Hydration Tracking Coaster with BLE Android App

A Major Qualifying Project

Submitted to the faculty of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements of the
Degree of Bachelor of Science By:

Rachel Scaer, Electrical and Computer Engineering
Anthony Chan, Electrical and Computer Engineering

Submitted to Project Advisor:

Professor Xinming Huang, Department of Electrical and Computer Engineering

On March, 20th, 2018

Abstract

Many people in America struggle to drink their daily recommended amount of water. To combat this we created a coaster that tracks the amount of water consumed by an individual to encourage drinking the recommended amount. The coaster uses a 5kg load cell attached to an Arduino via an HX711 ADC chip made especially for load cells to measure the change of the weight of the water. This information is sent via a Bluetooth Low Energy (BLE) Redbear shield on the Arduino to an app on an Android phone. The changes in weight detected by the load cell are converted to and displayed as the amount of water consumed. The app compares the current amount of water drank to the users' daily water consumption goal. The app also sends notifications reminding the user to drink. The system can track water consumption accurately to +/-3 grams.

Acknowledgements

This project wouldn't have been possible without the help of our many supporters.

We'd like to thank Professor Huang for his time and effort spent advising us, helping us to brainstorm new ideas, thinking of innovative coaster designs, and purchasing last minute parts.

We'd like to thank our many friends who went above and beyond in helping us with mechanical design and build.

We'd like to thank Shane Bell for his help in creating and building our earliest coaster designs, assisting with laser cutting, and helping to tap screw holes in our later 3d prints.

We'd like to thank Jonathan Luna for his help in the shop tapping 3d prints, and general mechanical advice and tips.

We'd like to thank Howard Vance for his advice on why you shouldn't 3d print screws.

We'd like to thank Andrew Olding for his advice on elevating the load cell for our final design.

We'd like to also thank Garrison Hefter for going above and beyond converting our paper designs into 3d cad models and using his own personal 3d printer to bring our designs into reality. We'd also like to thank him for his general mechanical advice and insights.

Table of Contents

Table of Figures

1. Introduction

Today many people are struggling with chronic dehydration. This constant state of dehydration can have many negative impacts on health such as kidney stones, cognitive impairment, sleep apnea, exacerbated hypertension, and renal damage. This chronic dehydration is often caused by individuals relying on their inaccurate sense of thirst to determine when they should drink water. Thirst isn’t felt until after an individual is already dehydrated. For many an impaired central nervous system can lead to feeling little or no thirst even when dehydrated, and for others the sensation of thirst is easily confused with the sensation of hunger, leading to a higher consumption in high calorie foods by the chronically dehydrated. Unable to rely on their sensation of thirst, chronically dehydrated individuals need to outsource the assessment of their hydration levels.

One way to monitor hydration levels is to manually log the water intake of an individual, but for many people who don’t even remember to drink water this can be challenging. Another solution is to automatically track water intake, and to provide automatic reminders to drink more water before the individual becomes dehydrated. Many such devices already exist. They rely on weight measurement, or an accelerometer built into an expensive water bottle or cup to track the amount of water leaving the vessel. This means to accurately track hydration you are only able to use that vessel.

In response we created a coaster that accurately tracks water consumption by measuring the changes in weight. Keeping the water measurement device separate from the vessel allows the user to switch between any cups, water bottles, or other preferred vessels while still accurately tracking water consumption. Being able to accurately assess hydration levels while using vessels an individual is comfortable with is a formula for increased hydration. The addition of an easy to use app where the water consumption data is tracked, and notifications are given if the user falls behind on their goals makes tracking hydration easier than ever.

2. Literature Review

Dehydration, especially when chronic, can have negative effects on both body and mind. Conditions caused by dehydration include kidney stones, cognitive impairment, sleep apnea, exacerbated hypertension, and renal damage.

The World Health Organization (WHO) recommends the average adult male drinks 2.9 liters of water a day and the average female drinks 2.2 liters. Beverages account for 70-80% of the average adult's water consumption, the rest being from food [9]. The National Health and Nutrition Examination Survey (NHANES) reports the average US adult male consumes 2.6 liters of water a day and the average US female consumes 2.2 liters [9]. 1.1 liters of this water come directly from tap water [9]. 23% of women and 30% of men are reported as consuming less water than the recommended amount [9]. This group was also more likely to consume high energy foods and beverages, as well as be overweight [9]. This may be due to the difficulty of distinguishing thirst and hunger. Thirst is a bad indicator of dehydration, and only is felt after 1-2% of body mass loss [9]. This is even worse for seniors due to impaired central nervous systems that lead to a limited sensation of thirst, as well as trouble remembering to drink water. 95% of men and 83% of women over 71 years old failed to meet the daily water consumption [17]. Dehydration, especially when chronic, can have negative effects on both body and mind. Conditions caused by dehydration include kidney stones, cognitive impairment, sleep apnea, exacerbated hypertension, and renal

damage [4, 15, 18]. Adults struggling with dehydration need a way to track their water consumption as well as be reminded and motivated to drink beyond their own thirst.

Promoting the drinking of plain water specifically over other sources of water such as food, soda, and coffee is important. Consumption of plain water is associated with decreased rates of chronic diseases, healthier lifestyles, and healthier diets. There are also individuals who must closely monitor the amount of water they drink to properly manage illnesses such as kidney disease and congestive heart failure. Drinking 16oz of water before each meal has shown to increase weight loss by 2kg in a two group yearlong weight loss study [2].

A few different ways to measure water intake currently exist. The simplest way is to manually record your water intake in a notebook, writing down the amount of water you consume throughout the day. The higher tech option of this would be a water tracking app such as Waterlogged. You record how much water you're drinking by manually entering the amounts into the app, or taking pictures of your cup or bottle. The app keeps track of how much water you've been drinking, as well as sends reminders to help you achieve your goal. Many people not meeting their hydration needs have trouble remembering to drink water, or find it distasteful. Manual logging their water consumption may be difficult to remember for users, as well as add to the tedium of water consumption. Busy individuals may find it difficult to find the time to constantly enter their water consumption.

A few smart water bottles and cups attempt to solve this problem by measuring and recording water consumption automatically. One of the most studied options is the Hidrate Spark [11]. The plastic water bottle is BPA free and contains a 3-axis accelerometer and capacitive touch sensor. The capacitive touch sensor senses whether the lid is on the water bottle to record if it is being filled or emptied so the action is not recorded as water consumption. The accelerometer tracks how much water is being drank, by recognizing which movements of the bottle are associated with drinking. This data is stored in a microprocessor which can record your water consumption patterns for up to two days, as well as send it to your phone via Bluetooth where it is recorded in the associated app. The water bottle's charge can last for 6 months to a year thanks to coin cell batteries. Preliminary studies show the accuracy of tracked water consumption is 95% [5]. Other features of the water bottle include a water consumption goal customized to your personal needs, a variety of fun and encouraging notifications reminding you to drink, and the water bottle glows when you need to drink as well. Many reviews speak highly of the water bottle, citing its ease of use, fun reminders, and slick design as a motivation for them to drink more water. Some medical studies are beginning to see if use of the water bottle helps to prevent kidney stones in patients by encouraging more water consumption, one of the cheapest and most effective ways to prevent kidney stones [6].

The first smart water bottle developed was the Sportline Hydracoach [12]. This water bottle is very simple. All information and reminders are directly from the processor on the water bottle. It tracks how much water you drink through an impeller flow sensor in the straw. The water bottle must be held up straight while drinking to work properly. This water bottle is a good choice for users without smartphones.

Another popular choice is the H2OPal [10]. It tracks your water consumption via a weight sensor and accelerometer housed at the bottom of the bottle. It works similarly to the Hidrate Spark in that your drinking information is sent to an app, and notifications are used to encourage more drinking. The weight sensor and accelerometer can be removed off the provided bottle and attached to a bottle of your choice if it has the correct dimensions.

Smart cups exist as well. One of the most popular is the Moikit Cuptime [16]. It uses a weight sensor at the bottom of the cup, as well as a temperature sensor. The temperature is displayed on the side of the cup, and your water usage is tracked via an app. The cup also offers wireless charging via a small charging plate.

The main complaint for all of these products is the material. Many people complain the products break after normal use: throwing it in a backpack, dropping it, running it through the dishwasher. These users need a way to continue to use the cups and bottles they know to be durable and dependable while still having automatic water tracking.

3. Objectives

Our goal for this project is to design and build a coaster that can accurately track the amount of water consumed by the user to allow for better knowledge of hydration status. The coaster will track water consumption via measured changes in weight of the user’s water bottle or cup caused by the consumption of water after the bottle or cup has been placed on the coaster. These changes in weight will be sent to an app in the user’s phone that will alert the user to how much water they have drank in a day and when they should drink more water. In addition the app will provide a history of the user’s daily water intake habits. The app will allow for each user to create a personalized account and personal water consumption goal.

To allow for ease of use the coaster will have wireless charging. This means the coaster will be useable even while it is charging.

4. Design

4.1 Arduino Uno

For the computer processing of our design we chose to use an Arduino Uno.

Figure 1 Arduino Uno

This is because the sensors we chose to use in the design can easily interface with the Arduino, and example code is readily available and well documented. This allowed us to avoid reinventing the wheel and begin prototyping as quickly as possible.

4.2 Red Bear Bluetooth Low Energy

To inform users how much water they have been drinking the weight of the cup or water bottle must be sent from the Arduino to an app on the user's phone. We are using a Red Bear Bluetooth Low Energy shield to do this.

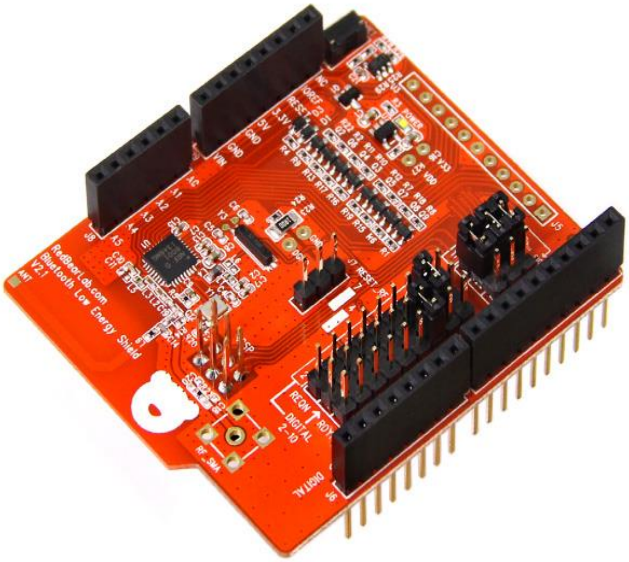


Figure 2 Redbear BLE Sheild

4.3 Force Sensitive Resistor

The first step of our design process was to choose a sensor to measure the changes in weight of the water held by a cup or bottle. The sensor needed to be small enough to fit inside a coaster. The average coaster has a diameter of 4 inches and a height of 2mm [19].

Although we allowed for more flexibility in terms of height, the goal was to create a coaster as close to the standard size as possible. The coaster also needs to be able to fit other necessary components that we will describe later on, making a small sensor size even more imperative.

In addition to size the sensor must be accurate and consistent. The average sip of water weighs approximately 20 grams, so the sensor must be accurate to at least that amount to allow for accurate tracking of water consumption.

The first sensor we tested for our design was a 0.5 inch force sensitive resistor sold by sparkfun.

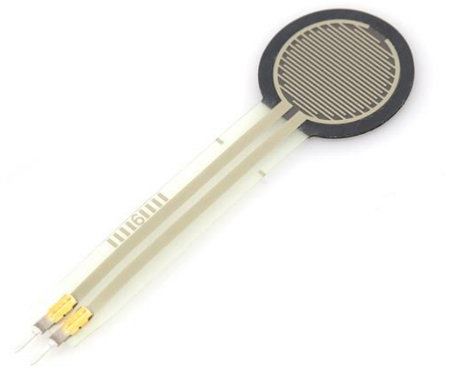


Figure 3 Force Sensitive Resistor

The sensor is made from interdigitating electrodes printed on a flexible substrate semiconductor. When pressure is exerted on the sensor the measured resistance from the sensor decreases, the ratio of pressure to resistance can be seen on the following graph in a log/log format [8].

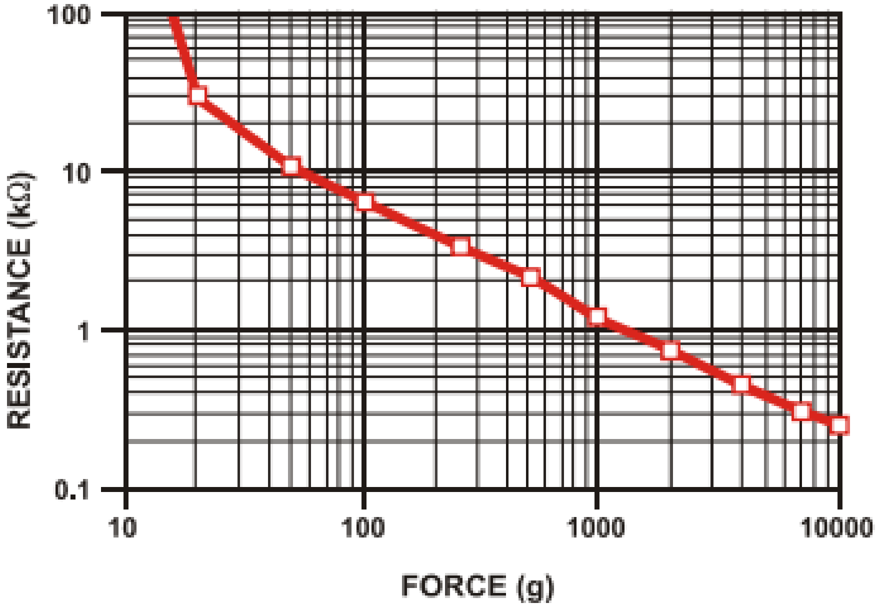


Figure 4 Applies Force vs Meaured Resistance for FSR

As you can see from the graph, the relationship between resistance and force does not become linear until at least 20g of force are exerted on the sensor [8]. To test the sensor in our application we first designed and 3-d printed a simple coaster. The coaster has four round feet. Each foot was placed on a force sensitive resistor. The force sensitive resistor was then attached to the Arduino as shown.

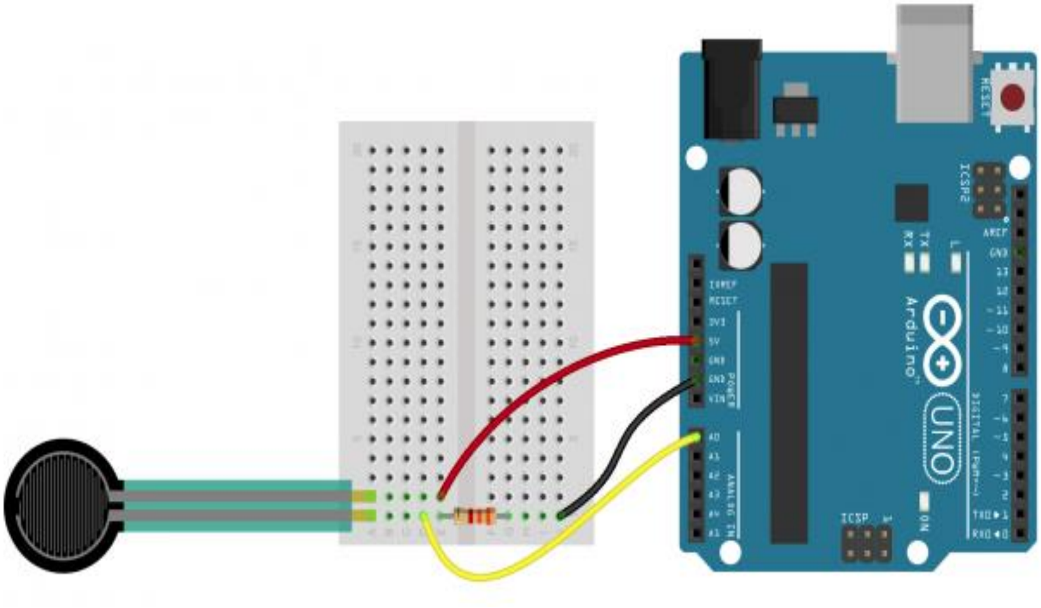


Figure 5 FSR attached to Ardunio via voltage divider

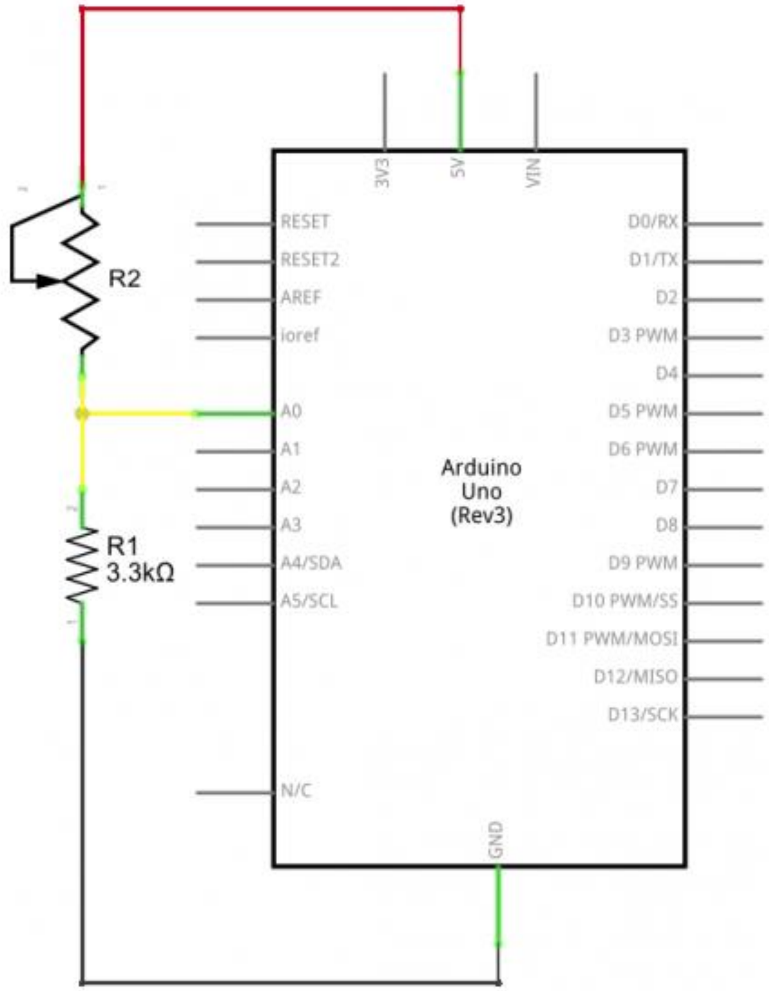


Figure 6 Schematic of Voltage Divider

We built a voltage divider by feeding the 5V output of the Arduino into the force sensitive resistor followed by a 3.3kΩ resistor wired to ground [7]. The voltage drop across the force sensitive resistor was then measured using the analog input (A0) of the Arduino. The voltage measured can be shown using the voltage divider equation below.

$A0 = 5v\ (3.3k\Omega\ /\ (R_{FSR} + 3.3k\Omega)$

When the Arduino measures A0 this voltage can be used to calculate the resistance of the Force Sensitive Resistor.

$R_{FSR} = (16.5k\Omega \bullet V)/A0 - 3.3k\Omega$.

The corresponding force for the resistance can then be calculated using the graph above.

At strong forces the relationship between force and resistance is not linear so if the inverse of R_{FSR} is less than 600 then

$\text{Force} = ((1/R_{FSR}) - 0.00075) / 0.00000032639$

Otherwise

$\text{Force} = (1/R_{FSR}) / 0.000000642857$

After setting up the code for the Arduino and setting up the voltage divider I began testing the sensors. At first I used a single force sensitive resistor and placed it under a single foot of the coaster. I then placed various calibration weights (as seen below) on the coaster immediately above the foot holding the sensor and read the force output on the Arduino serial monitor.

The Force Sensitive Resistor was responsive and good for measuring pressure but was too inaccurate for our application. While the calibration weight was placed stationary on the sensor the measured force fluctuated as much as 100 grams. With our need to track the amount of water drank accurately down to 20 grams this was too much room for error. So despite its small size and ease of setup the force sensitive resistor was not the ideal sensor for weight measurement for our coaster.

4.4 Load Cell

Looking for a sensor more accurate for measuring small amounts of weight we decided to try using a load cell. A load cell is simply a piece of metal with one or more strain gauges attached. A strain gauge is a coiled length of wire acting as a resistor shown below:

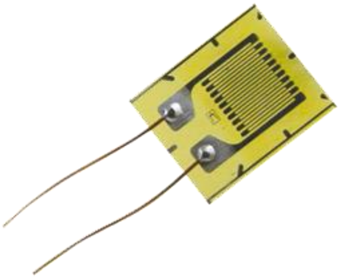


Figure 7 Strain Gauge

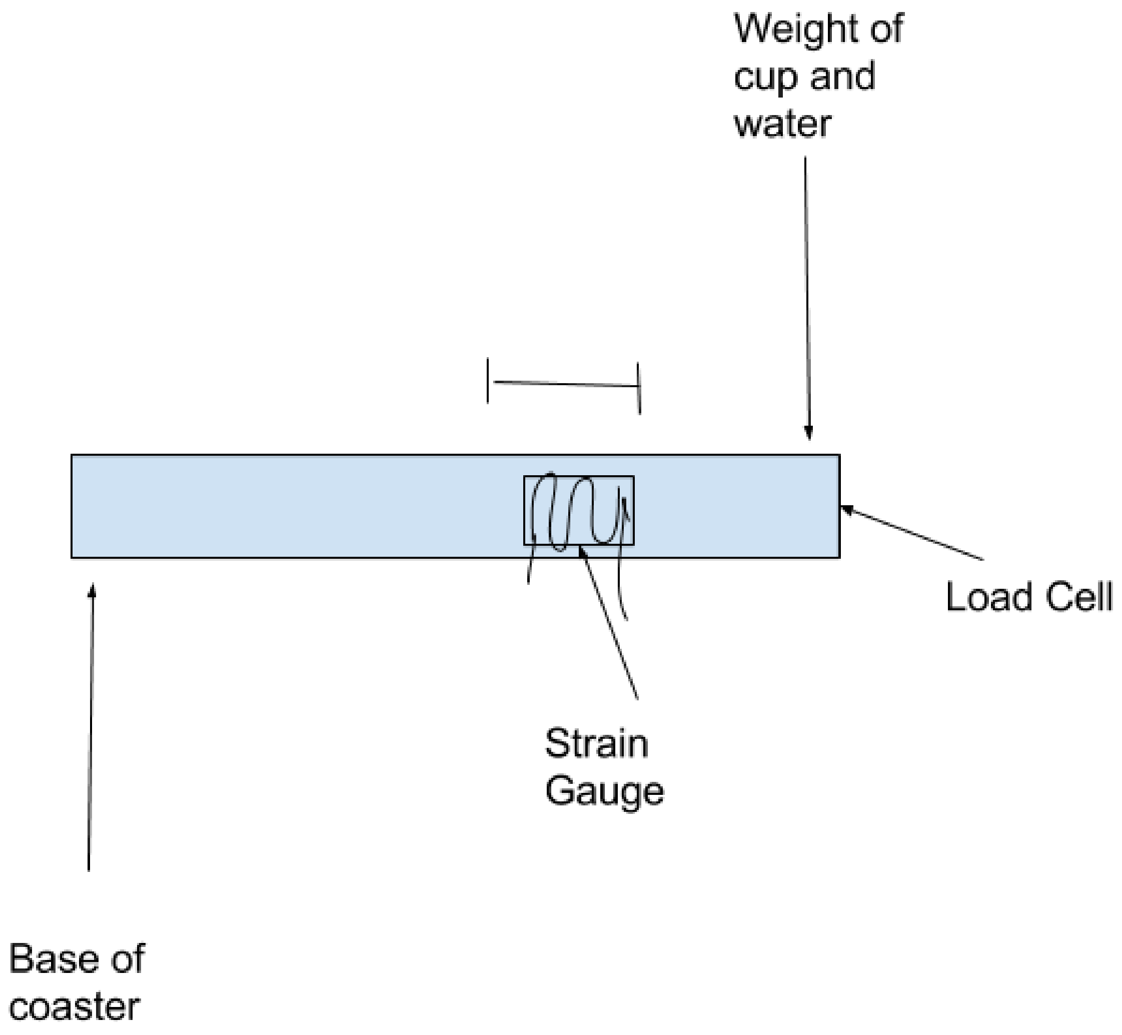


Figure 8 Straing Gauge on Load Cell

As shown in the above diagram the force of the weight of the cup and water is exerted on one end of the load cell, while the table the coaster is resting on is exerted on the other. These two forces combine to exert stress across the load cell. This stress bends the load cell as well as the strain gauge, slightly changing its resistance. The more water weight exerted on the load cell, the more stress exerted across. The more stress exerted across it, the more the load cell bends. The more the load cell bends, the more the strain gauge bends. The more the strain gauge bends, the more its resistance changes. Measuring the changing resistance in the strain gauge can be used to measure the corresponding force exerted on the load cell bending it [3].

In order to accurately measure the small changes in resistance in the strain gauge, it is used to form a Wheatstone bridge shown below. The other resistors can be additional strain gauges or just any resistor with a known value.

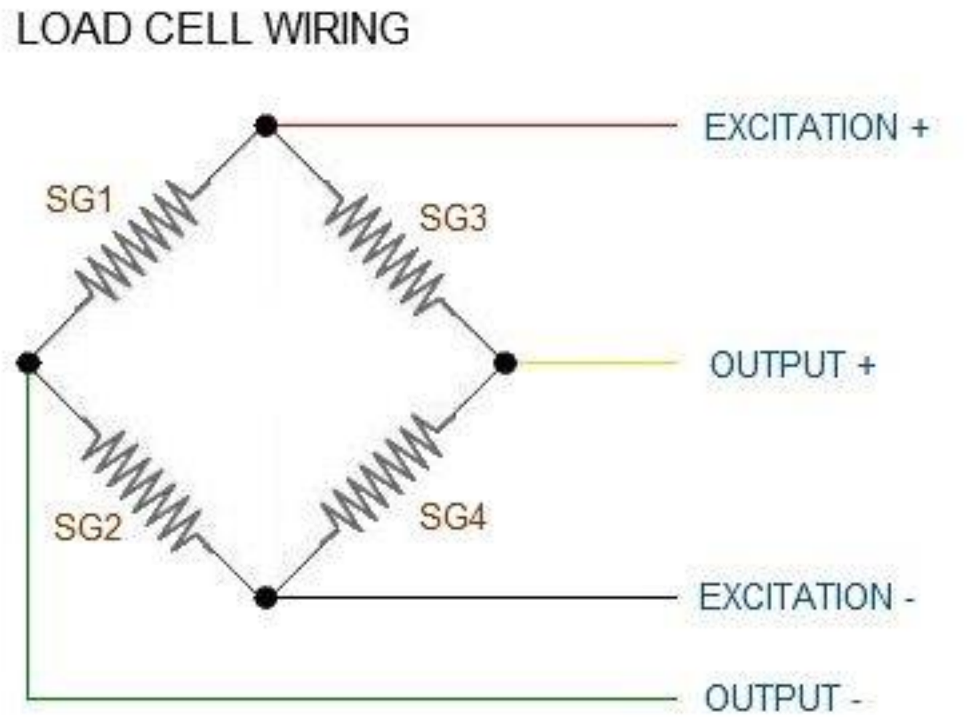


Figure 9 Load Cell Wheatstone Bridge

A voltage is excited across 2 ends the Wheatstone bridge, and an output voltage is read across the other ends of the bridge. This setup allows for small changes in resistance of the strain gauge to easily be measured [3]. The Wheatstone bridge consists of two voltage dividers: SG4 and SG3 form one, SG2 and SG1 form the other.

V_{output} = [(SG4/ (SG4 +SG3) - SG2/(SG1 +SG2)]

When SG1/SG2 = SG3/SG4 The output voltage is 0. When this happens the bridge is said to be "balanced". When the resistance of a strain gauge changes the change is reflected by unbalancing the Wheatstone bridge and causing an output voltage to be generated. This change in output voltage can be used to determine the change in resistance of the strain gauge [18].

4.5 HX711 Chip

To make reading the changes in the output of the Wheatstone bridge easier and more compatible with Arduino we chose to use an HX711 load cell amplifier and 24- bit ADC. One side of the HX711 connects to the strain gauge Wheatstone bridge in the load cell as shown below. The first two terminals: E+ and E- connect to the other end of the Wheatstone bridge to excite it with 5 volts. The: S+ and S- terminals connect to the other two terminals of the Wheatstone bridge and read the output voltage. This output voltage is amplified by a gain of 64 or 128 and converted to a digital signal by the chip. On the Arduino side of the chip there is a VCC input connected to the 5V pin of the Arduino, this voltage is used to excite

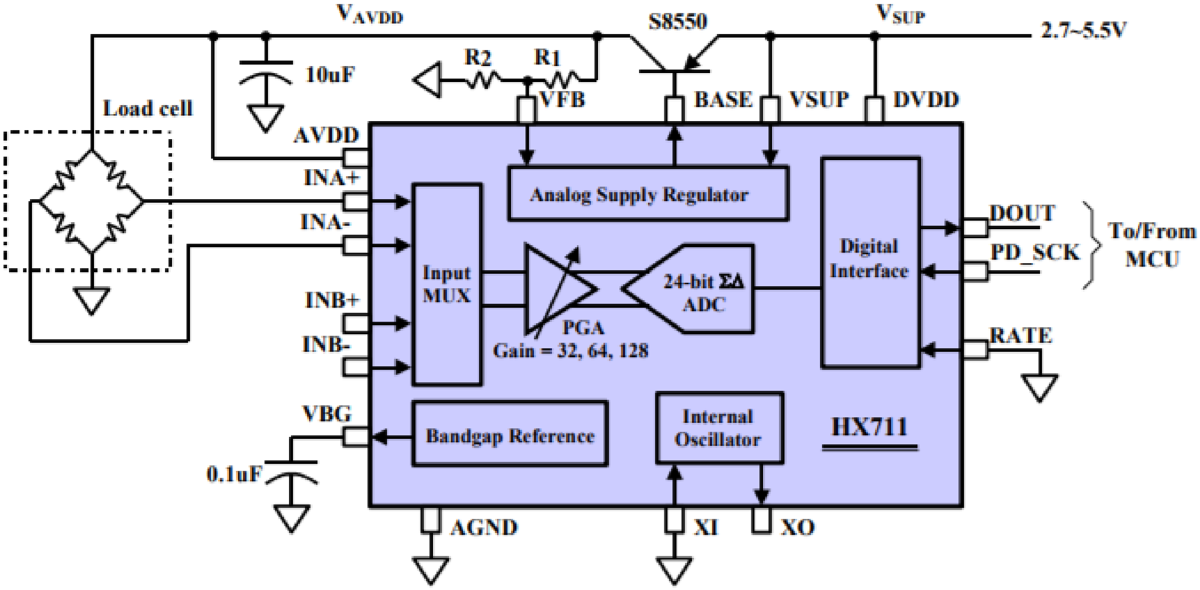


Figure 10 Schematic of HX711 Chip

the Wheatstone bridge at the E+ and E- terminals [1].The GND terminal of the chip connects to the ground pin of the Arduino to ground the circuit. The two other Arduino connects are to DT and SCK. As these pins are digital they can be programmed as any digital pin on the Arduino. We chose to use digital pin 3 for DT and digital pin 2 for SCK. DT is the digital output signal of the chip. SCK is connected to the clock of the Arduino and is used in the 24-bit ADC. The digital output of the chip is converted to kilograms by the HX711 library. The code we used to calibrate load cells and measure weight was designed specifically for the HX711 and can be seen in Appendices A and B.

4.6 5kg Load Cell

The first strain gauge we tested was a 5kg block load cell shown below. It contains four strain gauges.

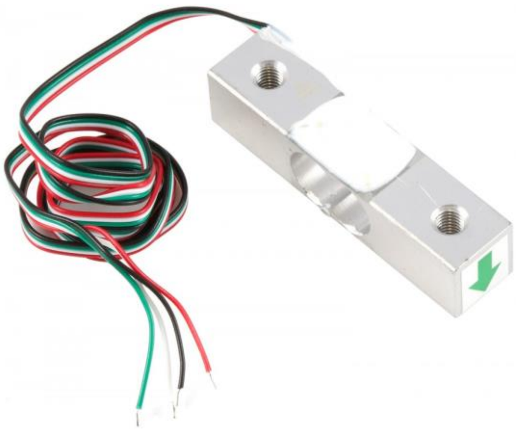


Figure 11 5kg Load Cell

For our initial testing of the strain gauge we screwed plywood boards on either end of the load cell, and clamped one piece of plywood to the table to stabilize it. The load cell was then wired to the HX711 amplifier which was wired to the Arduino. First, I calibrated the setup using a 500 gram calibration weight and the calibration code found in Appendix A. Once I calibrated the program I ran a weight measurement program, found in Appendix B, and tested different calibration weights. The weight measurements were very accurate, with only an occasional error of 1 gram. The load cell easily detected changes as small as 10 grams. This was well within the margin of error needed to detect a 20 gram sip of water. We had proven the load cell worked, but we needed a compact coaster design.

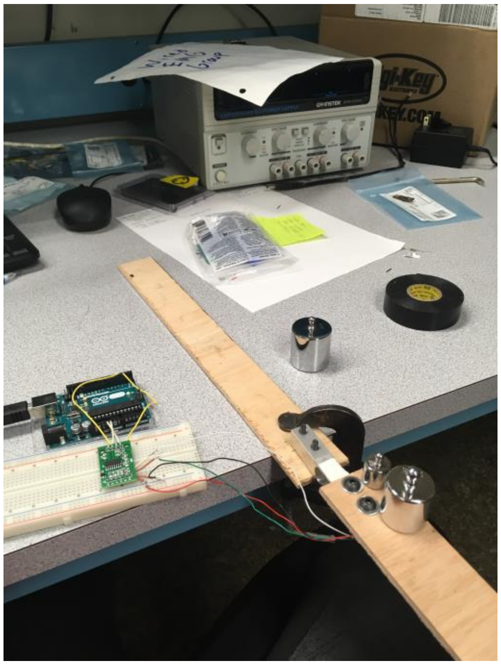


Figure 12 5kg Load Cell Test

4.7 50kg Load cell

Looking for smaller alternatives to the block load cell we tested a smaller 50kg load cell. This load cell is a small square with a u of bent metal protruding from it. The force of the weight should be exerted on the top of the u. This force slightly stretches out the metal underneath the u, which in turn stretches the strain gauge that is placed there, changing its resistance.

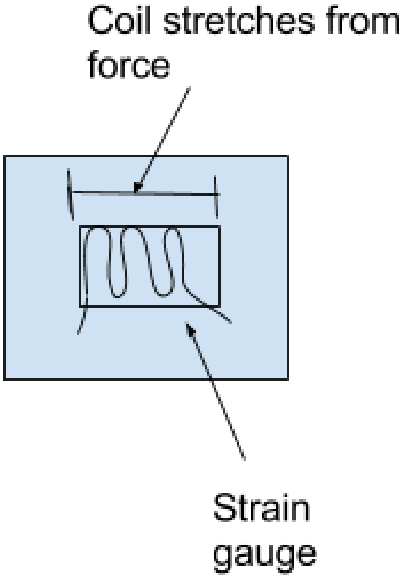


Figure 13 50kg Strain Gauge

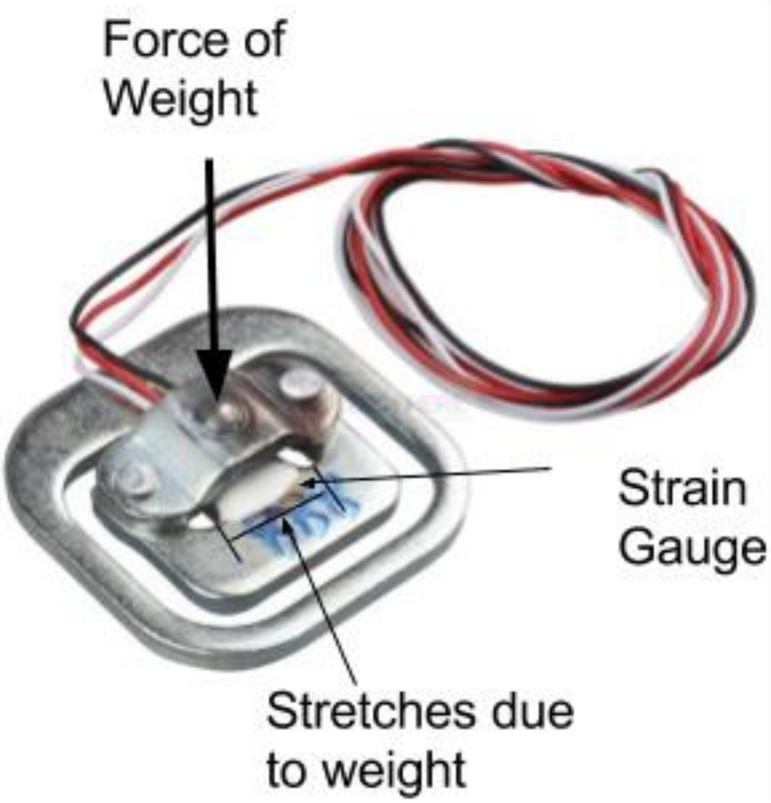


Figure 14 50kg Load Cell

Each 50kg load cell contains only a single strain gauge. Because of this 2 or 4 load cells must be used to create a half or full Wheatstone bridge. Alternatively resistors of known values can be used to complete the Wheatstone bridge. For the purpose of testing we chose to use two 50kg load cells to complete a half Wheatstone bridge. We glued their bases to the bottom of the 3d printed coaster base used to test the force sensitive resistors. We designed and 3-d printed a lid to rest on top of the load cells. This ensured that the weight would rest fully on the top of the u's of the load cells.

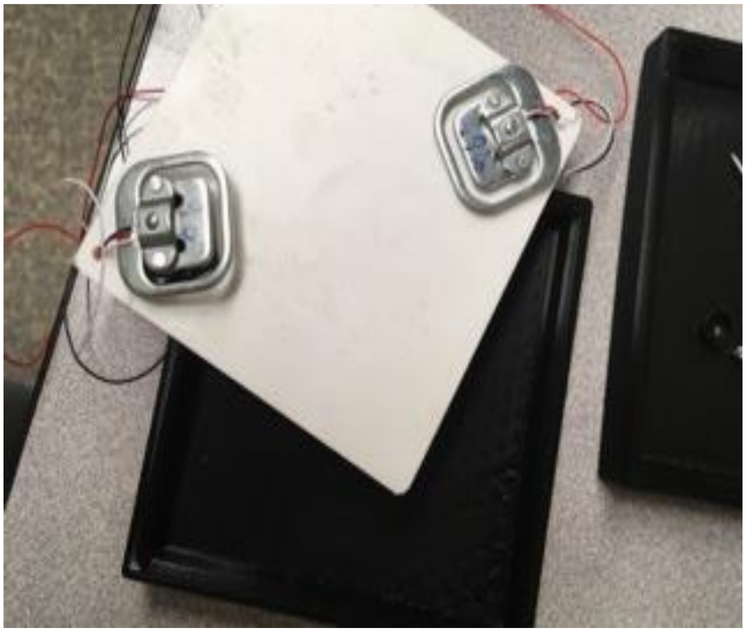


Figure 15 50kg Load Cell Coaster

The circuit we used connected black to red and red to black which was attached to the excitation ports of the HX711 and the output signal was read from the white wires. Using a half

Wheatstone bridge caused the weight to not be read correctly. The weight measured of one load cell was subtracted from the weight measured by the other, causing inaccurate results. In order to combat this problem we chose to test one load cell at a time and check for consistency. The measured weight on each load cell varied as much as 70 grams. This inaccuracy is likely due to the load cell being made for heavier weight applications such as bathroom scales. The u shape is not as responsive to small changes in weight as the block shape. Due to this inaccuracy we chose not to pursue using 50kg load cells any further, instead deciding to continue to use the 5kg block load cell.

4.8 Wooden Coaster design

After running initial tests on the 5kg and 50kg load cells we decided to use the 5kg load cell in our design due to its accuracy. We chose to build the first coaster using the 5kg load cell out of wood. We chose wood because it is easy to prototype with. It is easily cut with a laser cutter and can easily be screwed onto the block load cell. The design can be seen below.

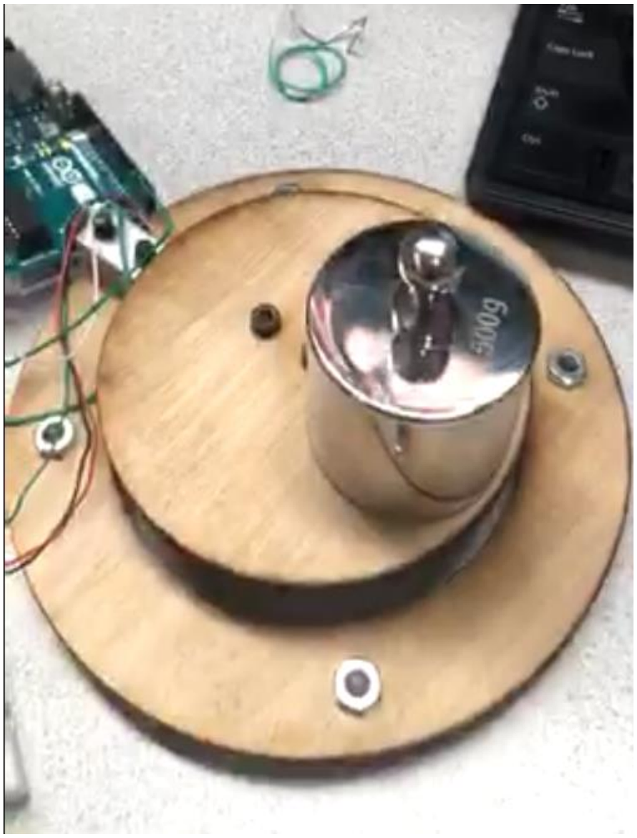


Figure 16 5kg Load Cell Wooden Coaster Design

We used a laser cutter to cut out two concentric circles of wood. The larger circle acted as the base for the 5kg load cell. The smaller circle was screwed onto the top of the load cell to support the cup or bottle. Our initial testing of this design quickly showed many issues. Although weight was being measured it was not consistent or accurate. Placing the weight at different places on the coaster resulted in different weights by as much as 50g. Considering the average sip of water is 20g this was too inaccurate to be used to measure water intake. This inaccuracy was caused by a few different design flaws. The first issue was the flexibility of the wood. The wood bent when force was exerted on it by the calibration weights, and it bent more when the weight was placed further from the load cell. This bending increases torque on the load cell causing the strain gauge to bend in ways not directly correlated to the force of the weight, causing inaccurate readings. The other cause of inaccuracy is the circle holding the weight is placed directly on top of the 5kg load cell instead of having spacers separate them. This causes some force to be transferred directly to the bigger circle supporting the bottom of the load cell. Using spacers between the weight bearing circle and the load cell ensures that the force of the weight is fully applied directly to the load cell. This allows for accurate weight measurements.

4. 9 Glass coaster design

To combat the issues with the wooden coaster we decided our next design would be made of a stiff material and include a spacer between the platform holding the weight and the load cell.

To test if these design improvements would be effective we used a 5kg Taylor Kitchen scale as shown below.

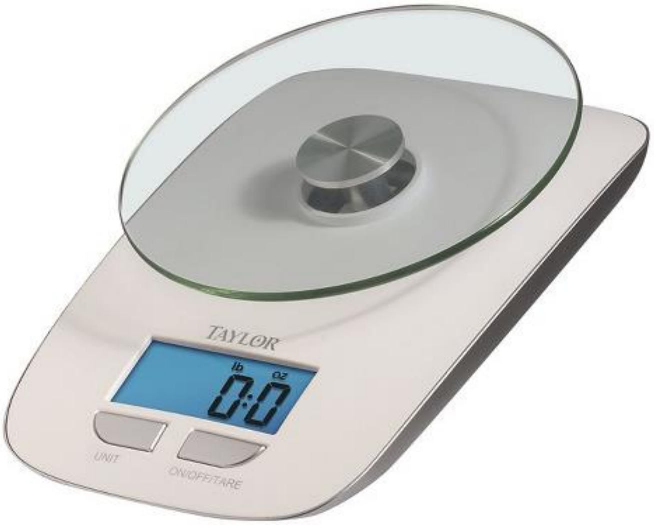


Figure 17 5kg Taylor Kitchen Scale

Inside of the scale is a 5kg load cell. The glass surface shown in the picture above screws directly into the load cell. The inside of the scale can be seen below.

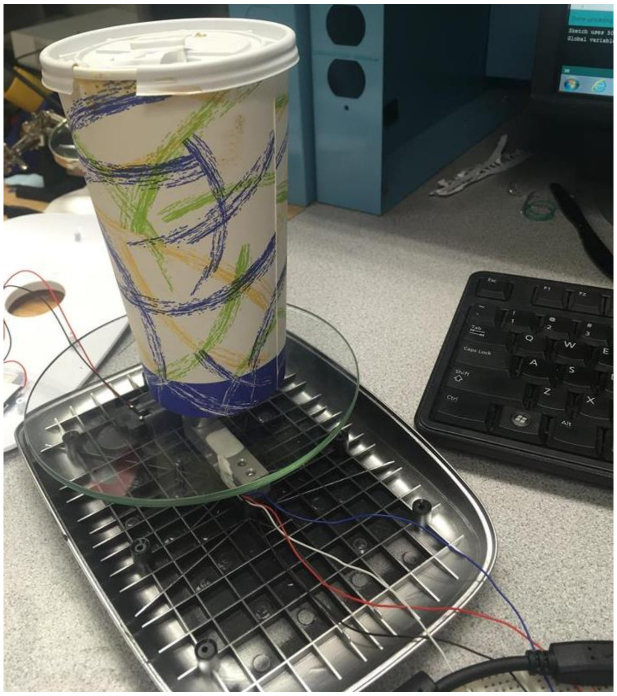


Figure 18 Scale Test

While leaving the scale intact we connected its 5kg load cell to the HX711 chip and Arduino and ran the weight measurement program we used to test our previous coaster designs. It was important to confirm the scale worked properly with our software to ensure the weight measurement inaccuracies we experienced with the previous design were caused by hardware issues and not software. The test measures weight accurately to 1 gram. This proved our previous inaccuracies were caused by hardware design and not software. We proceeded by removing the 5kg load cell and glass weighing surface from the base of the scale and attaching it to the wooden circular base we used in our previous coaster design as shown below

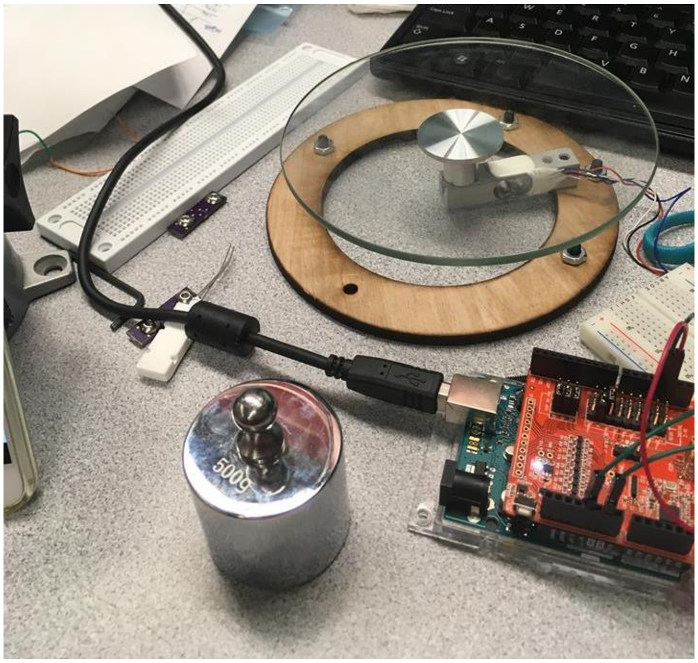


Figure 19 5kg Load Cell Glass Coaster Design

We tested the design using calibration weights and found it was accurate to 1 gram. The main issue was the wooden base was too flexible and caused the glass plate to tilt and the weights to slide off when placed too close to an edge. The design is also much larger than an actual coaster and must be miniaturized.

4.10 Small Coaster design

To miniaturize our coaster design we decided to use a small load cell extracted from the H2O Pal water bottle shown below.

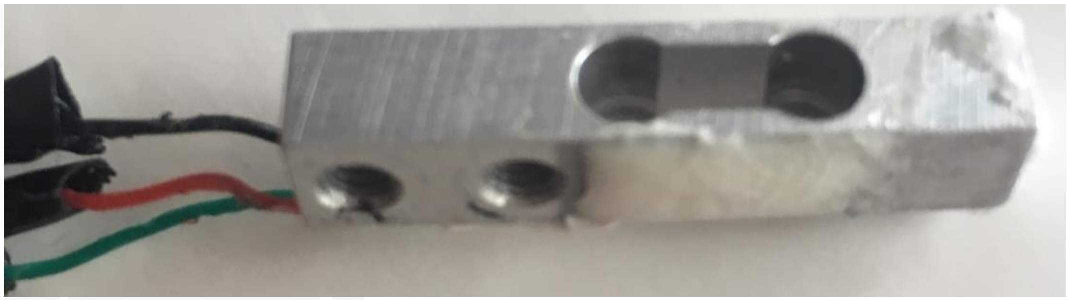


Figure 20 Small Load Cell

With its dimensions 30mm x 6mm x 6mm it is half the size of the previous 5kg load cell which measures at 60mm x 10mm x 10mm. This drastic decrease in size allows the coaster design to become much smaller and closer to an actual coaster size. With this in mind we chose to use a standard 4”x4” square coaster design. The coaster consists of two 3D printed 4x4” squares with the load cell resting between.

The load cell is screwed directly into the square bracket shown below using 3mm screws:

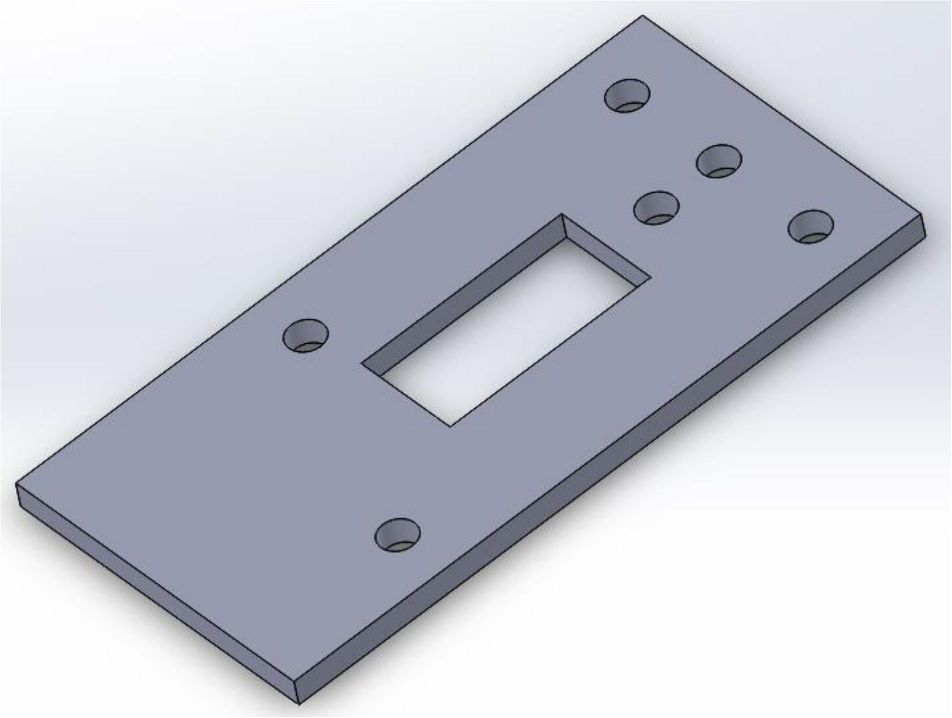


Figure 21 Load Cell Bracket

The square bracket is attached to the bottom 4x4 square by 2 pins and two 3mm screws. This is shown below:

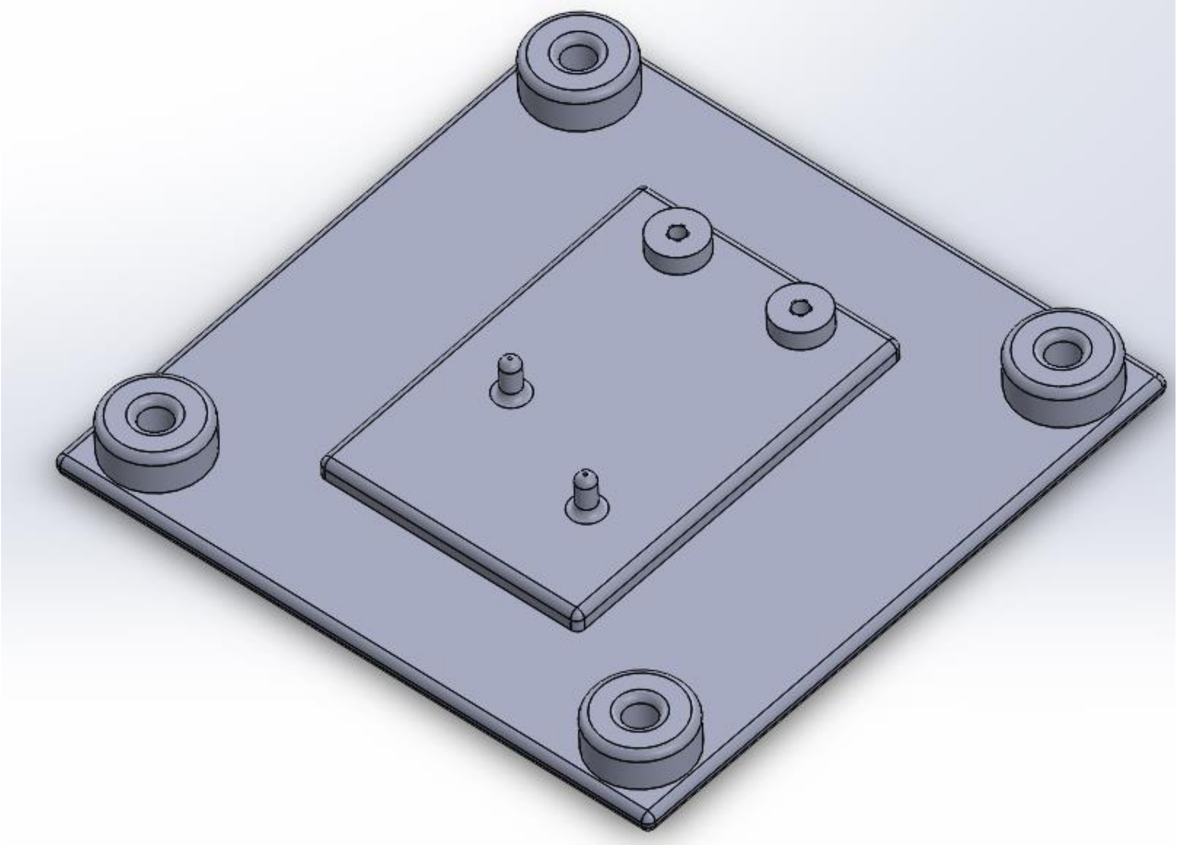


Figure 22 Coaster Bottom Plate

The small load cell only has screw holes on one end to attach the load cell to a static base. Unlike the previous 5kg load cell it has no screw hole to attach it to the load. To combat this the upper 4x4 square contains a small square extruded an 1/8th of an inch shown below. This extrusion rests on the top of the load cell when the coaster is assembled. This allows all the weight on the coaster to only be transferred to a single spot on the load cell allowing for a more accurate measurement.

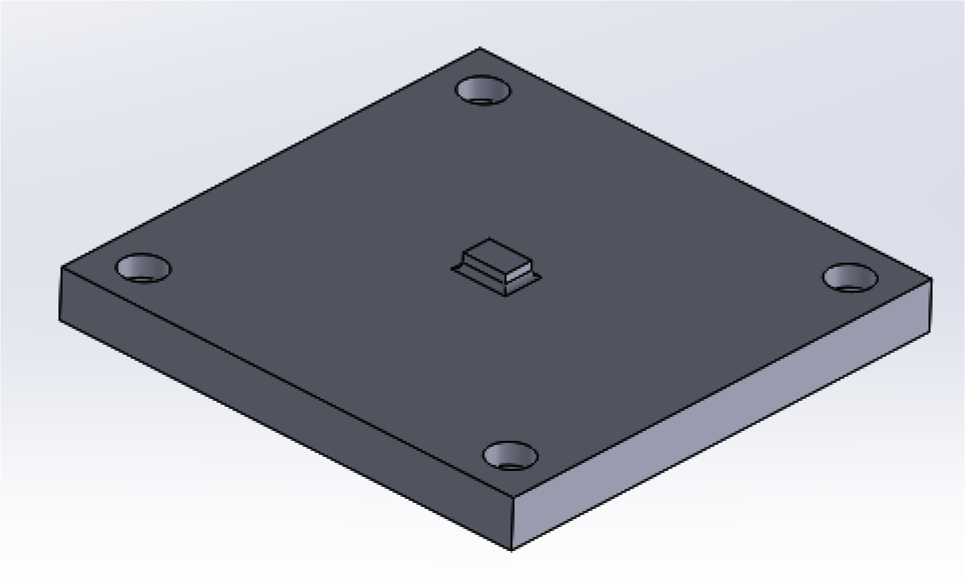


Figure 23 Coaster Top Plate

In the first version of the design the upper and lower 4x4 squares were attached to each other by 4 pins. Before 3D printing we replaced the pins with sockets for 6mm shoulder screws. This allows the top and bottom pieces to be securely fastened to each other while also allowing the top piece to move freely. This allows all weight place on the coaster to be transferred directly from 8mm square extrusion to the load cell to allow for more accurate measurements.

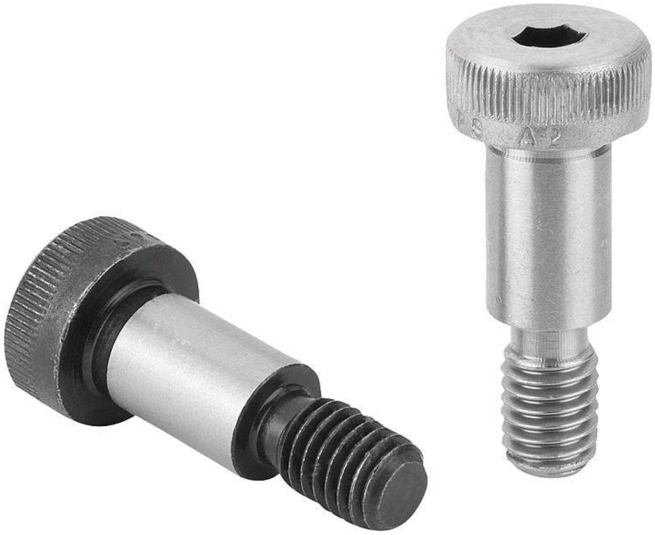


Figure 24 Shoulder Bolts

The threaded part of the screw is screwed into the socket of the bottom plate, and the unthreaded part slides freely in the socket of the top plate.

An assembled photo of the coaster is shown below:

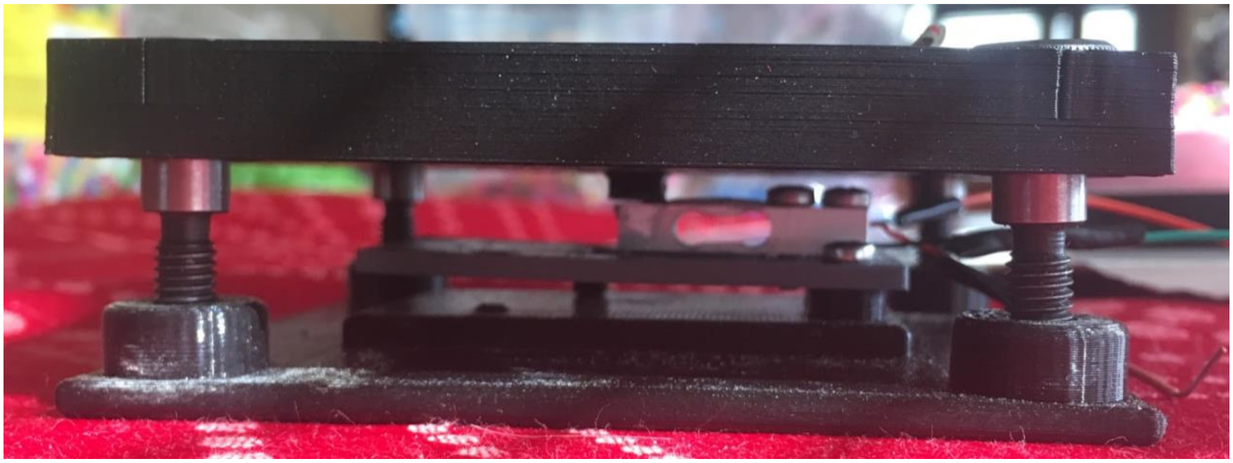


Figure 25 Small Load Cell Coaster from Side

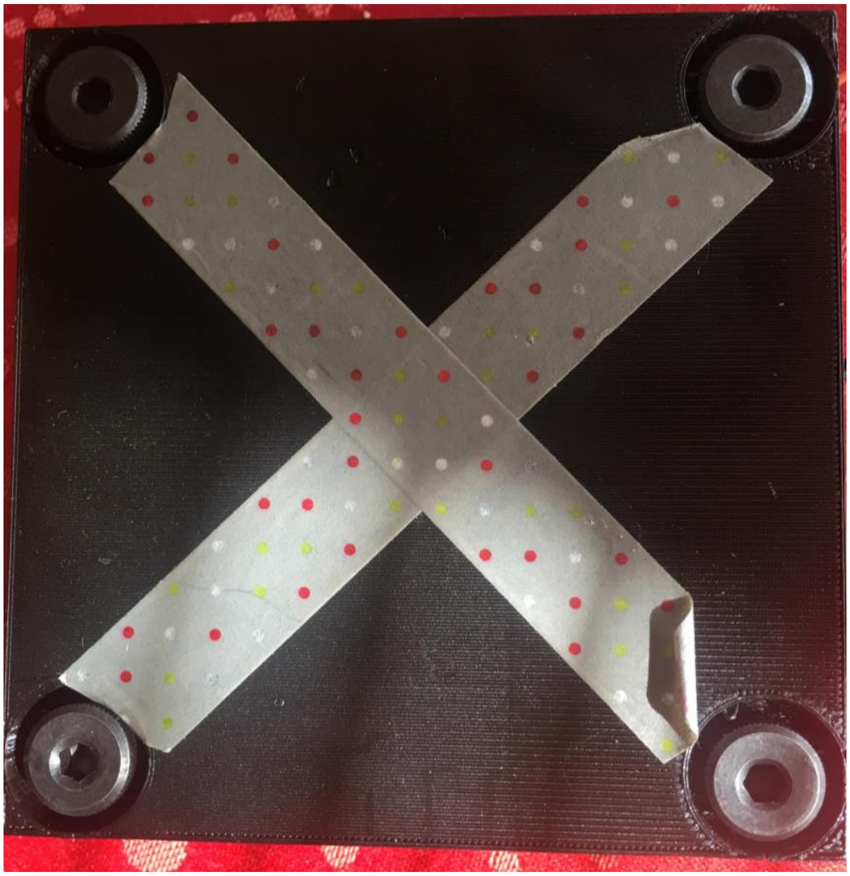


Figure 26 Small Load Cell Coaster from Top

After testing this design we ran into a few issues. Any pressure not exerted directly in the middle of the top plate causes the plate to tilt. As the plate tilted it rested on the shoulder screws, causing some of the force from the weight to be transferred onto the shoulder bolts. In order to measure weight accurately 100% of the force of the weight needs to be exerted directly onto the load cell. To fix the problem we removed the shoulder bolts and super glued the top of the scale directly onto the small load cell. This was very effective and caused the coaster to measure weight accurately to +/-1 gram. Despite the accuracy the superglue was not a long term solution. It was easy to crack the glue and cause the top plate to become disconnected from the load cell. We considered drilling a hole directly into the top of the load cell as this would allow the top plate be screwed directly into the load cell, just like the glass coaster design at a smaller scale.

Unfortunately we worried that adding an additional hole would compromise the accuracy of the load cell and make it unusable. The second problem was due to the thin nature of the wires attached to the load cell. The wires would constantly snap off the connectors they were soldered to, forcing them to constantly be resoldered. Every time the load cell lead wires snapped they became shorter and shorter, making soldering more and more difficult until it was essentially impossible. Due to all of these difficulties we decided to switch back to using the 5kg load cell used in the Glass Coaster design, and modifying the small coaster design to fit accordingly.

4.11 Final Working Coaster design

For the final working design we switched back to using the 5kg load cell from the glass coaster design and adapted the small coaster design to fit it. To ensure the end of the load cell was centered in the middle of the coaster we changed the coasters dimensions to 4.5x4.5 in. Next we moved the screw holes and platform that attach the load cell to the back of the coaster as shown below:

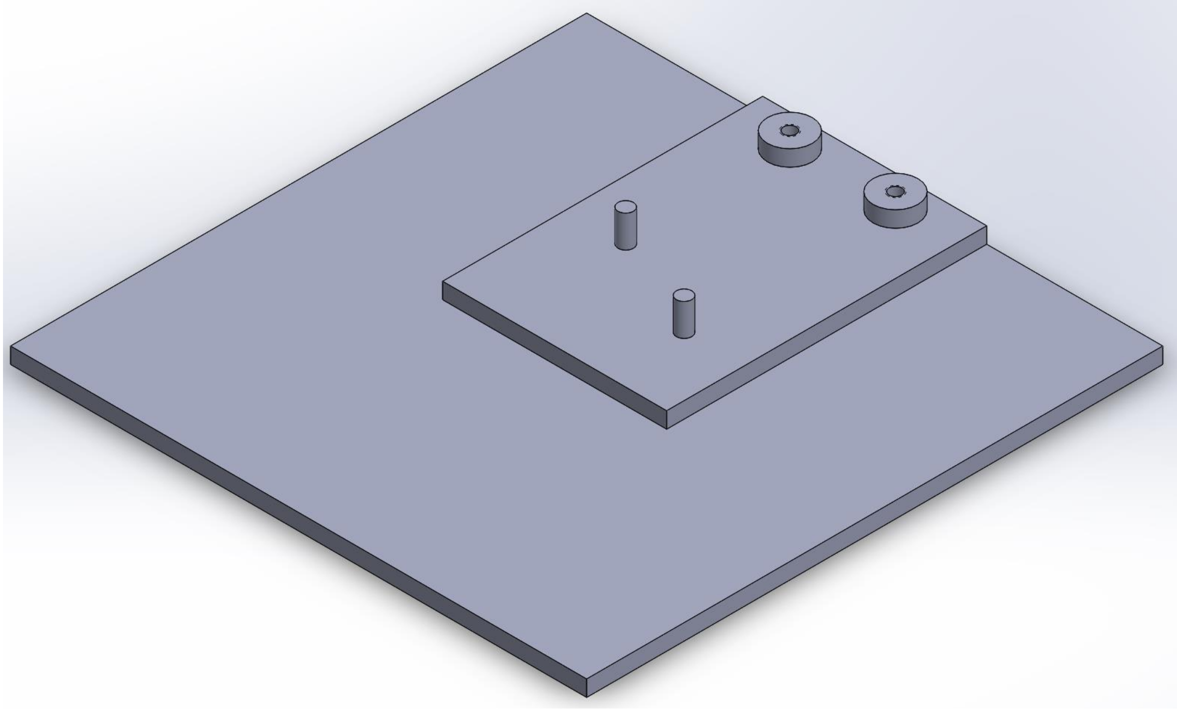


Figure 27 Final Design Bottom Plate

For the top plate we changed the small 8mm extrusion that pressed onto the small load cell to an 8mm screw hole. We also removed all holes for shoulder bolts on both top and bottom plates as they wouldn’t be needed. The new top plate can be seen below

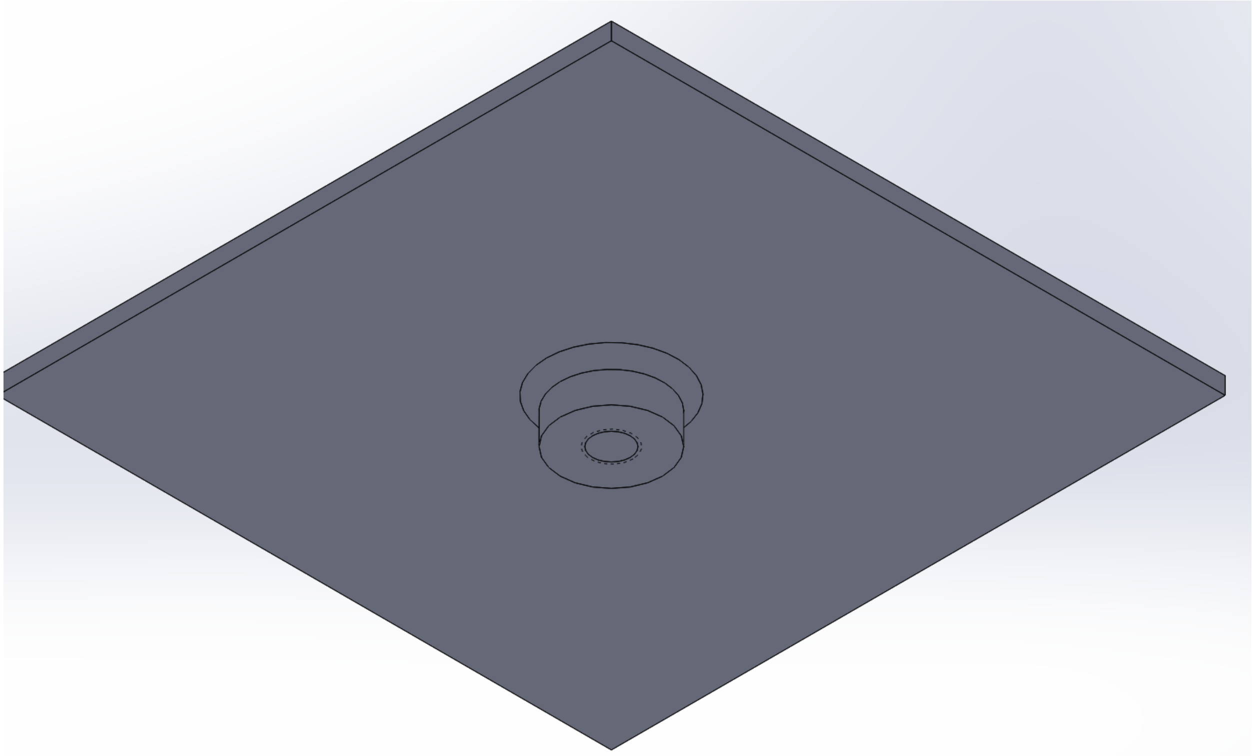


Figure 28 Final Design Top Plate

For the load cell bracket we increased the screw hole sizes to 4mm to match those on the 5kg load cell. We also removed one end of the bracket so the load cell wouldn’t transfer any force onto it while it was bending. The new bracket can be seen below:

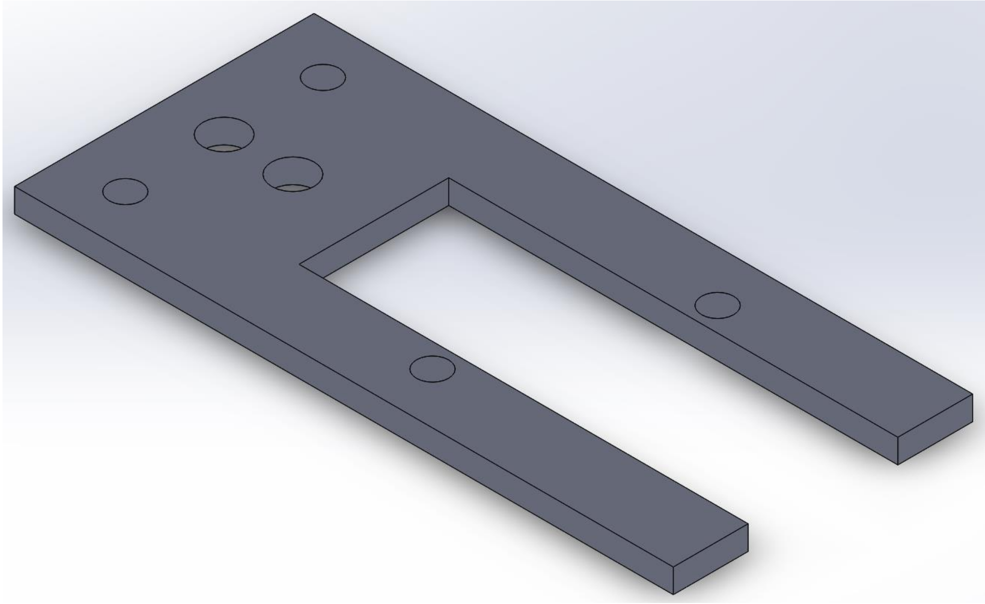


Figure 29 Final Design Load Cell Bracket

After initially assembling the coaster we realized that the 5kg load cell bends much more than the small load cell does and needed to be mounted higher to prevent the load cell from resting on the ground when weight was placed on it. To fix this problem we added a nut and washer to each screw to elevate the load cell as shown below. This fixed the problem and the load cell no longer touched the bottom plate when bent.

The other issue we had was screwing the 8mm screw into the screw hole on the upper platform. Due to the nature of plastic threads in 3d printing they can easily be warped by taking the screw in or out by excess torque that might bend threads. When weight was placed on the top of the plate the screw began to tilt and wobble in its screw hole due to the added torque. To eliminate this problem I lined the inside the screw hole on the top plate with superglue before screwing in the 8mm screw and making sure it was perfectly perpendicular to the plate. After the glue dried the screw stayed in solidly and no longer wiggled side to side.

The assembled design can be seen below:

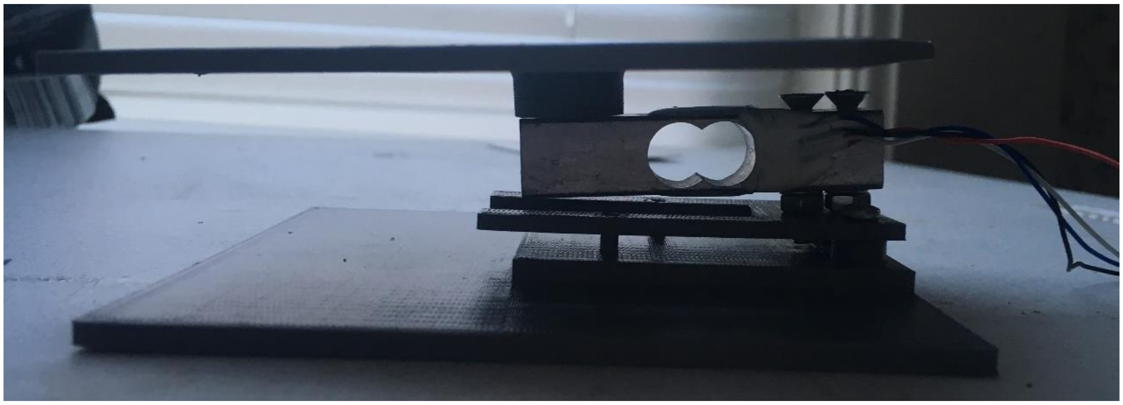


Figure 30 Final Working Coaster from Side

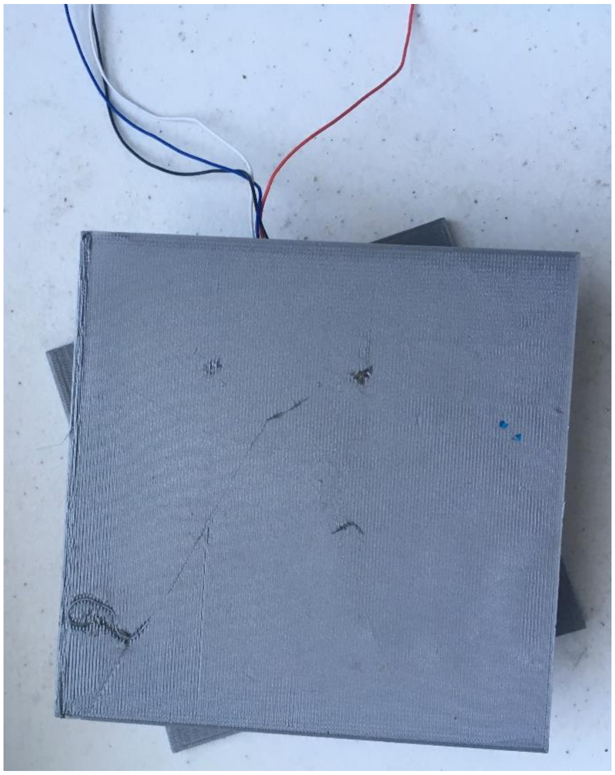


Figure 31 Final Working Coaster from Top

With these revisions we have a working coaster. It is not as accurate as the glass coaster, but by smoothing the data it is accurate to +/-2 grams. Due to an ounce of water being about 30 grams, this accuracy is more than enough to measure someone's water intake.

4.12 Application

The application was created using the Android Studio IDE. We first created a basic application that would receive data from our Redbear BLE shield and display it on the screen. To send data, we used the Bluetooth low energy SDK for Arduino library, which only allows for data to be sent one byte at a time. Because we needed to receive values of a float datatype, the data was sent in groups of four bytes, one byte at a time, since a float is four bytes large. Afterwards, a basic user interface was created for the sole purpose of displaying the given data in a working Android application. Once that was finished, we implemented functionality to be able to track user drinking levels, adding in code to determine when a given user was drinking water.

The application monitors the data until it detects that the weight value remains stable and compares that value to a previously saved weight measurement to find the difference in water. Next, we added database functionality to track hydration levels over long periods of time. For this, we used the Room Persistence library and created a database for two different data types. The first datatype was a User data type, which we used to hold all the user information, such as login information and basic measurement data. Each user was given a newly generated User ID which linked the user to the Data datatype. The Data datatype consisted of a User ID to link the data to a user, a water total, that would total the water drank in ounces for a given day, and a date, which was stored as a String for easier data retrieval. When the application was able to receive data and monitor water consumption properly, we focused on adding in other generic application functionality, such as user accounts, the ability to modify user information, and the ability to set a daily goal.

First-time users are first prompted with an initial login screen with a field for an email address, a field a password, a button for logging in, and a button for creating a new account.

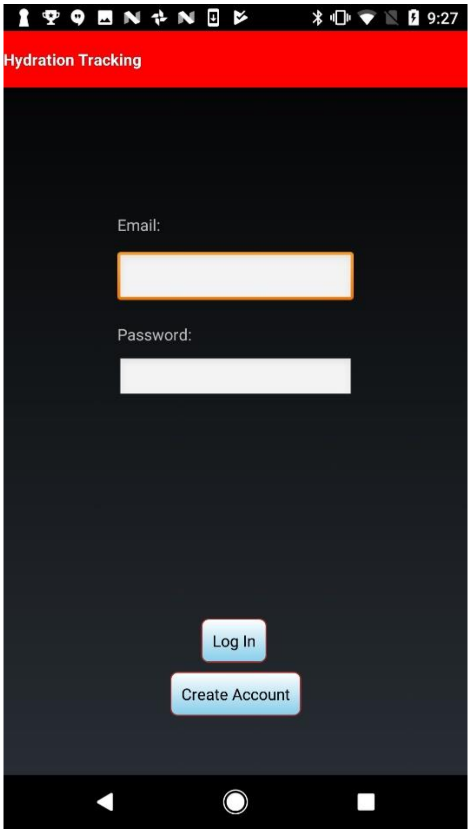


Figure 32 Application login page

Upon clicking the “Create Account” button, new users are prompted to fill in basic user information allowing them to create a generic profile. Creating this account prompts the user to a new screen, where additional personalized information can be inputted.

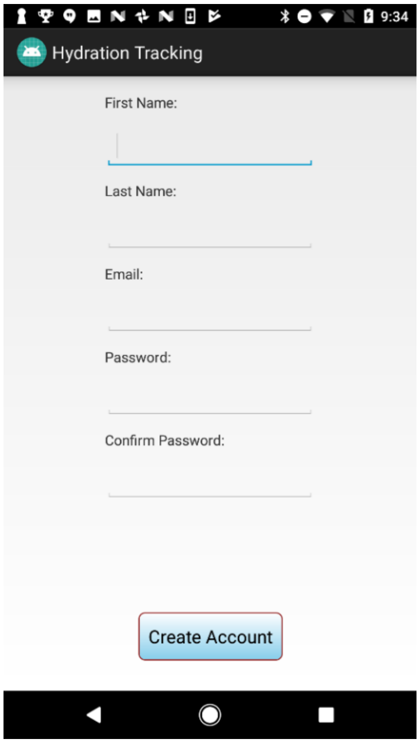
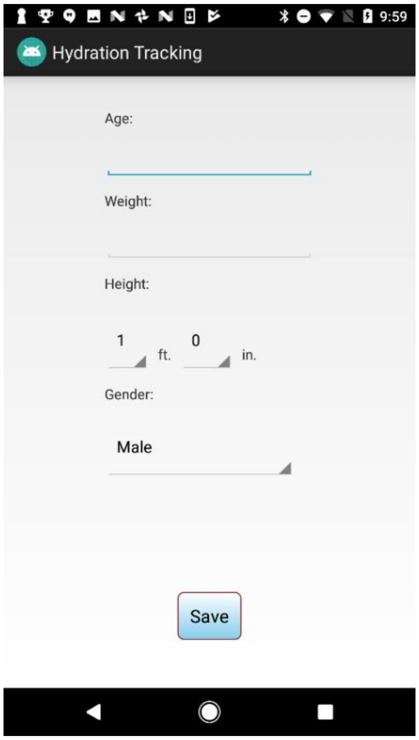


Figure 33 Account creation page



Age:

Weight:

Height:

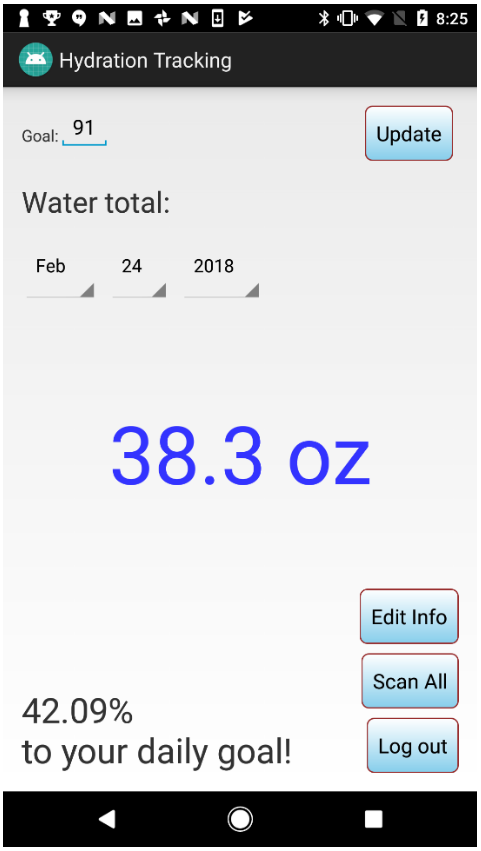
1 ft. 0 in.

Gender:

Male

Figure 34 Additional information page

Once a user profile is created, logging in displays the user home page containing a changeable goal value, a display of their total water for the day, a display of their percentage progress to their daily goal, a button to edit information, a button to scan for nearby BLE devices, a button to log out, and a button that will update the current page to display updated goal or total for a chosen day.



Goal: 91

Water total:

Feb 24 2018

38.3 oz

42.09%
to your daily goal!

Figure 35 User home page

4.13 Wireless charging

To ensure the coaster can always be used, even while charging, we chose to implement wireless charging our design. We chose to use a common standard for wireless charging called Qi. The system uses electromagnetic inductance to transfer power from a transmitter to receiver and is the standard used in many wirelessly charging phones. The transmitter, shown below, contains a transmitting coil that is 2 layers and 10 turns this results in an inductance of 24µH. This coil is shown as Lp in the diagram below. The capacitor, Cp shown in the diagram is 200nF. The coil has a resonant frequency of 140kHz. [17]

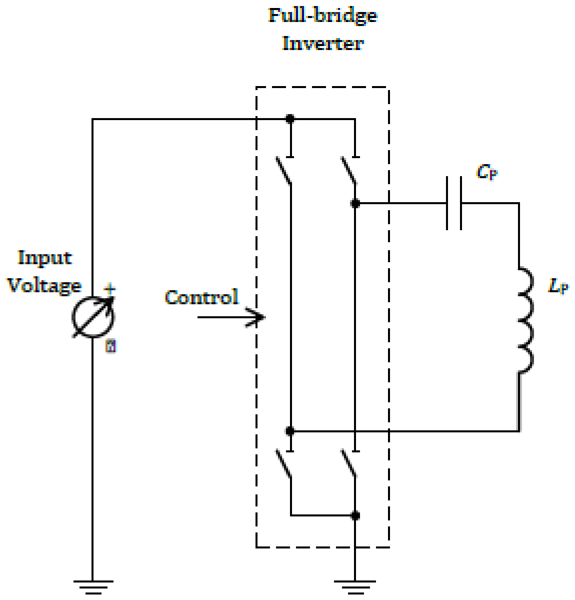


Figure 36 Qi Tx Cirtcuit

The input voltage is drawn from a micro USB charging cable that delivers power at 5V and 500mA. The full-bridge inverter converts the current to AC to allow for power transmission across the coil [17].

As shown in the diagram below the power is broadcast from the transmitter pad, shown below, to the receiver from coil Lp to coil Ls.

Figure 37 Qi Tx Pad

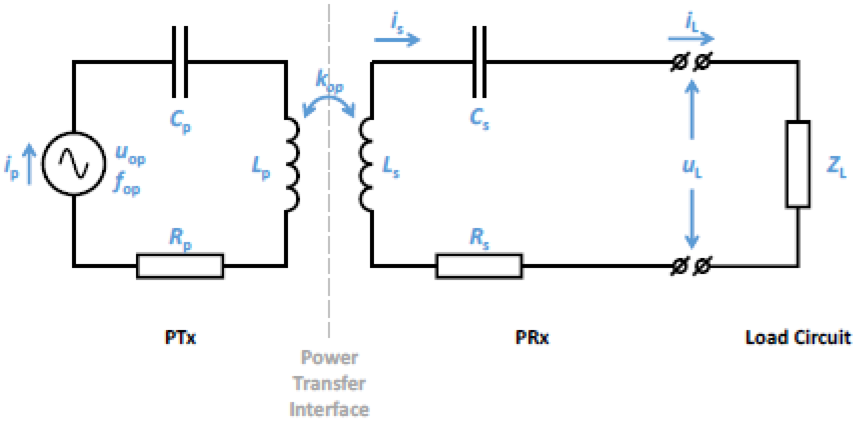


Figure 38 Qi Tx and Rx Circuit

The circuit on the left hand side represents a simplified transmitter and the circuit on the right hand side represents a simplified receiver [17]. For the receiver we chose to use a YKing micro USB B Qi Wireless Charger shown below.

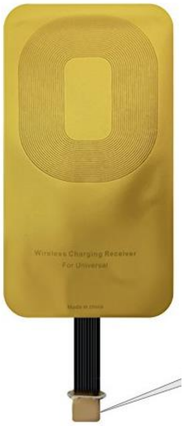


Figure 39 YKing micro USB B Qi Wireless Charging Rx

This is a simple circuit diagram of the receiver:

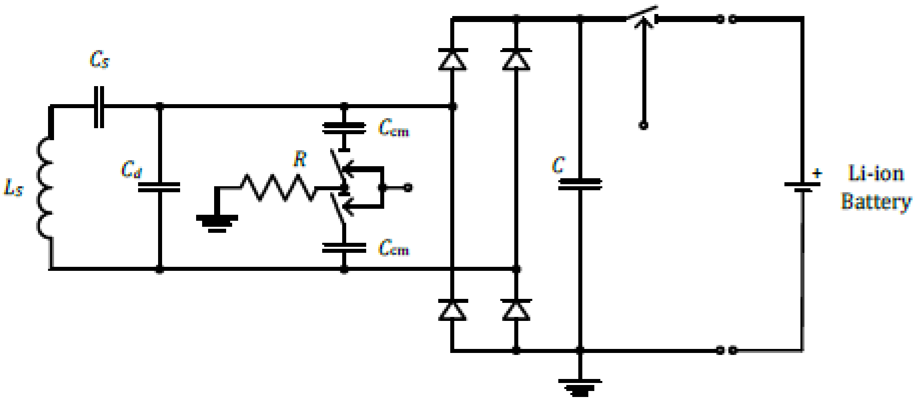


Figure 40 Qi Rx Circuit

The receiving coil, Ls, is square and can be seen in the photo of the yellow receiver. It consists of 1 layer of 14 turns. The circuit is dual resonant which means it contains two capacitors, one attached to a switch as shown below [17].

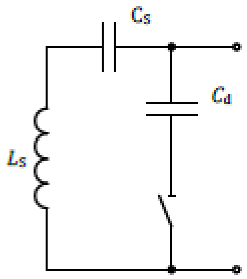


Figure 41 Dual Resonant Qi Rx Circuit

By opening or closing the switch connected to Cd the resonance of inductor Ls can be changed. When the switch is left open the inductor has a frequency of 20μH which is dependent on Cs capacitance value of 127nF. Cd has a capacitance of 1.6nF, and when the switch is closed Ls has an inductance of 15.3μH. The resonant frequency of the inductor is 100kHz. Being able to change the inductance allows for increased power transfer efficiency from the transmitter to receiver. The next stage of the receiving circuit is a full bridge rectifier with a 20μF capacitor to filter and change the signal to DC [17]. The circuit is then attached to a lithium ion battery which is charged with a voltage of 5v and a current of 500mA. For the lithium ion battery we first used an Anker Astro E1 battery power bank with 6700mAh. It connects to the Qi receiver using micro USB B.

Figure 42 Anker Power Bank

The Arduino is plugged directly into the USB port of a power bank to draw power. The rest of the system draws power directly from the Arduino. While active the Arduino draws a current of 50mA. The Hx711 amplifier chip draws current from the Arduino at a rate of 1.5mA. This results in a total system draw of about 51.5mA. We can calculate how long the battery will power the system by dividing the Amp hours of the battery by the current drawn by the Arduino. Hours = 6700mAh / 51.5mA = 130.01h. This means the system can be powered for 5.4 days before needing to be charged.

After testing the system we realized the Anker power bank was able to be charged by the Qi wireless receiver or power the Arduino, but was unable to do both at the same time. As the Arduino is always plugged in to and drawing power from the powerbank this made it impossible for the Anker power bank to ever charge. To fix this problem we switched to the Portable Charger iClever 15000mAh Power Bank, shown below, which can power a device while simultaneously charging itself. This ensures that power will always be provided to the Arduino.

Figure 43 iClever 15000mAh Power Bank

Due to the iClever's increased capacity compared to the Anker power bank, the coaster now needs to be recharged less frequently. Using the same computations as above and assuming the project still draws power at a rate of 51.5mA. Hours of Power = 15000mAh/51.5mA = 291.26 hours. If we divide this number by 24 we can see that the iClever power bank can power the coaster system for 12 days without needing to be recharged.

A picture of the assembled wireless charging system is shown below:

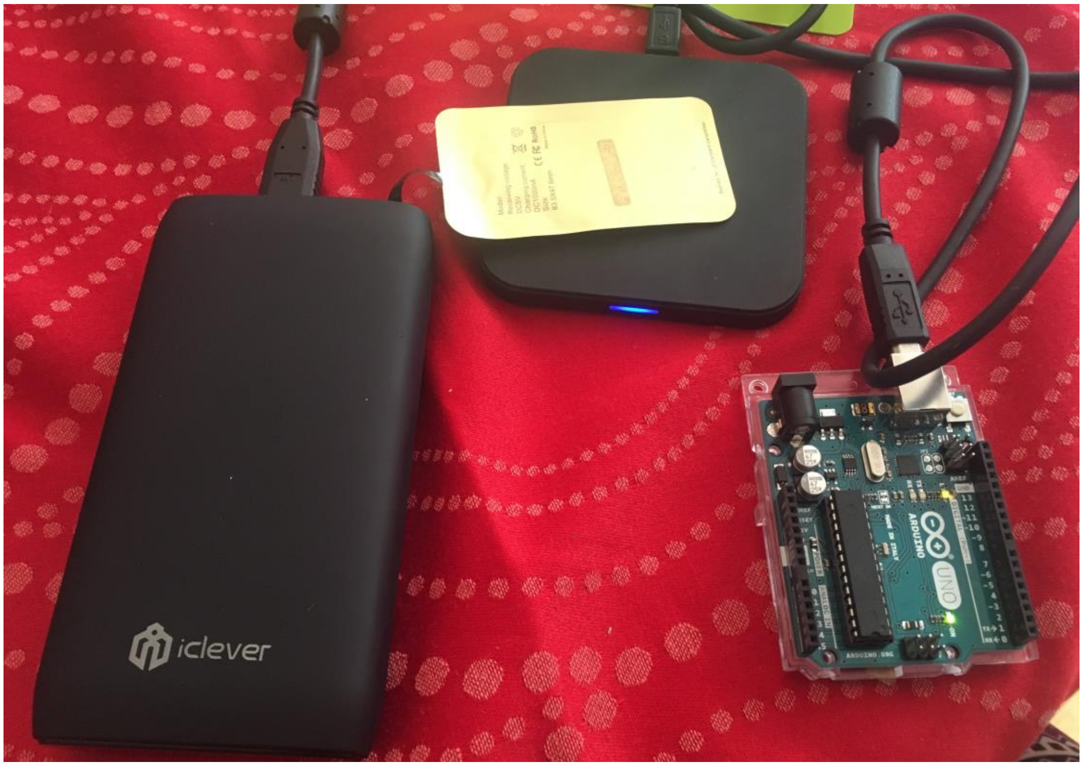


Figure 44 Assembled Qi Wireless Charging System

4.14 Looking Forward

As neither of us have much mechanical design experience, the mechanical design for the coaster was time consuming, haphazard, and very much a learning experience with each new iteration of coaster. This section will serve to describe our ideal final product. This is the coaster we would have created if given more time, resources, and mechanical experience such as 3d CAD design, machining skills, and 3d printing.

The first big change we would have made is to machine a smaller load cell. We would build a load cell identical to the 5kg load cell we are using in our final working design and shrink it down to the scale of the small load cell. A drawing of this load cell can be seen below:

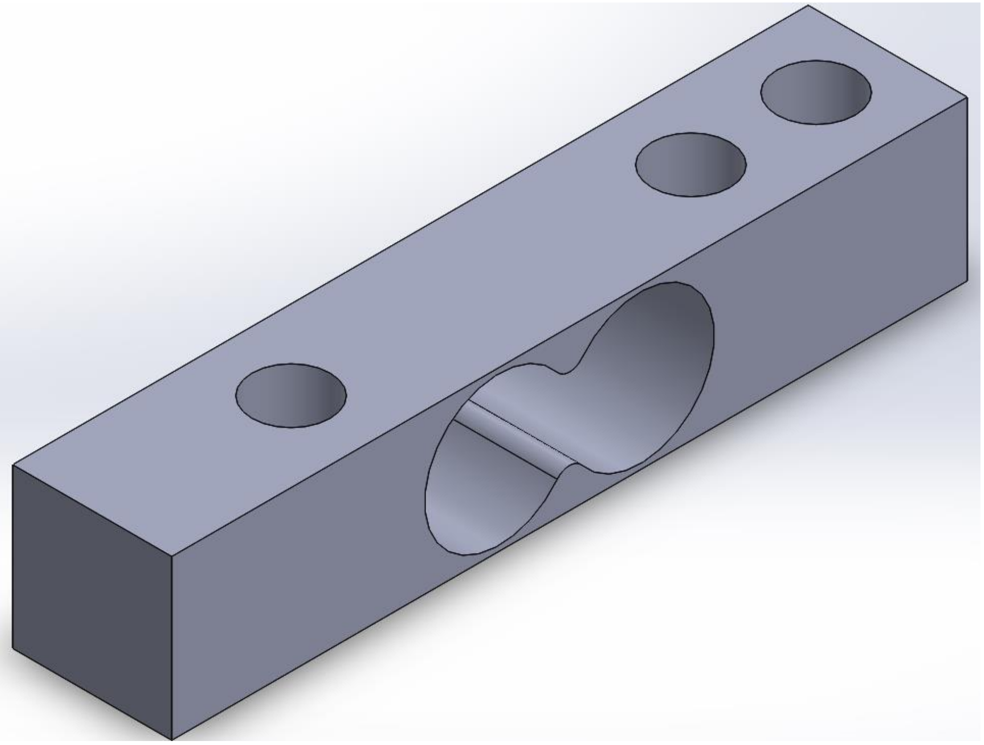


Figure 45 Proposed Small Load Cell Design

Because we would be using a small load cell we would use the bottom plate design used for the small load cell coaster. We would remove the screw holes and shrink the plate down to a size of 3x3in. The new design is shown below

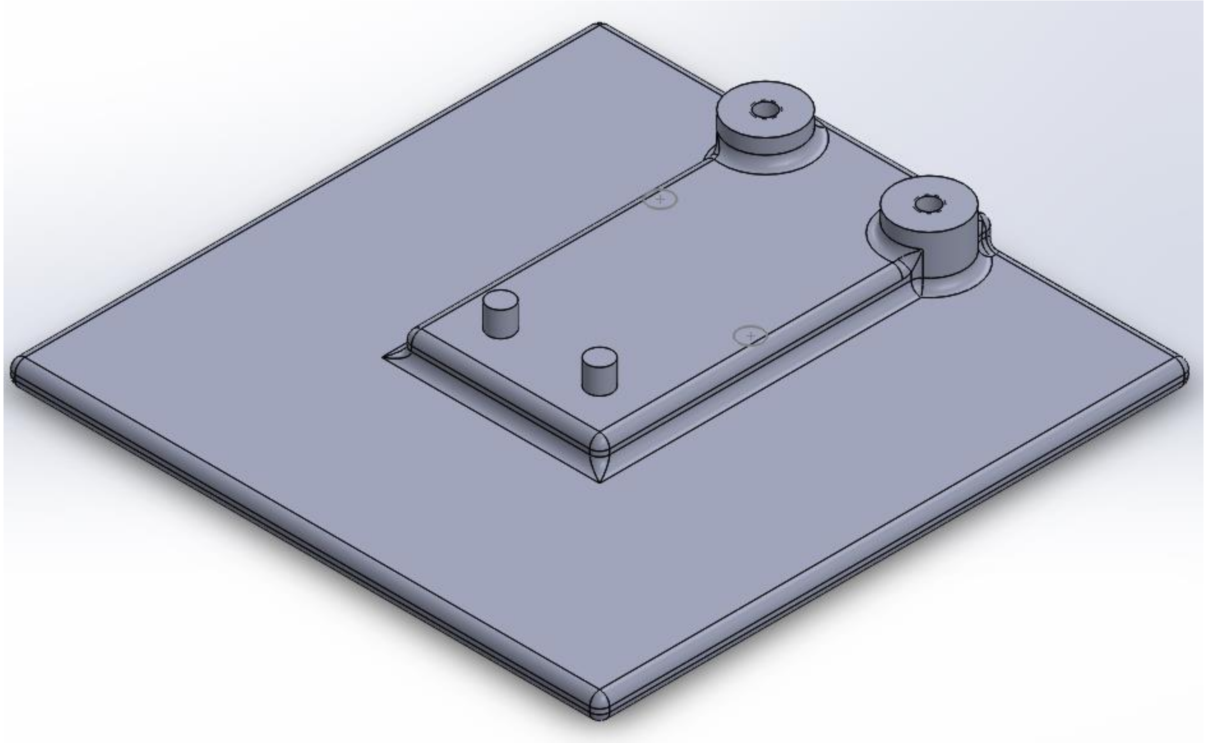


Figure 46 Proposed Bottom Plate Design

For the top plate we would use the same concept as our final design. It would contain one screw hole in the center for the load cell to attach to. Instead of a square plate we would switch to a round one with a 4.25 inch diameter. This would allow for the top to be screwed on without worrying about the square top perfectly aligning to the square bottom. We would also add sides from the top that don’t go all the way down to the bottom. This would protect the components inside the coaster without causing any weight to accidentally be transferred to the sides. A picture of this new design is shown below:

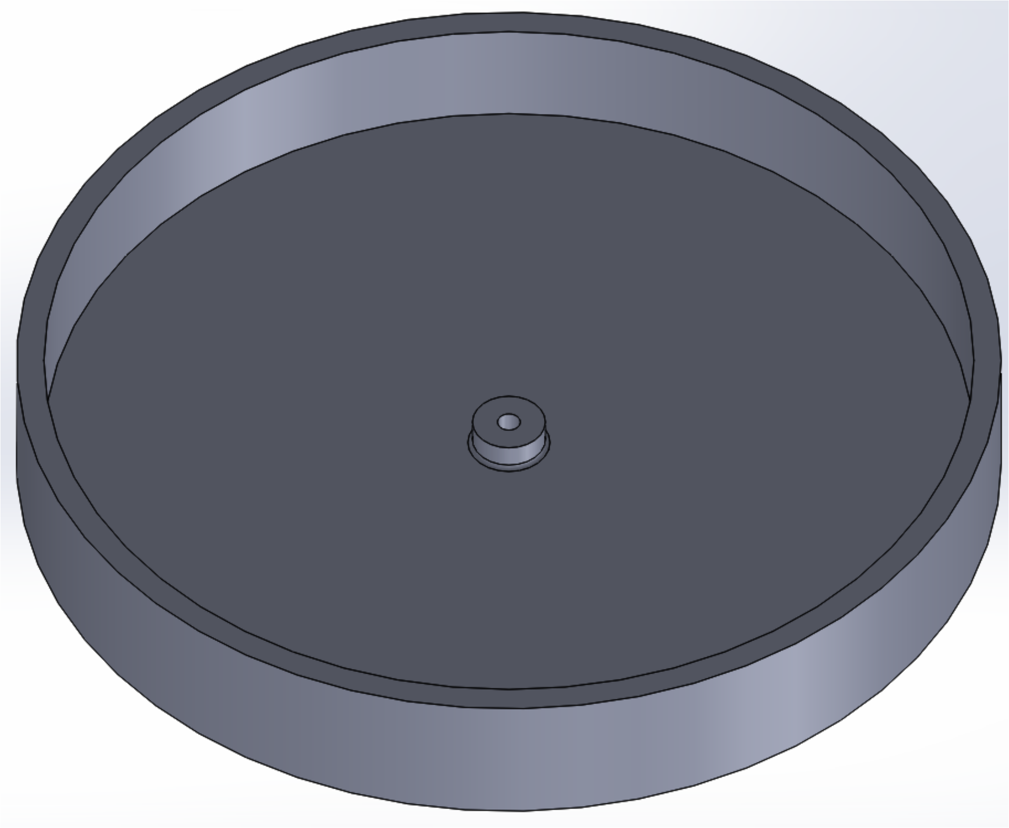


Figure 47 Proposed Top Plate design

To decrease the overall size of the product we would also like to use a small lithium ion battery that could fit neatly inside the coaster. We would also switch from an Arduino to a PCB with a TI BLE chip. This would allow all the components to fit inside the coaster. This would result in a much neater and smaller final product.

4.15 System Overall

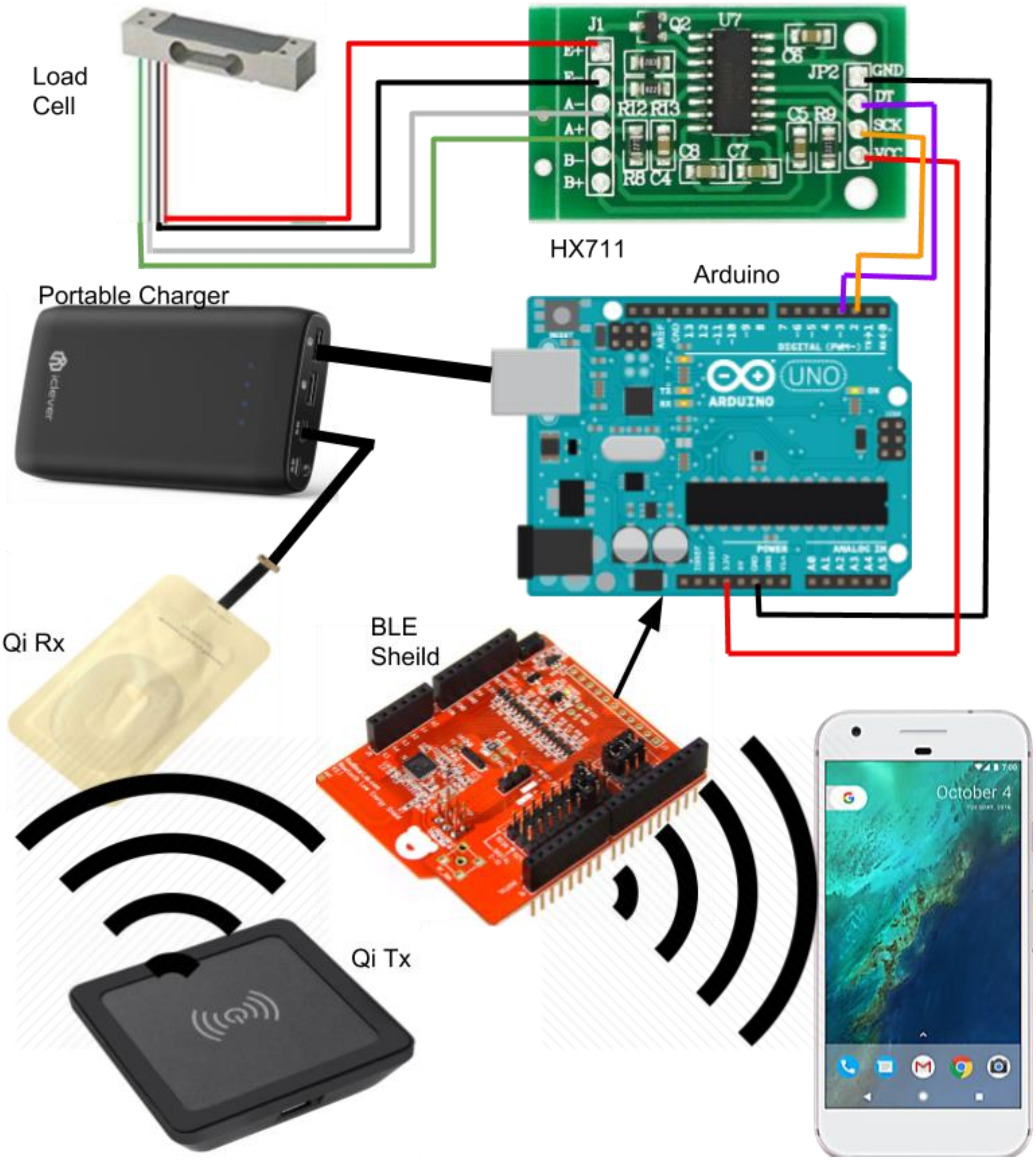


Figure 48 System Overall

5. Conclusions

After three terms of work we have created both a water tracking coaster and an app. The coaster measures weight accurately to +/-3 grams using a 5kg load cell. Because an ounce of water weighs about 30 grams, this is accurate enough to track water consumption. The water consumption data is sent via Bluetooth low energy to an Android app. This app then processes that data and stores it into a database. The database stores information in terms of users and data. Previous daily totals can be selected and viewed, and the data is live-updated. The system is wirelessly powered via a Qi transmitter and receiver powering an iClever power bank that powers the Arduino. This allows the user to use the coaster even while it is charging. Going forward we would like to design a smaller coaster with a custom made smaller load cell, and PCB to replace the electronic components. Despite its shortcomings, our coaster is a step in the right direction to combating chronic dehydration, and helping people become more aware of their water consumption habits.

References

[1] “24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales.” Avia Seminconductor.

[2] Akers, Jeremy D., et al. “Daily Self-Monitoring of Body Weight, Step Count, Fruit/Vegetable Intake, and Water Consumption: A Feasible and Effective Long-Term Weight Loss Maintenance Approach.” *Journal of the Academy of Nutrition and Dietetics*, vol. 112, no. 5, May 2012. *ScienceDirect*, doi:10.1016/j.jand.2012.01.022.

[3] Al-Mutlaq, Sarah, and Alex the Giant. “Load Cell Amplifier HX711 Breakout Hookup Guide.” *Sparkfun*, Sparkfun.

[4] Armstrong, Lawrence E. “Challenges of linking chronic dehydration and fluid consumption to health outcomes.” *Nutrition Reviews*, vol. 70, no. 2, 2 Nov. 2012. *Wiley Online Library*, doi:10.1111/j.1753-4887.2012.00539.x.

[5] Borofsky, Michael S., et al. “Mp90-03 Investigating Smart Water Bottle Technology As A Clinical Tool For Stone Formers.” *The Journal of Urology*, vol. 197, no. 4, 3 Apr. 2017. *ScienceDirect*, doi:10.1016/j.juro.2017.02.2822.

[6] Dotinga, Randy. "'Smart' water bottle may help with stone prevention: bottle's accuracy in measuring consumption also could aid clinical research." *Urology Times*, Feb. 2017, p. 10. *General OneFile*

[7] JIMBO. “Force Sensitive Resistor Hookup Guide.” *Learn at SparkFun Electronics*, Sparkfun.

[8] “FSR Force Sensing Resistor Integration Guide and Evaluation Parts Catalog.” Interlink Electronics.

[9] Gibson, S., et al. “Hydration, water intake and beverage consumption habits among adults.” *Nutrition Bulletin*, vol. 37, no. 3, 15 Aug. 2012, pp. 182–192. *Wiley digital Archives*, doi:10.1111/j.1467-3010.2012.01976.x.\

[10] “H2OPal Smart Water Bottle Hydration Tracker.” *H2OPal (Out of Galaxy, Inc.)*, www.h2opal.com/.

[11] Hidrate Inc. “Hidrate Spark.” *Hidrate Inc*, hidratespark.com/.

[12] HydraCoach, Inc. “HydraCoach.” *HydraCoach, Inc.*, hydracoach.com/.

[13] Lee, Nam Eui, et al. “A Smart Water Bottle for New Seniors: Internet of Things (IoT) and Health Care Services.” *International Journal of Bio-Science and Bio-Technology*, vol. 7, no. 4, 2015, pp. 305–314., doi:10.14257/ijbsbt.2015.7.4.30.

[14] “Measuring Strain with Strain Gages.” *Measuring Strain with Strain Gages - National Instruments*, National Instruments, 25 May 2016.

[15] Pross, Nathalie. “Effects of Dehydration on Brain Functioning: A Life-Span Perspective.” *Annals of Nutrition and Metabolism*, vol. 70, no. 1, 2017, pp. 30–36. *Web of Science*, doi:10.1159/000463060.

[16] “Revolutionary Hydration.” *Moikit*, en.moikit.com/cuptime2.

[17] “The Qi Wireless Power Transfer System Power Class 0 Specification.” Wireless Power Consortium, Apr. 2016.

[18] Thornton, Simon N., and Marie Trabalon. “Chronic dehydration is associated with obstructive sleep apnoea syndrome.” *Clinical Science*, vol. 128, no. 3, ser. 225, 1 Feb. 2015, pp. 225–225. *225*, doi:10.1042/cs20140496.

[19] “What you need to know when buying Drink coasters:” *Canada Coaster*.

Appendix A

Circuits4you.com (Nov 25, 2016) HX711 Load Cell Amplifier Interface with Arduino, Calibration Sketch [Arduino Sketch]. https://circuits4you.com/2016/11/25/hx711-Arduino-load-cell/

SM (July 29, 2015) Smoothing [Arduino Sketch] https://www.Arduino.cc/en/Tutorial/Smoothing

loadcell_callibration.ino

```
/*
 * circuits4you.com
 * 2016 November 25
 * Load Cell HX711 Module Interface with Arduino to measure weight in Kgs
 Arduino
 pin
 2 -> HX711 CLK
 3 -> DOUT
 5V -> VCC
 GND -> GND

 Most any pin on the Arduino Uno will be compatible with DOUT/CLK.
 The HX711 board can be powered from 2.7V to 5V so the Arduino 5V power
should be fine.
*/

#include "HX711.h"  //You must have this library in your Arduino library folder

#define DOUT  3
#define CLK  2

HX711 scale(DOUT, CLK);

//Change this calibration factor as per your load cell once it is found you many
need to vary it in thousands
float calibration_factor = -439000;

const int numReadings = 10;
float readings[numReadings];      // the readings from the analog input
```

```
int readIndex = 0;              // the index of the current reading
float total = 0;                  // the running total
float average = 0;                // the average


//==================================================================
========================================
//                        SETUP
//==================================================================
========================================
void setup() {
  Serial.begin(9600);
  Serial.println("HX711 Calibration");
  Serial.println("Remove all weight from scale");
  Serial.println("After readings begin, place known weight on scale");
  Serial.println("Press a,s,d,f to increase calibration factor by 10,100,1000,10000
respectively");
  Serial.println("Press z,x,c,v to decrease calibration factor by 10,100,1000,10000
respectively");
  Serial.println("Press t for tare");
  scale.set_scale();
  scale.tare(); //Reset the scale to 0

  long zero_factor = scale.read_average(); //Get a baseline reading
  Serial.print("Zero factor: "); //This can be used to remove the need to tare the
scale. Useful in permanent scale projects.
  Serial.println(zero_factor);

   for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = 0;
}
}

//==================================================================
========================================
//                        LOOP
//==================================================================
========================================
void loop() {
  // subtract the last reading:
  total = total - readings[readIndex];
  // read from the sensor:
 // long weight= scale.get_units();
  readings[readIndex] = scale.get_units();
  // add the reading to the total:
  total = total + readings[readIndex];
  //Serial.println (readings[readIndex], 3);
  // advance to the next position in the array:
  readIndex = readIndex + 1;

  // if we're at the end of the array...
  if (readIndex >= numReadings) {
    // ...wrap around to the beginning:
    readIndex = 0;
  }

  // calculate the average:
  average = total / numReadings;
  // send it to the computer as ASCII digits
  //Serial.println(scale.get_units(), 3);
  Serial.println(average, 3);
  delay(1);        // delay in between reads for stability

  scale.set_scale(calibration_factor); //Adjust to this calibration factor

  /*Serial.print("Reading: ");
  Serial.print(scale.get_units(), 3);

  Serial.print(" kg"); //Change this to kg and re-adjust the calibration factor if you
follow SI units like a sane person
  Serial.print(" calibration_factor: ");
  Serial.print(calibration_factor);
  Serial.println();*/

  if(Serial.available())
  {
    char temp = Serial.read();
    if(temp == '+' || temp == 'a')
      calibration_factor += 10;
    else if(temp == '-' || temp == 'z')
      calibration_factor -= 10;
    else if(temp == 's')
      calibration_factor += 100;
    else if(temp == 'x')
      calibration_factor -= 100;
    else if(temp == 'd')
      calibration_factor += 1000;
    else if(temp == 'c')
      calibration_factor -= 1000;
    else if(temp == 'f')
      calibration_factor += 10000;
    else if(temp == 'v')
      calibration_factor -= 10000;
    else if(temp == 't')
      scale.tare();  //Reset the scale to zero
  }
}
//============================================================
===================================
```

Appendix B

Circuits4you.com (Nov 25, 2016) HX711 Load Cell Amplifier Interface with Arduino, Final Code for Weight Measurement, [Arduino Sketch]. https://circuits4you.com/2016/11/25/hx711-Arduino-load-cell/

loadcell_callibrated.ino

```
/*
 * circuits4you.com
 * 2016 November 25
 * Load Cell HX711 Module Interface with Arduino to measure weight in Kgs
 Arduino 
 pin 
 2 -> HX711 CLK
 3 -> DOUT
 5V -> VCC
 GND -> GND

 Most any pin on the Arduino Uno will be compatible with DOUT/CLK.
 The HX711 board can be powered from 2.7V to 5V so the Arduino 5V power
should be fine.
*/

#include "HX711.h"  //You must have this library in your Arduino library folder

#define DOUT  3
#define CLK  2

HX711 scale(DOUT, CLK);

//Change this calibration factor as per your load cell once it is found you many
need to vary it in thousands
float calibration_factor = -439000.00; //-106600 worked for my 40Kg max scale
setup

//=============================================================
====================================
//                    SETUP
```

```
//===============================================================
======================================
void setup() {
  Serial.begin(9600);
  Serial.println("Press T to tare");
  scale.set_scale(-439000.00);  //Calibration Factor obtained from first sketch
  scale.tare();
  //Reset the scale to 0


}

//===============================================================
======================================
//                        LOOP
//===============================================================
======================================
void loop() {
  Serial.print("Weight: ");
  Serial.print(scale.get_units(), 3);  //Up to 3 decimal points
  Serial.println(" kg"); //Change this to kg and re-adjust the calibration factor if you
follow lbs

  if(Serial.available())
  {
    char temp = Serial.read();
    if(temp == 't' || temp == 'T')
      scale.tare();  //Reset the scale to zero
  }
}
//===============================================================
======================================
```

Appendix C

Android application source code

https://github.com/abchan021796/HydrationProject