Portable Battleship Display

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

1. Introduction 2
   Objective 2
   Background 2
   High-level Requirements 3

2. Module Description 5
   1. User Interface 5
   2. Speakers 6
   3. Microcontroller 7
   4. Bluetooth Chip 7
   5. Battery 8
   6. Boost Converter and Charger 9
   7. Physical Display 11

3. Tolerance Analysis 11

4. Ethics and Safety 12

5. Cost 13

6. Schedule 13

7. References 14
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 implementation of 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

Figure 1: Physical Design Sketch
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

1. User Interface

For the user interface, we needed to decide how to control our device. Initially we wanted to incorporate a button array in order to control the gameplay. This will add an interesting layer of complexity to our project as well as simplifying the design process.
**Figure 2: LED Button Array**

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Verifications</th>
</tr>
</thead>
</table>
| 1. Simple, easy to control user interface.  
2. No additional pieces required.  
3. Push pads ~1cm² | 1. Player and enemy’s display updates almost instantaneously after a button press with low latency.  
2. Player’s display updates pieces when buttons are pressed.  
3. Able to switch modes via button press.  
4. Touchpads are .36cm² and buttons are 1.2 cm² |

The Figure x 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.

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[4]. AT42QT1012 is very expensive actually, $5.95 a piece. For one board makes $595 with 100 pieces being used. We are currently looking into alternatives, however we need something just about cm squared pad finger contact and is lightweight[4]. The other motivation to us the AT42QT1012 is it comes with an led on it for when the button has pressed and ‘de-pressed’ so to turn on and off respectively[5]. This will be good for the board on saving up on area. Not only is the AT42QT1012 small (.6x.6cm squared), but the board having an attached, programmable light allows us to save on parts[5]. 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[5]. Figure x of the error shows that with defining a threshold Voltage to be set to observe signal will this establish the sensativity of the device. The output displays the actual reading that will be displayed if threshold is set over hysteresis as such[5].

![Figure x: error][5(12)]

The touchscreen being as expensive as it is, we will probably go with a button array, ~$50 for an entire board. For instance, COM-09190 is 0.12 cm squared, and costs $0.50 per button[6]. The area of the interface board should then exceed 12cm squared plus LED’s array 33.64cm squared, 45cm squared. 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.

2. Speakers
Figure 3: A speaker module to operate the speakers

For the speakers, we wanted to go with a lightweight, cheap part that is simple to incorporate into our design. We opted for a second microcontroller to store the sounds and to do 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.

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 10,000 Hz to get the full range of hearing for the speaker. To calculate values for RC, we’ll be using a 1 \( \mu F \) capacitor:

\[
\frac{1}{2\pi RC} = \frac{1}{10000} \\
R = \frac{10}{2\pi \times 10^{-6}} = 15.91\Omega
\]

Therefore, the capacitance will be 1\( \mu F \) and the resistance will be 16 ohms.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Verifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Output is audible from 5ft</td>
<td>1. Verify audio is clear 5 feet away</td>
</tr>
<tr>
<td>2. Microcontroller is capable of driving a PWM signal</td>
<td>2. Microcontroller needs a PWM signal</td>
</tr>
<tr>
<td>3. Sound is clear and the audio sample is similar to the original sound within 90%</td>
<td>3. Sample both sounds, verify that the original sound is 90% similar to the speaker sound</td>
</tr>
</tbody>
</table>

Control System

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 100 RGB LED displays, 18 additional RGB LEDs to be used as a legend, and receive inputs from each of the 100 buttons. Therefore, the microcontroller will need to be at least 32 bits.

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.
Requirements | Verifications
---|---
1. Device must be able to communicate and interface with each other component.  
2. Device can interact with a UART Bluetooth chip  
3. Device can drive 200 RGB LEDs and 100 buttons | 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. Make sure the device is at least

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 ft apart.  
2. Low latency, data travels between devices quickly. | 1. Device will interface properly 30 ft apart.  
2. Verify there is less than 50 ms between send and receive time

Power Supply

![Power Supply Diagram](image)

Figure 4: Typical application of the TPS61090 chip[7]
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. [8] There will be two of these batteries wired in parallel, to double the battery output.

6. Boost Converter and Charger

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[7] the resistances must be calculated for the undervoltage lockout and voltage regulation output. To calculate the UVLO resistors:

\[ R1 = R2 \times \left( \frac{V_{\text{Coeff}}}{500\text{mV}} - 1 \right) \]
Texas instruments recommends a voltage for R2 of 390kΩ. After the battery hits 3.0V, it has less than 5% of its charge remaining[7], therefore this will be our cutoff voltage.

\[ R1 = 390k \times \left( \frac{3}{5} - 1 \right) = 1.95M\Omega \]

These two resistors coincide with resistors R2 and R1.

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:

\[ R3 = R4 \times \left( \frac{V_{out}}{500mV} - 1 \right) \]

The TPS61090 document recommends using a 200kΩ resistor for R4. Calculate R3 as follows:

\[ R3 = R4 \times \left( \frac{7}{3} - 1 \right) = 1.8M\Omega \]

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

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Verifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Sustain a 3.2 &lt; 3.3V &lt; 3.4 and 4.5 &lt; 5V &lt; 5.5 output</td>
<td>1. Test the voltages are within range</td>
</tr>
<tr>
<td>2. Shut off the battery when the voltage is less than 3.5V</td>
<td>2. Test current drops off if battery is below 3.5V</td>
</tr>
</tbody>
</table>

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.
To calculate the voltages to program VPCC to 1.3V, we check the MicroChip MCP73871 manual [9]. For a voltage that's 1.3V less than 5V, we use VDR to get this voltage.
\[ V_{VPCC} = \left( \frac{R_2}{R_1} \right) \cdot V_{In} \]

The input voltage of the typical USB charger is 5V. Setting R2 to 100k gives us:
\[ 1.3 = \left( \frac{100k}{100k+R_1} \right) \cdot 5 \]
\[ .26(100000 + R_1) = 100000 \]
\[ R_1 = 284.6k \Omega \]
Since there is no resistor this precise, we will use a 280k Ohm resistor.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Verifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Charging takes less than five hours from empty</td>
<td>1. Test a full charge cycle to verify the duration fits the requirement</td>
</tr>
<tr>
<td>2. Charging stops at 4.2V ±.1V</td>
<td>2. Verify the circuit doesn’t overcharge the battery by not charging above the desired voltage</td>
</tr>
</tbody>
</table>

8. Physical Display

We would like to create a light, concise display with durable material. The display should be large enough to contain all the circuitry.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Verifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Relatively light display.</td>
<td>3. Design doesn’t exceed 10 Lb.</td>
</tr>
<tr>
<td>4. Visual elements sync with gameplay, is engaging and interesting.</td>
<td>4. Light changes demonstrate game mechanics.</td>
</tr>
</tbody>
</table>

For the Physical Display, we will use the 74AC154 4 to 16 mux. This will lead into a 10x10 array of RGB LED. For which now we are looking into COM-09264 for the LED’s[10]. As mentioned in the interface section of this paper, these may or may not be used in the interface display since we are not entirely settled on buttons or the touchscreen panels. The RGB LED’s colors leads all have max current of 20mA, but the forward voltage varies between 2.0V and 3.2V and the different colors. We will have to note the minimum voltage for these leads will have to be 3.2V just so we know all LEDs will turn on, and a maximum voltage 1.8 across the LEDs when the lights should be off[11]. The RGB LEDs are a little more expensive, ~4x more expensive than the buttons ($1.95 each, $195 per display). 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.
3. Tolerance Analysis

**Speakers** should be clear and at a controlled voltage.
1. Speakers may have a higher voltage.
2. Speaker may have background noise which maybe hard to diminish

**Power supply** will satisfy the circuit’s modules to act appropriately
1. Could get damaged by too high heat if connected incorrectly, may damage other devices if overheat.
2. With the number components to power, Voltage level may exceed.
3. May set the circuit too high, damaging devices.
   \[ 3.7V(\text{Power Supply}) > (\text{Bluetooth Chip}) + (\text{Speaker}) + (\text{interface}) \]

User interface is simple and engaging
1. The buttons could overheat if not powered correctly

Control System simplifies the communication between the modules within a single board
1. If not coded correctly, may lead to more miscommunication.
2. If modules exceed the physical tolerances of the microcontroller board, the board can be damaged and lead to more miscommunication

**Bluetooth Chip** for communicating between boards
1. If powered too high, could break the device
2. May be complicated for the players to keep up with the game if the two players are meditating on the boundary of the bluetooth chip(30ft)

**Physical Display** to display progression in the game against opponent
1. Not designed correctly, could lead to the giveaway of the opponents layout.
2. The game may lose information on game ‘score’ if the two players are meditating on the boundary of the bluetooth chip(30ft)

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:

<table>
<thead>
<tr>
<th>Member</th>
<th>Salary</th>
<th>Overhead</th>
<th>Hours</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jonathan</td>
<td>$32</td>
<td>2.5x</td>
<td>250</td>
<td>$20,000</td>
</tr>
<tr>
<td>Colin</td>
<td>$32</td>
<td>2.5x</td>
<td>250</td>
<td>$20,000</td>
</tr>
<tr>
<td>Elizabeth</td>
<td>$32</td>
<td>2.5x</td>
<td>250</td>
<td>$20,000</td>
</tr>
<tr>
<td>Grand Total</td>
<td></td>
<td></td>
<td></td>
<td>$60,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Manufacturer</th>
<th>Supplier</th>
<th>Part No.</th>
<th>Quantity</th>
<th>Cost</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speaker</td>
<td>CUI Inc.</td>
<td>DigiKey</td>
<td>CDMG15008-03A</td>
<td>2</td>
<td>$2.58</td>
<td>$5.16</td>
</tr>
<tr>
<td>Microcontroller</td>
<td>Atmel</td>
<td>Microchip Direct</td>
<td>ATxmega256A3</td>
<td>1</td>
<td>$8.93</td>
<td>$8.93</td>
</tr>
<tr>
<td>Voltage booster</td>
<td>Texas Instruments</td>
<td>Texas Instruments</td>
<td>TPS61090</td>
<td>1</td>
<td>$2.52</td>
<td>$2.52</td>
</tr>
<tr>
<td>Charging Module</td>
<td>Micro Chip</td>
<td>Digikey</td>
<td>MCP73871</td>
<td>1</td>
<td>$1.84</td>
<td>$1.84</td>
</tr>
<tr>
<td>Part</td>
<td>Cost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Labor</td>
<td>$60,000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parts</td>
<td>$1012.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grand Total</td>
<td>$61012.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 6. Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Jonathan</th>
<th>Elizabeth and Colin</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/27</td>
<td><strong>Build the voltage regulator circuit for the battery.</strong></td>
<td><strong>Wire the MUX-LED circuit, test for functionality.</strong></td>
</tr>
<tr>
<td>3/6</td>
<td><strong>Continue building the battery circuit, start building the audio circuit.</strong></td>
<td><strong>Wire the button array circuitry.</strong></td>
</tr>
<tr>
<td>3/13</td>
<td><strong>Continue building the audio circuit.</strong></td>
<td><strong>Program microcontroller to incorporate gameplay elements.</strong></td>
</tr>
<tr>
<td>Date</td>
<td>Task Description</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------------------</td>
<td></td>
</tr>
<tr>
<td>3/20</td>
<td>Program microcontroller to control audio output.</td>
<td></td>
</tr>
<tr>
<td>3/27</td>
<td>Code the Bluetooth wireless functionality and test.</td>
<td></td>
</tr>
<tr>
<td>4/3</td>
<td>Assemble physical container to attach the circuitry to.</td>
<td></td>
</tr>
<tr>
<td>4/10</td>
<td>Add extra gameplay features to expand the functionality of the device.</td>
<td></td>
</tr>
<tr>
<td>4/17</td>
<td>Assemble and test prototype.</td>
<td></td>
</tr>
<tr>
<td>4/24</td>
<td>Troubleshoot errors in the prototype and fix them.</td>
<td></td>
</tr>
<tr>
<td>5/1</td>
<td>Prep final report and demo.</td>
<td></td>
</tr>
</tbody>
</table>
7. References


<http://www.hasbro.com/common/instruct/Battleship.PDF>

<http://www.business2community.com/mobile-apps/overview-mobile-apps-market-enter-now-0994728#GPM1MphAHSbvVjpA.97>


<https://scottiestech.info/2015/06/21/lithium-polymer-vs-lithium-ion-batteries-whats-the-deal/>.


