

# GLO LASER Tag

A Fun, Affordable, At-Home LASER Tag Gaming System

Rayna Choi, <u>rchoi@nd.edu</u> Brianna Dewey, <u>bdewey@nd.edu</u> Allison Gentry, <u>agentry2@nd.edu</u> Leo Herman, <u>lherman@nd.edu</u> Kyle Tomasula, <u>ktomasul@nd.edu</u>

# May 3rd, 2025



Senior Design EE 40290 Department of Electrical Engineering Notre Dame, Indiana, 46556

No.	<u>Title</u>	Page
1	Introduction	4
1.1	Problem Overview	5
1.2	Solution Overview	6
1.3	Results	7
2	System Requirements	8
2.1	Blaster Module	8
2.2	Microcontroller	9
2.3	Transmitter/Receiver Subsystem	9
2.4	User Interface Subsystem	9
2.5	Power Subsystem	10
2.6	Web Server	10
3	Project Overview	11
3.1	Overall System	11
3.2	System Block Diagram	11
3.3	Blaster Module Design and Operation	12
3.4	Transmitter/Receiver Subsystem Design and Operation	14
3.5	User Interface Subsystem Design and Operation	16
3.6	Power Subsystem Design and Operation	22

3.7	Web Server Design and Operation	25
4	System Integration and Testing	27
4.1	IR Communication Testing	27
4.2	User Interface Testing	28
4.3	Power Subsystem Testing	31
4.4	Bluetooth/Web Server Testing	33
4.5	Total System Integration	34
4.6	Validation of Design Requirements	34
5	User Manual	37
5.1	Setup Manual	37
5.2	Operation Manual	38
5.3	Troubleshooting	39
6	To-Market Design Changes	40
7	Conclusions	42
8	References	43
9	Appendices	43

### 1. Introduction

As senior electrical engineering majors this semester, we have been tasked with developing a solution to a real-world problem by applying the knowledge and skills we have acquired over the past three years. This task is designed to help us bridge the gap between theoretical learning and practical problem solving, which will provide us with experience in designing and developing products that incorporate electrical devices, systems, and engineering knowledge. The project is intended to simulate the challenges that we will likely face in our future careers, and this will be our first opportunity to demonstrate our abilities to solve these problems, as many electrical engineers do across the world.

Furthermore, this project emphasizes the importance of teamwork, mirroring how teams of engineers work together in professional engineering environments. To meet the project requirements, we must create a solution that demonstrates what we have learned while incorporating or improving existing technologies. Lastly, we have been given a budget of five hundred dollars, ensuring our solution is cost-effective and practical.

Considering these specifications, our group has decided to design and create a portable laser tag system called Galactic Laser Operations (GLO). This solution provides the perfect platform to showcase our technical abilities and knowledge as it will incorporate many electrical components and subsystems, including sensors, wireless communication systems, power management and delivery systems, and integrated software. The laser tag system will provide a sufficient test for our team, as we will have to contemplate how to most effectively solve numerous engineering challenges, such as signal processing, embedded systems, user interface, and circuit design, all while staying within the five-hundred-dollar budget. We believe that this project aligns with the goals of the senior design class, as it simulates a real-world engineering problem, allowing us to come up with our creative solution that applies the theoretical knowledge we have picked up, while also allowing us to develop our ability to collaborate with others.

### 1.1. Problem Overview

Laser tag is a fun and engaging activity that has long been enjoyed by many people all over the world. It combines strategy, teamwork, athleticism, competition, and fast-paced action, making it a great way for friends and family to spend time together. Traditionally, laser tag has only been available at dedicated facilities with specialized equipment. These venues have provided this unique experience to players for years, however, the landscape of entertainment has greatly shifted in recent times. The rise of at-home entertainment options, including streaming services, gaming systems, and virtual reality goggles, has led consumers to become accustomed to entertainment they can enjoy in the comfort of their own homes. As a result, traditional laser tag venues have experienced a decline in popularity, with many opting for more convenient options compared to these at-home alternatives. While the idea of at-home laser tag systems has emerged as a more favorable alternative, current market offerings are far too expensive to be a justifiable purchase for the average consumer. For example, one of the leading at-home laser tag companies, Laserwar, sells its most basic kit for over \$1,300v(LaserWar). This kit

includes two laser tag guns and two headbands. Furthermore, many at-home kits lack the quality and immersive aspects that have made laser tag enjoyable for so many people. This has left a significant gap in the market for a product that delivers the same entertainment value as traditional laser tag while in the convenience of one's home at an affordable price.

### 1.2 Solution Overview

The solution that our team came up with is a battery-powered laser tag set that can be played anywhere at an affordable price, while maintaining all of the fun aspects that make traditional laser tag so exciting. The set will include two laser blasters, one for each player, and a web server that can be used to track scores. The blasters themselves house all of the devices necessary to make a laser tag system work. There is an IR emitter, an IR receiver, and an aiming laser mounted onto the front of the blaster, a trigger and buzzer mounted in the grip, a speaker on the inside, and LED strips on the sides. The blasters are powered by three AA batteries that can be turned on and off with an easily accessible switch. When the blaster is powered on, the LED strips flash three times to signify that the blaster has gone through its setup process and is ready to be used. During the game, when the trigger is pulled, one of four shooting sounds is randomly selected and played, the lights flash in a "running" pattern, the buzzer activates to simulate recoil, the aiming laser flashes a quick pulse, and an IR pulse is emitted. When the IR receiver has been hit, the blaster buzzes three times, the LED strips flash three times, and randomly selects and plays one of four death sounds. The logic of the blasters is controlled by an ESP32-S3 microcontroller, which also communicates through Bluetooth Low Energy to a web server for scorekeeping. On the host device, the players can connect to the blasters, and the web server will

6

display a real-time scoreboard that includes the number of times hit, the number of shots fired, and the accuracy for each player.

### 1.3 Results

In the end, our blasters worked almost entirely as expected. They were able to correctly trigger hits at a distance of 40 feet outside, and over 150 feet inside, far beyond our original goal of 30 feet. We believe that the loss of range outdoors is likely due to natural IR light from the sun that could be creating destructive interference. Furthermore, all of the feedback systems work, as the speaker, LED strips, aiming laser, and buzzer all act as intended when responding to inputs such as a trigger pull or a hit detected. The web server also works, in that it accurately displays scores, which include total shots, hits scored, and accuracy for each player. We tested the blasters, and the modules themselves are lightweight, easy to use, comfortable in one's hand, and durable enough to facilitate exciting gameplay. Lastly, all of our aesthetic goals were met, as the blaster is clearly a toy, yet it has a science fiction-inspired look to it, and the website interface also adds to the space theme. There were only a few minor drawbacks, but they do not severely hinder gameplay. On very rare occasions, the blaster's receiver would pick up a signal and count it as a hit erroneously, whose source was not from the other blaster. We are uncertain where these other signals come from, but we suspect very nearby cell phones and other groups using IR communication are likely the cause. Secondly, there is a very short delay in the website updating the hit counter; however, as players are running around while playing the game, and not constantly checking the scoreboard, this issue is considered negligible.

# 2. System Requirements

Before designing and implementing our attempt at laser tag, the group first set some system requirements to be met. By meeting these requirements, we can ensure that our product is a success and will deliver on the promises of a fun, affordable, and portable laser tag system. These requirements take various forms, with physical requirements, safety requirements, technical capabilities, and important features. The requirements are organized into sections based on which subsystem they are a part of: blaster module, user interface, transmitter, receiver, or control software.

# 2.1 Blaster Module

The blaster module involves the packaging of the other components into a final product that is both functional and aesthetically pleasing. For one, it must have enough space to house the user interface subsystem, the IR transmitter and receiver subsystem, the printed circuit board with the microcontroller, and the battery pack. In particular, the aiming laser, IR emitter, and IR receiver must be mounted in such a way that they are not obstructed. It also must be lightweight and fit comfortably in the user's hand to not be a hindrance to playing the game, while also being durable enough that if it were dropped, it would stay together. For safety reasons, it also must appear to be a toy so it is not confused with a real gun, while still appearing visually appealing to enhance the user experience.

### 2.2 Microcontroller

The microcontroller is what controls all of the logic functions of the blaster. Most importantly, it must have enough IO pins to communicate with the various other devices involved in the functionality of the blaster. Specifically, it must be able to take input from the IR sensors and trigger. It also must be able to output to the LED strips, speaker, aiming laser, buzzer, and IR emitter. In particular, the trigger and IR receiver must be connected to interrupt pins, so there must be at least two available. The microcontroller should also have an antenna to communicate through Bluetooth Low Energy, and be able to interface with the SPIFF file system to save and play audio files. Lastly, the microcontroller must have two cores, so that sound functions do not block other important functions when they are being deployed simultaneously.

### 2.3 Transmitter/Receiver Subsystem

The IR emitter and receiver subsystem is the most important part of the blaster, as gameplay cannot commence without them. The IR emitter must emit a pulse when the trigger is pulled, and it must have minimal spread while being directional and accurate at a distance of 30 feet or less. The IR receiver must correctly identify when it receives the pulse and send a signal back to the microcontroller. Therefore, the emitter and receiver must be able to communicate on the same frequency and be minimally affected by surrounding sources of IR noise.

### 2.4 User Interface Subsystem

The user interface subsystem involves all devices that perform actions in response to the user's inputs, and give feedback to the user when a signal is received. First, the trigger itself should be easy to pull, meaning that it is in an easy-to-reach place for people of all hand sizes, but not too easily actuated to prevent accidental shots. An aiming laser will be mounted on the front of the blaster and will flash a quick pulse when the trigger is pulled to show the user where they are aiming. The blaster will have two LED strips–one on each side–to provide visual feedback, a buzzer to provide haptic feedback, and a speaker to provide audible feedback when the blaster detects a hit or when the trigger is pulled. There should also be an ON/OFF switch that is easily accessible but out of the way to prevent accidentally turning the blaster on or off.

#### 2.5 Power Subsystem

The power subsystem will provide power to the blasters without needing to be hooked up to a laptop or other external power source. The power subsystem will need to provide both 3.3V and 5V power for the various devices safely and efficiently. The blasters will need to be able to run for at least 24 hours on a single set of batteries, and the batteries themselves must be able to be easily replaced.

#### 2.6 Web Server

The web server is how scores will be kept and shown during the game. The web server must enable the host device to connect to both blasters through Bluetooth Low Energy at a range of at least 30 feet. Once it has been connected and the game has started, the web server must keep track of each player's number of shots and how many hits they have scored. It must also calculate each player's accuracy and display all of these statistics in a visually appealing and intuitive scoring screen.

10

# 3. Project Overview

Our project aims to revolutionize the classic laser tag experience by creating a portable system that can be played anywhere. The kit includes two laser blasters that detect hits through infrared LED transmitters, IR receivers, and a microcontroller. The blasters include both the transmitter and receiver modules, as well as lights, sounds, and haptic feedback to let the user know when they have been hit or when the game has concluded. The system will also include a web interface that all the blasters will wirelessly communicate back to. The web interface is used to track scores.

### 3.1 Overall System

The project was divided into five different subsystems that all serve unique functions in the overall end product. The subsystems allowed us to divide the final product into smaller, more manageable parts with more specific objectives and requirements. They include a blaster module to house the components, a transmitter/receiver subsystem to enable IR communication, a user interface subsystem to provide feedback, a power subsystem to provide internal power, and a web server subsystem to track and display scores.

# 3.2 System Block Diagram



Figure 1. Overall System Block Diagram

# 3.3 Blaster Module Design and Operation

The main blaster was designed using three-dimensional CAD software that allowed us to import our PCB to visualize the precise assembly of the multiple pieces and how they would all fit together. Keeping in mind that the blaster was supposed to be different from a real gun, it was 3D printed in lighter colors, and it was designed with laser blasters from popular science fiction movies in mind, such as Star Lord's blaster from Marvel's 'Guardians of the Galaxy', as seen in **Figure 2**.



Figure 2. Star Lord's Blaster

The blaster was created with a modular design, as the grip, sides, and front pieces are all separate components that are screwed together to make the final blaster. The pieces create a hollow

shell that houses and protects the various electrical components inside, with holes to allow IR pulses to radiate from the emitter, light to radiate from the aiming laser, and IR pulses to radiate into the receiver. There are small gaps for the interior speaker's sounds to get out, as well as for the LED strips to wrap around to the sides. On the interior of the grip, there are cutouts for the trigger and buzzer to fit into. There is also a latch that opens, exposing the battery pack, allowing the user to easily replace the batteries when they die. There is also a gap for a switch to fit into, which allows the user to turn the blaster on and off without fully taking apart the blaster. Images of the CAD files can be seen in **Figure 3**, and images of the final shell can be seen in **Figure 4**. The blaster was designed in Fusion 360 before being split into multiple sections and printed in PLA using both the EIH and a personal BambuLab A1 Mini.



Figure 3. Images of CAD Model



Figure 4. Image of Final Product

### 3.4 Transmitter/Receiver Subsystem Design and Operation

The transmitter and receiver subsystems were designed to be able to send and receive IR signals at 38 kHz, meaning there was a 26.3 microsecond period. Beginning with the transmitter subsystem design, we selected a VSLY5940 IR emitter, seen in **Figure 5**. The VSLY5940 was selected because we found that it was directional, but still had a little bit of spread that made the aiming of the blaster a reasonable challenge without being too difficult during gameplay. The anode was connected directly to a 3.3V IO pin of the microcontroller, which modulated the input signal to the emitter high and low. Specifically, it set the IR emitter high for 13 microseconds, and then low for 13 microseconds, to match the 26-microsecond period that the receiver was looking for. It did this one hundred times to emit a pulse with a total duration of 2.6 milliseconds. A block diagram of the transmitter subsystem can be seen in **Figure 6**, and the code for the IR pulse can be seen in **Figure 7**.



Figure 5. VSLY5940 IR Emitter



Figure 6. Transmitter Subsystem Block Diagram



Figure 7. Code to send IR Signal

The receiver chosen was the TSOP32233, seen in Figure 8, as it is made to detect the

frequency we were targeting. The receiver block diagram can be seen in Figure 9. The receiver has

three pins: one connected to 3.3V, one connected to GND, and another signal pin that outputs to an

interrupt pin on the microcontroller. When an IR signal is not being detected, the signal pin outputs a

steady 3.3V, but when it detects an IR pulse, the signal pin is grounded. Therefore, the interrupt was attached as a falling interrupt, as each time the voltage falls, it means that an IR pulse has been detected. When the interrupt is triggered, it sets the 'HitFlag' high, and various feedback functions are performed.



Figure 8. TSOP32233 IR Receiver



Figure 9. Receiver Subsystem Block Diagram

# 3.5 User Interface Subsystem Design and Operation

The user interface involves multiple components, all of which are intended to provide feedback to the user when different events occur. The block diagram for the entire user interface subsystem can

be seen in **Figure 10**. The first component is the aiming laser, in which a KY008 laser module was selected, as seen in **Figure 11**. The KY008 was selected particularly because it was eye safe, and players would not have to wear eye protection while playing. The purpose of the aiming laser is to provide visual feedback to the shooter, so they know where they are aiming when they press the trigger and can adjust based on where they are missing. The aiming laser has 3 pins, but only two are used in our design. The positive pin is connected to 5V, while the ground pin is connected to the collector of an NPN transistor (MPS AO6), whose base is connected to an IO pin, and its emitter is connected to ground. This allows the microcontroller to control the aiming laser with only 3.3V output, while still providing a full 5V to the aiming laser so that it is bright enough at a distance.



Figure 10. User Interface Subsystem Block Diagram



Figure 11. KY008 Laser Module

The next component is the LED strips. There are two LED strips on each blaster, one attached to each side. The purpose of the LED strips is to provide visual feedback when they shoot or when they get hit. We selected the AO35 addressable LED strip, specifically the 15 LED/10cm strip as seen in **Figure 12**, which used the FastLED library to program. When the blaster is turned on, the LED strips flash three times to signify that the setup process is complete. During the game, the LEDs stay on and remain a solid color, either blue or green, depending on the team, except for when a player shoots or is hit. When a player pulls the trigger, the LED strips execute a 'run' function, in which the lights appear to move down the length of the strip, shown in **Figure 13**. When the player is hit, the lights will flash on and off three times. The LED strip has a positive pin that's connected to 5V, a ground pin connected to GND, and a data pin that's connected to an IO pin on the microcontroller.



Figure 12. AO35 LED Strip



Figure 13. Code to make LED Lights 'Run'

The buzzer is the next part of the user interface subsystem. The buzzer is a vibration motor, specifically the HD-EMC1203-LW20-R, shown in **Figure 14**. The buzzer has a positive pin wire that is connected to an IO pin on the microcontroller, and a ground wire connected to GND. The operation of the buzzer is simple: when the IO pin is high, the buzzer is on, and when the IO pin is low, the buzzer is off. The buzzer buzzes for a quick period of time when the trigger is pulled to mimic the effects of recoil, and it buzzes on and off three times when a player gets hit. The buzzer is mounted in the grip and is about the size of a dime.



Figure 14. HD-EMC1203-LW20-R Vibration Motor

The speaker and amplifier are the next components involved in the user interface subsystem. A MAX98357 Mono Amplifier connected to a 1W 8 $\Omega$  speaker is responsible for playing sounds when certain events occur. We chose the Adafruit STEMMA Speaker, as it was included in the breakout board that we purchased when implementing the subsystem prototypes, though any 1W 8 $\Omega$  on the market will work. The speaker itself can be seen in **Figure 15**, and an in-depth schematic of the amplifier can be seen in **Figure 16**. When the player pulls the trigger, one of four preset .wav files that correspond to the shooting action is randomly selected and played. Similarly, when a player is hit, one of four preset .wav files that correspond to the being hit action is randomly selected and played. The code for the speaker is reserved for the second core on the microcontroller, as the playing of sounds would otherwise block important functions. This code can be seen in **Figure 17**. The .wav files themselves are stored using the SPIFFS file storage system, so they are kept indefinitely even after power cycles and resets. The speaker is mounted to the inside of the blaster, but there are small slits that allow the sound to escape.







Figure 16. Schematic for the Amplifier

```
Core 1
void speaker(void *parameter){
   if (IRFlag) {
     if (now - lastSoundTime >= SOUND_COOLDOWN) {
       while (prevRandShoot == randNumShoot){
         randNumShoot = random(0, 4); // generates 0, 1, 2, or 3
       prevRandShoot = randNumShoot;
       switch (randNumShoot) {
         case 0: playWavFile("/PEW.wav"); break;
         case 1: playWavFile("/laser.wav"); break;
         case 2: playWavFile("/laser2.wav"); break;
         case 3: playWavFile("/ray.wav"); break;
       lastSoundTime = now;
   if (HitFlag) {
     while (prevRandHit == randNumHit){
       randNumHit = random(0, 4); // 0-3
     prevRandHit = randNumHit;
     switch (randNumHit) {
      case 0: playWavFile("/ohno_blasted.wav"); break;
       case 1: playWavFile("/wilhelm.wav"); break;
      case 2: playWavFile("/buzzer.wav"); break;
       case 3: playWavFile("/ouch.wav"); break;
```

Figure 17. Code for Audio Playback

The last component of the user interface subsystem is the trigger itself. The trigger is a DIYhz Single Pole Single Throw Push Button Switch, pictured in **Figure 18**. The trigger has two pins, one of which is connected to ground, and the other that is connected to an interrupt pin of the microcontroller. When the button is pressed, the IO pin reads a low value, so the interrupt is attached as a falling interrupt. The trigger is connected to the blaster's grip with a binding post so that the red cap sticks out, allowing the user to press it down.



Figure 18. DIYhz Single Throw Single Pole Push Button Switch

### 3.6 Power Subsystem Design and Operation

The power subsystem provides both 3.3V and 5V power to the various devices within the blaster, so that it doesn't have to be connected to a laptop or other external power supply. The power is supplied from three rechargeable AA batteries packaged inside the blaster, connected to a CRE22F4FBBNE ON/OFF switch, pictured in **Figure 19**. The batteries produce 4.5V when fully charged and are housed in the BC3AAW battery holder pictured in **Figure 20**. Therefore, two voltage regulators were needed, one to step the voltage up to 5V and one to step it down to 3.3V to accommodate the different voltage needs of the various devices. The PAM2423 boost converter is tuned using a voltage divider to output 5V, and the AZ1117 LDO regulator takes the 5V output of the PAM2423 and steps it down to 3.3V. These components were chosen with an estimate that the system could pull up to 2.4 A at full load. This is an overestimate, but the power system could support a load of up to 1.35 A of 3.3V power and 5.5 A through the 5V boost converter. A full block diagram of the

power subsystem can be seen in Figure 21, and schematics of the 3.3V and 5V regulators can be seen

in Figure 22 and Figure 23, respectively.



Figure 19. CRE22F4FBBNE ON/OFF Switch



Figure 20. BC3AAW Battery Holder



Figure 21. Power Subsystem Block Diagram



Figure 22. Schematic of the 3.3V Regulator



Figure 23. Schematic of the 5V Regulator

# 3.7 Web Server Design and Operation

The web server is used to connect to the two blasters and display scores. Namely, the web page will track each player's shots taken, hits scored, and their accuracy, and update and display these statistics in real time. Bluetooth Low Energy (BLE) is used to facilitate wireless communication between the two peripheral blasters and a central hub website that is responsible for managing connections and scorekeeping. Each peripheral blaster acts as a separate BLE server, and the central hub website functions as a client that connects to these servers. Each blaster tracks hits and shots through hardware triggers and IR sensors to keep track of two pieces of data per peripheral: the number of shots fired and the number of hits taken. It uses these values to update the scoreboard on the central hub via BLE. Note that the number of hits taken by a player is updated on the website as the number of hits scored for the other player. Each blaster's code has a trigger flag in the code that increases a counter whenever the trigger is pulled, which updates on the website as that peripheral's "Shots Fired". Each blaster also has a hit flag in the code that increases a counter when the IR receiver senses a signal, which then updates on the website as a "Hit Scored" for the individual who sent the signal. Two characteristic UUIDs are used for each peripheral–one for the IR transmission event and one for the IR receive event, which are set up with notification and write capabilities to allow for bidirectional communication. Custom server callbacks handle connections and disconnection events, which ensures that the blaster begins advertising automatically if the connection drops. A block diagram of the web server can be seen in **Figure 24**, and an image of the webpage can be seen in **Figure 25**.



Figure 24. Web Server Block Diagram



Figure 25. Image of Webpage

# 4. System Integration and Testing

Once the various components had arrived, we had to test them to make sure that they would work well enough to suit our needs. This involved an iterative process of trial and error during prototyping, and various issues were met and solved on the way to integrating the various subsystems, before finally putting everything together in an end product.

# 4.1 IR Communication Testing

Before our team decided to proceed with the VSLY5940 IR emitter, we tested another emitter, the TSTS7100. In the end, the VSLY5940 was selected due to its greater accuracy and farther range. The testing process began by using a generic ESP32 kitboard provided to all EE Senior Design students and breadboards to make connections to the emitter. Our original transmit code emitted the 38 kHz signal indefinitely, as opposed to the 2.6 millisecond duration as seen in the final code. On the other side, a receiver and LED were connected to another kitboard, and code was written so that if the receiver was sensing the IR signal, the LED would light up. One person would stand with the receiver while another person would move around the room to try and activate the receiver. This testing allowed us to get a good idea of the capabilities of the IR emitter, including its range and spread. We found that the IR emitter worked at a distance of about 45 feet. Progressing along, the code was changed to only produce 100 cycles of the 38 kHz signal, so the signal would only emit at the instant that the trigger was pressed. On the receiving end, an interrupt was attached, and testing was originally conducted similarly, just lighting up one LED if a signal was received and the interrupt was triggered. An image of the transmitter/receiver test board can be seen in **Figure 26**.



Figure 26. Transmitter/Receiver Test Board

# 4.2 User Interface Testing

Starting with the aiming laser, we quickly ran into an issue in that we mistakenly believed the module could be powered directly from the 3.3V output of the microcontroller. Once we found out that this was not the case, we found a solution using an NPN transistor to allow the 3.3V signal from

the microcontroller to deliver 5V to the aiming laser, after which we had no more problems with that module.

The LED strips were originally tested independently of the rest of the system, so the team could learn how to control them before integrating them with other functions. Furthermore, the LED strips began with very basic functions of flashing only when a player was hit, but over time, more functions were added, such as flashing on startup and the 'run' function when shooting to enhance the user experience.

The buzzer was very simple in its control and was originally tested independently before being integrated into the larger subsystem.

The speaker proved to be a difficult challenge for the team, as multiple iterations were implemented. Initially, a STEMMA Amplifier Audio Platform Evaluation Expansion Board—which included a PAM8302 amplifier and a 1W, 8Ω speaker—was used to prototype the audio system. However, this system did not have a digital-to-analog converter (DAC) included, which complicated the use of I2S for playing sound files. To try to work around this issue, code was written to play a .wav audio file from an SD card by using pulse-width modulation (PWM) to simulate analog audio output. It first initializes the SD card and reads the WAV file's header to ensure it is in the correct frequency and resolution. It then reads the audio samples from the file, maps the 16-bit values to 8-bit PWM levels, and outputs them in real-time using PWM. This allows a basic speaker to produce sound by using the PWM signal to mimic the behavior of a digital-to-analog converter. However, this resulted in poor quality sound, so we pivoted to the use of a MAX98357 I2S amplifier with the same 1W, 8Ω speaker. As we believed that sound files would be too large to allow for the use of SPIFFS filesystem, we prototyped a system that utilized this amplifier module, an SD card adapter, and an SD card for playing sounds. For every attempt, we started with playing a simple tone using the amplifier and speaker, then moved on to playing .wav files.

During testing, one amplifier was damaged as it was accidentally used above its maximum voltage. Furthermore, there were many issues with file storage and delivery to the speaker. In prototyping, SD card file storage and retrieval were successfully implemented. All necessary sounds were able to be called and played when the appropriate flag was triggered. Upon the arrival of the PCB, the mounted SD card reader was tested, but it would not allow for the SD card to be initialized properly. One initial problem was that the buffer used between the SD card and the microcontroller was for the wrong voltage (5V instead of 3.3V). Upon further research, we discovered that an SD card should be able to directly communicate with a 3.3V microcontroller, but we were unable to verify that on our board. To pivot without requiring additional PCBs to be made without the SD card mount, we opted to use the SPIFFS file system. Like many of the other devices, the speaker was tested independently before we tried to integrate it with the other functions, but there were issues with this as well. Specifically, there were issues with one I2S library that worked with the speaker by itself, but it conflicted with other parts of the code, and so this library was not able to be used in the main code. In the end, the libraries were determined to be unnecessary/other libraries could be used instead, and the speaker was finally successfully integrated with the other subsystem components.

30

The trigger was very simple to implement. First, we set up a code that just printed a message to the serial monitor when the trigger activated the interrupt, before integrating this interrupt with the rest of the code. The logic of the code also proved to be a big challenge to the group. The components of the user interface subsystem were added one at a time, including their code, and many print statements were included to make sure that certain functions were happening correctly. First, the trigger interrupt was integrated with the IR emitter. Then the LED strips were added and integrated with the trigger and receiver interrupts. After that, the buzzer and speaker were integrated with those interrupts, and lastly, the aiming laser was integrated with the trigger interrupt. Many race cases came up, specifically due to the use of two cores, as well as issues with cooldowns and reattaching interrupts correctly. These led to many instances where the sounds would play, or lights would flash, or the buzzer would buzz when they were supposed to be off during the one-second cooldown after every trigger pull. In the end, many of these issues were solved by creating many flags and cooldown variables that ensured these devices would not erroneously activate with 'if' statements. The initialization of these cooldown variables and flags can be seen in Figure 27. Other issues that came up included that occasionally, the receiver would pick up its own emitter's signal, and so the receiver interrupt is briefly turned off while the IR pulse is sent.

Lastly, when we moved from the kitboards to our unique PCB, we had trouble uploading the code. This issue was solved by removing an unnecessary pull-up resistor and using the boot and reset buttons to force the microcontroller into a programmable state.

31



Figure 27. Initialization of Cooldown Variables and Flags

### 4.3 Power Subsystem Testing

The next subsystem to be tested and integrated was the power subsystem. In order to do this, we attached a battery pack with three AA batteries to an ON/OFF switch and a regulator board, seen in **Figure 28**, to deliver the correct voltages. This regulator brought the 4.5V from the batteries up to 5V using the same schematic shown in **Figure 23**. In order to integrate these subsystems, a cardboard prototype blaster was built that housed the transmitter and receiver, all user interface components, and the power subsystem as seen in **Figure 29**. During testing, it was discovered that the regulator voltage was dropping to 4V when a load was applied. It was hypothesized that this was due to a poor connection to the inductor on the protoboard, and this issue did not occur on the actual PCB.



Figure 28. Prototype Voltage Regulator



Figure 29. Prototype Blaster

# 4.4 Bluetooth/Web Server Testing

To develop the final website for tracking laser tag scores, various tests were conducted to implement and refine real-time data communication between the blaster and the central hub. The website acts as a central BLE client that connects wirelessly to two peripheral blasters, each functions as a BLE server. The Bluetooth connectivity with the Web Server also proved to be a major obstacle. We experienced difficulties with connecting to multiple boards at once, as well as updating and displaying the player stats. We began with very simple steps and added components a little bit at a time. To begin, we connected to ESP32 kitboards and sent simple serial print messages to each device. Once this was successfully established, we integrated it with a shot counter on the PCB that updated every five seconds, to see if the communication between peripherals and the central hub could keep track of these changes. Upon completion, we uploaded the code to the PCBs created for this project to begin integrating with the rest of the code to ensure that the website could interact with the trigger and hit flags in the software of each finalized blaster, and to confirm that each shot fired and each hit taken could be correctly counted when they were detected. When incremented, these counters are then communicated via BLE characteristic UUIDs configured for write and notify access and displayed on the website as total shots, hits scored, and calculated accuracy. From there, tests confirmed that when one player registers a hit or a shot, it correctly updates on the website's scoreboard. Additional testing was done to ensure reliable reconnections were implemented using custom server callbacks that re-enable advertising if a blaster disconnects.

#### 4.5 Total System Integration

Once all of the subsystems had been completed and integrated, it was time to integrate and package everything into one comprehensive blaster module. As expected, the printed circuit board fit neatly into the 3D printed shell, as did the button, speaker, trigger, emitter, receiver, ON/OFF switch, and aiming laser. However, there were some minor issues with the latching mechanism for the battery

34

pack, which was solved in a second iteration, and the LED strips broke when bent too sharply, so their placement had to be slightly adjusted. An image of the shell with all of the components inside can be seen in **Figure 30**. The last step to complete the blaster is to put the shells together and screw them together.



Figure 30. Blaster Shell with all Components Inside

# 4.6 Validation of Design Requirements

Looking back at our system requirements set in section two, we see that they are broken down into six sections. The first section is the blaster module, which requires that it is large enough to house the other subsystems, which it does, and the emitter, receiver, and aiming laser are mounted in a way that they are unobstructed. Continuing, durability was another priority, and the outer plastic shell is certainly thick enough to protect the inner components without being too heavy. Comfort is another requirement, and although this is a subjective measurement, group members of various hand sizes all held the blaster in their hands and agreed that it was comfortable to hold. Another requirement was style and aesthetics. Members of other groups all said that they liked the look of our blaster. Lastly, our toy blaster needed to not resemble a real gun, and again, while this is a subjective goal, ChatGPT was able to correctly identify our blaster as a toy gun when placed next to an image of a Colt 1911, as seen

### in Figure 31.

Let's review both images carefully:

• First Image (gray device):

This looks like a **3D-printed object**. Its design is quite blocky and mechanical, not at all like a real firearm. It has visible screws, a hinged compartment, ventilation slots, and a very "toy-like" red button. Overall, it does **not** resemble a functional gun — it looks more like a prop, a toy, or a housing for electronics (such as a sensor or DIY project enclosure).

• Second Image (black handgun):

This is a **Colt 1911-style pistol**. It is a real firearm design, although without context (such as where the photo was taken or product details), it's impossible to say if the specific item in the picture is a *real gun*, a *replica*, or a *toy*. However, the Colt 1911 is a widely recognized real handgun.

#### Summary:

- First object: Almost certainly a toy, prop, or electronic device housing.
- Second object: Real firearm model; could be real or a realistic replica depending on the context.

Would you like me to help you further analyze if the second one could be a toy or airsoft version?

Figure 31. ChatGPT Correctly Identifies our Blaster as a Toy

The next set of requirements concerned the microcontroller, of which we selected an

ESP32-S3, pictured in Figure 32. The requirements concerned pin allocation, including interrupt

pins, memory for SPIFF files, and having two cores as well as an antenna, all of which were met.



Figure 32. ESP32-S3 Microcontroller

After the microcontroller requirements came the transmitter and receiver requirements. The IR emitter can trigger the receiver at a distance of over 150 feet inside and 40 feet outside, exceeding our initial goal of 30 feet, which confirms that they can correctly communicate with each other.

The next set of requirements is the user interface requirements. The first requirement is that the trigger is easily accessible, while guarded enough to prevent accidental shots. Once again, all group members agreed that the trigger is easy to actuate, and although misfires are possible, they are a negligible issue, so these requirements are considered correct. The next set of requirements is similar for the ON/OFF switch. Seeing as the ON/OFF switch is on the top of the gun, it would be very hard to accidentally turn the blaster off while playing, but it is still very accessible when the user wants to turn it on. Lastly, the user interface requires that all feedback systems, aiming laser, LED strips, speaker, and buzzer, activate properly when a player is hit or pulls the trigger, which have all been successfully implemented.

The power subsystem also had its own set of requirements. They mandated that the power must be regulated to both 3.3V and 5V, and that it is done safely and efficiently. These requirements have been met, seeing as all of the 3.3V devices function properly, including the microcontroller,

37

emitter, receiver, speaker, and buzzer, and all of the 5V devices also function properly, which includes the aiming laser and LED strips. None of the devices ever gets hot or burns out, which confirms that the voltage regulation is accomplished safely and efficiently. While the batteries have not been tested for a 24-hour time frame, they seem to have lasted quite a long time while testing the blaster, and in the spirit of the requirement, they last for a sufficiently long time. Continuing with batteries, the latch where they are stored is very easily opened by unscrewing one screw, where they can be replaced if necessary.

The last set of requirements are the web server requirements. All of these requirements have been met, as the host device can connect to both devices within at least 30 feet, and it does correctly display the score and player stats, including shots taken and accuracy percentage.

# 5. User Manual

The laser tag set will include two blasters already fully assembled and functional.

### 5.1 Setup Manual

As both of the blasters come preassembled, there is not much in the way of setup, but there are a few steps.

Place three fully charged AA rechargeable batteries in the battery pack under the latch. A screw
will need to be undone to pull the latch back. Replace the screw once batteries are inserted.
Turn the device on with the switch at the back of the blaster. When the lights flash three times,
the setup process has been completed.

- 2. Download the 'central\_hub\_website' folder and open it by double-clicking the .html file OR using the "Open Folder" button in Visual Studio Code. Then install the Live Server extension. Open the "index.html" file in the folder and right-click on the area where the code is and select "Open with Live Server". Either option should open the scoreboard in a web browser.
- 3. Once on the central hub website, click "Select Device 1" and pair with the "Blue Blaster". Then hit "Connect Device 1". In the device 1 log, you should see messages that confirm you are connected. Repeat this step for device 2 with the "Green Blaster".
- 4. Optional: Before beginning gameplay, try shooting at one another with each blaster to ensure that the scoreboard is updating as expected.
- 5. Have fun!

### 5.2 Operation Manual

Once the devices have been powered on and connected to the host device, you are ready to play! The rules of the game are simple:

1. To fire your blaster, press the trigger button down. Running lights, a shooting sound, and a quick buzz will confirm that the trigger pull was recognized and an IR signal was sent. An aiming laser will quickly flash to let you know where you are aimed, and to help you adjust your aim if necessary. Keep in mind that you are only allowed to press the trigger once per second to prevent spamming shots. If you press the trigger during the cool-down, no feedback will be activated, and no IR pulse will be sent.

- 2. If you are hit by your opponent, your blaster will flash the lights and buzz three times, and a 'blasted' sound will play. While these feedback functions are running, your IR emitter will be turned off, and you will not be able to shoot until your blaster stops flashing and buzzing. You are not allowed to cover your IR receiver while playing, as that is considered cheating.
- 3. The web server will display the scores in real time, including hits scored, shots taken, and your accuracy percentage, calculated by the number of hits scored divided by the total number of shots.
- 4. There is no set time limit for the game, so play as long as you want. The winner is determined by the player with the most hits scored at the end of play. If you want to reset the score and start a new game, there is a "Reset" button on the web server.

### 5.3 Troubleshooting

If you notice that one of the blasters is not operating correctly, multiple issues could be happening. If your blaster fails to light up when the switch is turned on, or the LED strip or speaker seems faint, make sure the batteries are fully charged. If one of the user interface devices is not activating correctly, check to make sure the wiring is still intact. If problems persist, it may be necessary to unscrew the shells and press the reset button while powered on. If this doesn't work, then it may be necessary to download the final code files from the GLO Senior Design Website and reupload them.

# 6. To-Market Design Changes

Due to the time and budget constraints placed on this project, combined with the team members' relative inexperience in large-scale projects like this, it is no surprise that there are some modifications necessary before this product can be offered on the market.

The first change to make to send the final product to market would be to standardize the 3D printing process. Currently, the two blasters have similar, but slightly different modular designs that are put together to build the final product. Furthermore, they are printed in multiple different colors of plastic, and we believe it would be best if they were all printed in the same white plastic material. Some of the printing is also rough on the edges, and a process such as injection molding might be a better way to make the shells. Continuing with the aesthetic design of the blaster, it would be nice if there were decals painted on the blaster, along with the LED strips, to give it a little more visual appeal.

Moving on to the functionality of the blaster, we had some issues involving the receiver being randomly triggered by other groups who were also using wireless communication during our testing. Therefore, we believe it would be best to use 27MHz or 49MHz, as these are unlicensed portions of the radio spectrum often used by toys such as remote control cars(liveabout), and we would also use a more selective receiver that only receives at our target frequency.

We would also like to adjust parts of the game itself to add more fun. For one, we would like to include sets that have more than two blasters, and have there be a way to set multiple players on a team, or have it be every man or woman for themselves. To accomplish this, we would need to modulate the IR signals so that each blaster has its own unique signature to correctly score hits. We would also like to include different game modes and scoring mechanisms, such as only being given a limited number of

41

shots, a limited number of lives, a game timer, or even power-ups. These different game modes and teams would be able to be set up from the main web server. We would also like to create a single-player set that has a training mode, which would include one blaster and multiple targets that could be set up and would also give feedback, such as flashing lights and sounds, which would allow players to improve their aim. We would also like to include an OLED or LCD screen, so each player can see their own stats without having to look at the main scoreboard.

The printed circuit board also needs some small adjustments. Currently, a transistor is soldered to wires between the board and the aiming laser to provide 5V power. This could be rectified with a surface mount transistor that would take up less space and have a safer connection. There is also a pull-up resistor that had to be unsoldered, as it prevented the board from going into a programmable mode, and it would be best to remove it from the board file for future iterations. There is also a surface mount SD card holder, as using an SD card to store audio files was considered, before making the ultimate decision to proceed with the SPIFFs file system. Therefore, this is an unnecessary part and can also be removed from the board file. Lastly, while only a small component, we would like to replace the push button trigger with an actuator that more closely resembles an actual trigger.

Additionally, we would like to have the ability to adjust the volume. Currently, the volume is dependent on the gain of the sound files themselves, and can to be changed by the user. This could be done with a potentiometer hidden on the board itself, which could be adjusted by opening up the blaster.

42

# 7. Conclusion

In conclusion, the GLO Laser Tag Senior Design project proved to be a worthy test for the team members. The process involved many aspects of electrical engineering and design thinking, such as designing a printed circuit board, designing a power subsystem, evaluating various devices and selecting the best option given the price and necessary features, coding and integrating these devices to interact with each other through a microcontroller, and even a little bit of 3D CAD experience. These tasks allowed the team members to flex their problem-solving skills that have led them to become great engineers, as well as apply the theory that they have learned in various courses at Notre Dame. It is sufficient to say that the team members were up for the challenge and were able to successfully meet almost all of the specified design requirements set out at the beginning of the semester.

The project also gave the team a chance to develop and show off some of their "soft skills", such as their ability to work efficiently with others, while minimizing conflicts and learning to work together towards a common goal, as well as meeting project deadlines and staying within a specified budget. These are all skills that are necessary for a successful career in engineering, as real-world engineers face many of these same challenges daily. Therefore, this capstone project also serves as a stepping stone for the team members to begin their careers, which will hopefully lead them all to do great things.

# 8. References

"LASERWAR Laser Tag." Top 6 Home Laser Tag Sets, laserwar.com/equipments/laser-tag-at-home.

Accessed 28 April. 2024.

"Radio Frequencies in the US for Radio Controlled Vehicles." liveabout dotcom,

https://www.liveabout.com/radio-frequencies-in-the-us-for-radio-controlled-vehicles-2862530

Accessed 28 April. 2024.

# 9. Appendices

Any useful file, link, datasheet, or other document have been included in the appendix

9.1 **Printed Circuit Board** - Download Link

# PCB Schematic:



PCB Design:



- 9.2 CAD Files Zip File
- 9.3 Final Code Zip File

# 9.4 Important Datasheets

TSOP32233 (IR Receiver)

VSLY5940 (IR Emitter)

KY-008 (Aiming Laser)

Adafruit STEMMA Speaker (Speaker)

HD-EMC1203-LW20-R (Vibration Motor)

BC3AAW (Battery Pack)

AO35 (LED Strips)

<u>CRE22F4FBBNE</u> (ON/OFF Switch)