Let's Have a Blast!
Laser Tag

Department of Electrical Engineering and Computer Science
University of Central Florida

Group 10
Marco Montero - Electrical Engineering
Anuj Yamdagni - Computer Engineering
Shannon Fies - Electrical Engineering
Karlie Brinthaupt - Electrical Engineering

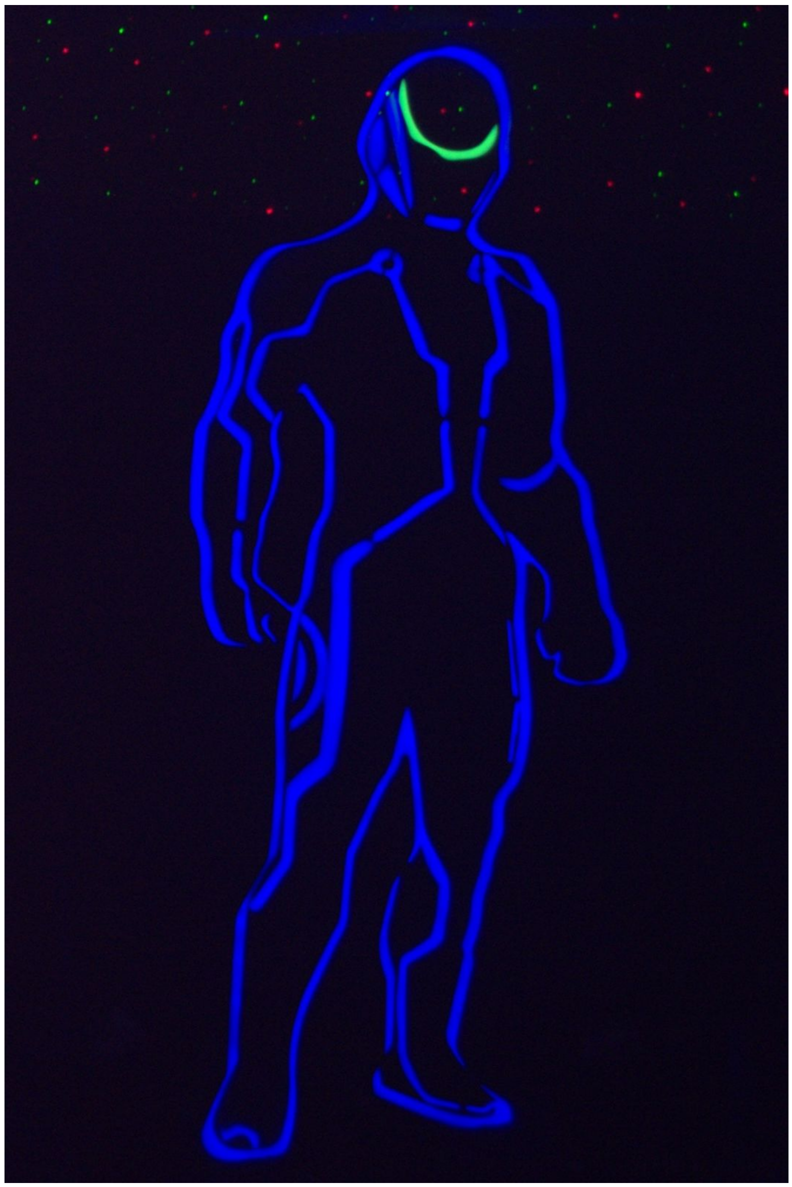

Table of Contents

Figure Index

Table List

1.0 Executive Summary

As videogames and electronic devices become more and more popular, physical activity in kids is on the decline. Traditional games of tag or playgrounds can not compete with the allure of modern technology within videogames. Our team hopes to bridge that gap by combining technology with physical activity.

Laser Tag has been a favorite of kids and adults alike, however it always seems to be limited to expensive arcades with limited game time. This project seeks to make laser tag available to more users at a lower price. By creating a cheap, standalone system that is easy to use, we can bring a fun technology that incorporates indoor and outdoor play.
Our intended audience are kids and adults who are looking for casual entertainment that is not too expensive or difficult to set up. This sets up the project for several different requirements. The laser tag blaster will need to be a light, portable blaster so that it is easy to carry around and use. All the technology should be housed within the blaster, so that the blaster can be taken anywhere. Since kids will be using this blaster in outdoor settings, the outer shell of the blaster should be clearly identifiable as a toy and not a real weapon.

Laser Tag is meant to be played with multiple people, thus the blasters should be able to synchronize with each other using a standalone network. The network should be a mesh network so that the group of players are not tied to any one location. The networking should allow the synchronization of game events such as the start and end of the game, as well as communicating the team score across to each player.

Another important aspect of this project will be to include an easy to use User Interface (UI). This will allow users to create and join games easily and better immerse themselves in the gameplay. The blaster should include a small embedded display to show players information about the game, as well as a menu for setup. The blaster should include buttons for interacting with the menu.

2.0 Project Summary

This section will go through an overview of the project and includes the following topics:

- Motivation
- Goals and Objectives
- Game Mechanics
- System Flowcharts
- House of Quality
- House of Quality Breakdown

2.1 Motivation

Laser tag has been a favorite for kids and adults for many years, however there has been little development in the game to keep up with modern technology. Advances in smartphone technology has driven significant advances in embedded electronics, including networking, batteries, sensors and onboard computing power. However none of these advances in the smartphone technology have been applied to the game of laser tag. Not only has embedded technology significantly advanced over the past decade, the cost of embedded electronics has significantly dropped. We believe we can apply these advances to bring the game of laser tag up to date while making it much cheaper for people to play.

2.2 Goals and Objectives

Hardware: Each gun's electronics should be all contained onto a single PCB that will be mounted within the shell of the gun.This will make assembly of the gun much simpler for mass production, as well as making testing each board easier as it will not require full assembly of the gun. The hardware inclusion should support future additions to software and provide multiple inputs and outputs that can be leveraged by software for expandability with future updates. The hardware should be able to last through hours of gameplay without any problem.

Software: The code will run all UI components smoothly and responsively to enhance the user experience. It should also include options for running multiple game modes (free for all, teams, etc). The software should be designed in such a way that the code is able to be expanded in the future without the need for major refactoring. The team hopes to develop the software using common industry standards such as Agile and Object Oriented Programming concepts. The firmware should leverage these design patterns to make development easier and ensure that bugs are kept to a minimum.

User Interface: The gun should have an intuitive method of interacting with the menus to minimize setup time and increase user enjoyment. The user interface will include buttons, a small display, haptic feedback and smartphone integration to allow for easy and quick setup. The user interface should be easy to understand and get started with to allow for an immersive experience for the user. The User Interface should also have some customizability that allows for users to select the best experience that suits them. The gun should be play tested with multiple people who are not involved with the development of the project to get an outside perspective of the usability of the project.

Power supply: The gun will contain its own battery and charging circuit to allow sustained gameplay and standby time. The gun should also be able to measure its own battery capacity and display it to the user. The gun should be able to sustain its own power for a long duration in standby mode just as most modern electronic devices do, for the convenience of allowing multiple gaming sessions on a single charge of the battery. During gameplay as well the battery life should be optimized to allow for several rounds of gameplay without the need for a recharge in between.

Reliability: The gun will have multiple software based checks to ensure the functionality of the gun is in working condition. This includes unit tests for all of the sensors and peripherals, as well as sanity checks on incoming data. This will allow for easy debugging of the product during the testing phase of the project, and quicken development time, as well as increase the overall reliability of the product. Reliability is what will separate this project from many others, by delivering a complete, ready to ship product instead of a proof of concept. This includes making sure that the gun works completely on its own, without the need of any auxiliary hardware and works in a variety of environments.

2.2.1 Stretch Goals

- App Development:
  - Design of a mobile app would enhance the project greatly. This app would include GPS and the ability to connect to the internet. The GPS would allow for several updates to the game. All users in a certain region of the GPS would automatically be added to the game. Location services would also allow for “zones” in the game to be created. These zones could be used in a “capture the flag” game mode as discussed in Section 2.8, or an opportunity for players to reload by simply going to a certain area rather than using the accelerometer or the pick-up stations described below and in Section 2.8.
  - The app would also allow for an online networking of players, allowing users to share stats about how often they’ve played and records they have broken, such as the number of points scored in a single round. This allows

for a possible community of laser taggers that makes the gameplay and user experience more immersive.
  - The ESP32 chosen for this project already includes a BLE module, which would allow for very power efficient communication between both an iPhone or Android smartphone. Using BLE notifications, the gun and smartphone can actively communicate during a game round and display relevant game information on the smartphone's screen. This means that this implementation does not require any extra hardware, and can be done at any point after the release of the initial product.

- Reload/Pickup Stations:
  - Creating standalone stations to reload “ammo” in each “gun” would also be an interesting feature to add. These stations could be used in the gameplay upgrades detailed in Section 2.8. These stations could be made of almost any material, as long as it was able to support the hardware required to connect the station to the rest of the existing gameplay set up. The “reload” stations would have IR receivers that can be shot just as normal guns can, and use the ESP32 Mesh networking capabilities to send back the upgrade information to the appropriate gun. Since the IR blast is encoded with the information about the shooter's player ID, it is possible to accurately determine exactly who shot the target and deserves the relevant upgrade. The upgrade could be in the form of health, ammo, or upgrades to the actual weapon system, such as max rate of fire, firing modes such as semi-automatic or automatic, and higher damage values for each bullet.

- Individual Targets:
  - Creating standalone targets with individual IR receivers would allow for the “single player mode” detailed in Section 2.8. These targets could be made of almost any material, as long as it was able to support the hardware required to enable the IR receiver and connect the target to the player's blaster. The goal of the standalone targets would be to place them almost anywhere, so a hanging bracket or support of sometime should be included.

2.3 Project Milestones

In this section the project milestones for Senior Design I and Senior Design II are displayed and are explained. These milestones were first brainstormed about in the senior design bootcamp. The first table shows the week count and what milestone tasks are being worked on at that point in time for Senior Design I and the second table shows the same information but for Senior Design II.

Table 1 below displays critical milestones for this project to be done in Senior Design I. The initial stages of the project (Weeks 1-10) included mostly concept development and research, and have all been successfully completed. The 60 and 100 page submissions of the design documents show the results of the work from those weeks. The senior design bootcamp encompassed a planning and brainstorming session for the group and allowed for the group to create shared values and goals. A broad outline of the milestones was created that have been modified over time to create the table that is displayed below. Project specifications were made as a group and were discussed at weekly meetings in order to create the best specifications while keeping the project achievable. The research for the subsystems was done individually and was talked about to one another at the weekly meetings to make sure all group members are learning about the entire project. Beginning in Week 8, the project became less abstract as actual part selection, design, and testing have begun. The designs done in weeks 8-16 are an important stepping stone in determining design limitations and places where the project can expand. With an initial PCB design and finalized design document, the project is in excellent shape to be refined and perfected beginning in Senior Design II.

Week	Milestone Task
Week 1	Form groups
Week 2	Attend SD Bootcamp and brainstorm ideas
Week 3	Finalize project idea
Week 4	Decide on project specifications
	Finish Initial Design Document
Weeks 5-7	Power research
	Infrared and serial communication research
	PCB research and initial parts list
	Finish 60 page draft
Weeks 8-10	Finalize parts list/order parts and PCBs
	Initial component designing and initial PCB design
	Finish 100 page draft
Weeks 11-13	Initial 3D models and test individual component designs
Weeks 14-16	System testing
	Finish 120 page draft

Table 1: Senior Design 1 Milestones

Table 2 below shows the major milestones for Senior Design II. The PCB design will go through any revisions that are necessary. The revisions needed will come from any issues that arise in the PCB prototype testing done in the late weeks of Senior Design I. The design for the housing will be finalized. This goes in hand with the PCB revision as the housing is designed around the PCB and assuring that all the components are secured in place. The game-play modes will be finalized and implemented in software and will be programmed to the microcontroller. In the following weeks, the final product construction will take place. This involves putting all the components within the housing and ensuring the wiring and connections for components are in good shape. During weeks 9-14 final testing will be done. Testing will be done with the entire system in one piece and will assure all subsystems are interfacing one another correctly. The entire system will be checked to make sure all of the goals and requirements are met. This process is being given a substantial amount of time as problems are expected to arise as different situations and environments are tested in to make sure the entire system works as a whole in any situation or environment it is put in. The final presentation will take place at the University of Central Florida and is expected to be in front of the Harris Corporation Engineering Center.

Week	Milestone Task
Weeks 1-3	Revise PCB
	Finalize housing
Weeks 3-5	Finalize and implement game-play modes (software)
Weeks 6-8	Final product construction
Weeks 9-14	Final testing
Week 15	Final Presentation

Table 2: Senior Design 2 Milestones

2.3 Project Budget/Bill of Materials

Because a main goal of the project is to make the final product accessible in all ways, it is important to keep the cost as low as possible without sacrificing performance. The budget shown below in Table 3 includes all significant components that have been ordered to be used in the final product. This table does not include items of insignificant value such as resistors or capacitors. The final price is given in both total cost (cost for four blasters) and unit price (cost for a single blaster). This budget also serves as a parts list/bill of materials.

The final total cost is well within the goal range. At the onset of the project, the group had expected to invest around $200 total, no more than $50 per blaster.

With the initial estimate being $135.38, spare parts or replacement parts can be ordered as needed while still remaining in the proposed budget. The unit cost of $33.84 is well below the maximum the group had in mind. With the initial cost estimates being so much lower than the initial budget, there is a lot of room financially to explore adding more features to the game in the future.

Item	**Unit Cost ($)**	**Quantity**	**Total Price ($)**
MCU (ESP32)	4.39	4	17.56
3D printer filament (ABS)	21.99	2	43.98
PCB	5.59	4	22.36
Lithium Ion Battery	6.50	4	12.00
Charge controller	0.99	4	3.96
Charge Protection (DW01A & FS8205A)	$0.60	4	2.40
OLED	3.48	4	13.92
Voltage Regulator	0.23	4	0.92
Accelerometer	0.99	4	3.96
RGB LEDs (WS2812)	0.99	4 sets	3.96
Tactile Button	0.10	4	0.40
Vibration Motors	1.00	4	4.00
IR LED & Receiver	1.35	4	5.40
Total Unit Cost:			$33.84
Total Cost (4 Units)			$135.38

Table 3: Budget/Bill of Materials

It is listed that this project will require four of most items to reflect that four identical guns will be made but it should be noted that most manufacturers do not sell these parts individually. As a result, a fluctuation in total cost will be seen if that factor is taken into account. For instance, though only four vibration motors are needed, they are sold in packs of ten so $10.00 was spent rather than $4.00.

When deriving the cost when buying in bulk, the total cost comes out to be $260.09. At this costs more than four guns can be made, however, since subsequently more devices have been purchased.

2.4 Game Mechanics

This section details how the game will be played, including both main gameplay goals and possible upgrades. Having varied game mechanics helps to keep the game more interesting.

2.4.1 Main Gameplay Goals

The basic mode of gameplay for this laser tag system is to enable four players to play individually. Players “shoot” at each other, aiming their infrared beams at the infrared targets attached to each gun. Each confirmed “hit” registers a certain amount of damage. When one person’s damage reaches a certain, predetermined amount, that person is considered out of the game. Data regarding the damage caused by each hit is transmitted between microprocessors via the infrared beam. The beams serve as the medium for serial communication, with each beam containing a specific value. The infrared receiver or target absorbs the beam and is able to discern the information contained in the beam. This data transfer is essentially the same as the communication between a TV remote and the receiver on the TV. The data is passed to the microprocessor and totalled. When the accumulated damage reaches the predetermined value, the microprocessor sends an alert to the OLED display included in the gun to tell the user. When three players have reached the damage limit, the remaining player has “won” the game. Different sounds will be implemented in the software and played through the speaker to alert a player of when they fire a shot, when they hit a target, and when they have been hit by another player.

2.4.2 Possible Gameplay Upgrades

When the main gameplay mode has been finalized, there are several options for varied modes of play. A list of modes would appear at start-up of the software, and the user would select the mode they want to play. The following list details these possible upgrades.

- Haptic Feedback: An option in addition to LEDs and the OLED display for feedback when a shot is fired or a target is hit is haptic feedback. This is a vibration that occurs when a certain action is performed.

- Team play: The four players are divided into two teams. The teams could consist of one, two, or three people. Teams will be designated by a certain color of the LED ring attached to the gun. The mechanics of the original play mode will remain, but a team will "win" if one of its players is the player that does not reach the damage total. Teams could be chosen by the players or randomly selected when the team play game mode is initiated. An important thing to note in this mode if someone from one team hits a player on their team, no damage will be registered.

- Reload stations: Each gun only has a certain amount of shots it can fire. Once that limit is reached, it will need to be "reloaded" by approaching an external infrared signal and waiting a predetermined amount of time until the counter of possible shots is reset. A separate sound effect would be added to alert the player when the reloading has started and when it has finished. Another possibility for a recharging "action" would be pointing the "gun" in a certain direction. An accelerometer would be used to detect the orientation of the "gun" and send the appropriate signal to the microprocessor.

- Single player mode: Only one player is needed. Instead of shooting at other players, multiple external infrared "targets" could be set up by the player. Each hit of a target would register a certain amount of points. Modes within this mode would include a "timed" mode where the player tries to gain as many points as possible in a predetermined amount of time, or a "practice" mode where the player registers points with target hits but without a time limit.

- External targets integrated into multiplayer modes: The external targets described in the single player mode would be integrated into either multiplayer mode. Registered hits of the external targets could subtract damage from the player that registered the hit, add damage to other players, or serve as an opportunity for a player to reload from a distance.

- Territory Control: This mode would utilize external targets and the "team play" mode. Each target would represent a territory or "base" that a team tries to capture. The team would capture the base by shooting its designated target enough times. The IR receiver in the base target would keep track of how many hits were registered and which blaster registered the hits. The base could also be recaptured by the opposing team in the same manner. This mode would be timed, and the team with control over the most bases would win. This idea is inspired by several existing video games, with the most inspiration coming from Star Wars: Battlefront.

- Capture the flag (in team play): Players try to maintain possession of a certain object containing an infrared target/receiver. Players on the opposite team can then target this object. If the object registers enough hits, the team that had possession of it must surrender, then try to recapture it in the same way. Game play would end after a certain time or when the object reaches a certain “safe” space in the designated playing field. There could be one “safe” space for both teams, or both teams could have their own “safe” space if the playing field could be easily divided between the teams. This could be implemented if a GPS tracking system was included in the object the teams were trying to capture.

- Immortality: Instead of losing “health” when a player registers a hit, their gun is disabled for a certain amount of time. Each player would receive a certain amount of points for each hit scored. This mode would be timed, and the player with the most points at the end of the time would win.

- Player “classes”: In this mode, each player can select a “class” of player. Each class has inherent strengths and weaknesses. Two simple examples would be one class with a high rate of fire but each shot does less damage, while the opposing class would have a lower rate of power but more damage with each shot. The firing rate information would be controlled through the microprocessor while shot strength would be packaged in the IR transmission.

- “Mercenary”: This mode utilizes the concept of player “classes.” In this mode, one player is randomly selected to be the “mercenary.” Their shots have more power than any other players’. The mercenary is initially not assigned to a team. The team that shoots the mercenary first acquires him or her for their team. However, if the opposing team shoots the mercenary, they steal him or her away. This mode is similar to capture the flag or territory control, but with an actual player rather than an external target.

- Zombie Attack: This mode starts much like the standard team play mode. One team is designated as “human” and the other as “zombies.” If a player from the zombie team shoots a player from the human team, the player from the human team becomes a player for the zombie team.

- Laser Musket: This mode could be played in the standard free-for-all mode or in team play. The difference is that the mode only gives each player a set amount of shots with a long reload time between each shot and no opportunity for restoration of ammunition.

2.5 System Flowcharts

In this section, the overall software flowchart and hardware block diagram are described. Each goes into a broad summary on how the system works as a whole with all of the subsystems working together.

2.5.1 Software Flowchart

In the project, there will be software running on the microcontroller that will control all user interface interactions and all of the communications within the system that are needed to make the system work smoothly. Figure 1 shows how the software will flow. At the beginning of each game the screen that is on the system will display the health of the player. Then the system awaits the press of a button or the receiving of an infrared beam from another one of the guns. Once one of these inputs occurs, the corresponding sequence of events for that interaction occurs. For the infrared beam, this would include decoding the message, taking damage, and updating the screen. For pressing the trigger, this includes transmitting an infrared beam with an encoded message and briefly turning the haptic motor on. Nearing the end of a game, if a player with low health gets hit with a beam and takes enough damage the system will show that the player is out through the display and will end the game for that player.

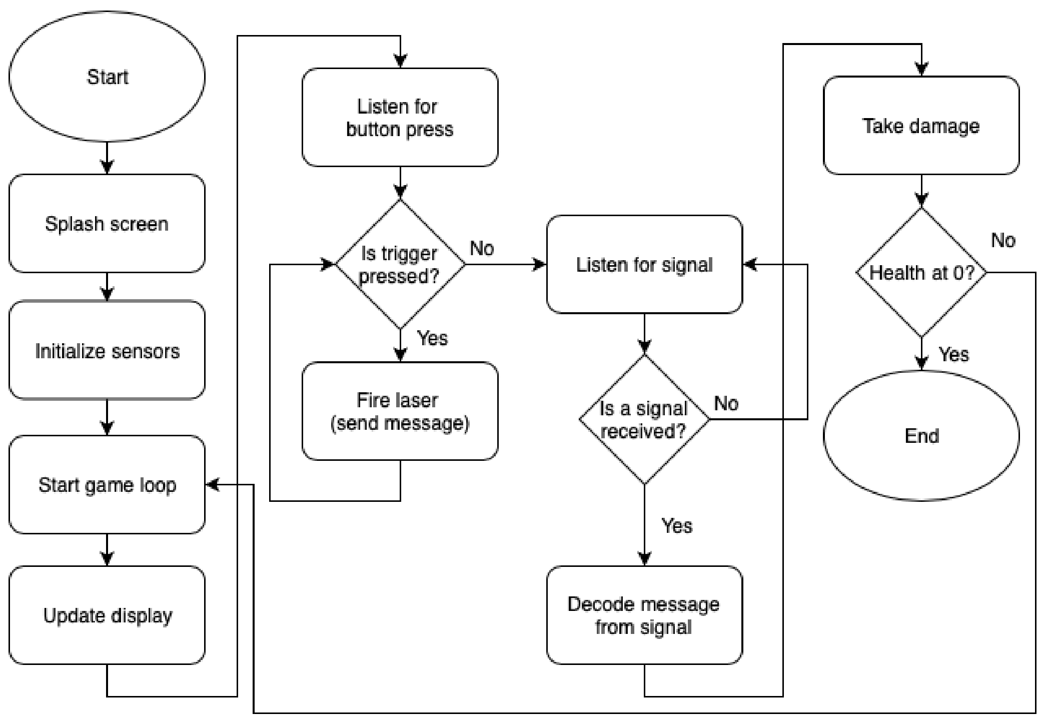

Figure 1: Software Logic Flowchart

2.5.2 Hardware Block Diagram

The system will consist of a controller, a display, an accelerometer, a trigger, RGB leds, an infrared transmitter, an infrared receiver, a buzzer, a haptic feedback motor, a WiFi mesh network, WiFi setup, and a battery. All of these components will interact with each other to make the system work as shown in Figure 2 below. These components will be discussed extensively throughout the course of the document which will include their purpose, research, as well as parts comparison and final part decision.

The input and output of each block are as follows. For the display, there will be the input from the microcontroller telling the display what to show on the screen. For the accelerometer, there will be output to the microcontroller telling the microcontroller what acceleration it is sensing. For the trigger, the output will be to the microcontroller which will tell the microcontroller when it is pressed and when it is not pressed. For the RGB LEDs, there will be an input from the microcontroller in order to program the color of each of the LEDs which will in turn chain from one LED to another in order to program all of them. For the haptic feedback motor, there will be an input from the microcontroller controlling when the motor is on and when it is off. For the buzzer, there will be an input from the microcontroller to designate what frequency the buzzer is set at and if it is on or off. The frequency will determine the tone. For the battery, the power will be output to the microcontroller. For the infrared transmitter, it will receive an input from the microcontroller that it will output towards another infrared receiver on a seperate gun. For the infrared receiver, it will receive an input from another infrared transmitter on a seperate gun and will output the information to the microcontroller. The WiFi Mesh and WiFi setup will be on the microcontroller. Both will input and output from the microcontroller to set up the game mode or change settings for the software. This is all is demonstrated in Figure 2.

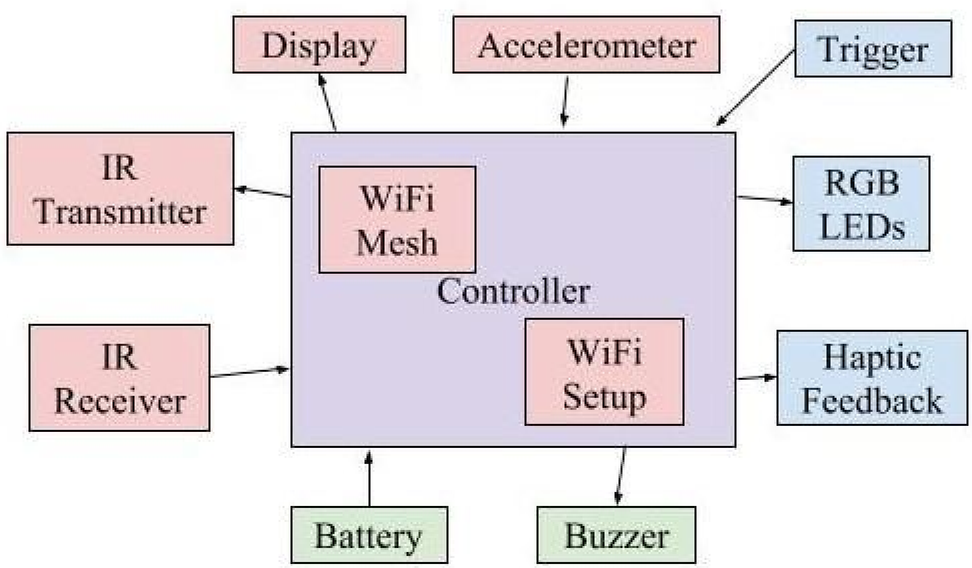

Figure 2: Hardware Block Diagram

In Figure 3, it is shown how the hardware block diagram is split up for the group. The background color of each box corresponds to who is in charge of that aspect of the project. By breaking up the project into sections, each team member will have a chance to use their particular set of skills to further improve upon the project. It also breaks up the workload to a manageable level for each person.

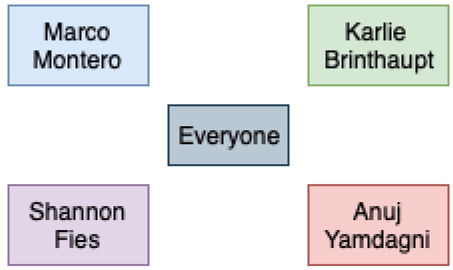

Figure 3: Group Breakdown

Tasks were assigned to each group member based on their strengths, coursework, personal interests and career interests. The most obvious task assignments were Anuj's. As the team's only computer engineering major, he has the most coding experience. The electrical engineering majors have a foundation in coding and can help with simple tasks like IR transmission, LED displays and buzzer sounds, Anuj is the best suited to handle the complex architecture that this game will require. While all group members are accountable to keep the project on track, Anuj was also designated as the "project manager" in situations that would require that specific role, since he was the most experience with similar projects.

Karlie was assigned design of the power supply because of her career interests and coursework. She is currently enrolled in Electronics II at UCF, which covers all topics relevant to designing a power supply, including op-amps and voltage regulators. She also has a strong chemistry background, which gives her the background knowledge required to make the best choice for a battery. Experience in power supply design is also relevant to jobs and internships she is currently applying for. She has been able to use the design process as a talking point in interviews. Karlie also used her software background and music background to write the code for the buzzer sounds.

Marco was assigned to do the 3-D modeling of the project because he is the only team member with previous experience with modeling software. Marco is able to efficiently create models and teach the other team members how to use the software. He also took charge of writing code for several auxiliary components like the IR receiver and transmitter and displays. Marco did the project research for serial communication, as he has experience dealing with it from Embedded Systems at UCF and had the most interest in the process. Marco is currently enrolled in Semiconductor Devices, so any research or designs involving transistors or LEDs were left to him.

Shannon was assigned PCB design because of her experience as an intern with Harris Corporation. While she was not directly involved with PCB design at

Harris, she saw many designs day in and day out and was able to get a general understanding of good PCB design and good design practices. Shannon is enrolled in Electronics II as well, and therefore gained a lot of practical knowledge from Dr. Weeks, who included as much real-life design advice as theoretical concepts in his lectures. She also did much of the work to establish the initial device-to-device communication network using the mobile hotspot feature on her iPhone, as she was interested in learning how to program a feature like that in the Arduino development environment.

While each group member was assigned certain tasks, another goal of the group is to make sure each member has a working knowledge of each aspect of the project. To achieve this goal, at each group meeting, each group member gives a short summary of their current tasks as well as their progress. This ensures that everyone has a well rounded design experience that will give them useful tools for any future career.

Each of the blocks status as of 11/28/2018:

- All blocks are currently in the design process
- None of the blocks have been completed
- All the blocks have been acquired for testing (trigger, display, receiver, transmitter, battery, haptic feedback, microcontroller, wifi module)

2.6 House of Quality

The House of Quality is a figure that defines the engineering requirements, marketing requirements, and their correlation. This figure allows us to easily compare and contrast the different requirements needed for our project and to describe their relation to each other in terms of four varying degrees of correlation: strong positive correlation, positive correlation, negative correlation, and strong negative correlation. The degree of the correlation between two requirements are completely subjective, however, help give the reader an understanding of how strong the relation is between the two. In addition to showing correlation, the House of Quality indicates whether more of a requirement is wanted or less. For example, for a system where it should be designed as small as possible, the engineering requirement would be ‘size’ and the polarity associated with it would be ‘-’ since it should be as low as possible.

Marketing requirements are requirements that are of priority to meet such that the product will be well received when put on the market. One common marketing requirement is cost of the product. In today’s economy, the lower the cost of a product the easier it will sell. For that reason, the requirement ‘cost’ has a negative polarity. Engineering requirements are requirements that are of priority for the engineering process. While cost is another common example, dimensions can be seen as another common requirement. In the case of our project, we would like the laser gun to be reasonably small such that it is portable. For that reason we gave dimensions a negative polarity.

Also included in the House of Quality are targets for engineering requirements. This section is related to the engineering requirements in which it defines related values to them. For instance, one engineering requirement is cost and our target for that requirement is that the project should cost less than $50 per device. All these targets should be attainable and more importantly demonstrable or be able to be documented. Below shown in Figure 4 is the House of Quality created for this project including all correlations, requirements, and targets.

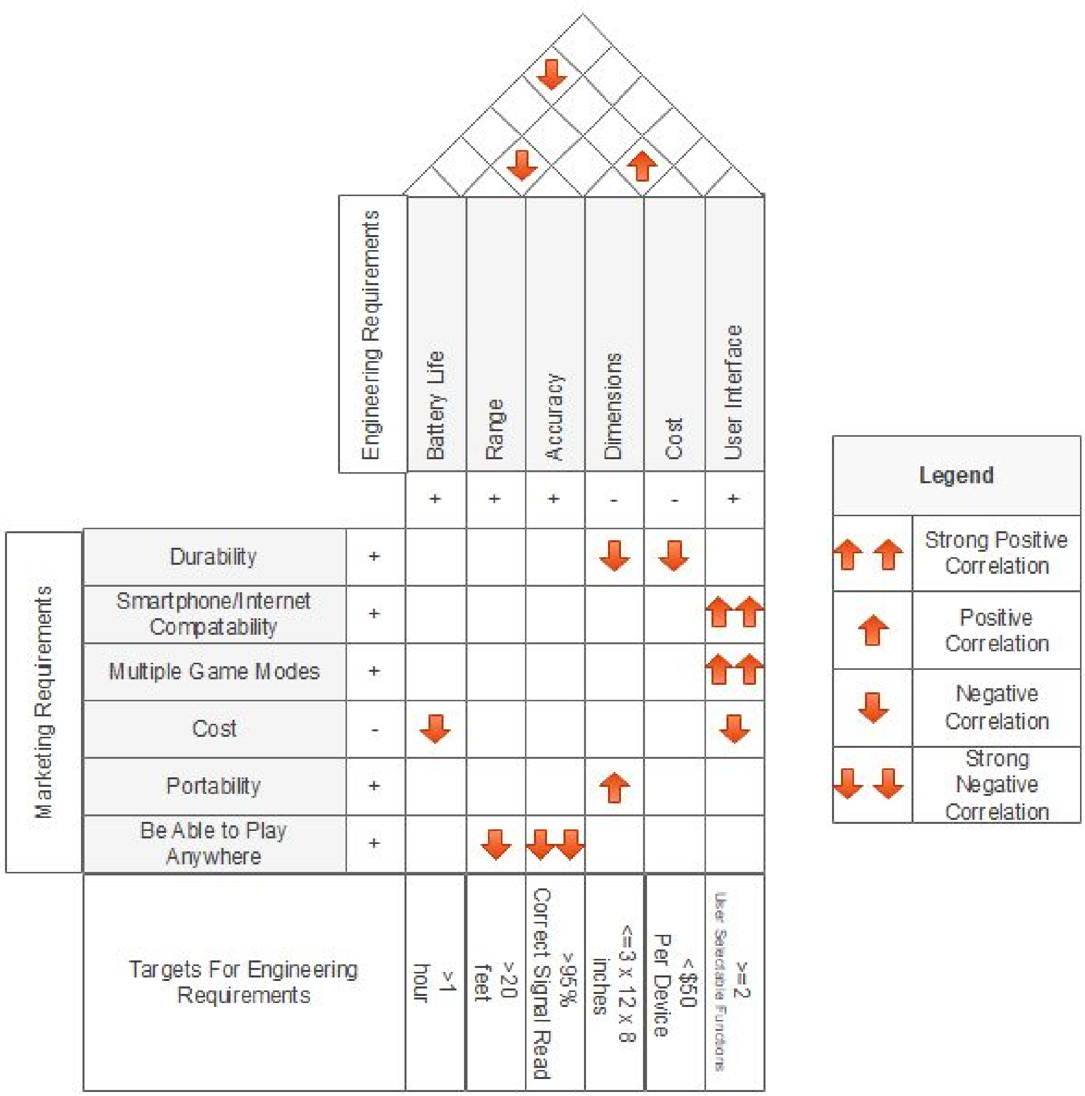

Figure 4: House of Quality

2.7 House of Quality Breakdown

The House of Quality is created unique for each individual product. This is because each project has its own particular needs and qualities which are reflected by the different marketing and engineering requirements seen. In general, the laser tag laser gun needs to be small and lightweight so that it is convenient to use and should be as accurate as possible when it comes to executing code and receiving data. In this section, we will describe each of the requirements as well as the engineering targets and explain why we chose to use them in the house of quality.

2.7.1 Marketing Requirements

As explained earlier, marketing requirements are defined such to bring awareness to what the consumers desire in a project of this type. For this project's House of Quality, we narrowed down these requirements to six core qualities that we believe are important when selling the product to the consumer: durability, smartphone/internet compatibility, multiple game modes, cost, portability, and the ability to play anywhere.

Durability: As defined for this project, durability is the characteristic that is directly related to the strength of the housing of the laser gun such that it will not warp, fall apart, or otherwise be compromised through the stresses of normal game play. Some things that may test the durability of the project are dropping onto the cement, accidentally clipping walls, liquid (such as rain or sweat) entering the housing that could destroy the electronics within, and so on and so forth. These events can occur during normal game play outdoors or indoors and is imperative to overcome such that the reusability of the laser gun is not in jeopardy. This would affect the market value of the project since the better made, stronger project is always better received in the market as compared to a dainty device that will break within one use if the prices are the same. As a result, we associated the positive polarity with durability since we would like the device as strong as possible.

Smartphone/Internet Compatibility: In a day in age where smartphones and likewise the internet virtually connect our world, it would be a great bonus to have our project also to be connected. Adding internet and smartphone compatibility to the laser gun will allow us to creatively explore more options within the playing realm to increase the likelihood that the consumer will enjoy the product as well as the ease of use. If the laser gun can be connected to the user's smartphone or computer, a wide array of customizability would be available for the user in a easy to use way. For example, live scores could be uploaded for everyone playing to see or the user can organize teams that will be relayed through the smartphone to the laser gun's ESP32 module. These features brought forth from internet compatibility would enhance the customer's experience using our project hence we attributed this characteristic with a positive polarity.

Multiple Game Modes: One important aspect of the commercial gaming products is that they should be able to keep the consumer entertained for an extended period of time. One way of doing this is offering multiple game modes such that the user can cycle through the different modes instead of only having the one option. This will allow the device to be used multiple times without quickly becoming stale or boring. Most of these game modes will be centered around them being multiplayer, such as one versus one or playing on a team against each other, but a stretch goal of ours is to provide a one player mode such as target practice. Since there is a positive correlation between increased game mode and overall market reception, we attributed this requirement with a positive polarity.

Cost: It is no secret that the cheaper the product is, the easier it will sell. There does exist a balance however between cost and the other requirements needs. For instance, if we drive the cost too low, we will be forced to use cheaper products and hence degrade the durability of the device. This will go against the standard we set earlier for durability and should be avoided. As a result, we will look at the product first then analyse the cost of it. If the product meets our standards and comes at an affordable price, we will choose it over the more expensive product. For the sake of the market requirement, we overall wish the cost to be as low as possible so we attributed this requirement with a negative polarity.

Portability: For a project of this type, it is imperative that it is portable over anything. Qualities that are tied to the portable characteristic are mostly governed by size and weight. If we make a laser gun too big or too heavy then it would be hard to sell as a product that can be played with by kids of all ages. In addition, the quality of the game may be inhibited by the cumbersome laser gun. For that reason, we gave this marketing requirement a positive correlation since the more portable it is, the better it will be received.

Be Able To Play Anywhere: The final marketing requirement defined for this project is the ability to play anywhere. This requirement highlights the goal that the game of laser tag should be able to be played inside or outside in both optimal or non-optimal lighting. This is to enhance the user's experience using the device and is seen as a positive correlation.

2.7.2 Engineering Requirements and Targets for Engineering

Recall that engineering requirements are requirements set for the engineering process and define specific criteria that the project should meet from the engineering standpoint. These requirements should be demonstrable as defined by the targets for engineering. For this project, we chose six requirements that are paramount for our project's success: battery life, range, accuracy, dimensions, cost, and user interface.

Battery Life: The first engineering requirement calls for the battery life of the device. We define battery life as the length of time between necessary charges. Ideally, we would want the battery life to be as long as possible such that the user's experience would not be compromised by causing limited play time due to the device being to be continuously charged. Besides using a battery with a longer lifetime, an idea that is one stretch goal to implement is having a low power mode setting that will be activated outside of game play. That way, the battery charge will be used in the most effective way possible. As a demonstrable target, we set that the length of time between charges to be greater than one hour such that the user can go through multiple games before the charge is needed. We attributed a positive polarity with this requirement since we would like the longest battery life as possible within the bounds of the other requirements such as dimensions and cost.

Range: The range requirement defines how far away that the IR led can be read by the receiving device. This is to enhance the the user's experience and to make the game of laser tag more enjoyable. When the user plays laser tag, it is in poor taste if the person has to be a few feet away from the other player in order to register a shot. With a longer range, the user can hide from the target and still be able to deliver a shot, which in our opinion would strengthen the immersion into the game. As a baseline target, we set our minimum standard range to be at least 20 feet which we believe would be sufficient enough to put enough distance between the user and the target. This target should be achieved in all lighting conditions as defined by the "be able to play anywhere" market requirement defined earlier. Since the greater the range, the better overall the project will be, we placed a positive polarity with this requirement.

Accuracy: The accuracy requirement is closely related to both the range and the ability to play anywhere requirements. Our definition of accuracy is the rate at which the IR led message is sent and received assuming that the sender and receiver is lined up and nothing otherwise would hinder the message signal on its way to the receiver. This factor should not be heavily affected by the range and the lighting conditions that the game is played in. We set the target accuracy to be greater than 95% such that we are accountable if an excessive amount of messages are lost due to the design of the device. This quality is easily demonstrable by simply shooting at another laser gun more than 10 times and recording how many shots were successfully read. Since we would like to accuracy to be as high as possible, we tied a positive polarity with this engineering requirement.

Dimensions: The dimensions is a requirement for the same reasons as the portability marketing standard. The idea is that the larger the gun, the harder it is to play with so it would be less desirable on the market. For engineering, though with larger dimensions we would be able to house a larger battery to improve battery life and could otherwise upgrade our components, we wish the size of it to be small so the device could be used for its intended purpose. As a target, we set the dimensions to be 3x12x8 inches which is the size of a standard sized Nerf gun. This is a requirement that would also be easily demonstrable for the senior design showcase. Since the smaller the dimensions will compliment the portability standard, we associated a negative polarity with this standard as well.

Cost: The engineering requirement for cost is there for the same reasons as the cost requirement for marketing. We chose to put it twice since cost will be a limiting factor for the engineering process and should be managed carefully. The cost can be limited by choosing low cost but quality electrical components, using 3D printer filament for the housing, and keeping the cost for the PCB and assembly as low as possible. As a demonstrable target, we would like to cost to be less that $50.00 per laser gun created which corresponds with our goal of creating a low cost laser tag gun. Since the lower cost is desirable, we put a negative polarity with this standard.

User Interface: The user interface requirement ties closely to the multiple game mode requirement and has similar objectives. As an engineering requirement, the goal is to implement at least three user selectable functions (such as game modes) such that the device will not become stale overtime. Since the device will be more attractive to the consumer if it has an easy to use, feature filled user interface, the positive polarity was attributed to this requirement.

2.7.3 Engineering Requirement Relations

Within and above the house of quality, relations have been made between the engineering and marketing requirements that are described as a positive correlation or a negative correlation. These correlations describe whether, for example, if a positive increase to an engineering requirement enhances a marketing requirement or decreases it. For instance, if the dimensions are decreased, this would enhance the market requirement of portability. On the other hand, if the battery life were to increase by purchasing an alternative battery with better efficiency, this would be a negative factor to bringing down the cost since the better battery will come at a greater cost. If the relation between the two are negligible then the block was left blank.

Cost, for the most part, had either a neutral or negative correlation with most of the requirements. The reasoning behind this is because the better the product, the more expensive it’s going to be, hence going against the notion that the costs should be as small as possible. The exception was made with dimensions since the smaller the dimensions used, the less amount of 3D filament will be used and money will be saved. Smaller dimensions also encourage the creation of a smaller PCB which again costs less.

Accuracy was chosen to have a strongly negative correlation with the “Be Able to Play Anywhere” requirement as noted by the two downward facing arrows. The rate of accuracy will peak in the ideal conditions of being indoors in the dark with no opposing light rays hitting it. However, the “Be Able to Play Anywhere” requirement expects that the user can also play outside in the day time or around other leds where the additional light rays could distort the incoming signal and degrade the accuracy percentage. This requirement will put our design to the test to see how well our IR led receiver can perform under these non-ideal conditions. For similar reasons, range has a negative correlation to the market requirement as well as accuracy.

The addition of the user interface as an engineering requirement, though negatively correlated with cost, was strongly correlated with the smartphone/internet compatibility and multiple game modes. All of these requirements are made to enhance the users experience and to make it more appealing in the tech savvy world we live in today. The two marketing requirements can interact, or in other words be displayed/enhanced, with the user interface and overall make the device look more appealing, user friendly, and fun.

2.8 Team Collaboration Tools

For collaboration the team will need to standardize and select tools to make teamwork more efficient. When selecting possible applications, many different factors were taken into account, including support for different operating systems and mobile platforms, and presence in industry.

Google Drive: The sharing of files of various types is made easier by using Google Drive. This is especially important for the sharing of design documents, schematics, as well as team management documents. The team’s Agile workflow schedule is also hosted on google drive, organized into a color coded spreadsheet. The online file hosting service provides a web based interface for editing documents and spreadsheets in real time, which makes the team’s workflow more efficient. Google Drive also has apps for both iOS and Android, which allows teammates to work remotely.

Github: Although Google Drive can handle any filetype, it is more effective to use a dedicated code repository for the management of software development. Using a code repository has several benefits that make it an indispensable tool for the development of the gun’s firmware. First of all, the Git protocol allows for multiple people to collaborate on the same file of code at the same time. This is done through slicing the file into its individual lines of code that are edited, thus one member can edit one section of the code without interfering with other sections of the program. The other method is through branches. Branches simply copy the state of the master branch code for further development, independent from the work of other team members. When the addition of a certain feature is complete, the branch can be merged back into the master branch. If two members edit the same section of code, a “merge conflict” occurs. Github provides an intuitive interface to resolve these merge conflicts to continue development. Another benefit of Github is its automatic versioning backup recovery options. Instead of just storing the latest copy of the source code, Github stores the changes between each commit of the source code, which means that the code can be reverted to any commit made to the repository. This is handy in the event that a bug is introduced to the repository and the code needs to be reverted to a previous version. Many bugs are expected in the codebase due to the quality of education provided by the engineering department at UCF, however the use of Github should provide an effective solution to this challenge.

Facebook Messenger: For general communication at all times during the day, a platform was chosen that supports instant messaging of text, memes and images. For this, Facebook messenger was chosen, for several reasons. Facebook Messenger is supported on iOS and Android, in-browser messaging, and a desktop app for Windows. This allows for the scheduling of meetings, sharing of dog pictures and general collaboration discussions. Facebook Messenger was also chosen due to its easy adoption, as all team members already had accounts and were familiar with Facebook.

3.0 Project Research

This section will go through research that has been done on the following topics:

- Similar Project Research
- Power Supply Research
- Communications Research
- Thermal Considerations
- Serial Communication Research
- Software Development Model Research

3.1 Similar Project Research

The following section goes into the research of previous similar projects that encompass ideas and components that are similar to the ones this project is intending on using.

3.1.1 Arduino Laser Tag

A simple version of a laser tag game can be created using an Arduino board, a lightweight gun housing, an IR transmitter system, an IR receiver system, a sound system (Peizo sounder), a transistor, a trigger, basic circuit components, and some visual effects (colored LEDs, etc). The IR transmitter must be matched to the receiver.

Resistors and capacitors are used to create a filter that allows only the desired IR signal to pass through to the Arduino. The filter is a simple low pass filter, with the resistor and capacitor values chosen to result in the desired cutoff frequency (fo) where $fo = \frac{1}{2\pi RC}$. An example of a simple RC lowpass filter circuit is shown below in Figure 5.

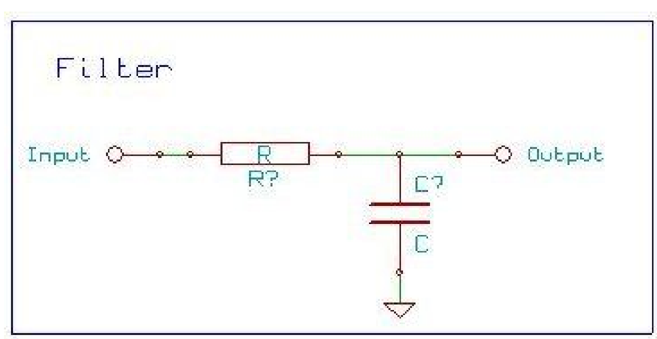

Figure 5: RC Filter Circuit (*Permission Requested)*

For the IR transmitter, an IR LED can be attached to the output pin on the Arduino. To increase the range that the LED can transmit over, an amplifier is needed, which is where the transistor comes into play. An example of a transmitter with a BJT is shown below in Figure 6.

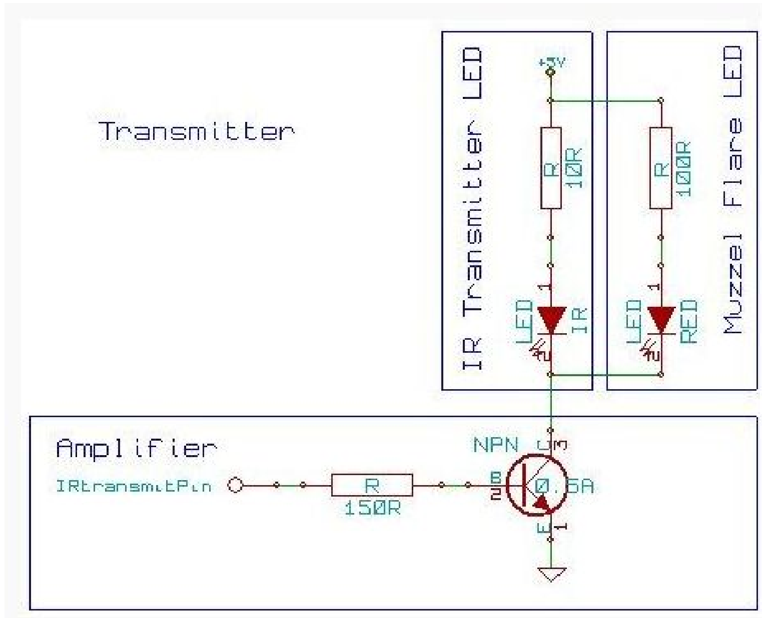

Figure 6: Amplifier Circuit (*Permission Requested)*

A lens can be used to focus the IR beam. For the receiver, a matching LED could be used (where the receiver operates on the same light wavelength as the LED). In the receiver, a signal received results in a digital zero voltage level. The receiver can be attached directly to the Arduino, but would work better when attached to capacitors and pull up resistors to regulate the battery. An example receiver circuit is shown below in Figure 7.

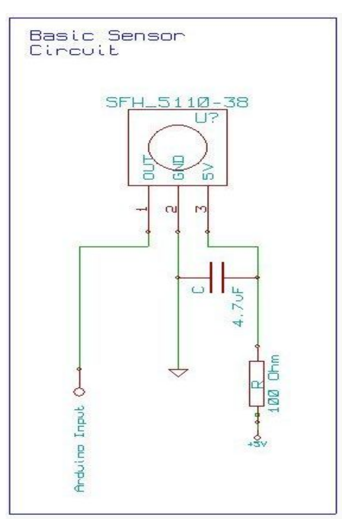

Figure 7: Sensor Circuit (*Permission Requested)*

For the sound effects, a simple Peizo speaker can be used. The speaker can be programmed to output standard electronic sounds. Depending on working versus desired volume, an audio amplifier can be added. For the visual effects, LEDs can be used for simple and immediate feedback and an LCD for more detailed displays. The LEDs can be powered through PWM pins on the Arduino. These pins can vary the brightness of the LED, indicating changing levels of "health" or shots remaining. The rest of the game play mechanics are determined by the code, which would include provisions for the different versions of game play described in Section 2 of the report [1].

3.1.2 University of Florida "Laser Tag Gaming System"

A senior design group at the University of Florida had a project similar to this project. The laser tag "guns" has an onboard system to keep track of scores. The targets are actually solar panels attached to vests that players wear. Each player's laser beam has a unique frequency, so other players are able to record who hit them. It also allows a filter to be set up on the receiving solar panel vest so that only transmissions of that particular frequency are received. Connected to the filter is also an amplifier which amplifies the voltage of the signal to a magnitude where it can be recognized by the microcontroller. An LED display on the "gun" displays "health" (how many times a player has been hit) and how many shots a player has remaining. There is also an external scoreboard, which is made of 14 7-segment LEDs that keeps track of the stats of all players and is visible to both the players and potential spectators. Hits and shots fired are transmitted through interrupts in the microcontrollers chosen for the "guns."

Instead of infrared technology, the group chose to use a laser diode, which produces a visible beam where IR would not. The signals are received by solar panels, as stated above. The devices communicate to the game control station using XBEE. The XBEE are connected to the microcontroller using EUSART. Each "gun" has an LCD display that displays the player's status and alerts the player when they have been hit. Audio circuitry and LEDs will also indicate a target hit/a hit of the players solar panels. The housing is a widely available Nerf gun, modified to hold the electronic components [2].

3.1.3 Pulse Modulated Laser Tag Game

This particular project has a relatively simple basis. A squirt gun is taken apart and modified so it can hold the battery pack from a standard flashlight. This battery pack is the power source and contains three AAA batteries. From pictures shown about the project, the batteries are alkaline. The "laser" consists of three consecutive pulses that fire when the trigger switch is pressed. The pulses are 50ms long and modulated with a 60% duty cycle (where the duty cycle is the "fraction of one period in which a signal or system is activated [3]"). The laser source is a simple, relatively available laser pointer. This particular game was created so there is one “gun” that a single player uses to shoot at stationary targets. The targets included a sensor to detect pulses and a display to indicate a hit. The build of the target package could easily be adapted to another “gun” to turn the game into a multiplayer game [4].

3.1.4 Desired Elements from Similar Projects

There were elements of each similar project that can be utilized in this project and some that do not fit with the goals. The Arduino Laser Tag project uses IR transmitters and receivers to transfer data. The University of Florida design team and the pulse modulated project designer chose to use laser diodes. Because IR is relatively easy to implement, it has been chosen in this project instead of laser diodes. An inherent challenge of IR is the dispersal of the signal and difficulty in focusing the beam. These challenges can be overcome in ways suggested by the Arduino project. The UF design team has similar solutions, but for the laser diode receiver. These suggestions include a filter to remove signals outside the frequency of the IR beam and a lens to focus the light. An amplifier can also be used to ensure the signal resulting from the IR beam is strong to trigger the desired response in the microcontroller. This project will implement these fixes as needed, using the circuits given in section 3.1.1 as a guide.

The University of Florida design team used solar panels as targets that were attached to a vest worn by each player and the pulse modulated laser project uses stationary target sensors, while the Arduino project uses an IR receiver attached to the main housing of the “gun.” Because this project is meant to be a multiplayer game, using only the stationary targets of the pulse modulated laser project is not an option. The idea of the stationary target from the pulse modulated laser project may possibly used for a gameplay upgrade that includes a single player mode where hitting stationary targets is the main objective. While the UF design team’s solution for targets is effective, a main design goal of this project is portability, so a simpler solution for IR receiving targets like what the Arduino features is more desirable. Solar panels like what the UF team used would also increase cost, and keeping the overall cost low is another main design goal of this project.

Both projects researched have feedback systems similar to what is desired for this project. The Arduino project has sound feedback using a Piezo speaker, LEDs, and an LCD display. The Piezo speaker emits standard electronic sounds, the level of brightness of the LEDs can indicate certain status levels, and the LCD screen can give more detailed information about the player’s status level. The UF design team has an LED display on the “gun” housing and an external scoreboard made of several 7-segment LEDs. This project will certainly utilize the Piezo speaker, LEDs to indicate status levels and possibly team associations, and instead of a LCD display giving details to the player, this project will use an OLED display. An external scoreboard will not be used to limit complexity and keep the game environment self contained. All necessary data will be displayed on each player's OLED display. In addition to the light and sound feedback, this project will include haptic feedback in the handle of the "gun" to indicate to the player that a shot has been fired or a hit has been received.

The pulse modulated laser project uses a battery pack made of three alkaline AAA batteries to power the device. While the AAA batteries are relatively inexpensive and easy to replace, this project strives to create a self contained game environment that doesn't require constant battery changes. Instead of alkaline AAA batteries, a single lithium, rechargeable battery will be used.

All three projects researched utilize similar software flow. The pulse modulated laser and the Arduino game use an Arduino microprocessor. There is an extensive amount of open source code for Arduino, so many function used in those projects were most likely pulled from available Arduino libraries. This project will use an ESP32 microprocessor, which is compatible with the Arduino development environment. The Arduino libraries will be utilized and the software will be similar to that of the existing projects. The general response of the microprocessor in response to certain actions will be largely the same across all projects. Variations will come from different gameplay modes, where the code will be specifically modified to count damage in a certain way or implement certain things like "recharging" and team cooperation.

In regards to the housing for the electronic components and the physical "gun" used in gameplay, the two projects researched use pre-existing housings. The Arduino uses an existing "light gun," the pulse modulated laser project uses a modified squirt gun, and the UF team uses a commercially available Nerf gun. To keep cost low and production more efficient, a "gun" housing will be 3D printed. Certain precautions will be taken to ensure it is evident that the "gun" is a toy and not an actually weapon. This includes a clear material over the top of the "gun" and the electrical components and bright orange tips at the end of the "barrel."

3.2 Power Supply Research

This section strives to present the current research found for power sources relevant to this project. Comparisons will be made between the different types of batteries available on the market to make the case for which battery will work best for this project given the standards we set at the beginning regarding power. Circuits directly related to power distribution and control that will be used in our PCB design, such as the voltage regulator, will also be discussed. The power objectives that correspond to goals for our project are outlined below.

General Power Objectives:

Lightweight laser tag gun
- LiPo battery

Short charging time
- LiPo battery
- USB 2.0 or 3.0

Long battery life
- LiPo battery
- Low quiescent current in microprocessor
- High regulator efficiency
- Low regulator dropout voltage
- USB charger capable of providing full charge

Long standby time
- LiPo battery
- Low quiescent current
- High regulator efficiency

Long game play time
- LiPo battery
- High regulator efficiency
- Low dropout voltage

Specific Power Requirements:

3.3V voltage regulator able to supply 500 mA to drive microprocessor
- One for each unit; 4 in total

Power supply (battery) capable of providing 500 mA of current and adequate voltage to drive voltage regulator
- One for each unit; 4 in total

USB battery charger compatible with chosen battery
- Only one required

3.2.1 Battery Technology

Our laser tag guns will each be powered by a single rechargeable battery. Batteries are driven by a relatively simple reaction that converts chemical energy into electrical energy. The most basic unit of a battery is a voltaic cell, and each cell includes two half cells. These half cells are connected by an electrolyte that conducts ions between the cells. The positive ions (cations) are in one half cell and the negative ions (anions) are in the other half cell. The positive terminal of the battery is called the cathode and the negative terminal is called the cathode.

The flow of ions is determined by the voltage across the cells. During charging, electrons are added at the cathode while electrons are removed from the anode.

During discharge, electrons are removed from the anode and added at the cathode. This flow of ions correspond to the current flow through the device when charging or discharging,

The example below is a simple copper/zinc cell, where the two electrodes (zinc and copper) are submerged in a salt solution. In the solution, zinc will tend to lose electrons. The zinc metal (Zn) and the zinc ions (Zn^{2+}) form a “redox” couple, since they contribute to the oxidation-reduction reaction that drives the battery. In the other cell, the copper metal will also lose electrons. The copper metal (Cu) and the copper ions (Cu^{2+}) form the other “redox” couple of the total reaction.

Zinc has a stronger tendency to lose electrons than copper. When the cells are connected by a conductor, the electrons from the zinc half cell will travel across the conductor and bond with the copper ions, form copper metal in the reaction $Cu^{2+} + 2\ e^{-} = Cu$. The zinc cell is the anode, since it produces electrons. This where the “oxidation” (electron release) part of the oxidation-reduction reaction occurs. The copper cell is the cathode, since it receives electrons. This is the “reduction” of the oxidation-reduction reaction, where electrons are absorbed. The voltage measures the electromotive force between the cells, or difference between potentials of the cells, which is essentially how easy it is for the metals to dissolve into solution [5].

All chemical batteries operate under the same basic principles. During discharging, the cells behave as described above. During charging, the flow is reversed due to an outside force (in our case, a USB charger connected to the batteries) driving ions in the opposite direction.

Chemical batteries are named for the chemicals that contribute to the flow of ions. Common types of rechargeable batteries include lead-acid, lithium-ion, and nickel-cadmium [6]. These three types of batteries will be discussed as well as their potential to be a good fit as the battery choice in this project.

Lead-acid batteries are inexpensive and have a high power to weight ratio, which makes them ideal for use in larger scale components like motors. Some disadvantages to lead-acid batteries is the low energy density and high weight compared to other chemical batteries. Lead is also a large scale environmental problem, especially when batteries are not disposed of properly [7].

Lithium ion batteries have several advantages over other chemical batteries. They are lighter and therefore ideal for use when the goal is to create a lightweight finished product. The chemical bonds have a high energy density that provides more energy per cell than other batteries. Lithium batteries hold a charge well and can handle hundreds of charging cycles. Li-po batteries are not affected by the memory effect, which occurs when the battery will drop in voltage if it reaches the voltage point where it previously began recharging. Lithium batteries contain less toxic materials than other chemical batteries and are therefore much less of an environmental concern.

Disadvantages of lithium ion batteries include sensitivity to high temperature, relatively quick degradation, necessity of an onboard battery manager, and the possibility of destruction if the battery is discharged completely. A risk but not necessarily a disadvantage of lithium ion technology is that overcharging of the battery can result in supersaturation of the ions in the battery which will quickly shorten the battery lifetime [8].

The lithium polymer battery (LiPo battery) is a specific kind of lithium ion battery. Where a traditional lithium ion battery uses a liquid electrolyte to conduct ions, LiPo batteries use a polymer electrolyte that can be dry, gelled, or porous. LiPo cells are more energy dense, being more than 20% lighter than other cells with the same energy capacity. The self discharge rate is low, which makes it ideal for notebook computers and mobile devices. The LiPo cells have the same problems as traditional lithium ion cells, including failures from over-charging or over-discharging. Another problem with all lithium ion batteries is that the electrolyte will vaporize after overcharging and expand. This creates bad contact and can damage battery life. This problem is more prevalent in LiPo batteries, where the cell may sometimes visibly expand [9].

While LiPo batteries tend to be smaller, they are more expensive and less energy dense than traditional lithium ion batteries. Lithium ion batteries are less volatile when put e) and therefore safer. LiPo batteries are better for current control mode, since the current drops off initially then slowly decreases. This slow decrease becomes sudden at about 75% discharge. Lithium ion batteries also have a large initial current drop off, but then the discharge steadily declines.

The nickel-cadmium battery (NiCad battery) are another choice for a rechargeable power supply. Advantages to using a NiCad battery include durability, long lifetime, resistance to damage due to deep discharge, and resistance to high discharge current. NiCad batteries have a lower internal resistance than other batteries and therefore can discharge more. Its disadvantages are mostly due to the cadmium used. Cadmium is typically more expensive than other elements, it is a heavy element and therefore results in a higher weight, and it is also an environmental hazard. Another important disadvantage is that it is susceptible to the memory effect [10].The table below, dubbed Table summarizes the drawbacks and benefits of each battery chemistry.

Battery Type	Pros	Cons
Lithium-Ion	Most energy dense, light weight, lowest self discharge rate, no memory effect	Sensitive to high temp, sensitive to over charging/over discharging
Lithium-Polymer (LiPo)	Most energy dense, light weight, low self discharge, no memory effect	Sensitive to over charging/over discharging, electrolyte may expand and damage battery life
Nickel-Cadmium (NiCad)	Durability, long lifetime, resistance to over charging/over discharging	Expensive (cadmium), heavy, sensitive to memory effect

Table 4: Battery Type Comparison

3.2.2 USB Charging

For convenience, we are looking to use a micro-USB charger for the reusable battery. This allows charging either from a wall outlet or a computer, and is a commonplace connector used in many power banks and smartphones. USB (Universal Serial Bus) technology came about to standardize cables, connectors, and protocols for communication and power supply. The USB doesn't require specific device settings for data format, interrupt configuration, or I/O addresses. The single cable form means that any cable can be used in any port. USBs have eliminated the need for excess power cables because several types of auxiliary devices can be powered directly from a USB. Limitations include length of cables, lack of direct connection between devices all connected to the same host, no ability for the host to communicate to all auxiliary devices at once, and the need for an intelligent controller in USB compatible devices to implement its complex protocol. Three generations of USB have been developed since its onset, with several variations of each generation being developed between generations.

The USB 1.0 was the first edition, with a low bandwidth/low speed and a full speed version. It was not compatible with extension cables and only worked with the larger type A and B connectors.

The USB 2.0 was the next release, and included a high speed version, which was faster than the full speed of the USB 1.0. The USB 2.0 was compatible with miniature versions of the standard A and B connectors as well as the micro-USB, and included support for dedicated chargers like behavior protocols for hosts when a device with a dead battery was recognized, and an increased charging current of 1.5A and a maximum current of 5A. The USB 3.0 offers a "SuperSpeed" transfer mode which provides 5.0Gbit/s.

USB charging works in a system based on a host (either the computer or the wall) and a device (in this case, the laser tag gun). Power flows from the host to the device, but data can flow both ways. All USB sockets have four pins to match the four wires in a USB cable. Two pins carry data and two pins carry the supply of 5V. USB 3.0 ports have five additional pins, which means USB 3.0 cables must have nine wires. Our implementation will have no data flowing over the USB cable, only power.

USB 1.0 can deliver 0.5A. USB 2.0 can deliver 0.5A under normal operations and up to 1.5A for charging. USB 3.0 can deliver up to 0.9A, and supports up to 1.5A for charging. For this project, a USB 1.0 or 2.0 are sufficient to charge the single battery used. However, caution must be taken to ensure the device being charged doesn't draw more than the 0.5A provided, otherwise the system could fail. An easy way to prevent this is to make sure the host includes a current limiting circuit, which will be researched when choosing specific charging hardware.

A USB charger charges a single lithium ion battery by applying a constant current to a voltage peak of 4.2V. When the voltage peaks, current drops off which may result in only a partial charge. When dealing with lithium ion batteries, this is not much of a problem because of the problems inherent with over saturating the battery. The tradeoff is obviously a shorter runtime [11] [12] [13] [14].

3.2.3 Battery Level Measurement

In order to display the remaining battery life left for the gun, we will need an accurate method of measuring the remaining capacity. There are two ways of achieving this, either measuring voltage or measuring and integrating the amount of current passed through from the battery to the gun. Both options have their pros and cons.

By measuring voltage, it is possible to roughly estimate the amount of charge left in the battery. This requires an onboard analog to digital converter to measure the battery voltage, and a set voltage reference to compare it to. Most microcontrollers include analog to digital channels, which means that minimal extra hardware is required to implement this method.

The other method to use is to measure current going in and out of the battery. This current can then be integrated over time to calculate the amount of energy that is remaining in the battery. This method is more accurate, and does not require a lookup table. This method also has the ability to keep track of battery characteristics, such as internal resistance and capacity losses over time. However, this method is difficult to implement over power cycles of the microcontroller, as the past integration data of the battery is lost on the reset of the MCU. Measuring current also requires more hardware, as measuring current is more tricky than measuring voltage. In order to measure current, a shunt resistor, which is a resistor with a very small, known value, is placed in series with the battery. The voltage drop is measured across this resistor to determine the current flowing through it. Since the resistance is so small, the voltage drop is also very small, which requires a specialized, higher precision analog to digital converter to measure.

We chose to use the simple battery voltage measurement technique to avoid the addition of extra hardware and system complexity. In order to implement the voltage based measurement, an onboard voltage reference is necessary as using VIN as the input reference may not be constant enough for an accurate measurement. Most MCUs have an onboard voltage reference that can be utilized for this case, however they are a fraction of the MCU's operating voltage. For the ESP32, the available onboard reference in 1.1 volts. When using this voltage reference, all measured voltages must be strictly less than this voltage to be measured accurately. Since the highest battery voltage per cell of a lithium ion battery is 4.25 volts, a voltage divider using two resistors will be needed. The voltage divider can divide the voltage on a 1:4 ratio, which fits the voltage range of the ESP32 perfectly. Since resistors always have some tolerance value, the exact division is not consistent or known prior. The solution to this problem is to compare the read voltage to the max voltage when the battery is fully charged to 4.25 volts, and using that as the baseline for future calculations.

In order to convert the measured voltage into a battery capacity measurement, the software will have to transform the voltage into a capacity percentage. Since the relationship is non linear, a lookup table will need to be created and referenced within the software. This can be created through testing and automatically generated for use within the code.

Another issue to tackle is filtering noise and fluctuations in battery voltage. Spikes in current consumption such as running the haptic feedback motor can induce noise and cause a dip in measured voltage. To solve this, the measured battery voltage will be averaged using an exponential moving average. This will weight the sampled values to recent readings but also slightly mix in past values to remove noise from current spikes. The exponential moving average is used at it uses the least amount of memory to implement and requires the least amount of operations to compute. A simple moving average requires an array of samples to be stored in memory and does not provide any significant improvement in performance.

3.2.4 Voltage Regulator

To protect our microprocessor from excess input voltage, a voltage regulator is needed at the output of the battery. Voltage regulators are used to maintain a constant voltage no matter what the input voltage is. Important characteristics of voltage regulators are dropout voltage, efficiency, output voltage, quiescent current, and type.

Dropout voltage is the “minimum voltage required across the regulator to maintain regulation.[15]” A regulator with “low dropout (LDO)” is able to regulate when the input voltage is very close to the output voltage. A low dropout is advantageous to prolonging battery life and increasing efficiency. Efficiency is the ratio of output power to input power, or: $efficiency = \frac{output\ power}{input\ power}$. To minimize operating temperature and maximize battery life, a high efficiency voltage regulator is desired. Regulators are chosen by their given output voltage. The microprocessor chosen for this project, the ESP32, is 3.3V, therefore a 3.3V regulator is needed. Common output voltages for regulators include 5V, 10V, 12V and 15V.

Quiescent current (IQ) is “the current drawn by the IC in a no-load and non-switching but enabled condition. [16]” It is the current required to run the basic functions of the component, including, but not limited to, internal reference voltage, oscillators, and logic gates. IQ does not run the power state or gate drivers. IQ is important in components that operate at no output load, where a lower no-load current can greatly decrease battery lifetime. However, if the component runs in a ‘“hibernate” mode rather than a no-load mode, the processor still draws current and controlling IQ becomes less important. It is important to have a low IQ in both the voltage regulator and the microprocessor, since we are trying to design a product with maximum standby time.

The two types of voltage regulators available are switching and linear. Each type comes with unique advantages and disadvantages. The basis of the linear voltage regulator concerns the operational amplifier, or “op-amp.” An op-amp is a high gain amplifier with a differential input and single output. In the ideal case, an op-amp has infinite input impedance and therefore zero input current, zero noise, infinite common mode rejection ratio (zero common mode gain) and infinite power supply rejection ratio.

The ideal op-amp is not realizable. All op-amps have finite input impedance, small but nonzero input current, random noise, finite common mode gain and finite power supply rejection ratio. Common mode gain is measured by the output when the voltages at both input terminals are equal and the ideal op-amp would have zero output. Common mode gain comes from non-matching resistors, and is used in the common mode rejection ratio (CMRR) characteristic where CMRR is equal to the differential gain divided by the common mode gain. The ideal

CMRR is infinite, since the ideal common mode gain is zero. The power supply rejection ratio (PSRR) is also finite, since the PSRR measures the indepence of the op-amp output from the supply. Realistically, the output will always be dependent on the supply.

Slew rate is another important characteristic of an op-amp. The slew rate determines how quickly a given voltage can change. When using circuits containing op-amps, it is important that the frequency of the input to the device does not exceed $\frac{slew\ rate}{2\pi(peak\ output\ voltage)}$ when there is a large input signal. If this maximum frequency is exceeded, the device voltage cannot change fast enough to give the correct output. If the input signal is small, the frequency is limited by the formula $\frac{cutoff\ frequency\ x\ DC\ gain}{(\frac{R2}{R1}+1)}$.

The op-amp output is the difference between voltages at the positive and negative terminals multiplied by a gain. In the closed loop configuration (feedback from the output is fed to the input terminals), the output works to make the voltage difference between the two terminals zero.This difference driven output is what makes the op-amp ideal for use in voltage regulators. When a reference voltage of the desired voltage of the regulator is applied to one of the terminals, the output is tied to the other terminal in a feedback loop, and the output is constantly adjusted to make the two voltages equal.

In addition to use in voltage regulators, op-amps are used in different configurations for a variety of applications. With no feedback, an op-amp can be used as a voltage comparator, where the output is binary and depends only one which input voltage (that on the positive or negative terminal) is larger. When the output is tied back to the positive terminal in a feedback loop, the op-amp is in the positive feedback configuration and can be used as a specific kind of comparator called the Schmitt Trigger. When the output is tied back to the negative terminal (negative feedback), the op-amp can be set to invert a signal and change the phase by 180 degrees or act as a simple, non-inverting amplifier [17].

The linear voltage regulator works by using a variable resistor and controller to constantly adjust a voltage divider/op-amp network that creates a steady output voltage. The op-amp drives the circuit to keep the voltages at the input terminals equal, and the resistors determine what the reference voltage is through a voltage divider network. Although linear regulators tend to be less efficient than switching regulators where, they are simple, cost less, and tend to have a low ripple voltage and noise floor. The efficiency does become higher when there is a small difference between input voltage and output voltage, as would be expected. Linear regulators are best used for lower powered devices. As stated above, low dropout linear regulators can regulate at an input voltage very close to the desired output voltage. A linear switching regulator can be created with an op-amp, a BJT, and resistors and capacitors.

Switching regulators have a similar feedback mechanism to the linear regulator, but work by rapidly switching an element on and off, storing input energy in an inductor then releasing it to the output. The duty cycle determines how much charge is released to the load. Switching regulators are smaller, more reliable, more efficient, and output a low quiescent current, but tend to be more complex and more expensive. The ripple voltage and noise floor associated with switching regulators tends to be higher than that in a comparable linear regulator. Special switching regulators called “boost” or “step-up” regulators can also produce a larger output voltage than input voltage [18].

3.2.5 Microprocessor Efficiency

Determining how to efficiently run our microprocessor is just as important as designing an optimum power supply. The ESP32 datasheet provides several characteristics of available power modes for low-power management.

In active mode, the chip radio is turned on so the chip can transmit and receive, and the power consumption varies depending on which functions are active. Depending on transmit/receive state and speed, the board can draw anywhere between 95mA and 240mA. In modem sleep mode, the CPU is operational but the Wi-Fi and radio are disabled. At normal speed (80MHz), the board draws 20mA-30mA. The current draw increases up to 68mA for a speed of 240MHz. In light sleep mode, the CPU is paused, but peripherals are running, and the CPU can be woken up by certain “wake-up” events. This mode only draws 0.8mA. In deep sleep, only the peripherals and the co-processor are on. Depending on which of the peripherals are on, the board will only draw a maximum of 150uA. If the co-processor is off, the board draws 10uA. In hibernation, the oscillator and processors are disabled, leaving one timer and certain peripherals on, both of which could wake the board from hibernation. During hibernation, if only the timer is on, the board draws 5uA. When the chip is totally powered off, it draws 0.1uA. The minimum voltage required to power the ESP32 is 2.3V, the maximum input voltage is 3.6V, and the typical operating voltage is 3.3V. The operating current is 0.5A.

Utilizing the available power modes decreases the operating current drastically, which means the battery powering the device lasts longer. This is a key design goal in this project. The device must be able to switch to certain power modes easily. This could be done via the user interface, allowing the user to power the device down or put it into one of the available sleep modes, a time-out clock that automatically puts the device into a lower power state after a certain period of inactivity, or a certain automatic shutdown at a certain voltage level to preserve battery integrity, since most rechargeable batteries are very sensitive to complete discharge. Although battery life is a key feature, the main feature is obviously gameplay, so certain actions taken by the user should be able to wake the device easily so a game can be played.

There are other steps outside of power modes that can be taken to preserve battery life. The most obvious is to turn peripherals not being used off or to a low power state when they are not in use to avoid unnecessary current draw. Efficient cache and memory usage is important, since flash memory tends to drain a battery quickly due to the cycling required to retrieve data from the memory. Using RAM and avoiding large jumps in code can go a long way to extending battery life. As mentioned above, faster frequency results in higher current draw. Working at 240MHz is excessive for most applications. Using 80MHz is fine for most applications, since the board is waiting for instructions most of its running time rather than performing any actions. Running at 240MHz draws about twice the current as 80MHz [19].

3.3 Communications Research

Several options for communication between devices are being considered. One option is Bluetooth technology. Bluetooth is a wireless communication technology that was developed as an alternative to RS232 cables. Bluetooth works by operating at a specific frequency range: 2400-2480 MHz, with a 2MHz guard band at the bottom of the range and a 3.5 MHz band at the top of the range. Data is transmitted through “frequency hopping spread spectrum (FHSS)” radio technology. FHSS works by transmitting radio signals through a carrier that is switched rapidly to different channels using a sequence known by the transmitter and receiver. Each frequency switch is called a “hop.”

Bluetooth transmission optimizes FHSS technology by utilizing adaptive frequency-hopping (AFH) spread spectrum. AFH avoids transmitting on “bad” frequencies, which can either be crowded or experiencing selective fading. AFH therefore requires a secondary device that is able to detect “good” or “bad” channels. The data is divided into packets, and each packet of data in standard Bluetooth communication is transmitted on one of 79 predetermined channels, each with a bandwidth of 1MHz. Using AFH, 1600 hops per second can be performed. Bluetooth Low Energy uses only 40 channels, each with a bandwidth of 2MHz.

Bluetooth is a master/slave architecture, where one “master” has control over the “slaves.” Bluetooth can support up to seven slaves. The Bluetooth network operates on the master’s clock, which typically ticks at 312.5us. Two ticks make up a slot (625us), and two slots make a slot pair (1250us). If data being communicated comes in single-slot packets, the master sends data in even slots and reads data in odd slots, and the slave sends and receives in the opposite slots. Data can also come in 3 or 5 slot packets, but will always be transmitted by the master in even slots and the slave's odd slots. For a device to use Bluetooth technology, it must be able to detect certain profiles that specify behaviors of other Bluetooth enabled devices.

Bluetooth is used where low power consumption and low cost is desired, and only short range communications are needed, with the lowest allowed range is 10 meters. The highest maximum permitted power (found in Class 1 devices) is 100 mW, with a typical range of 100 meters. Class 2 devices are typically found in mobile devices, and have a maximum power of 2.5 mW and have a typical range of 10 meters. Connecting two Class 1 devices can possibly increase the range beyond the typical 100 meter range, sometimes up to 1 kilometer.

The range of every device depends on antenna configurations, battery status, and propagation conditions. They typically perform best indoors because acoustic conditions inside decrease the range to lower than the typical line of sight ranges. Devices in communication do not need to be directly in sight of each other because of the broadcast communications system [20].

An alternative to Bluetooth is Wi-Fi. Wi-Fi allows enabled devices to connect to the internet through a wireless local area network (WLAN) and wireless access point, also called a hotspot. Each access point has a range of about 20 meters.

Data is transmitted between devices on the same network through radio signals. One device sends a signal, which is transmitted to a router that decodes the signal. The decoded data is then transmitted to the Internet via the Ethernet connection in the router. Information can also be sent from the Internet as a radio wave through the router to the computer's wireless adapter. Wi-Fi signals are transmitted on one of three frequencies, or "hop" between the three. Hopping gives Wi-Fi the same advantage as Bluetooth, in that it allows many devices use the network at the same time.

The most common frequencies used by Wi-Fi are 2.4GHz and 5.8GHz. The relatively high frequencies allow more data to be transmitted. Like Bluetooth, these bands are divided into channels. Multiple networks can share channels. Wi-Fi works best at close ranges with line of sight communications, since wavelengths where the data is sent can be absorbed by obstructing materials [21].

An interesting application of Wi-Fi is the "ad-hoc" network, where Wi-Fi enabled devices can communicate without a router. Each device in these networks is a "node." In a mesh network, which is being considered for this project, nodes are connected to each other. All nodes can be connected to each other (full mesh), or only certain nodes can be connected (partial mesh). Mesh networks split communication distances into short hops, with intermediate nodes in the data path boost the signal. Dependability comes from this idea that nodes only need to send data on to the next node, rather than to the signal's final destination. This can also be a disadvantage, however, since one node dropping out means that the link is broken and performance decreases. Mesh networks can operate with or without a central server [22].

Wi-Fi and Bluetooth have similar applications, but each technology has clear advantages and disadvantages in certain situations. Bluetooth is more suited to communication between portable devices, or the "wireless personal area network (WPAN)." Bluetooth communication between two paired devices tends to be symmetric, with both devices communicating an equal amount. Wi-Fi is more suited to general networks, or "wireless local area networks (WLAN)" and tends to have a more one-sided communication with all communications directed through the access point.

Wi-Fi connections work better than Bluetooth in situations where speed is important (transmitting at around 600 Mbps) and some degree of user configuration is required. Bluetooth is a more economical choice for simple applications where device communication is minimal and speed is not critical (transmitting at only around 2 Mbps), like in headsets and speakers.

Range is another important consideration in comparing the two technologies. The range of a Bluetooth connection has a maximum of 100 meters, but closer to a typical value of 10 meters. Wi-Fi has a typical range of 30 meters inside and close to 100 meters outdoors. These ranges can be increased by adding an antenna or changing the frequency.

The general setup and hardware requirements of each technology also play a factor. In both technologies, an adaptor is required for all devices in the network. Wi-Fi networks require additional hardware in a router or access point. Bluetooth also consumes less power than Wi-Fi, and tends to be less expensive.

3.4 Thermal Considerations

Another component of the voltage regulator is its heat dissipation capability. In electrical components, temperature changes occur because of power passing across thermal resistance, much like voltage changes occur because of current passing over a resistor. This resistance occurs in two "nodes:" one from the junction to the component case and one from the case to the ambient temperature of the room. The case to ambient resistance is typically much larger than the junction to case resistance.

Changes in temperature are not ideal, because too large of an increase in temperature can raise the temperature of the component above its rated operating temperature and cause it to fail. High power dissipation and high thermal resistance creates a large temperature change. It is obvious that lowering the power or thermal resistance would be the easiest way to lower the temperature change. These characteristics are inherent to the device, and are difficult to change without greatly impacting other aspects of the performance.

Thermal resistances add in series and parallel like electrical resistances. A large resistance in a series will dominate, and a small resistance in a parallel configuration will dominate. Since inherent resistances cannot be changed, the easiest way to lower thermal resistance is to add a small resistance in parallel. This can be done through a heat sink or convection cooling through air or a fluid.

Heat sinks are the more efficient choice for cooling, because of cost, simplicity, and space. Heat sinks are typically made of highly conductive materials spread over a large surface area. The surface area comes from fins and allows more heat dissipation in a smaller space. Figure 10 shows a typical heat sink on a voltage regulator package like that which will be used in the project. It is easy to see the fins and the added surface area they create.

Since the main system is expected to draw no more than 500 milliamps, the expected power dissipation within the can be calculated. The lithium polymer battery has a maximum voltage of 4.2 volts, and the regulator will have an output voltage of 3.3 volts. This means at maximum, there will be 0.9 volts of a voltage drop. At 500 milliamps, that works out to 0.45 Watts of heat dissipated. This is a minimal amount of heat dissipated, as it will bring the regulator nowhere close to thermal runaway.

The system's current draw also leads to heat dissipation within the battery. The battery also will be charged at a maximum of 1 ampere. The selected battery has an internal resistance of 100 milliohms, which works out to a power dissipation maximum of 1/10th of a Watt. This value is small enough to be a non-issue for the functionality of the gun. To be extra safe, the placement of the battery will be placed far away from the regulator as both are the largest sources of dissipated heat. The enclosure of the gun will also be developed to be open enough to have enough ventilations for both components.

3.5 Serial Communication Research

In order for some of our components to send and receive data we need to have a set serial communication protocol between them. Some of the most popular protocols are UART, I2C, and SPI. These stand for Universal Asynchronous Receiver Transmitter, Inter-Integrated Circuit, and Serial Peripheral Interface respectively. Each of these has different requirements and limitations.

3.5.1 UART

UART is a two wire serial communication protocol. These two wires are a RX (receiver) and a TX (transmitter). UART stands for Universal Asynchronous Receiver Transmitter. The asynchronous part is referring to having no clock as a reference between devices. Instead, UART uses a common baud rate that is predetermined between the two devices that are communicating. This reduces the number of wires that are needed for communication but also makes the data need more checks to make sure messages are sent and received correctly.

The check that can be used is called a parity check. This can be even or odd parity. In even parity the number of ones in the message will be an even number. If the number of ones before the parity bit are odd then the parity bit will be a one to make the number even. If the number of ones are even before the parity bit then the parity bit is a zero. The odd parity version is the same idea but the number of ones is odd instead of even. When parity is used, it is predetermined if the communication will have even or odd parity. In order to check the the communication doesn't receive any noise that changes a signal that parity bit is checked after transmission. The way it is checked is as follows. Device one sends a message and has an even number of ones. The system is using even parity which sets the parity bit to a zero. The message is sent to the second device. One the message is received, it is checked to see if the number of ones is even. If the number is even there is no error in the communication. If there happens to be an odd number of ones, this means that some noise along the transmission of the data altered one or more of the bits. This method is not perfect for checking if there is an incorrect message received but it is better than not having any check in place. Below is Figure 8, which shows the sequence and timing of the UART.

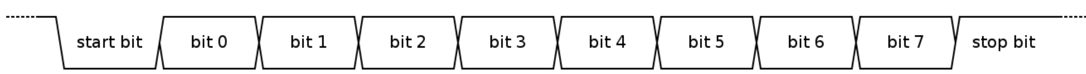

Figure 8: UART Timing Diagram. ***This work is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.***

UART sends the data using a start bit, data bits, an optional parity bit, and one or two stop bits. The bits are in order from start, data, parity, and then stop bit. The baud rate is usually one of the more common rates: 1200, 2400, 4800, 19200, 38400, 57600, 115200, and 230400 [23]. As seen in figure 12, the receive (RX_1) end of one device needs to be connected to the transmit (TX_2) of the second device and the receive (RX_2) of the second device has to be connected to the transmit (TX_1) of the first device. If there is a third device then the master will transmit to either slave and the slaves will receive the message and then transmit back information to the master's receive. This is shown below in Figure 9.

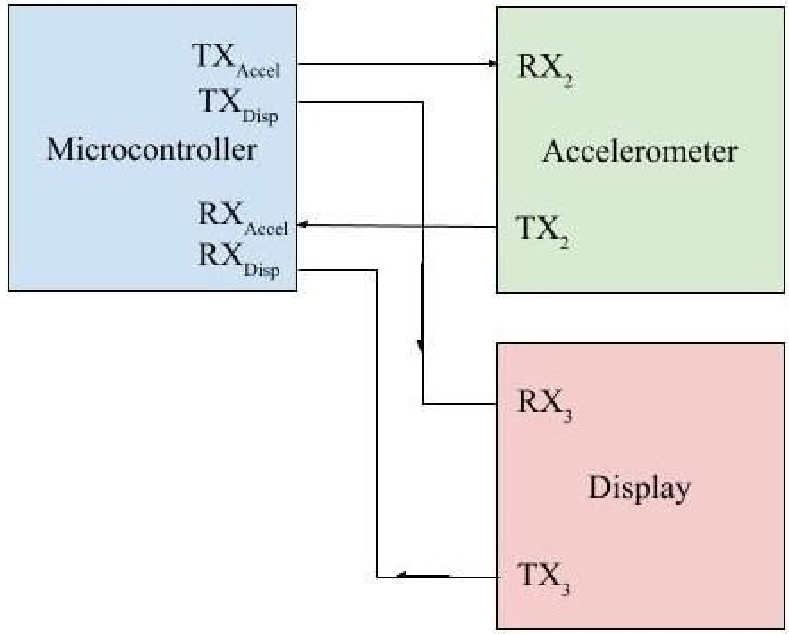

Figure 9: UART Serial Interface

UART is usually used to connect between two devices. When trying to connect to more than two devices there is the issue of more than one device trying to send data which may result in damage to one or more of the devices communicating to each other. For our project we will need to connect more than two devices. It is a concern if connecting the devices and communicating between them may lead to damaging them. If UART is the protocol we decide to use it will need to be very carefully executed in order to minimize damage to the devices.

3.5.2 I2C

I2C is a two-wire serial communication protocol as shown below in Figure 10. I2C stands for Inter-Integrated Circuit and was created by Philips Semiconductor in 1982 [24]. The company is now NXP Semiconductors. Although the protocol is free to use without a license since 2006, NXP Semiconductors charges customers to register slaves and give them addresses. I2C is intended for short distance communication.

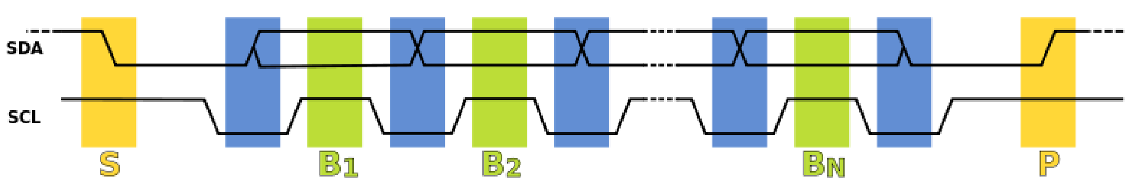

Figure 10: I2C Timing Diagram. ***This image was released to the Public Domain.***

I2C has a master slave relationship is set up where a master initiates a transfer of data, whether a read or write, to a slave that is addressed by the master. There can be multiple masters and slaves. I2C uses SCL and SDA, serial clock and serial data respectively, to communicate. The bus drivers are “open drain”, meaning they can drive a signal low and when not being driven it will be high. The two wires need a pull up resistor to bring them to +Vdd when inactive which makes them “open drain” [25]. Since only two busses are used, integrated circuits can be added and removed very easily. The interface between the Accelerometer, display, and MCU are shown below in Figure 11.

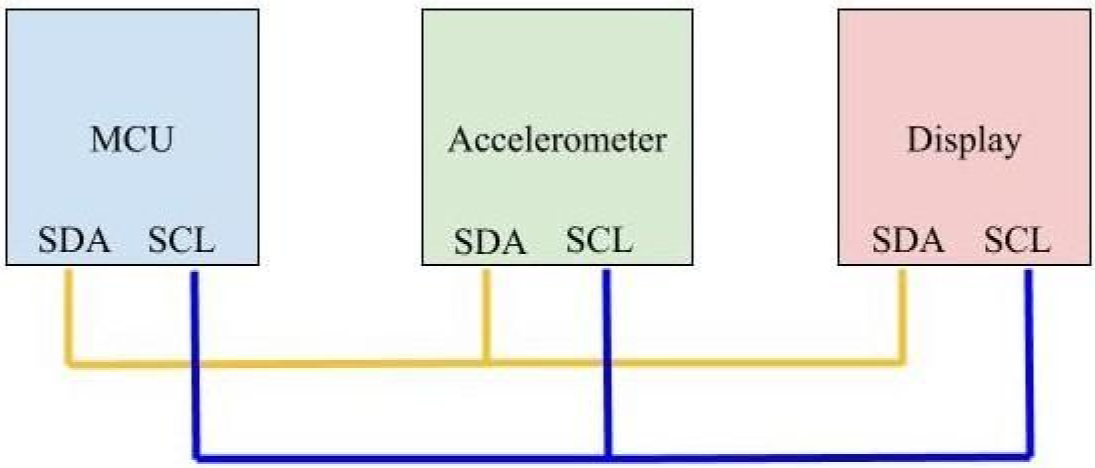

Figure 11: I2C Serial Interface

In I2C, the protocol starts with the data line being brought low. Then the address of the slave is transmitted on the data line. The address is usually 7 bits. The data is transmitted after the address and is usually in groups of 8 bits but can be any length. I2C has a few modes for which speeds can be used. It can be transmitted with 8 bits of data at 100 kbit/s in Standard mode, 400 kbit/s in Fast mode, 1 Mbit/s in Fast mode Plus, 3.4 Mbit/s in High speed mode, and 5 Mbit/s in Ultra Fast mode.

One aspect of I2C is the ACK (acknowledge) from a slave to the master. The way this occurs is as follows. The master pulls the SCL line low. The slave pulls the SDA line low once it sees the SCL line being pulled low and waits for a clock pulse on the SCL line from the master. Once the master creates the clock pulse on the SCL line, the slave releases the SDA line and the master can now drive data to the slave. The slave will do this process when the address the master has selected matches the address of the slave. The ACK lets the master know that the slave recognized the address and is ready to receive data.

For the project there will need to be connections to more than two devices. I2C easily allows for that and has a speed faster than what is possible with UART.

One of the different modes will have to be selected to decide which speed is being used and make sure that all the devices being communicated to support that mode. If needed we can have a second master, but it is most likely that we will use a single master and have a few slaves.

3.5.3 SPI

SPI is a multi-wire serial communication protocol. SPI stands for Serial Peripheral Interface and was developed by Motorola in the 1980s [26]. SPI supports one master and one or more slaves. SPI uses a clock to drive communication to the slaves. The master generates the clock to be used. SPI is a synchronous serial protocol due to the shared clock. SPI supports full duplex, as seen in Figure 12, it can send and receive data at the same time. To do so it has separate send and receive lines (MOSI and MISO).

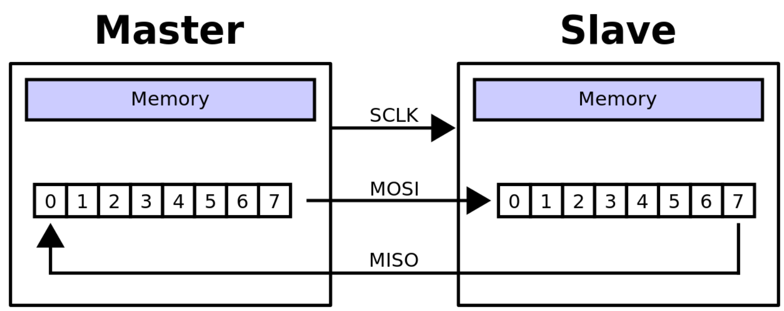

Figure 12: SPI Full Duplex Communication. ***The use of this image is governed by the GNU Free Documentation License.***

SPI uses a minimum of 4 lines for communication. These include, Serial Clock (SCK), Master Out Slave In (MOSI), Master In Slave Out (MISO), and a Slave Select (SS) line [27]. For more devices, as seen in Figure 13, more slave select lines are needed. Each Slave has a dedicated Slave Select line. For the master to talk to a specific slave it has to drive the slave select line low. The slaves can also be daisy chained if outputting the same thing to all of them, as seen in Figure 14.

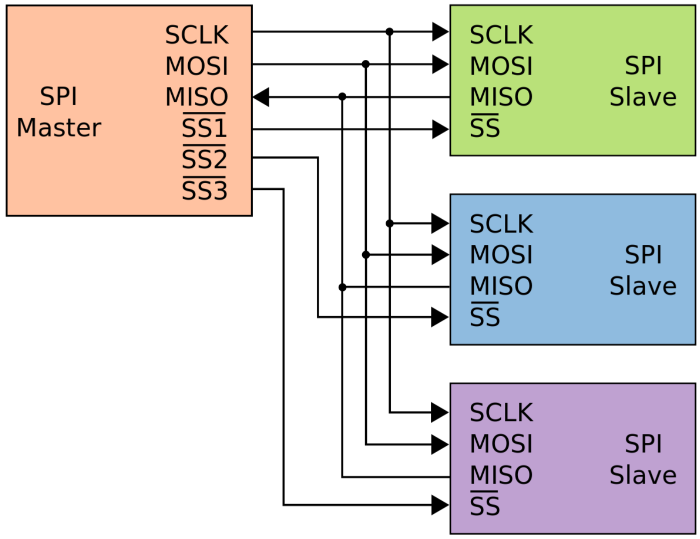

Figure 13: SPI Interface with Individual Slaves. ***The use of this image is governed by the GNU Free Documentation License.***

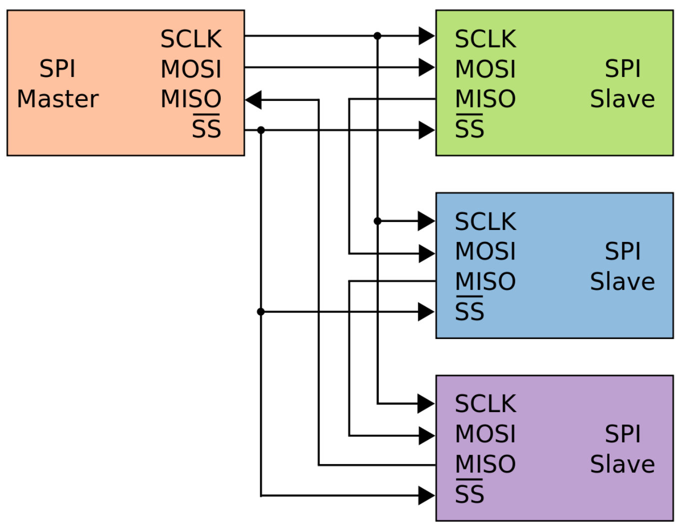

Figure 14: SPI Interface with Daisy-Chained Slaves. ***The use of this image is governed by the GNU Free Documentation License.***

SPI has to have a predetermined number of bits for data from slave. The master drives the communication so for the slave to send data to the master there has to be bits that the master leaves empty for the slave to fill with data that is being sent back to the master. Once the master drives the data to the slave, the empty bits are also driven to the slave. As the master writes to the new bits to communicate to the slave, the bits coming back from the slave have whatever information the slave needs to communicate to the master.

For the project there will need to be connections to a few devices. SPI allows for that whether it be individually or daisy-chained. The individually addressed mode will be needed which will require more wires than the other protocols. If SPI were to be used it would require more complicated wiring but would allow for much higher speeds. SPI does not have any form of error checking.

After researching UART, SPI, and I2C, we saw the advantages and disadvantages of each for being implemented as the serial communication protocol to communicate between some of the components we will be using in the laser tag gun. The components are the accelerometer and the display. The comparison of these protocols are shown below in Table 5.

Name of Protocol	Advantages	Disadvantages
UART (Universal Asynchronous Receiver Transmitter)	- Only two wires needed (TX and RX) - Parity bit checks data integrity	- Slower than I2C and SPI - Predetermined baud rate needed - Start, stop and optional parity bits not sending useful data - Only two devices can communicate at a time - Most complex hardware
I2C (Inter-Integrated Circuit)	- Only two wires needed (SCK and SDA) - Faster than UART - Supports multiple masters - Supports multiple slaves - ACK checks data successfully received by slave	- Slower than SPI - More complex hardware than SPI
SPI (Serial Peripheral Interface)	- Faster than UART and I2C - Supports multiple slaves - Simpler hardware than UART and I2C	- Needs minimum of 4 wires for two devices, one more for each added device. - Supports only one master. Slaves cannot talk to each other on their own - Communication must be well defined beforehand

Table 5: Advantages and Disadvantages of Protocols

After reviewing the different communication protocols, it was decided that the I2C protocol was the best for this project. I2C has less wires than SPI and has high enough speeds to be able to communicate between components easily. The microcontroller, ESP32, will be the master and will have the peripherals, accelerometer and display, as the slaves. The sensors found already support I2C which benefits the development of the communication between them.

3.6 Software Development Model Research

The development of the firmware will makeup a large portion of this project. Since the software needs to manage several subsystems concurrently, it will be a long and complex task. If the development of the software is not done carefully and methodically, challenges along the way can cause major slow downs in the progress of the project and even possibly a delay in graduation. To resolve this, the team plans to approach the software development using an industry standard approach to software engineering. There are two major options to choose from when adopting a development model, Waterfall and Agile.

Waterfall: The original, most simple software development model is the Waterfall method. This method simply has the team create entire iterations of the firmware before testing and verifying requirements. In this method the entire firmware is implemented, then reviewed and tested before another iteration of the code is created. Even though this method is quite common, it has several disadvantages. Firstly, Waterfall's design technique makes it difficult to parallelize the implementation of the firmware, as the entire codebase needs to be written all at once. This makes collaboration more difficult and inefficient, which will slow down development time. Secondly the implementation of the entire program without frequent testing can result in cascading failures due to bugs that were introduced at some point within the implementation. This makes tracking of bugs very difficult and tedious to patch.

Agile: The second software development technique is Agile, which breaks the project into smaller bite size chunks that are more manageable. These chunks are turned into tasks for each team member to complete, and the culmination of them results in a complete iteration of the firmware. Splitting the project into smaller tasks has several benefits, including easy collaboration as each team member can claim ownership over tasks that need to be implemented. Each task can be completed and tested independently. In practice, this results in faster development and less cascading failures to resolve later on in the code base. For this project, the team will organize and distribute tasks using a master task list hosted on Google Drive spreadsheet, which any team member can read and access to update their progress on the project. This gives a more clear picture of the progress being made on the project, and requires better planning of the project as a whole, which results in higher code quality.

4.0 Design Constraints and Standards

Standards are defined as a set of qualities that are used to describe features of, in our case, a product [1]. As engineers, we are expected to adhere to these standards such that the correct steps are taken to ensure that a quality product has been produced. Key points to standards are that they promote interoperability of both hardware and software, competition between suppliers, and ensure that the product will not impede on the users health. There are many forms of standards such as regulatory standards, voluntary standards, de facto standards, international standards, and consortia standards that are all in use in today's world. Even companies or organizations can develop their own standards to define the quality of products or services they set to emulate that is abided by their employees or members.

Standards are developed through a standards development organization (SDO) and then is accredited by the American National Standards Institute, or more commonly known as ANSI. ANSI is a not-for-profit institute that facilitates the development of standards to be used in the United States that apply to products, services, systems, and processes [2].

This section will outline the standards that apply to this project. Some of the standards listed may be voluntary standards or standards used by reputable companies such to be used as reference. Discussed will be the standards for the following:

- PCB Development and Soldering [3]
- Wireless Local Area Network (WLAN) Implementation [4]
- ABET Standards [5]
- C++ Standards[6]
- Battery Standards [7]

4.1 PCB Development

PCB development standards are outlined by the Association Connecting Electronics Industries, formerly known as the Institute of Printed Boards or ICP. ICP creates the standards from the beginning of the PCB developing process to the end and every step in between. This includes test methods, soldering, packaging, and everything between. Below in Figure 15 is an outline of how their standards connect as well as the identification number tied to these standards:

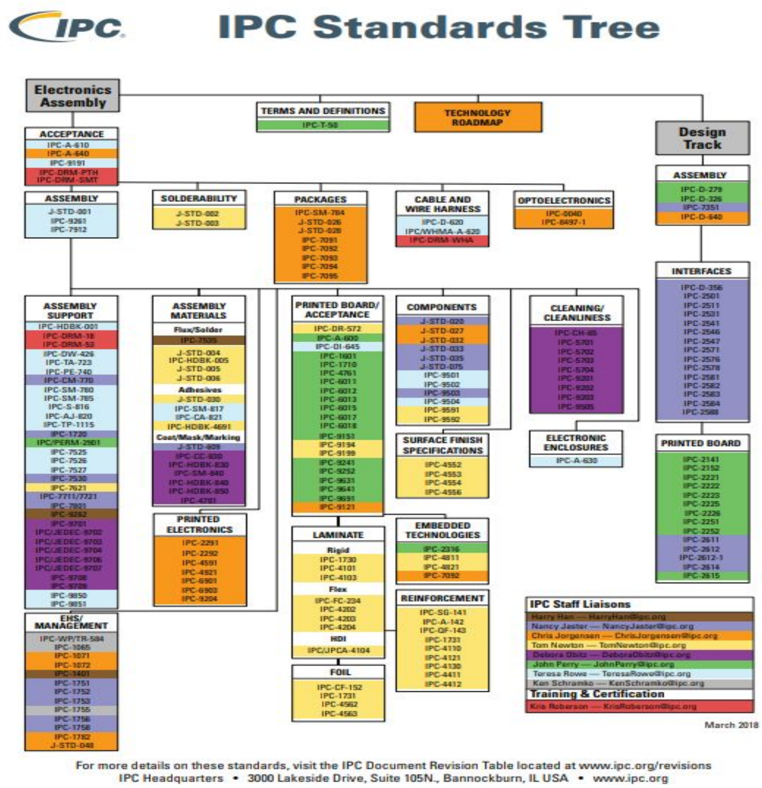

Figure 15: IPC Standards Tree *Permission to Use Requested*

Most standards can only be viewed if a membership is purchased through IPC. Since that expense is not in our current budget, we will refer to standards provided by IPC for no cost or refer to other standards that can provide us with the information we need.

One standard that is available for us to view through IPC is the IPC-J-STD-001ES which outlines the standards for soldering in electrical and electronic assemblies. Out of all of the requirements and standards proposed by the institute, standards involving soldering are the most important to us, as the PCB consumer, since these are the requirements we should abide by when the PCB is received. Though the other standards that outline how a PCB should be manufactured are just as important such that we will be receiving a safe and reliable product, the soldering portion is the only procedure we will personally perform when creating the finished prototype board. For that reason, a subsection will be dedicated to soldering which will outline the types of solder, what an acceptable amount of solder on a device is, and some description of acceptable techniques or processes.

4.1.1 Soldering

Soldering is the process of permanently connecting one electrical device to another using a metal alloy to facilitate the electrical connection. This is commonly performed to the surface mount and wired components on a PCB in order to connect the device to the copper traces, or more generally to provide an electrical connection. The verb soldering stems from the noun that is the alloy used to provide this connection: solder. The solder must be melted first before being applied to the device by the use of a hot soldering iron/soldering pen. There are multiple techniques to do this on a PCB but one common one is to have the device taped to the board to limit its movement and melt the solder over the external ports, or wires, of the device thus creating a secure and more permanent connection.

There are two divisions of solder that are used today: lead solder and lead-free solder. Lead solder was predominantly used in the past as the main soldering material but is in the process of being phased out by lead-free solder due to its negative environmental and safety effects. However, lead solder is still used today mostly when soldering by hand, such as what we will be doing once the PCB is received, since it has a much lower melting point than the lead-free solder at 188°C [8].

The standard compositions of solder alloys given in the IPC-J-STD-001ES document are Sn60Pb40, Sn62Pb36Ag2, Sn63Pb37, or Sn96.3Ag3.7 where Sn is the element name for tin, Pb for lead, and Ag for gold. Other solders can be used under the condition that they conform to the standards released by IPC in JPC-STD-006. As shown Sn60Pb40 and Sn62Pb36Ag2 are considered lead solders while Sn96.3Ag3.7 is considered a lead-free solder as it contains less than 3% lead in its total weight. The Sn60Pb40 and Sn62Pb36Ag2 are also more commonly for general application such as what will be needed for this project. While Sn62Pb36Ag2 proves to be the better quality solder of the tin-lead group since it is the most tough and can be used for a wide variety of applications (though not good for using on gold), the Sn60Pb40 is very inexpensive as compared to Sn62Pb36Ag2 and can be just as effective (though it does not do well with silver or gold either).

Also listed in the IPC-J-STD-001ES standards document is the proper technique to solder a wired component to a circuit board. First, the proper safety precautions must be met that are also defined by the standards document. According to the document, soldering should only be done in a well lit room where the lighting is at least 1000 lm/m^2. In addition, thermal protection precautions should be taken so that the devices on the board do not overheat and risk damaging themselves or the devices around it. Most datasheets for the different devices provide a time, typically in seconds, that is the maximum amount of time that a heat source such as a soldering iron can be applied for. In addition, safety precautions should be taken by the solderer such they don't get burned when handling the soldering iron. One way is to use a third hand, a set of clamps that can hold the PCB for you, such that no extra fingers are in the way during this hazardous process. Safety goggles or glasses would prove beneficial to wear in the off chance that anything sparks on the PCB or just in general to keep your eyes protected. Per usual, soldering with lead based alloys should take place under a protective hood such that the user will not accidentally inhale the fumes which could pose as a serious health risk. Just a short-term excessive exposure to lead can include abdominal pain, tiredness, headaches, loss of appetite, and even memory loss. Overtime, these symptoms can evolve to lead poisoning which can lead to kidney damage and potentially death. Since the lead's effects on health can be dampened by the use of the hood or other means to prevent inhalation, these precautions should be taken.

To ensure that our board will be soldered in the safest and most effective way as defined by the IPC standards, we will be looking to outsource this process to a reputable company unless it is deemed easily doable on our own. If it is determined that we could solder the parts on ourselves without much difficulty (ie. the parts or pads they rest on are not too intricate) we will be utilizing the space within the on-campus TI Innovation Lab where they provide the solder, solder pen, third hand, magnifying glass, goggles, and so on such that the parts can be soldered safely. Otherwise, if it is found that the level of soldering that needs to be done is beyond what the team can comfortably do, we will look to electronic companies in the area that are reputable and abide by the standards mentioned thus far in the document.

With our design, we are bound to use some surface mount components which use different soldering requirements than the wired components shown above. Since the surface mount device does not have any wire to electrically connect itself, it is impossible to exactly apply the standard shown in the figure above, however, some conclusions can be drawn. For instance, we know that the standard is in place such that not too much or not too little solder is applied. As a result, when we solder the SMDs into place, we should be mindful of how much we are applying. Though the IPC-J-STD-001ES does not exactly define the exact percentages, it does allude to the 25% of the height rule should be an acceptable percentage of solder seen around the device. In addition, SMDs require a flow of solder underneath them to secure a connection. The standards document does provide some direction on that factor saying that it should be a light amount of underflow as to not to come in contact with the contact lead seals.

Hundreds of other standards are in place just for the PCB design process alone and outline every process. For example, through IPC standards exist describing the proper amount of copper in a PCB, its composition, what is considered a defect, what are measles in a PCB, specifications for glass PCBs made with "E" glass yarn, how electrical devices should be shipped, and so on and so forth. Even though we will not be participating in the PCB manufacturing process until the very last step, it is still enlightening to be aware of the standards to understand how a good quality product should be made and to be conscious of the process.

To summarize, PCB standards, such as those involving manufacturing, packaging, and use and much more, are produced and maintained by the Association Connecting Electronics Industries also known as IPC. Focused in this document are standards involving soldering since this is the activity we currently set to do ourselves and it was determined to be in our favor to research these protocols to better educate ourselves on the process. These standards outline the type of solder that should be used, safety preparations prior to beginning the solder process, determines the proper solder that should be applied, and so much more. With this knowledge of standards, we will know how to properly solder our board such that we will limit the amount of excess solder, use the correct amount such that the device is in place and there is a secure electrical connection, and finally that our board will look more professional.

4.2 Wireless Local Area Network (WLAN) Implementation

Wifi Standards are in place in order to guarantee connectability and overall interpolarity between devices connecting over a wireless network. Since most electronics today such as smartphones, laptops, or tablets all require the use of wifi, LTE, or the up and coming 5G to connect to each other, it is vital that these standards are in place such that these devices can seamlessly jump from one wireless network to the next without any interpolarity issues. For our project, we will be using the ESP32 wifi microcontroller which uses the IEEE 802.11 security features for its WFA, WPA/WPA2 and WAPI.

The IEEE 802.11 is a set of requirements and standards the wireless local area network (WLAN) in the 900MHz, 2.4, 3.6, 5, and 60GHz frequency band range. There are more than seventeen sets of amendments within the 802.11 spectrum that cover half-duplex over-the-air modulation techniques or serve as updates or revisions to previous sets of standards. The first version (802.11 legacy) was released in 1997 and laid the ground work for the following amendments starting with specifying two separate bit rates: 1 or 2 megabits per second and the forward error correction code. It also outlines three physical layer technologies: infrared, direct sequence, and frequency hopping. That version is now obsolete but should be noted for its significance in the wifi standards field.

The standards that pertain to the ESP32 wifi module we will be using involve the ones in the 2.4GHz range, also known as the amateur radio band. The amateur radio band is a radio frequency used specifically for any non-commercial wireless exchange. This band is on a separate band than any public safety, military, aviation, or other “professional” industries that use those frequencies to communicate critical information and should not be inhibited by non-commercial or non-vital signals.

The specific amendments that ESP32 microcontroller abide by are the IEEE 802.11b, 802.11g, and the 802.11n [9]. The IEEE 802.11b was released in 1999 and was a part of a series of amendments that applied to the commercial sector such as in the home and office. Its main purpose is to redefine the throughput to 11Mbits/sec on the amateur radio band. These devices can suffer from interference problems when in the vicinity of other devices that operate in the 2.4GHz band such as cordless phones, bluetooth devices, or even microwave ovens, however, because of the increase in throughput, which defines how much of a signal can be guaranteed to be delivered successfully, and the subsequent cost decrease, the amendment was widely adopted. To prevent interference, some devices that are originally created to operate in the 11Mbits/sec realm are then scaled to send data at a lower bit rate to decrease the probability of interference affecting the signal.

The IEEE 802.11g can be seen as a revision to the IEEE 802.11b and is the standard that brings us the wifi we all know and use in today's world. The 802.11g was released in 2003 and can operate at an increased throughput of 54Mbits/sec at the same 2.4GHz frequency band, however, by using 1500 byte packets, which is the packet limit on the internet, a throughput of 31.4Mbits/sec is more likely to be seen. It is completely compatible with the devices that use 802.11b but the presence of the older devices can slow down the 802.11g network. Like the 802.11a revision, the 802.11g modulates by using the orthogonal frequency division multiplexing (OFDM) which operates by putting data on different frequencies to be sent without losing the original data and increases overall efficiency. The bandwidth that this occurs is within 22MHz.

The last of the wifi standards amendments relevant to this project is the IEEE 802.11n. The 802.11n was published in 2009 with a purpose to improve the throughput of the previously mentioned 802.11g standard. This is done by using an increased number of antennas which in return increases the data rate able to be sent successfully on the 2.4GHz to 5GHz range spectrum. The exact method of doing this is by using a multiple input, multiple output technology and by simultaneously using spatial division multiplexing; all together this process is called MIMO SDM. The use of multiple antennas allow for more information to be sent normally than a single antenna which helps increase the throughput from the 802.11g from 54Mbit/sec to a maximum of 600Mbits/sec. The 600Mbits/sec is achieved when four antennas in the 40MHz channel are used. However, this throughput is truly a maximum at best and is near impossible to achieve particularly in high data traffic environments such as in cities and other metropolitan areas. As mentioned before, the throughput is also affected by other devices within the 2.4GHz spectrum such as bluetooth. Another note worth mentioning is that the amendment only allows up to four antennas so there is a limit of how much the throughput can be increased through the MIMO method.

By multiplexing the signal, multiple streams of data, which can be independent of each other, can be sent under one bandwidth which again improves the throughput. Once the receiver has captured the signal, it can demultiplex it and get the complete data package. An analogy of this multiplexing/demultiplexing process is analysing how a movie is played on a computer. When saving a movie file on a computer, subfiles will be seen of specifically just the sound or from parts of the film (multiplexing). When all these separate files are played in unison by using the demultiplexing process, the entire movie can be seen with full audio and visual.

To summarize, standards are an important part of the engineering world and help us achieve excellence and inspire innovation. Without the IEEE 802.11 standards, wifi itself would be a product by product service that would most likely be inaccessible unless your device is designed for that specific wifi's specifications. In other words, there would be a lack of the interoperability that we enjoy today. With the 802.11 standards, particularly the 802.11b, 802.11g, and 802.11n, we now know the standard throughput in our products that use wifi in the 2.5GHz band range and can plan our project around that number accordingly.

4.3 Realistic Constraints

ABET, who is responsible for providing accreditation for engineering programs at universities, lists its own set of standards that students are expected to abide by while designing, building, and demonstrating their final project in order to get a degree in engineering. During this process, the students are expected to work on a team and solve engineering problems using the skills and resources they acquired in their undergraduate program. The project itself needs to abide by economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability constraints. In this section, these constraints as they apply to our laser tag project will be discussed.

4.3.1 Economic

Economic constraints will be one of the more important limiting factors to this project. Since this project is 1) self funded and 2) has a goal of producing an affordable laser tag game, the cost of this project should be low cost. One way to keep the effects of this constraint as minimum as possible is to utilize existing equipment and components offered at the University of Central Florida to keep the overhead as low as possible. These areas include the Senior Design Lab where oscilloscopes, multimeters, and computers are provided and the TI Innovations Lab where there is an available solder station as well as a 3D printer. Other methods include choosing economically efficient parts and, most importantly, double checking what is being ordered (from components to the

PCB) such that funding will not be wasted ordering a wrong part or a board that will prove to be not designed correctly.

4.3.2 Environmental

Environmental constraints are constraints that restrict the impact on the environment caused by the laser tag assembly. These constraints can be applied to the complete assembly as well as the manufacturing process. Some constraints are impossible to abide by, such as with soldering, since the current standard is not completely environmentally safe to begin with. However, environmentally conscious decisions will be made such that this project's impact on the environment will be as minimal as possible. One such way is to side with RoHS certified products which both can minimize environmental as well as health and safety impacts by setting the maximum level of restricted substances allowed in electrical products such as PCB boards. The laser gun itself has no major direct impact on the environment other than the electrical consumption it requires and, as a stretch, the IR led. The IR led emits light outside of the visible spectrum so it will not contribute to any sort of light pollution and with the addition of the low power function in the laser gun, the power can be more efficiently emitted and hence would require less frequent recharging. To further decrease this projects environmental impact, steps will be taken such that the least amount of components will be wasted and we will try to reuse them the best that we can.

4.3.3 Social and Political

With a goal of bringing children of all ages together to play laser tag, the social impacts on this project are positive in nature. However, in today's age where people are weary seeing even toy guns in public, special consideration needs to be taken for the look of the laser gun. This will be done as to protect ourselves and the people around that and to ensure that the laser tag gun will not be mistaken as a live gun. Title 15 of the Code of Federal Regulations part 272 [10] outlines regulations for toy guns or "imitation firearms". These constraints are necessary to abide by in order to produce or sell any gun look alike within the United States. Listed are approved markings for the laser tag laser gun:

- A blaze orange (Fed-Std-595B 12199) or other bright orange color cap or plug permanently fixed to the end of the barrel of the device
- A blaze orange or other bright orange color around the barrel of the device
- A transparent material used for the entire exterior of the device such that the interior can be clearly seen and identified as a imitation firearm
- White, bright red, bright orange, bright yellow, bright green, bright blue, bright pink, or bright pink colors used in any configuration as the primary color of the device

For this project, made sure to abide by these constraints and have plans in place in order to do so. For instance, the housing for the laser gun electronics has been manufactured using a 3D printer where the filament of the above listed colors are widely available. Other characteristics that were not listed in Title 15 will be used so that the laser gun looks as non-realistic and as safe as possible. One is the utilizations of sound effects when shooting that will seem more futuristic and fun sounding than that of a real firearm. Another is to not implement any type of scope to the device or any unnecessary aesthetic addition that will make it look too realistic. Other precautions will be made such as contacting the local law enforcement (i.e. UCF Police Department) prior to when we plan to bring the laser tag guns out to test or display for demonstration.

These considerations will mainly affect the physical design of the projects housing. The gun will have a two part design, with the top half comprised of a clear/translucent plastic cover, and the bottom half being a plain white. The plain white will make it clear that the laser gun is simply a toy, and the clear translucent housing will allow the onboard colorful LEDs shine through, which will make it even more evident that the there is no danger involved with the project. The 3D modelling will be done with inspirations taken from free-to-use sources that depict futuristic or alternative style weapons, instead of realistic firearms to further ensure the safety of our users.

4.3.4 Health and Safety

Overall, the health and safety effects caused by this project will be minimal. The main concerns that can affect the user’s health and safety stem from the electrical devices and the IR led. For the electrical devices in general, it is imperative for both our safety and the user’s safety that everything is wired correctly to the PCB such that the potential of electrical shocks occurring is as minimal as possible. In addition, ventilation inside the housing should exist such that the parts on the PCB and the battery do not overheat which in extreme cases can lead to malfunctions and could seriously harm the user.

Studies have shown that prolonged exposure to IR leds within the 760nm to 1400nm range in particular can damage cells. This, however, is seen in extreme cases and of these damaging IR waves, the worst ones are emitted by the sun and their effects are strengthen through heat exposure. After thorough research, no direct health effects have been documented in regards to the type of IR led we will be using with its intended use. Since the IR led will only be on intermittently and its intensity is very weak as compared to the sun, it is difficult to conclude that the user will have an increased health risk using the device in that respect as opposed to going outside to stand in the sun.

To promote the health of the user using the device, a pedometer may be implemented to record step count while playing laser tag. Seeing a visual number showing someone's athletic progress has a positive effect on the user and will encourage them to go out and play more. Though this will not make the laser tag any safer or more dangerous, it will encourage the user to go out and maintain a healthy lifestyle through exercise and fun.

4.3.5 Manufacturability

The manufactuability constraint brings awareness to how the project will be manufactured. It has been discussed throughout the document how the heart of the project, the PCB, will be manufactured by a qualified company as well as the housing where the PCB and the electrical components will sit. To summarize, the PCB will be manufactured by JLCPCB, or another economically friendly PCB manufacturer, the electrical components will be ordered through a reputable source or when available provided by UCF, and the housing will be built using a 3D printer.

4.3.6 Sustainability

Sustainability involves such constraints that may hinder the project from working under normal operating conditions. The goal of the project is that the laser gun will be able to be used both indoors and outdoors so the sustainability of the project will depend heavily on the environment. For instance, if played outdoors, humidity will be a concern as well as rain or excessive heat. Precautions will be taken in the design of the device such that the constraints that these conditions impose will not hinder the device's performance.

4.3.7 Ethical

The ethical standard is one that is paramount for the integrity of this team based project. IEEE offers a ten point standard of what they define as a proper ethical code to follow which is summarized below[11]:

1. To hold safety and welfare of the public of the highest concern; to disclose any factors that may impair the public's or environment's safety
2. To avoid conflicts of interest as much as possible and disclose any that may arise
3. To be honest and make claims that are based on accurate, available data
4. To reject bribery
5. To help others understand the new emerging technologies, their capabilities, and their net effect on society
6. To maintain and better our own technical knowledge
7. To be able to receive and give honest criticism in a professional manner and to credit appropriately the work done by others
8. To treat others as you would like to be treated, to not participate in any acts of hate towards another based on who they are

9. To avoid hurting others either physically, mentally, or in a way that ruins their reputation and/or employment
10. To assist others to follow this code

These ten points, though do not directly apply to the laser gun itself, do apply to the group and how we should behave as a professional team. The code drives to promote openness to the public on the emerging technologies particularly by teaching and overall educating in a responsible manner. This code also promotes accountability of oneself outside of the professional workspace by disstaining hatred of any form or favorability of one person over another because of biological reasons. As young electrical and computer engineers, we look to abide by this code beginning with this project and going forward in our professional careers.

4.3.8 Summary of Realistic Constraints

In summary, the ABET constraints bring up concerns in each step of the manufacturing, building, and deployment process that directly affect our project. For this particular project, the most important of the constraints proved to be the Social and Political constraints since there could easily be fatal repercussions if not abided by. Overall, however, each constraint should be treated equally as important since they all apply to the functionality of the project and likely affect the design of it.

4.4 C++ Standards

Since C++ will be the primary language used in this project and is integral to the project's functionality, it is appropriate to discuss some of the standards and constraints that come with using the language. For a complete overview of the language, the ISO/IEC JTC 1/SC 22/WG 21 document, which brings complete standardization to C++, will be referenced. The ISO/IEC JTC 1/SC 22/WG 21 was created in partnership between the Joint Technical Committee (JTC) and the International Electrotechnical Commission (IEC) which are subcommittees of the International Organization of Standardization (ISO). Since the document is dense and goes into more detail beyond the scope of this design report, other sources will be more heavily discussed that provide more concise and practical standards to follow. These sources, when discussing more concise techniques, will be as up to date as possible since it is imperative that the most modern techniques are used in an ever changing field.

The C++ script for this project should abide by a set of standards/constraints that ensure that the code will be accepted from the date of creation to well into the future. There exists a document produced by Lockheed Martin Corporation for one of their projects which outlines some constraints for time critical systems that our group agrees to and accepts as constraints we should follow for the course of the project [12]. Namely, the most important of these constraints are reliability, maintainability, testability, reusability, extensibility, and readability. Reliability entails that the final code should run as intended 100% of the time. The reliability of the code should not be hindered by any external conditions and should execute without fail every time. This is a core standard for this project since it would be in poor engineering taste to demonstrate a project that only works in very specific conditions or has buggy code. This is a sign that the project was either poorly designed or had lack of work put into it which will be avoided. Maintainability says that the code should be written in a way such that it could easily be updated and debugged at any point in the project's lifetime and beyond. Since it is likely that a few versions of this project's code will be built before putting together the finalized code, this is an important feature. The next standard, testability, says that the code created should be made in a way such that it is simple to test. For this, the code should be as simple as possible so it is simpler to debug manually and insert test statements. Test statements will be used extensively in this project's script so that the changing variables and states can be checked at multiple points which can help determine where and why the code fails. Another technique that will be utilized is making individual code for each function and testing them separately. If all the functions can run on their own then it can be ruled out later in the debugging process that the code does not work due to an error within the individual source codes. The reusability characteristic will be extensively used in this project's script to minimize the time spent creating working, commonly written code. One leading reason why it was chosen to work in C++ was that there exists easy to find, open source code that can be implemented into this project for the basic functions. Though using open source code is not the solution for the entire project, it will save time in building code for the commonly written functions such as reading the trigger's state. Extensibility ties in with maintainability in which as requirements evolve overtime, the code should be able to evolve with it. Hence, the code should be written in a way that it can be rewritten to conform to the new standards without heavily changing the core or source code. The final constraint is readability which says that the code should be written in a way that is easy to read by the user just as easily as the creator can read it. This can be difficult in C++ since it is not the most renown languages when it comes to simple syntax, but by using clearly defined, non ambiguous variables and adding comments in each section the code will be easier to read.

The following describes how the code should be initially set up to conform to the constraints listed above. To improve readability, the first line should contain a comment describing what the code's function is and what it does. This is so the user, or the person who is not the creator reading the code, knows exactly what the code should do and can decipher it better. Following should be the declaration of the preprocessors where the first should that be called are the header files. This is called by using the *#include* function then inserting the desired header between the less than and greater than symbols (< and >).

Following the Lockheed Martin document cited earlier, only *#include, #define, #ifndef*, and *#endif* directives should be used such to limit the number of preprocessors and to only use additional ones where absolutely necessary. This is to again improve readability by reducing the clutter in the beginning parts of the code. After all initial declarations have been made, a white space should be added then the int main () function will follow. This marks the start of the executable code where all the functions will be defined and executed. This is all contained between two curly brackets that should sit on their own line. Within the curly brackets, it should be noted that each expression or statement needs to be terminated using a semicolon.

Though this section will not go into too much detail regarding coding style as to constrain the creator, some important techniques will be discussed. The proper style that this project's code will abide by as well as the do's and don'ts of C++ should also be defined as to follow the characteristics listed earlier. To begin, the *#define* preprocessor should not be used to define constants, as the creator has no control over the length of the constant when using that preprocessor. Instead, the function const should be used in this scenario. To improve readability, the script should be formatted in a way so that each line does not contain over 120 characters and each expression or statement should be on its own line.

An often overlooked topic, the way variables or identifiers should be named is vital to the readability, maintainability, and extensibility of this project. If the identifiers are poorly named, it would be difficult to remember which identifier does what when reading through the code and can result in critical errors if accidentally misused. To reduce any confusion, the following naming conventions will be used. In general, the name should consist of the following qualities:

- Should be relevant to its function or use
- Should be descriptive yet not exceeding twenty characters in length
- Can include abbreviations as long as it is known what the abbreviation stands for
- Any acronym used should be uppercase
- Any words used should be separated by an underscore
- Will not differ from other identifies through the presence or absence of a single character or by capitalization
- Will not differ from other identifiers through the interchange of a single letter to a number or vice versa
- Will not begin with an underscore or other special character

All names of identifiers will be uniformly applied to the above characteristics such that to improve readability. With a proper naming scheme, the code can be easily organized and it will be simple to identify where specific variables came from.

Where it is difficult to name an identifier properly but not be excessive in length, comments will be utilized to further explain the identifier to the person reading the code. It should be cautioned that it would be undesired to add a comment for every variable or statement, however, since at a distance it would be hard to easily piece out the executable code from the comments and would overall look cluttered.

Next the cons of the C++ should be identified and it will be discussed how these limits to the language are seen as constraints. As compared to other object oriented languages, such as Python, C++ has a more complex syntax which can be difficult to master as a first time user. In addition, there is little to no flexibility to the syntax which makes it even more tedious to learn. However, there is a lot of support for the language so there exists many tutorials to help learn the language. Another con is that the language is very unbinding, or in other words the programmer is allowed to do a lot more. By most standards, this would be seen as a good thing but in this case it also allows for many things to go wrong. As a result, the programmer needs to be diligent and be able to check for any potential errors themselves using a series of test statements.

C++ has very poor memory management. It has little procedure in place in order to properly utilize its memory and as a result leaves the programmer to do it themselves. One notable instance of C++'s poor memory management is its lack of garbage collection. With the garbage collection, reclamation of memory space will be automatically done for memory tied to objects that are no longer in use. Since the microcontroller that is being used for this project has limited memory, having the garbage collection is an important feature, however with a little extra effort, the process can be done manually by the programmer.

Summarizing, the C++ language was chosen for this project since it can execute what is needed for this project, has a lot of support, and is compatible with the ESP32 microcontroller. For this project, it is important that the code abides by the standards of reliability, maintainability, testability, reusability, extensibility, and readability so that the code is made in the most efficient and smartest way possible. One method to achieving this discussed is to name the variables in a way that is clear and concise as to reduce any confusion between the function of any two different variables. Also noted were the constraints caused my the language itself, particularly its lack of memory management, which needs to be watched when the coding process begins.

4.5 Battery Standards

The final standard that is relevant enough to our project to the point that it should be discussed are the battery standards. Since the battery is arguably the more dangerous electronics we will be using, we felt it to be appropriate to discuss the standards existing standards in terms of testing and so on. These standards will tie into the health and safety standards discussed earlier since they involve ways to keep the user safe from a potential failure of the battery which may result in an explosion.

In this discussion, we will focus on the lithium-ion (Li-ion) battery standards since that will be the type of battery that will be used in this project. The standards for this type of battery are produced by the International Electrotechnical Commision, also known as the IEC, though other standards and reports are outlined by companies such as UL and IEEE. The IEEE Std 1625-2004 [13] presents standards for rechargeable cell packs intended to be used for cellphones and other "portable computing devices". More specifically, this standard document defines different aspects of the design, test, and evaluation of a cell battery such that they will be safe for the consumer to use. Though we will not be using batteries designed for cellphones, these standards can be viewed as reference to understand how smaller cell packs, similar to the ones we will be using, are designed, tested, and implemented. It should be noted that there exists a 2008 revision to this standard document, partially due to the lithium-ion battery explosion crisis occurring around that period, but we choose to look at the 2004 version since it is free for public use and offers, for the most part, the same material as the updated verison.

The first set of standards discussed in the IEEE Std 1625-2004 are the design standards. These standards do not describe the numerical qualities of the cell pack, such as height or weight, but stress how the battery should be built in order to reduce the probability of a critical failure. For instance, section 5.1.2 of the document sets that the material used for the separator should be thick or strong enough to prevent any penetration of the battery. It stresses that if a puncture were to occur by a conductive material then there is a risk of causing an internal short that will inevitably damage the device beyond saving. In addition, the electrodes preferably should be free of any wrinkles, cuts, or other impurities, however, if they do exist they should be evaluated by their severity and whether it is deemed that it will hinder the overall safety and function of the battery or not. Other design standards are in place that promote the safety of the battery such as implementation of vents to relieve internal pressure or the use of a overcurrent protector.

Though we do not participate in this design process, is it important to be aware of this set of standards such that we do not use the battery in a way that can hinder or limit the performance of these built in safety features. These user concerns are outlined later in the document under section 6.8. These "mechanical stresses" directly relate to how the battery should be placed in the housing in the safest way possible. One point made is that the cells should be electrically insulated from each other in order to prevent any unwanted connections. This should be perfectly feasible in our housing and will likely be implemented in the final design as a safety precaution. Other standards that are listed and will be abided by to the best of our ability are to alleviate any stress on the connectors running from the PCB to the battery pack and to not block any of the vents.

Other precautions are taken during the design process that include short-circuit, overheating, and overcharging precautions. To prevent severe damage to the load caused by short circuiting and the large current that would follow, it is stated that the battery should have at least two independent methods to limit the output current into the load device. One of these methods may include protective circuits such as a current limiter circuit. Likewise, a charge control circuit should be implemented to control the rate of charging and to prevent an overcharge from occurring. This limit to the maximum charging voltage and current should be defined by the manufacturer based on the battery's characteristics. On the opposite end of the spectrum, the over-discharge minimum voltage and current should be set by the manufacturer and enforced by at least one undervoltage protection circuit that will turn off the battery if triggered. A thermal protection circuit should be implemented as well to prevent the battery from running when it is running above the maximum temperature set by the manufacturer. In any case, when the maximum temperature is exceeded, the battery should have some system in place that would be able to recognize that it is too hot and perform a protective action, such as turning off, in order to protect the battery and the device it is powering.

The IEEE Std 1625-2004 standard also proposes some user alerts that should be implemented into a system to notify the user when something with the battery is awry with a recommended solution. These notifications may include abnormal battery temperature, abnormal DC power draw, low battery, and so on. This may be useful to have and may be made a stretch goal in order to make the device as safe as possible and to give the user fair warning when the battery is in danger of failing.

An excerpt was provided in the document from the IEC 62133 standard that describes test procedures and conditions to perform in order to properly test the cells. Though we will not be performing these tests ourselves for safety reasons, with the knowledge that the batteries we have purchased have passed these tests, we know some of the stresses they should be able to handle without breaking. All of these test occur in 25 +/- 2°C, except for the thermal abuse test, and test "foreseeable misuses" such as an external short circuit, free fall, crashing, forced discharge, overcharge, and low pressure. To keep the discussion brief, the tests reviewed in more detail will be the ones that are more likely occur during use of our project: free fall, crash hazard, thermal abuse, overcharge, and crushing. For free fall, the fully charged cell is dropped three times from a height of one meter onto a concrete or otherwise hard floor. If there is no fire or explosion, the battery has passed that test. Crash hazard tests brings the fully charged cell to its peak acceleration and presents three shocks of equal magnitude against three perpendicular directions. If there is no fire, explosion, or fluid leakage the battery has passed. With the above two tests, it can be reasonably assumed that the batteries incorporated into the project will be able to survive if the laser gun were dropped of if the user accidentally slams it into a wall or furniture. The thermal abuse test outlines that the fully charged battery should be left in an environment that is about 130°C for ten minutes. If it survives this test then it passes. This test tells us that the batteries could easily survive for a period of time if left in a hot car which can reach average temperatures of 47°C on a hot day. The crushing test involves reinforcing a force of about 13 kN to the entirety of the device to effectively flatten it. If it does not go up in flames then it has passed. The final test that will be presented is the overcharge test which is useful to know just incase the charging system fails and accidentally overcharges the battery. For this test, a discharged battery will be connected to a power supply greater than 10V and be provided a recommended constant current that is specified by the manufacturer. The time it should stay connected to this power supply is a function of the current and the total charge it can hold. If it survives without combusting then it has passed.

As a final note on the lithium-ion battery standards, the document gives a list of information that should be provided to the end user when the battery is in use. This list is summarized below:

- If the battery were to leak, do not let the liquid get into your eyes or skin. In that event, wash the area with sufficient amount of water and seek medical attention
- Seek medical attention if the battery has been swallowed
- Do not leave the battery charging for elongated periods when not in use
- Batteries generally perform best when stored and used at room temperature
- Do not disassemble or otherwise damage the battery
- Dispose of old, worn battery in the method provided by the manufacturer/provider

If this project is ever patented and sold to the public, these warnings would be placed on the packaging or otherwise in an instruction/warning manual for the device. For the sake of the senior design course, these warnings may be placed along with other legal/safety related disclaimers on a sheet of paper on display if deemed appropriate by the faculty.

To summarize, the battery standards we will be adhering to are the IEEE Std 1625-2004, though the 2008 revision would be beneficial to view as well. On the manufacturing side, safety features should be built into the battery to such that to reduce the probability for there to be a fatal accident in the event that the battery proves to be defective or damaged. Redundancy is used for safety features such as using short-circuit protector circuit in order to protect the load from the negative effects of a battery failing. A multitude of tests are performed on a random selection of batteries in each batch in order to test if the batch is safe and was built to the standard before sending them to the consumer or retailer.

5.0 Project Design

This section will go through the different components that have been selected and go more into detail as to the design that the part is intended for. Designing the the firmware part of the system which encompasses software options, multitasking, infrared communication, as well as wifi mesh networking design.

This section will go through design of the following topics:

- Initial Parts Research and Design
- Firmware Design

5.1 Initial Parts Research and Design

This section includes all parts considered for use in the project and the specific part ultimately chosen based on general project goals and specific project requirements. An important spec to pay attention to in all parts ordered is the soldering tolerance. The data sheet for each part will specify how long the part can be exposed to a certain temperature of solder. If this time limit or temperature limit is exceeded, the component could be compromised. Following these maximums is crucial when assembling the PCB.

The power supply for the “gun” consists of three main components: the battery, the battery charge/status monitor, and the voltage regulator. Design decisions, lists of potential parts, and final part choices are detailed in sections 5.1.1 (voltage regulator) and 5.1.2 (battery/battery charge monitor). The datasheets for the final selected parts can be found at the URLs given in Section 9.

5.1.1 Voltage Regulator Research

Because of the low operating voltage and relatively small power consumption of this project, both linear and switching regulators could be used. Both are considered in the following parts lists, and the decision will ultimately be made on price, availability, and preferable characteristics. The regulator must have an output voltage of 3.3V to power the microprocessor, a maximum current of 0.5A to preserve the integrity of the microprocessor, and it must be small enough to fit on the PCB with all other necessary components. It is also ideal if the regulator has a low quiescent current and low drop-out voltage to preserve battery life and maximize efficiency. The current and voltage characteristics of potential voltage regulators are given below in Tables 6 and 7.

Options for mounting type are through-hole or surface mount technology (SMT). Through-hole connections use leads that are inserted into holes in the PCB. The leads are then soldered to pads on the board. For improved contact, the holes for the leads are plated. The through hole connections are strong, but are more expensive and take away routing area for signal traces because of the need for drilling. Through hole connections are usually best suited to larger and heavier components. Surface mounted devices (SMD) are placed directly onto the PCB. SMT speeds up the production process but has an increased risks of defects because of the denser packaging of components and the layout of the board itself. SMT components require either very small leads or no leads at all, which means they are smaller than through-hole components. Alternatives to leads on SMDs include flat contacts, terminations on the body of the component, or solder balls. The part should have a solder mount connection so it is easily mounted onto the PCB. For the purposes of this project, a small, cheap, and simple solution is required for a device that does not draw extraordinarily large voltage or current. Therefore a surface mounted device is ideal. All parts compared in the tables below are surface mounted devices.

Mouser #	**Mfct. #**	**Mfct.**	**Min. Input Voltage (V)**	**Output Voltage (V)**	**Dropout Voltage (V)**	**Unit Price ($)**
595-UA78M33CDCYR	UA78M33CDCYR	TI	5.3	3.3	2	0.58
595-UA78M33QDCYRG4Q1	UA78M33QDCYRG4Q1	TI	5.3	3.3	2	0.67
595-UA78M33CKVURG3	UA78M33CKVURG3	TI	5.3	3.3	2	0.67
584-ADP3335ARMZ3.3R7	ADP3335ARMZ-3.3RL7	Analog Devices Inc	3.5	3.3	0.2	2.65
584-ADP124ACPZ-3.3R7	ADP124ACPZ-3.3-R7	Analog Devices Inc	3.43	3.3	0.13	1.33
595-TPS63070RNMR	TPS63070	TI	3	Variable	N/A	2.59
NN/A (found on LCSC)	HT7333-A	Holtek	N/A	3.3	0.09	0.23

Table 6: 3.3V Voltage Regulator Voltage Characteristics

Mouser #	Mfct. #	Mfct.	Output Current (mA) for Vo=3.3V	Quiescent Current (uA)	Unit Price ($)
595-UA78M33CDCYR	UA78M33CDCYR	TI	500	*not given in data sheet	0.58
595-UA78M33QDCYRG4Q1	UA78M33QDCYRG4Q1	TI	500	*not given in data sheet	0.67
595-UA78M33CKVURG3	UA78M33CKVURG3	TI	500	*not given in data sheet	0.67
584-ADP3335ARMZ3.3R7	ADP3335ARMZ-3.3RL7	Analog Devices Inc	500	1.00E-02	2.65
584-ADP124ACPZ-3.3R7	ADP124ACPZ-3.3-R7	Analog Devices Inc	500	4.50E-02	1.33
595-TPS63070RNMR	TPS6307x	TI	500	50	2.59
N/A(found on LCSC)	HT7333-A	Holtek	250	7	0.23

Table 7: 3.3V Voltage Regulator Current Characteristics

Based on the project goals, simplicity and cost effectiveness are favored, therefore making a linear regulator the best choice. The regulator found with the best characteristics is the Holtek linear regulator. It is incredibly low quiescent current and dropout voltage for such a low price make this regulator a clear choice. The application circuit used in the PCB for the Holtek regulator is shown below in Figure 16. Because of the wide range of input voltages, it is important to make sure that the input voltage is as close to the output voltage as possible to make the power supply as efficient as possible. As stated in the project research portion of the document, efficiency is the greatest when the input and output voltages are as close together as possible. This should not be a problem, since the chosen lithium ion battery is only rated at 3.7V, where the maximum allowed input voltage for the regulator is 12V.

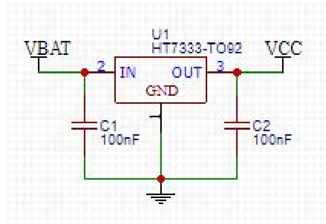

Figure 16: Holtek Voltage Regulator Application Circuit from PCB

5.1.2 Battery Research

The need for a light-weight but durable and energy dense battery has resulted in a lithium battery being chosen. A comparison of different lithium-ion batteries are shown below in Table 8 which aided in choosing the proper battery for this project. Lithium-ion specifically has been chosen, because it is cheaper and more energy dense than Li-Po batteries. It is necessary to find a battery small enough to fit in the housing of the laser tag gun and be compatible with the rest of the PCB but also have enough capacity to provide sufficient voltage to the microprocessor and an adequate playing time. Because of the way the battery will be attached to the battery monitor described below, a solder tab connection is what is needed for this project. The chosen battery will be soldered to the light blue connections shown at either terminal in the diagram.

Manufacturer	Digi-Key Number	Manufacturer Number	Rated Voltage	Capacity	Charge Time	Unit Price
SparkFun	1568-1488-ND	PRT-12895	3.7V	2.6mAh	4 hr	$5.95
SparkFun	1568-1490-ND	PRT-13189	3.7V	2.6mAh	4 hr	$6.50
SparkFun	1568-1494-ND	PRT-13856	3.7V	6mAh	N/A	$29.95
Adafruit Ind.	1528-1838-ND	258	3.7V	1.2mAh	N/A	$9.95
Adafruit Ind.	1528-1857-ND	2011	3.7V	2mAh	N/A	$12.50
Adafruit Ind.	1528-1835-ND	353	3.7V	6.27mAh	4 hr	$29.50
Great Power Battery Co.	N/A	18650	3.7V	2.6mAh	4 hr	$6.50

Table 8: Rechargeable Lithium Battery Initial Parts Comparisons

The Great Power Battery Co. lithium battery was selected because it delivers all desired characteristics and also has the termination type (solder tab) desired for this project. While a coin battery would be ideal given the size and weight goals, the capacity and connection type of this round cell battery makes it a better choice.

In addition to the battery, a module is needed to monitor the charge level of the battery. This is necessary because the lithium ion battery is so sensitive to extreme levels of charging/discharging. The part chosen is the TP4056 from NanJing Top Power ASIC Corp. This module maintains constant current and voltage while charging the battery, and is compatible with USB adapters. LEDs on the module represent different charging states. Different colors (red and green) show if the device is charging, finished charging, if the battery temperature is outside of normal operating conditions, or if the input voltage is too low. The application circuit in Figure 21 shows implementation of the TP4056 with the LEDs in place and all required inputs and other circuit components. Using the TP4056 has several advantages. Although much of the battery charging logic could be handled by the main MCU, it is safer to have it handled by a separate standalone module. This benefit is easily seen through an example: if a bug in the main firmware causes the CPU to crash, the protection circuit would for the battery would still be functioning and allow for safe storage and charging of the gun. The TP4056 also implements a NTC based thermistor for limiting charging current in the event that the lithium ion battery becomes warm during charging. We do not expect this to be a problem as the charging current is minimal in comparison to the cell's capacity and size. The implementation we plan to use comes in a premade package that has several convenient features already completed for us. These features are a pre soldered micro USB port, multiple LEDs for representing charging status, and two MOSFETs that help protect from overvoltage, overcurrent and undervoltage. This is important it provides a layer of security for the battery's operation. In the event that the main PCB short circuits, the battery will not see a high current draw which could start a fire or explosion. Furthermore, in the event that the battery is completely drained, the battery will be safely cutoff from the circuit to prevent damage to the lithium ion battery. These safety trips allow for safe operation, and once triggered, remain active until the power cable is plugged in. The application circuit in the PCB for the TP4056 is shown below in Figure 17.

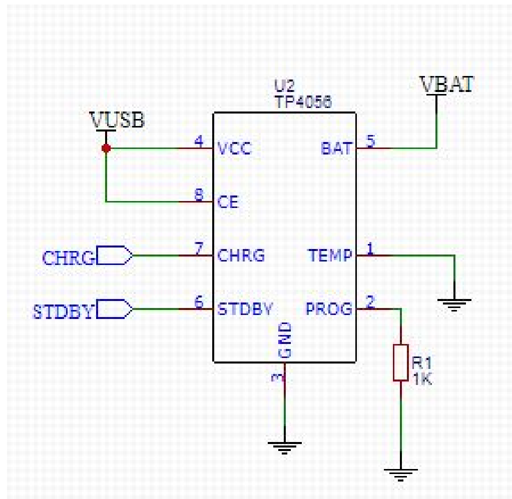

Figure 17: TP4056 Application Circuit From PCB

An additional battery monitor will be used to monitor battery life while the "gun" is not charging. A MOSFET will turn on if the battery voltage becomes too low and another will turn on if the battery voltage becomes too high to prevent the battery from over discharging or over charging, respectively. The chosen battery monitor is the DW10A. The application circuit for the DW10A in the PCB is shown below in Figure 18.

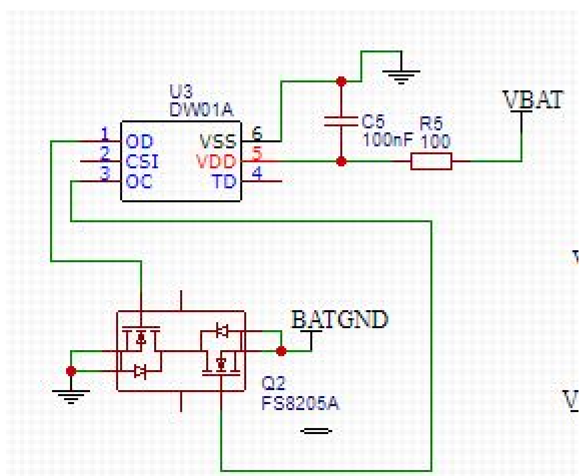

Figure 18: DW10A Application Circuit From PCB

5.1.4 Vibration Motor

For the project, there will be a vibration motor to give the user feedback when either the player is hit or when the player is pressing the trigger. The vibration could also be used when the user is reloading. In all of these situations the player would be more immersed in the experience of the game. If there were no vibration the game could be played but it would not be as immediate to notice if a player is hit or if the player is reloading or fired a shot. Instead of getting immediate feedback, the player would have to check the screen and see if there is a change in their health or in how many shots have been fired. Another application of the vibration motor would be to signal the ending of a game or the beginning of the game. The vibration motor will need to be mounted on the PCB.

There are two different types of vibration motors. One is an eccentric rotating mass (ERM) and the other one is a linear resonant actuator (LRA). The eccentric rotating mass works by having an unbalanced weight attached to the motor shaft. The spinning of the unbalanced weight causes a shaking feel for whatever the ERM is attached to. The ERM uses DC power to drive the motor. There is a subtype of the ERM called the coin vibration motor. It is an ERM that is all encased in a housing and looks flat like a pancake. The coin motor is smaller than the ERM but is weaker due to having a smaller rotating mass inside. The coin motor is much easier to mount and often comes with adhesive on the back. The linear resonant actuator has a magnetic weight attached to a spring that is driven by a voice coil. The LRA looks like the coin vibrator but works differently. LRAs have a quicker response and are more efficient. They need AC in order to work.

The vibration motor chosen to be implemented in this project is a micro flat coin vibrator designed for mobile cell phones shown in Figure 19. It requires 3V/0.05A, which can be powered in our current system, and it produces a powerful vibration that could be easily felt by the user. One advantage of using this motor is its small size. As described in the name, it is flat and only the size of a dime which means it will be a good fit for the small PCB that has been developed for this project. It is also relatively inexpensive with a ten pack costing less than ten dollars.

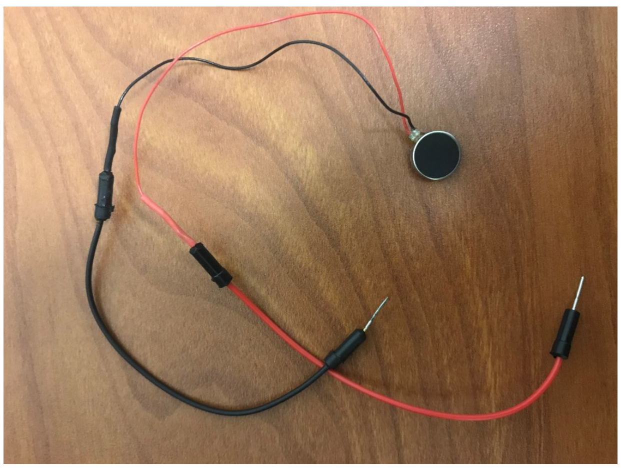

Figure 19: Vibration Motor *Photo Taken By Shannon Fies*

The vibration motor will be able to vibrate with different intensities. This can be done by changing the voltage going to the motor and also the frequency at which the motor rotates. This allows for being able to set different strengths of the motor that is able to be set by the player or for different situations. When starting a game a small vibration paired with other feedback components can effectively let the user know that a game is starting and to be ready to play. At the end of the game a vibration to let the user know that they are out will help them from being confused as to why they cannot shoot other players. Other feedback components will be helping to create the same effect but which one works best depends ultimately on the user.

Flyback Diode: The vibration motor will be switching on and off and will cause an abrupt drop in current. Since the motor is an inductive load, this causes for a large spike in reverse voltage cause by the inductors. The inductors induce the reverse voltage from the drop in current and if there were no flyback voltage the reverse voltage would go directly to the transistor. In order to protect the switching transistor from the voltage spike a flyback diode can be placed across the motor to prevent the large reverse voltage from going to the transistor and burning it up. Instead the voltage will be dissipated as heat through the windings of the motor. Without the flyback diode, the switching transistor could possibly burn up and cause it to be permanently stuck on or off, which would render the gun useless. The application circuit of the vibration motor from the PCB is shown below in Figure 20.

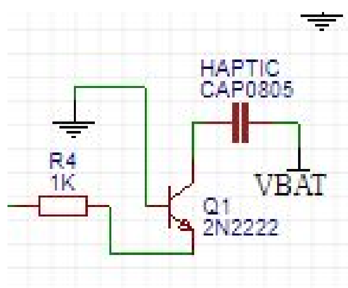

Figure 20: Vibration Motor Application Circuit From PCB

5.1.5 Trigger

For the project, there will be a trigger mechanism for the user to fire at other players. The mechanism will consist of a physical 3D printed part that is a part of the housing and rotates on a pivot to press against a button when the trigger is pulled. The 3D part will be printed in place when the housing is printed and will be given a small amount of space between the trigger and the housing in order to allow for movement along an axis. In order for the trigger to feel like a proper trigger the button used needs to have a tactile feedback to let the user know when it has been pressed in and released.

The button used for the trigger will be a momentary tactile button. This allows for the user to clearly feel and hear when the trigger has been pressed. Some buttons “squish” when pressed and do not give any feedback to the user which leads to a player not knowing if he or she has pressed the trigger. The button will be encased by the housing and will be soldered on the printed circuit board. The trigger will rotate and will convert a horizontal motion into a vertical motion in order to press the button.

The button will need to be debounced in order to prevent the button from causing multiple button presses when the button is only pressed once. If this issue is not addressed early on it will lead to multiple problems when using the trigger and may lead to additional problems. This can be fixed with either a hardware or software debouncing scheme.

A hardware approach to debouncing would involve a low pass filter that is made up of a resistor and a capacitor. This will allow for the press of the trigger to be gradually turned on rather than bounce from on to off and back a few times. This approach would require more components and could possibly lead to a minor loss in power through the extra components. The main issue with this possible debouncing scheme is that the extra components could lead to more hardware that is needed and will add a little more complexity to the design. Another option is debouncing through software.

A software approach to debouncing would involve adding some code to our project that takes care of the situation. The code would check for a change in the state of the button and usually would wait a small amount of time to allow for the button state to settle. Instead of using a delay or wait statement the code will have a task scheduler that will schedule the check of the state of the button for about fifty milliseconds after the initial change in state is detected.

Using a task scheduler allows for the program to do another task during the fifty milliseconds rather than wasting that time halting all processes. Once the other task that was scheduled for the delay time is finished the state of the button will be checked again in order to see if the button is still on. If the button indeed did change states from off to on or on to off then we go ahead and use that input. The finalized input from the button will be used to allow for the transmitting of an infrared message or if the button is held down this can start the wifi setup.

The software approach is going to be chosen instead of the hardware approach. One of the reasons is that the hardware approach adds more components to the system. One of these components, the capacitor, delays the input time from the button being pressed to being able to use the input which may cause a longer time between the press of the trigger to the firing of an infrared message or setting up of the wifi. Software instead can be used to keep the component count lower and any changes are done with code rather than having to change physical components or pick resistors and capacitors again. The application of the trigger mechanism from the PCB is shown below in Figure 21.

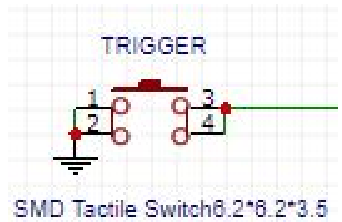

Figure 21: Application Circuit of Trigger Mechanism from PCB

5.1.6 LEDs

For the project, there will be color LEDs to differentiate which team each player is on. These LEDs have to be bright and use low power. The LEDs also have to be able to change colors in order to be able to change teams. In order to have color changing LEDs we can use an RGB LED where a red, green, and blue LED are used to make different color combinations. In order for the housing to be brightly illuminated the project will have five of the RGB LEDs. Since some of the housing is a material that is clear it will be able to easily show what team each player is on.

The LEDs will also be used in order to show a little animation or to better inform the player of what is happening. When a player is hit with a shot from a second player, the LEDs will flash to indicate to the first player that he or she was indeed hit. This also lets the second player, the one who fired the shot, that the first player was hit. The animation can be used when pairing the gun to a phone for setting up the game. This animation will be similar to how headphones and other bluetooth devices have a small LED that flashes or fades on and off. Either of these two options help the user know that the gun is ready for being set up for a game. Another animation that can be used is to blink one of the LEDs yellow or red depending on how the battery level is at.

The RGB LEDs that are going to be used are the WS2812s. This package has a 5050 RGB LED component with a WS2811 control circuit built in. The 5050 part means that the size of the package is 5 mm by 5 mm. The package is very small since it is all within the 5050 component. The LEDs are powered with 5 volts and then the data is done through NZR communication. NZR stands for non-return-to-zero. This means that for a 0 bit it’s a positive voltage and for a 0 bit a negative voltage is used. The data is sent in groups of 24 bits.

Each time a WS2812 is set to a specific color it is done with 24 bits. 8 bits are used for each of the different color LEDs. 8 for the green LED, 8 for the red LED, and 8 for the blue LED. As seen in Table 8, this allows for 16,777,216 different color combinations. Once one of the WS2812s are set to a color the data is sent out the data out pin and that is connected to the data in for the next WS2812. The remaining WS2812s are connected in the same way and the last WS2812’s data out is not connected to anything. All of the WS2812s are able to be programmed with a single GPIO pin from the microcontroller since the data is daisy chained through each other. Table 9 below shows the power and output specifications of the WS2812.

Specification	Value
Color Combinations	16,777,216
Supply Voltage	5 V
Data Rate	800 Kbps
Transmission Delay	300 ns

Table 9: Specifications of WS2812

The application circuit of the LED array from the PCB is shown below in Figure 22. This was the design that was ultimately implemented on the team's PCB final design. The layout of the circuit was derived from the recommended schematic that was offered in the WS2812B's datasheet.

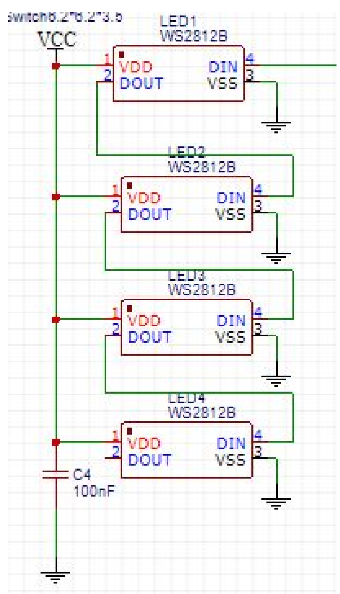

Figure 22: LED Array Application Circuit From PCB

5.1.7 Display

The majority of the user interface will be implemented on the display which can be used to show a wide variety of data. This allows players to get notifications about in game events and interact with menus for an easier setup. We had several options to choose from for the display at different costs with different advantages and disadvantages. Below, in Figure 10, contains a summary of the display research that compares and contrasts three types of display screens: Organic LED, TFT, and E-ink.

Display	Organic LED	TFT	E-ink
Price:	$9	$13	$22
Pros:	Fairly power efficient and cheapest option. Small form factor available which is crucial to embedding the display within the gun.	Supports full color, and some even support a touchscreen interface.	Most power efficient option. Has the ability to preserve contents even after power off.
Cons:	Only in black and white.	Not as power efficient. Too bulky to implement into a handheld gun.	Only in black and white, slow refresh rate. Expensive.

Table 10: Display Comparison

The display we chose to use is an OLED display that uses I2C to communicate with the master microcontroller. We had the option of choosing one that supports SPI but the two wire interface is simpler and we already have a bus available for it. The display is a supports 1 bit black and white color, which is partially limiting, but will not be a dealbreaker. The small form factor of the display allows it to be embedded neatly into the 3D printed housing.

The display operation is very simple, over I2C. The display is pinged at the address of 0x3D and the entire buffer of pixels is pushed to the display. The buffer will store 1 bit for each pixel, which on the display is 128x64 pixels. This comes out to a buffer size of 128*64 bits * 1 byte / 8 bits = 1024 bytes. This is not a problem for the ESP32 as it has 328k of usable DRAM, which means there is more than enough left over to be used for the rest of the program.

The display implementation will be split up by an object oriented approach. Each screen will be a class that can be instantiated. This is helpful as each screen has similar data points that can be reused. Each instance of the screen will also include pointers to subroutines that will be run when certain hardware events happen, such as when the trigger is pressed or the IR receiver receives a transmission. The organization our display will follow the following schema:

On bootup there will first be a splash screen image that indicates that the gun is booting and that the display is working correctly. Then the unit tests on each component of the gun will run and the results of each test will be displayed on the screen. Once the unit tests are complete the setup screen will be displayed. The setup screen will display the information needed to connect to the gun, including a randomly generated pin needed to connect to the smartphone. Once the gun is setup the display will move to the game screen.

The game screen will consist of pertinent in-game information. This information would include the remaining time left in the game, the remaining health and ammo of the player, as well as notifications from in game events such as team victories. Each screen will also include a small battery life indicator in the top right corner. This screen will need to be updated efficiently as to provide low latency updates with game events. The ability to show popups on game screens will also be implemented, alerting the user of game events such as being shot or needing to reload. The application circuit of the OLED from the PCB design is shown below in Figure 23.

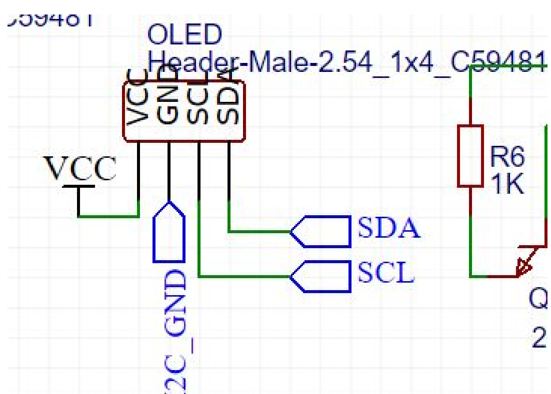

Figure 23: OLED Application Circuit From PCB

5.1.8 Microcontroller vs FPGA

A microcontroller unit (MCU) typically consists of at least one processing core, CPU, along with other key components such as memory and input/output peripherals. Some may have a small amount of RAM built in on the chip. Some microcontrollers run at low frequencies in order to save power but if more performance is desired a microcontroller that runs at a higher frequency and consumes more power is needed. In order to program a microcontroller a high level language such as C or C++ can be used and then a compiler and assembler are used to convert the program to machine code in order to be able to store the program on the microcontroller's memory. The program can usually be changed by erasing what is on the memory and writing to the flash or EEPROM depending on which reprogrammable storage the microcontroller uses. MCUs run sequentially, one task at a time and then move on to the next task.

For the project, the MCU will need to be able to take care of all the input and output signals. These include the encoding for the infrared beam when the trigger is pressed and flashing the LEDs. The infrared beam is then decoded by another gun and then the signal needs to be processed to see how much “damage” the player received. The LEDs will need to flash when the player is hit with a beam. The damage value is then displayed on the screen. The MCU will also have to take care of making the haptic feedback vibrate when shooting and when being shot. These processes will all have to be taken care of and need to run smoothly. In order for this to occur, careful task scheduling will need to be used to keep the different processes from looking jumpy or happening too slow.

A field-programmable gate array (FPGA) is made up of an array of programmable logic blocks that can be programed to be used as different logic gates. The gates are used together with other components that may be found on some FPGAs which include input/output peripherals, memory blocks, and even DSP blocks. FPGAs are programmed using hardware description languages such as Verilog and VHDL. Once the program is written, the code goes through synthesis, placement, and routing. Synthesis is a process that takes the HDL code and converts it into logic gates. Placement determines where to place the logic gates on the FPGA. Routing will define the wires that connect the necessary logic blocks to one another. There are many ways in order to make a program faster such as making critical routing paths as short as possible. FPGAs run processes in parallel.

For the project, the FPGA will need to take care of all the inputs and outputs. Since the FPGA can run different processes in parallel, each of the different processes will be able to run very smoothly without any delay from waiting for other processes. The processes include the infrared encoding and decoding, the buzzer, the haptic feedback when firing and getting shot, the screen updating with damage and other information, taking input from the trigger and the accelerometer, and turning the LEDs on and off and setting them to different colors if needed. Since the FPGA runs the processes in parallel a task scheduler is not needed, but careful programming to make sure all of the signals are taken care of and no loops or writing to the same registers at the same time occurs.

There are many differences between the MCU and the FPGA. The programming tools for each vary drastically. For the MCU, there are a lot of options to choose from. There are many libraries that have pre-made made code in order to test different components and make sure they work well on their own. Each time the code is ready to be written to the MCU the compiler has to translate the high level program to a lower level language such as machine code. This process is very quick. For the FPGA, there is not much to choose from. Most FPGAs have a dedicated software that needs to be used that is sold by the manufacturer. From there the process is much longer than for the MCU. Each time a different iteration of the code is written to the FPGA the software has to go through the lengthy process of placing and routing the different logic blocks within the array that is on the chip.

For the project, if the fastest possible chip was to be used the FPGA would be developed on and then changed for an ASIC (Application-Specific Integrated Circuit). The ASIC is basically an FPGA but it is programmed only once. ASICs are much faster and efficient than an FPGA since the one application can be developed thoroughly on an FPGA and then made into an ASIC. Once the design is made on the ASIC, there is no way to change the design. If there has to be a change, the whole process has to start over and work on improving the design from the FPGA stage. For the project, we will likely be changing and redesigning processes often which will not allow for use of an ASIC. If there were years worth of time to work on the project an FPGA and converting it to an ASIC would be very beneficial. Once an ASIC design is completed it is easy and cheap to fabricate numerous copies of the design to mass produce the product. The main issue is the design step takes much longer than an MCU. A MCU will allow for numerous redesigns and trial of different ideas for the project. Although a MCU is slower it is much easier to develop and design on which will be needed in order to design the system in a small amount of time compared to the FPGA or ASIC design time. A summary and comparison of FPGA vs MCU is shown below in Table 11.

	MCU	FPGA
Programming Learning Curve	Some learning curve to understand interrupts and threads among other concepts	Large learning curve to understand hardware description language
Flexibility	Software reprogrammable	Software and hardware reprogrammable
Development Length	Easily can recompile software or add a patch which is fairly quick	Have to place and route hardware all over again which take much longer
Speed	80/240 MHz	~100 MHz
Price	$5	$15

Table 11: MCU vs FPGA Comparison

5.1.9 Microcontroller Selection

The project will be based around a single microcontroller that can coordinate and interact with all the sensors and input/output devices on board the PCB. There are several options to use for microcontrollers for this project, but each of them have different pros and cons that can affect the development and the success of this project.

MCU:	ATMega328p (Arduino Nano)	ESP32	MSP430G2553
Speed	16MHz	80/240 MHz	16MHz
RAM	2kb	328kb	512 bytes
Cost	$2	$5	$1.98
Pros	Cheapest module of all options, and has the most amount of open source support for libraries.	In built wifi and bluetooth controller. Significant amount of available RAM and Flash memory. Fastest CPU speed of all MCUs.	Cheapest option, lowest power usage amongst all the MCUs
Cons	Need to have seperate wifi module. Limited RAM and CPU clock speed. Limited flash size also can be a problem for larger programs.	Most expensive option. Newest MCU so limited online documentation. Highest power draw since its the fastest clock speed.	Poorly written software development environment with limited online support. Need to have seperate wifi module. Limited RAM and CPU clock speed.

Table 12: Microcontroller Comparison

We chose to go with the ESP32 due to several factors: The higher cpu speed will be crucial to a smoothly running user interface for a firmware that has several background components. The inbuilt wifi radio also simplifies the project architecture and reduces the complexity of the PCB design. The fact that the ESP32 can be programmed using the Arduino IDE also simplifies the project, since pre-existing open source libraries can be used to integrate various sensors and input/output devices into software. Although the ESP32 is the most expensive option, its inclusion of the wifi radio reduces the overall cost of the project. Although it is not planned to be implemented in this project, the onboard BLE radio can be leveraged for future improvements to the project.

The ESP32 also has several other inbuilt features that make it a more attractive option. The ESP32 includes an onboard DAC which can be used to play accurate sounds on the buzzer for audio feedback. The ESP32 also has an internal encoder for infrared signals with implements the modulation through hardware, which frees up the CPU to handle other tasks in the meantime.

5.1.10 Audio Hardware Design

A major part of the user interface will be the audio feedback throughout game play. This audio feedback will play many simple tones and tunes at certain events throughout the game. Implementing this will require a speaker of some sort to play audio content. There are a few options to configure this setup that need to be considered.

The easiest, most simple method is to add a simple buzzer to the project. The buzzer takes in a simple DC voltage and has internal mechanical structures to make it automatically resonate at a set frequency. This is the classic method used to implement alarms for security systems and smoke detectors. Implementing this feature would require only a switching transistor such as a N-channel or P-channel MOSFET or BJT to handle switching the DC voltage into the buzzer. Since the buzzer is an inductive load, a flyback diode would also be necessary. This is to prevent voltage spikes that occur when powering off the buzzer from damaging the switching transistor. Damage to the transistor could cause the buzzer to remain stuck on which would be annoying to the user and ruin the user experience. The main disadvantages of the buzzer is the lack of customizability. The buzzer only supports one tone and cannot change volume levels, thus it would be a poor option for an audio feedback device.

The other option is to use a speaker. A speaker is a much simpler piece of hardware, but requires more finesse to make it operate. The speaker takes in AC voltage to produce sounds of high quality. This requires a digital to analog converter to produce various voltage levels to produce different sound levels and volumes. Luckily, the ESP32 already has two onboard digital to analog converter channels that can be used for this exact purpose. This allows the project to play specific sound samples, including voice recordings, music or audio samples. For example, during shooting, futuristic blaster noises can be created through the onboard speaker. One major limitation is the audio samples require large amounts of data to be stored for playback, and the ESP32 only has 3 megabytes of flash file storage, some of which is already allocated towards other features such as WiFi updating of firmware. This means that most audio samples will have to be low fidelity and short in total length. This challenge should not be entirely limiting as the amount of audio feedback will already be limited to make sure it is used tastefully without annoying the users.

Magnet driven speaker: The most common option for speakers is to use a magnet driven speaker. This features a moving membrane with a magnet a coil assembly that vibrates to create sound. This option can create more accurate, high quality audio feedback, but has two major disadvantages. The moving assembly can be quite delicate, and is especially susceptible to damage from impact, water and debris, which is not ideal for an outdoor play device. The other is that most speakers have a large footprint, which is not ideal for this compact solution.

Piezo Speaker: The alternative to a magnet driven speaker is a Piezo speaker. A piezo speaker is more power efficient solution, that is able to produce louder sounds with less power. The piezo speaker is closer to a solid state solution, that is more robust from impacts and environmental damage. Piezo speakers also have a much smaller footprint, which allows them to easily be mounted directly on the printed circuit board, which cuts down on product complexity and overall manufacturing time of the gun. Directly mounting onto the PCB also improves robustness of the project.

Amplifier: It is common to pair speakers with an amplifier circuit, since the power output of the onboard digital to analog converter is considered to be signal-level only, which is too low for most applications. An amplifier circuit of various qualities can be included using operational amplifiers to boost output power and allow for louder audio feedback. However, since our use case requires the user to hear the audio from less than 3 feet away, the inclusion of an audio amplifier seems unnecessary, especially when using an already efficient piezo speaker. The inclusion would increase the cost of the gun, as well as increase the complexity of the system being implemented on the printed circuit board. The amplifier circuit also has a high chance of receiving interference from the onboard bluetooth and WiFi circuitry, which would cause undesirable humming on the speaker. The power usage of amplifier circuits would also be a major concern, as the inclusion of an amplifier would dramatically increase current consumption during gameplay and have a slight impact on quiescent current draw during standby mode as well. The application circuit of the speaker from the PCB is shown below in Figure 24.

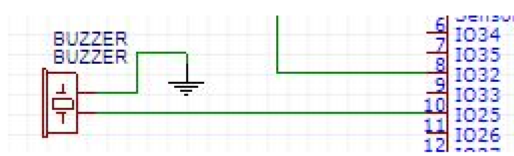

Figure 24: Speaker Application Circuit From PCB

5.1.11 Peripheral Enable Line

Through empirical testing it was found that the OLED displays and accelerometers available on the market for this project all have a common issue of having a high quiescent current. This is unacceptable as it significantly increases the amount of current drawn during sleep mode. The solution to this problem is to include some sort of switched power bus for these peripheral devices that can be controlled by the microcontroller. During regular gameplay the enable line will be switched on to allow for normal usage of the display and accelerometer. Right before the gun is put into standby mode, the microcontroller will pull the enable line low to disable all of the peripherals and essentially bring their quiescent draw to 0 microamps. This can be achieved using a simple P-channel MOSFET or PNP BJT that operates as a switch between the battery voltage potential and the VCC rail of the switched peripherals. The direct battery voltage is used since the accelerometer and display both have onboard regulators, which will help alleviate current draw from the MCU’s linear regulator. A P-Channel MOSFET will be used due to its higher efficiency, with a pull up resistor on the gate to ensure that the peripheral line is never enabled by default. A N-Channel MOSFET or NPN transistor is not a viable solution here since it is very common for disabled GPIO pins to be set to a logic low, providing a possible grounding path for the peripherals. Not only will this result in higher current consumption during standby mode, it will also possible damage the microcontroller. Most microcontrollers have a current limit of approximately 15 milliamps that cannot be exceeded for a continuous damage without possible damage to the GPIO pin or microcontroller.

5.2 Firmware design

One of the biggest parts of this project will be the firmware design. The firmware will tie in all of the individual components in the system and manage interactions between them to create an immersive digital experience for the user. The firmware will be a complex beast, as it will be handing a dozen different processes and peripherals at the same time, so careful planning is an important step to make in this project. Without making careful decisions beforehand, the software development portion of this project can spiral out of control and create massive delays in the completion time.

5.2.1 IDE Software Options

Espressif IDE: Uses C++ to implement all functions on a low level, with strong control over every component within the ESP32. Developing in this environment will allow very precise control over each subsystem of the ESP, however it requires a significant amount of development time to implement each sub-feature needed for the project. Furthermore, the Espressif IDE is fairly new compared to other options, which means that it is less polished and has less support available from open source communities.

Espressif Lua: A simplified programming language supported on the ESP32. Although Lua is lightweight, it is loosely typed and formatted. This leaves the language to be more error prone and more difficult to maintain. The simplifications in Lua make optimizations harder to make, whereas the team's expertise in C/C++ programming allow for efficient software development.

Arduino IDE: The arduino framework also uses C++ but natively implements several features to the microcontroller that make using it easier. For example, managing general purpose IO ports is broken out into functions, instead of having to directly manipulate each port using bitmasks. Furthermore the opensource community develops several libraries that implement functionality for various sensors and displays, which we can utilize to our advantage. This cuts down on development time and makes sure our time is better spent on testing and validation.

For the software design, we plan to implement the entire program using industry adopted coding standards for C++, which allows for easier collaboration on the project and makes the code base more maintainable. For a complex project such as this one, this is crucial as fatal bugs are not an option. Since the gun firmware will not be able to be updated after release, all bugs need to be addressed before the completion of the project.

5.2.2 Multitasking Implementation

One of the major challenges of implementing the software for the laser tag gun will be making sure the code runs quickly and smoothly, thus presenting a smooth user interface that is pleasant to interact with. The user interface subroutines will have to run smoothly alongside several other tasks, such as polling the trigger, reading data from sensors, addressing incoming wifi data and updating the display. The program will implement several tasks that occur concurrently, however they will all be running on a single thread which makes it a challenge to implement seamlessly.

The solution to this problem will be to implement a task scheduling algorithm that manages when and how often tasks will be run. It also provides an alternative to time delays needed in the code, by allowing the program to instead run other tasks in the meantime. This method also allows for more control in the allocation of processor times, as certain tasks that are less important can be scheduled to repeat less often than other tasks that are more time sensitive. The alternative to this approach is to use timers to automatically call interrupt service routines. However, timers are limited on any microcontroller, and their precise timing abilities are needed for other parts of the project, including the timing necessary for infrared modulation.

An example application of the task scheduler: The common process to read the state change of a button is to read the state, and if the state has changed, stall for small amount of time and read the button again. This is done to avoid debouncing errors. Our method would schedule a task to run after small amount of time, instead of just stalling the processor, and then continue on with other tasks that need to be run. Scheduling a task will be done by calling the scheduler with a specified delay in milliseconds and passing a callback function that will handle the task that needs to be run.

Another major benefit of implementing the firmware this was is that it also saves power, since the ESP32 can be put into sleep mode temporarily at times that the scheduler has zero tasks to run. This results in lower overall power usage and an increase in playable game time. Implementing this feature is quite simple, as at any given point, it is easy to determine exactly how many milliseconds are remaining before the next task needs to be run. Once the number is determined,

The implementation of the task scheduler will need a backing data structure that holds all the currently scheduled tasks to be run. This data structure will be added and deleted from continuously and will be polled with every loop of the main game code, which means that access to this data structure should have minimal overhead in number of operations needed to access it. The data structure will essentially be a priority queue. A priority queue is a form of a queue that includes an ordering to the inserted elements. The priority value for our implementation will be the time at which the task needs to be run. Tasks that need to be run at a sooner time will come first in the data structure. The priority queue must support the insertion of elements in order as well as peeking at the highest priority element in the queue. If the highest priority element is not ready to be run, then it is guaranteed that none of the tasks scheduled are ready to run, since time is monotonic. If the highest priority element is ready to be run, it can be deleted from the data structure and run. After that the queue can be checked again for any other tasks that also need to be run. The possibilities for the backing data structure has a few different options, including simple arrays, linked lists or min-heaps.

When comparing data structures, it is typical to compare the runtime order using Big-Oh notation. This notation describes how the runtime grows with a growing number of elements. For example, runtime that grows linearly with the number of elements is described as having an O(n) runtime. For our application since we know that the actual number of elements will be relatively low, it is more important to consider actual number operations rather than just the runtime order.

The simplest option for the tasks scheduler is a regular array, supported natively in C and C++. An array simply allocates a section of contiguous memory for all possible elements Using an array for the task scheduler requires all memory for the maximum number of supported tasks to be preallocated. That means a majority of space is wasted by the data structure even when the number of scheduled tasks is low. However, when the number of scheduled tasks is high, the usage of space is more efficient as arrays require zero overhead for each element, as they are accessed directly by their memory address. In terms of memory operations, the array requires an O(n) shifting of all elements whenever an item is inserted or deleted. This can cause massive amounts of operations that may slow down the operation of the task scheduler.

The second option to use for the task scheduler is to use a linked list. A linked list is a chain of nodes where each node contains the relevant data and a pointer to the memory address of the next item in the list. This chaining of data means that each node can be separately allocated without needing to pre-allocate all the memory needed. It also means that memory is freed as tasks are deleted, allowing other sections of the program to benefit from it. Using a linked list structure that allows easy dynamic insertion and deletion of elements without requiring any shifting of remaining elements in a backing array. Both insertion (adding a new task to the scheduler) and deletion (completing a task the is ready to run) will run in O(n) time as the spot to insert the element will need to be found first.

The most common implementation for a priority queue is a min-heap which has O(log n) insertion and O(log n) deletion. A min-heap works by using a regular backing array but accessing it with finesse. Items are placed in a tree style division based on the priority value. Although this has the best runtime order, the runtime benefit will be minimal to none, as the actual number of scheduled tasks will be low. The runtime benefits of minheaps is only evident when handling a large numbers of items. In fact the overhead of inserting nodes in the correct subtree or percolating nodes to the top of the minheap at deletion can results in more operations than the simple linked list.

Out of all of these implementations, it is clear that the linked list style implementation will be ideal for the task scheduler. The scheduler will also include pointers to the head and tail nodes of the linked list which will allow many operations to the list to possibly devolve to O(1) runtime, such as when adding a new task, since most likely it will be scheduled to run after all other tasks.

5.2.3 Infrared Communication

We plan on using infrared for the optics component of the “laser” tag gun. Infrared light is emitted at 600 to 3000 nanometers which is undetectable by the human eye. Although infrared light is very common outdoors, it can be modulated at certain frequencies which allow it to be distinguished from ambient infrared light. Its wavelength is longer than most visible colors in the spectrum, which means it can travel longer distances. The sun itself emits ambient infrared light, as does fire such as candles. This may cause some interference during outdoor play.

The infrared light will be emitted using an infrared LED encased in a tube used to direct the light into a narrow beam. The light will be modulated at 38 Kilohertz and use the NEC infrared encoding protocol to transmit the encoded signal. A lens at the end of the tube may be needed to increase the range of the gun as well as help increase reliability during outdoor play.

The transmitted signal, if directed towards a target gun, will be picked up by the onboard IR receiver. IR Receivers that internally demodulate 38KHz signals are readily available on the market. By internally demodulating the data, the microcontroller is freed from having to handle all of this precise timing on its own. The NEC encoded signal can then be decoded and split into usable chunks of data. NEC encoding works by having distinct time delays for the on and off time for 0 and 1 bits of data as it is sent over. The receiving device then measures the incoming high and low times and uses that to determine whether the bit received was 1 or 0.

5.2.4 Infrared Encoding

Each infrared signal will contain a complete packet of data to be read by the victim-gun for decoding and the appropriate game logic response. This requires the packet of information to have a standardized format. The packet will have 32 bits of information and will be arranged in the following fashion:

The first 8 bits will contain the packet's magic number, which is a constant number that is known by each gun. This will be consistent for each type of signal being sent. For a regular gunshot, the bullet packet will include the number 0xF4. This number itself is arbitrary, but it is used for the victim gun to understand how to interpret the rest of the information, as well as understand that this packet is not garbage data or malformed data. Other types of signals can be sent, and added to the system, including upgrades and health pickups, of which each will have its own magic number.

The next 8 bits will have the playerID, which includes the numerical index of the player from which the bullet originates. This gives information as to which team to award points to when processing the bullet. With 8 bits of data, 256 players can theoretically play together.

The next 8 bits will include the amount of damage done by the bullet. This number can change based on power ups and game settings, so it is important to include this information with each bullet. The 8 bits allows damage values from 0 to 255. Each gun is assume to record the amount of damage done to it based on the honor system. Although this solution is not ideal and fool proof, it is similar to how modern laser tag facilities operate today, and it is a similar system to how paintball and airsoft games are played.

The last 8 bits is the checksum. This value is used to make sure all the previous bits are correctly set. The checksum value is included in the packet by using the exclusive-or operator on all the previous 3 bytes (i.e bits 0-7, 8-15, and 16-23). This value is then calculated separately by the victim gun and compared to the included checksum. This allows for basic verification that all the bits included in the packet were sent from another laser tag gun, and not interference from other sources, such as remote controls for televisions. This also ensures that the packet's data does not have malformed data within it, such as an incorrectly received player ID.

What was programmed to be sent by the IR LED is shown below in Figure 25. Each gun will have an unique identifier with its own set of values outline by the parameters below.

Bullet encoding Layout: 32 bits of data

| 4bits |***| 4 bits |***| 8 bits |***| 8 bits |***| 8 bits |

Prefix **team ID** **player ID** **damage value** **XOR checksum**

Figure 25: Bullet Encoding Layout

5.2.5 Wifi Mesh Networking

The guns will all communicate using a mesh network topology, which is a network topology that does not have a single master connection as is common in the star topology. Each node in the topology is used to transmit messages and expand the network. By using a mesh network topology, each gun can transmit a message to any other gun, (or every gun within the network) as long as there is an intermediate path between them. That means that even if two guns are out of range of each other for their respective wifi radios to communicate, they can still deliver and receive messages by handing off packets to neighbors that are within range of the network. This is crucial for our game as it is meant to be played in large fields or areas where the normal range of the wifi radios will be exceeded.

The network topology will be a self healing network. This means that each intermediate connection between each node will be dynamically adjusted based on the proximity of each gun. If a gun moves out of range of one node and in range of another, the network will automatically detect that and reconfigure itself to adjust for it. This is all done continuously and automatically without any need for player intervention. Each packet of data has a unique identifier to it which helps each node identify if a packet has been lost or if a packet has already been seen before.

5.2.6 User Interface

A large part of the success of this project is to make sure the user experience while interacting with the gun is smooth and seamless. The majority of the user interface is taken care of by the display during game play. The display can show relevant information quickly to the user, which is pointing towards the user at all times. Buttons on the gun provide a method of getting user input into the gun for managing menus. However using buttons can be slow and non intuitive, so we would like to minimize the amount of interaction done through there. We hope to use the button mostly for waking the gun up from sleep mode and enabling the wifi based setup.

The ESP32's wifi capabilities can be used to run an HTTP server. An HTTP server is a web server that responds to GET and POST requests from an outside source that is connected on the same network by making the request to the appropriate IP address and port. Web browsers make GET or POST requests to HTTP servers to retrieve a .html file that describes what the page looks like. The html file includes information such as the text on the webpage, the buttons and input fields, background color and more. Javascript is a programming language that can be embedded into the webpage to make the page interactive.

All of these features can be used on the ESP32 to make a locally accessible webpage that the user can access through a smartphone in order to aid with setup. The javascript page can be made to have text fields, interactive buttons and toggle switches in order to input any information needed to set up the game.

Using the principle of AJAX, the webpage can continuously update with new information. AJAX is a principle of javascript coding where continuous requests are made to the http server to retrieve new information and dynamically update the webpage. For example, our web page can have a list of players that are locally joined and are ready to begin the game. As more players join, the page can update to show their respective names and status.

To be able to connect to the HTTP server, the ESP32 will have to be put into AP mode, which turns the wifi radio into an access point that any smartphone can connect to. This will be only temporary for the user to connect and setup the game.

Another aspect of the user interface will be the accelerometer. The accelerometer will have to major functions. The first one will be to check the orientation of the gun. We plan to implement a reload feature, where pointing the gun down at the ground for some number of seconds is used to reload the gun. The other major feature will be to implement a way to check for user activity. If there is no measured movement, then the gun can be put into sleep mode to conserve power. There are a few different options to use for the accelerometer as shown below in Table 13.

Accelerometer	MPU-6050	ADA2809	ADXL337
Cost	$5	$7	$19
Pros	Cheapest option that communicates over I2C. Includes gyro as well.	Communicates over SPI interface which is faster than I2C	High refresh rate of 1600hz
Cons	Requires several extra passive components for basic use.	More expensive. Bulkier chipset due to onboard ADC converters	Analog readings which need requires additional ADC pins and takes longer to read.

Table 12: Accelerometer Comparison

We chose to go with the MPU-6050 due to its low cost and digital access features. There exist several libraries for the Arduino platform that already implement functionality for this accelerometer. The board includes a 3 axis gyroscope as well but currently we have no plans to use it for any of the readings. The application circuit for the accelerometer is shown below in Figure 26.

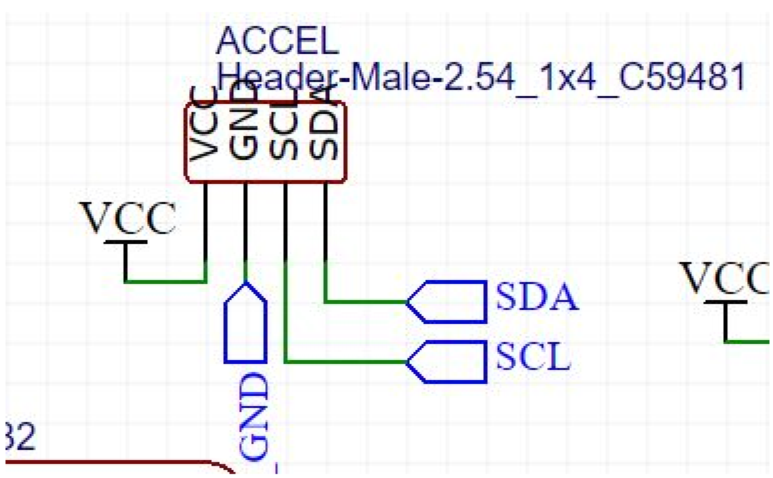

Figure 26: Accelerometer Application Circuit From PCB

5.2.7 Display Writing Algorithm

Through testing of open source libraries that are available, it was found that most libraries approach writing to the display in simple, inefficient manner. Most libraries implement that following structure to handle controlling the display over I2C: First the library stores a 128 x 64 boolean array as a display buffer that holds each pixel's value of either on or off. Then whenever a display write is requested, the entire frame buffer is pushed over I2C to the display. Although this method is simple to implement, it can have significant impact on memory and frame rate. Firstly, although a boolean stores only 1 bit of data, true or false, it still allocates one entire byte of data, which means it uses 8 times the amount of memory. The second inefficiency is that even if very few pixels are updated, all 8,192 (128 * 64) pixels are pushed to the display for drawing. This is terribly inefficient, and the slow transfer rate of I2C makes this even more impactful to the display's frame rate and the operation of other tasks on the processor. This inefficiency is unacceptable as maintaining a smooth user interface is crucial to this project.

Both of these inefficiencies have solutions: The storage of the pixel data onboard the CPU can be optimized by using all 8 bits within a byte and using bitwise operators to access all of them individually. This means that each row of the display will be divided into segment of 8 pixels, each stored within a single byte. To access pixel X on a certain row, first the corresponding block is found by dividing the column index by 8 (or bit shifting the column index to the right by 3 bits, which is the equivalent operation but requires less clock cycles to compute.) The remainder of X divided by 8 (or X ANDed with 0b00000111) determines the pixel within each block to access. The row index remains exactly the same between both implementations and does not require any fancy bitwise operations for access. This method requires minimal overhead for computing each individual pixel but provides incredible memory savings for the microcontroller.

The second inefficiency also has a solution: It is not always necessary to write all pixels to the display when minimal pixels are edited. Instead, it is more efficient to keep track of which pixels are edited, and then only push those pixels to the display to update the internal RAM buffer. This can be done by splitting the display sections into slices that have variable start and end positions. These slices will encapsulate all the edited pixels while excluding a significant amount of unedited pixels. The display will be set to use a limited number of slices, as using too many or too few has introduces inefficiencies. In the simplest case, having only one slice is the same as the base case of pushing the entire frame buffer with each write operation over the I2C bus. Having infinite (128x64 slices) is the same as writing each individual pixel independently. This method is also inefficient, as each write has to its own overhead of specifying which pixel is being written as well as overhead associated with initiating a I2C transmission. Initiating an I2C communication requires a handshake between the master and slave before any data can be passed, which takes time and adds to the overall runtime overhead of the subroutine. The optimal number of slices can be found using empirical testing, once the necessary display designs have been created.

5.2.8 Serial Updating of Firmware

The most common method of burning our custom firmware onto the ESP32 is by the UART Serial interface over USB. The serial interface connects an ESP32 to a computer that can then upload the firmware onto the microcontroller. This process is usually very fast, only taking a few seconds, and does not require a functioning sketch to already be present on the ESP32. First the ESP32 must be reset and put into a special mode by connecting the BOOT pin to ground. Then the serial interface can upload the sketch, bit by bit, to the ESP32. Once it is uploaded, the ESP32 regurgitates that information back to the computer to verify that the sketch has been uploaded correctly. The serial method of uploading sketches has many benefits, including the ability to upload firmware binaries quickly and reliably. This method can also recover microcontrollers that have been previously uploaded with corrupt sketches, which is a possible occurrence. The serial port on the ESP32 is able to be accessed by connecting ground and the relevant rx and tx pins to a USB to Serial converter. No extra hardware is needed on the PCB to make access to the serial port, which makes it a cost effective method of providing updating abilities to the gun. It was decided to not use the USB power of the USB/Serial converter to power the microcontroller as it may interfere with the power supplied by the battery and onboard regulator. Instead the 5 volts from the USB connection are wired to the TP4056 lithium battery charger, which then charges the battery and powers the MCU.

5.2.9 WiFi Based Updating of Firmware

The ESP32 runs using a firmware burned onto the SPIFFS, which stands for SPI Flash File System. Although it is usually updated using a usb serial interface, the SPIFFS can be used to update the firmware on the gun wirelessly. The ESP32 comes with 4 megabytes of flash memory space. By default the ESP32 is divided into a reservation of 1mb of flash memory space and 3mb saved as EEPROM memory used to persist data between resets of the microcontroller. The remaining 3 mb of space can be used to save a copy of the updated firmware.

The firmware can be uploaded as a file through the WiFi capabilities of the ESP32. This can be accomplished by hosting the firmware at a known location on the internet. When the ESP32 is connected to an internet connection, it can ping the update server for the latest firmware version. If the server's returned firmware version is later than the current firmware version, it can be downloaded into the SPIFFS. It is important to make sure that the downloaded binary is not corrupted, as installing a corrupt binary can result in bricking the microcontroller, rendering it unusable. The verification can be done by running an MD5 Hash algorithm on the downloaded file and comparing it with the MD5 hash of the file presented on the server. A hashing algorithm uniquely transforms a file's contents into a string of letters and numbers. Thus, if any of the file's contents become corrupt during the download process, the hash computed will not match the hash on the server, indicating a problem. If this occurs, the process can be repeated and attempt to download the file again. If the file is downloaded successfully, the downloaded binary contents can be copied to the base firmware section of the SPIFFs. The next time the ESP32 is reset, the updated firmware will be run.

5.2.10 Serial Based Debugging Interface

Throughout the development of the program, possible bugs will be an inevitable. Although bugs are unavoidable, they can be handled smartly to lessen the time wasted on them, and make sure they are caught before release. To efficiently find bugs, a serial menu interface will be created to track the operation of the firmware and report over the serial interface to allow for easy monitoring of the software's operation. A common solution to this problem is to have the serial port always blast all the information at all times, however this solution is less than ideal for several reasons. Firstly, blasting all the information over the serial port creates significant CPU overhead that can slow down normal operation of the firmware. Secondly, the software will be multitasking several tasks all at once, which means that the amount of debug information produced will be significant. This means that it is possible to create more debug information than the serial port can handle, since the port is limited to a finite baud rate. The baud rate that will be used is 115200 bits per second, which is a common bit rate supported by most usb to serial interfaces, and the ESP32. This value is not too high, which may cause data loss issues and introduce significant overhead, and not too low, which would also cause significant slow down in data transfer and program execution.

The solution to this problem is to make the debugging menu more interactive and only present the information that is requested. This is done in two ways: one by specifying a verbosity level, and second by allowing the developer to toggle debugging messages of specific subsystems such as infrared or the display writing subroutines. A verbosity level is a number the specifies the granularity of output that is desired by the user, the higher the level, the more granular information is supplied to the developer. At the default (0) level, only major events are recorded, such as low-power shutdown, and at the highest verbosity levels, almost every event such as each display refresh is announced. The debugging system will be implemented by creating a wrapper for the "Serial.println()" function, which takes in a string and outputs it to the console. The wrapper function will take in the original output string, as well as two constants that specify the verbosity level of the statement, and the subsystem that it relates to. The wrapper itself will check the current debugging settings before deciding whether to print the data or not. Setting the debugging setting in real time will be done through serial input. A set of characters can be used to adjust the settings during the operation of the firmware. For example, the character ‘q’ will toggle the infrared subsystem’s debugging settings, and the ‘+’ and ‘-’ characters are used to increase and decrease the verbosity levels respectively.

5.2.11 Standby Mode

A core feature that needs to be implemented into the project is a standby mode. During standby mode, every microamp needs to be conserved to maximize battery life while the gun is not being used. A lengthy standby duration capacity is important for convenience of use for the intended audience. This allows the gun to be charged and then transported to wherever the user intends to play. This also allows multiple rounds of the game to be played at different times on the same battery charge. This is important as the lithium ion battery has a finite number of charge/discharge cycles. As the lithium ion battery is charged and discharged, the capacity of it deteriorates. By implementing a standby mode, the battery’s capacity is used more efficiently, and thus allows for a longer life cycle of the battery.

Traditionally, battery powered toys directly use a physical switch for cutting the power to the main system. This has its advantages in that the quiescent current draw of all the components becomes irrelevant, which makes implementing this form of powering down very simple. The team decided not to use this method for several reasons. The first reason being that the inclusion of a physical switch adds to the physical footprint of the gun’s electronics. Since one of the core goals of this project is to create a compact product, including the switch makes this goal harder to achieve. The second reason is the team believes that a physical switch is a less polished solution. The user experience is an important aspect of this project, which makes this decision a priority. It is quite common for the physical switch to be accidentally left on, which will drain the battery. This will be annoying if it happens to the user, and using a software based solution can remedy this problem.

The first challenge to tackle is deciding when to put the ESP32 into sleep mode. The two options are to either have the user decide when the gun goes to sleep, or automatically put the gun in sleep mode when the right time is detected. Having the user decide is not ideal as it poses a similar problem to the physical switch where the user may forget to put the gun into sleep mode. Detecting the optimum time to put the gun to sleep mode through software is very doable, and provides a great convenience and benefit to the user. Firstly, the only time the gun can be put into sleep mode is if the gun is currently not engaged in an active game. The gun can only be put to sleep if it is in the setup mode, which will prevent the gun from accidentally going to sleep in the middle of an ongoing game. Secondly, the trigger and accelerometer inputs can be used to determine when the gun is not being used anymore. Using a timeout since the previous trigger press and motion detected from the accelerometer, the software can detect when to put the gun to sleep. An algorithm that combines both inputs will be devised which intelligently decides a timeout countdown.

Implementing standby mode can be done by minimizing power usage of of the CPU and auxiliary components. Firstly, any relevant data is stored to flash memory, such as default settings. This allows for preferences such as team name to be persistent across power cycles of the microcontroller, which enhances the user experience as less time is needed to set up a new game every time. It is important to save relevant data to flash memory as variables held in RAM will be lost and reinitialized on the reset of the microcontroller. Once the data is stored, the process to reduce power usage to minimal levels can be initiated. Firstly the wifi and bluetooth radios can be disconnected and disabled, which significantly reduces power draw. Then the auxiliary power switch can be disabled, which will turn off the OLED display and accelerometer. These devices have a significant quiescent current draw, so directly disabling the power bus is the best method of reducing its respective current draw. Since these devices will be completely turned off, they will need to be reinitialized on when exiting standby mode. Then the ESP32 itself can be put into sleep mode. There are several sleep modes to choose from, each with its own pros and cons. For our project, we will be using the deep sleep functionality, which completely shuts down the processor, until it is woken up from an external source. The two possible external sources are an onboard real time clock module or an external pin interrupt. Our project will use the the trigger as the wake up source.

5.3 ESP32 Pin Diagram

Prior to delving into the tentative/prototype PCB design, we found it to be appropriate to discuss the pinout of the ESP32 in terms of which pins are viable for this project and how they will be utilized. Extensive discussions will not be made regarding which part will go to what pin, however, it is vital to know the functions of each pin and whether it is possible to use them as an input, output, or both. This will prove important since devices will soon need to be assigned to pins such to proceed with the coding and the board design. This will also be important as to avoid any future problems that may arise from misassigning pins. The following figure shown as Figure 27 is the pinout for the ESP32 Development Board we will be using with the pins labeled by their type. Below is such a diagram which labeled each pin on the ESP32 Development Board V1 30 pin edition we will be using for this project.

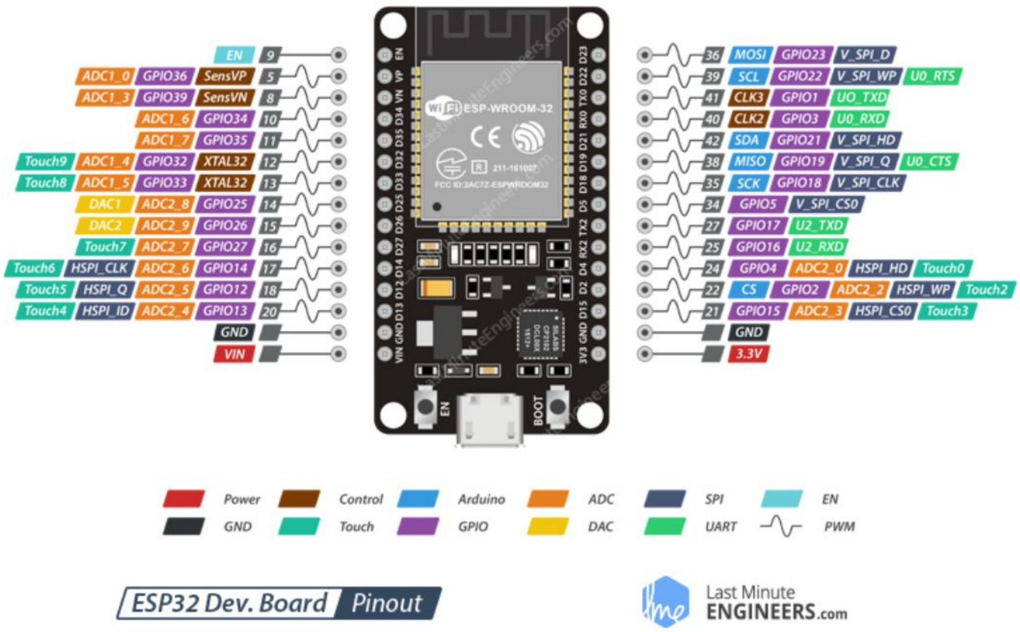

Figure 27: ESP32 Development Board Pinout *Produced by Last Minute Engineers [3]*

Power Pins and GND: There are two power pins located on the ESP32 Development board. The power pin marked as "VIN" is the pin that external power will be provided to the device on. This voltage can be anywhere from 6 to 20V though it is recommended that 7 to 12V should be used due to overheating that can occur at the higher voltages [4]. The pin marked as "3.3V" is an output voltage from the device caused from internal voltage regulation as an excess voltage. It could be used to power other external devices. The GND pins are ground pins that should be connected to ground.

Arduino Pins: GPIO pins 2, 18, 19, 21, 22, and 23 are known as the Arduino pins. They function as the I2C and SPI pins so they could be connected to any type of sensor or peripherals. It is projected that one of these pins will be used for the IR led receiver. The pair of hardware I2C pins will be used for communicating with the display and accelerometer.

GPIO Pins: Out of the thirty pins available on the ESP32, twenty-five are marked as GPIO pins. These pins can be used can be used for a variety of purposes. Most pins are marked as input and output supported but it should be noted that GPIO pins 34 and 35 are input only pins which do not contain any internal pull up or pull down resistors [5]. It should also be noted that the maximum current draw from these pins is 40mA per pin so special design considerations should be made such as to limit the current. All of these pins can be used as interrupts as well.

ADC Channels: On the ESP32 we will be using, there are fifteen pins across two channels that are used for analog to digital conversion. These pins can produce a digital signal of 12 bits and can register input voltages from 0V to 3.3V, however, it is difficult for this device to accurately differentiate two voltages within 0.1V of eachother so designs will have to be made such that there is at least a 0.5V difference between two input voltages. These voltages can be measured while the ESP32 is in sleep mode. Only one of these channels is needed to monitor the capacity of the onboard lithium ion battery during gameplay and charging.

DAC Channels: There are two digital to analog pins (GPIO 25 and 26) on the ESP32 that can accept an 8-bit signal. Once read, the pin can produce an output voltage from 0 to 3.3V. These pins could be used as an output that will provide voltage to different output devices, such as the buzzer which allows for all sorts of audio feedback to be incorporated into the project.

Touch Pads: Nine GPIO pins on the development board can be configured as touch pads. This pins use capacitive sensing to detect if an object, such as a finger, is nearby. We project that we will not be using these pins for this specific reasons and will instead treat them as generic GPIO pins to be used for various purposes. The availability of these pins, however, are convenient as they allow for future iterations to possibly incorporate touch interactions into the user interface.

UART Pins: Six UART pins are offered on the development with two interfaces (UART0 and UART2). Each interface has a TX and RX pin while UART0 also offers RTS and CTS pins on GPIO 22 and GPIO 19 respectively. These pins can communicate at speeds up to 5Mbps. Currently, the project uses no peripherals that communicate via UART, and thus only one bus will be needed to facilitate serial based debugging and uploading of firmware.

SPI Pins: Twelve SPI pins (or three sets of SPI pins with one MOSI, MISO, CS, and CLK pin per set) are available for use with three available modes (SPI, HSPI, and VSPI). These pins feature four timing modes, can support up to 80 MHz, and use the first-in-first-out approach of memory storage up to 64-bytes. Currently the project does not plan to incorporate any SPI based peripherals, and the SPI bus will only be used natively to interact with the SPI based flash file system located onboard the ESP32.

EN Pin: Simply stated, the enable pin (or EN pin) enables microcontroller to be on or off. However, in the ESP32's case, the "off" setting (ie. when the enable pin is low) puts it in a a low power "sleep" mode. The ESP32 WROOM is turned on when the enable pin is pulled high and runs at the sleep mode when the pin is pulled low. Since we wish to implement a low power mode to conserve power, special attention should be paid to this pin and it should not be simply tied to a battery and left high. A switch between this pin and the power supply may be useful to enable/de-enable this pin such that the user can control what power mode he or she may want the device in. The main advantage to this feature is that the power supplied by the battery will be used more efficiently such that there will be a longer duration between charging times leading to extra play time.

PWM Pins: Referring to the pinout figure, it should be noted that almost all GPIO pins can be configured as PWM pins with the exceptions being pins 34 and 35 since they are input only pins. PWM, also known as pulse width modulation, describes a square voltage wave where the voltage varies from V+ when high and 0V when low. Hence, the PWM pins have the ability to produce a square wave where the user needs to define the following features of the wave:

- Frequency
- Duty Cycle
- PWM Channel and Output Pin.

The initial testing and prototyping will include the ESP32 Development Board. This facilitates a quicker development time as the Development Board includes an onboard USB to serial converter for easy debugging and flashing of firmware. The Development Board is also compatible with breadboards which allows for quick changes to be made to the configuration of the prototype before the schematic is finalized. A development style board will not be incorporated for the final project due to several reasons. The development board includes many features that are not necessary for the final project, needlessly driving up the cost of the product as well as the overall footprint of the PCB. The development board also is not optimized for minimal current draw, as it is intended to be powered mainly by USB power. The quiescent current draw of the onboard USB/Serial converter and voltage regulator are simply too high to support battery power with an optimal amount of standby capacity. Thus the final product will instead use the barebones ESP32 WROOM module. This device is the basis for the development board and in theory should be relatively straightforward to convert from using the whole microcontroller to this one device. The main advantage of doing this is that we will be able to save space on the PCB and make it smaller, and give the team the flexibility and customizability with the rest of the power components to optimize battery life.

6.0 System Housing

For the project, there will be several electronic components that need to be secured in the laser tag gun. The system will need to be designed to withstand any conditions it is put in. These conditions can vary from playing indoors at room temperature, playing outside in the blistering heat and high humidity, or playing out at night in the cold.

6.1 Material

The material would need to be durable and portable in order to be able to be played anywhere. We are looking to have the system to be able to be held with one or two hands and not be very heavy. Most of the components are very small which helps with the weight and being able to fit in one housing. The housing will be 3D printed since that is a very cheap prototyping method and can be done fairly quick. In order to 3D print the housing we have to pick a material to print the housing out of. There are various kinds of 3D filaments to choose from with different characteristics. One of the characteristics to look at is the shore hardness. Other characteristics include flexibility, durability, price or how easy it is to work with.

6.1.1 Shore Hardness

Shore Hardness is a measurement of how hard a material is. It is common to differentiate between 3D filaments by this physical property. The higher the shore hardness the less flexible a material is and the more resistant to being indented a material is. The shore hardness is tested with a durometer which uses a spring loaded needle like indenter at a specific force depending on which scale is being used. The hardness is determined by the distance the indenter penetrates the material [1]. There are two main scales that are used. Shore A and Shore D. Shore A is used for more flexible and soft materials while Shore D is for more rigid and hard materials. There are overlaps from one scale to another which means a material with a shore hardness of 95A is the same as a material with a shore hardness of 45D [2].

6.1.2 TPU (ThermoPlastic Polyurethane)

TPU came from TPE (ThermoPlastic Elastomer). TPE was a plastic with elastic properties. It could be stretched up to two times its size and go back to its original form without any deformation. It has a shore hardness of 85A which led to it being handled poorly by extruders that weren’t meant for flexible filaments. TPU is a new form of TPE. It combines low polarity and high polarity segments into a single polymer chain. Low polarity segments are softer segments and high polarity segments are harder segments. TPU has a shore hardness of 94A which makes it easier to be printed from extruders. TPU is a rubbery material which is resistant to scraping or wearing [3].

6.1.3 ABS (Acrylonitrile Butadiene Styrene)

ABS is a petroleum based material that is strong and durable. It is heat resistant and has a smooth finish. It shrinks when it cools which leads to inexperienced users to have difficulty with high precision printing. ABS has a slight flexibility which helps it to be more durable and longer lasting. ABS usually needs to be printed on a heated bed [4]. ABS is commonly used in LEGOs and electronics housings. ABS is resistant to heat and will not melt since it is an amorphous material.

6.1.4 PLA (Polylactic Acid)

PLA is a material made of various renewable resources. In the US and Canada, it is usually made from corn starch. In Asia it is made of starch and tapioca roots. In other places sugarcane is used instead. A unique feature of PLA is that it is biodegradable and non-toxic. PLA also doesn’t need to be printed on a heated bed [5]. PLA will melt at around 173 degrees Celsius which is about 343 degrees Fahrenheit. This is much greater than the gun is going to encounter during gameplay.

6.1.5 PETG (Polyethylene Terephthalate - Glycol Modified)

PETG is a variation from PET. PET is a very common plastic that is used all over the world for various products. It is strong, resistant to extreme temperatures, and retains moisture. It is very often used to store food. PETG is PET but with added glycol. Adding glycol makes the plastic stronger, less brittle, more clear, and more easy to use. PETG is less likely to shrink and warp when printing. One very interesting feature of PETG is that it is recyclable. You are able to reshape it over and over again after applying heat to it. A drawback of PETG is that in order to print with it you will have to go through your printer settings and tweak them to make the printing strong and to a level of detail that satisfies your needs [6].

After looking at all the options, the 3D model will be printed with ABS. It is very strong and is one of the cheaper materials to use. It has been used numerous times in the past which will lead to easier printing and more help available if problems with printing are run into. The slight flexibility will help absorb any shock that would otherwise crack the housing if the gun is dropped or hit against a hard object such as a wall or tree.

6.2 3D Modeling Software

The housing needs to be 3D modeled. One software that can be used is Autodesk Fusion 360. This is a software offered by Autodesk and has all the part files stored on a cloud server. The tools within the software are very similar to Autodesk Inventor and AutoCad. The software is offered for free for students, teachers, and academic institutions. The requirements for running the software are outlined below in Table 14 [7].

Operating System	macOS version 10.12 or higher or Windows 7 or higher
CPU Type	64-bit processor (32-bit processors are not supported)
Memory	3GB or higher (4GB or higher is recommended)
Graphics Card	512MB GDDR RAM or more
Disk Space	~2.5 GB
Internet	DSL or faster

Table 14: Requirements for Fusion 360

Once the software is acquired, there are tutorials that can be used to practice and get better at modeling components. Some of the tools in the software are fairly straightforward but take some getting used to. Once we get over the learning curve we are able to model just about anything.

We are going to model all of the components that we are using. By doing this we can plan out how to design the housing the best way. We are able to move the components around in the software and resize the housing in order to make it fit everything and arrange it in the best way. In order to make sure the modeling is done accurately some of the components can be printed to compare the model to the physical version.

One of the best things about 3D modeling our housing is that it can be easily edited and reprinted in a short period of time. The modeling changes can be done in a very short period of time, as short as a few minutes if they are small changes. Depending on the size of the housing it can take a couple hours to print a new version of the housing.

6.3 3D Model Design

In Senior Design II, the 3D models were created for both the gun and trigger using the Autodesk software as shown below in Figure 28 and 29 respectively. These two hardwares are separate components but will fit together in the end product. The trigger works by utilizing a spring and pin located within the external frame. It will pivot about the pin located just above where the user’s finger will be. When pulled, the upper bar will pivot upwards and press the button located on the underside of the PCB. This will signal the ESP32 to turn on the IR LED as to fire the laser gun.

The IR LED will be pointing down the barrel of the gun and project the code out the hole shown in the figure. The receiver, however, will be perched on top as to be able to easier receive signals from enemy fire. The OLED screen will be facing outwards from the back of the gun, towards the user, so the user can easily see his or her ammo, health, and time left for the match. The PCB will be located inside the gun running parallel to the barrel while the battery will be located within the handle.

Figure 28: 3D Model of Gun

Figure 29: 3D Models of Trigger

7.0 System Testing and Demonstration

Once we are done designing and creating a prototype, we will go through multiple tests to check the performance of our project and make sure everything works well together. We will have to do testing both indoors and outdoors since the system is expected to work in both of those locations. Each location will be able to test different aspects of the project.

The outdoor location will be able to test the range of the infrared communication. We will try different angles and how well the communication works. The infrared beam should be dispersed when it hits the housing and then easily read by the receiver. The housing will also be able to be tested to see how well it holds up to heat and humidity. The durability of the housing will also be able to be tested. The visibility of the screen may be different depending on how bright it is outside or if it is cloudy or if the person testing is under a tree or near a building. All of these different situations will be checked to ensure that the screen is visible and able to be read at any situation. The haptic motor will be tested to see how responsive to different situations it is. If there is any delay in when the player is hit with a shot and when the gun vibrates we will have to go through and check the software to make sure there is not any methods or loops taking too long and slowing down the rest of the system. The trigger can also be tested to see that when the trigger is pressed the button is actuated. The haptic motor will also vibrate for a brief moment to indicate a “shot” or a message being sent to another one of the laser guns. Depending on the temperature, the housing may get condensation on it which may affect how well the LEDs are visible or how well the infrared receiver works. The LEDs will need to be able to be viewed from any situation and the display may fog up depending on the temperature as well.

The indoor location will be able to test how the IR communication works at short range. The screen will be able to be tested indoors to see that it be able to be read with more light more reflections. The haptic motor and trigger can also be tested to see that when the trigger is pressed the haptic motor vibrates for a brief moment to indicate a “shot” or a message being sent to another one of the laser guns. The haptic motor will also need to vibrate when the gun is shot by another player. The vibrations of the haptic motor should not loosen any of the components within the housing. If the vibration is too strong this may occur and the vibration intensity will need to be adjusted. The housing can be tested indoors to see if it is dropped on a hard floor or a carpet if it is resistant to falling and will not be damaged much. The infrared beam will need to be tested to see if the beam reflects off any mirrors, windows, or walls. The material of these objects will determine if the beam is reflected but how much is reflected will need to be tested to make sure little to no reflection is occuring.

Unit tests:

The program will start the bootup process by first running unit tests on all the components onboard, and reporting back if any errors are found. Unit tests are simple tests on each individual component of the system to verify that they are all functioning correctly, on a basic level. Unit testing is a industry standard practice, which has been proven to improve software reliability and reduce overall development time. This is formally known as test-driven development. We plan to follow this paradigm and integrate it into our project to reap similar benefits. These unit tests are meant to be run quickly and will not thoroughly test each hardware component, due to time and feasibility restraints. This will help with debugging any issues that arise with the development of the project, and can help check for errors during the testing phase as well.

Unit test:

- Ping all i2c devices and record the ACK responses, if any. For sensors, a sanity check on actual data received back from the devices can help determine proper functionality of the sensor.
- Run the haptic feedback motor for a short burst, the user will be able to feel the haptic feedback and confirm its functionality.
- Draw a basic splash screen on the display, as well as results from previous unit tests, then retrieve the frame buffer from the screen and check to see if the bits are correctly set.
- Draw a boot up animation on the LEDs.

8.0 Prototyping

Once it has been confirmed that the hardware works in its own right and it is time to assemble the hardware and software as a unit, the printed circuit board (PCB) will be necessary for a professional looking product. The function of the PCB is that it will tie all the hardware components together to the Wifi module and microcontroller so that it can fit inside the laser gun in a more organized manner.

This section will discuss:

- Things to consider when designing and ordering the PCB
- Comparing and contrasting different manufacturers
- General PCB Design
- Testing and Expectations

8.1 PCB Overview

The creation and implementation of the PCB is a requirement to the EEL4914/EEL4915 Senior Design course. Several initial concerns need to be addressed prior to constructing the design and going through with the ordering process. These concerns include finding a software that can assemble the PCB design into a gerber file and also finding a reputable manufacturer that balances low cost with good products and quick turnaround times. Both areas were researched and conclusions were able to be drawn as to which software and manufacturer will be chosen based on desirable characteristics found.

8.1.1 Software and Manufacturer Choice

The PCB constructing software that will be used in this project is called EagleCad or just Eagle for short. As a product of Autodesk, Eagle is provided to students at an accepted university or college for free and has a prolific community following. Because of this, there are many tutorials available that will be useful when learning the software. In addition, the software has the ability to produce a gerber file once the design is complete. This is important since most manufacturers accept gerber files when they ask for the design. All things considered, Eagle seems to be the best software for this project for its individual needs.

Another alternative PCB constructing software that may be used as well is EasyEDA. EasyEDA is a free online PCB tool similar to Eagle where you can create a schematic, PCB, and save it all as gerber file to be sent to the manufacturer. The advantage of using EasyEDA over Eagle is that it offers a sharing option, much like Google Drive, where a project can be shared to a group of people for review or to edit. This would be beneficial for our group since it is not required for us to be in the same space to view the current PCB design on one's computer. Since this team is inexperienced with using these PCB design tools, and EasyEDA is the most user friendly of the two having no need to install libraries and offers many tutorials, we will use this software.

Different PCB manufacturers from across the globe were analyzed to see which ones would be the best for this projects needs. Since this project is self funded and the odds are that more than one batch of PCBs may need to be ordered, it is preferred that the PCB cost and shipping would be as low cost as possible within the bounds that the PCB will still be of good quality and that the ship time is less than 20 days. To analyze these factors, quotes were created for five 100x100mm, 2 layered, 1.6mm thick PCBs with no stenciling or silkscreen. These may not be the dimensions or qualities needed for the PCB that this project will need (which will be designed later prior to the 100+ page report) but this represents a product that most PCB manufacturers can make and is strictly used to estimate general costs for manufacturing and shipping. Below, in Table 15, are the quotes of four different reputable manufacturers:

Company (Location)	Total Cost for PCBs	Shipping Cost and Time	Link
Elecrow (China)	$4.90	$31.34 in 14 Days (FedEx)	[1]
JLCPCB (China)	$2.00	$25.99 in 7 Days (UPS Worldwide Express)	[2]
BasicPCB (USA)	$98.00	Free UPS Shipping Unknown Ship Time	[3]
Bittele (Canada)	$172.37	$26.35 in 5 Days (Fedex)	[4]
Imagineering inc.(USA)	$31.50	Not listed, 1 week turn around	[5]

Table 15: Price Comparison of PCB Manufacturers

Summarizing the table above, the manufacturers from China offer the best prices for PCBs at competitive shipping prices to the North American manufacturers. The shipping time itself from China is very reasonable and with certain precautions, such as not waiting last minute to purchase the PCBs, can work with the Senior Design schedule. For the above reasons, JLCPCB is the manufacturer we will go to when ordering the PCB.

8.1.2 PCB Design

As described before, the PCB that we will be using will be designed with the intent of connecting all of the devices together. In this section, the design process will be outlined and some specific aspects of the PCB that will be needed for this project will be defined. In the process, some terminology will be glossed over that are standard in the PCB industry that should be addressed in order to get a fuller understanding of the PCB and what aspects are needed and not needed.

8.1.3 Final PCB Model

During Senior Design II, the final PCB was created using EasyEDA. Figure 30, shown below, shows the final PCB model that was created and sent off the manufacturer. Its long, rectangular, shape is due to the fact that this PCB will sit in the middle/barrel of the gun and needs to be a shape that complements the housing's dimensions. In terms of orientation, the OLED will be facing the laser gun user when in use and the ESP32 will be facing the outside barrel of the gun. The IR LED has been strategically placed near the ESP32 so it will be close to the barrel. The LED receiver is placed behind the LED assembly but it's designed so that it will be able to be perched on top of the housing. This is so it will be able to receive the IR sent code from any angle without interference from any of the onboard lights.

The battery and USB inputs will be soldered onto the top of the PCB using the four through hole components seen. These holes are traced to their respective power components that drive the voltage levels, charging, and safety devices. Also included are the programming pins beside the ESP32 so that the microcontroller can still be written to once soldered on the board.

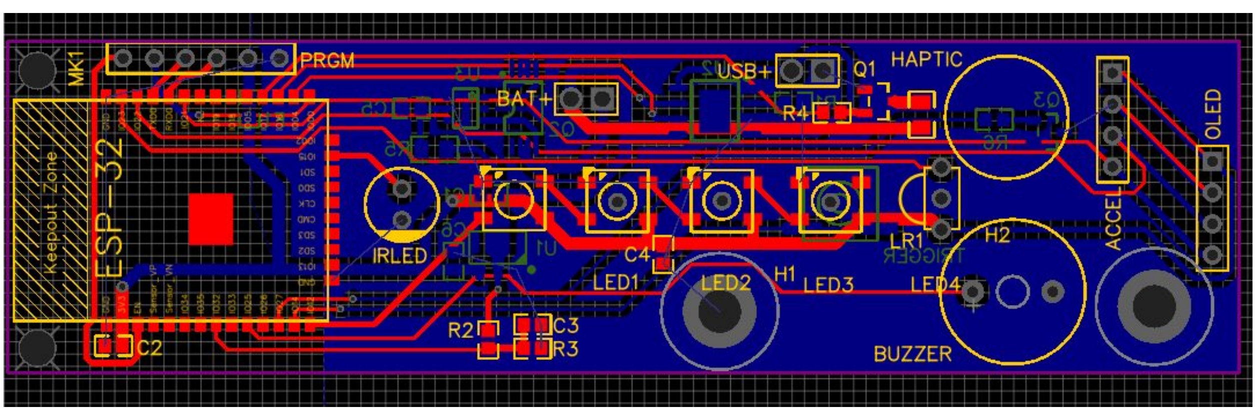

Figure 30: Final PCB Layout

8.1.4 PCB Assembly Considerations

While it is possible for external, surface mount components to be hand soldered to the PCB, a professional service will be used for most PCB construction to avoid potentially costly and time consuming mistakes. There are several companies available that offer such services.

The most promising company is Quality Manufacturing Services (QMS) in Lake Mary, Florida. QMS offers high quality PCB assembly that can be “fast-tracked” if necessary. QMS provides individual support for each project, which is ideal for this project, as this is the first time designing PCBs for most group members. The proximity of QMS and the ability to fast-track are also draws for utilizing QMS, since time management is a key part of success in this project. J.E.T.S. is a manufacturing company in Orlando that offers services similar to QMS. While J.E.T.S. is local, their company website has indicated that most products manufactured there are bulk products. The company has been contacted to see if they could accommodate the project. PFI Parts in Longwood, Florida, does not specifically advertise PCB assembly services but has been contacted to see if their facilities could accommodate this project. Again, the proximity of PFI Parts would make it another good choice of company.

There are several electronic device repair companies, both local and remote, that specialize in soldering. Their soldering typically involves circuit boards in cell phones, laptops, or tablets, but would most likely be able to perform the soldering required by this project as well. These businesses are small and have been contacted to see if they would be able to accommodate this project. One local company is Fix Stop, and has several convenient locations throughout Orlando. UBREAKIFIX is another company that has several locations as well.

There are two obvious drawbacks to utilizing an outside service for soldering. The first is cost. As keeping the final product low cost is one of the main objectives of this project, any extra cost is in direct conflict with that goal. Since this project has a very strict deadline, time is another drawback to sending the PCB out for assembly. Any back-ups at the manufacturer could potentially put the project as a whole behind schedule. However, soldering by hand could possibly have the same effects. Mistakes made soldering by hand could possibly result in ruining an entire PCB. If that were to happen, new PCBs and components would need to be ordered. This would also result in extra cost and delays in productivity.

The group has decided that most soldering work will be done in house by our own members. This allows us the opportunity to learn a new skill that will be useful later in our careers. If any soldermount parts are deemed too difficult to solder for our expertise level, an outside company will be contacted to do the soldering in a professional environment.

8.1.5 Terminology

Before designing, some of the terminology should be reviewed. These terms will be useful when designing and placing the order for the PCB board as most manufacturers request this information when submitting a design. Discussed as well are some decisions regarding the proper specifications required for our board.

Layers: The number of layers in a PCB is defined by the number of copper layers used. These copper layers are used to connect the components on the PCB and to facilitate in current transfer. The simplest board consists of 1 layer with all the components connected by traces (solid copper connections) on the top side of the device. A more common option is the 2 layered PCB which allows for the copper traces to connect components through both the top side and bottom side of the board. As shown, the more layers used, the more connections can be made without the worry that the traces being too close or even overlapping on the same layer, however, the more layers used, the more problems arise in terms of unwanted impedances or propagation delays [6]. Through JLCPCB, we have the option to choose either 1, 2, 4, or 6 layers for our design. Since the board will be relatively simple and problems should not arise with fitting all of the traces in if designed smartly, we will be using 2 layers for our design.

Dimensions: The dimensions of the PCB is defined in terms of the length and width sizes usually asked in millimeters. We aim for the PCB to be as small as possible such that the laser gun can be designed to be less bulky. The current goal is to design the board to be no bigger than 100mm x 100mm or about 4in x 4in. This can be achieved by having some of the bigger components, such as the battery, wire-soldered to the board and downgrading from using the entire ESP32 Development Board to just the ESP32 WROOM.

PCB Thickness: The thickness of the PCB is mostly made up by the thickness of the substrate. The thicker the board is, the better since it will be more durable and there is less concern for the ground plane being too close to the power and signal planes[7]. For our design, we will be using a thickness of 1.6mm since this is considered the baseline, standard thickness of a board that delivers a low amount of power.

Surface Finish: Three types of surface finishes are offered through the manufacturer we have chosen: HASL(with lead), Lead free HASL-RoHS, or ENIG-RoHS. HASL is most commonly used finish for PCBs since it is cheap and has some convenient features applied to it during the design process. For example, an “air knife”, which is hot air blown across the board, is used to remove excess solder and inevertantly applies hot air to the PCB. This allows for any delamination issues to present itself prior to any components being soldered to it [8]. This serves as an extra check to ensure that the quality of the board is maintained. The ENIG finish, on the other hand, is much more expensive and does not have those features in the manufacturing process. It does, however, utilize nickel and gold to both protect the copper and extend the shelf life of the board. For the sake of this project, the HASL-RoHS surface finish will be used since shelf life isn't going to be a huge concern given the duration of this project, it is low cost, and because of the manufacturing features explained above. In addition, with it being RoHS certified, the manufacturing processes will be more environmentally conscious as described in the corresponding ABET standard.

Copper Weight: The weight of the copper is proportional to how much current is being flowed through the PCB. For high power application boards, a higher copper weight is needed (2-3oz), while lower power applications need only 1oz. Since our design will only need to transfer a few milliamps, the 1oz weight would be applicable.

Programming Header: The program header is used in order to upload code onto microcontrollers on the PCB once they have been soldered. They typically are cheaper, require less current/power draw, and take up less space as compared to a USB or serial input. For this project, a Tag-Connect [9] is one type of programming header that has been researched and could work for our needs.

Silkscreen and Soldermask: The silkscreen is used to add words, logos, or acronymes to a PCB. The silkscreen is typically printed in white and while not necessary for the function of the board, it is convenient to have in place so that the different pads can be identified prior to anything being soldered on without needing to continuously look at the schematic. We will be using the silkscreen to identify the location of the specific components to make it easier when we solder them on. The soldermask, unlike the silkscreen, is there not purely for aesthetic reasons; it serves as a protection for the copper layer below. The soldermask can come in a variety colors which do not affect its functionality, primarily green, blue, red, or white. Through JLCPCB, the cheapest option is green so we will go with that color.

Screw Holes: The holes that will be designed in the corners of the PCB will be utilized as a mean to secure the board to its housing. Since our PCB will be housed within a 3D printed chamber of the laser gun, it will have to be secured to one of the sides with screws such that it will not be tousled around while in use. In addition, some raiser can be added so that the PCB has some distance between itself and where it is secured to such that sufficient air can reach all areas of the board which will assist in the cooling process. The most important factor is the secureness of the board since the external trigger is completely dependent on it. The external trigger, which will be pushed by the user when they want to "fire" the laser gun will be attached to a button on the PCB such that when the trigger is pushed, that button will be pushed and the microcontroller will flash the IR sender. If the PCB were to shift out of place and the trigger is no longer lined up with the on board button, the function will be unable to occur.

Surface Mount: Surface mount devices (SMD) are the standard electrical devices, such as resistors, capacitors, or diodes, packed in a form that it can easily be soldered to the pads on the board. Resistors and capacitor surface mounts are provided with a four digit number which identifies its dimensions in hundredths of an inch measurements. Since they require less surface are just as inexpensive as its wired equivalent, the majority of the components on our board will be surface mount.

8.2 Prototype Testing and Expectations

Prior to ordering the PCB, some testing will occur to ensure that our preliminary design will do the things we set as design specifications. Both the software testing and hardware testing along with preliminary expectations are detailed below. It is important that these testing procedures are in place and are as detailed as possible such that no potential flaw will be concealed due to a lack of testing. As the designers of this project, it would be in poor taste to present a project driven by a PCB that may fail in normal game induced scenarios that would go against the goals set in the beginning of this project. This, of course, does not include the extreme cases that we would not expect an electronic of this caliber to survive such as being dropped while not protected by the 3D printed housing or being submerged in water.

8.2.1 Hardware Testing and Expectations

The hardware testing procedure will be a continued process from the beginning of the design process to the final product and will work side by side with the software testing process. The PCB will originally be designed on a breadboard at first in parts then brought together as a whole on a simulation tool like LTSpice that will be able to measure theoretical currents and voltages on the simulated parts without risking damage to the real parts on the breadboard. This way, we can change the design on the fly if it does not work and we wouldn’t waste money on a PCB that is an outdated design or ultimately does not meet the demands. In addition, we can test the hardware and the software work together as a unit by implementing the prototype software to the microcontroller, connect the microcontroller to the hardware by utilizing the breadboard, and run through the functions we expect the final project to perform. These functions include pressing a button and turning on the IR led, receiving and interpreting the IR signal, having sounds and haptics run as a response to a specific event, running the wifi module, and working the display screen/user interface. So that the breadboard does not become overcrowded, the functions may be needed to tested in parts until the first prototype PCB is in to do further testing.

After the correct placement of the electrical components have been determined, the design will be implemented using the Eagle software. Once added and the PCB has been generated, Eagle has its own checking tool to ensure that the components are connected correctly and that the design is in fact buildable. After the gerber file generated from Eagle is sent to the PCB manufacturer, they will check it once more to ensure none of the traces are overlapping and, after its built, that all of the pads are connected. Once finished, the PCB will be built and shipped to us. When the PCB has been received, the next steps to test it will be able to be performed. Once all the pieces have been soldered on, we will be able to provide power to the assembly and run through the design specification functions that the laser gun was able to perform while connected on the breadboard. In addition to the main functions such as the wifi and the user interface, some other important functions can be tested now that the assembly is pact in a more mobile form as compared to the breadboard such as distance allowed between the sending and receiving led and accuracy. These tests would be most convenient in this phase rather than inside the laser gun so the board would be easily accessible if any modifications of the code needs to be done. If it all works then the main testing phase has been complete and the PCB can be loaded into the 3D printed mold. From there, testing can occur with the board loaded into the laser gun. The most important aspect of this phase of testing is the trigger’s placement.

To prevent any setbacks in the case where a PCB may come defective or break, a total of five PCBs will be ordered at a time such that we would have a spare to use to replace any bad boards. If all the boards come out good and we do not have a need to replace any, the extra can be potentially used in the senior design showcase to show off the PCB and its functions without taking one of the other four out of the housing within one of the laser guns. It does not cost very much to order the extra PCB so there are little to no disadvantages to ordering it and as outlined above, the cost which is the true disadvantage to ordering the extra board is outweighed by the benefits.

The intent is that the first edition will be the only edition that will be ordered, however, we fully expect that at least two different editions, or batches, of PCBs may need to be purchased. Since it will be this team’s first time designing a circuit that will be reproduced on a PCB, there is a great probability that the first batch that we will receive may not work as intended due to a design flaw, however, with the research that has been completed so far and the careful calculations that will be taken before finalizing the design, we expect that these mistakes will be minimal and that a revised PCB can clear most if not all of the issues seen in the first edition.

A timeline for the PCB creation, ordering, and testing has been form such that we are held accountable to a schedule. Since the PCB is very time sensitive, given the amount of time it takes to ship, it is vital to have such a timeline available to keep the project on track. This schedule is shown below in Figure 16.

Week	Milestone Task
Week 1-2	Finalize Initial Simulation/Breadboard Prototype
Week 3	Have First Prototype PCB Built in CAD Program
Weeks 3-4	Review CAD File Send to Manufacturer for First Print
Week 5	Receive PCB Get Components Soldered Begin Testing
Weeks 6-11	Repeat Previous Timeline if Revision is Needed
Week 12	Last Opportunity To Re-order PCB
Weeks 13-15	Have Final PCB Mounted In Housing (Deadline) Perform Final Testing
Week 15	Final Presentation

Table 16: PCB Development Timeline

Through Senior Design II the team has stuck pretty well to this schedule and only required two rounds of PCB ordering. This put the team on schedule to do perform necessary testing and reprogramming. Scheduling the PCB development timeline kept our team on time with our goals and overall showed us the importance of keeping to schedules. This will benefit us later as we navigate meeting in between work and have to keep to tight deadlines.

In summary, first the Agile system of testing will be utilized when testing the functions, part configuration, and circuit design on the breadboard and will be done in parts rather than testing all the functions at once. By using this method, we cut down on the number of devices on the breadboard at once, make it easier to debug, and reduce the likelihood of a short being created that can heavily damage the parts and the microcontroller. Once the breadboard testing is complete and exact parts have been selected, we can test everything together in a safe environment by using a simulation tool that can read the currents and voltages in the circuit design. If everything blends together correctly then that circuit design can be implemented into EagleCad and sent to the manufacturer. Once receiving the PCB, a series of tests will be performed that will test the execution of the various functions we set to implement. If it fails any of these tests, we will need to revisit the breadboard, simulation, and EagleCad circuit designs and fix the issue before ordering another batch of PCBs from the manufacturer.

8.2.2 Software Testing and Expectations

The software will be tested side by side with the hardware one feature at a time. For instance, the feature that turns on the IR LED when the trigger is pressed will tested by making a code dedicated for that feature, uploading it to the microcontroller, and running the script with the IR LED and button/trigger attached to the microcontroller. This will be done for all features using the microcontroller and Wifi module we plan on using, the ESP32. Once all of the codes for the different functions have been created, they will merged into one main script that will call the codes created when needed. Like before, the hardware will be connected to the microcontroller and the full script will be run to ensure that it all works as we intend it to. From there, the PCB will be designed, purchased, the hardware components will be soldered on. The code can written or rewritten to the soldered ESP32 by using the programming header.

Like any code, we expect to go through a few editions before it works just right. Unlike the hardware testing, however, it is free to re-upload script onto the microcontroller and cost little time to do so. With the help of the programming header, has been incorporated into the design of the PCB, the software will be able to be tested in modified during all phases of the project up until the time of presentation.

8.3 Component Testing

The project will have multiple components that need to be tested and work individually before being able to integrate them all together. It is assumed that all of the parts that are bought are in good working condition, but in order to develop the project each part will be tested. Once all of the parts have been confirmed to work individually, the components will be all put together and tested as one system. If each of the parts work and the system is not working it may be an issue of combining the different components into one system.

The different components will be tested as subsystems. The subsystems are as follows: the infrared transmitter and receiver, the battery and regulator, the trigger, the haptic feedback, the LEDs, the buzzer, the accelerometer and display, microcontroller, and the wifi mesh. Each of the subsystems will be tested differently since they each work differently.

Testing each component individually is also imperative to quickly determining whether parts will be compatible with the project or not. This needs to be done early in the project as tackling this issue later on during the development can cause significant delays. In the overall project. Completion of the initial testing on a breadboard is allows the team to move forward and begin working on other sections of the project, including the final firmware, and drawing up of the final schematic. Completion of the final schematic allows for the kickoff of initial PCB designs as well, which allows for more development iterations and testing time.

These tests are not meant to be complete implementations of the project, nor are they meant to be thorough testing of each component. These tests simply help gauge whether the part will be compatible with the project and are worthy of the power of Odin. This also helps to gauge what further challenges the team may face which helps the team prepare for them beforehand, while also giving more data to use when creating important decisions down the line in terms of software design or component selection.

Although these tests are not complete implementations, they do provide a significant amount of progress that can be reused for the final firmware design. Code blocks that test a certain component, such as infrared, are easily recycled for the final implementation. This means that the progress made during this phase of the project is twice as important.

These test also comprised the initial section of our Agile workflow, as each component’s testing makes up a simple, independent task that can each be referred to as a sprint. This helps kick off initial progress in the project and builds momentum to carry through the entire project to completion.

8.3.1 Power

The battery and regulator will be checked to make sure the battery is outputting the right voltage and is the proper voltage is seen after the regulator. The regulator will power the microcontroller and will need to be at a steady voltage and current. The microcontroller used demands for 500 mA of current and 3.3V. The microcontroller has a range of 2.3 to 3.6 V as an input voltage to the power supply pins.

The duration of the battery and regulator when being used with the rest of the system will be tested once all of the components are tested and are able to be integrated into one system. The efficiency of the battery can be tested by seeing how much voltage is output compared to the voltage that is used to charge the battery. The duration of charging the battery is also needed to be tested. This will be done numerous times to see the average charge time from different amount of use. The duration of the battery charge can be tested by timing how long it takes the battery to get to a specified level. Whether this be 25% or 50% we can see how long the battery lasts when playing in a game and when the system is in standby mode.

Another set of data to collect here is a time series of voltage vs. time remaining on the battery. This will allow for accurate calculation of battery capacity remaining for the user interface. The data collected here can be directly transformed into a lookup table that will can be used by subroutines that calculate battery capacity remaining. This testing will also allow the team to gauge the size of battery needed to complete the requirements for the usage time.

8.3.2 Trigger

For the trigger, we will need to test that the button is able to close and open the circuit. The button will also need to be kept at a safe voltage and current level. For the tactile button it said to be able to withstand up to 50 mA at 24V. We can test this by making a small circuit and putting the button to the limits in the datasheet and see if the button can withstand it. The system should not reach a voltage near this but the current may go up to 40 mA when drawn directly from the microcontroller.

The physical trigger will need to be tested to make sure it is able to depress the button by being pulled. In order to make sure this mechanism works well, the trigger part of the 3D model will be printed on its own in order to keep from making the whole housing and save printing material. Ideally this will only need to be done once and then the whole housing can be printed on future prints of the housing. The trigger housing should be able to withstand the full pressure applied by a human finger without snapping.

It is important to make sure whichever button we use has minimal bounce, and that the software is able to accurately determine state changes without losing trigger clicks and without double counting them either. This testing will implement state change code detection which will directly be used for the final implementation, as the only thing changed will simply be the subroutines called when a state change is detected. The value to optimize here is the debounce delay used to detect state changes.

In the testing, the button was debounced using software. This was tested with pressing the button and seeing the digital signal change from a 0 to a 1. The testing was done with an arduino development board for fast testing of the button. The program would show a change in the signal anytime the signal change from 0 to 1 or 1 to 0 to make sure the signal did not oscillate from 0 to 1 and instead gave a steady value. The developed subroutine can be directly ported to the ESP32.

The trigger will be shaped as an L shape that will look like a trigger to an actual gun. When pulled, a lever attached to the trigger will pivot upwards towards the tactile button. In theory, the button will be closed when the trigger moves the tactile, causing a closure to ground in the circuit that will trigger the IR LED to fire. This trigger mechanism was developed by our team through a 3D modeling software as mentioned before. It will be 3D printed to save money and reduce the total weight of the system.

8.3.3 LED

The testing for the LEDs will be done by making sure all of them turn on and off. From there the different diodes will be checked to see that they work and that they are able to receive data input to set the LEDs to different colors. The voltage across the entire package will be checked to make sure it is within the specifications that the datasheet provides. Once the basic aspects of the LED are tested the brightness of the LEDs can be tested.
The LEDs need to be bright enough to pass through and illuminate the housing of the system. It is expected that the top of the housing will change colors to that of the LEDs in order for holes in the housing to be kept to a minimum. The LEDs are being written to by only one data pin which can lead to a delay from when the data is sent to the first LED to when the last LED receives the information.

Another important aspect to test will be the quiescent current draw of the LEDs when in standby mode. This is important to make sure that the LEDs do not drain the battery while the gun is turned off. If the current draw is low enough, i.e. in the nanoamp range, then the LEDs can be hardwired to the main power supply bus. If not, then the LEDs will need to be switched by the peripheral power bus.

Since the delay is 300 ns for transmission and there are only five LEDs being used it is expected for the delay to not be detectable by the human eye. The LEDs will be tested when in series and when individually programmed in order to compare the brightness and responsiveness of each. Individually each should be more responsive than when programmed in series, but since the delays is so small it may be the case that the difference is not noticable.

Once the library used for the RGB leds is tested and verified, a simple wrapper function that can be called throughout the firmware can be implemented. This can be directly ported into the final firmware, which means that progress made here will contribute significantly to the final project.

Other test have been performed on the IR LED itself as well as the IR receiver and display screen. This was tested by running the deathmatch mode on the ESP32 and playing a mock game between two guns. When one IR LED fired at the receiving gun, the test was considered a success if the receiving gun was able to register the hit. This was done at increasing distances until we found the point where the IR receiver was not consistent with receiving the signal. This distance was deemed the range which was found to be about 25 feet, just above the goal that was initially set.

One very important concern that has been on the team's mind is the IR LEDs strength and range. The goal for the project was to not utilize any lenses as to keep the price of the project as low as possible. Subsequently, testing had to be done in order to determine if the lenses, which would be used to amplify the signal as it is sent towards another gun, would be a necessity so that the range and accuracy requirements the team set would be met. Following testing, however, it was determined that a lense would not be needed in order to meet our 20ft minimum range. Since adding a lens would be unnecessary to meet our range and the cost of the lens needed would push the cost of the project very close to the $50 limit, the team decided not to implement this feature. In the future, if the team decides to further develop the project, a biconvex lens could be added in order to increase the range.

8.3.4 Haptic Feedback

The haptic feedback motor needs to be tested to make sure that the vibrations created don't damage other components. The vibration motor is mounted on the PCB which will help the motor be more effective in creating vibrations throughout the whole housing. The vibrations can be tested to see how strong the motor is. This can be done right at the leads of the motor and at the handle of the housing The haptic motor can be through hole in order to be more stable and not just be held on by solder. The flyback diode will need to be checked to make sure that the change in current, when the motor is turned on and off, does not cause any excessive voltage that isn't dissipated into heat. The current limiting resistor that is used to make sure the transistor, that is doing the switching, needs to be checked to make sure the resistor is the correct value and that is it is limiting the current.

8.3.5 Infrared Transmitter and Receiver

The infrared transmitter and receiver will need to be tested in order to make sure the encoded data in the beam is able to be decoded by the receiver. In order for this test to work there will need to be two sets of transmitter and receiver available and in working condition. If that is not possible, then one set can be used but will need to be able to be moved around and not affixed in position as it will be once mounted on the PCB. If two sets of transmitter and receiver are able to be used, the range of the transmission can be checked. This will need to be done without the housing of the gun and also with the housing of the gun. Without the housing will make sure the two components, infrared transmitter and receiver, work well and are not faulty or broken parts. With the housing will make sure that once the gun is assembled and all the components are put into one system that the receiver and transmitter will still function correctly.

8.3.6 Buzzer

The buzzer will be tested in order to make sure that our sounds are working well. The sounds that the buzzer makes will be checked in order to make sure that it is not too loud and that it is audible from a distance of about one meter. The different sounds need to be tested to make sure that they work well and do not get distorted if moving around or if it is windy or in other adverse conditions.

Another important aspect to test here is RAM and Flash memory usage of implementing sound files to be played through the speaker, as audio files can cause of significant amount of memory usage if we are not careful. Once the usage of memory is determined, the team will have a better idea of how many audio samples can be selected and where the should be appropriate.

8.3.7 Accelerometer and Display

The accelerometer and the display are both using I2C to communicate to the microcontroller. The accelerometer will need to be tested to make sure that when the gun is being held by a person the readings are within a range that does not make the gun react to motion. The main aspect that needs to be tested is if the accelerometer will be able to register that the player is pointing the gun towards the ground for a set amount of time. This time will be from one to three seconds and can be changed if that period of time is too long or too short. The accelerometer chosen includes as gyroscope as well, which measures angular velocity. Although currently there are no plans to use this feature directly, the gyroscope may be used to smooth out the accelerometer readings if testing deems it necessary.

The display will need to be tested in order to make sure all of the pixels are working. If any of the pixels are burned out and are not working a new display will need to be purchased to replace the broken one. The brightness of the display will be checked to make sure that everything being displayed on the screen can be seen in any environment. This includes being inside with bright lights on, outside in the sun, inside in a dark or poorly lit room, or even outside at night with no other source of light. Depending on the temperature, the display may get foggy or build condensation on it. In either of these cases the screen should still be able to be seen and easily wiped off with a shirt or napkin in order to easily resume playing.

8.3.8 Microcontroller

The microcontroller will need to be tested in order to make sure that any pins being used to drive components are at the correct voltage and current as specified in the datasheet. The current will also need to be checked to see if it is needed to have any current limiting circuits. The temperature of the microcontroller can be checked to see if it needs any cooling system or if the housing needs to have vents dedicated for air circulation.

8.3.9 Wifi Mesh

The Wifi Mesh system needs to be able to connect to multiple microcontrollers and be able to send and receive simple messages. The complexity of this implementation lies within the formatting and content of these packets of data, but that is not a part of this section of testing. This testing needs to just verify that it works correctly.

Implementation here will include the verification of mesh networking libraries, and include wrapper functions. These functions will be called on reception of new packets and provide an access point to broadcast messages to the entire network. These functions will be directly added to the final program.

8.4 Unit Testing

Figure 31 below shows the initial testing done for the project to determine if the components will work in unison when controlled by the ESP32 Development Kit. The team was able to get multiple subsystems operating on a single breadboard simultaneously. Our testing was able to incorporate the trigger button, OLED display, WS2812 LEDs and IR Receiver on the same breadboard. This proves that the components selected are a good match and that they are all compatible with each other. The operation of one will not interfere with the operation of another, and will allow for the smooth continuation of the project.

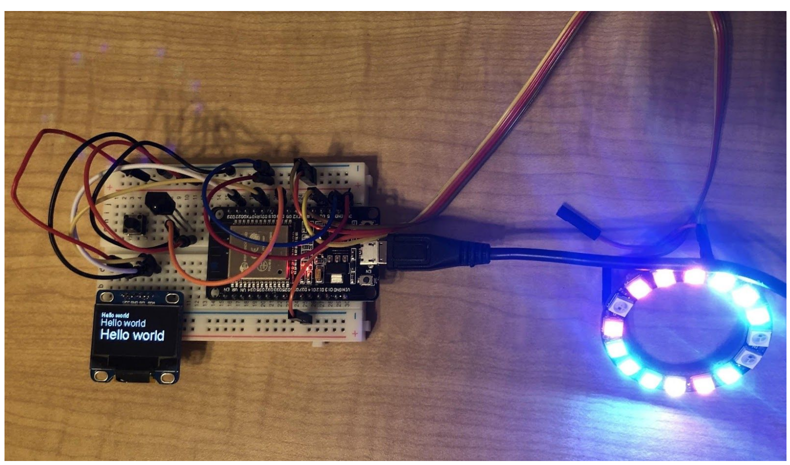

Figure 31: Breadboard Testing

Subsequent testing continued into Senior Design II that brought the design off the breadboard and onto a more permanent protoboard. This was the next serious step towards creating the PCB design as testing could now be done with soldered on pieces. Figure 32 below shows one of those testing protoboards with the OLED, accelerometer, IR LED and receiver, the tactile button/trigger, and underneath the ESP32, the vibration motor. Some issues were seen when pressing the button, as it would sometimes trigger the motor unintentionally, but that has been fixed with more secure coding. The OLED screen, at this juncture, can now the “Health” of the player as well as the “Ammo”. Future coding and testing on the OLED has now allowed it to display more features such as the gameplay timer, health and ammo, as well as battery life.

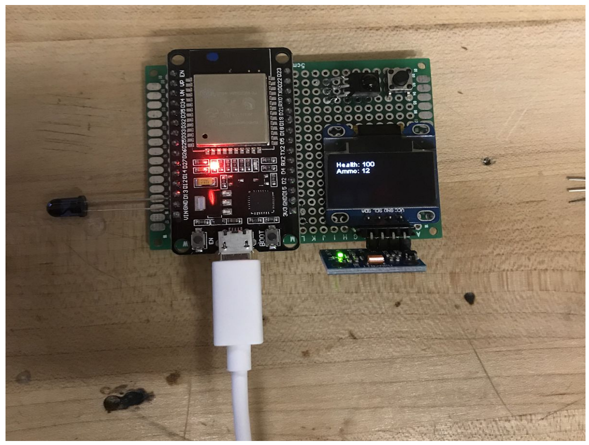

Figure 32: Protoboard Testing

As the semester progressed, the project testing became more and more elaborate. Following the successful protoboard testing, the PCB was ordered as to launch the next phase of testing. When the PCB arrived, the ESP32 was the first part to be soldered to the board. From there, parts were added and tested in increments in order to systematically test the functionality of the board. With some tender, love, and care the first board did run successfully but some optimizations were necessary for the final design. The second PCB arrived in April and the same style of testing was performed in order to confirm that the PCB correctly worked. The main hardwares and corresponding functions that were tested during this preliminary testing period were the LEDs, both IR and WS2812, the voltage regulator, the charge controller, the OLED, accelerometer, and speaker.

9.0 Conclusion

Before submission, this document was reviewed by all four members of the group. After detailed self-review and peer-review, it was determined that all research and designs were original and relevant for the given task. When outside sources were used, all references were cited properly in the appendices and referenced properly in the paper. Pictures used from outside sources were taken from the public domain or used with explicit permission from the owner of the work.

The paper outlines a project that is challenging but feasible to complete in the time allotted. The group expects to put in the time and effort required to make a final product that approaches perfection. The confidence in the project stems from the effort in research and design the group has done in Senior Design I and II. Acquiring parts and finalizing designs during Senior Designed I ensured adequate time to perfect the design in the spring semester. In an attempt to meet the project deadline, the team has allotted a significant amount of time to testing and planning, instead of jumping right into the build phase of the project. Precautions such as multiple prototyping iterations and the serial debugging interface are a priority for this project, as simple bugs can lead to large amounts of time being wasted. The team believes that the design, although complex, is certainly within our bounds to prototype, finalize, and demonstrate at the end of the course.

One major aspect of the project that separates this from many others is the fact that this project does not attempt to implement fancy brand new technologies that are yet to mature. The team is not looking to impress investors by implementing blockchain, machine learning, or flexible displays. That is not to say that working with these technologies is wrong, it just simply is not the priority of this project. This team will matured technologies to develop a full fledged product by the end of Senior Design II. The team is not looking to build a proof of concept, rather a professionally designed product that is to the polish of commercial retail products. The hope is that the project construction process may be open sourced one day so that others can replicate our project easily and be able to enjoy it as much as we have.

Major challenges regarding the project in general have come from PCB design and software design. The PCB design challenge comes from inexperience with the required software and design process in general. After a few initial attempts and constant research, the PCB design has proven to be to be less of a problem. The software design challenge comes from the way in which so many components of the blaster design are linked. A single trigger press results in a cascade of actions by the haptic feedback module, the speaker, and the IR transmitter. Making all components act in concert will be the biggest obstacle as far as software design goes. This challenge has been overcome by laying out the framework of the overall code very clearly before the actual code is written.

Other challenges presented themselves as the team progressed into Senior Design II and began to build the project. One main issue that was reoccurring was part procurement in an acceptable amount of time. Since the team was limited on time to build the project, shipping parts from overseas was not an option we could afford to choose for most parts. Many parts chosen were purchased through sites that allowed for free two-day shipping or other express delivery options. This did not hinder the quality of the final project but did cause a change in some interchangeable parts such as the vibration motor or speaker. Spare parts were redundantly ordered as well to avoid these extra shipping costs and wasted time in the case that an extra part was needed. The PCB took the longest out of all the parts we ordered since it needed to be custom fabricated then sent over from China. As a result, the same idea was applied to the PCB ordering process; an overall of 10 boards were ordered for each iteration when the project only required 4. The team ended up spending the extra dollar to have the PCB expressed shipped but was rewarded with the PCB coming in one week after the gerber file was submitted.

The physical aspect of building the PCB was another problem. Some confusion with using the PCB building software lead to some traces being connected to the wrong place on the board or shorting in some cases in the initial build. Though not recommended, the board was temporarily “fixed” by unshorting some of the traces through a little manual labor. With that done, the board was able to be used for some initial testing with some solder mount pieces attached. The second board that was sent out to be built would resolve the issues initially noticed in the first model so that a more professional board could be presented in the Senior Design Showcase. In addition, some pads were adjusted on the board so that it had a more desirable and smarter look than the first model.

Each of the four boards required to construct four blasters were tested according to the procedures outlined in section 8.3. Any issues were dealt with completely before putting the boards into the housing of the blaster, where it will be difficult if not impossible to isolate them to troubleshoot. Once the boards were installed in the housing, any ensuing problems were resulting from software issues. While the software is complex as mentioned above, a software problem may be easier to fix than a hardware problem, since it will not involve any reordering or redesigning of physical parts.

In the end, this project taught the team members more than how to build a circuit board or to assemble a project. It taught them the value of teamwork and how having a working relationship where everyone contributes can lead to something creative and fun. It also inspired the team to dream big and not give up just because the task ahead seems daunting. These are the ideals that we wish to give to those who play with our project and the legacy we want to leave at the University of Central Florida.

10.0 Appendix
Section 3

[1]
https://www.instructables.com/id/Duino-Tagger/
[2]
https://mil.ufl.edu/4924/projects/s11/final/Schoen_Stahl.pdf
[3]
https://en.wikipedia.org/wiki/Duty_cycle
[4]
https://www.designnews.com/gadget-freak/diy-laser-tag-pulse-laster-target-and-display/121215627457243
[5]
http://www.science.uwaterloo.ca/~cchieh/cact/c123/battery.html
[6]
https://en.wikipedia.org/wiki/Electric_battery
[7]
https://en.wikipedia.org/wiki/Lead%E2%80%93acid_battery
[8]
https://en.wikipedia.org/wiki/Lithium-ion_battery
[9]
https://en.wikipedia.org/wiki/Lithium_polymer_battery
[10]
https://en.wikipedia.org/wiki/Nickel%E2%80%93cadmium_battery
[11]
https://en.wikipedia.org/wiki/USB
[12]
https://www.extremetech.com/computing/115251-how-usb-charging-works-or-how-to-avoid-blowing-up-your-smartphone
[13]
https://batteryuniversity.com/learn/article/charging_from_a_usb_port
[14]
https://www.newnex.com/usb-connector-type-guide.php
[15]
http://www.ti.com/lit/ml/slup239/slup239.pdf
[16]
http://www.ti.com/lit/an/slyt412/slyt412.pdf
[17]
https://en.wikipedia.org/wiki/Operational_amplifier
[18]
https://www.renesas.com/us/en/products/power-management/linear-vs-switching-regulators.html
[19]

http://iot-bits.com/ultra-low-power-design-esp32-10-tips/
[20]
https://en.wikipedia.org/wiki/Bluetooth
[21]
https://computer.howstuffworks.com/wireless-network1.htm
[22]
https://en.wikipedia.org/wiki/Wireless_mesh_network
[23]
https://learn.sparkfun.com/tutorials/serial-communication?_ga=2.151809658.2127934353.1539696409-1041977720.1539696409
[24]
https://en.wikipedia.org/wiki/I²C
[25]
https://learn.sparkfun.com/tutorials/i2c
[26]
https://en.wikipedia.org/wiki/Serial_Peripheral_Interface
[27]
https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi

Section 4

[1]
https://www.standardslearn.org/coursedetails.aspx?key=44
[2]
https://en.wikipedia.org/wiki/American_National_Standards_Institute
[3]
https://www.allaboutcircuits.com/news/ipc-standards-the-official-standards-for-pcbs/
[4]
https://en.wikipedia.org/wiki/IEEE_802.11
[5]
http://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2016-2017/#GC3
[6]
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2017/n4659.pdf
[7]
https://www.mpoweruk.com/standards.htm#lithium
[8]
https://en.wikipedia.org/wiki/Solder
[9]
https://en.wikipedia.org/wiki/ESP32
[10]
https://www.law.cornell.edu/cfr/text/15/part-272
[11]
https://www.ieee.org/about/corporate/governance/p7-8.html

[12]
http://stroustrup.com/JSF-AV-rules.pdf
[13]
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1309986

Section 5: Datasheets

Holtek 3.3V Voltage Regulator:
https://datasheet.lcsc.com/szlcsc/Holtek-Semicon-HT7333-A_C21583.pdf

Great Power Battery Co. Lithium 18650:
https://cdn.sparkfun.com/datasheets/Prototyping/ICR18650%202600mAH%20datasheet.pdf

NanJing Top Power ASIC Corp TP4056:
https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf

Referenced

Developed by the Space Electronic Assemblies J-STD-001 Addendum Task Group (5-22as) of the Assembly & Joining Processes Committee (5-20) of IPC (2010) *Joint Industry Standard Space Applications Electronic Hardware Addendum to IPC J-STD-001E Requirements for Soldered Electrical and Electronic Assemblies*, Bannockburn, Illinois

Section 5

[1]
https://oscarliang.com/bjt-vs-mosfet/
[2]
https://www.eetimes.com/document.asp?doc_id=1278970
[3]
https://lastminuteengineers.com/esp32-arduino-ide-tutorial/
[4]
https://docs.zerynth.com/latest/official/board.zerynth.doit_esp32/docs/index.html
[5]
https://randomnerdtutorials.com/esp32-pinout-reference-gpios/

Section 6

[1]
http://www.substech.com/dokuwiki/doku.php?id=shore_durometer_hardness_test
[2]
https://www.smooth-on.com/page/durometer-shore-hardness-scale/
[3]
https://3dinsider.com/tpu-filament/
[4]
https://3dinsider.com/abs-3d-printing-filaments/

[5]
https://3dinsider.com/pla-3d-printing-filaments/
[6]
https://3dinsider.com/petg-filament/
[7]
https://knowledge.autodesk.com/support/fusion-360/troubleshooting/caas/sfdcarticles/sfdcarticles/System-requirements-for-Autodesk-Fusion-360.html

Section 8

[1]
https://www.elecrow.com/pcb-manufacturing.html?find=0&layer=2&size_x=100&size_y=100&quantity=5&bcountnum=5&designs=1&land=1.6&color=Green&charcolor=White&spray=HASL&bankong=0&copper=1&insidecopper=&sendtime=7
[2]
https://jlcpcb.com/quote?layer=2&width=100&length=100&thickness=1.6&qty=5&pcbColor=Green&surfaceFinish=1&surfacefinish=1&copperWeight=1&materialDetails=0&goldFingers=0&panelByJLCPCB=&differentDesign=1&country=US
[3]
http://www.basicpcb.com/
[4]
https://www.7pcb.com/pcb-quote-online.php?c5=5&c1980=10&e1980=10&dimension=cm&c9=0.062&c11=2&c13=1&c15=Green&c17=1&c19=1&country=USRT&turnaround=1&send=Calculate&x=83&y=11
[5]
https://www.pcbnet.com/Quotes/Board/Result/142547
[6]
https://www.electronicdesign.com/industrial-automation/what-s-difference-between-2-and-4-layer-pcbs
[7]
https://electronics.stackexchange.com/questions/56554/what-are-the-advantages-and-disadvantages-of-thinner-pcb-thickness-1-6-mm-or-0
[8]
https://www.epectec.com/articles/pcb-surface-finish-advantages-and-disadvantages.html
[9]
http://www.tag-connect.com/

Referenced

PCB Basics, Written by Mike Hord (SFUptownMaker)
https://learn.sparkfun.com/tutorials/pcb-basics

10.1 Permission

University of Waterloo Request

Permission to Use Figure/Information

Peter Chieh <cchieh@uwaterloo.ca>

Today, 7:38 PM

OK,
You may use it.
CC

Karlie Brinthaupt
Today, 2:12 PM
cchieh@uwaterloo.ca

Reply all

My name is Karlie Brinthaupt and I am an electrical engineering student at the University of Central Florida. I would like to request the use of a figure and information from the webpagehttp://www.science.uwaterloo.ca/~cchieh/cact/c123/battery.html for a design paper for a project I am working on for school. Any figures used as well as information referenced will be cited. Thank you!

Instructables Request

My name is Karlie Brinthaupt and I am an electrical engineering student at the University of Central Florida. I would like to request the use of some figures and information from the webpage https://www.instructables.com/id/Duino-Tagger/ for a design paper for a project I am working on for school. Any figures used as well as information referenced will be cited. Thank you!

Heat Sink Request

Permission to Use Figure

Karlie Brinthaupt
Today, 7:24 PM
docpetervis@gmail.com

Reply all

My name is Karlie Brinthaupt and I am an electrical engineering student at the University of Central Florida. I would like to request the use of a figure and information from the webpagehttps://www.petervis.com/electronics/Voltage_Regulator_Heatsink/Heatsink_for_TO-220_Voltage_Regulator.html

---

Heat Sink Voltage Regulator 7805 - Peter Vis

www.petervis.com

A heat sink is a piece of metal engineered to dissipate the maximum thermal energy into the ambient surroundings. It assists a component to remain below its maximum operating junction temperature by drawing this energy away, thereby preventing damage through excessive temperatures.

TI Request

Permission to Use Figure/Information

Karlie Brinthaupt
Today, 8:13 PM
univ@ti.com

Reply all

My name is Karlie Brinthaupt and I am an electrical engineering student at the University of Central Florida. I would like to request the use of figures and information from the TPS 63070 datasheet for a design paper for a project I am working on for school. Any figures used as well as information referenced will be cited. Thank you!

Karlie Brinthaupt
Electrical Engineering

Peer Advisor
Office of Honors Advising
UCF Burnett Honors College

Analog Devices Request

Permission to Use Figure/Information

Karlie Brinthaupt
Today, 8:17 PM
linda.kincaid@analog.com

Reply all

My name is Karlie Brinthaupt and I am an electrical engineering student at the University of Central Florida. I would like to request the use of figures and information from the ADP124 datasheet for a design paper for a project I am working on for school. Any figures used as well as information referenced will be cited. Thank you!

Karlie Brinthaupt
Electrical Engineering

Peer Advisor
Office of Honors Advising
UCF Burnett Honors College

Nanjing Request

Permission to Use Figure/Information

Karlie Brinthaupt
Today, 8:20 PM
tptopic@163.com

Reply all

My name is Karlie Brinthaupt and I am an electrical engineering student at the University of Central Florida. I would like to request the use of figures and information from the TP4056 datasheet for a design paper for a project I am working on for school. Any figures used as well as information referenced will be cited. Thank you!

Karlie Brinthaupt
Electrical Engineering

Peer Advisor
Office of Honors Advising
UCF Burnett Honors College

IPC Request

Permission to Use Figure/Information

Shannon Fies
Mon 10/22/2018 10:17 PM
To: HR@ipc.org

Reply all

Hello,

My name is Shannon Fies and I am an Electrical Engineering student at the University of Central Florida. I would like to request if I could reference some figures/information from the www.ipc.org website, namely the IPC-SpecTree 2016 PDF and any of the Free Documents [http://www.ipc.org/ContentPage.aspx?pageid=IPC-Free-Documents], for a design paper on a project I am working on for my school. Any figures used as well as information referenced will be cited.

Thank you and have a nice day!

Shannon Fies
shannon.fies@knights.ucf.edu

Last Minute Engineers Development Board Disclaimer

https://lastminuteengineers.com/disclaimer/

Last Minute ENGINEERS

ARDUINO PROJECTS ESP32 PROJECTS ESP8266 PROJECTS

Home › Disclaimer

Disclaimer

The information contained in this website is for general information purposes only. The information is provided by Last Minute Engineers and while we endeavor to keep the information up to date and correct, we make no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability or availability with respect to the website or the information, products, services, or related graphics contained on the website for any purpose. Any reliance you place on such information is therefore strictly at your own risk. Every material on this website may not be the original idea of the webmaster and may have been borrowed ideas and/or submitted by visitors.

The information provided on the LastMinuteEngineers.com may be used, copied, remix, transform, build upon the material and distributed for any purposes **only if provided appropriate credit to the author and link to the original article**.