ECE 445

Spring 2024

Senior Design Design Document

Automatic Ice Fishing Rod

Team 60

Luke Boelke
James Niewiarowski
Andrew Osepek

TA: Zicheng Ma

March 20, 2024

Table of Contents

1 Introduction

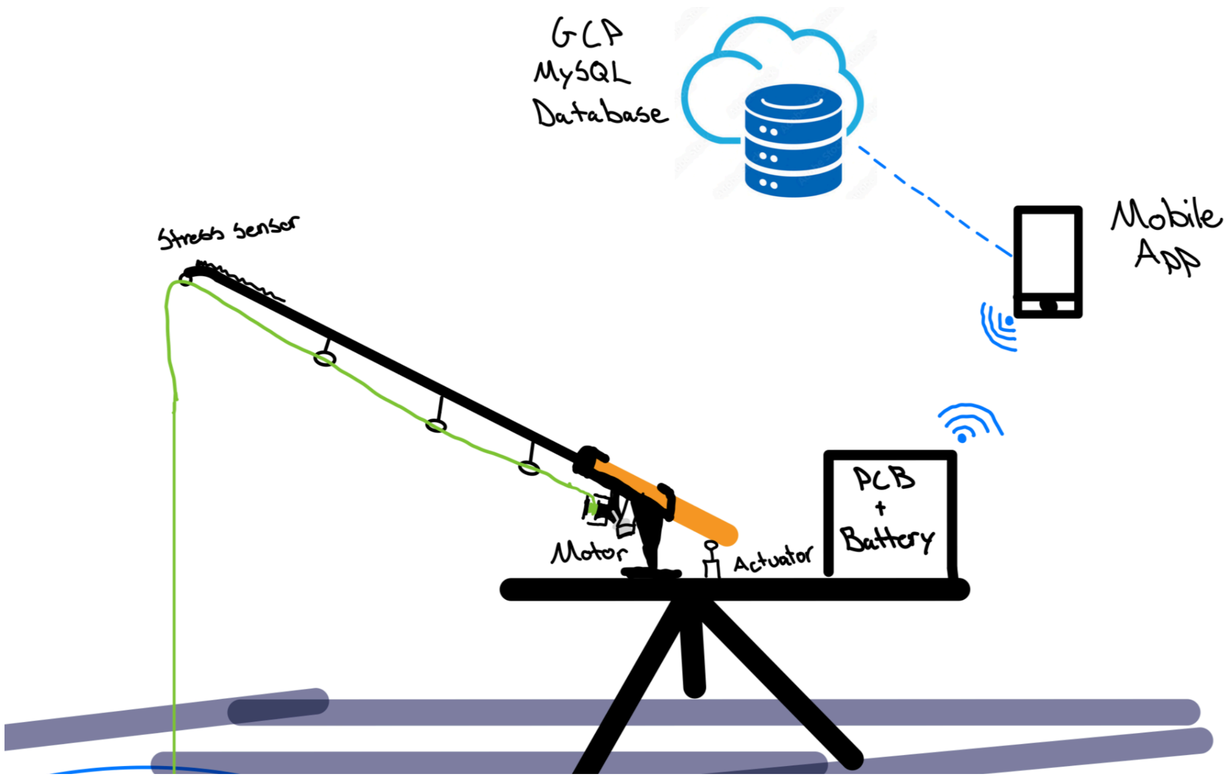
1.1 Problem

Ice fishing can be a very tedious and labor-intensive process. While it is being performed, the fisherman must dedicate all of their attention to the task at hand, constantly jigging the rod, making multitasking impossible. It must be done in a very cold environment as well, which gets uncomfortable after long periods of time. Additionally, there can be long stretches with little to no bites. If the fisherman did not have to constantly attend to the rod, these stretches of no activity would be perfect for taking a break to warm up, eat a meal, etc., but the nature of ice fishing makes this impossible.

1.2 Solution

Our project aims to create an automated ice fishing rod that eases the challenges associated with ice fishing. The user will have the ability to spool any lb-test line onto the device as with any lure when fishing. The fisherman can set the length of the line out from the reel. The fishing rod will have the ability to jig the attached lure in hopes of attracting fish. When a tug occurs at the line, the user will be alerted through an alarm and notification. A mobile app will allow the user to set preferences to the depth of the line and jigging.

1.3 Visual Aid

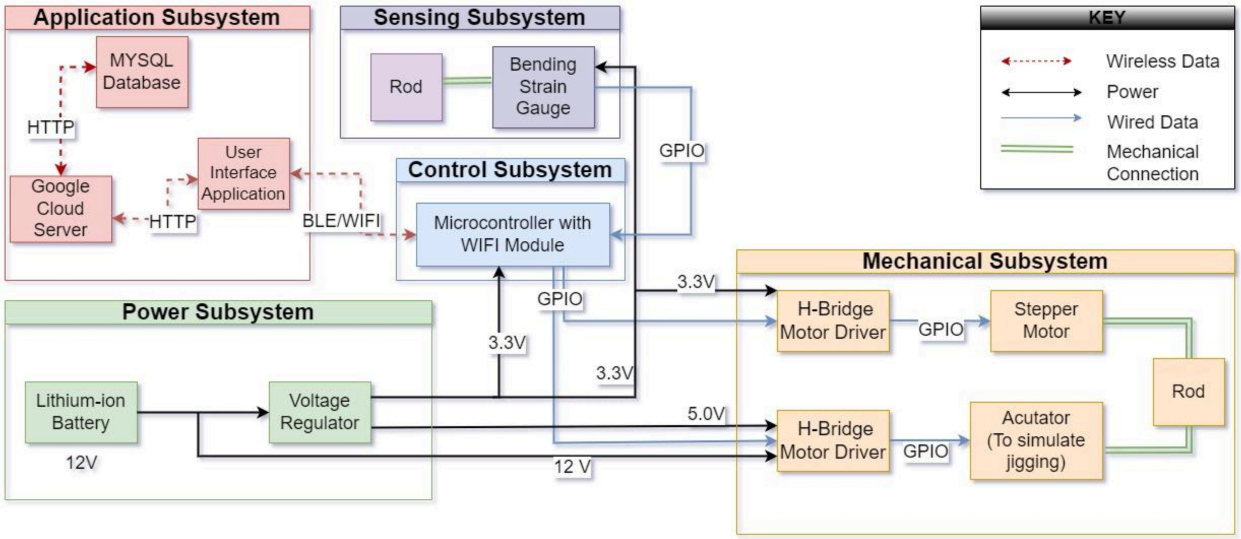


1.4 High-level requirements list

1. The user will be able to set up to 3 different jigging frequencies, a lure depth up to 50 feet (+/- 5 feet) in increments of 1 foot (+/- 0.25 feet), or enable auto reeling.
2. When a bend angle of 30 (+/- 10) degrees is detected, the jigging will halt within 5 seconds of detection, a notification will be sent to the user application, and, if auto reeling is enabled, the line will be reeled in automatically.
3. The user will be able to record their catches in the user application with 7 different data fields. Previous catch information can be viewed in the application.

2 Design

2.1 Block Diagram

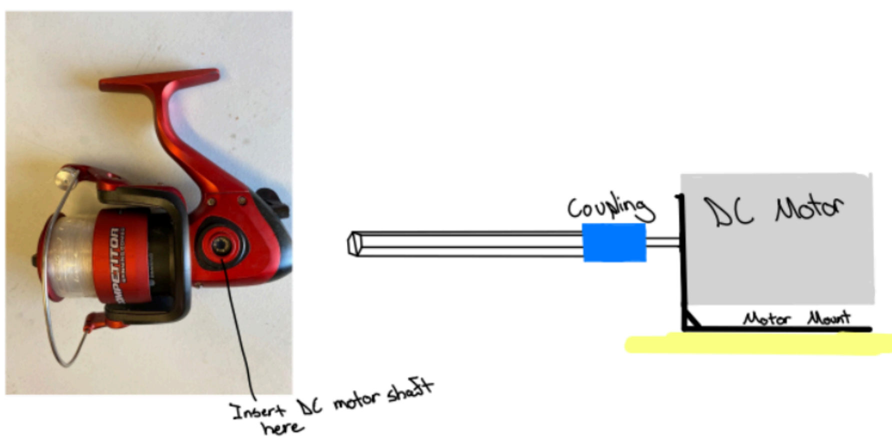


2.2 Physical Design Overview

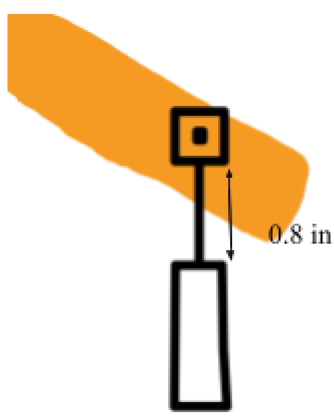
We plan to use a spinning reel rod holder for supporting our fishing rod as shown below. The base of our rod holder will be mounted to a plywood board that will hold our battery and PCB board too. The fishing rod holder has a joint that can be loosened which will allow our rod to move up and down to support our jigging feature.

For the automatic reel, we plan to modify a spinning fishing reel. A spinning reel is driven by a hexagonal shaft as illustrated in the figure below.

To connect our stepper motor to the spinning reel, we will join a hexagonal shaft to our stepper motor's shaft using a coupling. The hexagonal shaft will slide into the reel's slot where the original hand crank shaft was placed. Finally, the stepper motor will be mounted to the fishing reel. A spinning reel can spin in both directions (reeling in/out), so we can control the depth of our line by reversing the direction of the stepper motor to either raise or lower the lure.



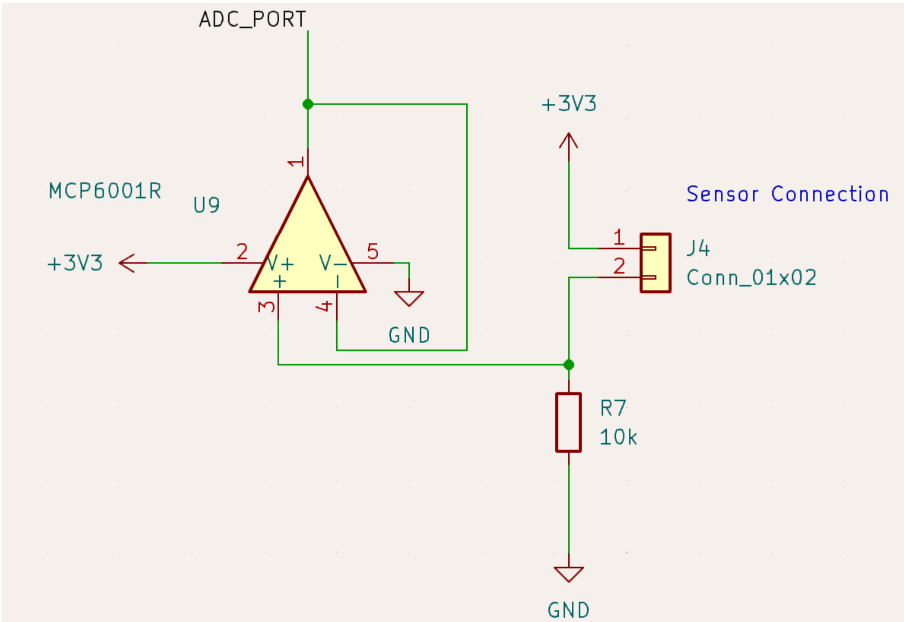
The actuator will be mounted at the tail end of the rod.



Note: We plan to use the machine shop for many of our mechanical builds.

2.3 Subsystem Descriptions, Requirements, and Verifications

Sensor Subsystem



Configuration for Sensor Subsystem

Purpose: The sensor subsystem will be responsible for detecting when there is a fish on the line, using the bend angle of the rod. This subsystem will consist of a flex bend sensor placed along the middle of the fishing rod. Flex bend sensors act as resistors, in which the resistance value is directly related to the bend angle of the sensor. We will use the voltage divider rule to allow the microcontroller to calculate the resistance value of the sensor, and then use this resistance to calculate the corresponding bend angle. One pin of the sensor will be connected to a 3.3V supply line, and the other will be connected to the ESP32 microcontroller's analog to digital converter port, as well as an additional known resistance value, which will then be connected to ground.

Interfaces:

- Control subsystem: The sensor subsystem output (labeled ADC_PORT in the above schematic) will be connected to one of the ESP32 microcontroller's GPIO pins. This output will be the voltage value in between the sensor and 10kΩ resistor, meaning that the voltage across the sensor will be 3.3V minus the value read from the GPIO pin. The microcontroller will then use this voltage value in a voltage divider rule calculation to determine the sensor's resistance value, which can then be used to determine the bend angle of the rod. If the bend angle of the flex bend sensor is determined to be greater than the threshold angle of 30 (+/-10) degrees, the microcontroller will alert the fisherman via the user application, halt the jigging function of the stepper motor, and, if the auto-reeling setting is enabled, the signal the stepper motor to reel in the line. The microcontroller will perform analogReads from the GPIO port in order to receive the sensor output.
- Power subsystem: The power subsystem will deliver 3.3V to one end of the flex sensor.

- Protocol: Voltage supply line

Components:
- MCP6001R 1 MHz, Low-Power Op Amp
- Spectra Symbol SpectraFlex Flex Sensor
- 10kΩ Resistor

Requirements	**Verifications**
• The bend angle calculated using the sensor subsystem's output voltage value is within 10 degrees of the actual bend angle of the rod	• Connect one end of a voltmeter to the sensor subsystem's voltage output and the other end to ground • Bend the rod to an angle of 45 degrees • Use a protractor to measure the actual bend angle of the rod and take note of this value • Take note of the voltage being read by the voltmeter • Use this voltage to calculate the resistance of the flex sensor, and then use this resistance to calculate the corresponding bend angle • Compare the calculated bend angle with the actual bend angle and ensure the difference is less than 10 degrees
• When auto-reeling is enabled the line is reeled in when the rod is bent at an angle of 30 (+/- 10) degrees	• Ensure that auto-reeling is enabled in the user application settings • Place the fishing device on an elevated surface (at least 5 ft) with the lure hanging over the edge • Change the settings in the user application to a line depth of 3 ft • Bend the rod to an angle of 30 (+/- 10) degrees to signal a fish on the line • Ensure that the stepper motor fully reels in the line

Application Subsystem

Purpose: The application subsystem will allow the fisherman to set the line depth and jigging frequency of the fishing device, as well as enable/disable the auto-reeling function. It will also allow them to store and view their previous catch information, and the application will receive a notification if there is a fish detected on the line. The application subsystem consists of three units: the user application, the Google Cloud server, and the MySQL Google Cloud database. The user application will force the user to create an account before using the application. Once an account has been created, the user will be able to login to their account, and control the configuration of their automatic ice fishing rod. The application will be presented in a user-friendly mode that easily allows the user to modify the settings of the fishing device. The user will also be able to insert/remove catch information from their account through a server hosted in GCP. This information can then be viewed within the app for future reference. Two tables would exist in this MySQL relational database: a user table that contains user information and a catch table that contains catch information. Attributes of the user table would include a unique username, password, first name, and last name. Attributes of the catch table would include time caught, location, depth of lure, type of fish, length of fish, weight of fish, and other information.

Interfaces:

- Control subsystem: The application subsystem will receive data from and send data to the ESP32 microcontroller via Bluetooth. When the fisherman updates their jigging and depth settings in the user application, these new values will be sent to the microcontroller via BLE, so that the microcontroller can then communicate this data with the mechanical subsystem to adjust to the desired settings. The microcontroller will also repeatedly retrieve data from the user application to check if auto reeling is enabled. This value will then be updated within the microcontroller's code to be used when a fish is detected. When the sensor subsystem detects a fish on the line, the microcontroller will send a signal to the user application to notify the fisherman.
    - Protocol: BLE

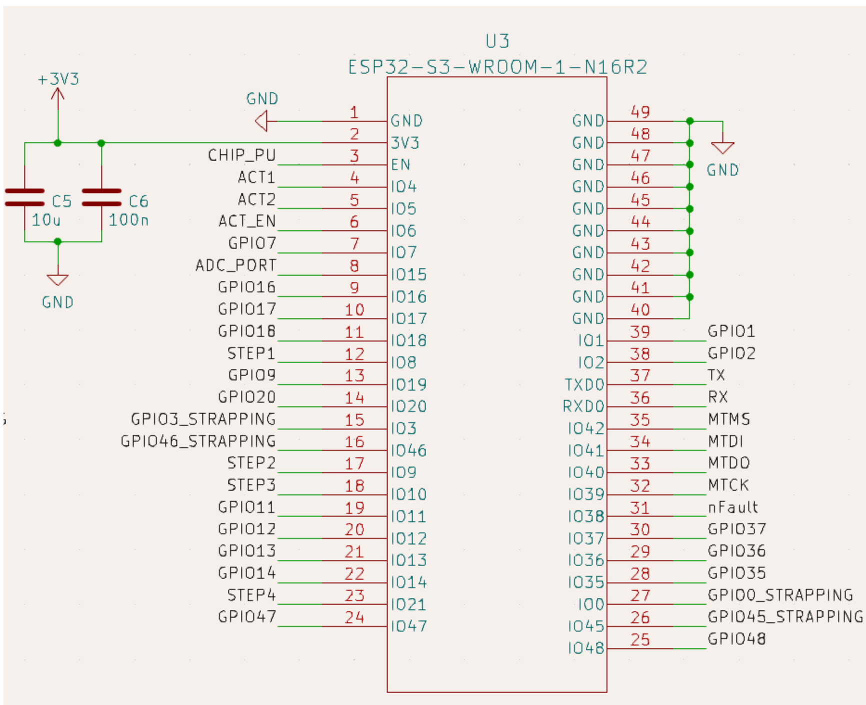
Components:

- Mobile User Application
- Google Cloud Server
- MySQL Google Cloud Database

Requirements	**Verifications**
• The user is able to create a unique account and login to the system	• Turn on the DB instance and application server • Have the user create an account in the system • Check the database to see that the users' first name, last name, username, and password have been stored
• The user is able to store catch information between 7 different attributes while logged in	• Ensure the instance is running the same as above • Have the user logged in already • Input a catch information entry into the app • Query DB to see successful store for user • Log out of the application • Log back in and go to view catch information to see that the user can still view ONLY their catch information
• The application can allow the user to set	• Navigate to the rod controls in application

the lure depth up to 50 feet and set 3 different jigging settings	• Input depth into field and option to enable jigging • Print packet sent from application to microcontroller and ensure data passed matches input • Ensure that microcontroller has received the packet/request from the mobile application

Control Subsystem



Configuration for Control Subsystem

Purpose: The control subsystem will consist of an ESP32 microcontroller which will be responsible for all processing data and executing instructions. This subsystem will receive, process, and send data to the other subsystems of the fishing device. It will signal the stepper motor driver how much to reel the line in/out, and signal the actuator driver the frequency at which the solenoid should jig the rod, based on the settings it receives from the user application. It will receive a voltage value from the flex bend sensor and use voltage divider rule to calculate the bend angle, and then use the bend angle to determine whether a fish is on the line. If the threshold angle is met, the microcontroller will signal the mechanical subsystem to halt the jigging of the rod, and, if auto-reeling is detected, reel the line in, and send a notification to the user application to inform the fisherman that a fish is on the line.

Interfaces:

- Power subsystem: The power subsystem will supply the microcontroller with 3.3V.

- Protocol: Voltage supply line

- Sensor subsystem: The output of the sensor subsystem will be connected to one of the microcontroller's GPIO ports. The microcontroller will repeatedly perform an analogRead of this port to obtain the voltage value after the flex sensor, and then use this voltage value to calculate the bend angle to determine whether or not a fish is on the line.
    - Protocol: GPIO
- Mechanical subsystem: The microcontroller will send data to the stepper motor driver and actuator driver in the mechanical system using its GPIO ports. When the fisherman updates their settings in the user application, the microcontroller will receive this data and communicate with the mechanical subsystem to make the appropriate adjustments. For example, if the user increases the line depth, the microcontroller will perform digitalWrites to the appropriate GPIO pins to communicate with the stepper motor driver to engage the motor, reeling out the desired amount of line.
    - Protocol: GPIO
- Application subsystem: The microcontroller will send data to and receive data from the user application using Bluetooth. The microcontroller will poll the user settings from the application subsystem in order to appropriately adjust line depth, jigging frequency, and auto reeling. If a fish is detected by the sensor subsystem, the microcontroller will signal the user application to notify the fisherman.
    - Protocol: BLE

Components:

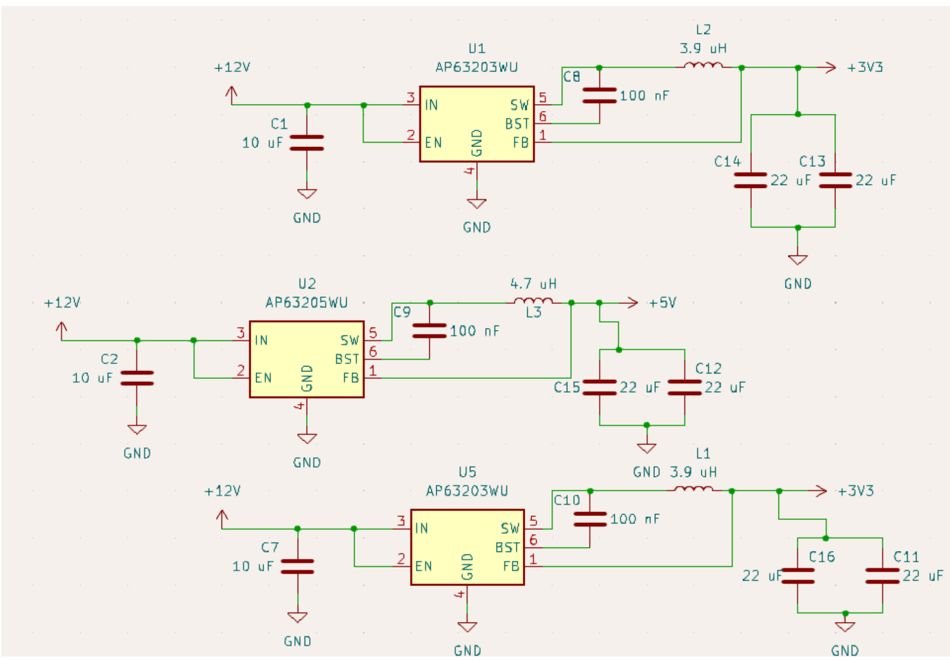
- ESP32-S3-WROOM-1-N8

Requirements	**Verifications**
• The microcontroller can receive requests from the application via Bluetooth	• Send request from user application • Program microcontroller to print out data taken from application • Ensure the data is consistent and that correct values exist in microcontroller functions
• The microcontroller is able to send a notification to the user application via Bluetooth when the sensor subsystem detects a bend angle of 30 (+/- 10) degrees	• Open the user application • Bend the rod to an angle of 30 (+/- 10) degrees to signal a fish on the line • Ensure that the user application has received a notification from the microcontroller
• The microcontroller is able to accurately read the voltage value output from the sensor subsystem within 0.1V	• Place one end of a voltmeter between the flex sensor and the 10kΩ resistor and connect the other end to ground • Take note of this voltage value (which is the output of the sensor subsystem) • Run code on the microcontroller to perform an analogRead from the

	appropriate GPIO port (GPIO15) • Take note of this voltage value (this is the voltage value being read by the ESP32) • Compare this with the previous voltage value and ensure the values are within 0.1V
• The control subsystem can receive instructions from the application subsystem to reel the line out (in) at 1 ft (+/- 0.25 ft) increments and signal the mechanical subsystem to execute this action	• Place the fishing device on an elevated surface (at least 5 ft) with the lure hanging over the edge • Change the settings in the user application to a line depth of 2 (3) ft • Use a tape measure to measure the actual length between the start and end of the line and note this length • Change the settings in the user application to a line depth of 3 (2) ft • Use a tape measure to measure the actual length between the start and end of the line and note this length • Take the difference between the actual length at the 2 (3) ft setting and the actual length at the 3 (2) ft setting • Ensure that this difference is between 0.75 ft and 1.25 ft
• When the bend angle of the rod is 30 (+/- 10) degrees the control subsystem will signal the mechanical subsystem to halt the jigging motion within 5 seconds of detecting the fish	• Bend the rod to an angle of 30 (+/- 10) degrees to signal a fish on the line and start a stopwatch • Wait until the actuator stops the jigging motion and stop the stopwatch • Ensure that the time on the stopwatch is less than 5 seconds
• When the bend angle of the rod is 30 (+/- 10) degrees and auto-reeling is enabled the control subsystem will signal the mechanical subsystem to reel in the line	• Ensure that auto-reeling is enabled in the user application settings • Place the fishing device on an elevated surface (at least 5 ft) with the lure hanging over the edge • Change the settings in the user application to a line depth of 3 ft • Bend the rod to an angle of 30 (+/- 10) degrees to signal a fish on the line • Ensure that the stepper motor fully reels in the line
• The control subsystem can receive inputs to alter the jigging frequency from the application subsystem and signal the mechanical subsystem to jig within 5 Hz	• Change the settings in the user application to one of the three possible frequencies • Start a stopwatch at one of the jigs • Stop the stopwatch at the next jig

of the desired frequency	• Note the time from the stopwatch (this is the period of the jigs) • Calculate 1 divided by the measured period to get the actual jig frequency • Compare this with the frequency set on the user application and ensure the values are within 5 Hz of each other • Repeat the above steps for the other two jigging frequencies

Power Subsystem



Configuration for Power Supply Subsystem

Purpose: The power subsystem will be responsible for delivering power to the other subsystems of the device. This subsystem consists of a 12V, 10Ah lead-acid battery to provide a power source, and three voltage regulators to drop the voltage to the values required by other subsystems. Two of these buck regulators will drop the voltage to 3.3V, one to be used for the microcontroller and the sensor subsystem and another to be used by the stepper motor driver. We will use two separate voltage regulators because if the microcontroller shares a voltage supply with the motor driver, the current to the microcontroller may drop when the motor engages, causing the ESP32 to reset. The other buck converter will drop the 12V battery to 5V, which will be used by the actuator driver.

Interfaces:

- Control subsystem: The power subsystem will deliver 3.3V to the ESP32 microcontroller's 3V3 input port, using the output of the AP63203WU-7 buck switching regulator.
    - Protocol: Voltage supply line

- Mechanical subsystem: The power subsystem will deliver 3.3V to the DRV8833PW's VM, VCP, and nSLEEP ports using the output of the AP63203WU-7 buck switching regulator. It will deliver 5V to the L293D's VCC1 port using the output of the AP63205WU-7 buck switching regulator, and 12V to its VCC2 port using the battery voltage.
    - Protocol: Voltage supply line
- Sensor subsystem: The power subsystem will deliver 3.3V to the sensor subsystem to be used as VDD in the voltage divider rule.
    - Protocol: Voltage supply line

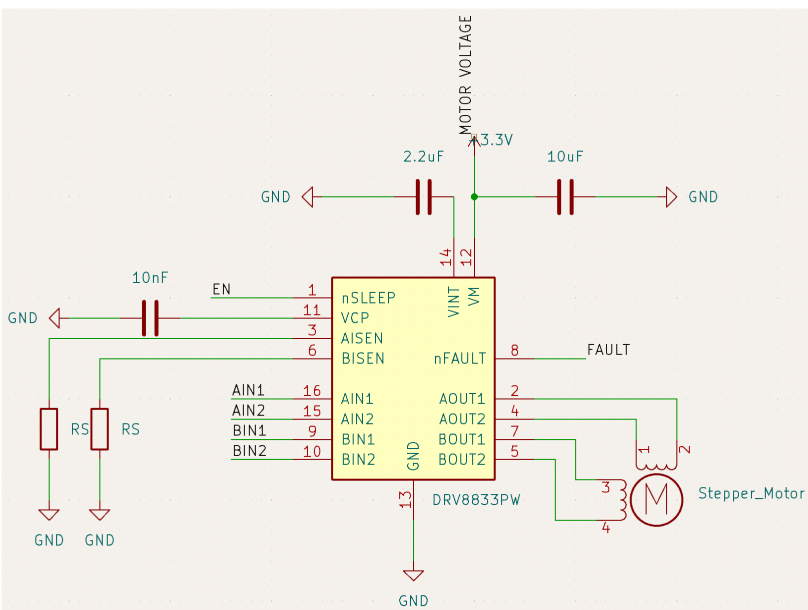
Components:

- AP63203WU-7 Buck Switching Regulator IC Positive Fixed 3.3V 1 Output 2A (x2)
- AP63205WU-7 Buck Switching Regulator IC Positive Fixed 5V 1 Output 2A
- 12V 10Ah Lead Acid Battery
- 10uF Capacitor (x3)
- 22uF Capacitor (x6)
- 100nF Capacitor (x3)
- 4.7uH Inductor
- 3.9uH Inductor

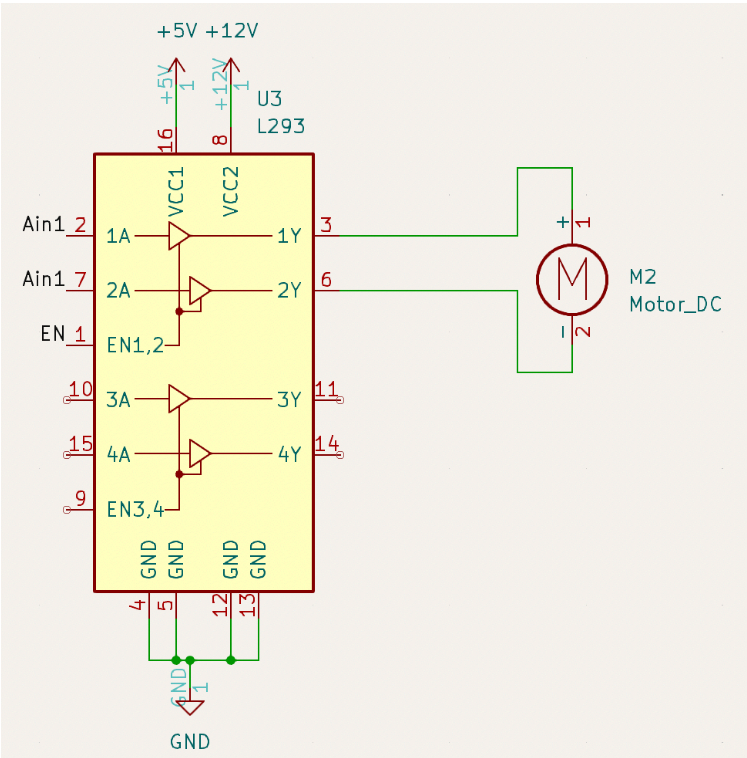
Requirements	**Verifications**
• The power subsystem is able to output 3.3V (+/- 0.1V) to be used as the sensor subsystem's VDD	• Connect the power subsystem to the battery • Connect one end of a voltmeter to the 3.3V output voltage line of the power subsystem (the one that will be used by the sensor and microcontroller) and the other end to ground • Note this voltage value and ensure it is between 3.2V and 3.4V
• The power subsystem is able to output 3.3V (+/- 0.1V) to be delivered to the ESP32	• Connect the power subsystem to the battery • Connect one end of a voltmeter to the 3.3V output voltage line of the power subsystem (the one that will be used by the sensor and microcontroller) and the other end to ground • Note this voltage value and ensure it is between 3.2V and 3.4V
• The power subsystem is able to output 3.3V (+/- 0.1V) to be delivered to the stepper motor driver (DRV8833PW)	• Connect the power subsystem to the battery • Connect one end of a voltmeter to the 3.3V output voltage line of the power subsystem (the one that will be used by the mechanical subsystem) and the other

	end to ground • Note this voltage value and ensure it is between 3.2V and 3.4V
• The power subsystem is able to output 5V (+/- 0.1V) to be delivered to the actuator driver's (L293D) VCC1 port	• Connect the power subsystem to the battery • Connect one end of a voltmeter to the 5V output voltage line of the power subsystem and the other end to ground • Note this voltage value and ensure it is between 4.9V and 5.1V
• The power subsystem is able to output 12V (+/- 0.1V) to be delivered to the actuator driver's (L293D) VCC2 port	• Connect the power subsystem to the battery • Connect one end of a voltmeter to the 12V output voltage line of the power subsystem (which is connected directly to the battery) and the other end to ground • Note this voltage value and ensure it is between 11.9V and 12.1V

Mechanical Subsystem



Motor Setup for Stepper Motor Driving Fishing Reel



Motor Setup for Actuator

Purpose: The mechanical subsystem will reel the line in and out and perform the jigging motion of the rod. This subsystem will consist of an ice fishing rod (short rod length) that is being held by a fishing rod holder (as described in the Physical Design Overview), which holds the rod upright and dangling the line above the water. A 12V actuator will be attached to the tripod stand, near the tail end of the rod, and will push the rod up and down repeatedly to simulate the jigging motion. An H-bridge driver will control the actuator's direction. A 3.3V stepper motor will be attached to the fishing reel to allow for line to be automatically reeled in or out. An H-bridge driver IC will reverse the polarity across the stepper motor to allow the stepper motor to switch directions for reeling in and out.

Interfaces:

- Control subsystem: The control subsystem will send data to the stepper motor driver and actuator driver via its GPIO ports. When the user changes their settings in the user application, the microcontroller will signal the stepper motor driver to reel in/out the line to reach the appropriate depth, and change the frequency at which it signals the actuator driver to move the actuator up and down, changing the jigging frequency to the desired value. When a fish is detected by the sensor subsystem, the microcontroller will signal the motor driver to reel in the line (if auto-reeling is enabled), and signal the actuator driver to halt the jigging. This communication will be done using digitalWrites to the microcontroller's GPIO pins, which will be connected to the drivers' input pins.
    - Protocol: GPIO
- Power subsystem: The power subsystem will deliver 3.3V to the DRV8833PW's VM, VCP, and nSLEEP ports using the output of the AP63203WU-7 buck switching regulator. It will deliver 5V

to the L293D's VCC1 port using the output of the AP63205WU-7 buck switching regulator, and 12V to its VCC2 port using the battery voltage.

- Protocol: Voltage supply line

Components:

- L293D Quadruple Half-H Driver
- DRV8833PW Dual H-bridge Motor Driver
- Nema17 Stepper Motor 3.3V 95oz-in 60mm 1.8A, Hybrid 2-Phase 0.65NM 4-Wire 17HS4218
- NORJIN 12V Mini Electric Linear Actuator 0.8" Stroke, 32N/7.2lbs, Speed 15mm/s

Requirements	**Verifications**
• The stepper motor is able to lower the fishing line to 50 ft (+/- 5 ft) when fully reeled out	• Go into a stairwell with a height of at least 50 ft and place the fishing device so that the lure is hanging above the stairwell • Change the settings in the user application to the maximum line depth (50 ft) • Wait for the stepper motor to fully lower the lure • Have one person hold a tape measure at the top of the stairwell where the line begins, and have another person at the bottom of the stairwell holding the other end of the tape measure where the lure is • Note this length and ensure that it is between 45 ft and 55 ft
• The stepper motor is able to **lower** the fishing line in increments of 1 ft (+/- 0.25 ft)	• Place the fishing device on an elevated surface (at least 5 ft) with the lure hanging over the edge • Change the settings in the user application to a line depth of 2 ft • Use a tape measure to measure the actual length between the start and end of the line and note this length • Change the settings in the user application to a line depth of 3 ft • Use a tape measure to measure the actual length between the start and end of the line and note this length • Take the difference between the actual length at the 2 ft setting and the actual length at the 3 ft setting • Ensure that this difference is between 0.75 ft and 1.25 ft
• The stepper motor is able to **raise** the fishing line in increments of 1 ft (+/- 0.25 ft)	• Place the fishing device on an elevated surface (at least 5 ft) with the lure hanging over the edge

	• Change the settings in the user application to a line depth of 3 ft • Use a tape measure to measure the actual length between the start and end of the line and note this length • Change the settings in the user application to a line depth of 2 ft • Use a tape measure to measure the actual length between the start and end of the line and note this length • Take the difference between the actual length at the 3 ft setting and the actual length at the 2 ft setting • Ensure that this difference is between 0.75 ft and 1.25 ft
• The actuator is able to jig the rod up and down within 0.01 Hz of the desired frequency	• Change the settings in the user application to one of the three possible frequencies • Start a stopwatch at one of the jigs • Stop the stopwatch at the next jig • Note the time from the stopwatch (this is the period of the jigs) • Calculate 1 divided by the measured period to get the actual jig frequency • Compare this with the frequency set on the user application and ensure the values are within 5 Hz of each other • Repeat the above steps for the other two jigging frequencies
• When auto-reeling is enabled and a fish is detected by the sensor subsystem, the stepper motor fully reels in all of the line currently out, returning to the original position of a depth of 0 ft (+/- 1 ft)	• Ensure that auto-reeling is enabled in the user application settings • Place the fishing device on an elevated surface (at least 5 ft) with the lure hanging over the edge • Change the settings in the user application to a line depth of 3 ft • Bend the rod to an angle of 30 (+/- 10) degrees to signal a fish on the line • Wait for the stepper motor to reel in the line • Measure the length of the remaining line out by placing one end of a tape measure at the tip of the rod and the other end at the lure and note this length • Ensure that this length is less than 1 ft
• The stepper motor is able to fully reel in a weight of 2 lbs (+/- 0.5 lb)	• Ensure that auto-reeling is enabled in the user application settings • Place the fishing device on an elevated

	surface (at least 5 ft) with the lure hanging over the edge • Change the settings in the user application to a line depth of 3 ft • Attach a 2 lb (+/- 0.5 lb) weight to the end of the line • Bend the rod to an angle of 30 (+/- 10) degrees to signal a fish on the line • Wait for the stepper motor to reel in the line • If the stepper motor is able to reel in the weight, the test succeeded
• The jigging motion halts within 5 seconds when the bend angle of the rod is 30 (+/- 10) degrees	• Bend the rod to an angle of 30 (+/- 10) degrees to signal a fish on the line and start a stopwatch • Wait until the actuator stops the jigging motion and stop the stopwatch • Ensure that the time on the stopwatch is less than 5 seconds

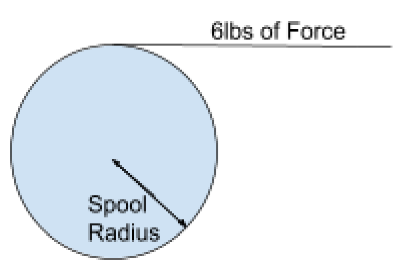
2.4 Tolerance Analysis

Force Exerted by a Fish on the Line

Our automatic ice fishing rig is designed to be used on small to medium inland lakes. Many fishermen prefer to use 2-8 lb test line when selecting a fishing line weight. Line weights in this range can withstand the stress of many species of fish commonly caught in inland lakes. During the Winter, fish are more sluggish and will not pull as hard as in warmer weather; as a result, we can assume up to a maximum 6 lbs of force will be applied to our line. When determining the power of our stepper motor, we need to select a stepper motor that can operate under the stress of 6 lbs of force.

$$6\ lb\ of\ Force\ = 26.689\ Newtons$$

Using the force calculated, we can determine the necessary torque that will drive our fishing reel.



Determining the spool radius is challenging, because as the line is reeled-out the radius will decrease and as the line is reeled-in the radius will increase. We believe on average our spool's radius measures between:

$$12.7\ mm\ -\ 19\ mm$$

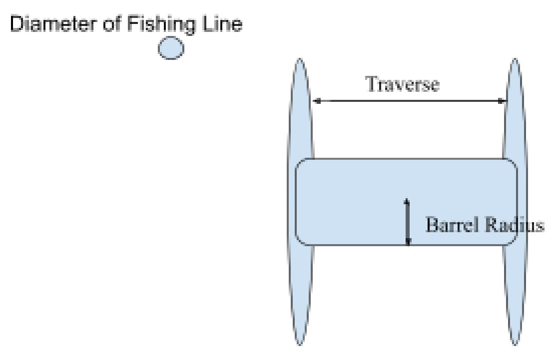
Calculating Torque needed by our stepper motor.

$$Torque\ (Nm)\ =\ 26.689\ Newtons\ \star\ (0.0127\ m\ -\ 0.0190\ m)\ =\ 0.34\ Nm\ -\ 0.51\ Nm$$

The stepper motor can supply up to 0.65 Nm of torque, so it can support the stress we expect.

Measuring Line Out

Using our spool's radius, the amount of fishing line on our reel, and the length of our fishing rod, we can determine how much fishing line is out.



Diameter of Fishing Line:

- 8 lb monofilament fishing line: 0.27mm = 0.00027 m

Barrel Radius:

- 0.0127 m

Traverse (length of spool):

- 0.0381 m

Circumference (No line on):

- $2\ \Pi r\ =\ 2\Pi(0.0127\ m)$ = 0.0797 m

Individual threads of line across spool's length:

- 0.0381 m / 0.00027 m = 141.11 ≈ 141 threads

Each layer of line will add 0.00027 m to the spool's radius.

Suppose 100 m of line is spooled onto the reel. Calculate the number of layers, n, of line on the spool:

- $100m\ =\ \sum_{x=0}^{n-1}\ (2\ \pi\ (0.0127\ +\ 0.00027x)\ )\ *\ 141$
- $100m\ =\ \sum_{x=0}^{n-1}\ 2\ \pi\ (0.0127)\ *\ 141\ +\ \sum_{x=0}^{n-1}\ 2\pi(\ 0.00027x)\ \ *\ 141$
- $100m\ =\ 2\ \pi\ (0.0127)\ *\ 141\ *\ n\ +\ 2\pi(\ 0.00027)\ \ *\ 141\ *\ n\ *\ (n\ -\ 1)/2$
- $n\ \approx\ 8.25$ layers of line on the spool

Considering there is 100 m of line on the spool. How many rotations of the spool are required to release x meters of line?

- $x = \sum_{i=n}^{8.25-1} 2\pi\ (0.0127) * 141 + \sum_{i=n}^{8.25-1} 2\pi(\ 0.00027i) \ * 141$
- $x = \sum_{i=0}^{8.25-1} 2\pi\ (0.0127) * 141 - \sum_{i=0}^{n-1} 2\pi\ (0.0127) * 141 + \sum_{i=0}^{8.25-1} 2\pi(\ 0.00027i) \ * 141 - \sum_{i=0}^{n-1} 2\pi(\ 0.00027i) \ * 141$
- $x = 100 - 2\pi\ (0.0127) * 141 * n - 2\pi(\ 0.00027) \ * 141 * n * (n - 1)/2$

Converting this into a quadratic equation in terms of n, we get

- $0 = 0.000135n^2 + 0.012565n - (100 - x)/(282\pi)$

We can then solve for n using the quadratic formula to get the layer reached after releasing x meters of line. 8.25 - n will then give us the number of layers needed to unreel x meters of line (where 8.25 is the initial number of layers and n is the final number of layers after unreeling), and because each layer contains 141 threads, we need 141 revolutions to unreel a single layer. This means we need 141*(8.25 - n) rotations to release x meters of line.

The previous calculation calculates the number of rotations *of the reel* needed to unreel x meters of line. The number of rotations *of the stepper motor* needed for this, however, will be different, due to the gear ratio of the fishing reel. The gear ratio is the number of times the spool rotates for every one rotation of the hand crank. Our reel has a gear ratio of 5.2:1, meaning that every time the stepper motor makes one full rotation, the reel will make 5.2 rotations. So, the stepper motor will need to rotate 141*(8.25 - n)/5.2 times to unreel x meters of line.

The maximum length of line will be 50 ft (15.24 m). Using the above equation, to unreel 15.24 meters of line, we will need 165.493 rotations of the reel. If we ignore the change in radius due to added layers, holding the radius at a constant 0.0127 m, 165.493 rotations will unreel 13.21 m of line (43.34 ft). This means that the difference between accounting for and ignoring the change in radius has a maximum value of 50 - 43.34 = 6.66 ft, which is greater than our tolerance of 5 ft for the depth of the line. This means that we *will* have to account for the change in radius when calculating line depth, using the summation equations shown above.

Microcontroller Data Reading

We will poll the ESP32 microcontroller ADC port to read data from the flex bend sensor. In order to ensure that the microcontroller has enough time to convert the voltage value read by the ADC port into the estimated bend angle, and determine if it is above the trigger threshold, we will set the ADC's polling frequency to be much slower than the ESP32's clock frequency. We will set the frequency of the ADC peripheral to 32kHz using the internal low-power 32kHz RC, meaning that the sensor's resistance value will be read 32 thousand times per second. If a fish is on the line, this high polling frequency will ensure that the change in bend angle is detected almost immediately. The microcontroller's CPU clock (80 MHz crystal oscillator with integrated trimming capacitors) operates at 80MHz, much faster than the polling frequency of the ADC port.

Instructions/second = 80,000,000
ADC polling frequency = 32kHz
Number of instructions performed between polls = ((32,000,000 instructions)/(1 second))/((32,000 polls)/(1 second)) = 1,000 instructions/poll

This means that 1,000 instructions can be executed by the microcontroller before the next poll from the ADC port. Even though some lines of code will take more than one clock cycle to complete, the analogRead, the conversion from voltage to bend angle, and the check for meeting threshold angle will take less than 1,000 instructions. This means that the polled voltage value will not change while calculations are being performed, giving us accurate results.

Calculating Bend Angle

Because the flex bend sensor's resistance changes based on the bend angle, the ESP32 microcontroller will need to convert the resistance to the corresponding angle, so it can then be converted to the corresponding force value. To obtain this resistance value, we will use the voltage divider rule, where one of the resistors is the flex bend sensor. By connecting one side of the flex bend sensor to a constant voltage value (3.3V) and the other side to the ESP32's ADC port, we can obtain the voltage drop across the sensor with 3.3V minus the value read by the ADC port.

Variables:

- v: voltage drop across sensor
- y: voltage read by the ADC port
- X: sensor resistance
- R: known resistance value
- X_i: sensor resistance when straight
- X_f: sensor resistance when bent at a 180 degree angle
- a: bend angle of sensor

Calculating voltage across the sensor:

$$v = 3.3 - y$$

Using voltage divider rule:

$$v = 3.3 * X/(X + R)$$

Isolating X to solve for the resistance of the sensor:

$$X = v * R/(3.3 - v)$$

Using this calculated sensor resistance, we can estimate the bend angle with the knowledge that there is a relatively linear relationship.

$$a = 180 * (X - X_i)/(X_f - X_i)$$

Because resistors typically have tolerances between 2% and 5%, our 3.3V voltage line (according to high level requirements) has a tolerance of 0.1V, and the flex bend sensor we are using has a tolerance of 30%, we must account for this when calculating the bend angle.

$$v = (3.3 \pm 0.1) - y$$

$$X = (3.3 \pm 0.1 - y) * (R \pm R * 0.05)/(3.3 \pm 0.1 - (3.3 \pm 0.1 - y))$$

Substituting in our value for R (10kΩ), and canceling out the supply voltages (3.3 ± 0.1) in the denominator because they are the same line and so will share the same tolerance:

$$X = (3.3 \pm 0.1 - y) * (10 \pm 0.5)/y$$

Substituting this value into the equation for bend angle:

$$a = 180 * ((3.3 \pm 0.1 - y) * (10 \pm 0.5)/y - X_i)/(X_f - X_i)$$

Then substituting in our values for the maximum and minimum resistance of the sensor, X_i = 10kΩ ± 3kΩ and X_f = 20kΩ ± 6kΩ (including corresponding tolerance of 30%):

$$a = 180 * ((3.3 \pm 0.1 - y) * (10 \pm 0.5)/y - (10 \pm 3))/((20 \pm 6) - (10 \pm 3))$$

We can then calculate the minimum and maximum values of a using these tolerances, to get a range of possible values:

Minimum value:

$$180 * ((3.3 - 0.1 - ADC) * (10 - 0.5)/y - (10 + 3))/((20 + 6) - (10 - 3))$$
$$= 180 * ((3.2 - y) * 9.5/y - 13)/19$$
$$= 288/y - 213.158$$

Maximum value:

$$180 * ((3.3 + 0.1 - ADC) * (10 + 0.5)/y - (10 - 3))/((20 - 6) - (10 + 3))$$
$$= 180 * ((3.4 - y) * 10.5/y - 7)$$
$$= 6426/y - 3150$$

Expected value:

$$180 * ((3.3 - y) * 10/y - 10)/(20 - 10)$$
$$= 594/y - 360$$

As can be seen from the above equation, the tolerances have a massive impact on the range of possible values when calculating the bend angle. For example, if sensor is bent to 90 degrees, the expected resistance value will be 15kΩ, meaning the voltage read by the ADC (y-variable) port should be 1.32V. Accounting for the tolerances, however, if we read 1.32V, the calculated bend angle will range between 5.0238 degrees to 1718.18 degrees. If our calculated angles are off by this much, our trigger for whether or not a fish is on the line will almost certainly fail, preventing our system from functioning properly. In order to fix this issue, we can simply measure the actual resistance values of our resistors and sensor (both when straight and when bent at 180 degrees), as well as measure the actual voltage of the 3.3V line. This will be one of our first tasks after receiving our parts and building our power subsystem, so that we can significantly decrease these tolerances (especially with regards to the flex bend sensor, as a 30% tolerance is an incredibly large window), making them near negligible when calculating the bend angle.

3 Cost & Schedule

3.1 Cost Analysis

We expect to spend 15 hours/week working on this project for the next 9 weeks. Therefore, we look to spend 135 hours in total per person for completing this project. A comparable industry salary for this type of position would expect \$45/hour. Each team member will cost \$45/hour x 2.5 x 135 hours = \$15,187.50 for the semester. The total team, consisting of three people, will cost \$45,562.50 in labor costs.

We expect to spend approximately \$174.45 in parts. However, we look to minimize our parts costs by using fishing gear that we have in our garages. The parts cost listed does not take into account this factor.

In total, we expect our project to cost: \$45,562.50 + \$174.45 = \$45,736.95

Description	**Manufacturer**	**Quantity**	**Extended Price ($)**	**Link**
Flex Sensor 95mm Male Pins	Spectra Symbol	1	25.45	LINK
Nema17 Stepper Motor 3.3 V 95oz-in 60mm 1.8A,Hybrid 2-Phase 0.65NM 4-Wire 17HS4218	CNCTOPBAOS	1	17.50	LINK
L293D Quadruple Half-H Drivers	Texas Instruments	1	8.11	LINK
DRV8833PW Dual H-bridge motor driver	Texas Instruments	1	2.84	LINK

NORJIN 12V Mini Electric Linear Actuator 0.8" Stroke, 32N/7.2lbs, Speed 15mm/s, Linear Motion Actuators with Mounting Brackets	Poweka	1	22.00	LINK
AP63203WU-7 Buck Switching Regulator IC Positive Fixed 3.3V 1 Output 2A	Diodes Incorporated	4	3.48	LINK
AP63205WU-7 Buck Switching Regulator IC Positive Fixed 5V 1 Output 2A	Diodes Incorporated	2	1.74	LINK
ESP32-S3-WROOM-1-N8	Espressif Systems	1	3.20	LINK
12v 10Ah SLA Rechargeable Battery - F2 Terminals	ExpertPower	1	24.50	LINK
360 Degrees MiniPort Fishing Rod Holder	METER STAR	1	14.97	LINK
Ugly Stik Complete Spincast Reel and Fishing Rod Kit	Ugly Stik	1	32.37	LINK
8lb Fishing Line	Zebco	3	3.29	LINK
Plywood 3/8 in		1	5	N/A
Miscellaneous Circuit Elements (Resistors, Etc.)	N/A	N/A	10	N/A

3.2 Schedule

Week	Task	Person
February 19th - February 26th	Finalize and submit design document	James
	Revise and resubmit project proposal	Andrew
	Review schematic for design	Luke
	Prepare for design review	Everyone
February 26th - March 4th	Complete design review	Everyone

	Order parts	Andrew
	Design PCB	Luke
	Order PCB	James
March 4th - March 11th	Measure actual resistance values of flex bend sensor and all other resistors	Everyone
	Solder printed PCB	James
	Begin writing code for microcontroller	Andrew
	Assemble/test mechanical system	Luke
March 11th - March 18th	Begin building power subsystem	Everyone
	Finalize software and program the microcontroller	James
	Test the finished PCB	Andrew
	Connect microcontroller to sensor and MCU calculates correct bend angle	Luke
March 18th - March 25th	Connect power subsystem to components and begin testing	James
	Build user application database instance and structure	Andrew
	Build front end and core application with connection to db	Luke
March 25th - April 1st	Test that the application can send requests to the microcontroller	Everyone
	Test Sensor Subsystem	James
	Test Control Subsystem	Andrew
	Test Application Subsystem	Luke
April 1st - April 8th	Continue to test system as a whole and make necessary	James

	changes	
	Continue to test system as a whole and make necessary changes	Andrew
	Continue to test system as a whole and make necessary changes	Luke
April 8th - April 15th	Fix minor existing bugs	James
	Fix minor existing bugs	Andrew
	Fix minor existing bugs	Luke
April 15th - April 22nd	Demo	James
	Demo	Andrew
	Demo	Luke

4 Ethics and Safety

When developing an automatic ice fishing rod, it's essential to consider various ethical and safety issues, both during the development process and in terms of potential misuse. The IEEE and ACM (Association for Computing Machinery) Codes of Ethics provide general guidelines that can be applied to our project. The issues listed below ensure our project upholds to the highest standards established:

Privacy Concerns (ACM Code 1.6)

Any time personal data is being collected and stored, there is a risk of a breach of privacy. Our application will allow users to upload data to a GCP database, which will contain two tables: one for personal information and one for catch information. The main privacy concern deals with the former, as this table will contain the user's first and last name. In the application's privacy policies, we will clearly state that the information they are submitting will be stored in a database, and ask that they only proceed if they give us consent to store their data. In the event that this database is hacked, this information could be used for malicious purposes that could potentially harm the user. In order to minimize these consequences in the event of an attack, this risk will be clearly stated in the privacy policies, and will inform the user that they may use an alias when submitting this information if they wish to take extra precautions.

Transparency and Honesty (ACM Code 1.3)

When developing a new product, it is important to fully document the entire design and implementation processes. Data should not be tampered with and no values should be altered to ensure honesty and transparency. The claimed capabilities of our design must accurately reflect its actual capabilities, so users receive the quality they expect. In order to make sure we are transparent throughout our development, we will thoroughly document the process in our lab notebooks. We will also make our code open sourced so that users can understand our system and how it works. The claimed values will be realistic and accurate, and will be reported with tolerance values (e.g., +/- 5 N) to account for slight differences in conditions, equipment, etc.

Mechanical and Electrical Safety (IEEE 1.1)

These codes involve safety issues related to mechanical and electrical failures in the rod or its deployment mechanisms.To mitigate these issues we will employ the following precautions. The battery and PCB will be enclosed in an element-proof box to minimize the risk of electric shock and mechanical/electrical damage to the system. The system components will also be properly grounded. The automatic reeling system will operate at a frequency that is safe for the user as the fisherman's line will not be reeled in excessively fast. We will rigorously test the mechanical stability of the rod to ensure that it can withstand the stress of catching multiple fish of varying sizes.

User Training and Guidelines (ACM Code 1.2)

This code relates to safety issues surrounding Injuries due to improper use or lack of understanding of the equipment. In our user manual we will have instructions for using the rod. To mitigate this issue we will provide clear user manuals, safety guidelines, and potentially implement features like emergency stop mechanisms. These instructions will include how to turn on and off the system. There will also be an analog off switch on the power delivery circuit to cut power in cases of extreme malfunction. The user manual will also have safety guidelines such as how far to stand from the rod and advise against putting your hands near the hook. Our user manual will also advise against tampering and have information about how to handle component breakdown such as battery corrosion or leakage.

References

Code of Ethics - Association for Computing Machinery, www.acm.org/code-of-ethics. Accessed 7 Feb. 2024.

"IEEE Code of Ethics." *IEEE*, www.ieee.org/about/corporate/governance/p7-8.html. Accessed 7 Feb. 2024.

Texas Instruments. "L293D Quadruple Half-H Drivers Datasheet." Texas Instruments, Jan. 2016, https://www.ti.com/lit/ds/symlink/l293d.pdf?ts=1710069624078&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FL293D.

Texas Instruments. "DRV8833 Dual H-Bridge Motor Driver Datasheet." Texas Instruments, July 2015, https://www.ti.com/lit/ds/symlink/drv8833.pdf?ts=1710011492587.

Diodes Incorporated. "AP63200/AP63201/AP63203/AP63205 3.8V TO 32V INPUT, 2A LOW IQ SYNCHRONOUS BUCK WITH ENHANCED EMI REDUCTION Datasheet." Diodes Incorporated, Jan. 2019, https://www.diodes.com/assets/Datasheets/AP63200-AP63201-AP63203-AP63205.pdf.

Espressif Systems. "ESP32-S3-WROOM-1 ESP32-S3-WROOM-1U Datasheet." Espressif Systems, 2023, https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf.

Spectra Symbol. "SpectraFlex FLX Datasheet." Spectra Symbol, 2014, https://cdn.shopify.com/s/files/1/0578/4128/7283/files/SPECTRAFLEX_DATA_SHEET_v1.0.pdf?v=1691015077.

Raj, Aswinth. "Programming ESP32 Board with Arduino Ide." *Circuit Digest - Electronics Engineering News, Latest Products, Articles and Projects*, Circuit Digest, 15 Oct. 2018, circuitdigest.com/microcontroller-projects/programming-esp32-with-arduino-ide#:~:text=Programming%20ESP32%20with%20Arduino%20IDE%3A%201%20STEP%201%3A,your%20ESP32%20is%20connected%20to.%20...%20More%20items.

"ESP32 Bluetooth Classic with Arduino IDE - Getting Started." *Random Nerd Tutorials*, RandomNerdTutorials.com, randomnerdtutorials.com/esp32-bluetooth-classic-arduino-ide/. Accessed 10 Mar. 2024.