Wirelessly Connected Laser Shooting Gallery

Group Information

Group Number: 22

Class: Senior Design EEL4915-22SPRING 0001

Advisor: Dr. Samuel Richie

Sponsor: Smart Charging Technologies LLC

Due Date: April 26, 2022

Group Members

Jamauri Balzourt: Electrical Engineering

Rachel Goodman: Electrical Engineering

Anna Malaj: Computer Engineering

Thomas Stoeckert: Computer Engineering

UCF

COLLEGE OF ENGINEERING AND COMPUTER SCIENCE

Table of Contents

List of Figures

List of Tables

1. Executive Summary

People in America, especially kids, teens, and young adults, love to play shooting games. Shooting gallery-type games are extremely common and popular around the country at theme and amusement parks, arcades, state and county fairs, and other similar entertainment venues. People experience a thrill when they hit the targets, and they want to play the games over and over again to hone their skills and compete against their friends.

Unfortunately, these experiences usually cost money, and they are limited to the aforementioned types of venues, which people may not always be able to travel to. Additionally, the targets for these games are often hardwired and fixed in place, and people have limited attention spans, meaning that there are only so many times a person can play them before they are bored, or before they have them so well-memorized that they become too easy.

Video games are usually seen as the at-home alternative to this experience. However, the average shooting video game often involves sitting still indoors for hours at a time staring at a screen, and little to no creativity, imagination, or real tangible feeling of accomplishment. While this is totally fine in small doses, it is much healthier for people, especially when they are young and their minds and bodies are still growing, to move around as much as possible, as well as to use their imaginations and creativity while having fun.

If there was a system that could bring the real shooting gallery experience into a user's home in a smaller, more portable, affordable, and easily reconfigurable form, it could provide hours of entertainment to young people that does not always involve staring at a screen and motivate them to move around a bit as well.

This is where the Wirelessly Connected Laser Shooting Gallery comes in. By building this project, we hope to create a source of entertainment that is fun, satisfying, endlessly replayable, and a good creative outlet for its user. The system involves a single laser "gun" controller and multiple portable targets, which are all wirelessly connected. By allowing the user the freedom to move and set up the targets wherever they desire, and providing multiple gameplay mode options, any place the user wants can become a laser shooting gallery, and the only limit on the fun is the user's own imagination.

This project was accomplished by designing and building one laser "gun" controller and multiple portable targets for the laser to hit. The gun and each target all have their own microcontrollers and PCBs for communication, display, and game tracking purposes. The gun "fires" a visible laser (so the user can see where they are aiming), and a light sensing component (a phototransistor) is located inside each target to pick up a successful hit.

2. Project Overview

2.1 Project Motivation

Typical proprietary shooting galleries have limited reconfigurability outside of their core structure. Most existing systems rely on fixed, hard-wired points that can typically only be set up in expensive, permanent installations, limiting both configuration possibilities and market reach. However, with the recent explosion in low-power and low-cost wireless computing, we believe that a solution involving multiple discrete, self-contained, low-cost wireless targets alongside a custom-designed "gun" controller and software suite would allow for increased user convenience, more interesting user experiences, and additional features that traditional systems are unable to achieve.

2.2 Project Objectives and Goals

In this paper, we propose a new design for a laser shooting gallery that is meant to improve upon traditional gallery systems. The purpose of this project is to create a wireless laser shooting system that is reconfigurable, expandable, and convenient to use, with the goal of creating a better user experience.

2.3 Project Description

2.3.1 The "Gun" Controller

The main controller for the system is a single device, styled after a traditional laser tag gun or pistol. Its core functionality is two-fold; it serves as the master controller for the entire game system, organizing and managing game data and device behavior for the various targets in the system, and it also serves as the main user interface for the system, with a traditional trigger input activating its laser diode. In addition, a simple status screen on the controller informs the user about the game state or device status.

A relatively modest microcontroller with built-in wireless capabilities was chosen for this role. While managing game and device statuses over a wireless network can be complicated, it's something that doesn't require a massive amount of computing power. It's mostly event-based, and as such, was designed to be energy and computationally efficient.

All these components require power, and as such, are powered via an onboard battery. This battery requires a chip to manage its status, as well as to provide some utilities for charging and discharging it. The battery is also able to be recharged while inside of the device.

2.3.2 One or More Target Devices

Targets are a critical component of a shooting gallery setup. The core functionality of a target, in this system, is to communicate with the main controller and wirelessly signal a successful hit whenever it receives a hit from the laser diode. This functionality is expanded with a display (through an LED array) in order to convey to the user some feedback that a hit has been registered and display extra information about the game state. These are all managed by an on-board microcontroller.

The microcontroller here also requires wireless capabilities, to signal to the main controller that a hit has been registered. While it is not managing game state, but rather acting as a follower to the controller device, it still requires some more advanced computing resources in order to drive the onboard display.

As with the controller, a target device contains a battery alongside a battery management circuit.

2.3.3 Embedded Software and Game Control

There are quite a few interconnected parts to this system, and joining them all together to create cohesive gameplay and overall user experience is the role of the software running directly on the microcontrollers on each device. The software running on these devices not only manages the devices' state, reacts to input events, etc., but also handles networked events, like game state alterations. As there are two different levels of device in this system, each with vastly different responsibilities and behaviors, the controller and targets each have their own bespoke software packages.

The controller's software is the most complex, as it carries the responsibility of managing and configuring game states, validating the network status of the entire system, interfacing with the user through physical controls, and overseeing its own system management (battery, laser shooting, etc.).

2.3.4 Game Modes

Providing a small collection of interesting, fun, and highly "replayable" game modes as part of the software was critical to the success of the final, larger product. The game modes that come with the system are described below. Playtesting was required to fine-tune the timing and the visual output for each game mode.

Time Trial

How many targets can you clear before time runs out? All targets light up and must be shot by the user. Once all are out, they all light up again. A timer on the controller's display informs the player how long he has left. The total number of targets hit is tallied and displayed at the end.

Whack-A-Mole

Like the Time Trial, Whack-A-Mole is a game mode in which the player must shoot targets as they appear – however, only a few targets light up at a time, and turn off if they're not hit fast enough. The player must be quick and accurate to score high.

Horde

Targets light up one by one, and multiple targets can be active at once. If a target is left on for too long, the game is over. As the game progresses, targets stay active for shorter periods of time. Scoring is evaluated by how long the player lasted and how many targets were shot.

One-Shot

This game is about accuracy. The player has three shots, which refill each time they successfully hit a target. Only one target illuminates, and the player has unlimited time to shoot it. Shoot the target, and another one lights up. If the player misses all three shots, they lose.

2.3.5 Project Stretch Goals

We have identified a number of stretch goals that we would like to add to our system in the future. These stretch goals are as follows:

Charging Dock for Controller and Targets

Instead of needing to plug in the controller and each individual target to charge, it would be extremely convenient for the user (as well as for us while we are testing it) if we could create a charging dock that could charge all of them at once. This would be similar in design to a USB hub or power strip, but it would be built to fit the specific dimensions of the controller and the targets, so that they can be conveniently placed and stored on it.

Multiple "Gun" Controllers

Adding additional controllers would allow for the game to be played by multiple people at once. Players could choose to either team up to take out targets together or to compete against each other to see who hits the targets first. This feature leans into the more social aspect of playing the game at home with friends, or at a party or other group function. It also increases the replayability of the game and the amount of time it is played in one session of use by taking advantage of the competitive nature of humans. People will often play a game as many times as it takes to win, especially when competing against friends.

2.4 Requirements Specifications

The minimum technical requirements for the system are listed below in Table 1. The requirements highlighted in blue are those that were demonstrated to our faculty panel. Our goal was to meet all of these requirements in our final design, so we aimed to make them as realistic as possible to avoid having to adjust them. Once the final prototype was built, a number of tests were performed on it to see if it met these requirements, and adjustments were made to the project to ensure it fell within our specifications.

Number	Purpose	Description	Requirement
1	Performance	The controller should have a high active uptime	$\geq$ 3 hours
2	Performance	Each laser target should have a high active uptime	$\geq$ 6 hours
3	Performance	The time between pulling the trigger and the target's visual response should be small	$\leq$ 0.1 seconds
4	Performance	The system should have a quick startup time	$\leq$ 25 seconds
5	Performance	Target pairing should be completed in minimal clicks	$\leq$ 4 clicks
6	Energy	The controller should be energy efficient, consuming a small amount of current	$\leq$ 3000 mA/4 hours
7	Usability	The controller should achieve a low maximum weight	$\leq$ 10 lbs.
8	Usability	The target should achieve a low maximum weight	$\leq$ 5 lbs.
9	Usability	The system should be in "ready to play" state within a short period after startup	$\leq$ 2 minutes

Table 1: Technical Requirements

2.5 House of Quality

The House of Quality is a matrix that assists with the product planning process. It is usually used by companies to relate the requirements of the customer to the more technical requirements of the company designing and producing the product.

On the House of Quality for this project specifically, the “customer requirements” have been converted to “marketing requirements,” since this project does not have a direct customer. However, these “marketing requirements” were created with the needs and desires of the average expected consumer of this laser target gallery in mind, in case the group should ever decide to try to sell the final product or idea.

The House of Quality for this project is shown in Figure 1 below [1].

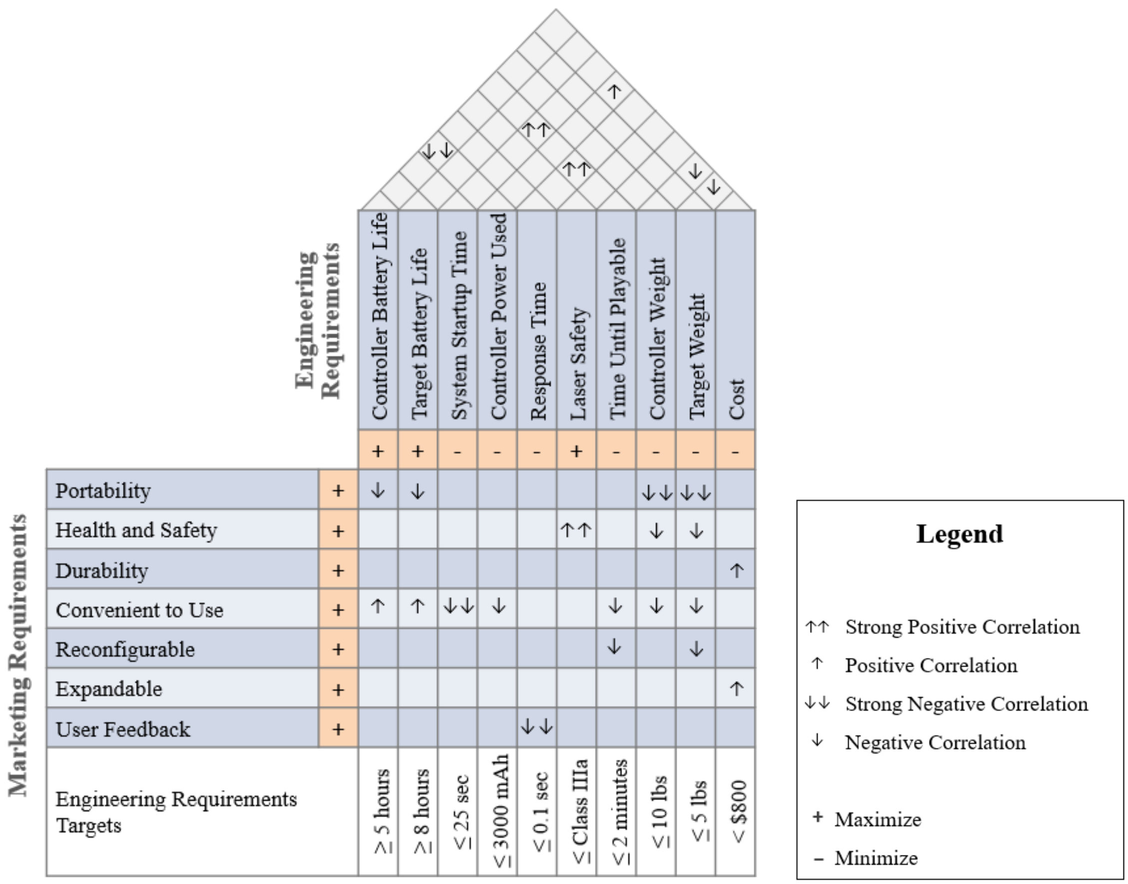


Figure 1: House of Quality [1]

2.5.1 Marketing Requirements

To determine the marketing requirements of this system, the best course of action was to consider the possible user(s) of this laser target gallery. The expected user would likely be a child, teen, or young adult. The primary demographic of laser tag players is between 6 and 14 years of age, although adults will also participate with their kids, or at a corporate event, party, or group gathering [2]. While this system is different than standard laser tag, it is expected that it would be attractive to a similar demographic.

The marketing requirements chosen for this project and included in the House of Quality are broken down in the following paragraphs. The goal was for all of these requirements to be maximized.

Portability: One of the main advantages of this system (and what distinguishes it from other similar projects and products) is that it is designed to be set up and rearranged as often as the user desires. Therefore, both the targets and the gun need to be lightweight and easy to carry so that the user can set the targets in a number of different locations and at a number of different heights.

Health and Safety: User health and safety is a very important marketing requirement because any potential user will want to know that the system is safe to use.

Since this system involves a laser, the number one concern will be eye safety. The laser was selected at an output beam power level and frequency that was not dangerous to the informed user and poses no greater risk than the average laser pointer.

Another important safety concern is battery safety. The batteries chosen to power both the targets and the gun should not ever be at risk of overheating to such a level that the user or the system will be harmed by them.

Durability: Since the expected user demographic is young, and since the gun and the targets are designed to be moved around quite frequently, it is important that the system be durable. The system should be able to handle being dropped, bumped, and otherwise jostled as the user is moving it around, and it should also be able to handle a reasonable range of different (not extreme) temperatures (both during use and storage).

Convenient to Use: This marketing requirement is for an overall positive user experience. The system should not be a hassle or chore to set up and take down, and it should be easy for any user to start it up and begin playing, with very little prior planning.

Reconfigurable: The entire system should be easily reconfigurable, meaning the targets can be moved around and the gameplay mode can be changed at any time the user desires.

Expandable: In order for the system to be expandable, the user should be able to add in new targets and/or remove existing targets at any time.

User Feedback: The system should provide adequate user feedback. The gun and targets should provide visual feedback to the user whenever a target is hit.

2.5.2 Engineering Requirements

The engineering requirements chosen for this project and included in the House of Quality were selected from requirements mentioned briefly in the “Project Description” and “Requirements Specifications” sections of this document. They are expanded on in the below paragraphs.

Controller Battery Life & Target Battery Life: In order to allow for maximum playable time for the user without needing to recharge, the batteries in both the “gun” controller and the targets should be able to last for multiple hours while constantly in use. Specifically, the goal is for the gun to have a battery life of at least 5 hours, and the targets should have a battery life of at least 8 hours. This time should be maximized as much as possible.

System Startup Time: The system starting up should take no more than 25 seconds. The less time the user has to wait for startup, the more the user is satisfied with the experience. This time should be minimized as much as possible.

Controller Power Use: The goal is for the controller to be as energy efficient as possible, and therefore it should consume very little power. It should require no more than 3000 mAh. This amount should be minimized.

Response Time: For accurate scorekeeping and maximum user satisfaction, a quick response time is necessary. Once the user has fired the laser, assuming the target is hit, the user should receive feedback from the target that clearly communicates the successful hit within 0.1 seconds of the laser being fired. Because the human visual response time is 0.25 seconds on average, with this response time, the game does not appear to the user to be lagging [3]. This time should be minimized as much as possible.

Laser Safety: As mentioned in the marketing requirements, laser safety for the eyes of the user and anyone nearby is extremely important. The FDA classes of visible lasers are Class II, Class IIIa, Class IIIb, and Class IV. Class II and Class IIIa are considered low risk for eye injury hazards, while Class IIIb and Class IV are increasingly higher risk, so only a Class II or Class IIIa laser will be used in this system [4]. Laser safety should be maximized.

Time Until Playable: Following the system startup, the time it takes until the system is "ready to play" (meaning the gun is paired to the targets and the user can begin gameplay) should be no more than 2 minutes. This time should be minimized as much as possible.

Controller Weight and Target Weight: As mentioned in the marketing requirements, the gun and targets should be as lightweight as possible to the system is portable and easy to reconfigure. Specifically, the controller should be no more than 10 pounds and the targets should be no more than 5 pounds each. These weights should be minimized.

Cost: The component costs, assembly cost, and testing cost of this system is expected to total up to just under $800. This amount should be minimized as much as possible to achieve a low-cost design. This is one of the more flexible engineering requirements, however, because the project is not held to a strict budget.

2.5.3 Marketing and Engineering Requirement Correlations

When designing a system, it is important to see where its engineering and marketing requirements align, as well as where they contradict one another. The requirements for which they align can be left as they are, but for the contradicting requirements, compromises may have to be made in the design to adequately satisfy both engineering and marketing requirements. The center grid of the House of Quality is used for these comparisons. The engineering requirements make up the columns and the marketing requirements make up the rows. The arrows in the boxes where they intersect indicate if there is a positive correlation or a negative correlation or no known correlation between them, and the quantity of arrows in the boxes indicate the strength of the correlation. In this case, a positive correlation means that the two requirements are directly proportional, whereas a negative correlation means that they are inversely proportional. The correlations shown in Figure 1 will be explained in the following paragraphs.

Correlations with Portability: Portability has negative correlations with both Controller and Target Battery Life. This is because as the battery life is extended, more or bigger batteries may be required, which will lead to the targets and the controller becoming heavier and/or larger, and therefore less portable. Since more weight is obviously detrimental to portability, Controller and Target Weight both have strong negative correlations to Portability.

Correlations with Health and Safety: Health and Safety has a strong positive correlation with Laser Safety, as increasing the safety of the laser increases the overall safety for the user. It also has a negative correlation with Controller and Target Weight, because as the weight of these items increases, the less safe it will be for the user to carry or hold them, especially for long periods of time.

Correlations with Durability: There is a positive correlation between Cost and Durability because more durable materials will likely be more expensive to acquire.

Correlations with Convenient to Use: Convenient to Use has some correlation with almost all of the engineering requirements, since they are already geared towards convenient use. It has positive correlations with engineering requirements that should be maximized, and negative correlations with engineering requirements that should be minimized.

Correlations with Reconfigurable: Reconfigurable has negative correlation with Time Until Playable because the longer it takes for the system to get into "ready to play" mode, the less easily reconfigurable it will be. It also has negative correlation with target weight because if the targets are heavy, it will be harder to move them around to "reconfigure" the game.

Correlations with Expandable: Expandable has a positive correlation with cost because increased expandability means more targets, which means more cost from materials and assembly.

Correlations with User Feedback: User Feedback has a strong negative correlation with Response Time, because the less time it takes the system to respond to a target hit, the better the user's feedback experience.

2.5.4 Engineering Requirement Correlations

The "roof" of the House of Quality is used to show any correlation between the engineering requirements, which uses the same arrow system to show correlation. The correlations shown in Figure 1 are described in a bit more detail below.

Controller Battery Life and Power Used: This is a strong negative correlation because the more power the controller uses, the less time its battery will last before a recharge.

System Startup Time and Time Until Playable: This is a strong positive correlation because the system startup time makes up a portion of the time until playable.

Response Time and Time until Playable: This is a strong positive correlation because the response time makes up a portion of the time until playable.

Controller Power Used and Cost: This is a positive correlation because a higher-power controller is likely to have more expensive components.

Controller and Target Weight and Cost: This is a negative correlation because the more lightweight parts used in the controllers and targets, the more the components and system housing may cost.

2.6 System Block Diagrams

This section contains the overall block diagrams which describe the design of the hardware and software systems involved in this project. They also list the group member who was responsible for ensuring that each subsystem was completed and worked as it should.

2.6.1 Hardware Block Diagram

The overall high-level block diagram for the hardware involved in this project is shown in Figure 2 below. In the diagram, most of the hardware blocks have been broken down into two distinct layers: the “Feedback Layer” and the “Power Delivery Layer.”

Each layer had one or more group members assigned to take responsibility for it. These assignments were made based on each group member’s individual preference, as well as what areas of the technology they have the most experience with.

The “Feedback Layer” encompasses the hardware that emits the laser from the “gun” controller when the trigger is pulled, the hardware that receives the laser signal in the targets, and the Wi-Fi modules that the gun and the targets use to communicate with one another.

The “Power Delivery Layer” contains the battery with its internal battery management system (BMS) and the DC output from the battery to the device, for both the gun and the targets.

The remaining blocks that are not in the layers are the main microcontrollers for both the laser “gun” and the target, as well as their visual displays. The responsibility for these blocks was assigned to all group members, as they are vitally important to every aspect of the project and therefore all group members were heavily involved in their design and implementation.

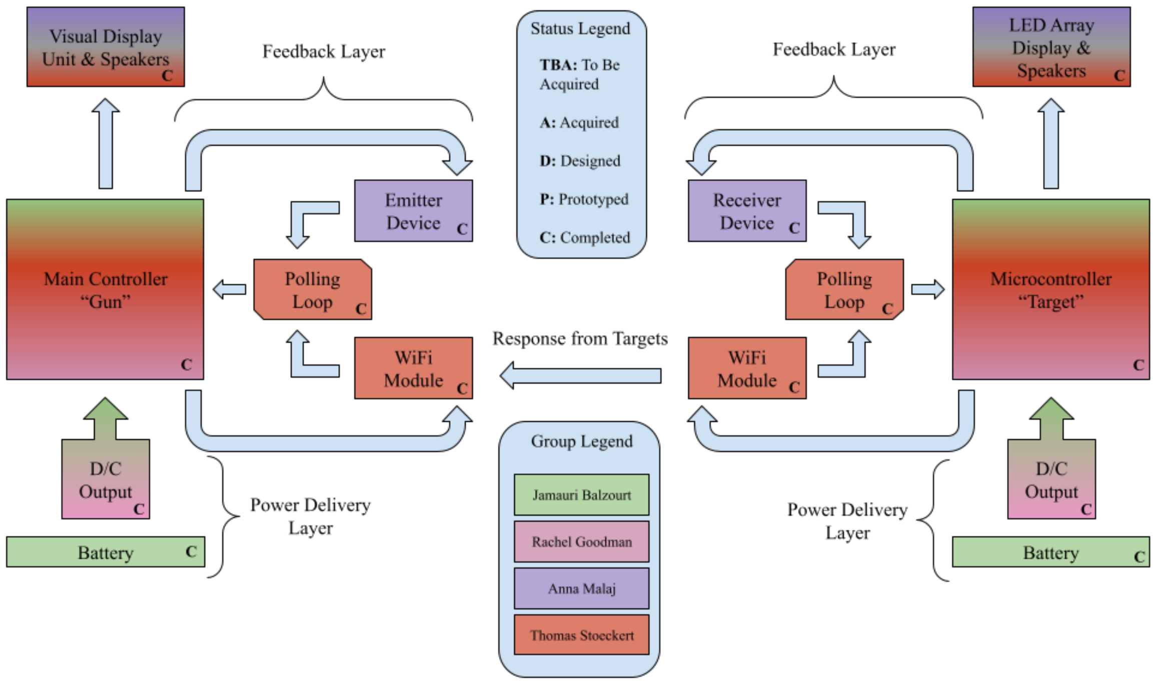


Figure 2: Hardware Block Diagram

2.6.2 Software Block Diagram

While the specifics of our software vary between the target and controller devices, the overall structure is preserved, allowing for the software team to develop one "core" software package that can be extended and reconfigured depending on the hardware it is being deployed to. To that end, our software block diagram showcases the critical "core" components – software modules shared between each platform – and denotes when our modules vary for a target platform. We also show the division on our system between two classes of software component: Managers and Modules.

When the system powers-on, each "manager" is given an opportunity to initialize. Most managers use this to establish long-lived interface settings (pin numbers, hardware control information, library classes), establish default values, and register any relevant callbacks or system tasks to the system scheduler.

Once all the startup steps are completed, the software falls into the core, which is handled via the real-time task scheduler that powers the system. Fixed-frequency updates are given to the managers that request them, and they in-turn issue callback events to modules that register interest. The active "module" is selected via the StateManager, which is responsible for switching between modules as required by the state of the system. The active module is given an opportunity to perform its fixed-frequency update, as well as callbacks for wakes/sleeps as required. As our system varies its response to events based upon what operating state it is in, splitting our primary functionality between nuclear states means that our code is cleaner, easier to maintain, and can be developed in parallel between multiple developers.

The “Ready” module holds the primary menu / user interface for the Controller device, and was assigned to Anna, due to her familiarity with user interface development processes. On the Target device, the “Ready” state serves as an idle state, waiting on network events to change its state away to an active “Play” state.

In the “Play Module,” we bundle all of the logic for how the gun or controller behaves while playing games. It’s the module where users spend the majority of their time, and as such, required a concerted development effort between both developers in order to create and polish. Our “Play Module,” too, was truly split into each of the dedicated game modes, however, they perform logically consistently.

The “Results” state is also very straight-forward and somewhat static. This state doesn’t actually exist on the Target device, just the Controller, in which the controller displays to the user their score from the last round of play. Due to its standard, simple structure, this can easily be handled by one developer. The focus on the user interface means that Anna was the best fit for this section of the software development.

While the “Pairing State” is the second-most complicated state to manage, it was still remarkably less complicated that the play state, meaning it can be handled by one developer. This state handles the management of connecting to and “pairing” with other targets. As this is more low-level and focused on wireless communication, Thomas was the the primary developer of this block – though it does include some User Interface elements, which Anna contributed to the development of.

After the event has been handled by the system’s current state, the microcontroller returns to an idle state. Our system actually is managed through a real-time scheduler, instead of a standard for-loop. This means that functionality and handling of events is managed through tasks and jobs, which are – in the end – set up to function identically to what is listed below. However, the task / job structure means that critical events, such as audio playback, wireless message communication, and other functionality, is automatically balanced with other computation requirements. This means that peripherals which rely on strict timing protocols will not be starved for CPU time during the course of use.

The sections, states, and events described above are documented in much greater detail in section 5.4 Software Design. The block diagram for the software involved in this project is shown below, in Figure 3.

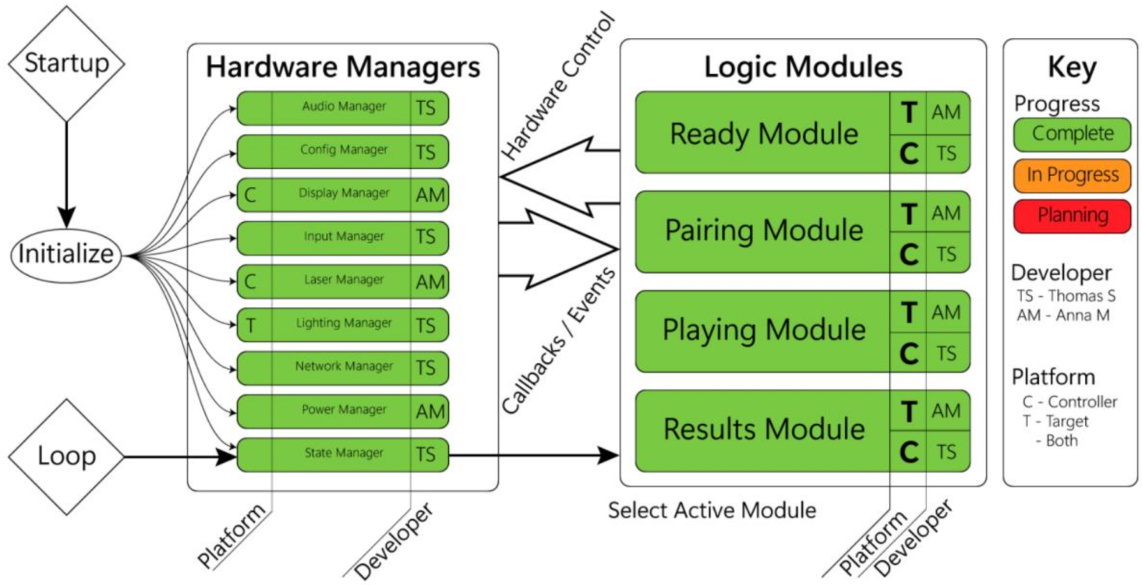


Figure 3: Software Block Diagram

3. Research and Part Selection

3.1 Existing Similar Projects and Products

From traditional shooting galleries maintained by large theme parks to in-home training and entertainment systems, the industry of laser shooting technology spans a wide array of uses, designs, and experiences. In this section, we look at both senior design projects and systems currently being sold on the market to determine what worked well and what can be improved in our design.

3.1.1 Modern Dry Fire Laser Training

While the industry of laser shooting galleries is fairly mature, companies still continue to develop innovative ways to improve the user's experience. LaserHIT is a company that specializes in creating laser shooting technology for at-home firearm practice. Their product works by allowing users to add a laser training cartridge to their own firearm. Users can practice by shooting a paper target, which sends training information to an app for the user to track. Users can choose to practice in one of multiple modes that specifically target key skills like accuracy and speed. Aside from just proving that laser shooting technology can be useful and successful on the market, LaserHIT offers insight into ways the user experience of our project can be designed. LaserHIT's product offers a remote restart option, which allows the user to reset their progress without needing to walk to interact with the app. Our implementation provides a similar mechanism, in which a user can restart a game directly from the controller, allowing for the most convenient interaction between the game and the user. LaserHIT's product differs from ours in that user scores and game mode choices are displayed on a mobile app. Our implementation provides visual effects to the user through the physical targets and gun, ensuring that the user can interact with the system without a dependency on a mobile device. [5].

3.1.2 Battle Action Laser Tag

Another company that has brought laser shooting technology to the market is Squad Hero. Their product, Battle Action Laser Tag, is a kit containing multiple infrared laser guns and vests for playing laser tag with friends. While each set comes with four guns and vests, more sets can be paired together to add more players to the game. The kit also includes a charging station for the guns and vests, allowing the user to play for about four hours at a time without having to change any batteries. The extensibility of this design is an aspect we included in our system. Our project differed from this design, however, in its process for starting new games. According to Squad Hero customers, Battle Action Laser Tag requires users to turn off and reconfigure the system each time they want to begin a new game. We improved this design by allowing users to conveniently restart games without needing to power off any of the subsystems [6].

3.1.3 Laser Target Gallery

Looking past what currently exists on the market, similar designs have been created by previous University of Central Florida Senior Design teams. One such design is the "Laser Target Gallery". This system, much like ours, involved creating a laser shooting gallery game. The design involved a standalone laser gun, a stationary target board, and a mobile application for tracking scores. The team used an Arduino microcontroller to communicate via Bluetooth between the target board and the mobile app. As this project was especially similar to ours, it provided useful insight into how our system could be designed. We improved on this design by introducing mobility and extendibility to the target board. While the Laser Target Gallery involves a stationary board, our design improved the user's experience by allowing for multiple mobile targets to be connected to the game at once, meaning we implemented a more robust and far-reaching communication system [7].

3.1.4 Let's Have a Blast! Laser Tag

Over the Fall 2018 – Spring 2019 Senior Design session, a project called "Let's Have a Blast" was developed at the University of Central Florida by a team of electrical and computer engineers. It was a laser tag system that utilized the "gun" controllers as both a laser gun component and as a per-user "target," meaning that the entire game system was composed of just a wireless network of laser tag guns. Users could just pick one up, turn it on, and join the game, as each "gun" was all a user needed to play.

This project was presented during the freshman year of the students of our group, with two members of our group knowing a member of this project personally. This project served as the initial inspiration for some of the form-factor and user experience considerations of our project.

The system had a relatively sophisticated software package as well. The system was based off of a wireless mesh network setup, using an ESP32 as the core microcontroller. Players could join teams and configure game settings. The system even had an OLED screen embedded inside of the controller to serve as a user interface. In addition, their hardware on-board the controller also included an accelerometer (to provide a "reload" function, by pointing the gun towards the ground), a "haptic" motor to provide tactile feedback, and a simple passive buzzer speaker.

Due to the complexity of the game experience, and the multi-user nature of the environment, detecting who shot what laser turned out to be an important component of their system. This was ultimately resolved by encoding user data inside of the laser shot – by flashing the laser on and off over time, they encoded information such as “team,” “user ID,” “damage,” and even error detection and correction information. It’s a remarkably clean solution for this problem, and while our system does not require such complexity – we only using one “gun” – it’s an interesting system to review [8].

3.1.5 Universal Studios Florida

Another multi-user laser shooting system is in use at the Universal Studios Florida theme park attraction, “Men in Black: Alien Attack,” where a set of up to 36 ride vehicles, each holding 6 laser-tag users at a time, simply allows only one laser to be shooting at a time. It cycles through all users in the span of milliseconds, resulting in no notable interruption to user experience, but results in a simple solution to the same problem. It, of course, relies upon precise timing and control between targets and “guns,”. This would be impractical for a wireless-based solution, but it’s interesting to see another approach.

3.2 Relevant Technologies

This section discusses a number of technologies related to the design of this project, gives a brief overview of how these technologies operate and what their purposes are, and connects them to their use (or how they were considered for use) in the project.

3.2.1 Battery Management System

The Battery Management System (BMS) is designed to protect the operator and the system from operating beyond its recommended specifications. The three basic topologies of BMS include: Centralized, Distributed and Modular topologies. Modular and Distributed designs will be repudiated with our focus being on Centralized designs.

- Centralized: This topology focuses on a chip carrying out the functionality of the BMS with sense wires extending from the chip itself. Depending on the level of robustness of the system, various safety measures may be included to preserve the health of the batteries such as overvoltage protection, overcurrent protection, undervoltage protection, temperature protection and cell balancing [9].
- The selected battery pack for this system contained a centralized BMS with overvoltage, overcurrent, and undervoltage protections for the entire pack. Advanced features such as cell-balancing and temperature protections were not required to safely meet the requirements specifications outlined previously.

3.2.2 Linear Voltage Regulator

Our design integrated linear voltage regulators at essential voltage levels to support the MCU, Laser Diode, Photosensor, Visual Display Screen and Target LED array, ensuring nominal operation and stability from voltage fluctuations. The regulator topology featured on-card regulation with internal current-limiting and thermal-shutdown features. The selected topology was non-adjustable so separate ICs were selected for 3.3V & 5V where required in designing the system. Figure 4 [10] shows the functional block diagram of the 3.3V linear fixed-voltage regulator selected for this project.

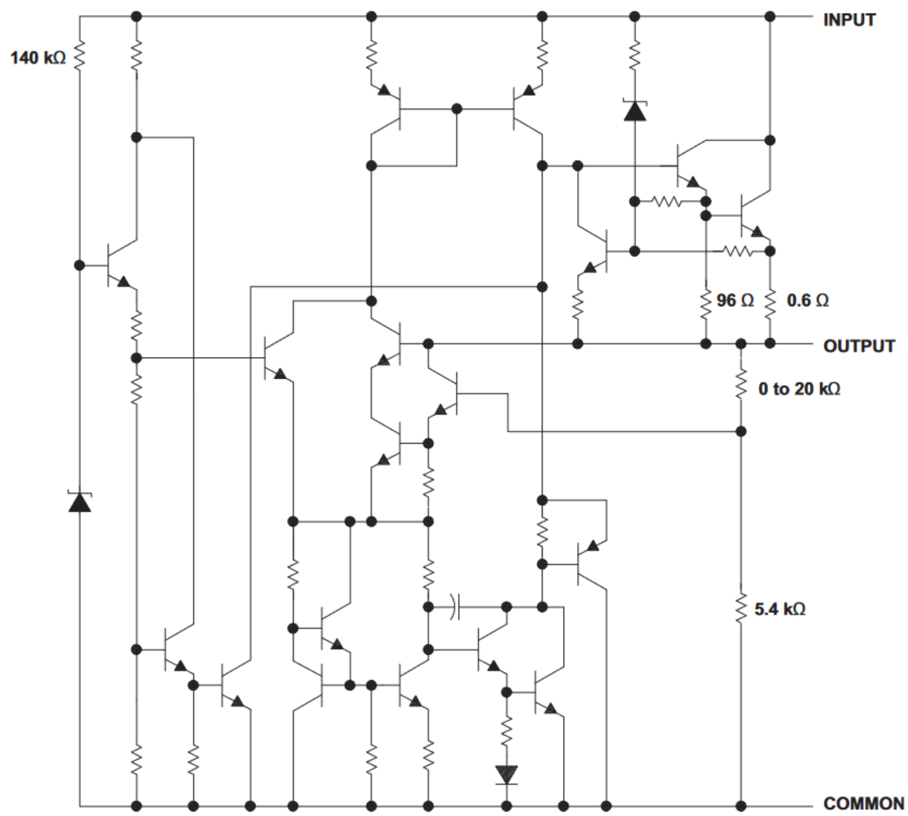


Figure 4: Functional Block Diagram of IC [Courtesy of Texas Instruments] [10]

Implementing a linear fixed-voltage regulator enabled the team to create an inexpensive, low-footprint design with stable voltage at reasonably high current outputs as compared to other Linear Voltage Regulator designs, which are typically $0.73 more expensive per IC.

3.2.3 Voltage Step Converters

The Voltage Step Converters covered in this section feature basic Switched-Mode Power Supply (SMPS) topologies and are non-isolated in design. Thus, there is no galvanic isolation between the output and input.

3.2.3.1 Voltage Boost Converter

The boost converter steps up a low input DC voltage to a high output DC voltage. The boost converter operates in two modes controlled via a switching MOSFET (some designs feature a transistor for switching). When the MOSFET is ON, as shown in Figure 5 [11], the inductor is being charged from the source voltage.

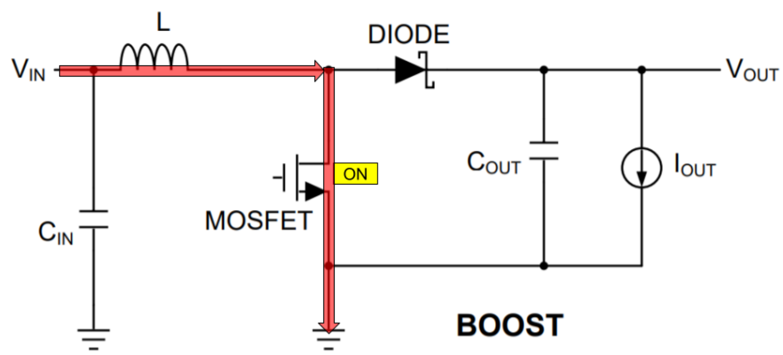


Figure 5: Boost Current Flow (MOSFET ON) [Courtesy of Texas Instruments] [11]

Subsequently, when the MOSFET is switched OFF, the source voltage and the energy stored in the inductor are released to the output voltage; this combined voltage is "Vout" at a higher level when compared to the source. During this operation, the capacitor in parallel is being charged. At the end of the switching cycle, the MOSFET switches ON. The capacitor is sized accordingly to maintain the output voltage enough for the next switching cycling while the inductor is being charged. Figure 6 [11] shows the direction of current flow for this switching operation.

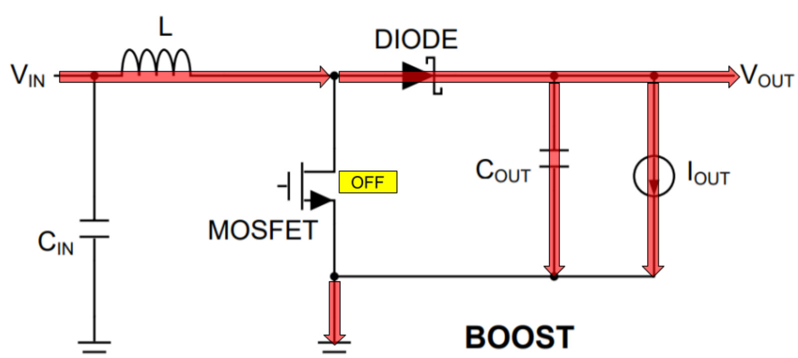


Figure 6: Boost Current Flow (MOSFET OFF) [Courtesy of Texas Instruments] [11]

3.2.3.2 Voltage Buck Converter

The buck converter steps down a high input DC voltage to a low output DC voltage. The boost converter operates in two modes controlled via a switching MOSFET (some designs feature a transistor for switching). When the MOSFET is ON, as shown in Figure 7, the inductor & capacitor are being charged from the source voltage [11]. The diode is in reverse-bias and blocks any return paths to the inductor.

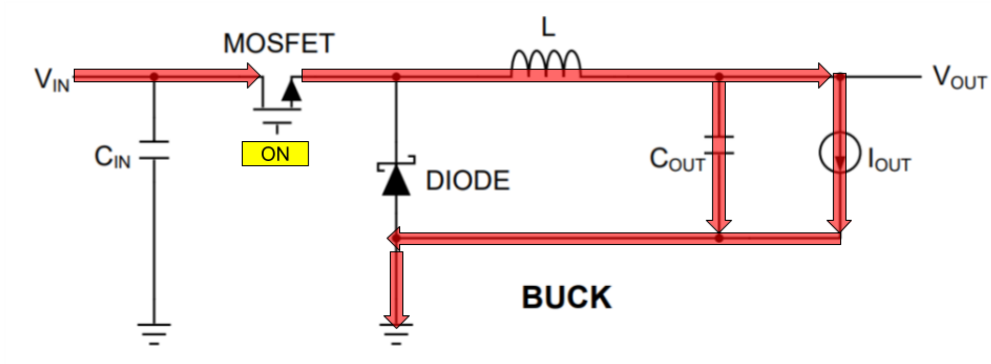


Figure 7: Buck Current Flow (MOSFET ON) [Courtesy of Texas Instruments] [11]

Subsequently, when the MOSFET is switched OFF, the source voltage is open, and the energy stored in the inductor and capacitor is released to the output voltage. The diode is in FB, thus allowing a return path from L1. Once the energy in L1 is near depleted, the capacitor will release its energy to maintain Vout at its nominal level. The inductor and capacitor must be sized accordingly to maintain the output voltage until the next switching cycle while the inductor is being charged. Figure 8 [11] shows the direction of current flow for this switching operation. At the end of the switching cycle the MOSFET switches ON.

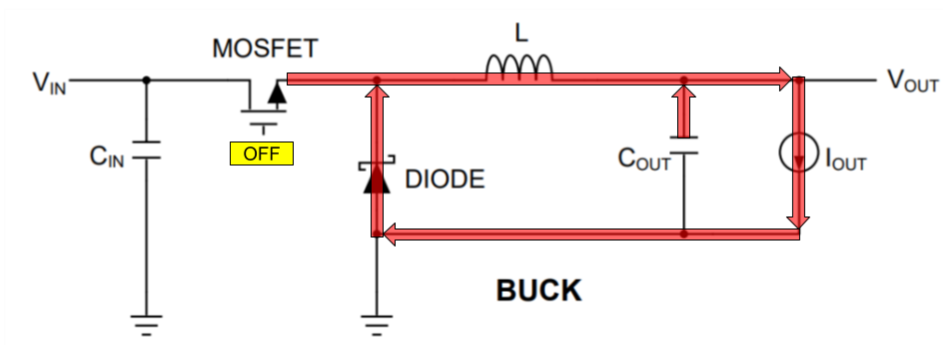


Figure 8: Buck Current Flow (MOSFET OFF) [Courtesy of Texas Instruments] [11]

A capacitor on the input may be required to filter out noise induced by the switching MOSFET to the input side of the converter.

3.2.4 MOSFETs

MOSFETs or Metal-Oxide-Semiconductor Field-Effect Transistors are three-terminal devices with the property of enabling precise control of the current flowing through the device. The current follows a path from the Drain (terminal D) to the Source (terminal S); modifying the potential of the Gate (terminal G) affects the amount of current flowing I_{DS} (Drain-to-Source Current). Figure 9 shows the basic structure of a MOSFET [12].

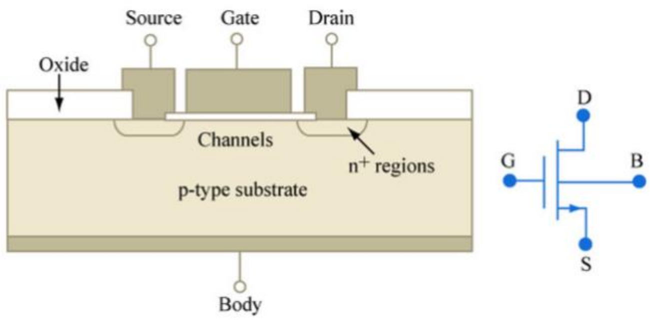


Figure 9: MOSFET Substrate Structure [12]

This device is one of the fundamental components of the ultra large-scale integration process for microchip fabrication [13]. Four fundamental MOSFET designs must be covered to fully understand its properties.

N-channel MOSFET & P-channel MOSFET

N-channel MOSFETs and P-channel MOSFETs are built in p-type & n-type silicon substrates respectively so that reverse-biased pn junctions isolate the conducting channels of nearby devices. N-channel MOSFETs with positive gate voltages having sufficient potential with respect to ground enables a conducting current. P-channel MOSFETs create a conducting current with sufficient negative potential with respect to ground. These are characteristics of an enhancement-mode MOSFET. Depletion-mode MOSFETS however, function in an inverted manner. At zero gate potential, the MOSFET is conducting current, at a sufficient positive or negative gate potential the MOSFET is OFF for enhancement-mode and depletion-mode MOSFETs respectively [14]. The main application of MOSFETs is voltage-controlled resistors in analog circuits or as ON/OFF switches in digital circuits. Applications of this characteristic are Voltage Controlled Current Source (VCCS), Microprocessors, Photosensor for laser detection, etc. Figure 10 summarizes the four fundamental MOSFET designs discussed above.

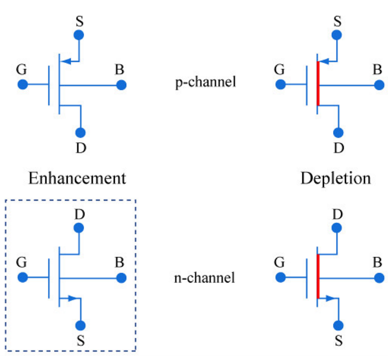


Figure 10: MOSFET Types & Modes [14]

3.2.5 BJT

A Bipolar Junction Transistor (or BJT) is a three-terminal device consisting of three separately doped regions with the property of enabling precise control of the current flowing through the device, much like the MOSFET. The current follows a path from the

Collector (terminal C) to the Emitter (terminal E); modifying the current of the Base (terminal B) affects the amount of current flowing. Figure 11 shows the basic structure of a BJT [15].

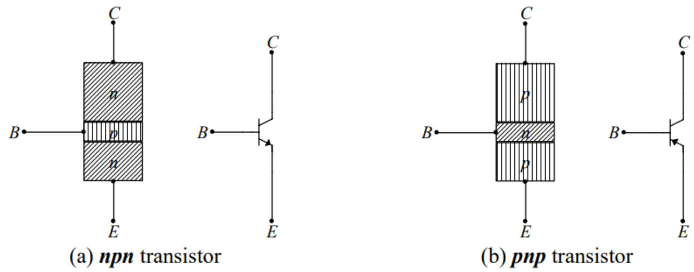


Figure 11: BJT Types [15]

This device is one of the fundamental components of the ultra large-scale integration process for microchip fabrication. Three fundamental BJT operating regions must be covered to fully understand its properties.

NPN transistor & PNP transistor

NPN & PNP describes the order of the regions and their respective bias. The output of a BJT is governed by a few properties: the current flowing through the Base terminal and the gain of the transistor, as well as the operating mode of the two junctions of the BJT: Base-Emitter junction and Base-Collector junction. When the operating mode is in cutoff region (Base-Emitter junction is reverse biased), no current can flow. For digital circuits, this is analogous to a switch that is open. In saturation region, both junctions are forward biased, and the output current is uncontrolled and may approach the limits of what the device can handle. For active region, the Base-Emitter junction is forward biased, and the Base-Collector junction is reverse biased. This region allows for the output current to be controlled as per the operator's specifications. Finally, breakdown region should be avoided since the BJT will be operating beyond the nominal specification of the device. Figure 12 summarizes all possible operation modes for the BJT.

NPN Transistor		
Mode	**E-B Junction Bias**	**B-C Junction bias**
Forward-Active	Forward	Reverse
Saturation	Forward	Forward
Reverse-Active	Reverse	Forward
Cut-off	Reverse	Reverse

PNP Transistor		
Mode	**E-B Junction Bias**	**B-C Junction bias**
Forward-Active	Reverse	Forward
Saturation	Forward	Forward
Reverse-Active	Forward	Reverse
Cut-off	Reverse	Reverse

Figure 12: BJT Summary

3.2.6 Laser Diode

LASER stands for Light Amplification by Stimulated Emission of Radiation. A laser diode is typically a semiconductor p-n junction structure that emits radiation by way of an excited forward current. The properties of a basic diode apply to Laser Diodes albeit with minor drawbacks. Thermal limitations originate from the small surface area of the Laser Diode. Figure 13 details the structure of a common PN junction Laser Diode.

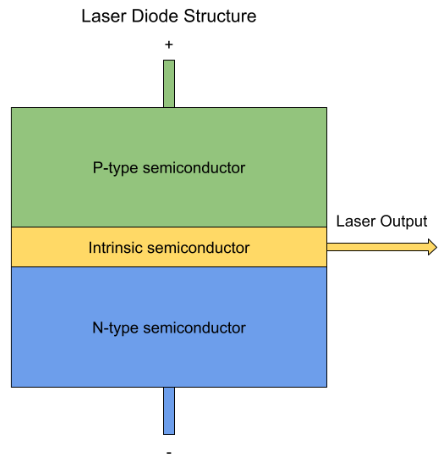


Figure 13: Laser Diode Structure [Diagram by Jamauri Balzourt]

Upon excitation the semiconductor experiences high current density due to the small area relative to the current flowing through the device. This small region results in significant heat generation (for high power Laser Diodes with low forward voltage). Physically, the Laser Diode must have an operating power limit constrained by the output mirror lens. Operating in any region beyond the output mirror lens may result in catastrophic damage rendering the device unusable [16].

All visible lasers operate based on an emitted wavelength range between ~400nm to ~700nm. Of course, some laser designs operate beyond 700nm+ in the infrared range of nonvisible light. Figure 14 [17] showcases the applicable wavelength described.

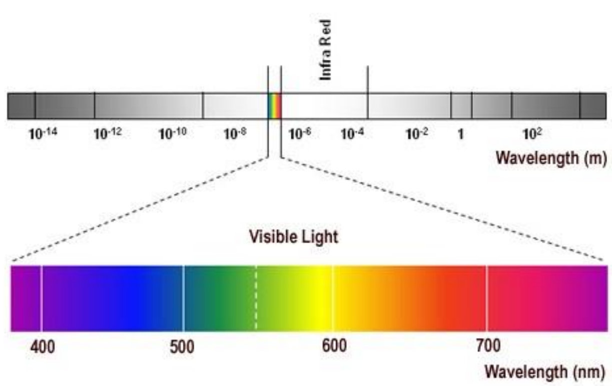


Figure 14: Wavelength Spectrum [Used with Permission] [17]

3.2.7 Light Sensing Technologies

A light sensor is a device that converts light energy (photons) into an electrical output. They are most commonly called "photoelectric" or photo-sensing" devices. The three most common devices used as light sensors are photoresistors, photodiodes, and phototransistors, all of which will be discussed in the following subsections [18]. In the Parts Selection area of this document, these three different devices will be compared, and one will be selected to use as the receiving sensor for the light coming from the laser "gun" controller.

3.2.7.1 Photoresistor

Photoresistors, sometimes called light dependent resistors (or LDRs), are devices often used to indicate the presence or absence of light, or to measure light intensity. Their resistance is an inverse (but nonlinear) function of light intensity. When they are in the dark, their resistance is at its highest, sometimes up to even 1 MΩ. Conversely, when they are exposed to light, depending on the intensity of the light, their resistance drops significantly, sometimes down to a few ohms. The more intense the light, the lower the resistance.

There are two types of photoresistors: intrinsic and extrinsic. Intrinsic photoresistors are made from undoped semiconductor materials such as silicon or germanium. When photons land on the device, electrons are excited from the valence band into the conduction band, creating more free electrons in the material to carry current, and therefore decreasing the resistance. Extrinsic photoresistors are made of doped materials, or dopants. These dopants form a new energy band above the existing valence band. This new band is populated by electrons. These electrons now need less energy to make the "jump" to the conduction band, because the energy gap is smaller. When photons hit the device, it is easy for these electrons to move to the conduction band and begin to carry current. The main difference between intrinsic and extrinsic photoresistors is the materials that they are made up of, which slightly alters how they operate. However, the result is still the same: a device that exhibits a decrease in resistance when exposed to light.

It is important to note that although photoresistors are still used in many applications, they are becoming less common due to the fact that almost all of them are made with lead or cadmium. This means that most photoresistors are not RoHS compliant, and they are banned in some countries due to concerns about their environmental impact. Their light-sensing functionality is now more often performed by other devices, such as photodiodes or phototransistors, which will be discussed next [19].

3.2.7.2 Photodiode

The photodiode operates by essentially converting optical light entering the semiconductor into a respective current based on the intensity of light received. Photodiodes can be made from a number of different semiconductor materials, including (but not limited to) Silicon, Germanium, and Indium Gallium Arsenide. The structure of a typical photodiode is of a P-N junction semiconductor similar in design to that of the Laser Diode. The p-type layer has excess holes, and the n-type layer also has excess electrons. A depletion region is formed from the diffusion of these excess carriers. This is a region in which no free carriers exist, which causes a built-in voltage to develop and create an electric field across it, which allows current to flow in only one direction, from anode to cathode. If a photodiode is forward-biased, the generated current will flow in the opposite direction. This means that most photodiodes are reverse biased, or not biased at all. Some photodiodes will be damaged if they are forward biased. Care must be taken to ensure that the diode is biased correctly to avoid damaging it.

When a photon with sufficient energy strikes an atom within the device, it releases an electron, which then forms an electron-hole pair. If the photon is absorbed into either the n-type or p-type region, the electron-hole pairs will be recombined as heat if they are at least one diffusion length away from the depletion region. On the other hand, photons that are absorbed into the depletion region (or near it) will create electron-hole pairs that will move to opposite ends of the region due to its electric field. Electrons will move toward the positive side (cathode because reverse bias), and holes will move towards the negative side (anode because reverse bias). These moving charge carriers create the current that the photodiode generates [20].

The basic properties of the diode apply to this semiconductor design along with additional drawbacks. The photodiode is highly susceptible to temperature deviations affecting its nominal operating characteristics. Additionally, all PN junction photodiodes contain dark current in the form of leakage current in reverse bias and increases linearly with temperature [21]. Figure 15 details the structure of a common PN Photodiode.

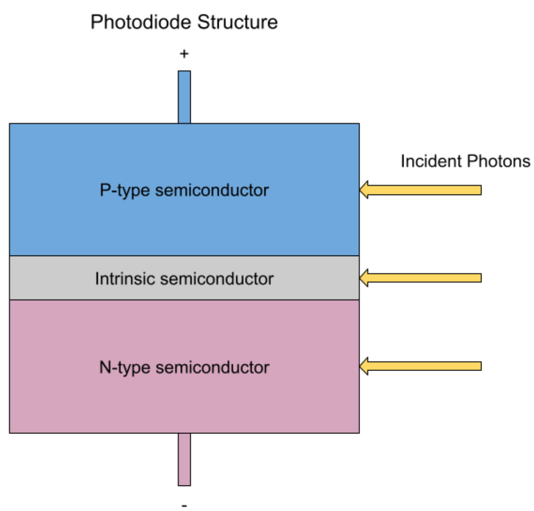


Figure 15: Structure of Photodiode [Diagram by Jamauri Balzourt]

The effectiveness of a photodiode is also determined by its effective Peak Wavelength Efficiency. This wavelength efficiency should be maximized for the expected light source and unwanted light either filtered out via a lens filter or through the efficiency curve of the photodiode. Figure 16 below shows what a typical PN junction photodiode Peak Wavelength Efficiency curve looks like.

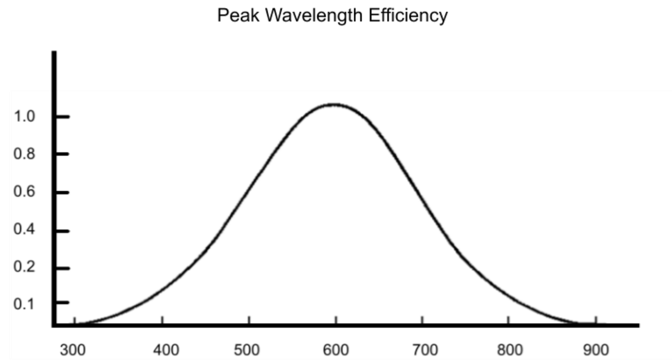


Figure 16: Wavelength Efficiency Diagram [Graph by Jamauri Balzourt]

3.2.7.3 Phototransistor

A phototransistor, much like a photodiode, is a semiconductor photojunction device. It is very similar to a normal transistor, except that it has a light-sensitive Base terminal (or, more accurately, a light sensitive Base-Collector Junction). Because the Base-Collector junction is the light-sensing part of the sensor, this junction is much larger on phototransistors than it is on regular transistors.

A phototransistor operates much like a photodiode, but with an amplifying transistor. Just like a photodiode, it converts photons into current, but it also provides a significant current gain. The light absorbed into the base of the phototransistor will induce a small current. This current is then amplified by normal transistor action, which results in a much larger current. When compared with a similar photodiode, the current generated by the phototransistor can be 50 to 100 times larger.

Phototransistors are essentially bipolar NPN transistors with a large Base-Collector junction, and therefore their characteristics are very similar to that of a simple Bipolar Junction Transistor (or BJT). They are available as both a two-leaded or a three-leaded device. For two-leaded phototransistors, the base terminal is made electrically unavailable, and the device is entirely dependent on light. The collector terminal is usually at a higher potential than the emitter in order to induce reverse bias at the Base-Collector junction. When there is no light to be absorbed by the phototransistor, it still has a small amount of dark current (or leakage current), just like a photodiode. When there is sufficient light being absorbed by the base terminal, a base current is produced, the amount of which is proportional to the intensity of the light. This base current will trigger the amplification process, which generates a collector current with a high gain. For three-leaded phototransistors, the use of the Base terminal is optional. When it is used, the phototransistor acts as a normal BJT, and when it is not used, it acts as a phototransistor [22].

3.2.8 Mesh Networking

Traditional wireless networking, as used in traditional domestic networks, consumer electronics, and in business settings, historically have relied upon a centralized structure. One primary controller, typically a wireless access point or traditional router, serves as the lead for all devices on the network. Any time a device connects to this network, that device’s networking activity is managed and organized by that controller. This works well in systems that are relatively concrete in structure – in a business or domestic setting, wireless access points are stationary, with a known topology. Capacity is relatively known, meaning there’s not to be drastic changes in how many devices are connected. Finally, centralized control allows for better efficiency in networking, allowing for administrative policies and actions to control behavior on the network. An example topology of a network like this is shown below in Figure 17.

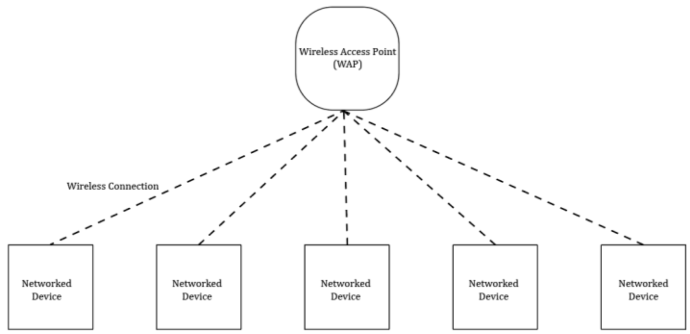


Figure 17: Standard Wi-Fi Topology

There are some drawbacks to this approach, however. Capacity of the network is inherently limited – you can only support as many devices as your wireless access points can handle. As you add more devices, the capacity remains the same, lending to congestion, slow-down, and user dissatisfaction. In addition, the range of a wireless network like this is limited to the wireless access points’ native range. Extending this in a high-quality fashion is difficult without the addition of more wireless access points – each of which must be connected via a hardline ethernet connection to provide service. Overall, this leads to an inflexible system with little room for growth.

With the advent of more mobile computing and wireless technology, “mesh” wireless networking has been developed in order to eliminate some of these concerns. In such a network, devices work on a peer-to-peer basis, establishing connections between each other to create a web of interconnected devices. Each one serves as a wireless access point on its own, offering service – in the form of relaying messages - to other devices connected to it. In a system like this, no one device is in charge, assigning IP addresses, managing network flow, or limiting network capacity. In this system, too, adding more devices just increases the overall capacity of the network.

This is typically referred to as a “full mesh network,” where every node is connected to every other node in the system. This allows for direct communication between each networked device, resulting in lower travel times for any given packet. Figure 18 showcases a full mesh network topology in a system with six devices.

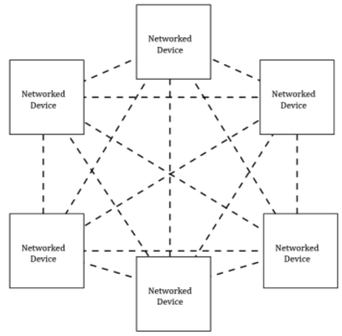


Figure 18: Full Mesh Network Topology

Mesh networks do include some drawbacks, however. Capacity is not truly infinite, as more and more complicated mesh environments can lead to more and more processing power required to properly distribute messages across the network. Not to mention the impact that more wireless interference has on a larger scale. This is in addition to its inherent expense – each device must be equipped with hardware to serve as a wireless access point, meaning not only is the cost of the device itself higher, but power usage on that device may grow more than would be expected under a typical situation.

Full mesh networks, too, are relatively impractical. For such a system, each device must be able to directly connect and message every other device, resulting in a system heavily reliant on each devices individual connections. A system like this could even be more unreliable than a standard network topology – if one device loses connection to another, and we do not route messages around that lost connection, then those messages are lost, too.

Instead, what is typically used is a "partial mesh network," wherein devices maintain between themselves a connected mesh, but messages are routed amongst nodes like in a traditional network. This brings about the advantages of both systems – the mesh properties allow for a dynamic, flexible network topology granting greater customizability and a more adaptive system, while the traditional networking behavior allows for a more stable, more reliable system. Figure 19 showcases an example of the topology of a partial mesh network.

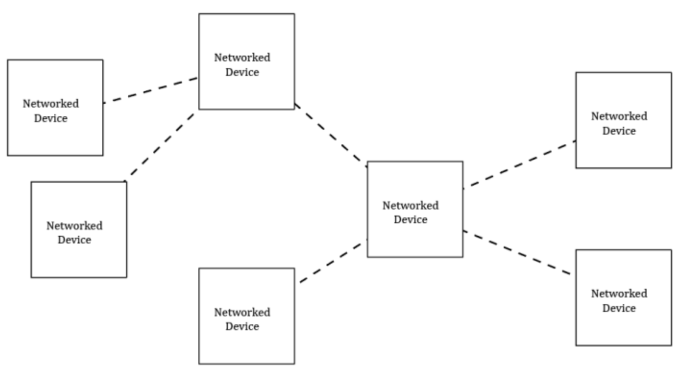


Figure 19: Partial Mesh Network Topology

The advantages of such a technology in our device is clear – by having practically unlimited nodes, we can create a system that can handle as many targets as we have the hardware resources to manage. We can ensure that targets in a longer-range setting can still connect between each other, and then connect with the controller over distances that a single target may not be able to manage. And the flexible nature of the system means that targets can be added and dropped as a user requires during the course of a gameplay session.

3.2.9 Battery Technology

The battery technology considered in this project led us to Lithium-Ion (Li-Ion) vs. Nickel Metal Hydride (NiMH) vs. Lithium Polymer (Li-Po) vs. Nickel-Cadmium (NiCad). Considerations for Lead Acid, Thin Film, SLA batteries etc. were dropped due to weight limitations, pricing, or availability. Power requirements were calculated in section 3.4.2.2 and we concluded that an appropriate capacity for the laser "gun" was a 5Ah battery pack at 12V. Evaluation of each battery technology took these criteria into account.

3.2.9.1 NiCad & NiMH

Nickel Cadmium was evaluated first, along with Nickel-Metal Hydride. A quick search of the battery market revealed an average nominal voltage of 1.2V per cell. [23] Creating an equivalent NiCad battery pack matching the above-mentioned criteria for our laser "gun" of 5Ah at 12V would have weighed approximately 1.68 lbs. This weight was unwieldy for the user and nine cells at a "D" cell battery size would have surpassed our Requirements Specifications size limitation. Similarly, the NiMH battery did not provide many performance benefits, either. A slight bump in energy density by approximately 50% places the battery near the Lithium-Ion energy density performance bracket. While the nominal cell voltage of the NiMH was still 1.2V, this drawback could have been remedied using a voltage boost converter for the 3.3V & 12V required voltage rails. Product availability and flexibility for the battery configuration over NiCad & Li-Ion initially made the NiMH battery choice a solid contender.

3.2.9.2 Lithium-Ion & Lithium-Polymer

Common nominal voltage ratings for Li-Ion batteries ranged from 3.3V to 3.8V, thus only 4 cells were required to reach the target 12V range. The average weight (for an 18650 Li-Ion cell) was 45 grams, totaling to approximately 7 ounces for a 4-cell battery pack [24]. While weight savings were drastically higher than comparable NiMH battery configurations, the pricing for a single Li-Ion battery cell was approximately $1.8 more per cell than NiMH cell pricing.

Lithium-Polymer would have been considered as a last resort solution if the Li-Ion or Ni-MH hydride solutions failed to meet the project's budget or Requirements Specifications. A 3.7V 2000mAh Li-Po battery nearly tripled the price of a single Li-Ion cell with comparable capacity and voltage. Using Lithium Polymer over Lithium-Ion batteries depended on if space was a premium regarding the laser "gun".

3.2.10 NeoPixel

The laser target required a RGB addressable light source that could be controlled via a microcontroller. Additionally, the system should be self-contained and not exceed the power limitations of the laser target. The NeoPixel was our first look at solving this issue and met or exceeded our requirements. For starters the maximum power draw of a single

NeoPixel LED is 60mA at full brightness with white color but reduces to approximately 20mA if we apply dynamic brightness and color variations throughout each LED. Using a 5V supply voltage we can estimate the power consumption for a given hour: ~18Wh. Surprisingly this calculation fits within our power budget analysis explained in section 3.4.2.2 in detail. With the power requirements dealt with the next parameter to investigate is twofold: is NeoPixel self-contained and is it scalable with little overhaul done on an existing design. The NeoPixel isn't a controller but rather a typical RGB LED with an integrated control circuit that receives a 24-bit input from a microcontroller and outputs a 3 channel 8-bit PWM signal from the IC [25]. The minimum input frequency allowed is 8MHz and the RAM storage per NeoPixel is 4 bytes [26]; the microcontroller selected in section 3.4.1 far exceeds these requirements however, one concern to account for will be the reduced storage once all built-in features, game profiles, etc. are added to each laser target. Thanks to the IC inside each NeoPixel cascading additional LEDs to expand a light array is as simple as soldering one LED sequentially and compensating for the increase in LED count through code driven from the microcontroller. Additional factors affecting this decision to select NeoPixel as the laser target's lighting mechanism was the expansive Arduino library on GitHub and guides offered on the Adafruit website. Other LED strips were considered through Amazon or Digi-key but were either sourced from Adafruit, or if a supplier offered competitive pricing documentation was missing to confirm the specifications of said product. After evaluation of all possible options NeoPixel was the selected product to implement the laser target lighting even if the pricing was not optimal (pricing is detailed in section 8.2.1).

3.3 Architecture Selection

In the world of low-level computing hardware, there are many choices to be made when it comes to computing architecture selection. While our selection of peripheral parts was critical for the core functionality of the device, and relatively inflexible in their requirements, in order to tie these components together into a functional, larger system, we had to implement some form of control hardware. As the world of embedded devices solves a wide array of problems, it also comes with a wide array of computing hardware, each with their own strengths and weaknesses. Here, we considered the use of several of these different fields for our project.

3.3.1 Field Programmable Gate Array (FPGA)

FPGA devices are developer-configured, "programmable" chips which perform advanced logical operations with unparalleled accuracy and reliability. By their nature, those logical behaviors are "baked" into the chip itself, meaning the chip is fast, consistent, but relatively inflexible. In addition, their re-programmable nature makes them more expensive than a custom-made chip to fit the same role, on a per-unit basis in large batch purposes.

An FPGA was initially considered for the project due to the simple configurability of the chip, which could be used to realize various designs. When simplifying down to just the laser interaction, management of a "Laser On/Off" state on the controller and a "Detected/Not Detected" state on each target would be simple – maybe even too simple for the use of a full FPGA chip. Designing and implementing a logic circuit for those states would be near instant due to the simplicity of their operation.

However, adding more additional functionality – namely, Wi-Fi communication – would be incredibly challenging. Functionality like that requires precisely timed, sequentially understood messages and communication. In addition, any sort of complicated state management would require a larger length of development time on that platform. We could probably implement everything, eventually, but FPGA chips are not designed for situations like this.

3.3.2 Digital Signal Processor (DSP)

A Digital Signal Processor (DSP) chip is hyper-specifically designed to process analog signals and perform various operations like filtering, measuring, or manipulating them in real-time applications. These chips are designed to do that one thing incredibly well and reliably, making them great choices when working with compatible tasks.

However, our system involves little to no analog signals in its design – the most notable being the current change from the phototransistor on the target itself when a laser is detected. However, such a signal required no advanced processing to become useful, with the only thing we particularly cared about being spikes in its value.

The use of a dedicated DSP chip for that one signal in this project would be not only excessive in budget and power consumption, but challenging for the team to implement, considering no members of the team had experience with these forms of processors. As such, a dedicated DSP chip was not used in our devices.

3.3.3 Microcontroller Unit (MCU)

A Microcontroller Unit (MCU) is generally a low-power, general-purpose chip package which typically boasts a feature-set comparable to a typical computer – standard processor instructions, the standard von Neumann architecture, and support for communicating and controlling other peripheral devices – though with the method of communicating via both digital and analog general-purpose input/output pins (GPIO).

Some microcontrollers even include wireless communication directly in the package, reducing the overall complexity of the device and increasing the ease-of-use for the developers. Additional boons came in the form of their low-power design, the wide array of existing documentation and support for the various processors, and the familiarity some members of our team had with this category of devices.

An MCU was an obvious choice for this project, with one being present in each device to control and organize aspects of both the game and the device peripherals themselves.

A summary of the hardware architecture selection process is shown in Table 2 below.

Architecture	Initial Design Time	Appended Design Time	Cost	Power Consumption	Team Experience
FPGA	Low	High	High	Medium	Medium
MCU	High	Low	Very Low	Low	Very High
DSP	X	X	X	X	Low

Table 2: Hardware Architecture Selection

3.4 Strategic Components and Parts Selections

When selecting components for this project, we decided to take a very hands-on approach to the selection process. We did not want our parts selection decisions to be made purely based on images and specs we found online. Therefore, for each part being considered, a pool of possible candidates was compiled, and the most desirable part(s) in the pool (usually only one or two different parts for each category) were ordered for preliminary testing. Then, each part could be selected based on its performance during testing as well as its compatibility with all other parts being tested.

This section will provide an overview of the selection process for each individual component, including criteria considered, calculations made, tests performed, and comparisons between multiple possible component choices.

3.4.1 Microcontroller

The world of microcontrollers is flooded with options, with chips designed around every possible feature-set and application you could imagine. As such, narrowing down that field into something that one can make sense of is a challenge in and of itself. To aid us in this process, however, we grouped our needs into more generalized categories:

3.4.1.1 Peripheral Requirements

A microcontroller is nothing without peripherals to control, so it's important that we make sure ours can handle the peripherals we've designed into our system. To that end, we need to look at the GPIO pin requirements, what communication channels we need, and what analog or digital requirements the final project may have. This information has been collated in Table 3 below.

Peripheral	Requirements	Notes
Trigger	1x Digital Pin	Only on the controller
Laser Diode	1x Digital Pin	Only on the controller
Screen	I2C Comm (2x Digital)	Only on the controller
Audio Module	1x Digital, 2x DAC	On both devices
Phototransistor	1x Analog Pin	Only on the target
Pairing Mode Bttn.	1x Digital Pin	Only on the target
LED Array	PWM (2x Digital)	Only on the target

Table 3: Peripheral Pin Requirements

We can see that both devices will require at least 7 digital pins to control their respective components, with at most all seven being digital, and at least one being analog, depending on the device. With this in mind, aiming for a microcontroller with at least 10 pins (in case of future expansion) where at least one pin is capable of analog to digital conversion.

3.4.1.2 Power Requirements

Both the "controller" and "target" device platforms are battery-powered, and as such, power consumption and delivery are critical parts of our design process. We aim for our targets and controllers to easily last through multiple sessions of gameplay without depleting their on-board batteries, and as both devices are designed to be somewhat small – the controller being a hand-held form factor – battery capacity and therefore size comes at a premium.

With our current understanding of the power draw of the other components, and our targeted battery life of the overall system, we've budgeted a maximum average current draw of about 500mA during regular operation of the microcontroller, with a hopeful maximum of about 10mA maximum while in a sleeping "standby" state.

When it comes to the supply voltage for microcontrollers, a few notable, standard voltage levels seem to appear – 3.3V and 5V being the most common. These also appear in the power requirements for many of our peripheral components, resulting in either being good choices for our power requirements.

3.4.1.3 Feature Requirements

While the ability to control peripheral devices and receive standard power supply levels are important factors in the selection of a microcontroller, the properties that set microcontrollers apart from one another in the market are some of the additional features that they include in their architecture. Some of these are simple and mandatory, such as the capabilities of the central processor or the capacity of the chip's on-board EEPROM (electrically erasable programmable read-only memory, typically used for non-volatile storage on microcontrollers). More advanced functionality can include integrated display hardware, high-quality digital to analog conversion for audio playback, or wired/wireless internet capabilities. It's these factors that we'll use to narrow down the field of candidate controllers with the most granularity.

Memory

In computing, having a large sum of volatile, high-speed memory is always a boon to both the development team and software complexity. As any program grows in complexity, the more data, more variables, and more resources it must store in memory. Unfortunately, high-speed, low-latency memory is both expensive and power hungry. Even in dedicated desktop or server systems, the amount of memory that system has is always a minor fraction of the storage capacity of that system. In microcontrollers, where everything is at an even greater premium, we must get by with even less.

While this will factor into our implementation of our system software, it's hard to try and calculate backwards what we might expect to handle. Data usage is volatile, and while we can design ourselves around minimizing memory usage, it can still shift and change depending on compiler settings, environmental factors, and more. We can make some assumptions, however, on what having more or less memory allows us to accomplish. If we go for a microcontroller with less built-in memory, we'll be bound in the complexity of game-modes we can implement as well as the number of potential targets a user could have in a game at a time. Inversely, having more memory allows us to have more complicated game-modes and more targets in a system.

As our system is designed primarily around simple, string-based communication, no graphical components on the displays, and very simple audio playback, no single resource will consume large swaths of active memory. This means that our lower-bound for our system memory is low – something around the range of about 100-200 KB should be more than enough. However, more memory allots us more space for development and expansion in the future, so we believe no reasonable upper-bound is required, as other factors (such as power consumption, price, or size) would impose their limits before a memory cap were to take effect.

Storage

While memory tends to be quite limited in microcontroller systems, long-term storage in the form of both ROM and EEPROM tends to be much larger comparatively. Storage is slower to access and shouldn't be used for storing data that is actively being manipulated or processed. But it's cheaper and non-volatile, which makes it great for holding lots of data at once. In the world of microcontrollers, it tends to be composed of flash storage, which is a technology that's grown quickly in the past few decades with the proliferation of solid state (flash) storage across consumer devices.

Our application will likely not require much storage space on the microcontrollers. Much of our data will be the software we write, which is compiled down to a binary application. The only accessory files to this will likely be our audio files, which are going to be relatively light, too. While we likely will be storing our audio data in an uncompressed format, the fidelity / quality of the audio data themselves will be relatively simple and low-quality, matching the capabilities of our playback hardware.

As with memory, the more space available the better for our development efforts. While most compilers include some optimization / compression capabilities in order to reduce the size of the final compiled program, it's nice to have more space than we could possibly need. However, it should come as a secondary factor to the more important power/price requirements.

Clock Speed

The clock speed of a processor determines the frequency of instruction execution for that processor. A processor with a high clock speed can execute more instructions than a processor with a lower clock speed over a given time period. This, however, comes at a cost. A faster clock also means higher complexity in the processor's design and construction, often proportionally affecting the price of the processor. In addition, it also means more power is consumed when compared to similarly-equipped processors operating at lower frequencies.

High clock speeds can be useful in situations where complicated operations with millions of instructions need to be performed in a very short amount of time, or in situations where precise timing is important for the functionality of the system. Communication protocols are an example of this sort of situation, with the speed of each controller being an important factor in not only how fast each device can communicate, but also how much data can be transferred between devices.

This is another requirement that is not a priority or limiting factor for our microcontroller selection. Most microcontrollers on the market today that fulfill our other requirements are more than fast enough for this to be negligible for our scale of work. We're not doing complicated, time-sensitive data processing, but just working to keep pace with the user's experience and inter-device communication protocols.

Wireless Module

This is probably the most critical component of the entire system – second to the laser diode and detector. The design's primary advancement upon pre-existing systems is the inclusion of the wireless aspect, enabling target/controller setups to be reconfigurable and portable. As such, the capabilities of a wireless module, and its inclusion in or exclusion from our microcontroller is important to the core design of our system.

A wireless module isn't commonly included in traditional desktop processors, but rather often stands as a separate component, sometimes mounted to motherboards. These modules are mounted via an M.2 port, which then uses the standard PCI Express lanes on a motherboard to communicate with the main processor [27]. This allows for some modularity in that environment; you can interchange processors without worrying about wireless capabilities of that processor, you can reduce the already high complexity of CPU architectures and develop products that don't require wireless communication – most desktops forgo wireless capabilities in favor of a standard ethernet connection.

However, when developing applications in an embedded environment such as ours, where space, power, and communication lines are at a premium, including a wireless module inside of our microprocessor is massive boon. Not only does this reduce the overall complexity of the system, but by having the communication integrated into the processor, we can rely upon the manufacturer's support and much larger base of documentation to resolve issues that may occur during our development process and speed up development time.

The capabilities of a wireless module itself aren't a particularly harsh set of requirements for us – our base requirement is that the module supports standard IEEE 802.11 Wi-Fi protocols and has a range that can support our target distances of our system. We intend to operate this system utilizing Wi-Fi only, not Bluetooth, in order to take advantage of that protocol's higher speeds, greater reliability, and greater operable distance. As such, the Bluetooth capabilities of any given wireless module are not important to our selection process.

Development Support

This project will involve a large amount of software development – controlling and organizing the various hardware elements of the system such as screens and audio playback, managing game states, communicating wirelessly, all involves a lot of code. As we want to focus our development time on not the low-level driver code, but the higher-level process of tying components and functionality together in order to build the final system, it's advantageous to use systems with established software libraries and development tool chains. Luckily, the past decade has resulted in a large growth of interoperability and support for various embedded hardware systems. The Arduino software environment, for example, supports a large number of microcontrollers and hardware driver modules, making it a perfect environment for us to develop software in. It's just a matter of choosing the right microcontroller that fits in that environment.

3.4.1.4 Conclusion

We can collate our requirements to see what we might be looking for in a microcontroller, and then sort through a few of the most common platforms to find the ones that satisfy our requirements the best.

Table 4 details those criteria as well as values that the microcontroller should meet or exceed.

Criteria	Value
Wi-Fi	Required
Wi-Fi Range	30m minimum (100m+ preferred)
BT	Optional
GPIO Ports	10 ports minimum
ADC	1 port minimum
DAC	2 ports minimum
Flash Storage	4MB minimum
RAM	10KB minimum (20KB+ preferred)
Power Usage	<500mA full load; <10mA standby
Software Environment	Arduino

Table 4: Microcontroller Criteria

There are a wide range of possible microcontrollers out there, but simply due to the amount of support and popularity these microcontrollers receive, we began with a narrow pool of microcontrollers we're familiar with – either through recommendations, research, or prior experience. These microcontrollers shown in Table 5, below.

Model	ATmega640	MSP430FR6989	ESP8266	ESP32D-WROOM
Clock Speed	16MHz	16MHz	80MHz	240 MHz
V_{supply} (V)	1.8-5.5	1.8-3.6	1.8-3.6	3.0-3.6
I_{draw} (mA)	~0.5-14	~1.6 (Controller)	~170 (Controller + RF)	~500 (Full Load)
Storage	4KB EEPROM	Unified /w RAM	16MB max	16MB max
RAM	8KB	128KB	~50KB	520KB
Wi-Fi/BT	N/N	N/N	Y/N	Y/Y
GPIO	86-pins	10-Pins	17-Pins	34-Pins
ADC/DAC	10-bit/N	12-Bit/12-Bit	10-Bit/N	12-Bit/2x8-Bit
RTC/WDT	N/Y	Y/Y	Y/N	Y/Y
Framework	Arduino	Arduino	Arduino	Arduino

Table 5: Microcontroller Selection

Legend

Within Specification	Out of Specification

The ATmega640 microcontroller is first on our list of comparisons thanks to its wide use in existing hardware platforms and broad support, but it is immediately marked as a weak choice due to its low clock speed, limited storage, and lack of wireless capabilities. While it has a massive number of GPIO pins, that one factor alone isn't enough to make up for its other weaknesses. As such, this microcontroller is not a good choice for our project.

Every member of our group has used the MSP430FR6989 in our prior coursework, providing us a strong understanding of how this microcontroller functions and how to make good use of its capabilities. However, it is similarly equipped as the ATmega640 chip: low clock speed and negligible storage and no wireless capabilities. It does have a DAC onboard, which is a boon, as well as a dedicated Real-Time Clock component. However, the limited power of this device and lack of wireless capabilities reduces this to a device that is likely not useful for our work.

This leads us to the last two microcontrollers we've selected to review, two product lines from Espressif Systems. The first model, the ESP8266, is the older version of the ESP32 processor, and this is reflective in the capabilities of each. Both microcontrollers support Wi-Fi through onboard wireless modules, have a similar maximum storage capacity, and have more than enough GPIO pins to support our overall system design. However, as with any generational shift, the ESP32 holds advancements over the ESP8266 in several categories [28].

The ESP8266, as a start, has a typical clock rate that is $1/3^{rd}$ that of the ESP32. While it draws less power, it also has a notably less RAM, and while we don't intend to utilize the Bluetooth functionality of the ESP32, it doesn't hurt to provide that opportunity if we decide we need that going forward. The ESP32 also includes a pair of dedicated DAC pins, which are required for our audio playback capabilities. The ESP8266 lacks this. In addition, the ESP32 allows for up to 10 station nodes compared to 4 on the ESP8266, when using the "PainlessMesh" networking library, which allows us more targets and controllers in the end [29].

More pins, more options, and only marginally more expensive, the ESP32 satisfies all of our requirements by a wide margin. In our first revision, we attempted to use the ESP32 MINI package, but we found it too difficult to reliably solder and communicate with. In the final product, the specific package we ended up using was the ESP32-WROOM-32D (M113DH3200PH3Q0) from Mouser, and it met all of our needs without being too difficult to solder.

3.4.2 Laser Diode

The laser diode is arguably the next most important part to select after the MCU, as this is the part that will emit the laser beam from the controller "gun" when its trigger is pulled, and it will likely have a relatively high power draw compared to other components. We made a few determinations in the very early stages of the project about what kind of laser we wanted to use.

Firstly, to ensure the best user experience while playing the game, we decided that we wanted the laser to be visible, so that when the controller is "fired," a visible dot of laser light appears where it is aimed. The purpose of this is so that the user can see where they are aiming when they fire, so they can use the light as a guide to readjust their aim as needed, and they can see when they hit the target. The spectrum of visible light spans the wavelength range of 380 to 700nm, so the laser selected needed to be in this range. Most laser diodes considered ended up having a wavelength of 650nm, meaning they emit red light. This decision to go with only visible lasers ruled out the use of infrared lasers, which were used in some similar past projects, because they are not visible to the human eye.

Next, to ensure that the laser will not be a hazard to the user of the system or any other people in the immediate vicinity, we decided to go with a laser that would not be very dangerous to the eyes, and not dangerous at all to human skin or other materials. We decided to go with a laser in the FDA/IEC Class IIIa/3R or lower, which are considered low-risk, and akin to a laser pointer. This limited the maximum output power of the laser to 5mW. Further discussion of laser safety and laser classes can be found in the Related Standards and Design Constraints section of this paper.

Size and weight were also taken into consideration. Since the diode has to fit into a handheld gun-like enclosure along with many other parts, it is advantageous for it to be as small as possible. We are also wanting to keep the controller "gun" as lightweight as possible for user comfort, so the lighter the diode is, the better, especially as it is the component that will be located farthest from the end of the controller that is gripped by the user.

Lastly, we took RoHS compliance into consideration, which will be discussed more in-depth in the Related Standards and Design Constraints section of this paper. In order to use as many RoHS compliant components as possible, laser diodes that are RoHS compliant were prioritized.

To make the final laser diode selection, we devised and followed a four-step process. Each of the steps will be detailed in the following subsections:

3.4.2.1 Step One: Gathering a Pool of Candidate Laser Diodes

In this first step, very early in the design process, we researched a number of different laser diodes and compiled a pool of nine of them. This was before we had completely decided on all of the criteria listed above, so in this first search, the only factors that we were considering were upper bounds for the FDA class/output power and price. The nine laser diodes initially considered will be listed in Table 6 below for comparison to one another. Now that the above criteria have been decided on, especially the requirement for a visible laser, the table shows which parts measure up to the chosen criteria. All laser diodes listed were in stock at the time this research was conducted, as well as at the time that Table 6 was made. Additionally, all prices listed were the prices given by the suppliers at the time of this research.

Part #	Visible?	Output Power	Laser Class	Cost	RoHS?	Other Notes
VLM-650-03 LPA-ND	Yes	2.5mW	Class IIIa	$19.16	Yes	-Red laser
VLM-635-04 LPA-ND	Yes	5mW	Class IIIa	$19.18	Not Specified	-Red laser -Strange shape
VLM-520-03LPT-ND	Yes	1mW	Class II	$20.68	Yes	-Green laser

1054	Yes	5mW	Class IIIa	$5.95	Yes	-Red laser -Great price -No real datasheet
1056	Yes	5mW	Class IIIa	$18.95	Yes	-Red laser -No real datasheet -TTL diode
365-1879-ND	No	1.5mW	Class IIIa	$15.97	Yes	-IR laser
365-1888-ND	No	1.5mW	Class IIIa	$6.00	Yes	-IR laser -Great price
38-1007-ND	Yes	5mW	Class IIIa	$12.50	Yes	-Red laser

Table 6: Initial Pool of Candidate Laser Diodes

Legend

Within Specification	Out of Specification

Parts that have criteria which are filled in with red in Table 6 were eliminated entirely from consideration in this first step, leaving four remaining laser diodes under consideration. Factors that caused these diodes to be eliminated include unknown RoHS compliance status, unnecessarily high price for no additional benefit, and lack of visible output light.

At the end of this step, the Adafruit #1054 laser diode was singled out immediately as our first choice at this stage. It met all the criteria from Table 6, and was also easily the cheapest. To us, this means that it will have less of an impact on our overall project spending as well as the cost of materials used in the final design if we decide to use it in the end.

We wanted to go ahead and order a couple of different laser diodes to start testing them with a sensor, so we decided to order both the Adafruit #1054 and the Adafruit #1056. The Adafruit #1056 was chosen as the second kind of laser diode to test alongside the #1054 for a number of reasons. Firstly, we could save on shipping cost and time if we ordered both diodes from the same place, and they would both arrive at the same time. Additionally, the #1056 meets all of the same requirements as the #1054, but it is a Transistor-Transistor Logic (TTL) diode, which provides an additional feature: it has a third wire attached to it that can be used to modulate or pulse the laser. We wanted to give this diode a try to see if the extra cost is worth the extra feature, as well as to decide if we want to make use of this for power savings.

This early decision to purchase these two Adafruit diodes does not mean that the remaining two diodes were taken out of consideration. This initial acquisition was purely for testing purposes, so that we could test both the cheapest option and the most advanced option from the list that we had compiled and compare the two.

The other two diodes from Digi-Key that were still under consideration were ordered a couple of weeks after the initial order from Adafruit. At this time, once they arrive, these are to be tested and used as backup options in case one of the original choices goes out of stock, unless they prove to work better than the initial choices while testing. If that is the case, the highest-quality laser that we feel best suits our project will be used, instead of the cheapest option.

3.4.2.2 Step Two: Power Analysis of Laser Diodes

To assist with the battery selection process, the approximate power consumption of each of the four candidate laser diodes was calculated based on their datasheet values of input voltage and current draw. The power draw of the laser diodes was separated into three distinct brackets: minimum, maximum, and average power draw for each laser diode. The motivation for separate power brackets gives the team an appropriate overview of potential use case scenarios by the operator. The design process structure decision was twofold: first, the most critical and choice limited component was the laser diode when compared to the wider availability of the battery selection, and second, it is a simple process to append additional laser diodes for consideration without having to recalculate any of our previous results for the existing pool of laser diodes. Table 7 below details the power consumption and power brackets calculated for the Laser Diodes considered in Step One.

Part #/Model	$I_{draw}(A)$	V_{input} (V)	Uptime Ratio	Power (W)
VLM-650-03-LPA-ND	0.035	5	0.5	0.175
1054	0.025	5	0.5	0.125
1056	0.035	5	0.5	0.175
38-1007-ND	X	X	0.5	0.0025
ESP32 (LP)	0.034	3.3	1	0.1122
ESP32 (HP)	0.379	3.3	1	1.2507
Speaker	X	X	0.5	1.5

Table 7: Laser "Gun" Controller Power Consumption Summary

Consideration for the laser diodes also includes all other power-hungry components in the system. The microcontroller, visual display unit, and speakers (if added to the final design) must also be considered to properly estimate the laser diode consumption. Including these factors helps the team decide if a specific laser diode is incompatible with our power budget or if we need to shave power consumption by modifying auxiliary components. The ESP32 microcontroller has different load profiles depending on its current state. The two load profiles considered was "full active" where the ESP32 is utilizing all cores, sending/receiving data, and actively managing GPIO pins such as having the laser diode set to "on" and running a display. This profile mimics the usage the microcontroller may experience if a game were running. The second load profile is "standby" where the ESP32 is in "modem-sleep" where the CPU is active, but the cache is idle, this profile was chosen specifically so that the system response time meets the requirements specification where the user should not notice a slow response from the system when waking up from sleep; choosing any profile lower would invalidate the response time requirement. Referring to Table 7, the uptime ratio is simply the total uptime a device will experience in one hour: 0.5 equates to 30 minutes, 1 equates to 1 hour. Table 8 provides a summary of the calculations and power brackets of the entire system paired with an appropriate battery.

The “battery voltage” is an estimated voltage from a single 18650 Lithium-Ion cell which has the highest likelihood of being chosen as the battery source for this project.

Laser “Gun”					
Power Bracket	**Total Power (Wh)**	**Battery Size (Ah)**	**Battery Voltage**	**Wh**	**Uptime**
Minimum	8.06	3	14.8	44.4	5.50544
Average	15.75	3	14.8	44.4	2.818421
Maximum	11.33	3	14.8	44.4	3.917493
Minimum	8.06	4	14.8	59.2	7.340587
Average	15.75	4	14.8	59.2	3.757895
Maximum	11.33	4	14.8	59.2	5.223325
Minimum	8.06	5.4	14.8	79.92	9.909793
Average	15.75	5.4	14.8	79.92	5.073158
Maximum	11.33	5.4	14.8	79.92	7.051488

Table 8: Power Consumption Summary

Legend

Within Specification	Out of Specification

According to the Power Estimation in Table 8, the 3000mAh battery that we had initially planned to use in our first few iterations of the Requirements Specifications was a reasonable choice for the Laser Gun. Our planned design criteria call for at least 5 hours of uptime. In order to meet this, we need to increase the total battery capacity to 5400mAh (or the next capacity increase allowable by the BMS/ battery cell capacity). Interestingly, the minimum power bracket meets the requirements in all three sizes of battery; working toward this efficiency level would bring costs down as the market price for a 5400 mAh battery pack approximates to $24.50. Furthermore, validation of this estimation through testing during Senior Design 2 is recommended to verify that all calculations were done successfully and validate that the datasheets match each device’s characteristics. These findings conclude the laser diode pool selection process. The next step is to find appropriate laser sensors for the system.

3.4.2.3 Step Three: Pairing Laser Sensor to Laser Diode

When selecting a laser diode, one very important aspect of the design to consider is the fact that once it is emitted from the controller, it needs to be received by the target on the other side. This means that the best strategy for laser diode selection is to select it alongside the sensor that will be receiving its light, in order to ensure mutual compatibility. To simplify the design of the DC-DC conversion and reduce the number of voltage rails on our system we selected sensors with a preferred operating voltage input range within 2.5-15V. Since the laser diodes under consideration are all 650nm, the sensor receiving the laser radiation should have a peak spectral sensitivity wavelength near that number. Using Digi-Key as the team's primary source of components, a pool of approximately 355 phototransistors and photodiodes were selected for initial consideration. Applying filtering fields for "Datasheet", "ROHS compliance" and "In Stock" availability reduced the total pool to approximately 30 components. Starting from there, each component was sorted based on five categories and matched from best compatibility to worst. The factors influencing compatibility include Operating Voltage, Peak Sensitivity Wavelength, Sensitivity at 650nm, Pricing and Parts Availability. Operating Voltage often named "Supply Voltage" was not always included on many datasheets. This value must be approximated from either the "Collector to Emitter Voltage (V_{CE})" or the average of the "Absolute Maximum Ratings for supply voltage". For the second category, every datasheet included a "Peak Sensitivity Wavelength" graph. "Sensitivity at 650nm" was extrapolated from each datasheet's graph and used to approximate the maximum signal strength a sensor would output in the form of a voltage. Pricing and Parts Availability were pulled from each component's website. Table 9 below summarizes the findings for the pool of sensors considered.

Sensor	V_{in} (V)	Peak Sensitivity (nm)	Sensitivity @ 650nm (A/W)	Pricing ($)	Availability	Source #1	Source #2	Source #3
PDB-C156	10	660	~0.34	~~1.76~~ 0.546	~~2,857~~ 1,299	~~Digi-Key~~	EIS	N/A
PDB-C152	10	660	~0.34	1.97	52,523	Digi-Key	N/A	N/A
PDB-C142	10	660	~0.31	3.21	11,695	Digi-Key	N/A	N/A
C30737LH	2.5-100	650	~25	~~40.94~~ 13.664	~~21~~ 3,769	~~Digi-Key~~	~~Newark~~	Micro-Semiconductor
ALS-PT19	2.5-5	630	~.7	0.46	41,849	Digi-Key	N/A	N/A
ALS-PT204	2.5-5	630	~0.9	0.50	1,464	Digi-Key	~~Arrow~~	~~Mouser~~
KDT00030	2.5-6	630	~0.5	~~0.75~~ 0.60	~~27~~ 2764	~~Digi-Key~~	~~Avnet~~	Arrow

Table 9: Initial Pool of Candidate Laser Sensors

Note: All components were accessed 11/12/2021.

In Table 9 for Source #1 - Source #3, whichever of the three sources for each component that is not crossed out was determined to have the best selection according to price & availability. Sometimes, the required minimum order caused a source to be removed from consideration, as was the case for the case for KDT00030ATR, where Avnet offered better availability with 18,000 at $0.2406 each, however, it required a minimum order of 3,000 [30].

One further obstacle to choosing a compatible laser sensor was the fact that on some websites the "Product Attributes" were listed incorrectly. One example is the C30737LH-230-83A [31] which states "Wavelength: 650nm" and "Responsivity of 35 A/W at 650nm." Examining the datasheet's "Spectral Response for 800nm & 900nm devices" shows a raw responsivity of ~33 (A/W) at 650nm [32]. This responsivity lowers further to 25 A/W if a 635nm filter is applied (it can be concluded that applying a 650nm filter would raise this responsivity by a few A/W). This discrepancy was observed frequently when sorting through the Laser Sensor pool and increased the difficulty of finding compatible parts for our 650nm laser-diode.

The figures below showcase a couple of examples of the discrepancies observed between the above components' Peak Sensitivity Wavelength graphs and any "Product Attributes" and actual datasheet specifications. Wavelength for the PDB-C156 [33] was specified at 660nm, but the true peak spectral sensitivity is 920nm, as shown in Figure 20 [34]. This wavelength is in the infrared range of light, and therefore not what we need. This discrepancy also applies to PDB-C152SM [35]. For PDB-C142 [36] the specified wavelength is "660nm" according to the "product attributes" on Digi-Key. The peak spectral sensitivity is 880nm, as shown in Figure 21 [37]. Once again, this wavelength is in the infrared range of light.

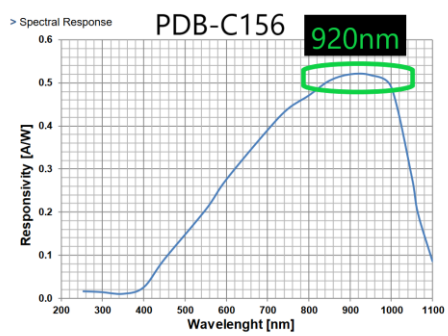


Figure 20: PDB-C156 Spectral Response Graph

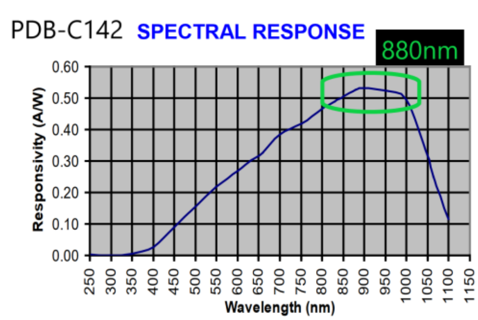


Figure 21: PDB-C142 Spectral Response Graph

The next graphs align closer with the Digi-Key listed attributes but suffer from an unrelated problem. Sorting through various Laser Sensors did not yield a sensor with a peak spectral sensitivity of exactly 650nm. Thus, we had to resort to including components with peaks slightly off the 650nm mark. This introduces a potential gain issue along with the possibility of a noisy signal due to the low Signal-to-Noise Ratio (SNR). The figures below detail the source of these concerns.

The ALS-PT19-315C [38] specifies a peak sensitivity wavelength of 630nm. As shown in Figure 22 [39] below, at 650nm it decreases below to 68% of its maximum. Compared to the other sensors, this gain is still reasonably high. However, the red oval encloses an exponential drop which happens to be inside our 650nm mark. If the Laser Diode is off by ±10nm the expected phototransistor gain will deviate by ~10%. This characteristic is observable for the KDT00030ATR as well.

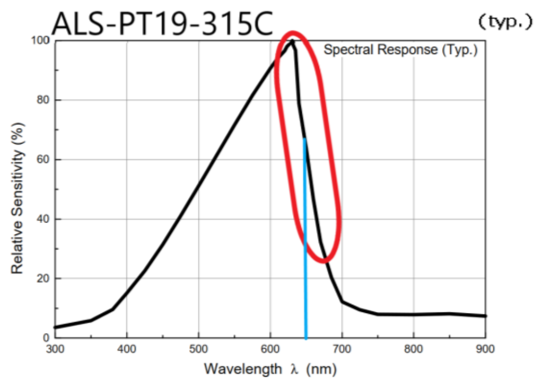


Figure 22: ALS-PT19-315C Spectral Response Graph

The ALS-PT204 offers the best 650nm sensitivity performance with the least gain fluctuation (2%) for every 10nm wavelength deviation, as shown in Figure 23 below [40]. There is reasonable availability of this component at a competitive price. This component will be our secondary backup choice. The Adafruit phototransistors took priority since they were the fastest parts to arrive and were able to undergo testing first before any other parts arrived.

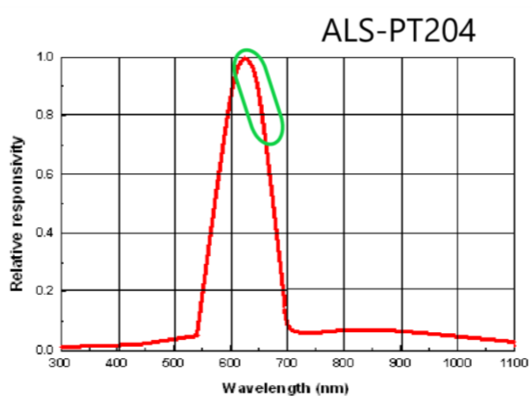


Figure 23:ALS-PT204 Spectral Response Graph

Table 10 below summarizes the hardware team's laser sensor selection pool. HW5P-1 is our Adafruit laser sensor and is our primary candidate for testing due to availability, shipping speed, relatively cheap price ($0.90) compared to the pool of candidates and is a through-hole component facilitating the initial testing process. Candidates #2-4 have been purchased and are in inventory in case #1 fails testing or presents unsatisfactory performance.

Sensor	Selection Order
HW5P-1	#1
ALS-PT204	#2
ALS-PT19	#3
KDT00030	#4
PDB-C156	#5
PDB-C152	#6
PDB-C142	#7
C30737LH	#8

Table 10: Summary of Selected Laser Sensor Candidates

Lastly, it should be mentioned as part of this discussion that every 2-3 weeks an inventory check will be performed for all critical components (Laser-Diode, Laser Sensor, ESP-32, Battery etc.) to make sure that any supply-chain availability issues are caught beforehand. Further selection criteria for the laser sensor beyond compatibility with the laser diode will be discussed thoroughly in the laser sensor selection section.

3.4.2.4 Step Four: Verifying Datasheet Information and Testing System

Returning to the discussion of the laser diodes themselves, our first choice, the Adafruit #1054 diode, was tested in our first round of preliminary testing. At this time, we have elected not to use the Adafruit #1056 in our design because we do not believe the one extra TTL feature is worth having to use a $20 laser diode. The actual testing procedure and specific results for the Adafruit #1054 laser diode are detailed later in the Overall Hardware Testing section, but a brief mention of these results will also be made here as part of the selection process. The Adafruit #1054 does not come with a proper datasheet, as previously mentioned, but some "technical details" are given on its webpage, including a "2.8-5.2 DC voltage input" and a 25mA max current draw. During preliminary testing, the given DC voltage range proved to be true, but we found that the current needed to be limited to closed to 15mA, and a resistor needed to be placed in series with the diode to protect it from burning out. Therefore, despite the fact that the information given on the diode's webpage is incomplete, we managed to make up for this fact by discovering the rest ourselves through testing. The Adafruit #1054 performed satisfactorily enough to justify it remaining our first choice, as it had an excellent brightness, range, and its light was picked up very well by our first-choice phototransistor during testing as both a close and longer range.

3.4.2.4 Final Laser Diode Selection

After a thorough examination of all the laser diodes in our initial pool for their prices, availability, power consumption, compatibility with available sensors, and a bit of performance/functionality testing, the laser diode we settled on as our first choice for use in the final design is the Adafruit #1054 laser diode. It is the cheapest in the pool by far, has excellent availability, is compatible with our first-choice sensor, is relatively low-power compared to the rest of the pool, and performed well during testing.

3.4.3 Laser Sensor

This section expands on the laser sensor research that was conducted during the laser diode selection process by discussing and comparing of all three types of light-sensing devices (photoresistors, photodiodes, and phototransistors) initially considered for use as the laser sensor, as well as describing why we settled on only photodiodes and phototransistors for our consideration. It then describes the testing and research steps that were followed to reach our final choice (including testing and researching for compatibility with our final choice for the laser diode, as was discussed in the previous Laser Diode section), and the final component choice is detailed with information from its datasheet as well as any other information that we found necessary to know about it before using it in our final design, such as how it may be adjusted or filtered to best receive the signal from the laser diode (wavelength spectral sensitivity considerations).

3.4.3.1 Photoresistor Considerations

Initially, when we were doing early research for the device we wanted to use to sense the laser, using a photoresistor seemed like an obvious choice. It is the first result you get on many sites such as Adafruit, Mouser, and Digi-Key when you search for a light sensor, and all sites have many of them in stock. It was also utilized for the same laser-sensing purpose in a number of somewhat similar past senior design projects done by UCF students. They are also tempting to use because of their simplicity. Resistance is one of the most fundamental electrical engineering topics and is covered extensively in our coursework as early as Linear Circuits I, so all group members are at an expert level of proficiency at dealing with resistors and resistances. The same is not true for diodes and transistors. While we are all still decently familiar with them, they were covered for the first time in our Electronics I course, which was much more recent for all of us, and therefore our level of proficiency with their use and understanding of their behaviors is not as high as it is for resistors.

However, despite all of these advantages that a photoresistor has (at least in our estimation) over similar components, it was removed almost entirely from consideration after further research. As it turns out, almost all photoresistors are made with lead or cadmium, which disqualifies them from being RoHS compliant. Our group decided early on in the project to endeavor to include only RoHS compliant components in our project as much as possible, because our group desires to be as environmentally friendly as possible in the creation of this project (A thorough discussion of RoHS compliance, why it is important, as well as an expanded explanation for why our group has decided it is important for us is included in the Environmental Constraints subsection of the Related Standards and Design Constraints section found later in this document). Searches for a “RoHS compliant photoresistor” only yield out-of-stock components, or usually no satisfactory results at all.

Therefore, unless success absolutely cannot be found using either a photodiode or phototransistor for our light sensing component, the use of a photoresistor is no longer under consideration. We did not even go so far as to research specific models to order and test, because we did not deem it necessary.

3.4.3.2 Photodiode Considerations

The photodiode is another light sensing device under consideration. Unlike photoresistors, most (if not all) photodiodes are RoHS compliant, so that was not a concern when considering a photodiode.

As mentioned previously in the laser diode selection discussion, the most important consideration to make is whether or not the sensor is compatible with the laser diode it will be receiving the signal from. Because we have chosen to use visible red laser diodes which have a wavelength of approximately 650 nanometers, the photodiode's peak spectral sensitivity should be at or near 650nm for it to best be able to pick up the light from the laser diode. Another consideration that needs to be made is input voltage. To keep the design of the DC-DC conversion simple, and to reduce the number of voltage rails on our system, we were searching for sensors with a preferred operating voltage input range within 2.5-15V. Other important considerations made included cost and power draw, neither of which is a big issue as far as photodiodes are concerned, as they tend to be both cheap and low-power.

With this criteria in mind, a search was conducted for candidate photodiodes. From this search, we ended up finding only one photodiode that we were interested in considering for the project. That being said, the one that we did find looks promising, and we decided to order a few of it for testing.

The photodiode that we ordered for testing is sold by both Digi-Key and Mouser (but it was ordered from Mouser), and it is made by OSRAM Opto Semiconductors Inc. Its manufacturer number is SFH 2440, and it is called a "DIL SMT Ambient Light Sensor," with DIL meaning "Dual In-Line Package" and SMT meaning "Surface-Mount Technology." It is a square, surface-mounted component housed in a clear epoxy package, with a light sensitive area of 7 square millimeters. It has a price tag of $1.54 per sensor, which is on the more expensive side for a sensor, but obviously still very manageable, especially if the quality is worth is. Its datasheet lists such features as spectral sensitivity adapted to human eye sensitivity, low temperature coefficient of spectra sensitivity, high linearity, and fast switching time, all of which are desired characteristics for our laser sensor. Most important to note is that its spectral range of sensitivity is 400-690nm, and its wavelength of max sensitivity is 620nm, very close to our laser diode wavelength of 650nm, so it should be able to pick up the signal from the laser diode extremely well.

The main drawback to the fact that we found only one suitable photodiode is the possibility of it going out of stock before we acquire all that we need for Senior Design 2. Mouser and Digi-Key both have a lot of them in stock at the time of writing this paper, but if we were to decide to use this photodiode, multiple would need to be ordered immediately just in case (especially since we are planning to build multiple targets), or a very close eye would need to be kept on the current stock of all sites that carry it. If the part were to go out of stock during Senior Design 2, it would likely need to be replaced with a phototransistor, the other RoHS compliant component under consideration. This would also cause a bit of a problem because a photodiode and a phototransistor cannot simply be swapped, as a photodiode and a phototransistor are not a 1:1 replacement for each other due to the high gain of the output from a phototransistor, which would need to be compensated for if a substitution is being made. Therefore, the plan is not to use the photodiode in the final design unless it proves to be significantly better than any of the phototransistors in testing.

3.4.3.3 Phototransistor Considerations

As discussed in Relevant Technologies, the phototransistor operates essentially just like a photodiode, but with an amplified output. Therefore, the considerations made for the phototransistor were identical to those made for the photodiode, especially as pertains to the search criteria. We searched for phototransistors with a peak wavelength spectral sensitivity at or near 650nm, the wavelength of the laser diode used for the "gun" controller, and an input voltage range within 2.5V to 15V. Also similar to a photodiode, most (if not all) phototransistors are RoHS compliant, so finding parts to meet this criteria was not an issue at all. However, unlike our search for the photodiode, our search for the phototransistor yielded four possible candidates, which were mentioned in the Laser Diode section (Sensor #1 - #4 in Table 10), will be discussed more in-depth in this section, and then summarized in a table. All four candidate phototransistors were ordered for preliminary testing.

The first phototransistor we found was the "Photo Transistor Light Sensor" from Adafruit, also referred to in its datasheet as "HW5P-1." It has a price tag of $0.95 per sensor, and there were plenty in stock at the time of writing this paper. It is a simple through-hole component with one long pin and one short pin, sensor dimensions of 5mm x 5.3mm, an operating voltage of 3-15VDC, and a built-in optical filter that gives a spectral response similar to that of the human eye. More specifically, from its datasheet, it has a spectral sensitivity range of 480nm to 1050nm. One disadvantage of this particular sensor is that its datasheet does not give its spectral sensitivity waveform or its peak spectral sensitivity wavelength, so we do not know if its peak is at or near 650nm, like we are looking for. However, if it performs well in testing, us not having this information should be of little consequence. A few of these were ordered for testing.

For this first option, when it is in use and provided power, once light hits it, it induces a current to flow from its longer pin to its shorter pin. To be tested and applied for use in this project, its longer pin will be connected to power, and its shorter pin will be connected to a 1kΩ to 10kΩ series resistor to ground. When there is no light shining on the sensor, there is almost no current flowing out of it, and the voltage across the series resistor should be nearly zero. When light is shone on the sensor, its current output should increase, and the voltage across the resistor will rise with it, proportionally to the light intensity. This voltage across the resistor can then be read by the microcontroller in the final prototype. This procedure for receiving the output for the sensor should be similar for all models of sensor being considered, but any differences will be detailed as each is described.

The next phototransistor under consideration is the "ALS-PT204-6C/L177" from Everlight Electronics Co Ltd. and sold by Digi-Key. It has a price tag of $0.50 per sensor, and there were plenty in stock at the time of writing this paper. It is a through-hole phototransistor with a 3mm lamp, and an input operating voltage range of 2.5V to 5.5V. Features listed in its datasheet include a response that is close to the human visible light spectrum, a light to current analog output, and a low sensitivity variation across various light sources. Its spectral sensitivity wavelength range is given as 390nm to 700nm, with its peak sensitivity wavelength being 630nm, which is very close to (but not exactly) our desired wavelength of 650nm, as mentioned previously in the laser diode compatibility discussion. In practice, it will work much like the first option discussed, with the input side connected to power, and the output measured across a series resistor connected on its other end to ground. The datasheet for this one also calls for a capacitor in parallel to the resistor, likely to stabilize the output measurement. A few of these phototransistors were ordered for testing.

Another phototransistor being considered is the “ALS-PT19-315C/L177/TR8” from Everlight Electronics Co Ltd. and sold by Digi-Key. It has a price tag of $0.46 per sensor, and there were plenty in stock at the time of writing this paper. It consists of a phototransistor in a miniature surface-mount device, with dimensions of 1.7mm (Length) x 0.8mm (Width) x 0.6mm (Height), and an input operating voltage range of 2.5V to 5.5V. Very similar to the previous sensor, features listed in its datasheet include a response that is close to the human visible light spectrum, a light to current analog output, and a low sensitivity variation across various light sources. Also just like the previous sensor, its spectral sensitivity wavelength range is given as 390nm to 700nm, with its peak sensitivity wavelength being 630nm (once again, very close to the desired 650nm, but not exact, and with a lower gain at 650nm than the previous phototransistor, as mentioned previously in the laser diode compatibility discussion). In practice, it also operates just like the previously described sensor. After a thorough examination of the datasheets for both of the Everlight sensors (this one and the previously discussed sensor), they can be considered nearly operationally identical in theory from the datasheets. If the decision comes down to both of them, the final choice will depend on how they perform in testing (particular in the area of their output gain based on wavelength spectral sensitivity), as well as any personal preference we may have when choosing between surface-mount and through-hole components, including which of the two orientations provides for a better user experience when trying to hit them with a laser light. A few of these were ordered for testing.

The final phototransistor in the group of four under consideration is the “KDT00030ATR” from On-Semiconductor and sold by Digi-Key. It has a price tag of $0.75 per sensor. It is a surface-mount phototransistor with dimensions of 1.7mm x 0.8mm, and a height of 0.6mm. The datasheet does not specify an input operating voltage range, but it does briefly mention using 5V for the input, so it is assumed to have a similar input range to the previously discussed phototransistors. Testing would be required to prove this assumption if this sensor is chosen. Features listed in its datasheet include a spectral response close to that of the human eye, a good output linearity across a wide illumination range, and a low profile. A specific range of spectral sensitivity wavelengths is not given, but the figure given in the datasheet shows that it is around 380nm to 700nm, and the peak spectral sensitivity wavelength is given to be 630nm, just like the previous two (which is desirable, but not exact, for reasons previously mentioned). The main drawback to this specific part is that there were only 27 in stock on Digi-Key at the time of writing this paper. Mouser similarly had only 150 in stock at the time of writing this, and they do not expect more until May of 2022. This low stock on both major reputable sites may prove to be a problem if this sensor is used for our project, and a close eye would need to be kept on the sites, or many would need to be ordered just in case. It would be wisest to avoid selecting this sensor to use in the final prototype, unless this one stands out exceptionally in testing. A few of these were ordered for testing.

3.4.3.4 Final Laser Sensor Selection

This section summarizes the entire selection process for the laser sensor. Table 11 below gives a summary of the criteria considered when narrowing down which of the three sensor types to use, and the results of this process of elimination, which highlight phototransistors as the most suitable and most likely choice.

Sensor Type	Usable for Project?	RoHS Compliant?	Suitable Options Found?	Under Consideration?
Photoresistors	Yes	Almost Never	Plenty	No
Photodiodes	Yes	Usually	Only one	Unlikely
Phototransistors	Yes	Usually	Plenty (4+)	Yes

Table 11: Summary of Sensor Type Process of Elimination

Legend

Positive	Negative	Less-than-ideal

At the end of Senior Design I, all five components discussed previously in this section were ordered for testing: the one photodiode and the four phototransistors. The "HW5P-1" phototransistor from Adafruit was the first to arrive and went through the most preliminary testing with the chosen laser diode. The details of this test will be described in a later section. The photodiode was tested just in case, but it was not used as a replacement or backup. The other three phototransistors ordered were tested upon arrival. During the testing, the "ALS-PT204-6C/L177" from Everlight Electronics Co Ltd. (sold by Digi Key) outperformed the "HW5P-1" in testing by having a greater output response to excitation from the laser. Because both components were through-hole, it was an easy adjustment to make, requiring only a voltage change for the phototransistor rail from 12V to 5V. The "ALS-PT204-6C/L177" was used in the final product.

3.4.4 Battery

Considerations for the NiMH & NiCad were eliminated due to many of the battery packs available either missing BMS protections including Over voltage, over discharge and over current; or not having a datasheet to examine the charger characteristics of each battery. Such battery packs like the ICR18650 Lithium Ion 3.7V 4400mAh available on Adafruit.com offers a competitive price of $19.95 over the Digi-Key $14.55 NiMH 2.4V 3700mAh battery pack [41]. While both products offer sufficient capacity for powering the laser "gun" & laser targets utilizing a low voltage would require the use of boost voltage converters and we cannot afford any efficiency penalties. Instead, the selected battery of choice was an Efest 18650 4S1P 14.4V 2600mAh battery pack; complete with a BMS meeting all the team's protection requirements and providing adequate capacity for prototype testing of the laser "gun". Furthermore, supply chain limitations for any suitable NiMH & NiCad battery packs that met our requirements were either too expensive in consideration of their capacity or had shipping delays of a minimum of five days plus standard shipping times. The choice for 14.8V allows for a simpler & efficient voltage buck design at the expense of capacity, this tradeoff will be further remedied using dynamic sleep profiles for the ESP32 microcontroller. The BMS considerations have been accounted for and have been omitted. Implementing a feature such as SOH of the battery would necessitate a replacement of the BMS with a more expensive chip that monitors each cells health per charge or discharge cycle and would either be a simple analog chip or have a communication bus to display real time data of the battery pack's health. This feature was considered but ultimately not implemented. Sourcing & discounts of the batteries was sponsored by Smart Charging Technologies and provided the team with four complete Efest battery packs for use in this project.

3.4.5 Voltage Regulator

Now that the battery selection has been finalized the voltage regulator topology can now be designed. Since the chosen battery is a 14.4V nominal battery with a minimum cutoff voltage of approximately 11-12V and a maximum charge voltage of approximately 16.8V we can easily design parallel voltage buck converters for each rail required by the laser system. This simplicity bypasses the efficiency penalty incurred with a sequential voltage buck design and eliminates the need for an expensive buck-boost IC thus requiring only one PCB design for both the laser target and laser "gun"; the laser target only requires the 12V rail (to power on the photodiode) and 3.3V rail (to power the ESP32 microcontroller), likewise the laser "gun" only requires a 5V rail (to power on the laser diode, and 3.3V rail (to power the ESP32 microcontroller). Additionally, a MOSFET was chosen to drive the laser diode through a 3.3V gate voltage and 3.8-5V supply voltage. A simple resistor divider is under consideration to reduce the 5V rail to the nominal gate voltage of the MOSFET. Figure 24 below details the basic system topology of the laser "gun" and laser target.

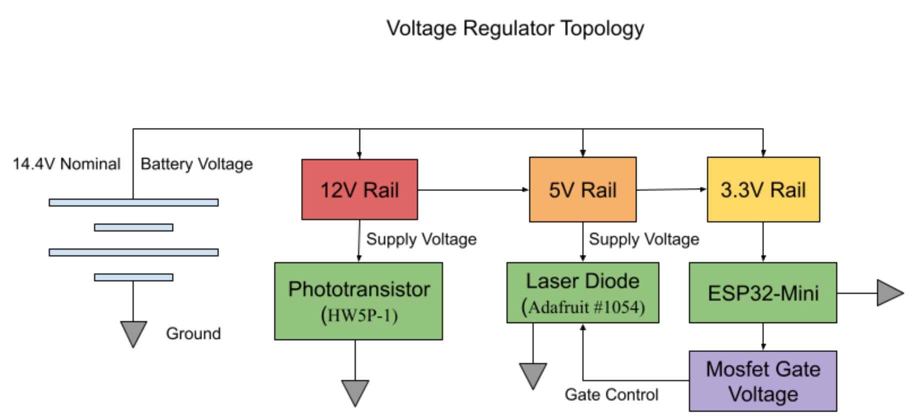


Figure 24: Voltage Regulator Topology

The voltage regulator design choice thus requires that each rail can handle the maximum current through the laser "gun" or laser target. The ESP32 alone may consume 379mA plus: the 15-30mA draw from the laser diode and OLED display or 1 Amp draw from the laser target LED strip display. We decided to design the system with at least a 25% current overhead to prevent any possible damage or heat dissipation due to current spikes from the system's loads.

The MC34063ADR was the team’s first choice for the voltage regulator due to it meeting all rail requirements and offering a competitive price of $0.78 per IC. The design of the feedback loop however, only contained a reference regulator and an inverting input comparator, no compensation was being performed on this feedback path [42]. Simulation of the device showcased unusual behavior of the switch emitter output and the simulation was unable to take a 12V input and output 12V even with tuning of the discrete components outlined in its datasheet [42]. (i.e., the duty cycle appeared to max out at 58.33%); the highest output voltage observed was 7.3V. Thus, the chip was eliminated from consideration due to modeling issues. The newly selected IC for this topology is the BD9227F manufactured from Rohm Semiconductor. It meets all voltage rail input-output requirements with sufficient efficiency (≥90%) at 300mA [43]. This IC allows for adjustment of the internal comparator frequency characteristics and digital adjustment of the output voltage via a PWM signal. Table 12 below summarizes the voltage regulator IC selection below.

Model #	Stock	Price ($)	Selection Order
BD9227F-E2	660	1.37	#1
~~**MC34063ADR**~~	~~45,665~~	~~0.78~~	~~ELIMINATED~~
LM2736XMK/NOPB	12,676	2.60	#2
LM2576SX-3.3/NOPB	1,357	4.06	#3

Availability & Pricing gathered on November 28th

Table 12: Voltage Regulator Selection

All of the above was the original plan at the end of Senior Design 1. However, during Senior Design 2, a couple of things changed. First, we switched from the “HW5P-1” phototransistor to the “ALS-PT204-6C/L177” phototransistor, which required only a 5V power supply rather than a 12V one. The second thing that changed was the fact that we were unable to get the BD9227F-E2 voltage regulators working consistently on our PCBs after a couple of revisions. This may have been a layout error, an issue of the BD9227F-E2 being too complicated to implement (it required several additional components to work properly), or some other unknown error. In our final PCB revision, we decided to eliminate the unreliability of our voltage regulators by placing the previously used BD9227F-E2 regulators in parallel with new linear regulators, which would be simpler to implement, solder, and troubleshoot if needed (and the only other components required for them were two capacitors for each, one for input and one for output). Whichever one worked more reliably would be left on the board, and the other would become an open circuit and left unused.

These linear regulators were quickly found with the help of Mouser’s search tool by looking for our desired input and output voltage ranges. For the 3.3V regulator, we chose the UA78M33CDCYR manufactured by Texas Instruments. It is a fixed voltage regulator with an input voltage range of 5.3V to 25V, and a fixed output of 3.3V and up to 500mA. For the 5V regulator, we chose the MC7805CDTRKG manufactured by onsemi. It is a fixed voltage regulator with an input voltage range of 7V to 35V, and a fixed output of

5V and up to 1A. Both linear regulators performed perfectly in testing (both run a little warm, but not dangerously so) and were used in the final design (instead of the BD9227F-E2 switching regulator).

3.4.6 Controller Display Screen

The display screen on the controller will be the primary means by which the entire system communicates with the user. It will be used to give startup and gameplay instructions to the user, to help the user select which game mode to play, keep track of the user's score and any other relevant gameplay information, and indicate to the user whether the controller or any of the targets need to be charged.

3.4.6.1 Screen Control - Touchscreen vs. Tactile Buttons

For controlling the screen, we considered two options: touchscreen or buttons. Touchscreens are very popular right now, and they are much flashier and more high-tech than buttons. However, buttons have some pretty big advantages of their own that should not be overlooked. This section will outline the process we followed for comparing and contrasting the pros and cons of buttons versus touchscreens, and then once all our reasoning behind our decision has been discussed, the final decision will be presented.

Intuitive for the User: Everyone knows what buttons are and what their purpose is, they are very intuitive. If a touchscreen is used, it needs to be made very clear that it needs to be touched, as well as where it needs to be touched to achieve desired results.

Cost: Buttons are cheaper than touchscreens, usually by a very considerable margin.

Space: Buttons will require additional space on the controller in addition to the screen, whereas the touchscreen would only take up the space of the screen.

Reliability: Both buttons and touchscreens are fairly reliable, although buttons will likely require debouncing and touchscreens may be too sensitive or not sensitive enough, depending on the user and the context.

User Feedback: The user feedback is better for buttons than touchscreens, because the user knows for sure when a button has been pressed, because they can feel it going down. With a touchscreen, the user is not sure that they have pressed it until the screen reacts in response.

Calibration: Buttons do not require calibration, unlike most touchscreens. In the context of our project, this will save us time and effort if we do not have to calibrate a screen [44].

Personal Preference: This was the true deciding factor in the end. As a group, we discussed all of the above pros and cons and made a decision about whether we would rather work with buttons or a touchscreen while designing the system, and also while using the system. We determined that touchscreen programming and calibration was more work in the design phase than it was worth. We also felt that a touchscreen may not even provide as satisfactory of an experience for the user. Buttons are much easier and simpler to work with for everyone, and in the context of this small handheld device, a touchscreen may have been too small to use without frustration for the user caused by accidentally touching it in the wrong place.

3.4.6.2 LED/LCD vs. OLED

Another decision we needed to make in regard to the screen was the choice between an LED/LCD screen and an OLED screen. For many years now, LED/LCD displays have been the display of choice for small electronic devices. However, OLED (Organic LED) displays are rising in popularity, and they offer many advantages over both LCDs and LEDs. These advantages will be presented in this section to explain our choice to use an OLED display for the controller's screen in our project. Specific focus will be placed on display size, weight, brightness, power consumption, and field of view.

Size, Shape, and Flexibility: OLEDs are made of plastic, organic layers, rather than the glass used for LEDs and LCDs. This means that the OLED is thinner, lighter, and more flexible than an LED or LCD. Because our goal is to make the controller as lightweight as possible for ease of use, even if the weight difference is small, we should prioritize lighter-weight parts, because weight does add up quickly. Therefore, the OLED is the more preferable display in the weight category.

Brightness: OLED displays are brighter than LED displays. This is because the organic layers of an OLED are much thinner than the equivalent inorganic crystal layers of an LED, so the conductive and emissive layers of an OLED can be multi-layered. LEDs and LCDs also require glass, which absorbs some of the light, while OLEDs do not require glass. The brighter the display on the controller, the easier it will be for the user to see and read, especially if the user is playing in a well-lit area. Therefore, the OLED is the preferable display in the brightness category.

Power Consumption: One major difference in the way that LCDs and OLEDs work is the use of backlighting. LCDs operate by selectively blocking sections of the backlight in order to display images, and most of the power consumed by LCDs is used for the backlighting. OLEDs do not require a backlight at all, as the individual LEDs produce their own light, meaning that they consume much less power than LCDs. This is extremely important for battery-powered devices like the controller. Less power consumed equals more battery life for the controller, which we are hoping to maximize, so the OLED is the preferred display in the power consumption category.

Field of View: OLEDs have a field of view of about 170 degrees, without much change in image quality if viewed from an angle. This is not the case with LCDs, which have a more limited field of view, and even inside that field, the quality of the image is extremely inconsistent, meaning that its brightness, contrast, saturation, and hue vary with the position of the viewer. They are best viewed head-on. Since the controller for this project is a handheld device that is meant to be held at different angles, heights, and positions during gameplay, the screen should be readable to the user at all of these angles and orientations so that the user can keep track of their score. Therefore, the OLED is the preferable display in the field of view category [45].

While OLEDs are preferred for all of the above categories, as well as for our project as a whole, it is prudent to discuss a couple of their disadvantages here, as these also had to be considered during the selection process, as well as a short discussion of why these disadvantages did not prevent us from choosing to go with an OLED display.

Cost: OLED displays in the size we need cost more than LED/LCD displays of the same size. However, this increase in cost (usually around $10 or higher in our research) can be easily justified by all of the advantages of OLED mentioned above. Additionally, as we are not on a strict budget and we only need one or two screens, an extra $10 or so spent on a nicer screen will not impact the project budget by much at all.

Water Damage: OLEDs are easily damaged by water. The user should be very careful to keep the OLED screen away from water at all times while the system is being used, transported, and stored. It must be stored in a dry location, especially if the storage is ling-term. Luckily, the system is meant to be used mostly indoors, and hopefully should not need to come in contact with water or too much moisture in the air [46].

3.4.6.2 Final Screen Selection

After research and a bit of preliminary testing, we have chosen to use the "Monochrome 1.3" 128x64 OLED Graphic Display" from Adafruit, which has Product ID #938 on their website. It is a small OLED display, about 1.3" diagonally across, and 128x64 individual OLED pixels. The complete device size is 35.6 x 33 x 6.2 millimeters, and its weight is 6 grams, making it conveniently small and light enough to fit into the controller's handheld housing with ease. The display makes its own light, so no backlight is needed. This greatly reduces the amount of power required to run the display, and provides it with a high contrast, making its image display crisp and easily readable. It comes with an attached driver chip, SSD1306, which has the option to communicate using either I2C or SPI, although I2C is its default (two jumpers will need to be cut if it is to be used in SPI mode). The device details on the website state that it requires a 3.3V power supply and 3.3V logic levels for communication, but a 3.3V regulator is included, and all pins are fully level shifted so that it can be used with 3V or 5V devices. It also states that power requirements depend on how brightly the display is lit, but on average the display uses about 40mA from the 3.3V supply, which means it consumes about 132mW on average.

This product has a price tag of $19.95, which is a bit more expensive than some other similar options offered from places like Amazon, some of which are $10 or less. However, we decided to go with this more expensive display because we believe that it will be more reliable and an give an overall higher-quality performance. Because this is the main way that the user will set up and control the game, the screen should be as high-quality, readable, and reliable as possible for the most positive user experience while playing the game. Therefore, a good display screen is very much a top priority, even when it comes to budgeting the project. Additionally, since there is only one controller planned, we will only need to acquire a couple of display screens (with at least one available for backup), so choosing a slightly more expensive option will not make a massive difference in the overall project spending.

3.4.7 Target LED Array

For the targets, we wanted some kind of display that would interact with the user and behave differently for each of the different game modes. We originally considered a number of LED matrixes from Adafruit, but these proved to be pretty expensive for the amount of area we were hoping to cover with them, so a new solution was sought. We decided to order a 1-meter LED strip from Adafruit for preliminary testing. It is called "Adafruit NeoPixel Digital RGB LED Strip - White 30 LED – WHITE," and goes by product ID #1376 on the Adafruit website. Each meter on the strip contains 30 RGB LEDs that can be individually controlled. It is cheaper than a LED matrix and uses less power, as well.

Because the lighting display for the targets is more of a feature than a critical design requirement, this part of the testing process was determined to be lower-priority and the specifics were designed during Senior Design 2. The plan was for the LED array display on the targets to provide fun visual feedback to the user whenever the target is hit successfully with the laser. The different displays and responses of the LED array differ depending on the game mode the user is playing in. As each LED in the array is individually addressable, the hardware design will ensure that the entire array is powered on, and the specific displays shown will be controlled in software.

3.4.8 Audio Components

For this project, we placed speakers in the "gun" controller as well as in all of the targets. The speaker in the controller provides aural feedback to the user when the trigger is pressed, and the speakers in the targets do the same when the targets are hit. This audio feedback, when combined with the visual feedback of the laser emitting from the controller and the targets lighting up when hit, adds an additional layer of fun and realism to enhance the user experience.

3.4.8.1 Audio Amplifier Chip

In order for the speakers to output audio sent to them by the microcontroller, they need an audio amplifier that acts as a digital to analog converter (DAC) to be the "middleman" to help them. For testing this purpose, we chose the "SparkFun I^2S Audio Breakout – MAX98357A" board. This audio breakout board uses a MAX98357A amplifier chip in order to convert the digital audio signal to an analog signal to drive the speakers. It uses the I^2S standard to convert the signal, and then amplifies the signal. It is a class D amplifier which has the ability to deliver up to 3.2W of power into a 4Ω load. The breakout board used for testing is a fairly simple board with only a few pin connections needed to operate. During preliminary testing with the selected speakers (which will be detailed later in this document), this component worked exactly as expected and as desired, and was approved by the group for use in the final project prototype. It should be noted that although the breakout board was used in testing, the final design includes only the MAX98357A amplifier chip on our designed PCBs, with the PCBs designed to fulfill all needed functions that were provided by the breakout board during testing. The MAX98357A can be found on Digi-Key for $2.54 each, and there were plenty in stock at the time of writing this paper.

3.4.8.2 Speakers

In order to be compatible with the previously selected audio breakout board, the speakers chosen needed to have a maximum rated power of 3.2W and a rated impedance of 4Ω. We found speakers on Amazon called "MakerHawk 2PCS 4 Ohm 3 Watt Speaker," and ordered them for preliminary testing. Each speaker is 31 mm long, 28 mm wide, and 15 mm thick, and they each have a power rating of 3W and an impedance rating of 4Ω. The speaker usage interval equates directly to the trigger pull of the laser "gun" (when the laser-diode emits light the speakers will produce sound); 1.5Wh maximum is drawn from the batteries at full load, this load behavior is almost identical to the ESP-32 (HP) load profile calculated in section 3.4.2.2. For a better visual of what these speakers look like, their product image from Amazon is shown in Figure 25 below [47]. After testing both the speakers and the audio breakout board with the ESP32 (which will be detailed later in this document), the speakers produced clear, understandable audio, and were approved by the group for use in the final project prototype.

Figure 25: Product Image of Speakers [Courtesy of Amazon] [47]

3.5 Parts Selection Summary

Table 13 below gives a summary of all parts selected for each of the categories previously to be used in the final design.

Part	**Supplier**	**Part Name**
Microcontroller	Mouser	ESP32-WROOM-32D
Laser Diode	Adafruit	Laser Diode - 5mW 650nm Red
Laser Sensor	Adafruit	ALS-PT204-6C/L177
Battery	SCT	Efest 18650 Li-Ion 14.4V 2600mAh
3.3V Voltage Regulator	Mouser	UA78M33CDCYR
5V Voltage Regulator	Mouser	MC7805CDTRKG
Controller Display	Adafruit	Monochrome 1.3" 128x64 OLED graphic display
Target LED Array	Adafruit	Adafruit NeoPixel Digital RGB LED Strip
Audio Amplifier	Digi-Key	MAX98357AEWL+T
Speakers	Amazon	MakerHawk 4 Ohm 3-Watt Speaker

Table 13: Summary of All Parts Selected

4. Related Standards and Design Constraints

4.1 Standards

A standard, specifically a technical standard, can be defined as "…a document that specifies design, predicted performance, and operation and maintenance specifications for a material, device or method [48]." Standards are typically made and published by committees, organizations, or government departments. Examples of such groups include the Institute of Electrical and Electronics Engineers (IEEE), the Institute of Printed Circuits (IPC), and the United States Food and Drug Administration (FDA). Standards can be national, international, or restricted to a specific company or industry.

Identifying related standards is an important part of the design process for an engineering project. Standards help to "standardize" the devices or methods used during design, so that any given design or process used will be compatible with all other devices or methods that follow that same standard. Many standards also deal with safety for the user or manufacturer of a product or device. These kinds of standards are the most important to be aware of and adhere to.

This section will be used to discuss a number of standards related to this project, including standards for documentation, hardware safety standards, manufacturing standards, and software-related standards.

4.1.1 Requirements Specification Standard - IEEE 1233

IEEE Standard 1233 is also known as "IEEE Guide for Developing System Requirement Specifications." This standard provides guidance for the development of a set of requirements specifications for a system that will satisfy a stated need or multiple stated needs. In the standard, these requirements are referred to as System Requirements Specifications, shortened to "SyRS." The SyRS typically states the requirements of the customer in such a way that the people designing the system can then take those needs and design and build a system that meets them. It has to be written in terms that both the customer and the designers will easily understand, and it should only describe what the system should do, not the process of constructing the system.

The documentation for this standard contains a definitive list of the properties that a good set of SyRS should have. They are as follows:

- *Unique Set* – Every requirement should be stated only once.
- *Normalized* – Requirements should all be independent and should not refer to one another or overlap with one another.
- *Linked Set* – Relationships between all requirements should be defined, and it should be clear how all of the requirements together form a complete system that meets all stated needs.
- *Complete* – All requirements stated by the customer must be included, as well as any others that may offer a more complete definition of the system.
- *Consistent* – All requirements should be formatted consistently, and they should not contradict each other.
- *Bounded* – Boundaries, scope, and context for each requirement should be given.
- *Modifiable* – The requirements should be able to be modified as needed.
- *Configurable* – Requirements should be configurable through time and through new versions.
- *Granular* – System requirements should be able to be broken down into specific distinguishable parts.

The document also breaks down the intended use of the SyRS. During the design process, the requirements are sorted into hardware, software, and other categories. The task of meeting those requirements is then assigned directly to the groups in charge of those specific categories. The SyRS are applied directly during the construction process, and they are also used during the testing process to write test plans for the hardware and software of the system, as well as the completed system [49].

While writing up our system requirements for this project (listed in the Requirements Specifications section of this report), we made an effort to adhere to IEEE Standard 1233, ensuring that all of our requirements had the properties listed above. Because this project was self-motivated and self-funded by all group members and had no direct customer, when deciding what requirements to include, we did our best to anticipate the needs and wants of the average user of this system by considering what we would want from the system if we were someone wanting to purchase it. As young college-age adults, we still fit within the expected consumer demographic, and therefore the requirements should be very similar to what a hypothetical customer would require.

4.1.2 Laser Standards - FDA & IEC

Unlike normal lights, the wavelength of a laser light is purposely amplified. This amplification results in a focused narrow beam of light that can be emitted in a single direction. As discussed previously in the Relevant Technologies section of this report, this amplification is actually the result of stimulated emission of radiation, which can be harmful to humans. Light, when concentrated to a small area, amplified, focused, and pointed in a single direction, achieves a very high intensity, even at a significant distance from the laser. Lasers have been classified for safety based on their output power levels, and their potential at each power level to cause injury to a person's eyes and/or skin. These classifications are recognized by the U.S. Food and Drug Administration (FDA), as well as the International Electrotechnical Commission (IEC). Both organizations recognize four major hazard classes (I to IV for FDA, 1 to 4 for IEC), as well as a few subclasses (IIa, IIIa, and IIIb for FDA, and 1M, 2M, 3R, and 3B for IEC). Labeling for classes II-IV are required to have a warning symbol that gives the output power and class of the laser. The different laser classes are outlined below in Table 14, alongside examples of products they are used in, and a description of the hazard that each class presents to the human eye or skin. Though these classes are not explicitly stated as "standards," they are essentially standards, and were considered as such for the purpose of this project [50].

FDA Class	IEC Class	Laser Product Hazard	Product Examples
I	1, 1M	Considered non-hazardous. Hazard increases if viewed with optical aids, including magnifiers, binoculars, or telescopes.	• Laser Printers • CD players • DVD players
IIa, II	2, 2M	Hazard increases when viewed directly for long periods of time or if viewed with optical aids.	• Bar code scanners
IIIa	3R	Depending on power and beam area, can be momentarily hazardous when directly viewed or when staring directly at the beam with an unaided eye. Risk of injury increases when viewed with optical aids.	• Laser pointers
IIIb	3B	Immediate skin hazard from direct beam and immediate eye hazard when viewed directly.	• Industrial lasers • Research lasers
IV	4	Immediate skin hazard and eye hazard from exposure to either the	• Laser light show projector • Medical device lasers

		direct or reflected beam; may also present a fire hazard.	

Table 14: Breakdown of FDA/IEC Laser Classes [50]

Figure 26 below shows each laser class and its output power range compared against a scale illustrating the eye injury hazard that they present. Lasers are only at a low risk of causing eye injury if their output power is 5mW or less, a fact that was taken into account during the laser diode selection process for this project [4].

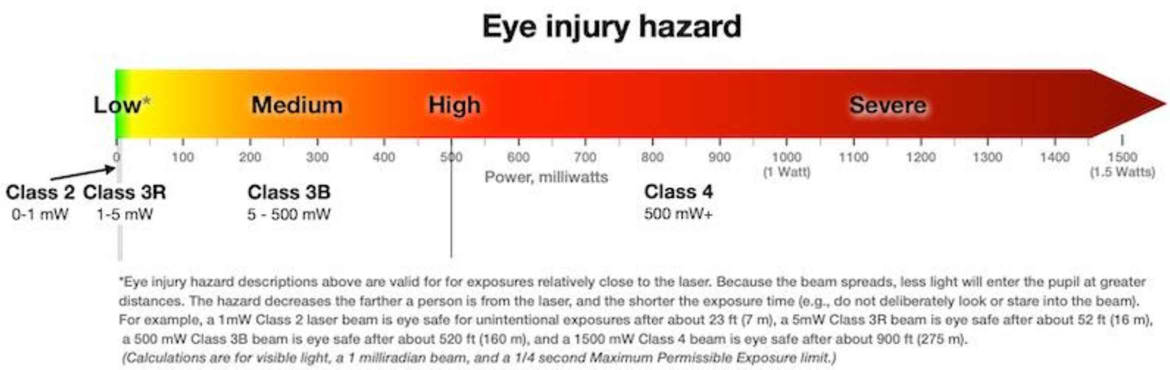


Figure 26: Laser Classes & Eye Injury Hazards [Used with Permission] [4]

For this project, it was absolutely imperative from the start that we chose a laser diode for our controller that was safe for an informed person to use, and by extension, was safe for them to use around other people. We wanted the level of risk to be no more than that of a laser pointer. A laser pointer is listed in Table 14 above as being a Class IIIa/3R laser, so only laser diodes of 5mW or less output power were considered during the selection process. This was narrowed down to only lasers of Class II/2, IIa/2M, and IIIa/3R being considered, since Class I/1 was deemed too low-power.

4.1.2.1 Class IIIa/3R Laser Details

Our project utilized a Class IIIa/3R laser diode to emit the laser light from the controller. This section will be used to discuss that particular class of laser in-depth, to show that we researched it thoroughly before including it in our project. To avoid confusion and unnecessary letters and slashes, it will be called by its IEC class (3R) for the remainder of this section.

Output power for Class 3R visible light lasers is 1-4.99 mW, which is considered low power, and perfectly safe when handled carefully. In the United States, they can be sold as laser pointers, and this is their most common use. They are not harmful to the eyes for a momentary exposure of less than 0.25 seconds, which is within the human aversion response time, at which point a person would turn away and/or blink to avoid continued exposure to a bright light. Laser protective eyewear is usually not required for a Class 3R laser, and it also cannot burn skin or materials. Staring directly into the beam for an extended period of time can be dangerous for the eyes, so it is best to avoid. Small children can use a Class 3R laser safely, but only with adult supervision [51].

4.1.3 Wireless Local Area Network Standards – IEEE 802.11xx

The IEEE 802.11xx standards are a series of different Wireless Local Area Network (WLAN) standards or protocols used by devices to communicate wirelessly with other devices. To distinguish the different standards from one another, the "xx" at the end of IEEE 802.11xx is replaced by one or two lowercase letters that identify the specific standard. Each specific standard has a purpose, sometimes upgrading from a previous standard, expanding capabilities, adjusting existing standards for use in different countries, or addressing security issues in previous standards. These standards dictate parameters such as data throughput and range of wireless communications, as well as frequency bands used.

The very first WLAN standard was released in 1997. This was the original 802.11. It was developed to operate on the 2.4 GHz ISM band and supported speeds of 1 Mbps - 2 Mbps, much slower than modern speeds. This original standard was the foundation on which all other IEEE 802.11xx standards were built. Now, there are over 40 total IEEE 802.11xx, each one an improvement over the previous, and several new standards are currently in development.

This standard was extremely relevant to our project because the controller and targets need to be in constant wireless communication with each other while the system is operating. This is required in order to send and receive controller input message, target hits, battery statuses, and other necessary gameplay data. Our chosen microcontroller, the ESP32 series chip, supports IEEE standards 802.11b, 802.11g, and 802.11n, the details of which are described in Table 15 below [52].

Standard Name	Alternate Name(s)	Year Released	Description
802.11b	Wi-Fi 1/ 802.11 High Rate	1999	Cheaper than previous standards (such as 802.11a), uses the 2.4 GHz band, speeds of up to 11Mbps.
802.11g	Wi-Fi 3	2003	Combines the advantages of 802.11a & 802.11b to achieve speeds of up to 54 Mbps on the 2.4 GHz band.
802.11n	Wi-Fi 4/ Wireless-N	2009	Makes use of both the 2.4GHz band and the 5GHz band and achieves speeds of up to 600 Mbps.

Table 15: IEEE 802.11xx Standards Supported by ESP32 Series Chip

4.1.4 IPC PCB Standards

IPC is an organization that provides industry standards for the assembly and protection of electronic equipment. The American National Standards Institute (ANSI) has accredited IPC as a standard-developing organization. It has more than 3,000 member companies in different areas of the electronics industry around the world, including design companies, suppliers, board manufacturing companies, assembly companies, and original equipment manufacturers. IPC currently has over 300 active standards and more than 1,000 standards in its resource library, all of which are used by electronic designers worldwide. The standards are written, edited, and voted in by committees of volunteers, which are made up of over 3,000 industry professionals all around the world.

IPC was founded in 1957 as the Institute for Printed Circuits, and later changed its name to the Institute for Interconnecting and Packaging Electronic Circuits when it expanded its services to include packaging and electronic assemblies from bare boards. Then in 1999, it began to use the current name IPC, with the tagline of: “Association Connecting Electronics Industries”.

IPC has a number of standards for PCBs. So many, in fact, that there is an IPC standard for every stage of the PCB production process, including the design, manufacturing, and testing processes. Adhering to IPC standards throughout the process helps to produce safe, reliable, high-performing PCB products [53].

IPC’s PCB standards were relevant for this project. They were followed during the design of our PCBs to ensure that they were of the highest quality, worked, and were compatible with all components that were attached to them. Most IPC standard documents are able to be purchased online from the IPC website. Unfortunately, most of them cost at least a hundred dollars, which did not fit into our project budget as college students. However, one standard that we could get some documentation for, and that appeared to be most applicable to our project and covered the most stages of the PCB design process, was IPC-221A. This is a standard that “…establishes the generic requirements for the design of organic printed boards and other forms of component mounting or interconnecting structures.” This standard covers important topics including but not limited to materials selection, mechanical/physical properties, electrical properties, thermal management, component and assembly issues, holes/interconnections, general circuit feature requirements, and quality assurance [54].

4.1.5 Soldering and Component Mounting Standards

J-STD-001 is another IPC standard, one which applies to soldered electronics and electrical equipment. This standard was originally released in 1992 as J-STD-001A, and it has received a number of amendments over the years, with its current form being J-STD-001H. It establishes the best soldering practices to ensure the highest quality and reliability of the soldered product. The standard gives a thorough explanation of the following important considerations for soldered products:

- Material, components, and equipment
- Soldering and assembly requirements
- Terminal and wire connections
- Through-hole and surface mounting of components
- Cleaning and residue requirements
- Coating, encapsulation, and adhesives

As with several other standards created by IPC, most versions of J-STD-001 can only be viewed for payment. However, there is one version called IPC J-STD-001ES, that can be viewed and downloaded from the IPC website for free and contains a number of relevant soldering standards. Information from this standard as well as other relevant research on good soldering and component mounting practice will be detailed in the following section.

We were sure to follow all these standards and best practices when we soldered our components to our PCBs [55].

4.1.5.1 Best Practices and Standards

Solder: Solder alloys permitted by J-STD-001ES are Sn60Pb40, Sn62Pb36Ag2, or Sn63Pb37. High temperature solder alloys such as Sn96.3Ag3.7 may be used only where specifically shown by approved engineering drawings. All solders listed in the standard are alloys of tin combined with silver and/or lead. A comparison of lead solder versus lead-free solder will be done below. The standard also states that other solder alloys that are of the same level of quality may be used if all other standards are met and all evidence of quality is reviewed and approved by the user prior to its use.

Lead Solder vs. Lead-Free Solder: Both lead solder and lead-free solder have pros and cons that will need to be weighed before making the final decision on which solder to use. Lead solder is still in use in the United States, although it is declining in use due to safety concerns, especially since it was banned in most consumer electronics sold in the European Union in 2006. Lead solder has been used heavily in PCB production because it cools more slowly than other metals, causing less joint cracking. It also wets joints well, providing a good electrical connection, and it has a lower melting point than any lead-free alternative, meaning it is less likely to damage heat-sensitive electronic components. As our group wished to extend our RoHS compliance beyond components to solder, a lead-free solder needed to be used. The main benefit to lead-free solder is that it is safer, but its main drawback is that it does not have a stable melting temperature, and its melting range is higher than that of lead solder, which can damage PCBs and electronic components [56].

Flux: If used, flux should be in accordance with IPC standard J-STD-004: "Requirements for Soldering Fluxes," or an equivalent standard.

Soldering Tools and Equipment: Soldering tools and equipment should be selected, used, and maintained in such a way that their use will not damage or degrade components in a way that would prevent them from performing their intended functions.

Lighting: The surface of workstations used for soldering should be well-lit, and the standard states that they should be illuminated to at least 1000 lumens per square meter.

Thermal Protection: The heat sensitivity of a component should be identified before it is soldered. When hand soldering, tinning, or reworking a heat sensitive component, measures need to be taken to protect the component. Examples given in the standard of measures that include: a heat sink, a thermal shunt, or preheating.

General Part Mounting Requirements: A number of different part mounting requirements are given in this section of the standard, so they will be listed briefly below:

- All components should be mounted and soldered using a process compatible with that specific part, especially if the part is temperature sensitive.
- Parts should be mounted with sufficient clearances between themselves and the PCB in order to make adequate cleaning possible.
- When both through-hole and surface mounted components are used on one PCB, all through-hole components should be mounted on a single side of the PCB, while surface mounted components can go on either side.

- Parts should be mounted so that their markings and reference designators are visible.

Soldering Defects: The following are listed in the standard as solder connection defects: fractured solder connections, disturbed solder connections, cold or rosin solder connections, solder that violates minimum electrical clearance or contacts the component body, solder bridging between joints (except when it is by design), overheated solder connection, blowholes and pinholes, excessive solder, insufficient solder, and contaminated solder. All of these defects should be avoided at all costs to ensure the best quality of the PCB, and to ensure that all components work properly when soldered to it [57]. Both electrical engineering students in our group, Rachel and Jamauri, have worked as interns in an electronics lab, and are well-acquainted with proper soldering techniques and how to avoid soldering defects.

4.1.6 Programming Language Standards – C++

Every single software program that is written must be written in a programming language. In order for the language to be uniform, as well as universally understood by programmers, students, and most importantly, the compiler, computer, or chip receiving the programming, the language must be standardized. The American National Standards Institute (ANSI) and the International Organization for Standardization (ISO) have developed a number of programming language standards. We adhered to these standards throughout the entire software design process for our project.

Originally, there was some uncertainty about whether we would be using C or C++ for our primary programming language for the software of this system. Both languages are equally compatible with the ESP32, our chosen microcontroller. However, after some comparison, we chose to use C++ for testing and developing the software for our system. This is because C++ is essentially a superset of C, meaning that C++ has all of the features of C, but with several additional features. A few of the major C++ added features are object-oriented programming, exception handling, and a rich C++ library [58].

The current and most up-to-date ISO C++ standard is known officially as ISO International Standard ISO/IEC 14882:2020(E) – Programming Language C++. It was published in 2020. Unfortunately, the only way to access this standard is by paying for it, but its abstract can be found online for free. The abstract states that this standard gives requirements for implementing the C++ language. It describes C++ as “a general-purpose programming language based on the C programming language”. It also explains, as mentioned earlier, that C++ provides many features beyond those provided by C, including additional data types, classes, templates, exceptions, namespaces, operator overloading, function name overloading, references, free store management operators, and additional library facilities [59].

Because we could not access the entirety of the exact ISO standard without spending money outside of our project budget, we found a page detailing a number of guidelines for best practices for C++ that were prepared by Luan Doan-Minh for Rational Software Corp.. These guidelines were based on the C++ standard. We followed these guidelines while developing the software for this project. The intention of these guidelines is to foster the development of robust, readable, and easy-to-maintain code, and to establish a project-wide programming style, which is especially useful for projects such as ours where there is more than one person doing the software development.

The page lists three fundamental principles for creating clear, understandable C++ source code. They are as follows:

1. Minimal Surprise Principle: Ideally, source code should be written so that it reads like an English language description of what it is doing, with the added benefit being that it will execute when it is run. Programs are written more for the benefit of people than computers, and source code is read much more often than it is written. Reading code is a mentally taxing process that can be eased greatly by uniformity in the code, also referred to in the guidelines as the "minimal surprise principle." A uniform style across an entire project is a huge asset to the software development team, and a major reason for a programming standard to be agreed upon.
2. Single Point of Maintenance Principle: Whenever it is possible, a design decision should be expressed at a single point in the source code, and most (if not all) of its consequences should be derived programmatically from that single point, thus creating a "single point of maintenance" for that design decision. If this principle is violated, the maintainability, reliability, and understandability of the code will be greatly reduced.
3. Minimal Noise Principle: Lastly, and most importantly for code legibility, the "minimal noise principle" should be observed. Following this principle means making an effort to avoid cluttering the source code with visual "noise," such as bars, boxes, and other text that contains minimal information or information that does not contribute to the reader's understanding of the purpose of the code.

The page also includes guidelines in much greater detail for code organization and style, comments, naming, declarations, expressions and statements, special topics, portability, reuse, and compilation issues. All of the details contained within are too much to include in this document, but they have been read and are understood by the group members to whom they are relevant [60].

4.2 Design Constraints

When an engineering design project is being planned, it is of the utmost importance to consider any constraints that the project may be placed under. In the context of this project, a constraint can be defined as a rule, requirement, or limiting factor placed on the design as part of the project assignment, project timeline, or the environments in which the completed system must be tested and then eventually presented as a final product.

Care was taken to ensure that all constraints were realistic for the group to successfully follow, and to make sure that all the constraints worked well together and did not conflict with one another.

During the ABET lectures given in our Senior Design course, several categories of constraints were introduced. Each one of these categories was researched thoroughly in order to come up with relevant constraints for this section of the document.

This section will be used to discuss all known constraints for this project, along with the reasons why these constraints apply. Some of the constraints in this section may be the result of standards mentioned in the above section, as existing standards often impose their own constraints on a design.

4.2.1 Economic Constraints

This project was almost entirely self-funded by the members of the group. This provided the group with the benefit of having no strict budget to follow, but also the responsibility of paying for all items within our own personal budgets. Taking into account all costs for our project, including parts used in the actual project design, parts used only for testing, surpluses of parts ordered to account for supply chain issues, and shipping, early estimates put the cost of our project at around $500 - $800. As such, our budget was capped at $800.

Another consideration for the economic constraints was the price of similar products. A rechargeable home laser tag game, such as the “Battle Action Laser Tag” product (mentioned previously in Section 3.1 as a similar product) is sold online for around $300. Since our system is mostly hand built and hand designed, serves a different purpose, and makes improvements on that system, it is reasonable that it would cost more. Taking into consideration that our budget included shipping and other excess parts, which could be significantly lowered if our system were made into a commercial product, our maximum cost was not substantially higher. This was important for our group to consider, in case we ever wish to push for our project to be considered as a marketable consumer product.

Economic constraints had the most impact on which components were chosen for the design, the quality of the components, and the shipping speed of some components, as shipping is a price that is not often considered, but does add up quickly. All components chosen needed to be reasonably priced, and such components may not have had all the extra features that a more expensive component might have offered. The group was able to work with the components that we could afford within our personal economic constraints.

4.2.2 Time Constraints

Perhaps the most pressing constraints for this project were its time constraints. It was limited to the duration of the group members’ Senior Design I and Senior Design II classes. Each class is the length of a semester, about 16 weeks each, with a month off in the middle between semesters. The group did some work during the break, particularly on the PCB design, but not as much as during the actual semesters. This gave the group a grand total of approximately 8 months to complete the project, from the conception of the idea to the final working prototype being presented.

There were also rigorous documentation timelines that needed to be met throughout the course of both semesters. As the project was being designed, the entire process had to be documented in this paper in no less than 120 pages by the end of the Senior Design I class. Further updates to this paper needed to be completed by the end of the Senior Design II class.

The time constraints had a massive impact on the project as a whole. They were a determining factor in which components were used for the design, since components on backorder or components with long or unreliable shipping times had to be taken out of consideration. They also affected the overall quality of the project due to there being a limited amount of time to test, double-check, or fix designs. Our group had to make the most of every opportunity to test each stage of the design while there was time, because fixing the final implementation of the project would have been more time consuming than catching a problem before it was implemented. Achieving stretch goals for the project, such as adding additional gameplay modes, additional features to the controller or targets, or building more than a couple of targets, depended entirely on how much time we had left when the prototype was completed in Senior Design II. Due these time constraints, we were not able to get to our stretch goals by the end of the Senior Design II semester. We leave these goals for any future work the group wishes to make on the project later.

4.2.3 Environmental Constraints

4.2.3.1 RoHS Compliance

RoHS compliance is a very important environmental constraint, and we decided as a group to prioritize RoHS compliant components for our project. RoHS stands for "Restriction of Hazardous Substances," and is also known as Directive 2002/95/EC. It has existed in the European Union since 2002, and it restricts the use of a number of different hazardous materials in electronics and other electrical products. RoHS compliance is important for the environment because the materials that it restricts are hazardous to the environment and they pollute landfills.

RoHS in its original iteration went into effect on July 1, 2006, after which all products to which it applied in the European Union had to comply with it. This original iteration restricted six hazardous substances. In 2011, RoHS 2 (or Directive 2011/65/EU) was published, and it went into effect on January 2, 2013. This update to RoHS expanded its coverage to include all electrical and electronic equipment, cables, and spare parts, as well as requiring that a product comply with RoHS 2 before being allowed to have a CE marking placed on it to indicate that it complies with Europe's health, safety, and environmental standards. Most recently, in 2015, RoHS 3 (or Directive 2015/863) was published, which added four additional restricted materials to the original list of six. RoHS 3 went into effect on July 22, 2019. Materials chosen for this project were checked for RoHS 3 compliance in order to be as environmentally-friendly as possible.

A complete list of all ten materials restricted by RoHS 3 and their chemical symbols or names, as well as the maximum levels allowed of these materials (in units ppm, which means "parts per million" or can also be converted to milligrams/liter) is shown in Table 16 below [61].

Material	**Chemical Symbol/Name**	**Maximum Amount Permitted (ppm)**
Cadmium	Cd	< 100
Lead	Pb	< 1000
Mercury	Hg	< 1000
Hexavalent Chromium	Cr VI	< 1000
Polybrominated Biphenyls	PBB	< 1000
Polybrominated Diphenyl Ethers	PBDE	< 1000
Bis(2-Ethylhexyl) phthalate	DEHP	< 1000

Benzyl butyl phthalate	BBP	< 1000
Dibutyl phthalate	DBP	< 1000
Diisobutyl phthalate	DIBP	< 1000

Table 16: Hazardous Materials Restricted by RoHS 3

Our choice to prioritize RoHS compliant materials led us to the decision not to use photoresistors in our design for the laser sensor in the targets, because all of the photoresistors that we could find were made with cadmium, which means they were not RoHS compliant.

4.2.3.2 Recyclability

Recyclability in this context refers to the ability to dismantle a product and reuse its parts in another product. In order to be as environmentally friendly and as waste-free as possible, we designed our project and chose our components such that if an initial prototype did not work, it could be easily dismantled without damaging the parts, and therefore the same parts would be available to use for the next prototype, with minimal materials having to be thrown away between prototypes.

4.2.4 Social and Cultural Constraints

Because the purpose of this project was to create a shooting-range-type game that could be played virtually anywhere, this meant that the game's controller had to be in the shape of a firearm. Obviously, this could have been a cause of concern for several reasons. We needed to meet in a lab in person to work on the prototype, and the final completed project also needed to be on campus for our final presentation. Also, anyone testing or using this product, especially if they were outdoors, risked encountering people who were not aware of what it was.

Therefore, it was of the utmost importance that the controller "gun" was different enough from a real firearm in appearance that it would not cause concern to the average observer. Care was taken to ensure that the coloring and shape of the controller "gun" did not resemble that of a real firearm too closely, and to ensure that anyone in the immediate area was aware of what it was. A bright white color was chosen for the controller's body, and the barrel of the gun was made shorter than typical firearms, indicating that it was a toy.

4.2.5 Political Constraints

After thorough research and a careful review of the definition of political constraints, no known political constraints were identified for this project.

4.2.6 Legal Constraints

The design, development, and testing of the system abided by the procedures outlined in UCF's "Weapons on University Property and at University Events" policy (UCF Policy 3-119.2). The policy states that "The University of Central Florida prohibits the possession, use, or storage of weapons on property owned or controlled by the University of Central Florida, including in a university vehicle, on one's person, or in one's office or residence hall. Additionally, the University of Central Florida prohibits the possession, use, or storage of weapons at events sponsored or hosted by the University of Central Florida, except as outlined in this policy." Then the document goes on to list a number of exceptions. Exception number 6 states that "Simulated weapons that are clearly identifiable (whether through design or decoration) to the casual observer as simulated are permitted for instructional or academic purposes" [62]. This went right along with the plan discussed in the Social and Cultural Constraints section to design and color the controller "gun" in such a way that the average observer would not perceive it as a real weapon, or as a threat.

We were also willing to coordinate with on-campus law enforcement to notify them when we were constructing, testing, and presenting our device on campus.

4.2.7 Health and Safety Constraints

4.2.7.1 Health and Safety with Lasers

The system should not expose operators to harmful laser radiation or cancerous materials. The laser used in the "gun" controller was no more harmful to the user or more powerful than the average laser pointer that is most often used for entertainment or presentation purposes. As discussed in the Standards section of this paper, the FDA has class ratings for laser safety. The laser diode used in the "gun" controller met the requirements of an FDA Class IIIa or Class II/IIa device.

Just because the Class IIIa and below lasers are safe for most everyday uses does not mean that they are safe for direct eye exposure. The user should never stare directly into the beam emitted by the "gun". Since we had planned to assemble this device and perform many tests that could have accidentally caused us to look directly into the emitter end of the laser diode, laser safety glasses were purchased from Amazon. These glasses block red and infrared laser light, and they are compatible with 405nm, 445nm, 450nm, 635nm, and 650nm laser light, and their visible light transmittance is 30%. They kept us well protected from unsafe eye exposure to lasers.

4.2.7.2 RoHS Compliance

RoHS compliance is also a health and safety constraint because the materials that it restricts can be harmful to people as well. People can be affected by them through occupational exposure during the manufacturing and recycling processes. Therefore, to preserve the health and safety of people who may be manufacturing or recycling these parts that we are using, it was our ethical responsibility to use as many RoHS compliant parts as we could [61].

4.2.7.3 Ergonomics/Lifting and Carrying Safety

In order for the controller and the targets to be ergonomic for the user to hold, carry, and move around to different locations, they needed to be light. As listed in the Requirements Specifications section, the goal was for the controller to weigh no more than 10 pounds, and for the targets to weigh no more than 5 pounds each. These were safe weights for the average user to lift and carry. The outer housing of all parts was also designed with ease of holding and carrying in mind.

4.2.8 Manufacturability and Maintainability Constraints

4.2.8.1 Maintenance and Modification Access

The targets and the controller needed to be able to be accessed for maintenance and modification without destructive manipulation of the product. Since this was a senior project resulting in a working prototype as the final product, the prototype needed frequent modification and maintenance. Therefore, it was necessary that all parts could be accessed at any time without being damaged.

4.2.8.2 Solo Operation

The devices operate without a mandatory connection to an online service or a mobile app. This was very important for the overall maintainability of the system. If the system relied on an online service or app, and that app or service was eventually discontinued, then the entire system would become worthless. This often happens with consumer electronic products, and it is extremely frustrating when a product that you (the consumer) spent money on and enjoyed using suddenly becomes worthless because the online service it relies upon is no longer available. We approached this design as if it were a marketable product, and we wanted to avoid this kind of frustration. Our system works completely independently of any application or service so that it can be played anywhere, anytime.

4.2.8.3 Ease of Updating Firmware

Downloading firmware updates to each device needed to be possible to allow for future improvements. This was also important for the overall maintainability for the system, both in the context of the system as a design project, and also in the context of the system as a marketable consumer product. With our design, it would be easy for a hypothetical average consumer to download firmware updates if this item were to be made available to consumers.

4.2.9 Usability Constraints

The controller and targets use tactile buttons for functions such as on/off, menu selections, and other inputs. This was chosen instead of having a touchscreen for some of these functions. In addition to the reasons previously stated in the parts selection section, we decided on this constraint for two reasons: cost and user accessibility. Buttons are much less expensive than touchscreens, so they were easier on all group members' personal budgets and fit within the economic constraints of the project. User accessibility was a consideration because touchscreens, especially those small enough to fit on a handheld laser "gun", can be difficult to operate, especially for a person who may have trouble seeing or reading small text on a screen.

5. Project Hardware and Software Design Details

5.1 Overall Hardware Testing

This section contains descriptions of the various hardware tests we conducted throughout the course of Senior Design 1. Within each description, the objectives and motivations of the test will be discussed, along with all the supplies and materials needed, the setup of the test (including the testing environment), and lastly, the results of each test will be presented and discussed. The goal for these tests as a whole was to test each major component individually (which also helped with the component selection process), and then to combine a group of components together to test the most basic subsystems of the overall system in order to determine that the base functions of the system could be performed successfully by all selected components. These tests were carried out over a period of a few weeks within the last couple of months of Senior Design 1. Any descriptions of tests planned for after Senior Design 1 is completed and into Senior Design 2 will be discussed in a later section.

5.1.1 Laser Diode and Phototransistor Test

Objectives/Motivations for Test: This test was performed with two goals in mind. The first goal was to determine if the laser diode and phototransistor that we ordered initially were fit to be used in the final design. For this goal, both the laser diode and the phototransistor were to be powered on and put through their basic functions. For the laser diode, we were expecting to see a laser beam output, and we wanted to determine what input was best to use to produce the brightest and highest quality laser light beam output. For the phototransistor, we were expecting to see an output voltage across a resistor connected to it in series, and to determine if that output changed sufficiently in response to the intensity of the ambient light in the room. If these expectations were met by the laser diode and the phototransistor, this would mean that both components are able to perform their basic functions successfully.

The second goal was to test the most basic subsystem of the design: the laser and receiver. For this goal, we planned to set up a test in which the laser diode would be aimed at the phototransistor and could be turned on and off quickly and easily, and the output from the phototransistor could be easily recorded as changes were made to the light around it, and as the laser diode was pulsed at it. We wanted to determine if there was a significant enough difference between the phototransistor's output in response to changes to ambient light in the room compared to pulses from the laser. Additionally, we wanted to test this at a close range and a farther range.

Materials and Supplies Needed for Test:

- Laser diode (Adafruit #1054) and phototransistor (HW5P-1) – the components being tested
- DC Power Supply – for powering on the laser diode and the phototransistor. Model: Aim-TTi MX100T
- Oscilloscope – for measuring and recording the input to the laser diode, as well as the output of the phototransistor (across a series resistor) and recording comparisons between both as the laser was pulsed and the ambient light was adjusted. Model: SDS1202X
    - Probes: 50m Ohm 1x magnification (tuned)
- 2 Breadboards – one breadboard for the phototransistor setup, and one for the laser diode setup
- Adjustable laser diode stand – used to fix the laser diode in place so it did not need to be aimed manually
- Assorted resistors, potentiometers, pliers, screwdrivers, other miscellaneous tools used to set up the test

Testing Environment: This test was conducted atop a workbench in an engineering lab in a warehouse-like space. The workbench had multiple lights of different brightness atop it that we could switch on and off in order to test the response of the phototransistor to the laser diode with different exterior lighting levels around it. The room could not go fully dark, as the lab was in use by multiple people besides just our group, so there was always some light hitting the sensor, which may have caused some interference when the phototransistor was picking up the laser. However, since this system is not intended to always be played in a dark room anyway, this interference was a welcome part of the overall testing experience, as it provided a more accurate representation of the possible light levels in the room when the game will be played by the user. The additional lights on the workbench were also used for this purpose, to test different levels of exterior light and determine if the laser light was still distinct and discernible, even with all the workbench lights switched on.

Test Setup:

- Individual Laser Diode Test: To test the laser diode individually, we simply connected its red wire to a DC power supply, and its black wire to ground from the same supply. An oscilloscope was set up to measure the input to the laser diode. However, we learned the hard way during testing that a laser diode should not be driven directly from a power source, so for the final testing setup we ended up adding a potentiometer in series with it to determine what resistance was needed there.
- Individual Phototransistor Test: To test the phototransistor individually, it was placed in series with a 1kΩ resistor and powered on by a DC source, and the output voltage across the resistor was measured by an oscilloscope.
- Close-Range Test: For the close-range test, where the laser diode was a few inches away from the phototransistor, the entire test was set up on one large breadboard that was provided to us by UCF's ECE department. This test combined the individual test setups for both the laser diode and the phototransistor, but this time with the laser diode aimed directly at the phototransistor. This setup can be seen in Figure 27 below.

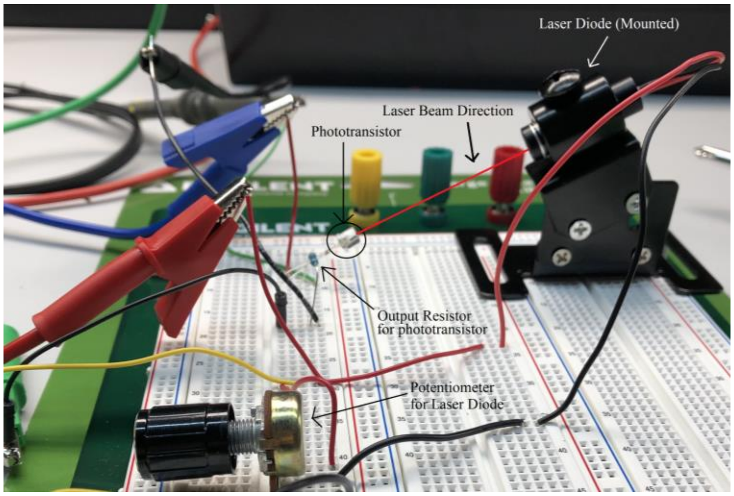


Figure 27: Close-Range Laser and Phototransistor Test Setup

- Long-Range Test: For the long-range test, the entire system was the same as for the short-range test, but we separated the laser diode and its potentiometer onto a different breadboard and moved the laser diode and the phototransistor breadboards as far apart from each other as the available power supply wires we had would allow. We measured the distance with a measuring tape, and it turned out to be approximately 6 feet 7 inches from the emitter end of the laser diode to the top of the phototransistor's lamp. The laser diode was once more aimed at the phototransistor from this distance, and then fastened firmly in place by the stand it was mounted on as well as a clamp fastening it to the workbench, so that its beam did not move off of the phototransistor. An oscilloscope was once more connected to the phototransistor's series resistor in order to measure the output as the laser is pulsed from this new distance.

Test Results:

- Individual Laser Diode Test: As mentioned in setup, we started this test with the laser diode connected directly to the DC source with no current limiting of any kind. As it turns out, this was a big mistake because laser diodes require a constant and controlled current. If the forward current is too high beyond the device's maximum rating the laser diode will burn out. This happened in our first test. Initially its input was set to 2.8V with a no current limit. The laser turned on, and it was very bright and the red "dot" from its beam could be seen very clearly all the way across the warehouse lab. We were about to consider the test a success when the diode suddenly died. Luckily, we had bought a backup, so once we realized that we needed to limit the current, we hooked the backup diode to the power supply with the same 2.8V input, but this time with a current limit and in series with a potentiometer. We gradually raised the current limit and adjusted the potentiometer until the laser output light was as strong and bright as we desired. The current limit we stopped at was 13mA, and the potentiometer value was 1kΩ. These values will be kept in mind as the power distribution for the laser diode is being designed for the PCB.
  Figure 28 below shows the oscilloscope measurement for the input to the laser diode (purple) at (left to right) 2.8V, 3.3V, and 5V. The very obvious peak in the wave at turn-on was likely the source of the burnout of the original diode. Since the peak was lowest at the 2.8V input, we decided to use only 2.8V for powering on the diode after this test. Some kind of driver or protection circuit will likely need to be used in the final hardware design to keep the laser diode from burning out.

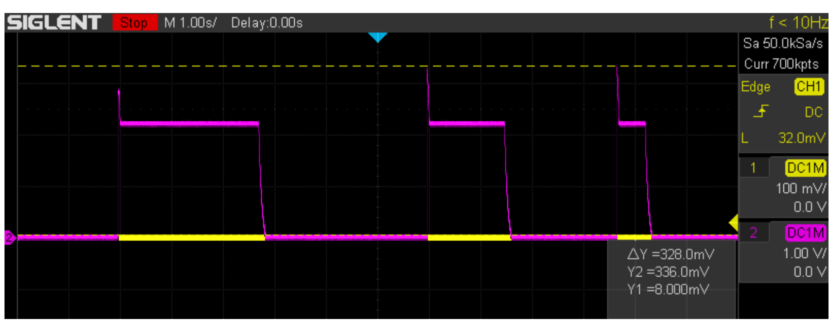


Figure 28: Laser Diode Input at 2.8V, 3.3V, & 5V

- Individual Phototransistor Test: This test, unlike the laser diode test, met all of our expectations. Once powered on with a DC source at 3.3V, the phototransistor produced an output across the 1kΩ series resistor that changed directly proportionally to the amount of light shining on it. Figure 29 below shows the output across the resistor as measured by the oscilloscope (yellow). It began powered off, and then it was powered on (the first step up), and then one at a time, multiple lights on the workbench of varying levels of brightness were turned on and left on (the remaining steps up and then the final highest reading) until the workbench where the phototransistor was located was extremely bright, giving us the response that the phototransistor would have in a very bright room. This test proved that the phototransistor performs its job of light sensing and providing a proportional output and a quick response time in a way that could be measured by the oscilloscope; we can consider this our noise floor for an indoor environment. This reading will also be useful in the combined test with both the phototransistor and the laser diode because we can compare the difference between the maximum output magnitude of the laser shining on the phototransistor in conjunction with the maximum output magnitude from the simulated bright room. This will ensure that there is a significant enough difference to be picked up by our system and registered as a target hit if the user is playing the game in a very bright location.

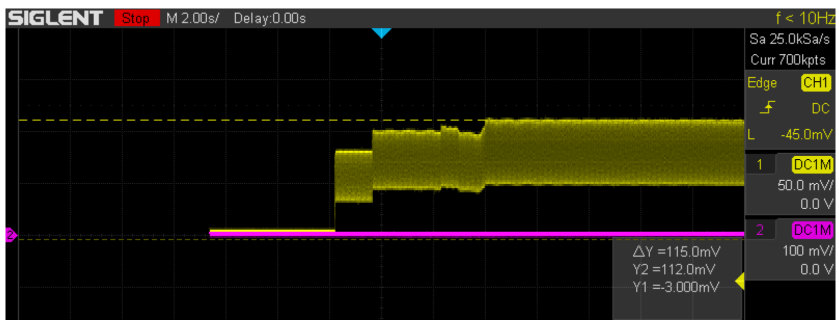


Figure 29: Phototransistor Output Across Varied Ambient Light Levels

- Close-Range Test: For this test, we wanted to see the response of the phototransistor to the laser diode pulsing, and also to determine what voltage to use to drive the phototransistor to best suit our purposes, since its datasheet specified a supply voltage of anywhere between 3-15V. The room lights and some of the workbench lights were left on in order to simulate a normally lit room. Supply voltages tested for the phototransistor were: 3.3V, 5V, 12V, and 13.5V. The procedure for the test at each voltage was as follows: starting with the laser diode on and the phototransistor off, the phototransistor was then powered on, and then the laser diode was pulsed off-on-off to show the response of the phototransistor to a single laser pulse. Once this data was recorded in the oscilloscope, we used it to measure the on and off response times of the phototransistor (extremely quick at every voltage level), as well as the signal-to-noise ratio (SNR) of the phototransistor at each voltage level, which would indicate the difference in the reading between the ambient light in the room and the light from the pulse of the laser diode. The response times were so short for all voltage levels tested (the wave was basically in sync with the laser diode wave), that the response time was not a determining factor for the voltage chosen. The signal-to-noise ratio, however, was significantly different at each of the voltage levels. The higher the supply voltage to the phototransistor, the higher the signal-to-noise ratio. We ultimately decided that a supply voltage of 12V gave the best response, with a signal-to-noise measurement of nearly 1V (984mV, to be exact), which was higher than those of the 3.3V and 5V measurements. The signal-to-noise measurement for 13V was higher, but its signal was distorted and not as clean as the 12V signal. Therefore, 12V was decided on as the best supply voltage for the phototransistor, because it provided the cleanest signal with nearly the best signal-to-noise ratio. The oscilloscope measurement for this test at a phototransistor supply voltage of 12V (with the SNR measurement) is shown in Figure 30 below, with the laser diode input signal shown in purple and the phototransistor output signal shown in yellow.

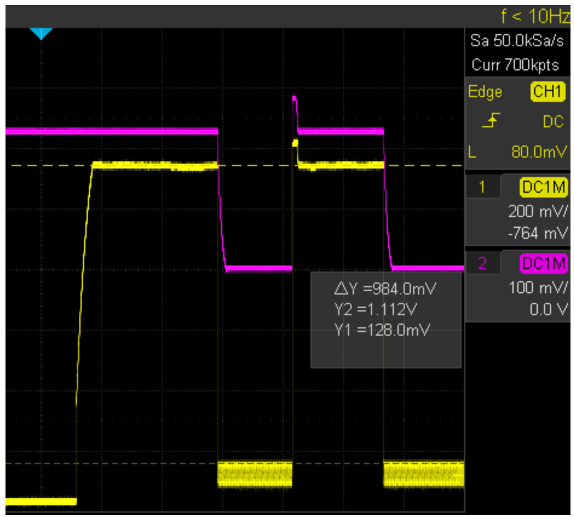


Figure 30: Close-Range Test (12V Supply, SNR)

- Long-Range Test: As mentioned in the test setup, nearly the same test was repeated as the close-range test, but with a distance of 6 feet 7 inches between the laser diode and the phototransistor. The lighting in the room was left at a normal light level, and the phototransistor was supplied at a DC voltage of 12V, and its output was measured across the resistor by the oscilloscope. The oscilloscope could only reach the phototransistor, so the laser diode input was not measured. Figure 31 below shows the phototransistor output in response to the pulsing of the laser diode. There was some noise in the signal (measured 160mV peak-to-peak), as was expected when the distance increased, but the signal-to-noise measurement when the laser diode was pulsed was still very significant, measuring 712mV. Therefore, we are fairly confident that our system could pick up the laser diode signal from this range and farther. One observation made which can be seen in Figure 32 at approximately 37 feet the laser diode light pattern forms an ellipse at a 45° angle across a 2.5 inch light pattern. We hypothesize that extending past this range of 6 feet 7 inches will decrease the systems SNR linearly and will reduce successful detection of the laser diode. Three theories have been developed to overcome this range limitation, should it prove to be an issue during further testing beyond Senior Design 1: Theory #1 is the replacement of the cheaper laser diode with a more expensive option with a comparable 5mW rating such as the 38-1007-ND mentioned in our pool of approved laser diodes. Theory #2 involves the replacement of the phototransistor with a model with a higher peak sensitivity wavelength gain such as the ALS-

PT204 from our pool of potential laser diodes. We expect that, should it be needed, we could increase our SNR using these two proposed methods. Additionally, Theory #3 involves developing a summing junction and utilizing multiple phototransistors in the target accumulating the total detected light at the target. Any losses from the unusual light pattern of the laser diode will be compensated for. Theory #1 and #2 can be easily adapted if necessary because both potential part replacements have already been ordered and are in reserve for testing. Theory #3 would involve a design change and is therefore the less preferable option, but still doable if needed.

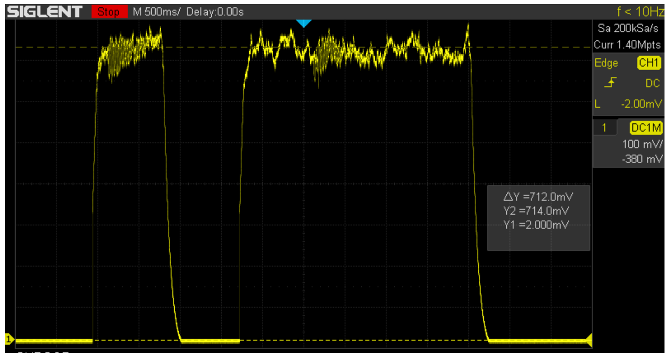


Figure 31: Long-Range Test (12V Supply, SNR)

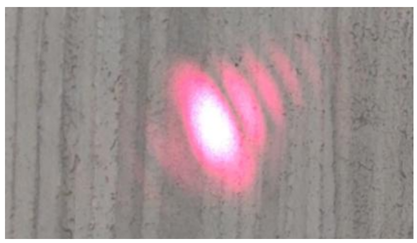


Figure 32: Laser Diode Light Pattern

5.1.2 Basic System Proof of Concept Test

Objectives/Motivations for Test: In order to effectively evaluate various components for final selection, the viability of certain hardware and software features, and the overall system design of the project, we devised a somewhat stripped-down test rig. While simpler than the final product, the test rig contains all the hardware necessary to not only evaluate a selection of parts for our use but also to plan and prepare for our later iterations and expansions of the feature-set to bring the device to our final, desired requirements.

Figure 33 is a rough block diagram showcasing the plan for our preliminary hardware evaluation platform.

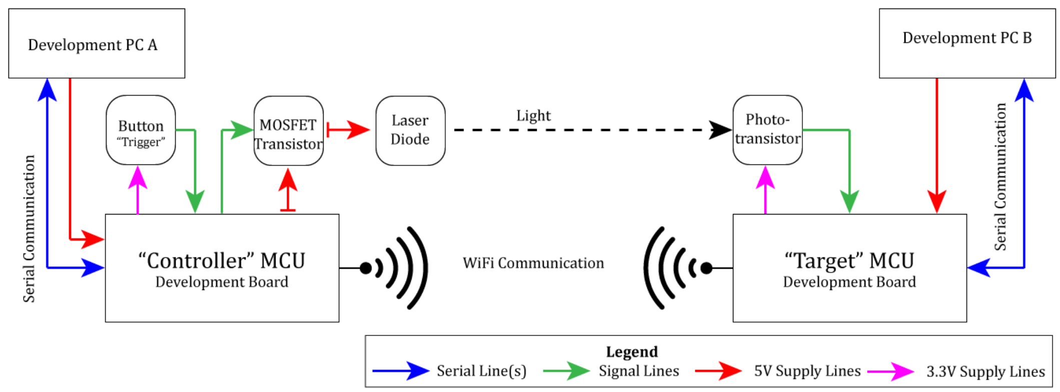


Figure 33: Hardware Proof of Concept Prototype Block Diagram

Materials and Supplies Needed for Test: To perform this test, we needed two development microcontroller boards, two development computers to write code and monitor serial output, a phototransistor, a MOSFET transistor, a laser diode, and a momentary button, alongside set of resistors and miscellaneous jumper cables in order to connect the various components on a large breadboard.

Testing Environment: This test was conducted upon one large breadboard, with each microcontroller electrically isolated from the other. The two development computers were also unplugged, running off of battery power, as to not connect the microcontrollers to a common ground or power supply. The test took place in a large engineering workshop, with strong fluorescent overhead lighting.

Test Setup: Our test environment was composed of two separate microcontroller development boards, one acting in place of the "Controller" device and one in place of the "Target" device. Each controller was connected via a serial connection to a development computer which tracks and logs information from their paired device. The devices themselves were interconnected only through a Wi-Fi-based connection, as well as an optical "connection" that was formed between the laser diode and the phototransistor.

The "Trigger" button was a standard momentary push-button switch, powered off of the "Controller" development board's power supply with a pull-up resistor on its data line, which feeds into a digital pin on the development board. It is set up to be active-low.

The Laser Diode was powered off of the 5V DC supply provided by the development PC through the microcontroller's development board. However, the laser diode was controlled by a 3V3 GPIO line on the microcontroller. In order to facilitate this, we utilized a MOSFET transistor to control the flow of the 5V line via the 3V3 line. This allowed us to turn on and off the laser diode with the microcontroller.

The phototransistor was powered by the target microcontroller's power supply, with its output current being directed through a resistor to ground. The microcontroller read the voltage across the resistor to measure the intensity of light the phototransistor is detecting via a digital pin, which triggers a hardware interrupt on the microcontroller.

The serial communication between each controller and the development computer is done via standard micro-USB cables.

Wireless communication is done through the onboard antenna of each microcontroller.

Test Results: The test revealed that our system worked, though it revealed some notable tweaks that we had to make during the testing process. Originally, the laser diode was being powered directly off of the control line from the microcontroller, only operating at the 3V3 level. Unfortunately, the luminosity output of the laser was far too low for the phototransistor to detect at that level, resulting in the adaptation of the system to include the MOSFET transistor to control it.

In software, the test revealed to us how we could use the TaskScheduler software library in order to properly schedule tasks to run after interrupts, and how we could use that in conjunction with traditional embedded programming techniques to debounce input signals – such as the trigger. We also were able to record some simple timing information – the amount of time it took for a message to be sent from one microcontroller to the other, and then parsed. The PainlessMesh library we used maintains a tight time synchronization between the nodes on the network, allowing us to use these system timestamps with notably high precision and reliability. Our measurements led to an understanding that these messages took on average about a tenth of a second to be prepared by the sender, sent wirelessly, and parsed on the recipient. The worst-case measured time was about two tenths of a second, and the best-case was about five hundredths of a second. As we expected, wireless communication is prone to wildly varying latency, so it shouldn't be relied upon for precise timing of system events. This will be taken into account when designing the game system in the coming months.

5.1.3 Audio Test

Objectives/Motivations for Test: Audio equipment can be fickle – ensuring that no compatibility issues exist between our choice of speakers and audio amplifier breakout boards is an important thing to verify early on. In addition, we wanted to ensure that we could play back audio samples with a high enough audio fidelity that a user can easily understand the sound effect being played when being held either in hand or at a short distance.

Materials and Supplies Needed for Test: Since each device involves the same audio setup – a single I2S chip, our standard ESP32 microcontroller, and a single speaker – we recreated the entire audio subsystem using those three major components and treated those results as representative of the behavior we'd see across our final implementation. In addition, we needed a computer to compile and flash the test software onto the microcontroller, various patch cables to wire up the connections, and a breadboard to hold our circuit.

On the software side, we needed to use the ESP8266Audio library in order to manage audio playback on our microcontroller. In spite of the name, the ESP8266Audio library is compatible with many microcontrollers that have similar processors – including our chosen ESP32. We used one of their provided example files (PlayAACFromPROGMEM) in order to ensure functionality with known-working code – and its included audio file.

Testing Environment: This test was conducted in a warehouse in an engineering lab on a workbench. This environment was not exceedingly conducive to high-quality audio playback, nor was it equipped to measure any quantifiable audio levels, as it was in a space with notable background noise due to other folks operating in the lab at the same time.

Test Setup: The setup for this test involved pinning half of the headers of a lone ESP32 development board into a breadboard, opposite a breakout board for the MAX98357A DAC. The 3.3V output pin of the ESP32 board was wired into the VDD pin of the breakout board, and the grounds were connected. BCLK and LRCLK, on the breakout board, were wired to GPIO pins 26 and 25, respectively, while GPIO pin 22 was connected to the DIN pin on the breakout board.

With the two boards interconnected, connecting the speakers to the audio breakout board proved to be the most finnicky part of the process. This was not due to any electrical or part fault, but due to the breakout board's non-standard vias that were provided as surfaces for connecting speakers. We ended up establishing a solid connection by using a set of digital logic probe wires which then were jumped into the standard audio connector the speakers were manufactured with.

Playback was initiated on startup of the microcontroller. Each time we pressed the "boot" or "reset" buttons on the development board, the audio test would be triggered after a short delay.

Test Results: The audio was played back reliably across all tests. Standing next to the device resulted in an audio listening experience that was clean enough to make out speech from audio samples and standing at a distance of at least 15 feet lent itself to and audio listening experience where a listener could discern between different audio samples being played. Further distances were not tested due to limited space and the test taking place in a shared environment.

5.1.4 Controller OLED Display Test

Objectives/Motivations for Test: Being the component responsible for displaying almost all the user input for the system, the OLED display is one of the most user-facing components of the controller. As such, it is important to ensure that the OLED can clearly display our intended designs for text, graphics, and animations, such that a user would be able to comfortably interact with our system.

Materials and Supplies Needed for Test: Our design involves a standard ESP32 microcontroller, which can be directly connected to control our Adafruit Monochrome 1.3" 128x64 OLED Graphic Display. As with our other subsystems, we created our setup by connecting the corresponding pins of our OLED to the corresponding pins on our microcontroller using a breadboard and various patch cables. Once the hardware was connected, we simply needed a computer to build and upload our test software onto our microcontroller.

For our software, we used Adafruit's OLED example file (ssd1306_128x64_i2c) to test our OLED output using the I2C protocol.

Testing Environment: This test was conducted in a warehouse in an engineering lab on a workbench. This environment was conducive to viewing our OLED feedback and determining its clarity and ability to smoothly transition between screens.

Test Setup: The setup for this test involved pinning the ESP32 board onto a breadboard. The OLED was pinned on the opposite side of the breadboard. To give the OLED the power it needed to run, the GND pin of the microcontroller was connected to the GND pin of the OLED and the 3.3V pin of the microcontroller was connected the Vin pin of the OLED. To setup the data connections, the D21 pin of the microcontroller, responsible for acting as the SDA pin for I2C, was connected to the Data pin of the OLED. Finally, the D22 pin, responsible for acting as the SCL pin for I2C, was connected to the CLK pin of the OLED.

After completing the required connections, we uploaded our test code to our microcontroller. The test code was responsible for creating a series of screens involving complicated graphics, animations, and text strings. As our design is limited to simple text, scrolling animations, and graphics, a successful display of this test would indicate that our OLED could serve its required purpose in our project.

Test Results: The OLED displayed the graphics with a reliable brightness and clarity. Looking at the device from a distance that a user would reasonably hold the controller, we could clearly make out each graphic, word, and animation that was displayed. We did not have any issues with a lagging or blurry display. Figure 34 shows an example of the output we received on our OLED display.

Figure 34: OLED Test Display

5.1.5 More Complete System Proof of Concept Test

This test was our next highest priority test and was performed very early in Senior Design 2 if. It was a repetition of the previous proof of concept test, but it also incorporated the other major hardware items required for the overall system. Once again, the basic principles were tested, involving the laser diode, the phototransistor, and two microcontrollers: one representing the "gun" controller, and one representing the targets. The signal was once again sent by the controller's MCU to pulse the laser diode, and the other MCU once again picked up the signal from the target "hit," and both signals were once again be displayed by the output of the software.

This time, however, the system also incorporated the "gun" controller's OLED screen and speaker, and the LED array display and speaker for the target. The screen on the controller provided a preliminary menu display that interacted with the menu buttons and with the system as a whole whenever the laser was pulsed and then when the phototransistor picked up the laser, and the speaker on the controller output a sound effect every time the laser was pulsed, to simulate the sound of "firing" the laser "gun". The LED array provided some basic lighting that changed when the phototransistor was "hit" with the laser pulse, and its speakers also output a sound effect to simulate the sound of a target hit as well. This test incorporated every single crucial feature of this design successfully, and gave us a pretty good idea of how to proceed with our PCB design layouts.

5.2 "Gun" Controller Hardware Design

This section contains designs for the hardware of the "gun" controller, including schematics that were used to create our final PCB design in Senior Design 2. Each of the main hardware functions will be discussed separately in the different subsections of this section. Hardware functions discussed will include the circuits driving the laser diode, the OLED display, the speaker, and the trigger, as well as the battery management and power distribution for the entire controller, and any other circuits deemed necessary. These schematics are the product of multiple revisions and reviews which ensured that they were exactly as we wanted them. These designs were converted into the final PCB design file sent to manufacturing during Senior Design 2.

5.2.1 Trigger and Laser Diode

During Senior Design 1, our original plan was to drive the laser diode by way of a MOSFET. However, during testing in Senior Design 2, we determined that this method was overkill and overcomplicated and we decided to eliminate the MOSFET entirely and drive the laser diode directly from the ESP32. Figure 35 below shows the schematic for the laser diode and trigger systems.

Starting with the laser diode system on the left, and beginning at the top, the laser diode is connected to the LSR_DVR net in the schematic. As will be shown later in this section, that net corresponds to three separate pins on the ESP32, all capable of outputting 3.3V, but combined to produce enough current to drive the laser diode directly from the ESP32. Moving down the laser diode schematic, the connector (labeled "CN3-LSR") is a 1729128 is a Phoenix Contact, also known as a screw type fixed terminal block. The positive and negative wires for the laser diode are screwed into this terminal block.

Moving on to the trigger system on the right side of Figure 35, the trigger acts as an active-low button. The trigger itself is the C3AW-1A-8F manufactured by Omron Electronics Inc. and sold by Digi-Key. It is a premade trigger-switch, already assembled with the trigger mechanism and shape in place. It was extremely convenient to use, requiring only to be installed into the controller and its wires to be connected to a screw terminal on the PCB. On the trigger schematic, the trigger (connected to the same model of screw terminal as the laser diode) acts like an open circuit until it is pulled. The constant 3.3V signal is applied to the 1k resistor R12, which is connected to the trigger itself. The "Trigger" net on the schematic is connected on this same node. The other end of this net is connected to the ESP32, which reads if it is a high or low signal. When the trigger is pulled, it acts like a closed circuit and the "Trigger" net is pulled to ground. The ESP32 reads this and knows that the trigger has been pulled, then acts accordingly based on how it has been programmed in software. These systems, while relatively simple in the end, formed the backbone of the entire controller design.

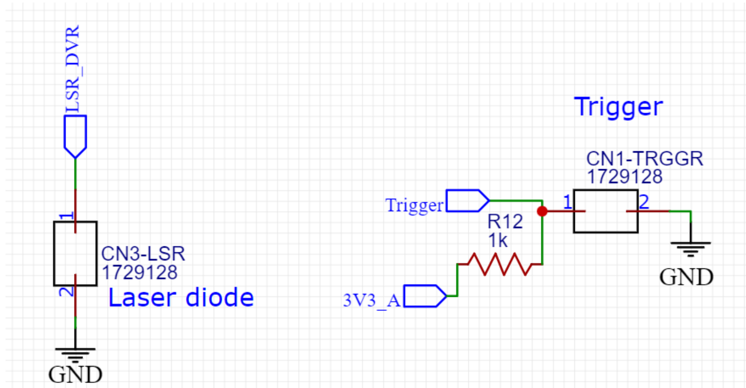


Figure 35: Laser Diode and Trigger Schematic

5.2.2 Controller Voltage Regulators

After performing a preliminary power consumption analysis of the controller, we were able to characterize the power limitations that the voltage regulators must exceed to insure stability of the system. We know the laser "gun" requires a 3.3V all critical components. The originally selected IC BD9227F-E2 was referenced from the typical application circuit provided on the datasheet with part replacement/tuning to meet or exceed the voltage-current limits of individual discrete components. As a switching voltage regulator, it requires a number of passive components to operate at the intended output voltage. The BD9227F-E2 was used in all early prototypes and the first two revisions of the PCB, with the idea being that all needed voltage levels could be accomplished using the exact same regulator. It was easily adjustable by just changing out a single resistor. However, we were unable to get all voltage regulators performing reliably (possibly due to PCB layout issues, the complicatedness of the circuit, or some other unknown factor), so we added the option for linear regulators in our third and final PCB revision to ensure reliability.

For the final version of the controller, all BD9227F-E2 regulators were placed in parallel with fixed linear voltage regulators with the same output voltage. While this took up more space on the PCB layout, it added a layer of redundancy that ensured that we would be able to get the power working. Because we had never tested the linear regulators before and had to order them at the same time as the final PCB revisions, we thought it would be wisest to leave the BD9227F-E2 regulators in place just in case the linear regulators did not work.

As mentioned previously in parts selection, the linear regulators we used were quickly and easily found on Mouser. For the 3.3V regulator, we chose the UA78M33CDCYR manufactured by Texas Instruments. We ended up having two 3.3V regulators on the board, one to power only the ESP32, and the other to power the rest of the components. While not strictly necessary, we were worried that the output power of only one regulator may not be enough, so we included two just in case to avoid overloading any single regulator. The schematics for both 3.3V voltage regulator circuits (nearly identical except for the "A" and "B" designations) are shown in Figure 36 and Figure 37 below. The schematics are relatively simple and were developed with the help of the regulator's datasheet, which gave a "best practice" example for creating a circuit with this regulator. Both have input and output stabilizing capacitors, and both take the battery voltage as input and put out a steady, fixed 3.3V output, with up to 500mA of current.

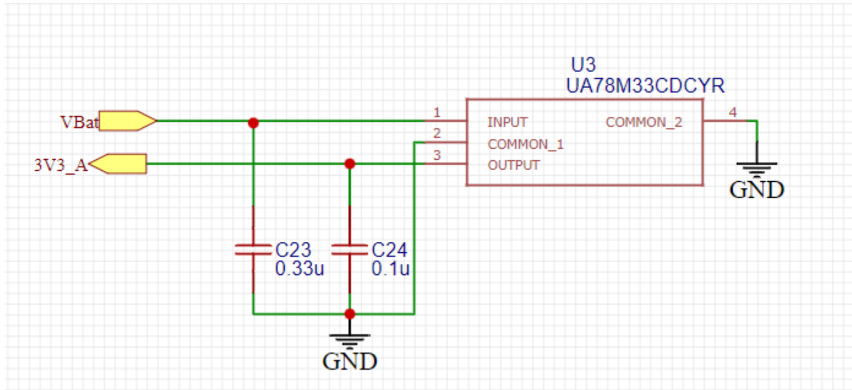


Figure 36: 3.3V (A) Linear Voltage Regulator

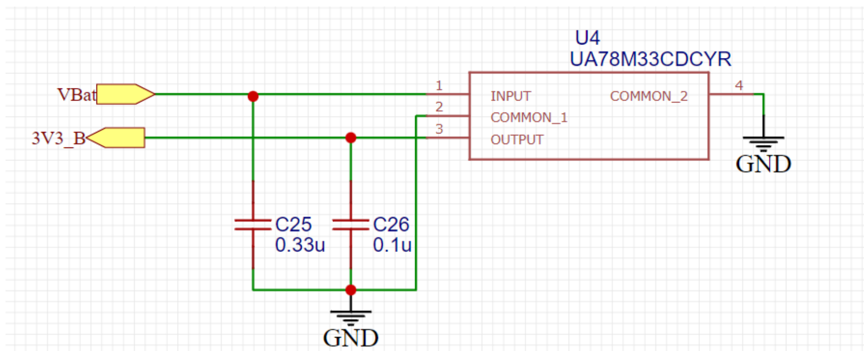


Figure 37: 3.3V (B) Linear Voltage Regulator

.

5.2.3 Audio System

During our preliminary speaker testing, we used an audio breakout board from SparkFun (specifically, the SparkFun I2S Audio Breakout - MAX98357A) as the “middleman” between our microcontroller and our speakers. This board worked perfectly for our purposes during testing, but our plan was to take only the necessary components of the board and integrate them into our PCB for the final design. This would save space inside of the system enclosures, make our PCB design more substantial to meet the project requirements, and allow us to choose only the components we needed.

The schematic part of this design was made easier by the fact that SparkFun provided a schematic of the entire breakout board as an Eagle file, which could be imported into EasyEDA as needed. One feature included in the original board was the ability to select different left and right audio channels, meaning you could drive two separate speakers if you wanted to. This feature was made possible by two different jumpers on the board, one that defaulted the system to one speaker that would need to be cut to select two, and one that would need to be soldered if you were selecting a specific (left or right) channel to use. We planned for our “gun” controller to only contain one speaker, so neither of these jumpers were deemed necessary, and they were removed from the original schematic. Additionally, the original schematic had a place for pin header connections, which was removed because all connections from the amplifier circuit to the microcontroller are contained within our designed PCB.

The final schematic for the audio amplifier circuit and the connector that goes to the speaker is shown in Figure 38 below. The main box is the MAX98357A integrated circuit chip, our chosen audio amplifier from the Parts Selection section of this document. It has the expected connections of input voltage (VDD) and ground (GND), which will be coming from the 3.3V bus and ground, respectively. The !SD_MODE connection is the shutdown and channel select pin on the amplifier. According to the datasheet of the MAX98357A, it can be pulled low to place the device in shutdown, high to select the left channel, pullup through a small resistor to select right channel, or pullup through a large resistor to combine the left and right channels into a single channel to drive a single speaker. Since we are driving a single speaker, we opted to connect this pin as pullup through a large resistor (1 MΩ), as shown in the schematic. It is also connected to the 3.3V bus. The GAIN connection sets the gain (in decibels) of the amplifier depending on how it is connected. However, the default gain is set to +9dB with no connection made to this pin, which provided sufficient audio through the speakers in testing, so this pin will be left floating, as is indicated by the "X" on the schematic. The remaining pins connect to pins on the ESP32. Both clock inputs (LRCLK and BCLK) require a connection to a DAC pin on the microcontroller, which correspond to the IO25 and IO26 pins on the ESP32, respectively. The serial data input connection (DIN) is connected to the IO19 pin on the ESP32. Lastly, the output of the amplifier (OUT+ and OUT-) is connected to the speaker by a screw terminal block.

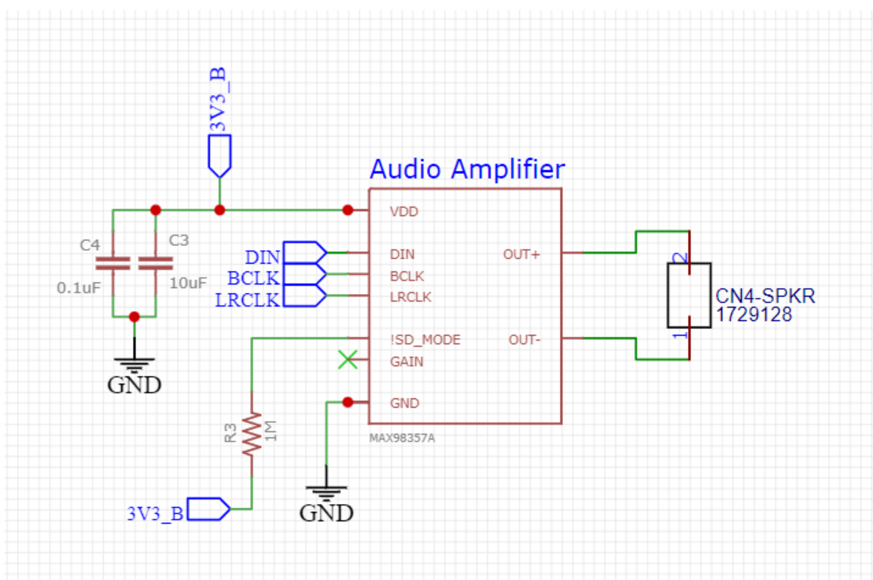


Figure 38: Audio Amplifier and Speaker Connection Schematic

5.2.4 OLED Display

The OLED display that we used in the "gun" controller (the Monochrome 1.3" 128x64 OLED graphic display from Adafruit) comes pre-mounted on its own small breakout board. After assessing the contents of this board, we felt it was best to continue using the display with it rather than attempting to create the circuits to drive the OLED display ourselves. It saved us time, money, and complications, and creating a circuit of that scale was well beyond the scope and constraints of this project, as well as the abilities of our group members. Therefore, the included breakout board remained with the screen. Adafruit included two STEMMA QT / Qwiic connectors on the board for "plug-and-play" purposes, and we made use of those. One of these connectors was used to connect the display to the controller's PCB. This was done by placing a compatible JST SH 4-pin Right Angle Connector (Adafruit product #4208) on our designed controller PCB and linking both connectors with a STEMMA QT / Qwiic JST SH 4-pin Cable (Adafruit product # 4210).

The schematic for the Qwiic connector that was placed on the "gun" controller's PCB is shown in Figure 39 below. The pinout of a Qwiic connector is as follows: Pin 1 is ground (black wire on cable), Pin 2 is 3.3VDC power (red wire on cable), Pin 3 is the I2C SDA data (blue wire on cable), and Pin 4 is the I2C SCL clock (yellow wire on cable). Therefore, as seen in the schematic, Pins 1 and 2 are connected to the ground and to 3.3V, respectively. Pins 3 and 4 are connected to the IO21 and IO22 pins of the ESP32 microcontroller, respectively, to receive the I2C data and clock signals.

The entire OLED display connection system shown in the schematic in Figure 39 was tested on a breadboard with the ESP32 microcontroller during the OLED display testing (by way of pins on the OLED display's breakout board) and performed satisfactorily. Therefore, it was expected that this system would perform just as well when integrated into our final designed PCB with the added use of the Qwiic cables and connectors, which were purchased from Adafruit as well, since the ones sold by Adafruit are guaranteed to be compatible with the OLED, which is also from Adafruit. The PCB footprint of the Qwiic JST connector is given on the Adafruit website and was used to check the PCB footprint for this connector. This system worked exactly as expected and did not need to undergo any revisions during the entire PCB design process.

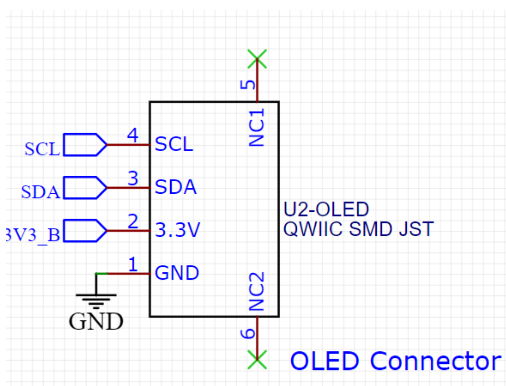


Figure 39: OLED Display Connector Schematic

5.2.5 Microcontroller and Programming Pins/Buttons

The most important aspects of the controller's design are all of the connections to the microcontroller, and the programming buttons and pins. Without the correct microcontroller connections, none of the system would be able to operate. Without the programming pins and buttons, programming the system with all necessary software would be impossible. We took great care to ensure that these schematics were connected properly and were pleased with the results. The connections to the ESP32 are shown in Figure 40 below.

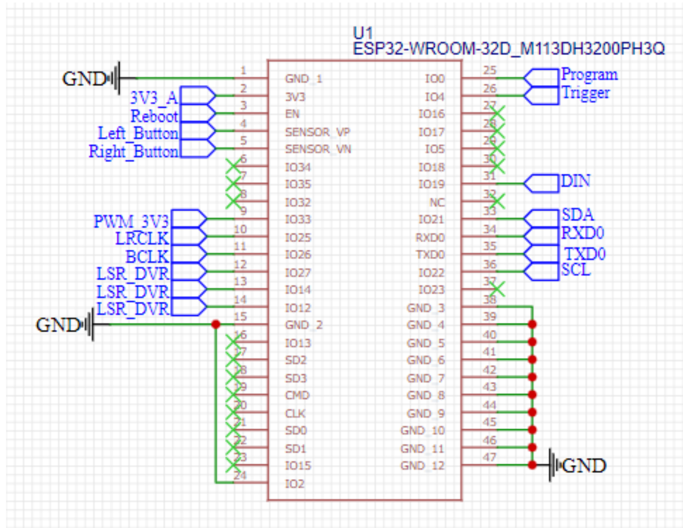


Figure 40: Controller ESP32 Connections Schematic

The programming pins and buttons connections are shown in Figure 41 and Figure 42 below. We used a four-pin header for four-pin UART programming, with connections for the 3.3V supply to power the microcontroller during programming, the TXD0 and RXD0 pins, and the connection to ground. The two buttons used were a "Program" button connected to IO0 on the ESP32, and a "Reboot" button connected to EN on the ESP32. Both buttons were set to active-low.

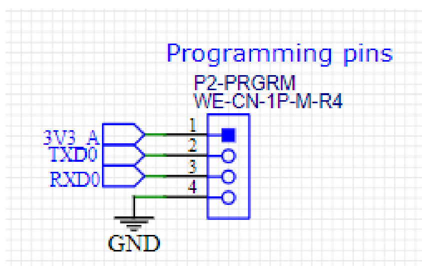


Figure 41: Controller Programming Pins Schematic

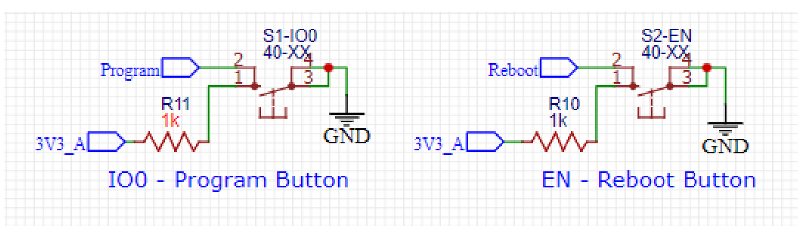


Figure 42: Controller Programming Buttons Schematic

5.2.6 Controller Daughter Boards

Due to the handgun-like shape of the controller, and the fact that the PCB was going to be placed in a permanent enclosure, we wanted to be able to place some components off of the board to make them more easily accessible. For that reason, we made two daughter PCBs: one for the controller's power on/off switch, and the other for the left and right menu buttons. These schematics and boards were finalized without need for revisions because they were so simple. Figure 43 below shows the schematics for both, from their connections to the main board, to the individual daughter board schematics, and lastly to the completed and assembled daughter boards themselves. It is also worth mentioning that the on/off switch connection for the main board included a diode for reverse-polarity protection and a fuse for overcurrent protection.

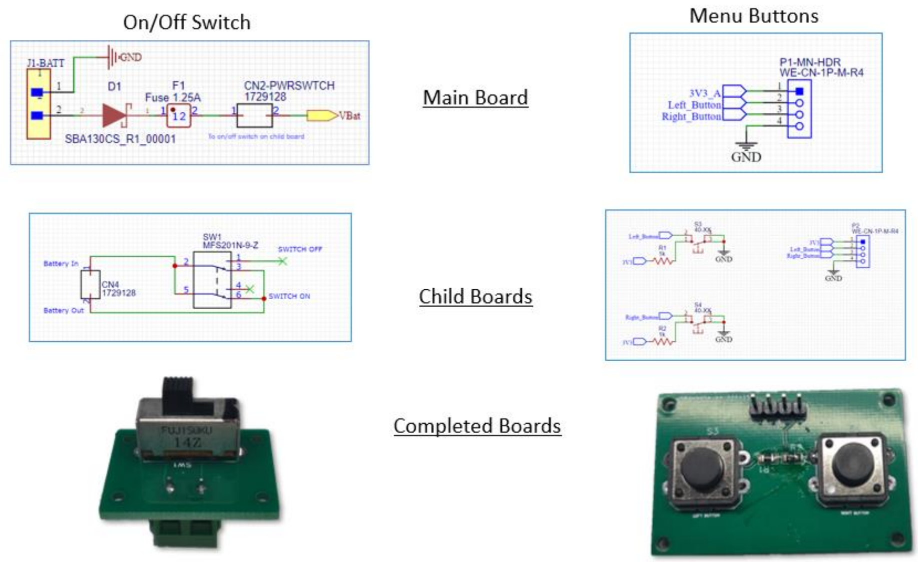


Figure 43: Controller Daughter Boards

5.2.7 Controller PCB Design

Due to the group's lack of experience with PCB design, a few PCB footprint mishaps, and a few component changes, our controller PCB went through three total revisions during the course of Senior Design 2. The layout of the final revision (as viewed in EasyEDA) is shown in Figure 44 below, and the final assembled PCB that was installed in the controller is shown in Figure 45 below.

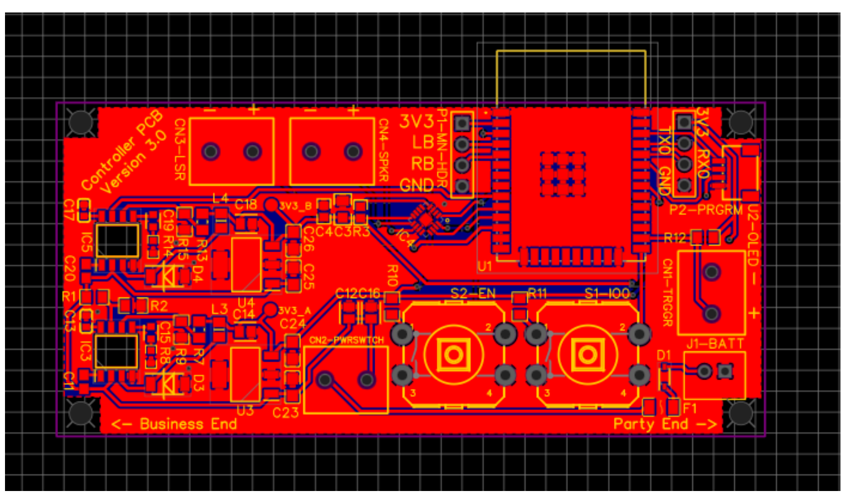


Figure 44: Controller PCB Layout in EasyEDA

Figure 45: Assembled Final Controller PCB

5.3 Target Hardware Design

This section contains our designs for the hardware of the targets, including schematics that were used to create our final PCB design during Senior Design 2. Each of the main hardware functions will be discussed separately in the different subsections of this section. Hardware functions discussed will include the circuits driving the phototransistor and reading its output, the LED array, and the speaker, as well as the battery management and power distribution for the entire controller, and any other circuits deemed necessary. These schematics are the product of multiple revisions and reviews which ensured that they were exactly as we wanted them. These designs were converted into the final PCB design file sent to manufacturing during Senior Design 2..

5.3.1 Phototransistors

The original plan for the target design included only one phototransistor to sense the laser beam. However, due to the small size of the lamp of the ALS-PT204-6C/L177 phototransistor, we opted to use five phototransistors arranged in a circular shape on the PCB and placed in parallel with one another on the target's schematics. This produced the exact same light-sensing affect as a single phototransistor but provides the user a larger area to aim the laser at, making the game easier and less frustrating to play. The schematic for the five phototransistors is shown in Figure 46 below. All five phototransistors are connected to the 5V line, which keeps them powered on. Upon excitation from the laser, any one (or multiple) of the phototransistors will create a voltage across R10 on the other node. When this voltage is created, it is sensed by the ESP32 by way of the pin connected to the "PtSense" net, and the software will register that voltage change as a target "hit."

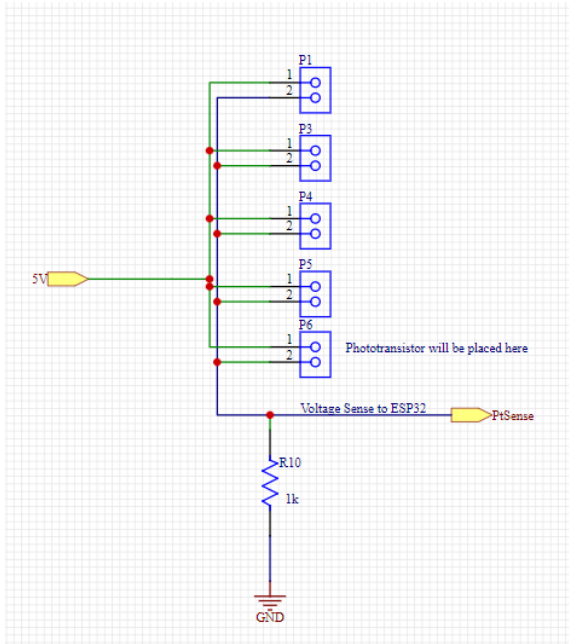


Figure 46: Phototransistors Schematic

5.3.2 Target Voltage Regulators

The same design principles applied to section 5.2.2 were applied to the target system. Just like the controller, it has two 3.3V linear voltage regulators (they are entirely identical to those on the controller, so images of them were omitted for this section). The only difference is that there is an additional 5V regulator, the MC7805CDTRKG manufactured by onsemi. The 5V rail is used by the target's phototransistor drive circuit and its LED strip. Figure 47 below showcases the schematic for the 5V regulator. Just like for the 3.3V regulators, the schematic is relatively simple and was developed with the help of the regulator's datasheet, which gave a "best practice" example for creating a circuit with this regulator. It has input and output stabilizing capacitors, and it takes the battery voltage as input and puts out a steady, fixed 5V output, with up to 1A of current. The target's LED array can draw a significant amount of power (1A maximum on full bright white), but this level of brightness is never achieved during normal target use, so only a single voltage regulator is required for the phototransistors and LEDs. No doubling required.

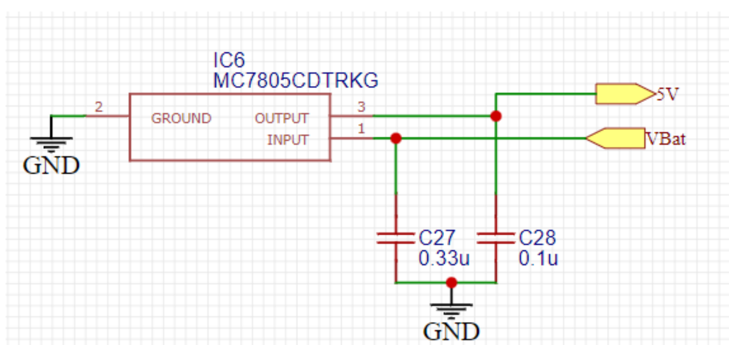


Figure 47: 5V Linear Voltage Regulator

5.3.3 Audio System

The audio system for each of the targets will be identical to that of the “gun” controller because the same speaker was chosen for each, and each device will have only one speaker. The same connections were made to the microcontroller, which is again the same between both devices (the ESP32). Therefore, when creating the schematic for the target’s PCB, the same schematic for the audio amplifier and speaker connection was copied over. So that we are not including the exact same schematic twice in this paper, refer back to Figure 38 to see the schematic for the audio system for the targets.

5.3.4 LED Array

The LED array is connected to the target PCB by way of a three-position screw terminal. One position is connected to the 5V rail to power the LEDs. The second position is connected by way of R4 to the LED_DATA net, which connects to a pin on the ESP32 which outputs the lighting data to the entire strip and its individually addressable lights. The third position connects to ground. The schematic for this connection is shown in Figure 48 below.

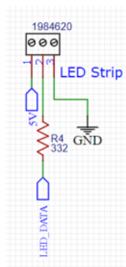


Figure 48: Led Array Connection

5.3.5 Power Switch and Pairing Button

Unlike the controller, the target’s power switch is mounted directly on the board. However, like the controller, it includes a diode for reverse-polarity protection and a fuse for overcurrent protection. It is located on the opposite side of the PCB from the other components for easy access while the PCB is installed in the controller. The Pairing button, used to pair each target to the controller, is also located on the back of the PCB. Figure 49 below shows the schematics for the power switch and pairing button.

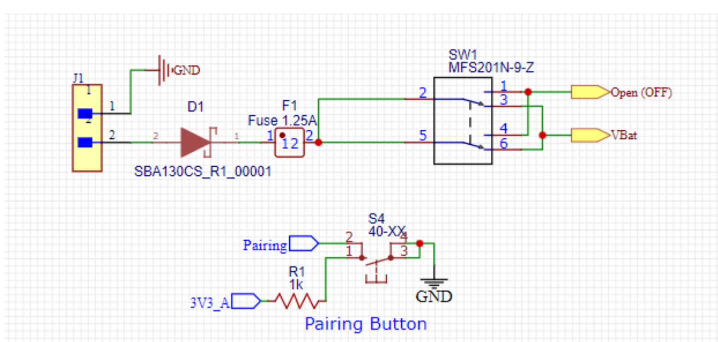


Figure 49: Power Switch and Pairing Button Schematics

5.3.6 Microcontroller and Programming Pins/Buttons

Following the same principles as for the controller, the target's microcontroller connections and programming pins and buttons were carefully designed to ensure that the system would be programmable, and all parts would work properly. The connections to the ESP32 are shown in Figure 50 below, and the programming pins and buttons are shown in Figure 51 below.

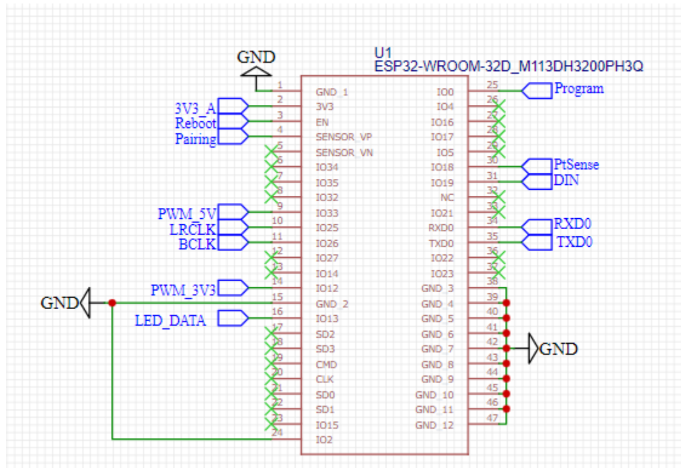


Figure 50: Target ESP32 Connections

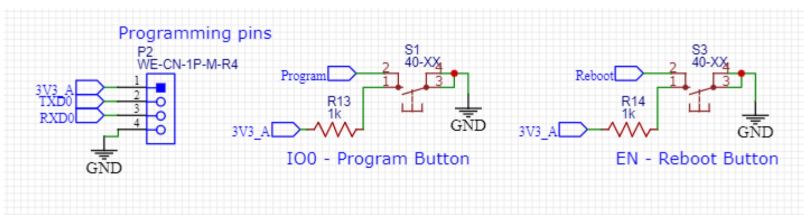


Figure 51: Target Programming Pins and Buttons

5.3.7 Target PCB Design

Due to the group's lack of experience with PCB design, a few PCB footprint mishaps, and a few component changes, our target PCB went through three total revisions during the course of Senior Design 2. The layout of the final revision (as viewed in EasyEDA) is shown in Figure 52 below, and the final assembled PCB that was installed in all targets is shown in Figure 53 below.

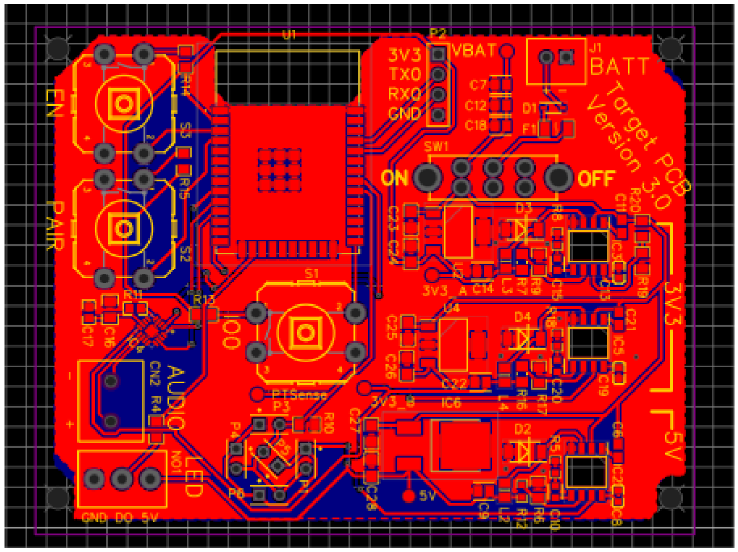


Figure 52: Target PCB Layout in EasyEDA

Figure 53: Assembled Final Target PCB

5.4 Software Design

5.4.1 Overall Embedded Software Architecture

As this system was designed to operate in a game-like nature, it was best to look towards other implementation of game systems (both virtual and practical) to see how these other forms managed game state, handled user input and inter-device communication, and even structured the code that ran and managed the internal systems of these devices.

In traditional, real-time video game systems, the application's software tends to run on indefinitely repeating loops, sometimes fixed, sometimes running as fast as the host processor can handle. A loop responsible for handling the rendering of visuals to the screen, for instance, will run as fast as is possible as to reduce apparent visual latency and more convincingly create the illusion of motion in the form of consistent, smooth changes on a user's display.

Some systems like physics and animation systems run on fixed-period loops, wherein the stability of the loop's execution time can affect factors like the predictability and deterministic quality of a physics simulation, stability and artistic quality of effect and character animations, or even the playback of audio, which requires a steady clock in order to convincingly recreate audio waveforms.

Our game system relied on neither simulation of physics or particles, nor a real-time rendering of graphics to a display. While we did include both aural and visual feedback to the player over the course of the use of our game system, all of these feedback systems could be pre-calculated, or "baked," so that our game system (and software) only needed to play back these pieces of media when necessary.

The question then became "How do we react to user actions?" If a player were to pull the trigger on the controller, how should the software check that this has occurred, and how should the software then ignite the emitter within a short enough window of delay as to not break the flow of play? How should a target detect a valid hit upon its sensor, and how should it then manage the reaction it must perform and present to the user?

The naïve approach to designing an embedded system's operating process is to rely entirely on an update loop of a fixed period. In a system like this, the microcontroller embedded in any given device would check each relevant source of input (time intervals, buttons, sensors, wireless modules, etc.) sequentially, then operate upon the state of each input.

In the solution described above, one notable problem that arises is the sequential nature of checking inputs. As the number of inputs increases, so too does the amount of time each loop takes. In addition, operating on the result of these inputs takes time, and since the entire behavior system occurs in a single loop, many operations could result in the loop taking longer than expected, even delaying the next iteration of the loop. In addition, this system results in a high-power draw as the CPU is never resting and must loop as frequently as possible to ensure no inputs are missed. As our system was entirely hand-held and battery-powered, it was important to find a solution that was a mix of both high-power efficiency and low response time.

A more advanced, but still flawed solution is to operate entirely on interrupts for input events. If the user pulls a trigger, wake the microcontroller's CPU from its low-power rest and perform whatever action must be accomplished immediately. After execution, the CPU falls back to sleep. This results in an incredible power-savings, as the CPU spends most of its time in a low-power mode. However, interrupt-based programming can be fraught with programming complexity.

Interrupts, especially on systems like the microcontrollers we targeted, are not supposed to perform long, arduous operations with complicated memory operations, allocations, or more. In addition, interrupts can sometimes be preempted by other interrupts, leading to issues with memory management, I/O operations, and more. In addition, just even developing interrupts properly with so many possible sources of interruption is complicated and leaves much to be desired.

For our solution to this problem, we looked at the "Event-Based" programming paradigm. Historically used most frequently in graphical user interface (GUI) applications, a software solution utilizing this paradigm reacts to incoming events and performs actions based upon the event type, event parameters, and any number of other factors. Events can be "triggered", or, placed into a queue to be processed, by hardware interrupts, timers, networking messages, or even other actions, allowing for easily extendable functionality in a system. This method helps combine the best of the previously described methods while covering some of their disadvantages.

An event-based system comes with other benefits that were especially compatible with our specific use-case. As events are non-blocking, meaning that they don't stop the flow of the overall program, they are innately asynchronous, meaning that the action of one event doesn't preempt the enqueueing of another event. This was critical for a system such as ours, where the timing, reliability, and overall consistency of the user-experience was critical for the success of the project. This also was useful and convenient for our system's wireless communication; treating incoming messages as just another event to process resulted in less software complexity.

If, for example, a user pulled the trigger of the controller at the same instant as a wireless message was received from a target, then in an exclusively interrupt-based system (or in a system that simply operated off of a fixed loop polling sensor data) that trigger pull might be missed entirely, resulting in the user's frustration as their input is – by their experience – simply rejected by the system.

A system like this could also be more straightforward to develop and extend – simply adding more handlers for more events could result in additional functionality being implemented with only care being taken to handle things in an appropriate order.

5.4.1.1 Audio Playback

Audio Playback was a critical component of the controller system – playing back a satisfying noise when the trigger was pulled gave the user feedback that a laser was fired. Our system also offered support for integrating audio into the target. As such, audio playback was a shared module of code that worked on both devices.

Audio playback was performed on the ESP32 via the pair of Digital-Analog Conversion (DAC) pins on the microcontroller, using the I2S protocol. This I2S feed was wired into the amplifier integrated circuit chip in order to improve the quality and volume of our audio playback. We shifted data from our program memory into SRAM, which acted as an output buffer for our I2S feed, which dumped data onto the line at an appropriate speed. Our main loop of our audio playback module loaded data into this buffer as frequently as reasonable in order to ensure our audio playback was properly saturated. In order to facilitate this, the ESP8266Audio library was used. Despite its name, this library is not particular to that device but instead is compatible with a wide array of microcontrollers – including the ESP32, our chosen MCU. We set up the files we wanted to play – which were already stored in the slower, higher-volume storage on the MCU, then just signaled to the library as to which memory location to draw from, then provided it time to update the memory buffer. This ensured our playback happens quickly, easily, and reliably.

5.4.1.2 Wireless Communication

The second critical shared component of our system was wireless communication. To establish connections between the controller and the targets, we used the PainlessMesh library for the ESP32. This library helped create and manage the network of devices, making communication between any node possible. Each ESP32 was associated with a 32-bit address, or chipID, that was used by the library to identity devices for sending messages over Wi-Fi. The library provided various methods for receiving and sending these messages, which were useful for communicating the events that happened throughout the use of the system. It also provided methods for adding new connections and reconnecting devices, which could be used during pairing for adding more devices to extend the functionality of the game. There was a very high limit on the number of devices that could be connected in a network, which accommodated the comparatively smaller number of devices that will be paired during gameplay [63].

To assist in the communication aspect of this project, we created a message protocol that all messages sent on the wireless network abided by. Each message was a string in the following format:

<timestamp>;<tag>|<data>.

The timestamp was an unsigned long representing the time at which the message was sent, the tag was a string representing a description of what was inside the message, and the data was a string containing the message to be sent.

5.4.2 "Gun" Controller Software Design

The controller held the unique role of being the "Lead" device in this system. This most notably included the management of game state. Managing game state is something that is another interesting computing problem and was approached in our project via the use of a "Finite State Machine," a system which has its behavior defined via various discrete states, between which the system shifts according to user input and other environmental factors. The states of the controller are listed below in Table 17.

STATE	DESCRIPTION
INITIALIZING	The microcontroller communicates with and configures the various hardware components of the device, including establishing wireless communication, loading configuration variables, and preparing the device for use.
PAIRING	The microcontroller waits in the "pairing" mode, connecting and communicating with all other pairing devices.
READY	The microcontroller is ready to operate, maintaining any valid connections and waiting on user input or system events.
PLAY	A game is currently running. The microcontroller manages the various aspects of the game state, as well as serving as the gun's systems controller.
RESULTS	A game is complete. The controller displays or logs these results and waits on user input to move forward.

Table 17: Controller States Table

The states in Table 17 change according to certain specific events. The relationship between each state, the transitions between each, and the events that trigger these changes are shown below in Figure 54.

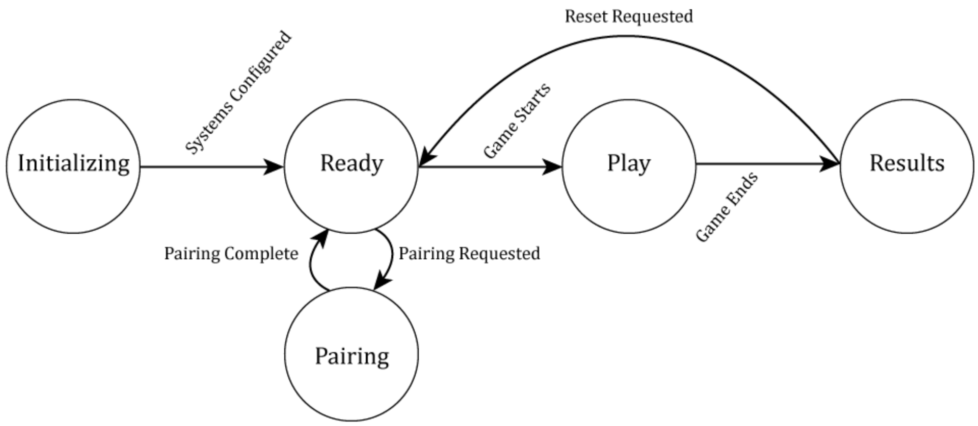


Figure 54: Controller Finite State Machine Diagram

Our system stays in the Initializing state until all peripheral systems are properly configured, without critical error, at which point we shift over to the Ready state.

While in the Ready state, the system offers some interface options to the user, allowing a user to configure an upcoming game session, start a game session (putting the controller in the Play state), or shift the controller into the Pairing state.

While in the Pairing state, the controller listens on an open network as the user shoots the targets they wish to pair. Once the user is satisfied with the pairing they've completed, the device returns to the Ready state.

When the user is in a game session, they're in the Play state. During this time, the controller processes user requests to fire the laser (trigger pulls) and handles the business logic of managing the overall game state – keeping track of hit events sent wirelessly from a target, signaling to targets that they need to ignite, and keeping track of overall timers. Once the game's time is up, or a loss condition is encountered, the controller is shifted into the Results state

While in the Results state, the controller displays information about the game session, such as the user's score, and offers the opportunity for the player to play again or to return to the Ready menu.

Below, Table 18 lists the various events that the controller can be expected to raise or handle.

Event	Channel	In/Out	Relevant States
INPUT_LEFT	Hardware	Incoming	All
INPUT_RIGHT	Hardware	Incoming	All
INPUT_TRIGGER	Hardware	Incoming	All
PAIR_REQUEST	Wireless	Incoming	Pairing
PAIR_REJECT	Wireless	Outgoing	Pairing
PAIR_ACCEPT	Wireless	Outgoing	Pairing
PAIR_COMPLETE	Wireless	Outgoing	Pairing
TIMER_ELAPSED	Software	Outgoing	Play
TARGET_HIT	Wireless	Incoming	Play
TARGET_IGNITE	Wireless	Outgoing	Play
TARGET_EXTINGUISH	Wireless	Outgoing	Play
GAME_START	Wireless	Outgoing	Ready, Play
GAME_END	Wireless	Outgoing	Play

Table 18: Controller Events

5.4.2.1 Screen Control

Onboard the controller is a unique element – a small, OLED screen to provide text and numeric feedback to the user. This display serves as a menu during the Pairing and Ready states and displays relevant gameplay information during the Play state such as remaining shots, round/point counter, and more. The display also serves to show the users their score in the Results state.

As this functionality is unique to the controller, the software to control it was only implemented as part of the controller's software. We used simple text graphics in order to convey information – which also helped save on program size and memory utilization.

The screen we chose to use on the controller made use of an I2C interface to communicate. The manufacturer provided two library packages for use with this device. One library, Adafruit_SSD1306, is a driver library for handling the lower-level logistics of I2C communication with the OLED screen [64]. The other library, Adafruit_GFX, is a higher-level library that provides access to graphical components like text, shapes, and animations [65].

In our software, we wrapped our own set of graphics instructions around the provided communication and graphics libraries to ease our development process and simplify code execution. A "screen manager" object managed the current state of the screen, handled the playback of animations, and responded to function calls from other parts of the software on the controller. In this way, we abstracted the screen's control so as to make it easy to use in software.

5.4.2.2 User Interface Design

Initializing

In the Initializing state, the user has no input options. This state is used to give the system time to configure its hardware and preparing the device for use. As such, there is no view associated with this state. When the controller is ready, the user will see the main menu options available in the Ready state.

Ready

After Initializing, the system will enter the Ready state. In the Ready state, the user will be provided with the screen in Figure 55 offering them main menu options.

Figure 55: User Interface for the Main Menu in the Ready State

The user can choose to begin a game, pair more targets to the system, change the system settings, or see information about the system. The information screen shows the user details about the last time the code was updated, the names of the creators of the system, and other version information. If the user chooses to select a game mode, they will see the screen shown in Figure 56 offering them the four game mode options. On this screen, the user can click the left or right buttons to scroll up or down, respectively. The user can also use the trigger to select their game mode, or return to the main menu using the Back option pictured in Figure 57.

Figure 56: User Interface for Game Selection in the Ready State

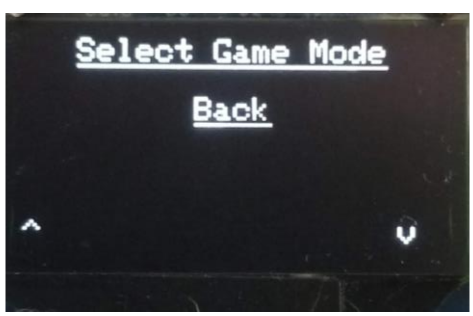


Figure 57: User Interface for the Main Menu - Back Option in the Ready State

If the user wishes to change their game settings, they will see the screen in Figure 58 offering them choices for how loud the audio should play.

Figure 58: User Interface for Settings in the Ready State

Pairing
In the Pairing state, the user connects their controller to the targets they wish to use in their game by shooting the targets. Users have the option to exit Pairing mode and return to the Ready state by clicking the left input button. This action saves the user's connections. Once targets are connected, the user is shown the number of targets they have connected. To delete all connections, the user will be able to click the right input. The view for the Pairing state is shown in Figure X below.

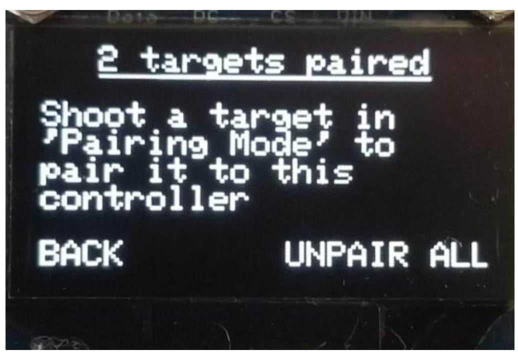


Figure 59: User Interface for the Pairing State

Play
In the Play state, the view the user will see on the controller will depend on the selected game mode. In all game modes, the user will be able to see the number of targets they have successfully hit. In Time Trial and Whack-A-Mole, the user will also be able to see their remaining time. In Horde, the user will be able to see how long they have survived. In One-Shot, the user will be able to see how many shots they have remaining. An example view for Time Trial in the Play State is shown in Figure 60.

Figure 60: User Interface for the Play State

In Play mode, the user may also decide to quit the game. To reach this option and go to the Results screen, the user can click the left input button next to the QUIT option on the screen.

Results

Once gameplay has ended, the user can view their game mode and final score. The user can click the left input button to return the controller to its Ready state. The results screen is shown in Figure 61.

Figure 61: User Interface for the Result State

5.4.2.3 Controller Event Control

The following is a discussion of which events are used to handle each state the controller may be in.

Initializing State

The controller does not receive or send any events in this state.

Ready State

When users first enter the Ready state, they see the main menu. To scroll up the menu, the user presses the left button, and the controller sends an INPUT_LEFT event. To scroll down the menu, the user presses the right button, and the controller sends an INPUT_RIGHT event. To select an option, the user presses the trigger, and the controller sends an INPUT_TRIGGER event.

Pairing State

In the Pairing state, an INPUT_LEFT event saves all connections and exits to the Ready state. An INPUT_RIGHT event deletes all saved connections. When the user shoots a target, the target will send a PAIR_REQUEST to the controller. When the controller receives that event, the controller sends either a PAIR_REJECT or PAIR_ACCEPT event to the target. Once the user has confirmed their connections, a PAIR_COMPLETE event is sent to each target to signal the end of Pairing mode.

Play State

The controller sends a GAME_START event to all targets when the user begins playing. From there, the events that occur depend on the game mode selected by the user.

<u>Time Trial</u>

When the game begins, the controller initializes a variable containing the number of targets that are connected and available for play. The controller sends a TARGET_IGNITE event to each target and sets a timer for one minute. Upon receiving a TARGET_HIT event, the controller sends a TARGET_EXTINGUISH to turn off the target that was hit. The number of available targets is decremented, and the score is incremented. Once the available target counter reaches zero, indicating all targets have been extinguished, the controller sends a TARGET_IGNITE event to turn on all targets and reinitializes the available target counter. At the conclusion of the game, triggered either by the controller's TIMER_ELAPSED event or by the user prematurely ending the round, the controller signals a GAME_END event to all the targets.

<u>Whack-A-Mole</u>

The controller sets a timer for one minute. The controller will randomly select targets and send them TARGET_IGNITE events. At each TARGET_HIT event, the controller increments the user's score and sends a TARGET_EXTINGUISH event to the hit target. If one second has elapsed on an active target and it has not been hit, the controller sends a TARGET_EXTINGUISH event to deactivate it. Once a TIMER_ELAPSED event is sent or the game is quit by the user a GAME_END event is signaled.

<u>Horde</u>

The controller randomly chooses a target to send a TARGET_IGNITE event to. More targets are ignited at random times, 1 to 6 seconds apart. The ignited targets will initially stay active for 12 seconds. With each successful hit, the amount of time the targets stay active decreases by 10 percent. When a target is hit, the user's score is incremented, and the controller will send a TARGET_EXTINGUISH event to turn off that target. If any target times out, or the game is quit, a GAME_END event is signaled.

<u>One-Shot</u>

All targets are sent a TARGET_IGNITE event. The score is incremented at each TARGET_HIT event. If, after the user's last attempt, the controller receives an INPUT_TRIGGER event but does not receive a corresponding TARGET_HIT event within 200 milliseconds, the controller will determine the target has been missed and signal a GAME_END event. Similarly, if the user quits the game, the controller will signal a GAME_END event.

Result State
In this state, the controller will display the user's score. To exit this screen, the controller waits for an INPUT_LEFT event, after which the screen returns to the Ready state.

5.4.3 Target Software Design

For simplicity of design, a target in this system is the simplest component. It handles little game logic on its own, relying upon wireless signals from the controller to change states and statuses. Thanks to this, the software structure of this device is simple and straightforward, with the only additional complexities coming from driving the feedback elements, particularly light.

Once again, we utilized the same base "finite state machine" model for managing the operating modes of the devices. States that the target holds are listed below in Table 19. A diagram of the target finite state machine can be seen in Figure 62 below.

STATE	DESCRIPTION
INITIALIZING	The microcontroller is communicating with and configuring the various hardware components of the device, including establishing wireless communication, loading configuration variables, and preparing the device for use.
PAIRING	The microcontroller is waiting in pairing mode, connecting and communicating with the controller.
READY	The microcontroller is ready to operate, maintaining any valid connections and waiting on system events.
PLAY	A game is currently running. The microcontroller waits for game events to be received wirelessly, or for a laser shot to be detected

Table 19: Target States Table

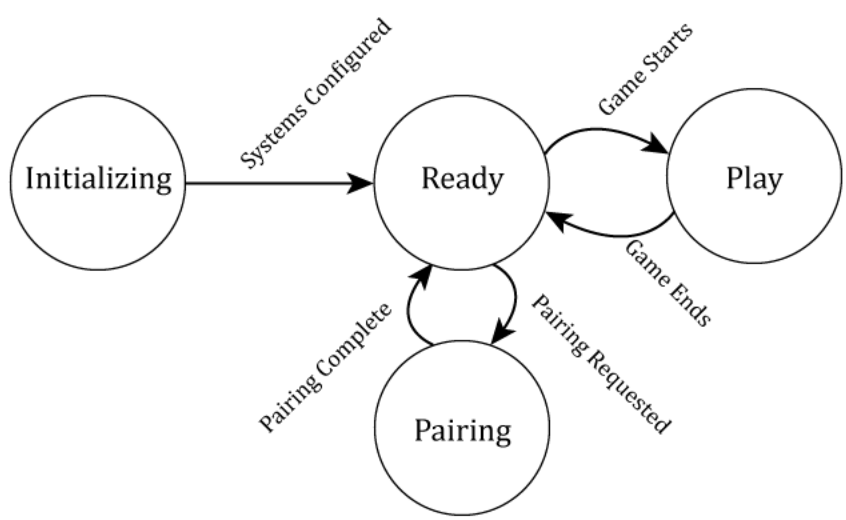


Figure 62: Target Finite State Machine Diagram

A target starts in the Initializing state, reading data from the non-volatile memory to configure and confirm the status of various peripheral devices on the target. Once all devices are configured, it enters the Ready state.

In the Ready state, the target sits waiting for a game start event from the wireless system – or, for a pairing mode event from the button input on the target itself, which shifts it into the Pairing state.

When in the Pairing state, the target sits on an open network waiting for a laser hit to occur – when it does, it sends out a pairing request to the controller on the network. Such an event can be either accepted or rejected – both of which leave it in the current state, until pairing is complete, upon which we return to the Ready state.

A game start shifts the device into the Play state, where the target reacts to game events, like being hit, or status commands from the controller. It stays in this state until the game ends, at which point it returns to its Ready state.

Below, Table 20 lists the events that a target can either raise or expect to handle.

Event	Channel	In/Out	Relevant States
INPUT_PAIRING	Hardware	Incoming	Ready
INPUT_LASER	Hardware	Incoming	Pairing, Play
PAIR_REQUEST	Wireless	Outgoing	Pairing
PAIR_REJECT	Wireless	Incoming	Pairing
PAIR_ACCEPT	Wireless	Incoming	Pairing
PAIR_COMPLETE	Wireless	Incoming	Pairing
TARGET_HIT	Wireless	Outgoing	Play
TARGET_IGNITE	Wireless	Incoming	Play
TARGET_EXTINGUISH	Wireless	Incoming	Play
GAME_START	Wireless	Incoming	Play
GAME_END	Wireless	Incoming	Play

Table 20: Target Events

5.6.3.1 Lighting Control

Each target has a unique "lighting" element onboard, which is an array of individually addressable RGB lights mounted to the user-facing surface of the device. This helps signal to the user not only where to shoot, but also the status of the target. It can also be used to draw attention or tell a story during gameplay.

"Lighting Programs" are written to describe and control a lighting effect that can be triggered by the TARGET_IGNITE or TARGET_EXTINGUSIH signals. These are just small functions used to control a pattern of lighting, stored constant in the microcontroller. A small library of these were authored for use in game modes and various system status signals.

The lights used on our targets are the WS2812 Integrated Light Source – a mix of an RGB LED and a small driver chip in order to control and configure the light output. These are chained together in order to give us control over a whole array of light sources with only one I/O point. These lights are controlled like a shift register, where the output of one device is daisy-chained into the input of the next. This way, we sequentially send control information to the first light device, which then feeds it into the next device in the chain, and so on.

As the WS2812 Integrated Light Source relies upon tight timings and precise data signals, we relied on one of the existing lighting control libraries to manage sending data from the microcontroller to the lights. While this is standardized, we still needed to create a manager to handle lighting triggers, specific lighting sequences, and other related functionality.

One of the functions of the manager is using a custom protocol we developed for creating lighting effects. When the controller instructs a target to change lights, it sends an effect code in the following format:

LightingPattern|Loop|Clear|StartTime|Timeout|Freq|PrimaryColor|SecondaryColor.

Lighting Pattern is a hex-coded single character describing the pattern the lights should make. Options include a static color, blinking of every light in the strip, or a marching blinking pattern. Loop is a Boolean for determining if the pattern should be played indefinitely, ignoring the given Timeout value. Clear is a Boolean for determining if the target's light strip should be cleared or remain as it was when the current lighting pattern has been stopped. StartTime is a 32-bit unsigned integer indicating the time in milliseconds that the effect began. Timeout is an unsigned long representing the amount of time the pattern should run. Frequency is an unsigned long indicating the frequency that the effect should play. Primary and Secondary Color are both 32-bit unsigned integers containing the RGB values of the colors being displayed on the light strip.

5.6.3.2 Target Visual and Design

Initializing

In the Initializing state, the system is waiting for the target to finish preparing its modules for gameplay, so the targets do not display any light or patterns.

Ready

In the Ready state, all paired targets turn green (RGB 0, 255, 0) in preparation for the game to begin.

Pairing

In the Pairing state, targets that have not been paired are completely lit in red (RGB 255, 0, 0). Targets that have successfully completed pairing turn yellow (RGB 255, 255, 0).

Play

The following is a discussion of how lighting effects are used to enhance the user experience of the various gameplay modes.

<u>Time Trial</u>

In Time Trial, all paired targets turn green. When a user shoots the target, all lights turn off. Once all targets have been hit, all targets light green again.

<u>Whack-A-Mole</u>

In Whack-A-Mole, when a target is first ignited, it turns green. When a target is hit or it has timed out, all lights turn off.

Horde

In Horde, the color of the targets changes depending on how long the user has left to shoot them. A target with a large amount of time remaining starts out as green, transitions to yellow as more time passes, then finally turns red and blinks as the user begins running out of time. As the amount of time the user has left for each target decreases over the course of the game, these lighting effects are defined as a percentage of the time allotted for a specific target. These specifications are listed in Table 21 below. Additionally, when a target is hit, it turns white before reactivating and becoming green again.

Time Remaining	Color	RGB Values
80 – 100% of Allotment	Green	(0, 255, 0)
40 – 79% of Allotment	Yellow	(255, 255, 0)
0 – 39% of Allotment	Red	(255, 0, 0)

Table 21: Color Specifications for Targets in Horde

One-Shot

In One-Shot, the targets remain completely lit green, as all targets stay active for the entire course of gameplay.

5.6.3.3 Target Event Control

Initializing

In the Initializing state, the targets do not send or receive any events.

Ready

In the Ready state, the target idles until either pairing is initiated or the game begins. If the user presses the pairing button on the target, an INPUT_PAIRING event is initiated, and the target enters the Pairing state. If the target receives a GAME_START event from the controller, it enters the Play state.

Pairing

In the Pairing state, the target waits until it receives an INPUT_LASER event, signaling the target has been hit. Once hit, the target sends a PAIR_REQUEST event to the controller and waits for either a PAIR_ACCEPT or PAIR_REJECT response. A target receives a PAIR_COMPLETE event once the user has finished pairing devices, returning the target to the Ready state.

Play

Play mode is entered when the target receives a GAME_START event. In all game modes, the target receives an INPUT_LASER event when it is hit. After being hit, the target sends the controller a TARGET_HIT event. The target responds to TARGET_IGNITE events from the controller by becoming active and applying effects according to the effect code sent by the controller. Similarly, the target deactivates on TARGET_EXTINGUISH events, also applying the effect code specified by the controller. A target is sent back to its Ready state when it receives a GAME_END signal from the controller.

5.4.4 Events Specifications

Events, in event-based programming, typically carry some informational data as a payload. This provides some further context to an event, outside of just the event's type itself. In the following Table 22, we expand upon the various events that these devices encounter during operation.

Category: Input Events	These are events which are triggered by some hardware input event – typically a momentary button or some change in an analog signal.	
Event	**Device**	**Data**
INPUT_LEFT	Controller	None
Description:	An INPUT_LEFT event occurs on the leading edge of a hardware button's signal change. Used to navigate left in menu options.	
INPUT_RIGHT	Controller	None
Description:	An INPUT_RIGHT event occurs on the leading edge of a hardware button's signal change. Used to navigate right in menu options.	
INPUT_TRIGGER	Controller	None
Description:	An INPUT_TRIGGER event occurs on the leading edge of the trigger button's signal change. Used to select items in menu options, or to signal to the controller to fire the laser.	
INPUT_LASER	Target	Analog Voltage (laser strength)
Description:	An INPUT_LASER event occurs on the leading edge of the analog input's signal change for the phototransistor. Used to detect incoming laser hits. The analog voltage can be used to determine how accurate the shot was – the dimmer the light, the lower the voltage, the further from center. A suitable threshold must be explored to find what is considered a "hit"	
INPUT_PAIRING	Target	None
Description:	An INPUT_PAIRING event occurs on the leading edge of a hardware button's signal change. Used to send the device into its Pairing state.	
Category: Pairing Events	These are wireless events which occur during the pairing process and are used to coordinate connection efforts. All are subsets of an abstract "Wireless" event, which happens whenever a message is received over wireless communication.	
Event	**Device**	**Data**
PAIR_REQUEST	Both	Device ID
Description:	A Pair Request is broadcast wirelessly by a target when it is hit by a laser in pairing mode. The target broadcasts to all devices information about itself, including its device	

	ID. A controller in pairing mode listens for this request, then evaluates it before sending a response.	
PAIR_REJECT	Both	Reason
Description:	A Pair Rejection event is sent by a controller when a PAIR_REQUEST it receives is somehow invalid or unacceptable. In such a case, it attempts to send this event wirelessly back to the device that sent it. When a target receives such a rejection, it maintains its un-paired state, and signals that something went wrong.	
PAIR_ACCEPTED	Both	Device ID
Description:	A Pair Accepted event is sent by a controller when a PAIR_REQUEST it receives is valid and acceptable. In such a case, it sends this event to the target that sent it, including some information about the controller's device information. When a target receives such a message, it signals that it is paired, but idles while waiting for PAIR_COMPLETE	
PAIR_COMPLETE	Both	None
Description:	A Pair Complete event is broadcast by a controller when a user decides that they're done pairing with targets (via a button press). When a pair complete event is received by a target, it shifts into the "Ready" state.	
Category: Target Commands	These are wireless events which control or signal the status of a target during gameplay.	
Event	**Device**	**Data**
TARGET_HIT	Both	Intensity
Description:	A TARGET_HIT event is sent to the controller by a target when it detects a laser hit. The "intensity" parameter is used to denote how bright the laser hit was, which could be utilized for accuracy measurements.	
TARGET_IGNITE	Both	Effect Code
Description:	A TARGET_IGNITE event is sent to the target by a controller to signal that a target should "turn on," with the optional control of "Effect Code" to signal visual or audio effects for the target to play. It can be sent to already "on" targets in order to shift the visual or audio effects.	
TARGET_EXTINGUISH	Both	Effect Code
Description:	A TARGET_EXTINGUISH event is sent to a target by a controller to signal that the target should "turn off," with the optional control of "Effect Code" to be used for triggering specific audio/visual effects to signal the end of being "ignited."	

Category: Game Control	The following are events that are broadcast wirelessly from the controller to the targets to signal various game control events.	
Event	**Device**	**Data**
TIMER_ELAPSED	Controller	None
Description:	A TIMER_ELAPSED event is sent when the controller's internal gameplay clock has expired. It can be used to signal the controller to send out a GAME_END, or it can be used to set up secondary timers for sending additional Target Command events.	
GAME_START	Both	None
Description:	This signal is sent at the start of gameplay from the controller to the target to send the target into its core gameplay handling state.	
GAME_END	Both	None
Description:	This signal is sent at the end of gameplay from the controller to the target to send the target out of its core gameplay handling state.	

Table 22: Event Details

5.5 Enclosure Design

Each device (Target and Controller) requires some form of protective enclosure both to protect the electronics from environmental factors such as dirt, dust, and accidental shorting of circuit components. In addition, a physical enclosure gives each device its form and structure, giving the user either something to comfortably hold or hang on a wall.

In most traditional consumer electronics, the body of a device tends to be composed of plastic or metal, either machined or injection-molded in order to construct a regular, structurally sound, and economically produced component of the device. However, the costs and challenges associated with a traditionally manufactured body are outside both the skillset and budget of our project, leaving us with a common manufacturing technique for prototyping physical components: additive manufacturing. Additive manufacturing is used frequently for developing prototypes, and is cheap, easy, and a straightforward process.

As each device has drastically different form factors, we needed to design each enclosure independently. In addition, the enclosure design impacted the overall design of the PCB and the internal layout of components for the controller, as the form factor of a hand-held "pistol" is demanding.

5.5.1 Target Enclosure Design

The Target has the simplest enclosure of the two devices and was the device whose external enclosure was completed first, due to its relatively simple construction. This provided members of the team an opportunity to learn more about the 3D printing process and become more comfortable with the design, manufacturing, and assembly process that we ended up becoming familiar with.

5.5.1.1 Requirements

The Target enclosure must, at a minimum, enclose the custom PCB that holds the critical components of the system. In addition, the battery and the associated circuitry for the battery system must be enclosed.

Some components, like the phototransistor, must be in specific places on the target. The phototransistors must be centered in the target, optically exposed to the environment to allow for a laser to hit it without obstructions. The speaker chosen for this system, too, must be facing forward out of the front of the target, with a speaker grille to allow air to carry soundwaves from the target to the player. The array of WS2182 lights should be facing the user, too, in order to signal to the user the state of the device, show lighting effects, and serve as indicators for where to aim.

Power and Pairing controls should be located on the rear of the device, as they are elements which will not be used during active play sessions, but still accessible for user access.

For maintenance access, the rear panel of the device should be able to be removed entirely and should be affixed to the front using a small count of common screws. An additional hardware requirement is a standard wall-mount hanging hole, like those found commonly on picture frames, in order to facilitate the device's mounting onto walls and other surfaces.

5.5.1.2 Initial Concept Sketch

An initial sketch of the exterior of the target's enclosure is shown below, in Figure 63. This sketch below is not an engineering drawing and has been made to no measured specifications. This was to be used as reference on layout of elements and overall location of components.

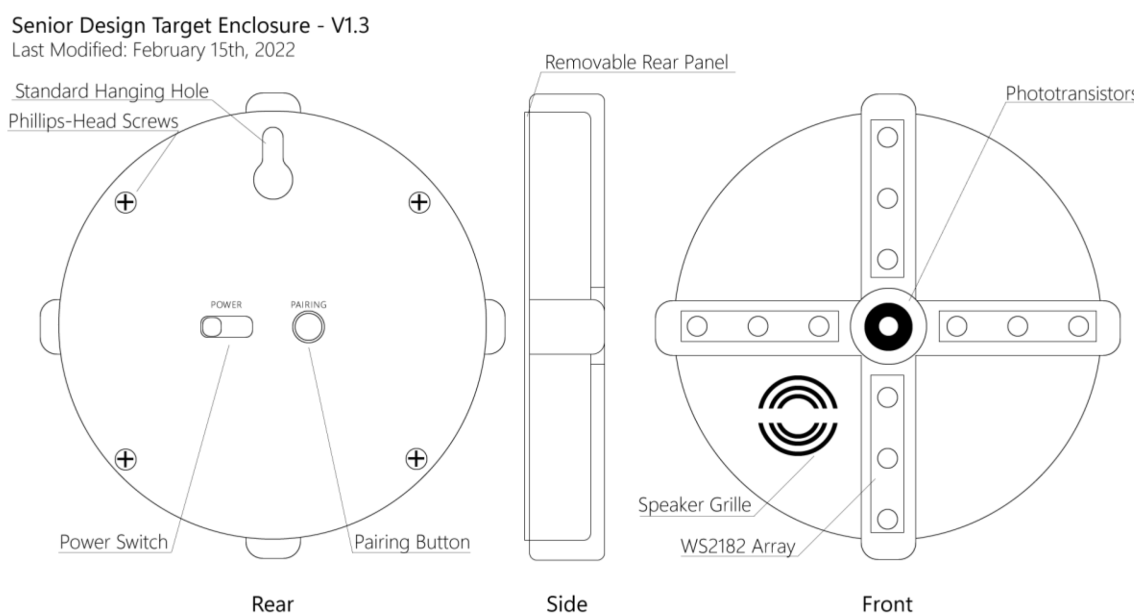


Figure 63: Target Enclosure Sketch - Exterior View

5.5.1.3 CAD Model

The design for the target enclosure was created in Autodesk Fusion 360. The enclosure is designed with three critical components: the main "case" body, which is a cylinder that's open on one side, the "backplane," to which the main PCB is mounted, and the "LED bed," an "x" shaped piece designed to tightly hold the WS2182 LED strips. A screenshot of the overall CAD model for the target enclosure is shown below, in Figure 64, and a screenshot showing just the main "case" of the device is shown in Figure 65.

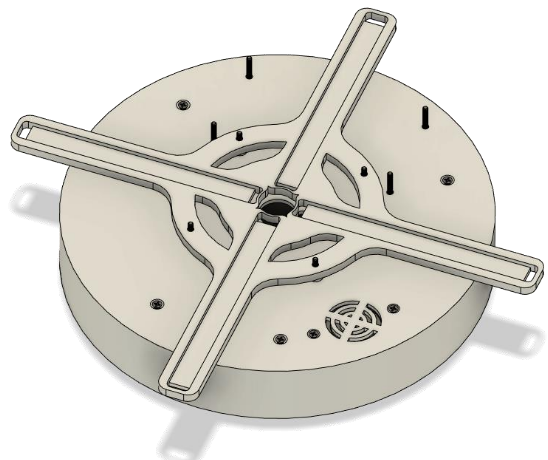


Figure 64: Target CAD screenshot

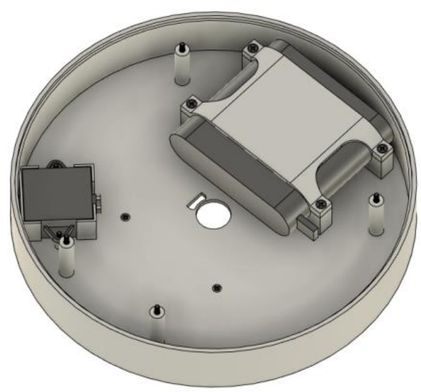


Figure 65: Target Body

The "body" case, shown in Figure 65, only has two components mounted directly to itself alone – the speaker and the battery pack. The battery pack is mounted via an additional part, a "battery clamp," that serves to hold the battery in place without adhesive. Both of these components are, like everything else in the design, assembled through the use of screws and nuts to hold every part in place.

The backplane of the target only has the target PCB mounted directly on its surface – the goal of this was to ensure that the PCB was both able to have directly-mounted components that were accessible by the user (power-switch and pairing button, mounted on the reverse side of the board), and all other assorted components on the other side of the board, for easy assembly. A screenshot of the CAD model for the target backplane is shown in Figure 66.

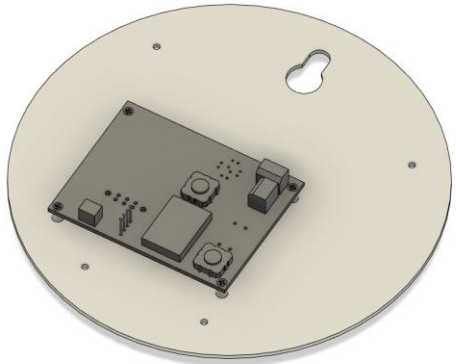


Figure 66: Target Backplane & PCB Mount

In the final assembly, the LED bed is screwed onto the main body with the addition of some plastic spacers, the battery is clamped in place, then the backplane is screwed in after all the internal connections are established. A photograph of the assembled device is visible below, as Figure 67.

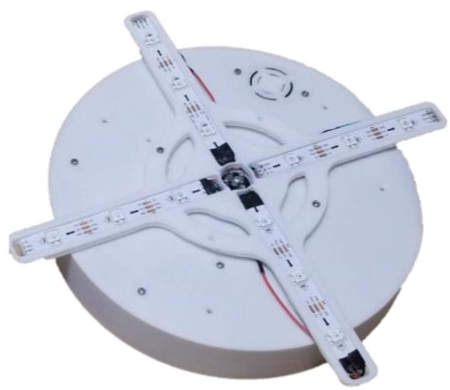


Figure 67: Assembled Target Enclosure

5.5.2 Controller Enclosure Design

The design of the controller's enclosure was easily the most challenging of the two, not only requiring a smaller form factor for more components, but also having to consider user ergonomics. The controller is designed to rest in the palm of a user's hand, much like a traditional weapon, but also contains some heavy components (primarily the battery) which can affect the usability of the "gun" – if the "gun" is poorly balanced, too heavy, or just cumbersome to use, a user could experience the overall interaction poorly, even deciding that playing was too uncomfortable.

Designing an enclosure that rests comfortably in the hand was an iterative process, one where we made small adjustments over and over to get the weight, balance, and grip of the device correct. On top of that, sandwiching all of our components together inside of the small enclosure proved to be another set of unique challenges.

5.5.2.1 Hardware Requirements

The notable hardware components of this device are the laser diode, speaker, OLED screen, control buttons, and the trigger. The laser diode is the easiest to place – it must point out the front of the controller, laid along the center line of the controller's barrel. The speaker is large and thick and must be pointed towards the user in some way in order to get the best effect – so pointing it up and out of the side of the barrel is our best bet. The OLED screen must be easy for a user to see, and the control buttons shouldn't be too far out of the reach of the user's position where they rest their hands.

The trigger's mechanics were taken care of for us by the part we selected, however, ensuring that the part stayed solidly in place under repeated, frequent toggling of the switch was another notable challenge, especially considering the lack of mounting points on the part itself.

5.5.2.2 Initial Concept Sketch

An initial sketch of the exterior of the controller's enclosure is shown below, in Figure 68 This sketch below is not an engineering drawing and has been made to no measured specifications. This was to be used as reference on layout of elements and overall location of components throughout development.

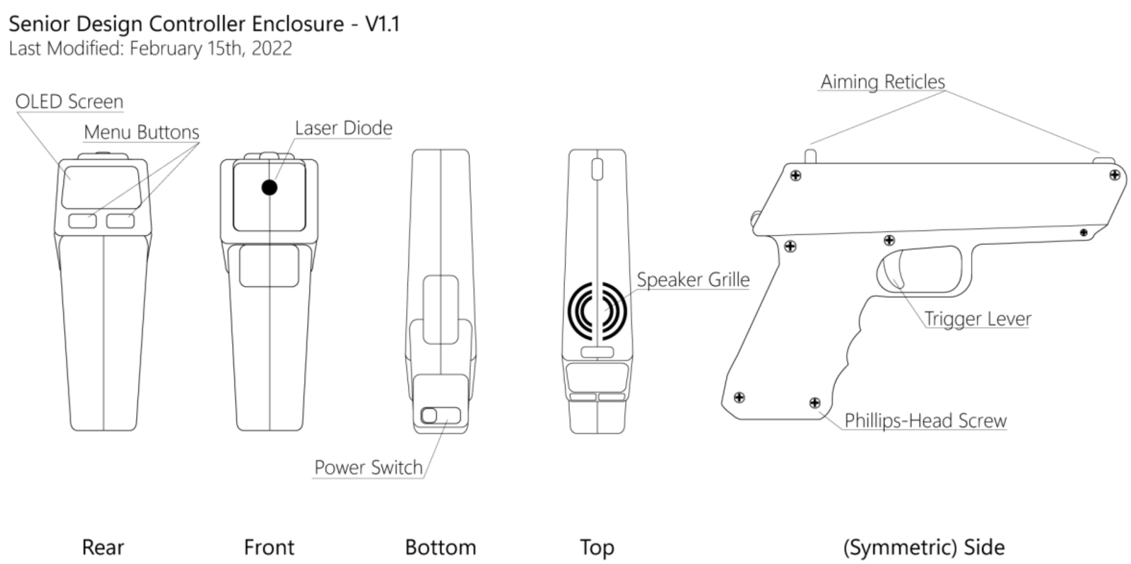


Figure 68: Controller Enclosure Sketch - Exterior View

5.5.2.3 Manufacturing & CAD Design

As previously discussed, the controller is a more complicated device than the target in construction and grip. In order to ensure our ergonomics were comfortable and accurate, we first designed and manufactured a system of parts that served to be a modular, adjustable system of pieces early during the second semester to serve as a first prototype of the controller. Looks, polish, and other factors were discarded in favor of modularity and print efficiency. This can be viewed as Figure 69. This helped quickly find out a combination of button, screen, and trigger positions that were comfortable and easy to use.

Figure 69: “Grip” prototype controller

After we solidified the part positions, we designed the final model in Fusion360 by just copying the final core part positions into a new document and designing around them, with the goal of making it similar in construction to the target – a core body and a body cover. However, due to the battery size, we were unable to fit the battery inside of the controller itself and mounted it on the bottom of the grip of the device, making it balanced evenly in the user’s hand. Figure 70 showcases the final cad model, including the main body on the right, the barrel cover on the left, and the “sandwich” grip in the bottom center – a solution to hold the trigger without adhesive or glue by squishing it between to plastic parts. You can see a picture of the finally assembled target in Figure 71.

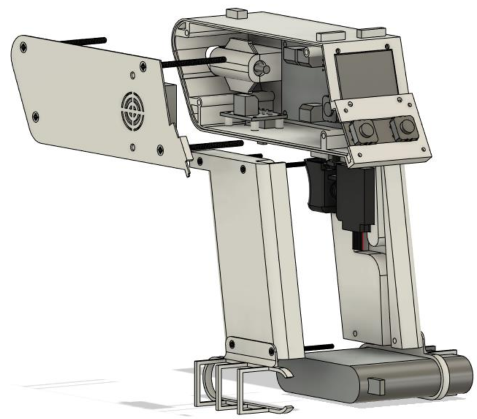


Figure 70: Controller CAD Model

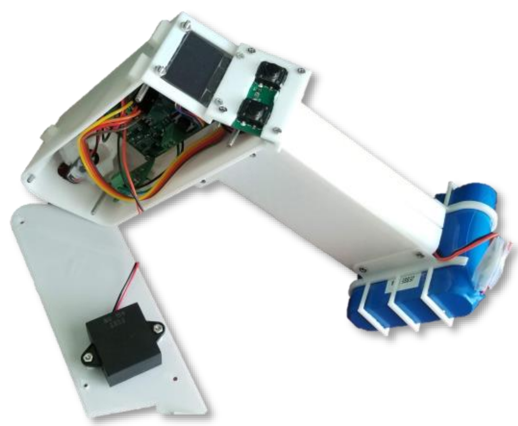


Figure 71: Final Controller Assembly.

5.6 Summary of Design

The overall design of this system was something uniquely challenging – as we were effectively building two separate devices, we needed to put extra effort into the development and planning process of each device in order to ensure that our final efforts were not wasted. Two devices meant two separate software builds, two separate PCB designs, two microcontrollers, etc. Therefore, our designs needed to be sound before even getting into the production phase of our project.

We attempted to reduce the complexity that came with having two sets of devices by ensuring that hardware was mostly standard between the pair – the same battery management system, the same basic voltage regulator design, the same microcontroller, and even audio hardware. This meant that the vast majority of the work could be done once, and only required minor alterations in order to accommodate the requirements of any specific device.

The same design goals held true for our software: write once, adjust twice. We laid out our plans on how to achieve this, building a core codebase to work from that then could be recycled into two different systems, each of which could stand alone, while not having to repeat our code or development time.

Unfortunately, the enclosure and PCB design for our two devices were unable to take advantage of our vertical integration efforts. Both devices had too wildly varying constraints for user experience, functionality, and design goals. Instead, our efforts were focused on working on both devices in parallel, communicating what knowledge and experience we gained on one set of electronics and enclosure design with the other set.

6. Project Prototype Construction and Coding

6.1 PCB Design Software

The first step in the PCB design process was to decide as a group which PCB design software we would be using to make our designs. This section will discuss some of our options and compare and contrast them in a way that is relevant to our needs for this project, as well as discuss our final choice and our reasoning behind it. PCB design software under our consideration included Autodesk Eagle, EasyEDA, and Altium.

6.1.1 Autodesk Eagle

Autodesk Eagle was an obvious consideration for the PCB design software because all team members had experience with it from our Junior Design course, and we are all able to access Autodesk Eagle for educational use for free by confirming our status as UCF students through the Autodesk website. It does bear mentioning, though, that all of our experience with it is still quite limited due to the fact that we all took Junior Design during the COVID-19 pandemic, meaning we took it online and in a very limited capacity, and did not get the chance to go through the full design process. This means that we are by no means attached to this particular software, as none of us consider ourselves to be fully comfortable with it. Having a little experience with a software is better than having none, however, so Eagle was still a strong contender.

On the Autodesk website, Eagle is described as a “PCB layout software for every engineer.” The website describes features of the three main sections of the software: Schematic Editor, PCB Layout Editor, and PCB Library Content. Features in the Schematic Editor include a SPICE simulator for testing and validating circuit ideas and performance, modular design blocks that can be dragged and dropped between projects, and electronic rule checking that can be used to check your schematic design to ensure that no electronic rules are violated. Features listed for the PCB Layout Editor include real-time design synchronization between the schematic and the PCB layout, intuitive alignment tools, push and shove routing, obstacle avoidance routing, and design rule checking. Lastly, features given for the PCB library content include managed online libraries, 3D PCB models, and complete out-of-the box component libraries that include the component’s symbol, footprint, 3D model, and parametrics [66].

One very important aspect of a good software for PCB design for our group is the ability to collaborate on the design as a group. After a thorough perusal of the Autodesk website, this does not appear to be a feature that Eagle has integrated into the software, meaning that if we were to use Eagle for our design, we would have to be constantly sending each other updated files every time we made a design change, which could become frustrating and confusing for all group members involved. While this is an obvious disadvantage, Eagle is still an extremely useful and powerful PCB design software that we all have a bit of experience with, so it cannot be removed entirely from consideration.

6.1.2 EasyEDA

EasyEDA is described on its website as "An Easier and Powerful Online PCB Design Tool." This statement, and more, is what made EasyEDA desirable to our group for use in the project. Because our group has extremely limited experience with PCB design, we are looking for an option that is easy to understand and user-friendly, which EasyEDA appears to be. It can be accessed online, and there is also the option to download its desktop client. The core functions and features of EasyEDA are free to use, but if for any reason our group wanted to purchase a subscription to it, the monthly subscription options are extremely affordable ($5 or $10 per month) compared to most other PCB software subscriptions (hundreds or sometimes thousands of dollars per month).

The two most attractive features of EasyEDA for our group are its web-based functionality and its team collaboration capabilities. EasyEDA can be opened and used in most web browsers, and if the user saves their progress, they can resume their work at any time from any device with web browsing capabilities, and files can be saved into the cloud. EasyEDA also offers real-time team collaboration over the internet, so multiple group members can work on the design simultaneously, and all group members will be able to access the latest version. This flexibility, ease of access, and ease of collaboration is much more preferable for our project than locally saved files that can only be accessed from a single device by a single user.

Because EasyEDA is free and easy to use for beginners, one might expect for it to be lacking in major important features. At least for the scale of our project, we found this assumption to be false. The EasyEDA Schematic Capture module offers SPICE simulations, a waveform viewer, and multi-sheet schematics. Its PCB layout module includes Design Rule Checking (DRC), multi-layer PCB layouts, Gerber file export, auto-routing, and more. It also includes bill of material (BOM) generation, and extensive component libraries containing over 1 million parts, which can be accessed directly from within the EasyEDA workspace. Between all of these features, EasyEDA should have everything we would need to design the PCBs for this project [67].

6.1.3 Altium

A third potential PCB design software for our project, Altium, was proposed by our group member Jamauri, who has done some work with it in the past. It is usually a paid software, but it offers a student license that we can take advantage of as UCF students. Just like Eagle and EasyEDA, it has all of the standard PCB design software offerings, including schematic capture, in which the schematic of the PCBs can be designed, as well as a SPICE simulator in which the design can be simulated. It also offers interactive and automatic routing for the PCB design, and a board layout tool in which the PCB layout can be viewed and planned in 3D.

One enhanced feature that it offers is an intelligent library management platform that provides schematic symbols, PCB footprints, lifecycle status, and supply chain planning all in one centralized location. This feature especially makes Altium a very desirable software, as finding schematics and footprints for all of the parts and ensuring that all parts are available in sufficient quantities from a number of vendors is one of the more time-consuming parts of the PCB design process.

Altium appears to also offer some team collaboration tools, though they are not as immediately obvious and easy to find as those offered by EasyEDA, but it is certainly a feature to investigate further if we decide to use this software.

6.1.4 Final PCB Design Software Choice

For the creation of the schematics in Senior Design 1, we ended up using both EasyEDA and Altium simultaneously. This was made possible by the ease with which each software can import schematics from the other. Originally, the plan was for all members working on hardware to use Altium for the schematic design. Unfortunately, the Altium student license took longer than expected to acquire, so only one group member, Jamauri, acquired it in time to use it during Senior Design 1. Jamauri worked on some of the schematics in Altium, and Rachel used EasyEDA to create additional schematics that could be worked on independently. Both programs allow schematics to be exported and imported between them. Therefore, the schematics were designed using a combination of Altium and EasyEDA.

During Senior Design 2, all final schematics were created, revised, and completed in EasyEDA, from which they were exported to the manufacturer directly.

6.2 PCB Manufacturer and Assembly

After selecting the software that will be used to design the PCBs, the next step in the PCB creation process is selecting a manufacturer to build the PCB, and to decide how the final PCB will be assembled with all of its components. This section will discuss PCB manufacturers under consideration (including their pros and cons) and our plan for assembling the final PCB with all of its components once the manufactured PCB arrives.

6.2.1 Manufacturers

Considerations for each PCB manufacturer include price, location, shipping speed, and each manufacturer’s reputation for overall PCB quality. Price is definitely a lower-priority consideration, however, because we are willing to pay a higher price if needed for a faster shipping speed and a higher-quality board. Manufacturing location is a factor that has become increasingly important in the past couple of years, since the COVID-19 pandemic caused major supply chain and shipping speed issues, particularly overseas. Therefore, assuming we choose an overseas manufacturer as our first choice, it would be wise for us to choose one or more backup manufacturers that are much closer (preferably located in North America) to ensure that we get our PCBs on time no matter what happens. This section will contain discussions of each manufacturer under consideration, as well as how they measure up to our desired characteristics.

Because our final PCB design layouts will be completed during Senior Design 2 and we do not yet know their exact dimensions, we arbitrarily chose a quantity and set of dimensions to use to get a quote from each website. This will not tell us exactly what our final PCB designs will cost to manufacture at each company, but it will tell us what the prices of each manufacturer are when compared with one another for the same design, giving us an accurate idea of what the price differences between them may be for any given final PCB design. All quotes were created for a quantity of five 100x100mm (3.94x3.94in), 2-layer, 1.6mm (0.062in) thick PCBs. Table 23 below compares quotes from a number of different PCB manufacturers.

Manufacturer	**Location**	**Quoted Cost**	**Shipping Cost (Shipping Speed)**	**Lead Time**
JLCPCB	China	$2.00	$18.80 (2-4 business days)	1-2 days
PCBWay	China	$5.00	$19.75 (2-4 business days)	24 hours
Elecrow	China	$4.90	$24.16 (5-8 business days)	4-7 business days
BasicPCB	USA	$125.00	Free USPS shipping	1-2 weeks
Bittele	Canada	$132.54	Not listed	5 days

Table 23: Comparison of PCB Manufacturers

As can be seen in the table, there are a great number of manufacturers in China (some not even listed), and they have the cheapest manufacturing costs by far, as well as incredibly reasonable shipping prices and speeds. In contrast, manufacturers in North America were extremely hard to find, and the prices were quite extreme, with cheaper shipping, but much slower shipping speeds, if they gave any shipping information at all. It is therefore our hope that the supply chain and shipping from China remains relatively uninterrupted throughout the course of our Senior Design 2 semester. However, if this does not turn out to be the case, the North American manufacturers listed in the table above can be used. This must be a worst-case scenario, however, because their prices are much more than we would like to pay.

The current plan is to use either JLCPCB or PCBWay. Both are incredibly cheap to manufacture and have extremely quick turnaround times, and very reasonable shipping speeds and prices, usually under a week for less than $20. The final decision for which manufacturer to use will be made as a group once our final PCB designs are nearing completion (during Senior Design 2), so we can ensure that our designs are compliant with the policies and procedures of the manufacturer of our choice, and we can get accurate quotes from both sites for the dimensions and specifications of our final designs. Once we place an order with either company, based on their estimated lead and shipping times, that order should arrive in around a week, give or take a few days. This will mean that we should plan ahead for the possibility of making multiple orders (at least three) and factoring in the time for them to arrive and time to test and adjust our designs, in case our first attempt(s) have problems with their design that need to be changed. This will be built into our schedule for Senior Design 2.

During Senior Design 2, we decided to go with JLCPCB because of its fast shipping times, low manufacturing costs, and direct integration with EasyEDA, which we used to design all PCBs. Shipping was more expensive than originally expected, however, because we chose to order stencils for all main PCBs. Stencils cost about $40 each to ship, and we had two of them in every order, so they ended up being the most expensive item in the project.

6.2.2 Plan for Assembly

The next consideration that needs to be made for our PCB design is the assembly of the finished PCB, with all components attached. There are a number of different methods and resources we can utilize to accomplish this task, all of which will be mentioned here. No matter which method we choose, we should be able to solder some of the bigger and/or more unique components ourselves, especially through-hole components such as pin headers and phototransistors. Rachel and Jamauri both have experience working in an electronics lab and should be able to handle the soldering of these larger components easily. However, we should absolutely make use of any professional PCB assembly services that we can afford for the smaller components, or the components with many pins (like the microcontroller) in order to avoid damaging our PCBs or any of our components by exposing them to excessive heat during hand-soldering. The remainder of this section will mention a couple of overseas manufacturers as well as a local manufacturer that offer assembly services, all of which are under consideration for our project.

The first possibility for attaching many of the components is using the assembly services offered through JLCPCB and PCBWay. At the time of writing this section, JLCPCB is offering their assembly services for free, and PCBWay is offering theirs for $30 per 20 pieces. This adds a few days to the lead time on both sites, but that time would be well worth it if we did not have to solder many of the smaller components to the boards ourselves or take them somewhere to be soldered. Both sites have an extensive parts library that can be pulled from for this assembly, which will likely contain many of the minor components that we will need. If we choose to have them pre-assembled, we can handle the soldering and/or assembly of the more unique components (such as the laser diode and the phototransistor) ourselves.

The next possibility is making use of a local manufacturer such as Quality Manufacturing Services, which offers PCB assembly services. They are located in Lake Mary, FL, and often work with senior design teams to help them assemble their PCBs. We know of multiple past senior design groups who have utilized their services and have been satisfied with them. Their services are known to be incredibly high quality, and they offer the option to “fast track” the assembly if necessary, which could be helpful in the context of our project, where time may become our most precious resource, as we are limited to building the entire complete prototype within one semester. It is not possible to get a quote from them before the PCB designs are completed, but their history with a number of UCF senior design groups leads us to believe that their services are affordable for out group. They will need to be contacted for a quote and to schedule the services at least a few weeks prior to the first time we need their services, so we will keep a close eye on our progress, especially as we finalize our PCB designs, in order to ensure that we give them plenty of warning before we are ready to have our boards assembled. Quality Manufacturing Services has the advantage of being a local business, making communication and scheduling easy, and eliminating the need to worry about shipping time. The main disadvantage is that we would likely need to acquire all of the parts ourselves and supply them with the finished PCB, which is a bit more involved than using the online libraries offered by PCBWay and

JLCPCB, but that is definitely doable and would not prevent us from utilizing their services if it works better for our timeline and budget, and if we would prefer to work with a local business.

During Senior Design 2, we elected to assemble all PCBs ourselves, which was the fastest option. With each PCB order, we also ordered stencils, which helped us to tackle all surface mount components, and we had little trouble with through-hole components. Because Smart Charging Technologies generously let us use a portion of their lab for testing and assembly, we had access to high-quality soldering irons and tools to make it possible for us to assemble the boards ourselves.

6.3 Final Development Plan

This section will outline the details of our development plan, which were implemented during Senior Design 2.

6.3.1 Design Approach

Thanks to the overall architecture of the software being shared between both the controller and target devices, we saved time and improved the consistency of our system by developing a shared "base" project to work from. After we established a common ground to work upon, we later extended our code as required by the functionality of each device.

We accomplished this by breaking our code down into two parts: Hardware Managers and Logic Modules. Hardware Managers included the code for peripheral devices that was shared between the target and the controller. For example, both the targets and the controllers used network messages, so we had a NetworkManager. Both the targets and the controller could simply include these managers whenever they became necessary. Logic Modules differed between platforms. They implemented the logic of the controller and targets, which depended on the state that the platforms were in.

The two computer engineers on the team, Anna Malaj and Thomas Stoeckert, worked in parallel to develop the code. To assist in organization, clarify functionality, and make the development process more efficient, we utilized some basic organizational tools to plan and schedule our software development.

The online service GitHub was primarily used in this effort. While its strength lies in remote repository storage and management, it also includes a wide array of project management tools like automated task / Kanban boards, issues, and documentation [68]. By using Git, we could create a master branch that held the basic, shared functionality of our system. We could then use the code from that branch inside of two additional branches – one for each device. We also used the wiki tools available on GitHub to document our code, helping each developer keep up with the changes that were occurring during the development process.

6.3.2 Development Environment

The integrated development environment (IDE) chosen for development is Visual Studio Code. This IDE was chosen as it is a lightweight, easy to install, and visually pleasing environment that comes with many available extensions for improving development [69]. Both the computer engineers on this project are comfortable with and regularly use this environment, making it a good choice for decreasing the overhead associated with learning new software tools.

To make embedded development on our microcontroller easier, we used an extension available for Visual Studio Code called PlatformIO. PlatformIO provides a simple interface for developing embedded code in the Visual Studio Code environment, including tools for building and uploading code, viewing connected embedded devices, and viewing terminal output. The greatest advantage of the PlatformIO extension is in how it manages the software integration process. PlatformIO makes embedded libraries easily accessible, which can be quickly searched and added to a project. Many popular embedded libraries for the ESP32 are available for download, including painlessMesh for wireless communication, TaskScheduler for cooperative task scheduling, the Adafruit libraries required to program the OLED screen, and additional peripheral device libraries. Once added, these software dependencies are attached by PlatformIO to the project, and they are automatically downloaded and built upon compilation, easing development complexity. Additionally, as a popular extension, PlatformIO comes with a large amount of documentation and user experiences [70]. This was a valuable resource for us when developing our project, and we used this information to diagnose issues quicker.

GitHub was chosen for the source control on this project. In addition to the documentation tools described previously, GitHub provides a tool for efficiently editing code among groups of developers. As GitHub is free, well-documented, and regularly used by both the computer engineers on this project, it was a natural choice for our project. The availability of task management tools on the same service was also important to our decision not to choose a different site for the version control component of our project [68].

7. Testing Plan and Results

Both during and after the construction of our system, we ran a series of tests. We wanted to ensure that the system performed as expected during normal gameplay, and that all parts of the system worked well together.

7.1 Unit Testing

As we were developing our software, we wrote a number of self-contained unit tests for debugging and validation purposes. These tests solely focused on the microcontrollers and were used to ensure they could perform their base functions. We tested the following abilities of both our target and main controller software systems.

7.1.1 LED Test

The target software was designed to be able to control the target’s LED array. To test this ability, we instructed the target to march an LED down the strip, ensuring that each LED was able to be turned on. We also tested various colors on the LED strip, including red, orange, yellow, green, and blue. The test passed, as the target software could successfully control the LEDs.

7.1.2 OLED Test

The controller software was designed to control the OLED screen, including displaying graphics. To test this ability, we instructed the controller to show a preprogrammed splash screen on its OLED display. This splash screen included text in a similar font, color, and position as the text used in the actual finished product. The test passed, as the OLED screen clearly displayed the splash screen.

7.1.3 Speaker Test

The controller was designed to play a “laser hit” sound effect each time the trigger was pulled and the laser was fired. To test the speakers and the controller’s ability to use them, the controller was instructed to play this sound on a loop. This test passed, as the sound was played clearly.

7.1.4 Wireless Communication Test

In our design, it was critical for the targets and controller to be able to communicate signals to each other. To test this ability, the controller was instructed to send each target a message. Each target was instructed to send a response back to the controller. The test passed, as both the targets and controller received their intended messages.

7.2 Integration Testing

Figure 72 shows our completed system, including one target and the controller. After we finished building this system, we tested it as a complete structure to ensure that each component was able to interact with every other component correctly.

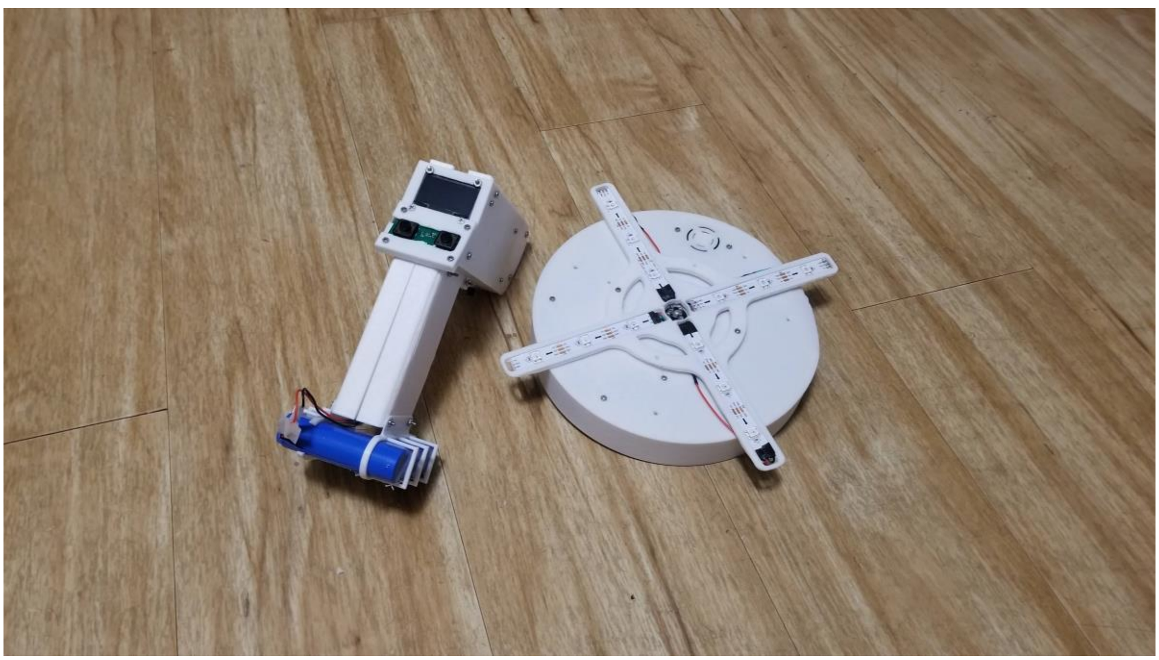


Figure 72: Picture of the Final System

7.2.1 Environment Test

We designed our game to be playable in diverse environments. As such, we tested our system in both an indoor and outdoor environment. Our indoor environment involved a completely enclosed area with over 70 feet of space available for play. Obstacles like walls, mirrors, and other typical indoor objects were available to test our laser’s interaction with different surfaces. For safety purposes, our outdoor environment involved an open, unoccupied area with plenty of space available for play. Obstacles like trees were available to test our system under different outdoor conditions. In both environments, we played a complete round of our game. We tested the system by playing from 6 feet away to ensure that the laser was able to be detected by the targets and that the systems were able to communicate from a reasonable distance. In the indoor environment, we tested our laser by shooting various obstacles and ensuring it was not dangerously reflected. In the outdoor environment, we tested our system in various positions to ensure it still worked as intended. Overall, this test passed, as our system worked according to the design outlined in this document in each of the environments under test.

7.2.2 Range Test

We tested the range of our system in our indoor environment to see how far away the system's network and laser would be able to reach. From approximately 70 feet away, our targets were able to detect the controller's laser, and each device was able to successfully communicate with each other. The only limiting factor to gameplay from this distance would be the user's ability to see and successfully hit the targets. This test showed that our project was able to be used from large distances, adding to our goal of creating a portable system that encourages participants to move around.

7.3 Requirements Testing

After completing development of our device, we tested our entire system to make sure it met all the requirements laid out in Table 1.

7.3.1 Requirements 1 & 2 – Uptime Tests

Requirements 1 and 2 are concerned with maximizing the uptime of each device. To calculate the uptime for each device, we first measured power draw. We plugged each device into a 16V power supply, then continuously measured the current draw for each device over two hours. The devices were tested in parallel, with a program on the controller pulsing the laser every second. The target would send messages back to the controller, ensuring that the power impact of gameplay, including network messages and lighting changes, was accounted for. Using this method, we measured the average current draw for the controller to be 167.88 mA. We measured the average current draw for the target to be 452.67 mA. Given that we were using a 2600 mAh battery pack, we calculated the expected battery lifespan to be 15.49 hours for the controller and 5.74 hours for the target. As such, Requirements 1 and 2 were met.

7.3.2 Requirement 3 – Laser Response Test

Requirement 3 was concerned with minimizing the response time of the system. For this requirement, response time was measured as the time between when the user pulled the trigger and when the target detected a laser hit. This was measured by software running on the controller, which tracked the network time of the most recent trigger pull. The controller listened for a network message from the target containing the time of the associated laser hit, which was emitted when a target detected the laser. When the target hit message was received by the controller, the response time was calculated and displayed on the device. A summary of the average response time at various shooting distances is shown in Table 24. The average response times at all tested distances hovered exceptionally close to our 0.1 second requirement, leading us to conclude Requirement 3 was sufficiently met.

Test Iteration	Time at 6in Distance (µs)	Time at 1ft Distance (µs)	Time at 6ft Distance (µs)
1	101756	108069	109829
2	101831	100109	109360
3	102875	100144	107306
4	104228	100209	101893
5	102084	106058	103798
6	101641	106588	92182
7	100289	106648	93474
8	102058	105840	117592
9	100870	106693	107076
10	102176	102727	104489
Variance (s)	**1.132079733**	**10.020969611**	**57.569254544**
Average (µs)	**101980.8**	**104308.5**	**104699.9**
Average (s)	**0.1019808**	**0.1043085**	**0.1046999**

Table 24: System Response Time Results

7.3.3 Requirement 4 – Startup Test

Requirement 4 was concerned with minimizing startup time. For this test, startup time was measured as the time elapsed between the power switch being turned on and the system being shifted into the Ready state. On the controller, the Ready state was shown by the display becoming the "Main Menu" interface, while an audio clip played. On the target, the Ready state was shown by the lights turning green. A summary of the average startup time for the controller and the target is shown in Table 25. Timing measurements were taken by hand using a stopwatch and should therefore be taken with the understanding of some human error. As shown, the combined startup time of both components of our system met our 25 second requirement.

Test Iteration	Startup Time (s)	
	Controller	Target
1	2.14	1.8
2	2.11	1.73
3	2.07	1.66
4	2.09	1.8
5	1.87	1.67
6	2.09	1.74
7	1.98	1.6
8	2.09	1.67
9	2.17	1.45
10	1.94	1.74

Variance (s)	**0.008939**	**0.010893**
Average (s)	**2.055**	**1.686**

Table 25: System Startup Time Results

7.3.4 Requirement 5 – Pairing Test

Requirement 5 is concerned with the number of clicks required to pair target to the controller. This requirement is included to aid in the expandability of the system. Less effort required by the user to pair targets means more targets can be reasonably paired to the system. To pair targets, the user must navigate menu options to reach the "Pair Targets" option. The user is then required to successfully fire the laser at the target he would like to pair, which may take more than one click if the user does not hit the sensor on his first try. Assuming pairing on the first attempt, pairing can be successfully completed in a minimum of three clicks. As a user can reasonably pair a target within four clicks, we determined this requirement to be met.

7.3.5 Requirement 6 – Energy Test

As described in Section 7.3.1, the average current draw for the controller over the two hours was 167.88 mA, and the average current draw for the target was 452.67 mA. This met our energy requirement for the system.

7.3.6 Requirements 7 & 8 – Weight Tests

Requirements 7 and 8 were concerned with minimizing the weight of the system. This was tested using a common kitchen scale. The weight of the gun was measured to be 427 grams, or approximately 0.94 pounds. The weight of the target was measured to be 441 grams, or approximately 0.97 pounds. As such, these measurements met our specifications.

7.3.7 Requirement 9 – Gameplay Test

Requirement 9 was concerned with minimizing the time the user must wait for the system to be in a playable state. For this test, the "ready to play" state was defined as the time taken from when the system was turned on to when the user could begin playing a game mode. On average, testing showed this time to be between 5 and 10 seconds. A user can reasonably begin playing the game within 2 minutes of startup, meaning this requirement was met.

8. Project Operation Guide

This section will provide an "owner's manual" for users of the Wirelessly Connected Laser Shooting Galley.

8.1 Controller Overview

The profile of the controller is similar to that of a handgun (Refer back to the enclosure design in Section 5.5 Enclosure Design for images). The battery is located at the bottom of the grip of the "gun." The trigger is located at the top of the grip on the side that faces away from the user, easily accessible for the user's pointer and middle fingers to pull the trigger and fire the laser. The laser diode is located on the end of the "gun barrel" that faces away from the user, and the power on/off switch is located on the underside of the "barrel" on the same end. When it is powered on, the controller can be operated with the help of the OLED display and the two menu buttons facing towards the user. Aural feedback is provided via an internal speaker, which is located on the interior left side of the "barrel" when the controller is pointed away from the user (as indicated by the speaker grille on that side). Three protrusions on the top of the "gun barrel" provide a sight to assist with aiming.

8.2 Target Overview

The target is a circular object, similar in profile to a wall clock handgun (Refer back to the enclosure design in Section 5.5 Enclosure Design for images). On its front, it has an LED array in a "+" shape, which provides visual feedback during gameplay. The grille for the internal speaker and the hole for the laser-sensing phototransistor array are located on the front. The power on/off switch and pairing button are located conveniently on the back of the target. There is also a standard hanging hole on the back of the target in case the user wants to mount it on a wall, but it can also stand upright on its own on a flat surface.

8.3 Controller Menu Navigation

When the controller is powered on, its OLED display screen takes the user directly to the Main Menu, which is operated by way of the two menu buttons and the trigger. The menu buttons are used to move the highlighted selection up and down, and the trigger is used to make a selection. From the Main Menu, the user can choose between four options: Select Game Mode, Pair Targets, Change Settings, or About. If the user chooses "Select Game Mode," they are taken to a list of the four game modes: One-Shot, Whack-A-Mole, Horde, and Time Trial. The user can select a game mode or go back to the Main Menu. If the user chooses "Pair Targets," they are able to pair as many targets as they have available. If the user chooses "Change Settings," they will be taken to the settings menu, where they can choose different audio volume and laser settings. If the user chooses "About," they will be able to view information about the device and the project, including project software build info, a page about the senior design group, and a device status page.

8.4 Target Pairing

The process of pairing targets begins at the Main Menu. The user selects the "Pair Targets" option, then the target pairing page will come up. This page shows how many targets the controller is detecting and tells the user to shoot a target in pairing mode to pair it. To place a target in pairing mode, the user must press the pairing button on the back(s) of the desired target(s). Once all desired targets are in pairing mode, the user must shoot each target with the controller to tell it which target is being paired. Once all desired targets have been hit, they are successfully paired, and the user can return to the main menu and begin gameplay.

8.5 Game Modes

The four game modes have been described more technically in other sections of this document, so this section will give only a brief summary of each.

8.5.1 One-Shot

In One-Shot, the user is given three shots to hit a target. If they do not hit a target in three or less shots, the game is over. If they do hit one within three shots, the three shots are replenished, and the round continues.

8.5.2 Whack-A-Mole

In Whack-A-Mole, the targets will all light up green once every second in random order. If the user hits a target while it is lit up green, the user gets a point. If it is not lit up green when the user hits it, a point is not earned.

8.5.3 Horde

In Horde, the game works like a "zombies" game. The targets will gradually progress from green, to yellow, and finally to red as the "zombies" approach. When a target has been hit, it will return to green and begin the cycle again. If a target stays red for too long, the "zombie" has gotten "too close," and the game ends.

8.5.4 Time Trial

In Time Trial, all targets light up green, and the user is given one minute to hit as many targets as possible. When a target has been hit, it is "out" and the lights will go off. All targets must be "taken out" before they will all light up again, meaning that the user cannot keep hitting the same target over and over again, they have to hit them all.

9. Administrative Content

9.1 Milestone Discussion

To ensure the best chance of project success, our team developed a set of milestones, deadlines, and associated tasks to be completed during the Fall 2021 and Spring 2022 semesters. For Fall 2021, each milestone is listed along with the date it was to be completed. Each milestone is further broken into a set of tasks, including information about which team member was to complete the task, and when each task needed to be started and finished. A similar table is provided for Spring 2022. The milestones for Fall 2021 are listed in Table 26 and the milestones for Spring 2022 are listed in Table 27.

9.1.1 Fall 2021 Milestones

Task	Assigned To	Progress	Start	End
Divide and Conquer 1.0 (9/17)				
Choose Project Idea	All	DONE	9/1	9/6
Establish Project Motivation and Goals	Thomas	DONE	9/6	9/8
Determine Requirements	All	DONE	9/6	9/8
Create Block Diagrams	All	DONE	9/8	9/15
Establish Task Breakdowns	Anna	DONE	9/8	9/15
Determine Budget	Jamauri, Rachel	DONE	9/15	9/17

Assign Project Roles	All	DONE	9/15	9/17
Divide and Conquer 2.0 (10/1)				
Refine Requirements, Constraints, and Standards	All	DONE	9/22	10/1
Create House of Quality Diagram	Rachel	DONE	9/22	10/1
60 Page Draft (11/5)				
Research Related Work	Anna	DONE	10/2	10/12
Research Related Technology	Jamauri	DONE	10/2	10/12
Order Parts for Initial Testing	All	DONE	10/12	10/19
Design Software	Thomas, Anna	DONE	10/19	11/5
100 Page Draft (11/19)				
Design and Breadboard Test Hardware	Jamauri, Rachel	DONE	11/5	11/15
Create Essential PCB Schematics	Jamauri, Rachel	DONE	11/15	11/19
Write and Test Software	Thomas, Anna	DONE	11/15	11/19
Finalize Part Selection	All	DONE	11/15	11/19
Final Report (12/7)				
Continue Prototyping	All	DONE	11/19	12/7
Define Testing Plan	Anna	DONE	11/19	12/7
Proofread Final Document	All	DONE	11/19	12/7

Table 26: Fall 2021 Milestones and Task Breakdown

9.1.2 Spring 2022 Milestones

Task	Assigned To	Progress	Start	End
CDR Presentation (2/18)				

Work On Hardware and PCB Design	Jamauri, Rachel	DONE	12/7	2/16
Work On Core Software	Thomas, Anna	DONE	12/7	2/16
Create Presentation	All	DONE	2/16	2/17
Complete CDR Reviews	All	DONE	2/4	3/4
Midterm Demo (3/21)				
Finish Hardware and PCB Assembly	Jamauri, Rachel	DONE	2/17	3/20
Finish Core Software	Thomas, Anna	DONE	2/17	3/20
Showcase Documentation (4/15)				
Complete Project Summary	Anna	DONE	2/17	2/20
Complete Showcase Demo Video	All	DONE	3/21	4/15
Final Presentation (4/20)				
Create Faculty Panel	Rachel, Anna	DONE	1/14	2/17
Fine-Tune PCB Design and Assembly	Jamauri, Rachel	DONE	3/21	4/1
Finish Enclosure	Thomas	DONE	3/21	4/1
Finish All Software	Thomas, Anna	DONE	3/21	4/1
Complete Device Testing	All	DONE	4/2	4/4
Write Conference Paper	Rachel, Anna	DONE	4/5	4/8
Create Presentation Video	All	DONE	4/5	4/19
Create Demo Video	All	DONE	4/5	4/19
Attend Faculty Review Meeting	All	DONE	4/20	4/20
Final Documentation (4/26)				

Complete Website	Anna	DONE	4/20	4/26
Complete Peer Reviews	All	DONE	4/20	4/26
Complete Exit Survey	All	DONE	4/20	4/26
Complete Paper Revisions	All	DONE	4/20	4/26

Table 27: Spring 2022 Milestone and Task Breakdown

9.2 Budget and Finance Discussion

The following is a discussion of our project budget and financing.

9.2.1 Total Cost of All Parts and Shipping

Table 28 shows the total cost of all parts ordered during our design process. This includes parts used to experimentally determine the best fit for our project needs. Multiples of most parts were also ordered as backups in case a part was faulty or was damaged during testing, and due to supply chain uncertainties. As such, this table includes parts that both do and do not appear in our final design. It also includes shipping costs, which made up a huge part of our expenditures. This amount came out to $630.67.

Item Type	Item Name	Bought From	Individual Price	Quantity	Total Cost
Microcontroller	ESP32-WROOM-32D	Mouser	$4.00	10	$40.00
Laser Diode	5mW 650nm Red	Adafruit	$5.95	3	$17.85
Phototransistor	ALS-PT204-6C/L177	Digi-Key	$0.50	15	$7.50
Battery	Efest Li-Ion 14.8V 2600mAh	SCT	FREE	4	$0.00
3.3V Regulator	UA78M33CDCYR	Mouser	$0.75	10	$6.58
5V Regulator	MC7805CDTRKG	Mouser	$0.69	10	$5.88
Controller Display	Monochrome 1.3" 128x64 OLED Graphic Display	Adafruit	$19.95	2	$39.90
Target LED Array	NeoPixel Digital RGB LED Strip	Adafruit	$16.95	2	$33.90
Audio Amplifier	MAX98357AETE+T	Digi-Key	$2.68	10	$26.80
Speakers	MakerHawk 2PCS 4 Ohm 3-Watt	Amazon	$10.99	2	$21.98
Misc. Passive Components	Misc.	Misc.	Varies	N/A	$75.00
PCB & Stencil Manufacturing + Shipping	3 Controller & Target PCB Orders	JLCPCB	$81.57 (1st) $74.57 (2nd) $79.14 (3rd)	2	$235.28
3D Printer Filament	White 1kg Spool 1.75mm	Amazon	$20	1	$20
Testing Parts (not in final design)	Misc.	Misc.	Varies	N/A	$100.00
Total	-	-	-	-	**$630.67**

Table 28: Overall Project Budget

9.2.2 Final Project Budget – Unit Costs (1x Controller, 3x Targets)

Table 29 shows our final project budget, which includes only parts that were used in the final product (also known as the unit costs) which were calculated in a spreadsheet. This is the total cost of materials included in the final product: one controller and three targets. As shown, the actual cost of materials for the project is quite low, just over $200. The main project expenses were shipping costs, testing materials, and surpluses ordered because of unreliability in the supply chain. Our project members split all costs evenly.

Item	Unit Cost	Quantity	Total Cost
Controller	$77.73	1	$77.73
Target	$40.96	3	$122.88
Total	**-**	**-**	**$200.61**

Table 29: Final Project Budget

10. Project Summary and Conclusion

10.1 Project Summary

This project idea began with a bit of inspiration from laser shooting game-type projects done by previous Electrical and Computer Engineering Senior Design Groups, but during the course of these past 8 months, we have developed the idea into a project that is unique and entirely our own.

Growing this project from the initial concept to a fully-fledged prototype has been a real learning experience for all of our group members. In our first semester and Christmas break, we started with brainstorming sessions around a whiteboard, chose an idea that we all liked, researched products and components that we thought might work in our design, planned out the software design, ordered components for testing, tested those components along with our software in the lab, created PCB schematics based on the results of our research and testing, created final PCB layouts, fine-tuned our complete hardware and software designs. In our second semester, we ordered and redesigned our PCB to be as effective as possible, assembled our PCB and hardware, created a complete software suite for our system, 3D-printed enclosure designs for three targets and our controller, thoroughly tested all components of our project, and presented our work to our peers, our advisors, and our faculty.

10.2 Concluding Thoughts

This process of going from a simple idea or inspiration to a full project is an extremely important part of the engineering process, and it is our main takeaway from Senior Design. By doing all the research, development, and testing needed to complete our Senior Design sequence, our group members have all learned a number of important skills that will be useful to us as we pursue our engineering careers. These skills include but are not limited to working as a team, taking a project from an abstract concept to a detailed design plan, component research and selection, software design, PCB schematic design, technical writing, and presentation skills.

The most valuable of these skills that we have gained is the ability to work with a team on a project that requires a significant amount of design. This is something that we will be required to do every day in our jobs as engineers, and therefore something that we should practice while we are still on college, which Senior Design has given us the ability to do.

We have learned how to compromise and make design decisions that will work for everyone in the group and not just ourselves. This helped to keep the work balanced between us and ensured that everyone was working to their strengths.

Most of our group members have never created an engineering project of this scale from start to finish, and we were very excited to get the chance to see the results of all of our hard work as a real, tangible device. Additionally, we decided to create this system because it was something that we could all see ourselves having fun playing, so we were very happy to play the game ourselves for fun. In our opinions, there is nothing more rewarding than a project that you are excited to have the final product of.

The end result of this project was a fun product that we are all proud of. We hope that it will reflect positively on us as part of our legacy at UCF as we move on to our engineering careers.

10.3 Acknowledgements

We would like to thank Smart Charging Technologies for providing our group with four free battery packs, as well as allowing our group to use their lab for development work. We would also like to thank our faculty advisor, Dr. Samuel Richie, and our faculty panel – Dr. Reza Abdolvand, Dr. Qun Zhou Sun, and Dr. Zhishan Guo – for taking their time to review our project.

Appendices

Appendix A: References

[1] FPPT, "Free House of Quality Template for PowerPoint (QFD template)," 2 October 2020. [Online]. Available: https://www.free-power-point-templates.com/articles/free-house-of-quality-template-for-powerpoint-qfd-template. [Accessed 25 September 2021].

[2] theWOWEffect.com, "Laser Tag Revealed," [Online]. Available: https://thewoweffect.com/PDF/LaserTagRevealed.pdf. [Accessed 9 October 2021].

[3] Backyard Brains, "Experiment: How Fast Your Brain Reacts To Stimuli," [Online]. Available: https://backyardbrains.com/experiments/reactiontime. [Accessed 9 October 2021].

[4] Laser Safety Facts, "Laser classification table - Laser Safety Facts," Laser Safety Facts, [Online]. Available: https://lasersafetyfacts.com/laserclasses.html. [Accessed 9 October 2021].

[5] LaserHIT, "LaserHIT - Modern Firearm Training at Home," [Online]. Available: https://www.laserhit.com. [Accessed 9 October 2021].

[6] Squad Hero Store, "Amazon.com: Rechargeable Laser Tag Set + Innovative LCDs and Sync - 4 Infrared Guns & Vests - Gifts for Teens and Adults Boys & Girls - Outdoor Games - Cool Group Activity Family Fun - Gift for Kids Ages 8-12 +," [Online]. Available: https://www.amazon.com/Squad-Hero-Rechargeable-Laser-Innovative/dp/B082XJRTMG. [Accessed 9 October 2021].

[7] C. Dobias, T. Hughes, T. Hernandez and E. Plummer, "Laser Target Gallery," [Online]. Available: https://www.ece.ucf.edu/seniordesign/su2019fa2019/g01/. [Accessed 9 October 2019].

[8] K. Brinthaupt, S. Fies, M. Montero and A. Yamdagni, "Let's Have a Blast! Laser Tag," Senior Design Archives, Orlando, 2019.

[9] M.-G. Jeong, S.-H. Kim and C. Yoo, "Switching Battery Charger Integrated Circuit for Mobile Devices in a 130-nm BCDMOS Process," IEEE, 2016.

[10] Texas Instruments, "Digi-Key Electronics," February 2017. [Online]. Available: https://www.ti.com/lit/ds/slvs059t/slvs059t.pdf. [Accessed 4 22 2022].

[11] Texas Instruments, "Working with Boost Converters," June 2015. [Online]. Available: https://www.ti.com/lit/an/snva731/snva731.pdf?ts=1636050891599. [Accessed 4th October 2021].

[12] S. D. Senturia, "Microsystem Design," Boston, MA: Kluwer Academic Publishers, 2001, p. 366.

[13] R. S. Muller, T. I. Kamins and M. Chan, Device Electronics for Integrated Circuits, 3rd ed., 2002, pp. 429-430.

[14] D. Halliday, Fundamentals of Physics, 4th ed., 1993, pp. 1224-1225.

[15] Chaniotakis and Cory, "Bipolar Junction Transistors (BJT)," Massachusetts, Massachusetts Insititute of Technology, 2006, p. 10.

[16] Samsung Electronics, "Laser Diode," Suwon-Si, 2003.

[17] Pro-Lite Technology, "Laser Saftey - Laser Basics," 2021. [Online]. Available: https://www.pro-lite.co.uk/File/laser_safety_laser_basics.php. [Accessed 29th October 2021].

[18] Newark, "Light Sensors," [Online]. Available: https://www.newark.com/sensor-optical-light-sensor-technology. [Accessed 4 November 2021].

[19] EE Power, "Photoresistor," EE Power, [Online]. Available: https://eepower.com/resistor-guide/resistor-types/photo-resistor/#. [Accessed 5 November 2021].

[20] Wavelength Electronics, "Photodiode Basics," [Online]. Available: https://www.teamwavelength.com/photodiode-basics/. [Accessed 6 November 2021].

[21] OSI Optoelectronics, "Photodiode Characteristics and Applications," [Online]. Available: http://www.osioptoelectronics.com/application-notes/an-photodiode-parameters-characteristics.pdf.

[22] Electronics Hub, "Basics of Phototransistor," 23 March 2019. [Online]. Available: https://www.electronicshub.org/basics-of-phototransistor/. [Accessed 6 November 2021].

[23] Digi-Key, "BGN5500 Rechargeable Sealed Nickel Cadmium Battery Pack," Digi-Key Electronics, [Online]. Available: https://batteryguy.com/files/technical-docs/tech-spec-sheets/bgn5500-data-sheet.pdf. [Accessed 18th November 2021].

[24] Dantona Industries, Inc., "L74A52-4-3-2WX," Dantona Industries Inc. , 2021. [Online].

[25] World-semi, "Adafruit," 2021. [Online]. Available: http://www.world-semi.com/technology/details-129-3.html | https://cdn-shop.adafruit.com/datasheets/WS2811.pdf. [Accessed 6th December 2021].

[26] Adafruit , "Adafruit NeoPixel #1376," 2021. [Online]. Available: https://www.adafruit.com/product/1376?length=1. [Accessed 6th December 2021].

[27] A. Cunningham, "Understanding M.2, the interface that will speed up your next SSD," ars technica, 8 February 2015. [Online]. Available: https://arstechnica.com/gadgets/2015/02/understanding-m-2-the-interface-that-will-speed-up-your-next-ssd/. [Accessed 10 11 2021].

[28] S. Santos, "ESP32 vs ESP8266 - Pros and Cons - Maker Advisor," Maker Advisor, 6 September 2021. [Online]. Available: https://makeradvisor.com/esp32-vs-esp8266/. [Accessed 11 November 2021].

[29] BlackEdder, "Home - Wiki -painlessMesh / painlessMesh - GitLab," painlessMesh, 24 June 2019. [Online]. Available: https://gitlab.com/painlessMesh/painlessMesh/-/wikis/home. [Accessed 12 November 2021].

[30] AVNET, "KDT00030ATR Phototransistor Chip 630nm 2-Pin PLCC T/R," 2021. [Online]. Available: https://www.avnet.com/shop/us/products/onsemi/kdt00030atr-3074457345625166830/. [Accessed 5 November 2021].

[31] Digi-Key, "Excelitas Technologies C30737LH-230-83A," Digi-Key Electronics, 2021. [Online]. Available: https://www.digikey.com/en/products/detail/excelitas-technologies/C30737LH-230-83A/5885823. [Accessed 5th November 2021].

[32] Excelitas Technologies, "C30737LH-230-80N - Si APD, 230um, LCC, 800nm enhanced," October 2017. [Online]. Available: https://www.excelitas.com/product/c30737lh-230-80n-si-apd-230um-lcc-800nm-enhanced. [Accessed 5th November 2021].

[33] Digi-Key Electronics, "Advanced Photonix PDB-C156," 2021. [Online]. Available: https://www.digikey.com/en/products/detail/advanced-photonix/PDB-C156/12675494. [Accessed 5th November 2021].

[34] Digi-Key, "Advanced Photonix PDB-C156," 11th July 2019. [Online]. Available: http://www.advancedphotonix.com/. [Accessed 5th November 2021].

[35] Digi-Key Electronics, "Advanced Photonix PDB-C152SM," 2021. [Online]. Available: https://www.digikey.com/en/products/detail/advanced-photonix/PDB-C152SM/480583. [Accessed 5th November 2021].

[36] Digi-Key Electronics, "Advanced Photonix PDB-C142," 2021. [Online]. Available: https://www.digikey.com/en/products/detail/advanced-photonix/PDB-C142/480581. [Accessed 5th November 2021].

[37] Digi-Key Electronics, "Photonic Photodetectors PDB-C142," 4th January 2016. [Online]. Available: http://www.advancedphotonix.com/. [Accessed 5th November 2021].

[38] Digi-Key Electronics, "Everlight Electronics Co Ltd ALS-PT19-315C/L177/TR8," 2021. [Online]. Available: https://www.digikey.com/en/products/detail/everlight-electronics-co-ltd/ALS-PT19-315C-L177-TR8/2675735. [Accessed 5th November 2021].

[39] Everlight Americas, "Ambient Light Sensor Surface - Mount ALS-PT19-315C/L177/TR8," 24th December 2013. [Online]. Available: https://everlightamericas.com/ambient-light-sensors/7/EAALSTIC1708A0.html?search_query=ALS-PT19-315C&results=1. [Accessed 5th November 2021].

[40] Everlight Americas, "3mm DIP Ambient Light Sensor ALS-PT204-6C/L177," 31 December 2007. [Online]. Available: https://everlightamericas.com/ambient-light-sensors/18/eaalst03rdaa0.html?search_query=ALS-PT204+&results=1. [Accessed 5th November 2021].

[41] Digi-Key, "Panasonic - BSG HHR-380AB27F2," Digi-Key, November 2021. [Online]. Available: https://www.digikey.com/en/products/detail/panasonic-bsg/HHR-380AB27F2/271985. [Accessed 26th November 2021].

[42] Digi-Key Electronics, "Texas Instruments MC34063ADR," January 2015. [Online]. Available: https://www.digikey.com/en/products/detail/texas-instruments/MC34063ADR/717456. [Accessed 28th November 2021].

[43] Digi-Key Electronics, "Rohm Semiconductor BD9227F-E2," 31 August 2017. [Online]. Available: https://www.digikey.com/en/products/detail/rohm-semiconductor/BD9227F-E2/7802693. [Accessed 2nd December 2021].

[44] Digital View, "Buttons or Touchscreen?," 8 November 2009. [Online]. Available: https://www.digitalview.com/blog/buttons-or-touchscreen/. [Accessed 12 November 2021].

[45] Diffen LLC, "LCD TV vs OLED TV," [Online]. Available: https://www.diffen.com/difference/LCD_TV_vs_OLED_TV. [Accessed 11 November 2021].

[46] C. Freudenrich Ph.D, "How OLEDs Work," HowStuffWorks.com, 24 March 2005. [Online]. Available: https://electronics.howstuffworks.com/oled5.htm. [Accessed 11 November 2021].

[47] Amazon, "MakerHawk 2PCS 4 Ohm 3 Watt Speaker," [Online]. Available: https://www.amazon.com/MakerHawk-Full-Range-Portable-Advertising-JST-PH2-0mm-2/dp/B08JCHK7GR?th=1. [Accessed 13 November 2021].

[48] Law Insider, "Technical Standard Definition," [Online]. Available: https://www.lawinsider.com/dictionary/technical-standard. [Accessed 28 October 2021].

[49] IEEE Standards Association (IEEE-SA) Standards Board, "IEEE Guide for Developing System Requirements Specifications," The Institute of Electrical and Electronics Engineers, Inc., New York, 1998.

[50] United States Food and Drug Administration, "Laser Products and Instruments," 26 March 2021. [Online]. Available: https://www.fda.gov/radiation-emitting-products/home-business-and-entertainment-products/laser-products-and-instruments. [Accessed 31 October 2021].

[51] Laser Safety Facts, "Class 3R (IIIa) laser safety information," Laser Safety Facts, [Online]. Available: https://www.lasersafetyfacts.com/3R/. [Accessed 1 November 2021].

[52] SignalBoosters.com, "What are IEEE 802.11 Standards?: 802.11a/b/g/n/ac/ax," 29 October 2020. [Online]. Available: https://www.signalboosters.com/blog/ieee-802.11-standards-explained-802.11abgnacax/. [Accessed 28 October 2021].

[53] Millennium Circuits Limited, "Guide to IPC Standards for PCBs," [Online]. Available: https://www.mclpcb.com/blog/ipc-standards-for-pcbs/. [Accessed 30 October 2021].

[54] IPC, "Generic Standard on Printed Board Design," IPC, Northbrook, 2003.

[55] Rasika, "IPC J-STD-001 Standard Soldering Requirements," Sierra Circuits, 23 September 2021. [Online]. Available: https://www.protoexpress.com/blog/ipc-j-std-001-standard-soldering-requirements/. [Accessed 31 October 2021].

[56] Huntron, "Comparing lead vs. lead-free solder for PCBs," Huntron Inc., 31 July 2020. [Online]. Available: https://huntron.com/news/comparing-lead-vs-lead-free-solder/. [Accessed 31 October 2021].

[57] Space Electronic Assemblies J-STD-001 Addendum Task Group (5-22as) of the Assembly & Joining Processes Committee (5-20) of IPC, "Space Applications Electronic Hardware Addendum to IPC J-STD-001E Requirements for Soldered Electrical and Electronic Assemblies," Bannockburn, 2010.

[58] vartika02, "Difference between C and C++," GeeksforGeeks, 15 February 2021. [Online]. Available: https://www.geeksforgeeks.org/difference-between-c-and-c/. [Accessed 24 November 2021].

[59] ISO, "ISO/IEC 14882:2020," December 2020. [Online]. Available: https://www.iso.org/standard/79358.html. [Accessed 24 November 2021].

[60] L. Doan-Minh, "C++ Programming Guidelines," Rational Software Corporation, 1997. [Online]. Available: https://sceweb.uhcl.edu. [Accessed 24 November 2021].

[61] "RoHS Guide," 2021. [Online]. Available: https://www.rohsguide.com/. [Accessed 22 October 2021].

[62] University of Central Florida, "UCF Policies and Procedures - 3-119.2 Weapons on University Property and at University Events," 24 July 2019. [Online]. Available: https://policies.ucf.edu/documents/3-119.pdf. [Accessed 22 October 2021].

[63] S. Franzyshen, "painlessMesh," 2021. [Online]. Available: https://gitlab.com/painlessMesh/painlessMesh. [Accessed 9 November 2021].

[64] Adafruit, "Adafruit_SSD1306," 2021. [Online]. Available: https://github.com/adafruit/Adafruit_SSD1306. [Accessed 9 November 2021].

[65] Adafruit, "Adafruit_GFX," 2021. [Online]. Available: https://github.com/adafruit/Adafruit-GFX-Library. [Accessed 9 November 2021].

[66] Autodesk, "PCB layout software for every engineer: Features," 2021. [Online]. Available: https://www.autodesk.com/products/eagle/features#pcb-community. [Accessed 19 November 2021].

[67] Software Radius, "EasyEDA vs Eagle - What's Best for PCB Designing," [Online]. Available: https://www.softwareradius.com/easyeda-vs-eagle-pcb-designer/. [Accessed 21 November 2021].

[68] GitHub, Inc, "GitHub," 2021. [Online]. Available: https://github.com/. [Accessed 9 November 2021].

[69] Microsoft, "Visual Studio," 2021. [Online]. Available: https://visualstudio.microsoft.com/. [Accessed 9 November 2021].

[70] PlatformIO Labs, "PlatformIO," 2021. [Online]. Available: https://platformio.org/. [Accessed 9 November 2021].

[71] M. U. Khalid, "Fundamentals of BMS selection," Electronics360, 30th August 2021. [Online]. Available:

https://electronics360.globalspec.com/article/17106/fundamentals-of-bms-selection#. [Accessed 12th October 2021].

[72] Texas Instruments, "SNVA558," 2011. [Online]. Available: https://www.ti.com/lit/an/snva558/snva558.pdf?ts=1636125822261. [Accessed 21st October 2021].

[73] Adafruit Industries, "ICR18650 4400mAh 3.7V Lithium Ion," 18th August 2014. [Online]. Available: https://www.adafruit.com/product/354. [Accessed 26th November 2021].

[74] Digi-Key, "Texas Instruments LM2736XMK/NOPB," December 2014. [Online]. Available: https://www.digikey.com/en/products/detail/texas-instruments/LM2736XMK-NOPB/723203. [Accessed 28th November 2021].

[75] Digi-Key Electronics, "Texas Instruments LM2576SX-3.3/NOPB," May 2021. [Online]. Available: https://www.digikey.com/en/products/detail/texas-instruments/LM2576SX-3-3-NOPB/366792. [Accessed 28th November 2021].

[76] Digi-Key, "Digi-Key Electronics," January 2015. [Online]. Available: https://www.digikey.com/en/products/detail/texas-instruments/MC34063ADR/717431. [Accessed 28th November 2021].

[77] Espressif Systems (Shanghai) Co., Ltd, "Unit Testing in ESP32," 2021. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/unit-tests.html. [Accessed 5 December 2021].

Appendix B: Copyright Permissions

- Request for Permission and Permission Granted from Laser Safety Facts:

Requesting Permission to Use Figure

RG Rachel Goodman
Fri 10/22/2021 1:28 PM
To: mail@lasersafetyfacts.com

Hello,

I am an electrical engineering senior at the University of Central Florida. I am using lasers in my senior design project, and I am writing to request your permission to use the "Eye Injury Hazard" figure (from the page: Laser classification table - Laser Safety Facts) with proper credits to your site in my final paper for the project.

Thank you very much,

Rachel Goodman
r_goodman99@knights.ucf.edu

Reply | Forward

Requesting Permission to Use Figure

PM Patrick Murphy <mail@lasersafetyfacts.com>
Fri 10/22/2021 4:08 PM
To: Rachel Goodman

Rachel —

Thank you for asking. Yes, you can use the figure, with credit to the site.

If anybody should give you any trouble about using lasers ("Oh, it's unsafe!") feel free to contact me for info.

Good luck with your project!

— Patrick Murphy, site editor,
Orlando, FL

- No Copyright for FDA Website Materials:
  URL: https://www.fda.gov/about-fda/about-website/website-policies

Linking to or Copying Information on the FDA Website

Unless otherwise noted, the contents of the FDA website (www.fda.gov) — both text and graphics — are not copyrighted. They are in the public domain and may be republished, reprinted and otherwise used freely by anyone without the need to obtain permission from FDA. Credit to the U.S. Food and Drug Administration as the source is appreciated but not required.

- Texas Instruments Terms of Use for Educational Purposes:
  URL: https://www.ti.com/legal/terms-conditions/terms-of-use/north-america.html

TI further grants to K-12 educational institutions, universities, and community colleges permission to download, reproduce, display, and distribute TI Services solely for use in the classroom, provided that such institutions identify TI as the source of TI Services and include the following credit line: "Courtesy of Texas Instruments." Unauthorized use of any TI Service is expressly prohibited by law, and may result in civil and criminal penalties. This grant of permission terminates if you breach any provision in these Terms of Use or Service Terms. Upon termination, you agree to destroy any materials relating to TI Services.

- Request for Permission and Permission Granted from Pro-Lite Technology:

RG Rachel Goodman
Sat 11/6/2021 11:55 PM
To: info@pro-lite.co.uk

Hello,

I am an electrical engineering senior at the University of Central Florida. I am using lasers in my senior design project, and I am writing to request your permission to use the figure in the "Laser Radiation and the Electromagnetic Spectrum" section of the following webpage: Laser Basics (pro-lite.co.uk) with proper credits to your site in my final paper for the project.

Laser Basics - Pro-Lite Technology

Laser Safety - Laser Basics. Laser Radiation and the Electromagnetic Spectrum. Electromagnetic radiation is a natural phenomenon found in almost all areas of daily life, from radio waves to sunlight to x-rays.

www.pro-lite.co.uk

Thank you very much,

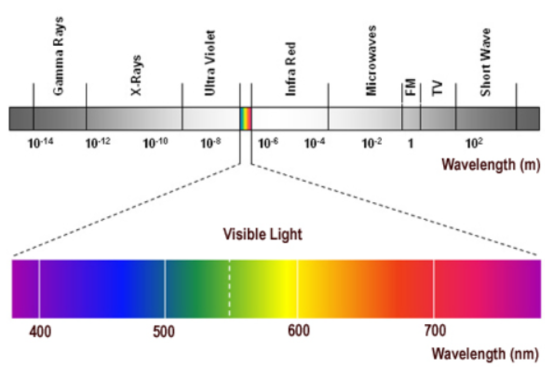
Rachel Goodman
r_goodman99@knights.ucf.edu

RE: Requesting Permission to Use Figure

J jagoda.zajac@pro-lite.co.uk
Tue 11/9/2021 2:24 AM
To: Rachel Goodman

Good Morning Rachel,

Thank you for contacting us. Yes, you can use the picture below.



Kind regards

Jagoda Zając
Pro-Lite Technology Ltd.
Innovation Centre, University Way, Cranfield, MK43 0BT
Tel: +44 (0) 1234 436110 Fax: +44 (0) 1234 436111
www.pro-lite.co.uk

Appendix C: Sponsorship

Gmail

Jamauri Balzourt <jbalzourt@smartchargetech.com>

SCT Sponsorship official email request

Yurany Lopez <ylopez@smartchargetech.com>
To: Jamauri Balzourt <jbalzourt@smartchargetech.com>

Wed, Dec 8, 2021 at 3:23 PM

Hello Jamauri,

Smart Charging Technologies LLC would be happy to sponsor you and your senior design team for your Senior Design Project.

We will be providing the battery packs and our lab space for your research.

Best of luck.

Kind Regards,

Yurany Lopez
Operations Manager

Smart Charging Technologies LLC

(407) 529-8143 | ylopez@smartchargetech.com

smartchargetech.com

[Quoted text hidden]