GNG 1103
Design Project User and Product Manual

In-Line Device Beer System

Submitted by:

All_They_Do_Is_Win - GROUP 3

Thomas Johnson Band, 300294705

Gabrielle Graceffa, 300310376

Emmett van den Broek, 300300285

Ashley Garofalo, 300280642

Alec Plourde, 300108326

December 7th, 2022

University of Ottawa

Table of Contents

List of Figures

List of Tables

List of Acronyms and Glossary

Table 1. Acronyms

Acronym	Definition
BOM	Bill of Materials
ILDBS	In-Line Device Beer System
LCD	Liquid Crystal Display
UPM	User Product Manual
CAD	Computer aided design
SG	Specific gravity

Table 2. Glossary

Term	Acronym	Definition
Orifice Plate	N/A	It creates a difference in pressure, in doing so we can: measures flow rate, reduces pressure, and/or restricts flow.
Specific Gravity	SG	Relative density, ratio of the density of a substance to that of a standard substance [1].
Volumetric flow	N/A	The amount of volume of liquid going through the a pipe per second
Brew master	N/A	The main person who is in charge of supervising the brewing process [2].

1.0 - Introduction

This User and Product Manual (UPM) provides the information necessary for small company brewers to effectively use the In-Line Device Beer System (ILDBS) and for prototype documentation. The intended audience are people who have a chemical engineer or a beer brewing background. The basic concept of the In-Line Device Beer system is to measure specific gravity in degrees plato accurately every 10 seconds self-sufficiently. If the measured specific gravity is in the acceptable range for beer brewing (0-30 degrees plato) there is a light function that blinks green. If the specific range is in the inacceptable range the light function blinks red. There are four main parts to the assembly. The first part of the assembly is a threaded pipe that is half an inch in diameter that connects to a volumetric flow meter and a pressure sensor. The second part of the assembly contains the orifice plate that will constrict the flow of the beer resulting in the increase of flow and the decrease in pressure. The orifice plate is 0.25 inches in diameter. The third part of the assembly contains a threaded pipe that is a 0.8 in diameter and the pressure sensor device is twice the distance of the first pressure sensor to the orifice plate. The fourth part of the assembly connects all the sensors together to the arduino mega and a breadboard. The LCD screen is attached anywhere near the assembly with the red and green lights attached on the opposite side of the LCD screen. The assumption is that the flow of the beer mainly relies on gravity.

2.0 - Overview

The client Shane Clark is the Co-CEO of Beyond the Pale Brewing Company. The company is currently in their expansion phase and requires assistance in updating their measuring devices. The brewing personnel take a significant amount of time to measure specific gravity and a certain percentage of their product gets discarded. Nonetheless, the requirement of measuring specific gravity is crucial in the beer brewing process to ensure that the quality of beer remains constant.

The client's fundamental need is the numerical accuracy of their in-line measuring device, preferably in degrees Plato. Through vigorous research in technical and user benchmarking there is no device on the market that measures specific gravity in an in-line system for beer in a small business. The device is self-sufficient and allows the brewers to spend less time to judge if the measured specific gravity is in the acceptable range.

The device measures the specific gravity approximately every 3 seconds while then converting the results to Plato. The results are then displayed on a LCD screen for the required workers to analyze in the brewery. Everytime the specific gravity displays the numerical value it also blinks either a green or red light. If the device blinks green then the specific gravity measurement is in the acceptable range of specific gravity for beer (0-30 degrees plato) and red if not.

The theory behind the device is based on a manipulation of the Bernoulli and continuity equations. In order to measure the specific gravity, the device deliberately temporarily converges the flow towards an orifice plate while allowing it to diverge after crossing the reduced diameter. This convergence of flow allows for a decrease in pressure and increase in flow speed. This is accomplished using an orifice plate which will cause the disruption. The final derived equation can be written as: $q = C_d * (\frac{\pi}{4}) D2^2 * [\frac{2\Delta P}{\rho(1-d^4)}]^{1/2}$, where d= D2/D1, D2= orifice plate diameter, Cd = coefficient of discharge, ρ = density, q = volumetric flow, and ΔP= difference in pressure [3]. Using this equation, we can isolate for density as Cd, d and D2 are known, while ΔP and q will be found using appropriate sensors. This will allow us to use these following equations:

$SG = \frac{\rho_{liquid}}{\rho_{reference}}, where\ \rho_{reference} = water,\ which\ has\ a\ typical\ value\ of\ 997\ kg/m^3$ and Plato = (-1 * 616.868) + (1111.14 * SG) – (630.272 * SG^2) + (135.997 * SG^3).

In order to find the q value, a volumetric flow meter sensor will be used to measure the amount of volume going through the in-line device. Likewise, to measure the ΔP, a differential pressure sensor will measure the difference in pressure at 2 points: 1 before the orifice plate, and another right after the orifice plate. This will allow to measure the difference in the pressure when the orifice plate drops the pressure at point 2. The final CAD model can be seen in figure 1.1 and the final built prototype can be seen in figure 1.2. To use the device, the user needs to connect it properly and safely into the established in-line system and look at the LCD screen to know the specific gravity of the beer.

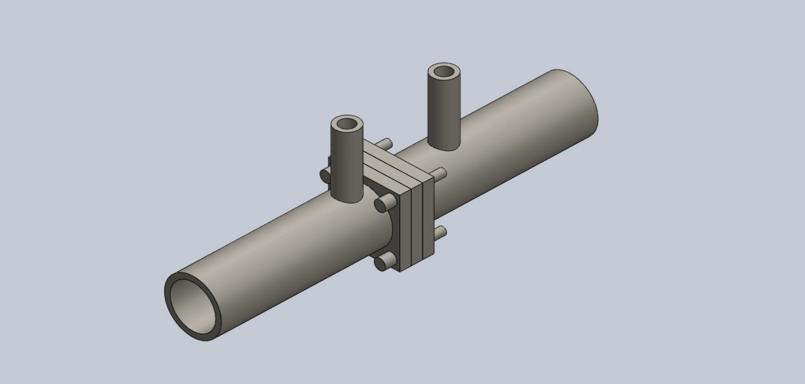


Figure 1.1: Final CAD prototype of the In-line system device

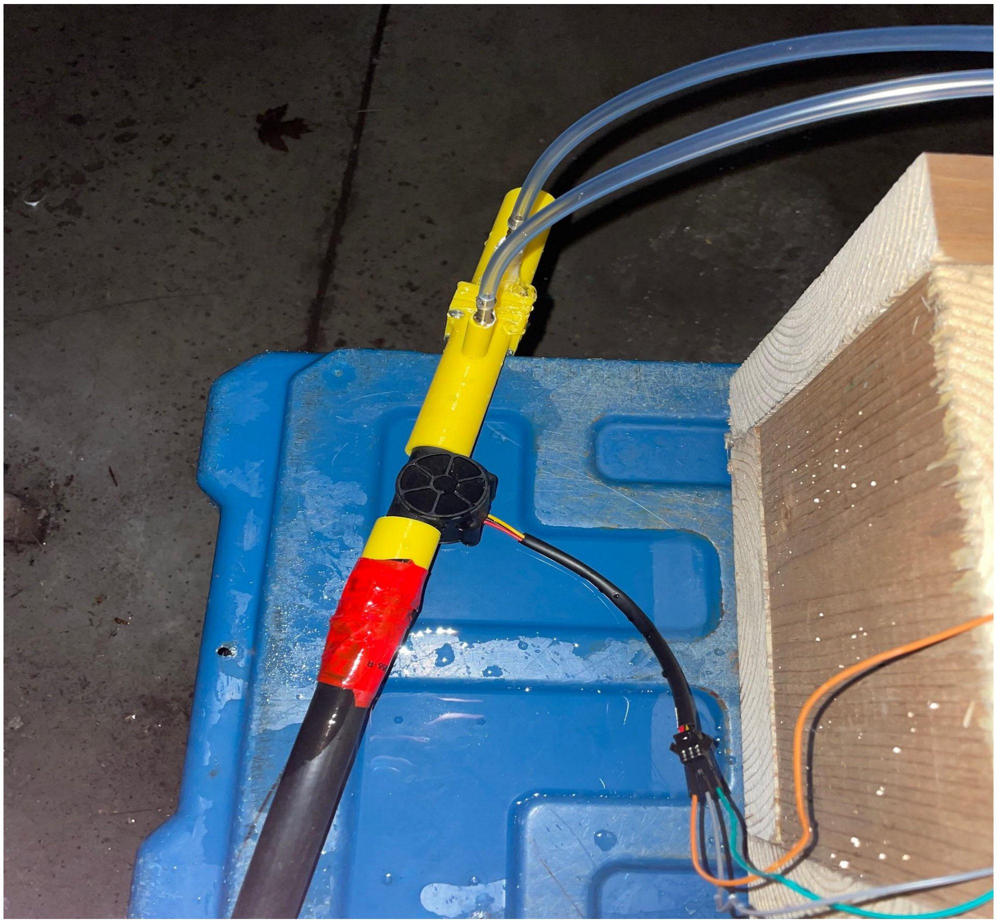


Figure 1.2 - Final physical prototype during testing phase

2.1 - Conventions

This device has no stylistic or command conventions.

2.2 - Cautions & Warnings

Ensure that the device seals properly into the already implemented system. If the device is not sealed properly, the in-line system will leak and cause uncalculated disasters.

3.0 - Getting started

3.1 - Configuration Considerations

The design for the in-line system involves various sections working together in unison to provide accurate and reliable specific gravity values. The equipment used can be broken down into an arduino mega (or arduino uno will do) to act as the main computer of the design. The breadboard used is to provide power and simplify data transactions to and from the arduino to all sensors and devices. A few other components used is a red and green led light and a few resistors to manage the voltage. the results will be displayed on an LCD screen for easy readability.

The sensors used are one volumetric flow sensor wich finds the amount of volume of liquid throug a pipe per second, and a differential pressure sensor which measures the difference in pressure at two points. These can be bought at most electronic stores. These sensors will act as output devices to send the results to the arduino in order to perform the correct calculations of specific gravity and plato.

3.2 - User Access Considerations

Users of our device include head brewers, assistant brewers and lab technicians that work at breweries. All of these potential users have the intention of brewing beer and knowing the specific gravity of it. All users will have the ability to read the value on the LCD along with the appropriate colour based on the range of value.

3.3 - Accessing/setting-up the System

Setting up the design is a simple and time effective process. The first step is to insert the in-line system into the existing piping system using a tri-clamp on each side to ensure it is securely connected. Once the device is connected in the in-line system, insert the power cord of the arduino to the wall mounted outlet. Once the power is turned on, the arduino should run the set-code and display the results upon flow through the pipe.

If the code is not pre-programmed in the arduino, after the device is installed, insert the communication cable form the arduino to the laptop to upload the code. Once the code is uploaded, remove the cable and the system should run.

3.4 - Exiting the System

In order to turn off the system, a simple one step process is required. To turn off the power to the device, simply unplug the main power supply to the arduino. If cleaning is required, simply isolate the pipe to safely remove the device. To clean, you can run water and cleaning solution through the pipe while ensuring no yeast buildup has occurred.

4.0 - Using the System

Due to our design, the device does not require constant use once assembled due to its self-sufficient nature. The automated process will measure and display the results of specific gravity and plato while also displaying the correct range of values with a green/red light system. The only user reuiqrment is to read the value while also looking for the correct light colour.

4.1 - Specific Gravity reading

The main feature of this design is the display of specific gravity located on an LCD screen on the pipe. The user of this device will be a brewmaster who will be within 5 meters of the device at all times. This allows us to add an LCD screen which will display the specific gravity of the liquid, this LCD screen can be located on the pipe or anywhere more advantageous for the brewmaster. There is no required input as the specific gravity on the LCD screen will constantly update every couple seconds. The specific gravity will be shown in plato as asked by the client but could be changed if needed.

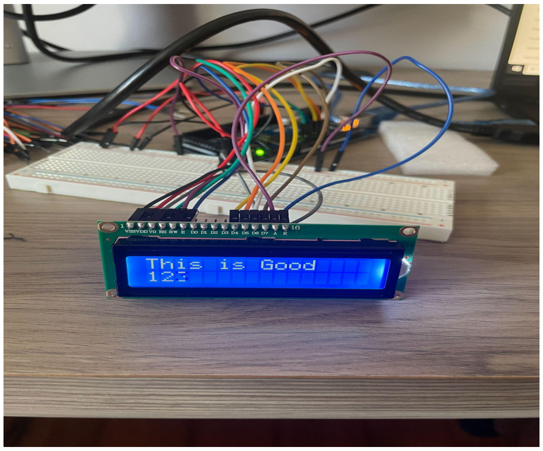


Figure 4.1 - LCD screen printing correct message for test #2

4.2 - Red/Green lights

A subfeature of this design is the red and green light located on the pipe. These two lights will be placed on the top of the pipe so that the brewmasters have easier accessibility to it. These two lights quickly tell the brewmasters if the specific gravity is in the correct range. Instead of displaying the exact specific gravity, the green light will turn on if the specific gravity is under thirty plato and red if it’s over 30 plato. This is the range of plato the client gave us so that if the brewmasters see the red light they can shut down the system to then fix the problem. There is no required input by the user as the red and green lights will automatically update every couple seconds.

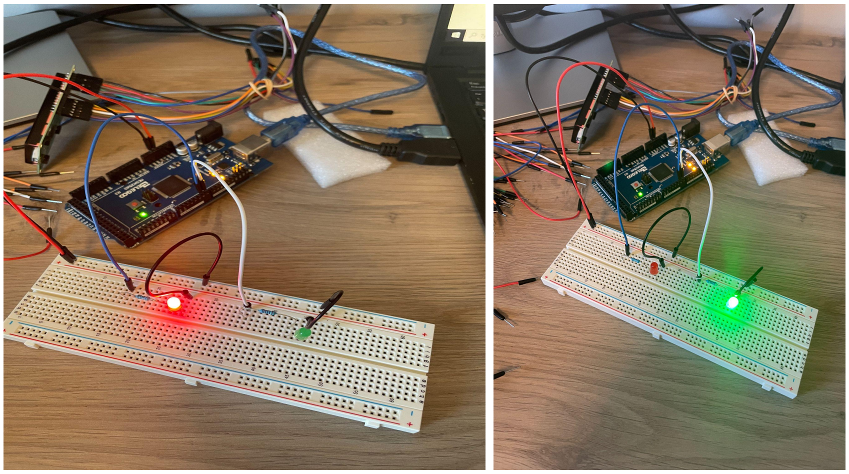


Figure 4.2/4.3 - Red/Green LED activated during trial test #4

5.0 - Troubleshooting & Support

5.1 - Error Messages or Behaviors

Electronic Components - It is important that all electronic components, (Arduino, wires, LCD screen and red/green lights), are kept dry at all times. If these parts come in contact with liquid, they are prone to breaking.

Pressure Sensors and Volumetric Flow Meter - Buildup of yeast on these two components may cause results to be inaccurate. This shouldn't be too big of an issue due to a consistent flow through these components but regular cleaning would only be beneficial.

Physical Build - It is important that the physical build of the prototype remains water tight. Clamps and sealants should be in working condition at all times. If any leaks in the system arise, it is important that they are resealed as quickly as possible.

5.2 - Special Considerations

Our prototype does not have any special circumstances or exceptions that need to be considered for troubleshooting.

5.3 - Maintenance

Prototype should be cleaned on a regular basis to avoid yeast buildup that could have an effect on the results. Clamps should be checked so it is ensured that the prototype is water tight. Box containing all electronic components should be kept dry at all times.

5.4 - Support

Emergency assistance and system support can be accessed by contacting Emmett van den Broek, through text or email at 905-269-4482 or evandenbroek2004@gmail.com. Any identified problems regarding the prototype should be reported as they are observed and instructions on how to fix the stated problem will be sent back.

6.0 - Product Documentation

In this section, you will learn the step by step process on how to build our device and its unique design points. Our designed had various tests to enable and ensure the proper functionality of all individual sensors and parts used. You will also see the exact bill of materials (our cost), all equipment used and every error that was made during design, building, and testing phases.

6.1 - Equipment Used and Bill of Materials (BOM)

6.1.1 - BOM (Bill of Materials)

# Parts	Device Name	$CAD	Explanation
1	Arduino Mega 2560 Rev3	0	Alec has one; an arduino mega is needed because there are a lot of sensors that need to be plugged in to it
1	Breadboard	0	Alec has one: in order to connect all required sensors and devices to the arduino
2	LED light (green and red)	0	Alec has one; to demonstrate the specific gravity and to blink the correct color that goes with the specific gravity
3	Resistors (2x10kOhm) and 1(220 Ohm)	0	Alec has them; they will be used to correctly wire the LED lights along with the volumetric flow meter
1	Orifice plate	0	The prototype is 0.5” ID in diameter; therefore, the orifice plate must be smaller. The plate is designed by onshape CAD and produced by a 3D printer on makers lab offered free for UOttawa students
1	Differential sensor plate	65.95	To read the difference in pressure between the 2 pressure points (one before and one after the orifice plate)
1	Tubes	0	To transfer the pressure readings from the taps to the differential pressure sensor device (5mm ID is required)
2	Pneumatic Hose Fitting	0	To connect the tubes into the piping device based on the 3/16” ID of the tube and the slot created for the fitting
1	Volumetric Flow Meter	34	To calculate the amount of liquid that is passing through the straight pipe (in mL/s) to give us the value “q” in our main equation to find the specific gravity
Total		**$99.95**	

6.1.2 - Other Equipment list

The main software used for this project was SolidWorks to design the CAD model of our theoretical design along with Cura to enable the printing process of our parts. The list of other materials can be seen below:

1. Epoxy Resin
2. 3D Pipe and Design Material (Plastic)
3. 3D Printer
4. Kevlar Tape
5. Wood
6. ⅛” screws
7. Hand Drill
8. Miter Saw/Jig Saw

6.2 - Instructions for Building Process

6.2.1 - Electrical

The items required for this section:

- Arduino Mega (arduino uno will also suffice)
- Breadboard
- (2x) 10Kohm and (1) 220 ohm resistors
- access to power
- red and green led lights
- Volumetric flow sensor
- Differential Pressure sensor

Step #1: Understand the theoretical wiring diagram for the physical design (see figure 6.2.1). Understanding the wiring process is critical in simplifying the physical wiring of the device. In our design, an I2C wasn’t available, thus it is also possible to wire the LCD screen using a traditional method (see figure 6.2.2 for wiring of the LCD screen in a traditional way).

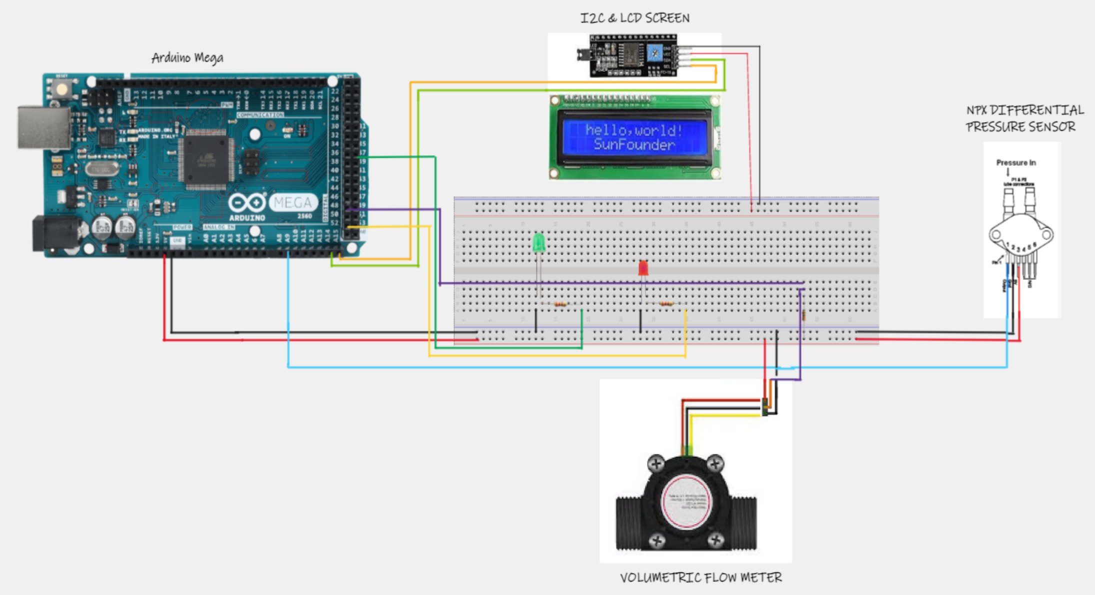


Figure 6.2.1 - Wiring Diagram for Project Design

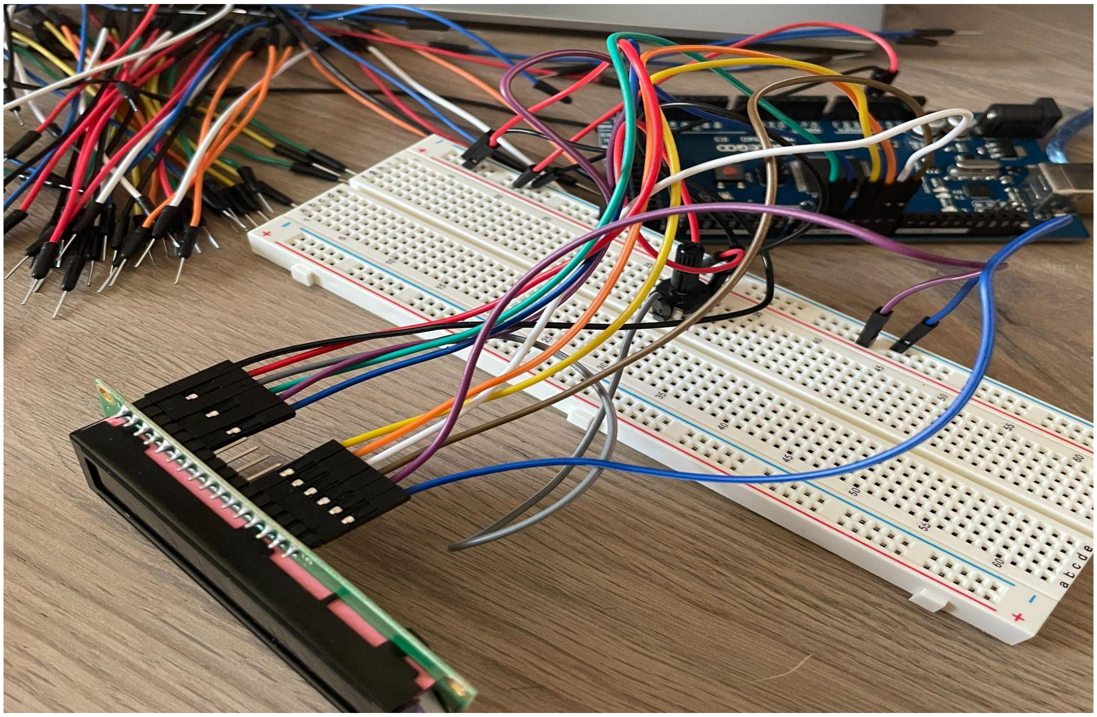


Figure 6.2.2 - Wiring of the LCD scree used in our project.

Step #2: Wire all components to the breadboard following the theroeetical wiring diagram. The best method found is to start by wiring each item/sensor one by one, emphasizing on the inputs (wich wire is sending/receiving information to the arduino) and the power/ground lines. Doing it one by one will limit mistakes and allow for better documentation (see Figure 6.2.3 for all components being wired together)

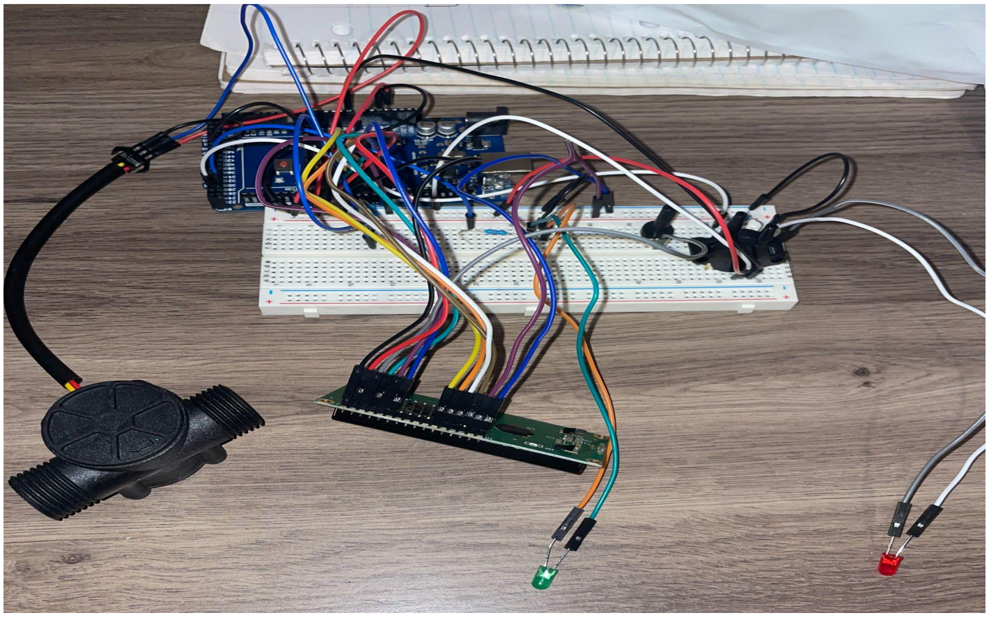


Figure 6.2.3 - Wiring of all components for the theoretical design

6.2.2 - Software

The items required for this section:

- Computer
- Connecting wire for arduino to computer
- SolidWorks
- Arduino coding software (Arduino IDE)

The software aspect of this project can be broken down into two aspects: Solidworks software for the 3D build of the design, and Arduino IDE for the coding of the arduino. If Solidworks is not an option, AutoCAD or OnShape are also feasible options to design the required parts. Solidworks was chosen due to its superior user interface and previous team experience using this design software.

For coding: The required final code for the design can be seen below. Various comments were added to each line of code, indicating what it is why it is there. This allows for a simpler method of digesting and understanding the code. Please analyze each line while trying to understand each code. Another useful step is to write the code for one device/sensor at a time which can be represented in the testing steps. Simply copy paste the code into Arduino IDE to verify and modify it if needed.

Final code used:

```
#include <LiquidCrystal.h> // include library for the LCD Screen

//Variables for the Differential Pressure Sensor
const float ADCtomV = 4.8828125; // Conversion multipler needed from Arduino ADC value at pin to voltage in mV
const float SensorOff = 200; // units in mV taken from data sheet for MX5050DP
const float sensitiv = 90; // units in mV/Kpa taken from datasheet
float diffpressure; // The value in Kpa of the differential pressure that we need
int PIN_P1 = A0; // If the diff pressure sensor is plugged into analog pin A0

//Variables for the Volumetric Flow Meter
int PinFlow = 2; // This is the flow meter input digital pin 2 on the arduino
double VolFlow; // this is the volumetric flow we want to find
volatile int sum; // intger is set as volatile for correct updates during interrupt = important as it will store #of pulses each second

//Variables for LED lights
int redledpin = 24; // define where red led pin is placed 24 digital pin on arduino
int greenledpin = 25; // define where green led pin is placed 25 digital pin on arduino

//Variables for the LCD screen
const int rs=1, en=8, D4 = 4, D5=5, D6=6, D7=7;  //rs = rs pin on LCD, en = enable pin on LCD, D4 = register pin D4, same for D5-D7
```

```
LiquidCrystal lcd(rs,en,D4,D5,D6,D7);   //Initialize the library by
associatting LCD interface pin with arduino pin number it is conected to

//Plato Calculations Variables
const float CD = 0.6; // coefficient of discharge for orifice plate finish
(for flat edge is typically 0.6)
const float Beta = 0.33333333333; // ratio between orifce diameter and
inside pipe diamter = 1/3 so 0.333333
const float Do = 0.00635; // orifice diameter in meters
const float pi = 3.141592654; // pi constant
const float rowWater = 1000; // constant for row of water
float row; // density of fluid constant
float SG; // density of specific gravity constant
float PlatoVal;  // define Plato Value, in float so it can display
Decimals

void setup() {

  lcd.begin(16,2); // setup LCD screen dimensions (16 x2)
  lcd.clear(); // clear LCD screen as a resest function

  pinMode(PinFlow, INPUT); // sets the Flow pin from arduino as an input,
pin needs to be set as an input before it can interrupt
  attachInterrupt(0, Flow, RISING); // makes the interrupt 0 (pin #2 on
arduino mega) to run the function "Flow" so the increment function

  pinMode(redledpin, OUTPUT); // setup red and LED as output on arduino
  pinMode(greenledpin, OUTPUT); // setup green LED pin as output on
arduino

  Serial.begin(9600); // start serial communication

}

void loop() {

  //Pressure in Kpa Calculation
  diffpressure = ((analogRead(PIN_P1) * ADCtomV - SensorOff) / sensitiv) *
1000; // result will be in Kpa since [mv * Kpa/mV] = Kpa so we then
multiply by 1000 to get Pa


  //Volumetric flow in m^3/s Calculations
  sum = 0; // resets the counter so we start counting from 0 again since
loops runs over and over again
  interrupts(); // allows interups on the arduino to run, so start
counting # of pules from sensor
  noInterrupts(); // Disable the interrupts on the arduino so stop
counting pulses
  // Now we must do the math to find the volumetric flow since we know
approx 2.25mL fluid/pulse (everytine it rotates)
  VolFlow = (sum*2.25); // #pulse in a second by the amount of 2.25mL of
fluid
  VolFlow = VolFlow / 1000; // convert to L/s
  VolFlow = VolFlow / 1000; //convert L/s to m^3/s

  //Math to find Plato Value
  row = (diffpressure *
(CD*CD)*(Do*Do*Do*Do)*(pi*pi))/(8*(VolFlow*VolFlow)*(1-(Beta*Beta*Beta*Bet
a))); // equation when soving for density
  SG = row/rowWater; //  finding SG value
  PlatoVal = (-1*616.868) + (1111.14* SG) - (630.272 * (SG*SG)) + (135.997
* SG * SG * SG);  // equation found converting SG to Plato

  //LCD code to print resulting Plato Value to Screen
  lcd.setCursor(0, 0); // set the cursor to column 0, line 0 which means
1st row, 1st column
  lcd.print("Plato: ");
  lcd.print(PlatoVal); // print specific gravity in Plato

  // Code for LED lights depending on PlatoVal
  if(PlatoVal >0 && PlatoVal <30) { // if the plato value is between 0 and
30 then show green light as it is good
```

```
   digitalWrite(redledpin, LOW); // digital write (LOW) = turn Red led off
   digitalWrite(greenledpin, HIGH); // digitalWire (HIGH) = turns green
LEd on and shows plato range is good

 }

  else if(PlatoVal <0 || PlatoVal > 30){ // if plato value is less then 0
or larger then 30 , show that plato value is bad with red light

    digitalWrite(redledpin, HIGH); // red light is turned on
    digitalWrite(greenledpin, LOW); // green light is turned off
  }

  delay(1000); // delay everything (Vol Flow and Pressure readings) for 1
second
}

void Flow() { // new function created that th einterrupt will run every
time a pulse is found

  sum++; // everytime this function is called an increment, ++ will ad 1
to that variable everytime the program runs

}
```

For the CAD design: Using Solidworks or Onshape, design the device based on the required dimensions. The engineering drawing of the parts together can be seen below along with a 3D view. Multiple parts must be designed in this stage. Part 1 is the device that attaches to the other side of the volumetric flow meter (see Figure 6.2.4). All units are in inches for this project. For any difficulties designing these parts, research and tutorials are recommended to further develop. The assembly drawing can be seen in Figure 6.2.5

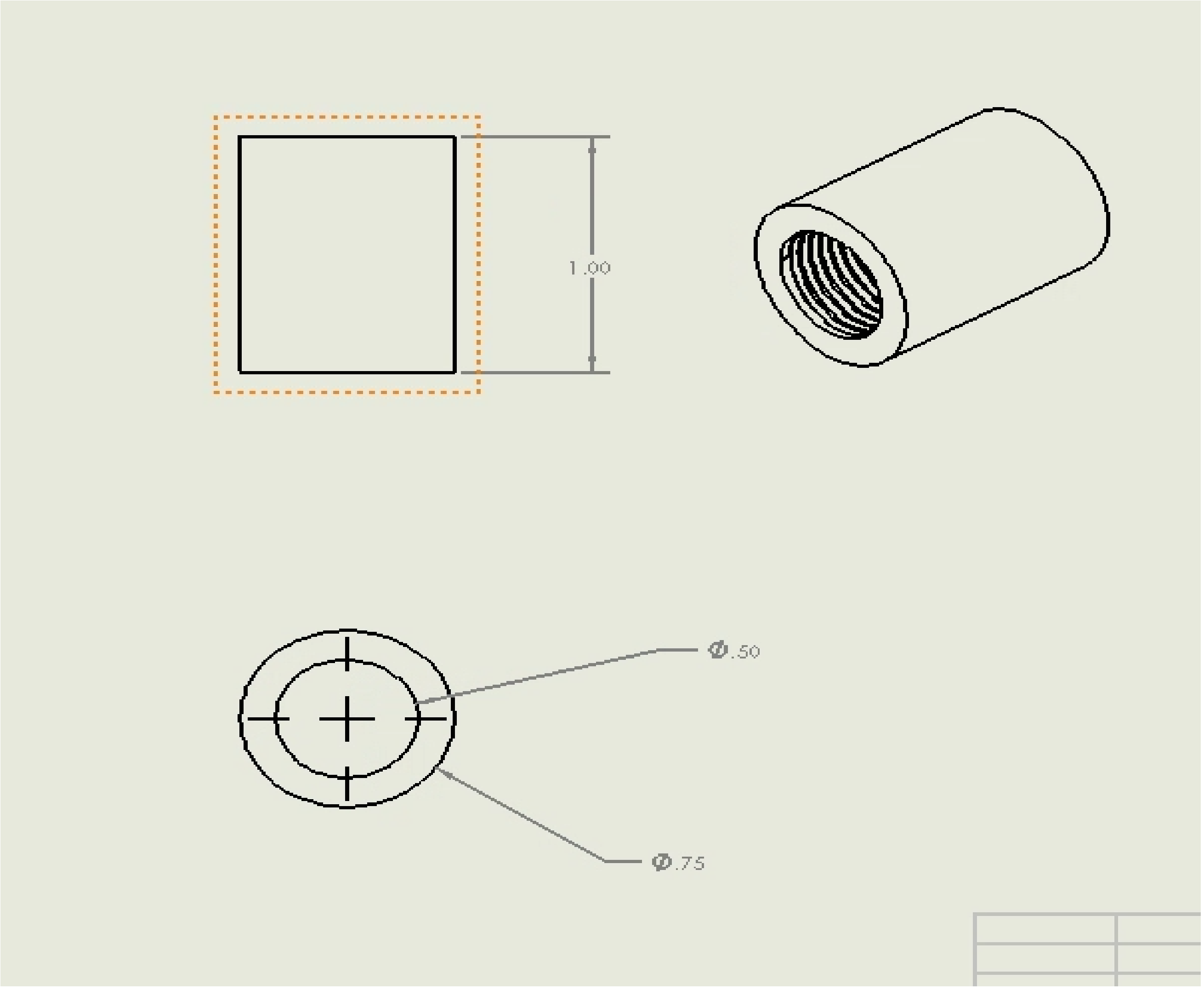


Figure 6.2.4 - Engineering Drawing for Part 1

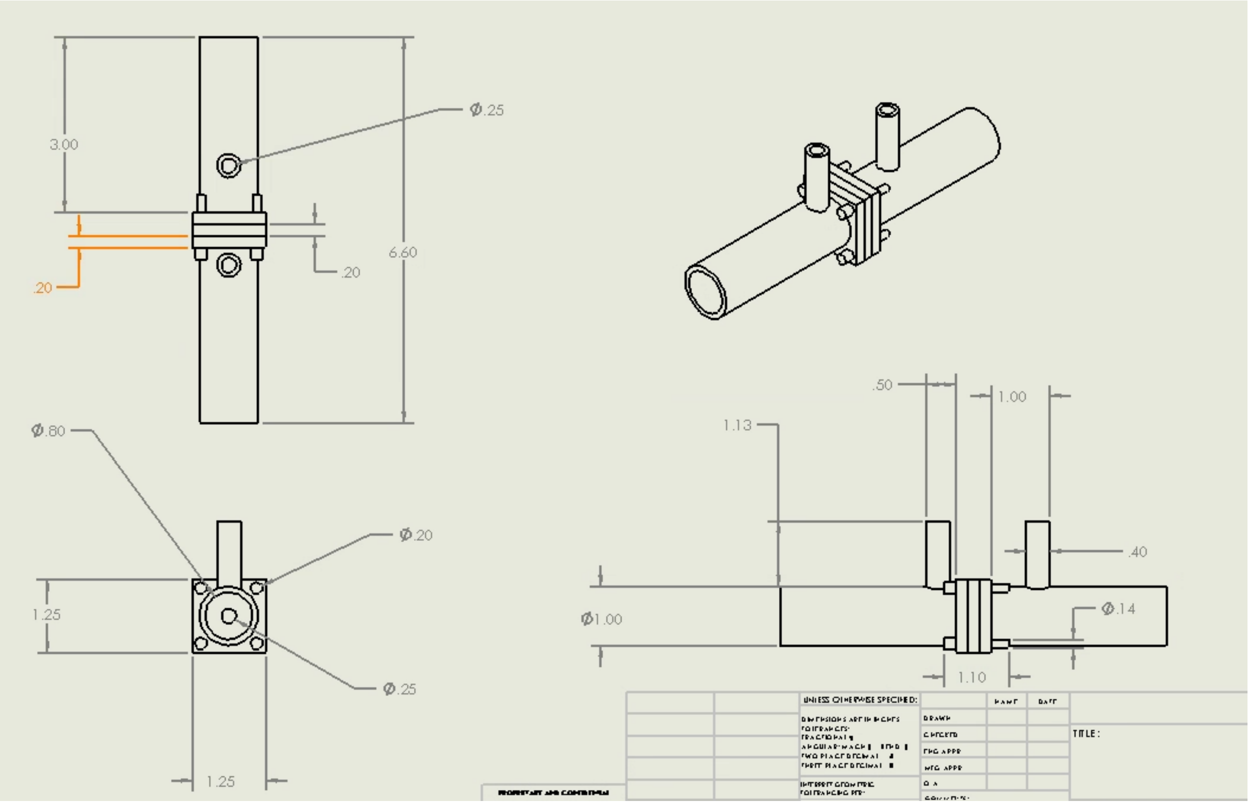


Figure 6.2.5 - Engineering Drawing for the assembly of parts 2,3,4 and the pins

6.2.3 - Mechanical

The items required for this section:

- 3D printer
- Used Wood and screws
- Required power tools
- Epoxy Resin and kevlar tape
- Pressure tap fittings (used double barb edged ¼" fittings)
- 0.17" ID acrylic flexible hose

Step #1: 3D print all required parts. Using the software CURA (free download online, simply add your solidwork design files to it) it will tell you an estimate of how long each 3D print will take. Using the software, upload your design files to an SD card and insert it into the 3D printer. An example of a 3D printed part can be seen below in Figure 6.2.6. Note: 3D printed material was chosen due to its cost effective benefits. However, to have the most accurate results and a better overall built, these parts can be machined with aluminum, stainless steel, etc (as long as its food-grade)

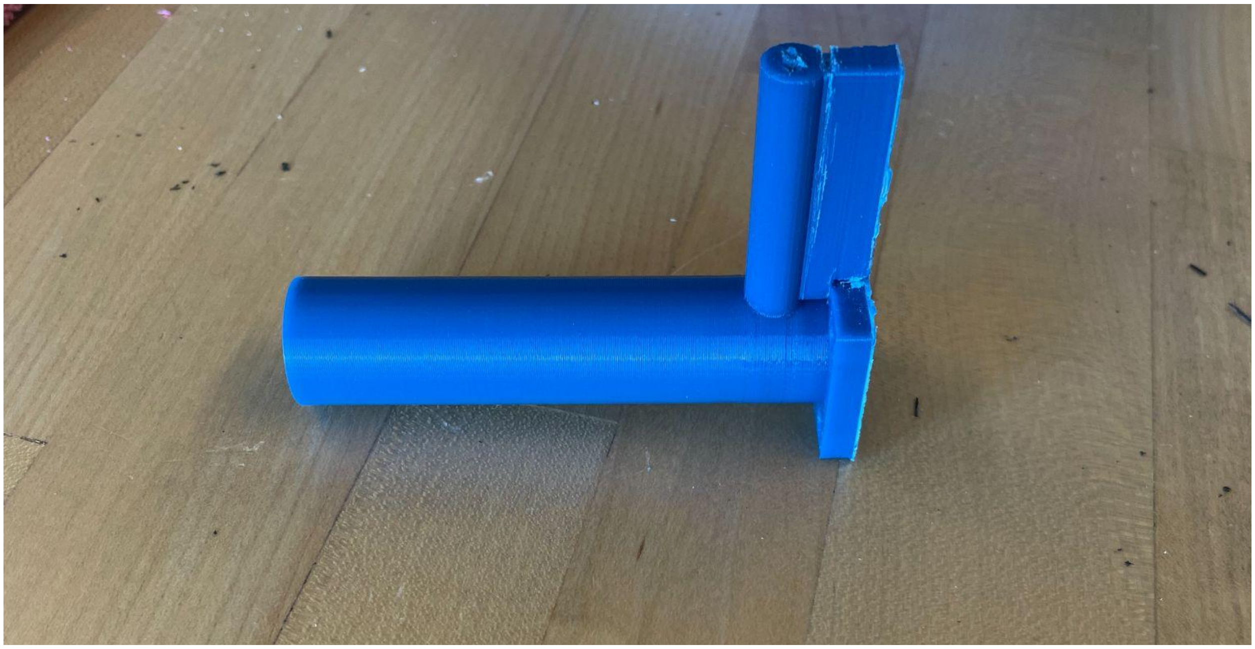


Figure 6.2.6 - Example of 3D printed piece (of part 2)

Step #2: Start the assembly process of all printed components. First, take parts 2,3 and the orifice plate (parts 2 and 3 have the pressure taps) and start by assembling them together using small screws that fit through the pre-designed slots. This connection should be as tight as possible. Once all components are together, add generous layers of epoxy resin to all cracks and edges. This option can be substituted by other waterproof/sealing methods like gaskets or sealants. After all epoxy is dry and sealed, prepare the pressure taps, ensuring that the holes are cleared of any debris or blockage. Once the pressure taps are prepped, insert the double edged fittings into the slots, along with kevlar tape (or any glue/gasket system) to ensure a water/air tight seal (see Figure 6.2.7 for assembly of parts 2,3 & orifice plate).

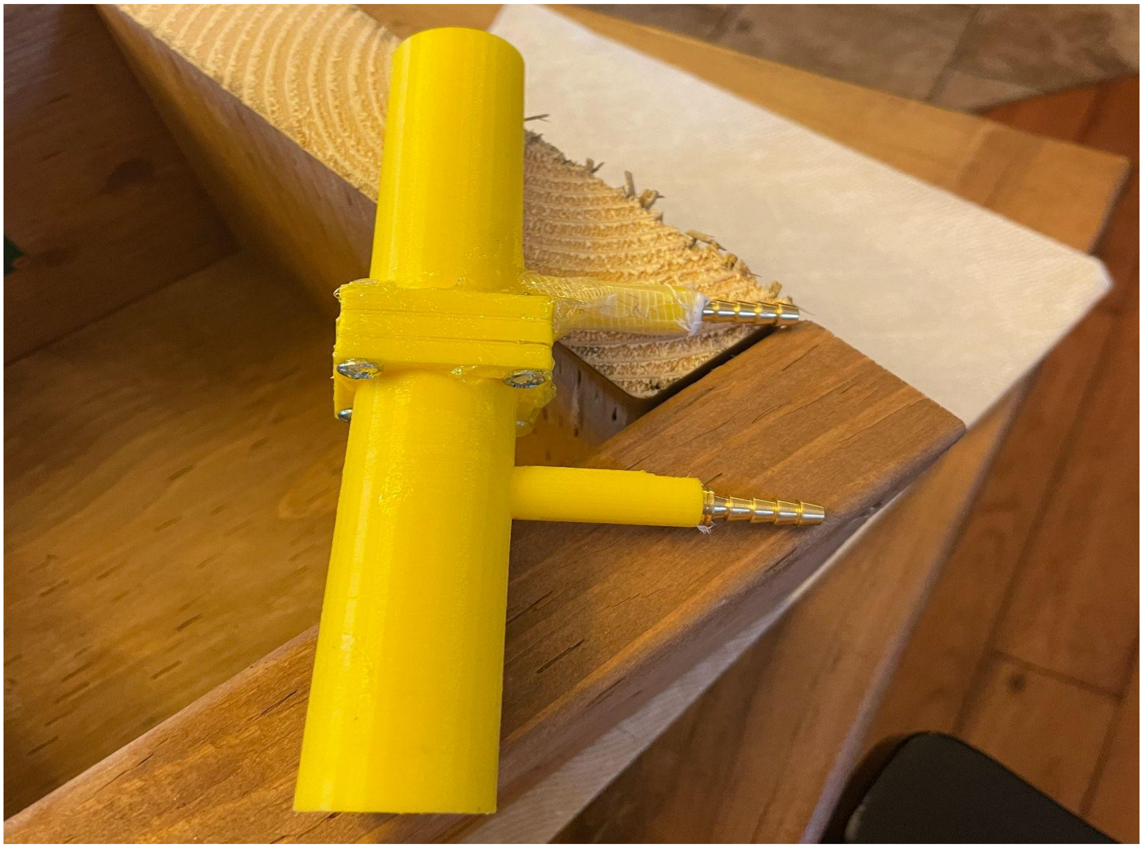


Figure 6.2.7 - Assembly of parts 2, 3, and the orifice plate

Step #3: Build the required box for all electrical components along with a slot for the LCD screen visibility. This part is optional in the sense that 3D printing a box is another valid option. However, upon trying to 3D print, the estimate printing time was approximately 27 hours which is impractical and unfeasible. Thus, we opted for a physical build of a box using screws, a drill and a miter saw to cut and assemble the pieces of wood (See Figures 6.2.8 and 6.2.9 for a more in depth look at the box built.

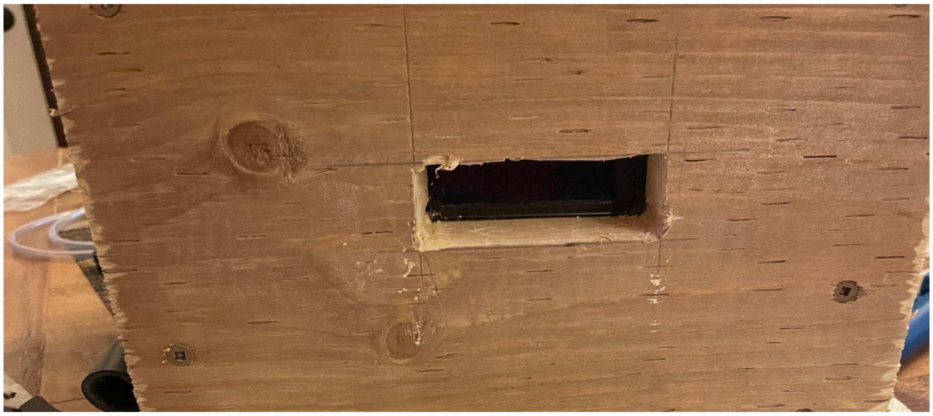


Figure 6.2.8 - Pre-built slot and the LCD screen in the box

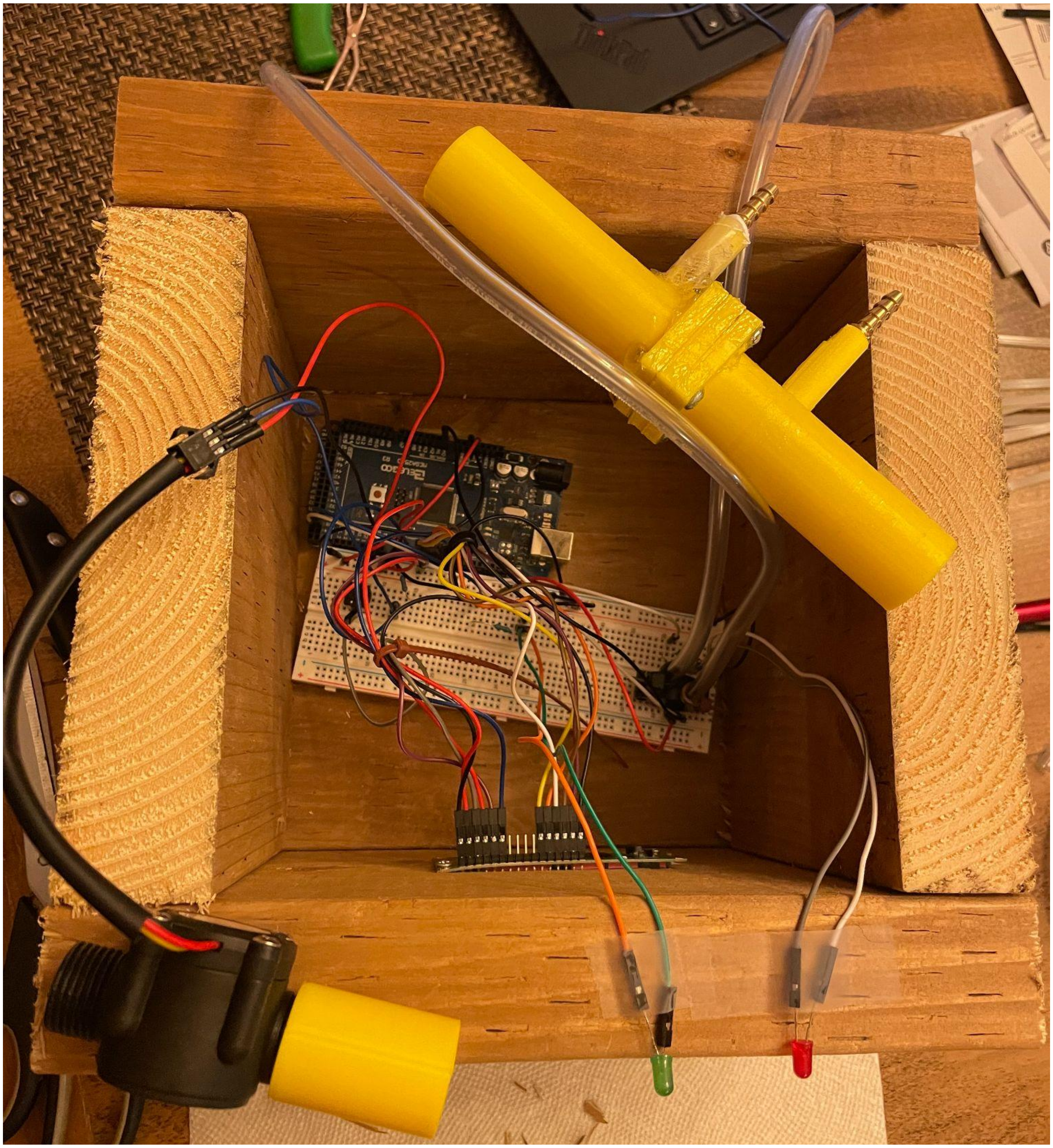


Figure 6.2.9 - Built box to hold all electrical components

Step #4: Install all electrical components inside the pre-built box and connect the device to the water hose. This step is for testing the accuracy of the design by testing it with water flow from a garden hose. To secure the water hose to the device, multiple layers of tuck tape were used to ensure an appropriate seal during testing. Any method to secure liquid flow in a sealed environment can be used (see figure 6.2.10 for the assembly of the device).

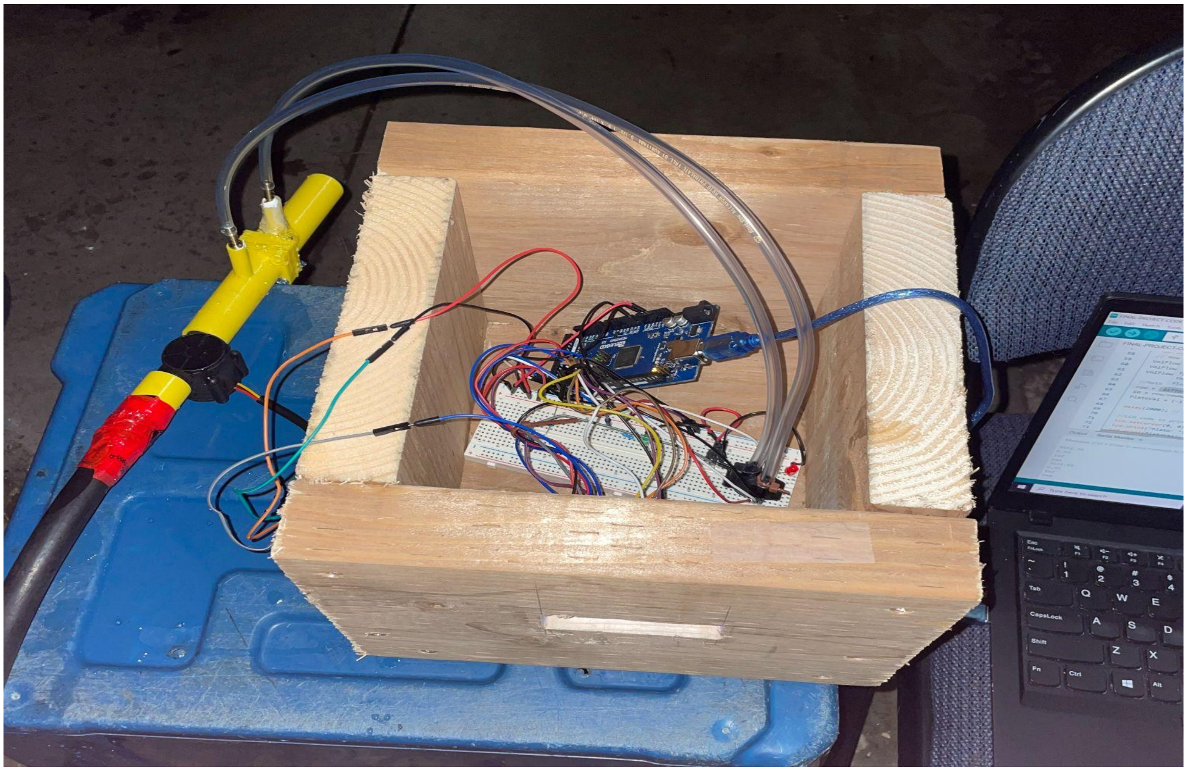


Figure 6.2.10 - Assembled device with the built box

6.3 - Testing & Validation

The first step in the testing phase was to individually test all important components of the design. The first test done was to ensure the LCD screen was in working condition and could print a required message to the screen (see figure A6.1 for the code used in this step) The test was carried out and completed on October 24th, to which an LCD screen, an Arduino Mega, a 10K Potentiometer, and a breadboard were utilized. Without the I2C device, the wiring and setup were more complex (See Figure 6.1 for this setup). The Arduino was able to print the required message successfully: “This is good” along with the length of it working in seconds, indicating that the LCD screen works and can print our desired value (see Figure 6.2 for the printed message).

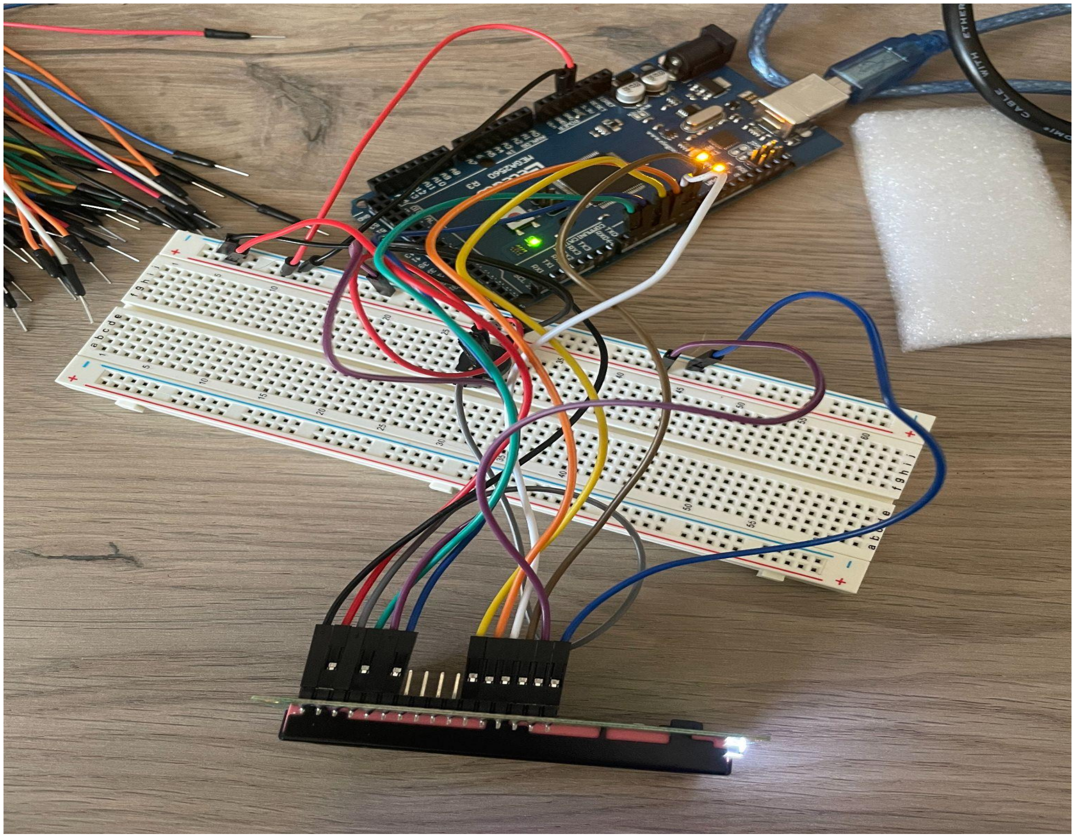


Figure 6.1 - Wiring Setup for LCD screen

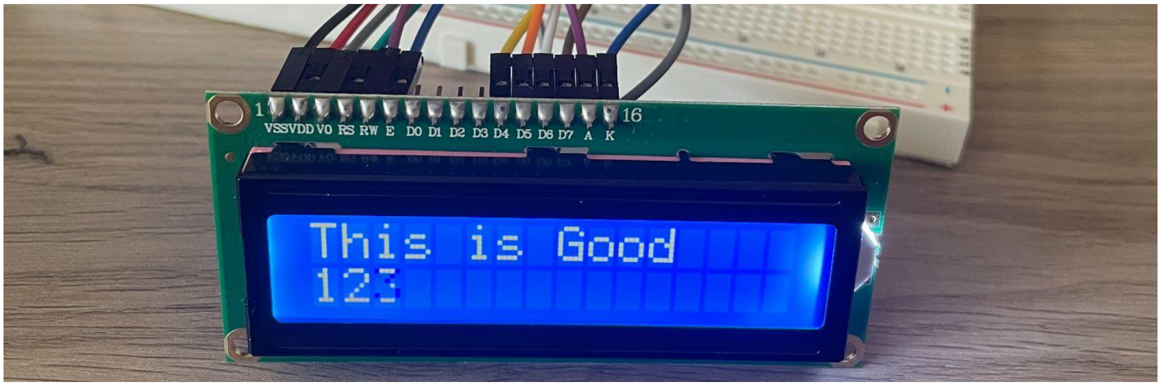


Figure 6.2 - Printed Message on LCD screen

The second test conducted was to test was to make sure the LED lights can turn on based on a required if/else statement and a set of values. The test was carried out and completed on October 24th, to which a red light and green light were utilized to turn on based on the coded requirements. For this test, the red light was activated for the first 20 seconds, while after 20 seconds, the green light was turned on. (See figure A6.2 for the code used in this test) This simulated the requirment for when a plato value is in the desired or non-desired range. The Arduino was able to correctly turn on the green and red led lights based on the if/else statement coded. This successfully indicates that the red and green lights can be used to set the boundaries for good and bad Plato values for our system (see figures 6.3 and 6.4 for the red and green lights being turned on).

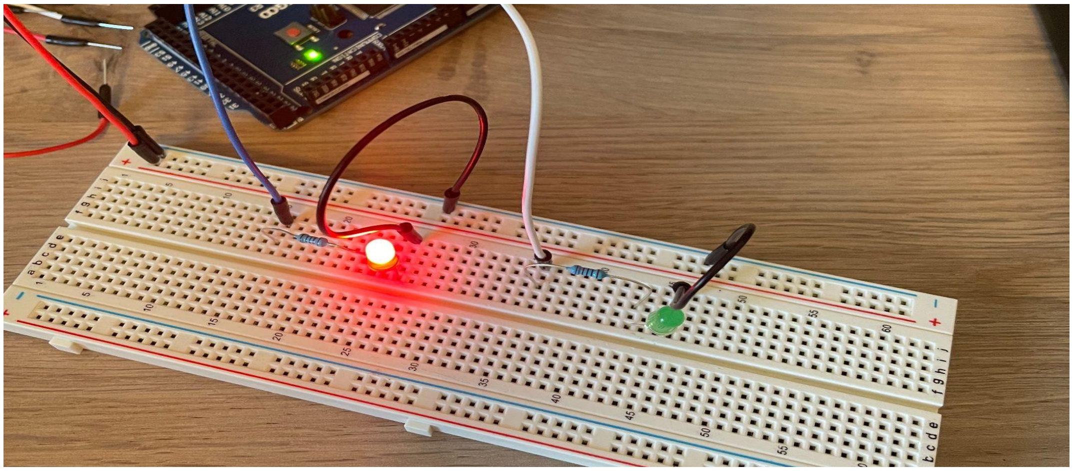


Figure 6.3 - Red light being turnt on when time is less then 20 seconds

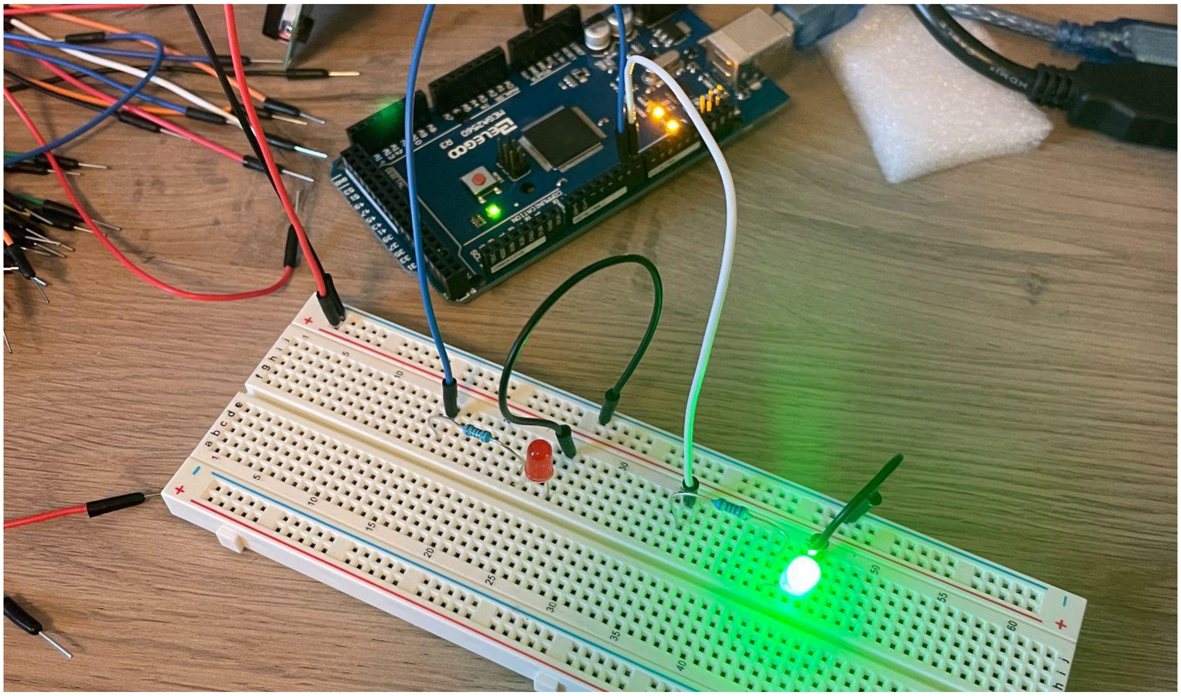


Figure 6.4 - Green Light turnt on when time is greater then 20 seconds

The third test accomplished for the project was to test the differential pressure sensor. The goal for this test was to ensure the differential pressure sensor could read the difference in pressure in Kpa under 2 conditions

1. Both ends are free (so the difference in pressure should be 0)
2. One end will is blown into (to change the difference in pressure, the value should be larger than 0)

The test was carried out and completed on November 5th (when the differential pressure sensor was received), to which the sensor was connected via the breadboard and original wiring diagram. The sensor was then attached by 2 pieces of clear vinyl tubing (with an ID of 0.17”) to simulate both attachments to the pressure taps. The code was run by Arduino and successfully implemented. Figure 6.5 demonstrates the layout of this prototype (See Appendix Figure A6.3 for the code used in this step). The differential sensor was able to read the difference in pressure under both conditions. Figure 6.6 demonstrates the difference in pressure when both tubes were left alone (should be 0 Kpa). A slight difference in error was seen which was expected due to the sensitivity of the sensor and its conditions (however this error was relatively minor). In Figure 6.7, the results for the trial with air blown into the positive pressure tube was completed and the results successfully demonstrated an increase once the blowing process started.

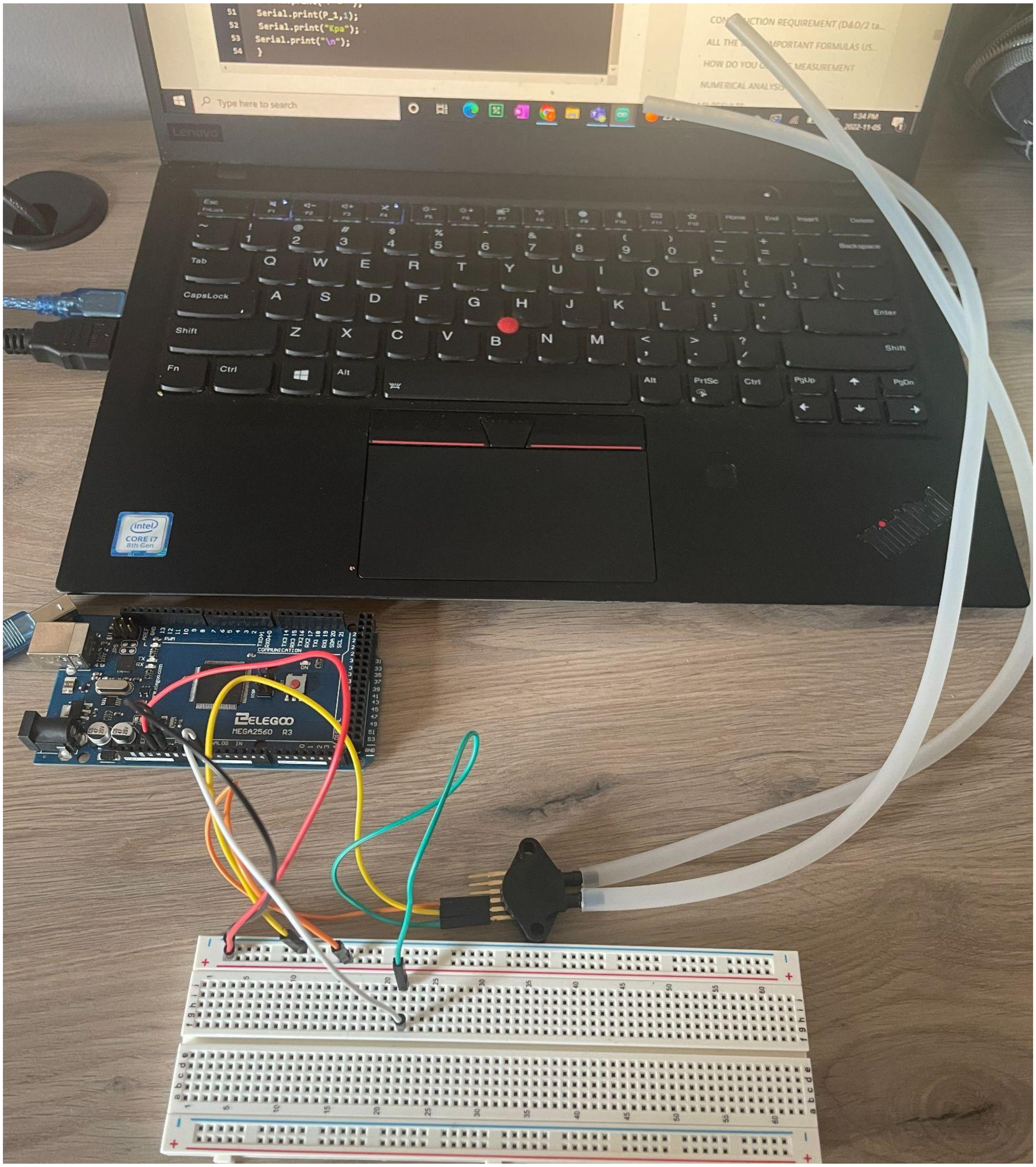
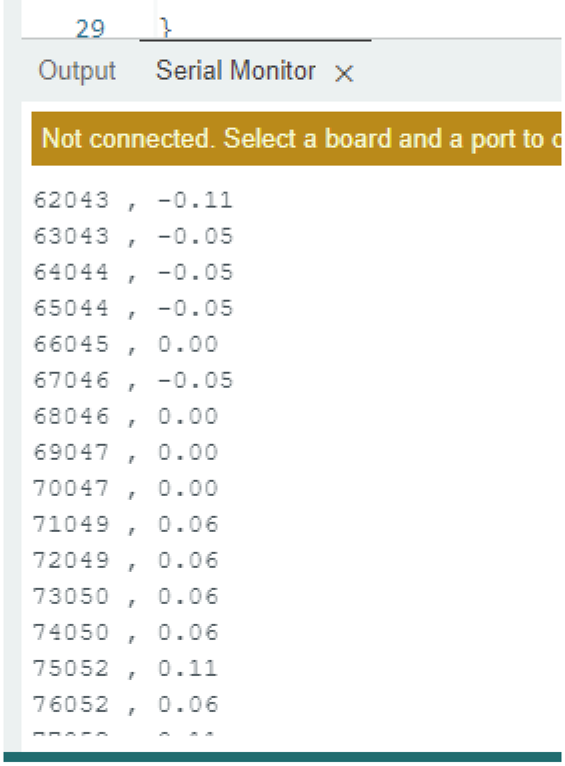


Figure 6.5 - Differential pressure sensor practice testing

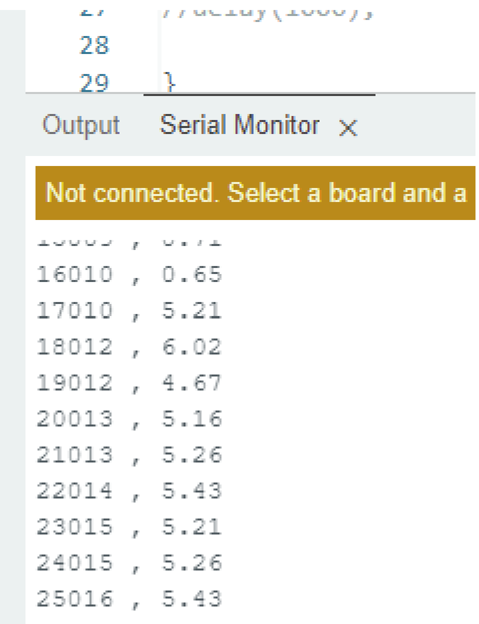


```
}
```

Not connected. Select a board and a port to

```
62043 , -0.11
63043 , -0.05
64044 , -0.05
65044 , -0.05
66045 , 0.00
67046 , -0.05
68046 , 0.00
69047 , 0.00
70047 , 0.00
71049 , 0.06
72049 , 0.06
73050 , 0.06
74050 , 0.06
75052 , 0.11
76052 , 0.06
```

Figure 6.6 - Results for Trial 1 (when both tubes remained untouched)



```
}
```

Not connected. Select a board and a

```
16010 , 0.65
17010 , 5.21
18012 , 6.02
19012 , 4.67
20013 , 5.16
21013 , 5.26
22014 , 5.43
23015 , 5.21
24015 , 5.26
25016 , 5.43
```

Figure 6.7 - Results for Trial 2 (when 1 side was blown air into it)

The fourth test carried out was to physically print the theoretical design using a 3D printer. The goal of this test was to ensure the design concept could physically be built in a cost-effective manner.The test was carried out on November 8th and continued until November 11th. The printing process occurred at the Makerspace. The first part printed was part #2 to which the orifice plate would stick to it on one end. (See Figure 6.8 below). The Parts were successfully printed with good material strength and toughness. Upon visual inspection, it seems the material and design should be successful in resisting water leaks and should resist the necessary pressure and flow rates in the in-line system. (However, a few printing errors occurred, see section 6.2.1). The rough edge of the pressure tap was caused by the printed support which was then removed.

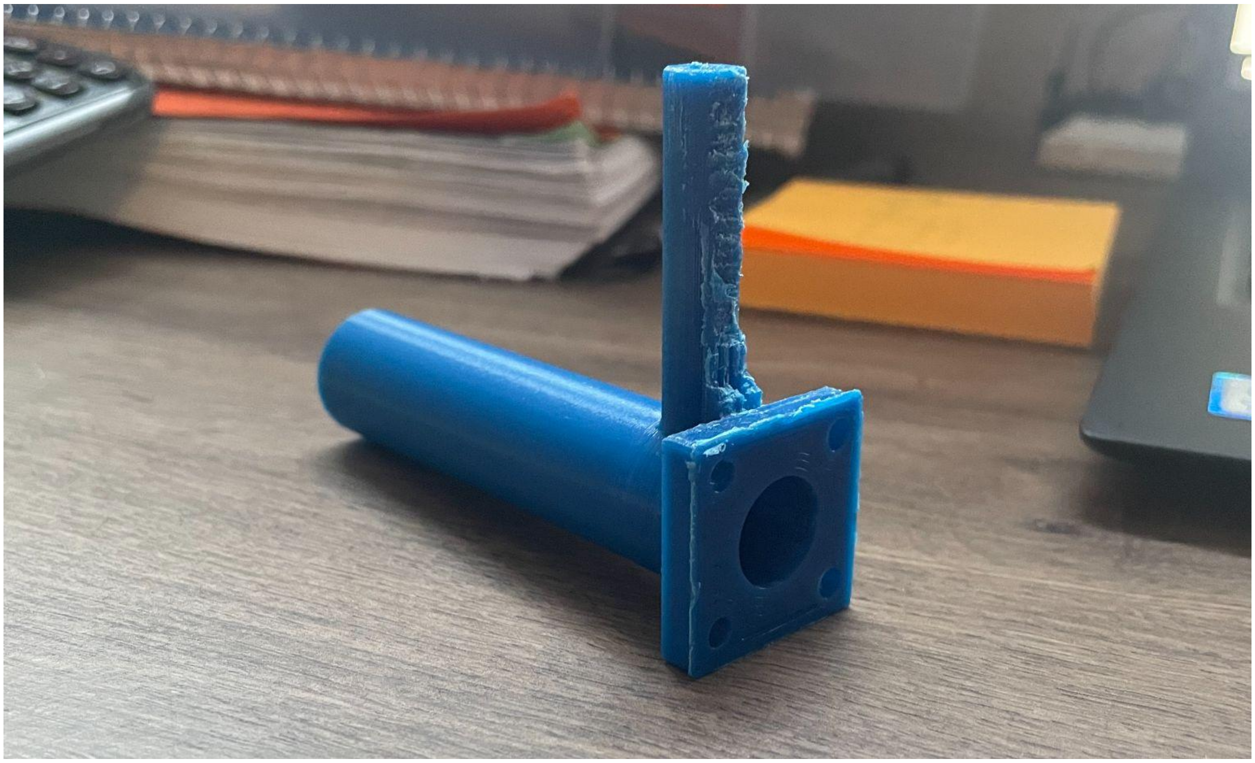


Figure 6.8 - Part 2 creation in 3D print

In addition, the fith test carried out was in regards to the orifice plate stability. The goal of this test was to ensure the orifice plate was thick and stable enough to resist the low pressure and flow rate throughout the pipe. The test was carried out on November 10th and occurred at the Makerspace. The orifice plate was successfully printed and was tested for stability by applying various weights from 2lbs, to 5lbs. In conclusion, the orifice plate is stable enough to withstand the forces required during the flow process (see Figure 6.9 for the physical orifice plate).

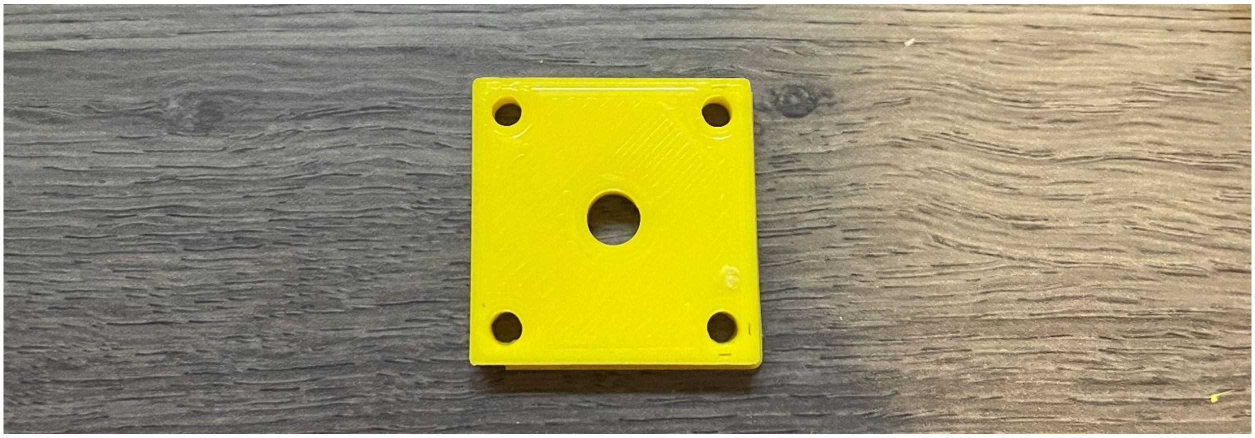


Figure 6.9 - Orifice plate successfully 3D printed

The sixth test carried out was regarding the assembly of all individual electronic components. The goal of this test was to ensure all components can be properly wired and connected to the Arduino and breadboard. The test was carried out on November 11th in which all components were wired and imputed to the correct Arduino pins based on the written code. All components were successfully connected to the breadboard and the Arduino was able to turn on without issues. The wiring assembly of all components can be seen below in Figure 6.10.

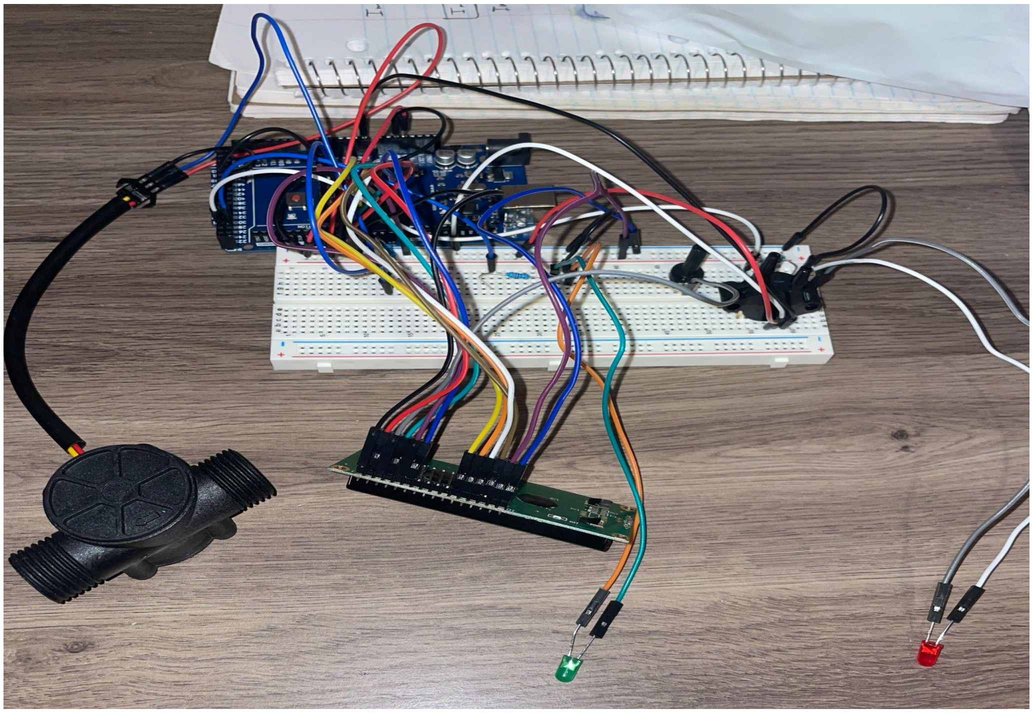


Figure 6.10 - Wiring of all components

The seventh test conducted was to test the volumetric flow meter. The goal of this test was to ensure that the volumetric flow sensor could read the flow value of water in ml/s and display the results in the serial monitor. The sensor was wired to the Arduino breadboard and the test was performed using a kitchen sink to create the water flow (see Figure 6.11) The sensor was able to read the value of water flowing through it while simultaneously printing it onto the serial monitor (see Figure 6.12). This demonstrated the functionality of the sensor under basic flowing conditions. The practice code for the volumetric flow meter can be seen in the appendix section 9.2.

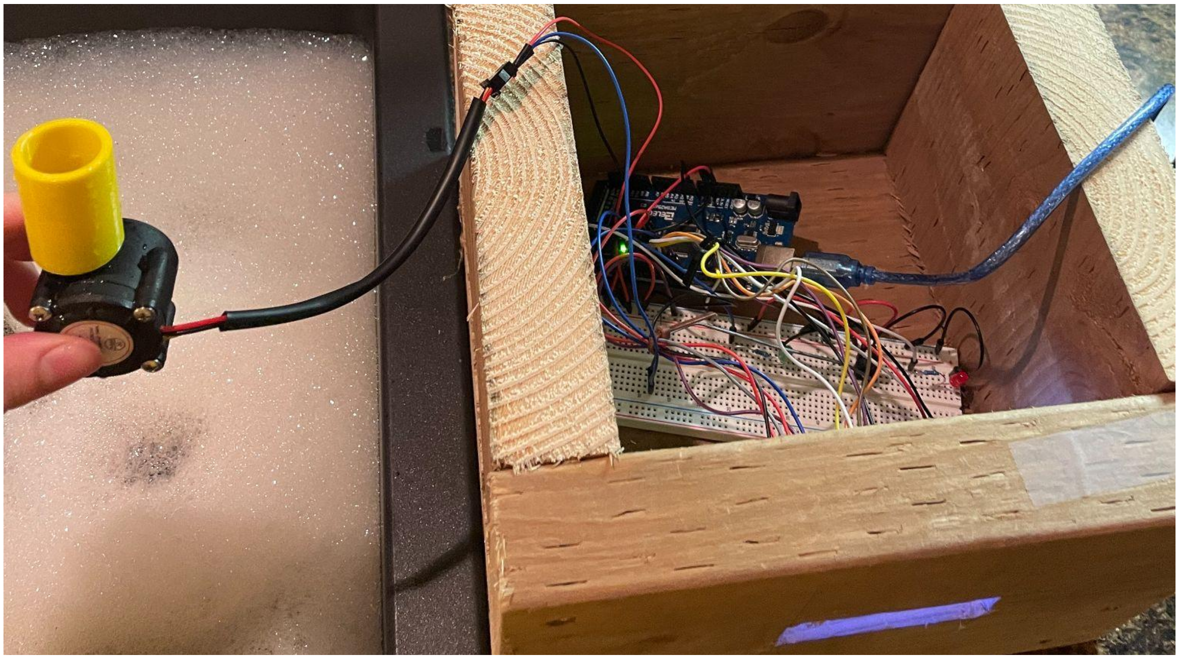
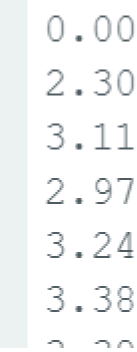


Figure 6.11 - Flow Meter Testing Layout



```
0.00
2.30
3.11
2.97
3.24
3.38
```

Figure 6.12 - Flow meter reading results during testing in m^3/s

The eighth test was in regards to finishing the printing process due to initial errors. The goal of this test was to build the 3D model to begin the testing phase of the design.The printing process was completed in the maker space on November 16th. The assembly process for testing was done on November 19th. The 3D printing process was successfully completed with the same blockage being present in the pressure tap designs along with a few errors during the building phase which will be explained in section 6.2.1. The built for parts 2,3, and the orifice plate can be seen in Figure 6.13.

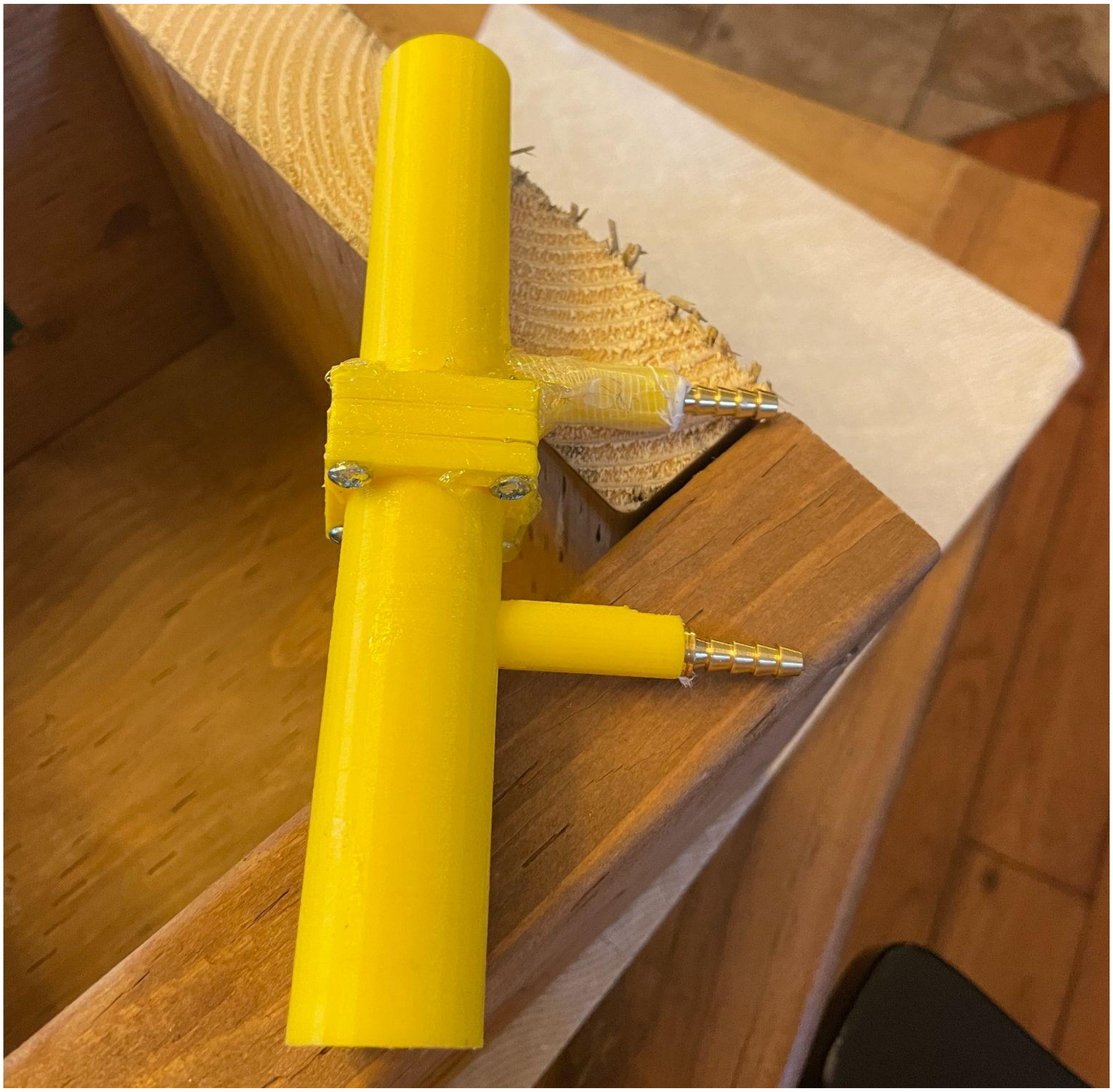


Figure 6.13 - 3D print assembly for parts 2,3 and the orifice plate

The final test was concluded to test the measurement capabilities of the assembled device. The goal of this test was to finish the building process while ensuring the physical device can be properly wired and connected. The finishing touches to the design building process were completed on November 19th. The testing phase of the specific gravity measurement occurred on the same day and the testing occurred using a garden hose to simulate the water in-line system (see Figure 6.14). The design was able to successfully read an SG value of the water. However, due to multiple errors that occurred, the value was approximately 0.68 before starting to fluctuate frequently in ranges that did not make theoretical sense. This SG value reading should have theoretically been 1 as the fluid used for testing was the reference fluid in the specific gravity equation. The average error percentage observed was approximately:

$$\%error = \left|\frac{0.68-1}{1}\right| * 100 = 32\%$$

The results were printed into the Serial Monitor and can be seen below in Figure 6.15.

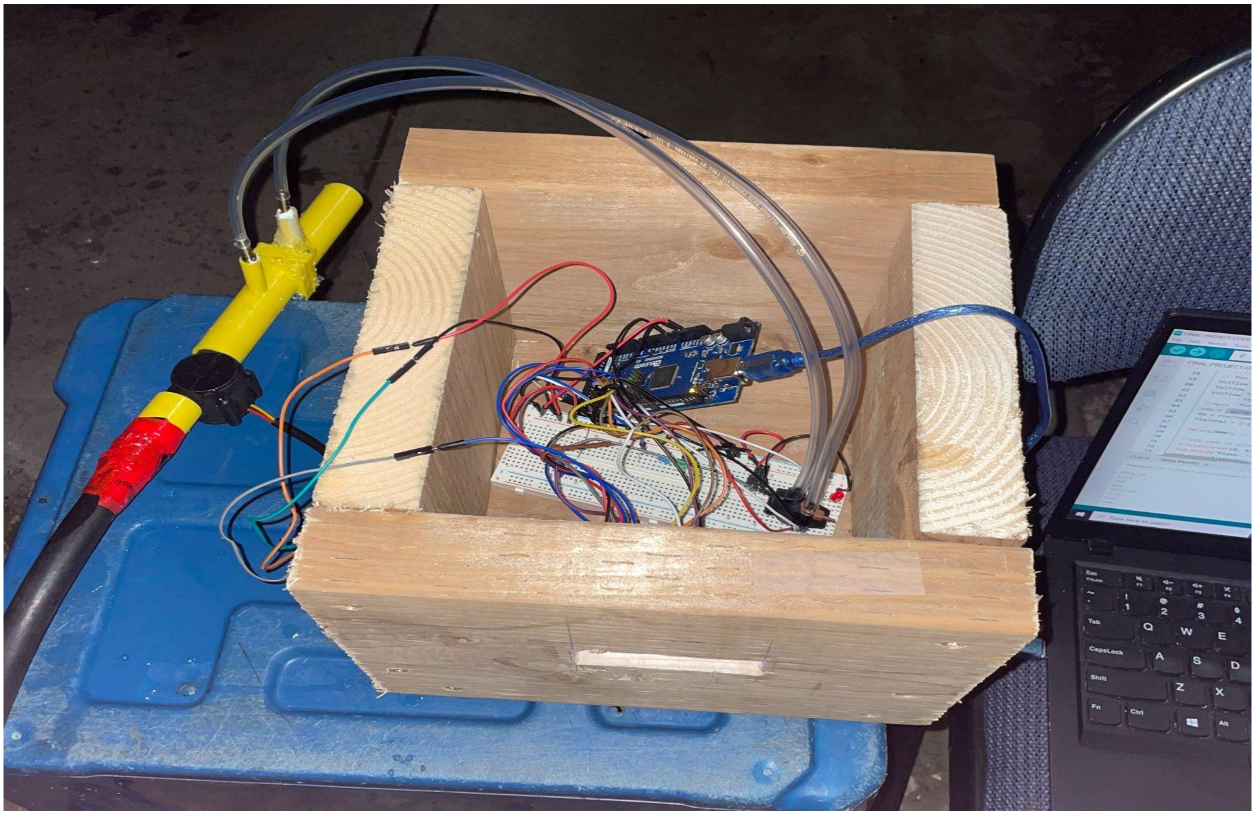
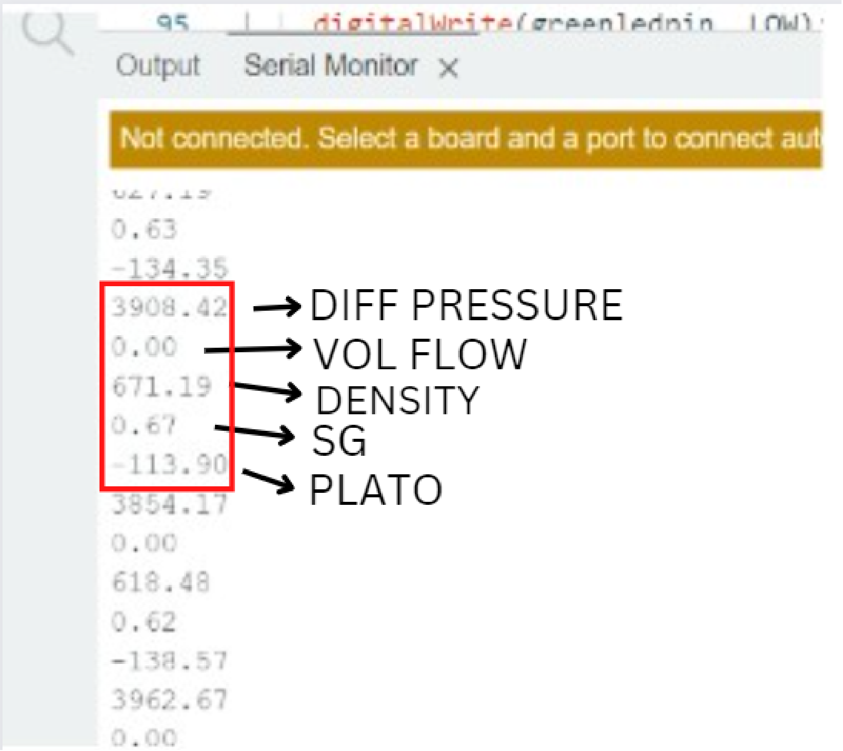


Figure 6.14 - Final Testing Layout



Not connected. Select a board and a port to connect aut

```
0.63
-134.35
3908.42
0.00
671.19
0.67
-113.90
3854.17
0.00
618.48
0.62
-138.57
3962.67
0.00
```

Figure 6.15 - Results on Serial Monitor for SG values

6.2.1- Errors and Difficulties

To begin, the first errors were found after the fourth test, where we printed the original design of our device. The 3D model design had multiple edits due to calculation mistakes and errors once prototyping occurred. The first design modification was the length and the diameter of the pressure tap holes. Figure 7.1 shows the error that occurred during the printing of part 2 for our prototype. Due to the orientation of the printing, the 3D print caused unwanted support to be formed throughout the pressure tube cylinder. This would cause unwanted restrictions in pressure and inaccurate values. To help this, a remade 3D sketch of the pressure taps was made, where the cylinder is now ¼” in diameter and the length of tubes was reduced by ½”. This will enable us to safely machine/drill the excess material out of the pressure tap once printing is done. The reprinting of this prototype will be completed for prototype 3.

The other issue with our main design is the inside pipe diameter. Upon the arrival of the volumetric flow meter, the diameter of the flow meter was too big for the printed part 2. This can be seen in Figure 7.2 in which both OD are the same. This required a redesign for the inside pipe diameter to be 0.8” and the outside diameter of 1” (since the OD of the volumetric flow sensor is 0.78”) which should allow for proper fitting. In addition, due to this increase in overall dimensions, the orifice plate and holders increased to a size of 1.25”x1.25” to allow for proper fitting of the pins. Following the 3:1 decrease ratio for the orifice plate, an adjustment of the orifice hole was changed to 0.25”. This change was also reflected in the code.

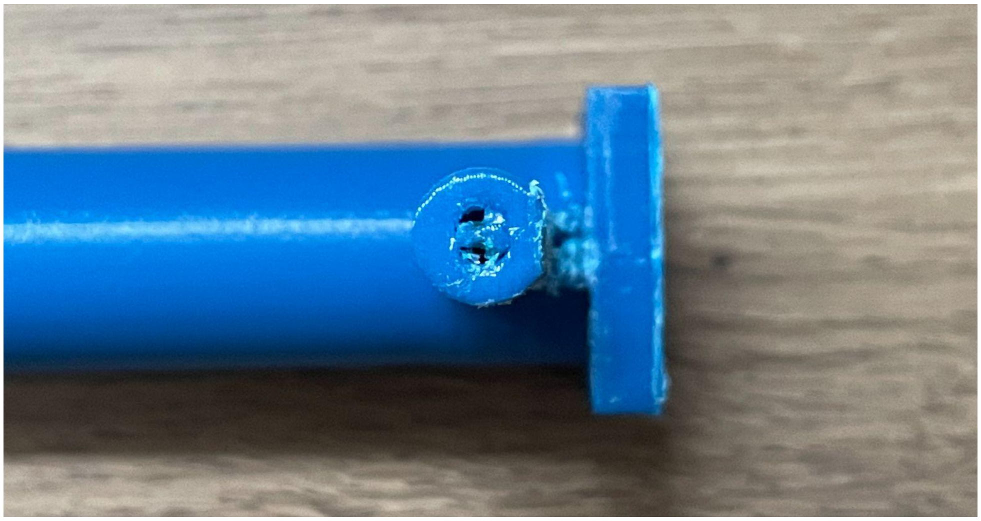


Figure 7.1 - 3D printing error for pressure tap of part 2

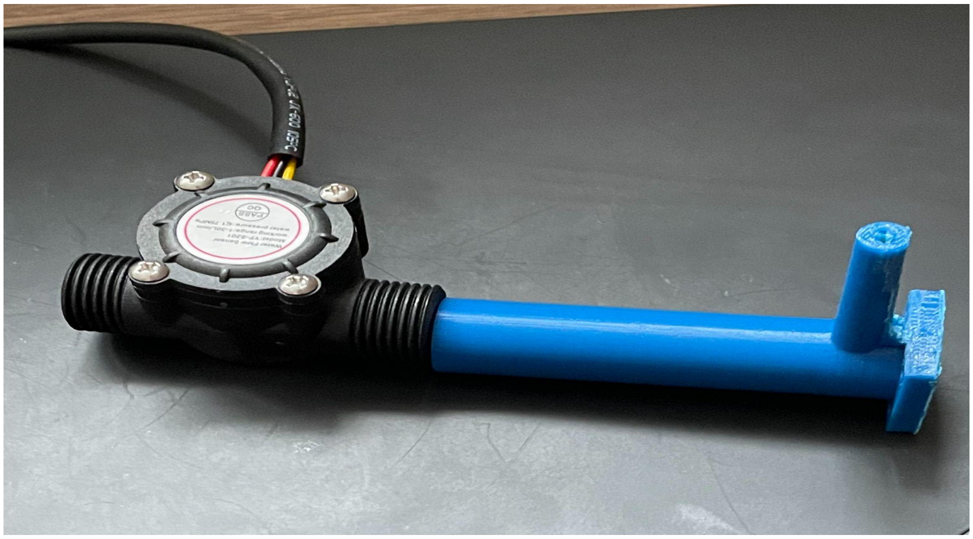


Figure 7.2 - Pipe Diameter sizing error

To continue, the next set of errors occurred during the building and testing phase for the newly printed design. During the testing and building phase, there were, unfortunately, multiple errors that arose, leading to inaccuracy in the results. The first error was during the assembly process of all 3D parts. For parts 2 and 3, careful and precise drilling was required to open up the blockage/supports created during the 3D printed process in the pressure taps. However, during the drilling process, one of the pressure taps cracked, causing a major hole in the physical device (see Figure 7.3). This hole was a major concern and disappointment due to the high chance of the taps no longer being waterproof/air sealed tight. In an attempt to fix this mistake, waterproof tape and epoxy resin was heavily added to all areas of the crack to try & create a sustainable seal. The next error occurred during the attempted seal of the orifice plate and the 2 plated areas for parts 2 and 3. All 3 rectangular sections were screwed together tightly with the premade slots. In addition to this, epoxy resin was generously added to all surfaces and cracks. Upon testing of the entire assembly, it was seen that both solutions did not sustain in keeping proper air and water seal. The crack area was ejecting large amounts of water while the 3 plates were slightly letting water pass (see Figure 7.4). This lack of seal obviously created multiple errors in our specific gravity measurement. Finally, another error that occurred is the accuracy of the volumetric flow meter sensor. As seen in figure 6.16, the flow meter reads a value of 0 during the trial process. This was an error from the sensor itself as the other values depend on the flow volume, thus meaning that the flow meter was reading a value, sending it to equations but not saving/displaying it. This error questions the accuracy of the cheap volumetric flow meter and leads to inaccurate results when calculating the specific gravity.

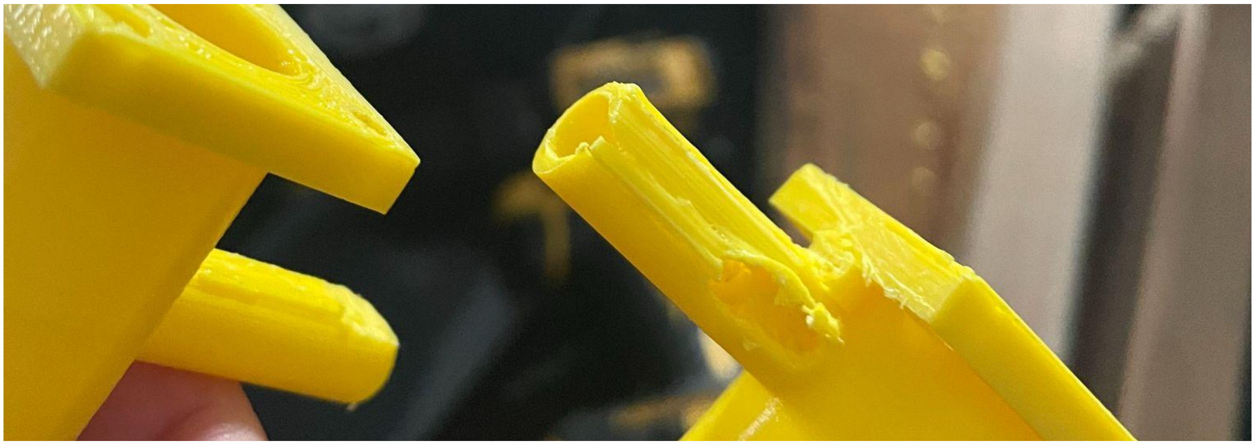


Figure 7.3 - Cracking in the pressure tap during drilling

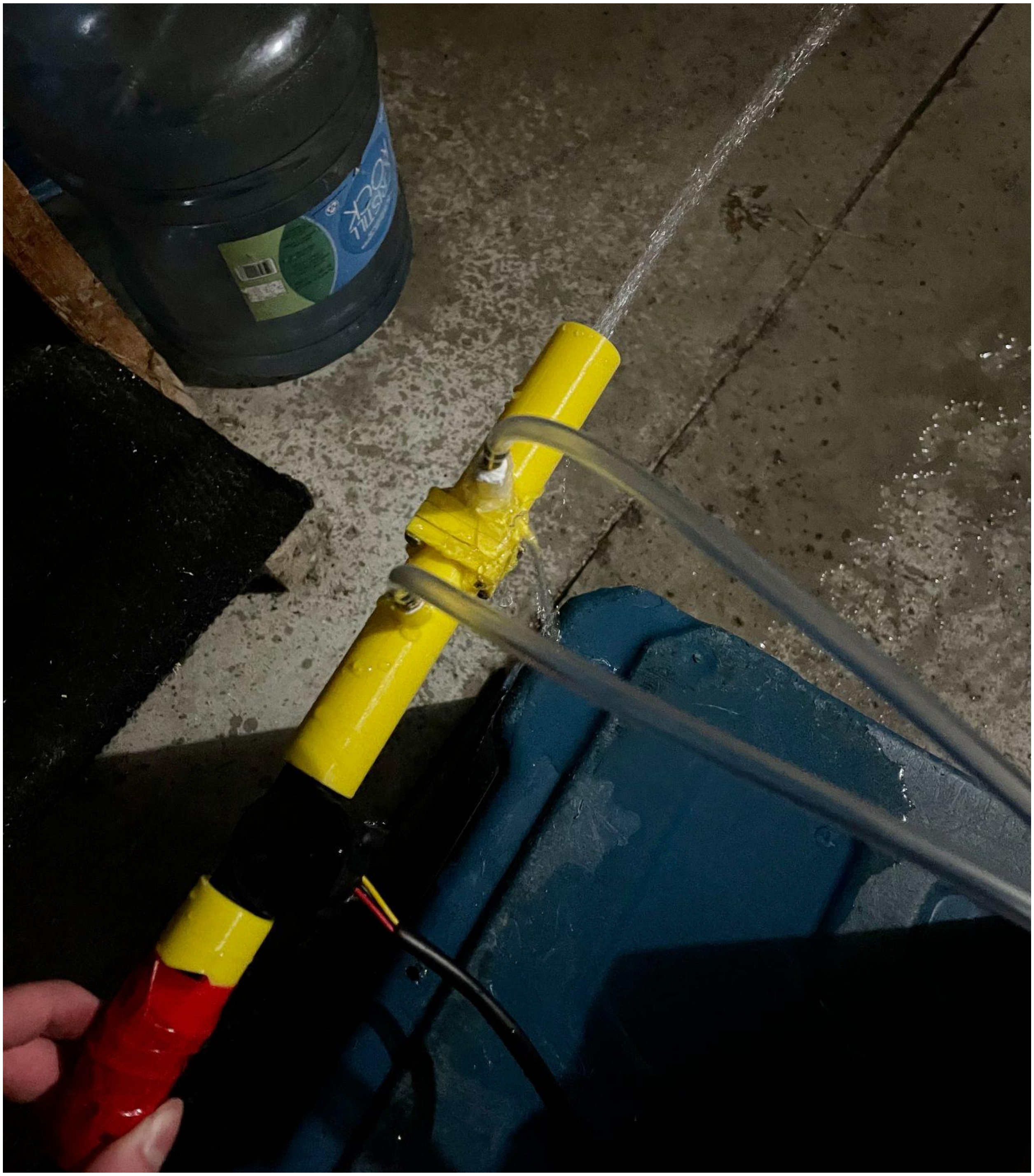


Figure 7.4 - Water leakage when testing

7.0 - Conclusions and Recommendations for Future Work

We have learned many lessons during this project that could have drastically improved our finished project if we were to redo this project. The biggest lesson we learned was time management, we didn't fully understand the importance of testing and the importance of making time for testing. Both of our different tests showed us new problems we didn't see before, given more time we would test more to try to finalize our design. Another lesson we learned was the importance of the makerspace, it is a very important resource that we didn't use enough. Given more time we would've printed our design many more times as there are some problems you can only see when physically holding and testing the design. The third lesson we learned was the importance of having perfect designs in solidworks. Many times we went to print our design but there was an error in our design which would make the print job useless and incapable of testing. Our biggest suggestions for new groups is to stay on top of time management, use makerspace as much as you can and begin testing as soon as you can. If we had a few more months to work on this project we would have done many more tests to work out all the kinks in our design and try to reduce our error in accuracy as much as possible. In the end we had to abandon the attempt of finding a better way to seal our design from leakage, given more time we would have found a more reliable sealing method.

8.0 - Bibliography

[1] “Specific gravity,” Encyclopædia Britannica. [Online]. Available: https://www.britannica.com/science/specific-gravity. [Accessed: 02-Dec-2022].

[2] “Brewmaster Definition & meaning,” Merriam-Webster. [Online]. Available: https://www.merriam-webster.com/dictionary/brewmaster. [Accessed: 02-Dec-2022].

[3] “Orifice, nozzle and Venturi Flow Rate Meters,” *Engineering ToolBox*. [Online]. Available: https://www.engineeringtoolbox.com/orifice-nozzle-venturi-d_590.html.[Accessed: 02-Dec-2022].

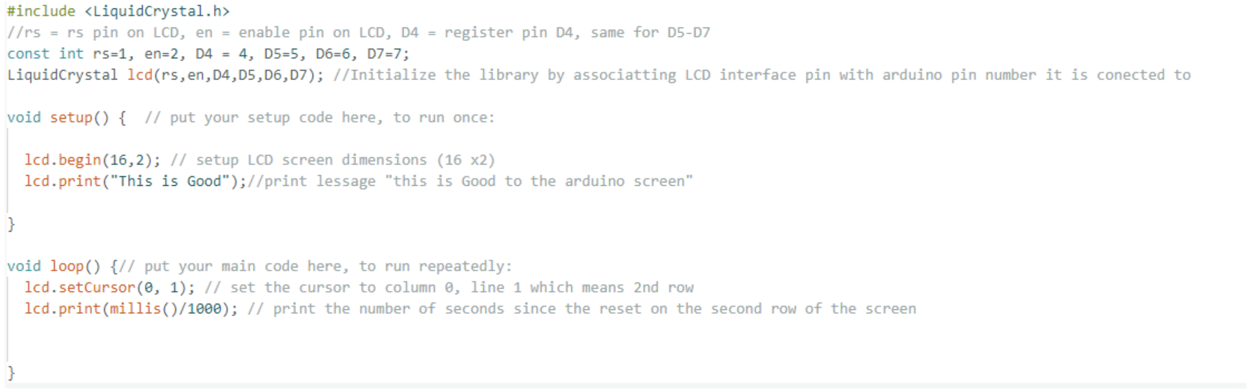
9.0 - APPENDICES

9.1 - APPENDIX I: Design Files

Table 3 - Referenced Documents

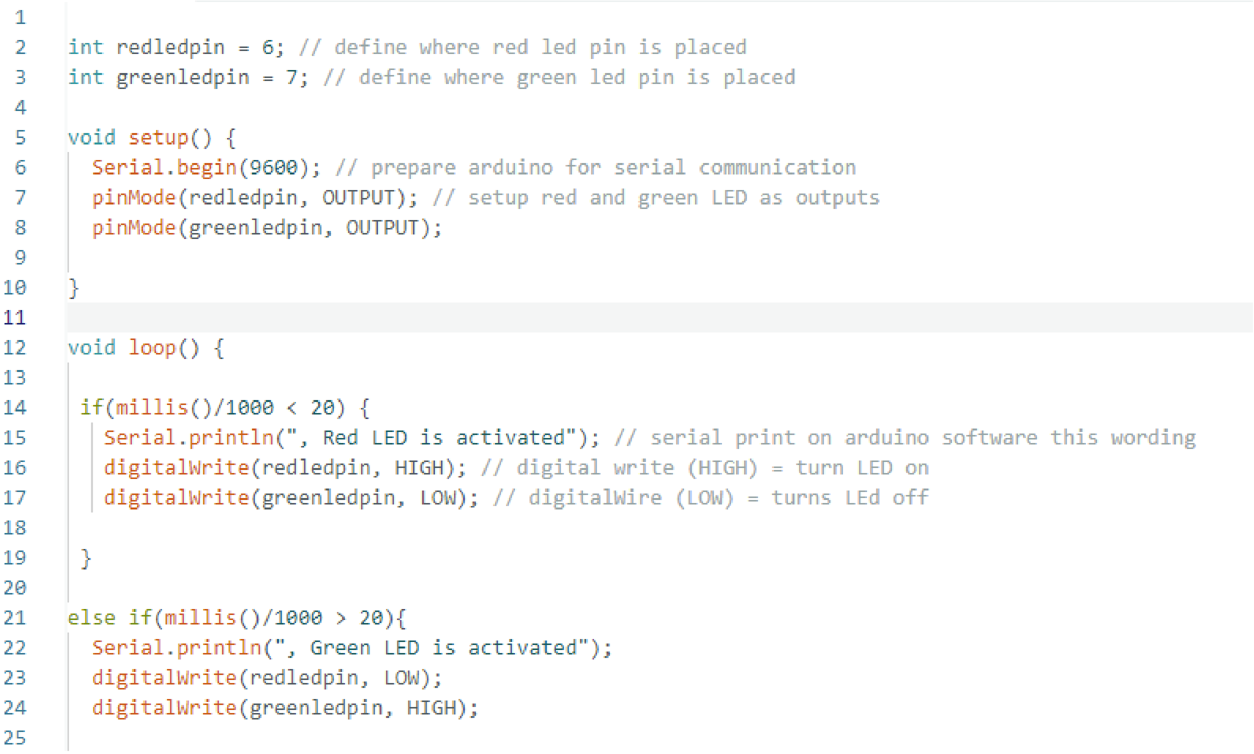
Document Name	Document Location and/or URL	Issuance Date
Deliverable B - Needs Identification and Problem Statement	https://makerepo.com/aplou2001/1319.gng1103b3specific-gravity-measurement-device	17/11/22
Deliverable C - Design Criteria	https://makerepo.com/aplou2001/1319.gng1103b3specific-gravity-measurement-device	17/11/22
Deliverable D - Conceptual Design	https://makerepo.com/aplou2001/1319.gng1103b3specific-gravity-measurement-device	17/11/22
Deliverable E - Project Plan and Cost estimate	https://makerepo.com/aplou2001/1319.gng1103b3specific-gravity-measurement-device	17/11/22
Deliverable F - Prototype I	https://makerepo.com/aplou2001/1319.gng1103b3specific-gravity-measurement-device	17/11/22
Deliverable G - Prototype II	https://makerepo.com/aplou2001/1319.gng1103b3specific-gravity-measurement-device	17/11/22
Deliverable H - Prototype III	https://makerepo.com/aplou2001/1319.gng1103b3specific-gravity-measurement-device	17/11/22
Deliverable I - Design Day Material	https://makerepo.com/aplou2001/1319.gng1103b3specific-gravity-measurement-device	17/11/22
Deliverable J - Project Presentations	https://makerepo.com/aplou2001/1319.gng1103b3specific-gravity-measurement-device	17/11/22
Deliverable K - User and Product Manual	https://makerepo.com/aplou2001/1319.gng1103b3specific-gravity-measurement-device	17/11/22
Final Code for the project	https://makerepo.com/aplou2001/1319.gng1103b3specific-gravity-measurement-device	17/11/22
Final assembly for the project	https://makerepo.com/aplou2001/1319.gng1103b3specific-gravity-measurement-device	17/11/22

9.2 - APPENDIX II: Other Appendices



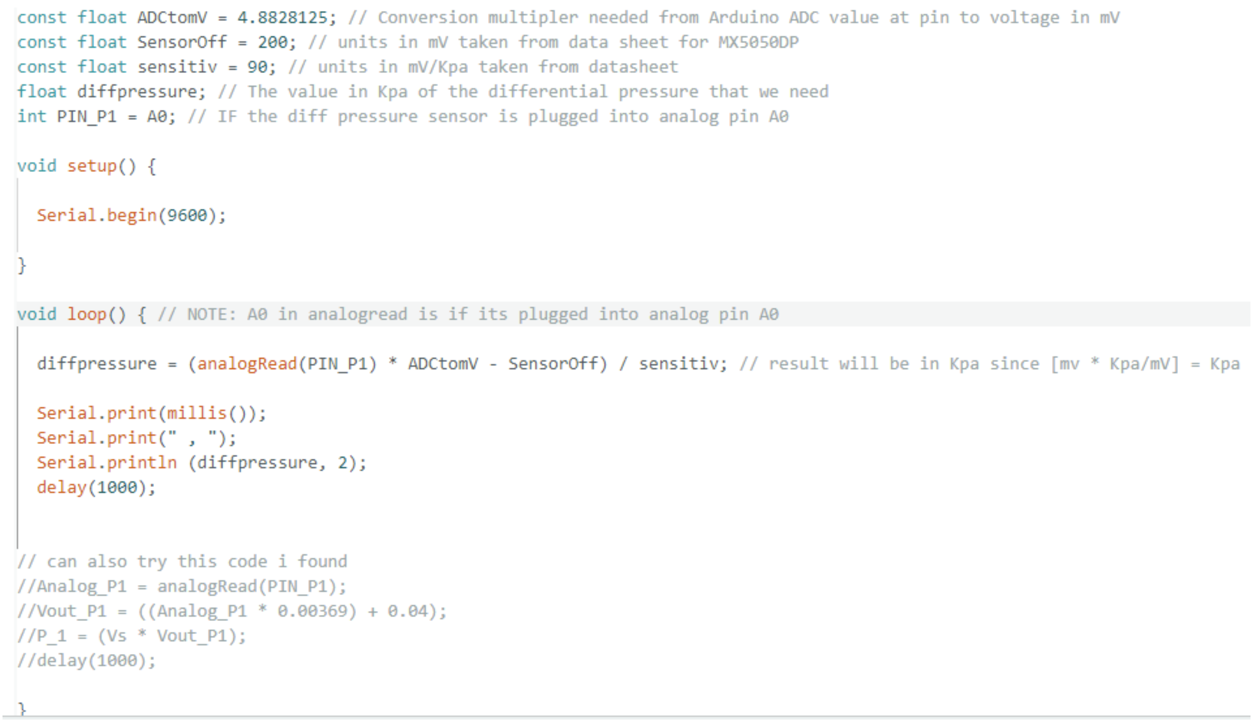
```
#include <LiquidCrystal.h>
//rs = rs pin on LCD, en = enable pin on LCD, D4 = register pin D4, same for D5-D7
const int rs=1, en=2, D4 = 4, D5=5, D6=6, D7=7;
LiquidCrystal lcd(rs,en,D4,D5,D6,D7); //Initialize the library by associatting LCD interface pin with arduino pin number it is conected to

void setup() {  // put your setup code here, to run once:

  lcd.begin(16,2); // setup LCD screen dimensions (16 x2)
  lcd.print("This is Good");//print lessage "this is Good to the arduino screen"

}

void loop() {// put your main code here, to run repeatedly:
  lcd.setCursor(0, 1); // set the cursor to column 0, line 1 which means 2nd row
  lcd.print(millis()/1000); // print the number of seconds since the reset on the second row of the screen


}
```

Figure A6.1 - Code for LCD Practice Test



```

int redledpin = 6; // define where red led pin is placed
int greenledpin = 7; // define where green led pin is placed

void setup() {
  Serial.begin(9600); // prepare arduino for serial communication
  pinMode(redledpin, OUTPUT); // setup red and green LED as outputs
  pinMode(greenledpin, OUTPUT);

}

void loop() {

 if(millis()/1000 < 20) {
   Serial.println(", Red LED is activated"); // serial print on arduino software this wording
   digitalWrite(redledpin, HIGH); // digital write (HIGH) = turn LED on
   digitalWrite(greenledpin, LOW); // digitalWire (LOW) = turns LEd off

 }

else if(millis()/1000 > 20){
  Serial.println(", Green LED is activated");
  digitalWrite(redledpin, LOW);
  digitalWrite(greenledpin, HIGH);

```

Figure A6.2 - Code for Green/Red Light Test



```
const float ADCtomV = 4.8828125; // Conversion multipler needed from Arduino ADC value at pin to voltage in mV
const float SensorOff = 200; // units in mV taken from data sheet for MX5050DP
const float sensitiv = 90; // units in mV/Kpa taken from datasheet
float diffpressure; // The value in Kpa of the differential pressure that we need
int PIN_P1 = A0; // IF the diff pressure sensor is plugged into analog pin A0

void setup() {

  Serial.begin(9600);

}

void loop() { // NOTE: A0 in analogread is if its plugged into analog pin A0

  diffpressure = (analogRead(PIN_P1) * ADCtomV - SensorOff) / sensitiv; // result will be in Kpa since [mv * Kpa/mV] = Kpa

  Serial.print(millis());
  Serial.print(" , ");
  Serial.println (diffpressure, 2);
  delay(1000);


// can also try this code i found
//Analog_P1 = analogRead(PIN_P1);
//Vout_P1 = ((Analog_P1 * 0.00369) + 0.04);
//P_1 = (Vs * Vout_P1);
//delay(1000);

}
```

Figure A6.3 - Pressure Sensor practice code

Practice code for Volumetric flow meter used:

```
//attachInterrupt(interrupt number, the function you would like to run when triggered, what you would like to set as trigger) this is how you can interpret this function
// I am using interrupt 0 to trigger "Flow" function when a pin changes low ot high = pulse arrives from sensor
int PinFlow = 2; // This is the flow meter input pin on the arduino

double VolFlow; // this is the volumetric flow we want to find
volatile int sum; // intger is set as volatile for correct updates during interrupt = important as it will store #of pulses each second

void setup() {

  pinMode(PinFlow, INPUT); // sets the Flow pin from arduino as an input, pin needs to be set as an input before it can interrupt
  attachInterrupt(0, Flow, RISING); // makes the interrupt 0 (pin #2 on arduino mega) to run the function "Flow" so the increment function
  Serial.begin(9600);

}




void loop() {

  sum = 0; // resets the counter so we start counting from 0 again since loops runs over and over again
  interrupts(); // allows interups on the arduino to run, so start counting # of pules from sensor
  delay(1000); // delay reading for 1 second
  noInterrupts(); // Disable the interrupts on the arduino so stop counting pulses

  // Now we must do the math to find the volumetric flow since we know approx 2.25mL fluid/pulse (everytine it rotates)
  VolFlow = (sum*2.25); // #pulse in a second by the amount of 2.25mL of fluid
  VolFlow = VolFlow / 1000; // convert to L/s
  VolFlow = VolFlow / 1000; //convert L/s to m^3/s

  Serial.println(VolFlow);
}

void Flow() { // new function created that the interrupt will run every time a pulse is found

  sum++; // everytime this function is called an increment, ++ will ad 1 to that variable everytime the program runs

}
```