Energy Storage and Direct Air Capture in the Deep Ocean

Team Number: 500-SP2-Aqua Carbon Capturers:

Project Manager: Chace Nielson (chacenielson@ucalgary.ca) (UCID: 30045674)

Other Team Members:

- Nabeel Amjad (UCID: 30061166)
- Claes Medrano (UCID: 30087924)
- Cale Morash (UCID: 30066719)
- Fahad Yasin (UCID: 30088651)

Teaching Assistant: Shahin Parvar

Academic Advisor: Dr. Marjan Jose Eggermont

Support Technician: Dr. Eric John Limacher

Sponsor Representative: Dr. Roman Shor (roman.shor@ucalgary.ca)

Sponsor Company: Energi Simulation Centre for Geothermal Systems Research (GeoS) at the University of Calgary

Table Of Contents

1. Glossary

Common terms found in this document

Hydrostatic Pressure	The pressure exerted by a fluid due to its weight. It increases in proportion to the measured depth from the surface because of the increasing weight of fluid exerting a downward force from above.
Renewable Energy	Energy from sources that are naturally replenishing but flow-limited. Renewable resources are virtually inexhaustible in duration but limited in the amount of energy that is available at one time.
Buoyant Force	The upward force exerted by a fluid that opposes the weight of an immersed object.
Simulation	The use of a computer model to replicate the behavior of a system or process. Virtual representation of a real system.
Prototype	The device that we are building to test the feasibility of a single gondola. It is a smaller, more practical device than the original gondola design. The prototype will be approximately one meter tall.
CAD	Computer-Aided Design. It is a technology that aids in the creation, modification, and optimization of a digital design.
UI	User interface refers to how users interact with and control digital devices or software applications, encompassing visual elements, input controls, and navigational components.
PPM	Parts per million. This measurement details how many parts out of a million parts are the specified substance.
Direct Air Capture	Technologies that extract CO_2 directly from the atmosphere, for CO_2 storage or utilization.
MVP	Minimum viable product. A basic version of a product with minimum features. It is enough to satisfy early customers and gather feedback for future development.
Amazon Web Services	Amazon Web Services (AWS) is a comprehensive cloud computing platform offering a wide range of services, including computing power, storage, and database solutions, to businesses and individuals.

2. Executive Summary

This project explored separating carbon dioxide (CO_2) from air using deep ocean hydrostatic pressure. Concepts for this process were created in a past project [1.1] that focused on energy storage using the buoyant force of water. It also had a bonus goal of separating CO_2. These concepts hold promise for addressing climate change. This current project aims to build a physical device that can bring these concepts into the real world.

The foundation for this project is a paper written by our sponsor Dr. Roman Shor regarding "Energy Storage and Direct Air Carbon Capture Solutions for Offshore Sources of Energy" [1.1]. It aimed to harness the buoyant forces of water for energy storage, positing a novel approach to using renewable energy. An important aspect of this research was its dual focus: energy storage and separating CO_2 from the atmosphere. When combined with the environmental conditions in the deep ocean this idea presents the possibility of improving sustainable energy sources. Building upon the achievements and insights from this past work, our project aimed to bring these high-level concepts closer to practical implementation with a physical prototype.

The approach to finalizing this project involved utilizing 3D printing, Flex Seal, epoxy, and other tools to construct a custom enclosure. This enclosure has balloons positioned above and below a valve. Each one would be compressed while underwater leading to CO_2 condensing. This liquid would drip to the bottom balloon and the valve would close to separate the chambers, each with a different concentration of CO_2. An electronics system would utilize this housing as an enclosure to record data for pressure and CO_2 concentration during the underwater test and trigger the valve at a set depth. The final test was performed in a pool to ensure the valve could close and data could be recorded.

In addition to the physical prototype a simulation was also created to showcase the prototype design, use case, and the energy storage system when combined with the CO_2 separation system. This simulation was a secondary goal to the physical prototype.

While testing the device in the deep ocean was not feasible for this project, the result of pool testing had the valve snap closed at a depth of 2.1 meters while recording pressure data. The CO_2 concentration in parts per million was also recorded in each balloon. The moment the valve was closed was also recorded and showed that the expected closure depth matched the set pressure threshold. The device was able to record data at a depth of 2.7 meters and we see no reason this electronic circuit couldn't be applied to far deeper bodies of water in future tests.

The primary goal was to create the system in hopes future groups would be able to take advantage of the electronics and housing plans to test deep enough to separate CO_2 and prove the possibility of separation. The housing will need to be improved to handle high pressure before future tests are performed.

This project represents an exploratory step in creating a physical device to test the concepts related to CO_2 separation. This project lays a blueprint that future work can use to improve the design.

3. Motivation and Objectives

Two focuses of the modern world are the build-up of CO_2 in the atmosphere due to human activity and improving the use of renewable energy sources. Our current renewable sources (solar, hydro, wind) lack efficiency and consistency as they depend on fluctuating environmental factors. For instance, wind power depends on the speed and consistency of wind. If there is little wind there is an energy shortage, if there is high wind there is an energy surplus. A way to circumvent this would be to store excess energy produced during high wind periods and release it during low wind periods.

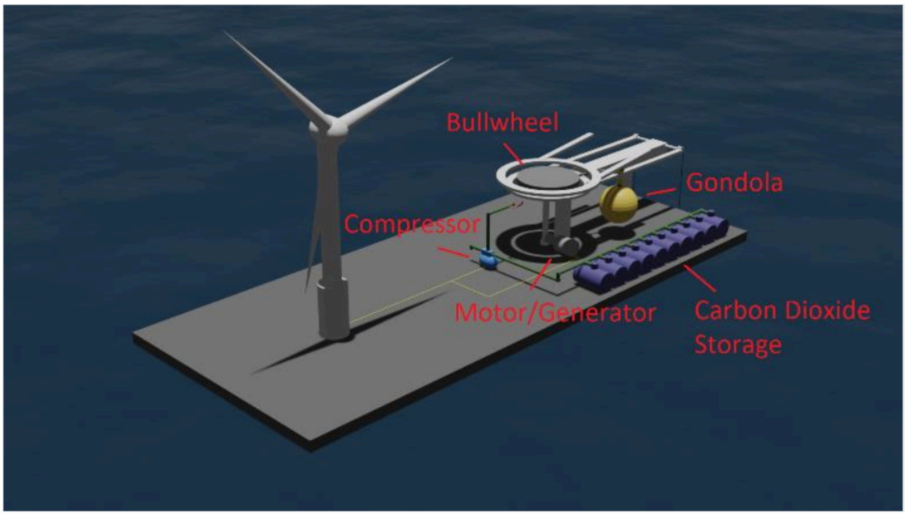

Figure 3.1. Concept design of the offshore energy storage system (top). This design operates by sending air-filled gondolas deep underwater during periods of high wind. They can be retrieved during low wind periods to extract potential energy created by the buoyant force of air. The concept also includes a device to separate CO_2 within each gondola (fig 3.2).

Previous work on this subject [1.1] from past capstone projects was to create a high-level and theoretical plan for an energy storage system (fig 3.1-3.2). This is a buoyancy-based clean energy solution that was conceptualized for offshore wind farms. It also includes the separation of CO_2, through deep sea hydrostatic pressure. The CO_2 system is conceptualized to take advantage of pressure to turn CO_2 into liquid at a depth of 500-1000 m (fig 3.3). This liquid can be separated into a chamber.

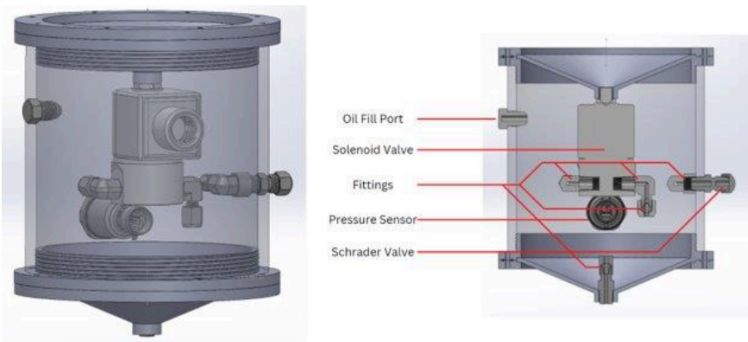

Figure 3.2. A CAD model to conceptualize the CO_2 separation device. This device is well thought out but lacks the mechanisms for real functionality. Our goal is to use this idea to create a physical device.

The existing work is theoretical with no blueprints for real-world testing. However, these concepts set the stage for producing an efficient, eco-friendly, and dual-purpose system that utilizes deep offshore regions to store energy and separate CO_2 at the same time.

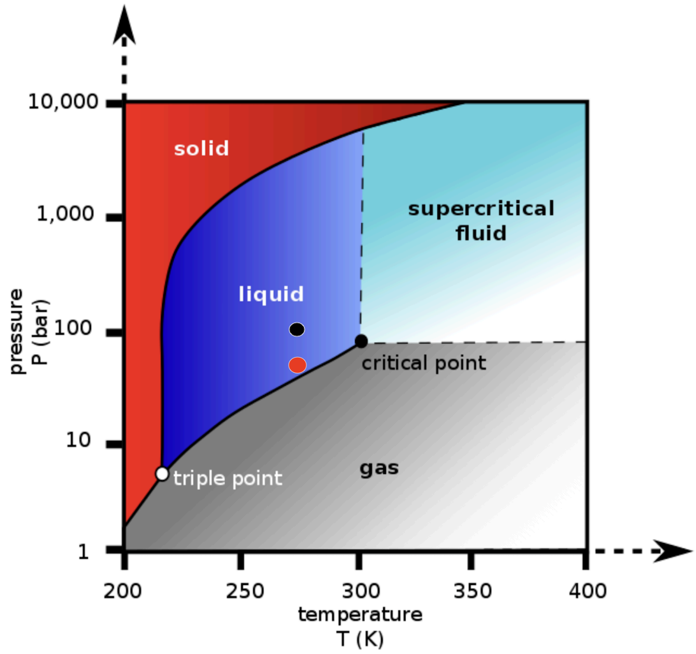

Figure 3.3. The phase diagram of CO_2[1.3]. The state this project would be aiming fo,r is marked with a red dot representing 500 m of depth at ~50 bar. The black dot represents the conditions at 1000 m of depth and ~100 bar. At both these points, the conditions are met for CO_2 to become a liquid.

Our project focuses on the CO_2 portion of the existing work and turning the existing concepts into reality. The main goal is to build a device to test the feasibility of underwater CO_2 separation. While it is based on the previous group's concepts (fig 3.2), our design will be unique, since it's a prototype and not the full envisioned system (fig 3.2). As this is an electrical and software project, the basis of this design is an electrical circuit that can read data from sensors at a depth of 500-1000m. A 3D-printed enclosure will also be designed to house these electronics and perform underwater tests. Due to our resources, testing will be done in a pool instead of the deep ocean. The design will be shown in more detail throughout the report.

In addition to building a prototype device, a second focus is expanding on the explainability of these concepts by building a software simulation. This will further explain the energy storage system and how the CO_2 separation system can be applied to it.

4. Methodology and Design

The methodology and design are broken into 4 sections. The first section is the simulation which is separate from the last 3. It relates to the overall goals but was completed separately. The next sections detail the 3D modeling, the electrical circuit, and the final testing procedure. Final deliverables include the prototype and the deployed simulation [3.1]. We also created a video detailing the process of building the prototype from assembly to pool testing [3.2].

4.1 Software Simulation

The software simulation is designed to showcase the solution for this project. It demonstrates the integration of the energy storage system with CO_2 separation and our prototype's use case. Originally, two separate simulations were planned, one for the existing energy storage system and another for the prototype use case. However, we opted to merge them into a single scene to offer users a comprehensive exploration of the project. Additionally, to enhance understanding, we included a third component illustrating the prototype's design, enabling users to inspect its internal components. The simulation serves as both a summary of physical deliverables and a method for presenting this project and the previous year's project as a combined entity.

The simulation leveraged Unity Game Engine [2.1] for development, which allowed for the integration and manipulation of 3D models made in Blender [2.3]. While Blender was primarily used for model creation, both Unity and Blender were employed for animation design, each suited for different use cases. For instance, Blender worked better for path animations, while Unity was employed for creating the exploded design view animation. These tools provided comprehensive support to flesh out the simulations and produce the final build effectively.

The completed build has been deployed [3.1] as a web application on Amazon Web Services [2.4]. It was also made into a standalone application compatible with MacOS. Although standalone applications offer slightly improved rendering capabilities, the web version's accessibility outweighed the marginal difference in rendering quality. The code is available on GitHub [3.3].

The simulation itself can be compared to a video game. It allows users to manipulate the scene using mouse controls and buttons (UI). This enables rotation, zooming, animations, and navigation within each section. Through these interactive features, users can delve into the details of each section, gaining a comprehensive understanding of the project.

The architecture of the simulation system (fig 4.1.1), employs the Model-View-Controller (MVC) design pattern, which is a widely recognized method for designing software with a clear separation of code. The View consists of the User Interface elements created using Unity UI Objects. It is responsible for displaying UI and recognizing user inputs such as mouse movements. The controller acts as the intermediary between the View and the Model, handling Unity 3D Objects. It receives inputs from the UI, triggers animations within the simulation, and communicates with the Model to fetch or return information. The Model stores animation states, calculations, and text information.

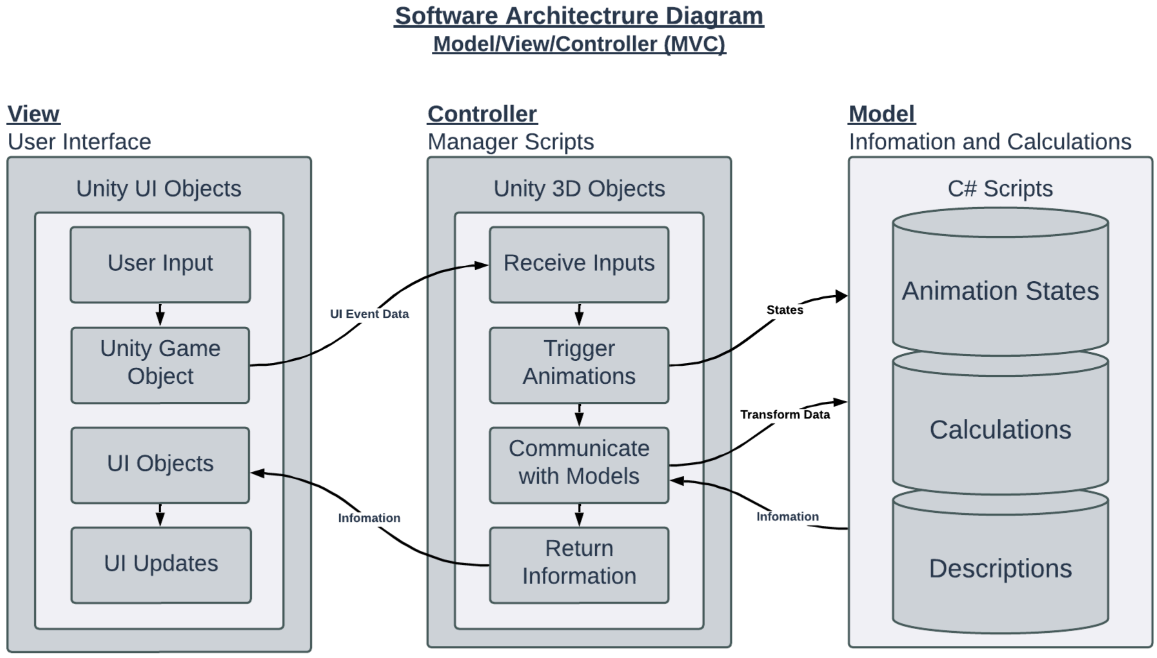

Figure 4.1.1. Architecture Diagram detailing the Model/View/Controller architecture (MVC).

The simulation components were created in Blender, where they were given materials and animations. For the design art (fig 4.1.2), many animation methods were used. This included shape keys to shrink the balloons on the prototype model. Shape keys are the ability to alter the mesh of an object and assign a value that morphs the object between two shapes. They aren't the same as animations but rather, can be used to create animations within Unity.

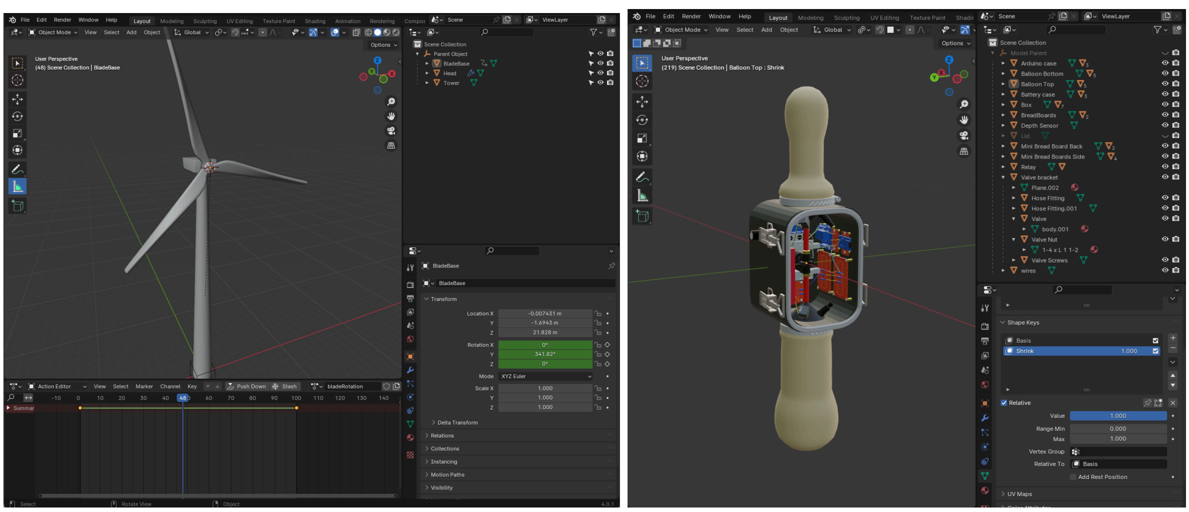

Figure 4.1.2. The Wind turbine (left) was given a rotation animation before export to Unity. The prototype design art (right) was designed with all the components having a material. It incorporated shape keys to change the mesh of the balloons. The top balloon has a shape key value that is shrinking it in the image.

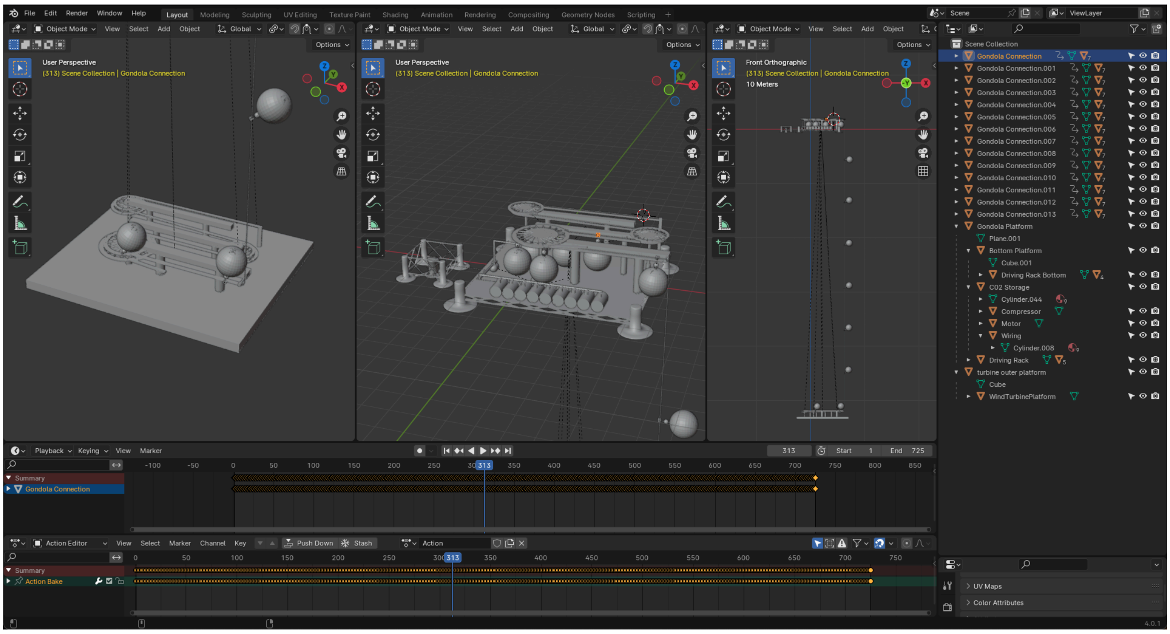

Figure 4.1.3. The Gondola platform for the system use case is being shown using 3 cameras in Blender. This technique would be replicated with the cameras in Unity for the final product.

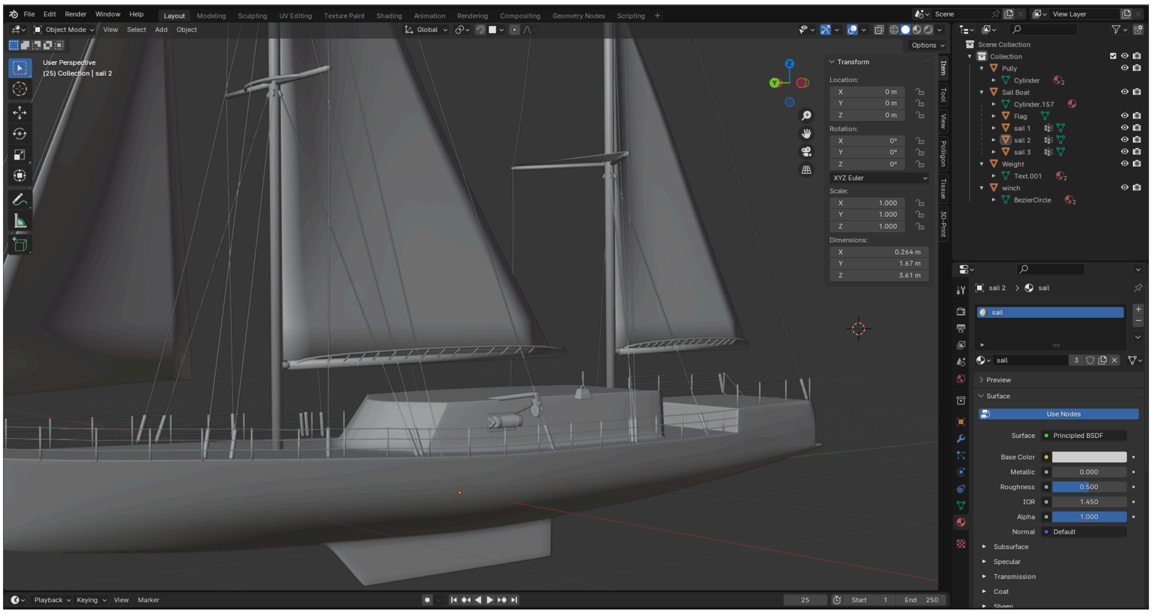

Figure 4.1.3. For the prototype type use case, a sailing boat was created to lead some authenticity to how the device could be used in a real-world setting.

The Simulation (fig 4.1.4) features 3 sections, the Prototype Design, the Prototype Use Case, and the System Use Case. The Prototype Design (fig 4.1.5) was added to the project goals midway through the project. We felt it would display the design both during the production of the physical model and allow a user to see the components that make up its structure. The Prototype Use Case (fig 4.1.5) Shows the real-world use of the device, which involves sending it 500 m underwater where the CO_2 can be separated. The System Use Case (fig 4.1.6) features the existing work on energy storage and incorporates the CO_2 separation work created in this project to showcase a combined system.

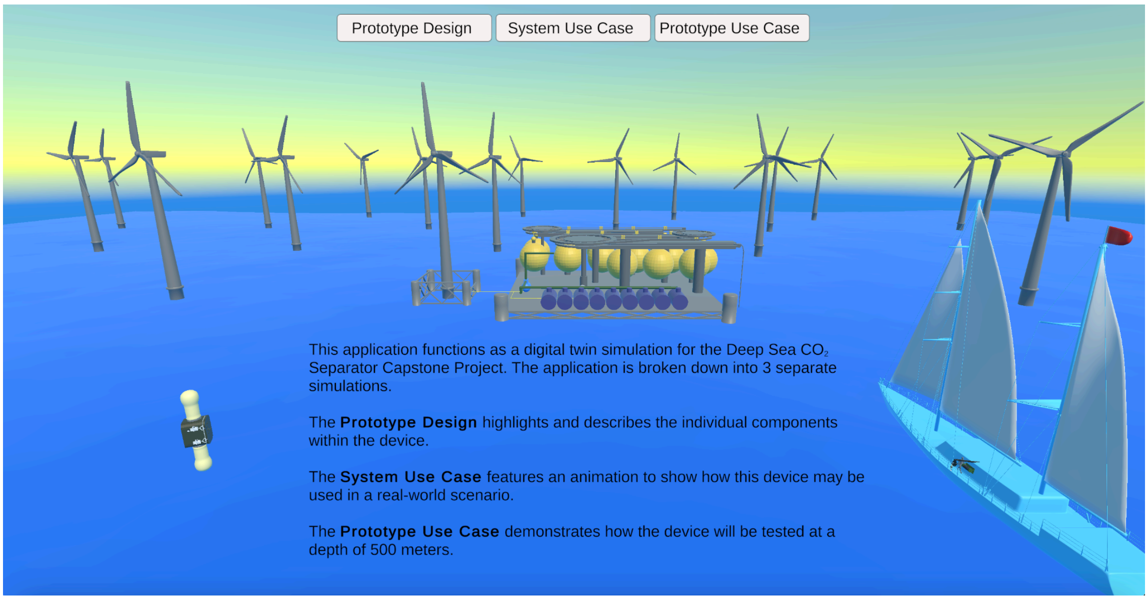

Figure 4.1.4. The main page for the simulation. This provides context for the 3 sections and the navigation at the top to snap to any of the sections.

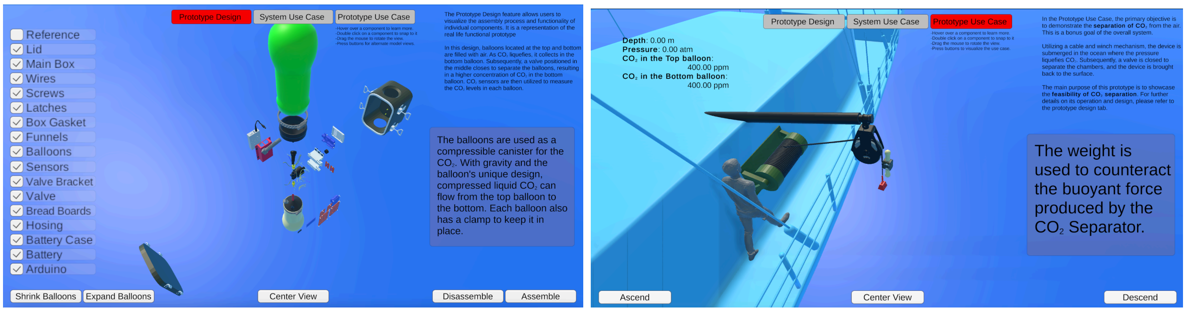

Figure 4.1.5. The Prototype Design (left) showcases an exploded view of this project's prototype. Highlighting each component displays a help box that explains its functionality. Buttons can change the view and play animations that help explainability. The Prototype Use Case (right) showcases the real-world use of the prototype. While this test wasn't completed during our project it is hoped that future work will allow a team to test the device in the ocean. Similar to the design section a user can play animations and hover over elements to see a help box with information.

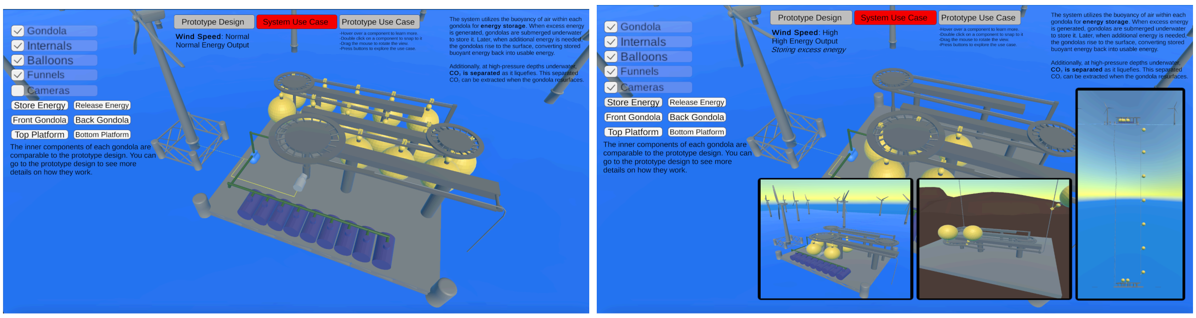

Figure 4.1.6. The system use case features a platform of gondolas filled with air. When wind energy is high the gondolas are sunk to a depth of 500 m where excess energy can be stored. When wind energy is low they can be brought back up harnessing the energy they are storing. As a bonus, each gondola contains a device similar to our prototype that can take advantage of the high pressure 500 m below sea level and separate CO_2. It also allows a user to toggle cameras showing multiple views and the scale of the system.

4.2 3D Modeling

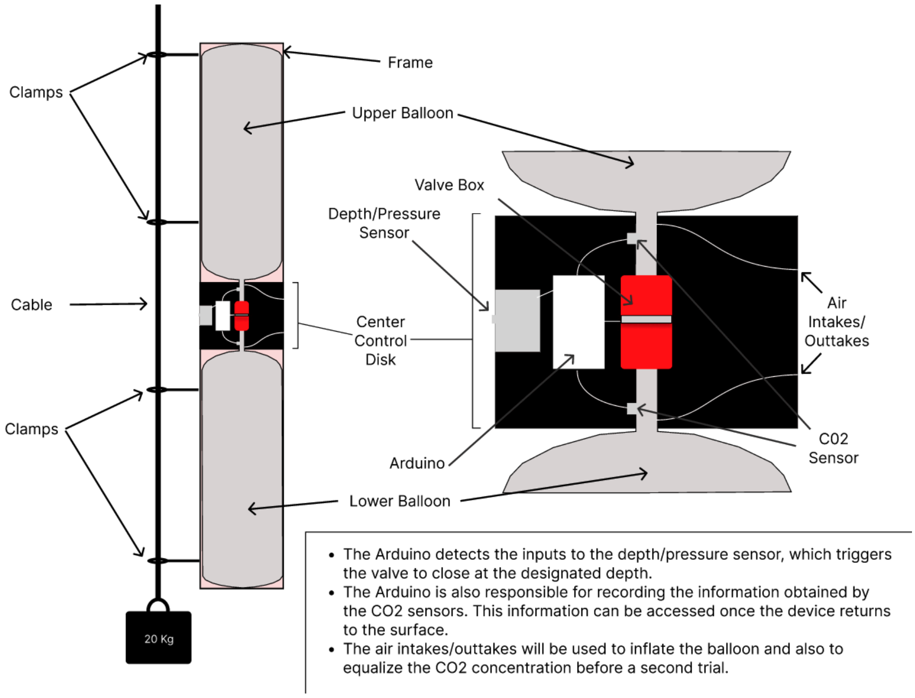

Figure 4.2.1. The initial design was modified as components were sourced. This diagram presents a depiction of the overall concept and the methods employed. Modeling led to changes to the methods but the fundamental design ideas remained consistent

While the main emphasis for the physical device was on the electrical circuitry, the 3D-printed housing (fig 4.2.2) was essential not only for testing the sensor in underwater conditions but also for delivering a design that met the necessary functional requirements. The 3D components served as a watertight container for the electrical circuit. It was designed using Blender [2.3] which proved to be an effective CAD tool, offering the benefit of simplifying the learning process since it was also utilized for the simulation aspects of the project.

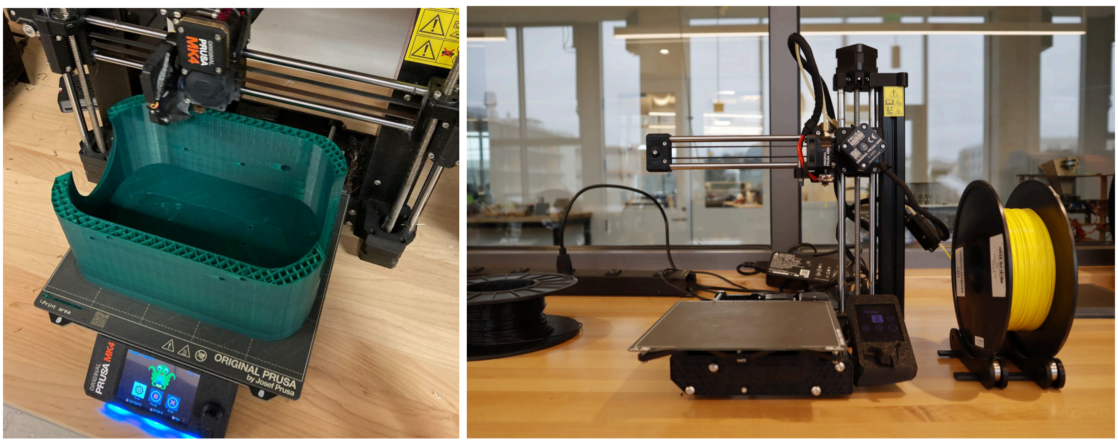

Figure 4.2.2. The box (left) and other components were printed at the Makerspace Lab [2.7] at the University of Calgary.

Six components of the box were 3D-printed and the rest were store-bought parts [4]. This includes the box, lid, battery case, valve bracket, and the two funnels. Tools such as screws and epoxy were used to attach the remaining components to the inside and outside of the box.

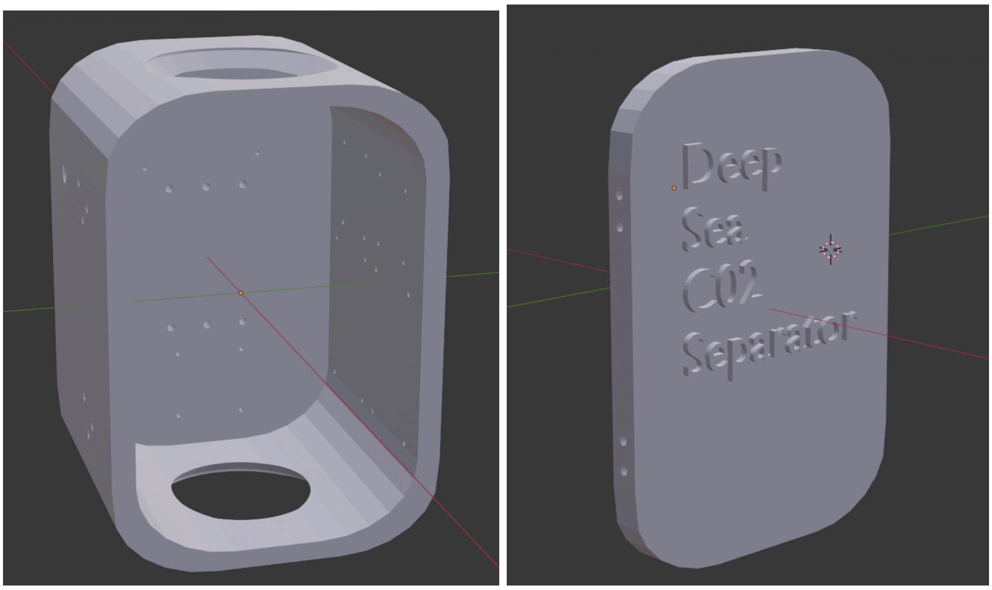

Figure 4.2.3. The 3D models for the box (left) and the lid (right). The funnels were added to the holes in the top and bottom of the box after they were all printed

The lid (fig 4.2.3) had holes on the sides to attach the latch clip. The box (fig 4.2.3) had screw holes on the outside for the latch, a hole through the wall for the depth sensor, and screw holes throughout the inside for breadboards and the other components. The valve bracket (fig 4.2.4) was designed to hold the solenoid valve in place. The Battery case (fig 4.2.4) was to hold the battery against the wall. The funnel (fig 4.2.5) was an extension of the box as it was later epoxied to the box. It was to attach the balloons and ideally serve as a path for liquid CO_2 to follow from the balloons.

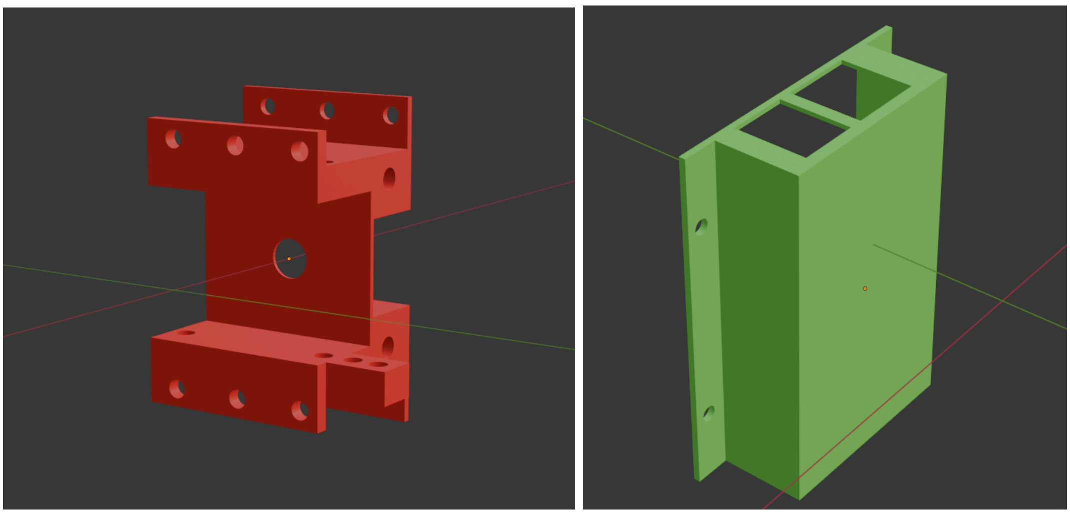

Figure 4.2.4. The valve bracket (left) was designed to hold the valve in place in the center of the design. It was screwed into the back of the box. The extra holes were designed to make wire management easier as the wires could go through the holes or be zip-tied to them. The battery case (right) was screwed to the left wall to lock the battery into place.

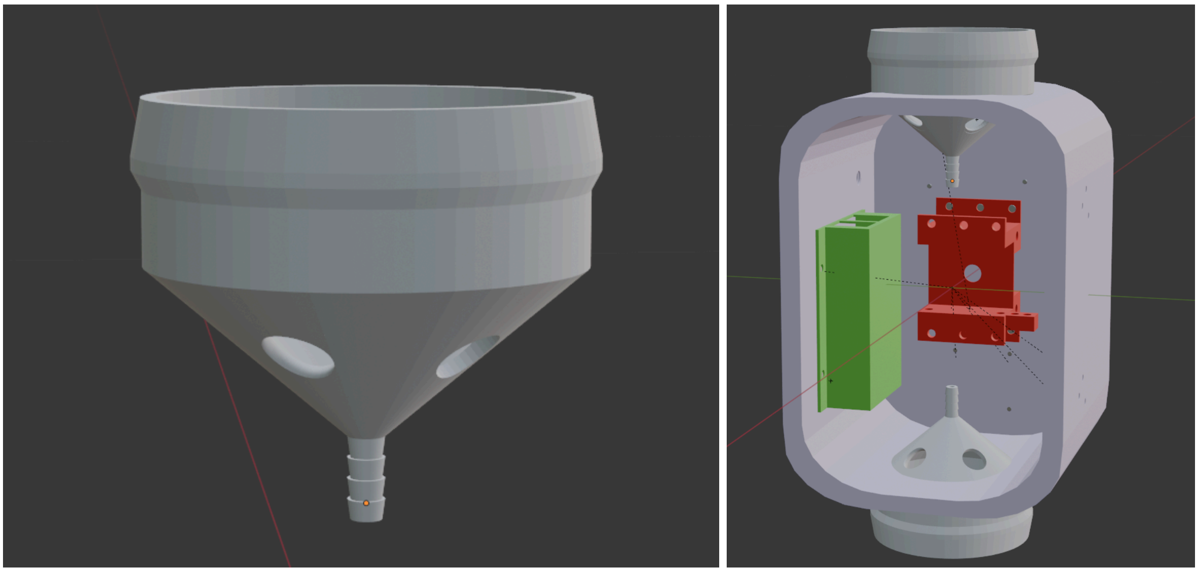

Figure 4.2.5. The funnel (left) was placed in the top and bottom hole of the box. It was designed to hold the CO_2 sensor and an air valve in the two holes on the side. The unique 3D-printed components without the lid are shown (right).

Once the modeling was completed the housing was assembled. The funnels were epoxied (glued) to the box and then the complete device was coated in flex seal (fig 4.2.7). This step was crucial since 3D-prints are very porous, letting water pass through easily. The sealant allowed us to print any shape and then waterproof it. Screw pegs were all attached to the inside of the box and the latches to the outside (fig 4.2.6). Finally, the balloons were attached (fig 4.2.7) using a clamp and also epoxy to ensure there were no leaks.

Figure 4.2.6. The images above depict the process of attaching the funnels with epoxy and adding a waterproof gasket to the rim of the box. This was a crucial set to waterproof the device. Inside the box, you can also see a peg attached to the walls with epoxy. These would allow us to screw in any components we need for the electronics.

Figure 4.2.7. After attaching the external latches, the box and lid were prepped for flex seal (left). The process of spraying the flex seal (right) allowed the box to become water-tight.

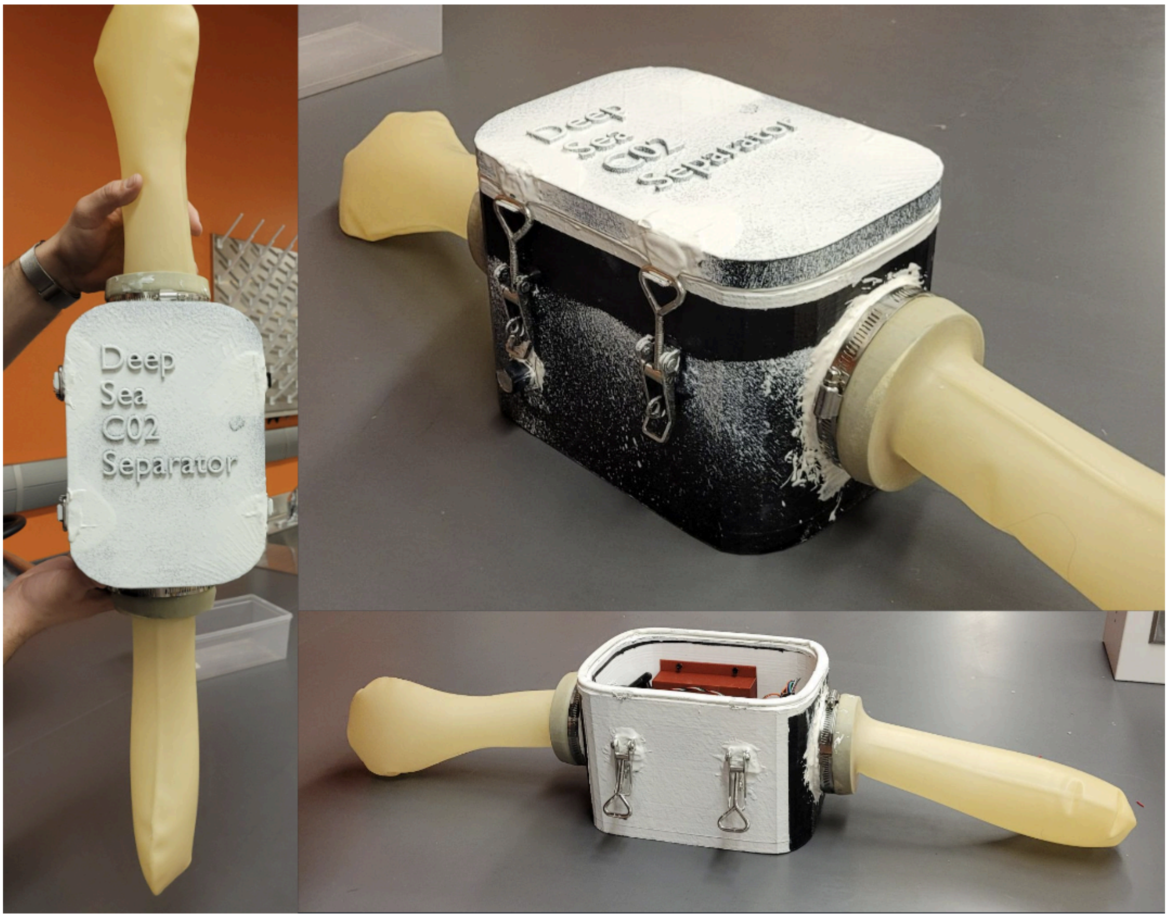

Figure 4.2.7. The balloons were attached to the funnels on both ends. They were epoxied at their base and then clamped around the base to ensure they would stay in place and stay watertight.

4.3 Circuitry

Circuit Diagram

The figure below (fig 4.3.1) is the overall circuit schematic used in the final prototype design. These connections were initially documented as pin configurations for each component before the creation of the circuit diagram and hardware implementation in the prototype. It shows the different connections needed for the Arduino to function and store data. It also shows how to power these components with either regulated, Arduino or power supply voltage.

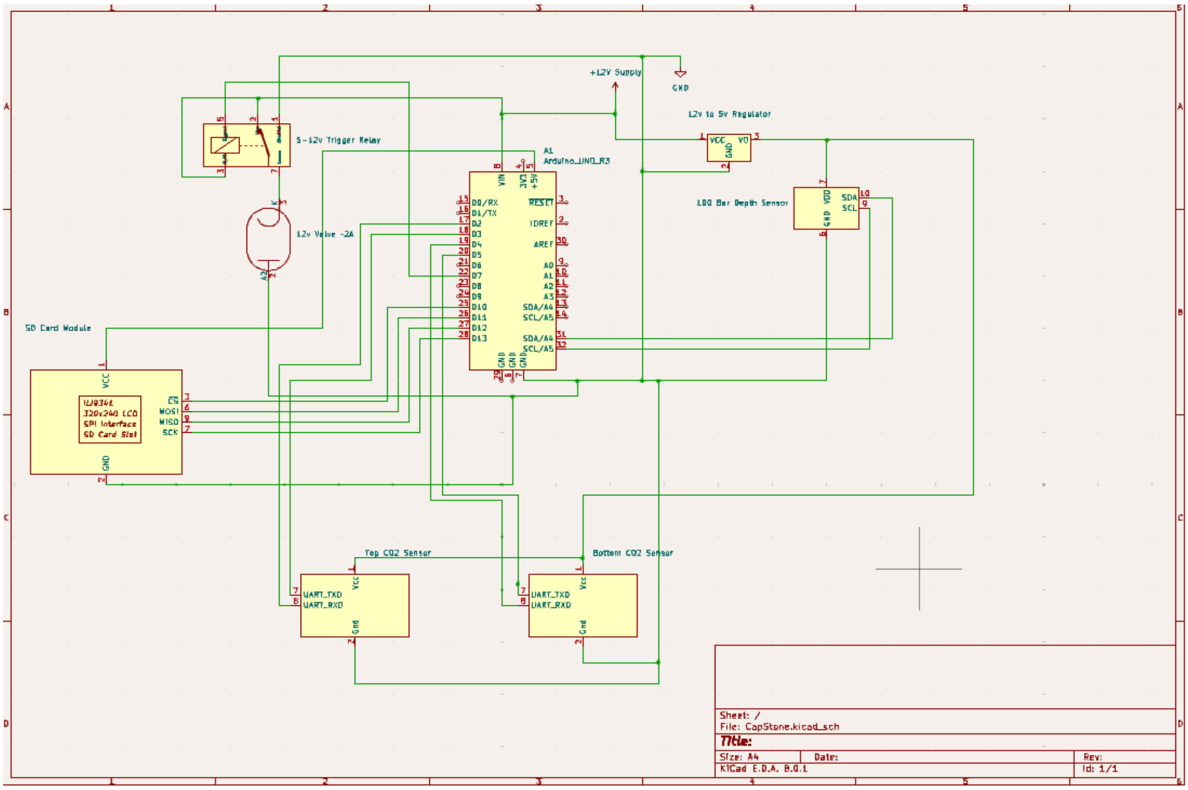

Figure 4.3.1. Overall Circuit Schematic based on initially written pin configuration. Show all the electrical hardware that was used for the prototype. Also shows how each component was powered (12v supply for Arduino, Relay + Valve, Regulated 5v for Depth and CO_2 sensors, Arduino supply for SD module) and the different connections to be made to the Arduino Uno.

As shown in Figure 4.3.1 there are several important components. The 12V Solenoid Valve [4.4] is used to separate the upper and lower chambers and is closed by default. Next, the two CO_2 Sensors [4.2], in combination with the Arduino UNO [4.1], can use UART communication to obtain data. The concentration range (0-50000 ppm) also gives a strong range of values. Last, the Depth/Pressure sensor [4.3] takes readings in absolute values which takes away variance. Its range of 0-100 bar also works in simple use cases like a pool and would work for a 500-1000 m test. It communicates to the Arduino using I2C communication and allows us to store data on an SD Card module [4.6].

Electronics Implementation

Overall, there were several iterations and sets of testing that were done to implement the circuit above into the final enclosure. The first set of work was implementing and testing the individual components. This involved testing basic functionality and power for each one. The primary purpose was to gain a better understanding of each component and ensure the theoretical diagram was consistent with using the hardware.

The next set of testing and implementation was using the entire circuit design (multiple components at once) on a solderless breadboard. The main purpose for this was to do more checks on the circuit design and make changes before switching to a soldered permanent system. This was a final check.

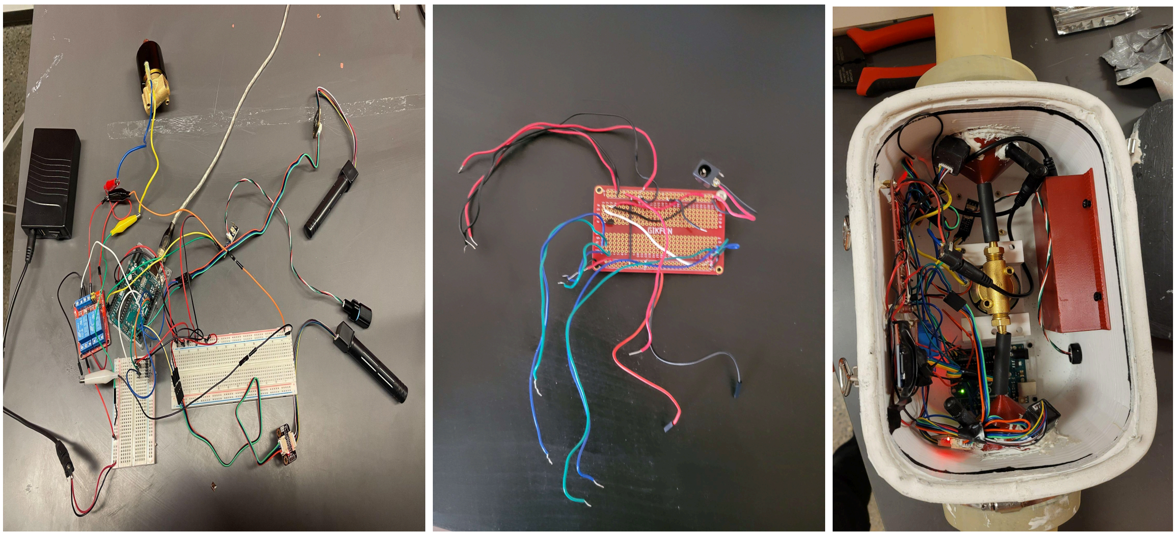

Figure 4.3.2. Testing schematic setup on Solderless board (left). Primarily testing to power and run all components at once. This setup was later transferred to the 3D-printed enclosure (right) before the pool test. The soldered breadboard (middle) is used in the final enclosure. Primarily deals with the different power connections for the component shown in (fig 4.3.1) and makes the connections from the CO_2 sensor to the Arduino easier.

Next was implementing the circuit on a soldered board. The final system needed to be secured in place inside the enclosure (fig 4.3.2 right) and therefore the wiring was soldered in place to prevent loose wiring. The same functionality tests were completed on this system both before and after it was mounted inside the enclosure. Connectivity tests were performed with a multimeter when adding connections. This was essential for preventing any short circuits and was something consistently used in this phase of implementation.

This circuitry was then mounted in the enclosure by screwing breadboards to the walls. This involved wire management by attaching wires to holes in the valve bracket with zip ties to provide a cleaner area. This was followed by more of the same testing and finally tests in and out of the water. A more detailed overview of the tests is seen below.

Testing Spreadsheet

The entire iteration and testing process was documented throughout the semester in a spreadsheet (table 4.3.1). This spreadsheet contains the components that were tested on certain dates, the test done for the components, the results, and what was learned from the test done. This style of documentation was important to have to keep track of componentry information and to address issues before the next iteration of testing. This allowed us to mitigate hardware issues arising during development.

	Date:	Item:	Test(s) Being Done:	Result?	Was it a success? (Y/N)	Lessons learned for next iteration(s):	Tester(s):
2	Jan.17 2024	CO2 Sensor	CO2 power up with Arduino	Sucessfully turned on. Indicator (LED) is on showing no changes to be made from original diagram	Y	How to setup multiple CO2 sensors with the arduino. Powered on from arduino power so no problems using regulated voltage	Cj+Fahad
3			CO2 Sensor communication with Arduino	Serial monitor on laptop when connected to the Arduino successfully shows the top and bottom CO2 concentration in ppm	Y	Pin setup thought of in circuit diagram still works as intended for CO2. Flexible to change but for now good to follow circuit diagram	
4			CO2 Functionality test (breathing test)	Serial Monitor shows readings increasing when breathing into the sensor as intended with the test	Y	CO2 can read change to CO2 levels. After breathing does take a period of time to get initial CO2 readings compared to fast increase. Potentially readings are over a period of time?	
5		Arduino + Power Supply	Testing Compatibility of arduino use while not wired on computer	Arduino power indicator (LED) turns on when the supply is connected and powered on	Y	Works as intended, no change needed to be made from current schematic	
6	Jan.23 2024	Valve	Powering up the Value with Power supply	Valve clicks on when provided power from the power supply	Y	Valve click is very loud, could be useful as an indicator for others sets of testing. Valve works with psu as intended	Cj+Fahad
7		Valve+Relay+Arduino	Use of Valve with a digital pin from the Arduino	Was able to turn on the Valve using a digital pin from the arduino when connected with the relay	Y	Slight modication to the circuit diagram made. Initially thought a 5v source should be enough to power the relay. Now diagram has 12v supply to power relay. Relay functionality works as intended with the Valve tho	
8			Intervaled use / Control test with the Digital pin Controlling the valve	Code made to control and flash Valve (turn on and off at a set interval) using the same arduino digital pin	Y	Now have an idea (code-wise) to control the valve to be on and off. Schmatic setup thought of works as intended after the power change to the relay.	
9	Jan.25 2024	Valve+CO2+ Power Supply+ Arduino	Power test to see that all components previously tested can be operated at once	Able to power on the Valve (on and off) while LED indicator were on showing CO2 sensors are on	Y	No changes to the circuit schematic needed to be made. Sucessfully combined some portions of the CO2 and valve code	Cj+Fahad
10		CO2 Sensor	Testing different intervals with sensor - arduino Communication	tested out effects of different delays on the reading of CO2 concentration and picked one out for better future testing (fast but still visible)	Y	Set intervals now are more ideal to see when verifying on the serial monitor. Delay affects how fast readings are taken	
						By itself the depth sensor will not work with the Arduino. power wise will work but the	

Table 4.3.1. The spreadsheet documenting the testing suite. The document records the date of testing, components tested for, what test was done, results of the test, and the different takeaways from doing the test. The testing document was used to record overall progress and show where potential setbacks can come from

Power of Components

Depending on the scenario in a test to drop the prototype in the water the amount of power consumption will also differ in time. Below is the amount of current each component will need to run based on the pin connections (fig 4.3.1):

- 12v Solenoid Valve: < 2.0A
- 5v UART Infrared Carbon Dioxide Sensor: 85mA
- 5v 100 Bar Depth Sensor: 10mA
- Arduino Uno: 50mA
- H/L trigger relay: 5mA
- SD Card Module: ~50mA (powered by Arduino, not the 12v supply)

Based on the stated component consumption, the amount of current drawn from the power supply can separated into two cases :

- The active mode of the system in which the valve is opened. This mode is used while descending. The consumption from the power supply in this mode is about 2A.
- An idle mode in which the Valve has been shut off and closed due to the enclosure reaching a pressure greater than the given threshold. This mode of power

consumption will be in effect when the prototype is ascending. The consumption from the power supply in this mode is less than 250mA

When considering power consumption, the duration for which the system can be powered depends on various factors. Key considerations include the duration of the descent and ascent cycles during testing, as well as the set threshold for valve closure.

Conservative/high consumption assumptions are that the device is ascending 50% of the time and descending 50% of the time and the threshold is reached at the bottom of the descent. With these assumptions, the system can still be powered off for trials over 2 hours. For ways to save power consumption, the most ideal change would be having the threshold to close at a lower pressure. A power-saving alternative (code also provided) would be to leave the closed during most of the descent and open only near the threshold pressure. This would lead to spending less resources during operation. The code descriptions and diagrams below will be based on the final code and not the power-saving alternative.

Electronics Code Format

Another crucial aspect to consider alongside the hardware is the software code that will govern its operation. In our case, programming was done in C++ due to the Arduino serving as our central board. The focus of electronic coding was to meet specific requirements based on the capabilities of the prototype. These requirements include:

- Reading and recording data from two different CO_2 sensors
- Reading and recording data from the pressure sensor
- Control of the solenoid valve based on pressure sensor readings
- Format recorded data to be able to be plotted later in a CSV file

Some of the code snippets are shown in the figures below. The code begins with setting global variables for the pins and components required by the rest of the code (fig 4.3.3). The valve is initially set as active (open). The rest of the sensors are also initialized and their inputs are recorded in the system's main loop. This loop also records data to the SD card.

The best method to record data was to record multiple CSV files for each sensor and valve. This was a best practice for SD cards and the Arduino and also allowed us to separate the code functionality.

```
#include <SPI.h>
#include <SD.h>
#include <SoftwareSerial.h>
#include <Wire.h>
#include "KellerLD.h"

KellerLD sensor;
SoftwareSerial CO2Serial(2,3);   //Rx, Tx
SoftwareSerial CO2Serial2(4,5);   //Rx, Tx
unsigned char hexData[9] = {0xFF, 0x01, 0x86, 0x00, 0x00, 0x00, 0x00, 0x00, 0x79};  // read gas density   command /dont change order

//change this to match your sd module
const int chipSelect = 10;
int solenoidpin = 7;  //Set valve and arduino connection to pin 7
File dataFile;  //CO2 #1
File dataFile2; //CO2 #2
File dataFile3; //Pressure
File dataFile4; //Valve state
```

```
void setup() {
// Open serial communications and wait for port to open
Serial.begin(9600);
// check for error in SD card
if (!SD.begin(chipSelect)) {
  Serial.println("Card initialization failed!"); //error in SD card initialization
  return;
}
//print card initialized if no error
Serial.println("Card initialized.");

CO2Serial.begin(9600);  //initializes serial communication between first CO2 sensor and arduino

Wire.begin(); //initialize I2C communication        //needed for data(SDA) and clock(SCL)
sensor.init(); //pressure sensor initialization
sensor.setFluidDensity(997);   // kg/m^3 (freshwater, 1029 for seawater)

//check for pressure sensor initialization
if (sensor.isInitialized()){
  Serial.println("Sensor connected"); //initialized
} else {
  Serial.println("Sensor not connected"); //not initialized
}

pinMode(solenoidpin, OUTPUT); //set pin for valve as output
digitalWrite(solenoidpin, HIGH); //Open valve
```

Figure 4.3.3 Code for Libraries, Global variables, and Program Setup for components. The setup runs only once when powered on or when reset. After setup, the code runs a loop.

After the initial setup, the key functionalities are coded in the main loop. The structure for the code focuses on reading and getting values from the different sensors one by one before writing to the CSV files at the end of the loop. The first sensor that is read for is the Pressure Sensor (fig 4.3.4). This is crucial because the pressure readings in this portion of the loop will also be used for updating the status of the valve. Having this early also marks a good position to check if the setting of the valve needs to be changed.

```
sensor.read();  //reads data from sensor
delay(500);
int pressure = sensor.pressure();  //retrieves pressure data
String pdata = String(pressure);
if (sensor.pressure() > 1100){ // should open around 2.1-2.2 meters underwater
  digitalWrite(solenoidpin, LOW); //close valve
  closed = 1; // set variable to print valve state
 }

//prints pressure in mbar

Serial.print("Pressure: ");
Serial.print(sensor.pressure());
Serial.println(" mbar");
```

Figure 4.3.4 Pressure sensor code. Reads pressure from the pressure sensor and closes the valve when the pressure exceeds the preset limit, 1100 mBar in this case.

The next portion of the loop is the reading of both CO_2 sensors (fig 4.3.5). A key insight to note is that the readings for CO_2 concentration are taken one by one. This is due to the Arduino Uno having to do UART communication one component at a time. This means the Arduino will start communicating with one CO_2 sensor, get the values from that sensor, and store them in a CSV file. It will then communicate with the second CO_2 sensor and do the same.

```
CO2Serial.write(hexData,9);
delay(500);
//CO2 #1
for (int i = 0; i < 9; i++){
  if (CO2Serial.available() > 0){  //if sensor setup properly run
    long hi, lo, CO2;
    int ch = CO2Serial.read();  //read values from sensor
```

```
  if (i == 2) {
    hi = ch;  //High concentration
  }
  if (i == 3) {
    lo = ch;  //Low concentration
  }
  if (i == 8){
    CO2sens1 = hi * 256 + lo;  //CO2 concentration
    // Print sensor values
    Serial.print("CO2 concentration:");
    Serial.print(CO2sens1);
    Serial.println("ppm");
    CO2Serial2.begin(9600); //Open communications with second CO2 sensor
  }
} else {
  Serial.println("CO2 sensor1 not working");  //print error if not working
 }
}
```

Figure 4.3.5 Code to read from CO_2 sensor 1. Reads the value from the first sensor, and records it in the CSV file. This is then repeated for the second sensor.

The last important section of the loop is the area where the data is being written to the different CSV files. As stated earlier, multiple files are generated and deal with a different set of data in each file. This was done for easier use of the SD Card and Arduino configuration. There are multiple functions to write to CSV (fig 4.3.6) but they all follow the same pattern but with different data and a different file output.

```
//Write CO2 #1 values
void writeDataToCSV(String data) {
  //Open the file
  dataFile = SD.open("data.csv", FILE_WRITE);
  //If file opened okay, write to it
  if (dataFile) {
    dataFile.println(data);
    //Close file
    dataFile.close();
  }
  else {
    //If file didn't open, print an error
    Serial.println("error opening data.csv");
  }
}
```

Fig 4.3.6 Function responsible for writing data to the CSV file. There exist similar functions for both CO_2 readings, the pressure reading and the readings for triggering the valve.

State Diagrams

Below are some visual representations of the code described in the previous section. First is the UML activity diagram (fig 4.3.7) which follows the flow of information from initializations to the writing of data to files. It also shows potential endpoints for the program and the reasoning for it. Next to note is the State diagram (fig 4.3.8) which focuses on the different transitions occurring in the program. Specifically, it shows the different occurrences in the loop, the command that represents each transition, and the reasoning for each change occurring.

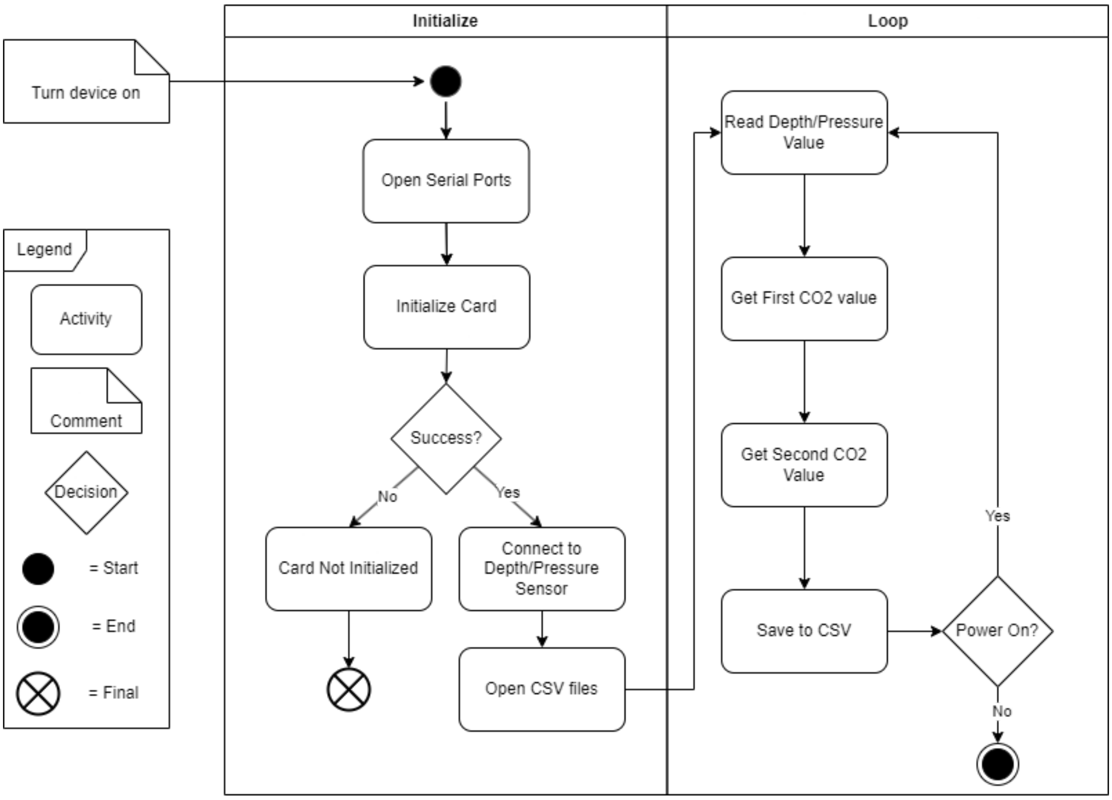

Figure 4.3.7. UML diagram for how the code runs. The programs start with setting up the sensors and components. It then initializes the SD card and creates the CSV files to write data into later. In the loop, the data is read and then sent to the CSV files for recording. The system ends when powered off or if the card is not initialized.

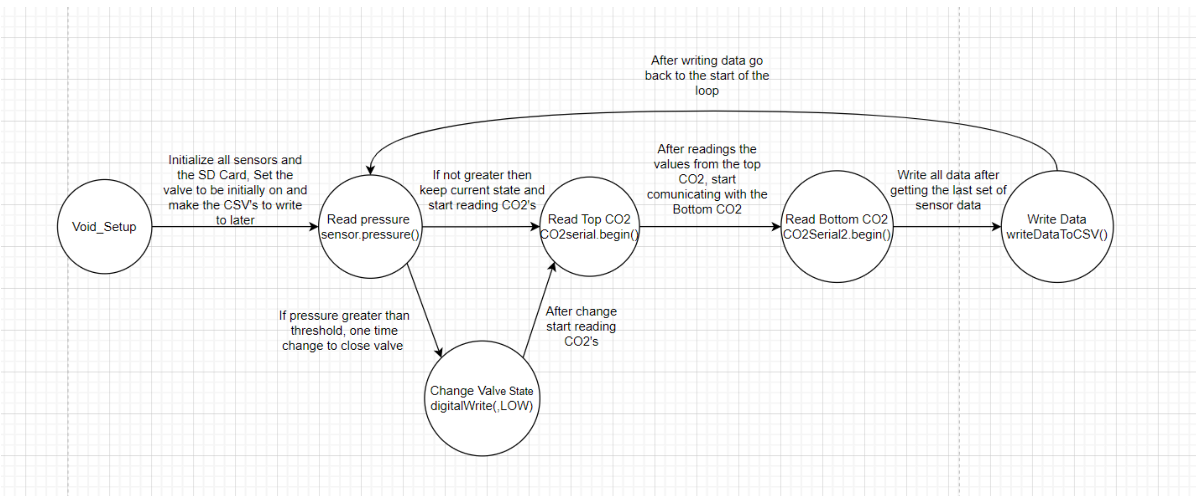

Figure 4.3.8. State diagram representation of the stated code. The title for each state is a representation of the function/command to do the process of the state name. The diagram shows the process of the program with more focus on the occurrences in the loop. Things to note are how the check for the state of the valve is done and the order in which sensor data is read.

4.4 Final Testing

Once the device was assembled with the electronics it was sent to a pool test. Two tests were performed. The first took place in 1.5 m deep water and the second test took place in the 5 m deep end of the pool. A GoPro [2.5] was used to record a video of each test.

The first test (fig 4.4.1) was meant to make sure the sensors could record data and fire the valve at a threshold depth. It served as a baseline for the second test. The goal was to force the device 1.5 m down and listen for the click of the valve. Data on the SD card was checked afterward to ensure we could record and double-check the valve closed. These results served to guide the plans for the second test as we made adjustments to the code to better record results.

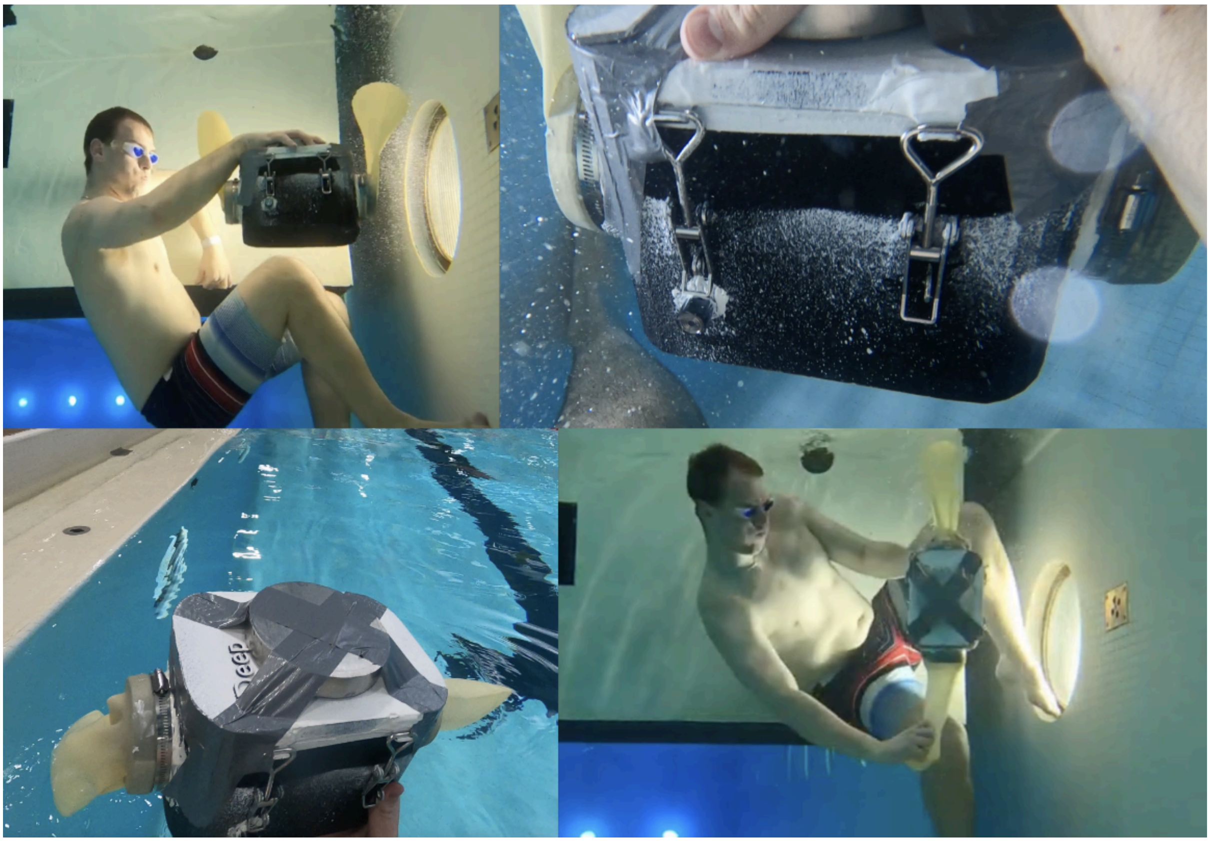

Figure 4.4.1. The first pool test. A weight was added to the device to make it easier to sink. This test was to make sure the valve could fire and the box could keep the electronics dry. The pool was only 1.5 m deep which let us repeat the tests easily. The valve was set to fire when the depth sensor was at 0.75 m. The valve was also reset when ascending so the device could be tested multiple times in a row.

The second test (fig 4.4.2) was the final version of the testing. It was done in deeper water and had a better setup to record data. One difference in this test was the placement of the weight. It was strapped to the bottom of the device with duct tape and served as a point where the tester in the pool could grab it and drag it down. The threshold depth was set to 2.1 m. This time the open water prevented hearing the click but data was recorded during 3 trials for CO_2 concentration in each balloon, the depth reached, and the moment the valve was closed.

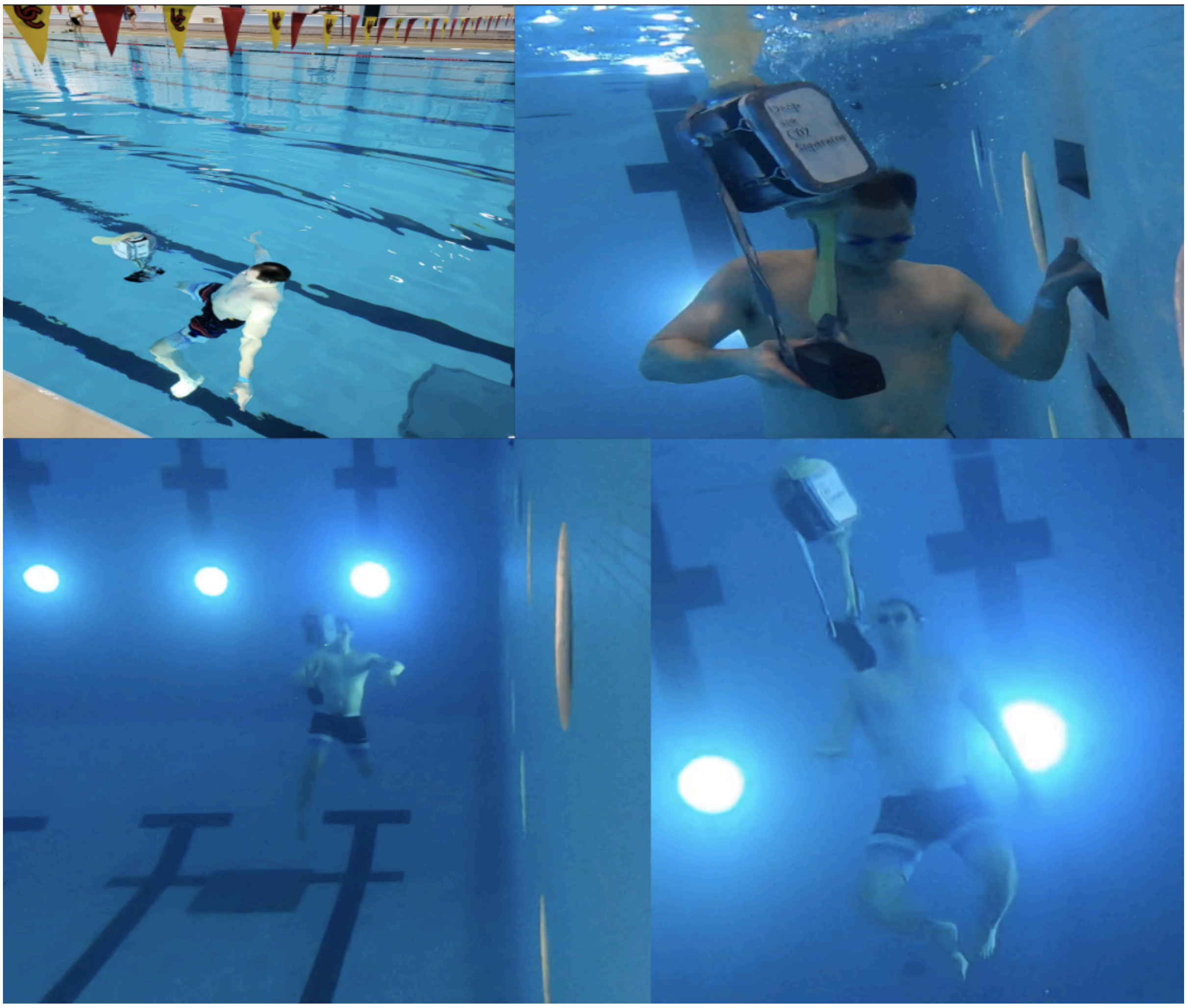

Figure 4.4.2. The second pool test was done at the 5 m end of the pool. This allowed us to go deeper with the test. For this test, the valve was set to close at 2.1 m and remain closed until the system was reset. The max depth the device reached was 2.7 m.

The final testing phase in a controlled pool environment was critical in validating the functionality and robustness of the device. Through two tests at varying depths, we successfully confirmed the sensor's data recording capabilities and the operation of the valve. The initial test in shallow water provided essential baseline data and informed necessary code adjustments, while the test in deeper water demonstrated the device's performance under deeper conditions. The adjustments made to the device, including the strategic placement of weights, ensured reliable and repeatable test conditions.

5. Product Scope and Functionality of the Final Product

The end goal of the project was to create a prototype device capable of separating CO_2 from the air using hydrostatic pressure. A simulation was also planned. Initially, five deliverables were decided including an initial prototype, final prototype, gondola simulation, prototype simulation, and user manual for the prototype. It's important to note that the final prototype was a hopeful goal and the initial prototype was the minimal viable product (MVP) for the physical portion of the project. These deliverables changed throughout the project and are discussed below.

5.1 Initial Prototype (Original Deliverable)

The MVP of the physical device was a 3D-printed version of the design with working electronics that could be tested in a pool rather than the deep ocean. While at this depth CO_2 separation would not be possible due to lack of pressure, the electronics were to be fully functional and capable of detecting pressure, CO_2 concentration, and the closing of the valve due to the pressure. The initial design (fig 4.2.1) showed the device as having two large balloons on the top and bottom of a central box. The central box houses sensors, a valve, and the rest of the electronics. The box was meant to be waterproof to keep the electronics dry while performing a pool test to close the valve and record data.

The initial layout didn't change while the methods used to create it did. The prototype was completed as required and its details are discussed in the methods and design section of the report.

5.2 Final Prototype (Original Deliverable)

The final version was a hopeful but optional goal. The design was the same as the initial prototype with the key difference being the housing. The electronics would be the same in both versions but the housing in the final version would be better suited to high pressure. Its purpose would be to perform a real-life test aboard a ship owned by the Sea Education Association [1.2]. Since the device would be sent to the coast it was also supposed to fit into a pelican case for transport. Its goal was to perform a true test of the feasibility of CO_2 separation. The deliverable was removed from the final deliverables as discussed in the following section.

5.3 Gondola System Simulation (Original Deliverable)

The gondola simulation was meant to create a virtual application to showcase existing work from the previous year's project on this topic. The simulation focuses on the concept of energy storage. The design stores energy by submerging a series of buoyant air containers (gondolas) along a cable system to a max depth of 1000 m. This energy will then be retrieved by releasing these buoyant air containers and using the stored potential energy to run a generator. As a bonus separating CO_2 from air is possible at this depth inside of each gondola. This idea was previously depicted in animation and concept art but the goal of the simulation was to take the concept and turn it into an interactive application that better showcases the use case. It aimed to provide explainability to the system and also highlight the CO_2 separation aspect of the system more than was previously explored. A previous system efficiency of 92.83% was calculated for the system and we aimed to show how this efficiency could change concerning the size of the gondolas for instance if a gondola was the size of our prototype.

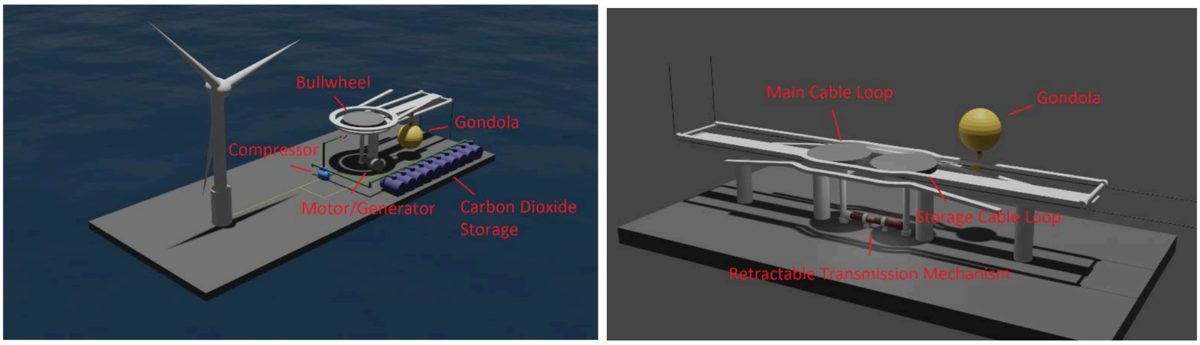

Figure 5.3.1. The concept art from the previous year's capstone project depicts the basic concept of the gondola system. These schematics provide an initial understanding of how the system operates. The simulation aims to make this system interactive and improve the use case by incorporating the CO_2 separation system.

5.4 Prototype Simulation (Original Deliverable)

This simulation aimed to showcase the use case of the prototype. One of its goals was to create a virtual system that could show the concentration of CO_2 in each balloon depending on the depth the prototype achieves. It also shows the physical effect on the prototype as depth changes. This section was to showcase the ideal capability of our prototype and provide a user with explainability.

5.5 User Manual (Original Deliverable)

This deliverable was a manual intended to be given to the crew on the ship that would use the final prototype. It would detail assembly instructions, how to conduct the tests, and record the results.

5.6 Main Deviations

Throughout the semester, multiple deviations were made to the project, and instead of having the original five proposed deliverables, there were three in the end.

- The initial prototype was completed as intended.
- The final prototype was not completed in its entirety, but the electronic design used in the initial prototype can be used in future projects.
- The gondola system simulation and prototype simulation were combined into one application. This did not change the requirements of each section of the simulation
- A third aspect was included in the simulation. This was a detailed exploration of the prototype design.
- The user Manual was adjusted to reflect the MVP iteration

5.7 Prototype Changes

The biggest changes were to the scope of the physical design. As more research was conducted and materials studied, it became clear that creating external housing that could survive deep ocean pressure was outside the budget and skill requirements of the group.
The group creating the design was made up of software and electrical students who did not have the knowledge required to make the mechanical design.

The project shifted to completing the initial prototype design with a focus on creating an electric system (fig 5.7.1) that would work for a pool test. The 3D-printed housing could be easily adapted to work with a future version. Ideally, a future group will take the electronic design and apply it to improved housing. This marked a renewed focus on the MVP version of the design and was still within the sponsor's original goals for the project.

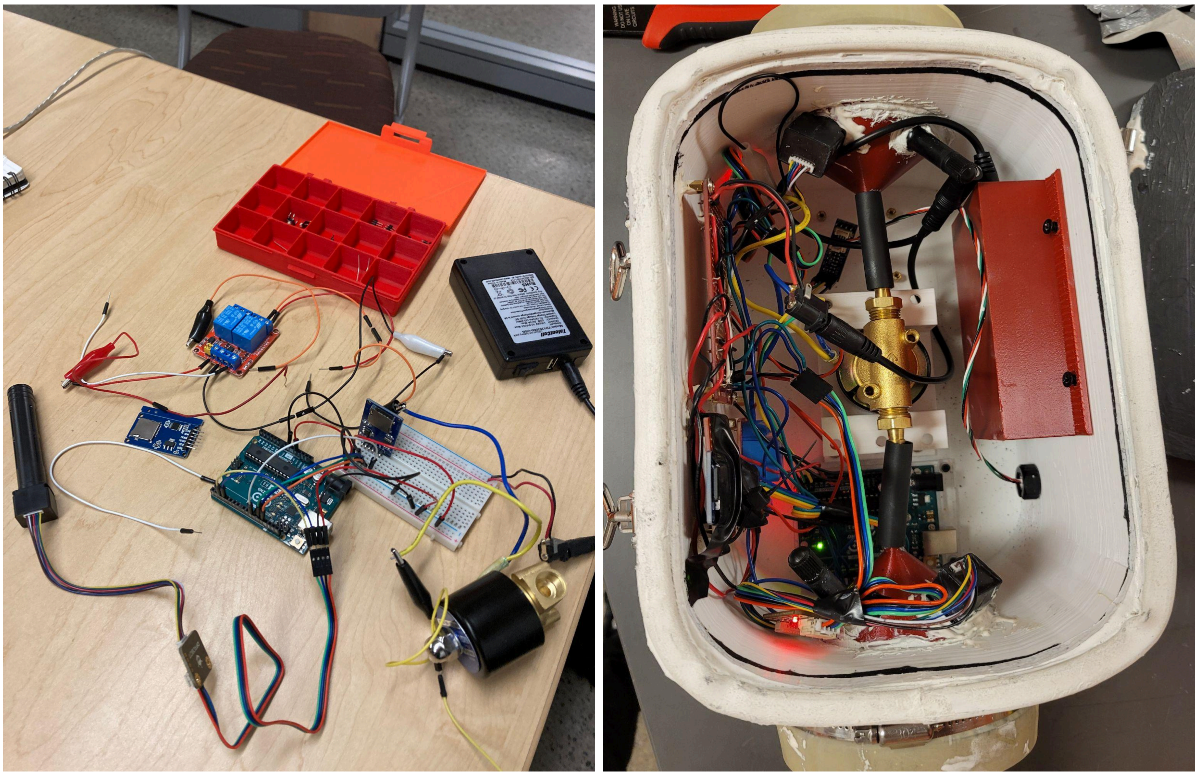

Figure 5.7.1 The circuitry system external to the housing (left), and the circuitry mounted inside the housing (right). The electronics became the focus for the prototype and the housing was created to test the electronics.

5.8 Simulation Changes

A slight change was in how the simulations were viewed. Instead of two separate simulations, it was decided to combine them into a single scene where users could quickly toggle between the different sections. This allowed all the points of the simulation to be seen together since all the components were related. The prototype design was incorporated into the gondolas so that the combined system could be seen more clearly. The gondola simulation was referred to as the Overall System Use Case. Another change to the system use case was to simplify the information being shown to the user. Efficiency calculations were deemed to distract from the goal of showcasing the use case. This information was removed to focus on the primary system and how it would work in more general terms.

A third section related to the design of the prototype was added (fig 5.8.1). This section was aimed to showcase the physical design of the prototype and allowed the user to understand the work that went into creating it.

Finally, one final addition was to deploy an online version of the simulation rather than just a user application. This web application allowed anyone with the link to access the simulation and explore it.

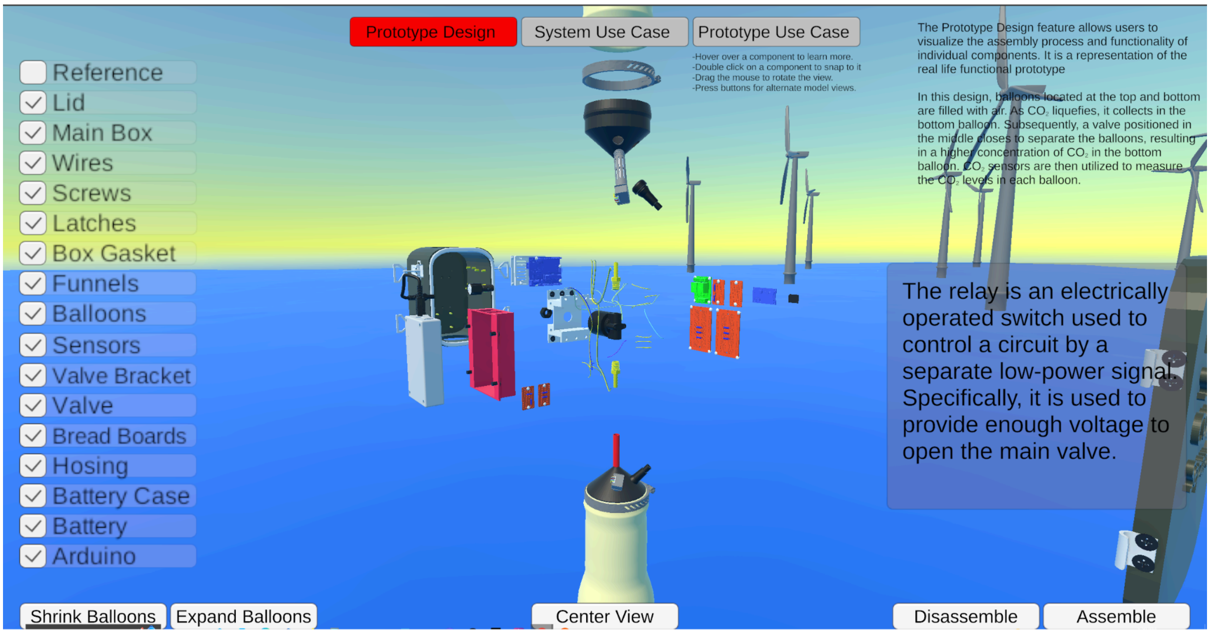

Figure 5.8.1 A screenshot of the disassembled MVP prototype with the cursor hovered on the relay to provide its respective information and explain usability on the right. This was an addition to the simulations to showcase the physical design of the prototype.

5.9 User Manual Changes

Given that the final prototype was not accomplished, it didn’t make sense to create a user manual of something that didn’t exist and wouldn’t be sent to the ship. Thus, the user manual was adjusted to accommodate the use of the MVP for anyone who wishes to continue working with the device. Most notably, the maximum tested depth is listed instead of the original target depth of 500 meters. The user manual now includes the following sections:

- Included components - A list of descriptions for each component with a corresponding labeled diagram.
- System requirements - The requirements a user must meet to get the most out of the device.
- Setup/Proper usage - A set of instructions to set up and use the device properly.
- Functionality - A list of specifications for the critical components.
- Maintenance and Care - Explains the proper handling of the device post-submersion.
- Appendix - Additional photos of the final prototype.

6. Technical Specifications of the Final Product

The following section outlines the technical specifications regarding the final product, along with the necessary deviations from the original set of specifications. The tabular structure used to illustrate the final specifications starts with the assembled device, followed by the specifications of each component.

6.1 Final Technical Specifications

Final Product (CO_2 Separator) [3.1]-[3.4]	Dimensions (L x W x H) - 16.6 cm, 16.6 cm, 24 cm Box Thickness - 1 cm Interior Volume - 6.6 Liters Total Weight - 2.133 kg Max Tested Depth - 2.7 meters Max Tested Pressure - 1150 mbar Max Tested Hydrostatic Pressure ~ 270 mbar Atmospheric Pressure on Date of Test ~ 870 mbar Material - ABS Plastic Budget - $2,000 CAD
Arduino Uno Rev3 [4.1]	Operating Voltage - 5V Input Voltage - 7-12V Digital IO Pins - 14 (6 of which provide PWM output) Analog Input Pins - 6 SRAM - 2 KB Clock Speed - 16 MHz
CO_2 Sensor [4.2]	Operating Voltage - 4.5-5.5V Measurement Principle - NDIR (non-dispersive infrared) Measurement Range - 0-50000 +/- 100 ppm Response Time - < 30s Average Power - < 430 mW @ 5V Operation Temperature - 0-50℃
Depth/Pressure Sensor [4.3]	Operating Voltage - 2.5-5.5V Maximum Mechanical Pressure - 350 bar Operating Pressure - 0-100 bar Operating Depth - 0-1020 m Absolute Accuracy - +/- 500 mbar (5.1 meters in freshwater) Resolution - 3 mbar (3 cm in freshwater)
Valve [4.4]	Operating Voltage - 12V DC Port Size - 0.25” NPT Orifice Size - 2.5mm
Battery [4.13]	Voltage - 12V Capacity - 3000 mAh Rechargeable - Yes

It should be noted that the “Max Tested Pressure” value for the Final Product is the measured value from the final set of conducted tests. This can be validated by summating the hydrostatic pressure from the “Max Tested Depth” and the atmospheric pressure on the day of testing [1.10].

6.2 Alternate Methods

Originally, the device dimension was set to be cylindrical with a diameter of 25 centimeters. This was to be used as a tube attached to an existing hydro carousel (fig 6.2.1) [1.4] aboard a research vessel [1.2]. Putting the device on the existing frame would simplify the lowering methods. It did require us to specialize the design to fit in the space provided by the hydro carousel. Since the space was small it was decided that we couldn't effectively design the prototype to fit in this space. This combined with the adjusted scope of the project led us to abandon this method.

Figure 6.2.1 An example of a hydro carousel. This device has cylindrical tubes attached to a frame. These tubes are utilized by deep-sea experiments. The exact version we could have had access to was a much smaller design.

6.3 Previous Technical Specifications

The following were the original technical specifications. The reasons for the changes are stated in the project scope section of the report. Overall the removal of the cylindrical design was the main change and this led to the other changes.

Specification	**Details**
Diameter	Within 10 inches / 25 cm of diameter
Length	Within 1 meter of length
Electronics	- CO_2 Sensor - Pressure/Depth sensor - Must be able to run from a 3.3V or 5V source. - Must be programmable with an Arduino for use.
Materials of Enclosure	Materials for the frame and any structure housing components must be able to withstand 100 kPA (Sunk 1000m depth)
Temperature Operation	Work in near-freezing conditions (0℃)
Size Restriction	Must be able to fit in a pelican case to be shipped ex dimensions (12" x 9" x 5") ← Subject to change

Budget Restriction	$2000 budget provided by the sponsor
Programming Language	- C++ to program the Arduino - C# for the unity simulation
Weight Restriction	The prototype should be less than 20kg of weight. Allows operation with a 20kg weight used as an anchor/ counterweight

6.4 Future Improvements

Methods that could be involved in future work to improve the enclosure include:

- Better waterproofing (Better epoxy and/or seal)
- Better modularity (Using PCBs) to shrink the enclosure
- Stronger balloon materials
- Stronger enclosure material (steel)
- Adding mineral oil inside the enclosure to prevent compression at high pressure

These would improve the ability of the device to handle strong pressure as the current MVP 3D-printed enclosure needs to be improved before it can be tested in the deep ocean.

7. Measuring Success and Results of Validation Tests

7.1 Final Measures of Success

The measures of success changed throughout the project. This is mostly due to the change of scope from a device to be tested in the ocean to a device tested in a pool. The measures of success were coordinated with the sponsor as they changed.

The final measures of success related to the prototype were:

- Record CO_2 concentration in the top and bottom balloons over time.
- Record the time the valve closes.
- Record pressure data over time.
- Data is recorded to an SD card in CSV format.
- Data can be displayed graphically.
- Close the valve of the device between 3-4 m underwater. This happens when the pressure sensor detects the pressure threshold corresponding to 3-4 m.
- A user manual explaining the use of the prototype.

The technical specifications for the prototype changed as the scope changed to building circuitry that could be tested in the pool instead of the ocean. The original plan was to fit the prototype in a pelican case to send to the coast. This was abandoned as the testing only occurred in the pool. The user manual was adjusted to account for the new scope. While it still details the potential applications and how an ocean test would work, it makes note of the limited capability of the current design.

The final measures of success related to the simulation were:

- Prototype design section with an exploded view of the prototype and information about its components.
- Prototype use case section displaying how the prototype would be used in ocean testing.
- The system use case section detailing how the CO_2 separation system can be incorporated into the energy storage system.

7.2 Validation Tests

Simulation Testing

The simulation focused solely on functional requirements, meaning success was not determined by specific quantitative targets. Instead, the requirements centered on aspects such as user experience within the simulation. While the inclusion of the three sections—prototype design, prototype use case, and system use case—was necessary, the main aspect of the simulation was to provide context to users learning about the project concepts. The success of this component was based on the sponsor. He was happy with the results of the simulation and felt it displayed the project well.

Circuitry Testing

The circuitry's measures of success were very different in contrast and it was the result of the measurability of sensor components. Many components were tested individually and within the overall system. We found that the CO_2 sensors did not have as much of an issue during the preliminary testing with ~500 ppm being recorded in open air. Observable error in the depth sensor was seen to be around ±15 cm in shallow water(~1m). It was observed to be the same as the factory-stated error in deeper water(>3 cm) at about 1-2.7 m. Data was recorded to the SD card in CSV format during each trial run. This data could be retrieved even if the device lost power.

Pool Testing

For the prototype, the final validation test was conducted in the pool. Three trials recorded data (fig 7.2.1-7.2.3) and provided the values for pressure, depth, CO_2 concentration, and valve closure with respect to time.

The pressure and depth data showed excellent results. Depth was calculated with respect to the water density and the pressure. It was observed that as the device descended the pressure proportionally increased with depth. The threshold pressure value to trigger value closure was 1100 mBar or 2.1 m. The results in Figures 7.2.1 and 7.2.3 show that this pressure was reached at about 210 seconds for each and the valve was closed as planned. Figure 7.2.2 shows that the device was not brought deep enough to reach the 1100 m bar threshold pressure and therefore the valve didn't close. The final trial shows the box repeatedly descending and ascending over about 150 seconds. The data record shows the number of attempts to reach the threshold depth before finally reaching it on the last attempt. These results show that the system design works, pressure can be detected accurately and it can be used to trigger the closure of the valve separating the upper and lower balloons.

The data recorded in these tests showed that CO_2 levels in the balloons were between 600 and 1200 ppm with minor fluctuations in each trial. While the CO_2 concentrations obtained in the pool test don't show any information about separating CO_2 they do show that the device can record the concentration. In the results (fig 7.2.1) it is seen that the CO_2 concentration changes right when the valve closes. This is likely due to the valve causing airflow the moment it closes. This indicates that the sensors are reading the air inside the balloon and valve section. The change in CO_2 concentration over each trial.

Most of the measures of success were met. CO_2 could be recorded in the top and bottom balloons. The time the valve closed and pressure data were recorded. All the data was recorded on an SD card and was used to graph the results.

Despite the initial plan of closing the valve at 3-4 m, the final test only reached a max depth of 2.7 m and closed the valve at 2.1. This was because of the safety concern of a person going deeper and time constraints to set up a better testing method. It was decided that there was no need to dive deeper while holding the device as the results showed the main measures of success. We believe from the results and from initial testing in shallow water that the pressure threshold would work at 3 or more meters.

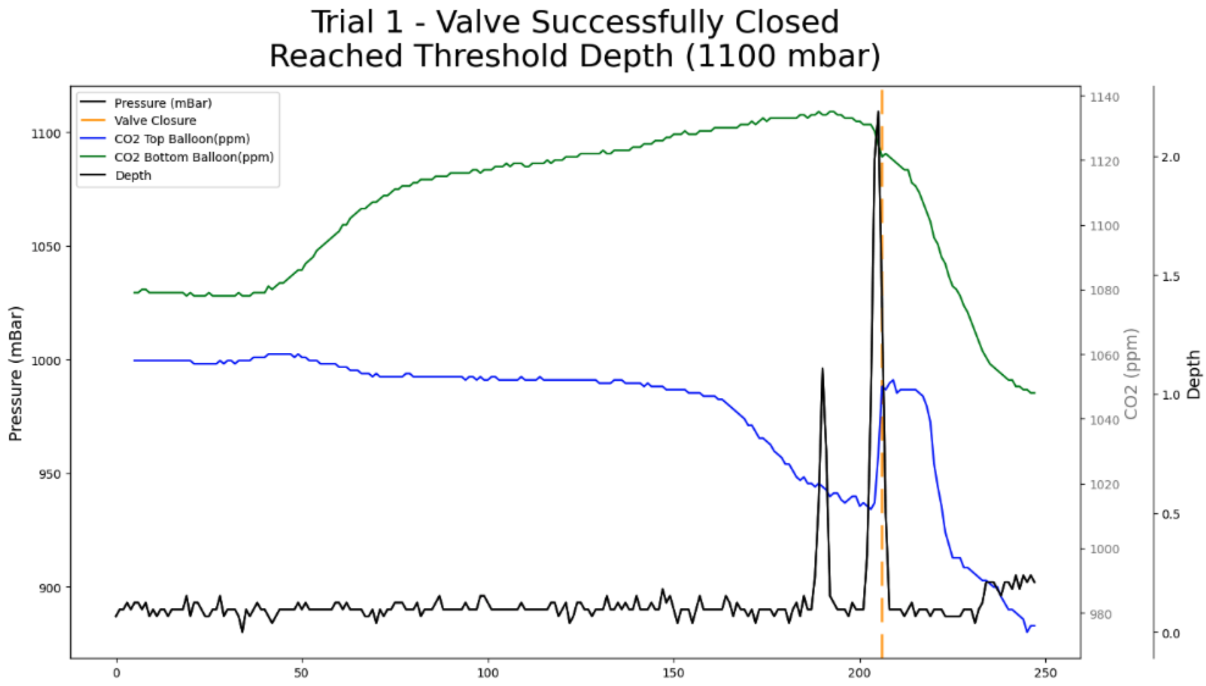

Figure 7.2.1 A graph showing a trial with a successful valve trigger after reaching the threshold pressure.

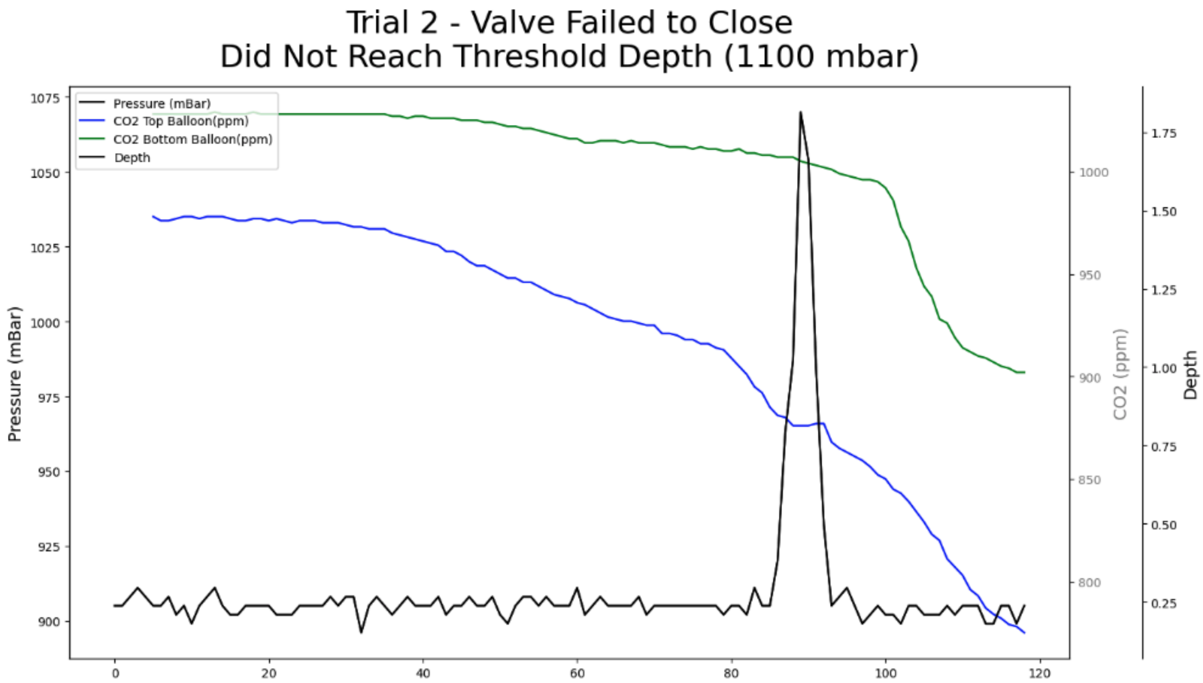

Figure 7.2.2 A graph showing a trial with a failed valve trigger. This is due to the device not reaching the threshold pressure or not getting deep enough.

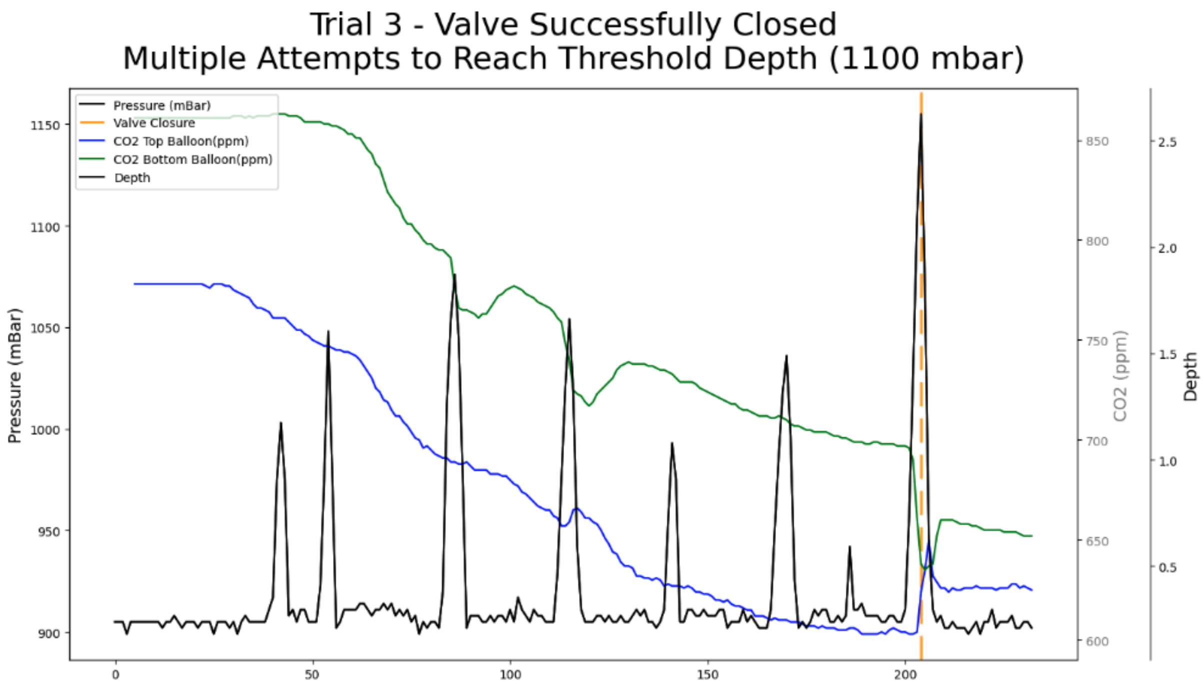

Figure 7.2.3 A graph showing a trial with multiple attempts at reaching the trigger depth until a successful valve trigger after reaching the pressure threshold.

8. List of Tools, Materials, Supplies, and Costs

This section details two comprehensive lists containing every tool, material, and component used to create the final product. The first list relates to the software component, while the following list highlights the physical (electrical) component.

8.1 Software Components

Software		
Blender [2.6]	Open-source (free)	Blender is a powerful 3D creation platform capable of rendering, modeling, and animating objects. It was used throughout the project, namely for modeling the prototype and animating components within the digital twin simulation.
Unity Game Engine [2.1]	Education License (free)	Unity Game Engine is a “real-time development platform” typically used to create games and industry simulations. It functions as the framework that supports the digital twin simulation.
Fusion 360 [2.2]	Education License (free)	AutoCAD Fusion 360 is a computer-aided design program, capable of creating 3D objects and testing them within real-world scenarios. It was used in conjunction with Blender to add specific measurements.
Draw.io	Free	Draw.io or diagrams.net is a website used to create diagrams, with a specific focus on UML diagrams. Specifically, it was used to create the UML activity diagram for the Arduino code.
Figma	Education License (free)	Charting application used the generate concept designs for prototype
Lucid Chart	Education License (free)	Lucid Chart is a collaborative charting web application similar to Draw.io
Programming Languages		
C#	Open-source (free)	C# is an object-oriented programming language created by Microsoft. It is used exclusively in the backend scripts of the digital twin simulation, controlling cameras, equations, and game objects.
C++	Open-source (free)	C++ is an object-oriented programming language developed as an extension to C. It is used exclusively to program the Arduino and control the electrical components connected to it.

8.2 Physical Components

Tools		
Makerspace [2.7]	Free	The Makerspace is a set of rooms with various machines used to help construct various projects. These machines include 3D printers, CNC mills, plasma cutters, and more. Specifically, the 3D printers were used to construct the box, lid, funnels, valve bracket, and battery case.
Screwdriver	From home	A typical screwdriver was used to fasten screws into the box.
Soldering Iron	From home	A typical soldering iron was used to solder wires to the electrical components.
Materials		
ABS Plastic	Free through U of C	The filament used within the Makerspace 3D printers. Used to construct the box, lid, funnels, valve bracket, and battery case.
Flex Seal [4.5]	$37.63 CAD	A waterproof rubberized coating applied with an aerosol can. Used in conjunction with the waterproof epoxy to seal in any openings within the box.
Waterproof Epoxy	$15.61 CAD	Epoxy is a thermoset polymer with sealing and adhesive properties. Due to the waterproof nature of epoxy, it was used in conjunction with the Flex Seal to fill in any openings within the box.
Teflon Tape	$7.34 CAD	Also known as plumber’s tape, it is typically used to seal pipe threads. Being naturally waterproof, it was used to seal connections between any components exposed to water.
Screws	$20.99 CAD	A fastener used to connect components within the box. Sizes range from M3 to M5, each with various lengths.
Zip Ties	$4.88 CAD	Used to connect the components without screw holes to the box and to help manage the cables.
Electrical Components		
Depth Sensor [4.3]	$445.92 CAD + $37.89 CAD Import Fee	An electronic sensor capable of measuring pressure and depth underwater.
CO_2 Sensors [4.2]	$272.79 CAD + $30.18 CAD Import Fee	An infrared sensor capable of measuring the concentration of CO_2 up to 50,000 parts per million (ppm)
Arduino Uno Rev3 [4.1]	$39.11 CAD	A microcontroller board that features 14 digital input/output pins. It is used as the central controller of all electrical components, including the sensors, the solenoid valve, and

		the MicroSD card.
Solenoid Valve [4.4]	$23.09 CAD	The solenoid valve controls the flow of fluid between the top and bottom of the device. It is normally closed to ensure efficient power draw and proper sealing of liquid CO_2 in the bottom balloon.
Shrink Tubing [4.15]	$19.94 CAD	The shrink tubing is used to connect the funnels to the hose fitting. Its flexibility allowed for easier installation and sealing.
Rechargeable Battery [4.13]	$50.39 CAD	The rechargeable battery is used to power the Arduino and its associated components. By being rechargeable, it allows for multiple tests to be run without wastefully replacing the power source.
Hose Clamp [4.14]	$10.49 CAD	The hose clamps are used to ensure a tight seal between the balloons and the funnels. Typically used to seal automotive hoses and industrial pipes.
Air Valves [4.12]	$19.24 CAD	The air valves are used to fill and extract air from their corresponding balloons. Typically used in a variety of wheels and tires, particularly for bicycles.
Mini Breadboards [4.7]	$13.95 CAD	The mini breadboards were used to subsidize further electrical connections that the main breadboard missed. Often used to design electrical circuits.
Test Breadboards [4.7]	$20.99 CAD	The test breadboards were used to design and test electrical connections between components before soldering.
Main Breadboard [4.7]	$13.63 CAD	The main breadboard was used to hold the majority of the electrical component connections. Often used to design electrical circuits.
Hose Fitting [4.11]	$13.62 CAD	The hose fitting is used to connect the shrink tubing to the central solenoid valve. Functions as an adapter with one threaded end and one ribbed end.
Latches [4.10]	$25.14 CAD	Found on the side of the box, the latches are used to seal the lid to the box. They are screw-adjustable to allow for a proper waterproof seal.
Latex Balloons [4.16]	$194.00 CAD + $22.65 CAD Import Fee	Found on the top and bottom of the device, the latex balloons hold the air and collect CO_2 when in use. The material allows the gas to compress without compromising the waterproof seal against hydrostatic pressure.
Arduino SD Card Attachment [4.6]	$23.99 CAD	The Arduino SD card attachment allows the Arduino to save measurement data to an external MicroSD card. Data was formatted as a .csv file.
Arduino Relay [4.8]	$10.38 CAD	The relay functions as a switch to control the power provided to the solenoid valve, determining whether it opens or closes.

Lid Gasket [4.9]	$12.83 CAD	Found attached to the opening of the box, the lid gasket is a rubberized ring that helps maintain the waterproof seal between the box and the lid.
Alligator Clips	Free Access	Alligator clips are commonly used as an alternative method of creating an electrical connection without soldering.
MicroSD Cards	$21.99 CAD	Used within both the GoPro and the Arduino SD card attachment, MicroSD cards are used to store data. More specifically, a large 128 GB card was used for the GoPro, whereas an 8 GB card was used for sensor data collection.

The total cost is $1307, which is $693 under the budget restriction as set by the sponsor. Most of the expenses came from the sensors as they were specialized and could be used at high depths.

9. Conclusion

While we deviated from the original goals, this project was able to successfully create a physical prototype that was tested in a pool and a simulation offering further explainability to the overall system.

This project offers numerous benefits for future endeavors. Further testing and refinement will not only validate the capabilities of the circuitry in deeper waters but also pave the way for real-world applications. Enhancements to the enclosure design, including improved mechanical principles and waterproofing, will ensure the device's reliability and durability under the extreme pressures encountered at depths of 500-1000 meters. Moreover, conducting tests in oceanic conditions at 500-meter depths represents the final crucial step toward demonstrating the practical feasibility of CO_2 separation. While this project is a stepping stone to better things, its practical application could help to reduce carbon emissions and be integrated into offshore energy storage systems in the future.

Final Products

The simulation [3.1] can be accessed at: https://dev3933.d18ixp0pwlw4ex.amplifyapp.com

A video [3.2] detailing the design process for the physical prototype type can be found at: https://www.youtube.com/watch?v=TvnBOPB7dhc&t

10. Appendix

10.1 Intellectual Property

This project represents a collaborative effort between Dr. Roman Shor and the team, resulting in shared intellectual property (IP). The IP encompasses all innovations, discoveries, designs, software, documentation, and other intellectual outputs generated as part of the project's execution.

10.2 Contributions

<table>
<tr><th>Team Member</th><th>Contributions</th></tr>
<tr><td>Chace Nielson</td><td><u>Physical Model Design</u>
- Created the design of the device. Sketches in Figma.
- Created the design for the internal mechanism in the blender.
- Designed the 3D components for the model. This involved aligning electrical components and finding a way to attach them to the inside of the device while maintaining a modular design.
- Found and ordered parts for the model.
- Created 3D print tests for screw holes and components to ensure a physical version would work. This led to adjustments to the design to make sure the final version would work.
- 3D printed and assembled the physical model. Created a waterproof seal for the complete box before adding electronics.
- Performed Waterproof tests.

<u>Unity Simulations</u>
- Created the 3D model design in Blender for the Unity simulation. This involved adding realistic components, animations, and materials to show the inner workings of the device.
- Set up and completed the Unity simulation scenarios.
    - Exploded animation to explore the design of the model.
    - Use case for the device.
    - The original system animation combined with a CO_2 separation device.
- Simulation UI and simulation scripts in C#.
- Blender models and animations for all components used in the simulation.
- Created a deployed version of the application on AWS.

<u>Project Management</u>
- Poster for the capstone design fair
- In charge of stakeholder meetings.
- Booked time at U of C Aquatic Center for pool testing.
- Performed the pool test while being in the water.
- Created a video of the design process and pool tests.
- Generated result graphs from the pool test</td></tr>
<tr><td>Fahad Yasin</td><td><u>Electrical Circuit</u>
- Designed the circuitry
- Soldering</td></tr>
</table>

	- Circuit testing - Waterproof tests - Circuitry Code <u>Physical Model Design</u> - Assembly of the circuit in the enclosure - Wiring
Claes Medrano	<u>Electrical Circuit</u> - Designed the circuitry - Soldering - Circuit testing - Waterproof tests - Circuitry Code <u>Physical Model Design</u> - Assembly of the circuit in the enclosure - Wiring
Cale Morash	<u>Physical Model Design</u> - Compiled a list of ideal materials for original prototypes - Found parts for the model - Calculated expected CO_2 for various balloon sizes - Modeled various screws and threaded openings in Fusion 360 - Wrote the user manual - Created UML activity diagram for Arduino code <u>Unity Simulations</u> - Wrote descriptions for components of the simulation - Translated CO_2 calculations into simulation with C#
Nabeel Amjad	<u>Electrical Circuit</u> - Assisted in the design of the circuitry - Assisted in the soldering - Waterproof tests <u>Project Management</u> - Sent weekly updates to stakeholders - In charge of stakeholder meetings - Booked Shouldice Aquatic Centre as an original test site - Delegated tasks as required to the group - First draft of design fair poster - Completed capstone fair deliverables

11. References

11.1 Existing Work and Reference Material

[1.1] G. Hands, K. Truong, Y. Unico, A. Ashar, A. Al-Saiedy, and R. J. Shor, "Energy Storage and Direct Air Carbon Capture Solution for Offshore Sources of Energy," presented at the IEEE Conference, June 11-16, 2023, Melbourne, Australia.

[1.2] “Study abroad with sea,” Sea Education Association, https://sea.edu/. 2023.

[1.3] B. Finney and M. Jacobs, *Carbon dioxide pressure-temperature phase diagram.* 2010. Available: https://en.m.wikipedia.org/wiki/File:Carbon_dioxide_pressure-temperature_phase_diagram.svg.

[1.4] Casco Antiguo Pro, 'MULTI WATER SAMPLER SLIMLINE MWS6,' Sampling Gear & Other Gear. Available: https://www.cascoantiguopro.com/en/oceanography/sampling-gear-other-gear/muestreador-multiple-de-agua-slimline-mws6.html.

[1.8] "Climate Change: Atmospheric Carbon Dioxide." Climate.gov. [Online]. Available: https://www.climate.gov/news-features/understanding-climate/climate-change-atmospheric-carbon-dioxide.

[1.9] "Model-View-Controller (MVC)." TechTarget. [Online]. Available: https://www.techtarget.com/whatis/definition/model-view-controller-MVC#:~:text=In%20programming%2C%20model%2Dview%2D,a%20specific%20set%20of%20tasks.

[1.10] “Station Pressure - Hourly data for Calgary.” Environment and Climate Change Canada.[Online]. Available:https://calgary.weatherstats.ca/charts/pressure_station-hourly.html

11.2 Tools

[2.1] Unity Technologies. (2022). Unity - Game Development Platform. [Software]. Available: https://unity.com/.

[2.2] Autodesk. (2022). AutoCAD. [Software]. Available: https://www.autodesk.com/products/autocad/overview.

[2.3] Blender, Home of the Blender project - free and open 3D creation software, blender.org. Available at: https://www.blender.org/.

[2.4] Amazon Web Services. Amplify. Available at: https://aws.amazon.com/amplify/.

[2.5] GoPro, "Cameras - GoPro," Available: https://gopro.com/en/ca/shop/cameras.

[2.6] OpenAI. "ChatGPT: Optimizing Language Models for Dialogue." OpenAI. Available: https://openai.com/chatgpt.

[2.7] University of Calgary Libraries & Cultural Resources. "3D Printing at Makerspace." University of Calgary. Available: https://library.ucalgary.ca/services/makerspace.

11.3 Final Products

[3.1] Nielson, C. Deployed Capstone simulation. Retrieved from https://dev3933.d18ixp0pwlw4ex.amplifyapp.com/.

[3.2] C. Nielson, "Capstone Prototype Design Process," YouTube. [Online]. Available: https://www.youtube.com/watch?v=TvnBOPB7dhc&t.

[3.3] Nielson, C. Capstone-Simulation [GitHub repository]. Available at: https://github.com/ChaceN89/Capstone-Simulation.

[3.4] Yasin, F. Capstone-Simulation [GitHub repository]. Available at: https://github.com/Fahady22/CO2-Seperator-Capstone-Final-Code.git

11.4 Parts for Prototype

[4.1] Arduino, "Arduino Uno Rev3," Arduino Store, [Online]. Available: https://store-usa.arduino.cc/products/arduino-uno-rev3?selectedStore=us.

[4.2] DFRobot, "Gravity: UART Infrared Carbon Dioxide Sensor (0-50000 ppm)," DFRobot, [Online]. Available: https://www.dfrobot.com/product-1565.html.

[4.3] "Bar100 High-Resolution 1000m Depth/Pressure Sensor," Marine Thinking Store, [Online]. Available: https://store.marinethinking.com/products/bar30-high-resolution-300m-depth-pressure-sensor

[4.4] HFS(R), "HFS(R) 12v Dc Electric Solenoid Valve Normally Closed for Water, Air, Gas - 1/4" NPT Available (12V DC 1/4" NPT)," Amazon, [Online]. Available: https://www.amazon.ca/HFS-Electric-Solenoid-Valve-Water/dp/B018WRJYPY/ref=zg_bs_g_12572427011_sccl_2/146-8962710-4606124?th=1.

[4.5] Flex Seal Products, "FLEX SEAL®," Flex Seal Products, [Online]. Available: https://flexsealproducts.com/#6.

[4.6] Generic, "2pcs Micro SD Card Module Storage Board 6-pin TF Card Memory Adapter Reader Module SPI Interface fit Arduino," Amazon, [Online]. Available: https://www.amazon.ca/gp/product/B07PFDFPPC/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1.

[4.7] Gikfun, "Home," Gikfun Store, [Online]. Available: https://gikfun.com/.

[4.8] Walfront, "Relay Module 2 Channel 12V Relay Module Isolated Optocoupler High and Low-Level Trigger Module 1PC," Amazon, [Online]. Available: https://www.amazon.ca/dp/B07Y8JY6G8?psc=1&ref=ppx_yo2ov_dt_b_product_details.

[4.9] Gangdise, "Gangdise Weather Stripping Seal Strip for Doors/Windows 9.8 Feet, Self-Adhesive Seal Strip Soundproofing Weatherstrip White D3White," Amazon, [Online]. Available: https://www.amazon.ca/dp/B09NBYDL1Q?psc=1&ref=ppx_yo2ov_dt_b_product_details.

[4.10] SYINE, "Lockable Adjustable Door Latch Buckle Cabinet Boxes Lever Handle Chest Latch Toggle Catch Latch Lock,2 Pcs (1S,Range:75-95mm)," Amazon, [Online]. Available: https://www.amazon.ca/dp/B07WRFTK8T?psc=1&ref=ppx_yo2ov_dt_b_product_details.

[4.11] Anderson Metals, "Anderson Metals 57001 Brass Hose Fitting, Adapter, 1/4" Barb x 1/4" NPT Male Pipe," Amazon, [Online]. Available: https://www.amazon.ca/dp/B002SAO7XQ?psc=1&ref=ppx_yo2ov_dt_b_product_details.

[4.12] "4pcs Car Tire Valve Stem Cap, TR414 Rubber Tubeless Tire Valve Core Stems Cap," Amazon, [Online]. Available: https://www.amazon.ca/dp/B07LG71DWT?psc=1&ref=ppx_yo2ov_dt_b_product_details.

[4.13] "Talentcell Rechargeable 12V 3000mAh Lithium-ion Battery Pack for LED Strip, CCTV Camera and More, DC 12V/5V USB Dual Output External Battery Power Bank with Charger, Black," Amazon, [Online]. Available: https://www.amazon.ca/dp/B01M7Z9Z1N?psc=1&ref=ppx_yo2ov_dt_b_product_details.

[4.14] NEBURORA, "NEBURORA 4 Inch Hose Clamp Adjustable 304 Stainless Steel Duct Clamp 4 Pack Worm Gear Clamps 3.6"-4.4" Diameter Fuel Line Hose Pipe Clamp for Dryer Vent Air Duct Automotive Plumbing," Amazon, [Online]. Available: https://www.amazon.ca/dp/B0BZSF25YG?psc=1&ref=ppx_yo2ov_dt_b_product_details.

[4.15] Wirefy, "Wirefy Heat Shrink Tubing Kit - Larger Diameter - 3:1 Dual Wall Tube - Adhesive Lined - Marine Shrink Tubing - Black, Red - 200 PCS," Amazon, [Online]. Available: https://www.amazon.ca/dp/B089D82FLG?psc=1&ref=ppx_yo2ov_dt_b_product_details.

[4.16] "Latex Stallion Condom," Revival Animal Health, [Online]. Available: https://www.revivalanimal.com/product/latex-stallion-condom.