

GNG1103
Design Project User and Product Manual

BREWERY UNO SYSTEM

Submitted by:

Team Brewery Uno, B1

Jasem Alenezi, 300230705

Juliana Barbieri, 300294242

Hiba Dahrabou, 300313246

Patrick Feraday, 300175321

Keenan Yiptong, 300312068

December 7, 2022

University of Ottawa

Table of Contents

Table of Contents	i
List of Figures	ii
List of Tables	iii
1 Introduction.....	1
2 Overview.....	2
2 Troubleshooting & Support	6
2.1 Error Messages or Behaviors	6
3 Final Prototype Documentation	6
3.1.1 BOM (Bill of Materials)	7
3.1.2 Equipment list.....	9
3.1.3 Instructions.....	9
3.2 Testing & Validation.....	11
4 Conclusions and Recommendations for Future Work	11
5 Bibliography	12
APPENDICES	14
6 APPENDIX I: Design Files	14

List of Figures

Figure 1: Final Prototype Hardware	3
Figure 2: Final Prototype Software.....	4
Figure 3: Arduino Setup Diagram.....	10

List of Tables

Table 3. Referenced Documents.....	14
------------------------------------	----

1 Introduction

This is the User and Product Manual for the Brewery Uno System. It is meant to provide context, instruction, and resources for the replication and improvement of this brewery instrumentation project. It will begin with an overview of the project design in its current state, detailing what the device is, what its purpose is, how it functions, and how it fits together. After a short summary of potential troubleshooting required for the project, it will then present descriptions of each aspect of the final prototype and what each part of the prototype was meant to test and represent. It lists a Bill of Materials needed for the prototype, tools used in the creation of the prototype, instructions for replicating the more complex aspects of the prototype, and links to necessary sources and files used in the process in both the bibliography and the project's MakerRepo repository.

2 Overview

Beyond the Pale Brewing Company is an independent microbrewery located in Ottawa, Ontario. They have recently begun expanding their personnel and facilities which has caused their previous method of measuring specific gravity, manually via hydrometer or refractometer with an extracted sample of fermenting liquid at regular intervals, to become unfeasible with the increasing scale of their process. As such, the company has contracted the GNG1103 Engineering Design class at the University of Ottawa to design a device that will allow them to precisely measure the specific gravity of their fermentation process without having to draw time and manpower away from the other stages of their brewing process. The briefing that follows will document the raw user data collected from an interview held with a representative from Beyond the Pale and how that data was interpreted, organized, and converted into a specific problem statement. It will also discuss the design criteria and specifications derived from these interpreted needs, as well as the user and technical benchmarking performed to obtain a better understanding of what this type of device may look like.

After an initial meeting with the client to discuss their needs and the requirements for the device, we created this problem statement that outlines the primary, fundamental needs of the user:

“Beyond the Pale Brewing Company requires an attached device that displays real-time data remotely, is easy to use, and can be removed for cleaning to precisely and automatically measure the specific gravity of their fermentation process.”

For our design, we decided to focus on the data display aspect of the device that the client asked for. We focused on this aspect because without a strong, easy to use and interpret system for displaying the collected data, the value of the specific gravity and temperature sensors and the data they collect is lost. A webpage that could be opened on the user’s laptop, which is connected wirelessly via Bluetooth to each of the devices in the tanks would probably a single, centralized location where all data from every fermentation tank could be viewed either comprehensively or with a focused view on an individual tank, which would make the monitoring process much easier than having a simpler display screen attached to each tank individually. The decision to make a mostly software-based data display system also reduced the complexity of the installation for each tank, as only the shell containing the sensors, Arduino Uno, and Bluetooth module would need to be installed. Images of the components of the final prototype can be found in the following two figures.

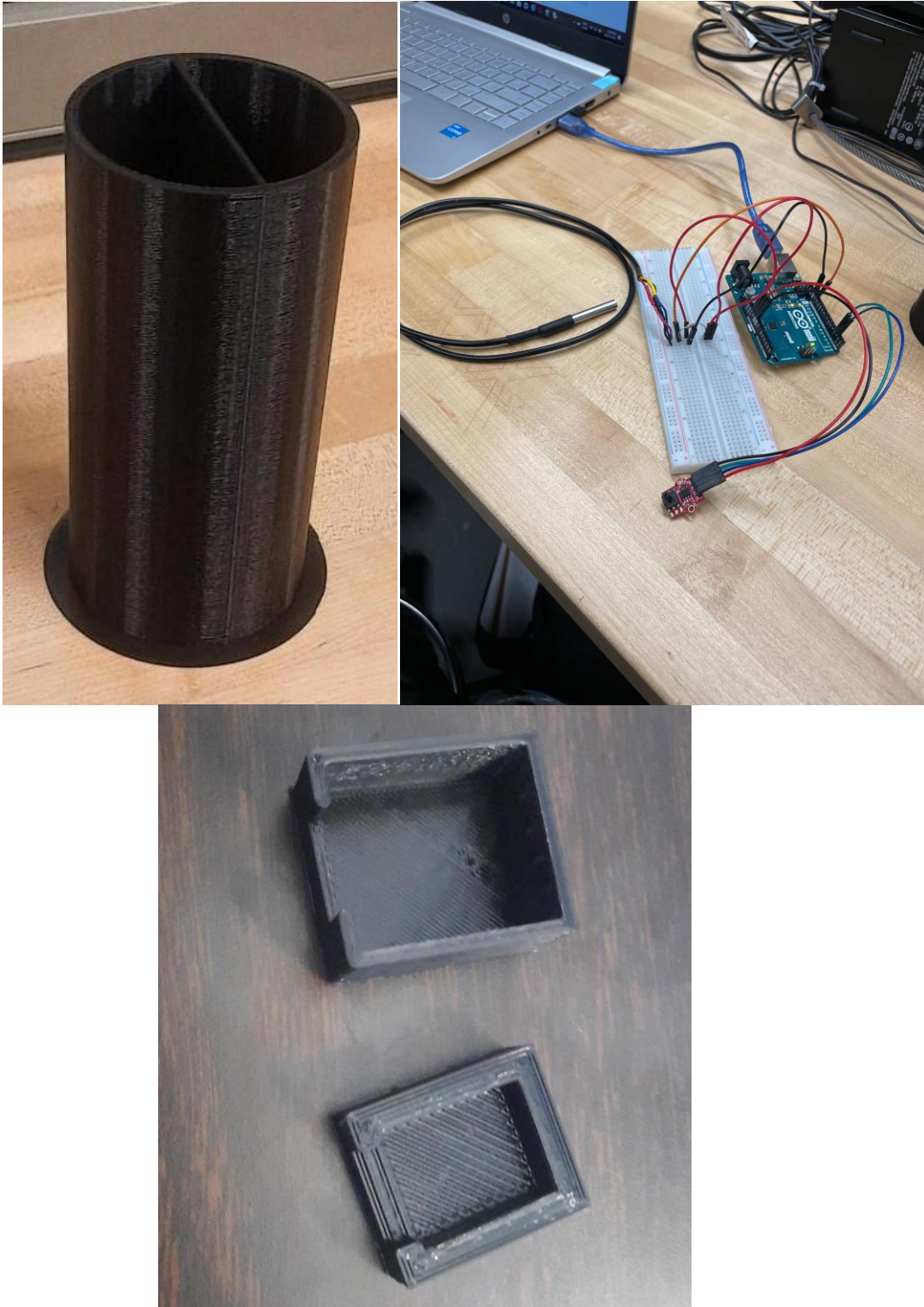


Figure 1: Final Prototype Hardware

For the hardware side of our project, the device consists of a temperature probe and pressure sensor connected to an Arduino Uno that's housed within a cylindrical shell. The purpose of the shell is to seal off the electronics from the fermenting liquid and allow for easier installation of the device into the tank. It will feature ports on one end for the sensors sealed with compression fittings. The Arduino Uno is the main component that runs the sensors and then exports the collected data to the data display via the Bluetooth module. The temperature probe is a standard sensor that will measure the temperature in the tank, allowing the user to track its value. The pressure sensor is the component that measures the specific gravity. By measuring the pressure in the tank, we can derive the fermenting liquid's density and then find its specific gravity from that.

For the software side of the project, the data collected over a specific time period will be sent to the data display where it will be logged in an excel sheet that can be uploaded to the webpage, as well as displayed in a graph that will update directly onto the web page. The web page features a display of all fermentation tanks and their most recent temperature and specific gravity reading, and each tank can be selected for a more detailed display with a list of readings and a graph generated from the measurements. A search bar is included to allow the user to look up the recorded values at a specific time in the process.

The device shell prototype is made of plastic via 3D printing, but the final product would likely be made of a sturdier food grade material such as stainless steel, it will have space inside it to house the Arduino Uno and the wiring necessary to connect all the electronics together. On the end facing the outside of the tank there will be a connection point for the tri-clover clamp used on the fermentation tanks, and on the end facing the inside of the tank there will be ports for the sensors sealed with compression fittings. The sensors will be installed in those ports, the temperature probe will insert through the port so that the metal part of the probe is in the fermenting liquid, and the pressure sensor will be contained within a waterproofing case with a longer tube attached to its sensor that feeds into the liquid.

The data display is an HTML webpage that features two main displays: comprehensive and detailed. The comprehensive display features a list of all the tanks connected to the system, as well as boxes for each connected tank that display only the current specific gravity and temperature reading in that tank. Each tank can be selected individually to bring up the detailed display related to that tank, including a table of the recorded specific gravity and temperature readings, and a graph displaying the change in those measurements over time. A summarized description of the data display is included below:

Main page: [index.html](#)

- The 4x4 table of all the tanks would have the current specific gravity and temperature readings of each tank.
- The row of 16 buttons below the title will each lead to a separate tank's data. Currently, only "Tank 1" is a functional button which leads to the "Tank1.html" page.
- The table on the left of the page would update every two minutes with the most recent data.

- On the right there is a search function (incomplete) that would allow the user to select a date and hour, and obtain the data for that hour in tabular format. The “Choose File” button demonstrates this by allowing to select a file from the hard drive. (Select “20221120_Tank1.csv” to see the most recent data)
- The “All Tanks” button on the top left leads back to the “index.html” page.

2 Troubleshooting & Support

Detailed below are some errors or problems you may run into primarily with the software component of the final prototype and suggestions for resolving these issues should they occur.

2.1 Error Messages or Behaviors

The following error messages that may appear when running the Python code:

Serial port cannot be accessed/does not exist: Make sure Arduino IDE is closed. Arduino IDE and Python cannot run on the serial port at the same time. Make sure the correct COM# is listed in the Python code when calling the serial port.

Four lines of error message that appear immediately after starting the code, referencing “pressure_”: The Python program expects the Arduino code’s output to be formatted as *Pressure (Pa), space, Temperature (°C), new line*. Occasionally, Python will start collecting the data at *Temperature* instead of *Pressure* and is unable to find a second data point on the same line separated by a space. The code does not yet have a way to deal with this error and displays an error message. To fix this, the code needs to be developed further. A short-term solution can be to press the reset button on the Arduino UNO at the same time as the code starts to run. This error is random; please simply try running the Python code again.

3 Final Prototype Documentation

What in this section is a description of each of the three aspects of the final prototype, mechanical, electrical, and software, and what the intended purpose of each aspect of the prototype is. Also included is a Bill of Materials for the entire prototype, a list of equipment used during the project, instructions for both setting up both the electrical and software components. Files for 3D models of the mechanical aspect can be found in the MakerRepo repository which is linked in the first appendix of this manual.

Software

The software aspect of the prototype is the focus of our design and is therefore the most thorough and important component of the final prototype. It consists of two html pages to emulate the structure of our data display pages, a graph plotter function written in Python that allows the Arduino to export the sensor readings and display them as a graph, and the Arduino code necessary

for running the pressure and temperature sensor inside the tank. This part of the prototype was necessary to prove that the focus of our project’s design was functional and sound. By testing that data could be reliably saved via the software and displayed in a central webpage as a graph and table, we would have a solid structure for the basis of our design.

Electrical

The electrical aspect of the prototype consists of the Arduino Uno and both sensors used in the device, along with the necessary wires and breadboard for setting up temporary circuits for testing. We needed to create this prototype in order to test the reliability of our sensors. This component was also crucial for testing the data display in the software aspect of the prototype, which is the main focus of our design. Without sensors to use to actually measure data, it would not be possible to test whether or not the software prototype actually functioned properly.

Mechanical

The CAD models for the device shell and waterproofing case were initially created from data we had researched rather than off the components themselves. The CAD model for the device shell was based off the design of the project’s first rough prototype, but with more precise measurements relative to the fermentation tank. The waterproofing case for the pressure sensor was similarly designed using the dimensions listed its website where it was purchased from and will likely be modified once the sensor has been received. The sensor code is based off research into examples of code used to run pressure sensors and will be altered and refined once it can be tested practically with our sensor. Later, we were able to refine the dimensions of this case to fit the shape of the pressure sensor more accurately.

3.1.1 BOM (Bill of Materials)

Materials needed		Cost (\$)	
1	Arduino Uno	9.00	https://edu-makerlab.odoo.com/shop/product/arduino-5#attr=5
2	Temperature Probe	14.00	https://edu-makerlab.odoo.com/shop/product/grove-temperature-sensor-47#attr=
3	Jumper Wires	0.10	https://edu-makerlab.odoo.com/shop/product/jumper-wires-44?search=jumper+wires#attr=46

4	Bluetooth Module	22.09	https://edu-makerlab.odoo.com/shop/product/bluetooth-module-9#attr=254
5	USB Cable A-B	7.00	https://edu-makerlab.odoo.com/shop/product/usb-cable-68?search=USB+cable#attr=80
6	Fluid Pressure Sensor	10.99	https://www.canadarobotix.com/products/2760?variant=39439020294193&currency=CAD&utm_medium=product_sync&utm_source=google&utm_content=sag_organic&utm_campaign=sag_organic
7	Ultrasonic Range Finder	4.00	https://edu-makerlab.odoo.com/shop/product/ultrasonic-sensor-60#attr=
8	Circuit Breadboard	2.50	https://edu-makerlab.odoo.com/shop/product/breadboard-53?search=breadboard#attr=58
9	Resistor	0.01	https://edu-makerlab.odoo.com/shop/product/resistor-6?search=resistor#attr=20
10	3D Print Material	10.00	
11	3" sheet metal tube	0	
12	Plastic wire mesh stand-in	0	
13	Cardboard	0	
14	Scotch tape	0	
15	Plastic Specific Gravity sensor stand-in	0	
16	Webpage visualization software	0	
Total		79.69	

3.1.2 Equipment list

Item Name	Description	Type	Source
CAD (Computer Aided Design) software (Onshape)	To design 3D models	Analytical (Software)	https://www.onshape.com/en/
3D printer	To 3D print design 3D models.	Equipment.	MakerSpace
Coding Software (IDLE, Python, Arduino IDE, and JavaScript code)	To implement code.	Software	Personal device
Arduino UNO microcontroller and parts (USB cable, jumper wires, resistor, breadboard)			
Temperature sensor			
Pressure sensor			

3.1.3 Instructions

Arduino Sensors

1. Install Arduino IDE.
2. Install the HX710B, OneWire and DallasTemperature libraries.
3. Connect Arduino UNO microcontroller to laptop using USB cable. Note the COM#.
4. Connect wires and sensors using the following diagram:

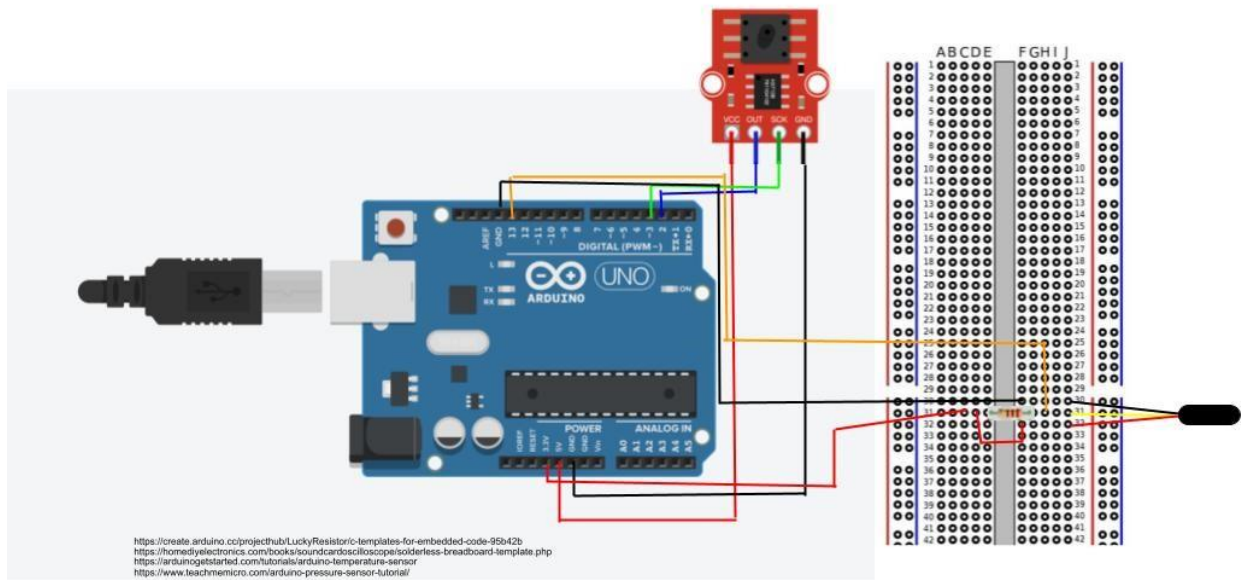


Figure 3: Arduino Setup Diagram

5. Upload the “Temp and Pressure Arduino code” in Arduino IDE.
6. Run the code. Output on the Serial Monitor should be a pressure reading in Pascals followed by a temperature reading in degrees Celsius on the same line, separated by a space.

Display Interface

1. Install IDLE and Python, as well as the PySerial and Matplotlib.pyplot libraries (you may need to install pip beforehand; numpy should install automatically with Python). Download the “myplot.png” and “20221120_Tank1.csv” files.
2. In IDLE, open the “Serial test sensor output” Python file.
3. Close Arduino IDE.
4. Select “Run Module” in the “Serial test sensor output” Python file.
5. If error messages appear, see ‘Error Messages or Behaviors’.
6. Ten pairs of measurements will be taken in 1 second intervals. An IDLE page will open to show this (there should also be a timestamp for each set of measurements).
7. Open Tank1.html. The most recent graph of the sensor data collected will be displayed. (The page may need to be reloaded first.)
8. Click “Choose file” and select the “20221120_Tank1.csv” file from the laptop’s hard drive. The table will display the data from the most recent sensor test.

3.2 Testing & Validation

The data display interface creation and testing's key outcomes were feedback, trial, and error. The original data display prototypes were recreated using HTML. Then, to obtain an updating graph, it was established that several measurement intervals would need to be added onto the same graph as the fermentation runs, and they would need to be saved onto a single file, with each new interval rewriting over that file. Most of the feedback was gathered during this phase. The data from the Arduino must be written to a CSV file, saved, then read and entered the interface to receive an updated data table. Python and JavaScript code were used to do this. The timestamp in the data is now generated by Python rather than Arduino; it was much easier and more straightforward. Finally, we began testing the blue fruit Bluetooth module for Arduino by downloading the iPhone app and successfully connecting the module. The temperature and pressure sensors were tested, and the tests consisted of obtaining several readings from each sensor and comparing them to a reference value. We tested the pressure sensor in the water of a known height and utilized the pressure data to calculate the density of the water, which we then compared to the known density value of water. The sensor did not appear accurate because of this test; the known water density is 997 kg (about 2198.01 lb.)/m³, but our computations returned inaccurate density readings. We were not sure if we were simply misusing the sensor because we could only dip the sensor's peg into the water from the top, but hydrostatic pressure is measured from the bottom of a container. To correct this, we ran another test, this time wrapping the sensitive sections of the sensor with a plastic bag, inserting the peg into a water bottle, and inverting it. This test did not yield a correct density, with the final density value of 474.29 kg (about 1045.63 lb.)/m³. We are unsure whether this issue is caused by the quality of our sensor or by user error. We had the temperature sensor scan the room's ambient temperature and compare its data to another thermometer reading the same temperature. Because of this test, the probe was 3°C close to the thermometer. This result is satisfactory, but we would like to do additional tests to confirm the accuracy and, if possible, improve it.

4 Conclusions and Recommendations for Future Work

Working on this project, our team certainly learned an important lesson about prioritization and time management. In order to meet deadlines, some aspects of the design did not receive as much attention as they likely should have in order to fully develop a working product. We decided to shift our focus onto the data display as we felt it was the most important and unique aspect of our design, but the mechanical and electrical aspects such as the device shell and the sensors would benefit greatly from extra work. Most notably, if we were to continue work on this project, we would focus on improving the pressure sensor, and our method for deriving specific gravity as a whole. Opting for a higher quality sensor that was more accurate and wasn't as difficult to get working in liquids would be a great improvement and measuring density from differential pressure with two sensors rather than hydrostatic pressure with only one would also greatly boost the quality of the design.

5 Bibliography

Adafruit Bluefruit LE Desktop Application. (2022, November 24). GitHub.
<https://github.com/adafruit/adafruit-bluefruit-le-desktop>

Amazon.com: Customer reviews: PLAATO Airlock V3 - WiFi Fermentation Analyzer for Homebrewing.
Retrieved October 2, 2022, from <https://www.amazon.com/PLAATO-Airlock-Fermentation-Analyzer-Homebrewing/product-reviews/B07KQPKSG5>.

Canada, P. S. (2022, August 4). *Power outages - what to do?* Public Safety Canada. Retrieved October 9, 2022, from <https://www.getprepared.gc.ca/cnt/rsrscs/pblctns/pwrtgs-wtd/index-en.aspx>

Change Tick Frequency in Matplotlib. (2020, October 28). Stack Abuse.
<https://stackabuse.com/change-tick-frequency-in-matplotlib/>

Digital Pressure Sensor– Arduino Workshop. (n.d.). Arduino Project Hub.
<https://create.arduino.cc/projecthub/sarful/digital-pressure-sensor-arduino-workshop-0e43ae>

FAQs (Frequently Asked Questions) hardware - plaato airlock. Plaato Help Center. (n.d.). Retrieved October 9, 2022, from <https://intercom.help/plaato/en/articles/5004647-faqs-hardware-plaato-airlock#:~:text=No%2C%20the%20Plaato%20Airlock%20needs,provides%20minimum%205V%20and%202amps>

Installation — Matplotlib 3.5.1 documentation. (n.d.). Matplotlib.org.
<https://matplotlib.org/stable/users/installing/index.html>

Introducing the Adafruit Bluefruit LE UART Friend. (n.d.). Adafruit Learning System.
<https://learn.adafruit.com/introducing-the-adafruit-bluefruit-le-uart-friend/wiring>

Jabbaar, A. A. (2019, September 17). *Ultrasonic Sensor HC-SR04 with Arduino Tutorial*. Arduino Project Hub. <https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6>

JavaScript - How to format CSV file to display like a table in html. (n.d.). Stack Overflow. Retrieved December 8, 2022, from <https://stackoverflow.com/questions/48999620/how-to-format-csv->

- Kazarinoff, Peter D. *Reading a Sensor with Python - Problem Solving with Python*. (n.d.). Problemsolvingwithpython.com. Retrieved November 14, 2022, from <https://problemsolvingwithpython.com/11-Python-and-External-Hardware/11.04-Reading-a-Sensor-with-Python/>
- Liechti, C. (n.d.). *pyserial: Python Serial Port Extension*. PyPI. Retrieved December 8, 2022, from <https://pypi.org/project/pyserial/#files>
- Probert, Brad. (2020, December 28). *Hands on review: Plaato Digital Airlock – WIFI fermentation analyzer for homebrewing*. Homebrew Finds. Retrieved October 2, 2022, from <https://www.homebrewfinds.com/2019/02/hands-on-review-plaato-digital-airlock-wifi-fermentation-analyzer-for-homebrewing.html>
- python - Changing the tick frequency on the x or y axis*. (n.d.). Stack Overflow. Retrieved December 8, 2022, from <https://stackoverflow.com/questions/12608788/changing-the-tick-frequency-on-the-x-or-y->
- Python Software Foundation. (2020). *csv — CSV File Reading and Writing — Python 3.8.1 documentation*. Python.org. <https://docs.python.org/3/library/csv.html>
- Python String split() Method*. (n.d.). https://www.w3schools.com/python/ref_string_split.asp
- Vishal. (2021, July 4). *Timestamp In Python*. PYNative. <https://pynative.com/python-timestamp/>

APPENDICES

6 APPENDIX I: Design Files

Below is a table of relevant files related to the project. All of them can be found and downloaded from the MakerRepo repository linked in the table. The Specific Gravity Device Technical Document is a comprehensive document detailing everything the original team did for the design process of the project. It contains a detailed analysis of client needs, various stages of prototyping and test results, and feedback from the original client. Part Studio 1 (1).stl is the stl file for the CAD model of the device shell used for 3D printing. Part Studio 1 – Part 1.stl and Part Studio 1 – Part 2.stl are the stl files for the two parts of the waterproofing case CAD model used for 3D printing. Serial test sensor output.py is the Python code used to take the sensor readings from the device and translate them into a graph in the data display. Index.html and tank1.html are html pages for the comprehensive tank display and the detailed tank display respectively. Temp and Pressure Arduino code.ino is the combined Arduino code that runs the temperature and pressure sensors in the tank, and Pressure sensor code (1).ino is the separated code for the pressure sensor itself.

Table 1. Referenced Documents

Document Name	Document Location and/or URL
Specific Gravity Device Technical Document	https://makerepo.com/PatrickFeraday/1370.gng1103b1brewery-uno
Part Studio 1 (1).stl	https://makerepo.com/PatrickFeraday/1370.gng1103b1brewery-uno
Part Studio 1 – Part 1.stl	https://makerepo.com/PatrickFeraday/1370.gng1103b1brewery-uno
Part Studio 1 – Part 2.stl	https://makerepo.com/PatrickFeraday/1370.gng1103b1brewery-uno
Serial test sensor output.py	https://makerepo.com/PatrickFeraday/1370.gng1103b1brewery-uno
index.html	https://makerepo.com/PatrickFeraday/1370.gng1103b1brewery-uno
tank1.html	https://makerepo.com/PatrickFeraday/1370.gng1103b1brewery-uno
Temp and Pressure Arduino code.ino	https://makerepo.com/PatrickFeraday/1370.gng1103b1brewery-uno
Pressure sensor code (1).ino	https://makerepo.com/PatrickFeraday/1370.gng1103b1brewery-uno