Final Report for ECE 445, Senior Design, Spring 2018

TA: Nicholas Ratajczyk

May 2018

Project No. 31

Raymond Chang

Christina Choi

Harrison Hsieh

By

Enhanced beverage Coaster

**Abstract**

This report entails the design process and the final product of an enhanced beverage coaster. The goals of the coaster are to communicate through Bluetooth to an iPad system in order to transfer data about whether the consumer’s beverage requires a refill and a notification system to get the server’s attention. The scope of the project is first introduced. Next, the details of the hardware, software, and physical design processes are described. Additionally, we explained our previous and re-evaluated design decisions and testing verifications. Lastly, the report is concluded with what was accomplished and what we would like to work on further in the future.

Contents

[1. Introduction 1](#_Toc513061379)

[1.1 Introduction 1](#_Toc513061380)

[1.2 Objective 1](#_Toc513061381)

[1.3 Block Diagram 1](#_Toc513061382)

[1.3.1 Bluetooth Chip/Microcontroller 2](#_Toc513061383)

[1.3.2 Load Sensor 2](#_Toc513061384)

[1.3.3 Button 2](#_Toc513061385)

[1.3.4 LED 2](#_Toc513061386)

[1.3.5 Data Collection Module 2](#_Toc513061387)

[1.3.6 Design Revisions 2](#_Toc513061388)

[2 Design 2](#_Toc513061389)

[2.1 Circuitry 2](#_Toc513061390)

[2.1.1 Schematic 2](#_Toc513061391)

[2.1.2 PCB 3](#_Toc513061392)

[2.2 Physical Design 4](#_Toc513061393)

[2.2.1 General Dimensions 5](#_Toc513061394)

[2.2.2 LED Visibility 5](#_Toc513061395)

[2.2.3 Water Resistance 5](#_Toc513061396)

[2.2.4 Accessible Battery Door 5](#_Toc513061397)

[2.2.5 Full PCB Enclosure 5](#_Toc513061398)

[2.3 Software 6](#_Toc513061399)

[2.3.1 Coaster Software 6](#_Toc513061400)

[2.3.2 Data Collection Module Software 8](#_Toc513061401)

[3. Design Verification 11](#_Toc513061402)

[3.1 Circuit Verification 11](#_Toc513061403)

[3.1.1 Load Sensor 11](#_Toc513061404)

[3.1.2 Battery 12](#_Toc513061405)

[3.2 Physical Verification 12](#_Toc513061406)

[3.2.1 Water Resistance 12](#_Toc513061407)

[3.2.2 Weight Support 12](#_Toc513061408)

[3.2.3 LED 12](#_Toc513061409)

[3.2.4 Battery Door 13](#_Toc513061410)

[3.3 Software Verification 13](#_Toc513061411)

[3.3.1 Control Module 13](#_Toc513061412)

[3.3.2 Data Collection Module 13](#_Toc513061413)

[4. Costs 13](#_Toc513061414)

[4.1 Parts 13](#_Toc513061415)

[4.2 Labor 13](#_Toc513061416)

[4.3 Total Cost 14](#_Toc513061417)

[5. Conclusion 14](#_Toc513061418)

[5.1 Accomplishments 14](#_Toc513061419)

[5.2 Challenges 14](#_Toc513061420)

[5.2.1 Circuitry 14](#_Toc513061421)

[5.2.2 Physical 14](#_Toc513061422)

[5.2.3 Software 15](#_Toc513061423)

[5.3 Ethical considerations 15](#_Toc513061424)

[5.4 Future work 16](#_Toc513061425)

[5.4.1 Circuitry 16](#_Toc513061426)

[5.4.2 Physical 16](#_Toc513061427)

[5.4.3 Software 16](#_Toc513061428)

[References 17](#_Toc513061429)

[Appendix A Requirement and Verification Table 18](#_Toc513061430)

# 1. Introduction

## Introduction

We would like to facilitate this inefficient call-for-attention by integrating a useful notification system that can bring more practicality without adding more objects to the dining table. We also want the enhanced beverage coaster to be able to detect when a beverage needs a refill and when a consumer needs the server’s attention, and then send that data to an iPad for the server to acknowledge.

## 1.2 Objective

We would like to facilitate this inefficient call-for-attention by integrating a useful notification system that can bring more practicality without adding more objects to the dining table. We also want the enhanced beverage coaster to be able to detect when a beverage needs a refill and when a consumer needs the server’s attention, and then send that data to an iPad for the server to acknowledge.

## 1.3 Block Diagram



Figure 1: Block Diagram of Enhanced Beverage Coaster

### 1.3.1 Bluetooth Chip/Microcontroller

The Bluetooth chip used in this project was on the same chip as the microcontroller. It is one of the core components of this project because it is responsible for the data transfer between the data collection module and coaster.

### 1.3.2 Load Sensor

The load sensor acts as a resistor which allows the coaster to sense the weight of the object, a cup and beverage in this case, placed on top of the sensor. This is the data that allows the data collection module to determine whether or not the customer needs a refill.

### 1.3.3 Button

This button is to be used by the customer in order to call the server. When the button is pressed, an LED will light up indicating that the server has been notified.

### 1.3.4 LED

The LEDs indicates the status of the server call. After the LEDs is lit from a button press, the server can turn this LEDs off by acknowledging the server call.

### 1.3.5 Data Collection Module

This will be where the data from coasters gets sent and processed to present to servers. Servers can also adjust parameters such as cup weight and can also acknowledge server calls from coasters.

### 1.3.6 Design Revisions

Throughout the semester, we revised our design quite significantly after receiving feedback from professors and peers. Below is a list of the major changes that were made to the project:

1. A button was added to enable customers to call servers via the coaster.
2. Beverage consumption tracking was simplified to only determining whether or not a glass needs to be refilled.
3. A UI was made in order to present information and control the coaster.
4. The coaster shape was changed from circular to square to make the size more compact.

# 2 Design

## 2.1 Circuitry

### 2.1.1 Schematic

In designing the circuit schematic, the goal was to keep it as simple as possible. The biggest component was the Bluetooth chip/microcontroller. It has a lot of inputs and outputs that were not needed so all the outputs that were not used were grounded. AND gates were also added before the LED with inputs from the power supply and the microcontroller because enough output voltage needed to be provided to power them. For the red LED, a voltage divider circuit was added to drop the input voltage from 3V to 2.1 V because that is what is needed to drive the red LED. Other design considerations included using a button so that the LED could be toggled on and off.



Figure 2: Circuit Schematic

### 2.1.2 PCB

With the PCB design, it needed to be small enough to be a good coaster size while also big enough to fit all the components. The biggest consideration was creating the PCB with through holes for components that were more unusual like a battery holder and a button. For those components, the solution was to manually measure out the distances between leads and put through holes where the leads are. For the Bluetooth chip/microcontroller, it needed to be placed in the corner of the PCB according to the datasheet. The LED was placed on the opposite side of the battery holder and towards the center of the PCB so that it could be seen through the middle of the coaster.

## 2.2 Physical Design

****

Figure 3: Top 3D Printed Layer of Coaster Design

****

Figure 4: Second 3D Printed Layer of Coaster Design

****

Figure 5: Third 3D Printed Layer of Coaster Design

### 2.2.1 General Dimensions

The coaster dimensions were designed to be 8.5cm by 8.5cm by 3.27cm. This was a reasonable dimension that mimicked the dimensions of most rounded-square coasters. It is able to support commonly seen cup sizes that can hold around 14oz of fluid. The thickness of the coaster is mainly due to the size of the button and battery holder parts that are soldered onto the PCB.

### 2.2.2 LED Visibility

In order to ensure the visibility of the LED, the top layer of the coaster had an opening to expose a part of the PCB as seen in Figure 4. There was also a transparent silicone material that sat on the surface of Figure 4 to ensure that the LED shined through.

### 2.2.3 Water Resistance

The coaster was made to be water resistant using the silicone cover. It was placed between Figure 3 and Figure 4. Figure 3 was a printed border used to cover the raw edges of the silicone cover and make the surface of the coaster look seamless.

### 2.2.4 Accessible Battery Door

The battery holder was purposely designed to be on the opposite side of the button on the PCB so that the coaster can have an accessible battery door. Our coaster design used coin cell batteries that are not rechargeable, so it was important for the design to have access to swap out the battery on the bottom. The enclosure is seen on the bottom right side in Figure 5.

### 2.2.5 Full PCB Enclosure

It was very important for the PCB to be fully enclosed in the coaster because there will always be a cup of fluid on or near the coaster itself. So, the PCB was completely closed off and the parts were to be glued together to ensure that no fluid enters the coaster and interferes with the circuitry.

## 2.3 Software

### 2.3.1 Coaster Software

The main purpose of the software on the coaster is to deal with button presses, the LEDs, and to send and receive data from the data collection module.

#### 2.3.1.1 Bluetooth and Microcontroller

With the advent of Bluetooth Low Energy, also known as Bluetooth Smart, low energy wireless transmission became more easily accessible. A major difference when comparing BLE and regular Bluetooth is that devices do not have to be actively paired in order to communicate with each other. BLE devices advertise their services that other devices can connected to and use.

When programming the chip for this project, we made a GATT profile that advertised a service with three characteristics: the serial number characteristic, read request characteristic, and an LED toggle characteristic. The serial number characteristic allows devices accessing it to read the serial number of the coaster. The read request characteristic allows users to send a request to read the current voltage coming from the load sensor and the current status of the LEDs. Finally, the toggle LED characteristic allows devices to turn the LEDs on the coaster off.

#### 2.3.1.2 Read Requests



Figure 6: Read Request Flowchart

When a read request is received, the microcontroller calls the voltage read function in order to obtain the current voltage over the load sensor. The return value from this call gives the 12 most significant bits of the voltage value in a 2 byte formfactor. Since the chip used did not support multiple handlers for read requests, the LED status also had to be returned in this read request. In order to do this, a bitmask was used. The return value of the voltage read call was bitwise ANDed with 0xfffe to zero the least significant bit. Then, the new value was then bitwise XORed with the LED status variable. The result of these operations was then passed to the return value of the read request.

#### 2.3.1.3 Button Press



Figure 7: Button Press Flowchart

The button serves to let the customer call a server when needed. The functionality of this button was implemented using a recurring timer that was set to one second. Whenever the timer elapsed, the code checks to see if the button is pressed and then adjusts the LED\_on flag to 1 if the button is indeed pressed. This implementation serves to prevent accidental presses due to the fact that the button must be held down for a period of time before is registers the button press.

#### 2.3.1.4 LED Status



Figure 8: LED Status Flowchart

There are two parts to the mechanism that determines if the LEDs should be lit or not. The first part of the mechanism works in conjunction with the button press mechanism. At every elapse of the timer, the code checks whether or not the LED\_on flag is currently set to 1. If it is, then the LEDs will be turned on. The second part of the mechanist deals with interaction from the data module. When a server acknowledges a server call, the data collection module sends a notification toggle to the coaster which raises the notification handler. After the handler is called, the program does a simple check of the toggle flag to see whether or not to turn off the LEDs.

### 2.3.2 Data Collection Module Software

The data collection module serves as the brains of the project and also serves as the way for servers to read the data collected from the coasters. This module was implemented on an iPad.

#### 2.3.2.1 User Interface

****

Figure 9: Screenshot of iPad User Interface Design

Since all the data is shown on the data collection module, it was important that the user interface be easy to read and interact with. It is split into three main sections: the assistance section, parameters section, and statistics section. The assistance section displays when a customer presses the button on the coaster in order to call a server. There is an acknowledge button in this section that lets servers indicate that they have received the server call. In the parameters section, it allows the servers to set the weight of the empty cup in order for the program to determine if the cup is empty or not. The last section which houses the statistics displays to the server the current statuses of the coaster. It shows the serial number, cup weight, and cup status. When the cup needs to be refilled, the text under cup status will turn red and display “Needs refill.”

#### 2.3.2.2 Data Collection Module and Coaster Interaction



Figure 10: Flowchart of Interaction Between Devices

The data collection module and coaster interact with each other through BLE. This is mainly done through the collection module which is set as the master while the coaster is set as the slave. Upon opening the application, a recurring timer is set to 15 seconds. Whenever the timer elapses, the code starts scanning for Bluetooth Low Energy devices. This scanning process has three main steps, first it scans for all devices. Next, it narrows the list down by scanning for the wanted service. Finally, it scans for the wanted characteristics. If all three scans are successful, the iPad establishes a connection with the coaster. When this happens, the iPad sends a read request in order to obtain the voltage data and LED status. If the server has acknowledged a server call, the iPad will then send the LED off toggle notification to the coaster too. After all this, the application disconnects from the coaster and waits for the next timer elapsed.

#### 2.3.2.3 Data Processing

In order to determine whether or not to indicate that a cup needs to be refilled, the collection module must process the voltage read from the coaster. After some testing, we determined that a threshold of about 1.25 times the cup weight is a good time to tell the server a cup needs to be refilled.

To determine the force on the load sensor we use these three equations:

$$\frac{V\_{LS}}{3V}= \frac{R\_{LS}}{1KΩ+R\_{LS}}$$

$$R\_{LS}=1.444707\*F\_{LS}$$

$$1g= .035274 oz.$$

These equations simplify to this equation which gives the force in ounces:

$$F\_{LS}=\frac{.035274 oz\*V\_{LS}(1KΩ+R\_{LS})}{3V\*1.444707}$$

Finally, the calculated value is compared with the threshold weight to determine if the cup needs to be refilled or not.

# 3. Design Verification

## 3.1 Circuit Verification

### 3.1.1 Load Sensor

We tested our load sensor by putting it in series with a test resistor and measuring the output voltage across the load sensor. Using three different valued resistors, we wanted to find the resistor that would give us the desired force sensitivity range. We tested each resistor with different loads of weight to measure the output voltage. When we tested with a 10 kΩ resistor, the values for VOUT would not reach a steady value so we did not want to use this resistor. Next, we tested with a 1 kΩ resistor and got relatively reliable numbers. The last resistor we tested with was a 470 Ω resistor. The values for VOUT from the 470 Ω resistor gave us numbers that, while steady, were not in the range of sensitivity that we wanted. The results of testing can be seen in Figure 11.



Figure 11: VOUT vs Force Testing Results

We decided to use the 1 kΩ resistor. Using the provided voltage divider equation, we calculated the resistance of the load sensor ($R\_{FSR}$) for every force measurement:

$V\_{OUT} = \frac{R\_{M}V^{+}}{(R\_{M} + R\_{FSR})}$ 🡪 $R\_{FSR}V\_{OUT} + R\_{M}V\_{OUT} = R\_{M}V^{+}$ 🡪 $R\_{FSR} = \frac{R\_{M}V^{+} - R\_{M}V\_{OUT}}{V\_{OUT}}$

The results of these calculations can be seen in Figure 12.



Figure 12: RFSR vs Force Calculation Results

### 3.1.2 Battery

We tested the output voltage of the battery by using a voltmeter and measuring across the leads. It provided 3.2 Volts which was within our accepted range.

## 3.2 Physical Verification

### 3.2.1 Water Resistance

The silicone layer that was placed between Figure 2 and Figure 3 was waterproof and the material from the 3D printer is as well. It covered the entire top surface, including the exposed load sensor, and it was able to be wiped down. However, it was not properly sealed on the edges, so if liquid was present on the surface there would be a chance that it would leak into the coaster.

### 3.2.2 Weight Support

The coaster was 3D printed so the material is very stiff and durable. Even with a 28oz glass cup, the coaster was able to remain intact.

### 3.2.3 LED

As seen in Figure B, there is a long hole that is near the center. That section is purposely made so that the LED would be able to shine through the coaster. However, the LED was not able to be placed onto the PCB due to soldering issues so we were not able to conduct this test.

### 3.2.4 Battery Door

The battery door that was designed was supposed to have drilled holes on either side so that the door can be screwed in. The enclosure was successfully executed but the door lacked the functionality for screws.

## 3.3 Software Verification

### 3.3.1 Control Module

We could verify that all of our software requirements worked because the final product worked. The control module was able to establish a connection with the data collection module to send voltage and LED status data over. Also, since the coaster was able to read the load sensor, the button press, and output a signal to the LED, it is established that it can support two inputs and one output.

### 3.3.2 Data Collection Module

All the functionality was working since the overall project worked. The iPad was able to establish connection and receive the voltage data and LED status data. Also, the interface organized the data in a manner that was processed and easy to read for the user.

# 4. Costs

## 4.1 Parts

Table 1: Parts Cost

|  |
| --- |
|  |
| **Part** | **Manufacturer** | **Quantity** | **Retail Cost ($)** | **Total Cost ($)** |
| Bluetooth/Microcontroller chip | Silicon Labs | 2 | $11.26 | $22.52 |
| Quad dual input AND gate | Nexperia USA Inc. | 2 | $0.39 | $0.78 |
| Coin Cell Battery | Panasonic | 2 | $1.14 | $2.28 |
| Battery Holder | Memory Protection Devices | 2 | $0.82 | $1.64 |
| Button | E-Switch | 2 | $0.53 | $1.06 |
| **Total** |  |  |  | **$28.28** |

## 4.2 Labor

Table 2: Labor Cost

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Hourly Rate** | **Hours** | **Total = Hourly Rate x 2.5 x Hours** |
| Raymond Chang | $30 | 100 | $7,500.00 |
| Christina Choi | $30 | 100 | $7,500.00 |
| Harrison Hsieh | $30 | 100 | $7,500.00 |
| **Total** | **$90** | **300** | **$22,500.00** |

## 4.3 Total Cost

After adding up both the costs for the parts and labor, the total amount come up to $22,528.28. Most of the associated cost comes from labor. This makes sense as we were trying to make the coaster as inexpensive as possible, since most times restaurants buy coasters, they will be more than likely to buy a decent amount at once.

# 5. Conclusion

## 5.1 Accomplishments

The 3D printed prototype of the coaster provided a great visual aid to the project. We wanted to envision this as a product and the model was able to successfully show that our PCB and part choices were, for the most part, practical and reasonable. The PCB was able to sit comfortably inside of the coaster and it was unexposed except for the portion that of the coaster that allowed for the LED to shine through. The transparency of the top also acted as a water-resistant layer to our project. The button was placed on the very corner of the coaster so that there was enough surface area for the drink to sit on the top. The battery holder was fully accessible and it was able to be closed entirely with an additional door piece.

The circuitry and software of the coaster provided the functionality we set out to achieve. The coaster was able to read the force of the load sensor, button press, and LED status and send it to the data collection module successfully. On the data collection module, it was able to establish a connection with the coaster in order to receive data and process it. Then it displayed the processed data in a manner that was easy to understand and interact with.

## 5.2 Challenges

### 5.2.1 Circuitry

The biggest challenge for the circuitry is the PCB. The attempts to solder on some components failed because the PCB through holes were too small for the leads of the button and the battery holder. The only way to solder those components on were to use a large amount of solder and stick it on without actually fitting the leads into the PCB.

### 5.2.2 Physical

The physical aspect of the project was considerably time-consuming. In general, 3D printing takes several hours to complete a small part and tolerance has to be accounted for in the design otherwise it will not fit the way it was intended. The design had be altered numerous times because of unexpected changes to the circuitry. Many of the larger parts in the design, such as the battery holder and button, had leads that were not the conventional shape and size. As a result, the parts were unable to be flush to the PCB which was an unforeseen circumstance. Additionally, to enforce the waterproofing aspect, we decided that a silicone material to cover the top portion of the coaster would be sufficient. The silicone is flexible so it would make sure that pressure from the beverage would definitely be pressed onto the load sensor and its transparency would allow light from the LED to shine through. However, we did not know that most commonly-used adhesives do not stick to silicone and it would require a different adhesive substance to work with the material.

### 5.2.3 Software

There were a few unique challenges that popped up while designing the software. The first bug that took a lot of time to solve was that new characteristics or services written to the Bluetooth chip would not show up on the iPad no matter how they were configured. This was due to the fact that iOS caches GATT profiles, so everytime a new service was added, a reboot of the iPad was needed. Additionally, voltage reading is not very reliable, since the reading gives us an integer with the max set to the input voltage on the chip. Finally, the scripting language used to write the code for the Bluetooth chip and microcontroller was a proprietary scripting language made by BlueGiga. It was based on visual basic which is not the most robust and easy to use to language.

## 5.3 Ethical considerations

The majority of the time spent on the project was in the Senior Design lab. There were many rules and safety guidelines that needed to be strictly followed in these environments. In terms of lab equipment, the most concerning was arguably soldering. We soldered onto our design in the designated workstations that included the soldering iron and the elephant hood [2]. If the iron was not properly turned off or if the hood was incorrectly positioned over the fumes, it would immediately pose a safety hazard to us and other people in the lab. We ensured that those safety hazards were not present. Since every design had electronic hardware components, there was the potential of electrical shock, electrocution, burns, and fires. To avoid these dangerous situations, there were printed safety procedures that are posted in the laboratory and general guidelines online that we followed to ensure that we stayed safe. One of the most important rules we abided by is the one-hand rule: Only one hand should be on an electrical circuit to prevent current from entering the body [2]. In addition, it was against the rules to work in the lab alone[2]. That means it was important to ensure that there was at least one other individual in the lab with us in case either person finds their self in a dangerous situation. So, we stayed with at least one other partner at all times in the lab. These safety rules adhere to #9 IEEE Code of Ethics [1]. When we worked on our project, whether in the design lab or in another facility, we always prioritized safety for the sake of ourselves and others.

The most dangerous component of our device was the lithium coin cell battery. Although it is a small component, it was imperative to be aware of all the possible safety hazards and how to practice safe testing with the battery. The batteries that were not in use were properly covered by insulating material, completely encased in a lithium battery bag, and stowed away in the yellow storage locker in the design lab [2]. The circuit design for our project did not lead to any short circuits in any case of failure. Generally, lithium batteries are very flammable and misuse of the component can cause a chemical fire. Because lithium batteries are especially flammable, we made sure that we knew where the fire extinguishers were located in the lab. If the battery in use became swollen, felt hot, or emitted unusual noises, there was a series of procedures that we would have strictly followed. First, we would have disconnected the battery from the circuit and immediately placed it in a battery bag far away from the other flammable objects. Then, we would have reported the situation to our TA and contact a power-centric TA or the Lab Coordinator so that the battery could be disposed as soon as possible. If the lithium battery exploded, 911 would have been called and we would have immediately evacuated the area. If it ignited, 911 would have been called and the fire should be extinguished with an a fire extinguisher. In any of these cases, our TA and the lab supervisor would have been notified of the situation immediately [3].

With regards to ethics, we strived to design what we can do ourselves and made detailed references to the designs or implementation strategies that we utilized. We made sure that all our designs are either made by us or properly cited in this final project paper. This complies with #1 of the IEEE Code of Ethics because we wanted to pursue an “ethical design” and “sustainable development practices” [1]. We would also fully disclose any possible hazard to the user in order to promote and practice good safety protocol. Since this is a team project within a large class, conflicts of interest inevitably appeared. The best way to we approached such situations was to follow #2 of the IEEE code of ethics and fully disclosed the issues with the affected parties [1]. While we were in the process of creating the design, we received criticism and advice from fellow colleagues, professors, teaching assistants. Criticism was important for us to correct all unforeseen mistakes and improve upon our idea. So, we readily accepted all criticism which led to a change in the scope of our design and a successful outcome. Lastly, it was important to know that even though this was an independent team project, we were able to support the other teams in “their professional development” as well [1]. This included sharing lab equipment, lending them an extra hand if needed, reminding them of safety rules, and encouraging them to further push themselves.

## 5.4 Future work

### 5.4.1 Circuitry

The PCB needs to be redesigned so that the button and battery holder leads can be inserted into the through holes and soldered on. Another step that can be taken is to decrease the PCB footprint to make the physical design easier and take up less space in the coaster.

### 5.4.2 Physical

The physical design can be improved in a couple of different areas. The most noticeable would be the thickness. The length and width dimensions, 8.5 cm by 8.5 cm, are very reasonable but the depth, 3.27 cm, makes the coaster seem too cumbersome. With better decision of parts, and having a rechargeable battery, the thickness could definitely be decreased. We would also like to pick a flexible material, that is not silicone, but is water-resistant and transparent to use as the surface of our coaster.  As mentioned earlier, silicone is a material that will not work with conventional adhesives.

### 5.4.3 Software

There are several aspects that could definitely be improved on the software side. The code for the Bluetooth chip and microcontroller could be cleaned up since some functionality was used in ways that it was not designed for, such as the LED notification toggle. Also, having a recurring timer every one second is a big power draw which definitely negatively affects battery life, perhaps a physical register could be used to hold the button press.

With regards to the data collection module, ideally servers would be able to select different cups to use that already had their weights entered. Also, currently, the application only supports connecting to one coaster, we would like to implement coaster profiles that store coaster data which would allow the application to cycle through multiple coasters. This should not be a huge challenge since we designed already designed the code modularly (It pings the coaster every set time interval, so it can ping other coasters in between). Additionally, since many restaurants have multiple server stations, we want to implement the ability for all the coasters to connect to multiple iPads. Again, this functionality should not be impossible since we designed our device interaction modularly.

# References

[1] "IEEE IEEE Code of Ethics." IEEE - IEEE Code of Ethics. Accessed February 08, 2018.

<https://www.ieee.org/about/corporate/governance/p7-8.html>.

[2] Smith, Casey. "Senior Design Lab Use and Safety." Lecture, January 30, 2018.

[3] Spring 2016 Course Staff. Safe Practice for Lead Acid and Lithium Batteries. PDF. April 13, 2016.

# Appendix A Requirement and Verification Table

Table 3: Requirements and Verification

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Verification** | **Verification Status** |
| *Physical Design: Water Resistance*1. Able to withstand spills and able to be wiped down
2. Encloses circuit completely
 | 1, 2.   Verification process:1. Spill water on the top portion of the coaster before circuit is enclosed and let the water sit on the coaster for 2 minutes
2. Use paper towel to dry off the top
3. Check underside and sides to see if water leaked through
4. If interior looks dry, PCB is placed and sealed within coaster with multi-adhesive waterproof glue
5. Repeat step a-c
6. Check if LED is lighting up through the acrylic piece
 | 1. N
2. Y
 |
| *Physical Design: Support Weight*1. Able to support weight of full cup
 | 1. Verification process:
2. Place a standard empty glass cup that can hold at least 14 oz (396.89 g) of fluid
3. Add water in approximately 50 g increments to the glass cup until full and check if there is any excessive stress to the coaster
4. Full cup should be left on coaster for at least 5 to 10 minutes in order to see any possible damages from the pressure to the coaster
 | 1. Y
 |
| *Physical Design: LED*1. LED is able to shine through
 | 1. Verification process:
2. User is able to see LED shine through acrylic when around 1 ft away from the coaster
 | 1. N
 |
| *Physical Design: Accessible Battery Door*1. Battery door should be able to full enclose the battery
2. Battery door should be able to open and close with a screwdriver
 |       1, 2. Verification process:1. User should be able to screw and unscrew the battery door with a screwdriver
2. User should be able to remove and replace the battery without other additional tools
 | 1. Y
2. N
 |
| *Control Module: Microcontroller I/O*1. Support two inputs and one output
 | 1. Verification process:
2. Attach 3V power supply to button.
3. Connect button output to input pin 1.
4. Attach LED to output pin.
5. Attach 1k resistor from LED to ground.
6. Load “I/O Test” program onto microcontroller.
7. Make sure microcontroller is powered on.
8. Press button and ensure LED lights up.
9. Repeat process for input pin 2.
 | 1. Y
 |
| *Control Module: Bluetooth*1. Successfully pair with data collection module.
2. Transfer data to data collection module.
 | 1. Verification Process:
2. Load “Test Bluetooth” program onto microcontroller.
3. Turn on microcontroller.
4. Go to bluetooth settings on data collection module.
5. Select scan for devices.
6. Pair with device “Test Bluetooth.”
7. Ensure “Test Bluetooth” is on the paired devices list.
8. Verification Process:
9. Load “Test Bluetooth” program onto microcontroller.
10. Turn on Microcontroller
11. Pair with data collection module
12. Open “Testing App” on data collection module.
13. Select “Test microcontroller data sending”
14. Ensure that “I came from the microcontroller” appears on screen.
 | 1. Y
2. Y
 |
| *Control Module: Data Processing*1. Reads data from sensors and stores it for sending later.
 | 1. Verification Process:
2. Load “Test data” program onto the microcontroller.
3. Attach 3V power supply to button.
4. Connect button to input pin.
5. Connect output pin to LED.
6. Connect 1K resistor to LED and ground.
7. Turn microcontroller on.
8. Hold button for 5 seconds.
9. Ensure LED turns off.
10. Reset microcontroller.
11. Don’t hold button down for 5 seconds.
12. Ensure LED stays on.
 | 1. Y
 |
| *Data Collection Module: Bluetooth*1. Pair with control module
2. Send data to control module
 | 1. Verification Process:
2. Load “Test Bluetooth” program onto microcontroller.
3. Turn on microcontroller.
4. Go to bluetooth settings on data collection module.
5. Select scan for devices.
6. Pair with device “Test Bluetooth.”
7. Ensure “Test Bluetooth” is on the paired devices list.
8. Verification Process:
9. Load “Test data module Bluetooth” program into microcontroller.
10. Turn microcontroller on
11. Go to “Testing App” on data module.
12. Click “Toggle LED” button
13. Ensure LED connected to microcontroller turns on and off.
 | 1. Y
2. Y
 |
| *Data Collection Module: Data*1. Process data sent from coaster (6 pts)
2. Allow users to see data collected (2 pts)
3. Allow users to adjust parameters (4pts)
 | 1. Verification Process
2. Go to “Beverage Coaster” on data collection module
3. Select “Data Processing”
4. Input random parameters
5. Hit submit
6. Go back to main page
7. Ensure stats are correct
8. Verification Process
9. Go to “Beverage Coaster” on data collection module
10. Select “Data Processing”
11. Input random parameters
12. Hit submit
13. Go back to main page
14. Ensure statistics are displayed under “Statistics”
15. Verification Process
16. Go to “Beverage Coaster” on the data collection module
17. Select coaster by tapping select coaster on the left side of the screen
18. Adjust parameters by filling in or changing values
19. Select “Data Processing”
20. Input random parameters
21. Hit submit
22. Go back to main page
23. Ensure statistics correspond to the parameters adjusted for coaster
 | 1. Y
2. Y
3. Y
 |
| *Circuit: Load Sensor*1. Load sensor can measure at least 28.6 oz. (about 810g)
2. Accuracy resolution of at least 3.5 oz. (about 100g)
 | 1. Verification Process
2. Connect sensor to a 5V power supply
3. Add 1K Ohm resistor in series with load sensor
4. Use a multimeter to measure output voltage across load sensor
5. Measure voltage with no weight placed
6. Measure output voltages with increasing weights placed on sensor (increment by 3.5 oz./100g) going all the way to 28.6 oz/810g
7. Plot the resulting voltages and find the line of best fit to determine voltage-force relationship
8. Verification Process
9. Measure output voltages with increasing increments of weight (3.5 oz./100g)
10. Make sure voltages are changing relatively linearly
 | 1. Y
2. Y
 |
| *Circuit: Battery*1. Battery Outputs 3V (+/- 0.3 V) DC
 | 1. Verification Process
2. Use a multimeter to measure the voltage across the battery terminals and determine output voltage
 | 1. Y
 |
| *Circuit: LED*1. Accepts a current range of 1 to 10 mA (1 pt)
2. Accepts a voltage range of 2.0 V to 3.2 V (1 pt)
 | 1. Verification Process
2. Use a multimeter and measure across the LED to determine current
3. Verification Process
4. Use a multimeter and measure across the LED to determine current
 | 1. Y
2. Y
 |