Laser Target Gallery

Senior Design 2 Documentation

Fall 2019

Group 1
Caleb Dobias - Photonics Science and Engineering
Travis Hughes - Computer Engineering
Triston Hernandez - Computer Engineering
Edward Plummer - Computer Engineering

Contents

1.0 Executive Summary

The Laser Target Gallery is inspired by the arcade games routinely seen at carnivals and festivals, where one uses their aim to try and hit a target using water guns, dart guns, or lasers. In most of these games, however, the use of accuracy is measured in a binary way, where either the target is hit or not. In this game, however, you score points where the more centered you hit the target, the more points you earn. This way, it encourages the player to reward both patience to hit each target squarely and the speed to hit targets fast, causing a sort of balancing act to try and achieve a high score. These high scores are updated on a phone application which automatically keep sand ranks the top 10 scores of the game.

The project is divided up into four parts, the laser gun, the target board, and the wireless microcontroller, and phone application. The laser gun subsystem is be a standalone piece that runs separate from the other parts. Using a power supply, microcontroller, and laser diode it fires a visible beam of light at the target board when the trigger is pulled. The microcontroller determines the time the beam is active as well as how often the trigger can be pulled. Inside the gun itself, a lens and aperture system is used to focus the beam to a small radius that can completely cover any target at the board.

The laser gun contains a class 3 laser a small microcontroller, and a lens and aperture system to create a controlled beam shape. The casing for the laser gun is 3D printed after creating the design in CAD software. Inside the laser gun's detachable handle, we have the power supply which is connected to an ATmega328 microcontroller, which in turn power the laser. The microcontroller is part of the pacing of the game, controlling the duration and fire rate of the shots fired by the gun.

The target board portion is simply a board comprised of a wooden board with four holes to shoot into, made of a 3D printed plastic casing. At the back of each of these plastic casings are a set of four photoresistors, which when put in series with a traditional resistor, changes voltage such that the microcontroller can read. The response is dependent on how much incoming light the photoresistor receives, as the photoresistor decreases resistance with light. By comparing the relative strength of each voltage signal, the accuracy of the shot can be judged and scored appropriately by a Bluetooth microcontroller attached to the board. Also attached to the board is a set of LED lights surrounding each target. These LEDs light up, indicating which target is shot at.

There are four printed circuit boards, each connected to the same microcontroller. As the photodiodes sends the voltage signals to the microcontroller, we are able to determine the position of the shot and score it accordingly. This data is needed to be communicated to mobile application we are developed. To do so, we looked at setting this data transfer up using either

Bluetooth or Wi-Fi, settling on Bluetooth. We attached this module to the microcontroller that is receiving the data from the four printed circuit boards, once the data is received from the printed circuit boards, it is relayed to the application.

The web application is built to display the users score after finishing their round of shooting. This data is received from the Bluetooth enabled microcontroller. Once the data is received, the application is displayed the user's accuracy for the round as well as their score. On the home screen we display the top 10 scores of all user's who have ever played the game, giving the user the ability to see how their score matches up. The mobile application features offline capabilities such that users can take the game board on the go. Each game session is stored locally on the mobile device. Users can check out previously recorded game sessions and track how they improve over time when interacting with our project.

The mobile application provides a native solution for multiple mobile devices in order to provide a sleek and natural feel for users. Performance, user experience, and security is a major focus for the final version of the mobile application. The application can update data tables both quickly and reliably to ensure accurate representation while reducing user wait times. Furthermore, the safety of our user's data is highly important and under no circumstances should it have the potential to be compromised. To do so, user credentials are protected using various encryption standards provided by modern day mobile applications.

Our end goal for this project is to provide our potential consumers with an enjoyable lighthearted entertainment system utilizing modern day technology. Engaging, portable, reliable, sleek, and simple: that is our laser target gallery.

2.0 Project Description

This next section will describe higher level details on each of the components of the project, including the motivations for the designs. These descriptions will outline a series of requirements that must be met for the project to be functional and a list of requirements that are planned to be included, but not necessary for the project to be considered working.

2.1 Project Motivation and Goals

When most people think of carnivals, it is almost inseparable from the ideal of the games played for prizes. Many of these games are skill based such as Whack-A-Mole or a ring-toss variant, but others have been inauthentic as an arcade experience, being scams or luck based. While the idea of a carnival game is generally enticing and can be quite fun, it on occasion can leave a bad taste in the mouth of the players from factors outside of their control. As such, one of the inspirations for the project came from the enchantment of the carnival game but removing the frustration of attaining a deceptively difficult standard by making the carnival game competitive. This inspiration of a completely fair and approachable skill-based game as a way to satiate the craving of carnival fun without the dishonesty is what drives this project.

Initially, the project was spun as game board, akin to a chess board, where you could interact with all the pieces via a pressure to play games such as Snake or other simple block-based games. This was done through an array of photodiodes, which would eventually turn into what we used as targets in the game. The board game idea, however, was quickly dropped as it did not integrate the novelty we were searching for. A few more game variants were tossed around, with varying levels of appeal to each group member. Finally, using the inspiration to emulate a carnival game, we can to the conclusion a quickly recognizable game like target shooting would be one of the most engaging pieces we could come up with. The main focus that we could all agree that we were passionate about was making a skill-based game that was immediately identifiable, easy to play and understand, and attention grabbing.

After these foundations were fleshed out, we moved onto the specifics of how to make this game stand out. The focus of many shooting games relied on accuracy or reaction time, and quite often both. However, unlike the disruptive nature of the sound of a real firearm, a laser gun offers the benefit of being far less loud, while maintaining a similar appeal as a game device. Using the laser gun to hit targets after they light up was the initial idea and seemed to carry well. Additionally, the flashing lights of both the gun and the targets seemed pretty fun to work with. However, it didn't seem like quite interesting enough by itself to compare to other projects of similar nature. Instead, we wanted to gauge the accuracy of each shot as well. After a few hours of deliberating, we concluded that we could, in fact, use a measure of accuracy on each target using a set of four photoresistors in a square arrangement using some clever voltage reading tricks. When a photoresistor and a resistor is placed in series, we can measure the voltage over the resistor. When light hits the photoresistor, the voltage goes up across the resistor due to the photoresistors drops in voltage.

The foundations of the project completed; the next step was to determine how our focus of entertainment should be applied to the project. The idea to make the game multiplayer was brought up, with a few struggles on how to implement it. This led to the design of tying the game to a phone application, which are connected to the internet, keeping a high score board of the game itself. However, not all changes were additions to the project. A common laser tag gun uses infrared light, which is invisible to the eye. However, because the project uses accuracy as an integral part of the design, we wanted to give feedback on where the shot actually landed. This led to the decision to use a red laser diode as well as implement a feature of the application where the shot would be recorded on a graph on where the hit landed. This later changed to a green laser diode for better beam shape management and photoresistor responsiveness.

This decision for feedback on playing the game was initially inspired by entertainment but had a secondary application as well. Obviously, hand eye coordination would be a part of the game, but by using the information collected by the targets, we can determine where inaccuracies tend to land on each target and thusly improve it. For example, if someone tends to shoot up and to the left of any given target, that can be shown on the graph and corrected for by the player in the next attempt. By being clear to the player what the issues are with their accuracy and how to fix them, we simultaneously give them an achievable and attainable goal, which is encouraging, but also makes them feel like they are accomplishing something by getting better accuracy.

Finally, the project was then divided up into four separate functional parts: The laser gun, the board, the wireless microcontroller used to run the game, and the phone application. The goal of the laser gun is to emit controlled bursts of light that are powerful enough to be registered by each target on the board, where the light maintains a reasonably uniform distribution of light. The laser pulse must also last long enough for the photoresistors to change their values, but since this change is in the order of milliseconds, this is unlikely to be an issue. The goal of the board is to house targets which can effectively communicate which target to hit as well as what receive adequate responses from the laser gun itself. The goal of the microcontroller is to capture data from the target board and deliver information to the phone app. The goal of the phone app is to analyze and record data, as well as update the high score board. These functions combined allowed us to divide up work into between the four of us.

2.2 Project Requirements

The following sections break down into two main parts, functional requirements and non-functional requirements. Each of the two parts further breaks down into specific pieces of the project, which are separated by physical aspects or in the case of the Web App, is separate as it functions on a smartphone layer, not a physical device included in the project itself.

2.2.1 Functional Requirements

This section outlines all of the features each component of the project must have in order to be considered a complete and functional product. If any of these were to fail, then the project itself would not work. In order to get a functioning laser based sharp shooting game we need a handful of components that are all connected. One of the obvious components being the Laser Gun, the gun is going to made of plastic, 3D printed (in several parts), with a hollow inside leaving space for a microcontroller, along with other electrical devices(Laser, speaker, motion detector, LEDs, etc). The next part of the process of the game involves a game board made up of targets, each target is made up of cardboard, LEDs, and printed circuit boards. Each target is independently connected to another microcontroller, this microcontroller is going to do a bulk of the work for the game, such as accuracy computation, hit detection, and sends information to web application. The web application supports a user database with records of highest game score , along with other game analytics.

2.2.1.1 The Laser Gun

The laser gun must project an eye safe beam of light outwards in a circular or square beam shape. It matters a lot less the color, means of generation, or the coherence level of the light. However, its intensity profile must be reasonably uniform or symmetrically distributed. The design of the detection system assumes that the incoming beam is symmetrical across at least the x and y axis of the target. This is because a perfectly centered hit should cause all of the photoresistors to respond with the same current, and any offset should cause a displacement in signals proportional to the offset of the beam. A powerful signal to one photodiode but not the others indicates the direction of offset is towards that one photodiode.

The laser gun should also function responsively when the trigger is pulled. This is done by connecting a power supply and switch to a microcontroller, the switch being activated by the trigger. After the switch is activated, the microcontroller determines if and what kind of pulse to send to the lighting system, sending the beam to the board itself. This means that at minimum, the microcontroller should be powerful enough to supply the necessary voltage and current to the photodiode and have enough input to watch a trigger then apply this voltage for the whole laser pulse duration.

Mechanically, the gun itself needs to hold the power supply, the microcontroller, light source, and any lens internally and securely. Any motion such as quickly swinging or acceleration should not damage or disturb the system, as that is an unavoidable mechanic of the game. These requirements can be kept by use of adjustable and sturdy mechanical construction such as a 3D printed part. These parts should not cause any discomfort via heat or protrusions that would otherwise cause harm to the user.

2.2.1.2 The Target Board

The board must hold itself upright and be large enough to hold four targets. Each of these targets must contain four photoresistors connected to the wireless microcontroller independently where the photoresistors are connected to another resistor in series. This is done optimally with a printed circuit board, however, this is not necessary for a functioning product. These photoresistors must be arranged in an evenly spaced square shape, such that the when hit with light, and communicate analogue signals to the wireless microcontroller. This is because we need to be able to rely on the displacement of where the beams hit. This of course means that there must be several wires strung along the back of the board of equal resistance for each target. The targets themselves must be shielded from light to prevent ambient light from causing noise or distorting the signal received by the photoresistors themselves.

The sharpshooter game board has targets that look similar to the Diagram 2.2.1.2.1 with five total targets. For each target there is a printed circuit board attached to the back, through a hole in the back of the target, at the back edge of the wall towards the printed control board is be an array of four photo resistors connected in series to the individual printed control board that is attached to the back wall of the target with a small gap in between to allow space for other components on the circuit board. To get a better understanding of how this looks, the target component itself looks like Diagram 2.2.1.2.1.

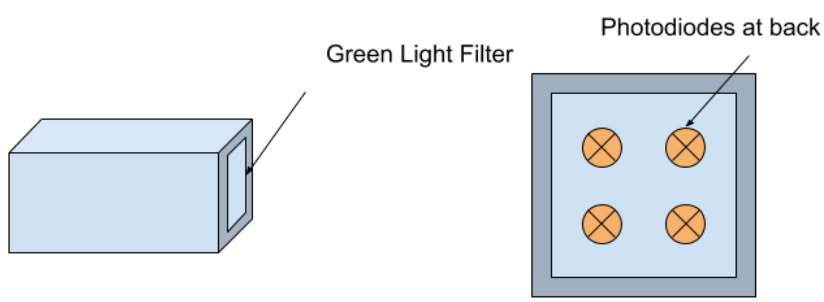


Diagram 2.2.1.2.1: Side and Profile View of Target

2.2.1.3 The Wireless Microcontroller

There is going to be one microcontroller attached to the board of the game that interacts with other devices via Bluetooth. For this element of the project we choose to use an Arduino: ATmega328P as our development board, there are a ton of other similar devices, our next choice was the Basic Stamp 2, the two options have similar and different contributions, so we had some decisions to make. Picking a microcontroller is a big step, especially for the sake of our game, and one that deserves a bit of thought, and the comparison can be seen in Table

2.2.2.3.1. Price is a driving factor, along with power consumption and other comparable trade offs amongst the devices.

Table 2.2.2.3.1: Microcontroller Pro and Con Comparison

Microcontroller	Pros	Cons
Basic Stamp 2	• Extensive documentation • Broad range of use • Low power consumption	• Limited memory • Expensive • Slow
Arduino	• Cheap • Small • Simple • Has a software programming environment	• Slightly limited capabilities
MSP430	• Varying capabilities • Cheap • On board LEDs	• Complicated software • Limited documentation

The main functionality of the microcontroller in the handle of the gun is receiving output from the trigger, and sending output to the laser. There is a script associated with the trigger that is used to count the number of times the trigger is pulled and the number of times the laser is activated, this information can be used for our accuracy calculations that can be sent to the web application for readable analytics.

2.2.1.4 The Web App

The web application is responsible for the conversion of raw data into comprehensive statistics that can be viewed by the user. It requires an intuitive frontend and a well-planned backend to build a successful web application. As a result, the application must be able to communicate with the microcontroller and it is imperative that the data is received and parsed correctly such that accuracy is not compromised in the process. In addition, the web application needs to be able to handle concurrent users.

When a user first opens the application, they are prompted to create their account. Once completed the user is directed to their dashboard. The application then detects if there is a game board nearby which it is capable of pairing with. If so then there is a button which allows the user to start a game session. After a completed session, the results are stored in the database and can later be compared to additional sessions. Each user has their own unique dashboard which contains their most recent recorded game records. Users can then view these records to find out specific details about their personal performance.

Accuracy, precision, and the position of each shot is determined by the results picked up from the photoresistors located on the target board, the score being passed onto the phone application.

2.2.1.5 Power Supply

As opposed to voltage, which is applied over a circuit part, current is the draw a circuit is actively taking from a power source, meaning the power supply, supplies different amperage depending on the load placed on it by the connected device. Therefore, the greater the current output of the adaptor should be enough for both the arduino and the laser. The minimum requirement for the arduino device is 250mA which should run with some measure of stability. However, since for the sake of our project we want to power other devices (LED's, Laser Diode, Speaker, etc.) then 0.5A to 2A enables us with the range of use we need. Useful Current Limits(Arduino):

- The USB port has a polyfuse shut-off of 500mA. Any connected devices that draw more than this causes instability. This may be a concern with the Arduino Mega used for the board, but a high photoresistor and resistor value should keep the current low.

- The absolute maximum current draw for a single digital or analogue I/O pin is 40mA (<35mA recommended max), with a total maximum current draw from all I/O pins of this type being 200mA combined.

- The 3.3V pin has a maximum current output of 150mA (recommended at 50mA).

- 5V pin has a maximum draw of 0.8A. It should be noted that the 3.3V and 5V circuits are combined, so 0.8A is also the combined maximum current draw of both these pins. It should also be noted that 0.8A is the theoretical maximum determined by the on board voltage regulators. The harder these voltage regulators are working, the less current you'll be able to draw, so a more realistic maximum figure is 0.5A. However, no part of the project is using both of these pins at the same time.

- The variable pin (Vin) bypasses the majority of the Adruino's circuitry, so there is no real maximum except that set by the diode that separates Vin from the other circuitry on the board. The diode is rated 1A, and the board traces are rated at 2A, therefore the theoretical maximum for Vin is 1A. We've seen reports of running devices higher than this, changing the diode, or even bypassing it completely, but it's not recommended.

2.2.2 Non-Functional Requirements

This section outlines all of the features of each component of the project does not need to have in order to be considered functional. These requirements instead could be considered instead to support the aesthetic, quality of life, or conveyance of information of the project.

2.2.2.1 The Laser Gun

The gun should reasonably reflect the idea that it is a gun, without indicating that it is a serious actual weapon. This can be accomplished by rounded edges, colors, or comical bloating of parts. There have been several incidents on campus when an object, thought to be a dangerous firearm, causes a disturbance and the police are called to investigate. Obviously, we would like to avoid this. The beam itself should have a wavelength within visible range, such that when the beam is shot, there is a vibrant indication of where the beam landed. This is most easily accomplished with red or green laser diodes. Haptic feedback may also function as an indicator that a beam has been shot through a vibrating motor near the handle or a hammer type trigger, so the user has a sense that the gun has been fired when a trigger is pulled. Additionally, the gun casing should have a weight to it that would replicate a similar weight to a traditional pistol, if only to provide a more comfortable feeling to the user.

The power supply be connected directly to the microcontroller, which controls the power the laser. The reason the power is directed through the microcontroller is to put limits on how often the laser can be fired and how long a laser pulse lasts. This way, reckless pulling of the trigger in the arcade would be punished with missed opportunities to score points. The laser diode itself may diffract larger or smaller than what we would like, and as such, we would use a beam collimating system to allow the gun to be used at a multitude of ranges. Additionally, an aperture should be used to control the ringing and size of the output beam shape.

The primary material for the silhouette is plastic, printed on a 3D printer, in two halves with the interior hollowed out. The hollow shell of the gun allots us room to place our electronic components inside, with the basic outline of the gun shown in Diagram 2.2.2.1.1.

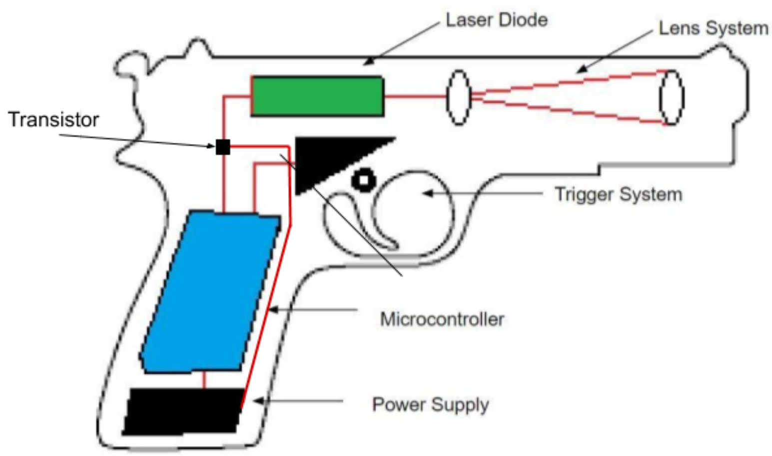


Diagram 2.2.2.1.1: Approximate Locations and Parts of Laser Gun

The shape of the gun in the picture isn't a 1:1 scale with the real model, however we are using this for the purpose of showing the location of the electric devices. In actuality, the microcontroller is above the trigger and the barrel is seperate. Here we are using the blue color block to represent the microcontroller, depending on which microcontroller chosen for this project, the size of the handle of the gun is adjusted. The microcontroller is linked directly to two devices the laser module's transistor and the trigger; the laser is marked in green on the diagram and emits through a lens at the aperture in the front of the gun. It is important to note that the parts in the diagram are not to scale, and the lens system is more robustly explained later.

2.2.2.2 The Target Board

Adjacent to each target, there should be a LED array in a decorative square design, as can be seen in Diagram 2.2.2.2.1. These units will light up when the wireless microcontroller expects to receive a hit, then flash repeatedly indicate that a hit has been registered. The board's color should be distinct and reflective enough such that when a light from the laser gun hits the board, the optical difference between the board and marker acts as a hitmarker.

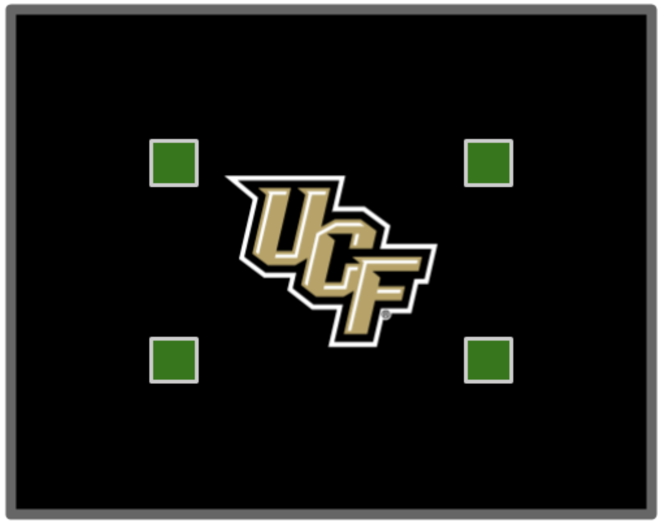


Diagram 2.2.2.2.1: Target Board Diagram Showcasing Decorative Color

2.2.2.3 The Wireless Microcontroller

Microcontrollers for all intents and purposes are going to be the brains of the game and project. Each microcontroller is going to have multiple jobs, for the wireless Arduino connected to the printed circuit boards on the back of the target display, the jobs for this computer in chronological order is:

1. Start a timer that is relayed on the time the user start their first laser target gallery round. This is default setup to one minute and can be adjusted to be longer or shorter amounts of time.
2. In that minute (or so) length of time the microcontroller is going to be focused on reading inputs from the five printed circuit boards. The Arduino computer uses counter methods to count up every time a photoresistor has been activated and stores that information in registers.
3. Once the timer is up the microcontrollers input pins no longer accept any inputs from the printed circuit boards and photoresistors. Data in the registers remain saved, and the lights stop working.
4. Data from the registers is wirelessly(Bluetooth) sent from the registers of the microcontroller to the database of the web application.
5. Once the microcontroller is notified that the application database successfully received the sent data from the registers. The data on the computer is reset and waits for commands from the user to reset the timer.

The microcontroller used inside of the laser gun obviously have different jobs. These jobs, again in chronological order are:

1. Start a timer, this timer is synchronized to that of the microcontroller attached to the target board, that way the game should start and end at the same time for both computers.
2. For the length of the timer the Arduino is going to be both receiving inputs and sending outputs, the inputs come from the trigger, and outputs are sent to the laser diode.
3. The last job is doing something similar to the other microcontroller, for game analytics we want a count of the number of times the trigger is pulled, before the timer ends we have a counter method storing this information in a register, along with that the microcontroller is putting restrictions on inputs from the trigger(trigger shot is only registered every half second for example).
4. When the timer ends the information saved is sent to the web application wirelessly, and the registers and timers are reset.

We needed another microcontroller on the back of the target board that allows for our data to be read from the photoresistors. This board meets a few different requirements to ensure our project works. We have 5 targets spread around the board, each of these targets each have four photoresistors attached to it. That means that we are going to need 20 input pins on our microcontroller, not including the power to be supplied to each photoresistor set. We also needed four pins set aside for our Bluetooth chip to attach to the microcontroller. This means we would need at least 25 pins on our Microcontroller. For this reason, we needed to use the Arduino Mega. The features of this board can be seen below in Table 2.2.2.3.1. The board comes with 54 input and output pins as well as 16 Analog input pins, which allows for us to attach all of the photoresistors to the board and read the data sent.

Once a target is shot at, we receive four different values back, one from each photoresistor. Each value returns a value between 0-1024, associating with different voltages, 0 through 5V. A value of 0 would indicate that the there’s no voltage read from the pin compared to ground. A value of 1024 indicates the pin has a voltage of 5V with reference to ground. If a shot hits the center of the target; theoretically, each of the photoresistors attached to that target would send the same value. We can assume a reasonable variance due to imperfections in the lens and beam shape, but this can be averaged out. The center of the target should be an equal distance away from each of the resistors. Once receiving these values, points are assigned. These points are what is then sent from the board to the application to be displayed for the user to see.

Table 2.2.2.3.1: Arduino Mega tech Specs

Microcontroller	ATmega2560
Operating Voltage	5 V
Input voltage	7-12 V
Input voltage (limit)	6-20 V
Digital I/O pins	54
Analog Input Pins	16
DC current per I/O pin	20 mA
DC current per 3.3 V pin	50 mA
Flash Memory	256 KB
Clock Speed	16 MHz

2.2.2.4 The Web App

The Web App receives data transmitted by the Bluetooth microcontroller and parse the data accordingly. Depending on the data received, a score is assessed based on the margin of error found relative to the distance from the center of the board. Under the allotted time frame given to the user, the score is updated in real time to track their progress. Once the run is complete, users are able to compare their scores with previous attempts in order to track their progress. Along with personal records we implemented a leaderboard that allows the user to compare their scores with that of a database of other users.

Other possible avenues of progression in our application can be the inclusion of a multiplayer mode. Users could compete to see who among them has the best aim.
The game sessions wouldn't be limited to just local interactions. Users could pair up across great distances by synchronizing their boards to the same game session instance. This could be expanded even further by hosting daily competitions where many individuals could challenge each other and compare scores.

The web application's task for this project is supposed to enhance the user's experience when interacting with the target board. Thus, it's important that the web application's design encourages users to want to spend more time on the application than off. Furthermore, being able to interact with different users can add a new layer of enjoyment to the Laser Target Gallery.

These are the proposed features which can accomplish this task.

1. Making a mobile application that is compatible for both Android and IOS using the native functions provided by each device platform. If designed properly, the codebase from one platform could be used as a direct guideline for the other.

2. Creating an intuitive design which gives users the ability to navigate the application without any guides. The application should prioritize a user's first-time experience. To do so, the application focuses on reducing unnecessary information that clutters the screen, distract users with interactive animations during loading sequences, and upholding a simplistic design. Next, the application promotes consistency in the design layout, minimize user input, show clear and concise communication, and follow the material design guidelines provided by both Google and Apple.

3. Implement a system which allows users to compare their scores to other users in the community. Users are able to keep track of how well their friends have played in their recent games along with how they match up with the top players.

4. Create a leaderboard which displays the top 10 game sessions of varying time frames. The leader board has the capability of being sorted based on the statistic of choice. This way users can compare the different styles and tactics utilized by others

5. Local record storage on any mobile device to allow offline functionality of the application. Creating a portable game station which can be taken on the go.

6. Work to minimize mobile device permissions. The application should only ask for permissions when it is absolutely necessary for it to complete a task properly. We want to make sure users don't get asked out of context.

3.0 Research Related to Project Definition

This section of our senior design document will go into detail about the research found by each of our members corresponding subsystem. In this section we look at projects that may have been completed previously but have a similar design and will help in the production of Laser Target Gallery. This research will give us a deeper understanding a grasp on what has worked for similar projects in the past, as well as what may have gone wrong. Doing so will allow our group to build our project in the most efficient way we see possible. Some of the topics we research will include Bluetooth vs Wi-Fi, the best stack to use to implement our webapplications, as well as the different types of boards that is used to ensure our project is fully functioning. The research found in this project will allow all our members to implement the design for each of their parts based on projects previously built.

3.1 Similar Projects and Products

Technology is continuously growing, and new more efficient components are becoming more readily available for use. This allows for new improved projects to be built as the years go on. This serves to our advantage, allowing us to improve on designs created by projects that may have been completed previously.

Advances in the way that devices communicate continue to grow and expand throughout the years. Larger companies such as Apple, now have access to Bluetooth 5. Bluetooth 5 offers the latest technology for communication across devices. It is the fastest way for devices to communicate, transferring data quicker than Bluetooth 4. Although it is such a great feature, currently this feature is only available to larger businesses with large amounts of money to spend, so we did not have access to Bluetooth 5 and would have to use Bluetooth 4. Although Bluetooth 5 is faster, Bluetooth 4 allows for fast efficient data transferring.

LaserHit is a company that produces a technology similar to what we are building. This company allows for safe firearm training from your home. Similar to what we plan to do, LaserHit implements a laser into a practice gun, and allows you to shoot at a target receiving points based on the location you hit on the target. This is like what we plan to design, a laser placed in a model gun that allows for reloading and real-life aiming practice. However, the difference in our project compared to what this company does is how the laser is registered as a hit. This company uses an iphone app that stands 1-10 feet away from your target and register the laser as a hit and show where on the target you hit. Giving you a score based on your accuracy. Additionally, the offer a mechanical firing mechanism where the laser casing is designed as a bullet itself. We can use this to help implement our design, allowing us to set up our laser “bullet” in a similar way. However, we are reading and sending the data in a different way using photoresistors and Bluetooth/Wi-Fi to transfer this data to our web application.

Another similar technology is the quadrant photodiode. The quadrant photodiode is a semiconductor device where four photodiodes share the same base, the organization of which can be seen in Figure 3.1.1. The current through each photodiode can be used to determine if a beam is off centered or not. The main application of this device, however, is centering beams. It is less often used to detect brightness or how far off a beam is from the middle of the quadrant photodiode. It can, however, be used to determine changes in position or angle after realignment.

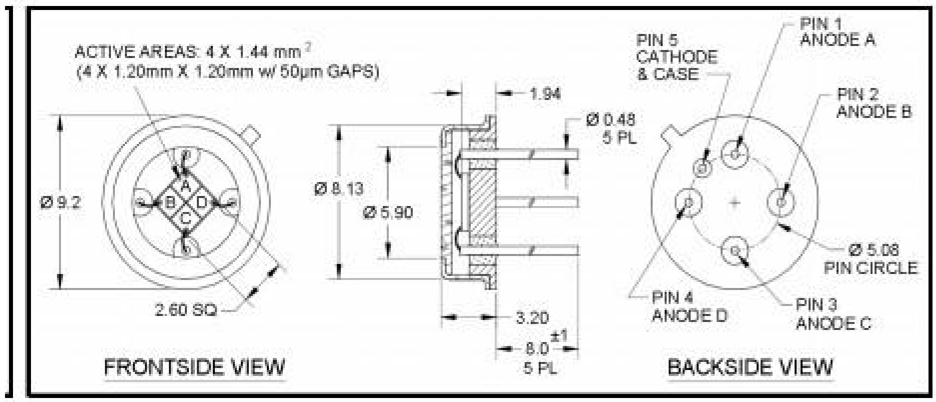


Figure 3.1.1: Diagram of a quadrant photodiode from the First Sensor QP Datasheet

3.2 Possible Related Architectures and Diagrams

In this section of the paper we looked at projects that were completed that are similar to ours. We looked at and discuss diagrams that were used that would be beneficial to our group moving forward and help us to build our final project. Previous laser tag projects were built using arduino boards. These projects show similar schematics to what we may need to create a laser gun as well as targets that receive the light and send out data.

3.3 Communication

There have been many technological advances in communication between devices. Bluetooth has greatly improved in the past few years and continues to grow. Although some of the newest technologies may not be available to us such as Bluetooth 5, we still have access to a wide variety of different Bluetooth chips. This allows for efficient communication between our laser targets and mobile application. Although Bluetooth is a great option, there is the option to use Wi-Fi as well. This also allows our devices to communicate with each other and send the necessary data. However, each of these choices have their pros and cons, which we will look at in this section. Allowing us to choose the best option that fits our project allowing it to run optimally.

3.3.1 Bluetooth

In this section we will look at the pros and cons of using Bluetooth as a means of communication between all the systems of our project. In the past few years, new Bluetooth technologies have come out. Bluetooth mesh is a new technology and is used for creating a large-scale device of networks that control, monitor, and automate the systems within the network, Bluetooth 5 is also a new, with Bluetooth 5 the range and speed at which devices can communicate has increased significantly. The diagram below gives a general overview of how the data would be sent using Bluetooth, sending information from the target board to the mobile application. In figure 3.3.1.1 below we can see the general relationship between signal strength and the distance the signal is being sent. It is not a linear relationship and can be disrupted by many environmental variables surrounding the area.

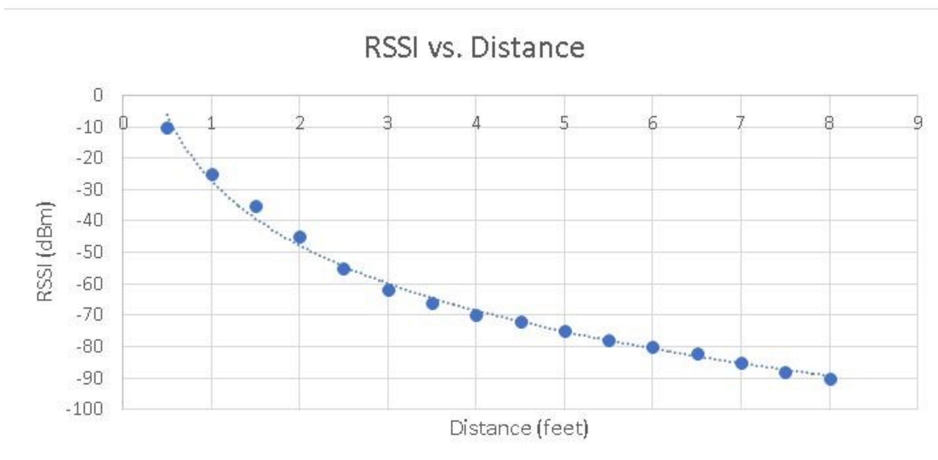


Figure 3.3.1.1: Bluetooth signal strength vs distance

There are a few options of Bluetooth we have the option to choose from. Bluetooth Low Energy (BLE) and Bluetooth Classic. Bluetooth classic is made for sending large quantities of data, which includes audio. Since it is built for sending such large amounts of data, it has a high connection latency as well as high power consumption. BLE does not send high quality audio data and to achieve low energy sacrifices high throughput. This makes BLE most suitable for sending data infrequently.

3.3.3.1 Bluetooth Module

In this section of the document we will look at the different Bluetooth modules that we could use to go along with the microcontroller that this is attached to.

There are many different options that could work well for us. Looking through the different features of each, will allow us to choose the module best for our project. In Figure 3.3.3.1 below it can be seen how the Bluetooth module would interact with the game. Once the target registers a ‘hit’, the data is sent using Bluetooth to our mobile application where the data is displayed for the user to see.

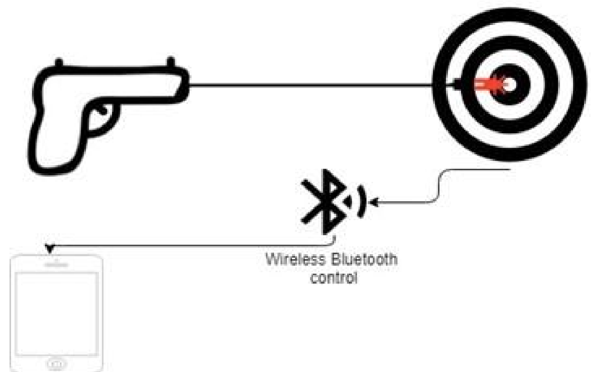


Figure 3.3.3.1: Bluetooth interaction

One of the first modules we will look at is the DSD Tech HC-06 Wireless Bluetooth. This module has an input voltage of 3.6 to 6 volts and the connection distance is 20 meters. This module also supports AT command, allows us to set the baud rate and the Bluetooth name and password. However, this module only supports connections between MCU, GPS, and PC. Since this device does not support IOS, we would not be able to create our mobile application, so this module would not work for our project.

Another Bluetooth module we will look at is the DSD TECH SH-HC-08 Bluetooth 4.0 BLE Slave Module to UART Transceiver for Arduino Compatible with iOS. This module has a 3.3-volt input voltage of 3.6 to 6 volts which prohibits any voltage that is over 7 volts. It can reach distances of up to 10 meters with a strong connection, anything over 10 meters is possible but not as effective. The issue with this Bluetooth module is that it only supports iPhones up to the 6s, anything later would not be supported.

Another option we have is the DSD TECH HM-10 Bluetooth 4.0 BLE iBeacon UART Module with 4PIN Base Board for Arduino UNO R3 Mega 2560 Nano. This module supports iPhone 4s and later, as well as Androids 4.3 and later. It has an operating voltage of 3.3 to 6 volts and has a range of slightly over 10 meters. Reading about this board there is also great documentation online for this module, which makes it easier to program and debug. This module is the best option for our project and can be easily implemented into our game.

3.3.3.2 Arduino uno vs. Bluno integrated BT board

We looked at a few different board options to use with our Bluetooth modules that were previously discussed. Our Bluetooth module chosen has 4 pins we need to hook up to, it has a Vcc, receiver, transmitter, and ground that must be attached to the microcontroller we choose.

We will first look at the Arduino Uno. The Arduino Uno has 14 digital pins on the Bluetooth that can be used as both input and output. Each of these pins operates at 5 volts and can receive a maximum of 20 mA. This board has two external pins that can be configured to trigger an interrupt on a low value. This could suit our project well since we need a way to trigger when the current reaches a high value, meaning the target is being hit by the laser. In Table 3.3.2.1 the features of this board can be seen.

Another option we have is to use the Bluno board offered by Arduino. This Bluno board integrates Bluetooth 4.0 (BLE) on to the Arduino board. The Bluno board integrates the TI CC2540 Bt 4.0 chip with the Arduino Uno board. This board provides the ability for wireless programming, establishing Bluetooth HID connections, and master and slave settings. Table 3.3.3.2.1 shows the features of the ATmega328 microcontroller.

Table 3.3.3.2.1: ATmega328 Features

Microcontroller	ATmega328
Operating voltage	5V
Input voltage	7- 12 V DC
Digital I/O pins	14 (6 with pulse width modulation)
Analog input pins	6
DC current per pin	20 mA
Size	60 mm x 53 mm
Flash memory	16 Kb

Comparing the two boards, they both use the same microcontroller. The difference being one is integrated on the board and the other would need to have a Bluetooth module added on. The issue with the Bluno board is that it can only connect with other members of the Bluno family, this could cause issues with the other microcontrollers we needed to communicate with. The best option for our project would be to use the Arduino UNO with the DSD TECH HM-10 Bluetooth 4.0 BLE iBeacon UART Module with 4PIN Base Board for Arduino UNO R3 Mega 2560 Nano

3.3.2 Wifi

We have discussed how we could use Bluetooth as a means of data transfer between our target board and our Mobile application. In this section we will look at the pros and cons of using Wi-Fi as a means of communication between our devices.

Using Wi-Fi would allow our application to communicate with our game board when both are connected to the same wireless network. This could allow our devices to communicate quickly and efficiently if on a strong network with a quality connection. In addition, by using wifi as a medium of communication we could add in more functionality to our project by tethering multiple microcontrollers each with a specific purpose. Such cases include comparing total pulses fired to registered hits, multiple player support, or synchronized tournaments. These proposed features are difficult to achieve when using Bluetooth as a medium of communication since there is a finite limit to how many devices can be paired together.

Another point to be considered is that to implement these additional features can come at an increased price to our budget. Since this project is not sponsored all expenses come out of pocket, We as a team must consider the efficiency of this trade off and determine if the potential for increased user satisfaction is worth the additional investment.

However, if not able to access a network, the user would be unable to play our Target gallery game. More so, our users could potentially face complications with our project if located near the edge regions of a wifi network. This would make our game stationary and conditional in its performance. Only able to be played when connected to a wireless network under ideal conditions.

To make the microcontroller of our choice be able to communicate wirelessly, we have the option to attach the CC3120MOD. This wireless network processor module integrates all protocols for Wi-Fi and internet, as well as includes built in security. It includes an 802.11 b/g/n radio, baseband and MAC with a crypto engine. This allows for a fast and secure connection. The module can be connected to an 8,16, or 32-bit microcontroller. Attached below is the diagram for how this chip would interact with our microcontroller. In table 3.3.2-1, the features of the CC3120MOD chip can be seen. How this chip functions can be seen below Table 3.3.2.1.

Table 3.3.2.1: CC3120MOD Features

Feature	Details
Wi-Fi Network processor Subsystem	Wi-Fi on chip TCP/IP Stack
Industry Standard socket APIs	8 Simultaneous TCP sockets 2 Simultaneous SSL sockets
Crypto Engine	Fast, Secure Wi-Fi connections 256 Bit AES encryption connections
WPA personal and Enterprise Security	Safe connections
TX Power	17 cbm at 1 DSSS 17.25 dbm at 11 CCK
Application throughput	UDP: 16 Mbps TCP: 13 Mbps Host Interface
Power Supply	2.3 to 3.6 V
Interface with 8, 16, 32 bit MCU	Up to 20 Hz clock
Low footprint Host driver	Less than 6 Kb
Support RTOS and No-OS Apps	Power subsystem
Integrated DC-DC converter with wide supply voltage	Direct Battery mode, 2.3 to 3.6 V
Low power consumption	Standby: 140 uA

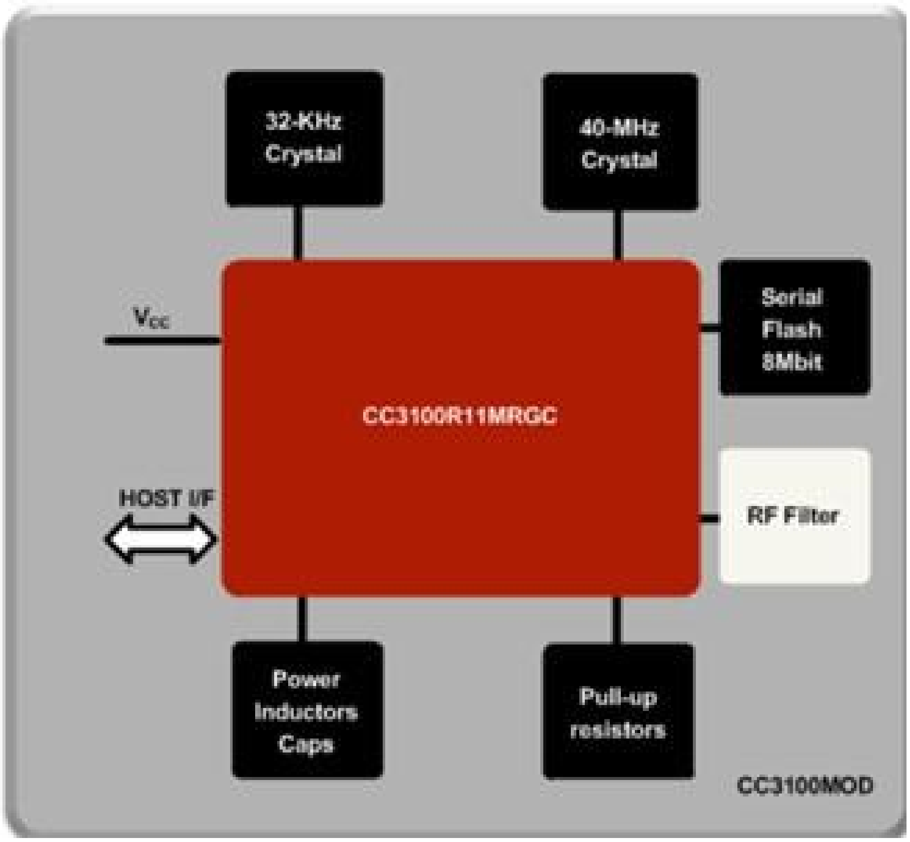


Figure 3.3.2.1: CC3120MOD functional diagram

3.3.3 Choice between Bluetooth and Wi-Fi

In this section we will look at the pros and cons of both Bluetooth and Wi-Fi. Going through the pros and cons of each will allow us to see what the upside and downside to each is and which would be most beneficial to use. Once going through the pros and cons, we will make our decision of which of these would be best to use for our project.

If deciding to use Bluetooth to send data from the target board to the mobile application there are a few pros and cons. There is a distance limitation using Bluetooth allowing for distances of up to 30 feet. Although this is a limitation of using Bluetooth, the game is intended to be played at close ranges of about 10-15 feet max, so this would not cause too many issues. There is also the issue of slower data transfer compared to Wi-Fi. Bluetooth operates on radio frequency, which can be an issue if many devices in the are using the same bandwidth. This could cause issues with the network. Some advantages of using Bluetooth are the availability of the devices at a low cost, low power consumption, and use of few wires. Low power consumption allows our battery to run for long periods of time without draining the battery. The low cost would allow us to implement this feature into our project without spending large amounts of money, and fewer wires keeps our game looking neat, giving a good experience to the players.

There are also a few pros and cons if deciding to use Wi-Fi to transfer the data from our target board to our mobile application. A few of the advantages of Wi-Fi include convenience, mobility, and deployment. Using a Wi-Fi allows our game to be played wherever there is a wireless network. However, this could also turn out to be a disadvantage if no wireless network is available in the area, the game would be unplayable. The setting process of using Wi-Fi is simple, not using many wires, also allowing our game to keep a clean look that makes the full game play experience enjoyable.

After looking at a few pros and cons for both Bluetooth and Wi-Fi, we can see that they are fairly similar in set up and use. For our project both options could be used to produce a working project. Although Wi-Fi may send the data faster compared to Bluetooth, this would not play too large a role in our project. The user playing the game will take approximately 30/40 shots in 30 seconds, which both components could handle. After consideration of both options, we have decided to go with Bluetooth. Our game only needs to set up one connection from our game board to the mobile application, which is made easy to set up using Bluetooth. Also, it allows for our game to be played from any location, with or without a wireless network around.

3.4 Software Comparison and Selection

There are many types of software development cycles incorporated by developers. Each cycle, however, starts from the same origin. By researching which tools are used to develop the project, developers can save both time and money by mitigating errors which arise from poor planning. Developers need to choose powerful tools to create amazing projects. Here our team has broken down the tools which we used in our mobile application into four distinct sections: Synchronization, Framework, Database, and Development Environment.

3.4.1 Synchronization Tools

Throughout this project, there are a numerous amount of changes to the code base. The potential to overwrite work that was implemented by another member can be a dangerous outcome which unchecked can lead to the failure of the entire project. In addition, there is a chance that the code base needs to roll back to a previous version to avoid errors. Due to this liability, there is a need to maintain proper documentation for each pushed version. Thankfully, this problem can be solved by incorporating synchronization tools into our project.

With version control systems, multiple developers can work remotely on the same code without having to worry about overwriting another's work. These systems can provide tools to keep watch over changes and can create sub-branches of the code base which can later be implemented into the main branch. Our goal for this project is to synchronize the codebase, documentation, and miscellaneous files.

3.4.1.1 Git

Git is one of the most popular examples of a version control system. Git is also a free software filed under the GNU General Public License version two. What Git offers is a list of features which allow developers to work in a non-linear development cycle.

These features include branching, merging, visualizing, and navigating the codebase. Git's branching system, a tool which clones the main repository of a code base, allows developers to hold a copy of the codebase locally for future editing. Developers can make changes to these local copies without disrupting the main repository.

A developer offers a commit when it's time to push the altered code back to the main branch. When the local branch is committed, the online branch is updated to match the pushed code.

The online branch must then be verified and approved before being pushed into the main repository. GitHub is used in tandem with Git to host the main repository. On the GitHub repository page, files can be viewed and downloaded by everyone with access to the link. Thus, acting as a secure synchronization tool for development teams.

3.4.1.2 Apache Subversion

Apache Subversion, or otherwise known as SVN, is another version control system that provides its users with functionalities similar to Git. Of course, there are a few noteworthy differences between the two software. While Git requires its users to download files from the main repository locally to their device, SVN works remotely.

Unlike Git, SVN does not require users to download an entire repository and instead offers the ability to work on sub-repositories. Also, SVN can achieve native compatibility with many popular IDEs with plugins. However, speed is a significant drawback to using SVN in comparison to Git.

By using Git, repository files are stored locally, meaning that accessing the files through operations is bottlenecked only by computer specifications. Since SVN works remotely, various factors such as internet speed, server stability, and computer specifications can adversely affect the performance of the software. A system comparison between the Apache Subversion and Git is seen in Table 3.4.1.2.1.

Table 3.4.1.2.1: Version Control System Comparison

	Git	**Apache Subversion**
Open Source	Yes	Yes
Free	Yes	Yes
Hosting	Local Machine	Remote
Branches	Copied Repositories	Sub-Repositories
Speed	Limited to Machine Specs	Various Factors
IDE Support	Third party support	Built in Plugins

Not only is Git software the team is familiar with, but the reliability it provides with its speed makes a compelling argument over SVN. There aren't enough benefits to SVN over Git to warrant learning new software from scratch. Using GitHub in tandem with Git is enough to properly document each implemented version.

3.4.2 Framework Selection

Frameworks are tools developers use to alleviate the tediousness of software development. Frameworks directly lead to increased development speed and stability when properly used. However, selecting a framework is not as trivial as finding the most commonly used tool.

A framework should be employed as a tool which assists in the desired goal rather than adapted to match the demand for the project. For mobile applications, developers have two different types of applications designs to choose which are native and non-native applications.

Native applications are applications which are designed for distinct platforms and are installed on devices which function off those platforms. Native applications utilize features provided by the operating systems that they run on and as a result have different development processes across platforms. The benefits to choosing a native design are abundant. Access to device features such as a phone camera, microphone, GPS, accelerometer, and swipe gestures can be extremely useful to a variety of apps.

Native applications are also capable of working without being tethered to a Wi-Fi or 4G network. In terms of performance, native applications will outperform a nonnative application most of the time. Native applications tend to be faster and more reliable since they have access to the devices operating system and are calibrated to the native language of the device.

As for a visual standpoint, the user interface provided by native applications provide users with a comfortable and engaging experience that is a staple to mobile devices. Lastly, developers are not limited to one platform when constructing a native application. Developers can take their finished application and use it a paradigm for all other platforms.

Non-native applications are distinctly different from their native counterparts. Unlike native applications, non-native apps run on many platforms without compatibility issues. The code base can be written only once and will work on most devices. As such, the non-native approach provides a more quick and affordable development cycle.

Because of their cross-platform capabilities, Non-native applications allow developers to focus on just a select few technologies. With a smaller pool of software languages and programs to learn, non-native applications are much quicker to deploy. With the ability to reach multiple platforms, non-native applications have the advantage towards reaching a large user base. However, non-native applications do have a caveat. Non-native applications do not have the built-in compatibility with each operating system that native applications have. There is a potent risk of non-native applications performing sub-optimally across some devices.

Our team is looking for a framework which can assist in the creation of a mobile application with a robust UI. Ideally, we want the framework to provide support for cross-platform development; however, support for Android is suitable enough. Additionally, our team would prefer if the framework has built in compatibility with our backend. Also, a well-documented framework is mandatory due to our collective lack of experience with mobile application development. Below are frameworks which meet the previously stated conditions.

3.4.2.1 React Native

As stated above, for mobile applications, developers must decide whether their application is native or non-native. React Native was made to bridge the gap between the two categories. React Native is a mobile framework devised to provide the accessibility that a non-native environment offered without losing the powerful features of a native environment. React Native uses the design philosophy of ReactJS, a web application framework, to create mobile applications with JavaScript. React Native constructs a hierarchy of UI elements to develop the codebase in JavaScript. This codebase can then be used to render UI for both Android and iOS devices.

Mobile applications built on React are natively rendered. Meaning they have access to features located on the device platform such as a camera or GPS. React
Native also adds improvements to the developer's experience. React is built with helpful error messages and developer tools that speed up the debugging process. Building the application is no longer necessary to see implemented changes to the codebase. By solely using JavaScript, React Native allows applications to be refreshed similar to websites. The time saved is not to be taken lightly. The constant need to rebuild projects over and over will add excessive hours of unproductivity to the development process.

Another aspect of React is its capability to reuse code across multiple platforms. Developers well versed in React has the proper tools to create applications for the web, Android, and iOS. Unlike native environments where a few developers each with a unique skill sets are required in order to achieve the same goal. Occasionally, developers have to switch between Java or Objective C depending on if they are working with Android or iOS; however, a majority of the codebase can be directly shared across platforms.

As for risks, React suffers from its youth. The documentation, while extensive, can still have room to grow. Also, React does not have support for every feature a native environment would provide. This means that depending on the scope of the application it might be better to choose a more native solution such as Android Studio or XCode.

Pros:

- Cross platform application development
- Native Features
- Robust user interface
- Code reusability
- Quick development cycle

- High performance

Cons:

- Relatively new
- Debugging is more complex at the host platform level

3.4.2.2 Ionic

Ionic takes the challenge to find a solution between native and non-native applications through HTML5. Ionic developers believe HTML5 will one day rule over all types of mobile applications in the same way it does with websites. Ionic particularly specializes in hybrid applications. These applications are, in essence, compact websites that run in a browser shell inside of a mobile application. With these websites, Ionic can access the native platform just like React Native. Similarly, Ionic takes care of all the front-end user interface interactions. Ionic excels at providing eye catching animations and designs with simplicity seen only by small selection of other platforms.

To function properly, Ionic needs to work in conjunction with a native wrapper. Programs like Cordova or PhoneGap are highly recommended. When working together, Ionic can utilize the low-level browser shells found on iOS and Android. This effectively distances itself from having to interact with applications such as Google Chrome or Safari and in turn providing a native design experience.

Ionic is completely stable and open source. Meaning there is no monetary risks involved when considering using the framework. It also comes standard with default user interface elements that developers could simply pick and choose without the need to code. Basically, Ionic functions similar to Bootstrap, a user interface framework used on websites, to reduce the time developers spend coding their mobile applications. Like React, Ionic's codebase is reusable which further reduces the time spent learning how to code in multiple platform languages. Lastly, Ionic is much easier to test with compared to other frameworks. Ionic offers the ability to test mobile applications remotely, through safari or web browsers (in iOS devices), or directly on a native platform.

Pros:

- Utilizes AngularJS framework
- One code for many applications
- Highly extensive user interface features
- Native Cordova/PhoneGap plugins
- Open source

- Robust community

Cons:

- Native plugins do not cover all use cases
- Development on Android is fragmented
- Potentially less performance than a native counterpart

Framework Comparison

Table 3.4.2.3-1: Framework Comparison

	React Native	**Ionic**
Free	Yes	Yes
Open Source	Yes	Yes
Cross Platform	Yes	Yes
Native	Yes, with JavaScript design	Yes, with native wrappers
Languages Used	Java, JavaScript, and Obj C	HTML5, CSS, JavaScript
Community	Vast	Vast
Target Platforms	iOS and Android applications	iOS, Android, Electron, Mobile and Desktop Browsers
Code Reusability	Different UI codebases with shared logic	Shared UI codebase across all platforms
User Interface	Native iOS and Android elements specific to each platform	Web UI elements that work across all platforms
Offline Capabilities	Yes	Yes
Performance	Fast and stable	Fast and stable
Development Speed	Medium time investment	Short time investment

It was particularly tough decision for the team to pick between these two frameworks. React and Ionic provide similar benefits with different design philosophies. In the end, our team chose React Native over Ionic because we plan to develop specifically on Android and iOS.

Ionic’s hybrid-web based user interface would work perfectly if we wished for our mobile application to work on any and every platform with web capabilities.

However, we wish to focus solely on providing the best possible user experience we can, and we feel React Native is the best choice to reach our goal.

3.4.3 Database selection

Databases are the backbone of software projects that requires data to be stored and retrieved at a given notice. Web and mobile applications use databases to keep track of user information and other relational objects for future use by the frontend.

In the case of our team, we need to choose a proper database which must not only store user information such as credentials, but data transmitted by the Bluetooth microcontroller on the target board. For this project, while the board is activated, data will arrive quickly. It is important that the database we choose is capable of quick storage and queries.

For a database to meet the team's needs, we require a gradual learning curve, speed in terms of storage and queries, efficiency, online documentation, and an active community capable of assisting in errors during the development process. Our team also wishes to use a database that can assign relations between objects for the potential of a user to user interaction features in the future.

3.4.3.1 Realm

Realm database is a database which functions for both offline and real-time applications. Realm is built from scratch and has its own database engine which does not solely rely on key-value storage. By not relying on key-value stores, Realm is excellent at handling vast amounts of data for applications that have a high-performance usage.

Realm is not a relational database; instead, it is object-oriented based. Because Realm is an object-oriented database, it bypasses the need to run many queries by representing information in the form of objects rather than tables. Realms objects directly represent the database. As such, during read-write operations, they do not need to be manipulated, and in turn increase performance.

Additionally, Realm objects can be accessed concurrently from multiple threads without the requirement of concise lock configuration.

Realm has many features that can be eye-catching to developers, some of the more notable characteristics are Realm's synchronization tool, security, and offline tool. In cases where the device is offline, Realm's synchronization process will remain active in the background and will record a user's interactions and requests locally. When the device reconnects, it will then send the recorded data and process the requests automatically. Realm is also a completely open source and free database. Factoring in all of the features, Realm fits all the criteria our team is looking for in a database.

Features:

- Real Time Collaboration
- Authentication
- Offline Synchronization
- Server-side Event Handling
- Concurrency support
- Data memory mapping
- Clear Documentation
- AES-256 Encryption for Android
- High speed

3.4.3.2. Firebase

Google Firebase is an incredibly powerful database which can provide incredible support as a backend for both web and mobile applications. Firebase is a real-time NoSQL database.

Like Realm, Firebase has the tools to support data synchronization and offline modification. Additionally, Firebase has the capability to act as both a model and a controller in a model view controller grouping. Firebase pairs well with software that aim to provide a user interface such as react native. When working in conjunction with React Native, Firebase delivers a cross-platform capable API which requires minimal setup.

Furthermore, no application server is necessary, since the real-time database Firebase provides can be reached directly from a mobile device. However, Firebase does have one complication. While security isn't the most notable concern for our project, Firebase does not encrypt data on the server side, and it would be ill-advised to ignore.

Features:

- Realtime Database
- Cloud Storage
- Cloud Functionality
- Performance Analysis
- Google Hosting
- Offline persistence
- Scalability
- Hosting

3.4.3.3 SQLite

SQLite is a database modeled to deliver local storage to mobile applications. SQLite operates as a relational database management system (RDBMS). SQLite like other RDMS, organize data in tables and are structured such that tables relate to each other in specific ways. Some examples are many to many, many to one, one to many, one to one, or one and only one. Developers can effectively control the privileges of each table by implementing these relationships in a way that models the intended plan for the project.

SQLite as a database is ACID acquiescent meaning it upholds consistency, isolation, and durability while maintaining most SQL based criteria. SQLite's architecture can handle all types of data without any complications or complexity.

While SQLite does not offer built-in offline storage, it can achieve offline constancy by incorporating external storage plugins from different frameworks. Finally, SQLite is a database which is open source and free to the public. A comparison of all of these databases are seen in Table 3.4.4.3.1.

Features:

- ACID compliant
- Offline persistence
- WxSQLite, SQLCipher, and SQLiteCrypt Encryption
- Zero-Configuration
- Serverless
- Stable Cross-Platform Database file
- Compact
- SQL language extensions

Database Comparison

Table 3.4.3.3.1: Comparisons of Databases

	Realm	Firebase	SQLite
Price	Free	Moderate	Free
Open Source	Yes	No	Yes
Database Type	Object Oriented	NoSQL	Relational
Documentatio n	Excellent	Excellent	Minimal
Supported Languages	6	8	Over 30+
Ease of Use	Simple	Moderate	Moderate
Performance	Excellent	Excellent	Excellent
Community	Robust	Robust	Robust
Security	Secure	Potential Risks	Moderately Secure
Hosting	None	Excellent	None

No mobile application developer can go wrong when choosing either of these three databases. SQLite is a database proven to work for many years when tracing back to its other forms: SQL and MySQL. Our team is most familiar with relational databases which makes SQLite the most comfortable and safe option. SQLite is very flexible when factoring in all of its supported languages and does not stray too far from the other databases when it comes to performance.

As we researched each database, Google Firebase was consistently praised by many different sources. Some sources included previous senior design groups from UCF. To get firsthand accounts of Firebase's performance from alumni stood out the most to our group. However, our team want to minimize the cost of our project as much as possible. Choosing Firebase when there are other equal quality free databases available was a hard-selling point.

Realm seemed to fit the niche that our group was looking for. Realm is free, offline capable, fast, secure, well documented, and open source. While it doesn’t have nearly as many supported languages as SQLite the languages it does support makes up most of our codebase. While Realm doesn’t come packaged with a hosting service like Firebase, the scope of our project primarily focused on offline use. Meaning the hosting service is a non-needed feature.

Our team did consider Google Firebase, but because we have no sponsor for the project, we decided it is in our best interest to choose a free option. As for SQLite, we couldn't find anything feature that could leverage our decision over Realm DB. Realm's storage and query times are significantly faster due to its object-based database design; moreover, Realm is arguably more secure than SQLite's single disk file. Realm is also highly recommended by many mobile application developers. Our team feels that choosing Realm provides an excellent learning opportunity with object-oriented databases.

Realm, Firebase, and SQLite are all amazing databases which are widely popular among developers. Each offers powerful features such as performance, security, offline persistence, hosting, scalability, or support. However, Realm DB stands out the most to our team. Realm's ability to provide all the previously listed criteria while also ensuring ease of use, clean documentation, compatibility, and free use is just what the team was seeking for in a local database. We hope that our choice will act as a boon to our development time as the project proceeds further.

3.4.5 IDE Selection

IDEs or Integrated development environments are software applications that aim to reduce the workload that typically occurs during software development. They provide developers with a boost in productivity by reducing the amount of memorization that is normally involved when programming in any language.

IDE’s come stock full of libraries and plugins that automate normal developer tasks. This includes auto completion, on the fly function searching, in application function definitions and syntax, variable memorization, and all types of debug options.

IDE’s can walk through the codebase step by step keeping track of every variable initialized and can even set breakpoints to skip to specific regions. IDEs are incredibly powerful tools that greatly reduce the complexity and time it takes to work on large programming projects. IDEs seek to provide the same functionalities of many different utilities under one application.

3.4.5.1 Visual Studio Code

Visual Studio code is an extremely popular IDE among developers. VsCode is the free version of Microsoft's visual studio (a premium IDE). VsCode is most unique for its extensions tab. Here, developers can search for all sorts of plugins for a particular language they feel like coding in. Any language you can think of VsCode has support for it. The community is very active and responsive. Our group has personal experience with VsCode. VsCode even provides tutorials on how to properly set up different programming languages on to the developer's desktop.

While the flexibility of VsCode is great, one glaring issue is that as an IDE VsCode requires more work than most IDEs to function correctly. Even though you can press install on any extension and instantly have it integrated into the software.
Sometimes the outcome isn't what is expected. Occasionally extensions require installation of some obscure software which can only be installed manually through the command line. Other times the extension is not working due to the installation path malfunctioning. Sometimes you can follow the directions exactly and still run into complications with the extension.

In other words, while in theory Vs Code can support any language and has hundreds of thousands of extensions to download it still just doesn't provide the simplicity developers seek when looking for a robust IDE.

Pros:

- Free
- Compatible with just about every language.
- An enormous and active community
- Well documented plugins
- Massive about of extension support

Cons:

- Confusing and unreliable at times
- Acts more as a text editor with plugins designed for language support
- Not a traditional IDE

3.4.5.2 JetBrains WebStorm

JetBrains WebStorm follows a more traditional IDE application. WebStorm is an incredibly fast and flexible IDE when analyzing code. Like a typical IDE, WebStorm will detect the programing language being used and notify developers of syntax errors. Furthermore, just like other JetBrains IDEs WebStorm will offer suggestions to improve the code. As stated before, WebStorm is capable of tracking all variables used and unused throughout the code base.

WebStorm includes debugging capabilities that rival most other IDEs. WebStorm can define its own tests allowing code to be inspected right inside of the IDE.

WebStorm is fully compatible with git and even includes advanced functionality which is not included in most other IDEs.

The downsides to WebStorm is that it is not a free software. Nothing else really sticks out too much otherwise.

Pros:

- Code inspection
- Quick refactoring
- Well documented
- Integrated testing
- Unused method detecting
- Built in code synchronization
- Extensive compatibility with Git
- Built in debugging
- Compatible with react native

Cons:

- Runs off Java
- Not Free

IDE Comparison

Table 3.4.5.2.1: IDE Comparison table

	VsCode	WebStorm
Price	Free	$449 - $649
Code Inspection	Mediocre	Excellent
Documentation	Excellent	Excellent
Plugins / Extensions	Abundant	Decent
Compatibility	Excellent	Good
Refactoring	Mediocre	Excellent
Debugging	Needs a plugin	Excellent
Code Synchronization	Nonexistent	Excellent

Under normal circumstances the price point for WebStorm would instantly be disqualified by any form of comparison; However, this time the program is free of charge for our group. As such it completely outclasses Visual Studio Code in just about every category except initial load time.

Thus, we are using WebStorm as our primary IDE for this project. The benefits it provides is astounding and are fully taken advantage of.

4.0 Hardware and Software Design

The following sections are the explicit details on how each system is to be constructed, including a full analysis on each component and how it is used in to achieve the goal of each part. We will look at where we will order the parts from, which places would allow us to save on cost and efficiently get all of the components that we needed.

4.1 Hardware Block Diagram

Diagram 4.1.1 below shows the active system of our Laser Target Gallery game. Each of the power supplies are feeding two microcontrollers on both sides of the project. The microcontroller on the left side is associated with the Laser Gun and is going to be passing data to the Bluetooth, along with the microcontroller on the right side of the graph, associated with the targets and printed circuit boards. Between the two Arduino computers is a majority of the hardware including the laser diode, along with a lens system, and photoresistors connected to printed circuit boards.

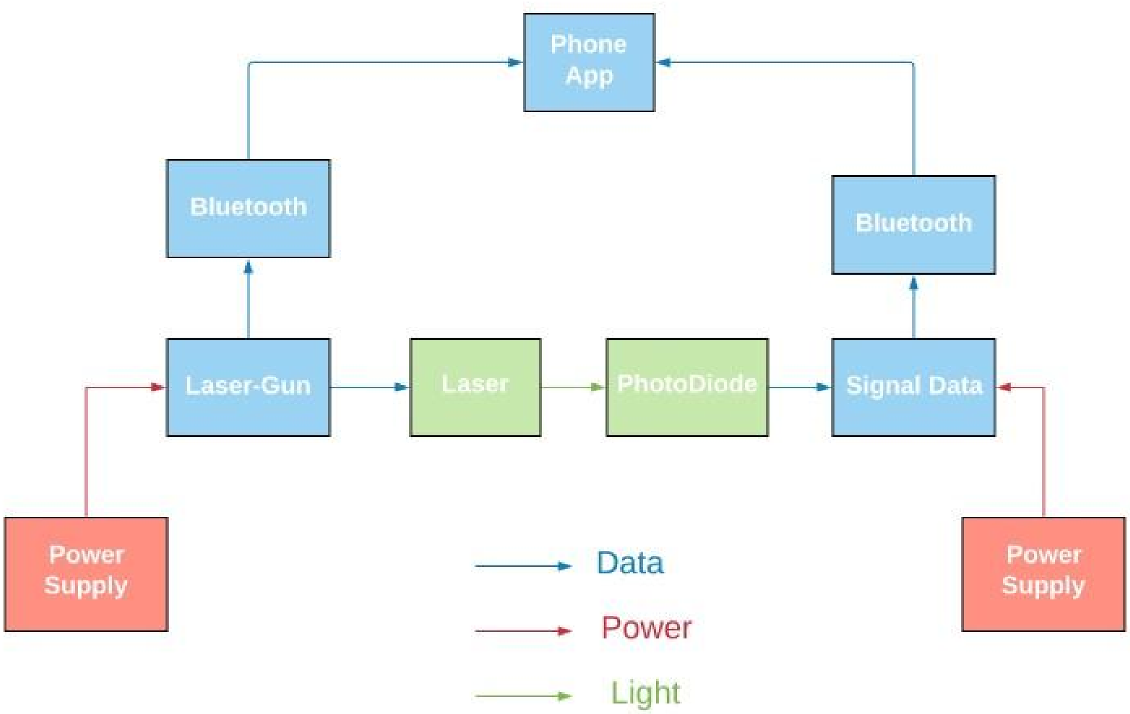


Diagram 4.1.1: Hardware and Project Design

4.1.1 Signal Data

For relaying the game analytics to another device our best choice was through a wireless form of communication, saving to a transferable memory apparatus would not have been a modern solution for the sake of our project. Our best option was through wireless, however we needed to choose between wifi and bluetooth; Though wifi is very useful and the modern epitome of wireless communication, bluetooth seemed to fit our project much better, why we chose bluetooth is explained more in an earlier section.

The two Arduino microcontrollers associated with our game, have two separate bluetooth chips. And they are called upon at different times to send signal data. The reason behind this being that we have to manage the output signal data from the two computers one at a time and with respect to the flow of the game.

The bluetooth module attached to the back of our target board is needed to transmit the analog input received from the target. The data is first run through our algorithm to decide the points the shot will receive. This should be fairly simple to process since we are only taking into account 4 inputs at a time. Once this number has been generated we pass this data to the mobile application that we have built. Allowing for this number to be displayed by the application.

4.2 Software Block Diagram

The Laser Target Gallery software process design is denoted in the Diagram 4.2.1 below, as stated above the two microcontrollers are the brains at the top of the process, and they are where the software process begins. After including any necessary libraries, Initializing global variables, enabling interrupts, and setting up and enabling the timers, the process is ready to enter its main loop. Inside the main loop for the microcontroller associated with the target boards, the main job is to increment counter when the photoresistor shows a resistance output, along with that this microcontroller has other jobs that is touched on in another section. The microcontroller associated with the gun has a similar game loop, it needs to count when there is input signal from the triggers hardware and send a power signal to the laser diode. After each has completed their respective jobs the last job of the two microcontrollers is to simultaneously send the information stored in registers wirelessly to our web application server database to be stored and manipulated.

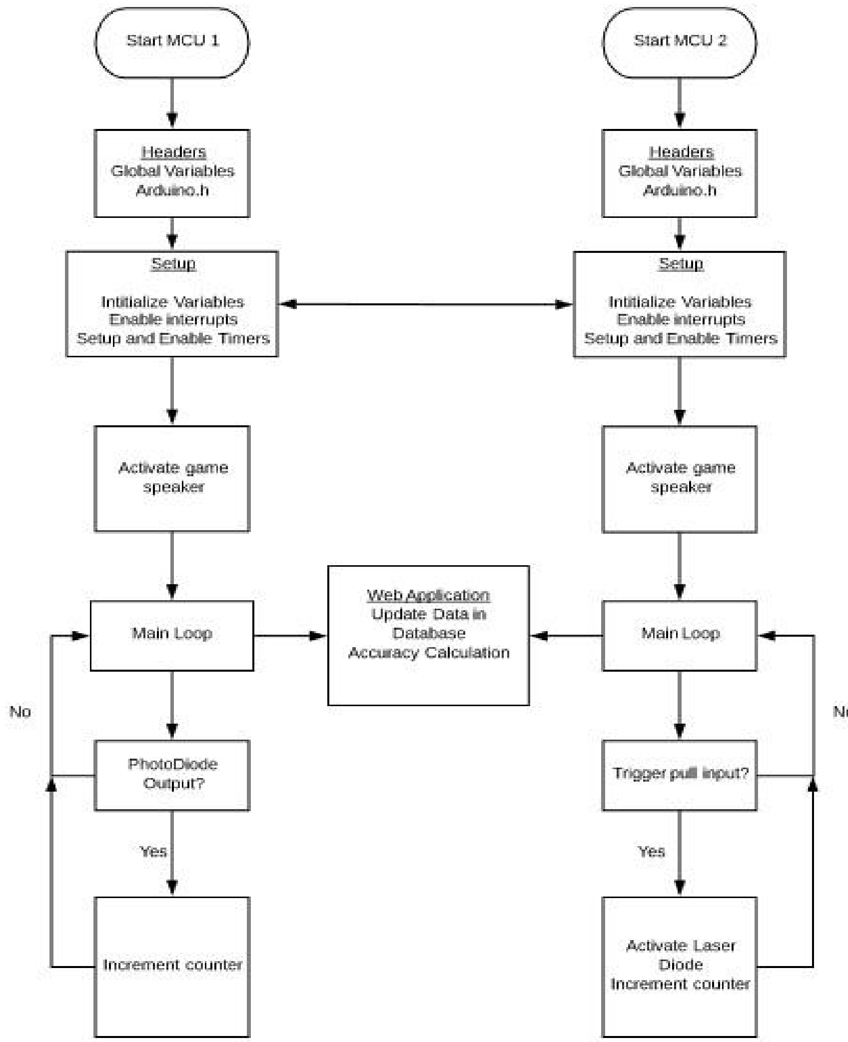


Diagram 4.2.1**:** Software Process Design

4.3 3D Printing and Modeling

This section tells you about how a lot of the hardware of this project is designed and manufactured through the 3D printing process.

4.3.1 3D Modeling Software

The first step in the 3D printing process is modeling the part to be 3D printed. This is most often done in CAD (computer-aided design) software, where parts are sketched, edited, then modeled into a complete and complex design. This is done by using a list of constraints and specifications on different additions, subtractions, and other modifications to a part. The end goal of using these types of software is to develop what's called an STL, or stereolithography file. This file can be given to a different piece of software called a slicer. This program takes printer settings then output them into G-Code. This G-Code is a set of instructions for computer automated tools, but specifically can be fed into common 3D printers, which if successfully done, allows for the printing of plastic components of the project

4.3.1.1 AutoCAD

This program, developed by Autodesk, is software used by many industries to design parts to be machined or printed. The advantages of using AutoCAD is that it is a very robust and powerful system, with well supported help and documentations. The system itself is very intuitive and has very minimal graphical or technical glitches. However, the tool does require a very powerful machine to run on. The biggest downside to this program is the very expensive cost of it. There's no license easily accessible as a student and the program even if used over the course of one month is $200. Chances to this program is applicable, but the time to be able to use it is much more strained. As the goal of this project is to output STL files, this software may be used sparingly.

4.3.1.2 FreeCAD

This program, similar to Autocad, is used by much smaller industries and for more personal use. This is an open source tool that covers many of the same features as AutoCAD, but still requires a powerful computer and has much more technical or graphical glitches. One of the side effects of it being open source is that many of the features are not as well documented, have varying degrees of effectiveness, work less well with each other, or are buried or obscured in different menus. The biggest upside, however, is that it is a free software with lots of documentation. This tool can be used to output STL files but is mainly be used for simpler or easier parts, as more complicated pieces in this engine is exponentially more difficult in this software.

4.3.1.3 Slic3r

After an STL file is created, it is then sent to a software which converts the shape into several lines of code which a machine can follow to print a part. These lines of code tend to follow only a specific set of parameters. Namely, moving the x, y, and z axis of the machine, and moving the motor controlling the filament output. Processes such as heat, size of the pen tip, bed leveling, and fan speeds are most often controlled manually outside the code.

Slic3r is the tool which is used to convert the STL files into G-Code. It functions by placing an STL object onto a bed, mapped to the bed of the printer you are using. By inputting several parameters such as speed, size of the nozzle, bed height, layer height, extrusion width, size of filament, and other parameters, it outputs the code into a file which can then be read by the printers hardware. The line of code that it produces is a linear set of instructions, where one line is read then executed at a time. For example, prints usually start with a command to go to just above the surface of the bed, start moving the filament motor, then move while the filament is extruding back and forth over the surface of the bed.

The advantage of using Slic3r is that it is a free open source program that one of our members has a lot of experience in. The amount of control and fine detail it allows in the print parameters should be robust enough to create the resolution required for smaller features such as those which would require set screws or heated press inserts.

4.3.2 3D Printing Hardware

Thanks to the support of the company nScrypt, we are allowed to borrow their high tech 3D printing technologies to print some of our parts. The way these tools work is on two levels, the gantry system and the gizmo system. The larger part of the machine, or the gantry system, moves the tool plate on the system, where smaller components called gizmos are attached. These gizmos have a number of uses such as depositing pastes with pneumatics or milling, but the product we are intending to use is the nFD, as can be seen in Figure 4.3.2.1. The ganry is controlling the x, y, and z components of the printing process while the nFD controls the heating and dispension of the filament onto the bed.

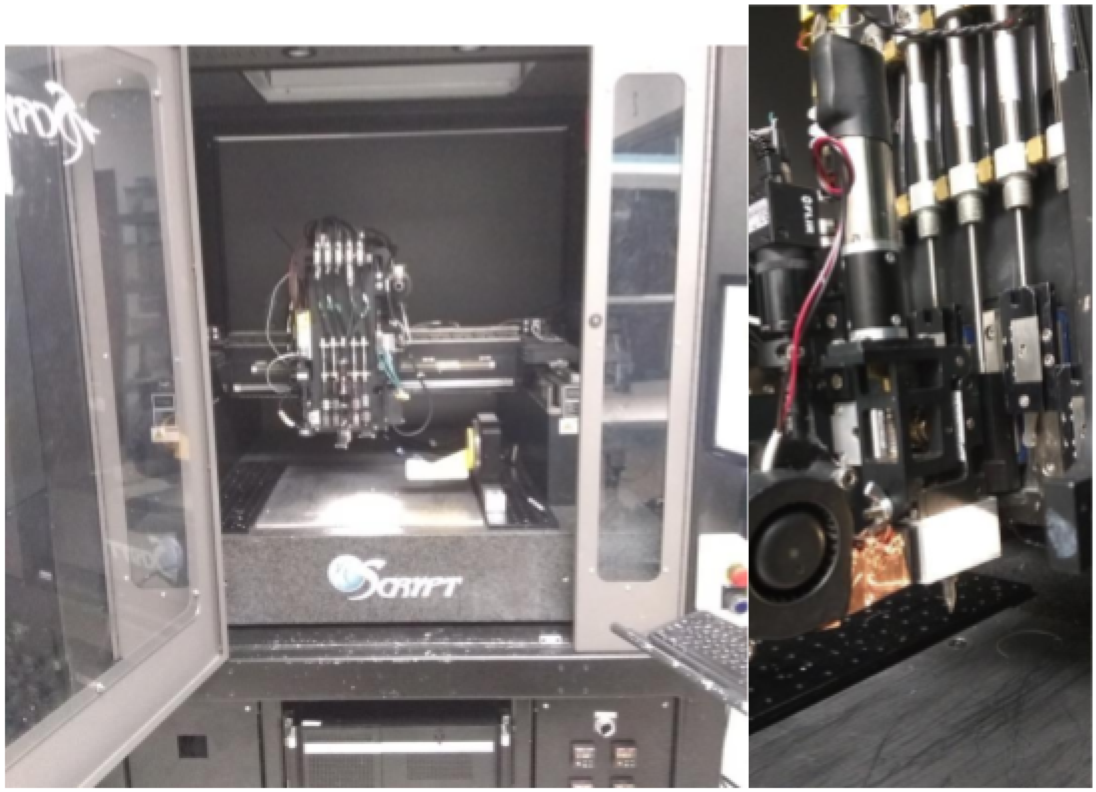


Figure 4.3.2.1: One of nScrypt's 3Dn Systems (left) and nFD plastic extrusion gizmo (right)

4.4 Laser Gun System

This section tells you about the different subsystems inside the laser gun in great detail, breaking down each piece into why each type of part was chosen.

4.4.1 The Power Supply

First and foremost it is important to look at the power limits that the Arduino Uno, being that the minimum operational voltage starts at 5 V, meaning the power supply we chose for our project needs to stay constantly above or near 5 V. The upper limit is 20 V, this is important for testing to avoid destruction of development materials. However, the recommended voltage to supply to the Arduino Uno is between 7-12 V.

For our project we chose to use a rechargeable battery pack to avoid having loose wiring hanging around the gun and display. A simple solution that we are using is a Portable Charger (Rechargeable Battery Pack) that separately supplies both Arduino device via USB. More specifically we are using:

- 2 x LC 1640 1300 mAh 3.7 V Battery

Table 4.4.1.1: Power Supply Specifications

LC 1640 1300 mAh 3.7 V	
Weight	.71 Ounces
Max Current	3 A
Charge	1300mAh
Size	16.7 mm Diameter 35.65 mm Height
Charging Voltage	4.2 V

The strength of this power solution is convenience and simplicity. The power consumption due to the microcontroller along with the other devices should be more than manageable leaving plenty of game play time, and a quick recharge, due to the very high 1300mA output, and the draw of both the Arduino and the diode combined being in the hundreds of mA.

Additionally, the power supply is compact enough to fit into the handle of a 3D printed model of a gun. To do this, we can use a clip design to attach the battery in battery cases as seen in Figure 4.4.1.1. This was designed in the FreeCAD Software, using the dimensions listed in the product. The way it works is by using the natural flexibility of the plastic in the 3D printing process, the print itself will snap and hold onto the power supply via friction, while still allowing the wires to leave the top and bottom. The idea was to avoid any drilling or other damaging means of securing the power supply. Adhesive could be used, but may be difficult in the case of troubleshooting.

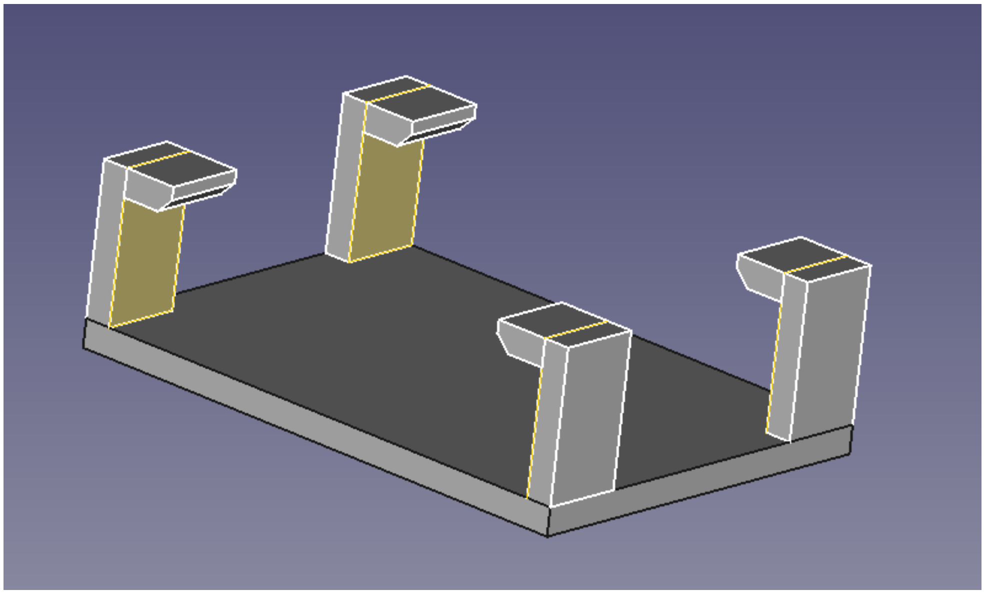


Figure 4.4.1.1: Clip designed in FreeCAD to secure the power supply inside battery cases in place

4.4.2 The Microcontroller

The Arduino Uno microcontroller has 6 different Analog Input pins, and 14 different digital input and output pins, which is important when it comes to our project. The computer for the laser gun needs to be able to handle inputs from a trigger(switch driven), and deliver appropriate output signals to a laser beam diode. We can see the configuration below as an example on how to connect the output power to the laser diode as well as how the trigger is attached to determine an input.The Arduino suite comes with a free independent development environment that is made specially for the ATmega328p.

Inside this development environment, we can set a program that runs checks at certain intervals, often thousands of times per second. This program will simply wait for a trigger, input, fire the diode for .1 seconds, then wait .9 seconds. This way, mashing the trigger will still only fire at most once a second, but then also control the pulse length of the beam. Additionally, a piezo speaker can be directly attached to the microcontroller such that when the during a laser fire, there's auditory feedback as well as visual feedback that the laser is being fired.

One of the more important parts of the microcontroller is that it should be very small. We are talking on the scale of less than 2 by 2 inches, else it would be problematic to mount inside the gun. To do this, the printed circuit board can be adjusted such that it does not need anything more than its chips, one input pin to detect a trigger pull, two variable power output supply pins to fire the laser and activate the piezo speaker and power inputs.

4.4.3 The Laser Diode

For most laser tag applications, the wavelengths used are usually a 950 nm infrared light or a more common 650 nm red light. The decision to use a 650 nm red light was largely based on the fact that to have a visible indicator on the board itself, it would be far less complicated to use an already visible light as opposed to a board that absorbs infrared light and emits visible light. However, this does not come without cost. There is much ambient light can that can interfere or reduce the change in intensity of the incoming light. This is because photoresistors are not sensitive to a specific band of light, but an entire spectrum of light, however, this will be discussed in the later section 4.4.1. The Photoresistor. It should be noted, however, that a later decision to use a green laser diode was made in Senior Design 2 and discussed in Section 11.

However, in the decision to have the power to the laser diode come directly from the microcontroller for a better form factor, we would have to understand that there are certain restrictions placed on the laser diode. For example, there wouldn't be an easy way to output more than 5V without a converter and the laser diode shouldn't overdraw more than what the microcontroller can output. The laser chosen in this case is the Lights88 "DOT" Laser generator, which fits the expected level of brightness and power draw, the specs of which can be seen in Table 4.3.3.1.

Table 4.4.3.1: Listed Specs of Purchased Red Laser Diode Stats

Wavelength	650 nm
Power Output	5 mW
Operating Voltage	3V-5V
Operating Current	<50mA
Operating Temperature	0 - 30° C
Size	13mm Diameter, 35 mm Length

There only potential issue here is that it is a class 3R laser. This means that it is not safe to look directly into the beam, but the beam spot it outputs should be safe to look at. We will decrease the output intensity through the lens and aperture system, however, such that the total output beam will be much weaker and even more safe, however, like all lasers, should not be looked directly itno. Also included in this diode though is a small adjustable lens. This will allow the beam to focus through a large spread to widen the angle of the beam, which will be much more useful later in the lens and aperture setup.

4.4.4 The Laser Emission Profile

When a semiconductor emits a laser, it will often create an intensity profile that is gaussian or normal, in nature. This means that we can replicate a cross section of the intensity profile of a beam to see the distribution of intensities. We can see this in the Figure 4.4.4.1 below.

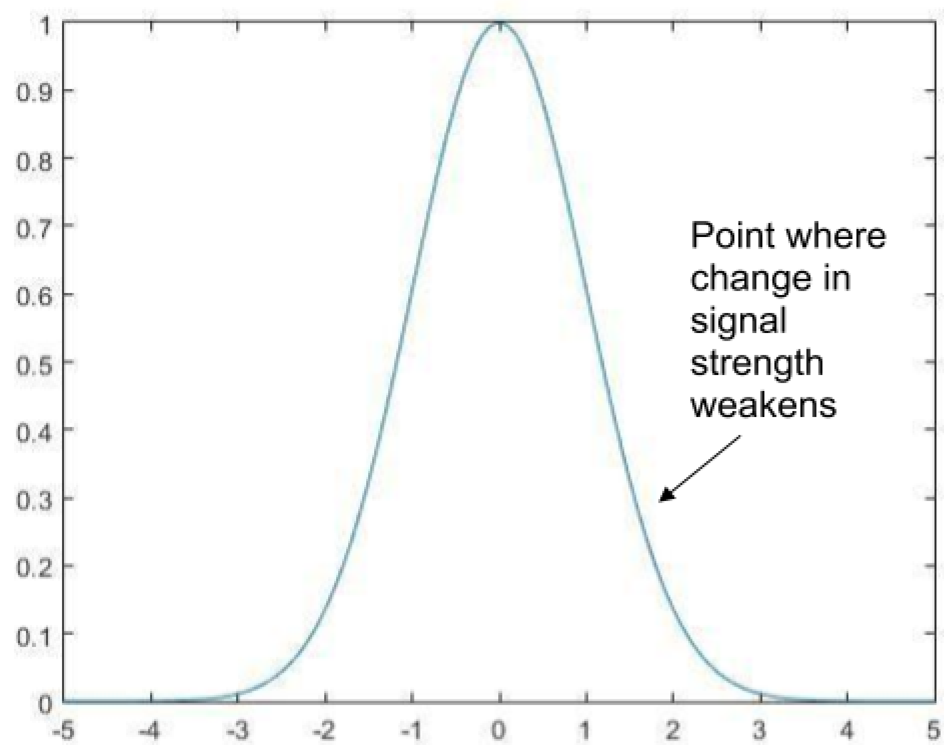


Figure 4.4.4.1: Expected unobstructed laser output along one axis, created in MATLAB

For the sake of simplicity, we only used this as a reference in a 1D plane and assume our photoresistors are 2 units apart. Given that a signal from the photoresistor is proportional to the incoming intensity of light, we can use this graph to visualize an approximate signal response from a photoresistor. For example, a direct hit would place the two photoresistors at -1 and 1. Because the light intensities are the same, any detected hit from the photodetectors where the two signals are the same can be considered a direct hit. Similarly, an off centered hit, placing the photoresistors at 0 and 2, would have a huge signal coming from one photoresistor but a very weak signal from the second. While this would count as a hit, this shot would obviously be worth less points. From this, we can deduce that there is some operation where we can determine how many points a shot is worth based on the difference between signals. However, there is a mild problem. Due to ambient light, the actual received level on signal is already at above 0, but that can be corrected for on the software side. But, if we used our photodetectors wrong and judging exclusively by signal ration, photoresistors at 0 and 1 would be worth less points than at 2 and 3. We can rectify this by limiting the outlying intensity profile on either side of the intensity distribution using the lens and aperture system and get a different expected output, as seen in Figure 4.4.4.2.

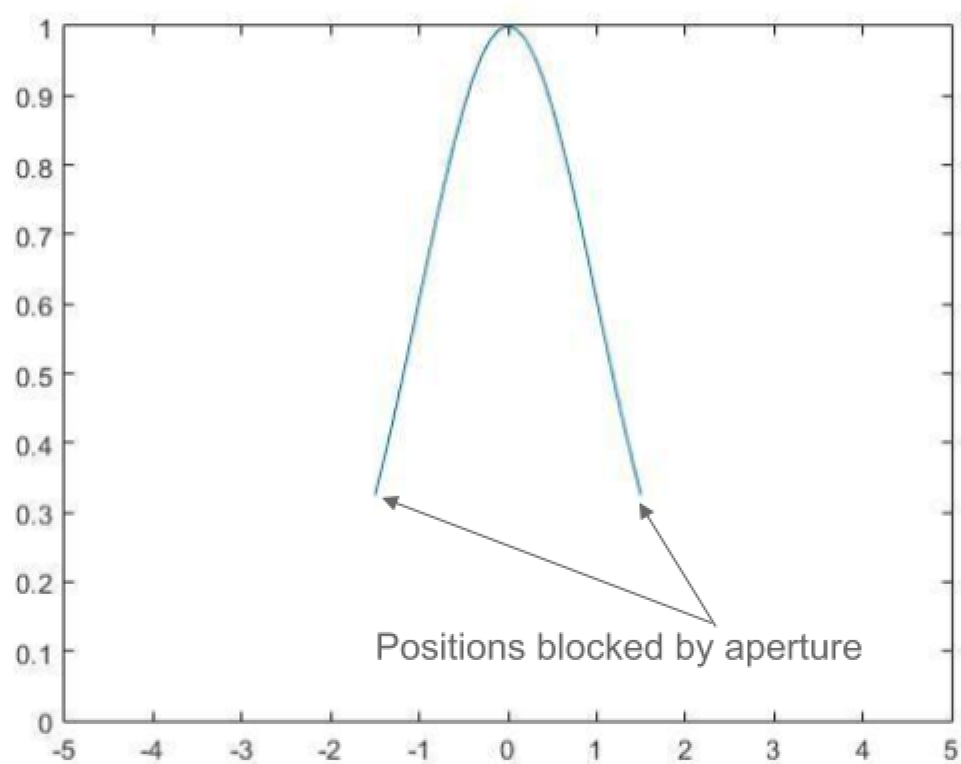


Figure 4.4.4.2: Expected Output Intensity Profile at Aperture

It should be noted, however, that as the beam passes through a hole such as an aperture, it will experience diffraction. Circular diffraction can be seen like that in Figure 4.4.4.3, called Airy diffraction. This is caused when coherent light interferes with itself constructively and destructively, causing rings to appear around the projected beam of light. These beams are repeated at even intervals until the light either becomes too weak to observe or the coherency is lost. It is anticipated, however, that due to the large size of the aperture, any diffraction can be considered negligible. This will, however, be tested for later.

Figure 4.4.4.3: Airy Diffraction From White Light Source Through Small Hole

The laser diode itself and negative lens was used to expand the laser beam onto the surface of the wall to test for issues with the beam itself. The lens was only about 5 mm in size, about the size of the aperture to be used, and would likely indicate any issues in the size of the aperture at all. We can see the output beam shape in Figure 4.4.4.4.

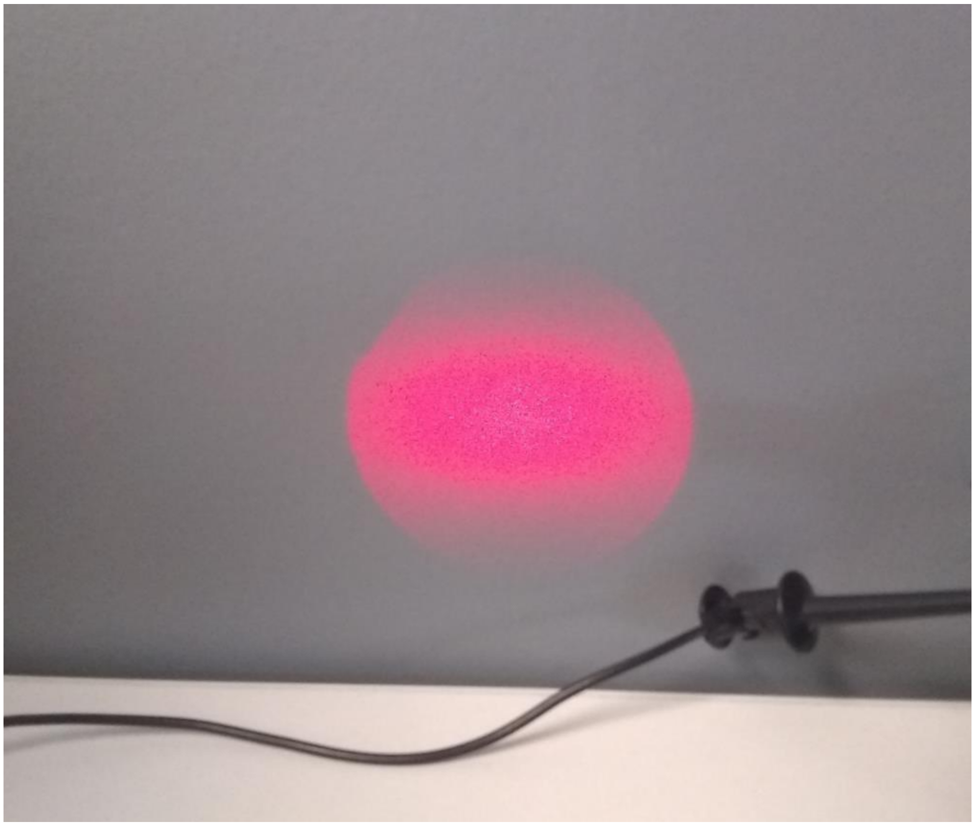


Figure 4.4.4.4: Output Beam Through Single Lens and Wide Aperture

4.4.5 The Lens and Aperture System

The beam expected to hit the board must be at some considerable size compared to the diode's regular output. This means we must expand the beam and resize it in a semi collimated fashion. What this means is that while the beam does not have to be completely collimated when it hits the board, it should be within a healthy range to cover at least the full width of the target, and preferably large enough such that an off centered hit is still registered and calculated. To manipulate the output beam created by the laser diode, we send it through a lens and aperture system outlined in Figure 4.3.5.1. It should be noted that in Senior Design 2, we adjusted this design such that the aperture is actually part of the second lens.

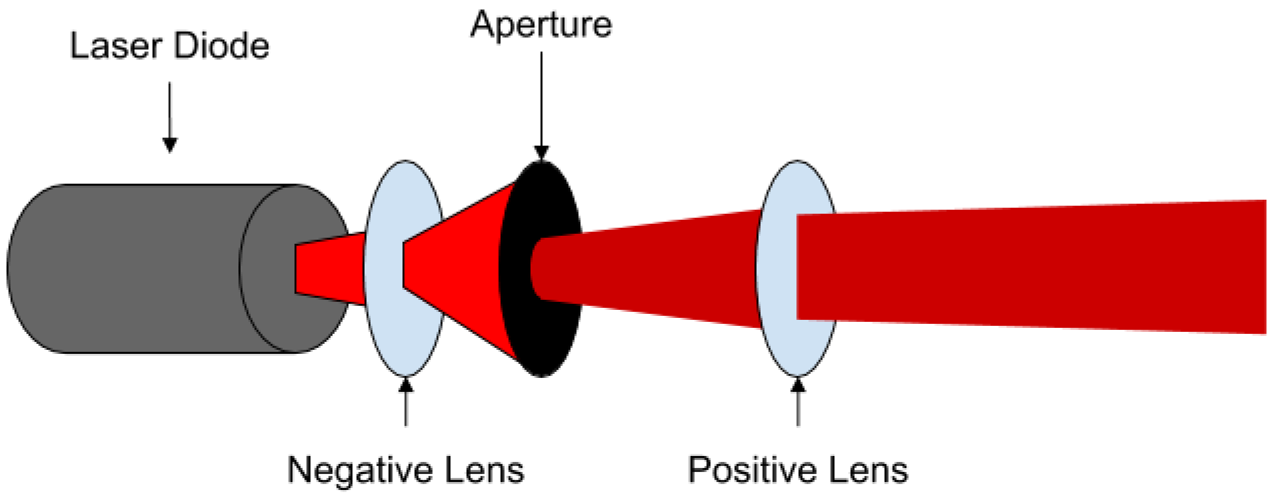


Figure 4.4.5.1: Expected Lens, Laser, and Aperture System

As we only want to sample the middle portion of the intensity profile of the laser, we can block out the portions we don't want with an aperture. The beam, as with all semiconductor diodes, will diverge upon leaving the chamber. However, we can expand the angle of the beam to be larger to increase the beam radius with a positive lens to a more controlled and healthier angle through the barrel. Then, while the beam radius is expanding, we place an aperture so that the unwanted light is filtered out. A negative lens is then used to decrease the angle of the beam and focus the filtered beam out to a more tolerable 50 mm radius at the board.

We could place the aperture before this negative lens, however, at a small radius, we would start to see some unwanted divergence and diffraction from how small the aperture would need to be. To calculate the diffraction angle from the small angle simplification version of the Airy diffraction disk equation.

$$\theta = \lambda / D$$

Equation 4.4.5.1: Airy Disk angle approximation formula

Assuming our aperture is .5mm, we can calculate our angle of divergence as 0.0013 radians. However, by placing the beam in front of the lens and increasing the radius to 3mm, we can reduce that angle down to an almost negligible 0.000217 radians. A much wider aperture will also reduce the associated diffraction to be negligible as well. This is how we sample the middle intensity of the beam.

Alternatively, we could also place the aperture after the second lens. This, however, would require the aperture to be a fair bit larger. The main issue, however, is that we can create a shorter and more form fitting gun barrel if we do not use anything in front of the second lens.

The other bonus of expanding the beam is that the intensity of the beam per unit area is much less. We can test the exposure of the beam with a power meter to determine how eye safe the beam is after it leaves the barrel of the gun.

4.4.6 The Gun Casing

The gun casing itself is 3D printed. The reason for this is because there's a lot of freedom involved in 3D printed involving spaces needed to fit parts like the microcontroller, but it also holds too little accuracy alone for an optical system to be set up like the one we require. We can however, use a few tricks to make the 3D printed model to have the accuracy we need. The one that is used is the use of set screws in the model. By drilling and tapping holes in a model around where an optical piece will go, the set screws can be used to align and stabilize the component. This is done by surrounding the lens, diode, or aperture with 2 sets of 3 set screws, each set screw arranged 120°. Each set screw can be adjusted independently to align one end of the optical piece across a plane perpendicular to the path of the beam of light. An example of this can be seen in Figure 4.4.6.1, which shows a 3D model to be printed. Inside this model, we can see the 3 holes to place set screws in to hold a lens in place, meanwhile a larger hole at the base is used to secure this lens mount to an optical system.

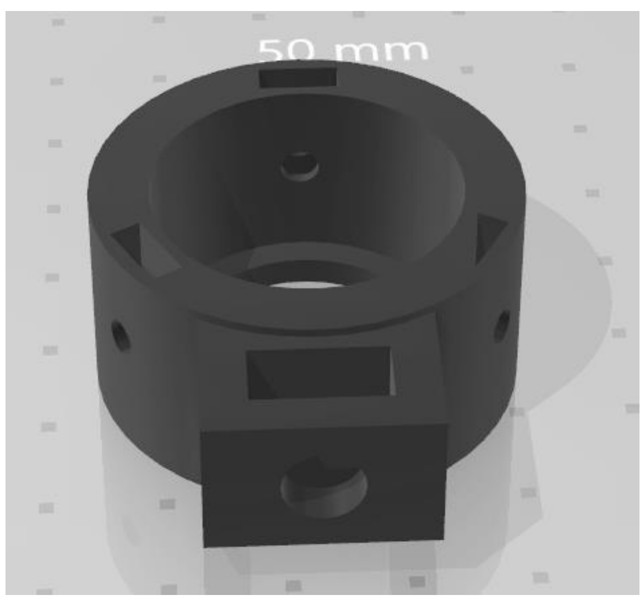


Figure 4.4.6.1: Open Source STL of Lens Mount With Set Screw Holes

There is one issue with using a lens mount system like the one above, and that is that it requires drilling and tapping of a plastic part. This isn't inherently an issue, but if we were to use a M2 size set screw, the required resolution of the print may be too much for the printer to use. In such cases as these, we are using heated press inserts. Heated press inserts are brass cylinder-shaped objects with a thread through the middle for a screw to be placed into. By making a wider hole and melting a heated press insert into it, we can secure a part with greater strength and stability than if it were simply drilled and tapped into plastic. This melting process is done through a soldering iron. We can see an example of heated press insert in 3D printed plastic in Figure 4.4.6.2.

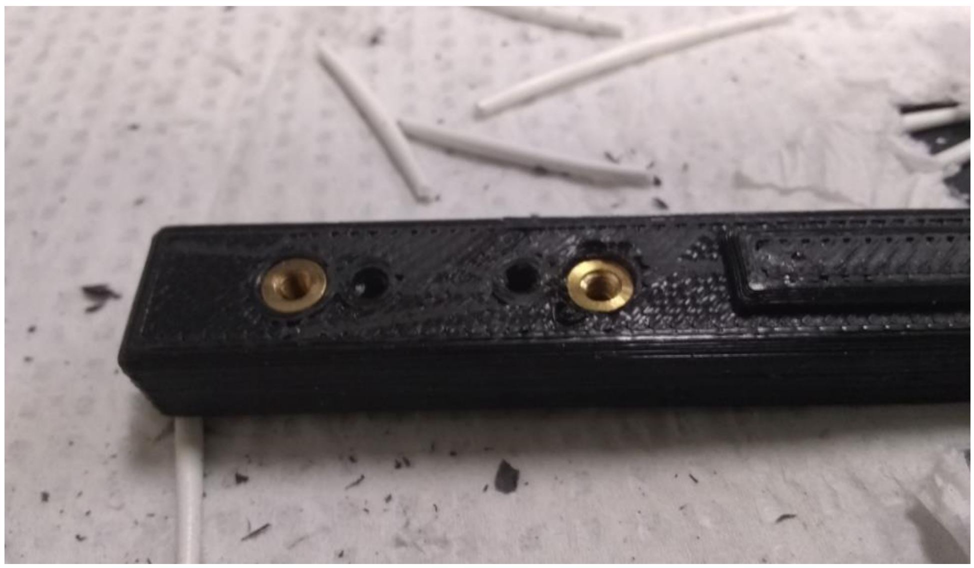


Figure 4.4.6.2: 3D printed part with heated press inserts.

4.4 Target Board System

The following sections are going to go into further detail on the target board system, including the board material, stand decoration,target design, and the LED response system. There are a lot of functioning parts of this target board that all needed to work together. Each of the different components needed to work together to make sure our project is fully functioning. The LED rings around each system have to work with PCBs, and programmed so that they are only lit up one at a time. These also have to communicate well with the photoresistors that are connected to the PCB of that target. The photoresistors are set to not take any input when the LED is not lit up so that the player cannot shoot away at any target and ensure the user is shooting the correct target. Once the target stays lit for 2 or 3 seconds, another LED lights up and in turn those photoresistors will now be turned on and be able to read a laser shot.

4.4.1 The Board Material, Stand, and Decoration

Creating the game board we chose to use materials that is not only be practical, but cheap and have an attractive, simple design. A list of how the board is put together along with the materials used for each target board element:

- The hull backdrop/Target holder and stand is going to be made out of mainly thick plastic and wood.

- The targets themselves are made out of a thinner type of plastic, allowing some malleability.
- The inside of the target hole is a short plastic box with a light filter at the opening.
- The printed circuit boards are screwed (metal screws) to the back of the hull, with jumper cables attached to the microcontroller.

The game board should be painted a dark color, this makes the target that is lit up easiest to see, even in daylight. Along with that since the game board is made out of smooth surface plastic; Adding custom decorations via stickers or painting is easy.

4.4.2 The Photoresistors

Photoresistors can be made of several different materials, however, not all of them are as sensitive or responsive to the same wavelength. Since we are targeting the 650 nm wavelength, we need to choose a photoresistor that will match it. A good choice would the the CdS photoresistor. It's peak sensitivity isn't near the 650 nm mark we are looking for, but it has better sensitivity at the 650 nm wavelength that several lead based photoresistors do at their peak. We can see the relative response of a CdS resistor in Figure 4.4.2.1. However, this wide range of response leads to another issue, which is that ambient light that is not 650 nm will still heavily impact the signal received by the photoresistor. This however, can be rectified by using a filter that blocks all but red light to the photoresistor.

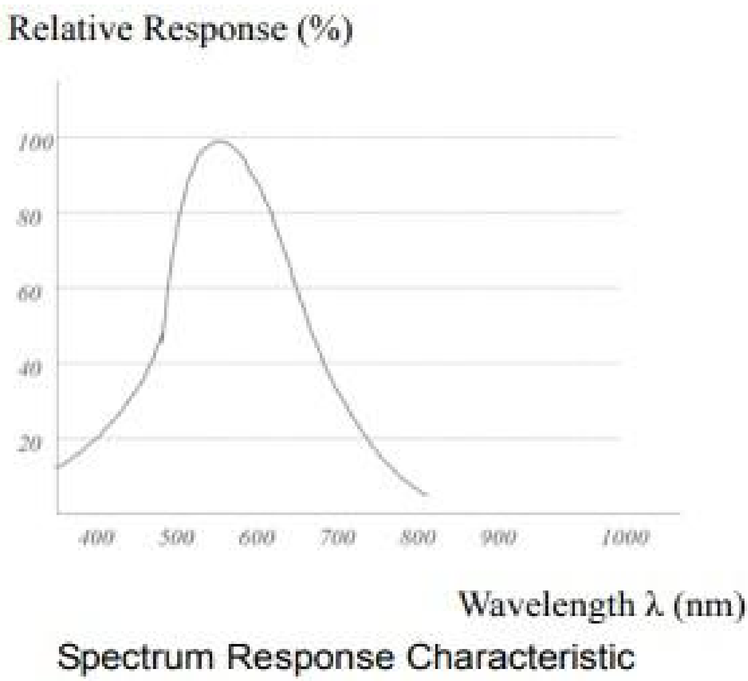


Figure 4.4.2.1: Sensitivities of the CdS resistor from the CdS Photoresistor Manual, Senba Optical & Electronic Co., LTD [5]

For this project, a large set of GL55 type photoresistors were purchased. This is in case a variable needs to be changed to adjust for the sensitivity, but a GL5528 photoresistor was selected. This photoresistor is rated for 150 V and a max power of 100 W, so it is well within the tolerance for the voltage and current being applied across it. At 10 Lux, the resistance is 10-20 KΩ, which should be approximately the Lux received by a shaded target indoors. By aligning this target in series with another 10 KΩ resistor and measuring the voltage over either the resistor or photoresistor, we can receive signals based on how much light is being cast over the photoresistor. Using the projections seen in Figure 4.4.2.2, we can expect the received incoming light to between 10 to 50 Lux, and as such, should change resistances between 10 and 3 KΩ.

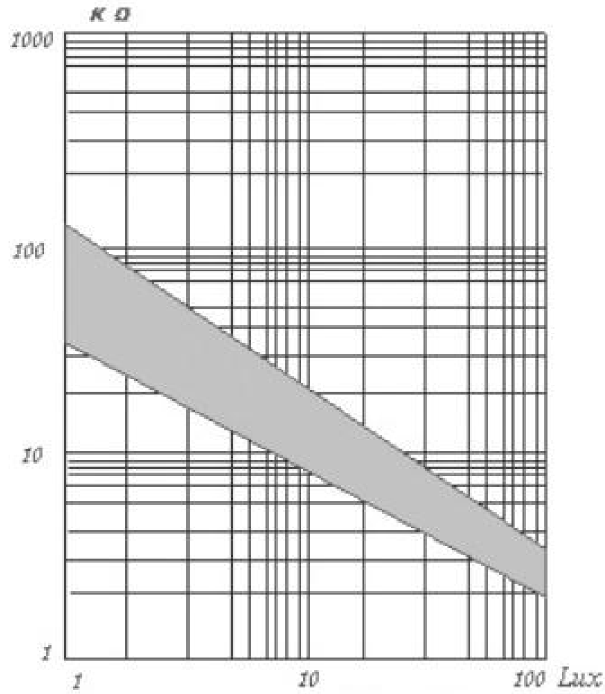


Figure 4.4.2.2: Relative Resistance of a GL5528 Resistor From th CdS Photoresistor Manual, Senba Optical & Electronic Co., LTD [5]

4.4.3 The Target Design

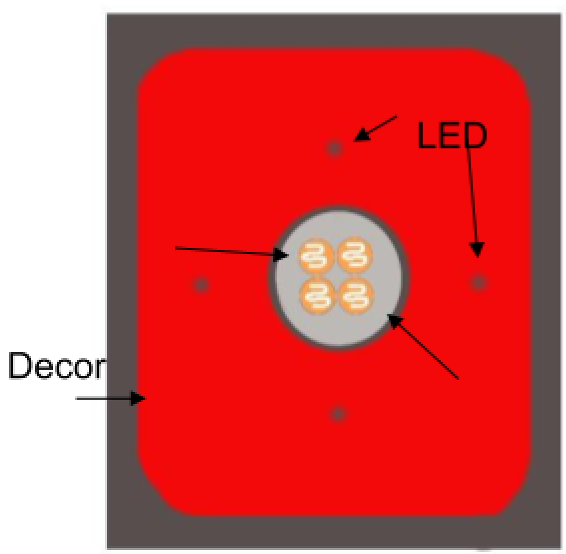


Diagram 4.4.3.1: Original Target Design

Some aspects of the detailed target design are important to note. In the diagram above there are four main elements to the target, that being the LEDs, photoresistors, light filter, and the plastic casing. The LEDs are placed in holes cut out of the plastic about an inch above the target hole. As mentioned in section 2.2.1.2 the light filter is in a hole with a diameter of about an inch, located dead center in the middle of the thin plastic casing. The LEDs are selected such that they contrast with the board, however the target is a darker shade of the LEDs color to maximize visibility granted the game is being played in a well lit area, targets are still easy to see with a bare eye. Each of the photoresistors should be connected to a 10 KΩ resistor in series with the voltage measured across each resistor. As the amount of light hits the photoresistor increases, the resistance will decrease, causing an increase in voltage and current over the 10 KΩ resistor. This can be done by supplying a bias voltage of 3.3 V, or in the case of Senior Design 2, 5V across each resistor/photodetector in series with a reference cable between the two, connected to one of the microcontroller's I/O pins. These pins should be able to pick up the change in voltage and convert that to a readable, usable signal. These pins should be mounted into a printed circuit board that attaches to a small 3D printed plastic tubing. The printed circuit board designed for this system and its associated wiring diagram can be seen in Figure 4.4.3.2 and Figure 4.4.3.3 respectively.

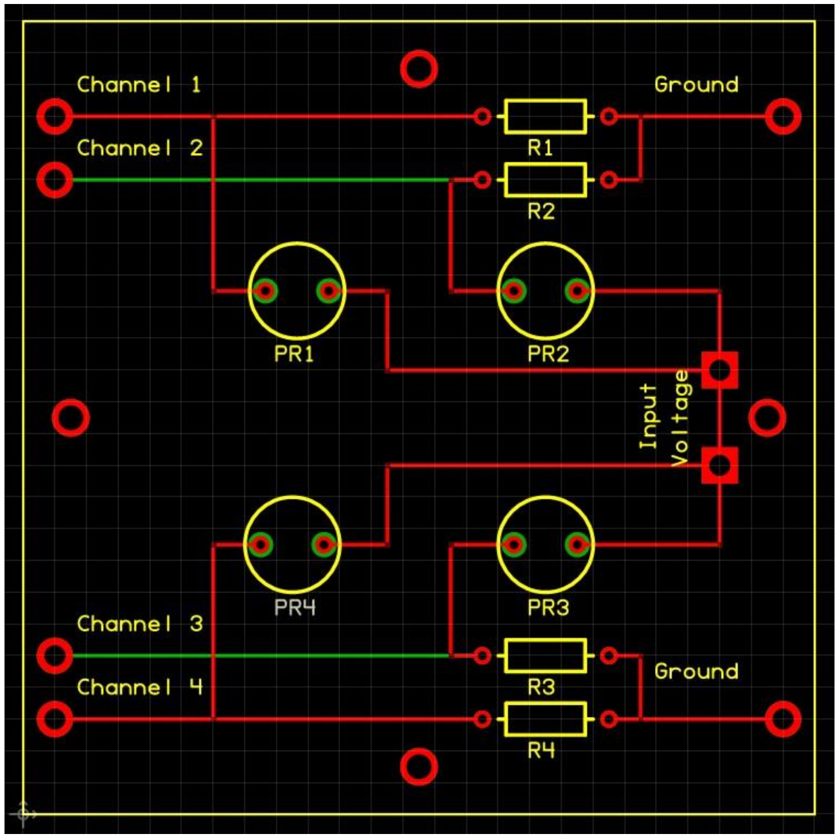


Figure 4.4.3.2: First Design for Target printed circuit board Schematic Designed in PCB Express

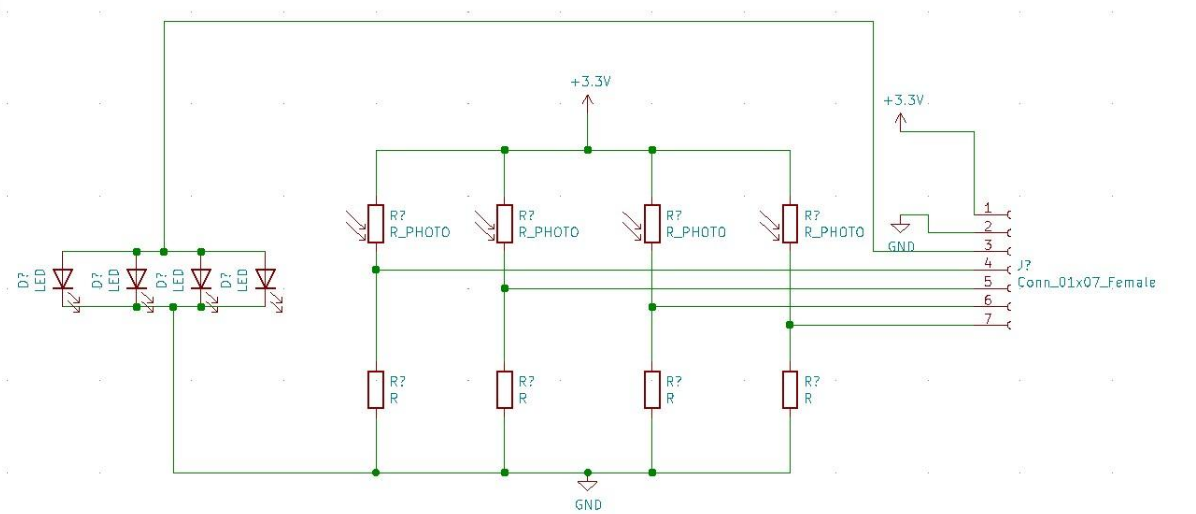


Figure 4.4.3.3: Original Wiring Diagram for printed circuit board Designed in Eagle

Please note that in Figure 4.4.3.3, there's a seven-conductor cable running to the printed circuit board, represented in the middle. There are more wires that should be shown.

The figure above is a schematic that can be given to a printed circuit board maker, created, and delivered to us. The dimensions of the printed circuit board are 2.5 inches by 2.5 inches and is mounted to the target via the four holes. One on the top, bottom, left, and right of the board. These holes are approximately 2 mm in diameter and is used to be mounted on the printed plastic target piece, modeled and shown in Figure 4.4.3.4. In this scenario, it would be difficult to use heated press inserts unless the model width was expanded. However, since the target is not something meant for mobile usage and is intended to stay stationary, a simple drill and tapping of the component should be good enough to secure the device.

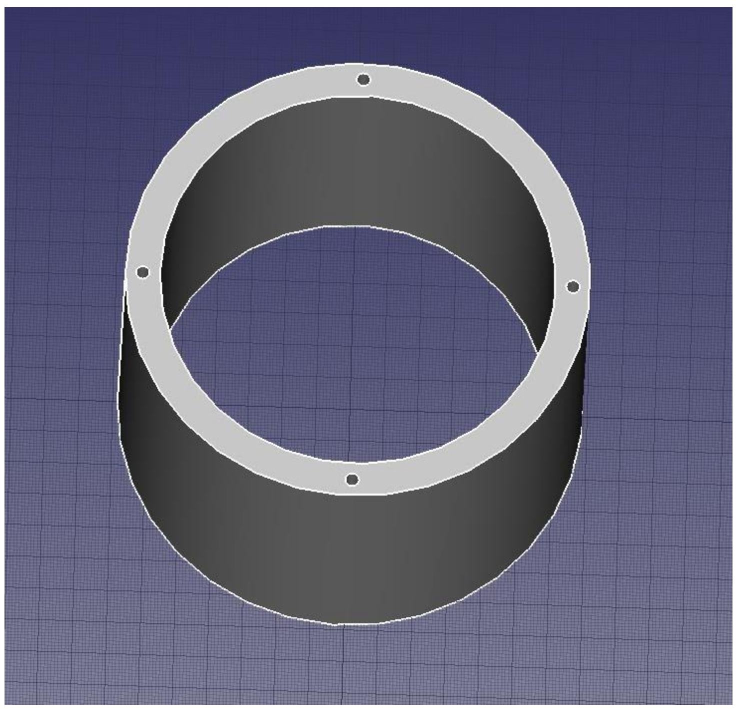


Figure 4.4.3.4: 3D Model of the Plastic Component of the Target System

Another way to see this how the rays work is through ray tracing. As seen in Figure 4.4.5.2. What is more distinct here is how light is directed on a more directional level. Consider the max possible angle of which the beam of light can leave the laser diode. This beam of light will leave the diode, then be directed at a harsher angle as it hits the positive lens. This beam will just graze the largest opening of the aperture, then continue on through the negative lens. The negative lens will then reduce the angle of the beam. The end result is that the beam has been expanded and trimmed through the system. While not very complicated, it is important to note that unlike many other optical systems, there is no imaging that is to be done. Instead, this works like some common telescopes do, enlarging the image, but not focusing the rays to a single point. Instead, we want to rely on emission shape as part of our project.

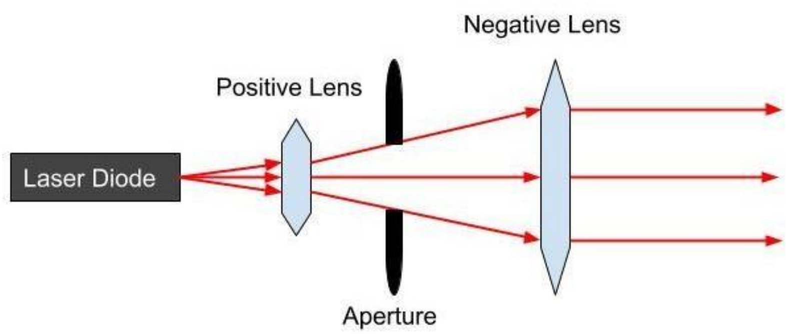


Figure 4.4.5.2: Ray Tracing Tiagram of Lens, Laser, and Aperture System

The types of lens we used specifically is also sold by lights 88. These lenses are 7 mm in diameter and have a thick edge. These thick edges are important as we need to be able to securely mount them to the inside of our device. Additionally, we can also find the aperture sold by the same company. The lens can be placed inside the aperture. We can see these lens and aperture in Figure 4.4.5.3.

Figure 4.4.5.3: Plastic Coated Collimation Lens for Laser Diode and associated aperture

4.5 Wireless Microcontroller System

The microcontroller used in this case is the Arduino Mega. This is due to the 54 pins that the Arduino Mega has. At best, each target has 4 signal wires, and with 5 targets, we are looking at 20 input signals alone. This is not including the incoming power, nor resistors that have to go in series with the photoresistors.

The signals read by the arduino can be read through the function analogRead(). This works by outputting 0 when 0 V is read and 1023 when 5V is read. Any value in between is represented by an integer between 0 and 1023. From this, we can expect an increase of an integer value to a threshold value to count as a hit. From then on, we can sample the peak threshold for four signals associated with said target. These values are to be stored and analyzed for points.

In figure 4.5.1 the printed circuit board that connects to all of our targets is seen below. This allows us to read all of the inputs from the photoresistors attached to each of the targets, as well as supply the 5 volts needed for the LEDs on each target board. This alsos allow for the data to be sent to our application by attaching the Bluetooth chip of our choosing to the board. Our board is also able to read all of the inputs necessary from the photoresistors varying from 0 to 1024.

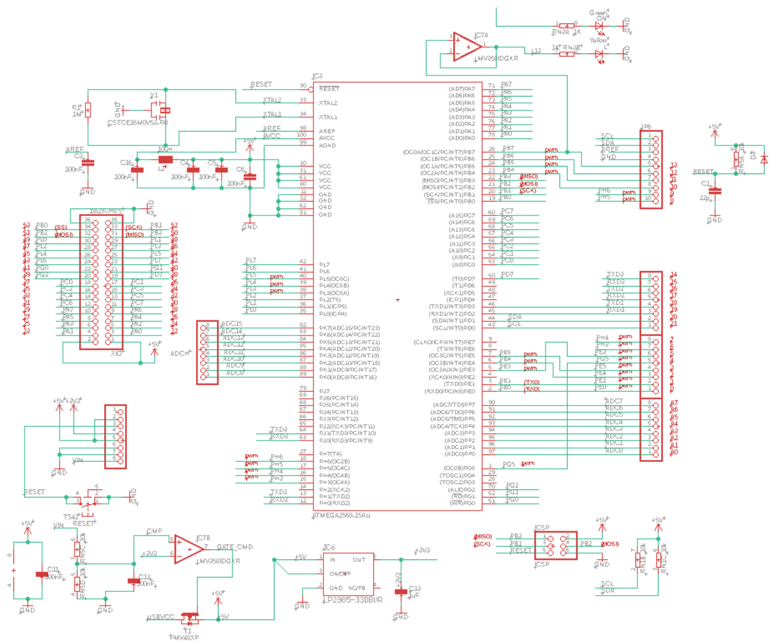


Figure 4.5.1: Printed Circuit Boards Connected to Targets

After these points of data are collected, we can compare these values as signal1, signal2, signal3, and signal4. It is likely that the laser output may have a wider intensity distribution than the horizontal distribution, as that tends to be the nature of semiconductors. As such, we compare the x axis and y axis separately. To compare the x axis, we can average the left two inputs and right two inputs, then compare the values. Similarly, we compare the top two inputs and the bottom two inputs to compare the y offset. Thusly, we can calculate the score

$$Horizontal\ Component = \frac{(averageLeftSide + averageRightSide)}{(MaxAverageHorizontalValue * 2)}$$

$$Vertical\ Component = \frac{(averageTopSide + averageBottomSide)}{(MaxAverageVerticalValue * 2)}$$

$$Score = 500 * (horizontalComponent + verticalComponent)/2$$

We built our printed circuit board designs in Eagle. Using Eagle to build our boards makes it very simple to order these PCBs through any company we like very easily. We can design the board to meet our specifications, once designing the board that we need we are able to order from any company we like, just having to export and import the files we have created into the company's website. We can get a quote based on the board that we build in this software. There is an order board option directly located inside of this software so we do not need to export the schematic we design to anywhere else.

We are building 6 printed circuit board boards, 4 of which all are the same, using the design seen in section 4.4.3. The other printed circuit board design is connected to all of the other boards. We have built the schematics for these printed circuit boards and plan to order these boards by the end of Senior Design 1. This gives the board time to be built and delivered to us, allowing us to begin putting our project as a whole together.

We used the company JLCPCB, this company allows for easily ordering multiple PCBs. Using their website, we can easily import the files we created in Eagle to this website. Since we needed four PCBs , connected to each target, this company allows for us to order all of these PCBs for cheap. This company also solders the smaller components on to the board for us. Since none of our group has great experience with this, this helped us build these boards without the issue of not knowing how to put on these components. They also do this for free, this keeps our costs low and makes it so we do not have to add this to the overall cost of our project.

4.6 Mobile App

The mobile application is in charge of enhancing the users experience. Users have immediate access after installing the mobile application to either an iOS or Android platform device. Upon opening the application users are directed to a login page as seen in figure 4.6.1.

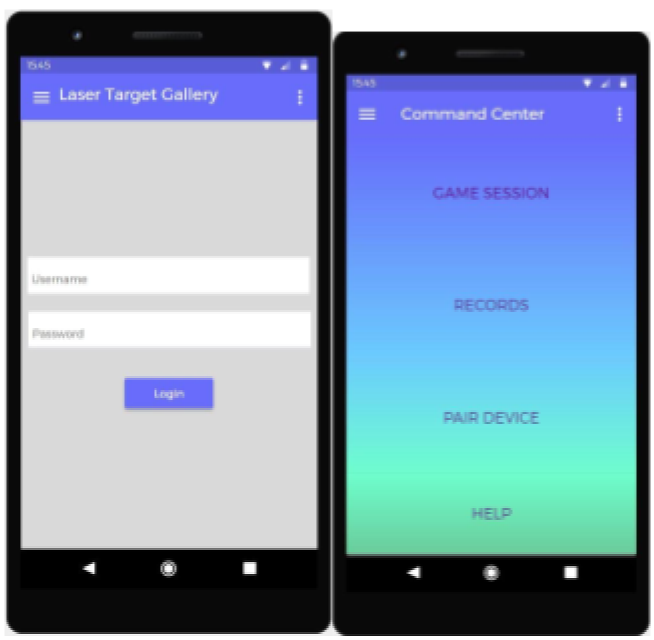


Figure 4.6.1: Login Screen (left) and Home Screen (right)

New users will need to create an account before allowed access to the home screen. Users who already have an account is prompted to enter their credentials. The submitted data fields will undergo an authentication process to match the correct account to the user. Users are directed back to the login page if no account matches the entered credentials. Once authenticated, users are directed to their own personal home screen. A quick and helpful menu will pop up if an authenticated user has no prior data records.

This menu will teach the user how to navigate the application for future sessions. Users who already have data records are navigated directly to their unique home screen. At the home screen, users can choose to view previously recorded game sessions or start up a game session. To further visualize the desired goal figure 4.6.1 shows an example of a typical user home screen and Figure 4.6.2 provides a flowchart to the process described above.

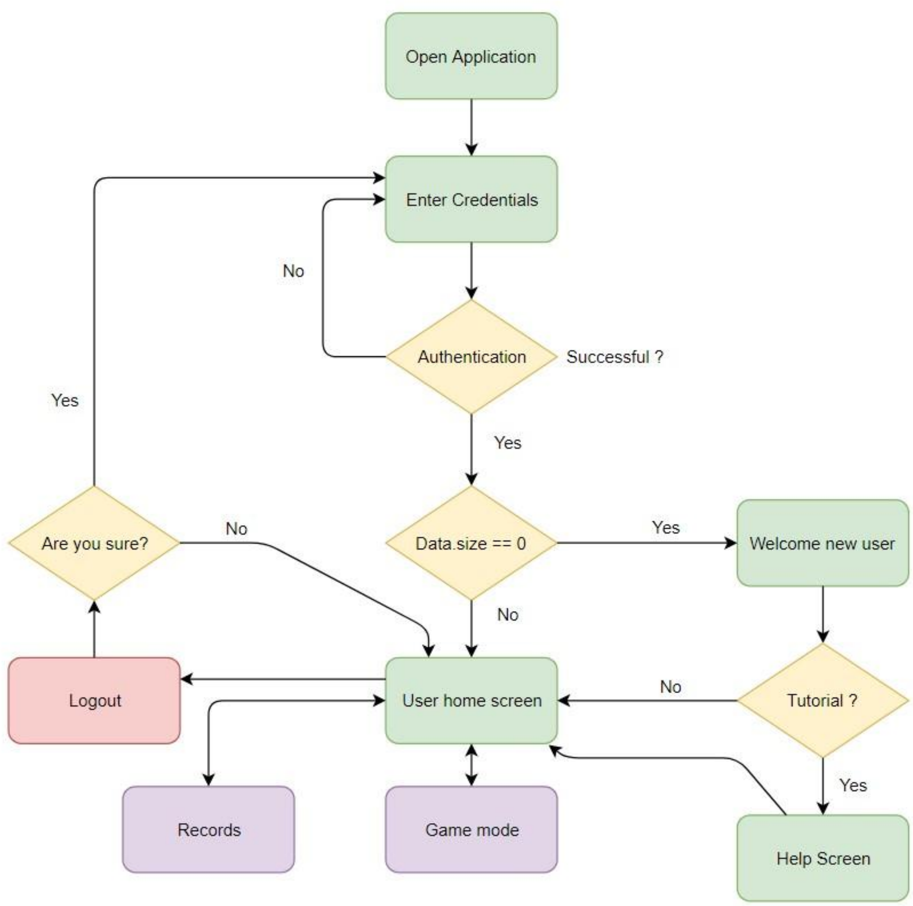


Figure 4.6.2: Mobile Application Navigation Flowchart

When starting up a game session, the mobile application will first check to see if it can detect and pair with a game board nearby through Bluetooth. If the application cannot detect a game board an error message will trigger to notify the user. Then the application will back out to the user's home screen.

Otherwise, the application will pair with the game board. In cases where there are multiple boards detected in the area, the application will prompt the user to select the desired game board. The game session will start once the application receives confirmation from the user and detects that the two devices are paired.

During the game session, the sensors on the board are actively receiving data. This data varies based on the proximity of the laser pulse in relation to the location of each sensor. The data is then transmitted from the Bluetooth microcontroller to the phone application. This process is looped for the entirety of the game session. Each packet of data transmitted is detected then parsed according to the specific guidelines that our team has set.

Finally, the data is inserted into the local database and stored on the user's mobile device where it is analyzed. When the timer for the game session ends the user is directed to another screen that tallies the final results of the session.

In order to replicate this process we must design a database that allows these processes to interact with each other. We again start from the top of the flow chart. In order for the user credentials to be stored we must create a class that holds attributes such as a username, password, and a userID. Some restrictions we must put in place are that no two users are allowed to have the same username. Also, passwords must contain some form of entry. User should be able to authenticate and login to their home screen.

Each user must have one and only one home screen. Home screens serve as the main navigation menu for the mobile application. They must have the capability to create many game sessions and hold many recordings. To make sure that each home screen is unique we must provide each home screen with its own personal ID.

Game sessions need to be able to locate and pair with our game board. Each game session has its own unique ID and store multiple variables that are passed from the microcontroller. Additionally, each game session must be able to record the data that it has been transmitted and create a record.

Records contain all the variables detected during a game session. Records must have a one to one relationship with game sessions. In each record there is a unique ID, a date, a score, and a string of data depicting laser pulses detected from the board sensors.

Below, Figures 4.6.3 and 4.6.4 shows a flowchart which showcases the logic followed in this process.

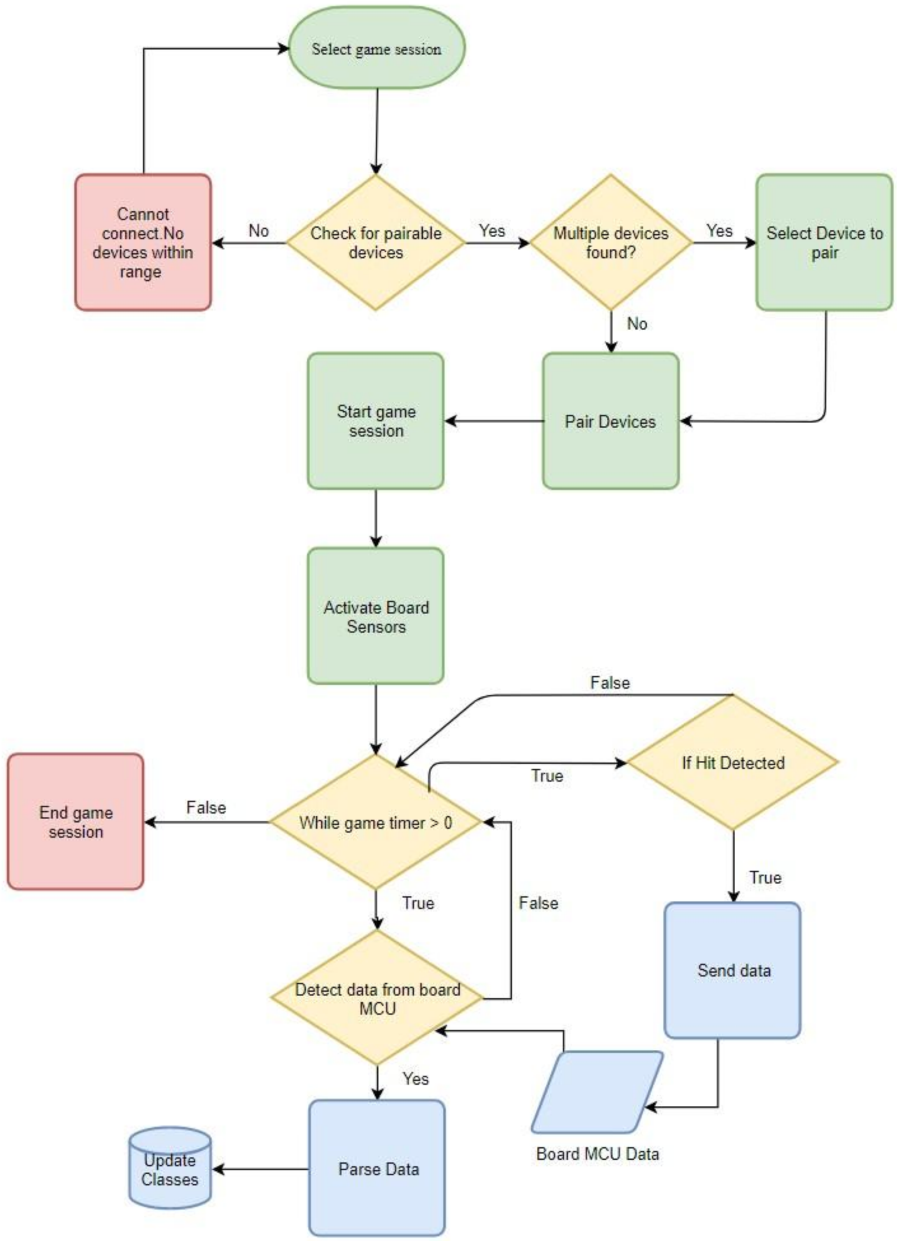


Figure 4.6-3: Mobile Application Game Session Flowchart

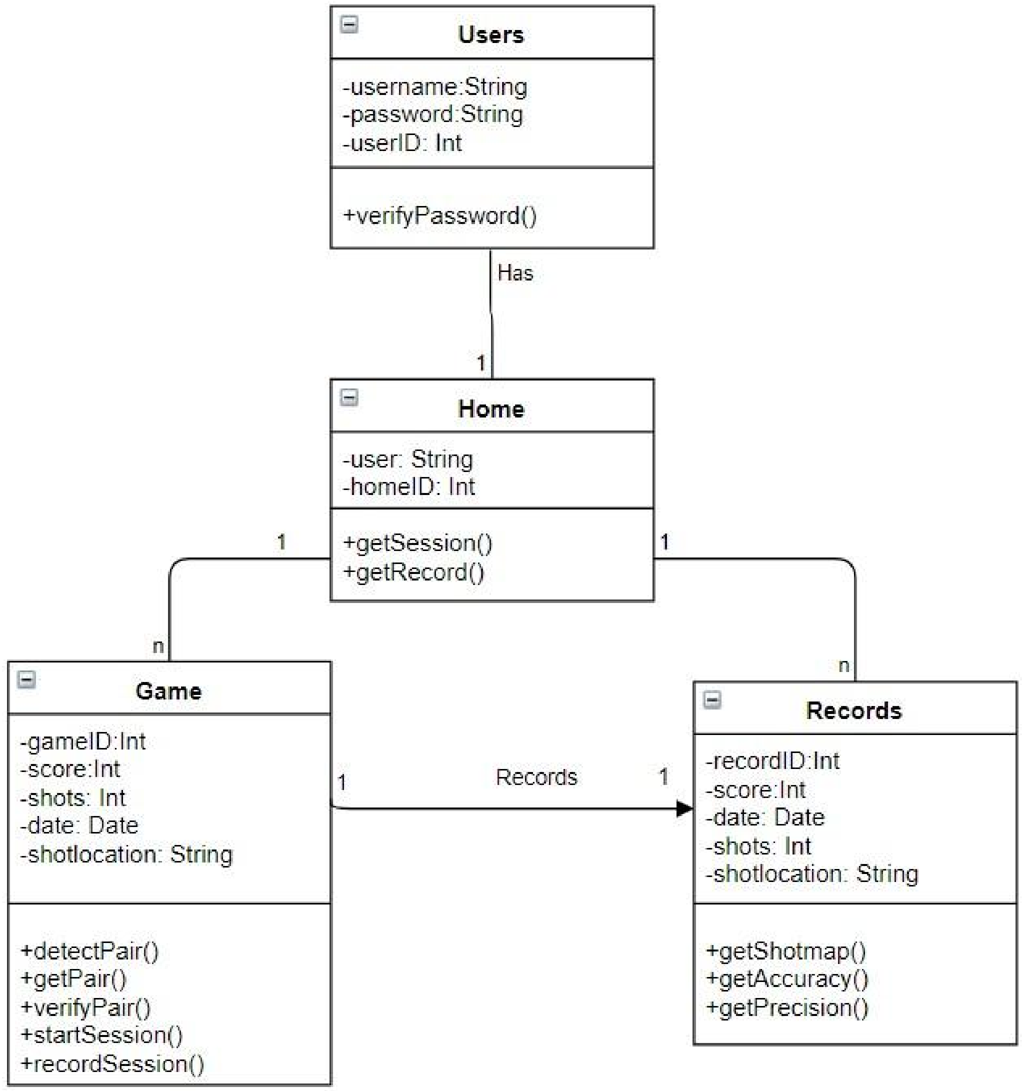


Figure 4.6-4: Mobile App Class Diagram

5.0 Design Summary

This section does a quick summary on what each part of the project's goals are, how those goals are achieved, and how these systems work together, as well as how these goals are to be accomplished through the construction of each part. There are many individual parts in this lab and we will take a look at how each individual part of this project is integrated together.

5.1 Laser Gun System

The laser gun itself is designed to put out a 50 mm beam of light at the target. This is done by using a lithium ion rechargeable battery pack, a microcontroller, and a laser diode and lens system. The goal is to have the gun operate properly and efficiently and by that we mean a reasonable amount of game play time on a single charge. To achieve this goal we need to minimize current draw from the guns component devices and install a battery that can hold the most charge and still properly fit in the gun.

Another goal we have with the laser gun system is to have some restrictions on the trigger for gameplay purposes. To accomplish this we need to add a script to the Arduino Code that only takes input from the trigger in interval times, for example a timer that counts down after the trigger is pulled, and while the timer is active, the microcontroller does not receive any inputs from the trigger switch.

One of the stretch goals is a noise the gun makes when fired. This is an active buzzer, allowing for a noise to be created each time that the trigger is pulled. Using an active component instead of passive, allows for us to only need to supply a DC voltage to the buzzer. The active buzzer comes with an internal oscillator making it very simple to use by simply connecting it to a 5V output of the attached microcontroller.

The final key functionality is a stretch goal, which is connecting the Bluetooth module to the phone web application of the laser gun system. The goal is to send data via Bluetooth to an application to be stored in a database, using MySQL. The data being sent needs to be manipulated in order to calculate analytics of gameplay. Making this possible is going to need to be done by sending information as organized as possible to make is easy for the web app to receive and store. Using registers, we can store information for the game such as the amount of times the gun was shot in the time of a single game. This can be factored into the amount of times the laser diode gets activated, all of this information is how we can calculate different analytics and display them on the application.

5.2 Target Board System

The target board system's main functionality is to capture and send data to the wireless microcontroller, as well as output information to the viewer. By housing a set of 4 distinct LED arrays which can be controlled independently, the board can display to the player which target to shoot at. By housing 4 plastic squares with a printed circuit board mounted to the back of each square, we create a chamber of which the laser is to be fired into. On each printed circuit board is four photoresistors. Each photoresistor is in series with another resistor, with an analogue voltage taken across the resistor. This board and LED array is wired to the Wireless Microcontroller System, controlling and reading the data from each printed circuit board.

The secondary objective of the Target Board System is conveying visually what is happening to the player. LEDs flashing on hit, a piezo speaker going off when a target is hit, the contrast between the laser and the color of the board all contribute to the player knowing what is happening with the game without having to think about it. Optimally, the player should understand what the game is about without having to be explained what it is. This is done through vibrant carnival style colors surrounding each target.

5.3 Wireless Microcontroller System

The microcontroller system is the brains of our project and handles almost everything when it comes to the gameplay. So, our goals for this microcontroller is to have it be directly connected to 5 printed circuit boards and should receive input from all of them, and send output to surrounding LEDs and data from registers via Bluetooth.

To accomplish that goal, we are going to have to custom build 5 printed circuit boards all with the same ramifications and have jumper cables connecting everything together so power can effectively flow to all 5 of the surrounding devices along with being connected to the microcontroller input pins. For the LEDs, a similar strategy is used, we need to use cables to connect to the output pins of the Arduino, to 5 different LED strips to be activated appropriately when it comes to gameplay, this is important to the game and going to need to be coded accordingly.

When the target LEDs light up this is the target that is needed to be hit in order for the photoresistors to register the information; Furthermore, the LEDs, and respective printed circuit board need to all activated (be powered) at the same instance.

The goal of this microcontroller is also to be responsible for sending Bluetooth data to the web application. As mentioned earlier (section 5.1) we are going to hold information from the input pins into registers; The Arduino Uno microcontroller has 16 input output pins meaning we have enough to take input from all 5 targets at once if necessary. At the end of the game timer, another process is executed which is the process of sending the information stored in the registers to the Bluetooth chip and furthermore to the web application to be stored in a database.

A basic look at how the pins in the Arduino are set up:

Table 5.3.1: Arduino and Bluetooth connection pins

Arduino Pins	Bluetooth Pins
Rx (Pin 0)	TX
TX (Pin 1)	RX
5 V	VCC
GND	GND

An LED strip for example can be connected negative to GND of arduino and positive to pin 13 with a resistance valued between 220Ω – 1KΩ.

5.4 Web App

This section describes the two main parts of the phone application. The database is how the information is stored, organized, retrieved, and added. The user interface describes how one who uses the product will experience the app, including how the app is understood by the user.

5.4.1 Database

The database acts as the supporting pillars for the mobile application. For the purpose of this project, our team needed a database which would be both compatible with the react native while also providing quick queries, stability, and security. The results of our research allowed us to choose RealmDB.

The database needs to have the ability to store user credentials, game session data, and data records of previous game sessions. To do this we created distinct classes that house the proper attributes required for each specific function of the mobile application to work correctly. Next, we must set the standards for the interactions a user has with the mobile application. This means that we must map out how the three main functionalities of the mobile application interact with each other. As of now, every user must have their own unique home screen. The home screen has the authority to host multiple game sessions and view multiple records. Lastly, Each game session store the data collected into a unique game record.

RealmDB allows our team to create a locally stored database which is both responsive and stable. RealmDB uses object-oriented based design structure that improves the query speed in comparison to a typical relational database such as MySQL. Because data is streaming in at a fast pace during each individual game session it is important that our database is capable of updating both fast and reliably.

5.4.2 User Interface

We want the user interface to make the game enjoyable for the user and make it so that the user will want to play the game. We found a color scheme that captures our game well and is also pleasing for the user. The interface is very important because this is all that the users will see. It can cause the player to prejudge the game in a good or bad way. Making it very important for us to make this pleasing for the user. We made it so that the user has to create an account first, ensuring their data is saved to the database. Once creating an account or logging in, the user can choose to begin the game. With a button on the screen we designed to be easy for the user to find and press.

The scores of each shot are displayed in large numbers so that the user can easily see how accurate their shot was and how many points they received. The user interface is tested by a group of test users we have first play the game. This will allow us to get good feedback from those who are not part of our development team. This is important, although we may believe our design to be pleasant for the user, we needed actual user feedback to ensure that we have the best display possible.

6.0 Project Prototyping

In this section we will take a look at project prototyping. We will discuss how we are building our prototype. We will look at how each individual part of the project would be put together, and then how we will construct the project as a whole. We will take a look at the overall cost of the project as well as building the first prototype of our project.

6.1 Bill of Materials

In this section we show our Bill of materials. This is a current list of materials that we will need to begin building our project. As the semester goes on, we updated this table and add in the materials we needed.

Table 6.6.1: Bill of Materials

Module	Item	Cost	Qty	Sub-total
Game System	PCBs	$8.99	5	$44.95
Power System	Rechargeable battery	$4.00	2	$8.00
Power System	Wires and Components	$8.99	1	8.99
Power System	Photoresistors	$4.99	30	$4.99
Power System	Bluetooth Chip	$2.07	2	$4.14
Microcontroller	ATMegas	$14.99	2	$29.98
Game System	3D Printed Laser Housing	$29.99	1	$29.99
Game System	Thick Hull Plastic	$29.99	1	$29.99
Game System	Thin Target Plastic	$4.99	5	$24.95
Power System	Laser Diode	$9.99	2	$20.00
Game System	Light Filter	$0.49	5	$2.95
Game System	PVC Pipe	$0.79	5	$3.95
Accessory	Individual LEDs	$0.22	20	$4.40
Accessory	Paint	$19.99	1	$19.99
Total Cost	$237.25			

6.1.1 Target Board

After doing some research on good types of wood to use, we decided to use cedar wood for our target board. Cedar wood has a specific gravity of .47 and a bending strength of 8,800. This makes it a very strong wood to cut and design as we wish. It is also light enough to be supported by the legs we built to make the board be at about shoulder height. This will make it easier for the player to not have to bend down aim to hit the target. We drilled 4 holes into the board using a power drill available in the technology innovations lab located in the engineering building on the UCF campus, to allow for us to put our target inside of each of the holes. Each hole is about 3 inches in diameter. This allows for us to put our LED ring around each of the holes, and then have plenty of space to put our target with photoresistors attached as well as all needed PCBs.

We have the required tools to create and cut the board. We needed to purchase the necessary wood. Cedar wood costs anywhere 4.50-7.50 per square foot. This cost us approximately 20-25$ depending on the price at the location we get the wood from. Currently, we can purchase this from Home Depot for approximately $25. We painted the board to give it a fun design and good user game experience. We may need to hire outside help to ensure we get the best quality paint job possible. This added to our overall costs, but allowed for our game to have a much better appeal to it.

6.1.2 Lens and Aperture System

The lens and aperture system are tested at nScrypts Lab. In this lab, a table is used to align various components of the system in a row. Firstly, the laser diode is secured in place and outputted on a white sheet of paper to observe any premature issues with the diode. A positive lens was then used to focus the beam through a small aperture. It is also possible that if the lens is small enough, the beam does not need an aperture since the lens acts as such. After an appropriate beam shape is outputted onto the white paper, a final lens is used to collimate the beam. You can see an outline of this in Figure 4.4.5.1. Additionally, if a spherical lens is to be used, we can align this in the same optical system.

However, this does not directly translate to the 3D printed gun. What it does do, however, is give us approximate dimensions and locations of where to put each part of the optical system. This can be drawn and drafted in a CAD software, then 3D printed as secured in place using some of the techniques described in section 4.3 and its related subsections. The main technique to be used is the set screw casing as seen in section 4.6. These set screws have the advantage of translating optical pieces perpendicular to the direction the beam is going, helping to center the beam.

6.1.3 Laser Gun

Prototyping the laser gun required multiple steps, beginning with 3D printing out the gun mold, painting the hollow casing, installing anchors, installing the devices, and finished by clipping together.

In order to print the gun mold we are going to have to use a 3D modeling software, and there is no shortage to choose from FreeCAD, AutoCAD, Blender, Adobe Illustrator, etc. For the sake of making a precise design in order to match perfectly with the dimensions required to fit all the devices necessary, we chose FreeCAD for those reasons and we could save some money on an expensive piece of software.

FreeCAD is an on-premise, open source CAD software under the PLM, CAx and MCAD class, which uses parametric 3D modeling. It suits the needs of a wide range of professional and personal users, such as product designers, engineers, architects, educators, and hobbyists. The software helps users create detailed objects in a variety of specs. As a design platform, it is ideal for both small business and enterprise. For engineering, FreeCAD can be extended for a scriptable CAD solution, allowing it to perform complex designs, for instance, electrical schematics and architectural blueprints. FreeCAD can be scaled to the characteristics you presently need using a modular configuration. It is possible to extend the features via plugins. Likewise, to configure and expand capacities, you can access its Python Interpreter and internal and macro scripts. The main goal in using this program is to output STL files to later convert them into G-Code. This is done with a program called Slic3r, which traces an outline of the 3D model in the STL file and creates instructions for the 3D printer to build the part.

This software is used to create the gun in several steps, dividing once in half longways so each side can clip together, then split further into subsections. For example, the handle houses the microcontroller and power supply and be printed in two parts. The barrel of the gun is printed in a similar fashion, housing the optical system of the gun. The trigger system is similar except for the fact that the trigger mechanism itself was needed to be printed as a 3rd component between the two haves but can be used to bridge the sections between the barrel and handle of the gun.

As the gun is printed, there may be several issues. 3D printing isn’t a precise or perfect process. In fact, there are many common issues that occur in the process of printing. The filament can over-extrude, under-extrude, bridge, warp, clog, slump, miss the bed surface entirely, or any number of other issues. Some of these can be fixed manually either by adjusting settings in slic3r or by slipping off stray bits on your print. However, some features require high precision and will need more attention to. You can see some of these issues in Figure 6.1.3.1.

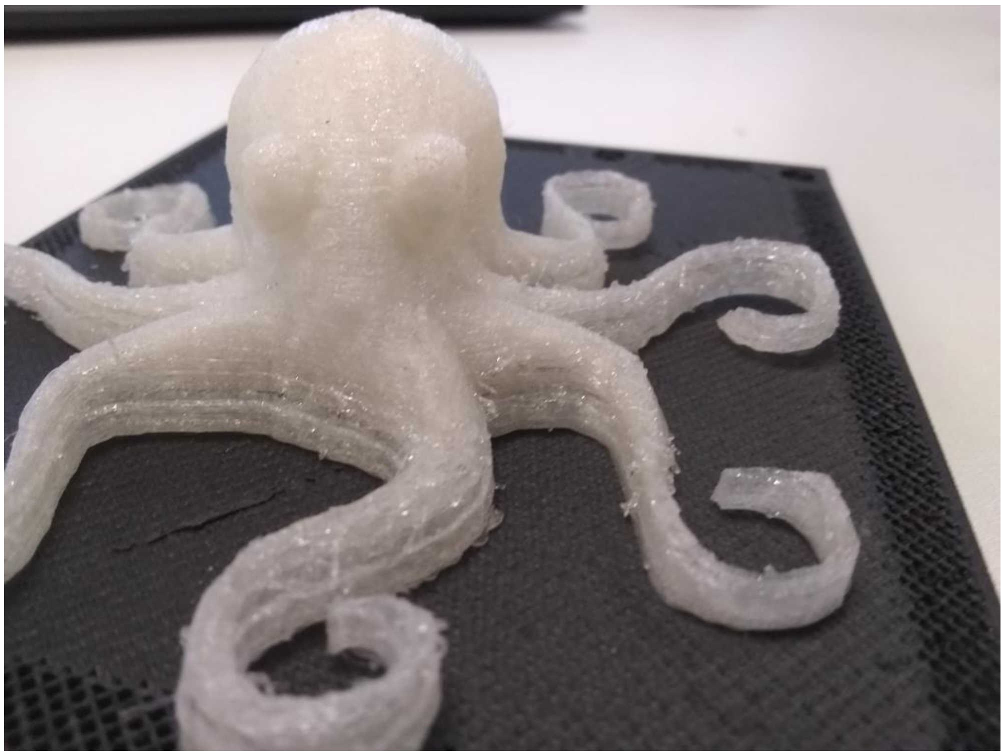


Figure 6.1.3.1: A 3D print featuring poor layer adhesion and several burrs

Painting the Casing of the gun was the next step, it needs to be done before anything is installed in the gun. First thing after the two hollow shells are printed we have to Wipe down the entire surface with rubbing alcohol and allow to dry, next Use a fine grit sandpaper to remove any and all shiny surfaces from the plastic. Lastly we spray a light, thin layer of paint. With this first layer, our gun should not be fully covered. Giving each layer a few minutes to dry before adding the next one. It should take multiple thin, even layers to refresh the guns's color without over-painting.

Installing the anchors was the next step, and require post 3D printing work. Heat pressed anchors is the main approach at connecting the anchor to the inside of the gun. An example of one is seen in Figure 6.1.3.2 Aforementioned in an earlier section we are using metal clips to secure the different devices in the gun. The main purpose of using clips inside the casing is to allow interchangeable parts. Allowing for easy changing of parts is beneficial for a couple reasons, Upgrading and replacing damaged or broken devices.

Figure 6.1.3.2: Metal Anchor Clip

A separate anchor piece was needed to be attached to the different devices allowing them to connect to the clips mounted in the gun.

6.1.3.1 Laser gun microcontroller

For adding the microcontroller and printed circuit board in the casing of the gun there is going to be a couple steps. First off we have to program the chip outside the device, then situate the printed circuit board and the wiring that connects to the external devices, install clips on the microcontroller, and lastly place the device in the gun.

Programming the chip outside of the gun is going to be the longest part of installing the microcontroller, any kind of programming is going to consume some time. To limit excess time on debugging we are going to follow these steps:

1. Write code in small chunks and test each of them.
2. Provide meaningful names for variables and functions
3. Use functions
4. Use constants rather than numbers
5. Write comments to explain coding choices for future reference

6. Make sure the code has a proper indentation and remains readable at all times.

Along with properly debugging understanding the IDE is going to be extensive time wise. In the Arduino Integrated Development Environment (IDE), Arduino programs are produced. Arduino IDE is a unique software operating on your machine that enables you to write drawings for separate Arduino boards (synonym for Arduino language program). The programming language of Arduino is based on a very easy language of hardware programming called processing, comparable to the language of C. It should be uploaded to the Arduino board for execution after the sketch is written in the Arduino IDE. The first step in Arduino board programming is to download and install the Arduino IDE. Windows, Mac OS X, and Linux run the open source Arduino IDE. Download the Arduino software from the official website (depending on your OS) and follow the installation directions.

After programming the device we need to simulate the effect on the respective devices and the printed circuit board. There are a couple popular simulation services offered for this exact reason. One of them being Electronic Lab on Tinkerboard; the platform makes it easy for users to simulate real-world electronics through a circuit simulator. This online simulator allows the user to drag in an Arduino board and start programming. It also offers the user a circuit diagram maker, multimeter measurement tool and oscilloscope. This Arduino simulator allows you to build your design from scratch whilst enabling you to take precise measurements of the power supply throughout your circuit. The first step in Arduino board programming is to download and install the Arduino IDE. Windows, Mac OS X, and Linux run the open source Arduino IDE. Download the Arduino software from the official website (depending on your OS) and follow the installation directions.

Next in the process of prototyping the microcontroller in the gun is appropriately connecting a printed circuit board to the microcontroller and preparing it to be placed in the gun. In the diagram it shows some wiring that is similar to how the prototype Arduino and printed circuit board looks, with wires tucked away in an organized manner such that nothing is hanging out of the casing of the gun.

Figure 6.1.3.1: Arduino chip connected to PCB

To simulate the real-world effects of the printed circuit board and the Arduino there is a piece of software we can utilize, the Virtual Breadboard. Virtual Breadboard has established a name for itself as one of the most powerful and advanced Arduino simulators available today. Virtual Breadboard has grown within the electronic circuit industry and today can simulate Arduino devices, Netduino and PIC microcontrollers. On start up the user has access to a complete virtual development environment where they can program directly to an Arduino board. It’s also worth noting that it can act as an AVR emulator as well. The first step in
Arduino board programming is to download and install the Arduino IDE. Windows, Mac OS X, and Linux run the open source Arduino IDE. Download the Arduino software from the official website (depending on your OS) and follow the installation directions.

When we start using Arduino, debugging can seem like a complicated process, but when we break it down into steps, we were able to overcome the problems and learn a lot along the way.

6.1.4 Printed Circuit Controller

The printed circuit controller on the board is designed to take in 20 analog inputs, which comes from the 20 photoresistors we have on our target board. It also has room for the Bluetooth module chosen. There are 4 pins on the board dedicated to Bluetooth and transmitting the data to our application. We also needed a five volt input source, and power source to power our board and LEDs around each of the targets.

The design of the five boards attached to the targets, seen in previous sections, allowed us to get a reading from our targets. These boards are built using the Eagle software. The testing of the photoresistors was done using the Arduino Mega board and then transferred over to the PCBs, which made it very simple to ensure that we are getting and sending the correct responses from the photoresistors.

The board itself is smaller than our target board. Allowing for us to easily be able to nail the board to the back of the board and run the necessary wires and components to each target. These five boards are also nailed to the target board, which are connected to each target. The target board itself is planned to be 3ft x 3ft. The PCBs are relatively small, giving us plenty of room to put them on the back and make sure the board is not too wide.

We used JLCPCB order our PCB. Originally planning to use ExpressPCB, we did some more research and discussion. After discussing with students who had taken Senior Design previously, we were told that JLCPCB solders parts on the board for free for us. Since none of us have experience doing this, this was the most cost-efficient way to get our PCB ordered and complete. JLCPCB also allows us to order multiple PCBs in larger quantities, since we needed 6 of these, the price was lowest by using this company.

6.1.5 Bluetooth Module and Microcontroller

During the previous section (3.3), we discussed the research done to determine the best way to implement a way to send data between our board and our Mobile application. We determined the best way to do so would be to use the DSD TECH HM-10 Bluetooth 4.0 BLE iBeacon UART Module with 4PIN Base Board for Arduino UNO R3 Mega 2560 Nano attached to the Arduino Mega board. This communicates with the other microcontrollers on the board and send the received data to the application.

This bluetooth module is attached to the designed pcb that was displayed earlier in the paper. We attached the 4 necessary pins to the board, and transmit the data to the application. The module itself is tiny and allowed us to keep the game slim so the target board is not too big and taking up too much space. The bluetooth module was first tested and coded on the Arduino mega, which made the transfer over to the PCB very easy. We can see this PCB in Figure 6.1.5.1.

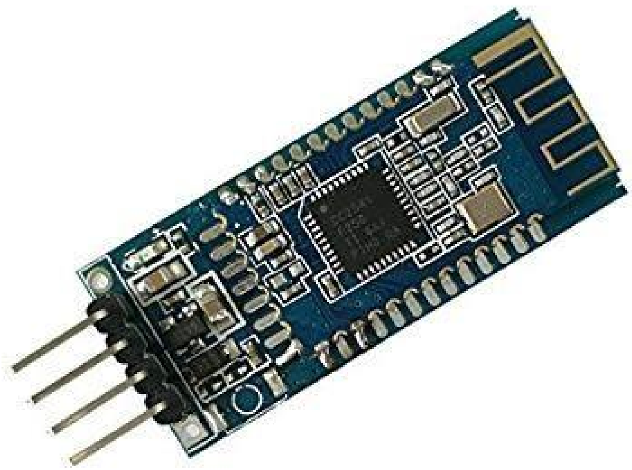


Figure 6.1.5.1: TECH HM-10 Bluetooth 4.0 BLE iBeacon UART Module

6.2 Software Prototyping

The mobile application has six parts to its development cycle: Synchronization, Database, Mobile Framework, Event mapping, Data Synchronization, Data analysis, and Final Tweaks.

First, to secure a successful development cycle our team began by creating a repository which we can document each version of our mobile application. We are using Git to perform such a task. This is particularly important for when we encounter errors that require a rollback to a previous save state to our application. This prevents dangerous scenarios where our team can potentially lose all of our progress due to a corrupt application.

Next, we designed and implemented a database model which best meets the needs for our project. Here we will define our classes, objects, and general interactions that our users have with the application. We initially designed the database with minimal constraints. The reasoning is such that we maintain a flexible development cycle that can be adjusted throughout the duration of the project. The purpose for the database at this stage is to provide a bare bone testing ground for the application. As such, we constantly make changes as we add more design constraints.

After our initial database is set up, we began to construct the visual components of the mobile application. Here our team is utilizing the framework React Native to provide an intuitive native UI design. The UI was written in JavaScript and focuses on providing users with a sleek design to enhance the experience when interacting with our project. Our goal was to create all the required pages needed for the whole application. We intend to follow mobile application design guidelines such as minimizing user input, breaking tasks into smaller sections, decluttering unnecessary information, visual consistency, and offer visual distractions during loading times.

Once we have a foundation for our mobile applications UI design, we then began to create events for our application. Here we will map out where each interactable button will direct the user. We want to make sure that each section is linked correctly to avoid confusion. Directing users to the wrong location will cause frustration. This would be a direct conflict to our design constraints and as such is intolerable.

Data synchronization will finally connect the game board to the mobile application. Here we will make sure that data is correctly being transferred and stored from one device to another. We developed a system that correctly pairs the devices and parse the incoming data. This data will then be directed into the proper objects that were defined in our database. We will perform multiple tests to assure there is no direct loss of data. We want to avoid cases where their mobile application cannot keep up with the data transfer rate of our microcontroller located on the game board. In the case of such a scenario, our team will make adjustments to the database such that a similar situation will not arise.

Once our team has confirmation of a secure data transfer medium, we implemented our data analysis methods. Here we intend to create visual renders of the game sessions and graphical representations of different variables that are tracked throughout a single game session. Our team intends to replicate an exact mapping of the game session by calibrating the sensors on the gameboard. We can pinpoint impact zones by comparing the detected data outputs to the relative position of each sensor. As such, users are able to visualize their own performance similar to how someone at a firing range would.

Finally, we began to finalize specific sections of our mobile application. This will include reusing our codebase to provide native support for multiple device platforms like iOS, implementing constraints to the database, and adjusting the visual layout of the mobile application. Here we also decided as a team whether or not we will include other features such as different game modes, player to player interactions, global leaderboards, or profile customization.

6.3 Microcontroller Coding Architecture

The Arduino ATmega328p, along with all Arduino microcontroller projects follow the same architectural mold. The Atmega328p is an 8-bit microcontroller that uses CMOS transistors for its instruction set, speaking more on the instruction set, this Arduino product is based off of RISC(Reduced Instruction Set Computer) methodology. Here are more details on the ATmega328 architecture:

1. The data is uploaded in serial from the port, which is being uploaded from the computer's Arduino independent development environment. The signal data is decoded and then the instructions are sent to IR (Instruction Register) and it decodes the instructions on the same clock cycle.

2. The very next clock cycle the next set of instructions are loaded into the instruction register.

3. The architecture of the Atmega328 provides 29 8-bit general purpose registers and 3 16-bit.

4. Status and control: are necessary to control the flow of execution of commands(conditional statements: if/then, etc.) by checking other blocks inside the computer processing unit at regular intervals.

5. Arithmetic and Logical unit**:** The high performance ALU operates directly with all the 32 general purpose registers. The ALU operations are divided into 3 categories – arithmetic, logical and bit-function.

6. I/O pins The digital inputs and outputs (digital I/O) on the Arduino are what allow us to connect our laser diode, LED strips, and trigger input.

7. EEPROM: stores data permanently, regardless of power being supplied or not.

8. Interrupt Unit: checks whether there is an interrupt for the execution of instructions to be executed in ISR (Interrupt Service Routine).

9. Analog comparator: compares the input values on the positive and negative pin, when the value of positive pin is higher the output is set.

The Arduino microcontrollers like most other computers harbor a firmware that is stored in non-volatile ROM memory, which stores a set of instructions. Another useful feature associated with the ATmega328 is the automatic reset functionality, this allows us to upload new code onto the device through the independent development environment, as opposed to having to physically press the reset button on the device. The Diagram below shows a block diagram representation of the ATmega328 architecture.

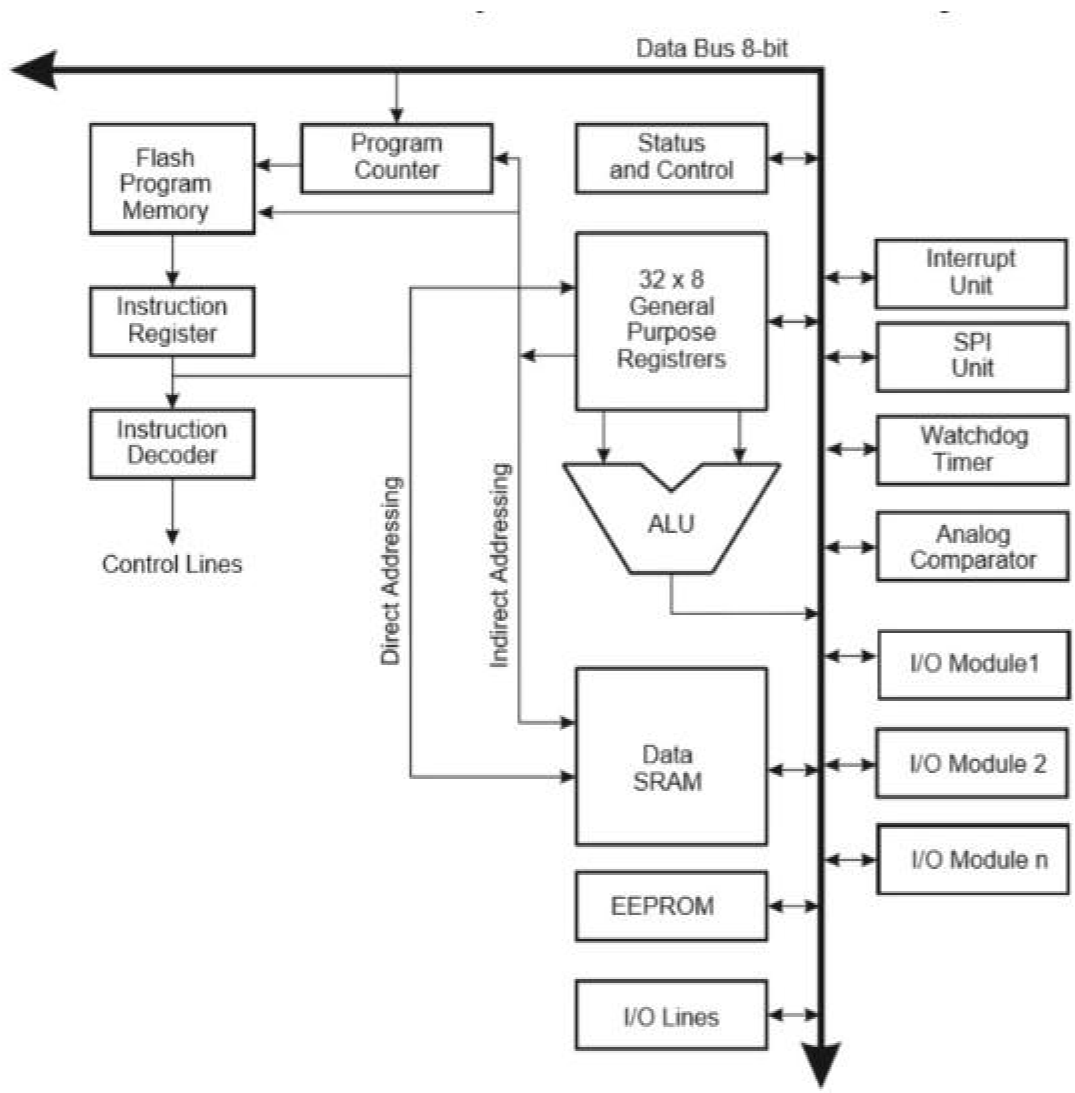


Diagram 6.3.1: Arduino ATmega328p Architecture Logic Diagram

Our project is Arduino based, and we used Arduino custom independent development environment to program both devices. There is an endless selection of programming languages to program the ATmega328 computer, however to keep our code concise and readable we chose to use Java as our standard language.

6.3.1 MCU connected to game board

The microcontroller which connects to the game board receives data from the targets. Each target has four photoresistors that send data to this microcontroller. We are using the Arduino Mega microcontroller. When connecting a light dependant resistor to this microcontroller analog pin. As the light increases the value returned by this analog pin will increase. This happens since the light dependant resistor decreases its resistance as the light increases, therefore more power will pass through to the microcontroller. The analog pin will return values between 0 and 1024, the higher the value the more light passing through the resistor, meaning the shot was closest to that pin. We have the option to put the photoresistor in the digital pin as well but then only values of 0 or 1 and would not allow us to return values letting us see how close the light was to the photoresistor. Using the Arduino Mega allows for us to connect all of the targets to one board, giving efficient responses to our mobile application. This will allow for us to be able to send one signal from the game board to the mobile application. This eliminates the issue of trying to connect multiple different boards to our mobile application and differentiate the data and determine which target was hit.

The Arduino Mega was tested throughout the first few weeks of Senior Design. Once fully tested and getting all correct readings and correctly sending data properly, we will switch all of the components over to the designed PCB. The PCB has been designed to have all of the same components as the Arduino Mega. Since all the components are very similar, transferring everything over to the PCB was a simple process. We can see many of these components in Figure 6.3.1.1

Figure 6.3.1.1: Various Arduino Related Electrical Components

6.3.2 MCU connected to Laser gun

We connected an Arduino Uno to the laser gun. This allowed for us to regulate how often a shot can be taken. Limiting users so that they go for an accurate shot and not just try to rapid fire and hit the target. It also reads when the trigger is pulled by the user and initiate a “shot”. Then the delay begins and will be able to take another shot once the delay is complete.

The Arduino will also trigger a slight sound to enhance the game experience. We want to give some feedback to the user that they can see so they are not blindly shooting, unsure whether they are actually shooting the gun and hitting the target. To do so we can attach a Piezo Buzzer to the arduino using a simple voltage divider. Then we have the ability to code our Arduino to trigger a slight noise each time the Arduino reads that the trigger has been pulled.

6.3.2 LEDs About the Targets

We also put LEDs abound each target. This allows for us to randomly light one LED ring up at a time to indicate this is the target that should be shot at. Each target is a different color and will light up at random times. When the LED lights up, the four photoresistors connected to the is initialized to input. The rest of the photoresistors on the board are put on a short delay, so that the user is unable to get point for shooting at an incorrect target.

We used an LED strip and split it up into five separate sections, each individually wrapped around a target. The input voltage of the leds is 12 volts. The power consumption of these LEDs is 4.8 M/W. The wattage of these LEDs is 24 W. There are a few different ways we could go about powering these LEDs. We could use the PCB attached to each of the targets to also power each of the LEDs. We could also use a battery source. We can create a common power source, strip each wire and connect it to this power source. Eventually, it was settled on using transistors to power the LEDs along with independent power supplies.

7.0 Testing

In this section we will discuss how we went about testing each of our individual components of the project, as well as the project as a whole. Testing each individual component as we go along made the overall testing of our project much easier. It allowed us to ensure each part is working before combining all of the parts together.

7.1 Testing of Laser

The testing of the laser is to be done at the nScrypt Laboratory. This is mainly due to the optical tools available in the lab itself, as well as the support of other staff when difficulties are encountered during testing.

7.1.1 Light Intensity

The main danger of the laser is that it is an intense focused beam of light. If looked directly into, light will tend to collect into a narrower spot at the eye's retina compared with light given off by a more ambient source such as a lightbulb or a fire. This is why it is imperative not to look directly at a laser, even a low level one like the one we are using in this project. While this is a class 3R laser product, this diode can still cause lasting damage, there is still allow risk of injury.

We can increase the eye safety of the beam to that of a class 2 laser by limiting the beam intensity to less than 1 mW and by reducing the amount of time the beam can remain active.

Obviously, any direct and intentional eye contact with a laser is dangerous, even a Class 1 laser, but we can test the max output intensity of our beam through the use of an optical power meter, as seen in the setup in Figure 7.1.1.1. We expect the power per centimeter to drop dramatically as well, due to the expansion of the beam. However, a worse case scenario would be firing the laser directly into the eyeball of a user, and we want to have safe exposure for that time period during a full laser fire.

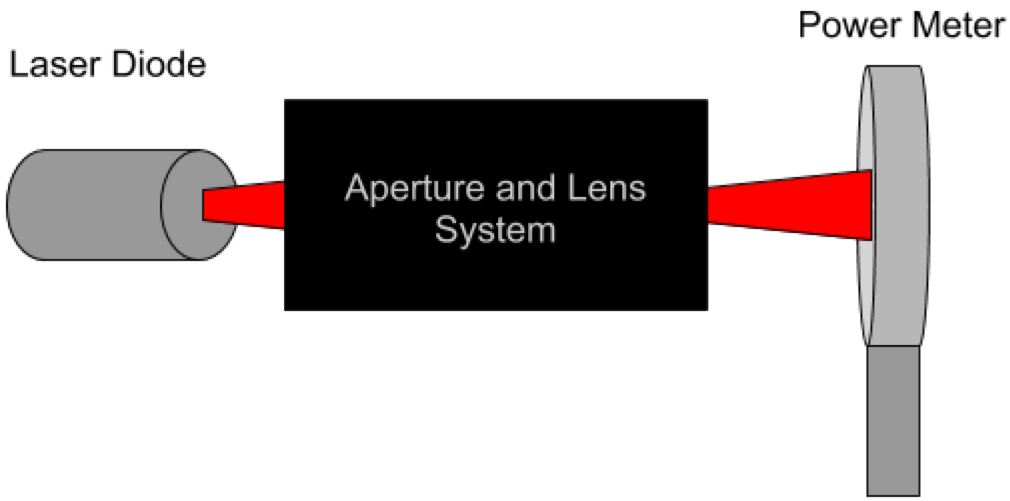


Figure 7.1.1.1: Optical setup to test new laser strength

7.1.2 Output Beam Shape with Aperture

One of the other things we are concerned with on the laser is the beam shape. The response that we received from the photodetectors and the scoring relies on the fact that there is a reasonably even distribution of light throughout the system. This is verified by simply expanding the output beam after it passes through the aperture and lens system with a positive lens and projecting it onto a white screen.

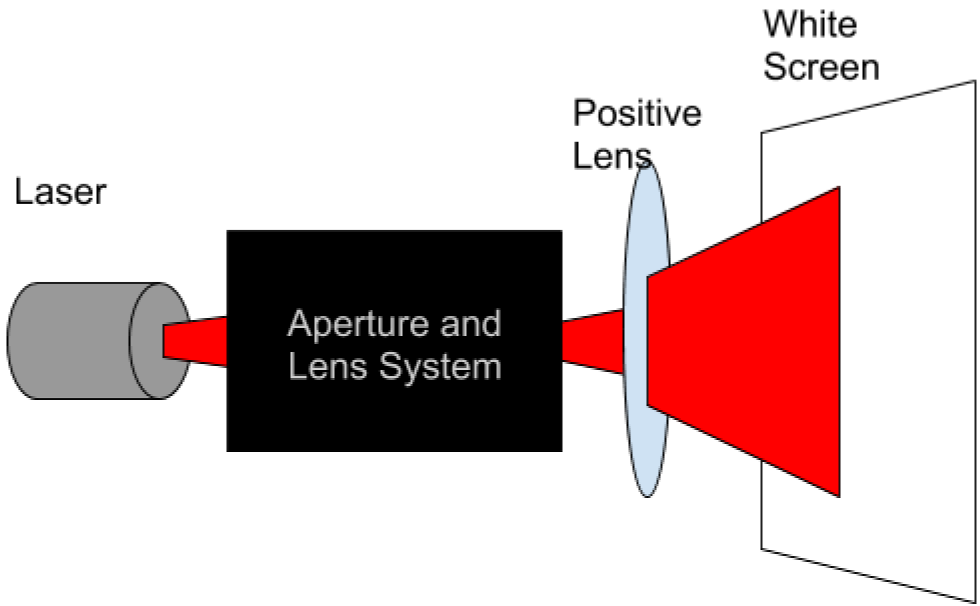


Figure 7.1.2.1: Experiment to observe beam shape output

What we expect to observe is a small circle, brightest at the midpoint, then dimming slightly around the edges before being cut off entirely by the aperture. It is entirely reasonable that the dimming on the vertical axis could be far greater or far less than the horizontal axis, due to the nature of semiconductor manufacturing. This can be mildly rectified in two ways. The first is a lens that focuses on only one axis, but these can be difficult to find. The second is to readjust the lens to further expand the beam before the aperture such that the change in intensity vertically and horizontal become negligible.

During this test, there are a few other things to watch out for. Specifically, there are several types of lens aberrations and distortions which could cause errors in the projected beam. Dust, scratches, and flaws in manufacturing can distort or cause ripples in the beam projection which can be fixed with cleaning. However, most distortions that are usually associated with camera lenses can be ignored. This is because the spherical aberration and chromatic aberrations caused by the shape of the lens and difference in refractive index based on the color of light are negligible. This is because we are not imaging the beam so much as projecting and directing the beam, so clarity isn't a huge factor in terms of spherical aberration. The chromatic aberrations also don't matter because we are only projecting on a very narrow linewidth.

7.1.3 Photoresistor Response to Laser

There are a couple of potential issues when using a photoresistor. photoresistor manufacturing can be imprecise, causing a wide range of resistivities and responses when interacting with the laser. A short test of this is done with the

GL5526 photoresistor. Testing in a room light office setting without much outside light interference, we can test in a similar environment the final product is used in. We can see a small setup of the system below in Figure 7.1.3.1. This setup is made up of two optical stands, which house the laser diode and lens, to align the beam output and project it to an approximate 2 inch diameter on the wall. Care was taken such that the most intense part of the beam is at the middle, and the horizontal axis resembles the distribution as described in Figure 4.4.4.2. Thankfully, no diffraction was noticeably observed, however, the y axis did not output the expected rotationally symmetrical beam that is optimal. This can be rectified with a different laser or through a cylindrical lens, but these are options that can be considered at a later time, as only the horizontal axis is needed in this experiment.

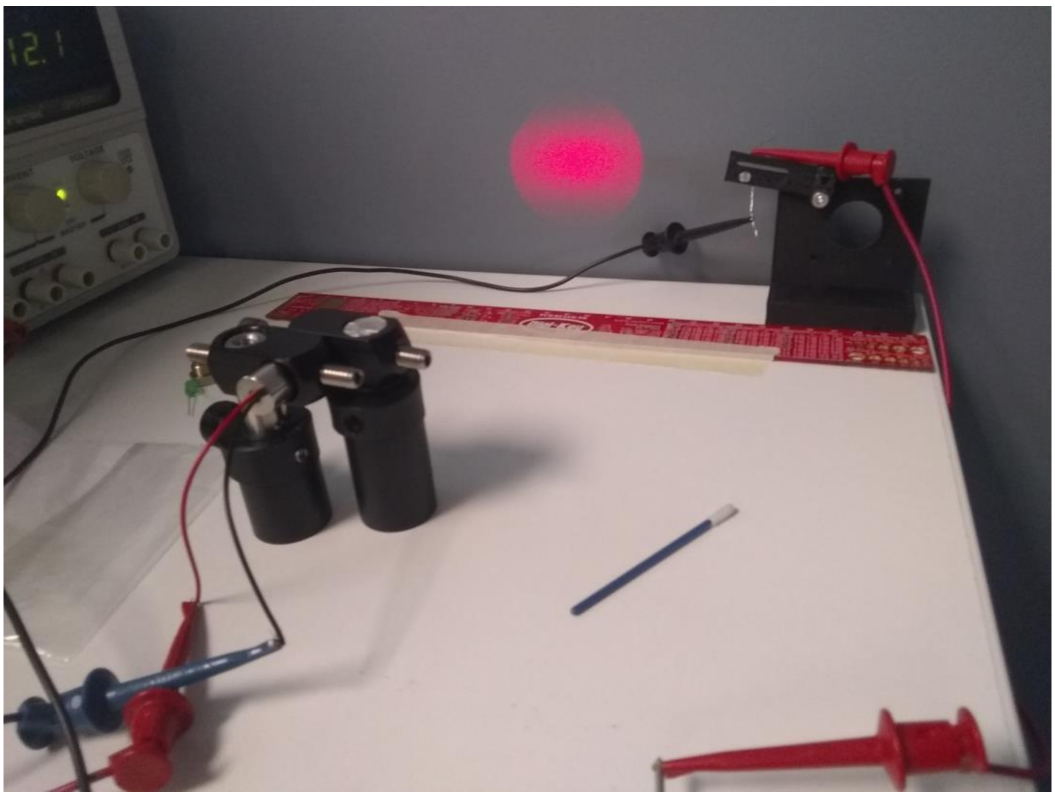


Figure 7.1.3.1: Photoresistor Resistance Testing

A photoresistor was then secured into a 3D printed holding bracket. This bracket is designed to house photodiodes and secure them to a board. The board used here is another 3D printed bracket with a large heavy based that is used to slide against the wall. The photodiode bracket was secured down and angled to run the photodiode across the center of the beam. A ruler was then loosely secured to the table to measure movement.

The experiment is then run as such: The photoresistor is placed a healthy distance away from the beam, and it's resistance is recorded with a digital multimeter. The bracket is then moved in 5 mm increments, the resistance of the photodiode recorded at each step. This trial was run a total of 10 times with 10 different photoresistors to track important changes or distinctions between photoresistors. The laser was adjusted in between some trials, so some variance in responsivity and brightness is expected, as would be expected in real life applications. We can see the results in Table 7.1.3.1 and in Figure 7.1.3.2..

Table 7.1.3.1: Resistances of Photoresistor Across Beam in KΩ

Trial #	1	2	3	4	5	6	7	8	9	10
0 mm	6.52	-	4.37	6.01	5.27	5.32	4.47	-	-	5.4
5 mm	6.27	5.6	4.42	6.05	5.24	5.35	4.39	6	-	5.64
10 mm	6.52	5.63	4.48	5.85	5.26	5.32	3.03	3.95	7.96	5.21
15 mm	6.27	2.86	4.33	3.74	3.81	3.75	2.88	2.85	8.01	3.41
20 mm	3.1	2.65	2.61	2.92	2.71	2.72	2.14	2.61	6.85	2.63
25 mm	2.83	2.1	2.26	2.7	2.49	2.5	1.98	2.43	3.04	2.38
30 mm	2.52	2.2	2.01	2.46	2.31	2.32	1.83	2.21	2.55	2.13
35 mm	2.32	2.17	1.95	2.3	2.18	2.15	1.76	2.11	2.16	1.89
40 mm	2.19	2.03	1.84	2.18	2.04	2.04	1.73	2.09	1.945	1.75
45 mm	2.1	2	1.72	2.13	2.02	2.01	1.72	2.07	1.782	1.57
50 mm	2.15	2.05	1.69	2.15	2.02	2	1.8	2.15	1.93	1.57
55 mm	2.26	2.16	1.7	2.25	2.057	2.1	1.936	2.31	2.26	1.64
60 mm	2.49	2.27	1.78	2.38	2.2	2.21	2.04	2.43	2.78	1.78
65 mm	2.76	2.54	1.93	2.5	2.35	2.36	2.22	2.66	3.15	2.02
70 mm	3.08	2.72	2.06	2.76	2.51	2.5	2.42	2.87	3.55	2.3
75 mm	6.53	3.022	2.28	3	2.69	2.73	3.31	3.27	8.56	2.6
80 mm	6.74	5.91	2.45	3.6	3.4	4.38	4.65	6.35	-	2.85
85 mm	-	-	3.94	6.4	5.33	5.5	4.65	-	-	5.63
90 mm	-	-	4.7	6.45	5.44	-	-	-	-	5.8

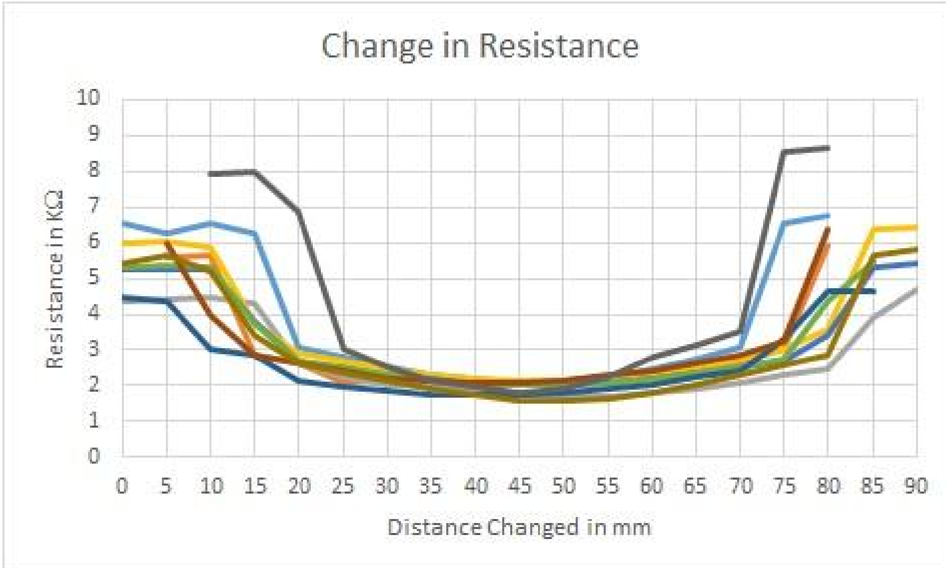


Figure 7.1.3.2: Change in Resistance of Photoresistors Across Beam

One of the more noticeable things about this graph is you can see the trial where the beam was adjusted to be smaller distinctly from the rest of the trials, but it does not actually change the magnitude of response that much. However, because we see the values of resistance run from 8 kΩ to 2 kΩ, we would like to choose a resistor to put in series to be able to output an appropriate change in voltage. One of the more concerning bits of the above is that the idle response of some photoresistors has a fair amount of variance. However, we can simply choose to not use these resistors or account for this change on the software side of the project.

Choosing a 3kΩ resistor, we again set up the project with a circuit as seen in Figure 7.1.3.3. Then, using the same process and tracking the voltage over the 3kΩ resistor instead of the resistance, we should find a response able to be tracked by the Arduino later. We can see the results in Table 7.1.3.2 and Figure 7.1.3.4

Table 7.1.3.2: Change in Voltage Across Resistor in V

Trial	1	2	3	4	5	6	7	8	9	10
0 mm	1.46	-	-	1.74	1.88	1.46	-	1.66	1.73	1.45
5 mm	1.47	1.14	1.62	1.75	1.89	1.46	1.73	1.67	1.72	1.47
10 mm	1.52	1.15	2.6	1.78	2.01	1.81	1.74	2.02	2.45	1.79
15 mm	2.54	1.15	2.78	2.71	2.88	2.39	2.29	2.55	2.72	2.45
20 mm	2.72	1.53	2.94	2.9	3.02	2.56	2.64	2.74	2.9	2.65
25 mm	2.91	1.78	3.11	3.07	3.15	2.74	2.79	2.88	3.06	2.79
30 mm	3.09	1.9	3.23	3.2	3.32	2.92	2.94	3.06	3.23	2.96
35 mm	3.25	1.97	3.43	3.34	3.43	3.02	3.16	3.13	3.3	2.98
40 mm	3.34	2.01	3.4	3.43	3.53	3	3.14	3.15	3.32	2.97
45 mm	3.37	2.07	3.3	3.53	3.47	2.89	3.03	3.06	3.22	2.9
50 mm	3.25	2.05	3.01	3.37	3.28	2.67	2.84	2.85	3.04	2.74
55 mm	3.07	1.98	2.87	3.16	3.14	2.52	2.74	2.72	2.88	2.62
60 mm	2.83	1.87	2.63	2.91	2.89	2.33	2.58	2.56	2.73	2.46
65 mm	2.66	1.82	1.99	2.75	2.7	2.15	2.42	2.33	2.52	2.31
70 mm	1.97	1.69	1.35	2.55	2.37	1.43	1.69	1.64	1.64	1.74
75 mm	1.31	1.05	1.3	1.83	1.86	1.42	1.65	1.59	1.53	1.38
80 mm	1.28	-	-	1.8	1.84	-	-	1.58	1.52	-

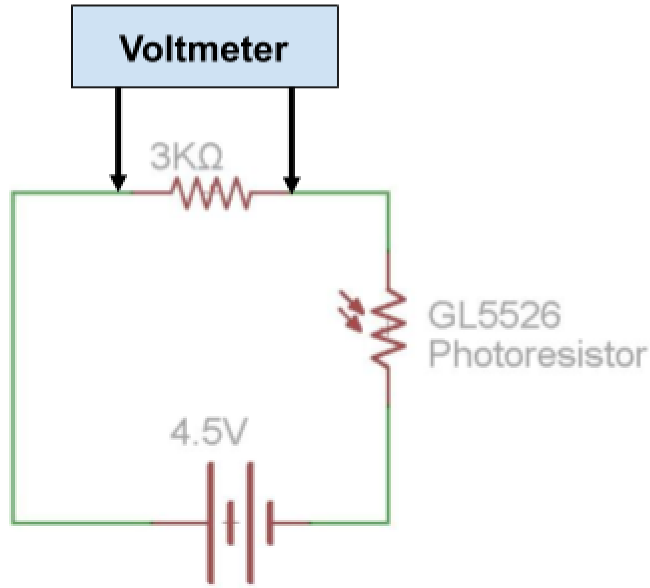


Figure 7.1.3.3: Circuit Diagram for Voltage testing

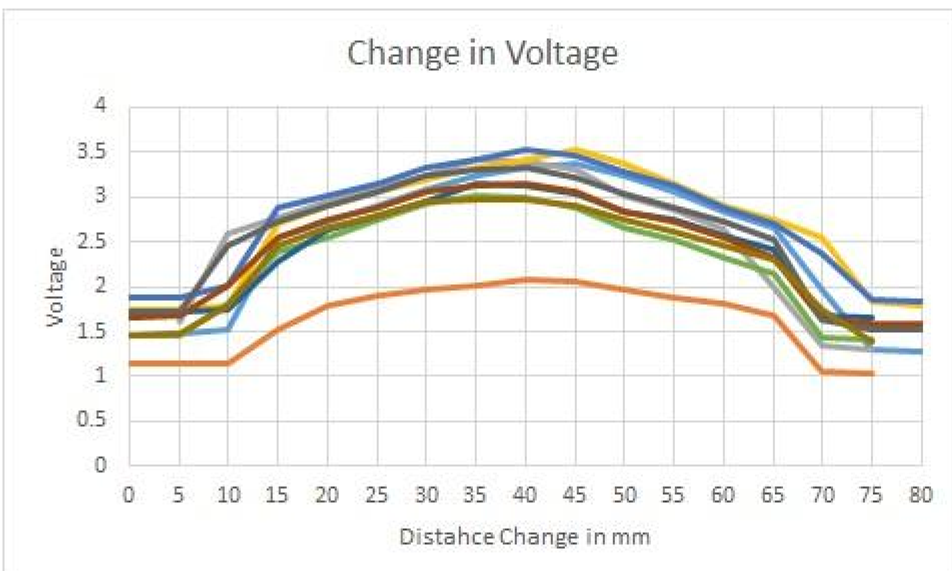


Figure 7.1.3.4: Change in Voltage Across Resistor

What we can read from this experiment is that our responses have a nice distinct curvature which was great to work with. With the exception of the orange line, which may very well be a photoresistor with greatly different expected resistance values. Even still, there is a distinct response which can be tracked and recorded by an Arduino. We can clearly see where the photoresistor starts and stops responding, allowing us to comfortably count anything above 2 V as a hit.

The next step from here is to get the microcontroller responding to the photodiodes.

7.1.4 Microcontroller response to Laser

We first needed to test if the microcontroller response is working with one diode before we can attach all four and make sure it is properly running. To test this, we set up one photoresistor and attach it to our main microcontroller in a similar fashion to 7.1.4. We then shot this photoresistor with the laser we are using for our game. We first tested that we are getting a response from the microcontroller when shooting this photoresistor directly and then from a slightly inaccurate shot. Setting this up was simple. We set one photoresistor up as seen in the diagram below. We then checked that values between 0 and 1024 are being returned. Once we are able to get data from one photoresistor, we can begin to attach all of our photo resistors. This is when we began to test the overall target. We then were able to set up our accuracy feature for the game. Depending on how high the voltage signal sent by each photoresistor will indicate which resistor the light was closest to.

When beginning actual simulation testing, we started by simulating the response of one photoresistor when exposed to light. To test the photoresistor response to the amount of light it has been exposed to. We built a schematic that would allow for us to test the values we would get back when the photoresistor was exposed to different amounts of light. Using TinkerCard, we were able to build the board and test the output values the photoresistor would give us. In figure 7.1.3-1, the analog pin A0 is connected to the photoresistor. When exposed to different amounts of light we will get different values varying between 0 and 1024. We were then able to simulate this circuit and have the values outputted every 15 milliseconds as we varied the light input into the photoresistor. The more light the photoresistor was exposed to the higher the value returned. This allowed us to see how one photoresistor would respond when hit by the laser. To test one full target being hit we needed to now add four photoresistors to the board and simulate the response from each, when all exposed to different amounts of light. As the photoresistor is exposed to more light the LED increased in brightness showing the photoresistor was being exposed to different amounts of light. We coded this board as well to show the analog input from the light intensity.

Testing one photoresistor further allows us to test multiple and allow for us to create our printed circuit board that we used for each of our targets. We created 4 different printed circuit boards, one for each of the targets on our board. Each with four photoresistors on our board allowing for us to get an accurate response from the laser. Points are given out based on accuracy so we need as many of these photoresistors per target that we can, giving us the most accurate score possible.

The Arduino Mega has been ordered and testing on the board began immediately once it arrives. However, the delay in getting the board in allows us to only perform online testing up to this point on the Arduino board. The online testing still allows for us to test the theoretical results we will receive from the photoresistor, and not the actual values. This does, however, allow us to structure and come up with our code for creating the response and giving out scores for each shot. The code for each target has been created and will take into account the four possible targets that can be shot. Depending on the current/voltage sent from each one, we send it through our created algorithm and a point value was designed.

Test Number	Condition, without target casing	Photo-Resistor 1	Photo-Resistor 2	Photo-Resistor 3	Photo-Resistor 4
1	no laser , testing Ambient light	430	476	429	482
2	no laser , testing Ambient light	465	399	487	465
3	no laser , testing Ambient light	402	431	443	404
4	no laser , testing Ambient light	498	489	490	399
5	no laser , testing Ambient light	476	434	500	451
6	laser in center of resistors	832	743	737	752
7	laser in center of resistors	801	751	742	742
8	laser in center of resistors	789	790	737	743
9	laser in center of resistors	790	787	742	742
10	laser in center of resistors	802	745	790	738
11	laser aimed at resistor 1	950	702	701	720
12	laser aimed at resistor 1	948	711	723	698
13	laser aimed at resistor 1	963	732	798	715
14	laser aimed at resistor 1	1001	798	745	705

15	laser aimed at resistor 1	993	732	701	700
16	laser aimed at resistor 2	711	989	703	715
17	laser aimed at resistor 2	702	963	731	721
18	laser aimed at resistor 2	720	942	745	698
19	laser aimed at resistor 2	820	938	800	767
20	laser aimed at resistor 2	723	980	732	707
21	laser aimed at resistor 3	740	703	998	721
22	laser aimed at resistor 3	780	710	950	712
23	laser aimed at resistor 3	723	698	1021	720
24	laser aimed at resistor 3	712	732	989	689
25	laser aimed at resistor 3	709	719	899	714
26	laser aimed at resistor 4	703	721	803	950
27	laser aimed at resistor 4	698	712	756	987
28	laser aimed at resistor 4	683	707	812	950
29	laser aimed at resistor 4	692	690	723	934
30	laser aimed at resistor 4	683	714	740	1000

Table 7.1.3.1: Testing one target with no target casing

Test Number	Condition, with target casing	Photo-Resistor 1	Photo-Resistor 2	Photo-Resistor 3	Photo-Resistor 4
1	no laser , testing Ambient light	320	313	297	303
2	no laser , testing Ambient light	306	279	302	311
3	no laser , testing Ambient light	315	301	311	304
4	no laser , testing Ambient light	298	310	302	280
5	no laser , testing Ambient light	289	278	300	305
6	laser in center of resistors	784	782	770	781
7	laser in center of resistors	786	803	786	798
8	laser in center of resistors	743	732	812	743
9	laser in center of resistors	801	721	801	798
10	laser in center of resistors	765	804	783	803
11	laser aimed at resistor 1	980	702	701	720
12	laser aimed at resistor 1	978	711	723	698
13	laser aimed at resistor 1	970	732	798	715
14	laser aimed at resistor 1	899	798	745	705
15	laser aimed at resistor 1	993	732	701	700
16	laser aimed at resistor 2	700	976	703	715
17	laser aimed at resistor 2	723	903	731	721
18	laser aimed at resistor 2	704	1002	745	698
19	laser aimed at resistor 2	779	945	800	732
20	laser aimed at resistor 2	711	934	732	773
21	laser aimed at resistor 3	803	703	998	721
22	laser aimed at resistor 3	683	710	950	687
23	laser aimed at resistor 3	676	698	1011	734

24	laser aimed at resistor 3	801	732	980	704
25	laser aimed at resistor 3	773	719	899	736
26	laser aimed at resistor 4	703	721	803	890
27	laser aimed at resistor 4	698	712	756	932
28	laser aimed at resistor 4	683	707	812	980
29	laser aimed at resistor 4	692	690	723	965
30	laser aimed at resistor 4	683	714	703	993

Table 7.1.3.2: Testing target with 3D printed casing

7.1.5 Faculties Used

The main testing facility for the laser is the Engineering Building's Senior Design lab. Inside this lab is a few of the components we need for testing. Primarily, what we need is the power supply, a multimeter, and a computer to load code onto the arduino boards. However, there are some alternative locations which may be used for board construction. Primarily, this would be the Texas Instrument Innovation Lab, which has many woodworking and electrical supplies for the soldering of the PCBs and securing of the components to the board itself.

We tested the Arduinos online using TinkerCard. This tool allows for online testing of Arduino boards. We can choose all the components that are used and test them on an online breadboard and microcontroller. This made the actual testing of the board and components much easier. We have the code pre-built using TinkerCard and only had to make some slight modifications when actual testing the board.

7.2 Application Testing

In this section we will take a look at how we tested each part of our mobile application. To test our mobile application most efficiently, we will break down the testing into individual parts. To test the quality of our game we has a few game testers play the game before we complete our final prototype. We needed to ensure the user interface of the application creates a good experience for the user.

Specifically, we want to know each tester's initial thoughts on the UI of the application. The first experience is a crucial part to a well-designed mobile application, and we focused on making sure it will draw the user into interacting with the application. Our desired result is a UI which is sleek and thorough in its communication. Users shouldn't need to ask for a tutorial in order to navigate the mobile application. Instead it should come seamlessly and intuitively based on the communication line that is established which each UI element.

Next we asked the testers to attempt a game session. We want feedback as to how quickly the mobile application pairs with the game board and if the application automatically navigates the tester back to their home screen upon a failed connection. We provided the testers with multiple scenarios that potentially affected how the application interacts with the microcontroller.

First, we tested the maximum distance allowed between the two devices for shared information. One trial consisted of an open field with no obstacles around. The range recorded among our testers was the maximum ideal distance possible from the device to the game board before the pair breaks. Afterwards we tested the pairing strength in a much more crowded area. This test gave our team an idea of the realistic distance users can be from a device.

Next we want to make sure that the microcontroller and the application are transmitting error free data from the board. We have a controlled and calculated laser pulse target a specific spot repeatedly on the board. These pulses act as a control group for the photodiode sensors.

If the data is error free, then we should be consistently receiving similar data throughout each pulse. Some marginal differences are to be expected when factoring in ambient lighting and other outside factors that could affect the photodiodes keen sensitivity. A faulty data retrieval was noticed if there are major variances found between the data samples. However, we can't directly assume it is the microprocessor or the mobile applications fault.

There could be a fault PCB or photodiode which corrupts the data process. To accurately test such an issue, we also needed to take a multimeter and measure the current passed through the photodiode to be completely positive with the data collected. If the multimeter shows signs of malfunction then we can assume the issue isn't related to the communication between the microcontroller and the mobile application.

Once testing of each individual part of our application is complete, we moved on to testing the application as a whole. We needed to perform, end to end testing, first as a team. We tested playing the game, data being displayed, and scores being displayed. This is all done internally as a team so that we ensure that each part is working properly together before showing a test group.

Once testing as a team is complete, we tested our application with test users. This also made sure that we can handle large amounts of data without crashing our mobile application. Since our game could be played as many times as one would like to as well as by as many people, the data began to gather quickly. Our database should be able to handle all this data no problem but testing this makes sure that we do not come across any issues before our final presentation. Once testing all the game play, we also made sure the players are pleased with the user interface of the application. We then continued testing and updating until we are pleased with the feedback from them, we can roll out our final prototype for our mobile application.

7.2.1 Receiving data when target hit

To determine the accuracy of the shot, we have four photoresistors around each of our targets. When one of the targets lights up, and the user shoots we are able to determine where on the target is hit. In the diagram below, the four blue dots would be our photoresistors.

Depending on where the laser hits on our target, the current through these photoresistors would vary. The closer the laser is to the photoresistor, the higher the current this photoresistor would generate. If a shot was hit directly in the center of the target the current through all the photoresistors would be equivalent, meaning it was a direct hit. We used a microcontroller at each of these targets to read the current through each of these photoresistors and determine the accuracy of the shot. This data is then sent to the Bluetooth microcontroller, which all the photoresistor reading microcontrollers was hooked up to. Once the data is received by the Bluetooth microcontroller, the point value is sent to our mobile application.

To test that the data is being received when the target is shot. We initially set up one target with the four photoresistors. We hooked this target up to the printed circuit board as well as connect it to the microcontroller. We shot the target and ensure the data is being transmitted properly to the microcontroller. Once we can verify that one of our targets is fully functioning, we can begin to build all of the targets for our game. We tested each target as we go along, making sure that our target board as a whole was functioning properly.

7.2.2 Displaying points

For our game to be competitive, we needed to come up with a points system. We decided as a group to give points based on accuracy of each shot. Depending on how close the shot is to the center, the higher the points are. We set a max score per shot at 500 points. If the user shoots farther from the target they will get less points. As the round goes on, the user will then be able to see their overall score displayed at the end of the round. Once the round is over, the player is given their overall score and see how it matches up against all the users who have ever played the game.

We also needed to test that these points are being displayed properly. To do so, we first made sure the equation we came up with to issue out points, is generating the correct number of points on the computer. Once we determine that the points are correct, we send this data to our mobile application and verify that the number being displayed on the mobile app is the same.

Figure 7.2.2.1: Score Displayed

7.2.3 Displaying previous high scores

Testing that previous high scores are being displayed properly should be fairly simple. To do so we just needed to play the game multiple times on our own. If running properly, the 10 highest previous scores are displayed on the game screen. When testing we may need to modify how many high scores are displayed.

Depending on legible and clean text size as well as how much room on the screen, we needed to adjust accordingly so that the user interface looks clean when playing the game. One option we can try to achieve a smoother look would be to make the leaderboards a scrollable list with finite entries. This would solve the potential problem of running out of space on the phone screen.

The scores should be displayed on the results page as well. Once the player finishes playing the game, they are taken to their scores page. On this page there should be the results displayed. This allows for the player to see how their score compares to the high scores stored in the database. We can easily test this by playing our game and ensure that the scores are stored at the end.

Figure 7.2.3.1: Possible Home screen displaying scores

In figure 7.2.3.2 we depict the leaderboard records among our userbase. The high scores of the most skilled users are displayed along with their names. Users are able to compare their current game score to those high scores similar to how old arcade games such as Pac-man or Galaga would. Once a player achieves a score which is high enough to make it into the top 10 listing, they overtake all scores underneath theirs. The lowest high score is bumped off the leaderboard and will need to achieve a score even higher than their previous to appear again.

Figure 7.2.3.2: Example Game Over Screen

After a multiplayer game session, users can compare their score among their competitors. If a player wins, they will receive a token. Tokens add up after each win. They act as milestones for users. A reminder for every time they beat out all competitors. Functionality was added to calculate the users win ratio during multiplayer competitive sessions. Once implemented there is another category in the top 10 leaderboards.

There are many ways our team can implement the application leaderboards, but in the end we had to wait until the mobile application has reached its operational point before we can tweak the visuals.

8.0 Project Constraints

The gun is going to be a medium size handgun barrel approximately 10 inches, weighing about 5 pounds, to be held with two hands. . The laser diode itself is eye safe and have a spot diameter of about 2 inches at 15 feet away, such that the spot size is about to hit all four photodetectors at once. The power supply for the gun should be 5V and provide enough power to the gun for at least 2 hours. The printed circuit board, the laser diode, and the power supply should fit completely encased by the gun housing.

The board itself is a 3 by 3 ft project board. Each of the 5 targets are placed in an X pattern in a 2.5 x 2.5 ft box around the middle of the board. Each target is made up of a hollow 1.5 in diameter box angle slightly to point at a position of 15 ft away. This is such that when firing at the board, the targets itself angle to where the gun may be held away from the body.

Inside each box is a printed circuit board with four photoresistors arranged in a square with a width of 0.5 in centered around the middle of the box. At the edge of each box is LED array which lights up, indicating that this target is the one to shoot.

Each printed circuit board on the target also supplied power to the LED ring around each target. The printed circuit boards are wired behind the board to a Bluetooth controller, such that all the data channels are streamlined into one board controlling 5 targets, 4 photoresistors each, and 5 sets of LED circles.

The application should be used on both Apple and Android phones. Normally to achieve this process two different applications would need to be created using the native platforms; however, this process has the potential to cause multiple conflicts when trying to achieve cross compatibility. To try and prevent this from causing issues we used react native, which should allow our app to run on both of these devices. To create the game to be multiplayer, we set up a timer within the app and let each user shoot for one minute. Once the minute is up, the user is given their score and be able to compare their score to everyone who has played before them. A scoreboard is constantly updated for the top 10 players.

Table 8.0.1: Design Constraints

Eye Safe Laser	Below 1mW of Power
Board Size	3 Feet by 3 Feet
Gun Weight	Greater than 4 pounds but less than 10 pounds

Table 8.0.2: Design Standards

Laser Color	632 nm
Phone App Framework	React Native
Programming Languages	JavaScript, Objective C, Java, and Python
Database	RealmDB

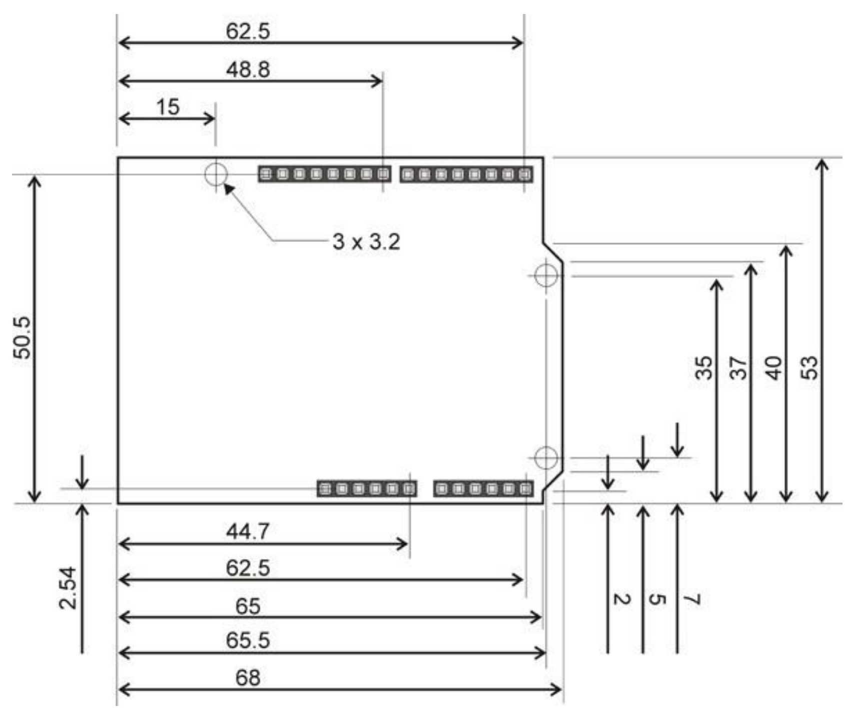
Another constraint was the size of the hollow shell of the gun, we are going to have limited space inside of the pistol to place all the necessary components. In total we have a power supply, an Arduino Uno, a laser diode, and a switch trigger. All that needs to be in the gun with room to function, the more space inside the pistol the better, there is also room set aside for weights and anchors for the components. For a better understanding of how these devices are all going to fit here is a list of the dimension of everything.

Table 8.0.3: Device Measurements

Device	Width	Length
Arduino Uno	68.6 mm [2.7 in]	53.3 mm [2.1 in]
Portable Charger (Rechargeable Battery Pack)	90.424mm[3.56in]	62.484mm[2.46in]
Laser Diode	13mm[.51in] (Diameter)	35 mm Length[1.37in]
Switch Trigger	25mm[.98in]	30mm[1.18in]
Weight	15mm[.59in]	10mm[.39in]
Anchor	5mm[.2in]	5mm[.2in]

We faced a few project constraints when designing the printed circuit board for the game as well. We needed 20 input pins that are able to read input from the printed circuit board connected to the targets. Each input is connecting to this printed circuit board and is sending different voltage signals to the board, so each needed its own input.

We also needed toggleable power to 5 LED rings, which allowed us to turn on the lights on one LED at a time, indicating that target is the target to be shot at. We also needed to be able to supply a constant voltage of approximately 3.3 to 5 volts to the photoresistors, allowing them to constantly to sense the laser has shot at that specific target.



Figure(8.0.1): Arduino Uno 3 Dimensions

Figure(8.0.2): Rechargeable Lithium Ion Battery Pack Figure and Dimensions

9.0 Administrative Content and Project Operation

In this section we will discuss the overall operation of how our game is played. The game is started from the mobile application. Once choosing to start the game, the mobile application communicates these to our game board and begin the game. The steps of playing the game as well as troubleshooting is highlighted in this section if there are any difficulties when trying to play. This helps explain how the user would go about troubleshooting our application if an issue arises, or where to find a solution to a problem the user may be having.

9.1 Game setup

To set up the game, the user first has to create an account if a new user to the game. This allows for the scores of all the players to be tracked and saved in the database. If the player already has an account, they will just have to login. We then made it so the user has to initiate the start of the game. Once the user has chosen to start the game. The LED rings on the game board begins to light up one at a time, the target that lights up indicates that this target should be shot. The rules of the game is displayed on a help screen of our mobile application. Allowing for the rules of the game to be explained as well as any questions the user may have to be answered.

9.2 Playing Speed Target Arcade

Once the game begins the timer begins. The player has a set amount of time to hit as many targets as possible, scoring points each time a target is hit. The gun is programmed to delay for a short period of time in between shots so the player cannot shoot continuously at the target. The player will continue to shoot at the targets until the timer is up. Each time the gun is shot, a noise is sounded so the player will know that they have successfully shot. There is also a noise made when a target is hit, this gives good feedback to the player, keeping them involved and entertained while playing the game. The player will continue to play the game until the timer is up and the round has ended.

Once the timer ends, a noise is made, signaling the end of the game. There is then an option for the user to play again, view high scores, and logout. If choosing to logout the user is done and another player can sign in and play the game. If choosing to play again, you are taken back to the home screen and have to click the button to start the game again. The player has the ability to play as many times as he/she wishes and stores all of their scores in the database. This gives them the ability to see their scores of all the times they have played, showing their highest score as well as lowest.

9.3 Results

Once the timer is up and the game is over. The total score for the round is displayed on the screen for the player to see how they did. They are also able to see the overall top scores of all users to ever play the game. The player is then brought back to the home screen, which also displays the overall top scores, and have the option to play the game again if they wish to. The results are all to be saved in our database. Each user has all of their scores saved to our database. If they have played the game multiple times they are able to see all of the scores from the times they have played. Giving them the ability to see their best score and worst scores from all of the game plays. The highest scores that are stored in the database is displayed on the home screen as well as the end of the game screen.

9.4 Troubleshooting

To help the player with any issues they may have, we have a section dedicated to help the user with all aspects of the game. We have an area that has the rules and explanation of how to play the game. We also have a few troubleshooting questions and answers, in case the player runs into any trouble when attempting to play the game.

This help button has some frequently asked questions we gathered when testing with test users. We put some solutions to problems we could see coming about when trying to play the game. However, we were unable to cover all of the problems that could come up. This section had to be continuously updated with solutions. This will ensure the player has the ability to find the answer to any questions that they may have, and never be unable to find a solution.

9.5 House of Quality Diagram

The house of quality diagram is a tool used to determine priorities when building the project. This was developed early on in the works and shows what we decided to be the most important parts of the project as well as what features of the device we wanted to minimize and maximize. We can see the house of quality diagram in Figure 9.5.1.

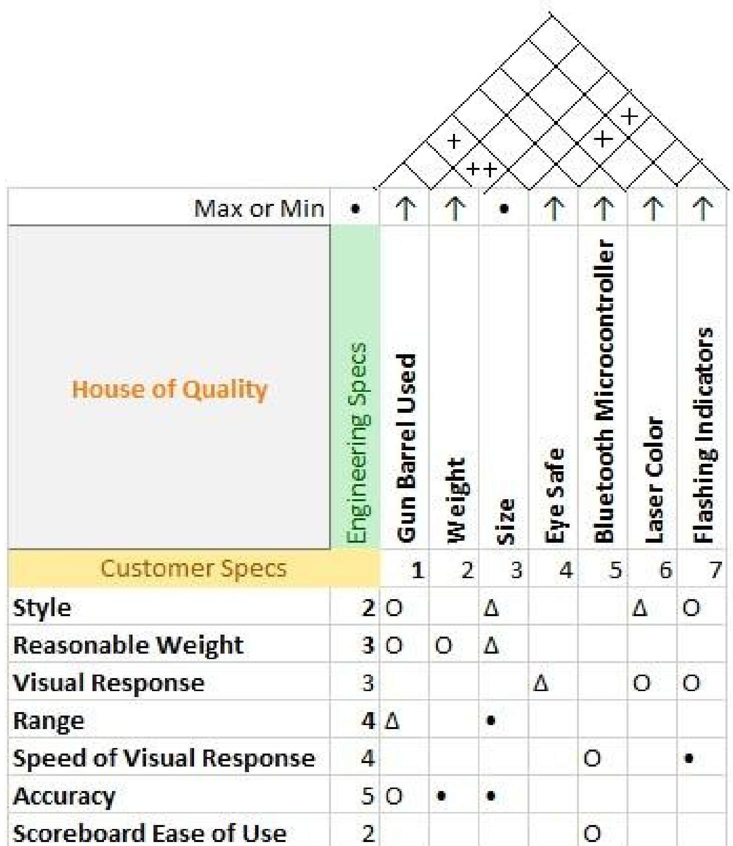


Figure 9.5.1: House of Quality

In the house of diagram, triangles are used to indicate heavy relationship between features, the open circle indicates medium relationships between features, and small circles indicate a mild relationship. We use this in junction with the min or max chart to try and prioritize what the customer wants versus what can be feasible to accomplish.

Developed around the same time is the table of engineering requirements. This was developed very early on, just after conception, to try and organize what are the base components needed in developing the project. From our initial observations we thought about what were the key components that would make up the foundation of our project.

We thought about our effective game range based around the emitted laser, the portability of our project, safety features to protect users of our product, and the visual stimulus our project would provide.

Table 9.5.1: Engineering Requirements

Engineering Requirements	
Target #	5
Target Size	2 inch radius per target
Photoresistor Count	4 per target, 20 total
Board Size	3 feet by 3 feet
Microcontroller #	2
Bluetooth	Yes
Phone Compatibility	Both Android and iPhone
Laser Diode	Visible Red
Beam Size	2.5 inch radius
Gun Type	Pistol
Working Distance	10 to 30 Feet
Working Environment	Indoors

9.6 Budget and Financing

Our team finds this project as a great learning opportunity to a real-world design challenge. Unfortunately, this project does not have a sponsor which could help in alleviating the production cost or guide us in the design process. As such, each member of our team shares part of the burden of the total cost of materials.

When it came to budgeting, our team took great strides in our research to select parts which were both high performance and cost efficient. In section 9.3, we take a deeper look at the overall cost and how we plan to get the parts for our project at a lower cost which would save some money and be equally effective. For example, the mobile application only costs the team in time rather than money.

By narrowing down multiple suppliers through the benefits and constraints of each product we managed to reduce the total cost of the project by a significant margin.

The current estimated cost for the entire project rounds up to around $400. While a bit too pricey we expect the actual amount to be somewhat cheaper. One factor is that basic circuit components can be found at prices lower than what is seen on online delivery suppliers. Also reusing pre-owned parts further helped to alleviate the grand sum.

As of right now, each member is expected to contribute around $100 for the components needed to build our project. Most members have paid-internships which will help in financing this requirement. We found that it would be in our best interest to just split the bill four ways when it comes to any purchase. That way no member takes an unequal amount of financial responsibility to the project.

In the case of a malfunction we increased the budget up to around $750 as a secure buffer to re-order our parts. Our lower end to the budget is around $250. We found this to be the breakpoint where we will not have diminishing returns if we decide to market our project in the future. Once we are able to successfully build our first prototype the total cost of production was drastically reduced. It is at this point where we were able to reuse previously designed portions of the project such as the software codebase.

We also are able to fine-tune the construction of our game board. Our first prototype is by no means be the most cost effective and efficient model. We likely found many parts that only bring redundancy into the design process. Also, we likely fund parts that suit the scope of the project much better than our current list of materials.

10.0 Personnel Management

In this section of our senior design document paper, we will discuss the steps that were taken into account so that our group could easily stay on track staying organized throughout both semesters. We created a month by month plan that allowed us to stay on top of our responsibilities. This allows our group to stay on schedule and ensure we are successful in creating a project that is fully functioning at the end of senior design two.

10.1 Milestones Discussion

May

In the month of May, we first began Senior design. Our goal was to create a group that was all motivated to create a fun and successful senior design experience. In this first month our main goal was to come up with ideas that we would like to work on. We set the last week of May to be our deadline for coming up with a project idea, and the weeks leading up to this would be focused on researching old projects that we may be able to build and improve on for our project.

Table 10.1.1: May Benchmarks

Task	Due date
Research and Planning	May 31
Deciding on final project	May 31

June

In June, we began to narrow down our research and focus it on the specifics of our project since we had determined what we were going to build. Once completing the research and planning part of the paper, we moved on to the executive summary section. We planned for that to be done by June 12, while still doing research for our project.

Researching for the project never really ends because we need to continuously be working and ensuring we are going to create fully functioning project. After completing the executive summary, we began work on the project description.

The project description was to be finished by June 12 as well, since we have four group members, we were able to divide the work for the project to be most efficient. By June 30th, we planned for the research related to our project, the hardware and software design, and the personnel management section to be complete.

Completing the sections up to this point would allow for us to meet our first project deadline of having 50% of our document complete.

Table 10.1.2: June Milestones

Task	Due Date
Research and Planning specific to project	June 7
Executive Summary	June 12
Project Description	June 12
Research related to our project	June 30
Hardware and software design	June 30
Personnel Management	June 30
50% of paper due	July 7

July

In July we continued work on our paper to meet the rest of our deadlines. We began work on the design summary and project prototyping sections of our paper, setting a completion date of July 14. This would allow us to meet our second deadline for the semester.

After completing all of this, we began work on the final few sections of the paper. In the last two weeks of July we began to work on the testing section, administrative content and project operation, and the summary and conclusion. We planned for the deadline for these last few sections to be completed by July 27. This gave us a few days to read over the document and check for any errors we may need to fix.

Completing these last few sections allowed us to complete our final senior design paper deadline. Throughout the whole process of writing this paper, we continuously do research and begin to order and test parts. Doing so allows us to be most prepared to begin building our final working project.

Table 10.1.3: July Milestones

Task	Due Date
Design summary	July 14
Project prototyping	July 14
Testing	July 27
Administrative content and project operation	July 27
Summary and conclusion	July 27
Read through document	July 31
Final paper due	August 2

We also created a general plan for Senior Design 2. Although there is a high chance that the dates of these deadlines will change, we had a general plan of attack. We may have events come up that could cause delays but having a general overview allowed us to stay on track, ensuring at the end of Senior Design 2 we have a fully functioning project.

So far what we have planned during senior design two is to have a general 3d printed model of the laser gun. With a physical representation of our project available, we can finally start to measure how large or small other components need to be. With the laser gun complete we can begin structuring the locations of each photoresistor on our game board. By then the mobile application should be at a functional but non-refined state.

Table 10.3.4: Senior Design 2 Milestones

Task	Expected Due Date
Laser Gun 3d printed	July 31
Laser Gun complete	August 7
First Mobile/ Web app prototype	Early October
Final Mobile/Web app	End of October
Photoresistor integration on target board	Early August
LED integration on target board	Middle of September
Bluetooth integration on target board	End of September
Microcontroller integration with photoresistors	End of September
Final testing	End of October
Final project complete	Middle of November

10.2 Group Ethics Discussion

This section of the paper talks about the topic of Ethics, which was discussed in depth at the Senior Design Bootcamp hosted by Engineering Leadership and Innovation Institute that our team attended. This bootcamp helped our group to help gain a deeper understanding of what helps make an enjoyable senior design experience for all our group members. One of the best activities we did were the values and ethics discussion.

The group values and ethics discussion helped our group to gain a better understanding of the individual values that each of our members have. During this activity we were to write down 10 values to us that were important and 10 values that were not so important from the large list provided to us. Once each of our group members had written down 10 values that were important to us, we discussed these among our group.

Discussing these values helped each of us to gain a better understanding of where each of our group members values stood. Once completing the individual values of our team, we discussed values that were important to us as a team. We discussed many different options that could be important to our team. Our group came up with the following values: Commitment, Efficiency, Dependability, Motivation, and Communication. After discussion, we came up with these values on the basis that following these values would allow for us to work together and create a fun and enjoyable experience for all the members in our group.

Having the values of our group align and agreed upon allowed for us to be successful in our project building. If not agreeing upon these, we could run into issues as far as arguing about what in our project is important, time management, and workload. We have all decided that the success of this project is the main priority for all of us over these next few semesters. This gave us the best Senior Design experience as a team.

10.3 Budget and Finance Discussion

In this section of the paper we look at the budget and how we funded this project. Since we came up with the idea for this project on our own, we covered the costs of this project amongst ourselves. There was no outside funding from any larger companies or groups, so as a group we have decided to split the total cost for this project 4 ways. We have estimated the initial budget to be approximately 400$ to 500$, adding extra costs for any extra parts we may need to purchase.

There are many different resources for finding used parts that costed our group less. These used products are in just as good quality as new products but come with a lower cost and allows us to keep our spending lower. We looked at websites such as a HobbyKing.com, as well as the company AloftHobbies. We have decided as a group to put a priority on these parts over new more expensive products. This should keep us right around our budget and end up costing each of the members in our group approximately 100$.

10.4 Discussion of Budget

In this section of the paper we look at the budget and how we funded this project. Since we came up with the idea for this project on our own, we covered the costs of this project amongst ourselves.

There was no outside funding from any larger companies or groups, so as a group we have decided to split the total cost for this project 4 ways. We have estimated the initial budget to be approximately 400$ to 500$, adding extra costs for any extra parts we may need to purchase.

There are many different resources for finding used parts that costed our group less. These used products are in just as good quality as new products but come with a lower cost and allows us to keep our spending lower. We looks at websites such as a HobbyKing.com, as well as the company AloftHobbies. Local hobbyist stores also have the potential to be stocked with all the materials we need for our project. These shops generally have bins full of used components which we can take advantage of to save extra money. However, unlike online stores we can't guarantee the current stock inside one of these stores. Still, we have decided as a group to put a priority on these parts over newer and more expensive products. This should keep us right around our budget and end up costing each of the members in our group approximately 100$.

10.5 Distribution of Work

Once we had determined all the members of our group and what our final project would be, we began to determine the strengths and weaknesses of each of the members of our group. This would allow us to split up the work of our project and have each of our members work on something they are comfortable with. Going to the Senior Design Bootcamp helped us to come up with this idea. What we came up with during this boot camp is illustrated in the table seen below.

Table 10.5.1: Group members with their contributions and roles

Name	**Strength**	**Weakness**	**Constraints**	**Role**
Travis	Microcontroller	Syntax	Work Load Time management	Bluetooth Microcontroller Software
Edward	Mobile Application	Hardware	Resources Time Management	Software
Triston	Microcontroller	Circuits	Resources Time Management	Game System
Caleb	Optic Hardware PCB Building	Software	Time Management Work Load	Optics

After we determined the strengths and weaknesses of each of the members of our group, we began to split up the workload of the Senior Design 1 paper first. We determined that after getting all the requirements for the paper we had approximately 8 full weeks to finish the document. Needing a minimum of 120 pages, each member would need to complete 30 pages by the end of the semester.

In the 8 weeks we must do so, if each member wrote 4 pages a week than by the end of the semester, we would finish the paper and have time to proofread. Having a photonics major, Caleb Dobias, he was tasked with working with the laser gun and the photoresistor. The rest of the group is composed of computer engineers, so we split up the tasks of microcontrollers, Bluetooth data transfer, and Web/Mobile app amongst the three of us.

After splitting up the work for writing our Senior Design 1 paper, we began to discuss and talk about how we would split up the work for the building and testing of our prototype.

Since Caleb is the photonics major, his focus the laser gun. He would end up designing the optics, targets and target PCBs, the gun's entire electrical system, and the guns casing. He would even do the programming for the microcontroller.

The rest of our team members being computer engineers would split up the rest of the work amongst us. Travis was to work on the microcontroller/pcb that would connect to the other five pcbs. This pcb needed to take into account the data sent from the four photoresistors that the laser triggered. It would then need to run the formula we came up with to give each shot a score based on accuracy and send this data to our mobile application via Bluetooth.

Travis worked with Edward as well to develop the mobile application; which Edward took lead on. Edward was setting up the database as well as taking project lead for how the application was developed. Edward worked towards creating a mobile application that was cross compatible with multiple device platforms such as Android and iOS. The communication between the microcontroller and the mobile application was extremely crucial to the success of this project.

Triston was working on the microcontroller within the handgun, ensuring the laser gun only allows a shot every few milliseconds, creating a short delay each time the trigger is pulled. He also worked on the wiring the board's PCB. We also needed to power LED, which we all worked together to do. Possibly using the pcb attached to each target to power these LEDs as well.

Each member had to make sure our individual sections communicate properly to our neighborhood component. Caleb's laser and photoresistor design act as both the signal and receiver to our project. Triston is in charge of powering and ensuring the board is capable of delivering the sensitive data securely to the microcontroller.

Travis must be able to take the encoded data and successfully pass it along through Bluetooth to the mobile application. Once the mobile application receives the signal from the photodiodes it must quickly store each and every pulse. It then is required to perform a series of logic guided by equations related to the relative proximity of each photodiode and provide a quality visual representation of the user's skill.

We have created an effective design cycle by dividing these individual pieces among the team and distributing the responsibility of each section to the member who is most comfortable.

11.0 Summary and Conclusion

In this section of the paper we discussed the success and all the work that went into senior design 1. The first semester was all about researching and beginning testing. We took a look at the success that we had as a team and what we needed to be successful going forward.

11.1 Overall Project Summary and Changes

The original goal of all of our team members was to create a project that would allow us to work with things we may not have had the chance to throughout college, while also creating a game fun for all to play. Caleb coming up with the idea to create a laser target game allowed for us to achieve all of these goals.

When beginning to research everything we needed for this project we had come to realize there were going to have to combine together to create the final project we had imagined. We spent the first few weeks of this summer semester researching all the parts that would be needed. We continued the research process throughout this whole semester, making sure we were creating our project to be the best that it could be.

After we completed some research on our topic, we began to narrow down the specifics of our project. We chose which components would be used for the laser gun, the targets, the game board, as well as the mobile application. Since we had to fund the project ourselves, we found the most effective and cost-efficient way to design and build our project.

We then began the testing portion of our project. We started to test the individual smaller components of our project, such as the laser and photodiodes. This testing gave us the values that we can expect when the player shoots the target during the game. Further testing was complete online, using online microcontrollers. This testing made building our project throughout senior design 2 much easier.

In our second semester of senior design we began to do more in depth testing. We needed to test our microcontrollers that we plan to use. Ordering our designed PCBs are also a main priority early on in the semester. We continuously tested each part of our project throughout the semester until creating our final project.

In our second semester we came across issues that we were also able to fix. A major issue we faced was with our LEDs. When purchasing the LEDs, we purchased a strip that was one single color and ran on a voltage of 7-12 volts. However, our ATMEGA2560 chip only was able to supply a voltage of 5 volts. To fix this problem, we added a transistor to act as a gate. When not supplying voltage from the pcb to the transistor the LED strip would be shorted. We would then supply voltage to the transistor from the pcb when the LED was desired to be turned on.

One of the bigger issues we faced was the beam shape of the laser from the gun. The most cost effective and kindest solution would be to use a different laser, where the frequency doubling of the laser would output a round beam shape. This laser was the lights88 green laser module, seen in Figure 1.1.1. Same dimensions as the other laser diode, however this used a frequency doubler and had a working voltage of 3.7 V. The most significant part of this change, however, is this diode required much more current (350 mA), which is much higher than what the microcontroller could safely output on one pin (35 mA). As such, we had to be more clever. The first attempt to solve it was through a MOSFET design. This was discarded after it was recommended that we instead control the current through the laser using a 2N2222a type transistor. This transistor had a beta value of about 100 and was rated for 0.5 W of power. This rating was not initially taken into account, breaking several of the transistors used. However, once the issue was realized, an adjustment was made to the circuit design and some high voltage load bearing resistors were used. We can see the laser and load bearing resistances as R3 and R4 on Figure 11.1.2 and the completed gun casing, minus the backshell, on Figure 11.1.3.

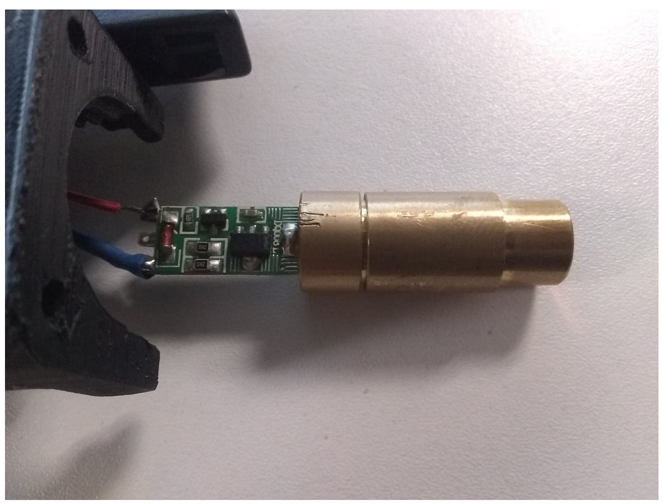


Figure 1.1.1: 532 nm Green Laser Diode

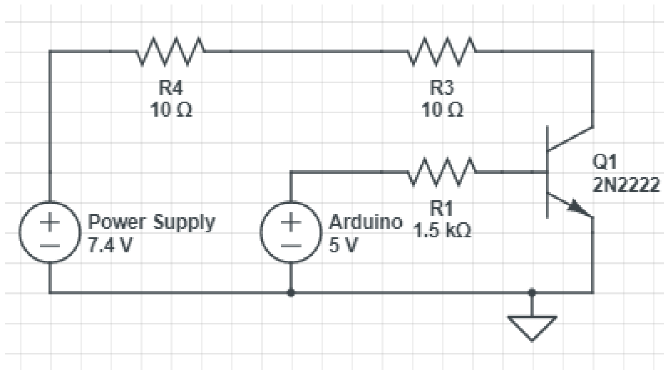


Figure 1.1.2: Circuit Diagram of transistor used with the new laser diode

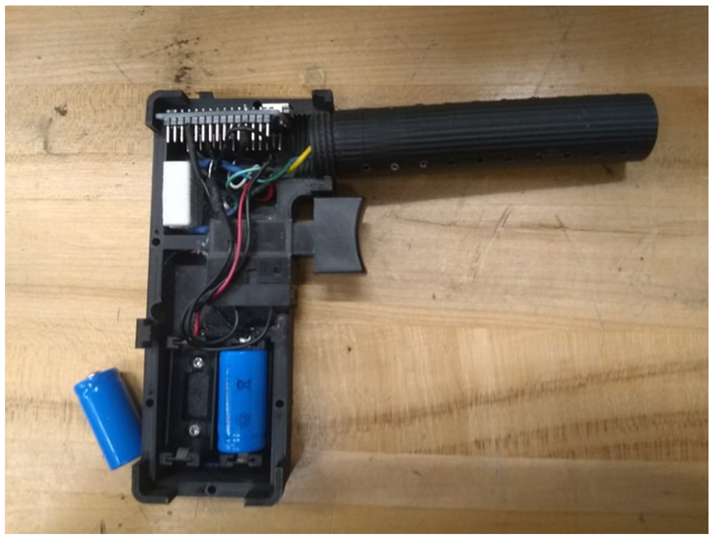


Figure 11.1.3: Completed gun with load bearing resistors (upper left)

We faced another issue when we attempting to order parts for our PCB which would allow us to have a fifth target. At first ordering the wrong parts, and then having to reorder from another country. When finally receiving the correct parts we were out of time and would not be able to set up the fifth target in time. We made a group decision to move forward with four targets that would all be fully functioning rather than have one target that would not work at all.

11.2 Project Success Summary

In our first semester of senior design we found a lot of success as a team. Once coming up with a project we would build, we began to effectively delegate the tasks up. We communicated well as a team, and assigned people tasks based on their strengths, which we determined during our Senior Design bootcamp.

Our team was on top of all of the assignments required of us during this semester. Each member was on top of their assigned work, and diligently worked to complete what was needed. The communication of our team was also very good. Each member was very reliable and could be counted on to do their work and be transparent about how much work was completed, and what they were still working on. Carrying this same work ethic allowed us to be successful in our second semester of senior design.

We as a team feel very confident over the progress, we made this semester in senior design one. Our team managed to form a solid foundation moving into senior design two while overcoming the shortened time limit that summer provides. Over the course of this semester we have commenced implementation and testing of components, solved our greatest design challenge of inter-part communication, and found similar projects which we used as guides to complete our goals.

Our team plans to utilize this time between semesters to continue working on our project. Parts have been ordered and are expected to arrive over the upcoming break. Our software is under development and had a working prototype by the start of the next semester.

Appendix A References

[1] “Arduino Bluetooth Basic Tutorial.” Arduino Project Hub, create.arduino.cc/projecthub/mayooghgirish/arduino-bluetooth-basic-tutoriald8b737.

[2] figure 7.1.3.1. “Circuits on Tinkercad.” *Tinkercad*, www.tinkercad.com/circuits.

[3] “The Guide to Bluetooth Modules for Arduino.” Into Robotics, www.intorobotics.com/pick-right-bluetooth-module-diy-arduino-project/.

[4] “USE a BUZZER MODULE (PIEZO SPEAKER) USING ARDUINO UNO.” Arduino Project Hub, create.arduino.cc/projecthub/SURYATEJA/use-a-buzzermodule-piezo-speaker-using-arduino-uno-89df45.

[5] “CdS Photoresistor Manual” Senba Optical & Electronic Co., LTD http://roscansat.com/wp-content/uploads/2019/02/GL55-Series-Photoresistor.pdf

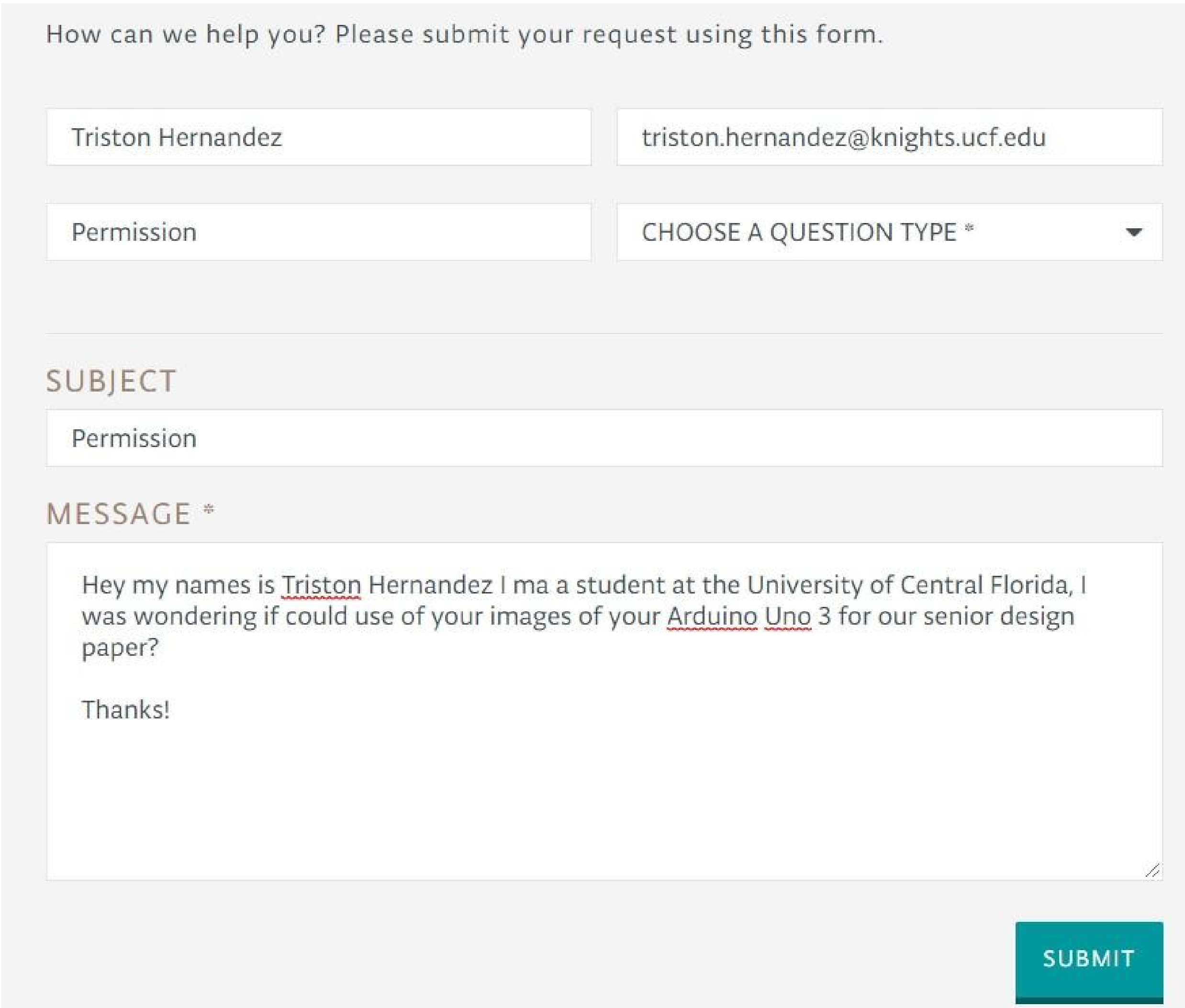
[6] “React Native Database - Choosing the right database for your React Native app” Simform, Hardik Shah https://www.simform.com/react-native-database-selection-guide/

[7]”Top Mobile App Development Framework in 2019” Mindinventory, Mehul Rajput https://www.mindinventory.com/blog/mobile-app-development-framework-2019/

[8]”How To Choose The Right Framework For Your Mobile App” Appinventiv, Peeyush Singh https://appinventiv.com/blog/choose-best-framework-mobile-app/

Appendix B Copyright Permissions

Arduino Store:



How can we help you? Please submit your request using this form.

Triston Hernandez

triston.hernandez@knights.ucf.edu

Permission

CHOOSE A QUESTION TYPE *

SUBJECT

Permission

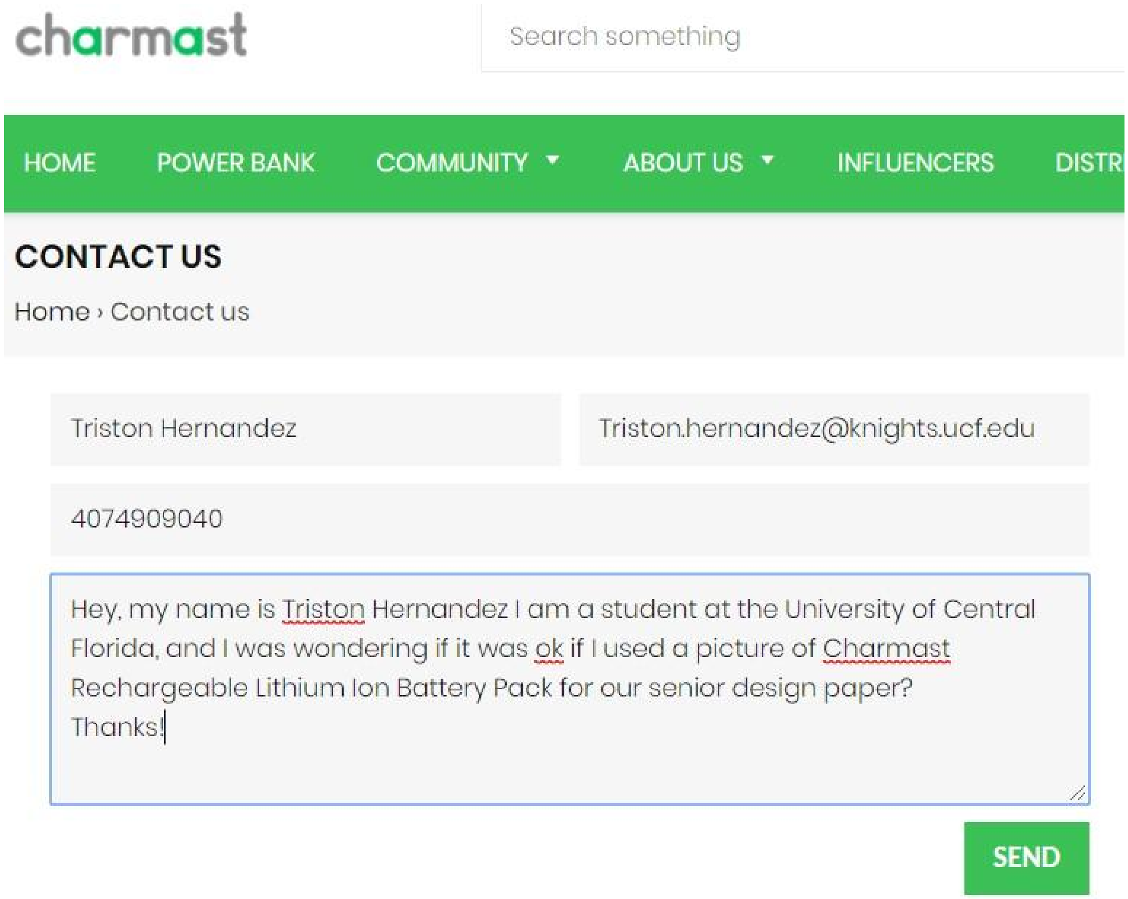
MESSAGE *

Hey my names is Triston Hernandez I ma a student at the University of Central Florida, I was wondering if could use of your images of your Arduino Uno 3 for our senior design paper?

Thanks!

SUBMIT

Charmast:



charmast

Search something

HOME | POWER BANK | COMMUNITY | ABOUT US | INFLUENCERS | DISTR

CONTACT US

Home › Contact us

Triston Hernandez

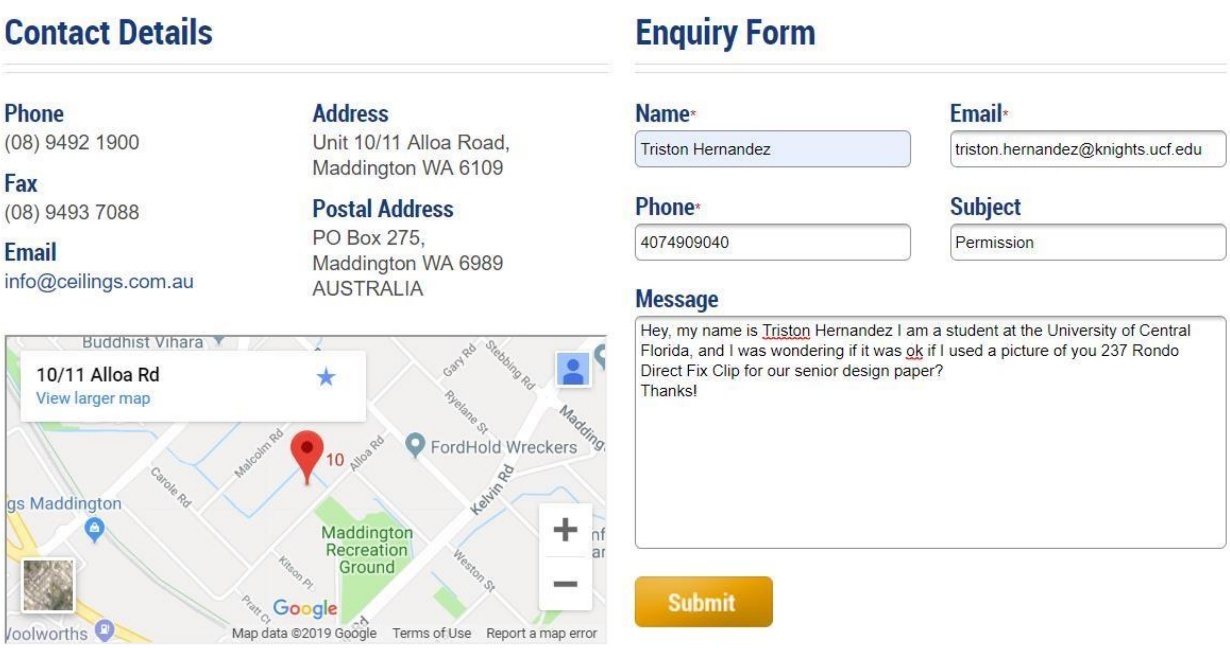
Triston.hernandez@knights.ucf.edu

4074909040

Hey, my name is Triston Hernandez I am a student at the University of Central Florida, and I was wondering if it was ok if I used a picture of Charmast Rechargeable Lithium Ion Battery Pack for our senior design paper?
Thanks!

SEND

Ceilings.com :



Contact Details

Phone
(08) 9492 1900

Fax
(08) 9493 7088

Email
info@ceilings.com.au

Address
Unit 10/11 Alloa Road,
Maddington WA 6109

Postal Address
PO Box 275,
Maddington WA 6989
AUSTRALIA

Enquiry Form

Name*
Triston Hernandez

Email*
triston.hernandez@knights.ucf.edu

Phone*
4074909040

Subject
Permission

Message
Hey, my name is Triston Hernandez I am a student at the University of Central Florida, and I was wondering if it was ok if I used a picture of you 237 Rondo Direct Fix Clip for our senior design paper?
Thanks!

Submit