Deliverable H

Prototype 3 and Client Feedback

GNG 1103
Design

To:
Professor David Knox
And
Pankaj Kumar Rathi/Chetan Kumbhar

By:
Jack Bridgeland
Wei Chen
Nicolas O'Brien
Balpreet Singh
Johann Wehrstedt

Team: Eternal Hoptimists
Submission Date: November 29th, 2022

University of Ottawa
Faculty of Engineering

Table of Contents

Introduction

In this report, our team will outline all steps we took to accomplish our final prototype. We will discuss the printing of the electronic housing, the manufacturing of the sensor housing, the final wire diagram will be produced, as well as the code used to finally run the sensors. We will also discuss any changes related to our design, we will go into detail on how the product works and what feature are included as well as which ones were removed.

Prototyping

3D Printing and Manufacturing

Our team has printed housing units for our screen, electronics, and wiring. They were 3D printed on November 21st in the Maker Space Solidworks provided by the University of Ottawa as seen in figure 1. The casing was redesigned since previous models only accounted for the exact measurement of the screen and remote, which has been a problem since without extra space both can fit the holes/holders they were meant to be in. This housing unit can hold all the important electronics such as the screen, wires, clock module, IR receiver and Arduino with all the wires that come with that as seen in figure 2.

The housing unit for our pressure sensors are made by Saxon Precision Incorporated as seen in figure 3. The length between our sensors is ~12.5 inches and is made of stainless steel. The pressure sensors will screw in on either side, with their wires being protected by making sure the unit is waterproof by using a sealant.

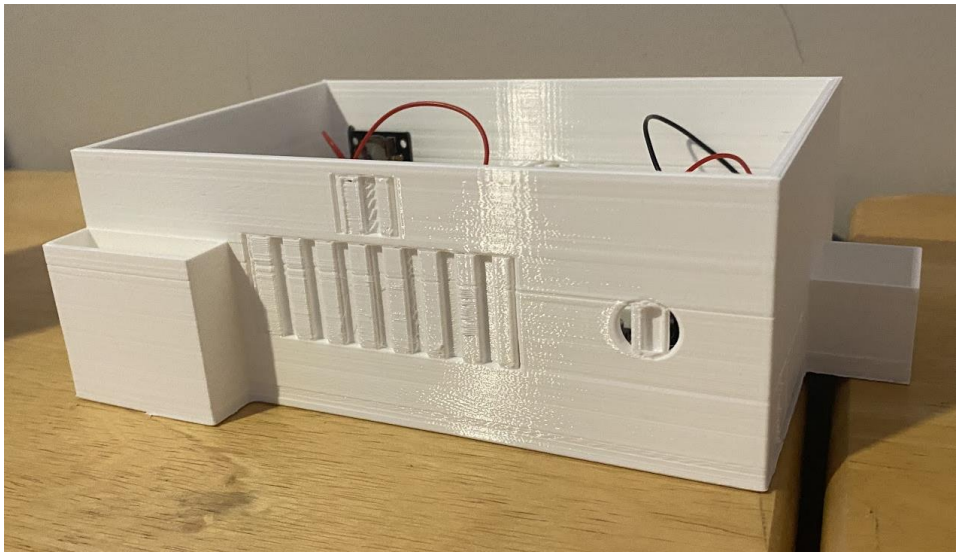

Figure 1: Solidworks Case

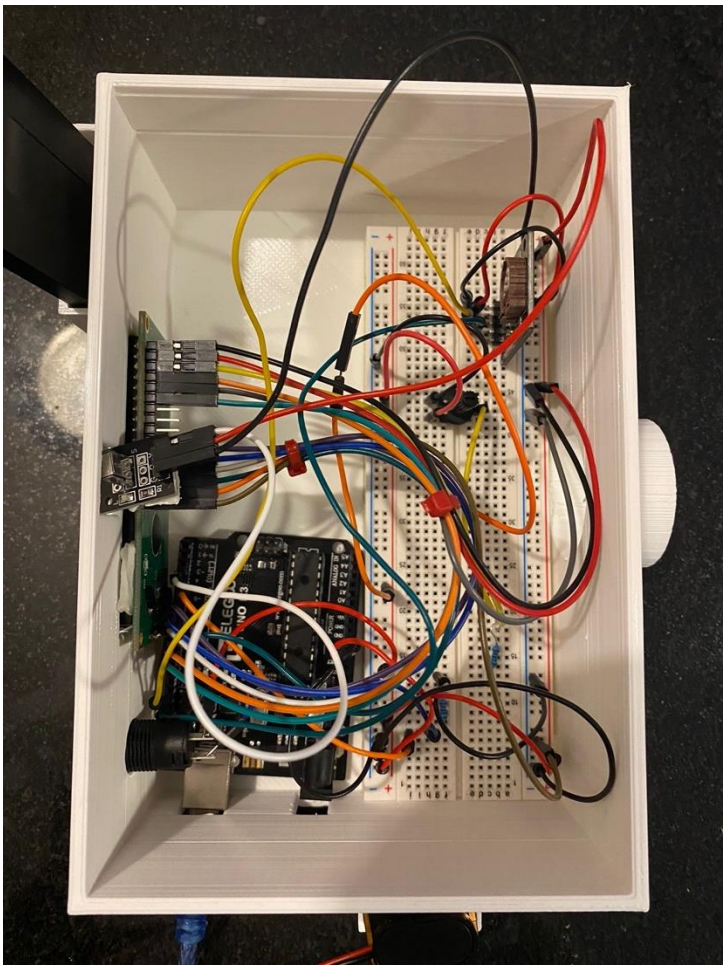

Figure 2: Wiring of Arduino in Solidworks Case

Figure 3: Pressure Sensor Housing Unit

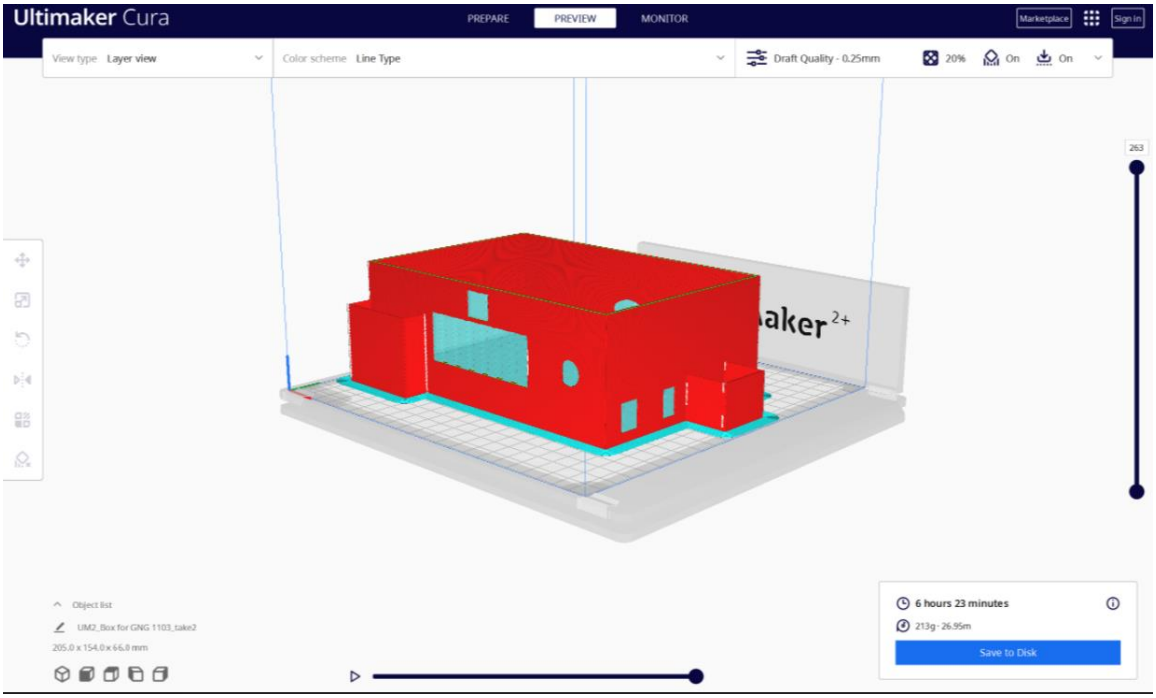

Figure 4: 3D Printing Slice Preview and Time Estimate

Changes Since Deliverable G

Since the last deliverable, we have decided to switch from Bluetooth to hardwiring our computer into the Arduino to log the data. Our team is having issues with our HC-05 Bluetooth sensor. The HC-05 Transceiver is a device which can receive and send data but during our testing, we were only able to get the device to receive data from our external device, but we could not get the HC-05 to send our data to our external device. After numerous hours of testing with this device, we concluded that we believe the pin related to transferring data is defective due to manufacturing reasons. Our data will now be logged by plugging our Arduino into our laptop and feeding the data from the serial monitor into python.

Prototype Test Plan

Table 2: Final Prototype Tests

ID	Test Objective	Description of Prototype	Results to be Recorded	Duration of Test
1	Determine accuracy of pressure sensor	Our 2 pressure sensors will be connected to an Arduino and be tested in different liquids of known density at different heights to test accuracy	1.height of completed test 2.the fluid used 3.temperature of fluid 4.difference from theoretical value	Test will last until we are within our design spec accuracy
2	Determine if the entire system is working	This prototype will consist of 2 pressure sensors, 1 temperature sensor, 1 Arduino, 1 LCD screen, 1 IR receiver, 1 button, 1 Bluetooth module, 1 real-clock module.	1.Any bugs with our code 2.Input conditions	Test until the system works as planned

Test 1: Pressures Sensor

The testing of the pressure sensors did not yield the desired results. Our value of specific gravity was found to be around 0.80, which is off by a fair margin. Our team thought of why this was, as the results in the previous tests were much better in comparison to ones we got here.

One problem could be the wires, as we unknowingly used wires with different specifications for voltage and amperes. This may cause the electricity to travel through the wires differently from one another, hence giving us an error in our signal. To solve this potential problem, removing the wires we had soldered on, and we will be soldering wires with the same type specifications to make sure this is not the issue. Another foreseeable problem is that our pressure sensors may not be accurate enough as we want. Though with our previous tests, this is the least likely. We are also seeing if the housing unit for all the wires and bread boards are not also interfering with the wires in any way in our next experiments.

Overall, we hope to improve our accuracy and figure out any other problems we may have before design day on December 1st.

Figure 5: Testing All Components

Test 2: LCD and IR System

To ensure our user interface is working exactly as planned, we wired everything together except our pressure sensors and temperature sensors to simplify our values. When the system is turned on, you will see the message “Hit Start to begin” as seen in figure 5. Once the red power button is hit on the remote control, the system will begin to measure specific gravity and temperature. On the LCD display, the specific gravity is in kg/m^3 and in Plato as seen in figure 6. To view the temperature, you hit the white button to switch the display as seen in figure 7. To reset the system to the initial state, we hit the Vol + button as seen in figure 8. While the system is running, data will be shown on the serial monitor as seen in figure 9 in the format of Date, Plato, and Temperature. This data will then be streamed into a CSV file as seen in figure 10. The stream of data was only done with the date to simplify the test.

Figure 6: System after First Being Turned On

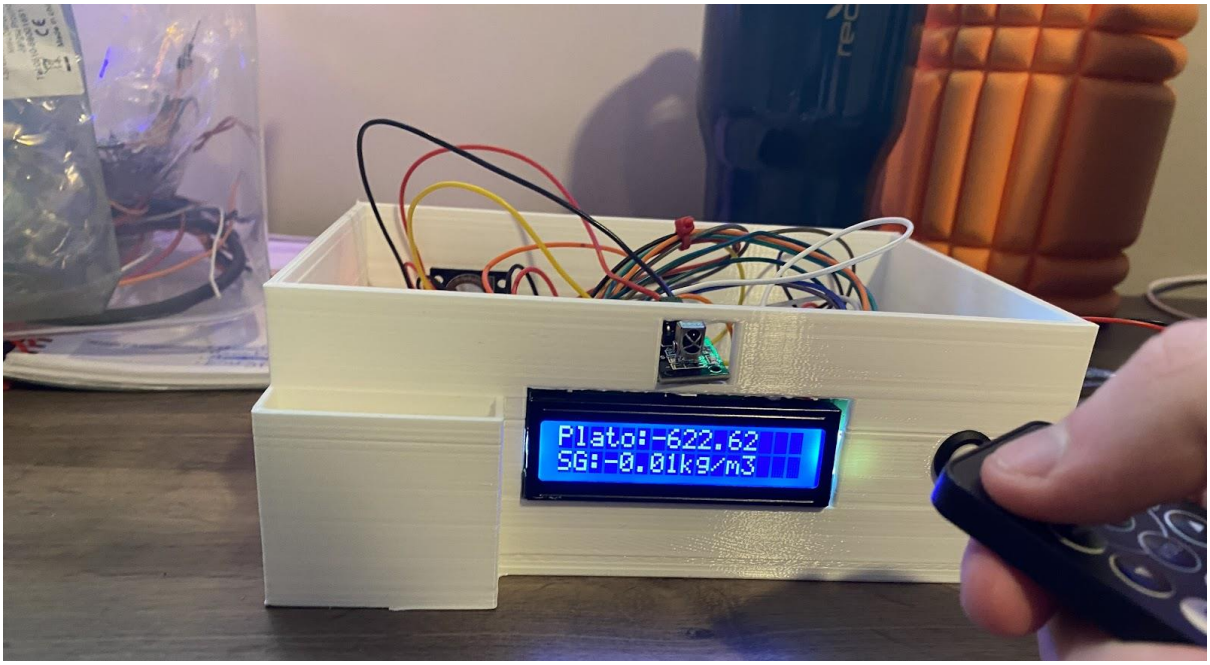

Figure 7: Display after System is started

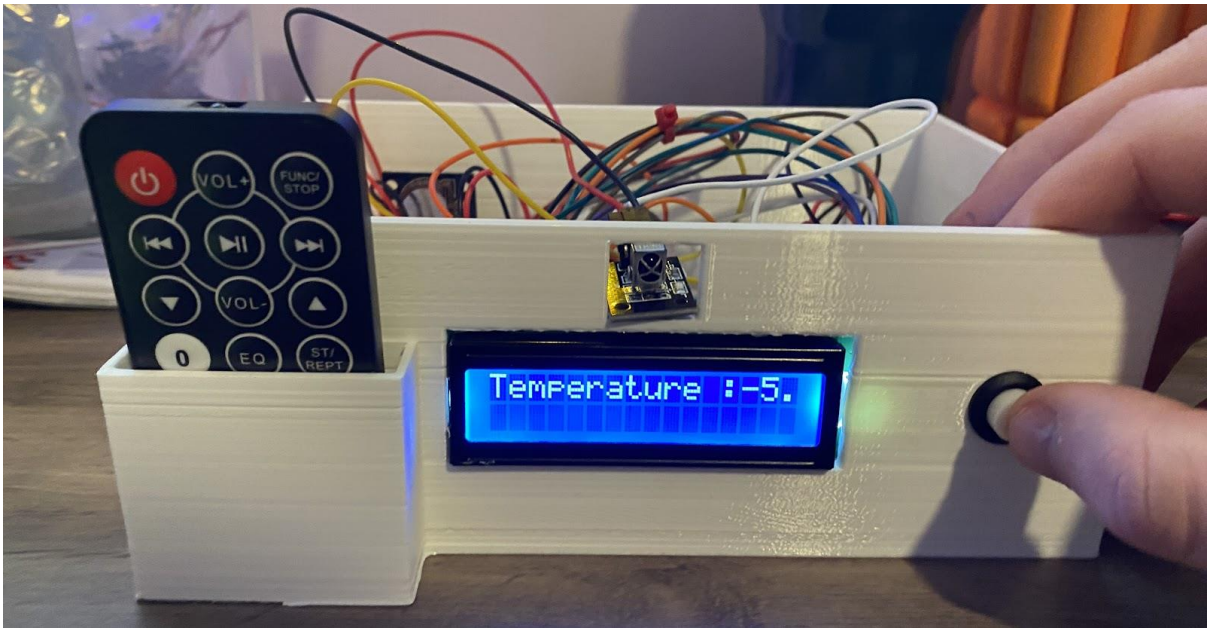

Figure 8: Display after Button was Pressed

Figure 9: Button after Vol+ was hit

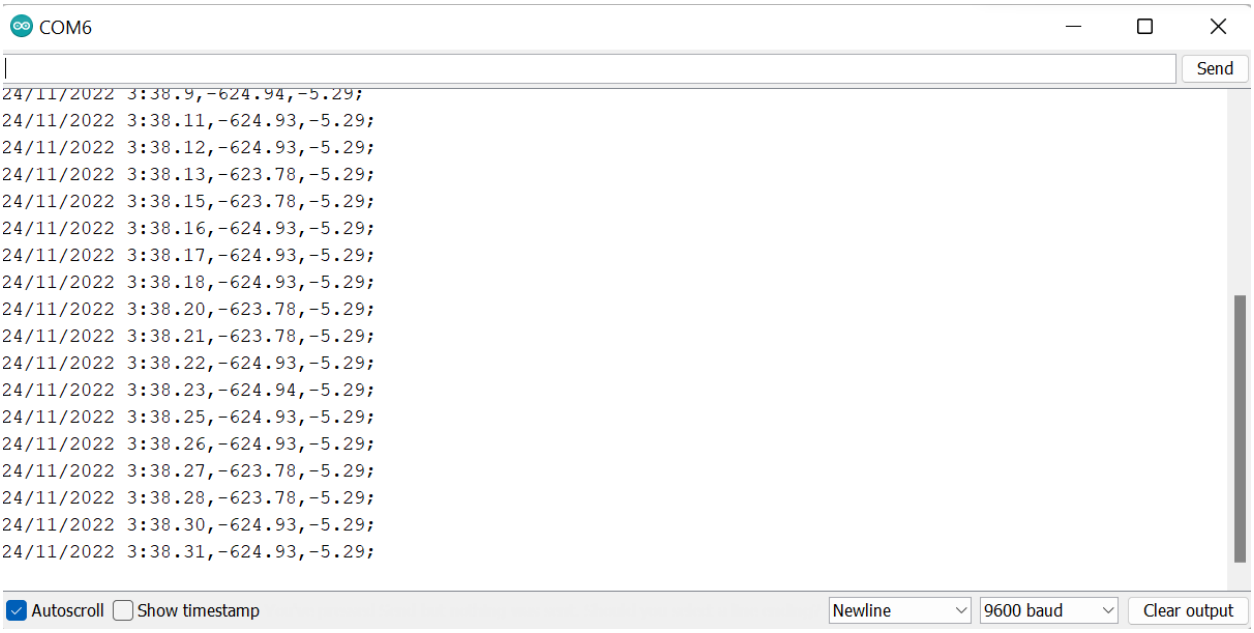

Figure 10: Serial Monitor Format

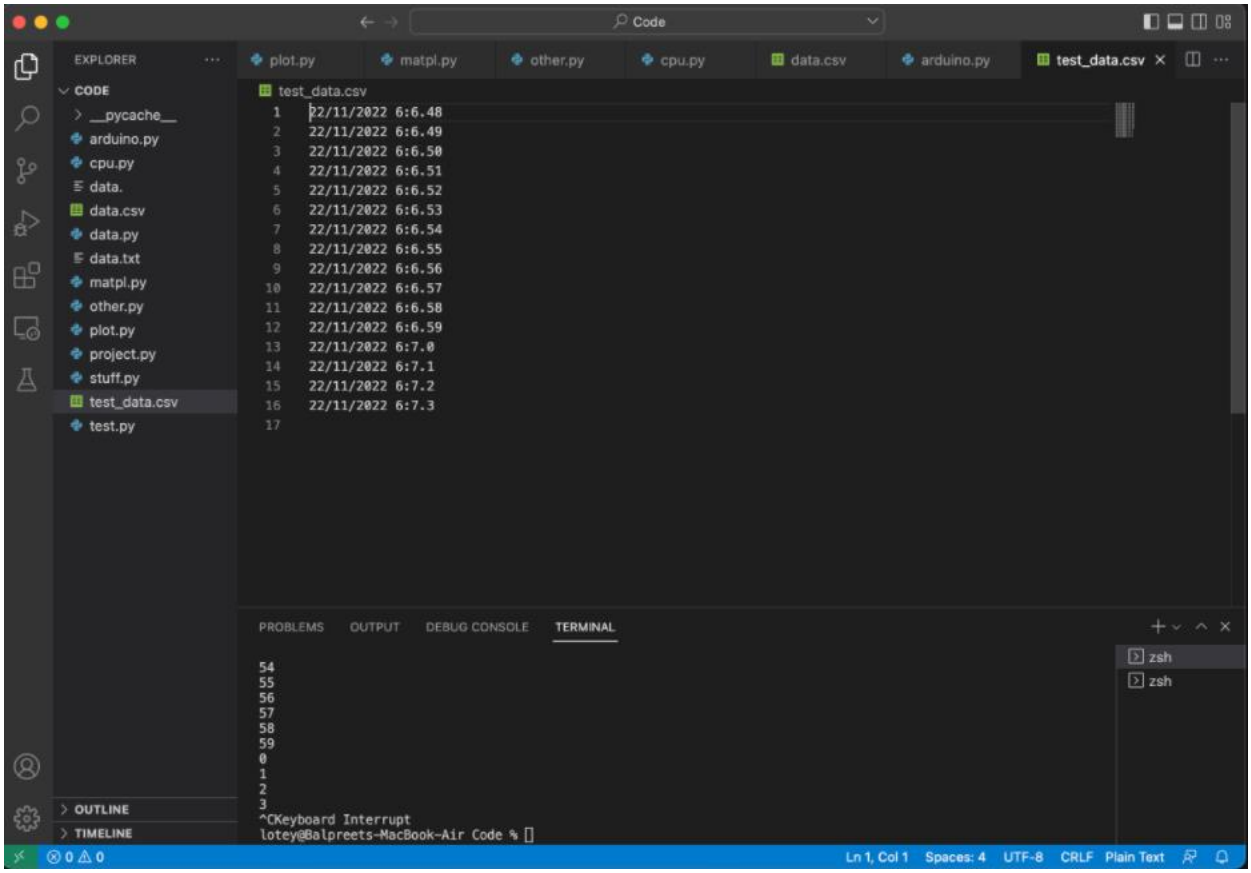

Figure 11: Data being Saved to CSV File

Final Code

```
#include <IRremote.hpp>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <LiquidCrystal.h>
#include <Wire.h>
#include <ds3231.h>

int m; // averaging factor for temperature
int n;//averaging factor for pressure sensors

int Psensor1 = A2;
int Psensor2 = A3;

int tempPin = 2;
int IR_RECEIVE_PIN = 3; //Connect IR sensor to Pin 3
int buttonpin = 4; //Connect Button to Pin 4

int i; // Used to enter in loop to measure data

float APZero = 102.4 ; //Analog Pressure Reading when P = 0 Psi
float APMid = 512; //Analog Pressure Reading when P=15Psi
float APMax = 921.6; //Analog Pressure Reading when P = 30 Psi
int PressureMax = 30; // Max rated Pressure of Transducer is 30 Psi

float height = 1; // This is the constant difference in Height in meters of the 2 Pressure
Transducers
float g = 9.81; //Gravitational Constant in N/kg or m/s^2
float row; // Value for the density of the fluid
float rowWater = 998.23; //Density of water at 20 degree Celsius in Kg/m^3
float SG; //Value will be used to Store SG value
float Plato; //Value will be used to store Plato conversion

float TopPressure;// Value will be used to store Psi value
float TopPressureRead; //
float BottomPressure; // Value will be used to store Psi value
float BottomPressureRead;//
float PaTop; // Value will be used to convert and store Psi to Pa
float PaBottom;// Value will be used to convert and store Psi to Pa

float tempValue; // Value to Store Temperature Value
float TempSum;
```

```
float TempRead;
int buttonState; // Value to Save Current State of the Button

struct ts t; //Array for RTC

OneWire oneWirePin(tempPin); //Defining temperature sensor
DallasTemperature sensors(&oneWirePin); //Passing sensor through Dallas Temperature
LiquidCrystal lcd(10, 11, 6, 7, 8, 9);

void setup()
{
 IrReceiver.begin(IR_RECEIVE_PIN, ENABLE_LED_FEEDBACK); // Start the receiver
 Serial.begin(9600); //Start Serial monitor
 sensors.begin();//Starting Temp sensor to request data
 lcd.begin(16, 2); //Start LCD Screen
 pinMode(buttonpin, INPUT); // Set buttonpin as an input

 Wire.begin();
 DS3231_init(DS3231_CONTROL_INTCN);

 //t.hour=10; //Next 7 Lines are used to set time. If clock dies, replace battery, and replace
values with your time to reset time
 //t.min=53; //switches minutes to 53
 //t.sec=30; //switches seconds to 30
 //t.mday=8; //switches day to the 8
 //t.mon=11; //Switches month to the november
 //t.year=2022 // switches year to 2022
 //DS3231_set(t); //sets clock to selected time
}

void loop() {

 if (IrReceiver.decode()) // Loop to receive input from remote
 {
  switch (IrReceiver.decodedIRData.decodedRawData) {
    case 0xBA45FF00://This is for the Start button on the remote
      i = 1; //Set i=1 to enter in loop to start measuring and saving data
      break;
    case 0xB946FF00://This is for the vol+ button
      i = 0; //This is to stop the loop
      break;
  }
  IrReceiver.resume(); // Enable receiving of the next value
 }
```

```
 if (i == 0) // If vol+ button is hit, it stops measuring data and displays following message
 {
  lcd.clear();
  lcd.setCursor(0, 0); // Sets cursor on first row, first column
  lcd.print("Hit Start to");
  lcd.setCursor(5, 1); //Sets curson on fifth row, first column
  lcd.print("begin");
  delay(500);
 }

 if (i == 1) //If Start button is hit, enter this function to begin measurements
 {

  if (m < 25) {
   sensors.requestTemperatures();
   TempRead = sensors.getTempCByIndex(0); // Storing Temperature Value
   TempSum = tempValue + TempRead;
   m++;
  }

  tempValue = TempSum / 25;
  m = 0;

  if (n < 25) {
   TopPressureRead = analogRead(Psensor1); //Reads Analog value from the top pressure
sensor
   TopPressure = TopPressure + TopPressureRead; // Takes the sum of all 25 readings
   BottomPressureRead = analogRead(Psensor2); //Reads Analog value from bottom pressure
sensor
   BottomPressure = BottomPressure + BottomPressureRead; // Takes the sum of all 25
readings
   n++;
  }

  TopPressure = TopPressure / 25;
  BottomPressure = BottomPressure / 25;
  n = 0;

  TopPressure = ((TopPressure - APZero) * PressureMax) / (APMax - APZero); //Converting
Analog Pressure to Psi
  PaTop = 6.89476 * TopPressure * 1000; //Converting Psi to Pa
```

```
  BottomPressure = ((BottomPressure - APZero) * PressureMax) / (APMax - APZero);
//Converting Analog Reading to Psi
  PaBottom = 6.89476 * BottomPressure * 1000; //Converting Psi to Pa

  row = (PaBottom - PaTop) / (g * height);
  SG = row / rowWater;
  Plato = (-1 * 616.868) + (1111.14 * SG) - (630.272 * SG * SG) + (135.997 * SG * SG * SG);

  buttonState = digitalRead(buttonpin); //Read current state of button

  if (buttonState == HIGH) {
    lcd.clear();
    lcd.setCursor(0, 0); // Sets cursor on first row, first column
    lcd.print("Plato:");
    lcd.print(Plato); // Print specific gravity in Plato
    lcd.setCursor(0, 1); //Sets cursor on first row, second column
    lcd.print("SG:");
    lcd.print(SG);//Print Specific Gravity
    lcd.print("kg/m3");
    delay(500);
  }
  else if (buttonState == LOW) {
    lcd.clear();
    lcd.setCursor(0, 0); // Sets cursor on first row, first column
    lcd.print("Temperature :");
    lcd.print(tempValue); // Print Temperature value on the screen
    lcd.setCursor(0, 1); //Sets cursor on first row, second column
    lcd.print(""); //Print whatever we need
    delay(500);

  }
  DS3231_get(&t);
  Serial.print(t.mday); //Displays current date and time
  Serial.print("/");
  Serial.print(t.mon);
  Serial.print("/");
  Serial.print(t.year);
  Serial.print(" ");
  Serial.print(t.hour);
  Serial.print(":");
  Serial.print(t.min);
  Serial.print(".");
  Serial.print(t.sec); //End of Current Date
  Serial.print(",");
```

```
   Serial.print(Plato);
   Serial.print(",");
   Serial.print(tempValue);
   Serial.println(";");

  }

}
```

Wrike Link and Project Plan

Our project plan for the last 2 weeks is to prepare our prototype for Design Day on December 1st. We plan to do one last test on the pressure sensors November 29th for our final test. In addition to preparing for Design Day, our team will be going through all our documentation completed during deliverables A to H to put together the user manual to accompany our prototype into our MakerRepo.

https://www.wrike.com/workspace.htm?acc=4975842&wr=20#folder/966341779/list?filters=status%3Dactive&sidePanelItemId=969519166&sortOrder=1&spaceId=-1&viewId=108931260