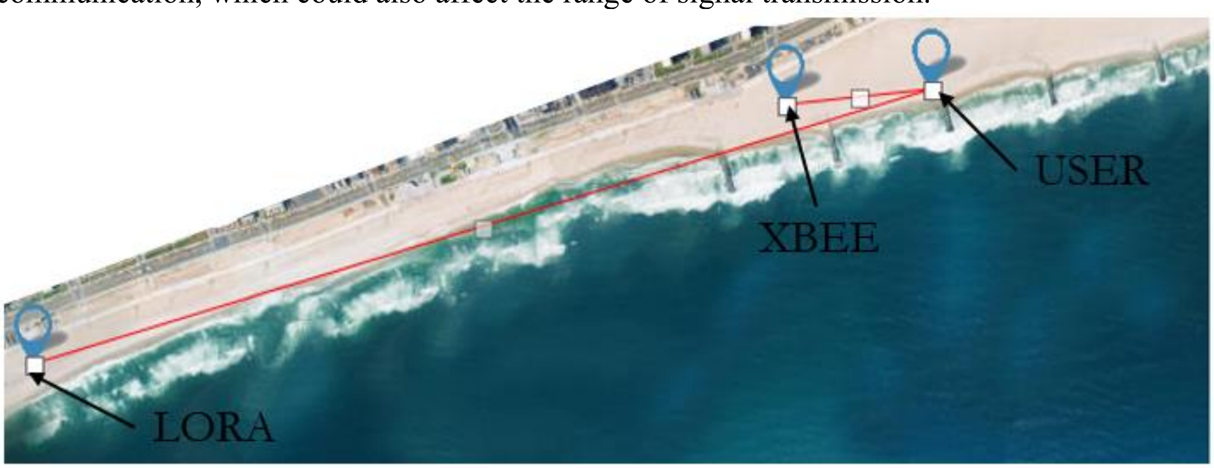


Figure (59). *Visualization of Range Tests.*

As seen in Figure 59, the difference in range for the XBee and Lora module was very significant. To ensure the Lora signals were working properly, the Arduino was turned off and back many times during testing to make sure the signal was still received. It was found that the further the distance of the Lora module, the longer it would take for the first signal to be received however after the first signal, communication would resume. When this same procedure was applied to XBee, it was found that sometimes the signal would be able to be picked up again after being lost however it was not reliable and so the range was capped to the distance of reliable transmission.

Another evaluation, shown in Figure 60, was conducted with an Arduino compatible GPS module (UBlox Neo6m); however, it was found that this module only works when the antenna is in view of the sky and it is a sunny day. When moving the GPS in the same environment the modules were tested in, the GPS did account for small changes in the location and compared to the GPS locations on Google Maps, the error was about 0.3 degrees for both latitude and longitude. This GPS was low cost and so these errors were expected; however, it portrays the capability of putting a more precise GPS module on the platform and transmitting the location data to the user.

Figure (60) a

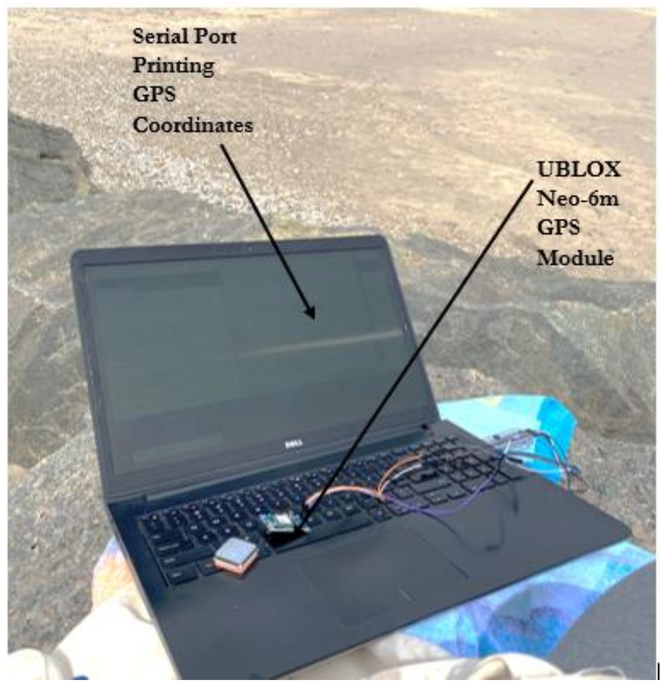


Figure (60) b

Figure (60). *Set up to measure accuracy in GPS Module.*

8.5. Table of Results

Parameter	*Engineering Specification*	*Final Specs*	*Did we meet spec?*
Platform	Relatively small; waterproof; sturdy	33 in by 18 in by 5.4 in, waterproof	Met Spec
Propulsion	Minimum travel: 2 mi/day	Hydro Pumps: 0.76 mi/day (w/ recharging)	Below Spec
		Propellers: 3.46 mi/day (w/ recharging)	Exceeds Spec
Sensor Deployment	Required depth: 5 ft	5 ft	Met Spec
Power	Minimum run time: 6 hrs	Hydro Pumps: 6.48 hrs (w/o recharging) Hydro Pumps: 11.48 hrs (w/ recharging)	Exceeds Spec
		Propellers: 3.6 hrs (w/o recharging)	Below Spec
		Propellers: 8.6 hrs (w/ recharging)	Met Spec
	Renewable energy component preferred	Solar Panel Included	Met Spec
Data Transmission	Minimum distance: 300 m	320 m	Exceeds Spec
	Minimum transmission: 1 rdg/ 5 min	1 rdg/ 30 sec	Exceeds Spec
Control	Autonomous preferred, remote control acceptable	Remote-controlled	Met Spec

Table (12). *Table of final results compared to initial specifications.*

9. Cost Analysis

The chart below (Table 13) displays the cost of each item and the total cost of the project as well as the actual amount of money spent for the project.

Item	Quantity	Count	Cost	Amount Spent
Power Components				
Solar Charge Controller	1	1	$10.92	$10.92
Portable Battery Changer	1	1	$15.92	$15.92
Solar Panel	1	1	$29.98	$29.98
SLA Rechargeable Battery 12V, 9 Ah	1	1	$19.95	$19.95
Electrical Components				
Push Button Switch Attachments	1	20	$14.90	$14.90
Arduino Uno R3	2	1	$46.00	-
Adafruit XBee Shield v2.1	2	1	$21.00	-
Arduino Relay Shield v3	1	1	$20.00	-
Hydro Pump Bilge Model 27D	2	1	$106.94	-
Waterproof Junction Box	1	1	$28.14	$28.14
MultiStar 5200mAh	1	1	$13.00	-
MicroSD card breakout board+	1	1	$7.50	$7.50
SD/MicroSDMemory Card (8 GB SDHC)	1	1	$9.95	$9.95
Gear Motor w/ Encoder (12 V 16.7RPM)	1	1	$18.50	$18.50
LCD Displays	1	2	$5.88	$5.88
Motor Drive Controller	1	1	$2.29	$2.29
Lora Module Antenna	2	1	$39.00	$39.00
Temperature sensor	1	1	$7.99	$7.99
GPS controller module	1	1	$13.49	$13.49
Heat-Shrink Cable Splitter	1	1	$18.36	-
Mechanical Components				

3D printer Stepper Motor with 300mm Lead Screw	1	1	$36.00	-
8mm Diameter, 330mm Length Shaft	1	1	$5.33	-
4-40 .5in slotted flat head screws (~16 used)	1	100	$4.36	$4.36
0.25in diameter 7ft long cable	1	5	$8.15	$8.15
1ftx2ft Clear Acrylic	1	1	$18.49	-
8inx1ft Black Acrylic	1	1	$11.49	-
2inx48inx8ft waterproof foam board	1	1	$34.95	-
3/8 in. -16x6-1/2 in. Zinc Plated Carriage Bolt	4	1	$5.52	$5.52
Bearings	1	4	$12.77	$12.77
Zinc Yellow-Chromate Plated Hex Head Screw	1	5	$14.02	-
17-7 PH Stainless Steel Washer	1	5	$13.12	-
Medium-Strength Steel Hex Nut	1	100	$8.79	-
Black-Oxide Alloy Steel Socket Head Screw	1	10	$5.77	-
316 Stainless Steel Washer	1	100	$7.11	-
Misc				
Paint Brush	3	1	$5.97	$5.97
Paint	1	1	$7.99	$7.99
Waterproof Electrical Tape	1	1	$4.79	-
			Total Cost	**Total Cost**
			$654.33	**$269.17**

Table (13). *Total Cost vs. Total Spent.*

A budget of $400 was granted to each group to spend for the project. For this project, a total of only $269.17 was spent on project supplies and materials.

Costs were kept low because the group was able to utilize some supplies and materials, such as the hydro pumps, batteries, foam, and Arduinos, were available to the group without cost. Additionally, for the acrylic features in the sensor deployment mechanism of the platform, the group only used scrap materials available to all the senior design groups. Although the customer offered to fund the project an additional $500 for parts such as batteries and a solar panel, the funds were not needed in the end.

Overall, the total cost of the project was $654.33, but the group only spent $269.17. The difference is due to the supplies and materials made available to the group without cost.

Based on future project improvements, the group anticipates that the total cost will change in the future. For example, the propellers are more efficient than the hydro pumps, and also less expensive by $36.94. This improvement could reduce the total cost of the project.

In summary, by choosing the most cost-effective parts and materials, the total cost of the project, $654.33, was well below the total $900 available. The group met most of the project specification goals, given the COVID-19 disruption. However, the group could not determine the manufacturing and 3D printing costs. Since the group did not have access to EPIC to send the parts to be 3D printing, those costs were not included in the total cost.

10. Discussion

10.1. Structural Subsystem

The foam platform, acrylic securing piece, and cover were designed in order to give the design stability and protection. Though the platform was not able to be tested in water, it is believed that the chosen dual hull design will give the platform the stability it needs to safely deploy a sensor as well as withstand any wind or waves that may be present in the drinking water reservoir. Additionally, the buoyancy calculations allowed for the creation of a platform large enough to hold the weight of all components aboard but at the same time small enough to be held by one person. Additionally, the ramp and roller system at the front of the platform will allow for smooth deployment and reeling in of the sensor.

As it is difficult to fasten something securely into foam, the acrylic securing piece was a simple solution to attach all the necessary components rigidly onto the platform. Additionally, the seal that will be placed into the outer groove on the acrylic piece allows for as much waterproofing as possible for the internal components. Once the cover is placed on top and bolted into place above the acrylic, the sealing material will compress, thus waterproofing the edges of the platform. The cover acts as the main protection feature for the internal components from the elements and surroundings. With this in place, nothing should disturb these components.

10.2. Sensor Deployment Mechanism

Although Iteration 2 of the sensor deployment mechanism was not able to be constructed, all the improvements to the design stemmed directly from testing of the first iteration. By testing the first iteration in water, it was found that more well-designed roller placement was needed to guide the cable on the threaded spool. Therefore, the roller cover was designed on CAD and houses the same rollers that were used for the testing method. Another result of testing demonstrated the need to weigh the sensor with at least a few ounces more weight so that it sinks into the water easily. The solution to this was to design a simple sensor casing that could either be 3D printed with full density or machined from a metal such as aluminum.

To determine how to connect the wiring from the sensor to the Arduino, modifications such as drilling holes into the threaded spool and aluminum shaft were made to the testing model. The goal was to thread wire through the spool, into the shaft and outwards into the Arduino. Testing was then conducted and although the wires twisted initially, it was determined that with enough slack wiring, the twisting did not affect the wiring to the Arduino. This can be seen in the final demonstration videos at the following link: Sensor Testing Videos Therefore from these experimental testing procedures, the final design employed more sophisticated cable routing. Since the testing model had a 0.25 in thick acrylic piece that bent significantly towards the center, additional changes were made to account for this in the CAD Iteration by including a sturdy base to hold the mechanism.

10.3. Propulsion System

After running the simulations of both hydro pumps and propellers, the following results were obtained (Table 14). While the hydro pumps get the terminal velocity of the platform to be 0.44 mi/hr (0.19 m/s), the propellers achieve a terminal velocity of 2.68 mi/hr (1.2 m/s). With these speeds and taking into account a realistic run time of a typical day of recording data, the "typical run time" mentioned in previous sections, the hydro pumps could run the platform for 6.48 hours and the propellers for 3.6 hours without the battery being recharged by the solar panel. However, and again following the typical run time of the platform during the day, the maximum miles the hydro pumps and propellers could travel per day recharging the battery with the solar panel were 0.76 and 3.46 mi/day respectively.

		Typical run time (w/o recharging)		**Typical run time (w/ recharging)**	
	mi/hr	**mi/day**	**Hours**	**mi/day**	**Hours**
Hydro pumps	0.44	0.43	6.48	0.76	11.48
Propellers	2.68	1.45	3.6	3.46	8.6

Table (14). *Terminal velocities and propulsion capabilities.*

As shown in Table 14, a solar panel is needed in the device, just as the customer suggested, because neither the hydro pumps nor the propellers meet the minimum distance travel requirement of 2 mi/day with the current battery the team owns. In addition, without recharging the battery, the propellers would not even meet the minimum time requirement of 6 hours. If it were considered a bigger battery with larger capacity, the platform would need to be bigger, and consequently it would be heavier, and therefore it would not meet the requirement of being as smaller and light as possible. With the recharging capability provided by the solar panel and on a day with good sun conditions (assumed a 5 hour of useful sunlight during a day in Louisiana), it is believed that the propellers would also be able to achieve the 6 hour minimum run time, as the battery will not reach discharge if there is sunlight. Therefore, the propellers would achieve both minimum run time and minimum distance traveled requirements.

Between the two mechanisms, it was determined that for future iterations of this project the propellers should be the primary propulsion system rather than the hydro pumps. Besides the fact that with a recharging component the propellers would achieve both requirements listed before, the propellers also have the advantage over the hydro pumps in that they have a variable speed capability while the hydro pumps do not. Further, if the propellers were run at a lower speed, they would draw less energy from the power systems. This would be ideal as the customer has outlined that the platform does not need to travel at any high speeds.

Another aspect to consider about the variable speed capability of the propellers is that it allows for more controlled maneuvering for the operator. For example, with the hydro pumps operating at their one speed, if the operator wants to stop the platform at a specific spot they would have to estimate how long the platform would coast for after the hydro pumps were turned off in order to actually stop at the desired spot. With the variable speed propellers, as the platform is approaching the desired location, the operator may throttle down the speed of the propellers for a more controlled and accurate stop. This variable speed option is also advantageous for delicate maneuvering, for example if the platform only needs to move a small distance.

Another reason to consider the propellers over the hydro pumps is that propellers are generally more efficient than hydro pumps at low speeds. And as mentioned before, the customer does not require high speeds, therefore the propeller seems to be the best of the two options.

10.4. Electronics Subsystem

The results of the range testing procedures, depth measurement accuracy, and bi-directional communication protocol indicate that the designed wireless transmission system is successful in meeting the relevant customer requirement. This wireless transmission system also exceeds the basic requirements for the code by optimizing the data transmission and user control function structure. With each milestone reached in finalizing the code, more versatility for the subsystem was reached. For example, with the ability to communicate bi-directionally, the platform is able to send updates for which procedure it is currently executing (such as deploying the cable), as well as transmit temperature and GPS data. Another milestone of implementing a user menu on the controller display allows for more options in data collection as well as creating a user-friendly console.

One important observation to make is that the motor speed control is not developed enough to precisely control the depth the cable descends and ascends. The error in displacement is approximately 1.5 in and further testing with the mechanism on the platform would have been required to observe how this error affects sensor placement in the “Home” position. The “Home” position is simply the initial position of the sensor on the platform ramp.

The data acquisition procedure was found to be very successful in that the user has the option to choose how long to acquire data and to select a rate at which they want to receive real-time data. At the same time, all the data is saved onto an SD card. One issue that arises is the compatibility with sensors that require constant current and a high voltage. Although for the prototype we used an Arduino compatible sensor, the electronics design is alternatively capable of implementing a 4-20 mA loop as a mediary between the sensor and the Arduino. The code to collect and store data would not be affected because using this 4-20 mA loop would still output usable voltage readings to the Arduino processor.

Another improvement made from the feasibility demonstrations was the adjustment to speed control and direction. The new iteration was not able to be fully tested with either the pumps or the propellers due to COVID-related circumstances and therefore could not be debugged for optimized control. However, with the instruments on hand, the code tested intended to allow the user to control the speed with the use of a rotary encoder dial and two buttons that need only to be pressed once to change the direction. In the conceptual design system, when both buttons needed to be pressed, there were significant issues in signal loss because the data was being sent too rapidly. In this updated code, a toggle switch drives both motors forward in the upper position and both motors turned off in the downward position. This allows for one signal to be sent instead of a continuous stream and the controls are able to be implemented more reliably and more quickly.

While the system performs as expected, some debugging would still be necessary to optimize processing power on the Arduino. Currently both the controller and platform Arduino storage is at an 80% capacity. At this level, instability occurs not because of the wireless transmission or the code but because of the processing needed. A solution to this is to replace both of the Arduino's with Arduino Mega boards because they offer more processing power and storage for variables used throughout the code. Another debugging feature that was observed to be needed was the recalibration of the "Home" position of the sensor cable after many iterations of use. This is due to the accumulation of accuracy errors over the cycles and a method to check whether the cable is in the correct position will require an external input such as a button press to be included in the "Homing" function.

10.5. Power Subsystem

As mentioned previously, the battery alone was able to successfully sustain the platform with the hydro pumps for 6.48 hours as required by the customer but was unable to power the propellers for the full 6 hrs. The batteries and propellers alone ran for 3.6 hrs.

With the solar panel connected to the system, this run time of the platform was augmented. The solar panel chosen has the capability of charging the battery faster than it discharges to the hydro pumps. For example, if there are 5 hours in the day with usable sunlight this would add 5 hours to the run time of the boat, since the battery can be recharged. If there is no sun and the solar panel cannot charge the battery, the platform would still run for more than 6 hours when using the hydro pumps.

It was also considered a design using propellers. The only downfall of the propellers is that without the sun, the battery would only last for 3.6 hrs. This shorter run time comes from the higher current draw needed for the propellers. The higher speed propellers require more power compared to the hydro pumps. If the sun is present, the solar panel will add to the run time. So, if there are 5 hours of usable sunlight during the day, the platform would be able to meet our 6 hour requirement while using the propellers.

11. Conclusions and Future Work

11.1. Conclusions

In conclusion, this project successfully meets the majority of the given requirements. Overall, the team's ingenuity, hard work, and persistence allowed the customer's vision to become a reality.

Firstly, the foam platform was skillfully constructed based on important weight estimations and buoyancy calculations and has the necessary dimensions to safely house the platform components as well as provide sufficient stability in rough waters. The design decision to add the acrylic securing piece ensured proper fastening of critical internal components of the design. Moreover, the cover works to shield these components from the elements in addition to holding the solar panel towards the sun.

Second, the sensor deployment mechanism allows the sensor to be deployed up to a depth of 5 ft and is also capable of deploying different sensors with interchangeable cabling. This mechanism can be deemed very successful because every changed feature on the CAD Iteration resulted from rational choices based on the testing and analysis of these features on the testing model. Therefore, the proposed CAD Iteration would be a neater, and more aesthetically pleasing design that is more compact, easily fabricated, and structurally sound.

The wireless transmission system is also completely functional for the sensor deployment controls with demonstration videos available in the team google drive. The sensor deployment controls exceed the customer basic requirements by allowing the operator to choose between automatic or manual data collection. Other options such as *Sweep Temperature* allows the user to have an initial read of the temperature in that area before deciding to collect data. The propulsion control part of the wireless control was not able to be fully tested with either propulsion method however it is believed, based on past testing, that only minor debugging would be necessary to perfect these controls.

In terms of the power capability of the final design, the battery alone can power the hydro pumps in the "typical run time" schematic for the 6 required hours, but without meeting the minimum travel distance requirement of 2 mi/day. For implementation of the propellers, the solar energy recharging capability is integral in achieving this required data collection time of 6 hours. As mentioned above, although neither final propulsion system was able to be physically tested, the team was determined to complete a SolidWorks Flow Simulation in order to ascertain each system's terminal velocity for power calculations. With the terminal velocities, the hydro pumps were found to have a distance traveled per day of 0.76 mi/day, and the propellers of 3.46 mi/day, thanks to the recharging capability provided by the solar panel.

The final design consists of a structural mobile platform that houses a sensor deployment mechanism to reliably lower and retrieve a sensor, a remote-control system that controls the propulsion and sensor deployment mechanism, and a solar-powered battery recharging system. With the controller, the user can control the speed and direction of the platform as well as to what depth the sensor will be deployed. A solar panel mounted on top of the cover recharges the batteries onboard and enables the customer to use the platform for extended periods of time. The final design incorporates the hydro pump system since the propellers could not be physically tested. This design has paved the way for the creation of a fully autonomous, renewable-energy powered vessel in the future.

11.2. Future work

Aspects of this design that should be considered for future work include weighing the hulls to assure the platform does not flip, implementing a more efficient propulsion method such as propellers, adding a higher capacity battery to the propulsion circuit, and obtaining a more reliable GPS module to allow for a full autonomous capability of the device. Although water reservoirs are mostly calm, the team considered weighing the hulls by adding a heavy material inside them to assure the platform does not flip over due to the wind waves or other external elements.

Adding a more efficient propulsion method such as propellers will allow for both distance and time requirements to be met as well as more precise maneuvering. In order to exceed these distance and time requirements, a higher capacity battery should be considered.

Autonomous capability was an aspect of this design that the customer was hoping to have accomplished with this iteration; however, for the prototype it was determined that a remote-controlled option was most feasible to complete. Two features that would need to be improved to allow for autonomy are a robust GPS Module and a more precise propulsion system. With these implementations and the addition of some coding, autonomous capability is certainly possible. The current Arduino code would only need to be slightly altered by including more commands and procedures as well as more possible cases of functionality.

Furthermore, although the current wireless transmission system with XBee's does meet requirements, it is recommended to upgrade to a more sophisticated wireless transmission system such as Lora modules to expand the transmission range. With the implementation of the Lora modules, first, bi-directional communication would need to be improved. After that, the rest of the Arduino code can then be adjusted from the XBee codes. Along with data transmission, future testing should solely be done with sensors that are more similar to the customer's sensors to account for any issues when adding additional circuitry to read the sensor data.

Another aspect of the customer's request that we did not specifically implement was a separate component to deter the birds. Although the cover does prevent animals from disturbing the main components, there is nothing to stop the birds from sitting on the cover. This could interrupt or even damage the solar panel as well as put strain on the cover and change the balance of the platform in the water.

With our current design used as a skeleton system for future improvements, it is reasonable to conclude these adjustments could be made as improvements to the current solution rather than selecting and implementing completely new systems.

12. References

[SANC19] Sanchez, H, "What's Up With The Algae Blooms In Colorado And Why Are They So Hard To Track?", CPR News, Aug. 2019. https://www.cpr.org/2019/08/29/whats-up-with-the-algae-blooms-in-colorado-and-why-are-they-so-hard-to-track/

[AGGR18] Aquatic Germplasm & Genetic Resources Center (AGGRC), "Dr. Teresa Gutierrez-Wing", 2018. https://www.aquaticgermplasm.com/teresa.html

[HERO19] Herold, N., "5 Facts About Water Scarcity: Global Water Sustainability", MECO, 28 April 2019. www.meco.com/facts-about-water-scarcity/.

[RUTH15] Rutherford, A., L. Benedict, "Keep Louisiana's Water Resources Plentiful and Good", LSU AgCenter, Louisiana State University, 12 Oct. 2015. www.lsuagcenter.com/portals/communications/publications/agmag/archive/2011/fall/keep-louisianas-water-resources-plentiful-and-good.

[NOAA09] US Department of Commerce, and National Oceanic and Atmospheric Administration. "What Is Nutrient Pollution?" NOAA's National Ocean Service, 1 Sept. 2009. https://oceanservice.noaa.gov/facts/nutpollution.html

[TCHO12] Tchounwou, P. B., C. G. Yedjou, A. K. Patlolla, D. J. Sutton, "Heavy Metal Toxicity and the Environment", Experientia Supplementum (2012), U.S. National Library of Medicine, 2012. www.ncbi.nlm.nih.gov/pmc/articles/PMC4144270/.

[EPA_17] Environmental Protection Agency (EPA), "Epidemiology & Health Effects of Cyanobacteria", 9 Nov. 2017. www.epa.gov/water-research/epidemiology-health-effects-cyanobacteria.

[AMS_19] Autonomous Marine Systems Inc (AMS), "Mark 8", 2019. www.automarinesys.com/mark8.

[PB__16] ProprioBateau.ca, "Aquila, 44 Aquila (2016)", 2016. https://www.propriobateau.ca/boats/motor-boats/catamaran/aquila-44-aquila-2016-24368.html

[JOYC19] Joyce, N., "After building a dynasty and leaving unlimited hydroplane racing 15 years ago, Miss Budweiser still fuels sport", The Seattle Times, Aug. 2019. https://www.seattletimes.com/sports/other-sports/15-years-after-building-a-dynasty-and-leaving-unlimited-hydroplane-racing-miss-budweiser-still-fuels-the-sport/

[AMIT19] Amitai, P., "Image result for monohull boats design", Pinterest, 2019. https://www.pinterest.de/pin/745697650775642780/

[MINN19] MI News Network, "Main Types of Catamarans Used in the Shipping World", Marine Insight, 13 Oct. 2019. www.marineinsight.com/boating-yachting/main-types-of-catamarans-used-in-the-shipping-world/.

[PIKE11] Pike, J., "Hydroplane Planing Hull", 7 Jul. 2011. www.globalsecurity.org/military/systems/ship/hydroplane.htm.

[PARK18] Parkinson, A., "Sailboat Debate: Monohull vs. Catamaran", Yachts International, 30 Jul. 2018. www.yachtsinternational.com/owners-lounge/sail-debate-monohull-vs-catamaran.

[OLEA09] O'Leary, N., S. Shelly. "Which Metal Corrodes the Fastest?", TeacherVision, 13 Aug. 2009. www.teachervision.com/chemistry/which-metal-corrodes-fastest.

[WIKI19] Wikipedia, "Underwater Thruster", Wikimedia Foundation, 3 Oct. 2019. https://en.wikipedia.org/wiki/Underwater_thruster.

[BR__19] Blue Robotics, "T200 Thruster for ROVs, AUVs, and Marine Robotics", 2019. bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster/.

[MORT__] Morton, G. E., "The Paddlewheel". www.americansternwheel.org/Boats/articles/the_paddlewheel.pdf.

[MASO15] Masoomi, S., S. Gutschmidt, X. Chen, M. Sellier, "The Kinematics and Dynamics of Undulatory Motion of a Tuna-Mimetic Robot", International Journal of Advanced Robotic Systems 12(7) 1-11, DOI: 10.5772/60059, Figure 2: Link mechanism of the tail penduncle, July 2015.

[YANG70] Yang, X., Z. Wu, J. Yu, "Design and Implementation of a Robotic Shark with a Novel Embedded Vision System: Semantic Scholar", 1 Jan. 1970, www.semanticscholar.org/paper/Design-and-implementation-of-a-robotic-shark-with-a-Yang-Wu/40a662d2a739bd7182b6377663420c8556639ab1.

[WIKI20] Wikipedia, "Propeller", Wikimedia Foundation, 11 Apr. 2020. https://en.wikipedia.org/wiki/Propeller#Types_of_marine_propellers

[THD_19] The Home Depot (THD), "Nature Power. 90-Watt Monocrystalline Solar Panel", 2019. https://www.homedepot.com/p/NATURE-POWER-90-Watt-Monocrystalline-Solar-Panel-50092/206344353

[SST_19] Shenzhen Sanyifeida Technology Co. (SST), "High Efficiency Semi Flexible Solar Panels", 2019. http://www.syfd-solar.com/en/products.aspx?ProductsCateID=120&CateID=120

[BURD20] Burden, T., "Do-It-Yourself: Charging With Solar Panels", West Marine, 2020. www.westmarine.com/WestAdvisor/DIY-Charging-With-Solar-Panels.

[ASIR19] Asira, F., "Solar Battery Charging Basics: Use a Solar Panel to Charge Your Battery", Green Coast, 9 October 2019. greencoast.org/solar-battery-charging-basics/.

[ES__20] EnergySage, "How to Choose the Best Battery for a Solar Energy System", 15 May 2020. www.energysage.com/solar/solar-energy-storage/what-are-the-best-batteries-for-solar-panels/.

[BDN_08] Boat Design Net, "Paddlewheel Discussion", 13 Dec. 2008. www.boatdesign.net/threads/paddlewheel.25292/.

[PZE_20] Power Zone Equipment Inc (PZE), "Centrifugal Pumps – Working, Applications & Types", 2020. https://www.powerzone.com/resources/glossary/centrifugal-pump

[ZORO20] Zoro Tools Inc, "Bilge Pump, Automatic Operation, 12V, 3.7A", 2020. https://www.zoro.com/rule-bilge-pump-abs-nylon-ss-12vdc-1-27sa/i/G2044791/#specifications

[AHMA19] Ahmari, H., S. M. Imran Kabir, "EXPERIMENT #10: PUMPS", Applied Fluid Mechanics Lab Manual, Mavs Open Press, UTA Libraries, 2019. https://uta.pressbooks.pub/appliedfluidmechanics/chapter/experiment-10/

[GCC_19] GrabCad Community, "Propeller", 2019. https://grabcad.com/library/propeller-337

[JOHN__] Johnson, L., "BATTERY CHARGING TUTORIAL", Battery Charging Tutorial | ChargingChargers.com. www.chargingchargers.com/tutorials/charging.html.

[UN__20] United Nations, "Sustainable Development Goals (SDGs)", April 2020. https://www.un.org/sustainabledevelopment/sustainable-development-goals/

13. Appendices

13.1. Sustainable Development Goals (SDGs)

The Sustainable Development Goals (SDGs) have been created to achieve a more sustainable and better world for all [UN__20]. In total, there are 17 goals, two of which are directly related to the topic of this project. The main goal related to this project belongs to the biosphere dimension, whereas the secondary goal belongs to the society dimension.

The primary SDG affected by the project is goal no. 6: Clear Water and Sanitation. The goal of this SDG is to achieve clean, accessible water in every part of the world, as many people die every year from diseases related with inadequate water supply, sanitation, and hygiene. Worldwide, one in three individuals are affected by water scarcity, and at least 10% of the human population consumes food that is irrigated with wastewater containing dangerous chemicals or harmful bacteria [HERO19].

Because of this unfortunate reality, it is essential to engineer solutions for the betterment of mankind. Although there is no defined method for doing so across the globe, it is necessary to start with a small-scale issue. In the state of Louisiana, over one-third of the land is covered by water resources [RUTH15]. All this water is utilized for recreation, human consumption, crop irrigation, and storage. As mentioned earlier, water is an incredibly important and versatile resource. It is necessary to monitor and protect water resources from any depletion or contamination for the betterment of those who depend on it.

In the context of this project, a body of water contaminated by nutrient runoff, which is commonly a result of excessive fertilizer usage by humans in coastal regions, will saturate the water's nitrogen and phosphorus levels. Excess nitrogen and phosphorus cause an overgrowth of algae in a short period of time, also called algal blooms. Harmful algal blooms are mainly the result of a type of algae called cyanobacteria. If a body of stagnant water contains high levels of nutrients, cyanobacteria will quickly grow and degrade the water quality [TCHO12]. If left unchecked, cyanobacteria pose a big problem in reservoirs because it damages the local ecosystem, disrupts water purification processes, and produces toxins that affect humans [EPA_17].

With that in mind, the objective for this project is to prototype a platform for measuring water parameters, as the customer requires an efficient method to monitor Louisiana reservoirs. Thanks to this project, with the device made by the team, early toxins due to cyanobacteria in water reservoirs can be detected so that water can be treated as soon as possible to avoid its contamination. Therefore, this device will help keep the water clean and available for humans to drink.

A secondary SDG related to this project is goal no. 7: Affordable and Clean Energy. The goal of this SDG is to achieve universal access to energy, increase energy efficiency and increase the use of renewable energy in order to reduce environmental issues like climate change. In this project, to recharge the batteries which power both the propulsion system and the sensor deployment mechanism of the device, the team implemented a solar panel. With it, not only the customer will not waste time charging the batteries outside the water but will also recharge them in an ecological way, without harming the environment.

In the following table (Table 15) it is shown each SDG with its respective dimension, role and goal in the project.

SDG dimension	SDG identified	Role	Goal
Biosphere	SDG 6: Ensure access to water and sanitation for all	Primary	By 2030, improve water quality by reducing pollution, eliminating dumping, and minimizing release of hazardous chemicals and materials, halving the proportion of untreated wastewater, and substantially increasing recycling and safe reuse globally.
Society	SDG 7: Ensure access to affordable, reliable, sustainable, and modern energy	Secondary	By 2030, expand infrastructure and upgrade technology for supplying modern and sustainable energy services for all in developing countries, in particular least developed countries, small island developing States, and land-locked developing countries, in accordance with their respective programs of support.

Table (15). *SDGs related to the project.*

13.1.1. Quantification of the impact of the contribution to the SDGs

The main aim of this project is focused on the idea of goal no.6: Clean Water and Sanitation. The main purpose of the construction of this device is to help detect toxins in the water so that it can be treated as early as possible and therefore keep the water clean.

The customer, Dr. María Teresa Gutiérrez-Wing, focuses on water and wastewater treatment, currently performs tests on water samples that are collected near the shore. Although she is able to verify the presence of cyanobacteria, her samples are inaccurate representations of the ecology in the center of reservoirs. Also, as she measures water parameters manually, her methods are extremely inefficient, as she obtains one reading of water parameters each hour (1 rdg/1 hour). Currently, she walks along the shore to a specific point, deploys the sensor to an approximate depth and measures water parameters. Then, she connects the sensor to the computer and reads the data, and so on. She would like to retrieve more data at a faster rate, at a rate of one reading each 5 minutes (1 rdg/5 min) for different depths of water over a period of at least 6 hours.

Thanks to the controller the team built to control the platform and the sensor deployment mechanism, the customer can control the direction of the platform, moving it to the desired place of the water reservoir, and send a command to deploy the temperature sensor to the desired depth up to 5 ft (where algal tend to concentrate, and therefore cyanobacteria). She can also choose how long the sensor can be measuring data. For the amount of data storing time, the device can store data into an SD card at a rate of one measurement per millisecond and transmit data wirelessly every thirty seconds. However, this interval is adjustable in the hardwired code, and can be lowered at a minimum of two and a half seconds.

Therefore, thanks to the device the team built, and assuming it follows the typical run time of the platform recording data, in which it moves for 1 minute and stops and records data for 6 minutes, the customer can receive 10 readings of water parameters in 7 minutes (in the 7 min process of the platform moving to the desired spot and stop to measure); this is, 10 readings in a single process of data collection. This process, shown in Figure 61, is repeated during the operation time of the platform in a day. In an hour, the device can transmit up to 85 readings, 85 times more than manually testing. However, if the customer wishes, she could receive data wirelessly even at a faster rate, as the device can transmit data at least every 2.5 seconds. The team considered that transmitting data every 30 seconds was a reasonable and sufficient time to transmit accurately.

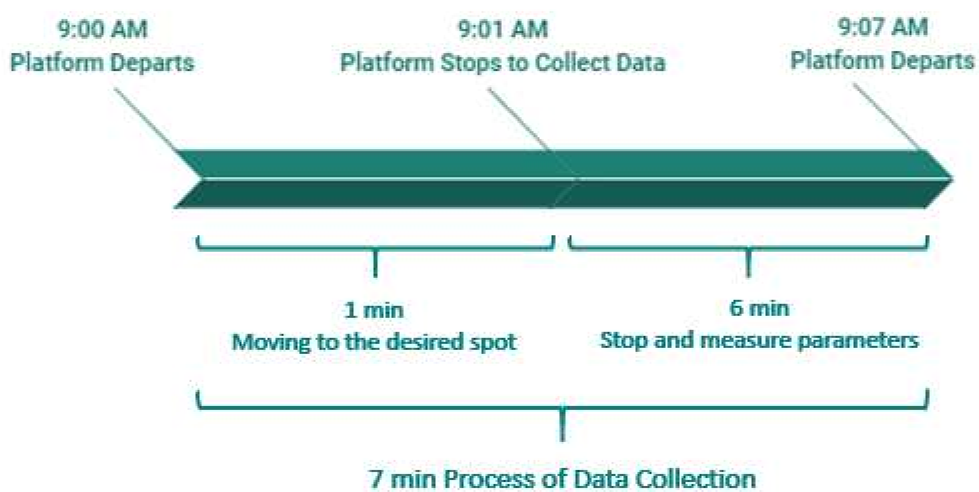


Figure (61). *Process of data collection of the platform.*

Regarding the impact on this project of goal no.7: Affordable and Clean Energy, it mainly affects the achievement of two requirements, more specifically the ones dependent on both the power subsystem and the propulsion system of the device.

One of the requirements is that the platform must be able to run a minimum of 6 hours. However, it is known that realistically the platform is not going to run continuously for 6 hours. An important aspect to consider is the actual run time, as mentioned above, so a hypothetical format of a typical run time of the platform recording data was created. This typical run time includes the 7 min process of data collection where the platform would move for 1 min and then stop and record data for 6 min (5 min to process the sensor measurement and 1 min to stop, drop and reel up the sensor). With this more realistic run time, the hydro pumps and propeller system could run for 6.48 hours and 3.6 hours respectively (Appendix 13.3.6), as the hydro pumps are slower and require less energy than the propellers.

The second requirement is the minimum distance traveled by the device of 2 mi/day. With the more realistic run time of the device, with the hydro pumps as the propulsion system the platform can travel 0.43 miles until the battery discharges in a day, while with the propellers it can travel 1.45 miles.

However, this changes when adding a solar panel that recharges the battery powering the propulsion system. The solar panel is capable of recharging the battery at a faster rate than the battery discharges (as shown in the calculations in Appendix 13.3.6.), so it could run endlessly as long as the sun is present. Assuming that there are 5 hours of useful sunlight in Louisiana, in the water reservoir where our platform will be operating in first place, the platform could theoretically run for 11.48 hours a day with the hydro pumps as the propulsion system, traveling 0.76 miles, and for 8.6 hours with the propeller system, traveling 3.46 miles. All these results are shown in the table below (Table 16).

	Typical Run Time		Typical Run Time with Solar Panel	
	Battery Run Time (hrs)	Distance Traveled (mi)	Battery Run Time (hrs)	Distance Traveled (mi)
Hydro Pumps	6.48	0.43	11.48	0.76
Propellers	3.6	1.45	8.6	3.46

Table (16). *Results considering the typical run time of the device in a day.*

The addition of a solar energy recharging component considerably improves the operation time of the device. Thanks to it, the user has more time to measure water parameters without having to take the device out of the water to recharge the batteries. Furthermore, it allows the user to recharge them in an ecological way without harming the environment, since it uses renewable energy.

Assuming there are 5 hours of useful sunlight in the water reservoir in Louisiana, the platform will be running 77% more using the solar panel to recharge the batteries, therefore traveling 77% more. Following the typical run time, it can repeat the 7 min process of data collection 43 times more in a day. In the event of using hydro pumps as the propulsion system, without the recharging component, the platform could repeat the process of data collection 55 times, whereas with the recharging component it could repeat it 98 times. If using propellers as the propulsion system, without the recharging component the platform could repeat the process of data collection 30 times, whereas with the recharging component it could repeat it 73 times.

This can be linked to what it is mentioned above; if the platform can stop and measure water parameters as long as there is sunlight, thanks to the solar panel the device can do 430 more readings of water parameters (assuming there are 5 hours of useful sunlight), as in each process of data collection the platform transmits 10 readings. This confirms the main purpose of this project: obtaining as much information as possible about the water parameters in order to detect if there are toxins in the water, and if they are treat the water as soon as possible to avoid its contamination.

In this project, the use of this device without a solar panel recharging the batteries would have had great negative impact on its capabilities. Not adding a solar panel would have implied a large and heavier battery with larger capacity in order to meet those requirements, and consequently it would have needed a bigger platform, failing to meet the "the smaller and lighter, the better" requirement. In addition, the device would have been less efficient, as it would have travelled less distance in the water reservoir and consequently it would have recorded less readings of water parameters.

Therefore, the impact of this device on data collection is astonishing. When manually testing, the customer obtained 1 reading of water parameters in an hour. Now, thanks to this device, and following the platform's operation time in a day, with hydro pumps the platform can be running for 11.48 hours (assuming there are 5 hours of useful sunlight in Louisiana's water reservoir), transmitting 980 readings in a day. This is 90 times the amount of data manually collected in a day. With propellers, the platform will be operating less time, but areas further away from each other in the water reservoir can be investigated. With the recharging component, 730 readings can be obtained when using propellers in 8.6 hours.

In the following tables (Table 17 and 18), all this information is shown.

Hydro Pumps	Without Solar Panel	With Solar Panel
Operation Time	6.48 hours	11.48 hours
No. of readings platform	550 rdgs	980 rdgs
No. of readings manually	6 rdgs	11 rdgs

Table (17). *Impact of the hydro pump-driven platform on data collection.*

Propellers	Without Solar Panel	With Solar Panel
Operation Time	3.6 hours	8.6 hours
No. of readings platform	300 rdgs	730 rdgs
No. of readings manually	3 rdgs	8 rdgs

Table (18). *Impact of the propeller-driven platform on data collection.*

13.2. Bill of Materials

Item	Quantity	Count	Unit cost	Cost
Power Components				
Solar Charge Controller	1	1	$10.92	$10.92
Portable Battery Changer	1	1	$15.92	$15.92
Solar Panel	1	1	$29.98	$29.98
SLA Rechargeable Battery 12V, 9 Ah	1	1	$19.95	$19.95
Electrical Components				
Push Button Switch Attachments	1	20	$14.99	$14.99
Arduino Uno R3	2	1	$23.00	$46.00
Ardunio XBee Shield v2.1	2	1	$10.50	$21.00
Arduino Relay Shield v3	1	1	$20.00	$20.00
Waterproof Junction Box	1	1	$28.14	$28.14
MultiStar 5200mAh	1	1	$13.00	$13.00
MicroSD card breakout board+	1	1	$7.50	$7.50
SD/MicroSDMemory Card (8 GB SDHC)	1	1	$9.95	$9.95
Gear Motor w/ Encoder (12 V 16.7RPM)	1	1	$18.50	$18.50
LCD Displays	1	2	$5.88	$5.88
Motor Drive Controller	1	1	$2.29	$2.29
Lora Module Antenna	2	1	$19.50	$39.00
Temperature sensor	1	1	$7.99	$7.99
GPS controller module	1	1	$13.49	$13.49
Heat-Shrink Cable Splitter	1	1	$18.36	$18.36
Mechanical Components				
3D printer Stepper Motor with 300mm Lead Screw	1	1	$36.00	$36.00
8mm Diameter, 330mm Length Shaft	1	1	$5.33	$5.33
4-40 .5in slotted flat head screws(~16 used)	1	100	$4.36	$4.36
0.25in diameter 7ft long cable	1	5	$8.15	$8.15
1ftx2ft Clear Acrylic	1	1	$18.49	$18.49

8inx1ft Black Acrylic	1	1	$11.49	$11.49
2inx48inx8ft waterproof foam board	1	1	$34.95	$34.95
3/8 in. -16x6-1/2 in. Zinc Plated Carriage Bolt	4	1	$1.38	$5.52
Bearings	1	4	$12.77	$12.77
Zinc Yellow-Chromate Plated Hex Head Screw	1	5	$14.02	$14.02
17-7 PH Stainless Steel Washer	1	5	$13.12	$13.12
Medium-Strength Steel Hex Nut	1	100	$8.79	$8.79
Black-Oxide Alloy Steel Socket Head Screw	1	10	$5.77	$5.77
316 Stainless Steel Washer	1	100	$7.11	$7.11
Hydro Pump Bilge Model 27D	2	1	$53.47	$106.94
Misc				
Paint Brush	3	1	$1.99	$5.97
Paint	1	1	$7.99	$7.99
Waterproof Electrical Tape	1	1	$4.79	$4.79
				Total Cost
				$654.42

Table (19). *Bill of Materials.*

13.3. Calculations

13.3.1. Pump Performance Curve

The Head Pressure - Flow Rate curve (Graph 1) was obtained looking at the specifications that appeared in the hydro pump's online website. It specified certain flow rates given different head pressure values.

However, the efficiency curve was more complex to obtain. The efficiency of the hydro pump is measured with the following equation:

Efficiency of the hydro pump: $\eta = \frac{Worf\ of\ the\ pump}{Work\ of\ the\ motor} = \frac{\rho_{fluid} \times g \times h \times Q}{44.4}$ (1)

where:

- $\rho_{fluid} = \rho_{water} = 997\ kg/m^3$
- $g = 9.81\ m/s^2$ (gravity)
- h is the head pressure at the exit of the nozzle (in m)
- Q is the flow rate at the exit of the nozzle (in m^3/s)

The work of the motor was calculated multiplying the voltage (12 V) times the current (3.7 amps) of the hydro pump's motor: $Work\ of\ the\ motor\ =\ 12 \times 3.7 = 44.4\ W$

The most efficient point in the pump performance curve (Graph 1) was obtained by looking at the maximum point of the efficiency curve. That peak is equivalent to the maximum efficiency flow rate point of the hydro pump.

13.3.2. Platform Design

Buoyancy equation: $B = \rho_{fluid} \times g \times V$ (2)

where:

- B is the buoyant force (in newtons, N)
- ρ_{fluid} is the density of the fluid the object is immersed in (in kg/m^3)
- g is gravity (in m/s^2)
- V is the volume of displaced liquid (in m^3)

In order to make the platform float: $B \geq W$ (3)

where:

- W is the weight of the platform (in newtons, N)

Consider the worst case scenario, when the hulls are completely submerged in water. In this scenario, when the water reaches the base of the platform there is risk that the elements being carried by the platform can get wet.

$$V = 2 \times (volume\ of\ one\ hull) \tag{4}$$

The hulls have a unique shape as they were carved in the front in order to achieve a more streamlined shape, so an estimation of its shape was made. This estimate can be seen in Figure 62 as V1. The team made an underestimation of the shape of the hull to consider the worst case scenario. Each hull is 3.6 in thick.

$$Volume\ of\ one\ hull \simeq V_1 + V_2 \tag{5}$$

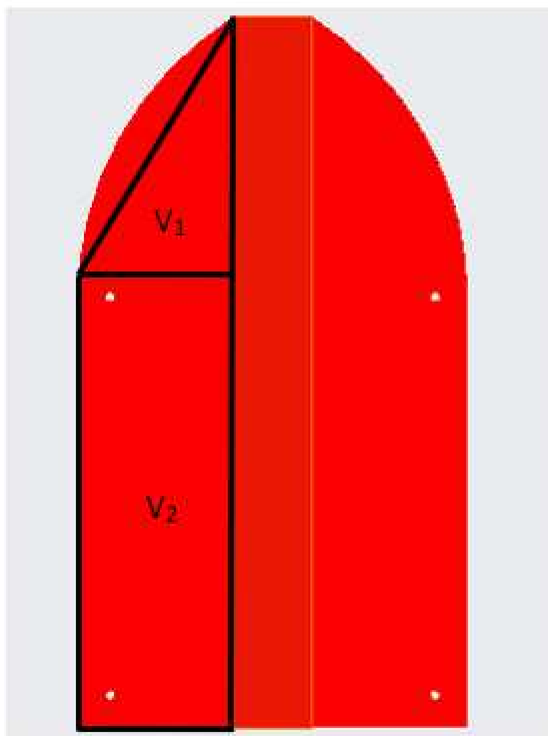


Figure (62). *Estimation of the volume of the hull for a worse case scenario.*

$$V_1 = \frac{7.2" \times 13"}{2} \times 3.6" = 168.48\ in^3$$

$$V_2 = 7.2" \times 20" \times 3.6" = 518.4\ \ in^3$$

Substituting in equation (4):

$$V = 2 \times (V_1 + V_2) = 1373.76\ in^3 = 22511.189\ cm^3 = 0.022512\ m^3$$

Knowing $\rho_{liq} = \rho_{water} = 997\ kg/m^3$ and $g = 9.81\ m/s^2$, substituting in equation (2):

$$B = 997 \times 9.81 \times 0.022512 = 220.18\ N$$

In the worst case, $B = W = 220.18\ N.$ (using equation (3)). If the weight of the device is bigger than the worst case scenario buoyancy force, the device will sink.

With this information, the maximum mass that the platform can carry without sinking can be calculated.

Weight of the whole device:

$$W = weight\ platform + weight\ carried$$

$$W = (\rho_{platform} \times V_{platform} \times g) + (mass\ carried \times g) \tag{6}$$

where:

- $\rho_{platform}$ is the density of the platform, which is made of foam (in kg/m^3)
- $V_{platform}$ is the volume of the platform (in m^3)
- $mass\ carried$ is the total mass the platform carries on top (in kg)

The density of the foam was calculated with the information given in the specifications of the waterproof foam board bought online. Each board is 7.5 lb (453.6 g) and with dimensions 8 ft x 48 in x 2 in (243.8 x 121.9 x 5.1 cm).

$$\rho_{foam} = \frac{453.6\ g}{243.8 \times 121.9 \times 5.1\ cm^3} = 3 \times 10^{-3} g/cm^3 \tag{7}$$

$$\rho_{platform} = \rho_{foam} = 3 \times 10^{-3} g/cm^3 = 3\ kg/m^3$$

$$V_{platform} = 2 \times (volume\ hull) + volume\ base \tag{8}$$

$$V_{platform} = 2 \times (V_1 + V_2) + (33" \times 18" \times 1.8") = 2442.96\ in^3 = 0.04003\ m^3$$

So, in the worst case scenario, the maximum mass that can be carried by the platform is (substituting in equation (6)):

$$220.18 = (3 \times 0.04003 \times 9.81) + (mass\ carried \times 9.81)$$

$$Maximum\ mass\ carried = 22.31\ kg = 49.19\ lbs$$

As the platform has to carry 16 kg (35 lbs), it has been made sure that it will float without the hulls being completely submerged, and with this design of the platform, the worst case scenario will never be reached.

Knowing this information, it can be proceeded to calculate how much the hulls will be submerged in the water for the desired weight.

Desired weight:

$$W = weight\ platform + weight\ carried$$

$$W = (\rho_{platform} \times V_{platform} \times g) + (mass\ carried \times g) \tag{9}$$

where:

- $\rho_{platform}$ is the density of the platform, which is made of foam (in kg/m^3)
- $V_{platform}$ is the volume of the platform (m^3)
- $mass\ carried = 35\ lbs = 15.9kg$

$$\rho_{platform} = \rho_{foam} = 3 \times 10^{-3} g/cm^3 = 3\ kg/m^3$$

$$V_{platform} = 2 \times (volume\ hull) + volume\ base = 2 \times (V_1 + V_2) + (33" \times 18" \times 1.8") =$$

$$= 2442.96\ in^3 = 0.04003\ m^3$$

Substituting in equation (9):

$$W = (3 \times 0.04003 \times 9.81) + (15.9 \times 9.81) = 157.16\ N$$

As $B \geq W$:

$$157.16 = 997 \times 9.81 \times V$$

$$V = 0.01607\ m^3 = 980.65\ in^3$$

$$V = 980.65\ in^3 = 2 \times (volume\ of\ one\ hull) = 2 \times (\frac{7.2" \times 13"}{2} \times x + 20" \times 7.2" \times x)$$

$$x = 2.57\ in$$

So, when carrying the desired weight of 35 lbs, the hulls will be submerged 2.6 in. That is, 72.22% of the total thickness of the hull will be submerged in water.

13.3.3. Hydro Pumps Simulation: Thrust Calculations

The first iteration of the simulation is done in a special way that differs a little from the rest of iterations.

For the first iteration, it is assumed an arbitrary value for the flow rate of the pump which will be used to obtain both the thrust force of the pumps and the speed of the platform. Once known the speed of the platform, the drag force is obtained with the simulation, and will be compared to the thrust force.

If both values are not the same, a second iteration will be done. It will start with the value of drag force obtained in the previous iteration as the new thrust force value. Knowing the thrust force, the speed of the flow out of the pump will be known too, and with that, the speed of the platform. After running the simulation, a different drag force will be obtained. This process is to be repeated until the drag force and the thrust force are the same.

13.3.3.1. The first iteration

The initial value of the flow rate for the simulation was chosen with respect to the pump's performance curve (Graph 1). The graph shows the point where the pump acts in its most efficient way, that is, at a flow rate of 0.6 L/s, which is equivalent to 0.0006107 m^3/s.

Flow rate equation: $Q_{pump} = V_{pump} \times A_{pump}$ (10)

where:

- Q_{pump} is the flow rate coming out of the nozzle (in m^3/s)
- V_{pump} is the speed of the flow coming out of the nozzle (in m/s)
- A_{pump} is the area of the nozzle (in m^2)

$$A_{pump} = \frac{\pi \times (Diameter_{nozzle})^2}{4} = \frac{\pi \times (1.125")^2}{4} = \frac{\pi \times (0.02875\ m)^2}{4} = 0.0006413\ m^2$$

The speed of the flow coming out of the nozzle of the pump at the most efficient point is, substituting in equation (10):

$$0.0006107 = V_{pump} \times 0.0006413$$

$$V_{pump} = 0.95228\ m/s$$

As the volume of water displaced by both pumps is the same as the volume of water displaced by the platform, the speed of the platform can be calculated with the following equation:

$$2 \times (V_{pump} \times A_{pump}) = V_{platform} \times A_{front} \quad (11)$$

where:

- Volume of water displaced by both pumps = $2 \times (V_{pump} \times A_{pump})$
- Volume of water displaced by the platform = $V_{platform} \times A_{front}$

The area at the front is the submerged front area of the platform in the water, that is, for the desired scenario:

$$A_{front\ submerged} = 0.0116\ m^2$$

So the speed of the platform is, substituting in equation (11):

$$2 \times (0.95228 \times 0.0006413\ ) = V_{platform} \times 0.0116$$

$$V_{platform} = 0.105\ m/s$$

This is the estimated velocity with which the first simulation will be carried out. With this velocity, the drag force obtained in the simulation is:

$$F_{drag} = 0.1301\ N$$

Now the drag force is to be compared to the thrust force made by the two pumps.

Thrust force equation of one pump: $F_{thrust} = P \times A_{pump}$ (12)

where:

- F_{thrust} is the thrust force done by one pump (in N)
- P is the pressure at the exit of the nozzle (in Pa)
- A_{pump} is the area of the nozzle (in m^2)

The pressure at the exit of the nozzle is related to the head pressure at the exit of the nozzle. The head pressure is related with the flow rate as shown in Graph 1.

$$Q = 0.0006107\ m^3/s\ \ {-}{>}\ \ H = 1.646\ m$$

Pressure at the exit of the nozzle equation: $P = \rho_{fluid} \times g \times h$ (13)

where:

- $\rho_{fluid} = \rho_{water} = 997\ kg/m^3$
- $g = 9.81\ m/s^2$ (gravity)

- h is the head pressure at the exit of the nozzle (in m)

$$P = \rho_{fluid} \times g \times h = 997 \times 9.81 \times 1.646 = 16096.86\ Pa$$

So the thrust force made by one pump is (substituting in equation (12)):

$$F_{thrust} = P \times A_{pump} = 16096.86 \times 0.0006413 = 10.323\ N$$

As mentioned before, the drag force has to be equal to the thrust force made by the two pumps:

$$F_{drag} = 2 \times F_{thrust} \tag{14}$$

In the first iteration, substituting in equation (14): $0.1301 \neq 2 \times 10.323$

As both forces are not the same, a second iteration has to be done.

13.3.3.2. Procedure for the following iterations

For the next iteration, assume the drag force obtained in the previous iteration as the new thrust force made by the two pumps:

$$F_{drag\ prev.iteration} = 2 \times F_{thrust\ new\ iteration} \tag{15}$$

$$0.1301 = 2 \times F_{thrust\ new\ iteration}$$

Knowing the thrust force made by one pump, the pressure at the exit of the nozzle of the pump is also known (with equation (12)), and with it the head pressure (equation (13)). The head pressure relates with the flow rate at the exit of the nozzle as shown in Graph 1, as mentioned before, and with the flow rate, the speed of the flow coming out of the pump is also known (equation (10)).

With the equation (11), the speed of the platform is known. This will be the estimated flow velocity introduced in the simulation, which will give its respective drag force. This drag force will be compared with the thrust force assumed in this iteration. If they are different, this process will be repeated until their values match.

13.3.4. Propeller Simulation: Propeller and Motor Thrust Calculations:

Initial Values:

Propeller Initial Values:

- Height of propeller [m] = 0.0135
- Diameter of propeller [m] = 0.06
- Degree/propeller = 90

Motor Initial Values:

- KV [rpm/V] = 350
- Thrust (at 12V and 6A) [kg] = 1.12

13.3.4.1. Motor Force Calculation

$$\begin{aligned} \text{Force} &= thrust\ [kg]\ *\ gravity\ [m/s^s] \\ &= 1.12\ [kg]\ *\ 9.81\ [m/s^2] \\ &= 10.98\ [N] \text{ per motor} \\ &= 21.9\ [N] \text{ for two motors} \end{aligned} \quad (16)$$

13.3.4.2. Sample Coefficient of Drag Calculation

Initial Values

- Wet Area ~ approximate frontal area of hull
  $\sim 5.4\ [in]\ *\ 7.2\ [in]$
  $\sim 77.76\ [in^2]$
  $\sim 0.05016\ [m^2]$
- Density of water = $997\ [kg/m^3]$

Calculation:

- Drag [N] = $\frac{1}{2} * coefficient\ of\ drag * wet\ area * density\ * velocity^2$ (17)
  Rearrange the equation to solve for the coefficient of drag. Values used for drag and velocity are from the last row of Table 9.

- Coefficient of Drag (Cd)
  $= (2 * 18.1\ [N])/(0.05\ [m^2] * 997\ [kg/m^3] * 1.2\ [m/s]) = 0.502$

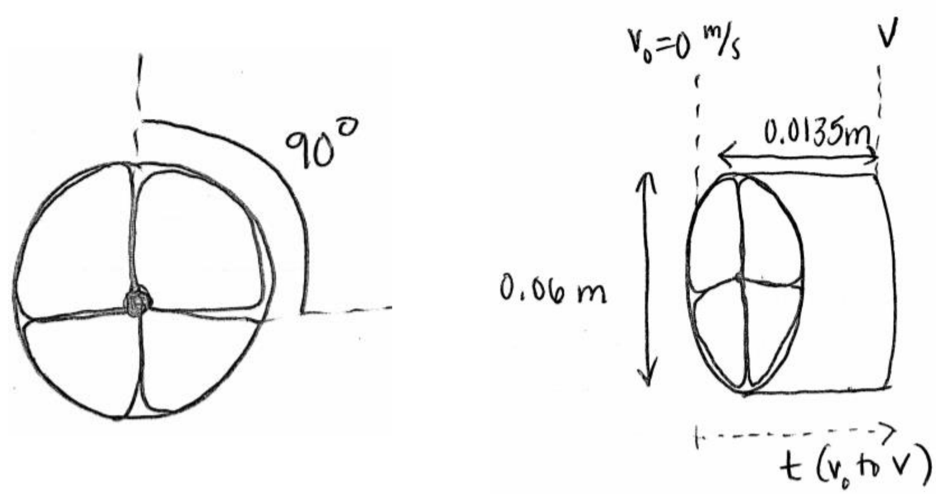


Figure (63). *Drawing of propeller dimensions.*

13.3.5. Battery Lifetime Calculation

Peukert's Battery Discharge Law: $t = H(\frac{C}{IH})^k$ (18)

where:

- t is the battery lifetime (in hours)
- C is the battery capacity (in Amp-Hours or Milliamp-Hours)
- I is the discharge current (in Amps)
- H is the time at which the capacity is evaluated at (in hours)
- k is the Peukert constant (depends on battery type)

Our Battery Specifications: k = 1.1, H = 20 hours, C =9 Ah

For I = 7.04 A at 12 V (hydro pumps)

$$(20\ hr)(\frac{9\ Ah}{(7.04\ A)(20\ hr)})^{1.1} = t = 0.97\ hr = 58.3\ minutes$$

For I = 0.96 A at 2.1 V

$$(20\ hr)(\frac{9\ Ah}{(7.04\ A)(20\ hr)})^{1.1} = t = 8.69\ hr$$

For I = 12 A at 12 V (propellers)

$$(20\ hr)(\frac{9\ Ah}{(12\ A)(20\ hr)})^{1.1} = t = 0.54\ hr = 32.4\ minutes$$

13.3.6. A one day story of the boat: Typical Run Time

Calculation of run time and covered distance of the boat during 1 day; here the boat is scheduled to run for at least 6 hours.

13.3.6.1. Boat with two hydro pumps functioning at full thrust

In the typical run time scenario, the platform moves for 1 minute, and then stops to measure the water parameters for 6 minutes. This means that the platform would move a total of 9 times in an hour; 9 minutes of movement per hour.

The platform moves for 1 minute at a speed of 0.1968 m/s. So, each time the platform moves it will move a total of 11.808 m or 0.007 miles.

$$0.1968[m/s] * 60\ [s] = 11.808\ [m] = 0.007[mi]\ each\ minute$$
$$11.808\ [m] * 9 = 106.272\ [m] = 0.066\ [mi]\ each\ hour$$
$$106.272\ [m] * \ 6 = \ 637.632\ [m]\ each\ 6\ hours = 0.4\ [mi]$$
$$106.272\ [m]\ *\ 24\ =\ 2550.52\ [m]\ each\ day\ =\ 1.58\ [mi]\ each\ day$$

Using Peukurt's law, the battery would take 58.3 min to fully discharge when the pumps are used at full thrust (See Appendix 13.3.5). And if the battery runs for 9 minutes each hour, then the battery will last for 6.478 hrs.

$$58.3\ [min]\ /\ 9[min/hr] = 6.478\ [hr]$$

Following the typical run time, the distance traveled by the platform until the battery discharges is:

$$0.066\ [mi]\ each\ hour * 6.478\ [hr] = 0.43\ [mi]$$

Finding the solar panel charge rate:

$$10\ [W] * 6\ [hr] = 60\ [W * hr]$$
$$60[W * hr]\ /\ 0.57[A] = 105.263\ [V * hr]$$
$$105.263\ [V * hr]\ /\ 6[hr] = 17.544[V/min]\ /\ 60[min/hr] = 0.292[V/min]$$

So, if 1 minute is 1.724% of the battery time, then the battery will lose 0.207 V each time it moves.

$$1.724\% * 12[V] = 0.207\ [V]$$

But, if the solar panel charges the battery during the 6 minutes stops, the battery could replenish up to 1.752 V at each stop.

$$0.292\ [V/min] * 6[min] = 1.752\ [V]$$

Since 1.752 V is larger than 0.207 V, the platform could theoretically run endlessly while there is sunlight.

If it can be assumed that there are 5 hours of useful sunlight in Louisiana, then the platform could theoretically run for 11.478 hours and move a total of 1219.70 meters, that is 0.76 miles.

$$6.478[hr] + 5[hr] = 11.478\ [hr]$$
$$11.478\ [hr] * 106.272[m/hr] = 1219.70\ [m] = 0.76\ [mi]$$

13.3.6.2. Boat with two propellers functioning at full thrust

In this scenario, the platform moves for 1 minute, and then stops to measure the water parameters for 6 minutes. This means that the platform would move a total of 9 times in an hour; 9 minutes of movement per hour.

The platform moves for 1 minute at a speed of 1.2 m/s. So, each time the platform moves it will move a total of 72 m.

$$1.2[m/s] * 60\ [s] = 72[m] = 0.044\ [mi]\ each\ minute$$
$$72\ [m] * 9 = 648\ [m] = 0.402\ [mi]\ each\ hour$$
$$648\ [m] *\ 6 =\ 3888\ [m]\ each\ 6\ hours\ = 2.42\ [mi]$$
$$648\ [m]\ *\ 24\ =\ 15{,}552\ [m]\ each\ day\ = 9.66\ [mi]\ each\ day$$

Using Peukurt's law, the battery would take 32.6 min to fully discharge when the pumps are used at full thrust (See appendix_). And if the battery runs for 9 minutes each hour, then the battery will last for 3.6 hrs.

$$32.6\ [min]\ /\ 9[min/hr] = 3.6\ [hr]$$

Following the typical run time, the distance traveled by the platform until the battery discharges is:

$$0.402\ [mi]\ each\ hour * 3.6\ [hr] = 1.45\ [mi]$$

Finding the solar panel charge rate:

$$10\ [W] * 6\ [hr] = 60\ [W * hr]$$
$$60[W * hr]\ /\ 0.57[A] = 105.263\ [V * hr]$$
$$105.263\ [V * hr]\ /\ 6[hr] = 17.544[V/min]\ /\ 60[min/hr] = 0.292[V/min]$$

So, if 1 minute is 3.086% of the battery time, then the battery will lose 0.370 V each time it moves.

$$3.086\% * 12[V] = 0.370\ [V]$$

But, if the solar panel charges the battery during the 6 minutes stops, the battery could replenish up to 1.752 V at each stop.

$$0.292\ [V/min] * 6[min] = 1.752\ [V]$$

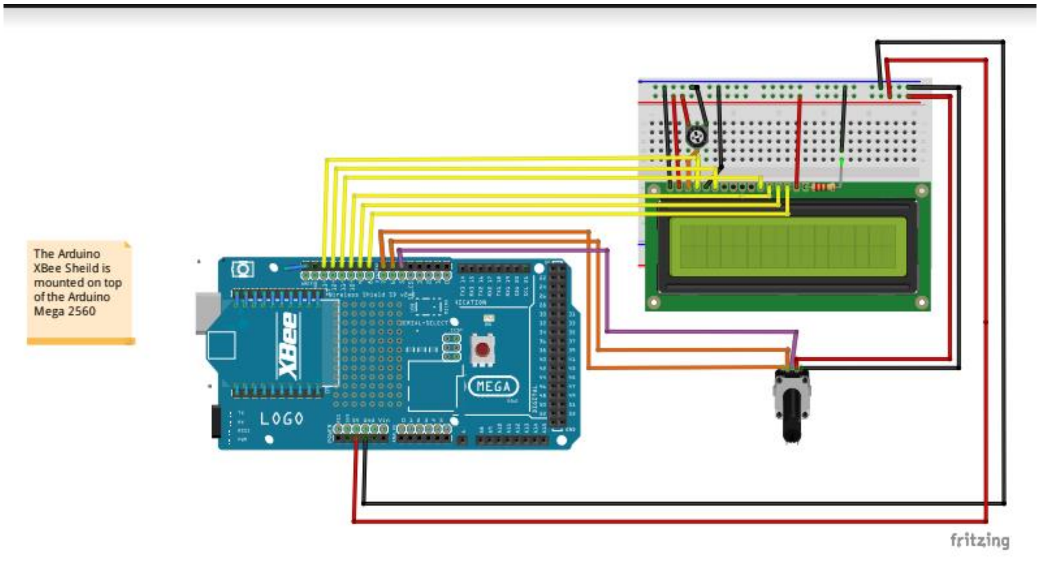
Since 1.752 is larger than 0.370 V, the platform could theoretically run endlessly while there is sunlight.

If it can be assumed that there are 5 hours of useful sunlight in Louisiana, then the platform could theoretically run for 8.6 hours and move a total of 5572.800 meters, that is 3.46 miles.

$$3.6\ [hr] + 5\ [hr] = 8.6\ [hr]$$
$$8.6\ [hr] * 648\ [m/hr] = 5572.800\ [m] = 3.46\ [mi]$$

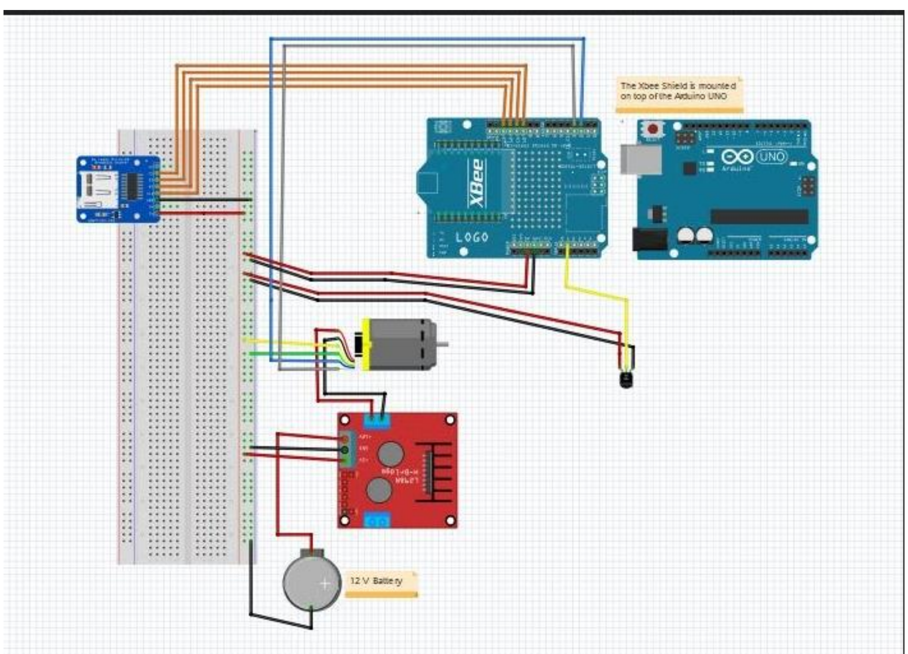
13.4. Electrical Schematic Diagrams

13.4.1. XBee Controller



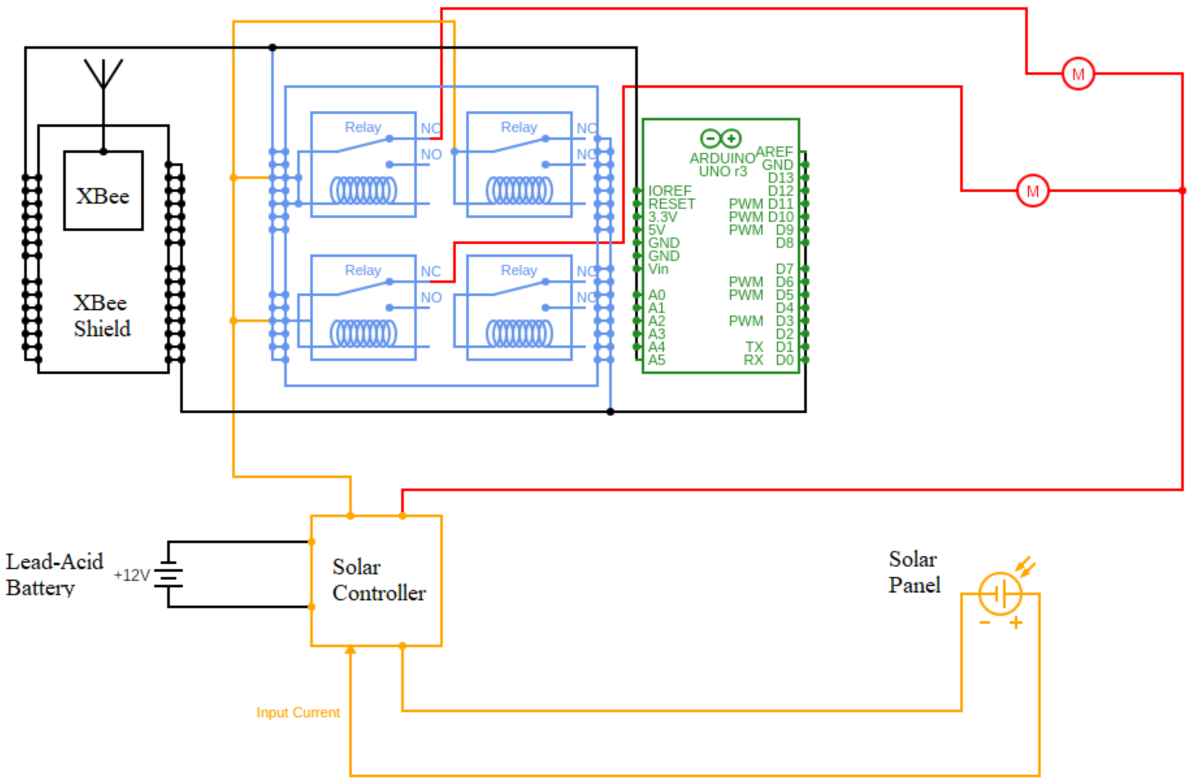
Link: [Xbee Controller]

13.4.2. XBee Platform



Link: [Xbee Platform Receiver]

13.4.3. Solar Charger



13.5. Arduino Codes

LINK : [Arduino Codes]

13.5.1. XBee Range Testing

13.5.1.1. XBee Range Testing - Controller

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
#include <OneButton.h>
SoftwareSerial mySerial(2, 3); // RX, TX
int buttonPin =5 ;
OneButton button0(buttonPin, true);
const int rs = 8, en = 9, d4 = 10, d5 = 11, d6 = 12, d7 = 13;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
String cad;
#define CLK 0
#define DATA 1
#define tempButton  5

void setup() {
 Serial.begin(9600);
 mySerial.begin(9600);
 button0.attachClick(singleClick);
 lcd.begin(16, 2);
 lcd.print("Starting ...");
 lcd.setCursor(0,1);
 lcd.print("Signal Testing");
 pinMode(CLK, INPUT);
 pinMode(CLK, INPUT_PULLUP);
 pinMode(DATA, INPUT);
 pinMode(DATA, INPUT_PULLUP);
}

static uint8_t prevNextCode = 0;
static uint16_t store=0;
static int8_t c,val;

void loop() {
   if( val=read_rotary() ) {
      c +=val;
```

```
     Serial.print("Position: ");
     Serial.print(c);
     lcd.clear();

 if ( prevNextCode==0x0b) {
      Serial.println("CW");
       mySerial.print("1");
       lcd.clear();    lcd.setCursor(0,0);
       lcd.print(">Sending: CW" );
       delay(1);
     }

     if ( prevNextCode==0x07) {
       Serial.println("CCW");
       mySerial.println("2");
       lcd.clear();    lcd.setCursor(0,0);
       lcd.print(">Sending: CCW");
       delay(1);
     }
  }

  button0.tick();

 if (mySerial.available()) {
     cad=mySerial.readString();
     Serial.println(cad);
     lcd.setCursor(12,1);
     lcd.print((char) 223);
     lcd.print("C");
     lcd.setCursor(0,1);
     lcd.print(">Recieved:" + cad);
   }
}

// A vald CW or  CCW move returns 1, invalid returns 0.
int8_t read_rotary() {
  static int8_t rot_enc_table[] = {0,1,1,0,1,0,0,1,1,0,0,1,0,1,1,0};

  prevNextCode <<= 2;
  if (digitalRead(DATA)) prevNextCode |= 0x02;
  if (digitalRead(CLK)) prevNextCode |= 0x01;
  prevNextCode &= 0x0f;
```

```
  // If valid then store as 16 bit data.
  if  (rot_enc_table[prevNextCode] ) {
     store <<= 4;
     store |= prevNextCode;
     //if (store==0xd42b) return 1;
     //if (store==0xe817) return -1;
     if ((store&0xff)==0x2b) return -1;
     if ((store&0xff)==0x17) return 1;
  }
  return 0;
}

void singleClick() {
   mySerial.println("3");
   Serial.println("T");
   lcd.setCursor(0,0);
   lcd.clear();
   lcd.print(">Sending: TEMP? ");
}
```

13.5.1.2. XBee Range Testing - Receiver

```
#include <SoftwareSerial.h>
#include <OneWire.h>
#include <DallasTemperature.h>
SoftwareSerial mySerial(2, 3); // RX, TX
#define ONE_WIRE_BUS A0
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

void setup() {
 Serial.begin(9600);
 mySerial.begin(9600);
 sensors.begin();
}

String cad;
int icad;

void loop() {
```

```
 sensors.requestTemperatures();
 int temp = sensors.getTempCByIndex(0);
 //Serial.println("Temp: "); Serial.print(temp);

 if (mySerial.available()) {
   cad=mySerial.readString();
   Serial.println(cad);
   icad=cad.toInt();

switch (icad) {

   case 1:
     mySerial.println("OK-CW  ");
     break;

   case 2:
     mySerial.println("OK-CCW  ");
     break;

   case 3:
     String Tstring=String(temp)+ String((char) 223) + "C  " ;
     mySerial.println(Tstring);
     break;

   default:
    Serial.println("L");
 }
 }
 else
   icad=0;
   Serial.println(icad);
   delay(500);
}
```

13.5.2. LORA Range Testing

13.5.2.1. LORA Range Testing - Controller

```
#include <LiquidCrystal.h>
#include <OneButton.h>

unsigned long lastTransmission;
const int interval = 1000;

#define ledPin 2
String incomingString;
String PrStr;

void setup(){
 lcd.begin(16, 2);
 lcd.print("Starting ...");
 lcd.setCursor(0,1);
 lcd.print("Signal Testing");
  pinMode(CLK, INPUT);
  pinMode(CLK, INPUT_PULLUP);
  pinMode(DATA, INPUT);
  pinMode(DATA, INPUT_PULLUP);
  Serial.begin(115200);
  pinMode(ledPin,OUTPUT);
}

static uint8_t prevNextCode = 0;
static uint16_t store=0;
static int8_t c,val;

void loop(){
    if (millis() > lastTransmission + interval){
        Serial.println("AT+SEND=0,8,Testing!");
        digitalWrite(ledPin,HIGH);
        delay(1000);
        digitalWrite(ledPin, LOW);
        lastTransmission = millis();

    }
}
```

13.5.2.2. LORA Range Testing - Receiver

```
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 7
#define ledPin 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
String incomingString;
String PrStr;
void setup(){
   Serial.begin(115200);
   sensors.begin();
   pinMode(ledPin,OUTPUT);
}

void loop(){
   if (Serial.available()>0){
      incomingString = Serial.readString();
      Serial.println( incomingString );
      if(incomingString.indexOf("Testing!") >  0){
         digitalWrite(ledPin,HIGH);
         delay(1000);
         digitalWrite(ledPin,LOW);
      }
          }
}
```

13.5.3. XBee Controller Code

```
#include <LiquidCrystal.h>
#include <rotary.h>       // rotary handler
#include <SoftwareSerial.h>
#define CLK 6
#define DATA 7
#define PUSHB A0
#define MAX_WORLD_COUNT 5
#define MIN_WORLD_COUNT 1
SoftwareSerial mySerial(4, 5); // RX, TX
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
```

```
char *Words[MAX_WORLD_COUNT];
char *StringToParse;

// Initialize the Rotary object
// Rotary(Encoder Pin 1, Encoder Pin 2, Button Pin) Attach center to ground
Rotary r = Rotary(CLK, DATA, PUSHB);        // there is no must for using interrupt pins !!

int columnsLCD = 16;
char* MenuLine[] = {" Test Signal", "Automatic Test", " Manual Test", " Home Sensor", " Sweep
Temp"};

int MenuItems = 5;
int CursorLine = 0;

int backlightPin = 10;   //PWM pin
int brightness = 255;
int fadeAmount = 5;

unsigned long startMillis;
unsigned long currentMillis;
const unsigned long period = 100000;  //the value is a number of milliseconds
static int8_t c,val;
float depth_ft; int time_m;

void setup ()
{
 pinMode(CLK, INPUT);
  pinMode(CLK, INPUT_PULLUP);
  pinMode(DATA, INPUT);
  pinMode(DATA, INPUT_PULLUP);
  digitalWrite (PUSHB, HIGH);
  pinMode(backlightPin, OUTPUT);
  digitalWrite(backlightPin, HIGH);
  Serial.begin(9600);
  mySerial.begin(9600);
  lcd.begin (16, 2);
  lcd.setCursor(0,0);
  lcd.print(" Mechanical Engineering Capstone 2020                    ");

  for (int positionCounter = 0; positionCounter < 36; positionCounter++) {
    // scroll one position left:
    lcd.scrollDisplayLeft();
```

```
    // wait a bit:
    delay(270);
}



  lcd.clear (); // go home
  lcd.print ("Sensor Deploy");
  lcd.setCursor(0, 1);
  lcd.print("Please Select");
  delay(3000);
  startMillis = millis();  //initial start time
  print_menu();
} //End of setup



static uint8_t prevNextCode = 0;
static uint16_t store=0;

void loop ()
{


  currentMillis = millis();  //get the current "time" (actually the number of milliseconds since the
program started)
  if (currentMillis - startMillis >= period)  //test whether the period has elapsed
  {
   // lcd.clear (); // go home
   lcd.setCursor(0, 0);
  lcd.print ("  Sensor Deploy        ");
  lcd.setCursor(0, 1);
  lcd.print("  Please Select      ");

  }

  if (val=read_rotary()) {
    init_backlight();
    c +=val;
    CursorLine=c;
    Serial.println(CursorLine);
```

```
   if (CursorLine < 0) { CursorLine = MenuItems - 1; // roll over to last item } if (CursorLine >
MenuItems - 1) {
     CursorLine = 0;
     c=0;              // roll over to first item
   }

   print_menu();
 }

 if (r.buttonPressedReleased(15)) {

  init_backlight();
  Serial.println( CursorLine );
  selection();
 }


if (mySerial.available()) {


String cad=mySerial.readString();

   //Serial.println(cad);
  char Ints[50]; byte word_count;
  cad.toCharArray(Ints,50);
  StringToParse = Ints;
  //Serial.println(StringToParse);
  word_count = split_message(StringToParse);

 for (byte sms_block = 0; sms_block < word_count; sms_block++) {
   Serial.print("Word "); Serial.print(sms_block + 1);  Serial.print(" : ");
   Serial.println(Words[sms_block]);
 }

String Case = String(Words[0]);
String desc = String(Words[1]);
String temp = String(Words[2]);
int icase=Case.toInt();

 switch (icase) {

   case 1:
```

```
  Serial.println(desc);
// delay(1000);
  lcd.setCursor(0, 0);  //(col, row)
  lcd.print("Sending: Hello?            ");
  lcd.setCursor(0, 1);  //(col, row)
  lcd.print("Recieved: Yes           ");
  delay(1500);
  print_menu();
  break;

case 2:
  lcd.setCursor(0, 1);
  lcd.print("                       ");
  lcd.setCursor(0, 0);
  lcd.print("Depth:                ");
  lcd.setCursor(7, 0);
  lcd.print(desc);
  lcd.setCursor(8, 0);
  lcd.print("ft         ");
  lcd.setCursor(0, 1);
  lcd.print("Temp:");
  break;

case 3:
 Serial.println(desc);
 lcd.setCursor(0, 0);  //(col, row)
 lcd.print("Descending                   ");
 lcd.setCursor(0, 1);  //(col, row)
 lcd.print("Please Wait                ");
  break;

case 4:
 Serial.println(desc);
 lcd.setCursor(0, 0);  //(col, row)
 lcd.print("   Storing Data             ");
 lcd.setCursor(0, 1);  //(col, row)
 lcd.print("Temp:                       ");
 delay(100);
  break;

case 5:
 Serial.println(desc);
```

```
 lcd.setCursor(7, 1);  //(col, row)
 lcd.print(desc);
 lcd.setCursor(9, 1);  //(col, row)
 lcd.print(char(223));
 lcd.setCursor(10, 1);  //(col, row)
 lcd.print("C");
 delay(1000);
 lcd.setCursor(7, 1);  //(col, row)
 lcd.print("                ");
 break;


case 6:
 Serial.println(desc);
 lcd.setCursor(0, 0);  //(col, row)
 lcd.print(" Okay, Done        ");
 lcd.setCursor(0, 1);  //(col, row)
 lcd.print("          ");
 delay(3000);
 print_menu();
 break;

 case 7:
 Serial.println(desc);
 lcd.setCursor(0, 0);
 lcd.print("OK Starting            ");
 lcd.setCursor(0, 1);
 lcd.print("                        ");
 delay(2000);
 break;

case 8:
delay(1000);
Serial.println(desc);
lcd.setCursor(0, 0);
lcd.print("                  ");
lcd.setCursor(0, 0);
lcd.print("Returning Home >           ");
lcd.setCursor(0, 1);
lcd.print("                        ");
delay(2000);
break;
```

```
    default:
    Serial.println("L");
  }
    }
}

/************FUNCTIONS**************/

void print_menu()
{
  lcd.clear();
  lcd.setCursor(0, 0);     //(col, row)
  lcd.print("   Main Menu  ");
  lcd.setCursor(0, 1);    //2nd row
  lcd.print("<"); lcd.setCursor(columnsLCD - 1, 1); lcd.print(">");
  lcd.setCursor(1, 1);
  lcd.print(MenuLine[CursorLine]);

}

void selection()
{
  switch (CursorLine) {

    case 0:
    lcd.setCursor(0, 0);  //(col, row)
    lcd.print("Sending "  + String(char(126))+" Hello");
    lcd.setCursor(0, 1);  //(col, row)
    lcd.print("                  ");
    mySerial.println("1");
    delay(1000);
    break;

    case 1:
     lcd.setCursor(0, 0);
     lcd.print("  Auto Enabled   ");
      //set a flag or do something....
     lcd.setCursor(0, 1);  //(col, row)
     lcd.print("Sending "  + String(char(126))+" Auto     ");
     mySerial.println("2");
     break;
```

```
    case 2:
     Serial.println("Manual Chosen");
     ManualT();
      break;

    case 3:
      Serial.println("Homing Sensor    ");
      lcd.setCursor(0, 0);  //(col, row)
      lcd.print("Homing Sensor "+ String(char(126)));
      lcd.setCursor(0, 1);  //(col, row)
      lcd.print("                ");
      mySerial.println("4,Calling Home");
      delay(3000);
      break;


    case 4:
      Serial.println("HCalling Sweep   ");
      lcd.setCursor(0, 0);  //(col, row)
      lcd.print("Sweeping Temp "+String(char(126)));
      lcd.setCursor(0, 1);  //(col, row)
      lcd.print("               ");
      mySerial.println("5,Calling Sweep");
      delay(1000);
      break;

    default:
    Serial.println("L");

    break;
  }

  delay(2000);

  CursorLine = 0;
  c=0;// reset to start position

}

void LCDfadeOut()
{
  while (brightness > 0) {
```

```
    analogWrite(backlightPin, brightness);
    brightness -= fadeAmount;
    delay(20);
  } //End while
  digitalWrite(backlightPin, LOW);
  lcd.clear();
} //End LCDfadeOut

void ManualT(){

     lcd.setCursor(0, 0);
     lcd.print(" Set Depth:        ");
      //set a flag or do something....
     lcd.setCursor(0, 1);  //(col, row)
     lcd.print(" Set Time:        ");
     lcd.setCursor(0, 0);
     lcd.print(">");

        while (!r.buttonPressedReleased(15)){
          CursorLine=0;
//        Serial.println(CursorLine);
//        Serial.println("Insert depth");
              if ( val=read_rotary()) {
                //CursorLine=0;
                c +=val;
                CursorLine=c;
             // Serial.println(CursorLine);
               lcd.setCursor(11, 0);
               lcd.print((CursorLine-2)*0.2);
               depth_ft=(CursorLine-2)*0.2;
               lcd.setCursor(14, 0);
               lcd.print("ft");
   //          Serial.println(depth_ft);
  }
        }
        delay(1000);
        lcd.setCursor(0, 0);
        lcd.print(" ");
        lcd.setCursor(0, 1);
        lcd.print(">");
        c =0;
          while (!r.buttonPressedReleased(15)){
```

```
        CursorLine=0;
            if ( val=read_rotary()) {
              //CursorLine=0;
              c +=val;
              CursorLine=c;
              Serial.println(CursorLine);
             lcd.setCursor(11, 1);
             lcd.print(CursorLine);
             time_m=CursorLine;
             lcd.setCursor(13, 1);
             lcd.print("min");
  //         Serial.println(time_m);
  }
       }
          delay(1000);
          lcd.setCursor(0,0);
          lcd.print(String(char(126)) + String(depth_ft)+"ft for " + String(time_m)+"min");
          lcd.setCursor(0,1);
          lcd.print("Confirm?                        ");

  while (!r.buttonPressedReleased(25)){
          Serial.println(depth_ft);
          Serial.println(time_m);
       }
String info=String(3) + (",")+String(depth_ft)+(",")+String(time_m);
mySerial.println(info);
delay(1000);
}

void init_backlight()
{
  digitalWrite(backlightPin, HIGH);
  startMillis = millis();  //initial start time
  brightness = 255;   //reset to initial brightness
}

// A vald CW or  CCW move returns 1, invalid returns 0.
int8_t read_rotary() {
  static int8_t rot_enc_table[] = {0,1,1,0,1,0,0,1,1,0,0,1,0,1,1,0};

  prevNextCode <<= 2;
  if (digitalRead(DATA)) prevNextCode |= 0x02;
```

```
  if (digitalRead(CLK)) prevNextCode |= 0x01;
  prevNextCode &= 0x0f;

   // If valid then store as 16 bit data.
   if  (rot_enc_table[prevNextCode] ) {
      store <<= 4;
      store |= prevNextCode;
      //if (store==0xd42b) return 1;
      //if (store==0xe817) return -1;
      if ((store&0xff)==0x2b) return -1;
      if ((store&0xff)==0x17) return 1;
   }
   return 0;
}

byte split_message(char* str) {
  byte word_count = 0; //number of words
  char * item = strtok (str, " ,"); //getting first word (uses space & comma as delimeter)
  while (item != NULL) {
    if (word_count >= MAX_WORLD_COUNT) {
      break;
    }
    Words[word_count] = item;
    item = strtok (NULL, " ,"); //getting next word
    word_count++;
  }
  return  word_count;
}
```

13.5.4. XBee Platform Code

```
#include <SoftwareSerial.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
SoftwareSerial mySerial(4, 5); // RX, TX
#define ONE_WIRE_BUS A0
OneWire oneWire(ONE_WIRE_BUS);
```

```
DallasTemperature sensors(&oneWire);
#define ENCODER_A   2
#define ENCODER_B   3
#define MAX_WORLD_COUNT 5
#define MIN_WORLD_COUNT 1
char *Words[MAX_WORLD_COUNT];
char *StringToParse;
unsigned long int TimeDesc;
int v=140;                              // Initial voltage
float kp=10.0E-03, kd=15.0E-03, ki=4.00E-03; // Controller coefficients
int t_wait=200, t_measurement=100;
float PreviousDepth = 0;
float RunDepth = 0;
#define GEARING 270  // Thislet us convert ticks-to-RPM
#define ENCODERMULT 7 //DEPENDENT ON MOTOR
int motorPinC1 = 6; //CONNECTIONS FROM MOTOR TO MOTOR DRIVER
int motorPinC2 = 7;
const int enablePinC= 8;
volatile float RPM = 0;
volatile uint32_t lastA = 0;
volatile bool motordir = true;
unsigned long int t_old=0, t=0, t_start=0;
unsigned long int t_start2 = 0;
float err=0.0, rpm_target=20, err_old=0.0,derr=0.0,err_cum=0.0;
float Spool_C= 1.82*2*3.14159; //inches

File myFile;

void interruptA() {
  motordir = digitalRead(ENCODER_B);
  digitalWrite(LED_BUILTIN, HIGH);
  uint32_t currA = micros();
  if (lastA < currA) {
    // did not wrap around
    float rev = currA - lastA;  // us
    rev = 1.0 / rev;            // rev per us
    rev *= 1000000;             // rev per sec
    rev *= 60;                  // rev per min
    rev /= GEARING;             // account for gear ratio
    rev /= ENCODERMULT;         // account for multiple ticks per rotation
    RPM = rev;
  }
```

```
  lastA = currA;
  digitalWrite(LED_BUILTIN, LOW);
}

void setup() {

Serial.begin(9600);
   if (SD.begin(10)) {
    Serial.println("initialization okay!");
 }
  mySerial.begin(9600);
  sensors.begin();
  pinMode(motorPinC1, OUTPUT);   //L298N Control port settings direction of motor C (originaly
L298P)
  pinMode(motorPinC2, OUTPUT);  //L298N Control port settings direction of motor C
  pinMode(enablePinC, OUTPUT);
  pinMode(ENCODER_B, INPUT_PULLUP);
  pinMode(ENCODER_A, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(ENCODER_A), interruptA, RISING);
}

void printRPM() {

    Serial.print("Direction: ");
    if (motordir) {
      Serial.println("forward @ ");

    } else {
      Serial.println("backward @ ");
    Serial.print(int(RPM));
    Serial.println(" RPM");
    }
    }

void send_voltage(int v){
  analogWrite(enablePinC,v);
  delay(t_wait);
}

void RunReverse(){
   digitalWrite(motorPinC1, LOW);
  digitalWrite(motorPinC2, HIGH);
```

```
}

void RunForward(){
   digitalWrite(motorPinC1, HIGH);
  digitalWrite(motorPinC2, LOW);
}


void runAuto(){
  //Go to 1ft , 2 ft  , 3 ft & Record data for 2.5 minutes
      unsigned long  AutoSenseTime = 0.5*60*1000;
      float depth=12;
      int depthi=12;
      float NumRot = depth/Spool_C;
      unsigned long int TimeDesc = (NumRot/rpm_target)*60*1000; ///in milli seconds

      for (int i=0; i<3; i++){
        Serial.print("i: "); Serial.println(i);
        Serial.print("depthi: "); Serial.println(depthi);

      long int currentT3=millis();

          while (millis() < currentT3 + TimeDesc){
            Serial.println("moving");
            RunForward();
            analogWrite(enablePinC,130);
            RunMotor();

          }

           mySerial.println( "2," + String(depthi/12) );
            mySerial.flush();
            Serial.println("reached it");
            analogWrite(enablePinC,0);
            //delay(1000);
            delay(1000);
            SaveData(AutoSenseTime);
            delay(100);
            depthi=depthi+depth;
            mySerial.println("2");

      }
```

```
  PreviousDepth=3.5*12;
  delay(1000);
  RunDepth=PreviousDepth;
  returnHome();


}


void RunSweep(){

    float depth=6;
    int depthi=6;
    float NumRot = depth/Spool_C;
    unsigned long int TimeDesc = (NumRot/rpm_target)*60*1000; ///in milli seconds

    for (int i=0; i<8; i++){
      Serial.print("i: "); Serial.println(i);
      Serial.print("depthi: "); Serial.println(depthi);

    long int currentT3=millis();

        while (millis() < currentT3 + TimeDesc){
          Serial.println("moving");
          RunForward();
          analogWrite(enablePinC,130);
          RunMotor();

        }

        float dec = float(depthi) / 12.0 ; Serial.println(dec);
         mySerial.println( "2," + String(dec) );
          mySerial.flush();
          Serial.println("reached it");
          analogWrite(enablePinC,0);
          delay(1000);
           int temp = sensors.getTempCByIndex(0);
    // Serial.print("Temp: "); Serial.println(temp);
      String Tstring=String(temp) ;
      Serial.print(Tstring);
      mySerial.println("5,"+ Tstring);
```

```
        mySerial.flush();
        delay(1500);
        depthi=depthi+depth;
        mySerial.println("2");

      }

   PreviousDepth=10*6;
   delay(1000);
   RunDepth=PreviousDepth;
   mySerial.println("6");
}

void returnHome(){

   Serial.println(PreviousDepth);
   float NumRot = PreviousDepth/Spool_C;
   unsigned long int TimeDesc = (NumRot/rpm_target)*60*1000; ///in milli seconds

   long int currentST=millis();
   mySerial.println("8,HOME");

          while (millis() < currentST + TimeDesc){

                analogWrite(enablePinC,255);
                //mySerial.println("2");
                RunReverse();
                RunMotor();
                Serial.print("MOVING UP");
          }

   analogWrite(enablePinC,0);
   delay(1000);
   mySerial.println("6");
   PreviousDepth=0;
   Serial.println(PreviousDepth);
}

unsigned long  itime ;
float idepth;
int FileNumberI = 0;
```

```
void loop() {

 if (mySerial.available()) {

   String cad = mySerial.readString();
   Serial.println(cad);
   //icad=cad.toInt();

  char Ints[50]; byte word_count;
  cad.toCharArray(Ints,50);

  StringToParse = Ints;
  Serial.println(StringToParse);

  word_count = split_message(StringToParse);

 for (byte sms_block = 0; sms_block < word_count; sms_block++) {
   Serial.print("Word "); Serial.print(sms_block + 1);  Serial.print(" : ");
   Serial.println(Words[sms_block]);
 }

String Case = String(Words[0]);
String depth = String(Words[1]);
String tim = String(Words[2]);

idepth = (depth.toFloat())*12 ; //Convert to inches
itime = tim.toInt();
int icase=Case.toInt();

 switch (icase) {

   case 1:
    delay(1000);
    mySerial.println("1,GotIt");
     break;

   case 2:
    delay(100);
    mySerial.println("7,Ready");
    mySerial.flush();
    delay(4000);
    FileNumberI = FileNumberI = + 1;
```

```
    runAuto();
     break;

   case 3:
   FileNumberI = FileNumberI = + 1;
   ManualT();
    break;

    case 4:
    delay(100);
    Serial.println("Requested to be Homed");
    returnHome();
     break;

  case 5:
    delay(100);
    RunSweep();
    delay(100);
    Serial.println("Requested to be Homed");
    returnHome();
     break;

   default:
   Serial.println("L");
   break;
  }
   PreviousDepth = idepth;
   Serial.print("Updated PD:");
   Serial.print(PreviousDepth);
  }

  else
   int    icad=0;
   //Serial.println(icad);
    delay(500);
}

void ManualT(){
        mySerial.println("3,Descent");
      mySerial.flush();
      delay(1000);
```

```
unsigned long  SenseTime = itime*60*1000;
//      Serial.print("Time (min): ");
//      Serial.println(itime);
//      Serial.print("SenseTime (milli s): ");
//      Serial.println(SenseTime);

      if (idepth >= PreviousDepth){
        Serial.print(" idepth: "); Serial.print(idepth);
        Serial.print(" PreviousDepth: ");Serial.println(PreviousDepth);
        RunDepth = idepth - PreviousDepth;
        Serial.print(" RunDepth: "); Serial.println(RunDepth);

         float NumRot = RunDepth/Spool_C;
         Serial.println(NumRot);
         unsigned long int TimeDesc = (NumRot/rpm_target)*60*1000; ///in milli seconds

        //Run the motor forward for the amount of time it takes to descend given depth
          long int currentT2=millis();
          while (millis() < currentT2 + TimeDesc){
            RunForward();
            analogWrite(enablePinC,130);
           // RunMotor();
            }

        delay(1000);
        analogWrite(enablePinC,0);

      }

      else {
     Serial.println("idepth:"); Serial.print(idepth);
        Serial.print("  PreviousDepth:");Serial.println(PreviousDepth);
        RunDepth = PreviousDepth - idepth;
        Serial.println("RunDepth:"); Serial.print(RunDepth);

       float NumRot = RunDepth/Spool_C;
       Serial.println(NumRot);
       unsigned long int TimeDesc = (NumRot/rpm_target)*60*1000; ///in milli seconds
      Serial.println(TimeDesc);

        //Run the motor backward for the amount of time it takes to ascend given depth
          long int currentT3=millis();
```

```
        while (millis()-currentT3 < TimeDesc){
          RunReverse();
          analogWrite(enablePinC,255);
         // RunMotor();
          }

        delay(1000);
        analogWrite(enablePinC,0);
     }

      delay(1000);
      mySerial.println("4,GettingData");
      delay(1000);
      SaveData(SenseTime);
      delay(100);

      mySerial.println("6,Done");
}

byte split_message(char* str) {
  byte word_count = 0; //number of words
  char * item = strtok (str, " ,"); //getting first word (uses space & comma as delimeter)

  while (item != NULL) {
    if (word_count >= MAX_WORLD_COUNT) {
      break;
    }
    Words[word_count] = item;
    item = strtok (NULL, " ,"); //getting subsequence word
    word_count++;
  }
  return  word_count;
}

void RunMotor() {

t_start=micros();
printRPM();
unsigned long t_start=micros();

send_voltage(v);
```

```
  if (RPM==0){
   // Serial.println("Kp="+String(kp*1000)+"E-03, Kd="+String(kd*1000)+"E-03,
Ki="+String(ki*1000)+"E-03");
   // Serial.println("Time [s]\tVoltage[V]\tSpeed[rev/min]");
    delay(3000);

    t_start2=micros();
  }

  t_old=t;
  t=micros();

  // Derivative control
  err_old=err;
  err=rpm_target-RPM;
  derr=(err-err_old)/float(t-t_old)*1E+6;

  // Integral control
  err_cum+=err*float(t-t_old)*1E-6;

 // Serial.println(String((micros()-t_start)*1E-6)+"\t"+String(v)+"\t"+String(RPM));

  v = v +  int(kp*err + kd*derr + ki*err_cum); // Correction = Sum of proportional, derivative and
integral
  if (v>255)
    v=255;
  if (v<0)
    v=0;
}

void SaveData(unsigned long Sensetime){
unsigned long previousMillis=0;
String FileName = "TESTINGFILE_NUMBER_" + String(FileNumberI);
myFile = SD.open(FileName, FILE_WRITE);
sensors.requestTemperatures();

unsigned long interval = Sensetime;
int period = 2500;

unsigned long currentMillis = millis();
unsigned long currenttime = millis();
```

```
while (millis() <  currentMillis + interval) {

            if (myFile) {
              Serial.println("Writing to test.txt...");
              myFile.println("testing 1, 2, 3.");
             }

currenttime = millis();

if(currenttime - previousMillis > period) {

       int temp = sensors.getTempCByIndex(0);
      // Serial.print("Temp: "); Serial.println(temp);
        String Tstring=String(temp) ;
        Serial.print(Tstring);
        mySerial.println("5,"+ Tstring);
        mySerial.flush();

    previousMillis = currenttime;
  }

}

 Serial.println("DONE DONE DONE");
 delay(1000);
 mySerial.println("6,Done");

}
```

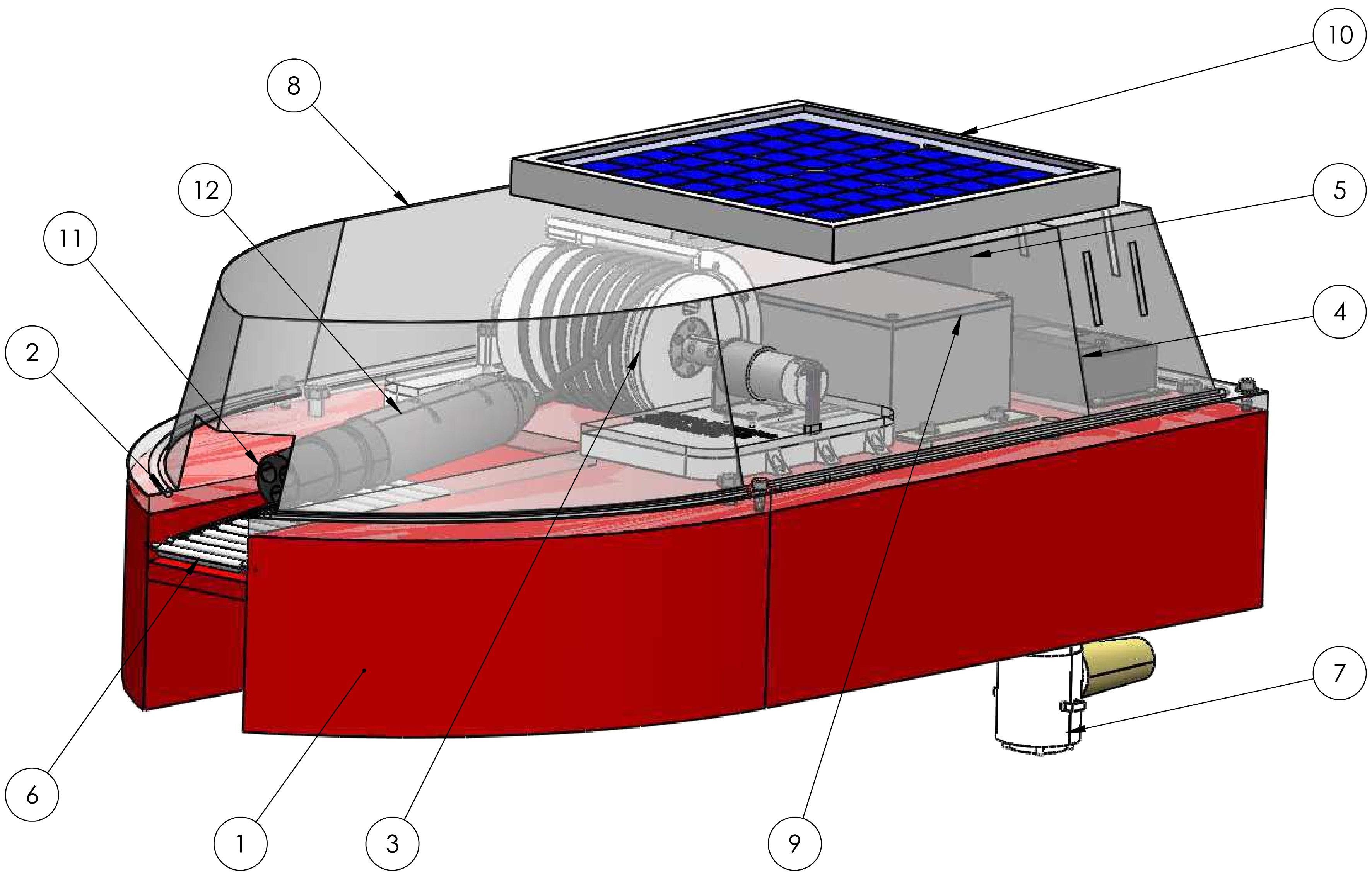
DOCUMENT NO.2:

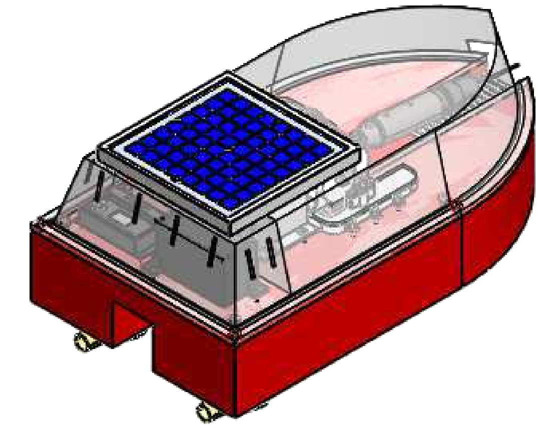
DRAWINGS

13.6. SolidWorks Drawings

For ease of reading, the page numbers are shown below where the following drawings can be found.

13.6.1. Full Assembly for hydro pumps





ITEM NO.	PART NUMBER	QTY
1	Platform_Pumps	1
2	Acylic_Platform_Pumps	1
3	SensorDeployAssembly	1
4	solarchargercontroller.stp	1
5	rechargeable_battery.stp	1
6	roller_rotating_platform	15
7	Pumps	2
8	Cover.prt	1
9	WaterproofBoxAssem	1
10	solarpanel.prt	1
11	temperaturesensor	1
12	SensorWeight	2

INFORMACIÓN CONFIDENCIAL Y DE MARCA

LA INFORMACIÓN INCLUIDA EN ESTE DIBUJO PERTENECE EXCLUSIVAMENTE A <NOMBRE DE LA COMPAÑÍA>. QUEDA PROHIBIDA LA REPRODUCCIÓN TOTAL O PARCIAL SIN EL PREVIO CONSENTIMIENTO POR ESCRITO DE <NOMBRE DE LA COMPAÑÍA>.

SIGUIENTE ENSAMBLAJE | UTILIZADO EN

APLICACIÓN

SI NO SE INDICA LO CONTRARIO:
LAS COTAS SE EXPRESAN EN PULGADAS
TOLERANCIAS:
FRACCIONAL ±
ANGULAR: MÁQUINA ± PLIEGUE ±
2 LUGARES DECIMALES ±
3 LUGARES DECIMALES ±

INTERPRETAR TOLERANCIA GEOMÉTRICA POR:

MATERIAL

ACABADO

NO CAMBIE LA ESCALA

	NOMBRE	FECHA
DIBUJADO		
VERIFICADO		
INGENIERÍA		
FABRICACIÓN		
CALIDAD		

COMENTARIOS:

Boston University

TÍTULO:

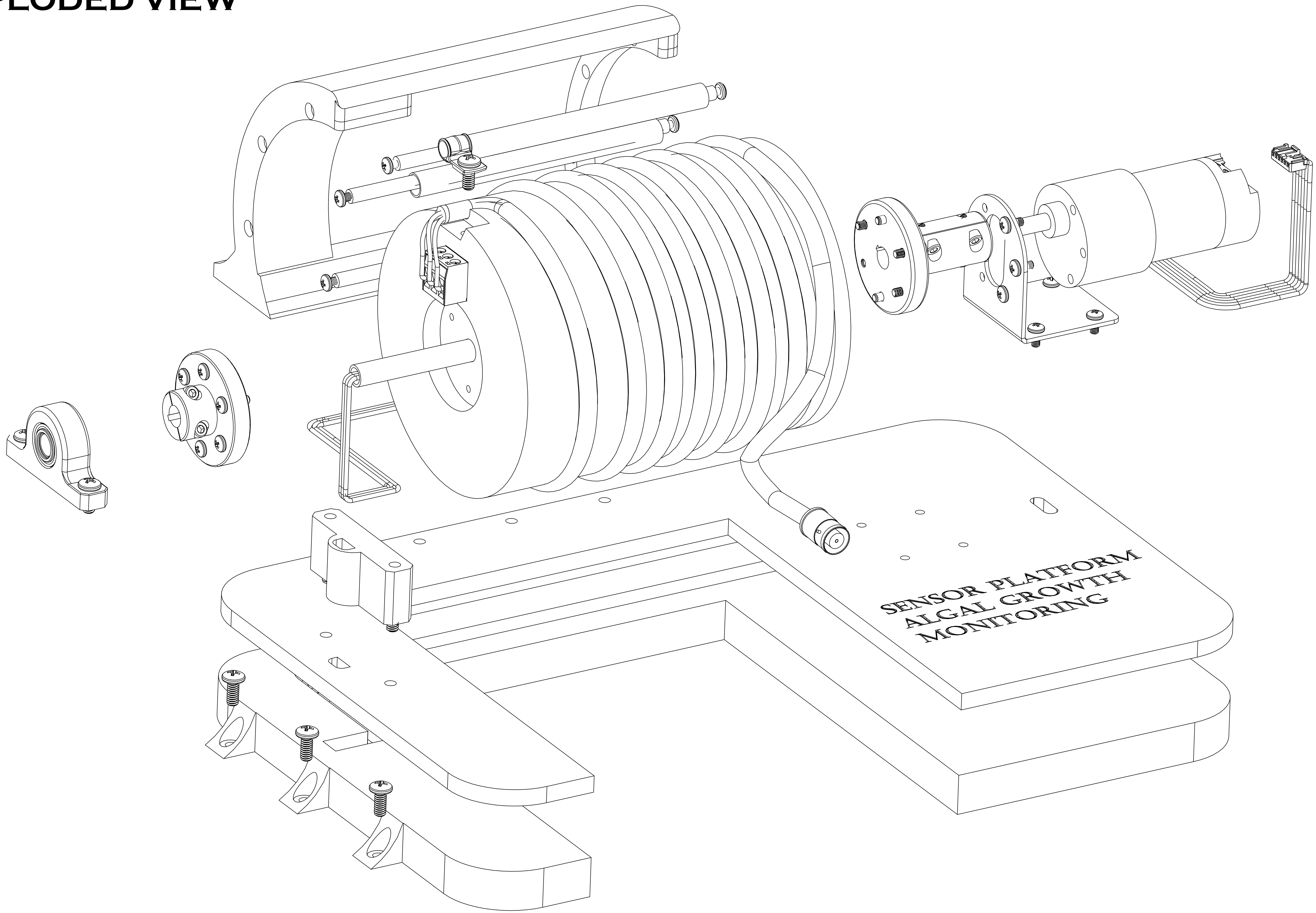
Platform Assembly Pumps

TAMAÑO B | N.º DE DIBUJO 1 | REV

ESCALA: 1:12 | PESO: | HOJA 1 DE 1

UNLESS OTHERWISE SPECIFIED:		NAME	DATE	GROUP 22	
DIMENSIONS ARE IN INCHES TOLERANCES: 0.05	DRAWN			TITLE: Sensor Deployment Assembly	
	CHECKED				
INTERPRET GEOMETRIC TOLERANCING PER:	ENG APPR				
	MFG APPR				
	Q.A.				
MATERIAL:	NOTES:			PART NO.	REV.
FINISH:				WEIGHT:	

EXPLODED VIEW



Section View to Display

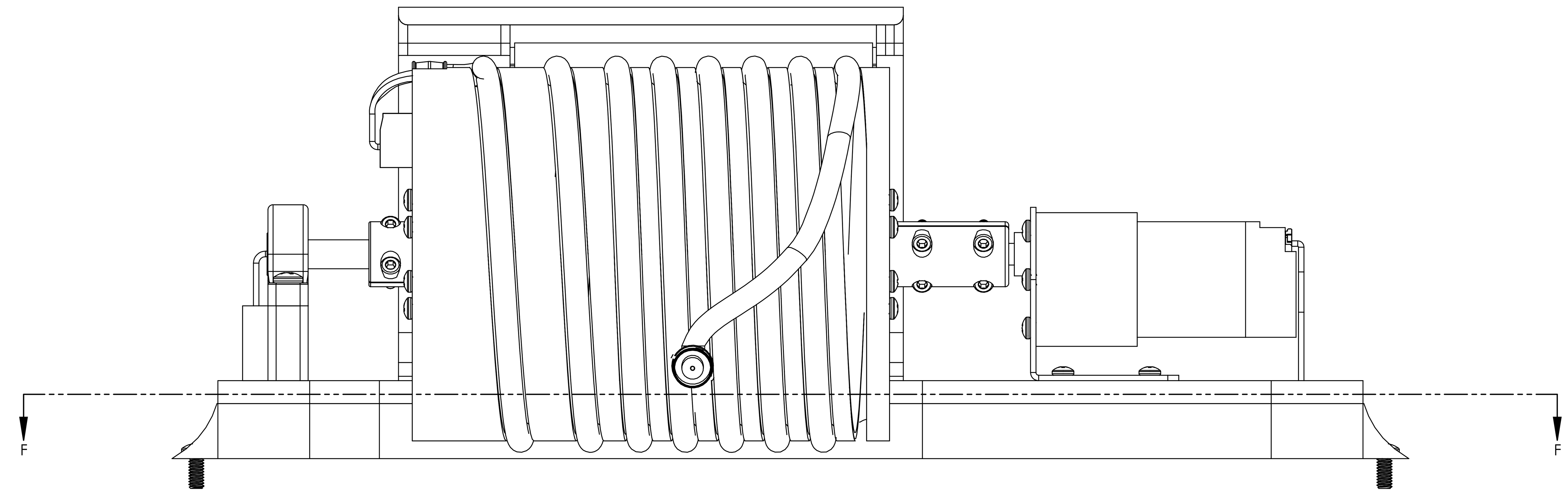
Rollers inside roller cover

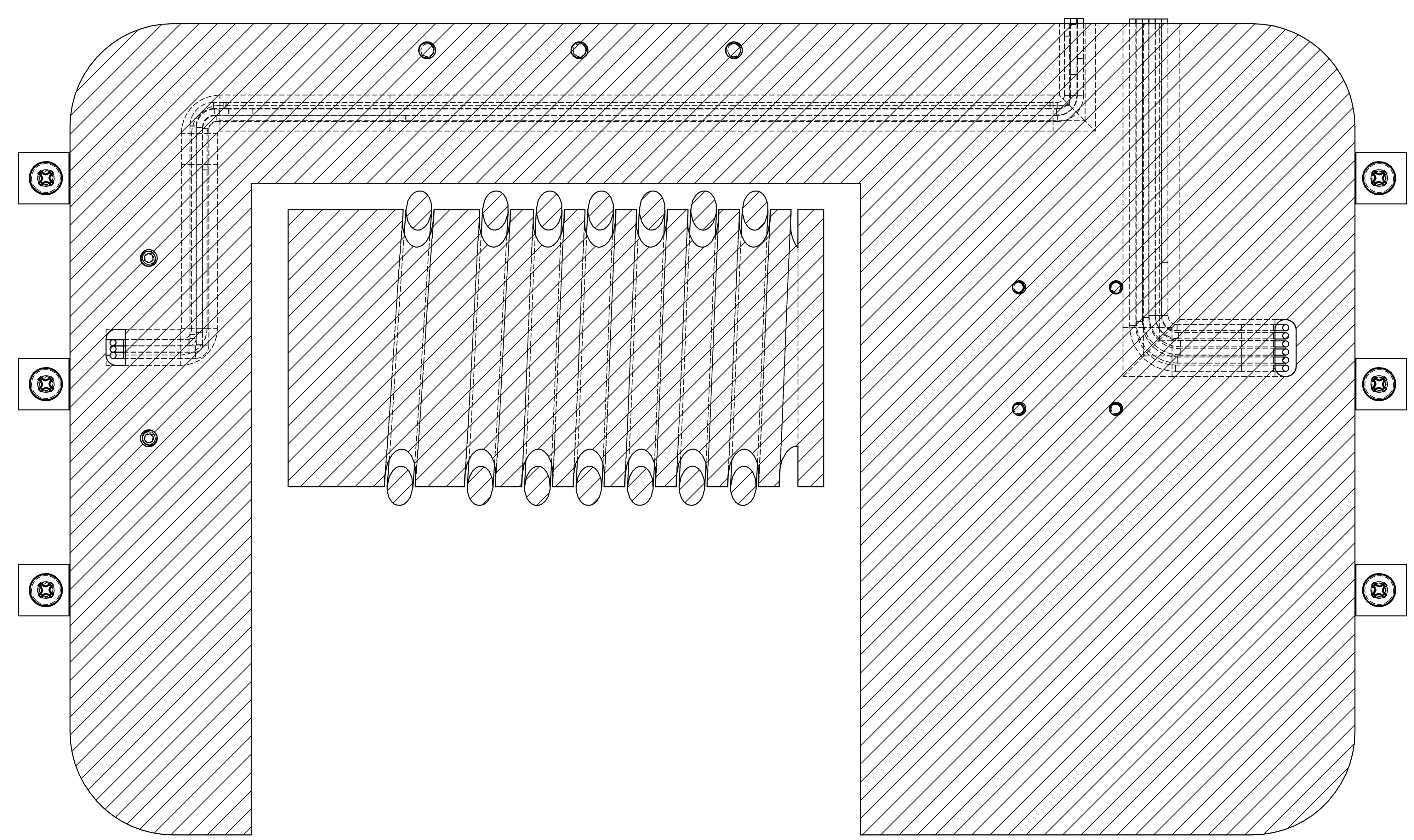
SECTION VIEW

TO DISPLAY

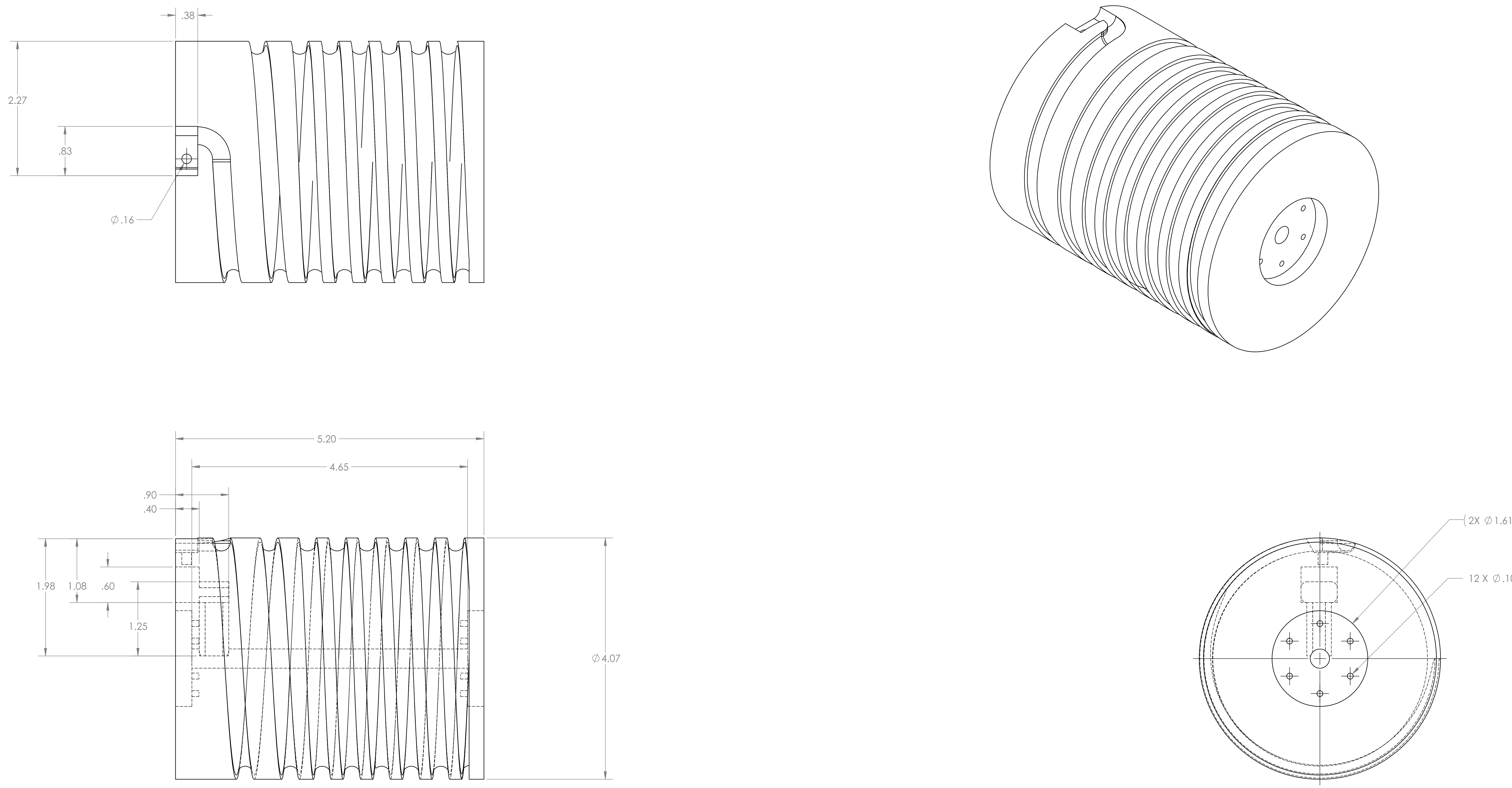
WIRE CONNECTIONS

INSIDE HDPE BASE

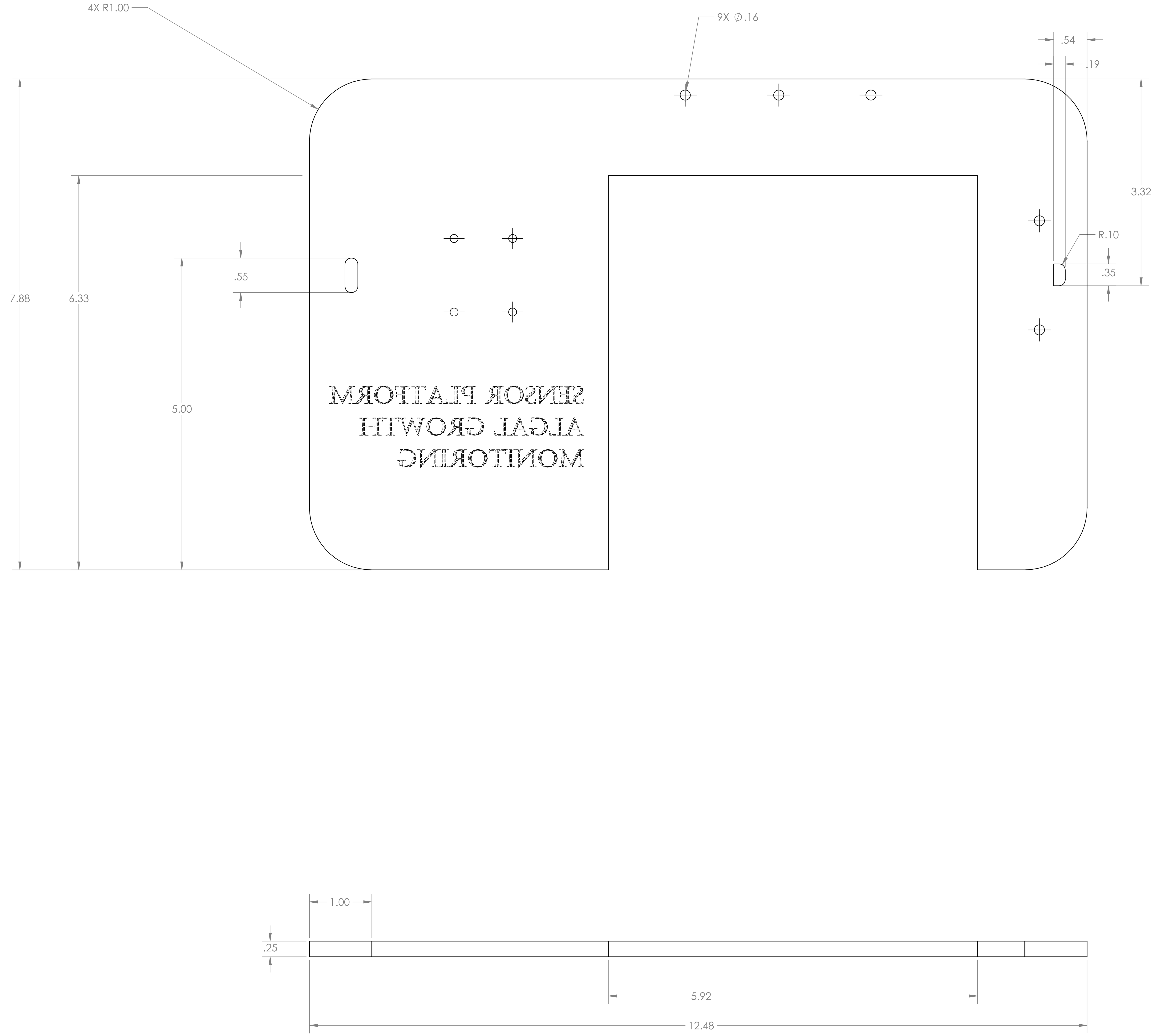




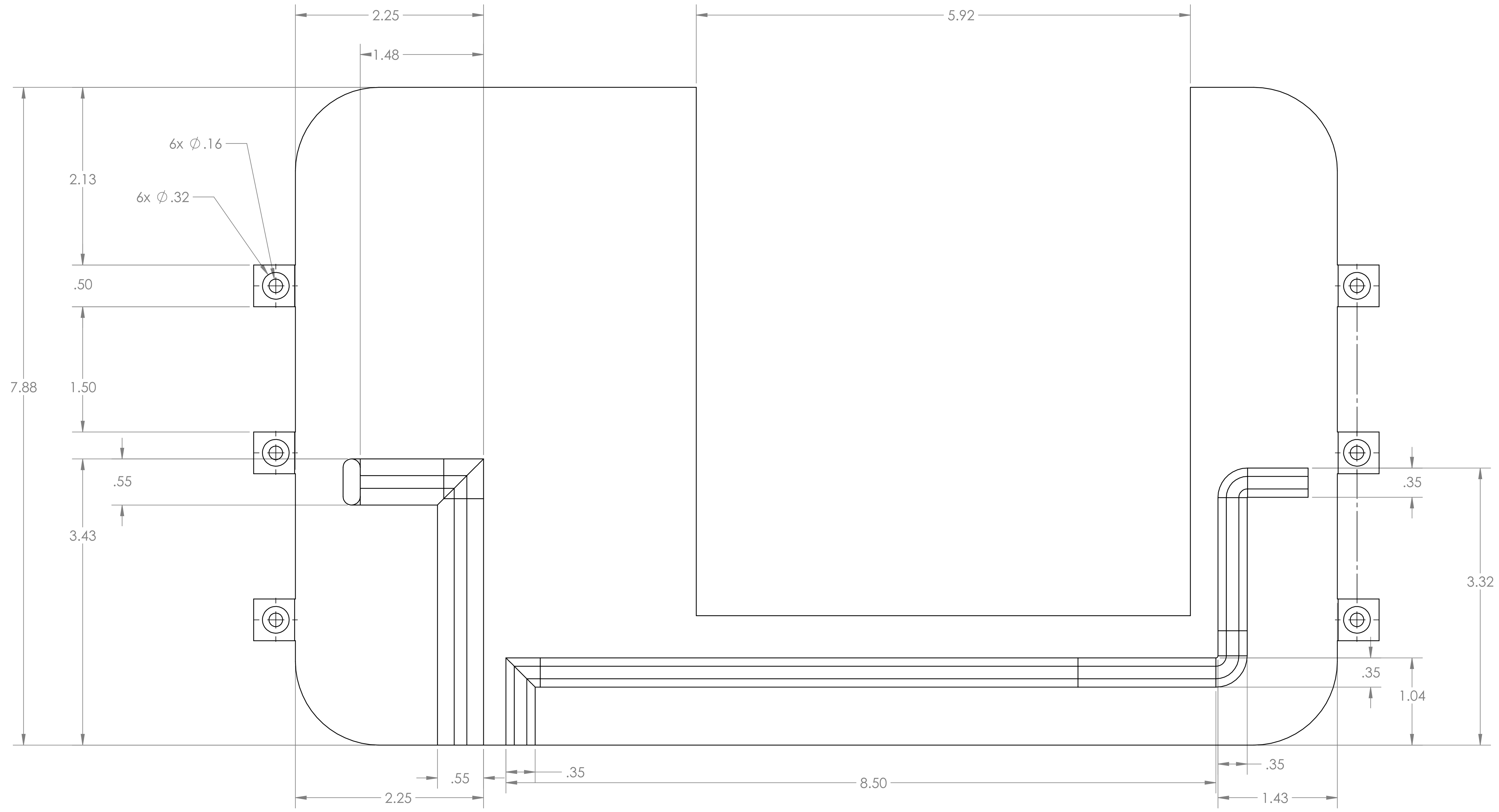
SECTION F-F
SCALE 1 : 1

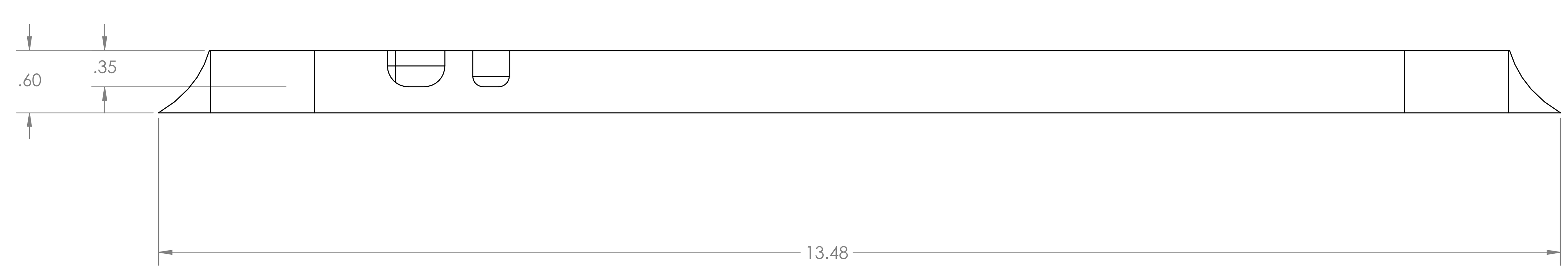


Threaded Spool

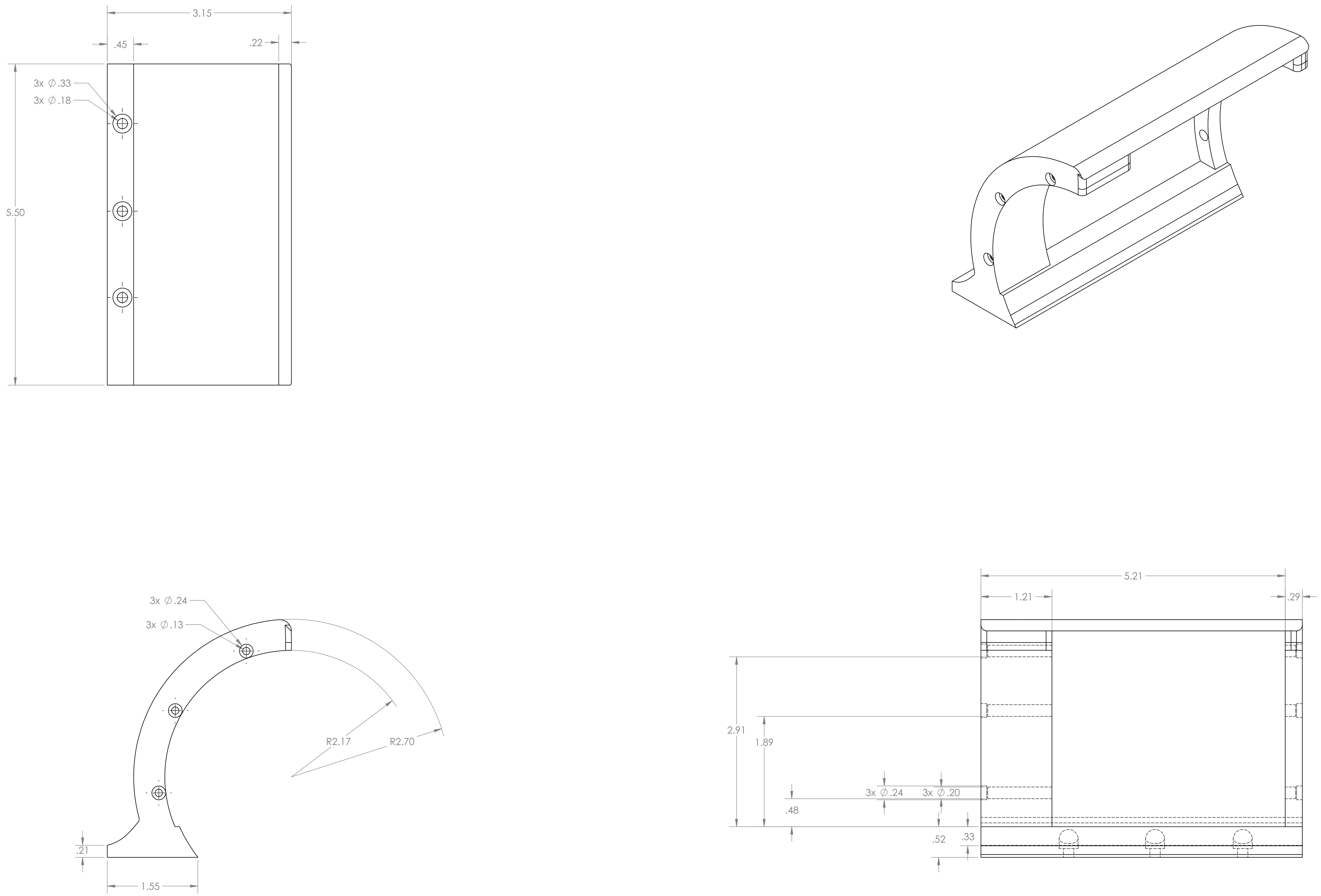

4X R1.00
9X ∅.16
.54
.19
3.32
R.10
.35
.55
7.88
6.33
5.00
SENSOR PLATFORM
ALGAL GROWTH
MONITORING
1.00
.25
5.92
12.48

Acrlyic

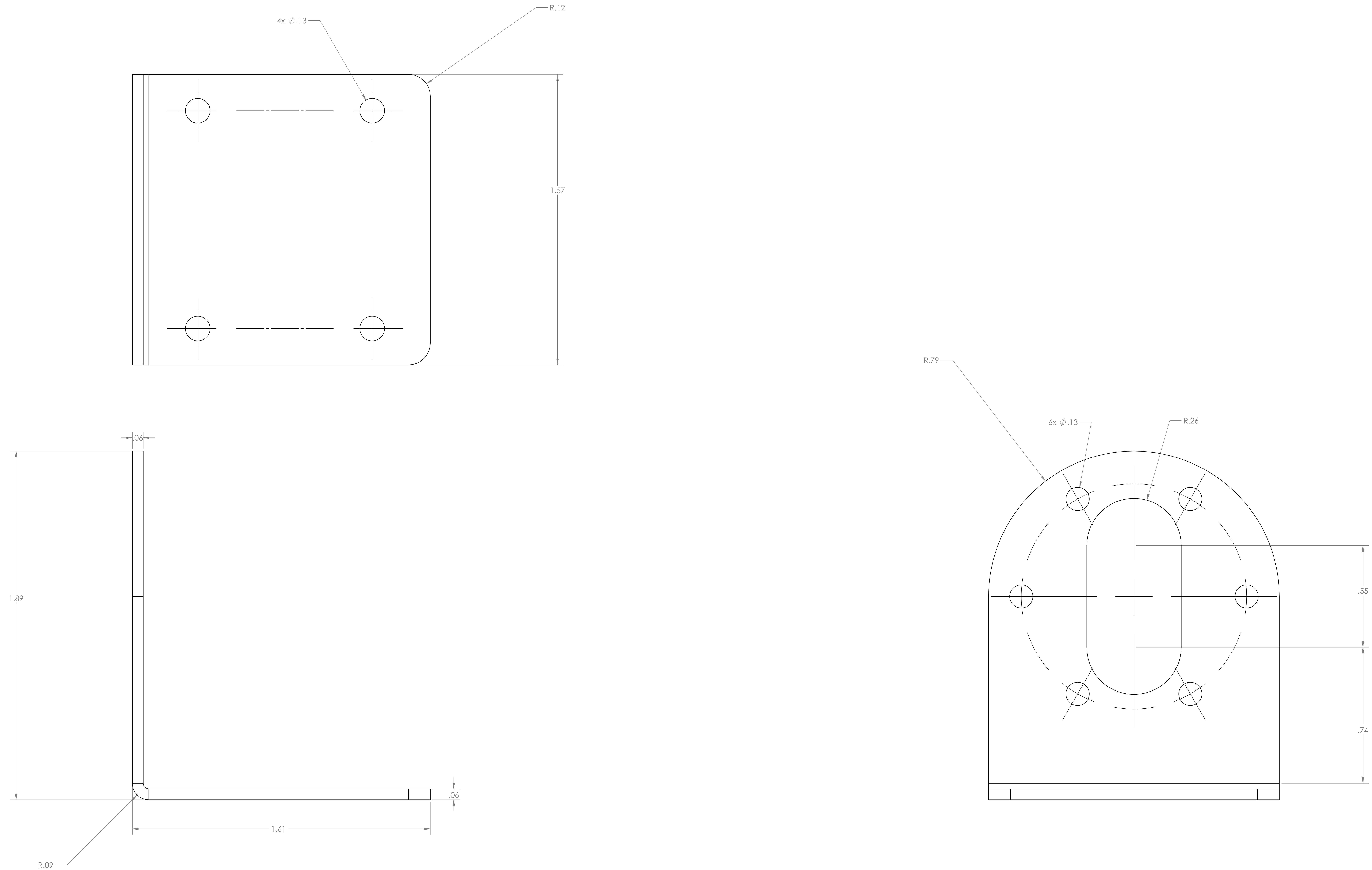




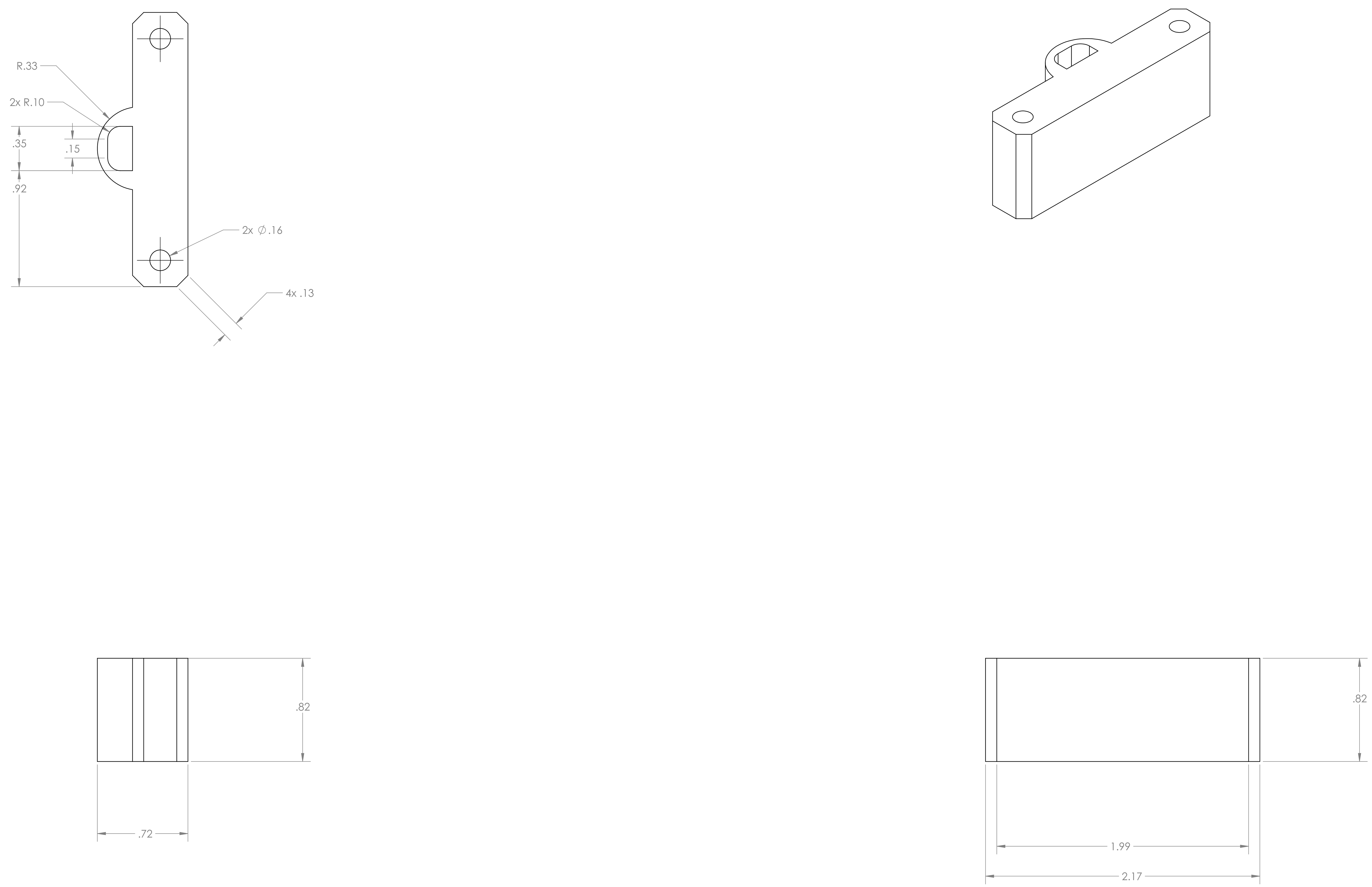
HDPE Support



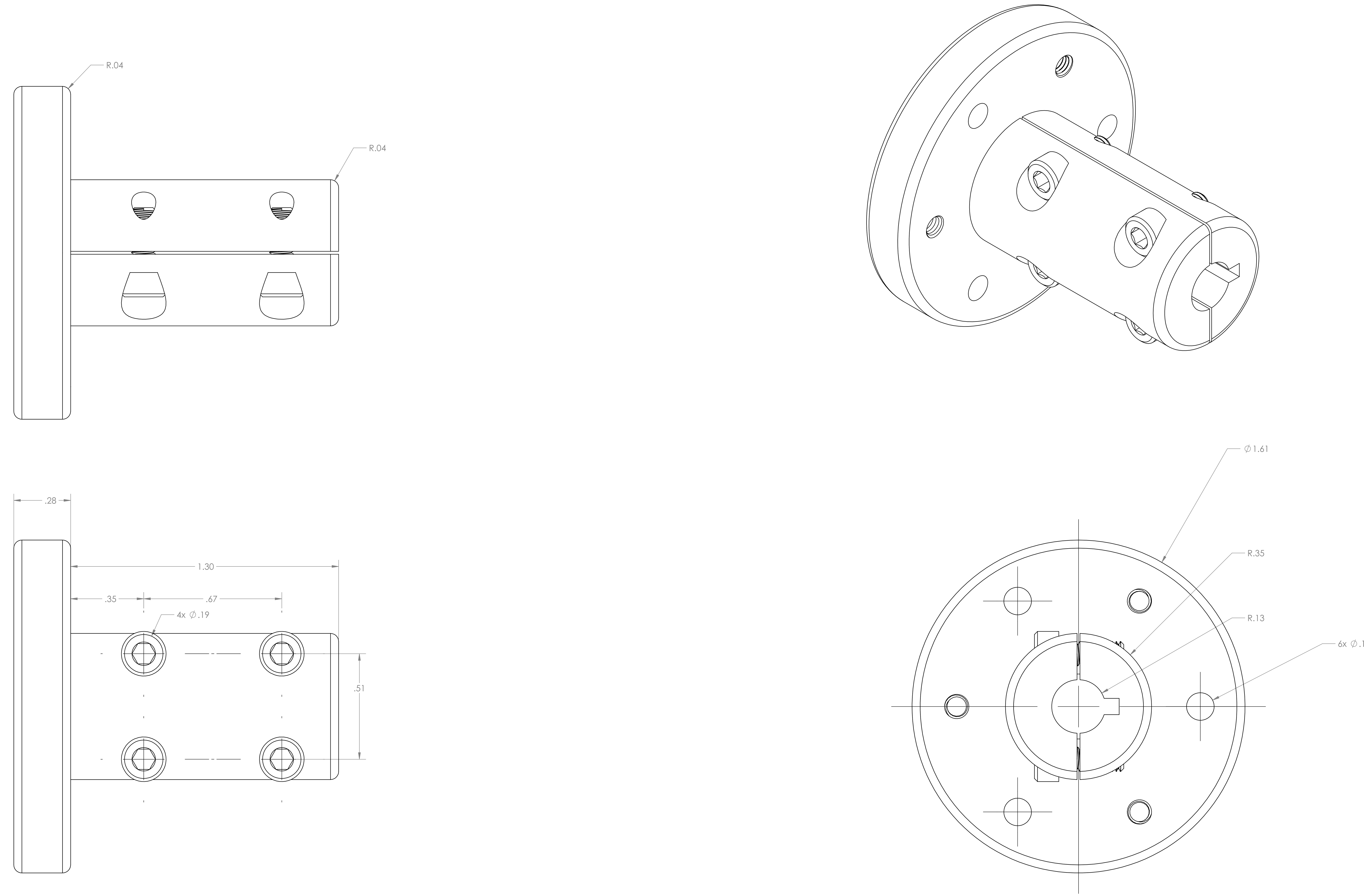
Roller Cover



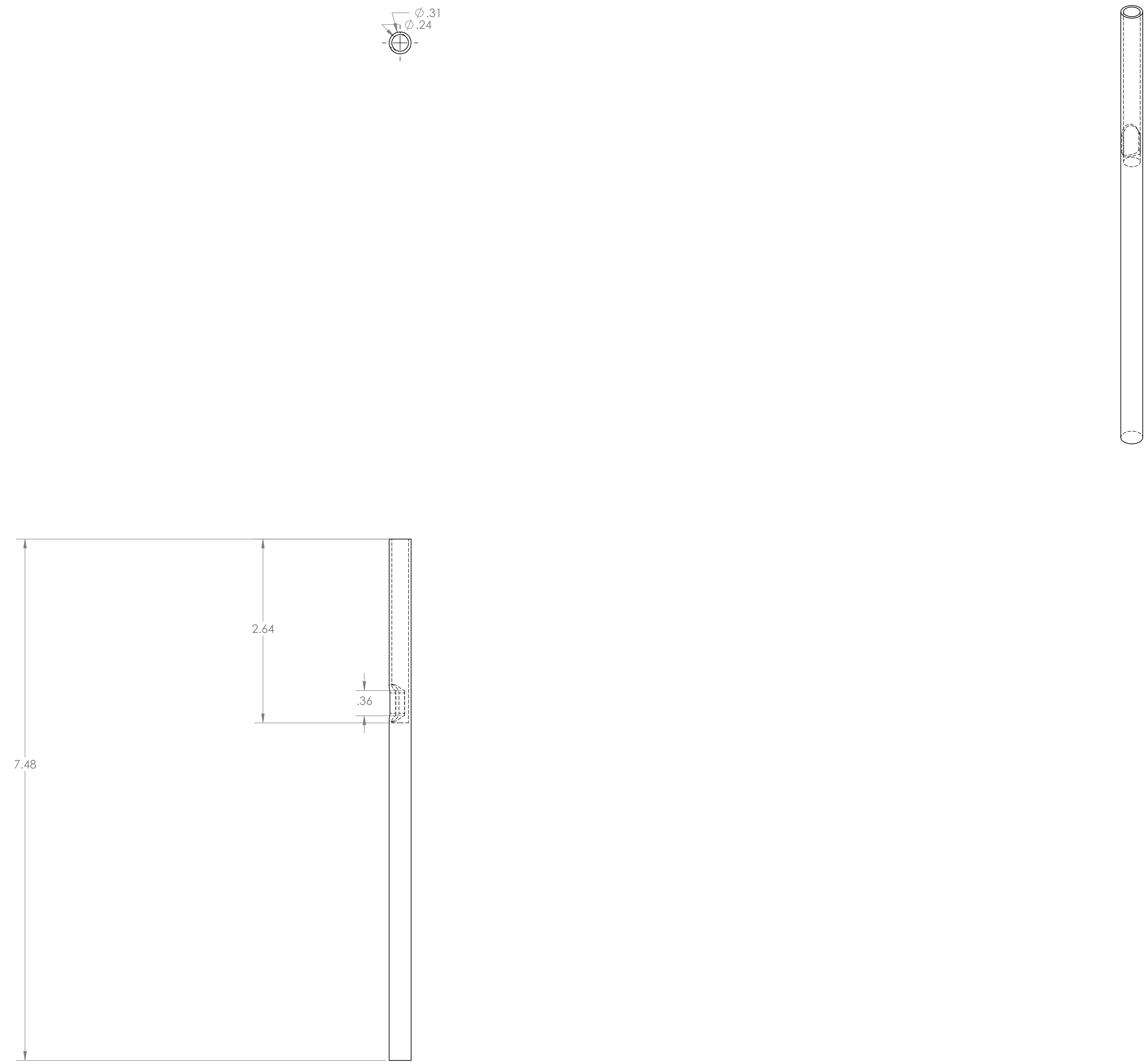
Motor Bracket



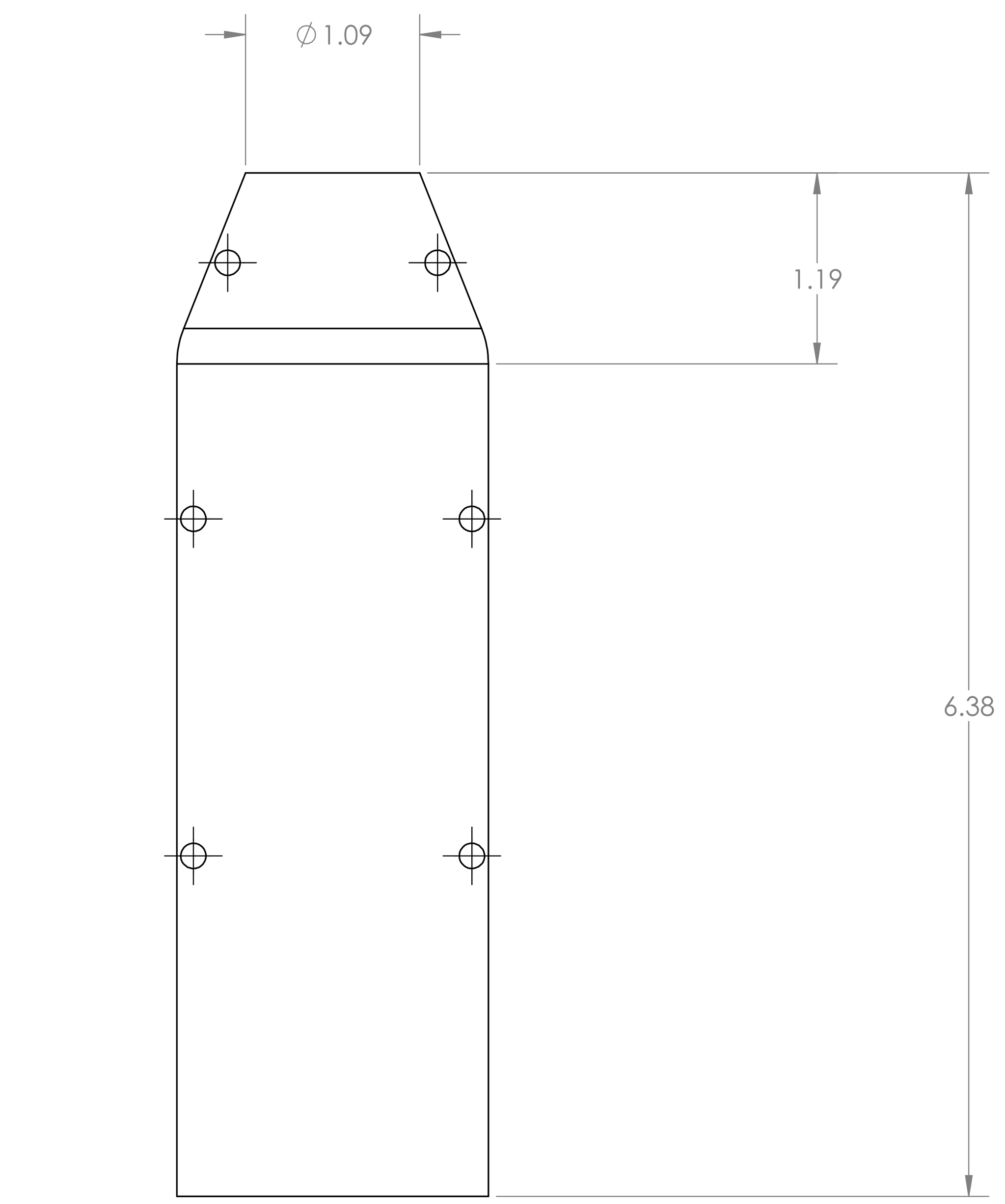
Bracket Support

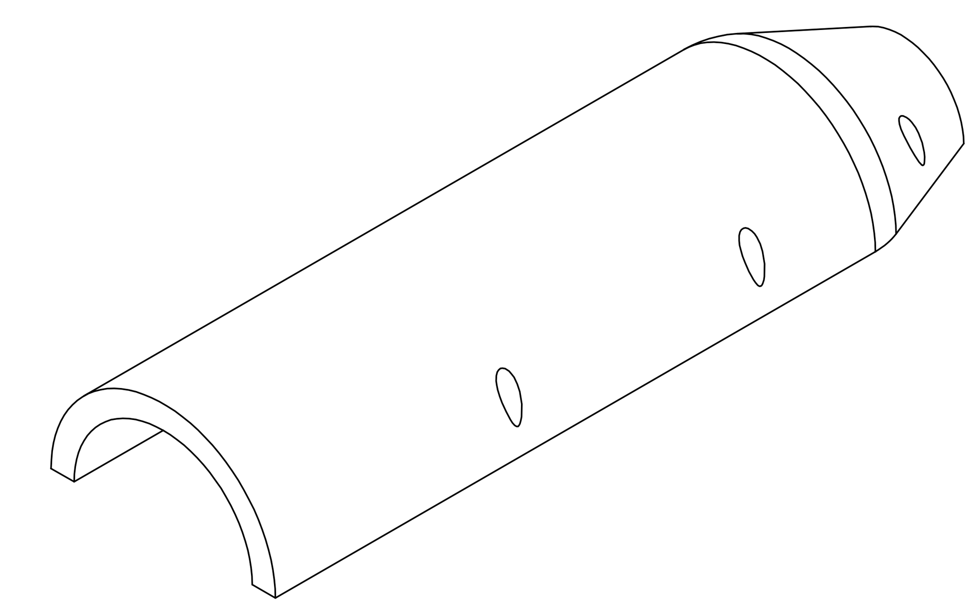


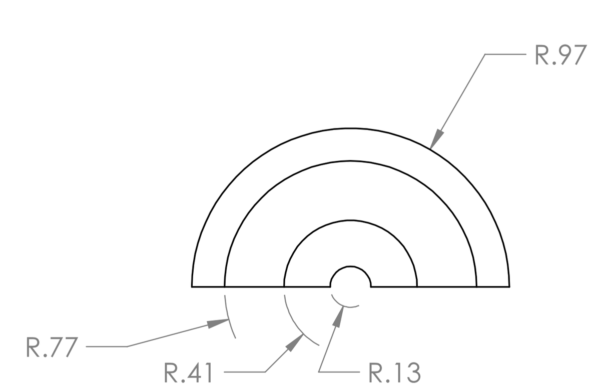
Shaft Collar

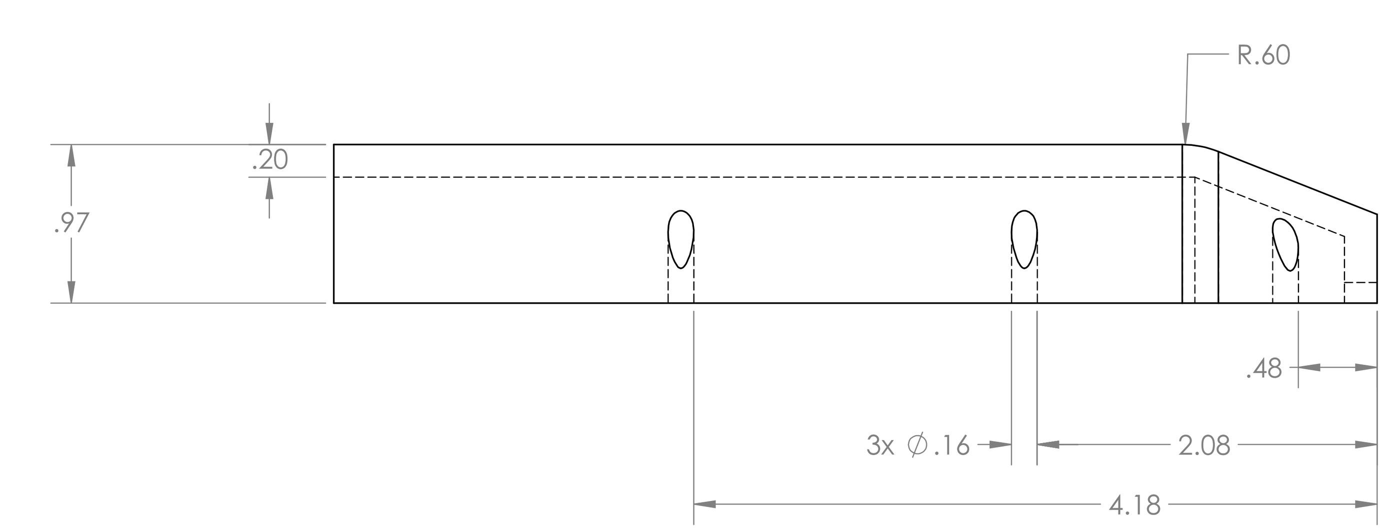


Aluminum Shaft



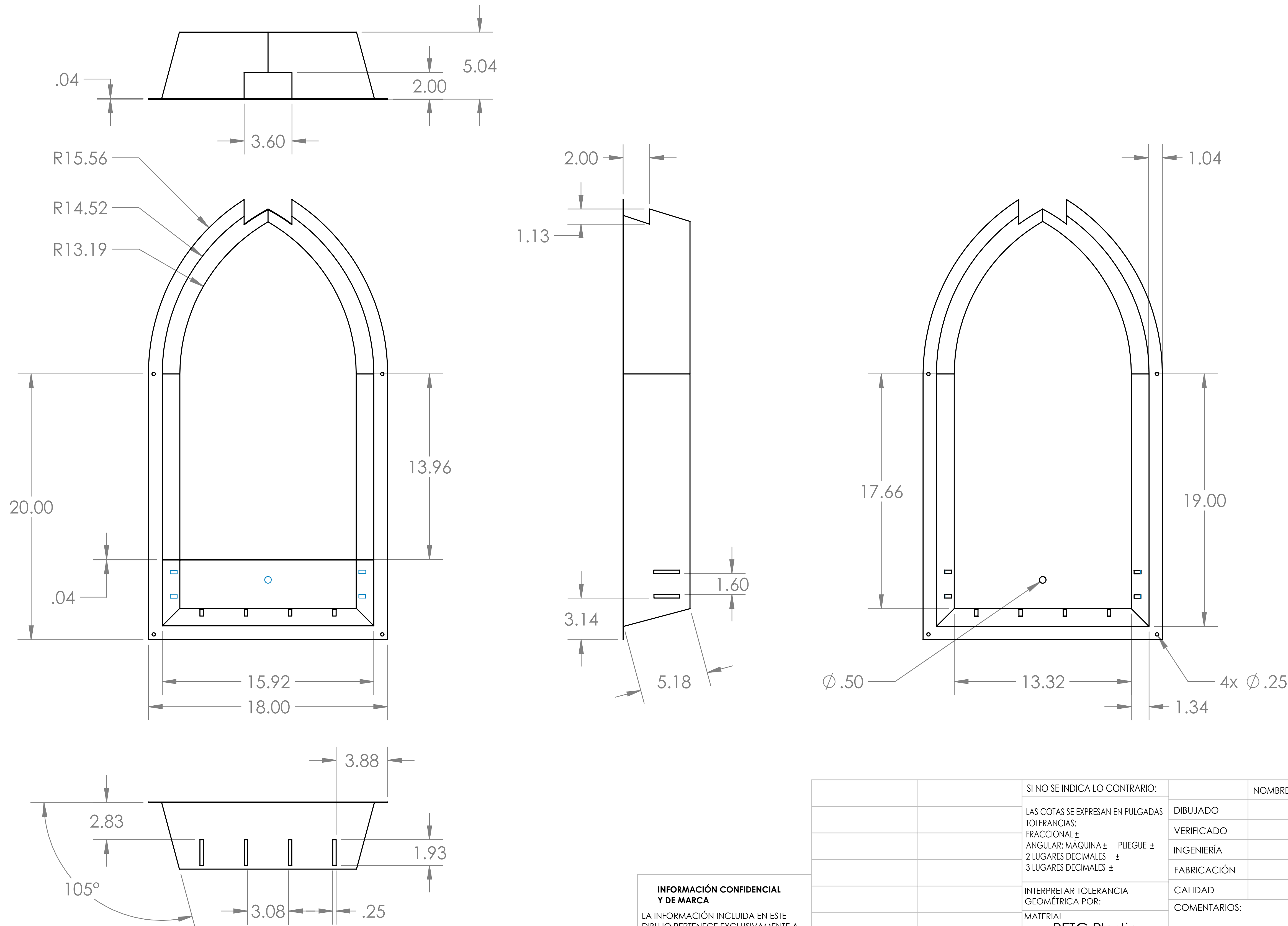


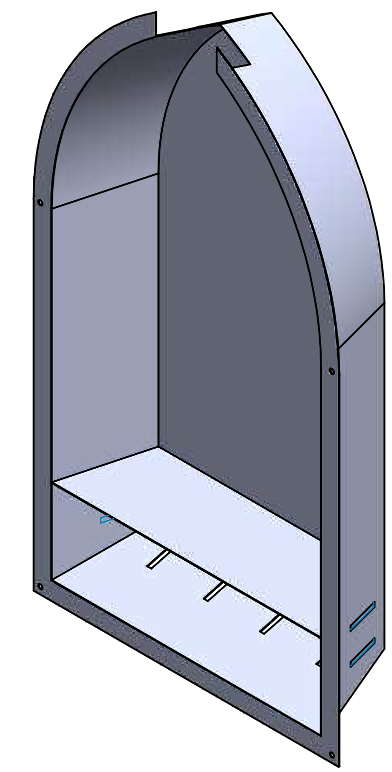




Sensor Casing

13.6.3. Cover





INFORMACIÓN CONFIDENCIAL Y DE MARCA

LA INFORMACIÓN INCLUIDA EN ESTE DIBUJO PERTENECE EXCLUSIVAMENTE A <NOMBRE DE LA COMPAÑÍA>. QUEDA PROHIBIDA LA REPRODUCCIÓN TOTAL O PARCIAL SIN EL PREVIO CONSENTIMIENTO POR ESCRITO DE <NOMBRE DE LA COMPAÑÍA>.

SIGUIENTE ENSAMBLAJE	UTILIZADO EN
APLICACIÓN	

SI NO SE INDICA LO CONTRARIO:
LAS COTAS SE EXPRESAN EN PULGADAS
TOLERANCIAS:
FRACCIONAL ±
ANGULAR: MÁQUINA ± PLIEGUE ±
2 LUGARES DECIMALES ±
3 LUGARES DECIMALES ±

INTERPRETAR TOLERANCIA GEOMÉTRICA POR:

MATERIAL
PETG Plastic

ACABADO

NO CAMBIE LA ESCALA

	NOMBRE	FECHA
DIBUJADO		
VERIFICADO		
INGENIERÍA		
FABRICACIÓN		
CALIDAD		

COMENTARIOS:

Boston University

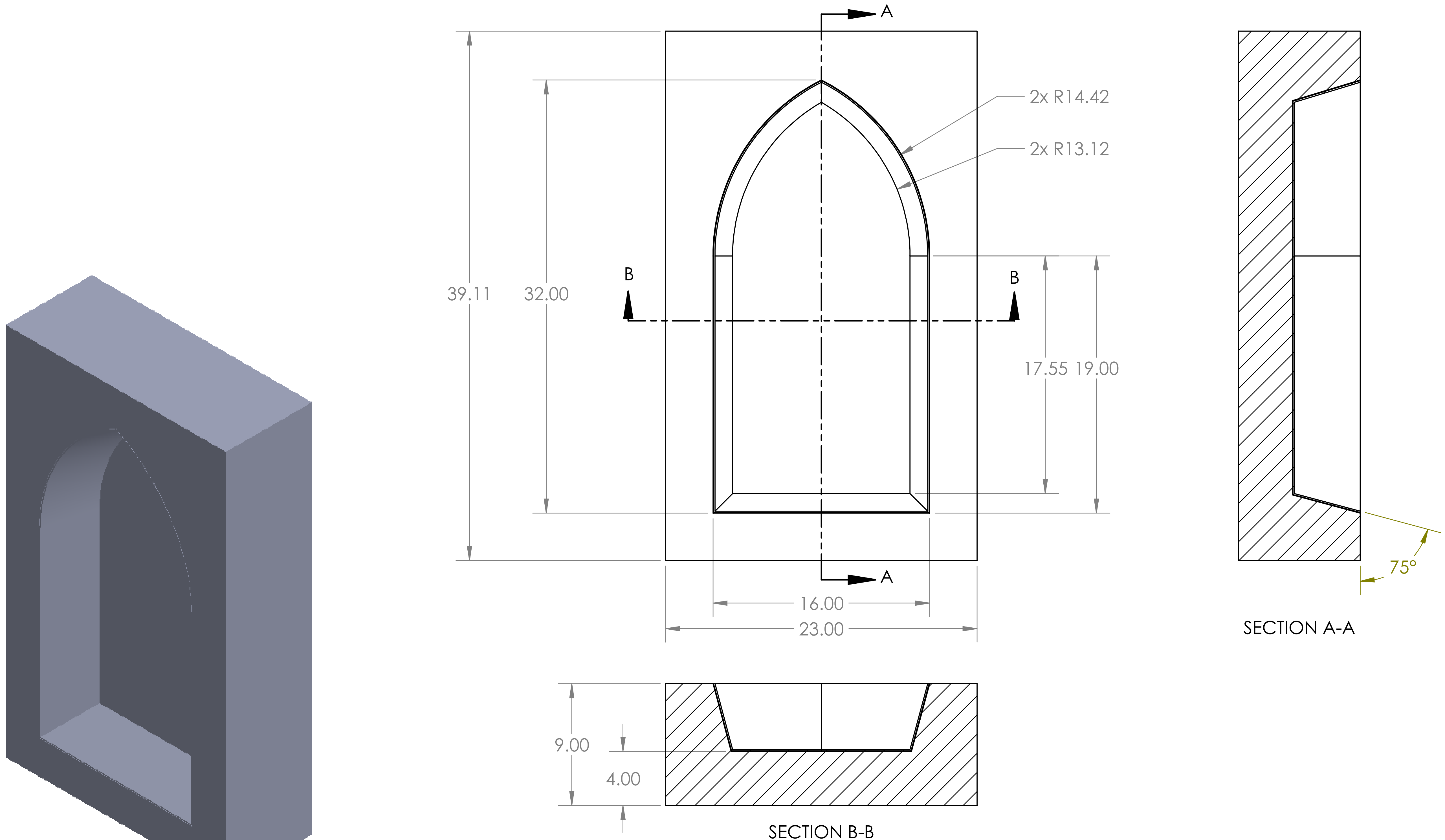
TÍTULO:

Cover

TAMAÑO	N.º DE DIBUJO	REV
B	3	

ESCALA: 1:8 PESO: HOJA 1 DE 1

13.6.4. Cover Mold



INFORMACIÓN CONFIDENCIAL Y DE MARCA

SIGUIENTE ENSAMBLAJE | UTILIZADO EN
APLICACIÓN

SI NO SE INDICA LO CONTRARIO:
LAS COTAS SE EXPRESAN EN PULGADAS
TOLERANCIAS:
FRACCIONAL ±
ANGULAR: MÁQUINA ± PLIEGUE ±
2 LUGARES DECIMALES ±
3 LUGARES DECIMALES ±
INTERPRETAR TOLERANCIA GEOMÉTRICA POR:
MATERIAL Aluminum
ACABADO
NO CAMBIE LA ESCALA

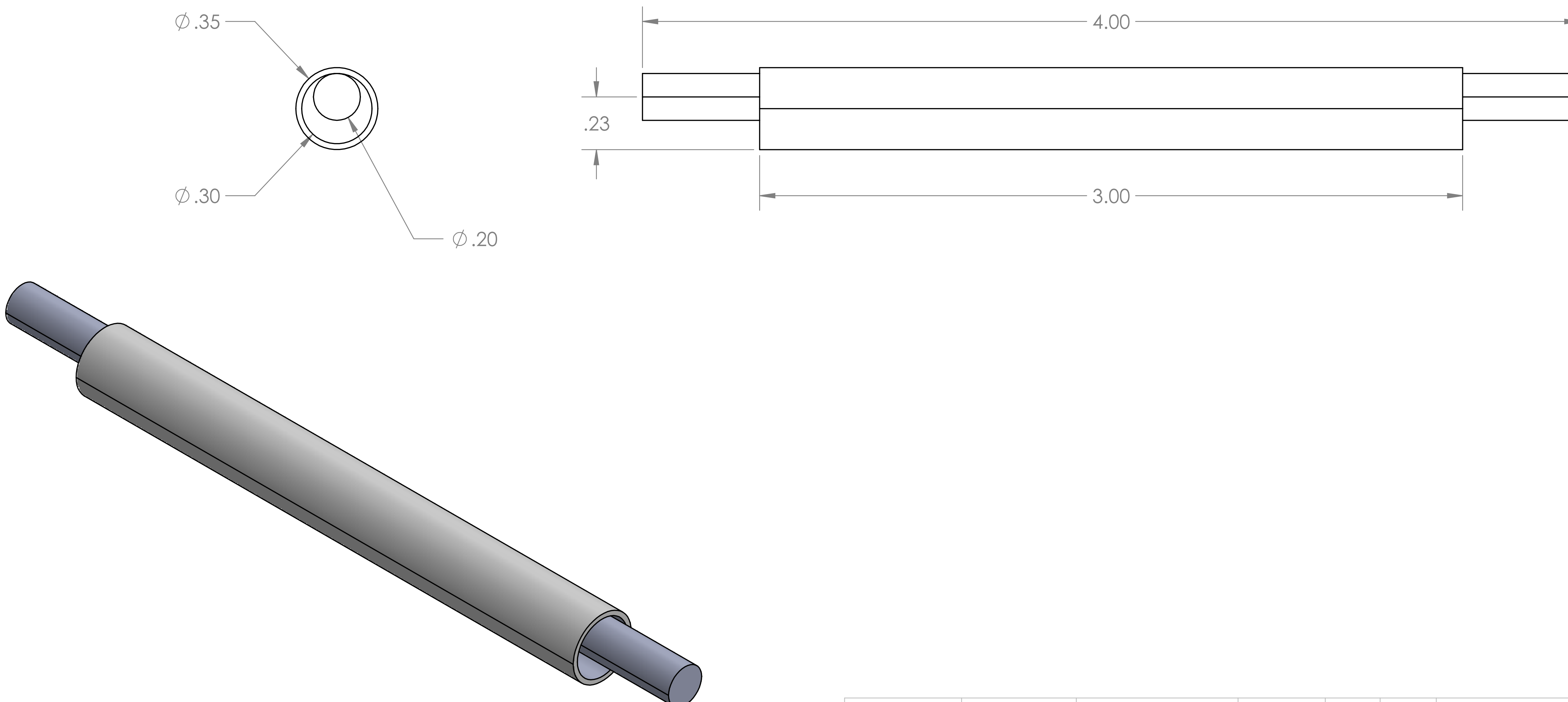
	NOMBRE	FECHA
DIBUJADO		
VERIFICADO		
INGENIERÍA		
FABRICACIÓN		
CALIDAD		

COMENTARIOS:
Boston University

TÍTULO:
Cover Mold

TAMAÑO B | N.° DE DIBUJO 4 | REV
ESCALA: 1:8 | PESO: | HOJA 1 DE 1

13.6.5. Rollers



INFORMACIÓN CONFIDENCIAL Y DE MARCA

LA INFORMACIÓN INCLUIDA EN ESTE DIBUJO PERTENECE EXCLUSIVAMENTE A <NOMBRE DE LA COMPAÑÍA>. QUEDA PROHIBIDA LA REPRODUCCIÓN TOTAL O PARCIAL SIN EL PREVIO CONSENTIMIENTO POR ESCRITO DE <NOMBRE DE LA COMPAÑÍA>.

SIGUIENTE ENSAMBLAJE | UTILIZADO EN

APLICACIÓN

SI NO SE INDICA LO CONTRARIO:
LAS COTAS SE EXPRESAN EN PULGADAS
TOLERANCIAS:
FRACCIONAL ±
ANGULAR: MÁQUINA ± PLIEGUE ±
2 LUGARES DECIMALES ±
3 LUGARES DECIMALES ±

INTERPRETAR TOLERANCIA GEOMÉTRICA POR:

MATERIAL
PVC + metal shaft

ACABADO

NO CAMBIE LA ESCALA

	NOMBRE	FECHA
DIBUJADO		
VERIFICADO		
INGENIERÍA		
FABRICACIÓN		
CALIDAD		

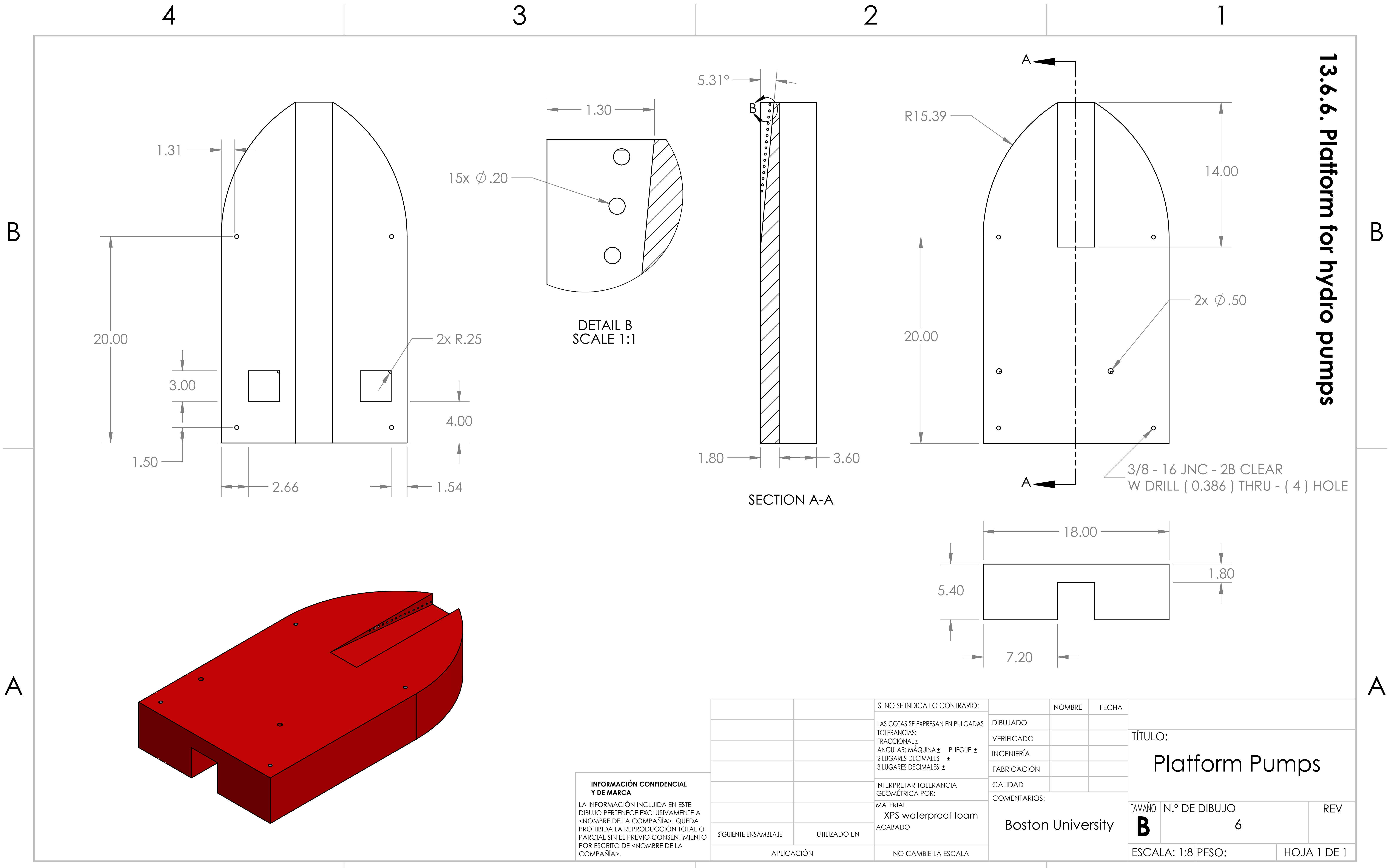
COMENTARIOS:
Boston University

TÍTULO:
Rollers

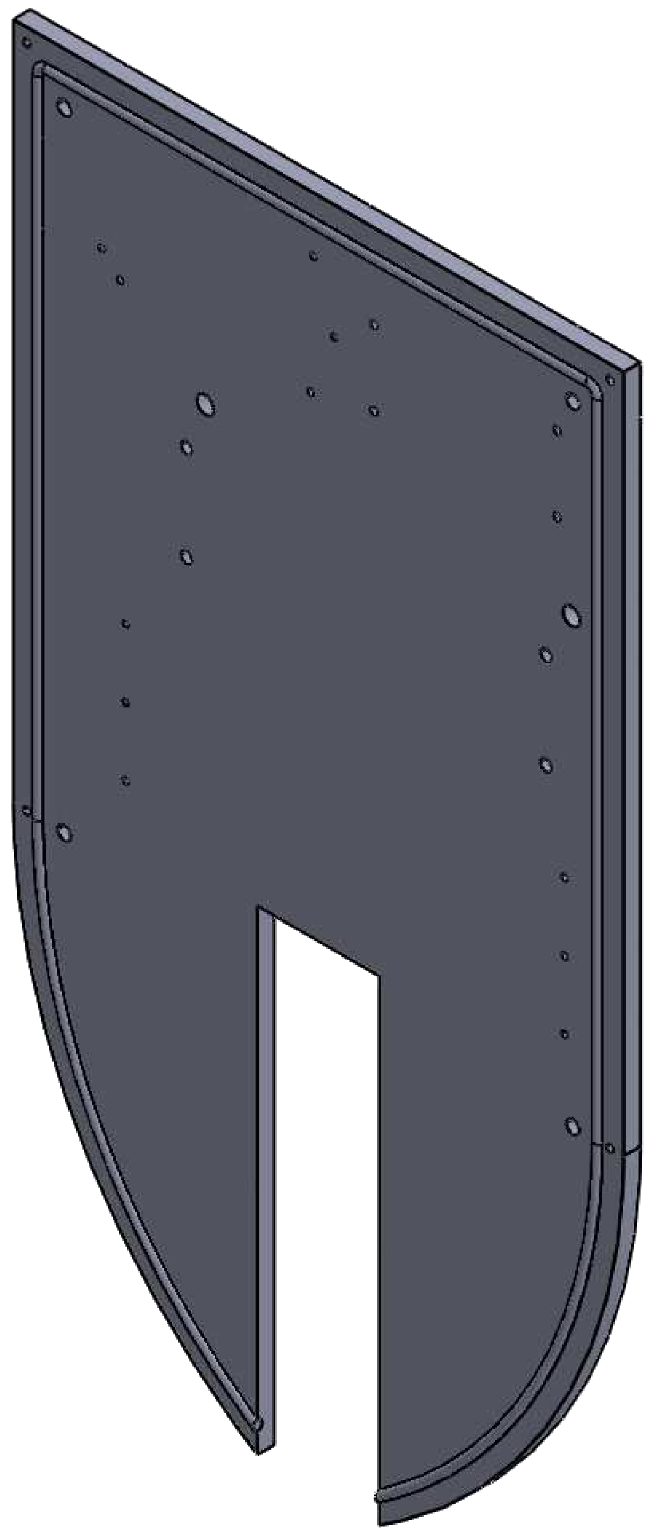
TAMAÑO B | N.º DE DIBUJO 5 | REV

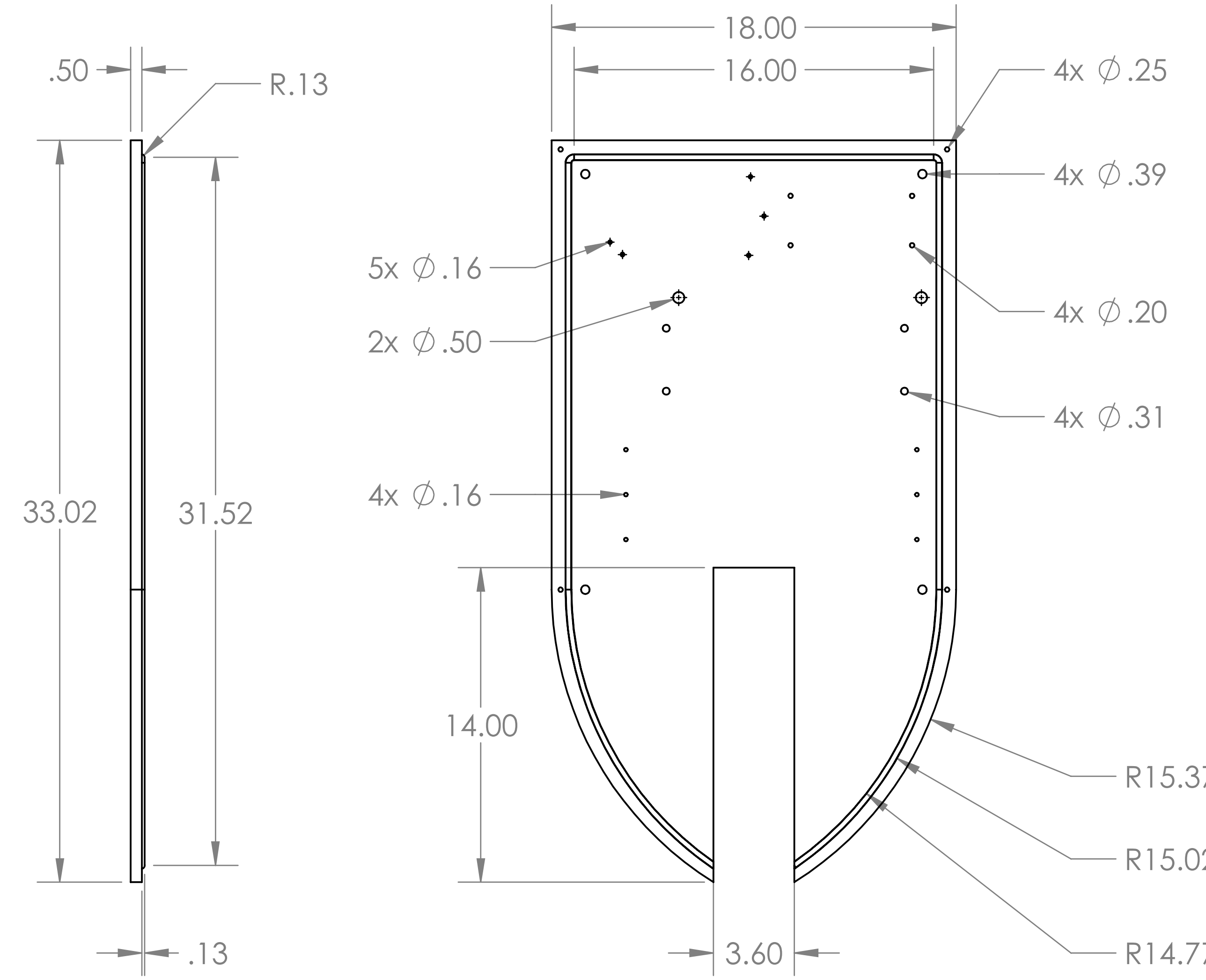
ESCALA: 2:1 | PESO: | HOJA 1 DE 1

13.6.6. Platform for hydro pumps



13.6.7. Acrylic piece for hydro pumps





INFORMACIÓN CONFIDENCIAL Y DE MARCA

LA INFORMACIÓN INCLUIDA EN ESTE DIBUJO PERTENECE EXCLUSIVAMENTE A <NOMBRE DE LA COMPAÑÍA>. QUEDA PROHIBIDA LA REPRODUCCIÓN TOTAL O PARCIAL SIN EL PREVIO CONSENTIMIENTO POR ESCRITO DE <NOMBRE DE LA COMPAÑÍA>.

SIGUIENTE ENSAMBLAJE	UTILIZADO EN
APLICACIÓN	

SI NO SE INDICA LO CONTRARIO:
LAS COTAS SE EXPRESAN EN PULGADAS
TOLERANCIAS:
FRACCIONAL ±
ANGULAR: MÁQUINA ± PLIEGUE ±
2 LUGARES DECIMALES ±
3 LUGARES DECIMALES ±

INTERPRETAR TOLERANCIA GEOMÉTRICA POR:

MATERIAL: Acrylic

ACABADO

NO CAMBIE LA ESCALA

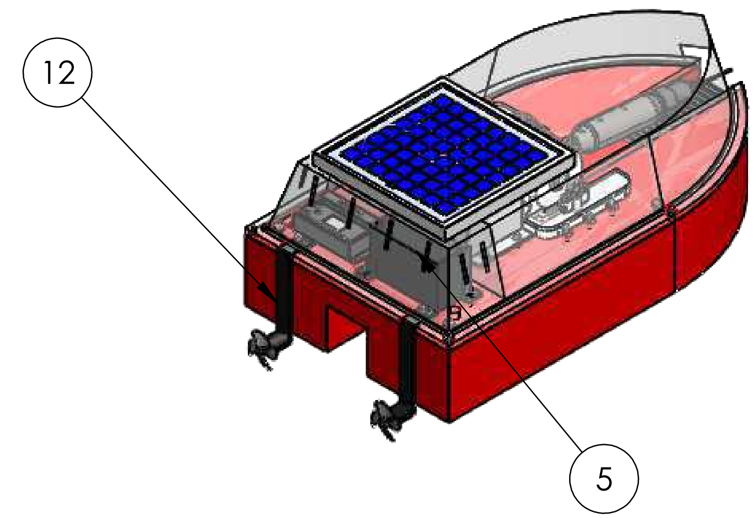
	NOMBRE	FECHA
DIBUJADO		
VERIFICADO		
INGENIERÍA		
FABRICACIÓN		
CALIDAD		

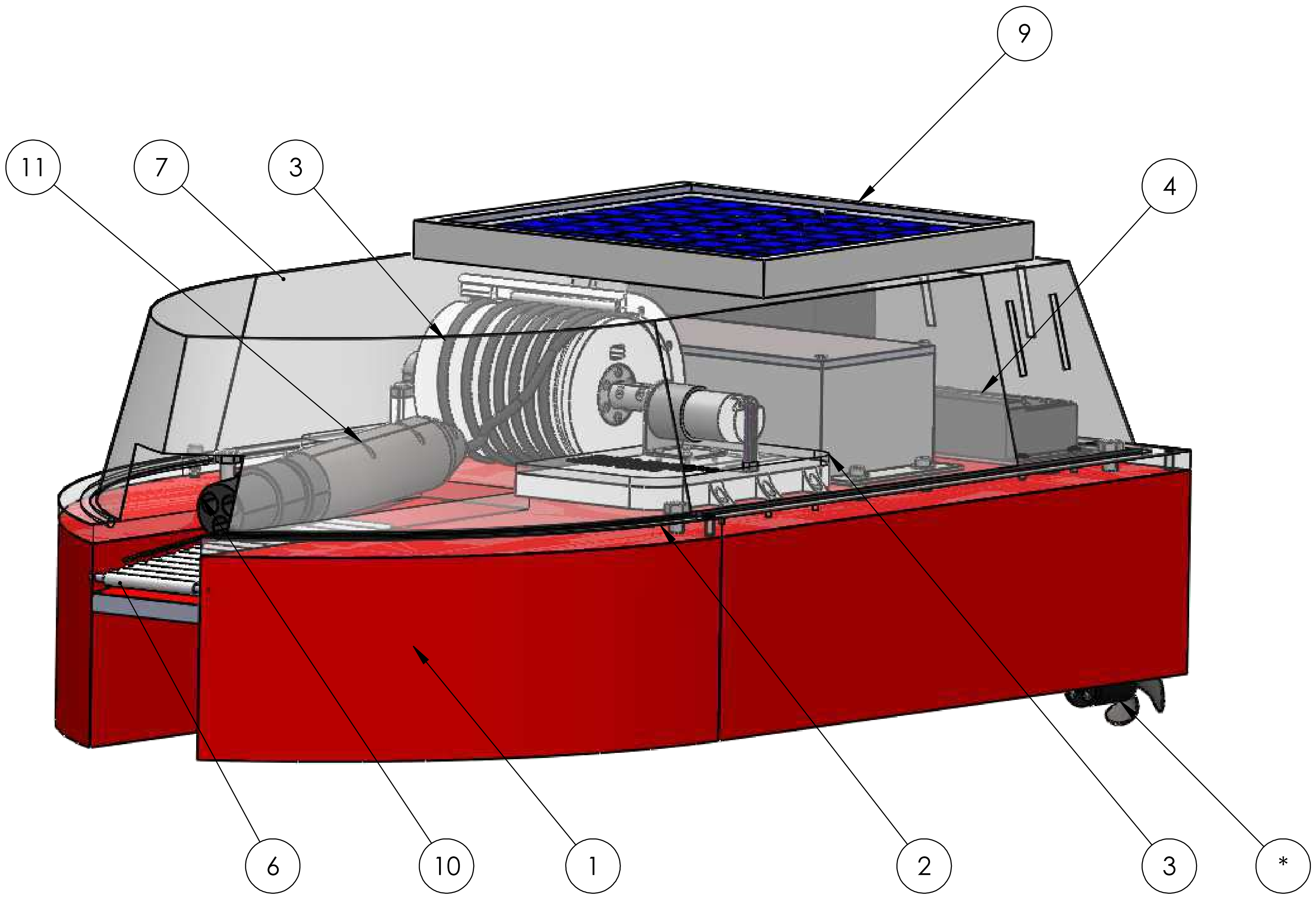
COMENTARIOS:
Boston University

TÍTULO:
Acrylic Platform Pumps

TAMAÑO	N.º DE DIBUJO	REV
B	7	
ESCALA: 1:8	PESO:	HOJA 1 DE 1

13.6.8. Full Assembly with propellers





ITEM NO	PART NUMBER	QTY
1	Platform_Propellor	1
2	Acrylic_Platform_Prop	1
3	SensorDeployAssembly	1
4	solarchargercontroller.stp	1
5	rechargeable_battery.stp	1
6	roller_rotating_platform	15
7	Cover.prt	1
8	WaterproofBoxAssem	1
9	solarpanel.prt	1
10	temperaturesensor	1
11	SensorWeight	2
12	propmount.prt	2
13	Propeller	2

SIGUIENTE ENSAMBLAJE | UTILIZADO EN

APLICACIÓN

SI NO SE INDICA LO CONTRARIO:
LAS COTAS SE EXPRESAN EN PULGADAS
TOLERANCIAS:
FRACCIONAL ±
ANGULAR: MÁQUINA ± PLIEGUE ±
2 LUGARES DECIMALES ±
3 LUGARES DECIMALES ±

INTERPRETAR TOLERANCIA GEOMÉTRICA POR:

MATERIAL

ACABADO

NO CAMBIE LA ESCALA

	NOMBRE	FECHA
DIBUJADO		
VERIFICADO		
INGENIERÍA		
FABRICACIÓN		
CALIDAD		

COMENTARIOS:

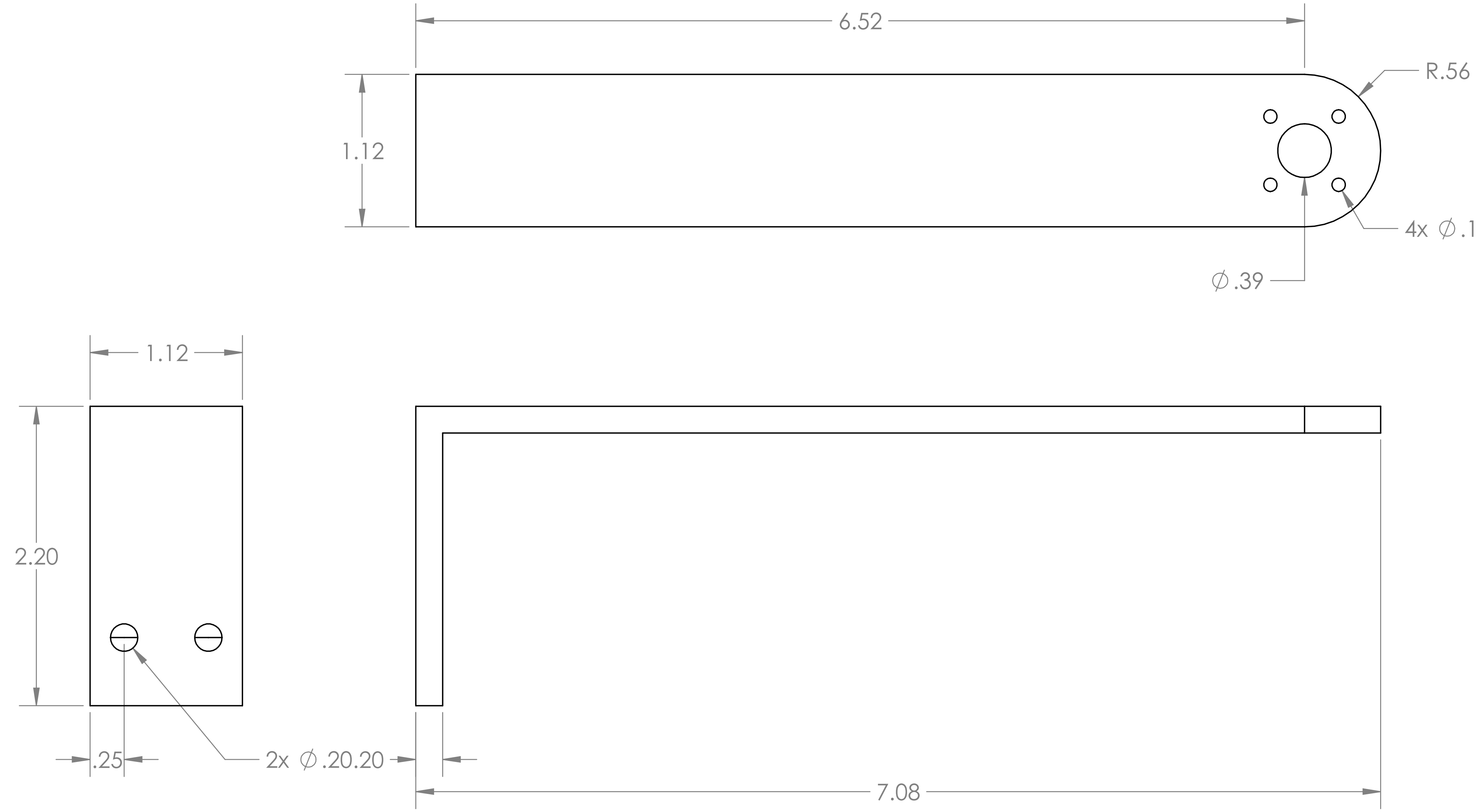
Boston University

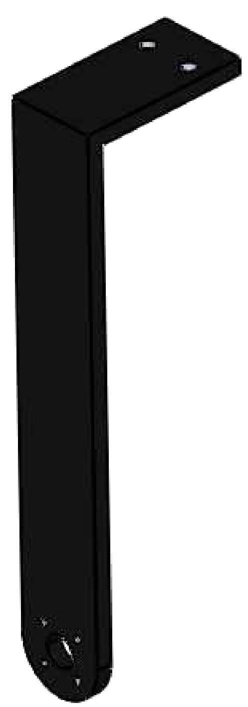
TÍTULO:
Platform Assembly Propellers

TAMAÑO B | N.º DE DIBUJO 8 | REV

ESCALA: 1:12 | PESO: | HOJA 1 DE 1

13.6.9. Propeller mount





INFORMACIÓN CONFIDENCIAL Y DE MARCA

LA INFORMACIÓN INCLUIDA EN ESTE DIBUJO PERTENECE EXCLUSIVAMENTE A <NOMBRE DE LA COMPAÑÍA>. QUEDA PROHIBIDA LA REPRODUCCIÓN TOTAL O PARCIAL SIN EL PREVIO CONSENTIMIENTO POR ESCRITO DE <NOMBRE DE LA COMPAÑÍA>.

SIGUIENTE ENSAMBLAJE | UTILIZADO EN

APLICACIÓN

SI NO SE INDICA LO CONTRARIO:
LAS COTAS SE EXPRESAN EN PULGADAS
TOLERANCIAS:
FRACCIONAL ±
ANGULAR: MÁQUINA ± PLIEGUE ±
2 LUGARES DECIMALES ±
3 LUGARES DECIMALES ±

INTERPRETAR TOLERANCIA GEOMÉTRICA POR:

MATERIAL: Plastic

ACABADO

NO CAMBIE LA ESCALA

	NOMBRE	FECHA
DIBUJADO		
VERIFICADO		
INGENIERÍA		
FABRICACIÓN		
CALIDAD		

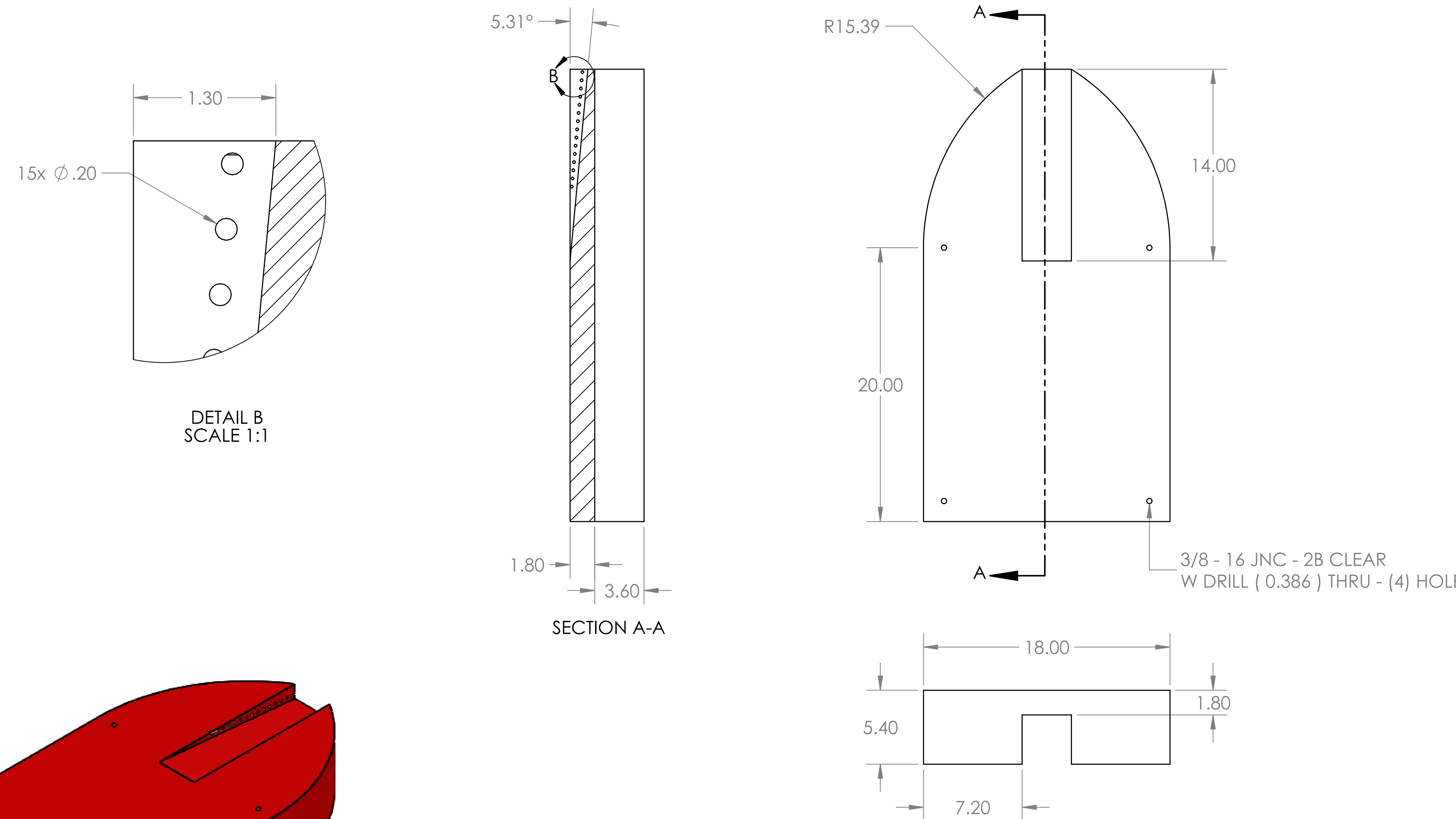
COMENTARIOS:
Boston University

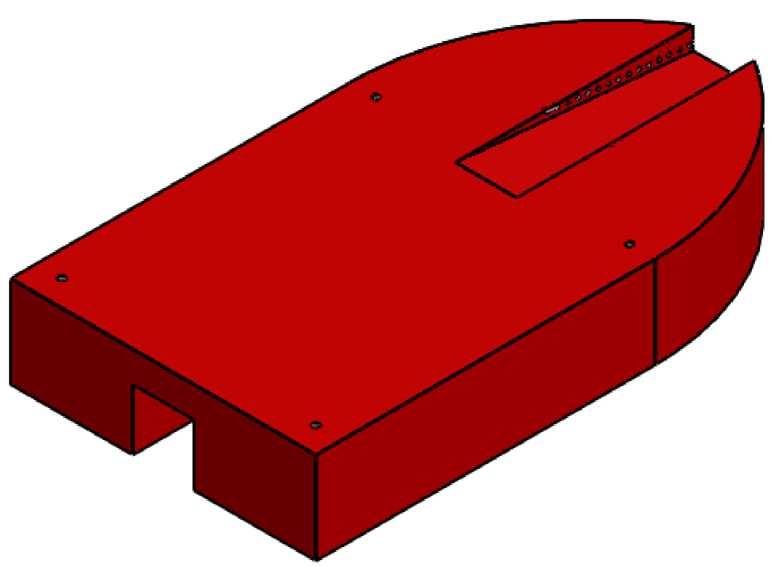
TÍTULO:
Propeller Mount

TAMAÑO: B | N.º DE DIBUJO: 9 | REV

ESCALA: 1:1 | PESO: | HOJA 1 DE 1

13.6.10. Platform for propellers





INFORMACIÓN CONFIDENCIAL Y DE MARCA

LA INFORMACIÓN INCLUIDA EN ESTE DIBUJO PERTENECE EXCLUSIVAMENTE A <NOMBRE DE LA COMPAÑÍA>. QUEDA PROHIBIDA LA REPRODUCCIÓN TOTAL O PARCIAL SIN EL PREVIO CONSENTIMIENTO POR ESCRITO DE <NOMBRE DE LA COMPAÑÍA>.

SIGUIENTE ENSAMBLAJE | UTILIZADO EN

APLICACIÓN

SI NO SE INDICA LO CONTRARIO:
LAS COTAS SE EXPRESAN EN PULGADAS
TOLERANCIAS:
FRACCIONAL ±
ANGULAR: MÁQUINA ± PLIEGUE ±
2 LUGARES DECIMALES ±
3 LUGARES DECIMALES ±

INTERPRETAR TOLERANCIA GEOMÉTRICA POR:

MATERIAL
XPS waterproof foam

ACABADO

NO CAMBIE LA ESCALA

	NOMBRE	FECHA
DIBUJADO		
VERIFICADO		
INGENIERÍA		
FABRICACIÓN		
CALIDAD		

COMENTARIOS:

Boston University

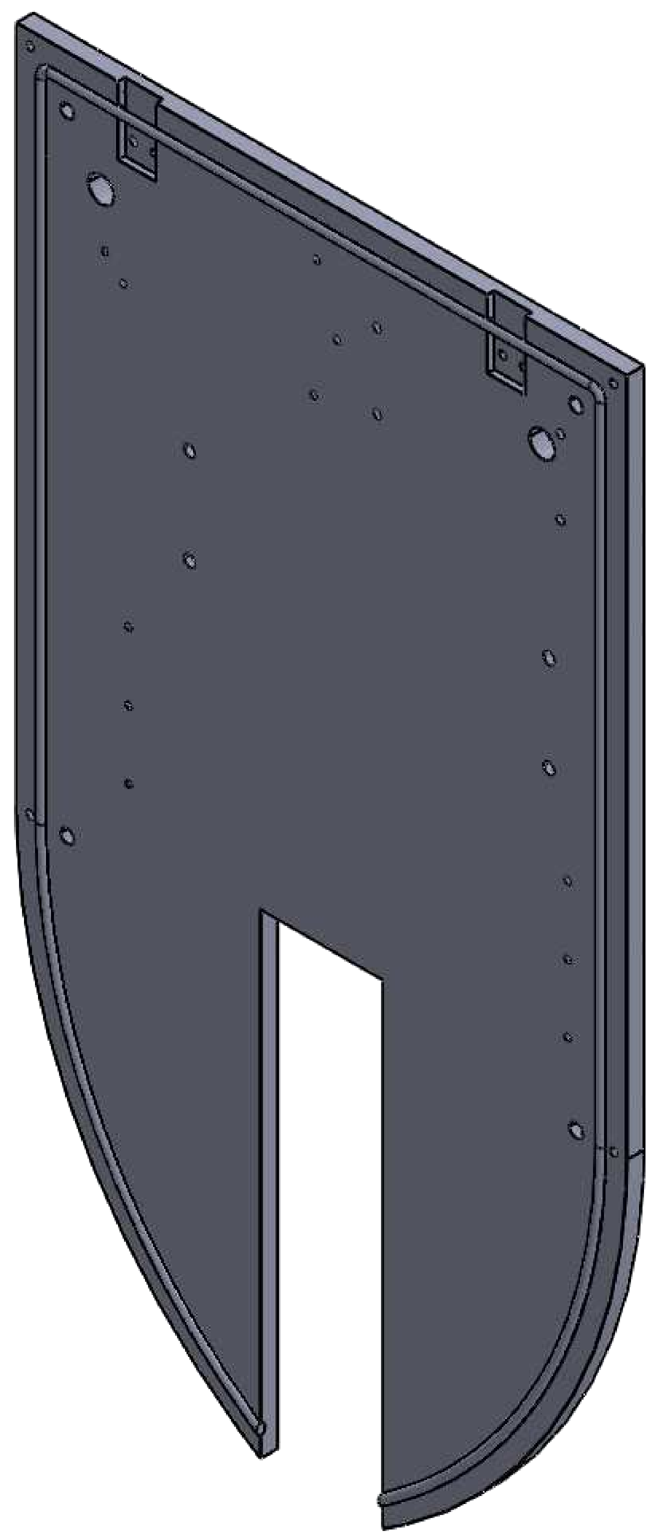
TÍTULO:

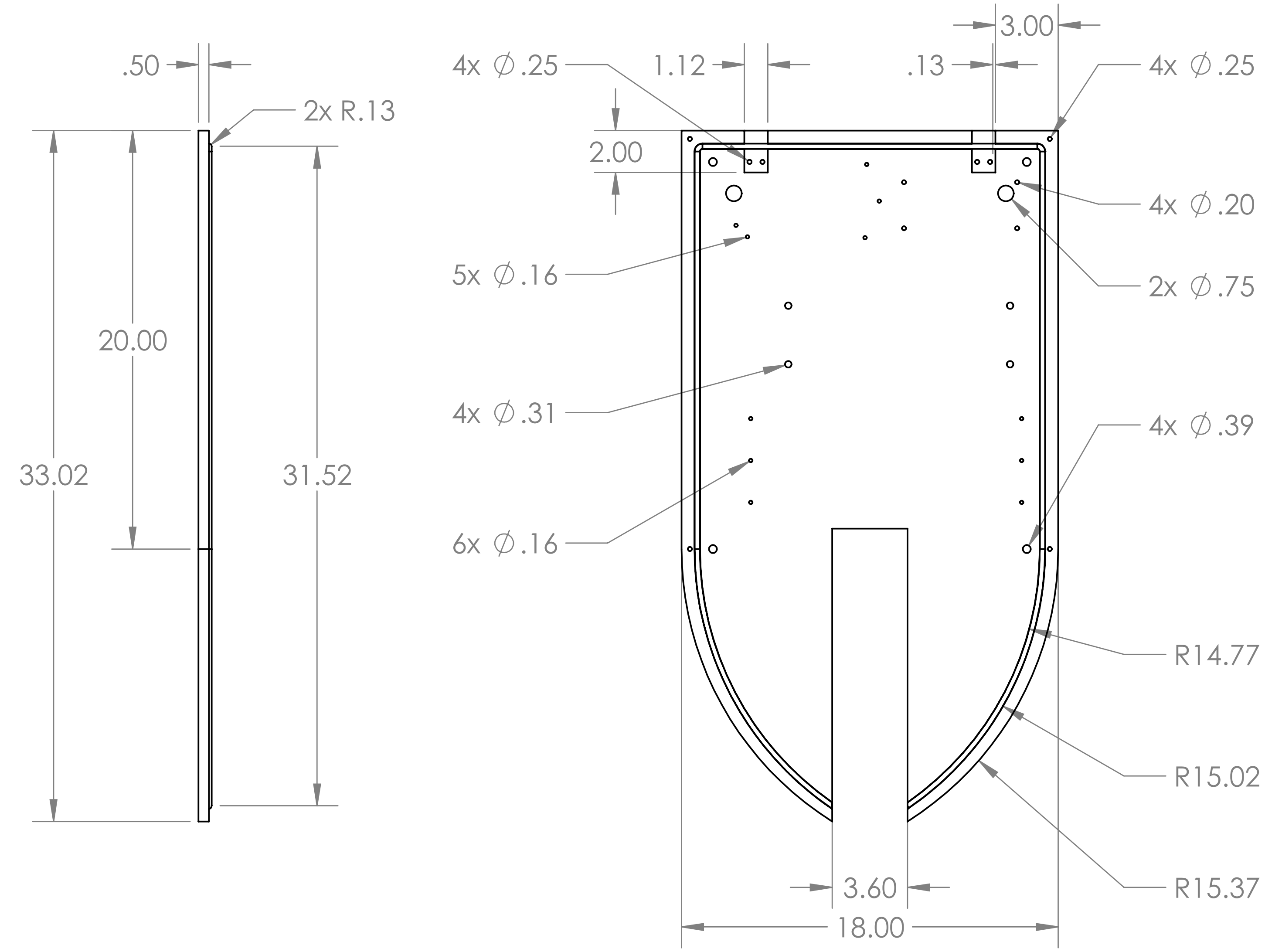
Platform Propellers

TAMAÑO	N.º DE DIBUJO	REV
B	10	

ESCALA: 1:8 PESO: HOJA 1 DE 1

13.6.11. Acrylic piece for propellers





SIGUIENTE ENSAMBLAJE | UTILIZADO EN
APLICACIÓN

SI NO SE INDICA LO CONTRARIO:
LAS COTAS SE EXPRESAN EN PULGADAS
TOLERANCIAS:
FRACCIONAL ±
ANGULAR: MÁQUINA ± PLIEGUE ±
2 LUGARES DECIMALES ±
3 LUGARES DECIMALES ±
INTERPRETAR TOLERANCIA GEOMÉTRICA POR:
MATERIAL: Acrylic
ACABADO
NO CAMBIE LA ESCALA

	NOMBRE	FECHA
DIBUJADO		
VERIFICADO		
INGENIERÍA		
FABRICACIÓN		
CALIDAD		

COMENTARIOS:
Boston University

TÍTULO:
Acrylic Platform Propellers

TAMAÑO: B | N.º DE DIBUJO: 11 | REV
ESCALA: 1:8 | PESO: | HOJA 1 DE 1