GNG1103

Design Project User and Product Manual

Project Deliverable K: User and Product Manual

Submitted by:

GROUP 16

Alan Wang - 300140660

Rebecca Kimbell - 300233280

Christopher Al-Rahi - 300304285

Abby Falkenham - 300324956

Mohamed Reda Raki - 300281028

12 - 07 - 2022

University of Ottawa

Table of Contents

List of Figures

List of Tables

1 Introduction

This User and Product Manual (UPM) provides the information necessary for all possible users whether it be future students enhancing our design, workers at BTP, or at other breweries to effectively use Beer Goggles and for prototype documentation.

Beyond the Pale is a local brewing company in Ottawa who required a specific gravity and temperature monitoring system for their company. Measuring specific gravity is a vital step in the beer making process. Beyond the Pale's current process for measuring specific gravity is inefficient, resulting in loss of time and product. As the company is looking towards expansion, it is imperative that these systems are put in place to prepare for the increased work they will take on. That is why BTP is in need of a specific gravity measuring system to streamline and expand the brewing process; allowing for simple and accurate data monitoring and collection. The system must be able to monitor at a consistent rate, meet all hygienic standards of the brewing process, and resist the brewing process condition.

Our solution to this problem was to create a specific gravity measuring device using mass. This device measures the mass of the beer using a load cell and measures the volume using an ultrasonic sensor; the specific gravity is later calculated using d=m/v within the arduino. This data would then be sent to a computer to be graphed so that the user can visualize the results.

2 Overview

At the beginning of this course, the client explained that keeping track of the specific gravity of beer is crucial to the fermentation process. Currently, Beyond the Pale uses a hydrometer to measure the specific gravity. This requires workers to manually take beer out of the tank to find the specific gravity. This method is not only wasteful but also takes a long time. This led to BTP's need for a more efficient way of measuring specific gravity in an automatic, fast, and accurate way that does not waste the beer.

After understanding the customer's needs, we designed a set of guidelines to solve the problem. Among the many designs, our product has several unique features.

1. Our project is operated in a closed system that is contained in a side tank. No waste or pollution could occur. Also, a side tank is easy for the users to manage.
2. Our product uses gravity as the measurement standard, which is more reliable, and its error in our experiment is very small.
3. Our product uses custom code so that the user can determine the interval of each measurement and record the data.
4. Our product can use both an external power supply and a built-in battery, so there is no need to worry about power failure.
5. Our product contains a lot of recycled materials and are very environmentally friendly.

The main function of our project is to measure the density of beer. After the user decides to start measuring, the pump inside the side tank pumps the beer out of the tank and into the side tank. We then design a program that measures the density of the beer itself and displays it on the user's computer screen. At the end of the measurement, the beer is pumped back into the tank, which prevents waste and pollution.

2.1 Conventions

Our project was designed to be easy to use. Most of the time, users just press buttons and read data. After setting up the product, users simply turn on the computer, run the program, and turn on the water pump when the program is finished (about nine seconds).

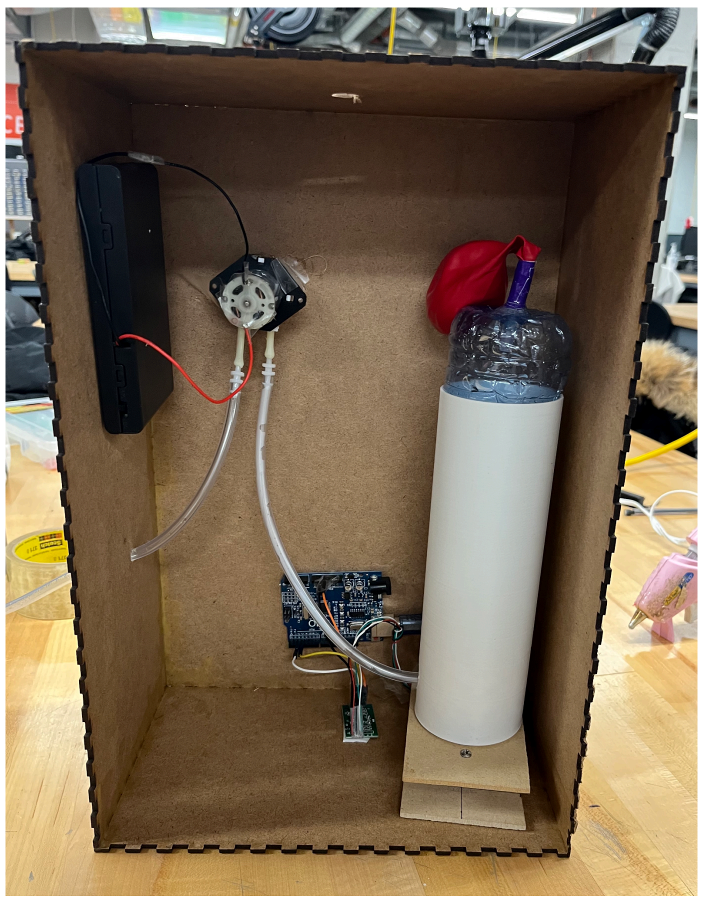
The program measures the density of the beer itself. At the end of the measurement, the user flicks a switch and the pump sends the beer back to the tank. In our future design, these processes will be further simplified. Users only need to press a button once to get the data, and after the measurement, the pump will send the beer back to the tank without any manual operation.

2.2 Cautions & Warnings

Our product design is simple and easy to maintain, but there are a few caveats.

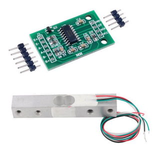
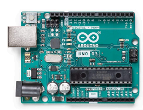
1. Take care to clean water pumps, pipes, and side tanks. Users should clean these parts after each brew to avoid contamination.
2. Pay attention to the battery status. If there is a bulge or leakage needs to be replaced in time. Our equipment has a wooden case, which may pose a great danger if the battery catches fire.
3. Do not disassemble the device while it is running. Also, please do not disassemble the side tank in brewing. This can lead to serious incidents such as beer leaks.

For the convenience of users and the quality of beer, regular inspection and maintenance are necessary.

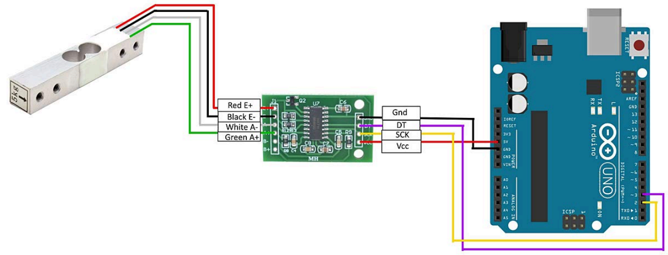


3 Getting started

The components of our products are:

- A peristaltic pump 
- A 12V battery 
- Silicone tubing
- An HX711 load cell and amplifier 
- An Arduino Uno 
- Arduino Male-female wires 
- A 3D model
- An airtight bottle that serves as a side tank

- To begin with, wire the battery to the peristaltic pump by connecting the red wire to the positive pole of the pump, and the black one to the negative pole.
- Connect the silicone tubing to the pump’s entry.
- Connect the exit tube to the bottle that serves as side tank, and make sure it is airtight. If it is not, add some silicone along the side of the tube.
- Put the bottle inside the 3D model and get the tube out of the hole at the bottom.
- Connect that tube to the pump’s exit.
- Now set up the load cell and arduino. To do that, start by setting up the scale.
- Screw the two platforms to the sides of the load cell, using 6-32 and 10-32 screws.
- Connect the load cell to the amplifier, then the amplifier to the arduino board as follows.



- Connect the arduino uno to the computer.
- Get the entry tube in a tank that contains a liquid.
- Turn the battery switch on, and upload the code to the arduino uno.

3.1 Configuration Considerations

As described before, this device uses a 12V peristaltic pump, which is powered by a 12V battery, and connects the two containers using silicone tubes. The pump transfers the liquid between the containers, one of which is placed on a load cell. The latter works as a scale that weighs the liquid inside the bottle to help determine its density and therefore specific gravity.

3.2 User Access Considerations

Keep out of children's reach. Once the device is all set, avoid touching the silicone tubes, not to disconnect them from the pump, which would mess up our closed system. Avoid touching the load cell as it is weighing the liquid.

3.3 Accessing/setting-up the System

Once the device is set and the tubes are connected to the entry and exit containers, turn the battery switch on. Once you do that, you should see the liquid travelling from the main container into the side bottle. As soon as the liquid reaches the entry point of the bottle, upload the code into the Arduino Uno to start the calculations.

3.4 System Organization & Navigation

It all starts with the pump that gets the liquid out of the main container and pumps it into the bottle. Once the liquid is at the entry of the bottle, the code is uploaded to dynamically calculate the specific gravity. How it works is that the load cell is a scale that is placed at the bottom of the bottle, and weighs the liquid that is being pumped into it. A balloon at the top end of the bottle serves as a pressure releasing system, so the bottle doesn't blow up. All the air that the liquid replaces goes into the balloon to maintain the same pressure inside the bottle. The rest is done by the code, which gets the weight from the load cell, and calculates the volume as the liquid is being pumped using the flow-rate of the pump. It divides the weight by the volume to calculate the density dynamically. It then divides the density of the liquid by the density of water, which is basically 1g/mL, to give us the specific gravity.

3.5 Exiting the System

Once done with the system, turn the battery switch off. Then reverse the connections between the battery and the pump. The red wire would then be connected to the negative pole, whereas the black one would be connected to the positive pole. Turn the battery switch on again, and wait for all the liquid to go back to the main container. Make sure to turn the battery switch off as soon as the liquid leaves the bottle completely to avoid a pressure deficit inside the bottle.

4 Using the System

There is only one feature to be used manually from the system itself which is the on/off switch. The rest of the features are seen from the mobile output of the product which allow the system to view data, reverse pump flow, and restrict pump flow completely.

4.1 ON/OFF Switch

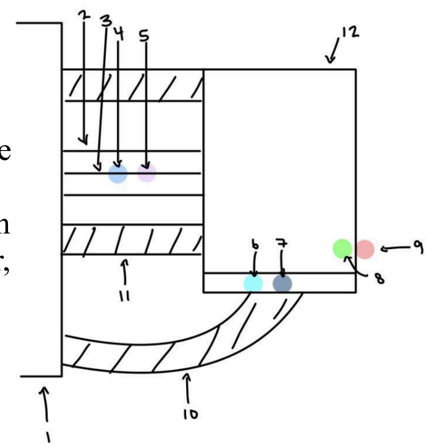
First Function will be an ON/OFF switch placed on the top right according to the photo shown to the right.

Once turning the system ON, five things will occur:

1. Solenoid Valve Opens allowing beer flow into the pump to occur
2. Pump turns on sending beer into the tank
3. Scale starts measuring mass
4. Ultrasonic Sensor starts measuring volume
5. Code starts calculating specific gravity once Volume reaches atleast ¾ full.

When turning the system OFF, Four things will occur:

1. Pump removes all excess beer from the product.
2. Ultrasonic sensor makes sure that the system is empty
3. Solenoid Valve will closed preventing any more beer to enter the system.
4. Once Ultrasonic Sensor ensures that there is no more beer in product and valve closes, all other systems (Pump, Scale, Sensor, Arduino) Turn off as well.



4.2 Output Features

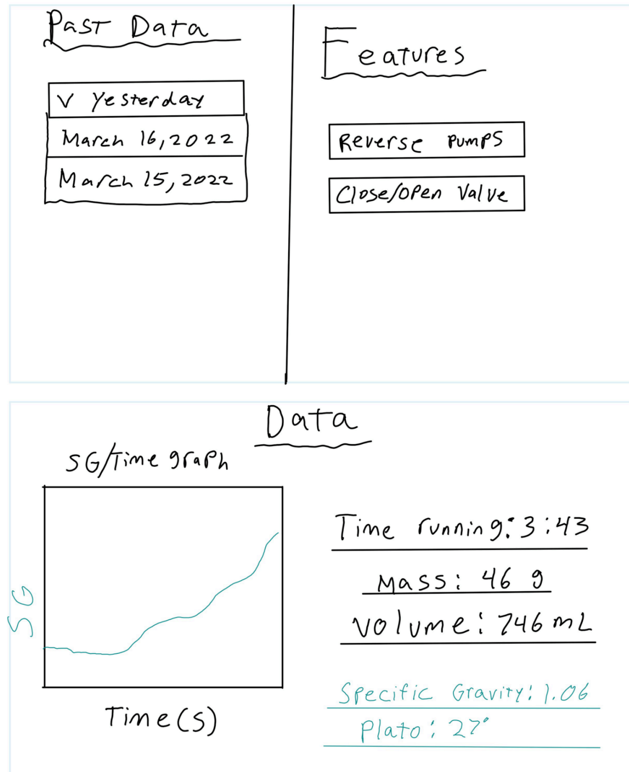
Since the systems output will be able to be seen through a phone, there will be some features that are controlled with the use of the mobile app. Not only does the app display the data received from the arduino, there are also three extra features that can be done with the app.

1. Reverse Pumps
2. Close/Open Valve
3. View Past Data

4.2.1 Reverse Pumps

The Reverse Pumps feature allows the user to switch the flow of the pumps. This can be done if there was a problem in the system or if the flow of the pumps somehow managed to jam. For this to work, the system would send the input to the arduino which would then send a signal to the pump and switch the flow.

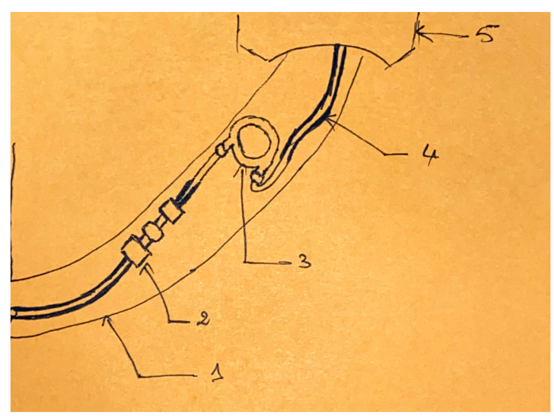
Figure 1



4.2.2 Close/Open Valve

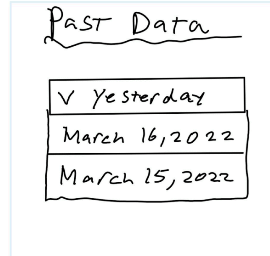
The Close/Open Valve Button is used to Close or Open the Solenoid Valve. Closing the Valve will restrict the flow of beer into the peristaltic pump and putting the machine to a stop. Opening the Valve allows the flow of beer into the pump as the valve is located on the fermentation tank side of the peristaltic pump. The solenoid valve is labelled as “2” on the image located to the right.

Figure 2



4.2.3 View Past Data

This Final Feature is located on the top left of the output which is shown in figure 1 and it is used to view the past graphs and specific gravity data taken by the product. All of the data stored will be sent to a cloud that can then reopen the data by clicking on the date of the data needed. After pressing the past data box, a drop down will appear with the most recent dates as well as a calender that will show all dates that data has been stored on. After clicking on a set date, a page just like figure one will appear but with completed data and a completed graph.



5 Troubleshooting & Support

5.1 Error Messages or Behaviors

5.1.1 Load Cells with Arduino

When calculating the mass using the load cells, we initially had issues when taring the load cell. This was just on the first use of the load cell, once we had it set up and had the right code, it worked well every time we ran it. If this error with the arduino and load cell occurs again, there were videos that we found to correct it. These links are in the bibliography on this document. [2]

5.1.2 Peristaltic Pump

Another issue we encountered was when trying to run the peristaltic pump with the arduino. In order for this product to be usable in a brewery it would need to be fully automatic. This was something we were trying to solve very close to design day but we didn't have all the parts available. There is a link for the parts that are needed with the arduino to run the pump automatically. [1]

5.2 Maintenance

All the parts that were used for our final prototype have been taken apart and have been returned. But if it was recreated, in order to maintain it, it would be important that every part that is attached to the side wall of the tank needs to be properly secured. Another thing would be that if the prototype is being tested with a liquid other than water, that liquid would need to be flushed out with water before being stored to make sure that that buildup doesn't grow in the tubing. If the product is being stored for a long amount of time, make sure to empty out all water from the tank and the pump to make sure that water doesn't break the arduino and load cell amplifier. Another solution to this would be to make a casing for the load cell amplifier and the arduino so that they are protected in case anything were to happen.

5.2 Support

5.2.1 Arduino Help

If you encounter issues with arduino, there is an arduino help centre available at (https://support.arduino.cc/hc/en-us)

6 Product Documentation

The prototyping stages of this product were broken down into the following subsystems: load cell, peristaltic pump, and reservoir and housing.

6.1 Load Cell

The sensors used in this subsystem are a 20kg load cell weight sensor and a HX711 load cell amplifier. The purpose of the load cell is to measure the mass of the fermenting beer samples by converting forces exerted on it into electric current. The load cell amplifier captures this signal, and converts it to a higher level of electrical signal that can be read by the Arduino.
To create a scale using this load cell, two platforms and spacers were cut out of MDF, and screwed into the load cell. This ensured the load cell was steady, and was capable of taking accurate mass measurements. The wires in this load cell were soldered to the amplifier, and the amplifier was connected to the Arduino.

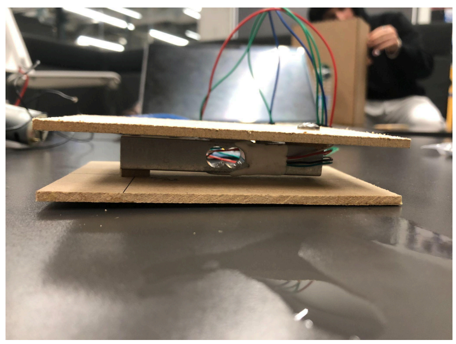


Figure One: Load Cell Scale

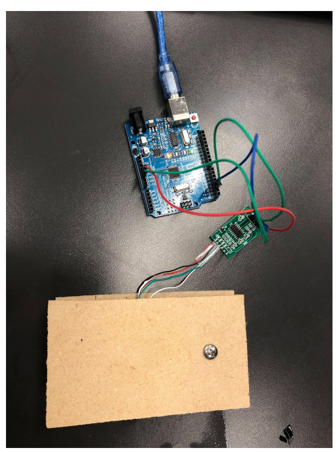


Figure Two: Load Cell and Amplifier Connected to Arduino

6.11 Equipment List

- 20kg load cell weight sensor and a HX711 load cell amplifier

- MDF
- Jumper wires
- Soldering iron
- 2 screws
- Arduino Uno

6.2 Peristaltic Pump

In this subsystem, the peristaltic pump is connected to silicone tubing through which the fermenting beer is pumped. With the use of the peristaltic pump, the beer can be pumped out of the fermenting tank and into the reservoir, and then the flow of the pump can be reversed to pump the beer back into the tank. The pump was connected to and powered by an external battery pack, and controlled by the button. Initially, the plan for the final prototype was to connect the pump to the Arduino, and control and reverse it with Arduino software. However, due to time and material restrictions, in the final prototype, the pump was controlled manually.

6.2.1 Equipment List

- Peristaltic pump
- Silicone tubing
- Water
- Containers

6.3 Reservoir and Housing

The material used for the reservoir is a plastic water bottle. This was chosen because in the conceptual design, the reservoir is depicted as a cone with the tubing at the bottom, to ensure that the beer is fully drained when returned to the fermenting tank. The water bottle was an easily available and affordable option that had a cone shape. One of the main considerations of this system was a pressure release system. In the actual design, this issue could be resolved using a spunding valve, but for the prototype, this was emulated by using a balloon. To create this pressure release system, a hole was made at the top of the water bottle, and a small piece of silicone tubing was put through the hole, and sealed. The balloon was fitted around this tubing, which was secured by sealant.

The next part of this system is a 3D printed base for the water bottle. This is a cylinder shape that the water bottle fits into, so that it can sit on the load cells. It has holes on the bottom for the tubing to go through. This model was designed on onshape, using the dimensions of the water bottle.

Figure Three: Housing of Water Bottle

A box was created to act as the structure that houses the system and would be attached to the fermentation tank in the final product. This box was designed using Inkscape. This design was laser cut out of MDF, and glued together. Creating this box allows us to have an outer layer to store all pieces of our product inside. This box will be used to store the Arduino, pump, and reservoir.

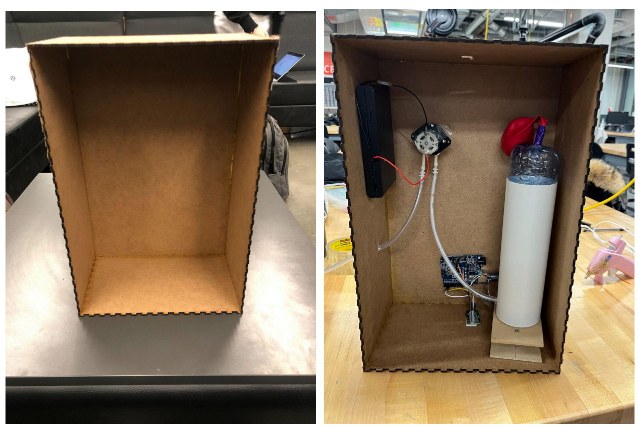


Figure Four: Housing for System

6.3.1 Equipment List

- Plastic water bottle
- Balloons
- 3D printer and filament
- Laser cutter
- MDF
- Wood glue

6.4 Bill of Materials

Software				
Part Name	Description	Quantity	Cost	Link
Arduino IDE	Software to write and compile code, and send it to the Arduino board.	1	Free	https://www.arduino.cc/en/software
HX711 Library	Library in Arduino software to process load cell input.	1	Free	https://github.com/bogde/HX711
Hardware				
Part Name	Description	Quantity	Cost	Link
Arduino Uno	Microcontroller	1	$17.00	https://makerstore.ca/shop/ols/products/arduino-uno-r3
Load Cell and Load Cell Amplifier	Measures the weight of the substance and converts to electrical current.	1	$15.99	https://www.amazon.ca/Weight-Electronic-Kitchen-Weighing-Geekstory/dp/B079FQNJJH/ref=sr_1_5?crid=72MRVMVQEYFQ&keywords=load+cell&qid=1670471812&sprefix=load+cell%2Caps%2C104&sr=8-5
USB Cable	Connects Arduino to computer	1	$1.70	https://makerstore.ca/shop/ols/products/usb-type-a-to-usb-typeb-cable-3ft
Product				
Part Name	Description	Quantity	Cost	Link
Silicone Tubing	Tubing to transport fermenting beer to and from the side tank.	1	$9.48	https://www.amazon.ca/**Easter-Promotion-Month】**-Transparency-Bioengineering/dp/B08RJ8HNT2/ref=zg_bs_12572497011_sccl_8/130-6880706-4331446?pd_rd_i=B08RJ8HNT2&th=1
Peristaltic Pump	Controls flow of beer into and out of the side tank.	1	$15.00	https://edu-makerlab.odoo.com/shop/product/peristaltic-pump-77#attr=
Battery Pack	External power source for peristaltic pump.	1	$10.76	https://www.amazon.ca/8-Slot-Battery-Holder-Plastic-Storage/dp/B07H93ZWQT/ref=sr_1_17?crid=T1S418QYJC2Q&keyw

				ords=battery+pack+with+wires+AA+batteries&qid=1670471910&sprefix=battery+pack+with+wires+aa+batterie%2Caps%2C96&sr=8-17
Jumper Wires	Wires for connecting Arduino	3x10	$3.00	https://makerstore.ca/shop/ols/products/jumper-cables-per-10
Subtotal Cost: $72.93				
Total Cost: $82.19				

6.4 Testing and Validation

6.4.1 Load Cell

The first step in testing the load cell is to calibrate the sensor. To do this, the HX711 library on Arduino IDE is needed. Using the calibration library, the load cells were calibrated using a known weight that was confirmed with a kitchen scale. Once the load cell was returning the same weights as the kitchen scale, it was calibrated correctly.

6.4.2 Peristaltic Pump

To test the peristaltic pump, the tubes were connected to the pump. One end was submerged in a container full of water, and the other in an empty container. The functionality and reversibility of the pump was tested by pumping the water from one container, to the other, then reversing the flow and pumping it back.

6.4.3 Final Testing

The final testing of the prototype tested all of the subsystems together, with an objective of obtaining accurate specific gravity measurements. One thing to consider before this testing took place was how to measure the volume inside the water bottle. Ideally, the volume could be known using an ultrasonic sensor, that would be situated at the top of the reservoir, and measure the volume of liquid entering the reservoir. Due to time and material constraints, an alternative way of determining volume had to be found for the purpose of the prototype. To do this, the flow rate of the pump was measured. This was measured by pumping water into the reservoir that was situated on a kitchen scale, and noting the change in mass of the reservoir over a thirty second period. This test was executed with multiple trials, and then the average flow rate of 1.08mL/sec was found. In the arduino code, the average pump flow rate was inputted as the volume, and the mass was taken from the load cells every five seconds. These values were combined in the density equation d=m/v, to find the density of the liquid. This density value was then divided by the density of water, to find specific gravity.

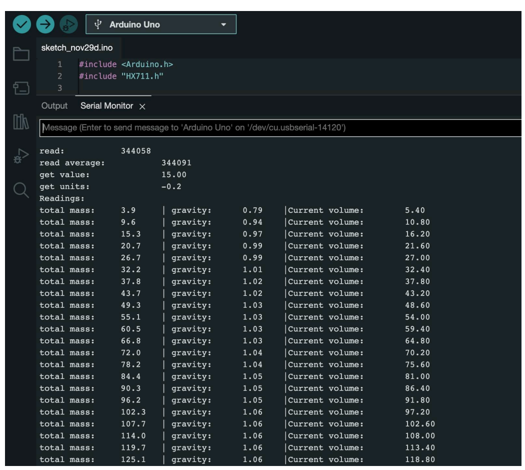
To test the validity and accuracy of the system, the code ran while water was pumped into the water bottle. The specific gravity value that was outputted by the code was very close to 1, which is the expected value. The system was then tested with apple juice, which gave a higher specific gravity than water, which is to be expected. The accuracy of the system was significantly impaired because the flow rate was used to measure volume, which is not an accurate means of measuring. If an ultrasonic sensor was used, the accuracy would improve.

6.4.3.1 Specific Gravity Code

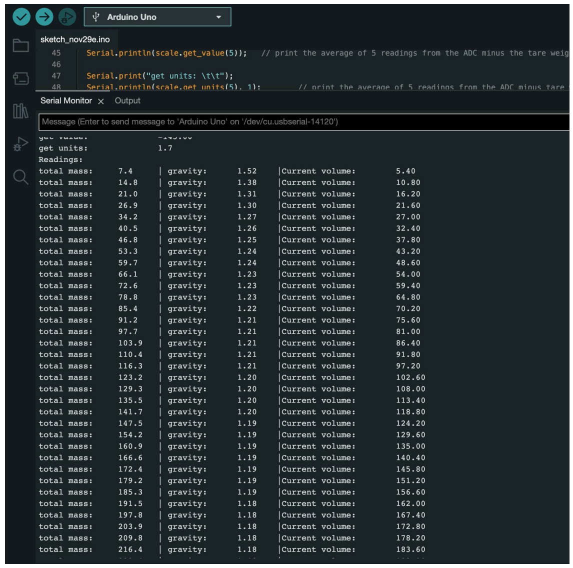
```
#include <Arduino.h>
#include "HX711.h"


// HX711 circuit wiring
const int LOADCELL_DOUT_PIN = 2;
const int LOADCELL_SCK_PIN = 3;
float curVolume = 0;

HX711 scale;

void setup() {
  Serial.begin(57600);
  Serial.println("HX711 Demo");
  Serial.println("Initializing the scale");
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
  Serial.println("Before setting up the scale:");
  Serial.print("read: \t\t");
  Serial.println(scale.read());      // print a raw reading from the ADC
  Serial.print("read average: \t\t");
  Serial.println(scale.read_average(20));   // print the average of 20 readings from the ADC
  Serial.print("get value: \t\t");
  Serial.println(scale.get_value(5));   // print the average of 5 readings from the ADC minus the tare weight (not set yet)
  Serial.print("get units: \t\t");
  Serial.println(scale.get_units(5), 1);  // print the average of 5 readings from the ADC minus tare weight (not set) divided
  scale.set_scale(-100.1);                      // this value is obtained by calibrating the scale with known weights; see the README for details
  scale.tare();               // reset the scale to 0
  Serial.println("After setting up the scale:");
  Serial.print("read: \t\t");
  Serial.println(scale.read());                 // print a raw reading from the ADC
  Serial.print("read average: \t\t");
  Serial.println(scale.read_average(20));       // print the average of 20 readings from the ADC
  Serial.print("get value: \t\t");
  Serial.println(scale.get_value(5));   // print the average of 5 readings from the ADC minus the tare weight, set with tare()
  Serial.print("get units: \t\t");
  Serial.println(scale.get_units(5), 1);        // print the average of 5 readings from the ADC minus tare weight, divided
  Serial.println("Readings:");
}
```

```

float get_gravity(float mass, float volume){
  float density;
  density = (mass/volume);
  return density;
}

float update_volume(float v){
  v = v + (1.08)*5;
  return v;
}

void loop() {
  float result;
  float curMass;
  float plato;
  delay(5000);
  Serial.print("total mass:\t");
  Serial.print(scale.get_units(), 1);
  Serial.print("\t| gravity:\t");
  //Serial.println(scale.get_units(10), 5);
  //do the calculation of density here
  curMass = (scale.get_units(10));
  curVolume = update_volume(curVolume);
  result = get_gravity(curMass,curVolume);
  Serial.print(result);
  Serial.print("\t|Current volume:\t");
  Serial.print(curVolume);
  plato = (result/4)*100;
  Serial.print("\t|Plato:\t");
  Serial.println(plato);
}
```

6.4.3.1 Specific Gravity Code Output



Specific Gravity of Water Output



Specific Gravity of Apple Juice Output

7 Conclusions and Recommendations for Future Work

This User Manual provides all the information needed to understand and recreate our product. In our final prototype, we had the pump functional but not automatically. The load cell was able to find accurate mass measurements and we were able to use flow rate to find the volume and calculate the specific gravity of the liquid we were testing with.

If we had more time to work on this project, an important thing to complete would be the addition of an ultrasonic sensor to find and recreate an exact volume inside our tank. We abandoned this task because we didn't have enough time to add this to our prototype. We also were trying to run our pump automatically so that you wouldn't have to manually turn on/off the pump and reverse it. Instead, the code would step the pump once it had filled to the proper volume and pump it back out once the calibration had completed. We were unable to complete this aspect of our prototype because we didn't have the necessary parts for the arduino. However, given more time, for future groups this would be a key part of the product to work on. Finally, creating a sturdier tank than a water bottle would be needed when keeping a constant volume.

8 Bibliography

[1]

Osmithy, and Instructables. “Automatically Water Your Small Indoor Plant Using Arduino + Pump.”*Instructables*,Instructables,16Oct2017, https://www.instructables.com/Automatically-water-your-small-indoor-plant-using-/.

[2]

Shoeffler, Michael, director. *Tutorial: How to Make a Weighing Scale with the HX711, a Load Cell and an Arduino | UATS A&S #14*. *YouTube*, YouTube, 9 Dec. 2017, https://www.youtube.com/watch?v=S12Mp8gDJmI. Accessed 7 Dec. 2022.

APPENDICES

9 APPENDIX I: Design Files

https://makerepo.com/ChristopherAlRahi/1336.gng1103b04group16beergoggles

Table 3. Referenced Documents

Document Name	Document Location and/or URL
Water Bottle Holder Design	https://cad.onshape.com/documents/8397584180e772e780efa930/w/c3d99a6c5612a0bbc01bd64e/e/4437f7ea3a5e4323f2502ab4?renderMode=0&uiState=63915bcefc417240dc52f5d7