Beverage Coaster with Sensing Capabilities

By

Shivam Patel

Shray Chevli

Suraj Sinha

Final Report for ECE 445, Senior Design, Spring 2018

TA: Dongwei Shi

02, May 2018

Project No. 61

Abstract

The purpose of this project is to design and build a platform that allows businesses and entrepreneurs to analyze and understand patterns at which individuals consume beverages. This platform collects data via a set of coasters with mounted pressure sensors and streams data in real-time about how much liquid is left in their cup. These coasters all communicate via Bluetooth Low Energy (BLE) to a central hub that aggregates this information.

Table of Contents

1. Introduction

1.1 Objective

IoT and smart connected devices have been successfully infiltrating the food & consumer services market. These devices can provide restaurants the ability to monitor chefs, food preparation time and identify & limit inefficiencies [1]. Connected devices in restaurants have a slight overhead cost of getting started but the value that they provide to both the customer and management far exceeds its initial investment [2]. Using these smart devices also provides insights to management that could not have been benchmarked before. Using specific sensors and data collection mechanism, software is better able to track and organize data so that trends and restaurants conclusions can be made to better serve people. These trends could have to do with saving costs, time or being better prepared during peak hours [3].

It is our goal to tap into this smart sensing IOT market and produce something worth using in the restaurant space. Restaurants compete in a very competitive food services industry where innovation is needed to acquire and retain customers.

Applications for our idea can be seen across the restaurant and beverage industries. The Hub in Champaign offers unlimited Mimosas for brunch, but how do they price that deal? Our coaster is able to make an impact by producing tangible data that suggests how much an average participant drinks. And this doesn't stop here, if we look at the largest chain restaurants, which are Panera Bread and Olive Garden, we see that there is a ton of research done by these companies or parent companies into what the optimal size for a cup is so that they optimize costs. Certain sizes can lead customers to be less likely to refill their cups. Our product allows smaller chains to be able to do the same analysis bring cost cutting optimization to practice.

By producing a beverage coaster with sensing capabilities, we are entering the restaurant IOT market space that can be transformed through the ability to collect useful beverage consumption data and enact better service protocols from the beverage coaster data.

Our prototype design as well as our demo design for the coaster is displayed in Appendix B

1.2 Background

IOT and connective devices have been a hot topic for a while in the restaurant business. We see this from companies like TGIF and Olive Garden placing idea like devices to allow customers to order, pay and entertain themselves. Move over to the bar and club industry and we see even anther application, this time with a coaster too. This Bluetooth enabled coaster connects to your phone and notifies you if someone tampers with your drink when you enable the feature. This works by putting the coaster on top of the drink and motion sensors [4]. This device is available for purchase at $39.99, showing that there is a viable market for these types of enhanced coasters.

We felt that the solution of putting the coaster on top of the drink is not essential and felt that it can be more useful if the drink always sat on top of the coaster. Our goal is to produce a more affordable coaster that instead uses it weight-sensing capabilities to send accurate liquid information to a central hub that management can use to service tables in a priority queue style.

With all of this information about what is shaping the industry, we are excited to be able to produce something that can make meaningful impact around our desired goals.

2 Design

Our coaster requires five main modules: a control module, a communication unit, a sensor unit, a power supply and a central hub. The power supply will be a small coin cell battery that can fit inside the form factor of the coaster, power the other three modules, and provide 5V. The control unit will handle collection of data from the load sensor as well as storage of the average weight in the storage. The communication module will collect the data and transmit it through Bluetooth from the control unit to a central hub. The central hub will collect data from all coasters in the vicinity that are on and create a platform on which users can perform analytics. We originally envisioned our coaster as a low cost IOT device that had the capabilities to track the weight of a 1-pound cup and the liquid inside of it, which should be around 1 to 2 pounds as well. Figures 1 and 2, as shown below, are the block diagrams for the coaster and hub respectively.

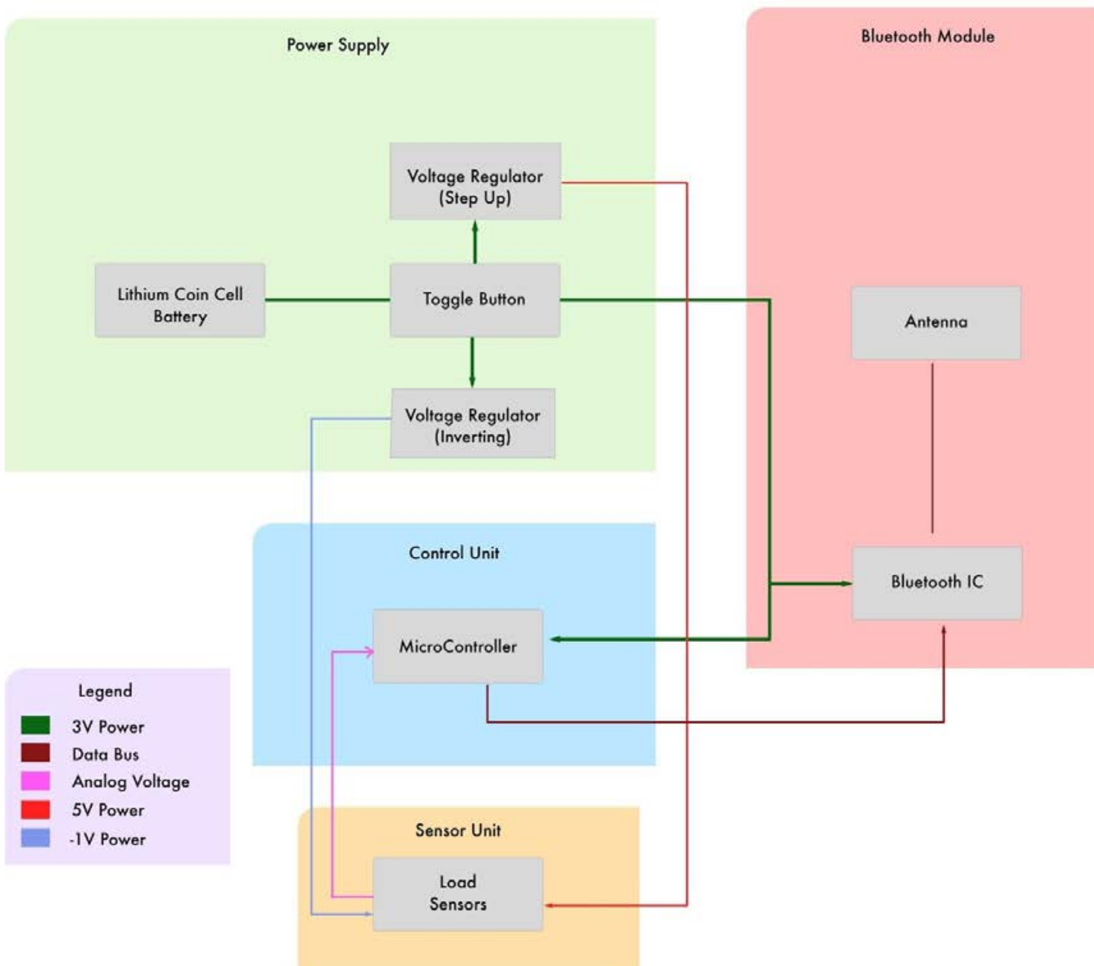

Figure 1. Block Diagram of the coaster

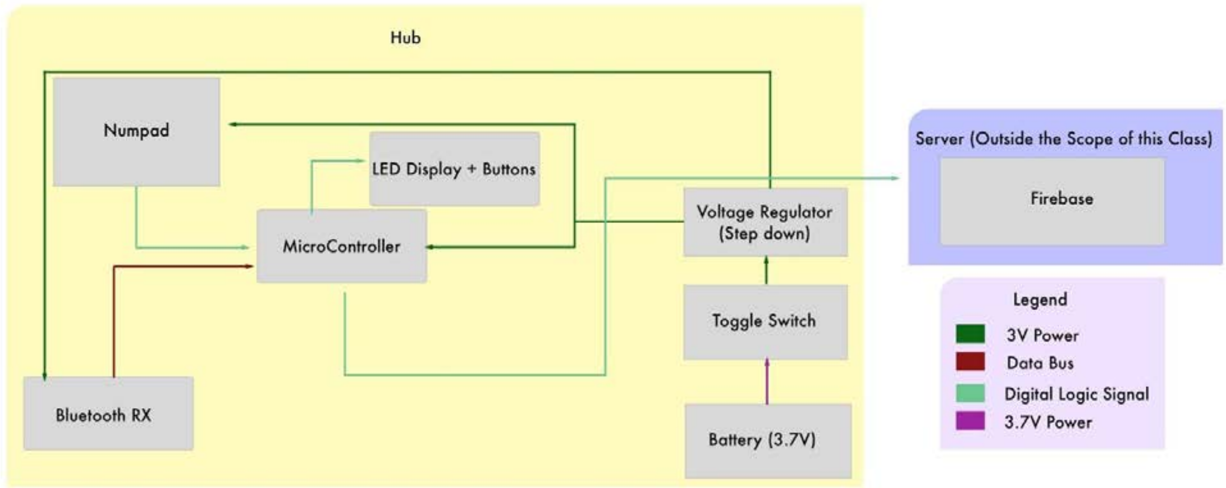

Figure 2. Block Diagram of the hub

2.1 Control Module

For the control module, we chose to use the nRF51822 package because it had the ability as a microcontroller to also double as a Bluetooth IC. The microcontroller read the analog voltage output of the sensor, converted that to weight and added it to the current rolling average. Once a minute has passed, it prepared the current average weight reading to send to the hub via Bluetooth. The layout for the microcontroller was obtained from the user manual [5]

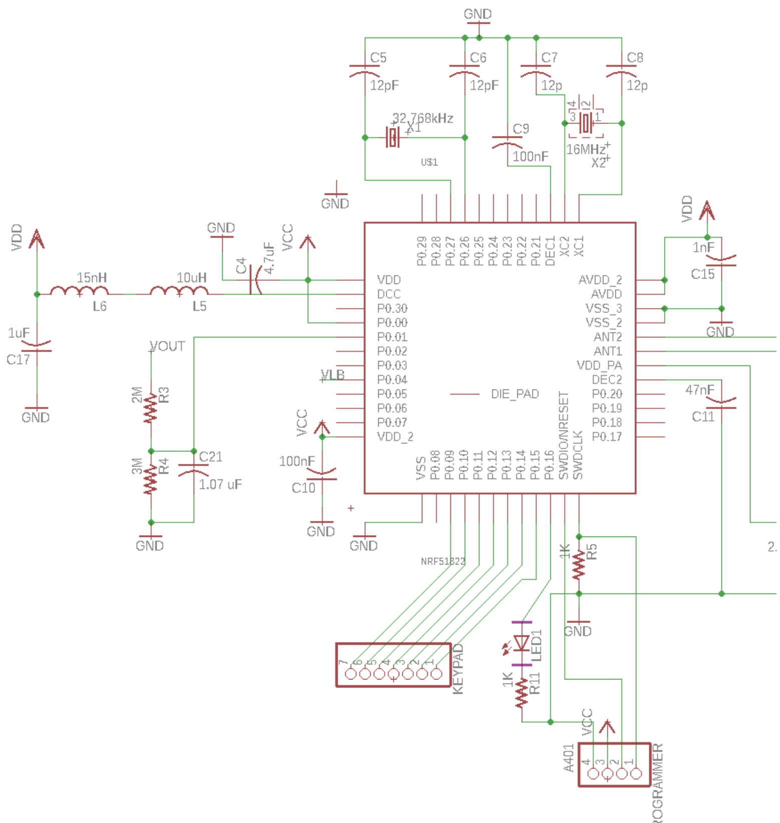
Figure 3. Microcontroller Circuit

2.2 Sensor Module

Our design utilized a flat membrane load sensor, which converted the pressure applied to the sensor into an analog voltage. We placed the load sensor on the top layer of the coaster along with its load disc. We used the load disc so that the weight of the drink, which would have a base larger than the sensor itself, could have all the weight focused on the sensor.

For our load sensor, we chose to use a Flexiforce A401 Sensor due to it sensitivity and range. The sensor, at its best, could read a range of 0 to 7000 lbs. but with the modifications of the supporting circuitry, we could make the sensor more sensitive so that it could read smaller ranges and allow for a greater yield of different values to the weight.

The circuit schematic as shown in Figure 4 shows the options we took to control the range of weight measured. The first part involved biasing one of the leads of the sensor with -1 Volts and the other part involved adjusting the potentiometer in the circuit. By controlling these, we could limit the max weight reading of the coaster to 5 to 10 lbs. [6].

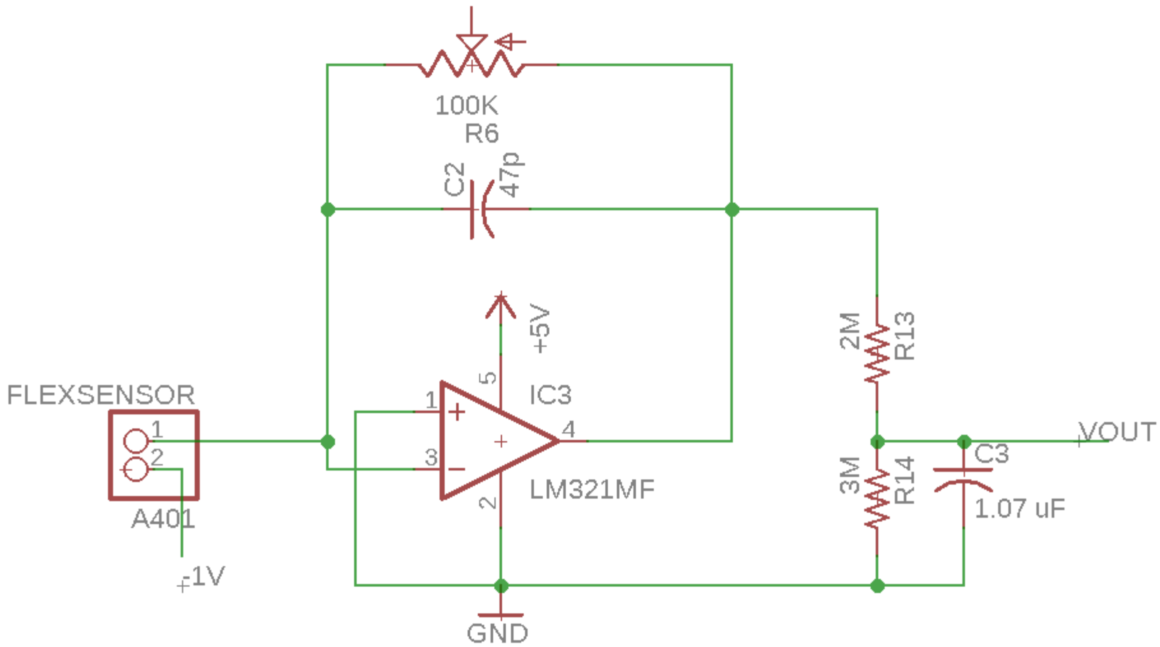

Figure 4. Load Sensor Circuit

The final part of this circuit was to control the ADC voltage reading because the load sensor circuit can output up to 5 Volts but the microcontroller could only read up to 3 Volts. Due to this, we chose to use a simple resistor divider circuit to convert the 5 Volt max down to 3 Volts and we put a capacitor in parallel to the resistor we were reading as a filter that would charge up to the voltage reading for when the ADC is ready to sample the voltage. This value of the capacitor was determined mathematically from the following equations [7] and the user manual of the microcontroller. From the manual, I found that the input resistance for the IC GPIO pin was 129.7 kΩ and the sampling time was 68 μs. On top of that, the 10-bit ADC has a .6 V reference and resolution where 1 bit 1.17 mV [5].

$$V_{Sampling} = V_{sensor} * \frac{R_{14}||R_{pin}}{R_{14}||R_{pin}+R_{13}} + V_{ref} * \frac{R_{13}||R_{14}}{R_{13}||R_{14}+R_{pin}} \tag{2.2.1}$$

$$V_{NotSampling} = V_{Sensor} * \frac{R_{13}}{R_3+R_4} \tag{2.2.2}$$

$$V_{Discharge} = V_{Capacitor} * e^{\left(-\frac{t_{samp}}{(R_{13}||R_{14}||R_{pin})C}\right)} \tag{2.2.3}$$

Using the equations above, I found the $V_{Sampling}$ to be 0.8341 Volts and the $V_{NotSampling}$ to be 3 Volts and the difference to be 2.1659 Volts, which is the max voltage of the capacitor, $V_{Capacitor}$. The discharge voltage of the capacitor is the difference between the capacitor voltage and the 1-bit resolution, $V_{Discharge}$, which was 2.16472 V. After plugging everything into equation 2.1.13, I solved for the capacitance, which was 1.07 µF.

2.3 Bluetooth Module

Our Bluetooth Module originally consisted of a special IC that would take in data from the microcontroller and then send it to the antenna but because of combining our Bluetooth IC and MCU into one component, we were able to save space on our coaster and reduce delay in connections.

The other component to our Bluetooth module was our antenna. When it came to our antenna, we discussed our options, which was to design a meander trace antenna etched onto the PCB or a chip antenna, which may not have the best range but guarantees to work under our conditions. After discussing the options, we felt it would be safer to use a chip antenna because there would be less variability. With a trace antenna, we were not confident in our abilities to design a trace antenna and we were unsure how they would turn out when printed. To ensure that the power delivered to the antenna was the max available, we added a balun as shown in Figure 5 to the circuit to match the chip antenna we used to the ANT output pins on our microcontroller.

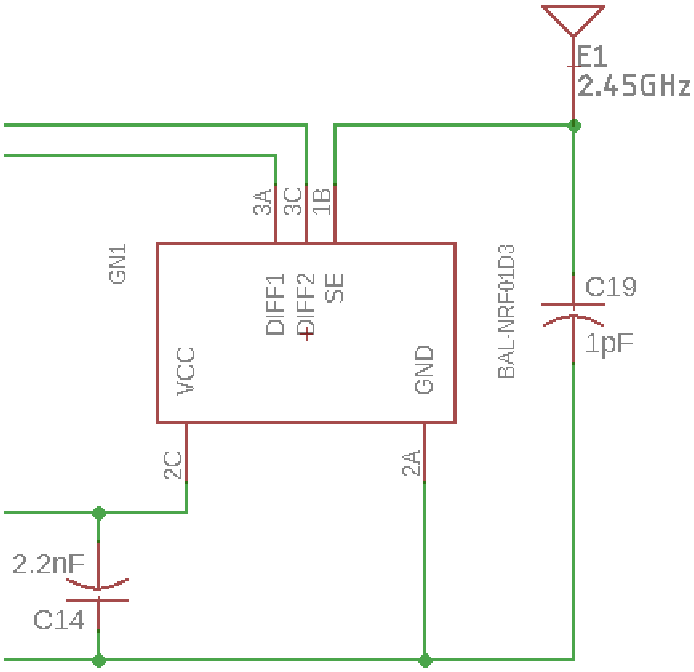

Figure 5. Bluetooth Circuit

2.4 Power Module

Due to our size constraints of the coaster, we decided to use a coin cell battery with 240 mAh for our design. The battery should be able to provide 3 Volts, which is enough to power the microcontroller.

We used a boost converting circuit to convert the 3 Volts provided by the coin cell to 5 Volts to power the op amp in the load sensing circuit. By using 5 Volts here, we can set the voltage limit for the output to 5 Volts and still use the full range from our specified setup. Our 5 Volt boost converter also had an internal comparator that could be used to check for battery detection by comparing the voltage at the pin to its internal 1.25 Voltage Source. The wire labeled VLB in Figure 6 is the output of the comparator and R7, R8, R9, and R10 are part of the circuit. The values for R8, R9, and R10 were all provided in the datasheet and they provided an equation as shown in equation 2.4.1 to determine what R7 based on what voltage you want the circuit to activate [8]. We chose our low battery voltage to 2.75 Volts and R7 was then determined to 41.2 kΩ

$$R7 = \frac{V_{trip} - 1.25}{35.1\mu A} \tag{2.4.1}$$

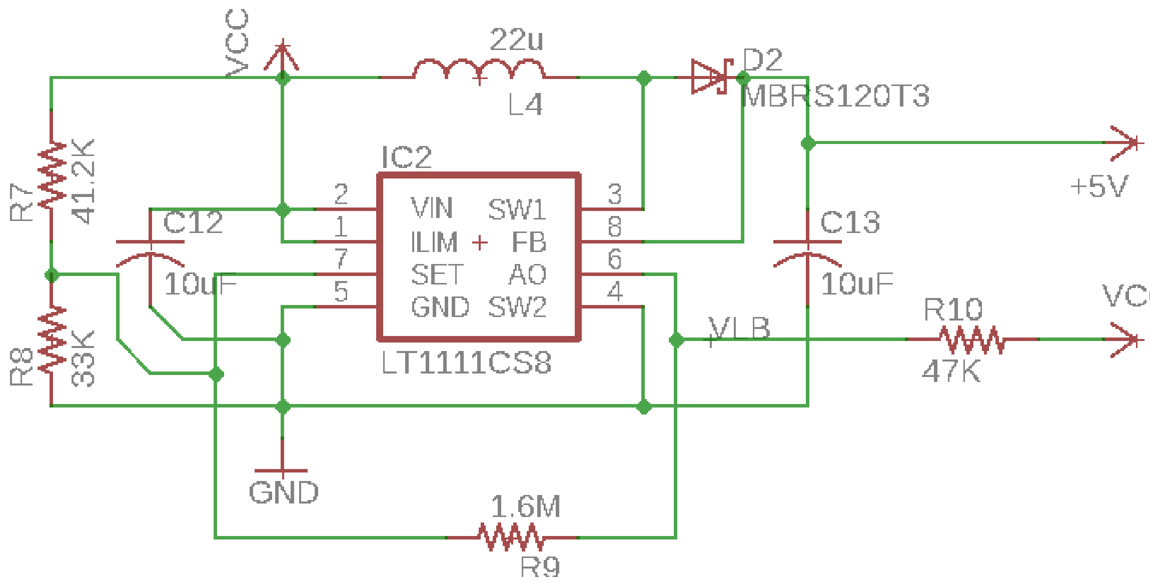

Figure 6. Boost Converting Circuit and Low Battery Detection

We also used an inverting op amp to convert the 3 Volts from the coin cell to -1 Volts to further control the weight range we are measuring. The resistances we used to get the voltage needed was based on the standard inverting op amp configuration that follows equation 2.4.2 and R2 was determined to be 10 kΩ and R1 was determined to 30 kΩ

$$Vout = -Vin * \frac{R_2}{R_1} \quad (2.4.3)$$

2.5 Hub

The Hub was like our coaster with respect to the fact that it has a load sensor attached and could communicate with the coasters via Bluetooth. The purpose of the hub was to collect the surrounding weight data from the coasters and then subtract the weight of the cup from those measurements. Using the keypad, we would choose a specific coaster and the hub would store the current reading from the attached load sensor circuit as the cup weight. Once we know the cup weight, we removed it from the incoming coaster measurement and now have the weight of the drink itself. This drink weight value is then sent to a database. For the sake of our project, we used a very popular time-series database called influxdb. We then proceeded to use an open sourced graphing platform called Grafana to visualize this data in real-time.

2.5.1 Hub Power Supply

Since our hub does not need to fit into the size of the coaster, we can use a bigger battery, so we chose to use a 9 Volt Battery and then used a buck converter to convert it to 5 Volts to be used by the load sensor circuit. On top of that, we used another buck converter to convert the 9 Volts to 3.3 Volts to be used as the supply voltage for the microcontroller.

Since we also had a load sensor attached to the hub, we needed to invert the voltage to obtain our drive voltage. Using the inverting op amp equation in 2.4.2, we used the five Volt power line to invert since it was a consistent 5 Volts and then determined R2 was 10 kΩ and R1 was 50 kΩ

2.5.2 Hub Keypad

The other difference between our coaster and hub was the keypad, which we connected to the GPIO pins of the microcontroller. The plan for the keypad was to input a coaster number on the keypad, and then the microcontroller would take the current reading from the load sensor and use that as the base cup weight.

With the keypad, we had trouble interfacing it using our microcontroller. We were unable to register any sort of reading from any of the pins. In order to understand the keypad, we tried interfacing the keypad with an Arduino, but from there we saw that after we pressed a button, it would hold that value for a while even when we let go of the button.

3. Design Verification

We tested all non-power supply modules using our BLE400 Motherboard as a secondary power supply and then used a power supply set to the voltage of its respective battery to test the power supply modules. We tested each module independently of each other.

3.1 Microcontroller/Bluetooth Verification

In order to test the microcontroller, we had to make sure that we were able to flash the microchip with our own software that was running on some soft device. This was our first verification, which was to get the program to "blink" across all the LEDS on our debug board. Once this was successful, we knew that the programming aspect and microcontroller capabilities were verified as working.

As for the Bluetooth verification, after we had the microcontroller up and running with peripheral software, our main verification was having our nRF51 chips send a pack to each other and also receive a data packet from each other.

We tied both of these verifications together into one larger one where we had the microcontroller send and receive BLE packets and when they were sending using, the ADC so we had some value attached to it as well. This was the basis for our project thus getting it to verify was significant.

3.2 Power Unit Verification

3.2.1 Coaster Unit

We tested the boost converting circuit on the coaster by setting a power supply to 3 Volts and inserting into a breadboard model of the circuit we used. Then using an oscilloscope, we probed the output voltage to confirm that we got the 5 Volts we originally expected and we did.

We ran a similar test to confirm that we were getting the correct inversion voltage of -1 Volts in our design. In order to provide -3 Volts, I used to power supplies, each set to 3 Volts, but connected the ground of one power supply to the voltage high of the other power supply, so that I would then have a reference voltage of 3 Volts, 0 Volts, and -3 Volts. We then probed the output of our op amp circuit and saw that we were getting -1 Volts as we expected.

3.2.2 Hub Unit.

Since we powered our hub with a 9 Volt Battery, we set our power supply to 9 Volts and then used a breadboard model to test both of our buck converting circuits. By using an oscilloscope, we probed the output voltages for both circuits and saw the voltages we expected.

We ran a similar test but with 5 Volts and -5 Volts for an inversion, circuit and we saw the -1 Volts that we expected

3.3 Load Sensor Verification

In order to test the load sensor, we powered all components using our BLE400 Motherboard and connected the output of the load sensor circuit to one of the ADC pins on our microcontroller. We then used a UART bridge to collect those values from the microcontroller to display on our terminal. We were able to confirm that we obtained readings from the load sensor because the value changed as we applied pressure to the sensor. As expected, the output changed as we changed the pressure applied.

Now that we were able to collect and read data from our sensor, the next step involved calibrating the sensor and determining a relationship to convert the decimal reading we got from the microcontroller to an actual weight. In order to do this, we measured out various different amounts of liquid on a digital scale and then recorded the output decimal value. Once we had a decent amount of points for the range we wanted to measure, we plotted the data in Matlab to determine a line of best fit. To our disappointment, we did not have a linear relationship, but instead found that a fourth degree line was the ideal relationship from our data as shown below in Figures 7 and 8. The blue lines indicate the fourth degree line of best fit.

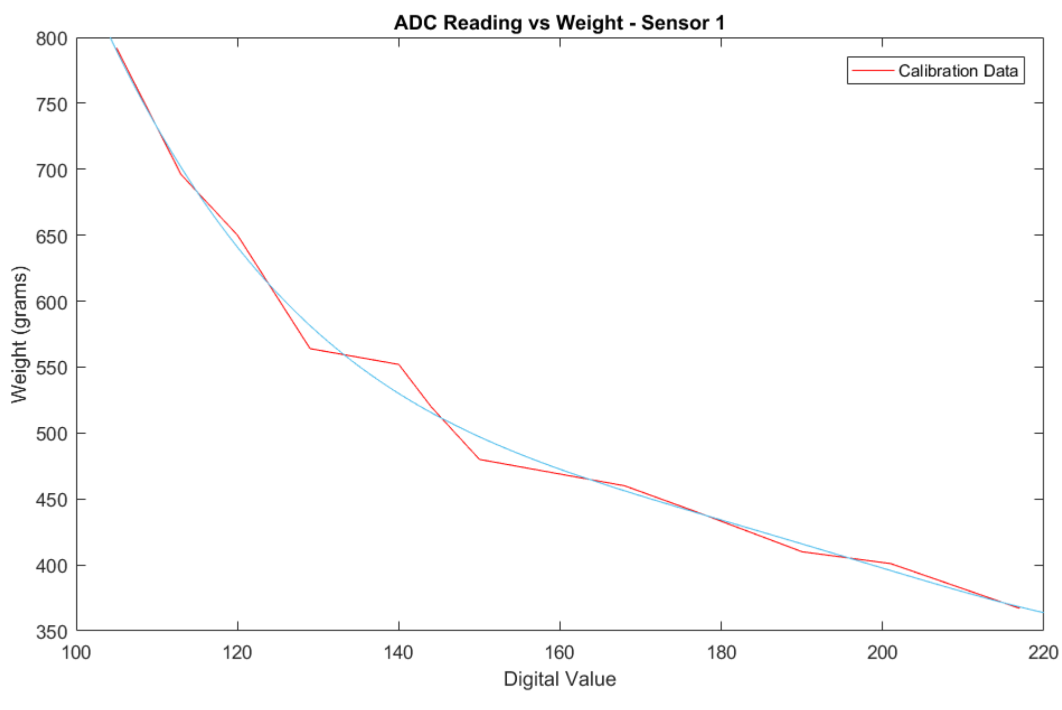

Figure 7. Coaster Load Sensor

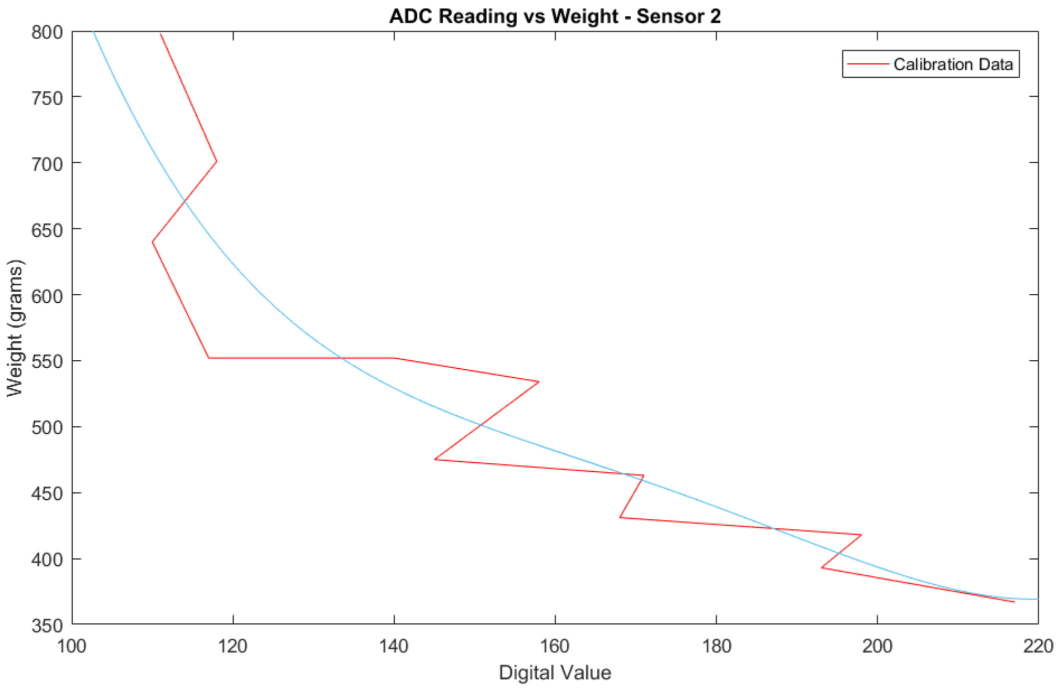

Figure 8. Hub Load Sensor

4. Costs

4.1 Parts

Table 1. Parts Costs

Part	Manufacturer	Retail Cost ($)	Quantity Purchased	Actual Cost ($)
Load Sensor	Flexiforce	24.99	4	99.96
nRF51822 Package	Nordic Semiconductor	3.89	4	15.56
OpAmp	Texas Instruments	0.57	10	5.70
Balun	STMicroelectronics	0.53	10	5.30
Antenna	Linx Tech	2.09	5	10.45
16 MHz Crystal	Kyocera	0.53	5	2.65
32 kHz Crystal	Kyocera	0.59	5	2.95
Boost/Buck Converter	Linear Technology	3.61	5	18.05
Circuit Passives	RLC	0.50	30	15.00
Coin Cell	Energizer	0.75	5	3.75
9V Battery	Energizer	1.50	4	6.00
Number Pad	Adafruit	3.95	1	3.95
PCB Coaster	PCBway	3.10	4	12.40
PCB Hub	PCBway	3.10	1	3.10
Total				**204.82**

4.2 Labor

Table 2. Labor Costs

Name	Hourly Rate	Total Time Invested	Total Cost ($) (Total Hourly Pay
Shivam Patel	$40	180	$18,000
Shray Chevli	$40	180	$18,000
Suraj Sinha	$40	180	$18,000
Total			$54,000

4.3 Total Costs

From Tables 1 and 2, the total costs were $54,204.82

5. Conclusion

5.1 Accomplishments

As part of our project, we successfully completed all of the requirements and verifications that we had set for ourselves. We created a system of coasters and a hub that could communicate with each other, send data, and receive data. Along with successfully creating a means of communication between these coasters and the hub, we also successfully calibrated the sensors on these coasters to get correct data readings. We used a regression model to correctly understand how the analog voltage readings from the sensor translated to readings in grams. In conclusion, in order to have this system functional, we got the load sensors, microcontrollers, RF transmission and reception and power supply all working.

5.2 Uncertainties

After completing the project, we found that all components of what we had planned are working. However, in our demo, we only had the resources to create one coaster and one hub. In order to better highlight and test the functionality that we had planned, we need to create multiple coasters as well as hubs. Another uncertainty in our project was that the number pad connected to the hub did not work. Therefore, because there was only one coaster, the system worked, however, we had no means of checking if this would work with multiple coasters.

Due to some issues with the PCB, we were unable to test the full capabilities with our PCB due to some issues we realized after the final round of PCB orders. For the inverting op amp configuration on the PCB, as shown in Figure 4, I connected the lower bound voltage of the op amp to ground and thus our voltage will never be inverted. On top of that, we connected the passives and crystals to the microcontroller as specified by the user manual, yet when we tried to program the microcontroller on our PCB, we ran into issues and we were never able to confirm that we were actually writing to the PCB.

On top of that, we are using a 9 Volt Battery, which has a standard lifetime of 500 mAh, to power our hub and our current consumption of the Hub is around 15-20 mA because it is constantly receiving via Bluetooth. This setup gives us an estimate of 25-hour lifetime for the hub, which is inadequate in every sense for a restaurant.

5.3 Safety Considerations

5.3.1 Hardware Safety

For this project, we will be using a lithium-based coin cell battery. In general, any energy storage device (battery) carries the risk of overheating or exploding. Cells with ultra-thin separators of 24µm or less are more susceptible to impurities, so we will make sure we do not use such batteries [9]

A common issue with using lithium-ion batteries tends be thermal runaway. This occurs when microscopic metal particles converge to one spot and a short develops and a sizable current will flow between the positive and negative plates. The temperature rises; this is also referred to as 'venting with flame.' [9] In order to avoid this, we will closely monitor the temperature of the lithium-ion coin cell temperature and keep the cell away from sensitive technology. This also means that we should be very careful with our calculations and ensure that we do not have too much voltage going through, prior to attaching the battery.

Our goal with this product is to make this as versatile as possible. As a result, we aspire to create a product that functions with any beverage. Therefore, we must take into account the varying temperatures that this coaster may be exposed to. This poses a risk, as there will be technological components such as batteries and sensors in close proximity with somewhat hot or cold temperatures. As a result, we need to ensure that the coaster is thick enough or is of some material that shields the battery and sensors from the potentially extreme temperature.

Similar to the previously raised concern, since this is a beverage coaster, there will be inevitable exposure to liquids. We must ensure that the components consisting of charge or electrical components are separated from any liquid exposure potential. This means that the coaster itself must have no leaks or gaps that expose the internal circuitry.

5.3.2 Network Security Safety

When using a number of connected devices, sometimes, with the large numbers of devices that have wireless communication capabilities, we run into danger of overloading the network causing it to fail to deliver data packet. If too many IOT/Bluetooth devices try to send packets at the same time, the congestion could cause a buffer overload, if there even is a buffer. The reliability and functionality of the network would be broken, thus is a risk we need to hedge against.

5.4 Ethical considerations

We, the members of group 61 in ECE 445, understand the impact that our project can have on the daily lives of many users. As a result, we understand that it is of utmost importance to follow the highest professional and ethical standards and commit to all ten of the IEEE Code of Ethics [10]

More specifically, in the process of constructing this beverage coaster, we must take into account #1 and #9 of the code of ethics. We must hold the safety, health and welfare of the public paramount. We will be dealing with beverages that to be consumed by our consumer. As a result, we must not actually alter the beverage in any way. We designed our product to take readings on the pressure applied by the drink and cup, and we must stop at that. #9 also falls under a similar situation. We must avoid injuring others, their property, reputation, or employment by false or malicious action. The same applies of not altering the actual liquid consumed.

We must also take into consideration #3 on the code of ethics. We must be honest and realistic in stating claims or estimates based on available data. Various services such as restaurants will be relying on our system to know when to serve/refill beverages. We must ensure that these readings are accurate, as advertised. Throughout the building of this project, we must also take into account #7 of the code of ethics. We must seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contribution of others. We must take into account all that our TA and course staff suggest and put significant consideration into those pieces of advice.

5.5 Future work

Though our current proof of concept is self-sustaining, there is no way of graphing the data collected in real-time. The current system stores the data into comma-separated values (CSV) that we later export into a database. The goal is to make this entirely self-sustaining and in order to achieve this; we would need to add Wi-Fi capabilities to our hub. By adding Wi-Fi capabilities, we can export the data into a database and therefore see the results in real-time.

In addition to the real-time visualization, we would also like to add a feature to the coaster that serves as a button to call the server. Since we have the technology in place, this added feature would only add as a selling point to our product. We could create some sort of queueing system so the server knows whom to serve and when.

Other future work also involves fixing the issues with the PCB such as the lower bound voltage for the op amp. By fixing these issues, we can then actually read the load sensor reading using only the PCB instead of the perf board additions. Along with the op amp, we also need to go through the user manual again to confirm all connections to the microcontroller are correct so that we can use the microcontroller on the PCB to transmit the data instead of an external nRF51822. The other option that we can choose to use in the event we still have trouble with the microcontroller on the PCB is to use an ATMega328P to handle all data collection and then have a separate IC to control Bluetooth TX and RX. This would be a simple fix and allow us to use the Arduino IDE to program but it may also run into issues with power consumption.

Regarding the power supply, we would like to transition our hub from a 9 Volt battery to a wall outlet since the hub does not need to move, and it will allow us to extend the hub lifetime well past the current estimates of 25 hours.

Bibliography

[1] J. Kiesel, "The Internet of Things in Restaurants," Restaurant Technologies, 8 February 2018. [Online]. Available: https://www.rti-inc.com/internet-things-restaurants. [Accessed 22 February 2018].

[2] F. Technologies, "Top 10 Ways Internet of Things Can Be Used By Restaurants," Finoit, 3 October 2017. [Online]. Available: https://www.finoit.com/blog/top-10-ways-internet-of-things-can-be-used-by-restaurants/. [Accessed 19 February 2018].

[3] M. Kern, "'Internet of Things' Insights for the Food Service Space," Modern Restaurant Management, 8 May 2017. [Online]. Available: https://www.modernrestaurantmanagement.com/internet-of-things-insights-for-the-food-service-space. [Accessed 22 February 2018].

[4] A. Carman, "This connected coaster is a silly solution to keep your drink safe at bars," The Verge, 20 June 2017. [Online]. Available: https://www.theverge.com/circuitbreaker/2017/6/20/15838366/bluetooth-coaster-brio-release-announcement. [Accessed 17 February 2018].

[5] Nordic Semiconductor, "nRF51822 Product Specifcations," July 2016. [Online]. Available: http://infocenter.nordicsemi.com/pdf/nRF51822_PS_v3.3.pdf. [Accessed March 2018].

[6] Tekscan, "SparkFun," [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Pressure/A401-force-sensor.pdf. [Accessed March 2018].

[7] Nordic Semi, "Determing capacitor size and maximum sampling frequency for a nRF51 ADC voltage divider," [Online]. Available: https://devzone.nordicsemi.com/attachment/428b530562fded0e484c166e76d88129. [Accessed March 2018].

[8] Linear Technology, "LT1111 - Micropower DC/DC Converter Adjustable and Fixed 5V, 12V," [Online]. Available: http://cds.linear.com/docs/en/datasheet/1111fd.pdf. [Accessed May 2018].

[9] "BU-304a: Safety Concerns with Li-ion," Battery University, 4 January 2018. [Online]. Available: http://batteryuniversity.com/learn/article/safety_concerns_with_li_ion. [Accessed 9 March 2018].

[10] IEEE, "IEEE Code of Ethics," [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed March 2018].

Appendix A Requirement and Verification Table

Table 3. System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
Microcontroller		
1. Ensure that the microcontroller can send out signals to the Bluetooth IC for Bluetooth communication	1. Verification a. Prepare a basic signal to be sent out from the microcontroller to the IC antenna b. Using another antenna for receiving, confirm we receive the same signal original sent out	Y
2. Ensure that the microcontroller has a GPIO pin that can read an analog voltage of up to 3 Volts with accuracy of 10%	2. Verification a. Connect a function generator to one of the GPIO pins on the microcontroller b. Have the function generator sweep through different voltages c. Have microcontroller display analog voltage reading on terminal d. Confirm that analog voltage is within threshold	Y
Load Sensor		
1. The load sensor should be able to track the weight of at least a 7.5 lb. glass cup with up to 1.25 lbs. of liquid in it within 5% of the actual weight. (This weight allows for versatility in cup and drink selection)	1. Verification a. We can measure a cup with some amount of liquid and its specific amount on a digital scale b. Then we can measure the same cup with the same amount of liquid c. Using our microcontroller, we can read the analog voltage d. Convert the analog voltage to weight and compare it to digital scale reading	Y
Bluetooth Module		
1. The IC should be able to transmit a Bluetooth signal at 2.4 GHz to the antenna	1. Program the IC to transmit a basic signal at 2.4 GHz through the antenna and confirm that signal is received at another receive	Y
2. The antenna should have a range of 18-22 meters	2. a. Program the IC to transmit a signal.	Y

a. After further review, we appended this to 50-75 meters	b. With a Bluetooth Receiver, confirm we can still receive the Bluetooth signal within the specified range	
Power Module		
1. Boost Converting circuit should output a voltage of 5 Volts with an accuracy of 5%.	1. Using an oscilloscope, measure the output voltage and confirm it is within specified range	Y
2. Inverting circuit should output a voltage of -1 Volts with an accuracy of 10%	2. Using an oscilloscope, measure the output voltage and confirm it is within the specified range	Y
Hub Module		
1. The number pad should allow you to press a button and the resulting number should be sent to the microcontroller	1. Connect microcontroller to terminal via UART and the digit of the button pressed should be displayed on the terminal	N
2. Bluetooth receiver should be actively listening for a signal from the coaster and receiving it when it does come in	2. Program Bluetooth transmitter on coaster to broadcast and confirm that we receive the same signal on the receiver	Y
3. The microcontroller should be able to receive UART signals	3. Verification a. Connect microcontroller to a USB UART bridge and then connect it to a terminal b. Send it a string of 50 characters and have it echo back the characters c. Confirm that received string matches sent string’	Y

Appendix B Coaster Designs

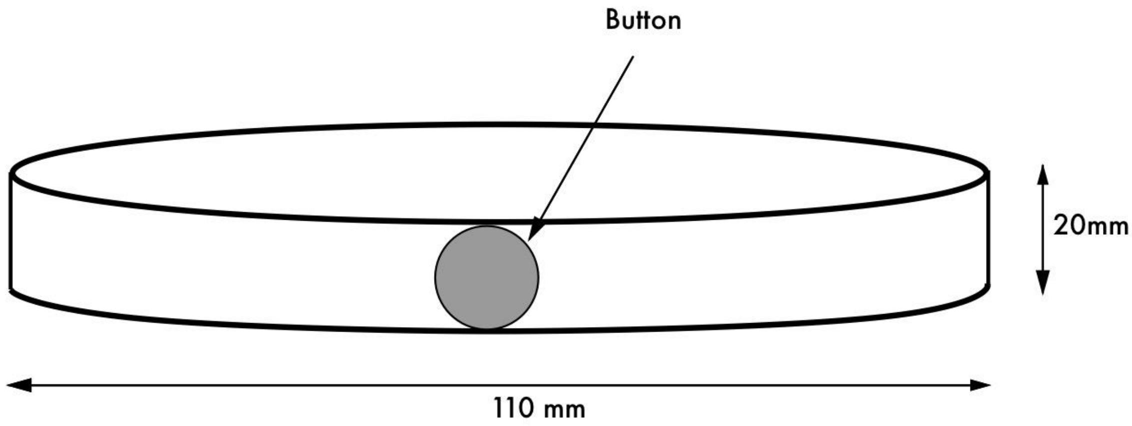

Figure 9. Coaster Side View

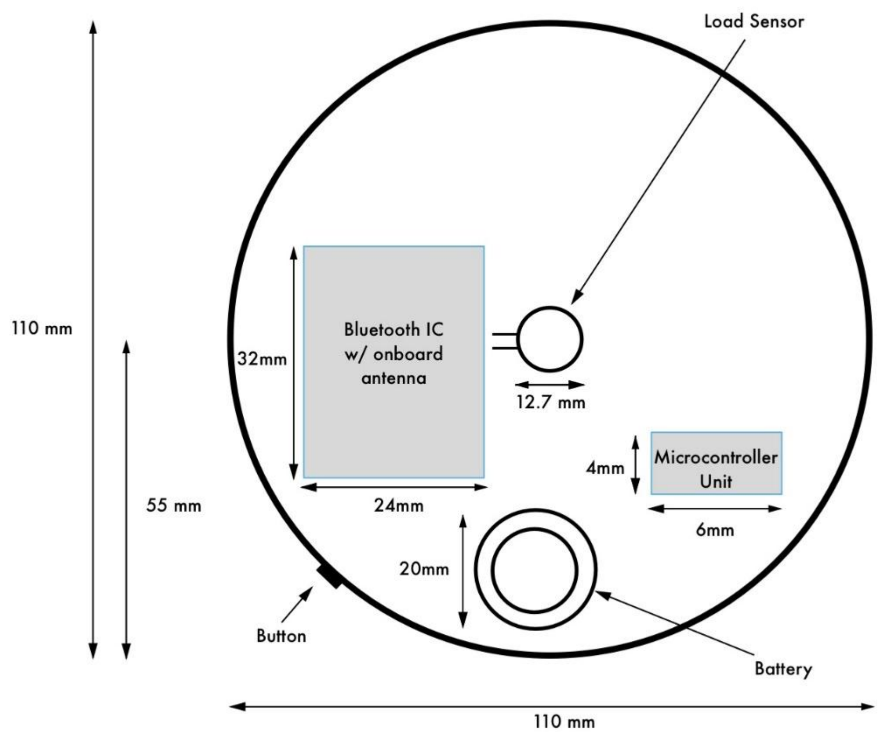

Figure 10. Coaster Overview

Figure 11. Demo Coaster