Autonomous Hydrographic Survey Vessel

Andrew Alley
Department of Electrical Engineering
Tennessee Tech University
Cookeville, TN
atalley42@tntech.edu

Carter Ashby
Department of Electrical Engineering
Tennessee Tech University
Cookeville, TN
cashby42@tntech.edu

Levi Daniel
Department of Electrical Engineering
Tennessee Tech University
Cookeville, TN
lbdaniel42@tntech.edu

Joshua Herrera
Department of Electrical Engineering
Tennessee Tech University
Cookeville, TN
jherrera42@tntech.edu

I. INTRODUCTION

Bathymetry is the science of measuring and observing the depths of different bodies of water, such as lakes, rivers, oceans, etc. These analyses are crucial for determining the shape of the water body. Furthermore, data obtained from bathymetry can be used to monitor the effects of climate change, beach erosion, and subsidence. The Tennessee Tech Water Center also uses bathymetry to create simulations to predict natural disasters and locations that are more susceptible to flooding.

The Water Center currently uses remote-controlled boats that take bathymetric projections by navigating the boat back and forth across the body of water. Performing this task manually is labor-intensive and prone to error. Depending on the size of the body of water, it could take several days to collect all the necessary data. The resolution of the data is also too low to create a useful bathymetry map. There are autonomous hydrographic survey vessels (HSV) commercially available; however, these products greatly exceed the budget of the Water Center with autonomous HSVs starting at a price of $70,000.

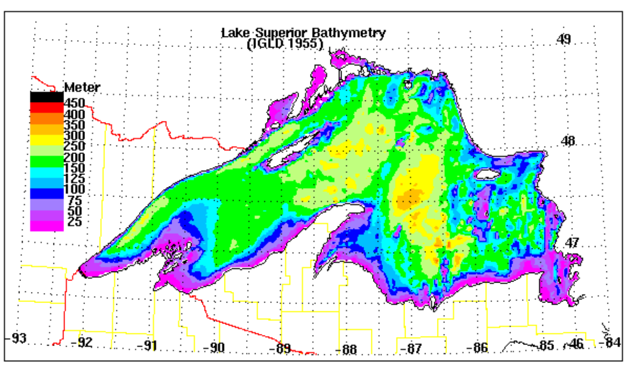

Figure 1: Bathymetry Map of Lake Superior

A. Overall Idea

The requirement to deliver a boat that will navigate the entirety of a body of water without direct control from the user will be met by constructing a watercraft driven by software that intelligently plots and follows a course that will allow the boat to collect measurements over the full area of the body of water. To collect the required bathymetry data samples the boat will be equipped with sonar and GPS. A sonar sensor is the most likely solution for depth measurement device. An accurate GPS-generated location reading will be recorded with each sonar measurement, and data processing software will be used to generate maps and charts from the collected data. A software application will also be developed that will allow someone onsite to transmit commands to the boat while also being able to view the collected bathymetry data in real time.

B. Constraints and Regulations

The boat was given these initial specifications:

- The boat must successfully measure the entirety of Cane Creek Lake (56 acres) in one test run
- The boat needs to travel at a speed of 3-5 ft/s to ensure accurate sonar data measurements
- Data such as hydrographic measurements, GPS location, and battery voltage levels need to be transmitted from the boat to a land base in real time.
- The users on the land base need to be able to switch between manual and autonomous control.
- Obstacle avoidance needs to be implemented to prevent any collisions when the boat is in autonomous control.
- An abort option needs to be implemented if the boat is receiving inaccurate data measurements.
- The overall design and construction price should not exceed commercially available HSVs.
- The design must be easily replicable for any group interested in using this project to take their own hydrographic measurements.

The regulations that govern remote control radio signals require that the boat only be controlled via radio signals in the 26-28 MHz band and that power output from the transmitter be no more than 4 Watts. The radio signals used to control the boat must also only control the boat; they cannot be used to transmit data from the collection process [5].

The regulations that govern Wi-Fi signals require that Wi-Fi frequencies be restricted to 902-928 MHz, 2400-2483.5 MHz, or 5725 to 5875 MHz, as they are considered unlicensed wireless equipment devices [6]. The transmitter is also required not to exceed a peak output power of 1 Watt, and the Effective Isotropic Radiated Power (EIRP) cannot exceed 4 Watts [7]. The EIRP is simply a combination of the output power and the antenna gain.

II. Salient Outcomes

A. *Battery Capacity and Monitoring:*

The Water Center has requested that the autonomous hydrographic surveying boat have a run time of at least one hour and an accurate battery monitoring system. The current design of the power subsystem allows surveying teams to operate the boat for approximately 3 hours by use of 2 Liperior Lithium-Polymer batteries (14.8 V, 22000 mAh) connected in parallel. Additionally, 2 quad op-amp circuits were constructed and implemented that provide voltage readings of each cell in the Liperior batteries. These voltages can be verified by using the battery charging station that displays the actual cell voltages.

B. *Data Transmission:*

The specification to communicate data such as hydrographic measurements, GPS location, and battery voltage levels from the boat to a land base in real time has been addressed by the data transmission subsystem. A pair of Arduino Unos and two nRF24L01+ radio modules were used in the subsystem design. The subsystem uses 2.4 GHz radio frequency (RF) to transmit the collected data to the base station PC. This system tested on Cane Creek, and it was confirmed that it could reliably transmit GPS data and battery voltage levels measured from the onboard battery monitoring system. The most up-to-date version of this system had a reliable connection at up about 550 meters and was transmitting GPS and battery data successfully. Additional work will be needed to complete the integration with the sonar system in order to transmit hydrographic data.

C. *GPS Accuracy and Precision:*

The use of Real-Time-Kinematic GPS receivers allowed us to achieve GPS data with much more accuracy than normal receivers. This project makes use of two simpleRTK2b receivers with a long-range transmission to receive latitude and longitude coordinates with centimeter level accuracy and precision. The two receivers are set up in a base and rover configuration. This allows the "rover" receiver, which is on the boat itself, to receive RTCM corrections from the "base" receiver which is connected to a PC on land. For this project, centimeter level accuracy and precision is required for the boat to navigate the predefined path. This can only be achieved if the location of the PC with the base station is known and uploaded to the base station receiver. After testing this configuration, the accuracy was found to be 2 cm and the precision was about 3 cm.

III. Outline of Report

The objective of this paper is to present the problem that this project aimed to solve and the steps in developing the solution. The goal of this project is to develop an autonomous machine that measures the depth of bodies of water. These measurements will then be used to create an accurate bathymetry map of the lake, river or pond that is being mapped. The rest of this paper will describe the science and engineering behind this project and the process by which this project was completed.

IV. Literature Review

Two of the commercially available products for autonomous HSVs are the CEE USV from CEE Hydro Systems, and the Seafloor HyDrone-RCV survey boat by FLT Geosystems. Both boats allow the user to switch between manual and autonomous navigation. They are both used for taking hydrographic surveys of bodies of water using autonomous navigation. We modeled our original idea for the navigation program after these products.

Figure 2: CEE USV

The main difference between our project and these products is the cost and functionality of the devices. The CEE USV has been noted to be upwards of $70,000 and the HyDrone-RCV with the autonomous navigation upgrade is around $15,000. One of the biggest cost cuts came from the RTK GPS receivers. The CEE USV uses a receiver that is well over a few thousand dollars while we are using a simpleRTK2b receiver kit which is less than $900 [12].

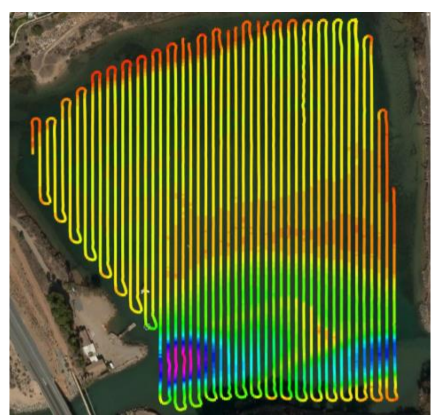

Figure 3: Intended Path Generation Method

The method of autonomous navigation is also a significant difference between the commercial products and our project. This model uses a predefined path that is specific for the body of water that needs to be surveyed. These paths can be easily generated using path planning software such as ArduPilot. The

GPS points can be entered into the autonomous program and the Raspberry Pi will process them and use them as waypoints.

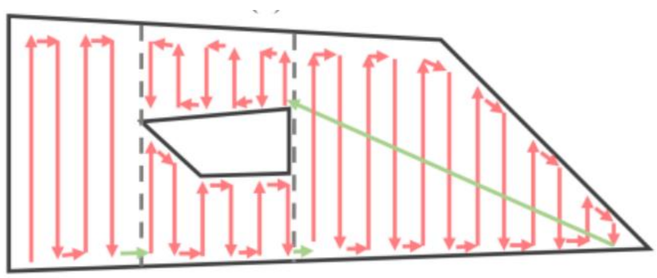

Figure 4: Boustrophedon Cellular Decomposition

Originally, we planned to use an algorithm known as Boustrophedon cellular decomposition. This is a commonly used path planning method to effectively cover large surfaces such as fields [13]. Autonomous lawn mowers use this method to cover entire fields efficiently. This method involves splitting an area up into sections and determining the best path to follow in order to minimize travel distance and travel time. Unfortunately, we were not able to incorporate this due to time. So, the previously mentioned method of predefining a path was used instead.

V. Design Process

The Water Center has asked us to develop an autonomous vessel that is capable of navigating a given body of water. The specifications include that the boat should be able to navigate Cane Creek Lake within its battery life as well as travel at optimal speeds for bathymetric data collection. The boat must also be able to transmit data back to an on-land computer. The design that was implemented for this project includes a navigation system, power system, and data transmission system. The purpose of the navigation system is to make the boat autonomous while keeping the boat at a specified speed and keeping the boat on a path that is optimal for bathymetric data collection. The power system is responsible for distributing power to each component and regulating voltage supplies. The data transmission system transmits GPS data, battery voltages, and eventually sonar data.

A. Original Navigation System

The manual control portion of the navigation system will assist in meeting the specifications that the boat will be autonomous and that the boat will have an abort operations function. For the boat to be able to autonomously navigate the body of water, the GPS coordinates of the perimeter of the lake must be measured. This process will require the user to manually drive the boat around the lake with a controller. The manual control system also needs to have programmable inputs for the abort function. In an emergency the user will press a button on the manual controller that tells the boat to stop collecting sonar data and return to the user. The user will also be able to tell the boat when to start collecting perimeter data, when to switch to manual control, and when to begin autonomous navigation. These will be assigned to inputs on the remote controller by the user. These functions were determined to be necessary to improve autonomy or ease of use.

The components that will be used for this system include the Spektrum DX6E controller and the Spektrum AR620 receiver. The decision to use this controller and receiver is based on its available programmable switches and its ability to operate the current boat model that was built by the Water Center at TN Tech. This was observed when a test run was conducted with the current boat model on Cane Creek Lake. The Spektrum controller will allow the project to meet the specification that the boat will be able to collect bathymetric data, the boat will be autonomous, and the boat will have the ability to abort data collection in case of an emergency. For the boat to begin collecting bathymetric data, it will need to be driven manually around the perimeter of the lake so that GPS data for the lake can be recorded. To do this, the Spektrum controller and receiver will need to output a signal that can control the electronic speed controllers (ESCs) that control the motors of the boat. This was observed by test driving the current model of the boat which implemented the Spektrum controller and receiver as well as Blue Robotics Basic ESCs and Blue Robotics T200 brushless thrusters. These are the same ESCs and thrusters that will be implemented in the autonomous boat. The range for the receiver is listed as 2 kilometers [1]. This distance is suitable for the purposes of navigating Cane Creek Lake. The length of Cane Creek is just over half a mile and since the user must always be able to see the boat, it will be easy for the user to stay within range. This controller also has extra switches on it that can be programmed for desired functions [2]. These switches can be assigned to the receiver outputs which will connect to the Raspberry Pi 4 in the onboard navigation system. These will control the boat's mode of operation.

Figure 5: RC controller and Receiver

B. Navigation System (Autonomy)

The autonomous portion of the navigation system will directly meet the specification that the boat will be autonomous. It will use a GPS device to record data for the perimeter of the body of water. A microcontroller will process this data and create a path for the boat to travel that is optimal for sonar data collection. The microcontroller will then send signals to the motors that control the thrust to steer the boat along the created path. Signals from the user will be sent to the microcontroller to adjust the boat's mode of operation.

The Blue Robotics T200 thrusters are currently what the Water Center has for an old model of the boat. The Blue

Robotics T200 thruster will be able to meet the specification that the boat should move at 1-3ft/s. Using the CAD files for the hull of the boat, an estimate of the weight was calculated. The total volume of the hull was found to be 9,381 cm^3. To be safe with the calculation of the hull's weight, the highest density 3D printed material was used which was 2.16g/cm^3. Multiplying these two values results in a total weight of 20.264kg. This equates to an estimated weight is 44.67lb. The total weight of the hull and other components turns out to be about 52.18 lb. The maximum thrust of the T200 motors is 11.6lb of force [3]. Assuming the boat will accelerate to the desired speed of 3 ft/s or 0.9144 m/s within 1 second, the acceleration will need to be 0.9144m/s/s. Using the equation F = m*a, the force needed to accelerate a mass of 52.18lbs or 23.67kg at a rate of 0.9144m/s/s would be 21.64 Newtons or 4.86lbf. This is well within the capabilities of the T200 motor.

Figure 6: T200 Thruster

The Blue Robotics ESCs will be able to control the motors since these ESCs are specifically designed to control T200 or T100 motors. These ESCs are what the Water Center is currently using for their model of the boat. Here is a performance chart that displays the thrust of the T200 motor with different operating voltages while using the Blue Robotics Basic ESC R3 [3]. From the chart the thrust is highly adjustable using the Basic ESC R3 and adjustments to the operating voltage.

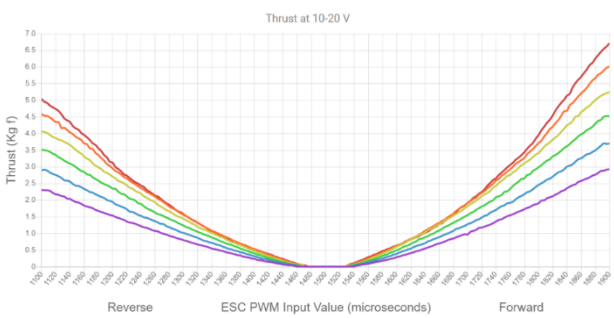

Figure 7: T200 Thruster Performance Chart

Figure 8: Blue Robotics Basic ESC

The Raspberry Pi 4 is necessary to meet the specification that the boat will be autonomous. The Raspberry Pi 4 will have ROS installed and the appropriate drivers necessary to process the GPS data from the simpleRTK2b board. The Raspberry Pi 4 will be capable of controlling the ESCs based on the documentation for both components. There are designated pins on the Raspberry Pi that can produce a PWM signal which is the required input signal for the ESCs [4][5]. Also, the voltage output of a Raspberry Pi 4 is 3.3V which is within the 3.3V-5V range required for the operation of the ESC [4][5].

Figure 9: Raspberry Pi 4 B

The simpleRTK2b long range kit will allow the project to meet the specification that the boat will collect bathymetric data and that the boat will be autonomous. This kit will allow the Raspberry Pi to know its latitude and longitude to within a few centimeters [6]. The boat will be driven around the perimeter of the lake before the sonar data collection begins. The GPS data for the perimeter of the lake will be processed by an ROS program in the Raspberry Pi to create a path that the boat will travel while it is collecting bathymetric data. Centimeter level accuracy is necessary to produce sufficient data to create bathymetric maps. The simpleRTK2b LR kit will be able to communicate NMEA protocol data messages that contain the GPS coordinates [6]. This kit provides two simpleRTK2b boards. They will be set up in a "base and rover" configuration in which one board will be on the boat and one will be in a fixed position with a clear view of the sky. This allows the position of the boat to be calculated to within a few centimeters. This is necessary for the generated path to be optimal for sonar data collection. The GPS data is transferred between the boards using two antennas and two XBee radio modules [6]. The range of this kit is 10 km which exceeds any limitation set by the controller or other systems [6].

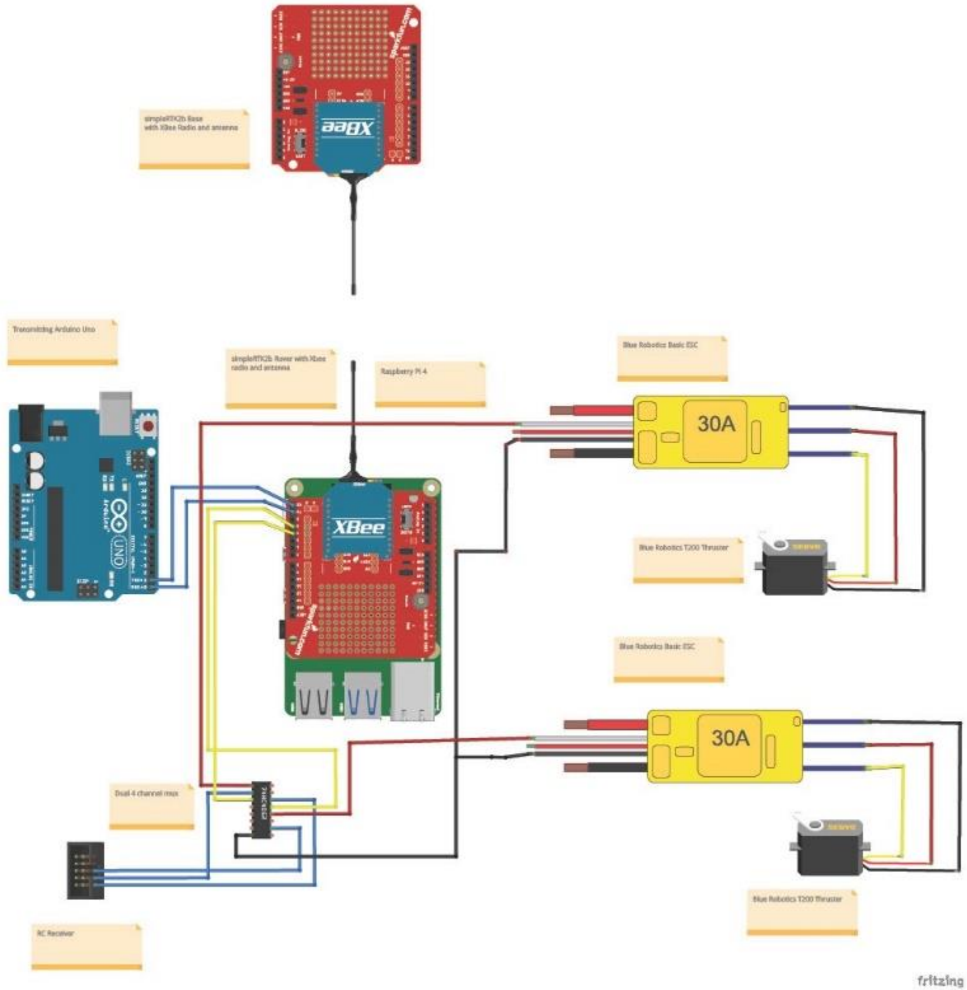

Figure 10: Full Navigation Schematic

Figure 11: SimpleRTK2b Long Range Kit

The 4-channel mux will assist in meeting the specification that the boat will have an abort function. In case of an emergency, the user will flip a switch on the Spektrum RC controller. This will send a signal to the Raspberry Pi which will toggle one of the select lines on the mux [7]. The output of the mux will switch from the Raspberry Pi's motor control signal to the RC receiver's motor control signal. The simpleRTK2b adapter for the Raspberry Pi 4 will be used for its intended use case which is to connect a Raspberry Pi 4 to a simpleRTK2b board [8].

C. *Navigation System Changes*

The original design of the navigation system included a microcontroller, steering mechanism, manual control, mapping software, obstacle avoidance system, and a GPS system. While most of these components and concepts remained, the functions of each component were changed. For the microcontroller, we ended up using a Raspberry Pi 4b with 4 GB of RAM. The intended functions of the Raspberry Pi included processing GPS data, sending GPS data to a mapping software on an on-land PC, controlling the thrusters to keep the boat on the generated path, and to avoid obstacles. Of these functions, the final product processed GPS data and controlled the thrusters. We decided that obstacle avoidance would be taken care of by switching the boat to manual control and driving around the obstacle. The original plan for generating the path was by recording the perimeter of the lake and having the Raspberry Pi process the data and generate a zig-zag path across the lake. In the end we decided that software such as Ardupilot could be used to generate the waypoints of the path and the Raspberry Pi will handle the rest. Switching between manual and

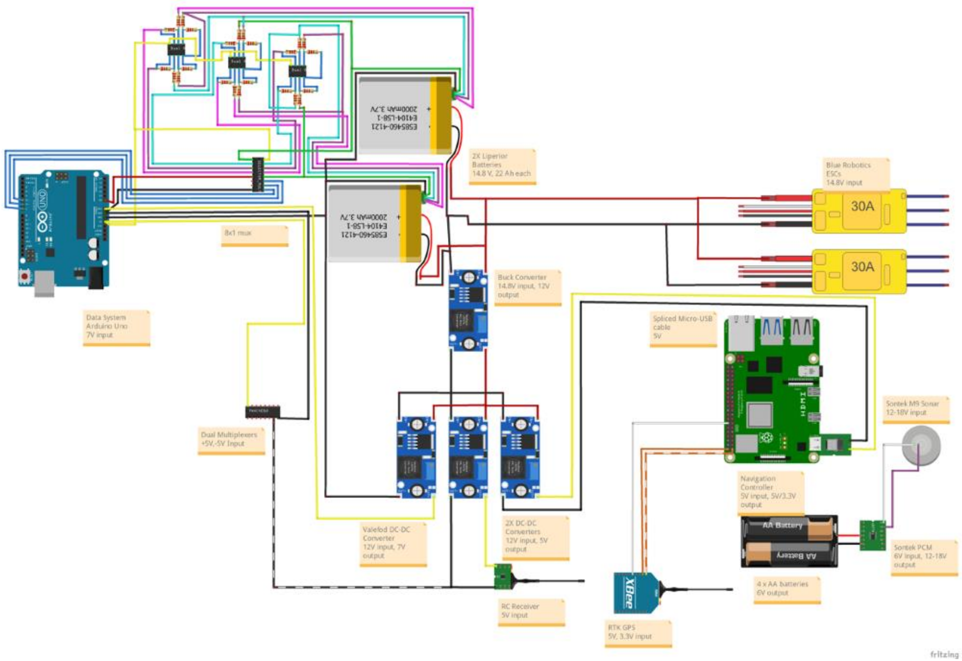

Figure 12: Initial Power System Schematic

autonomous was originally going to be done by sending a signal to the Raspberry Pi through the remote control. Then Pi would then control the select lines of an analog mux that switched which device was controlling the thrusters. The use of ROS to create the autonomous program was scrapped as well due to issues with installation and usability of Ubuntu Mate. We ended up using Raspberry Pi OS which is not compatible with ROS.

D. *Power System Initial Design:*

Understanding the power requirements for each individual component was critical to the decision on battery selection. A compromise was necessary to balance the physical size and weight of the battery or batteries in order to keep the boat afloat and maneuverable, with the voltage and current it could supply and the period of time it could last. It was determined that two 14.8 V, 22000 mAh batteries were the best choice. They weigh 3.75 lbs each, which is negligible compared to the weight of the boat and its buoyancy. Connecting the two batteries in parallel extends the time that the given capacity can last, as the electronic systems draw power from each boat evenly, reducing the drain on a single battery by 50%.

These batteries are needed to provide the appropriate voltages to the Arduino, Raspberry Pi, and RC receiver. To address this, a buck converter was implemented on each component with the batteries as the input. The output of a buck converter is directly proportional to the input by what is known as a duty ratio. If the input voltage of a buck converter was to increase, the output was also decrease at a proportional rate, and if the input voltage of a buck converter decreases, the output also decreases. The input of the buck converters needed to maintain a constant voltage even though the batteries' voltages would eventually decrease. A buck-boost converter was then designed into the circuit to provide a constant 12 V to the buck converters.

Both buck and buck-boost converters output voltages with some ripples. These ripple voltages described in their datasheets were too large and would damage the components if left as is. To rectify this issue, filters were placed at the outputs of all the converters.

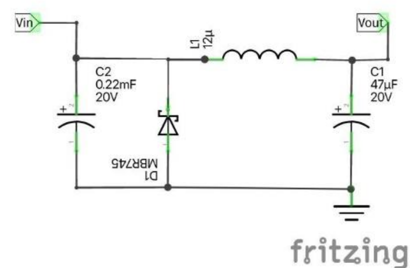

Figure 13: Buck Filter

Measuring the batteries voltage output was necessary for monitoring the capacity, since the battery voltage must be within an optimal range for the ESCs to operate. If the battery drops below this voltage, the operation will be less than optimal. The voltage measured by the Arduino Uno will be compared to the optimal voltage. If it is less than the optimal voltage, the Arduino Uno will send a signal to the Pi. The Pi will change the course of the boat to the starting point so that the user can recharge the batteries. The Liperior batteries consist of 4 cells that are each 3.7 V. The voltage of each cell needs to be above 3.5V to be within optimal operating voltage. If the batteries drop below this voltage, there is a risk of damaging the batteries. The batteries have a balance plug that is connected to each of the cells in the battery. These were used to measure the voltage of each of the cells. The Arduino Uno compares each of the 8 battery cells voltages to see if they are above 3.5 volts. This was done using an 8x1 multiplexer and differential amplifier circuits. The Arduino Uno is only capable of measuring voltages of up to 5

V. Since the batteries are connected in series, the output voltages from the balance plugs were predicted to be 3.7 V, 7.4 V, 11.1 V, and 14.8 V. There were going to be 6 differential amplifier circuits. The team planned to use 3 dual op amp ICs. Each circuit used would have consisted of four 1.0 kΩ resistors. The output would have been the difference between two of the battery cells. An 8x1 mux will be used to multiplex the voltages. The Arduino controls the select lines and "loop" through the voltages and measure each battery cell. The multiplexers and the dual op-amps were to be powered by the same 5V pin that is powering the other mux. The mux needs between 4.5 and 5.5 volts to power it.

The thrusters have a maximum current draw of 20 A at 14 V each, when operating at full throttle. The required thrust calculated to maintain the specified boat speed was approximately 4.86 lb*f or 2.20 kg*f. The thrusters can provide this amount of force while drawing approximately 6 A at 14 V. This was assumed to be the average current required from a single thruster. The Raspberry Pi 4 B itself has a maximum current draw of 3 A and 1.2 A for any peripheral components, such as the GPS device. The Arduino board has a wide range of current draw depending on the purpose of operation, with the maximum being approximately 100 mA. The RC receiver, when the boat is operating in manual mode, will allow the thrusters to draw the same amount of current that they would draw operating in autonomous mode. The maximum continuous current draw for the receiver itself is 3 A. Then the total average current draw of all systems during normal operation is approximately 21.87 A. The current draw may have been more or less, depending on the throttle of the motors, data collection, and data transmission. Each Liperior battery has a capacity of 22 Ah. When connected in parallel, the combined capacity will be 44 Ah, which will provide a runtime for the boat of approximately 2 hours. This met the Water Center's desired specification to have the autonomous boat operate for at least 1 hour.

E. Power System Changes:

Once the team received the desired components, circuit construction and analytical testing were done by using solderless breadboards. The buck converters were sufficient in outputting the required voltage ranges of operation to the Raspberry Pi, Arduino, and RC receiver. The filters were successful in reducing the ripples of the buck converters from 60 mV to approximately 30 mV. When testing the buck-boost converter, however, the output ripple was significantly larger than predicted. The designed filter's capacitors were then modified in a series of analytical testing to reduce this ripple, but testing proved unsuccessful. The team then backtracked and reviewed the behavior of the buck converters that were received. The output of a buck converter was set to 5.0 V, and the input was set to 14.8 V on a DC power supply. This input was then changed to determine how the output voltage would be affected. It was concluded that the buck converters kept a constant output voltage if the input was at least 1.0 V greater than the output. The team then concluded that the buck-boost converter was not necessary and removed it from the power subsystem.

Once the batteries came in, the balance plug could be tested. It was found that all 4 cells of each battery would need to be implemented into the differential amplifier circuits. Instead of using 3 dual op-amps, the team decided to use 2 quad op-amp

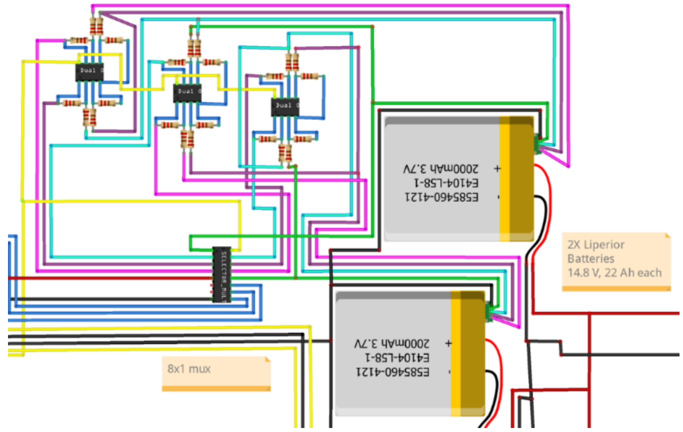

Figure 14: Battery Monitoring Circuit

circuits with 4 equal resistors with each individual circuit. Another issue arose when observing the output pins of these circuits. Most of the pins would output accurate readings, but some pins would have voltages ranging from 5-10 V. This was due to the lack of negative voltage supply that the team initially thought was unnecessary. A temporary solution was to use two 9 V batteries with one of the polarities reversed. This was not optimal for the final design, for teams would not be able to realistically monitor the capacities of the 9 V batteries. This was then addressed by implementing an inverter IC. The input of the IC is the Liperior batteries. The positive output is connected to V^+, and the negative output is connected to V^+. Testing then verified that the voltages of the outputs were consistent with the individual battery cell readings that are displayed on the battery chargers.

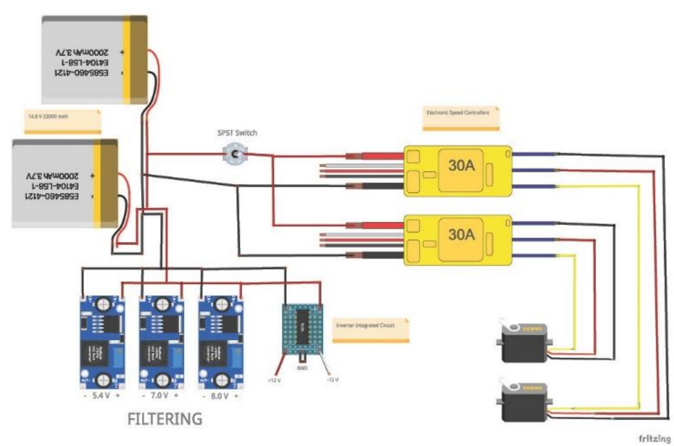

Figure 15: Final Power System with Dual V+ and V- Supply

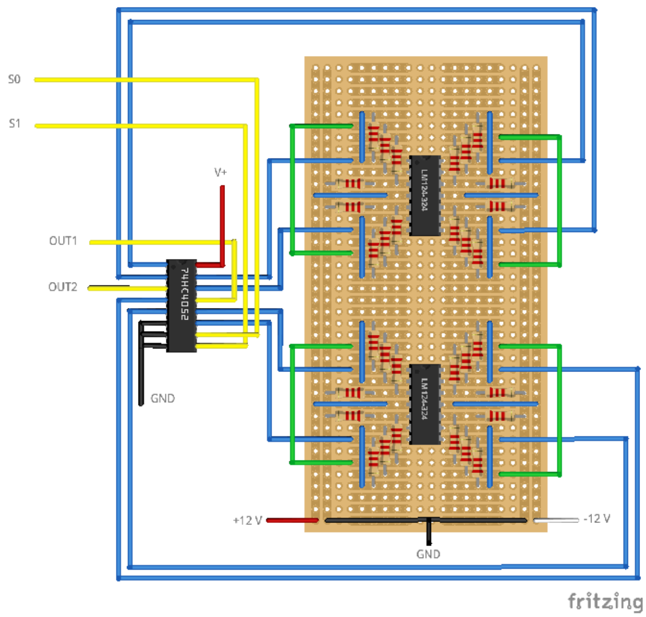

Figure 16: Updated Battery Monitoring Circuit

F. *Transmission System Initial Design:*

To meet the requirement to observe data in real-time as it is collected, a system has been designed to collect measured data and transmit it to a local computer system so it can be viewed by the user. The design of this system has evolved over the course of this project according to new knowledge and changes of ideas.

The original conceptual design of the data transmission system was split into two main parts: the base station section and the onboard section. The design for the onboard section included a microcontroller to collect data from the onboard systems and an RF transmitter to send the data to the base station. The design for the base station section included a microcontroller and an RF receiver. The microcontroller was connected to the base station PC with a USB connection, and it was designed to be used to operate the RF receiver and communicate the received data to the PC. Also, the designs assumed the SonTek HydroSurveyor sonar system would be used and included the SonTek Power and Communications Module (PCM) as the primary method for the transmission of hydrographic depth data.

When first selecting components for the conceptual system, a pair of 433 MHz RF modules was selected for the transmitter and received components. At the time of the detail design, the decision was made to use 2.4 GHz modules instead. The first reason this change was made was because in the United States the 433 MHz frequency band is restricted by the FCC to only certain licensed devices. Secondly, although 433 MHz frequency transmitters offer much greater range for the same transmission power, 2.4 GHz devices can support much greater message signal bandwidths. In addition, the connection for the onboard GPS module to the onboard microcontroller was intended to be from the microUSB connection on the simpleRTK2B GPS module to the USB-B port of the microcontroller. This connection was later redesigned to be made via a serial UART connection between the TX/RX pins of the GPS and microcontroller.

G. *Transmission System Updated Design:*

The nRF24L01+ PA LNA wireless transceiver modules were selected for the transmission and reception of collected data. These modules support programmable output power and have a low power consumption. Available output power levels include 0 dBm, -6 dBm, -12 dBm or -18 dBm, and the module only consumes around 12 mA during transmission at 0 dBm. All the parameters can be configured through a Serial Peripheral Interface (SPI). This type of interface uses the concept of Master and Slave. In this case the microcontroller was the Master, and the transceiver was the Slave. Also, this version of the nRF24 module uses an external antenna, a power amplifier, and a low-noise amplifier to increase the transmission range from 100 meters to up to 1000 meters, a distance that satisfies the range specification.

Both the base station section and the onboard section will use a microcontroller and an nRF24L01+ PA LNA wireless transceiver module with the same connections. The schematic detailing these connections can be seen in the figure below. The radio frequency receiver module receives a message signal encoded with collected data and will decode the signal to extract the measurement data. The microcontroller was designed to control the radio frequency modules that receive an encoded radio wave. Once it has been decoded, the microcontroller will transport the received measurement data to a connected computer via a USB-B to USB-A serial connection. In addition, the SonTek radio dongle will be connected to a USB port on the computer and will receive measurement data collected by the M9 and transmitted from the PCM.

The Arduino Uno was selected to be used as the microcontroller for each section because of its ability to handle various baud rates, utilize analog and digital input/output header pins, and communicate with other devices over different digital communication protocols simultaneously. Specifically, the Uno can communicate using an SPI, making it compatible with the selected RF transceiver for data transmission. The GPS module was confirmed to meet the specifications for a GPS compatible with the Hydrosurveyor M9. The specifications were as follows:

- Data output rate between 1-10 Hz (10 Hz recommended)
- Baud rate set to 38400
- NMEA output sentences: GPGGA and GPVTG (if used for position) and/or GPHDT (if used for heading).

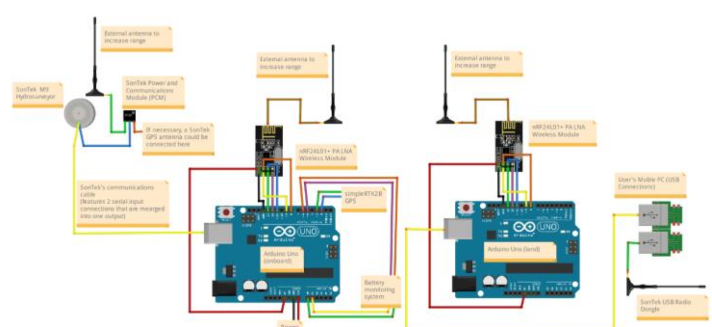

Figure 17: Transmission System Design Schematic

VI. SAFETY

With any autonomous system, the first major concern is how the system interacts with humans and other living creatures. In the case of an autonomous vehicle, the concern is that the vehicle must be able to stop operation when its actions are detrimental to life or limb. For a hydrographic survey vessel, there may be people or animals swimming in the lake or river that it is scanning, and an autonomous HSV must not provide risk to those people or animals it may encounter.

The design of this specific boat relies on the responsibility of the on-site technician to recognize and mitigate any potential risks. This autonomous boat is equipped with a manual override system, which immediately stops the boat and gives control to an operator when activated. This system is activated by a switch on a Radio-Controlled device that also functions as the controller when in manual operation. This autonomous system is not designed to be operational without supervision. A technician must be present while the boat is in operation and must be aware of the water body and the presence of any humans or animals or other non-living obstacles in order to manually drive the boat in areas where human precision is required.

VII. RESULTS

The Water Center had a specification that the autonomous boat must be able to successfully drive the entirety of Cane Creek Lake (0.2266 km^2) in one run. The base station was set up at the dam of Cane Creek Lake and the boat drove along the dam 56 times at a constant speed of 3-5 ft/s. The speed was determined by using the speeds used by the Water Center to obtain accurate depth measurements. The boat was kept close to the dam due to the boat not being completely waterproof. The inside of the hull was not coated in fiberglass, which allows water to get into the hull and weigh it down. If the boat was to have an issue with water getting in the hull, the team could easily get the boat out of the lake and prevent any damage to the circuits. The battery monitoring circuit showed that the voltages of each cell were 4.2 V, and when the batteries are considered "low", the voltage is approximately 3.5 V. The boat successfully drove the distance determined to be equal to the area of Cane Creek Lake. The team then determined how long the boat could run, and it was concluded that the boat could remain fully operational for approximately 3 hours. The test came to an end when the team noticed that the thrusters started to slow down after full thrust, thus becoming a risk if the boat was hypothetically in the middle of the lake. Additionally, the battery monitoring circuit was transmitting data to the land base, which showed that the cells in the batteries had a voltage of 3.7 V.

```
3: 3.82   3.82
4: 3.82   3.90

1: 3.85   3.88
2: 3.85   3.82
3: 3.86   3.83
4: 3.81   3.92

1: 3.88   3.88
2: 3.85   3.83
3: 3.84   3.82
4: 3.81   3.92

1: 3.86   3.89
2: 3.87   3.81
3: 3.83   3.83
4: 3.81   3.90

1: 3.88   3.89
2: 3.87   3.83
3: 3.86   3.83
4: 3.81   3.92
```

Figure 17: Live Data Stream of Battery voltages

To test the product's ability to transmit collected data wirelessly, the base station was set up at the dam of Cane Creek Lake, and the boat drove away from the base station until accurate data could not be obtained. The transmitter was operating with a transmission power of -6 dBm. The final distance the Arduino was able to receive data was approximately 1200 feet or 365 m. The team also tested the range on campus because the hull was taken to get coated in more fiberglass. One team member stood outside of Foster Hall on Tennessee Tech's campus with the laptop and land radio module, while two other team members carried the rover radio module down University Drive until the land module could not receive any data. The transmitter was operating with a transmission power of -6 dBm. Data transmission slowed down once the team members crossed North Willow Avenue going towards Tech Village. The last received transmission was at about 596 m.

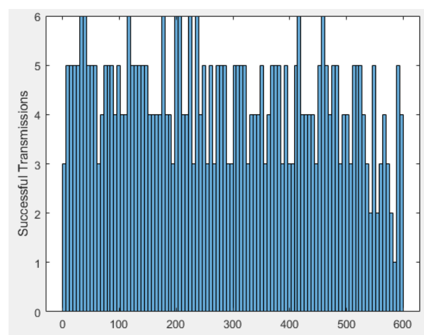

Figure 18: Number of Transmissions vs. Distance

Figure 19: Experiment Location and Transmission Distance

Using two simpleRTK2b receivers, the accuracy and precision of the recorded GPS data was observed. The receivers were in a base and rover configuration. Using just the rover receiver, the GPS coordinates were accurate to within 1.4 meters and precise to within 10 centimeters. Using the base station and the rover, the accuracy was around 1.9 meters, and the precision was 2 centimeters. The base station and rover were tested again with the known location of the base station uploaded to the base station receiver. The accuracy was withing 2 cm and the precision was within 3 cm.

Time	Latitude	Longitude	Distance From Previous Point (Meters)	Distance Error (meters)
1	36.17530383333334	-85.50685999999999	0.11514982545816933	1.367
2	36.17530466666666	-85.50685966666667	0.09737298196112842	1.387
3	36.17530583333333	-85.50685966666667	0.1297274148812091	1.461
4	36.175307	-85.5068595	0.13058711471873602	1.53
5	36.17530766666666	-85.5068595	0.07412995090921236	1.58
6	36.17530816666667	-85.5068595	0.05559746397199674	1.618

Figure 20: Rover Data

Time	Latitude	Longitude	Distance Traveled	Distance Error
1	36.174919	-85.50587817		1.914406463
2	36.174919	-85.50587817	0	1.914406463
3	36.174919	-85.50587817	0	1.914406463
4	36.17491883	-85.50587817	0.018532489	1.931961436
5	36.17491883	-85.50587817	0	1.931961436
6	36.174919	-85.50587817	0.018532489	1.914406463

Figure 21: Base and Rover Data

Time	Latitude	Longitude	Distance Traveled	Distance Error
1	35.15380617	-84.934224		0.133968694
2	35.15380633	-84.934224	0.018532489	0.116705191
3	35.1538065	-84.934224	0.018532488	0.099897442
4	35.15380667	-84.93422417	0.023938386	0.076667046
5	35.15380683	-84.93422417	0.018532488	0.060091975
6	35.15380683	-84.93422417	0	0.060091975

Figure 22: Base and Rover (Preset Location) Data

VIII. TIMELINE

The goals that the team chose to pursue, and the order of priority were determined by the necessity of each subsystem to the boat. As illustrated by the Gantt chart, some of the major goals for the project were to identify specifications and constraints, complete a detailed design and analysis for all subsystems, implement and integrate all subsystems, debug all software, and test the final prototype.

The order of these goals presents a linear progression of the project and makes it a logical projection for the completion of goals. There was also some room allowed for overlap across goals so that progression was not prevented by a delay in completion of previous goals. Once the specifications could be identified for a particular subsystem, design could begin for that subsystem without waiting for all specifications for all systems to be finished. In the case of implementation and integration, software could be written and tested at the same time other circuits were being built and integrated into the overall design without interfering with the software or other systems.

Specifically, the team had goals to build a self-contained, battery-powered system, which could plot a path around a body of water, navigate that path autonomously, transmit its GPS coordinates and its battery levels to an on-shore computer, and would have the ability to switch into manual operation at the user's discretion. While the specifications came directly from the Water Center, the goals were set to meet these specifications in the most efficient manner for the team to have a final product. Subsystems with specifically desired features were addressed with higher priority, and additional features were addressed as there was opportunity to include them.

Tasks were planned based on these major factors: predicted duration, urgency, and difficulty. The navigation subsystem, for example, was anticipated to take a lot of time to design and implement due to potential troubleshooting and experimental analysis. Furthermore, it was crucial that autonomous navigation have a solid foundation, for this was the most prioritized specification made by the Water Center. These reasons are what made the team plan to have navigation take up the most time on the Design Phase 2 Gantt chart as shown below. Other subsystems were expected to have a shorter time of completion, so these were not as urgent as the navigation system. These systems could also be tested in the lab at any time, whereas the navigation system had to be tested in a body of water when the weather is in optimal condition.

The team would have one meeting a week discussing progress on individual tasks as well as determining the next realistic step for the project's construction. Members informed the other teammates if they believe that the individual tasks could not be completed by the next deadline. The team provided assistance if necessary and did not hold these moments against one another. If a group task was not completed by the desired deadline, the team would continue to work on that task as well as starting ones of less difficulty but of equal importance.

When distributing tasks amongst the team, the members discussed each other's strengths and weaknesses. Joshua Herrera excels in programming languages such as Python, C, and C++. The Water Center required that the boat could send bathymetric measurements and GPS coordinates to the land base in real time. Levi Daniel possessed the skill of working with radio frequency devices and wireless communications. Regarding hardware, the circuits on board the boat must follow all safety practices to ensure that there are no accidents when the boat is in a body of water. Carter Ashby excels in circuit construction and component application, soldering, etc. Andrew Alley is also proficient in circuit design construction as well as having insight on power electronics.

As previously mentioned, a commercially available HSV is priced around $70,000. For this project, the price is greatly reduced due to the manual assembly of all systems, the ability to

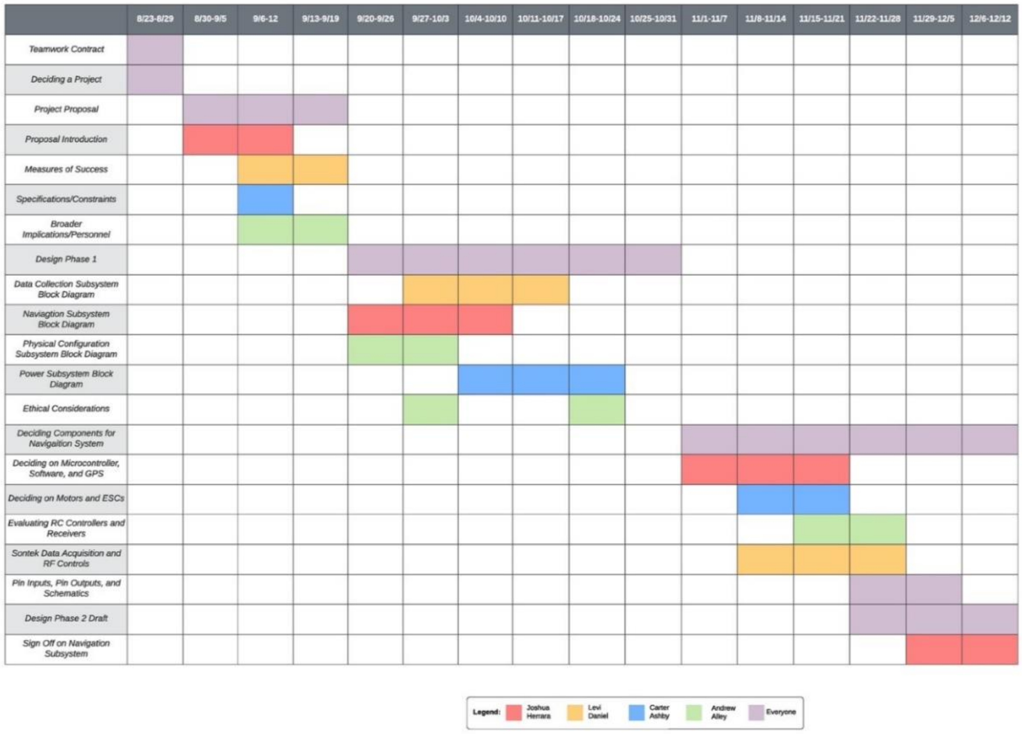

Figure 23: Gantt Chart

pay only for specific components, and the use of open-source software. The total cost of everything included in this project is approximately $2,300, as shown in Bill of Materials in Figure 24. This is well below the starting cost for a commercial unit.

IX. Ethical Responsibilities

Throughout the duration of this project, the team always took into consideration economic and environmental ramifications regarding design as well as troubleshooting and testing. The Water Center has emphasized the importance of keeping this project cheaper than commercially available options. Not only would this save the Water Center money, but the project would allow for easier replicability for any communities interested in using this design to take their own hydrographic surveys. There were scenarios where the team needed to either purchase new components or replacements due to damage, so the team tried to maintain a balance of low-budget and overall performance when making these purchases.

In order to do effective experimental analysis, the boat needed to be tested in a controlled body of water because the T200 thrusters use the water to lubricate the plastic brushings as well as cool the motors. The team used Cane Creek Lake as the control environment due to it being within proximity to campus. Additionally, the Water Center requested that the boat be able to cover all of Cane Creek Lake in a single run. There was going to be no realistic scenario where the lake would be empty, so the team had to be cautious when doing testing. The team made sure to keep the boat away from any people to not cause any disturbances and prevent any injuries. Furthermore, the boat was always operated away from any wildlife, and if any animals were to get close, the user would drive the boat away and bring it back to land base.

X. Lessons Learned

There were a few things that did not go as planned. These include issues regarding the circuitry of the battery monitoring system, programming the Raspberry Pi, receiving good GPS data, and the structural integrity of the hull.

The battery monitoring circuit had several problems to begin with. The first being that the output voltage of the balance plug on the LiPo batteries was not what we expected. We designed the battery monitoring circuit using differential amplifiers that were meant to output the difference between two pins on the balance plug. However, it was designed to find the difference between positive voltages, and 3 of the 5 pins on the balance plug output negative voltage. This required a redesign of the battery monitoring circuit, this time using quad op-amps since more voltage differences were required than previously thought. Another issue was found regarding the supply voltage of the quad op-amps. There was an issue with the way the op-amps were receiving power that made it difficult to output the correct voltages. The op-amps needed isolated power supplies to work properly. To solve this problem, we used a dual negative and positive supply circuit.

Programming the Raspberry Pi proved to be more difficult than we anticipated. The first issue was attempting to install Ubuntu Mate to the Pi. It took multiple attempts to determine that the problem was that the Pi was not able to output a display to the monitor we were using. For weeks, it was only possible

Part Number	Description	Manufacturer Name	Quantity	URL to Product	Cost per Item	Total cost of Quantity
SKU#: ISDT-D2M2	Battery Charger	ISDT	1	https://rcbattery.com/isdt-d2-	$120	$120
NRF24L01	3PCS NRF24L01+PA+LNA RF Transceiver Module with SMA Antenna	Aideepen	1	https://www.amazon.com/dp/	$17	$17
A000066	Arduino Uno Rev3 (Microcontroller)	Arduino	2	https://www.amazon.com/Ard	$26	$52
ASIN: B01N9IP8LF	USB A to USB B cable	Directsrc	1	https://www.amazon.com/Dat	$15	$14.95
VS-MBR735-M3	Schottky diode	Vishay Semiconductors	4	https://www.mouser.com/Pro	$0.94	$3.76
LP220004S12	Lithium-Polymer Battery (14.8V, 22000 mAh)	Liperior	2	https://rcbattery.com/liperior-	$160	$320
VA-0189	DC-DC Converters (2-pack)	Valefod	2	https://www.amazon.com/Con	$14	$28
UWE-Q12	Isolated DC-DC Converter	Murata Power Solutions	1	https://www.mouser.com/Pro	$62	$62
CD74HCT151E	8x1 Multiplexer	Texas Instruments	1	https://www.mouser.com/Pro	$0.62	$0.62
LM358P	Operational Amplifier	Texas Instruments	3	https://www.mouser.com/Pro	$3.99	$3.99
CFR-25JR-52-1K	1 kOhm resistor	YAGEO	24	https://www.mouser.com/Pro	$0.047	$1.18
33785	33 microFarad capacitor	Jameco	4	https://www.jameco.com/z/TN	$1.95	$7.80
30496	220 microFarad capacitor	Jameco	4	https://www.jameco.com/z/R2	$0.29	$1.16
B82141A1123K000	12 microHenry inductor	TDK	4	https://www.mouser.com/Pro	$0.46	$1.84
	Clear Waterproof Plastic Box 32L	At Home	1	https://www.athome.com/clea	$15.99	$15.99
T100-P-BRACKET-R1-RP	T200 Thruster Brackets	Blue Robotics	2	https://bluerobotics.com/store	$6.00	$12.00
101983	Lexel Rubber Sealant	Sashco	1	https://www.paintsupply.com/	$5.95	$5.95
T200-THRUSTER-R2-RP	T200 Thruster	Blue Robotics	2	https://bluerobotics.com/store	$180	$360
BESC30-R3	Basic ESC	Blue Robotics	2	https://bluerobotics.com/store	$30	$60
SPMR6650	Spektrum DX6e and AR620 Reciever	Spektrum	1	https://www.amazon.com/Spe	$200	$200.00
	simpleRTK2b LR starter kit (headers soldered)	Ardusimple	1	https://www.ardusimple.com/	$818.24	$818.24
AS-ADP-ARDUINO-TO-RPI-00	Raspberry Pi adapter for simpleRTK2b	Ardusimple	1	https://www.ardusimple.com/	$20.45	$20.45
	Raspberry Pi	Raspberry Pi	1	https://www.amazon.com/Ras	$110	$110
13119	Dual 4-channel Analog Multiplexer	Jameco	1	https://www.jameco.com/z/CD	$0.49	$0.49
AS-ACC-ARDHEADERS-00	Headers	Ardusimple	1	https://www.ardusimple.com/	$5.67	$5.67
	TPS5430 Positive Negative Dual Output Module	Dongker	1	https://www.amazon.com/dp/	$17.99	$17.99
						$2,261.07

Figure 24: Bill of Materials

to display the Ubuntu Mate desktop through one monitor. This was solved by altering the configuration files of the Pi so that it could properly send an output through HDMI. Another issue with programming the Pi involved using C. It was intended to use C in ROS to create the code for the autonomous navigation. The issue with that was that the RTK (Real Time Kinematic) GPS could not interface properly with the Raspberry Pi using C. After multiple attempts, we resorted to using Raspberry Pi OS instead of Ubuntu Mate to program the Pi. Also, we switched to using Python because it was found to be much simpler. We were unable to use ROS as well since it is not compatible with Raspberry Pi OS.

One of the significant issues that arose when trying to test the autonomous code, was that the Raspberry Pi would randomly receive invalid GPS data that could not be parsed or decoded. To solve this problem, the code was adjusted so that it would account for every case of invalid data. This included checking whether the data was encoded properly, whether there were enough data fields after it was parsed, and whether the latitude and longitude data was numeric.

The structural integrity of the hull was a big hindrance to the progress of the project. The hull was 3d printed by the TN Tech Water Center and coated with epoxy resin to waterproof the outside of the hull. This worked as planned, however, the inside of the hull was very porous and absorbed a lot of water when testing the boat. In the center of the middle piece of the hull, there is a hole for the River Surveyor M9 sonar sensor. While testing the boat, we did not have anything in the hole, so nothing was stopping the water from seeping into the 3d printed material. This caused the boat to nearly sink after a certain amount of testing. This limited any more tests that could be done with the autonomous navigation code until the inside of the hull was waterproofed.

A word of advice for future students would be to never assume anything works without testing it. It doesn't matter if it is an extremely basic concept or design. It should be tested. The main thing that we would have done differently is to start the project using python and Raspberry Pi OS.

XI. Conclusion

The objective of this project was to design and construct an autonomous hydrographic surveying boat for the Tennessee Tech Water Center. The boat must transmit depth measurements to a land base in real time as well as switch between autonomous and manual control. Additionally, there needed to be an abort button/switch if the data being transmitted was inaccurate and obstacle avoidance for when the boat is in autonomous control.

The team was successful in creating a boat that can change between autonomous and manual control by flipping a switch on a remote controller. Additionally, the boat can accurately measure and transmit GPS and battery voltage data to a land base from approximately 600 meters away. The observed final operational time of the boat is 3 hours. The boat also successfully measures its current location with accuracy and precision in the centimeters by using simpleRTK2b GPS receivers, and a planned path can be created and implemented into the Raspberry Pi for autonomous navigation.

There have already been some discussions regarding potential ideas for future work and improvements to the autonomous hydrographic boat. The Water Center mentioned that the boat should be able to operate if the boat remains within the land base's field of sight. The data, however, can only be successfully transmitted from approximately 600 meters from land base. The team speculates that this transmission range could be improved if the transmission bit rate was reduced on the Arduino Uno and nRF24L01+ RF modules. Additionally, the Python code could be improved by allowing users to drive a desired perimeter on a body of water and have the Raspberry Pi generate a set of parallel lines for the boat to traverse.

XII. Individual Statements

A. *Joshua Herrera*

For this project, I used the knowledge that I learned from Electrical Engineering Lab II, Circuits II, and Digital Systems Lab. From EE Lab II, I used what I learned about operational amplifiers to test and develop the battery monitoring circuits. Basic knowledge of circuitry assisted me in determining which resistors to use to create voltage dividers for testing the inputs to the Raspberry Pi. The knowledge that I learned in Circuits II allowed me to design the unused filters that were meant to convert PWM (Pulse Width Modulation) signals to DC voltages. Digital Systems Lab helped me when designing the analog multiplexer circuits. It allowed me to determine where pull-down resistors were needed and which voltages to use for logic 1 on the select lines.

The knowledge that I gained while working on this project was interfacing with a Raspberry Pi and learning to program in Python. Using the Raspberry Pi proved to be more difficult than I anticipated. Especially when online instructions made assumptions about pre-installed dependencies that were not pre-installed. I learned how to search for very specific errors every time they appeared when working with the Pi. I have also learned how to use Raspberry Pi add-ons and external devices such as the RTK GPS receiver. It used a serial connection which required several different dependencies to be installed.

Programming in Python was the biggest thing that I have learned while working on this project. I had avoided learning it at the beginning since it was a programming language that was unfamiliar to me. However, it proved to be a simple language to learn and became a valuable tool for the development of this project.

B. *Andrew Alley*

When deliberating on design choices and constructing the boat, I was able to implement my knowledge from ECE 4630, Power Electronics. This class proved beneficial to the project, for the practical batteries available in the market were outside of the operational voltage ranges of the Raspberry Pi, Arduino, and RC receiver. Power Electronics introduced the basic knowledge of DC-to-DC converters, which made the overall construction of the power system shorter than the team had anticipated. Additionally, ECE 3060, Electrical Engineering Lab 2, and ECE 3300, Electronics 1, provided insight regarding filter design for the buck converters as well as circuit design and soldering. Electrical Engineering also emphasized the importance of troubleshooting every component and node.

The main lesson I learned during my time working on this project is that real-life applications will almost never be equivalent to theoretical calculations. I was aware of this when starting the project; however, I did not realize how critical it was to be able to adapt quickly to these failures to stay on schedule. For example, when running LTSpice simulations on filtering a buck-boost converter, the designed schematic reduced the ripple voltage such that the output voltages of the connected buck and buck-boost converters would stay within the appropriate ranges for each device. The actual designed filter, on the contrary, reduced the ripple of the buck-boost to approximately 1.2 V, which was significantly larger than predicted in the LTSpice schematic. This required the team to quickly explore other options for powering the components in order to meet a desired deadline. This happened throughout the entirety of the project, but the team was able to change plans without negatively affecting overall performance. When working on engineering projects, it is important to have a plan with a strong foundation when in the design phase. Engineers must also understand that these plans may not be effective in the construction process and need to be able to quickly adapt when these likely scenarios occur and still be able to present a project by the final deadline.

C. *Levi Daniel*

Knowledge and experience gained in the signals and telecommunications courses provided me with a foundation to build on when I was researching and designing subsystems for this project that involved radio frequency communication and GPS. Also, knowledge gained in courses related to the communication of digital systems helped me to grasp how to design a system using serial communication between two or more components. Knowledge from circuit analysis and electronics courses was used to analyze circuit schematics and troubleshoot designs. Knowledge from my computer science courses was useful when programming microcontrollers and reading and implementing other algorithms.

I have learned a lot over the course of this project. A lot of the knowledge I have gained has either been built on a foundation of already established knowledge or has reinforced an area with additional learning or experience. Knowledge gained in this way is hard to measure and is easy to overlook when looking back. Other areas of learning where I learned or improved a skill or uncovered a new topic area are easier to recall. To name a few, I learned how to analyze a radio frequency connection using Link Budget Analysis. I used this to analyze and confirm the designs for the data transmission subsystem. I learned a lot about serial communication protocols and their differences, specifically SPI protocol. This was useful when designing the data transmission subsystem and the interface with the data acquisition subsystem. I also learned a lot about best practices when testing and troubleshooting a prototype. This was learned by experience when working with the developed prototype. And finally, I learned a lot about working with a team through the various stages of the engineering process. I learned better, more efficient ways to communicate and share ideas, and I learned how to handle various unique circumstances one might encounter in a team of unique engineers.

D. *Carter Ashby*

I found that my experience in Electrical Engineering Labs provided help in designing and building circuits for specific tasks in the project. More importantly, I was able to troubleshoot problematic circuits and fix unexpected issues, which I had developed the skills to do while building and testing circuits in our lab courses. It was also helpful to have programming knowledge and experience, especially when working to code the Arduino boards. Basic knowledge of electrical components, how they behave and interact in conjunction with other circuits, power performance, and other general electrical topics were all used and necessary to contribute to a project focused on autonomy.

I learned a lot about the implementation of specific ICs. Educational courses provide only the overview of a working device or circuit, and often require overlooking specific applications or functionality. I read through many device datasheets to understand the pinouts, composition, features, specifications, and functionality of several ICs or circuit boards in order to properly implement them into our systems. This is very important to the project as we had a few issues where systems did not work properly because the pre-built circuits we were using were not integrated correctly. It quickly became clear that we should learn every detail of the ICs before trying to use them in our applications.

References

[1] "FAQs: Spektrum - the leader in spread spectrum technology," *Spektrumrc.com*. [Online]. Available: https://www.spektrumrc.com/Support/FAQs.aspx. [Accessed: 26-Jan-2022].

[2] "DX6e 6CH Transmitter Only," *Spektrumrc.com*. [Online]. Available: https://www.spektrumrc.com/Products/Default.aspx?ProdId=SPMR6650. [Accessed: 26-Jan-2022].

[3] "T200 thruster," *Blue Robotics*. [Online]. Available: https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/. [Accessed: 26-Jan-2022].

[4] "Blue robotics," *Blue Robotics*. [Online]. Available: https://www.bluerobotics.com/store/thrusters/speed-controllers/besc30-r3/. [Accessed: 26-Jan-2022].

[5] "Raspberry pi GPIO Pinout," *Pinout.xyz*. [Online]. Available: https://pinout.xyz/. [Accessed: 26-Jan-2022].

[6] "simpleRTK2B Long Range Starter Kit IP67, based on u-blox ZED-F9P," *ArduSimple*, 10-Nov-2018. [Online]. Available: https://www.ardusimple.com/product/simplertk2b-starter-kit-lr-ip65/. [Accessed: 26-Jan-2022].

[7] "CD4051BC • CD4052BC • CD4053BC single 8-channel analog multiplexer/demultiplexer • dual 4-channel analog multiplexer/demultiplexer • triple 2-channel analog multiplexer/demultiplexer," *Jameco.com*, 2000. [Online]. Available: https://www.jameco.com/Jameco/Products/ProdDS/13119.pdf. [Accessed: 26-Jan-2022].

[8] "Raspberry Pi adapter for simpleRTK," *ArduSimple*, 24-May-2019. [Online]. Available: https://www.ardusimple.com/product/raspberry-adapter-for-simplertk2b/. [Accessed: 26-Jan-2022].

[9] "Lithium-ion battery (14.8V, 15.6Ah)," *Blue Robotics*. [Online]. Available: https://bluerobotics.com/store/comm-control-power/powersupplies-batteries/battery-li-4s-15-6ah/. [Accessed: 26-Jan-2022].

[10] "5V 6A power supply," *Blue Robotics*. [Online]. Available: https://bluerobotics.com/store/comm-control-power/elec-packages/bec-5v6a-r1/. [Accessed: 26-Jan-2022].

[11] "Raspberry Pi documentation - Raspberry Pi hardware," *Raspberrypi.com*. [Online]. Available: https://www.raspberrypi.com/documentation/computers/raspberry-pi.html. [Accessed: 26-Jan-2022].

[12] "CEE-USV," CEE Hydrosystems. [Online]. Available: https://ceehydrosystems.com/products/unmanned-survey-vessels/cee-usv/. [Accessed: 20-Sep-2021].

[13] Acta.uni-obuda.hu. 2022. [online] Available at: <http://acta.uni-obuda.hu/Horvath_Pozna_Precup_81.pdf> [Accessed 3 May 2022].