GNG1103

# Design Project User and Product Manual

## Plato Measuring Device

Submitted by: Group 10 (The Imagineers)

Annalise Wey (300287029)

Emma Williams (300115833)

John Scales (300317435)

Dooshita Dhawka (300251015)

Omar Wahba (300291047)

Submission date: 12/07/2022

University of Ottawa

# Table of Contents

# List of Figures

# List of Tables

# List of Acronyms and Glossary

**Table 1. Acronyms**

| Acronym | Definition |
|---|---|
| S.G. | Specific gravity |

**Table 2. Glossary**

| Term | Definition |
|---|---|
| Accelerometer | Measures the acceleration of a desired object. |
| Fermentation | A biological enzymatic process in which molecules are broken down. An example of this process is beer brewing. |
| Food-grade | A material that is safe to use for the production and processing of food. |
| Hydrometer | An instrument used to measure the specific gravity of a liquid using the principle of buoyancy. |
| Refractometer | An instrument used to measure the specific gravity of a liquid using the principle of index of refraction. |

# 1 Introduction

Our client is Shane Clark, co-owner of Beyond the Pale Brewing Company. Shane has a need for a device to measure the specific gravity and time throughout the brewing process. This device needs to be affordable, non-free-floating, have to ability to be easily cleaned between batches, and be made out of food-grade materials. Shane Clark is the Client however the user varies among a few of his employees. The employees who most often monitor the specific gravity in BTP's brewing process are his head and assistant brewers and, most frequently, his lab technician. Beyond the Pale Brewing company are currently in the midst of an expansion and this device is to improve the efficiency of the brewing process while not impacting the accuracy of alcoholic percentage and uniform taste of the beer produced.

The general concept of this design is an in-unit density meter used to measure density (which is converted into specific gravity). A rod attached to a motor is placed inside the tank though the port. The motor will oscillate the rod at a constant speed then as the density of the brew changes the rod will oscillate at a different speed then an accelerometer will measure the difference and be able to calculate the density. For this to work you must know the starting density which can be measured with the same product and entered by an external device. The motor will be waterproofed by a sealed shell with the one side being a flexible martial to allow movement in the rod. There will be a screen to show the exact value as well as a plotted graph. The device can be programed to record data in any intervals of time. A temperature probe is added into the rod to take tempeerature. All the data recorded is saved and sent to a hard drive. This device is hardwired for power but has a backup battery for when needed which can be use full if power goes out to see if the batch has gone bad. The device will latch into the tank to allow for easy

removal for cleaning. The client wanted an inline device that acutely measures specific gravity and temperature this device will fit all of the client's needs.

# 2 Overview

Shane Clark and Beyond the Pale Brewing Company are in the midst of planning an expansion and this device is to improve the efficiency of the brewing process while not impacting the accuracy of alcoholic percentage and uniform taste of the beer produced. A device to measure the specific gravity inside the tank will minimize the amount of their product being exposed to the air and wasted to do so. The product cannot be free-floating in their tanks and must be easily cleanable between batch processes. The product must also be made of food-grade materials keeping the safety of employees and Beyond the Pale's customers in mind.

Shane Clark would love it if the product not only recorded the specific gravity but the time and could show a graph of the fermentation time as well, however, this is not a necessity it would add extra value to the product in his eyes.

Through empathizing and the first client interview we learned that Shane Clark is the customer but not the user. At Beyond the Pale the part-time lab technician and sometimes the lead/assistant brewers are the users of this requested device. The client's problem is that they would like to be able to measure the specific gravity of the beer in his fermentation process accurately and efficiently without wasting his product. Based on the first client interview the following information and client opinions were discovered.

**Table 3: Empathize Client Needs**

| ***Client needs:*** |
| --- |

| |
|---|
| Something efficient which can monitor specific gravity in real-time (routine measurements every 2-3 min) |
| A product that is easy to maneuver and not time-consuming to use |
| Not expensive: budget around $25k for 16 tanks |
| A product that is easy to clean (the previous device used to often have yeast build-up) |
| Can resist temperatures up to 85-90 degrees (cleaning done at that temperature) |
| Can resist the fermentation process of 2-3 weeks. |
| The product should measure the specific gravity in a sanitary manner. |
| The device's dimension should be about 3inches in width and at least 4 inches in length |
| The device should be of food-grade material |

**Table 4: Empathize Client Would Like**

| ***Client would like:*** |
|---|
| A product with backup batteries. |
| Client mentioned that it would be nice to have the device measure temperature also. |
| A device that manipulates the data collected and generates a curve and stores it forever. (An app with no subscription fee) |
| The client does not mind using Wi-Fi or Bluetooth to connect the device to the main computer. |
| A device whose power consumption is as minimal as possible. |
| It would be cool to monitor the specific gravity remotely. |

**Table 5: Empathize Feature Client Would Not Like**

| ***Features client would not like:*** |
|---|
| The product should not be free floating since they tend to lose the previous device they were using (had to empty the whole tank to get it out. Not a great ideal!) |
| No subscription fee of any kind on the product. |

**Table 6: Empathize Opinions Based on Client's Own Benchmarking**

| ***Opinions based on client's own benchmarking:*** |
|---|
| The client previously used products made of plastic and stainless steel, which did not cause much trouble |
| Refractometer: least accurate |
| Hydrometer: most accurate |

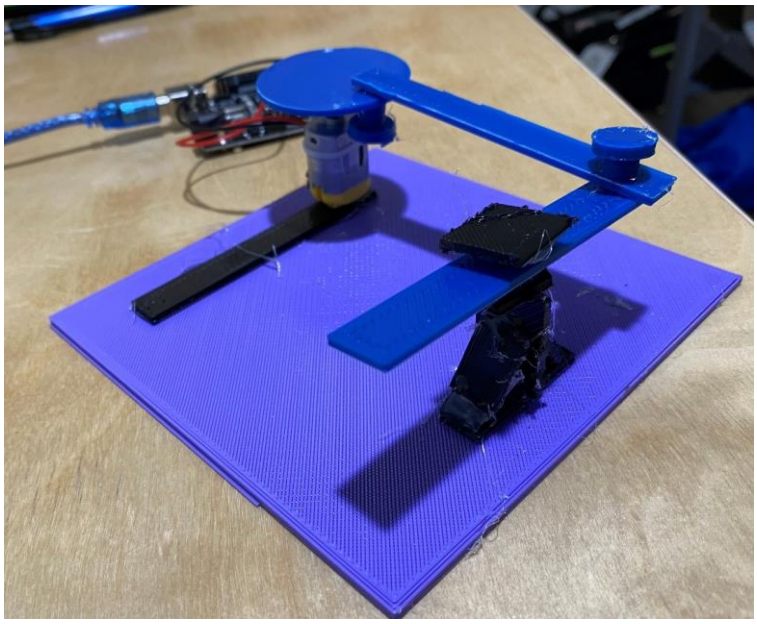


**Figure 1: Final Physical Prototype**

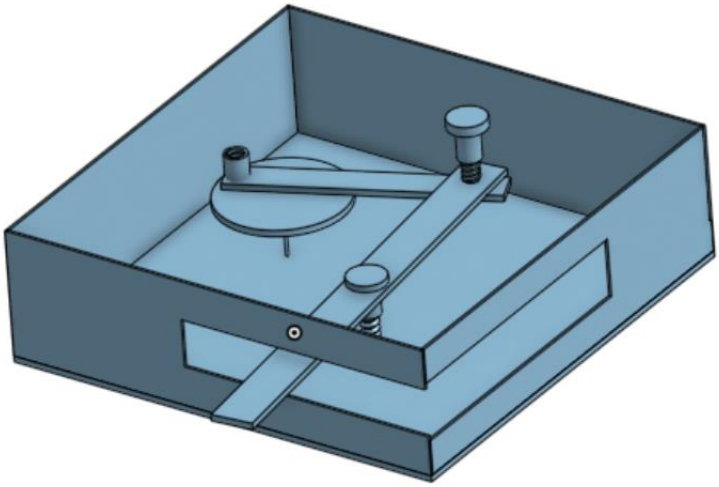


**Figure 2: Final Analytical Prototype**

**Table 2: Simple Analysis of Design**

| ***Critical Components of Design*** | ***Explanation*** |
|---|---|
| Accelerometer | The accelerometer is very important since it is used to measure the change in speed of the oscillating rod and therefore can be used to find specific gravity. |
| Temperature sensor | Although the temperature sensor was not originally critical to the user needs our prototype implements it as a bonus for the user experience of the product. |
| Oscillating rod | The oscillating rod is what the accelerometer measures the movement of during the brewing process. |
| Arduino software | The Arduino software is critical for using all aspects including the sensors. Also for graphing the fermentation curve as Shane requires. |
| Motor | The motor in the process powers the oscillating rod. |

## 2.1 Conventions

Action is required for the user at the beginning of each batch process. The initial values of the batch need to be imputed into the system. This is something that the client said was very possible to implement since they know those values. For this design to work you must know the starting density which can be measured with the same product and entered by an external device. This is the only action required by the client prior to measuring.

## 2.2 Cautions & Warnings

The only cautionary remark for this device is that the product needs to be made of food-grade durable materials. The cleaning process is extensive and for the user to be able to sell the product there cannot be any non-food-grade materials in the fermentation tank.

# 3 Getting started

## 3.1 Configuration Consideration

This device is designed for the workers of breweries. This device is used to measure specific gravity. To set up the rod is inserted though the port and the box is attached to the side of the tank. The box is plugged into a computer to run the program. A screen on the box will display the specific gravity and tempter. The original density must be imputed on the computer before the device can run.

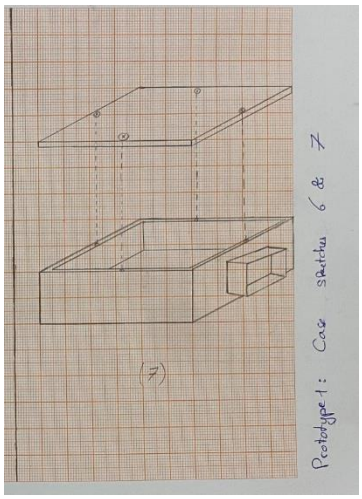
## 3.2 Accessing/setting-up the System

To set up, insert the rod thought the port and sucre the box to the side then plug in the device to a computer and input the original density and run. The computer must be plug into a power sorce and be able to be acceced to imput the original gravity.

## 3.3 System Organization & Navigation

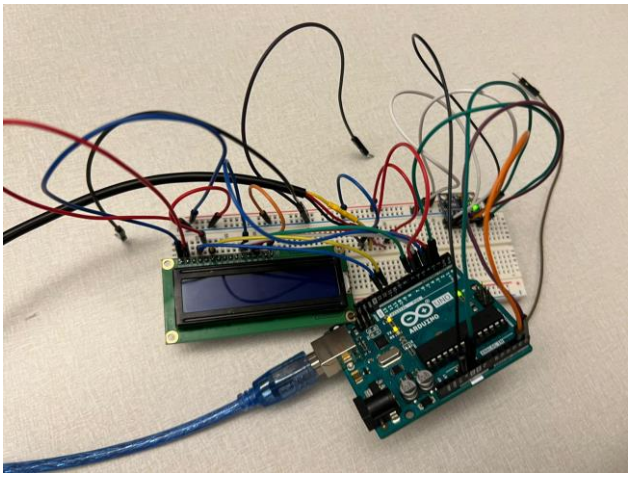
The main parts of the system is  the mechanical components inside the box and the outer box.

1.1




This is the outer box. this will attach to the outside of the tank and hold the mechanical componets as well as a screen to show to data.

1.2



The mechanical components inside the box consist of an Audino uno, a dc motor for the rod, and an lCD screen.

## 3.4 Exiting the System

In order to remove the device to clean simply unplug the device and detach from the side of the tank. All data will be automatically saved to the computer and graphed.

# 4 Using the System

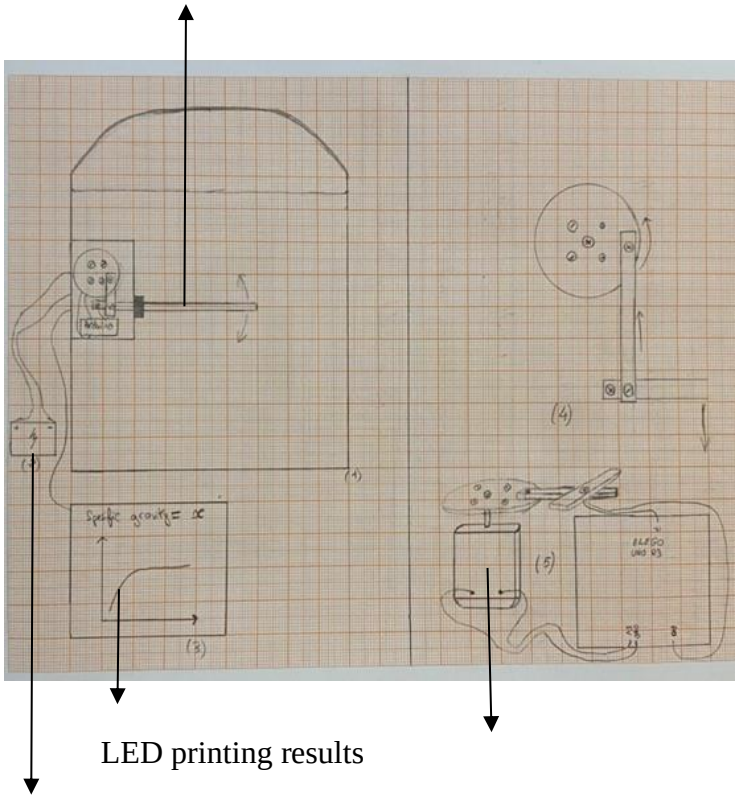



**Figure 3 :sketch of prototype**

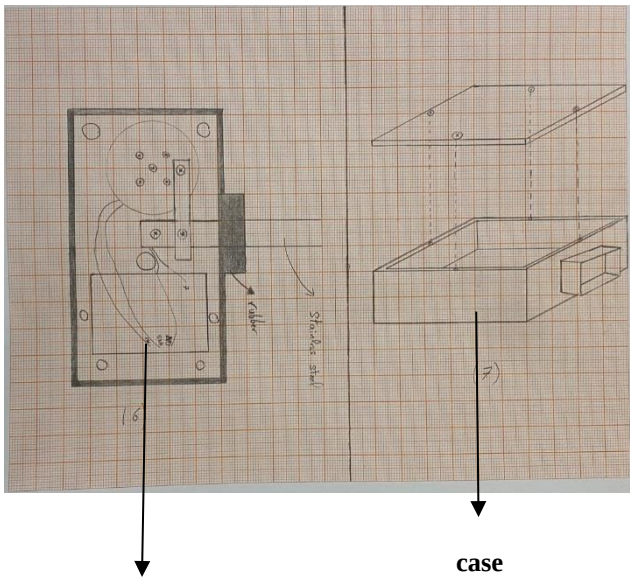



**Figure 4: sketch of device case**

## 4.1 Case with all the components

The case is to waterproof the all the components of the device. This case is to be latched on the inside wall of the tank.

## 4.2 Oscillating rod

The rod will oscillate in the liquid and as fermentation progresses, the speed of oscillation will change, and an accelerometer will measure this change.

## 4.3 LED screen

The LED screen will print out the results and this part of the device will be attached to the outside wall of the tank.

## 4.4 Motor

The motor will be attached to the rod and will oscillate the latter at a constant speed.

## 4.5 Back up Battery

The battery will provide the necessary power to the motor to oscillate the rod.

# 5 Troubleshooting & Support

There are 3 main places where a problem could occur: physical component, Server, and user interface.

## 5.1 Error Messages or Behaviors

Physical:

- If any of the plastic 3d printed Pieces are cracked or damaged, they can easily be reprinted using the cad design provided. Then reassembled using the instructions in the assembly section.
- If the motor or is not working but the Arduino lights are flashing when the program is running, repurchase a new motor from the sources provided in the part list.
- If the sensors are giving bogus readings or not reading at all, repurchase from the sources provided in the part list.
- If no lights turn on on the Arduino when plugged in, test another cord, if it persists, repurchase new Arduino from the sources provided in the part list.

Server:

- If, when opening site for user interface, it displays error 404 server not found, make sure server address in the search bar has been correctly entered. If so, relaunch the server using the launch server instructions.

User Interface:

- If graph is not showing up on the user interface, but the page does show up (it stills says Plato vs time graph), reload page, if it persists, contact customer support by texting “GNG1103 server help” to 613-290-3679.
- If user interface is slow, this is likely because too many data points are being graphed, reload page, if it persists contact customer support.

## 5.2 Special Considerations

If any error or problem occurs that is not covered by the trouble shooting section: contact customer support by texting “GNG1103 server help” to 613-290-3679.

## 5.3 Maintenance

The server should be backed up daily to free up space for new data points and to prevent sluggish user interface caused by too many data points. Currently this can be done by copying the contents of the data Json file in the server files and replacing it with an empty one and restarting the server..

## 5.4 Support

Talk to John Scales, our customer support agent by texting “GNG1103 server help” to 613-290-367 or emailing nhoj8gjs@gmail.com with “GNG1103 server help” in the heading. Our agent will ask questions to try, and narrow done the cause of the problem, and may ask for screenshots or pictures to better understand the problem.

# 6 Product Documentation

Our prototype can be divided into 2 main subsystems: Motor and oscillating rods (Mechanical aspect) and Arduino and wire placements (Coding and electrical aspects). The material to be used needs to be food grade which is why we chose stainless steel. Also since the brewery already had their tanks made in stainless steel we thought using the same type of material would be good for the cleaning process because they're already familiar with it. We would've used a more developed and accurate accelerometer but due to our budget limits we settled for the one we have in our BOM (bill of material).

## 6.1 Subsystem I: Motor and oscillating rods (mechanical aspect)

### 6.1.1 BOM (Bill of Materials)

| **item** | **quantity** | **cost** | **total** | **link** |
|---|---|---|---|---|
| Arduino UNO R3 clone* | 1 | 17 | 17 | https://makerstore.ca/shop/ols/products/arduino-uno-r3 |
| cables (10 pack) | 1 | 1 | 1 | https://makerstore.ca/shop/ols/products/jumper-cables-per-10 |
| motor | 1 | 2 | 2 | https://makerstore.ca/shop/ols/products/compact-dc-hobby-motor |
| OnShape(online software) | N/A | N/A | N/A | https://www.onshape.com/en/ |
| Ultimaker Cura(Software) | N/A | N/A | N/A | https://ultimaker.com/software/ultimaker-cura |
| | | **total** | 20 | |

An oscillating rod in stainless steel would've been included but the budget we had was limited therefore we went with just 3D printing all the parts for the final prototype.

### 6.1.2 Equipment list

Equipment needed to build subsystem:

- 3D printer + printing software (Ultimaker Cura was used in our case)
- Glue gun (Food Grade glue would be preferable but due to the time and availability we had to go with glue gun)
- CAD (computer aided design) software (we used OnShape)

### 1.1.1 Instructions

#### 6.1.2.1 OnShape CAD

##### 6.1.2.1.1 Base

First, start by creating a base. Choose a plane and then sketch a square with 14in sides and then using the extrude button create an extruded version of the base with 0.197in as depth. Then proceed by creating a sketch on the face of the base as the following picture shows using the same measurements.

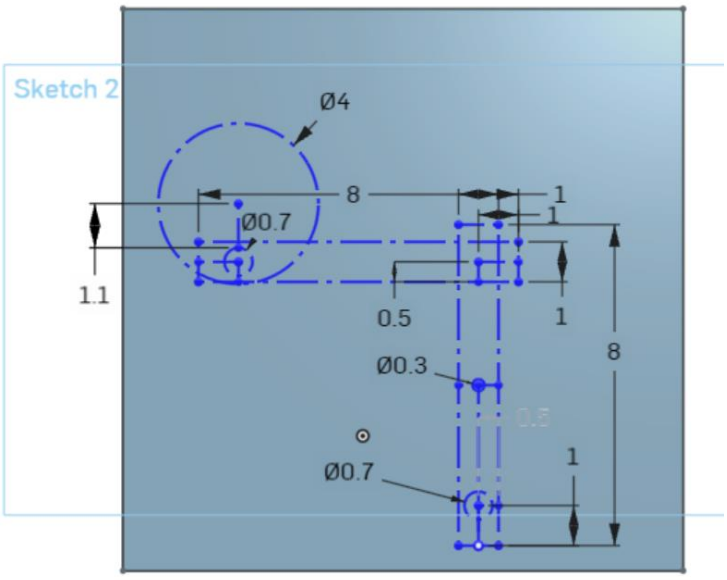



**Figure 5: Sketch on base**

Then extrude the 0.3in circle to a depth of 1.693in make sure to set the parameters to Solid and Add as shown in the photo.

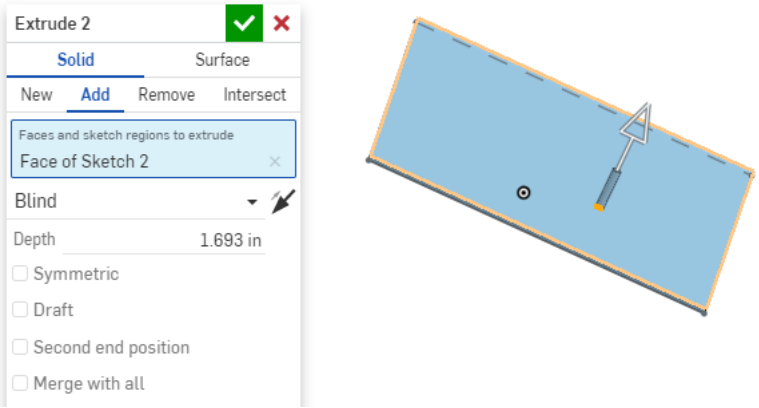



**Figure 6: Properties of extrude**

And then, proceed by drawing a helix using the helix function as shown in the photo, around it to start the process of creating a screw to insert the oscillating rod's hole.

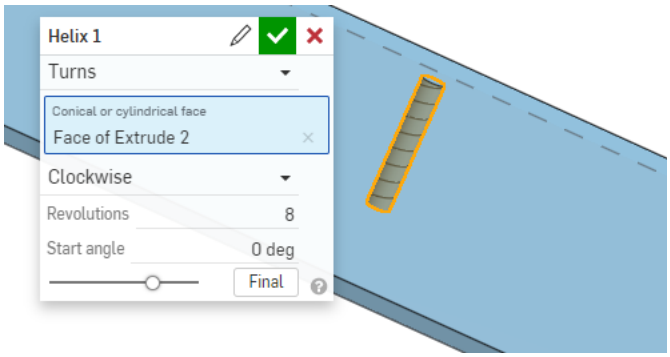



**Figure 7: Properties of helix**

Then proceed by creating a plane that cuts the extruded cylinder in half and then draw an equilateral triangle of 0.15in sides and create a Pierce (highlighted in blue) relation between it and the helix as shown in the photo.

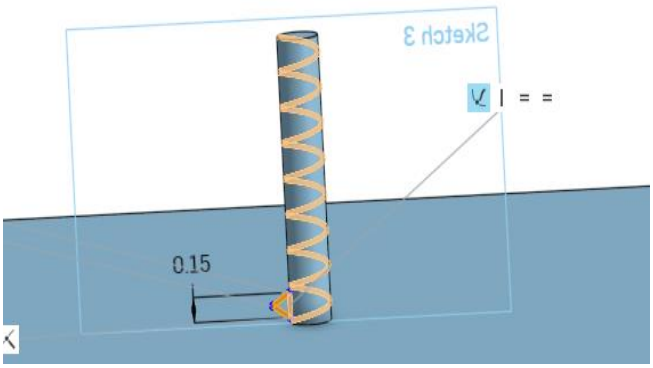



**Figure 8: Create a pierce relation**

Finally, using the Sweep function from OnShape, select the face of sketch of triangle and for the sweep path choose the helix, with the shown parameters in the photo to obtain the desired screw.

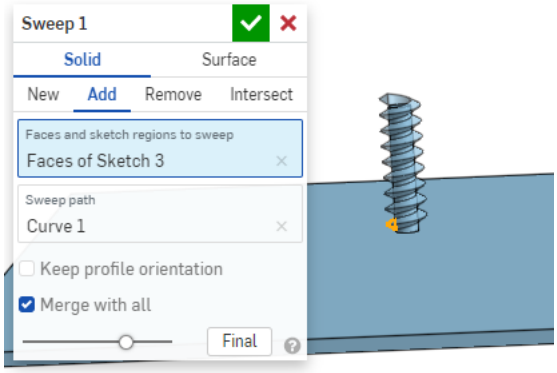



**Figure 9: Sweep properties**

#### 6.1.2.1.2 Rods

For this step, we will build the two rods that will be connected to each other. Starting with the first rod the linear one connected to the oscillating rod and the rotating disk. Start by creating a new Part Studio. Then, select a plane and sketch a rectangle with the following dimensions 8in in length and 1in in width. From the center of both 1in sides draw a 1in construction line from each side, and from the end of each line draw a circle of 0.8in diameter (one of the circles need to be a construction circle) like shown in the photo.

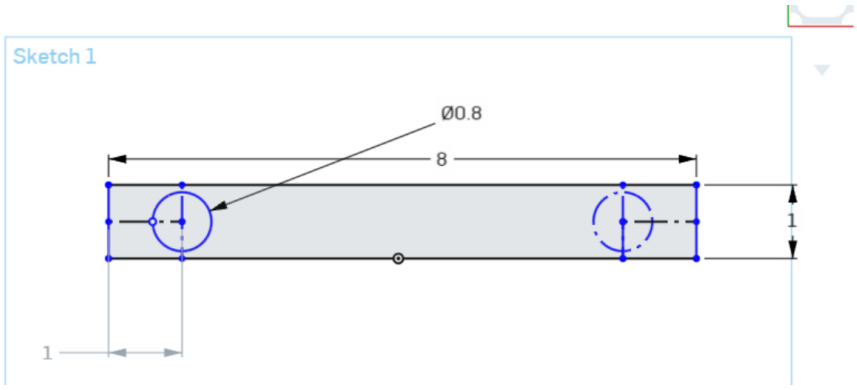



**Figure 10: Sketch of linear rod**

Then extrude the sketch to a 0.197in depth. You're supposed to have a hole in one side and nothing on the other. On the side with no holes, draw a sketch with the same circle drawn except this time don't make it a

construction circle and then extrude it with a 0.5in depth and make sure to select “Add” as shown in the photo.

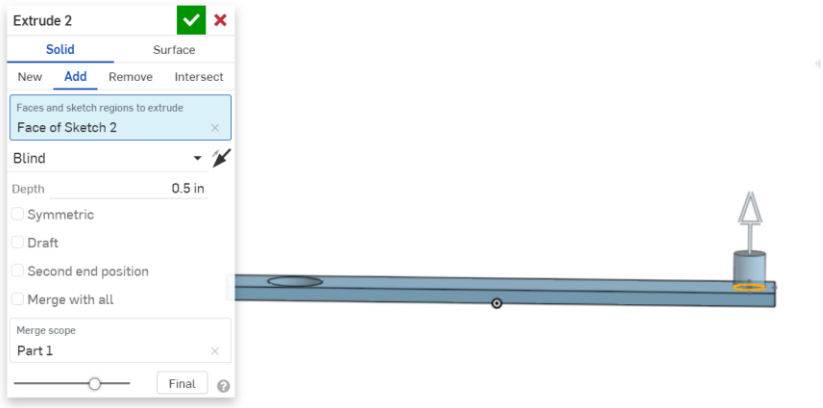



**Figure 11: Properties of extrude (linear rod)**

The next step is creating the helix around the 0.5in cylinder to create the form of a screw. To do this we’ll need to create a helix around it by using the helix function and selecting the face and setting as parameters clockwise and 6 revolutions as shown in the photo.

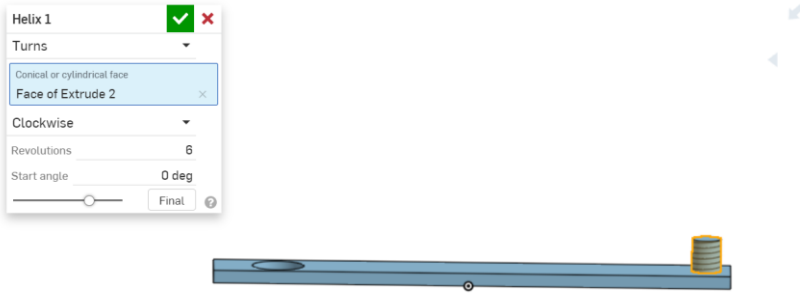



**Figure 12: Properties of helix (linear rod)**

Then we will proceed to sweeping it to make it have the actual 3D look of the screw. To do this we will need to create a plane perpendicular to the rod’s face and cuts the cylinder in half as shown in the photo, to create a sketch of an equilateral triangle with a side of 0.05in.

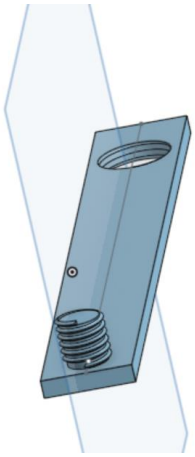


**Figure 13: Plane position**

After doing that we will create a pierce (highlighted in blue) relation between the triangle's base and the helix as shown in the photo.

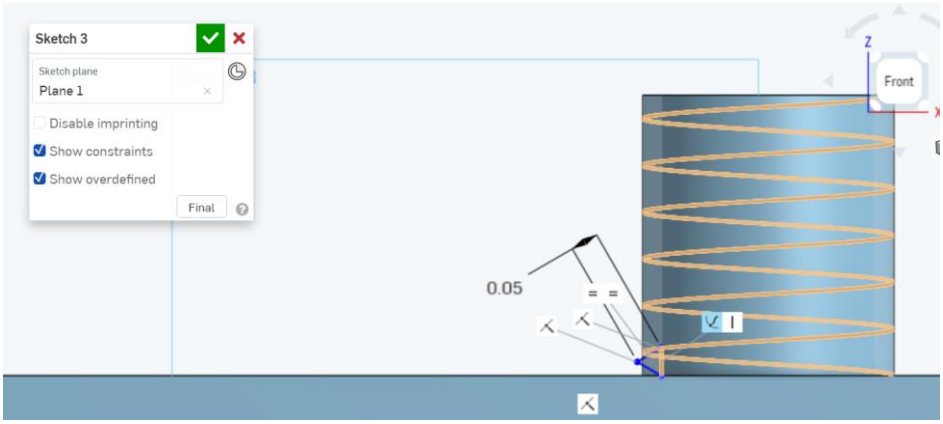



**Figure 14: Creating a pierce relation (linear rod)**

Then we finish by placing the triangle at the bottom and using the sweep option from OnShape select the face of sketch of triangle and for the sweep path choose the helix, with the shown parameters to obtain the desired screw.

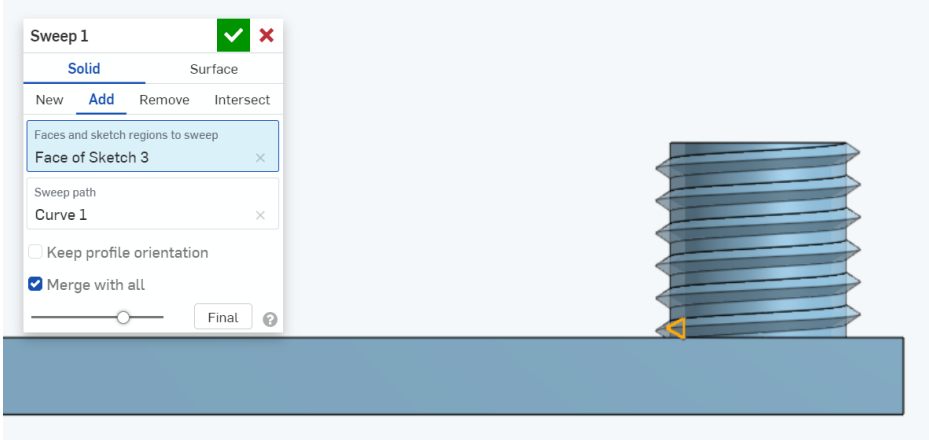



**Figure 15: Sweep properties (linear rod)**

Now we move the second rod which will have mostly the same properties. As usual create a new Part Studio, select a plane and sketch a rectangle with the following dimensions: 16in in length and 1.3in in width. Create a circle at the center of the rectangle with a 0.8in diameter as shown in the photo.

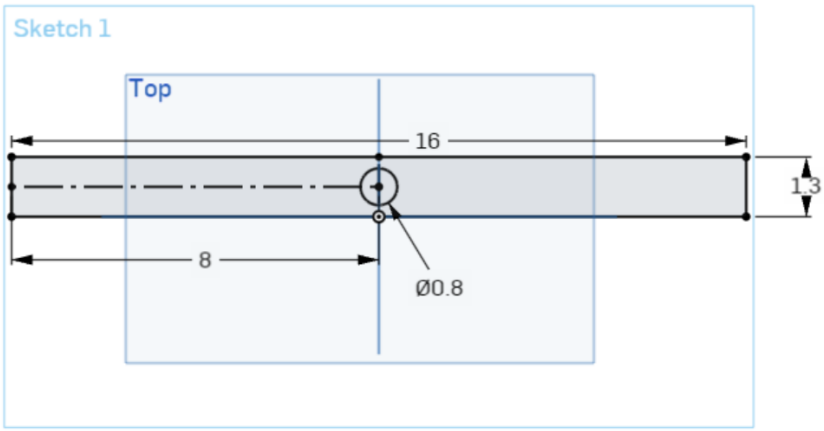



**Figure 16: Sketch for oscillating rod**

Now extrude it to a 0.197in depth and create a sketch on the top face of the rectangle and draw a circle at 1.841in from the center of the right side of the rectangle and draw a circle with a diameter of 0.45in as shown in the first photo, and then extrude that circle to a depth of 1in and with the same parameters as in the second photo.

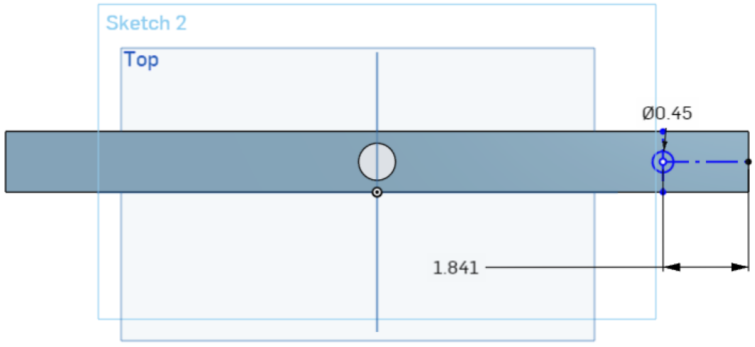



**Figure 17: Measurements for oscillating rod**

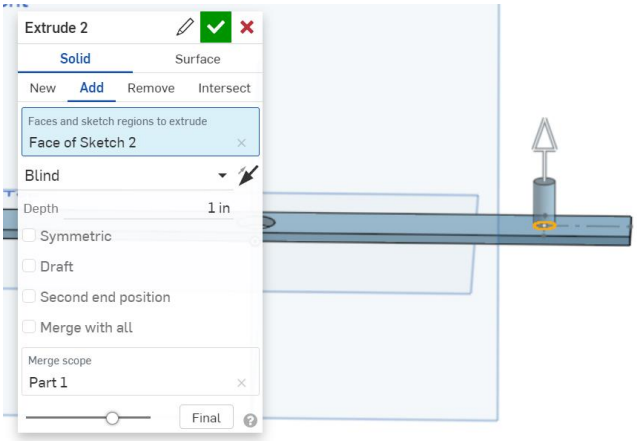



**Figure 18: Properties of extrude (oscillating rod)**

And then follow the same steps to create the screw form as the past rod with the same parameters.

#### 6.1.2.1.3 Rotator Disk

Start by creating a new Part Studio and then select a plane and start a sketch and draw a circle with a diameter of 4in. Then we will create a construction line of 1.5in from the center of the circle and create a smaller circle at the end of it with a diameter of 0.8in as shown in the photo, and then extrude it to 0.197in.

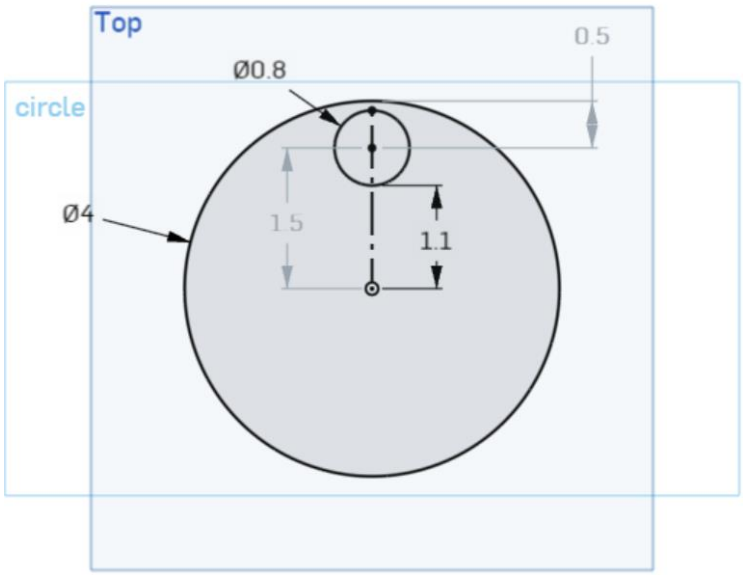



**Figure 19: Disk sketch**

After extruding, flip the disk to have the bottom view and start a new sketch on the bottom face of the disk then draw a circle of 0.3in diameter and another one inside of it of 0.18in diameter both having the center of the big disk (the bottom face) as their center as shown in the photo.

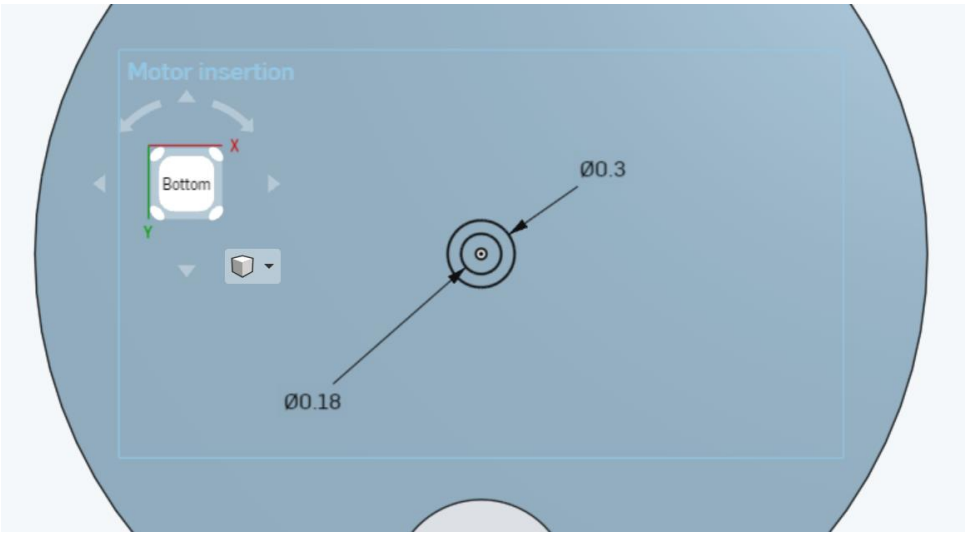



**Figure 20: Motor insertion sketch**

Then extrude the sketch to 0.394in with the same parameters shown in the photo to obtain this figure at the end.

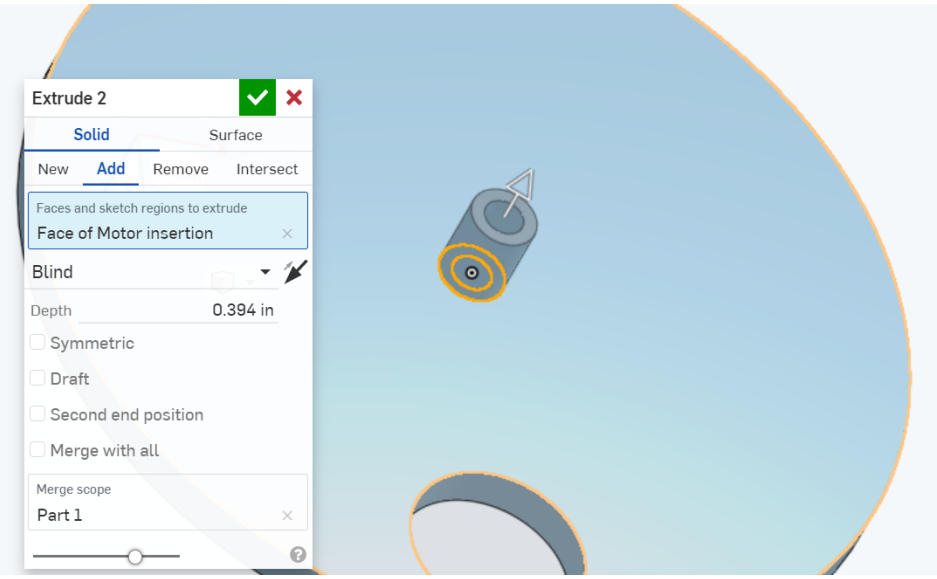



**Figure 21: Motor insertion extrude properties**

#### 6.1.2.1.4 Clips

We'll have two types of clips one for the fixed oscillating rod that we will call Fixed clip and the other for the Linear rod that we will call Clip. The only difference between the two is just the measurements, but everything else (building process, extruding, sweeping, etc..) is the same.

We'll start with the fixed clip. Start by creating a new Part Studio and select a plane to create a sketch. Draw a circle of 1.2in diameter and then extrude it to 0.197in. Then select the top face of it and create a sketch. Draw 2 circles having the center of the face as their centers with the measurements shown in the photo.

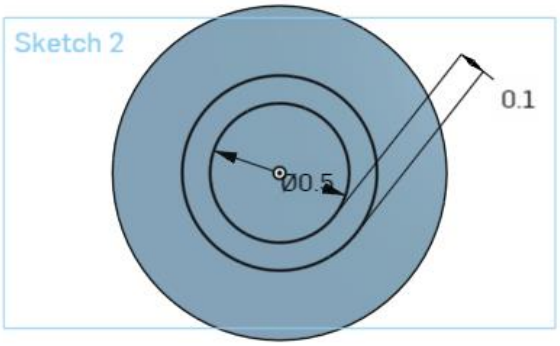



**Figure 22: Extrude properties (fixed clip)**

Then extrude the sketch to a depth of 1in to obtain the following shape.

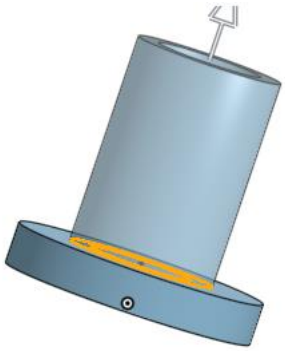


**Figure 23: Extrude shape**

Afterwards we'll draw helix inside the cylinder to then create the screw insertion form. Select the inside face of the cylinder and select helix and use the following properties.

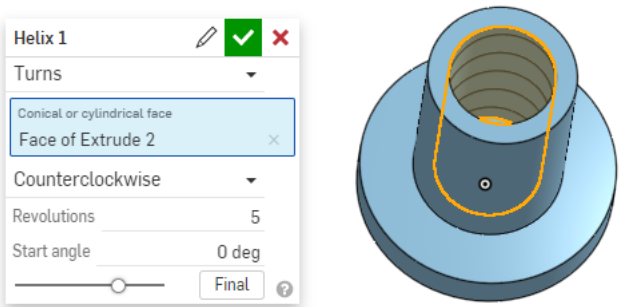



**Figure 24: Helix properties**

Then select a plane that cuts the cylinder in half and create a sketch (we recommend using the front plane to keep working with the same steps and measurements) and draw an equilateral triangle of 0.05in sides and create a Pierce relation between the base and the helix as shown in the photo.

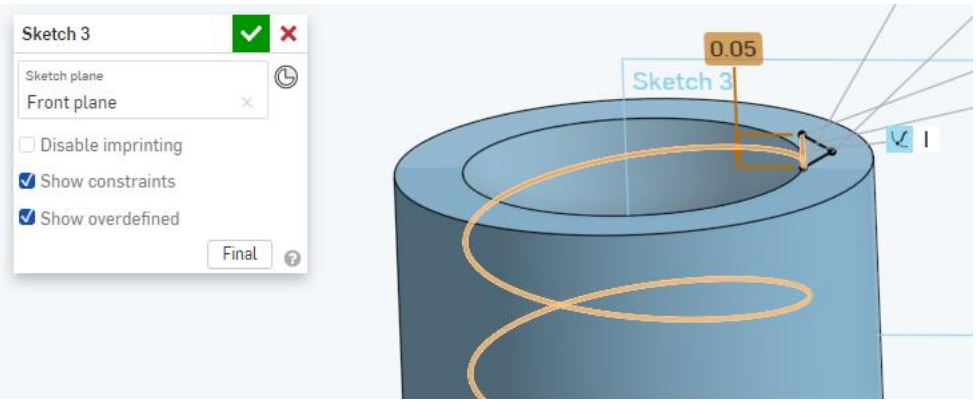



**Figure 25: Pierce relation**

Then using the sweep function, select the face of sketch of triangle and for the sweep path choose the helix, with the shown parameters to obtain the desired screw insertion but make sure to select Remove instead of Add to obtain this result.

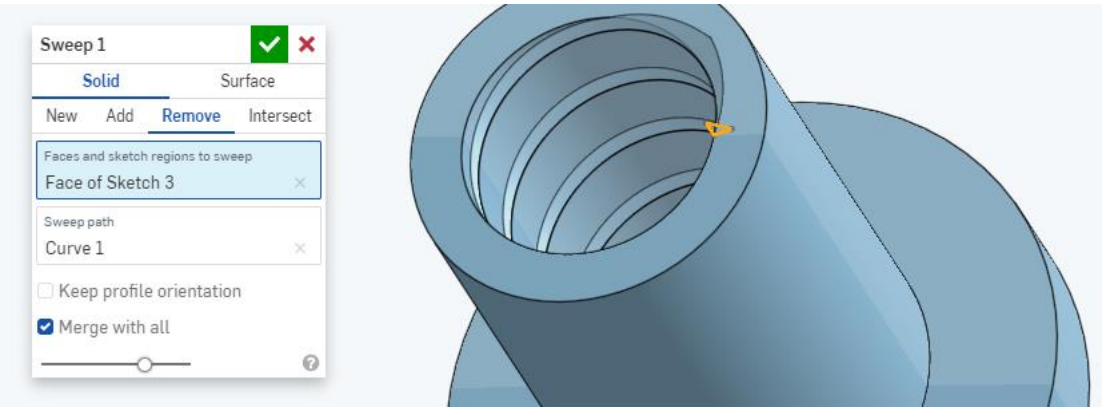



**Figure 26: Sweep properties**

Follow the same steps for the Clip but change the measurements for the following:

- The first extrude changes from 0.197in to 0.3in.

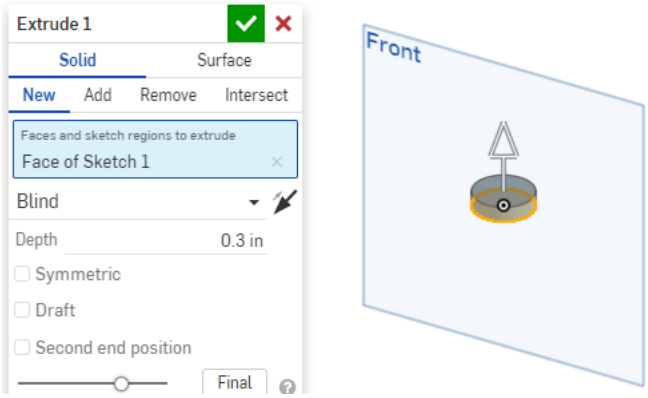



**Figure 27: Changing extrude measurement for clip**

The rest remains the same.

### 6.1.3 3D printing: Ultimaker Cura

This part is for 3D printing the parts to create the final prototype. We used Ultimaker Cura because it's the application that's compatible with the printers we have. First export your files from OnShape by right clicking on the respective part studio tab in the bottom of the page and selecting export. Choose the file format as STL and units in millimeter. After doing that for all your parts, open the Ultimaker Cura App and select the file shaped icon and open your downloaded files. You will find them to be very big in size compared to the printer so we'll change their size by doing the following:

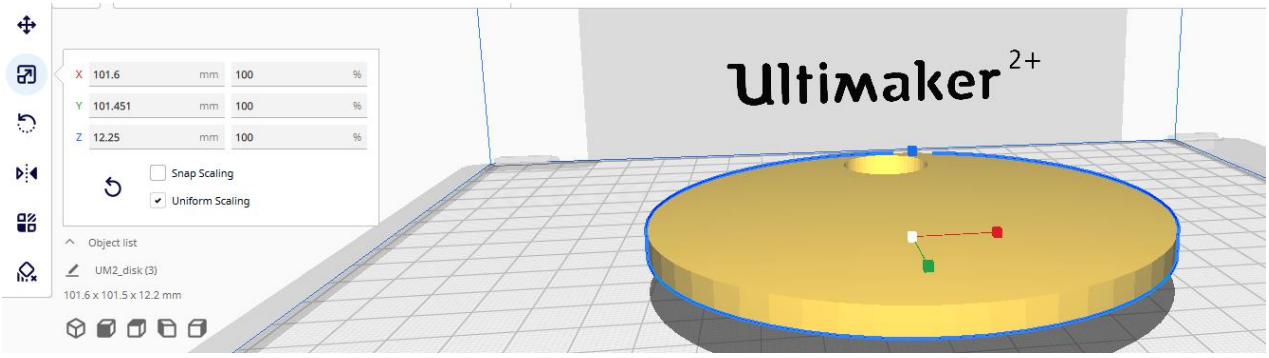



**Figure 28: Initial disk size (Ultimaker)**

Change the percentage for X, Y and Z to 45% and flip the disk so it sits on its top face to minimize the printing time. Flip it by using the flip icon and rotating it to when it's flipped as desired using the green circle and make sure to press the circled button to lay flat the disk.

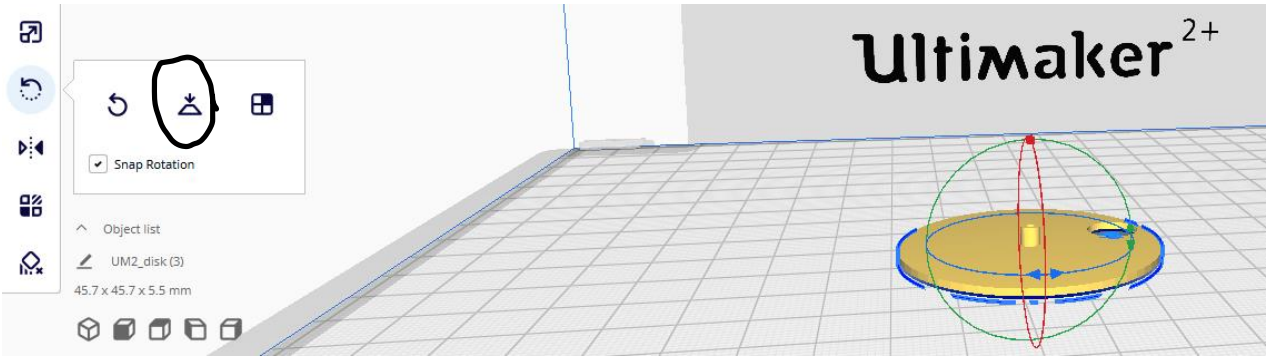



**Figure 29: Correct position for printing**

Repeat these steps for the rest of the parts and then select Nozzle size that goes with your printer from the top tool bar as shown in the photo.

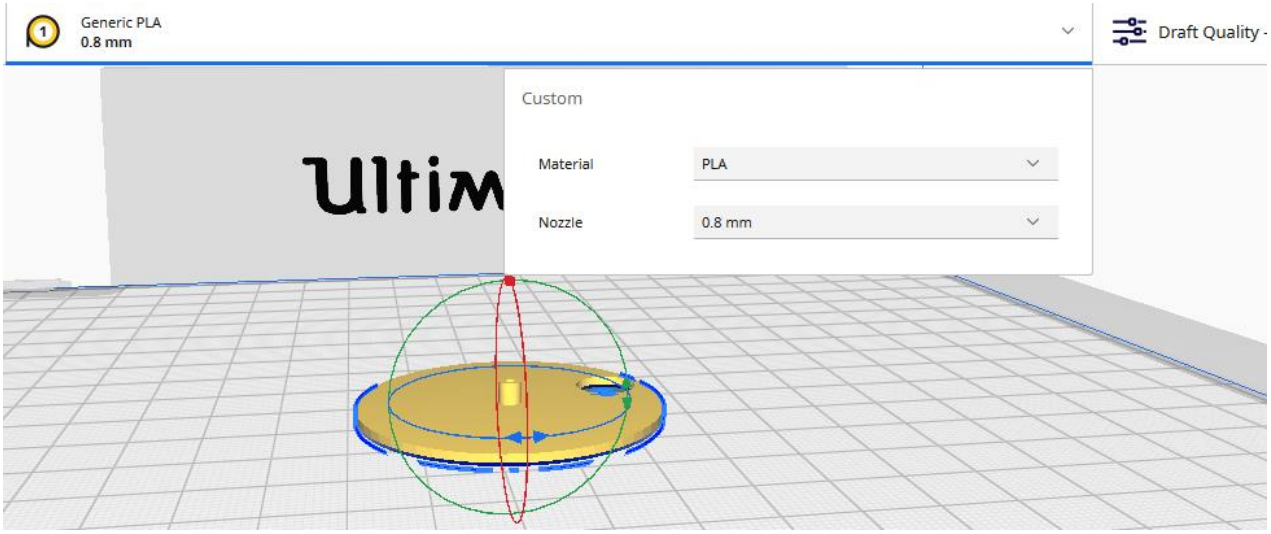



**Figure 30: Nozzle selection**

Make sure to change the settings to following:

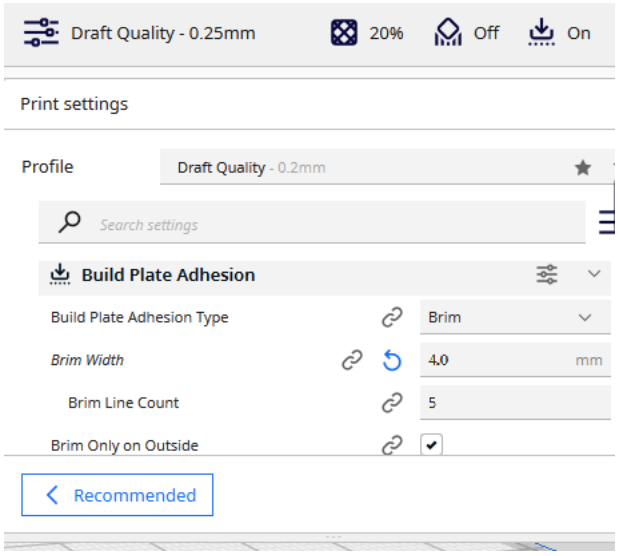



**Figure 31: Build plate adhesion settings**

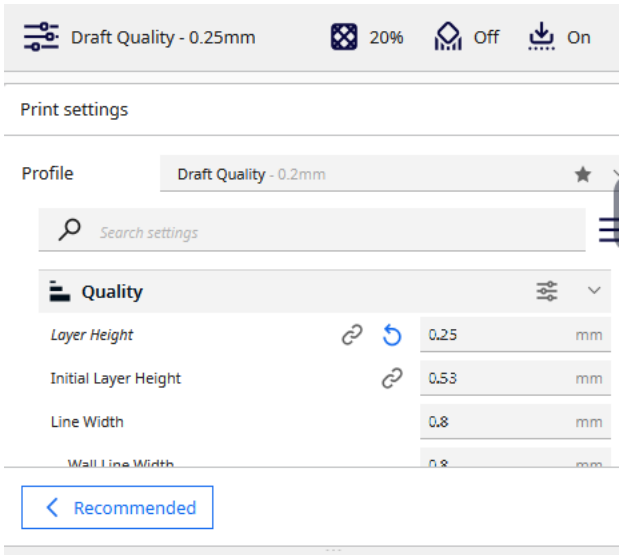



**Figure 32: Quality settings**

Then press slice and save it to a memory card and insert the memory card in the printer, select your file and print it.

### 6.1.4 Assembly of the pieces

After everything is printed, start by inserting the oscillating rod first in the designated screw on the base and use the fixed clip to seal it off. Then do the same for the linear rod and the disk then stick the motor beneath the disk and glue it on the base using the glue gun. You'll obtain a final product looking like this:

Insert photo of final prototype.

## 1.2 Subsystem II: Arduino and wires placements

### 6.1.5 BOM (Bill of Materials)

| **Item** | **quantity** | **cost** | **total** | **link** |
|---|---|---|---|---|
| bread bord | 1 | 5 | 5 | https://makerstore.ca/shop/ols/products/breadboard |
| Arduino UNO R3 clone* | 1 | 17 | 17 | https://makerstore.ca/shop/ols/products/arduino-uno-r3 |
| cables (10 pack) | 1 | 1 | 1 | https://makerstore.ca/shop/ols/products/jumper-cables-per-10 |
| accelerometer | 1 | 25 | 25 | https://makerstore.ca/shop/ols/products/adxl335-5v-ready-triple-axis-accelerometer |
| motor | 1 | 2 | 2 | https://makerstore.ca/shop/ols/products/compact-dc-hobby-motor |
| temperature sensor | 1 | 14 | 14 | |
| wire hook-up | 1 | 1.6 | 1.6 | https://makerstore.ca/shop/ols/products/usb-type-a-to-usb-typeb-cable-3ft |

| | | | | |
|---|---|---|---|---|
| LED screen | 1 | 5.33 | 5.33 | https://edu-makerlab.odoo.com/shop/product/oled-display-139#attr=247 |
| Arduino (IDE) | N/A | N/A | N/A | https://www.arduino.cc/en/software |
| | | **total** | 70.93 | |

### 6.1.6 Equipment needed

For this part you will need the following:

- Arduino Software (IDE)

### 6.1.7 Arduino code

For the functioning of the full prototype, we need an Arduino code. The one we used was missing just a formula to calculate the specific gravity from the found acceleration. Here's the code we used:

Code_for_SG §

```
#include <Wire.h> // Library for I2C communication
#include <LiquidCrystal_I2C.h> // Library for LCD
#include <LiquidCrystal.h>
#include <DallasTemperature.h> //library for temp sensor

LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 20, 4);
float temp_sensor=6;//digital pin
float specific_gravity;
float temperature=0;//store temp value
OneWire oneWirePin(temp_sensor);
DallasTemperature sensors(&oneWirePin);
int x;
int y;
int z;

void setup() {
Serial.begin(9600);
sensors.begin();
pinMode(temp_sensor,INPUT);
  // set up the LCD's number of columns and rows:
lcd.init();
lcd.backlight();
}

void loop() {
//read analog inputs for accelerometer:
  x=analogRead(0);
  y=analogRead(1);
  z=analogRead(2);
  Serial.print(x, DEC); // print acceleration in the X axis
  Serial.print(" "); // prints a space between the numbers
  Serial.print(y, DEC); // print acceleration in the Y axis
  Serial.print(" "); // prints a space between the numbers
  Serial.println(z, DEC); // print acceleration in the Z axis

//temperature measurements:
temperature = sensors.getTempCByIndex(0);
//temperature measurements:
temperature = sensors.getTempCByIndex(0);
Serial.println("Temp:");
Serial.println(temperature);


  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print("SG:");
  lcd.print("Temp:");
  lcd.println(temperature);


//delay between each time it prints
delay(5000);
}
```

**Figure 33: Arduino Code**

### 6.1.8 Wires placements

For the wires placements we will show them part by part in the form of drawings/photos followed by the positions on the Arduino Uno if necessary. Check the Appendix for more information about Arduino and its properties.

#### 1.2.1.1 Accelerometer

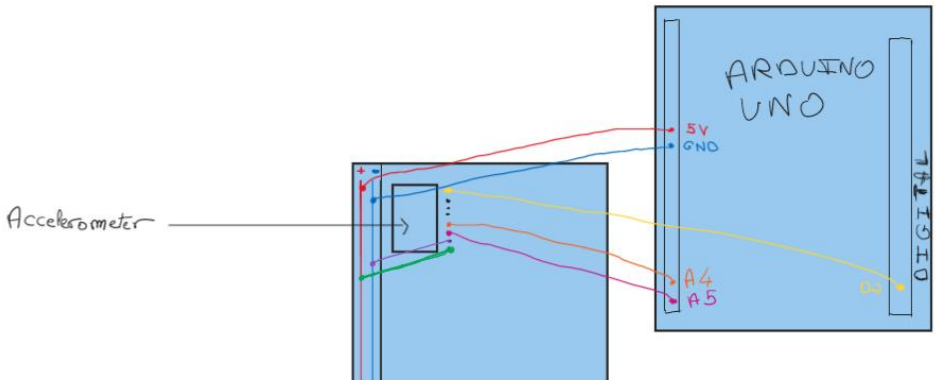



**Figure 34: Wiring for Accelerometer**

#### 1.2.1.2 Temperature sensor

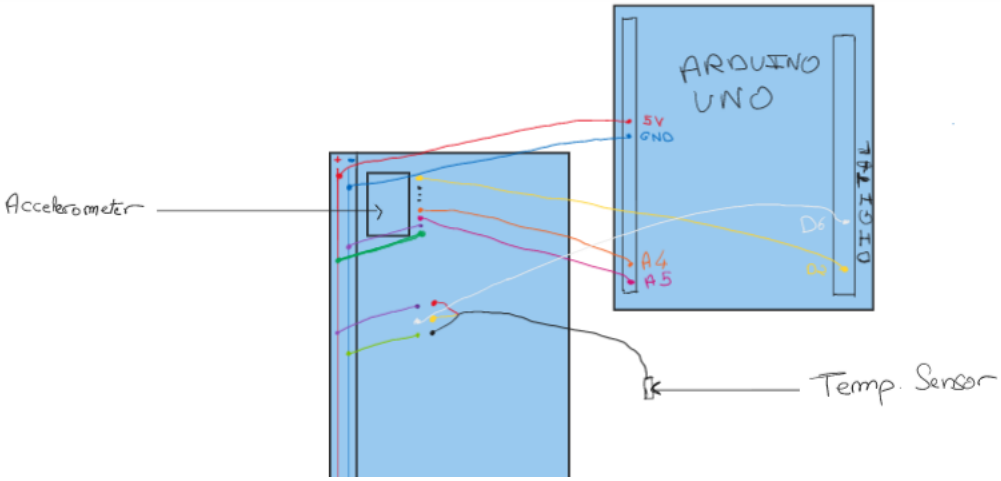



**Figure 35: Wiring for temperature sensor**

### 1.2.1.3 LCD screen

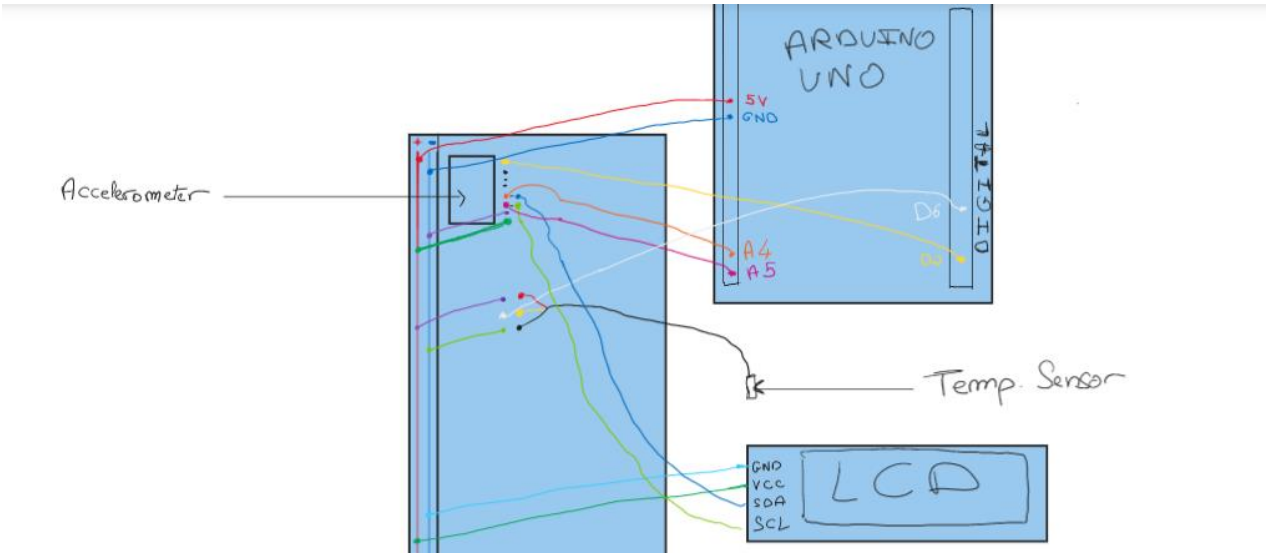



**Figure 36: Wiring for LCD screen**

SDA needs to be connected to A4 so we put it on the same horizontal line as the orange cable for the accelerometer and the SCL needs to be connected to the A5 so we put it in the same horizontal line as the purple cable for the accelerometer.

#### 1.2.1.4 Motor

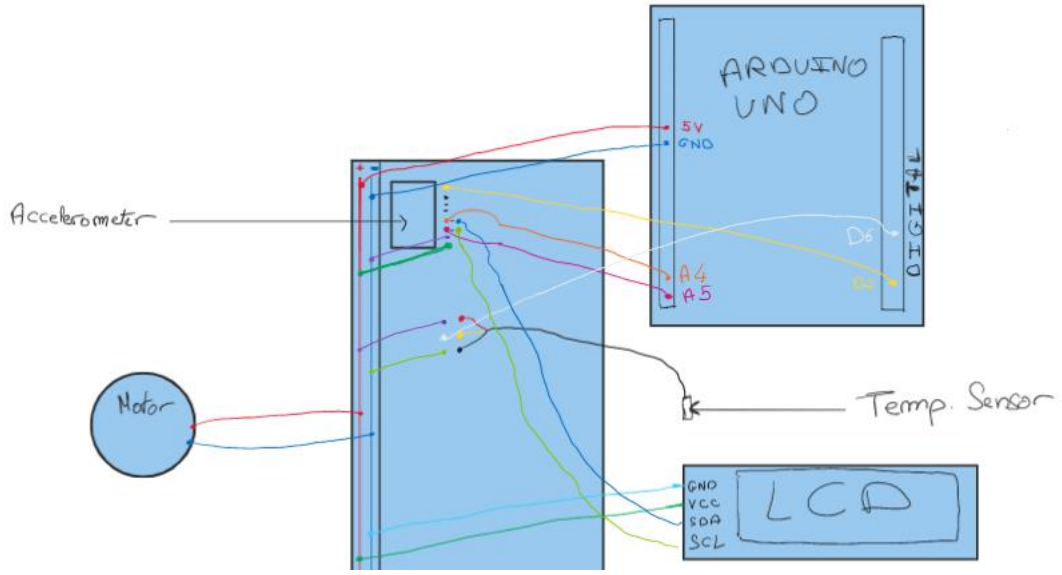



**Figure 37: Wiring for Motor**

## 6.2 Testing & Validation

### 6.2.1 Voltage testing

We tested our prototype to see which voltage was better to maintain the prototype safe and get good results, for the 3.3V it was fine, the prototype was working smoothly but in theory the faster the disk rotates the more accurate the measurements of specific gravity. So we moved to the 5V and the prototype barely held on at first, but when we worked again on the measurements and added some glue using the glue gun it worked perfectly fine.

### 6.2.2 System working altogether

We assembled everything together to test the code and system working, we ran the code and everything worked and was making sense, except for specific gravity measurements because the prototype was not waterproofed yet therefore we couldn’t run an actual specific gravity measurement and the accelerometer should be placed on the end of the oscillating rod that isn’t submerged in the beer to get the readings of acceleration more accurately.

# 7 Conclusions and Recommendations for Future Work

Our team learned many lessons during the duration of this design project. The first is the importance of feedback in the design thinking process. The feedback received from both our client and other sources helped us shape our product into what it is. The client reinforced design requirements and gave us insight into which of our subsystems were his favourite. Feedback was a pivotal factor in the design process. We also learned the importance of unbiased opinions from the client. Before the first client meeting our team had an idea of which subsystem would be our favourite. Regardless we decided to go into the meeting without telling him which one this was. Thanks to his unbiased opinion we ended up changing the design to better fit the clients needs and desires. Our team also learned a lot about organization and the importance of the design thinking process.

Regarding our prototype the team learned lots about using Arduino and setting up a breadboard. An important lesson that the client emphasized during this process is that glass is not a food-grade material. The client was very clear that there should be no glass in the fermentation tank.

Some recommendations for future work of this project would be to build a full comprehensive prototype made of food-grade materials that is waterproofed. Since our prototype is not made of food-grade materials we cannot test it to the same degree in liquids. This makes it difficult to find errors in our results and to test the coding aspect of the design including the graphing of the fermentation curve. If the group had more time and a larger budget we would have purchased waterproofed and or food-grade materials to build this prototype. Then we would have tested it first with water then with other liquids. Another improvement that the team would have made if given more time and a larger budget would have

been to purchase more accurate sensors. The sensors we used contained a lot of error compared to more expensive sensors that can be found online.

# 8 Bibliography

*Beyond the Pale Brewing Company*. (2022, 10 02). Retrieved from Beyond the Pale Beer: https://btpshop.ca/collections/cans-473ml

Lemis USA. (2016). *Density Meters*. Retrieved from http://www.lemis-usa.com/?mid=42&measure=43&pid=4

# APPENDICES

# 9 APPENDIX I: Design Files

**Table 7. Referenced Documents**

| Document Name | Document Location and/or URL | Issuance Date |
|---|---|---|
| MakerRepo | https://makerepo.com/ewill038/1314.imagineers-specific-gravity-device | 12/07.22 |

# 10 APPENDIX II: Other Appendices

## 10.1 Arduino Board

Arduino senses the environment by receiving inputs from many different types of sensors, and can also affect its surrounding environment with outputs that control lights, motors, heaters, or other actuators. These input and output signals are made available on a digital pins header (see diagram below).

## Digital Pins

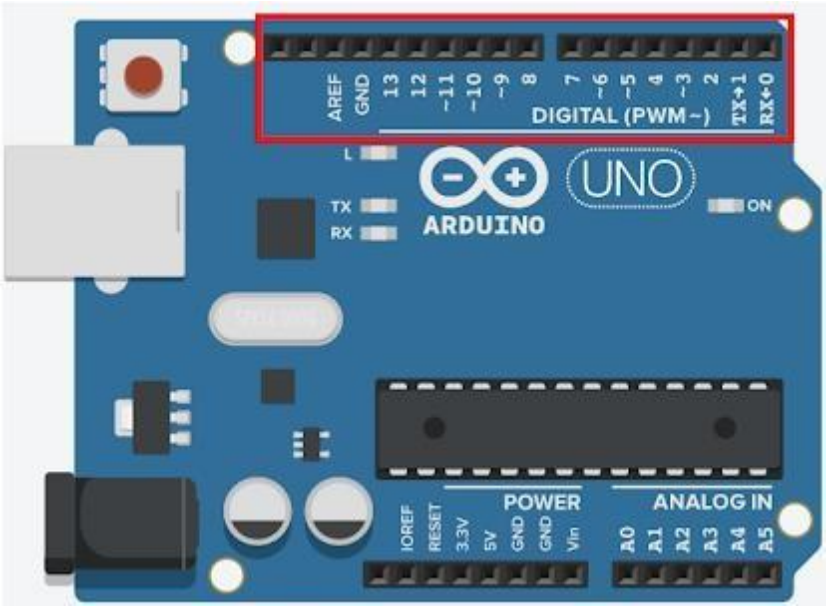



**Figure 38: Arduino UNO digital pins**

- These pins can be configured as digital INPUTS and used to send signals into the Arduino processor (e.g. to read sensed logic voltage levels). Digital signals are normally binary and only have two distinct voltage values (0V or 'low' or 5V or 'high).
- The pins can also be configured to act as an OUTPUT to drive digital 'high' or 'low' logic voltages from the Arduino processor (e.g. to turn things on or off).
- **Pin13 (LED)**: There is a built-in LED pre-connected to digital pin 13. When the value of

the pin is driven HIGH by the processor, the LED on the board is illuminated, when the pin is LOW, it's turned off. This can be used as a status indicator when programs are running.

- **Serial.print()**: Prints data to the serial port as human-readable text, see https://www.arduino.cc/en/Serial/Print for more information. (which only uses binary digital signals). Analog signal inputs can be accepted for conversion into

- **Comment bars**: "/* */" and "//": These allow you to write commentaries in your code. Everything you write in between "/* */ "or after "//" will be a comment and will not affect the code.
    - end of the current line/statement'. A compiler is a computer program that transforms source code written in a programming language into another computer language, usually one that can be executed more easily on a real processor.
- initially required to run the rest of the program, such as initializing any peripheral components and setting

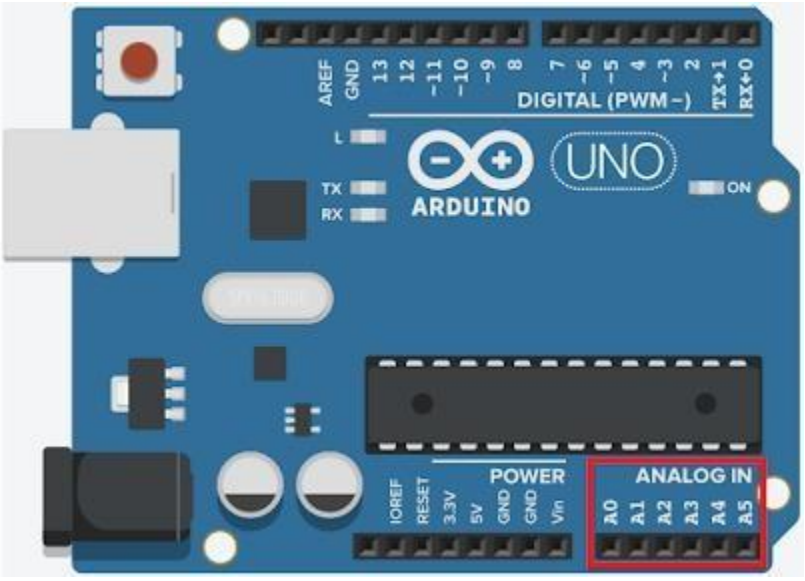



**Figure 39: Arduino Analog pins**

the communication frequency between the Arduino board and PC. HYPERLINK "https://www.arduino.cc/en/Reference/Setup"https://www.arduino.cc/en/Reference/Setupcan be executed more easily on a real processor.

- **"loop" function**: The loop function acts as the programs driver; it runs in a continuous loop, and specifies the order of operations that the microcontroller will perform. https://www.arduino.cc/en/Reference/Loop. Execution starts at the top, goes through the contents of the loop and then starts executing from the top again. This procedure is repeated forever. frequency between the Arduino board and PC. https://www.arduino.cc/en/Reference/Setup

- Start and ends a function or is used to group a set of different statements together, making them effectively a single long sequential statement, that is executed as a ‘glob’. the order of operations that the microcontroller will perform. https://www.arduino.cc/en/Reference/Loop. Execution starts at the top, goes through the contents of the loop and then starts executing from the top again. This procedure is repeated forever.

- **pinMode()**: Configures the specified pin to behave either as an INPUT or an OUTPUT, see https://www.arduino.cc/en/Reference/PinMode for more information.

- **Serial.print()**: Prints data to the serial port as human-readable text, see https://www.arduino.cc/en/Serial/Print for more information.

- **For more information, see https://www.arduino.cc/en/Serial/Begin for more information.**

- Syntax: Serial.begin(9600) sets the serial interface to run at a rate of 9600 serial bits/sWrites a HIGH or a LOW value to a digital pin.

- **Serial.print()**: Prints data to the serial port as human-readable text, see https:/ HYPERLINK "http://www.arduino.cc/en/Serial/Print"/www.arduino.cc HYPERLINK "http://www.arduino.cc/en/Serial/Print"/en/Serial/Print for more information. see https://www.arduino.cc/en/Reference/DigitalWrite for more information.

  - Syntax: Serial.print(value) value)you write in between "/* */ "or after "//" will be a comment and will not affectthe code.

- **"if" statement**: See HYPERLINK "https://www.arduino.cc/en/Reference/If"https://www.arduino.cc/en/Reference/If : This lets you conditionally execute the following statements, depending on a tested condition.compiler is a computer program that transforms sourcecode written in a programming language into another computer language, usually one that can be executed more easily on a real processor.

  Breadboard delay(ms) HYPERLINK "https://www.arduino.cc/en/Reference/Delay"https://www.arduino.cc/en/Reference/Delay to run the rest of the program, such as initializing any peripheralcomponents and setting the communication frequency between the Arduino board and PC. https://www.arduino.cc/en/Reference/Setup

- **Variable**: a variable is a storage location that is paired with an associated symbolic name

(an identifier), which contains some known or unknown quantity of information referred to as the value of that variable. order of operations that the microcontroller will perform. https://www.arduino.cc/en/Reference/Loop. Execution starts at the top, goes through the contents of the loop and then starts executing from the top again. This procedure is repeated forever.

- **Brackets** "{ }" Start and ends a function or is used to group a set of different statements together, making them effectively a single long sequential statement, that is executed as a 'glob'.

- In this lab we for more information.going to use two types of variable value types: “int” and “float”.
- **Serial.print()**: Prints data to the serial port as human-readable text, see https://www.arduino.cc/en/Serial/Print for more information.

- For more information, see HYPERLINK "https://www.arduino.cc/en/Serial/Begin"https://www.arduino.cc/en/Serial/Begin for more information. digital pin.
- Syntax: Serial.begin(9600) sets the serial interface to run at a rate of 9600 serial bits/s The result is that these kinds of variables can have a much larger range than is possible for integers, but they will have some roundup errors because they are implemented with finite precision (i.e. only a certain number of decimal digits can be used). See HYPERLINK "https://www.arduino.cc/en/Reference/Float"https://www.arduino.cc/en/Reference/Float for more information.HIGH, 0V (ground) for LOW, see https://www.arduino.cc/en/Reference/DigitalWrite for more information.

- **Serial.print()**: Prints data to the serial port as human-readable text, see https://www.arduino.cc/en/Serial/Print for more information.
  - Syntax: Serial.print(value)This starts off the serial communications interface hardware. You will need this, if you want to print anything using the serial monitor.
- **“if” statement**: See HYPERLINK "https://www.arduino.cc/en/Reference/If" https://www.arduino.cc/en/Reference/If : This lets you conditionally for more information. the following statements, depending on a tested condition.

Breadboard bits/s(an identifier), which contains some known or unknown quantity of information referred to as the value of that variable.

- **Serial.print()**: Prints data to the serial port as human-readable text, see https://www.arduino.cc/en/Serial/Print for more information.

- Syntax: Serial.print(value) lab we are going to use two types of variable value types: “int” and “float”.

- **“if” statement**: See HYPERLINK "https://www.arduino.cc/en/Reference/If"https://www.arduino.cc/en/Reference/If : This lets you conditionally execute the following statements, depending on a tested condition. counting whole) numbers.https://www.arduino.cc/en/Reference/Int.

## 10.2 Breadboard more information, see https://www.arduino.cc/en/Serial/Begin for more information.

- Syntax: Serial.begin(9600) sets the serial interface to run at a rate of 9600 serial bits/sThe result is that these kinds of variables can have a much larger range than is possible for integers, but they will have some roundup errors because they are implemented with finite precision (i.e. only a certain number of decimal digits can be used). See https://www.arduino.cc/en/Reference/Float for more information.

- **Serial.print()**: Prints data to the serial port as human-readable text, see https://www.arduino.cc/en/Serial/Print for more information.
    - Syntax: Serial.print(value)This starts off the serial communications interface hardware. You will needthis, if you want to print anything using the serial monitor.

Breadboard: See HYPERLINK "https://www.arduino.cc/en/Reference/If"https://www.arduino.cc/en/Reference/If : This lets you conditionally execute for more information.the following statements, depending on a tested condition.

Breadboard bits/s

- **Serial.print()**: Prints data to the serial port as human-readable text, see https://www.arduino.cc/en/Serial/Print for more information.
    - Syntax: Serial.print(value)

Breadboard: See https://www.arduino.cc/en/Reference/If : This lets you conditionally execute the following statements, depending on a tested condition.

## 10.3 Breadboard

A breadboard is used to prototype a temporary circuit. You can build, test and analyze a circuit without any permanent connections. It is made up of terminal strips and power rails. The terminal strips are used to hold any number of components in place and make electrical connections in a horizontal row. The **power rails** are the long vertical strips and are used to facilitate power (+) and ground (-) connections by placing them all in one column. For some general background on how such a breadboard actually works, take a look at the YouTube video about breadboarding at: https://www.youtube.com/watch?v=6WReFkfrUIk.

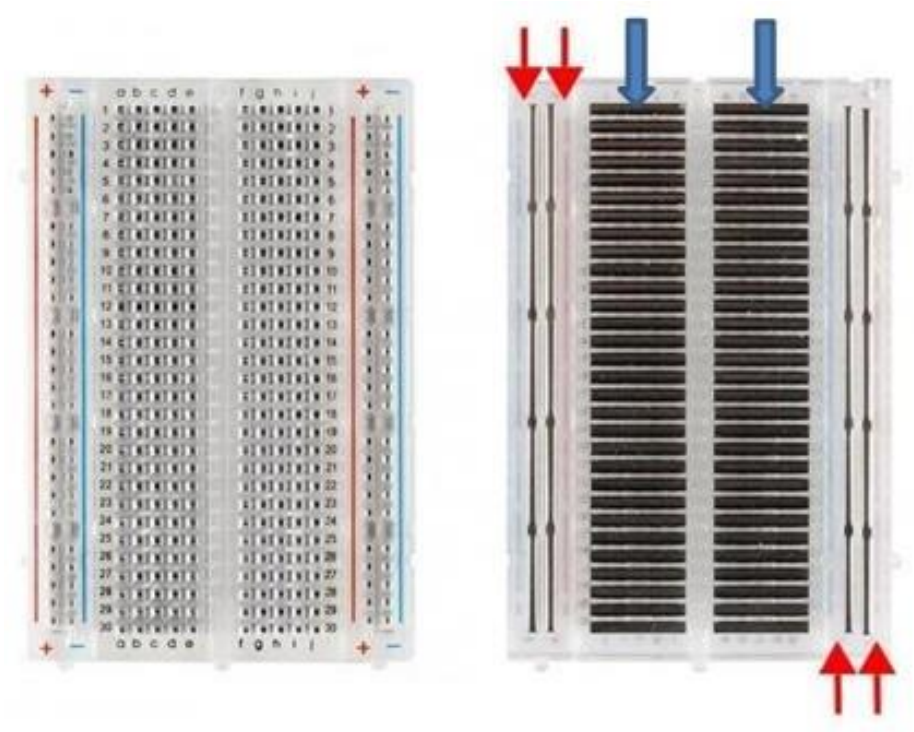


**Figure 40: Breadboard**