University of Central Florida

Depart of Electrical & Computer Engineering

EEL4915

Senior Design II

Silent Aluminum Fishing Boat

Senior Design Group 5

Tyler Brown, Electrical Engineering

Carley Camarotti, Computer Engineering

Alec May, Computer Engineering

John Santiago, Electrical Engineering

Sponsors of This Project

Correct Craft

Watershed Innovations

Torqeedo

SeaDek Marine Products

Contents

Figure Index

Table Index

1.0 Executive Summary

This project report is based around the design of an electronics and tracking system for an aluminum fishing boat that partakes in either recreational or sportsman activities. The various sensors and electronics on board the fishing boat allows for monitoring of various parameters that go hand in hand with operating a boat. These parameters are displayed to the user of the boat in real time via wireless communications to a mobile app and adjustments can be made as such. A detailed list of functions and features about this project can be found in the specifications section of this report. This project is collaborative effort that includes two other teams from the mechanical engineering and computer science departments. The mechanical engineering team piloted the design of the boat itself while the computer science department worked closely with software and databases needed to provide the best user experience possible. This electrical and computer engineering team wanted to use senior design as an opportunity to create something entirely on our own and bring all three of these designs together in one collaborative project that could have the potential to fully reach the consumer market.

The systems of this fishing boat were designed with ease of use, access and safety of the user and electronics all in mind. The main point of the project is to enhance the boating experience by using technology, but not take away from the fun of being out on the water. The most integral part of the devices on board is the GPS tracking system which allows the user to mark locations that are prime for fishing such that they can come back to the area another day. The tracking also allows the user to view their location and where they have traveled from live on the mobile app map. If a user is lost within a waterway they can clearly and easily find their way back to where they started by following their own trail. Other various sensors are placed around key locations within the boat such as the water temperature sensor on the stern of the boat to transmit the temperature of the water that day. There is also barometric pressure and accelerometers located in the center console since those are important for the users fishing experience

Contained within the following report is a fully documented design process of the system. In the preliminary sections of the report one will find the motivation for working on this project and all featured goals this project aimed to achieve. Following this there is a lengthy section of design requirements that all had to be accomplished either as personal goal of the team or goals from the sponsors of the project. After this there is an entire section dedicated to researching components that is used within this project and comparing them to components that were not chosen to be used on the design, these components include but are not limited to the MCU, the power components, wireless communication module and more.

2.0 Project Description

Almost all at home consumer devices in the modern day come with some form of smart features, with this being stated there is no reason why something as abstract as the world of boats shouldn't be able to incorporate these features in their designs as well. A user can not only enjoy the amenities of being out of the water and in nature but with this project they can now enjoy modern features on an otherwise empty platform like an aluminum boat that before only included a throttle and steering.

The discussion contained in the following sections is as follows:

- The group's main motivation for participating and engaging on this project
- The main objectives to be accomplished with this project
- Discussion that was had during a meeting with the sponsors of this project.
- Communication and deliverables to the sponsors of this project.
- An in depth listing of all requirements set forth by the group that must be closely adhered to during the design and fabrication phase

2.1 Motivation

Initially seeking an interdisciplinary project, our group decided that the 'Quiet Aluminum Fishing Boat' sponsored by Correct Craft would provide a challenging and unique experience do demonstrate our hardware and software skills. With one of the Electrical students having a strong interest in power and the others having experience with marine electronics, the electrical aspect of this project was of strong interest. The software and programming requirements were also of great interest to the Computer Engineer members of our group, as Bluetooth capability and interfacing between various boards for sensor measurements and GPS offered a unique opportunity to design an engaging user experience aboard this craft.

It was important to our team that we ensured new experiences and new subjects were explored as we were interested in using our project as a way to learn skills we did not have the ability of learning in course work. Working with Mechanical Engineers and Computer Science students was also of interest to our group, as this would afford us the opportunity to perhaps glance into another side of engineering not seen with our tailored coursework and broaden our horizons with the intensive material design aspect of this project. With this interdisciplinary project came a confirmed sponsor, Correct Craft, which provided guaranteed funding of $1500 and would enable us to design a high-quality electronics/software system with a degree of financial freedom. Collaborating with Correct Craft would give every group member a new experience of working and consulting on a client-based system. With the requirements that Correct Craft requested, this allowed each team member to understand the flow and process of working with a client-based project. Upon consulting with the sponsor, further reassurances were made with regards

to this financial freedom and our group gained a greater understanding of the design specifications which were being sought with this project.

2.2 Objectives

The "system" refers to the boat itself but more specifically the electronics and sensors on board, along with its mobile app interface.

- The system shall be safe to use and will have protection measures. Given that this is a water-based project safety is the most important factor during the design process and during use. The voltage used to power the motor and electronics is dangerous and the system will have short circuit protection or water damage detection such that it will shut off in the event of an emergency.

- The system will have low power consumption such as to not interfere with the motor drawing its power from the battery or reducing runtime by a significant amount. The microcontroller, GPS and various sensors will be designed or chosen in such a manner that the outputs are reliable and advanced while the power consumption remains low.

- The system's GPS and tracking capabilities as well as its sensors will be accurate. The accuracy and timing of the GPS on board the boat is imperative for the end user to have a reliable source of information such that they may find their way back to a starting point or desired fishing location. The sensors must also be accurate such that the end user can have viewing of various parameters throughout the day to determine whether spots are useful for recreation or not.

- The system will be low cost while maintaining good production quality. The cost of the boat itself without this system can already be quite an expense for the consumer, this project aims not add much to the final cost of the boat such that it is appealing to the consumer to purchase this boat with its enhanced features.

- The system will be user friendly and intuitive to use. All electronics can be turned on along with the motor via a simple switch. The mobile app that will be paired with all the electronics and sensors on board will allow easy navigation through its pages and measurements such that the user can find what they're looking for quite easily. All displays, and values will be clear and concise.

- The system will employ wireless communications via Bluetooth connection in order to send all of the data to the mobile application. If possible, the system will also employ a 4G LTE connection in order to have remote connection.

2.3 Company Sponsor Meeting

Following the week after choosing to sign onto this project there was a meeting held at one of Correct Craft's subsidiary company locations which is Watershed Innovations world headquarters that included all three teams assigned to this project where we would

all introduce ourselves and discuss more in depth about the specific goals of the project and give a brief time table about how tasks can get completed. There it was learned that not only would Watershed Innovations would be sponsoring the project, but we would also be receiving sponsored products to use for our project from Torqeedo and SeaDek Marine Products. Watershed Innovations mainly supported the mechanical engineering team with the development of the boat itself, as well as helped them manufacture and produce the design that they created which was the main task of their project. Our electrical and computer engineering team received direct support from Torqeedo by them providing the boat motor that was used for the final product. Torqeedo also agreed to provide the two batteries that was used to power the motor and our entire microcontroller and electronics design. SeaDek Marine Products provided their non-skid mats as a form of sound deadening and insulation once the design of the boat and hull was complete and manufactured. Overall, the initial meeting was informative and a good introduction to what all of the sponsors would like to see produced from this project which will be given in detail during the respective section of the report.

2.3.1 Sponsor Communications

In any senior design project or large project such as this communication is essential to assure everyone involved is up to speed on the project progress as a whole and that work is actually being completed as it should be. Communication between our sponsors and the University of Central Florida Mechanical, Electrical & Computer Engineering, and Computer Science teams was essential in order to provide information on the progress of each team and to report any issues or misunderstanding as they arised. During the first meeting in person with Watershed Innovations, the sponsors discussed using an application called Microsoft Teams in order to allow all team members within all the interdisciplinary teams to discuss their ideas and post progress updates, files and photos as the project moves along during both semesters. This allowed for an easy way for all individuals involved in the conference call to share their screens in order to share information fluidly amongst each other and also to have the convenience of speaking on the conference call either by phone or from the comfort of their home computers.

A weekly video conference meeting with sponsors, advisors and team leads was held every Friday for the opportunity for each interdisciplinary team to inform the others what they have accomplished. Any updates or announcement that the sponsors or team leads had were usually discussed in these meetings. This form of communication was vital to the flow of our project as requirements and specifications are easily changed. Aside from weekly video conference meetings, occasional on-site meetings were held at the Watershed Innovations world headquarters. These in-person meetings were the opportunity for each team to present what they created to the sponsors as well as to other interdisciplinary team. Not only was this a way of updating other team members of our success and failures, but it was also the opportunity to receive feedback or updated requirements from the sponsors themselves. This cycle made mistakes within the project happen much less frequently.

2.4 Specifications

The “system” refers to the boat itself but more specifically the electronics and sensors on board, along with its mobile app interface.

- The system will enhance the user boating experience regardless if the boat is used in recreational, sportsman or other activities.
- The system will not interfere with wildlife such that it will hinder the user boating experience, meaning extreme disturbance of habitats through wake and loud noises will not be present. All the electronics and sensors will not cause harm to the environment if they must be placed in water.
- The entire system will be powered via the same batteries that power the motor, ideally the consumption will be less than 3A such that lifespan of batteries is not significantly reduce if at all.
- The system will include several environmental data such as ambient air temperature and barometric pressure that will be routed back to the mobile app for viewing by the user.
- The system will include vessel information such as depth, bilge, proximity, and Live well data. The data will be routed from these sensors to the mobile app for viewing by the user.
- The sensors shall be secluded in a weatherproof body to assure accurate readings and lengthy life-span.
- The system will transmit its data over Bluetooth with a wide enough range such that the user can be anywhere along the boat and still be able to reliably receive data.
- The entire system will be contained inside a small water and weatherproof box of less than 12” x 12” such that it will not take away the already limited space of a boat.
- The entire system will adhere to the standards and codes set forth to ensure the safety of the end user

2.5 House of Quality Analysis

For many projects, especially one such as this, where it was being created from the ground up it was important to always keep track of specific tradeoffs that can be made related to the market and the engineering aspects of the project. The Figure below is called the house of quality and its purpose is to help visualize trade-offs between requirements mandated by engineering aspects and those mandated by the consumer themselves. The marketing requirements shown in the house of quality depict what a consumer would be looking for

in this product. These can be things such as cost, quality, ease of use and overall design and look of the project itself. The consumer ideally will want an easy pick and go type of system and will not want to hassle with setting up complex programs in order to make the system run, therefore overall the system must be simple. Consumers would also still like to enjoy the current cost of small boats currently available on the market, but with only a small cost induced by adding our system onto the boat or perhaps costs can be cut elsewhere, and it will remain the same. Finally, the consumer will like the system to be durable and safe at the same time, given that the entire system is based around a water application it is imperative that even if a critical failure were to occur it will not harm the user and ideally the system can save itself. General comparisons between these and the general engineering requirements of this project can be seen in the house of quality below. Regarding the engineering requirements for the house of quality, there are several key factors which will be assessed. These key factors are applicable for different components aboard the system. The boat hull for example is expected to be compliant within the weight requirement of 400lbs and be within the dimensions of seventeen by five feet. The MCU and PCB are of greater scrutiny however, as they have several requirements to be met. Cost is expected to be less than $1500 for this system and interfacing, the efficiency is expected to less than 10W (Regarding Power Usage), and it is desired to fit in a six by six-inch housing unit of weight less than one kilogram. The batteries, while provided from the sponsor, do have requirements to meet. They are expected to use 10kW of output power and weigh 36.5kg or less.

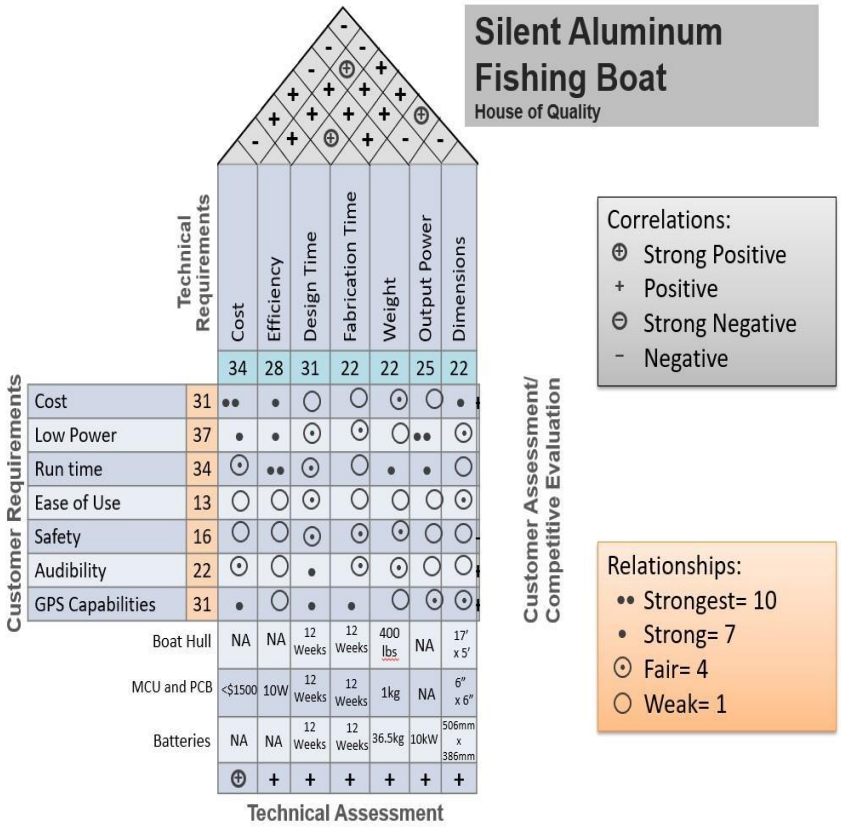


Figure 1: House of Quality

2.6 System Block Diagram

This diagram shows the high-level components and what other parts that they interface with. The initial owners of these pieces of the project are also shown to help clarify where responsibility falls. “Given” refers to pieces of the project that were provided by our sponsors, and that we were only be able to interface with rather than design ourselves. “CS” refers to the Computer Science sub-team and their work. As the project progresses this block diagram is subject to change as workload will be distributed evenly amongst all members of the team. In general, however, the division of teams with regards to the electrical and software is accurate and adjustment will only occur in the integration phase if it is needed at all. Another potential limiting factor regarding the implementation of this block diagram is the pace at which the mechanical, CS, and sponsor teams are capable of producing the needed material. This is to say that some sensors may not be ordered or tested for implementation before a CAD model of the boat design is produced so as to mitigate compatibility issues as well as ensure that the financial side of this project does not exceed the budget needlessly.

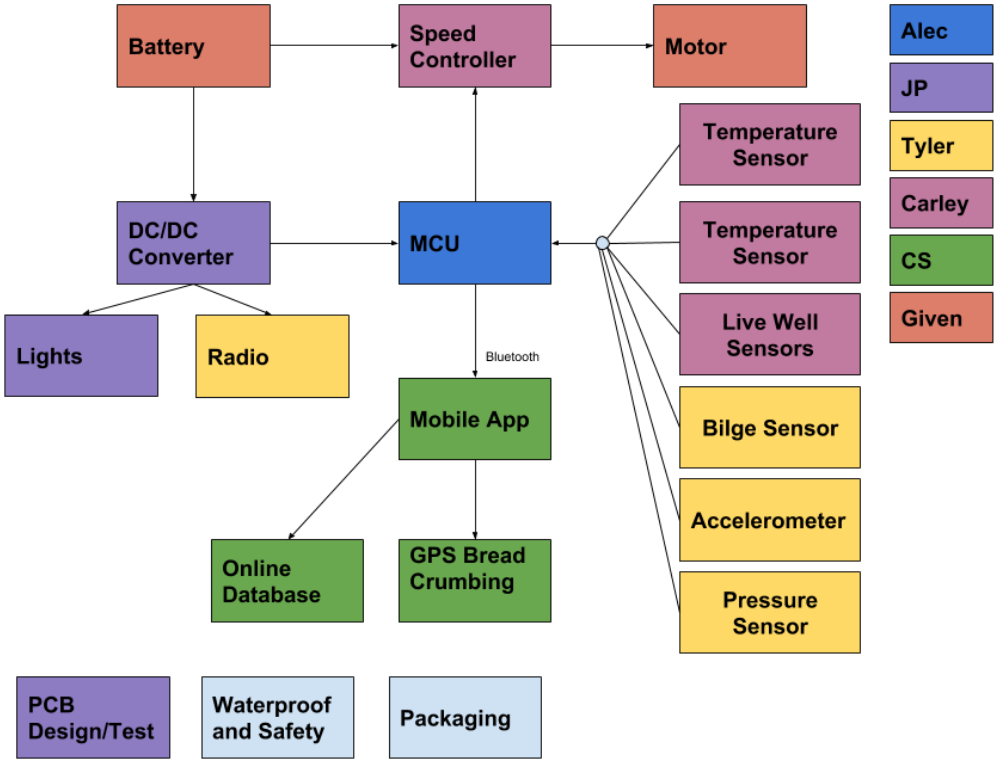


Figure 2: Block Diagram of Team Responsibility

2.7 Operation Manual

As the end user receives the boat will all the hardware and necessary components already pre-installed and mounted onto the boat there is no installation guide needed for our project. This is a great advantage because the end user doesn't have to deal with the hassle of running their own wiring and ensuring all the components are placed properly. It wouldn't be a fun idea to have the end user have to do more work to enjoy a system that can easily be mounted onto the boat via the manufacturer and run the risk of them wiring something improperly and potentially ruining our electronics and/or causing bodily harm to themselves. In this user manual we will discuss all the steps from turning on the boat for the first time during the fishing day all the way to powering down everything safely and saving data that was recorded, and everything in between.

2.7.1 Startup Procedure

Ensure that all belongings are secured within the boat to prevent unwanted damage or movement and that the appropriate life jackets and safety materials are on-board the boat and inspected before even attempting to go out onto the water. Ensure that the user has completed a communications check with the radio on board in the event the user doesn't have a cellphone or other method of communication that they are bring along with them. Once all the aforementioned are complete, the user should inspect the boat physically and ensure there is no damage or unseen hazards around the body of the boat that could potentially cause issues while out on the water. After these steps are complete the user in ready to turn on the battery system via the button switch that is located in the center console this will make the batteries go into an 'on' state, the batteries will have a small startup procedure in which they blink a sequence of lights to ensure that they are ready for operation. Once the battery lights are either solid green lit or blinking green lit they are ready for use and are awaiting a load i.e. the motor or pumps.

If the batteries are blinking red or are lit red then that means the batteries have detected a short circuit or some error within the wiring and are in a protective state to protect themselves, the user and all the equipment on board. If this scenario happens the user should immediate turn off the battery switch to ensure nothing accidently short circuited for a long period of time and for their own safety. The user shouldn't attempt to diagnose the issue on their unless it is clearly obvious that a connection is broken, or a fuse has blown, and in that case, they should bring the boat to a reputable servicer.

Continuing along, the batteries are in their green state the user is ready to turn the master switch located inside the battery compartment as well that will deliver power to the motor and the rest of the electronics on board. After this master switch has been turned to the "on" position the user should take care to observe the battery lights one last time and ensure that they are in their green positions and the boat is ready to be used.

2.7.2 Checks Before Hitting the Water

Now that the battery and power system has been initialized and turned on. The user should take care to perform the following checks described before hitting the water. Firstly, the user should give a small amount (~5%) of throttle to the motor via the throttle control on the center console in the both the forward and backwards direction to ensure the motor is properly spinning and receiving power. Care should be taken to ensure the prop isn't in danger of colliding with the ground or a trailer piece during this check as this will easily destroy the prop. Once this check is complete the user should draw their attention to the center console's switch panel which allows for the control of the bilge pump, live well pump and aerator and the navigation lights. The user should flip each of the switches and ensure that all of the pumps activate and that the navigation light at the front of the boat becomes illuminated. The user should take care in the fact that this is only a test to ensure the pumps are working, they should not be left 'on' for more than a couple seconds as running the pumps dry with no water can cause them to burn out. Finally, the user should take a look at the electronics box within the center console that houses our microcontroller and sensors and ensure the PCB has a green LED light on indicating successful power to the board and no errors are occurring while our microcontroller is running and gathering data.

2.7.3 Connecting Your Smartphone to the Boat

Now that the boat is ready to hit the water and the procedures in the previous sections have been completed it is time for the user to pair their smartphone to the boat. Along with a green light for power being received by the board, the GPS should soon start emitting a blue blinking LED to inform the user that there is a lock onto the satellite to receive data and the Bluetooth module should begin blinking its red LED. On their mobile device they should enable Bluetooth communications and connect to the Bluetooth source named 'CorrectCraft'. Once the phone has a connection to the Bluetooth signal, the Bluetooth module's LED should stay lit and should no longer be blinking inside the waterproof box within the center console. In the event of the connection not being completed, the user should look double check to ensure Bluetooth is enabled on their smartphone and try reconnecting, otherwise is there is a hardware error with the Bluetooth the red LED inside the waterproof box will being to blink meaning something has gone wrong with the module itself and it is no longer being recognized. Once the smartphone has its connection, the user may navigate to the 'Watershed' application and data will begin to be displayed on the appropriate pages.

There are several different screens that users will experience when utilizing the mobile application. The first screen that will greet the user is the sign in screen. This screen, shown in figure 5, will allow users to sign into their account via google accounts while looking at a pleasant screen that shows our sponsor's logo. The login screen will utilize the user's Google account to sign into the mobile application. The next major page is the home screen shown in figure. This is the page that would serve as a digital gauge cluster for the driver of the boat during normal operation. This screen includes relevant sensor data such as speed and heading for navigation, as well as data such as temperature and pressure that affect fish activity. The weather page is also very critical for the user's experience. The weather app helps the user to recognize how ideal the current conditions are, as well as how long the conditions will stay ideal. Factors such as temperature, tides, wind, and humidity all affect how active the fish are. The tides specifically in saltwater conditions play a large role in the user's safety in shallow water conditions in addition to tracking potential fish activity. In general, this allows the user to monitor these conditions will aid in maximizing their productivity and catch rates while fishing.

Figure 5: App Login Page

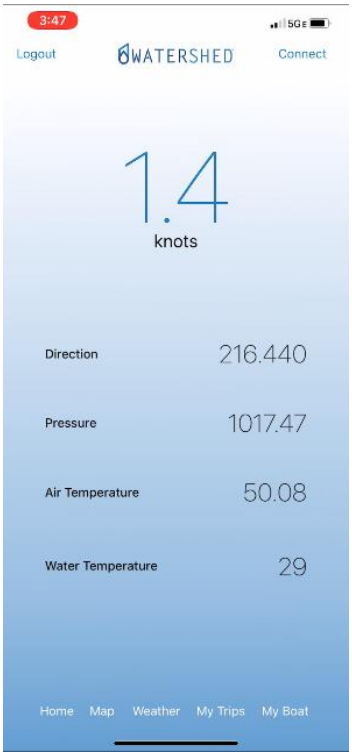


Figure 3: Sensor Readouts

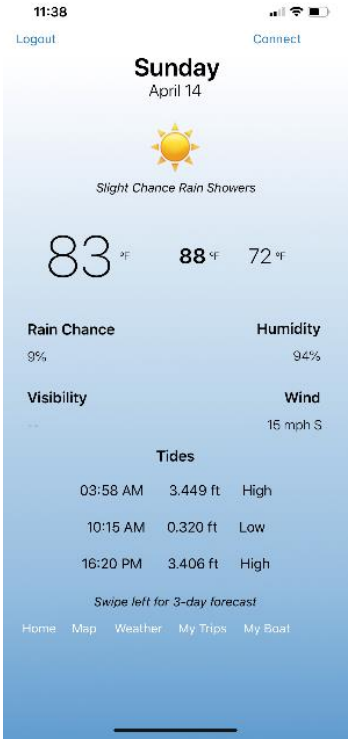


Figure 4: Weather Page

The last major page is the trips page. The trips page allows the user to record and review any trips that they take on their boat. This allows the user to backtrack if they get lost, which will prevent them from running out of battery power while they are out on the water. Keeping track of trips allows the user to remember what areas that they have recently been

to, which would help the user to discover new areas to fish. The user can also leave notes about each trip, so that they may remind themselves what that location was good for. They can use the notes to discuss what fish were in that area, what wildlife they saw, features of the landscape, or how the tides affected the area. The user can also drop pins to mark that a fish was found at a specific location. This can be used to document how many fish were caught at a location, and what type they are. This creates an objective way to compare the success of one area vs the other and allow the user to assess a given areas production of specific species of fish. The trip page is seen in figure below. Since owning a boat is a huge accomplishment, many owners will want to share this with others. Allowing the owner to be able to share the boat with others is a key requirement in order to make the end product viable. In order for another user to be able to use an owner's boat, the user must create an account. The owner can then give them a code to allow their data to be stored on the owner's page. When the new user does anything on the boat, their data will be stored on both their account and the owner's account.

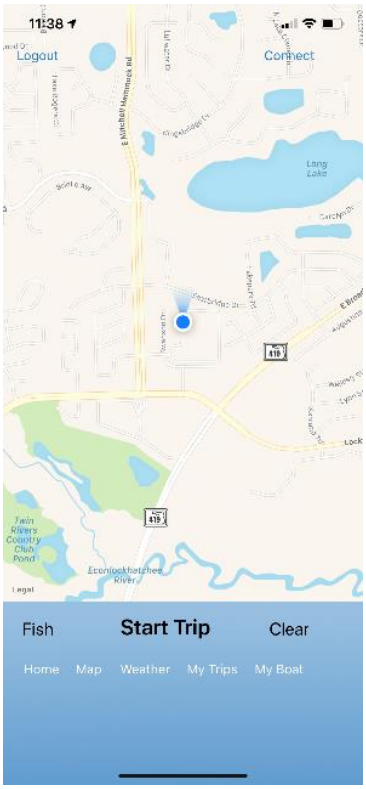


Figure 6:Trips Page

2.7.4 Shutting Down the Electrical System

After a full day of the user being out on the water and enjoying the amenities and hopefully having success while fishing, it is imperative that all systems are shutdown properly and safely. After returning to the dock and preparation for trailering the boat back home begin, the user must first turn off the battery button in the center console, when this happens the batteries should remain in the 'on' state for a couple of seconds and then shutoff completely. After this is complete the user should also go into the battery compartment and turn the master switch for the motor into the 'off' position, such that in the event of someone accidently pressing the battery button the motor will not receive power and have a danger of spinning while obstructed. All electrical systems within the boat should be powered off now and can be confirmed via checking that the battery state LED is off and not lit with any color. Now the user should take care to put away the navigation lights back into their respective compartments for safe keeping if they were used that day out on the water and removing the safety from the throttle, so it cannot be activated anymore. The user should finally close all hatches and compartments and move the handles into the locked setting after removing any belonging they wish to take themselves. If the batteries are drained quite heavily through the days use, the chargers can be accessed through the battery compartment as they are mounted inside and can be plugged into a standard wall outlet to charge overnight for the next trip at the user's home.

3.0 Research Related to Project

In order to begin to develop the system that will accomplish all tasks within the project has set out to achieve we must first research all component types and possible technologies that can be incorporated into the project. This research and selection is crucial to the project such that in the later stages time is not wasted replacing or repurchasing new components because of malfunction or incompatibility. The main research done within this section of the report deals specifically with the hardware side of the project these include the microcontroller (MCU), various sensors that were used to monitor specific parameters, communication module and protocols that were used to transmit data as well as discussion on PCB design and fabrication.

3.1 Similar Designs on the Market

Currently, there are no mass produced electric powered aluminum fishing boats which measure up to the design goals of this project. To this extent, the design and production of this boat certainly offers an opportunity to reach a market which has yet to be tapped. However, while a similar system/boat as a whole has not been produced, various marine electronics companies have been integrating electrical components to be used with wireless communications. Companies such as Lowrance and Raymarine offer a level of connectivity with their electronics, albeit at a large introductory and maintenance cost. The goal of this project to use low cost electronics with a smartphone and not require a high cost GPS unit to act as a hub for sensor readouts is largely unique and benefits the user as it permits much more flexibility with device cost and use.

3.2 Microcontroller Options

The following section will compare various microcontrollers that may be used for the project. A microcontroller or MCU is the brain of our entire electronic and sensor system all the data is fed into the MCU and then released depending on the communication system chosen in its own section. Various manufacturers have developed MCUs for all types of projects in the modern era, the main criteria of choosing the MCU would be whether it allows full integration of our main goals and whether the hardware implementation is feasible. This means that the MCU chosen should be able to be fully interfaced both in the hardware and software world without conflicts depending on the types of external devices used to measure data. The goal of this section is to choose two microcontrollers from the following list of microcontrollers and do a final selection.

3.2.1 CC1352R1

The Texas Instruments CC1352R1 wireless MCU is one of the newest products introduced to their line of powerful QFN microcontrollers. At first glance this MCU appears very intimidating as there's not much information out there in terms of using it in a practical design because the product is still new. One feature of working with this component is that is can be purchased mounted onto a development board such that it may be programmed and debugged within the code composer studio. The is especially useful because all of the

group members have experience working with the code composer studio environment especially when it comes to utilizing the environment in order to program embedded systems whether in C language and assembly language. On top of this Texas Instruments has also enabled this MCU to be used within their own Sensor Control Studio such that many different applications with specific sensors can be easily programmed for use within a custom environment dedicated to such devices. The development kit that includes the CC1352R1 costs $40 which is quite a lot, on top of this fact since the MCU is a surface mounted component and extremely sensitive it cannot be taken off the development board and used within a custom PCB. This means a development board must be purchased for learning and testing purposes only, and a standalone CC1352R1 must be purchased for an additional 10$ and placed onto the PCB and programmed on the PCB itself which is quite a task to achieve.

A huge benefit of this type of MCU is that it has wireless communication on board and it is ready to be configured depending on the external hardware components used in correspondence with the datasheet. This MCU is capable of producing 1 GHz and 2.4 GHz signals that can be used to communicate with other devices via Bluetooth generation 5 at quite a wide range. This is perfect because that means external communication hardware is no longer a concern because it is already built it. In terms of power consumptions while in its active mode the MCU only consumes 2.82 mA while running the clock speed at 48 MHz, all of this processing and communications power only requires around 3V from the supply which is great. Now in terms of general-purpose use with other devices and sensor that must be programmed to this MCU, it has 28 I/O pins and of course all 28 pins can be configured for digital devices right out of the box. As well as 8 of these 28 I/O pins are analog capable at up to 12-bits which is quite powerful and allows for quite a range of analog devices and sensors to be used in correlation with this device. This device can do all this while at the same maintained an incredibly small package at 7mm by 7mm size, as long as the external hardware on a PCB is proper it can be virtually placed with an extremely small footprint meaning more room can be used for other devices or just space saving in general.

Taking a good look at this QFN MCU has revealed that it is extremely powerful MCU with quite a lot of built in features that can ease up the hardware selections of this project. That being said there is a lot of downsides to this MCU, one main one being the fact and so many external components are needed with tight tolerances to make every module on board run properly. This can quickly clutter a PCB design and become a nightmare if something were to go wrong or if a component is placed somewhere it was not intended to be. The more components needed the more chances something has to fail and of course with a package this small it will not be an easy fix. Another downside is the cost of designing our system using a product like this, this CC1352R1 MCU is mainly to be used straight off of the development board because the product is so new. This means the group would be learning how to interact and properly control the MCU on a development board which is fine, but the downside is that the MCU cannot be taken off the development board easily and then placed onto a PCB once programmed. Therefore, in order to utilize this MCU in our system design we would virtually need to purchase it twice which will get quite costly. All things considered, while this is probably one of the smallest yet most

powerful QFN MCUs made by Texas Instruments it has too many large drawbacks when being implemented for a design such as ours, as such it was not considered.

Table 1: CC1352R1 Hardware Specifications

Component	Value	Description
Processor Clock Speed	48MHz	Very fast clock speed, can run even complex programs fast
Storage Amount	352 KB	Will fit even the most complex programs
Communications	1GHz or 2.4GHz Bluetooth	Communications already present within the chip
Temperature & Battery Monitors	Hardware integrated on board	Can monitor and report back environmental changes

3.2.2 MSP432P401R

The MSP432P401R is within the line of typical Texas Instruments microcontrollers and is derived from the entire platform of the MSP430X family of MCUs. Unlike the previously mentioned MCU, there is a quite a library of information and online resources regarding this MCUs. The main benefit of using this MCU is the fact that it is very similar to the MSP430 Launchpad which is used within the embedded systems laboratory at the University of Central Florida. All of the group has taken a course dedicated to programming this family of microcontrollers within the code composer studio environment and as such we are very familiar for how they work in both C and assembly languages. This entire family of MSP430X MCUs all operate in a similar nature and share quite a lot of the same characteristics. Purchasing an MSP432P401R chip itself would only cost $8 and purchasing its subsequent development board for testing purposes would cost an additional 29$. The number of external components in order to create a custom PCB is significantly reduced, meaning it would be feasible to purchase the standalone chip and create a PCB that allows debugging and programming.

Aside from this MCU already being user friendly towards the group because of the familiarity of its architecture it has some features that make it quite appealing. The power consumption of this MCU is also incredibly and it even has an ultra-low power mode that will only utilize 80 uA per MHz of speed needed. The supply voltage needed to power this MCU is also around 3 volts which is normal around these types of MCUs. While in its highest power mode the clock can operate at a speed of 48 MHz which is a speed that is perfect for the type of operations that must be performed. When discussing general purpose uses with other devices the MSP432P401R has 48 I/O pins which is easy to imagine because it is a 64-pin device. This also leaves quite a lot of room for analog enabled pins which can handle up to 16 bits of analog data and convert it into digital. This

type of 16-bit analog conversion basically opens the entire world of sensors and parameter measurers, because this board can freely use all those devices with no need for shifting bits within the program because of not having enough bits. This bit amount is more than enough to suffice when analog devices are being considered for hardware integration. Once more this device is also a small package coming in at 14mm by 14mm although not as small as a QFN it can be easily placed onto a PCB along with its required external components to function properly at the desired levels.

Table 2: MSP432P401R Hardware Specifications [2]

Component	Value	Description
Processor Clock Speed	48 MHz	Very fast clock speed, can run even complex programs fast
Storage Amount	256KB	Can hold quite a lot of programs within itself
Analog to Digital Conversion	16-bit high precision	Allows usage of powerful analog sensors with accuracy

3.2.3 MSP430G2553

Continuing on with the MSP430X family of microcontrollers, another popular choice of controller would be the MSP430G2 also made by Texas Instruments. The architecture and programmability of this MCU follows the same as the MSP432X models which allows it to be easily used within a multitude of projects. This is the specific model that was used within the embedded systems laboratory, so all of the group has experience with this MCU. This model of microcontroller is not a small surface chip like the two previously mentioned MCUs, it is actually a PDIP package or it can be offered in an TSSOP style package. This has its advantages during prototyping because it can quickly be socketed into a breadboard and other various components can be connected with it for testing. Meaning that this MCU can be quickly programing and debugged on its launchpad board and then taken off the socket and placed onto a prototype breadboard to test its functionality. The launchpad itself can be purchased for around $10 while the chip package itself only costs around $2, this is incredibly inexpensive for the amount of flexibility that one may achieve from such a small platform. Also, since the chip can be readily moved from launchpad to breadboard there may not even be a need to purchase standalone chips lowering the cost even further.

Moving on to the technical aspects of this MCU, for such a small platform it is actually quite diverse. As with almost all of the Texas Instruments microcontrollers, this is also a low power consumption only needing about 2.2 volts for its low power mode. The clock frequency of the processor can also be as high as 16MHz which although is nowhere near

the two previous discussed models, it is still good considering the type of package. If the group were to choose the 28 pin TSSOP package type, we would be able to fully utilize 24 general purpose I/O pins. Similar to the other MCUs 8 of these pins are analog capable up to 10 bits which allows room for plenty of analog devices that must be used. Considering the fact this is the entry level MCU constantly advertised by Texas Instruments, it is still quite powerful and capable on many types of operations depending on the application.

Table 3: MSP430G2553 Hardware Specifications [3]

Component	Value	Description
Processor Clock Speed	16 MHz	Average speed given the size of the package
Storage Amount	16 KB	Can still hold some complex programs
A/D Conversion	10 bit general purpose	Capable of ADC conversion
Cost	$10	Lowest cost of all MCUs researched thus far

Although the MSP430G2553 may be the lowest performing microcontrollers that was researched, that does not mean we should count it out. This MCU itself is still a good package to be working with during our project and even more so when discussing the PCB design factors that go into this chip functioning within a design. Overall this MCU based off of still maintaining a good I/O pin amount and low power consumption as a TSSOP package leaves it in good shape to be applied to a project of this scale. The only main downside of this chip is the fact that the processor it quite slow when it comes complex calculations and programs, as well as the memory is nowhere near the size of other MCUs discussed. Those two issues can be remedied with proper programming and memory management, that being said this left the MSP430G2553 as a strong candidate for consideration within our project.

3.2.4 ATmega1284

Steering away from the common Texas Instruments microcontrollers, we decided to begin to look into the Atmel AVR family of microcontrollers. The Atmel MCUs are widely popular within the technology and hobbyist community with tons of forums and backend support because of how many people have created interesting projects with them. As most students are aware of this is the main choice of MCU for Arduino boards, and Arduino is quite a popular brand with all hobbyists. Most of the team has experience working in the Arduino environment based on tinkering on small hobbies, and it is very user friendly and has a lot of community support. These MCUs are quite powerful for the low price point as

such the first MCU we have decided to investigate is the ATmega1284. This chip itself only costs about $5 with its respective development board only costing $20. Similar with previously mentioned MCUs, this chip is also a PDIP package meaning it is easily portable between development board and breadboard prototype testing.

The ATmega1284 does consume a small amount more voltage at 5 volts, but it still operations at a very small current per MHZ. At its active mode the ATmega only consumes 1.1mA which is small on its own, and at idle mode only consumes 0.35mA. The maximum clock speed achievable by the processor is 20 MHz which is still a decent amount of processing power as previously discussed. The package comes with 40 pins on which 32 of them are available as general-purpose usage I/O pins and all are digital capable. Of these 32 pins, 8 of them are capable of 10 bit analog to digital conversion such that they may use analog devices and sensors. Also, within the software itself, certain programming aspects with registers can be used if analog devices with more bits are needed to be interfaced. Given the fact that this MCU is used in so many Arduino based projects such as smart systems and smart sensors, it was a good idea to take this MCU into consideration for a project such as this where there were many sensors and peripherals needed.

This ATmega1284 microcontroller offers quite a competitive package based around a community full of projects and support. Overall this is a very attractive MCU that makes it a very strong candidate for consideration to be used within the project. In terms of a PCB design this is simplest of all possible designs when it comes into hardware integration, placing the ATmega32 on a PCB would only require about 4 external components to ensure it runs properly. The only downside of this chip would be its large memory size when compared to the other MCUs that have been mentioned, because they may seem a bit overkill and all that space may not be necessary. The most important factor was that the processor can handle all the calculations and manage of all the things that are going on, which at this speed it can and remain at low power consumption while still being able to interface with the tons of Arduino based sensors on the market. This left the ATmega as for sure a candidate for use within the project.

Table 4: ATmega1284 Hardware Specifications [4]

Component	Value	Description
Processor Clock Speed	20 MHz	Average speed given the type of package
Storage Amount	128 KB	Double amount of the previous package
Analog to Digital Conversion	10 bit ADC pins	Can still handle most analog devices
Speed modes	3 different speeds 4 MHz, 10 MHz and 20 MHz	Speed dependent on power supplied, programmable

3.2.5 PIC16F877A

The Microchip family of microcontrollers is also something that should be investigated further. The PICXX line of MCUs offered by them are some of the most popular MCU on the market because of their small package yet powerful design. Although the team is not familiar with the programming environment of the MCU, the architecture is similar to that of the MSP43X family of microcontrollers. These MCUs are also pin packaged which allows for easy portability, but better yet the package only costs $2.

Like most of the other MCUs mentioned this too is a low power MCU only needing around 2 volts to sustain good clock speed. The processor clock speed of the PIC also operates at the same speed as the ATmega which is 20 MHz which is good because that means other factors can be used to make a decision about this MCU. Similarly, since it is a 40-pin package, it has 33 general purpose I/O pins and of which 8 of these are 10 bit analog to digital conversion ready for use with analog devices. This chip also comes pre-enabled with bootloader software such that it can be placed into a development board for a different device but still be programmed and debugged in a specific environment which is quite a useful feature no other MCU has had stock already.

This microcontroller shares a lot of similar aspects as the ATmega MCU, it even has quite a lot of enhanced features on board that are not available with other MCUs outside of the box at least. That being said there is one critical flaw of this MCU being that its memory size is incredibly small, this does not leave much room for complex programming and calculations that must happen given the scale of this project. It is for this reason this microcontroller was not considered because although while a great package, it simply cannot store and process what we need it to.

Table 5: PIC16F877A Hardware Specifications [5]

Component	Value	Description
Processor Clock Speed	20 Mhz	Average speed given the type of package
Storage Amount	8 KB	Very small, is probably too small given the needs
Low Voltage Programming	less than 1V	Advantage for on the fly programming without test
Three on chip timers	1 16 bit and 2 8 bit timers	Useful for controlling different aspects of code

3.2.6 Microcontroller Selection

Overall the previous section has produced quite a lot of information based upon various microcontroller and their respective hardware parameters, all of this data is available in Table 6 below for an easier comparison of all of these components that are considered for the project. All of the microcontroller discussed in the sections above were great candidates for use within our project each in their own respect, but only one may be chosen for final use within the project. This is a decision that cannot be made lightly as the microcontroller dictates how all of the hardware and PCB design are completed during the end of senior design 1 and beginning of senior design 2. Since the microcontroller controls virtually all aspects from the goals and objectives that must be met for this project, those goals and objectives were always be in the back of the mind.

Table 6: Overall Microcontroller Comparison

Microcontroller	Memory Size	Clock Speed	Voltage	I/O pins	Cost (chip)
CC1352R1	352 KB	48 MHz	4 V	28	$30.00
MSP432P401R	256 KB	48 MHz	4.5 V	48	$7.72
MSP430G2553	16 KB	16 MHz	3.3 V	24	$2.59
ATmega1284	128 KB	20 MHz	5 V	32	$5.15
PIC16F877A	8 KB	20 MHz	4 V	33	$1.96

The previous sections shed some light on five microcontrollers based on their respective hardware specifications and interface capabilities that could have let them to be considered for use in this project. From this list we have decided to choose two and do a final comparison of these two based on multiple categories. The two microcontrollers that will be debated in the following sections are the ATmega1284 and the MSP430G2553, and the end of this section a final selection will be made based upon the data and comparisons presented.

3.2.6.1 Memory Size and Processor Clock Speed

When dealing with any type of microcontroller it is important take into account how much memory it has which can come in two main forms that are useful for projects that require calculations and allows for higher complexity of code to be used within the programs themselves. The first main type of memory is flash memory, meaning how much can the microcontroller chip store on-board within itself after programs have been written on to it. Flash memory also has the characteristic of being non-volatile meaning that even after the

device have been powered down it still retains its flash memory for future use once it is powered on again. The larger the flash memory storage, the more code we can fit inside chip itself this is especially useful to have a lot of overhead space so that programming doesn't have to be strictly contained due to memory constraints. The second main type of memory comes in the form of RAM or random-access memory that allows the microcontroller to retain data during calculations or important procedures, meaning it can analyze more data at one time improving the overall speed of programs written. RAM has the characteristic of being volatile, meaning this memory is only conserved when power is on to the device during programs running and is lost once power is turned down. Overall, it is beneficial to have more RAM such that the microcontroller can process more things at once.

Another important specification of the microcontroller that must be looked into is how fast the processor clock speed, meaning how fast the processor can execute its command and instructions. Of course, the faster the clock speed the faster the microcontroller can run the programs and there will be no latency or lag between completing large instructions or tasks within the programs themselves. Given that within the scope of this project it is necessary to run and collect the data from the microcontroller as fast as possible, a higher clock speed would be ideal in the microcontroller.

Table 7: Microcontroller Memory Size and Clock Speeds

MCU	Flash Size	RAM	Clock Speed
MSP430G2553	16 KB	512 B	Up to 16 Mhz
ATmega1284	128 KB	16 KB	Up to 20 Mhz

As can be observed from the above Table 7, the ATmega1284 vastly outperforms the MSP430G2553 when it comes to memory sizes and clock speed. In RAM size especially, the MSP430G2553 just offers no competition when compared the ATmega1284 as stated before having this amount of RAM as our disposal is a great tool. The ATmega1284 has the processor power and the memory capabilities desired for this project and as such it is considered a winner when it comes to this category.

3.2.6.2 Power consumption and General Purpose I/O Pins

When it comes to any project the main goal is minimize power consumption as much as possible while retaining a reliable design. In general, most modern microcontroller come in such small packages that the chip itself does not consume much power but this is still an important factor to take into consideration. Although we have the added benefit of having powerful batteries for our project, we still would like to minimize power consumption such that it does not cause hindrance to the motor which used most of the

power. The main goal is to keep the system online for the during of the fishing boat use and for a few minutes afterwards to send final data for the day.

Another important aspect of the microcontroller if how many pins it allows to be used for input and output. These go hand in hand with power consumption because microcontrollers have very different tolerances for the amount of voltage and current at their I/O pins, which could severely limit the performance or even compatibility with sensors and communications devices. Special care should be taken when considering how the pins can actively transmit and receive data, this is crucial for the wireless communication system which is responsible for doing this task. As well as taking into consideration how many pins are receive pins and how many are transmitting pins.

The table below illustrates quite clear differences between the MSP430G2553 and the ATmega1284 microcontrollers. As can be observed, it appears as though the MSP430G2553 actually can be operated at lower voltage within its active mode and can consume less power overall because the output current of each pin virtually half of what the ATmega1284 can produce. On the other hand, the MSP430G2553 has nowhere near the number of pins that the ATmega1284 has, this is vital to have as many pins as possible for all of the sensors and devices that must be interfaced within the design. The ATmega1284 does consume a bit more power, but overall it can provide the required current for a large range of higher-powered devices to operate from its pins and aside from this it has triple the number of I/O pins when compared to the MSP430G2553. All things considered, although the ATmega1284 does consume more power, it is still not even enough to cause a significant difference in overall consumption therefore the ATmega1284 is the winner in this category as well.

Table 8: Microcontroller power consumption and I/O pins

Microcontroller	Voltage (Active Mode)	Output Current in each pin	General I/O Pin count	RX/TX pin count combined
MSP430G2553	3.6V-5V	48mA	16	2
ATmega1284	4.25V-5.5V	100mA	32	2

3.2.6.3 Cost and Hardware Integration

Perhaps one of the most important aspects of all the ones discussed within these different categories is the cost of these microcontrollers. It is important to take note that buying a standalone microcontroller chip general is very cheap and inexpensive, when it comes to being able to program the board either through a development board or custom program board that has to be made, that is where things can get higher in cost. Along with just purchasing the microcontroller itself we must also take into account the cost of the external components if any that are required to get the chip to run on its own.

Also discussed within this category will be the practicality of using the microcontroller in our own PCB design which is a requirement for this project. Of course, these microcontrollers must be programmed within a development board and then be able to be placed on a PCB design, it is for this fact that the small surface mount non-DIP package microcontrollers have been left out. A development board comes with all necessary hardware needed to flash the program onto the chip and allow it run, but of course a development board cannot just be implemented into the final design. As such once it comes time for the final implementation of the board, the microcontroller than has less external components to worry about would be the wise choice, because it allows for less error possible when designing the final PCB.

As can be inferred from the below table the MSP430G2553 is the lowest of cost in terms of standalone chip and development board needed for programming. This is one of the few areas that it wins over the ATmega1284, solely based on pricing. If the budget provided for this project was more constrained, then this could've caused an issue. In terms of hardware integration, the MSP430G2553 requires quite a lot of external components to get it running on a custom PCB, multiple different value capacitors and inductors need to be tied to certain pins in order to control the internal devices on board the chip. The ATmega1284 on the other hand only requires two capacitors and a crystal in order to easily run the flashed program on any board, which is great because that saves space on the PCB design and allows for less of a clutter when it comes to the number of external components needed.

Table 9: Microcontroller Cost Comparison

Microcontroller	Cost of chip	Cost of development board
MSP430G2553	$2.80	$15.00
ATmega1284	$5.15	$30.00

3.2.6.4 Microcontroller Final Selection: ATmega1284

The sections that were previously discussed outline several different factors when it comes to choosing between two very popular and powerful microcontrollers. The ATmega1284 has won in every single hardware category except for cost, but because of that higher cost we can see the extra features that we are being provided with by choosing this as our microcontroller for this project. Simply put even though the package remains small, the ATmega1284 is the best bang for buck within all of the microcontrollers discussed in the all the previous sections. The hardware capabilities of such a robust MCU allows us to implement a unique design for the amount of communications and devices that must be interfaced to collect data to this controller and also be sent off to where we need it to go. Another very important aspect when it comes is the feasibility of programming and executing complex instruction on it, given the fact that the ATmega1284 is part of the

AVR family of microcontrollers it has full Arduino support and an entire community of DIY projects from novice to expert levels. This is beneficial because that means there are plenty of resources and tutorials on how to work with this microcontroller when interfacing multiple different devices and how to use it on large scale projects that require a lot of parameter sensing such as ours.

3.2.6.5 ATmega1284 Hardware Communication

In all types of microcontrollers there is certain communication protocols that are followed by the hardware in order to allow interfacing between sensors and other devices. Most modern-day microcontrollers support different types of communications in order for them to properly communicate with all the hardware and control registers and other internal devices. Given the fact that we have chosen the ATmega1284 as our microcontroller for this project the section will go into more detail the communication protocols available on this chip and example of what exactly they are capable of.

The first communication tool for use within the ATmega1284 is SPI or serial peripheral interface which is used as a variation of an interface bus such that the microcontroller knows how to send out data to various devices. The devices that mainly use SPI are sensors, which is why it is important that this board supports this as our project relied on a lot of sensors data sending and receiving between multiple areas. The way SPI works is by manipulating clock cycle and data lines that are chosen by the user in order to dictate to the microcontroller which device it should be talking to via select lines much like a multiplexer. The main benefit of using SPI is that it works in a synchronous manner, meaning that you can be sure that the multiple data lines that are being requested are operating at the same time which is great for the reliability of the transmission of data and such will not cause mismatches within programs. This also allows for the customizability of clock times and controlling with which edges of the clock that data transmits. Features like this are extremely important because certain devices that must be interfaced with the microcontroller have specific needs when they are being operated around a clock cycle. SPI also can allow for the receiving of data back from the devices by utilizing the master and slave technique, the microcontroller acts as the master for our application while any other devices and sensors are the slaves that send back data to the microcontroller. The way this is accomplished is by the master already having predetermined when the slave is going to send back data based on the functions defined within the programs and instructions. Below is an example graphic on how this master and slave communication is accomplished. One of the final methods of communications from hardware that the ATmega1284 is capable of, is analog to digital conversion which is transmitted at a max bit amount that is quite generous when it comes to the sensors available on the market that we used for this project. Although it is common that most projects do not end up using analog device sensors, it may still be a viable option for specific sensors that require higher power and more accuracy of data retrieved from the device. The way this is accomplished is by the master already having predetermined when the slave is going to send back data based on the functions defined within the programs and instructions. Below is an example graphic on how this master and slave communication is accomplished.

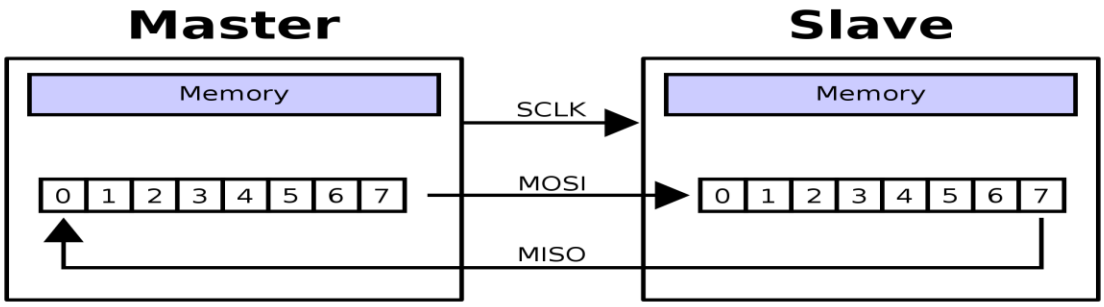


Figure 7: SPI Communication

On a project of this scale that requires multiple sensors and devices connected, this is a great tool to have because there can be multiple slaves all talking to one master. Fortunately, program can be written in such a way that not all slaves have to be talking at the same time and most modern devices or sensors have a sleep mode from which they can be woken from and begin to transmit data when needed. This process is better known within programming as using the select lines to dictate which device to begin talking to. Unfortunately, one disadvantage of SPI communication is that while it is reliable and synchronous, it can only handle transmitting data for short distances. Given the scale of our project this may not be such a large concern depending on the devices being strategically placed or using wireless transmission and then connecting from a receiver and then interfacing into the SPI. There is also a limited number of slaves than depend on the amount of select lines, if there are too many devices one may run out of space to continue interfacing their devices.

The second communication tool that is able to be used because of the ATmega1284 hardware is the I2C protocol. This is probably one of the most popular communication protocols when it comes to embedded systems. The main drawback of the previous mentioned SPI communication is that is requires a lot of hardware (wires) that can easily use up the amount of available I/O pins. I2C is quite an effective remedy to this problem because it operates in the middle of asynchronous transmission and SPI, this allows devices to only need two wires to communicate versus four or five for SPI. The I2C protocol utilizes SCL and SDA in order to control the clock and data respectively, of course the clock cycle and signaling will always be controlled by the master which in our case will always be the microcontroller. Overall, I2C allows for easier integration of devices without eating up all the hardware resources like SPI would, it even allows for devices to operation at differing voltage levels than the microcontroller by controlling internal pull up or pull-down resistors. Another great benefit of using I2C communication is that although only one SCL and one SDA pin location exists, that does not mean only one device can be interfaced. This is because of the Arduino libraries that are available for specific devices which each have their own address on the I2C line such that multiple devices can be used at the same time on the microcontroller. Shown below is a graphic of the I2C communication protocol in action with three slaves and their respective lines:

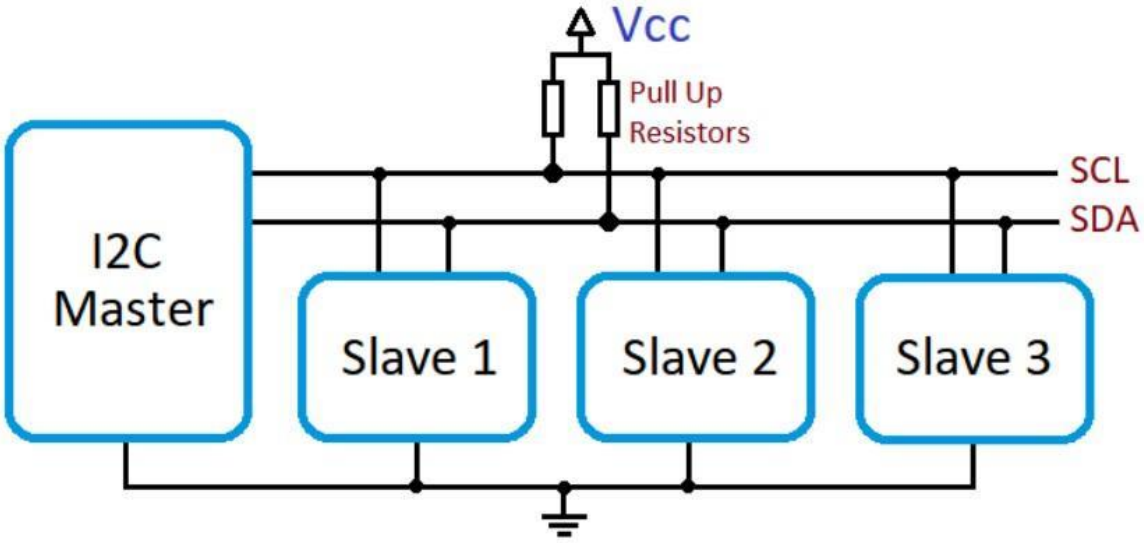


Figure 8: I2C Master and Slave

Given the scope of this project, it would be wise to choose sensors that are best suited and reliable to work with the I2C protocol, such that space can be saved on the I/O pins and the hardware will not be so cluttered. This is reflected in our final sensor choices utilizing I2C serial communication for easy data transmission.

The third communication protocol on-board the ATmega1284 and the most important one is the serial connection or better known as UART. Commonly on most modern microcontrollers there are two pins that support serial communication which are the receive pin generally denoted as RX and the transmit pin generally denoted as TX. The UART is an asynchronous device that can talk between data lines and those same RX and TX pins. The reason this UART connection is so vital is that this is the way our wireless communications modules are interfaced. The only way to interface a Bluetooth connection regardless of which module is purchased is through wire up the Bluetooth module itself onto the UART pins, the same goes for the GPS sensor that are a part of our project. There is a drawback that comes with using UART connections within these low-cost microcontrollers, and that is the fact they only have one TX pin and one RX pin meaning that for wireless communication only one active device may be connected to them at the same time, meaning that in standard configuration both a Bluetooth module and GPS cannot both be connected to these pins at the same time. Some high cost microcontrollers do come available with more than just one RX and TX pin, but those microcontrollers are overkill in both functionality, cost and hardware integration within the scope of this project. Fortunately, there is a single way ATmega1284 is capable of allowing more than one device interface onto the microcontroller such that both can be used, this method is by enabling two of the available I/O pins as virtual UARTs. By enabling two more pins that are virtual UART this would allow specific baud and bit switching procedures within these pins that allow another device to be interfaced. There is only one small drawback to enabling this feature, and that is the fact that the data transmitted is often slower and needs to be less complicated because it is not technically a true UART port.

Consequently, all of the communication protocols discussed in this section are excellent for use within our project. This only further illustrates the point that the ATmega1284 microcontroller is more than capable of accomplishing the tasks set out by this project and is quite a powerful microcontroller that can be interfaced with all the sensors and technology that are used. Device integration with the ATmega1284 must ideally be as fluid as possible and allow for smooth integration with the software side of the project as well such that data can be collected and transmitted reliably. Accordingly, the Atmega1284 was chosen as our Microcontroller Unit and the PCB was designed with the pins and logic in mind. All sensors and communication devices interface with this unit and necessary code for each would utilize the programmability of this chip.

3.3 Sensors

The following section will compare various types of sensors that are used for the project. Of these various types of sensors, one was chosen in order to efficiently develop the system. The desirable characteristics of an ideal sensor for this project are low power, high accuracy, and easy communication with the MCU. Sensors that obtain these traits are vital to the overall flow of the system since the sensors communicate the information needed by the driver of the boat. The goal of this section is to choose the desired type of sensors from the following list and make a selection while also choosing the preferred communication with the MCU.

3.3.1 Analog Sensors

Analog sensors create a continuous signal of output voltages, proportional to the quantity measured, that usually range from 0 to 5 volts. Analog signals are time-varying values which create a smooth continuous signal or reading. Although, this signal is subject to noise which can heavily damage the signal-to-noise ratio and thus create an unclear and false signal. This signal could be received by wiring to the MCU but could also be received through radio frequency waves. For the purpose of this project wires were used to transmit data to the MCU.

However, when using an analog sensor an analog to digital converter (ADC) would be required when connecting the sensor to our microcontroller. With regards to the microcontrollers being considered, each microcontroller contains analog pins which will internally create an ADC. An ADC is composed of multiple bits which converts to the amount of discrete analog levels it is able to detect. Smaller amounts of bits translate to a less precise signal while a larger number of bits translates to a more precise output signal. Since the ATmega1284 has been chosen, it is important to take into consideration that the ADC on this MCU has 10-bit ADC pins which means it will be reading 1024 values. Although this ADC bit size is good enough for most small projects, it is vital to ensure that the readings from each sensor is as accurate as possible with high resolution since these sensors are giving information to the operator of the vessel.

A disadvantage with analog sensors with relation to ADC is that noise from the digital supply will affect the ADC and in turn, will create an output signal that reads incorrectly and inaccurately. It is possible to work around this issue and come up with a solution to

lower the noise, but it will cost more. The noise would be avoided by having an analog voltage reference instead of using the digital supply. For this method to work fully, it is important to have the voltage supply separate and away from the digital circuit with its own wiring. Other factors to consider when depending on the ADC to convert the analog signal into digital, is the output range of the sensor compared to the input range of the ADC as well as the MCU's speed and throughput. Since the most common output range for sensors is 0V-5V and the input range of the ATmega184 is 5V, this will allow the sensor to read a more accurate signal. The ATmega's clock speed of 20 MHz is average given the type of package, though this might affect the accuracy of the sensor compared to faster MCUs mentioned. Overall, analog sensors would require more cost in order to create a more trustworthy reading and would also require extra circuitry which was deemed unnecessary upon final design of this project.

3.3.2 Digital Sensors

Digital signals are a representation of data in discrete values, namely ones and zeros which means the signal is either on or off. Compared to analog sensors, digital sensors create square waveforms while analog singles create sine waveforms. Digital signals are not nearly as susceptible to interference which is why digital signals have replaced many analog-based technologies. Compared to analog, digital is more reliable across distances which can limit the use of analog in certain environments such that the analog signal could not accurately read input from a particular distance. With sensors that are required to read in data from one area of the boat such as temperature or depth, long wires would be required in order to transmit the data back to the MCU which would require the reliability of digital signals over a longer distance. Since digital signals are transmitted in the form of zeros and ones, this allows the readings from the signals to be copied at any time, unlike analog signals which worsen over time. This feature of digital signals is desirable when recording data for later use which occurred with the mobile application for this project as it was pushed to the user's smartphone.

Digital sensors also digitally convert and transmit data in some form of zeros and ones. Digital sensors are often referred to as the modern successors of analog sensors since they overcome the disadvantages of analog sensors. The components used in digital sensors are the sensor, cables, and transmitter. To a certain threshold, digital sensors have a higher tolerance to noise compared to the low tolerance of analog sensors and in return produce a more accurate reading. As stated above, a digital signal is unaffected by the length of the distance from the signal being read and the signal being processed. This allows for sensors to be placed farther away from the MCU, connected by a long wire, while also giving an accurate reading. In relation to this project, this feature of the digital sensor is an important advantage to obtain due to the fact that the water temperature sensor required six feet of routing to be mounted on the transom of the boat for accurate readings to be achieved.

For reference, temperature sensors were placed both in an area where the sensor is in in contact with the water and in an area for an accurate reading of the ambient air temperature. This required the sensor circuitry to be connected to the MCU or PCB by an extensive wire in the case of the water temperature sensor, and directly connected via header pin in the case of the air temperature sensor. Due the issue of the water temperature sensor

routing, the distance of the sensor from the MCU was taken into consideration when choosing the final sensor to be implemented for this purpose.

3.3.3 Ultrasonic Sensors

Ultrasonic sensors contain transmitters, receivers, and transceivers which in total converts an electrical signal into ultrasound, ultrasound into electrical signals, and both. Ultrasonic sensors detect distances from objects by transmitting a sound wave signal and receiving another signal back once the target object is detected. The ultrasonic sound wave is transmitted by using a transducer which will send a pulse towards the target object and in return also receive the reflected waves back. The pulse emitted towards the object is a short, high-frequency sound pulse. Once the sensor receives the signal reflected back, the sensor itself will calculate the distanced by tracking how long it took to reach the object. By measuring the time, it took to reach the target object, an ultrasonic sensor therefore is able to suppress background interference since it is not taking into consideration the force of the sound waves returned. These sensors are independent of color, light, temperature, fog, and other environmental noises which also allow the readings to be more accurate especially when under the conditions while on the vessel.

The three different types of ultrasonic sensors include object detection, distance measuring, and sensors that perform both. Object detection transmits a discrete output while distance measuring transmits an analog output. Ultrasonic sensors are the most commonly used for object detection due to the accuracy of the ultrasonic transducer which could be helpful and taken into consideration when choosing sensors such as the depth and bilge sensors. Although, air temperature has the largest influence on the accuracy of ultrasonic sensors which could harm our goal considering the sensors are exposed to unpredicted weather. Since the distance measured from the object after the object has been detected is calculated using the speed of sound, this causes inaccuracy since the speed of sound changes by 0.17% per degree Kelvin. However, other environmental influences such as humidity, air pressure, and air currents have little to no effect on ultrasonic sensors so in return, this type of sensor is generally desirable for a sensor to be used outside. This characteristic is important to this project considering the vessel is exposed to all of these environmental changes.

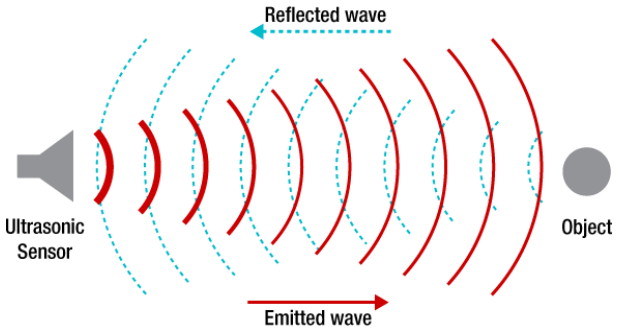


Figure 9: Ultrasonic Sensor

Ultrasonic sensors were considered to be used in this project for the depth and bilge sensors. Although, since lower cost ultrasonic sensors only detect up to about 4 meters away, it would be inefficient for deeper waters. To obtain a larger range a costlier sensor, compared to digital and analog, would be needed. The usual power supply of ultrasonic

sensors is 3-5.5V. Also, on the sensor is a ground, trigger, and echo pins along with the transducers. Overall, the ultrasonic sensor appears to be the main option for depth and proximity sensors but also require a higher monetary value which was deemed non-advantageous to our final low cost design.

Table 10: Comparison table of sensors

	Analog	**Digital**	**Ultrasonic**
Connection to MCU	Required analog to digital converter	Directly	Directly/ADC
Noise Level	Low	Low	High
Cost	Low	Low	Medium

3.3.4 Interfacing with MCU

With the MCU chosen, it is important to understand the different communications each type of sensor has with the MCU. The ATmega1284, as mentioned above, acquires three communications: SPI, I2C, and UART. It is vital to understand the different communications the board has available since different sensors require specific communication. An analog sensor will normally require two to three wires while a digital sensor requires an SPI, I2C, or a one-wire.

Analog sensors need to communicate their signal using an Analog-to-Digital Converter (ADC) since the microcontroller only computes digital information. Although it is recommended to use an external ADC instead of an internal one with MCU's, modern day internal ADC's are beginning to exert the same functionality. With regards to the chosen MCU, an ATmega 1284, this microcontroller contains a built-in ADC which the sensors will communicate with. The ADC ultimately produces a digital signal that is a percentage of the analog signal which allows the MCU to communicate with the sensor without external circuitry. These ADC's on the MCU are important to understand as it will make the analog sensor more or less accurate when transmitting the data. The converter simply requires an input of an analog sensor while the embedded converter translates the signal into zero's and one's. If the ADC is a lower bit converter then the reading of the analog sensor will be less accurate, while a higher bit converter will create a more accurate reading. For the ATmega1248 used during this project, a 10-bit ADC is being used. Although the ADC is less accurate than a higher bit ADC, this converter would still suffice for the purpose of this project. It is also important to keep in mind that the longer the distance of the wires from the MCU, the more likely noise is capable of interfering with the signal read by the sensor. This is a clear disadvantage for the analog sensors since important readouts such as water temperature require a considerable distance for routing from the microcontroller and PCB.

3.3.4.1 I2C Communication

Digital sensor communication allows for three different types of communication. The first, highly considered, form of communication is I2C. I2C is a serial communication where the data read in through the sensor is transferred bit-by-bit. This communication has power and ground pins but also obtains two other pins. One pin is the Serial Data Line (SDA) which is the main line for the master and the slave to transmit and receive data that the sensor has recorded. The last pin is the Serial Clock Line (SCL) which is the clock signal created by the master of the bus. Although there are instances where this clock signal could be controlled by the slaves, which would occur when the master is sending an abundant amount of data or if the frequency needs to be lowered for more processing time. With regards to sensor readings, the I2C is beneficial for the instance where a sensor records more than one point of data. This allows for the sensor to be connected to the MCU using only two wires which is ideal when trying to minimize the amount of wires. It is also possible with the I2C to connect multiple sensors with I2C protocol using the same data communication pins. The requirement to do so would be to connect the two sensors using the 7-bit address from the data sheet. This is a viable option especially with the wide arrange of sensors and other components utilizing this method of communication when communicating with our microcontroller unit.

3.3.4.2 SPI Communication

The next form of communication for digital sensors is the Serial Peripheral Interface (SPI). This form of communication is known to be faster than the I2C. The pins for SPI are the power and ground but the SPI use four pins to communicate the data. The four pins for data communication are the serial clock, Master Output Slave Input (MOSI), Master Input Slave Output (MISO), and Slave Select (SS). The Slave Select pin is the digital output pin while the MOSI and MISO are specified pins on the microcontroller. Although SPI is faster than I2C, SPI clearly requires more wires than I2C which can end up cluttering the MCU. This draw-back, however, is negligible considering that SPI speed is significantly higher due to the full-duplex communication. When considering communication with the sensors, it is important to understand the power of each protocol. Since one of the main overall goals of this project is to keep a low-powered boat, it is notable that the I2C draws more power than the SPI.

The advantages of higher speed and lower power are key characteristics, but other disadvantages compared to the I2C are the deciding factor when choosing with protocol to communicate with the sensors. Although I2C is more complex to set up, the cost of implantation for I2C is cheaper compared to the SPI implementation.

3.3.4.3 1-Wire

Primarily regarding sensor communication with the MCU, 1-Wire is a method of data and signal transmission through a single conductor (Typically a wire). Using a data bus system as means of transmitting the appropriate data, these devices will typically be tied to a respective voltage source, ground, and a wire or data line connected to the data pin of our Microcontroller unit. In this sense, it is similar to I2C with regards to methodology for

sending data. However, there are key differences and certain applications which may benefit from using this form of data transmission. Primarily devices that are not reliant on high-speed data transfer and not in need of large amounts of data to be transferred, the 1-Wire interface is a lower power option with longer range. The sensor for water temperature for example would benefit from using this method as the data and information required to be transmitted is comparatively low.

It is of key importance to recognize that while this method of data transmission may be preferable for some systems and devices, it is not applicable to all devices. Many devices, especially those which are NMEA compliant will require the use of several data lines to transfer data. In this case the 1-Wire bus system would be rendered ineffective and as such the availability of components would reflect this. Overall, sensors utilizing lower data transmission rates would benefit from this and this is the ideal choice with regards to low power options for components requiring data transmission.

Table 11: Comparison table of MCU communication

	I2C	**SPI**	**1-Wire**
Number of data communication wires	Two	Four	One
Clock features	Clock Stretching	No clock stretching	Does not use clock signal
Power Consumption	High	Low	Low
Cost to Implement	Low	High	Low

3.4 Electrical

Of key importance to this document is the inclusion of the electronics which was utilized aboard this system as well as explanations for their role in the integration between the microcontroller unit and potential software to be included. Comparisons will be drawn to determine vital components and specifications provided for parts which are to be provided or utilized which will aid in the overall design and future analysis. The primary components of the electrical system as a whole are discussed below, and they must meet the standards set forth by the coast guard, IEEE, and other organizations to be discussed in the standards portion of this document.

3.4.1 Batteries

Due to the nature of sponsorship with this project, battery choice was limited to those provided by 'Torqeedo' which are compatible with the electric motor chosen to power the silent aluminum boat. However, the specifications and limitations of these batteries are of great importance to this document as these held a decisive aspect regarding the component choices outlined beyond. For convenience, a figure detailing the appearance of the batteries to be used is included to the right. It is of key importance to note the housing was large enough to hold two of these along with the corresponding charging hardware. The housing for the batteries also met the coast guard standards which are outlined in the standards portion of the document. The large power of these devices was considered with regards to housing so as to ensure that the material chosen was capable of withstanding large amounts of heat or the harsh marine environment. In Table 12 below, specifications are given for this battery which were considered when designing the appropriate step down and voltage convertors. These specifications are per Torqeedo directly and reflect accurate ratings which we are designing our system to utilize.

Figure 10: Torqeedo Battery

Table 12: Torqeedo Battery Specifications

Specification	Rating
Voltage	48V
Electrical Power	5 kW
Dimensions	506mm x 386mm
Charge Time	10 Hours
Full Load Operation Time	1 Hour
Waterproofing	IP67

3.4.2 Electric Motor

Figure 11:Torqeedo 10.0RL

As with the battery choice, our sponsorship and partnership with Torqeedo was afforded an opportunity to use their proprietary electric motor for propulsion of the boat. The specifications of the motor being used for this project are listed in the table as the power consumption was monitored and balanced with the other electrical components in use. It is a single prop outboard motor with adjustable trim and is controlled via a steering wheel and throttle. A tiller option was available for the motor, however to provide the most user-friendly product the center console design with a hydraulic powered steering wheel was chosen. The electric nature of propulsion also offered a greater degree of noise reduction than a typical gas-powered outboard motor which falls in line with the expectations of the sponsor. It is worth noting that telemetry and various readouts of the motors can be obtained through a CAN bus or Bluetooth, however as this information is proprietary, we were unable to incorporate them into our design directly. Bluetooth connectivity with the motor is possible via the smartphone however and can be integrated into software designed in the future if deemed necessary. However, sensor integration and the availability of off the shelf marine electronics allowed for work arounds to suit the design needs of this project. Particularly if a different electric motor or even a gas-powered motor is needed to replace the expected Torqeedo motor, our electronics will still allow the necessary readouts from sensor data to be displayed. Nonetheless, Table 13 below provides the specifications for various aspects of the Torqeedo 10.0RL motor which is to be utilized. The design and physical appearance can be seen in Figure 11 to the left which enables us and the mechanical team to design accordingly to suit this specific motor.

Table 13: Torqeedo 10.0RL Motor Specifications

Specification	Rating
Horsepower/Propulsive Power	~20 HP
Electrical Power	10 kW
Voltage	48 V
Waterproofing	IP67
Weight	58.9 kg

3.4.3 DC-DC Converters

Using the batteries provided by one of our sponsors Torqeedo, power distribution and management is a key factor with regards to component choice and design. DC-DC convertors are a necessity for this project as the 48V batteries used would need to have the voltage stepped down before delivery to the microcontroller unit, sensors, and other electrical components. There is a plethora of options regarding this area, with those of key interest being discussed in detail.

Switching Regulators pose a viable option with regards to power management and distribution of the power from the Torqeedo batteries. Switching regulators provide a higher efficiency than many linear regulator options which are often hampered by comparatively poor performance. With the goal of maintaining above 70% efficiency and best utilizing the batteries, switching regulators offer a wide range of choices to suit the design needs of this project. Ideal candidates being buck, and buck-boost converters which will be discussed in the following paragraphs. A general comparison of efficiency levels of a switching regulator vs. linear regulators is seen in the following Figure 12. Note that the switching converters/regulators offer a substantially higher level of efficiency.

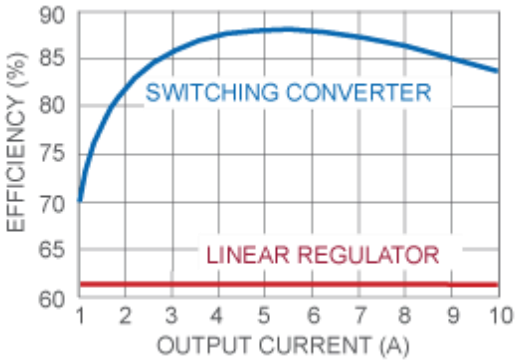


Figure 12: Switching vs. Linear Regulator

3.4.3.1 Switching Regulators to MCU

With the MCU operating voltage expected to be between 3.3V and 6V, regulator choice is crucial to step the voltage from the batteries down to an acceptable level. Comparisons will be drawn based on the provided specifications for each of these regulators, with efficiency, cost, and the desired regulated voltage playing a key role in determining our final choice. The comparisons are illustrated in Table 14 below. With the comparisons above, the LM2576HV and the TPS54560 modules lend more to the idea of cost-effective design for this project while also providing the necessary amenities to regulate our desired voltage for distribution to the Microcontroller and other components. Both boast nearly identical cost, duty cycle, and operating temperatures. However, the lower switching frequency of the LM2576HV conceivably offers lower power loss than the considerably higher switching frequency of the TPS54560. In a system desired to be as power efficiency as possible, this lead us to the choice of the LM2576 to ensure the longevity and safety of

the system. Maximum current ratings are similar, however the 3A input of the LM2576 was enough for our purposes in powering generally low powered sensors and electronics.

Table 14: Switching Regulators Comparison

Parameter	LM2576HV	TPS54560	LT1956
Vin (min)	4 V	4.5 V	5.5 V
Vin (max)	60 V	60 V	60 V
Vout (min)	3.3 V	0.8 V	0.6 V
Vout (max)	37 V	58.8 V	
Iout (max)	3 A	5 A	1.5A
Switching Frequency (min)	42 kHz	100 kHz	460 kHz
Switching Frequency (max)	63 kHz	2500 kHz	540 kHz
Operating Temperature	-40 to 125 C	-40 to 125 C	-40 to 125 C
Cost	$6.09	$6.07	$10.52

3.4.3.4 Cllena Buck Transformer

To meet specific coast guard standards and ensure the functionality of the systems set forth by our sponsors, a separate step-down device was used for power distribution to the lighting and bilge system. As the bilge pump requires a higher voltage than what is needed by the MCU, (12V in this case), a separate device from one of the above listed switching regulators was utilized. Of primary interest was the Cllena DC 48V Buck Transformer for this purpose. It is worth noting that as the lighting systems and Bilge are desired to be wired to physical switches for safety purposes, they are capable of being separate from the MCU powered system.

Looking at the Cllena Buck Transformer, the specifications meet our needs and do so with a level of stability which is needed for the voltage sensitive components. Conversion efficiency is given as being between 95-97% which further lends to the stability of the device and ensures longevity for this component which will play a crucial role in ensuring safety standards are met. In addition, with a rated input voltage range of 30 - 56V DC, the Cllena transformer falls in line with the expected input from the batteries to be used. Outputting a maximum of 30A at 12V, the device also ensured that current issues did not occur as the devices which were integrated are of low power consumption.

3.4.4 Lighting

With the marine applications of this project come the standards associated with safely operating a vessel in state or federal waters. These standards entail a procedure with regards to lighting and requirements which should be met to be compliant with the coast guards' regulations. For our project, a boat under 12 meters in length, a lighting system is entailed as including the following lights with their respective visibility; Masthead light (2 Miles), Sidelight (1 Mile), Stern light (2 Miles), Towing Light (2 Miles), White/Red/Green/Yellow All-round light (2 Miles), Special flashing light (2 Miles). LED's were the primary choice with regards to fulfilling these requirements as their general power efficiency and low cost are ideal for our system. As these lights are required for safe operation of the boat in low light conditions, it was of interest to route these lights to physical switches with capability of MCU interfacing. Should the user's smartphone fail, the physical switches still allow the LED's to be turned on and off to ensure the safety of those on the water as well as meet coast guard standards.

3.4.5 Printed Circuit Board (PCB) Technology

The following sections discuss the different PCB platforms that were researched for use when creating designs. Once the discussion concludes, a table comparing each platform was used to decide which tool provided us the most benefits.

3.4.5.1 EasyEDA

This free web-based Electronic Design Automation tool allows engineers to design, simulate, and share schematics, simulations and printed circuit boards. EasyEDA was created by Dillon He and Eric Cui in an attempted to provide a free, independent and easy to learn platform for creating designs. The platform launched in 2014 and has since become a popular demand for designing and testing PCB designs. Since this tool is web-based and free, no downloading is necessary, but it is recommended to use Google Chrome or Firefox when using the online tool. Since no group members on our team have had experience with creating and designing PCBs, this tool is appealing to our team for being a simple and user-friendly tool. Another important note about this platform is that it has a plentiful amount of open source projects which will allow our team to be able to see other examples of how more experienced designers have created their PCB.

To begin with creating a design, an account using a google email will need to be created in order to begin a project. Once the account has been made, the team will have to conspire a plan to create the design. Luckily, EasyEDA provides flow charts which will allow us to visually see the plot of the idea before we begin to implement the design. From the previous courses taken before enrolling in Senior Design, every team member has had experience with creating schematics and are more than capable of understanding the interface when designing or drawing the circuit schematic. Not only have we had experience with the schematic portion of EasyEDA, but we have also had experience regarding simulating the schematic. This is done on EasyEDA by using the "Run the Document" jogger.

In order to have a PCB design of the schematic that was just created and simulated, EasyEDA has a feature that converts the project to a PCB. This conversion feature will return an outline of the PCB board with the components connected by direct wires. EasyEDA allows for creativity when placing these components on the board, although it is important to understand which placement will be most beneficial. Once the circuit is placed on the board, EasyEDA provides the option of auto routing which requires some input of size of the tracks or if certain areas should be avoided. Obviously, this auto routing feature should not be completely trusted. Once the tracks have been adjusted to the most optimal position, the PCB design will be ready to order.

3.4.5.2 EagleCAD

Easily Applicable Graphical Layout Editor is a common and useful CAD software used for PCB designing. EagleCAD provides a library with a large number of electrical and electronic components. The team will mainly use EagleCAD for the components of our PCB. EagleCAD is capable of being imported into EasyEDA but version 6 or later must be used as this is the only versions compatible with EasyEDA. To begin using EagleCAD, it is first required to download the application for free. Once the software is downloaded, a project can easily be created. With the free version of EagleCAD, the board created can be no larger than 100mmX80mm unless the EagleCAD Standard is bought in which the maximum size of the board will increase. Similar to EasyEDA, a schematic must first be designed to be able to move forward to a PCB design. Once the schematic is made with all supply connectors, I/O pads and test points, it is important to enter component values to be displayed onto the schematic and parts list.

Unlike EasyEDA, EagleCAD does not have a built-in simulation feature. Simulation features could be important from other applications or tools, which would require integration with other platforms. This is a disadvantage since it is important to understand and verify that the PCB is working properly and as expected. Although, converting to a PCB board from a schematic is also as simple as EasyEDA. An outline of the PCB is created with the components waiting to be placed on the board next to it. Finally, the last step requires the components to be placed on the board and auto-routed.

3.4.5.3 KiCAD

KiCAD is a tool used to design PCB boards which means it first requires a download in order to use the application. The process of creating a PCB in KiCAD is similar to that in EagleCAD. To begin, a schematic must be made and the component values to be inserted and declared into the circuit. Once again, KiCAD does not have the feature of a built-in and easy to use simulator within the application itself. After the schematic is finished, there is a converter that will, again, create a blank canvas for the PCB board with the connected components waiting to be placed onto the board. Now the PCB board is ready to be routed using the built-in tool. As with other tools for PCB design, it is important to verify and check the routing after auto-routing.

Although, with the free tool, an unlimited amount of designs is permitted. Unfortunately, this application tool seems to fall short of the necessities of this project. The extra tools

needed to have a complete process is a disadvantage to KiCAD when compared to the accessibility and offerings of the other design tools.

3.4.5.4 DipTrace

DipTrace is a software tool to create PCB design which comes in multiple different editions that offer a varying number of pins and signal layers. DipTrace does not offer a free tool but does offer a perpetual license. This tool allows for a schematic capture, PCB layout, and library creation. Although, it is important to point out that this software tool is not capable of creating a simulation response form the schematic capture. This tool does convert a schematic into a PCB design that also makes a blank PCB board with components next to it. Once the components are place onto the board, it can then be auto-routed. Again, it is important to re-route and verify the routing before sending the PCB to be designed. It is important to point out the key features of DipTrace which includes differential parts, 3D previews and STEP export. For easy comparison, Table 15 on the following page outlines the inherent disadvantages when utilizing each PCB design tool as well as the associated cost. From these disadvantages and comparative prices a final conclusion would be drawn for which tool or tools suits our design needs best.

Table 15: PCB Software comparison table

PCB Tool	Disadvantages	Price to use
EasyEDA	Requires internet connection to create design	Free
EagleCAD Student	No simulation environment	Free
EagleCAD Standard	No simulation environment	$100.00/Yr
KiCAD	No reliable auto-router	Free
DipTrace Lite	Limited number of pins without software upgrades	$145

3.4.5.5 PCB Tool Selection

From the above description and comparison table, a final selection of multiple tools was decided upon. With a very tight budget, it is important for our group to make budget cuts where budget cuts can be made. The high price of KiCAD when compared to the other available tools in this regard, as there are other tools for free which accomplish similar tasks. Since EasyEDA is a free online tool and very easy to use for first-time PCB designers, this tool was our main source for creating and designing PCBs. Although, after extensive research, it has been found that integrating other platforms such as EagleCAD

could be beneficial towards the PCB design. EagleCAD has proven to be useful with regards to the design and creation of components. When these two platforms were combined to make the overall product, a successful PCB design was achieved which ensured the functionality and proper integration of necessary components.

3.5 Communication

Wireless communication and interfacing are of chief interest regarding this project as GPS and cellular applications are both requirements desired by our sponsor. This section will analyze the key factors analyzed when choosing Bluetooth, 4G/LTE, and GPS modules as well as discuss the capabilities and limitations associated with each. It is crucial for the wireless components chosen to enable proper interfacing via a smartphone to control desired sensors and provide accurate readouts of the electrical systems on board the system. Transmitting and receiving data from these sensors provides a unique problem which was solved with our design as various aspects limited our component choice. This was taken into account moving forward and it posed even greater problems during the fabrication phase as well as integration phase with the mechanical engineer and computer science teams also working on this project. The computer science team in particular had a hand with regards to communication methods chosen as they wrote the code to receive the data from our microcontroller.

3.5.1 Bluetooth Connectivity

Bluetooth module technology is built upon the concept of transmitting and receiving data over relatively short distances (Typically less than 40 feet) in what is known as a "Personal Area Network" or 'PAN'. The general standard for WPAN or Wireless Personal Area Networks is IEEE 802.15; This standard is discussed in detail in the design standards portion of this paper. Interfacing with a given MCU is available utilizing UART communication between the Bluetooth module and the desired unit. Often a limiting factor of Bluetooth is the range and multiple classifications are used for the various modules. In order of range; Class 1 is generally between 66-98 ft, Class 2 is between 16-33 ft, and Class 3 modules are 0-3 ft. Due to the nature of the systems containment within a planned boat design of 17 feet, Class 2 Bluetooth connectivity is a viable choice with regards to transmitting the data received from on-board sensors and allows for lower powered module choice as distance is not a limiting factor. With low power consumption a key factor in the Bluetooth modules, it also offers less stress on the MCU being used for interfacing and benefits the overall low power goals of this project. It is worth noting however that while the waterproof housing and aluminum design of this boat was expected to adversely influence the transmitting/receiving capability of the Bluetooth sensor and paired cellular device, it was determined to be a non issue. Nonetheless, the strength of signal offered by a given module was considered to provide the most secure connection and reliable data transfer. Frequency hopping spread spectrum was analyzed as useful as it would limit narrowband interference and permit the user a degree of security as the spread-spectrum signal is often difficult to intercept. However, security features in the app itself resolved the most notable issues of connectivity.

3.5.1.1 HC–05 Module

When analyzing potential candidates for Bluetooth modules, a determining factor was the modules ability to act as both master and slave for optimal interfacing and control between the smartphone app and MCU. Perhaps the most popular choice for the ATMEGA1284 MCU, (Our chosen MCU moving forward), is the HC-05 Bluetooth module. With a maximum operation range of 30 ft, it suits the design constraints of the boat and is capable of talking to our board and smartphone via serial connection provided the user remains on board. Utilizing the IEEE 802.15.1 standard and Frequency Hopping Spread Spectrum, this module promises a level of quality which is desired for our communication needs. Further lending compatibility is the modules ability to operate with 3.3V or 5V which is crucial as the MCU operates under identical loads. This module transmits and receives data via the Rx and Tx pins associated with our respective MCU choice which will influence design in the future as there are limited transmission pins. In total, the capabilities offered and compatibility with the AVR microcontroller to be used makes the HC-05 a potential option. However, as it operates under Bluetooth 2.0 rather than BLE there will be interfacing issues with newer smartphones such as iPhones. To this extent, the utilization of the HC- 05 module for communication in our system was severely hampered when compared to others available on the market.

Figure 13: HC-05 Module

3.5.1.2 TEL0084 BLE Micro

Moving on to BLE or Bluetooth 4.0 devices, the BLE Micro is a strong candidate as it offers connectivity to Apple devices such as the iPhone. Compatibility with the respective ATmega microcontroller unit to be used also lends to the desirability of this module. Operating at a supply voltage of 3.3V would enable this module to work with our microcontroller unit however a voltage divider or regulator would be required given the typical 5-6V output of the device. Again, operating using 2.4GHz as the typical Bluetooth 4.0 method for serial communication this device boasts a range which is compatible for the boat and system as a whole to ensure connectivity is little to no issue. IoT compatibility specifically on iOS devices is of key priority, and the upgraded firmware and Bluetooth generation as compared to the HC-05 module is a considerable and necessary upgrade. However, the drawback with this module is the noticeably higher cost of around ten dollars when compared to the relatively cheap HC and HM modules. If the efforts are to keep cost down for all electrical components, then a disparity of cost approaching five times greater than that of the other

Figure 14: TEL0084 BLE Micro

options is to be considered. The TEL0084 module can be seen in Figure 14 with a noticeable size and packaging difference than the HC and HM modules. The lack of pins and general surface mounting component of this device would also provide a more taxing design for the PCB and generally increase cost while providing similar benefits to the module discussed in the following section. Potential design of a shield or utilization of a shield available on the market would resolve this issue however.

3.5.1.3 HM–10 Module

Another Bluetooth module utilizing 4.0 Bluetooth low energy technology is the HM-10 Bluetooth module. Boasting an input tolerance of 2.0 to 3.7 volts with logic levels on 3.3V, this module is compatible with the constraints of the microcontroller being utilized. The onboard chip of this module is the TI CC2540 which boasts 256KB of flash memory available and 8KB of RAM which is suitable for our design. Low power consumption is the goal of our electrical system and this module is expected to do that handily with the given design. The packaging and overall layout of the HM-10 module are illustrated with Figure 15 to the right, Master and slave mode capabilities of the HM-10 module as well offered us the opportunity to control other sensors as desired as well as provide a more user-friendly experience. As with the previous module, the updated firmware and 4.0 Bluetooth low energy capabilities of this module made it an ideal candidate with regards to connecting to the IoT and interfacing with iOS devices which are of key interest to the computer science members of this project who seek to create an app using swift. When compared to the previous module as well, the design of this module with six available pins enabled easy soldering and the exemption of surface mounting. This in addition to the low cost of each module drove in the idea of a low-cost design which benefited our system as a whole and meet the sponsors requirements. In all, this was our prime candidate with regards to Bluetooth communication aboard the system and interfaces well with the ATmega MCU choice for sending and receiving serial communications.

Figure 15:HM-10 Module

3.5.2 BLE

Bluetooth Low Energy (BLE) is a specific type of Bluetooth communication released as part of Bluetooth 4.0. BLE is designed specifically for low power systems, so that products using it minimize power consumption when they are not streaming data constantly. BLE has enabled many systems to operate for months or even years at a time off of a battery the size of a quarter. The main thing that differentiates BLE from normal Bluetooth is that it stays in a sleep mode until a connection is initiated. This allows the device to keep power consumption extremely low while it is not communicating. BLE also connects in less than 10 ms, compared to the approximately 100 ms it takes for normal Bluetooth to connect. Similarly, to Bluetooth, it still operates in the 2.4 GHz band and utilizes frequency hopping to send its data. BLE does have some drawbacks however, mostly related to their data rate. The payload size of a BLE packet is limited to 29 bytes, which then limits the data rate to

a couple hundred kilobytes per second. Unlike regular Bluetooth communications, BLE devices use the Generic Attribute Profile (GATT) to define behavior after a connection is made. BLE supports both star and mesh network topologies. A star network is a one-to-many network design that can allow many nodes to report to a single central node, where the data can be aggregated and processed as desired. Mesh networks are a many-to-many network topography that allow all nodes to act as relay points for other nodes, so data can be sent on many different paths to its final destination. Mesh networks allow more flexibility with data routing but introduce another layer of complexity.

3.5.2.1 GAP

The Generic Access Profile (GAP) is a very important component of Bluetooth. The GAP used on a device will determine how it connects and advertises to the outside world. It also defines which devices a device can or cannot connect to. One of the most important jobs of the GAP is to define what role the device will fill. There are four roles that a node can fill. A node can either be a peripheral/slave, central/master, observer, or a broadcaster.

Peripheral devices are typically smaller devices that are storing the desired data, such as a sensor. Central devices are typically larger devices that will interface with users or other devices. A single central node can theoretically connect to an infinite number of peripheral nodes, but in reality, this will be limited by the capabilities of the hardware. The central node will send requests to the peripheral nodes to do some action, such as sending stored data or writing to a value. The peripheral nodes are then responsible for completing this action and sending any requested data back to the central node. Looking back at the example of a sensor being a peripheral node, a central node such as a smartphone could send a request to the sensor asking for a measurement from it. The sensor would then either read its last measurement or take a new one, and then transmit this data back to the smartphone. The smartphone would then display this data to the user to read or do whatever other calculations with it.

If a node is set to be a broadcaster or an observer, then the node will utilize a method that will allow them to communicate data without being explicitly connected. Before going into this process, we must understand how the Bluetooth advertising process works. When a device is not connected, it can do two things: send out advertisement packets or scan for a device. In the central/peripheral relationship, the peripheral typically would be sending advertisements packets and the central node would be scanning for them. If a node that is scanning finds an advertising node that it is interested in, it can then send a scan request, which requests additional information from the advertiser. This could be things such as the device's name, or it could be custom data. One advantage of Bluetooth is that it allows this scan response to be custom data, which can allow data to be transmitted from one node to another without ever connecting. This type of connection is an observer/broadcaster relationship. In this case, the previously central node is now an observer, and the previously peripheral node would be a broadcaster. The broadcaster can just keep advertising while including its scan response, allowing any observers to read its message. Since Bluetooth only allows a peripheral node to connect to one central node, using a node in the broadcast mode is the only way for multiple central nodes to read its information. For the observer or central node, there are two types of scanning, active and passive.

Active scanning allows the scanner node to send the scan requests while passive scanning does not. Passive scanning means that the scanner node will transmit no packets and will simply just read whatever advertisements it receives.

Once a central node has found a peripheral node that it wants to start communications with, it can then initiate the start of a connection. There are three different ways that two Bluetooth devices can start a relationship: connecting, pairing, and bonding. Connecting is the most basic and is simply just a link that allows two devices to communicate. This is the bare minimum for two devices to send data over Bluetooth. Pairing involves the exchange of encryption keys after the connection is started and allows secure connections where both devices are authenticated. Bonding is when the devices store the exchanged encryption keys after pairing, and this allows two devices to "remember" each other for future connections. Most connections we see today will always be bonded or paired, due to security purposes. On many modern mobile devices, bonding is done without even notifying the user.

The initial handshake at the start of a Bluetooth connection involves the negotiation of many key parameters to the connection. The first component is the Bluetooth address, which is just the key that the nodes will use to identify each other. A Bluetooth address can be public, static, or resolvable private. A public address is the hardware id programmed into the device and cannot be changed. A static Bluetooth address is one that is generated during the start of every power cycle. A resolvable private address is generated on demand, and requires an identity resolving key in order to decipher it. The next important step to the handshake is the key exchange and encryption. A short-term key will be created by the devices initially using public key encryption methods, in order to have a secure connection before exchanging more permanent keys. A long-term key will then be made and exchanged by the devices over the secure connection. The identity resolving key will also only be exchanged after the connection is made using the short-term key.

The last critical piece of the GAP is the connection parameters. The three parameters are connection interval, slave latency, and time-out length. Connection interval is extremely important since Bluetooth utilizes frequency hopping. This ensures that both nodes know when to jump to the next channel. Additionally, the connection interval is the key player into how much power you will consume. A higher connection interval means that the node has to come out of sleep mode less often and will conserve more power. The slave latency allows the peripheral node to skip connection events if it has no new information. The higher the slave latency is, the more connection events the peripheral will be allowed to skip. This also helps to conserve power, since the peripheral node can stay in sleep mode for a longer period of time. The timeout length is the time it takes for the connection to be terminated if no messages are received. This length must always be higher than the effective connection interval which is found using the following formula: Effective connection interval = (Connection Interval) * (1 + Slave Latency)

Therefore, if our connection interval is 100ms and our slave latency is three, the effective connection interval is four since the peripheral node would have to transmit after 400ms since it has skipped three connections.

Effective connection interval = (100ms) * (1 + 3)

Effective connection interval = 400ms

Since the effective connection interval is 400ms, the timeout length must be greater than 400ms so that the peripheral node has time to transmit in the situation where it doesn't get any new information for the time during all three connections.

3.5.2.2 GATT

GATT profiles are the definitions for how data will be transferred between two BLE devices. GATT profiles are only taken into effect after a connection is established between the two nodes. A GATT profile defines a specific usage of the connection, and one should be made for each intended use case. GATT profiles consist of two main parts: characteristics and services. When the central node sends a request to the peripheral node, the peripheral node will then determine which service is being called based on a 16-bit identification number called a UUID. Once the service is identified based on this UUID, it will check the GATT profile in order to determine what the intended behavior of the service is. For example, if the central node sends a request containing a call to the getData service, the peripheral node will then look at what related services this service may call upon, as well as what characteristics it may have to read or write to. In this case, we would find two characteristics in the service, data1 and data2, and we would use their UUID's to find them in the lookup table. It would then package these values up and send them back to the central node.

3.5.3 GPS

Location services aboard this project are of great interest to the sponsor, in particular GPS or GPS bread crumbing technology. GPS, or the Global Positioning System utilizes a satellite network to transmit radio signals to modules throughout the globe. At any given time, four or more satellites are available to transmit location data to a user's module. This location data gives the users distance relative to each respective satellite and can be used to triangulate the precise location of the module itself. The respective GPS signal sent by a satellite will carry three forms of information; The current position of the satellite, orbital information, and general time data otherwise known as the 'Ephemeris', 'Almanac data' containing orbital information which is generally less precise than the 'Ephemeris', and an amount of identification code used for identification of a unique satellite. Bread-crumbing utilizes this GPS system to send the user's locational information at regular intervals to be stored on a respective server which can be used to plot navigational data. For this project, this navigational data would be expected to be sent to a smartphone app to chart the user's travel information as well as provide a database to be referenced for future use of the boat. As with most sensors and electronic equipment transmitting or receiving data, this GPS module would be routed through the MCU to provide ample user control. It is worth noting that if we sought to reduce cost, the GPS modules effectiveness in an enclosed environment such as a dry box may be hampered. As such, it is likely that the module would have to be exposed to air to limit interference and allow satellite information to properly be received for bread crumbing. An issue with GPS modules

compatible with our MCU is NMEA 2000 compliance as many products are still in the process of integrating this capability.

3.5.3.1 Adafruit Ultimate GPS Breakout

GPS choice is critical to the design as a low powered module is desired while also providing reliable results to ensure user safety and accessibility to the necessary bread crumbing data. Of interest is the Adafruit Ultimate GPS breakout, boasting an AVR friendly implementation and ensuring communication between our microcontroller unit is not an issue. An input voltage of 5V will drive this module and is ideal as that matches the regulated voltage for our system which would eliminate the need for a voltage divider or other external regulators to power this device. With an accuracy of approximately three meters, accuracy shouldn't be an issue especially for the purposes of bread crumbing. Low power is another impressive feature of this GPS module, with a rated operating current of 20mA. Flash memory aboard this module will also enable the module to store up to 16 hours of data. This data would then be expected to be stored on the database being developed by the computer science members of this project which would allow a log of locations visited to be available to the user. To this extent, this module offers excellent features which satisfy requirements desired by our sponsors. A reasonable cost of approximately $40 is associated with this module, however it is worth noting that the implementation of this device may require a second microcontroller due to a lack of the needed transmit and receive pins on a single MCU. This would drive up the cost with regards to PCB design and potential wiring issues, however it is still feasible and in the long run may be what is required to produce a functional design. Despite this, there are noticeable specifications which made this a potential candidate with regards to GPS choice. A detailed list of the available specifications for this module are seen in Table 16 on the following page which allows for easy comparison to other GPS units. The key items of interest are the accuracy, cold start first fix, and general power consumption of the device.

Table 16: Adafruit Ultimate GPS Breakout Specifications

Feature	Specification
Input Voltage Range	3.3V - 5V
Current Draw	20 - 25 mA
Sensitivity	-165dBm
Maximum Velocity	515 m/s
Positional Accuracy	< 3 m
Cold Start First Fix	34s

3.5.3.2 NEO–6 Series GPS

Another viable option for GPS tracking in this system was the NEO-6 series GPS module which has integration potential with our chosen microcontroller unit. When compared to the Adafruit module, there are similarities, however several key differences lend to a stronger argument to be made for choosing this module. With the ability to communicate effectively with our given Microcontroller and a comparatively much lower price of approximately $5, this option provided ideal capabilities for our system. Surface mounting of this unit proved difficult with regards to available board space, however there were multiple available shields which allowed for easy mounting of this module. An image of an option regarding surface mounting choice is seen in Figure 16 to the right. The module utilizes various interfaces; SPI, UART, USB, and DDC, all of which are available for methods of communication. Complete libraries for the NEOgps series are widely available as well, especially when compared to other modules which would allow us to provide more accurate information and a much more stable design with regards to interfacing with our chosen MCU. The accessibility to these libraries and potential for proper embedded integration further outline why this GPS is the prime candidate for design choice. In the best interest of design however, operating parameters of this were analyzed to ensure choosing this device is of sound mind. The parameters for which this GPS will function at can be seen in Table 17 on the following page; Note that these are meant to be compared to the Adafruit GPS breakout. As such, the NEO-6 series GPS provided the most viable option with regards to cost effectiveness and available specifications to meet our needs.

Figure 16:NEO-6 Module

Table 17: NEO-6 Series GPS Specifications

Feature	Specification
Input Voltage Range	2.7V - 3.6V
Current Draw	10 mA
Sensitivity	-162 dBm
Maximum Velocity	500 m/s
Positional Accuracy	2.5 m
Cold Start First Fix	26s

3.5.4 LTE/4G

Potentially another option regarding Wireless Communications on our project is the integration of a 4G or LTE module. Mobile broadband offers a sweeping range of benefits with regards to integration into the Internet of Things (IoT) and is quite adept at providing an adjustable user experience with phone applications. Utilizing a variety of lower frequency bands, from 700 MHz to 2600 MHz, mobile 4G limits interference and is capable of transmitting data through multiple paths and utilizing a multiplexing design. As with most wireless communication, a method of antenna system or design is associated with 4G, however the antenna is usually packaged with the chosen module. With many 4G modules comes the addition of GPS or GNSS integration with the chip itself which may be used to satisfy the desired bread crumbing aspect of this project. This factor in addition to the replacement of a Bluetooth module in favor of 4G/LTE connectivity does lend viability to this option as a whole. As an IoT approach is also being taken with regards to a smartphone application being utilized for MCU control or data reception/transmission, having a 4G connection would expectedly increase the quality of the user experience. Of course, with these benefits comes the caveat of cost as being the main prohibitor for design. 4G modules which can effectively be integrated are typically costlier than a separate GPS or Bluetooth module and a typical monthly or annual fee for the mobile SIM card does impose some restrictions. User experience and interfacing capabilities with a smartphone however may offset the initial and continued costs if desired.

3.5.4.1 SIM7100A

With our sponsors desiring a method for 4G/LTE communication, options for accomplishing this with our chosen MCU are limited. There are few shields with a 4G chip available, with many of them being prohibitively expensive to the point of cost exceeding the value of functionality offered. However, there is an available option which suits the needs desired and is of a reasonable cost when compared to other modules. The SIM7100A module provides us this option, at a cost of approximately $85 for the module with the appropriate shield. It is worth noting that should integration of this specific GPS module be pursued, the inclusion of GNSS on this particular module would eliminate the need for a separate GPS module. With an input voltage range of 3.4 to 4.2V it is within the available range for connection to our MCU and PCB, and with relatively small dimensions on the order of 30mm it would not take up considerable space. Regarding data transfer, this module offers LTE with download speeds of up to 100Mbps and upload speeds of 50Mbps which would allow for quick logging of all available data sent through this device. USB driver's firmware updates would ensure reliable software is available for further interfacing with our device. Smartphone integration is another key factor which would potentially benefit from the use of the SIM7100A chip as well. With a 4G module on the boat, the user's smartphone would not necessarily require a connection as it could connect via the module itself and potentially limit consumption of the user's data. Talking capabilities however are limited on this chip and it would primarily be used to transfer data from the sensors and GNSS modules to the respective storage developed by the CS team.

3.5.5 Wifi

Wifi is an extremely common wireless ethernet protocol designed for use in wireless local area networks (WLAN). Wifi is the common name for IEEE 802.11. Wifi has had many versions in its history, and the current common version is IEEE 802.11n. Wifi leverages primarily the 2.4GHz frequency band. Newer versions of Wifi have also added the ability to use the 5 GHz band. This new 5 GHz band allows a higher throughput due to the less congested frequency band. This comes at the cost of less penetration through obstacles, which means a typically smaller range. The 2.4 GHz band has 14 channels available, but not all of them are available without a license. Due to Wifi's use of the 2.4 GHz band, there is a lot of other devices that interfere with it.

The IEEE 802.11 standard specifies the design of its datagrams. Each one is called a frame and is made of a MAC header, a payload, and a frame check sequence. The MAC header includes all sorts of information that is needed for the data to be sent. Some of the important information is the version of the protocol, an identifier for the frame, power info, and information about the ordering of the frames. The FCS is also known as the Cyclic Redundancy Check (CRC) and is used to check for integrity. Wifi also allows for management frames, which are frames made for overhead purposes such as connecting, disconnecting, and authentication. Control frames are used to control the exchange of data frames. The most common control frames are Acknowledgement frames (ACK), request to send frames (RTS), and clear to send frames (CTS).

Wifi also has many features that help refine it and provide a better quality of service (QOS) for its users. Through improvements of the MAC layer, there are two different control schemes available. The first is called the Distributed Coordination Function (DCF) and it uses the Enhanced Distributed Channel Access (EDCA). EDCA allows traffic to be prioritized but is still contention less and can't guarantee a higher QOS. This allows transmitters with a lower priority to occasionally trump one with higher precedence. When a high priority message is received, the contention window shrinks to increase the probability that the message will get access to the channel. Since this is still probability based, there is no guarantee a lower priority message won't trump the higher priority one. The second control scheme is called the Point Coordination Function (PCF) and it utilizes the HCF Controlled Channel Access (HCCA). The HCCA utilizes a polling mechanism to provide a guaranteed QOS. This allows a transmitter to use a channel for a given period of time before being pulled off of it. The central node always has precedence in the channel, and will poll all transmitters to see which are ready to transmit. It then will pick the one with the highest priority to transmit.

Wifi utilizes a security protocol called WPA2 in order to secure communications. The current version of this protocol utilizes an AES based encryption mode called CCMP. This protocol is required on all modern Wifi devices and is updated regularly to ensure security from new found breaches. Newer versions of IEEE 802.11 include higher bandwidth than older versions, as well as better power saving modes. New antenna technology has also helped aid in better performance in the newer radios.

3.5.5.1 Adafruit 2471

The Adafruit 2471 presented a viable option regarding component choice for a Wifi shield which utilizes the IEEE 802.11 standard to ensure higher bandwidth and a low power option. This shield utilizes the ESP8266 module carrying an antenna onboard and available pins for proper data transmission. The ESP8266 module itself draws approximately 250mA under full load at a voltage output of 3.3V which is considerable when compared to other components to be utilized in this design. This hurtle is overcome however by the low power options available for other components ensuring the functionality of this device. As expected, this module is compatible with the ATmega microcontroller unit and the chip itself offers FCC certification and a more than adequate amount of storage. With 64 KB of instruction RAM, 96 KB of data RAM and 4 MB of 32 Mb flash memory, there is a considerable amount of room for the necessary data to be stored. This chip allows for one analog input with a 1V max, 2 UART pins, and two 3-6V power inputs. The module can also utilize I2C or SPI for communication and has 2 available UART pins for connection to the microcontroller.

On the PCB of this component includes a reset button as well as a user button that can also allow the chip to operate in boot loading mode. The PCB also offered two diode-protected power inputs which allows for one USB cable and another cable for a battery. This would be useful when connecting to the MCU should future revisions utilize Wifi. Table 18 below includes the outlined specifications for this breakout.

Table 18: Adafruit 2471 Specifications

Parameters	Specification
Protocol	802.11
Frequency Range	2.4Ghz – 2.5Ghz
Operating Voltage	3.0 – 3.6V
Operating Current	80mA
Operating Temperature	-40 – 125 C
Package Dimensions	5x5mm
Software Security	WPA/WPA2
Encryption	WEP/TKIP/AES
Interfacing	Android, iOS, Cloud

3.6 Available Component List

In Figure 17 below, the utilized parts for testing for functionality are seen. Table 19 also provides the contents seen in this picture and the associated names for each.

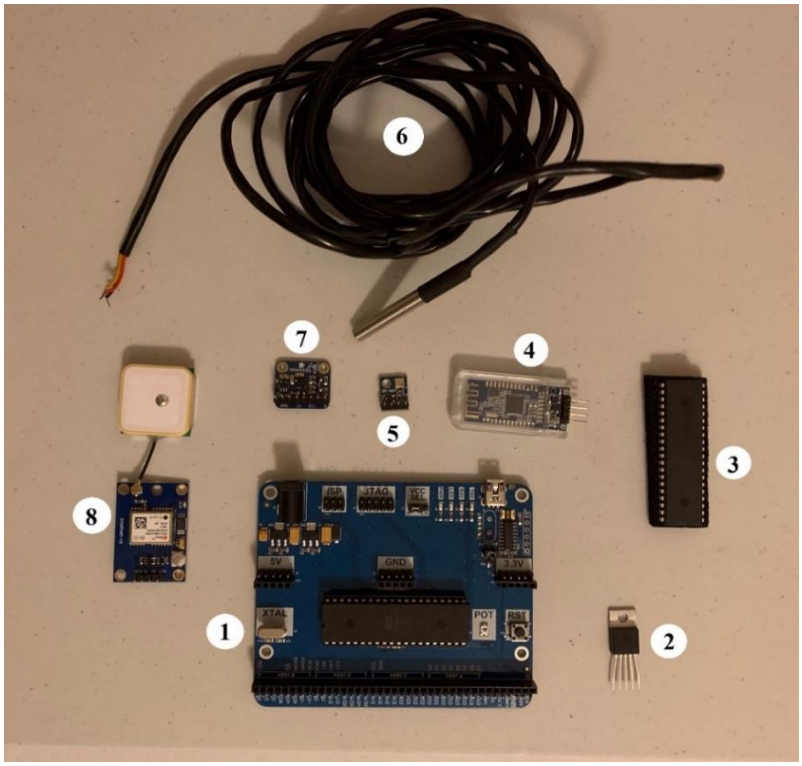


Figure 17:Components for Testing

Table 19: Component List

Number	Component
1	Development Board
2	LM2576 Linear Regulator
3	ATmega1284 MCU
4	HM-10 Bluetooth Module
5	BMP180 Pressure Sensor
6	DB18S20 Water Temperature
7	MMA845 Accelerometer
8	NEO-6 Series GPS

4.0 Standards & Design Constraints

Marine applications placed a high demand on the design constraints with regards to safety standards enforced by the coast guard. As such, various standards directly affect many of the components which were chosen and even deemed some components to be necessary for safe operation of a vessel in state or federal waters. The extent to which these standards influenced each aspect of the design is discussed in detail below with the corresponding requirements to be met.

4.1 Marine Standards

Since our sponsors intend to use this project as a prototype for a future product, all safety standards that apply to commercial boats must be applied to our project. This includes numerous Coast Guard safety standards as well as some electrical standards. We must ensure we follow these standards to ensure we do not injure our boat's occupants, other boaters, or the ecosystem that it is in.

4.1.1 Coast Guard Lighting Standard 83.22

With most marine vessels, the coast guard has standards regarding the visibility of lights aboard to ensure the safety of those on the water. The degree of lighting visibility and intensity of lighting varies with the size of each respective vessel. Table 20, courtesy of the U.S. Coast Guard Navigation center, shows the minimum intensity for each light relating to the range of visibility.

Table 20: Coast Guard Lighting Visibility

Range of visibility (luminous range) of light in nautical miles D	Minimum luminous intensity of light in candles for K = 0.8
1	0.9
2	4.3
3	12
4	27
5	52
6	94

As per the coast guard manual of rules and regulations, vessels exceeding 50 meters must have the following lights and corresponding visibility; Masthead Light (6 Miles), Sidelight (3 Miles), Stern light (3 Miles), Towing light (3 Miles), White/Red/Green/Yellow all-round light (3 Niles), and a special flashing light (2 Miles). Vessels exceeding 12 meters in length but less than the previously stated 50 meters must have the following lights and

corresponding visibility; Masthead light (5 Niles, or 3 Miles if vessel is less than 20 meters), Sidelight (2 Miles), Stern light (2 Miles), Towing Light (2 Miles), White/red/green/yellow all-round light (2 Miles), special flashing light (2 miles). Vessels of a size less than 12 meters must have the following lights and corresponding visibility; Masthead light (2 Miles), Sidelight (1 Mile), Stern light (2 Miles), Towing Light (2 Miles), White/Red/Green/Yellow All-round light (2 Miles), Special flashing light (2 Miles). For this project specifically, the standard of interest is for a vessel of less than 12 meters in size as the specification for the boat is to be 17 feet. As coast guard or Fish and Wildlife Commission checks are likely to occur while on the water to ensure the safety of the user, it is crucial that the electronics controlling the lighting system meet these standards.

4.1.2 CFR 80.371 Public Correspondence Frequencies

While vessels under 20 meters in length are not required to have a VHF radio aboard, it comes with high recommendation by the coast guard to have access to such a device in the event of an emergency. Designated channels for the VHF radio in use are listed by the Federal Communications Commission (FCC) with each serving a role for specific message delivery. Table 21 lists the most common channels and their respective frequencies utilized for maritime purposes.

Table 21: Marine VHF Channels

Channel	Frequency (MHz)	Purpose
6	156.300	Internship Safety
9	156.450	General Purpose communication
13	156.650	Navigational, Lock and Drawbridge Communication
16	156.800	Distress and Safety
70	156.525	Digital Selective Calling (Distress and Safety)

4.1.3 NMEA2000

NMEA2000 was developed as a network standard for marine electronics to allow for the implementation of a Controller Area Network or CAN to send and receive data from various interconnects. With this standard, a range of sensors, wireless modules, and embedded electronics would be allowed to interface freely through a common CAN bus

before eventual connection to the MCU. As listed by NMEA, the layers of a compliant NMEA2000 network are; Physical Layer, Data Link Layer, Network Layer, Network Management, and Application Layer. The physical layer details the electrical and mechanical areas of the NMEA2000 network, specifically the electrical aspects of the CAN. For our project in particular, if this standard is utilized for design then a backbone cable would be used to distribute power to the various sensors before a final interconnect with the MCU. A backbone cable is crucial to NMEA data communication, where respective interconnects are made for the sensors. Following this is the Data Link Layer where the end requirements are dependent on the CAN being used for serial data distribution. Regardless of CAN choice, the standard is defined as using an 8-byte Data field and 29-bit identification field for all serial communication. With this the information sent would be normalized allowing all components to read or write necessary commands with a common language. Further reinforcing this is the Network Layer of the NMEA2000 standard, where addresses of the network and devices are assigned after being connected to the network. Network Management further refines this data transfer through the NMEA network where all devices can interact on a single segment. The final portion of this standard is the application layer in which the received data is compiled through a database using a common set of values or functions.

When looking into the functionality of this standard with regards to our project as a whole, it offers marketability of our design as the NMEA2000 standard is used on many proprietary products utilizing various forms of software. Replacing or switching sensors would be streamlined into a 'plug and play' format allowing for better ease of use for the consumer. However, as non-proprietary information is generally unavailable for this network it is conceivable that interfacing the system with this standard may prove difficult. Nonetheless, it is a viable design choice which may be integrated in the future if desired.

4.1.4 Waterproofing Standards

There are two general waterproofing standards when looking at marine applications; the IP codes and NEMA Standards. To a large degree they are comparable, however minor differences and specialized areas of each may hold applicability to different parts of the electrical system.

4.1.4.1 IP Code

The International Protection Marking or IP Code, described by the International Electrotechnical Commission standard 60529, is in place to provide classification of various levels of protection against intrusion, dust, contact, and water for mechanical and electrical encasings. With aim to provide data on the level of 'waterproofing' or general resistance, the IP code provides a standard for various enclosures and electronics housing to meet for compliance. The IP code in the format is IPxy where 'x' increments from 1 to 6 indicating increasing levels of particle protection, and 'y' increments from 0 to 9 indicating increasing levels of liquid intrusion protection. Specifically, applicable to our project is the IP67 rating, in which the enclosure for our MCU and other electronics were 'dust tight' and capable of water immersion of depths to 1 meter.

4.1.4.2 NEMA

The NEMA standard for enclosure types is comparable to that of the IP code for degrees of protection with regards to particle and liquid intrusion. As such, the aim of providing a rating scale with corresponding degrees of protection is in place. NEMA utilizes a rating system from Types 1 to 13 with intermittent sub-types such as 6P or 12K. For the purposes of this project, the types of interest are Type 6 and Type 6R with both being comparable to IP67 in terms of waterproofing capabilities.

4.1.5 Coast Guard Electrical Code

The Coast Guard has various standards regarding the safe operation of various systems and constructions of boats. The standards outline a range of Federal Requirements in the United States and are act as a reference for the design of modern boats. For this document however, the Coast Guard standards directly concerning our electrical system design will be addressed. Discussions of these topics can be seen in their correspondingly labeled sections, found respectively in Subchapter S, Chapter I of Title 33 in the code of federal regulations.

4.1.5.1 Standard 183.420 Batteries

Batteries have various requirements to be determined code adherent as per the US Coast Guard. There are six main descriptors for these requirements, all of which are to be taken into account when designing the electrical system and boat as a whole. The first requirement requires that each installed battery not move more than one inch in any direction when a pulling force of 90 pounds or twice the battery weight is applied through the center of gravity. Each battery must also be installed so as to ensure that metallic objects cannot come in contact with the terminals of the battery. In addition, each metallic fuel line or fuel system component within 12 inches and above the horizontal plane of the battery top surface as installed must be shielded with dielectric material. The battery must also not be directly above or below any fuel tank or line Note that for this project as the motor is power via electricity and the battery system itself, these two requirements should be satisfied with any design used. However, a vent system must be in place to allow the hydrogen gas released from the battery to exit the boat. The final requirement states that each battery terminal connector must not depend on spring tension for its mechanical connection to the terminal. While primarily concerned with the physical placement of the batteries, the code 183.420 does impose a level of constraint on our electrical system as its physical location will affect wire routing and power distribution to our MCU and sensors. As such, care must be taken when designing the storage method to be compliant with these codes and satisfy our own expectations.

4.1.5.2 Standard 183.425 Conductors: General

As the nature of our project requires a degree of conductor (wire) routing from the MCU to various sensors and the power supply, the CFR regarding conductors is expected to be obeyed. Seven subsections compose this code with each applicable to various designs which may be utilized. For all cases, each conductor must be insulated, stranded copper.

Each conductor must not carry a current greater than that listed for the respective gauge and temperature rating as per Table 22 seen below.

Table 22: Gauge and Temperature Relation to Maximum Current

Conductor size (AWG)	Temperature rating of conductor insulation						
	60 °C (140 °F)	75 °C (167 °F)	80 °C (176 °F)	90 °C (194 °F)	105 °C (221 °F)	125 °C (257 °F)	200 °C (392 °F)
18	10	10	15	20	20	25	25
16	15	15	20	25	25	30	35
14	20	20	25	30	35	40	45
12	25	25	35	40	45	50	55
10	40	40	50	55	60	70	70
8	55	65	70	70	80	90	100
6	80	95	100	100	120	125	135
4	105	125	130	135	160	170	180
3	120	145	150	155	180	195	210
2	140	170	175	180	210	225	240
1	165	195	210	210	245	265	280
0	195	230	245	245	285	305	325
00	225	265	285	285	330	355	370
000	260	310	330	330	385	410	430
0000	300	360	385	385	445	475	510
NOTES							
1. See the following table:	60 °C	75 °C	80 °C	90 °C	105 °C	125 °C	200 °C
	(140 °F)	(167 °F)	(176 °F)	(194 °F)	(221 °F)	(257 °F)	(392 °F)
Temperature rating of conductor	0.58	0.75	0.78	0.82	0.85	0.89	1.00
2. See the following table:	Correction						
	factor						
Number of current carrying conductors:							
3	0.70						
4 to 6	.60						
7 to 24	.50						
25 and above	.40						

In addition, for conductors in engine spaces, amperages must be corrected by the appropriate correction factor as per the table seen previously. Each conductor in a multiconductor sheath must be at least a No.18 AWG conductor. For conductors installed separately, they must be at least a No. 16 AWG conductor. Note that each No. 18 AWG conductor in a multiconductor sheath may not extend out of the sheath more than 30 inches. This is to say that the exposed wire must not extend past the covered/sealed area for distances greater than this. It is important to recognize that these requirements are not necessary to be met for the following; Communication systems, electronic navigation equipment, electronic circuits with a current flow of less than one amp, conductors inside an equipment housing, resistance conductors that control circuit amperage, high voltage secondary conductors and terminations in ignition systems, and pigtails of less than seven inches of exposed length.

Each of these was taken into account for our design as the gauge of wire vared based on the components needs, be it current draw or expected temperature. In some cases, these requirements were not used however; The enclosure for the MCU and control hub for example is exempt due to the final clause in this code. A general wiring diagram for our system with appropriate gauges is seen on the following page.

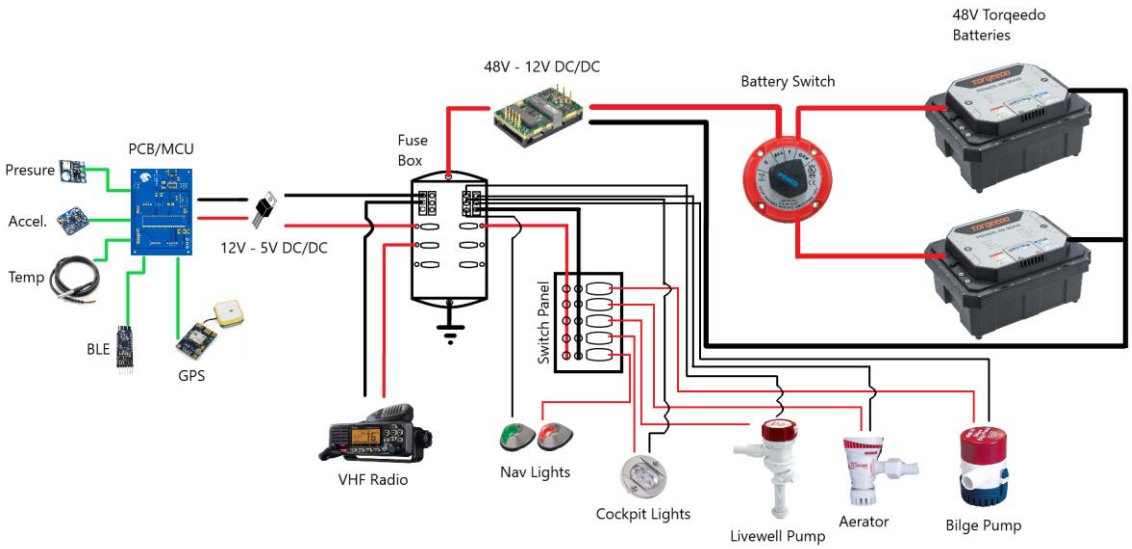


Figure 18: Basic Wiring Diagram With Gauge

4.1.5.3 Standard 183.445 Conductors: Protection

Code requirements for protection of the conductors are relatively limited, however they are crucial to the design of our system and routing/fabrication portion of our interfacing between the sensors and MCU. The first requirement is; Each conductor or group of conductors that passes through a bulkhead, structural member, junction box, or other rigid surface must be protected from abrasion. This is to say that the shielding and routing material used for the wires must protect the conductor(s) to an adequate degree so as to live up to the expected applications of this project. In addition, each undergrounded terminal or stud that is continuously energized must have a boot, nipple, cap, cover, or shield that prevents accidental short-circuiting at the terminals or stud. This is vital to the health and safety of those on board as the large voltages which were utilized bear potentially life-threatening effects.

4.1.5.4 Standard 183.455 Overcurrent Protection: General

Of key interest to this project is overcurrent protection to ensure safety with operation of the electrical system designed. 183.455, as per the code of federal regulations details the measures which are to be taken to provide overcurrent protection for the system. Five main requirements are to be met with several conditions for some of these detailing specific exemptions. In general, each undergrounded current-carrying conductor must be protected by a manually reset, trip-free circuit breaker or fuse. A manual reset, trip-free circuit breaker or fuse must be placed at the source of power for each circuit or conductor. However, this requirement is exempt should it be deemed physically impractical to place the circuit breaker or fuse at the source of power. Instead, it can be placed within seven inches of the power source for each respective circuit or conductor. Should this not be physically practical, this breaker or fuse can be placed within 40 inches of the given power source however a sheath must enclose the conductor for the length. Note the current rating

of each circuit breaker or fuse for this system (Less than 50V) must not exceed 150% of the value for the current given in the table seen in section '4.0.6.2'. Additionally, the voltage rating of each circuit breaker or fuse must not be less than the nominal circuit voltage of the circuit being protected. Note the exemptions are similar to the conductor's segment of standards; Resistance conductors that control circuit amperage, secondary conductors and terminations in ignition systems, and pigtails of less than seven inches of exposed length. In general, this standard is useful in protecting our system and the electrical components which are especially sensitive to fluctuations of current. The microcontroller and sensors specifically stand to benefit from following this standard so as to ensure measures are in place to guard these components.

4.2 Communication Standards

There are many communication standards that will have to be taken into account for this project. Both IEEE and the FCC both have standards and requirements relating to communications systems, and affected how we designed our communications systems. We cannot simply broadcast on any frequency available, and even if attempted, we would be limited by the hardware we have. It is important to respect frequency domains in order to not interfere with other products that may be in the same spectrum as our project.

4.2.1 IEEE 802.15.1

Bluetooth communication played a large role regarding interfacing the MCU with the user's cellular application. As such, the Bluetooth module chosen will adhere to the standard of IEEE 802.15.1. This standard specifies Bluetooth as a low tier standard for wireless communication on a Personal Area Network or PAN. Bluetooth operates at the 2.4GHz frequency with 3 different classifications for the varying effective ranges of a given module. As per the IEEE standard; Class 1 range is approximately 100 meters, Class 2 has a maximum range of 30 meters, and Class 1 has a maximum operating range of 1 meter. As discussed previously, the Bluetooth module chosen for our interfacing will fall under the qualifications for Class 2 and is designed to be adherent to the IEEE 802.15.1 standard. It is worth noting that various subsections of the 802.15.x standard are in place or under development, with some operating under the same frequency band as Bluetooth (802.15.1). Varying methodology with regards to signaling methods however limits or prevents interference between other networks in this frequency band however.

4.3 PCB Electrical Standards

Electrical standards governed the fabrication of this design as a whole, with those discussed below detailing rules to ensure the safety and functionality of our system. IPC standards for the PCB were of key interest as our microcontroller depended heavily on the reliability of this design.

4.3.1 IPC Standards for Printed Circuit Board (PCB)

IPC, the Association Connecting Electronics Industries, has a purpose of creating industry standards along with supporting the advancement of the electronics industry. IPC has led

the industry though changes and has also strived to become the most respected organization for standards and quality programs that support the electronics industry. IPC creates criteria and standards for implementing and printing many types of PCBs including single-sided, double-sided and multilayer. Within the standards, there are many subgroups of documentation, definitions specifications and standards. A few of these subgroups include Material Specifications, Design Specifications, and Performance and inspection documents.

It was vital to follow and pay attention to these standards and requirements when creating and implementing our PCB. Since no member of our team had experience designing or implementing a PCB, this research served a purpose when proceeding with this project. Relevant standards that will assist us in creating a PCB are detailed in the sections below

4.3.1.1 IPC-2221

This standard explains the standards of the designing a Printed Circuit Board Design and the materials that should be documented along with it. This standard first clarifies that certain documents should be listed including the schematic/logic diagram as well as the parts lists. It is important to note that the entire testing of this design was documented in the case where a failure occurs, and it became vital to revert back to the initial designs. Although performed by the printed board supplier, Bare board testing is required to ensure every lone board is properly working before applying a design onto the board. Regarding the layout design of the board, IPC recommends several layout standards. Some of these standards include ensuring all components have testable points accessible, that the design have feed-throughs and component holes placed away from board edges and requires that the board be laid out on a grid which matched the design team's testing concept.

4.3.1.2 IPC-1331

This standard expresses the safety standard for electrically heated process equipment. To begin, this standard requires the users verify the equipment and all manufacturers or suppliers are compatible with the intended solution. This is vital to clarify before beginning the project to avoid any type of danger. Within this standard, many different cases are provided and a description of what to pay attention to when mixing different chemicals or forms of electrical power. There are multiple design and installation requirements and standards which include control circuit design, control system installation, testing and heating system maintenance.

4.4 Software Standards

While software standards do not always seem like they have the same degree of importance as hardware standards, they are equally as valuable. Software standards help guarantee that time and money is not wasted during the software development process. Time can be saved by ensuring you understand the requirements ahead of time, as well as making code that is designed to be easy to maintain going forward. A critical part of making good, maintainable code is to write code that has clear and consistent formatting that other developers can easily read. It is also extremely important to follow the guidelines

of whatever language you happen to be writing in, in order to make sure you follow best-practice for that language and that you will not encounter issues from programming things in unorthodox methods.

4.4.1 IEEE 830

One of IEEE's most important software standards is IEEE 830-1998 Recommended Practice for Software Requirements Specifications. This standard is made to inform businesses and developers how to set good goals and requirements for a software project. The main point of the standard is to develop a Software Requirement Specification, or SRS. Your SRS should detail all the requirements and goals of the project, in order to keep the development process as streamlined as possible. A good SRS should be able to serve as a contract between the customer and developer as to what the product should be able to do. This makes it much easier for the developer to not waste any time on unwanted features and allows them to work towards the goal as efficiently as possible. It also makes it so both the customer and the developer can have an estimate of how long development will take and how much it will cost. You can also then use the SRS to test the product, since the product's requirements are met. It allows the developer to just check that the product meets all the requirements listed in the SRS. Having a good SRS will also mean that if you go to expand on the project after development is complete, you have a foundation with a known state to work from.

4.4.1.1 Role of an SRS

A good SRS has an important role to play in a large project. No matter whether the software is the entire product or just a small subsystem, the SRS should define what the software is expected to do. If it is a part of a larger project, an SRS is expected to include details about the interfaces with the other systems. An SRS should define every requirement of the software, no matter whether the requirement was explicitly stated or was created by the characteristics of the problem itself. An SRS also should not specify anything about the design of the project, only what is required of it. The design should only be done during the design stage of the project. An SRS lastly shouldn't create any additional constraints, since any other constraints to be found should be in other documents.

4.4.1.2 Characteristics of an SRS

The IEEE 830 standard discusses the many characteristics of a good SRS. The requirements of a good SRS are as follows:

- A good SRS firstly needs to be correct, meaning that every requirement listed is one that the product shall meet, and that it encapsulates all requirements desired by the consumer.

- An SRS also needs to be unambiguous, meaning that it must be very clear for everyone to read. This ensures that there will never be a circumstance where the customer and developer interpret the requirements differently.

- An SRS must also be complete, meaning it includes all significant requirements for the project, how the project will respond to all foreseeable input, and has every piece of supporting material fully defined.

- An SRS must be consistent, meaning it must not contradict any higher-level documents such as a system requirements paper. This includes contradictions such as scheduling conflicts, two requirements describing the same element in mutually exclusive ways, or simple logical conflicts such as one saying A must be greater than B and another document saying B must be greater than A.

- An SRS needs to have its elements ranked in terms of importance. While all requirements must be met, sometimes conflicts can occur down the road, and it is important to have an understanding with your customer what requirements they value the most. This also ensures that the developer divides up their time effectively, ensuring they do not dedicate 50% of their man hours to a requirement the customer felt was least important.

- An SRS must also be verifiable, meaning that every requirement included in it has a feasible process to objectively test whether the requirement has been met. This means that a requirement cannot use subjective language such as "it must guess the correct result usually". In this case, we want to rewrite the language in the SRS to be objective, such as "it must guess the correct results 69% of the time". This ensures that the developer can always know that the product objectively meets every requirement in the SRS, and that the customer won't be able to refute them.

- An SRS must also be modifiable. Requirements can often change after the SRS is written, so it is very important that the SRS can be modified as needed. IEEE explains that you can do this by keeping your SRS organized with an easy-to-understand structure, not allowing duplicates of a requirement, and by making sure each requirement is stated independently of all other requirements. This will allow the developer and customer to make changes as they agree are needed.

- It must be easy to determine where every requirement originated from, making the SRS traceable. This means you should be able to tell which documents the SRS references and which documents reference the SRS. Backward traceability means that you can find all documents the SRS references, which is usually accomplished by stating where a requirement came from explicitly. Forward traceability means that all documents that reference the SRS can be traced back to the SRS. If all requirements have a unique identifier, then simply including the identifier of the requirement is usually enough to accomplish forward traceability.

4.4.1.3 Making an SRS

It is extremely critical that an SRS is made with both the developer and customer present. If the customer or developer tries to write the SRS on their own, it will always be incomplete. The developer will never fully understand the wants and needs of the customer and will likely disappoint them if they try to make the project without the customer's input

in the requirements. Likewise, a customer is usually unfamiliar with the software development process and will not understand what is or is not feasible to accomplish. This could lead to many poor design decisions if the customer prepares the

SRS by themselves. The only situation where this may not be true is if the system's requirements and the software subsystem's requirements are being developed concurrently.

Not allowing the evolution of an SRS can cause critical failures down the road. Requirement changes are nearly inevitable in every project, so trying to set the requirements in stone from the get-go is never a good decision. Even if you know a revision is inevitable, you should not revise it ahead of time at the expense of the clarity of the SRS. The SRS's completeness should be valued above its ability to be future proofed. It's important to create a process to formally change your SRS from the start, so that you can create procedures that will prevent you from causing more issues than you fix.

Prototyping is a key task in any project, and an SRS is no exception. Making a short prototype of an SRS will make it more appealing for a customer to review and provide feedback on, which will it make it less likely to need revisions down the line. Prototyping an SRS will be much faster than writing a complete draft and will help you to identify flaws in it earlier on, thus saving you precious development time.

An SRS should never specify the implementation. Think of an SRS as a description of the problem, not a description of the solution. It should always be open to various solutions, and shouldn't specify which one to use. This will help remove any bias that may be had, leading to the best solution possible.

4.4.1.4 Parts of an SRS

IEEE 830 also specifies what should be included in the SRS, which is as follows:

- An introduction to the problem, which includes the purpose, scope, any needed definitions, references, and an overview of the issue that you are trying to solve.
- An overall description of the things that will affect the project and the requirements you set for it. This should include sections about the product perspective, product functions, user characteristics, constraints, assumptions, and how you will select requirements.
- All the specific requirements that the developer and customer agree upon. These requirements should meet the conditions discussed earlier and should be easy to read and understand.
- Supporting info such as a table of contents, an index, or appendices.

4.4.2 Embedded C Coding Standard

Most embedded systems in use today run off the programming language C, or some derivative of it. Therefore, it seems critical that we discuss standards for writing software in C. When it came to choose which C formatting standard we would follow, we decided to follow the "Embedded C Coding Standard" book published by the BARR group due to its focus on embedded programming. The book covers nearly every caveat of formatting for the C language, and we will cover most of it below. This standard helps ensure that our software is easy to read, and easy to maintain for any future developers.

4.4.2.1 File Format Guidelines

The overall schema of your C program files will play a critical role in the organization of your software. If you don't organize the file properly, it will be incredibly hard to find the portions of the program that you are looking for. Also, not following these guidelines can create numerous small issues that will become annoying down the road. One of the biggest rules from this standard as well as many others is that a single line should not expand to column 80. Going past column 80 will make it so the code will not be readable in the default command prompt window size, which can make it a hassle for others to quickly look at the code, or for an error to point you to where in a line an error occurred. Also, if you go past column 80, it can be very hard to read if you have to do a code review with your code printed, since it will have to wrap to the next line or will overflow off the page. Going past column 80 almost makes it extremely difficult to do side by side code comparison since your editor only be as wide as half your monitor. The order in which you place your code is also very important. The order is as follows:

1. A short introduction comment to explain what the program does. Should also include any authors, copyright info, and version control info.

2. A list of included header files. Any built-in or custom libraries that you use should always be together at the top of the program.

3. Any definitions included. Since these are done in the preprocessor stage, we do not need them intermingled with the rest of the program. Keeping them at the top will make them easy to find and edit. We follow the order constants macros, function macros, typedefs, then finally enums.

4. Any global variable declarations, and if the you have them, it should be externs, then non-statics variables, and finally static variables.

5. The functions used in your program. These will be the final part of your program, and you should order them using whatever method will best suit your implementation. If you're unsure, put the highest-level functions at the top, and the lowest level functions at the bottom.

Following this order will ensure that any other programmers that look at your code will quickly be able to find whichever part they are looking for. Additionally, local variable

declarations should always be put at the top of the function they are contained in, and no variables should be declared mid-function. This keeps it so it is easy to keep track of all variables that are being used in a function. All conditional statements and loops will utilize curly braces, even if they only contain a single line, to ensure easier code maintenance. Additionally, when doing comparisons in a conditional, the constant must always be on the left side of the comparison operator.

4.4.2.2 Whitespace

Whitespace is a key factor into the readability of your code. It determines how easily you can identify groups of code and where new sections begin. Below are a few of the rules to follow for whitespace:

- Four spaces shall be used for indenting a nested block of code.
- Every block of code mentioned above, as well as each individual function, should have a blank white line separating it from other blocks.
- Each function declaration should have a space between the function name and the parenthesis.
- An opening curly brace should be on the line after the line it is attached to, in the same column as the first character of the line it is attached to. The close curly brace should be in that same column as well.
- A unary operator such as ++ or -- should contain no whitespace.
- Any binary operators such as +, -, = etc. should have a space on both sides.
- In all for loops, there must be a space after each semicolon in its elements.
- Each comma in a function's parameters must be followed by a space.
- When declaring a pointer, there should be a space on both sides of the *. If you're not declaring it, there should not be a space on the side that the variable name is on.
- The square brackets on an array declaration will contain no spaces.
- The parenthesis for function calls should not have spaces on the outside, except for during the function declaration.
- Semicolons that are ending a line shall not have a space before them.
- There should be no excess blank lines, except when separating blocks of code.
- The two operators used to dereference points shall not have a space on either side.

4.4.2.3 Comments

Commenting is an incredibly crucial part of all software development. It helps keep your thought process clear to your future self and to anyone else who reads your code. Having good comments can make code incredibly easier to understand and can allow even intermediate programmers to be able to understand complex code.

- Every block of code should have a comment explaining its purpose, as well as all functions and any particularly complex pieces of code.
- All comments should start with a space at the start for readability and should be written as concisely as possible.
- Comments should be placed directly above the code that are describing and should be at the same indentation level.
- Either inline comments or block comments are acceptable to use in C, however an inline comment is preferred to ensure code is not accidentally commented out.
- Comments should not include any preprocessor characters.
- Code should not be commented out, use a preprocessor to disable it if needed to avoid potential unintended consequences.
- Use WARNING, NOTE, and TODO for leaving important messages for future developers.

4.4.2.4 Data Types

Working with variables in your code will be what you are doing the vast majority of the time, therefore it is extremely important that they are named and declared consistently, to make them as understandable as possible. These naming conventions will help make it clear what is happening in the code to yourself and to any future developers.

- All new types should end with _t, indicating is a type and not a variable name.
- Variable names should use either camelCase or underscores to make them more readable.
- Fixed-length integers shall be used in place of native data types such as int, char, and long.
- Fixed-length floating point types shall be used in place of typical floating-point types.
- Do not do operations using both unsigned and signed integers.
- Do not check for equality of floating-point variables.

- The smallest data type required for proper functionality shall always be used
- Function return types and parameters types must be explicitly declared, even when it is void.
- All variables shall be initialized.
- All local variables will be declared at the top of all functions for easier configuration and to make it easier to keep track of variable usage.

4.4.3 IEEE 829

IEEE standard 829 details a universal framework for the test of software. This standard is intended to apply to all software life cycle models. It details methods that should be followed when testing software, what documents should be made, and how it fits into the software development life cycle. It discusses different levels of testing, and when each of them is applicable.

4.4.3.1 Objectives of Testing

Software is composed of many moving pieces. It is always important to not only ensure your software meets the needs of the system, but also that it interfaces properly. IEEE 829 defines several reasons that software must be tested, listed below:

- Ensuring the environment that the software is contained within is accounted for. This includes considering the entire range of conditions that may affect the software.
- Making sure all possible user input is considered, and that all needed statuses are properly displayed back to the user. This means that the software should never be able to enter an undefined state, no matter what input the user enters. This also means that the software is verified to be following all procedures and protocols that apply to it, such as security protocols.
- Ensure that the software interfaces with the hardware properly, and that it can respond properly to hardware failures.
- Ensure that the software properly interfaces with all other connected software. This can help to guarantee that the software will fit in the environment without issues.

By testing our software thoroughly, we can guarantee that we meet our system's requirements and that we are satisfying our customer's needs.

4.4.3.2 System Integrity Levels

IEEE 829 also details integrity levels that may be used to identify what parts of a project are the most sensitive and will require the most attention. This helps the developer to better decide how to dedicate their time when performing software test. There are several ways

that these integrity levels can be defined, often times based on the system. IEEE recommends the consequence-based integrity level scheme if you are uncertain what to use. Below are the details on what is included in each of those levels.

- Level 4 - Catastrophic. This includes parts of the software that absolutely must work, or horrible consequences will occur such as complete failure of the system, loss of life, or extreme economic damage. It is impossible to mitigate a level 4 failure once it has occurred.

- Level 3 - Critical. This includes parts of the software that must operate properly, or the purpose of the software will be nullified. This can also include failures that cause things like injury, degradation of the system, or environmental damage. Level 3 failures can typically be partially mitigated after they occur.

- Level 2 - Marginal. This includes parts of the software that will cause minor issues when they fail. This could be things such as returning incomplete data or user interface issues. Level 2 failures can typically be completely mitigated.

- Level 1 - Negligible. This includes parts of the software that will cause the functionality to have small issues. Things such as small bugs, minor performance issues, edge cases, and inefficiencies in the interfacing. Level 1 failures typically require no mitigation.

4.4.3.3 Process Breakdown

IEEE 829 specifies the methodology for breaking down a process into smaller tasks in order to be thorough and have everything well documented. It is composed of three different levels of broadness. The highest level is a process, which is composed of related activities, and turns inputs into outputs. Activities are is a group of work to be completed, and typically has an estimated duration and cost. Finally, activities are composed of tasks. A task is either a test case or a test procedure that will need to be executed. By breaking processes down into activities, and activities into tasks, it allows you to more accurately track what is complete and what needs to be completed. IEEE 829 also recommends using the processes detailed below, to ensure all aspects of the project are tested.

- Management - This process is focused of the creation of the test documents, as well as the overhead control of the project. The biggest part of this is making sure all procedures are followed. It also looks at the analysis of failures and assessment of issues that are found. Management is also responsible for ensuring all results are complete and accurate. Lastly, management should include the determination of when other tasks are completed. If the project owner is not the management itself, management will be in charge of reporting status updates to management.

- Acquisition - This process is completely focused on choosing and ordering any parts necessary for the project. In relation to test, tasks are often focused on ensuring suppliers are meeting their specifications. Tasks may involve things such as finding supplier test documentation from their sources, to determine what quality

testing has been done by them, as well as identifying what specifications are guaranteed and tested for. Another major set of tasks in this process is performing acceptance testing for the received parts. This is extremely critical, since using a faulty component has the potential to damage critical systems. This process is also a good place to evaluate all system interfaces and ensure that the system requirements capture all these interfaces.

- Supply - The supply process is when the product has been contracted by a customer. For testing, this usually entails altering system requirements according to the customer's demands. This is a good time to provide a scope for what testing will be completed and identifying how you will measure the success of your product. It also is important to ensure the product's external interfaces match the needs of your customer, and to alter them as needed.

- Development - This process includes tasks made for the development team of a software project. Overall, this process includes evaluation of requirements, design, and the actual coding of the project. Related to test, this process is focused on verifying the initial product that is produced. The exact details of this project will vary based on what software lifecycle procedure your team follows, but there are some tasks that are likely to be included regardless of which system your team uses. One of the main things for test to do during this process is evaluate whether the conceptual product will meet the requirements that are set. This means you should evaluate the preliminary architecture to try to find any issues that may prevent it from solving the problem that your customers are facing. Considering the traceability of the conceptual is a key part of this process, as it will have a big impact in your later testing. It's also a good idea to begin identifying the integrity level of various component of the conceptual solution at this point. While identifying the integrity level, be sure to consider what risks are apparent in the initial design, and what security issues may arise through this approach. At this point you should ensure your test plans are completed, so that it does not become a holding point later on. As development reaches later stages, it is extremely critical to perform component testing and unit testing. This can ensure that all components of the system are completed before testing them as a whole. It is also important at this point to ensure all the systems you need in order to test the complete product are in place so testing can be completed as soon as possible. Reporting discovered issues in a timely matter. The last activity of this process is usually testing focused. During this activity, your initial systems test should be taking place. This stage will involve performing tests, preparing reports, and reporting results as needed.

- Operation - This process is focused on full system testing as well as supporting users. This means refining user interfaces and performing tests on them. Operational testing will be held in this process, and should be used to find bugs and issues with the system as well as its interfaces. Risk and security testing should also be held at this point, to ensure the project is safe to release to customers. At this point, you should also be testing the full product against its system requirements, to guarantee that it meets the customer's needs. The test effort here should be appropriate for the integrity level you are testing against.

- Maintenance - This process begins after the product has been released and a modification must be made. Whether this is a fix, update, expansion, or adding new support, it will enter the maintenance process. This usually happens when bug fixes occur, the customer wants additional features, or when you must support new versions of the software in your environment. When in maintenance, testing will be done very similarly to when you're in the design process. All pieces must be retested, so that they may be verified with the changes. While this often seems like overkill, often times small changes have unintended consequences in other parts of the software. Each piece should also be tested against its level of integrity once again, to be certain that critical portions of the project are still safe. Along with performing the same testing as in the development process, it is important to revise all test documentation that may be affected by the changes that were performed. Also, perform additional testing for anomalies in portions of the project that were changed, to ensure new, untested issues haven't began.

4.4.3.4 Documentation

Test documentation is extremely crucial to any large project, as it is one of the last things to be checked before a product is released to the customer. Having clear and concise testing documentation will guarantee that anyone who needs to review it will have a good understanding, and that they will understand the implications of what is written in it. While this seems easy to understand, IEEE 829 has detailed many of the pitfalls that you can fall into when writing test documentation.

One of the most important part of test documentation is providing a reference to anything referenced in other documentation. If proper references are not included, people reviewing the test documentation may second-guess what's written, and waste valuable time trying to find their own sources. Providing clear references provides a clear source to why certain points are made, which helps alleviate any potential debates from occurring.

Eliminating excess content is another key part of making quality test documentation. There are many places that content can be reiterated, but adding to test documents will make them cluttered, less clear, and more likely to causes mistakes to be made. First, be sure not to cover topics that are covered by the process. Not all software life cycles require all documentation to be made, and it is key to only make the documentation that is required by your organization. This is another place that could lead to time being wasted for what will be, in the end, a lower quality document. Be sure also not to discuss topics that are completed by automated tools. Long descriptions are not needed for automated tools that have already been validated. Since these tools are validated and require no human input, we can trust their results without discussion. Lastly, be sure to combine and eliminate as many topics as you can justify. By keeping your documentation as condensed as you can, it will make it clearer to anyone who has to read the report. Keeping tangential information in the report will only serve to waste time and confuse the reader.

4.4.4 Agile Development Cycle

Agile is one of the most prevalent software development lifecycle methodologies in use today. It was created in 2001 in order to make a software development approach that was more focused on a few key aspects: focusing on individual performance, focusing on software development rather than documentation, working with customers throughout the development process, and being responsive to change. The creators of the Agile development process felt that the methodologies in use at the time were too stringent, and that they were not flexible enough to be able to respond to the changing needs of the product. They saw that too often software development was focused around a spec sheet, and that it took far too long for changes to be made. This led them to create Agile, which is largely focused on software development teams being flexible and responsive in order to be able to faster enact changes that may occur during development. There are 3 main roles in the Agile development process. The first is the user, who is the one who will be using the product and is having their needs satisfied by the product. The next is the product owner, they are responsible for determining how the needs of the customer will be met. The product owner is responsible for determining the goals of the product, and working with the development team to make sure they are reached. The last role is the software development team, who is responsible for creating all the pieces of the project.

4.4.4.1 Scrum Meetings

Since Agile is more focused on creating a working product rather than meeting a spec sheet, communication is key to ensuring that all pieces of the product will interface correctly. In order to do this, the development team should meet very frequently. They do this by holding frequent meetings called scrum meetings. The focus of scrum meetings is to plan sprints. Sprints are a period of time that are designated to create a piece of the product during. At the beginning of a sprint, priorities will be decided in scrum meetings to determine what pieces will be developed during that sprint. The next step is to commit to certain user stories in order to decide how much work can be done and what work will be completed during the sprint. For the rest of the sprint, the scrum meetings will consist of daily status updates on how progress is going on each user story, and what strategies they have been using. This allows all team members to be up-to-date on what the interfaces between each piece of the project will look like.

4.5 Constraints

Various constraints are placed on this project, with some bearing more importance than others with regards to design requirements to be met. As discussed previously, there are a sweeping range of marine standards to be enforced by the coast guard. Sponsor satisfaction and their availability will also impose potential constraints regarding the design and fabrication of this boat. These and other constraints placed on our boat design will be discussed further in the subsections located in the following pages.

4.5.1 Environmental Constraints

With a project such as this, interacting with the marine environment and ecosystem was to be expected. As such, it was of great humanitarian and ecological concern to ensure the electrical system aboard this system had as little of an effect on the environment as possible. Having a purely electrically powered system eases the level of pollution and strain on an already fragile system. The reduction of noise pollution with the intended design of a "Silent aluminum fishing boat" also lends to solving a considerable problem with typical gas-powered outboard motors. To this extent, the chosen/provided electric motor by Torqeedo does an excellent job in meeting a wide range of environmental constraints to ensure some level of 'Green' design in our system.

Further influencing the environment is the interacting of the boat and exterior sensors which came in contact with the marine environment. It was crucial that non-organic materials and well protected heavy metals were utilized to limit water pollution. As such, corrosion was mitigated with proper sheathing and shielding for the devices in question. The expected nutrient and organic deposition in a marine environment will govern the extent to which the environmental constraints are employed, however it is an ever-present factor which must be addressed.

Generally, the health of the environment and marine life in the expected environment was of interest with regards to the design of this boat. Specifically, with the expected implementation in coastal or inshore Florida waters, the prime example would be seagrass and manatees. With the use of an electric outboard motor, while noise pollution is limited, there is a risk of contact with manatees or other sea life. As such, in areas of shallow water where destruction of seagrass beds or marine life harm is apparent, the electric motor will be disabled in favor of a 'Push Pole' or manual method of propelling the boat. As protected species are observed to inhabit the areas of intended use, it is of critical design choice that our exterior sensors and electronics do little to interfere with said marine animals. To a great extent, these environmental concerns were to be observed and played an integral role with component choice.

4.5.2 Economic Constraints

With all senior design projects, there is an inherent economic constraint placed on any design. As our project is sponsored however, the constraint was outlined and design choices must meet this to ensure satisfaction of our sponsors. With an intended cost not exceeding that of $1500, the electrical system aboard the aluminum fishing boat would have to be low cost while also ensuring that the necessary standards enforced by the coast guard were met. Marine electronics and capable equipment is generally of greater cost than typical methods. The sponsor has outlined that this design is to be as low cost as possible while reaching the desired goals and their satisfaction is of top priority. As such, the economic constraints for this system are of great importance.

4.5.3 Health and Safety Constraints

Of chief concern to the design and use of the electrical system are the health and safety constraints which are ever prevalent in a product such as this. User safety is paramount with the application of our design, as their life may quite possibly depend on the functionality of our systems. As per the coast guard standards listed above as well as other risks seen while boating, there are a plethora of safety constraints which are placed on our equipment.

Specifically, the lighting system which is to be used must meet desired standards and be functional at all times while the user is aboard and using the electrical system. Low light scenarios and collisions are an ever-present threat when dealing with the marine boating industry and necessary precautions must be available to prevent such an event. This is why the choice to hardwire the lights to the battery/step down directly rather than be controlled through an app is of crucial design concern. Should the user's smartphone die or Bluetooth connectivity falter, a necessary safety precaution on any boating vessel would be made unavailable. Fewer components to fail in the way of a functioning LED or lighting system is of the best interest for the user and our design as a whole.

As well, the power delivery and distribution system via the chosen regulators and step downs is of great scrutiny with regards to the user's safety. Not only is the system quite dangerous with the use of the 5000kW batteries being utilized, the distribution of power is to an array of sensors and electronics which provided the user with crucial information. For example, the chosen GPS module was routed through the MCU and any degree of power failure would leave the user potentially lost or without any form of navigation software. In an unfamiliar area this could have drastic effects on the safety of the user and other passengers. Of course, proper sheathing and routing of the wires carrying any amount of voltage was monitored and carefully designed so as to not harm those who may come in contact with it.

Of question regarding safety constraints is the reliability of the equipment to be used, especially under potentially corrosive and harsh salt water environments/applications. Waterproofing of rating IP67 is desired for all electronics not sheathed, and the reliability of the batteries and motor provided were taken into account. As these are sponsor provided items in use on many boats already on the market, they are expected to meet the safety constraints expected with any boating application.

4.5.4 Time Constraints

With any design and implementation comes the constraint of time; Be it fabrication time or design time. The design and fabrication phases of this project are expected to take approximately one semester each, in senior design one and senior design two respectively. With a total time period of around eight months, in addition to sponsor implemented milestones and expectations, design choices are critical when considering the time each will take to implement. Stretch goals are present, however given the time constraints for the design they are unlikely to be met. Of primary concern regarding design choice and time allotment are first the features to ensure the safety of the user, second the features

expected by the consumer, and thirdly additional amenities to improve user experience aboard the boat. With many of these features comes the design and fabrication time associated with a PCB and as such this will play a vital role with regards to providing a functional system.

To ensure the design and fabrication constraints are met, a schedule with project milestones was outlined in the administrative portion of this document. It is important to make note that these milestones were flexible to change should a more efficient design or idea present itself. Notable is the time required for the mechanical and computer science teams to produce their respective designs and models. As the mechanical team expected a full model to be completed in early January, some components such as the bilge, lighting, and live well sensors were not ordered until final integration. However, changes were made primarily on the allotted time to complete this project.

4.5.5 Fabrication Constraints

Fabrication constraints are of real concern with this project as there are a litany of standards and requirements for the overall design of this project. While the design of the hull was handled by the mechanical engineering students, our electrical system had to be integrated properly with their design to withstand the pressures and strain associated with the marine environment. The goal set forth by the sponsor was to utilize off the shelf sensors to enable easy user accessibility while guaranteeing a level of performance ready for market if desired. Choice of materials and electronics is important as there must be a balance between quality and price. Accordingly, quality was of top priority with cost being the deciding factor should a choice between two similar components arise.

Communication with the sponsors and proper utilization of the tools is also a constraint with regards to the fabrication of this system as well. Due to the large-scale nature of this design and construction, off campus housing for the boat and power tools will no doubt be needed. This is to say that the boat as a whole is directly influenced by the sponsors availability to assist in the fabrication of the larger components, such as the hull and proper motor mounting as we the students lack the necessary tools to do so.

4.5.6 Ethical Constraints

The ethical constraints regarding this project are primarily aimed at controlling the quality and originality of the final product. With limited components or systems comparable to this on the market, it was of our best interest to ensure the product represented an original design which met the requirements set forth by our sponsor. As this project is funded by a sponsor, our duty as students was to perform work up to the expected standard which reflected positively on the university as a whole. With the project bearing the weight of a prototype design with a potential future in market, it was also expected that our system would meet all user demands in an ethical manner.

In this vein, with a product that is being produced as a prototype with potential market aspirations such as our boat or electrical system, it was our responsibility to apply all the required constraints and standards discussed previously.

5.0 Design

Based upon the research of relevant background technology while also researching and making final selection of components for use within the project. Constrained to these selections are also the relevant constraints, protocols and interfacing methods that belong specifically to each of the components.

This section contains various topics regarding the design of the system for our project:

- The conceptual flowcharts and block diagrams of the hardware and software systems and subsystems that show how they are built and interfaced together.

- Schematic diagrams and software diagrams for exactly how each component individually should be placed within their respective environments and what external components should be added in for proper function of certain devices as well as the software foundation of interfacing all components.

- Entire schematic designs for the complete design of communication systems, sensor and tracking systems and power delivery for the system.

- Each component will also be discussed in more detail about how they will tie in to each part of the system as a whole in both the hardware and software areas.

It is important to recognize that the integration issues which was faced moving forward with the design were taken into account as well. As the mechanical team refined the design moving into the fabrication phase, the organization of sensors and location of wire routing were subject to change. To this extent, these areas were of less scrutiny and more available to flexibility with regards to modifiability to suit the end design. Quality user experience in addition to safety was the main concern when designing our final product and this was reflected with component choice and the implementation of the electrical system as a whole.

5.1 Hardware Design

Overall the hardware design of the entire boat system included the microcontroller subsystem, the communications subsystem, sensor subsystems, extra amenities and the power supply subsystem. All of these subsystems were designed with the standards and constraints laid out within the report earlier while at the same time attaining the goals of the project as a whole. The PCB design contained the microcontroller base and all external components and power delivery required while all other components were attached via their respective shields or through wire interfacing. Our design mostly revolved around the use of one microcontrollers unit.

5.1.1 DC/DC Conversion

One of the most important factors to the successful implementation of our entire design was the fact that the power delivery to all components must be reliable while at the same time remaining as efficient as possible in order to not sacrifice operating time from the battery. As the main power draw of the battery was powering the motor of the boat, controls and any other amenities on board it also needed to power our entire system design reliably without causing error in operation to any other parts of the system or boat itself. An important aspect to keep under consideration was the fact that the boat needed to power its bilge pump and external lighting via the same battery as well; the energy consumption and use of these components was not taken into account within our systems design because these components are required for the boats proper operation and could not be modified in any way. The first step in order to achieve this objective, was that the battery voltage must first be stepped down to 12 volts while providing high amperage in order for the boat to run the bilge pump and lights that are on board. For this a simple transformer, better known as large buck converter was used to step down the voltage. A basic flowchart of how this design will work is show below in Figure 19:

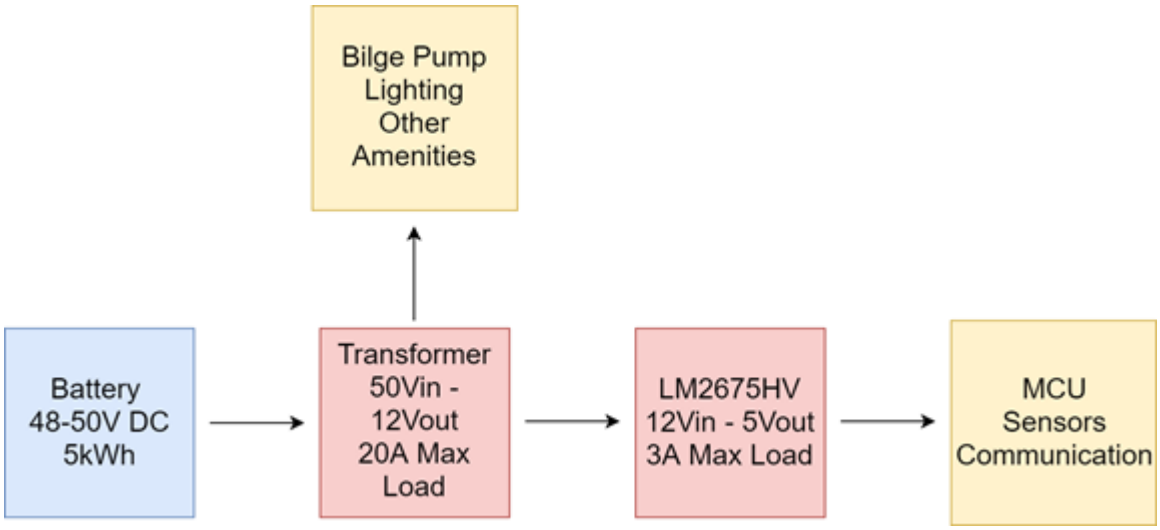


Figure 19: DC/DC Conversion Flowchart

An important factor to achieving a reliable design with the LM2576HV switching regulator was ensuring that any changes in load did not affect the voltage output and current provided to the circuit to drive all the components. This factor was even more important because as a user is either starting up or increasing throttle on the boats motor, the load of the motor drew a lot of the current in order to properly operate and maintain cruising speeds. Also taken into consideration was when the driver of the boat decided to run the bilge pump as that will also sink current along with the motors constant running, such while all of this is going on it was imperative that the LM2576HV could still provide constant power to our system as the loads outside of it are changing as well. Another significant factor of the regulator design was the generation of heat, meaning the switching regulator generated more loss as heat due to voltage changes at its input. Given the fact

the system only works with DC voltages there was no need to add a rectifier into the design like other regulators that work with AC, this further simplified our needs. The LM2576HV regulator is also considered an LDO, meaning the dropout voltage for when regulation stops happening is low. Thankfully this was not a problem for our design because given the fact the batteries have a turnoff voltage of 36V, we were above the dropout voltage for our design for the duration of the battery life. The goal of the regulator circuit was to take in the 12V input from the transformer and step it down to 5V for in order to power the ATmega1284 microcontroller and this 5V was also simply divided for our modules and sensors that required 3.3V for logic or power. Since the LM2576HV was already decided on its use during the hardware and component selection section of the report all that was left to be done with the regulator was design its schematic. Using the Texas Instruments online WEBBENCH switching design tool the team entered the following parameters shown in Table 23 in order to achieve a preliminary schematic of how the regulator circuit should be designed.

Table 23: WEBBENCH design parameters

Input Voltage	Output Voltage	Maximum Load Current	Ambient Temperature
12V	5V	3A	30°C

Inputting the above parameters into the WEBBENCH tool this following schematic was produced, which was a great starting point of how our DC/DC converter circuit should be laid out.

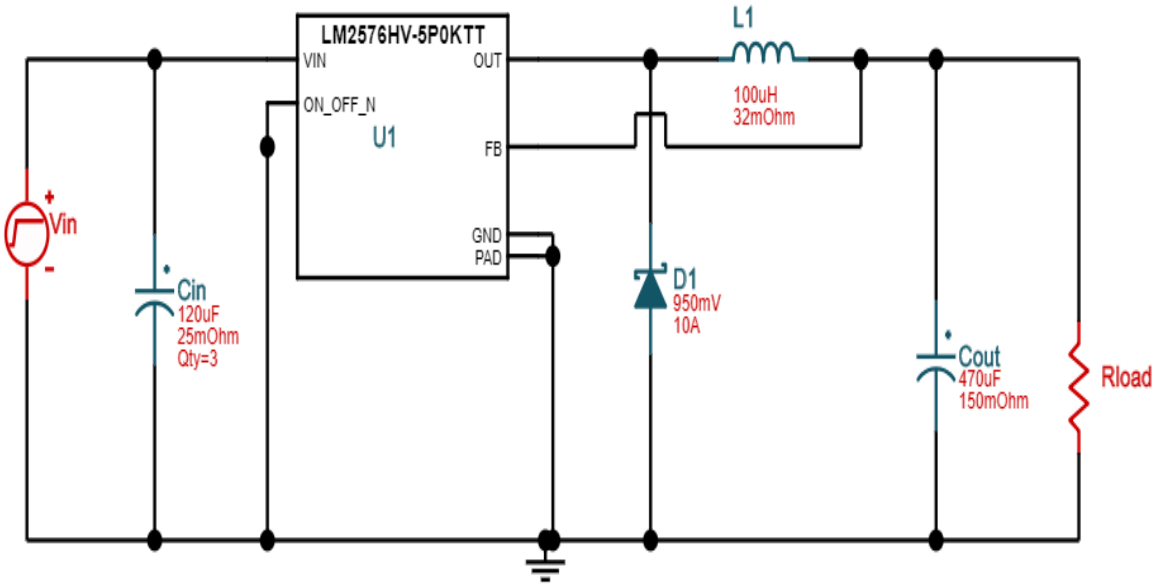


Figure 20: LM2576HV Schematic

As can be observed in the above schematic aside from the regulator itself, the design inherently required a very small number of external components which was ideal when it comes to designing the PCB for the DC/DC converters which saved board space. Of course, the less board space we needed the cheaper the PCB was to manufacture as long as the constraints and standards were met. Found below is a table of the amount of current consumed from each component compared to the total current capable on being handled by the LM2576HV.

Table 24: Devices powered by the switching regulator

Component	Quantity	Operating Voltage	Max Current Draw	Power Consumed
MCU	2	4.5V	40mA	.18W
Bluetooth Module	1	3.3V	20mA	66mW
GPS Module	1	3V	10mA	30mW
Temperature Sensor	1	3.3V	1.5mA	4mW
Accelerometer Sensor	1	3.3V	2µA	6.6µW
Pressure Sensor	1	3.3V	0.5µA	1.65µW
Depth Sensor	1	12V	1A	12W
Various LEDs for 'states'	3	2.5V	100mA	750mW

Based on the table above and knowing the fact that the switching regulator can only handle a load of maximum 3A we observed that all of our components current consumption were below this limit and thus our switching regulator was safe from overheating and from not being able to handle the load. Another benefit of not being close to the current limit was that the device ran at a better operating temperature and since the load isn't too large the voltage regulation differences during switching or load shifts was quite small. This was also another benefit because some electronics are quite sensitive to even small voltage changes when they are used to a specific operating voltage, bearing this in mind the more stable our regulator the more reliable our entire design was. Any components not stated within the table above for the 5-volt switching regulator were powered by the transformer

as those components as discussed before needed a higher voltage and could not be modified for our design.

One of the best advantages of using this simple 5 pin switching regulator was the overall low cost of the regulator itself along with its external components. Although this wasn't the most cost intensive part of the design in comparison with other components, it was still great to cut costs wherever and whenever possible. As such the total item costs and amounts for the regulator circuit can be seen below in Table 25:

Table 25: Bill of materials for the switching regulator

Component	Parameters	Quantity	Cost of Component
LM2576HV	10-60Vin 3A Max	1	$1.84
1N5822 Diode	40V 3A Max	1	$0.50
Input Capacitor	1000uF 63V	3	$0.75
Output Capacitor	100uF 35V	1	$0.25
Inductor	100uH	1	$1.47
Total Cost			$4.81

Following the design based off of WEBBENCH the following schematic in Figure 21 was made within easyEDA for our switching regulator circuit which was used. As can be viewed within the schematic there is 3.5mm screw terminal jack for voltage inputs and outputs since we plan for the DC/DC converter to be on its own PCB while our main microcontrollers had their own standalone PCB as well. Also noted the 5 pin design is compact and for the final PCB design this lead to quite a small board such that we could conserve cost when the manufacturer processed the order for our team.

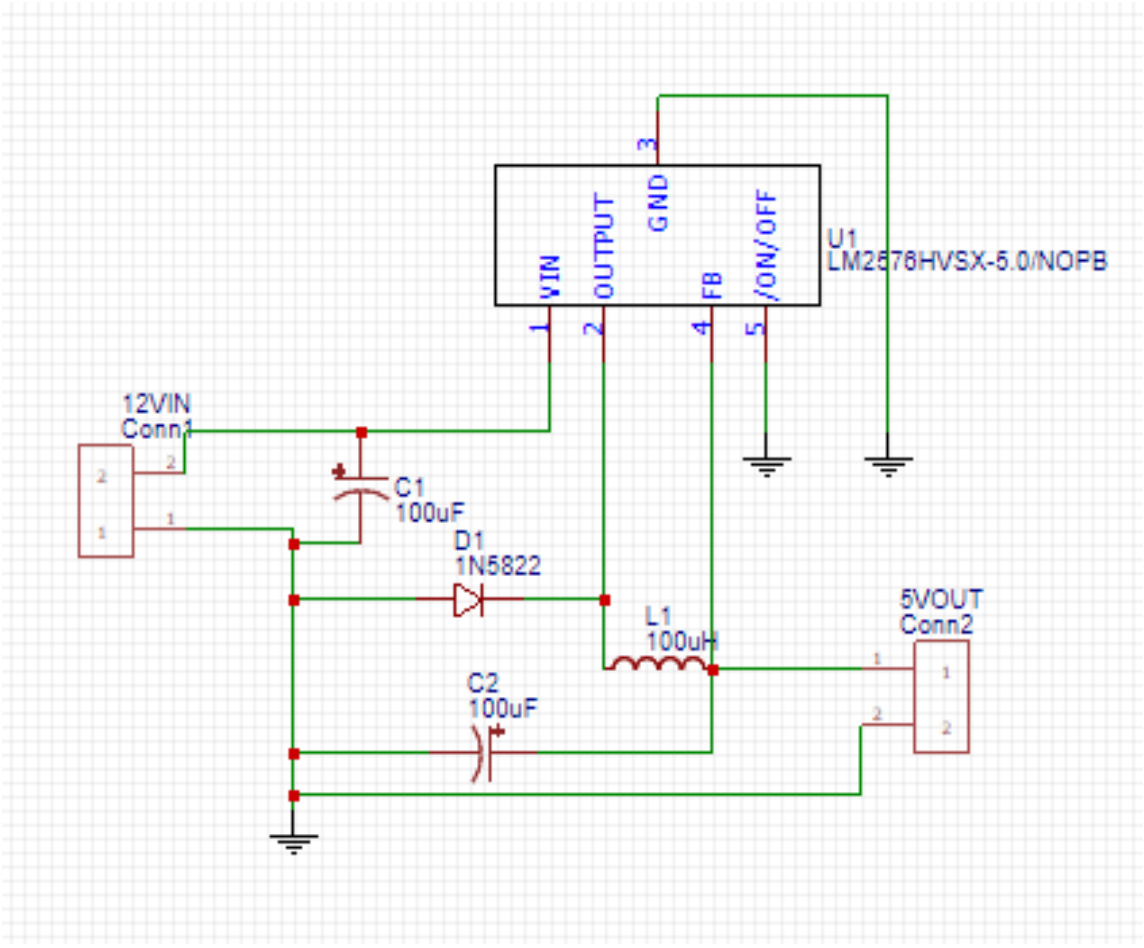


Figure 21:DC/DC Converter Schematic

Although not seen on the schematic but can be observed on the LM2576HV regulator itself in real life there is a pad on the back of the regulator. This pad also acts as a ground for the regulator and more importantly is a way for heat to escape for being inside the regulator and dissipate into either the board or through a heatsink that can be attached. As can also be observed is that the ON/OFF pin of the regulator is grounded per the datasheet provided for the LM2576HV, as such this pin will not be needed and is not used. The capacitors included within the schematic are primarily used to smooth out the voltage as the regulator performs its switching at high frequency and they are also good to have because they allow for decoupling of the voltage lines to ground which also helps negate noise and even more voltage ripple that would naturally occur. For safety reasons, it is always best that a covered or shielded inductor always be used such that a user can accidently touch the bare open coils and potentially produce an electric shock to themselves or short out the components on the board or the what the board is feeding to.

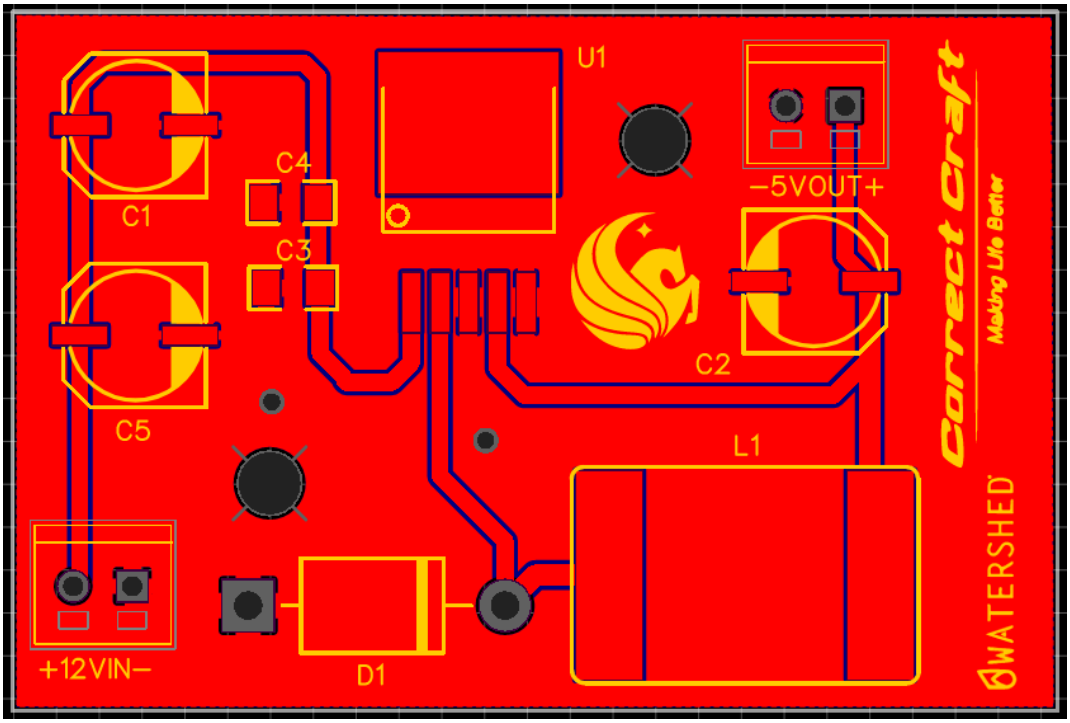


Figure 22: DC/DC Convertor PCB Layout

Following the schematics made and shown in the discussions above, the DC/DC convertor PCB design has been laid out within EasyEDA and considered a preliminary design for use within our project. As can be noticed when observing the PCB, both the top and bottom layers have been copper filled and are designated as grounds. This is a standard practice when making simple dual-layer circuit board as it is always best for all the components to have the shortest path to a ground to reduce internal resistances and capacitances caused by large ground traces as if there we're no ground planes present. Another main importance of having a ground layer on the bottom is because the pad of the LM2576HV switching regulator also can be connected to ground and it will dissipate heat throughout the board instead of having all the heat stuck inside the regulator internally. As can be noted from the above PCB layout, there are also multiple vias placed through the board such that the top ground pads have a sturdy connection to a ground plane and both planes are directly connected as well. Perhaps the most important aspect of the entire layout of the board is the fact that the diode and inductor must make their complete loops in order for current to flow properly and reliably as the regulator performs it's on and off switching application to regulate the voltage on the output. The two connectors on the top left and top right of the board are screw terminals such that the design can be fully modular and easily tested within a laboratory environment and within its actual application can be quickly disconnected for servicing or inspection. Another benefit of using quick detaching screw terminals for wire connection, is that in the event the entire design including DC/DC and microcontroller controller board needs to be moved or serviced outside of its housing that this can easily be done by any user. Along with this aspect, this allows for most free mounting of components because they are connected with any length of wire necessary.

5.1.2 Hardware for MCU Programming

The choice for the microcontroller being the ATmega1284 comes with a multitude of advantages as discussed in previous sections. Although the ATmega1284 microcontroller is part of the AVR family and can be programmed within the Arduino environment along with its simple designation of pins and baud rates, there is a specific way the ATmega1284 must be setup on a breadboard and PCB for programming. Based on the ways a standard Arduino uno and nano works, we know that the communication with computers on the small development boards in through UART (serial) communications. In order to achieve easy programming through the UART ports on the ATmega1284 microcontroller chip a USB to FTDI converter must be used tied to specific pins. In more simple terms the USB to FTDI converter will convert USB communications to RS232 UART communications via a FTS232L chip on the converter. The FTDI converter contains multiple pins including a 5V source pin, a ground pin, a DTR pin, a RXI pin, a TXO pin and a CTS pins, all of these pins and the chip itself are shown in Figure 23 below:

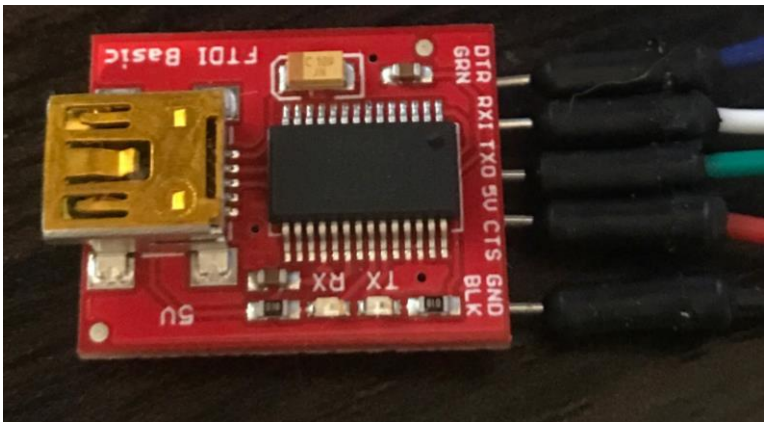


Figure 23: USB to FTDI Converter

The jump wires leading off from this FTDI converter must be placed to the proper pins on the ATmega1284 microcontroller in order for serial programming to properly work. Another few critical components to ensure that the ATmega1284 microcontroller can be programmed anywhere, is the fact that there must be a 16 or 20 MHz crystal tied between the two designated XTAL pins on the microcontroller. It is always good practice to the clock external controlled this way because most integrated hardware already has the necessary components inside the main purpose of the crystal is to be used a filter. Attached to the crystals should be two 22pF capacitors in order to reduce any extra noise produced, these capacitors should go from the crystal legs to ground on their respective planes on the board. The final piece of the puzzle is ensuring that all Vcc pins are properly delivered with the nominal voltage they require for active mode operation which is +5V and also that all ground pins have a clear and shortest path to ground. All of the pin assignments are shown in Figure 24 below which is a schematic of how the preliminary breadboard design was wired up. Also included below is Table 26 that shows what wire connects to which pin on the microcontroller layout.

Table 26: FT232L to ATmega Pin Conversion

FT232L Connection	ATmega1284 Pin Connection
DTR	Connect to pin 9 through 10k resistor
RXI	Connected to pin 15
TXO	Connected to pin 14
5V	Connected to Vcc Traces
CTS	No Connect
GND	Connected to ground plane

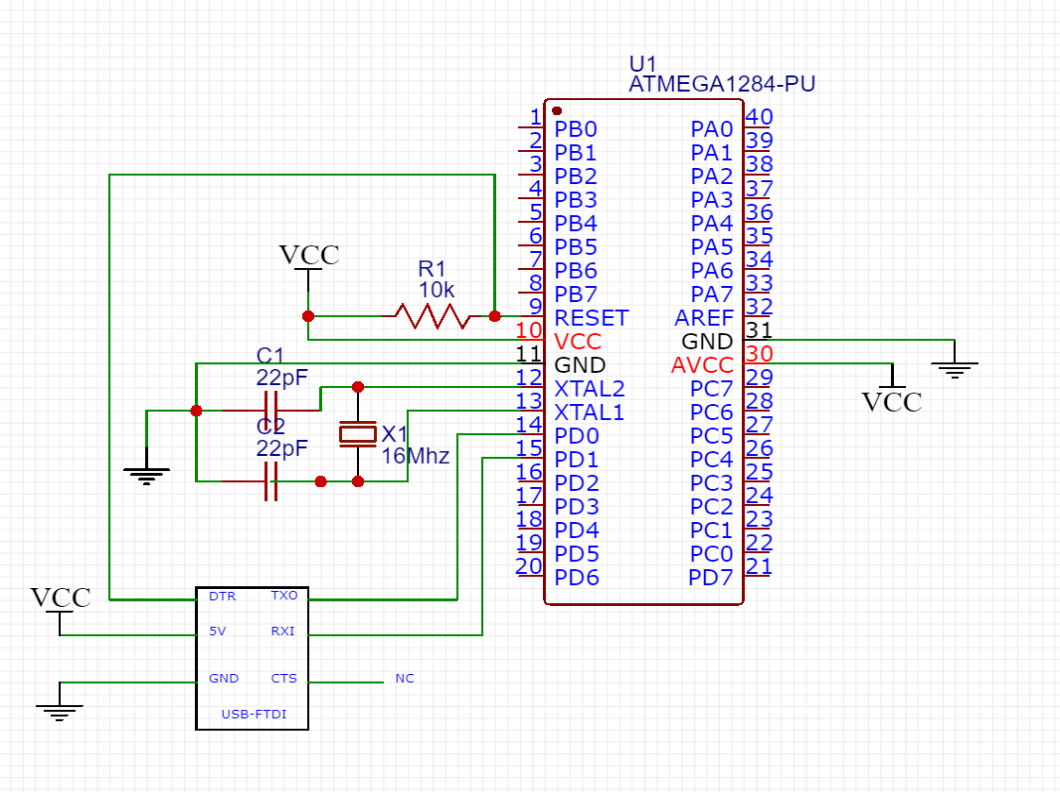


Figure 24: ATmega1284 Programmer Setup Schematic

As can be observed in the above schematic this completes the hardware side of the setup for the ATmega1284 microcontroller to be detected via USB connection to a computer and then it is possible to write programs within the Arduino environment using the pin type of

coding. This process that has just been described to achieve programmability of the microcontroller of the final PCB design is known as in circuit serial programming or ICSP. The ICSP is a critical component to our entire design because without it there would be no way to program the chip once it has been mounted on the final PCB and soldered into place and if something were to go wrong or a unique error was to occur that microcontroller would then have to be de-soldered and programmed again and once more soldered again. There is simply no need for the added stress to the components on the board nor the microcontroller of constantly soldering. Initial programming and testing of course was complete on a development board but ICSP is an important aspect of our project to on board the final PCB as in the future software updates or hotfixes can be easily uploaded to the entire system via a laptop and USB cable. ICSP also allows for serial monitoring before the entire system is even built in order to test all sensors that are reporting values back to the microcontroller via a computer, this would provide a fast and easy method for easy debugging and transmission of data to and from the microcontroller until wireless communications is setup and running.

5.1.3 Sensor Layout

The following section will outline the design and implementation of the sensors that were used during this project. Each component was chosen in order to ensure the power of the overall system remains low while also keeping in mind the accuracy of each sensor. Below in Figure 25 is the intended layout of each sensor on the boat while Table 27 defines the unnamed sensors. In the MCU diagrams below it is important to understand that the Arduino UNO board is not the board being used but instead is to show how the sensors connect to a development board.

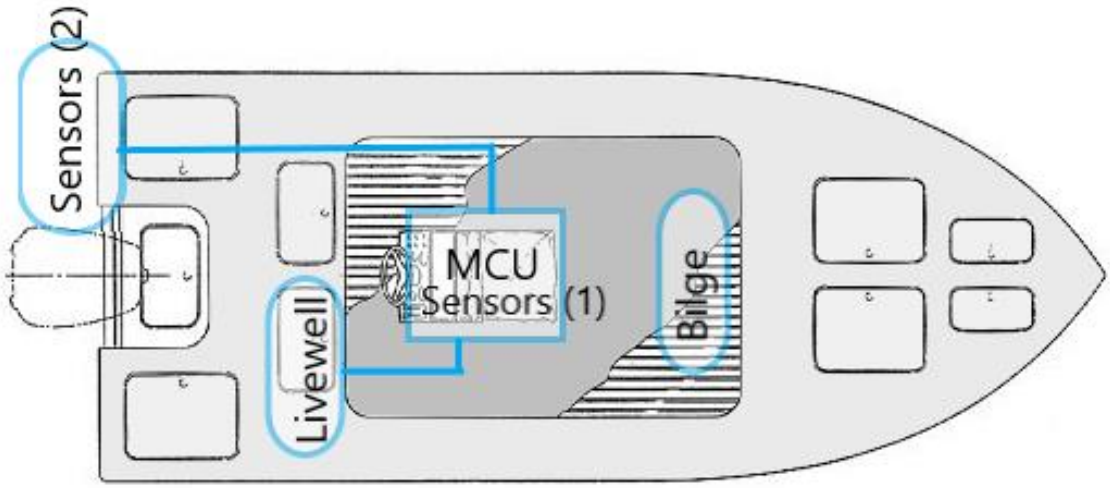


Figure 25: Sensor Layout on Boat

Table 27: Sensor Layout Relation

Sensors (1)	Sensors (2)
Accelerometer	Temperature Sensor
Pressure Sensor	N.A.

5.1.3.4 Temperature Sensor

The DS18B20 Waterproof Digital Temperature sensor was chosen for the water temperature sensor for this project. This specific sensor has a probe made of stainless steel that is waterproof, moisture-proof and anti-rust which are desirable characteristics for this project. This temperature sensor communicates to the MCU through 1-Wire communication which only requires one port in order to transmit data. When connecting to the MCU, we pinned the data wire along with the power supply and ground wires. It measures temperatures from -55 degrees Celsius to 257 degrees Celsius with ±0.5 degrees Celsius accuracy. The power supply range is from 3.0 V to 5.5 V which allows this sensor to connect straight to the MCU. This sensor transmits data digitally, which as discussed above, allows for the reading of the temperature to be more accurate. Since this sensor was placed at the back of the boat away from the MCU, it was the deciding factor on choosing the digital over analog sensor. With a longer distance of where the data is being read, the less accurate an analog signal becomes. This temperature sensor converts a 12-bit temperature digital word in a maximum of 750 milliseconds which allows for quick feedback to the user. When connecting the sensor to the MCU the data wire was connected to a digital pin, the power supply will connect to the 3.3V pin, and the ground wire will connect to the grounded pin, such as in the Figure 26 below. Before continuing, a 4.7 kΩ resistor will act as the pull up resistor. The pull up resistor is required in order to guarantee that the power supply is going to the logic circuit and is high. Finally, once the sensor is properly connected to the MCU, it is ready to communicate through code.

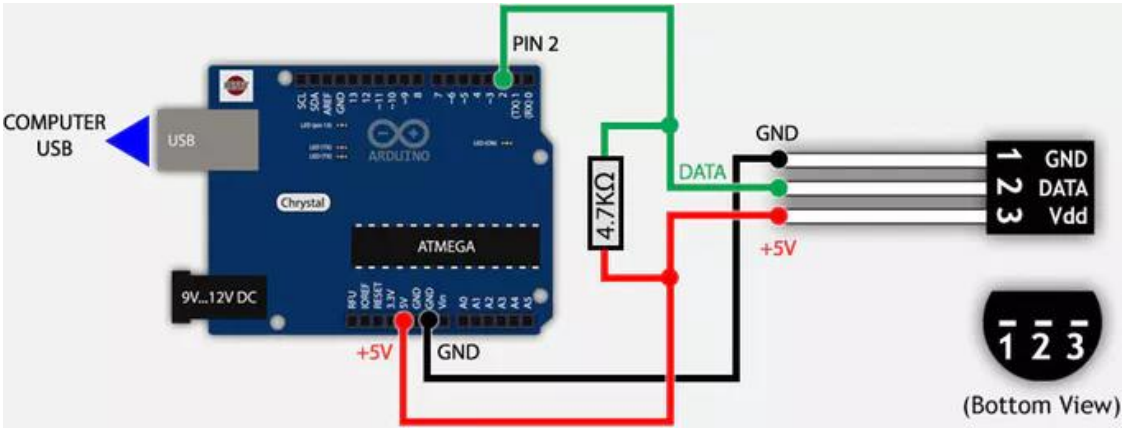


Figure 26: DS18B20 Sensor Connection with MCU

5.1.3.6 Pressure Sensor

The BMP180 Digital Barometric Pressure Sensor was chosen for the pressure sensor that was mounted onto the boat for this project. The actual pressure sensor is soldered onto a PCB that has a 3.3. V regulator and pull-up resistors. This sensor has an analog to digital converter embedded into the chip and also uses a serial I2C interface. Since the 3.3 V regulator and I2C circuit is included onto this chip, the sensor can safely be used with 5 V logic and power if needed. The sensor measures pressure in the range of 300 hPa to 1100 hPa with about 0.03 hPa resolution. Since pressure changes in altitude this sensor is also capable of measuring altitude as well as temperature. We ultimately decided to not use this particular sensor to measure temperature along with pressure because the range measured for temperature is -40 degrees Celsius to 85 degrees Celsius. This range is significantly lower than that of the DS18B20 temperature sensor and therefore less useful for this project. Although this sensor uses an analog signal to transmit the data read from the barometric pressure, this analog sensor is relatively closer to the MCU when placed on the boat and thus does not raise concerns of noise with the distance away from the MCU. This sensor was ideal for the accuracy, range and price.

As shown in the Figure 27 below, when connecting this sensor to the MCU the Vin is connected to the 5V pin and the ground will connect to the MCU's ground. The SCL pin will connect to the I2C Clock pin while the SDA pin will connect to the I2C Data pin which will allow us to use the I2C protocol for receiving the data. This sensor also includes EEPROM which stands for Electrically Erasable Programmable Read-Only Memory which is a non-volatile memory. This is important to keep in mind when receiving the data and must be handled before reading and converting the pressure that is recorded.

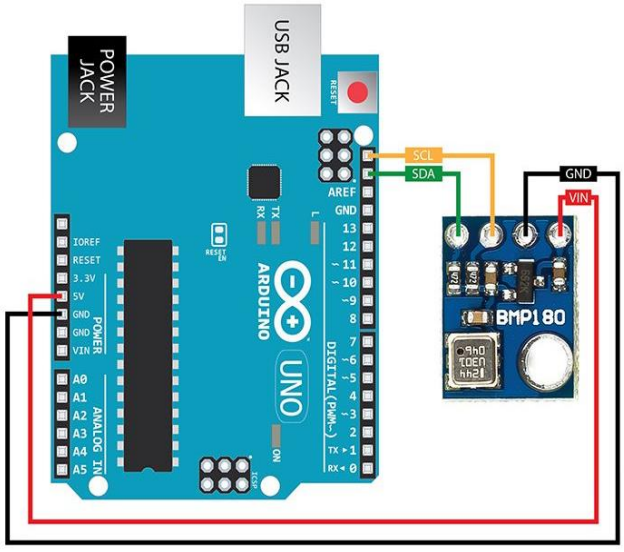


Figure 27: BMP180 Sensor Connection With MCU

5.1.3.7 Accelerometer

The MMA8451 Adafruit Triple-Axis Accelerometer Sensor was chosen for the accelerometer sensor that was mounted onto the boat for this project. The board that is being connected to the MCU is actually a PCB that contains the actual MMA8451 sensor. The extra components such as the 10k resistor is used for the purpose of a pull-up resistor. Since this sensor is a 3V sensor, a regulator was also added such that the accelerometer can be used with a 3V or 5V power and logic. The precision of this sensor was a deciding factor when decided which type of accelerometer to use. This sensor has a 14-bit ADC built into the PCB which, as discussed in previous sections, transmit a very accurate reading due to the large amount of values being read from the analog signal. A concern with analog signals discussed when researching different types was the precision of the data transmitted with regards to the signal and the ADC. Since the ADC is built directly onto the PCB that the sensor is mounted to, this relieved those concerns and we were able to confidently choose this analog sensor. The accelerometer detects tilt, motion, and basic orientation with a wide range from +-2g up to +-8g. Although this sensor will only be used for the purpose of the data recorded in the x direction, this sensor also records the other two axis', y and z.

As shown in Figure 28 below, when connecting to the MCU the Vin is connected to the 5V pin and the ground will connect to the MCU's ground. The SCL was connected to the I2C clock line while the SDA was connected to the I2C data line. Since this sensor transmits data through I2C, it allowed us to share the I2C pins and allowed for the readings to be received faster. Once the sensor was properly connected to the MCU, it was ready to communicate through code. Finally, this sensor was positioned inside of the box containing the MCU since this sensor is not recording any environmental data

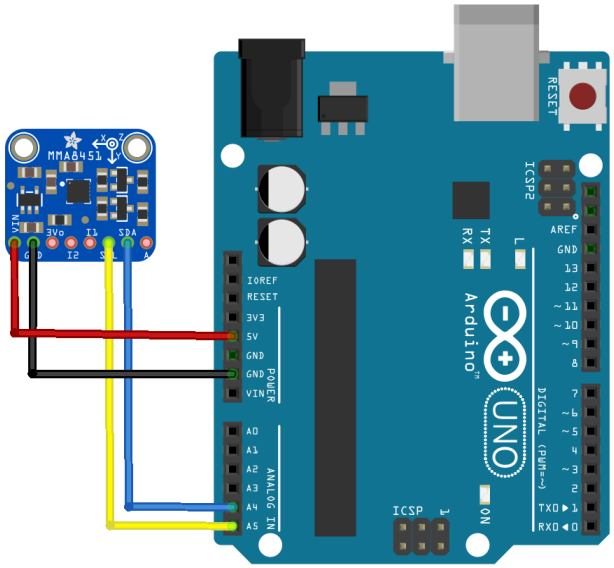


Figure 28: MMA845 Sensor Communication with MCU

5.1.3.9 Bilge Sensor

An ideal bilge sensor would be able to sense when the water level is too high which would then in return allow the user to pump the water from the boat. Although, devices like these are already offered with both the sensor as well as the switch that will pump the water. Therefore, connection with the MCU was not be necessary when using one of these devices but considering it entails its own sensor hardware, it is important to understand for the purpose of power for the overall system. The device chosen for the bilge pump was the MAXZONE Automatic Submersible Boat Bilge Water Pump. This pump has a flow rate of 1100GPH with a built-in float switch which means no separate switch is needed and would be able to automatically release the water. The feature of this bilge water pump that was most appealing when choosing a pump was the use of no power until the pump turns on from the water rising. Although the power when used requires 12V with 3.8A, since the power is being regulated the power drained from this pump remains relatively low.

In order for the bilge water pump to work, this device has a built-in float switch which is a level sensor used to the detect the level of liquid or water in a tank or container. The float switch uses a mercury switch which is able to open and close a circuit as a certain amount of liquid metal mercury connects in order to close the circuit. When the float switch is able to detect a rising in water within in the container, the electrical pump begins to pump the water out until the level of the water has been reduced back to the point of which the switch is off again. Once the switch has detected in overflow in water, the water will then be pumped out of the outlet with a 19mm diameter. Overall, the integrated bilge water sensor inside of the bilge water pump performed the necessary duties in order to keep the water level low.

Table 28: Overview of chosen sensors

Sensor	Part Number/Name	Cost	Connection	Sensor Output
Temperature Sensor	DS18B20	$7.99	1-Wire	Digital
Pressure Sensor	BMP180 Digital Barometric Pressure Sensor	$8.99	I2C	Digital
Accelerometer	MMA8451 Adafruit	$9.89	I2C	Analog
Bilge Sensor	MAXZONE Boat Bilge	$79.99	No connection	N/A

5.1.4 Wiring

Of great importance with any marine vessel design is the choice with regards to wiring and proper routing of wires throughout the hull or cabin of the boat. Well outlined in the standards section of this document, specifically the standard 183.425 and overcurrent protection for conductors, is the need for wire utilizing a gauge of no less than AWG 16. Inherent resistance for this gauge and above is of note as minor drops in voltage is possible when interfacing devices such as the switchboard and lighting system, to be discussed in sections 6.2.5 and 6.2.6 respectively. However, with the comparatively confined space in which the wires are routed, a 17-foot boat, the resistance associated with the necessary wires did not weigh heavily on the overall design. Of primary consideration was the proper insulation to be used for wire sheathing as the conductors are routed throughout the boat as ensuring the safety of the user and functionality of the electronics is of top priority. The utilization of tin coated wires along with vinyl insulation in our design met these demands and was the best option with regards to withstanding the harsh marine conditions endured. In addition to these safeguards is the Type 3 stranded design of the wire to be used which reduced physical stress placed on the wire and reduced the risk of strain related issues.

5.1.5 Switchboard

The switchboard for this vessel controlled the switches for the lights, the pump for the bilge, the pump for the Livewell and the aerator for the Livewell. The switchboard is connected to the 12 V step down voltage connected to the boat's battery. This provides the sufficient amount of power towards the light, bilge pump, Livewell pump and Livewell aerator. The switchboard is intended for the user of the vessel to have control over the available switches by manually turning on the lights, pumps or aerator. With regards to the switches for the lights, the vessel must follow the Coast Guard lighting standards and thus will require control over the lights for when the boat is in use or docked. With regards to the Livewell pump, this is used when the Livewell is needed to fill in order to keep the fish on the boat. With regards to the Livewell aerator, this is necessary for when the oxygen sensor communicates a warning level of oxygen within the Livewell. Since there is no automatic aerator, it is the responsibility of the user of the boat to use the switchboard in order to relieve the fish stored in the Livewell.

The switchboard was mounted in the center console of the vessel to allow the user to have easy accessibility to the switches.. In order to install this into the boat there was a need to be a designated slot for this switchboard which we then sealed with silicon and proper mounting hardware in order to ensure safety of the circuitry by making it waterproof. Once the switchboard had been installed, the wiring of the of the switchboard to the 12V step-down voltage which allowed the switches to be powered was performed. The switches will also be wired to the functioning piece being switched on or off within the vessel. Each of the light switches was wired to the corresponding lights on the different areas of the boat while the Livewell pump and aerator connect to the Livewell through the wiring on the boat.

5.1.6 Lighting

The lighting which was mounted onto the boat included all of the Coast Guarded standard lighting. This lighting was connected through the switch board which was powered by the 12 V step-down voltage connected to the battery of the boat. The lighting consisted of the Stern light and Bow lights. The stern light was mounted on top of the Poling Platform which is located at the back of the boat. The stern light is a white light serves the purpose of being able to the light regardless of whatever angle the boat is being viewed at since it sits higher than the boat and is aft. The side lights were mounted on the side of the boat and vary in color depending on which side. The side lights allow others on the water to know which way the boat is coming or leaving towards and is a way to communicate to others on the water. If the light is red, then the boat is being viewed at its left side while if the light is green then the boat is being viewed on its right side. The cabin light is located under the gunnels or on the sides of the interior of the deck. These lights are white and face downward to project onto the floor of the vessel in order to allow the user of the boat to operate during sundown.

5.1.7 PCB Design

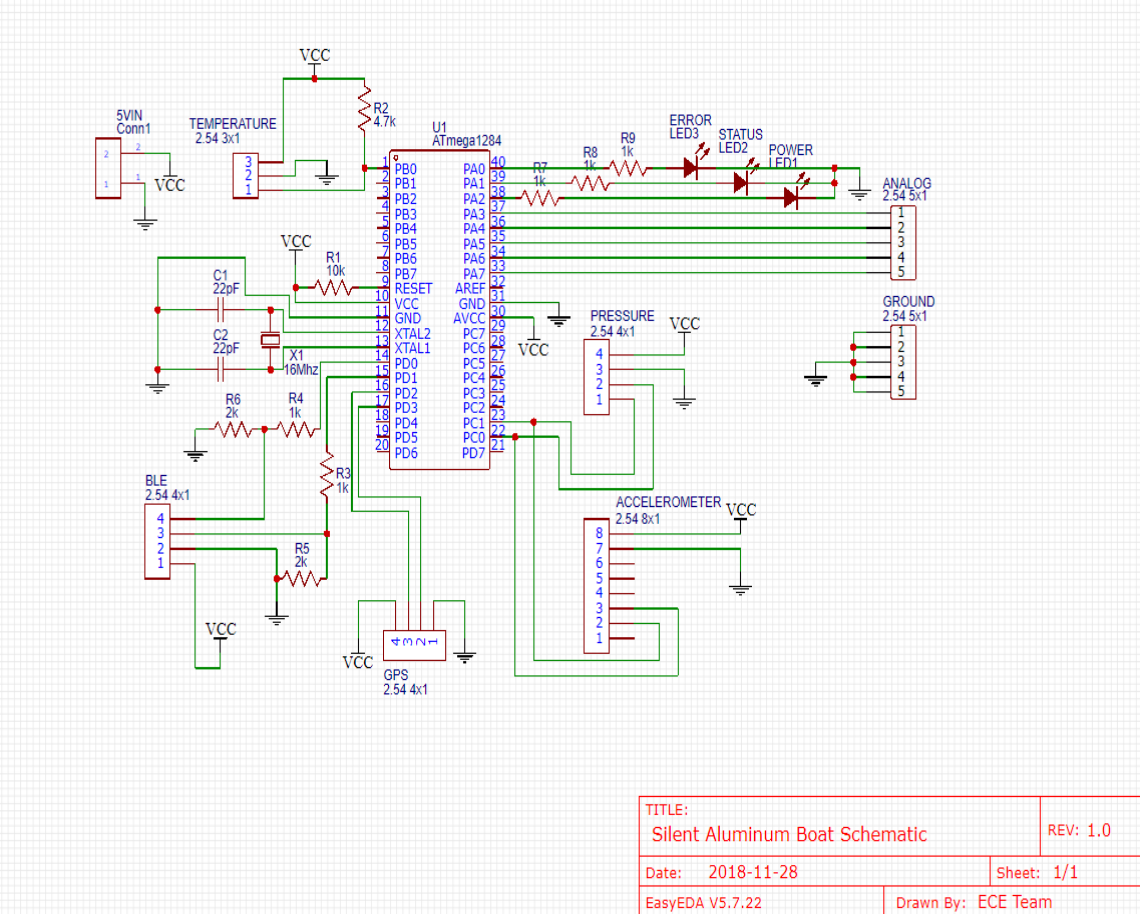


Figure 29: Overall Schematic of Microcontroller

In Figure 29 above of the schematic shows all the hardware connections that were made in order for the ATmega1284 microcontroller to be interfaced with all the different communications and sensors modules used within this project. All female headers made were placed within the orientation of which the respective components are connected, this orientation must be strictly adhered to because the sensors cannot just be connected to the headers with any orientation. As can be noted, the sensors are not receiving their 5 volts or 3.3 volts of power from the chip itself in order to prevent the microcontroller from delivering too much current when all sensors operate at once and get hot. In order to remedy these issues, the sensors receive their voltage directly from the VCC flag which is set to the input of the 5-volt connector that receives its power from the DC/DC converter discussed in the previous section of this report. As can be noted, there are multiple LEDs on the top right of the schematic, these denoted as 'status' LEDs that will allow for easy troubleshooting of the final PCB because these LEDs can be programming to trigger when different issues occur, in the current design one LED is used as a power on light, another is used for a status or transmission light and the final one is used for an error light these of course all have their current limiting resistors attached to them. The schematic also includes the external components needed for the ATmega1284 itself to run on a bare board, which are the resistor to the reset pin via VCC, and the crystal and two 22pF capacitor across the XTAL pins. Along with these components the VCC must be connected to the VCC and AVCC pins of the microcontroller and all ground flags are obviously denoted with a ground flag. Aside from many of the different header types that are used for specific components and modules, there is also a 5x1 header that has been attached to the many analog to digital conversion pins on the ATmega1284 for future use within the project, this is also paired with another header of the same length to include five extra ground connections such that the only grounds available won't be component specific and extra external modules can easily be interfaced and tested as the project moves along. The only module that actually requires 3.3-volt logic is the BLE device that we are going to use, it uses 3.3 volts for logic from the RX and TX pins respectively and thus as shown within the schematic a simple voltage divider from those two pins must be used. The pins occupied by the GPS header locations are tied to the virtual UART pins which are RX1 and TX1, these must be enabled within the software design of the project and properly setup in order to take data in from the GPS which will report coordinates and other data at specific intervals dictating within the programming. Of course as can be noted, there are a few pins of the right side of the chip that are available for use, these are utilized if it is determined by our sponsors if specific NMEA sensors must be used within the project and if we received access to being able to program these sensors within the software itself. Overall, the schematic was shown to be fully modular such that when service or system updates must be performed to the final product, all of the components can be disconnected such that the main microcontroller can be programmed and serviced or replaced if needed. The following table shows all the components denoted within the schematic and their type that are used in the final PCB layout.

Table 29: Overall schematic external components

Schematic Part Number	Value	Part Number or Size	Type
R1	10k Ω	1210 1/3W	Carbon Composite
R2	4.7k Ω	1210 1/3W	Carbon Composite
R5, R6	2k Ω	1210 1/3W	Carbon Composite
R3, R4, R7, R8, R9	1k Ω	1210 1/3W	Carbon Composite
X1	16 MHz	HC-49S	Quartz Crystal
C1, C2	22pF	1206	Ceramic
Female Headers	Various lengths	2.54mm	Connector on pin

Following the schematic and components tables shown above, the following PCB design layout was produced within the EasyEDA environment and this design is also considered preliminary as updates to the components of the board or the layout itself may need to be updated depending on events that will happen during senior design 2. The more intensive breadboard testing of components within the lab and the results that are produced over winter break will solidify the best ways to interface components and if any changes need to be made to this design as a whole, but overall the team is satisfied with this design thus far. Also, for now most of the traces have all been made a decent size to handle the currents that they are exposed to within the PCB layout, although more research must be performed in order to determine the optimal width of traces that must be made. Additionally, as per common design when it comes to PCBS, the traces have no harsh 90-degree angles when connecting across the board, all of the traces have been made with the typical 45-degree angle when needing to change direction. This is always standard practice because making traces have harsh angles and direction changes are usually never good for the final design of the PCB. Finally, all female and male headers as denoted above are the common 2.54 millimeter size that appear on almost all development boards that exist in the market and can easily interface all the communications modules and sensor devices that are used during our project and these will basically act as shield locations.

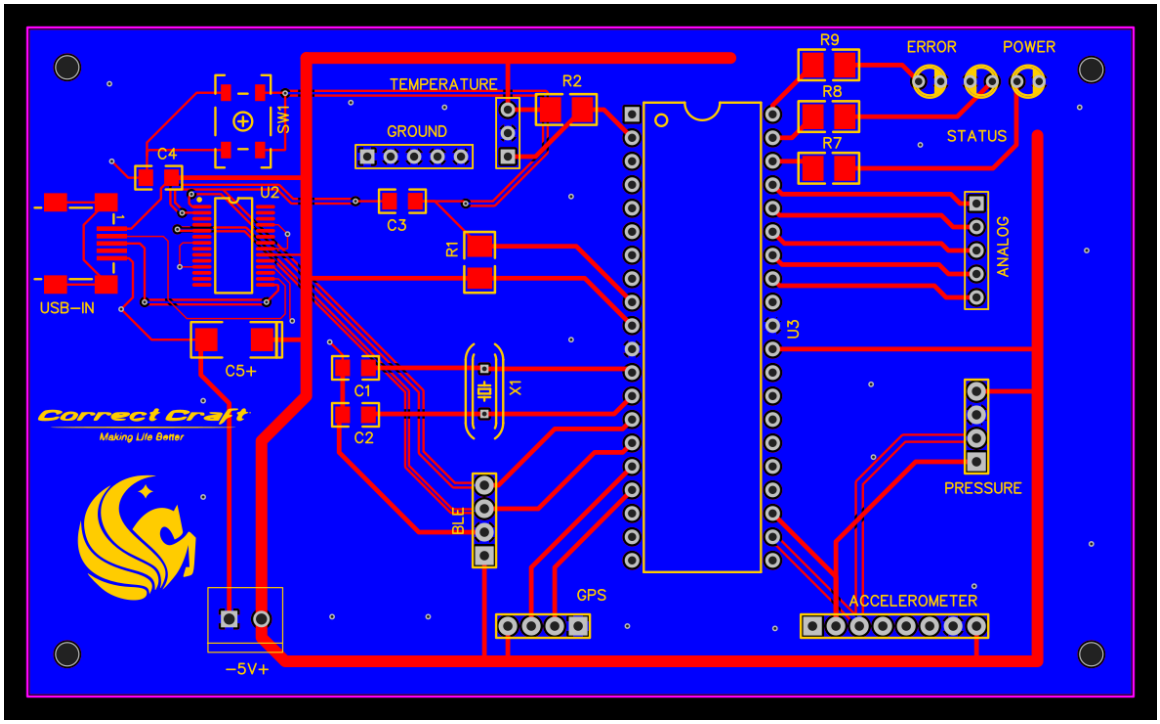


Figure 30: Microcontroller Board PCB Layout

Once more describing the above Figure 30 which is the microcontroller PCB layout, as can be denoted and as previously discussed, both layers are designated as ground especially in this case so all the components that are a decent distance away have an almost immediate path to ground. Also observed within the PCB layout are the small vias that are placed around the board in strategic areas that are close to header ground pins, this ensures that both layers are electrically connected on the board as ground. The ATmega1284 microcontroller chip itself is placed at the center of the PCB for obvious reasons because all traces will have to run from the chip to all headers and external components and it looks better because the microcontroller will appear as the 'heart' of our board which it is. The capacitors are also closely connected to ground together along with their trace to the 16 MHz crystal such that the mutual inductance created by trace length and distance is kept as low as possible. Fortunately, the same screw terminal connectors discussed in previous sections will also once more be used again here to provide a 5-volt source to all VCC pins denoted in the schematic as needed. The trace leading from this terminal was increased in width from the default width such that it can accommodate the current draw by each component and the current will not 'sink' into areas that it is not supposed to go nor will it overload the trace if there is a load spike which could kill the entire board. The next placement aspect that can be observed is that the headers are all out of the way of each other such that each component has the proper space needed to be interfaced and when the final design is to be built everything can easily be troubleshooted if need be and the board isn't cluttered. As it stands the dimensions of the board itself is about 4.5 inches by 3.2 inches which is quite small for all the components that are interfaced to the board itself. The final aspect that should be observed from the PCB layout are the mounting holes that

are equally placed in all four corners such that when the final design is placed within its respective component secure box, it can be securely fastened into place.

In terms of the project it was important to know which company are manufacturing the PCB and then sending it back to the team for final soldering and creation of the design. Fortunately, EasyEDA provided in-house PCB manufacturing at decent rates overseas and gladly generated the Gerber file from the PCB file, checked and reviewed it for design issues and then built the board and sent it back to the team.

Table 30: PCB manufacturing cost estimates

PCB Name	Manufacturer	Amount of boards	Lead Time	Cost
DC/DC Converter	JLC PCB	5	7-15 business days	$17.26
Microcontroller Board	JLC PCB	5	7-15 business days	$8.00

All of the boards that were cost estimated in the table above, are constrained to the parameters in the following table below. The parameters that are about to be listed dictate the cost and complexity of the boards will we be ordering; these parameters were chosen to best suit our needs and to have safety as priority when it comes to working with the boards in a soldering environment depending on the nature of the materials used to manufacture the board.

Table 31: PCB manufacturing parameters

PCB Thickness (mm)	PCB Color	Surface Finish	Copper Weight (ounce)
1.6 mm	Standard Green	Lead Free HASL-RoHS	1 oz

The EasyEDA PCB design tool also includes a DRC tool and a 3D previewer of what the design will look like after manufacturing. This is a good way of double checking our work on designing and laying out the board, by running the DRC (design rule checking) tool it will automatically check for possible errors in placement of components or errors in traces that have been made to and from all areas with the PCB. After using this DRC tool with our designs above it was stated that no errors exist and this preliminary check is complete. After both of these tools have been used to double and triple check our work, the team decided it was satisfied with all results and designs within this section and this accomplishes a large task that is usually a part of senior design 2, and now that the bulk of the work has been done only small changes would need to be made as time goes on instead of making an entire design from scratch all over again. With these goals accomplished, the team could focus on other areas such as the software development and mobile data delivery sides of the project because those are equally as important.

5.2 Software Design

The design of our software subsystem was efficient, versatile, and stable. Our software accounted for all foreseeable situations, and mitigated any other failures. It was able to leverage our hardware to its full potential, while also was able to communicate all of its data to the app and then the database. The software on the MCU is at the heart of quite a large system, and had to manage a balancing act to keep it all running under all the possible circumstances. Due to the real-world implications of a failure of our project, it was crucial that our software system was reliable and stable. This was to ensure the quality of the user experience and overall guarantee a degree of safety in a potentially dangerous environment.

5.2.1 Software Flowchart

In order to development software efficiently, you should always think about the basic flow of the system that you are developing. This will help you determine what states you will need to create and how they should connect to each other. This will also help you to identify states that you may end up in unintentionally. Planning this ahead of time helped alleviate some of the issues that may arise while also showing the Computer Science team what they can expect from our end. Our initial software flowchart is shown below in Figure 31.

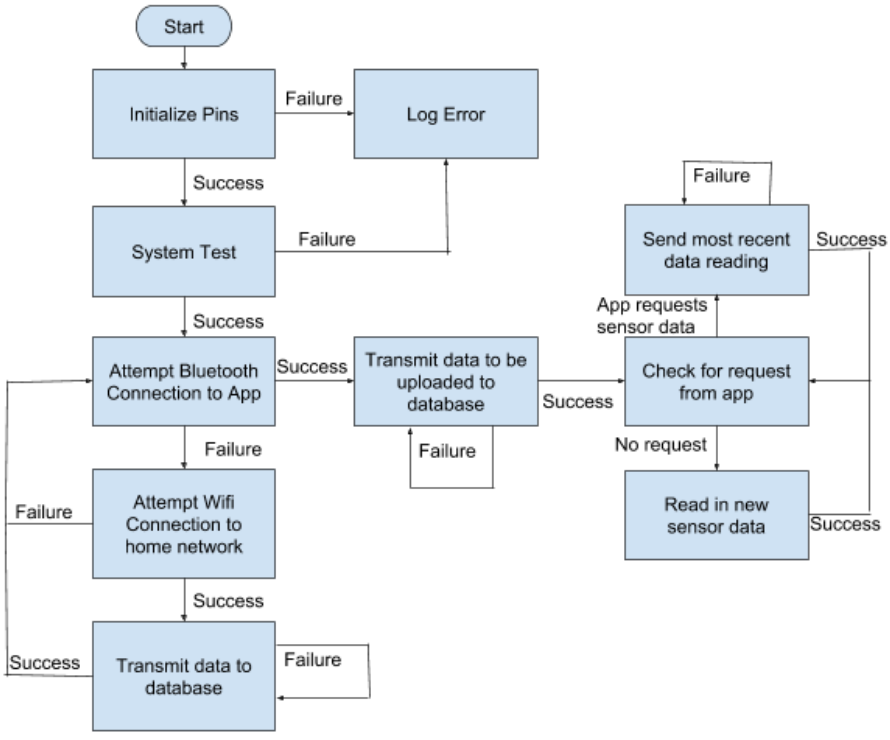


Figure 31: Basic Software Flowchart for the MCU

5.3 Network Design

Having a well-made network topography is a key part of any project involving wireless networking. Having a proper topography makes the difference between clean network traffic and a mess of missed requests and duplicate messages. Inefficient networks can lead to using extra processing power on our MCU, thereby slowing down our entire system and potentially leading to CPU issues if we are too sloppy. Having a clear understanding of what protocols, we are using and how each one of them behave will lead us to an efficient system. Having a good network design is also critical to ensuring we can have good interfaces with the Computer Science subsystem, since the main interface with it is through BLE. As integration becomes more apparent with the progression of this project, the importance of network design were refined to further aid both teams. Our high-level network topography is shown below in Figure 32.

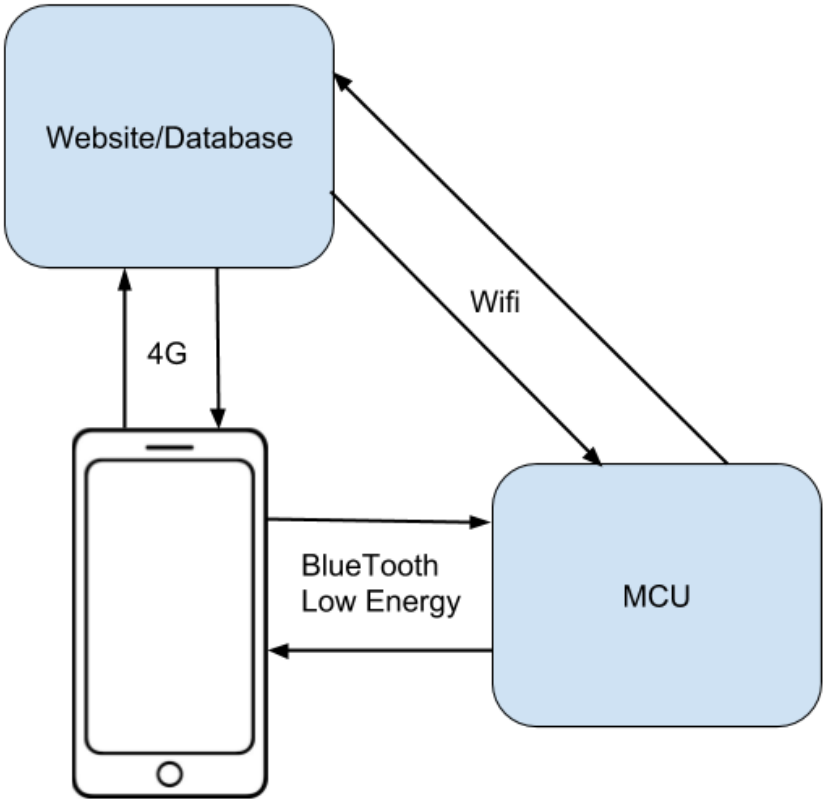


Figure 32: High-Level Network Topography

5.3.1 NMEA0183

With the marine applications of this project, the NMEA protocol holds strong influence with regards to many of the sensors available on the market. Specifically of interest for many of the transducers and sensors is the NMEA 0183 protocol. Similar to NMEA 2000, it is a serial data bus standard for communication with a variety of sentences used to transmit data. For many of the transducers used in the boating industry, this is the preferred

method of data communication. However, as direct compatibility with the ATmega microcontroller unit is an issue, there is an inherent need to design a hardware solution for data conversion. This solution comes in the form of an RS-232 or RS-422 to TTL converter for data sent from the NMEA 0183 device to be properly read by the microcontroller. The use of the respective RS-xxx converter is dependent on the device to be used and available data lines for transmission.

Devices utilizing only one wire or conductor for data transmission for example will utilize the RS-232 protocol in order to properly interface with the ATmega microcontroller. For this case, the NMEA 0183 device will utilize a single transmit pin connected to the converter before the data is received on the Rx pin by the microcontroller for proper integration with the necessary software. While useful for low power transducers, the components utilizing this method for data transmission are limited. Typically, of favor are the NMEA 0183 devices utilizing multiple lines or wires for data transmission.

These devices which utilize multiple lines for serial communication require a different method for translation; the RS-422 or RS-485 to TTL converter. With multiple lines for serial data transmission, the RS-485 is the primary choice as there are a larger number of marine electronics devices which may use this protocol for conversion. With the added wires for data transmission is the added level of complexity when wiring the converter to our microcontroller. For general wiring purposes, the Vcc pin of the RS module would be routed to the corresponding 5V or output voltage pin of the Microcontroller/PCB. The data pins of the shifter are connected to the respective data lines of the sensor being utilized, typically labeled NMEA+ or NMEA-. For interfacing the ATmega microcontroller, only one receive is needed as the necessary data can be transmitted through this route.

Of note is the voltage difference when comparing the NMEA sensors or transducers to that seen/utilized by the microcontroller. As per the NMEA standard, 12V is utilized to power these electronics typically on a backbone for the serial bus. However, the voltage utilized by the microcontroller is approximately 5V. In order to prevent damage when interfacing these two, it is imperative to implement a regulator or utilize an RSxxx converter capable of reducing the input signal from the NMEA device to the necessary 5V.

5.3.2 BLE

In order to communicate to the smartphone from our MCU, we utilized a Bluetooth Low Energy (BLE) connection. This connection was responsible for communicating all information to the mobile phone that the user are interfacing with. This includes a lot of data that was uploaded to the database, as well as information that would only be given to the user at the time. Data such as current heading and velocity was sent to the phone to be shown to the user directly, while other items of less importance were just uploaded to the database. The data needed by the app was different depending on what the user is doing at the time, and the MCU served as a peripheral node to the mobile phone.

In order to set up our BLE connection, we had to set up a GAP profile. A GAP profile will determine what connection parameters are used in BLE. This will then affect things such

as data rate and power consumption. Since our MCU is a peripheral node, it was not able to set the connection parameters, but it was able to request them be changed to what it desires. Since our team is managing the BLE connection, we simply requested that the mobile phone follow the parameters we set. We used a low connection interval, since the additional bandwidth was helpful, and the MCU was not consuming a significant amount of power compared to the motor even without power saving modes. This also helped to reduce latency in our user interface's data display. In our code, we had to set these GAP parameters in order to have the connection initialized the way we desire.

Once the BLE connection was established, we utilized custom GATT services in order to communicate the data we wanted between the MCU and the mobile phone. The MCU behaved as a GATT server, and set all the GATT services that it wished to have. The mobile phone was then able to choose which services it wanted to use, and the MCU responded with the requested data. We had services set up for various sets of data that the mobile app was trying to collect at one time, consisting of characteristics that it's asking for. For example, if we had a basic gauge cluster screen, it may only display heading, velocity, and battery life, so we would have a service made for this gauge cluster that included all the data from the corresponding sensors. We also made a service for each screen in the app, as well as any overhead ones that we deemed fit. This allowed us to only send over the information that we need to the mobile phone, and not waste power and computations on sending data that is not needed.

In the user interface of the mobile application, we had to include a section dedicated to setting up the BLE connection. This was so that we were able to set up a password on connection so that it remained secure. We set an initial password, and then allowed the users to change it once they had registered their boat. This screen also included the area used to connect the boat to the user's home Wifi network. Lastly, it's important that we made a "limp" mode for the boat that makes it consume less power, and BLE played into that by modifying the connection interval to conserve more power for the MCU. This allowed users to conserve power in emergency situations, so that the motor may last as long as possible.

5.3.3 Wifi vs 3G/4G

The discussion of the usage of Wifi or 4G was one of large amounts of discussion on our project. Both these modules would allow our MCU to connect to the internet, but in very different ways with very different effects. The goal of these units was to provide the MCU with the ability to upload data from the boat to the database storing the information related to the boat without having to be connected to a smartphone over Bluetooth. This helps to ensure that any data on the boat can be always be uploaded, regardless of the circumstances. This also opens up new functionalities that we can pursue, since the boat can be connected to the app and website remotely over the internet. This could allow owners of the boat to check that their boats are charged and ready to go remotely. This would allow someone to check their boat's condition from work before they go home for an evening of fishing.

A 3G or 4G device on the boat is an ideal goal for our design, as it provides a consistent, reliable internet connection under all circumstances. This would mean that no matter where the boat is, we can always connect to it to monitor it. Some potential use cases for this are a parent wanting to monitor their child if they are using the boat, or wanting to remotely start it before you head outside to climb aboard. While Wifi would allow you to interact with the boat remotely sometimes, it requires the boat be left close to a Wifi router. Another advantage of using a 4G module would be that if anything happened to your phone while on your boat, it would still have a connection. This is good for emergency situations as well as keeping the system reliable.

There are still many reasons to use a Wifi module instead of a 3G or 4G one. Most importantly, we have struggled quite a bit to find any libraries for the ATMega that support 4G modules. We have however, found one or two for 3G modules, but there is a wide abundance of support for Wifi modules. This makes Wifi the most straightforward solution, so that we do not spend a lot of time reinventing software. In regards to the need to be connected to a Wifi network, the type of fishing boat that we are making is not normally stored at a dock due to its small size, and would therefore normally be stored at the owner's house, within Wifi range. Due to this, we do not expect it to be much of an issue. We also do not foresee any issues with being on the water and not having the 4G from the phone, due to the fact that we have a phone charger on the boat itself. Additionally, even if the phone were to be damaged, this situation is less likely to happen than a loss of cellular signal. Since the 4G module and the phone would both rely on this, they would both be left without a connection to the database and website. Since that failure is so much more likely, we don't see it necessary to commit a lot of time to solving such a small issue. In Table 32, we will compare the advantages of a Wifi module vs a 3G module.

Table 32: Comparison between use of a Wifi module and a 3G module

3G/4G Module	Wifi Module
Boat can be parked anywhere	Boat must be parked within range of router
Can provide internet connection while out on the water	Leaves the boat relying on the mobile phone while on the water
Minimal online libraries that are compatible with our MCU	Several libraries with support for our MCU
Modules typically in the $70 - $100 range	Modules costing less than $40 typically

After evaluating both of these modules, we determined that it would be in our best interest to simply use a BLE module instead as it provided the capabilities to push all data to our smartphone app.

6.0 Testing and Prototyping

Having good quality assurance is a huge improvement on any project. Being able to refine the product to its best state while also increasing its robustness will lead to a happier customer as well as having a better product under the developer's belt. Like many others before us, we have built numerous prototypes to be able to test all our components as we receive them. As we validated our components, we combined them into bigger and bigger subsystems, performing more complex tests as we went. This allowed us to efficiently find issues in our design and know with pinpoint accuracy where they stem from. It's essential to test your system iteratively, as this allows you to much more easily identify what is causing the issue, so you do not waste time retesting other components over and over again. Through our testing, we solved many issues that existed in our original design and have learned a lot about exactly how our system functions.

6.1 Microcontroller Testing

Once the initial setup of the microcontroller was completed on a breadboard based upon the previous hardware sections and the voltage sources have been verified to deliver the necessary power to the microcontroller, we could begin initial testing of the microcontroller for use in terms of programming. Then a USB type B cable was to be plugged into the female side of the FTDI chip in order to begin testing. Opening the Arduino IDE environment, one should search for the proper board in the tools option and select the 'ATmega1284' microcontroller at a frequency of 16 MHz The option for the programmer that should be selected is 'Arduino as ISP', the reason this is selected as the programmer option and not the default AVRISP option is because given the fact the ATmega1284 chip has been bootloader with the proper software to run on the Arduino IDE the way the programmer speaks to the microcontroller chip as if it was being flashed through a standalone Uno development board. All the connections either via a breadboard or development board should be made according to the above section that details the hardware setup and these guidelines should be strictly followed. In order to test the functionality of the microcontroller and ensure that it is programmable, one should select the 'bare minimum' code and choose the 'program' icon inside that Arduino IDE. Once this happens the console should output that it is communicated with the microcontroller and the RX/TX lights should begin flashing rapidly as serial communication and data begins to transmit. Note that this is not necessarily the case for all components, as some will only utilize the transmit LED as data will not always be received directly from the microcontroller. Once this process of flashing the bare minimum code is complete the console should output a 'success' message and the microcontroller should be flash with this code. This code should remain imprinted within the microcontroller even during power cycles as per the large size of flash memory and the inherent way in which it works. Once this process is fully complete and the Arduino and microcontroller successfully communicate with each other, this meant the microcontroller was ready to be programmed with any code that can be compiled within the IDE itself and ensures that the hardware design is working as well. Implementation was the next step, as components were now capable of being combined for final integration with the device as a whole.

6.2 Power Testing

The testing portion of power distribution was a crucial step to the design as the stability of this system plays an integral role the voltage sensitive parts. The switching regulator of primary interest with regards to ensuring the safety of the microcontroller and various sensors is the LM2576HV. As discussed in the research portion of this document, this voltage regulator was utilized to step down to 5V from the 12V Cllena regulator. The necessary components for this regulator to function are seen previously in the design regarding DC/DC conversion. Below is the corresponding breadboard image for the regulator in Figure 33. Following this is Table 33 which contains a range of the output voltages based on the range of inputs.

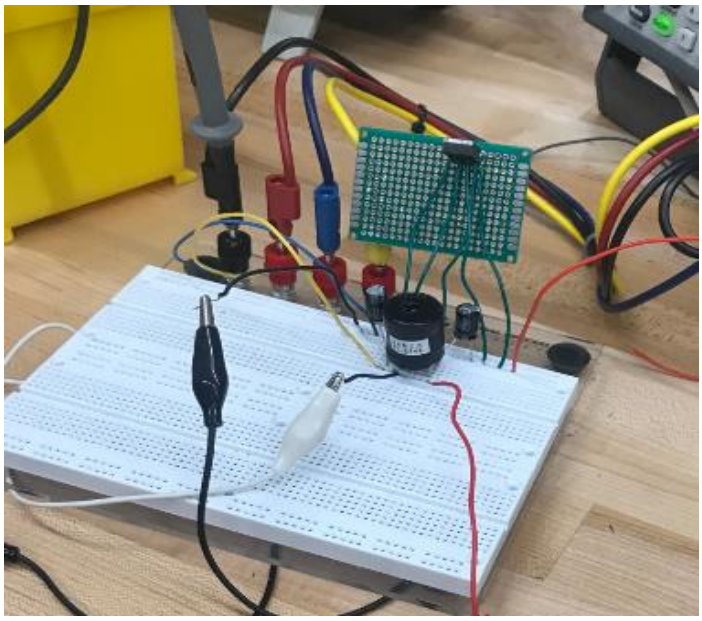


Figure 33: Breadboard Test of LM2576HV

Table 33: LM2576 Regulated Voltage

Input Voltage	Output Voltage
8V	4.96V
9V	4.99V
10V	4.99V
11V	5V
12V	5V
13V	5.1V

From the table and testing, it was determined that the voltage regulation with the LM2576HV was quite steady and more than suitable for our purposes. As it would be

stepping down from the 12V transformer, testing was concluded after 13V to ensure there was some degree of error available.

6.3 Component Testing

Component testing was underway with the various sensors, pumps, and lighting to be implemented on the boat in the initial phases of testing. It is worth noting that these devices were tested at the appropriate voltages which were to be expected after each step down. Accordingly, the regulators saw heavy use with the testing of each component and their reliability was also further judged. Detailed descriptions and the process used to confirm the functionality of these respective devices is discussed in the following sections. A picture of all currently available and tested sensors is seen previously in the report. This is due to financial constraints in addition the desire to wait for a finished CAD model of the boat to ensure the parts purchased are compatible and useful.

6.3.1 MMA845 – Accelerometer Testing

The MMA845 was tested for accuracy with regards to functionality and implementation onto our chosen microcontroller unit and development board. The following photograph illustrates the implementation onto the dev board and connection made via a USB for proper programming to be done on a PC. The device was determined to be functional and the corresponding test plan used for functionality confirmation is discussed following the breadboard image.

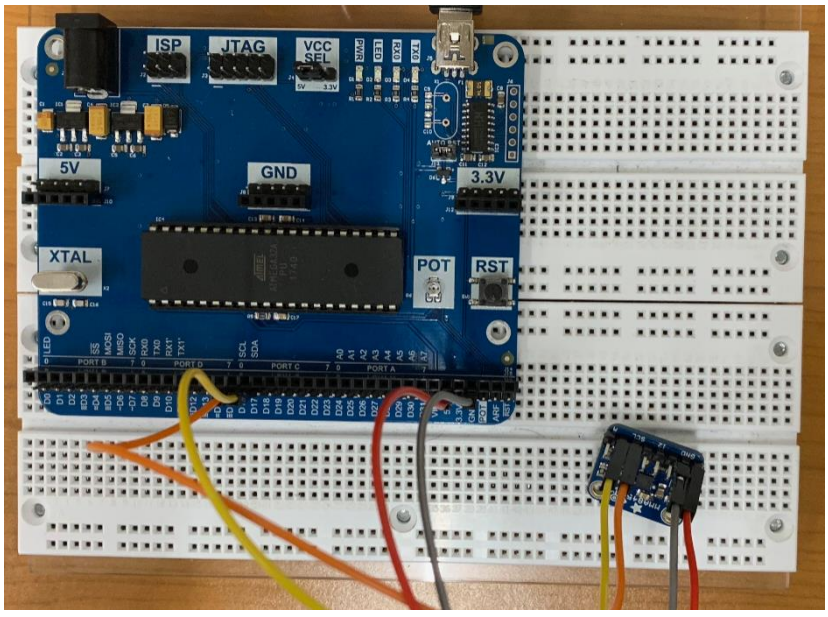


Figure 34: Accelerometer Testing

Table 34: MMA845 to MCU Pin Conversion

MMA845 Connection	Microcontroller Pin Connection
SDA	Connect to pin 16 (D17)
SCL	Connected to pin 17 (D16)
Vin (5V)	Connected to pin 33 (5V)
GND	Connected to Pin 35 (GND)

The first step to testing this component would be proper implementation of the respective libraries and sketches into our chosen IDE. Code would be utilized to output the data available with this sensor; particularly acceleration and relative location of the device. Regarding the testing plan for component functionality and verification, the device would be rotated and moved along the x, y, and z-planes to determine if accurate information was being transmitted to our microcontroller unit. The output on the serial information tab in our IDE would confirm proper integration with the MCU with this device with information being output that was reasonable with our assumptions.

This unit was found to bear full functionality and produced reliable results when rotating and physically manipulating the device to the degree necessary in the planes of orientation. It is worth noting the importance of finding the zero point or setting the initial zero point of this component when analyzing the acceleration in the x-axis which is the primary application of this device. To ensure accurate speeds and accelerations are read when implemented onto the final product, the boat, the device needs proper orientation with respect to how the device reads and transmits the data. Regardless, initial testing of this device proved its functionality and confirmed the reliability of our purchase.

6.3.2 BMP180 – Pressure Testing

The BMP180 Pressure sensor was tested for accuracy with regards to functionality and implementation onto our chosen microcontroller unit and development board. The sensor was tested based off of the base pressure that was declared in the code and was then determined to be relatively accurate to our knowledge. The following photograph illustrates the implementation onto the development board and connections made via a USB for proper programming to be done on a PC. After testing, the device was determined to be functional and the corresponding test plan used for functionality confirmation is discussed following the breadboard image.

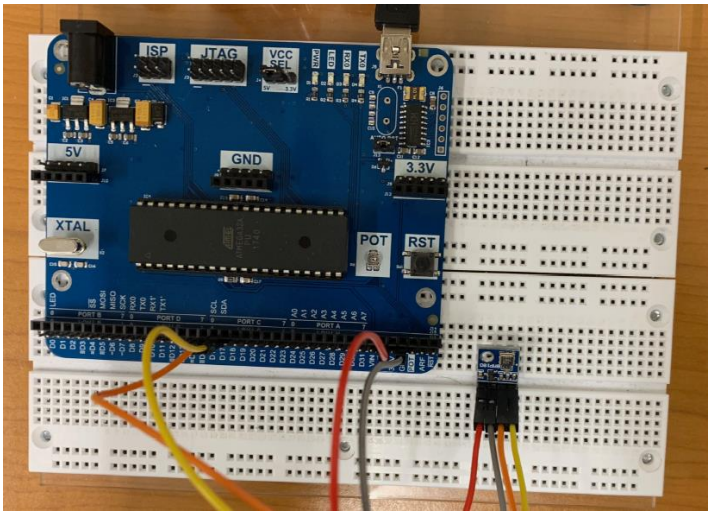


Figure 35: Pressure Sensor Testing

To begin testing this component, it was first required to download and install the necessary Arduino libraries and sketches that will assist us in testing and executing this sensor. To test the sensor, the code must be written to be able to read in the pressure with a given baseline pressure. Since pressure varies with weather, the pressure must first be read at a baseline altitude before the relative pressure can be presented. The board that is connected to the sensor obtains a 3.3V voltage regular and therefore the voltage input may be connected to the 5V pin on our development board. The rest of the pin connections are displayed below in Table 35. Once the code has been written and the boards are connected, the sensor is then ready to be tested. After compiling, the code is running and the serial monitor shall first display the baseline pressure that was read and then display the pressure that is constantly being read and updated.

To sensor tested to correctly and accurately read the current pressure altitude. Since we were unable to have range in pressure or height with the sensor, then the reading of the sensor was trusted by reading the pressure with regards to the stated baseline pressure. We were able to see that the pressure had very minimal (one to two feet of altitude) difference in altitude from the initial baseline pressure. This clarified that the sensor was correctly reading the pressure from the baseline pressure since it was not reading values far from the initial pressure.

Table 35: BMP180 to MCU Pin Conversion

BMP180 Connection	Microcontroller Pin Connection
SDA	Connect to pin 17 (D17)
SCL	Connected to pin 16 (D16)
Vin (5V)	Connected to pin 33 (5V)
GND	Connected to Pin 35 (GND)

6.3.3 DS18B20 – Water Temperature Testing

The DS18B20 Water Temperature sensor was tested for accuracy with regards to functionality and implementation onto our chosen microcontroller unit and development board. The sensor was tested in a bottle of water to ensure the readings and data transmitted was relatively what would be expected from the sensor. The following photograph illustrates the implementation onto the development board and connections made via a USB for proper programming to be done on a PC. After testing, the device was determined to be functional and the corresponding test plan used for functionality confirmation is discussed following the breadboard image.

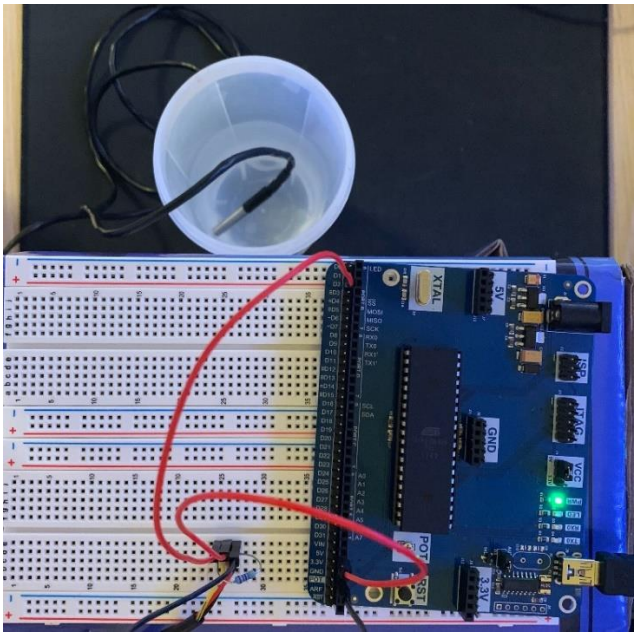


Figure 36: Water Temperature Sensor Testing

In order to begin testing this temperature sensor, it was first required to download and install the necessary Arduino libraries and sketches that will assist us in testing and executing this sensor. To test the sensor, the code must be written to be able to read in the water temperature through the data wire and then translated and converted from the 16-bit integer into a temperature in Celsius and in Fahrenheit. The data is read in through the single data wire and will require the OneWire library within the code. The input voltage will require a total of 3.3V but the data and input voltage will have to be split by a 4.7kΩ resistor. The rest of the pin connections are displayed below in Table 36. Once the code has been written and the boards are connected, the sensor is then ready to be tested. After compiling, the code is running and the serial monitor shall first display the baseline pressure that was read and then display the pressure that is constantly being read and updated.

The sensor was determined to be accurate when the sensor was placed into a cup of water. We were able to read the temperature of the water in both Celsius and in Fahrenheit. Although not knowing the exact temperature of the water, we tested the sensor by placing

the sensor in both warm and in cold water and ensuring that the temperature drastically changed for each cup of water. By having the temperature react to both hot and cold water, we were able to confirm that the sensor did in fact work properly and how we desired.

Table 36: DS18B20 to MCU Pin Conversion

DS18B20 Connection	Microcontroller Pin Connection
Data	Connected to pin 2 (D2)
Vin (3.3V)	Connected to pin 34 (3.3V)
GND	Connected to Pin 35 (GND)

6.3.4 Sensor Interfacing

After ensuring that all of these sensors are capable of working separately, it is imperative to clarify these sensors can properly work together and still be accurate while connected to the MCU. This will entail connecting all sensors to the MCU and transmitting all data read by each sensor. The following photograph illustrates the implementation of all three sensors onto the development board and connections made via a USB for proper programming to be done on a PC. After testing, the sensors were determined to be functional and the corresponding test plan used for functionality confirmation is discussed following the breadboard image.

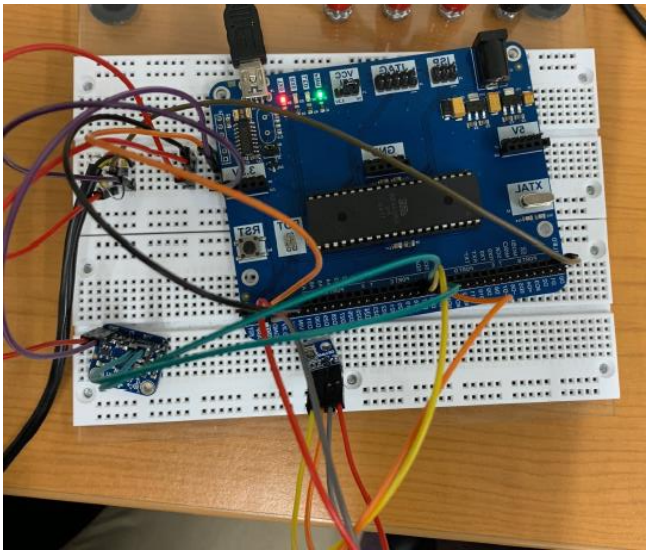


Figure 37: Interfacing Multiple Sensors

To begin testing all three sensors, the sensors were first connected to the MCU with their respective connected pins. With the sensors that required a shared pin, those sensors were pinned outside of the board and connected through wires. The respective pins are displayed below in Table 36. Once all of the sensors are connected to the MCU, the sensors can then be programmed to all be read and transmitted at the same time. The required libraries for the three sensors to be ran are the Wire, OneWire, BMP180, and MMA8451 libraries. With

all of the respective libraries and sketches, the sensors are then ready to be ran simultaneously. Each sensor is read and transmitted and displayed to the serial monitor. From there, we were able to analyze the readings of each sensor to ensure that the sensors are still properly working together.

In order to ensure that each sensor is still accurate when working together, the same procedures for each sensor separately were done. In order to clarify the temperature sensor, the sensor was placed into the cups of water while the accelerometer was placed in different directions to confirm the sensor was properly tracking the change of direction. The pressure sensor was also confirmed to be accurate by ensuring the pressure altitude was remaining constant while testing. Overall, once analyzing each sensor reading in the serial monitor, it was confirmed that the sensors were properly accurate when interfacing and transmitted together.

Table 37: DS18B20 to MCU Pin Conversion

Sensor Connection	Microcontroller Pin Connection
DS18B20 Data Wire	Connected to pin 2 (D2)
DS18B20 Vin (3.3V)	Connected to pin 34 (3.3V)
DS18B20 GND	Connected to Pin 35 (GND)
BMP180 SDA	Connected to pin 17 (D17)
BMP180 SCL	Connected to pin 16 (D16)
BMP180 Vin (5V)	Connected to pin 33 (5V)
BMP180 GND	Connected to pin 35 (GND)
MMA845 SDA	Connected to pin 17 (D17)
MMA845 SCL	Connected to pin 16 (D16)
MMA845 Vin (5V)	Connected to pin 33 (5V)
MMA845 GND	Connected to pin 35 (GND)

6.3.5 NEO-6 – GPS Testing

Testing of the NEO-6 series GPS module was performed to ensure functionality of the device and determine the degree of exposure to the outside world for accurate results. For general programming and microcontroller interfacing with the device, the development board was used along with the corresponding pins for receive, transmit, Vin, and GND. An image is included in Figure 38 below for reference on how the device was tested.

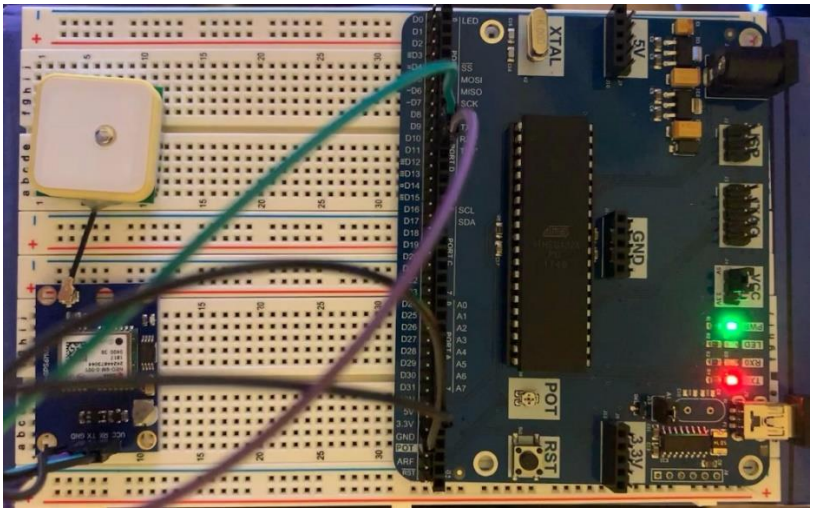


Figure 38:NEO-6 GPS Testing

Table 38: NEO-6 to MCU Pin Conversion

NEO-6 GPS Connection	Microcontroller Pin Connection
TX	Connect to pin 11 (D11)
RX	Connected to pin 10 (D10)
Vin (5V)	Connected to pin 33 (5V)
GND	Connected to Pin 35 (GND)

Upon connection to the microcontroller, the device was programmed with the IDE to test the capabilities of the module. Connectivity strength variances with physical location and potential obstructions was the first step in our testing process. Regarding this step, three conditions would be tested; Complete obstruction of module to the open sky, Partial obstruction with a thin roof/cover, and complete exposure of the module to the sky. It was determined that the module was ineffective in reliably transmitting the GPS data received from the satellites when completely obstructed as inside a building. With partial obstruction, i.e. a thin roof or housing, the module did adequately transmit the received data from the satellites. Complete exposure to the sky bore similar results with the module showing no issue in reading and transmitting the necessary GPS data. To this extent, it was determined that the module would need exterior waterproof housing on the boat to be placed on what is expected to be the center console.

Of note is the data which was determined to be transmitted by the device in the form of NMEA sentences. Various NMEA sentences are produced by the device as this is the standard for global positioning communication and a fundamental understanding of how the data is received is crucial to implementation of this module. Sentences of key importance are included in the table below with a description of their use for our system.

Table 39: NMEA Sentences

Sentence	Purpose
$GPGGA	GPS Fix Data
$GPGSA	Active Satellites
$GPGSV	Satellites in view of module
$GPGLL	Latitude and Longitude
$GPRMC	GPS Data (Time, Position)
$GPVTG	Velocity made good

6.3.6 HM-10 Module Testing

Initial testing was performed purely to ensure the module powered on correctly and was capable of interfacing with the correct data lines. In Figure 39 is an image of the device being breadboarded and connected to the microcontroller unit and development board to ensure it was powered on properly.

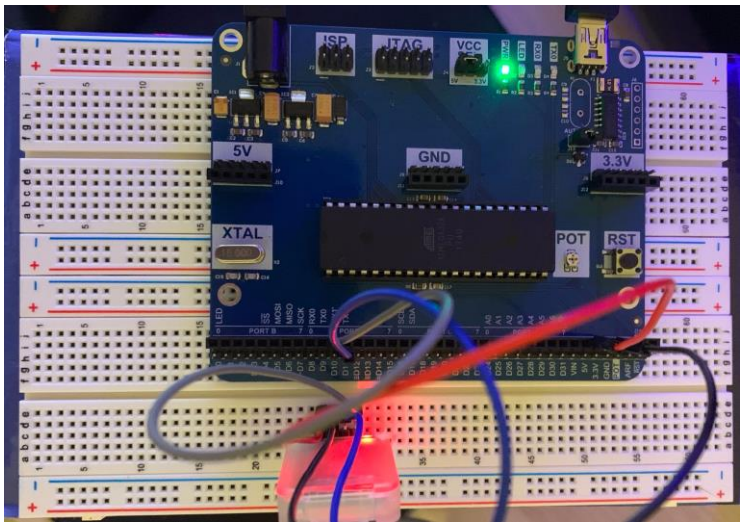


Figure 39:HM-10 Module Testing

Table 40: HM-10 to MCU Pin Conversion

HM-10 Connection	Microcontroller Pin Connection
Tx	Connect to pin 11 (D11)
Rx	Connected to pin 10 (D10)
Vin (3.3V Logic Level)	Connected to pin 33 (5V)
GND	Connected to Pin 35 (GND)

Of note is the logic level utilized by the HM-10 module; It is tolerant to 3.3V which ensures that the functionality for the hardware is optimized for this value. The module has the necessary voltage divider on the breakout to divide an input of 5V from the development board and microcontroller to this logic level input. Accordingly, for a streamlined design of the PCB, a 5V power rail is utilized to power this device and ensure connectivity.

6.4 Software Testing

Just like any part of the system, the software on the MCU was to be tested extensively. The MCU was the control center of the entire boat and needed to be verified to be safe for people to use. By doing thorough testing of software, we were be able to guarantee a higher level of safety on our craft. There are several different components to testing that we would have to go through in order to have a high level of confidence in our work. This also served as a check that we are meeting our customer's needs, and that our sponsors requests will all be served to the best of our ability. Testing revealed issues in our systems, which were dealt with accordingly, which is why it was key to begin testing as soon as possible.

One of the most important parts of software test is unit testing. Unit testing is the testing of individual subsystems, to ensure that each one is validated to be functioning properly before it is combined with other subsystems. By doing this, we made it significantly easier to determine what is causing an issue since we can rule out certain subsystems and focus on the interfaces between them. This testing prevented us from wasting time backtracking to determine which subsystem is failing in the full system and greatly assisted in the fabrication phase with troubleshooting should it arise.

6.4.1 BLE Network Testing

Our BLE network was one of the most critical systems on our project. Getting it up and running early was one of our highest priorities. We chose our BLE module extremely early in our design process so that we could order it and begin testing with it early. There are several stages to the testing of our BLE network, but they all rely on our MCU already being validated to have basic functionality, as well as being able to program it using the external programming board.

The first important step to testing the BLE network was to verify the functionality of the module itself. This means testing it with sample code found online, to make sure that it works with software that is proven to be working. We programmed it with the code found online, then connect to it via an app that we will download on our smartphone. We then checked to make sure messages flow properly between the two devices. We then disconnected and reconnected and retried this several times to ensure we can rely upon the module.

The next step was to connect to the first prototype app from the Computer Science sub team. This was very important, since the third-party apps we use may not support our custom profiles that we added. We verified this by connecting to the smartphone running their app, and sending data to it just like we did with the third-party app. We once again disconnected and reconnected the devices several times so that we can ensure consistency.

This was an extremely important step, since it is unit testing for both our subsystem and the Computer Science team's app as well. This was likely the most challenging phase, since we had to test both of these subsystems at the same time, and there will likely be disagreements on where the issue is if there is one.

The next step was testing the BLE module with our own custom GAP profile. This will allow us to change small connection settings such as the connection interval. We did this by modifying the validated code with a different connection interval and testing if it still functions properly. We then checked that the connection interval has actually changed, before modifying the connection interval again and reperforming testing.

The next major change was testing a simple custom GATT profile. We did this by making a custom service that sends a basic counter variable to the smartphone. We modified the code and then retested it on the smartphone, seeing if the value goes through when we use that service. Once that works, we tried adding more data to it and retesting the process. This process was then repeated with two separate services that each send their own data, to make sure multiple can be run at once. This ensured that our connection will work with all the custom services that we added throughout our development.

The last component we needed to test is the range of the BLE connection. We did this by running it as we did before with all sensors attached and having the mobile phone user slowly back away until they disconnect. We repeated this process and determine what the average distance is. At this point repeated this same process with an obstacle in between the two BLE nodes, to see how much it will affect the connection range. To begin this process, we did the basic testing to determine the effective range of our BLE signal currently. For now, we kept a clear line of sight to get a baseline measure for us to use. Signal strength is measured by its RSSI or received signal strength indication. This shows us in dBm how strong the signal is. Below is a table showing what different ranges of RSSI values effectively mean.

Table 41:RSSI Range vs. Quality

RSSI Range	Quality of Signal
Greater than -50 dBm	Excellent
-50 to -60 dBm	Good
-60 to -70 dBm	Fair
-70 to -90 dBm	Weak
Less than -90 dBm	Practically Unusable

We then measured the RSSI between an out-of-the-box app on one of our mobile devices and our BLE module on our breadboard. While these values did fluctuate quite a bit, we tried our best to average them out. Later in our testing process, we tried using different smartphones, different apps, and different settings for our BLE module, as these will all likely affect the signal strength. Below is a table detailing our results.

Table 42:BLE Signal Reliability Testing

Mobile Phone Distance from Microcontroller	Signal Strength/RSSI
1 Foot	-50 dBm
2 Feet	-53 dBm
3 Feet	-55 dBm
4 Feet	-57 dBm
5 Feet	-60 dBm
10 Feet	-64 dBm
15 Feet	-69 dBm
20 Feet	-75 dBm
25 Feet	-82 dBm
30 Feet	-89 dBm
35 Feet	-92 dBm, cannot connect reliably

From this table, we can see that our signal wasn't ever in the excellent range, which means we likely need to tinker with the settings on the BLE module or try a different app. We could also clearly see the signal strength drop as we got farther away, with it eventually becoming basically unusable.

7.0 Administrative

This section discusses all overhead material for our project. This includes materials not just relevant to our sub-team, but the project as a whole.

7.1 Project Milestones

Project milestones are intended to create measurable goals that a team can use to gauge their progress throughout project development. We used these milestones to ensure we do not fall behind and fail to meet expectations set on us by our sponsors and department. A good milestone should be measurable, achievable, time-sensitive, specific, and results oriented. This criterion will ensure that our milestones are pace so that we meet our goals on time without causing our team members to fail to meet other commitments they may have.

7.1.1 Senior Design 1 Research Phase Milestones

The following table details the milestones that the ECE team needs to meet during Senior Design 1 in the research phase. These milestones are primarily focused on deciding how to tackle our problem, what needed to create our project, and comparing technologies that may be useful for our project. At the end of this phase, we knew all the components and technologies that we wanted to include in our project. These milestones were meant to pace the team and ensure we stayed on track to complete the design with a high degree of confidence in it.

Table 43: Senior Design 1 Research Phase

	Milestone	**Completion Date**	**Status**
Senior Design 1 (Research Phase)	Select Project	8/31/2018	Completed
	Submit Initial Document	9/14/2018	Completed
	Discuss Requirements	9/18/2018	Completed
	Meet with other disciplines and setup team communication	9/18/2018	Completed
	Research requirements	9/27/2018	Completed
	Submit Revised Initial Document	9/28/2018	Completed
	Submit First Draft for Final Document	11/02/2018	Completed

7.1.2 Senior Design 1 Design Phase Milestones

The following table details the milestones that the ECE team needs to meet during Senior Design 1 in the design phase. These milestones are primarily focused on design and test of small subsystems and components. Ideally, a completely fleshed out design should be completed by the end of this phase and semester. These milestones are meant to pace the team and ensure we stay on track to complete the design with a high degree of confidence in it.

Table 44: Senior design 1 design phase milestones

	Milestone	Completion Date	Status
Senior Design 1 (Design Phase)	Determine Preliminary Component	11/10/2018	Completed
	Begin Breadboard Testing of Individual Components	11/16/2018	Completed
	Begin Writing Software	11/16/2018	Completed
	Submit Second Draft for Final Document	11/16/2018	Completed
	Begin Integration of Components	11/16/2018	Completed
	Begin designing and creation of the PCB board	11/20/2018	Completed
	Complete Breadboard prototype	12/03/2018	Completed
	Order final draft of PCB board design	12/03/2018	Completed
	Submit Final Document	12/03/2018	Completed

7.1.3 Senior Design 2 Milestones

The following table includes milestones that will need to be met in the Spring semester during Senior Design 2. The Spring semester should be almost entirely focused on the development and implementation of the actual product. There should be minimal design

work being done, and only when issues are found with the design created during Senior Design 1.

Table 45: Senior Design 2 Milestones

	Milestone	Completion Date	Status
Senior Design II	Complete Initial Software Implementation	02/14/2019	Completed
	Test PCB Design	03/01/2019	Completed
	Test Software and Hardware Integration	03/05/2019	Completed
	Perform Peer Design Review	03/07/2019	Completed
	Test System Integration with Computer Science Team	03/21/2019	Completed
	Mount System on the Hull	04/12/2019	Completed
	Begin Testing of Complete System	04/12/2019	Completed
	Perform Critical Design Review	04/17/2019	Completed
	Complete Testing of Full System	04/18/2019	Completed
	Complete Final Presentation and Documentation	04/22/2019	Completed

7.2 Projected Budget

Below is a table showing the estimated cost of each component of our project. This table will undergo many changes as our design is finalized. Since these are estimates, they refer to not to the price of a specific product, but rather the portion of the budget that we believe we can set aside for a component. A finalized budget with actual prices will be included

in our final design. It is worth noting that the total cost of this project is expected to remain under $1500 so as to meet the requirements set forth by our sponsors. Again, these are estimates and as the final design is refined and has undergone the necessary changes for final integration with the boat produced by the Mechanical Engineering team the total cost will reflect this. However, as seen by the current estimate for the budget, there is a considerable amount of room for error should the need for additional components be desired after further testing.

Table 46: Total Team Budget

Component Name	Description	Count	Total Price
MCU	The control unit for our system	3	$15.45
48V-12V Converter	Converts 48V from the batteries to 12V given to fuse box	3	$164.10
12V-5V Converter	Converts 12V from fuse box to 5V for PCB	10	$32.00
GPS Module	Location tracking and marking	3	$34.97
Bluetooth Module	For transmitting all data to mobile app	4	$39.96
Power FETS & Transistors	For use in power management system	Many	$18.49
Sensors	Water Temperature, Pressure, Accelerometer	3	$26.87
Fuse Box	For connecting all 12V components	1	$33.86
Waterproof Enclosure	Contains all of electronical components	1	$33.36
PCB	Heart of system	10	$140
FTDI	For PCB	1	$15.95
Miscellaneous Expenses	All other various small components, wires, development boards, etc.		$244.23
Total Cost			**$796.24**

Table 47: Total Production Cost

Component Name	Description	Count	Total Price
Torqeedo 48-5000	Batteries that will power the motor and all electronics	2	$10,398
Torqeedo 10.0R	Motor that will provide propulsion to boat	1	$8,999
Charger for 48-5000	Charger specific to the battery type used	1	$899
Navigation Light	For Marine Standard	1	$45.00
MCU	The control unit for our system	1	$5.15
GPS Module	Location tracking and marking	1	$8.99
Bluetooth Module	Transmits data to app	1	$9.99
FETS & Transistors	For use in power management system	Many	$18.49
Sensors	Water Temperature, Pressure, Accelerometer	1	$26.87
Bilge Pump	Remove excess water from bilge	1	$79.99
Livewell Pump	Fill livewell with oxygen and water	1	$44.99
48V-12V Converter	Converts 48V from batteries to 12V for fuse box	1	$39.99
12V-5V Converter	Converts 12V to 5V for PCB	1	$3.20
PCB	Send off final design to be made	1	$15
Waterproof Enclosure	Maintains all electric components	1	$33.36
Miscellaneous Expenses	Various components, wires, and fabrications		$1,489.30
Total Cost			**$22,116.22**

7.3 Methods of Communication Within the Team

Communication within a team is critical for any project, to ensure all members are moving towards the same goals. If members do not all work in the same direction, the end product will not be cohesive, if even functional. Along with plenty of in person meetings, it is important to have common places with written communications between all members so that members can refer back to them as needed, and a cooperative knowledge-base can be built. Given the fact that this is an interdisciplinary project with a total of fourteen team members and quite a few sponsors on board the project as well having stable communication platforms is critical to our design being realized and coming to fruition. That way if any team has questions or needs assistance in creating something from a different team, they may have an easy route of communication within the project as a whole. Similarly, for the benefit of our sponsors they must be able to quickly get into contact with individuals or entire teams as a whole if they have questions or request periodic updates as they have already mentioned. Given the fact this is sponsored project of course our sponsors would want access to any current figures or information we are currently working on the following platforms allow for all that information to be passed through the entire chain. The five major platforms of communication include Discord, Slack, Microsoft Teams, Google Drive and in person meetings. The following sections describe how those platforms are used to this team's advantage during this project.

7.3.1 Discord

The tool chosen for communication between all teams including the Electrical & Computer Engineers, Computer Scientists, and Mechanical Engineers, is Discord. Discord is a free software application built on Electron framework which was allows for multiple channels within a Server. Once a Server is created, an invitation to the potential team members and once the invitation is accepted, the member has access to the several text-channels, including general, and the voice-channels. These multiple channels serve a purpose and each interdisciplinary team acquires their own channel that others are able to join. This allows for the topic of conversation to be automatically set within a channel. A general channel is also an important channel for the use of broader conversation such as time and places to meet with the entire interdisciplinary team.

Other types of channels include voice-channels. These channels allow for anyone on the Server to join the channel at any time in which they have the option to turn on their microphone and speakers. The Electrical & Computer Engineering team utilizes the Voice-Channels often as it is difficult to collectively find a time where we are all available in-person. Also, the Discord application is compatible on iOS and Android which allows every team member to easily access the Server from their mobile phones at any time.

7.3.2 Slack

Within the Electrical & Computer Engineering team, the collaboration tool Slack is used. Slack is similar to Discord in the sense that there are different channels within a Server. Although, in Slack these Servers are called Work Spaces. Our team began our current slack channel by sharing invitations through emails after the Work Space was created.

Once the invitation was accepted, our team acquired access to the general channel as well as direct-messages with each other. Although Slack does not provide a Voice-Channel but does provide Text-Channels including general. Since this tool of communication only consists of the Electrical & Computer Engineering team, we are able to create channels purely for the benefit of our team. For example, after the general channel, the first channel to be created was a channel made specifically for receipts. This allows us to be organized and does not overcrowd the general channel.

Other benefits of slack include the integrated third-party services and applications. Since our final document is written in Google Drive for the capability for all team members to work on the document at once, it is convenient for Slack to integrate with Google Drive. This makes communication about, and with Google Drive. simple within the Slack Channels. Other applications such as Simple Poll is a Slack application that allows users to create a simple poll within Slack. This application allows for quick and easy decision making when time is limited. Another feature the ECE team utilizes is the file-transfers since 5 GB of storage for file-transfer is given to each Work Space for free. Finally, similarly to Discord, this application is available on iOS and Android which allows for all of us to easily and quickly be available to one-another.

7.3.3 Microsoft Teams

Microsoft Teams is the platform used for all interdisciplinary teams, the sponsors, and the advisors. Microsoft Teams is a communication application which is integrated with Microsoft Office 365. This was convenient for our entire team since everyone has an Office 365 account through UCF Knights Email. Similar to other communication platforms used, Microsoft Teams allows for Teams which have messaging through channels. The channels can also include sub-channels to allow the channel to be more organized rather than cluttered. Since Microsoft Teams is created from Microsoft, that also allows for the platform to be integrated with other Microsoft applications such as OneNote, Forms, or Stream. With regards to message history limit, unlike Slack, Microsoft Teams has unlimited amount of history a channel could have. This is important to the use of this platform considering there are roughly 20 people in this channel, which leads to a lot of communication where messages might be lost.

The main advantage that Microsoft Teams has for the purpose of this project is being able to host video conferences with the whole team. Since Microsoft is integrated into our emails, we are able to receive the weekly event reminder that a video conference is occurring. This allows for our team to have meetings while in different areas which leads to a more availability. Also, within this video conference application is a feature that allows screen sharing for all participants in the video conference. For whomever is leading the meeting, they are able to put their notes, slides, or pictures onto the screen, so every team member is able to follow along the presentation. Finally, each Team has a pseudo OneDrive that everyone on the Team is allowed to access. Within this drive, a folder is marked for each interdisciplinary team to provide organization for the files we deliver to the sponsors and advisors.

7.3.4 Google Drive

Google Drive is a cloud storage service available through using a Google Gmail account which permits files to be stored and even shared between other G-mail. Within Google Drive, different files are allowed such as Google Sheets or Google Docs. Google Docs has been used for documentation of research which lets all group members to input their findings onto the same document. Since an extensive report is required during the research of this project, it was vital that our team had a reliable and easy-to-use platform that could handle a large document being shared by multiple people. It was also important to ensure that this platform was trustworthy regarding storage since the amount of writing could not afford to be lost. Google Sheets has allowed group members to manage budgeting with large parts and have also created an inventory list of all parts purchased. Whenever a part has been bought or received, it is simple to enter the Google Drive in order to submit this information into the Google Sheets. If a team member needs to reference whether or not a part or component has been bought or received, the Google Sheet containing this information can be easily accessible. Other features the Google Sheets have provided help towards the group has been to allow each team member to communicate and mark-down what their given task was and whether the task has been completed or is in process. This let our group members to easily stay up to date with each other regarding the status of their personal tasks and break-off tasks.

7.3.5 Trello

Trello is a web-based project management application that was originally created in 2011. This web-passed application makes it easy for teams to collaborate in a simple and user-friendly environment. Trello has been useful in many different fields including real estate management, project management, school bulletin boards, gaming, and software project management. For our team, this application is especially useful since there are many small tasks to complete that could easily be recorded without having to have a conversation about. This allows for us to create our tasks and then mark them as complete when they are finished. In return, other team members will no longer have to ask each other if simple tasks were complete and can instead refer to Trello for an update on tasks.

There are numerous features within Trello that allow for our team to smoothly move along in our project and responsibilities. Some of the features that are helpful include organizing tasks by tags and labels. For example, if tasks are pertaining to writing the document, ordering parts, or just testing and building, the board can be easily filtered or organized to have a better understanding of what has been done within those tags. Another important feature for our team has been the ability to assign tasks to others. If a team member has not created the job or marked it as done, we are able to remind other members by creating and assigning the duty to that team member. Overall, Trello will continue to assist us throughout both semesters of senior design.

7.3.6 In Person Meetings

Although all of the online communication described above is essential as teams split off and work on their own subsequent documents and designs during the course of the two

semesters. There was still a need for meeting in person from time to time to either show models with accurate descriptions or to have a large meeting where issues revolving large areas of the project can be discussed without much time constraint and with possible solutions being created. Our group had weekly or bi-weekly in person meetings outside of class to discuss current issues and progress on each group members sections of the project they should be researching and working on, another important point to in person meeting is as the paper goes along it would be wise to all work on the paper while in the same area as formatting or section errors can easily be fixed. Also, when the group is typing and working together as a whole, something one group member may have missed can be easily caught by another group member in the same space without waiting for a response if the error has to be communicated online. The group also planned to begin meeting with the computer science team that is assigned to this project bi-weekly as well to discuss plans on how our communications from our system design tied in their database and mobile app work since they are spearheading that side of the project. Eventually late during the semester and during senior design two all team met to discuss housing of our system which was largely a task for the mechanical engineers to assist us with. Overall this entire collaborative effort to realize the project relied on each member filling their responsibility and assigned roles. Having a face to face conversation with all other team mates made this a reality.

7.3.7 GitHub

GitHub is by far the most widely used online repository for coding. GitHub allows users to easily manage different versions of their code and combine them as desired. This enables multiple developers to work on the development of code simultaneously, with minimal hassle. GitHub's branching system makes it very easy to combine different files and code from all of its users, meaning that all users have to do is to do is decide on what features they want from which branch. GitHub utilizes Git, the widely used open source version control system. This allows users to integrate all sorts of addons that have been developed for Git. GitHub has many features that were extremely useful to our team as we try to keep our code organized. GitHub's features are perfect for managing a project with multiple developers, such as ours. In Figure 40 below, the start of our repository with the Computer Science subteam.

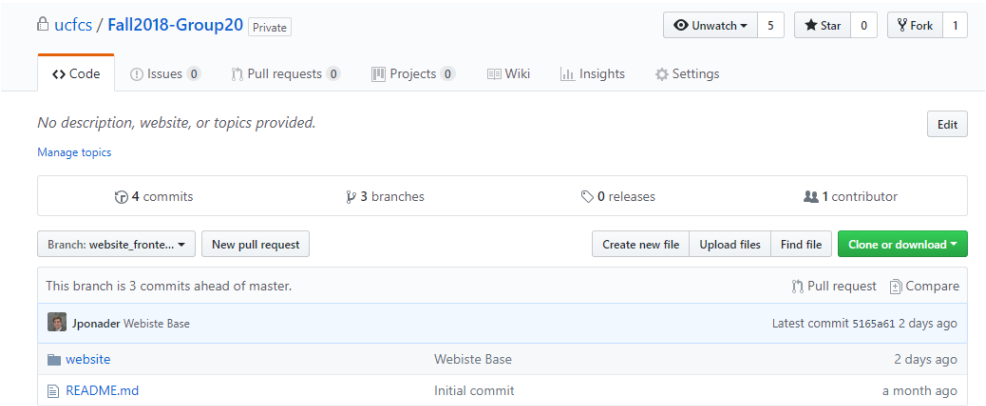


Figure 40: GitHub Repository

7.3.7.1 Repositories

A repository (repo) is a project location on GitHub. The repo contains all files for the project, as well as all the related metadata for the project. Repos can be set to public or private, depending on whether you want to control who can view or make pull requests for the repository. Repos can then be shared to team members who you wish to have access to it, and then they would be able to reach it by looking at their repos on their account or by going to the associated URL. Lastly, repos can also be forked. Forking a repo means creating a new project based off of the forked repos. This allows other people to make continuations off of a project that was no longer being worked on.

7.3.7.2 Branches

A branch in Git is an isolated piece of development work. When someone begins working on a new feature or fix, they will create a branch from the master branch that will allow them to make changes without destroying the working master branch. Once a branch is deemed to be complete, it can be merged back into the master branch in order to update it. Branches should be used to fix bugs, create new features, and to experiment with potential ideas. When your code is ready to be publicly available, you can simple pull from your master branch in order to get to the most up-to-date version without risking including unfinished changes.

7.3.7.3 Pull Requests

A pull request is a request to merge a branch back into the branch it came from. The developer can also assign reviewers to ensure that the changes that they are merging are approved by all stakeholders in the project. Pull request will show the differences in all files that have been modified in the branch that you are merging. When the merge goes through, you will have to choose which changes should come from which branch. This allows you to ensure you are not overwriting any features that have been added since you created your branch. Pull requests are also a great forum to discuss the changes that are occurring, to ensure that the changes are being executed in the correct manner. If it is decided that the changes are not correct or done correctly, the pull request can be rejected and closed. Otherwise, the pull request should be approved so that it may be merged.

7.3.7.4 Documentation

GitHub has many built in features to support the creation of documentation. Documentation is a key part in any software project, but even more so in ours since we passed development off to Watershed Innovations after we graduate. By making documentation within GitHub, it’s extremely easy to contextualize what you’re writing, since it can directly reference the code itself. GitHub allows you to create Wikis that support your code, with complete version control and an easy to use text editor. GitHub pages also allow you to easily publish your documentation alongside your code, making deployment even easier.

8.0 Project Summary and Conclusion

Through our research and design process for this year, we have encountered many issues that we have been able to overcome. We all have learned a lot about PCB design, drawing schematics, and writing well-structured code. Through the struggles that we have faced over the last nine months, we feel prepared to enter the workforce with a greater appreciation of design. While we are sure that many more challenges await us, we feel prepared to tackle them head on, and come out on top to ensure we deliver a quality product in all aspects of work or school.

Through our research over the course of two semesters, we have created and implemented a design that meets all the requirements of our department, sponsors, and their customers. We have created a unique system that combines modern Internet of Things (IOT) concepts with classic marine design. Through this mixture of wireless connectivity and a reliable propulsion design, we enable Watershed Innovations to offer an extremely unique product that has nearly no competition in the current marketplace. Our wireless communications and usage of a microcontroller allow Watershed Innovations to easily iterate on our design long after we all graduate. The microcontroller easily allows new sensors and addons to be added, while the software design allows them to easily add new functionality through the use of a common language, C. The usage of Bluetooth Low Energy profiles make it extremely easy for new data packages to be sent between the microcontroller and the app/database.

The boat we are creating is not only unique due to its modularity, but also due to the high-level goals provided by Watershed Innovations. Very few flats fishing boats are built from aluminum, due to the quiet properties of fiberglass. However, an aluminum flats fishing boat is not only cheaper to build, but also stands as proof to the ingenuity of Watershed Innovations. Due to the inclusion of the electric propulsion system provided by Torqeedo, we also have an extremely quiet method of approaching the desirable fishing grounds without startling any of the fish that the customers are trying to catch. This hasn't been done by any of the competition, as they all still rely on a gas-powered motor and the user to paddle it the last five hundred yards or so to their destination. This makes our boat extremely unique, and is truly a revolutionary product in its market.

We have successfully created a design that meets all of Watershed Innovation's requirements, while also staying well below our budget. Due to the innovative thinking of our team, we were able to help save Watershed Innovations a significant portion of our budget. We have also stayed noticeably ahead of Watershed Innovation's timeline for almost the entire duration of this semester. This allows them to focus more on setting the best requirements for us and on ensuring that the hull design for this project is perfect, since it is one of the most critical parts of this project. Due to the criticality of the hull design, our advisor and we thought it would be best to invest in a miniature boat that allows us to build and test our system regardless of the state of the actual hull. This ensures that even if the mechanical sub-team has a critical failure, that we are still be able to test the validity of our design. This is key to making sure we are able to deliver our subsystem to Watershed Innovations, so that they do not potentially experience a catastrophic cascading failure if

the hull design does not succeed. This type of project segmentation is a key piece of an Agile development cycle, as discussed earlier in our paper.

Our team has considered many possible failure points that we could encounter during the build phase. We did our best to plan for these potential issues through our test procedures, as well as leave extra time and budget so that we were able to overcome these obstacles. For example, one of the biggest issues we were worried about is whether our microcontroller was able to have enough computation power to keep up with the demands put on it by our system. We also tried to draft rough safety procedures that our system will follow to ensure we never put the users of our boat in a dangerous situation. We tried to tackle as many of our foreseen issues through our design itself, and many of the issues we encountered and solved. While our design is very thorough and complete, there is always issues that cannot be foreseen, and we tried to ensure that we had the resources to handle these issues when the time comes.

When it came to part procurement and testing, we were fortunate in most regards. Almost all of our parts were delivered within a week of ordering them. The only exception to this was the development board, which took a few weeks to be delivered since it is from Europe and is custom made. Over the initial weeks of senior design II, we ordered our PCB, and immediately began testing of it and basic subsystem testing. This included ensuring that the PCB was built to the standards that we have set in the design of it. We then began integrating all the parts we already have, ensuring that all interact properly with the PCB. We expected to have some issues with at least one of these parts, which is why it was so critical that we performed this testing early. We also started building our complete program for the MCU. We then worked with the CS team to validate our initial product with their baseline mobile application.

In the closing weeks of Senior Design II we were able to fully fabricate and integrate our electronics with the final design of the boat. Wiring posed the largest challenge with regards to integration as thick data and power cables from the battery and motor systems proved difficult to pass through the wireways. Nonetheless, wiring was accomplished and the system functioned as expected based on the initial testing performed in the first semester of Senior Design. The mobile application worked cohesively with our electrical system as well, ensuring that our design did indeed produce reliable data which could be pushed and stored utilizing the users smartphone.

Our team has learned many things already from this project so far. Our technical skills within our respective areas have grown exponentially. Not only have we gained experience with working with actual PCB’s but we have also learned about designing marine-grade electronics. We have also learned a lot about working with realistic standards and constraints, which will certainly be beneficial to us all when we graduate and move into our future careers. We have also learned a lot about real-world project workflow, and how to properly communicate with not only our team members, but also other development teams and our customers. Working with Watershed Innovations has given us a truly unique insight into what our future careers may hold.

9.0 Appendices

The appendices consist of the various resources utilized in this paper with regards to figures, tables, datasheets, or other necessary information for the design of this project.

9.1 References

[1] "NEO-6 Series". U-Blox, 2018, https://www.u-blox.com/en/product/neo-6-series. Accessed 4 Nov 2018.

[2] "Linear Vs. Switching Regulators". Renesas Electronics, 2018, https://www.renesas.com/sg/en/products/power-management/linear-vs-switching-regulators.html. Accessed 16 Nov 2018.

[3] "IP Rating Chart | DSMT.Com". DSMT.Com, 2018, http://www.dsmt.com/resources/ip-rating-chart/. Accessed 16 Nov 2018.

[4] "Introduction To RF & Wireless Communications Systems - National Instruments". Ni.Com, 2018, http://www.ni.com/tutorial/3541/en/. Accessed 26 Oct 2018.

[5] "Consent Form | Salt Water Sportsman". Saltwatersportsman.Com, 2018, https://www.saltwatersportsman.com/building-an-nmea-2000-network. Accessed 5 Nov 2018.

[6] Olimex.Com, 2018, https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf. Accessed 11 Nov 2018.

[7] Wireless.Fcc.Gov, 2018, https://wireless.fcc.gov/marine/vhfchanl.pdf. Accessed 16 Nov 2018.

[8] Gpo.Gov, 2018, https://www.gpo.gov/fdsys/pkg/CFR-2010-title33-vol1/pdf/CFR-2010-title33-vol1-sec83-22.pdf. Accessed 3 Nov 2018.

[9] Nmea.Org, 2018, https://www.nmea.org/Assets/2000-explained-white-paper.pdf. Accessed 8 Nov 2018.

[10] Nema.Org, 2018, https://www.nema.org/Products/Documents/nema-enclosure-types.pdf. Accessed 27 Oct 2018.

[11] Uni-Valve.Com, 2018, http://www.uni-valve.com/files/pdf/teknik/ip.pdf. Accessed 16 Nov 2018.

[12] Ww1.microchip.com. (2018). [online] Available at: http://ww1.microchip.com/downloads/en/devicedoc/39582b.pdf [Accessed 16 Nov. 2018].

[13] MSP432P401R, (2018). [online] Available at: http://www.ti.com/product/MSP432P401R [Accessed 16 Nov. 2018].

[14] CC1352R1, (2018). [online] Available at: http://www.ti.com/tool/LAUNCHXL-CC1352R1 [Accessed 16 Nov. 2018].

[15] MSP430G2553, (2018). [online] Available at: http://www.ti.com/product/MSP430G2553 [Accessed 16 Nov. 2018].

[16] WEBBENCH, (2018). [online] Available at: http://www.ti.com/tools-software/design-center/webench-power-designer [Accessed 16 Nov. 2018].

[17] Microchip.com. (2018). ATmega1284P - 8-bit AVR Microcontrollers - Microcontrollers and Processors. [online] Available at: https://www.microchip.com/wwwproducts/en/ATmega1284P [Accessed 16 Nov. 2018].

[18] Learn.sparkfun.com. (2018). I2C - learn.sparkfun.com. [online] Available at: https://learn.sparkfun.com/tutorials/i2c/all [Accessed 16 Nov. 2018].

[19] Learn.sparkfun.com. (2018). Serial Peripheral Interface (SPI) - learn.sparkfun.com. [online] Available at: https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/res [Accessed 16 Nov. 2018].

[20] UltraSonic Image, (2018). [online] Available at: http://www.ti.com/content/dam/ticom/images/products/ic/sensingproducts/diagrams/what-is-ultrasonic-sensing.png [Accessed 16 Nov. 2018].

[21] EasyEDA, (2018). [online] Available at: https://easyeda.com/feather/EasyEDA_for_Electronic_Circuit_DesignrUU70mawt [Accessed 16 Nov. 2018].

[22] EagleCAD. (2018). [online] Available at: https://intranet.ee.ic.ac.uk/t.clarke/EAGLE/The%20EAGLE%20Guide.pdf [Accessed 16 Nov. 2018].

[23] Docs.kicad-pcb.org. (2018). [online] Available at: http://docs.kicad-pcb.org/stable/en/getting_started_in_kicad.pdf [Accessed 16 Nov. 2018].

[24] Radio-electronics.com. (2018). 802.11e QoS :: Radio-Electronics.Com. [online] Available at: https://www.radio-electronics.com/info/wireless/wi-fi/ieee-802-11e.php [Accessed 16 Nov. 2018].

[25] Arduino Project Hub. (2018). DS18B20 (digital temperature sensor) and Arduino. [online] Available at: https://create.arduino.cc/projecthub/TheGadgetBoy/ds18b20-digital-temperature-sensor-and-arduino-9cc806 [Accessed 16 Nov. 2018].

[26] Industries, A. (2018). BMP180 Barometric Pressure/Temperature/Altitude Sensor-5V ready. [online] Adafruit.com. Available at: https://www.adafruit.com/product/1603 [Accessed 16 Nov. 2018].

[27] How-To Geek. (2018). *What Is GitHub, and What Is It Used For?*. [online] Available at: https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/ [Accessed 1 Dec. 2018].

[28] Learn.adafruit.com. (2018). *Introduction | Introduction to Bluetooth Low Energy | Adafruit Learning System*. [online] Available at: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction [Accessed 1 Dec. 2018].

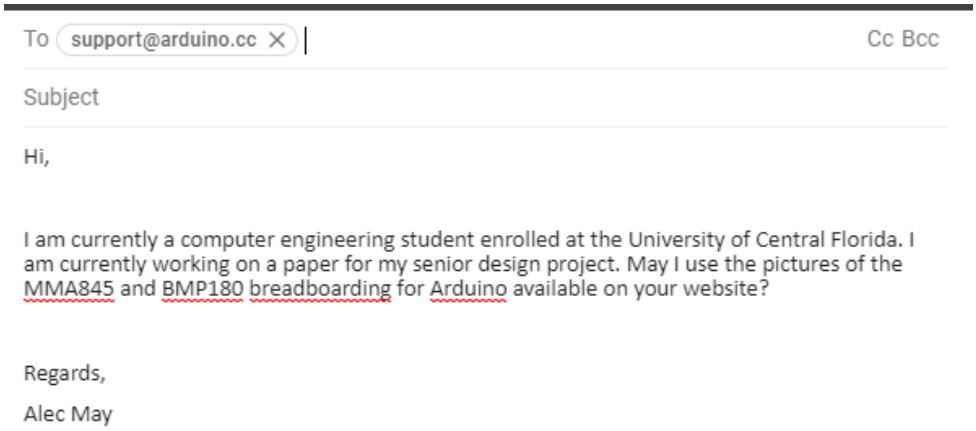
[29] Pepperl+Fuchs. (2018). *Ultrasonic Sensors Knowledge (Part 4): Influences on Measurement Accuracy*. [online] Available at: https://www.pepperl-fuchs.com/usa/en/25518.htm [Accessed 1 Dec. 2018].

[30] Cow.ceng.metu.edu.tr. (2018). [online] Available at: https://cow.ceng.metu.edu.tr/Courses/download_courseFile.php?id=6054 [Accessed 1 Dec. 2018].

[31] Math.uaa.alaska.edu. (2018). [online] Available at: http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf [Accessed 1 Dec. 2018].

[32] Sacolick, I. (2018). *What is agile methodology? Modern software development explained*. [online] InfoWorld. Available at: https://www.infoworld.com/article/3237508/agile-development/what-is-agile-methodology-modern-software-development-explained.html [Accessed 1 Dec. 2018].

9.2 Copyright Permissions


To support@arduino.cc ×

Cc Bcc

Subject

Hi,

I am currently a computer engineering student enrolled at the University of Central Florida. I am currently working on a paper for my senior design project. May I use the pictures of the MMA845 and BMP180 breadboarding for Arduino available on your website?

Regards,

Alec May

Figure 41: MMA845 and BMP180 Breadboard Permission

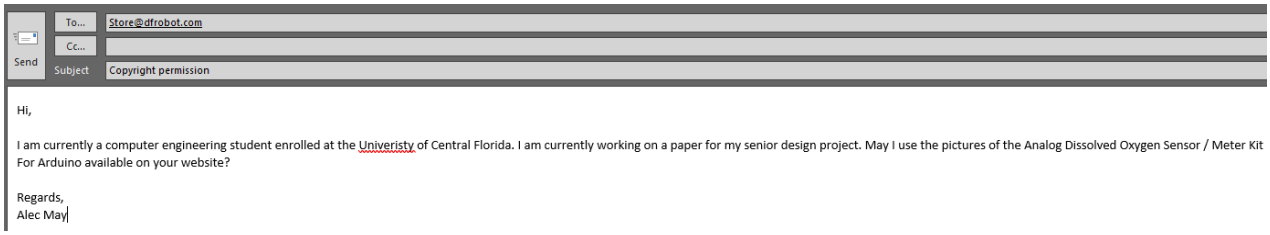


Figure 42:Dissolved Oxygen Sensor Breadboard Permission

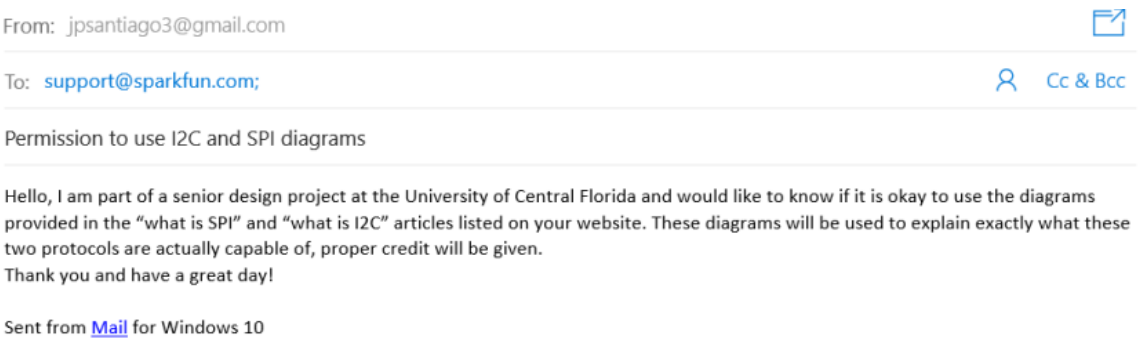


Figure 43:SPI and I2C Figures Permission

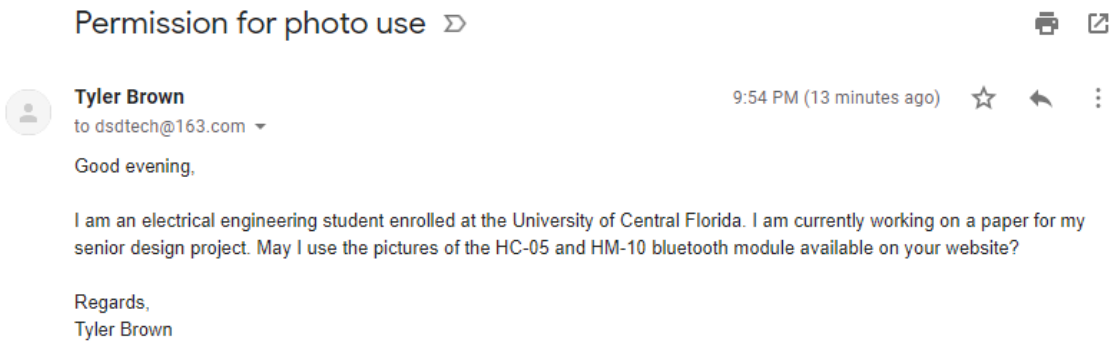
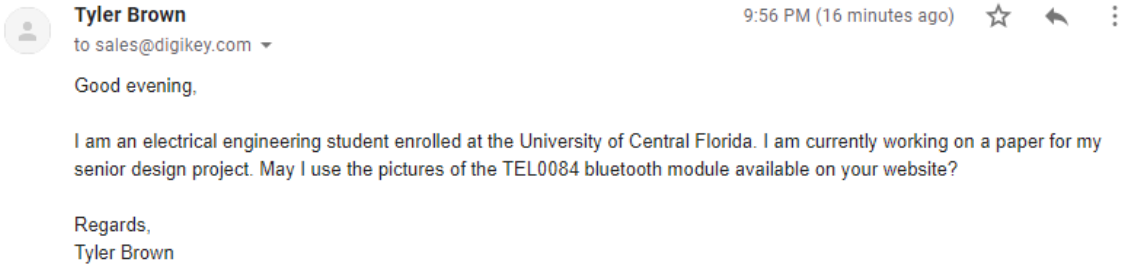


Figure 44: HC-05 and HM-10 Bluetooth Module Permission



Tyler Brown 9:56 PM (16 minutes ago)

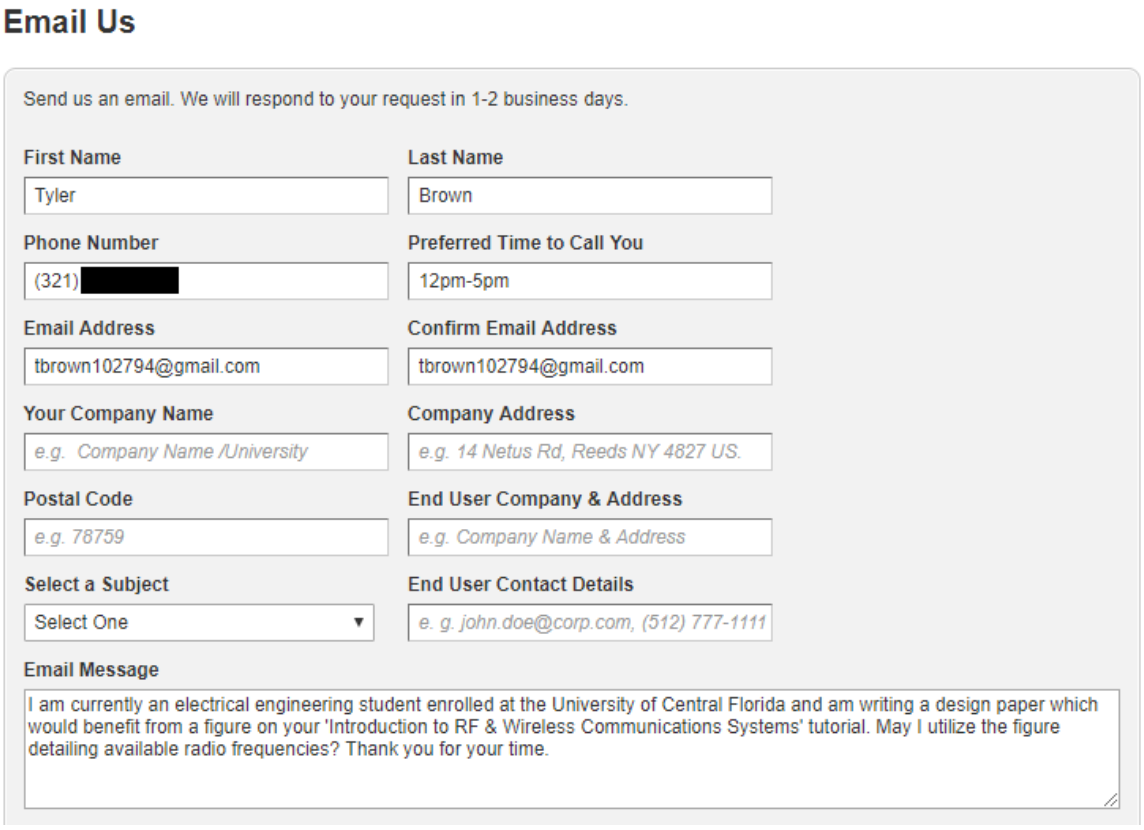
to sales@digikey.com

Good evening,

I am an electrical engineering student enrolled at the University of Central Florida. I am currently working on a paper for my senior design project. May I use the pictures of the TEL0084 bluetooth module available on your website?

Regards,
Tyler Brown

Figure 45:TEL0084 Bluetooth Module Permission



Email Us

Send us an email. We will respond to your request in 1-2 business days.

First Name: Tyler

Last Name: Brown

Phone Number: (321) [redacted]

Preferred Time to Call You: 12pm-5pm

Email Address: tbrown102794@gmail.com

Confirm Email Address: tbrown102794@gmail.com

Your Company Name: e.g. Company Name /University

Company Address: e.g. 14 Netus Rd, Reeds NY 4827 US.

Postal Code: e.g. 78759

End User Company & Address: e.g. Company Name & Address

Select a Subject: Select One

End User Contact Details: e. g. john.doe@corp.com, (512) 777-1111

Email Message

I am currently an electrical engineering student enrolled at the University of Central Florida and am writing a design paper which would benefit from a figure on your 'Introduction to RF & Wireless Communications Systems' tutorial. May I utilize the figure detailing available radio frequencies? Thank you for your time.

Figure 46: RF and Wireless Communications Table Permission

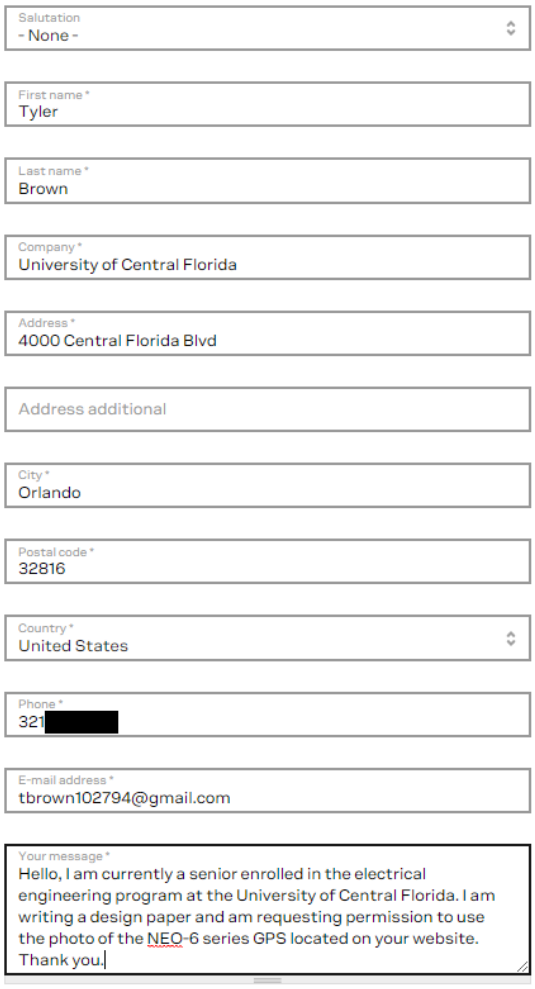


Figure 47:NEO-6 GPS Picture Permission

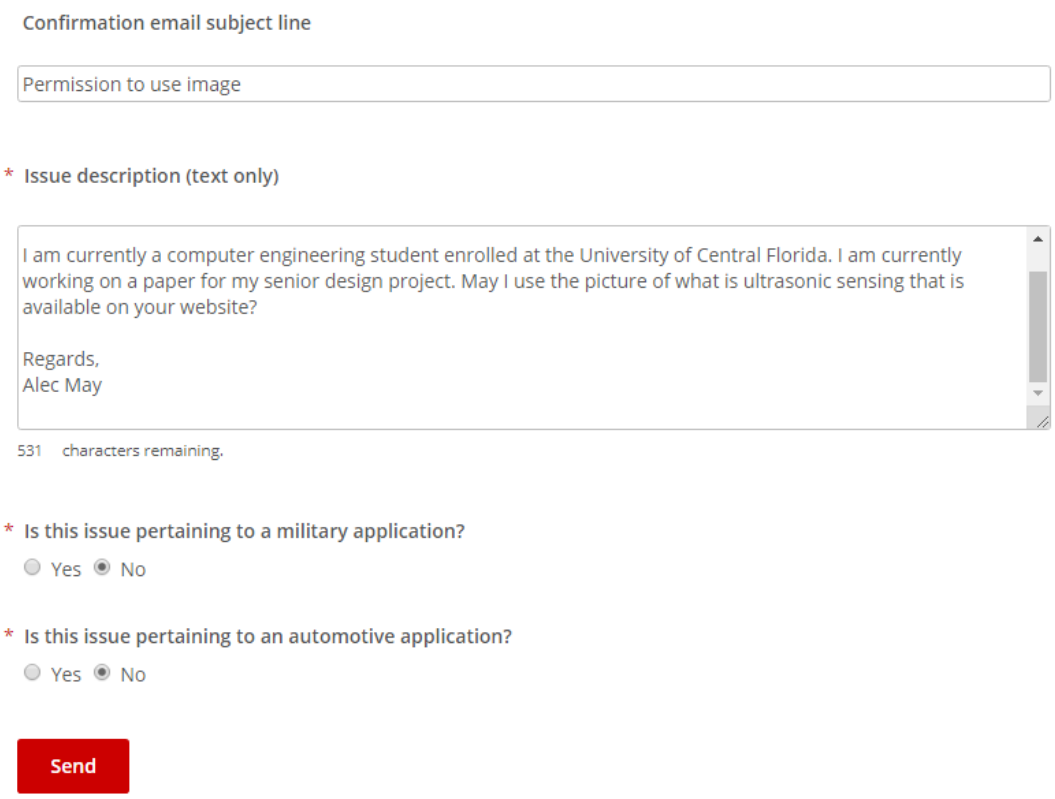


Figure 48:Ultrasonic Diagram Permission