Deliverable F - Prototype I and Customer Feedback

GNG 1103
Introduction to Engineering Design

To:
Professor David Knox

By: Group 03 - AllTheyDoIsWin
Alec Plourde
Ashley Garofalo
Gabrielle Graceffa
Thomas Johnson Band
Emmett van den Broek

Due Date: November 6th, 2022
Submission Date: November 6th, 2022

University of Ottawa
Faculty of Engineering

Table of Content

Wrike Link: **https://www.wrike.com/open.htm?id=969517648**

1.0 - Feedback from Client Meeting

The client meeting with Shane from Beyond the Pale went very well. We presented our idea for the in-line system based on our three sub-categories. Shane gave us very good feedback regarding a few key points in our design. Starting with the method to read the specific gravity, Shane loved the idea and believes this method will work in measuring the specific gravity value based on basic Bernoulli concepts/principles. In regards to the display method, Shane explained that this device will be a few feet high off the ground (as the recommended pipe placement is at a lower point). This information has allowed us to place the LCD screen in a horizontal orientation so the users can easily see the result displayed. In addition, the feedback presented has allowed us to modify our method to display if the set range of Plato values is passed. With Shanes's feedback, he explained that the red and green light system would be more beneficial when compared to the buzzer since the noise can be quite loud in the brewery. With the light idea in mind, Shane provided us with the ideal range of 0-30 Plato which will be used to identify the range for the green light to be turned on when compared with the red light. Finally, Shane has reassured many of the basic fluid dynamic assumptions made in order for this design to be used. The assumptions of no work applied, incompressible fluid, and constant temperature is all valid in order to use the orifice flow meter system. For future design requirements and decisions, the feedback received will be crucial in order to make correct modifications. All feedback received will be used to create a design based on the user's requirements.

2.0 - Analysis of Critical Components

Measuring the Specific Gravity Value (recap from the previous deliverable):

Concept: This subsystem will calculate the specific gravity results based on readings from various sensors installed into the pipe device. The sensors utilized in this design will consist of one orifice plate, one volumetric flow meter, and a differential pressure sensor. In obtaining the volumetric flow (q) of the wort flowing through the system and the differential pressure (ΔP) we can theoretically find the density which would lead us to the specific gravity. The theory is represented below. We can use Bernoulli's law (1) and the continuity law (2) as a basis of fluid dynamics in order to understand the complexity of solving for density (ρ) [1]:

Bernoulli's Equation (1):

$$P1 + \frac{1}{2}\rho v1^2 + \rho g h1 * = P2 + \frac{1}{2}\rho v2^2 + \rho g h2 *, \ where\ P = pressure,\ h = height, v = flow\ speed$$

** Note: we can eliminate the potential energy terms as there*

We can then add an orifice plate in the middle of our system in order to create a temporary change in pressure in the straight pipe which would allow us to find *is no difference in height in a straight pipe*

Continuity Equation (2):

$$q = v1A1 = v2A2,\ where\ q = flow\ rate\ [m^3/s],\ A = cross\ sectional\ area\ of\ pipe,\ v = flow\ speed$$

Combining these equations under ideal conditions while adding a discharge coefficient (Cd for the orifice opening and function of jet size) we can thus create this relation:

$q = C_d * A2 * [\frac{2*\Delta P}{\rho(1-(\frac{A2}{A1})^2)}]^{1/2}$ the density of the liquid. The orifice plate has a diameter of D2, while the pipe diameter can be expressed as D1. With these 2 diameter values, we can combine them into this simple ratio: d = D2/D1. The ***final equation*** then becomes:

$q = C_d * (\frac{\pi}{4})D2^2 * [\frac{2\Delta P}{\rho(1-d^4)}]^{1/2}$, where the variable we are looking for is density (ρ) of the liquid

Once we find the density of the liquid flowing through the system we can use this simple equation to find the Specific Gravity (SG)

$$SG = \frac{\rho_{liquid}}{\rho_{reference}},\ where\ \rho_{reference} = water,\ which\ has\ a\ typical\ value\ of\ 997\ kg/m^3$$

We would then modify the units in order to get the required units of °Plato as requested by beyond the Pale. The sensors used would find the q value (using the volumetric flow meter) and the differential pressure sensor will find the ΔP value represented in the equation. Knowing these values while the d,D, Cd are constants, we can thus code the Arduino mega to solve for the density value which would give us our SG.

Once we have a specific Gravity Value we can convert this into °Plato with this equation:

*Plato = (-1 * 616.868) + (1111.14 * sg) – (630.272 * sg^2) + (135.997 * sg^3)*

The Required sensors needed for Bernoulli equation

1. The volumetric flow meter:

The flow meter uses a rotary system, sending a signal at approximately every 2.25mL of the liquid as it flows through it. Knowing this, all we have to do is multiply the number of pulses in a second by the 2.25mL which will give us results in mL/s. We can then divide this by 10^6 to turn the units into the required values of m^3/s (the q value required in our final equation).

2. The Differential Pressure Sensor:

The differential pressure sensor will be used to obtain the ΔP value in our required equation. The sensor is connected by two plastic tubes and inserted into the premade pressure taps to which it will read the difference in pressure. Now in order to get the most accurate results possible, the taps and tube connections must be sealed as best as possible to avoid any unnecessary pressure losses. In addition, research was completed on how to correctly code this complex sensor to get the best results. Upon research, it was discovered that obtaining the best pressure results in the sensor offset and sensitivity value will be introduced which will give us a value in KPa. Then we only need to convert this value to Pa by multiplying x1000 (see the practice code).

3.0 - Prototyping Test Plan

For Test ID#2: Testing LCD screen

Goal: The goal of this test was to ensure the LCD screen was in working condition and could print a required message to the screen

Process: 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 Appendix Figure A1 for the code used in this step)

Results: 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.

Photos:

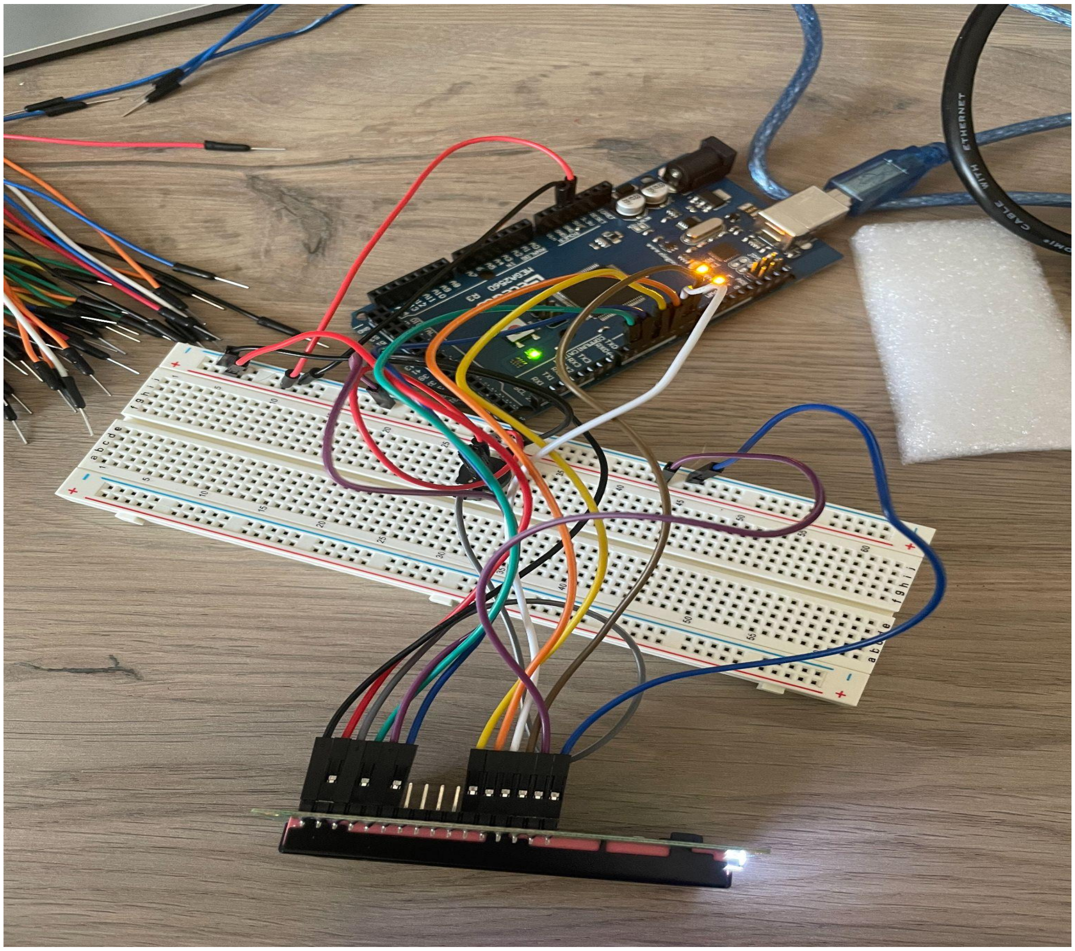


Figure 1.0 - Wiring for LCD Screen Test

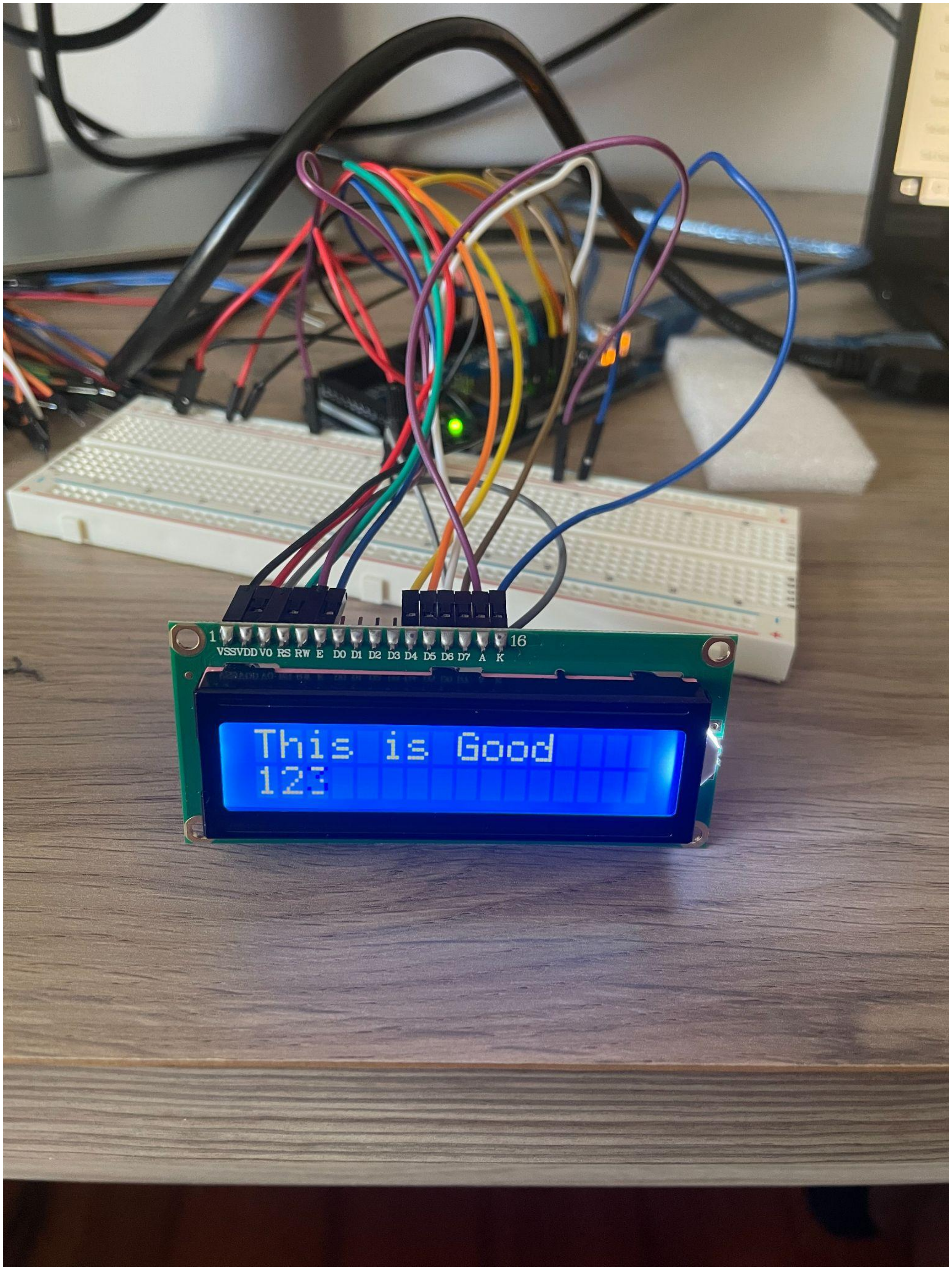


Figure 1.1 - LCD screen printing correct message for test 2

Test ID #4: Testing LED lights

Goal: The goal of this test was to make sure the LED lights can turn on based on a required if/else statement and a set of values

Process: 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 Appendix Figure A.2 for the code used in this step)

Results: 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.

Photos:

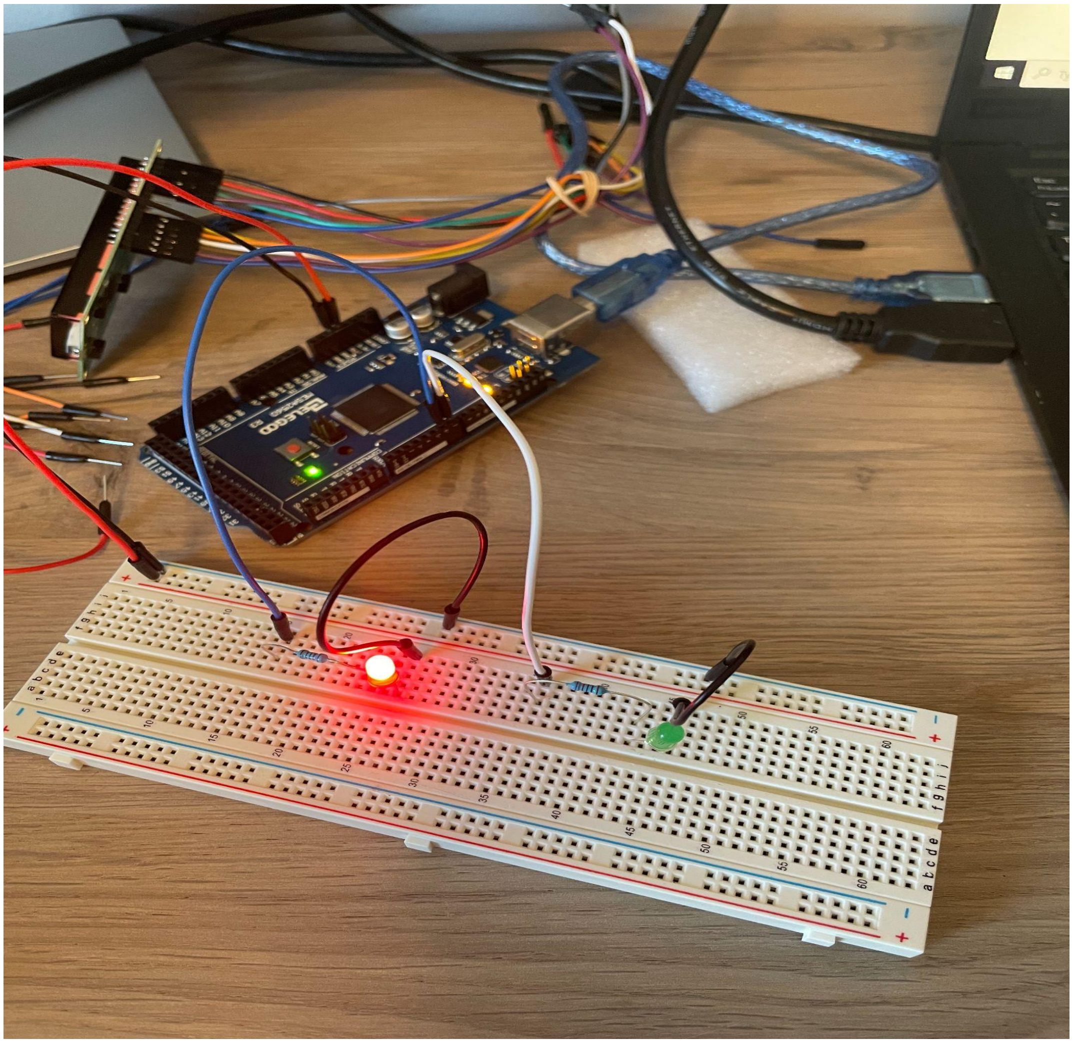


Figure 2.1: Red LED activated during trial test #4

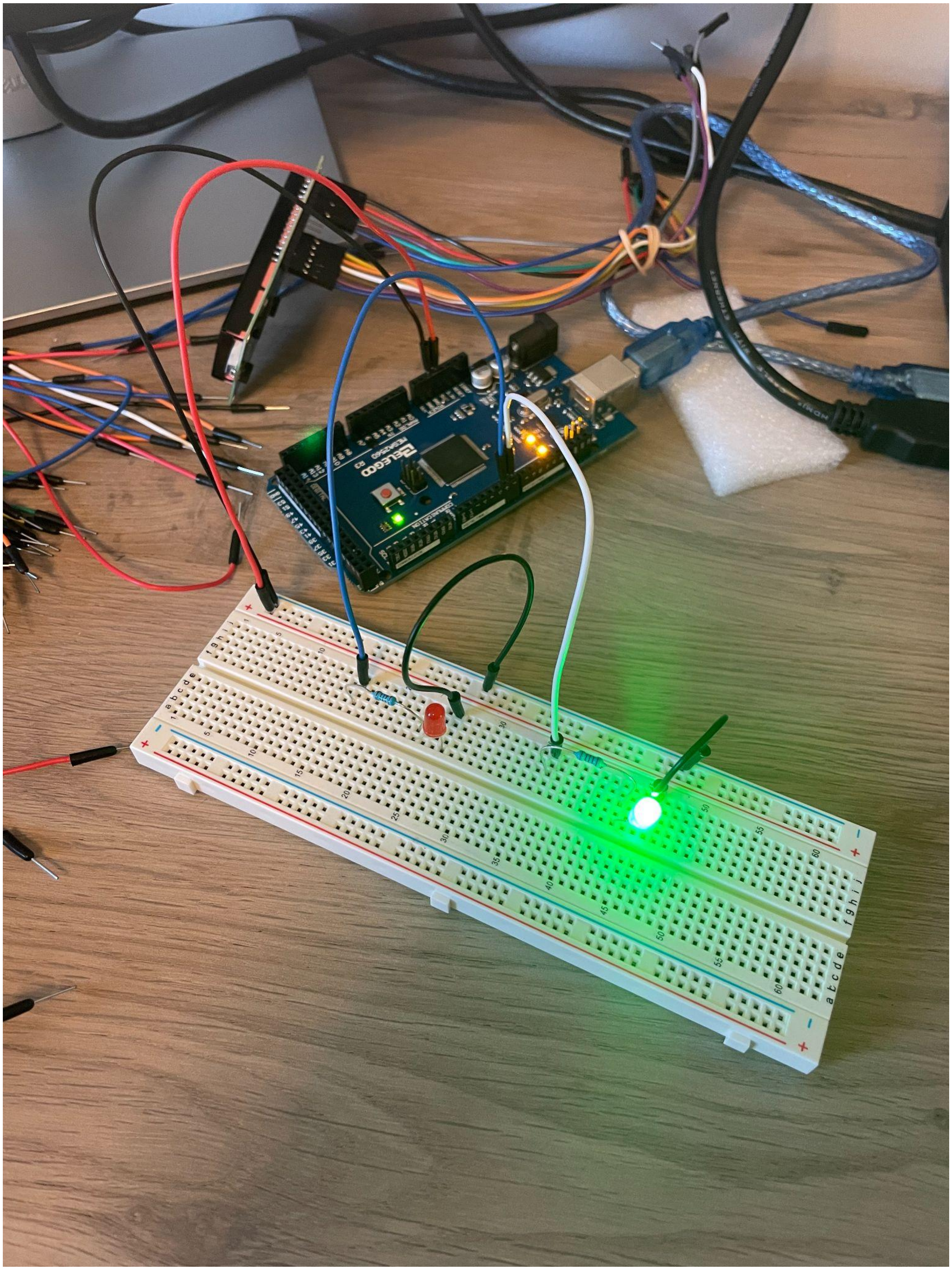


Figure 2.2: Green LED activated during trial test #4

Test ID #5: Testing Differential Pressure sensor

Goal: The goal of 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)

Process: 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 3.1 demonstrates the layout of this prototype

(See Appendix Figure A.3 for the code used in this step)

Results: The differential sensor was able to read the difference in pressure under both conditions. Figure 3.2 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 3.3, the results for the trial with air blown into the positive pressure tube were concluded and successfully increased once the blowing process started.

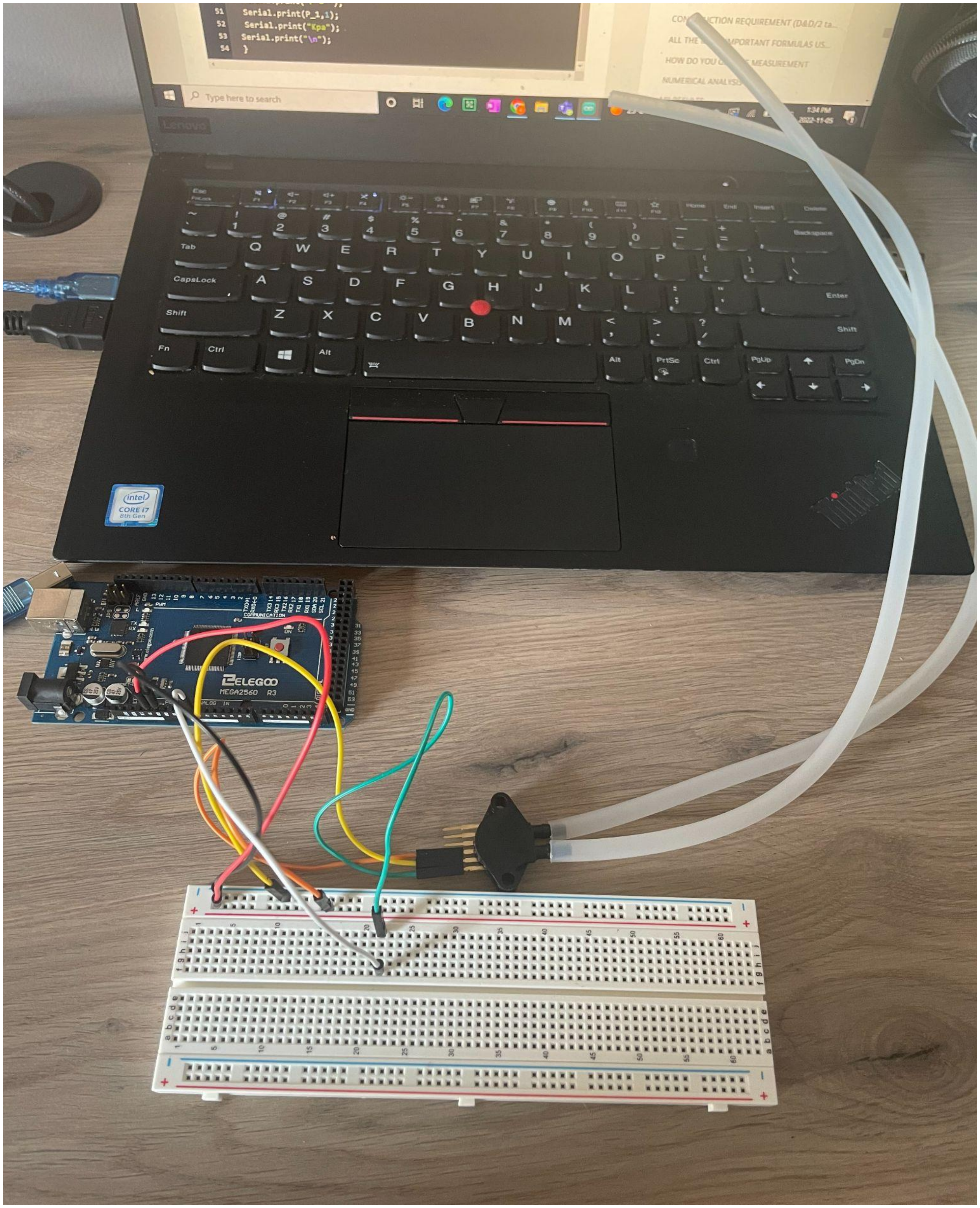
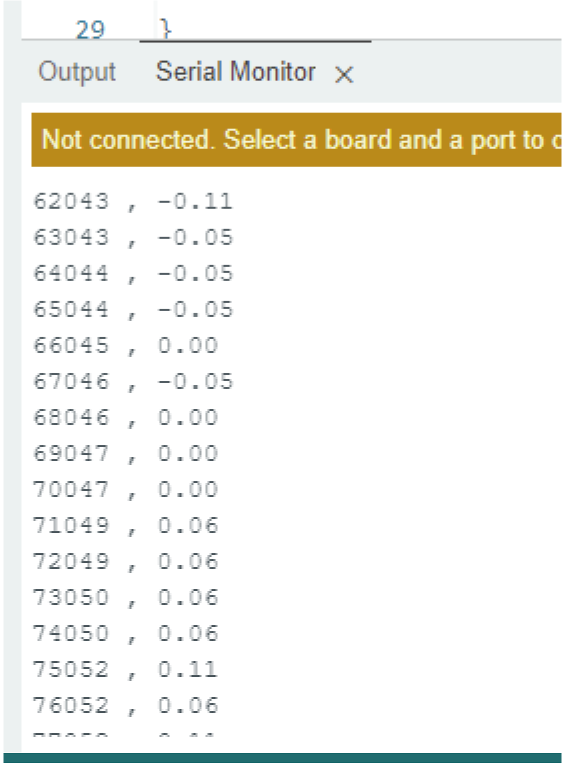


Figure 3.1: Layout of Differential Pressure Sensor Test

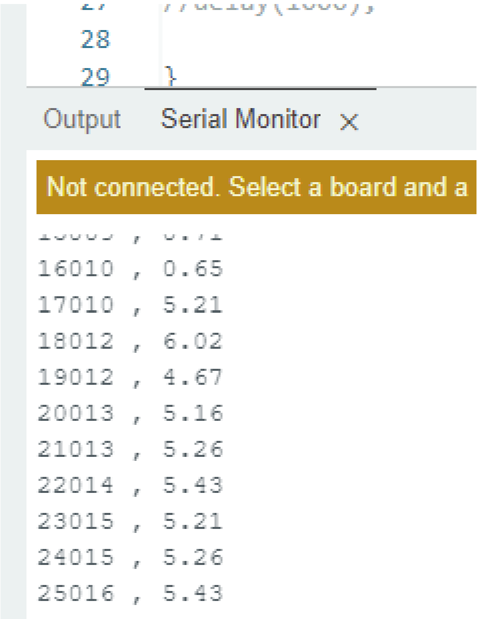


```
}
```

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 3.2: 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 3.3: Results for Trial 2 (when 1 side was blown air into it)

4.0 - Revised Documentation

After discussion with the TAs and review of materials at home, the new BOM can be seen below:

# Parts	**Device Name**	**Cost $CAD**	**Explanation**	**Source**
1	Arduino Mega 2560 Rev3	0	Alec has one; an arduino mega is needed because there are a lot of sensors that need to be plugged in to it	Arduino
1	Pipe and Design Material (Plastic)	0	The pipe and physical design will be made with recycled plastic and 3D printed plastic parts from the Uottawa printers	N/A
4	Filters	9	To capture all the yeast	Filter
1	Breadboard	0	Alec has one: in order to connect all required sensors and devices to the arduino	N/A
2	LED light (green and red)	0	Alec has one; to demonstrate the specific gravity and to blink the correct color that goes with the specific gravity	N/A
3	Resistors (2x10kOhm) and 1(220ohm)	0	Alec has them; they will be used to correctly wire the LED lights along with the volumetric flow meter	N/A
1	Orifice plate	0	The protoype is 0.5” ID in diameter; therefore, the orifice plate must be smaller. The plate is designed by SolidWorks and produced by a 3D printer in the maker's lab offered free for UOttawa students	N/A
1	Differential sensor plate	65	To read the difference in pressure between the 2 pressure points (one before and one after the orifice plate)	Differential Pressure Sensor
1	Tubes	0	To transfer the pressure readings from the taps to the differential pressure sensor device, tubes were found at Alec’s home thus avoiding the cost of purchasing them	Tubes
2	Pneumatic Hose Fitting	0	Alec has some at home that can be used to connect the tubes into the piping device based on the ¼” OD of the plastic tube	Tube 3/16” fitting
1	Volumetric Flow Meter	26	To calculate the amount of liquid that is passing through the straight pipe (in mL/s) to give us the value “q” in our main equation to find the specific gravity	Flow Meter
Total		**100$**		

The 3D model design was slightly modified since the last deliverable. The new up-to-date assembly can be seen below. A few changes that were made were a slight increase in the orifice plate and connecting devices to which the orifice plate and encompassing plates are 0.2” thick. This should add a bit more structural integrity when faced with water flows. In addition, the thread direction was added to the wrong part in the last deliverable (part 2 had the threading where part 3 was the correct one to have it) thus the change to the correct part was made. These changes now bring our total length to 10.1” which is respectively under the 12” limit discussed with Shane. In addition, pins were also designed to be printed and inserted into the 4 premade slots in order to connect the orifice plate to the other pieces.

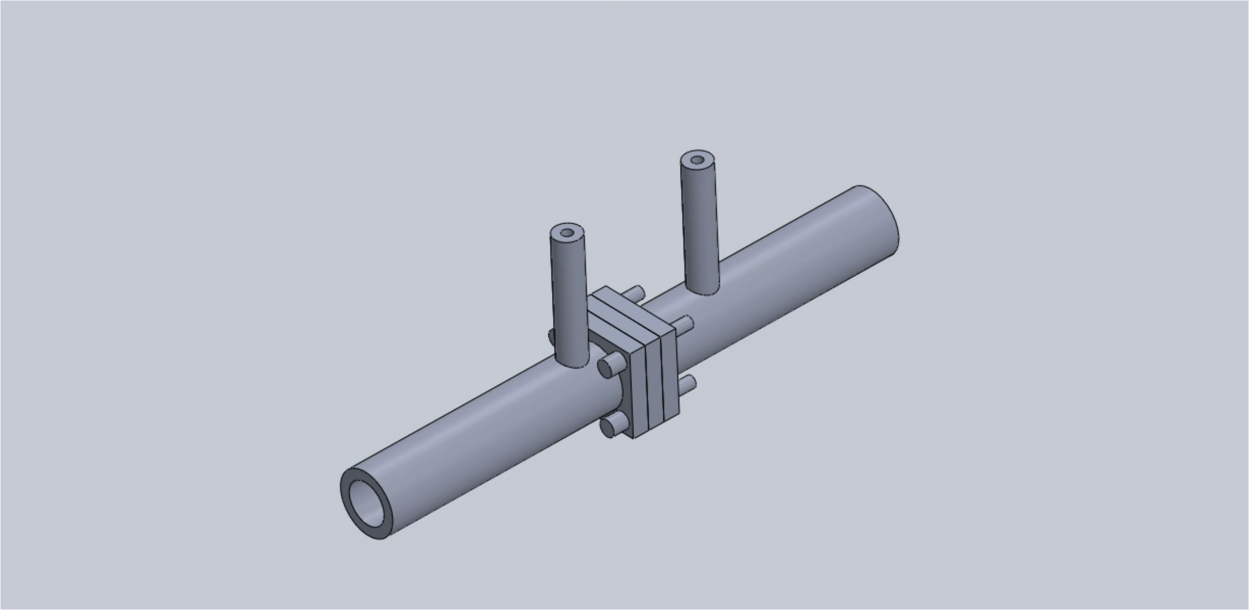


Figure 4.1: Revised Assembly Model of Project

5.0 - Revised Prototyping Test Plan

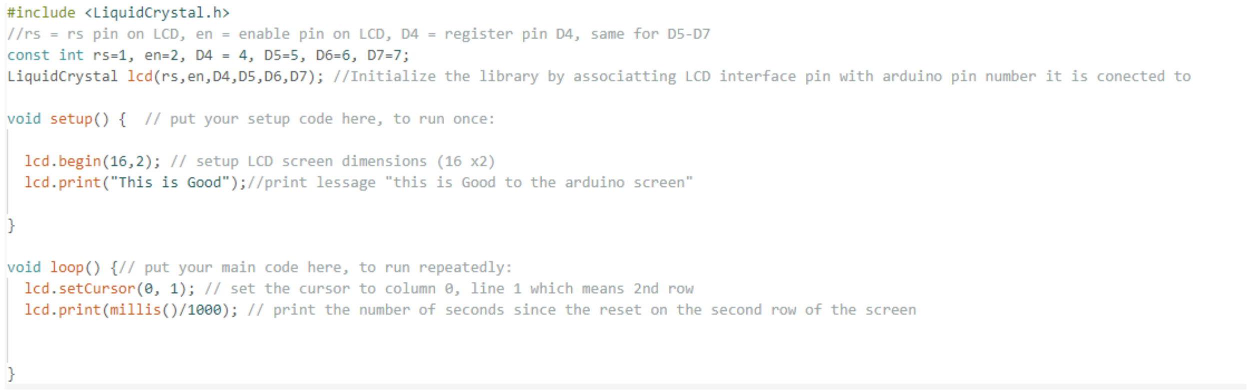
Table 1.0 - Prototyping Test Plan

Test ID	What is it	What it tests (Objective)	Results Recorded	Test Duration & Start Date
3**	Testing the volumetric flow meter and its code	We need to ensure the volumetric flow meter can correctly measure the flow of liquid per second	Measurements of “ml/s or L/s” is read and processed by the Arduino	Test duration (2-3 days) once material arrives (shipping delays have occurred)
6	Print 3D Model of Design	In order to test the remaining systems, a physical model must be 3D printed and built. This test will ensure the physical model can be achieved based on theoretical design.	The structural integrity of the design (is it waterproof?)	Testing duration is estimated at (3-4 days) Starting on Nov 7th
7	Test the orifice plate.	This test will allow us to ensure the orifice plate is resistant to potential flows (structurally intact) and to ensure the edges are straight (so our assumption of Cd= 0.6 is valid)	Correct sealing of the plate and pressure sensors in prototype 2 design	Testing will take 3-4 days (starting November 8th)
8	Test the display and calculation of our data.	Once we are sure the specific gravity can be measured using our design, we have to display this data. This prototype will test our coding for the calculation of specific gravity, displaying the value on the LED screen.	Correct Specific gravity value is shown on the screen (in Plato) prototype 2	Testing will be (3-4 days) starting on November 5th)
9	Ensure the prototype is correctly built	Check if the physical design of the built is fully functional with the electronic layout prebuilt	No water leaks of any kind, and everything fits	Test will take 2 days (starting on November 10th)
10	Add all final touches and design specifics into the final prototype	After we have ensured that we can measure specific gravity and have a way of displaying it, we can put the two together into one last prototype. This will allow us to see everything working altogether and will be close to our final design.	All parts are connected and work under normal operating conditions	This test will last 4 days (beginning on November 8th)

**Note: Unfortunately due to shipping delays, the volumetric flow meter will only arrive in time the next deliverable, thus this test will be moved to prototype II.

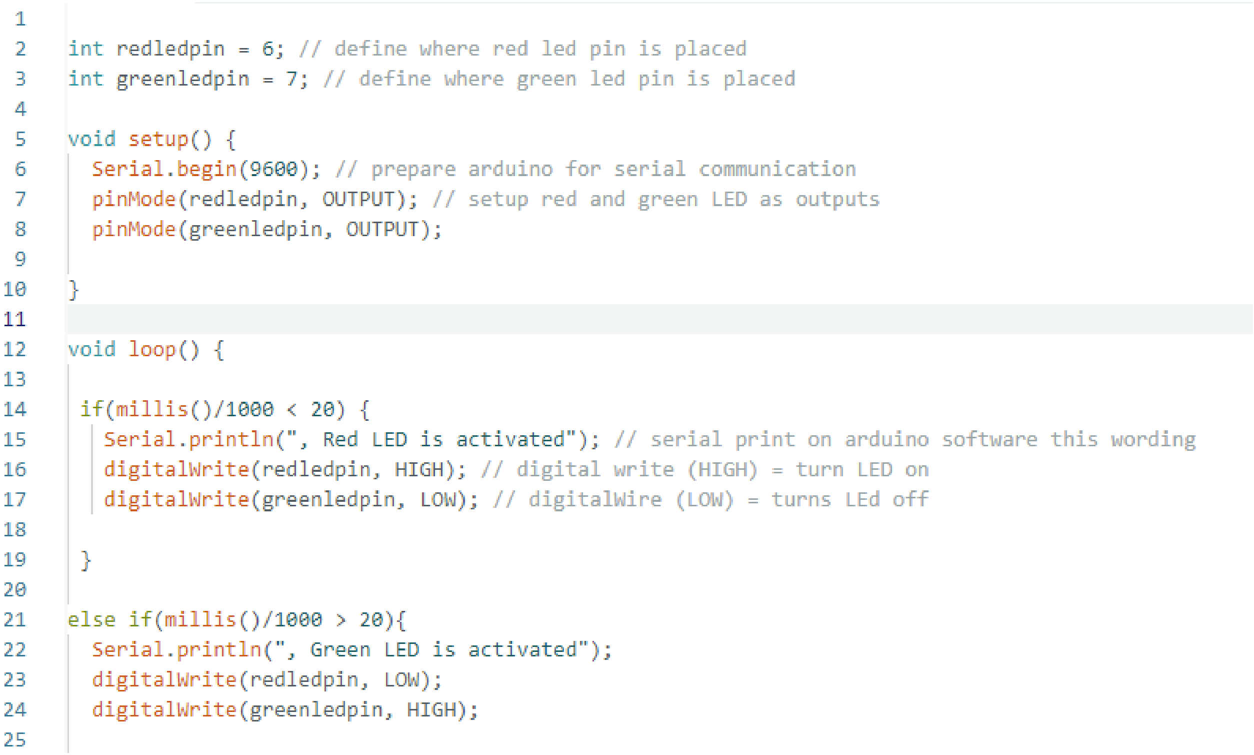
6.0 - Appendix

The following figures in the appendix represent the code for each test ID trial:



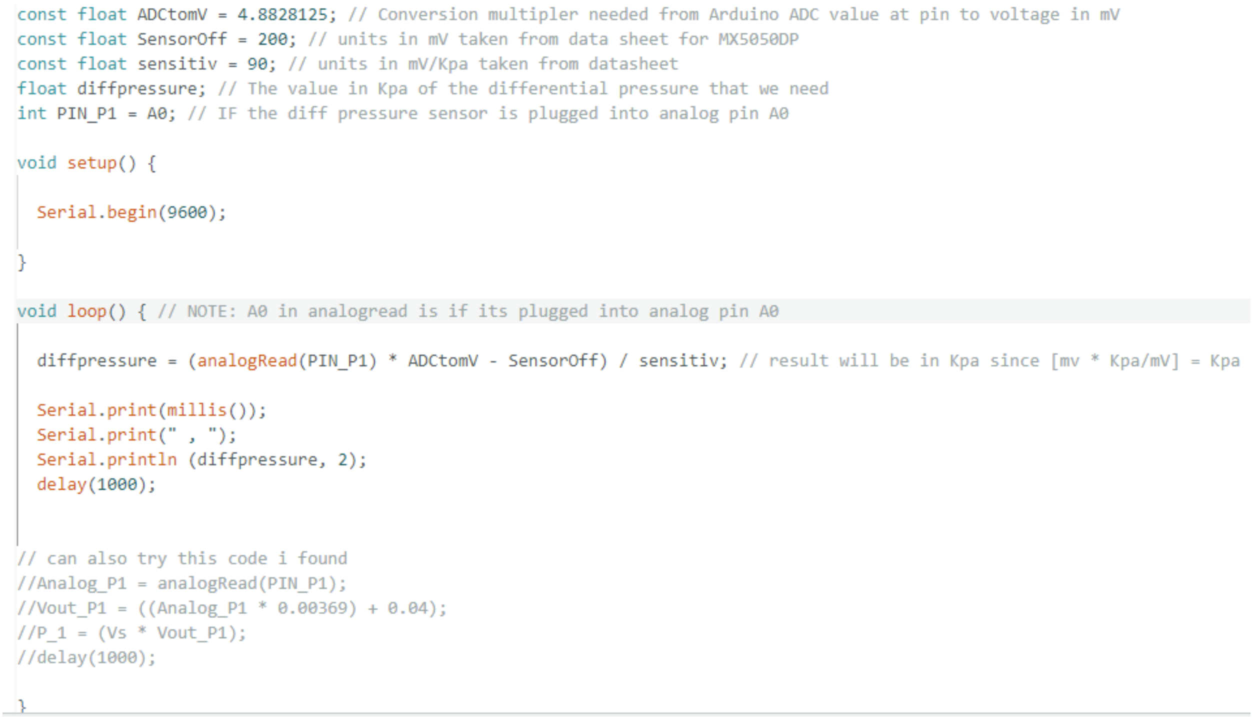
```
#include <LiquidCrystal.h>
//rs = rs pin on LCD, en = enable pin on LCD, D4 = register pin D4, same for D5-D7
const int rs=1, en=2, D4 = 4, D5=5, D6=6, D7=7;
LiquidCrystal lcd(rs,en,D4,D5,D6,D7); //Initialize the library by associatting LCD interface pin with arduino pin number it is conected to

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 A.1- LCD practice code



```

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 A.2 - LED lights practice code



```
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 A.3 - Pressure Sensor practice code