Deliverable G - Prototype II 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 13th, 2022
Submission Date: November 13th, 2022

University of Ottawa
Faculty of Engineering

Table of Content

Wrike Link: https://www.wrike.com/open.htm?id=969517652

1.0 - Feedback from Client Meeting

The client feedback from prototype 1 with Shane from Beyond the Pale went very well. We presented our prototype 1 (3D model of the design) while explaining very briefly how the concept would work. Shane’s feedback was brief but positive and a few points were brought up about sizing. He started off by reiterating that he believed this idea was great and that it should work in theory. He then answered our questions regarding the Plato value and physical layout of our device. He explained that the correct range for the Plato value would be between 0-30 which will allow us to confirm the range for our display of green or red lights. In addition, when asked how he would like the physical casing system to be situated, Shane replied by emphasizing a lot of flexibility is present with our placement (case can be hung up somewhere, can be on the ground, etc). This allows us to continue with our current casing design while implementing simple designs relating to the layout. The final portion of the feedback was related to our design specifications. Shane asked about the sizing of the pipe, the orifice plate size, and the overall length of the device. Our group explained the pipe size would be ½” in pipe diameter, while the orifice plate diameter would be ⅙” with a total device length of approximately 11”.

2.0 - Prototyping Test Plan Part 2

Test ID #6: Print 3D model of device

Goal: The goal of this test was to ensure the design concept could physically be built in a cost-effective manner using the 3D printers at home.

Process: 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 2.1 below).

Results: 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 4.0)

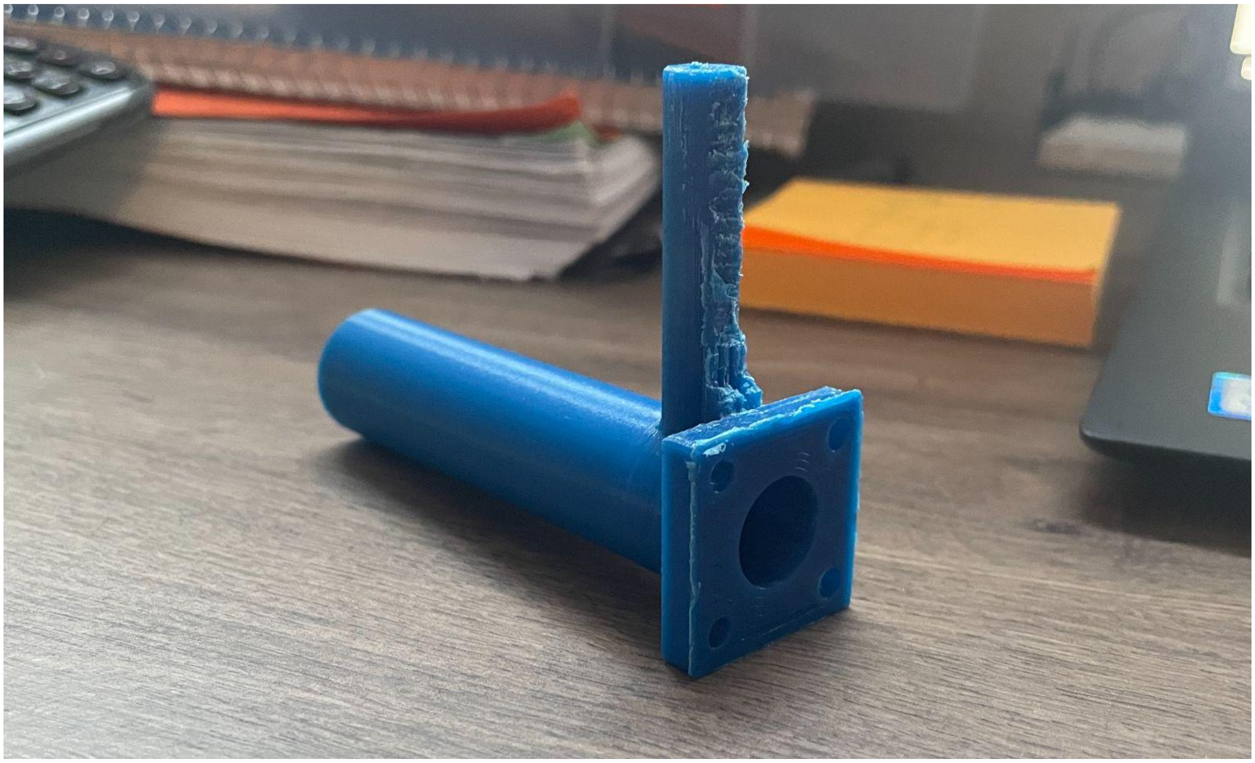

Figure 2.1: Part 2 creation in 3D print

Test ID #7: Testing the Orifice Plate

Goal: 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.

Process: The test was carried out on November 10th and occurred at the Makerspace.

Results: The orifice plate was successfully printed and is stable enough to withstand the forces required.

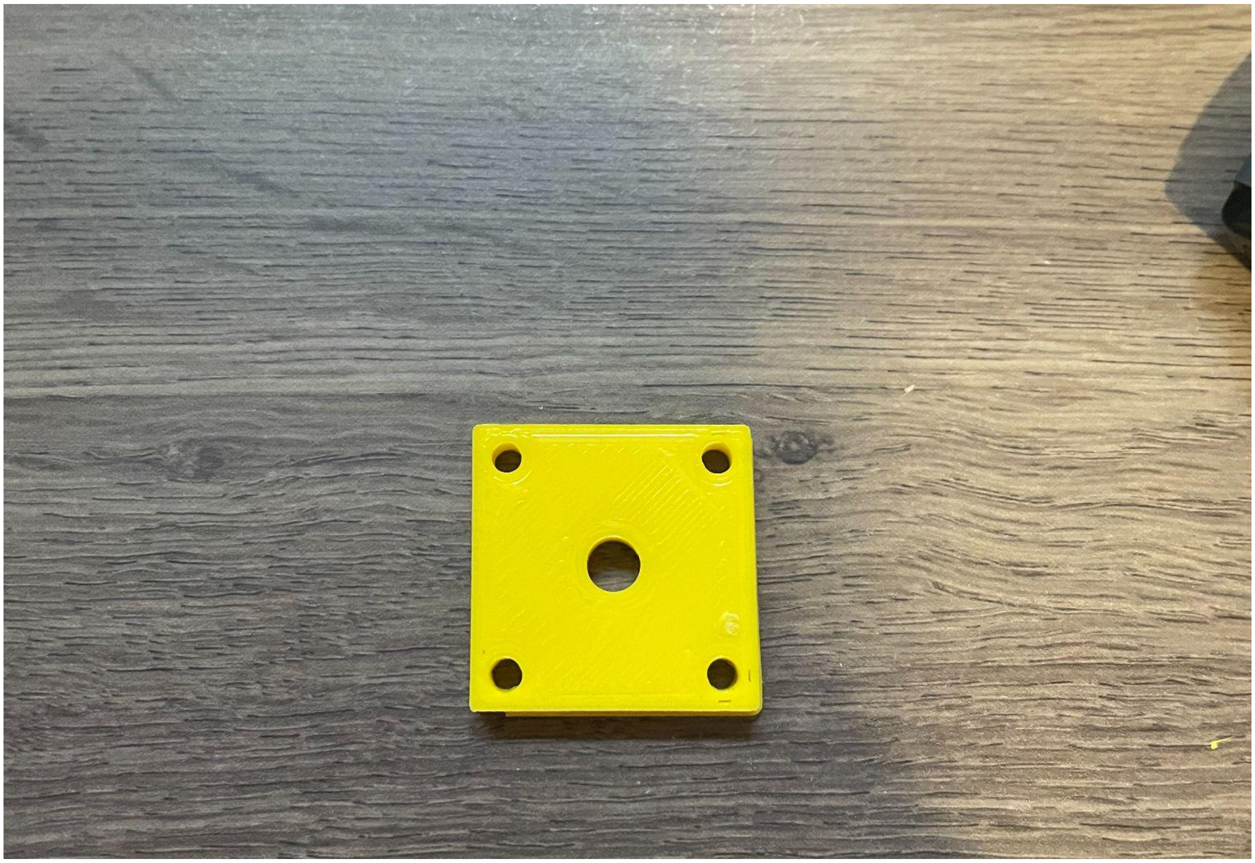

Figure 2.2: Orifice plate successfully 3D printed

Test: Wiring of all components

Goal: The goal of this test was to ensure all components can be properly wired and connected to the Arduino and breadboard.

Process: 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.

Results: All components were successfully connected to the breadboard and the Arduino was able to turn on without issues. The results can be seen below in figure 2.3

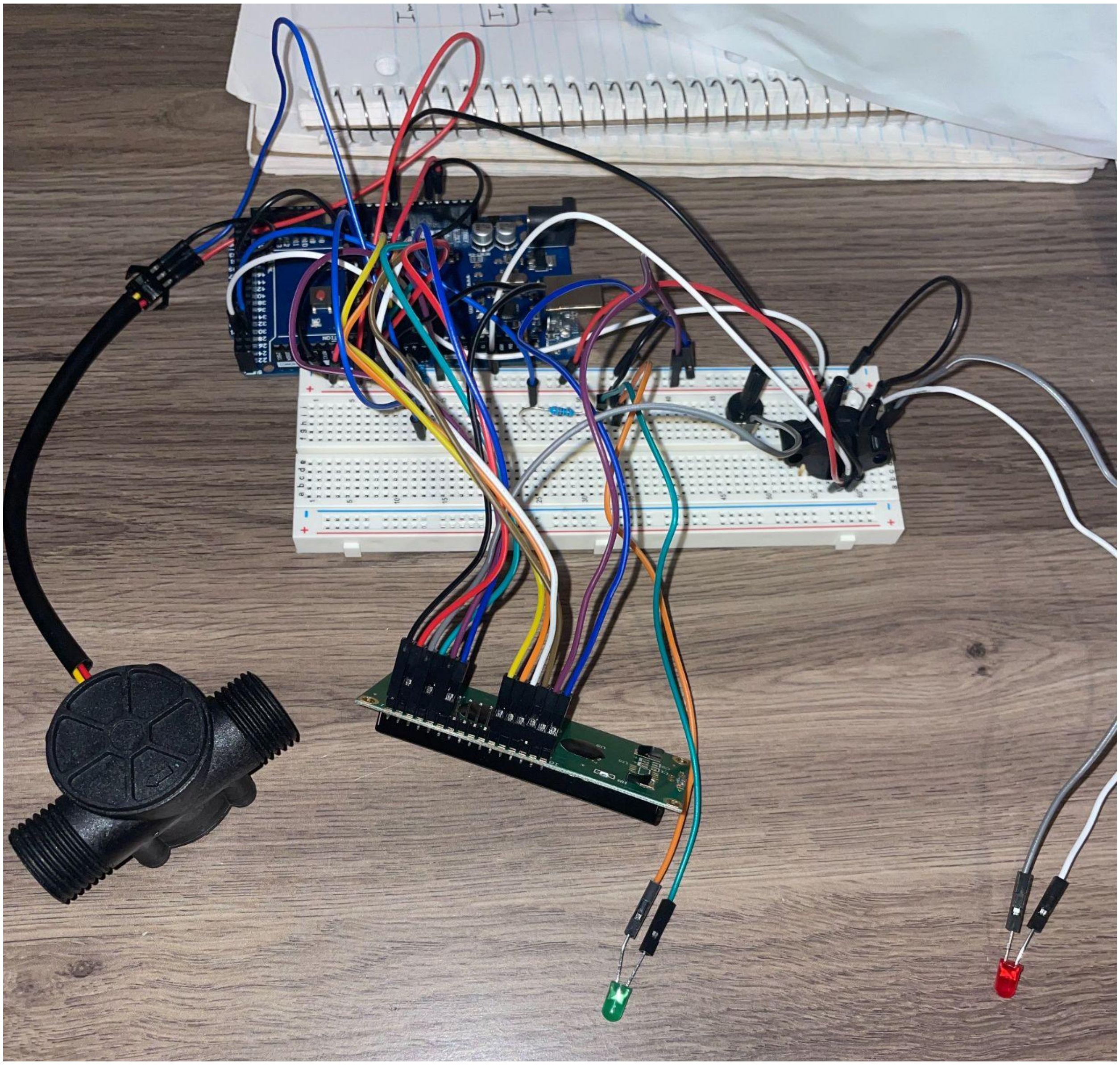

Figure 2.3: Wiring of all components

Final Code:

The final code was completed for this design. This will be the code used to test the complete setup of the device once it is built. The code can be seen below:

```
#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

}
```

3.0 - Analysis of Prototype

Using Solidworks, we are able to perform a few analysis for our design concept, assuming the material built would be AISI 1035 Steel (SS).

1. Temperature analysis Test to see the most susceptible areas of high-temperature

Assumptions:

- Convection coefficient: 100
- Temperature exposed to: 70C

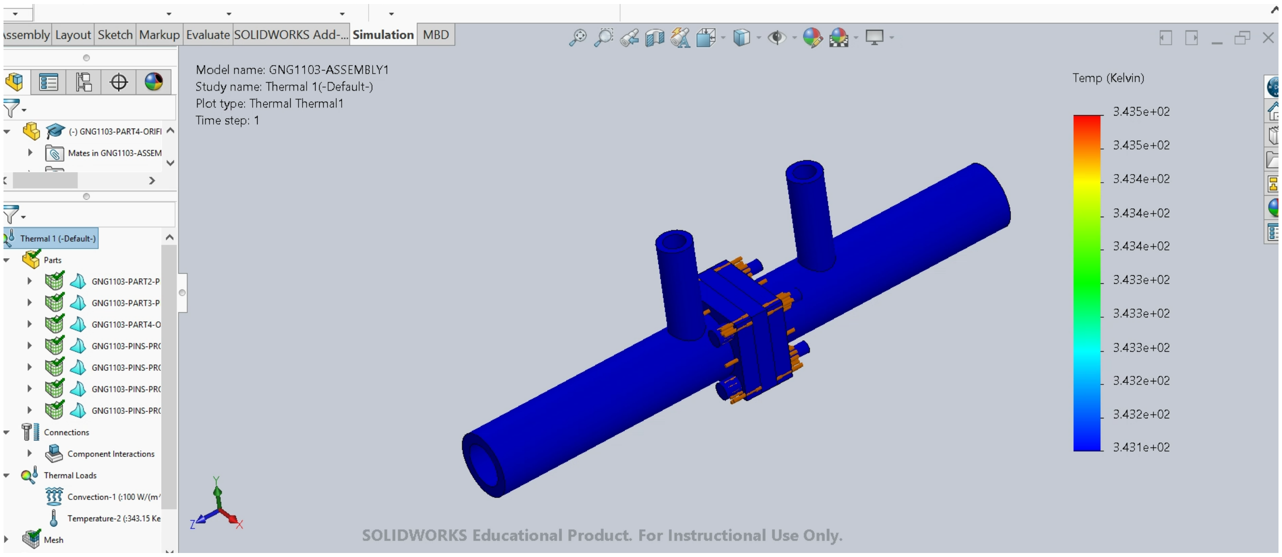
Model name: GNG1103-ASSEMBLY1
Study name: Thermal 1(-Default-)
Plot type: Thermal Thermal1
Time step: 1

Figure 3.1 - Temperature Analysis of Critical Points

A basic Solidworks simulation analysis was completed based on a few engineering assumptions to which the results were as expected. The point to which the temperature would be most critical would be at both surfaces of the plates holding the orifice plate. This information now allows us to estimate and understand the temperature distribution in those critical edges and areas.

4.0 - Revised Documentation

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 4.1 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 4.2 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. The final assembly (omitting part 1 which will be attached to the other side of the volumetric flow sensor) can be seen in figure 4.3.

Finally, the casing device for all electronics was modified to minimize the 3D printing time (estimated at 1D and 12 hours). The new case design can be seen in figure 4.4 in which a slot was added for the LCD screen along with 2 small circular slots will be added for the red and green LED lights (to improve visibility when they are on).

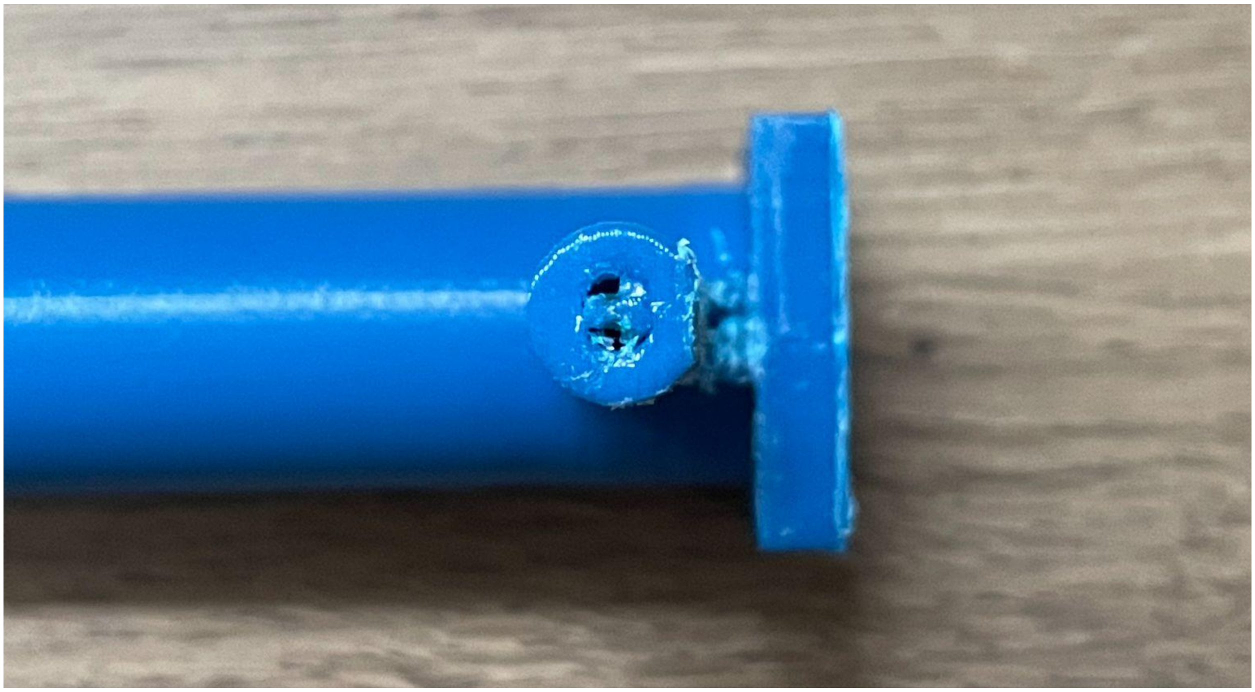

Figure 4.1: 3D printing error for pressure tap of part 2

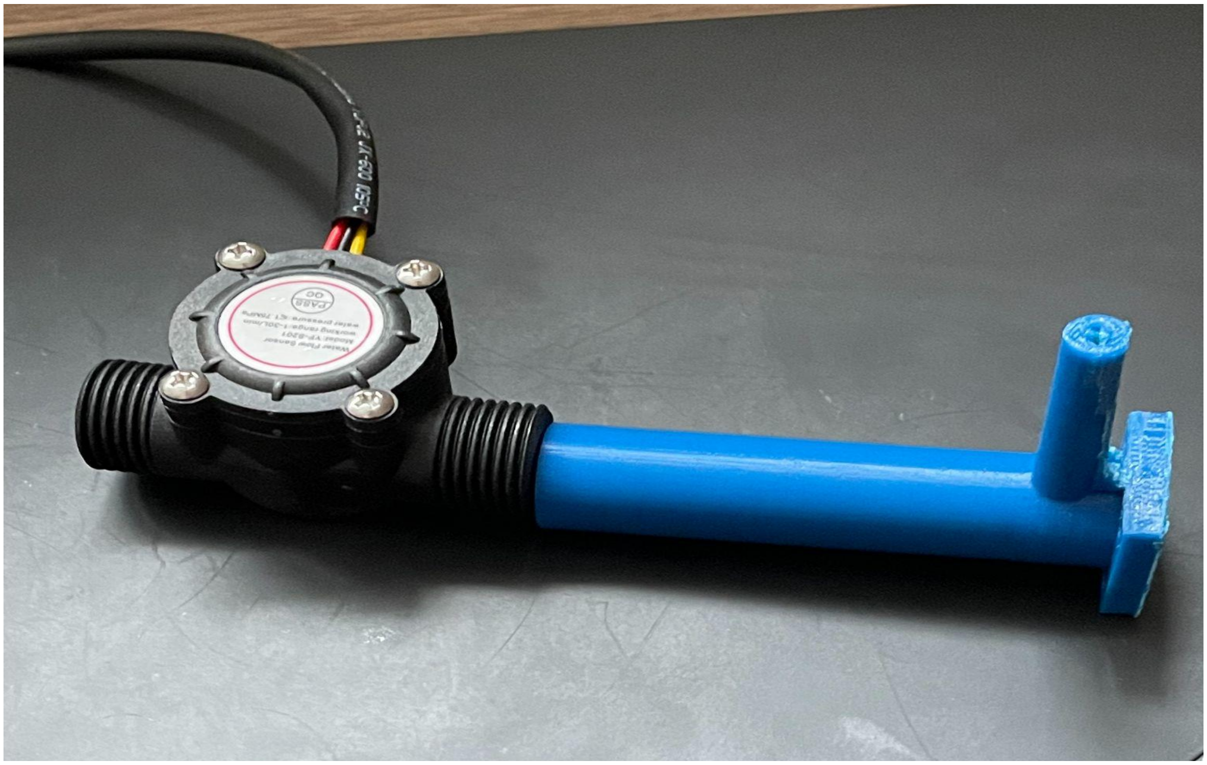

Figure 4.2: Pipe Diameter sizing error

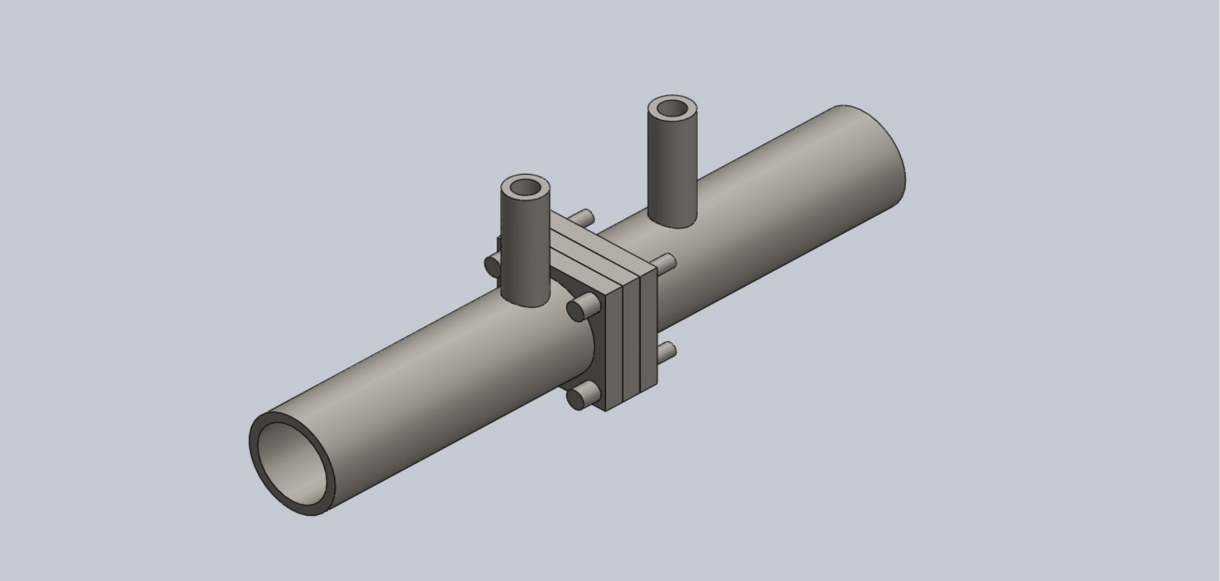

Figure 4.3: Revised Assembly Model of Project

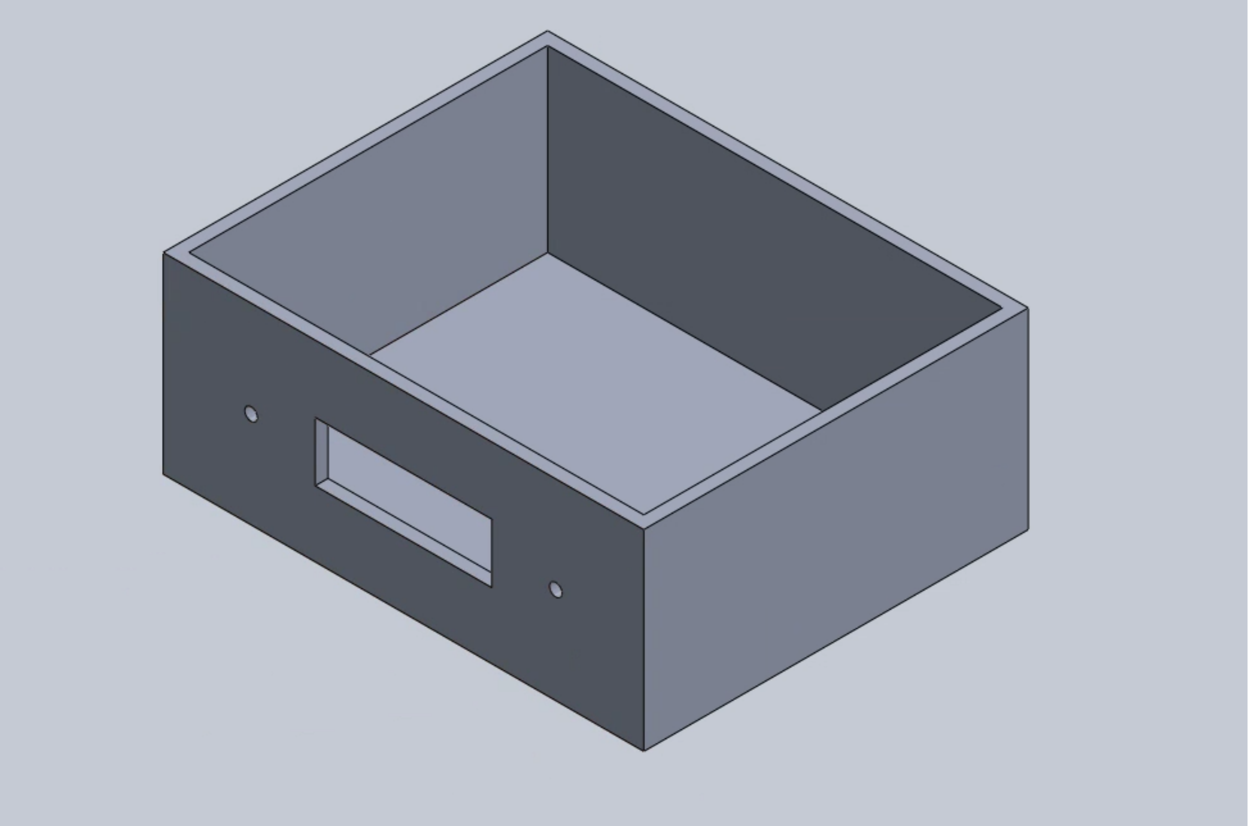

Figure 4.4 - New Case Layout for Project

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)
8	Finish Print 3D Model of Design	Due to design errors, the model must be reprinted to ensure it will fit with the volumetric flow sensor.	The structural integrity of the design (is it waterproof?)	Testing duration is estimated at (3-4 days) Starting on Nov 14th
9	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 17th)
10	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)
11	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)