Battlebot

Autonomous Target Recognition System

SENIOR DESIGN I

University of Central Florida

Group 2

Clayton Cuteri	Computer Engineering
Corey Nelson	Electrical Engineering
Kyle Nelson	Electrical Engineering
Alexander Perez	Computer Engineering

Sponsor

Lockheed Martin MFC: Department of Applied Research

TABLE OF CONTENTS

1.0 EXECUTIVE SUMMARY

Autonomous Targeting or Automatic Target Recognition (ATR) utilizes advanced technology to independently identify specific objects. ATR is achieved through complex intercommunication between computing algorithms and various sensors. Target specification can vary greatly depending on the type of detection algorithms and sensors employed. To receive accurate classification as an autonomous system, the said deployed unit must have the capability to fulfill its intended purpose free from external human influence. The purpose-built algorithm serves to correctly distinguish between targets and non-targets; protecting against complications in object recognition. Furthermore, several sensors may be integrated, primarily functioning as the line of sight. Factors such as environment and objective type are considered when choosing the best sensors for the job. Environmental considerations affecting sensor choice include altitude, distance, temperature, visibility, etc. Additionally, it is important for designers to know whether the intended targets are of biological or inanimate nature.[1]

Initially introduced as a solution to searching for and destroying military targets in the 20th Century, ATR has evolved into a fully mature product in the Aerospace and Defense industry and finding newer innovative and useful applications in the domestic consumer market. Defense contractor Lockheed Martin alone, currently produces and delivers a diverse portfolio of targeting products which support a wide range of combat operations for its customers. Worldwide, the automotive industry recently began incorporating ATR in its domestically sold luxury vehicles to automatically detect and avoid collisions with pedestrians and other vehicles. Almost every major automotive manufacturer is now investing heavily in such technologies to help deliver fully autonomous vehicles by the year 2020.[2]

In simplistic terms, successful ATR is achieved using at least one sensor mode and a detection algorithm. Long range systems might approach target acquisition using radar. Visible spectrum imagery requires sufficient light and has limited range but radar[3] doesn't rely on external factors to function as it sends and receives self-generated signals to track objects. Forward Looking Infrared can be combined with visible spectrum imagery to produce a capability that overcomes the impairment of poor visibility conditions such as fog. If the target in mind produces a distinct thermal signature, infrared provides an attractive solution. Once a mode of sensing is selected, an algorithm will need to be developed to mitigate error in target selection. The detection algorithm[4] will then be thoroughly developed and trained by example. A compiled database[5] of these experimental scenarios will provide the background information required to judge whether a detected object is the intended target.

This document will serve as a comprehensive overview of the research, design, and prototyping for an autonomous targeting system that will ultimately see integration into a manually remote-controlled ground-based robot. With fully functioning ATR, the system will no doubt find its home suited to a field of offensive and defensive applications. The autonomous targeting system will allow a synced projectile-launching device to engage and accurately hit two long range targets, and a mobile enemy vehicle. The all-up system will provide a user-interface that utilizes wireless video imagery overlays, allowing the user to both manually control the movement of the robot and simultaneously determine when the weapon has achieved target lock.[6]

A multitude of sensor modalities and additional supporting electronic hardware will be examined to select the best possible technologies suited for the job. Several sensors coupled with a mature detection algorithm will coordinate utilizing image fusion techniques to achieve high target accuracy.

2.0 INTRODUCTION

2.1 NARRATIVE

2.1.1 Motivation

In the 21st century, human civilization has achieved great milestones in international cooperation however, the existence of radical factions continues to threaten lives across the globe. Defensive measures are taken to protect against such hostility. Modern weapons[7] are deployed to detect and eliminate these targets which are too often manually operated. Targets aren't always designated for military action either. With so many cars on the road, pedestrian drivers are tasked with maintaining constant situational awareness to protect themselves and others from harm. For all these occasions, although it is simpler to rely entirely on a human component for target locating, as an autonomous system driven by algorithms, sensors, and computational processing, ATR allows for potentially much greater accuracy, reliability, and safety. Human beings are naturally subject to fatigue and other physical limitations such as a lack of visual processing abilities. More specifically: ATR can detect objects in more adverse conditions for greater periods of time without experiencing fatigue, reaction delay, indecisiveness etc. Depending on the variety of objectives undertaken, an ATR system will be custom tailored to ensure mission success. For example: a human operator simply couldn't compete in a low light situation where a thermal infrared imaging or radar system are more effective options. In time-critical applications, high performing, well trained ATR systems will drastically outperform human operators as they can lock onto a target almost instantaneously: protecting the warfighter and saving civilian lives both at home and abroad. In that way, ATR provides a safer alternative, lending itself as substitute contender in a dangerous environment that could potentially end in fatal consequences

2.1.1.1 Achievement

As in any competitive industry, a customer could have many contractors compete in the development of a product. Group 2's autonomous system will go head-to-head with several other UCF robot groups to demonstrate a winning product to the customer: Lockheed Martin. To meet the necessary requirements, each robot group will work closely with local Lockheed MFC representatives: Kenny Chen and Johnathan Tucker. As in industry, regular telephone conferences (telecons) will be held to keep the customer up-to-date with the current progress of the teams. By testing the final products head-to-head with other teams' designs, a realistic assessment can be made of the design effectiveness in the field. The best system will not only meet the requirements of the customer but need to go above and beyond in terms of cost-effectiveness and performance standards. Only exceptional innovation will pave the way for a victorious outcome. A broad spectrum of Firmware and Software engineering skills will need to be called upon to deliver a product that is worthy to the customer. If the product is worthy of making a lasting impression on key evaluators Kenny and Johnathan, several opportunities might be made more tangible with respect to full time employment at Lockheed Martin.

2.1.1.2 Collaboration

As previously discussed, the robot project will ultimately result in a competition between three groups. Although it is a high value interest of which group will achieve the best design possible, it is of notable interest that a location on the UCF campus, apart from the rest of the senior design groups, could be reserved in sole favor of the Lockheed Martin robot groups. In a similar way, it is comparable to research and development environments commonly found in professional engineering corporations. Intermingling and coexistence are qualities highly encouraged by the UCF faculty and Lockheed Martin overseers. Many senior design projects, though tremendous learning experiences in themselves, simply do not come close to this level of realism established by the UCF/Lockheed Martin robot project. When working in a real-world setting, engineers across all matters of disciplines will need to co-exist and work together towards a single goal: the final execution and successful delivery of the required product – in this case a robot with a fully autonomous targeting system. This can be challenging, especially when each professional individual is trained so vigorously in a specific manner. Each robot team will consist of Electrical Engineers, Computer Engineers, Mechanical Engineers, and Computer Science experts. Although the skills each discipline brings to the table are of priceless value, many conflicts will no doubt arise. The conflicts in reference are not of inter-social categorization, but of sheer difference in engineering & design goals. In an environment where a single group consists of several individual teams, each side will vie for the most effective solution in their realm of focus. A complex medley of balanced cooperation will need to take place to ensure the ripest deliveries from each team. Only then will a great product emerge from the caverns.

2.1.1.3 Affiliation

Approaching nearly a century of technological advancement, it is safe to say that the target recognition modalities have reached climactic maturity. Though steady advances in processor manufacturing continue to improve the speed and reliability of sensor products, industry leaders such as Lockheed Martin are delving into new areas of sensor innovation. One of their more recent projects, the JAGM missile, aims to combine the best capabilities of several variants of the Hellfire Missile. By pooling multiple sensor modes into a single system, known as image fusion, the probability of successful objective completion is greatly increased through the JAGM missile. If one sensor fails or is insufficient of the means required to locate the target, another sensor would carry forward to complete the task. Accepting the assignment from Lockheed Martin and collaborating with a UCF mechanical engineering team, further research and development of these types of technologies will be made.

2.1.2 Objectives

The Electrical and Computer Engineering team is responsible for all the electronic systems of the robot including the sensors and core target acquisition software however, a mechanical team and a computer science team will focus on ensuring maximum performance of other indispensable systems. Only through intense planning, integration, and testing with the mechanical team and the computer science team will the electrical and software integration of a fully autonomous targeting system onto the manually controlled mobile unit be possible. For a broad look at the system design, see Figure 2.1.2 i.

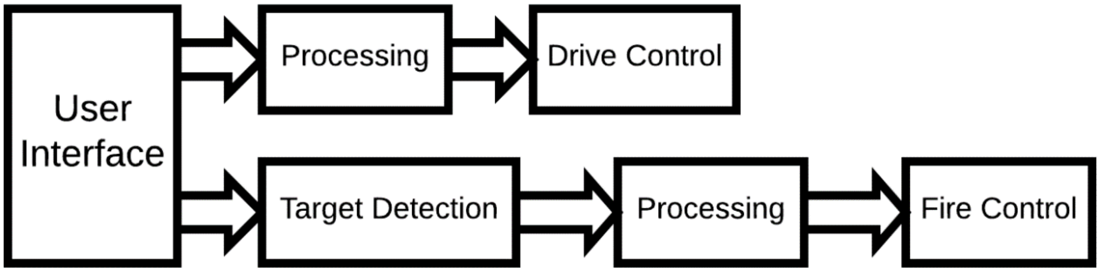


Figure 2.1.2 i: Autonomous Targeting System Overview

2.1.2.1 Engineering Analysis

Following a thorough analysis of the requirements outlined by Lockheed Martin, different sensors such as radar, ultrasonic, infrared, LiDAR, etc. will all be considered. In the end, the most efficient and accurate sensor will be integrated based on the findings and successful image fusion between the two systems. It is important that the detection algorithm be sufficiently trained to effectively identify and engage the objective targets. Lockheed Martin specified to the robot groups that the robot will need to recognize several targets. Two stationary targets will be located on the competition grounds. Two obstacles will be of close to medium distance and two targets will be of long range distance – at the opposite corners of the course. The robot will need to evaluate those 2-dimensional targets and determine that they meet the criteria to be fired upon. In addition, successful hits on the enemy team’s robots will score even more points, so the ability to locate and quickly track the enemy teams fast moving robot will be a high priority. A final case to consider: if the enemy team’s robot were to malfunction, a “medic” can step into the course to retrieve said robot. Lockheed Martin specified that targeting and successfully firing upon an enemy medic is of fair evaluation and will act as additional source of points. It is critical that the robots autonomous targeting system can detect and process these various possible threats. Ultimately the system will accomplish as many hits as possible on the enemy targets and demonstrate accurate, stable, target tracking functionality.

2.1.2.2 Efficiency

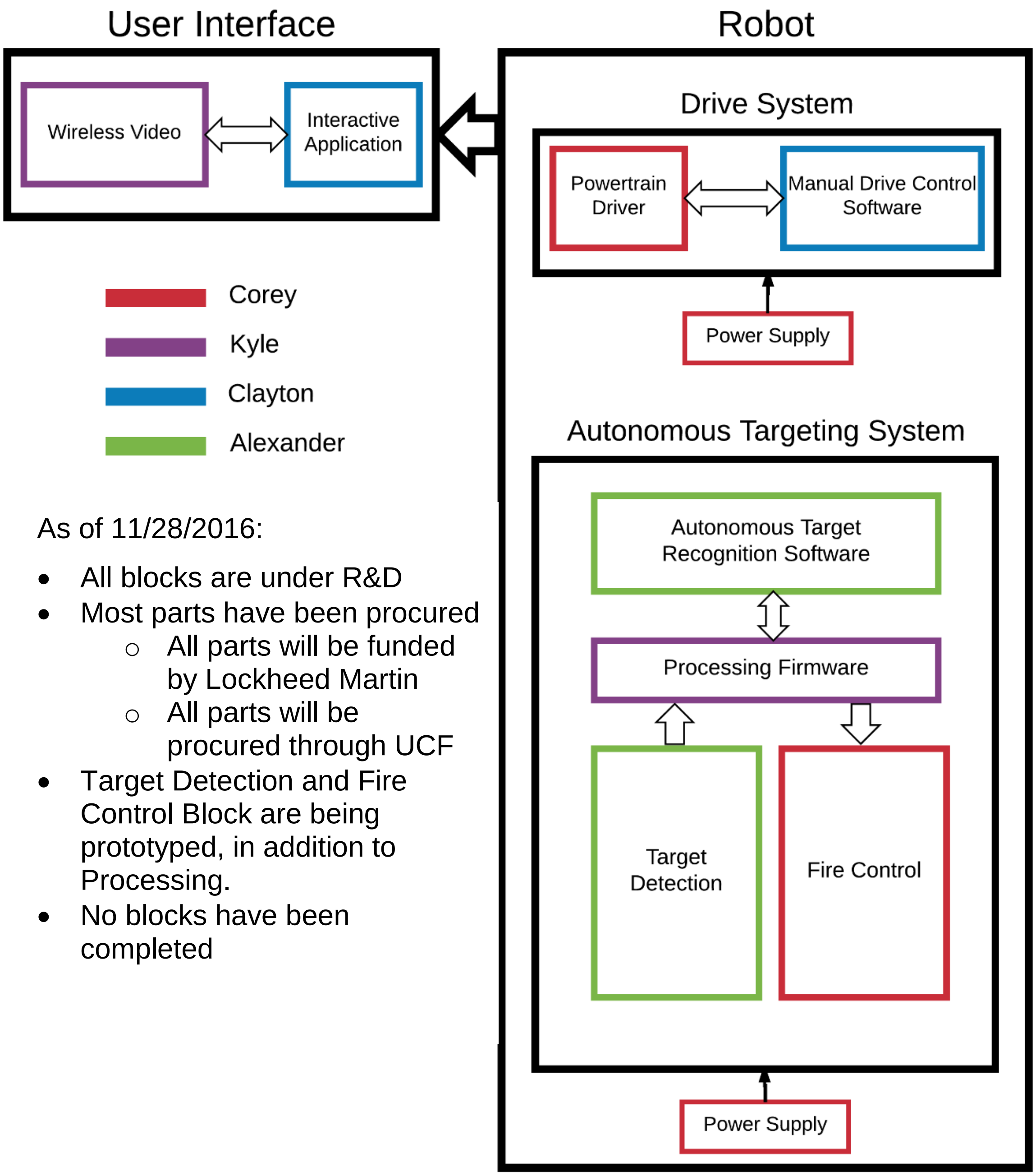
The autonomous targeting system will be developed in coordination between Software, Firmware, and Mechanical engineering students. The unique individual expertise of the team’s will allow the design of an optimized mobile structure to support the best possible custom targeting system executed by the electrical and software team. This is because the final product specified requires the ability both to manually navigate the indoor course and locate targets autonomously.

The project was proposed and accepted with a total budget of $2000 allocated by Lockheed Martin. Designs will be thoroughly researched and prototyped to achieve successful resource management. Prototyping of parts and designs will minimize wastefulness of time and resources. A master plan of the project will be created. The master plan allows the electrical and software team to efficiently lay out the broad scheme of the project. Milestone tasks will be tracked and completed on time to ensure successful delivery of the project by the promised due date. Through successful resource, budget, and time management the UCF robot team will deliver an attractive and effective solution to the proposed requirements.

The robot will need to consume power at a rate efficiently enough to last for at least the 10-minute duration of a single match while at the final competition. Many components are going to vie for juice while operational so it is important to choose energy efficient parts and have enough raw energy storage. Additionally, the larger the drive system and targeting system, the heavier the robot will be. The heavier weight of the robot would mean requiring more energy to move. Thus, component weight should be kept as minimal as possible. Another factor to consider is the method of wireless communication. A speedier connection with a higher resolution video feed is bound to negatively impact the duration of the robot. Although a snappy connection could prove a huge advantage when competing, it is a higher priority to ensure reliable and steady operational capability.

2.2 REQUIREMENTS SPECIFICATIONS

Lockheed Martin made it clear in their requirements for the robot to achieve fully autonomous target recognition and tracking via at least one sensor modality. The robot should primarily be manually drivable and fire upon targets automatically once detected and given firing permission. Figure 2.2 i provides a significantly detailed look at the composition of the autonomous targeting system and how the tasks will be divided within Group 2 for management purposes. Figure 2.2 i also shows the most updated stated on the progress of the individual blocks. The robot will be controlled wirelessly via a user interface. The independent Drive System and Autonomous Targeting system will be powered individually within the robot. Utilizing efficient integration of a printed circuit board (PCB) and a commercial microcontroller, the automated targeting software will be able to process the information received from the sensors and aim the weapon accordingly. Detailed hardware specifications are numbered accordingly in table 2.2.1 i and detailed software specifications are provided subsequently in table 2.2.2 i.



As of 11/28/2016:

- All blocks are under R&D
- Most parts have been procured
  - All parts will be funded by Lockheed Martin
  - All parts will be procured through UCF
- Target Detection and Fire Control Block are being prototyped, in addition to Processing.
- No blocks have been completed

Figure 2.2 i: Subsystems Flowchart

2.2.1 Hardware Engineering Specifications

Item	Engineering Specification	Justification
1	System cannot exceed the *weight* of 15 lbs. and must not exceed the *dimensions* of 3' x 3' x 3', when stationary in a static position.	The system demands reduced weight for high mobility to avoid incoming enemy fire and to reduce power consumption. The robot must accommodate the specific size to satisfy customer requirements.
2	The system must function of on-board power for at least 30 minutes.	The robot needs to perform for 3 10-minute rounds during the final competition.
3	A maximum of *two* weapon systems and *two* sensor modalities will receive integration into the mobile unit.	As required by Lockheed Martin.
4	The sensor(s) will produce accurate target acquisition of an arbitrary mobile target at a distance of up to 30'.	Lockheed Martin requires the robot to fire at and hit a robot and/or human medic up to said distance(s).
5	The sensor(s) will accurately target a stationary picture representation of a human face measuring 12" x 18" at a distance of up to 30'.	Lockheed Martin requires the robot can fire at and hit said targets.
6	A wireless connection will provide a user with administrative authority over the autonomous targeting system and the ability to manually control the movement of the robot at a distance of at least 20'.	Lockheed Martin requires wireless control of the robot
7	A Micro-controller will support the intercommunication and processing between the sensors, weapon system, and transmission	Necessary to support the computers command and controller of target detection and fire control.

Table 2.2.1 i: Detailed device hardware requirements

Item	Engineering Specification	Justification
8	A custom PCB with at least one surface mounted microcontroller will interface with the servos, sensor, and on-board computer.	As required by UCF ECE Department
9	At least Motor Driver card will control the direction and speed and of the mobile unit’s motors	To allow manual control of the robot as required by Lockheed Martin.
10	A Durable and Sturdy Weapon Mounting Structure will enable the capability to pan and tilt up to a 3-lb. weapon system	To support accurate target acquisition.
11	At least one power supply will ensure the function of the Drive System	To prevent interference between subsystems.
12	At least one power supply will ensure the function of the Autonomous Targeting System	A power supply will allow the system freedom and modularity

Table 2.2.1 i: Detailed device hardware requirements (continued)

2.2.2 Software Specifications

Item	Engineering Specification	Justification
1	Wireless security protocols will prevent unauthorized access to the robot.	It is important that unauthorized users cannot tamper remotely tamper with the functionalities of the robot.
2	A user interface will provide wireless administrative control of the robots targeting system in addition to video imagery overlays up to 40 feet.	As required by Lockheed Martin
3	Autonomous Targeting Program will provide capability to autonomously locate, track. and fire at targets.	As required by Lockheed Martin and UCF
4	The weapon system will *notify* the user upon target lock and provide the designated targets *distance*.	Supports checks and balances to assure accurate target acquisition.
5	Upon achievement of autonomous target lock, the user can *authorize* firing of the weapon.	Fulfills autonomous targeting requirement but allows the administrator to prevent misfires
6	The user can wirelessly stop the weapon from firing within 40 feet.	Acts as a safety mechanism to remotely disable the system for maintenance
7	The robot speed will be manually controlled wirelessly within 40 feet.	As required by Lockheed Martin
8	The robot direction will be manually controlled wirelessly within 40 feet.	As required by Lockheed Martin

Table 2.2.2 i: Detailed device software requirements

2.2.3 Course Specifications

When creating the autonomous targeting system, Green Team / Group 2 will rely in part on the course layout as specified by Lockheed Martin. The layout provided in Figure 2.2.3 i, will drive the design and research of the system to accommodate the distance specifications and ensure that the robot functions successfully by tracking and attacking the enemy target anywhere within the specified ranges. As discussed, Green Teams robot cannot leave the “friendly” zone. If Green Teams robot crosses into enemy territory, a penalty will be applied. This means that, to score maximum points, Green Team / Group 2s robot will need to “see” the enemy, targets, and obstacles at distances of no more than 40 feet.

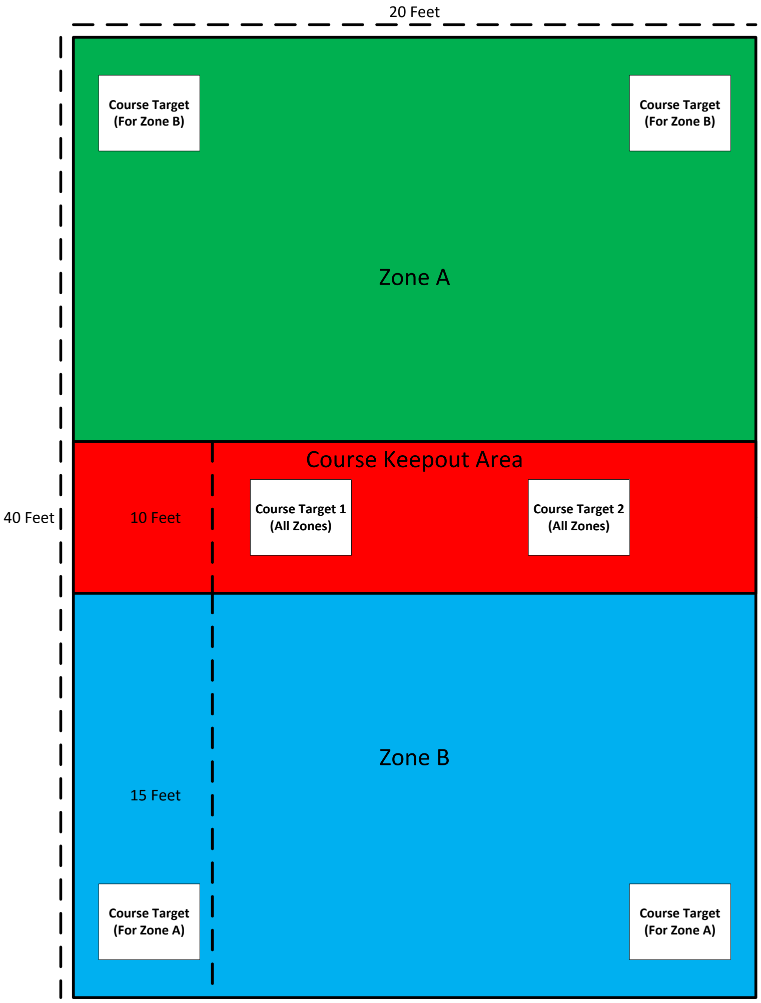

20 Feet
Course Target (For Zone B)
Course Target (For Zone B)
Zone A
Course Keepout Area
40 Feet
10 Feet
Course Target 1 (All Zones)
Course Target 2 (All Zones)
Zone B
15 Feet
Course Target (For Zone A)
Course Target (For Zone A)

Figure 2.2.3 i: Course Layout provided by Lockheed Martin

2.3 HOUSE OF QUALITY

Every day, engineers must consider tradeoffs. They must decide what is more important for the overall quality of a product. It is understood that there are requirements in engineering the product and requirements to market the product. Table 2.3 i helps to establish what is most important for both.

			Engineering Requirements				
			Range	Accuracy	Dimensions	Power	Cost
		Polarity	+	+	-	+	-
Marketing Requirements	Quality	+	↑↑	↑↑		↑	↓
	Installation	-			↓		↓↓
	Portability	+	↓		↑↑	↓	↓
	Cost	-	↓↓	↓		↓↓	↑↑
Objective Measures	Units of Measurement		Feet (ft)	Inches (in)	Feet (ft)	Watts (W)	Dollars ($)
	Group 2 Autonomous Targeting System		15 ft	1 in	3 ft	25 W	$500

Table 2.3 i: House of Quality

A single, green, up arrow ↑ indicates a positive correlation; two ↑↑ signify a strong positive correlation. A red arrow ↓ indicates the opposite. The polarity is either positive + or negative - which indicates if the intent is to increase or decrease the requirement, respectively.

For example, look at the first box that relates range and quality. The product must be able to launch a projectile a certain range to meet the engineering requirement. This requirement will have an impact on how Group 2 / Green Team could market this product. Therefore, if the product can shoot further, it increases the product’s overall quality. Therefore, there is a strong positive correlation between quality and range – increasing one, increases the other.

To meet requirements and produce a high-quality product, it will be difficult to make it cost efficient while creating the product. The graph shows us this will be the hardest requirement to control. This is observed by the number of red arrows in the “Cost” row and column. Luckily, a sponsor will cover the costs. However, if Group 2 / Green Team were creating a product for the public, the issue of cost would be much greater since there is more interest in a profit.

Since cost will be difficult to control, Group 2 / Green Team must look to how Group 2 / Green Team can use cost to benefit other aspects. One place it can help the product, is in its overall quality. If Group 2 / Green Team puts forth our resources into increasing the product’s range, accuracy, and power, it will create a very high quality product. Increasing these aspects will decrease our ability to transport the product. Again, since this is not going to be sold to the public, this requirement is not as troublesome for our main goals.

Another marketing requirement that will be hard to achieve, is the installation requirement. Now, it is still early to know how long certain components will take to install, but Group 2 / Green Team can gauge by the House of Quality, that aspects will harm, more than help, improving this requirement. In our environment, Group 2 / Green Team will have engineers who know the equipment well enough that installation will not be an issue. If Group 2 / Green Team were to give this product to someone else to install it, then this requirement would be a lot more important. For our sake, there are more important requirements to pursue.

3.0 RESEARCH

3.1 EXISTING SIMILAR PROJECTS AND PRODUCTS

Autonomous weapon systems can come in all different types of sizes, designs and uses. There can be many uses for these types of systems that can be used for hobbyist or for expert use like the military. For this project nerf projectiles will be used instead of paintballs or actual bullets. For our purpose, Group 2 / Green Team will also need the system to be small enough to mount onto a platform that will be manually controlled by the user. Our system must be able to autonomously detect another opponent's robot and targets that are in the course. Our research was mostly explored in this area by looking at previous senior design projects related to turrets. This was also the same method to research similar projects that were completed on various websites. From this research, Group 2 / Green Team could find the Nerf Vulcan Sentry Gun from Instructables.com and Self-Targeting Automated Turret System from the senior design project of 2014. Each of these designs also featured different ways of object detection and weapon. After reading through these projects, Group 2 / Green Team could gather our own ideas and implement them into our own design.

3.1.1 Nerf Vulcan Sentry Gun

A project that most closely relates to the group would be a project called the Nerf Vulcan Sentry Gun, as shown in Figure 3.1.1 i. This project makes use of a webcam to see when a target is in front of the gun. After an object is in front of the webcam, it will determine if it should shoot or not. A program was used to achieve this. A certain image or color would be deemed “safe” or “not safe” to shoot. This same process would be used for our project; however, Group 2 / Green Team would have a much smaller weapon system and ours would need to work at a further range.

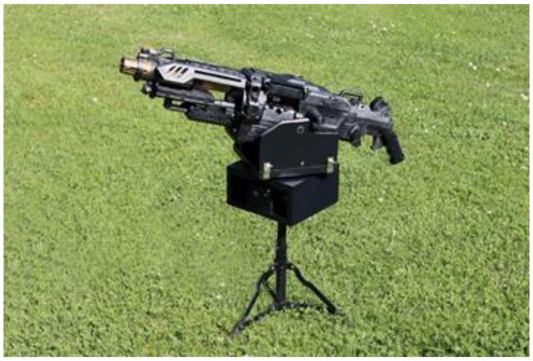


Figure 3.1.1 i: Nerf Vulcan Sentry Gun

However, the process of the webcam detecting a close-range object, in our case an opposing robot, would be useful. The entire system is controlled by a microcontroller, which is what our project will be using as well. The biggest disadvantage to this design is the size of the system. For the group's purposes, the weapon system must be mounted on top of a pre-existing platform and must also be as lightweight as possible for maximum mobility of the platform. Researching this project clarified what exactly a sentry turret will accomplish and how it will work. This gives a clearer idea on how to modify it for our own project.[8]

3.1.2 Self-Targeting Automated Turret System

Self-Targeting Automated Turret System or STATS, shown in Figure 3.1.2 i, is a UCF senior design project from 2014. It is the work of Elso Caponi, Michael Lakus, Ali Marar and Jonathan Thomas. STATS is meant to be a lightweight, mobile and self-contained turret that can be placed in a large area to protect against unwanted intruders. A webcam is used along with the XBee Wi-Fi transmitter is connected to a tablet to receive live video from the webcam. This is one of the requirements for the group's project and Group 2 / Green Team will also be using a similar system to achieve this. STATS makes use of three different servos that will each complete a desired task. Two servos are used for pan in the X and Y directions. The third smaller servo is used to pull the trigger on the weapon system. STATS uses a microcontroller to process the information that is gathered from the webcam that is sent through targeting software. The targeting software was built using open source libraries and by using the Processing program language due to the availability to support and versatility. The tracking system that was used is called blob detection. Blob detection is a popular method of detecting and tracking motion in the OpenCV library. This would be a beneficial method to use since it is readily available and many projects have used it in the past. Overall this project was helpful in learning about a popular object detection library that is available to use. Learning about this detection method was effective in choosing a proper method and eliminating other methods that would not be beneficial for the final product.[9]

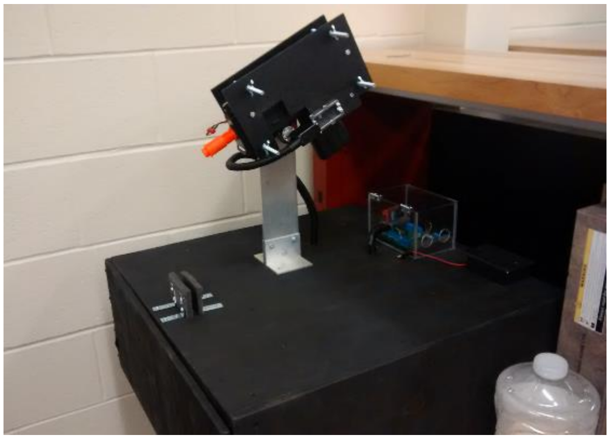


Figure 3.1.2 i: Self-Targeting Automated Turret System

3.1.3 Automatic Nerf Sentry Gun

Another project that is close to the group's project is called the "Automatic Nerf Sentry Gun" or ANSG, which was created by John Park. The ANSG uses an ultrasonic distance sensor to track motion in front of the weapon along with an Arduino to run the process. The ANSG also makes use of servo motors to control the weapon's horizontal movement. The weapon is constantly moving back and forth, scanning for a disturbance that the ultrasonic sensor detects. Our group will be using a sensor comparable to the one the ANSG used to detect an object. The trigger of the weapon is controlled by the Arduino and will be told to shoot if the ultrasonic sensor detects an object in front of it. A preset value is set into place in the weapon's program for a reference point. Once an object appears closer than that reference value, the gun will then stop sweeping and will then begin to start firing. Overall, this project is relatively akin to what our group is trying to achieve. However, this project only has the weapon sweeping on the horizontal axis only. This will become an issue when an object is not the same height as the weapon. For the actual size of the weapon, it is relatively small in comparison to some of the other projects. This is beneficial because Group 2 / Green Team need the system to be able to be mounted on top of a robot. Overall, this project has useful information that is beneficial to the group's project design. It uses similar technology and performs the basic task that the group is looking for it to perform.[10]

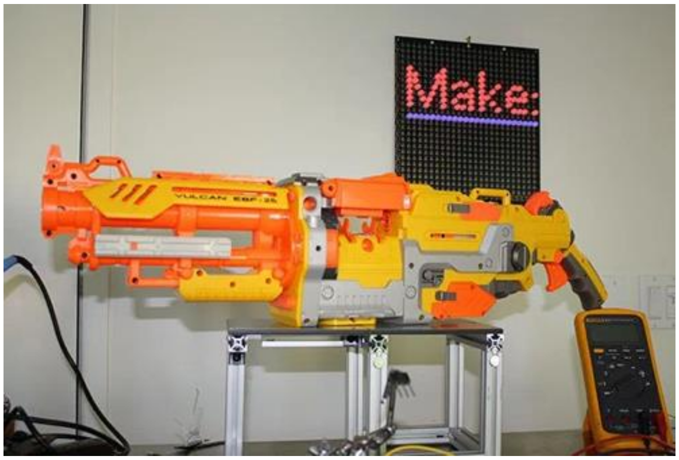


Figure 3.1.3 i: Automatic Nerf Sentry Gun

3.1.4 Autonomous Visual Rover

The autonomous visual rover (AVR), designed in 2009 by Diante Reid, Liem Huynh, and Sean Day, is a robot designed with several different capabilities that involves movement while tracking objects. One of the goals of the AVR, shown in Figure 3.1.4 i, is to recognize and process object images. The AVR will decide between whether the object is an obstacle that needs to be avoided or an object that could do harm to the robot. If it's an obstacle, the rover will efficiently maneuver around the object. If it is the latter, the robot is designed to take defensive measures using its weapon defensive system. These features of the AVR are very close to the functionality of our very own project. Our weapon system will need to fire at targets upon recognition. Once the system has identified a target and said target is within range, our weapon system will need to accurately shoot a projectile, hitting the target. The accuracy is key when dealing with targets. Without accuracy, it would render the system obsolete. The AVR uses servos, a custom printed circuit board, and a power supply - all things that will be used to create our project. As described, this project holds many similarities to how ours will look and operate.[11]

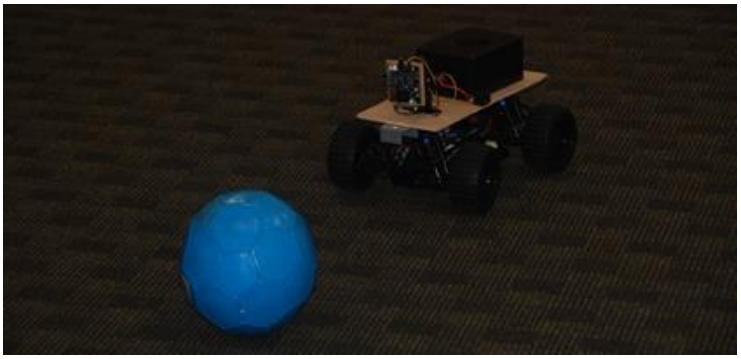


Figure 3.1.4 i: The Autonomous Visual Rover

3.1.5 S.H.A.S.bot

The S.H.A.S.bot was designed in Fall 2012 by Daniel Lanzone, Mike Roosa, Ryan Tochtermann, and Charlie Grubbs. A picture of the robot can be seen in Figure 3.1.5 i. The robot has a sensor to observe the terrain around it. It also has microphone that can detect noise. This functionality is utilized for security measures. The idea of the S.H.A.S. was to design a robot for both commercial and military use to survey the world around us. A consumer would use the robot for house surveillance to identify if there is an intruder in their residency. A military application of the robot would include scouting areas that there is a known threat and they do not want to send in people with the fear of tripping an explosive or walking into an ambush. The robot had a couple important functionalities that are similar and uncommon to our robot. One functionality the robot has in common, is that it's camera is integrated with a wireless link so that the feed can be transmitted back to the user. Group 2 / Green Team will also be using a wireless video stream with our robot along with a similar image processing system as the S.H.A.S.bot. The video stream will be sent wirelessly to an application. This wireless transmission will be similar in design to how the S.H.A.S.bot group implemented this connectivity on their robot.[12]

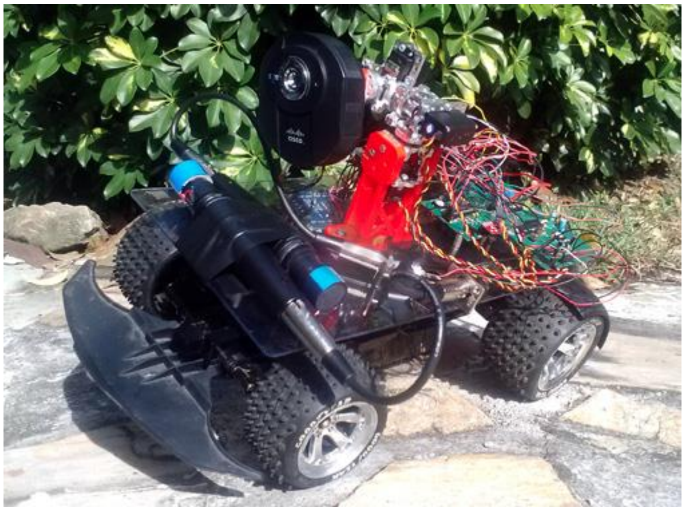


Figure 3.1.5 i: The S.H.A.S.bot

3.1.6 Automated Targeting Proximity Turret

In Fall of 2010, Hector Colon, Adam Horton, Kyle Steighner, and Nicolas Yielding became design on an Automated Targeting Proximity Turret. The group's goal was to create an automated turret that replicated a defense turret. A functionality that is uncommon to ours, is the turret's server-database connection. The Automated Target Proximity Turret takes photos and logs what it sees to a database so that there is a history of what it has seen. This live feed functionality is comparable to the live feed Group 2 / Green Team will be implementing with our project, not only because it is a camera, but because this project is also comparable to ours in the sense that it has an automated turret. The turret is designed to detect, track, and target foreign entities. These are three similar requirements that Group 2 / Green Team have set forth for our weapon system. The Automated Target Proximity Turret, as is the same with ours, is operated using judgement of the surrounding environment to recognize targets. This functionality is what defines the Automated Target Proximity Turret and is a functionality that will be significant to the success of our own project.[13]

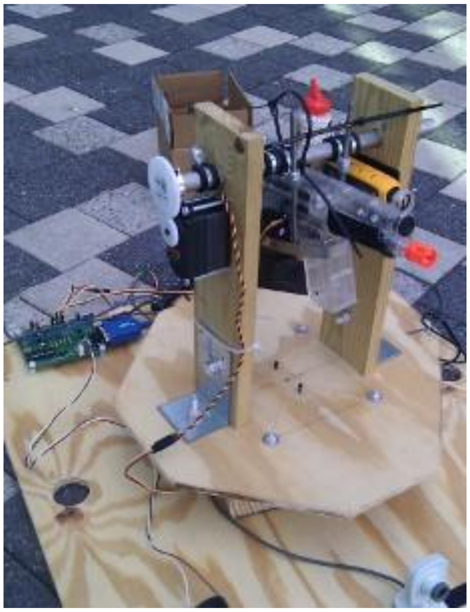
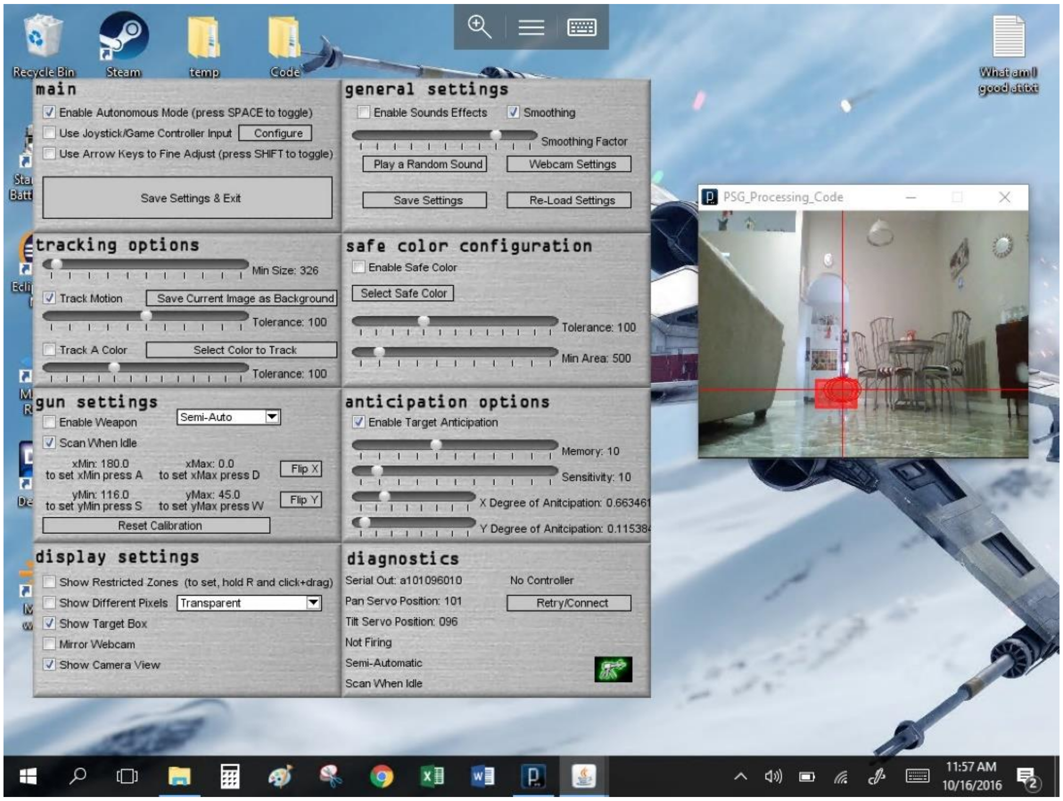


Figure 3.1.6 i: The Automated Targeting Proximity Turret

3.1.7 Project Sentry

At http://projectsentrygun.rudolphlabs.com/ a similar, though stationary, emplacement was designed than can be adapted to a multitude of physical weapon systems to fully automatize them. Complex and in-depth resources are provided which explain how to connect a computer running object oriented software to autonomously communicate with an Arduino to control the servos attached to a weapon and correctly aim it at moving or stationary targets. The software in question is customizable and is multi-modal in implementation. A draw back of the system is that it requires a laptop to run the software. Such a powerful computing device is way out of the budget and mobility requirements Green Team

/ Group 2 must adhere to. This project gives Green Team / Group 2 an excellent example by which reverse engineering could allow the system to be modified to suit the requirements outlined by Lockheed Martin. The open-source Eagle schematics could be improved by having additional areas designed to accommodate a second or even third sensor into the design. Additionally, thanks to rapid advancements in computational technology, even though the original project (created several years prior to 2016) required a laptop to run the high-powered image processing software, Green Team / Group 2 could procure a smaller, more energy efficient processing system that has enough capability to run similar software. This is a critical area of insight given the great requirements of the robot to meet advanced target recognition techniques. Figure 3.1.7 i shows a standard laptop running project sentry guns open-source software using the Processing IDE to track a small domestic cat.[14]



main

☑ Enable Autonomous Mode (press SPACE to toggle)

□ Use Joystick/Game Controller Input [Configure]

□ Use Arrow Keys to Fine Adjust (press SHIFT to toggle)

[Save Settings & Exit]

general settings

□ Enable Sounds Effects ☑ Smoothing

Smoothing Factor

[Play a Random Sound] [Webcam Settings]

[Save Settings] [Re-Load Settings]

tracking options

Min Size: 326

☑ Track Motion [Save Current Image as Background]

Tolerance: 100

□ Track A Color [Select Color to Track]

Tolerance: 100

safe color configuration

□ Enable Safe Color

[Select Safe Color]

Tolerance: 100

Min Area: 500

gun settings

□ Enable Weapon [Semi-Auto ▼]

☑ Scan When Idle

xMin: 180.0 to set xMin press A | xMax: 0.0 to set xMax press D | [Flip X]

yMin: 116.0 to set yMin press S | yMax: 45.0 to set yMax press W | [Flip Y]

[Reset Calibration]

anticipation options

☑ Enable Target Anticipation

Memory: 10

Sensitivity: 10

X Degree of Anitcipation: 0.663461

Y Degree of Anitcipation: 0.115384

display settings

□ Show Restricted Zones (to set, hold R and click+drag)

□ Show Different Pixels [Transparent ▼]

☑ Show Target Box

□ Mirror Webcam

☑ Show Camera View

diagnostics

Serial Out: a101096010 No Controller

Pan Servo Position: 101 [Retry/Connect]

Tilt Servo Position: 096

Not Firing

Semi-Automatic

Scan When Idle

Figure 3.1.7 i: Laptop running ATR software to target a mobile domestic cat

Figure 3.1.7 ii demonstrate streaming the video processing software via Windows 10 Remote Desktop Connection to an iPad mini 2 and accurately tracking human movement. This is a good representation of what Group 2 is trying to achieve because it shows a control panel and video imagery overlay running native on the processing unit.

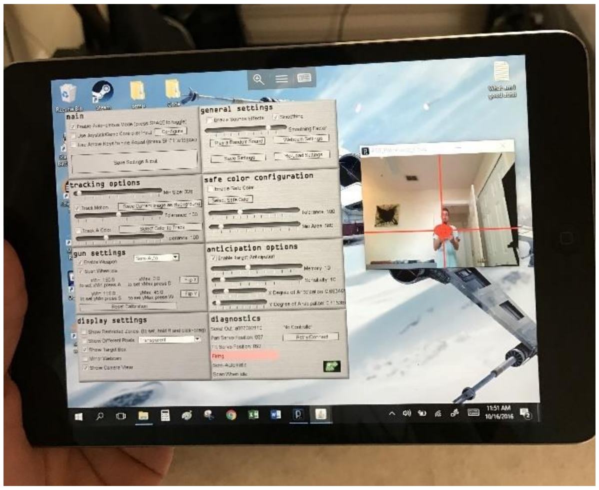


Figure 3.1.7 ii: Streaming ATR Software via Remote Desktop Connection

The video can be viewed on a mobile device which an important requirement from the customer. Additionally, though the targeting is autonomous, several customizable tweaks to the system will be necessary. The control panel provides this calibration feature which is excellent. This provides a good example of how Green Team / Group 2 could wirelessly control and view the autonomous target recognition system on the robot. Currently the software is quite complex and optimized to track quick-moving targets of substantial size such as humans or large animals.

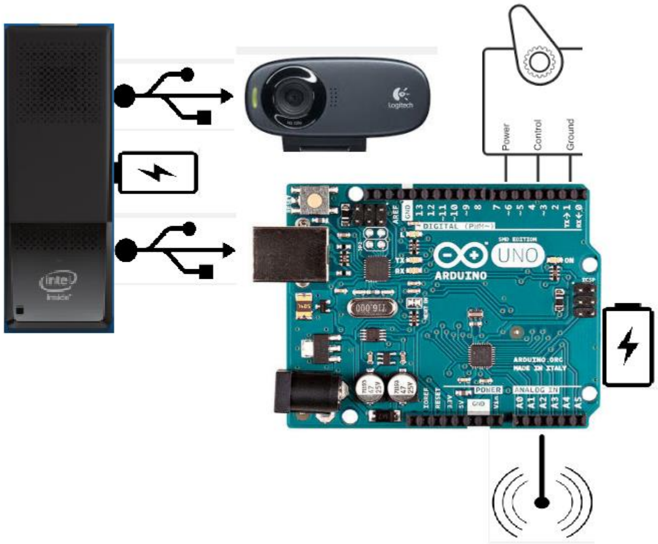


Figure 3.1.7 iii: Hypothetical Project Sentry Gun adaptation
*(Part sizes not shown to scale and are not indicative of final part selections) **

The software works best at close to medium ranges of about 10 feet. With modifications, the software could be improved to detect more specific visual features at further distances and work in tandem with additional sensors, not just the camera. There is a color tracking feature and a manual targeting option but those would largely be useless since Group 2 / Green Team is not looking for any specifically colored targets and is not allowed to manually aim or control the weapon other than to give it authorization to fire and toggle between targeting modes.

Figure 3.1.7 iii provides a very rough hypothetical layout of how project sentry guns’ design could be modified to suit Green Teams / Group 2s requirements. The minicomputer (far left) connects to the camera, processing the visual imaging; and communicates with the microcontroller (Arduino UNO for example) to analyze sensor data and operate servos which aim the weapon. The mini computer and microcontroller are power by their own unique energy sources to suit their specific needs. By replacing the feature-packed laptop with a task-specific minicomputer that has a high power to energy usage ratio, a similar system compared to project sentry guns could be implemented into Green Teams / Group 2s robot.

3.2 RELEVANT TECHNOLOGIES

In this section, Group 2 / Green Team will describe what kind of components are needed to implement the project. For each component, there will be several different types to choose from. Over the course of this research the advantages and disadvantages will be evaluate for each part, and Group 2 / Green Team will decide whether to use the corresponding part for the project or not. This will be one of the most important steps of the project. Deciding what parts to use will determine how well the end project will work and function.

3.2.1 Sensors

There are a few different types of sensors that can be used for this project. In this next section, these different sensor technologies will be researched.

3.2.1.1 Infrared

There are two different types of infrared (IR) sensors that can be used based on their output, either analog or digital. Those sensors with digital detectors will provide a digital high or low to indicate if an object is at a distance that was previously defined. Those sensors with an analog output the actual distance from the sensor to the object. These are the ideal type of sensor to be used because this project requires to object to be in a range of distances.

These sensors use triangulation and a small CCD array to determine the distance of the object in its field of view (FOV). A pulse of IR light is emitted, and this light hits the object and is reflected or it keeps going. Once it bounces off the object it comes back to the detector and creates a triangle.[19] From this, the distance can be measured. Figure 3.2.1.1 i shows a visual representation of triangulation.

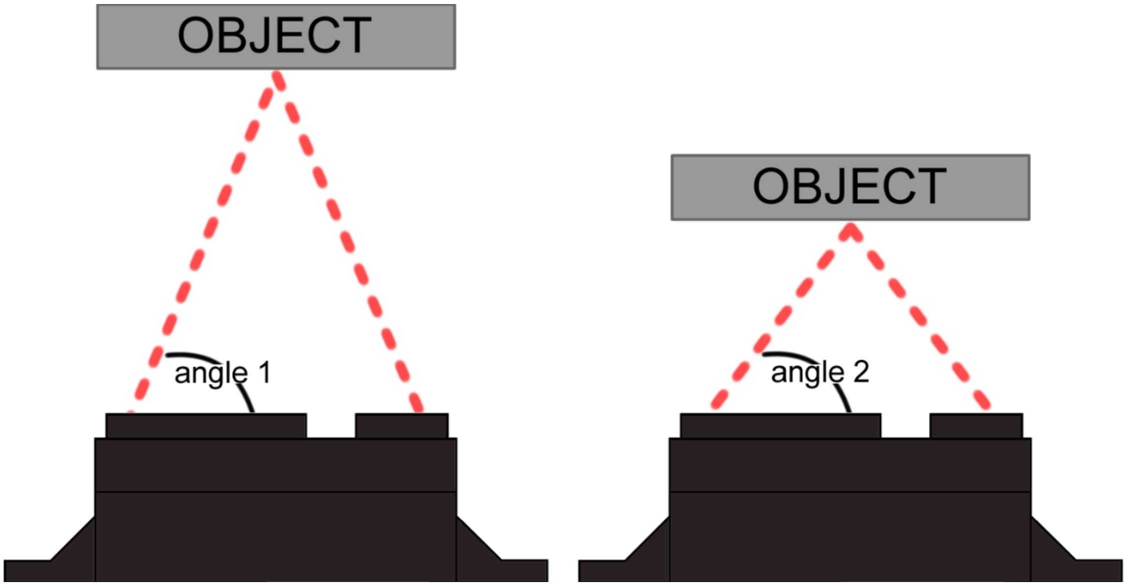


Figure 3.2.1.1 i: Triangulation Overview | *Credit: Justin Gregg @ https://acroname.com/blog/sharp-infrared-ranger-comparison*

3.2.1.2 LiDAR

Light Detection and Ranging (LiDAR) is another type of sensor that can be used to serve the purpose of determining the distance of an object. Usually, LiDAR's are part of a bigger system that can use data to create billions of points. The LiDAR shoots a laser light at an object. Its distance is then calculated by using the time it takes from leaving the sensor to returning to the sensor. These points can then be used to create a point cloud. Depending on the distance (or elevation) a different color is used to display the point.[20] From this, an image is created. An example of the image created by a LiDAR system is displayed in Figure 3.2.1.2 i.

This type of system has great accuracy, but can be very expensive to implement. LiDAR would suite the requirements of the project, but will not be considered due to the cost of using such a system.

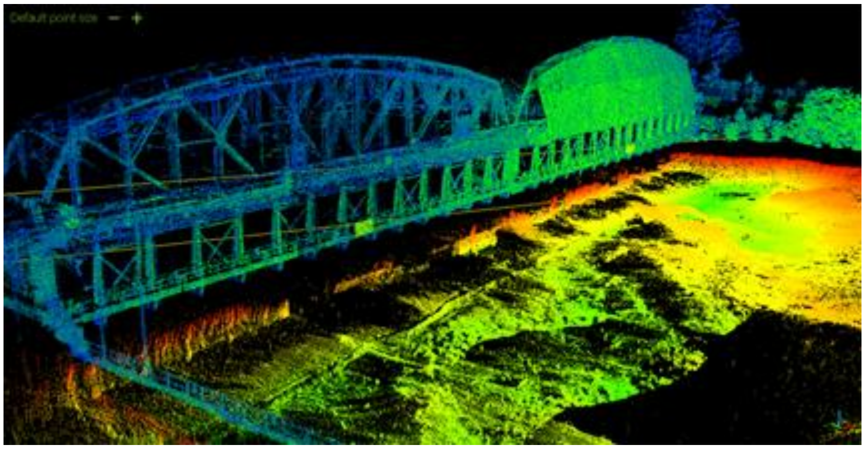


Figure 3.2.1.2 i: Example of a LiDAR Point Cloud

3.2.1.3 Ultrasonic

Another type of sensor that can be used is an ultrasonic sensor. These sensors work by emitting a sound pulse that reflects off an object. This reflected sound is then sent back to the sensors receiver. This detection of sound generates the output signal. This signal is either digital or analog.

The reason that sending a sound signal to sense an object is based on the principle that sound has relatively constant velocity. Since the time for the sensor's beam to hit the target and be sent back is directly proportional to the distance of the object, Group 2 / Green Team can calculate the distance between the two. These sensors can detect many different objects. If the object that the sound pulse hits does not absorb the sound its distance can be calculated.

Ultrasonic sensors are made up of four basic components. These are the transducer/receiver, comparator, detector circuit, and solid-state output. The transducer is responsible for the sending the sound wave to an object. It also receives the reflected waves from the object.

The comparator and detector circuit are responsible for calculating the distance from the object to the receiver. It does this by comparing the emit-to-receive timeframes to the speed of sound.

The solid-state output generates the electrical signal that can be used by a micro-controller. The digital signal will indicate whether there is an object or not. The analog signal will specify the distance to the object.[21] A picture of the overview of the sensor is shown in Figure 3.2.1.3 i.

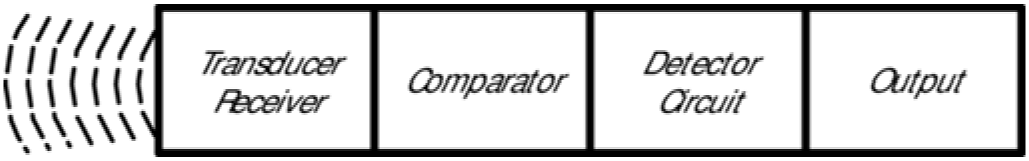


Figure 3.2.1.3 i: Ultrasonic Sensor Components Overview (approval pending)

3.2.2 Webcam

Another important part of the robot is a webcam. This will essentially be the “eyes” of the robot. By using a webcam, the robot will have eyes like that of our own. The video feed from the webcam will be able to be viewed in real-time. An object detection algorithm will then be applied to the video feed. From this, desired objects that are found will be highlighted on the screen. This information will then be used, along with the other sensor technologies, to determine if the object is another robot and whether that object should be shot with a nerf bullet or not.

Each webcam will have its own video resolution. If the video resolution is higher the objects will be easier to detect. If there is a clearer picture the algorithm will be more efficient in determining if the object is of interest. Since this is a very important aspect of this project, a good webcam will need to be procured.

3.2.2.1 Visible Spectrum Video Quality

Group 2 is responsible for ensuring that Green Teams robot has a camera that is capable of accurately targeting several targets during the completion. Lockheed Martin has specified that 2 targets at the opposite corners of the course will be comparable if not identical to a human face. The human head is 6 to 7 inches wide and 8 to 9 inches long however, Lockheed Martin stated on October 14th that the designated targets will be slightly larger at around 12 inches by 18 inches. These dimensions are important to note for Green team as they, along with the displacement, will be factored into the calculations for choosing a sufficient camera. The camera will also need to recognize the enemy team’s human medic at distances of up to 40 feet. Additionally, the enemy team’s robot will need to be detected and, although the max robot size is 3 feet by 3 feet, Green Team should be able to target a robot that might be much smaller.

Many factors can affect the quality of the image processed by a video camera including contrast ratios, lighting conditions, atmosphere, and optics however, for this specific project Green Team can ignore these variables since it is known that the completion will take place inside a large, well-lit, indoor location on UCFs main campus. In-other words, Lockheed Martin has not made it a point to test the groups on their ability to handle extreme lighting or atmospheric conditions. With such conditions ruled out, Green Team can address the most relevant criteria for choosing a well-performing camera as outlined by COHUHD:

- Object Size
- Camera Field of View
- Image Resolution

Green Team is informed of the various probable target sizes and their respective designations and therefore must choose a camera with the correct resolution level. If the camera does not have enough resolution to detect the pixels of the target, a great or even total loss in accuracy could occur. If the camera has too much resolution than that required of it, a larger amount of monetary funding and computational power could be unnecessarily spent since higher resolution cameras cost more and put greater demand on both the processor and video stream; so, either the system performance would decrease overall or even more money would be wasted on a higher performing computer that isn't needed.

There are three increasingly complex levels of object assessment: Detection, Recognition, and Identification. Green Team will not only need to be able to detect objects but recognize and distinguish them to correctly establish a reliable and accurate target lock without trying to target everything in the room. Per "The Johnson Criteria" which establishes the industry standard definition for Detection Recognition Identification, the camera will need to visually identify "8 vertical pixels on target" to accurately provide target recognition.

Field of view is another important component of a camera. The field of view is the area capturable by the camera as given by the title. The components of the camera that determine the field of view are the: lens, sensor format, and the cameras zoom. If the camera has a larger field of view, it will be able to see smaller objects. There are several tools available online for calculating and determining what field of view is required to locate targets at specific distances. In addition to Field of View, Camera Image Resolution is another specification that varies from camera to camera and can affect the accuracy of target recognition. Higher image resolution generally coincides with increased cost. Consumer grade cameras typically either have a resolution of 720p at 1280 x 720 pixel or 1080p at 1920 x 1080 pixels. A study by CohuHD.com found that it is more important to have better camera resolution than optical magnification capability. So, a camera with standard definition and 10x optical zoom will not be able to see objects as well as a high definition camera without zoom[16]. Given these findings, Green Team will look to procure a high definition camera since the robot will need to provide flexibility in constantly searching for close and long range targets.

3.2.3 Servo Motor

To move the position of the gun on the robot a motor will need to be used. This motor will pan and tilt the gun and its assembly so it can hit its desired target. A great type of motor to use will be a servo motor. A servo motor is small, but is very efficient. They are perfect for controlling the position of a nerf gun for this project. They are cost effective and are easy to incorporate.

They are made up of a few different parts. There is a DC motor, control circuit, and a potentiometer. The DC motor is attached by gears to the control wheel. When the wheel rotates, the potentiometers resistance changes. The control circuit can then control and regulate how much movement there should be. The shaft of the motor will dictate when the motor is in its desired position. When it is in its desired position, the power is cut off. If it is not at its desired position, then the motor is turned in the correct direction. This desired position is sent via electrical pulses through the signal wire[22].

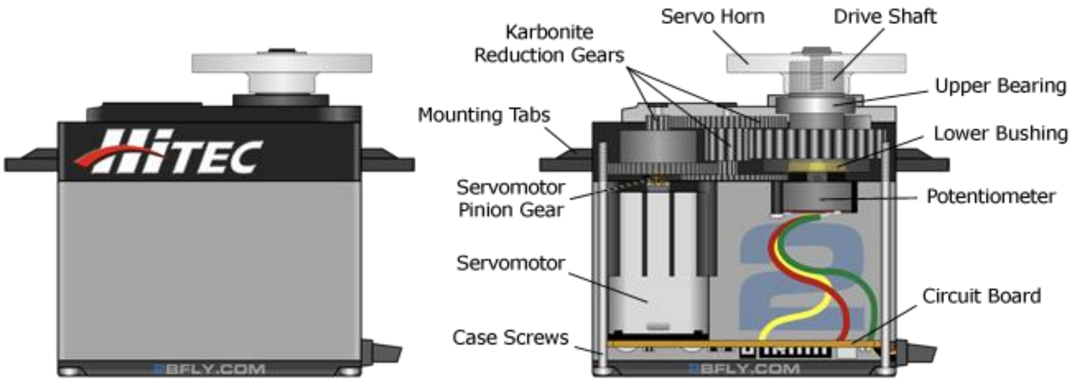


Figure 3.2.3 i: Internal Components of a Servo Motor

3.2.3.1 Pulse Width Modulation

The servo motor is controlled by sending pulse width modulation (PWM) through the control wire. Each servo will have a minimum and maximum pulse as well as a repetition rate. The neutral position of the motor is described as the position where the servo has the same amount of potential rotation the clockwise or counterclockwise direction. The PWM sent to the motor will determine the position of the shaft. Based on the duration of this pulse, the rotor will turn to the correct position. The length of the pulse determines how far the motor will turn.[22] Figure 3.2.3.1 i is a representation of how PWM works with the motor.

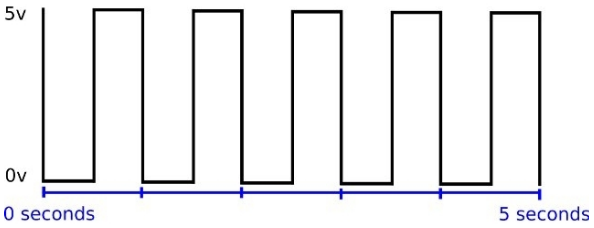


Figure 3.2.3.1 i: PWM with Control Servo Position

3.2.3.2 Analog VS Digital

Choosing the right servo to control the aim of the weapon will considerably affect the accuracy, reliability, and stamina of the robot. Regarding commercial RC servos, there are two different options: Analog and Digital. Ultimately, Analog and Digital make themselves useful for interchangeable applications. One could technically be used in place of the other since they ultimately provide the same functions and operations. Both types of servos even come in the same size and both servos interface control with the system by equivalent means. Digital and Analog servos both have a three-wire connection: one wire provides voltage, another is used as the ground, and the third wire communicates the control signals via the previously aforementioned pulse-width-modulation method. Analog and Digital servos are available in several sizes including micro, standard, and giant. Since the chosen pan and tilt servos will be required to support a relatively large NERF weapon weighing several pounds, 2 high-performance servos will accommodate the weapon to ensure stability and reliability. Digital and Analog Servos may come in water and dust resistant housings however, since Green Team will use the robot to compete in-doors, such specification will not be necessary. Additional disregard will be given regarding servo gear wear and tear. Digital servos have received some criticism for the excessive amount of stress that can be applied to their gears due to a low “Dead-band” tolerance. In other words: the servo will allow very little fluctuation in un-authorized movement before correcting itself. Regardless, for the specific applications called upon for this project, a high quality Digital servo from a respected brand should be able to at least withstand the required 3 rounds of completion.

The real difference between Digital and Analog servos is the technology inside them. The two types of servos process received signals differently than one another and this in turn provides distinctive results in terms of control and energy consumption. From the receiver to the motor, Analog servo control is executed after processing is conducted through several electronic components which is outlined in further detail in figure 3.2.3.2 i.

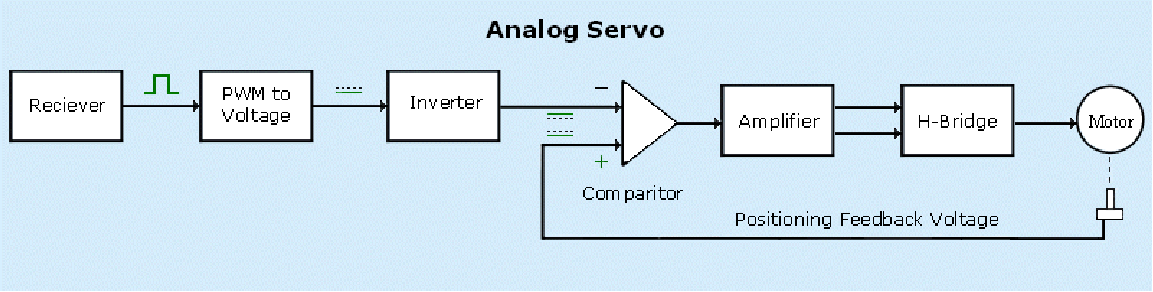


Figure 3.2.3.2 i: Analog Servo | Credit: Tony @ http://www.sailservo.co.uk/anvdig.html

With analog servos, constant on/off voltage signals are applied to the motor. The on/off frequency turns the motor and is proportional to the torque produced. Digital servos do not depend on archaic analog components. They instead have an integrated microprocessor inside the package to process the received signals pulses at a much higher frequency. The shorter-length Digital servo pulses compared to the longer Analog ones can be seen in figure 3.2.3.2 ii. With shorter pulses occurring at a much higher rate, the response time of Digital Servos dramatically outperforms Analog servos and near-constant torque is supplied.

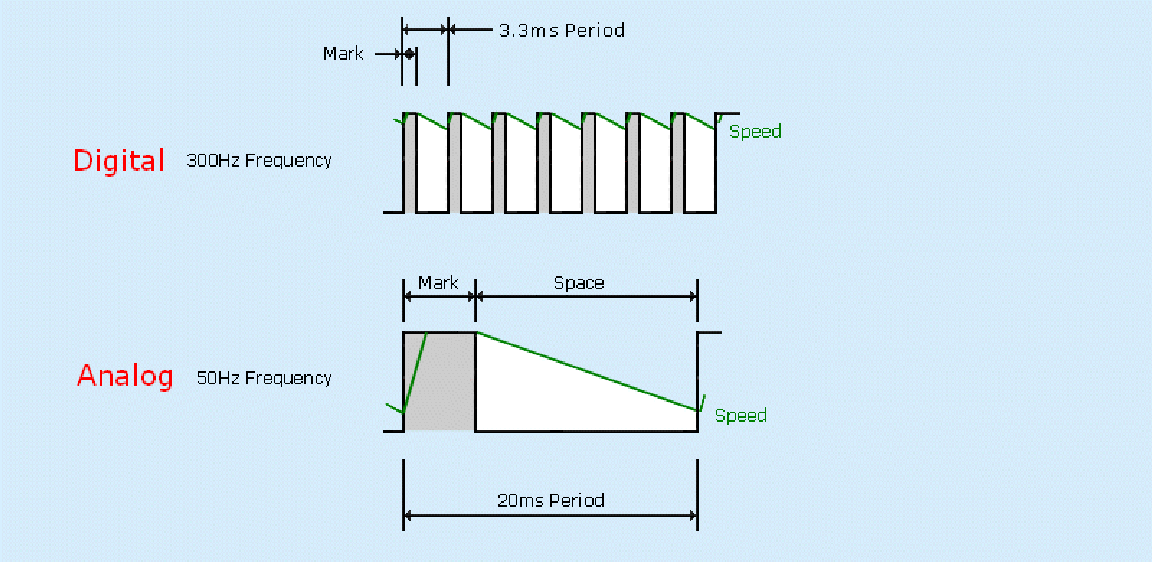


Figure 3.2.3.2 ii: Digital vs Analog | *Credit: Tony @ http://www.sailservo.co.uk/anvdig.html*

This means that a digital servo controlling the weapon would be able to react much quicker to a changing enemy position. Unfortunately, the decrease in response time also mean the Digital servos will consume more power than their Analog counterparts. This does not necessarily mean however, that Digital servos aren't more efficient at using power than analog servos. An analysis performed by David E. Buxton on July 25, 2014 (Figure 3.2.3.2 iii) shows how Digital servos "internally generated voltage is proportional to RPM."

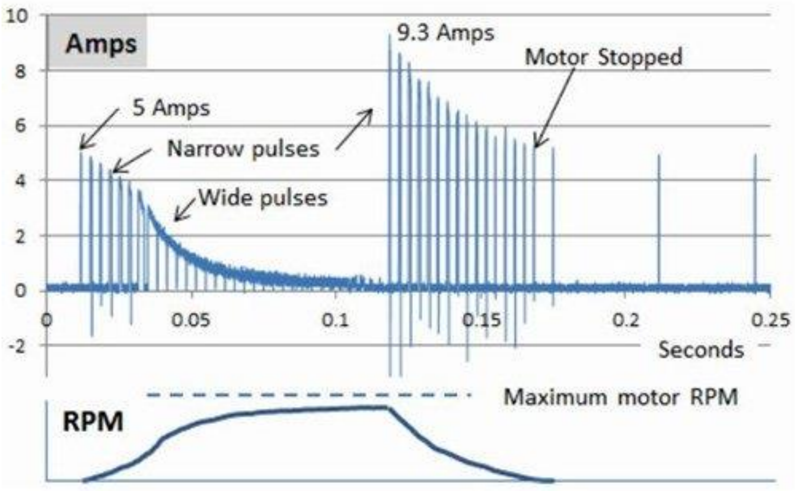


Figure 3.2.3.2 iii: Digital Servo Energy Performance Characteristics compared to RPM |
Credit: David E. Buxton @ http://www.rchelicopterfun.com/digital-servos.html

In his verification of Digital servo power consumption, Buxton concluded that Digital servos provide greater efficiency than analog servos. With respect to peak power consumption, Buxton also discovered that peak energy usage is achieved when the servo is either at full speed or when the motor completely reverses direction – a valuable observation for Green team to consider when choosing servos for the targeting system. Since much more importance will be weighed on the pan servo rapidly tracking targets left and right and the tilt servo will most likely be of much less use other than locating the stationary targets, a multi-faceted combination could be considered where the pan servo is digital and the tilt servo is analog.

3.2.4 Microcontroller: Hardware Integration

Perhaps the most integral part of this project is the microcontroller (MCU). This is the “brain” of the robot. Without this the robot, cannot “think”. This means that it will not be able to process the information from the webcam and sensors. If it does not process this information it will not be able to send signals to the servo motors and the trigger of the nerf gun.

So, what exactly is a microcontroller? It is basically a mini-computer. They are composed of a central processing unit (CPU), memory, and input/out devices. They are also low-power devices, so this allows them to be used in a myriad of applications.[23]

This device will be used to perform a few tasks for this project. It will need to be able to communicate with the program used to detect objects while analyzing sensor data, convert those communications into controlling the Nerf weapon and ultimately firing the Nerf weapon. It must also have enough inputs to take information in from all the hardware components. Along with the inputs, it needs to be able to send signals to the servo motors and the trigger of the nerf gun. A MCU will fulfill all the needs described. One must be chosen that is cost effect, efficient, and powerful enough to run the program created.

There are a multitude of microcontrollers available out there. Each one has its own advantages and disadvantages. The discussion of which microcontroller that will be used will be in a future section of this document.

3.2.5 Computer Architecture: RISC vs CISC

When investigation various options of small single board computers to use as the mobile processing unit for a project relying heavily on rapid image and senor decoding at peak energy efficiency, it is important to understand two classes of computer architecture: Reduced Instruction Set (RISC) and Complex Instruction Set (CISC).

It would require a specific research paper of its own to thoroughly investigate the two architectures so a reduction of such a case study (“A Tale of Two Processors: Revisiting the RISC-CISC Debate,” by Ciji Isen, Lizy John, and Eugene John) follows, cited from unpublished material written by group computer engineer: Alexander Perez in April 2016. This analysis will provoke Green Team / Group 2s effective selection of necessary computing hardware to run the custom autonomous target recognition software.

CISC and RISC both have a variety of shortcomings. RISC typically yields much larger instruction counts. The CISC Common Programming Interface (CPI) magnitude often surpasses RISC CPI. In “A Tale of Two Processors,” Pat Gelsginer stated the gap in performance between RISC and CISC is relatively insignificant and getting smaller. Many Instruction Set Architectures (ISAs) recently developed have been RISC but still, the dominant architecture used by Intel and Advanced Micro Devices (AMD) is the x86 CISC ISA. The performance gap between CISC and RISC is rapidly narrowing however, a majority consensus between experts still concludes that CISC will not surpass the performance of RISC. CISC ISA would require more area, energy, and delay to achieve equality to RISC in terms of raw power.[24]

A modern analysis of performance statistics was conducted by comparing an Intel CISC processor to an IBM RISC processor. Each processor was made with a unique microarchitecture even though both launched to market around the same time and have a similar number of transistors. The biggest difference between the Intel processor and the IBM processor was their memory hierarchy, where the CISC processor gained an advantage.

The IBM and Intel processors utilize several cores and data caches. The Intel CISC processor has 2 data caches while the IBM RISC processor leads with 3 data caches to assist address translation which begins searching with the Effective-to-real address translation (ERAT). In contrast, the Intel CISC processor does not have a Level 3 (L3) cache and branch prediction occurs within the instruction fetch unit. Some similarities include comparable clock frequencies, pipeline stages, memory bandwidth, and transistor figures. The Intel CISC processor out-specs the IBM RISC processor with a die size smaller by over 100mm^2.

The Standard Performance Evaluation Corporation (SPEC) CPU2006 Benchmark Suite was used to compare the chip performance of the IBM RISC processor and the Intel CISC processor in individually tailored compiling environments. The findings revealed both processors use several dedicated registers to count the number of transacted instructions and track unique performance events. The benchmark revealed that the Intel processor surpassed the instruction count of the IBM RISC processor nearly fifty percent of the time. This is due in part because, unlike the RISC processor, the CISC processor needs to convert instructions to more basic instructions called micro-ops or uops as shown in figures 3.2.5 i and 3.2.5 ii. A higher uops/instruction ratio results in more work exercised per instruction. Improvements in modern CISC processor design have made the uops/instruction ratio so close to 1 that the performance impact to the CISC chip is minimal.

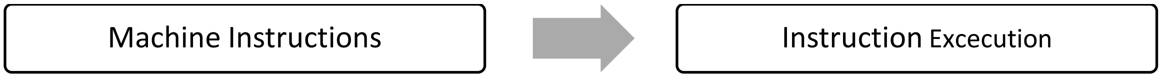


Figure 3.2.5 i: Reduced Instruction Set Architecture

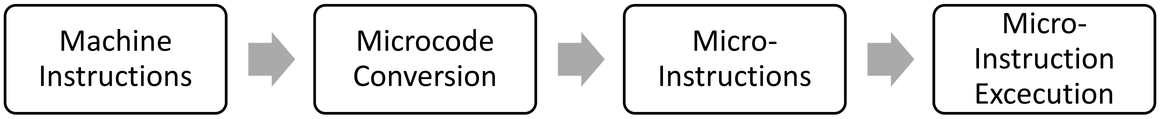


Figure 3.2.5 ii: Complex Instruction Set Architecture

The variety of instructions observed provided performance insight with regards to the branch predictor and cache but could not provide enough refinement to point out bottlenecks in the chip. Larger quantities of CISC binaries were present in comparison to the RISC processor. The fraction of branch instructions in C++ programs was lower in both RISC and CISC processors. Branch prediction ensures pipeline interruptions are avoided. Large pipeline malfunctions such as wastefully instruction executions in the misprediction path will occur if branch misprediction and misprediction penalty become frequent problems.

Regarding branch prediction, CISC leads the performance benchmark over RISC by an averaging 2% difference. Floating point programs notoriously corroborated the most branch prediction problems for the RISC processor.

Among the greater and more significant systems in the processor is the cache hierarchy. Of this cache hierarchy is was previously divulged that the RISC has a 3-tier cache. Compared with the CISC, the RISC L2 cache is functionally smaller yet it has a larger shared L3 cache to compensate for this size difference.

A major factor in providing enhanced capabilities, speculation techniques have recently narrowed concentration in the fields of “control flow” and “memory disambiguation.” Speculation frequencies are presently higher in the studies CISC processor. Specifically, over 40% of CISC instructions are speculated whereas the RISC speculation shorts this by 11% fewer speculations – a 27.5% advantage. Taming the amount of performed speculation is a top priority since this concerns critical areas of energy usage and power. The study conclusively defines RISC and CISC processors as yielding almost negligible difference when “micro architectural” techniques are utilized.[25]

3.2.6 Operating Systems

Two operating systems were considered for the on-board computer of the robot. The Linux-based operating system: Raspbian was compared functionally to the Windows 10 operating system. Performance-wise, each operating system contributes a unique advantage but only one showed a much wider range of versatility and reliability.

3.2.6.1 Raspbian

The Raspbian operating system is Linux-based. This operating system was developed as a simple and user-friendly counterpart for the educational Raspberry PI single board computer. In terms of features, for a free operating system, Raspbian provides a great deal of useful features including several pre-packaged interactive development environments. The drawbacks of this operating system are that it is not widely used in the comparative scope that Windows or even Mac OS are. The amount of open-source material and software that are available on Windows greatly outnumbers that of Raspbian.

3.2.6.2 Windows 10

Windows 10, though occupying a greater amount of storage space and requiring more processing power to run, is a highly preferable choice to Raspbian for any serious project or comparable task. The operating system is not without its own faults, namely unpredictable updates, memory leaks, and bootloader failures. Such drawbacks are not common however. Windows 10 is widely available on an enormous amount of systems. Sticking with windows 10 allows more flexibility when choosing both hardware and software. Compiling code using Raspbian is not ideal for that reason. It is simply easier and more efficient to use Windows. Additionally, the Arduino environment that Green Team / Group 2 chose to use in servo controlling for example, is much friendlier to a Windows platform. Windows 10 also has built in wireless control functionality through its Remote Desktop feature. Such a feature allows for much simpler reduction in logistical planning with regards to controlled the software remotely. With remote desktop, a user only needs to sign in using another device on the same wireless secured network, and they instantly have a reliable connection to command said computer remotely.

3.2.7 Computer Vision

Computer Vision gives a computer the ability to visualize the world given several available parameters. The way humans see the world and the way computers see the world are not similar in any respect. Humans see the world as defined by experience, knowledge learned, feelings/emotions, and so on. Computers see things based on how they are programed to. The programmer defines for the computer how to see and recognize this. The programmer tells the computer that there are gray areas for a computer to recognize an object.

Human beings receive input from all their surroundings and extract information naturally in an instinctual manner. Furthermore, objects are “pushed” away while others are brought forward as important to see based on a given mindset and the surroundings. For a computer, in contrast, everything is equal and has no meaning to start. Any image the computer receives from a camera or video is full of ambiguous information, which is overwhelming to analyze and understand in the original state it is received in.

In World War II, camouflage was a major obstacle in finding enemies and their equipment. For people with full-color vision it is difficult to identify concealed objects. With that regard, color blind individuals were employed during World War II to assist in locating said concealed objects. People whom can see color are strained by objects that are camouflaged, but color blind people are not affected by such camouflaged emplacements based on color. Color blind people are unaffected by the idea of camouflage simply by the removal of color recognition.

Like color blind people, when analyzing video and images, when a computer is searching for an object/image not pertaining to color, it converts the image to grayscale to simplify the analytic procedure. Many additional techniques are available to conducting such processing but this color removal technique usually sets things in motion.[18]

3.2.8 Facial and Object Detection

Simple object detection vital to fulfilling the requirements outlined by Lockheed Martin since it supports the capability of the robot to autonomously track targets – a primary function. For a computer to detect a specific object, it must be programmed and trained by using a collection of images called a “template.” As soon as the template is accessed by the program, the computers database can commence learning what objects are and are not acceptable. The problem with this method is that certain algorithms could cause the robot to target the wrong object. These types of algorithms are not 100% reliable and only work once the requirements are met.

One type of object detection category is facial detection. Facial detection is the process of detecting faces in any given image. Facial detection is useful because the robot needs to be able to target the enemy medic upon entering the battlefield. Facial recognition will also be useful for locking onto the stationary targets. The stationary targets, as specified by Lockheed Martin, will have enlarged facial profiles. Like object detection, facial detection algorithms can cause problems when defining what should or should not be targeted. The face in the database must have the same pose as the face that is targeted or it could possibly not work correctly. Furthermore, if the input face has any sort of alternative rotation or differs in lighting, this could also cause problems. Even more concerning is the facial recognition algorithm might detect faces where there are none.[18]

3.2.8.1 AdaBoost/Viola-Jones Algorithm

AdaBoosting, short for Adaptive Boosting, is a learning algorithm which selects week “experts” to be pooled together to collectively be better than a single standalone “expert.” From the title, Adaptive explains that once a weak expert is selected, the next expert is chosen based on how compatible it is with the other selected experts. AdaBoosting is famous for being used in the Viola-Jones algorithm. The Viola-Jones algorithm trains the computer to differentiate what is or is-not a face.

A form of object recognition, the Viola-Jones algorithm applies a group of patterns to a group of images to conclude what is or is-not the desired object to be tracked.

The Viola-Jones algorithm is reliable but some flaws could greatly affect the object detecting process. The Viola-Jones algorithms flaws are not very noticeable. Once disadvantage of the Viola-Jones algorithm is that it can only detect an object that appears in a 100% relatable way. If the program was shown a face appearing at an angle or an area with greater or less lighting that the test examples, that face will not be detected. The Viola-Jones algorithm can also detect an object more than once which is not necessarily a good thing.[18]

3.2.8.2 Eigenfaces

Eigenfaces are a type of facial detection. Eigen Faces takes a selection of faces and then changes facial image into what's called an "eigenvector." An eigenvector is a vector that can be converted into a scalar value upon assignment of a value called an eigenvalue. Like AdaBoosting, the Eigenface algorithm requires a group of images for targeting purposes. Continuing its similarities with AdaBoosting, the Eigenface algorithm is hampered by its inability to detect faces that are not the same as the faces used in the template. Eigenfaces biggest drawback is its exclusive usefulness as a facial detection tool. For general object detection purposes this poses a problem but generally it's not an issue since the algorithm was purpose-built for facial recognition. The Eigenface algorithm is based off a general algorithm: The Primary/Principal Component Analysis (PCA). PCA uses a similar template that holds a random selection of objects and then differentiates what is or is not acceptable as a target. Both algorithms run in the same manner. The only difference between the two is what is used for detection.[18]

3.2.8.3 Fisherfaces

Fisherfaces is another form of facial detection. Fisherfaces focus on discerning what is or is not an acceptable object to target. Fisherfaces does this by checking for similarities between the image that is passed to the algorithm and the template used for creating the program.

To show the differences, the eigenvectors that are formed from this process, which is comparable to the input and the testing images are put side by side in an eigen space. Meanwhile, the eigenvectors that are different between the images are placed farther apart in the same eigen space. This is different from eigenface because eigenfaces use the largest eigenvalues associated with the eigenvectors of the image, which is better to use for finding something that represents the testing images. Fisherfaces are used to find objects that are classified as comparable to the testing images used to train the algorithm to recognize the images. It should be noted that, comparable to the eigenfaces that we talked about earlier are based off PCA, Fisherfaces are based off the Linear Discrimination Analysis (LDA) algorithm. The LDA sets the foundation for how the Fisherfaces algorithm determines what is and is not an acceptable target. It is important to note that the Fisherfaces algorithm cannot detect non-faces by itself. If we are to use a program that is similar, it might be best to use the more general LDA. This way we would be able to recognize non-face imagery. This is important for recognizing the barriers.[18]

3.2.8.4 How Humans Detect Faces

Here Green Team will talk about how facial recognition is achieved by a human being. This may seem trivial, but it is important to discuss and point out the obvious which is done by oneself to transform the innate knowledge into something that can be generated as concrete and understandable for a computer algorithm to detect a face.

For a computer to recognize a face, an effective algorithm needs to be written. To do that, we must investigate how the human brain recognizes a face to be a face. The components of facial recognition are a natural human instinct and inconceivable for a computer without proper coding. To program an algorithm designed to recognize a human face, we must identify the parts of a face that makes it a face, like the pieces of a puzzle, and form the picture with these indicators. As humans, there are key features that make a human face. Faces are comprised of a pair of eyes horizontally aligned at an inch or two apart in length, a nose located below the horizontal line of the eyes, below the nose a mouth which may consist of many different forms such as being closed and straight, closed and concave up, closed and concave down, and open for every type of ellipsoid shape between a straight line and the circular shape the mouth make when fully open. Other important features include the cheeks, chin, jaw, eyebrows, forehead, hair, and ears. These characteristics of a human face very in size and shape allowing for the common identification of someone being human as well as the ability for someone to have their own individuality.

This ability to identify something as a face, or what seems to resemble a face, is an innate ability that we are born with and therefore we do not have to be explicitly taught it. But, this does not mean that we do not need to be taught, it just means that we have general idea or preconception of what a face looks like. Researchers say that we, as infants, have a certain time range for which we need visual input to fully perceive faces correctly. Otherwise, we could still identify faces, but we might have difficulty in the ability to differentiate faces from one another.[18]

3.2.8.5 How Computers Detect Faces

From the references described above, group 2 will discuss how computers detect faces and how those methods differ from that of a human. A computer cannot simply learn how to distinguish faces without any information provided. A computer must be given a sample of example human faces to begin the process to differentiate faces from one another. A process that can be used to accomplish this is a process called, boosting. Boosting is a process where a group of weak experts will be trained with the end goal of them becoming strong experts. Through this process, the weak and strong experts will be used to create acceptable targets that will allow the robot to aim and shoot at those targets. This can mostly be accomplished by using the AdaBoost algorithm.

The experts first must be programmed to detect various features of the human to begin to make the distinction of one another. The experts would need to be able to accurately identify the ears, mouth, eyes, nose and the actual head of the face. For this process, each of the experts will be given a specific shape to begin looking for those specific features. These specific shapes are used at the weak experts and this is referred to as Haar-like features. Through a process called cascading, these experts will go through the images that was originally provided. Cascading is what will allow each of the experts to determine what is not a face in the input. When there is a successful detection, all the remaining faces will move onto the next expert. Meanwhile, all the non-faces that are found will be discarded. After the face, has gone through and passed all the Haar-like features, it will officially be considered as a potential face. The same process is then repeated until the algorithm reaches the end of the image.[18]

3.2.8.6 Facial Detection with OpenCV

The Face Recognizer class from the OpenCV library is one part of the library we plan on using to help with this project. This class offers the following methods: train, update, predict, save, and load.

The training method is what allows our robot to discern what is and is not a target. To do this, it needs an example of what we want the robot to target. In this situation, it is given a database of images that will be ingrained into its “memory”. Armed with this knowledge, the robot can now properly lock onto and shoot predetermined targets that appear in a visual output. An example of this would be giving the algorithm a database of objects that include multiple stop signs at different angles. In this case, the algorithm should theoretically target any stop signs in each image.

The update method is used to, for lack of a better word, update the Face Recognizer if it is supported by the algorithm. In the case of the Local Binary Patterns Histogram program, it would be able to use the update method. A problem occurs if you try to use this method with the Eigenface or the Fisherface algorithms because it is not possible for them to utilize this method. To combat this problem, a workaround can be achieved by using the train method. By using this method again, it empties the current model and learns a new one.

In terms of the other main methods in this class, the predict method can preemptively detect a given label and its associated confidence.

The save and load methods can save an instance of a Face Recognizer class and its corresponding model state in either an XML file or a YAML file.[18]

3.2.8.7 Motion Detection

This project heavily relies on identifying and tracking the movement of objects in the field of vision. One of the main focuses for this project is motion detection.

Motion detection uses a fundamental idea of real-time segmentation of moving regions in an image sequence. This idea is critical in vision systems such as visual surveillance, human-machine interface, etc. One of the most common methods of motion detection is background subtraction.

The idea of background subtraction involves using a reference image. This reference image is an image that is taken before any movement has occurred inside the frame. It is usually taken right as the program starts, or prior to the start of the program during the setup process of the equipment. Background subtraction calculates the reference image and subtracts each new frame from it. The result is a binary segmentation of the image which highlights regions of non-stationary objects within the frame. This image is known as the threshold image.

Background subtraction, although being simple and quite effective, has inherent problems. The algorithm cannot distinguish between shadows and actual movements. It also suffers when there are gradual changes in the lighting conditions in the scene. Adaptive algorithms for background subtraction have been developed to compensate for very small changes in the reference image. For the purposes of this project, where there is a non-static camera with an ever-changing reference image, simple motion detection algorithms will not work.[18]

3.2.8.8 Object Tracking

Object tracking is probably one of the simplest tasks that a human being does in their day to day activity, but for a computer (robot) it is a very difficult job. The robot must identify an object and must be able to track it if the object is in its field of vision. For a human brain the job is easy enough that we look for that specific object and simply watch it move. However, a robot simply does not know whether the object exists in the frame as it moves on from the previous one. This is where feature detection comes into play.

Feature detection is a concept where a robot would analyze a frame and look for specific patterns or features that are unique to the object which needs to be tracked. An object that is to be tracked needs to have features that can easily be compared. A few examples would be a large ‘X’ mark, distinct colors, or certain shapes. This allows the algorithm to distinguish between different objects within the field of vision and assert that object as one to be tracked.

For this project a large ‘X’ mark is not going to be placed onto the opposing robot, so tracking features like feature detection would not suffice. So, other methods need to be employed to identify and successfully track opposing robots.

One of the key components of object detection and tracking is edge detection. Edge detection is implemented by looking for regions with maximum variations in pixel components when moved by a small amount, in an arbitrary direction. In other words, all objects have edges that can be separated from the background in ideal lighting conditions. The algorithm then uses this idea to separate edges and single out objects. These objects can then be used with facial detection and feature detection to identify targets.

One of the most crucial parts to this project is being able to track the objects that our robot has detected while it is in motion. One of the biggest problems with this situation is that our robot is almost always moving. This coupled with the fact that not only are two of the main targets (the enemy robot and the enemy medic) are constantly in motion, but there are multiple targets that the robot must be able to keep track of to be able to target the correct objects efficiently. This is not the only problem, as we will not know what the enemy robot will look like. To counteract this, we will use motion detection.

Using motion detection is what will allow our robot to be able to figure out where the enemy robot is without knowing what it looks like. Another silver lining is that two of the targets that our robot needs to detect are not only stationary, but they also can only be shot once. This means that our robot will only have to target one to two different targets on average and four targets at most. This allows us to focus on prioritizing which target to shoot at. To do this, we plan on equipping our robot with an ultrasonic sensor to figure out how far certain objects are. With this parameter being read in, we can have the robot prioritize targets by having it shoot at the target that is not only closest, but is also not obstructed by one of the two obstacles on the arena.[18]

3.2.9 Android Studio

One possible method for Green Team / Group 2 to control the robot and targeting system would be through an application through a device such as a tablet or a mobile phone. Android Studio is a good way of executing such a design. The application built using Android Studio would be compatible with a phone or a tablet. The application made through Android Studio could control the robot in addition to providing the user with visual feedback. The feedback would contain various information about the environment and possible targets while it is manually driven by a user. This idea is stretching the time Green Team / Group 2 has for the project. It would take a lot to build a whole Android application from scratch when the system could simply be controlled via remote desktop. The effort, if done, would need to be a collaboration between all team members including computer science and computer engineering. If it is found that Green Team / Group 2 is struggling for time in the project, a regular remote controller will be used to manually drive the robot and the remote desktop application will provide wireless oversight of the targeting software.[18]

3.2.10 Bluetooth

One of the main things to consider is if it will be feasible to work with Bluetooth on the mobile application as mentioned in the above section. If time is available to use Android Studio, the mobile application should be able to not only allow the user to move the robot, it should also be able to provide the user with a visual feed of what the robot is seeing in real time. If we use the external controller, we would have to provide the visual feed using an external streaming service such as gStreamer or MJPGStreamer. While that may be the case, the external controller will only have the joystick and will also be much easier to link up with the rest of the robot. It will ultimately come down to if team will have enough time to implement the mobile application. If it seems that it is too much work, then more than likely the team will utilize an external controller device to allow the robot to move.

The android platform supports the Bluetooth network stack, which allows wireless communications between devices. Every Bluetooth activated device has a Bluetooth adapter built into it. This is the base of operations for all Bluetooth related activities. Using the Bluetooth adapter, the device can search for other devices, or instantiate a Bluetooth Device or a Bluetooth Server Socket. Once instantiated a Bluetooth device can then request connection through Bluetooth Socket or query information about another connected device. The Bluetooth socket is the connection point on a Bluetooth device. This allows application to exchange data with another device via Input Stream or Output Stream. Every Bluetooth enabled device also has a Bluetooth Profile on it. This is a wireless interface specification for Bluetooth based communication between device.

The Intel Stick is an already Bluetooth enabled device which is a convenient because it eliminates the need for an external Bluetooth dongle to be connected. An android device can connect to it and control commands can be passed into it.[18]

3.2.11 Arduino

An Arduino board is used to build digital interactive devices. It uses standard connectors, which lets the Arduino connect to a wide variety of modules called shields. Most of these shields connect to the Arduino using various pins and the CPU chip on board the Arduino can use these pins to access the shields.

The Arduino has both digital inputs and analog pins. These analog pins can be used to read in a range of different sensors. These sensors can be read in more accurately as analog signal does not just read binary, it measures the change in voltage. While the digital pins allow the board to control and communicate other interfaces. The Arduino controller can connect using Universal Serial Bus (USB) cable. The script runs on board the receiver computer, and would Serial communication to “talk” to the Arduino. By “talk”, it means the computer would be able to relay data back and forth with the Arduino. The serial communication also allows the script running C++ code to read in inputs from the sensors and send commands to the servo motors. Arduino is fully compatible with Windows. Bluetooth signals sent by a user can be translated into Arduino code, which can send commands to the motors via the board. The Arduino will also read in inputs from the ultrasonic sensor and send it back to the intel stick. The intel stick can then process the information to find the trajectory needed to fire at the target, and will signals through the Arduino once again to fire the Nerf projectile.[15]

3.3 STRATEGIC COMPONENTS: INVESTIGATIONS AND SELECTIONS

This section will describe the different parts that will be needed to realize the project. Each of these parts will be compared with similar parts of the same category. After this comparison, a part will be selected based on what it provides in terms of needs to the project.

3.3.1 Infrared Sensor

Infrared sensors may be used with this project, and they will be discussed further here.

One of the possible IR sensors that may be used is the Sharp GP2Y0A21YK. This sensor is not very expensive, but it does not have the greatest of ranges.

Another option would be the Sharp GP2Y0A60SZ0F IR Sensor. This sensor is better than the GP2Y0A21YK previously discussed. Figure 3.3.1 i shows a picture of the sensor.

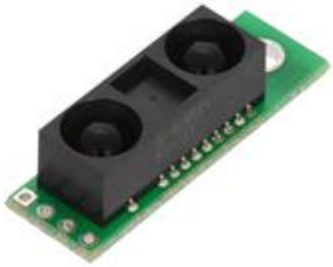


Figure 3.3.1 i: The GP2Y0A60SZ0F IR Sensor

The next option that can be used is the Sharp GP2Y0A02YK0F IR Sensor. This sensor is better than the previously discussed IR sensors. Some specifications of this sensor are included in table 3.3.1 i.

The last option is the Sharp GP2Y0A710K0F IR Sensor. This is the best IR sensor out of all the sensors previously discussed. This is also the most expensive sensor, but it will be the most effective type of IR sensor to use if chosen. Figure 3.3.1 ii shows a picture of the sensor.

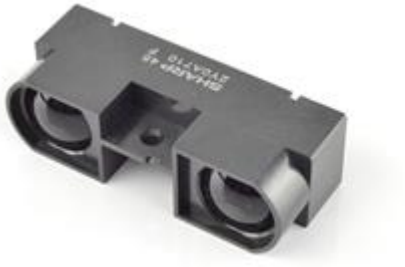


Figure 3.3.1 ii – The Sharp GP2Y0A710K0F IR Sensor

Each of these sensors come with its own set of specifications. Table 3.3.1 i compares each of the four infrared sensors.

COMPONENT	GP2Y0A21YK	GP2Y0A60SZ0F
Average Current Consumption	30 mA	33 mA
Operating Supply Voltage	4.5 to 5.5 V	4.5 to 5.5 V
Detection Area Diameter (at 80cm)	12 cm	-
Minimum Range	10 cm	10 cm
Maximum Range	80 cm	150 cm
Price	$9.95	$11.95
COMPONENT	GP2Y0A02YK0F	GP2Y0A710K0F
Average Current Consumption	33 mA	30 mA
Operating Supply Voltage	4.5 to 5.5 V	4.5 to 5.5 V
Minimum Range	20 cm	100 cm
Maximum Range	150 cm	550 cm
Price	$12.95	$17.78

Table 3.3.1 i: Comparing the Infrared sensors from datasheet specifications[26] [27] [28] [29]

3.3.2 Ultrasonic Sensor

Another type of sensor that may be used for this project is ultrasonic. This sensor is more expensive than the IR sensor, but it is better suited for the type of distance measuring needed for this project. The first and cheapest option is the PING)))™ Ultrasonic Distance Sensor. Figure 3.3.2 i shows an image of this sensor and the important specifications for it.

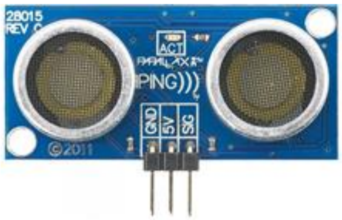


Figure 3.3.2 i: The PING)))™ Ultrasonic Distance Sensor

The next option is the Davantech SRF10 Ultrasonic Range Finder. Another option is the DFRobot URM04 v2.0 Ultrasonic Sensor. The last option is the Devantech SRF08 Ultrasonic Range Finder. Figure 3.3.3 ii shows an image of the sensor.

Each of these four ultrasonic sensors will be compared. From this comparison, a part will be selected. Table 3.3.2 i shows the comparison of these components.

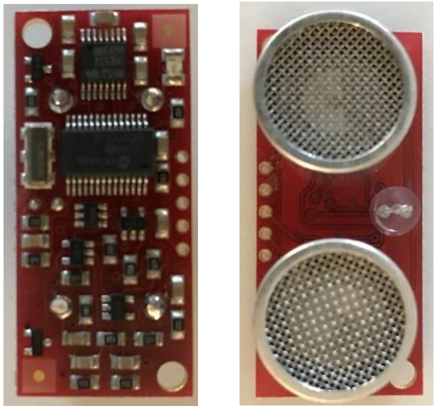


Figure 3.3.3 ii – The Devantech SRF08 Ultrasonic Sensor (front and back)

After considering all available options for a sensor, the Davantech SRF08 Ultrasonic Range Finder was chosen in Fall 2016.

COMPONENT	PING)))™	SRF10
Supply Current	30 mA typ. 35 mA max	15 mA typ. 3 mA stand-by
Operating Supply Voltage	5 V	5 V
Minimum Range	2 cm	3 cm
Maximum Range	3 m	6 m
Input Trigger	2 µs min; 5 µs typ.	-
Delay	200 µs	-
Price	$29.99	$33.68
COMPONENT	URM04 v2.0	SRF08
Supply Current	< 20mA	12 mA typ. 3 mA stand-by
Operating Supply Voltage	5 V	5 V
Minimum Range	4 cm	3 cm
Maximum Range	500 cm	6 m
Frequency	40 kHz	40 kHz
Resolution	1 cm	-
Price	$25.90	$49.00

Table 3.3.2 IV: Comparing Ultrasonic sensors with Datasheet Specifications[30] [31] [32] [33]

This sensor, while the costliest, is the best sensor for this project. The range can also be increased to 11m which is what is needed for this project to succeed. This sensor is also very easy to connect to a microcontroller, and its distance output can easily be read. Its beam pattern, while not as narrow as we would like, is enough to cover the area needed to gather an accurate reading.

3.3.2.1 LIDAR Sensor

Upon further research, it was determined that a LIDAR sensor would be another option to investigate.

One of these sensors is the LIDAR-Lite v3. This sensor, although expensive, has the greatest range detection out of all the other sensor technologies. Figure 3.3.1.1 i shows a picture of the sensor.

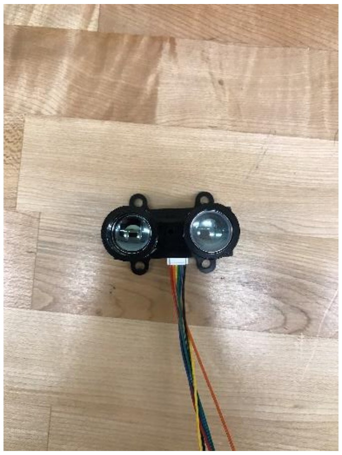


Figure 3.3.2.1 i: The LIDAR-Lite v3 Sensor

The specifications of the LIDAR sensor are shown in Table 3.3.1.1 i.

COMPONENT	LIDAR-Lite v3
Average Current Consumption	130 mA
Operating Supply Voltage	4.75 to 6 V
Minimum Range	5 cm
Maximum Range	40 m
Price	$149.99

Table 3.3.2.1 i: LIDAR-Lite v3 specifications[43]

After further investigation, the ultrasonic sensor did not fit the specifications required for this project. It did not provide enough range to be helpful in the project. For this reason, the LIDAR-Lite v3 replaced the sensor. Although the costliest, this sensor provided more than enough range that would benefit the project.

3.3.3 Servo

Two different servos will need to be used for this project. There are two different options, digital and analog. Each of these types of servos have various models that may be used. A few of these models will be discussed in this next section.

3.3.3.1 Digital Servo

We have a few different types of digital servos at our disposal. There will be two featured digital servos. The first digital servo is the Hitec RCD HS-5625MG. This servo is a high-speed servo motor, so it is perfect for the application needed for this project. The next digital servo is Hitec HS-5645MG. Figure 3.3.3.1 i shows an image of this servo. These two components are shown in Table 3.3.3.1 i.

Figure 3.3.3.1 i: Picture of Hitec HS-5645MG (Awaiting Approval)

COMPONENT	RCD HS-5625MG	HS-5645MG
Bearing Type	Dual Ball Bearing	Dual Ball Bearing
Speed (4.8V)	0.17	0.23
Speed (6.0V)	0.14	0.18
Torque (4.8V)	7.9 kg/cm	10.3 kg/cm
Torque (6.0V)	9.4 kg/cm	12.1 kg/cm
Size	1.59 x 0.77 x 1.48 in	1.59 x 0.77 x 1.48 in
Price	$39.34	$40.24

Table 3.3.3.1 i: Comparing the two digital servos[34] [35]

One of these servos would be used for the pan of the nerf gun assembly, and the other would be use for the tilt of the nerf gun assembly. The Hitec RCD HS-5625MG would be used for panning because of its speed. The Hitec HS-5645MG would be used for the tilt because of its strength.

3.3.3.2 Analog Servo

Another type of servo that may be used for this project is an analog servo. The internal components remain the same, but these ones do not have a small microprocessor inside. They also consume less power than their digital counterpart.

The first analog servo to be considered is the Hitec HS-805BB Mega Giant Scale 2BB Servo. This servo is known as the "monster" servo because of its heavy-duty internals and its high torque output. Figure 3.3.3.2 i shows an image of this servo. The next analog servo that is going to be discussed is the Hitec HS-645MG High Torque 2BB Metal Gear Servo. These two servos are compared in Table 3.3.3.2 i. After careful consideration, it has been decided that the two analog servos will be used for this project.

Figure 3.3.3.2 i: Picture of the Hitec HS-805BB (Awaiting Approval)

COMPONENT	HS-805BB	HS-645MG
Bearing Type	Dual Ball Bearing	Dual Ball Bearing
Speed (4.8V)	0.19	0.24
Speed (6.0V)	0.14	0.20
Torque (4.8V)	19.8 kg/cm	7.7 kg/cm
Torque (6.0V)	24.7 kg/cm	9.6 kg/cm
Size	2.59 x 1.18 x 2.26 in	1.59 x 0.77 x 1.48 in
Price	$38.99	$49.99

Table 3.3.3.2 i: Comparing the two analog servos[36] [37]

The Hitec HS-645MG will be used for the pan, and the Hitec HS-805BB will be used for the tilt. Analog servos are a bit easier to implement in this project. They are also a bit cheaper. The pan servo was chosen because of its speed. Group 2 / Green Team will need a servo that can pan the nerf gun quickly. The tilt servo was chosen because of its strength. The servo will need to be able to tilt the nerf gun up and down, and this servo is perfect for this application.[17]

3.3.4 Microcontroller Boards

Table 3.3.4 i: shows a comparison of the microcontroller boards. The Arduino IEIK UNO ATmega328P shown in Figure 3.3.4 i, was the first board Group 2 / Green Team knew would help us achieve our goal of creating an autonomous weapon system. Though the computer has a rather low clock speed, Group 2 decided that a faster clock would not provide us much of an advantage with our product. The main functionality of the ATmega328P will be for our firing system. Group 2 will use this computer to process data sent to it regarding the location of the target. When the ATmega328P receives this data, it will then send signals to the servos to position the weapons and then fire when ready. The software in this computer will need to be efficient due to its lack of memory and slow clock speed. The ATmega328P will require an on-board voltage source between 7 and 12 volts.

	101	**Uno**
Analog Inputs	6	14
I/O Pins	14	6
Clock Speed	32 MHz	16 MHz
Size (mm)	68.6 x 53.4	4.7 x 3.3
SRAM	2 KB	32 KB

Table 3.3.4 i: Arduino 101 vs IEIK UNO

The Arduino 101 was one of the last contenders to be our on-board controller. This controller offered a couple features that Group 2 found intriguing for our system.

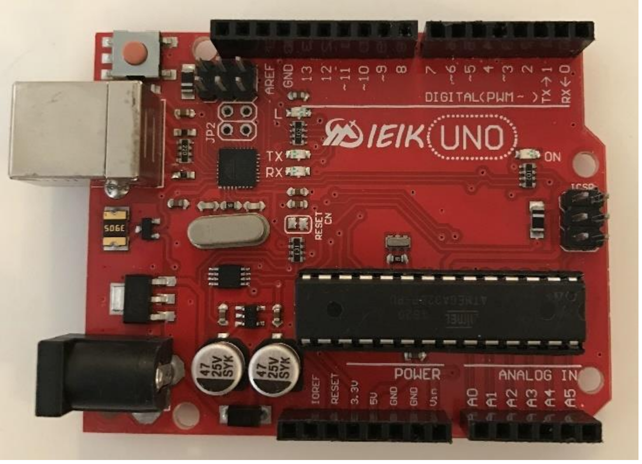


Figure 3.3.4 i: IEIK UNO

The Arduino 101 has an accelerometer and a gyroscope. Initially, Group 2 thought that this would be a huge benefit when calculating where to position our weapons. After discussion, Group 2 realized realize that this may be useless information, since our robot would not be moving quickly. Group 2 believes believe that if Group 2 added this, it would require more extensive software testing and it would not provide us with a significant increase in accuracy. The 6 analog inputs and 14 I/O pins show that the 101 is more than capable to support most of our external devices. However, with only 1 USB, Group 2 decided to look elsewhere.

3.3.5 Computer

For our autonomous weapon system to function, Group 2 needs need a processing unit on the robot. This processing unit, or computer, will be responsible for supporting the user video stream, target acquisition, and moving the servos to position the mounted Nerf weapons. The computer will send the camera stream to an application for the user to see what the robot sees. Apart from giving a view to the user, the video provides the computer with graphical imagery to detect faces, or targets. Once the computer can determine the location of the target, it must relay the positioning to the servos to align the Nerf gun with the target. There are four computers Group 2 considered to place on-board the robot - the Raspberry Pi, Panda Latte, Arduino, and Intel Compute Stick.

Note: Research shows that all computers mentioned fall into the range of approximately $100. Since all computers are roughly the same price, the cost will be left out of determining the best computer for us to use.

3.3.5.1 Raspberry PI VS Latte Panda

When we began, the Raspberry Pi was the first computer we considered to put on board the robot. The Pi was the most well-known and provide the most compatibility. Along with the specs, the Pi offers Bluetooth and wireless LAN. These features are important for syncing with the user application. Also, the Pi comes with a Micro SD card and card slot. This port allows for us to use more memory, though it will not be likely needed. The Pi is powered by a 64-bit quad-core ARMv8. This was another attractive feature of the Pi. If we choose this processor, this ensured that we would be using the most up-to-date hardware.[38]

The panda latte also comes with Bluetooth and Wi-Fi compatibility. It also satisfies most of the requirements we are interested in for our computer. The only issue was the lack of ports for connecting other components. This is the main reason we decided to stay away from this model. The Raspberry PI 3 Model B is compared to the Latte Panda in table 3.3.5.1 i.

	Raspberry PI 3 Model B	Latte Panda
Cores	4	4
Clock Speed	1.2 GHz	1.8 GHz
RAM	1 GB	2-4 GB
Ports	8	2

Table 3.3.5.1 i: Raspberry PI 3 Model B vs Latte Panda

3.3.5.2 Intel Compute Stick STK1A32SC*

The Intel Compute Stick, shown in Figure 3.3.5.2 i, is 118 x 38 mm in size. The computer is comprised of five I/O ports. It has a standard HDMI output and three USB ports. The USB ports are comprised of a Micro USB power port, a USB 2.0 part, and a USB 3.0 port. In addition, the Intel Stick has a Micro SD card slot on its side. Bluetooth 4.0 is also a compatibility featured with the Intel Compute Stick. This pocket-sized-quad-core computer comes stock with 2GB RAM, along with 32GB of storage and a clock running at a speed of 1.44GHz. A plus is the computer's portability. This would easily fit on top of our robot to control our system. It is also powerful enough to process our targeting algorithms and video stream. Though it only has 2GB of RAM, this is still achievable from its power and Bluetooth compatibility. Another issue is that the Intel Stick will need its own power supply. It appears that the AC power supply will need to be included on the robot. The Intel stick is another computer that we plan to purchase. The purpose of the stick will be for the software to detect targets and to relay the video stream to a user application.

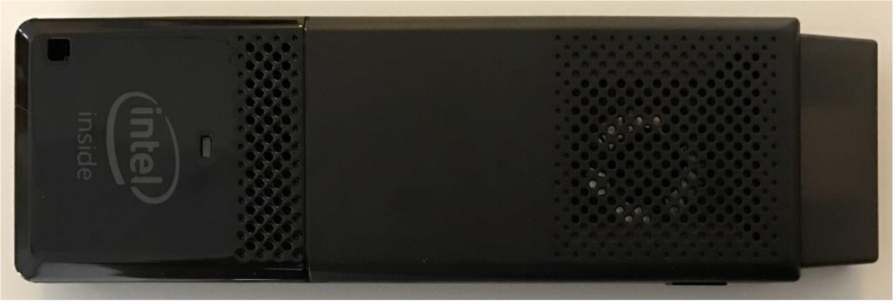
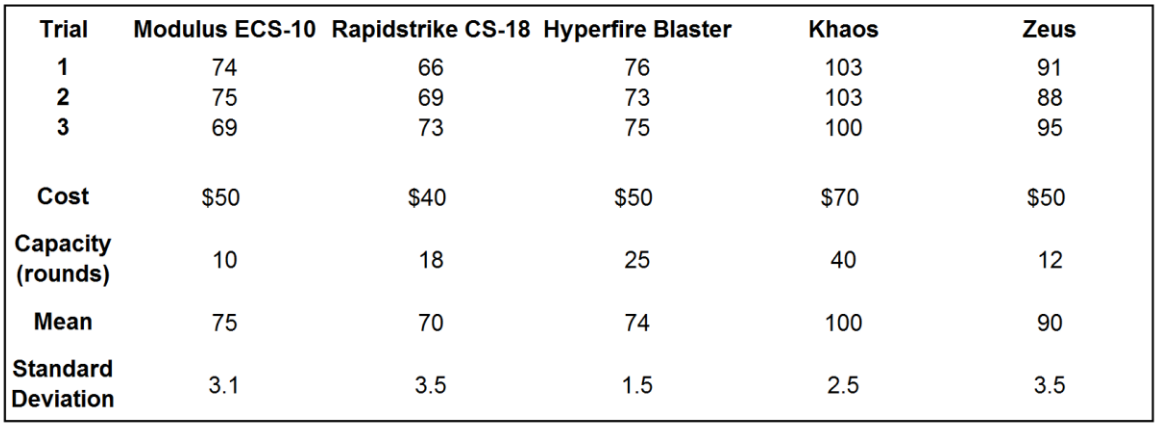


Figure 3.3.5.2 i: Intel Compute Stick

3.3.6 NERF Gun

If the weapon selected is inaccurate and cannot hit the designate course targets or the enemy robot, even the most accurate target recognition system will be rendered useless; therefore, it is critical to select a high performing NERF weapon to meet the requirements outlined by Lockheed Martin. NERF weapons were investigated regarding velocity and repeatability. Up to two NERF weapons are allowed on the robot: 1 NERF ball weapon, and 1 NERF dart weapon. Each system can carry a maximum of 50 rounds.

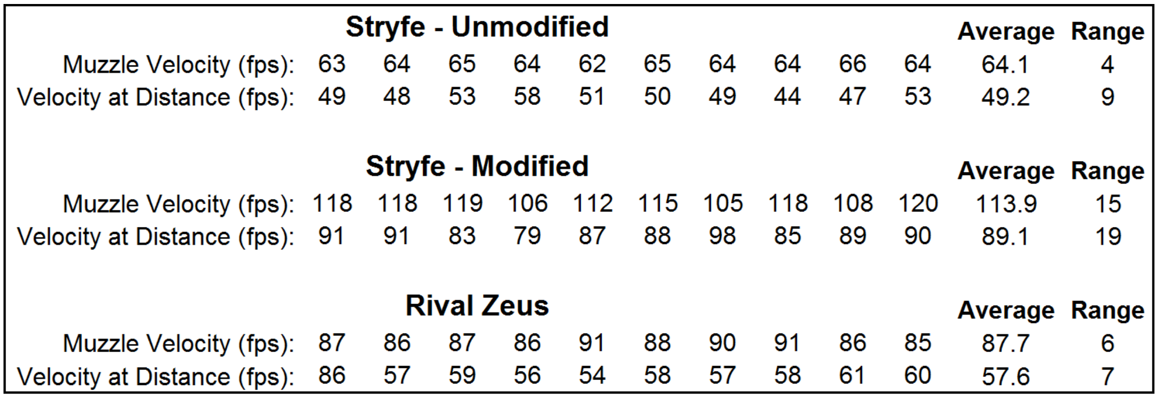


Trial	Modulus ECS-10	Rapidstrike CS-18	Hyperfire Blaster	Khaos	Zeus
1	74	66	76	103	91
2	75	69	73	103	88
3	69	73	75	100	95
Cost	$50	$40	$50	$70	$50
Capacity (rounds)	10	18	25	40	12
Mean	75	70	74	100	90
Standard Deviation	3.1	3.5	1.5	2.5	3.5

Figure 3.3.6 i: Average Initial Velocities with Standard Deviation

The first specification in focus was velocity of the projectile leaving the barrel of the gun. A list was compiled consisting of several NERF weapons under study: The Modulus ECS-10 (dart firing), the Rapidstrike CS-18 (dart firing), the Hyperfire Blaster (dart firing), and lastly the Rival Khaos and Rival Zeus ball firing guns. Figure 3.3.6 i shows a list of each gun's average initial velocities with standard deviations.

All the blasters in question utilize a flywheel and require a brief delay between shots for max velocity to be achieved for each round fired. As such, adopting a semi-automatic firing system instead of a fully automatic system will assist in keeping more consistent velocity and better precision in between shots. Figure 2.3.6 ii gives an idea of velocity drop-off at distance. It was found that the darts do not lose as much speed as the balls.



											Average	Range
Stryfe - Unmodified												
Muzzle Velocity (fps):	63	64	65	64	62	65	64	64	66	64	64.1	4
Velocity at Distance (fps):	49	48	53	58	51	50	49	44	47	53	49.2	9
Stryfe - Modified											**Average**	**Range**
Muzzle Velocity (fps):	118	118	119	106	112	115	105	118	108	120	113.9	15
Velocity at Distance (fps):	91	91	83	79	87	88	98	85	89	90	89.1	19
Rival Zeus											**Average**	**Range**
Muzzle Velocity (fps):	87	86	87	86	91	88	90	91	86	85	87.7	6
Velocity at Distance (fps):	86	57	59	56	54	58	57	58	61	60	57.6	7

Figure 3.3.6 ii: Velocity Drop-off at Distance

It is important to know how each firing system performs in terms of precision. For that reason, the levels of repeatability were examined for the NERF darts and

NERF balls. Knowing the effective combat range (the maximum range at which impacts will happen) will allow the team to determine exactly how to program the robot with respect to firing modes dependent on range. Figures 2.3.6 iii & IV show the grouping of each type of ammo at various ranges.

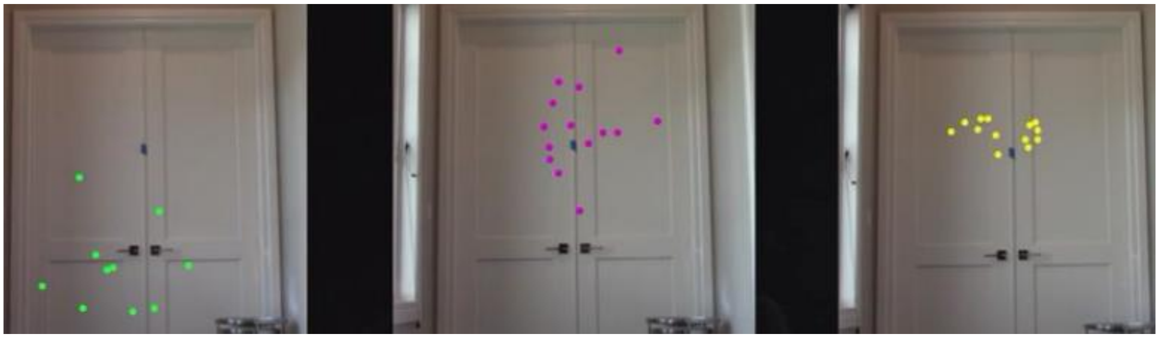


Figure 3.3.6 iii: grouping at 20-30 feet

Regarding precision, the balls dramatically outperform the darts. Darts are even more imprecise when distance is increased. Using darts at a range greater than 25 feet would be ill-advised. At a distance below 20 feet the NERF balls would almost definitely hit the enemy robot.

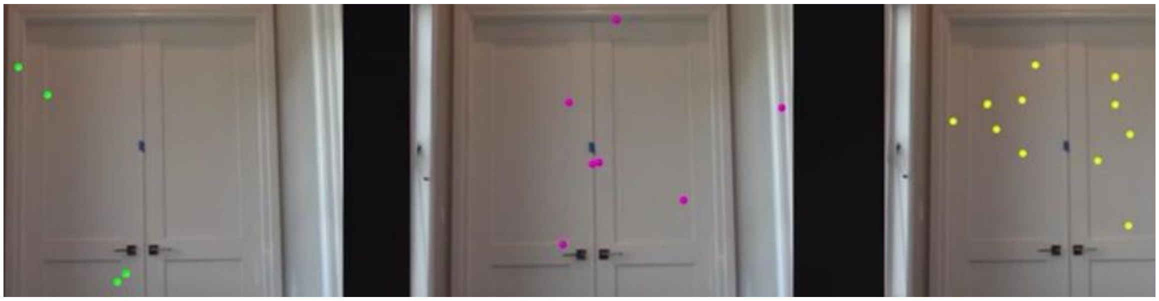


Figure 3.3.6 IV: grouping at 40-50 feet

It can be concluded that semi-automatic firing would be the ideal mode as it conserves ammunition and allows the round to achieve maximum initial velocity for each shot thereby increasing repeatability. Utilizing a NERF ball system in tandem with a NERF dart system would be ideal since more available ammunition means more potential points during the competition. Also, standard NERF darts shouldn’t be used for targets further than 25 feet; they may prove useful for targets closer than 20 feet if the team can incorporate a second weapon into the budget and weight/height requirements. As the primary weapon, Green Team / Group 2 procured the NERF Rival Khaos blaster due to its ammo capacity, precision, and range. The blaster did not yield a large price increase over the other blaster and has an electric trigger for simple integration. The NERF Rival Khaos will be evaluated to confirm performance specifications (figure 2.3.6 V) and to see what parts of the weapon can be modified/removed to reduce weight without affecting the original projectile velocity of the rounds.

Figure 3.3.6 V: NERF Rival Khaos

3.3.7 Webcam

The section will discuss a few different options in terms of webcams.

3.3.7.1 Logitech Webcam Series

One of the webcams to be considered for the project will be the Logitech HD Webcam C270. This webcam can record in HD 720p and is not very expensive, but the quality of the camera is not the best it can be. A picture of the C615 camera can be found in figure 3.3.7.1 i.

Figure 3.3.7.1 i: The Logitech C270 (Awaiting Approval)

	Logitech C270	**Logitech C615**
Photo Quality	3 Megapixels	8 Megapixels
Field of View (FOV)	60°	74°

Optical Resolution (True)	1280 x 960 1.2 MP	True = 2MP,
		Interpolated = 8MP

Video Capture (16:9 W)	360p, 480p, 720p;	360p, 480p, 720p, 1080p;
Frame Rate (max)	30fps @ 640x480	30fps @ 640x480
Focus Type	Always Focused	Auto Focus

Table 3.3.7.1 i: Specifications for the Logitech C270 and C615

Some of the important comparisons follow about the camera in this section in table 3.3.7.1 i.

Another Logitech webcam to consider is the C615. The C615 HD webcam is capable of recording video in 1080p, which is higher quality than the C270, but it is more expensive.

Figure 3.3.7.1 ii: The Logitech C615

The main difference between the C270 and the C615 is the number of pixels that make up the image and the number of pixels displayed across the screen (as shown above). The C270 is 720p while the C615 is 1080p. The main difference between 720p and 1080p is the number of pixels that make up both images. For 720p the number of pixels is about 1 million and it is about 2 million pixels for 1080p.

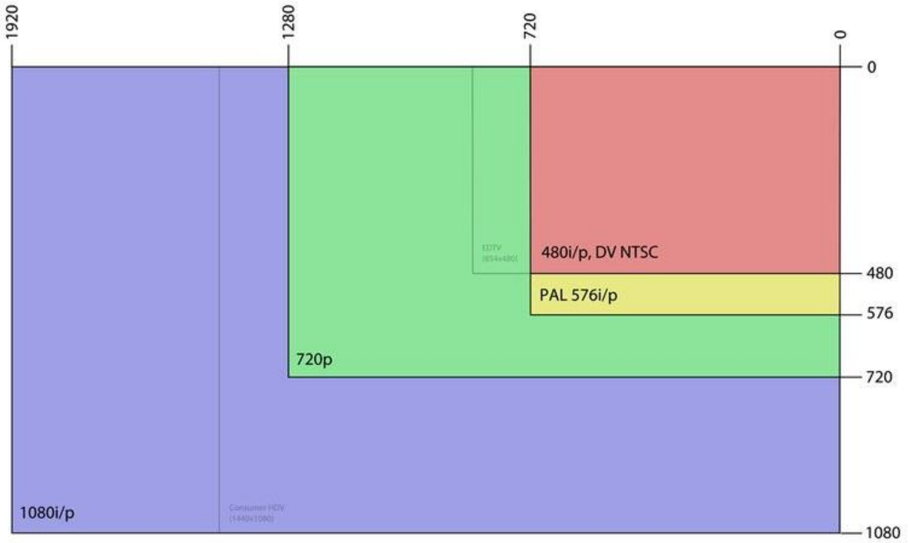


Figure 2.3.7.1 iii: Video Resolution Chart - 480i to 1080p
Image via Wikimedia Commons - Public Domain

3.3.8 Power

A major component of this project is power. Without power, nothing can be used. This section will describe a few of the different options available for powering the numerous components.

3.3.8.1 External Battery Pack

A power supply will be needed to power the Intel stick, webcam, and the microcontroller. This power supply must supply at least 5V to power everything.

There are various options when choosing what kind of battery pack to be used. External battery packs provide sufficient output voltage for the needs of this project. These packs are also rechargeable, so they provide an advantage to battery packs because if you run out of power you can charge it back to full strength within a few hours.

One available option is the RAVPower 13000mAh external battery pack. This pack provides the necessary voltage output to power what is needed. It also has a long battery length. A picture of this battery pack is shown in Figure 3.3.8.1 i.

Figure 3.3.3.2 i –Picture of the RAVPower 13000mAh battery pack

Another option is the Jackery Giant+ external battery pack. These two external battery packs are compared in Table 3.3.8.1 i.

Component	RAVPower	Jackery Giant+
mAh	13000	12000
Output Voltage (V)	5	5
Output Current (A)	4.5	3.1
Number of USB Inputs	2	2
Weight (lbs.)	0.71	2
Dimensions	5.00 x 0.88 x 3.20 in	0.8 x 3.1 x 4.3 in
Price	$26.99	$19.99

Table 3.3.8.1 i – Comparing the two external battery packs

After some consideration, the RAVPower external battery pack was chosen. This pack has some extra battery life to it and it is about half the weight of the Jackery Giant+.

3.3.8.2 Battery Holder

The last portion that needs to be powered is the servos and the ultrasonic sensor. These all require at least 5V of input power. To achieve this a battery holder will be used. These come in different voltage variations and some have switches while others do not.

For this project, a battery pack with a switch will be used. This will allow us to conserve some power because we can switch it on and off as needed.

Our first option is to use a 4 x AA battery holder case. This battery holder outputs 6V. It also has an on/off switch with a cover, so the batteries are protected and we can conserve battery life. The AA battery holder is shown in Figure 3.3.8.2 i.

Table 3.3.8.2 i – AA Battery Holder

Another option is to use a 9V battery holder. This one also has a switch, so battery power can be conserved. This holder outputs 9V.

After some consideration, it was decided that the 4 x AA battery holder case. This holder provides sufficient voltage to suit our needs for this project.

3.4 INVESTIGATED ARCHITECTURES AND RELATED DIAGRAMS

Before settling on a final architectural model for the all-up targeting system, it was important to brain-storm several ideas to ensure that it was why the group wanted to pursue. The final plan needed to be achievable from a university-level standpoint but also original and unique. The plans stem from weeks of research into the subject matter and developing such design plans were key in helping the group to achieve a clear vision of what needed to be accomplished.

3.4.1 Block Diagrams

Two primary systems were considered for the Autonomous Targeting System. One system was dependent on the raspberry PI, the second was dependent on the Intel Compute Stick. The flow charts in sections 3.4.1.1 and 3.4.1.2 provide a high-level investigation on how the processing units would fit into the scheme of the robots central targeting functions. Ultimately, the Intel Compute Stick was chosen primarily for its flexibility and higher performance specifications.

3.4.1.1 ATR Processing: Raspberry PI with a Camera and LIDAR

Figure 3.4.1.1 i shows a theoretical setup if Green Team / Group 2 were to use a single Raspberry PI 3 as the sole conduit of processing received sensor input to control the aiming and firing of the NERF weapon in addition to transmitting video feedback to the user.

The Pi would be given information from human input, sensors, and the confirmation of the turret movement. Also, the Pi would pass out information to the motors for movement, to the servo motors to control the turret, to fire the gun, and to pass the visual feed out for the judges.

The human input consists of a radio control such as ones seen with remote control cars and such. Also, another potential human input is from a mobile application via a Bluetooth connection. The sensor input consists of the camera and the LIDAR. The camera we are using is a Logitech camera and the LIDAR was never decided on and changed to an ultrasonic sensor described further later. But, LIDAR stands for Light Detection and Ranging and is measures distances using a laser light which illuminates its target.

The output from the Raspberry Pi consists of the visual feed which is needed for the judges. The motors for moving the car, and this output depends on the input from the human input because the driver will physically control the robot. Also, output to how to move the servo motors depending on the angle needed to make an accurate shot. Then, that would require an output signal to fire the gun.

This setup was later discarded after concluding that the Raspberry PI alone would not be powerful enough to handle the ATR software effectively to the extend required. LiDAR was also discard as the secondary sensor choice due to its high cost.

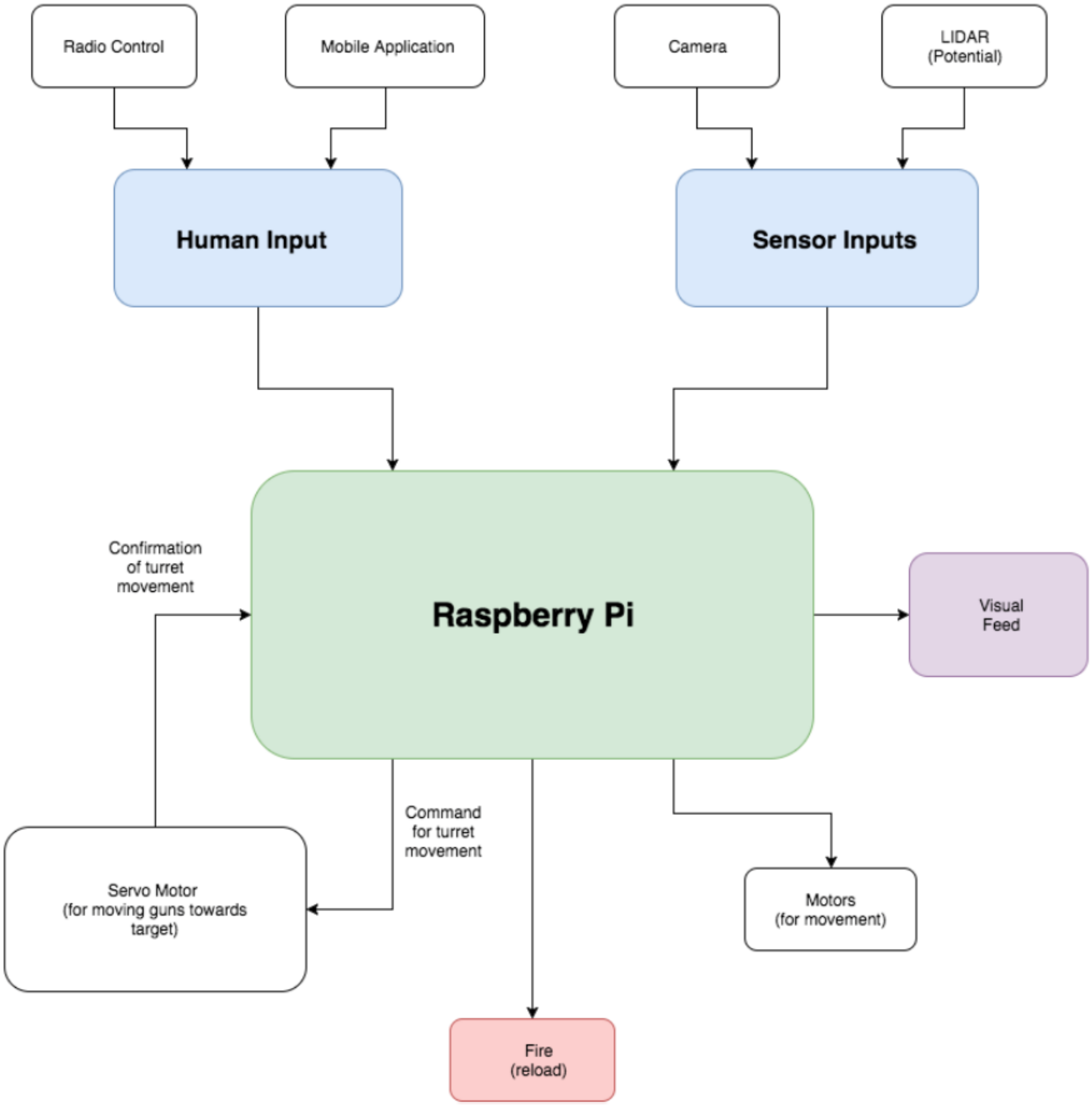


Figure 3.4.1.1 i: Proposed ATR System with Raspberry PI
Image via Daniel Healy: Computer Science Green Team Group Member

3.4.1.2 ATR Processing: Intel PC with Webcam and Ultrasonic Sensor

Figure 3.4.1.2 i shows the Intel Compute Stick with the use of the Arduino microcontroller. The Intel Compute Stick would be passed in information from the mobile application and the camera. The Arduino would be passed in information from the radio controller, the Ultrasonic sensor, and the confirmation of turret movement. Another input for both the Intel Compute Stick and the Arduino is the transfer of data between the two systems. Outputs from the Intel Stick are only to the Arduino. Outputs from the Arduino include to fire the gun, to move the motors for movement, to move the servo motors for the gun, and to the Intel Stick.

The only real difference in this setup from the Raspberry Pi system with the Arduino is the change from the Raspberry Pi to the Intel Compute Stick. Everything else stayed constant with the Arduino taking in the same inputs and the data transfer between the Raspberry Pi and what is now the Intel Compute Stick still existing.

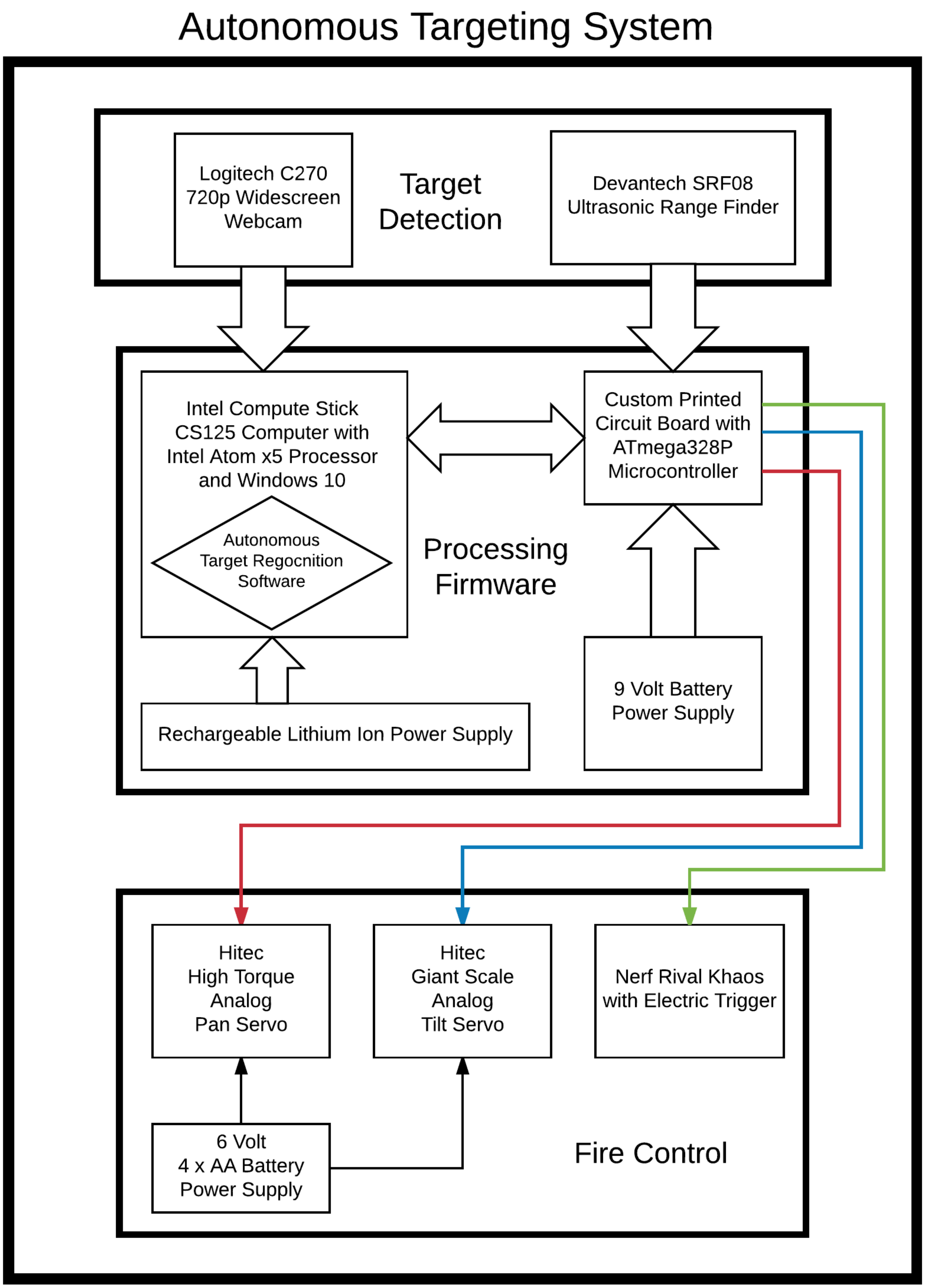


Figure 3.4.1.2 i: Autonomous Targeting System

3.4.1.3 Hypothetical Autonomous Targeting Software Function Overview

The flow chart shown by figure 3.4.1.3 i is subect to future modification however, it represents a combination of the required performance specifications of the software as defined by Lockheed Martin in addition to functionalities that Green Team / Group 2 hopes to deliver.

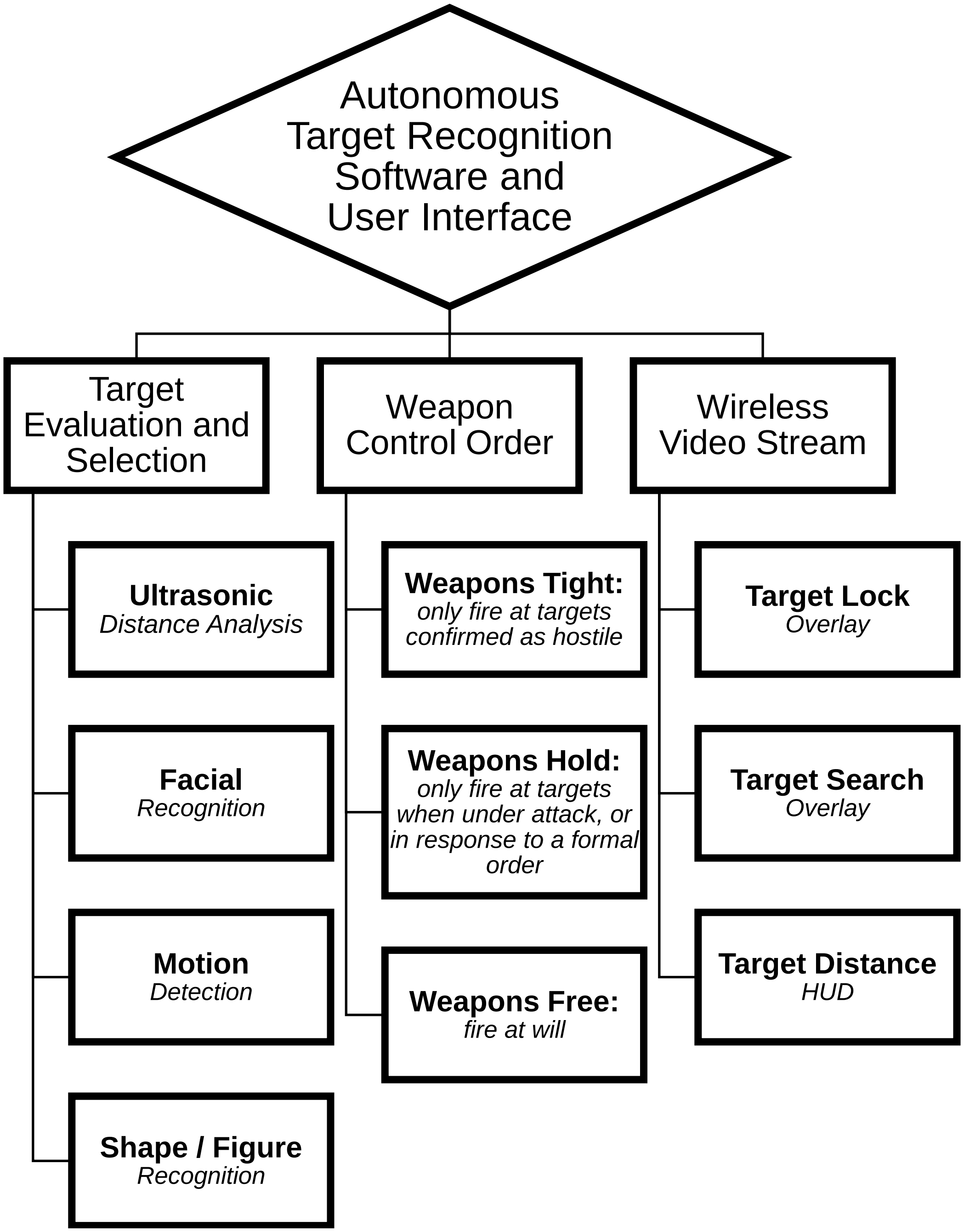

Autonomous Target Recognition Software and User Interface
Target Evaluation and Selection
Weapon Control Order
Wireless Video Stream
Ultrasonic Distance Analysis
Facial Recognition
Motion Detection
Shape / Figure Recognition
Weapons Tight: only fire at targets confirmed as hostile
Weapons Hold: only fire at targets when under attack, or in response to a formal order
Weapons Free: fire at will
Target Lock Overlay
Target Search Overlay
Target Distance HUD

Figure 3.4.1.3 i: Autonomous Target Recognition Software and User Interface

3.4.2 Other Related Design Diagram / Design Architecture

This section includes additional related schematic details for various components that are useful for fully visualizing the capacity and usefulness of selections.

3.4.2.1 Intel Compute Stick Schematic

The Intel Stick as shown in figure 3.4.2.1 was chosen for its high power to cost ratio. With the new 14 nanometer manufacturing of silicon fully matured, Intel has created some of the best chips in terms of both raw power provisions and energy efficiency. Green Team / Group 2 only needed a small and very straightforward device to run the necessary targeting software. The Stick Provides exactly what is needed without overcompensating or compromises.

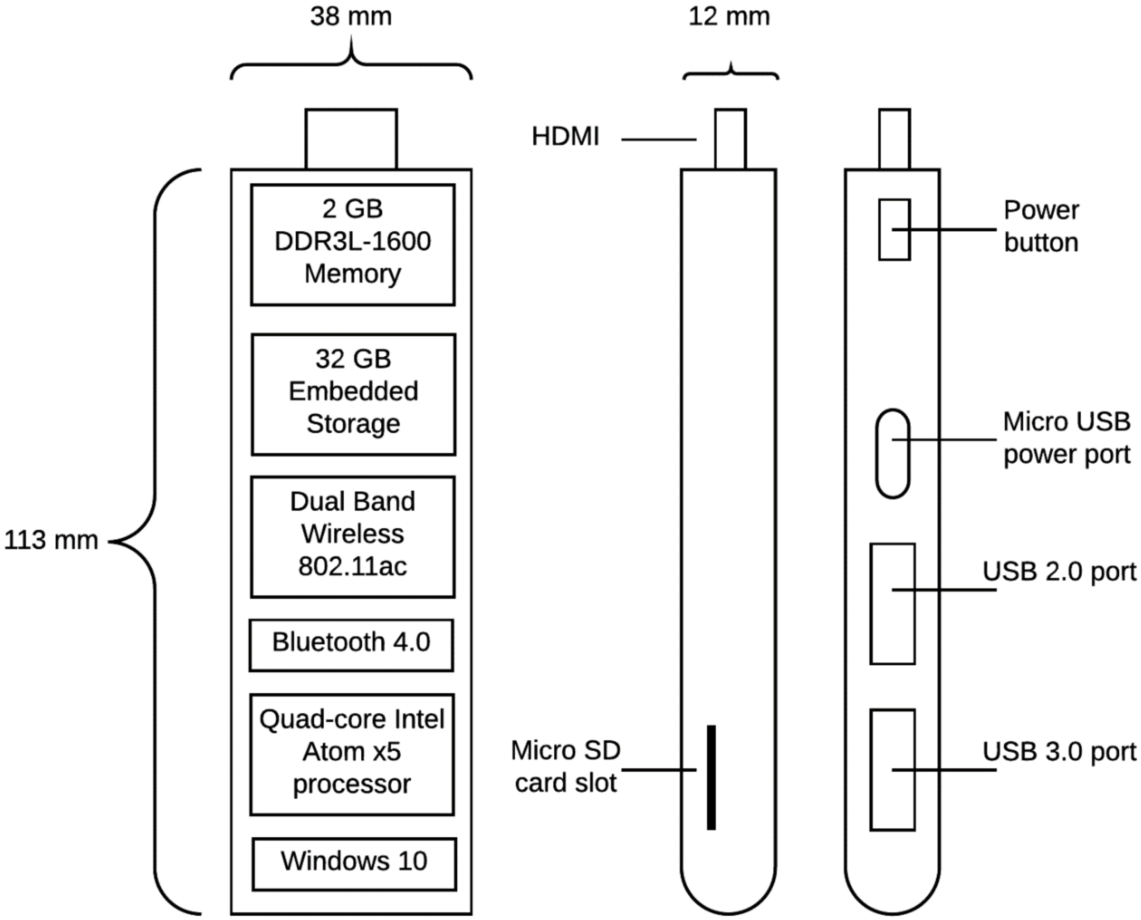


Figure 3.4.2.1 i: Intel Compute Stick Schematic

For development purposes, it comes with an HDMI port for plugging into any modern display. Without an embedded display, this makes it more portable and less costly. The small footprint makes it easy to mount on the robot in a variety of locations. The small footprint also means that rapid transfer between group members for development purposes will be a non-issue.

The Intel Stick has exactly the number of data ports required. One USB to connect the camera, one USB to connect the PCB, one port for power input from the rechargeable battery, and finally one SD cart slot to expand the storage. Upon further analysis, the software designers concluded additional storage space will be needed to house the targeting program so this will be a useful feature and excellent for its size. Inside the Intel Stick comes the latest standard in Wi-fi. This is extremely imperative considering the groups reliance on over-the-air broadcasting of the video signals to a remote device for viewing and administrative control as specified in the requirements by Lockheed Martin. Other features included inside the Intel Stick are the latest operating system by Microsoft: Windows 10 and two gigabytes of memory.

3.4.2.2 All-Up Proposed Robot: Solid Works Model

This section provides a visual look at an early Solid Works model for the robot in Figure 3.4.2.2 i, created by Green Teams mechanical engineers. The model shows a design where the camera is in a forward-mounted fixed position. The wheels are omni-direction for dynamic mobility control. The turret which houses the NERF gun has 180-degree horizontal motion capability as well as tilt capability. The electronics designs by Group 2 will be housed within the chassis to prevent thermal leakage that could be used by the opposing team to detect the robot via Forward Looking Infrared sensors. Green Team / Group 2 will consider using a material called white optics to shield the robot from visual spectrum imaging sensors.

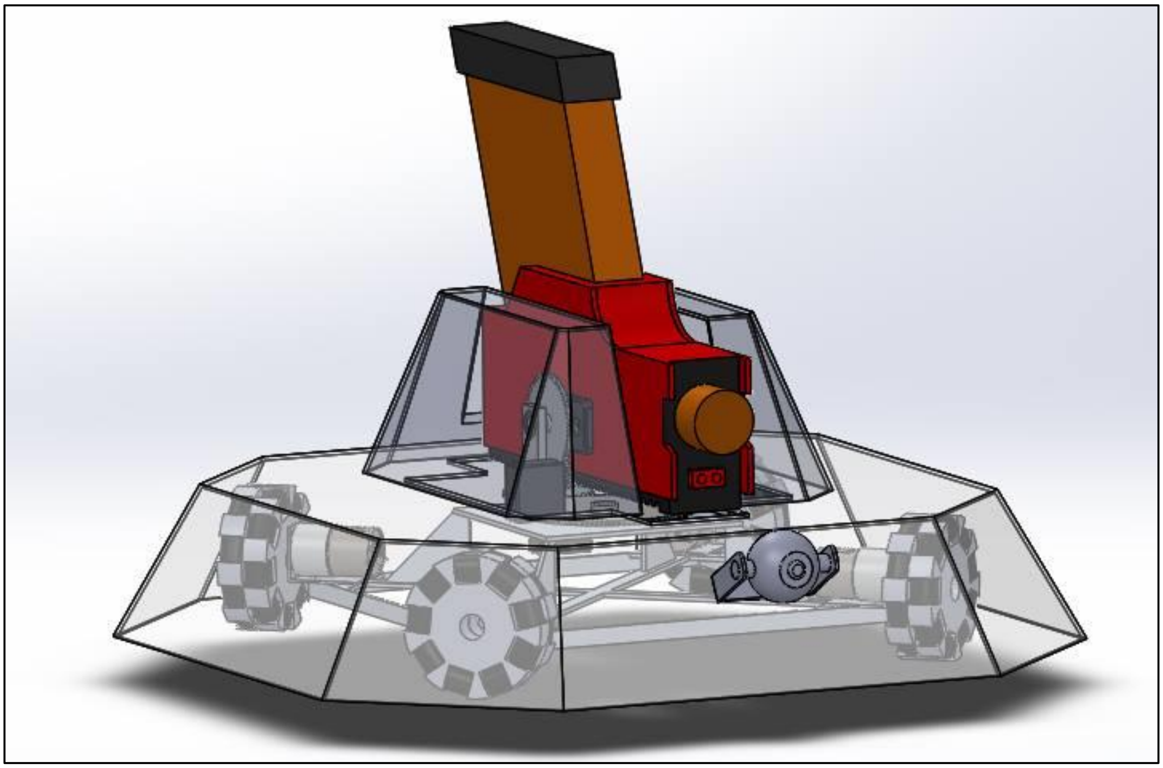


Figure 3.4.2.2 i: Early Robot Solid Works Model (via Green Team)

4.0 STANDARDS & REALISTIC DESIGN CONSTRAINTS

Successful societies, communities, and even religions have always had a set of ethics to guide the way forward and not back. For example, Christianity believes in the Ten Commandments, which provides the basic construct for their belief system. Another example is the United States of America. This country established ten governing amendments over 200 years ago, that our society still follows today. For our project, we must follow the Institute of Electrical and Electronics Engineers Standard Association (IEEE) standards. They too have a set of ethics and a similar set of amendments to progress the community of electrical and computer engineering. The code of ethics and conduction of professionalism is agreed upon by all members and communities of IEEE.

It is also important to note, that there is a system in place to update their code of ethics. Interestingly, like the United States, there is a vote held between members of the Board of Directors to determine if an ethical code needs to be changed or added to the list.

4.1 RELATED STANDARDS

Though the code of ethics is a good starting point, IEEE has set standards for working with specific systems that are found throughout the electrical field. This holds the benefit of regulating what the industry is producing and how they produce it. This ensures that systems are consistent. In an ever-changing world, consistency, especially in the electrical industry, is imperative to the progression of the community. Due to the nature of our project, there are standards by IEEE that we adopt to guide us through the development of the product.

1. 1044-2009 – IEEE Standard Classification for Software Anomalies: This standard provides a uniform approach to the classification of software anomalies, regardless of when they originate or when they are encountered within the project, product, or system lifecycle. Classification data can be used for a variety of purposes, including defect causal analysis, project management, and software process improvement (e.g., to reduce the likelihood of defect insertion and/or increase the likelihood of early defect detection).
2. 2700-2014 – IEEE Standard for Sensor Performance Parameter Definitions: A common framework for sensor performance specification terminology, units, conditions and limits is provided. Specifically, the accelerometer, magnetometer, gyrometer/gyroscope, barometer/pressure sensors, hygrometer/humidity sensors, temperature sensors, ambient light sensors, and proximity sensors are discussed.
3. 1554-2005 – IEEE Recommended Practice for Inertial Sensor Test Equipment, Instrumentation, Data Acquisition, and Analysis: Test equipment, data acquisition equipment, instrumentation, test facilities, and data analysis techniques used in inertial sensor testing are described in this recommended practice.

4. 208-1995 – IEEE Standard on Video Techniques: The methods for measuring the resolution of camera systems are described. The primary application is for users and manufacturers to quantify the limit where fine detail contained in the original image is no longer reproduced by the camera system. The techniques described may also be used for laboratory measurements and for proof-of-performance specifications for a camera
5. 1754-1994 - IEEE Standard for a 32-bit Microprocessor Architecture: A 32-bit microprocessor architecture, available to a wide variety of manufacturers and users, is defined. The standard includes the definition of the instruction set, register model, data types, instruction op-codes, and coprocessor interface. A 32-bit microprocessor architecture, available to a wide variety of manufacturers and users, is defined. The standard includes the definition of the instruction set, register model, data types, instruction op-codes, and coprocessor interface.

The above list is flexible and likely to change. We will mainly be using the standards listed above, though it is possible we incorporate other standards into the project.

4.2 DESIGN IMPACT OF RELEVANT STANDARDS

All the standards are important - relating to the project one way or another. These standards will help guide us through working with the electrical components in the project. They will also enable us to have a standard approach to dealing with issues that may arise. It is also important to have standards for the community.

The community depends on standards for keeping things, well, standard. These standards allow other engineers to look at components of the project and know how we implemented them. This will make the project easier to debug from a third party when we face adversity.

The 1044-2009 – IEEE Standard Classification for Software Anomalies will be important for the project. This standard is important for us because we will be dealing with software and one never gets software right on the first try. Though we may run the software through a multitude of tests, it is likely we will experience abnormalities and need to be ready for them. When they do, it is vital Group 2 / Green Team will know how to handle such bugs.

Another important standard is the 2700-2014 – IEEE Standard for Sensor Performance Parameter standard. We will be using sensors to locate targets for our weapons. This is the first time that most of the team members have used sensors and we will need to understand as much as we can to ensure we produce the best quality product. This standard will show us how to define the sensors' capability, performance, and the units for the sensors.

Being able to define the sensor performance is one thing, but it is another when we need to test it. The 1554-2005 – IEEE Recommended Practice for Inertial Sensor Test Equipment, Instrumentation, Data Acquisition, and Analysis standard will be useful for the testing of the sensors. This standard will tie into the anomalies standard. We must be able to test the project and ensure that it will work in the competition. We will need to set up realistic test scenarios. These test scenarios will give us one of two outcomes. They will either give us the confidence that the project works or the bugs to get the project to work.

The project will also include a video feed of what the robot sees to an application. The video feed will utilize the 208-1995 – IEEE Standard on Video Techniques. This video will be used, alongside the sensors, to find and acquire targets. It ensures that Group 2 / Green Team can get the feed to run smoothly without latency or lag.

4.3 REALISTIC DESIGN CONSTRAINTS

This section will describe some of the constraints Group 2 will face.

4.3.1 Economic Constraints

Before considering any type of project and what specific parts to purchase, there are economic constraints that first need to be considered. The first issue to consider is the total budget that is available for the project. For the project, we must keep the maximum as-demonstrated cost under $1000 and the overall maximum project budget is $2000. The budget and the financial assistance is set by the project sponsor, Lockheed Martin. This must all be taken into consideration when determining which parts to purchase to keep cost at the minimum. One way to lower the overall cost of the project is to purchase a smaller sensor. Although the more expensive the sensor, the larger the range it has, it is not economically viable for this project. The least expensive type of sensor is the ultrasonic sensor. There are many different ultrasonic sensors from various companies that are all typically priced from $20-$50. These sensors also typically have similar ranges that are all within 0-10 feet to each other. Once you start going into ranges of 50 feet or more, the sensors become more than twice as expensive and are also not easily compatible with a microcontroller. Thus, from this constraint, we had to choose a sensor with a smaller range to reduce costs. Another effect of economic constraints on the project is the choice of servo motors that will be used. Originally, we were going to use smaller servos that are not as strong, but were also cheaper than a larger servo. However, for the project, we need a stronger servo to power the pan and tilt system. Since we need a more powerful servo it will become more expensive. To reduce cost multiple companies will be considered to find the most effective and cheapest part to use for the project. Overall, the budget will be discussed extensively amongst the group and the Lockheed Martin representative to keep the project within budget.

4.3.2 Time Constraints

The amount of time that will be allocated for this project is a predetermined time frame set by the university. For the senior design project, we will be able to work on it over the period of two semesters. During the first semester, the group will be focusing primarily on the planning of the project. This will include the schematic of the project and the preparation of constructing it. In the second semester, will be primarily focused on building the project. During this time, there will also be testing phases and plans will be put into action to complete a fully functioning project. This timeline is more fully discussed and outlined in the Milestone Discussion section of this report. Another time constraint for this project is the schedule of the individual team members. All members each have their own schedules that will affect their progress on the project. This could be due to leisure activities, academic reasons and job related reasons. This two-semester time constraint must be incorporated in varying schedules. In doing so, it will ensure that the project will be completed in a timely manner and the project's milestones will not be tardy. To optimize the time that we have, we must plan accordingly to the predetermined project schedule to keep on the correct timetable.

4.3.3 Environmental, Social, and Political Constraints

In terms of environmental constraints on this project, there are not many to consider. The project will be designed to work in an indoor environment and not to be used outside. With that said, the only environmental constraint would be if the project were to be disposed. The project consists of many electrical components and will have a battery pack that could affect the environment if improperly disposed of. However, this is an unlikely scenario so the environment constraints can be virtually ignored. As social constraints are concerned, there are not any to consider for this project. Also, there will be no political constraints to consider for this project.

4.3.4 Ethical, Health, and Safety Constraints

Ethically, this project will have very few constraints as the intended use of the project is on other objects, not on humans. However, one of the most major safety constraints to be considered is the well-being of all the participants that will be involved in the competition. Upon completion of the project, there will be a competition held indoors with two other teams with similar projects. The teams, along with the judges of the competition, will be all be within feet of each of the robots. We must make sure that the robot does not go outside the boundaries of the playing field to lessen the risk of injuring another person. This could be from the robot itself colliding with one of the members. Also, the robot will be shooting projectiles autonomously, which can cause injury if a person is struck by that projectile. The weapon being using is a nerf weapon that will capable of launching a projectile over 20 feet and with speeds of up to 30 m/s. If a person is struck in the eye with the projectile, it could inflict harm upon that individual. The weapon itself must not be modified to overcome this 30 m/s limit that the nerf weapon has. Although you may be able to gain greater distance and accuracy by increasing the speed of the projectile, we must find alternatives to protect all the individuals that will be at the competition. Safety and health constraints will go hand in hand and will not be considered apart from one another.

4.3.5 Manufacturability and Sustainability Constraints

This project will be heavily influenced by manufacturability constraints that will determine the overall design. One of the main constraints is to achieve a modular type system that can be easily built and integrated onto any type of platform. For the project, the weapon system will be mounted on a remote-controlled robot. We must use easily accessible parts to create the system and to keep cost at a minimum. Along with this, it must all be modular and it must easily be able to be taken apart by the customer. This will be achieved by creating multiple subsystems for each individual section of the project. This will include the actual weapon, the sensor(s), the pan and tilt mechanism (which will also include the servo motors) and the microcontroller. Also, the entire system must be as lightweight as possible and the overall size must be kept at a minimum. In doing so it will allow for maximum speed and mobility of the robot of which the system will be mounted on top of. If the weight of the system is too great, it will cause major issues with the robot itself and will heavily affect the overall performance in the competition with the other teams. Another constraint to consider is the sustainability of the entire system. The competition will consist of multiple rounds, so the system must be able to consistently track, target and fire upon possible candidates. This will be heavily influenced on the power supply of the system. We must keep in mind of this issue and must ensure that the system will be able to fully function for an extended period. Each of the rounds will be a total of 10 minutes, so the system must have a battery life beyond 30 minutes to ensure that it will not run out of battery during the competition. Along with this, the system will be fired upon from the opposing team. Projectiles will be either a nerf ball or nerf dart and will be shot upon the system at a maximum speed of 30 m/s. The user will not know where the system will be struck, so we must make sure the system can withstand being struck with these projectiles. All the parts of the system must be able to withstand the force of the projectiles and must not be able to be compromised. This will be ensured by encasing the microcontroller to make sure none of the components can be knocked loose or destroyed. Overall the system must be able to survive these constraints over multiple rounds to be the most effective at the desired task.

5.0 DESIGN

5.1 INITIAL DESIGN ARCHITECTURES AND RELATED DIAGRAMS

The design of our autonomous weapon system will integrate three different subsystems to efficiently and effectively determine a target, align the gun, and fire at the target. The placement of all the components on the robot will be considered because we have a limited area to place all the subsystems. They will need to be arranged efficiently to optimize the space on the robot.

5.1.1 Top Level Robot Design

The primary focus of the Robot Design for the ECE group within Green Team will be the Autonomous Targeting System, specifically integrating the external peripherals using the chosen microcontroller. ECE will work with CS to ensure the software performs well with the electronic hardware. The Mechanical Engineering group will focus primarily on the Drive System. Figure 5.1.1 i shows the top-level block representation of the overall system design including how the user will interact with the drive system and targeting system.

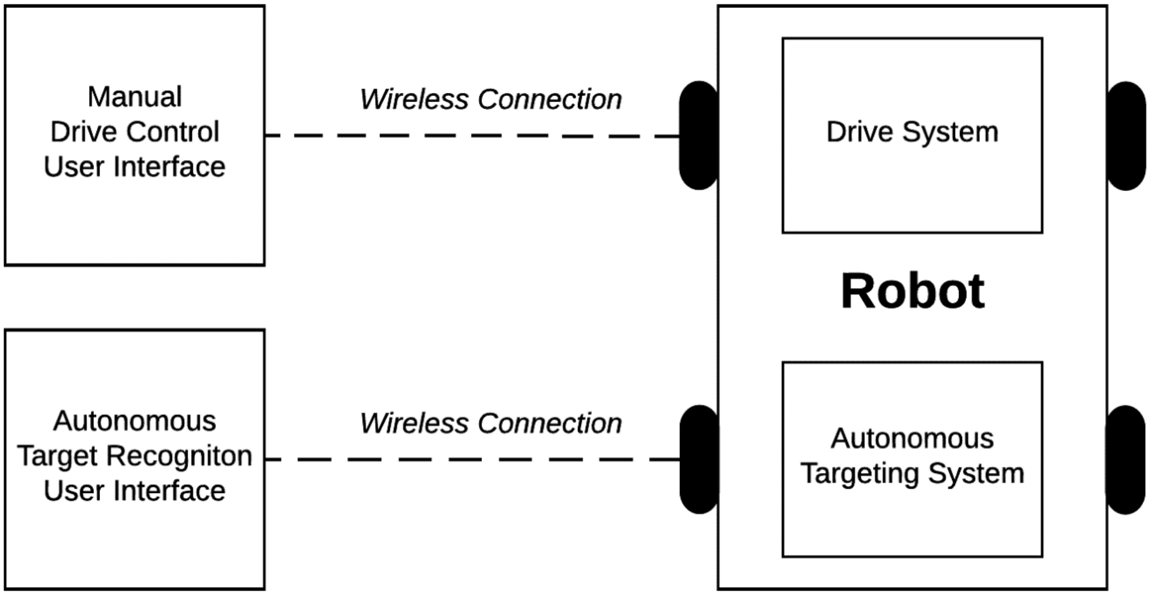


Figure 5.1.1 i: Top-level System Design Block Diagram

5.1.2 Autonomous Targeting System Breakdown

Focusing on the Autonomous Targeting System block from Figure 5.1.1 i, the three primary subsystems are broken down in figure 5.1.2 i. Figure 5.1.2 i displays an overview of what is included in all the subsystems. Though the figure is a little vague, it gives a high-level representation of each subsystem. You will see below, each subsystem at a closer, more engineering design level. Subsystem 01 (Fire Control) will move the gun and determine when to fire the gun. The responsibility of determining how far and location of the target lies in Subsystem 02 (Target Detection). Subsystem 03 (Processing) is the processing that will run our algorithms that brings together the Fire Control Subsystem and the Target Detection Subsystem.

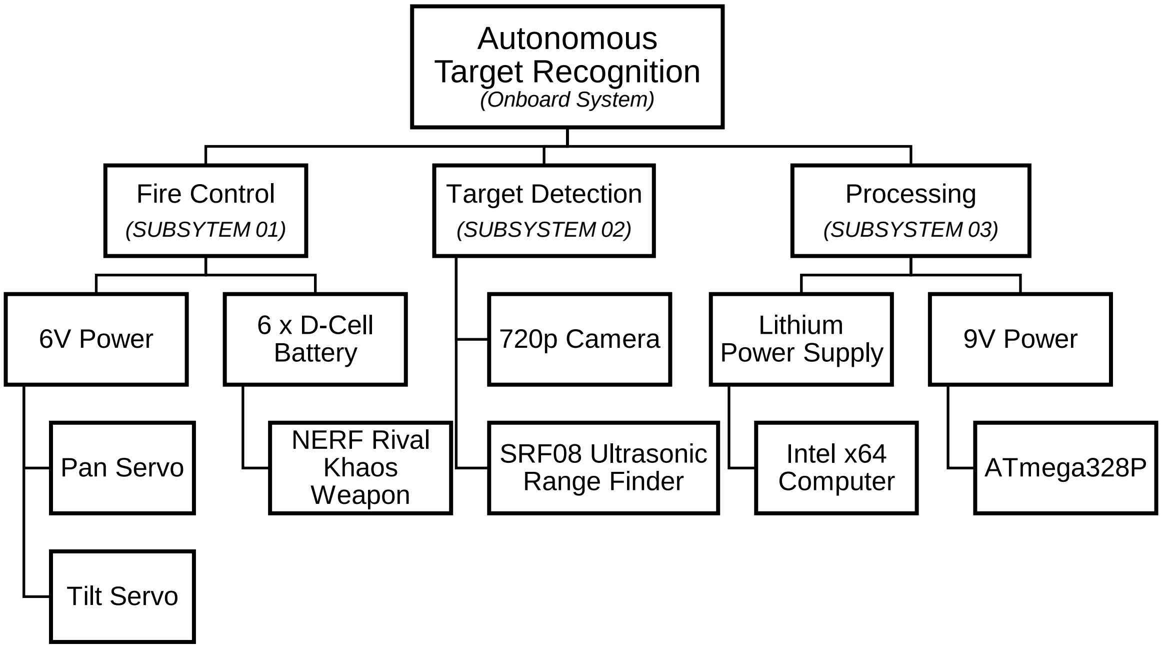


Figure 5.1.2 i: Autonomous Target Recognition Systems Flow

5.2 FIRE CONTROL (FIRST SUBSYSTEM)

The first subsystem is the Fire Control Subsystem. This subsystem covers much of the space on the robot. This subsystem gets most of its size from the NERF gun, which is roughly two feet. This takes up about two-thirds of the length. There are ways to shorten the gun without losing the purpose of it. Shortening the length of the gun will help reduce torque experienced by the pan/tilt servos. The Fire Control Subsystem includes the pan and tilt servos, the NERF Rival Khaos weapon, and their power supplies. This subsystem relies on the other two subsystems for it to be effective. This subsystem will receive information from processing subsystem. The information received will direct the pan/tilt servos to align the NERF Rival Khaos with the target. Once aligned, the system will fire and hit the target.

5.2.1 Subsystem #1: Fritzing Schematic

Figure 5.2.1 i was created using a free software called Fritzing. This software allows us to show how the Fire Control Subsystem is set up. The servos will be directly wired to the 4x AA battery pack and have a connection to the Arduino to receive data on how to move the servos to align the gun with the target. The Arduino will also be providing data to each of the transistors responsible to act as switches to the motors. The reason for this, is because we do not want the gun to constantly be revving and firing. A bigger and clearer picture of this connection is displayed in Figure 7.2.1.4 ii. The two switches allow us the ability to turn off the flywheel and fire when the algorithm is ready to do so. The flywheels are also extremely loud. So, implementing this design will allow the Fire Control Subsystem to be quiet, until it shoots. Furthermore, this setup prevents battery drain, such that the 9-volt power supply is not constantly on. If Green Team / Group 2 has trouble timing when to turn the flywheel on relative to firing the gun, we will directly connect the flywheel motor to the 9-volt battery supply. This will ensure that we do not have issues waiting for the motor to warm-up between being off and taking the shot that proceeds after the flywheel is moving at its highest velocity.

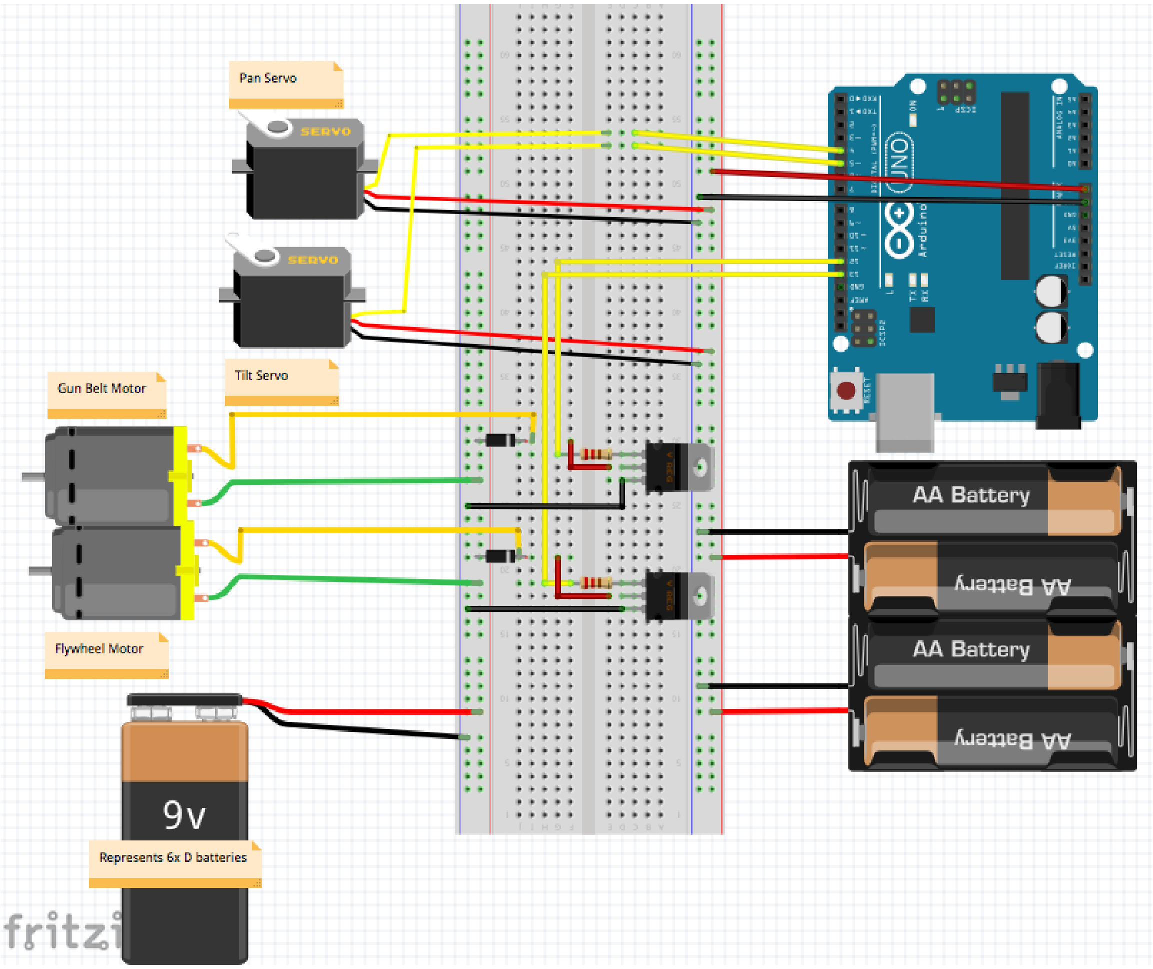


Figure 5.2.1 i: Fritzing Schematic of the Fire Control Subsystem

5.2.2 Subsystem #1: Test

Figure 5.2.2 i shows the in-lab breadboard test of the Fire Control subsystem. This breadboard test was conducted to ensure functionality of specific parts that are in control of the actual aiming and firing of the gun. Between the Arduino Uno and the placed parts on the breadboard, a mockup of the PCB can be devised. In Figure 5.2.2 i the NERF Gun, pan servo, Arduino Uno, and 6-volt battery pack can be seen. The giant scale tilt servo was not available yet at the time of the test. Regardless, the system should work once the additional servo arrives. Overall, the breadboard test of the Fire Control subsystem proved that the gun can be fired using microcontroller signals while simultaneously aiming and controlling the servo(s). The only issue that was quickly resolved was getting the computer to recognize the microcontroller USB connection. This was resolved by reconnecting the microcontroller and uploading the Arduino code from the computer.

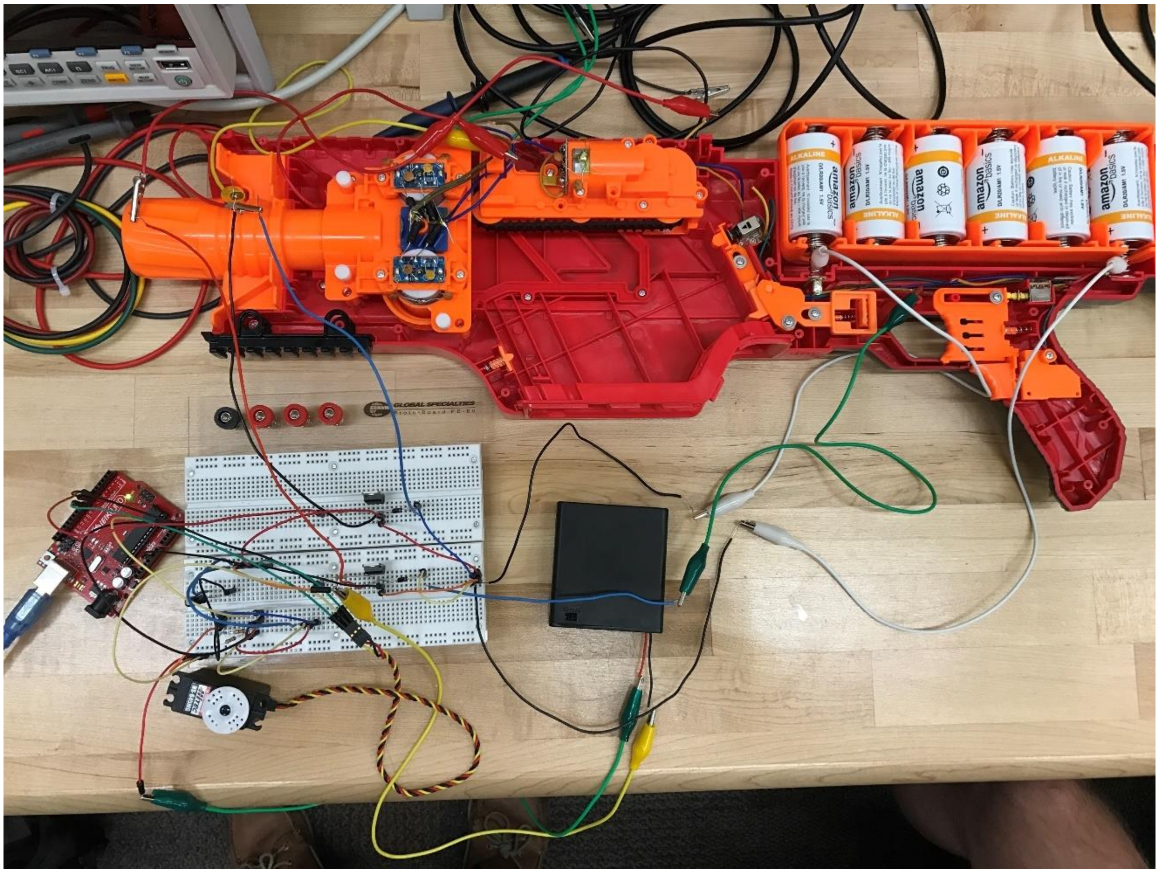


Figure 5.2.2 i: Fire Control Breadboard Test

5.3 TARGET DETECTION (SECOND SUBSYSTEM)

Subsystem 02 is our eyes on the robot and is referenced as the Target Detection Subsystem. It comprises of the SRF08 Ultrasonic Range Finder and the 720p camera. These components will relay visual events to the Processing Subsystem. This way the processing can process the images to determine what is being visualized in front of the robot. Once complete, the Target Detection Subsystem will use facial recognition software to recognize a target. For testing the entire autonomous system, we will design the target detection to rely on red dots to signify a target. This software will be much simpler than the facial detection, but will serve the purpose of relaying information to the on-board computer to position the NERF gun at the target and fire.

5.3.1 Subsystem #2: Fritzing Schematic

The Target Detection Subsystem is displayed as a pictorial representation in Figure 5.3.1 i. This picture, also created using the Frizting, shows how we plan to connect all the components of the Target Detection Subsystem to each other. It is far simpler than the Fire Control Subsystem. The first component to illustrate, is the webcam. We will connect the Logitech Webcam to the Intel Compute Stick via a USB connection. Fritzing does not have an accurate picture to represent the Intel Stick, the Logitech Webcam, or the USB connection so we needed to improvise. Ultimately, we made our own version of the connection. Though it looks different, it is important to note that we will not need a breadboard to hold the connection. The two components will be connected directly to each other via the USB port. On the right side of Figure 5.3.1 i is how we will implement the Ultrasonic Range Finder. Like the Logitech webcam, its connection is simple. Data received from the range finder will be sent to an analog pin on the Arduino. This data will be used to determine how far away a target is from the robot and used to align the gun accordingly. As shown in Figure 5.3.1 i, the Arduino is also going to be responsible for powering the sensor.

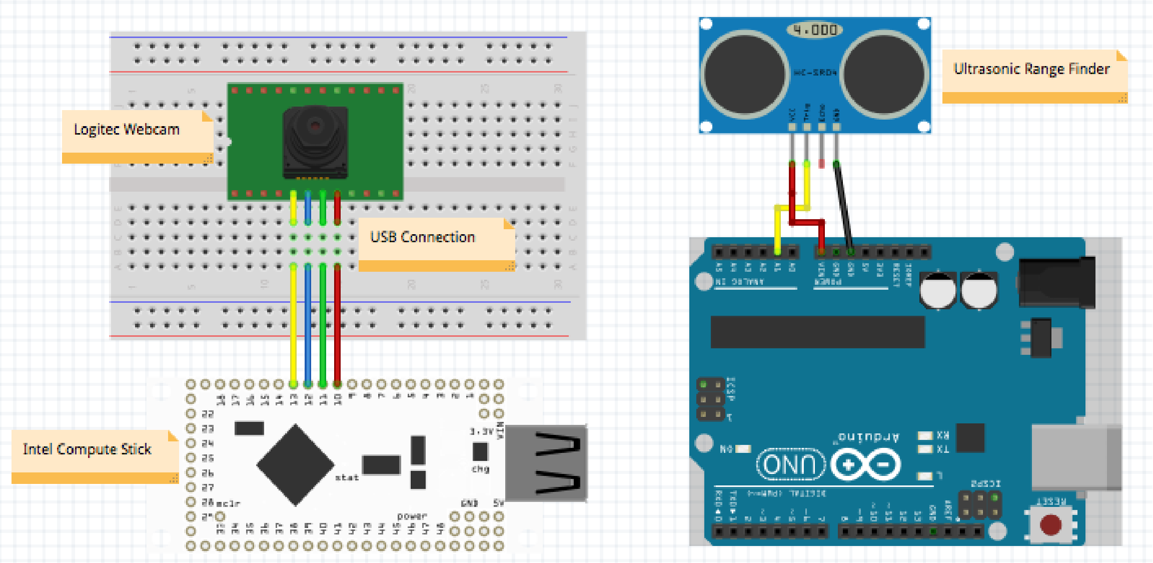


Figure 5.3.1 i: Fritzing Schematic of the Target Detection Subsystem

5.3.2 Subsystem #2: Test

Figure 5.3.2 i shows the in-lab breadboard test of the Target Detection subsystem. This breadboard test was conducted to ensure functionality of specific parts that are in control of sensing, detecting, and tracking targets. Setting up the circuit for the ultrasonic and achieving compatibility with the webcam affects the successful prototyping of the printed circuit board. In Figure 5.3.2 i the ultrasonic sensor, the 6-volt battery pack, and the webcam can be seen. Unless further design changes are made, these are the only components that will comprise the target detection system. The system should work equivalently as well once implemented in the final stages of development. Overall, the breadboard test of the Target Detection subsystem proved that the ultrasonic sensor and webcam can work together in the system. There were no major issues when conducting the test however, addition work will need to be done to refine the ultrasonic sensor performance and integration into the all-up software. All issues will be resolved by the final development stages.

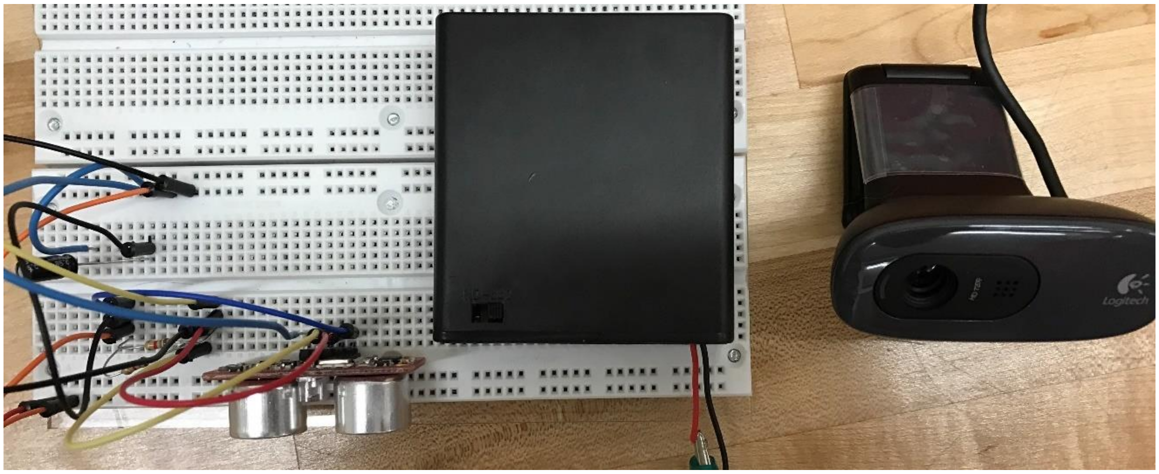


Figure 5.3.2 i: Target Detection Breadboard Test

5.4 SOWTWARE PROCESSING (THIRD SUBSYSTEM)

The Processing Subsystem is the brain of our weapon system. This subsystem includes the computers on board the robot. It is responsible for determining a variety of things. Most important, it must run efficient algorithms to determine if there is a target in front of the robot. This functionality, combined with the pan/tilt mechanism, is what determines if our robot will successfully compete in the competition. The Processing Subsystem is also responsible for angling the gun to point at the target and fire when ready. It will send electrical signals to maneuver the gun's position to align with the target. The other two subsystems are useless without the third, the brain. This is like the inner-workings of the human body. We think of something that we want our body to do and then it sends electrical signals to get the task done. The autonomous function of our robot heavily relies on the processing power of our on-board computers.

5.4.1 Subsystem #3: Fritzing Schematic

Figure 5.4.1 i is a pictorial representation of the Processing Subsystem. The Intel Compute Stick is powered by a rechargeable lithium ion battery. This power supply will give the stick the necessary amount of power needed to support the stick's computation and to power the webcam. The Intel Compute Stick will be connected the Arduino by USB. This connection provides an extension to pass vital data from the Intel Compute Stick to the Arduino. This data is used to determine what changes in the state of the motors, servos, or gun need to be adjusted to make an accurate shot. The Arduino is powered by the 4x AA batteries. This power supply will distribute 9 volts to the system.

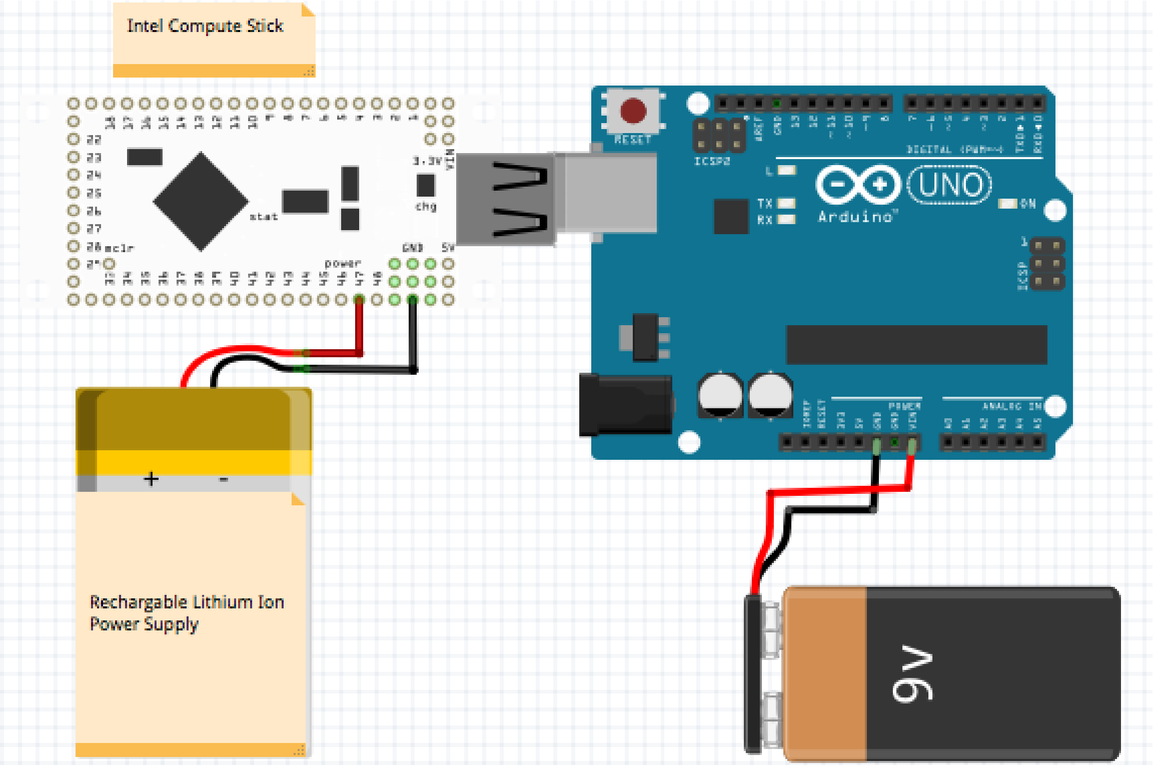


Figure 5.4.1 i: Fritzing Schematic of the Software Processing Subsystem

5.4.2 Subsystem #3: Test

Figure 5.4.2 i shows the in-lab test of the Software Processing subsystem.

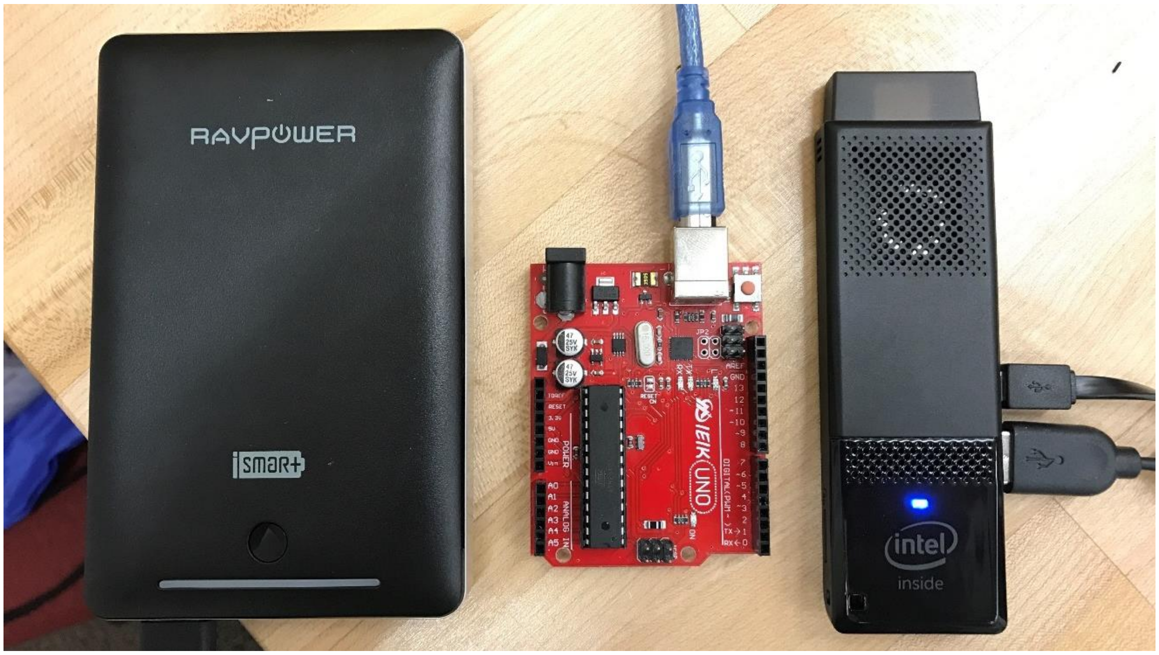


Figure 5.4.2 i: Software Processing Subsystem (In-lab test)

This test was conducted to ensure functionality of specific parts that are in control of processing the software to detect the objects and control the peripherals. Compatibility was achieved with the Intel Stick, Arduino microcontroller, and the rechargeable battery. The 9-volt battery is omitted from the picture, however, it is compatible as well. Unless further design changes are made, these are the only components that will comprise the Software Processing system. The system should work equivalently as well once implemented in the final stages of development. Overall, the test of the Software Processing subsystem proved that the rechargeable battery can power the intel stick while the intel stick and microcontroller communicate information. There were no major issues when conducting the test however, addition work will need to be done to develop and order a custom circuit board that is compatible with the Intel Stick. All issues will be resolved by the final development stages.

5.5 SOFTWARE DESIGN

For Green Team / Group 2, effective software design is absolutely crucial – almost the entire basis of the project is on autonomous targeting which focus' heavily on software. The software will need to be user-friendly and support a range of functions which will be explained in the program overview section.

5.5.1 Program Overview

Although the final system will work to successfully compete against enemy robots, the program itself will be highly modular and adaptable. The focus of Group 2 is to create a program that will recognize and follow a predetermined object. It is important to get proper software up and running to support the various hardware peripherals that will be mounted on the robot. There will need to be code that operates the pan and tilt of the NERF gun, the electric trigger, the flywheels and the ultrasonic. This software will talk directly to the target recognition code on the intel stick by means of the printed circuit boards surface mounted microcontroller. Figure 5.5.1 i shows the program overview developed by green team's computer science group.[39]

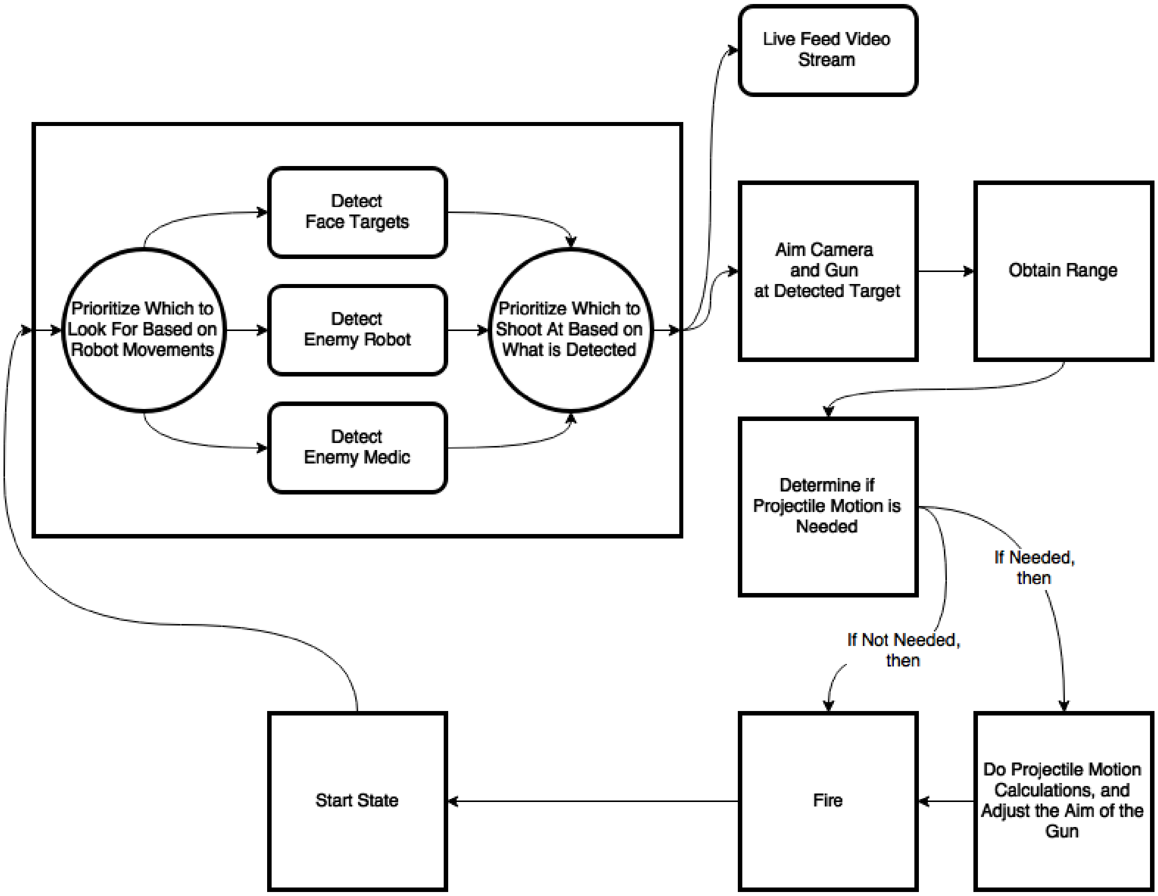

Live Feed Video Stream
Detect Face Targets
Prioritize Which to Look For Based on Robot Movements
Detect Enemy Robot
Prioritize Which to Shoot At Based on What is Detected
Detect Enemy Medic
Aim Camera and Gun at Detected Target
Obtain Range
Determine if Projectile Motion is Needed
If Needed, then
If Not Needed, then
Start State
Fire
Do Projectile Motion Calculations, and Adjust the Aim of the Gun

Figure 5.5.1 i: Program Overview

5.5.1.1 Template Matching

Lockheed Martin has specified that the targets will be a human face on a poster board mounted about four to six feet above the ground. Identifying the targets involves the ability to do facial detection on webcam video.

Template Matching uses a template image to find an image inside of another image. For example, have a close picture of a person’s face and another image where the same person’s face is located at different arbitrary spot in the image. Template matching could find that person’s face in the image.

The templates images that will be used would not be faces for the detection of the enemy robot. Instead, images of possible components of an enemy robot will be considered. Possible components considered could be different wheel types, structural components and different cameras.

The negative impact of template matching is that is seems to only consider matches that are the exact size and look of the provided templates. This means that if a template of a wheel that is 20 inches is used, it would only look for wheels that match that size. If there is a smaller or larger wheel that is seen, it would not consider the wheel as a match because it is not the same size. To get around this issue, a possible solution would be to take either the template image or the video frame and scale one of them until a match is made.

One of the possible ways to shorten this process is to determine what the largest and smallest wheel size will appear on the camera input. Then, the template image will only be called from the largest to the smallest it possibly could be.[39]

5.5.1.2 Arduino Software (IDE)

Green Team / Group 2 will use the open-source Arduino Software (IDE) to program the microcontroller. Using this software makes for simple and rapid prototyping to take place using any low-cost Arduino board. The Arduino software is compatible with the Windows operating system. This is important because the software will need to work with the selected intel compute stick hardware. The Arduino software is also written in Java which makes for easy understanding.

5.5.1.3 OpenCV

One of the main focuses of the software for the project is how the camera will be able to detect an object that is in front of it. Currently, a standard webcam can just output the video onto the screen. The only information that is gathered is the video and the camera does not know what that object is. To figure out exactly what objects are being shown from the camera, you must use software. One of the most commonly used and trusted libraries to accomplish this task is Open Source Computer vision or OpenCV. OpenCV is an open source C++ library that is primarily used for image processing and computer vision, originally developed by Intel.[40] It has a large amount of open source material and projects that currently use it. Therefore, there are a large amount of resources available for use to learn from and reference. The OpenCV library gives Green Team / Group 2 the ability to access several thousand, optimized computer vision and machine learning algorithms that are fundamental and cutting-edge.

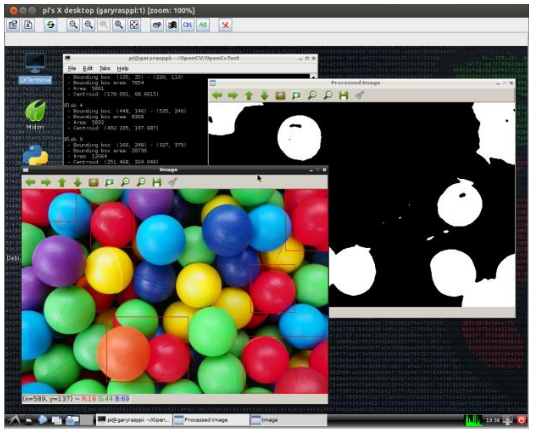
OpenCV is most commonly used in C++, but there are also versions for C, Python and Java. OpenCV is compatible with C++, C, Python, Java, and MATLAB. OpenCV also supports the operating systems that Green Team / Group 2 will use while developing the targeting software.

The library has more than 2,500 optimized algorithms, which includes algorithms for detecting faces, objects and detecting moving objects.[41]

As previously discussed, the group has been tasked with detected targets which may consist of faces, obstacles, robots, or human figures. OpenCV provides the best possible capability for doing this. The algorithms that are available to use will help in all manners of detecting such objects including faces, general object detection, the ability to track static and mobile objects, and identifying similar images from an image database.

5.5.2 CVBlob Library

The focus for Group 2 will be looking at how to detect an object of a specific color. One way to accomplish this is by using a method called blob detection. Blob detection is based off a detection method called the Laplacian of the Gaussian. This process can be used to detect rapid changes in the image and to find out the edges of what is in the image. This method is out of the scope of the project to thoroughly explain. However, OpenCV has a library called cvblob which uses this process described. The cvblob library is capable of distinguishing different colors from the video feed of the camera. From the predetermined color that the user sets, the camera will process the image and will turn anything that matches the color to white. Anything else that is not in that color range will be turned into black to distinguish the two for each other. This library breaks down the video image into a separate processed image that is viewable to see what the cvblob is doing. Figure 5.5.2 shows an example of what cvblob is capable of.


pi's X desktop (garyraspi:1) [zoom: 100%]
pi@garyraspi: ~/OpenCV/OpenCvTest
File Edit Tabs Help
Processed Image
Image

```
 - Bounding box: (135, 25) - (220, 113)
 - Bounding box area: 7654
 - Area: 5861
 - Centroid: (178.891, 68.6615)

Blob 4
 - Bounding box: (448, 146) - (535, 240)
 - Bounding box area: 8360
 - Area: 5892
 - Centroid: (492.195, 197.687)

Blob 9
 - Bounding box: (166, 248) - (327, 375)
 - Bounding box area: 20736
 - Area: 12064
 - Centroid: (251.468, 324.048)
```

```
(x=589, y=137) ~ R:18 G:44 B:69
```

Figure 5.5.2 – Sample Blob Detection of Red Color (awaiting approval)

5.5.3 Background Subtraction Library

Another method of detecting objects with using OpenCV is by using background subtraction. This is a commonly used technique for generating a foreground mask by using static cameras. Background subtractions works by calculating the foreground mask by performing a subtraction between the current frame and a background model. The first step in this process is to take an initial model of the background. The second step is that this model will be updated to see if there are any changes in the scene. This is then used as the base background model and will be then subtracted from the original current frame to detect the object in the scene. Figure 5.5.3 shows a sample of how this process will work.

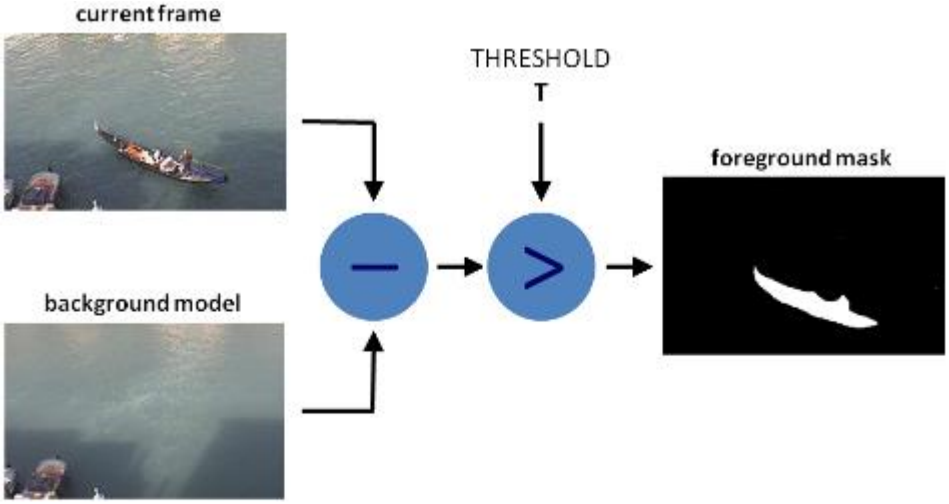


Figure 5.5.3 – Example of Background Subtraction

One of the key disadvantages of background subtraction is that it works best when the scene is static as opposed to constantly moving. Also, if more objects are introduced to the image while this process is taking place, it can cause issues of detection. If the new objects that are introduced to the scene and stop, they will continue to be detected making it difficult for new objects that pass in front of them to be seen. For the scope of this project this will not be a major issue because it will be mainly used in a controlled environment with limited number of objects entering and leaving the scene.

5.5.4 Face Recognition Library

OpenCV also features a library that can be used for face recognition. Three different classes are available in OpenCV and they are Eigenfaces, Fisherfaces and Local Binary Patterns Histograms. Generally, face recognition is based on the geometric features of the human face. The most useful algorithm to use will be the Local Binary Patterns Histograms.

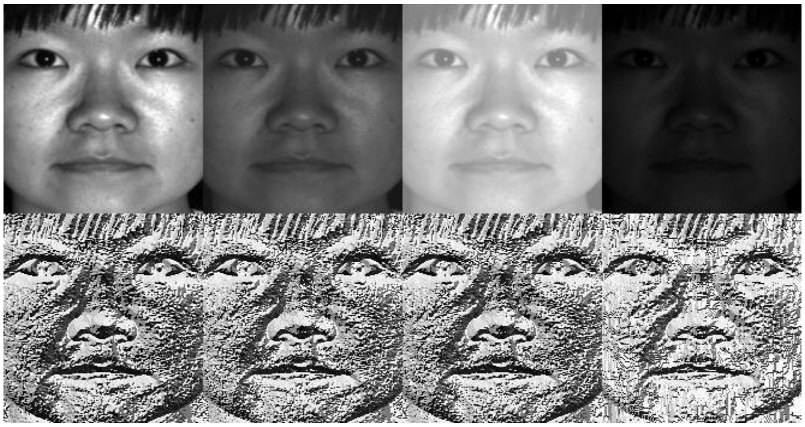


Figure 5.5.4 – Example of Facial Recognition

The main disadvantage of Eigenfaces and Fisherfaces is that you need to supply some multiple images of the same face with good lighting to get an accurate reading. However, Local Binary Patterns Histograms does not look at the image but instead looks at each individual feature of a face as an object. The basic idea of Local Binary Patterns is to summarize the local structure of an image by comparing each pixel with its neighborhood.[42] This is done by comparing the intensity of the central pixel with the intensity of its neighbor pixel. If the central pixel has a higher than its neighbor, it will become a 1 and a 0 if it is not. Figure 5.5.4 shows an example of this method described.

5.5.5 Methods

As shown in Figure 5.5.5, each state associates a different priority to each targeting mode by assigning it a number (where a low number means it has a high priority and a high number means it has a low priority).

For the Moving State, the prioritization is in the order to first detect face targets, second detect the enemy medic, and third detect the enemy robot. For this state, we decided that detecting the face targets should be of the highest priority since they will most likely be the easiest targets to identify, aim, and fire at especially while the robot is in motion. It must be taken into consideration that the face targets can only be hit twice. Therefore, once those targets have been hit twice it is not practical to shoot at them again. This means once the face targets have been hit twice the Moving State considers the enemy medic targeting mode as its primary detection mode followed by the enemy robot targeting mode.

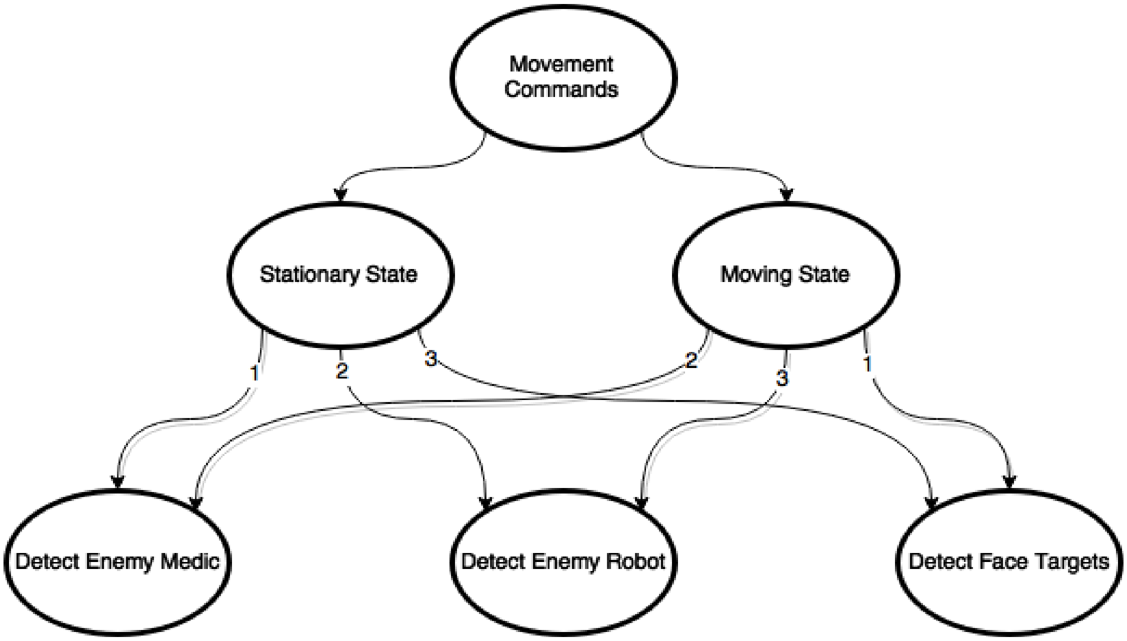


Figure 5.5.5: Targeting Mode Priority States

The enemy medic has a higher priority than the enemy robot because the points gained by hitting a medic are significantly larger than hitting a robot. Though, it is most probable that once in this phase of the Moving State the detection of the enemy robot will be mostly used. Even though it has a lower priority than the medic it will mainly be used because the medic will only be on the field once for a small amount of time.

For the Stationary State the prioritization is in the order of first detect the enemy medic, second detect the enemy robot, and third detect face targets. We wanted to consider the enemy medic and enemy robot targeting modes over the face targeting mode in the Stationary State because the algorithms we want to use for detecting motion are of better use to us when the robot is stationary rather than when it is in motion. Therefore, in the Stationary State we prioritize moving objects over detecting the face targets because of the convenience. If there is more than one object moving on the field this means that both the medic and the robot are on the field and moving, so we can shoot at them both. But, the enemy medic will be prioritized to be shot at first over shooting at the enemy robot. The last target to shoot at would be the face targets so long as there is not an enemy medic or enemy robot visible and so long as the face targets have not already been hit twice.

To identify object as being inside of the arena the targeting system will need a general sense of where its boundaries are located. The way we see it there are two potential ways to solve this problem. First, the arena will be taped off so that the driver knows the boundaries of the arena by eye sight. So, this would give us the opportunity to use the camera to also identify the tape and determine the boundaries of the arena. Second, would be to do mathematical calculations based on the starting position of the robot and the known size of the arena and through use of the range sensor.

Since, the arena is 40 feet in length and 20 feet in width this helps get the distance the robot is away from the right boundary and the top boundary by using the formulas $20-x$ and $40-y$ respectfully.

To calculate the x and y variable we will use the circumference of the robot's wheels. The wheels will be of some constant radius allowing us to use the circumference of a circle formula $C = 2\boldsymbol{\pi}r$. This circumference allows us to determine the distance the robot physically moves in the x and y directions, therefore giving is the values for the x and y variables.

So, the x and y variables represent the distances and we plan to initialize these values at that point (0, 0) at the arena's bottom left corner. We want to set the robot at that corner and then from there move the robot by rolling it to the spot at which we want our robot to start at. The tricky part of this and the key component is transforming those wheel movements into the distance that the robot has moved in the x and y directions. Once we can get those distances based on revolutions of the wheels then all the other values fall into place due them being dependent on the x and y variables.

Once we are aiming at the object we obtain the range via the ultrasonic sensor and compare that value with the *R* value and if the range is greater or equal than *R* then the object is outside of the arena and if the range is less than *R* then the object is inside of the arena (therefore shoot at it).

This is our solution to finding out if object we detect are inside of the arena or not. But, here we state some cons to this method that have some work arounds but also may not. This method obtains the *x* and *y* variables via the circumference of the robot's wheels. But, depending on the motor sizes and how much torque they can produce slippage is a factor that may make our *x* and *y* variables here have inaccurate values. This method is under the assumption that the robot will be moving perpendicular to its last direction. The means the robot is always facing forwards and the whole robot moves forward, left, right, and backward at always the same orientation. We decided this to avoid having to consider the wheel being on a slant when the robot takes a turn. This method could also become affected by rounding errors the make our distance calculation less accurate or even the circumference of the wheel is not measured to a tee therefore could affect values as the match progresses.

5.5.5.1 Arduino Pin Functions

The many digital and analog pins on the microcontroller will serve mostly to connect the peripherals including the servo, ultrasonic, and trigger. These assignments will be broken up for the Arduino microcontroller as seen in figure 5.5.5.1.

Item	**Pin #**
Pan Servo	8
Tilt Servo	9
Firing Indicator LED	12
USB Indicator LED	11
Mode Indicator LED	13
Electric Trigger Pin	7
Flywheel Trigger Pin	6

Figure 5.5.5.1: Arduino Analog Pin Assignments
(subject to change)

5.6 SUMMARY OF DESIGN

Stability and Vibration avoidance as it relates to the webcam and image processing. The webcam should not be interfered with by the servos, gun, robot movement, etc. This section will summarize the hardware and software design specifically including all finalized part decisions and software spec decisions.

Section 5 outlines the entire design of the overall project by breaking down each of the components and placing them into three different subsystems. These three subsystems will all work together to create one main system and complete the desired task of the project. The first subsystem mentioned was the fire control subsystem. This is the system that will handle the actual firing of the weapon. Along with this, it is also the system that will move the weapon from left to right with the pan and tilt servos. The next subsystem is the target detection system. The target detection system is what will give the weapon the eyes and how the weapon will know which direction to orient itself in to hit the target that is in front of it. To do this, there will be the webcam that will provide video that the user will be able to see objects. The ultrasonic sensor in this system will be what will receive the distance information from any object that moves in front of the weapon. The ultrasonic sensor will be mounted on the nerf gun itself so it will act as the sights of the weapon without the aid of a human operating the weapon. The webcam will be in a fixed position so that it can see everything that is in front of it. This will also prevent that webcam from getting dislodged from the system. The final subsystem is the software processing subsystem. This is the most vital part of the entire overall system because this is essentially the brains of the system. Here is where all the information that is gathered through the target detection system will be analyzed. After this information is analyzed it will then relay that information back to the fire control subsystem to tell the weapon when and where to fire. The decision was made to split the system in three parts is to ensure that it will be modular. This is so that it can be easily changed and adapted to fit the need of any platform. Also, it allows for improvement with individual components because you will be able to simply take it apart without destroying the entire system.

Altogether these subsystems must all work individually as well as with one another. If one of the subsystems fail, the entire system will fail. It is crucial that each subsystem is in working condition to maximize the success of the overall goal of the entire weapon system. Once this is completed the entire system will be mounted on a platform, like a piece of plywood for example. This then can be mounted on any robot or other platform that can support the size and weight of the system. Overall this was determined to be the best way to accomplish the goal of this project.

6.0 PROTOTYPE: PROOF OF CONCEPT

6.1 INTEGRATED SCHEMATICS

This section will have the eagle schematic. The schematic will be explained and divided into sections.

6.1.1 Top Level Schematic

Figure 6.1.1 i shows an early draft (revision 1) of the top-level Eagle schematic for the preliminary design of the printed circuit board. This schematic shows all the to-be-integrated systems. Some highlights from the schematic include areas for several peripherals to be connected.

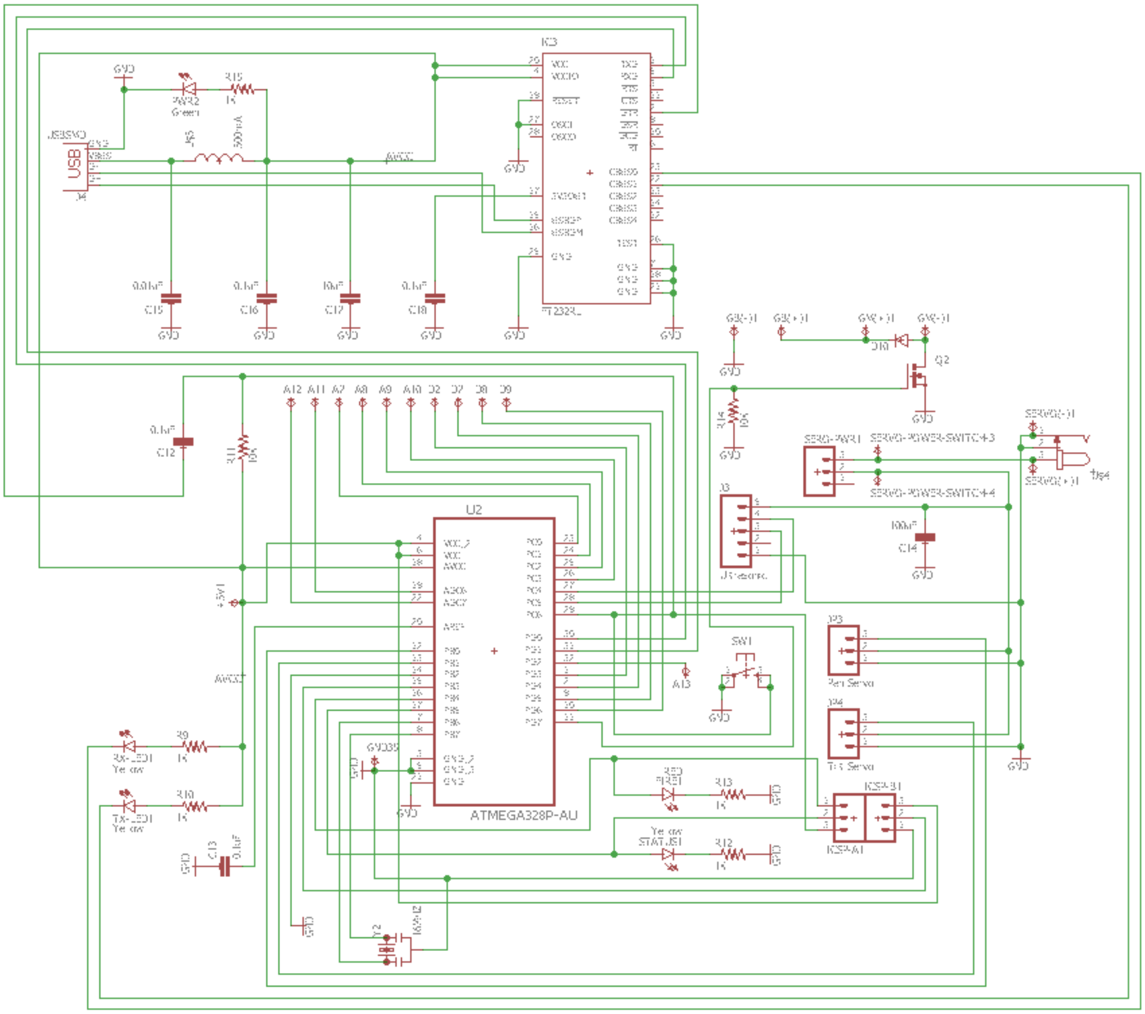


Figure 6.1.1sf i: Top-level Schematic (rev 1)

The robot needs connections for the pan and tilt servos so there are two separate 3-pin areas for the servos to be connected. Additionally, there is an area for the ultrasonic sensor to be connected. The heart of the circuit board will be the ATmega328 microprocessor. The surface mounted microprocessor will allow USB programming and will talk to the servos and ultrasonic sensor. The USB connection will be used to talk to the Intel Stick. This is very important because the system will need to constantly receive and transmit information to controller the servos and transfer ultrasonic sensor data.

6.1.2 USB Input to FT232RL Pins

The first part of the overall schematic that is needed is the USB input. A 5 pin Mini-B USB connector will be used. This connector is generally used for external peripherals. It is a smaller, more compact USB connection type with low cost.

This input is necessary to be able to load our program onto the microcontroller. The output of this USB input will need to be converted to serial. To do this the output is connected to a USB to serial UART interface, the FT232R. A schematic of this part of the printed circuit board is shown in Figure 6.1.2 i.

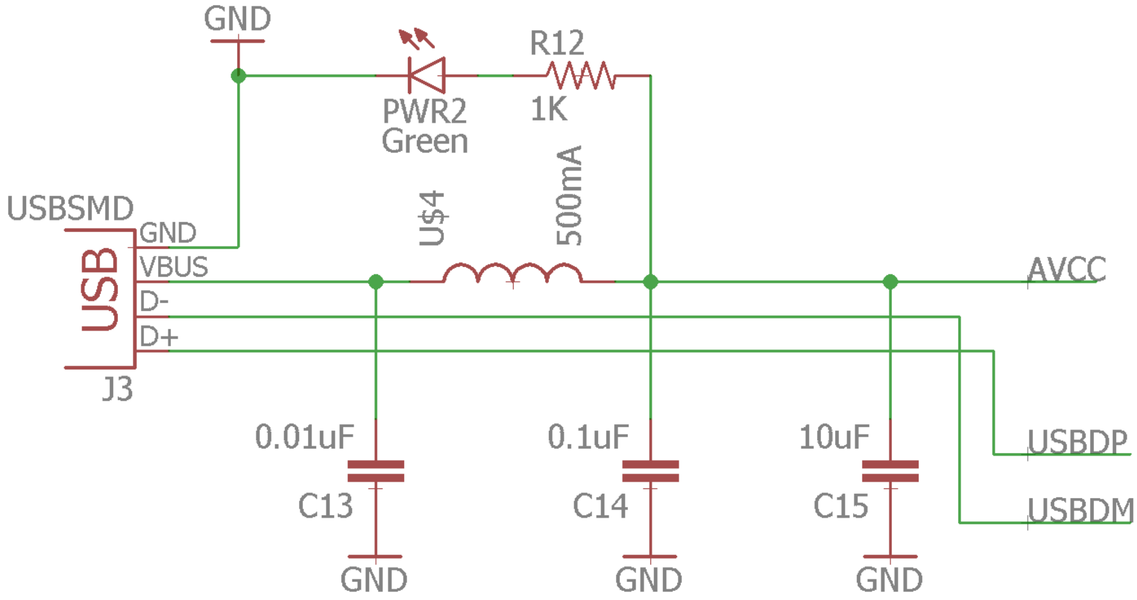


Figure 6.1.2 i: USB Input Schematic

6.1.3 FT232RL Main Input / Output

The USB input must now be converted to serial. This is done by using a FT232R chip. The output of this chip will be connected to the microcontroller.

The FT232R is perfect for our application. It has a lot features that will benefit us. For instance, the entire USB protocol is handled on the chip itself. This means that there is no USB specific firmware programming needed. Since it has a fully integrated clock generation, this chip also does not require an external crystal. It also converts the 5V input into a 3.3V for USB I/O.

In Figure 6.1.3 i, is a schematic of the FT232R portion of this project.

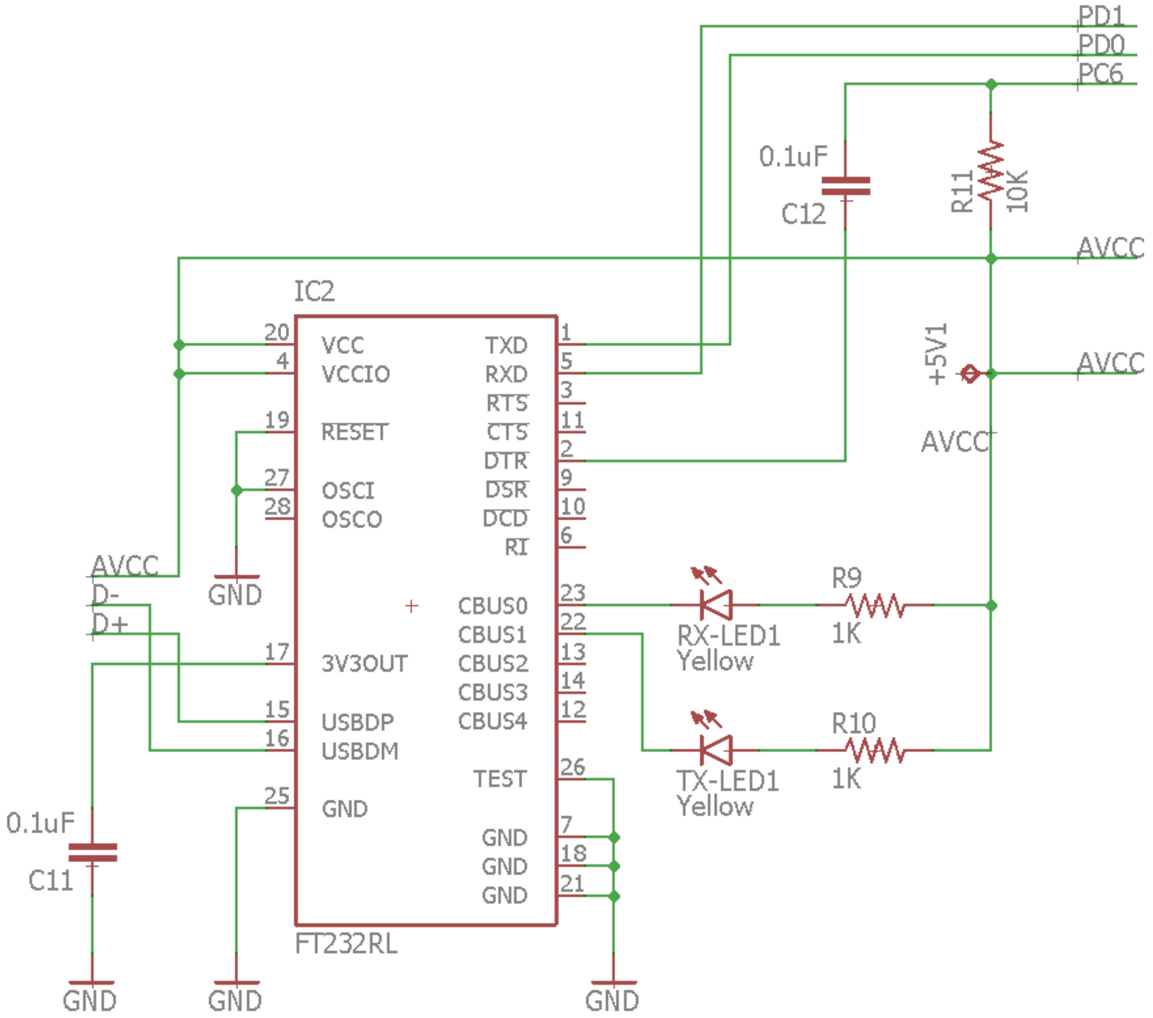


Figure 6.1.3 i: FT232RL Schematic

6.1.4 Microcontroller Main Input / Output

Figure 6.1.4 shows all the currently planned connections to the microcontroller. As can be seen in the labels, there are pins going to the ultrasonic sensor, the servo motors, the electric trigger, and a crystal oscillator among other things like power and the USB controller. The crystal oscillator will create an electric signal with a precise frequency of 16 megahertz to provide a stable clock signal for the integrated circuit.

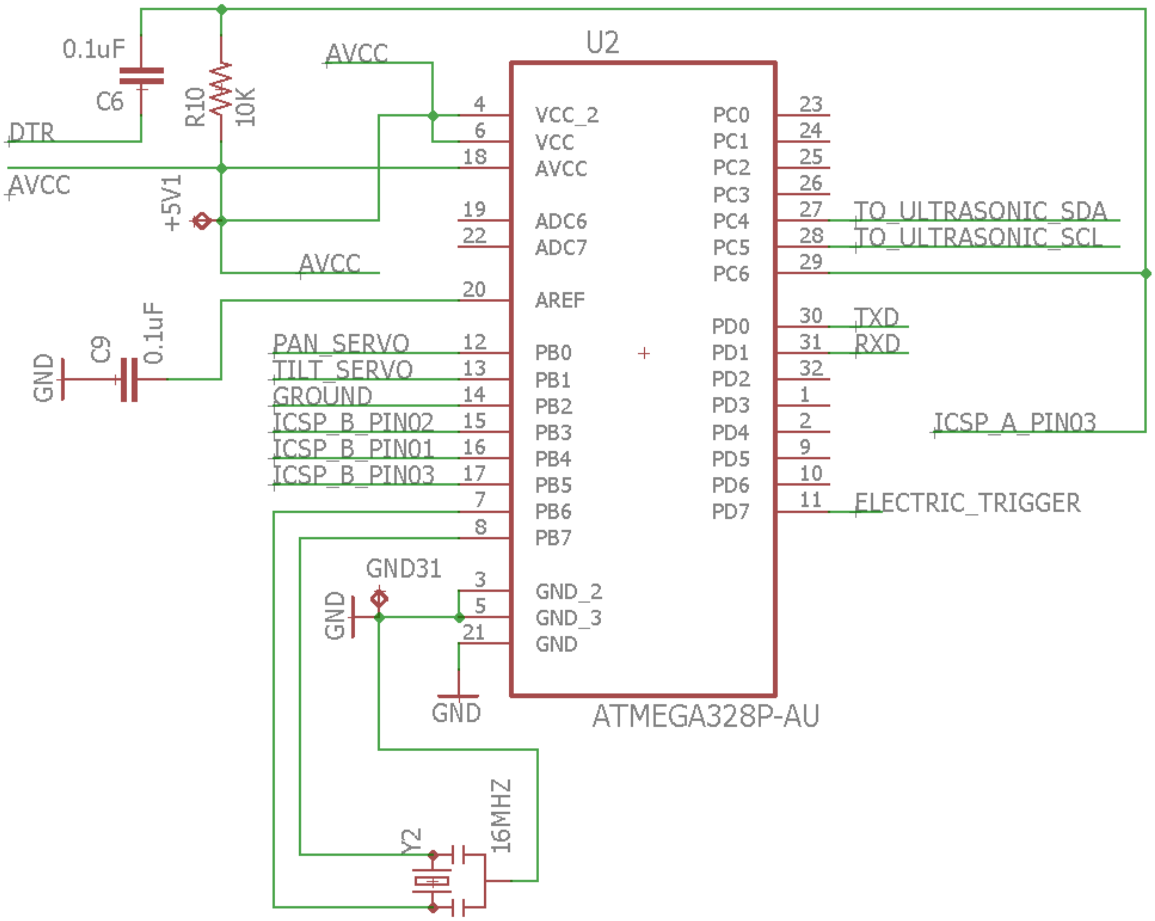


Figure 6.1.4 i: Microcontroller Main Input / Output Schematic

6.1.5 ICSP Headers and LEDs to Microcontroller Inputs

For most purposes, it is uncommon to program controllers prior to them being soldered onto a printed circuit board. For that reason, “in-system programming” (ISP) headers usually are incorporated to support microcontroller programming. The microcontroller Green Team / Group 2 is using, made by Atmel, has a unique method for being programmed called “in-circuit serial programming” or ICSP. Because the board will be Arduino compatible, the ICSP headers will be of a 2 by 3 layout. 3 of the pins are allocated to break out the power, ground and reset pins which are needed to connect the programmer and re-flash the firmware on the board. Figure 6.1.5 i shows a small breakout section of several status indication LEDs and ICSP header pins.

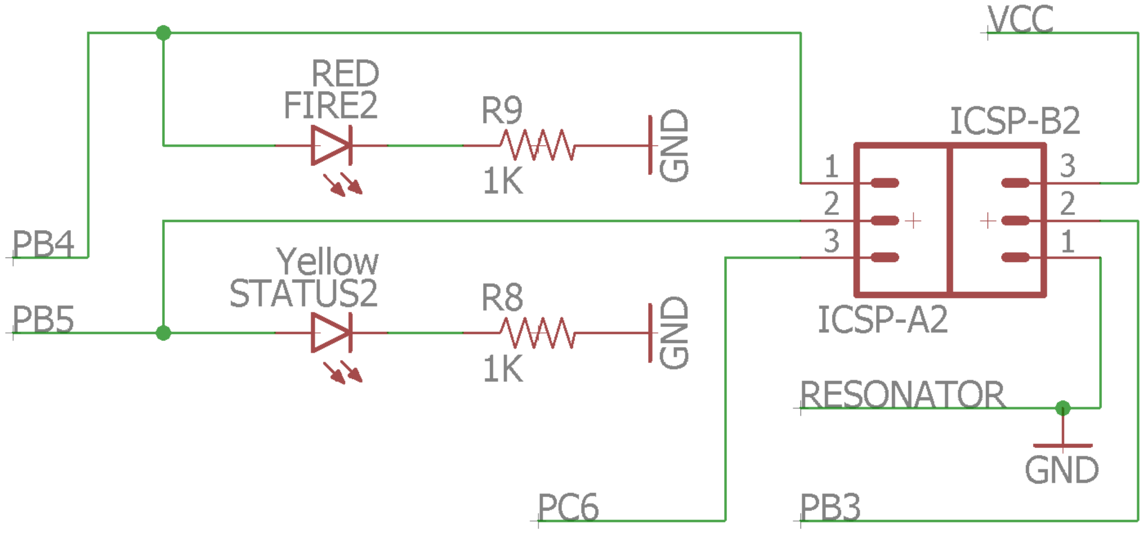


Figure 6.1.5 i: ICSP Header Pins and LEDs

6.1.6 Power Input

The pan and tilt servos, as well as the ultrasonic, need their own power supply. The output of the power supply will be connected to the inputs of the sensor and servos. A 6V AA battery pack will be used to power everything. It will be connected via a barrel connection. Figure 6.1.6 i shows the schematic for the power input.

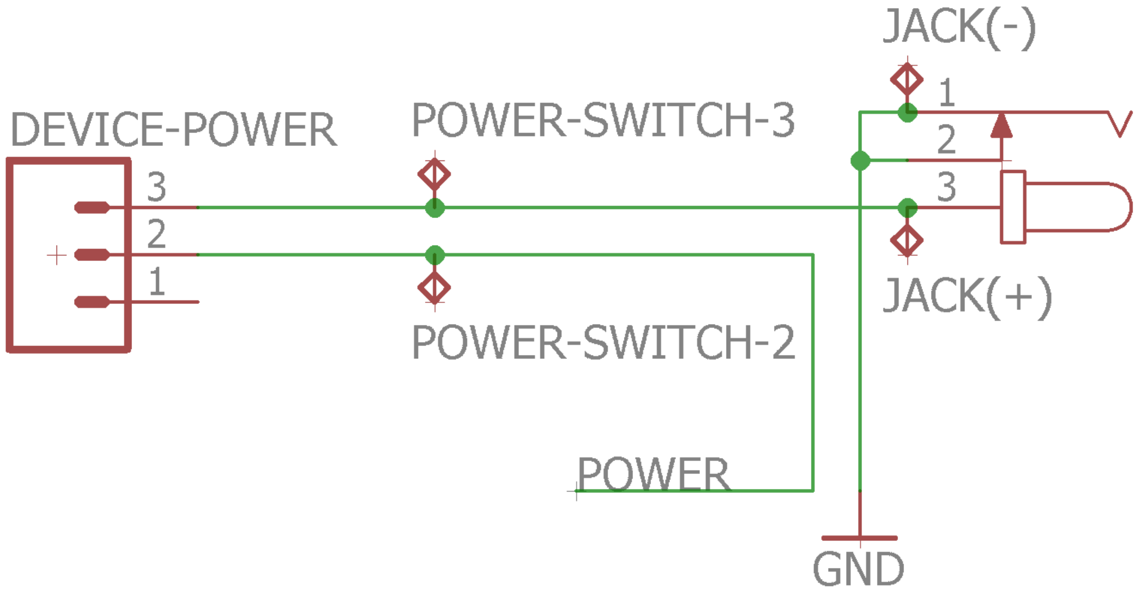


Figure 6.1.6 i: Power Input

6.1.7 Peripheral(s) Input to Microcontroller

There are several peripherals that will need to be connected to the PCB. This section includes Eagle schematics of the circuit sections that are responsible for integrating these components.

6.1.7.1 Flywheel and Trigger Motor Control

For the nerf gun to be autonomous both the trigger and the flywheel need to be controlled via the microcontroller. It is relatively simple to do this.

A transistor will be used as a switch. When the microcontroller sends a high signal to the switch portion of the trigger it closes and completes the circuit. This will then fire the gun. The microcontroller will then send a low signal to open the switch which will stop the gun from shooting. This same transistor switch is applied to the flywheel.

The plus and minus portions of the gun battery, trigger motor, and flywheel motor are used to complete the switch.

Figure 6.1.7.1 i shows the schematic of the trigger and flywheel motor control circuit.

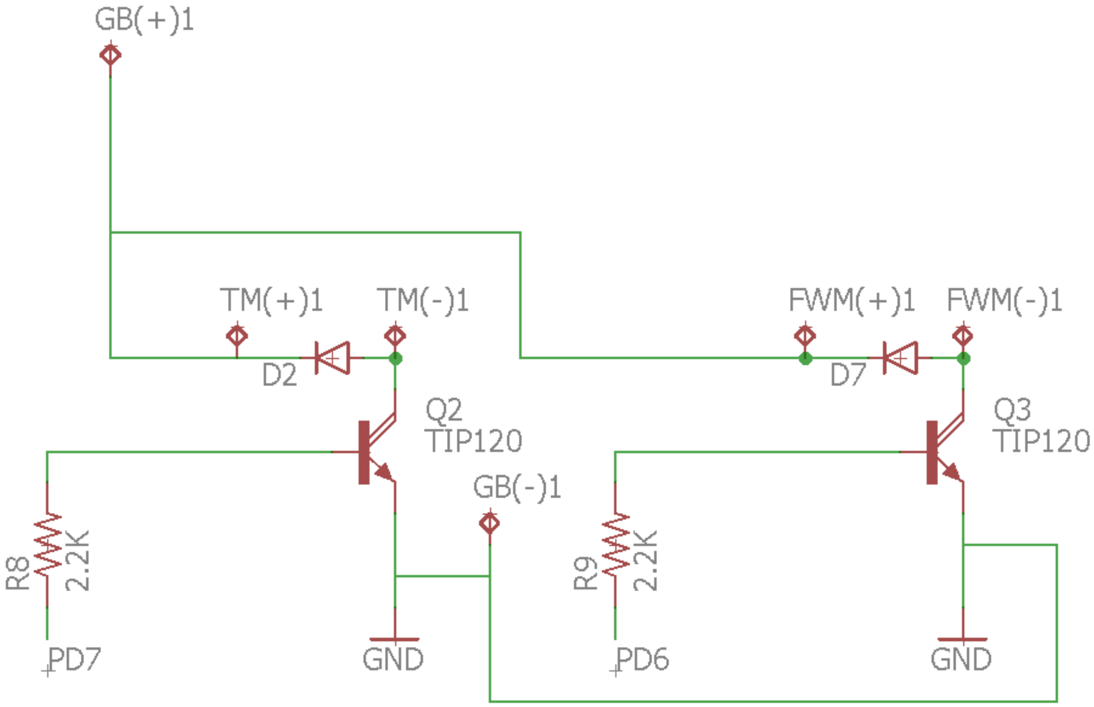


Figure 6.1.7.1 i: Flywheel and Trigger Control Schematic

6.1.7.2 Servo and Sensor Control

The last portion of the schematic is the servo and sensor control. These three peripherals are powered by the 6V battery pack. The data outputs of these devices are connected into various inputs of the microcontroller. The microcontroller can now send and receive data from these peripherals. Figure 6.1.7.2 i shows the schematic for the servo and sensor control.

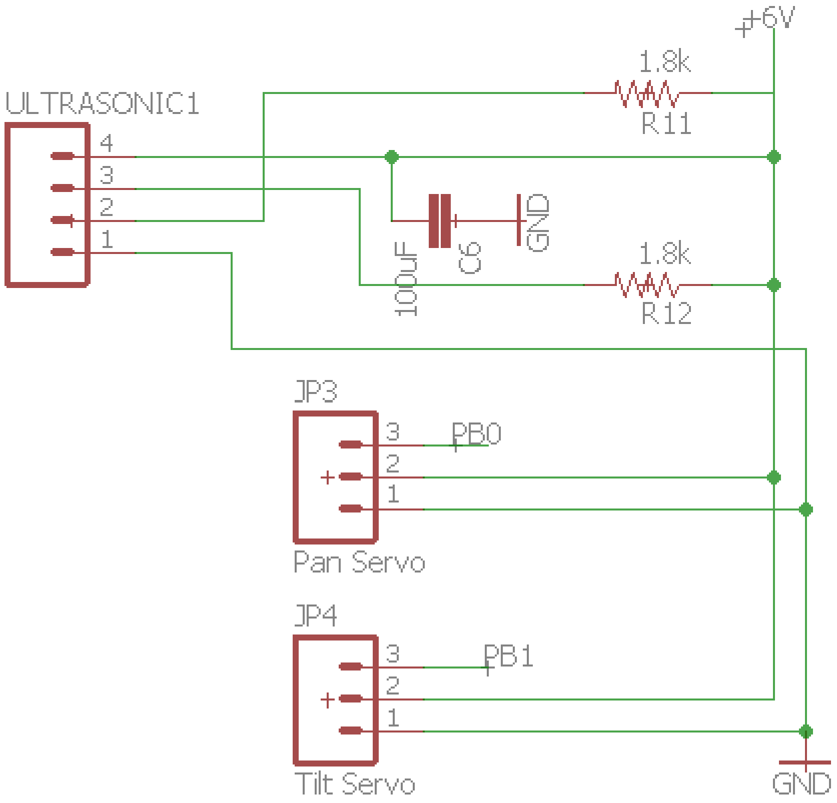


Figure 6.1.7.2 i: Servo and Sensor Control Schematic

6.1.7.3 Final Servo and Sensor Control

Since the ultrasonic sensor was replaced with a LIDAR sensor, the schematic has changed slightly. Figure 6.1.7.3 i reflects this change.

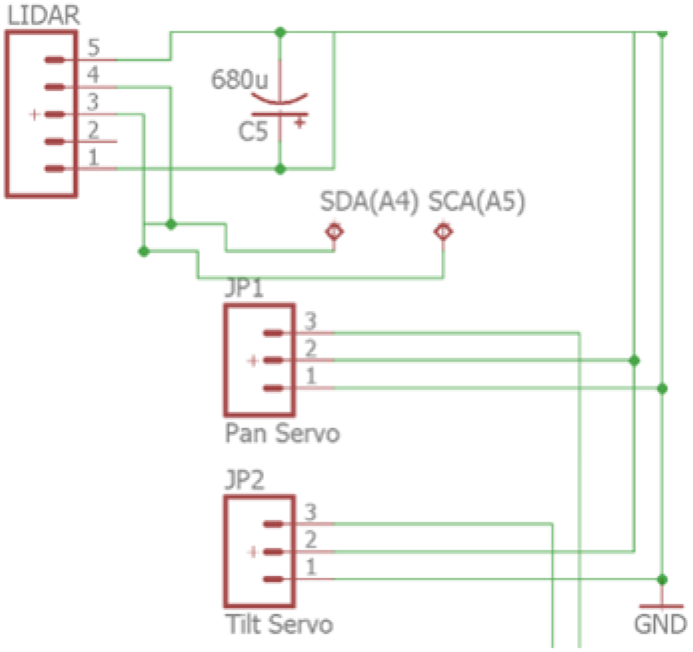


Figure 6.1.7.3 i: Servo and Sensor Control Schematic

6.1.8 PCB Layout

The next step after creating the schematic is to layout the PCB using the board view option. Figure 6.1.8 i shows a preliminary design of the PCB. This design is not finalized, but the finished product will be comparable to this.

Each part of the PCB needs to be placed in such a way that parts are not overlapping one another. You also want parts that need to be connected to one another are close enough to do so. If any of the same color traces do not touch the PCB layout will be sufficient.

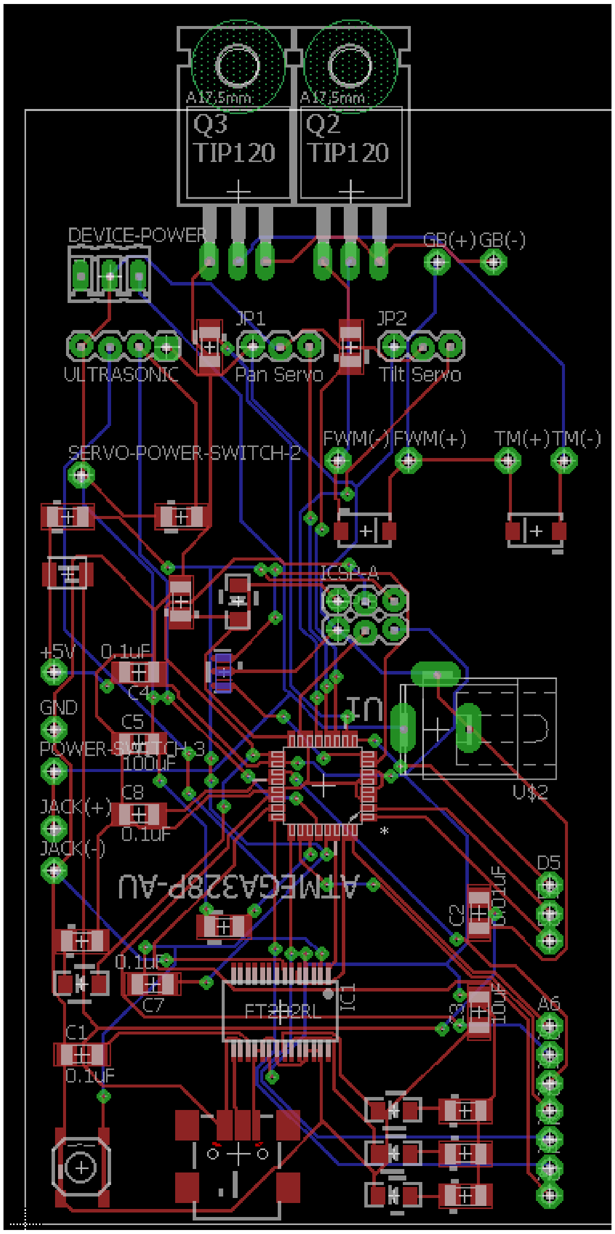
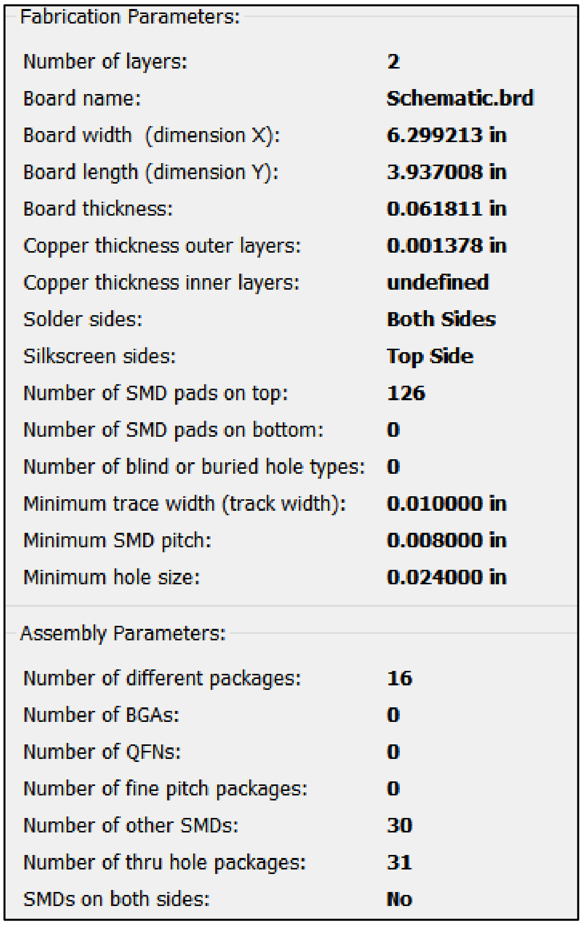


Figure 6.1.8 i: PCB Layout Prototype

The PCB consists of various elements that make up the circuit. This design is a two-layer design. This reduces the overall cost of manufacturing the PCB. The parts themselves are mainly surface mounted parts. These allow the PCB to be smaller in size, which again will save money. Figure 6.1.8 ii shows a detailed list of specifications.



Fabrication Parameters:

Number of layers:	**2**
Board name:	**Schematic.brd**
Board width (dimension X):	**6.299213 in**
Board length (dimension Y):	**3.937008 in**
Board thickness:	**0.061811 in**
Copper thickness outer layers:	**0.001378 in**
Copper thickness inner layers:	**undefined**
Solder sides:	**Both Sides**
Silkscreen sides:	**Top Side**
Number of SMD pads on top:	**126**
Number of SMD pads on bottom:	**0**
Number of blind or buried hole types:	**0**
Minimum trace width (track width):	**0.010000 in**
Minimum SMD pitch:	**0.008000 in**
Minimum hole size:	**0.024000 in**

Assembly Parameters:

Number of different packages:	**16**
Number of BGAs:	**0**
Number of QFNs:	**0**
Number of fine pitch packages:	**0**
Number of other SMDs:	**30**
Number of thru hole packages:	**31**
SMDs on both sides:	**No**

Figure 6.1.8 ii: Detailed PCB Specifications

6.1.9 Final PCB Layout

The PCB layout discussed in the previous section was modified after further critiques were made to it. Figure 6.1.9 i shows a side-by-side comparison of the layout and the actual PCB after completion.

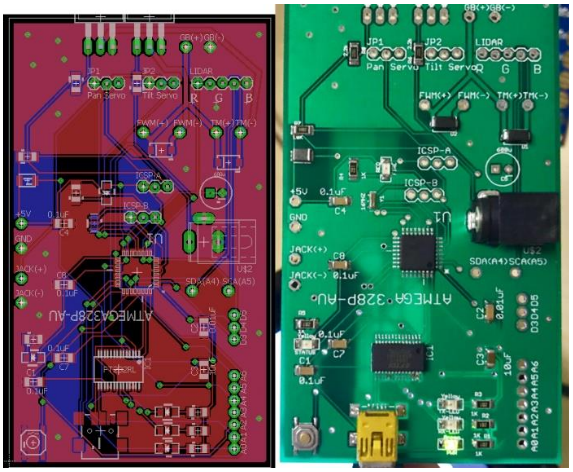
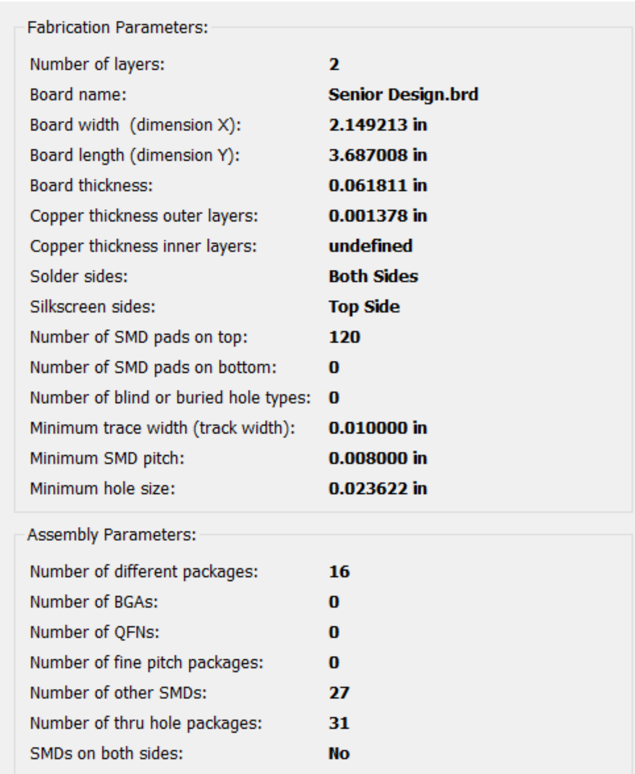


Figure 6.1.9 i: Side-by-Side Comparison of Schematic and Actual PCB

The finalized PCB comes with a new list of specifications. It is still a two-layer design with surface mounted parts. This insures that the PCB is very small and compact. These new specifications are shown in Figure 6.1.9 ii.



Fabrication Parameters:

Number of layers:	**2**
Board name:	**Senior Design.brd**
Board width (dimension X):	**2.149213 in**
Board length (dimension Y):	**3.687008 in**
Board thickness:	**0.061811 in**
Copper thickness outer layers:	**0.001378 in**
Copper thickness inner layers:	**undefined**
Solder sides:	**Both Sides**
Silkscreen sides:	**Top Side**
Number of SMD pads on top:	**120**
Number of SMD pads on bottom:	**0**
Number of blind or buried hole types:	**0**
Minimum trace width (track width):	**0.010000 in**
Minimum SMD pitch:	**0.008000 in**
Minimum hole size:	**0.023622 in**

Assembly Parameters:

Number of different packages:	**16**
Number of BGAs:	**0**
Number of QFNs:	**0**
Number of fine pitch packages:	**0**
Number of other SMDs:	**27**
Number of thru hole packages:	**31**
SMDs on both sides:	**No**

Figure 6.1.8 ii: Detailed PCB Specifications

6.2 PCB VENDOR(S) & ASSEMBLY

Printed circuit boards are one of the most important pieces of green team's project and any project. The PCB is where all the electronics of the project will be placed and it will be what will be controlling the project with the software. If any PCB fails in any product that is electronic, the product will not be able to function. The same rule applies for this project as well. To ensure the integrity of the project, high quality PCB vendors must be considered while keeping the cost in mind. A minimum of at least three PCBs will be needed as a precaution if one fails. Different factors will be taken into consideration to choose the best PCB vendor that satisfies the team's needs. One of the biggest factors will be the cost of the PCB. The cost of the PCB is determined by the size of the board that is needed and how many layers that is needed. Typically, there are two-layer boards and four-layer boards. For this project, a two-layer board will be used which will help keep cost lower. In the section below, different PCB vendors will be analyzed and compared to find the one that best satisfies the needs of the project. The estimated size for this project is about 6x4 inches or 24 square inches.

6.2.1 OSH Park

OSH Park is a company that is based in the USA and offers free shipping to anywhere in the world. Shipping of the PCB occurs 12 days after the order for a two-layer board. The shipping time of a four-layer board will be about two weeks after the order. This is a relatively quick service that is beneficial due to the time constraints of the project. Below are a list of specifications and pricing details:

- $5 per square inch, includes three copies, free shipping for two-layers
- Estimated price: $120 ($40 each)
- 12 days shipping time
- Board Thickness: 1.6mm
- FRF4 substrate, purple mask over bare copper
- ENIG (immersion gold) finish
- Minimum design rules: 6 mil trace clearance, 6 mil trace width, 13 mil drill size and 7 mil annular ring

6.2.2 Advanced Circuits (4PCB)

Advance Circuits (4PCB) is a company that is based in the USA and offers discounts for university students. For students, they offer a two-layer board for $33 apiece. With this offer you do not need to order a minimum number of boards which can help reduce the cost. 4PCB also offers a PCB design check tool for free. This can prove to be extremely beneficial because it can determine if there are any flaws in the design before the order is placed. Below are a list of specifications and pricing details:

- $33 per board (no minimum number required)
- Estimated price: $99 ($33 each)
- Lead-free solder finish
- Custom shape
- Minimum design rules: 0.006” line/space, no internal cut-outs, min 0.015” hole size (maximum 35 drilled holes per square inch)

6.2.3 ExpressPCB

ExpressPCB is a company that is based in the USA and offers free software that can be used to design your PCB. Using their software, you will be able to see if there are any faults in your design and you can get an instant quote. ExpressPCB offers a two-layer board for a flat fee. Below are list of specifications and pricing details:

- $166 for four boards (plus shipping)
- Up to 21 square inches or smaller
- Shipped in two days
- Tin/Lead finish
- .059” FR-4 substrate
- Minimum of 0.021” space between adjacent holes
- Hole location tolerance of 0.005”

6.2.4 PCB Vendor Conclusion

Each of the three vendors have their own advantages and disadvantages. The biggest difference between them all is the total cost of the board. However, this is all dependent on the final size of the PCB. OSH Park’s price is determined per square inch and does not offer a flat rate. However, both 4PCB and ExpressPCB offer flat rates, which can reduce cost because the final PCB size will not exceed these limits. The final decision on which vendor to use will be made in the spring 2017 semester.

During the Spring 2017 semester, a new PCB vendor was discovered. PCBWay was the vendor chosen for this project. This vendor was by far the cheapest out of the ones previously discussed. The cost for five boards was $10 (plus shipping). For this reason alone, PCBWay was the chosen vendor.

6.3 FINAL CODING PLAN

The software of our autonomous targeting system is analogous to the blood ruining through humans. Without proper functionality, all the effort put into the design is rendered useless. This illustrates Group 2 knows just how important an efficient algorithm is to operate a functioning autonomous targeting system.

6.3.1 Component Software Integration

Our coding will interact with all subsystems insuring that the correct process is being activated at the correct moments. Without proper timing, our robot could experience catastrophic difficulties. Figure 6.3.1 i displays a flowchart of when the program is on each subsystem. It illustrates when the code is manipulating specific components and gaining important information from other components. The whole process starts with the webcam on the target detection subsystem. The webcam will use the software to recognize a target. Once a target is detected, this will trigger the sensor to pulse and capture a range. Then, this information is sent to the processing subsystem for evaluation. The processing subsystem must check the positioning of the gun with respect to where the target, found by the webcam, is on the field. If the gun is not aligned with the target, more calculations need to take place. These calculations will compute the angle that each servo needs to pivot for the gun to be pointing at the target. Once aligned, the code will turn on the flywheel motor. The motor takes about a second or two to get to full speed. If the code does not wait for the flywheel to reach max speed, the gun will be inaccurate – shooting balls at different speeds. Once at full speed, the code will execute the fire command, which runs the convey belt that loads the balls into the flywheel mechanism.

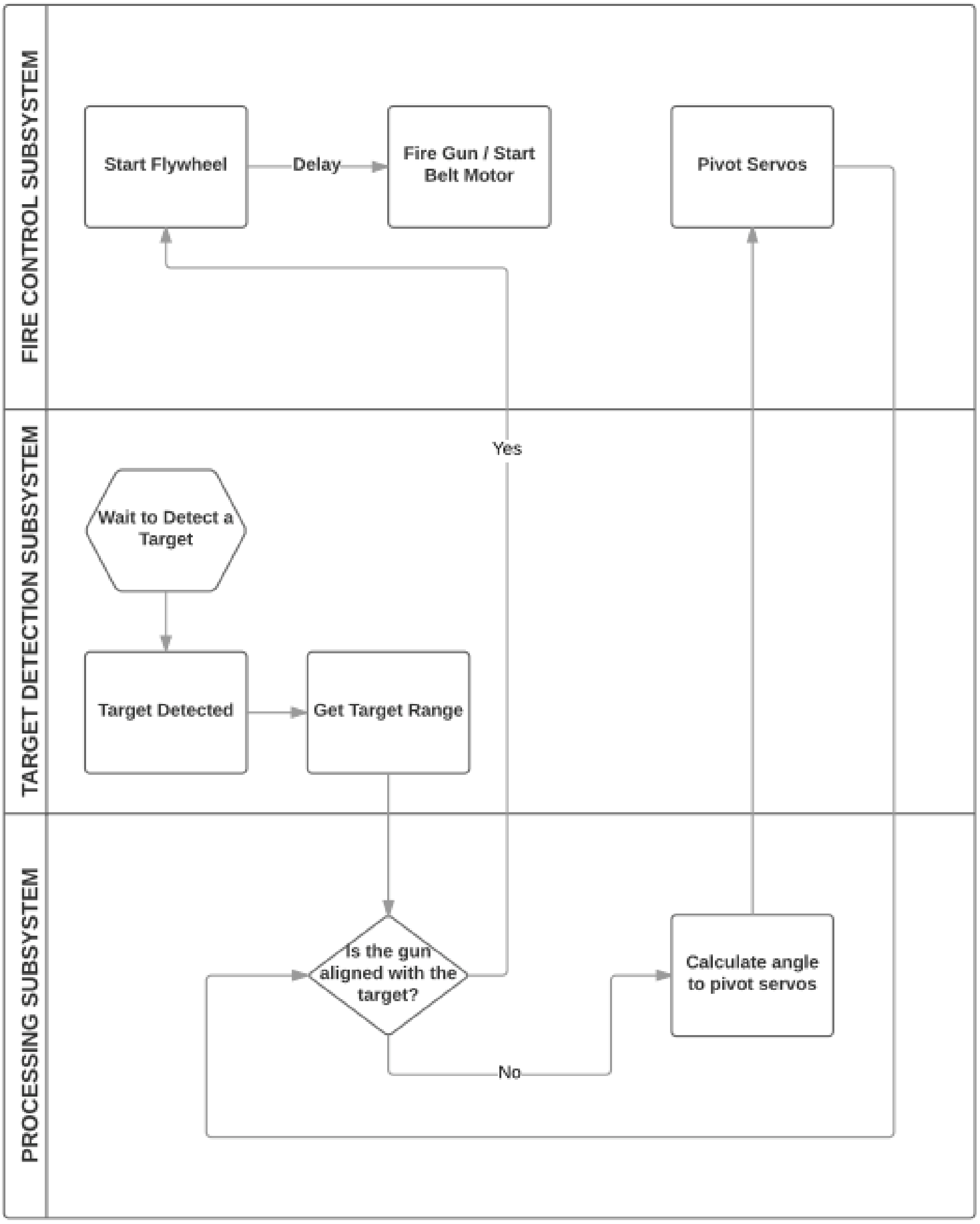


Figure 6.3.2 i: Subsystem Integration Flowchart

The main library we will be using to create the algorithm is going to be OpenCV. OpenCV is an open source computer vision library that interfaces with C, C++, Python and/or Java. Group 2 will primarily utilize the libraries that are written for C and Java. These classes will be helpful for recognition. That is the primary reason we are using these libraries. They are for the initial detection. The coding part that will be difficult is when aligning the pan/tilt servos. It is difficult to know the angle of adjustment needed between the camera and the current positioning of the gun. This area will require a lot of trial and error before our confidence in the accuracy of the gun.

6.3.2 Red Square Targeting

Our autonomous weapon system will perform under two different algorithms. The Computer Science group is creating one of them. They are creating the facial recognition algorithm, which is going to be implemented for the competition. This algorithm will recognize faces and identify the faces as targets. This algorithm is not going to be discussed in detail in this paper. This is because it is the objective of the Computer Science group. The other algorithm is the focus of Group 2.

Package	Class	Method	Description
org.opencv.core	Background Subtractor	getBackgroundImage(Mat backgroundImage)	Retrieves the area in the picture that is behind the foreground.
org.opencv.imgproc	Imgproc	GetRectSubPix (Mat image, Size patchSize, Point center, Mat patch, int patchType)	Retrieves pixels that match a given Mat.
org.opencv.objdetect	Cascade Classifier	Many different methods from this class.	In general, this class detects objects based on color.
org.opencv.imgproc	Imgproc	matchTemplate (Mat image, Mat templ, Mat result, int method)	This method takes in the image and the template and determines if they match.

Figure 6.3.2 i: The API used for the algorithms

The algorithm Group 2 is creating is the red square targeting algorithm. When a red square appears in the view of the webcam, the red square targeting algorithm must realize that a target is in sight and align the gun accordingly. The focus of the algorithm is to find the red square and relay its coordinates to the Arduino. This algorithm will be used for the demo. Figure 6.3.2 i is the application program interface (API) functions used from OpenCV in both algorithms to achieve efficient functionality.

7.0 TEST PLAN

7.1 HARDWARE TEST ENVIRONMENT

Hardware testing is one of the most important aspects of any type of project. This is when each individual component of the overall project will be tested. There will be different subsystems of the overall project and each subsystem has its own individual components. It is crucial that each of the components in the subsystems are working properly as in figure 7.1 i to ensure that overall system is working properly. If this is any individual hardware failure, it can prove to be catastrophic to the overall system. For this project, each of the individual hardware components will be tested inside a laboratory. In this lab, an oscilloscope, a multimeter, IEIK Uno and a computer will be used to conduct these tests. The oscilloscope along with the IEIK Uno and computer will be used to test both the pan and tilt servos. The IEIK Uno and computer will be used to test the SRF08 sensor and both digital triggers. The multimeter will be used to test the power supplies. The IEIK Uno and computer will be used to test the intel stick and the webcam. In the section below, there will be a more in depth description of these tests and their results.

The consumer environment will not differ much from the lab that the initial hardware testing was completed. The difference is that the consumer will be using the project in a larger, wide open area compared to a more confined space inside a lab. These differences will prove not to be a major concern since both are indoor environments and have similar conditions.

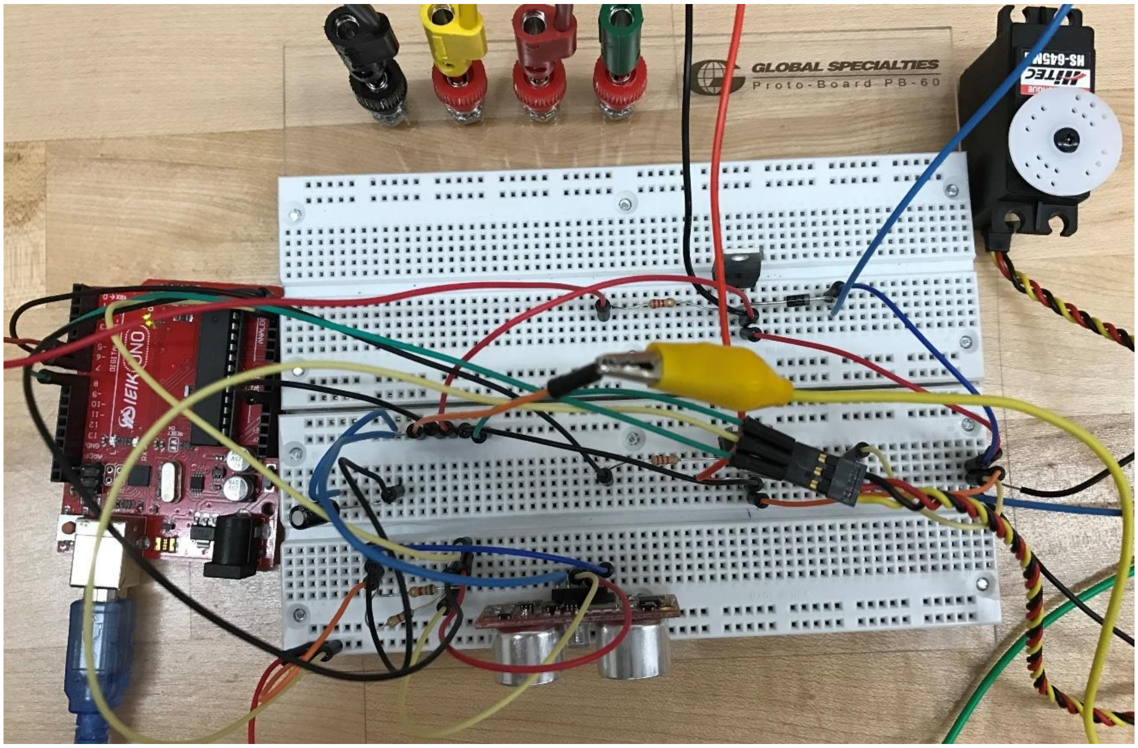
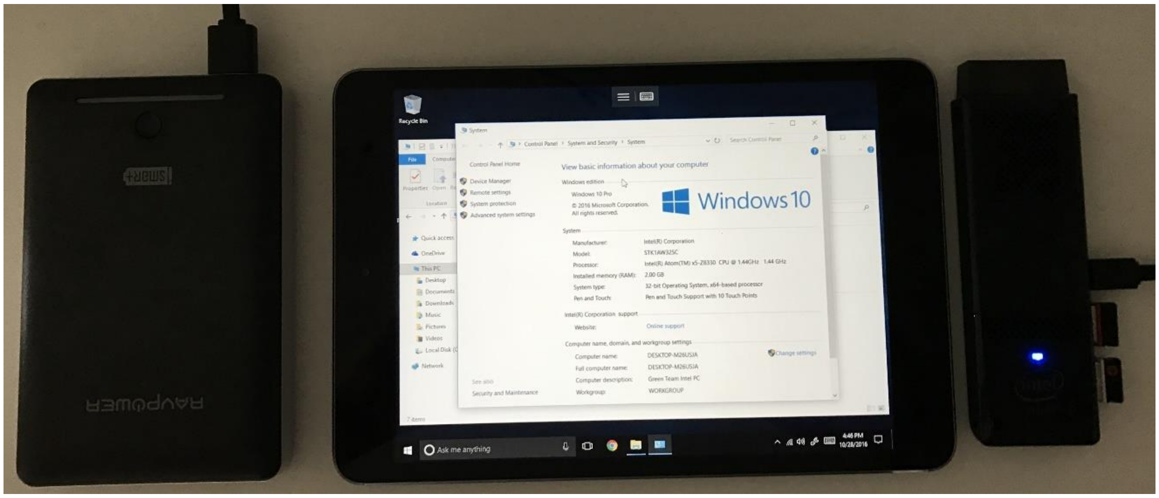


Figure 7.1 i: All Inclusive Breadboard Circuit Test

7.1.1 Computer System Test

Received information from the sensors will need to be processed on-board the remote system in real time and then viewed wirelessly from a mobile device. To test this functionality, the computer [which will be mounted on the robot] was powered on via battery pack and wirelessly logged into with a tablet. Figure 7.1.1 i proves this functionality. Range was also tested to ensure the devices could communicate when placed at distances of up to 30’ apart. The range test proves that a user will be able to view any processing information that occurs on-board the robot in real time.



View basic information about your computer

Windows edition

Windows 10 Pro

© 2016 Microsoft Corporation. All rights reserved.

System

Manufacturer: Intel(R) Corporation

Model: STK1AW32SC

Processor: Intel(R) Atom(TM) x5-Z8330 CPU @ 1.44GHz 1.44 GHz

Installed memory (RAM): 2.00 GB

System type: 32-bit Operating System, x64-based processor

Pen and Touch: Pen and Touch Support with 10 Touch Points

Intel(R) Corporation support

Website: Online support

Computer name, domain, and workgroup settings

Computer name: DESKTOP-M26USJA

Full computer name: DESKTOP-M26USJA

Computer description: Green Team Intel PC

Workgroup: WORKGROUP

Figure 7.1.1 i: Intel Stick powered by battery and Logged-in to via tablet

7.2 HARDWARE SPECIFIC TESTING

In this section, we check each component / piece of the project to ensure basic functionality and what we observe. For example, we tested the NERF gun and it fired but multiple buttons need to be pressed and minor abrasive damage to the foam bullets occurred. We also waveform tested the servo to observe the correct functionality of the servos and understand how they perform in the system.

7.2.1 Initial Device Under Test

Upon receipt of hardware it is important to test each individual component to ensure proper functionality. This way, if any faults are detected, said hardware components can be immediately exchanged or entirely swapped out for a more optimal counterpart.

7.2.1.1 IEIK Uno

Although a genuine Arduino Uno R3 would have been the most reliable choice of microcontroller test board, Group 2 / Green Team procured the IEIK Uno due to its extreme cost savings: over 50% less expensive than an original Arduino Uno R3. The functionality of the IEIK was proven as per the following test plan and the results are shown in table 7.2.1.1 i:

Initial Test Results		
Step	Test Description	Successful: Y/N
1	Plugged in / Powered On via USB to PC	Yes
3	Automatic Driver Installation	Yes
4	Serial Port Selection	Yes
5	Uploaded Blink LED Test Program	Yes
Conclusion		
The successful execution of the Blink LED test proved functionality of the microcontroller.		

Table 7.2.1.1 i: IEIK Uno Test Results

7.2.1.2 Intel STK1AW32SC (PC)

An initial test was conducted on the Intel PC to ensure functionality. The results are shown in table 7.2.1.2 i.

Initial Test Results		
Step	Test Description	Successful: Y/N
1	Powered ON vis wall adapter power source	Yes
3	Powered ON via battery pack	Yes
4	Confirmation of product specifications	Yes
5	Deleted bloatware	Yes
6	Installed development software	Yes
7	Upgrade Operation System	Yes
8	Longevity Test	Yes
Conclusion		
The computer successfully powered on and was fully functional using several power source methods. The computer was powered via battery pack for 5 hours to ensure proper functionality on a mobile system such as the robot it will be mounted on. Bloat software was removed to save storage space and development software was installed to support requirements. It was noted during testing that the device ran considerably hot but did not show signs or performance slowdown. Alternative cooling methods will be investigated.		

Table 7.2.1.2 i: Intel STK1AW32SC Initial Test Results

7.2.1.3 SRF08 Ultrasonic

The SRF08 Ultrasonic Sensor was purchased and tested to verify that the sensor is in working order as shown is Figure 7.2.1.3 i. The sensor itself has five pins that each need to have a right-angle pin headers soldered to each of the stand offs. After this was completed the sensor was then ready to be breadboard tested. The sensor was connected to a circuit on the breadboard that was supplied by the manufacturer of the sensor. This circuit was used strictly to conduct a simple test and may not reflect the final design. The sensor was then connected to the IEIK

Uno with sample code uploaded to it. The sample code would display the distance that the sensor is detecting in inches when an object crosses in front of the sights of the sensor. Multiple distances were tested to make sure the sensor is in working condition. However, the sensor had trouble at larger distances, but this could be due to the limitations of the sample code that was used. In the future, a more robust code will be written to ensure the sensor is working in the way intended for the project.

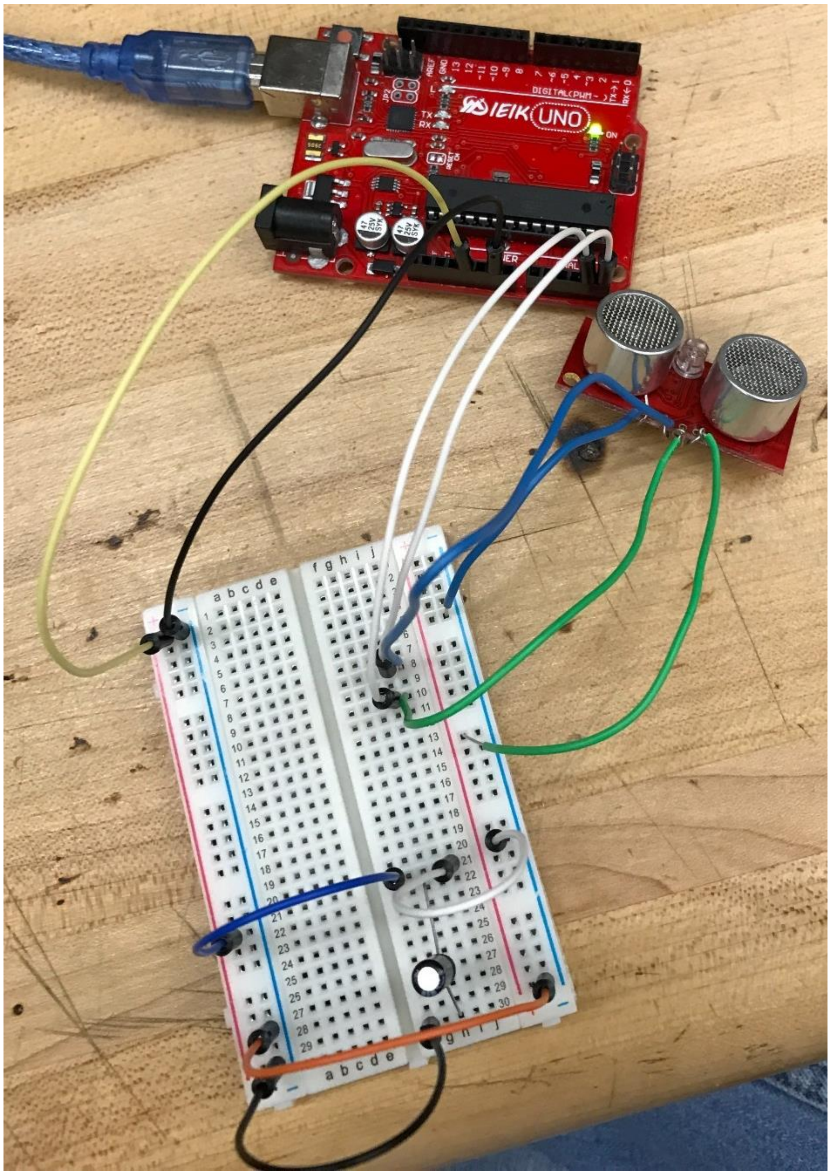


Figure 7.1.2.3 i: Ultrasonic Hardware Specific Test

7.2.1.4 NERF Gun

The NERF Gun was dismantled and tested by sending electric signals to activate the weapons trigger and fire several NERF projectiles. The weapon was tested to determine accuracy and capabilities of successfully hitting targets at long range.

Results of testing confirmed that NERF balls provide tight precision and high hit probability which will translate into more points during the competition.

The nerf gun was originally tested for it’s out of box state. The nerf gun requires six D batteries to power the weapon. The first test was to make sure that the battery pack was properly supplying the nine volts. To verify this, the group used the digital multimeter to read the voltage across the battery pack. With this test, it was shown that the battery pack was supplying above nine volts, which was expected. The next test was to make sure the weapon can fire properly. This weapon has a physical trigger that will be pulled when the user wants to shoot. When this trigger is pulled, it sends a signal to a motor that will begin to turn a conveyor belt. This conveyor belt is what will be feeding the next ammo to a part called the flywheels. This weapon has two flywheels that will spin to project the ammo out of the weapon. Along with this trigger, there is another trigger called the rev trigger. This rev trigger is how the two flywheels will be spun. To maximize the distance traveled, the rev trigger must be pulled (as in Figure 7.1.2.4 i) for at least one second to have the flywheels spin at the maximum rate.

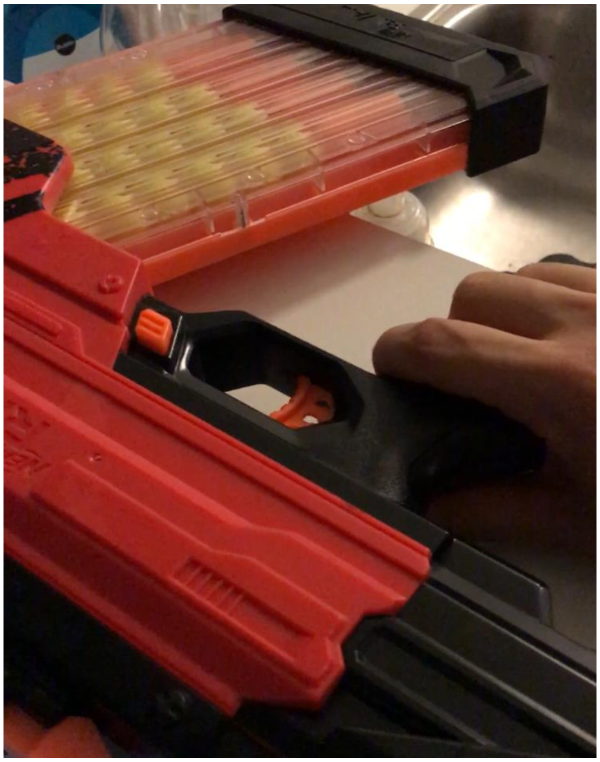


Figure 7.2.1.4 i: Revving the NERF Gun Flywheels

As the user has the rev trigger pulled in, they can begin to fire as they wish. The same process needs to be repeated once the user stops pressing the rev trigger. The group completed these steps and determined the weapon is in fully working condition. Table 7.2.1.4 i shows the results of these tests.

Initial Test Results		
Step	Test Description	Successful: Y/N
1	Voltage Read Across Battery Pack Is 9 Volts	Yes
2	Physical Trigger Feeds Ammo to Flywheels	Yes
3	Rev Trigger Spins the Two Flywheels	Yes
4	While Both Triggers Pulled, Gun Fires	Yes

Table 7.2.1.4 i: NERF Gun Test

After testing the general use of the nerf gun was conducted and determined to be working properly, tests were conducted to have the trigger and flywheels of the weapon be controlled by the IEIK Uno. For the weapon to fire, there is a separate trigger to spin the flywheels to propel the ammo out of the gun and a separate trigger to feed in more ammo. Initially, the idea was to secure down the flywheel switch to always be running and only worry about the firing trigger. In the end, it was determined to be better to have both triggers activated on command. To accomplish this, both triggers needed to be bypassed and two switching circuits needed to be designed to create digital triggers. The nerf gun was first disassembled to reveal the electronics of the weapon itself. The first step that was taken was mapping out what each component of the weapon is doing and what the wires are connected to. After this was completed the trigger was the first part to be taken apart. The goal was to be able to bypass the physical trigger on the weapon itself to have it fully controlled by the IEIK Uno. To accomplish this the correct wires needed to be found and cut to break the connection between the motor of the conveyor belt and the physical button. The positive and negative terminals of the motor were found and the leads to them as well. To bypass the physical trigger, the wires of the positive and negative terminals were cut to break this connection. This will ensure that the physical trigger no longer controls when the motor will be running to feed the ammo to the flywheels. Instead of the physical trigger, a simple switching circuit was constructed on a breadboard as seen in figure 7.2.1.4 ii.

The switching circuit consisted of a resistor, diode and BJT transistor. The leads of the motor were connected to this switching circuit to become the new digital switch. When no voltage is supplied to this circuit, the switch would act as if it were opened, not completing the circuit. When voltage is applied to the circuit, the switch will be closed, completing the circuit. In return, when the switch is opened, the motor will not be running and when the switch is closed, the motor will begin to run.

To verify that the switching circuit was working properly, pre-built software called Project Sentry Gun was used. This program uses pin 7 on the IEIK Uno as the digital trigger. The program has a manual fire mode that works with any webcam. When you click on the video window of the webcam, the weapon will begin to fire, sending a HIGH signal to the pin. When you release the mouse the weapon will cease fire, sending a LOW signal to the pin. With the circuit constructed the program was ran to verify that the digital trigger is working properly. The test was successful and the motor ran while the mouse was clicked and did not run when the mouse was not clicked. Initially, the same plan was used, but instead of cutting the wires between the negative and positive terminals of the motor, a wire was conducted to each terminal to the switching circuit. When this plan was used, the motor would always be running the conveyor belt, no matter what the status of pin 7 was set to. This was since the terminals were already wired to the physical trigger, always having a complete connection. To fix this issue, it was determined that those connections needed to cut to disrupt the signal. After this issue was fixed, the same process was used to create a digital switch to control the flywheels.

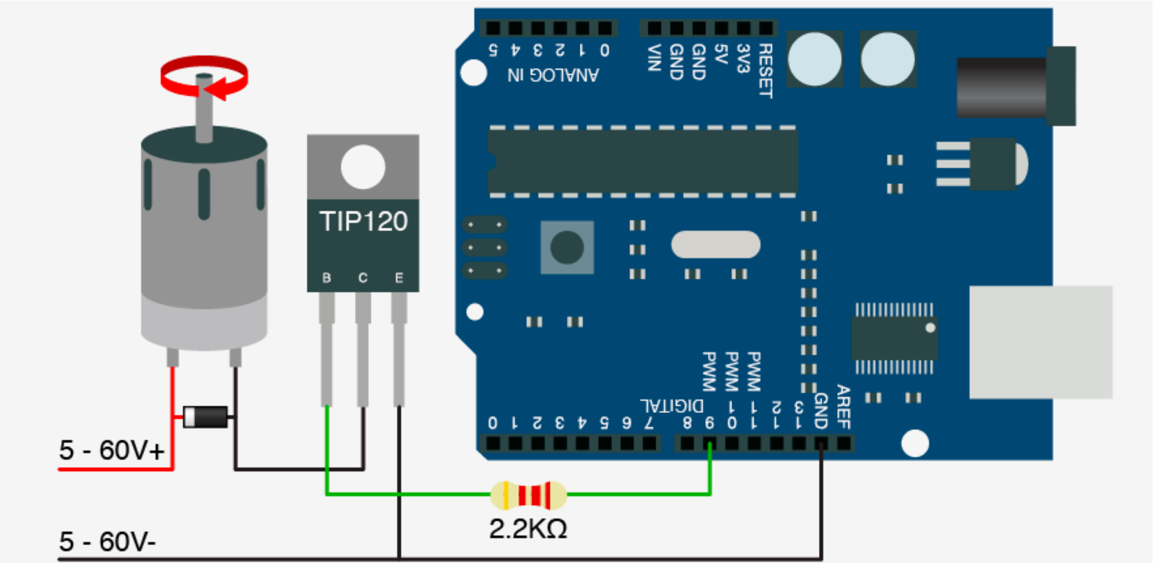


Figure 7.2.1.4 ii: Simple Switching Circuit

The flywheels on the weapon work in a similar way to that of the conveyor belt and the fire trigger. The flywheels are connected to its own trigger independent of the physical fire trigger. The result was to have this physical flywheel trigger be replaced with a digital trigger controlled by the IEIK Uno. The same process above was used to create this digital switch. During the first test of this digital switch, the group noticed that the flywheels were noticeably turning slower than the original test of the weapon. To troubleshoot this issue, a digital multimeter was used to measure the voltage across the flywheels connected to the weapon in the original out of the box state. With this test the voltage was above six volts. However, with the switching circuit, the voltage was about five volts, which was significantly lower. During this test a noticeable smell was coming from the switching circuit. It was found that this smell was coming from one of the wires of the circuit, due to the wire becoming very hot. It was deemed that this wire was not in complete working order and the wire was then replaced. After the wire was replaced the test was conducted again. During this test, the voltage across the flywheels were back around six volts. The digital switch for the flywheels were now working as intended. The final test of the weapon was having both digital switches connected at the same time to the IEIK Uno. Figure 7.2.1.4 iii shows a lab test of a single digital switch circuit to control the conveyor belt of the NERF gun.

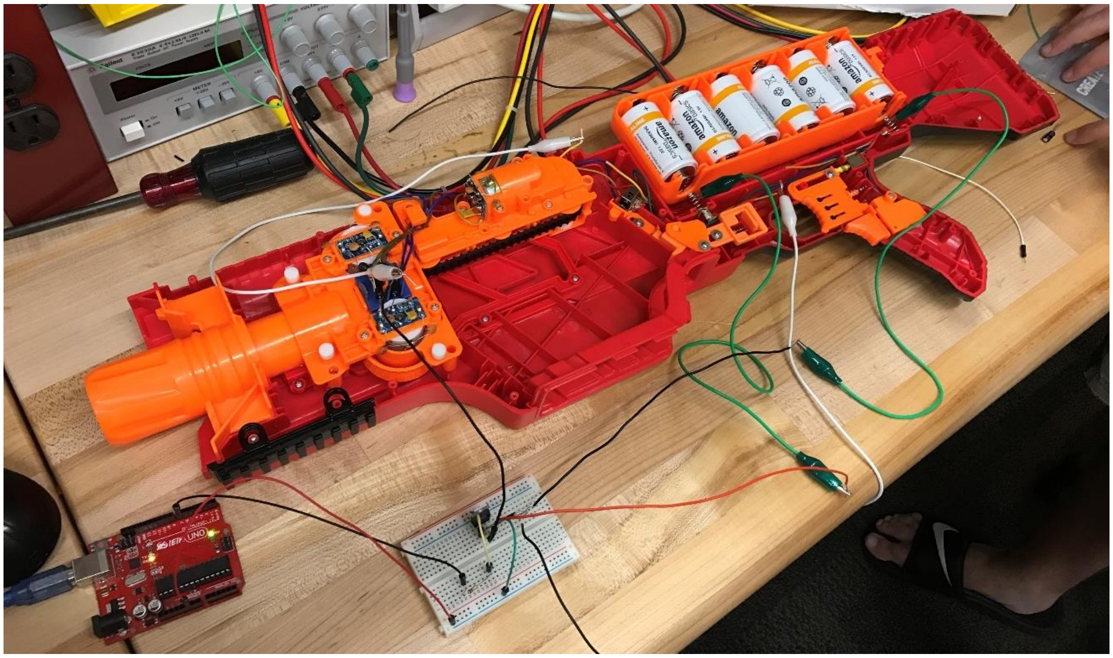


Figure 7.2.1.4 iii: Testing the NERF Gun

The same program was then conducted to determine if when the mouse is clicked, that both the flywheels and conveyor belt were working. To accomplish this, the sample program needed to be modified to accommodate for both the digital switches. Originally the program is designed to control one digital switch connected to pin 7 on the IEIK Uno. However, for the project’s purpose, two digital switches are needed. The flywheels were connected to pin 6 and the program was then modified to include both switches. When the mouse was clicked, both pin 6 and 7 were set to HIGH, turning the flywheels and the conveyor belt. When the mouse was not clicked, pin 6 and 7 were set to LOW, stopping the flywheels and conveyor belt. It was deemed that both switches were working as expected. In the future, a delay will be implemented for the flywheels to give them more time to create more rotation. Doing this will increase the fire speed of the ammo and ensure that the ammo would travel the greatest distance. Table 7.2.1.4 ii shows the summary results of the above tests.

Initial Test Results		
Step	Test Description	Successful: Y/N
1	Power provided to the trigger/flywheel	Yes
2	Trigger Fires When Pin 7 is HIGH	Yes
3	Trigger Stops Fire When Pin 7 is LOW	Yes
4	Flywheels Turn When Pin 6 is HIGH	Yes
5	Flywheels Do Not Turn When Pin 6 is LOW	Yes
6	Trigger and Flywheel Both Work Together When Pin 6/7 Are HIGH	Yes
7	Trigger and Flywheel Both Off When Pin 6/7 Are LOW	Yes

Table 7.2.1.4 ii: NERF Gun Test

7.2.1.5 Battery Packs

Both the 9V battery pack containing one 9V battery and the 6V battery pack containing four AA batteries were tested. To test to make sure both battery packs were working properly, the voltage was read using the digital multimeter. On each of the battery packs, there is an ON/OFF switch which was also tested. Table 7.2.1.5 i shows the results from these tests.

	9V Battery Pack	**6V Battery Pack**
Expected Voltage with Switch ON	> 9 Volts	> 6 Volts
Expected Voltage with Switch OFF	0 Volts	0 Volts
Actual Voltage with Switch ON	9.74493 Volts	6.15384 Volts
Actual Voltage with Switch OFF	~0 Volts	~0 Volts

Table 7.2.1.5 i: Battery Pack Test

Figure 7.2.1.5.1 i shows the result of the digital multimeter reading. The expected result was for it to read at least 6 volts. While the 6V battery pack switch is in the OFF position, the expected result was 0 volts. The expected values were then verified and in conclusion, the 6V battery pack was functioning properly.

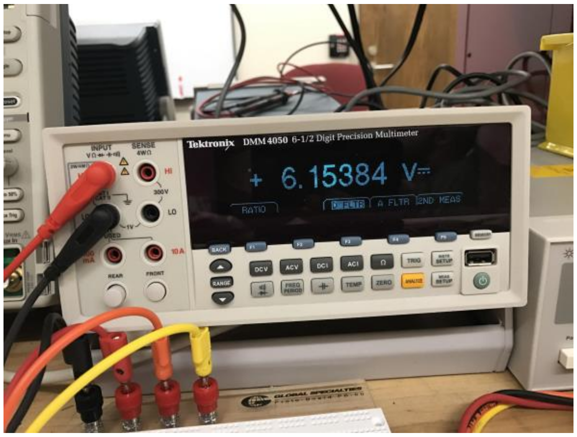


Figure 7.2.1.5 i: 6V Battery Pack with Switch ON

Figure 7.2.1.5.1 ii shows the result of the digital multimeter reading. The expected result was for it to read at least 9 volts. While the 9V battery pack switch is in the OFF position, the expected result was 0 volts. The expected values were then verified and in conclusion, the 9V battery pack was functioning properly.

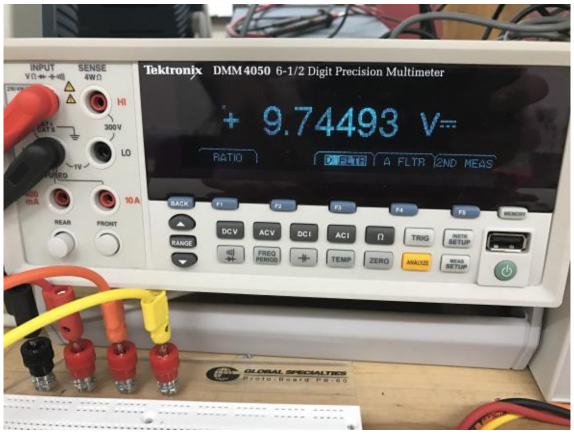


Figure 7.2.1.5 ii: 9V Battery Pack with Switch ON

7.2.1.6 Rechargeable Lithium Ion Power Supply

The rechargeable lithium ion power supply was tested to ensure that it was outputting the proper voltage. This power supply will be used to power the intel stick which requires at least 5 volts. To test to make sure the power supply is working properly, the power supply was connected to the intel stick. After, the power supply was turned on to make sure that the intel stick was powered. Table 7.2.1.6 i shows the results from this test.

Expected Voltage	> 5 Volts
Expected Result	Intel Stick ON
Actual Result	Intel Stick ON

Table 7.2.1.6 i: Power Supply Test

After the test was conducted is was concluded that the power supply is working properly.

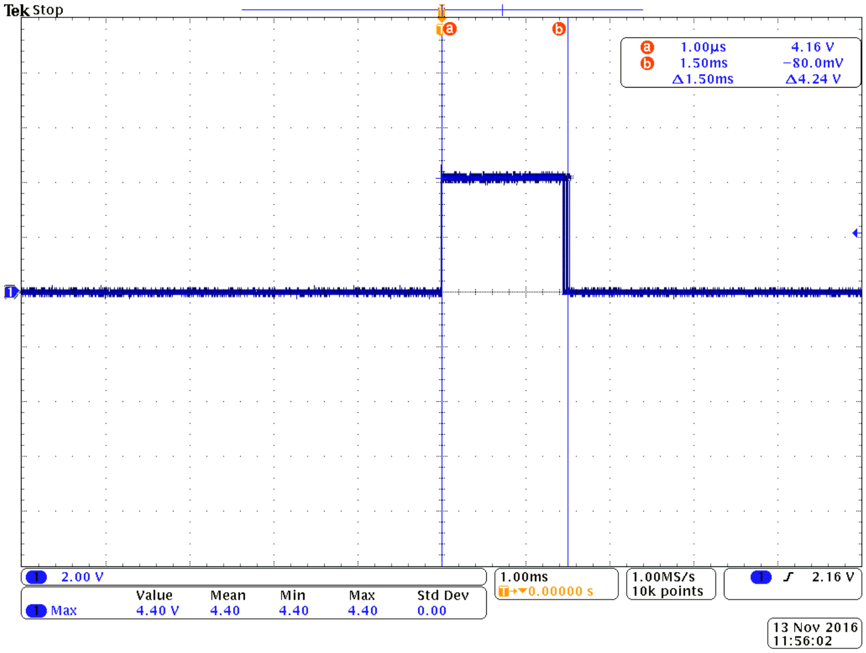
7.2.1.7 Servos

Both servos were connected to the Arduino IEIK UNO, and a sample program was conducted to test the functionality of the servos. The servos were then connected to the oscilloscope to produce the waveforms. A pulse width waveform is expected from this test. Table 7.2.1.7 i shows the results from this test.

	Hitec HS-645MG		**Hitec HS-5645MG**	
Expected Time Delay	1.5ms	2ms	1.5ms	2ms
Expected Turn Direction	90° CW	180° CW	90° CW	180° CW
Actual Time Delay	1.5ms	2ms	1.5ms	2ms
Actual Turn Direction	90° CW	180° CW	90° CW	180° CW

Table 7.2.1.7 i: Servo Test

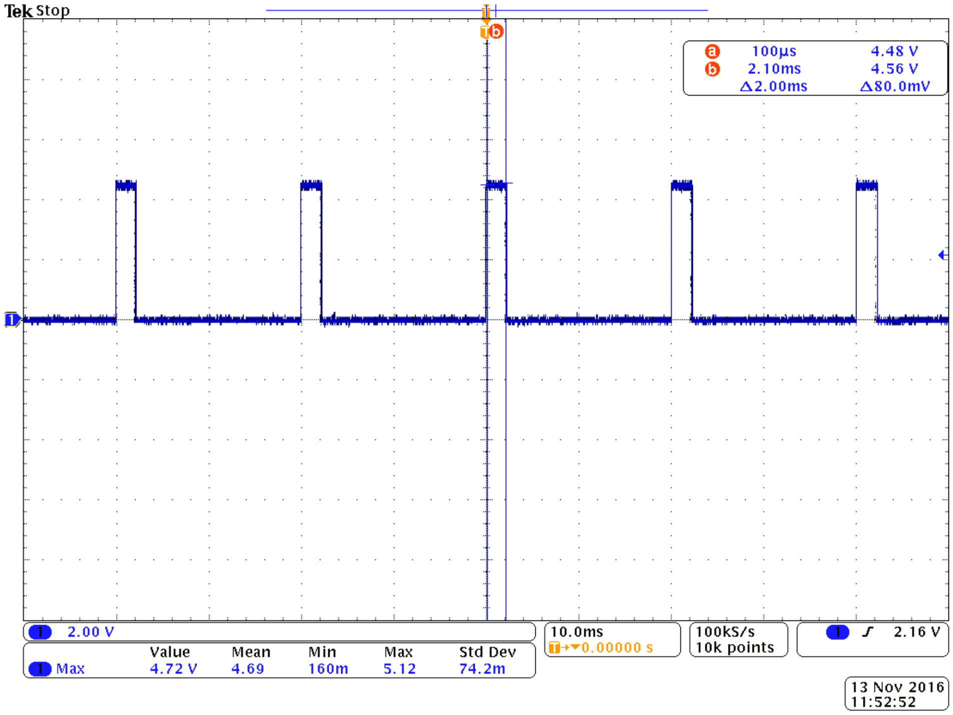
Figure 7.2.1.7 i shows the pulse width waveform with a time delay of 1.5ms of the Hitech HS-645MG servo. It was expected when the servo turned 90° CW that a time delay of 1.5ms would occur. The waveform confirms this expected result and confirms that the servo is working properly.



	Value	Mean	Min	Max	Std Dev
1 Max	4.40 V	4.40	4.40	4.40	0.00

Figure 7.2.1.7 i: Hitec HS-645MG Servo 90° CW

Figure 7.2.1.7 ii shows the pulse width waveform with a time delay of 2ms of the Hitech HS-645MG servo. It was expected when the servo turned 180° CW that a time delay of 2ms would occur. The waveform confirms this expected result and confirms that the servo is working properly.



	Value	Mean	Min	Max	Std Dev
1 Max	4.72 V	4.69	160m	5.12	74.2m

Figure 7.2.1.7 ii: Hitec HS-645MG Servo 180° CW

After these tests were completed it was concluded that both the servos are functioning properly.

7.2.1.8 LIDAR Sensor

The LIDAR-Lite v3 Sensor was purchased and tested to verify that the sensor is in working order as shown in Figure 7.2.1.8 i. The sensor has four connections that will be connected to the IEIK Uno. A 680 uF capacitor was added between power and ground connections to provide stability. Sample code was loaded onto the IEIK Uno to test the functionality of the sensor. It outputted the distance the sensor was detecting. These values were converted to feet to better understand if we were receiving proper values. It was determined that the values were correct, and the testing was concluded.

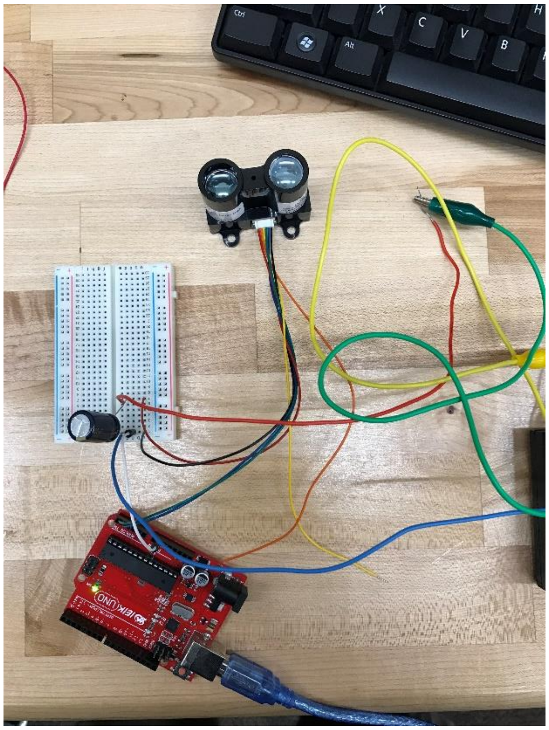


Figure 7.2.1.8 i: Lidar Hardware Specific Test

7.3 SOFTWARE TEST ENVIRONMENT

The target detection software was tested in the lab. For demonstration purposes this closely simulates the baseline requirement from the UCF faculty. The webcam software was proved to be compatible to the intel stick. This is important because the intel stick will be running the java code.

A few different types of software will need to be tested to ensure everything is working properly. The pieces of software that are being used are Arduino Software (IDE) and Processing. To test these pieces of software a Windows computer running Windows 10 was used. Both programs ran efficiently on the computer.

7.4 SOFTWARE SPECIFIC TESTING

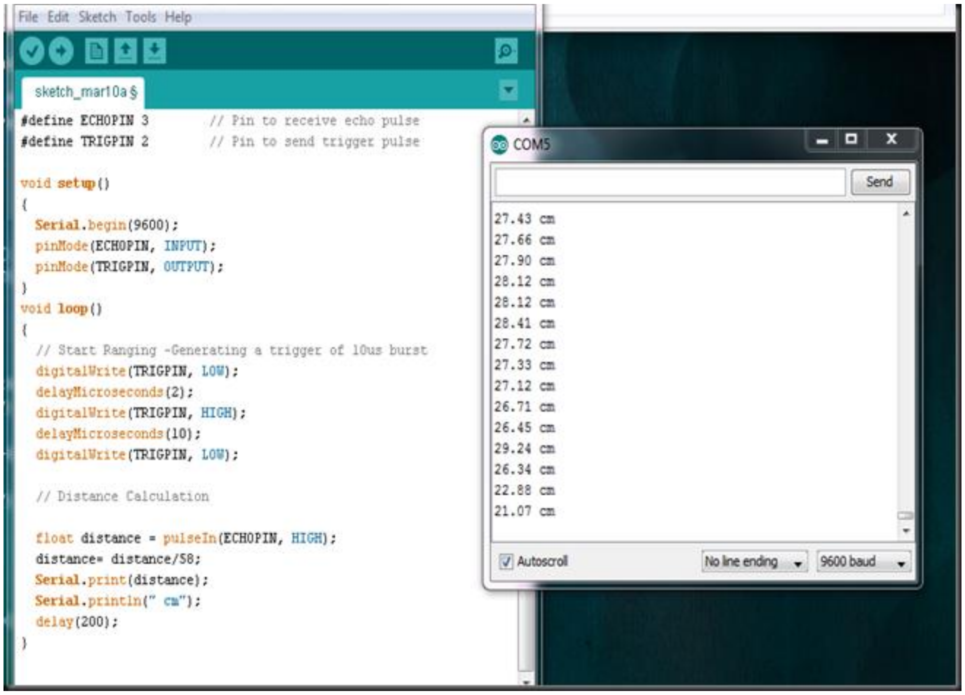
This section will describe how each piece of software was tested to ensure proper functionality. There are a few ways to test each one, and that is what will be discussed.

7.4.1 Arduino Software (IDE)

This section will describe the different areas that must be tested using the Arduino Software to make sure that it is functioning properly.

7.4.1.1 Ultrasonic Sensor Test

To make sure the software will work with the ultrasonic sensor, test code is created and ran using the Arduino Software. Figure 7.4.1.1 i shows an output example of what is to be expected. Since the values that are expected are displayed in the software it is determined that the software is working properly.



```
#define ECHOPIN 3        // Pin to receive echo pulse
#define TRIGPIN 2        // Pin to send trigger pulse

void setup()
{
  Serial.begin(9600);
  pinMode(ECHOPIN, INPUT);
  pinMode(TRIGPIN, OUTPUT);
}
void loop()
{
  // Start Ranging -Generating a trigger of 10us burst
  digitalWrite(TRIGPIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIGPIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGPIN, LOW);

  // Distance Calculation

  float distance = pulseIn(ECHOPIN, HIGH);
  distance= distance/58;
  Serial.print(distance);
  Serial.println(" cm");
  delay(200);
}
```

```
27.43 cm
27.66 cm
27.90 cm
28.12 cm
28.12 cm
28.41 cm
27.72 cm
27.33 cm
27.12 cm
26.71 cm
26.45 cm
29.24 cm
26.34 cm
22.88 cm
21.07 cm
```

Figure 7.4.1.1 i: An Example of an Ultrasonic Sensor Test

7.4.1.2 Pan and Tilt Servo Test

The pan and tilt sensors need to be able to work using the Arduino Software. The software must be able to control each of the servo motors. To do this, a test piece of software must be written. This software will tell the servos to rotate clockwise and then counterclockwise. Each of the servos will need to be tested this way. If they operate as expected, then Group 2 / Green Team will know that the software is functioning properly with the servos. Both servos operated as expected, so the software was working.

7.4.1.3 Trigger and Flywheel Motor Test

The next step to insure the software is function properly with the trigger and flywheel motors. A piece of software was written that told the predetermined pins of each of the motors to go high at a certain point and low at another. If the motors turn on and off, then Group 2 / Green Team that the software is functioning the way it should be. Upon doing this, both trigger and flywheel motors turned on and off.

7.4.1.4 OpenCV Library Test

A library that is to be tested with the Arduino Software is OpenCV. This library has different object detection methods built into it. One such detection method is background subtraction. This part of the library was accessed and test code was written. Figure 7.4.1.4 i shows the result of using this object detection method. There is a hand in the right side of the screen with the rest of it black because the hand was not originally there. This shows that the OpenCV library along with the Arduino Software is functioning.

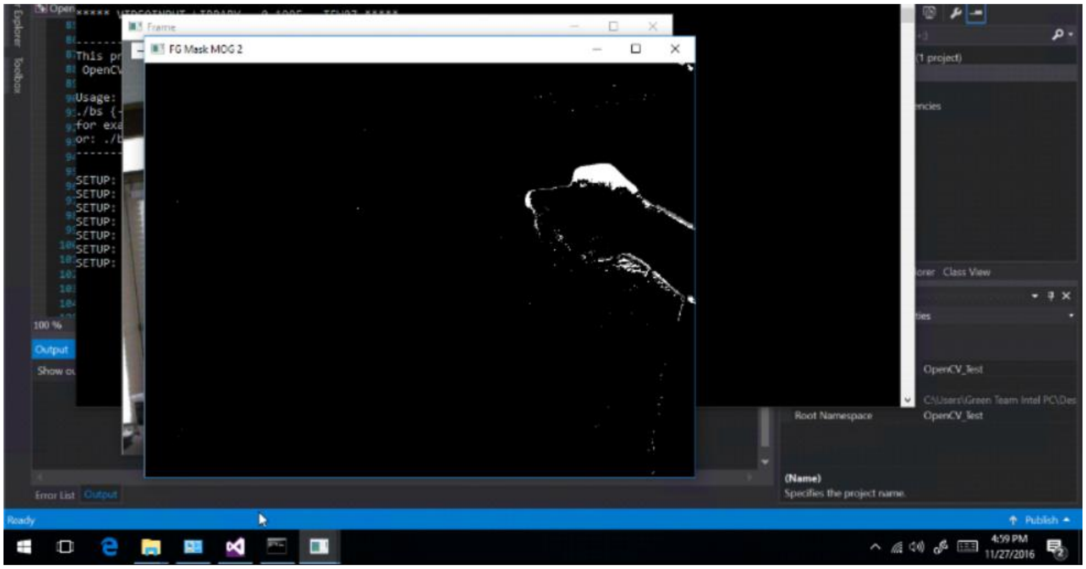


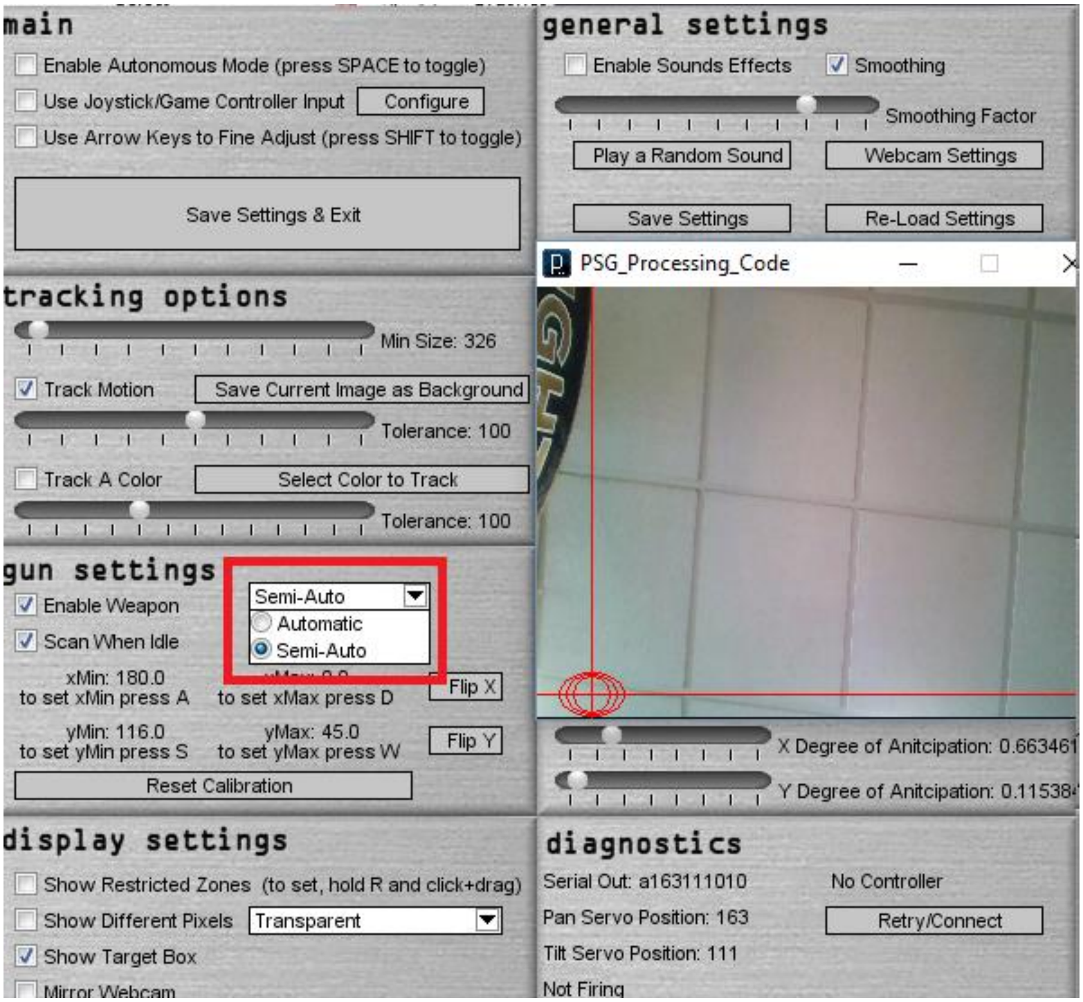
Figure 7.4.1.4 i: Testing the OpenCV Library

7.4.2 ATR Processing

The next piece of software that must be tested is the processing code. Using java code in the processing IDE this software detects objects and will "fire" depending on what mode is set. There are a few different modes that will be tested, and those will be discussed in the sections ahead.

7.4.2.1 Semi-Automatic Mode

The first mode that will be tested is semi-automatic mode. In this mode, it is expected that when the mouse enters the webcam window the software will "fire" with a delay in-between each instance of "firing". This mode was set and the software indicated that the gun was "fired". Figure 7.4.2.1 i shows the testing of this mode of fire.



main

☐ Enable Autonomous Mode (press SPACE to toggle)

☐ Use Joystick/Game Controller Input

☐ Use Arrow Keys to Fine Adjust (press SHIFT to toggle)

tracking options

Min Size: 326

☑ Track Motion

Tolerance: 100

☐ Track A Color

Tolerance: 100

gun settings

☑ Enable Weapon

☑ Scan When Idle

Semi-Auto

○ Automatic

● Semi-Auto

xMin: 180.0
to set xMin press A

to set xMax press D

yMin: 116.0
to set yMin press S

yMax: 45.0
to set yMax press W

display settings

☐ Show Restricted Zones (to set, hold R and click+drag)

☐ Show Different Pixels Transparent

☑ Show Target Box

☐ Mirror Webcam

general settings

☐ Enable Sounds Effects ☑ Smoothing

Smoothing Factor

PSG_Processing_Code

X Degree of Anitcipation: 0.663461

Y Degree of Anitcipation: 0.11538

diagnostics

Serial Out: a163111010

No Controller

Pan Servo Position: 163

Tilt Servo Position: 111

Not Firing

Figure 7.4.2.1 i: Testing the Semi-Auto Option in Processing

7.4.2.2 Automatic Mode

The next part of the software to test is the automatic mode. This mode functions almost the same as the semi-automatic mode, but instead of a delay in-between each “fire” it will “fire” consistently until the mouse leaves the webcam window. When this test was done, the software indicated that this was the case.

7.4.2.3 Manual Mode

This part of the software will be tested by moving the mouse in the webcam view window. Once the mouse is clicked the software will “fire” the gun. When this was tested, the software indicated the gun “fired” when the mouse was clicked.

7.4.2.4 Color Tracking

The last piece of the software to test is its color tracking ability. To test this a color was selected to track. In this case, it was chosen to be red. A red notebook was then placed in front of the webcam. The software picked up this red notebook and put a crosshair on it. The software then followed the notebook as it was moved around. This ensured that this portion of the software is working.

7.5 Spring 2017 Software Debugging and Finalization

During the Spring 2017 semester, Group 02 finalized the basic targeting software and integrated the LIDAR hardware into the Arduino software for the computer science team.

7.5.1 LIDAR Handshaking

Upon acquisition on the LIDAR sensor, it was quickly testing and verified for functionality. The LIDAR could accurately detect range readings much farther away than the ultrasonic sensor. The code for the LIDAR sensor was incorporated into the Arduino file but caused parts of the functionality to break. Namely: the servo turret movements would stop working as soon as the LIDAR was activated. The root cause of this problem was found to be that the Arduino and intel stick needed to have handshaking implemented for the serial data traveling over the USB to be properly organized. Once this was taken care of, the turret movements and LIDAR could both be handled simultaneously by the system.

7.5.2 Targeting Calibration

The computer vision software could detect and track color – a main requirement. Upon final construction of the robot the camera was mounted far below the muzzle of the NERF weapon. This caused the targeted to be off by a great margin. The Arduino targeting software was revised to accommodate for this. The results show greater than 70% of NERF rounds hitting the 16” x 16” target at a 15’ range. This was demonstrated live for a committee at UCF on April 17th, 2017.

8.0 ADMINISTRATIVE

8.1 PROJECT PLAN & MILESTONE DISCUSSION

Within only a short period, the group 2 electrical and computer engineering (ECE) team must go from bare concept to a fully matured product. Two semesters do not offer much room for error so the construction and administration of a thoroughly detailed master plan will allow for significantly greater efficiency in goal accomplishment. A detailed look follows in this section of how group 2 achieves major tactical advantages in terms of delivering a successful product in a timely manner to the customer.

Within the first week, group 2 forms through online UCF communications and miscellaneous project ideas are submitted to investigate various project possibilities. Electrical and Computer engineering offer many areas of exploration, so it is important to narrow down and confirm a solid, exciting project that yields an expressive learning experience while providing a solid resume-boosting platform.

By September 9th Group 2 decides to develop the Battlebot autonomous target recognition system. That same day, a several page introductory draft proposal is submitted for review by Dr. Lei Wei. Within this “Initial Project Documentation,” the overall project is introduced, objectives are abstractly covered, and additional details such as specifications, important dates, and a draft house of quality are included.

On 9/20 group 2 meets with Dr. Wei for a professional critique and advisement on the outlook of the project. Over the remainder of the semester, Group 2 will check in with Dr. Wei to make sure the final report contains all required material and adheres to all necessary formatting specifications. Now the project is officially approved for group 2 to fully commence working on the project. Group 2 won’t receive an official debriefing on the specifics of the project by Lockheed Martin sponsors/representatives Kenny Chen and Jonathan Tucker until September 21. Initially consisting solely of electrical and computer engineers prior to the said-meeting, group 2 merges with computer science and mechanical engineering to form the greater: Green Team – one of the three autonomous robot factions to face off in the robot evaluation and competition in Spring 2017. Group 2 will continue to participate in bi-weekly telecons with Lockheed Martin to keep on schedule and to provide and receive important status updates. To gain additional clarity and focus, most Green Team attends Senior Design Boot Camp on 9/24 – an informational retreat for all UCF undergraduate engineers currently enrolled in senior design I.

Following the September 20th & 21st meetings, group 2 refines the first several sections of the final document and submits a revised 10-page report on 9/30. By this time thorough research and development by group 2 is well under way and the first weekly status meeting is held to kick off collaboration between the multiple disciplines of green team. From this initial meeting, the electrical, mechanical, and computer science sub-groups will understand their role in the overall design of the autonomous robot. The second Green Team status meeting will be joined by professional technical advisor: Ray Gardner.

Fall 2016 Timeline (Senior Design I)		
<u>Description</u>	**<u>Duration</u>**	**<u>Dates</u>**
ECE Group Formation & Project Idea Inception	1 Week	8/22 – 8/26
Discussion / Divide and Conquer	2 Weeks	8/26 – 9/9
Initial Project Documentation	1 Week	9/2 – 9/9
Research Parts/Past Projects/Similar Products	2 Weeks	9/9 – 10/26
Define & Conquer Half Hour Meeting	-	9/20
LM Sponsor and Robot Group Introduction	1 day	9/21
Senior Design Bootcamp	1 day	9/24
Revise Initial Project Document	1 Week	9/23 – 9/30
First Green Team Status Meeting & LM Telecon	-	9/30
Project Documentation: Draft Development	1.5 Weeks	9/30 – 11/11
Ray Gardner Initial Consultation	-	10/14
Initial Submission of Parts Order to J. Fackler	-	10/23
Initial Parts Pick-Up, begin initial tests	-	10/28
Table of Contents	-	11/4
Prototyping	2 Weeks	11/4 – 11/18
Finalize Project Documentation	2.5 Weeks	11/11 – 12/6
Draft Review Meeting with Dr. Wei	-	11/15
Lockheed Martin Preliminary Design Review	-	12/13

Table 8.1 i: Fall 2016 Milestone Timeline

Green team will discuss their current plans with Ray Gardner and in turn, will be consulted with 30 years of expert defense and sensor knowledge. Intermittent communications with Ray will continue throughout the remainder of the project; hopefully this will allow for the prevention of any catastrophic mishaps. In addition to these communications with Ray Gardner, Green Team will continue to meet on at least a weekly basis to maintain a steady pace to complete all necessary milestones on schedule.

Increasingly acute design focusing will occur during the months of October and November. Parts will be selected and procured after rigorous research is conducted and approval is received by each sub-group (ECE, ME, & CS). On November 4th, group 2 submits the required “Table of Contents,” outlining the final document in detail. Shortly thereafter, a major rough draft consisting of approximately 60 pages is uploaded for review on November 11th. With an ideal count of at least 60 pages complete, the remainder of the final document is constructed while simultaneous prototyping of critical components occurs for the rest of the month of November. On December 6th, group 2 turns in the final 120+ page document to professor Lei Wei detailing all the objectives, research, design, testing, and administrative aspects of the project.

After the conclusion of Fall 2016, bi-weekly telecommunications with Lockheed Martin will continue during the month of December and well into Spring 2017 until the successful delivery of the autonomously targeting Battlebot. Prior to delivery, the Battlebot will of course undergo intense testing and tuning to avoid every conceivable flaw in design.

Please refer to table 8.1 i for a summarized list of critical milestone and their respective duration and date(s) of occurrence for Fall 2016. Please refer to table 8.1 ii for a summarized list of critical milestones and their respective duration with some tentative dates for Spring 2017.

Spring 2017 Timeline (Senior Design II)		
Description	**Duration**	**Finish Date**
Build Prototype	8 Weeks	March 17th
Lockheed Martin Demo Day	2-4 Weeks	March 17th
Finalize Project & Readiness Review	1 Weeks	April 12th
Lockheed Martin NERF Battlebot Competition	-	April 14th
Final Committee Presentation	1 Week	April 17th

Table 8.1 ii: Spring 2017 Timeline

8.2 BUDGET & FINANCE

As the official sponsor of the robot project, Lockheed Martin has financed each robot team (red, blue, and green) with $2000 that is available for acquisition through the University of Central Florida. The total cost of developing and delivering the all-up robot absolutely cannot exceed the $2000 financed benchmark. Lockheed Martin has also specifically required that the as-demonstrated all-up robot must cost no more than $1000.

Being an interdisciplinary project, the budget is shared between Mechanical Engineering, Electrical and Computer Engineering, and Computer Science. Roughly speaking, the mechanical design of the robot is expected to occupy much of the cost with electrical components arriving second in expense magnitude. Software development itself will only cost man hours and not physical dollars. This is largely in part because 1) the team is not charging an hourly rate to write software and 2) the interactive development environments (IDEs) required to construct the autonomous software, manual drive control software, and user-interface software are entirely void of monetary requirements. Other factors contributing to software affordability include the vast amount of open-source code available that can be studied by Green Team. To summarize: software development will not significantly impact budget if at all; only the mechanical/electrical design and implementation will.

Since this is a budgetary analysis for Group 2 Electrical and Computer Engineering and not the cumulative inspection for Green Team, only the cost impact of such relevant components and resources will be discussed in detail. Group 2 is responsible for procuring and supporting the implementation of the sensors into the robot which will provide the iris for all environmental data collected and processed by the robot. Without the sensors, object detection and tracking would be virtually impossible, rendering the entire robot void of any use to the customer. Very clear requirements and specifications are given by Lockheed Martin which paint an obvious picture of what capabilities will be called upon from the sensors. Though not industrial standards by far, the sensors will need to support target detection on a course that is 20 feet by 40 feet – a significant area. High quality sensors will need to be purchased that can meet these standards. The sensors themselves are not expected to be rarely available nor is group 2 concerned about the reliability of said sensors. Therefore, spare sensor procurement will not need to be factored into the budget. Group 2 will spend time to carefully select high performing and budget-friendly sensors.

Given the diverse requirements of the robot, group 2 will purchase separate power supply’s to individually support the drive system and targeting system. It is important these power sources be separate to ensure that the sub-systems do not interfere with each other. The power supplies will need to be reliable and efficient enough to last for the duration of the two 10-minute rounds but not so overbearing that excessive weight is added to the robot. A heavier robot would require even more energy to travel across the course and would be significantly restricted in terms of mobility.

Lockheed Martin has made very strict requirements regarding the type of weapon that is to receive integration. It must be NERF branded and not modified in any way that changes the velocity or distance of the projectiles it fires. Given that no third-party weapons may be considered, the NERF weapon will account for a large portion of the budget. Group 2 has researched many options for NERF weapons and the ideal choice is one with an electric trigger that fires “NERF balls” (small spherical projectiles). The NERF Rival series weapon fires NERF balls at extremely high velocities with accurate trajectories and its electric trigger provides an ideal gateway for assimilation into the targeting system.

As Built Robot ≤ $1000			
Item #	**Nomenclature**	**Price**	**Comments**
1	Sensor Modalities	$80	Required
2	Power Supply(s)	$100	To power the subsystems
3	NERF Weapon(s)	$100	Required
5	Custom Printed Circuit Board (PCB) with Microcontroller	$30	3 will be purchased as a safety precaution
6	Computer	$100	To support ATR processing
7	Motor Driver Card	$20	To support robot drive control
8	Custom Pan / Tilt Assembly	$100	For mounting and control of the NERF weapon
9	Software	$0	Free for students
10	Chassis	$200	To support the ATR system. Built by the mechanical engineering team.
	Total ≤ $1000	~$700	

Table 8.2 i: Estimated Group 2 Budget Breakdown of All-up Robot

Spare NERF ammunition will need to be purchased as well since Green Team will need to compete in several rounds during the demonstration. To provide the highest chance of success possible, Green Team will incorporate a duel-weapon system. Adjacent to the NERF ball projectile launcher will be a secondary mounted NERF dart projectile launcher. Although it is costlier to incorporate two weapons, Lockheed Martin has specified that this is the only way to achieve the maximum allowable ammunition load-out. Having two NERF weapons will allow broader distribution of capabilities allowing one to handle long range targets and the other to handle close range targets.

For the weapon to precisely target enemy objects a pan/tilt assembly will need to be either purchased or constructed by the mechanical team. Affordable heavy duty tilt assemblies that can reliably wield two NERF weapons do not generally come with built-in servos therefore, compatible servos will be purchased as well. The tilt assembly and servos are expected to last throughout prototyping and testing and as such will not need spare replacements on the ready.

Additional Expenses ≤ $1000			
Item #	**Nomenclature**	**Price**	**Comments**
1	Spare Printed Circuit Board	$60	In case of short-circuit
2	Spare Power Supply(s)	$50	Batteries etc.
3	Spare NERF Ammunition	$20	Will need additional for tests
4	Test components	$100	For breadboard testing
5	Miscellaneous components i.e. connectors, cables, etc.	$30	Any other small parts
Total ≤ $1000		~$250	

Table 8.2 ii: Estimated Group 2 Budget Breakdown of Spare Components

A computer will need to be purchased as well to quickly process the autonomous targeting algorithm. A premium computer purchase is imperative to the success of green teams' robot because it needs to be able to smoothly handle the heavy processing loads of complex image decoding and video signal transferring. The computer will provide object oriented capability to the video-fed user interface which will functionally be made possible through the acquisition of a wireless connection. Separate from the targeting firmware, a motor driver card will need to be included in the budget to ensure precision control of the manual drive system. Speed, direction, and overall snappy maneuvering capabilities will come via the motor driver card. All these computational components will interact with the weapon hardware and power sources through means of a custom designed printed circuit board (PCB) and micro-controller. Given the fact that it will be group 2's first time designing and ordering a printed circuit board, there is a high risk for error. Even though Green Team will unite to ensure maximum reliability of the PCB, many possibilities put the team at a high risk for failure if certain fail-safe measures are not put in place. It is therefore reasonable that Group 2 orders at least 3 printed circuit boards just in case the manufacturer delivers a defective board or prototyping causes irreparable damage to a PCB etc. This decision will significantly harm the Green Team's cumulative budget but it is a much safer option in the long run.

If Green Team is ahead of schedule and still has a large portion of remaining funds, a small fraction of the budget may be allocated to extraneous design features such as sound, aesthetic design improvements, countermeasures to protect against enemy targeting, etc.

Table 8.2 i provides the estimated budget breakdown for the all-up robot. Table 8.2 ii provides the estimated budget breakdown for spare components.

8.2.1 Final Battlebot Cost Breakdown

Figure 8.2.1 i shows the cost breakdown of the robot.

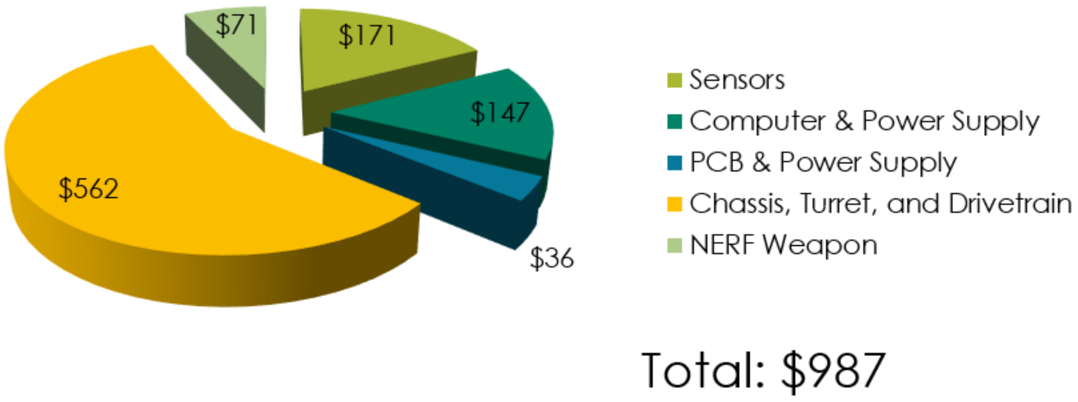


Figure 8.2.1 i: Battlebot Cost Breakdown

8.3 PARTS AQUISITION

This section includes lists of the actual parts acquired during Fall 2016. Most of, if not all, the items procured were ordered and funded through the UCF/Lockheed Martin account. Table 8.3 i includes all the components and items that were procured to support research and development for Green Team. The items procured for research and development purposes will not affect the budget for the actual all-up robot that will be used for the competition in Spring 2017. Table 8.3 ii shows the list of all components and items that will be included in the budget for the as-built robot.

As can be deduced from table 8.5 i, very little has been spent on development. Given the original goal specified was that less than $500 would be allocated to research expenditures, this is only good news. It is still not expected that more than $500 will be spent on research. Regardless, there is no penalty to spending the maximum $1000. With that knowledge, additional resources may likely be allocated to experimenting with additional sensor modalities in the future.

Though closer to the maximum allowance regarding final build components, actual expenditures remain at a minimum with plenty of buffer room. Notably, no components have been procured regarding the construction of the manually controlled robot chassis. The estimated cost for the robot chassis is still under approximation by the mechanical engineers however, early estimations point at a cost of around $200 which is excellent.

Research and Development: Components and Spares (not to exceed $1000)						
Item #	**Vendor**	**Part Nomenclature**	**Catalog #**	**Qty**	**Price**	**Total**
1	Amazon	IEIK UNO R3 Board ATmega328P with USB Cable for Arduino	B00P2FX9WY	1	$9.99	$9.99
2	Amazon	2 4 x AA 6V Battery Holder Case with ON/OFF Switch	B00HR93NJM	1	$4.98	$4.98
3	Amazon	AA Performance Alkaline Batteries (20-Pack)	B00NTCH52W	1	$7.90	$7.9
4	Amazon	2 9V Battery Holder with ON/OFF Switch	B00FHJTOVU	1	$2.22	$2.22
5	Amazon	9 Volts Alkaline Batteries (8-Pack)	B00MH4QM1S	1	$9.99	$9.99
6	Amazon	D Cell Alkaline Batteries	B00MH4QKP6	1	$12.34	$12.34
7	Amazon	200 mm Male – Female Jumper Cables	B00A6SOGC4	1	$1.09	$1.92
8	Amazon	2 x 40 Right Angle Male Pin Header	B008999TAG	1	$5.22	$5.22
9	RobotShop	Lynxmotion Large Pan / Tilt Kit	RB-Lyn-681	1	$39.99	$39.99
					Total	$94.55

Table 8.3 i: Research and Development: Components and Spares

All-Up Robot: Competition Final Build MAIN Components
(not to exceed $1000)

Item #	Vendor	Nomenclature	Catalog #	Qty	Price	Total
1	Amazon	Nerf Rival Khaos MXVII-4000 Blaster (Red)	B01ASW62QK	1	$64.88	$64.88
2	RobotShop	Hitec 31805 HS-805BB Mega Giant Scale 2BB Servo		1	$39.99	$39.99
3	Amazon	Hitec 32645S HS-645MG High Torque 2BB Metal Gear Servo	B003T6RSVQ	1	$29.49	$29.49
4	Amazon	Portable Charger RAVPower 1300mah (5V / 4.5A Dual USB Output) Power Bank External Battery Pack - Black	B00MPIGPUY	1	$26.99	$26.99
5	Amazon	Logitech C270 Desktop or Laptop Webcam, HD 720p Widescreen	B004FHO5Y6	1	$20.98	$20.98
6	Amazon	Intel Compute Stick CS125 Computer with Intel Atom x5 Processor and Windows 10	B01AZC4NHS	1	$132.49	$132.49
7	RobotShop	Devantech SRF08 Ultrasonic Range Finder	RB-Dev-02	1	$49.00	$49.00
8	Amazon	64GB microSD Card	B010Q588D4	1	$19.29	$19.29
					Total	$363.82

Table 8.3 ii: All-Up Robot: Competition Final Build Components

Figure 8.3 i, shows a cleaned-up view of all the currently procured components. Some components are still in the process of being shipped such as the SD card. Within the picture: the NERF gun, webcam, ultrasonic, rechargeable battery, intel stick, Arduino microcontroller, pan servo, 9-volt battery, 6-volt battery can be seen. Several other components, used to create the bread board test circuit, are featured as well such as several resistors, diodes, capacitors, and transistors. Further in development, the actual cost of the printed circuit board will be included. The total should not exceed $100 for the printed circuit and that cost would include at least 3 copies. So, the direct affect to the all-up robot cost would be approximately $30.

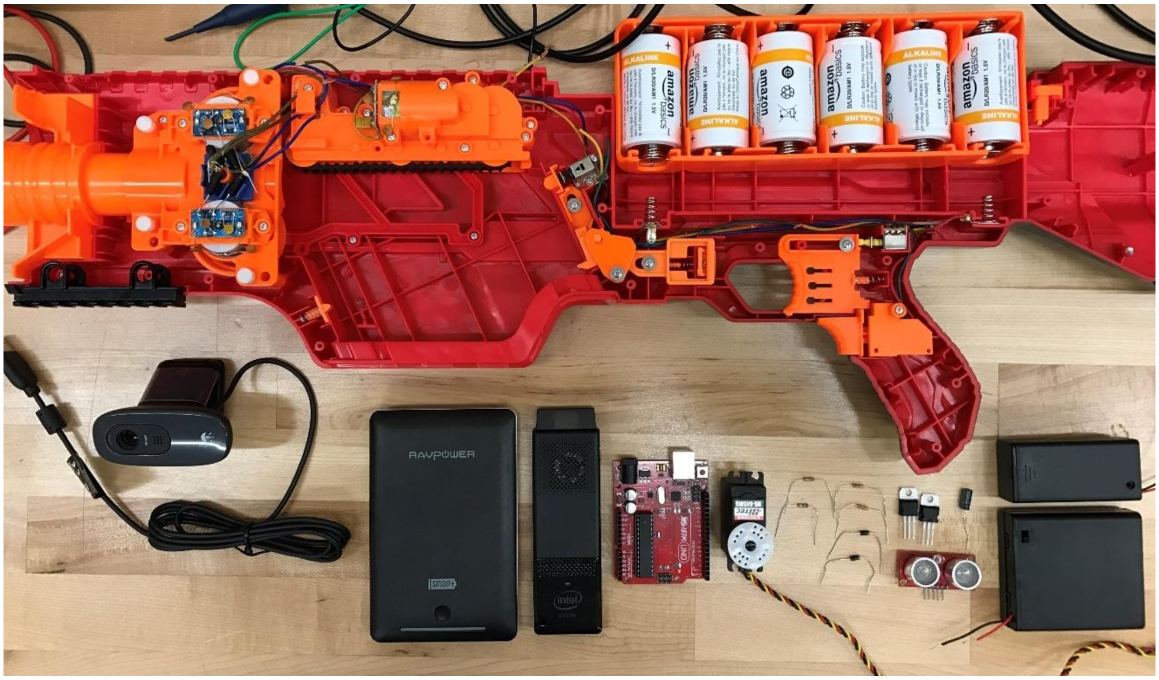


Figure 8.3 i: Parts Selection Currently Procured and Available

APPENDIX

I. REFERENCES

[1]
Thurnher, Jeffrey S. Legal Implications of Fully Autonomous Targeting. N.p.: NDU, 2012. Web. 18 Sept. 2016

[2]
"Collision Avoidance System." Wikipedia. Wikimedia Foundation, n.d. Web. 19 Sept. 2016.

[3]
Hou, Zhiqiang, and Chongzhao Han. A Target Tracking System Based on Radar and Image Fusion. N.p.: Institute of Automation School of Electronics and Information Engineering Xi'an Jiaotong U, 2003. Web. 18 Sept. 2016

[4]
"Automatic Target Recognition." Wikipedia. Wikimedia Foundation, n.d. Web. 19 Sept. 2016.

[5]
Verly, J.G., R.l. Delanoy, and D.E. Dudgeon. "Machine Intelligence Technology for Automatic Target Recognition." Lincoln Laboratory Journal 2.2 (1989): n. pag. Web. 20 Sept. 2016.

[6]
Verly, J.G., R.l. Delanoy, and D.E. Dudgeon. "Machine Intelligence Technology for Automatic Target Recognition." Lincoln Laboratory Journal 2.2 (1989): n. pag. Web. 20 Sept. 2016.

[7]
Radar and Sensors Products and Solutions · Lockheed Martin. N.p., n.d. Web. 19 Sept. 2016.

[8]
BrittLiv. "Nerf Vulcan Sentry Gun." Instructables.com. N.p., 2013. Web. 27 Sept. 2016.

[9]
Caponi, Elso, Michael Lakus, Ali Marar, and Jonathan Thomas. STATS Self-Targeting Autonomous Turret System. N.p.: University of Central Florida, 2014. PDF.

[10]
Park, John. "Automatic Nerf Sentry Gun." John Park. N.p., 08 July 2016. Web. 27 Sept. 2016.

[11]
Reid, Diante, Liem Huynh, and Sean Day. The Autonomous Visual Rover. N.p.: University of Central Florida, 10 Aug. 2009. DOCX.

[12]
Lanzone, Daniel, Mike Roosa, Ryan Tochtermann, and Charlie Grubbs. S.H.A.S. Bot. N.p.: University of Central Florida, 3 Aug. 2010. PDF.

[13]
Colon, Hector, Adam Horton, Kyle Steighner, and Nicholas Yielding. Automated Targeting Proximity Turret. N.p.: University of Central Florida, 2010. PDF.

[14]
"Project Sentry Gun." Project Sentry Gun. N.p., n.d. Web. 28 Nov. 2016.

[15]
"Installing an Arduino Bootloader." Installing an Arduino Bootloader. N.p., n.d. Web. 28 Nov. 2016.

[16]
"CohuHD Rugged HD Surveillance Video Camera for Critical Infrastructure." CohuHD Rugged HD Surveillance Video Camera for Critical Infrastructure. N.p., n.d. Web. 28 Nov. 2016.

[17]
"Understanding RC Servos – Digital, Analog, Coreless, Brushless." RCHelicopterFun.com. N.p., n.d. Web. 28 Nov. 2016.

[18] & [39]
Daniel Healy, Nick Ho Lung, and Sayeed Tahseen. Battlebot Final Project Documentation. N.p.: University of Central Florida. Web. 3 December 2016

[19]
"Sharp Infrared Ranger Comparison." Acroname. N.p., 31 May 2013. Web. 19 Sept. 2016.

[20]
"What Is LiDAR?" LiDAR-UK. N.p., n.d. Web. 19 Sept. 2016.

[21]
"Ultrasonic Sensing." Rockwell Automation, n.d. Web. 27 Sept. 2016.

[22]
Reed, Frances. "How Do Servo Motors Work." How Servo Motors Work. Jameco, n.d. Web. 27 Sept. 2016.

[23]
Brain, Marshall. "How Microcontrollers Work." HowStuffWorks. N.p., 01 Apr. 2000. Web. 27 Sept. 2016.

[24]
Kalinich, Sean. "Intel Vs. ARM; RISC against CISC All over Again..." DecryptedTech. N.p., 16 Jan. 2013. Web. 23 Oct. 2016.

[25]
Roberts, Eric. "RISC vs. CISC." RISC vs. CISC. Stanford University, n.d. Web. 24 Oct. 2016.

[26] SHARP Corporation," Distance Measuring Sensor Unit," GP2Y0A21YK datasheet, 2005

[27] SHARP Corporation," Distance Measuring Sensor Unit," GP2Y0A60SZ0F datasheet,

[28] SHARP Corporation," Distance Measuring Sensor Unit," GP2Y0A02YK0F datasheet, 01 Dec 2006

[29]
SHARP Corporation," Distance Measuring Sensor Unit," GP2Y0A710K0F datasheet, 01 Dec 2006

[30]
Parallax Inc., "Ultrasonic Distance Sensor," Parallax PING))) ™ documentation, 11 September 2009

[31]
"SRF10 Ultrasonic Range Finder." Robot Electronics. N.p., n.d. Web. 12 Oct. 2016.

[32]
Wei, Jessy. "URM04 V2.0 (SKU:SEN0002)." Robot Wiki. N.p., 16 Aug. 2010. Web. 24 Oct. 2016.

[33]
"SRF08 Ultrasonic Range Finder." Robot Electronics. N.p., n.d. Web. 24 Oct. 2016.

[34]
"HS-5625MG High Speed, Metal Gear Digital Sport Servo." HITEC RCD USA. N.p., n.d. Web. 24 Oct. 2016.

[35]
"HS-5645MG High Torque, Metal Gear Digital Sport Servo." HITEC RCD USA. N.p., n.d. Web. 24 Oct. 2016.

[36]
"HS-805BB Mega Giant Scale Servo." HITEC RCD USA. N.p., n.d. Web. 24 Oct. 2016.

[37]
"HS-645MG High Torque, Metal Gear Premium Sport Servo." HITEC RCD USA. N.p., n.d. Web. 24 Oct. 2016.

[38]
Murray, Matthew. "Raspberry Pi 3 Model B." PCMAG. N.p., 09 Mar. 2016. Web. 24 Oct. 2016.

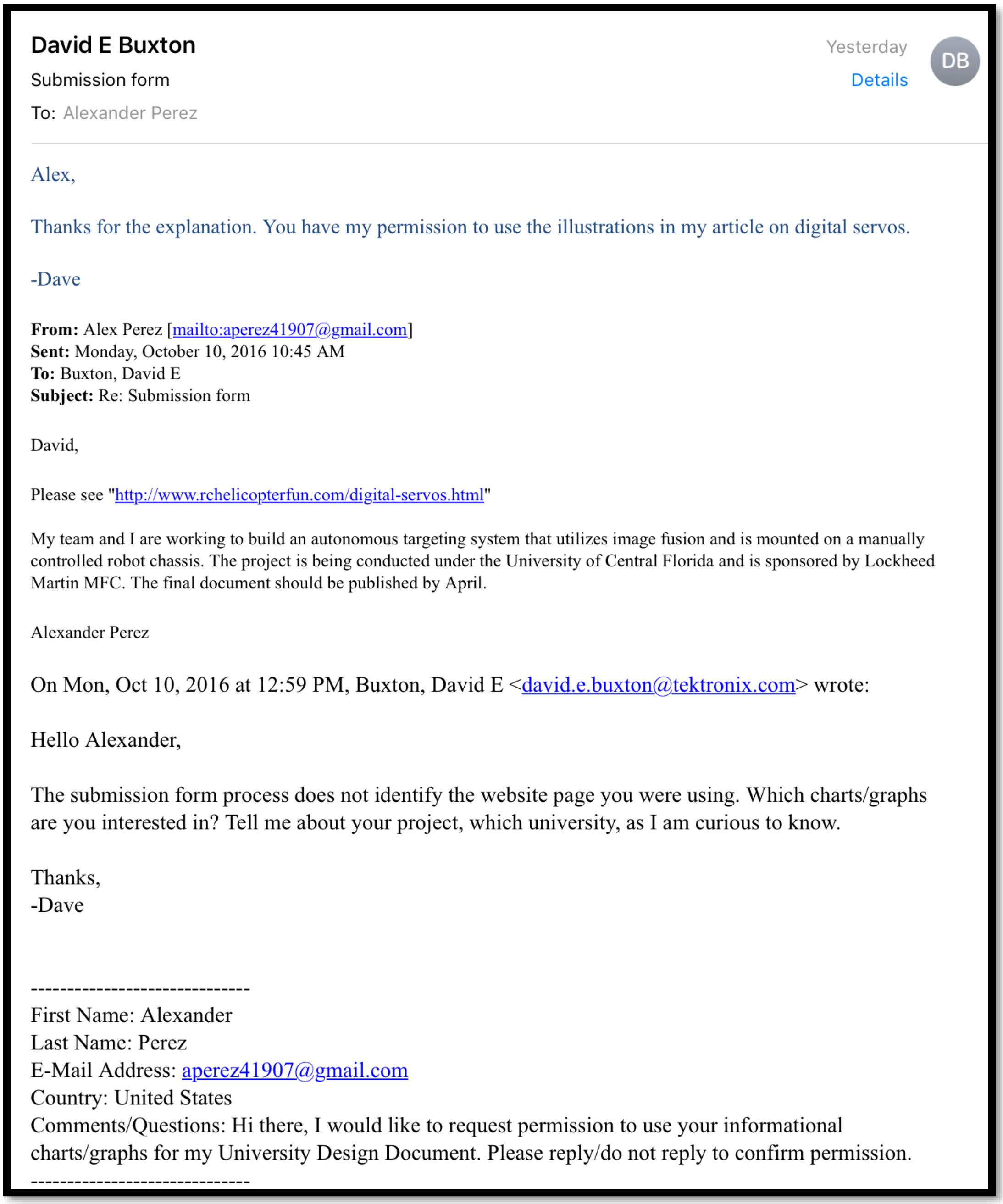
[40]
Fernando, Shermal Ruwantha. "What Is OpenCV?" *What Is OpenCV?* N.p., n.d. Web. 02 Dec. 2016.

[41]
"About." *About OpenCV*. N.p., n.d. Web. 02 Dec. 2016.

[42]
"Face Recognition with OpenCV." *OpenCV 2.4.13.1 Documentation*. N.p., n.d. Web. 02 Dec. 2016.

[43]
Garmin, and Garmin Ltd. or Its Subsidiaries. "LIDAR-Lite V3." Garmin. N.p., n.d. Web.19 Apr. 2017.

II. PERMISSIONS



David E Buxton

Submission form

To: Alexander Perez

Alex,

Thanks for the explanation. You have my permission to use the illustrations in my article on digital servos.

-Dave

From: Alex Perez [mailto:aperez41907@gmail.com]
Sent: Monday, October 10, 2016 10:45 AM
To: Buxton, David E
Subject: Re: Submission form

David,

Please see "http://www.rchelicopterfun.com/digital-servos.html"

My team and I are working to build an autonomous targeting system that utilizes image fusion and is mounted on a manually controlled robot chassis. The project is being conducted under the University of Central Florida and is sponsored by Lockheed Martin MFC. The final document should be published by April.

Alexander Perez

On Mon, Oct 10, 2016 at 12:59 PM, Buxton, David E <david.e.buxton@tektronix.com> wrote:

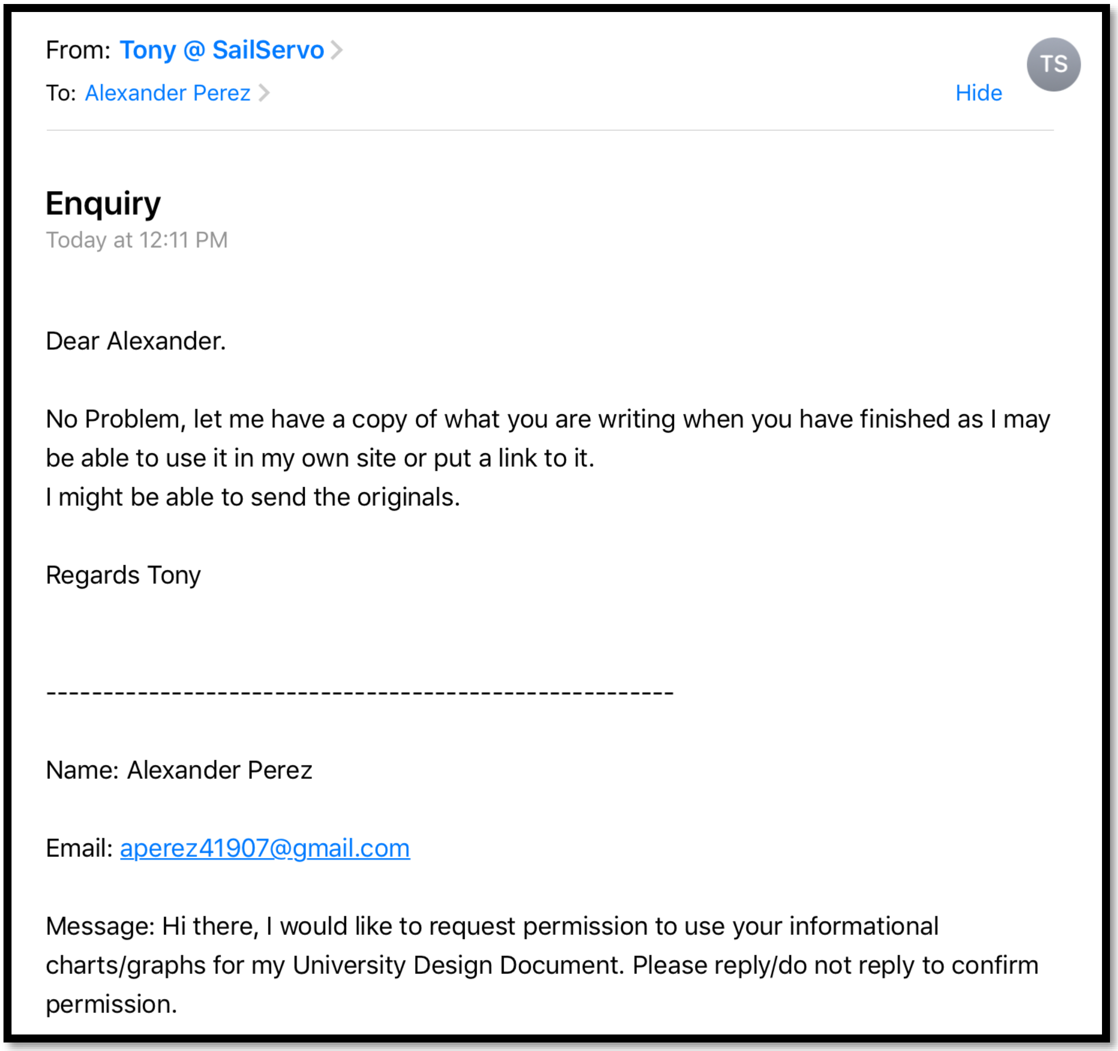
Hello Alexander,

The submission form process does not identify the website page you were using. Which charts/graphs are you interested in? Tell me about your project, which university, as I am curious to know.

Thanks,
-Dave

First Name: Alexander
Last Name: Perez
E-Mail Address: aperez41907@gmail.com
Country: United States
Comments/Questions: Hi there, I would like to request permission to use your informational charts/graphs for my University Design Document. Please reply/do not reply to confirm permission.

Screen Capture i: Permission to use Figure 3.2.3.2 iii



From: Tony @ SailServo

To: Alexander Perez

Enquiry

Today at 12:11 PM

Dear Alexander.

No Problem, let me have a copy of what you are writing when you have finished as I may be able to use it in my own site or put a link to it.
I might be able to send the originals.

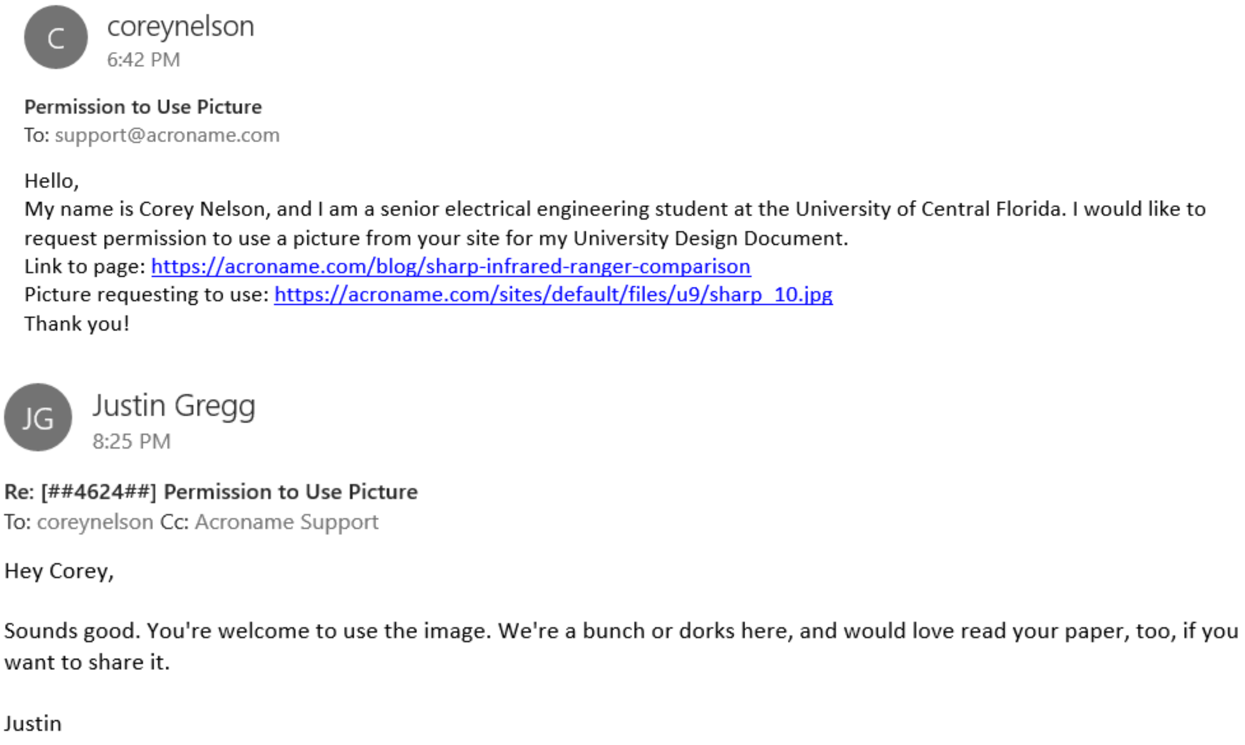
Regards Tony

Name: Alexander Perez

Email: aperez41907@gmail.com

Message: Hi there, I would like to request permission to use your informational charts/graphs for my University Design Document. Please reply/do not reply to confirm permission.

Screen Capture ii: Permission to use Figure 3.2.3.2 i & Figure 3.2.3.2 ii



coreynelson
6:42 PM

Permission to Use Picture
To: support@acroname.com

Hello,
My name is Corey Nelson, and I am a senior electrical engineering student at the University of Central Florida. I would like to request permission to use a picture from your site for my University Design Document.
Link to page: https://acroname.com/blog/sharp-infrared-ranger-comparison
Picture requesting to use: https://acroname.com/sites/default/files/u9/sharp_10.jpg
Thank you!

Justin Gregg
8:25 PM

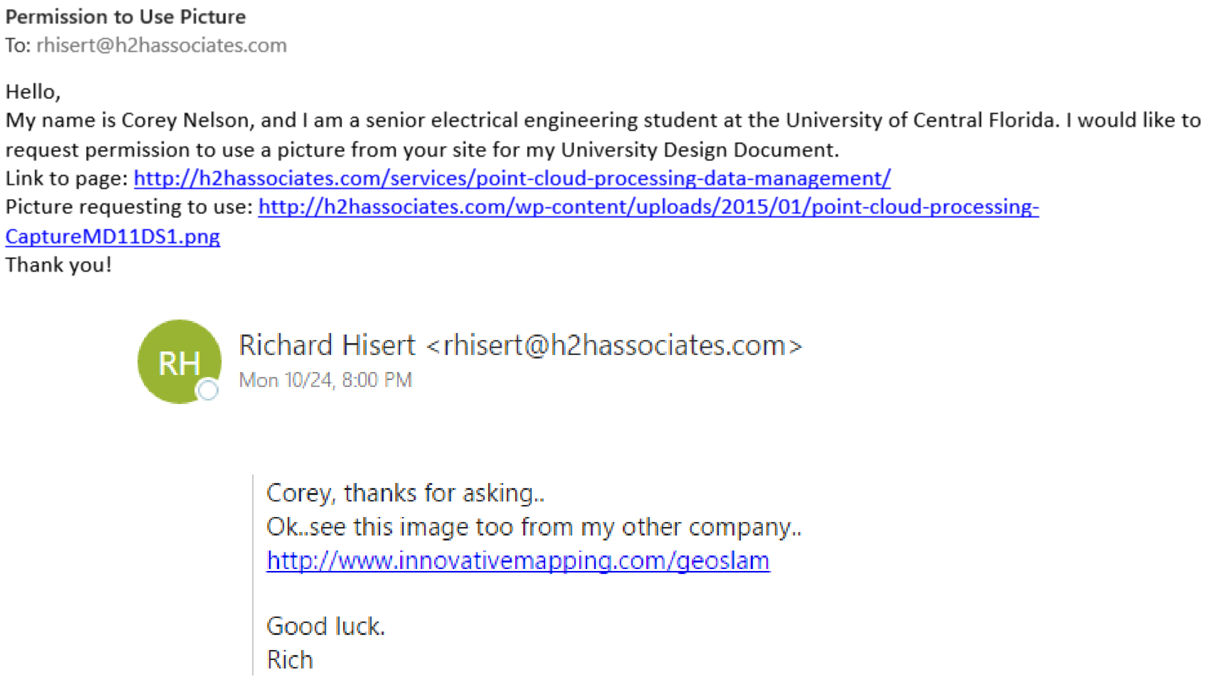
Re: [##4624##] Permission to Use Picture
To: coreynelson Cc: Acroname Support

Hey Corey,

Sounds good. You're welcome to use the image. We're a bunch or dorks here, and would love read your paper, too, if you want to share it.

Justin

Screen Capture iii – Permission to Use Figure 3.2.1 i



Permission to Use Picture
To: rhisert@h2hassociates.com

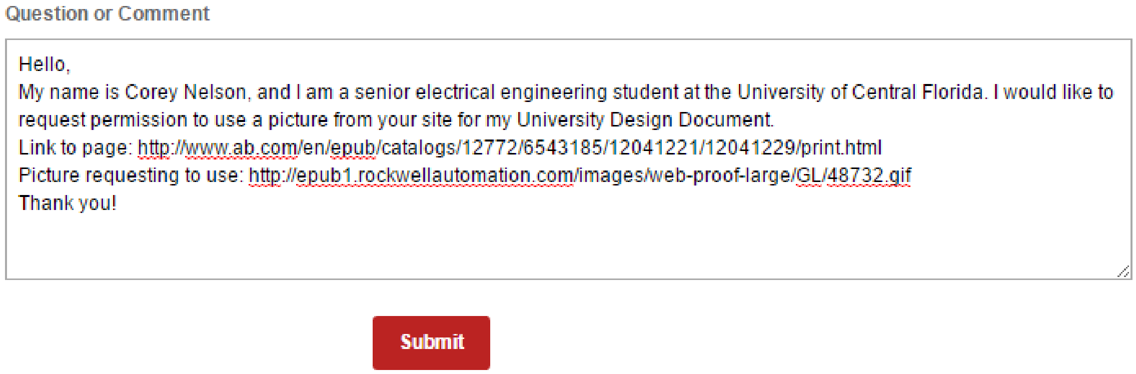
Hello,
My name is Corey Nelson, and I am a senior electrical engineering student at the University of Central Florida. I would like to request permission to use a picture from your site for my University Design Document.
Link to page: http://h2hassociates.com/services/point-cloud-processing-data-management/
Picture requesting to use: http://h2hassociates.com/wp-content/uploads/2015/01/point-cloud-processing-CaptureMD11DS1.png
Thank you!

Richard Hisert <rhisert@h2hassociates.com>
Mon 10/24, 8:00 PM

Corey, thanks for asking..
Ok..see this image too from my other company..
http://www.innovativemapping.com/geoslam

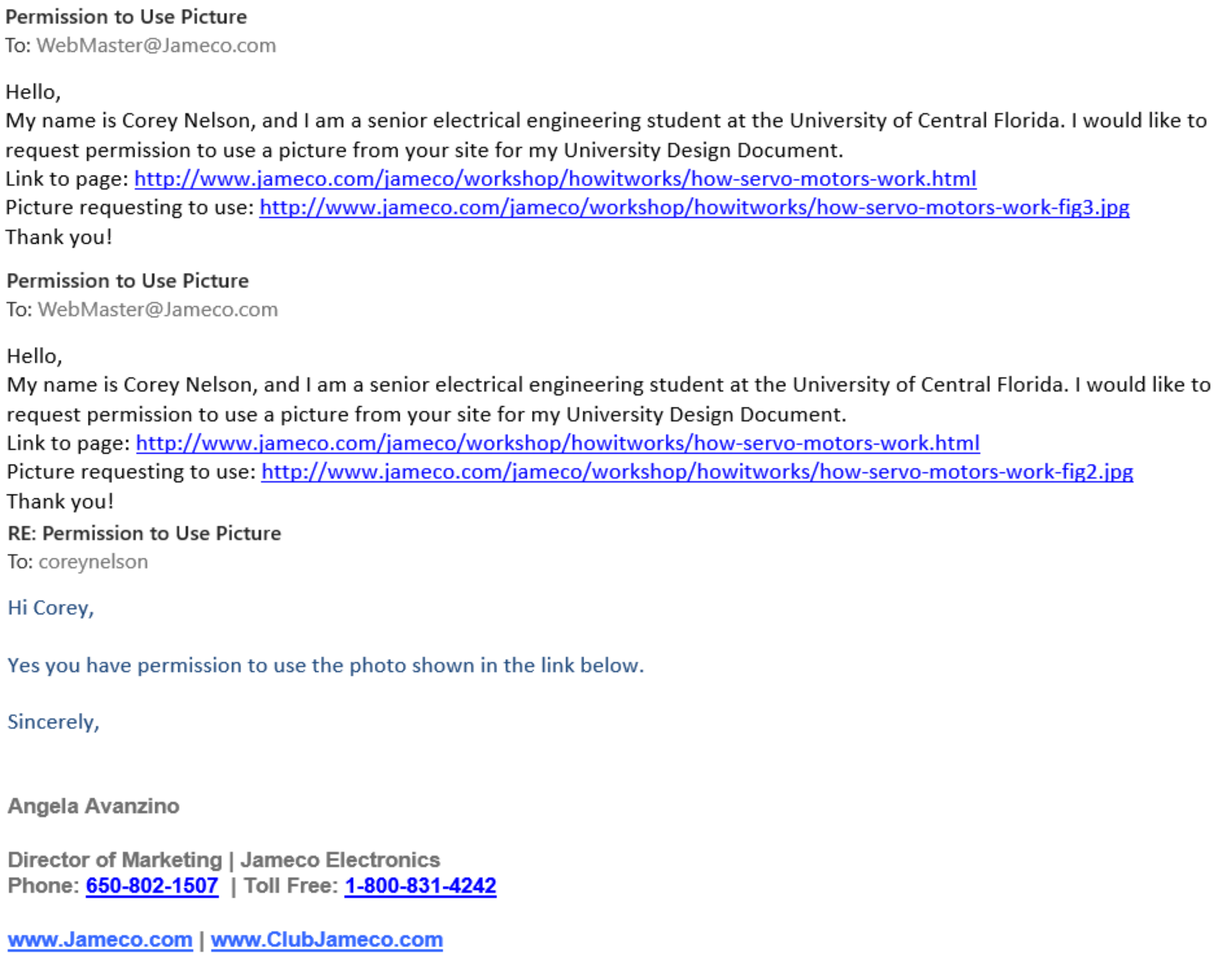
Good luck.
Rich

Screen Capture iv – Permission to Use Figure 3.2.1.2 i



Hello,
My name is Corey Nelson, and I am a senior electrical engineering student at the University of Central Florida. I would like to request permission to use a picture from your site for my University Design Document.
Link to page: http://www.ab.com/en/epub/catalogs/12772/6543185/12041221/12041229/print.html
Picture requesting to use: http://epub1.rockwellautomation.com/images/web-proof-large/GL/48732.gif
Thank you!

Screen Capture v – Permission to Use Figure 3.2.1.3 i



Permission to Use Picture
To: WebMaster@Jameco.com

Hello,
My name is Corey Nelson, and I am a senior electrical engineering student at the University of Central Florida. I would like to request permission to use a picture from your site for my University Design Document.
Link to page: http://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html
Picture requesting to use: http://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work-fig3.jpg
Thank you!

Permission to Use Picture
To: WebMaster@Jameco.com

Hello,
My name is Corey Nelson, and I am a senior electrical engineering student at the University of Central Florida. I would like to request permission to use a picture from your site for my University Design Document.
Link to page: http://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html
Picture requesting to use: http://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work-fig2.jpg
Thank you!

RE: Permission to Use Picture
To: coreynelson

Hi Corey,

Yes you have permission to use the photo shown in the link below.

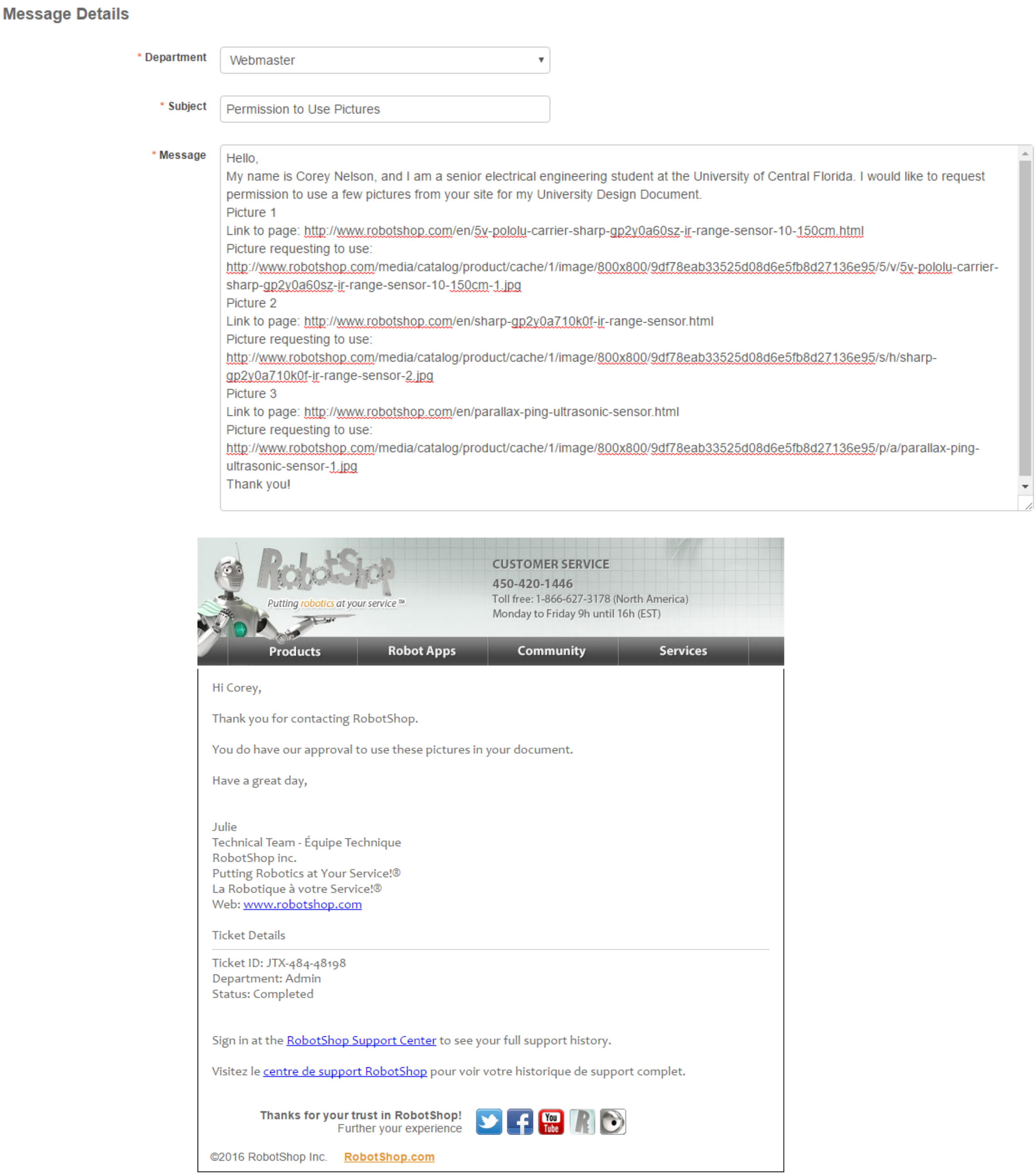
Sincerely,

Angela Avanzino

Director of Marketing | Jameco Electronics
Phone: 650-802-1507 | Toll Free: 1-800-831-4242

www.Jameco.com | www.ClubJameco.com

Screen Capture vi – Permission to Use Figures 3.2.3 i and 3.2.3 ii



Message Details

Hello,
My name is Corey Nelson, and I am a senior electrical engineering student at the University of Central Florida. I would like to request permission to use a few pictures from your site for my University Design Document.
Picture 1
Link to page: http://www.robotshop.com/en/5v-pololu-carrier-sharp-gp2y0a60sz-ir-range-sensor-10-150cm.html
Picture requesting to use:
http://www.robotshop.com/media/catalog/product/cache/1/image/800x800/9df78eab33525d08d6e5fb8d27136e95/5/v/5v-pololu-carrier-sharp-gp2y0a60sz-ir-range-sensor-10-150cm-1.jpg
Picture 2
Link to page: http://www.robotshop.com/en/sharp-gp2y0a710k0f-ir-range-sensor.html
Picture requesting to use:
http://www.robotshop.com/media/catalog/product/cache/1/image/800x800/9df78eab33525d08d6e5fb8d27136e95/s/h/sharp-gp2y0a710k0f-ir-range-sensor-2.jpg
Picture 3
Link to page: http://www.robotshop.com/en/parallax-ping-ultrasonic-sensor.html
Picture requesting to use:
http://www.robotshop.com/media/catalog/product/cache/1/image/800x800/9df78eab33525d08d6e5fb8d27136e95/p/a/parallax-ping-ultrasonic-sensor-1.jpg
Thank you!

RobotShop
Putting robotics at your service™
CUSTOMER SERVICE
450-420-1446
Toll free: 1-866-627-3178 (North America)
Monday to Friday 9h until 16h (EST)
Products
Robot Apps
Community
Services

Hi Corey,

Thank you for contacting RobotShop.

You do have our approval to use these pictures in your document.

Have a great day,

Julie
Technical Team - Équipe Technique
RobotShop inc.
Putting Robotics at Your Service!®
La Robotique à votre Service!®
Web: www.robotshop.com

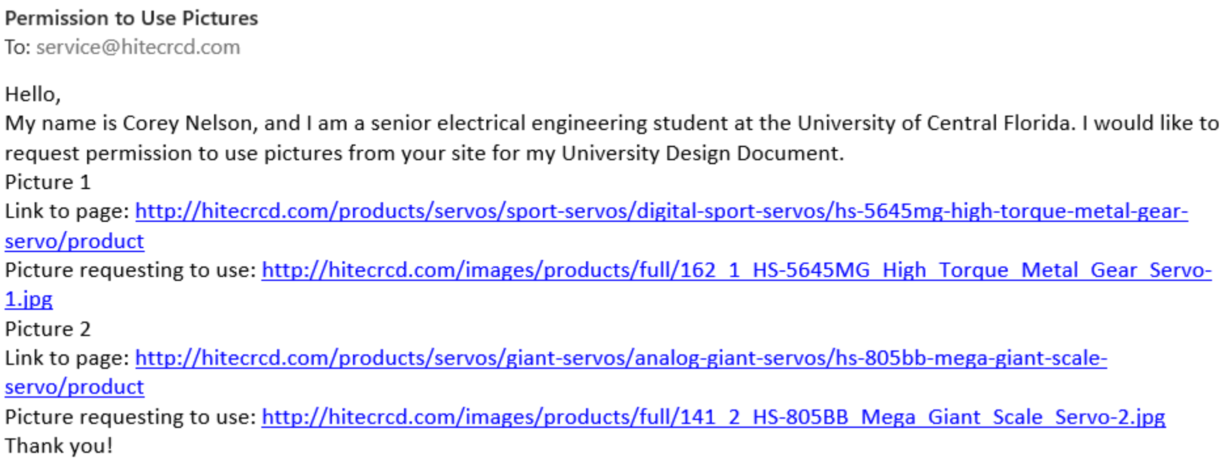
Ticket Details

Ticket ID: JTX-484-48198
Department: Admin
Status: Completed

Sign in at the RobotShop Support Center to see your full support history.

Visitez le centre de support RobotShop pour voir votre historique de support complet.

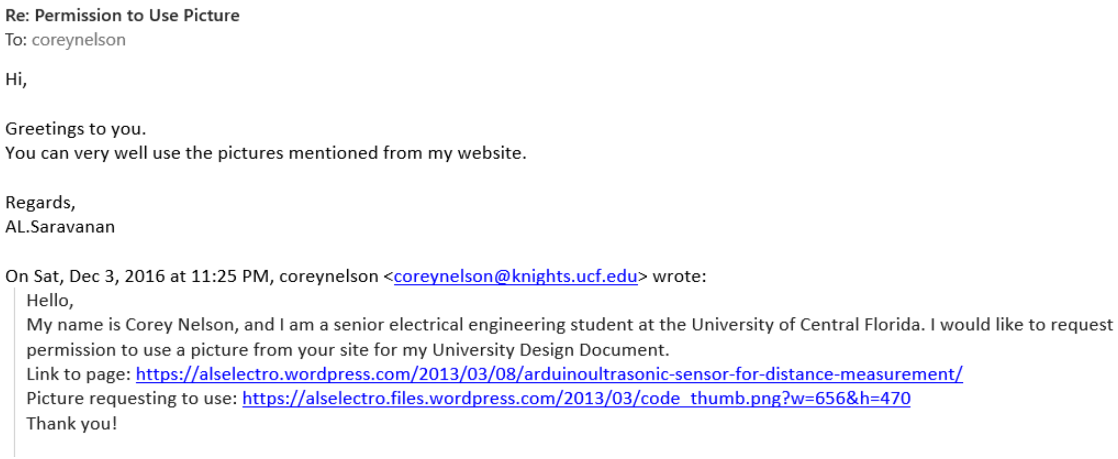
Screen Capture vii – Permission to Use Figures 3.3.1 i, 3.3.1 ii, and 3.3.2 i



Permission to Use Pictures
To: service@hitecrcd.com

Hello,
My name is Corey Nelson, and I am a senior electrical engineering student at the University of Central Florida. I would like to request permission to use pictures from your site for my University Design Document.
Picture 1
Link to page: http://hitecrcd.com/products/servos/sport-servos/digital-sport-servos/hs-5645mg-high-torque-metal-gear-servo/product
Picture requesting to use: http://hitecrcd.com/images/products/full/162_1_HS-5645MG_High_Torque_Metal_Gear_Servo-1.jpg
Picture 2
Link to page: http://hitecrcd.com/products/servos/giant-servos/analog-giant-servos/hs-805bb-mega-giant-scale-servo/product
Picture requesting to use: http://hitecrcd.com/images/products/full/141_2_HS-805BB_Mega_Giant_Scale_Servo-2.jpg
Thank you!

Screen Capture viii – Permission to Use Figures 3.3.3.1 i and 3.3.3.2 i



Re: Permission to Use Picture
To: coreynelson

Hi,

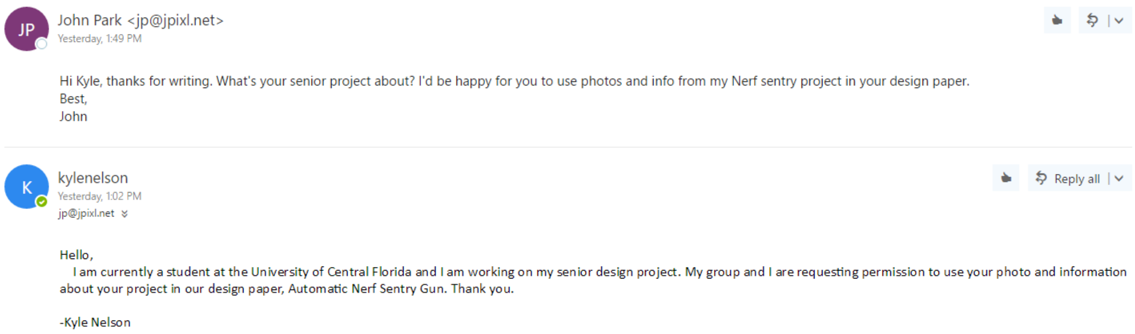
Greetings to you.
You can very well use the pictures mentioned from my website.

Regards,
AL.Saravanan

On Sat, Dec 3, 2016 at 11:25 PM, coreynelson <coreynelson@knights.ucf.edu> wrote:

> Hello,
> My name is Corey Nelson, and I am a senior electrical engineering student at the University of Central Florida. I would like to request permission to use a picture from your site for my University Design Document.
> Link to page: https://alselectro.wordpress.com/2013/03/08/arduinoultrasonic-sensor-for-distance-measurement/
> Picture requesting to use: https://alselectro.files.wordpress.com/2013/03/code_thumb.png?w=656&h=470
> Thank you!

Screen Capture ix – Permission to Use Figure 7.4.1.1 i



John Park <jp@jpixl.net>
Yesterday, 1:49 PM

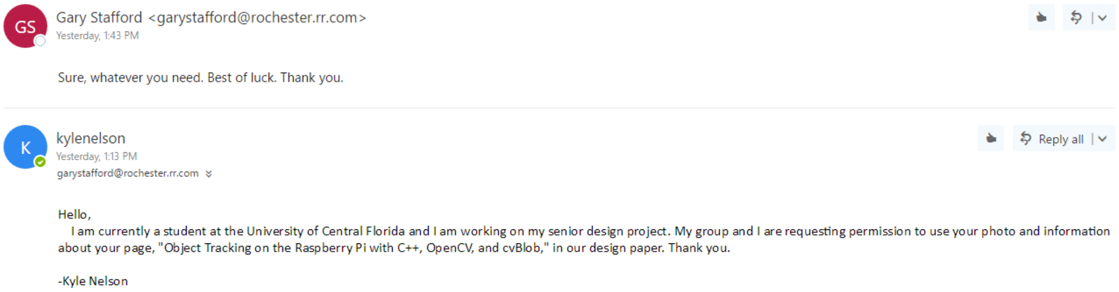
Hi Kyle, thanks for writing. What's your senior project about? I'd be happy for you to use photos and info from my Nerf sentry project in your design paper.
Best,
John

kylenelson
Yesterday, 1:02 PM
jp@jpixl.net

Hello,
I am currently a student at the University of Central Florida and I am working on my senior design project. My group and I are requesting permission to use your photo and information about your project in our design paper, Automatic Nerf Sentry Gun. Thank you.

-Kyle Nelson

Screen Capture x– Permission to Use Figure 3.1.3 i



Gary Stafford <garystafford@rochester.rr.com>
Yesterday, 1:43 PM

Sure, whatever you need. Best of luck. Thank you.

kylenelson
Yesterday, 1:13 PM
garystafford@rochester.rr.com

Hello,
I am currently a student at the University of Central Florida and I am working on my senior design project. My group and I are requesting permission to use your photo and information about your page, "Object Tracking on the Raspberry Pi with C++, OpenCV, and cvBlob," in our design paper. Thank you.

-Kyle Nelson

Screen Capture xi– Permission to Use Figure 5.5.2 i