Laser Tag Glove

Team 8 - Keng Yan Lim, Alex Korfel, Carlos Lara
ECE 445 Final Paper- Spring 2018
TA: Jacob Bryan

Abstract

Team 8 engineered a portable laser tag glove and vest combination that allowed for easy-to-join games of laser tag. This document explains the design and implementation of the various subsystems required to make a functional product. The laser tag glove uses Infrared (IR) LED emitters and IR receivers to fire and register “shots”, respectively. Game modes and scores were implemented using the sending and receiving of information through a portable central hub. That same information was then displayed on the players’ and central hub’s LCD screens. Accomplishments, future considerations, and ethical concerns involving the end product are also discussed.

Table of Contents

1 Introduction

1.1 Purpose

Laser tag, a fun game, often requires a facility with clunky equipment in order to play. It can be expensive and inconvenient to go to those facilities in order to play a game of laser tag. Our goal was to increase laser tag portability and immersion. To do this, we developed a glove that contained the laser, a vest to detect the laser, and a wrist gauntlet displaying game information. As each player will wear a laser glove and small vest, our game ensures portability and can be ready to play any time. The ability to play a game anywhere will boost immersion and the fact that the player needs to make a “finger gun” to fire their laser, will make the game more entertaining and novel. A wireless transceiver was used to sync all players into the game and keep track of score.

1.2 Functionality

To ensure the project’s success, we constructed three high-level requirements that would adequately cover the overall goal of our design. The high-level requirements are listed below.

- A player’s shot (IR pulse) will register as a hit by the IR receiver if they shoot the receiver at 20ft or closer.
- The glove and vest pair must be small and compact enough to be easily moved or carried compared to those on the market. Preferably, more than four glove and vest pairs can be carried in a backpack.
- Create a playable game of laser tag for 2 players that can be expanded to include 20 total players.

One of the first considerations when making the game was how far apart the players would be from one another. Laser tag facilities often have players moving through small corridors, so we felt that 20 ft would ensure a fun distance that allowed for more movement. Therefore, this requirement actually ensures that a fun game can be played rather than having the players too close. The overall purpose of our project also wanted to make the game portable rather than relying on facilities. Our group interpreted “portability” as the ability to easily move the gear from place-to-place. This meant that the equipment could be carried in backpacks with ease. Each component’s size was taken into consideration in order to ensure that it would be portable, even with all of them combined together. Lastly, the overall design had to be easy to join and needed to have the ability to include more players. As long as a player owns a glove and vest pair, they can easily join a game with their friends. When considering technical limitations and overall fun, we felt that 20 total players would meet our requirement.

1.3 Subsystem Overview

1.3.1 Block Diagram for Player

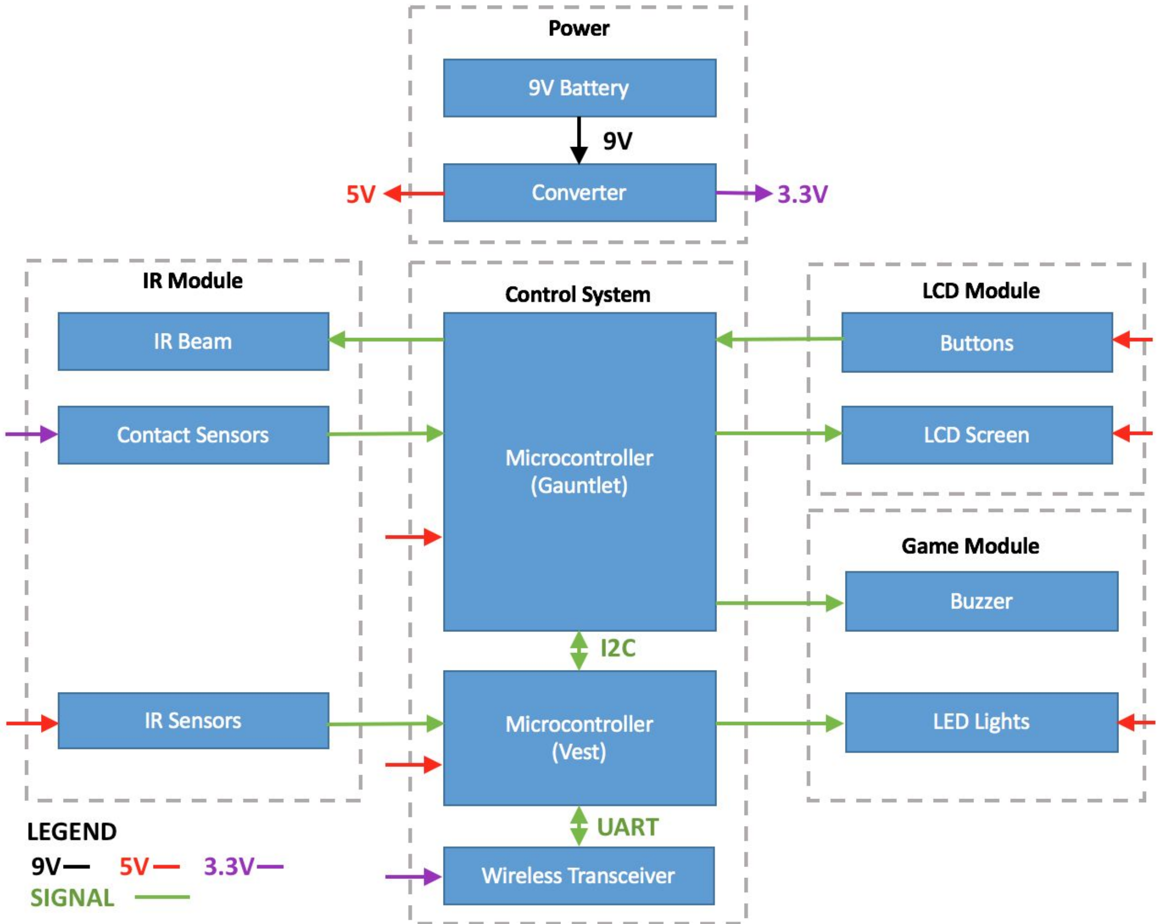


Figure 1: Laser Tag Glove Block Diagram

The player's block diagram as referenced in Figure 1 can be split into 5 main modules. They are the power module, IR module, control system, LCD module, and game module.

The power module is mainly in charge of converting 9V from our battery to 5V and 3.3V to power various components in the system. For example, the contact sensors and wireless transceiver will require 3.3V to operate.

The control system will be communicating with a central hub wirelessly and it consists of microcontrollers that will coordinate the behavior of the IR module, LCD module, and game module. For example, it changes the colors of the LED lights whenever a valid signal is read from the IR sensors.

1.3.2 Block Diagram for Central Hub

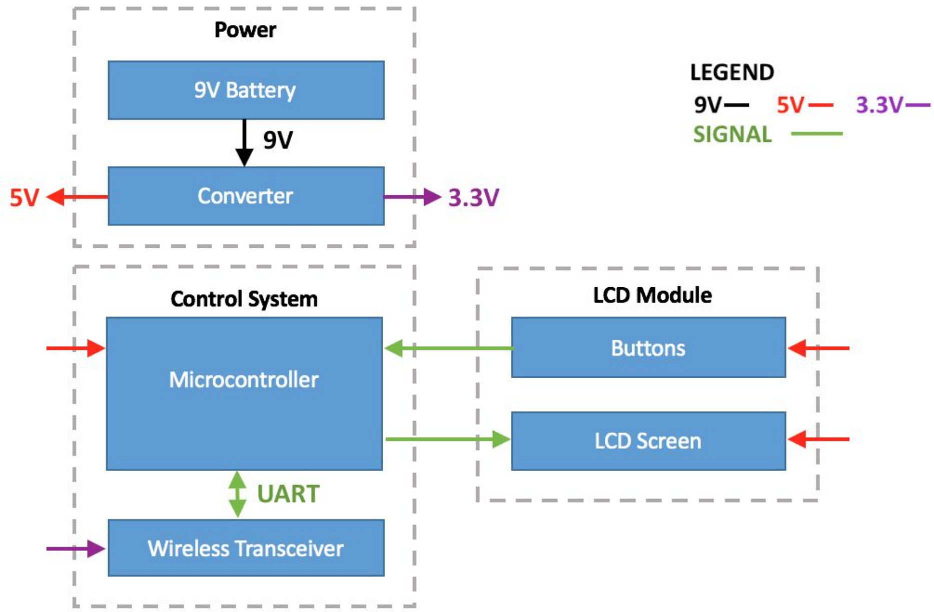


Figure 2: Central Hub Block Diagram

The block diagram as referenced in Figure 2 is very similar to the player's block diagram, with the exception of the IR and game module. In addition, the LCD module of the hub will have more buttons to enable users to interact with a game menu.

2 Design

2.1 IR Module

The IR Module consists of the IR sensors, IR transmitters, and contact sensors. The module as a whole is responsible for acting as the 'gun' and letting the system know that the player has been shot.

2.1.1 IR LED

The IR laser in our proposal has been changed to a plano-convex focused 950nm IR LED. This decision was made due to unreliable NEC pulsing, excessive required accuracy (player required to precisely aim at IR receivers), and resulting sub 20 ft range. This new IR LED is aimed and pulsed towards the opponent's IR sensors using an ATMEGA328P. This IR is transmitted in defined pulses that allows the IR sensors to differentiate beams coming from the different players. The IR receiver will be looking for a carrier frequency of 38kHz and thus, the IR emitter will be modulated at the same 38kHz. The data pulses will be sent according to NEC protocol (founded by Nippon Electric Company). An example of NEC can be seen in Figure 3 defining the pulse required to depict a logical "1" and "0" [1].

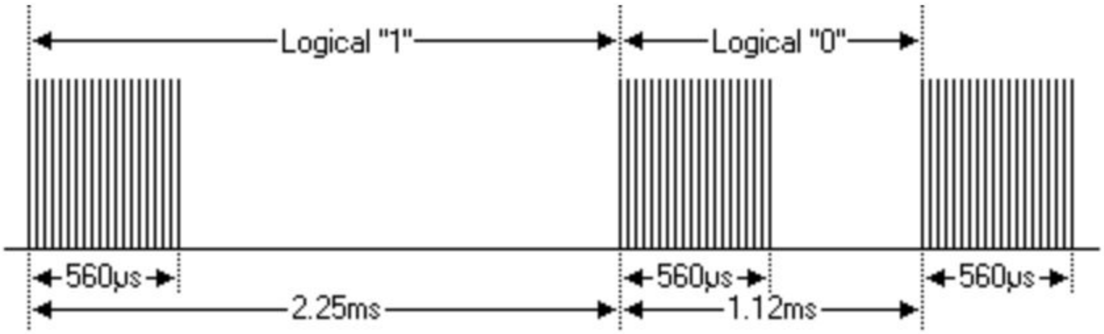


Figure 3: NEC Protocol Pulsing [1]

Once the IR LED was pulsed according to NEC and the IR receivers were confirmed to read the emitted pulses (test in “IR Sensors” section), the range of the IR LED was tested. Preliminary tests were done by connecting an IR receiver to an Arduino and reading the output using the Arduino serial monitor as the IR LED was pulsed using another Arduino. First the IR receiver was placed in the hall and shots were made at 5ft intervals. If the shot was correctly registered (binary 1 and 2 for player 1 and 2 respectively), we ran towards the receiver as we fired to confirm reliable transmission. This was done until a max distance was registered. The final reliable transmission distance was 32ft while max transmission distance was 55ft (NEC pulse was read but not the correct code). A picture of the final assembled max distance test can be seen in Figure 4 below. The 32ft was re-confirmed with the final vest assembly.

Figure 4: Max Distance Test

2.1.2 Red Laser

As IR is invisible to the eye, a visible red laser sight was added to aid in aiming. In addition to aiming, the red laser acts as an indication as to the state of the “safety”. When the pinky, ring, and middle-fingers are touching the palm, the red laser will turn on indicating the “safety” is off and the player can shoot. When the red dot is off, the “safety” is on and the player cannot fire their gun.

2.1.3 IR Sensors

These sensors are the VISHAY TSOP38238 which are responsible for receiving the pulsed IR beams mentioned above that indicate whether the player is shot. These pulses are the players’ unique binary tag. The IR receivers have a carrier frequency of 38kHz allowing for NEC protocol. This carrier frequency are represented as vertical lines in Figure 3 which means that the IR LED output is set high by switching at 38kHz for that defined duration [2].

To verify the IR transmission capabilities between the IR LED and receiver, the IR circuit in Figure 15 was assembled on a breadboard and final printed circuit board (PCB), and connected to an oscilloscope. The IR LED was then pulsed according to NEC using an Arduino and the following waveform was achieved.

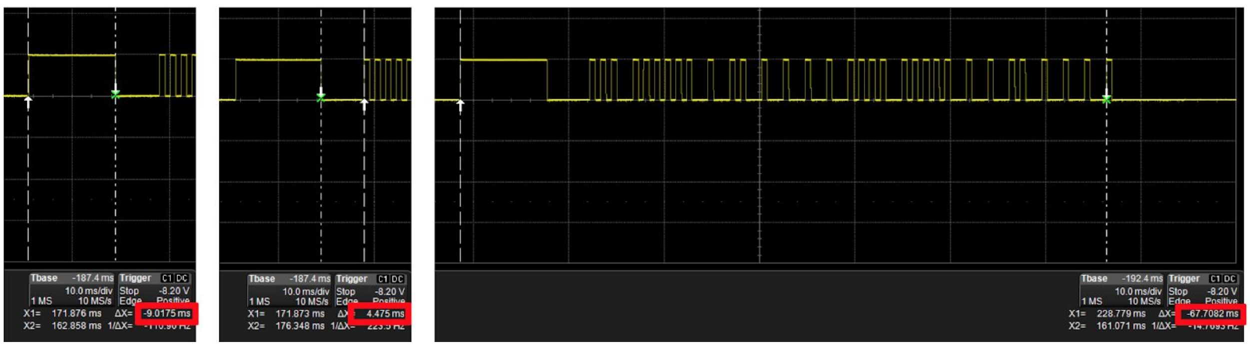
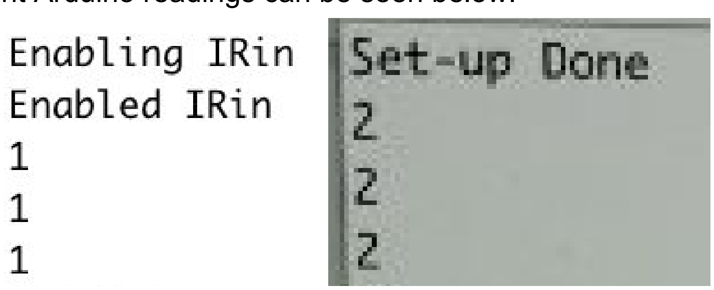


Figure 5: NEC Protocol, Oscilloscope Results

This proved that we could send and receive NEC codes at the standard 38kHz carrier frequency, matching the required leading pulse of 9ms, a space of 4.5ms, and a total transmission time of 67.5ms with a termination bit as defined in the NEC protocol.

To confirm we can send the unique pulses assigned to each player, the IR receiver input was monitored using the Arduino IDE serial monitor and the IR LED was pulsed with two unique codes. In this test, and in final implementation, player 1 pulses a binary 1 and player 2 pulses a binary 2. The resultant Arduino readings can be seen below.



(a) (b)

Figure 6: (a) Player 1 NEC Reading (b) Player 2 NEC Reading

This confirms that the IR LEDs can pulse unique codes and the IR receiver, combined with the microcontroller, can decode them for use in Arduino code. To reaffirm this, the output from the IR receiver was connected to an oscilloscope and the output waveform recorded as seen in Figure 7 where the x-axis is time .

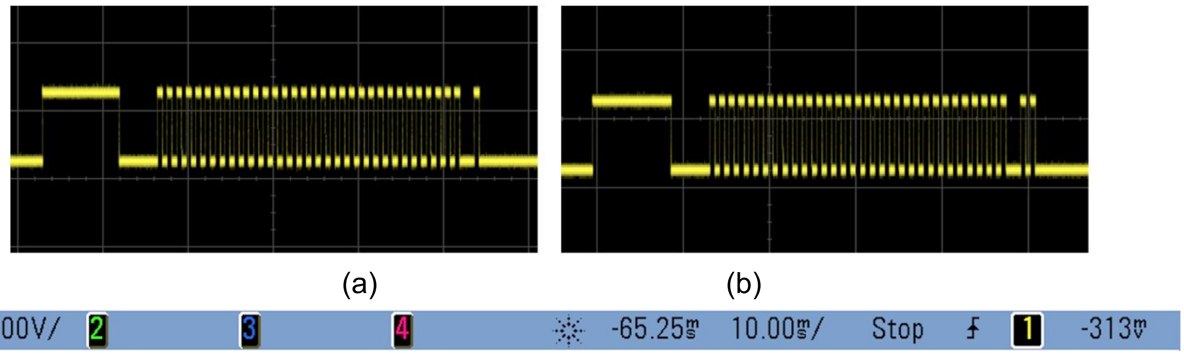


Figure 7: (a) Player 1 Oscilloscope Reading (b) Player 2 Oscilloscope Reading (c) Oscilloscope Scale

To test the field of view of the receivers, the IR receiver was connected to the Arduino and the IR inputs read using the serial monitor. Here, the IR emitter was pulsed from one meter away as we walked a perimeter around the receiver. Continual readings were registered from all angles.

2.1.4 Contact Sensors

The contact sensors ensure that our laser tag gloves only works when the hand is in the "finger gun" position. Our design would require the pinky, ring, and middle fingers to be simultaneously pressed in order to function. Conductive thread behaved like switches since they were sewn with a gap in between them. Each of the three fingers had the gapped conductive thread sewn onto them. The conductive fabric was then sewn onto the glove's palm. When the fingers make contact with the conductive fabric, the switches would close. The conductive thread and fabric implementation created the contact sensor circuit found in Figure 8.

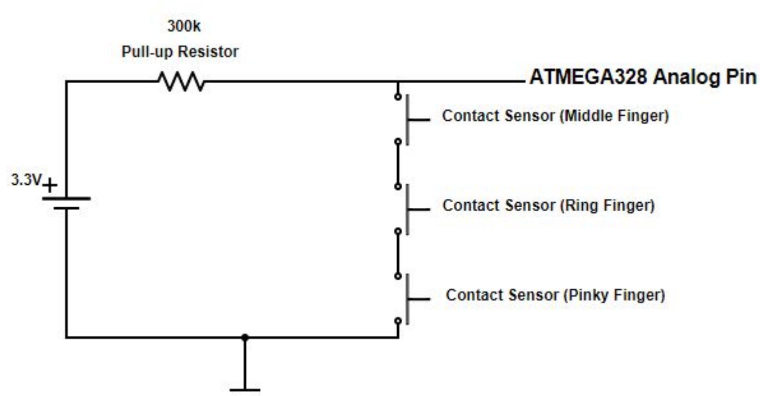


Figure 8: Contact Sensor Circuit [3]

The circuit took the 3.3V from the power module and it is connected to our ATMEGA microcontroller. Therefore, whenever all three fingers make contact the microcontroller was grounded and received a logical low. The glove “trigger” circuit works the same way,but it only involves conductive thread on the thumb and conductive fabric alongside the pointer finger. The microcontroller’s analog comparator ensured that the laser only fired when the pin in Figure 8 was low and the trigger was pressed. The 300kΩ pull-up resistor is used to meet our requriement that the contact sensor circuit draws less than 9mA in order to stay withing the microcontoller’s current rating whenever the contact sensors are not pressed. We probed the current with an ammeter and measured around .01 mA which easily meets our requirement.

We also had to ensure that our conductive fabric would conduct enough current to send the logical low to the microcontroller. The fabric ranges from 10000 to 10000000Ω, but it was closer to the 10000Ω value since the gaps created from the conductive thread were very small. When testing this we connected conductive thread and conductive fabric and used the continuation setting on our digital multimeter. The multimeter confirmed that enough current was able to pass through the fabric for a connection to be made. When the conductive fabric is grounded the voltage is close enough to zero for the microcontroller to read a logical low. The contact circuit then needed to be tested with the red laser to ensure only the “finger gun” placement would activate it. This was tested by placing various finger combinations on the conductive fabric. For example, if only the middle finger is pressed against the fabric, the red laser should not turn on. The glove passed our requirement since only the three fingers pressed would turn on the red laser. The final design of the glove with the laser, fabric, and thread can be seen below in Figure 9.

Figure 9: Contact Sensor Circuit End Result

2.2 Control System

2.2.1 Microcontroller

Being the brain of the project, this module is responsible for: powering the LEDs when necessary, writing various displays to the LCD screen, powering the IR emitter and buzzer, reading contact sensor inputs, and controlling the wireless transceiver.

As the project progressed, it was decided upon to use two ATMEGA328P microprocessors per player rather than one. This was due to the fact that we needed additional PWM pins, the inability to pulse the IR LED while the IR receiver reads (limitation on simultaneous function calls), and simpler code development. One ATMEGA is now on the wrist gauntlet, the other is on the vest. To transmit info from the vest to gauntlet microcontroller, the two use Inter-Integrated Circuit (I2C) protocol with the microcontroller on the vest acting as the master while the one on the gauntlet functions as the slave. Using I2C [4], the serial clock pins (SCL) and serial data pins (SDA) are used to send and receive data between the microcontrollers. The SCL is used to keep timing while the SDA functions as a path to send or receive a single bit of data at a time. The wrist gauntlet microcontroller controls IR pulsing, buzzer sounds, printing to the LCD, and reading contact sensor states. The vest microcontroller connects with the wireless XBEE transceiver to permit Radio Frequency (RF) transmission with the central hub (explained further in Section 2.2.2). The vest microcontroller controls the LED states, internal life count (Last Man Standing), score count (Deathmatch), and connection to the XBEE. The vest microcontroller sends data via I2C, such as time remaining in the game and remaining lives, to the gauntlet to print this information on the screen. As noted, we designed two main game modes that the vest microcontroller is responsible for: Deathmatch (who has the highest points in a limited amount of time) and Last Man Standing (who is the last surviving player). The finite state machines of these two modes can be seen in Figures 12 and 13 of the Appendix respectively.

As the ATMEGA can pulse the IR LED as verified in Section 2.1.1 and 2.1.3, and the IR LED only pulses when the contact sensors are closed as verified in Section 2.1.4, this confirms our first requirement for the microcontroller.

When testing the piezo buzzer with the microcontroller, we used a 500 Hz signal to generate the best blaster sound. With this frequency, our decibel test using a phone placed a meter away resulted in a max sound reading of 44 dB, 6 dB short of our proposed 50 dB as seen in Figure 10 below with blaster shots indicated by arrows.

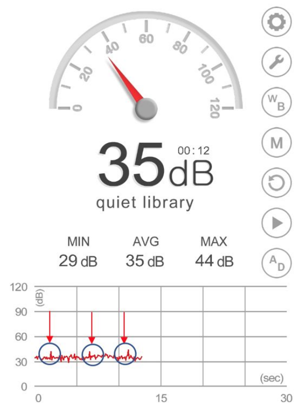


Figure 10: Buzzer Sound Results

Through incremental tests, game functionality was added to the microcontroller. These incremental tests included connecting the gauntlet and vest via I2C and shooting oneself to confirm lives decremented and reflected back on the gauntlet LCD screen. From here, two gauntlet-vest pairs were assembled, communicating via I2C. To test life count, player 1 shot player 2 and we confirmed that the vest LEDs flashed yellow and a decrement in lives was displayed on the screen. During the player's cool-down period, we also confirmed that player 2 cannot shoot. Player 1 shot player 2 until they ran out of lives, the LEDs stayed red, and that player 2 can no longer shoot or get shot. Next, player 2 shot player 1 and the process was repeated until player 1 ran out of lives.

Once XBEE communication was established between the players and the central hub, we tested the ability of the vest microcontroller to keep track of lives in Last Man Standing as well as score and time remaining in Deathmatch. The starting number of lives in Last Man Standing was set using the central hub and we confirmed the vest microcontroller received this number. This game mode was played through, the LED states were confirmed to work, and the winner was correctly relayed back to the central hub. Deathmatch was next chosen and the total time was set. The game was started and each player shot one another until the time ran out. The winning player was correctly relayed to the central hub. Here the LED states were also confirmed to work. In each game mode, the ability to shoot/get shot when in cool down was confirmed to not be possible as designed.

2.2.2 Wireless Transceiver

The wireless transceiver in the control system is mainly used to facilitate communication of game information between the central hub and the player systems. This includes the choice of game modes (Deathmatch or Last Man Standing), vital game parameters (e.g. time left in the game), and signals to start or stop the game. The transceivers that were being used were Series 1 XBEE modules that only required 3.3V to operate.

To verify that the transceiver is able to send and receive signals, a setup that consists of two arduino boards being hooked up to different serial monitors was being made. In one of the arduinos, a button was hooked up to it via a breadboard. When the button was pressed, the transceiver sent a signal that requested the other transceiver to send over a message. Right after sending the message, the transceiver was automatically configured to be in active listening mode to wait for the reply. The reply was then shown on the serial monitor. This success of this experiment proved that the transceiver could be configured to send and receive signals when required.

To prove that it works for our project, two players were asked to play the game in Deathmatch mode. In the beginning, the selected gamemode was relayed correctly to the gauntlet LCD screen of the two players. In addition, the players were consistently receiving from the central hub a common time countdown that was reflected accurately on their LCD screen. The hub also requested at the end the scores from each of the players and displayed the winner correctly with the amount of shots each player has fired. By having two players play out a full gamemode successfully, the wireless transceiver has demonstrated to be able to receive and send signals as configured in our project.

2.3 LCD Module

For the LCD Module on the hub, players are presented an interactive module where they could navigate a programmed game menu via buttons. It has 2 different game modes that players can choose from. In addition, players could adjust parameters such as the starting amount of lives, number of players in the game, and the duration of the game. Once the mode and game parameters were set, players could choose to start the game.

To verify that the LCD screen could be viewed comfortably, it was taken to two rooms with different lighting - one was well lit, the other was dark. The screen was then turned on and the menu system could be read comfortably without straining the eyes. The menu system was also tested to be functional as players were able to navigate to the game mode they wanted to play and set the necessary parameters that were correctly reflected on the screen.

As for the LCD Module on the gauntlet of a player, it displayed vital game information such as the amount of lives left or the amount of time remaining before the game ends. Also, it provided instructions to the players to press a button in order to join a game.

2.4 Game Module

The game module acts as the unit to provide visible and audible feedback to the player. This includes a piezo buzzer that buzzes for a quick moment whenever the user shoots a beam (triggered by the contact sensors) or dies (triggered by the microcontroller). It also has LED lights attached to the vest to act as a visual cue to the other players denoting their status.

However, we did not meet the 50 decibel requirement as outlined in Section 2.2.1.

2.5 Power Supply

The power module will power the components in our design. The laser tag glove will be battery powered with regulation so that each component gets the necessary voltage and current to function. The power supply will be located near the vest with battery holders.

2.5.1 Battery

Our power supply was a 9V Alkaline battery in a battery holder for each glove/vest pair and the central hub. Once we had completely assembled a glove and vest pair we could measure the total current drawn from the battery. In order to test the worst-case battery life scenario each component was left on even though they would not necessarily be on when actually playing. For example, when testing the battery capacity the red laser module was kept on even though a player would periodically turn it on during actual use. The current measured with everything on was 162.3 mA. The Energizer batteries used had around a 400 mAh battery capacity when the current draw exceeded 100mA. With battery capacity and total current drawn we can then calculate total battery life. Our requirement was for one battery to last 100 minutes of game time before being replaced. Based on our results, Equation 1 is used in order to find worst-case battery life.

$$Estimated\ Hours = Battery\ Capacity * 0.7\ (mAh)/Load\ Current(mA) \tag{1}$$

The 0.7 multiplier in Equation 1 accounts for external factors that may affect the efficiency of a battery [5]. When entering our values into the equation the expected battery life is 1.72 hours, which equates to 103.51 minutes. This meets our 100 minute requirement and it takes into account the worst-case scenario. They typical case is closer to 100mA, which would put us well above the requirement.

2.5.2 Voltage Regulators

All electric components in our design required either 5V or 3.3V. Voltage regulators were necessary in order to step down our 9V battery source. In the early phases of the design an SOT-23 package converter was used. This package was challenging to test and debug since it was 2.90 mm x 1.60 mm. The output was an incorrect voltage and the problem likely stemmed from the challenge in soldering such a small component. We moved on to the TO-220 package since it was easier to work with and we could maintain our planned schedule. The LM317T-DG linear voltage regulators were purchased from Mouser [6]. After following [6], I created the schematic shown in Figure 11 by using the Eagle software.

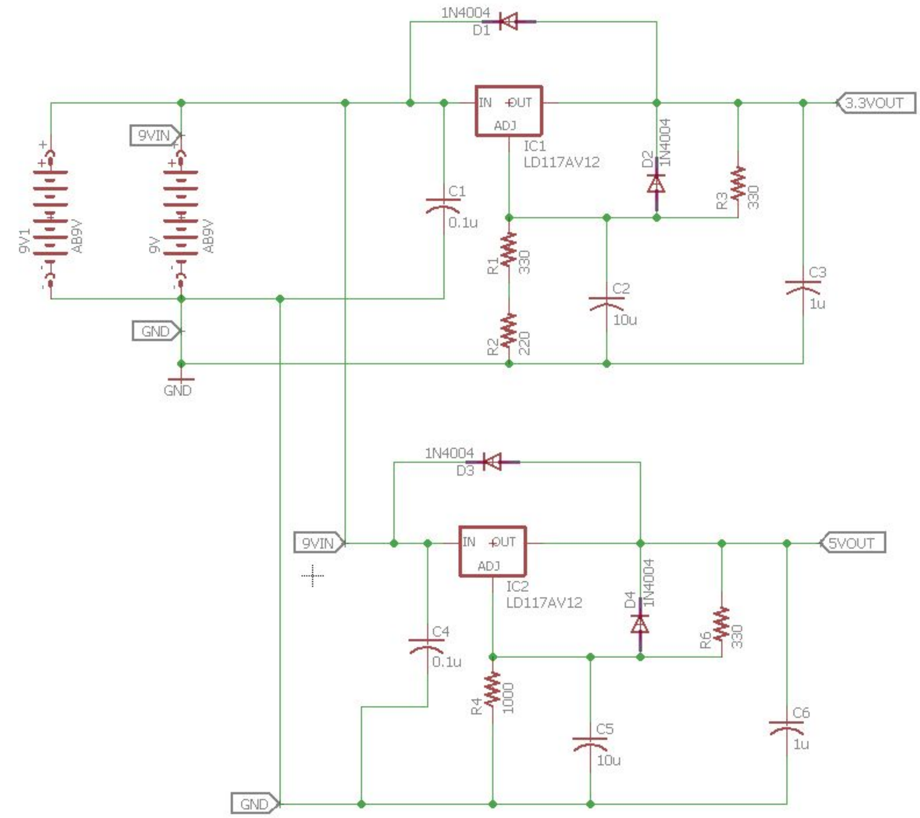


Figure 11: 9-5V/9-3.3V Voltage Regulator Schematic [6]

The design is the recommended protection schematic found in the STMicroelectronics datasheet [6]. The proper resistor values had to be calculated in order to get the proper output voltage based on our 9V input voltage. Two resistors (R1 and R2) are used in a voltage divider circuit in order to output a desired value. Equation 2 is the voltage divider equation that determines our output voltage. Vref is an internal property of the linear voltage regulators, which is 1.25V [6].

$$V out = V ref(1 + R2/R1) \tag{2}$$

R1 was selected to be 330Ω for both regulator circuits since it is a common value that is readily available in the lab. After computing the numbers with equation 2, R2= 1000Ω for the 5V output and R2=550Ω for the 3.3V output. The voltage regulator circuitry will ensure that a voltage ripple of ±5% is maintained. Capacitor values in Figure 11 were found by using the recommended values specified in the voltage regulator datasheet. The voltage regulators successfully output 5V and 3V within the ±5% requirement. When inputting 9V into the voltage regulators through the power supply or 9V batteries, they output 5.16V and 3.36V. .

3 Cost and Schedule

3.1 Cost

Table 1: Total Cost

Parts (Costs are for Two Players)				
Part Name	Supplier	Unit Cost	Quantity (For Two Players)	Total
Infrared Receivers	Mouser	$0.881	8	$7.05
IR Laser (1mW)	Ali Express	$12.00	2	$24.00
Red Laser (1mW)	Ali Express	$3.50	2	$7.00
Sensing Fabric	SparkFun	$24.95	1	$24.95
Conductive Thread	SparkFun	$3.95	1	$3.95
Work Glove	Amazon	$9.99	2	$19.98
			IR Module	**$86.93**
Microcontroller	Adafruit	$5.95	4	$23.80
XBEE Wireless Transceiver	Adafruit	$22.95	2	$45.90
			Control Module	**$69.70**
LCD Screen	Adafruit	$10.95	1	$10.95
Buttons (10 pack)	Adafruit	$2.50	1	$2.50
			LCD Module	**$13.45**
Piezo Buzzer	Adafruit	$1.50	2	$3.00
Tri-color LED (10 pack)	Adafruit	$9.95	1	$9.95
			Game Module	**$12.95**
9V Alkaline Battery (2 pack)	Walgreens	$6.00	1	$6.00
Converter (9V-5V)	Mouser	$1.30	2	$2.60
Converter (9V-3.3V)	Mouser	$1.30	2	$2.60
			Power Module	**$11.20**
			Part Total	**$194.23**

Labor				
Team Member	Hourly Rate	Total Hours	Multiplier	Cost
Alexander Korfel	$32	150	2.5	$12,000
Carlos Lara	$32	150	2.5	$12,000
Keng Yan Lim	$32	150	2.5	$12,000
			Labor Total	**$36,000**

Grand Total = Labor + Parts	**$36,194.23**

4 Conclusions

4.1 Accomplishments and Uncertainties

Overall, our project worked as desired during the final presentation. All of our high-level requirements were met. Our maximum reliable range was 32 ft which met our required 20 ft. Our vest has a lot of unused space, but it was still small enough to carry around in a backpack. Since we have to create each glove and vest pair, we only tested the product with two players. However, the code our central hub uses allows for easy integration with multiple players. The central hub asks for the number of players before starting a game. For example, if we expand the game to four players the central hub would wait for four “Player Ready” signals as opposed to two. The “Player Ready” signals would be sent with the circuitry in the vest so it is only a matter of owning a glove and vest to join the game. For our demonstration, both deathmatch and Last Man Standing game modes were showcased. Deathmatch correctly displayed our scores and determined the winner of the game. We demonstrated a match that was 4 minutes in length. Last Man Standing was showcased with two lives for each players. Once one player ran out of lives the game ended. These parameters of 4 minutes and two lives were chosen since they were short enough to demonstrate, but our final product allows the parameters to be changed as the players please. Obviously since the project was an overall success, most of our requirements were met. The only requirement we failed was making our buzzer ring at 50 decibels. This requirement did not end up being vital to the project since the buzzer was still audible enough to let the player know their shot registered. In terms of final functionality, we were able to create a playable and portable game of laser tag.

While we had successes, our final project also had some uncertainties. The biggest unsatisfactory result from our final product was that an Arduino was used for our XBEE communication rather than our printed circuit board. This issue ended up being unresolved since the debugging process was taking so long that it would halt the overall progress of the project. This issue was also strange since in essence the Arduino was making the same connections that we made in our PCB. At first we thought the potential error was due to a floating ground due to the inductance of the long wire connections. In order to remove the added inductance, we added 0.1μF capacitors in parallel. This method solved a previous issue with our microcontrollers, but it did not solve the XBEE communication. We also tried adjusting the voltage applied to the microcontroller to fix the issue. When powered, the Arduino Vcc pin was measured to be 4.98V. As mentioned before, our converter's output voltage was 5.16V. We did not think this would solve the issue since both values fall within the microcontroller's rated voltage. This did not end up fixing the issue, and we no longer knew of a timely method to fix the issue. The Arduino was then left in the final product since we wanted the product to work rather than fixing every small issue. Now that we have presented, we believe the error occurs due to our voltage divider circuit that steps down the microcontroller 5V into 3.3V for the XBEE connection. Slight differences with the microcontroller voltages might give the XBEE's slightly different values and cause errors in their communication. We also had a small issue with our LEDs that did not greatly impact the overall presentation. One of the vests LEDs were dimmer when compared to the other vest. When debugging we found that one of vest's four LEDs had a much higher resistance between two pins compared to the rest. For example, when operating as expected the ohmmeter read 330Ω between the two pins. The odd LED, however, was reading 700kΩ. This was most likely the reason for the vest being dimmer, but unfortunately we were not able to fix the issue. The issue would most likely have been solved by replacing the LED, but we did not purchase spares and did not have time to wait for more. Our project had successes and failures, but the overall functionality worked as intended.

4.2 Future Work

After completing the project, we have many ideas to improve the project if we were to continue working on them. Obviously we would further debug and fix the issues under the "Accomplishments and Uncertainties" section. We would also want to make player syncing a smoother experience when starting a new game. At the moment, when a game ends each player has to turn the glove and vest off then on again. We would change our code to display a "Play Again?" prompt at the end of each game. More game modes would also be added. For example, we could implement a game mode to see who could draw the gun and shoot the opponent faster. Also we would want to increase the vest's portability by decreasing the vest size. At the moment the vest is very large since it was a cheap option that let us save on costs. Lastly, we would sew the glove more carefully since shorts occured at times due to the conductive thread fraying. Potential design alternatives to the glove include the use of conductive paint rather than thread. This would prevent the shorting issue and remove the potential current-to-skin contact on the glove. Our product worked very closely to what we envisioned, so future work would polish the idea further.

4.3 Ethical Considerations

Many ethical and safety concerns were taken into consideration when creating our project. An obvious issue that comes to mind is that the project revolves around lasers. When developing the project, we were mindful of the lasers since they can cause damage when pointed directly at one's eyes [7]. Our project is not meant to cause harm to anyone, and it is only meant to be an enjoyable experience for those who use it. However, intentional misuse of the product could lead to severe eye damage such as blindness. This can occur if the laser is pointed directly at people, vehicles, or any living thing with their eyes exposed. In order "to hold paramount the safety, health, and welfare of the public," we must design the project in a way which minimizes any safety hazards [8]. Since the laser portion of the project is a key component, we could not simply make it safer or remove the hazards. We minimized accidents where players point at each other's eyes with our sensor location. For example, by placing the sensors below the shoulders, on the stomach, and on the back we make sure that one does not need to aim towards the face which limits exposure to the eyes. Another factor we considered was the power of our laser module. Lasers can cause eye damage even at low wattage, such as 5mW. A regular 60W light bulb uses more power, but the light is spread around making it much less dangerous than the concentrated laser. Due to this safety concern, we used a 1mW red laser module. At one milliwatt, the laser will need to hit the eye for seconds in order to cause permanent damage. The laser was classified as Class II, which is relatively safe when used as instructed. For the IR portion, an IR LED was used, as opposed to a laser module, so it was not powerful enough for us to worry about safety. Also, our product is intended for people of varying ages and background. Therefore, many of them might not know of the dangers associated with lasers and IR light. The final product would come with safety warnings and information about lasers in order to notify them about proper usage of the product.

Another safety issue associated with our product is the contact with the human body and electrical equipment. Our laser tag glove is meant to be worn and it will contain electrical components within. We made sure that all of the circuitry is properly closed off in order to prevent current to travel through the person's body. Our soldering and connecting of the circuit was reliable since everything stayed in place regardless of each player's movements. The conductive thread was sown into the glove, so a rubber glove was worn under the laser tag glove as an insulator. Lastly, since our product will be battery powered we must make sure each component is powered at its appropriate voltage and current ratings. This is necessary since we would not want components to blow up or heat up to dangerous levels. Our product used relatively low current, however, so this was not a large concern in the end. Overall, our project comes with safety hazards that must be considered. The hazards are unavoidable due to the need for electricity and lasers, but we took the proper considerations and made our end product as safe as possible.

5 References

[1] Bergmans, S. (2018). SB-Projects - IR - NEC Protocol. [online] Sbprojects.net. Available at: https://www.sbprojects.net/knowledge/ir/nec.php [Accessed: 20-Feb-2018].

[2] YouTube. (2018). EEVblog #506 - IR Remote Control Arduino Protocol Tutorial. [online] Available at: https://www.youtube.com/watch?v=BUvFGTxZBG8&index=2&list=LLL8M-vN0wzoV1t_wvdf6ONQ&t=388s [Accessed 13 Feb. 2018].

[3] Wong, Timothy, et al. "American Sign Language Alphabet Interpreter." ECE 445 Web Board, 2017. [Online] Available: https://courses.engr.illinois.edu/ece445/getfile.asp?id=9185 [Accessed 13-Feb-2018]

[4] Arduino - MasterWriter, www.arduino.cc/en/Tutorial/MasterWriter. [Online] Available: https://www.arduino.cc/en/Tutorial/MasterWriter [Accessed: 22-Feb-2018]

[5] "Battery Life Calculator." Battery Life Calculator | DigiKey Electronics, www.digikey.com/en/resources/conversion-calculators/conversion-calculator-battery-life. [Accessed 20-Feb-2018]

[6] "LM217, LM317," STMicroelectronics. [Online]. Available: https://www.mouser.com/ds/2/389/CD00000455-249730.pdf . [Accessed: 09-Mar-2018]

[7] "Can a pocket laser damage the eye?" Scientific American, Available: www.scientificamerican.com/article/can-a-pocket-laser-damage/ [Accessed: 03-Feb-2018]

[8] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html [Accessed: 28- Jan- 2018].

6 Appendix A: Requirement and Verification Tables

Table 2: Requirements and Verifications (RV) for IR LED

Requirements:	**Verifications:**	**Status (Y/N):**
IR emitter must transmit at the carrier frequency of 38kHz.	1) Connect IR receiver OUT pin (pin 1) to the oscilloscope. 2) Aim the IR laser towards the IR receiver. 3) Light the IR LED using 38kHz programmed within the microcontroller by firing a shot. 4) Monitor the transmitted pulse using the oscilloscope and zoom in until the frequency of IR pulses can be determined. 5) Verify this is 38kHz.	Y
The IR laser must be able to pulse code according to NEC protocol.	1) Connect IR receiver OUT pin (pin 1) to the oscilloscope. 2) Aim the IR laser towards the IR receiver. 3) Light the IR LED using 38kHz carrier frequency sending pulses following NEC protocol. 4) Monitor the transmitted pulse using the oscilloscope and measure pulses. 5) Verify the pulses sent are to NEC protocol by measuring the time HIGH and LOW for binary 1 and 0.	Y
IR laser must have a range of at least 20 feet.	1) Connect IR receiver OUT pin (pin 1) to the oscilloscope. 2) Place IR laser 20 feet away from receiver. 3) Shoot the vest. 4) Monitor the raw output on the oscilloscope and confirm it matches transmitted pulse.	Y

Table 3: RV for IR Sensors

Requirements:	Verifications:	Status (Y/N):
IR sensor must have a field of view of at least 45 degrees from a meter away.	1) Setup operating IR receiver facing upwards with the OUT pin (pin 1) monitored using oscilloscope. 2) Use IR diode to manually sweep across the receiver dome and view raw data response on the oscilloscope. 3) Make sure data reads HIGH between ± 22.5 degrees.	Y
IR receiver must distinguish different pulsed signals sent at the 38kHz carrier frequency.	1) Setup operating IR receiver facing horizontally from an IR diode. 2) Apply 5V to +5V pin and monitor the OUT pin (pin 1) using an oscilloscope. 3) Pulse the IR LED according to NEC protocol at 38kHz and verify the receiver output matches the inputted pulse using the oscilloscope. 4) Cross check by connecting pin 1 to the Arduino and monitor the decoded pulses using the serial monitor.	Y

Table 4: RV for Contact Sensors

Requirements:	Verifications:	Status (Y/N):
The microcontroller reads a logic high whenever the thumb makes contact with the pointer finger and the remaining three fingers are making contact with the palm.	1) Probe pin connecting contact sensor and microcontroller. 2) Make contact between fingers/palm and thumb/pointer finger to mimic “finger gun” shape. 3) Verify that the microcontroller is reading a logic high.	Y
The microcontroller reads a logic low if the palm or the non-pointer fingers are not making contact with pointer finger and palm, respectively.	1) Probe pin connecting contact sensor and microcontroller. 2) Verify that the microcontroller is reading a logic low for the scenarios in steps 3-5 3) Test with an open palm. 4) Test with thumb/pointer finger contact, but no fingers/palm contact. 5) Test with fingers/palm contact, but no	Y

	thumb/pointer finger contact.	
The contact sensors maintain a max current flow of 9mA.	1) Probe the connection between the contact sensor and digital microcontroller pin with an ammeter. 2) Make contact with thumb/pointer finger and fingers/palm to give microcontroller a high signal. 3) Record current reading and make sure it is below 9mA.	Y

Table 5: RV for Microcontroller

Requirements:	**Verifications:**	**Status (Y/N):**
Must be able to pulse the laser module at 38 kHz when contact sensor sends high signal.	1) Produce a high signal with the contact sensor by making contact with the pointer finger/thumb and fingers/palm	Y
Powers the Piezo Buzzer at 4 kHz when contact sensor sends high signal to ensure at least 50 decibels.	1) Produce a high signal with the contact sensor by making contact with the pointer finger/thumb and fingers/palm 2) Use a mobile app that can read decibels in order to measure Piezo Buzzer noise 3) Verify that the buzzer produced a noise at 50 decibels minimum	N
Keeps track of game information such as player lives, score, and round time.	Round Time and Score: 1) Switch Game Mode to ‘Deathmatch’ 2) Player 1 to fire a certain number of shots at Player 2 3) Player 2 fires a certain number of shots at Player 1 4) Wait till specified Time period is over 5) Both players will try to fire shots again (but must not be able to) 6) Check the scores on the central hub’s LCD screen Player lives: 1) Switch Game Mode to ‘Last Man Standing’ 2) Player 1 to fire shots at Player 2 until Player 2’s LED lights turn red	Y

	(indicates death) 3) Player 2 will try to fire some shots at Player 1 but must be unable to shoot	

Table 6: RV for Wireless Transceiver

Requirements:	**Verifications:**	**Status (Y/N):**
Transmission range of at least 20 feet indoors even with physical obstacles such as walls.	1) Switch Game Mode to ‘Deathmatch’. 2) Players 1 and 2 to stand 25 feet apart. 3) Player 2 will fire 3 shots at Player 1 4) Player 1 will fire 1 shot at Player 2 5) Wait for the game timer to run out 6) The wireless hub should display the scores of player one and two as 1 and 3 respectively	Y

Table 7: RV for LCD Module

Requirements:	**Verifications:**	**Status (Y/N):**
Screen must be backlit and comfortably viewed in both a dark and well lit room (developed for indoor use). Brightness must be adjustable.	1) Step in a dark room with LCD screen. 2) Place screen 1 ft away, as if telling the time on a watch. 3) Make sure text on screen is visible without strain or squinting. 4) If not, adjust brightness up or down to a point where strain is alleviated. 5) Repeat steps b) through d).	Y
LCD acts as an interactive menu for players to set game settings.	1) Click each of the four buttons. 2) Confirm each menu/sub-menu can be accessed and/or changed. 3) This update in game modes must reflect in the Arduino code.	Y
Screen response time of less than 300ms.	1) Power on display and wait in main menu. 2) Use slow-motion camera on phone. 3) Traverse through all menus and film. 4) Convert from measured frames to total seconds using the camera’s fps rate. 5) Verify time is less than 300ms.	Y

Reflect a decrease in lives when a player is shot or an increase in score when a player shoots another. Based on the game mode.	1) Grab a set of gloves. 2) Shoot each other's vests. 3) Confirm that when you're shot, the total lives on the screen decreases when in Last Man Standing mode. 4) Confirm that when you shoot someone, your score increases and that is reflected in the hub when the game ends 5) Repeat for all game modes and players.	Y

Table 8: RV for Game Module

Requirements:	**Verifications:**	**Status (Y/N):**
All four LEDs, per vest, must alternate colors based on the current state of the player. Red: Player ran out of lives and is out. Green: Player is in the game, has lives remaining. Yellow: Player is paralyzed, can't shoot and can't be shot.	1) The four LEDs will be assembled according to Figure 15 and 16. 2) Run through the three states using the Arduino. 3) Visually monitor the LED output. 4) When IR pulse communication is achieved, point the IR laser at the receiver 1 ft away. 5) Pulse IR laser. 6) When hit, visually monitor the LEDs and make sure the change from Green to Yellow and back. 7) Fire IR laser until player is out of lives. 8) When player is out, inspect that the LEDs switch to red.	Y
LEDs must be bright enough to be viewed from a meter away in a lit room (developed for indoor use).	1) Stand a meter away from the source of the LEDs. 2) Verify LEDs are easily visible.	Y
Buzzer must be able to output at least 50 db from one meter away.	1) Place decibel meter, or similar audio measuring device, a meter from the glove. 2) Shoot gun. 3) Measure intensity and verify it is at least 50 db.	N

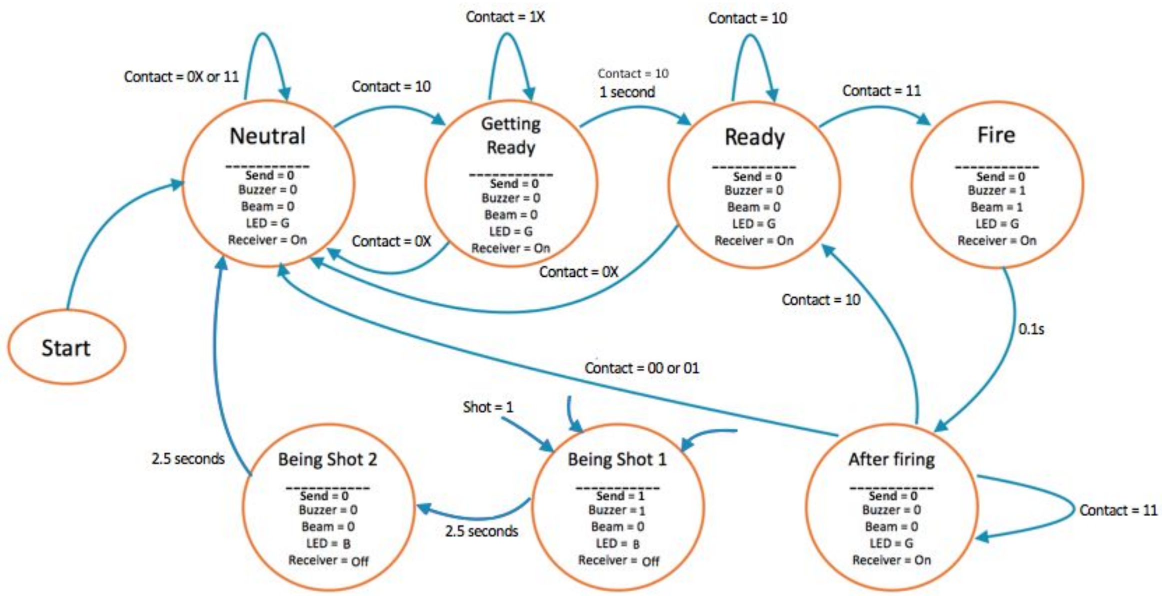
Table 9: RV for Battery

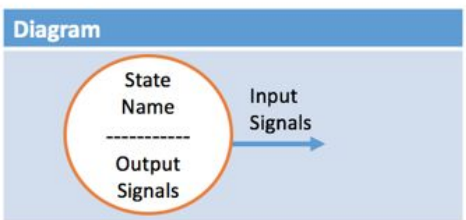
Requirements:	**Verifications:**	**Status (Y/N):**
A 9V 500mAh Battery will provide at least 100 minutes of play.	1) Power the laser tag glove with 9V and probe the set-up with ammeter 2) Record current after one 10 minute session 3) Turn off power for a few seconds and repeat step 2 for 6 sessions 4) Total current drawn in the six sessions 5) Verify the average current from the six session is at maximum 250mA	Y

Table 10: RV for converter

Requirements:	**Verifications:**	**Status (Y/N):**
9.0V to 5.0V DC-DC Buck converter must supply 5.0± 5%V with a maximum output current of 3A.	1) Probe the 9-5V Vout Pin with a voltmeter when the circuit is supplied by 9V DC. 2) Ensure that the voltmeter reads the desired 5V±5%V. 3) Probe the 9-5V Vout Pin with an ammeter when the circuit is supplied by the 9V DC. 4) Ensure the ammeter is below the converter's 3A rating.	Y
9.0V to 3.3V DC-DC Buck converter must supply 3.3± 5%V with a maximum output current of 3A.	1) Repeat steps 1-4 above for the 9-3.3V Vout Pin.	Y

7 Appendix B: Game Logic Finite State Machines and Schematics





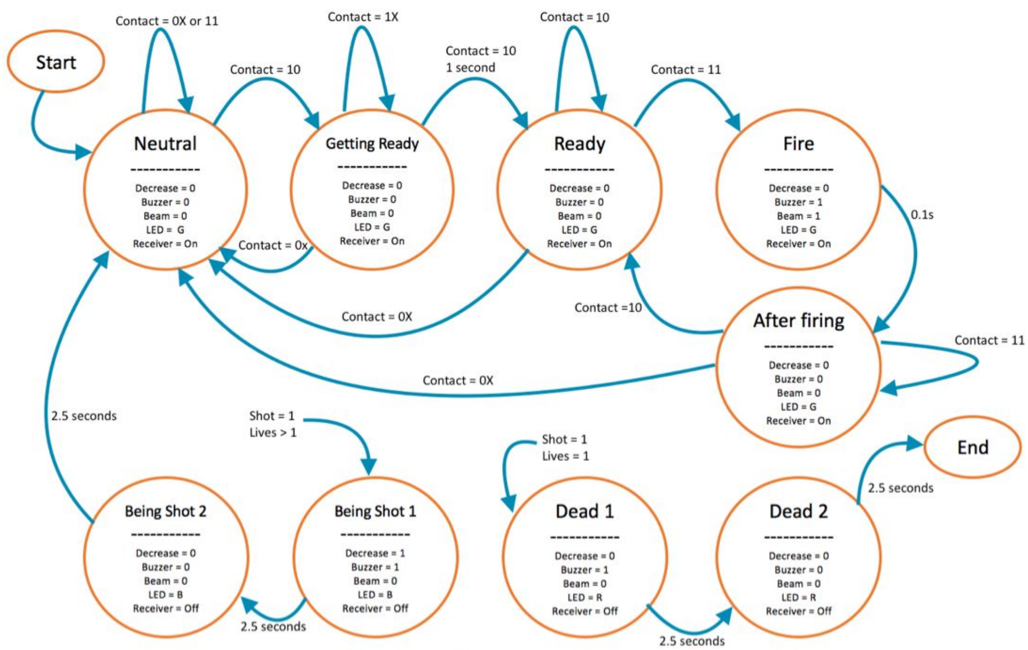
Signals

Signal	Values	Meaning
Buzzer	0	Buzzer is off
	1	Buzzer sounds
Beam	0	Beam is turned off
	1	Fire the IR Beam
LED	G	Neutral
	B	'Untouchable' after being shot
Receiver	0	Turn off receiver
	1	Turn on receiver
Send	0	No sending of id code
	1	Send shooter's id to master
Contact 1 and 2	0	Contact sensors are not touching
	1	Contact sensors are touching
Shot	0	IR Receiver did not receive signal
	1	IR receiver received signal (shot)

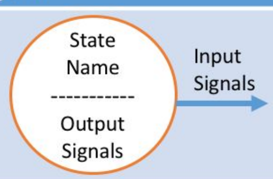
State Names

State	Picture	Remarks
Neutral		Player is open to shots Player is not able to fire any shots
Getting Ready		The transitional state that lasts a period of 1 second as long as the player does not let go of Contact 1 Sensors
Ready		Player is in static state where he can fire a shot just by triggering the Contact 2 sensors and not letting go of Contact 1 sensors
Fire		The brief transitional state that lasts a period of 0.1s. The IR beam will be fired and the buzzer will sound.
After firing	None	A static state to prevent the player from firing another shot until he lets go of any of the contact sensors
Being Shot 1	None	The transitional state that happens when the shot signal is active. The player can't be shot again during this state. The LED's light is blue.
Being Shot 2	None	The transitional state to turn off the buzzer.

Figure 12: FSM for Deathmatch



Diagram



Signals

Signal	Values	Meaning
Buzzer	0	Buzzer is off
	1	Buzzer sounds
Beam	0	Beam is turned off
	1	Fire the IR Beam
LED	G	Neutral
	B	'Untouchable' after being shot
Receiver	0	Turn off receiver
	1	Turn on receiver
Decrease	0	Do not subtract 1 from player HP
	1	Subtract 1 from player HP
Contact 1 and 2	0	Contact sensors are not touching
	1	Contact sensors are touching
Shot	0	IR Receiver did not receive signal
	1	IR receiver received signal (shot)
Lives	>= 1	Amount of lives left by the player

State Names

State	Picture	Remarks
Neutral		Player is open to shots Player is not able to fire any shots
Getting Ready		The transitional state that lasts a period of 1 second as long as the player does not let go of Contact 1 Sensors
Ready		Player is in static state where he can fire a shot just by triggering the Contact 2 sensors and not letting go of Contact 1 sensors
Fire		The brief transitional state that lasts a period of 0.1s. The IR beam will be fired and the buzzer will sound.
After firing	None	A static state to prevent the player from firing another shot until he lets go of any of the contact sensors
Being Shot 1	None	The transitional state that happens when the shot signal is active. The player can't be shot again during this state. The LED's light is blue.
Being Shot 2	None	The transitional state to turn off the buzzer.
Dead 1	None	Similar to being "Being Shot 1" with the exception that the LED will be lighted red instead to indicate that the player is out of lives.
Dead 2	None	The transitional state to turn off the buzzer

Figure 13: FSM for Last Man Standing

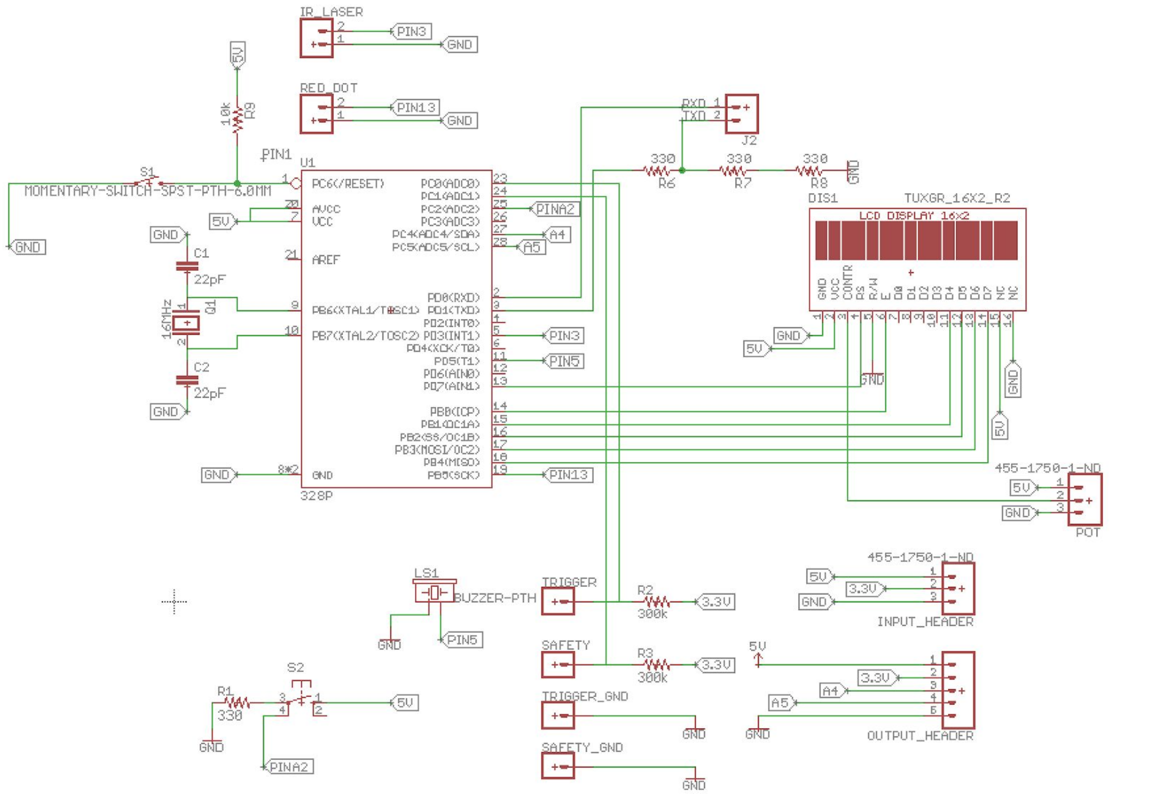


Figure 14: Glove Schematic

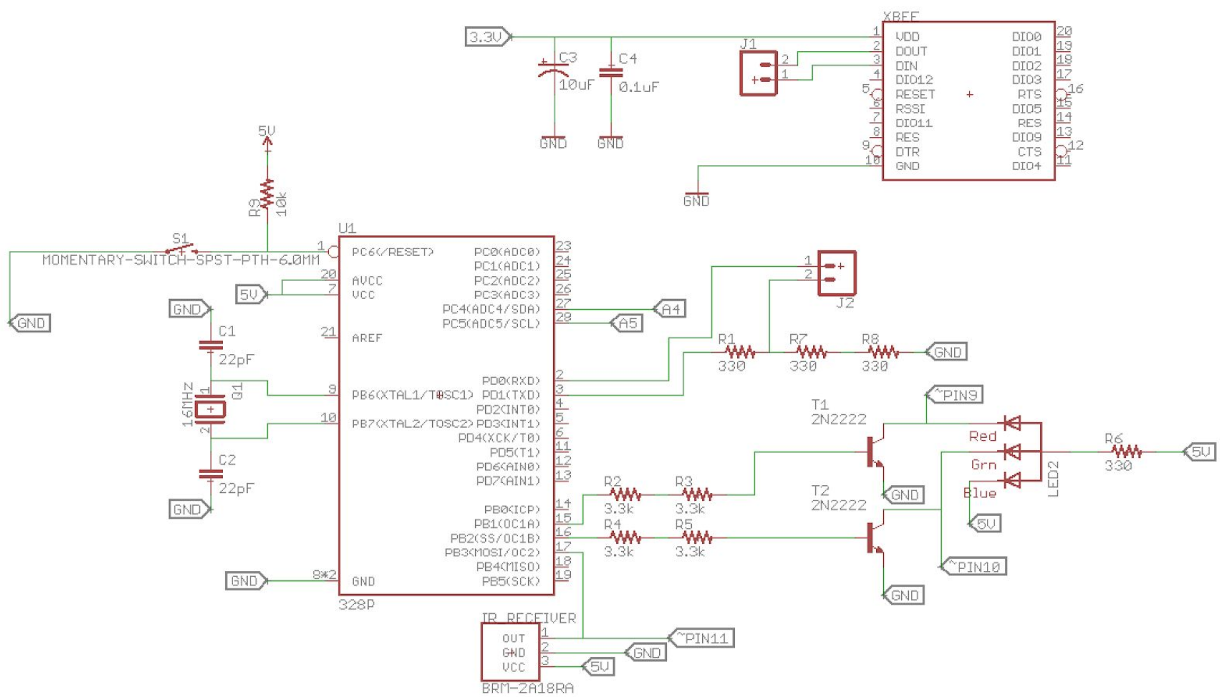


Figure 15: Vest Schematic

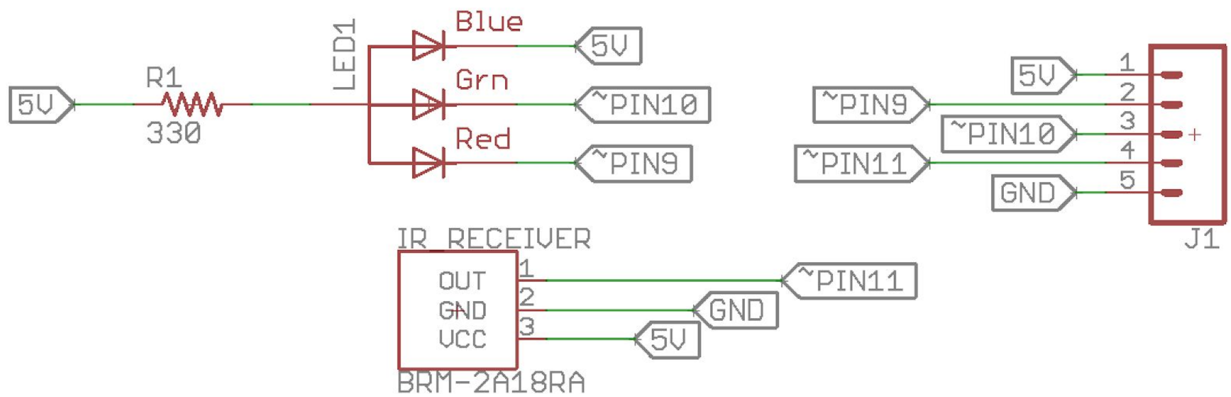


FIgure 16: Shoulder Pieces Schematic