Portable Battleship Display

Design Document

Group #80
Jonathan Rakushin, Elizabeth Roels, Colin Lu
TA: John Capozzo

1. Introduction

Objective

Originating in the 1930s as a rudimentary pen-and-pencil game[1], the game Battleship has grown into what is known as one of the most popular and prolific board games the general public enjoys. It has spawned countless spin-offs, including video games, movies, and electronic reimaginations. However, throughout all these various forms, there is a main issue Battleship failed to address: the idea of portability. The current standard design of Battleship includes two large plastic boards, 84 white “miss” pegs, 42 “hit” pegs, and five ship pieces[2]. As a result, Battleship is a hassle to take out, set up, and play. Furthermore, many pieces get everywhere, leading to a mess where the pieces are prone to get lost. The tedium of taking Battleship off the shelves, setting up the game board, and marking coordinates with pegs each turn has made it near impossible to play on the go, an important aspect of a game the public values greatly in an era where mobile apps and handheld games are an arm’s reach away.

Our goal for this project is to bring back the fun of Battleship through a fun new medium: a portable wireless display. In order to do so, we will create a battery operated digital display, complete with visual and audio elements that will be simultaneously controlled by a microprocessor. We will implement a chip into each device as well so they can wirelessly interface with each other, removing the need for wires or any other interconnection. We hope that with the completion of this project, we could reignite the joy of Battleship in people of all ages while minimizing the hassle its setup has become associated with.

Background

It can be difficult to keep people, especially children, occupied on the go. It is why handheld and mobile games are so widespread and prevalent -- the convenience of having a portable distraction to entertain themselves while traveling or waiting around has led to the public spending increasing amounts of their income on mobile apps or video games[3]. However, both of these forms of entertainment have certain drawbacks associated with them. For mobile games, many require internet access or a mobile carrier, two things people may not have or want access to. Furthermore there are relatively few mobile apps which support multiplayer functionality. On the other hand, video games do not from these same problems, but instead from having a complicated set of controls. The user interface for handheld video games frequently include a wide variety of controls. Although this leads to varied gameplay, it can be difficult for the public, especially younger children, to learn. Furthermore, small features like buttons or styli are prone to get damaged or lost.

In order to address these issues in our project, we plan on implementing a sleek, simple user interface that is easy to understand yet engaging. We will also include multiplayer support because we believe that including a social aspect to a game is key to making it enjoyable and memorable. We hope that our device will be able to be enjoyed anywhere, at any time, for audiences of all ages.

High-level Requirements

The three main goals we aim to meet for this project are:

1. The game is aesthetically pleasing
    a. The included audio elements are interesting and meaningful.
    b. The included visual elements are similarly interesting and meaningful.
    c. Good simple interface design.
2. The game is light and portable
    a. Devices can interact wirelessly through bluetooth chips.
    b. The physical design minimizes extra bulk, allowing the product to be carried easily in a travel bag.
    c. No additional pieces needed, everything is included within the display.
3. The game engages audiences of all ages
    a. Adding a handicap for younger children.
    b. Adding a CPU for an audience who cannot play against others

The main component of the design will be a centered microcontroller. This microcontroller will handle all of the inputs and outputs made by both players throughout the game. Each player will need their own handheld device, and the two devices will communicate wirelessly. The device will be powered by a lithium ion battery, which should have a battery life of at least 8 hours when playing the game. The microcontroller will also be connected to, and powering a microphone, speaker, button array, wireless module and LED display. The microphone will be used for the game's voice commands. Since the old method of playing Battleship involves saying your move to the other player, we'd like to keep this in place but update it with voice recognition. The player will state his/her command and the game will get confirmation, and when approved, will "fire" at the opposing player's board, detecting a hit or not. The connected speaker will play sounds associated with the firing, including hit, miss and sunk noises. The button array and LED display will be the game board in the player's eyes. The player will place their pieces on the board using the button array at the beginning of the game, and from then on the board will show the opposing player's shots taken.

2. Module Description

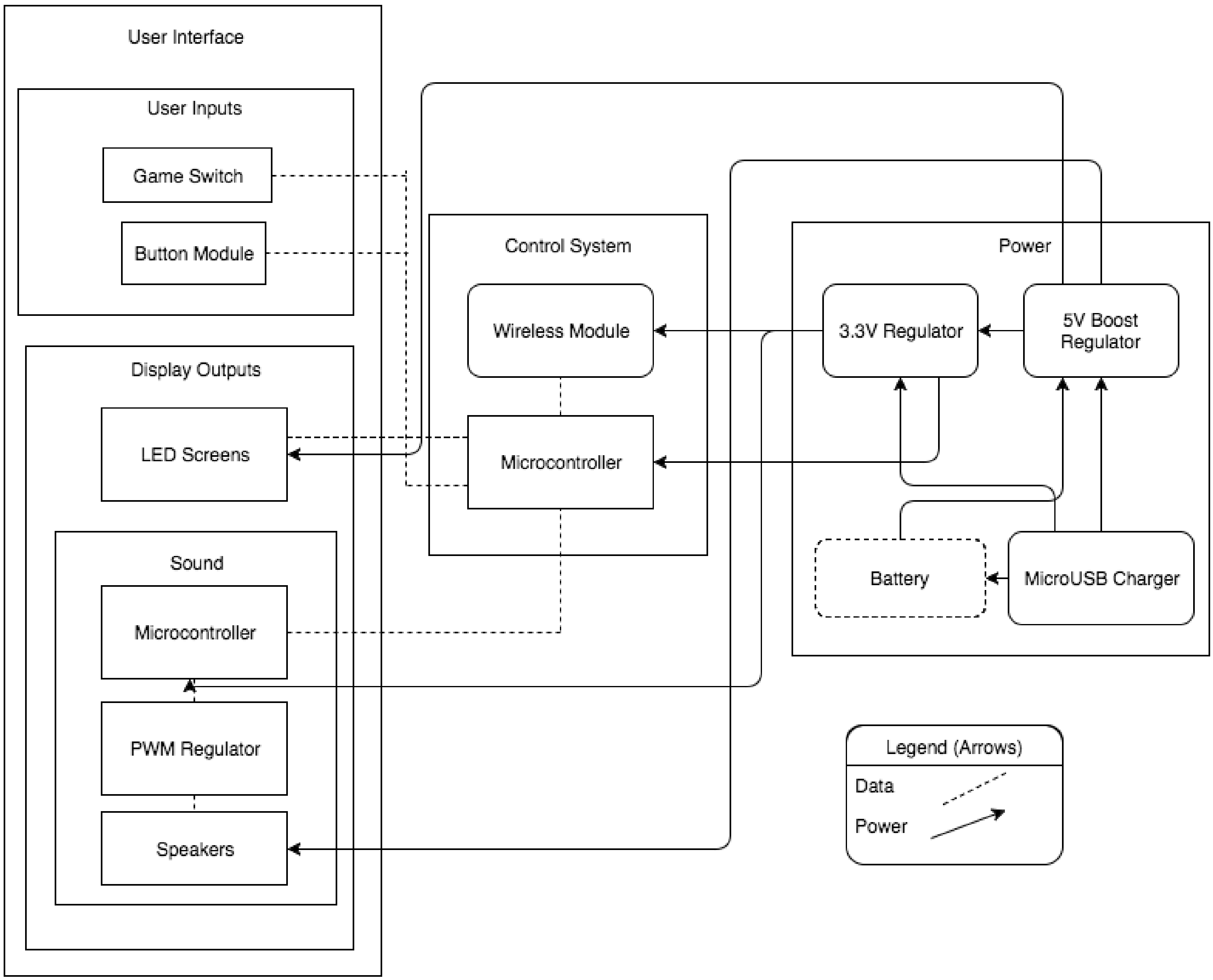

Figure 1: A block diagram depicting the logic of our project.

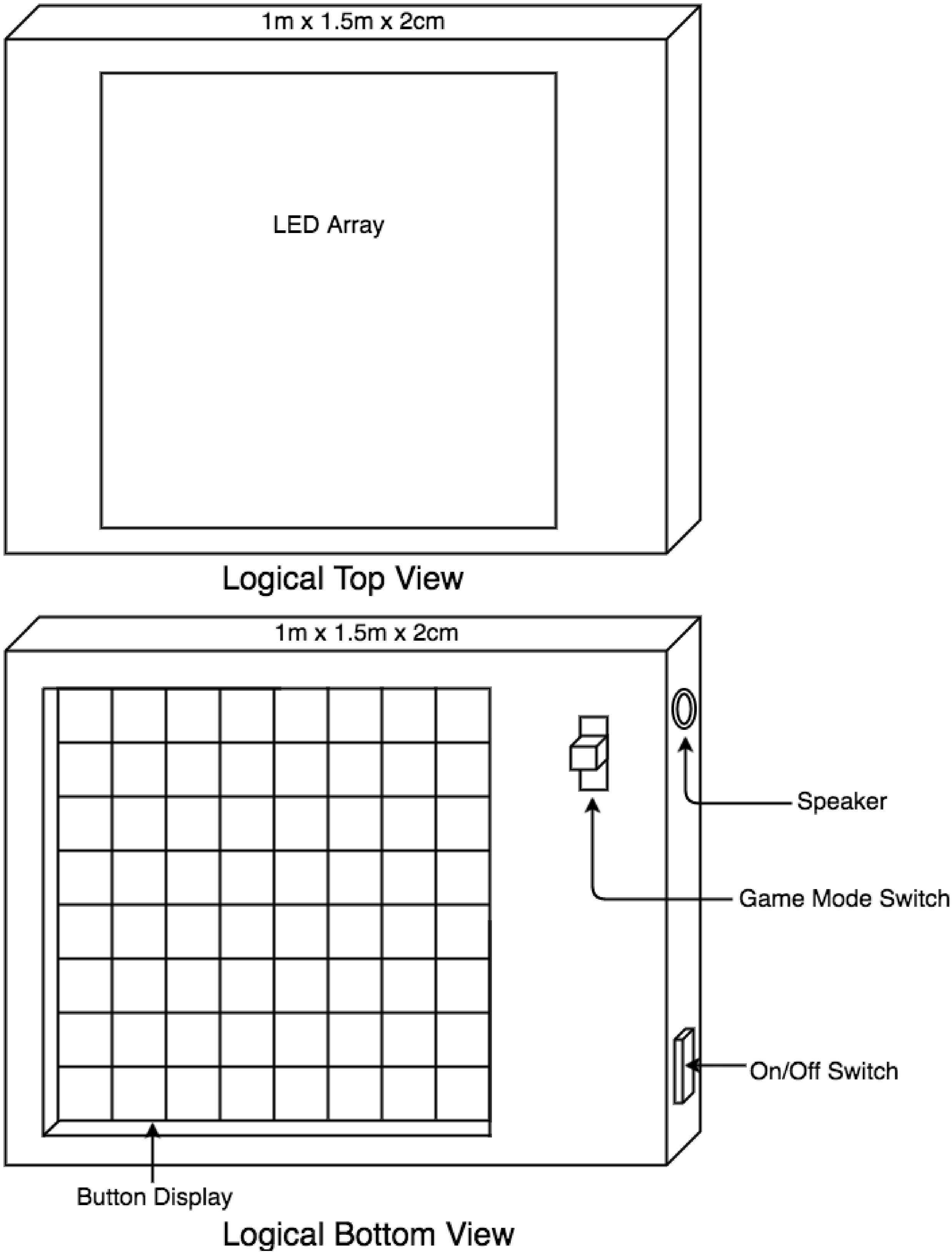

Figure 2: Physical Design Diagram

1. User Interface

For the user interface, we needed to decide how to control our device. Originally we wanted to use a touchscreen interface in order to control the device. Ultimately we decided to go with a button-controlled display due to the fact that it was much easier to program and a lot more cost effective.

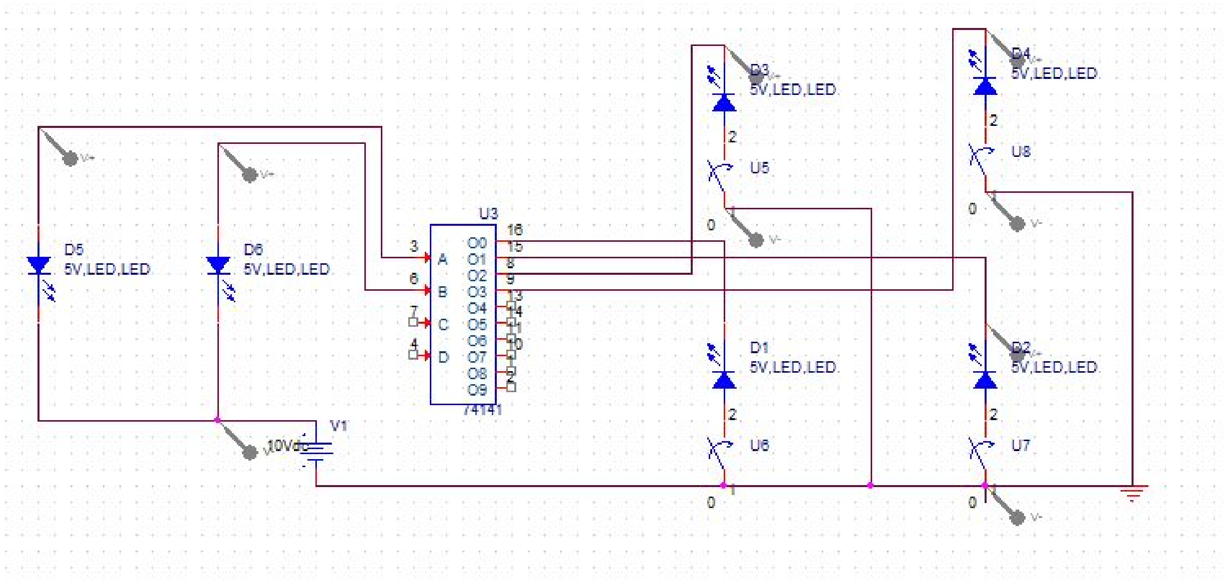

Figure 3: LED Button Array

For the Physical Display, we will be using the 74AC154 4 to 16 mux, Jameco 2219559 RGB LED, Atmel microprocessor, and GPTS203211B push button. Fig. 3 was our initial idea to lead into a 10x10 array of RGB LED and push buttons(switches). The circuit for Fig. 3 is for testing the constraints of the LEDS for reading in the logic. For now we are looking into COM-09264 for the LED's for testing since that was with our original design, but we have change to a cheaper alternative, the Jameco 2219559[4]. We were considering an array of touchscreens instead of an array of pushbuttons, but the price seemed to steep. We did hope for a 10*10 board, but with prices exceeding our consideration, we have chosen to go with a smaller board(8 by 8, 5 by 5, etc.)

The Figure 3 is a 2x2 array, a test run to test our idea of a 10x10 array would look on a smaller scale. The decoder above is a 74141 10 to 4 decoder to conserve on inputs into the microcontroller. Using an exclusive 10 to 4 mux will save board area in designing. We are looking into a LED driver in its place to control the multitude LEDs, but still considering.

These will be used in the interface display since we have settled on buttons instead of the touchscreen panels. As for the switches we plan on using Standalone Toggle Capacitive Touch Sensor Breakout - AT42QT1012. It would settings pre-programmed into the device for reading if the device has been touched more than once[5]. AT42QT1012 is very expensive actually, $5.95 a piece. For one board makes at most $595 with 100 pieces being used. We are currently looking into alternatives. One significant problem to consider is the AT42QT1012 does have drift signal, an error between internal capacitances of the device that must be compensated for, otherwise there will be delays and sensitivity error[6].

The touchscreen being as expensive as it is, we will probably go with a button array. ~$100 for an entire board. For instance, the push button GPTS203211B is 1.44 cm squared, and costs $0.70 per button[4]. We also chose this push button because it is an SPST button toggling between on and off, for we only need the button to hold once pressed and 'depressed'(similar function with the touchscreen boards), sudo memory of the board to keep information of ship locations, but the problem would be the player would have to depress the ship location if hit or programming to reset the buttons and noting whether it was pressed to read negative values. We are currently looking into possibilities of using OnMom Off, meaning it will constantly read off if not pressed. Doing OnMom-Off gives us as the designer to have more opportunities when it comes to resetting the game, and the downside would be having enough storage to instantaneously to catch this data. The one problem we are currently having is not enough information on datasheet for the pushbutton to run simulations. The mention its rating, assuming to the rating being when the device is off, giving a high resistance of possibly 30 Ohms and dissapating 30W. This is fine for our purposes; however, it still doesn't give enough information for when the device is on. For our simulations, we assumed the button to toggle the button to have a high resistance of 30 Ohms when off and an on resistance like a short. The lack of information will not be so damaging to our idea circuit design with our assumed constraints of the resistance of the button, for the LEDs and microprocessor are assumed to dissipate a low amount of power. However, there are no specific or customizable push buttons on PSpice which leaves room for error when testing the real constraints of the push button to behave differently from what we will test in the lab of the actual constraints of the Push Button GPTS203211B.

The former LED, China Young Sun LED Technology Co., LTD. YSL-R596CR3G4B5C-C10, was used in testing since there are no datasheets available online for the Jameco 2219559, but it is good to learn from testing the former LED the constraints the LED datasheets may give and what we can take away from it when we test the actual device, Jameco 2219559. The RGB LED's colors all have a forward current of 20mA and a peak forward current of 30mA[8]. The forward voltage varies between different different colors which limits our design to a circuit that will read the the different colors at different voltages, but what want to avoid is to maintain turning on the devices individually. If voltage is 3.0V across a parallel circuit of the LED's different colored LEDs, all of the colors could turn on. To get Red, the LED has to have a minimum Voltage of 1.8V or a max of 2.2V[8], and to get Blue or Green to light, the LED has to have a minimum Voltage of 3.0V or a max of 3.4V[8]. Using the typical readings from the datasheet, the Red and Blue/Green power dissipation is 40mW and 64mW respectively, which were used in customizing the LEDs to match close with the actually LEDs since the specific part on PSpice is not available. Leading into the error in simulation, the LEDs needed to be customized to match as close as possible, but may have different readings when device is tested. The lack of information from the data sheet required me to run some simulations to get as close as possible to the actual LED, but this shouldn't hurt us at all in the long run, for the LED we simulate for is not what we will use in the end design. The things we had to acquire in simulation was the forward resistance. Through simulation, there can only be a maximum forward resistance, for every forward resistance of the LED will never reach the maximum forward current of 30mA as listed in the data sheet, it gets as close as ~28mA and to map a specific RGB's minimum voltage. In Tolerance Analysis and Risk does it go further on proving the constraints of the findings. My results have found the maximum forward resistance for the Red and Blue/Green are 0.4 and 0.7 Ohms respectively. Again, these are not the RGB LEDs the team will use. The RGB LEDs the team will use are $0.39 each, a fairly good price.

The YSL-R596CR3G4B5C-C10 is too expensive and Jameco 2219559, after calling Jameco, didn't have a datasheet for the product. With these constraints, it led us to go forward with the DHL Free F5 386452483 LED.

The new LED we plan on going forward with is the DHL Free F5 386452483[15]. The forward voltage range for Red is 2-2.2V and forward resistance of 0.52Ohms, 3-3.2V and 0.72Ohms for Blue/Green[15]. All LED leads have a forward current 20mA[15]. Forward resistance was acquired by using the same method used for obtaining the forward resistance of the YSL-R596CR3G4B5C-C10, keeping the constraints of Reverse Voltage 5V, Leakage Current 10e-6, and Max Forward Current Power Dissipation is 2.1V*0.02A = 4.2W for Red and 3.1V*0.02A = 6.2W for Blue and Green. The luminosity of 2000-3000mcd, 15000-18000 mcd, and 5000-7000mcd for red, green and blue LEDs respectively[15]. With a viewing angle of 30 degrees, the luminous intensity is fairly bright per every LED color is in the ultra- bright region of .3 to 1.4 lumens about the luminosity of a candle[17][16]. This is not too bright, but visible and fun for children.

$$\Phi_v = I_v \times \Omega_v$$

$$\Omega_v = 2\pi(1 - cos(\tfrac{\theta\pi}{360}))$$

Φ_v = [lm] : luminous flux
I_v = [cd] : luminous intensity
Ω_v = [sr] : solid angle
$\boldsymbol{\theta}$ = [degrees]

Equations [17]

Each RGB LED will be .58cm in diameter, roughly taking up 33.64cm squared of area, more than twice as much as the button array alone. The area of the interface board should then exceed 144cm squared plus LED's array 33.64cm squared, 177.64 cm squared assuming 100 button and LED array[4]. That would still go well with our design seeing that they are inexpensive, about parameters in of area. This will still match with our design ideas. The same LED ideas will be displayed similarly with the display module, but without buttons or decoder.

Requirements	Verifications

1. Simple, easy to control user interface. a. User interface is limited to three controls: on/off switch, button array, and game mode switch. b. Board offers low latency between commands, <50 ms. 2. Users are able to place pieces on the logical display through button presses. 3. Interface is attractive to the eyes and not harmful by having too high of luminosity 4. Can easily toggle button to be On/Off	1. Audiences of all ages understand the interface controls. a. Verify that the interface has 3 controls only. b. Test latency of the device through a multimeter, confirm the signal delay is <50 ms. 2. Test the touchscreen capabilities, press two coordinates in order to place each piece. 3. The luminosity is 1 which is similar to looking into candle light. 4. Nearly open with 30Ohms when not pressed, and resistive when not pressed with a low resistance, 100mOhm

2. Speakers

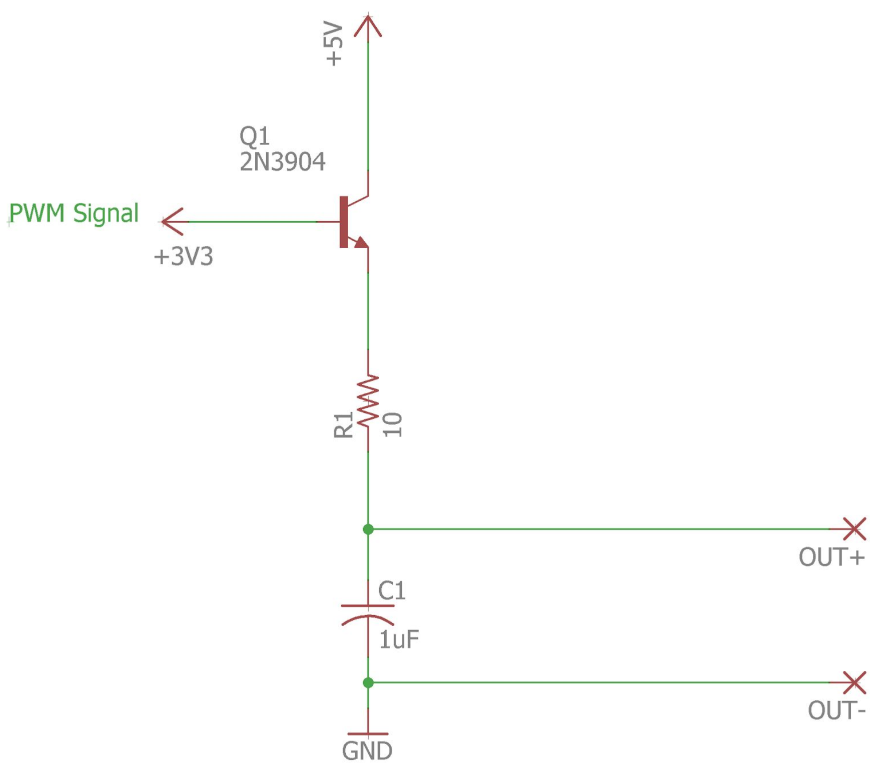

Figure 5: A speaker module to operate the speakers

For the speakers, we wanted to go with a lightweight, cheap option that is simple to incorporate into our design. We opted for a second microcontroller to store the sounds and to run PWM calculations on WAV files. Since this is very CPU intensive, another microcontroller is needed to drive the sound. A mosfet with use the PWM signal to drive the sound coming off the 5V line, with a low pass filter to smooth the audio. The speaker used will be small in wattage, at .3W.

The microcontroller being used is an Atmel ATxmega256A3 because it will have 256kB of flash storage to work with, with PWM output. The audio will be sampled at 8,000Hz. We'll want the PWM to sample and send out at 100,000 Hz. We'll want to cut off the frequency at 20,000 Hz to get the full range of hearing for the speaker. To calculate values for RC, we'll be using a 1 μF capacitor:

$$f_{cutoff} = \frac{1}{2\pi RC}$$

$$20000 = \frac{1}{2\pi R * 1 * 10^{-6}}$$

$$R = \frac{1}{20000 * 2\pi * 10^{-6}} = 7.96\Omega$$

Therefore, the capacitance will be $1\mu F$ and the resistance will be 8 ohms. Since the limit for the resistors is lower, at 10 ohms, we will use that instead, which should still give us most of the range we need.

The microcontroller that controls the battery will be connected to the main microcontroller by a 2 bit bus. When the sound microcontroller receives the signal, it will play one of 4 sounds: an error noise, sinking noise, hit noise or miss noise. These will all be in the microcontroller as WAV files.

The speakers will be driven by a mosfet in standard operation mode. The source will be connected to the 5V line, while the drain will be connected to the speaker with the low pass filter on it. There will be no volume control. The gate will be connected to the PWM signal, therefore when the PWM signal is high, the sound will play. If sampled enough, this can imitate the sound we need.

Requirements	Verifications
1. Output is audible from 5ft, with a frequency range of 1-4 kHz. 2. Microcontroller is capable of driving a PWM signal. 3. Sound output is >90% similar to the sound sample.	1. Verify audio is clear 5 feet away using a sound level meter. 2. Microcontroller uses PWM signal to encode the sound sample. 3. Sample both sounds, verify that the original sound is 90% similar to the speaker sound

3. Microcontroller

For our microcontroller, it will need to be able to interface with a lot of different parts at once. It will need to drive two 64 RGB LED displays, 18 additional RGB LEDs to be used as a legend, and receive inputs from each of the 64 buttons. Therefore, the microcontroller will need to be at least 32 bits. The datasheet doesn't have enough information on what it's input/output limitations are which causes problems if it is not compatible to read 30W buttons or 2-3.4 volts and 20-30mA to light the LEDs. Latency is not available on the data sheets for the LED or the buttons.

This will be the brains of our project. It will receive inputs from the bluetooth module and input module, and output sounds, lights and signals as needed. It will also handle all of the game calculations such as deciding a winner and checking valid moves.

Shown below is a diagram of what we would like to program the microcontroller to do. We will follow this flowchart when designing the algorithm to interface with the speakers, LED/Button Array, and bluetooth chips.

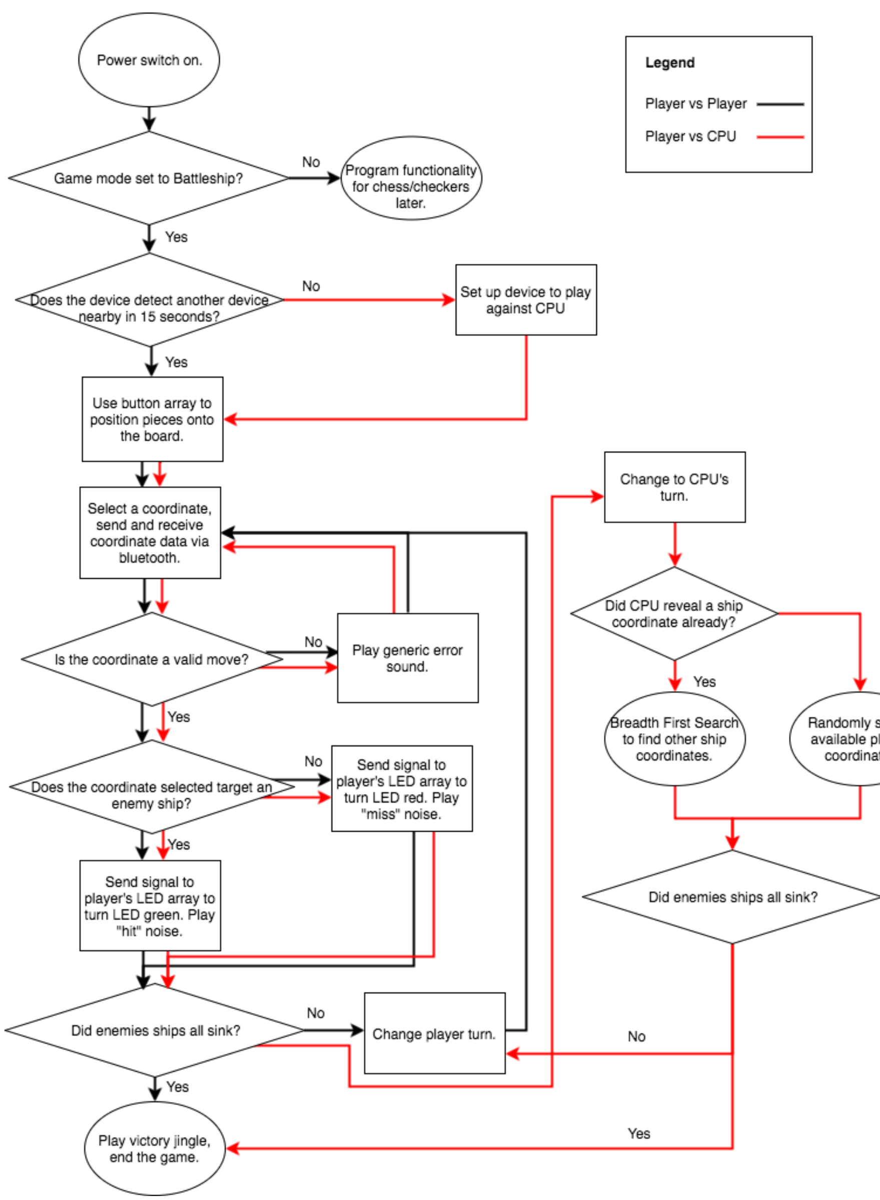

Figure 6: A flowchart of the microcontroller functionality.

Requirements	Verifications
1. Device must be able to communicate and interface with each other as well as each component with <50 ms. 2. Device can interact with a UART Bluetooth chip 3. A single display is able to light all 100 LEDs at once.	1. Properly interfaces with each module in order to output the correct display info and audio. 2. Verify UART data is received and sent correctly by the MC. 3. Plug in power into the display, press the button array to see if LEDs light up correspondingly on the display board.

4. Bluetooth Chip

In order to wirelessly interface the two devices together we would need to include bluetooth chips in each device. We opted to go with HC-05 Bluetooth chips, due to the its lightness and great range of communication. There is also a lot of documentation included with the device, and it is usable with a microcontroller, which is ideal for use. There will be no additional PCB as the board does not need an additional interface with the microcontroller, just solder the connections properly.

Requirements	Verifications
1. Able to transmit information between devices at least 30 ± 4 ft apart. 2. Low latency, data travels between devices quickly (50ms).	1. Have two devices about 30 ft apart, test to see if devices transfer data properly. 2. Verify there is less than 50 ms between send and receive time for the data transmissions, using a video recorder and checking frames.

5. Battery

We wanted to select a cheap, potentially rechargeable battery with a good operating range for our device, which both a lithium-ion and lithium-polymer battery provides. Ultimately we went with the lithium-ion battery due to it being more affordable and maintaining the same potential voltage ranges as the latter. [9] There will be two of these batteries wired in parallel, to double the battery life.

We will be using Li-Ion 18650 batteries, with a connection system to insert them into the device. The maximum draw for the batteries is 30A, so there should be no danger of drawing too much current from the batteries with the 1A limit on the boost converter. The holding system will use two spring loaded contacts, which will be connected together in parallel.

6. Boost Converter and Charger

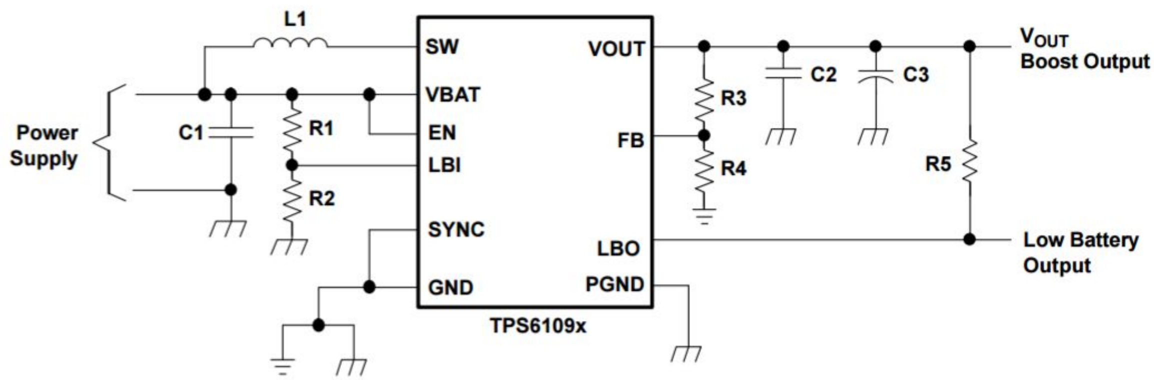

Figure 7: Typical application of the TPS61090 chip[7]

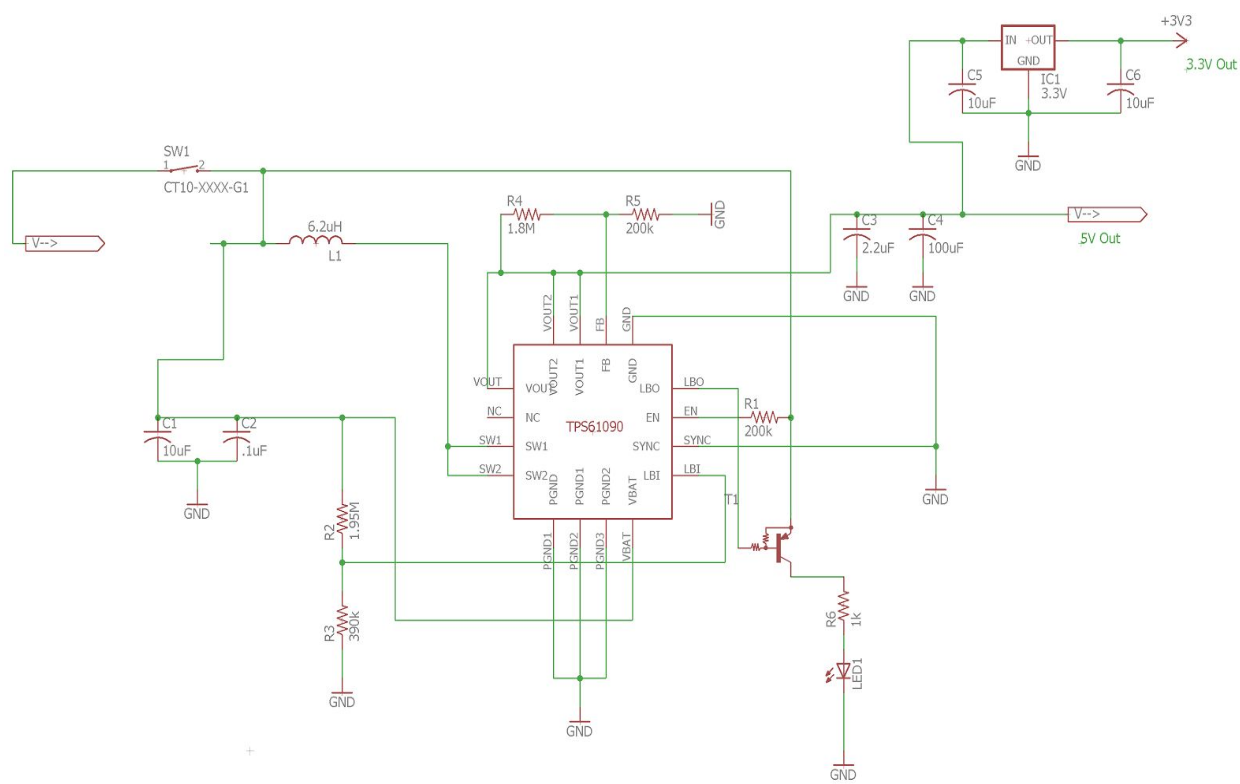

Figure 8: Our application of the TPS61090 chip with a 3.3V regulator attached

The batteries will be connected to a TI TPS61090, which will regulate the output voltage, boost voltage from 3.7V to 5V, and allow for undervoltage lockout.This device is also current regulating so the battery cannot output too much current, which would be a safety concern. A 3.3V regulator will also be included for the microcontrollers.

According to the TPS61090 document[10] the resistances must be calculated for the undervoltage lockout and voltage regulation output. To calculate the UVLO resistors:

$$R1 = R2 * (\frac{V_{Cutoff}}{500mV} - 1)$$

Texas instruments recommends a voltage for R2 of $390k\Omega$. After the battery hits 3.0V, it has less than 5% of its charge remaining[7], therefore this will be our cutoff voltage.

$$R1 = 390k * (\frac{3.0}{.5} - 1) = 1.95M\Omega$$

These two resistors coincide with resistors R2 and R1 in Figure 6.

For the voltage amplification, another resistor set is needed to set the output voltage. These are also found in the TPS61090 document. To calculate these resistors:

$$R3 = R4 * (\frac{V_{Out}}{500mV} - 1)$$

The TPS61090 document recommends using a $200k\Omega$ resistor for R4. Calculate R3 as follows:

$$R3 = R4 * (\frac{5}{.5} - 1) = 1.8M\Omega$$

Therefore resistors R4 and R5 in Figure 6 are 1.8M ohms and 200k ohms respectively.

Requirements	Verifications
1. Voltage outputs 5±.5V for the boost regulator and 3±.1V for the voltage regulator 2. Shut off the battery when the voltage is less than 3.0V ±.1V	1. Use a voltmeter to verify both lines, verify that the voltage is in the acceptable range 2. Use a multimeter and verify that current drops

7. Charging Circuit

For recharging, the MCP73871 will be used. This chip can take power from a micro USB input and charge the batteries. The chip will also allow the game to be played while the batteries charge, and includes overvoltage cutoffs.

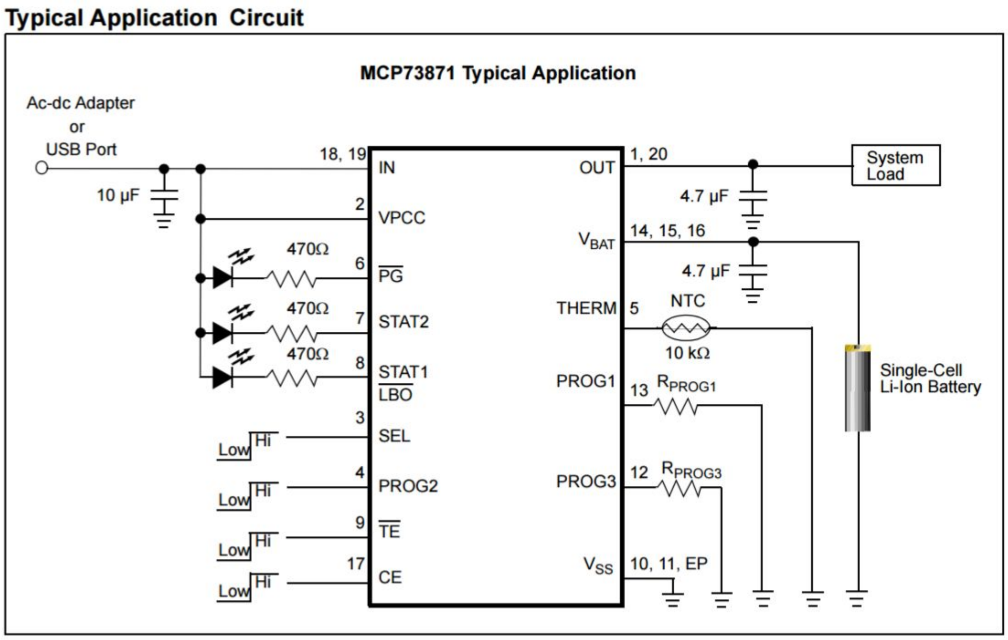

Figure 9: Typical application of the MCP73871 chip[9]

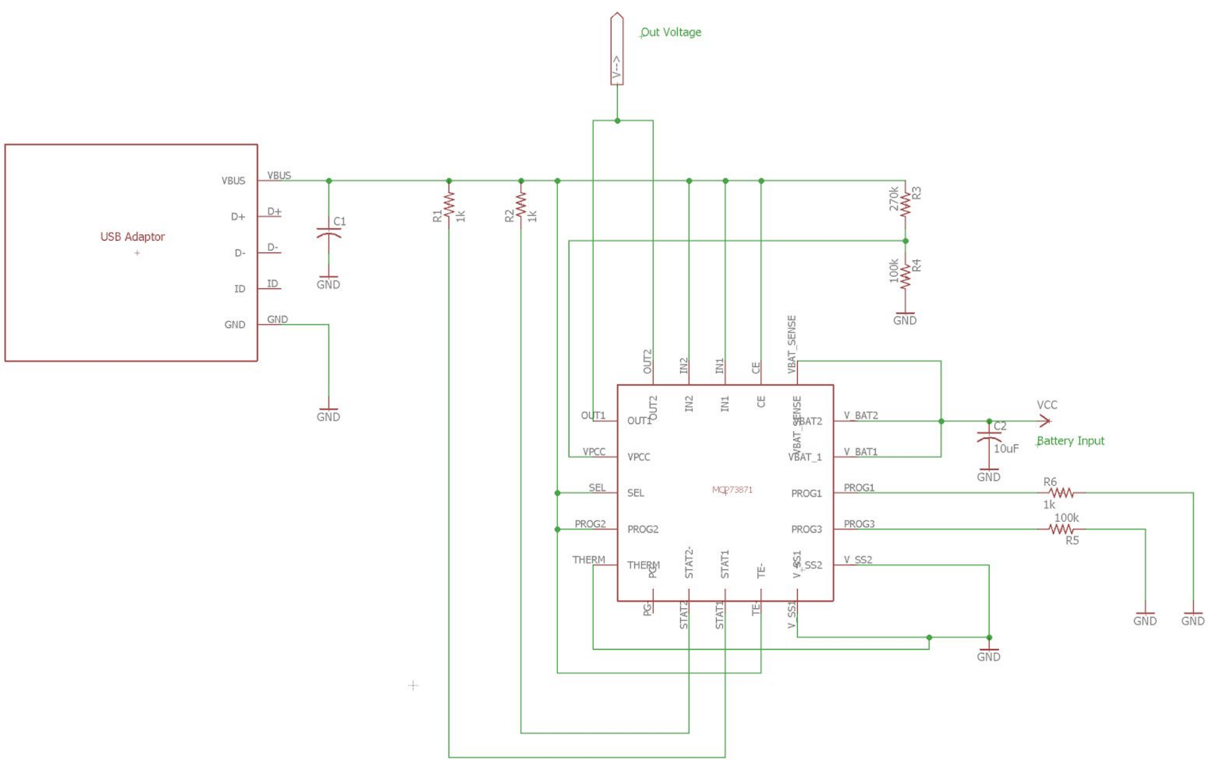

Figure 10: Our application of the circuit

To calculate the voltages to program VPCC to 1.3V, we check the MicroChip MCP73871 manual [11]. For a voltage that's 1.3V less than 5V, we use VDR to get this voltage.

$$V_{VPCC} = \left(\frac{R_2}{R_2+R_1}\right) * V_{In}$$

The input voltage of the typical USB charger is 5V. Setting R2 to 100k gives us:

$$1.3 = \left(\frac{100k}{100k+R_1}\right) * 5$$

$$.26(100000 + R_1) = 100000$$

$$R_1 = 284.6k\Omega$$

Since there is no resistor this precise, we will use a 280k Ohm resistor.

Requirements	Verifications
1. Charging takes less than five hours from empty to full. 2. Charging stops at 4.2V $\pm$.1V	1. Test a full charge cycle with a timer to verify the duration fits the requirement 2. Use a voltmeter to test that the current running into the battery is effectively 0A at 4.2V

3. Tolerance Analysis

In order to save space on our logic board and make our product safer, we are planning on using a boost converter to supply a higher voltage output without having to connect more li-ion batteries in series. However, the RGB LEDs we are planning to use for our project cannot operate unless a certain voltage threshold is reached. The responsibility of maintaining the proper voltage to operate these LEDs will fall on the boost regulator.

Below is a voltage input simulation that shows which voltage input ranges will allow the LEDs to operate.

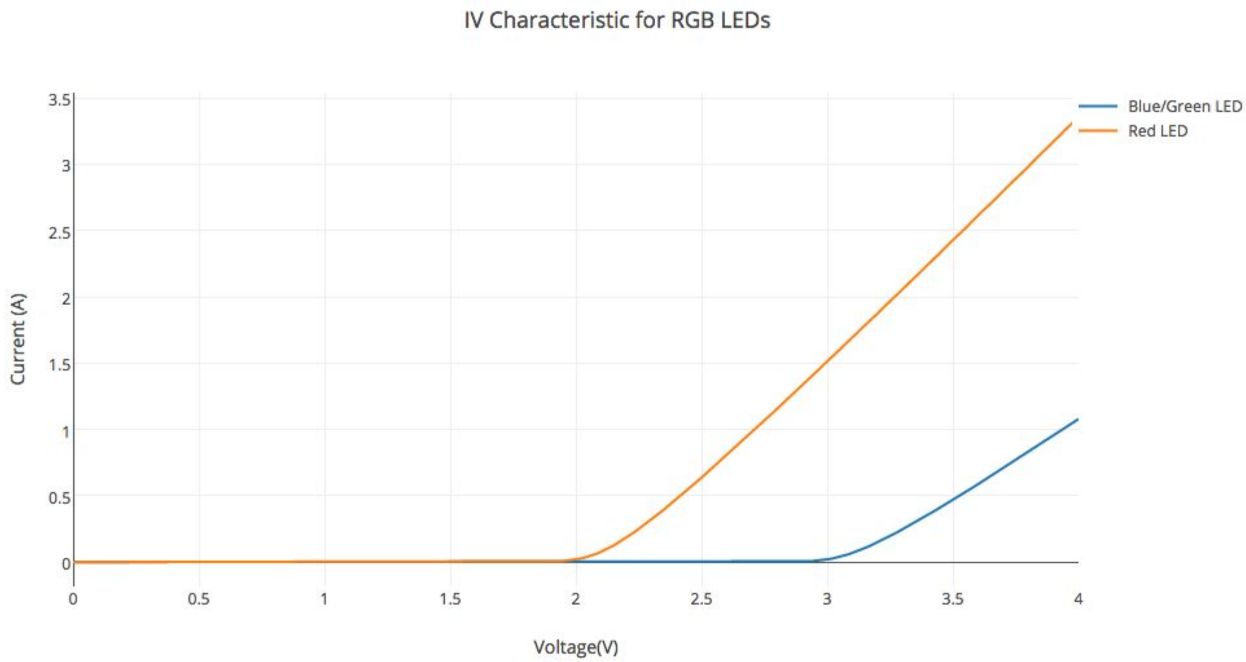

Figure 11: Operating voltage ranges for the LEDs.

As shown above, the red LED will only turn on with a forward bias of about 1.8V, and the blue and green LEDs will only operate with a forward bias of about 3V. The reason why this image is so important is the proof, with the limited requirements on datasheet and calculations for the diode, it is attainable to define what the possible forward resistance for the diode, power, and limitations on data provided. We will need to be mindful of these operating ranges in order to successfully implement LED functionality for our game. Anything smaller would not hit the datasheets minimal Voltage, and anything higher would define the diode to have a smaller minimum voltage . The operation region, square law is fairly small, about the forward and peak forward current from datasheet(20mA and 30mA).

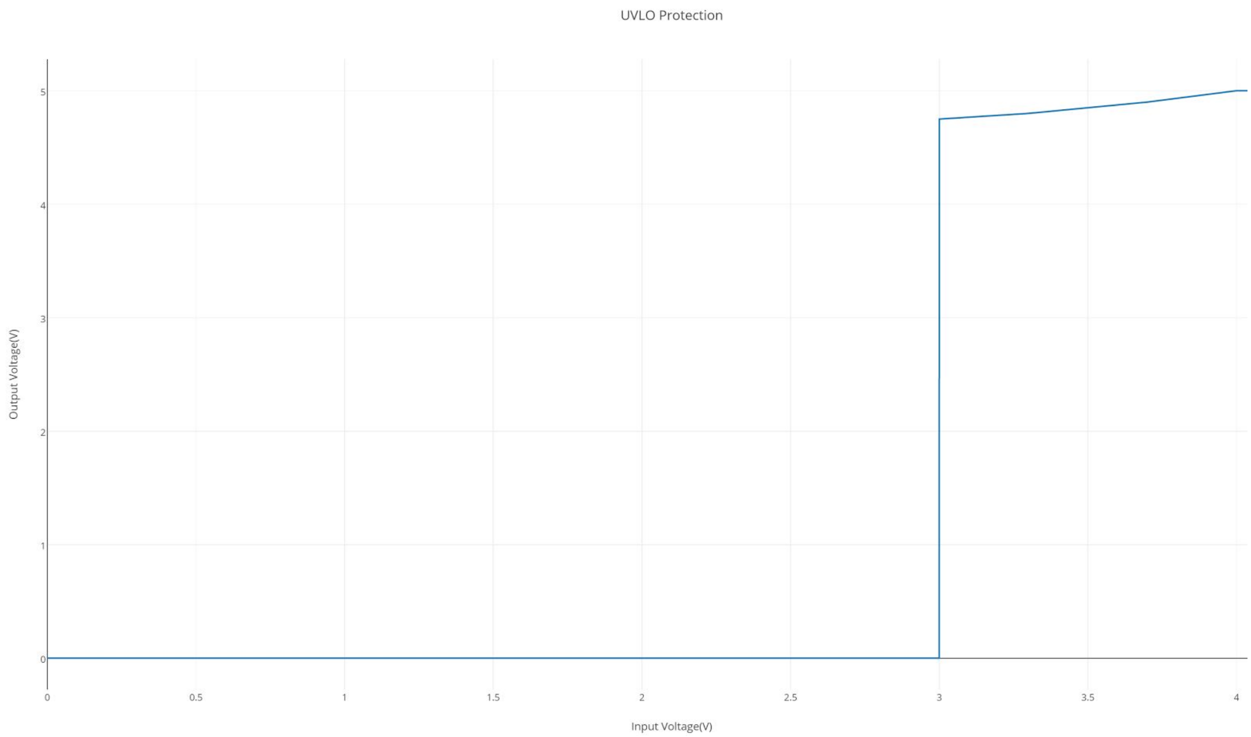

Figure 12: UVLO for the power chip

Figure 11 shows how UVLO will affect the power chip. When the battery is fully charged around 4.2V, the power output will be 5V. As the battery loses voltage, the output voltage will dip a small amount. When the UVLO cutoff is reached, the chip will stop the output voltage, which will occur at 3V.

The last source of errors we would like to consider is the bluetooth chip, and how increasing its distance will affect the latency of the devices. For the bluetooth chip we are using, the baud rate *r* will be 38400 and the bits transmitted *b* will be 8, according to the datasheet. Because the device is wireless, the propagation speed *s* will be the speed of light, 3×10^8 m/s. We will graph the latency for a range of distances up to 34 meters, the maximum operating distance we are considering for our device.

In order to calculate the propagation time, we will use the formula $\frac{b}{r} + \frac{propagation\ distance}{s}$

This is equal to the sum of the transmission delay and the propagation delay.

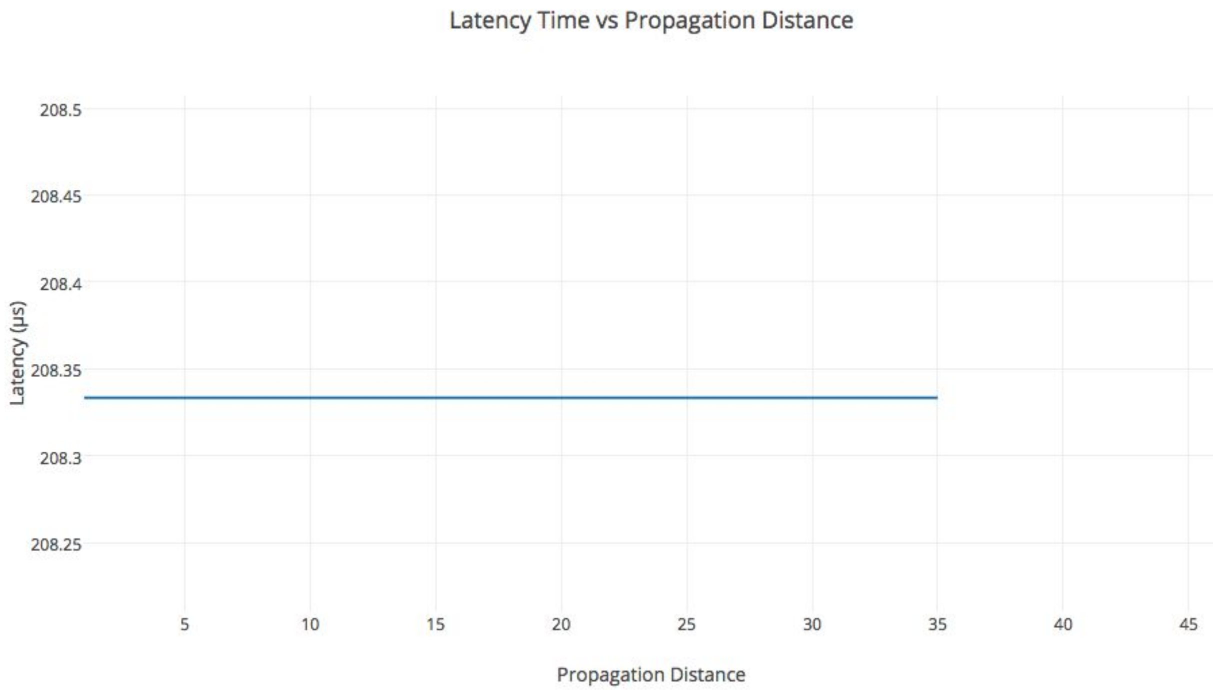

Figure 13: Latency vs Propagation Distance Graph

As shown, the distance between the devices should have a minimal effect on the latency, due to the fact that the transmission delay far outscales the propagation delay. As a result, below the cutoff distance of 35 feet, latency time should be on average $2x10^{-4}$ s, far below our requirement of 50 ms. Above the cutoff distance latency is theoretically infinite, as no transmission will occur.

4. Ethics and Safety

One of our constraints for this project is to make the project portable, so we figured to go forward with the Lithium Ion battery which is large enough in energy for portability. We have been accepted to use Li Ion batteries. For testing, we will use the bench initially if we can avoid the battery, but for testing the power circuit we will test with the battery. When we are done with the batteries, we will place the batteries in the battery bag and ammo box before storing in the yellow locker. We did look at certain safety conditions and procedures in the case of a fire or burning for Li Ion batteries[12].

We will be sure no one will be electrocuted by changing the circuit only when the power of the circuit is off. We will use the soldering station. The team will make sure the station stays nice and clean. If there seems to be something wrong with the station's equipment, we will notify the TA for help. For any case, we do know where and how to use the Fire Aid Stations, Spill Kit and Fire Box.

What was listed so far does follow the IEEE Code of Ethics(1) and (9)[13] To obtain complexity, our project may develop more by certification from our TA to make changes as well as getting advice and/or approval on materials to help make the changes necessary. This statement does show our respect and understanding of the IEEE Code of Ethics (3),(5) and (6)[13]. The team will be sure to be honest about the safety of the product for public use. When it comes down to the public, we will follow the IEEE Code of Ethics (1),(3),(7) and (8)[13]. If the occasion related to the IEEE Code of Ethics (2) and (4,) we will abide the code[13].

5. Cost

Labor costs will be standard wages for graduating EE majors from the University of Illinois. This data was taken from the University of Illinois Salary Averages[14]. The salary is $67,000 which is approximately $32 an hour. Accounting for overhead and a project time of 250 hours, the labor cost is as follows:

Member	Salary	Overhead	Hours	Total
Jonathan	$32	2.5x	250	$20,000
Colin	$32	2.5x	250	$20,000
Elizabeth	$32	2.5x	250	$20,000
Grand Total				$60,000

Description	Manufacturer	Supplier	Part No.	Quantity	Cost	Total
Speaker	CUI Inc.	DigiKey	CDMG15008-03A	1	$2.58	$2.58
Microcontroller	Atmel	Microchip Direct	ATxmega256A3	1	$8.93	$8.93
Voltage booster	Texas Instruments	Texas Instruments	TPS61090	1	$2.52	$2.52
Charging Module	Micro Chip	Digikey	MCP73871	1	$1.84	$1.84
Microcontroller	Atmel	Microchip Direct	AT32UC3L016	1	$3.67	$3.67
RGB LED	DHgate.com	DHL Free	386452483	146	$0.04	$5.84

Push Button	CW industries	digikey	GPTS203211B	64	$0.70	$44.80
3 Stage Switch	AliExpress	AliExpress	DSC0011	10	$3.42	$3.42
Bluetooth Chip		Newegg	HC-05	1	$5.41	$5.41
Custom Case				1	~$50	~$50
MOSFET	Fairchild	Digikey	2N3904FS-ND	1	$0.18	$0.18
Battery - 2500 mAH, Li-Ion		Gear Best	INR18650	2	$4.73	$9.46
Battery Connectors				1	~$5	$5
Grand Total:						$143.65

These are the grand totals for one controller. Since we will need two controllers, the price will be doubled.

Part	Cost
Labor	$60,000
Parts	$365.38
Grand Total	$61012.10

6. Schedule

Week	Jonathan	Elizabeth	Colin
2/27	Build the voltage regulator circuit for the battery.	Wire the MUX-decoder circuitry. Test circuitry functionality.	Connect MUX-decoder circuitry to the LED. Test circuitry functionality.
3/6	Finish battery circuit, start audio circuit.	Wire the button array circuitry.	Connect button array to microcontroller.

3/13	Continue building the audio circuit.	Program microcontroller to include LED controls.	Program microcontroller to manage button-LED interfacing.
3/20	Program microcontroller to control audio output.	Continue programming gameplay elements.	Continue programming gameplay elements.
3/27	Code the Bluetooth wireless functionality and test.	Connect bluetooth to microcontroller.	Code bluetooth wireless functionality.
4/3	Assemble physical container to attach the circuitry to. Test functionality.	Assemble physical container to attach the circuitry to. Test functionality.	Assemble physical container to attach the circuitry to. Test functionality.
4/10	Code LED functionality to change LED colors depending on different game pieces.	Assemble extra circuitry for additional functionality of the game.	Code microcontroller to manage LED array when game-mode switch is toggled.
4/17	Assemble and test prototype.	Assemble and test prototype.	Assemble and test prototype.
4/24	Debug battery circuit.	Debug LED and button array.	Debug bluetooth and microcontroller.
5/1	Prep final report and demo.	Prep final report and demo.	Prep final report and demo.

7. References

[1] "History of the Game." Battleship Game Page. Wordpress, n.d. Web. 08 Feb. 2017.

[2] "Battleship Rules." (1990): 1-3. Battleship. Milton Bradley Inc. Web. 7 Feb. 2017. <http://www.hasbro.com/common/instruct/Battleship.PDF>

[3] Zinevych, Sasha. "The Overview of Mobile Apps Market: Why You Should Enter Now."Business 2 Community. B2C, 1 Sept. 2014. Web. 08 Feb. 2017. <http://www.business2community.com/mobile-apps/overview-mobile-apps-market-enter-now-0994728#GPM1MphA HSbvVjpA.97>

[4]Member. "LED - RGB Diffused Common Cathode." *COM-09264 - SparkFun Electronics*. Sparkfun, n.d. Web. 25 Feb. 2017. <https://www.sparkfun.com/products/9264>.

[5]Industries, Adafruit. "Standalone Toggle Capacitive Touch Sensor Breakout." *Adafruit Industries Blog RSS*. N.p., n.d. Web. 24 Feb. 2017.

[6]"One-Channel Toggle Mode Touch Sensor." *Signal Processing: – Self-calibration, Auto Drift Compensation, Noise Filtering One-channel Toggle-mode* (2010): n. pag. Atmel. Web. 24 Feb. 2017. <https://cdn-shop.adafruit.com/datasheets/AT42QT1012.pdf>.

[7] (n.d.): n. pag. *Push Button Switch*. CW Industries, 22 Mar. 12. Web. <http://switches-connectors-custom.cwind.com/Asset/GPTS203211BR2.pdf>. <http://www.digikey.com/products/en/switches/pushbutton-switches/199/page/9?ColumnSort=1000011>

[8] Texas Instruments Incorporated. "Synchronous Boost Converter With 2A Switch."*Synchronous Boost Converters With 2A Switch (Rev. A)* (n.d.): n. pag. Texas Instruments. Web. 20 Feb. 2017. <https://cdn-shop.adafruit.com/datasheets/tps61090.pdf>.

[9]Scottie. "Lithium Polymer vs Lithium-Ion Batteries: What's the Deal?" *Scottie's Tech.Info*. Scottie's Tech Info, 21 June 2015. Web. 22 Feb. 2017. <https://scottiestech.info/2015/06/21/lithium-polymer-vs-lithium-ion-batteries-whats-the-deal/>.

[10]"Stand-Alone System Load Sharing and Li-Ion / Li-Polymer Battery Charge Management Controller." (2009): n. pag. *Microchip*. Web. 20 Feb. 2017. <https://cdn-shop.adafruit.com/datasheets/MCP73871.pdf>.

[11]Administrator. *RGB Triple Color LED With Diffused Lens* (n.d.): n. pag. *100LED.com*. Web. 20 Feb. 2017. <https://www.sparkfun.com/datasheets/Components/LED/YSL-R596CR4G3B5W-F12.pdf>.

[12]Buchmann, Isidor. "BU-304a: Safety Concerns with Li-ion." Safety Concerns with Li-ion Batteries – Battery University. Cadex Electronics Inc, 27 Sept. 2016. Web. 08 Feb. 2017.

[13]"IEEE IEEE Code of Ethics." IEEE - IEEE Code of Ethics. IEEE, 2017. Web. 09 Feb. 2017.

[14] Services, Engineering IT Shared. "ECE ILLINOISDepartment of Electrical and Computer Engineering." *Salary Averages :: ECE ILLINOIS*. ECE Illinois, 2017. Web. 25 Feb. 2017. <https://www.ece.illinois.edu/admissions/why-ece/salary-averages.asp>.

[15]"2017 Dhl Free F5 5mm 4pins Red Green Blue Rgb Led Common Cathode Tricolor Emitting Lights Diodes Round Water Clear Lamp From Yushitech, $43.21." *DHgate*. N.p., n.d. Web. 14 Mar. 2017.

[16]"Candela." *Wikipedia*. Wikimedia Foundation, 12 Mar. 2017. Web. 14 Mar. 2017.

[17]"Mcd to Lumens Converter." *Ohm's Law Calculator*. N.p., n.d. Web. 14 Mar. 2017.