Team 11: Lazer-Ops Final Design Report
Calvin College Engineering 340
May 15, 2014
Matthew Block
Eric Bouwkamp
Kristen Herder
Mark Willard
Executive Summary

Team Lazer-Ops examined the feasibility and took on the task of designing and prototyping a home laser tag system. Current laser tag systems involve travel time, cost, and game limitations. Lazer-Ops wants to bring a refreshing twist to traditional laser tag allowing for a differentiated product focused on convenience, portability, game mutability. The system targets a wide market of competitors, gamers, and fun-lovers ranging from elementary aged children to young adults. The system will operate wirelessly through a Wi-Fi network. The vests and laser gun relay data to the game’s base station which performs all the data processing and analysis. From there, it is transmitted to a mobile phone application which relays game statistics in real-time to each player and presents them with alternative game play options. The major risks of the project include sub-system cohesion and communication. Through preliminary research and rudimentary system design, Team Lazer-Ops has estimated that it will cost $340 to build a preliminary system consisting of two laser, two vests, a central hub, and mobile application. This system will be able to operate within a 30-40m radius of the central hub. It will also be able to send infrared signals upwards of 30 feet between players. These deliverables will be completed by May 2014. In the event that the system goes into production, cost estimates and market research have led to a potential retail value of $70 for a home laser tag starter kit consisting of 2 laser guns and 2 vests in addition to the hub and mobile application. Expansion packs of 2 additional guns and vests will be sold for $40.
# Table of Contents

Executive Summary ............................................................................................................................................... ii  
Table of Contents ........................................................................................................................................... iii  
Table of Figures ............................................................................................................................................. vii  
Table of Tables ............................................................................................................................................... vii  
Table of Acronyms ......................................................................................................................................... ix  

1 Project Overview .......................................................................................................................................... 1  
1.1 Calvin Senior Design ................................................................................................................................. 1  
1.2 Problem Statement ..................................................................................................................................... 1  
1.3 Project Overview ......................................................................................................................................... 1  
1.3.1 Project Functions ..................................................................................................................................... 2  
1.3.2 Differentiation ......................................................................................................................................... 2  
1.3.3 Target Consumers ................................................................................................................................. 4  
1.3.4 Project Goal ......................................................................................................................................... 4  
1.4 Team Organization ..................................................................................................................................... 5  
1.4.1 Team Member Biographies .................................................................................................................... 5  
1.4.2 Team Member Responsibilities ............................................................................................................. 7  
1.4.3 Design Norms ....................................................................................................................................... 8  
1.5 Schedule Management ............................................................................................................................... 9  
1.5.1 Work Breakdown Schedule ................................................................................................................ 9  
1.5.2 Cumulative Hours ................................................................................................................................ 9  
1.6 Document Structure ................................................................................................................................. 9  

2 Requirements .............................................................................................................................................. 10  
2.1 Technical Description ............................................................................................................................... 10  
2.1.1 Central Hub ......................................................................................................................................... 10  
2.1.2 Laser Gun .......................................................................................................................................... 11  
2.1.3 Laser Vest .......................................................................................................................................... 12  
2.1.4 Mobile Android Application ............................................................................................................... 13  
2.2 System Features ....................................................................................................................................... 13  
2.3 Customer & Market Requirements .......................................................................................................... 13  
2.3.1 Functional Requirement ..................................................................................................................... 14  
2.3.2 Weight Limit ...................................................................................................................................... 14
2.3.3 Type of Environment & Use .......................................................... 14
2.4 Mechanical Requirements .................................................................. 14
  2.4.1 Central Hub Enclosure ................................................................. 15
  2.4.2 Sensor Vest Enclosure ................................................................. 15
  2.4.3 Laser Gun Enclosure ................................................................. 15
2.5 User Interface Requirements .............................................................. 15
  2.5.1 Inputs ..................................................................................... 15
  2.5.2 Outputs .................................................................................. 16
  2.5.3 Heads-up Display & Mobile Android Application ......................... 16
2.6 Communication Requirements ............................................................ 16
2.7 Power Requirements ........................................................................... 16
2.8 Environmental Requirements .............................................................. 16
  2.8.1 Power Efficiency ..................................................................... 17
  2.8.2 RoHS Compliance .................................................................... 17
2.9 Deliverables ....................................................................................... 17
  2.9.1 PPFS ...................................................................................... 17
  2.9.2 In-Class Presentations ............................................................... 17
  2.9.3 Project Posters .......................................................................... 17
  2.9.4 Final Report ............................................................................ 17
  2.9.5 Final Presentation ...................................................................... 17
  2.9.6 Working Prototype .................................................................... 18
  2.9.7 Design Notebooks ..................................................................... 18
  2.9.8 Team Website ........................................................................... 18
  2.9.9 System Software ....................................................................... 18
  2.9.10 Drawings and Schematics ......................................................... 18
3 System Architecture ................................................................................ 19
  3.1 Overview ...................................................................................... 20
  3.2 Components .................................................................................. 20
    3.2.1 Central Control Hub ................................................................. 20
    3.2.2 Laser Gun .............................................................................. 20
    3.2.3 Sensor Vest ............................................................................ 21
    3.2.4 Mobile Android Application ................................................ 22
3.3 System Power Consumption .................................................................................. 23
3.4 Communication Overview .................................................................................. 23
  3.4.1 Laser Gun IR Transmitter to Sensor Vest ...................................................... 23
  3.4.2 Sensor Vest IR Receiver to Laser Gun ......................................................... 24
  3.4.3 Laser Gun to Central Hub ............................................................................. 24
  3.4.4 Central Hub to Mobile Application .............................................................. 24
4 Central Hub Design ......................................................................................... 24
  4.1 Requirements ................................................................................................. 24
  4.2 Mechanical Design ......................................................................................... 25
  4.3 Hardware Design ............................................................................................ 27
    4.3.1 Raspberry Pi Linux Computer ................................................................. 28
    4.3.2 Linksys Wireless Router ......................................................................... 29
    4.3.3 Power Supply Circuit Board ..................................................................... 30
    4.3.4 Display Circuit Board ............................................................................... 31
  4.4 Software Design .............................................................................................. 33
    4.4.1 Programming Language Choice ................................................................ 34
    4.4.2 Software Design Structure ...................................................................... 34
    4.4.3 Graphical User Interface .......................................................................... 35
    4.4.4 Receiving and Transmission of Information ............................................ 36
  4.5 Budget ............................................................................................................ 42
5 Laser Gun Design .............................................................................................. 42
  5.1 Electrical Hardware Design ............................................................................ 43
    5.1.1 Controller ................................................................................................. 44
    5.1.2 Reload Button .......................................................................................... 47
    5.1.3 Trigger ..................................................................................................... 48
    5.1.4 On/Off Rocker Switch ............................................................................... 48
    5.1.5 Speaker ..................................................................................................... 49
    5.1.6 Infrared LED ............................................................................................ 49
    5.1.7 Lens .......................................................................................................... 50
    5.1.8 Laser Sight ............................................................................................... 50
    5.1.9 WiFly ........................................................................................................ 53
  5.2 Software Design .............................................................................................. 57
5.2.1 Software Phases ................................................................. 57
5.2.2 Testing ................................................................. 59
5.2.3 Requirements ................................................................ 60
5.2.4 Budget ................................................................. 61
5.2.5 Production Model ........................................................... 61

6 Sensor Vest Design ................................................................. 62
6.1 Electrical Hardware Design ...................................................... 62
       6.1.1 Sensors ............................................................. 62
6.2 Budget ................................................................. 65

7 Mobile Android Application .................................................... 65
7.1 System Requirements ............................................................ 65
7.2 Software Design .............................................................. 65
       7.2.1 Functionality .......................................................... 66
       7.2.2 Application Flow .................................................... 68
       7.2.3 Design Decisions ..................................................... 70
6.2.4 Software Updates ........................................................... 72

8 System Prototype: Lazer-Ops 1.0 ................................................ 72
8.1 Purpose ................................................................. 72
8.2 System ................................................................. 73
8.3 Hardware ................................................................. 73
8.4 Software ................................................................. 73
8.5 Mechanical ............................................................. 73

9 Production Model ................................................................. 73
9.1 System ................................................................. 74
9.2 Central Hub ............................................................. 74
9.3 Laser Gun & Sensor Vest .................................................... 74
9.4 Mobile Android Application ................................................ 74

10 Conclusion ................................................................. 74
10.1 Possible Future Improvements .................................................. 75

11 Acknowledgments ............................................................. 75
12 References ................................................................. 76
### Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System Mock Diagram</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Laser Gun Inputs/Outputs</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>Laser Gun/Vest Requirements</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>Sensor Placement Diagram</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>System Diagram</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>Sensor Vest Mock Diagram</td>
<td>21</td>
</tr>
<tr>
<td>7</td>
<td>Android Statistics Activity</td>
<td>22</td>
</tr>
<tr>
<td>8</td>
<td>IR pattern sent by Laser Gun</td>
<td>23</td>
</tr>
<tr>
<td>9</td>
<td>Bandwidth Calculations</td>
<td>25</td>
</tr>
<tr>
<td>10</td>
<td>Electronics Mounting Board</td>
<td>26</td>
</tr>
<tr>
<td>11</td>
<td>Central Hub Front Panel</td>
<td>26</td>
</tr>
<tr>
<td>12</td>
<td>Central Hub Rear Panel</td>
<td>27</td>
</tr>
<tr>
<td>13</td>
<td>Central Hub Block Diagram</td>
<td>28</td>
</tr>
<tr>
<td>14</td>
<td>Power Supply Circuit Board Schematic</td>
<td>30</td>
</tr>
<tr>
<td>15</td>
<td>Power Supply Circuit Board Layout</td>
<td>30</td>
</tr>
<tr>
<td>16</td>
<td>Central Hub Power Board Calculations</td>
<td>31</td>
</tr>
<tr>
<td>17</td>
<td>Central Hub Display Board Schematic</td>
<td>32</td>
</tr>
<tr>
<td>18</td>
<td>Central Hub Display Board Layout</td>
<td>33</td>
</tr>
<tr>
<td>19</td>
<td>Central Hub Software Structure</td>
<td>34</td>
</tr>
<tr>
<td>20</td>
<td>Central Hub Windows Application</td>
<td>35</td>
</tr>
<tr>
<td>21</td>
<td>Block Diagram of Laser Gun</td>
<td>43</td>
</tr>
<tr>
<td>22</td>
<td>Laser Gun Schematic</td>
<td>46</td>
</tr>
<tr>
<td>23</td>
<td>Implementation of the Reload Button</td>
<td>47</td>
</tr>
<tr>
<td>24</td>
<td>Sample Push Button with Arduino Schematic</td>
<td>47</td>
</tr>
<tr>
<td>25</td>
<td>Trigger Button Mechanical Design</td>
<td>48</td>
</tr>
<tr>
<td>26</td>
<td>Laser Gun On/Off Button</td>
<td>49</td>
</tr>
<tr>
<td>27</td>
<td>Laser Diode</td>
<td>50</td>
</tr>
<tr>
<td>28</td>
<td>Laser Sight Schematic</td>
<td>51</td>
</tr>
<tr>
<td>29</td>
<td>Laser Diode Circuit Board</td>
<td>51</td>
</tr>
<tr>
<td>30</td>
<td>Laser Sight Preliminary Implementation</td>
<td>52</td>
</tr>
<tr>
<td>31</td>
<td>Final Implementation of Laser Sight</td>
<td>52</td>
</tr>
<tr>
<td>32</td>
<td>Laser Gun Scope Drawing</td>
<td>53</td>
</tr>
<tr>
<td>33</td>
<td>Laser Gun Scope</td>
<td>53</td>
</tr>
<tr>
<td>34</td>
<td>WiFly Implementation Schematic</td>
<td>54</td>
</tr>
<tr>
<td>35</td>
<td>Microchip WiFi Transceiver</td>
<td>55</td>
</tr>
<tr>
<td>36</td>
<td>Sample IR Pattern read from Arduino EEPROM</td>
<td>58</td>
</tr>
<tr>
<td>37</td>
<td>Laser Gun Software Flow Diagram</td>
<td>59</td>
</tr>
<tr>
<td>38</td>
<td>Sensor Vest Prototype</td>
<td>62</td>
</tr>
<tr>
<td>39</td>
<td>Sensor Circuit Board</td>
<td>63</td>
</tr>
<tr>
<td>40</td>
<td>Sensor Vest Prototype</td>
<td>63</td>
</tr>
<tr>
<td>41</td>
<td>Sensor Vest Prototype</td>
<td>64</td>
</tr>
</tbody>
</table>
Figure 42 - Sensor Vest Prototype ................................................................. 64
Figure 43 - Android Application Get Player Command ........................................ 67
Figure 44 - Statistics Activity Layout ................................................................. 68
Figure 45 - Application Layout without Expanded Text Editor ............................. 69
Figure 46 - Android Application Flow Diagram .................................................. 70
Table of Tables

Table 1 - Cumulative Team Hours ................................................................. 9
Table 2 - Signal Information regarding the System Architecture .................. 19
Table 3 - Viewable Statistics ........................................................................ 23
Table 4 - BeagleBone Black vs. Raspberry Pi Comparison Chart .................. 29
Table 5 - Command Word Bit Fields .............................................................. 36
Table 6 - Data Word Bit Fields ....................................................................... 36
Table 7 - Device Type Identifiers ................................................................... 37
Table 8 - Player Identifiers ............................................................................ 37
Table 9 - Android Application Commands ..................................................... 38
Table 10 - Get Statistics Data Fields ............................................................... 39
Table 11 - Laser Gun Commands .................................................................... 39
Table 12 - Server and Base Station Commands ............................................. 40
Table 13 - Central Hub Bill of Materials ......................................................... 42
Table 14 - Wireless Communication Protocol ............................................... 45
Table 15 - Decision Matrix for Laser Gun Microcontroller ........................... 45
Table 16 - Laser Gun Budget ......................................................................... 61
Table 17 - Sensor Vest Budget ....................................................................... 65
Table 18 - Android Application Command List ........................................... 66
Table 19 - Mobile Application Platform Decision Matrix ............................. 71
Table 20 - Decision Matrix for User Control ................................................. 72
# Table of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>Amperes</td>
</tr>
<tr>
<td>AC</td>
<td>-</td>
<td>Alternating Current</td>
</tr>
<tr>
<td>DC</td>
<td>-</td>
<td>Direct Current</td>
</tr>
<tr>
<td>DHCP</td>
<td>-</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>EEPROM</td>
<td>-</td>
<td>Electrically erasable Programmable Read Only Memory</td>
</tr>
<tr>
<td>HUD</td>
<td>-</td>
<td>Heads-Up Display</td>
</tr>
<tr>
<td>I2C</td>
<td>-</td>
<td>Inter-Integrated Circuit</td>
</tr>
<tr>
<td>iOS</td>
<td>-</td>
<td>iPhone Operating System</td>
</tr>
<tr>
<td>IR</td>
<td>-</td>
<td>Infrared</td>
</tr>
<tr>
<td>kHz</td>
<td>-</td>
<td>kiloHertz</td>
</tr>
<tr>
<td>LASER</td>
<td>-</td>
<td>Light Amplification by Stimulated Emission of Radiation</td>
</tr>
<tr>
<td>LED</td>
<td>-</td>
<td>Light Emitting Diode</td>
</tr>
<tr>
<td>PPFS</td>
<td>-</td>
<td>Preliminary Project Feasibility Study</td>
</tr>
<tr>
<td>PWM</td>
<td>-</td>
<td>Pulse-Width Modulated</td>
</tr>
<tr>
<td>RAM</td>
<td>-</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>SRAM</td>
<td>-</td>
<td>Static Random Access Memory</td>
</tr>
<tr>
<td>SPI</td>
<td>-</td>
<td>Serial Peripheral Interface</td>
</tr>
<tr>
<td>UDP</td>
<td>-</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>WLAN</td>
<td>-</td>
<td>Wireless Local Area Network (WiFi)</td>
</tr>
<tr>
<td>V</td>
<td>-</td>
<td>Volts</td>
</tr>
</tbody>
</table>
1 Project Overview

Team Lazer-Ops consists of four senior engineering students in the electrical/computer concentration who are currently enrolled at Calvin College. Team member biographies and responsibilities are outlined in the following sections. The home-laser tag system fulfills the senior design project requirement of the college. The following section will discuss the senior design project including an overview, differentiation, goals, and target consumers of the project.

1.1 Calvin Senior Design

During senior year, engineering students at Calvin College participate in a one year long (two semesters) senior design project – termed ENGR 339 and ENGR 340. Students from all four concentrations are obligated to take part in a senior design project as a requirement for graduation. The course allows students to form their own groups, develop project ideas, and manage several business and project aspects from start to finish. Throughout the year, students are able to work with the idea and develop a prototype to demonstrate these abilities.

1.2 Problem Statement

Laser tag has been around since 1984 as a fun and interactive alternative to contemporary video games. Laser tag demands similar team-based strategy but gets kids off the couch and moving around. However, for 29 years it has not changed much in concept. Teams of players are equipped with laser guns and vests. The players fire at members of the opposing team. Successful hits gain points for that team and prevent the hit player from firing for a given amount of time. The game ends after a set amount of time, victory going to the team with the highest accumulation of points. At the conclusion of the game, players receive their individual scores.

Instead of going to an arena in order to play laser tag, we aim to bring the laser tag experience to the consumer in the convenience of their own home or any other area with a secure Wi-Fi connection. This will give users the freedom to alter their game play and environment however they desire. This will differentiate the experience from that of traditional laser tag and add an additional benefit of convenience. There are several home-based systems currently available on the market. These systems do not include a mobile application, central hub, and sensor vests making our system the single most immersive and complete laser tag system available.

1.3 Project Overview

This section overviews various components essential for project completion including team organization, individual responsibilities, and the work breakdown schedule. It also states the project goal, product differentiation, and the targeted consumer market.
1.3.1 Project Functions

The system will utilize four separate components to create an enjoyable and thrilling game experience. This system will contain a central hub, sensor vests, laser guns, and a mobile application. This system can be seen in Figure 1. The central hub will visually display each team’s score, a count-down timer, as well as three status LEDs showing the central hub’s status. The mobile application will deliver real-time game statistics to individual players. These statistics include each player’s shooting percentage, remaining ammunition, and number of times the user has been hit. The user will be able to view their own personal statistics, any other player’s statistics, or an overview of the current game’s statistics. The laser gun will provide sound feedback to the user each time they fire.

![System Mock Diagram](image)

**Figure 1- System Mock Diagram**

1.3.2 Differentiation

A standard laser tag experience involves locating a nearby laser tag arena, travel to the said arena, and a cost of about 5-10 dollars per game. Often laser tag facilities will offer group rates or special deals on certain days of the week. Games last from 10-30 minutes, with a time for briefing and vesting. Some laser tag arenas are able to handle up to 30 players, 15 per team. Depending on the system, players are given a printout revealing their team and individuals’ scores at the conclusion of the game. Our system aims to differentiate itself in three key areas: user awareness, convenience, and game flexibility which will be explained in further detail below.
Firstly, our system is designed to provide the users with more awareness throughout game play. The proposed system would establish a more transparent form of gameplay where players can access real-time game statistics in real time as they play. Current systems, as described above, either do not inform players of their scores and team scores until the game’s conclusion or they provide users with their personal score alone. Without the ability to know how their team is doing or how they are playing in reference to other players, it is hard to develop a meaningful game strategy. By providing the players with player scores, team scores, and other various statistics, the system will convey our design norm of care which is explained further in section 1.4.3. Our product aims to promote problem solving and logic building, all of which are enhanced by having more knowledge on the current situation in the game. This knowledge and self-awareness will allow for a solid basis on which to base game strategy and critical thinking and add a new level of complexity to game play. For example, if players know that the game is near its end and their team is down by a few hits, they could quickly plan a strategic attack on a weaker member of the opposing team in order to pull ahead. Conversely, if their team is winning by a small margin near the game’s conclusion, they could see the opposing team’s lead scorer and avoid an encounter. Mid-game a better strategy may be to try to ambush the opposing team’s lead scorer and plan a secondary strike while their lead scorer’s laser gun is temporarily inoperable.

Convenience is a large benefit to the Lazer-Ops system. The mobility of the system differentiates this system from standard laser tag systems. It takes time and planning to get a group of people together and to the laser tag arena. It is dependent upon the arenas’ availability and hours of operation. Having a home-system allows the users to play laser tag without having to coordinate transportation, depend on the facility, deal with crowds, waiting time, and cost per game. A game of laser tag can be played at the users’ convenience with the Lazer-Ops system.

Traditional laser tag has been enjoyed by many for decades. However, without much game variability, it loses its draw. The Lazer-Ops system aims for a game flexibility. Players will be able to choose different game modes (team or individual play) as some current systems also offer. Additionally, through home use, the potential to play in various environments increases greatly. Users can play outdoors and indoor and create their own “arenas” and alter the game based on user characteristics such as age or skill level. Bringing laser tag to the users allows them to interact with the game and incorporate their own creativity into game play. This is perhaps the biggest restriction of existing laser tag systems. In contrast to current “one-size-fits-all” laser tag games, Lazer-Ops desires to create a dynamic system where the user experience is the central focus and where the user has control over when, where, and how game play is played.
Throughout its history, laser tag technology has remained relatively the same (Schurek 1). However, technology in general has been changing at an exponential rate. Our system incorporates this rapid change in technology as another point of product differentiation. Creating a user-friendly, easy to use mobile application will provide players with options they have never experienced before. Not only will they be relayed live statistics, but they will be able to choose a type of game play, team or individual play with great potential for expansion, such as a capture the flag mode - all by simply downloading the newest application. If the team or individual is playing particularly well (accuracy, consecutive hits, etc.), the system will incorporate bonus packages which could range from extra ammunition, rapid fire, or extra health to information on other players’ whereabouts or concealing one’s own whereabouts.

1.3.3 Target Consumers

The focus of the Lazer-Ops project is to develop a system which essentially targets the same market as traditional laser tag. One website advertises that laser tag is for anyone ages 6 to 96. Three targeted consumer groups in particular are existing laser tag arenas, young adult gamers, and families. Laser tag arena which still use old systems could benefit from technology that is reliable and much more profitable. Customers will be attracted by new game features and technology.

Young adult gamers will be intrigued by the new system and the system will appeal to technology lovers who desire a new twist on a classic game. Also the convenience factor makes it easy to start impromptu games at any time. Children could benefit physically and socially from an at-home laser tag system. Laser tag systems encourage active participation, quick thinking, teamwork, and communication. Laser tag provides the perfect opportunity for children to play together, but in a controlled situation within the home. It also eliminates the stress on parents having to deal with recurring costs and transportation, while providing their children with a “birthday party” type activity, able to easily accommodate various amount of players.

1.3.4 Project Goal

Team Lazer-Ops team has fully-functional product, which was obtained by the conclusion of the spring 2014 semester (May 10). The Requirements chapter of this report specifies what fully-functional is defined as, but a brief overview can be found below:

- Operational gun/vest systems
- 1 central hub for processing the wireless signals (live)
- 1 operating Android application that will track game statistics
- Interface for assigning player identities to gun/vest systems

In addition to our final product, another objective is to research and present the feasibility of the project, specifically in areas of cost, consumer demand, and an overall business strategy.
1.4 Team Organization

This section outlines the team organization by giving a short biography of each team member, and their respective responsibilities.

1.4.1 Team Member Biographies

This section describes the different teammates along with a short history of each.

1.4.1.1 Matthew Block

Matthew was born and grew up in Grand Rapids, Michigan. There he attended Grand Rapids Christian High School. His interest in Electrical and Computer Engineering comes from inspiration from his father and his interest in computers. In his spare time, Matt enjoys hanging out with friends, playing video games, and developing android applications. He has participated in a study abroad trip in Europe.

Matthew has a strong interest in software development. Following this interest, Matthew has accepted a technical services position at the company Epic Systems starting in June. While working at Epic Systems, Matthew hopes to continue expanding his knowledge and skills developing mobile applications in his off time and eventually hopes to release his own games for Android, iOS, and Windows Mobile.

1.4.1.2 Eric Bouwkamp

Eric was born and raised in Grand Rapids, MI where he graduated from Grand Rapids Christian High School in 2010. Post-graduation, Eric immediately started the Engineering program at Calvin College where he will graduate in May 2014 with a Bachelor of Science in Engineering with a concentration in Electrical and Computer Engineering.

During the past summer, Eric gained significant software experience working as an engineering intern at L-3 Avionics systems where he developed and verified test software. He has continued his employment through his senior year of college working on hardware design and sensor testing.
1.4.1.3 Kristen Herder

Kristen's roots lie in Portage, Michigan where she grew up and graduated from Kalamazoo Christian High School. Her interest in Electrical Engineering comes from her curiosity and love for creative problem solving. In her spare time, she enjoys leading Calvin's Soccer Club, studying other languages, and serving on the honors student council. She has participated in two study abroad programs in Europe.

Kristen worked as an intern for two years in the Automotive Electronics Business Unit at Johnson Controls Incorporated, four years as a peer tutor, and two years as the Society for Women in Engineering (SWE) student leader for the engineering department. Similar to Matt, she has accepted a job at Epic Systems in a technical services role beginning in June. She looks forward to entering and exploring the medical industry and discovering how to help others through the application of electrical/computer engineering.

1.4.1.4 Mark Willard

Mark is from the small town of Yale, located in the thumb of Michigan. Following high school, he enrolled at Calvin College where he will graduate with a Bachelor of Science in Engineering with a concentration in Electrical and Computer Engineering. Mark played one year of basketball for Calvin College during his sophomore year, and he continues to play recreationally. His other interests include leading Calvin’s Chess Club, as well as making other people laugh.

Apex Controls is Mark’s current employer. His previous work experience includes being a product design intern at Innotec, where he utilized his training in Computed Tomography to perform X-ray and CT scans on particular lighting and illumination products in order to discover failure modes. He has also had an industrial engineering internship at Benteler Automotive, and a conveyor assembly technician job at Dematic, both located in Grand Rapids, MI. Upon graduation, Mark will start as a controls engineers at Apex Controls.
1.4.2 Team Member Responsibilities
   This section describes the different responsibilities taken during the duration of the project.

1.4.2.1 Matthew Block
   Matt’s main job for the design project is to develop and create the android application. The android app will serve as a personal user interface for each player. The basic function of this application will be to allow players to access their stats while playing the game. This has mostly consisted of User Interface Design, and UDP communication.

1.4.2.2 Eric Bouwkamp
   Eric’s focus during the project was primarily on the central hub and its integration with the other system components. This included designing specific data protocols for information transfer between various components of the entire system. Because of this, and the management of game data, the central hub portion of this project consists of a significant amount of programming, Wi-Fi data communications, and some physical hardware integration. This hub will have the ability to send and receive data between the system components accurately and efficiently, and display some essential data on the unit itself.

1.4.2.3 Kristen Herder
   Kristen’s primary focus is on the laser gun design. The main task is to produce a laser gun that sends an infrared signal unique to that gun so that it can be received and recognized by other players’ vest and processed correctly. The laser gun will keep track of the number of shots fired and ammunition remaining. This information is sent to the sensor vest through a wire connection between the two sub-systems. It will also give the user sound feedback upon each fire. It will have an on/off switch, reload and trigger buttons, and touch-activated laser sight/scope. Due to the reliance on the infrared receiver in the vest, Kristen will be working closely with Mark to develop these subsystems simultaneously and incrementally for testing purposes.

   Kristen has experiencing working with micro-controllers as well writing software in assembly and C/C++ code. Though a previous project, she was able to gain experiencing with designing and implementing a custom PCB which gives her a good foundation for working on creating a custom PCB for the laser gun. She also has industrial contacts who will provide feedback on optimizing the laser gun design, improving safety, and aesthetics.

1.4.2.4 Mark Willard
   The focus of Mark’s work was targeted on the vest of the system, as well as the gun/vest communication with each other and with the central hub. The complexity of these tasks include the reading in the IR signal, processing the data, producing user feedback, and constructing messages to send to the central hub. In addition to these functions, the vest had to appropriately respond to Kristen’s IR signals. Thus, there was consistent communication between both members, especially regarding processing coded signals.
Mark had some experience in signal processing from previous projects, including an RFID-reader project which implemented an ATMEGA 328. He was very interested in signal processing, as well as serial communication with Wi-Fi transmission. Additionally, the PCB design of our vest and gun intrigued him.

1.4.3 Design Norms

Lazer-Ops seeks to provide a fun, new way to engage people of all ages in the classic recreational activity of laser tag. We seek to do this by producing affordable equipment and software that is user-friendly. Lazer-Ops will also provide a strong support structure for customers via cutting-edge technology and satisfaction guarantees. The company is focused on being a strong competitor in the field of home laser-tag systems in the near future. There is also great potential for breaking into commercial laser tag locations.

In order to deliver the goals discussed, the principles that guide our mission are the following:

**Transparency**: Transparent design is implemented through open and understandable communication and the creation of a consistent, reliable, and predictable product. Lazer-Ops will produce a system that has easily-defined purposes and parts, that is, the system will be exceptionally user friendly. Easy-to-follow documentation will be included in the product.

**Care**: A caring product is one which takes into account the physical, social, and psychological effects on the targeted consumer. Lazer-Ops will take into account the effect on individuals physically, socially, and psychologically. From a physical standpoint, the game will promote physical activity and fitness. Socially, the product gathers people together. Psychologically, Lazer-Ops intends to promote a message of team-work and competitiveness, rather than violence that some products convey.

**Stewardship**: A design which uses thoughtful allocation of its resources is one that implements the design principle of stewardship. Lazer-Ops will be efficient in its allocation of time and resources. Cash flow management will be consistently effective such that the highest rate of return on assets may be maintained. Additionally, manufacturing principles will secure environmental friendliness.

**Trust**: An honest, dependable, and reliable product is one with trustworthy design. Lazer-Ops will produce cost-effective and structurally sound equipment. The product will be dependable and reliable, and it will avoid any conflicts of interest regarding business practices.
1.5 Schedule Management

1.5.1 Work Breakdown Schedule

A copy of the team’s work breakdown schedule can be found at the following location: http://www.calvin.edu/academic/engineering/2013-14-team11/Files/team11_scheduledWBS.pdf.

1.5.2 Cumulative Hours

A table containing the cumulative hours worked for each team member can be found below in Table 1. While the cumulative column contains the total number of hours, this value is also broken into number of hours worked per semester.

<table>
<thead>
<tr>
<th>Name</th>
<th>1st Semester Hours</th>
<th>2nd Semester Hours</th>
<th>Cumulative Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matthew Block</td>
<td>93</td>
<td>201</td>
<td>294</td>
</tr>
<tr>
<td>Eric Bouwkamp</td>
<td>102</td>
<td>291.5</td>
<td>393.5</td>
</tr>
<tr>
<td>Kristen Herder</td>
<td>94</td>
<td>244</td>
<td>338</td>
</tr>
<tr>
<td>Mark Willard</td>
<td>90</td>
<td>304</td>
<td>394</td>
</tr>
</tbody>
</table>

Table 1 - Cumulative Team Hours

1.6 Document Structure

In the chapters that follow, the design team describes the requirements and specifications of the proposed laser tag system, gives a general overview of the laser tag system as a whole, and provides detailed descriptions of each subsystem. The final chapters are composed of descriptions of the prototypes that have been created. This document will conclude with a statement about the overall design of the completed system. The design team will acknowledge any and all people that provided support and insight into the design of the system. Finally the design team will report all sources that were used as reference material in the design of this project.

The next five chapters deal with providing a description of the system as a whole as well as descriptions of its individual subsystems. The first chapter focuses on an overview of the laser tag system as a whole. In this chapter the design team will describe the components that comprise the system and the communication that will be necessary between the subsystems. The next four chapters describe the central control hub, laser gun, sensor vest, and android application. These chapters cover the design decisions that were made for the hardware and software components of the applicable subsystems.
The final five chapters of this report detail the prototypes that have been created or are currently being tested, a description of the desired production model, concluding remarks from the design team and an assessment of the overall systems design, acknowledgements to anyone who has given advice to the design team, and lastly a list of any references that were used in the creation of this project. The prototypes chapter provides all the specifics of all prototypes that have been created or are being created. Next, the description of the desired production model will detail what the design team would hope to accomplish if the team was able to bring the prototype to a full scale production. The following chapter will include any final remarks that the design team has. This chapter will also have a statement on the success of the overall system design. The final two chapters will give acknowledgements to any outside contributors, and will provide any references used in the design process or in the writing of this report.

2 Requirements

This section breaks down the project requirements based on category and subsystem. The categories covered include technical, customer, user interface, communication, and environmental requirements. It concludes by establishing the project deliverables.

2.1 Technical Description.

This section gives an in-depth description of the technical requirements set in place for each subsystem. These requirements will be met by the completion of this project in the form of the May deliverables of the basic system

2.1.1 Central Hub

The central hub is considered to be the central link between the several components of the system. In order to provide an adequate game experience which provides game performance feedback, the central hub will receive and interpret data that is being sent from the other system components. Depending on the type of device, it will change or respond with game statistics, and manage the game’s environment settings.

The system will also feature a simple Microsoft Windows application that will allow users to view and change different game settings from anywhere within Wi-Fi range, as well as view and print different game statistics from the previous game. In addition to sending and receiving data between various devices, the central hub will also feature an easy-to-use front display that will be visible from short to medium range distances. This data will include both team’s current score, and the game time remaining in the current match.
2.1.2 Laser Gun

The laser gun will send coded IR signals when the trigger is pulled. The laser gun sends coded IR signals when the trigger is pulled. The use of infrared is standard in laser tag systems as the signal can be encoded to identify players and also reduces interference from other devices. Each IR signal is unique to a given laser gun. When decoded by the vest this will enable it to distinguish which particular user the signal is coming from. The infrared beam in the prototype system must have a range of accuracy with a minimum at 10 ft. Similar to standard laser tag systems, the Lazer-Ops system incorporates a laser sight for the visual effect of firing though the laser itself produces no transmission of a firing signal. When a player pulls the mechanical trigger a button within the gun enclosure is depressed signaling that the user is firing. With each trigger pull, a buzzer sounds and a red LED located on their gun flashes to give feedback to the user that the system is operating properly. An on/off status LED built into the on/off switch also lights while the system is activated and receiving power. The software of the laser gun keeps track of the amount of ammunition left. When a player has run out of ammunition, their gun ceases to fire and the status LED does not light until the player reloads their gun restoring their ammunition count. The player reloads their ammunition upon pressing the reload button. Upon reload, a reload status LED which is the same color as the player’s team stays lit while their gun is reloading. The player cannot fire during this time. The laser gun sends information such as shots fired and ammunition count via a tether to the sensor vest. All the circuitry will be enclosed in a plastic housing. A PVC tube and lens are used to focus and collimate the IR LED and to extend its firing range. The laser gun contains a 9V battery which powers both the laser gun and sensor vest. Figure 2 shows the basic inputs and outputs of the laser gun sub-system.

![Figure 2 - Laser Gun Inputs/Outputs](image)

Figure 3 below is a chart of the Team’s requirements of the gun and vest. Some of these requirements were considered goals to attain if time and budget permitted, but the basic goals were necessary for a basic prototype.
Core Features of the project were not absolute requirements, but many were still obtained. The required distance of the laser gun, for example, was only required at short distances (< 5’), but anything above that was strongly desired by the team.

Any Upgrade Features were to be used if all of the Core Features were first taken care of. Battery life was successful in this category, as well as a 10’ accuracy and distance.

The Stretch Features of the requirements were, in actuality, a strong wish list of what the team could have supported with not only more planning, but extensive time and a vastly larger budget.

2.1.3 Laser Vest

An important design for the sensor vest was that it cover the majority of IR reception—that is, 360 degrees. The final prototype incorporated 4 infrared sensors to maximize the total reception of the signals. This can be seen in Figure 4 - Sensor Placement Diagram.
The power for the LEDs and IR sensors must be generated by a standalone power supply (namely, a battery). Additionally, the collector pins for each transistor that control LED lighting (team color and red) must be wired to the microprocessor. The demodulated output signal and ground wires will also be required to connect with the Arduino base. Either a tethered connection or wireless connection is also required between the gun and vest.

Bluetooth is an obvious option of communication between the gun and vest. Despite the clear advantages of wireless communication such as Bluetooth (higher freedom of movement, e.g.), the team decided that, while wireless communication would add immense gameplay advantages, the addition was simply beyond the scope of our project. To add to the complexity, two power supplies would be required to power the vest and gun separately, and the protection that the gun provides for the microcontroller would be negated, as it would have to be placed in the vest. This would become a viable improvement for future projects.

2.1.4 Mobile Android Application

The mobile application will be responsible for showing players their in-game statistics. The application will communicate with the central hub system over Wi-Fi using the User Datagram Protocol. This communication will require the sending and receiving of UDP packets. Once the UDP packets from the central hub are received, the mobile application will need to decode the data that the packets contain. Once the data has been decode, it will be displayed to the user on the phone’s screen. This system will feature an intuitive and simple to use application layout and activity flow.

2.2 System Features

A complete laser tag system package would include 2 guns, 2 vests, and 1 central hub. In order to utilize the Android application during gameplay, the application will be downloaded from the Google Play Store. This application will be free, and more information on how to download it will be included with the unit. This laser tag system will feature infrared communication between laser guns and sensor vests, and Wi-Fi communication between sensor vests, Android application, and the central hub. This system will feature in-game statistics tracking allowing players to see what the current score during the game on their mobile device.

2.3 Customer & Market Requirements

This section outlines various requirements for everyday use by an average consumer. In order for our system to be reliable, these requirements must be implemented.
2.3.1 Functional Requirement

For the system to function properly, the laser guns must be able to fire consistently within 1ms of trigger press and also reload within 1ms of the reload button press. Consistency is key in order to build trust in the product's ability to perform as expected. The user feedback which includes the status LED and piezo buzzer must give feedback within 1ms of trigger press and the sending of the IR signal. These feedback systems allow for transparent operation letting the user know exactly how the system is functioning. The reload LED must also light within 10ms of the pressing of the reload button. The IR signal must be able to be received at least 15 feet away from a player. When the infrared signal is sent and received, it must take no more than 30ms to be decoded and restrict the player’s ability to fire. The sound feedback from the buzzer must not exceed 85dB to care for physical effects on the user such as hearing damage. Likewise, the laser sight must be class 2 or 3R. It should not exceed 5mW to avoid any potential eye damage to users. The system must also record ammunition usage and record when the infrared sensors are hit. The hub must be able to receive this data and process all the game information into useful statistics. Once the data has been processed, the hub must relay the information to the mobile application which must display it to the users. The hub must allow the users to sign into the game’s elapsed time (with display), and time left in each game.

2.3.2 Weight Limit

The maximum weight desired for the vest as a whole was 1.8 kg +/- 0.5kg. This is the approximate weight of a winter jacket, as found by the team. The limit should be a fairly easy goal meet, as a single-layer PCB is less than 20 grams (leiton), which is the largest contributing factor to the weight of the vest (aside from the material). The material weighs 9.25oz per square yard. Using an overestimate of 4 square yards per vest gives a weight approximation of 1.05kg.

The gun’s weight should imitate the weight of a common NERF© gun; the gun weight shall not exceed 2kg. The user should be able to hold the gun comfortably for at least 4 hours at a time, without exhibiting noticeable fatigue due to the weight of the gun.

2.3.3 Type of Environment & Use

This production system will be versatile and durable and have the ability to be used both indoors and outdoors. It order to create a reliable product, it must be capable of handling various elements from the outdoors such as wind, rain, and humidity. The prototype, however, focuses on software and hardware feasibility and it is not able to stand harsh outdoor elements such as rain, mud, or snow. Because of this, the prototype home laser tag system is intended for indoor settings only. The system will withstand temperatures from 0 to 32 degrees Celsius, as well as a relative humidity of up to 95%.

2.4 Mechanical Requirements

This section outlines several of the mechanical requirements that the subsystems have to ensure product operation and integrity. All subsystems will implement safety measures to protect them from the rough use that they might receive during gameplay.
2.4.1 Central Hub Enclosure

In order to protect the components of the central hub, all of the components will be mounted in an enclosure. These components will be mounted in an organized fashion to the inside of the enclosure in order to create a stable environment that provides proper ventilation for all components. The central hub enclosure must be at least 9” long, 6.5” wide, and 3” tall to ensure that all electrical components are able to be enclosed properly.

2.4.2 Sensor Vest Enclosure

The primary requirement for the encasements is to protect the IR sensors and the LEDs. Additionally, the material must be constructed of a transparent plastic, so as to limit any infrared signal blocking and to maximize visual feedback via the LEDs. As far as the limiting force that the encasements are required to handle, it should withstand up to 1.14 kN, which is the force due to the weight of a 250-pound person falling on the enclosure. (113.4kg*9.8kgm/s²=1.11kN)

2.4.3 Laser Gun Enclosure

A plastic housing will be used to contain and secure the laser gun’s circuitry. The case must be durable enough for rough gameplay, yet light enough so that it does not burden the user. Typical wall thickness of 3-4 mm should be used to accomplish these purposes in the enclosure. A material with a tensile strength of no less than 34 MPa is ideal to provide the casing enough strength to withstand game roughness. The ideal total weight for the gun circuitry and casing should be no more than 2.0 kg and it should be no longer than 40cm. The casing must incorporate a trigger (way for the user to fire), a way for the user to turn the system on and off, and a way for the user to reload their gun. The laser gun must provide some feedback to the user to allow for transparent operation of the laser gun system; making sure the user is aware whether the product is functioning properly or not. The production model will incorporate a “Heads-up Display” (HUD) in order to allow players to attach their phone directly to their gun to display the game information hands-free and in a convenient location as they play. This structure should be adjustable to adapt to differing mobile phone devices (up to 6” screen size). This structure will be 4-6mm in thickness for extra stability.

2.5 User Interface Requirements

This section outlines the basic requirements needed for the system to operate as far as features and functions are concerned. In order to control the system and the game experience, these requirements must be in place.

2.5.1 Inputs

The laser-tag system will consist of only two sources of input. The first and primary source would be through the graphical user interface of the central control hub. This would allow a user to initialize the game, change game settings, and even view the previous game’s statistics.

Other inputs that exist within the laser gun include the main power switch, the reload button, and the trigger button.
2.5.2 Outputs
The Android application will relay gameplay statistics to the players via a digital display on an Android device. These statistics will include the amount that the laser gun was fired since the start of the game, the number of shots that hit another player, the percentage of shots that hit opposing players, the average number of shots fired per minute, the score for an individual player or for a team, and the number of times the player has been hit. These statistics will be recorded or calculated by the central hub using data collected from the vests and guns of all the players in the current game. The game is started and can be stopped through the windows application. As players turn on their guns and vests, connecting to Wi-Fi, the hub recognizes them as players and assigns them a player number. There will be additional outputs for user feedback. LEDs will blink on a player’s vest when they get hit. Also whenever a player pulls the trigger on their laser gun, LEDs will flash. Additionally when a game is finished the lights on both the gun and the vest will blink for a few seconds.

2.5.3 Heads-up Display & Mobile Android Application
The purpose of the Android application is to provide Lazer-Ops players, and potentially spectators, with a way to interact with the game. The application will provide an immersive way to play laser tag, because it will allow players to see how they are performing relative to the other players participating in the game. This will be accomplished by creating an intuitive, but engaging user interface. A user interface that is able to accomplish this will need to be easy to use, to be able to provide statistics in a reliable manner, and to be captivating and intriguing enough that players will use it while they use the Lazer-Ops system. For the prototype model, this display screen will not be customizable and will show stats only after a request has been sent to and a response has been received from the central hub. In a production scale application the user interface would be customizable.

2.6 Communication Requirements
In order for the system to operate to its full potential, the subsystems must be able to communicate with the central hub at any point in the duration of the game cycle. The team requires that the communication between the central hub and the gun/vest subsystems shall be wireless, whereby the hub is recognized as the server, and each player carries a wireless client.

2.7 Power Requirements
In order to provide a fun, enjoyable experience for the consumer, the team’s goal is to provide a sensor and laser gun with a minimum 4 hour use time per charge. As for the prototype, we require that the battery power last for at least two hours--that is, long enough to get through Senior Project night with no battery changes. This requirement meant that each gun/vest system must only consume 600mAh for an average lithium 9V battery to be sufficient (1200mAh).

2.8 Environmental Requirements
This section outlines the environmental requirements that were considered during the duration of this project.
2.8.1 Power Efficiency

The central control hub unit will be plugged into a standard wall receptacle (115V). This signal will then be transformed, rectified, and regulated to provide clean power for the Raspberry Pi, wireless router, display board and router circuit boards. The supply must provide at least 1.5 Amps of current at 12V for the Wi-Fi router, and about 500 mA of current at 5V for the Raspberry Pi and display board.

2.8.2 RoHS Compliance

In order to comply with the Restriction of Hazardous Substances Directive, lead-free solder will be used to construct the PCB’s for a full-scale model. Also, our system will not contain any other banned substances. For a detailed list of these banned substances, see http://www.bis.gov.uk/nmo/enforcement/rohs-home.

2.9 Deliverables

This section describes the deliverables that will be presented as a result of the project in its entirety. These will be delivered in May 2014.

2.9.1 PPFS

Team Lazer-Ops will submit a complete project proposal and feasibility study, which will encompass a proposed solution to a presented problem. The document will also detail the hurdles involved in reaching the proposed solution.

2.9.2 In-Class Presentations

Team Lazer-Ops shall give four presentations to the senior design class regarding the progress of the project, as well as what will be done on the project following the presentations. These presentations will be distributed throughout the year.

2.9.3 Project Posters

The team will complete a poster describing the team and project that will be displayed at the team’s project station in the engineering building. The poster will be updated throughout the year to include updated block diagrams, drawings, and details.

2.9.4 Final Report

The team will submit a final report detailing the solution to the problem outlined, the research behind the solution, the obstacles presented, the construction and testing of a prototype. Also, the document will cover the conclusions gathered from prototype testing, including the changes implemented into the final construction.

2.9.5 Final Presentation

The team will be present on Senior Design Night, Saturday, May 10, 2014. Team Lazer-Ops will demonstrate the final project and its functionality, in addition to answering any questions from interested persons.
2.9.6 Working Prototype

Team Lazer-Ops will demonstrate a complete, functional prototype, the requirements of which are discussed in section 2.

2.9.7 Design Notebooks

Each member of Team Lazer-Ops shall submit all of his/her relevant documents and folders to the faculty advisor, including individual contributions to the project.

2.9.8 Team Website

Team Lazer-Ops will publish a website available to any person interested in the project at http://www.calvin.edu/academic/engineering/2013-14-team11/.

2.9.9 System Software

All control software will be submitted with the final report. This software will be linked to using a directory path or link to the repository.

2.9.10 Drawings and Schematics

All CAD drawings, circuit schematics, and PCB designs shall be submitted with the final report.
3 System Architecture

This section outlines the system’s structure as a whole. This is done by first examining the overview of the system, and then each subsystem in more detail.

![System Diagram](image)

Figure 5 - System Diagram

Table 2 - Signal Information regarding the System Architecture

<table>
<thead>
<tr>
<th>Connection</th>
<th>Type of Communication</th>
<th>Voltage (if Applicable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>①</td>
<td>Wi-Fi</td>
<td>NA</td>
</tr>
<tr>
<td>②</td>
<td>Wi-Fi</td>
<td>NA</td>
</tr>
<tr>
<td>③</td>
<td>Serial</td>
<td>9V and 5V</td>
</tr>
</tbody>
</table>
3.1 Overview

The Laser Tag system will consist of four main components. The first component, the central control hub, will establish and maintain communication between all the separate components of the system, as well as manage the game environment. The hub will consist of a Raspberry Pi Linux computer which will store and interpret data from the other components of the system. The second component is the mobile application. This, initially using the android development framework, will provide a user interface that can be used to update game settings, view statistics, or create a new game. The next two components consist of the sensor vest, and the laser gun. The sensor vest will be used to sense incoming IR signals from other players, while the laser gun will send these modulated infrared signals. When game events occur (such as gunshots, hits, and new player initializations), the sensor vest will communicate with the central hub in order to update the score.

3.2 Components

This subsection describes the individual components of the laser tag system individually. These components are all essential to the operation and performance of the system as a whole.

3.2.1 Central Control Hub

The central control hub will consist of a small processor, a wireless router in order to control the game data, a display board to control the hardware display, and a power supply board. This computer will be running a simple server in order to update game information, store game information, and to provide the backbones for the general communication between the other components. This hub will also include basic status information displayed on the front of the unit. This status information will include the scores of either teams, the time remaining in the current game, and status LEDs for the system’s status.

3.2.2 Laser Gun

Two laser guns are to be a part of the May deliverable. The laser gun is the apparatus for transmitting the IR signal when the trigger is pulled. Each player holds the device and points it in the direction of the other players. A player is hit when the IR beam is received by the vests of a player on the opposing team. A key objective of the laser gun sub-system is to ensure that each gun transmits an infrared signal specific to each laser gun upon trigger press. Upon game initiation, the central hub stores which players are members of each team. The system can process statistics on a team and individual basis, such as personal and team hit count, shooting count, shooting percentage, personal score, and overall score. Another essential objective of the laser gun is to enable the IR signal to be transmitted long-range without losing the integrity of the particular encoded signal (because only certain programmed signals affect the receiver vest). The prototype system has a tested range of averaging around 5 feet in broad daylight. The laser gun houses the microprocessor as well as the Wi-Fi communication chip so as to be protected by the plastic housing unit.
3.2.3 Sensor Vest

The sensor vest will consist of a vest and four housings for the sensing circuit. The encasements will house the LEDs and the spatially placed IR sensors. The LEDs and sensors are controlled by a single control system, and are thus wired to the microcontroller. Figure 2, below, shows a conceptual diagram (with component placements) of the vest.

From the conceptual diagram, note that the yellow circles represent LEDs and the purple blocks indicate IR sensors. The cords between the shoulder and back encasements will contain one wire for each of the following signals: 9V, 5V, GND, 2 LED strands, and the IR receive signal. When the IR receive wire is activated via a PWM IR signal, the output current is a maximum of 5 mA (at 5V). The constant supply current on the 5V line is a maximum of 3 mA. In either status of a player (standby or “shot”), the 9V pin transmits approximately 125 mA to either the team LED strands or the red LED strands, respectively. The maximum possible operating current consumption is 240 mA. This is calculated using a current regulating resistor of 180 ohms. From the calculations below, we find that the maximum current consumed by all active LED strands is limited by the LED maximum current rating (30mA).

\[
\begin{align*}
R(\text{worst case}) &= 180 \times (0.90) \text{ ohm} = 162 \text{ ohm} \\
V(\text{LEDmin}) &= 1.8 \text{ V} \\
V(\text{resistor}) &= 9 - 1.8 \times 2 \text{ V} = 5.4 \text{ V} \\
I &= \frac{5.4 \text{ V}}{162 \text{ ohm}} = 33.3 \text{ mA} \\
33.3 \text{ mA} &> 30 \text{ mA}, \text{ the maximum forward current through this LED}
\end{align*}
\]
30 mA * 8 strands = 240 mA

The current from the 9V signal flows out of either of these strands through one of the two LED strand wires.

Provided the IR sensors chosen cover at least 90 degrees of reception, the placement of the sensors will allow for near-complete coverage of the individual. The LED locations provide the users feedback that a player was shot. The cable to the gun connects from a median sagittal position. This placement lets right- and left- handed people have the same gun-extension distance, and the cable will be less likely to get in the way.

3.2.4 Mobile Android Application

The mobile application will be responsible for showing players their in-game statistics. Figure 7 shows the current layout for the prototype. The user will be able to view the any player’s statistics in the upper portion of the application. Table 3 below lists the stats that an application user will be able to view depending on the whether they have chosen to view a team or a player. In the bottom portion of the application, users will be able to chat with other application users. Users will need to decide on a username before being able to chat. This function is found in the menu, which can be opened by touching the three grey squares in the upper right corner of the application.

![Figure 7 - Android Statistics Activity](image-url)
### 3.3 System Power Consumption

The power consumption of the team’s prototype has been measured to be approximately 310mA, 205mA of which is drawn from the LED strands. 100mA are drawn from the Arduino and WiFly total, and the remaining current is drawn because of the 5V regulator, IR receiver, and any given input or output that is activated.

### 3.4 Communication Overview

This section outlines the communication methods that will be in place between the subsystems. These protocols must be in place for the system to operate as a whole.

#### 3.4.1 Laser Gun IR Transmitter to Sensor Vest

The laser gun and sensor vest will communicate via infrared signals. The signals will operate at 38 kHz and they will be pulse-width modulated (PWM) to allow different sensor vests to distinguish between players. In Figure 8, a sampled IR signal can be seen. As seen, this signal is considered to be an active-low signal.

![Figure 8 - IR pattern sent by Laser Gun](image)

<table>
<thead>
<tr>
<th>Player Selected</th>
<th>Team Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual Score</td>
<td>Team Score</td>
</tr>
<tr>
<td>Shots Fired</td>
<td></td>
</tr>
<tr>
<td>Shots Hit</td>
<td></td>
</tr>
<tr>
<td>Shot Accuracy</td>
<td></td>
</tr>
</tbody>
</table>
3.4.2 Sensor Vest IR Receiver to Laser Gun

As a result of a successful signal being read by the IR receiver on each sensor vest, the received signal will be directly sent to the laser gun, via a wired connection, in order to be interpreted. This active low signal is considered to be pulse width modulated.

3.4.3 Laser Gun to Central Hub

The central hub and laser gun will communicate wirelessly. Packets will be sent from the gun to the hub indicating several events in the game, namely a gunshot, a hit, and a player initialization. In order to provide live statistics for players in the game, the latency for this level of data transmission should not exceed 10 seconds.

3.4.4 Central Hub to Mobile Application

The Android application will communicate over Wi-Fi with the central hub. UDP packets will be sent from the mobile application in response to user inputs. These packet will be received and decoded by the central hub. The central hub will create an appropriate response packet. This response packet will in turn be sent using UDP to the android application, which will in turn receive the packet and decode the data contained within.

4 Central Hub Design

This section outlines the different specifications, requirements, and design choices that have been implemented within the central control hub. This includes both the hardware and software aspects of the component.

4.1 Requirements

The central hub was built in order to create a link between the different systems of the Lazer-Ops system. In order to do this, the central hub must effectively receive, interpret, manage, and relay information from the laser gun and sensor vest to the Android and Windows applications. This is used to provide the user with a real-time game experience where they can view game information at any point.

In order to relay information between the different components of the system, a simple web server was implemented. This server allows for the central hub to interpret data, update statistics, or reply with requested information, as necessary, all within, at the most, 10 milliseconds. All of the data sent to the central hub must match the data protocol, as defined in section 4.4, in order to be processed. Once the data has been processed, the appropriate value is stored within several structures within the server.

In order to establish the central hub’s server, a Linksys WRT400N router was used. Because of the limitations on other components, 802.11b was used to transfer data between devices. As the IEEE documentation for 802.11b states, the maximum achievable throughput that can be obtained is 11 Mbps. If this were to be utilized to its fullest extent, a 320 byte message, including UDP and IP headers and the 8 bytes of data, can be sent in only 0.028 milliseconds. These calculations can be found in Figure 9.
As mentioned previously, a Windows application was created in order to view live game statistics, as well as change different aspects of the central hub’s behavior. Some of these things include things like the game duration, or variables used for debugging purposes. In order for the Windows application to maintain a real-time state, a timer was implemented so that it will update game and player statistics every 200 milliseconds.

In addition to the different software requirements, the system users are also able to examine the current game’s remaining time, as well as the score of two teams, on the front of the central hub’s display. These are displayed using three different 4-digit 7-segment displays. The scores that are able to be displayed can be no less than zero, and no greater than 999. Anything outside of these limits will be coerced to these values before the display is changed. In addition to these segmented displays, three status LED’s will also be visible to the user. The leftmost LED, which is red, indicates the central hub’s power status. The center green LED indicates the server’s readiness to receive messages from the different system. The rightmost green LED indicates when a message has been received and is being processed.

### 4.2 Mechanical Design

In order to provide an adequate environment for the components of the central hub, both electrically and environmentally, all of the components are placed inside of a plastic enclosure. The enclosure, produced by Hammond Manufacturing, was chosen to allow at least ¾” on either side of the electronics mounting bracket. As seen in Figure 10, the bracket is 8.5 inches (216mm) wide, with a small overhang of the Wi-Fi router board (approximately 0.33 inches). This bracket allows for the red-lined components to be mounted using either ½” or 1” standoffs, while the blue-lined components use 1 ½” spacers.
In order to create an aesthetically pleasing enclosure, three rectangular holes were made in the front of the enclosure to allow the 7-segment displays to be placed, along with three holes for the status LEDs to be mounted. A diagram showing the front panel can be seen in Figure 11, while a diagram showing the rear panel can be seen in Figure 12.
Additionally, power for the enclosure is provided through a strain relief connector through the back panel of the enclosure. This allows the external power to be fed into the unit while protecting components from the possible strain from transportation. While power is provided through a power brick, the end user may exchange the power cords providing the brick with power. This means that our minimum power cord length is about 25 inches. In order to toggle the power provided to the unit, a simple power switch was also mounted to the back panel. This allows for the entire unit to be powered off.

### 4.3 Hardware Design

In order to create the central control hub, four main components will be used. These four components, the Raspberry Pi Linux Computer, the Linksys router, the display circuit board and power supply circuit board, will be used to establish a communication server, manage game data, and to visually display game information. For the purposes of the system prototype, an external power supply will be used. If the product were to go into production, the power would be supplied directly into the unit where it would be rectified and smoothed. The block diagram for the central hub system can be seen in Figure 13.
In order to meet the communication and data management requirements of the system, a central processor must be used to facilitate these aspects of the central hub. In order to do this, two possibilities were considered. While both systems had the potential to fulfill the systems requirements, the Raspberry Pi was chosen over the BeagleBone Black for several reasons. The specifications of these systems can be examined closer in Table 4. These different development boards were chosen because of their incredible amount of online support, as well the Linux development environment that they both provide. Both of these development boards easily provide the processing power and speed required for less than 50 dollars.

The Raspberry Pi was chosen over the BeagleBone Black primarily because of the $10 price difference, as well as the availability at the beginning of the project. Because of these specifications, the Raspberry Pi is an ideal Linux-based computer for the central hub.
In order to provide a reliable connection between the subsystems and the central hub, a Wi-Fi router will be used to establish a 20-30 meter area where the subsystems can communicate with the central hub. This range can also easily be extended to span large areas through the use of wireless repeaters. To fulfill these requirements, a Linksys WRT400N was chosen. This router was primarily chosen because it meets all the system requirements for rate of data transfer, and also because it fully supports both 802.11b and 802.11n protocols.

Through the use of this router and the 802.11b protocol, a maximum data transfer rate of 11Mbps can be achieved, which would allow our system messages to be transferred in less than 1 millisecond, as outlined in section 4.1. This router will be connected to the Raspberry Pi Linux computer using a short Ethernet cable, and will be powered directly from the external power supply, through the power supply circuit board.

### Table 4 - BeagleBone Black vs. Raspberry Pi Comparison Chart

<table>
<thead>
<tr>
<th></th>
<th>BeagleBone Black</th>
<th>Raspberry Pi</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Price</strong> (USD)</td>
<td>45</td>
<td>35</td>
</tr>
<tr>
<td><strong>Processor</strong></td>
<td>1 GHz TI Sitara AM3359 ARM Cortex A8</td>
<td>700 MHz ARM1176JZFS</td>
</tr>
<tr>
<td><strong>RAM</strong></td>
<td>512 MB DDR3L @ 400 MHz</td>
<td>512 MB SDRAM @ 400 MHz</td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>2 GB on-board eMMC, MicroSD</td>
<td>SD Card</td>
</tr>
<tr>
<td><strong>Video</strong></td>
<td>Micro-HDMI</td>
<td>HDMI, Composite</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>5V, 210-460 mA</td>
<td>5V, 150-350 mA</td>
</tr>
<tr>
<td><strong>GPIO</strong></td>
<td>65 Pins</td>
<td>8 Pins</td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td>10/100 Mbps Ethernet</td>
<td>10/100 Mbps Ethernet</td>
</tr>
<tr>
<td><strong>USB</strong></td>
<td>1 USB Host, 1 Mini-USB Client</td>
<td>2 USB Host, 1 Micro-USB used for Power</td>
</tr>
</tbody>
</table>

4.3.2 Linksys Wireless Router

In order to provide a reliable connection between the subsystems and the central hub, a Wi-Fi router will be used to establish a 20-30 meter area where the subsystems can communicate with the central hub. This range can also easily be extended to span large areas through the use of wireless repeaters. To fulfill these requirements, a Linksys WRT400N was chosen. This router was primarily chosen because it meets all the system requirements for rate of data transfer, and also because it fully supports both 802.11b and 802.11n protocols.

Through the use of this router and the 802.11b protocol, a maximum data transfer rate of 11Mbps can be achieved, which would allow our system messages to be transferred in less than 1 millisecond, as outlined in section 4.1. This router will be connected to the Raspberry Pi Linux computer using a short Ethernet cable, and will be powered directly from the external power supply, through the power supply circuit board.
4.3.3 Power Supply Circuit Board

In order to provide adequate power for the wireless router circuit board, the Raspberry Pi, and the display circuit board, an external power supply has been used. This power supply is capable of providing up to 5 amps at 12 volts. This supply will be used to power the wireless router directly using standard terminal blocks and the standard barrel connection. In order to provide power to the Raspberry Pi, the 12 volts will be regulated using a Texas Instruments LM2576 adjustable switching regulator that is able to provide up to 3 amps at 5 volts. This will be connected to the raspberry pi using standard terminal blocks wired to a standard micro USB connector. The schematic for this circuit board can be seen in Figure 14, as well as the board layout in Figure 15.

![Figure 14 - Power Supply Circuit Board Schematic](image)

![Figure 15 - Power Supply Circuit Board Layout](image)
All components for the central hub’s circuit board were calculated using the LM2576’s datasheet. These calculations, which can be found in Figure 16. When choosing the filtering inductor, a certain $E\cdot T$ value is calculated. In order to determine the resulting inductor, Figure 27 of the datasheet is reference. From this figure, it can be seen that a 68uH inductor is needed. According to the output capacitor calculations, a minimum capacitance of 447uF is needed to ensure stable operations. Because of this, a 1000uF capacitor was chosen to increase the ability of the power supply to provide transient currents to the Raspberry Pi and display board. All other passive components were recommended directly from the datasheet.

![Figure 16 - Central Hub Power Board Calculations](image)

4.3.4 Display Circuit Board
The central hub will also be equipped with three 7-segment displays, each having four digits. These devices (BL-Q56C-43) were chosen because of their common cathode arrangement, allowing them to be driven with the Maxim Integrated MAX6955 Integrated Circuit, a multiplexing segmented display driver. This will allow players currently in the game to view basic game statistics during the course of the game. These 7-segment displays will include each team’s score, and the remaining time, in the common hours and minutes format, in the current game.

In addition to displaying game statistics using the 7-segment displays, there are three LEDs that display the central hubs current power status, server status, and communication status. These LED’s (SSI-LXH1090SRD) were chosen because they could easily be operated with the MAX6955 device, and would only draw about 20-30mA each, at least 10mA beneath the maximum GPIO current for the device.
In order to drive the various 7-segment displays, as well as the LED’s, a display board was developed. This board, which connects to the Raspberry Pi, features two MAX6955 devices that can be used to change the status of each individual segment on the front panel. In order to do this, a schematic and board were created. The schematic can be seen in Figure 17.

![Figure 17 - Central Hub Display Board Schematic](attachment:image.png)

In order to create this board, the appropriate Gerber files were set to 4PCB, a cheap PCB fabrication service. This was necessary because of the pin spacing that was needed for the MAX6955 devices. The board layout that was developed can be seen in Figure 18. This chip directly connects to the individual LEDs using JP7, JP8, and JP9. In order to connect the 7-segment displays, headers JP4, JP5, and JP6 are used.
In order to control the power entering the central hub, there will be a power switch capable of disabling all power entering the central hub. This power switch (JWM11RA1A) was chosen because it was capable of handling the maximum power that the central hub might use during the worst case operation. It is capable of handling 10A at 30V DC. This will allow a user to turn off power to the unit once the Raspberry Pi has been properly shut down.

4.4 Software Design
This section outlines the different design choices and structure involved in the design of the central hub’s software. All Central Hub Software can be viewed at:

https://bitbucket.org/ericbouwkamp/lazerpi/src
4.4.1 Programming Language Choice

The central hub’s software was written using the common programming language, C++. While other interpreted languages in the C-based family, C++ was ultimately chosen because of the team’s experience and understanding of the language. This was beneficial to the speed of the central hub because of the nature of C++ being a compiled versus an interpreted language. This allows the program to execute without the overhead of being interpreted in real-time. In order to compile the source code, g++ was used to compile the source code into an executable that could then be started.

4.4.2 Software Design Structure

In order to create a well-structured program that incorporates readability as well as simplicity, a simple structure was created using classes. The class structure can be seen in Figure 19.

![Figure 19 - Central Hub Software Structure](image)

The software was design this way to provide an organizational method to managing the various aspects of the server. This allows only the classes that require the use of services such as I2C or UDP to gain access to their methods. The UDP and I2C classes were written in order to incorporate more protection from the lower-level functions that drive these protocols, while making them easier to use in the instantiating classes.
4.4.3 Graphical User Interface

In addition to allowing users to view game information using their Android application, or using the visual display on the front of the central hub, a Windows application was developed. This application displays both teams’ current score, the remaining time, along with several gameplay statistics for the current players. These statistics include individual score, shots fired, hits confirmed, and an accuracy ratio. A screenshot of the windows application can be seen in Figure 20.

![Lazer-Ops Control Panel](image)

Figure 20 - Central Hub Windows Application

In addition to viewing game information, users can also set the game time for the next game, and configure different central hub debug variables. The source code for the lazerWin Windows application can be found at:

S:\Engineering\Teams\Team11\Notebooks\Central Hub Notebook\Design Work\System Software\Central Hub Windows Application
4.4.4 Receiving and Transmission of Information

In order to establish valid communication that can be used throughout the different devices within the laser tag system, a data protocol was defined. In order to transfer the data necessary for proper messages and actions, the server will send and receive 8 bytes (64 bits) per message. In order to effectively send and receive the data, the data will be encoded into a two 32-bit command words. The first command word will be considered the command value, containing information about the sender’s device type, player number, and a command. The details of this word can be seen in Table 5. The second word will contain data. This value will not always be utilized for every command. When it is not utilized, its value is to be set to all zeros. The details of this word can be seen in Table 6.

As stated previously, the first command word is composed of several fields encoded within the value. The first of these, the expansion space, was provided to allow spare space in case more data would be needed. For the purposes of the prototype, this value will be always set to all zeroes, as it is not used.

The second field, referred to as the device type, allows the receiving device to verify the sender’s type of device. This allows a level of security, preventing the sensor vest or laser gun from executing administrative commands. Valid messages that will be accepted can be seen in Table 7.
The third field encoded in the value is the player Identification value. Currently, the highest two bits are used to distinguish the team that the player is currently on, unless the game type does not require teams. If the system were to be put into production, the team data width could possibly be expanded in order to support more teams. For instance, if the team field width were to increase to 3 bits, the system could support 8 different teams, all having 128 players. This arrangement can be seen in Table 8.

### 4.4.4.1 Android Application Commands

The commands found in Table 9 are able to be used when the command set identifier value is 0x1. This allows an android application to execute these commands at any point within the game. These commands are detailed below.
Table 9 - Android Application Commands

<table>
<thead>
<tr>
<th>Value (8 bits)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Acknowledge</td>
</tr>
<tr>
<td>0x01</td>
<td>Get Number of Players</td>
</tr>
<tr>
<td>0x02</td>
<td>Get Remaining Game Time</td>
</tr>
<tr>
<td>0x03</td>
<td>Get Team Score</td>
</tr>
<tr>
<td>0x04</td>
<td>Get Player Statistics</td>
</tr>
</tbody>
</table>

4.4.4.1.1  **Acknowledge()**
This command, when received, sends an acknowledgement signal back to the Android application in order to confirm the correct communication between the mobile application and the server.

4.4.4.1.2  **GetNumberOfPlayers()**
This command, when received, accesses the number of sensor vests and laser guns that have been initialized within the system. This information is then sent back to the Android application.

4.4.4.1.3  **GetRemainingGameTime()**
When received, the central hub accesses the remaining amount of game time left, if the game is in session. This information is then transmitted back to the Android application. The data value passed represents the number of remaining seconds.

4.4.4.1.4  **GetTeamScore()**
This command, when received, accesses the team identifier that was sent within the player identifier, and accesses the correct score for that team. That information is then relayed back to the Android application.

4.4.4.1.5  **GetPlayerStatistics()**
This command accesses the game statistics for the player number whose identification is passed in the player ID field. This information, including shots, hits, and ammunition, is encoded into the data value according to Table 10. This data value is then sent back to the android application.
4.4.4.2 Laser Gun Commands

The commands found in Table 11 are able to be used when the command set identifier value is 0x2. This allows a laser gun to execute these commands at any point within the game. These commands are detailed below.

Table 11 - Laser Gun Commands

<table>
<thead>
<tr>
<th>Value (8 bits)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Acknowledge</td>
</tr>
<tr>
<td>0x01</td>
<td>Add Player</td>
</tr>
<tr>
<td>0x02</td>
<td>Set Shots Fired</td>
</tr>
<tr>
<td>0x03</td>
<td>Set Hit Count</td>
</tr>
<tr>
<td>0x04</td>
<td>Set Ammunition Count</td>
</tr>
</tbody>
</table>

4.4.4.2.1 Acknowledge()

This command, when received, sends an acknowledgement signal back to the Laser Gun in order to confirm the correct communication between the device and the server.

4.4.4.2.2 AddPlayer()

When received, this command increments the number of players currently initialized within the game. This allows other devices, when needed, to request an accurate count of active game players.
4.4.4.2.3  *Set Shots Fired()*

When received, this command identifies the player’s identification using the playerID field, and then sets the player’s shots fired statistic to the appropriate value. This value is passed to the central hub using the data word.

4.4.4.2.4  *Set Hit Count()*

When received, this command identifies the player’s identification using the playerID field, and then sets the player’s hit count statistic to the appropriate value. This value is passed to the central hub using the data word.

4.4.4.2.5  *Set Ammunition Count()*

When received, this command identifies the player’s identification using the playerID field, and then sets the player’s current ammunition statistic to the appropriate value. This value is passed to the central hub using the data word.

4.4.4.3  **Base Station Commands**

The commands found in Table 12 are able to be used when the command set identifier value is 0x4. This allows a Windows application or Android application to execute these administrative commands at any point within the game. These commands are detailed below.

**Table 12 - Server and Base Station Commands**

<table>
<thead>
<tr>
<th>Value (8 bits)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Acknowledge</td>
</tr>
<tr>
<td>0x01</td>
<td>Set Global Debug</td>
</tr>
<tr>
<td>0x02</td>
<td>Seg UDP Debug</td>
</tr>
<tr>
<td>0x03</td>
<td>Set I2C Debug</td>
</tr>
<tr>
<td>0x04</td>
<td>Unused</td>
</tr>
<tr>
<td>0x05</td>
<td>Set Game Duration</td>
</tr>
<tr>
<td>0x06</td>
<td>Set Game Mode</td>
</tr>
<tr>
<td>0x07</td>
<td>Clear Previous Game</td>
</tr>
<tr>
<td>0x08</td>
<td>Start Game</td>
</tr>
<tr>
<td>0x09</td>
<td>End Game</td>
</tr>
</tbody>
</table>

4.4.4.3.1  **Acknowledge()**

This command, when received, sends an acknowledgement signal back to the Windows Application in order to confirm the correct communication between the device and the server.
4.4.4.3.2  *SetGlobalDebug()*
This command, when received, turns sets all of the global variables to true if the data word is equal to one, otherwise it will set all of the global variables to false.

4.4.4.3.3  *SetUDPDebug()*
This command, when received, sets the UDP debug variable to true if the data word is equal to one, otherwise it will be set to false.

4.4.4.3.4  *SetI2CDebug()*
This command, when received, sets the I2C debug variable to true if the data word is equal to one, otherwise it will be set to false.

4.4.4.3.5  *SetGameDuration()*
This command will set the game duration to the amount of seconds that is passed using the data word. This command will take effect during the next game or will be ignored if a game is currently in session.

4.4.4.3.6  *SetGameMode()*
This command will be used for future implementation. It will allow the Windows application to change the type of game that is being played. This will affect data scoring options as well as team arragement. The desired type of game will be defined and passed using the data word.

4.4.4.3.7  *ClearPreviousGame()*
This command, when received, will allow the central hub to clear all of the previous game’s statistics for the game, and for each individual player. This will have no effect if a game is currently in session.

4.4.4.3.8  *StartGame()*
This command, when received, allows the windows application to initiate the game and start the game’s timer, providing that it is not currently in the middle of a game.

4.4.4.3.9  *EndGame()*
This command, when received, allows the windows application to stop the game and stop the game’s timer, providing that it is currently in the middle of a game.
4.5 Budget

In order to effectively manage the construction of the central hub, a small budget was created in the initial stages of design. Initially, the central hub was allocated an amount of $128.00. After all of the components were purchased, a total of $125.84 was spent. Table 13, found below, outlines the costs of the different components of the central hub.

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Description</th>
<th>Cost per Unit</th>
<th>Quantity</th>
<th>Line Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>4PCB</td>
<td>Central Hub Lighting Control Board</td>
<td>$33.00</td>
<td>1</td>
<td>$33.00</td>
</tr>
<tr>
<td>CUI Inc</td>
<td>Power Supply 12V 5A</td>
<td>$26.98</td>
<td>1</td>
<td>$26.98</td>
</tr>
<tr>
<td>QSHPark</td>
<td>MAX6955 Breakout Board Manufacturing</td>
<td>$16.20</td>
<td>1</td>
<td>$16.20</td>
</tr>
<tr>
<td>Hammond Manufacturing</td>
<td>BOX ABS 9.84X7.09X3.98 GREY</td>
<td>$13.77</td>
<td>1</td>
<td>$13.77</td>
</tr>
<tr>
<td>Adafruit</td>
<td>White 7-segment clock display - 0.56&quot; digit height</td>
<td>$4.95</td>
<td>2</td>
<td>$9.90</td>
</tr>
<tr>
<td>NKK Switches</td>
<td>SWITCH ROCKER SPST 10A 125V</td>
<td>$6.31</td>
<td>1</td>
<td>$6.31</td>
</tr>
<tr>
<td>Texas Instruments</td>
<td>IC REG LDO 5V 3A DDPAK</td>
<td>$2.64</td>
<td>2</td>
<td>$6.28</td>
</tr>
<tr>
<td>Adafruit</td>
<td>Red 7-segment clock display - 56&quot; digit height</td>
<td>$3.95</td>
<td>1</td>
<td>$3.95</td>
</tr>
<tr>
<td>Lumex Inc</td>
<td>LED 10MM SUPER RED PANEL MOUNT</td>
<td>$1.86</td>
<td>2</td>
<td>$3.72</td>
</tr>
<tr>
<td>Lumex Inc</td>
<td>LED 10MM GRN PANEL MOUNT</td>
<td>$1.77</td>
<td>2</td>
<td>$3.54</td>
</tr>
<tr>
<td>CUI Inc</td>
<td>CONN JACK 2X5 5MM VERT POWER PNL</td>
<td>$2.29</td>
<td>1</td>
<td>$2.29</td>
</tr>
<tr>
<td>On Shore Technology Inc</td>
<td>CONN USB TYPE A R/A WHITE</td>
<td>$0.45</td>
<td>2</td>
<td>$0.90</td>
</tr>
</tbody>
</table>

The two most expensive components of this system were the manufacturing of the display control board and the power supply. The first was a necessary cost as the board necessary could not be manufactured on site as the equipment cannot obtain the necessary accuracy needed. The second of these costs, the power supply, would be considered a cost that is necessary for the prototype unit, but if the system would be manufactured on a large scale, the power rectified and regulated internal to the central hub. Because of this, an external power brick would not be necessary in that case.

5 Laser Gun Design

This section outlines the different mechanical, electrical, and software aspects of the laser gun, as well as communication with the sensor vest.
5.1 Electrical Hardware Design

The components of the laser gun needed to be contained in some sort of electronics casing. The casing for the prototype was not crucial in proving the feasibility and functionality of the system. This casing could be designed and built specifically for our system, but due to convenience and project budget constraints the Lazer-Ops chose to purchase and re-purpose an existing casing. The team purchased and modified water guns to use as casings for the laser gun. The production model would have a casing specifically designed for this system in order to provide component mounts creating a more organized and trustworthy product. The water gun was made of plastic, making it a lightweight and inexpensive. The inside was stripped of all unnecessary plastic pieces. The team decided that the majority of circuit components should reside in the laser gun as its casing gives them more protection than the cloth of the vest. The laser gun must hold the main components including the gun and vest power source, two status LEDs, two buttons, and on/off rocker switch, a speaker, focusing lens, IR LED, various resistors, a microcontroller and Wi-Fi module, a 5V regulator, and transistors and capacitors needed for the vest. A conceptual layout of this is shown in Figure 21. A 9V battery supplied enough power for the sensor vest LEDs which were the most power-consuming components putting out at least 40mW. A production model would have two 9V batteries in parallel in order to provide a longer-lasting system life and more user convenience. The laser gun features two attachments including a scope and smartphone holding device which are attached to the top of the laser gun. A cable composed of six wires exits the laser gun leading to the vest supplying the power (lines for 5V and 9V), a line for receiving IR signals, two LED supply lines, and a ground line. The prototype implemented 24 gauge wire with a max current of 3.5A\textsuperscript{x}. The production design would provide a protective tubing around this cable in order to prevent the product from damage were the cable to be pulled.

![Figure 21 - Block Diagram of Laser Gun](image)

43
The aim of the mechanical design is to provide the required functionality in a way that makes it clear to the user that this system is a game intended for entertainment purposes and not to actually cause or encourage harm to other people. The laser gun is plastic and brightly colored like typical children’s toys and does not resemble an actual weapon. The team purchased a plastic laser gun casing and altered it to be compatible with the Lazer-Ops system. The team drilled holes were made for the button inputs (on/off, reload) and status LED outputs and inserted a focusing lens and speaker. Additionally, a laser light scope was used to provide the user with some visible feedback pertaining to where they are aiming. This is discussed in section 5.1.7. The diameter was also a size which could be incorporated into the barrel of the water gun being used to house the electronics.

When a player depresses the trigger of the laser gun, the laser sight displays a red dot near where the IR beam would be firing were it to be visible light.

5.1.1 Controller

Initially, a PIC microchip was going to be used to control the functions of the laser gun. Several free samples were acquired and a PIC programmer were purchased. Initial software testing was done using an Arduino microcontroller board because the team was already familiar with the board. An Arduino Uno board was used. After discussing the product’s microcontroller further, it was determined that using an Arduino microcontroller for the system prototype would be satisfactory to prove the ability and feasibility of the project to go into production with further funding and time. However, were the Lazer-Ops system be developed further into a production model, For comparison, the PIC 16F684, common in many consumer electronics products, has 14 pins, PWM capabilities, 3.5KB of program memory, 128 Bytes of RAM, and 256 Bytes of EEPROM. This could be used in the production model and perhaps provide some benefits over using an Arduino. It is about 0.8 x 0.3” which would take up much less physical space and potentially cost less when ordered in mass quantities. (Microchip)

5.1.1.1 Gun to Vest Communication

Initially, the team considered using Bluetooth or Wi-Fi to communicate between the vest and laser gun. Shortly after, it was decided that this was possible but out of the scope of the project. Bluetooth tends to have shorter range while the range of wireless networks can more than double this range, for example extension from 10 to 35 meters indoors. The table below shows several other features of Bluetooth and Wi-Fi communication. The final decision was to wire the laser gun and vest together, as in the standard laser tag system. However, the production model would likely implement wireless communication over Wi-Fi due to the wired connection, it was also decided that one microcontroller provided enough input and output pins to be used for both the laser gun and the vest. A comparison can be found in Table 14.
An Arduino microcontroller was chosen using the decision matrix below to be the microcontroller for the prototype system. A PIC microcontroller was also considered as a potential microcontroller for the system. While it might better suit a production model, due to its lower cost and smaller size, it was not ideal for prototyping the system. It was used to sense user inputs and control the resulting actions. Specifically, the Lazer-Ops system uses the Arduino Uno board which is based on the ATmega328. It has fourteen output/input pins which operate at 5V, all of which were used in our prototype. It also has six analog input pins which were not used. Each pin can safely supply up to 40mA of DC current. This is enough current for all function except the IR LED which is rated for 100mA which is discussed further in section 5.1.6. The Arduino has input voltage limits of 7-20V with a recommended input of 7-12V. The Lazer-Ops system is powered by a 9V battery because it is easily accessible and is much lighter weight and a smaller volume than using AA batteries to provide an equivalent voltage. The ATmega328 has 32KB of flash memory, 2KB of SRAM, and 1KB of EEPROM. This more than doubles the memory in the PIC 16F684 offers. It operates with a clock speed of 16 MHz. Physically, the Arduino is 2.7x2.1”. A decision matrix for the laser gun microcontroller is shown below in Table 15.

Table 15 - Decision Matrix for Laser Gun Microcontroller

<table>
<thead>
<tr>
<th>Specification</th>
<th>PIC</th>
<th>Arduino</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td># of I/O Pins</td>
<td>8</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>Experience</td>
<td>0</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>PWM capabilities</td>
<td>10</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Memory</td>
<td>5</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>Size</td>
<td>10</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Cost</td>
<td>10</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>143</td>
<td>193</td>
<td></td>
</tr>
</tbody>
</table>
Figure 22 - Laser Gun Schematic
5.1.2 Reload Button

A hole was made in the sidewall of the plastic laser gun casing and secured using hot glue as seen in the Figure 23 below. A four-pin push button was inserted to provide the user with the ability to reload their ammunition during game play. The team could have used any normally open push button, but we had several 4-pin push buttons in our possession which were ideal for the prototype. The simple circuit for connecting a button with the Arduino is shown below in Figure 24.

Figure 23 - Implementation of the Reload Button

Figure 24 - Sample Push Button with Arduino Schematic
5.1.3 Trigger

The water gun which was purchased implemented a plastic mechanical trigger for its purposes. The Lazer-Ops system uses this mechanical trigger to depress a two-pin mountable push button which sets its corresponding Arduino pin high. The team constructed an aluminum button holder to hold the button in place behind the trigger. The team screwed this holder into place and hot glued it for extra security and consistency with continuous use. The team wired all buttons using the same schematic as the reload button. The mechanical trigger design can be seen in Figure 25.

![Figure 25 - Trigger Button Mechanical Design](image)

5.1.4 On/Off Rocker Switch

The team made a hole in the side wall of the plastic laser gun casing and inserted a three-pin illuminated rocker switch to provide the laser gun’s on/off capability. When switched on, a built-in status LED is illuminated and the power supply is connected via a 2.1mm center-positive plug to the Arduino board’s power jack. The large size and built in LED of the rocker switch was ideal for the on/off input as it provided convenience to the user. The laser gun implemented the switch as seen in Figure 26.
5.1.5 Speaker

To provide noise feedback upon trigger fire, a piezo buzzer is supplied voltage via a transistor which turns on when the trigger is pressed. The buzzer emits a 3.8kHz sound when supplied 5V by the Arduino. This frequency of sound is just above that of voice frequency making it distinguishable and able to provide useful feedback yet not harmful to the human ear. The buzzer is circular and (24mm diameter) and is fixed on the inside wall of the laser gun casing.

5.1.6 Infrared LED

The gun uses an infrared beam to shoot other players. This spectrum of light is just above that visible to the human eye. Infrared wavelength is typically 870nm and 950nm. The laser gun in our system uses a 940nm infrared LED with a 45 degree angle of half intensity (Part number 276-143). The most common user protocol for consumer electronics, NEC, uses a standard 38 kHz for communication (http://techdocs.altium.com/display/ADRR/NEC+Infrared+Transmission+Protocol). As typical for short range communication, a plastic lens, discussed further in section 5.1.7, focuses an infrared LED into a narrow collimated beam. This beam is modulated (switched on and off) to encode the data. Each laser gun’s beam is pulse-width modulated (PWM) such that each player has a unique signal that can be received and decoded by the IR receivers in players’ vests. This way it can be known which specific player has hit another player. Pins 3, 5, 6, 9, 10, and 11 on the Arduino are capable of PWM output. The IR Library requires that Pin 3 be used for sending the IR beam (http://forum.arduino.cc/index.php?topic=49630.0). Thus, the Lazer-Ops systems’ laser guns’ IR LEDs are connected to pin 3.
5.1.7 Lens
A lens was used to focus the IR LED by collimating the beam and extending the range of the infrared signal. The lens had a 12.5mm diameter and a focal length of 173 (Part number AX27429). The team chose this particular lens as it had the longest comparative focal length from the other available options which gives it a longer range. A PVC pipe held the IR LED in place as suggested by Miles Tag Systems (http://www.lasertagparts.com/mtoptics.htm). The required range of the prototype was 5 ft. The prototype produced successful IR transmission and reception at a maximum of 30ft.

5.1.8 Laser Sight
The laser gun fires an infrared beam invisible to the human eye. The purpose of the touch-activated laser sight is to give the player an idea of where their infrared beam is being fired. When the player rests their finger on the laser gun trigger and places their hand on the gun handle, a red dot from the laser diode is displayed indicating an approximate target area.

The laser sight is comprised of components including an N-channel FET, 9 Volt Power Supply (Battery), and red laser diode. The laser diode creates a red (650 nm) beam of light seen as a dot. This dot is focused using a lens so that it remains accurate and clear even at a distance. This lens is built into the laser module which is twist-adjustable (as in Figure 27 below). The particular laser module used is very low power (5 mW) which is typical for handheld laser pointers.

![Laser Diode](image)

The field effect transistor (FET) acts as an on/off switch for the laser sight. The main current flows through the gate of the transistor and the FET is switched on by the presence of a voltage potential at its gate. When this potential is above the threshold voltage (2V) the laser lights. When the transistor is turned on it has a low resistance (about 1.2Ohm). However, when the transistor is turned off, there is a small leakage current around 1uA. When the player pulls the trigger, the two contacts connect and this current produces a voltage above the threshold causing the laser to light upon trigger pull. Figure 28 below shows the circuit for the laser sight.
5.1.8.1 Implementation

The laser sight components were placed on vector board and soldered together. The battery, laser diode, and touch contacts were attached to this. The touch contacts were connected to a two-way terminal block from which additional leads continued on to the locations where the player would make contact, the laser gun handle and trigger. The laser sight was then placed in the laser gun apparatus. A hole was made for the laser diode which was initially taped into its desired position. The battery was also taped down in a convenient location. Figure 29 and Figure 30 show the preliminary implementation of the laser sight. The laser sight was glued into place in the final prototype. It was incorporated into the existing circuitry and received its power from the existing 9V battery. Figure 31 shows the final implementation.
A clasp was milled out of aluminum to attach an aluminum tube to the top of the laser gun acting as a scope. The team used aluminum as it was readily available and able to be easily cut into the shape desired for the purposes of holding the laser diode in place. This scope also holds the laser diode in position. The scope is firmly attached to the gun on one end and adjustable on the other so that the red dot can be aligned with the location the infrared beam is being fired. The piece of aluminum used was 1cm in thickness. This precise thickness is was not required but was determined by what was readily available to the team. The clasp design can be seen in Figure 32 and Figure 33 below.
5.1.9 WiFly

Initially, the team chose to implement a standalone MRF24WB0MA Wi-Fi chip with the necessary components--forming to a custom Wi-Fi Shield. The circuit below (Figure 34) was constructed on a breadboard for testing with the central hub.
The tests results from this application proved to be unsuccessful, as the device could not get be viewed on the team’s personal WLAN. There were also no results from a ping test of the programmed IP address on the network.

The circuit, as well as the software, went through several hours of testing (incorporating different Arduino Wi-Fi libraries, e.g.). Due to projected schedule constraints, the team ordered two replacement devices which incorporated the same MRF24WB0MA chip with a daughter board (See Figure 35 below).
Despite the support found online for integrating this chip with an Arduino Uno, the team was unable to interface the devices successfully.

The MRF24WB0MA chip has Wireless-N capabilities, which increases the networking bandwidth. This is the most recent 802.11 version, and it increases the stability of signal transmission through physical barriers. Although Wireless-N capabilities is a desired quality of Team Lazer-Ops’ final product launch, it was not considered a design requirement for our final prototype. As a result, the search was expanded to include devices without Wireless-N capabilities, but there was much open-source information for interfacing with an Arduino Uno.

The team eventually settled on using a WiFly shield for vest-HUB communication. A SparkFun WiFly shield was made available by the staff at Calvin College, and it was able to successfully broadcast after utilizing the Arduino WiFly library supported by SparkFun. Immediately after the WiFly was interfaced with the Arduino, a new device was made visible on the WLAN host page. The WiFly test parameters were programmed through the SPI terminal on the Arduino. After configuration, the WiFly maintains that configuration until a factory reset is performed. Various parameters were set for both WiFlys in the prototype using a wifly_config sketch in the Arduino environment.

```cpp
runCommand("set ip dhcp 1",true);
This command enables dynamic host configuration protocol for automatic IP assignment by the router.

runCommand("set wlan ssid LazerOps",true);
This command sets the name of the network to connect to.
```
runCommand("set wlan phrase 0",true); //set password
This command ensures that the WiFly has no password in memory to join the network, as
the network is open.

runCommand("set ip host 192.168.1.200",true); //set the host IP address.
This command sets the address of the host of the Wi-Fi network. This is the address of the
Pi.

runCommand("set ip localport 2000",true); //set the listen port
This command sets the port to listen on. This is the local port of the Pi.

runCommand("set ip remote 5000",true); //set the send port
This command sets the sending port of the WiFly.

runCommand("set ip protocol 1",true); //set the TCP and UDP
This command sets the protocol to UDP.

runCommand("set sys sleep 0",true);
This command disables the sleep timer of the WiFly.

runCommand("set comm timer 429496795",true);
This command sets the system communication timer to the longest possible value.

runCommand("set sys wake 0",true);
This command disables the wake timer.

runCommand("set comm match 0",true);
This command disables the use of a command terminator for any command written over SPI
to the WiFly.

runCommand("set comm time 0",true);
This command disables the interval after a byte is received to begin forwarding data.

runCommand("set comm size 8",true);
This command standardizes the size of each packet sent (in bytes).

runCommand("set broadcast interval 0",true);
This command disables the interval for sending UDP broadcast messages.

runCommand("set wlan join 1",true);
This command enables the automatic connection of the WiFly to the programmed network
configuration upon power-up.
5.2 Software Design

The Arduino language is based on C/C++ (http://arduino.cc/en/Reference/HomePage) and the Arduino IDE was used to program the microcontroller. The software for the laser gun/vest was written in four phases. The first phase implemented the functionality of the three button inputs, their corresponding LEDs, and game sound. That is, the on/off, reload, and trigger functions. (Trigger meaning the reception of a button press and LED flash and sound). The second phase implemented the infrared functionality (i.e. sending unique infrared patterns for each player). The third phase of software implemented the vest’s functions and communication with the central hub (i.e. team LEDs, a “player hit” sequence, inability to fire when hit, and Wi-Fi communication).

5.2.1 Software Phases

The following sections describe the progression of the software development in creating the final software of the sensor vest and laser gun prototypes.

5.2.1.1 Phase One: Button Implementation Software

Three buttons were used to provide the on/off, trigger, and reload inputs from the user. If a button is pressed, the Arduino pin connected to that button is connected to the 5V pin on the Arduino (see Figure 22). The software has a central loop that checks if these pins are high (in other words, if there has been a button pressed by the user). The software provides button-debouncing. If a button has been pressed, the Arduino software initializes the corresponding reaction. If the on/off button is pressed a status LED lights and stays lit until the button is pressed again. The reload button is only operational if the system is turned on. When the reload button is pressed the reload status LED comes on. This light stays on for 3 seconds using software delays in order to simulate the time needed to physically reload a gun. The software only allows for an operational trigger button if the system has been turned on and the player has ammunition. If these conditions are met, the player can fire which also causes a speaker to sound and status LED to blink upon each shot. This completed the software dealing with the system inputs.
5.2.1.2 Phase Two: Initial Infrared Software

The patterns received by the sensor vest from the laser gun (up to 10) were decoded and stored in the EEPROM of the Arduino. The serial monitor was used to change code memory. Code memory is where in EEPROM the codes were being stored and taken from when the gun receives an IR signal or fires an IR signal. Each IR pattern contains 36 data pairs which correspond to the time (in microseconds) that the pin is set high and low, respectively (See Figure 36). The developer enters an “s” command in the serial monitor to fire. Each gun holds 10 different IR patterns in the EEPROM of the ATMEGA328 which is 1KB. These patterns remain in memory independent of whether power is supplied or not. Depending on the time a player turns on their gun, the base station assigns that gun/vest a player number. This player number corresponds to which IR pattern will be taken from the EEPROM and sent by means of the pulsating IR LED. In a production model, where more than 10 players are possible, other IR patterns could easily be programmed into EEPROM to easily accommodate these additional players.

```c
// ON, OFF (in 10's of microseconds)
880, 432,
80, 52,
60, 54,
58, 54,
58, 54,
58, 54,
58, 54,
60, 54,
58, 160,
60, 162,
58, 162,
58, 162,
58, 162,
58, 162,
58, 162,
58, 162,
58, 162,
58, 162,
58, 162,
58, 162,
60, 160,
60, 160,
60, 54,
58, 162,
58, 162,
58, 162,
58, 162,
58, 162,
58, 162,
880, 218,
58, 0;
```

Figure 36 - Sample IR Pattern read from Arduino EEPROM
5.2.1.3 Phase Three: Sensor Vest Software

The focus of phase three software was to integrate Wi-Fi communication, LED lighting control, trigger control, reception control. Figure 38 below is a state diagram of the Arduino code. Using this structure, the components were tested on our initial breadboard. The green blocks represent system condition checks, the blue blocks are the results of those conditions, and the grey blocks are the physical outputs of the Arduino.

Figure 37 - Laser Gun Software Flow Diagram

5.2.1.4 Phase Four: Merging Previous Phases.

The software was finally merged into one software file which is called an Arduino ‘sketch’. If an input was given repeatedly (i.e. reload, fire), the software could get stuck in a loop where it cease to recognize that the player had been shot because the software would never make it into the gotShot() loop. These timing issues were exposed and neutralized with various function calls in the form of a series of software interrupts within the main loop. This sketch was compiled and we implemented it on a breadboard in order to sort through these timing issues.

5.2.2 Testing

The laser sight was constructed and the contacts were touched resulting in a successful red dot sight. It was discovered that if two people touched each other and each person held onto an end of the touch contacts, the laser diode would also light. However, after doing so, the sight would no longer work. The cause of this was determined to be the doubled resistance from two bodies as opposed to one. It caused a voltage too high above the threshold voltage to fry the transistor. The sight was rebuilt several times due to transistors that had been fried in such a manner. The laser diode was also surrounded in heat shrink to avoid shorting with the aluminum clasp.
Once the sight was working again, the built in lens on the laser module was used to focus the beam in order to provide a focused red dot at a distance. Once the sight was fixed into position in the laser gun, it was adjusted until it displayed a red dot in a location that corresponded fairly closely to the area where the IR beam was being fired.

Initial testing used an IR LED with a half angle viewing angle of 45 degrees. This signal was successfully received by the IR receiver in the vest, however, only at a short range of about 3 feet. Several used television remotes were purchased and the IR LEDs were extracted and used for further testing to extend the range of IR reception. Their range of IR reception was tested without the use of a lens. The IR LED with the furthest range was chosen. This LED had a range of about 1ft without a lens. The IR beam’s short range of accuracy was surprising and it was found that the LED was rated for much higher current than what the Arduino was supplying it. It was rated for 100mA, but the Arduino output pins supply up to 40mA. The LED also has a forward voltage of 1.28V. Using this information, a power calculation was performed the resistor value was correspondingly decreased from 330 to 4.7 Ohm. This change dramatically increased the IR LED’s transmission and reception range to its maximum of 30ft.

5.2.3 Requirements
The gun meets the requirements determined to be in the scope of the project. These aim to make a product that takes user-friendliness into account. A button in used for the reload function, a trigger to fire, and a rocker switch for the on/off function. In order to indicate to the users that the system is working properly, the laser gun makes use of status LEDs. There are three which indicate when the trigger has been pulled, when the reload button has been pushed, and when the system is turned on and ready for play. Additionally, there is sound feedback upon each trigger fire. The system software eliminates the ability for a player to rapid fire and will automatically reload after a player has been hit and “respawns”. The software will also automatically reload a player’s ammunition when they run out. The reload process uses an intentional software delay to halt the player’s ability to fire for a set time to simulate a more realistic reload situation. The player remains able to be hit during this delay. The software is easily expandable to provide for extended game options. For example, it keeps track of how many times a player has reloaded their ammunition in the event that a future version of the game may want to limit the number of times a player may reload. It similarly could allow player’s a certain number of lives and eliminate them from the game when they have no lives remaining.

The primary requirements of the Vest were to provide adequate lighting and IR reception. These requirements were met by using vinyl pockets to store the PCBs on the vest. Additionally, the vest was to somewhat resemble the final production model, in terms of board and sensor placement. The final vest prototype meets this requirement also, as there are front, back, and shoulder sensor boards successfully integrated with the vest material.

Because the power supply and main circuitry is found in the gun, only 6 signals were sent to the vest. The vest itself has vinyl pockets so each PCB is supported, while it also allows for IR reception and LED lighting for other players. Much thanks is extended to Douglas Herder for the construction of the vests and pockets attached.
5.2.4 Budget

Table 16 shows the finances spent on the implementation of the laser gun. Not all of these components were used in the final prototype of the laser gun as some of the purchases were for parts that were later deemed to be out of the scope of the project. Additionally, this list does not include parts that were donated or already available for the team’s use. Some parts were also ordered in higher quantities than necessary due to ordering convenience and redundancy in the case of a last minute part failure.

<table>
<thead>
<tr>
<th>Laser Gun Budget</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmer</td>
<td>$22.40</td>
</tr>
<tr>
<td>IR LEDS</td>
<td>$6.28</td>
</tr>
<tr>
<td>Arduino</td>
<td>$13.85</td>
</tr>
<tr>
<td>Speakers</td>
<td>$5.50</td>
</tr>
<tr>
<td>Laser pointer</td>
<td>$3.44</td>
</tr>
<tr>
<td>2N7000 FET</td>
<td>$0.99</td>
</tr>
<tr>
<td>Squirt guns (Walmart)</td>
<td>$5.30</td>
</tr>
<tr>
<td>TV remotes</td>
<td>$4.24</td>
</tr>
<tr>
<td>Voltage regulator</td>
<td>$0.95</td>
</tr>
<tr>
<td>Terminal blocks</td>
<td>$2.95</td>
</tr>
<tr>
<td>Illuminated rocker switch</td>
<td>$3.12</td>
</tr>
<tr>
<td>Arduino Prototyping shield</td>
<td>$12.00</td>
</tr>
<tr>
<td>Piezo Buzzer</td>
<td>$2.62</td>
</tr>
<tr>
<td>2N7000 FET</td>
<td>$1.56</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$84.10</strong></td>
</tr>
<tr>
<td><strong>Allocated Budget</strong></td>
<td><strong>$108.96</strong></td>
</tr>
</tbody>
</table>

5.2.5 Production Model

The production model would incorporate more advanced feedback to the players during the game. This would be done with more sounds signifying to the player various game situations, such as reload, player hit by fire, player with a successful hit, low ammunition, game ending soon, game end, and a game won. The production model would have a more rigid electrical gun casing. It would use Wi-Fi communication rather than being wired to the vest, and have a method for attaching the players’ phones to the laser gun as a HUD. It would use a custom designed casing and use a PIC microchip rather than an Arduino board.

In the production model, the laser gun casing would be designed with the system specifications in mind so that it would be more economical by not using unneeded space and material and to save on cost. Also, the system components could be placed more purposefully and more securely to provide an extended system life. A production model would have significant thought put into the effects of the realism of the system. It would aim to create a system that encouraged all aspects of the product to encourage the positive team goals and discourage product misuse.
6 Sensor Vest Design
The following sections outline the different aspects of the sensor vests design and prototype.

6.1 Electrical Hardware Design
This section outlines the electrical implementation of the Sensor Vest.

6.1.1 Sensors
Visible wavelengths generally reach a maximum of 750 nm (red), which signals the start of infrared light. Therefore, most photodiodes filter anything below 800 nm, but some poorer-quality components (or design-specific components) only filter below 700 or 750 nm wavelengths. Since the team desires full operability even in lighted areas, which contain some high-frequency infrared light, there was a strict filtration limit placed on this component. Additionally, the photodiode should be exposed enough that the vest can be targeted from an angle that is not directly “above” the photodiode. If the IR receiver is too concealed within its case, the angle of reception will be greatly reduced. The initial prototype was constructed on a breadboard, which simulated LED lighting, Wi-Fi communication with the Pi, and IR decoding. Figure 38 is a picture of our breadboard prototype.

Figure 38 - Sensor Vest Prototype
Because the multiple boards with the same receiving circuit were needed, designing and creating our own PCBs was the best option due to the ease in repeatability of the design (LED spacing, centering of IR receivers). The board layout below (Figure 39) was used for construction of the PCBs.

![Figure 39 - Sensor Circuit Board](image)

These PCB’s were traced on a vest prototype provided by Douglas Herder. After tracing, pockets were added for storing the PCBs. The initial vest and final vest prototypes can be found in Figure 40, Figure 41, and Figure 42 below, respectively. The production model would incorporate a hard shell enclosure for each PCB to ensure a safe, reliable, and trustworthy product.

![Figure 40 - Sensor Vest Prototype](image)
Figure 41 - Sensor Vest Prototype

Figure 42 - Sensor Vest Prototype
6.2  Budget

Table 17 shows the laser vest expenditures. The budget was exactly 71 cents over.

<table>
<thead>
<tr>
<th>Purchased?</th>
<th>Manufacturer</th>
<th>Part Number</th>
<th>Vendor</th>
<th>Vendor Part Number</th>
<th>Description</th>
<th>Cost per Unit</th>
<th>Quantity</th>
<th>Line Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>-</td>
<td>-</td>
<td>Sparkfun</td>
<td>VRL-09954</td>
<td>WiFi shield</td>
<td>$69.95</td>
<td>1</td>
<td>$69.95</td>
</tr>
<tr>
<td>Yes</td>
<td>Microchip</td>
<td>MRF24WB0M</td>
<td>Mouser</td>
<td>MRF24WB0M</td>
<td>579-</td>
<td>$28.49</td>
<td>1</td>
<td>$28.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MRF24WB0M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Standalone Wifi Transceiver)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>Vishay</td>
<td>TSOP382</td>
<td>Mouser</td>
<td>TSOP382</td>
<td>IR sensors</td>
<td>$1.35</td>
<td>2</td>
<td>$2.70</td>
</tr>
<tr>
<td>Yes</td>
<td>Texas Instruments</td>
<td>TIP32A</td>
<td>RadioShack</td>
<td>2762017</td>
<td>TIP31 Transistor</td>
<td>$1.99</td>
<td>3</td>
<td>$5.97</td>
</tr>
</tbody>
</table>

|            |              |              |         |                    | Sensor Vest Total |          |          | $124.71    |
|            |              |              |         |                    | Preliminary Budget |          |          | $124.71    |

7  Mobile Android Application

The following subsections describe the requirements and design aspects of the Android application.

7.1  System Requirements

The main requirement for the mobile application was to be able to communicate with the central hub. To accomplish this, the mobile application will send UDP packets to and receive UDP packets from the central hub. The goal, and required outcome, of this communication was for the mobile application to be able to request, receive and decode the UDP packets containing the current statistics of the players using the Lazer-Ops laser tag system. Another requirement for the application was that it needed to be simple enough so that any person would be able to operate it with success.

7.2  Software Design

The sections that follow describe various aspects of the mobile application design process. The design team desired to create an application that would be easy enough for any person to use. The design team also wanted the players to be able to use the application as a way to enhance the Lazer-Ops experience. This enhancement would, in turn, make the Lazer-Ops system more modern and immersive than many laser tag products currently available. The main function of the application, with regards to the laser tag system, is to provide in-game statistics to players. The design team decided to develop the application on Google’s Android mobile platform rather than on Apple Inc’s iOS or Microsoft’s Windows Mobile. This decision is outlined below in section 7.2.3.
If brought to a production level product, the design team would not focus on one single operating system. The design team would instead use a cross-platform development suite to create the mobile application. This would allow simultaneous development and release multiple platforms. In turn, this would allow the Lazer-Ops application to reach even more of the market.

7.2.1 Functionality

The main function of the mobile application will be to display in-game statistics for all players. These stats will be viewable in two ways, by player or by team. The first way that application users will be able to view statistics is by team. When viewing team statistics on the Lazer-Ops prototype, the user will only be able to view the overall score of the team selected. The production scale model of the application will include total score as well as total shots fired, total shots hit, overall shooting accuracy, game time remaining, and fire rate. The second way that application users will be able to view statistics is by individual player. The user of the application will be able to see the statistics of any player playing a game on the central hub to which the smartphone is connected. Score, shots fired, shots hit, and shooting accuracy are the statistics that the application user will be able to view.

The statistics will be received by communicating, over Wi-Fi, with the central hub. In order to distinguish between multiple hubs and devices the application user will need to know the specific central hub that they wish to connect to. The user will determine which central hub to connect to by connecting the desired central hub’s Wi-Fi network through their phone’s settings menu. As the application is used, it will send request signals to the central hub through the Wi-Fi connection between the two systems. The central hub will process this request and report the relevant data to the application. This communication will be non-blocking. This means that the central hub will not only wait for the application to send a message, but instead will periodically check for messages. Table 18 below lists the commands that will be sent to the central hub.

<table>
<thead>
<tr>
<th>Message Name</th>
<th>Description</th>
<th>Device Expansion</th>
<th>Device Type</th>
<th>Player ID Expansion</th>
<th>Player ID Type</th>
<th>Command ID Expansion</th>
<th>Command ID Type</th>
<th>Command Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player Number</td>
<td>Number of players in the current game.</td>
<td>0x00000</td>
<td>0b00010</td>
<td>0b000000000000</td>
<td>0b0001</td>
<td>0b000000001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Player Statistics</td>
<td>Time remaining in the connected game.</td>
<td>0x00000</td>
<td>0b00010</td>
<td>0b000000000000</td>
<td>0b0001</td>
<td>0b000000010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green Team Statistics</td>
<td>Statistics for the green Team.</td>
<td>0x00000</td>
<td>0b00010</td>
<td>0b000000000000</td>
<td>0b0001</td>
<td>0b000000111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yellow Team Statistics</td>
<td>Statistics for the yellow Team.</td>
<td>0x00000</td>
<td>0b00010</td>
<td>0b010000000000</td>
<td>0b0001</td>
<td>0b000000111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Player Statistics</td>
<td>Statistics for the selected player.</td>
<td>0x00000</td>
<td>0b00010</td>
<td>0b000000000000</td>
<td>0b0001</td>
<td>0b000000100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The player statistics command will vary depending on which player option is selected. As shown in the table, the separate teams are requested by one command with two distinct Player ID values. The player statistic command values vary from the team statistic command values because the number of teams, for the prototype, is limited to two, while the player count does not have a limit. Because of the restriction on the number of teams, the hexadecimal commands to request the teams has been hardwired into the application. The team score requests are also a different command number than the player statistics command. This was done for the prototype because the design team had more than enough bits to use for commands. This choice also simplified the process of distinguishing between players and team statistic requests. For the production model, however, the design team would have built the team statistics commands into the “getPlayerCommand()” function. To merge these two functions together, the team score would be viewed as player zero for the specified “playerNumber” value shown in Figure 43 below.

```java
public int getPlayerCommand(int position) {
    if(DEBUG_MODE) Log.d(TAG, "Inside getPlayerCommand()");
    int team = position % 2;
    int playerNumber = position - 2;
    int playerID = (((int)team & 0x3) << 8);
    if(DEBUG_MODE) Log.d(TAG, "playerID without playerNumber is: " + playerID);
    playerID |= (int)playerNumber & 0xFF);
    if(DEBUG_MODE) Log.d(TAG, "playerID is: " + playerID);
    int command = (((int)0x80 & 0x1F) << 27); // Expansion space
    command |= (((int)0x82 & 0x1F) << 22); // Device type
    command |= (((int)playerID & 0x3FF) << 12); // Player ID
    command |= (int)0x1 << 8; // Command Identifier
    command |= (int)0x4 & 0xFF; // Command Number
    Log.d(TAG, "getPlayerCommand will return: " + command);
    return command;
}
```

Figure 43 - Android Application Get Player Command
7.2.2 Application Flow

Figure 46, above, illustrates the application flow. The first, and only true screen that users will see and use is the statistics activity screen. For the prototype, the statistics activity will show the green team’s score by default when the application is launched. Figure 7 shows the actual layout designed for the prototype. The user will be able to select which player or team’s statistics they want to view. This selection will be made through a spinner menu. A spinner menu is a small menu that only displays the currently selected item until it itself is selected using touch. The design team decided between using either a spinner or a tab system for this function. The thought behind this decision is shown below in section 6.2.3. The spinner items will be organized with team options at the top and player options following thereafter. The spinner items will also be readable names such as “Green Team” or “Player 1”. The spinner can be seen in its expanded form below in Figure 44. Once selected, the spinner menu will expand to show all of the items that it contains. The activity will send a request to the central hub for statistical data on the selected item after an option has been selected from the spinner. Once the activity is done receiving and deciphering the response packet from the central hub, it will display the data opposite its corresponding category.

![Statistics Activity Layout](image)

Figure 44 - Statistics Activity Layout
The statistics view will also allow the user to communicate with other users that are connected to the same central hub unit. This is accomplished through the text input shown at the bottom of the screen in Figure 45. To use these chat functions a user must first decide on a user name. This can be done by selecting the Set Username option in the options menu. The options menu, represented by 3 vertical dots, can be found in the upper left hand corner of the screen shown in Figure 45. Once the Set Username option is selected, a dialog prompting the user to input a username will be shown. After the user enters their username, the application will show a toast at the bottom of the screen displaying the user’s current username. The user will now be able to enter and send chat using the text editor at the bottom of the screen. The chat messages produced by this system will also be sent using UDP. This communication will not interfere with the UDP communication used by the central hub. This is possible because, the chat communication will operate on a different UDP socket port than what the central hub uses to communicate.

![Figure 45 - Application Layout without Expanded Text Editor](image)

If, at any time, the application user presses the home button, the user will return to the Android home screen and the application will enter a paused state. In this state, the application will be running at a minimum level in the background until it is either terminated by the Android operating system, or resumed by the user. When entering the paused state, the application will close its open communication sockets so that it will no longer receive any UDP packets while operating in the background. If it is resumed, the application will reopen its UDP sockets so that communication can continue.
Which mobile platform to develop the application on was the largest decision that the design team made. To choose the development platform, the team created a decision matrix. This decision matrix can be seen below in Table 19. The mobile platforms that the design team decided to investigate were Google’s Android, Apple’s iOS, and Windows’ Phone. These three represent the largest shares of the mobile operating system market (idc.com). Although Windows’ Phone only amounts for 3.0% of the fourth quarter market share for 2013, it had the largest market share growth of all the smartphone platforms (idc.com). The categories that the team deemed important in making this choice were cost of developing and publishing applications, the team’s prior experience working with the platforms, how easily applications can be developed for each platform, and how much variation there is in screen sizes for phones that use each platform. The team weighted previous exposure and experience developing with the platforms the highest of all the criteria. This was done because developing on a familiar platform would, hopefully, lead to a stronger application. Accessibility to development was weighted at a 7 because being able to program and test the application as easily and as often as possible was important to the design team. The next highest criteria was the cost of development and publication of the created application. Because this product is being created on a relatively tight budget a weight of 6 was given to the cost criteria. Finally, variation of model specs, mainly relating to screen size, was considered to be a potential hardship during development. This was included in the design matrix because varying screen sizes makes it harder to create cohesive and visually pleasing activity layout.
Using the above decision matrix, the design team decided that the Android mobile platform would be the best choice for this product. Although the Android platform scored lower than both of the other platforms in the variation of model specs, it scored as high as or higher than the other two platforms in the rest of the category. The Android platform scored an 8 in cost, because the only cost related to developing for Android was a one-time $25 registration fee with Google.

Apple’s iOS and Windows Phone, on the other hand, both have annual development costs of $99 (Apple Inc.) and $19 (Microsoft) respectively. Android scored a 9 in the prior experience category because one member of the design team has already created an application for Android prior to the start of this project. There is only one program, although there are two additional files that need to be downloaded as well, that needs to be downloaded and installed in order for any computer to be capable of developing applications for the Android platform, thus Android scored a 7 in the accessibility category. Similarly, Window’s Phone also requires the developer to have one program and a few files installed in order to develop. Apple Also only requires one program and a few files to be installed to develop for iOS. Using a Macintosh computer, however, is required to be able to download the required program. There are ways around this road block, such as more complex programs like Unity, which could be used, however this adds another level of complexity. Screen sizes and model specs for android devices have an incredible amount of variance, even within the same phone line. This amount of variance caused the design team to give the Android platform a 2. As a result of these ratings the Android platform received an overall score of 242, while iOS and Windows Mobile scored 205 and 197 respectively.
A smaller, but still significant, decision that the design team made was how users would choose the set of statistics to display. The two major contenders for this job were a spinner system or a tab system. Table 20 below shows the decision matrix that was used in making this decision. The two options that the team considered to fulfill this role were either a spinner system, or a tab system. The spinner system involves using expandable and collapsible menus for selecting items. The tab system would use multiple tabs that would each display their respective statistical data. The categories that were considered when making this decision were ease of use, layout size, and visual appeal. Both systems score extremely close to each other in every category. The spinner system edged out the tab system in the ease of use category only because all of the available options are easily access at one time when using a spinner. The spinner system also edged the tab system in the size category because the spinner system is more collapsible, only showing more than one item when the user desires to change what is selected. The tab system did outperform the spinner system in the visual appeal category. This was because spinners are simple, while tabs will take on the theme of the activity.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Weight</th>
<th>Spinner</th>
<th>Tabs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of Use</td>
<td>8</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Size</td>
<td>9</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Visual Appeal</td>
<td>4</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>156</strong></td>
<td><strong>147</strong></td>
</tr>
</tbody>
</table>

7.2.4 Software Updates

Software updates will be handled using the Google Play Store’s update system. These new software patches, or updates, will be published to the Google Play Store when finished. After these patches are published, the user will need to connect to an internet network and update their phone’s instance of the application. The updates that will be released will include such things as bug fixes; application performance enhancements, such as reduced application file size or faster algorithms; or functionality additions. These updates will not be released on any sort timeline, but will be released when they have been completed.

8 System Prototype: Lazer-Ops 1.0

The system prototype section describes the goals, purpose, and a description of the hardware and software components of this system. The chapter following this will be describing the production model of the system. This prototype will be delivered in May of 2014.

8.1 Purpose

The system will include two vests and two guns, a central hub, and an Android application. The players will initiate play using the central hub or the android application to begin a game of designated length. The laser guns will then actively communicate with the central hub in order to update various game statistics. These values will then be sent to the Android application when they are requested. The hub will output the team’s scores and remaining game time, while the Android application will display game statistics and status to users during play.
8.2 System
The hub will utilize an 802.11b Wi-Fi network to receive data from the laser gun, process it, and store it within the central hub. This information will then be interpreted to generate various game statistics, including player score, shots fired, hits, and accuracy. The mobile application can then request this information from the central hub, which will be sent in response. This data will then be viewable using the applications graphical user interface. All data within the system will be communicated using the UDP protocol.

Multiple Infrared sensors must be incorporated into the prototype’s sensor vest to receive information from the laser guns. This information will then be decoded and transferred to the central hub.

8.3 Hardware
Each sensor vest and laser gun combination will each have their own microprocessor. The gun will make use of long-range LEDs in order to send signals to other sensor vests. In order to keep track of all the game events and scores, a central hub is needed. This consists of a small processor capable of establishing communication with all components of the system using Wi-Fi and the 802.11b protocol. The central hub will directly communicate with both the sensor vest and the android application.

8.4 Software
This system contains several aspects of software. The central hub will have a server program running constantly in order to maintain, update, and configure the gameplay throughout the life of the game. There will also be software in the sensor vest and gun which will be used to encode or decode the IR signal between the sensor and vest.

In addition to this system, there will also be an android application in order to see game statistics, update game settings, and to initialize or end the game. This software will be directly communicating with the central hub.

8.5 Mechanical
The prototype will include a basic housing for the initial system that will protect the electrical components from natural conditions and basic wear due to use.

9 Production Model
While the previous chapter describes the prototype design, the purpose of this section is to describe the actual production model and analyze different aspects if the design were to be manufactured on a large scale.
9.1 System
The production model of our system will be sold in two ways. Firstly, our production model will be distributed in the form of a starter pack. The starter pack will contain two laser guns, two sensor vests, and a central communication hub. Secondly, our product will be sold in the form of an expansion pack. This expansion pack will include two laser guns and two sensor vests. This will allow for great modularity with our product. This means that our customers will be able to increase their team size by one player per team with each additional purchased expansion pack. The android application will still be available for free.

9.2 Central Hub
In order for the central hub to be manufactured on a large scale, several components of the hub’s design would change dramatically. In order for home users to integrate the Lazer-Ops system with their existing network, the central hub would use a Wi-Fi transceiver in order to connect to the existing system.

In addition to this, a development board, such as the Raspberry Pi, would not be used. Instead, a circuit board would be created in order to utilize an ARM based processor, or perhaps a simpler microprocessor. These changes would dramatically increase the performance of the system, while greatly reducing the cost.

9.3 Laser Gun & Sensor Vest
The gun and vest subsystems will maintain the same physical structure as discussed for the final product. Some differences would include a sleek design (custom plastic molds) for the gun encasement and sensor casings, as well as a display on the gun for ammunition readings. The microcontrollers in the systems would only support all the functionalities that have been made available for the gun and vest and there would be wireless communication between the two components.

9.4 Mobile Android Application
The production model of the mobile application would be developed on cross-platform program. Some candidates for cross-platform development would include the Unity engine, Xaramin, Titanium Mobile, RhoMobile, and Appcelerator among others. Using a cross-platform development suite will allow the mobile application to be developed for Android, IOS and Windows Mobile operating systems simultaneously, allowing almost all smartphone users to use the application.

10 Conclusion
This section concludes and summarizes possible improvements that could be implemented using our system.
10.1 Possible Future Improvements

Additional functionality could be added to the laser gun/vest system that would include multiple IR receiver signals, or a signal bus. By differentiating between where a player is shot with either of these methods, more interesting feedback could be supplied to the end user. Additionally, the gameplay itself would be far more intricate, as different areas could cause different damage to a player’s “health”. These added facets would also impact the final application dynamics, as new categories could be displayed. The Wi-Fi transceiver could also be improved with Wireless-N protocol, which would nearly double the Wi-Fi networking range.

These improvements are made available via the UDP designed by the team. Our protocol leaves room for nearly thousands of distinct messages to be sent from the gun to the central hub for processing, and the same is true for the application to hub communication.

11 Acknowledgments

Team Lazer-Ops would like to thank the following individuals for their continued support throughout the design and development of this project.

- Sang Mark Chung - created initial team posters
- Robert DeKraker - provided equipment and deliveries.
- Doug Herder - stitched laser vests
- Chuck Holwerda - Provided guidance and some electrical materials for gun, vest, and central hub construction
- Phil Jasperse - Provided advice regarding mechanical construction and materials.
- Encouraged safe and cheerful working atmosphere
- Stanley Stanley - Kept work area clean and joyful.
- Steven VanderLeest - Encouraged, pushed, and guided Team Lazer-Ops to success, particularly in documentation and presentation
12 References


---

1 http://m.citinterface.in/images/portfolio/MediumLaserPistol.jpg
2 http://www.gurustump.com/media/images/laser_tag.gif
4 http://Qcfec.com
5 http://lalasertag.com
6 http://www.zonehead.com/e-mail-us/faqs.html
7 http://web.princeton.edu/sites/ehs/labpage/laserpointersafety.htm
8 https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcSVshe8q39JEeKZIz--tTBIz2pA4k5mZKjy2em0KmgpJXiobpXz
9 https://az523807.vo.msecnd.net/site/media/PhoneFrame.png
10 https://www.eol.ucar.edu/rtf/facilities/isff/LOCAL_access_only/Wire_Size.htm