

Deliverable F

Prototype 1 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 6th, 2022**

**University of Ottawa
Faculty of Engineering**

Table of Contents

Introduction	3
Feedback from Client Meeting 2	4
Prototyping	5
IR Sensor Testing.....	5
Code for IR Receiver.....	7
LCD Screen + Button	8
Code for LCD Screen + Button	10
Temperature Sensor Prototype	11
Code for Temperature Sensor	12
Engineering Analysis – Pressure Sensors	13
Updated BOM	15
Component Links	17
Library Links	17
Prototype Test Plan.....	17
Design Changes.....	18
Wrike Link and Project Plan	18

Introduction

In this report, our team will discuss the feedback received during client meeting 2 with Shane from Beyond the Pale, our team will be outlining the different early prototypes used to ensure multiple sub systems and sensors function as planned, we will be outlining a new prototyping test plan for the remaining prototypes as well as discussing what changes have been made in our design.

Feedback from Client Meeting 2

At our client meeting with Shane. We had presented our three final solutions. By explaining how each idea would measure specific gravity and temperature. Shane was also able to get an idea of how the device could be integrated into his tank and the multiple ways he was able to see the results.

After hearing the design for solution 1. Shane had explained that he was familiar with the process of using pressure sensors to calculate the specific gravity of the wort. He had mentioned that it was a harder solution to make accurate but with the access to technology and software now it wouldn't be as much of a challenge compared to when he had tried a similar technique. However, he mentioned he liked the concept the most.

Shane liked solution 2 because this unique approach has already been implemented with home brewing kits. Shane had mentioned the accuracy of this solution wouldn't be as great as the others automatically but by testing and adjusting the system we could make it accurate. Unfortunately, because of time constraints and the budget, this solution wouldn't be possible to complete to the standard given by Shane.

Just like Solution 2 Shane also took an interest in solution 3. Shane stated the idea has been successful commercially. After further discussion after the meeting, the team decided that this solution wasn't going to be easy to manufacture as well install and remove consistently.

A common comment that Shane had made was that he was interested in having the results of the specific gravity shown on his computer in real time. Shane also liked that the data could be saved on a SD card that allowed him to manually save it on his computer after the fermentation was complete.

Overall, Shane was able to give us good and helpful feedback. This led the team to make the final decision to continue with solution 1. We chose solution 1 because it follows the design criteria the most. As well because of the budget allocated to the project and the deadline of the project this was the only solution we can complete to satisfaction.

Prototyping

IR Sensor Testing

To test the IR sensor, I connected the receiver to the Arduino as seen in figure 1. This prototype consists of an IR receiver, a remote, an Arduino, jumper cables and a breadboard. The goal of this prototype was to ensure the IR receiver is working and accepts the correct commands. To begin, once the code was set up, I opened the serial monitor and printed all raw data being received by the receiver, I individually hit every button and recorded the reading that was being shown on the monitor. Once I had these values, I edited my code to begin doing commands for different buttons. Since our project only requires this as a start and stop button, I set $i=1$ when the power button was hit and $i=0$ when the vol+ button was hit. This i integer will be used to control whether the Arduino will enter the loop to begin measuring data or continue awaiting a command from the remote. When I uploaded my code to my Arduino when I hit the power button, the message "Data is being measured" popped up on my serial monitor continuously and when the vol+ was hit, the message stopped rolling in which is exactly what our system will do.

Table 1: Button values

Button	Raw Value of Each Button
Power	0xBA45FF00
Vol+	0xB946FF00
FUNC/STOP	0xB847FF00
Back	0xBB44FF00
Play/Pause	0xBF40FF00
Forward	0xBC43FF00
Down	0xF807FF00
Vol-	0xEA15FF00
Up	0xF609FF00
0	0xE916FF00
EQ	0xE619FF00
ST/REPT	0xF20DFF00
1	0xF30CFF00
2	0xE718FF00
3	0xA15EFF00
4	0xF708FF00

5	0xE31CFF00
6	0xA55AFF00
7	0xAD52FF00
8	0xAD52FF00
9	0xB54AFF00

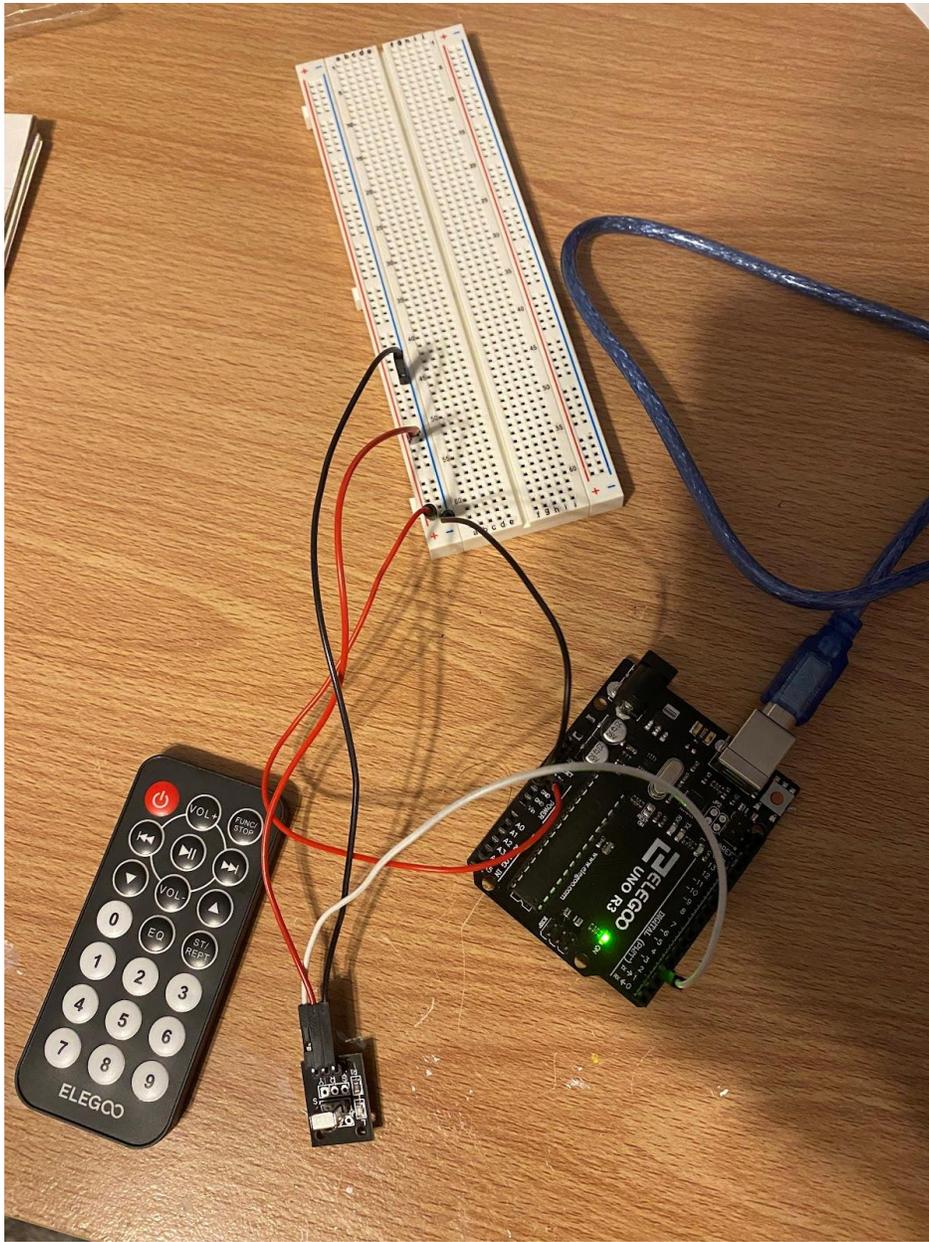


Figure 1: IR receiver and Remote Testing Prototype

Code for IR Receiver

```
#include <IRremote.hpp>

int IR_RECEIVE_PIN = 3;
int i;

void setup()
{
  IrReceiver.begin(IR_RECEIVE_PIN, ENABLE_LED_FEEDBACK); // Start the receiver
  Serial.begin(9600); //Start Serial monitor
}

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 == 1) //If Start button is hit, enter this function to begin measurements
  {
    Serial.println("Data is being measured");
  }
}
```

LCD Screen + Button

To test the LCD Screen and the button, I wired the LCD screen and the button to a breadboard which was connected to an Arduino. The main goal of this test was to ensure that when the button was pressed, the screen would switch to display other quantities and it would clear the screen of the other quantity. In addition to the LCD screen and the button, I also needed a 10K Ω resistor for the button and a 10K Ω potentiometer for the LCD screen. The potentiometer is needed while we wait for the I2C module to be approved for purchase.

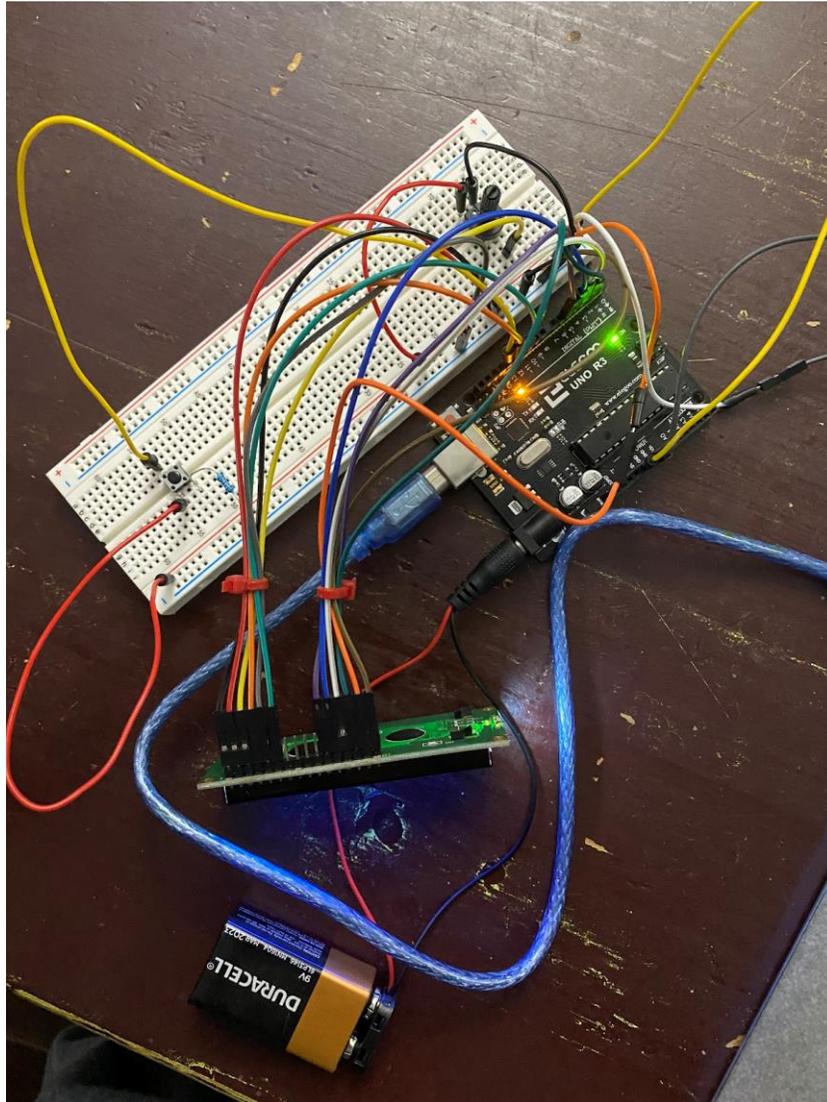


Figure 2: LCD Screen Prototype

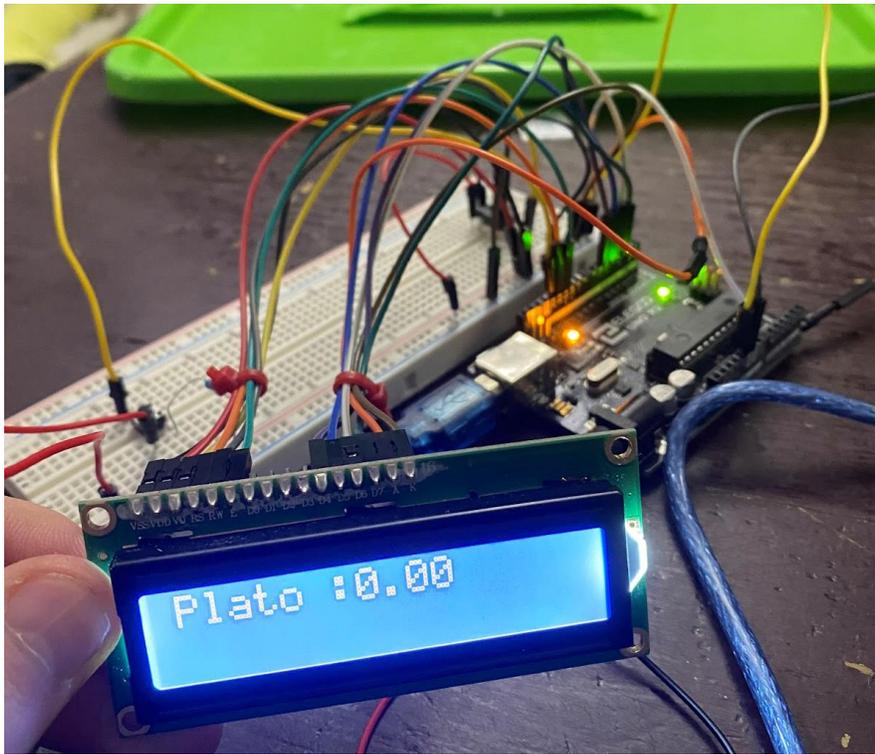


Figure 3: Case 1 - The button was not pressed.

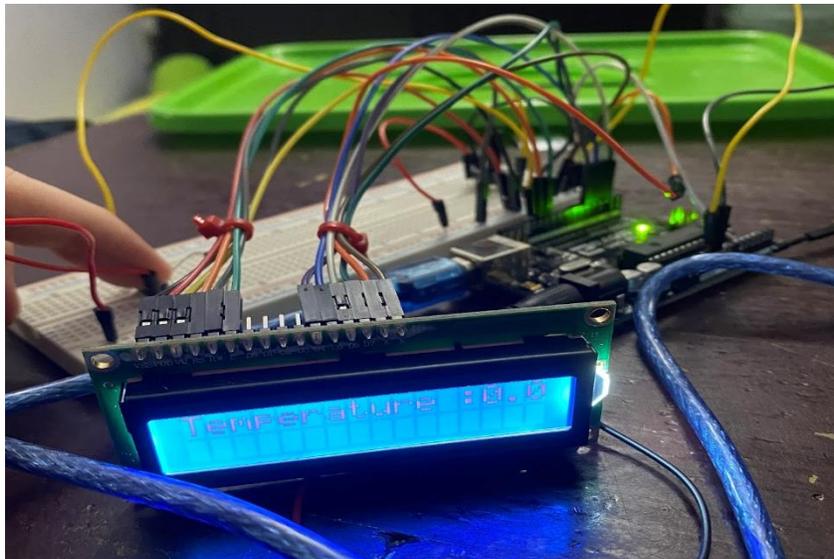


Figure 4: Case 2 - The button was pressed

Code for LCD Screen + Button

```
#include <LiquidCrystal.h>
```

```
int buttonpin = 13;
```

```
float Plato;
```

```
float tempValue;
```

```
int buttonState;
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
void setup() {
```

```
  lcd.begin(16, 2); //Start LCD Screen
```

```
  pinMode(buttonpin, INPUT); // Set button to input
```

```
  lcd.print("Hello World");
```

```
}
```

```
void loop() {
```

```
  buttonState = digitalRead(buttonpin); // Read Button Values
```

```
  if (buttonState == HIGH) {
```

```
    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("Spec Grav:"); //Print whatever we need
```

```
    delay(500);
```

```
    lcd.clear();
```

```
}
```

```
  else if (buttonState == LOW) {
```

```
    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);
```

```
    lcd.clear();
```

```
}
```

```
}
```

Temperature Sensor Prototype

To prototype the temperature sensor, our temperature sensor was wired up to the Arduino using a 5k Ω resistor, wires and a breadboard as seen in figure 5. The goal of this prototype was to see if our temperature sensor was working. To test if our temperature sensor is working, we put a pot of water on the stove and filled a glass full of water and ice. After letting the water heat up on the stove and letting the ice melt to mix with the room-temperature water. Using the serial monitor, we were able to display the readings of the temperature sensor as seen in Figures 6 and 7 and we were able to cross-reference these readings with a digital thermometer.

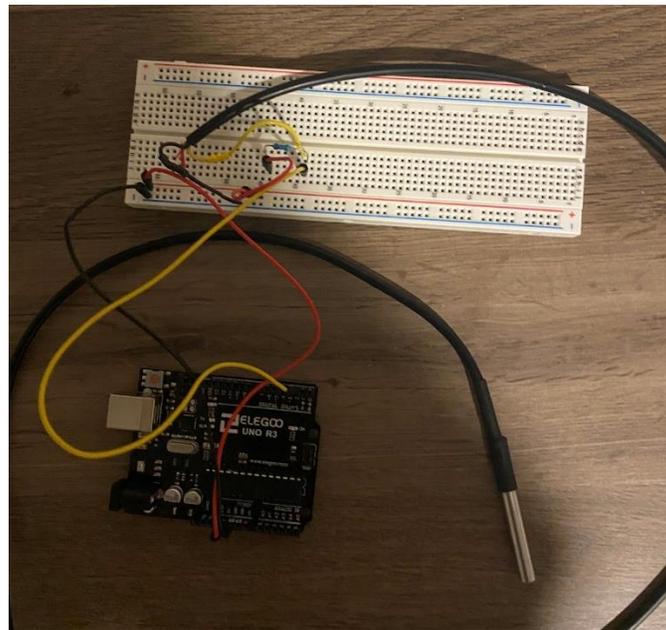


Figure 5: Temperature Sensor Prototype

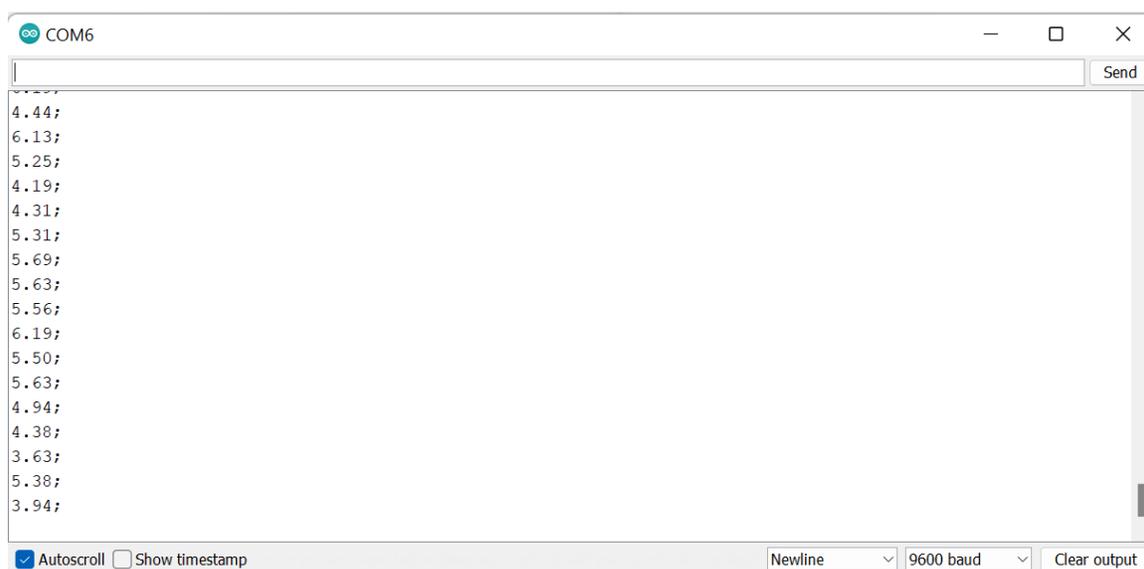


Figure 6: Temperature sensor in almost freezing water

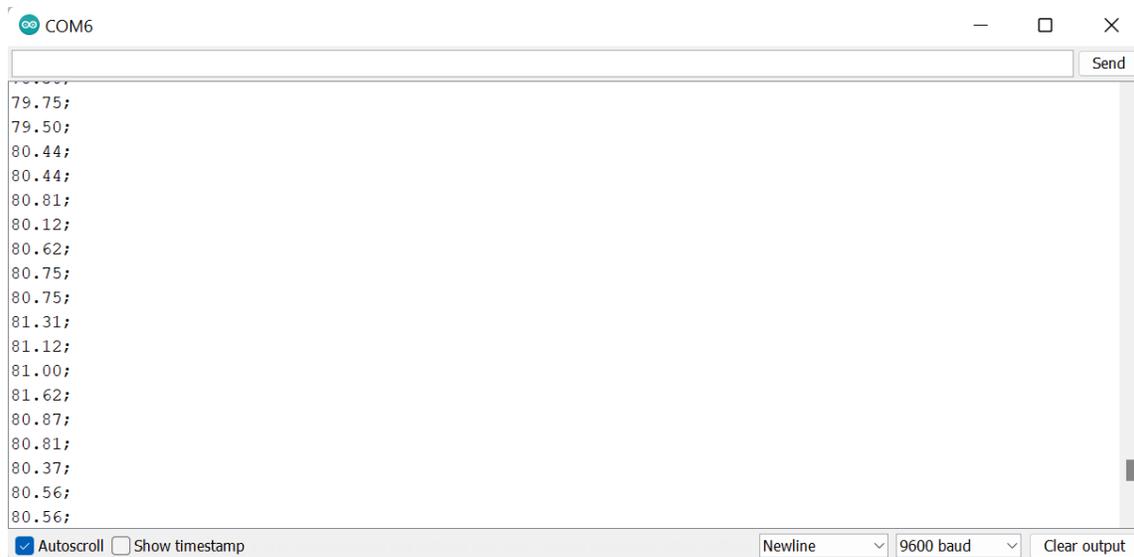


Figure 7: Temperature sensor in almost boiling water

Code for Temperature Sensor

```
#include <OneWire.h>
#include <DallasTemperature.h>

int tempPin = 6; //Connect Temperature sensor to pin 6
float tempValue; //Value from sensor will be saved here

OneWire oneWirePin(tempPin); //Defining temperature sensor
DallasTemperature sensors(&oneWirePin); //Passing sensor through Dallas Temperature

void setup() {

sensors.begin();//Starting sensor to request data
Serial.begin(9600); //Start Serial monitor

}

void loop() {
sensors.requestTemperatures();
tempValue = sensors.getTempCByIndex(0);// Getting Temperature Value
Serial.print(tempValue); //Printing Temperature Value to Serial Monitor
Serial.println("");
delay(1000); //delaying 1 second

}
```

Engineering Analysis – Pressure Sensors

An engineering analysis was performed on pressure sensors to see two things: 1) the distance between the two sensors that should be down to both fit in the fermentation tank and that the pressure difference should be substantial enough to get reliable results when measuring and to see if the change due to change in density is substantial enough. 2) To see if sensors are accuracy is low enough that we can use these sensors reliably. Measurements were made in pascals instead of PSI, due to PSI being converted to pascals before the calculation of change of pressure is made in our prototype.

It was found that 8 to 12 inches in length is optimal for the measurements and is best size-wise in terms of installation. Accuracy is in a reasonable range. The main worry is that if the accuracy of the pressure sensor is so large that it would mean that our calculations would not be accurate to the actual density.

Engineering analysis - Pressure sensor

$$g = 9.81 \text{ m/s}^2$$

$$\text{wort before fermentation} \Rightarrow P_{\text{wort } 1} = 1.100 = 1100 \text{ kg m}^{-3}$$

$$h_1 = 5 \text{ inches} = 0.127 \text{ m}$$

$$\text{wort after fermentation} \Rightarrow P_{\text{wort } 2} = 1.000 = 1000 \text{ kg m}^{-3}$$

$$\Delta P = \rho g h$$

$$h_2 = 2 \text{ inches} = 0.2032 \text{ m}$$

$$h_3 = 10 \text{ inches} = 0.254 \text{ m}$$

$$h_4 = 15 \text{ inches} = 0.381 \text{ m}$$

$$h_5 = 20 \text{ inches} = 0.508 \text{ m}$$

Calculating the change of pressure difference between the wort before and after fermentation to see if there would be any significant change to find to optimal distance between pressure sensors

h_1 :

$$\Delta P_1 = (0.127 \text{ m})(9.81 \text{ m/s}^2)(1000 \text{ kg m}^{-3}) = 1245.87 \text{ Pa}$$

$$\Delta P_2 = (0.127 \text{ m})(9.81 \text{ m/s}^2)(1100 \text{ kg m}^{-3}) = 1370.5 \text{ Pa}$$

$$129.63 \text{ Pa}$$

h_2 :

$$\Delta P_1 = 2192.73 \text{ Pa}$$

$$\Delta P_2 = 1993.321 \text{ Pa}$$

$$199.34 \text{ Pa}$$

h_3 :

$$\Delta P_1 = (0.254 \text{ m})(9.81 \text{ m/s}^2)(1000 \text{ kg m}^{-3}) = 2491.74 \text{ Pa}$$

$$\Delta P_2 = (0.254 \text{ m})(9.81 \text{ m/s}^2)(1100 \text{ kg m}^{-3}) = 2740.21 \text{ Pa}$$

249.17 Pa \Rightarrow The best between distance, and difference = pressure

h_4 :

$$\Delta P_1 = (0.381 \text{ m})(9.81)(1000) = 3737.61 \text{ Pa}$$

$$\Delta P_2 = (0.381 \text{ m})(9.81)(1100) = 4111.37 \text{ Pa}$$

$$373.76 \text{ Pa}$$

h_5 :

$$\Delta P_1 = 4983.48 \text{ Pa}$$

$$\Delta P_2 = 5481.828 \text{ Pa}$$

498.35 Pa \Rightarrow Best change, but need to keep in mind that this may be too big

\rightarrow So we can say that 10 inches is optimal

Engineering analysis - Pressure sensor accuracy calculations

Testing the pressure sensors for accuracy to see if sensors are good to use for experiment, to see if accuracy is too much.

$$\text{Error } \pm 2.010 = \pm 0.02$$

Vessel dimensions - Based off of cylindrical fermentation.

Figure 8: Calculations of change in Pressure Difference

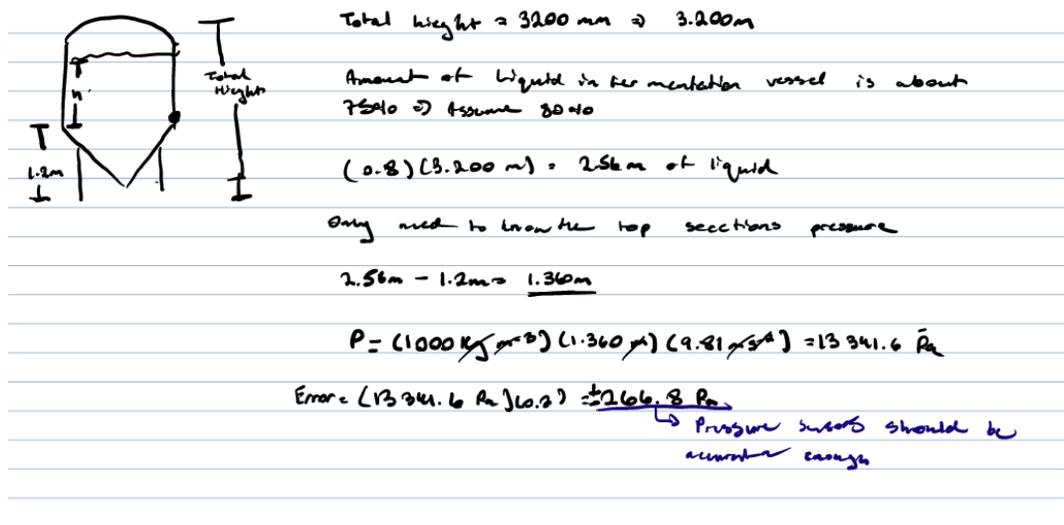


Figure 9: Calculations of Uncertainty

Measurements Used of Fermentation Vessel used:

https://www.winemakersdepot.com/Assets/ProductImages/15BBL_FV-1_2_800.png

Updated BOM

Table 2: Updated Bill of Materials

Parts	Quantity	Unit of Measure	Unit Cost (\$)	Estimate Cost(\$)	Link
Hardware Required					
Mil. Spec Aluminum Tubing 1OD 1ft Length	2	FT	0	0	Owned
Aluminum Rod 1-5/8 Dia 1ft Length	1	FT	0	0	Owned
Aluminum 2x6x1-1/2	1	EA	0	0	Owned
Food Grade Silicone Sealant	1	EA	0	0	Owned
3D printed Electrical Compartment Bottom	1	EA	0	0	MakerLab (to be printed)
3D Printing Filament	1	EA	10	10	MakerLab
3D printed Electrical Compartment Top	1	EA	0	0	MakerLab (to be printed)
O-Rings	6	EA	0	0	Owned
Screws	8	EA	0	0	Owned

Electrical Components Required					
Arduino	1	EA	0	0	Owned
Breadboard	1	EA	0	0	Owned
LCD Screen	1	EA	0	0	Owned
Pressure Transducer Sensor 30 Psi	2	EA	13.49	26.98	See Link Section
Latching Buttons	1	EA	0.54	0.54	See Link Section
Bluetooth connection	1	EA	12.99	12.99	See Link Section
DS18B20 Temperature Sensor	1	EA	14	14	See Link Section
IR Sensor	1	EA	0	0	Owned
Clock Module	1	EA	10	10	See Link Section
Remote	1	EA	0	0	Owned
4.7 K Ω Resistor	1	EA	0	0	Owned
10 K Ω Resistor	1	EA	0	0	Owned
10 K Ω Potentiometer	1	EA	0	0	Owned
Wiring	50	EA	0	0	Owned
Software Required					
Arduino IDE	1	EA	0	0	Owned
OnShape	1	EA	0	0	Owned
Python	1	EA	0	0	Owned
Libraries					
Liquid Crystal	1	EA	0	0	See Library Links
OneWire	1	EA	0	0	See Library Links
Spi	1	EA	0	0	See Library Links
DS3231	1	EA	0	0	See Library Links
DallasTemperature	1	EA	0	0	See Library Links
Irremote	1	EA	0	0	See Library Links
Software Serial	1	EA	0	0	See Library Links
Total Cost (Without Tax)				74.51	
Total Cost(With Tax)				79.32	

Spreadsheet links:

https://docs.google.com/spreadsheets/d/1HLeKHimS2K6xT5xADkeC8T_iUnEW0eZaAFPuYZtyU8s/edit#gid=0

Component Links

Pressure Transducer Sensor 30 Psi: https://www.amazon.ca/Thread-Stainless-Pressure-Transducer-Compatible/dp/B08G4ZX4LY/ref=sr_1_11?keywords=pressure%2Btransducer&qid=1666567830&qu=eyJxc2MiOilzLjk5IiwicXNhIjoiMy4yNilslInFzcCI6IjluNTAifQ%3D%3D&s=industrial&sprefix=Pressure%2BTran s%2Cindustrial%2C153&sr=1-11&th=1

Latching Buttons: <https://edu-makerlab.odoo.com/shop/product/push-button-switch-81#attr=148>

Bluetooth connection: <https://edu-makerlab.odoo.com/shop/product/bluetooth-module-9?search=bluetooth#attr=255>

DS18B20 Temperature Sensor: <https://edu-makerlab.odoo.com/shop/product/grove-temperature-sensor-47?search=temperature+sensor#attr=>

Clock Module: https://www.amazon.com/AT24C32-Replace-Arduino-Batteries-Included/dp/B07Q7NQTQS/ref=sr_1_3?keywords=Arduino+RTC&qid=1667612354&qu=eyJxc2MiOilzLjY1IiwicXNhIjoiMy41NyIsInFzcCI6IjMuNDUifQ%3D%3D&sr=8-3

Library Links

Liquid Crystal: <https://www.arduino.cc/reference/en/libraries/liquidcrystal/>

OneWire: <https://www.arduino.cc/reference/en/libraries/onewire/>

Spi: <https://www.arduino.cc/reference/en/language/functions/communication/spi/>

DS3231: <https://www.arduino.cc/reference/en/libraries/ds3231/>

DallasTemperature: <https://www.arduino.cc/reference/en/libraries/dallastemperature/>

Irremote: <https://www.arduino.cc/reference/en/libraries/irremote/>

Software Serial: <https://docs.arduino.cc/learn/built-in-libraries/software-serial>

Prototype Test Plan

Table 3: Prototype Test Plan

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 Bluetooth is working	The Bluetooth module will be connected to the Arduino. We will see if the Bluetooth is passing the data.	1.data being send by Bluetooth	Test until data is passing correctly
3	Determine if the logging system is working	This prototype will consist of the Bluetooth sensor, the real-clock sensor, and the pressure sensors.	1.Data being sent over Bluetooth 2.if correct data is being received	Test until correct data is being received
4	Determine if the whole 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

Design Changes

Due to issues with obtaining an I2C module for our LCD screen, we now need to wire in our screens using the pins on the screen. Wiring an LCD screen this way is a bigger problem than expected due to the number of digital pins required. When using an I2C, we only need 2 analog pins and without it, you need 6 digital pins. Our Arduino Uno only has 14 digital pins, so we did not have enough pins to wire in the LCD screen. Since the screen is a vital part of our design, we needed to eliminate one of our digital components. Our design contains a Bluetooth module to transfer/log our data in addition to an SD card module as a fail-safe in case the Bluetooth connection was ever dropped, so we decided to eliminate the SD card module from our design freeing up 4 digital pin slots. In addition to removing the SD card module, we also removed a button since we can achieve the same result with one latching button instead of 2 momentary buttons.

Wrike Link and Project Plan

Our project plan for the remaining weeks is to attempt to finish our pressure sensor prototype since we were not able to obtain the sensors before the prototype 1 deliverables due date and then we will combine all sensors and modules that have all been tested into one big prototype to ensure the system works correctly and can be tested further. Our team will also start to manufacture and print our casing for our final prototype.

<https://www.wrike.com/workspace.htm?acc=4975842&wr=20#folder/966341779/list?filters=status%3Dactive&sidePanelItemId=985887791&sortOrder=1&spaceId=-1&viewId=108931260>