Acoustic Battleship

Liam Weston, EE, Justin Forgue, CSE, Adrian Sanmiguel, EE, and Xinyu Cao, CSE

***Abstract*— The traditional battleship game is an easy and fun two-player game. As engineers, we want to innovate the game with some technological taste. We designed a sound localization system which converts the traditional battleship game to a precision-based game. We call this new game “Acoustic Battleship” as the game is designed based on an acoustic localization system. Players attack their opponent’s board with a ping pong ball and our system determines the location where the ball lands based on the time of arrivals received by sensors placed around the board; LEDs then indicate hit or miss with distinct colors, based on pre-set target positions at the start of the game. The core of our project is to precisely record the time of arrivals at every sensor for each hit and to calculate the sound location using triangulation.s**

I. Introduction

THE traditional boardgame has fallen out of favor in the gaming market. We want to put a technological spin on the game of Battleship.

A. Significance

In the current era of electronic video games, one may ask, “where have the classic board games gone?” One may say that the classic board games such as Monopoly, Risk, Clue, and even Battleship have become outdated. Actually, within the past decade, there has been an increase in sales of tabletop games each year, especially around the holiday season [4]. What we are trying to do is add an electronic medium to the classic game of Battleship to enhance the entertainment amongst players. Although the classic game is actively being played today, we are trying to introduce a technologically modified version of it that will be more intriguing towards the audience of this generation.

B. Context and Existing Products

This isn’t the first time someone thought of bringing electronics and classic board games together. In 2017, A senior design project created at the University of Massachusetts Amherst modified the classic board game of Castle Quest using electronics and software to provide group-centric technological entertainment [5]. Throughout the years, we have seen mobile application versions of these board games such as Scrabble, Monopoly, and even Battleship rise in popularity. One of the critical viewpoints on these digital versions of board games is that it doesn’t bring people and families together in one place like their traditional counterparts [1].

There are electronically developed board games such as Electronic Battleship: Advanced Mission which has special sound effect features that respond to the player’s imputed coordinates [2]. Another product that has been electronically implemented is Monopoly, where you are now able to use a bank card to make transactions rather than using Monopoly dollars [6]. Acoustic Battleship takes a different approach, it turns the classic board game into an electronically assembled accuracy-based game that will be interactive and entertaining. We hope to make this technology adaptive enough that it can be implemented in other classic games such as darts or ping-pong as well.

C. Societal Impacts

We hope that Acoustic Battleship will make a social impact by bringing a more interactive game to the table. A game that can bring the community together to enjoy a new version of the classic board games that we love. We hope to see this project be used in public settings such as arcades and game rooms. Hopefully, one day we can have a version portable enough to be carried with you and played anywhere.

D. Requirements Analysis and Specifications

We want our design to meet the specifications shown in Table 1. The accuracy of our times captured and the algorithm will play a key role in calculating the location of where the projectile hits on the playing surface. We also want our project to be able to make the calculations quickly so the response time from the impact of the projectile to displaying the location of that impact will be minimal.

Requirement	Specifications	Value
Accuracy	Distance Error	<= 5 cm
Responsiveness	Response Time	<= 500 ms

Table 1: Requirements and Specifications

II. Design

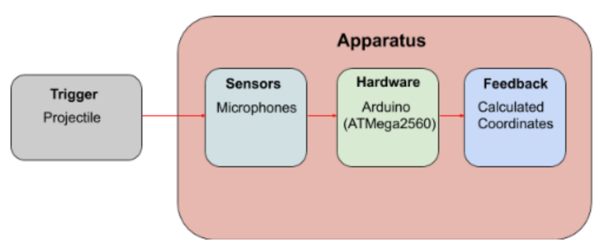

Figure 1, Block Diagram

A. Overview

Our design can be broken down into three essential categories. Sensing, processing, and Interfacing. Our current solution for these categories are microphones [7] for sensing, a microcontroller [8] for storing and processing data, and LEDs the react to the output of our microcontroller. Our microphone sensors will receive sound produced by a projectile striking the surface of our playing surface. Using an ADC circuit we convert our analog sound signals from the microphones to 3.3V digital signals. Once a change in one of these signals occurs it triggers our timer circuit to store the time of the change in memory. These stored times are representative of our Time Difference of Arrivals (TDOAs). By polling the memory locations that these times are stored we are able to accurately measure our TDOAs. These signals are then computed in our multilateration algorithm to produce the coordinates of the sound source. Once we have the coordinates of the sound source we will light up the LED located at those coordinates on the board. Byte-addressable LEDs (i.e. WS2801 4-wire controlled by SPI) are used as part of the user interface to show the location hit and status of the hit (hit or miss). The color of the LED that corresponds to coordinates calculated will depend on whether or not that location currently contains a battleship (we anticipate using contrasting colors such as red and green to mark a “hit” or a “miss”). To position a player’s battleships, we would like to implement a controller for each player and an additional, smaller coordinate system that indicates the positions of a player’s own battleships.

B. Sensors

Our design implements 8 omni-directional microphones that are capturing the sounds produced on the playing board between the frequencies of 5kHz and 10kHz. We found that the distinct voltage signal of the projectile hitting the surface of the playing board is adequate for an Analog to Digital Conversion (ADC) by implementing a Schmitt Trigger (refer to Appendix B). The voltage consistently reaches levels greater than 2 volts. This allows us to implement a Schmitt Trigger with a supply voltage of 3.3 Volts that will consistently produce an ADC for our signal in all coordinates within our plane. To ensure we are able to process the data we are receiving at the microphones, we produced a sound source at 12 different coordinates on our prototype playing surface. Using a Saleae Logic Analyzer (introduced to us in ECE353), we measured the TDOA of two microphones. After measuring TDOA versus theoretical TDOA values, we can confidently calculate the position of a sound source.

C. Clock

To begin the data collection, we pre-scaled our system clock to 250kHz. In theory, this clock frequency will allow for roughly 2mm accuracy (much greater than we initially anticipated). We reached this figure by calculating each clock tick to be 4 microseconds, which in turn, equates to a maximum distance error of 1.372 millimeters. We anticipate this resolution to increase our accuracy of localization greatly when compared to our initial specification.

D. Timers

We used a 16-bit timer for each microphone in our system. 16 bits gives us more than enough accuracy for each recorded time. Each timer also had its own register to save the current time when it was triggered. Since the time difference between each trigger event (a microphone signal going low) is faster than one cycle of any reasonable single-core processor for this project, a single shared register would be overwritten before we could read it. This would result in no perceivable time difference between each trigger event. Having individual timers allowed us to read each register several cycles after they were set, preserving the recorded time difference. The accuracy and independence of each timer allowed us to read such minute differences in time.

E. Algorithm

To produce a calculation for our TDOAs that will produce an accurate coordinate, we derived a two-dimensional multilateration algorithm.

Our derivation of a two-dimensional planar algorithm for multilateration uses TDOAs and energy waves of a known propagation speed (the speed of sound in dry air at 20°C, 343 $^{m}/_{sec}$). To comprehend the algorithm we will be implementing, it is useful to refer to . As can be seen, our initial time of travel *a* (the red circle) to our first microphone is an unknown. This is evident, as we do not know when the projectile will strike the surface of our playing surface, we only can mark the arrival of the energy wave in time. We do know the *time difference of arrival* of that same energy wave to our second and third microphones, located at either end of the hypotenuse of our triangle in Figure 1.

These known values are represented by *b* and *c* in Figure 1. It is then helpful to imagine TDOAs *b* and *c* as the radii of circles generated by the formula: $d_{b,c}$ (distance) = rate (the speed of sound in dry air at 20°C, 343 $^{m}/_{sec}$) x time ($TDOA_{b,c}$).

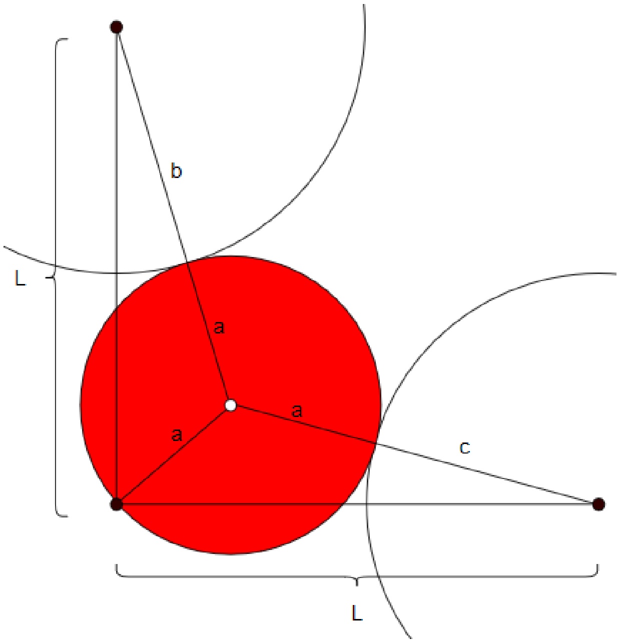

Figure 2. A visual representation of the initial sound source being received at 3 microphones arranged in an isosceles triangle

These calculations produce two known radii (though without a known angle of arrival), we can produce the initial location of sound using multilateration. For a comprehensive visual of our process, please refer to Figure 2. Through analytical geometry, we can determine the precise point of a sound source. If we allow the unknown distance of *a* to go to zero, we are left with two known distance *b* and *c*, which are the radii of two circles whose origin is at a given microphone, and whose perimeter coordinates are stored in an array, we can form a new circle, with radius *d*, around the microphone that first receives the sound signal. As we increase *d*, we increase both *b*, and *c*, by *d*. As *d* increases, we will eventually find a common intersection of the three circles. This intersection is the source of the sound.

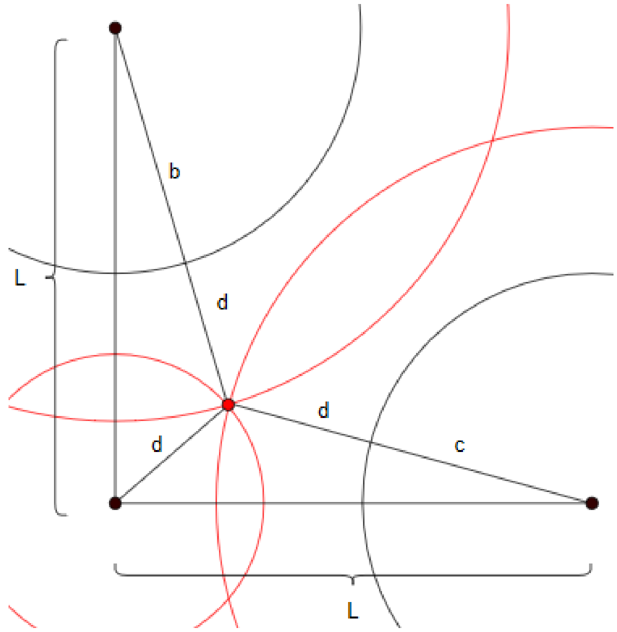

Figure 3. Iterating through different values of d produces the unique intersection.

We have produced a C++ function and in Appendix B have a further explanation for the experiments we performed to ensure our calculations are accurate.

C. *User Interface*

Our User Interface will be responsible for providing feedback for the end user, as well as allowing the end user to view the positions for their upon striking the surface of the board, our user interface would display the LED response necessary to indicate if the coordinate contained a target or if the coordinate was empty. To achieve this, we anticipate using byte addressable LED strips that allow us to choose the color and location. At the current time, we anticipate randomly generating the positions of each battleship. As our ambitions for this project have grown, we need to first implement the core of the project before reaching our stretch goals.

III. Project Management

MDR goals	Status
System on a single board 5*5blocks, with each block 10*10cm	Accomplished
Arduino as microcontroller	Accomplished
Calculate coordinates	Not Working correctly
light up LED accordingly	Not Yet
Error distance less than 8 cm	Accomplished
Response time less than 1 s	Accomplished

Table 2. A review of or proposed MDR goals

By MDR, our team have built a wood playboard whose dimension is 50 centimeters by 50 centimeters. The board is divided into 5 columns by 5 rows. We used Arduino Mega2560 as the microcontroller development board. We developed an algorithm that provides an analytical solution of calculating the coordinates of the positions of the sound source given an ideal input of the time of arrivals of all four microphone sensors. We simulated the algorithm in both MATLAB and C++ and achieved 100% accuracy with ideal inputs which are generated from known dimensions of the board and the speed of sound in the air. We programmed our Arduino Mega2560 microcontroller to capture the time of arrivals; we then integrated our algorithm into the code for microcontroller to have it calculate the results of sound source coordinates based on the predefined coordinate plane.

However, the results generated by our system are incorrect. This might be caused by inaccurate time of arrivals captured by the system or the incompatibility of the algorithm used due to neglected conditions. We found that the biggest challenges come from making four registers of the microcontroller listen to four microphones independently and capture each time of arrival asynchronously.

We weren't able to move on to displaying our results with LEDs because of failure to get accurate time of arrivals. We also could not calculate our error of distance for the same reason. However, our algorithm model makes it clear that as long as we manage to get the accurate time of arrivals, the error of distance should be negligible.

We did get a response time less than 1s. Actually, we had it under 500ms for our MDR prototype. There might be extra delay of time when LEDs are connected to the system, but the entire response time should still be reasonably small.

IV. Conclusion

We plan to keep working on the Arduino code to allow the microcontroller to capture the time of arrivals as accurately as possible, which will involve a lot of reading of datasheets, experiments of different approaches and testing of the accuracy. We also consider this as our biggest difficulty. Once we are able to obtain accurate time of arrivals and calculate the accurate coordinates, we will then move on to displaying the results in LEDs, scaling up the board and building a two-board system as required for the game. Table 3 is a Gantt Chart which displays our plan of task distribution for next semester.

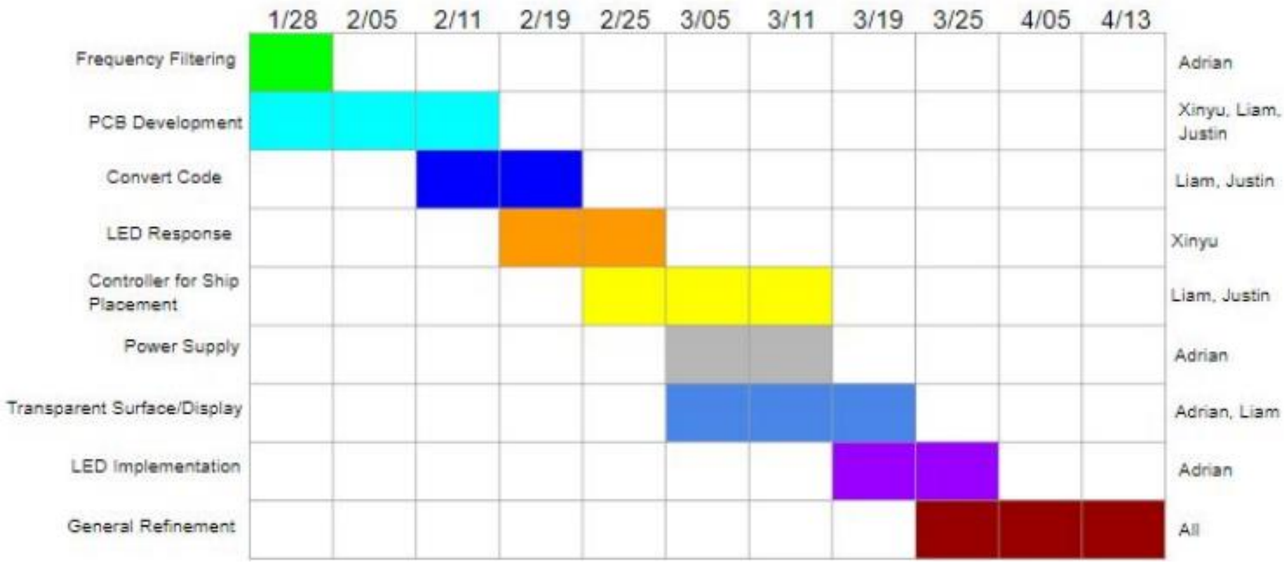

Table 3. Gantt Chart for the upcoming semester

References

[1] Rioux, Sean. "Classic Family Board Games in Electronic Editions for the Holidays." Apartment Therapy, Apartment Therapy, LLC., 10 Aug. 2019, W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[2] "HASBRO Battleship Advanced Mission Instructions Manual." *Manuals Library*, https://www.manualslib.com/manual/580178/Hasbro-Battleship-Advanced-Mission.html.

[3] R. Kaune, "Accuracy studies for TDOA and TOA localization," 2012 15th International Conference on Information Fusion, Singapore, 2012, pp. 408-415.

[4] Whitten, Sarah, and Cnbc. "Why Board Games Are On Trend for the Holidays." NBCNews.com, NBCUniversal News Group, 21 Dec. 2016, www.nbcnews.com/business/consumer/why-board-games-are-trend-holidays-n698786

[5] "Team Castle Quest." Team 13, University of Massachusetts Amherst, 2017, www.ecs.umass.edu/ece/sdp/sdp17/team13/.

[6] Smith, Chris. "Monopoly Ultimate Banking Goes Cashless to Oust the Family Cheat." Trusted Reviews, 15 Feb. 2016, www.trustedreviews.com/news/monopoly-ultimate-banking-goes-cashless-to-oust-the-family-cheat-2946006.

[7] CUIINC, "ELECTRET CONDENSER MICROPHONE ," CMA-4544PF-W datasheet, Sep. 2013.

[8] Arduino, "Arduino Mega 2560 Datasheet,", July. 2004.

Appendix

A. Design Alternatives

We considered having two displays, each facing one player, to show the shots and results. The purpose is to only show the players their scores (hit or miss) while preventing them from peeking their opponent's target map. Then we changed our mind and thought that the blocking screens in the middle of two boards are not necessary and distracting to players. We decided to just show the relevant scores on the board without disclosing the full target map as the system will keep track of the progress of the game and let the players know if the game is over. However, private displays could be used to show the players a full map of the targets, with the updated hit or miss.

We also considered implementing a sensor network consisting of infrared LEDs and light sensors to detect position rather than an acoustic method. However, we felt this system would be overkill and unimpressive for our application as is has been done many times before. Additionally, managing and isolating a positioning system based on light would have been more difficult in our environment.

B. Testing Methods

Experiment 1 – Time Reality Check

To test the quality of the digital input for microcontroller, we monitored the input signals with logic analyzer, obtained the time difference of the edges of arrival at different sensors, calculated the difference of distances time travels and compared it to the actual difference of distances from the sound source to each sensor. Figure 4 shows an example of how we measured the TDOA between two microphones using logic analyzer.

We found that the results calculated are very close to the actual values when sound source happens at a reasonable distance (not too close or too far) from the sensor. For this experiment, we are essentially testing the quality of our ADC component. The results indicated that the output of the ADC was of good quality. The restriction of the distance from sound source to microphone might be caused by the location of the sensor network on the board. Further testing is required to find out what distance range will give the most accurate results and make our play area mostly fall in that range.

Our eventual output on the logic analyzer produced satisfying results that would allow us to readily capture time differences between sound sources striking the playing surface.

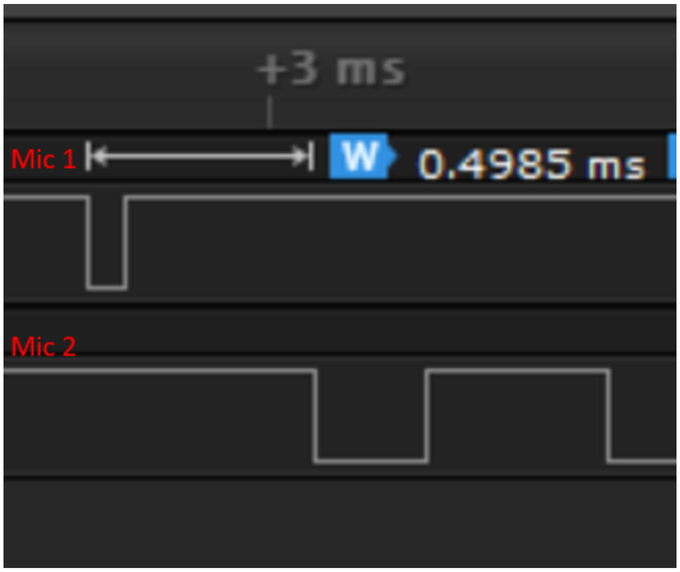

Figure 4. the TDOA between the Digital Signals Processed from the Analog Output Signals of Two Microphones

The purpose of this experiment was to ensure we will be able to receive consistent signals to all microphones with a measurable TDOA. This is an imperative portion of our testing, as without the ability to measure the TDOA with good accuracy, we will not be able to produce accurate coordinates. Upon completion of our coordinate calculation, we will test further to ensure accuracy.

Experiment 2 – Algorithm simulation

To test the validity of our analytical algorithm, we implemented the algorithm in MATLAB first and then translated into C++ which is needed for Arduino. We simulated some sets of ideal time of arrival inputs and used those as the input to our algorithm. The output of our algorithm matched with the original data we used to the simulated time inputs, which proved that our algorithm is able to calculate accurate coordinates of the sound source given accurate inputs of time of arrivals. No further testing is needed until we got our microcontroller to capture the correct time of arrivals. For this experiment, we were testing on the algorithm running on the microcontroller.

C. Team Organization

We consistently communicate on a regular basis on what is currently being worked on. Discussions are held through group messenger, and we collaborate together in graphical web-based interfaces which currently include Github and Slack. We also meet once a week with our advisor to discuss our updated progress and our plans moving forward. Communication is prioritized within our group in order to maintain structure and organization, as well as provide help and feedback when one comes across a problem. Some issues have come up along the way as far as completing tasks according to our group's desirable deadlines. We strongly plan moving forward to time manage more efficiently in order to provide positive feedback and complete the project by CDR.

D. Beyond the Classroom

Adrian Sanmiguel - What I have learned on this project so far is better understanding in wave shaping applications towards the analog behaviors being outputted through the use of electret condenser microphone devices. I look forward to developing skills and knowledge about filtering specific sound frequencies to provide an output only at frequencies made of a ping-pong hitting a specific surface.

Liam Weston - I have utilized analytical geometry in my first semester of SDP. I have applied plane wave theory and used my understanding of electronics 1 to produce tangible results that will be beneficial to our group going forward. I hope my experience as group manager continues to grow myself as a person and a professional (as well as developing the managerial skill as a whole).

Xinyu Cao - I learned about how interrupts work in Arduino Mega2560 and setting modes for the microcontroller through bit manipulation according to the datasheet. I also learned how to break an engineering problem into theoretical math models and applied the math model back to solving the problem; this gave me good experience in applying what I learned to a real world problem, which was very important in helping me grow as an engineer. Internet has always been a great source to learn from, so are my professors and my fellow engineers.

Justin Forgue - As far as skills I had to learn, C++ and microcontroller programming were definitely the top two. I certainly also had to learn how to read datasheets and when to ask for help! I was also reminded of how important testing is, never assume something works. Our advisor has been a great resource to me and really helped steer me in the right direction. Without him I would have bought a better, more expensive microprocessor and still have the same problem.

The project has been very realistic for me as working with ASICs and other embedded systems is what I want to do. Working on similar problems here has really helped me understand the embedded systems world. Beyond that I plan to create our solution with an FPGA, which will allow me to learn VHDL and get a better understanding of that side of embedded systems too.