Clever Coasters

University of Central Florida

Department of Electrical Engineering and Computer Science

Dr. Lei Wei

Group C

Rubba Ashwas	Computer Engineering
Mitchell Crozier	Computer Engineering
Teodotas Kursevicius	Computer Engineering

Table of Contents

Figures

Tables

1.0 Executive Summary

The Clever Coaster system was designed with cost and ease in mind, while also tailoring to the fact that different customers require different attention. Some want their water refilled when its halfway, while some never even take a sip of their drink until their meal arrives. Whichever the case, the Clever Coaster System integrates weight sensors and wireless communication to provide a tailor-made experience for every restaurant goer.

Clever Coasters design team had three goals in mind when we set out to design this product

- Low power
- Low cost
- Low maintenance

We accomplished these goals by utilizing the TI CC2540, an efficient BLE-enabled microcontroller, a NFC reader/writer to make swapping coasters between tables seamless, and an easy to use charging station that eliminates the need to have a wire for each individual coaster, and a rugged case that can weather the abuse that a coaster might see while maintaining the integrity of its looks and electronics. We reached these design choices after considerable consideration of other options available for us to pursue.

There aren't many commercial products available to compete with ours. The closest thing would be the terminals some restaurants employ that let you pay your tab right at the table, but those lack the personalized touch that Clever Coasters provides, so it could potentially by an untapped space in one of the biggest industries in the world; the food industry.

Once we hit market, having a competitor copy our design and try to encroach on our market space is a serious concern since we don't have a large team that is trying to sell these products.

Our design team strived to make as many cost saving choices as possible without sacrificing quality to be able to provide as low as cost of manufacturing as possible. This would help us appeal to a wide range of restaurants, from the high-end chains to the locally owned stores.

Clever Coasters has the potential to be as ubiquitous as the party-ready buzzer, in fact we took some design cues from that design to come up with our charging circuit.

2.0 Project Description

The project description is the explanation of the motivation behind the project, the goals and objectives of the project, the function of the project, and the requirements for the project that arise these ideas. It also includes an analysis of the different marketing and engineering requirements and how they affect one another. Finally, it has an operations manual that describes how to use the project for its intended purpose.

2.1 Motivation

Most people who have eaten at restaurants know that their experience can vary wildly depending on the attentiveness of their waiter. A conscientious waiter can always keep your drinks filled up, take your order when you decide what you want, make sure you have everything you need, and hand you the check when you're ready to go, all while seeming almost invisible and completely unobtrusive.

Unfortunately, restaurants tend to get very busy at certain times of the week and each waiter has to handle a large number of tables, all of which could be scattered around the restaurant. A waiter has to constantly monitor all of his tables while going back and forth to the kitchen to get food moving. It can become exhausting and stressful for the waiter managing numerous tables and frustrating for the thirsty customer with their empty glass at the corner of the restaurant. The customer would like to ask the waiter a question as well, but hasn't seen him for several minutes.

What if a waiter had tools that let him know when to refill a specific customer's drink or when to check in on a table that requires attention? That is the motivation behind the smart coaster system for restaurants: A tool to help improve the workflow of restaurant staff and the improvement of the guests' dining experience.

2.2 Goals & Objectives

The goal of this project is to create a system of low-power, cost-efficient smart coasters that can wirelessly connect to a device at the table, which then connects to a main display where information about their current state is provided. The coasters should be easily chargeable in large numbers, ensuring that a minimal number of wires are needed. The coasters should last through at least a full day of service before needing to be re-charged. Also, they should seamlessly connect to the device at any table they are placed. The coasters should have the ability to detect if a drink placed on them is empty or full, the logic of which is outlined in Figure 3. They should give the customer the ability to call their waiter, request to not be disturbed, and ask for the check using a combination of sensors and buttons(Figure 2) .

The table devices and smart coasters should be rechargeable to save costs. The table devices should act as the communication point between the coasters and the central display panel. They should connect to the coasters wirelessly and transmit their status to the central display.

The central display panel should provide information to the waiters about the status of each table. It should be able to separate the tables into groups based on which waiter is assigned to which table.

2.3 Function

Clever Coasters would allow waiters to spend less time worrying about refilling drinks and more time having meaningful interactions with customers. The smart coasters would individually estimate when a cup needs to be refilled and notify the waiters wirelessly. They would also give customers the ability to change the modes of the system to better suit their individual preferences. This system's function is to increase the efficiency and effectiveness of waiters, thereby decreasing their work-related stress while improving the customer experience.

2.4 Requirements and Specifications

The following set of requirements and specifications define all the features that reflect the goal of our clever coasters. These are, for the most part, measurable, and attainable. Our main objective is to ensure accuracy while reducing power-consumptions.

1. Water shall not be able to enter the electrical system
    a. Coasters shall follow IP44 standards [7]
2. Coaster shall have wireless connectivity
3. Table device shall have wireless connectivity
4. Display shall have wireless connectivity
5. Coaster shall detect weight within a margin of 50g.
6. System shall have an input that allows user to page a waiter
7. Table device shall retrieve information from the individual coasters
8. Table device shall send information to the employee display
9. Employee display shall follow guidelines of User Interface Design
    a. Employee display shall follow principle of consistency

10. System shall have multiple states
    a. System shall have a do not disturb state
    b. System shall have a stealth state (all LEDs off)
    c. System shall have a neutral state
    d. Only one state can be on at a time
    e. LEDs shall indicate which state is active
11. When cup reaches 30% capacity, LEDs shall turn on unless in stealth state
12. Cup shall not fall over when placed on coaster
13. Coasters shall establish wireless connectivity with table device within 10 seconds
14. Coasters should be able to send data to table hubs within 15 seconds
15. System shall follow Wireless Standards of communication
16. The Coaster should be able to pair with specific table hubs
17. System shall cost below $500
18. Power consumption shall be under 5 W
19. Wireless Connectivity shall reach lengths greater than 2 meters for interconnecting coasters to table hubs
20. The system shall be rechargeable
21. The system shall be power efficient
22. The system shall be safe to use
23. The coasters shall be able to stack on each other

2.5 House of Quality

The house of quality is a tool used in business and engineering design to correlate marketing requirements with engineering requirements and specifications. It is a useful way to organize the two against each other and see how they help and hinder each other's' achievement.

The key marketing requirements for Clever Coasters are a low cost system, good battery life, easy to charge, durable, and smart. The engineering requirements relative to these are cost, power consumption, wireless range, sensor accuracy, water resistance, and response delay. Figure 1 is the house of quality for Clever Coasters.

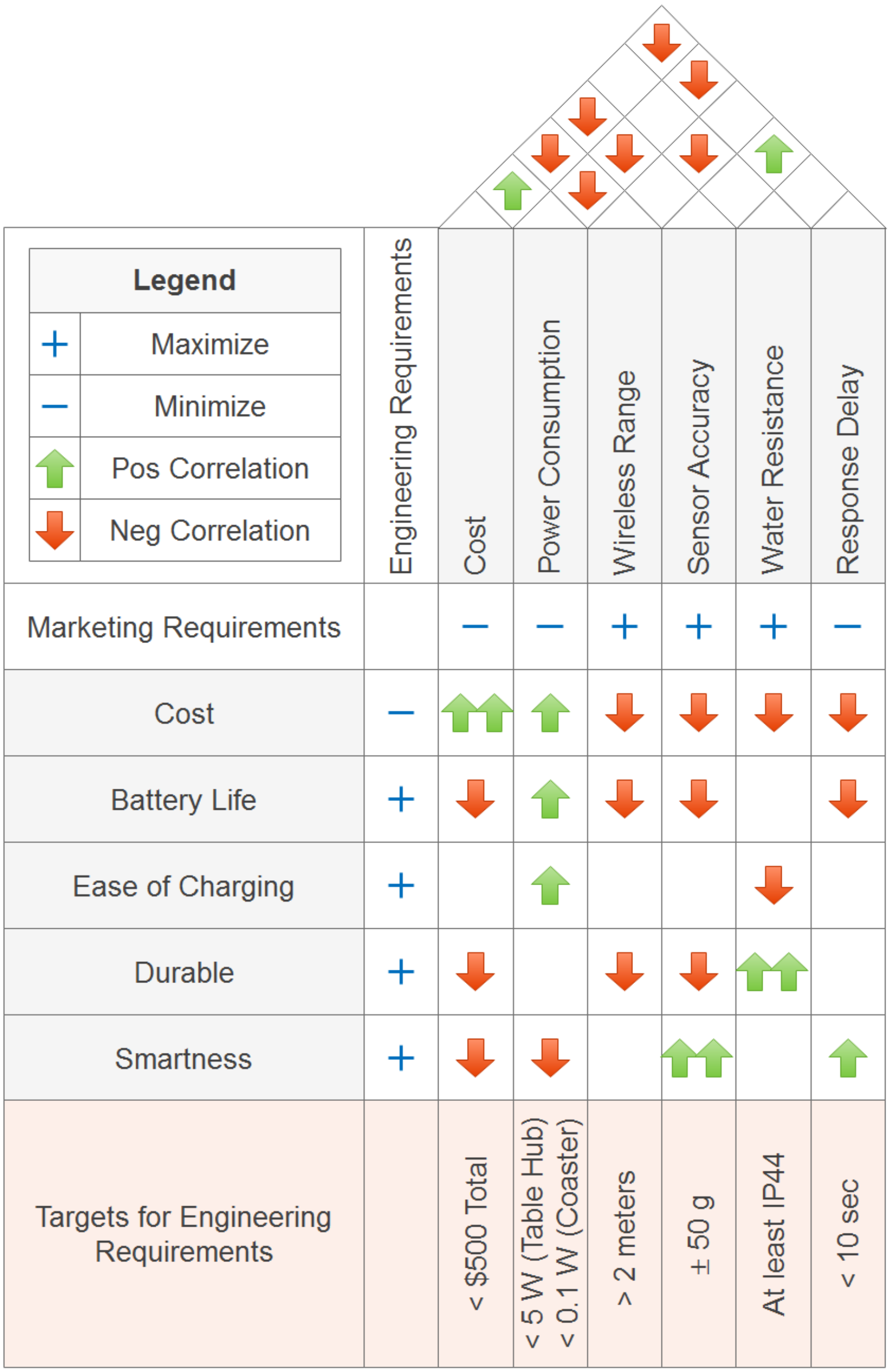


Figure 1: House of Quality

It is clear from Figure 1 that while cost is a very important factor, it correlates negatively with most of the other requirements. This means that much research will be required into the components to ensure that these requirements can be balanced against each other. Cheaper components may result in poorer performance while expensive ones will make the end product unattractive to cost-cutting customers.

The other key requirement is power consumption which is important because these are wireless devices operating on battery power. It is, however, positively correlated with cost, battery life, and ease of charging so if we focus on this requirement is may be possible to also meet the other three easier. That said, there will still need to be lots of research into the power of the system because it is a vital component.

Ease of charging will be more related to the design of the charger than to the other requirements, but there will still be several options with their own advantages and limitations. Aside from water resistance, which is a real engineering concern in a restaurant environment, the other requirements don't affect this too much. The requirement of water resistance means that we may have to come up with interesting ways to power and/or charge the coasters.

Durability is mostly going to be related to water resistance. It does, of course, also increase costs and affect how well the sensor and wireless communication components can operate. A metal box would be quite durable, but it would be completely unable to fulfill the function of the system. Since all of the components are solid-state, movement should not be too much of an issue. Water penetrations could be a big problem since all of the components are electrical and could take damage from it.

The smartness of the coasters will largely be determined by the features we add to them, but the primary idea is related to measuring the weight of the cup placed on top of them. Research into this area and testing of different options will be critical to ensure that the coasters can actually accomplish the tasks that they are meant to do. Unfortunately, better solutions here are likely to be far more expensive and might use more power. Cheaper solutions, on the other hand, could be unable to meet the requirement to detect a change of 50 grams which may mean they are not suitable for our application.

2.6 Hardware Block Diagram

Before researching individual components, it is important to recognize the overall modularity and major hardware components needed to make the Clever Coaster system a reality. As shown in Figure 2, the microcontroller acts as a central control system for integrating and connecting all components together. Likewise, the pressure sensor will be important, and lots of research will need to be done to determine the most accurate and reliable method of detecting the weight of the cup. With this time-sensitive data, quick and energy-efficient wireless communication technology will need to be incorporated as well, to facilitate the flow of the overall system. We will focus mostly on hardware research for this initial phase of the Senior Design project.

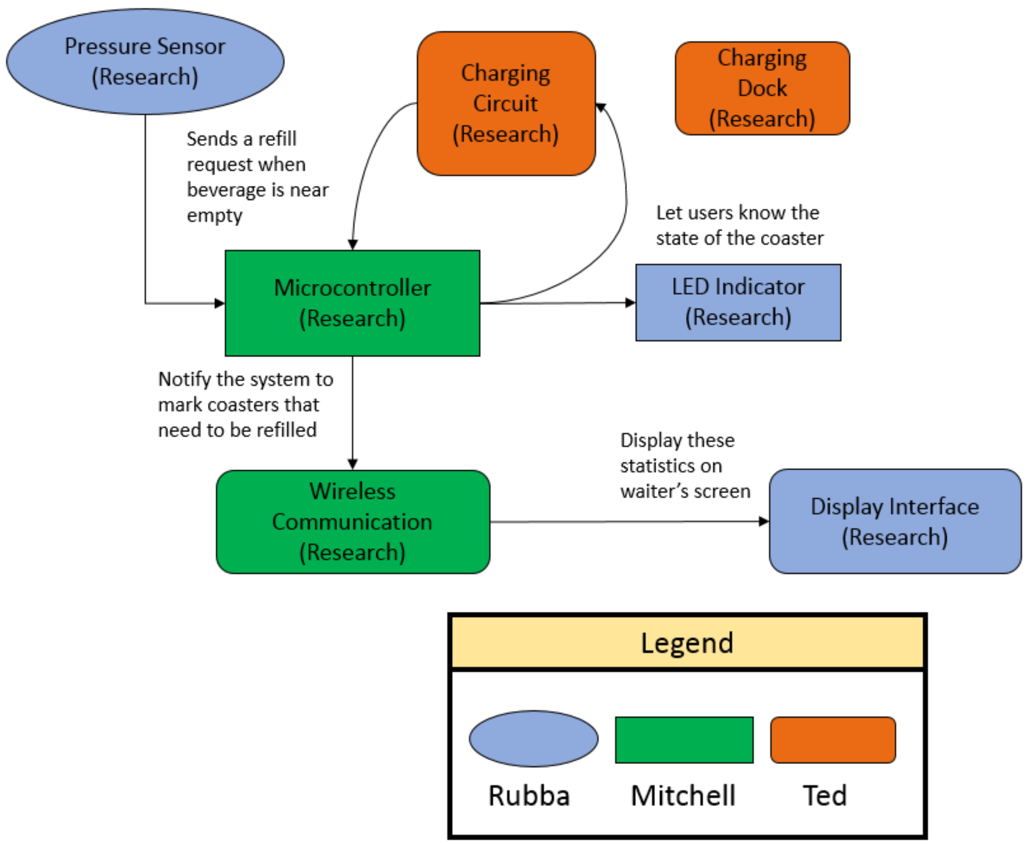


Figure 2: Hardware Block Diagram (Research Phase)

2.7 Project Operations Manual

Clever coasters is a small, low power, wireless coaster with an easy pairing system to allow it to seamlessly move between tables if needed. Each coaster comes equipped with both a Bluetooth low energy and NFC (Near Field Communication) module, and at each table is a device that also has Bluetooth, with an NFC tag containing its Bluetooth address on it for pairing.

To start using an unpaired device, the user or waiter must pair the coaster with the table device. When a coaster comes into extremely close proximity to this NFC tag, its onboard NFC reader activates and reads the Bluetooth address on the tag, and then the Bluetooth module attempts to connect to the table device.

Once connected, the coaster monitors the force sensor attached to the top surface, and calculates the percentage remaining in whatever drink is placed upon it. After the drink drops below a certain threshold, it uses the Bluetooth module to send an alert to the table device that its drink is getting low. All that is required of the user during this period is to enjoy their beverage.

Upon receiving the drink low alert, the table device uses WiFi to send a request to the main hub that notifies a waiter that the device's table has a drink that needs a refill. When the waiter comes and fills the cup up, the coaster recognizes this long term change in weight as a refill and starts monitoring for a drop below the threshold once again. A user can bypass needing the drink to be completely empty by pressing a button and requesting a refill.

3.0 Project Research

The first step in the development of Clever Coasters is to research prior implementations of similar systems and all of the different components that will be part of our system. There are design considerations that need to be accounted for when choosing a component. Components tend to have both advantages and limitations that correlate with the requirements of the system and must be balanced against them.

Not only that, but a choice of one component affects the choice of another. In some cases, there are compatibility issues between components. In other cases, the specifications of one component may require that another component is either more versatile or more limited in its own range of specifications.

For example, a microcontroller and wireless communication module combination will, necessarily, draw a certain amount of power. Given the requirements of the system to function for at least one whole day of restaurant service, this imposes a requirement on the capacity of a battery. If a larger battery is required, the cost and size must be considered. It is unreasonable for a coaster to be very large and unwieldy so if the battery capacity should increase, then the energy density and shape of a battery becomes an important consideration. This limits the type of battery that may be used. For any given battery type, the method by which it may be charged can change.

Research into components is necessary to ensure that the system meets requirements and functions properly after it has been designed and assembled. Almost every component has some interaction with other components and a choice of one may limit the choices of the others.

For this research, we are considering different microcontrollers, wireless communications options, serial communications technologies, charging methods, battery types, low dropout linear regulators, weight sensors, tablets for use as a table hub, and miscellaneous components.

3.1 Existing Projects and Products

Although smart coasters aren’t made commercially available to restaurants now, there have been quite a few attempts at innovating similar prototypes. The presence of similar projects can help guide our product in the right direction in terms of component choice, feature integration, and overall functionality.

Our goal is to have our smart coaster exceed the expectations of the current prototypes out there, and provide much more detailed documentation that will help other people recreate the “smart coasters”. In addition to some smart coaster prototypes, there exist quite a few tablet applications with similar functionality. Integrating them will be key to creating a smart system.

3.1.1 The Ziosk Tablet

The Ziosk Tablet is a Wi-Fi enabled, android touchscreen tablet that has a great amount of functionality useful in a restaurant setting. Its purpose is to streamline the restaurant process and make things run quicker. The way it works is, it sits on the center of the table and acts as a connection between the server and customer. The customer can order from the tablet, bring attention to the server, pay with the credit card reader, print their receipt, and even take pictures to post on social media. Other features include completing surveys and participating in member rewards. This invention has proven to increase Return on Investments, and even increase the amount of customer surveys completed [13]. It's not necessarily smart coasters, but it provides a lot of the same functionality we hope to implement within our Clever Coasters system. We would like to incorporate some sort of existing, POS system into our overall design, to have coasters communicate with the tablets and provide interactive and behind-the-scenes features. The sample image below (figure 3) shows the different applications a Ziosk tablet can provide. In our product design, we are mainly interested in the connectivity and gaming functionality that can be provided through a tablet.

Figure 3: Ziosk Tablet

3.1.2 Arduino Controlled Smart Coaster

Mario Lukas has created a similar and interesting prototype of an intelligent coaster. Inside his version, there is an Arduino Mini Pro microcontroller, a Texas Instruments temperature sensor, and some LED strips. Also, his smart coaster uses Velostat to measure pressure and different weights placed on its surface. By cutting the Velostat into a round piece, he can place it within a round coaster, which is ideally something we would like to establish in our own coaster design. As for programming the LEDs and reading the analog input from the sensors (both temperature and weight), he writes in C/C++ in the Arduino IDE. For batteries, he uses a 3.7 V LiPo battery that is rechargeable. Rechargeable batteries can give our coasters a big advantage if the process of recharging these coasters was made easy. He poses new ideas and possibilities that the smart coaster can have, such as wireless charging, communication between coasters, and party games. [10] These features are ones we would like to try and focus on once figuring out the main weight sensor functionality. His version can be seen in Figure 4.

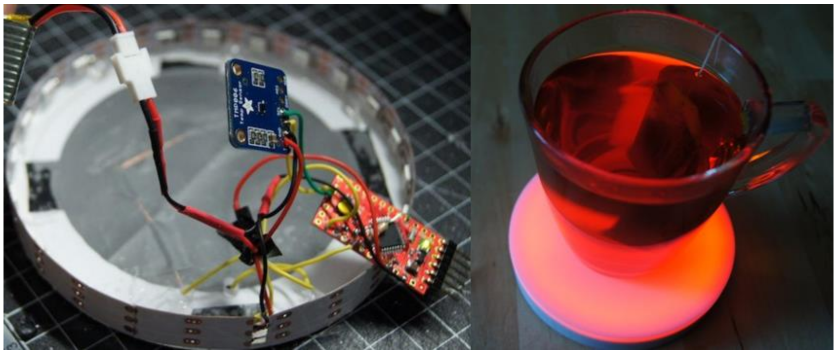


Figure 4: Internal Components (left), and heat sensor functionality (right)

3.1.3 WaiterLite Smart Coaster

George Crichlow has his smart coaster design laid out in figure 5 below. It shows where the order and layering of components lie within the casing design, which gives us some insight on how to organize everything. What is special about George's smart coaster design is that he goes in depth into the user-focused research. To understand the real-time application his coasters would provide to a restaurant owner, he interviews bartenders and waitresses from local restaurants to gain some perspective on what features and considerations to include in his design.

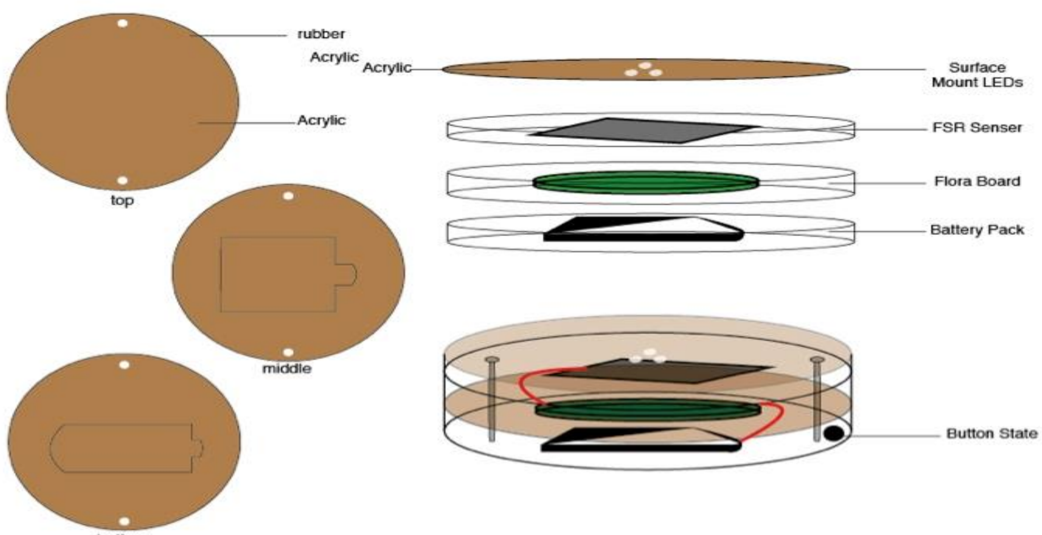


Figure 5: Sketch of existing coaster design

One piece of important feedback he received was to place the LEDs closer to the center in dimly lit situations to avoid distracting the customer [11]. His coaster detects when the glass is 25% full, whereas we are aiming for about a 30-35 % glass-full detection margin. George's uses the Arduino Floraboard, a Square Force-Sensitive Resistor (FSR), and RGB LEDs. The FLORA seems like a good option for a microcontroller because of its round shape. Future concepts mentioned in the tutorial include the addition of a “check out” button that allows you to notify your waiter to bring your check.

3.2 Wireless Communication

To design smart coasters with functionality that updates servers on near-empty cups, an efficient and reliable wireless communication system needs to be established. The clever coasters would then be able to send their data readings to the system that will notify the waiters of empty-cup locations and other relevant statistics. In this section, we will explore several wireless communication options and determine which one our project will move forward with and purchase.

3.2.1 Protocol Options

For communication to happen between two devices, there needs to be some sort of agreement between the two. Once this set of rules or “protocol” is established, packets can be sent and received via wireless communication (or wired). There are numerous kinds of protocols, but the ones we would like to utilize for our project will be based on the OSI Model.

The OSI (Open Systems Interconnection) Reference Model has seven layers that each have a specific layer function and aim to minimize information flow between the interfaces. Each layer uses its own protocol separate from the other layers, but performs some services for the layer above it. The lowest is the physical layer, which is responsible for transmitting raw bits through a communication channel. The topmost layer is the Application layer, which contains widely-used protocols such as HTTP (web browsing) and electronic mail. The remaining layers in between include the Data Link (error checking), Network, Transport, Session and Presentation (formatting) layers.

The advantage to sticking with the OSI model is that changes in one layer don’t affect the other layer, which makes troubleshooting, design, and software development easier. Considering it has been standardized as well, compatible technologies will be more readily available to integrate into our system, as we will discuss below.

3.2.1.1 Bluetooth Low Energy

Since the wireless coasters will need an idle mode for conserving power, energy-efficient Bluetooth connectivity can be established to ensure a low-cost and secure connection between devices. Bluetooth Core Specification version 4.2 provides several pairing mechanisms such as Passkey Entry, based on the specific capabilities of the devices. Encryption and privacy features can be implemented with this version of Bluetooth through methods such as "Man-in-the-Middle" and Passive Eavesdropping. [1] What is beneficial about Bluetooth Low Energy (LE) it doesn’t waste power maintaining a connection and has a low bandwidth. Bluetooth LE only connects as needed, making it a suitable wireless connection for reading sensor data.

3.2.1.2 ZigBee

ZigBee is a wireless protocol that operates in the 2.4GHz band, like Wi-Fi and Bluetooth. However, it operates at much lower data rates than Wi-Fi and Bluetooth. The main advantages of using ZigBee come from its very low power consumption and interoperability. With up to 65,645 nodes, it is a very robust and reliable wireless protocol, allowing easy addition or removal of nodes from the network. With an average cost of about $18, it seems to be costlier than using simple Bluetooth. [3] However, since it allows lower power consumption overall, it would save on battery costs. These advantages make ZigBee ideal for sensor networks or other control applications.

In ZigBee, the networks can be clustered, star shaped, or mesh. In each network, there is a device that initializes and controls that network. Routers can be setup for passing data, and other end points can be set to sleep mode. This would use very little power if the coasters need to be battery operated. [3]

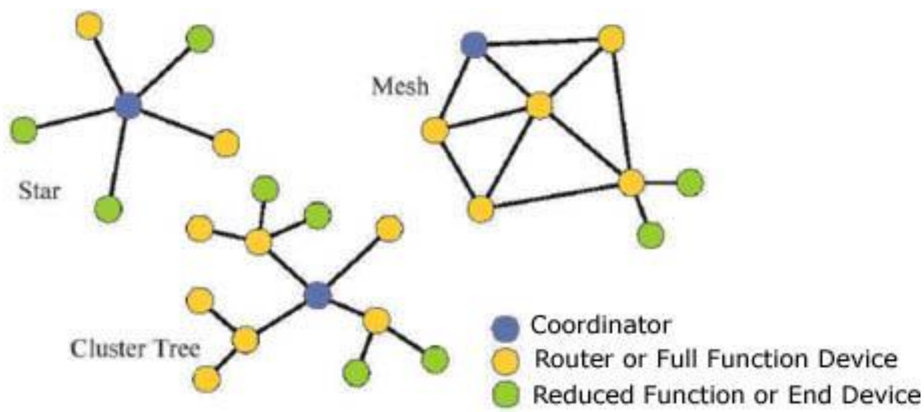


Figure 6: ZigBee Wireless Network Types

3.2.1.3 Wi-Fi

The wireless LAN (Local Area Network) standard is 802.11, which is more commonly known as Wi-Fi. It operates in unlicensed bands, where all devices can use its spectrum of frequencies. To allow other devices connected to Wi-Fi to coexist, clients must limit their transmit power. Clients (such as laptops and mobile phones) and access points (APs) or “base stations” make up the Wi-Fi network. What makes Wi-Fi useful is that it can handle movement of a client by switching to a nearby access point. It also can prevent collisions during multiple transmissions through the CSMA (Carrier Sense Multiple Access) scheme. [2] However, having the coasters frequently send sensor data through Wi-Fi may be an issue. Unlike Bluetooth and ZigBee, Wi-Fi uses more power and is more complex to connect to.

3.2.1.4 Radio Frequency IDentification

Radio Frequency IDentification (RFID) allows any non-electronic object to be part of a computer network. A stamp-sized RFID tag (sticker) is affixed to the object to allow for tracking. This tag consists of a microchip with a unique identifier that can be detected by an RFID reader. There are two types of RFID tags: active RFID and passive RFID. The passive RFID doesn't require a power source because energy is supplied in the form of radio waves emitted by the RFID readers themselves. The disadvantages associated with using RFID as the main source of communication between devices include its short range (1 meter) and its security issues. If there are multiple tags within a reading range (multiple coasters surrounding an RFID reader), collisions can occur if other measures aren't taken. Overall, RFID may still be useful in registering any coaster to a specific central table device using the RFID reader associated with that device. It is probably not the best for sending frequent data over the communication channels.

3.2.1.4.1 NFC

Near Field Communication (NFC) is a sub-field of RFID. NFC utilizes electromagnetic radio fields, and is intended for devices within close distance to each other. An active NFC device can read information and send it, while a passive device can only be read. There is security encryption in NFC, by establishing a secure channel over which sensitive information can be sent. Since NFC is specifically for short-range, we wouldn't be able to utilize it for sending sensory data over to the main hub. It would be most useful in pairing individual coasters with the table they are currently placed at.

The challenge in using NFC as a communication channel is that we would need compatible devices that can read the NFC tags associated with the object it is affixed to. This could add to our costs overall.

3.2.2 Summary of Options

Table 1 summarizes the different wireless communications under consideration. Our main concern is that we can achieve low power consumption and low cost. RFID tags seem to be the cheapest option and consume the least power. They have a short range though, which wouldn't be applicable. However, in terms of connecting multiple devices where data is sent frequently over a good range, Bluetooth Low Energy might be the better option here.

3.2.3 Selection of Communication Protocol

We will be moving forward with Bluetooth Low Energy, as the very low power consumption and low price is very important to us. We will be needing to receive continuous data from several sensors, which could get very costly when using a different communication channel. Bluetooth encryption features make it desirable for ensuring multiple coasters can all simultaneously connect to the main hub without risk of collision. We will also move forward with NFC tags to ensure that each coaster can be paired with its appropriate table hub.

Table 1: Summary of Wireless Options

	Bluetooth LE	ZigBee	Wi-Fi	RFID (passive)
Cost per tag or module	~$10	~$18	~ $7-20	~$2
Range (meters)	50	up to 100	50 - 250	< 3 m
Power Consumption	Very low	Very low	High	None
Network Latency	< 1 sec	< 30 ms	3 - 5 sec	< 100 ms

3.3 Microcontroller

The smart coaster system will be using one microcontroller for each coaster unit. The microcontroller units in the coasters will be used to monitor the sensor data and transmit data to the hub device wireless. The microcontroller will need to be low power to extend the amount of time between charges, have a fast-enough processing speed to interpret sensor data and relay it back to the hub device without any slowdowns or stutters, interface with our wireless connectivity modules seamlessly and be budget friendly since multiple coasters are needed.

3.3.1 Microcontroller options

There is a plethora of microcontrollers to choose from. Our team's first instinct was to choose the TI MSP430 variants due to it meeting most of our requirements of being low power, high speed, wirelessly compatible and budget friendly, plus we had all worked with it before and there is the option to go to the TI lab on campus for support. We also considered other more popular options, such as the Raspberry Pi, Arduino, and Beagle Bone, as their popularity would make finding solutions to any problems that might arise easier to find since odds are hundreds of people ran into the same issues.

3.3.1.1 TI MSP430G2553

This board was our initial first choice because of the familiarity we had with the product from previous classes. It is ultra-low power, consuming only 230 µA when the CPU is operating at 1MHz and 0.5 µA in standby mode. However upon further research we discovered it only has 24 GPIO pins[1], which could possibly limit how many sensors we want to add in the future to expand the functionality of the coasters. It is compatible with TI's NFC add on the TRF7960A, which can be used to both read and write NFC tags. The TRF7960A communicates over the MSP430G2553's SPI port[2], allowing us to use a RN-42 Bluetooth module soldered onto a PCB board, which communicates over the UART port. However, the cost of these devices start to add up, with the TRF7690A coming in at $4.17

per unit, and the RN-42 at a whopping $18.42 per unit, which while we initially thought was outside our budget, after evaluating the options below we came to the conclusion that we had to revise our budget since adding NFC was more expensive than we anticipated for most devices.

3.3.1.2 Raspberry Pi Zero

Our second initial choice, the Raspberry Pi has a lot of community and manufacturer support due to it being so popular. Its specs are incredible when compared to the MSP430- 1 GHz processor, 512 MB Ram, built in WiFi and Bluetooth, and all for only $10. However, the NFC add on for the raspberry pi is nearly $40 per unit, and the Pi is a power hog compared to the MSP430, with idle power draw close to 80 mA[3] compared to the mere 0.5 µA used by the MSP430. This high power usage means that we would need a huge battery, in excess of 2500 mAh most likely, in order to power each coaster for just one day. The extra battery size would make it more difficult to make an unobtrusive coaster. In general, the Raspberry Pi Zero would be overkill for our project in almost every aspect- processing speed, amount of memory, WiFi capability, HDMI out, while still not delivering the functionality that we require right out of the box such as NFC. For these reasons, plus the fact that the Raspberry Pi Zero is almost always out of stock, we decided to pursue other options.

3.3.1.3 M24LR Discovery

This board features a built in NFC reader/writer that can be used over SPI and UART[4]. The price is $22.61 per unit, significantly cheaper than the other NFC options. The processor on board is the STM32F103CB, an ARM-based 32-bit microcontroller with 128Kbytes of memory that operates at 72 MHz, which is much more similar to the MSP430 than the Raspberry Pi Zero. It has 80 GPIO pins, enough for our coasters, as well as a USB 2.0 port which we can use to address our Bluetooth needs. The NFC module supports all devices supporting ISO/IEC 15693 so if we went with this option we would have to ensure that our table hub can communicate with the coaster via NFC.

3.3.1.4 Arduino Uno

The Uno is based on the ATmega328P, a high performance, low power 8 bit microprocessor with up to 16Mhz throughput and 32KB of flash memory [5]. It only has 14 digital pins, but can go up to 24 by using the analog pins. The power consumption is relatively low, .2 mA in the active mode and just .75 micro amps in power save mode, which our application would most likely be using until it has updated data to send, meaning we could cut down on battery size for a slimmer coaster outer shell and less heat. Like the other microcontrollers, adding third party NFC capabilities to this board will be costly at $25 per unit. The unit itself is also not cheap, coming in at nearly $25 as well and is quite thick due to the USB port and tall pins, which could post problems for the outer shell design process

3.3.1.5 ATtiny85

The ATtiny85 is a high performance low power 8-bit RISC-based microcontroller, with a max speed of 20MHz 512 bytes of RAM, and 8KB of flash memory. With only 8 GPIO pins, it might possibly limit our ability to expand functionality with additional sensors in the future but might be worth the long-term trade off the future capabilities for better power consumption now. It also has the fewest number of serial ports of the other options, with only one SPI and one I^2C port. It is very cheap, with good documentation that should aid greatly with designing a PCB to socket the chip into.

3.3.1.6 ATmega32U4

The ATmega32U4 is in the same family as the ATtiny85, meaning it's still a low power 8-bit RISC-based microcontroller, but boasts a few improvements over the ATtiny85, such as 32KB of flash memory instead of the tiny 8KB, 2.5KB of ram instead of 512 bytes, and more port interfaces than the tiny, such as UART and an extra SPI port, as well as 44 GPIO pins compared to 8. All this extra functionality comes at the price of having a higher operating voltage and subsequently more power consumption.

3.3.1.7 TI CC2540

The CC2540 is a chip similar to the M24LR Discovery in the fact that it has a built in wireless component, this time a Bluetooth module instead of the M24LR's NFC. It has the most flash memory, RAM, and highest clock speed of any options we've looked at to date, and would allow for future expandability with a decent number of GPIO pins. Similar to the MSP430, TI making the chip allows us to utilize the on campus resources available to UCF students, as well as have excellent documentation, support and examples on TI's website.

3.3.1.8 Beagle Bone Black

The most advanced microcontroller of the ones listed here, the Beagle Bone Black is an option we considered for the table hub when weighing the pros and cons of using a consumer-ready tablet or building our own solution for the table hub. It comes equipped with plenty of IO ports

3.3.2 Microcontroller Comparisons

We narrowed down our microcontroller search to 4 units, the MSP430G2553, the M24LR Discovery, the Arduino Uno and the ATtiny85, selecting them based on power consumption, processing power required, cost, number of pins, and memory size. Items labeled in green means good with respect to the other options, yellow means decent, and red means poor.

3.3.2.1 Power Consumption

As the coasters are network connected small embedded systems that aren't meant to be constantly plugged in, power consumption is perhaps the single most important factor to consider when evaluating what board to use. We could eliminate the Raspberry Pi based on this factor alone, as it used multiple times more power than its competition. The smart coasters cannot exceed a certain dimension, which means that every bit of space matters and an easy way to save said space is with a smaller battery, which of course means we need a microcontroller with a small power footprint.

Table 2: Microcontroller Overall Comparison

	MSP430	M24LR Discovery	Arduino Uno	ATtiny85	TI CC2540
Power consumption	0.851 mW	0.722 mW	0.740 mW	0.370 mW	0.851 mW
Max Clock	16 MHz	16 MHz	20 MHz	20 MHz	32 MHz
GPIO pin count	16	41	23	8	21
Current output per pin	48 mA	80 mA	100 mA	40 mA	20 mA
Flash memory size	16 KB	8 KB	32 KB	8 KB	256 KB
RAM size	512 B	2 KB	2 KB	512 B	8 KB
Cost	$2.32	$21.25	$24.95	$1.20	$4.73

The power consumption numbers were obtained simply by using the formula P = IV, where I is the current at 1 MHz, and V is our assumed operating voltage of 3.7V. We assumed that we would be at a low clock speed and at normal temperatures due to the simplistic nature of the coasters. As stated in the cost analysis, there might be an issue with the CC2540 running at 1 MHz due to needing to keep the clock speed above a certain threshold to maintain a Bluetooth connection, but we're choosing to disregard that for the purposes of this analysis.

From Table 2, these microcontrollers are essentially neck and neck with each other, with the exception of the ATtiny85. Between the higher power usage chips, the M24LR comes out slightly ahead, using roughly 16% less power than the MSP430 and 3% less than the Uno and CC2540. However, they are put to shame by the ATtiny85, which would provide the best battery life for our coasters.

3.3.2.2 Cost

Cost turned out to be a huge issue for considering which microprocessor to go with. A typical restaurant might have anywhere from 75-125 seats, which means 75-125 coasters for the restaurant. Every dime saved in manufacturing adds up very quickly when you're producing that many units, so we strived to make the coasters as cheap as we could without sacrificing reliability and performance. For considering the cost, we only factored in the price of the microcontroller itself, and not the cost to get the wireless functionality required since it will be analyzed separately; an NFC module meant for the MSP430 might not work with the M24LR for example so we would have to factor in the cost of a wireless solution that works for the microcontroller of choice when evaluating our wireless options.

As show in Table 2, the ATtiny85 is the clear winner here in terms of raw cost coming in at just $1.20 per unit. The TI CC2540 also got a great grade here, due to the inclusion of the Bluetooth module built into the chip. The cost of adding Bluetooth LE to the ATtiny85 would be around $4.75, bringing the total cost for Bluetooth and MCU to over $6 per unit, so the TI CC2540 is the better value. The Arduino Uno is by far the most expensive, with a staggering $40 needed just to get NFC functionality working, while the M24LR is a more modest total of just over $40.

3.3.2.3 Memory Size

We considered 2 types of memory when evaluating the boards: Flash memory, where the programming on the microcontroller is stored and RAM, where temporary data such as the result of sensor calculation is stored.

From Table 2, the TI CC2540 leads the pack here by an enormous margin, providing 32x more flash memory and 16x more RAM than the weakest performer, the ATtiny85. Since our application will be monitoring sensor data and comparing it to previous data readings, RAM becomes important since we want to have enough data points stored to paint an accurate picture of what the status of the cup is. The ATtiny85 and MSP430's low amount of RAM could potentially present issues if we decide to go with it. Flash memory helps in writing more complex code, so if we implemented some type of game for customers to play with their interconnected coasters, the extra flash memory would be important to ensure that our game can be fully realized without being limited by how many lines of code we can fit, giving the CC2540 an advantage in overcoming both short and long term issues.

3.3.2.4 General Purpose Input/Output

GPIO pins were another important consideration for our project, since we might want to expand on the functionality of our coasters later and did not want to be constrained by number of pins available to us. We can use GPIO pins to monitor our sensor data, communicate with our other peripherals such as NFC and Bluetooth modules, and control LEDS and respond to button presses. Without any GPIO pins MCU's essentially become useless, so having enough to make our project work is critical.

According to Table 2, the M24LR leads the pack in terms of GPIO pins, allowing us enough ports for our team to feel comfortable that if we wanted to expand functionality in the future there will be enough resources there to do so. The ATtiny85 loses here with only 8 pins. Originally, our team thought that this number was just enough to allow us to achieve what we needed to, but upon further research, it would not be possible to have both Bluetooth and NFC function with this chip, because the SPI and I^2C protocols share some pins, detailed in Figure 9: Pinout Diagram for the Trinket 3.3V and we need to be able to communicate with both simultaneously when connecting to and from a hub. This discovery led us to re-evaluate what microcontroller was the best fit for our project. The TI CC2540 has the lowest current output, though the output current shouldn't make too much of a difference, as sensors and LED's don't typically need that much power draw to operate.

3.3.2.5 Clock Frequency

Clock speed is one of the least important factors to consider for our envisioned application since sensor monitoring is not a CPU intensive task. However, ultra-high clock speeds are usually correlated with high power usage, which is why our selected options all have similar clock speeds.

With all the options coming in at roughly the same speed with the exception of the TII CC2540 as seen in Table 2, it's hard to evaluate the best choice when it comes to frequency. Research led us to discover that the clock speed on the CC2540 needs to be roughly 32 MHz to ensure a reliable Bluetooth connection, however since the Bluetooth is integrated into the chip and the chip is very efficient itself, the power usage was in line with other MCU combinations during the worst case scenarios (Bluetooth on and transmitting, MCU operating at max speed). However, when considering the low power usage of the ATtiny85 compared to the other microcontrollers, it's reassuring to know that we won't be compromising on processor speed if needed. It is worth noting that while the clock speed may be similar, the ATtiny85's 8-bit CPU means that it will be less efficient.

3.3.3 Microcontroller Choice: TI CC2540

While all 4 microcontrollers evaluated had their own set of pros and cons, it all came down to price and support for the chip. The price savings of having Bluetooth built in and compatibility with one of the cheapest NFC modules we could find, as well as the chip itself being a bargain. In addition to this is the fact that TI provides excellent documentation and that there are detailed PCB design files available to guide us in the right direction for designing our PCB to accommodate the chip, along with the chips very small form factor, allowing us more freedom when designing the coaster enclosure all led to us choosing this microprocessor to power the coasters.

3.3.4 Bluetooth Module Comparison

Since each one our coasters will be using Bluetooth to communicate with the table hub, having a reliable Bluetooth module in each coaster is critical to ensuring a good user experience with the coasters. We will be evaluating 3 different Bluetooth modules, the RN42, the TI CC2540 and the Nordic nRF51822 modules. The CC2540 we talked about in the microcontroller section since it's both a MCU and a Bluetooth module all on one chip. The Nordic nRF51822 is similar to the TI CC2540, with a 32 bit ARM Cortex M0 processor, 256k flash memory and a huge 16 to 32 KB of RAM, making it more powerful than the CC2540,at the cost of less documentation and GPIO pins.

3.3.4.1 Bluetooth Module Overview

There are going to be 2 categories that we base the Bluetooth decision on are cost and power consumption. While we could have expanded our criteria into looking at things such as maximum effective distance, size of compatible antenna, etc., we decided that we could learn to live with any shortcomings of whatever chip we chose as the driving factor behind any choice was going to be cost and power no matter what.

Table 3: Bluetooth Module Overall Comparison

	RN42	TI CC2540	nRF51822
Cost	$18.95	$4.73	$4.62
Compatibility	UART	SPI/UART	SPI/UART
Bluetooth LE	No	Yes	Yes
Idle power consumption	26 µA	0.9 µA	2.6 µA
Active power consumption	45 mA	23.8 mA	8 mA

3.3.4.2 Bluetooth Module Cost

As with all our comparisons, cost plays an important role since we aim to deliver a low-cost solution, but sometimes picking the cheapest option is not the best, especially for system critical parts such as Bluetooth in our case. A product might be cheaper but lacking functionality that we would prefer such as Bluetooth low energy, or be lacking in quality control, leading to a shorter product life span, faulty connections which can be difficult to pinpoint, or even danger to the user, as they could possibly be fire hazards.

The RN42 is significantly more expensive than the TI and Nordic offerings, due to it being sold as a standalone solution that you connect to your project using GPIO pins vs the CC2540 and nRF51822 being SoC that we would need to solder onto the PCB, which is why we are considering the RN42 for these comparisons despite the fact that it lacks BLE, as having our coasters be non-functional due to an error with our PCB design trying to get multiple SoC to integrate successfully would be more expensive than just going with the tried and true solution of the RN42 in the long run. Table 3 shows that the CC2540 and nRF51822 are pretty much neck and neck, with the nRF51822 being slightly less expensive, and still offering the BLE that we seek to keep energy consumption low.

3.3.4.3 Bluetooth Module Power Consumption

Following a similar theme to the key attributes of the microcontroller evaluation, power consumption is a big concern when considering hardware for small embedded system projects like the smart coasters. A few key hardware changes could potentially mean the difference between a single charge barely lasting a work day and the charge lasting three or more days.

As seen in Table 3 the RN42 loses here as expected, since it isn’t Bluetooth low energy compliant. The connected and transferring power consumption were for the case where signal strength was -5dB, as we don’t expect to be pushing the range limit of Bluetooth under normal use. The nRF51822 has a clear edge here, managing to only use 1/3rd of the power of the CC2540, however the idle power consumption is quite a bit higher. The power usage over time would depend on how often we wanted to be transmitting data; if we had an application that needed to communicate constantly with the paired device, then the nRF51822 would be the best choice for power use. However, with our intended usage, it wouldn’t make a difference if we reported the status of the beverage 10,000 times a second or only 10 times a second, which is why the CC2540 would be the best choice for our purposes, since most of the time it will be in the idle stage and the time it takes to enter and exit idle power mode is miniscule.

3.3.5 Bluetooth Module Choice: TI CC2540

This is the Bluetooth module of choice due to its low cost, Bluetooth low energy functionality and ultra-low power consumption in idle mode. As discussed in section 3.1.4.2, the lower power consumption over the Nordic nRF51822 in idle mode is actually more impactful over an entire charge since we spend most of the time in idle mode during typical use. This, coupled with the fact that it was a strong competitor in the MCU comparison as well, cemented our choice to utilize this chip.

3.3.6 NFC Module Comparison

NFC will be used to essential give users an easy way to swap coasters between table hubs, by prompting the table hub to connect to the Bluetooth connection of the coaster whose tag it just read. Our group has decided to evaluate three different NFC solutions, the NXP PN532, the TI TRF7970A, and the NXP MFRC522, to determine what chip best fits our needs.

3.3.6.1 NFC Module Cost

Cost is once again a major consideration for all parts of the coaster.

The MFRC522 is slightly cheaper than the PN532 and TRF7970A according to Table 4. At time of first researching, it was actually listed for a huge discount at around $1.40 but the sale ended and its price jumped back in line with similar offerings. Now that it's not such a great price, it puts the other NFC modules on a more even playing field.

Table 4 NFC Module Overall Comparison

	PN532	TRF7970A	MFRC522
Cost	$4.80	$5.82	$4.14
Maximum read distance	50 mm	N/A *	50 mm
Supports SPI	Yes	Yes	Yes
Supports UART	Yes	No	Yes
Supports I^2C	Yes	No	Yes
Idle power consumption	2 µA	0.5 µA	5 µA
Alert power consumption	45 µA	120 µA	10 µA
Active power consumption	50 mA	70 mA	60 mA

3.3.6.2 NFC Module Maximum Read Distance

Read distance is another important metric to consider when evaluating NFC Modules. Too low of a read distance can, in the worst case, mean that the NFC capabilities don't function at all because it would be impossible to get the reader close enough to the tag, since our tag will be inside the coaster and have to travel at minimum the thickness of the case before it is able to be read. Even in not so extreme cases, it could ruin the user experience by requiring a lot of precision in getting the tag to be read by needing to have the coaster lined up in the perfect spot to get close enough to register, effectively failing at making the process smooth and easy.

*The TI TRF7970A had no maximum read distance listed in the data sheet, since this chip is designed to be socketed into a PCB where you design your own antenna. Another TI document about NFC antennas stated that a good rule of thumb is that your max read distance is double the antenna's diagonal measurement. Our group will assume that we will be able to achieve a 50 mm max read distance as well if we chose this part and designed our own antenna.

Having roughly the same size antenna, both the PN532 and MFRC522 predictably have the same max read distance. 50 mm is quite low for us since the signal must go through our entire outer shell of the coaster before it can read a tag, but it is too hard to tell if it will be too low at this stage of development. Since we will be building our own circuit board, there might be a way to design our own more powerful antenna to ensure that low read distance is not an issue.

3.3.6.3 NFC Module Supported Tags

Tag support is useful for future-proofing the coasters so that they can be used for a longer period without becoming a security concern.

Table 5: NFC Module Tag Support

NFC Module	ISO/IEC 14443A	ISO/IEC 14443B	ISO/IEC 15693 & ISO/IEC 18000-3	ISO/IEC 18092 & ECMA 340 P2P	FeliCa	NTAG
PN532	Yes	Yes (Read only)	No	Yes	Yes	No
TRF7970A	Yes	Yes	Yes	No	Yes	No
MFRC522	Yes	No	No	No	No	Yes

The only tag supported by all the NFC modules is the ISO/IEC 14443A, or MIFARE, has been proven to be able to be hacked and could present a security threat to the user if chosen. That leaves the MFRC522 with only NTAG as the other option, which leaves no other options if NTAG gets compromised as well. The PN532 and TRF7970A both have quite a few other options though, making them the better choice.

3.3.6.4 NFC Module Supported Interfaces

A variety of supported interfaces will allow the coaster to be more flexible with the other modules that are connected to the microcontroller.

The lack of support for the TRF7970A is concerning, as it limits any use of SPI that we might want to get out of our microcontroller of choice, the TI CC2540, in the future. SPI is by far one of the most popular interfaces we've come across for peripherals, so losing the functionality despite having extra pins that can take in more inputs would be bad for the flexibility of the device. However, there are ways to get around the lack of official ports being used for certain interfaces with a process called 'bit bashing'

Bit bashing essentially enables you to use software instead of dedicated hardware to define what certain pins do. For example, if we used our official SPI pins on the TRF7970A's communication line and had another SPI device we wanted to utilize in the future, we could assign generic ports on the device and program them to behave like SPI ports, with no hardware changes necessary. It's a bit of a hack and like any software emulated hardware action, it comes at a cost to performance, both in power usage, CPU utilization and speed, but it is better than not being able to add a sought-after feature to our project even if we have other pins available.

3.3.6.5 NFC Module Power Consumption

NFC is a relatively low power task compared to running the Bluetooth and the microprocessor, but any power saved will extend the battery life, which is always a good thing. In this section, our group will analyze both the hard power down, which requires an external input to awaken the module, and the soft power down, which keeps the module awake by default.

An interesting design consideration our group can take into account is having a button that hooks up to a communication port and can be used to wake up the NFC module from hard sleep to soft sleep, reducing the average power consumption of the NFC module by nearly 60x in the TRF7970 according to Table 4, at the tradeoff of increased case complexity and more openings where the case could fail to protect the internals.

Design considerations aside, there are good and bad things for each NFC module here. If our group went with no button to activate the NFC, we would have to use soft power down, which keeps the RF level detector on so when a tag comes near, the NFC reader can fire up and start transmitting. The best overall choice would be the MFRC522 in that case, due to its very low soft power down power draw, with the TRF7970A being 12x worse. However, if our group did choose to activate the NFC module with a button, then the TRF7970A would be worth considering due to its very low 0.5 µA power draw, with the MFRC522 now being 10x worse in this scenario. The PN532 is a jack of all trades, only slightly beating out the other modules in terms on transmitting power consumption, but under normal use cases that should not be occurring often enough to value it more highly than idle power consumption.

3.3.7 NFC Module Choice: PN532

The PN532 is the best fit for our project due to port availability, acceptable power performance, cost and a good selection of supported tag types. While not as effective as the TRF7970A in some areas, the fact that it has the I^2C port to allow us to use our selected microcontroller and Bluetooth module without the need for bit bashing to enable future SPI peripherals to work. While the price was nice, the MFRC522 lost to the PN532 based on tag selection and power usage.

3.4 Serial Communication Technology

Serial communication is the process of sending one bit of data at a time over a communication channel or bus. It is used in most computer networks and is a design aspect that needs to be considered when implementing the clever coasters. In this section, we will discuss the most common serial communication technologies, SPI and I^2C, which can be found in many development boards.

3.4.1 SPI

Serial Peripheral Interface Bus, or SPI as it is more commonly called, is a synchronous serial communication method developed by Motorola in the late eighties, and is used primarily in embedded systems, often being used in SD cards and LCD's. The SPI bus has 3 important wires for our uses:

- Master In Slave Out: a unidirectional signal that lets the slave send data to the master. Since we will be mostly monitoring sensor data, this is the signal that will be used the most often.

- Master Out Slave In: The MOSI is another unidirectional signal going the opposite direction of the MISO signal, letting the master communicate with the slave peripheral.

- Serial Clock: The clock that synchronizes all the signals, generated by the master peripheral.

There are two other signals used often in SPI communication. These signals are the Serial Data I/O, a bidirectional I/O, and Slave Select, which outputs from the master. Our device won't be using the Slave Select signals because we don't plan to have multiple slaves connected to one master, and our selected NFC chip, which can communicate over the SPI protocol, doesn't utilize the SDIO capabilities.

SPI signal transmission sends out 8-bits of data at a time from a shift register, located both in the master peripheral and the slave peripheral. The shift register in the master is swapped with the shift register in the slave, with the most significant bit from one being replaced with the least significant bit from the other, and repeated for as many clock cycles as it takes until the shift registers have swapped values completely. If using multiple slave devices, after transmission has ended, the slave is released from the SS (Slave Select) line. Since SPI is a synchronous protocol, the transmission is synced up with the SCLK, which has 4 defined modes for communication as follows:

- Mode 0: Clock Polarity (CPOL) = 0, Clock Phase (CPHA) = 0. This means that the SCLK is active high and data is captured on the rising edge.

- Mode 1: CPOL = 1, CPHA = 0. SCLK is active high and data is captured on the falling edge
- Mode 2: CPOL = 0, CPHA = 1. SCLK is active low and data is captured on the rising edge.
- Mode 3: CPOL = 1, CPHA = 1. SCLK is active low and data is captured on the falling edge.

Though we will not be using multiple slaves within the core functionality of our coasters, if we add expanded functionality that communicates over the SPI protocol, we will have to look at using multiple slaves. There are two approaches to using multiple slaves, the independent slave select where each slave has its own select line leading from the master, and the daisy-chained slave select where there is a single slave select line that branches to each slave.

Figure 8 details the daisy-chained slave select, which utilizes only one slave select line from the master, which then propagates throughout the other slave devices via the MISO of the current slave being used to communicate with the MOSI of the next slave, with all slaves receiving the same slave select signal. This approach saves pins on the microcontroller, at the cost of less security because slaves can communicate with one another and pass potentially sensitive data through each other.

According to Figure 7 each slave device, colored green, blue and purple all have their own slave select line attached to them, via the SS1, SS2 and SS3 ports on the master respectively. The MISO pins are all tied together, which means they need to be tri-state pins. This method uses more pins on the microcontroller however, which as stated in our microcontroller research section, is a big concern.

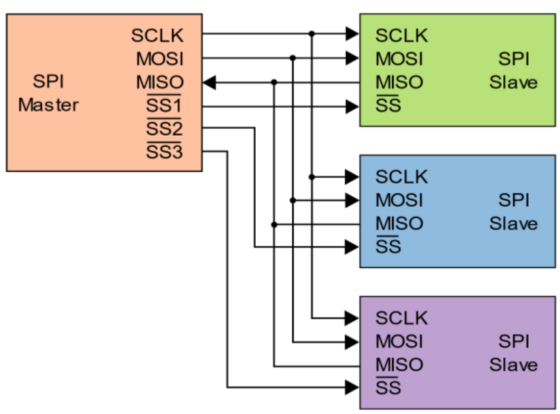


Figure 7 : Independent slave select. The use of this image is governed by the GFDL.

Figure 8 details the daisy-chained slave select, which utilizes only one slave select line from the master, which then propagates throughout the other slave devices via the MISO of the current slave being used to communicate with the MOSI of the next slave, with all slaves receiving the same slave select signal. This approach saves pins on the microcontroller, at the cost of less security because slaves can communicate with one another and pass potentially sensitive data through each other.

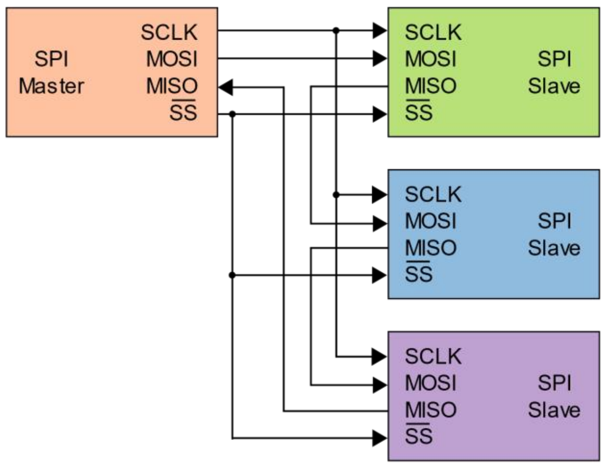


Figure 8: Daisy Chained Slave Select. The use of this image is governed by the GFDL.

3.4.2 I^{2}C

I^{2}C, or Inter-Intergrated Circuit, is communication protocol developed by Philips Semiconductor, now called NXP Semiconductors, in 1982, making it older than SPI protocol. As of 2006, its protocol is available to the general public free of charge. It is typically used for low speed communication between different integrated circuits.

I^{2}C has a master-slave architecture similar to SPI as well, with multiple speed modes, with speeds of 3.4 Mbps in high speed mode, 400 Kbps fast mode, 100 Kbps standard mode and 10 Kbps slow mode. It uses a 7-bit address bus vs SPI's 8 bit, and contains 2 signals vs SPI's 4. These two signals, the serial data (SDA) and serial clock (SCL) both utilize pull up resistors. The SDA signal is bidirectional, allowing data to be transferred between master and slave(s). The SCL signal is unique in that the clock signal generated by the master may be commanded by a slave to lower frequency to allow extra processing time or to slow down the flow of data.

I^{2}C classifies connected devices into either a master or slave role and each role dictates the characteristics of the device and what it is capable of doing. The masters and slaves are classified as nodes, where a master node is the node that sets the clock signal and begins transmitting to the slaves, the microcontroller in our case. The slave node is simply the node that acknowledges the clock and receives data transmitted from the master. I^2C can supported up to 1008 slave nodes, and can swap around what node is considered a master or slave in between data transmission, however a master node cannot communicate with another master node simultaneously. As with SPI there are four modes of communication for I^2C:

- Master transmission: Master node is sending data to a slave device
- Slave transmission: Slave node sends data to master device
- Master reception: Master node receives data from a slave device
- Slave reception: The slave node receives data from a master device

I^2C basic messages start and end with a START and STOP bit, and when a message is received by the slave, the slave will send an ACK bit to acknowledge that the message made it.

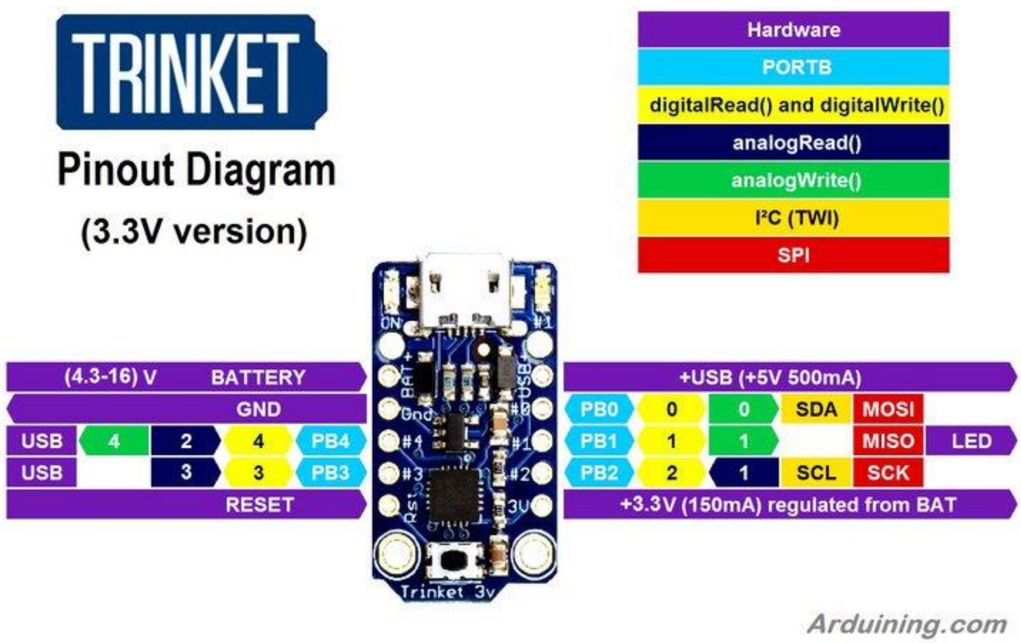


Figure 9: Pinout Diagram for the Trinket 3.3V

I^2C commonly uses 3.3 volts as an operating voltage, perfect for our projected operating voltage of the system, meaning we wouldn't require additional voltage regulators to safely use this protocol. This coupled with the fewer number of pins it would take up, 2 GPIO pins vs 4 for SPI mean that I^2C would be the preferred choice when considering our options. As shown in Figure 9, with a limited number of pins, the difference between using I^2C and SPI can have a huge impact on the long-term flexibility of the project, in this case saving an extra pin to use as a read/write or led output.

3.5 Device Power

Powering our system will be a very important requirement that needs to be looked at very carefully. Restaurant owners have electricity bills to pay, which can add up when implementing a technological system such as the clever coasters. Powering the coasters should only utilize what it needs and be self-containing. In this section, we will explore the requirements and considerations for power, as well as charging designs to place within the coasters themselves.

3.5.1 Requirements and Considerations

The Clever Coasters are the main devices under design consideration. As per the marketing requirements, they are wireless to be convenient for restaurant customers and waiters who will need to move them around, take them from table to table, or move them out of the way of plates and other things that may be placed on top of the table. Their wireless nature, and the fact that they have function beyond their use as a simple coaster, necessitates that they are also powered wirelessly. This means that they will have a battery inside them that will power the microcontroller, wireless communications chips, the PCB, sensors, and all of the included LEDs. The battery should have enough capacity to last a reasonable amount of time while the coaster is in operation and be able to output enough current to power all the components.

It would be extremely inconvenient and annoying if a coaster stopped working in the middle of a patron's meal. The service quality might decrease and, where the purpose of the coasters was to make waiters more efficient and effective at their job, waiters would now have to find these out-of-power devices and replace them with ones that have sufficient charge. All of this would make the restaurant and the service provided look worse than if the Clever Coasters weren't being used at all.

Optimally, the batteries should last all through an entire day's service so that waiters would have to spend no time during operating hours managing the recharging or replacing the batteries of the coasters. Another marketing requirement is that the Clever Coasters are easy to recharge. This could mean replacing batteries, or plugging in the coasters, or placing them on some kind of charging dock. The following is an exploration of the benefits and drawbacks of these three methods of maintaining the coasters in operation.

3.5.2 Charging Method

Four different methods were researched and considered for Clever Coasters. Using replaceable batteries (which is not really a charging method), wired charging, a charging dock, and induction based charging. The following sections are a discussion of the four options and their advantages and limitations as pertaining to the Clever Coasters application.

3.5.2.1 Replaceable Batteries

Though not truly a way to recharge the Clever Coasters, using replaceable batteries is the simplest solution from an engineering design perspective, but it comes with a number of drawbacks. First, the restaurants would need to have a large number of batteries in stock that they would use to replace the batteries in the coasters that run out of power. As a corollary to that, the restaurants would need to properly dispose of used batteries, which adds another burden to them. These things are highly undesirable because they would increase operating costs for restaurants, require a special place to store new and used batteries, and require the restaurant to spend time and resources on the procurement of new batteries and on the disposal of used ones.

Additionally, waiters would either have to spend time keeping track of coasters that should be approaching the end of their battery life or risk the chance that a coaster runs out of power in the middle of service and the embarrassment that would cause to them and the restaurant. Replacing the batteries could also be a tedious process for the waiters, though it could be outweighed by the potential benefit of having to do it very rarely. Depending on the charge held by the disposable batteries, it is possible that waiters would have to do less work over all than with rechargeable ones.

Another consideration with disposable, replaceable batteries is that the opening through which they are replaced would need to have water proofing when closed. This is to ensure that the internal circuitry does not get exposed to liquid if it were to drip or get splashed onto the coasters during the normal operation of the restaurant. If the internal circuitry gets exposed to liquids, it could short circuit and/or corrode which would damage the coaster and cause it to stop operating properly. This would be costly to the restaurant as the coaster would need to be replaced.

As a safety consideration, it is also important to note that a short circuit on a battery can be very dangerous and result in fires and explosions. If property or people were harmed in such an incident, the restaurant would be liable and it is doubtful that a restaurant would want to risk the increased liability considering all the other drawbacks of having replaceable batteries in the coasters.

3.5.2.2 Plug-In Charging

Plugging in the Clever Coasters, on the other hand, would involve the use of rechargeable batteries. These would be permanently housed within the coaster and sealed from the outside environment, including any spilled beverages near the coasters. The plug-in charging method would mean that each coaster has some kind of port to which a powered wire can be attached that will allow it to charge.

The key benefit of this method is that it is safe to charge devices this way and most people have experience plugging in devices to charge them.

One of the main drawbacks is the extra work this creates for the waiters. After every service, the waiters would not only have to collect all of the coasters from every table, which is not that much of a problem, but they would have to plug each one in individually with a wire attached to a central charger. Depending on the size of the restaurant, this could result in a very large device with a huge mess of wires that have to be plugged in and then unplugged for every single coaster. This is very inconvenient and time-consuming for the waiters. The wires also have the potential to wear out over time and require replacement.

This method does not scale well with the size of a restaurant because there will either have to be a larger charging station with more ports to plug wires into or a restaurant will have to buy many of them to be able to charge all of their Clever Coasters. Either way, the waiters will have to do much more work to manage the wiring and ensure all coasters are charged.

From a design perspective, the coasters will have to have a charging circuit that limits voltage and current so that their batteries get charged safely and efficiently. This requires more components and complexity in the PCB design compared to replaceable batteries. The plug will also have to be water resistant enough to handle getting wet so that it could protect the components inside the case and not short out.

3.5.2.3 Charging Dock

Just like plug-in charging, a charging dock solution also requires the PCB design to accommodate a charging circuit within each Clever Coaster. In that regard, the complexity remains the same. It does, however, offer other benefits.

Specifically, a charging dock could be very convenient for waiters to use. If designed with convenience in mind, the charging dock could accommodate many coasters, potentially as many as one could reasonably fit in the space. The dock would not have to be different or have more wires like the plug-in charging option for more coasters. Instead, it could allow waiters to stack the coasters on top of one another and place the stack on the charger. This is convenient for wait staff because it doesn’t require any fiddling with wires or cases or batteries. All a waiter has to do is put all of the coasters in the same place.

This does, however, require serious design considerations for the coasters and the charging dock. First, the coasters must not only have a charging circuit within them that controls voltage and current going to the battery, but they must also be trivial to stack, not requiring too much attention to their orientation or else the time savings and convenience disappear. Therefore, they must be designed to charge no matter their orientation in the charging dock and how they’re placed on top of each other, within reasonable constraints. If the coasters naturally stack with each other and are of a symmetric shape, which they should be, then they should charge while stacked.

Second, the coasters must be able to transfer power through the stack, so there must be some kind of exposed connection points, or terminals, on the coasters that allow current to flow. There are safety considerations here. The terminals cannot be connected directly to the battery because it should be impossible to short circuit the coasters during normal operation in the restaurant. The terminals might be required to allow substantial current to pass through them without harming the coasters because the coaster closest to the charging dock will have to allow the current that will go into every other coaster to pass through its terminals.

Third, the charging dock itself will have to have exposed terminals and allow coasters to stack onto it. These terminals will have to be at the correct voltage for the batteries in the coasters to charge. Furthermore, the charging dock must be current limited so that even if it is short circuited, it does not cause a safety issue. This is a potential issue because if the current limit is too low, the coasters will charge very slowly and slower the more of them that are stacked on top. If it is too high, it can have the potential to be dangerous and heat something that falls on it and causes a short which could lead to a fire hazard.

3.5.2.4 Induction Charging

Another option that is similar to the charging dock, but involves no exposed terminals is induction charging. Induction charging works by having a coil in the charger and a coil in the device being charged. The charger converts normal wall AC power to high frequency alternating current and sends it to the transmission coil. An alternating current in the transmission coil creates a rapidly changing magnetic field around it. This induces a current within the receiving coil of the device. This current is then converted to DC and used to charge the battery within the device. [16]

The key benefit of this method is that it is very easy to use, just like the stackable charging dock. All that needs to be done is for the waiters to collect the coasters and place them onto the induction charging station.

Another benefit is that it is safe from electrical shock. There are no exposed terminals so there is no potential for a short circuit. A new safety consideration arises, however, as the power output of the inductance charger increases. The charger will leak flux around it. This may be a health hazard for people constantly near the device or those with medical devices like pacemakers. It could also cause nearby metal objects to heat up, creating another hazard.

Furthermore, the two main designs of this type of charging dock offer different advantages and drawbacks and neither seems to completely fit the requirements of this system. There is a coil array and a perimeter coil geometry. [16]

The coil array is several small coils arranged close to each other inside a flat surface. They can selectively turn on and off by sensing that a device that needs charging is nearby. It provides high efficiency and very little leaked flux while being able to charge many devices at once because they are all over different coils. Unfortunately, this type of geometry is not very good for the coaster system because it would require a massive flat area upon which to place all of the coasters for charging.

The perimeter coil geometry allows for a single large coil that fills a volume with electromagnetic flux and allows a device to be charged when it is anywhere within that region. This could be implemented as a single large station in which the Clever Coasters can be stacked by the waiters, or it could be added to every table so that the coasters would get constantly charged without the need to do anything at all. This would be the most convenient option for the waiters, since they would never need to move coasters from a table except maybe to clean them. On the other hand, this type of geometry is also very inefficient and not good at charging multiple devices. It results in a lot of leaked and unused flux. Therefore, it would waste lots of power and charge the coasters slowly. Putting it in every table would also be far more costly than having some kind of centralized charging station. In addition, the flux created by the tables could interfere with electronic devices like phones and the ability of the coasters to send wireless messages.

Overall, the induction charging method requires every coaster to have a coil and a circuit to convert from high frequency alternating current to direct current. While it does provide a high level of convenience for the employees, it has drawbacks related to safety and efficiency which may be too costly to be part of this system. To overcome these obstacles, it requires a more complex design.

3.5.2.5 Comparison of Charging Solutions

Table 6 compares the properties of the four charging solutions that were considered for the Clever Coasters application. In this table, the top row is the list of the charging solutions. The left column is a list of properties that were considered when choosing the solution. Red boxes indicate properties that are very important for the decision, yellow ones are somewhat important, and green ones are least important.

In the body of the table, green boxes indicate that the charging solution is good with respect to the property in the left column, yellow indicate that it is decent, and red indicate that it is a poor choice. These are all fairly subjective measures because the properties considered are, themselves, mostly subjective.

3.5.2.6 Most Important Properties of Charging Solutions

The first of the two most important considerations of the charging solutions is the reliability of the charge. As discussed previously, it is undesirable to have the coasters run out of power in the middle of a service. Replaceable batteries, by their very nature, would keep working until they ran out and then would have to be replaced by an employee. This is very unreliable. The wired and charging dock options both offer a high reliability if the coasters are set to charge every day. The induction charging method might be sufficient depending on how slowly it charges and how much of the charge is depleted over the course of one day of operations.

The second very important property is the long-term cost. This is already a very technical and somewhat expensive solution, so additional, unnecessary costs should be avoided. The wired and charging dock solutions would have very low long term cost because they are efficient and there is little to no cost related to replacement. The induction charging option doesn't have any costs related to replacing anything, but it is inefficient so it would lead to a higher electricity bill for a restaurant. The replaceable battery solution is potentially the worst because batteries would have to be frequently obtained, replaced, and disposed of.

On top of that, if a solution were implemented to increase reliability (the first important property) by indicating that the batteries are close to discharged, it would further increase costs because batteries would be replaced before completely running out of energy.

Based on these two important aspects of the charging solutions, we can rank the four options from best to worst. Tied for best are the charging dock and the plug-in charging. Following them is induction charging for which the main drawback is energy cost. Last is the replaceable battery option. This one is so bad it is not necessary to continue considering it.

3.5.2.7 Moderately Important Properties of Charging Solutions

Several other aspects are still significant for consideration. The first is the ease of use and the added work for the employees. The purpose of the coasters is to assist the waiters and make their jobs easier, not to increase their workload. As such, the charging dock and induction charging options do the least to increase the workload. Plug-in charging is somewhat inconvenient and increases work moderately.

As far as safety goes, the primary concerns are related to shorting out the circuits for all but the induction charging method. This could cause damage to the batteries or the electronics in the coasters for the plug-in chargers and the replacement battery option. For the charging dock, the short circuit danger is in the dock itself, not in the coaster. If something metal were to fall into the charger dock and come in contact with the active terminals, it could create a short that causes the metal to heat up. This will need to be taken care of by ensuring that the charger is current limited. Safety for the induction charger is related to other electronics that could come near the charging station along with metals that could heat up, just like with the charging dock.

The last moderately important property is scalability. This is significant because some of the problems and difficulties discussed above, like the additional work for the employees, are exacerbated by increasing the number of coasters.

Table 6: Comparison of Charging Solutions

	Replaceable Batteries	Plug-in Charging	Charging Dock	Induction Charging
Added work for Employees	Check and replace batteries for dead/low coasters	Collect and plug in coasters	Collect and stack coasters	Collect and stack coasters
Reliability of Charge	Low (could run out at any time)	High (charged every day)	High (charged every day)	Medium (depending on implementation, slow charge)
Safety	Liquids or improper insertion of batteries could cause short circuit	Liquids could cause short circuit	Dock could be short circuited	EM fields could affect people and electronics
Added Complexity to Coasters	Must make battery compartment accessible and waterproof	Must include charging circuit in coaster and waterproof plug	Must include charging circuit in coaster and have exposed terminals	Must include charging circuit in coaster, special coil, and HFAC->DC converter
Complexity of Charger Design	None	Must support large number of charging cables	Must transfer power to many coasters at once and be current limited	Must detect when coasters in charging range, must convert wall AC to HFAC, must consider coil geometry
Long Term Cost	Frequent replacement and disposal of batteries	Electricity for charging, occasional replacement of wires	Electricity for charging	High electricity cost due to inefficiency
Scalability	Poor	Poor-Medium	High	Medium-High

The charging dock and induction charger have the best scalability because they offer the easiest to use system. The induction charger, however, may not be as scalable when it comes to energy efficiency because the power will be distributed among more coasters. This can decrease the reliability of ensuring a full charge. The plug-in charger becomes more annoying to deal with and replaceable batteries are also very bad.

Adding the above properties to the consideration, the charging dock becomes the best solution, followed by the induction charger, and then the plug-in system. It is still worthwhile to mention the least important properties and why they aren't as critical to the Clever Coasters application.

3.5.2.8 Least Important Properties of Charging Solutions

The two properties that were judged to be least important are related to the added complexity of the design. Even though these things make the design process more difficult, they ultimately have little impact on the customer if they don't affect the other properties like cost.

The added complexity to the coaster design comes primarily from having to include the charging circuit within the coaster itself for all but the replaceable battery option. Even though this means additional hardware, this circuit should not be very large and is required for each of the good options. The induction charger has even more added complexity because of the requirement of a large coil and circuit to convert from high frequency alternating current to direct current. That would increase the costs of each coaster, which is undesirable.

The added complexity to the charging station comes from designing it for safety, convenience, and ability to charge many coasters. Once again, there is little complexity in the replaceable battery solution, but this doesn't matter because it is not a good option for more important reasons. The plug-in and dock charging options both must be designed to scale with additional coasters. The induction charger is the most complex to design because it needs to have a large coil (or many small coils) to create the electromagnetic fields that will charge the coasters. That solution also suffers from a need to detect if coasters are placed on it because otherwise it is constantly wasting power into the air if it is turned on and not charging anything.

3.5.2.9 Charging Solution Choice (Dock)

Considering all the options discussed above, their advantages and limitations lead us to choose the charging dock as the best solution for Clever Casters. Most importantly, the charging dock is cost effective and will provide a reliable charge to the coasters. It is also very easy to use for the employees, which is a marketing requirement for the system. Finally, it is scalable both in its ease of use and in its power consumption. It is convenient and efficient. The minor issues that do arise from it will have to be taken into consideration when designing the dock, but the comparative benefits outweigh the drawbacks.

3.5.3 Battery Types

Since the coasters will have batteries to power them, it is important to consider the different types of batteries that may be used in them. The most common consumer batteries are alkaline, lead acid, nickel cadmium, nickel metal hydride, lithium-ion, and lithium-polymer.

3.5.3.1 Disposable Alkaline Batteries

Alkaline batteries would be used if the choice were made to use replaceable batteries. They are quite inexpensive at first, but their costs increase with time. Additionally, they take up a significant amount of space and they have a standard voltage of 1.5V which means multiple cells (potentially three or four depending on the microcontroller used) would be required since many microcontrollers need 3V to 5V. The Clever Coasters would have to be large to accommodate the use of these types of batteries which is not realistic.

3.5.3.2 Reusable Alkaline Batteries

Reusable alkaline batteries are very similar to their disposable version with respect to their size and cell voltage, but it would be possible to recharge them. In theory, this makes them better than their disposable cousins, but in practice they have other disadvantages that impose limitations on their use. They have a comparatively high internal resistance. They have very few useful life cycles after which their capacity drops far more than other battery types. They also have a low acceptable discharge current. One benefit of alkaline batteries is that they have the lowest self-discharge, but that is not as important for the Clever Coasters application as the other aspects which are limitations.

3.5.3.3 Lead Acid Batteries

Lead acid batteries are common in automotive applications and other high energy and industrial applications. They are very inexpensive and have existed for a long time. They don’t lose much charge though self-discharge, though this isn’t very important for Clever Coasters because the idea is that they would be recharged almost every day. They don’t have memory, which is a very good feature, and are more than capable of decent discharge rates. Unfortunately, they suffer from low energy density because of their high weight on account of the lead. Lead is also an environmentally harmful chemical. Moreover, lead acid batteries can release gasses when charging and discharging and may need to be maintained by refilling them with water periodically. Several of these issues make it a poor choice for mobile applications.

3.5.3.4 Nickel Cadmium Batteries

Nickel Cadmium, or NiCd, batteries are capable of producing high current, undergoing many charge-discharge cycles, and recharging quickly. This comes at a cost, however, in that these batteries are heavy for their capacity so they have low energy density. They also form memory if they are not constantly being discharged completely. Memory is the tendency of a battery to lose its ability to discharge completely if it is not always discharged to near empty. This poses a problem for the coasters because if they are constantly recharged before they run out of power, they will quickly lose much of their capacity. Furthermore, these batteries require maintenance every one or two months and are made of toxic metals that are harmful to the environment. All of the best features of NiCd batteries are unnecessary for the purposes of the Clever Coasters so the disadvantages outweigh the advantages.

3.5.3.5 Nickel Metal Hydride Batteries

Nickel Metal Hydride, or NiMH, batteries have higher capacity for the same weight, have higher self-discharge rates, and cost more than NiCd. Unfortunately, they still require regular maintenance, though not as often as NiCd. These batteries also prefer to operate at much lower discharge current which is fine for the Clever Coasters application. They are not as environmentally detrimental either. They require careful charging because they generate a substantial amount of heat while being charged and require more time to charge than NiCd. The benefits of NiMH do not seem to outweigh their disadvantages, especially when compared to the advantage of Lithium based batteries.

3.5.3.6 Lithium-Ion Batteries

Lithium Ion, or Li-ion, batteries have high energy density, high capacity, and have a fairly flat discharge curve which means that they maintain roughly the same voltage while their capacity goes down. The biggest advantages of Li-ion batteries are that they have no memory effects and require absolutely no cycling for maintenance. They also have low self-discharge, though that is not paramount for the Clever Coasters application. They do have some drawbacks since they require a voltage and current limiting protection circuit (usually built into the battery pack). They also lose capacity as they get older. Their discharge current is more than high enough for mobile applications like Clever Coasters. The biggest limitation, in this case, is that thin Li-ion batteries are expensive when compared to other batteries of the same capacity.

3.5.3.7 Lithium-Ion Polymer Batteries

Lithium-Ion Polymer, or LiPo, batteries are very like Li-ion in all characteristics except form factor. LiPo uses a solid/gel electrolyte which makes them easier to manufacture into different shapes. This also means they don’t have a metal shell which reduces weight. They are less prone to overcharge damage than Li-ion batteries. On the other hand, they have slightly lower number of life-cycles before their capacity decreases. These seem to be the best for the Clever Coasters application because of their moderate discharge rate, their slim form factor, and no maintenance requirement while still having a fairly fast recharge time and a good lifetime.

3.5.3.8 Direct Comparison of Battery Types

The following table visually compares various characteristics of the battery types discussed above. The top row is the list of all of the rechargeable battery types that have been considered. The left column is a list of properties of the batteries. The color scheme for the left column indicates the level of importance of the property. Green properties are those that don't matter very much for the Clever Coasters application. Red properties are those that are very important or even critical for this application. Yellow are those that matter, but may have a wide range of acceptable values. In the body of the table, green squares indicate good values, yellow squares indicate decent values, and red squares indicate bad values.

If a square in the left column is red (critical importance), then batteries that have a green square in the body of the table are likely to be good candidates to be selected for the Clever Coasters applications, yellow may be fine depending on the exact numbers, and red indicates a serious problem that needs to be overcome if possible and if it isn't possible those battery types are likely to not be a good candidate. Still, overcoming them may be costly or difficult.

If a square in the left column is yellow, then the values and colors in the body may influence the decision towards or away from a certain battery type, but it is more likely to be a tie-breaker than it is to be a primary determinant. If a square in the left column is green, then it is very unlikely to affect the final decision unless the value is exceptionally good or bad.

3.5.3.9 Most Important Properties of Battery Types

First we will look at the most important aspects that will affect the decision. The energy density of the battery is a very important because of the limited size of the coasters. This means that Li-ion and LiPo batteries are the best choices out of this set. Alkaline, Lead Acid, and NiCd are bad in this respect. NiMH has a range that could be bad or good or in between.

Overcharge tolerance is another critical property of the batteries and can greatly affect the design of the charging solution. The more sensitive a battery is to being overcharged, the more thought and care has to be put in connecting it to a charging circuit. Lead Acid batteries have the best tolerance and lead to the least design complexity. Alkaline and NiCd batteries have moderate tolerance so some kind of cutoff circuit would probably be a good design decision to implement. NiMH and the lithium-based batteries are the most sensitive to being overcharged and may lead to a fire hazard or even explosion of mishandled. This complicates the design of their charging and may require a very carefully calibrated charging circuit to ensure that they are only charged to the right voltage.

Nominal cell voltage is also important for the same reason as energy density. Because a particular voltage is required to run the microcontrollers, low cell voltage means either that multiple cells will have to be used in series or that the choice of microcontrollers will be severely limited. The space that multiple cells take up is significantly more than a single cell because when batteries are put in series the total capacity of the battery pack is not the sum of the capacities, but equal to the lowest capacity. On top of that, charging a multi-cell solution is much more complicated and requires a more complex circuit. This increases costs, design time, and the number of points of failure. Only the lithium-based batteries are good in this respect. Depending on the choice of microcontroller, lead acid batteries might be okay. Alkaline or nickel-based batteries are a bad choice because they will require multiple cells.

Batteries that require maintenance are not a good choice for this application either because the coasters should be self-contained and easy to use. Depending on the type of maintenance, it might be possible to take care of it with an intelligent charger, but not always. Nickel-based batteries may need to be forced to undergo a deep discharge and recharge cycle to ensure that they do not develop a memory. This increases the complexity of operating the Clever Coasters and would not be a good design decision. Lead Acid batteries, on the other hand, release gasses when charging and discharging sometimes. This leads to them needing to be refilled with water. This doesn't work for Clever Coasters.

Based on the above four metrics, several types of battery are consistently better for Clever Coasters and others are consistently worse. In order from best to worst we currently have the lithium-based batteries, Alkaline, Lead (below alkaline mostly because of the maintenance requirement), NiMH, and NiCd. The nickel-based batteries have so many limitations that it doesn't make sense to even consider them anymore.

3.5.3.10 Moderately Important Properties of Battery Types

The properties marked in yellow will help determine which of the remaining batteries have the edge in the Clever Coasters application. Internal resistance matters to some extent because higher internal resistance leads to more of the battery's energy being dissipated as heat instead of being used to power the microcontroller and other components of the coasters. A decent portion of the internal resistance of some of the batteries (like the lithium-based ones) comes from a protection circuit that ensures they don't output more current than they are rated for if accidentally short circuited. Only the alkaline batteries have high internal resistance. Lead Acid has very low internal resistance. The nickel and lithium-based batteries have moderate levels of internal resistance.

Cycle life is the property that determines how many discharge and recharge cycles the battery can undergo before it loses part of its capacity. This is somewhat important because one marketing requirement is that the coasters operate through a full day of restaurant work. Depending on the initial capacity of the battery and what it drops down to after a few hundred cycles, they may not be able to function all day. This would mean they have to be replaced, which is an undesirable outcome. If, on the other hand, the batteries are able to function fine for a full day even at 80% capacity, this property is less relevant.

Table 7: Battery Type Comparison [17]

	Reusable Alkaline	Lead Acid	NiCd	NiMH	Li-ion	LiPo
Gravimetric Energy Density (Wh/kg)	80	30-50	45-80	60-120	110-160	100-130
Internal Resistance (mΩ, including protection circuits)	200 to 2000 6V pack	<100 12V pack	100 to 200 6V pack	200 to 300 6V pack	150 to 250 7.2V pack	200 to 300 7.2V pack
Cycle Life (to 80% of initial capacity)	50 (to 50%)	200 to 300	1500	300 to 500	500 to 1000	300 to 500
Fast Charge Time (hrs)	2 to 3	8 to 16	1 (typical)	2 to 4	2 to 4	2 to 4
Overcharge Tolerance	Moderate	High	Moderate	Low	Very low	Low
Self-discharge / Month (room temperature)	0.3%	5%	20%	30%	10%	~10%
Cell Voltage (nominal)	1.5V	2V	1.25V	1.25V	3.6V	3.6V
Load Current (peak)	0.5C	5C	20C	5C	>2C	>2C
Load Current (best)	0.2C or lower	0.2C	1C	0.5C or lower	1C or lower	1C or lower
Operating Temperature (discharge only)	0 to 65°C	-20 to 60°C	-40 to 60°C	-20 to 60°C	-20 to 60°C	0 to 60°C
Maintenance Requirement	Not req.	3 to 6 months	30 to 60 days	60 to 90 days	Not req.	Not req.

This property is also not a critical property because the batteries will be charged and discharged rather slowly in the Clever Coasters application which leads to more cycles than rapid charging and discharging. Of the batteries still under consideration, only Li-ion have a high cycle life. They are followed by LiPo which are followed by lead acid batteries. Rechargeable alkaline batteries have by far the worst performance here with only about 50 cycles which lead to a 50% decrease in capacity, far worse than the 80% decrease after a few hundred cycles for the other batteries.

Fast charge time is not very important except to ensure that the batteries can recharge during the off hours of the restaurant. All but the lead acid batteries are capable of recharging in under four hours which should be acceptable for all restaurants. Lead acid can take 8 to 16 hours to recharge which may work for some restaurants, but many restaurants are open for most of the day. This means that lead acid batteries might not work well for Clever Coasters.

The best load current under which the battery can operate compared to what the battery actually operates at affects how quickly the battery loses capacity. It matters because operating under this current is better for the cycle life of the battery. In all likelihood, however, the microcontroller, wireless module, and other components will drain less than 0.1C from the battery so all of the batteries would be acceptable for Clever Coasters. Still, it's better to have a larger than a smaller range to maneuver in for the design. Hence, the lithium and nickel-based batteries score higher here than alkaline and lead acid.

Ranking the batteries based on the moderately important metrics results in an even greater discrepancy. Now the order from best to worst is Li-ion, LiPo, Alkaline/Lead. At this point, however, alkaline and lead acid batteries simply don't compare to the batteries based on lithium. Therefore, the choice is really just between Li-ion and LiPo. An analysis of the three remaining properties of these batteries is largely unnecessary except to explain why they are unimportant in this application and under what circumstances they would be more important.

3.5.3.11 Unimportant Properties of Battery Types

The final three properties are not very important for Clever Coasters. The first is the self-discharge rate. This is important in situations where they batteries are left unused for long periods of time. It is measured in percentage of capacity per month. The intended use of these batteries has them being charged and discharged every day so this property is largely irrelevant. Still, the nickel-based batteries have the worst self-discharge, lithium batteries are in the middle, and alkaline and lead acid have very low self-discharge.

The second property is the peak load current. The coasters will never really utilize a high peak current so this property is even less important than the best load current that was previously discussed. It would be important in quadcopters, RC, and automotive applications or for a powerful camera flash. NiCd batteries tend to have the highest peak load current capabilities, but the other batteries are all fairly decent at it as well. Only the rechargeable alkaline batteries don't have high discharge rates.

The final property that is fairly unimportant is operating temperature. That said, this would be more critical if the batteries did not all have such wide operating temperatures. For drink coasters in an indoor restaurant, the worst range (0 to 60°C) is far more than enough. Even if a restaurant has outdoor tables and puts these coasters on them, it is a property that is unlikely to affect them. In the first place, people tend not to dine outside in sub-freezing temperatures. Secondly, the battery will be contained in a plastic shell which will offer some insulation. Combined with the heat produced by the battery itself, it should easily stay within operating temperature, even if the outside air is somewhat colder than the freezing point.

3.5.3.12 Battery Type Choice (LiPo)

Having considered all of the above properties, only two battery types were left as reasonable choices: Li-ion and LiPo. They are almost identical in all relevant aspects except cycle life and overcharge tolerance. As discussed earlier, cycle life is less important because the batteries will not be used at anywhere near their maximum discharge rate so they will likely have higher than expected lifetime. Overcharge tolerance is a very important property that LiPo batteries are bad at, but Li-ion batteries are worse. This means that there is less room for error when charging the batteries and increases the difficulty of the design. It is better to have a higher overcharge tolerance, even if only slightly, because it reduces the risk of damage to the battery cell and the risk of fire or an explosion. A final factor, which wasn't part of the table, that makes LiPo better for Clever Coasters is its form factor. While Li-ion batteries are housed in a solid metal casing, which limits their shape and makes them more bulky than they need to be, LiPo batteries use a dry/gel electrolyte that allows them to be thin and very compact. This is great for our application because they have to fit inside the coaster. Taking all of that into account, LiPo batteries were chosen for the Coasters.

3.5.3.13 Specific Battery Choice

Having made the choice of Lithium-Ion Polymer for the battery to power the Clever Coasters, we must choose a specific battery to go into them. We know that we will use a since cell battery, because we will only need the 3.7V that comes from one cell to power the microcontroller, wireless modules, and all of the peripheral electronics. The key properties of LiPo batteries that make them different from each other are their capacity, their continuous discharge rate, their physical dimensions, and their price.

For the Clever Coasters application, a higher capacity is better. Although the microcontrollers and wireless modules are being specifically picked to minimize their power consumption, it is better to have a margin of safety. This margin of safety will be useful if a restaurant forgets to charge the coasters, if the batteries start losing capacity with age, and if the microcontroller and wireless modules use more current than expected when in operation. Of course, there is a limit to what is required here since the assumption is made that the coasters will be charged every day. Also, based on the power consumption of the wireless modules, microcontrollers, and NFC, the highest continuous current drain is around 50mA when constantly transmitting, but it will likely be much less during normal operation. As such, a 500mAh battery would have enough charge to power the devices at least 10 hours.

The continuous discharge rate is largely going to be unimportant for this application because the maximum current drain will be around 100-150mA when connecting with NFC. This means that if the batteries have at least 150mA of maximum current drain, they will be sufficient. Most LiPo batteries are able to provide much more current than that.

The physical dimensions of the battery matter because the batteries will be placed inside the Clever Coasters. The bulkier the battery, the bulkier the coasters. This is undesirable because these are supposed to be portable, easy to use devices that don't over-clutter the table at a restaurant. If they're too bulky, they will be an eye-sore and customers will not want to use them. It will also make it harder on employees to gather all of them and charge them. The charging dock will have to be bigger to accommodate the increased size. Therefore, for this applications, smaller dimension are better. Unfortunately, smaller dimensions translate to lower capacity so those two requirements must be balanced with each other.

Table 8: LiPo Battery Comparison

Battery	Capacity (mAh)	Dimensions (mm)	Price ($USD/battery)	Capacity per dollar (mAh/$)
F.Dorla	600	58 x 18 x 7	3.17	189
HEIOKEY (battery 1)	520	44 x 20 x 8	3.50	149
HEIOKEY (battery 2)	720	43 x 25 x 8.5	3.32	217
Keenstone	500	42 x 21 x 8	3.67	136
Morpilot	720	42.5 x 24 x 8	3.50	206
Tattu (battery 1)	380	38 x 18.5 x 8	2.33	163
Tattu (battery 2)	600	41 x 25 x 9	3.00	200
Tattu (battery 3)	800	44 x 24 x 9	3.33	240

Finally, the price of the batteries also matters. In theory, this would be a high-volume, low-cost device. If it were to be mass produced, having cheaper batteries is better because it lowers the price per unit for both the manufacturer and the customer which is the restaurant. Higher capacity batteries tend to be higher price, which is another pair of requirements that work against each other.

In Table 8, we compare eight batteries based on these four considerations. The top row has a list of the properties of the batteries and the left column has the eight batteries. The colors indicate how good the value is compared to that of the other batteries. Green boxes indicate that the battery has a relatively better value for that property, yellow indicates an average value, and red indicates a value that is worse than normal.

All of the LiPo batteries that were considered have a continuous discharge rate of 20C or 25C which is significantly more than necessary for the Clever Coasters application, so this value can be ignored when comparing them and was omitted from Table 8. Instead, a new value was added as a measure of price-efficiency which is the capacity per dollar of the battery. The higher this number, the better the battery.

The dimensions of all the batteries are very similar. There are slight differences in length and width and only a ±0.5mm difference in thickness for all but one battery. In the end, all of the batteries have a reasonable size, even the ones marked with green and red. Because of this, it is not a very important consideration for the final decision.

The most important consideration is the capacity because it will determine how long the Clever Coasters can work. Without knowing exactly how often they will need to transmit data, it is better to err on the side of caution and choose a higher capacity. As such, we narrowed down the choices to batteries that had at least 720 mAh of capacity.

Given that, the price was the last consideration. Of the three batteries, the Tattu 800mAh battery had the highest capacity per dollar ratio and was only one cent more expensive than the cheapest battery, which had 720mAh. Because of this, we chose the 800mAh Tattu battery for the Clever Coasters.

3.5.4 LiPo Battery Charging Theory

Different types of batteries require different types of charging. For example, if attempting to quickly charge nickel-based batteries, a constant current source is required in addition to some kind of end-of-charge detection based on temperature or voltage. Lithium based batteries, on the other hand, charge at a constant voltage and a limited current. Since LiPo was chosen as the battery type, it is only relevant to discuss in-depth the theory behind charging a LiPo battery.

3.5.4.1 Constant Voltage Charging

Lithium-ion polymer batteries charge in two stages: the current limited stage and the constant voltage stage. There is a safe charging current, usually 1C (or 1 times the capacity of the battery, so for a 800mAh battery, 1C equates to charging at 800mA). This is the current that the charging is limited to in the first stage. As charging continues, around when the battery reaches two thirds of its charge, it reaches its target voltage. In most cases, the target voltage for a LiPo is 4.2V. When this happens, the constant voltage charging stage begins.

The constant voltage charging stage maintains the same battery voltage while the current decays exponentially. This constant voltage stage charges the remaining third of the battery capacity though it takes about twice as long as the first stage. As such, the first two-thirds of charging takes one-third of the time and the remaining one-third of charging takes two-thirds of the time, assuming the battery is being charged at its maximum charging rate.

The reason that charging continues at a constant voltage is that a battery has an internal resistance. When the current is high, the battery voltage, as seen by an outside element, is the combination of the cell voltage and the voltage drop across the internal resistance due to the current. This means that the cell is only fully charged when the battery voltage is 4.2 and the current passing through the battery is miniscule. Of course, using the same approach it is possible to charge the battery slowly if the current limited stage of charging is set to a current of less than 1C.

One important consideration when charging a LiPo battery is charging to almost exactly the rated cell voltage, 4.2V. Since LiPo batteries are sensitive to overcharging, care must be taken to ensure that they do not charge above this voltage or battery damage could occur. At the same time, charging to less than this voltage will result in a significantly undercharged battery. This balancing act can be accomplished with a linear regulator circuit, but the values of the resistors used to bias it must be carefully tested and finetuned since the ±5% values can make a significant difference.

3.5.5 Low Dropout Linear Voltage Regulators

Based on the choice of Lithium-ion Polymer batteries for use within the coasters, and the choice of charging dock as the charging solution, there is a need to have a special circuit within the coasters to charge the battery. This is because the LiPo batteries are very sensitive to current, voltage, and any kind of overcharge. Because the charging dock solution is somewhat simple, and because it attempts to charge multiple coasters in parallel, it is not able to precisely control the voltage and current it provides each coaster and so we must control that from within the Clever Coasters with a charging circuit.

This circuit will be able to take in a somewhat variable DC input voltage and it will force current into the battery until it reaches the correct voltage. We will use a low-dropout linear voltage regulator with a feedback circuit to guarantee that the battery charges as a LiPo should: with a current limited phase followed by a constant voltage phase as discussed in the LiPo battery charging theory section. To choose a low dropout linear regulator, we have several important considerations that will translate into requirement specifications for the regulator. These are discussed below.

3.5.5.1 Important Considerations

The first and most important thing is to ensure that the charging circuit only pushes current into the battery until the voltage across the battery terminals is 4.2V and no more. The more precise this value is, the more charged the battery will be. If it goes over the limit, the battery may take damage which would be a safety risk and would decrease the lifespan of the battery. This means that the linear regulator must have an output range that encompasses the 4.2V value and it must have a variable current output.

The second consideration is that the current in the current limited phase should not exceed 1C. This is important for the same reasons that the voltage is: too high of a current can damage the battery, lead to a safety risk, and decrease the lifespan of the battery. This means that the linear regulator must have an output of less than 1C. We chose the 800mAh Tattu battery so our linear regulator must output no more than 800mA.

The third consideration is the amount of current that will be leaked through the resistors in the feedback loop when the coasters are not in the charging dock and power is not being provided to the linear regulator. Since the battery will be directly connected to these resistors, they will dissipate power constantly. This needs to be as low as possible so that the Clever Coasters can last through a restaurant service. Ultimately, this means that the equivalent series resistance of the feedback resistors needs to be as high as possible. This will require looking into the technical specification of the linear regulators to determine what range of resistors they are able to support in their feedback loop.

To calculate how long it would take this feedback resistance to drain the battery completely, we can calculate the average current that would be drawn by it when in series with the battery. We then divide the capacity of the battery by this current to find the number of hours like so:

$$\frac{800mAh}{\frac{3.7V}{R_{FB}}} = Hours\ to\ fully\ drain\ battery$$

In reality, if this resistance is in the KΩ range, this drain would not affect daily operation. Based on the above equation, a 1KΩ resistance would take over 200 hours to drain the battery. This is far longer than is expected for the coasters to remain uncharged. This does, however, become a safety issue. If the coasters are manufactured with partially charged batteries, they will spend many days being transported or sitting in a warehouse. It is unrealistic to expect all of them to be constantly connected to a charger. Other situations may involve the coasters being drained during their normal operation and then being misplaced for some reason. Since LiPo batteries are sensitive to overcharge and extreme discharge, it is better if they take longer to passively discharge. Based on the equation, a feedback resistance of over 1MΩ would discharge so slowly that it would take upwards of 25 years to empty a full battery. This is desirable as a safety feature, more so than something relevant in normal operation. In theory, if a LiPo battery were discharged below a certain point, even this slowly, it could get damaged and light on fire. A timeline of decades, however, ensures that it will fully outlive its expected lifetime and be recycled before it gets to that point.

Furthermore, the input voltage must be high enough to charge the battery to 4.2V while overcoming the voltage drop across three 1N4001 diodes. Two of these diodes are part of the design for the charging circuit, as seen in figure 10, to allow the coasters to be placed in the charging dock in any orientation. The last one is used to prevent the battery from sending current back into the linear regulator when there is no input voltage. This amounts to a worst case minimum input voltage of:

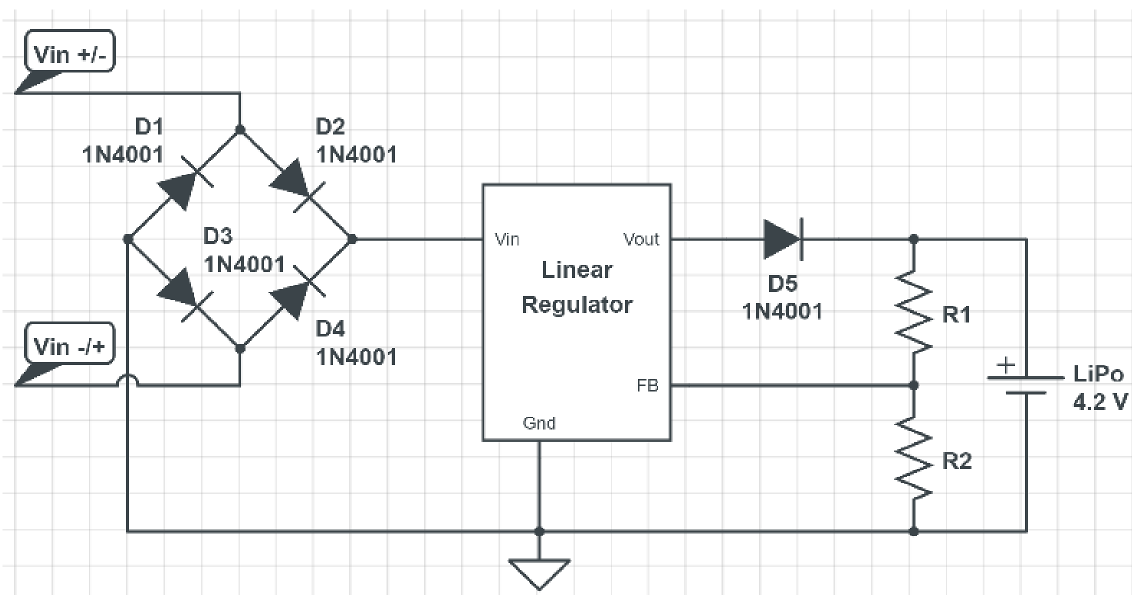


Figure 10: Charging Circuit Block Diagram

$$V_{in} = 4.2V + 3 * (1.1V) + V_{do}$$

Where V_{in} is the input voltage, 4.2V is the voltage across the battery, the three times 1.1V is the sum of the worst-case forward voltage of the three diodes, and V_{do} is the dropout voltage of the linear regulator. This is a minimum requirement of a voltage regulator.

Another consideration is the price of the linear regulator. As discussed before, Clever Coasters are intended to be manufactured at scale so choosing more inexpensive components is a better decision. The best regulator to choose would be the one that can satisfy the requirements and is also cheaper than the alternatives.

One minor consideration is that the charger should recharge the batteries in the coasters in a reasonable amount of time. In general, charging at 1C can be expected to fully charge a LiPo battery in about two hours because of the constant voltage phase of charging described in the LiPo Battery Charging Theory section. For the Clever Coasters application, we want the coasters to fully charge from empty in less than ten hours because most restaurants would use them for at least 12 hours a day.

Of course, this requirement can be somewhat relaxed as long as the battery can be expected to not be fully drained by the end of service, which is part of the requirement for the coasters. Since we are using 800mAh batteries, they may take longer to charge if charged below 1C, but they are likely to not need a full recharge after a day of use.

3.5.5.2 Comparison of Linear Regulators

Based on the basic requirements for the low dropout linear voltage regulators, only those regulators that had a variable current output and a voltage output that falls into the 3V to 4.2V range of the batteries were considered for use with Clever Coasters. All of the linear regulators considered are from Texas Instruments because they have a wide range of products with extensive and detailed technical documentation.

Table 9 compares five low dropout linear voltage regulators on the basis of their maximum output current, input voltage range, output voltage range, dropout voltage, quiescent current, accuracy, pin arrangement, total resistance through feedback loop, and cost. Once again, the table is color coded to indicate whether a particular value is good or bad. Green is good, red is bad, yellow is decent. The properties are also color coded based on how important they are, red being the most important and green being the least important. All of the regulators meet the minimum requirements. Those that did not, were removed before further consideration.

Based on the values in Table 9, it is clear that not all of the linear regulators will suit the Clever Coasters application. LP38798 is a complex and feature-rich regulator which also causes it to be the most expensive of the group. Not only that, but it has a low feedback loop resistance which may be troublesome when the coasters are not being charged. The pin setup on it is also larger and more difficult to connect to a circuit board. It is not a good choice.

TPS7A49 isΩ another poor choice mainly because of its price, which is the second highest of the group. Besides its shortcomings in these areas, it doesn't have an easy to connect pin package, nor does it offer an impressive output current that would make up for its limitations. It is best suited for other applications.

Table 9: Low Dropout Linear Voltage Regulator Comparison

	LP38798	TLV1117	TPS7A19	TPS7A49	LP2951
I_{out} Max (A)	0.8	0.8	0.45	0.15	0.1
V_{in} range (V)	3-20	2.7-15	4-40	3-36	2-30
V_{out} range (V)	1.2-19.8	1.25-13.7	1.5-18	1.2-33	1.2-29
V_{do} (mV)	200	1200	240	260	380
I_q (mA)	1.4	1.7	0.015	0.06	0.075
Accuracy (%)	2	1.6	2	2.5	2
Pin/Package	12WSON	4SOT-223	8SON	8SON	8SOIC
FB loop resistance (kΩ)	<250	Variable (can be 1000+)	<100	<780	Variable (can be 1000+)
Cost per unit ($USD)	3.25 (1) 1.30 (1k+)	0.72 (1) 0.19 (1k+)	1.61 (1) 0.59 (1k+)	2.75 (1) 1.10 (1k+)	0.68 (1) 0.18 (1k+)

TPS7A19 is in the middle of the pack regarding price, but it also has the same problematic pin arrangement that would require a special design on the PCB. Additionally, it has the worst feedback resistance which makes it a poor choice for Clever Coasters. It does have two good features: very low dropout voltage and the lowest quiescent current. These are, however, related more to efficiency of charging and are not as relevant as the other properties.

TLV1117 is the second cheapest and only by four cents when buying individually and one cent in mass quantities. Not only that, but it can have a feedback loop resistance greater than 1MΩ which, as discussed earlier, is key to ensuring a safe passive discharge if the coaster were left uncharged for a long time. The main downsides of this particular low dropout regulator are that it actually doesn't have a very low dropout and it has the highest quiescent current. The dropout is the highest of the group at 1.2V which is over four times higher than the average of the others. This is not too much of an issue because these are mostly related to efficiency. On the other hand, it has a maximum output current of 800mA which is exactly what is needed to charge the LiPo batteries as quickly as is safe. Out of the five, it also has the best accuracy. It is definitely a good option for Clever Coasters.

Finally, LP2951 is the cheapest linear regulator of the group coming in at only 68 cents in low quantities and 18 cents in mass quantities. Its main drawback is that its maximum output current is only 100mA. This will mean that the batteries take a long time to charge, but they will definitely charge at a safe rate. Assuming they charge overnight, this shouldn't be a problem. All of the other features of this linear regulator are fairly good. Just like TLV1117, it has the potential for an extremely high feedback resistance. It is another very good choice for the Clever Coasters application.

3.5.5.3 Linear Regulator Choice

Of the five linear regulators considered, two had great numbers in the most important categories: LP2951 and TLV1117. Instead of choosing one of these, we will design circuits and test both of them to determine the better option based on their real-world performance. They are both very cheap to obtain and allow for similar circuit designs that meet all the requirements of the project. It may come down to the stability of the output voltage.

3.6 Weight Sensors

To detect whether a customer's glass is empty or full, weight sensors must be placed within the coaster. For the employee request mode, a touch sensor or simple button may be used to turn the mode on and off. This sensor data must then be sent over to the waiter through our chosen communication channel.

For the weight sensors, several options were explored to maximize cost-efficiency and accurate data readings. The goal is to convert the analog pressure input into digital data that can be sent over wireless communication. Once a sensor is chosen the challenge will become how to position the sensor so that it can contact the cup placement while still keeping the electrical components enclosed and water-proof.

3.6.1 Force-Sensitive Resistor

Force-Sensitive Resistor (FSR) sensors can detect physical pressure and weight. The FSR is made up of two layers: a flexible substrate with printed semi-conductor and a flexible substrate with printed interlocking electrodes. A spacer adhesive is then placed between these two layers to measure resistance as the sensor is pressed. When no pressure is applied to this sensor, then it basically acts as an infinite resistor, which goes down as pressure increases. If a pull-down resistor configuration is used, then the voltage reading will range from 0 V to around the same voltage as the power supply of the microcontroller. One advantage to using the FSR is that they use less than 1 mA of current. Figures 11 and 12 are examples of FSRs and their layers. [6]

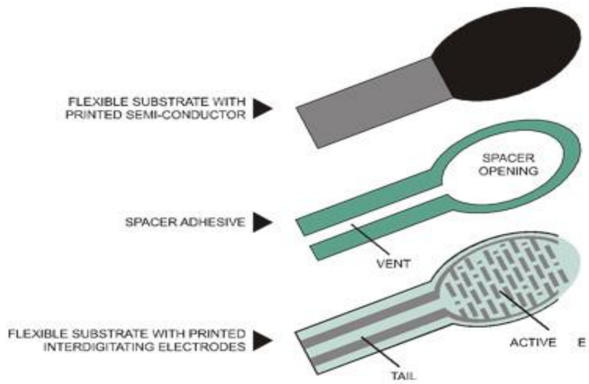


Figure 11: FSR Layers

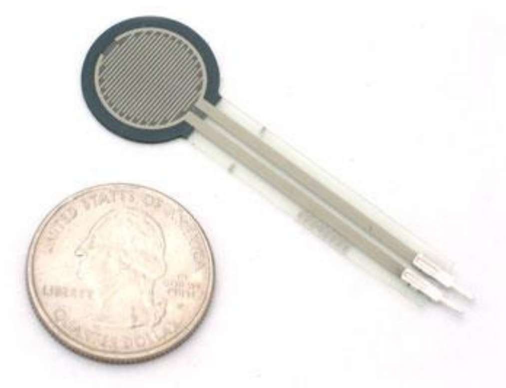


Figure 12: FSR Compared to Quarter

They come at a low cost of $7 per ½ diameter sensing region. The smaller the resistance detected on the sensor, the heavier the object on it is. This inverse relationship between force and resistance is not accurate enough to detect exact measurements, but can be used to determine what range of weight an object is in. Therefore, for detecting a drink that is around 30% full, this sensor is a viable option to place in the coaster's center. However, if six coasters were prototyped, costs will certainly add up. The images below show the small size of the round FSR by adafruit and how the FSR works through layering.

3.6.2 Velostat

Velostat is pressure-sensitive conductive material that can be purchased at $4 per 11"x 11" sheet. It is flexible, making it a convenient and cost-efficient method for measuring resistance. The sheet can be cut into any necessary size and shape, allowing at least 4 coasters to be equipped with a weight sensor, with just one sheet. To create a pressure-sensor with Velostat, a conductive thread needs to be touching the Velostat (by taping on) which would then be interfaced with analog input ports on the microcontroller. These analog input ports would need to convert to digital data, in order to interpret the readings into the software. How Velostat works is by pressing onto the flexible piece of material, resistivity goes down as voltage goes up, which directly correlates to the amount of pressure applied. Since the material isn't ready-to-use when purchased, more time and effort is required in ensuring a consistent weight sensor is implemented for each coaster.

3.6.3 Load cells and Wheatstone bridges

A load cell creates an electrical signal with a magnitude proportional to the current force being measured. Our load cell choice of interest would be a type of compression load cell, with a compact size. Our constraint for this option is their relatively high-cost per load cell. If we need six coasters made, this can ramp up costs by a high amount. However, load cells do provide a great amount of accuracy and can see a large range of forces.

A Wheatstone Bridge provides more accurate resistance measurement of an unknown resistive value. Since the change in resistance indicates a change in the load, a Wheatstone bridge can be implemented in addition to any of our weight sensing methods.

3.6.4 Piezoelectric Force Sensors

Piezoelectric Forces sensors uses the effect of piezoelectricity to measure dynamic forces. Piezoelectricity is the appearance of a voltage across the sides of a crystal when you apply mechanical stress (squeezing) to it. [5] A piezoelectric transducer can then convert this energy into electrical sensors that can detect the amount of force applied to the crystals. The issue with Piezoelectric Force Sensors is that they leak over time, so if a cup was placed on the sensor, it would eventually read as the cup being empty, even if the cup was full. However, if the beverage was picked up (such as the case for a beer bottle or any beverage without a straw) and put down again, the Piezoelectric sensor would re-calculate the new weight. This dynamic property makes the sensor a viable option for our coasters.

3.6.5 Summary of Weight Sensor Options

For our project, we believe that low cost and moderate accuracy is important for selection of a Weight Sensor. The table below summarizes the 4 sensor options under consideration. Although Load Cell is very accurate, the cost is too high to implement into our project. If we were to embed a load cell into 6 coasters, we would waste $120 dollars just to employ a simple functionality. Therefore, we will move forward with FSR and Velostat.

Table 10: Summary of Weight Sensor Options

	FSR	Velostat	Load Cell	Piezoelectric
Cost	~$7	~$4	~$20	~$3
Accuracy	~25% error (not very accurate)	Not very accurate	< 0.1% (very accurate)	± 5 (less accurate)
Force Range	0 – 20 lbs	~ 1000 g	0-10 kg	-

3.6.6 Selection of Weight Sensor

We are in the testing phase of deciding whether FSR or Velostat is more accurate. For now, we will purchase both options and see which one gives us more accurate measurements. In chapter five, we discuss our selection based on the component testing we do.

3.7 Restaurant Tablet Applications

Since we will want our smart coasters to wireless communicate to the table hub and waiter/waitress's tablet, an application design is needed where they can seamlessly work together to provide an enhanced restaurant experience. Instead of creating a restaurant application from scratch, we can modify an existing one by integrating our weight sensing feature and party games into the mix.

3.7.1 Open-Source Server POS

A point of sale (POS) system is a transaction tool used by businesses to keep track of their day-to-day operations. Restaurants can utilize specific POS software that is tailored to keeping track of their customer orders, table occupants, and monetary exchanges. For our project, finding a compatible open source software that already has restaurant organization set in place will be useful when trying to add the "empty glass" notification for the server to see when checking their POS system. It will be linked to the table corresponding to that specific glass. We would like this system to be simple to use, and modular in design to allow for flexible functionality.

3.7.1.1 Floreant

This intuitive and feature rich software is open source and has the modular specifications we would like in a POS system. Floreant is easy to install and has a lot of features that assist a business in managing their processes. Some of the business types it caters to include bars, fine dine-ins and cafes. The backend of this software is in Java, and is available for download on Mac, Windows, or Tablets, making it a very practical choice.

3.8 Other Potential Components

Aside from our main components, a few other pieces of technology can be useful when making design considerations. We will be purchasing and testing some of the following components, so that we have a better idea on how to put everything together into a functional and seamless design, with unique features.

3.8.1 LEDs

In terms of LED requirements, a few things come to mind. The coaster needs to be able to fit a lot of components into a small space, so our LEDs should be as miniature as possible. Also, to further save space, RGB LEDs would be useful in providing multiple colors indicating coaster status. These LEDs will be soldered onto the PCB, and they require on average about 3.3 V of forward voltage. Static electricity and surge can damage the LEDs, so when handling them, we need to make sure we are electrically grounded.

3.8.2 Buttons

A simple tactile switch button can be utilized to allow customer to request service at any given time. The lifetime of one of these buttons consists of around 100,000 presses. These buttons would also need to send individual data to the microcontroller, while allowing current to travel to the LEDs when pressed.

Another option consists of tactile switch buttons with built-in LEDs. Using those kinds of buttons could further save space on the overall coaster's internal design. These buttons are slightly costlier than the ones without LEDs, but not enough to make it bad choice to consider.

3.8.3 Touch Sensors

Likewise, instead of using buttons that protrude from the coaster's body, it might be sleeker to include a touch sensor that allows similar functionality. This would make integration and waterproofing of the casing design to be simpler and more reliable. However, this option might be costlier for production. Another issue that could arise is the accidental pressing of the touch sensor. If a customer brushes their arm on the coaster, this could accidently activate the request mode. To reduce this issue, capacitive-based touch sensors can be used instead of resistive. Likewise, an algorithm can be implemented to require the user to keep their finger pressed to the sensor for a few seconds, instead of an instantaneous reaction.

3.9 Selection of Major Components

We have narrowed our component selection to several major parts, with a few extra ones that will be tested and compared. For weight sensing, we purchased the Round Force-Sensitive resistor from Adafruit and a sheet of Velostat. We also purchased two different NFC modules which we will test to decide which to move forward with. Figure 13 shows almost all of the major components with labels of what they are.

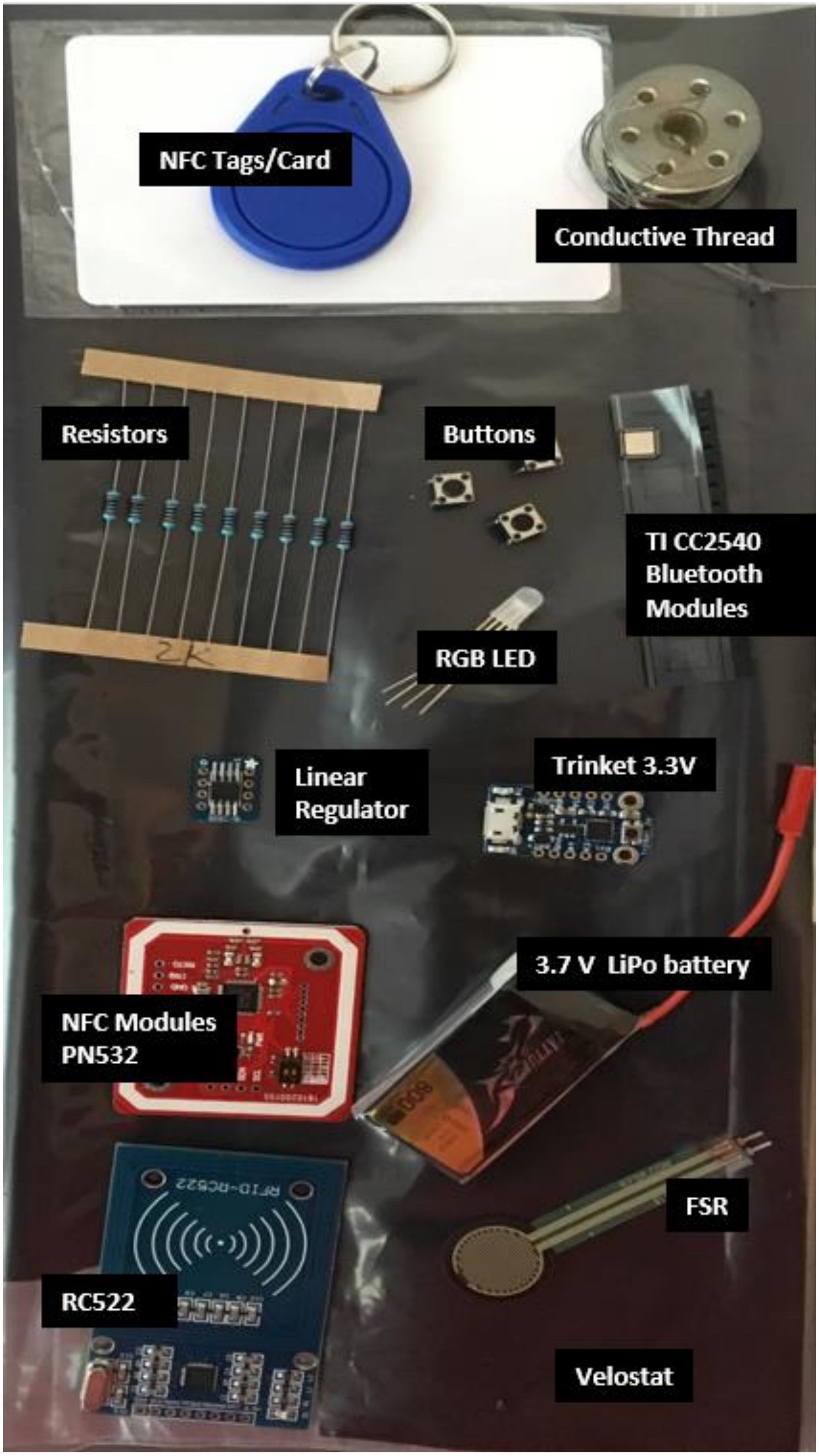


Figure 13: 90% of major components

4.0 Design Constraints and Standards

In any engineering related project, it is important to address the various design constraints and standards that are relevant. Engineers must consider the technical, economic, social, environmental, and political constraints when they design products and processes. Doing so will lead to an overall better design and reduce errors in Senior Design 2. We will focus on standards directly related to our project that are accessible to read for free.

4.1 Realistic Design Constraints

It is important to be aware of the design constraints that our project will face. Failing to make notice of these could result in impairments or complications in the final product due to poor advanced planning on our part. All constraints must be realistic when considering the overall design of our clever coaster system.

4.1.1 Economic Constraints

As college students with limited income and no sponsors, we require a lot of budgeting and workarounds to keep total production costs low. Likewise, because the intended customers of our products are casual restaurant owners in the Orlando area, their budget for purchasing the clever coaster system is probably much lower than that of a high-end five-star restaurant. Therefore, to make this product desirable to the average restaurant owner, it must come at a low cost that won’t break the bank, while also guaranteeing an improvement to the overall customer experience at the restaurant. With this, if we try and make the coasters as simple, yet functional as possible, many restaurants will be able to easily integrate this system into their existing environment. The more modular we design our components, the simpler they will be to interface and control.

There are several costs we need to consider when designing our clever coaster system. These include manufacturing costs, development costs, and other resources needed to build the system (capital, machines, printers, etc). To save on costs, we can receive free samples from Texas Instruments to help with our testing, as well as utilizing the Senior Design and TI Lab at the University of Central Florida. Having these free resources on campus will make the development process much more cost-efficient, and we will mainly have to focus on budgeting on component purchases. Since we need to buy backups for every component we choose, it is important to budget appropriately. We do not have sponsors, so development costs are basically the time that we as a team put into the project. Likewise, if this product were to be commercially distributed, we should budget for an overall system that is compact and easy to ship to local restaurants.

4.1.2 Time Constraints

Time is one of the most important constraints to consider, since our team consists of only three people, all of which are Computer Engineers. This means it will take longer to figure out how to design all the electrical components of the coasters, as well as building the prototypes. To mitigate this, we decided to focus a lot of features into the software, and keep the PCB design simple enough to incorporate the main functions such as weight sensing, charging, and wireless communication.

There is a strict time constraint for the overall project, and since we will be completing Senior Design 2 in the Fall, we get about an extra 12 weeks of planning, research, and design decisions, which we should take advantage of to start off strong in Senior Design 2. For the initial phase, Senior Design 1, we also have a time constraint of considering each of our busy, individual school schedules. We are either working part-time or taking a full load of other classes that will make time management an important consideration when planning the design schedule. Our project schedule can be found in the end of this report, where we decide when our milestones should occur, and what to expect over the next few months.

Likewise, since we only have a year to complete this project, budgeting plenty of time for ordering components is essential. Shipping, especially if ordering parts internationally, can take anywhere from 2 days to several weeks. This means that we should aim at ordering all our major components during the Spring semester and summer, to ensure system testing and prototype building can begin as soon as the month of August. The PCBs will be ordered in the beginning of the Fall semester so that we can re-design or re-order as soon as possible if any issues arise.

4.1.3 Environmental Constraints

Protecting the environment is critical to the long-term well-being of all creatures on Earth. It is important to keep in mind all of potential hazards and environmental impacts that can happen due to the use of certain materials. While our project is not designed to directly deal with the environment, it still has potential to cause impacts and even have negative consequences.

The type of battery used is an important choice from an engineering perspective, but it also can have serious environmental impact, too. For example, Nickel based batteries and lead based batteries are composed of heavy metals that, if they enter the environment, can cause severe harm. Heavy metals are toxic and as such must be limited in their use. That said, there are ways to recycle those metals, but it requires forethought and preparation. Restaurants cannot be expected to know that they shouldn’t just throw away a defective coaster. Instead, it should be made very clear that they must send it back to the company or manufacturer for recycling. This would greatly reduce environmental impact.

Lithium based batteries, on the other hand, are very environmentally friendly because their composition is not nearly as toxic as heavy metals. If they end up in the environment, they quickly react with other elements and return to their stable, inert state that they were originally extracted from. Of course, there is also the consideration of how the lithium was obtained. These metals have to be mined and mines can sometimes cause environmental damage as well, by contributing to erosion or destroying the habitats of local animals. Still, the upside of not using heavy metals, which have to be mined as well, is significant. Lithium batteries can also be recycled which adds further value to them as an environmentally friendly option.

Finally, the use of plastic in the coaster casing is another environmental consideration. The two main plastics we considered for use are PLA and ABS. PLA is completely biodegradable. It is used in medical implants and various other items that are expected to enter the environment and need to safely dissipate. Unfortunately, PLA is not very strong or durable. It is quite brittle and it would not be the best option for our application. ABS is a much better option, but it is not environmentally friendly. It will not degrade easily. To combat this, we should again encourage recycling of the coasters instead of throwing them out. ABS is very recyclable and so it can be melted and reused for other applications. In this way, we can help protect the environment which we all share. [20]

4.1.4 Ethical and Legal Constraints

There must be continuous and careful consideration of how the product is created to ensure that every aspect follows ethical and legal guidelines. When changing a part last minute or considering a new feature, there must be deliberate testing involved to ensure ethical practices to continue. There are several prototypes of smart coasters floating around the internet, so exact schematics should not be copied, only used for inspirational guidelines. Likewise, there shouldn't be any "cutting of corners" as this could lead to major complications with how the system functions. We need to be sure that our wireless communication modules do not mess with the mobile phones and devices that customers may bring to the restaurant. Public safety is a crucial ethical constraint to consider when designing the electrical components, especially the charging circuit. We do not want our product overheating or blowing up, so we will focus our efforts in producing quality circuit designs and perform deliberate testing before letting it become commercially available in the market. Customers want to avoid legal repercussions whenever possible.

4.1.5 Social Constraints

This product will be used in a very social environment. Restaurants are inherently social because of the interactions between customers at the same table and also their interaction with waiters and other restaurant staff. It could be claimed that Clever Coasters are socially damaging because they take away from the waiter's job by doing part of it.

That said, we believe that our system inherently augments and doesn't take away from the social aspects of a restaurant. By taking care of the issue of checking on customers for water refills, it saves time for waiters which they can spend interacting with customers more. In that way, we believe that it can actually promote a social environment.

Another social constraint is the ease of use of the system. It must be intuitive, or at least easy to learn, for the waiters and definitely intuitive for the customers of the restaurant. If it is not, it may actually do more harm than good because some users might ignore it which could result in waiters not refilling their cups as often as they should, or otherwise being called back to the same tables and not giving others enough attention. This could negatively impact the experience of customers at the restaurant which could leave them with a bad impression of the establishment and/or the employees. That in turn could decrease the amount they give as a tip which would affect the livelihood of the employees that depend on receiving gratuity.

4.1.6 Privacy, Health, and Safety

Since the product will be in direct contact with the consumer, one constraint is to ensure that no recording is done with the device. Tracking data of the weight of the cup is allowed, but sensors should not detect data about the customer without their consent. Likewise, the product should not create any wireless communication interference with the customers' phones as well.

In terms of safety, the prevention of shocking the user is crucial, as the coasters will be in contact with varying liquids and condensation. This can be accomplished by ensuring that liquids cannot penetrate into the areas where there is any kind of power source.

Furthermore, consideration for usage by children must be accounted for. This means that warning signs need to be placed in clear view as required, so that parental guardians can ensure the risks posed (if any) of nearby technology. This will ensure the mitigation of legal repercussions towards the restaurant owners if an incident were to occur. The less blame the owner can take, the more inclined they will be to purchase our product.

This product will be around food items. It will be handled by people who handle food. It is important that it is not easily contaminated and that it does not itself pose a health hazard. Since the casing will be made of plastic, it will be inert and not cause any danger that way. It should also be large enough that it cannot possibly be swallowed by accident because there are children in restaurants.

Furthermore, it must be cleanable because if something like raw meat juices got on it, or if someone sneezed or coughed on it, it should be possible to wash so that another customer is not harmed by bacteria or viruses that are trapped on the surface of the coaster.

Training is necessary to ensure that the servers understand how the clever coaster system operates. A training or operations manual should be provided to the customer in a manner which is easily understood and concise. A support hotline should be available in case any confusions, questions or comments arise during the use of this product. Likewise, the product should not cause excessive damage to the environment around it, when shipped or opened by a customer.

4.1.7 Manufacturability and Sustainability

Modularity will be an important factor in whether our product can be reproduced easily and efficiently. The system should be split into three main components: the casing, the electronic component, and software component. The most important part is the electrical component, as it must stay consistent from coaster to coaster.

Likewise, our system must be sustainable if a part becomes damaged. If components within the coaster aren't easily accessible or replaceable once manufactured, then customers will assume low reliability and sophistication from the system. One way to ensure higher sustainability is to make sure to match the lifetimes of each component to be as close as possible. If the weight sensor deteriorates sooner than the battery, then one would see the system as designed poorly. Overall, with subsystems that are both compatible and replaceable, our components can be split and manufacture independently.

- Surface Mount Devices shall be limited to standard package sizes
- All components should be sourced from a reliable and fair vendor, especially when ordering the printed circuit boards
- Casing design should have a well-thought out CAD design that can be printed across several 3D printers with the same outcome
- Material used for casing components should be durable and sustain a lifetime of at least two years
- Should try to "future-proof" our system design as much as possible to ensure longevity in the market, and make updates to the software or hardware as seamless as possible
- Assembly of the system by the customer shall be minimal, to ensure ease of use and good ergonomic design

4.2 Standards

In context to our project, there are relevant standards that must be considered when deciding the components and design of our clever coaster system. This is to ensure responsibility is taken in case the project becomes commercialized in the future, and the questions of whether correct practices are being utilized or not.

4.2.1 Ingress Protection

The IP Code (International protection rating) classifies the degrees of protection against dust, water, and accidental contact in electrical enclosures. [7] For our coasters, we are most concerned with the IP44 rating, which says that the electrical enclosure shall be protected from a water spray in any direction. This makes sense, as the coaster will mainly just have contact with condensation and cleaning supplies such as paper towels, as well as cleaning solution. Following this standard ensures that we maintain the safety of the customer to avoid any sort of electrical shock from water contact.

The first digit in the IP44 rating indicates the level of protection against hazardous solid particles, while the second digit indicates the level of protection that the enclosure provides against harmful liquid. In the IP44 rating, level 4 in the first digit says that the enclosure should be effective in protecting against solid particles with a size greater than 1 mm. This can include most wires, slender screws, large ants, etc. The level 4 in the second digit specifies that the enclosure should protect against splashing water from any direction. This splashing of water can be tested by using a spray nozzle with no shield for at least 5 minutes all over the enclosure. Ensuring IP44 standard is followed will allow for a product that is ready for commercialization, and will work well under restaurant conditions. The inclusion of an Ingress Protection rating has become increasingly common for use in consumer electronics, and is especially important for “smart” devices.

4.2.2 WPA Standard

IEEE 802.15 is the standard regarding Wireless Personal Area Networks such as 802.15.1 Bluetooth technology. It is also referred to as the WPAN Task Group 1 (TG1). It specifies the Media Access Control (MAC) and physical layer of wirelessly connected devices through Bluetooth. Its operational spectrum is on the 2.4 GHz ISM band, and it uses a Master-Slave configuration. WPA was designed specifically to work with wireless hardware products.

With WPA, it is important to recognize security standards and encryption. One encryption protocol used by WPA is the Temporal Key Integrity Protocol (TKIP) which generates a 128-bit key for each new packet. It is used in the IEEE 802.11 wireless networking standard, and was originally a replacement for WEP since it has link-layer security within. We do not want customers using their mobile device to hack into our system, so it is important we employ some security standards so that this issue is minimized. Since the internet is open-access to all, it becomes increasingly difficult to restrict communication between two devices, so this Standard will be a big consideration throughout our design.

4.2.3 IEEE 802.15.4 Standard

In order to utilize the ZigBee wireless module into our design, we must follow the IEEE 802.15.4 standard when configuring the management software needed to control and regulate ZigBee. This standard is for low-data-rate WPANs. It defines the physical and media access control layers of the OSI model. Furthermore, it only uses the first two layers plus the logical link control (LLC) and service specific convergence sub-layer (SSCS). [8] For ZigBee, it includes encryption for security, and a data routing and forwarding capability that enables mesh networking. [8]

4.2.4 Battery Standard

There are general standards and specifications set by the American National Standard on how rechargeable cells and batteries should be operated. This is the ANSI C18.2M standard, which is based on the following electromechanical systems: nickel-cadmium, nickel-metal hydride, and lithium-ion (including lithium ion polymer). We cannot use this standard, as there is a $100 cost associated with reading it. However, we will ensure that we use common, safe battery practices.

4.2.5 Software Testing Standard

The ISO/IEC/IEEE 29119 Software Testing Standard can be used within any software development life cycle, or by any organization. This breaks down the overall Test Management Processes into three parts: Test Planning, Test Monitoring & Control, and Test Completion. Keeping aware of this standard will help us focus and create a good test method to make sure our software integrates smoothly with the hardware. The purpose of this standard is to approach risk-based testing and can support test planning and strategy development. This allows testing to be within the scope of problem space, cost, and schedule. We want to test the best and worst case scenarios in our clever coaster system.

When following this standard, we will start off by test planning, with an objective, procedure, and expected results section. The next step is to monitor the test procedure and use a debugger to view the step-by-step walkthrough of each software process and just observe what occurs (and document it of course). Lastly, is the test completion phase which will confirm whether the test passed or failed, and should provide further steps if the latter occurred. Overall, using this standard will help streamline the software testing process and allow it to be modular. If it is modular, then it can be distributed amongst multiple team members.

4.2.6 RoHS Standard

The Restriction of Hazardous Substances standard was created in the European Union as the RoHS Directive 2002/95/EC. It limits the use of certain dangerous substances in electronic components. If a device is to be sold in RoHS countries, it must meet the restrictions, measured in parts per million, on those substances. The materials in questions are heavy metals like lead, mercury, cadmium, and hexavalent chromium and flame retardants such as polybrominated biphenyls or polybrominated diphenyl ethers. Since our coasters will use LiPo batteries and not NiCd or Lead Acid, this will not be a difficult standard to meet. We must also check that each of the components that we use meets this same standard.[18][19]

4.2.7 WEEE Standard

The Waste from Electrical and Electronic Equipment standard maintains requirements for the collection and recycling of electronic waste. This is because most electronic waste ends up in landfills and could instead be recycled. This standard requests that producers of the electronic equipment have some sort of collection scheme such that end users could return the used item back to the producer at its end of life. [21]

It requires that producers provide information about how to properly dispose of the equipment to end users. This is relevant and even beneficial to the Clever Coasters application because most part of it can be recycled. If one part of the coaster is no longer capable of working as intended, it may be possible for us to fix it without completely replacing it. Even if it does require replacement, the recycled materials could be reused to make more.

5.0 Project Design

The overall design of the project comes down to three major parts: the hardware design, the software design, and the system housing design. Hardware and software are relevant to Computer Engineering and the casing design is not. It is, however, still a necessary part of the project.

5.1 Hardware Design

In this section, we explore the chosen designs that will make up overall system design, including schematics and component testing.

5.1.1 Weight Sensing

Before designing and choosing overall schematics, we must first test the two weight sensor options that we purchased. We wanted to make sure we picked the right one before ordering in bulk. We tested the FSR and Velostat by measuring the different resistances seen when different weights and pressures were applied. Then, we will design a way to read the analog input seen by the chosen sensor.

5.1.1.1 FSR Test

Objective: The objective of testing the FSR is to ensure that it detects weight by simply adding objects on the round surface. This testing procedure will also look at the effect squeezing, tapping, and light pressure have on the FSR.

Procedure:

1. Place an alligator clamp on the two ends of the FSR, (different alligator clip for each tail strip).
2. Connect the other ends of these alligator clips to a digital multimeter. We used the DT830B model. Ensure no clips or wires are touching each other.
3. Turn the digital multimeter dial to the 2000k Resistance (ohms) setting.
4. With the black side touching table, place your finger lightly on the round FSR's surface and measure the results.
5. Now, press harder on the surface and record the results. The resistance should be much smaller than when the FSR was lightly pressed.
6. Lastly, test the FSR with different weight combinations (1.7 grams - 13.5 grams). We used 1.7 g and 4.5 g gaming mouse weights from Logitech. Record the results.

Results: There were some positive results when pressing the sensor. As shown in figure 14 below, when lightly pressing the FSR, there was a resistance reading of 19.07. When pressing the FSR more forcefully, the resistance reading dropped all the way to .75. However, when testing the different weights, we got less than ideal measurements. As shown in the table below, every weight combination placed on the FSR without the aid of pressing, gave a resistance reading of 1, which indicates infinite resistance. When a light tap was applied to the top of the weight(s), a fluctuation of resistance readings occurred. To make sure it wasn't just a glitch, we also put heavier objects on its surface, such as a bottle of water. This ended up still giving a reading of infinite resistance.

Figure 14: Reading resistance seen by the FSR. Light Pressing (left) and Hard Pressing (right)

Weight (grams)	0	1.7	3.4	4.5	9	13.5
Resistance 2000k	infinite	infinite	infinite	infinite	infinite	infinite

Table 11: Reading resistance seen by the FSR when different mouse weights are applied

Conclusion: If this was tested correctly, then based on the results, the round Force Sensitive resistor is not applicable to our needs. It appears it only detects squeezing and sudden applied pressure, rather than increasing weight on a flat surface. We will probably do some additional testing to see if we can somehow get the FSR to detect different weights.

5.1.1.2 Velostat Test

Objective: The objective of testing a piece of Velostat is to ensure that there is a way to interface it to our design that allows resistance and voltage to be measured across it. That way, we can measure the weight of a cup over the larger surface area that Velostat provides.

Procedure:

1. Cut out a 3" x 2" piece of Velostat from Adafruit.
2. Cut out two 4" pieces of medium, 3 ply conductive thread. Tape one piece of thread on the back and another one on the front, so that 4 ends are sticking out.

3. Alligator clamp the opposite ends of each thread with two different clamps
4. Similarly, to the FSR testing, connect the other ends of these alligator clips to the same digital multimeter, with dial setting on 2000k.
5. Place Velostat on either side on a flat surface. Now place an empty cup on the piece of Velostat. Measure the resistance.
6. Now fill the cup halfway with water and measure its resistance. You should read a much lower resistance than when the cup was empty.
7. Lastly, test the Velostat with different weight combinations (1.7 grams - 13.5 grams). Record the results.

Results: As shown in figure 15 below, the empty cup (on the left) had a resistance measurement of 11, while the cup filled with water (on the right) made the resistance drop to 5. When testing different weights, we encountered similar results to the cup measurements. When no object or weight was placed on the Velostat, there was a resistance reading of 15. The table below shows that, as more weight was added, the resistance steadily decreased to 8 for the 13.5-gram weight combination. What's interesting is that from 3.4 - 9 grams, there was a constant resistance reading of about 9.

Figure 15: Empty Cup (left) and Full Cup (right) placed on top of piece of Velostat

Table 11: Velostat Test

Weight (grams)	0	1.7	3.4	4.5	9	13.5
Resistance (2000k)	15	12	9	9	9	8

Conclusion: The constant resistance from 3.4 - 9 grams could indicate that Velostat is more useful for measuring ranges, rather than exact weight. Hypothetically, it could be summarized that if the resistance measurement was below 8, then there were at least 13.5 g left in the cup, while a measurement above 9 indicates that the cup would be nearing empty. Overall, Velostat seems like a much more viable option when compared to the FSR. It has behavior similar to what our goals are for the project, and it is very cost effective. Its potential to span large surface areas and have flexible interfacing make it the better choice for placing inside the smart coaster.

5.1.1.3 Analog Input with Voltage Division

As shown in the diagram below, we decided on choosing Velostat, as it provided more flexibility and worked as needed during the component testing phase. In the component testing phase, we looked only at how the resistance changed as pressure/weight was applied to the surface of the Velostat or FSR. However, we need to be able to have the microcontroller detect this change through the measurement of output voltage. To do this, we can set up a simple voltage divider circuit as shown below. The voltage measured between the resistor R and the Velostat will be connected to an input port pin on the microcontroller, which will then need to convert this voltage reading to a digital value. This can be done through the microcontroller's internal Analog to Digital Converter. As more weight and pressure is applied to the sensor, the voltage will drop because the resistor will simultaneously drop, V=IR (Ohm's law).

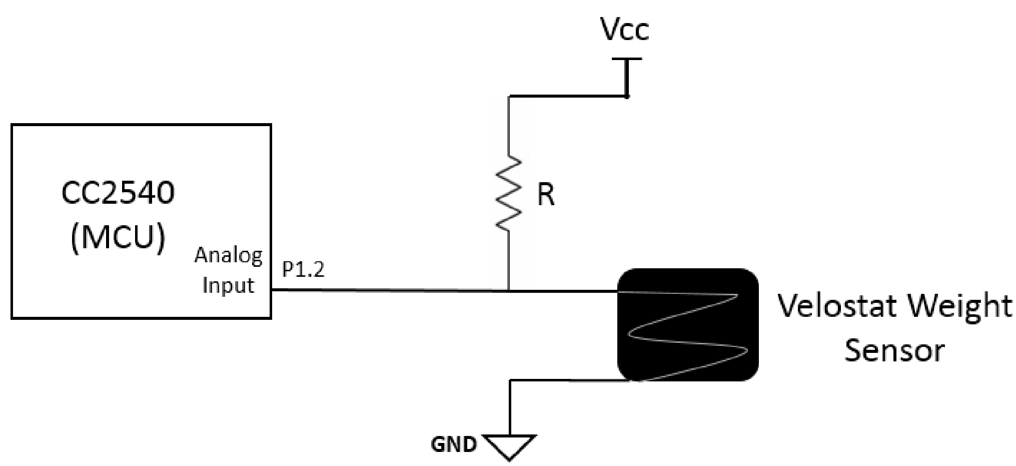


Figure 16: Weight Sensing Schematic

5.1.2 Charging Dock

In the research section, we determined that the best type of charging solution for Clever Coasters would be a charging dock. For this dock, we will use an off-the-shelf AC-DC switching power supply adapter. It needs to be equipped with short circuit protection as this was one of the safety considerations related to the charging dock. Other types of protection that will increase the safety of the system include overvoltage, overload (overcurrent), and overtemperature.

We will create a solid casing that the power supply will plug into. Inside, it will split the positive and negative terminals and attach them to two rails. The described setup is shown in figure 17 as a visual aid. The casing will be shaped to allow the vertical stacking of coasters. The two rails will go up two of the corners of the charging dock as exposed terminals. In the figure, the red rail represents the positive terminal and the black rail represents ground. When a coaster is placed within, two of the coaster's corners will be in contact with these terminals. The coasters will have their own exposed metal terminals in these corners. From there the power will be transferred into the charging circuit within the coasters and finally a well-regulated output will reach the LiPo batteries and charge them at a constant current and then constant voltage.

The choice of power supply will depend on the choice of linear regulator. If the TLV1117 is used, then the power supply will need to be able to output a substantial current. Assuming the system is going to have 6 coasters (as will be in our demonstration situation) it will have to output at least $800mA * 6\ coasters = 4.8A$ of DC current. If the LP2951 is used instead, then the dock will need to output a minimum of only $100mA * 6\ coasters = 0.6A$. The benefit of using the LP2951 is also that more coasters can be stacked on the same charging dock, assuming a more powerful power supply. A five-ampere power supply could support almost 50 coasters if they use the LP2951.

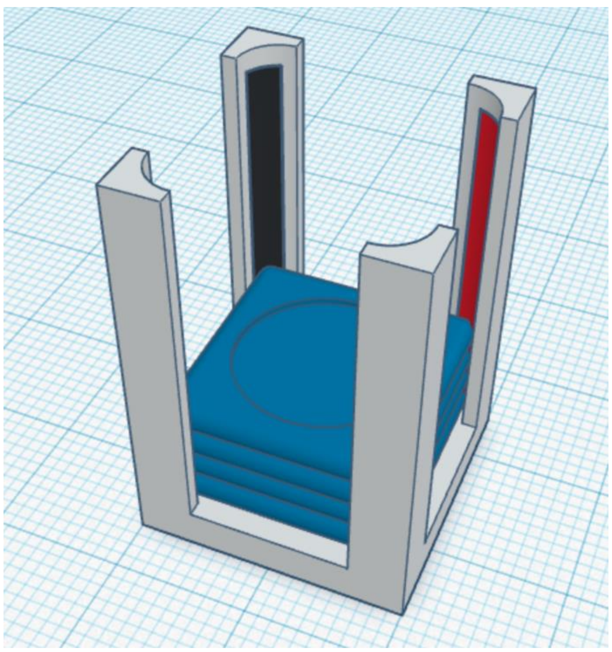


Figure 17: Charging Dock

The DC voltage that the power supply must output is also determined by the charging circuit. As discussed in the linear regulator section, it will have to be greater than the sum of the desired battery voltage, the dropout of the linear regulator, and the voltage drop across three 1N4001 diodes. These totals almost 7.9V for the LP2951 regulator and 8.7V for the TLV1117. Based on these values, it makes sense for the power supply to be able to source at least 9V DC to ensure that the regulators get the proper input voltage and the batteries get fully charged by their circuits.

5.1.3 Charging Circuit

The idea for the design of this circuit came from Texas Instruments literature on battery charging. In the document, there is a design for a single-cell Li-ion charger that does exactly what our circuit needs to do. It is, however, a slow charger that only charges at a current of 100 mA during the current limited phase (this is according to the datasheet for the LP2951 low dropout linear regulator, though the document claims it actually outputs a maximum of 160mA) [24][25]. This may or may not be a limitation of this circuit when considering multiple designs.

Based on the research on battery charging, it was determined that a linear regulator would be used inside a charging circuit in each coaster. Two linear regulators were chosen as potential candidates for this application: the LP2951 and the TLV1117, both from Texas Instruments.

5.1.3.1 LP2951 Charging Circuit Design at 100mA

The first option for a charging circuit was found in the TI literature for a Lithium-ion battery (it should work equally well for a LiPo). It uses the LP2951 in a variable output mode by connecting the shutdown pin to ground, the feedback pin to the voltage divider, the input pin to the power supply, the output pin to the diode, and the Vtap, sense, and error pins to nothing. This configuration is shown in figure 18. They use a trimpot trimming potentiometer as part of the design because the resistors have some variability (1%) due to manufacturing precision. Our resistors will have 5% error which is more than a 50kΩ trimpot can accommodate for. Because of this, we will need to test the resistors we use individually and choose the right ones so that the feedback circuit results in the proper voltage at the terminals of the battery. [25]

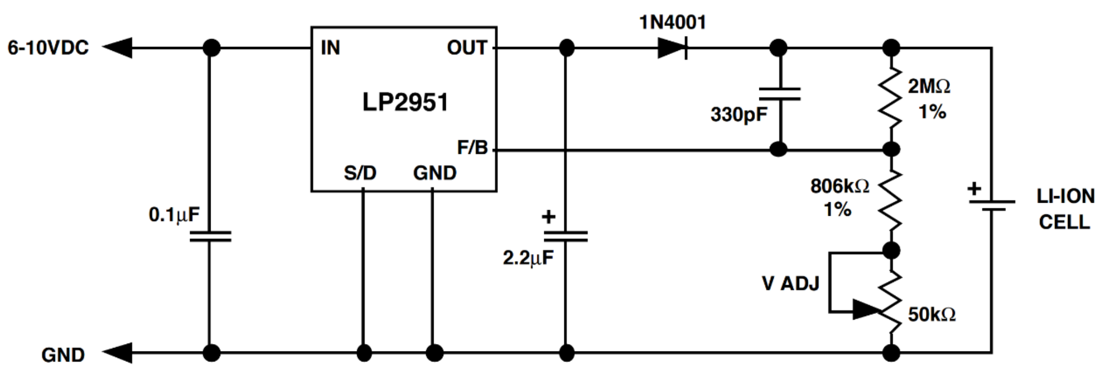


Figure 18: LP2951 Battery Charger

In our circuit, the only changes we make are that we replaced the 330pF capacitor with a 270pF one, and we replaced the 806kΩ resistor and 50kΩ trimpot with a 820kΩ in series with a parallel combination of 75kΩ and 22kΩ resistors. These were chosen experimentally because of the previously mentioned errors and will likely have to be chosen individually for each Clever Coaster. As a prototype, proof of concept system, this is fine, but for a mass-produced system, 1% resistors should be used and chosen in a more systematic and predictable way.

5.1.3.2 TLV1117 Charging Circuit Design at 800mA

The second option for a charging circuit is similar to the first, and most of the circuit is found in the datasheet of the TLV1117 linear regulator. The main difference would be that we would use a 1N4001 diode instead of 1N4002 and we would have another diode separating the output pin and the V_O node along with the feedback loop. Figure 19 shows the refrence schematic from the datasheet. [23]

This diagram also uses a trimpot potentiometer for R2 to make careful adjustments to the output voltage. Based on our calculations, we could use 2MΩ for R1 and 58.2kΩ for R2 which would be made up of two resistors: a 56kΩ resistor in parallel with a 2.2kΩ resistor. Some adjustment would be required because of errors in the resistances.

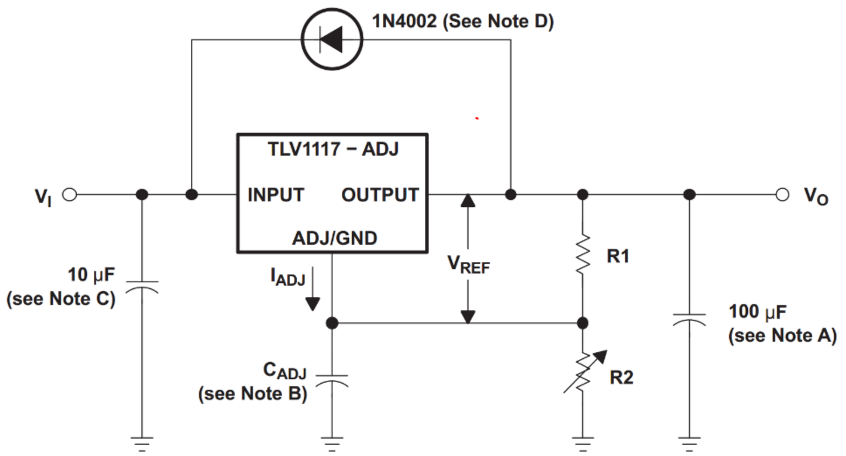


Figure 19: TLV1117 Battery Charger

The main advantage of this solution over that of the LP2951 is that it should be able to provide significantly higher current (800mA) to charge the battery which would get it charged in a fraction of the time. It would, however, require a much more powerful power supply for the charging dock. In order to choose between the LP2951 and the TLV1117, we purchased samples of each and tested them with breakout boards. The testing section explains the results and the reasons for choosing the LP2951 over the TLV1117.

5.2 Overall Schematic

Below is an overall first draft of what we intend to be our final PCB to look like, created in EAGLE. We intend to work on it more during the summer as our group was brand new to PCB design and there are almost certainly issues with this rough draft. However, it contains the basics of what we seek in our PCB: A PN532 NFC module and its antenna, connected with SPI wires to our MCU, the TI CC2540. Since this serves as our Bluetooth module as well, we needed to add an antenna to the PCB design for that as well, which we modeled from the recommended antenna design in the CC2540's datasheet. In addition to these two components, we have the force sensor, which uses a simple voltage divider and has through holes for us to attach the wires from our force sensor once the board is printed. There is also the charging circuit, which uses the LP2951 voltage regulator discussed above in addition to several other parts to form the charging circuit. The battery is also represented with through holes, since we will just be connecting the wires from the battery to the circuit once its printed similarly to the force sensor.

The PCB schematic as shown in Figure 20 is quite big due to all the antenna circuitry required, so it is tough to make out what fits where since the image gets compressed so much. As a rough guide, from the top of the page down, there is the PN532 antenna matching circuit, followed by the PN532 module itself, which is connected via virtual tags to the CC2540 as shown by the purple boxes. In between these connections is the force sensor diagram in gold, and next to that is the CC2540 module itself, with the other end of the virtual tags connected to ports P0_2 through P0_5, outlined in purple again. Above, or to the right depending on how you’re looking at the diagram, is the charging circuit outlined in gold. This is connected to the voltage source for the MCU, and by extension through the SPI wires, the NFC module as well. Finally, the Bluetooth antenna diagram in the gold box at the bottom most point of the picture. We will also take a closer look at each major component in the sections to follow since visibility of the overall schematic is so poor.

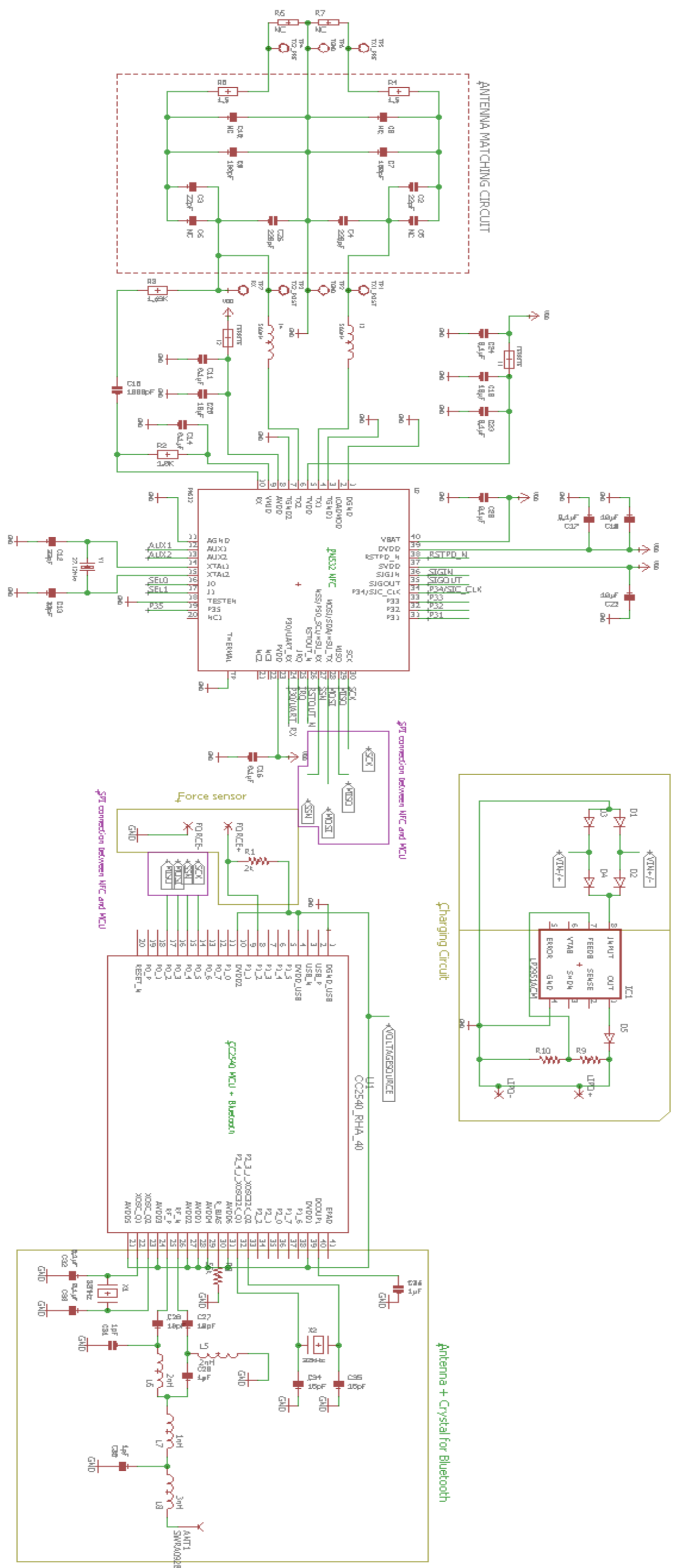


Figure 20: PCB Schematic

5.2.1 Antenna Schematic

The two antenna are the most complex portion of the PCB diagram, so we relied heavily on previous designs to figure out what should be done to get an antenna that performs well. For the PN532 antenna shown in Figure 21, we used the reference antenna provided by Adafruit for their PN532 shield reference diagram [14]. This antenna utilizes an antenna matching circuit which allows us to make last minute changes by swapping out discrete components vs ordering a new board, since the antenna capabilities come from the values of the capacitors placed on the board itself. Hopefully we won't need to do that but it is more flexible than TI's suggested antenna design shown in Figure 22, found in the microcontroller datasheet [15], which involves soldering an actual antenna part onto the board, which is connected by a simple circuit of capacitors and inductors before the signal reaches the microcontroller to be interpreted.

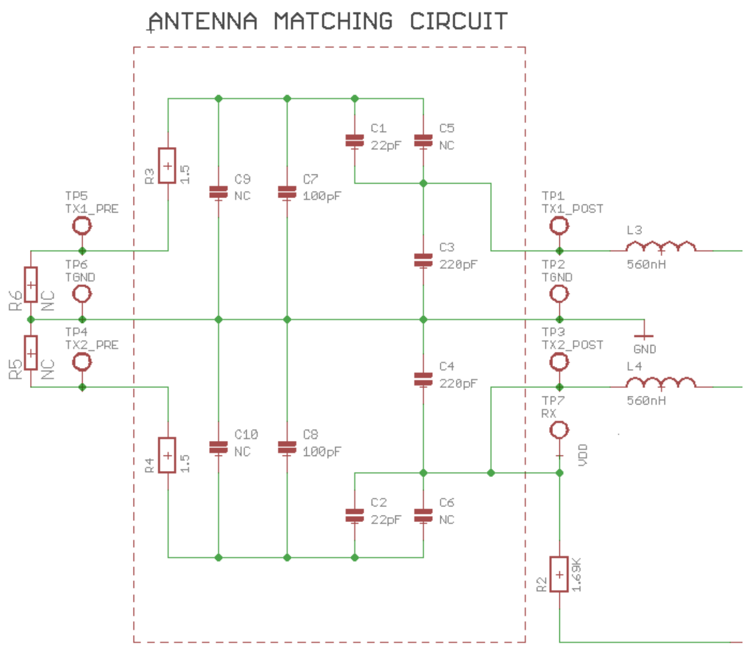


Figure 21: PNC Antenna Circuit

A simple but important inclusion to the schematic is the timing crystals for the MCU. Without these, it wouldn't be possible to run the MCU at a lower clock speed to enter idle mode and save battery, causing our power consumption to be extremely higher than anticipated. The CC2540 has two crystals that it uses to keep time, the 32-MHz crystal connected to ports XOSC_Q1 and XOSC_Q2 for the active operations, and the 32-kHz crystal attached to XOSC32K Q1 and Q2. These are all but required for our project to ensure that we get as long of a battery life as possible from each device.

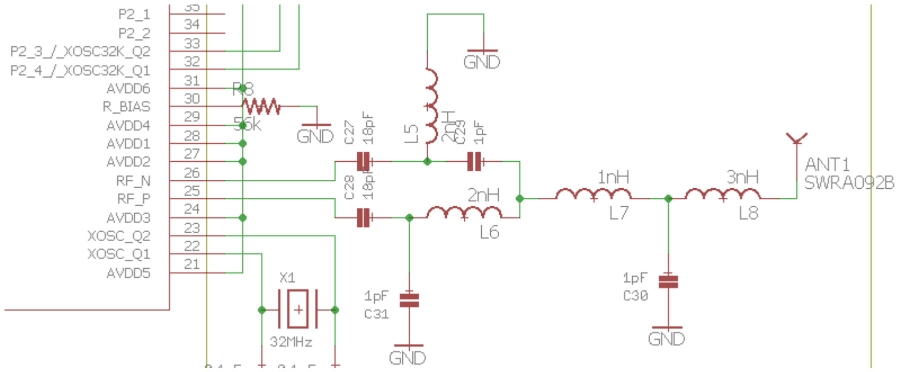


Figure 22: CC2540 Antenna Circuit

5.2.2 Force Sensor and SPI Connection

The area between the PN532 module and CC2540 houses the SPI connection between the two modules, as well as the schematic for the force sensor. To reduce clutter created by multiple wires crossing each other, we used tags to virtually link the SPI wires, which is why they look like they're disconnected but EAGLE knows that they're connected when we go to send the files out for fabrication. The CC2540 on the right uses pins 14-17 as its SPI dedicated pins [15] which is why those pins are selected as shown in Figure 23.

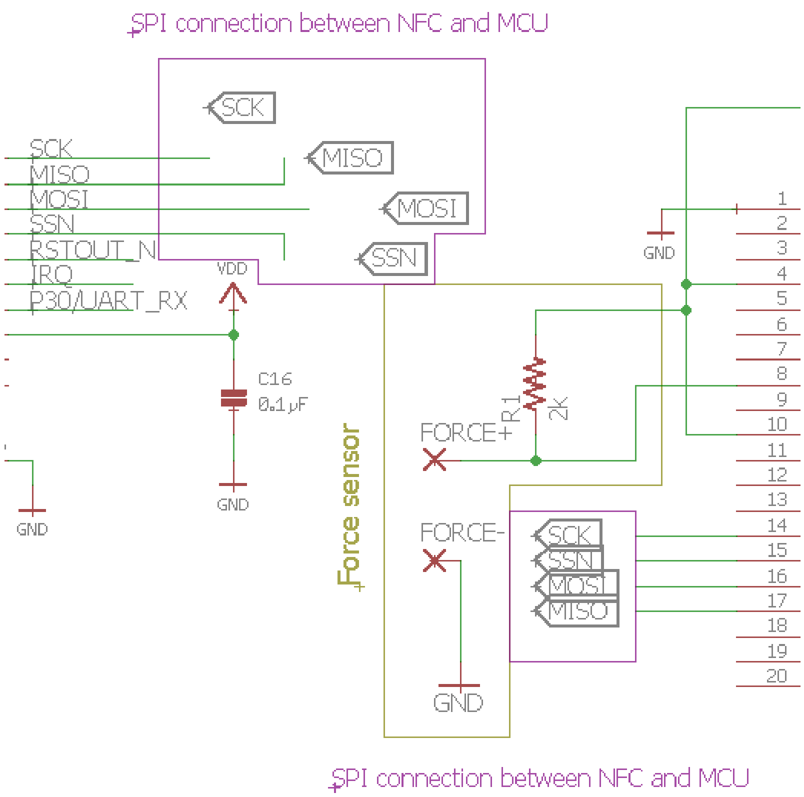


Figure 23: SPI and Force Sensor closeup

5.3 Software Design

Our software design must be able to have seamless regulation between all the interconnected devices, as well as error handling. Since we will need to continuously send updates and requests to receive the sensor data, some type of idle mode switch needs to be implemented into the design. A coaster should go into low-power mode when there is no cup on the coaster, or have a way to shut off the system completely. Likewise, all this sensor data needs to be analyzed then sent to a different user-interface.

5.3.1 Algorithm Description

Our code will be split into 2 main sections; sensor monitoring and wireless communication. Each task will be described to show the general logic of the functions that go into accomplishing these two goals. Assume that all initial setup for functions such as setting the initial starting weight to the max possible to ensure that the first read in weight value is set to the current weight, range of feasible cup weights, etc. are all preset.

There is also a special mode that can be activated by a specific signal from a Bluetooth-connected tablet. This mode is used for calibrating the weight sensor in a given coaster and saving the different possibilities of cups and their weight levels in the coaster's memory. This is especially important in the prototype system because everything will be assembled by hand and there is much room for error in the sensitivity of the weight sensor.

5.3.1.1 Sensor monitoring

Purpose: To periodically read and calculate the weight of the object placed on the coaster.

- START
- Read in the sensor data and determine the weight of the cup
    - If the weight is reasonably constant for at least 5 seconds, less than the last measured weight and within the acceptable range for a cup (e.g. something lighter than a cup was placed on the coaster), set the new current weight variable to this weight.
    - If weight is not reasonably constant or within acceptable range for a cup, restart function
    - If current weight is more than last measured weight and within acceptable range for a cup, (e.g. the cup got refilled) set the new current weight variable to this weight.

- If the current weight is within the drink low/empty range, start the wireless communication function to relay this information back to the hub.
- If the current weight is above the drink low/empty range, place MCU into sleep mode for 10 seconds to conserve power
- END

5.3.1.2 Wireless communication Bluetooth

Purpose: To relay information back and forth between the hub.

- START
- Wait for interrupt from MCU or timer to awaken Bluetooth from sleep mode
- Establish connection with most recently used table hub if possible
    - If unsuccessful, reenter sleep mode
    - If unsuccessful because hub is busy with other devices, try again
    - If successful, send a status report to the hub, with details of estimated drink percentage, last refill time, etc.
- If the button was the cause of the interrupt, send a request to the main hub to take appropriate action
- If a new table hub discovery from NFC was the cause of the interrupt, pair with the hub's Bluetooth address.
- Reset timer countdown
- Disconnect Bluetooth and reenter sleep mode
- END

5.3.1.3 Wireless communication NFC

Purpose: To connect to the Bluetooth module with the hub's Bluetooth.

- START
- Poll the NFC module for a nearby tag
- Upon tag discovery, switch NFC to active mode, read the tag and extract the Bluetooth address contained
- Wake up the Bluetooth and pair with the hub

- Upon successful connection, return NFC to low power mode
- Restart loop
- END

5.3.1.4 Calibration Mode for Weight Sensor

Purpose: To setup a coaster's memory such that it has reference values for weights of cups it can expect to be placed on it. This is a manual operation mode that requires a person to activate and control.

- START
- Special signal from tablet interrupts MCU. It enters calibration mode.
- Flashes LEDs in a specific pattern to indicate calibration mode.
- A weight type is selected in the tablet and sent to MCU.
- MCU detects average value of weight sensor and stores it in memory.
- MCU sends a Bluetooth message to the tablet to indicate it is done with that particular weight type.
- New weight is placed on MCU.
- New weight type is chosen.
- Loop continues until tablet gives signal to end.
- MCU goes back to standard operation mode.
- END

5.3.2 Software Block Diagram

In the diagram below (figure 24), a general overview of how the control flow states will function in our clever coaster system is laid out. It is important to keep each software component module and easily modifiable for different tablet systems. A big part of our software design will come from how our wireless communication components will be integrated into the overall design.

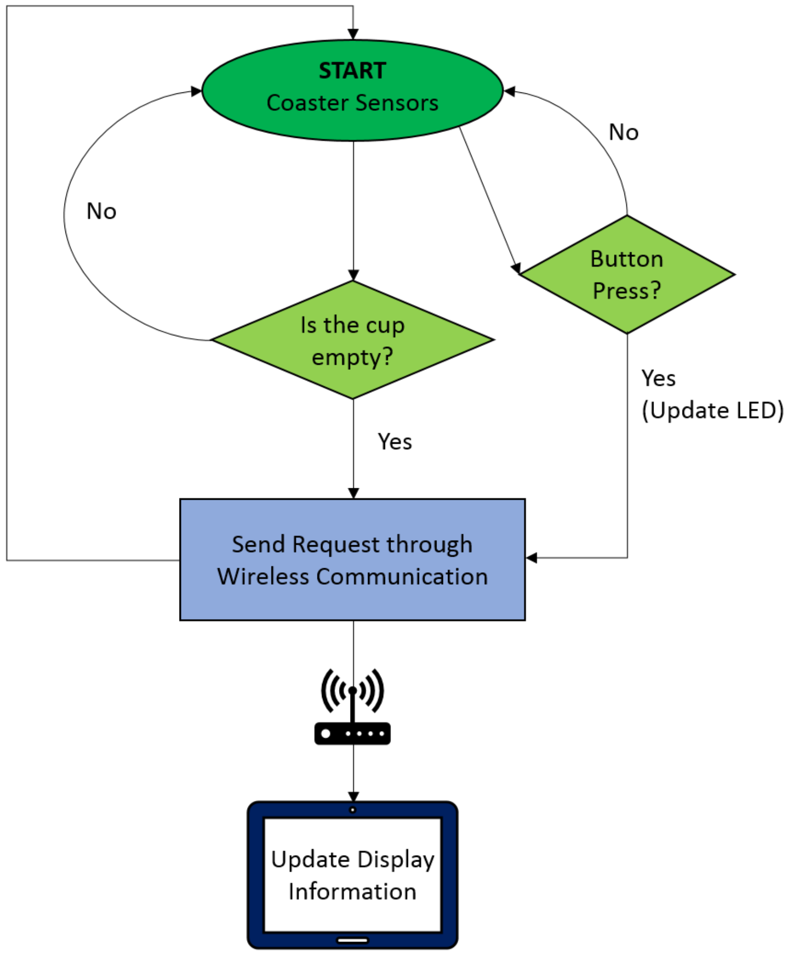


Figure 24: Software Block Diagram

5.3.3 User Interface Design

When designing the user-interface that the customer sees, it is important to follow principles of consistency and other guidelines that increase user experience of the product. There shouldn't be a plethora of options all on one screen, as this would overwhelm the user. Likewise, it should be quick for the user to reach their desired destination within the program, and the number of clicks to get there should be minimized.

The user-interface that the waiter or waitress sees should also follow these guidelines, so that they can clearly see which tables and cups need to be attended to. We will look at existing restaurant apps to gain some insight on popular design themes and conduct a few usability studies to ensure overall accessibility.

5.3.4 Development board testing

The first order of business is to test the microcontroller and ensure that it is functioning as intended. Since we can't design, print and ship our PCB to solder on our microprocessor of choice at this stage, we used a prebuilt microcontroller to do preliminary testing. It should provide an accurate assessment of any issues that we will run into with the real device in the future. Our test device is the MSP430 since we have experience coding it. Since its made by TI a lot of the same syntax for code will carry over and has a USB input for easy debugging. As shown in Figure 25.

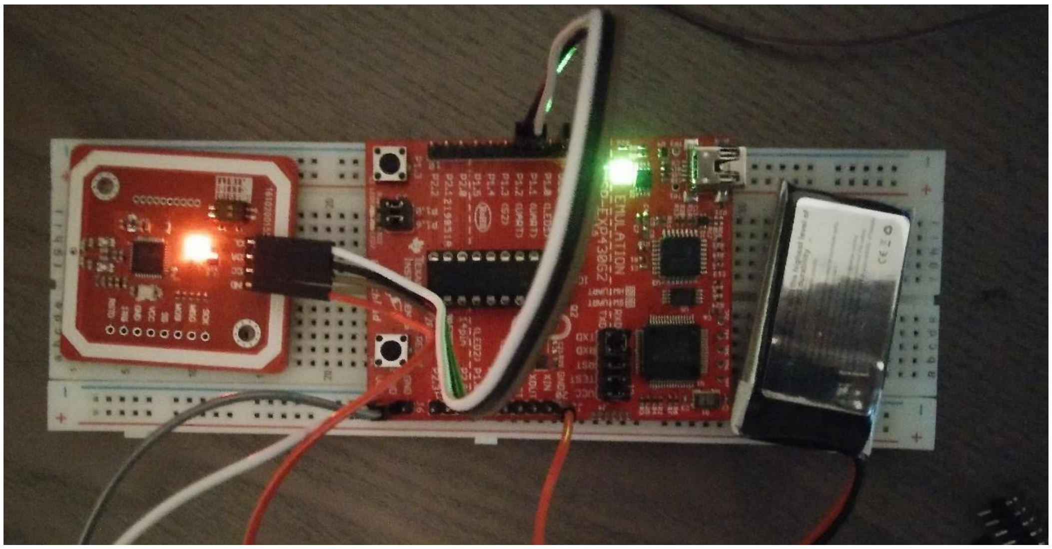


Figure 25: Development board testing

To verify that the NFC module was working correctly with the development board, we ran a simple program to test that the NFC was communicating properly over the SPI bridge. To do this, we placed an NFC tag, like the one shown in Figure 20, and formatted it, were met with the following prompt shown in **Error! Reference source not found.** that confirmed that the NFC module was working. Unfortunately, we couldn't get our hands on a TI CC2540 development board to ensure that our chip works, but since TI makes both the MSP430 we used to test and the CC2540 we will use in our final build, we can safely assume that the code between the two will be roughly similar.

Figure 26: NFC Software Output

5.4 System Housing

The system housing is not the most important aspect of this project, nor is it particularly related to Electrical or Computer Engineering, but it is still essential to discuss, albeit briefly. The coasters, after all, are products that need to be self-contained, even as prototypes, if they are to work correctly.

5.4.1 Functional Design

Primarily, the Clever Coaster casing needs to be functional. It needs to allow and facilitate the coasters in their ability to perform their required functions and meet the requirements as set forth in the project description.

5.4.1.1 Wireless Communications

To allow for wireless signals to travel into and out of the coasters, the material that they are made of must not block those wireless signals. The main danger to blocking radio waves is conductive material such as metal. Because of that, we would not use metal to make the casing of the coasters. Materials like plastic do not block radio waves, so they would make a good choice.

5.4.1.2 Durability

The coasters must be able to withstand general use and take some hits such as being dropped on the floor. Metal would obviously be the most durable material for this purpose, but because of wireless communication requirements and a desire to decrease costs, it doesn't make sense. Wood would not be a good choice for this either because it would not do well with getting wet. Plastics make the most obvious choice in this case.

The two main plastics that would make prototyping simple via the use of 3D printing are PLA (Polylactic Acid) and ABS (Acrylonitrile Butadiene Styrene). PLA is environmentally friendly and biodegradable. It is also easier to work with because it does not require a heated bed. ABS, on the other hand, is more durable and more flexible. It is easier to fix by gluing and it is more resistant to heat. For durability ABS makes the most sense.

5.4.1.3 Charging

The casing also must allow for charging of the battery in the coaster by putting it in the charging dock. For this to work, it has to have exposed terminals on the corners. These must be conductive and so will be made of metal. The rest of the casing must be non-conductive so the current doesn't go through it when it goes through the terminals.

5.4.1.4 Dust and Water Proofing

Dust proofing and water proofing are part of the IP44 standard. We chose IP44 as a minimum, but in reality it would be better to aim for IP66 or IP67. This would be complete dust protection and an ability to withstand temporary flooding of water or immersion into 15cm to 1m of water respectively.[7]

The main way to prevent ingress of water and dust will be to use some type of plastic, which doesn't absorb water, to make most of the case. Another way to incorporate water and dust proofing into our design is by using a type of clear-coat nail polish so that water can slide right off the coaster surface. Instead of nail polish, different types of glue could be used to seal every opening in the plastic casing, the seam between the top and bottom parts and the areas around the exposed metal terminals.

5.4.1.5 Weight Sensing

The problem of having a solid, durable case and still being able to sense the weight of the cup is a difficult one. The weight sensor will be inside the casing, but the plastic used in the coaster casing will not be so flexible that the weight of a cup will make a large difference in its deflection. This means that the sensor will not get a good reading if the whole casing is solid.

To combat this, we came up with an idea for the indented part of the casing, the center part where a cup is placed, to be connected to the rest of the casing by a rubber sheet. This way, the center will be flexible relative to the rest of the coaster and most of the weight of the cup will be transferred to the Velostat weight sensor. It will still require some calibration for each coaster. In a prototype environment, this will have to be done individually, by hand. In a mass-production environment, this would still have to be done individually, but it could be a pre-programmed automated process.

5.4.1.6 Cost

Cost is one more consideration for the choices on how to make the coaster casing. Once again, metal would be a prohibitively expensive material for this purpose. Plastic is very cheap and as such is best for cost. In regards to the manufacturing process, large scale production would be best served by injection molded plastic, but for prototyping 3D printed material should be fine. Furthermore, 3D printers are commonly used to print ABS, which seems to be the best material for the Clever Coasters system.

5.4.1.7 LED State Indicators

To inform the user and waiter about the state of a user's cup or request status, an RGB LED can be used. A blue light can indicate a near-empty cup, a green light can detect a waiter request, and a red light can detect a do not disturb mode. These LEDs should be placed in the coaster design in a way that isn't too distracting to the user, but visible enough to instantly notice color changes when the button(s) are pressed. We will want to make sure the LEDs are inside the casing, but still visible through an open perimeter on some area of the coaster design.

5.4.1.7.1 RGB LED Test

Objective: The objective of testing the RGB LEDs is to ensure that they are not malfunctioning, and that each color works when connected to a 3 V source.

Procedure:

1. Connect all four pins of the RGB LED to a solderless breadboard (in parallel) to each other.
2. Clamp a wire to the positive and negative terminal of a 3 V lithium cell. We used the Sony CR2032 model.
3. Now connect the wire connected to ground to the ground pin of the LED (in series).
4. Connect the other positive wire to the Red LED pin (make sure its in parallel to the ground pin). Make sure the correct light emits.
5. Repeat Step 4 for the Green and Blue LED pins.

Results: As shown in figure 28, the correct lights emitted when the RGB LEDs was connected to a voltage source.

Conclusion: To make effective use of the RGB LED, we will need to test it with a potentiometer or the microcontroller to ensure that there is seamless switch between colors. This will allow us to program different states into the coaster (green for button press, blue for water, etc.).

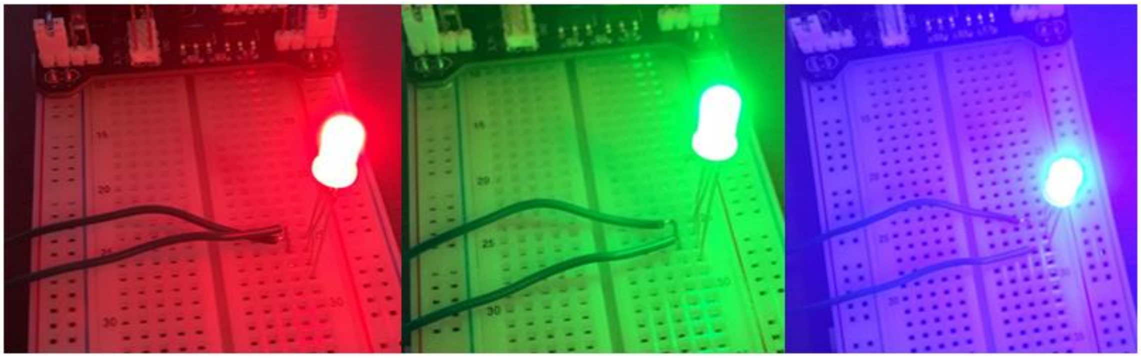


Figure 27:(Left to right) RGB LED with red, green, and blue lights

5.4.2 Aesthetic Design

Based on the functional design requirements, the aesthetic design will be quite flexible. Because 3D printing will be used, it should be simple to create specific shapes and sizes. That said, it makes sense to create a standard shape for the coaster. We decided to use a flat square as the basic shape. It will have beveled edges and rounded corners. In addition, it will have a depressed center area where the cup can be placed. This center area will be connected to the rest of the coaster via a rubber sheet that will be glued to it and to the coaster walls. This will allow for the weight sensor to detect changes in pressure.

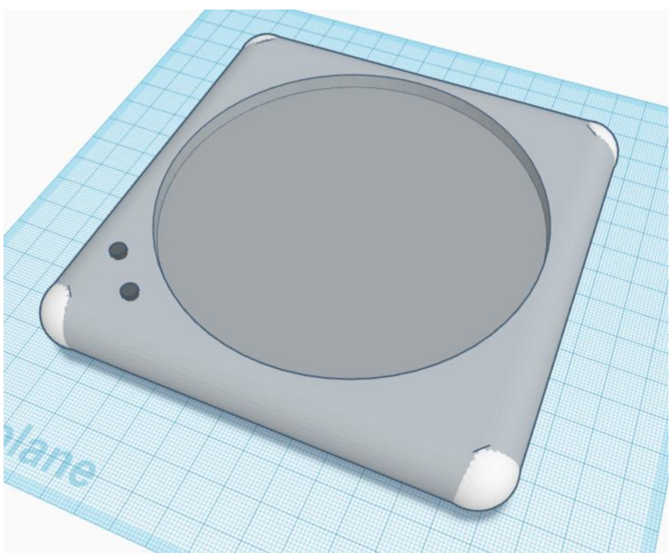


Figure 28: Coaster Design

Figure 29 is a CAD mockup of the coaster design. The darker area in the center is the depressed area in which a cup will be placed and under which the force sensor will be placed. The white corners are the metal terminals that will allow the coaster to charge when it is placed in a charging station. The dark circles on the left corner are two buttons for various functions of the coaster.

6.0 System Testing

While component research is very important to selection because it allows one to qualify many components based on their data sheets and specifications, it does not always lead to the best components for a given job. Since we are not experts in sensors, batteries, chargers, and other electronic components, the best we can do is disqualify certain components and select others for testing. Sometimes it is unclear how exactly a component will behave in each application because the datasheet is not focused on that specific purpose.

Testing allows for each component to be verified for its ability to do what is required in each system. It can reveal certain shortcomings of the components that a theoretical view based solely on numbers will not. Because of this, we shall test both the hardware and software related components.

6.1 Hardware Testing

Individual hardware testing of each of the main components early on will allow us to gauge if we made the right choice and to make any last-minute decisions. It is especially important to test the weight sensors and microcontroller, as those will be the components that need to work flawlessly when integrated into the PCB design. For all testing, we will limit it to indoors, as we will only account for table inside the restaurant (not patio). Therefore, we do not need to worry about sun damage and other outside factors.

Since we are on the fence about which method of weight sensing we should use, we have narrowed it down to two options: The Force-Sensitive Resistor (FSR) and the Velostat. Depending on which method demonstrates the desired behavior best (lower resistance as more weight is added), we will conclude on which to move forward with.

6.1.1 Voltage Divider with Velostat Test

After choosing components, we can now begin testing some circuit designs, including the one with weight sensing. Since this will be our main functionality in our clever coasters, it is important to make sure the design works in a consistent manner and measures with accuracy. The difference between a full cup, near-empty cup, and empty/no cup should be apparent when reading the voltage changes. An algorithm can be later determined in Senior Design 2 that looks at what thresholds should be used in detecting full or empty within the digital readings.

Objective: The objective of testing the voltage division circuit is to ensure that it can accurately detect the voltage output between the Velostat and resistor. If the voltage drops when more weight is added, than this means our test is successful.

Procedure:

1. Cut out a 6” x 4” piece of Velostat from Adafruit and fold it in half.
2. Cut out two 4” pieces of medium, 3 ply conductive thread. Tape one piece of thread on the back and another one on the front, so that 4 ends are sticking out. Make sure to coil them so that they resemble a resistor.
3. Alligator clamp the opposite ends of each thread with two different clamps
4. Connect a Vcc voltage of 3.3 V to the breadboard and its corresponding ground terminal.
5. Put a 2 kilo-ohm resistor in series with Vcc and then put the other end of the resistor in series with the velostat thread end (opposite side for terminal connected to ground.
6. Now turn on Vcc (In our case, we took Vcc from a MSP430 microcontroller.
7. Using a digital multimeter, measure the output voltage between the resistor and velostat and record the results. (Use 20 V setting)
8. Repeat step 7 for an empty cup and a full cup placed on top of the velostat sheet.

Results: Our results proved to be successful in detecting the voltage difference between a full and an empty cup. As shown in figure 30 below, the empty cup measured a voltage of 1.87 V, while the full cup measured a voltage of 1.2 V. That’s a difference of .67 volts. The next step would be to test out different coilings and configurations of the Velostat to get even more accurate readings of the cups. Likewise, in Senior Design 2 we will focus on coding up the ADC readings and deciding on threshold values (especially for varying cup sizes and detecting 30% fullness).

Figure 29: Weight Sensing Test with Velostat and Voltage Divider

6.1.2 Charging Circuit Test

Objective: To test the two low-dropout linear voltage regulators and their circuits to ensure that they can provide a well-regulated voltage and the proper current to charge the LiPo battery.

Procedure:

1. Connect the charging circuit on the breadboard as shown by the schematic in the Charging Circuit Section.

2. Connect the power supply to the input and ground pins.
3. Connect the multimeter to the ground and output locations where the battery would be connected (the output location here is not the output pin of the linear regulator, but the node after the diode which is connected to the output pin.
4. Set the multimeter to measure voltage and turn on the power supply.
5. Check that the voltage is correct. Ensure that it is stable by increasing the input voltage from the power supply within specifications for the linear regulator.
6. If voltage is stable, either change the multimeter to measure current or use another multimeter for the task. Ensure that current does not exceed 1C for the selected battery.
7. If voltage is not correct, change the resistors until it is the correct voltage. This may be required because the resistors have an error of 5% which is significant for this application.
8. Plug the battery into the circuit.
9. Ensure that the voltage and current are still within bounds for proper operation.
10. Turn off the power supply and check if the voltage and current change to that of battery connected to the voltage divider circuit.
11. Repeat the above steps for each linear regulator (in this case only one other) and its circuit.

Results: As shown in figure 31, the LP2951 linear regulator was able to produce the correct voltage of 4.19V, which is close enough to 4.2V for our application, and a slightly higher than expected current. The current was still well within bounds. In the off-state, it reverted to battery voltage and the current drained by the feedback loop resistors was only 5.1µA.

There is no figure for the test of the TLV1117 linear regulator, but the testing of it did not produce desired results. It did not produce a stable voltage, but closely followed that of the power supply input. Its current was also higher than desired.

Conclusion: The results of the test were very good for LP2951, but surprisingly poor for the TLV1117, which was a promising candidate linear regulator for the charging circuit. Upon review, two issues were revealed with its design that are a problem for the Clever Coasters application.

First, the data sheet mentions that "The device passes its bias current to the OUTPUT pin. The load or feedback must consume this minimum current for regulation or the output may be too high"[23]. This is a problem for our application because when the battery is fully charged, it should not have any current flowing into it. Since our resistors are large, they do not fully consume the bias current and the output is too high. This could be fixed by inserting an additional, smaller resistor between the output and the diode to consume the 5-15mA of bias current.

Unfortunately, the second problem could not be fixed as easily. The reference voltage in the TLV1117 is between the output and the ground/adj pin. Because there's a diode in between those, and the diode voltage can vary, this does not lead to a stable enough system for LiPo battery charging.

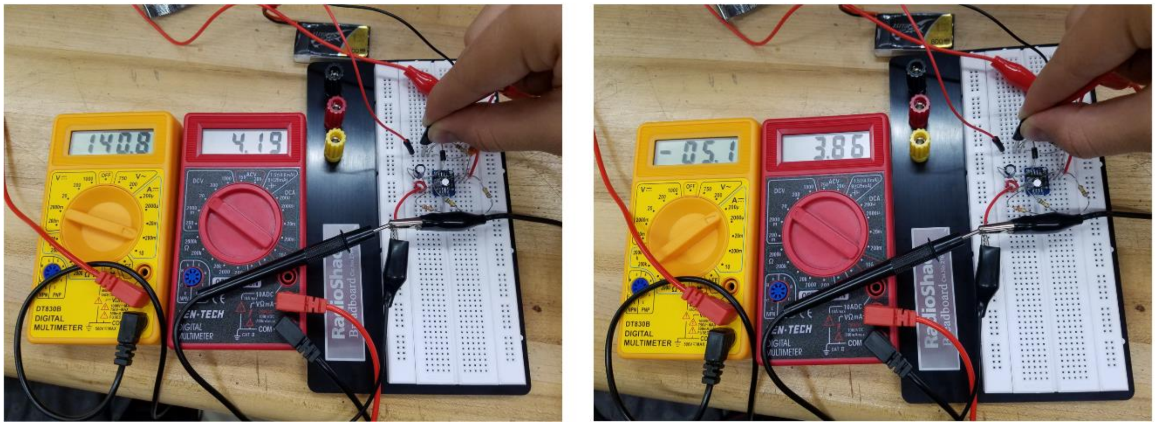


Figure 30: Charging Circuit Test with LP2951

The LP2951, on the other hand, has its reference voltage between the feedback and ground pins which allows the voltage after the diode (instead of just before the diode) to be very precisely controlled [24]. When changing the input voltage on the power supply, the output remained stable at 4.19V, as needed. For these reasons, the LP2951 was chosen as the linear regulator for Clever Coasters. We will still need to test and modify every circuit's resistors because of error in the resistance values.

6.2 Software Testing

Software testing is vital to ensure that individual software components are working and will not cause issues with integration in the future. With robust documentation, and plenty of example projects and libraries to guide us, software testing can be done easily before being put into real use. We can use emulation software if needed to test before transferring the code to the physical microcontroller.

6.2.1 Microcontroller test

To test our microcontroller code, we could run simulated testing. This gives us the ability to run code in a controlled scenario, even allowing us to step through the code line by line and analyze what happens in the registry and trace stack calls. However, simulated testing is less precise when involving other components, especially in our case where the components are made by parts from different manufacturers. In our case the most difficult to emulate would be the interaction between the NFC and CC2540 because the NFC is made by NXP Semiconductors while the CC2540 is made by TI. However, for simple simulations such as the sensor input which we can easily assume, or outgoing Bluetooth data that we have full control over, simulation becomes a great option to use before beginning testing on the fully integrated hardware. This simulated testing would give us a reasonable amount of confidence that if issues were to appear during the hardware test so that we know that the issue most likely lies with the hardware instead of the software.

There are several tests we could do to assess how well the coaster has had all its components integrated, and upon discussion our team has decided on the following set of tests that the coaster will need to be able to perform to pass a complete software integration test. It will need to be able to send a 1000-character string with 100% accuracy via Bluetooth to the table device. It will need to be able to connect with an NFC card and read the data off it with 100% accuracy assuming perfect contact. It will need to be able to exit sleep mode, reestablish its Bluetooth connection, send 10 kB of data to the table hub, and re-enter sleep mode all within 3 seconds. It would need to read the same values within a 10% error margin for 3 objects of different weights to be placed 3 times each on the force sensor. Finally, it would need to register every button click in a series of 15 clicks.

7.0 Administration

Things will get hectic as important deadlines start to approach and components need to be replaced. It is crucial that during our project, we keep an organized schedule, manage our personal time well, and keep finances in check when purchasing components. We should always buy in bulk to save on shipping costs, and keep in good communication to avoid any conflicts.

7.1.1 Estimated Project Budget and Financing

For each coaster, we will require a micro processing unit to run the code responsible for analyzing the data from the sensors, controlling the LEDs, and connecting and wirelessly sending data to the table device. The display device will be like a table device, with a full screen and ability to interact with. However, our team already owns devices suitable for the job. We budgeted for extra PCB boards to accommodate for any mistakes we make on our first designs or while soldering components.

With our demo system of six coasters, two table devices and a display device, we need six batteries, for the coasters. The outer shell for the coaster will need to be of a material that's not only waterproof, but also heat and cold resistant, pliable enough for our pressure and touch sensors to get a good reading, and fit all our components inside without being flimsy. In addition to all this, the material cannot reduce the signal strength of the signal to the point where it can no longer satisfy our requirement for minimum connectivity range. The LEDs will be of several different colors to allow user feedback with a variety of flashing patterns. They also serve to indicate to the server at a glance which drink needs to be refilled, and but not be so bright that customers might find them obnoxious.

Description	Quantity	Estimated Cost (each)	Total Cost
MCU for Coaster	6	$4	$24
Table/Display Device	3	$100 (already have)	-
Wireless adapter	2	$40	$80
Wireless Components	6	$5	$30
PCB	8	$12.50	$100
Battery pack	6	$6	$36
Coaster outer shell	6	$10	$60
LEDs (pack of 100)	1	$5	$5
Weight sensor	6	$7	$42
Button	6	$1	$6
Charging Station	1	$25	$25
Miscellaneous components	-	-	$25
TOTAL			**$499**

Table 12: Component Costs for Entire System

7.1.2 Project Schedule

A lot needs to go into the overall project, including several reports, presentations, and a working prototype demo. Figure 32 below, shows an estimated timeline of where we need to be at each month until December, when the final project is due. Luckily, we have an extended amount of time over the summer to tweak our designs and continue research before Senior Design 2.

A lot needs to go into the overall project, including several reports, presentations, and a working prototype demo. The figure below shows an estimated timeline of where we need to be at each month until December, when the final project is due. Luckily, we have an extended amount of time over the summer to tweak our designs and continue research before Senior Design 2.

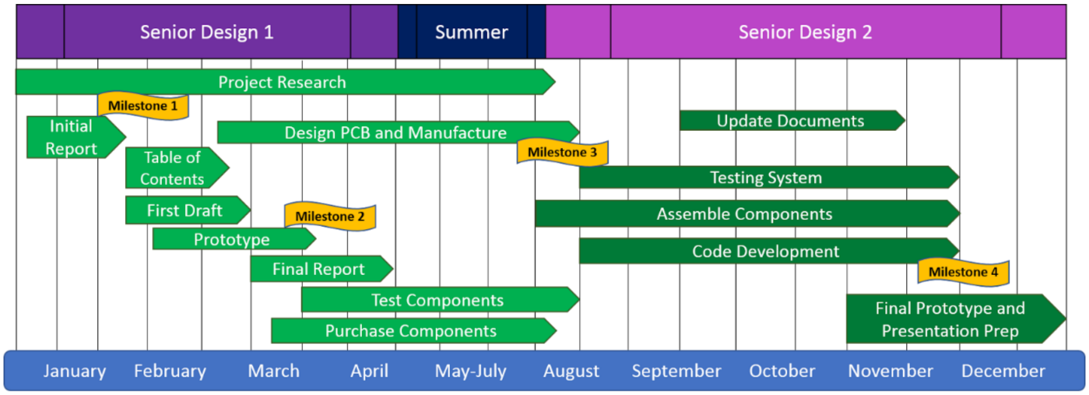


Figure 31: Project Schedule and Milestones

- Milestone 1 was our agreement on a project idea
- Milestone 2 was finishing our design and final report
- Milestone 3 will be finishing and manufacturing the PCB design, and testing all parts
- Milestone 4 will be finishing the final working product

8.0 Project Summary

The Clever Coaster system is an interesting project that we wish to continue working on. Many engineering challenges were faced during the research and testing phase of our project. Time management seemed to be a big issue, because most of our team had busy schedules, personal issues, and workloads to prioritize. Senior Design 1 turned out to be more intensive than we expected. Learning from these challenges and mistakes will allow us to continue meaningful progress over the summer and start off strong come Fall semester when we begin Senior Design 2.

Another challenge we faced is funding the project and receiving support and aid in how to design the clever coasters. We were given no sponsors to choose from this semester (a deeply held belief created from conversations with past Senior Design students), so funding the project from our pockets was difficult, as a lot of pricey components that need to be purchased in bulk made costs add up quickly. Top that with the fact that we are all computer engineers and so there was a big learning curve in figuring out PCB design and the exact requirements for the project.

8.1 Conclusion

Overall, we have considered every component needing to be optimized during the development phase of our project (Senior Design 2). This includes the Velostat sensor, which needs to be further researched and tested in order to optimize the design. Likewise, our schematic is a very rough draft version of what we expect to use and purchase in the fall. We are right on schedule in terms of milestones and project calendar.

All in all, we believe that our project will benefit the end users by providing an intuitive and cost-efficient system available for continuous usage. We predict our biggest challenge to be getting all the wireless components to connect seamlessly together, and employing an energy-efficient system that doesn't waste too much power when idle. We also want to focus on making a very accurate weight sensor that fits into the overall coaster design well.This project integrates computer engineering and electrical engineering evenly, allowing us as a team of 3 CpE students to meet PCB requirements. We look forward to continuing this project in the Fall and hope to give our full undivided attention to it. Now that we are more familiar with the project scope and objectives, it will be much easier to focus and finish out senior year strong. We plan on having very open and free schedules next semester so that we can catch up and earn the A+ that we know we can achieve.

9.0 Appendices

9.1 References

[1] "Bluetooth Low Energy | Bluetooth Technology Website." Bluetooth. N.p., n.d. Web. 27 Apr. 2017. <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/low-energy>.

[2] Tanenbaum, Andrew S., and David Wetherall. Computer Networks. Boston: Pearson, 2014. Print.

[3] Admin, Kanda. "WiFi, Bluetooth or ZigBee Wireless." Kanda Electronics Blog. N.p., 29 May 2012. Web. 27 Apr. 2017. <https://www.kanda.com/blog/wireless/wifi-bluetooth-zigbee-wireless/>.

[4] Introduction to Piezoelectric Force Sensors. N.p., n.d. Web. 27 Apr. 2017. <http://www.pcb.com/Resources/Technical-Information/Tech_Force>.

[5] Woodford, Chris. "Piezoelectricity - How Does It Work? | What Is It Used For?" Explain That Stuff. N.p., 20 July 2016. Web. 27 Apr. 2017. <http://www.explainthatstuff.com/piezoelectricity.html>.

[6] Ada, Lady. "Force Sensitive Resistor (FSR)." Using an FSR | Force Sensitive Resistor (FSR) | Adafruit Learning System. Adafruit, n.d. Web. 27 Apr. 2017. <https://learn.adafruit.com/force-sensitive-resistor-fsr/using-an-fsr>.

[7] "IP Rating Chart." DSMT.com. N.p., n.d. Web. 27 Apr. 2017. <http://www.dsmt.com/resources/ip-rating-chart/>.

[8] Frenzel, Lou. "What's The Difference Between IEEE 802.15.4 And ZigBee Wireless?" Electronic Design. N.p., 22 Mar. 2013. Web. 27 Apr. 2017. <http://electronicdesign.com/what-s-difference-between/what-s-difference-between-ieee-802154-and-zigbee-wireless>.

[9] "Near Field Communication." Android Developers. N.p., n.d. Web. 27 Apr. 2017. <https://developer.android.com/guide/topics/connectivity/nfc/index.html>.

[10] Lukas, M. "Arduino Controlled Smart Coaster." Instructables.com. Instructables, 12 May 2016. Web. 27 Apr. 2017. <http://www.instructables.com/id/Arduino-controlled-smart-coaster/>.

[11] Cavailer. "Siduri – an Arduino Control Smart Coaster." Instructables.com. Instructables, 12 May 2016. Web. 27 Apr. 2017. <http://www.instructables.com/id/WaiterLite-Coaster/>.

[12] "Open Source Free Point of Sale." Floreant POS. N.p., n.d. Web. 27 Apr. 2017. <http://floreant.org/>.

[13] Ziosk - Industry Leading Tabletop Ordering, Entertainment and Payment Solutions. N.p., n.d. Web. 27 Apr. 2017. <http://www.ziosk.com/>.

[14] Ada, Lady. "Adafruit PN532 RFID/NFC Breakout and Shield." Overview | Adafruit PN532 RFID/NFC Breakout and Shield | Adafruit Learning System. N.p., n.d. Web. 27 Apr. 2017. <https://learn.adafruit.com/adafruit-pn532-rfid-nfc/overview>.

[15] "2.4-GHz Bluetooth® Low Energy System-on-Chip." Ti.com. Texas Instruments, June 2013. Web. 27 Apr. 2017. <http://www.ti.com/lit/ds/symlink/cc2540.pdf>.

[16] Mishriki, Fady. "Design Considerations in Modern Wireless Power Transfer Systems: Coil Geometry." PowerbyProxi. N.p., 27 Nov. 2016. Web. 27 Apr. 2017. <https://powerbyproxi.com/2014/09/design-considerations-modern-wireless-power-transfer-systems-coil-geometry/>.

[17] "What’s the Best Battery?" Basic to Advanced Battery Information from Battery University. N.p., 21 Mar. 2017. Web. 27 Apr. 2017. <http://batteryuniversity.com/learn/archive/whats_the_best_battery>.

[18] "The RoHS Directive." Electrical and Electronic Waste - Environment - European Commission. N.p., n.d. Web. 27 Apr. 2017. <http://ec.europa.eu/environment/waste/rohs_eee/index_en.htm>.
[19] "RoHS Guide." RoHS Compliance Guide: Regulations, 10 Substances, Exemptions, WEEE. N.p., n.d. Web. 27 Apr. 2017. <http://www.rohsguide.com/>.
[20] "What Material Should I Use For 3D Printing?" 3D Printing for Beginners. N.p., 22 July 2016. Web. 27 Apr. 2017. <http://3dprintingforbeginners.com/filamentprimer/>.
[21] "WEEE: Waste Electrical and Electronic Equipment." Export.gov - WEEE Information. N.p., 3 Jan. 2017. Web. 27 Apr. 2017. <http://2016.export.gov/europeanunion/weeerohs/weeeinformation/index.asp>.
[22] "How NFC Works." How Near Field Communication Works - NearFieldCommunication.org. N.p., n.d. Web. 27 Apr. 2017. <http://nearfieldcommunication.org/how-it-works.html>.
[23] Texas Instruments Incorporated. TLV1117 Adjustable and Fixed Low-Dropout Voltage Regulator (Rev. L) (n.d.): n. pag. Web. 27 Apr. 2017. <Instruments Incorporated>.
[24] Texas Instruments Incorporated. LP295x Adjustable Micropower Voltage Regulator with Shutdown (Rev. I) (n.d.): n. pag. Web. 27 Apr. 2017. <http://www.ti.com/lit/ds/symlink/lp2951.pdf>.
[25] Simpson, Chester. "Battery Charging." (2011): n. pag. Texas Instruments Incorporated. Web. 27 Apr. 2017. <http://www.ti.com/lit/an/snva557/snva557.pdf>.

9.2 Permissions Requests

Adafruit Industries <support@adafruit.com> 10:46 AM (6 minutes ago)

to me

totally ok.

cheers,
adafruit support, phil

On Wed, Apr 26, 2017 at 10:24 AM, Rubba Ashwas <support@adafruit.com> wrote:

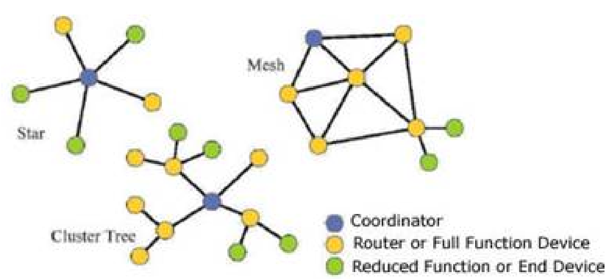
> contactname : Rubba Ashwas
> email address : rubba.ashwas@gmail.com
> Message : Hello, I am working on a senior design project in which we are making smart coasters with pressure sensors. Can I request permission to use some of your FSR diagrams (with proper citation of course) in our report?
>
> Thank you,
> Rubba
> Client IP: 72.188.99.89

Permission for image

o support@kanda.com

Hello,

I am working on a senior design project for school involving wireless communications. I would like to request permission to use the following diagram in my report (with proper citation of course).



Rubba Ashwas
University of Central Florida
Computer Engineering
rubba@knights.ucf.edu

Permission of Images

From RubbaA to cavailer on 2017/04/26 07:20 am

Hello,

I am working on a project for my senior design class in which we are trying to make a smart coaster. Can we have permission to use your images/diagrams in our report? (with proper citation of course)

Thank you,

Rubba

Image Permission

From RubbaA to mlukas on 2017/04/26 07:19 am

Hello,

I am working on a project for my senior design class in which we are trying to make a smart coaster. Can we have permission to use your images/diagrams in our report? (with proper citation of course)

Thank you,

Rubba