The Beer Pong Mat

By

Nishita Amberkar

Keith Bevans

Spencer Gallagher

Final Report for ECE 445, Senior Design, Spring 2023

TA: Zicheng Ma

03 May 2023

Project No. 20

Abstract

We designed and constructed a portable beer pong mat that can be used to ensure that cups are correctly placed and filled as well as keep score of the game. Our mat is capable of accurately measuring the weight of each cup to within 10 grams, and features precisely measured indents to ensure that each cup is placed exactly where it should be and that each team has an identical cup array. In addition, our mat is portable, being reducible in size by up to 75% and being battery-powered to negate the need for a wall outlet. The mat also features an LCD display on each side as well as two buttons that are used to keep score of the game. Despite having to make a number of design changes over the course of the project, we were able to build a functional mat that is portable, accurate, and intuitive to use.

Contents

1. Introduction

You would be hard pressed to find a college student or graduate in the United States who has never played a game of beer pong. This game, in which players take turns attempting to throw a ping pong ball into cups of beer across a table, has over the years evolved from a simple drinking game to a ubiquitous American pastime. People of all ages play this game, with or without alcohol, at parties, tailgates, and even on their iPhones, where users can use the GamePigeon app to challenge their friends to a virtual game of pong over text. Professional tournaments are even held for the game, with the largest such tournament, the World Series of Beer Pong (WSOBP), regularly having over a thousand participants. Given that it is commonly played in an informal setting, beer pong, similar to games of wiffle ball or pickup basketball, is often the source of heated arguments amongst participants. Throughout an average game, cups may be shifted, spilled, or tilted in ways that give one team an unfair advantage over the other, and with no impartial official to make final decisions on what to do in these situations, players are often left feeling slighted. In addition, especially after many games are played in a row, it can be easy to lose track of the score, how many games each team has won, and whose turn it is. For such a prevalent game to have so much unnecessary unpredictability is unacceptable, and our goal was to ensure that future beer pongers are presented with an even playing field, so that each game is fair and every victory is that much sweeter.

To address these problems, we constructed a mat that indicates where to place each cup and whether each cup has the correct amount of liquid. In addition, our design will indicate to players the current score of the game. This mat is able to be laid flat upon a 6' folding table, the typical surface used for a game of beer pong. In addition, this mat is portable, so that users can bring it wherever they feel a game of pong must be played. The placement of cups is indicated through the use of LED rings, which also light different colors to indicate whether the correct amount of liquid is in each cup. Our mat senses whether a cup has the correct amount of liquid through the use of pressure sensors placed under the cups. Indentations in the mat ensure that the cups are placed where they are supposed to be. A mini-LCD screen on each side of the mat displays to both teams the score of the current game. The components used in this project can roughly be divided into four distinct subsystems: the cup sensors, which consisted of the force-sensitive resistors (FSRs) used to measure the weight of each cup; the user interface, which consisted of the LCD scorekeeping display, the buttons used to control the scorekeeping display, and the LED rings used to indicate whether a cup had the correct amount of liquid; the control subsystem, which consisted of a microcontroller used to read data from the buttons and FSRs and send data to the LEDs and LCD displays; and the power subsystem, which consisted of a 9V battery and linear regulator used to power the LCD display, LED rings, and microcontroller. Figure 1 below is a block diagram illustrating these subsystems and how they connect to each other to form the overall project, while Figure 2 is a diagram demonstrating an aerial view of how we wanted our final product to look and be used. Think of our solution as being to beer pong what a robot plate umpire is to baseball. By regulating the game through the use of technology, we eliminate the possibility of human error and ensure a fair game for all players.

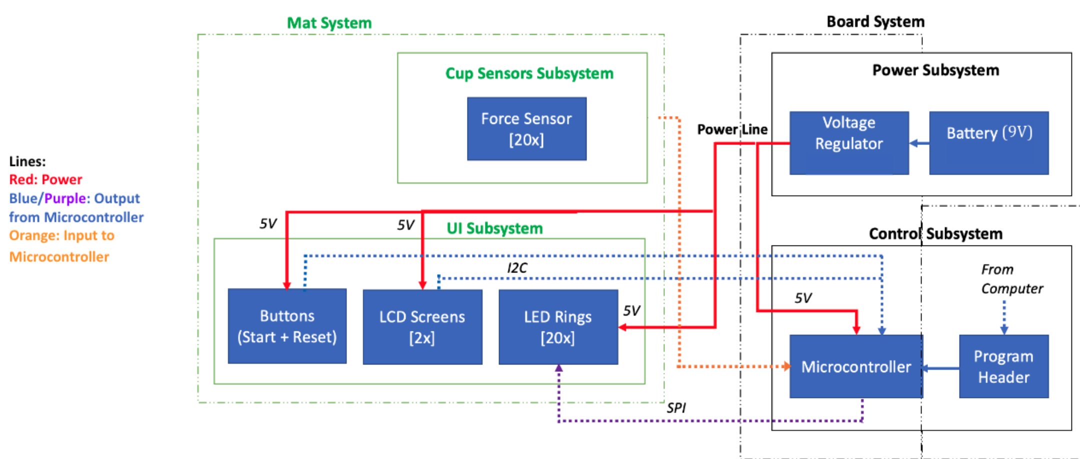

Figure 1: Original Block Diagram

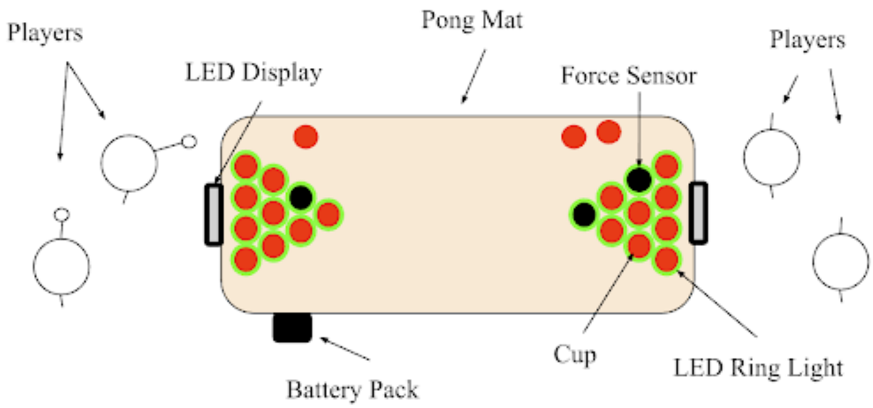

Figure 2: Aerial View of Design

1.1 High Level Requirements

The three main characteristics we felt our design had to exhibit in order to successfully solve the problems stated are as follows:

- Accuracy: Our design had to incorporate tight tolerances to ensure the proper placement and filling of cups, or else it will merely reinforce the problem that it is supposed to solve. To be considered accurate, our mat had to:
    - Update the LCD displays and LED rings no more than a second after a button was pressed or a cup was filled, respectively
    - Have cup indentations that were precisely measured and the same on both sides of the mat
    - Feature force sensors with tight tolerances so as to ensure the weight of each cup was in an acceptable range
- Portability: One of the most appealing aspects of beer pong is its ability to be played wherever there are cups and balls. Our design should be portable enough to follow even the most adventurous ponger to wherever his or her chosen playing location may be. To be considered portable, our mat had to be:
    - Reducible in size by at least 50%

    - Usable without a wall outlet (battery-powered)
- Intuitiveness: Players need to be able to focus on perfecting their shot or defending against bounces, not figuring out how to operate the mat. Our design needs to be extremely simple for users in order to improve the game of beer pong and not serve as an anchor on the boat of fun. To ensure a positive user experience, we set out to make a mat that:
    - Had a simple, easy to understand user interface
    - Would function with minimal user input (i.e. minimal button presses)

The purpose of this report is to provide a detailed description and overview of the design and functionality of our project. We will also highlight the challenges we faced during development, the strategies we used to overcome them, and our main takeaways from the project. The chapters of this report will provide an overview of our project including our design and implementation, hardware, and software components. We will also discuss the cost, testing and validation, as well as our conclusions.

2. Design

2.1 Mat System

The mat system is the body of our project and includes technological features as well as non-technological features. The two main subsystems of the mat were both built for user experience. The first and most important one being the cups and sensors and the second one being the screen and buttons. The third subsystem includes our non-technologic solutions to our problems because as a group we found that we did not need technology to solve each problem.

The mat itself was two yoga mats put together, each with 10 circular cutouts on each end of the mat for each cup subsystem. Unfortunately for our demo we only had four LEDs due to unforeseen shipping issues, but each LED was placed on top of the mat. In between the yoga mags there were wires, force sensors, and our waterproof wrap. Each cup cutout contained one force sensor and one LED, then once all wired and bundled together, we placed the waterproof wrap over everything. On each end of the mat there were two buttons and an LCD screen. The LCD screen was glued down to the mat while the buttons stuck out from in between the mats.

2.1.1 Cup Sensor Subsystem

Each cup sensor consists of one force sensitive resistor (FSR). The FSRs are used as weight sensors to determine whether a cup is filled with the proper amount of liquid. This is done using a voltage divider circuit, as shown in Figure 3. Each FSR will have one pin tied to ground and the other tied to a resistor of constant resistance that is connected on the other side to the positive power supply. The voltage in between the FSR and fixed resistor will be connected to an analog read pin on the microcontroller. When a cup is placed on the FSR, its resistance will go down, thus increasing the voltage drop across it, decreasing the voltage drop across the fixed resistor, and changing the voltage read by the microcontroller. It is in this way that the microcontroller will be able to know when the weight of the cup changes, and through careful tuning, we can determine a voltage range that corresponds to the acceptable weight range of the cup. The force sensitive resistors are passive components, and therefore do not require a dedicated connection to the power subsystem.

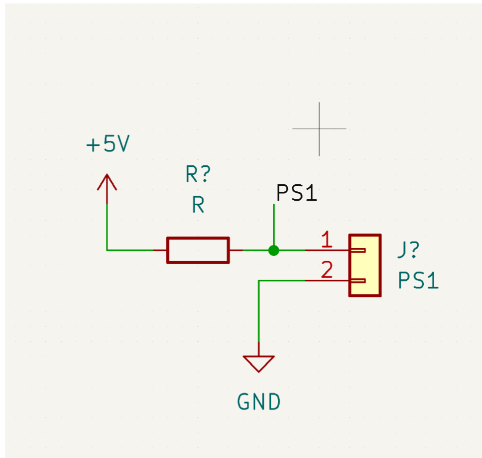

Figure 3: KiCad Diagram of Voltage Divider Circuit

While FSRs are effective, cheap, and simple, they are not nearly as precise or accurate as a typical load cell. This is acceptable for our purposes, as we only need the force sensitive resistor to confirm whether the cup is within a wide weight range (137 g to 250 g, as stated earlier) rather than give precise readings on the weight of the cup. However, we still would prefer our force sensors to be as accurate as possible, and in order to do so, we must carefully select the resistance of the fixed resistor in the voltage divider circuit to maximize the precision of the voltage reading sent to the microcontroller. In order to do so, we want to have the widest possible voltage range representing the range of forces on the FSR (5V in our case, as the DC bias across the divider circuit will be provided by our power supply). Typically, an FSR will have infinite or very high resistance (> 10 MΩ) when experiencing no force and a resistance of down to 20Ω when experiencing maximum force. If the fixed resistor in the voltage is, say, 20Ω, then the voltage read by the microcontroller when the FSR experiences no force will be 0V while the voltage read by the microcontroller when the FSR experiences maximum force will be 2.5V. This can be understood by using the previous facts given about an FSR's resistance and referring to Figure 4, which shows a diagram of a typical voltage divider circuit. When the FSR experiences no force, it has such a high resistance that practically the entire voltage drop will be across it and practically none will be across the fixed resistor, meaning that V_{out}, which will be read by the microcontroller, will be 0V. When the FSR experiences maximum force, it has a resistance of about 20Ω, and if the fixed resistor is chosen to be 20Ω, then the voltage read by the microcontroller will be 2.5V, as there will be a 2.5V drop across each resistor ([20Ω / [20Ω + 20Ω]] x 5V). Clearly, then, 20Ω is not an acceptable value for the fixed resistor, as this gives a voltage range of 0V to 2.5V, which is only half of the range we have available. If we instead gave the fixed resistor a value of 10kΩ, then the voltage read by the microcontroller when no force is experienced by the FSR would be 0V, for the same reason as above; however, the voltage read by the microcontroller when the FSR experiences maximum force would be 4.99V ([20Ω / [20Ω + 10kΩ]] x 5V). This gives us a voltage range of 0V to 4.99V, which is clearly superior to the range given by the 20Ω resistor, and sufficiently wide for our purposes.

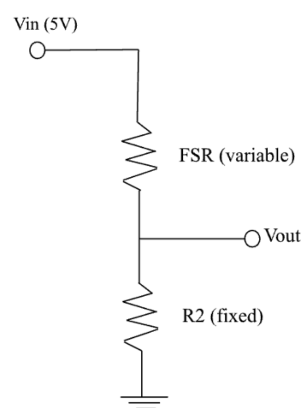

Figure 4: Voltage Divider Circuit Diagram

2.1.2 UI Subsystem

The UI subsystem consists of the LED rings, LCD screen that displays the score of each team and the current/next player, and two buttons - Reset and Stop. The LCD screen is power supplied 5V and connected to the microcontroller which decides the characters (max: 32) printed, and updates the variables score and player number respectively. The component of the LCD screen is connected to the potentiometer which provides the contrast resolution of the blue background and white characters. On the other hand, the buttons Reset and Stop are connected alongside the LCD screen. The button Reset restarts the values to default values i.e. player 1 and score 0. The button Stop stops the values from getting updated and ends the game. The buttons are connected to the microcontroller and power supplied by 5V.

The LED rings are active components, and each one must be connected directly to the power subsystem (see Figure 5). In addition to these two connections, each LED will also be connected to a digital output pin on the microcontroller (see Figure 6) that can feed it serial data and tell it when to turn on/off and what color to turn to. A program written to the microcontroller will effectively read the weight of each pressure sensor through the use of the aforementioned voltage divider circuits and, using this information, tell the LED what to do. When a cup is within the acceptable weight range, the corresponding LED ring will shine green; otherwise, it will shine red.

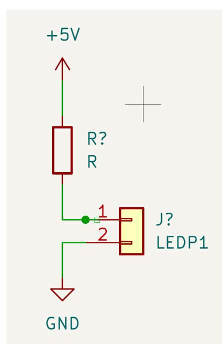

Figure 5: KiCad diagram of LED power connection

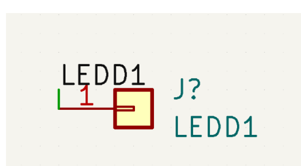

Figure 6: KiCad diagram of LED data connection

Each force sensor is connected to its respective LED ring with individual pins to account for easy data transfer from the analog readings to the LED color readings. Each LED and FSR is one item in an array to use in a for loop. We then run through all the force sensors using a loop and check the readings from the force sensors one by one. If the readings from the force sensors are within the acceptable range defined, the LEDs turn green else they turn red. Considering that the LEDs are RGBs, the shift from red to green is an easy change in the position of 255.

The software for the buttons and LCD used conditional statements followed by code to execute. It first checks if the button is in a high or low state. If the button was being pressed, the code would then update the LCD screen using libraries and data sent through I2C.

2.1.3 Non-Electrical Components

The project we built included a lot of technology, but due to the simple nature of beer pong, we realized that we did not need technology to solve all of our proposed problems.

One of our big problems we set out to solve was how cups move when they should not. Our simple, not technological, solution was to cut individual cup holes on the top mat to keep the cups steady. The second solution was in the form of a high-level requirement of portability.

Many common electric beer pong tables are full tables and are not very portable due to weight and size. Using yoga mats that are easy to fold and roll and are lightweight helped add a layer of simplicity to our pong mat system.

The last part of the non-electrical system was waterproofing. The mat contains a lot of wires and connections along with parts that have exposed soldering. This means to protect our circuit design, we needed to make sure that there was an absolute zero amount of water that could contact our wires. Our solution to this was to use a plastic wrap that went over the wires on the bottom mat, and below the top mat. A very simple solution was how we were able to ensure protection for the parts of our design.

2.2 Board System

The board system refers to all subsystems and components which reside on our custom PCB. This includes our control subsystem and the linear regulator, which is a portion of the power subsystem.

2.2.1 Power Subsystem

The power subsystem is the foundation of our project. Without power, we would not be able to run any other subsystem. The power will be coming from a 9V battery in which we connect with our PCB which then delivers that power elsewhere in our project. As shown in the block diagram (Figure 1), we use a voltage regulator to ensure a drop down from 9V to 5V. Being the foundation, it is crucial that we supply enough power to our project in order to get everything to run smoothly, for example if we supply too little power, the LEDs will not be bright enough to give the user any feedback. In Figure 7 below we show how the power is provided to the system.

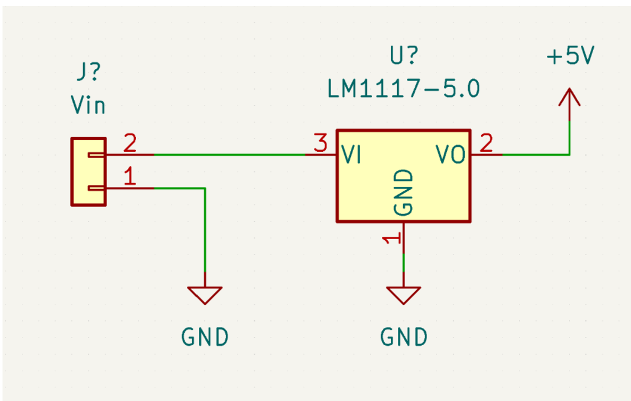

Figure 7: KiCad diagram of voltage regulator circuit

2.2.2 Control Subsystem

The control subsystem is the headquarters of our project. We need to make sure everything goes smoothly. Our PCB will need to connect everything together, whereas our microcontroller will take in data from various sources and send outputs to the user interface parts. Our microcontroller was used on a development board, and was programmed through a USB port with Arduino IDE.

Data tracking was done in the memory section of our microcontroller. The force sensors act as the data provider, and once the data is received, we then send signals from the microcontroller to LEDs and the LCD display to act as our UI. The SDL and SCL pins were used for data sending from the microcontroller to the LCD screen for our interface.

2.2.3 Development Board

With our project needing software, we needed a microcontroller to handle all of the data processing. But, with some unforeseen circumstances, our original microcontroller was unable to run the necessary code for our product to execute. Late in the semester this led to a panic, and we opted for a brisk switch to a development board to handle our data I/O.

Our development board used an ATMEGA2560 chip and had more than enough data input and output that led to the microcontroller. The software was burned into the chip by using a USB port. Once we were able to use the ATMEGA chip, our software was able to run smoothly. The Arduino IDE also allowed us to read values and calibrate each force sensor properly, quickly send data to each LED ring, and read buttons to send data to the LCD screens.

3. Design Verification

Our design was verified using both unit testing and evaluating the success of our high-level requirements.

3.1 Success Metrics and Satisfaction of the High-Level Requirements

Based on our high-level requirements i.e. portability, accuracy, and intuitiveness, we evaluated the success of our project.

3.1.1 Portability

With regards to portability, the goal was to ensure that the mat could be carried and played anywhere thus requiring for the mat to be reducible in size and not needing any electrical outlets to power the components.

We were successfully able to reduce the size of the mat by 50% and use batteries to power the components present.

3.1.2 Accuracy

Accuracy was one of our main high-level requirements where we ensure that the game of beer pong is played correctly by calculating the acceptable range of the liquid in the cup, the accurate positions of the cups, and calculating and displaying the scores of each team using a LCD screen.

We were successfully able to satisfy all the high-level requirements.

Cup Placement:

Measurements were made to ensure that the cup formations were the exact same on both sides. Considering that the map was two feet wide, each cup having a bottom diameter of

47.6 mm, and top diameter of 79.4 mm, the cup indentations were cut precisely on both sides where the center of each indentation is 80 mm apart from the next and 70 mm higher than the last. This is shown in Figure 8 with a triangle placement of cups along with the measurements used to ensure accuracy.

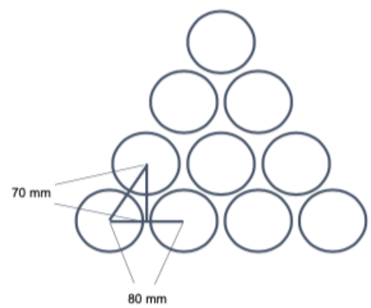

Figure 8: Cup Indentations

Acceptable Range of Liquids:

The FSRs were able to detect small changes in weights resulting in the LEDs shifting colors from red to green and vice versa. The LEDs shift red when the weights are outside the acceptable range of 137 - 250g and the green in the acceptable range. The response time with regards to the LEDs and the FSRs is low thus providing instantaneous results. Figure 9 below showcases the upward trend between the resistor values and the weight with the green box highlighting the acceptable range for the liquids.

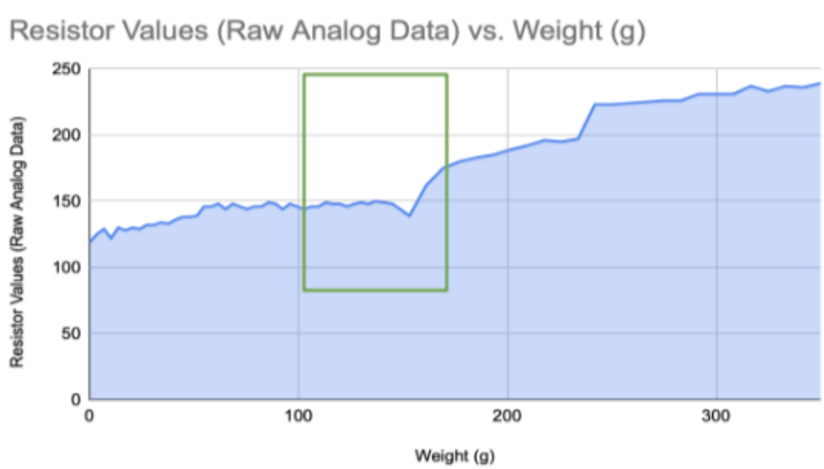

Figure 9: Analog data vs weight graph for the FSRs

Low Latency:

The data transfer between the LEDs, LCDs, buttons, and the MCU should be less than one second to ensure the proper synchronization amongst the components of the game.

To ensure low latency rates, a testing program was run which calculated the start time and the end time for each component, and then the difference between them to calculate the required latency. Each component was less than one second.

3.1.3 Intuitiveness

The game of beer pong, on its own, is simple and easy to understand. As a result, we wanted our mat to satisfy the same purpose. The mat should be simple, easy to use, with the minimal number of button presses to update the score or restart the game, and visual cup cut outs so users understand the cup placements.

Essentially, we had two buttons per team to update the score (colored green) and restart the game (colored red). Additionally, we had cut outs for each side of the table.

3.1.4 Others

There were two other requirements that we initially overlooked:

1. Waterproofing: all sensitive electronics must be waterproofed to prevent damage to the equipment or harm to the users
2. Adequate power supply: power supply should be capable of powering the project for at least an hour

Waterproofing:

To account for the spills and ensure that the cups remained in position throughout the game, the mat cutouts were cut to the size and the components were plastic wrapped to further ensure water proofing of the components and the mat.

When tested by dropping liquids, the components remained functional and we were able to continue playing the game.

3.2 Unit Testing

As and when we got each of our components, we unit tested them to ensure proper functionality.

We made sure that our 9V battery with a linear voltage regulator was enough to power each component - LCDs are turned on and displaying text, LEDs shine at their maximum brightness, and buttons change their respective values.

Additionally, we made sure that each component was functional such as the LED turning red/green based on varying ranges of liquids using the force sensors, the buttons restarting and incrementing the score on the LCD screen, the LCD screen displaying the team number and score for each team.

Each of these components were successfully functional during unit testing. Additional information on the requirement and verifications for each subsystem are present in Appendix A.

4. Costs and Schedule

Due to the numerous parts needed for our project, it became very costly. We needed 20 FSRs, 20 LED rings, two LCD screens, and numerous resistors, capacitors, and diodes for our PCB. In addition, we had to purchase a development board late into the project once we realized our microcontroller would not work, as well as a number of non-electrical components (yoga mats, cups, etc.) to put together the final product. Table 1 below lists all the parts we purchased for this project, as well as their unit cost, quantity, and total cost.

4.1 Parts

Table 1: Parts and Costs

Part	Manufacturer	Retail Cost ($)	Quantity	Actual Cost ($)
Yoga Mats	All in Motion	$18	2	$36
24 x 5050 RGB LED Rings	AdaFruit	$16.95	17	$288.15
WS2812B 24-bit RGB Ring LEDs 5-pack	DIYMall	$25	1	$25
Dev Board	Arduino	$21	1	$21
Connector Cables - 120 pack	ELEGOO	$6.98	2	$13.96
0Ω Surface Mount Resistors	Vishay/Dale	$0.109	10	$1.09

1 kΩ Surface Mount Resistors	Vishay/Dale	$0.109	30	$3.27
10 kΩ Surface Mount Resistors	Vishay/Dale	$0.109	25	$2.73
1 μF Surface Mount Capacitors	Kyocera AVX	$0.29	20	$5.80
LM1117MPX-5 5V Linear Voltage Regulator	Texas Instruments	$1.16	2	$2.32
ATSAMC20G18 A-AUT Microcontroller	Microchip Technology	$4.06	3	$12.18
20 Pin Connection Header	Molex	$2.28	6	$13.68
MBR0520L-TP Diode	Micro Commercial Co	$0.45	2	$0.45
Wire Connectors	Amazon	$9.69	1	$9.69
Saran Premium Plastic Wrap	Saran Wrap	$6.99	1	$6.99
Scotch	3M	$2.99	1	$2.99

Electrical Tape				
Thin Film Force Sensing Resistors	Walfront	20	$4.14	$82.70
9V Battery Holder	Tegg	$4.30	2	$8.60
Wire	ECEB	$0.00	~50 ft.	$0.00
I2C 1602 LCD Display Module	GeeekPi	$5.50	2	$11
Solo Ultra Clear 9 oz Cups - 50 pack	Solo	$7.49	1	$7.49
16 mm Push Button - 5 pack	Weideer	$8.99	1	$8.99
Total				**$564.12**

4.2 Labor

On average, UIUC ECE graduates make $92,500 out of college. Assuming a 40-hour work week, and taking into account vacation and sick days, this averages out to about $47.20 an hour, which is the number we will use to calculate labor costs for the three members of the team. We spent about 10 hours per week each working on this project, and with approximately 13 weeks spent on this project, our labor costs are estimated to be $18,408 (3 people x $47.20 an hour x 10 hours / week x 13 weeks).

4.3 Schedule

Table 2 below shows the schedule our team followed throughout the semester, with each team member's individual tasks listed.

Table 2: Schedule

Notable Dates	Major Deadlines	Keith	Spencer	Nishita
Feb. 28	PCB Board Reviews	Component Research & KiCad design	Component Research & KiCad design	Component Research & KiCad design
Feb. 29	Design Review	Design Document Revisions & Order Parts	KiCad Design for PCBs	Design Document Revisions
Mar. 7	First Round of PCBway Orders	Final Touches on PCB Design	Final Touches on PCB Design	Final Touches on PCB Design
Mar. 8 - Mar. 11	Building	Soldering	Soldering	Software Research & Begin Framework for use of components
Mar. 12 - Mar. 21	Spring Break	Having Fun!	Having Fun!	Having Fun!
Mar. 20 - Mar. 28	Testing	Software design for individual components & User flow designing	Hardware to Software integration	Software design for individual components & test environment for testing programmable components
Mar. 28	Second Round PCBway Orders	KiCad Design & Software	KiCad Design	KiCad Design & Software

		improvements		improvements
Mar. 29 - Apr. 4	Building	Soldering & component wiring	Soldering & component wiring	Software Improvements & Component wiring
Apr. 5 - Apr. 17	Testing	Software & user flow improvements & hardware fixes	Hardware design fixes & component integration	Software improvements & hardware to software component integration
Apr. 17 - 21	Mock Project Demo	Began final report outline. Prepared subsystems for demonstration to Zicheng.	Began final report outline. Prepared subsystems for demonstration to Zicheng.	Began final report outline. Prepared subsystems for demonstration to Zicheng.
Apr. 24 - Apr. 28	Final Demo And Mock Presentation	Integrated subsystems, finished project, began work on slide deck	Integrated subsystems, finished project, began work on slide deck	Integrated subsystems, finished project, began work on slide deck
May 1 - May 3	Final Presentation & Papers	Final Presentation Rehearsal & Final	Final Presentation Rehearsal & Final	Final Presentation Rehearsal & Final

		Paper Edits	Paper Edits	Paper Edits

5. Conclusion

5.1 Accomplishments

Our finished product fulfilled all of our high-level requirements. We aimed to create a portable product by reducing it in size by 50% and for it not to be plugged into the wall, and was exceeded by reducing the mat by 75% as well as using batteries for power. Next, we set out for accuracy with low latency, precise cutouts for the cups, and correct tolerance. We accomplished this with proper cutouts that allowed for very minimal shifting of cups, correct force tolerance within 10 grams, and zero issues due to latency. For our product to be intuitive, we had success with two buttons per team to update score and restart the game, and we did not affect or slow the game of beer pong with our product.

Taking into account not only our finished project, but also all we learned about PCB design and implementation, power systems, and product design, we'd consider this semester an overwhelming success. While our project was not perfect, the experience we gained from designing it and overcoming the obstacles that we faced throughout the semester is invaluable.

5.2 Uncertainties

The main issue with our overall design was the fact that we were unable to successfully program the microcontroller on our PCB, and instead had to resort to using a development board. Our inability to program the microcontroller was a software issue rather than a PCB design issue, as every IDE we tried to use with the microcontroller failed to properly program it. While this did not impede the functionality of the finished product, it did require us to have an extra board, which added a non-trivial amount of bulk to the mat.

Throughout the semester, we also ran into issues powering our components, specifically our LEDs. While testing our LED rings, two of them burnt out unexpectedly within the first ten minutes of usage. We were initially confused, until we realized we had connected our 9V battery directly to the LEDs without the use of the 5V linear regulator, and, as the LED rings were only rated for 5V [1], burnt them out as a result. We ran into the opposite problem later on in the project. When attempting to power the LED rings through the PCB, we noticed that only the red LEDs on each ring were lighting up, while the green and blue LEDs were completely off. After troubleshooting, we realized that this was the result of the current-limiting resistors featured on our original PCB design. We had used 1kΩ resistors to connect each LED ring to +5V, not realizing that the LED rings had current-limiting resistors built in; as a result, there was a voltage drop across each 1kΩ resistor that dropped the total voltage across the LED ring below the turn-on voltage for the green and blue LEDs (the red LEDs still worked because the turn-on voltage for red LEDs is lower than that of green and blue LEDs [2]). This was fixed by simply eliminating the external current-limiting resistors. We desoldered the 1kΩ resistors and replaced them on the board with 0Ω resistors, effectively shorting the path. After doing so, the LEDs worked perfectly.

The remainder of the issues we faced throughout the semester were logistical. A number of delayed and canceled orders led to us falling behind schedule, and unfortunately not being able to have all 20 LED rings on the mat in time for the demo. In addition, we were not able to secure a PCB encasing in time, which meant that we had a number of wires sticking out of the side of our mat in plain sight, which was both unsightly and detrimental to the functionality of the mat, as they could very easily come loose. While these delays were rather unfortunate, they granted us insight into how easy it is for a project to fall behind schedule, and required us to devise makeshift solutions to ensure the project could still be finished. In the case of the wires sticking out of the mat, we used tape to ensure that the wires were in one neat package rather than strewn about haphazardly.

5.3 Ethical considerations

The main ethical concern we had over the course of the semester is the fact that our project, being an accessory for a game commonly played with alcohol, may encourage unhealthy and unsafe drinking habits, which would be in direct contradiction to our duty as engineers to ensure the good health and safety of the users of our project and in conflict with Section I.1 of the IEEE Code of Ethics [3] as well as 1.2 of the ACM Code Of Ethics [4]. However, considering the widespread popularity of the game of beer pong, we do not believe that our project will be introducing anyone to this game and perhaps by extension binge drinking. Theoretically, if our product was mass produced, we would include with it a disclaimer stating that the creators do not encourage binge drinking, as well as a warning detailing the harmful effects of binge drinking and the danger of alcohol poisoning. We do not want to encourage unhealthy drinking habits of any sort – rather, we hope with this project to help streamline an already immensely popular game that can be played in a safe and controlled manner.

5.4 Future work

If we were to continue developing this project, there are a number of areas we could improve on. If we were to use high-precision load cells rather than force-sensitive resistors to detect the weight of each cup, we could develop a version of the mat that could sense when a cup has been hit. Another possible improvement would be building the mat directly onto a table, as this would do away with the necessity of carrying around both the mat and a table and would provide a dedicated place for the battery pack and PCB encasing to reside. One final improvement we'd like to add would be the inclusion of audio, adding sound effects when a cup is hit or when one team wins. This would require both the addition of speakers and a microcontroller with a significant amount of memory, as even one second of audio, assuming a sample rate of 44.1 kHz and a bit depth of 16, would require 88.2 kB of storage.

References

[1] Adafruit, “SMD Type LED Datasheet,” SK6812RGBW-XX, 31 July 2015. Available: https://cdn-shop.adafruit.com/product-files/2757/p2757_SK6812RGBW_REV01.pdf [Accessed: 1-May-2023].

[2] Crowell, Gary. “The Forward Voltages of Different LEDs.” *CircuitBread*. 2 May 2019.

[3] “IEEE Code of Ethics,” *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 17-Feb-2023].

[4] “ACM Code of Ethics and Professional Conduct” *Code of Ethics*. [Online]. Available: https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct. [Accessed: 17-Feb-2023].

Appendix

Appendix A: Requirement and Verification Tables

System Requirements and Verifications for the Cup Subsystem

Table 3: Force Sensors

Requirement	Verification	Verification status (Y or N)
The force sensors need to ensure that weight changes within 10 grams should be detected	• Connect the two tabs of the force sensitive resistors to the MCU and display the weights/ analog readings on the System monitor • Vary the ranges of weights on the force sensors to showcase changes • Graph analog readings vs weights based on the values obtained	Y

	• Using the graph, analyze the tolerance levels of the force sensors	
The force sensors should have a well defined resistance to weight graph to ensure that the acceptable liquid range can be identified Range of weights based on level of water: 137 - 250 g	• Connect the force sensor to the MCU and the LEDs • Vary the range of weights on the force sensor from 137 - 250 g (+/- 10 g)	Y

	• Identify and experiment when and how the LED values change based on the weights	

System Requirements and Verifications for the UI Subsystem

Table 4: LED Rings

Requirement	Verification	Verification status (Y or N)
The LED rings need DC Voltage of 5V to draw a current of 1.44A to be powered on	• Connect the LED rings in parallel to the 9V power source with a linear voltage regulator and in series to a multimeter in diode mode • When a voltage of 5V is supplied to the LED rings, detect if the LEDs turn on and measure the current required by each LED • Additionally, measure the current and voltage used by all the LED rings to differentiate the	Y

	varying brightness of the LED rings	

The power consumption for the 20 LED rings should be at least ~0.36A such that all the LEDs can be lit up synchronously	• Compare power consumption on different colors with varying levels of current • Measure current used by all LED rings and calculate power consumption and compare with brightness • Ensure that an effective level of brightness is achieved to differentiate between on/off LED rings. If not, increase the brightness offered and current (max brightness current: <1.44A)	Y

LED rings should remain off until the the weight of the beer pong cup (weight of the cup + liquid + ball) has fallen below 137 g or above 250 g thus, turning it on	- Vary range of weights on force sensor and test tolerance - Measure the analog read values from the force sensors in the acceptable range of 137 - 250 g - Measure the time and range of values for LEDs to turn on/off - Use software to ensure that the LEDs turn on/off corresponding to the range described	Y

Table 5: LCD Screen

Requirement	Verification	Verification status (Y or N)
The LCD Screen needs DC Voltage of 5V to be powered on	• Connect the LCD Screen to a 9V power source with a linear voltage regulator and in series to a multimeter in a diode mode • When a voltage of 5V is supplied to the LCD Screen, detect if the LCD Screen turns on with characters being visible on a blue/white contrast ratio	Y

Display words and counts on screen - “Score: xxx, Player Name: xxx”	• Detect if words are visible on screen when paired with software • Ensure maximum number of characters is 32 through software to prevent clipping • Record boolean value for data values printed to LCD as read from microcontroller using serial debugging	Y
Refresh rate is high to ensure reduced delays between frame switching Screen should update to a base state upon a reset button press	• Calculate time taken for words to get updated upon a frame update • Connect to reset button and software to test frame switching	Y

	upon a button press	
The LCD screen should be able to update to 0 when Reset button is pressed and stop updating once the Stop button is pressed	- Connect the LCD Screen to a 5V power source and microcontroller - Press the Reset button and detect if the words and counts on the LCD screen return a default value. A boolean variable can be used to test this through the microcontroller - Press the Stop button and detect if the words and counts on the LCD screen change	Y

	when there's a move made by the player. A boolean variable can be used to test this through the microcontroller	

Table 6: Buttons

Requirement	Verification	Verification status (Y or N)
Ensure that pressing the buttons affect the values showcased by the LCD screen i.e. default or unchangeable values	• Connect the LCD Screen and buttons to a 5V power source and microcontroller • Connect buttons to a 5V power source and microcontroller • Press reset button and use boolean variable to test values passed through software • Press stop button and detect changes on LCD screen by using boolean variable	Y

System Requirements and Verifications for the Control Subsystem

Table 7: Control

Requirement	Verification	Verification status (Y or N)
The control system needs to have a low latency rate (less than 1 s) to ensure quick syncing of the components.	• Connect system to components with a power supply • Test code written to microcontroller by increasing/ decreasing the weight of the liquid pong cup (weight of the cup + liquid + ball) below 137 g or above 250 g • Detect and record data transferring by measuring time taken for turning on and off LEDs and shift in values on the LCD Screen	Y

System must ensure current state of components remain constant unless charged	• Detect and record if LEDs remain off unless the weight of the beer pong cup (weight of the cup + liquid + ball) has fallen below 137 g or above 250 g • Detect and record if the LCD screen showcases the same values of the team score and player number when there is no player movement or the buttons have not been pressed	Y

System Requirements and Verifications for the Power Subsystem

Table 8: Power

Requirement	Verification	Verification status (Y or N)

The battery needs to supply a DC Voltage of 9V to the numerous components	- Connect the battery in parallel to a voltmeter - Identify the value obtained by the voltmeter and compare it to ~9V to ensure that the voltage is sufficient for the circuit	Y