Initial Project and Group Identification Document

Motion-Detecting Sentry

University of Central Florida

Department of Electrical Engineering and Computer Science

Dr. Lei Wei & Samuel Richie

Senior Design 1

Group 33 Members:

Liderma Guerry – Computer Engineering

Quintin Jimenez – Computer Engineering

Michael Macallister – Computer Engineering

Kaitlyn Martin – Electrical Engineering

Table of Contents

1.0 Executive Summary

Our project idea came about while brainstorming. Initially, we were looking into making a robot that could play cornhole. However, when we started doing a little research and considering physics, it became apparent that a project of that caliber would become quite costly and physically complex. As a result, we began looking at alternatives that were still in the same scope, and we eventually came to the idea of making a turret. This turret would use paintballs to mark individuals of interest and make them easily identifiable.

This project aims to design and manufacture a turret that uses a paintball gun. The turret would consist of a modified paintball gun on a motorized stand, which would allow it to aim at the defined targets. The aiming system will use computer vision software and a camera to detect its targets. The aiming system will communicate with the hardware which will be responsible for the motors which will turn the turret. Our goal is to create a turret that uses computer vision to identify and hit its targets. The turret should be able to use computer vision to detect when someone is moving within its range and then tag them with a paintball.

Our hope for this project is to develop an additional security measure that could be used by businesses to help deter crime or make the perpetrator easier to identify in the event that criminal trespassing does occur. This product would provide a nonlethal alternative in comparison to using live rounds, while also decreasing risk to security officers. This project could be an excellent way of furthering less violent safety protocols.

This project could also be used for entertainment, specifically it could add an additional challenge to a game of paintball. For example, each team could have one turret to strategically place on the field, forcing the other team to have to work around it, or allowing the team to chase the opposing team into the turrets of sight. Our project would also be able to be easily modified to use a laser instead of a paintball gun, such that it could be used in a laser tag arena.

The outcome of this project will be a lightweight, movable and accurate paintball turret capable of hitting a moving target, all while remaining cost effective. Such that it could be usable and accessible by the majority of those interested.

This report will document the decision-making and design process that will go into creating this turret. This paper will cover the motivation and goals for this project, as well as the necessary functions and objectives required for the system. The following will document the research and each of the subsystems of the project, and the processes we include in our prototype.

2.0 Project Description

The members of Group 33 decided to create a stationary, motion-detecting, automatic sentry turret. First, the sentry turret will be stationary, meaning that it will not actively seek out targets; rather, it will passively wait for a potential target to come within range, then determine the necessary course of action. The turret will utilize motion sensors to determine when a potential target has entered its range of effects. Next, the sentry turret will activate a camera and accompanying computer vision algorithm to determine whether the object that triggered the motion sensor is, in fact, a target. Finally, the turret will either enter a rest state until motion is detected once again, or the turret will arm itself and fire at the target using a mounted paintball gun.

This section will explain the motivations behind the turret as well as the system requirements the team will fulfil for the project. In addition, the functional and design goals will be outlined within this section.

2.1 Motivation

The primary motivation behind this project was to develop unique technology using concepts learned through the study of electrical and computer engineering at the University of Central Florida. The members of Group 33 have made the decision to accomplish this goal by constructing a stationary sentry turret which uses computer vision to locate and fire projectiles at humanoid targets both moving and stationary within range of the device

Further motivation for this project was to create a device with variable applications in the real world. The turret developed by Group 33 will be useful for property owners to defend their homes and businesses. Scalability and modifications to the prototype will allow for potential military applications, as well. The device will be modifiable and scalable to allow for safer use in different scenarios. Such applications include using the modified system in paintball tournaments as well as games of laser tag if the system were to be outfitted with a laser diode for hit indication instead of a paintball gun. These modifications would be simple to apply for any individual or group looking to repurpose the sentry turret project to perform some task other than that which is accomplished by the prototype being presented.

Additionally, it will be possible for the project, either in its entirety or simply in part, to be reverse engineered so that the technologies used/introduced by the sentry turret can be applied to different devices and systems altogether. The ability to make a device that is easily modifiable and affordable can influence people to explore more with these technologies and to make not only security devices, but assistive devices as well. Another possible implementation of the sentry turret project after modification is an automatic fire extinguisher capable of recognizing fires a distance away and propel some fire retardant onto the fires, extinguishing harmful flames. The utilization of computer vision to automate different processes and activities has become an ever-growing trend and to be able to explore these fields and the technologies specific to them would be invaluable knowledge.

2.2 Goals/Objectives

Our main goal for this project is to design a fully mod-able turret utilizing cost effect materials and software. We also wanted to use a device that is intended to be non-lethal to show this type of device could be used for many applications such as games, putting out fires, or marking targets. Using computer vision with different types of software technologies available can not only mark humanoid targets, but also fires or plants.

With our personal design in mind, our turret will be used more on the security side. If there was a break-in or trespassing, our device would fire non-lethal ammunition in the form of paintballs to mark a target for police capture. This could also work as a deterrent as the device could act as a warning to intruders.

The objective is to design a method where computer vision will be able to identify targets from a reasonable distance for marking. It will also scan the area to determine if there is a target present. Computer vision will be one of the key specifications to focus on as it need to accurately identify the target at 70% accuracy. It will also be the driving force for searching for the target within the 30-foot radius. With identifying the targets, it must also be able to fire the paintball gun.

The frame of the structure will also be a goal to work towards as it needs to be less than or equal to 40 pounds for easy transport. The paintball gun also needs to be well balanced for transversal in the horizontal and vertical planes. This will be important for firing because if the paintball gun is not well balanced the angle of the shots will not be as accurate or miss.

The turret will also have a warning light to alert possible bystanders. When a person should come into range it will set off a red light where they have a few seconds to remove themselves from firing radius. This will serve as a preventative measure of people wandering into the firing radius. The device is not meant to be hidden, but in plain view to serve as a deterrent.

With all the components that will be incorporated into the device having sufficient power will be an issue. Our goal is to have a battery that will be able to power all of our components for 3 hours. A rechargeable battery will be ideal for this situation, as replacing batteries would not be cost effective.

With all of the previous goals in mind, the project must remain cost effective. It will give others in the future a better chance to improve and replicate this project. **Table 1** (below) outlines all the major goals discussed for the project.

#	**Primary Goals**	**Secondary Goals**	**Advanced Goals**
1.	Be a modifiable turret.	Implement a motion sensor.	Use a solar powered method of recharging.
2.	Use non-lethal ammunition.	Go into a low powered mode.	More compact design.
3.	Use computer vision to identify targets.	Have the motion sensor trigger the turret wake up from a low powered state.	Try to make as water resistance as possible.
4.	Must be able to display a warning light.		Use high-cost materials to build housing structure, like metal.
5.	Must be able to balance paintball gun.		Have an app to alert when and where a target is detected.
6.	Able to be easily transported.		
7.	All parts must be cost effective.		
8.	Paintball gun must be able to fire and hit targets.		
9.	Have a sufficient power supply.		

Table 1: Goals of the Project

The majority of our goals will be primary goals. This will ensure all basic needs are met for the turret. Without these set of goals our prototype would not have the complexity required of a Senior Design Project. Although we want to maintain some complexity, we do not want to stretch our goals too far into a project that we cannot build in a semester. Therefore, our primary goals will be outline what is sufficient to the project, and what we will need to show our panel. First, we will have some complexity with implementing computer vision. A warning light will alert targets that they will need to remove themselves from the premises. The turret will mark target with paintballs that are non-lethal. Our power supply will help power components connect to the PCB. A this could be a modifiable project as the paintball gun can be switched out. Of course, we will take steps to ensure everything will be lightweight, and an average adult could transport the turret. For our basic goals, this prototype must be cost effective for others to build.

If there is enough time and buget, we will move onto our secondary goals. This part can only be included if our testing is sufficient and we have met all our primary goals ahead of time. Of course, we must account for testing, and any troubleshooting that will arise later in the project. There might also be some oversights where we will have to re-analyze the design of our project to work. It would involve ways to save power. If our prototype could go into sleep mode it would save on power, and able to last longer on the battery. When the motion sensor senses movement, the turret would fully power back on. Our priority is to have a working prototype before we can move onto any additional goals.

Lastly there are advanced goals. This is something our group would like to have if our project is to be expanded upon one day. For this we would need a much larger budget, and a longer time period for planning. The implementation would need to be significantly longer than a semester, but the project would be something far more optimal than our primary goals of just working. Having the ability to solar charge would greatly prolong the use of the prototype, and would be a cost-effective solution for home protection over time. A stronger structure could also increase the longevity of the prototype. Using a metal like aluminum would increase the structural integrity and keep the structure lightweight, and making the components water resistant would mean less upkeep and movement. An app that could update homeowners of when and where the turret activates would be a welcomed safety feature as it will alert homeowners a possible threat is near.

2.3 Overall Design

The following three sections will outline the project further. The design idea will go into what is necessary for the device integrity and stability. The function section describes our expectations of the device, and finally implementation delves into the possible uses.

2.3.1 Design Idea

The prototype would need to be lightweight yet large enough to house our chosen components. For the structure which houses the paintball gun, it would have the ability to rotate and pivot, such that the attached camera will have a wider area of view to parse to the computer, which will allow the sentry turret to find targets using computer vision (see link below). With this in mind, a sturdy material to support the weight of the chosen paintball gun is needed, so the prototype would not bend, sag, or become otherwise warped. **Figure 1** shows one such model built with sturdy enough material to support the projectile launcher attached. A base would serve as both a support and as a means to allow the structure to rotate with the guidance of a motor. The entire structure with the base would sit on top of a tripod for added stability and portability, refer to **Figure 2**. The paintball gun used is intended to be non-lethal and would fire paintballs as a deterrent, as well as to mark individuals who have been targeted by the device so that they may be identified at a later point in time. The firing action would be accomplished via a motor attached to the trigger of the paintball gun. The motor will rotate upon receiving a signal indicating that a proper target has been identified and that the paintball gun has been appropriately aimed at said target. The camera would be mounted on part of the base to detect targets with computer vision. A sensor would also be used to detect targets to turn on device when in a low powered state. Below are concepts of the prototype for visualization but will not be the final product.

https://hackaday.com/2015/12/06/airsoft-sentry-gun-keeps-your-house-guarded/

This link shows an example of the intended motion of the turret.

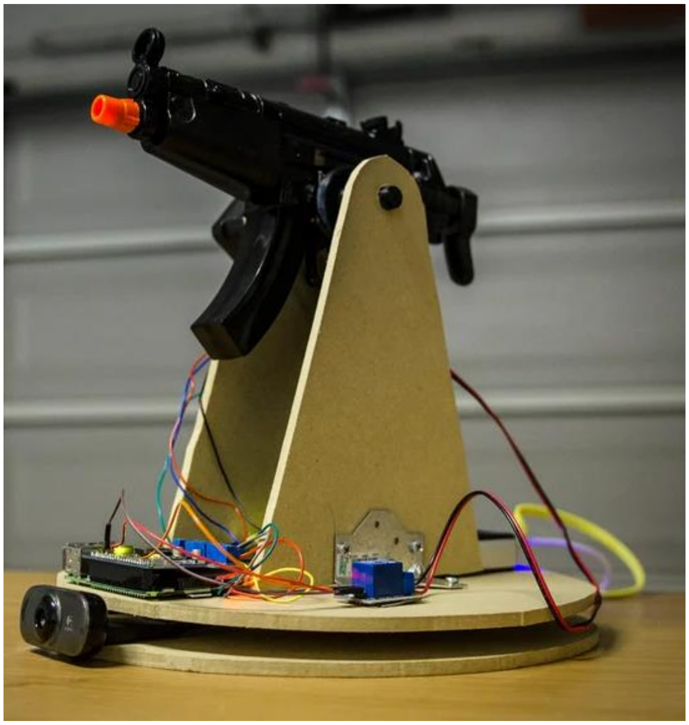


Figure 1: Prototype Concept 1

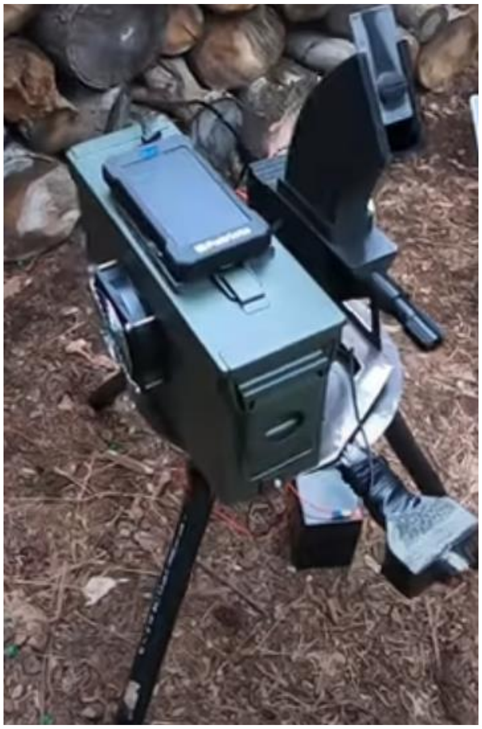


Figure 2: Prototype Concept 2

2.3.2 Function

The main purpose of this project is to create an accurate aiming system that utilizes computer vision to analyze images taken through a camera lens and identify whether there are any humans within the image, then to aim a targeting device at any humans and fire some hit indicator at the human(s). It will be able to detect targets within a 75-foot range, aim at the targets using their positions as calculated through computer vision and image analysis, and fire a projectile at the target, marking it as hit for future identification purposes. The prototype will have a power supply that will last for approximately 3 hours without recharging.

1. Once the prototype is armed, it will begin scanning for any targets within range.

2. When the prototype senses a target within range, a red light will flash to warn any individuals nearby that they may be targeted by the device. This feature will be included for safety purposes.

3. Computer vision will track the movements of any potential targets, then fire upon any individuals in range which have been deemed suitable targets using a paintball gun fitted to the device. The ammunition for this projectile launcher will be

standard paintballs so that any individual(s) struck by the sentry turret device will be marked with paint.

4. At any point in time during which there are no targets identified within range of the sentry turret, the prototype will cease firing.

The points enumerated above outline the main functions our prototype will be capable of displaying, within reason. With all parts integrated and functional these are the expected outputs. They will be sufficient to test the viability of the prototype.

2.3.3 Implementation

The sentry turret product will have a variety of implementations, not limited to the prototype's capabilities, as the features introduced will be applicable in more situations than the sentry turret itself will function in. As mentioned previously, the sentry turret may be used as crime-deterrent for homeowners and businesses alike; however, the product will be designed with scalability in mind, such that the device can be recreated at a larger scale or used with multiple similar devices for military defense.

The prototype will be designed using motion sensors and a camera linked to computer vision algorithms to locate and identify potential targets within a specific range. The sentry turret device will have several states, including an 'off' state, an 'on and disarmed' state, and an 'on and armed' state. To prevent the sentry turret from becoming armed prematurely, such as in the event that an individual intends to complete some task within range of the sentry turret after turning it on, the turret will have a unique activation method which is unlikely to be triggered without the intent to do so. In addition, the turret will include a warning system indicating when it is active and ready for use. This warning system will utilize a bright red light which ought to be visible to any potential targets within range of the device. This will further prevent any unintended damages from being caused by the sentry turret device.

Property owners will be able to use this device legally within the limitations of their property, assuming they place the product in an appropriate location. To do this, property owners should consider the range and reach of the device when determining where to place it. When in the 'on and armed' state, the device will use motion sensors to decide if there is a potential target within firing range. If placed in a location where these sensors can detect motion outside of property boundaries, owners may be liable for damages to others. Upon detecting motion, the camera connected to the device will turn on and rotate in the direction of the motion. At this point, the processor will begin using computer vision algorithms to determine whether the motion was caused by a human in the vicinity. Only after the sentry turret recognizes a human form through the execution of these computer vision algorithms, the firing device fitted to the sentry turret will be aimed toward the individual recognized, then it will be fired periodically for the remainder of time during which the target is within range of the camera and projectile launcher. For the purpose of returning the sentry turret to the 'on and disarmed' or 'off' states without injury to operators, the sentry turret will include a remote deactivation method such that any sentry turret device owners/operators can safely deactivate the device. The sentry turret will also include a manual override as a failsafe in case the remote activation/deactivation system is rendered unusable for some reason.

2.4 Requirement Specifications

The following sections on engineering requirement specifications outline the hardware, software, and key specifications that the sentry turret project will satisfy. All specifications detailed within these sections are reasonable specifications in accordance with the design idea and intended purpose. The specifications will be needed to refine the functions of our device to a working prototype whose functionality and usefulness can be quantifiably measured based on predetermined requirements.

2.4.1 Hardware Requirements

1. The sentry turret should be light-weight, roughly 40 pounds or less. This will ensure that an individual can safely lift and relocate the device as desired

2. The device should have a motion sensor to detect motion within a range of 30 feet. This sensor will serve to activate the turret from a low-power state.

3. The turret should have a camera with a high enough resolution such that humanoid figures can be determined from a range of up to 75 feet. This will allow computer vision algorithms to identify targets based on images provided by the camera.

4. The turret's hit indication device should be capable of fully automatic fire, with a magazine/loader size of at least 20 rounds. This means that the device should be able to fire at least 20 rounds during a single activation period without any further interference from an operator.

5. The device should be capable of rotating 180 degrees horizontally. This horizontal rotation will increase the viewing area of the camera and the target area of the hit indication mechanism. Therefore, the range of effect of the sentry turret device will be significantly increased through this requirement.

6. The device should allow for 45-degree vertical movement of the hit indication mechanism. This will allow targets of varying heights at varying distance to be marked by the sentry turret.

7. The gun should have an accuracy of at least 70%, even while the target is moving. This means that at least 70% of shots fired from the sentry turret device should strike their mark.

8. The power supply for the sentry turret should be capable of recharging with a wall plug. This will allow device operators to position the sentry turret near an outlet and ensure that the device will remain active over a period of time for which the power supply could not otherwise deliver sufficient power.

9. The sentry turret should have an internal power system that can power the machine for 3 hours. This requirement allows for relocation of the sentry turret device without reliance upon external power sources for a duration of 3 hours.

10. The device should be built with a warning system to indicate the turret is about to fire (a flashing red light), with a timer of 5 seconds. The warning system on the turret will prevent unnecessary harm/damage to any individual who unintentionally entered the range of effect of the sentry turret system.

11. The development of the sentry turret project should cost no more than $400. This is to keep the system cost-effective throughout the course of the project,

2.4.2 Software Requirements

1. Should use computer vision to identify targets within range of 30 feet

2. Should use motion sensor to activate the camera to check for targets (when sensing movement within 30 feet), in order to save energy

3. Should be programmed to aim at and fire upon identified targets when they are within range of 30 feet

4. Should be programmed to stop shooting when a target leaves its 30-foot range

5. Should be programmed to switch to a low-power mode after a period of 5 minutes without apparent targets, in order to conserve energy

6. Should be programmed to give a 5-second-long warning when a target enters the turret's range

7. Should be programmed to adjust the turret's aim as the target moves to maintain at least 70% accuracy

8. Should be able to identify up to 3 targets at once and prioritize them according to distance to turret and time spent trespassing within the turret's range

2.4.3 Key Specifications

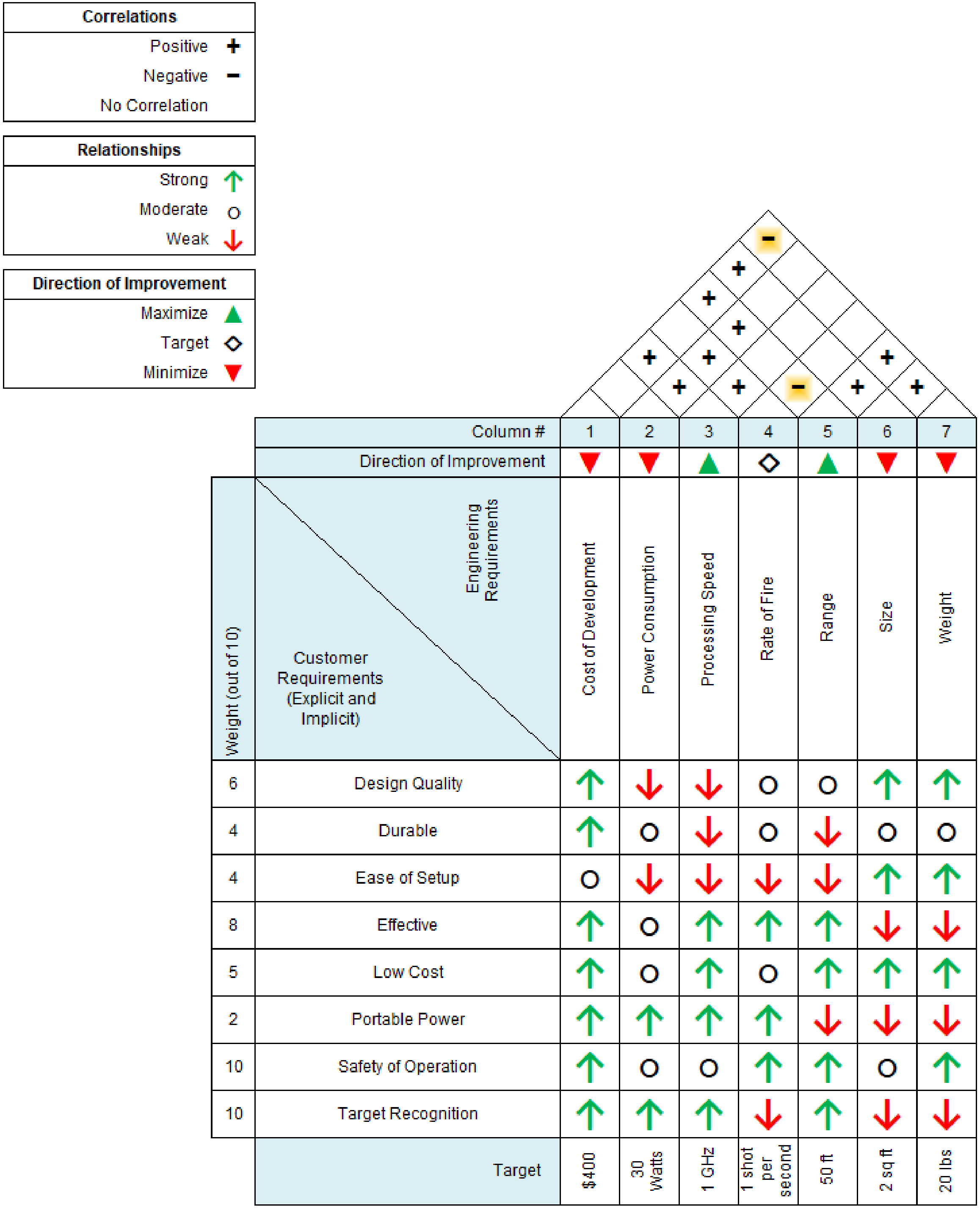
Table 2: Key Specifications (below) lists the pivotal specifications that will be implemented in our prototype. These specifications all pertain to both external and internal features of the prototype. The three highlighted specifications located at the top of the table have been selected as required specifications which will be demonstrated to our panel of those who will assess the sentry turret project and whether it has successfully accomplished its goal(s). For the prototype to be considered successful, it must demonstrate that it is capable of accurately marking targets with 70% efficacy. To locate and identify targets, the prototype must be able to traverse 180 degrees horizontally and 45 degrees vertically. Finally, the range of the sentry turret device must be between 10 and 75 feet. The conjunction of these required specifications signifies that any individuals within a cone that

is 180 degrees wide, 45 degrees tall, and between 10 and 75 feet away from the front of the device when it is in the ‘on and armed’ state will be targeted and marked by paintballs fired from the device with 70% accuracy.

Key Specifications	
Accuracy (minimum)	**70%**
Traverse	**180° horizontally, 45° vertically**
Range	**10-75 feet**
Power Supply Duration	3 hours
Ammunition Capacity (minimum)	20 rounds
Weight (maximum)	40 pounds
Multiple Target Acquisition	Up to 3 separate targets
Warning Time	5 Seconds

Table 2: Key Specifications

2.5 House of Quality



Correlations	
Positive	+
Negative	−
No Correlation	

Relationships	
Strong	↑
Moderate	○
Weak	↓

Direction of Improvement	
Maximize	▲
Target	◇
Minimize	▼

	Column #	1	2	3	4	5	6	7
	Direction of Improvement	▼	▼	▲	◇	▲	▼	▼
Weight (out of 10)	Engineering Requirements Customer Requirements (Explicit and Implicit)	Cost of Development	Power Consumption	Processing Speed	Rate of Fire	Range	Size	Weight
6	Design Quality	↑	↓	↓	○	○	↑	↑
4	Durable	↑	○	↓	○	↓	○	○
4	Ease of Setup	○	↓	↓	↓	↓	↑	↑
8	Effective	↑	○	↑	↑	↑	↓	↓
5	Low Cost	↑	○	↑	○	↑	↑	↑
2	Portable Power	↑	↑	↑	↑	↓	↓	↓
10	Safety of Operation	↑	○	○	↑	↑	○	↑
10	Target Recognition	↑	↑	↑	↓	↑	↓	↓
	Target	$400	30 Watts	1 GHz	1 shot per second	50 ft	2 sq ft	20 lbs

Figure 3: House of Quality

The house of quality presented in **Figure 3** (above) demonstrates the relationships between the engineering requirement specifications determined by the members of Group 33 and the customer requirements expected by any consumer of this product. This model provides a practical view of the importance of different features in the development of the sentry turret, allowing Group 33 to prioritize the appropriate aspects of the project. For instance, as represented by **Figure 3**, the cost of development has a strong relationship with each of the customer requirements; as such, the development team must ensure the effective management of any applicable budget to provide customers with a desirable product. Additionally, the processing speed of the CPU on board the sentry turret has a moderate-to-strong relationship with a majority of the customer requirements, implying that a notable portion of development efforts should be dedicated toward maximizing the processing speed.

The target values located at the bottom of the house of quality in **Figure 3** are the outcomes for each engineering requirement based on the relationships represented in the table. These values take into consideration the direction of improvement for each specification, as well as their impact upon each other, to determine the appropriate target.

Due to its extreme relevance to the project, the target development cost was determined by summing the maximum desired budget of each member in Group 33. While the direction of improvement for this cost is downward, it was recognized that applying the maximum budget of each member would result in the highest customer satisfaction.

For the purposes of this project, power consumption was only a moderate concern. This fact is evidenced by the net impact of relationships between engineering specifications and customer requirements in **Figure 3**. However, the power consumption of the device has a positive correlation with several of the other requirement specifications. Therefore, while the direction of improvement is downward, the target value was set higher than the minimum necessary value for functionality of the product.

As mentioned previously, the processing speed of the CPU used for the sentry turret maintains a net-strong relationship with the customer requirements. As a result, the target processing speed is several gigahertz higher than the minimum necessary value.

The rate of fire for the sentry turret has no direction of improvement due to the fact that neither a high nor low rate of fire is desirable for the device. A high rate of fire could result in reduced accuracy and/or excessive force, while a low rate of fire (lower than the target) would deem the product ineffective. These factors are represented through the customer requirements in the house of quality above, with their relationships to this engineering requirement displayed. The target rate of fire was decided deliberately, rather than limited by hardware/software capabilities.

The range of the device has a negative correlation with the rate of fire, meaning that a larger range would require a lower rate of fire, and vice versa. Because the rate of fire was decidedly placed at a relatively low value (compared to the maximum fire rate of the projectile launcher used to create the sentry turret), the target range of the sentry turret could be placed at a higher value.

The size of the turret maintains a net-moderate relationship with the customer requirements, and the direction of improvement for this engineering requirement is downward. With these details taken into consideration, the target size of the product was based on the necessary amount of space required to fit the components together in a functional manner, without taking up excess space.

The weight of the turret has a negative correlation with the cost of development due to the fact that sturdy light-weight materials tend to be more expensive than either sturdy, heavy materials or non-sturdy, light-weight materials. With that in mind, the target weight value was determined partially based on the target cost of development.

2.6 Block Diagram

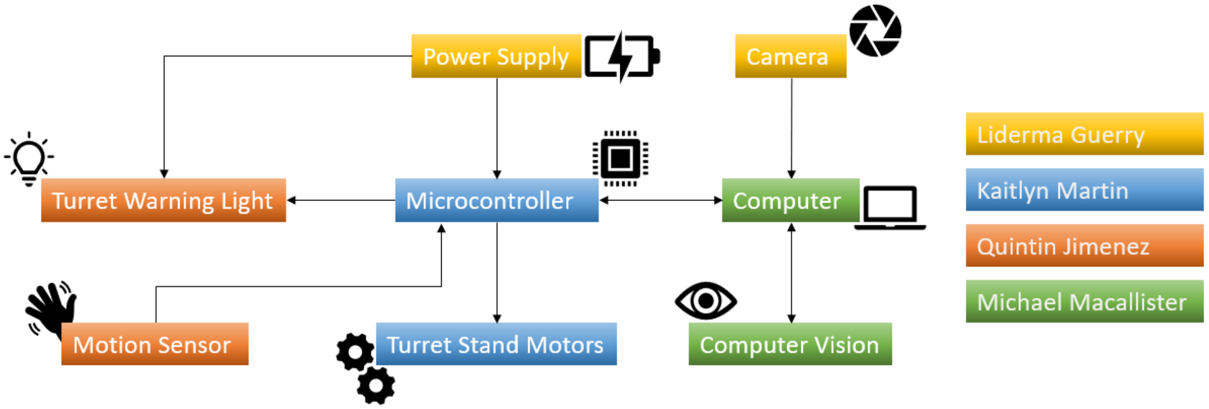


Figure 4: Block Diagram and Responsibilities

The block diagram in **Figure 4** (above) shows the responsibilities of each member of Group 33 in developing the sentry turret project. Members Liderma Guerry and Kaitlyn Martin are responsible for hardware-related aspects of the project, such as circuitry and part selection, while members Quintin Jimenez and Michael Macallister are responsible for software development and integration with the hardware.

In **Figure 4**, hardware responsibilities are indicated by red fields, while software responsibilities are indicated by blue fields. While hardware and software are very closely related, the fields were defined based on relevance to each position. For example, the software team will utilize the selected power source to run their code, but the hardware team must ensure that the power source is capable of providing sufficient power to the processor and peripherals in order for that code to run.

One notable aspect of the block diagram in **Figure 4** is that the camera has been marked by a blue field, indicating that it is the responsibility of the software team. This is because computer vision algorithms can be quite complex and require a suitable camera to provide images on which the algorithms can be performed. As such, the software team will rely more heavily on the camera used than the hardware team, who will simply ensure that the camera itself functions when connected as part of the sentry turret.

2.7 Group Breakdown and Responsibilities

The members of Group 33 have divided themselves into two teams to conquer different aspects of the project. One team, composed of members Quintin Jimenez and Michael Macallister will be primarily responsible for areas of the project more directly related to software. The other team, composed of members Kaitlyn Martin and Liderma Guerry will be primarily responsible for areas of the project more directly related to hardware. This division of tasks by software and hardware will act to ensure that two members of Group 33 will be able to share and discuss information in such a way that will allow others to compensate for absences in the case that emergencies arise. This does not detract from any personal duties on the project, as there will be much work to accomplish. The division of labor based on individual requirements is detailed in **Figure 4: Block Diagram and Responsibilities** (above). As shown, Liderma Guerry will manage the power supply and camera integration for the sentry turret project, Kaitlyn Martin will manage the microcontroller and turret stand motors, Quintin Jimenez will manage the details relating to the turret warning light and motion detection system, and Michael Macallister will manage computer integration and computer vision applications.

Quintin J. – Software Lead (Make sure this person agrees with software decisions)

Kaitlyn M. – Hardware Lead (Make sure this person agrees to hardware decisions)

Michael M. – Computer Vision Specifics, Assist with Hardware/Software

Liderma G. – Assist with Hardware/Software

- All group members will assist with decisions regarding the building of the project; this list is not a final and absolute division of responsibilities.

3.0 Technology

These sections will explore different possibilities for technology used in our prototype. For the type of software technologies, we consider both computer vision and machine learning. The most popular framework for computer vision will be OpenCV which is a free resource that is continually updated with computer vision algorithms. For machine learning we will go into the TensorFlow framework, this is the OpenCV equivalent for machine learning. Our prototype must be able to detect a humanoid target so either OpenCV or TensorFlow could be good options to use for software.

For hardware technologies there are various options to explore. In our project we will employ the use of camera technology, motion detection, motors, microcontrollers and microprocessors; each of these can be implemented in a variety of ways and will therefore require much deliberation and careful consideration to determine the best technology to use when implementing each aspect.

3.1 Software Technology

There are many software programs and programming languages at our disposal to use in conjunction with computer vision. The algorithms needed could be written in a multitude of programming languages, ranging from functional programming languages to object orient programming languages. Some languages being considered for use in this project include C, C++, Java, and Python. There are still many other languages which can be applied to this project, though these are some of the most popular resources available. Using a popular programming language would benefit the members of Group 33, as it would increase their likelihood of finding help in online forums should they encounter any issues throughout the course of the project.

Another popular software resource under consideration by the members of Group 33 is OpenCV. OpenCV is a useful source provided by the Intel Corporation, and it provides libraries and algorithms for the implementation of computer vision into user projects.

3.1.1 Computer Vision

Computer vision has had a significant impact on the world today and continues to make positive changes to businesses, society, and individuals’ quality of life. Computer vision technology is extremely innovative and utilizes statistics to allow computers to make observations and draw conclusions through the analysis of images. It is programmed in such a manner that it is able to quickly learn about and perceive the modern world. By means of statistical applications of mathematics and logical algorithms, computer vision can deconstruct and then reconstruct images and visual objects into meaningful data. Our group seeks to implement computer vision technology to allow the sentry turret project to detect process images and determine whether a valid target is within firing range of the device. With a well written library and source of algorithms like OpenCV, humanoids objects/forms can be distinguished from other objects within an image. The recognition of humanoid objects/forms is necessary for the purposes of the sentry turret project, as the device is intended to target humans only. Furthermore, as the sentry turret must function automatically and without the continued intervention of a human at each step of the way, it is important for the computer vision software implemented to be capable of recognizing humans with a large degree of certainty.

The function of computer vision algorithms as they will be applied in this project is to observe the objects within the frames parsed from the camera to the computational device, then to discerning and distinguish any humanoid forms contained among those objects. The algorithms will then react to the objects within range of the camera by returning values about the presence and positions of any possible targets observed. Computer vision uses data obtained through detecting the edges, corners, image intensities, and matched templates to perceive and identify objects from an image. The camera used in conjunction with a decently powerful processor will be able to gather data and track targets efficiently.

3.1.2 Open CV

The OpenCV (Open-Source Computer Vision) Library is a library of programming functions mainly aimed at real-time computer vision implementations. OpenCV will be utilized to enable the turret to recognize targets through its camera. OpenCV ships with a pre-trained Histogram of Oriented Gradients (HOG) and Linear Support Vector Machine (SVM) model that can be used to perform pedestrian detection in both images and video streams. These tools provided by OpenCV can be repurposed for usage with the processor for the sentry turret project being constructed by the members of Group 33. OpenCV is a free resource available to use for computer vision. Another important detail about OpenCV is that it is routinely updated. This means that the tools, libraries, and software included will be up-to-date at all times to keep up with hardware improvements and the constantly changing technology available in the world.

3.1.2.1 Histogram of Oriented Gradients

A Histogram of Oriented Gradients (HOG) is a feature descriptor—an algorithm which takes an image and outputs feature vectors. Feature vectors indicate the position and curvature of different objects and object parts recognized in an image. Feature descriptors encode interesting information into a series of numbers and act as a numerical “fingerprint” that can be used to differentiate one feature from another. Simply put, HOG extracts and describes the features and details found in an image. Some important aspects of HOG that make it different from other feature descriptors include:

- The HOG descriptor focuses on the structure or the shape of an object. HOG is also able to provide both edge features and edge direction, by extracting the gradient and orientation of the edges.

- These orientations are calculated in localized portions, meaning the complete image is broken down into smaller regions, and for each region the gradients and orientation are calculated.

- The HOG generates a histogram for each of these regions separately. The histograms are created using the gradients and orientations of the pixel values, hence the name “Histogram of Oriented Gradients”.

First, the subject image is resized to bring the width to height ratio to 1:2, preferably to 64x128. This is because the algorithm divides the image into 8x8 and 16x16 patches to extract the features. Therefore, the 64x128 dimensions make these calculations much easier. **Figure 5** gives a basic example how HOG divides and processes images.

Next the gradients are calculated for every pixel in the image. The gradient is obtained by combining magnitude and angle from the image. Considering a block of 3x3 pixels, first Gx and Gy is calculated for each pixel, using the formula below for each pixel value.

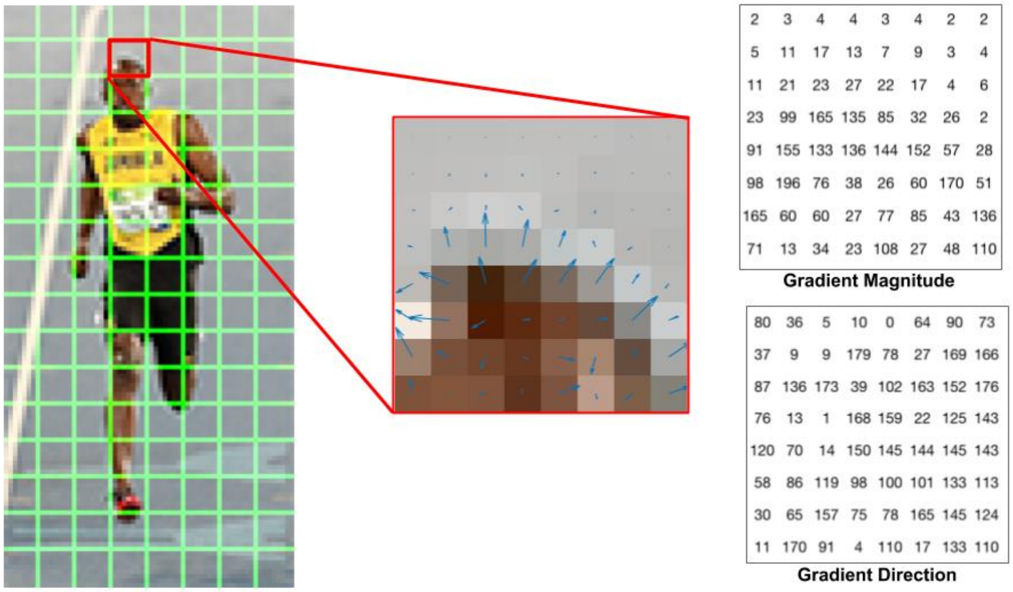
$\mathrm{Gx}\ (\mathrm{r}, \mathrm{c}) = I\ (\mathrm{r}, \mathrm{c}+1) - I\ (\mathrm{r}, \mathrm{c}-1)$ $\mathrm{Gy}\ (\mathrm{r}, \mathrm{c}) = I\ (\mathrm{r}-1, \mathrm{c}) - I\ (\mathrm{r}+1, \mathrm{c})$

Where r, c refers to rows and columns respectively

After calculated Gx and Gy, magnitude and angle of each pixel is calculated using the formula below.

Magnitude(μ) = sqrt(Gx^2 + Gy^2)

Angle(theta) = |tan^-1(Gy/Gx)|



2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

Figure 5: The 8x8 RBG patch represented using arrows and as numbers

Credit: learnopencv.com

After obtaining the gradient of each pixel, the gradient matrices (magnitude and angle matrix) are divided into 8x8 cells to form a block. For each block, a histogram is calculated, split into 9 separate bins, with each bin corresponding to angles from 0 to 180 in increments of 20. A bin is selected depending on the pixel's angle, and the value that is subsequently placed inside that bin is dependent on the pixel's magnitude. If a pixel's angle is halfway between two bins, then it splits up the magnitudes accordingly depending on its distance away from each respective bin. These histograms are then concatenated in groups of four into a 36-feature vector. This vector is then normalized by the L2 norm in order to reduce the effect changes in contrast between images of the same object (e.g., changes in lighting). This feature vector is then fed into the Linear SVM which classifies the target human bodies according to the feature sets.

3.1.2.2 Linear Support Vector Machine

The algorithm widely used for learning how to recognize pedestrians given HOG feature is called a Support Vector Machine. Such a scheme models an object (in this case the pedestrian) with respect to a set of parameters, which usually undergo an optimization step, which is the essential "learning-part" of the proposed scheme.

Given the HOG features "x" of a window in an image, the SVM assigns a score, which determines how certain the algorithm is that this object is a pedestrian or not. More formally to classify a window with a feature vector x, an SVM computes the following function:

h(x) = w^t * x + b

where w is the weight vector and b is bias.

These two parameters give rise to a hyperplane in feature space, which separates windows containing pedestrians from background windows in a sliding-window-fashion. A binary decision is commonly achieved by using the sign function on the output of the SVM: g(x) = sin(h(x)) = sin(w^t * x + b). This hyperplane is depicted in **Figure 6** below.

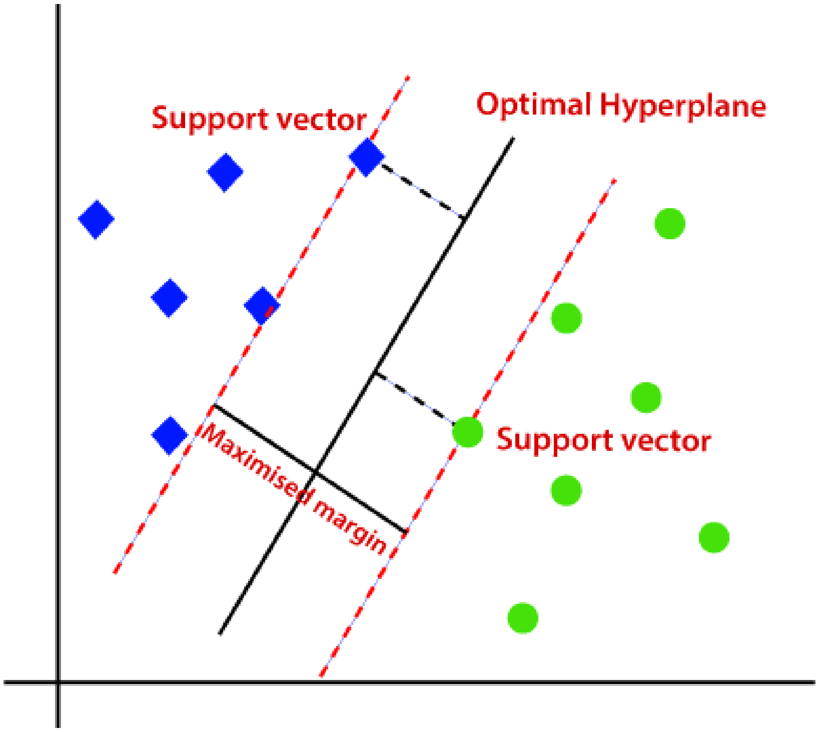


Figure 6: Linear SVM Classifier in 2D space

OpenCV comes with a pre-trained SVM, so we don't have to worry about training it ourselves. We can simply use the SVM to determine if the feature vector given by the HOG indicates a pedestrian or not. Then, if we have detected a pedestrian, we can simply draw a bounding box around them and use this bounding box for our turret aiming algorithm.

3.1.3 Machine Learning

Machine learning has greatly evolved throughout history as we now see many examples in everyday use such as Netflix suggesting shows to watch based on viewing history or fraud detection where algorithms look at common uses of how you spend money and alerts findings accordingly. This method is entwined with artificial intelligence and uses machine learning algorithms that recognizes patterns. It has the idea that computers do not have to be specifically programmed for a task, but rather learn from data and take actions accordingly.

3.1.4 TensorFlow

TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that let researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

TensorFlow Lite is TensorFlow's solution for lightweight and mobile applications (in this particular case, usage on a microcontroller). TensorFlow Lite takes a pre-existing trained model (usually gotten by training a set of data on a high-performance machine) and converts it to a special format that can be optimized for speed or storage. If we were to use TensorFlow for our project, using TensorFlow Lite would be a necessity.

3.1.4.1 Keras

Keras is a high-level API (Application Programming Interface) that TensorFlow uses. It runs with TensorFlow and allows for fast development and evaluation of deep learning models. It is very powerful because of its simplicity. It reduces cognitive load which allows the developer to focus on other problems. It implements the Python coding language only uses a minimal amount of code to train neural networks.

3.1.4.2 Generative Adversarial Networks (GANs)

Generative Adversarial Networks or GANs is a model in machine learning where two neural networks are trained to distinguish features against each other. This helps train the models to tell real images apart from fakes. The models are called the generator and discriminator. The generator's job is to create images that attempt to look real while the generator has to differentiate between what is real and fake image.

The generator and discriminator go through many attempts and each progressively gets better. The generator will start to make images that start to look more realistic as each attempt goes by, and the discriminator will get trained better to tell what is fake. The whole process will stop when the system reaches an equilibrium. The discriminator will no longer be able to tell what is real or fake. This process is demonstrated in **Figure 7** where we see the attempts of both generator and discriminator progress.

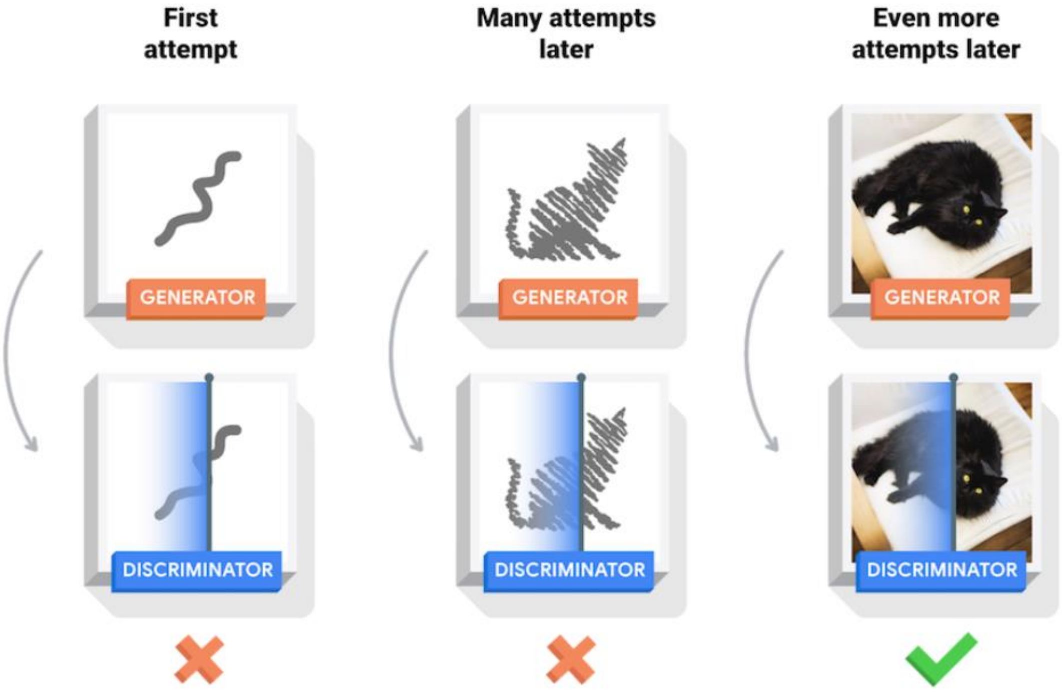


Figure 7: GANs process

3.1.5 Computer Vision versus Machine Learning

Comparing OpenCV to TensorFlow, the latter is a framework for machine learning, while OpenCV is a library for computer vision. If we were building a new deep learning model for a specific task, then TensorFlow would be preferable. In this case, however, we are using a pre-trained open-source computer vision model – there is no requirement to train a model ourselves. Not only that, but OpenCV generally has better performance on microcontrollers than TensorFlow – a crucial requirement for this project. As such, OpenCV suits our project much better than TensorFlow. Below we can see a **Table 3** for the comparison of technologies.

OpenCV	**TensorFlow**
Computer Vision	Machine Learning
Image processing and detection	Pattern Detection
Uses C++, Python, Java and MATLAB	Uses C, C++, Java, and Python
Operates on Windows, Linux, Android and Mac OS.	Operates on Windows, Ubuntu, macOS, and Python 3.7-3.9.
Efficiency of real-time applications	Mathematical solutions using dataflow charts

Table 3: Technology comparison of OpenCV and Tensor Flow

Both OpenCV and TensorFlow frameworks have algorithms that are specific to computer vison and machine learning respectively. They both uses coding languages that our group members are familiar with and have online support via article and libraries. Since our project relies more on image processing rather than deep learning for AI, OpenCV is better to use.

3.1.6 Python

Python is one of the most popular high-level programming languages and there is a multitude of support articles and documentation that can be found online for projects utilizing this language. It is interpreted, high-level, general purpose, and object-oriented, which allows it to be used in modern processes such as computer vision and machine learning. It reads lines of code one by one and performs the actions read. It does not require you to compile the program before executing, but you will run a .py file and there is automatic compilation. Its structures are easily read and translated for the CPU. It is also portable and can be used on different computers with almost no modifications.

Python is an option for our prototype specifically for the computer vision portion. It is one of the languages used by OpenCV and will relatively be an easier coding experience. The ability to be portable and simple to learn will help with efficiency and time management. It will also be a good experience as Python is widely used.

3.1.7 C++

C++, like Python, is also an object-oriented programming language. It offers portability, total control over memory management, community support, compatibility with the C language, and scalability. Also like Python, C++ grants programmers the ability to run programs on different computers, which is a highly valuable feature which will be needed if our program must be moved to another computer. Total memory management is beneficial from a hardware perspective, as it ensures that the hardware used is being applied to its fullest potential. This total control, however, is problematic from a software perspective due to the fact that it complicates code and requires an in-depth understanding of pointers and addressing to avoid errors when developing a program. C++ manages memory through Dynamic Memory Allocation (DMA) with the use of pointers and addresses.

This is the second programming language for our prototype. It is robust and will fulfil the coding requirements for our prototype. It does provide may benefits, but it is also harder to implement than Python requiring the need for managing memory. However, C++ is a powerful language as it has a diverse function library, but the code tends to be larger and predefined syntaxes and structures.

3.1.8 Python versus C++ for OpenCV

Both C++ and Python are superior options for using OpenCV. The differences between them largely lie with the learning curve to implement code. Both have a large community to help without code implementation and provide robust solutions to computer vision problems. **Table 4** will outline the benefits of each language, and what parts of our project we will implement them with.

C++ (will be used for microcontroller)	Python (OpenCV)
Object-Oriented	Object-Oriented
Harder to use	Easier to use
Has predefined syntaxes and structure	Syntax is easier to remember
Dynamically typed	Statically Typed
Pre-compiled	Uses Interpreter
Faster speed	Slower Speed
More lines of code	Less lines of code
Manages memory through pointers	Uses a garbage collector to manage memory

Table 4: Comparison of C++ and Python

For our main OpenCV project, we will use Python. It will be easier to write our programs and start testing fairly quickly. OpenCV also has a library of Python bindings available for computer vision. This will be important as we will have to use two different coding languages in our project. Our microcontroller on the other hand will use C++ to be programmed. Both languages are portable, which is an added bonus if we need to switch computers at any time. Our program with the microcontroller will be fairly simple to engage motors and the warning on our system leaving the more complicated part to communication with the code on the laptop.

3.2 Hardware Technology

For hardware that will be used for our prototype, there are many considerations that must be made to support the optimal use of computer vision. A microcontroller with good processing power for software must be employed for computer vision to run smoothly. In addition, a compatible camera must be chosen to track targets efficiently. Our motors must be able to complete the required steps to pull the trigger on our paintball gun. There will be heavy scrutinization of what will be chosen for our embedded component. The two biggest contenders on the market are Raspberry Pi and Arduino. Raspberry Pi is a microprocessor and has superior capabilities when dealing with projects highly catered to software. Arduino on the other hand is chosen for more hardware intense projects. Our project is a split between hardware and software so either could be selected.

Our next major selection would be motors. There are three types to choose from, DC motor, Servo motor, and Stepper motor, which will do the job, but we will consider which one will be the best from a price and performance point of view. Even though motion sensors are a secondary goal we will go into their technologies are see which one could be a prospective fit. Technologies for hit indication device will be analyzed to see what method is best to use for marking targets. From there most of the hardware and specifications will trickle down from the needs of our specific components.

3.2.1 Motors

Stepper motors have become extremely common in the field of robotics due to their simplicity and affordability. Stepper motors are Direct Current motors that move in discrete steps, rotating one step at a time. The rotation is a result of the magnetic current induced by an electrical current generated inside the motor. Stepper motors have multiple coils organized in phases. The steps for motion are computer-controlled. Through the steps, these motors can achieve precise positioning, speed control, and low-speed torque with high precision. **Figure 8** below depicts a stepper motor and its interior design.

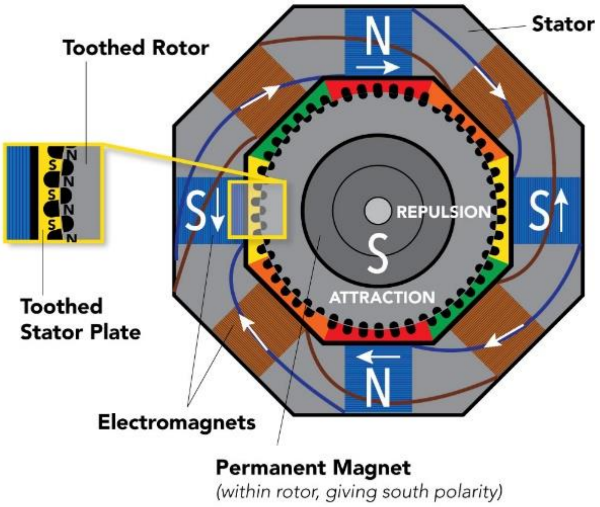


Figure 8: Stepper Motor Diagram

https://clippard.com/cms/wiki/how-stepper-motors-provide-precision-control

The motors have 48 outer and inner teeth. To magnetize the inner teeth, the motors contain 8 separated coils. Four steps will be created by the magnetic field that results from the electrical current. Multiplying the teeth (48) by the steps (4), the motors get 192 steps per rotation, equivalent to 1.8° per step. This type of stepper motor is most common, known for the full steps (200 steps per rotation).

There are two types of stepper motors: unipolar and bipolar. For each type, different circuits are necessary. Unipolar steppers work with one winding and center tap per phase. Bipolar steppers have only one winding per phase. Bipolar steppers are more efficient than unipolar steppers, because unipolar steppers use more coil wire, which then causes greater resistance in the motor and thus higher power consumption. Bipolar steppers also produce greater torque than unipolar steppers. However, unipolar steppers are cheaper, simpler to use, and require a simpler circuit than bipolar steppers.

Stepper motors also have several characteristics that determine their performance and suitability to specific tasks. Motor size is defined by NEMA number, which defines standard faceplate dimensions for mounting the motor. Generally, larger motors provide greater torque than smaller motors.

Step count determines the positioning resolution of the motor. The number of steps per revolution ranges from 4 to 400. Commonly available step counts are 24, 48, and 200. A higher resolution means the motor can be more precise in its movements. The tradeoff for a higher resolution is lower speed and torque.

Gearing is another way to get high positioning resolution. A 32:1 gear-train applied to the output of an 8-steps/revolution motor will result in a 256-step motor. A gear train will also increase the torque of the motor. The tradeoff is speed. Geared stepper motors are generally limited to low RPM applications. Backlash is another issue with geared motors. When the motor reverses direction, it needs to take up any slack there may be in the gear train, which can affect positioning accuracy.

Stepper motors do require a driver chip in order to operate them. These drivers offer low level interfaces like inputs for directly initiating each step/movement. An external microcontroller is typically required for generating these low-level signals.

DC motors are electromagnetic devices that use the interaction of magnetic fields and conductors to convert electrical energy to mechanical energy for rotation. There are many types of DC motors out in the market. The brushed and brushless motors are the most common DC motors. A diagram detailing a DC motor and its interior is pictured in **Figure 9** below.

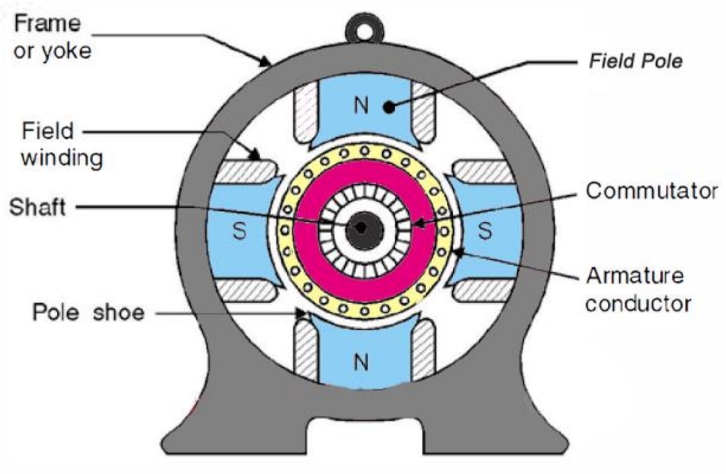


Figure 9: DC motor diagram

https://www.technicalbookspdf.com/construction-of-dc-machines-motor-generator-with-diagrams/

A DC motor consists of coils connected to segments of a ring, or commutator. The coils are surrounded by a pair of magnets, or a stator, that envelopes the coils in an electric field. When current is passed through a wire in a magnetic field, the wire experiences a force, and so the coils in the motor experience a force that pushes the coil and begins the rotation.

DC motors are as simple to control as a switch – you need only to apply a voltage to start driving them. DC motors slow down when voltage is lowered, and spin in the opposite direction when voltage is reversed.

Our group decided to use stepper motors for reasons outlined in the parts selection section.

3.2.2 Stepper Motor Driver

A stepper motor driver is an actuator which can transform pulse signals into angular displacement signal. Stepper drivers drive stepper motors to rotate at an angle called a step angle in a set direction when receiving a pulse signal. An external microcontroller is typically required for generating these low-level signals. The motor speed is up to the pulse frequency given from the controller, and the displacement is decided on the pulse quantity given from the controller. The stepper system consists of a stepper motor, a stepper driver, and a microcontroller. Performance of a stepper system is not only up to the motor, but also depends on the stepper driver itself.

A stepper motor drive is chosen based on several characteristics: Minimum operating voltage, maximum operating voltage, maximum continuous current per phase, peak current per phase, micro stepping (allows greater resolution in a stepper motor's steps, thus providing a higher resolution, at the cost of decreased torque per micro step), and other special features such as Auto Gain Control. One generic, commonly used stepper motor driver is the DRV8824, and is pictured below in **Figure 10**. Most stepper motor drivers have a similar design to the DRV8824.

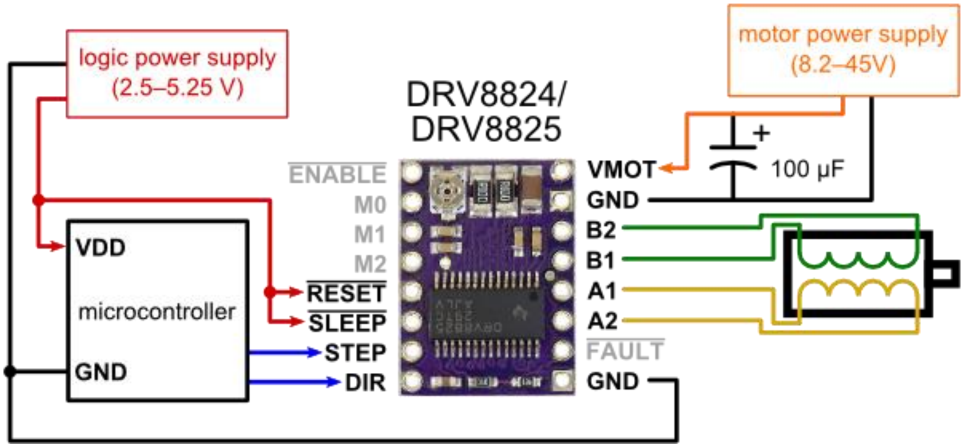


Figure 10: Circuit diagram of a standard driver, the DRV8824

https://www.pololu.com/product/2131

There are two primary types of drives for stepper motors: constant voltage drive (also referred to as L/R drives) and constant current drives (also referred to as chopper drives). One difficulty with stepper motor operation is that time the time constant (L/R) of the motor windings prevents current from increasing rapidly during pulses. This means that unless the voltage is very high, the current can never reach its full rated value, especially when the pulse rate is high (I.e., at high motor speeds). This limitation is governed by two equations:

Ohm’s Law: $I = V/R$

Current rise & Induction Relation: $dI/dt = V/L$

In order to get high current, and therefore high torque, at high speeds, the voltage must be kept as high as possible and the inductance as low as possible. But in traditional L/R drives, the voltage must be kept low in order to keep the steady-state current from becoming excessive.

A chopper drive addresses the problem of obtaining high torque at high speed from a stepper motor by turning the output voltage to the motor on and off rapidy (“chopping”) to control the motor current. At each step of the motor, a very high voltage (typically eight times higher than the motor’s nominal voltage) is applied to the motor windings. This causes the current to rise rapidly, according to the relationship between current rise and inductance. It also allows higher current to be produced, according to Ohm’s Law.

3.2.3 Motion Sensor

There are several types of motion sensors on the market, each with different modes of activation. These can be separated into active and passive sensors. Active motion sensors rely on radio or microwave waves. These waves are sent out and hit targets which come back at a certain frequency. When the waves hit a moving target the frequency changes. The sensor then picks up on the change which in turn triggers the device attached for a warning signal, such as an alarm. Passive motion detection on the other hand does not use waves but detects infrared (heat) levels. The sensitivity of these sensors can be adjusted to a desired heat level so they will not activate for unwanted targets.

There are also hybrid sensors that include both active and passive detection. This type of sensor is used to lower the possibility of false alarms. However, this type of sensor needs to trigger both types of sensors to activate, so it will not show a detection if only one of the sensors is triggered. For our prototype only active and passive motions sensors will be considered.

Microwave motion sensors are active sensors that can detect various types of motion by using electromagnetic radiation. These sensors send out continuous radiation waves at a certain frequency, and when those waves encounter a moving object, the frequency is shifted. Although these sensors use electromagnetic radiation, they are designed to be at a safe level. These motion sensors are comprised of a transmitter that sends out the waves, the receiver that receives incoming reflected waves, and a device that sets off an alarm of some sort when an altered frequency is obtained by the receiver. This is also known as the Doppler Effect where a change of frequency is detected by the receiver. This is the same mechanism that is used in radar guns which law enforcement use to track speeders. **Figure 11** shows how the waves behave to detect motion. These motions sensors are also called Doppler motion sensors because of the named effect. These sensors cover a large area compared to other sensors due to the waves' ability to penetrate through walls, but this is also a downfall as it can generate false alarms. Most microwave motion sensors that detect under 15 feet are generally affordable with some being as low as $4.

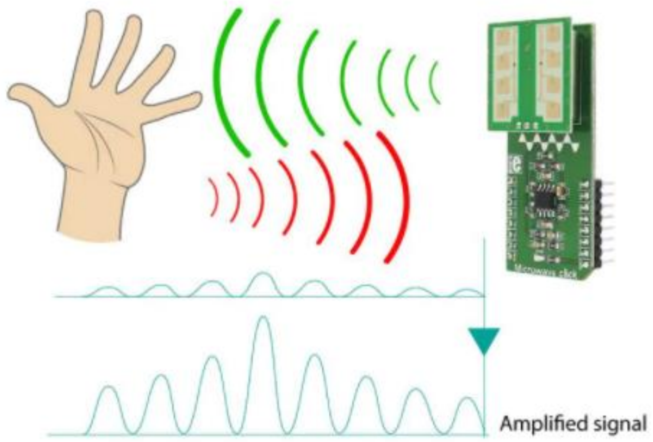


Figure 11: The detection of Microwave sensors

Another type of active motion sensor is the Ultrasonic motion sensor. This sensor similarly to the Microwave sensor, but it sends out sounds waves at a high frequency. These frequencies are generally higher than a human can hear. This sensor sends out these waves and they hit objects and return. If any interruption occurs when these waves return it will trigger the attached alarm. Most often this sensor is used to detect the distance between itself and an object.

Figure 12 demonstrates the action of an Ultrasonic motion sensor. For long range Ultrasonic motions sensors that could be used for detection rather than range finding, it is a bit expensive being greater than $20. For smaller applications and ranges under 200mm you can find these sensors for around $2.

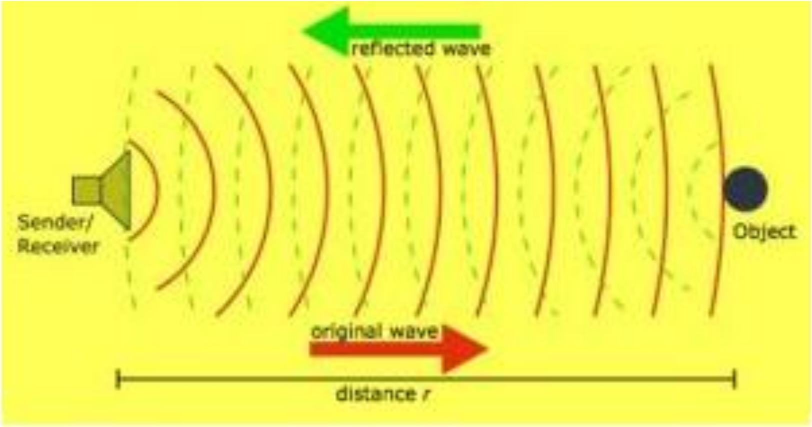


Figure 12: The detection of Ultrasonic sensors

A Passive Infrared Sensor (PIR) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. The sensor itself is actually split into two halves. When the sensor is idle, both sides detect the same amount of IR – the ambient amount radiated from the environment (be it a room’s walls or the outdoors). When a warm body like a human or animal passes by, it first intercepts one half of the PIR sensor, which causes a positive differential change between the two halves. When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. These change pulses are what is detected and registered as movement. **Figure 13** shows the detection mechanism.

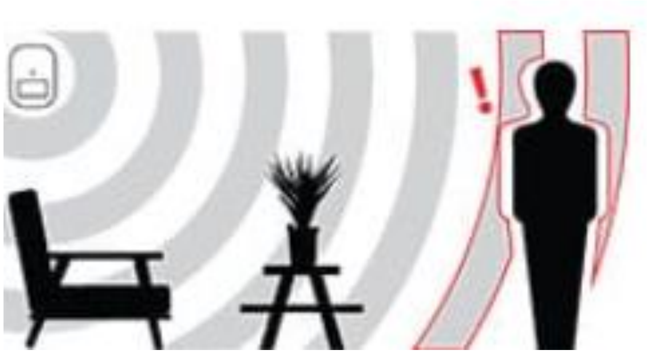


Figure 13: The detection of PIR sensors

The IR sensor itself is housed in a hermetically sealed metal container to improve noise, temperature, and humidity immunity. There is a window made of IR-transmissive material (typically silicon) that protects the sensing element. Behind the window are the two balanced sensors.

PIR sensors are rather generic, and mostly vary only in price and sensitivity. The PIR sensor and circuitry is fixed and costs a few dollars. The lens costs a few cents and can change the breadth, range, and sensing pattern very easily. The sensors generally use Fresnel lenses, which condenses light to provide a larger range of IR to the sensor. The sensor lens is also split up into multiple sections, each a small Fresnel lens. The different faceting and sub-lenses create a range of detection areas interleaved with each other, as shown in **Figure 14** below.

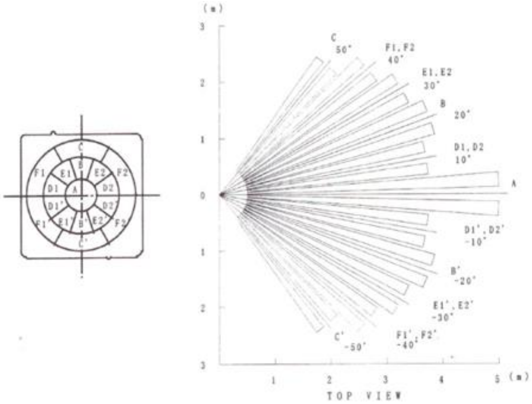


Figure 14: PIR range of detection

https://cdn-learn.adafruit.com/assets/assets/000/010/135/original/NL11NH.pdf

PIRs are small, inexpensive, low power, easy to use, and do not wear out. However, they often have limited range (usually to ~20 feet) and are very vague in the information they provide. PIR’s can detect if there is movement present but cannot discriminate exactly how many objects are moving or how close they are to the sensor. We can use the PIR sensor to activate the turret, but we will need a camera to aim it properly.

4.2.4 Motion Sensor Technology Comparison

In the previous section the benefits and downsides of each motion sensor type were explained in detail. In **Table 5** we give a comparison of these pros and cons and make our final selection.

Microwave Sensor	**Ultrasonic Sensor**	**PIR Sensor**
Not affected by temperature	Sensitive to variations in temperature	Affected by temperature
Continuous power draw	Consumes less energy	Consumes less energy
Wide detection range	Wide detection range	Smaller detection range
Cost effective	Expensive	Cost effective
Works in intervals, might miss	Not good at defining edges of an area	Reliable
Prone to false alarms	Soft material may absorb waves	May have false alarms.

Table 5: Pros and Cons of Motion Sensors

3.2.4 Camera

A camera with a high resolution and frames per second is needed to utilize OpenCV's object detection effectively. In most computer vision applications 24-30 frames per second is sufficient for capturing targets in motion. It must be compatible with our chosen microcontroller and small enough to attach somewhere on the prototype. The camera's function will record digital images for the OpenCV algorithm to detect targets. This will be an especially important component of our project, and we will budget extra to have one that performs well. The better the resolution of our camera will increase the chances of targets being correctly detected.

This camera must also be compatible with our chosen microcontroller or microprocessor. This must be carefully considered, and it is one of the most important parts of our project. It receives images for processing that when processed a target is determined. This will start a chain of events in the code and cause our turret to fire upon a target.

From searching online there are many articles that are helpful in listing compatible options for Raspberry pi and Arduino. There are camera modules and webcams that can be considered for use with both. While camera modules are cheaper, they do not provide as high a quality of resolutions as a webcam. While webcams are more expensive, they offer superior resolution and frames. The following **Table 6** will outline a comparison between camera modules and webcams.

Camera module	Webcam
Soldered on and interfaced	Connected via USB
Lower resolution	Better quality resolution
Lower cost	Higher cost
Lower Megapixels	Higher Megapixels

Table 6 comparison of camera technology

3.2.5 SBC/Development Board

Originally, we had planned for a single board computer (SBC) to run computer vision programming to acquire targets using the attached camera, as well as coordinate the stepper motors and gun. The SBC would have to receive the footage from the camera, (accurately) detect any human silhouettes, calculate the location of the center-of-mass in relation to the turret's barrel, derive the necessary motor movements to aim at this target, then drive these stepper motors until the gun is aimed at the target point, and then fire the gun. The board also has to account for range, erratic movement, and multiple targets (and therefore the process of prioritizing targets based on their distance).
We eventually concluded that the SBCs capable of these processes were far too expensive for our budget. The SBCs within our budgetary range would not have the processing power to maintain accuracy. The program would lag, and thus the turret would find itself consistently missing moving targets.

Therefore, we decided to replace the SBC with a group member's laptop, which can handle the computer vision processes. This laptop will be directly connected to the USB camera, as well as to a microcontroller that will coordinate the turret's motors in accordance with input from the laptop. The lower requirements of the SBC allow us to use a cheaper development board instead.

A microprocessor is the brain of all computing systems – the unit responsible for all necessary calculations which allow a system to work and produce the expected output. A microprocessor cannot work alone because it needs to receive data from other units (such as registers, memory units, and input/output ports).

A microcontroller is an embedded system, meaning it embeds several units into one chip. It includes a microprocessor along with important units like memory and I/O. However, the microcontroller cannot work alone either. It requires a power supply and an interface to load and flash programming onto it.

A development board is basically a microcontroller (or microprocessor) with a USB pot, HDMI port, power input port, and display unit (such as LEDs or an LCD screen). Development boards cut down on time spent carefully selecting and assembling the individual parts so that the user can get to developing and testing their projects faster.

However, since development boards are sold as a whole unit, the developer must be sure that the board they use meets all of their project's requirements.

Single board computers (SBCs) are basically development boards that run an operating system. They are capable of running programs far too complex for development boards, such as machine learning or image processing algorithms. In comparison, development boards and microcontrollers only run a single program iteratively, reading inputs and reacting to them according to their programming.

3.2.6 Servo Motor

Servo motors are part of a closed-loop system and are comprised of several parts namely a control circuit, servo motor, shaft, potentiometer, drive gears, amplifier and either an encoder or resolver. The motor is controlled with an electric signal, either analog or digital, which determines the amount of movement which represents the final command position for the shaft. A type of encoder serves as a sensor providing speed and position feedback. This circuitry is built right inside the motor housing which is usually fitted with a gear system. A diagram of standard servo motor is pictured below in **Figure 15**.

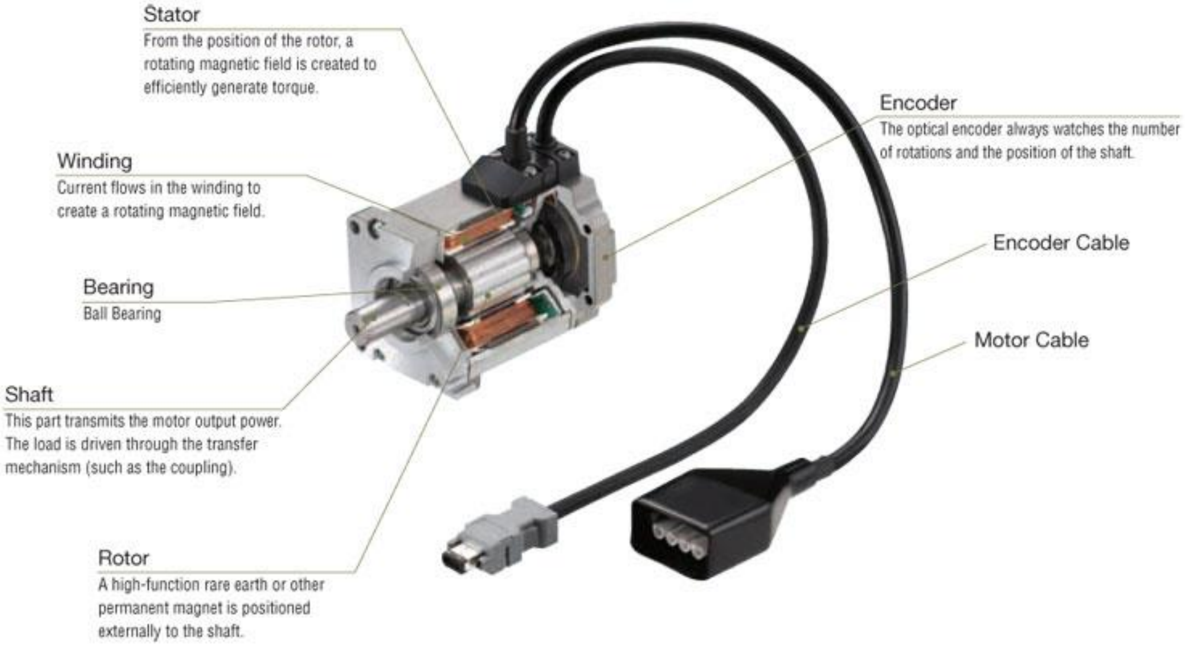

Stator
From the position of the rotor, a rotating magnetic field is created to efficiently generate torque.
Encoder
The optical encoder always watches the number of rotations and the position of the shaft.
Winding
Current flows in the winding to create a rotating magnetic field.
Encoder Cable
Bearing
Ball Bearing
Motor Cable
Shaft
This part transmits the motor output power. The load is driven through the transfer mechanism (such as the coupling).
Rotor
A high-function rare earth or other permanent magnet is positioned externally to the shaft.

Figure 15: Structure of a standard servo motor

Servo motors are classified into different types based on their application, such as the AC servo motor and DC servo motor. There are three main considerations to evaluate servo motors. First based on their current type (AC or DC), secondly on the type of commutation used (whether the motor uses brushes), and thirdly on the rotor (whether the rotation is synchronous or asynchronous).

The primary difference between AC and DC motors is in the inherit ability to control speed. With a DC motor, the speed is directly proportional to the supply voltage with a constant load. In an AC motor, speed is determined by the frequency of the applied voltage and the number of magnetic poles. AC motors will withstand higher current and are more commonly used in servo applications such as robots, in-line manufacturing, and other industrial applications where high repetitions and high precision are required.

A DC servo motor is commutated mechanically with brushes, using a commutator, or electronically without brushes. Brushed motors are generally less expensive and simpler to operate, while brushless designs are more reliable, have higher efficiency, and are less noisy. A commutator is a rotary electrical switch that periodically reverses the current direction between the rotor and the drive circuit. It consists of a cylinder composed of multiple metal contact segments on the rotor. Two or more electrical contacts called "brushes" made of a soft conductive material such as carbon press against the commutator, making a sliding contact with segments of the commutator as it rotates.

The majority of motors used in servo systems are AC brushless designs, however, brushed permanent magnet motors are sometimes employed as servo motors for their simplicity and low cost. Brushless DC motors replace the physical brushes and commutator with an electronic means of achieving commutation, typically through the use of Hall effect sensors or an encoder. AC motors are generally brushless, although there are some designs (such as the universal motor that can run on either AC or DC power) that do have brushes and are mechanically commutated.

While DC motors are generally categorized as brushed or brushless, AC motors are more often differentiated by the speed of their rotating synchronous or asynchronous field. In an AC motor, speed is determined by the frequency of the supply voltage and the number of magnetic poles. This speed is referred to as synchronous speed. In a synchronous motor, the rotor rotates at the same speed as the stator's rotating magnetic field.

In an asynchronous motor, referred to as an induction motor, the rotor rotates at a speed slower than the stator's rotating magnetic field. The speed of an asynchronous motor can be varied utilizing several control methods such as changing the number of poles and changing the frequency.

3.2.7 Hit-Indication Device Selection

The primary goal of Group 33's sentry turret project is to be used as a nonlethal property security device. With this goal in mind, the members of Group 33 must employ some indication device to clearly identify the target hit by the sentry turret. The options considered include a laser diode, a catapult-like launcher, and a paintball gun.

When deciding whether or not a laser diode would be appropriate for the project, many factors were taken into account. First, the laser diode would not leave a lasting impression on the target, meaning the device would only function as a proof of concept for a sentry turret with an ideal, linear projectile. Causing some noticeable effect to the target of the sentry turret was an important objective for the development team, as the threat of harm imposed by the security turret would discourage any trespassing to begin with, and a visible mark on any targets would identify to law enforcement officials any individuals who have trespassed upon a location where the sentry turret was active. The failure of a laser diode to render some effect unto the sentry gun's target was a factor that was relevant enough in-and-of itself to convince the members of Group 33 that the laser diode would not satisfy their preferences for the project.

In considering the use of a catapult-like projectile launcher, the team not only had to consider the ability of the launcher to fire the projectile, but also the projectile's ability to maintain its course while traveling the required distance. After construction, the catapult was expected to be large in size and in weight, while also requiring much technology to perform physical calculations when firing each projectile. The sentry turret project is expected to be stationary in use, but easily placed in an unassuming location to avoid being tampered with; because of this, the size and weight of a catapult launcher would result in sacrifices to the overall objective of the project. The technology, calculations, and adjustments required to calibrate the catapult each time it is fired would also have a significant impact on project feasibility due to the constraints of time and money. Ultimately, while the catapult idea was an interesting concept, it was determined that this option would not be the most appropriate for the purposes of the project.

The next option discussed was to attach a paintball gun to the sentry turret as the hit-indication mechanism. A paintball gun would fire projectiles at a near-linear trajectory within a certain range, while also leaving a lasting impression on the target hit. This option accounts for the benefits of the laser diode (within a given range) by firing the paint-filled projectiles along a relatively straight path toward the target, as well as by leaving a visible impression upon the target(s) hit. Unlike projectiles fired from a catapult, the projectiles fired from a paintball gun are all uniform spherical objects. Firing spherical paintballs through the barrel of a pneumatic gun would result in a much more replicable outcome than firing some projectile (even the same projectiles) from a catapult. This means the use of a paintball gun would be more efficient for the sentry turret, as each projectile is more likely to hit the target.

The information presented in **Table 7**: **Device-Benefit Technology Comparison** (below) summarizes the important factors considered while selecting which hit-indication device would be best for the sentry turret project. The table uses numerical values to indicate the degree to which each option satisfies the goals of the hit-indication mechanism, as defined by the members of Group 33. The degrees of satisfaction within the table range from 1 to 3, with the lowest value being the least satisfactory. These values are relative to one another and do not accurately represent any performance gap between the different devices/mechanisms. The benefits selected for use in the table were derived in part through reference to the house of quality in **Figure ***. The weight of each benefit present in the table is used to quantify the value placed upon each factor. The scale used for the weight values in the table ranges from 1 to 10, with the highest value being the most important. The weights of each benefit are not unique, and it is evident that several benefits are considered to be equally valuable. Total values are the sum of the products of each satisfaction value and the corresponding weight for that row of the table. Using this method, the device with the greatest amount of satisfaction with regard to the importance of each benefit will have the highest total value.

As demonstrated by **Table 7** (below), the paintball gun hit-indication mechanism was determined to be the most appropriate device for the task of deterring and identifying targets using the sentry turret product. The paintball gun was the most moderate option of the choices listed, but it did not suffer from the shortcomings of either of the other choices.

			Device		
Benefit	**Weight***		Laser Diode	Catapult	Paintball Gun
			Satisfaction**		
Accuracy	10	x	3	1	2
Ease-Of-Use	5	x	3	1	2
Effectiveness as a Deterrent	10	x	1	2	3
Effectiveness as an Identifier	10	x	1	2	3
Range	7	x	3	1	2
Reusability	8	x	3	1	2
Size	4	x	3	1	2
Versatility	3	x	1	3	2
Total***	-	-	**125**	**83**	**134**

*** The values under the table heading ‘Weight’ are subjective values indicating the importance of each benefit on a scale from 1-10. The higher the weight indicated on this scale, the more importance that benefit holds.**

**** The values under the table heading ‘Satisfaction’ are relative values indicating the degree to which each device satisfies the corresponding benefits, with respect to the other devices.**

*****Weight and Satisfaction values are multiplied before being summed for the ‘Total’ value of each device**

Table 7: Device-Benefit Technology Comparison

4.0 Parts Selection

For this section we will be discussing the parts which were selected for each section of the turret's design. Each part was decided upon after careful considerations based on the importance of the part, the cost of the options available, the benefits of each option, and the caveats of each option. During development of the prototype our selections have changed to fit the need or budget.

4.1 Development Board

Originally, we planned that the microcontroller/SBC (single board computer) would operate alone – recording input from the camera, processing this input to locate targets, calculating necessary adjustments to the stepper motors, and firing the gun all at the same time. The complexity and nature of these requirements necessitates a powerful microcontroller, and we eventually concluded that SBCs capable of all of these tasks would be far too expensive for our team's budget. Therefore, a laptop will take the SBC's place for computer vision processes, and then relay commands to the turret's stepper motors through a cheaper microcontroller/development board. However, we have kept the SBC information in this document in the event that we determine the SBC would be a more suitable solution.

The three single-board computers we considered before determining that none of them would be sufficient for our requirements were the Raspberry Pi, Odroid, and BeagleBone. The most promising SBC from the outset of our research was the Raspberry Pi, due to its dominance in the hobbyist SBC market. The Raspberry Pi has a large number of useful peripherals available and has a large community of hobbyists to provide technical support through online forums if our team ever needs it. There are a couple variants of the Raspberry Pi 4, which each have a different amount of RAM on board.

The specifications of the Raspberry PI 4 are as follows:

- Broadcom BCM2711 quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz CPU
- SDRAM depends on the variant
    - 2GB variant costs ~$35, 4GB variant costs ~$55, 8GB variant costs ~$75
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet port
- 2 USB 3.0 ports; 2 USB 2.0 ports
- Raspberry Pi standard 40 pin GPIO header
- 2 micro-HDMI ports (supporting up to 4kp60)
- 2-lane MIPI DSI display port & 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- OpenGL ES 3.1, Vulkan 1.0
- Micro-SD card slot for loading OS and data storage
- 5V DC via USB-C connector (minimum 3A)
- 5V DC via GPIO header (minimum 3A)
- Operating temperature of 0 – 50 degrees C ambient
- Power consumption of 575mA (idle) to 885 mA (loading an OS)

  - Citation -> https://raspi.tv/2019/how-much-power-does-the-pi4b-use-power-measurements

Another possible choice is the Odroid XU4, which is a high-performance microprocessor for ~$50. Its specifications make it similar to, but slightly more powerful than the 2GB Raspberry PI 4. The problem with this board is that it runs at 1.8V rather than the Raspberry-set standard of 3.3V, which requires a Shifter Shield and makes the electrical wiring of the turret more complicated (and gives us another piece of equipment we need to purchase). The Odroid XU4 may also lack the number of accessories, drivers, and community support that the Raspberry PI has, due to being an "alternative board" and much less popular.

Odroid Specs:

- Samsung Exynos5422 Cortex-A15 2GHz and Cortex-A7 Octa core CPUs
- Mali-T628 MP6 (OpenGL ES 3.1/2.0/1.1 and OpenCL 1.2 Full profile)
- EMMC5.0 HS400 Flash Storage Interface (eMMC module sold separately)
- 2 USB 3.0 Host, 1 USB 2.0 Host
- Gigabit Ethernet port
- HDMI 1.4a for display
- Power: 5V/4A input

https://wiki.odroid.com/old_product/odroid-xu3/hardware/xu3_hardware

The final SBC is the BeagleBone Black, a Linux based community supported development platform for hobbyists. It has greater storage capabilities than the Raspberry Pi, at the expense of fewer connections (USB & wireless) and weaker RAM for the same price. Its specs are as follows:

- AM3358 ARM Cortex-A8 Processor
- 512MB DDR3 RAM
- 5V Power
- Connectivity:
  - 1 USB Host
  - 1 Mini-USB client
  - 1 10/100 Mbps Ethernet
- 2 x46 pin headers
- 4GB on board storage using eMMC
- Costs $55

In **Table 8** below, we compare the three SBCs based on their processor, RAM, power, connectivity, GPIO Pins, data storage, and their retail price. We initially decided that the Raspberry Pi 4 was the best of them.

Specs	Rasberry Pi 4	BeagleBone Black	Odroid XU4
Processor	Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz	AM3358 ARM Cortex-A8	Samsung Exynos5422 Cortex-A15 2Ghz (2 Cortext-A7 Octa core CPUs)
RAM	1, 2, or 4GB	512MB DDR3	2GB LPDDR3
Power	5V	5V	5V
Connectivity	• 2 USB 3.0 ports • 2 USB 2.0 ports • 2.4 & 5.0 GHz IEEE 802.11ac wireless • Bluetooth 5.0 • BLE Giga-bit Ethernet	• 1 USB host • 1 Mini-USB client • 1 10/100 Mbps Ethernet	• 2 USB 3.0 ports • 1 USB 2.0 port • 1 Gigabit Ethernet • 1080p HDMI
GPIO Pins	Standard 40-pin GPIO	2 x46 pin headers	Standard 40-pin GPIO
Storage	MicroSD	4GB on-board storage using eMMC	16GB eMMC module (Sold seperately)
Price	$55	$55	$60

Table 8: SBC comparison chart

After further consideration, we realized that even the Raspberry Pi 4's hardware specifications were insufficient to meet the demands that the computer vision software would place upon it. If we used the Raspberry Pi, our program would run into frame rate issues severe enough to impact the turret's reaction time, and in turn reduce the accuracy of the turret to an unacceptable level. An SBC that would theoretically be able to handle the computer vision software would likely cost too much for our budget. So, as mentioned earlier, our group changed our plan of using an SBC to using a microcontroller in conjunction with a laptop. The first development board to consider is the Arduino, of which there are several versions. A comparison of these versions is shown in **Table 9** below.

Arduino	CPU	Memory	I/O Pins	Operating Voltage	Price
Uno	16MHz ATmega328P	32KB SRAM, 32KB flash memory	14 digital I/O pins (6 PWM); 6 analog input pins	5V	$23.00
Leonardo	16MHz ATmega32u4	2.5KB SRAM, 32KB flash memory	20 digital I/O pins (7 PWM); 12 analog input pins	5V	$20.70
Due	84MHz AT91SAM3X8E	96KB SRAM, 512KB flash memory	54 digital I/O pins (12 PWM); 12 analog input pins; 2 analog output pins	3.3V	$40.30
Mega	16MHz ATmega2560	8KB SRAM, 256KB flash memory	54 digital I/O pins (15 PWM); 16 analog input pins	5V	$40.30

Table 9: Arduino Comparison Chart

Below is a rundown of additional Arduino board specs for consideration. These will include recommended input voltages and limits.

Arduino Uno

- 7-12V input voltage (recommended)
- 6-20V input voltage (limit)

Arduino Leonardo

- 7-12V input voltage (recommended)
- 6-20V input voltage (limit)
- Inbuilt USB 2.0, allowing direct communication with a PC
- Less popularity & support than Uno, and lacks compatibility with Uno shields

Arduino Due

- 7-12V input voltage (recommended)
- 6-16V input voltage (limit)
- Much more powerful than Uno but can’t easily interface with 5V devices.

Arduino Mega

- 7-12V input voltage (recommended)
- 6-20V input voltage (limit)
- Compatible with Uno shields
- Not as powerful as the Due, but has far fewer compatibility issues

The Arduino Uno is sufficient for our project – there is no need to spend more on the additional computing power that the Arduino Due or Mega would grant us. Arduino Leonardo sacrifices too much power for too little of a reduction in cost.

Controller	Tic T500	Tic T834	Tic T825	Tic T249	Tic 36v4
Operating voltage range	4.5V to 35V	2.5V to 10.8V	8.5V to 45V	10V to 47V	8V to 50V
Max continuous current per phase (no cooling)	1.5A	1.5A	1.5A	1.8A	4A
Peak current per phase (requires additional cooling)	2.5A	2A	2.5A	4.5A	6A
Microstep resolutions	Full Half ¼ 1/8	Full Half ¼ 1/8 1/16 1/32	Full Half ¼ 1/8 1/16 1/32	Full Half ¼ 1/8 1/16 1/32	Full Half ¼ 1/8 1/16 1/32 1/64 1/128 1/256
Automatic Decay Selection	yes	no	no	yes	yes
Automatic Gain Control	no	no	no	yes	no
Driver IC	MP6500	DRV8834	DRV8825	TB67S249 FTG	Discrete MOSFETs
Price	$27.95	$39.95	$39.95	$49.95	$69.95

Table 10: Specs for Pololu Tic stepper motors controllers

Another option that would take the place of both the development board and stepper motor driver would be the Pololu Tic stepper motor controllers. The Tic controllers could theoretically be used in lieu of both the microcontroller and stepper motor drivers, at a highly affordable price. The Tic controllers can be connected directly to the laptop via USB, but it raises the problem of available USB ports on the laptop. The camera will already be using a USB port, and we will need two Tic controllers (one for each stepper motor). However, the logistics concerning USB connection its power supply eventually caused us to discard the Tic controllers. The specs are listed in the **Table 10** above.

Ultimately, after deciding that we would have the laptop handle the turret's computer vision software, we decided to use a PCB utilizing the ATMega328 microprocessor (the same microprocessor used by the Arduino Uno). Practically speaking, our microcontroller will be a home-brewed Arduino board, with a design specialized for our purposes. This would satisfy the hardware needs of the project while keeping expenses as low as possible. The only difficulty that buying & utilizing the ATMega328 separately poses is the issue of the bootloader for the Arduino IDE. Most lone ATMega328 chips will not come pre-installed with the bootloader. This means that we will not be able to use the Arduino IDE for the microcontroller, which is one of the most important benefits of using an Arduino board / ATMega328 microprocessor. However, our group possesses two knock-off Arduino boards we were planning to use for prototyping purposes. These boards can be used to burn the bootloader onto the ATMega328.

4.2 Motor Selection

The blueprint for our turret calls for three separate motors. One for the turret pan (horizontal transversal), one for the turret yaw (vertical transversal), and another for the trigger mechanism. The operation and requirements of the pan and yaw motors are similar enough to group them together as transversal motors, while the requirements for the trigger motor are slightly different. This means we will be using the same motor for both x-axis and y-axis transversal, and a second type of motor for the trigger mechanism. We only need to deduce what type of motor is best suited for each task.

We have three motor types to choose from: stepper motors, DC motors, and servo motors. Each type has its own advantages and disadvantages compared to the others, which make them more suitable for some tasks than the others. The table below compares the advantages and disadvantages of these motors.

For panning of the sentry turret along the x-axis and yaw of the turret along the y-axis, the members of Group 33 determined a stepper motor would most adequately suit their needs, particularly due to its capabilities in setting position and providing holding torque to maintain that position. The yaw of the device will require a reasonable amount of holding torque to ensure that the vertical aim of the paintball gun will result in a hit on the target upon being fired. Neither a DC motor nor a servo motor could suit this purpose, as a DC motor lacks the precision necessary, while a servo motor lacks the holding torque needed.

For the trigger activation mechanism used to pull the trigger automatically, a servo motor was deemed most appropriate. To complete this task, a motor with precise positioning and positioning feedback was required. This positioning information would be used to verify the location of the point which makes contact with the trigger both while being fired and while at rest. This location data is significant because the position of the contact point determines the amount of rotation necessary to fire another single round from the paintball gun at the target in question. Of the options available, the motor with the most precise control and feedback over its position is the servo motor. **Table 11** will outline the advantages and disadvantages of each motor for consideration.

Motor Type	Advantages	Disadvantages
Stepper	+ Precise positioning + Precise speed control + Excellent torque at low speed + Excellent torque to maintain position	– Limited torque at high speed – Low efficiency – More complex to control
DC	+ Efficient + Reliable + Simple control	– Some brushless motors require a specialized regulator to control – Imprecise
Servo	+ Consistent torque at varying speeds + Excellent torque at high speed + High variety in size and torque ratings + Direct control over positioning	– Limited range of motion, usually 180 degrees – Small adjustments while attempting to hold a steady position

Table 11: Advantages & Disadvantages of Motor Types

4.2.1 Turret Transversal Mechanism

For the turret's transversal mechanism, we chose to use stepper motors. Stepper motors have maximum torque at low speeds (less than 2000 rpm), making them better for applications that require high precision, if you are willing to sacrifice high speeds. Normal DC motors and servo motors do not have much torque at low speeds, and thus have lower precision. Since we want our turret to have a high accuracy, greater precision is very important for the transversal motors. This precision trumps the DC motor's advantages in cost, efficiency, and ease of control. While servo motors are also capable of precise motion control, they don't exactly have holding torque, which results in jittering while maintaining a steady position. This could harm the turret's accuracy, so the stepper motor was chosen instead.

For comparison, DC motors are simpler to control than Stepper motors, as they do not require a microcontroller like stepper motors do. DC motors are controlled entirely by voltage.

Since the current consumption of stepper motors is independent of load, and they constantly draw maximum current, the efficiency of stepper motors is generally much lower than that of DC motors.

After deciding what type of motor to use, it was time to decide what type of stepper motor to use, specifically what NEMA size would it be. The three sizes that would work best for our project were NEMA 17, NEMA 23, and NEMA 24. Their typical specifications are listed in **Table 12** below.

Frame size	Diameter (mm)	Typical torque range (Nm)	Typical speed range (RPM)
NEMA 17	42	0.2 - 1	0 - 1000
NEMA 23	57	0.5 - 3	0 - 1000
NEMA 24	60	1.2 - 4.6	0 - 1000

Table 12: Nema stepper motor size comparison

After estimating the size and weight of the turret's structure and gun, we decided to use the NEMA 23 category stepper motor. A NEMA 17 motor could also work, a smaller motor could mean a smaller turret exterior, thus making the turret lighter and easier to transport. However, using a smaller motor may also result in insufficient torque for the turret to function properly. Using a larger NEMA 23 size would ensure that the motors chosen would have enough torque while not making the turret itself too large. We narrowed the choice down to three specific motors as shown in **Table 13**.

Diameter isn't everything when it comes to stepper motor power. Changing stack length generally won't impact speeds, but it will have a major impact on torque. For example, the ZD2N2318 and ZD10N2318 stepper motors are both NEMA 23, but the ZD2N2318 is 42mm long while the ZD10N2318 is 104mm long. The ZD2N2318 has 0.6Nm while the longer ZD10N2318 has 2.4 Nm. The extra length has allowed a greater electrical power to get into the motor at any one time, and thus delivers more torque to the motor shaft.

Motor	E-Series Nema 23	P-Series Nema 23 x76	Nema 17
Dimensions	57x57x56mm	57x57x76mm	42x42x34mm
Step Angle	1.8 degree	1.8 degree	1.8 degree
Holding Torque	1.26Nm (178.4oz.in)	1.9Nm (269oz.in)	0.26Nm (36.8oz.in)
Rated Current / Phase	2.8A	2.8A	0.4A
Number of Leads	4	4	4
Lead length	300mm	500mm	1000mm
Price	$26.78	$32.05	$22.89

Table 13: Stepper motor selection

We eventually decided to use the E-Series Nema 23 stepper motor available on StepperOnline, pictured in **Figure 16.** It was chosen primarily for the balance it struck between holding torque and price. The torque of the E-Series was deemed sufficient, and was chosen over the stronger P-Series because it was about $6 cheaper.

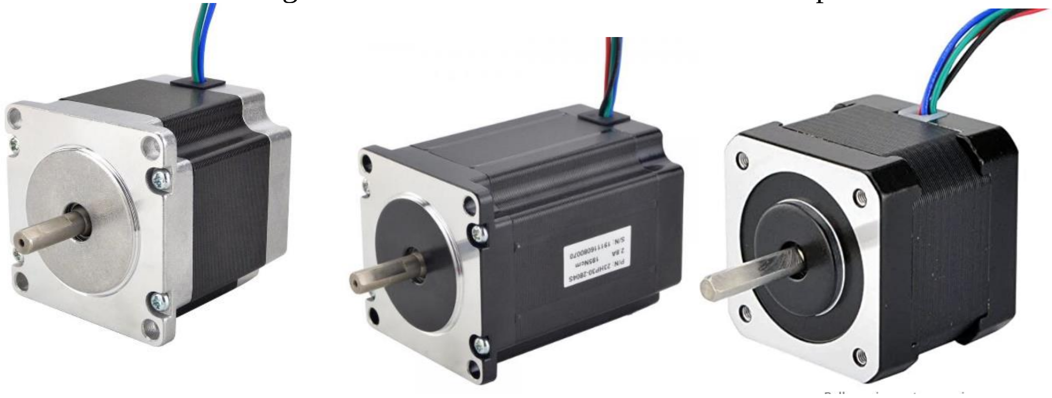


Figure 16: From left to right: E-Series NEMA 23 Stepper Motor, P-Series Nema 23 x76, and Nema 17

https://www.omc-stepperonline.com/e-series-nema-23-stepper-motor-bipolar-1-8deg-4-8-nm-679-87oz-in-6-0a-86x86x80mm-4-wires.html

4.2.2 Stepper Motor Driver

The stepper motor driver chosen can affect the performance of the motor. If the controller is not able to deliver more power than the motor can handle, then it is unlikely that the motor will be able to achieve its maximum possible mechanical performance
The correct stepper motor driver depends on the specific stepper motor used. Some drivers will fit different motors better than others. **Table 14** shows the specs of each stepper motor.

Driver	Op Voltage	Continuous current/phase	Max current/phase	Microstep	Price
DRV 8825	8V - 45V	1.5 A	2.2 A	Full, ½, ¼, 1/8, 1/16, 1/32	$11.95
DRV 8880	6.5V - 45V	1.0 A	1.6 A	Full, non-circular ½, ½, ¼, 1/8, 1/16	$8.95
A4988 (Black)	8V - 35V	1.2 A	2.0 A	Full, ½, ¼, 1/8, 1/16	$7.49
TB67S128FTG	6.5V - 44V	2.1 A	5.0 A	Full, ½, ¼, 1/8, 1/16, 1/32, 1/64, 1/128	$13.95

Table 14: Stepper Motor Driver Selection

The DRV 8825 has the thermal & electrical protection for overtemperature, overcurrent, undervoltage, short-to-ground, and short-low. It also has a built-in 3.3 regulator, making it easy to interface with Raspberry PI SBCs.

The DRV 8880 stepper motor driver carrier has the thermal & electrical protection for overtemperature, overcurrent, short-to-ground, and short-low. It offers dynamically scalable current limiting and “AutoTune”, which automatically selects the delay mode each PWM cycle for optimal current regulation performance based on factors like the motor winding resistance and inductance and the motor’s dynamic speed and load.

The A4988 (Black Edition) has protection for over-temperature thermal shutdown, under-voltage lockout, crossover-current protection, short-to-ground and shorted-load protection.

The TB67S128FTG stepper motor driver carrier has protection against under-voltage, over-current, over-temperature, shorting, and reverse-voltage protection (up to 40V). It has adjustable current control allowing you to set the max current output with a potentiometer.

We chose the TB67S128FTG stepper motor driver carrier for this project (Pictured in **Figure 17**). It can easily handle 24 volts, interfaces well with the chosen microcontroller, has a small size, comes with the terminal blocks and header pins (this isn’t unusual but still convenient), has 32 micro steps, can hit the amperage needed for the stepper motor’s top torque, and is on the cheaper side of stepper motor drivers.

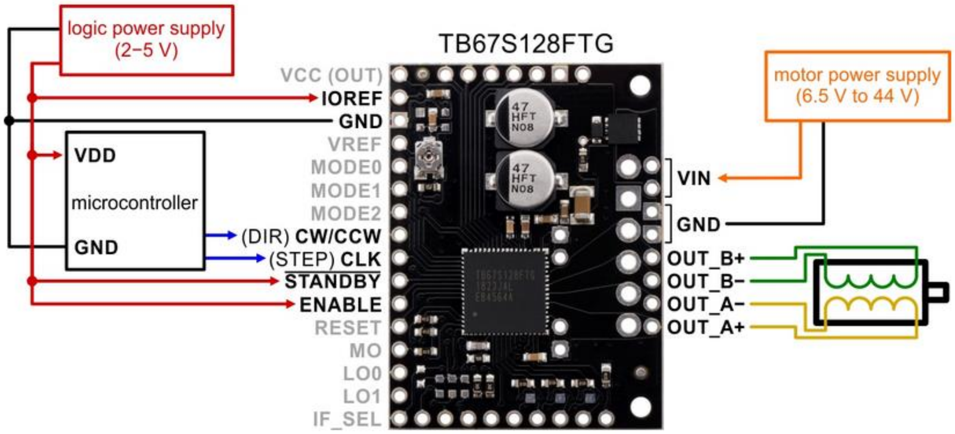


Figure 17: Minimal wiring diagram for connecting a microcontroller to a TB67S128FTG

From its Pololu product page: https://www.pololu.com/product/2998

4.2.3 Trigger Mechanism

For the trigger mechanism, we will be using a servo motor. Our choice of servo motor will primarily be decided by the servo's torque, speed, size, power requirement and rotation angle. The most important requirement is torque, since it won't matter how fast the motor is, or how large its rotation angle is if the servo can't pull the trigger in the first place. The servo must be fast enough to pull the trigger at an acceptably high rate to improve the chances of hitting the target. The servo must also have a limited size and weight so that it can fit on the paintball gun and not affect the gun's aim with its own weight. A rotation angle of at least 90 degrees is necessary for the trigger mechanism to function.

Servo motor torque is most often measured in kg-cm or oz-in, while the trigger pull weight of paintball guns is generally measured in pounds. The Tippmann 98 paintball gun's trigger pull is 2.5 pounds. In terms of torque, we are estimating that the trigger pull is 2.5 pound-inches. 2.5 pound-inches is equivalent to 2.88 kg-cm, or 40 oz-in. To ensure that whatever servo motor we choose for the trigger mechanism is capable of actually pulling the trigger, we will aim far above this torque. **Table 15** shows available specs for each servo motor and the final selection we chose.

Servo	Power	Speed	Torque	Rotation Angle	Size (L x W x H)	Price	Notes
Tower Pro MG995	4.8V - 6.0V DC	60 deg in 0.2 sec	8.5 kg-cm	120 deg	40.7mm x 19.7mm x 42.9mm	$11.99	Metal gears. Ball bearing design.
Tower Pro MG995 R	4.8V - 6.0V DC	60 deg in 0.20 sec	9.4 kg-cm	120 deg	40.7mm x 19.7mm x 42.9mm	$19.95	Upgrade to MG995
Hitec HS-311	4.8V - 6.0V DC	60 deg in 0.19 sec	3.0 kg-cm	96 deg; 202 deg with travel turner	40.0mm x 20.0mm x 36.5mm	$13.49	Top resin bushing, nylon gears
Hitec HS-645 MG	4.8V - 6.0V DC	60 deg in 0.24 sec	7.7 kg-cm	90 deg; 197 deg with travel turner	40.2mm x 19.8mm x 39.0mm	$35.99	Metal gears, ball bearing design
Feetech Fi7635 M	6.0V - 7.4V	60 deg in 0.17 sec	28.8 kg-cm	180 deg	55.0mm x 20.0mm x 38.0mm	$29.49	Digital, coreless, ball bearing, metal gears
Miuezut h DS3235 35KG	6.0V - 7.4V	60 deg in 0.13 sec	32kg-cm	180 deg	40.0mm x 20.5mm x 40.5mm	$20.00	Digital, coreless, ball bearing, metal gears
Feetech FT5335 M	6.0V – 7.4V	60 deg in 0.20 sec	35kg-cm	120 deg	62.8mm x 32.5mm x 55.9mm	$45.95	Digital

Table 15: Servo Motor Selection

The servo motor decided upon by the members of Group 33 for use with the trigger activation mechanism is the Tower Pro MG995, depicted in **Figure 18: Tower Pro MG995 Servo Motor with attachments** (below). This motor satisfies the torque required to activate the trigger of the selected paintball gun, the speed required to fire each round reliably, and the rotation angles required to position the point of contact as desired between each shot. The motor operates within a voltage range that will not draw too much power and is acceptable for its purpose. Also, the size of the motor was an important consideration, as the attachment of the trigger activation mechanism should not be unreasonably large. This motor is small in size and can be fitted into the design of the attachment's housing with ease. Perhaps most important of the factors considered when selecting the appropriate servo motor, the Tower Pro MG995 is very cost-effective compared to other options; a pack of two of these motors can be purchased for only $11.99 per pack.

Figure 18: Tower Pro MG995 Servo Motor with attachments

4.3 Power Supply Selection

Power supply must be selected nearly last, as the option selected is entirely dependent upon the power requirements of the system as a whole. This means that the power consumption of each element included as part of the sentry turret project will have an impact on the viability of each option considered. To determine the best power supply for the purposes of this project, the members of Group 33 analyzed the advertised values of voltage, current, and total power consumption over time for each part of the sentry turret device. After summing the values for each factor listed previously in their respective categories, the best power supply was determined to be one which would output at least the power necessary, if not more, to exceed the totals calculated. The possible selections are listed in **Table 16** below with their respective specs and prices.

Power Supply	Type	Voltage (V)	Capacity (Ah)	Weight (lbs)	Size (in)	Price
Tmezon Power Adapter	Power Adapter	12	N/A	N/A	N/A	$8.99
Universal Battery UB1280	Rechargeable Battery	12	8	4.96	5.94 x 2.56 x 3.94	$20.89
TalentCell PB240A1	Rechargeable Battery	24	22.4	1.43	0.94x 2.48x 4.13	$72.79
Duracell Ultra DURDC12-55P	Battery	12	55	42.26	8.98x 5.39x 9.06	$174.99 **(FREE)**

Table 16: Power Supply Selection

The use of power adapters such as the Tmezon adapter was rejected in order to improve the portability and independence of the turret. The reliance on a wall outlet massively reduces the possible placements of the turret.

The TalentCell PB240A1 is very flexible, small, and lightweight (especially compared to the heavy Duracell battery). However, it is very expensive and has a much lower capacity than the Duracell battery.

A 24V power supply would be better and cleaner, however, the 12V Duracell Ultra DURDC12-55P (Pictured in **Figure 19**) was previously owned by a group member (and thus free), so we decided to use the Duracell battery instead. The savings from this are simply too immense to pass over. Another advantage of the Duracell battery is its large capacity of 55-amp hours, which is much more than any of the other batteries we considered. Since we have two Duracell batteries, they can be placed in series to provide a total voltage of 24V.

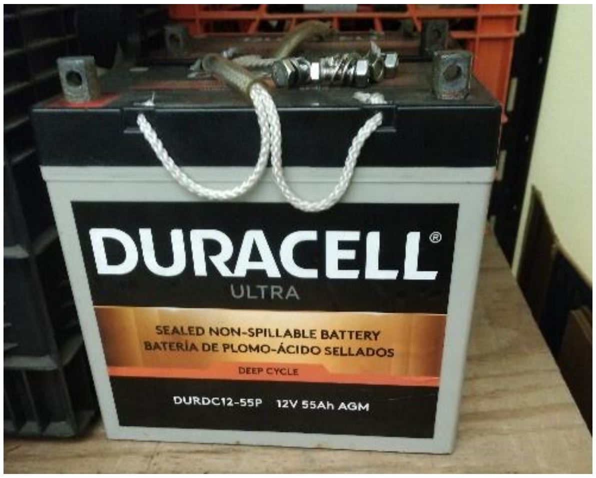


Figure 19: The Duracell Ultra DURDC12-55P Battery

4.4 Lighting/Warning System Selection

If the turret robot's camera does not have night vision capabilities, then it will be necessary to add a light source for use in dark places and at night. This light source will be activated only after the motion sensor has been triggered; this will save energy as opposed to activating the light constantly when the sentry turret device is in a poorly-lit location. It is in the best interest of Group 33 and any consumers of this product to reduce the power-consumption of the product where possible. For ethical reasons, it is also in our interests to provide a warning system to the trespassers being targeted by the turret. Failing to provide a fair warning to potential targets of the system before firing at said targets may result in an unnecessary amount of force being used upon the targets.

To indicate to individuals that they have become targets of the sentry turret device, the members of Group 33 have considered equipping the sentry turret with a red light source which will serve as both the lighting and the warning system for trespassers. One possible issue in implementing this red-light system is the impediment of the camera's ability to recognize targets through computer vision algorithms. Detecting trespassers requires the device to maintain a capacity to contrast the silhouette of a potential target from any other data captured within view of the device's camera. If the light used is tinted red, the contrast between the silhouette and the background will be reduced, which can make detection more difficult, or even impossible. The reliability of the target-detection system is paramount to the functionality of this product.

Another solution considered is to utilize a regular fluorescent lightbulb for lighting, and a red LED to serve as the warning system. This combination would be more complicated to implement than the previous option, given that it also requires the addition of a standard fluorescent lightbulb and a socket for the lightbulb; however, it is expected that this option would not only overcome the obstacles presented by the previous consideration, but it would even improve the detection-rate of potential targets compared to having no light source whatsoever. The expectation for improved detection-rates compared a system without lighting is due to the fact that the use of a fluorescent bulb will increase the visibility of the camera and draw greater contrast between the silhouettes of potential targets and their surroundings. Furthermore, the red LED light warning system can be substituted for a laser pointer attached to the turret's gun. The addition of a laser pointer to the gun would have a greater intimidation factor than a simple red light, so long as it is noticed, and it would convince the trespasser to leave the turret's field of vision as soon as possible.

Another, more elegant solution is a speaker with a voice recording which could audibly warn the trespasser to leave the area before the turret fires upon them; although, this solution would be the most expensive in comparison to the previous solutions, and it would be far more difficult to implement on the sentry turret. Any speaker selected would, in itself, most likely be more expensive than any LED light or laser pointer, and it would require additional programming and on-board memory for the voice recording, which will assumedly loop until all targets have left or until the turret is otherwise deactivated.

4.5 Structure Materials Selection

The structure for the base of the turret is going to need to serve as a stable mounting point for the turret itself. The goal is to keep it relatively light while not sacrificing the necessary strength or stability that will allow the gun to shoot, and not have the base react to the recoil from the gun or shift if the motors move the gun too rapidly. For this the two best options for building the base would either be wood or metal. In order to remain within budget either pine or MDF would be used for the wood option. And for the metal option, aluminum or steel would be options.

To build the structure out of pinewood would come with the complication that given the turret is going to be primarily an outdoor system and would require sealing and general maintenance that comes with wooden objects subjected to the elements especially in more humid climates. Another issue pertaining to a wooden base is the fact that the strength of wood is much lower than the strength of metal, this could be a potential issue depending on how much and how fast the motors shift around the gun and if the wooden base is going to be heavy enough and strong enough to counteract the movement of the gun. If wood were to be used in the design of the structure the two options would either be to use pine or medium-density fiberboard (MDF). With pine wood the boards would need to be cut, requiring a circular saw or a table saw to allow for the proper angles to be cut. The other wood option would be to use a medium-density fiberboard which would be similar in cost to the pinewood structure, but it would be easier to manufacture. The simplest way to cut the MDF would be to use a laser cutter, which is a viable option as once the structure is modeled the cuts needed from the MDF can be exported to a DXF to be used by the laser cutter.

While MDF would be a light easy to manufacture solution for the base of this project the notable issue is the fact that it may be too light, and shift if the gun were to rotate too fast, and susceptible to the elements, meaning it would not last long in practical use. However, these issues could be negated by further weighing down the bottom of the base and sealing the wood, but the MDF would still be prone to cracking under stress.

The other options for the base of the structure include steel and aluminum. Both metal options would be feasibly manufacturable, the main deciding factor between the two metals would be the weight of the final structure. In which case steel is be nearly three times the weight of aluminum, so even though steel is notably cheaper than aluminum the structure would become very difficult to move. Comparably aluminum, while more expensive, is considerably lighter than steel and would allow the structure to remain within the weight specifications. In order to manufacture either of the metal types would require the use of a cold cut saw and a welder, this way manufacturing the structure can be done without needing outside assistance, which would keep the costs lower and comparable to the cost of manufacturing the wooden structure.

When comparing the pros and cons between the four proposed options: pine, MDF, steel or aluminum. The best choice would be aluminum as it will provide the structural integrity that cannot be obtained from a wooden base and wouldn’t be too much more cost wise. As well as much lighter than steel, allowing it to be easily movable. Finally, aluminum would be easy to assemble and seal to be used in outdoor environments. **Table 17** shows the comparison of material with cost and what we have determined to be the final choice for structure material. This may change due to budget constraints or if it cost is too great to have the aluminum cut.

Material	Amount	Cost
Pine Wood (2”x4”x8’)	2	~$20
Medium-Density Fiberboard (3/4 inch.)	1	~$30
Aluminum (1 inch. sq. tube)	12ft.	~$30
Steel (1 inch. sq. tube)	12ft.	~$50

Table 17: Structure Materials Cost Comparison

The next component of the structure to consider is the casing for the electronics. This will need to house the PCB, battery and all other necessary electronic components. Water-resistant housing will not be necessary for any of the motors, as the NEMA 23 motors have an IP65 rating, meaning that they are protected against water jets from all angles, which will be water-resistant enough for the purposes of this project. Considering cost is essentially equivalent between all three options it will not be a determining factor. The biggest factor when making the decision on how best to create the electronics casing will come down to what will be the cleanest and most manageable unit.

The options for water-resistant electronics housing include 3-D printing, laser cutting acrylic or purchasing a premanufactured case. If the project were to use a premanufactured water-resistant case it would be the easiest option as the only customization that could possibly be needed would be if more holes for the wires and cooling fan were necessary. So, while this would be the simplest option, it is not going to be the cleanest or best option. Mostly since the case will more than likely not be the correct size so the wire management will not be the best and the additional holes that would be added would cause the case to require additional water protection.

The next option to consider is 3-D printing a case for the electronics which would be easy to design and assemble. The biggest constraint would be the time it would take to print and if one of the prints happened to fail or pint incorrectly then it would add more time to construction of the housing. The case would be assembled using epoxy to seal the seams of the piece and help to waterproof the case.

The final option to discuss is acrylic. The acrylic, similar to the 3-D printed option would be simple to customize to the correct dimensions necessary and would be assembled with epoxy. This option has the most positives as it will be easy to design and manufacture, as it can be cut from one piece of acrylic and laser cut. This option would also allow for the electronics to be viewable and for the warning LED to also be placed inside the clear encloser. **Table 18** shows the material comparison to use for camera and LED casing.

Material	Amount	Cost
Acrylic (23.75”x47.75”)	1	$14.77
Filament (1.75mm)	1	$15.99
Waterproof case	1	$20.00

Table 18: Material for the 3D printed case

The last piece on the system that would need to be protected from the elements is the camera. At this point it is assumed that the camera will not be included in the main electronics housing thus it will require its own separate case with an unobscured window for the camera to look through while maintaining a decent level of water resistance. The options for this include creating an acrylic box with a small port for the cord or buying a premade waterproof box. The most important part being that the viewing window for the camera remains clear. Considering the premade box, the biggest issue would be finding a suitable size with mounting holes oriented correctly. Assuming the main electronics housing is already made of acrylic then it would be easy to add the cuts needed for the camera case to the sheet of acrylic that had already been purchased.

4.6 Camera Selection

As mentioned in our technology section, a high-resolution camera is needed for computer vision in our prototype. The frames per second would need to be sufficient for computer vision to recognize targets. Also, our camera needs to be compatible with our microcontroller. Fortunately, there are many options that are compatible on the market today. There are many improvements of webcams with high quality pictures and accommodating frame rates. The following are three models of webcams our group considered.

4.6.1 Logitech C270 webcam

The first potential webcam is the Logitech C270 webcam shown above in **Figure 20 (left)**. It is a high-definition camera that has a supported resolution of 720p at 30 frames per second. The picture quality is very clear with a plastic lens and a fixed focus with a diagonal field of view of 60 degrees. It also comes with a clip that can be used as a means of stability. At the time of writing this on Amazon it retails at $25.47, and on the main Logitech website, it is $27.99. This camera would meet the needs of our prototype as one of the best contenders for high quality at a low price point. The focus it has unmatched quality for an older Logitech. If we wanted to stay more towards the budget end of the spectrum this would be the webcam to pick.

4.6.2 Logitech C920s webcam

Our second choice is the Logitech C920s shown above in **Figure 20 (middle)**. This model offers upgraded features such as a 1080p resolution with the option to switch to 720p, for compatibility with some apps, and both resolutions support 30 frames per second. The picture quality is even better with this model as it has autofocus as its focus type. The lens is now glass, and its diagonal field of view is 78 degrees. This increased field of view would be greatly beneficial has in would increase the field of detection. The design of this model still accommodates a clip that can be used for stability. The increase of 1080p resolution could impact computer vision in a positive manner by increasing accuracy in distinguishing what is a target. The autofocus would also be a welcomed feature as it would boost accuracy of locating targets. Currently on Amazon, this model retails for $71.89 and $59.99 on Logitech’s website. This model is a large jump in price from the Logitech C270, but it adds better features and quality along with the price. This would potentially increase the accuracy of detection if willing to spend more.

4.6.3 Logitech C922 webcam

Our pick for a final contender would be the Logitech C922 webcam shown in **Figure 20 (right)**. This webcam is marketed as more of a professional web streaming camera, but it does offer excellent features. It supports a 1080p resolution with 30 frames per second or 720p resolution with 60 frames per second. With this model having double the frames there could be major increase to target detection with more samples, but it would decrease the quality of the image. It also has a autofocus focus type similar to the Logitech C920s as well as a glass lens. The diagonal field of view also remains the same at 78 degrees. Aside from the increased frame rate the quality is very similar to the Logitech C920s. Currently it retails at $75.49 on Amazon, and $99.99 on Logitech’s main website. It does come with a mini tripod, but it is not a necessary feature has it come with a clip as with the previous models.

4.6.4 Final Camera Choice

The camera used in our prototype will be heavily relied on due to computer vision, so we expected that it would be one of the more expensive components. Logitech is a very reputable company when it comes to computer peripherals. After reading many online reviews, Logitech’s webcams reliably perform at an affordable price point. Therefore, we preferred to choose from their selection for the comparison. All three cameras are excellent choices, but we ultimately chose the Logitech C920s webcam. It provides a flexible resolution of either 1080p or 720p with 30 frames per second which is sufficient for our prototype to perform with computer vision. Comparing the quality of Figures *, *, and *, the resolution quality of the C920s outperforms the C270 and is on par with the C922. The field of view is also larger compared to the C270 and the same as the C922. Overall, the C920s outperforms the C270, and is very similar in quality to the C922. The C922 is mainly used for streamers as it comes with Logitech Capture Software, advanced lighting features, and a miniature tripod. We will not be using any of these extra features, so the C920s fits the needs and budget of the project. **Table 19** shows a complete comparison of the possible selections of cameras.

Camera	Resolution	Frames Per Second	Field of View	Lens Type	Weight (ounces)	Cost
Logitech C270	720p	30	60°	Plastic	2.65	$27.99
Logitech C920s	1080p/720p	30	78°	Glass	5.71	$59.99
Logitech C922	1080p/720p	30/60	78°	Glass	5.71	$99.99

Table 19: Camera Comparison

Figure 20: From left to right, Logitech C270, Logitech C920, Logitech C922 webcam

4.7 Motion Sensor Selection

In the technology section three different types of motion sensors, we explained which were Microwave, Ultrasonic and Passive Infrared. After exploring their capabilities, we will look further into their specs. We will compare and contrast their differences to see which most fits the needs of our prototype.

4.7.1 RCWL-0516 Microwave (Doppler) RADAR Motion Sensor

The RCWL-0516, pictured in **Figure 21 (left)**, has many different manufacturers and is readily available from many sellers. It has a default range of 7 meters (22 feet). It uses waves to propagate as its method of detection and field of view which forms a cone. It uses 4V- 24V for operation, and 3mA max current draw. It is very bare bones compared to the other motion sensors considered, but it is readily available and very cost effective coming in at $4. When purchasing it does not come with any instructions or the pins to solder. There are many different datasheets online that are translated to help. Also, many have had success working with this cost-effective chip.

4.7.2 Maxbotix Ultrasonic Rangefinder - LV-EZ4 ID 982

The Maxbotix was considered as a possible for its long range of 6.45 meters (21 feet), with a wave that propagates out in a cone shape. Many Ultrasonic motion sensors do not reach a very long-range selection with some being under 200mm. It has a 2.5V- 5.5V supply with 2mA current draw. The sensor operates at 42KHz with readings that occur every 50mS at a 20Hz rate. It uses Pulse Width Modulation (PWM), analog voltage output VCC/512 volts pre inch, and serial digital output of 9600 baud. It is also compatible with Arduino and Raspberry Pi. Datasheets are readily available online to help with wiring. Out of the three choices of motion sensors this is the most expensive at $24.95 on adafruit.com. Below in the middle of **Figure 21 (middle)**, is an image of the Maxbotix Ultrasonic Rangefinder.

4.7.3 Parallax PIR Sensor (RevB)

Parallax PIR Sensor, **Figure 21 (right)**, (Rev B) has 15-30 max feet (9 meters) range and a 90 degrees wide diagonal field of view. It uses 3 to 6 VDC, 12 mA @ 3V, 23mA @ 5 V. The sensor itself has built-in LEDs that light up when movement is detected. It costs $14.95. This particular model (Rev B) is appealing for its high range of 30 feet, which outranges most other PIR sensors (which generally have max ranges of 20 feet).

PIR sensors are also easy to use with a microcontroller such as the Raspberry PI and Arduino. If we wanted a motion sensor with a greater range (such as 50 feet) it would require getting a motion sensor that may not have perfect compatibility with the microcontroller. This would result in a lot of time spent jury-rigging the motion sensor that could instead be spent on more important features such as fine-tuning the aiming system for higher accuracy.

4.7.4 Final Motion Senor Choice

The motion sensor will currently be used for two things turning on the turret from a low powered mode and also acting as a backup to track targets. Each of these motion sensors are heavily varied and could potentially meet our needs. For that reason, they must be heavily scrutinized based on functionality and what is sufficient. Also do their pros outweigh their cons. Table 20 explores key factors in deciding which motion sensor to use.

Sensor	Range	Voltage	Current Draw	Price
RCWL-0516	22 feet	4 – 24 V	3mA	$4.00
LV-EZ4 ID 982	21 feet	2.5 – 5.5 V	2mA	$24.95
PIR Sensor RevB	30 feet	3 – 6 V	23mA	$14.95

Table 20: Motion Sensor Comparison

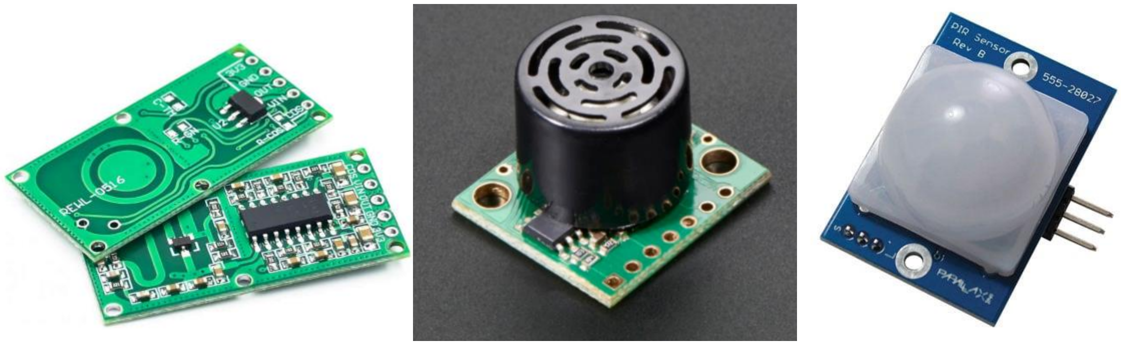


Figure 21: From left to right: RCWL-0516, LV-EZ4 ID 982, and PIR Sensor RevB

4.8.1 Paintball Gun Selection

There are several types of paintball guns available for consumer purchase. In sifting through the selections, the members of Group 33 encountered their first issue—the cost of most "new" units exceeded the budget for hit-indication device portion of the sentry turret project. Because of this, the members of Group 33 have elected to search not only for "new" paintball gun products, but also "used" or "refurbished" products as well.

Although product price was the first constraint in selection of the paintball gun used, there were still many other variations between the available paintball markers. Some markers were designed to be accurate across greater distances with sacrifices to their compactness, while others were designed for accuracy at shorter ranges with smaller, more manageable sizes.

When choosing the paintball gun that we will be implementing in our final design we began by determining what we would need out of the paintball gun. After filtering through the paintball markers by price, we looked at the size and weights of the different paintball guns, then we looked at the customizability of each of the different guns, as they will need to be outfitted to mount onto our system, and finally we looked at which type would be easiest to implement electronically.

Following the decision to search for lower-end paintball markers, we quickly realized that many of the more affordable models would not be as reliable nor as modifiable as needed for the project. The next issue that was encountered when choosing a paintball gun was the fact that the easiest gun to implement would have been an electric paintball gun, but even looking at used models, the electric paintball guns available were still above the $100 budget for this item. To work around this problem, we found a slightly older model of paintball gun that is reliable, modifiable, and well within budget.

4.8.1.1 First Selection: Tippmann Stryker XR1

- Firing modes: semi, burst, ramping, full auto
- Firing rate: 15+ bps
- Operating pressure 800psi

- Capable gases: HPA (compressed air) and Nitrogen
- Weight: 4lb

The first selection for paintball guns is the Tippmann Stryker XR1. Is one of the quietest and smoothest shooting paintball guns because it is fully electro-pneumatic and has a 9V battery. It can be set to various firing modes such as semi-automatic, burst, ramping, and full-automatic. It has a .68 caliber and fires 15 shots a second. It uses High Pressured Air (HPA) and nitrogen gas for firing. The effective range is 150+ which meets the requirement of standards. Reviews had nothing negative to report, but enjoyed the fact that they could change modes and it was fully electronic. It is also lightweight coming in at around 4 pounds.

4.8.1.2 Second selection: Tippmann Stormer Tactical .68 Caliber

- Firing modes: semi
- Firing rate: 8+ bps
- Capable gases: CO2, HPA
- Weight: 3lb 13oz

Second choice was the Tippmann Stormer Tactical .68 Caliber. It is very lightweight at 3 pounds without an air tank and made from a composite material. It is semi-automatic firing with a .68 caliber and fires 8 shots a second. It uses High Pressured Air (HPA) or CO2 to fire. The effective firing range is 150+ feet which is over the effective range of our prototype. There are no definite drawbacks reported by users only that it around 10 pounds with the air tank which is a little heavy when running around. This would not be a drawback for the prototype as the air tank will be attached to the struck so the paintball gun will not be carrying the full weight.

4.8.1.3 Third selection: Tippmann Model 98 Paintball Gun

- Firing Modes: Semi
- Firing rate: 8+ bps
- Capable gases: C02, HPA, Nitrogen
- Weight: 2.9 lb

Our third paintball gun is the Tippmann Model 98. It is a very reliable and durable paintball gun model. This model is very lightweight and weighs 2.9 pounds without a CO2 tank. It has semi-automatic firing with a .68 caliber and fires 8 shots per second. It functions with CO2, compressed air, or nitrogen gas to fire. The effective firing range is 150+ feet which would be very effective for our prototype needs. The only drawback to this paintball gun is that it has some recoil, but that can be accommodated with the design of our structure. A visual of the paintball gun can be seen in **Figure 22 (right)**.

4.8.1.3 Comparison and Choice of Paintball Gun

The paintball gun will be one of the most important components of our project. This will be the device that will mark our targets and is controlled by motors. The following **Table 21 (below)** shows the specs and the costs related to each model. Our final choice was based on cost, weight, and reviews on performance.

Paintball Gun Model	Trigger	Firing Modes	Firing Rate	Weight	Price (Ebay)
Stryker XR1	Mechanical	Semi, burst, ramping, full auto	15+ bps	4lb	$200
Stormer Tactical	Mechanical	Semi	8+ bps	3 lb	$169.95
Model 98	Mechanical	Semi	8+ bps	2.9lb	$72.75

Table 21: Comparison of paintball guns

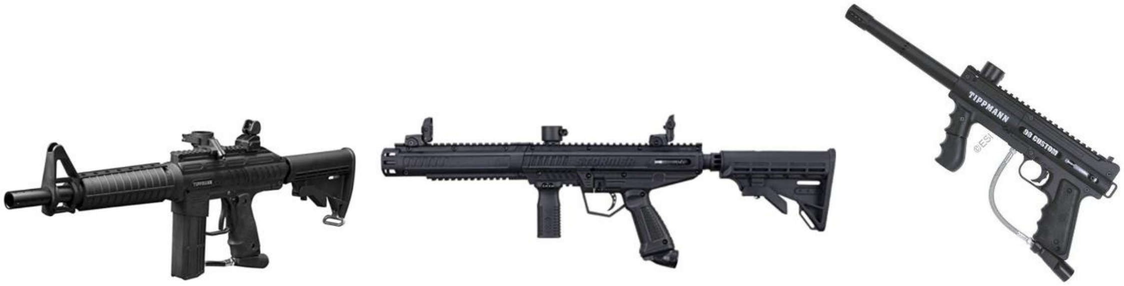


Figure 22: From left to right: Tippman Stryker XR1, Tippman Stormer Tactical .68 Caliber, and Tippman Model 98 Paintball Gun.

Figure 22: Tippman Stryker XR1, Tippman Stormer Tactical .68 Caliber, and Tippman Model 98 Paintball Gun, depicted below are images or the top three candidates identified for inclusion within the sentry turret project. As indicated, the Tippman Model 98 Paintball Gun was the final selection for use with the sentry turret project, as it suited the needs of Group 33 best among the options presented. Many paintball guns are very costly so we have decided to buy a pre-owned model that works. This will help us stay within budget and keep the selection of our hit-indication device. Some of the preowned models have been customized, so when we receive our device, it will need to be compared to what it should be and if any adjustments need to be made toto structure to accommodate for an unexpected customization, it will need to be made preferably before the structure is manufactured. If we are to receive a model that has been modified in a way that would cause it to function differently than how we expect it to function then we will with either need to make counter adjustments or adjust another aspect of our design and if too many modifications have been made our worst case scenario would be having to return and order

a new one from a different seller, which is easily done given how abundant these guns are in resale.

4.9 Warning Light Selection

Before our turret fires upon a target for marking, we will have a method of alerting a target. We have decided to use a bright warning light so it catches the attention of a person who may have accidentally wandering into the path of firing. We have chosen three different options of lights to choose from. We settled upon choosing colors of red and amber because those are universally use as hazard and stop light. The colors would send a message to the target that something is wrong, and they need to move away.

4.9.1 AgriEyes Amber Beacon Light

This is our first choice for a warning light. This light has a permanent mounting system with a screw. There are other methods of attachment such as pipe if there is a circular opening or magnet, but they cost more and are not preferable methods of attachment. It is waterproof and can be used night or day in rain, snow or fog. It has seven different flash modes that can be programmed either at fast or slow speeds. It is recommended at 12-24 V and uses 30 high intensity LEDs. The wiring has 3 wires red, black, and yellow. The red and black and positive and negative respectively while the yellow is for switching modes. It comes at a price of $26.99 which is pricey, but it does offer more features than a standard LED light.

4.9.2 Industrial Warning Safety Flashing Beacon

Our second choice comes in a red color that has many different features. It also has a permanent screw mount which comes with a mounting base. The light structure is made up of 15 strobe LEDs. It is waterproof and can be used outside. The most interesting feature is that it comes with a customizable speaker that plays a warning message. The sound is 105db which is quite loud, but it can be adjusted. A downside to this product is that the instructions are not in English, but it can be easily installed. There are four wires red, black, green and yellow. Red and black are standard positive and negative. Green wire is for the speaker and yellow for the different modes. This warning light is sold at a price of $29.99 and uses 12-24V.

4.9.3 Bolt Beam 12mm LED Light

Our final choice is vastly different from the previous model as it comes as a small LED, but very bright. It comes in various colors such as red, amber, green, and blue, but it does not contain any extra features. It is waterproof and the screw design on the back makes it easy for attaching and removing for projects. Compared to the previous lights this one is far easier to integrate into a microcontroller. It contains only red and black wires. It contains three small LEDs in a glass housing and requires 9-14.5V. This light is very cost effective as it comes in at a price of $2.95.

4.9.4 Final Warning Light Selection

All of the lights had great features and would work well with our project, but with the budget in mind, and this part not affecting the primary function of the prototype, the best option for us would be Bolt Beam 12mm LED Light. With this light being very affordable we could buy a couple a program the lights to flash at different intervals to make our own warning system. It can be easily installed and has a lower power consumption compared to the other two models. In **Table 22** below you can see a comparison of the factors we took into consideration when making our final choice.

Warning Light	Voltage used	Features	Wire Installation	LEDs contained	Cost
AgriEyes Amber Beacon Light	12-24V	Seven different flashing modes	Red, black, yellow	30	$26.99
Industrial Warning Safety Flashing Beacon	12-24V	• 4 different flashing/strobing modes • Plays an audio warning	Red, black, yellow green	15	$29.99
Bolt Beam 12mm LED Light	9-14.5V	None	Red and black	3	$2.95

Table 22: Comparison of warning lights

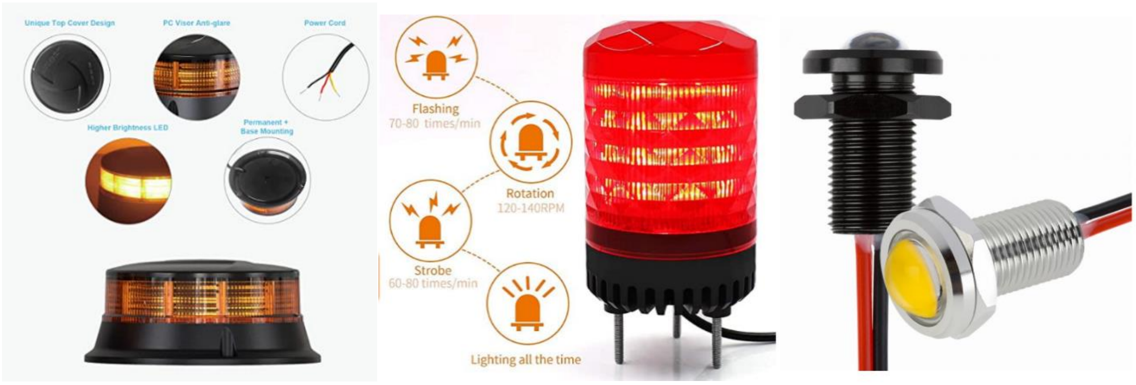


Figure 23: From left to right: AngriEyes Amber Beacon Light, Industrial Warning Safety Flashing Beacon, and Figure: Bolt Beam 12mm LED Light.**

4.10 Laptop Selection

The computer vision software has to run on a laptop, which is connected to a camera and microcontroller by USB. The computer has to be able to run OpenCV with Python and be able to connect to the PCB which houses our microcontroller. The factor in our final choice will be processing power. **Table 23** outlines the specs of the laptops that are available to use from group members.

The only concern with the laptop that we have chosen, from the available group members, Is that the battery may not be what it was when it was purchased this can be amended by just using a portable laptop battery. We considered trying to add a plug from our battery supply to go to the computer however due to the size and the restrictions we'd need the laptop would most likely pull more from the battery than we'd want. hence if we do run into an issue with the laptop battery, we will instead add a portable laptop battery specifically made for that on to the laptop such that it'll still be able to run on its own battery and have an extended lifetime. if we were to have a larger budget for the system, we would be able to incorporate a mid-range durable tablet to attach onto the system for the user interface, but due to budget constraints we will simply use the laptop with a portable battery.

Laptop	CPU	OS	RAM	Ports	Weight	Battery
Michael's Lenovo Thinkpad E570	Intel Core i5 – 7200U (2 cores @ 2.5-3.1 GHz)	Windows 10	16GB	3 USB Ports; 1HDMI port	2.3kg	~1 hour
Liderma's Predator PH315-52	Intel Core i7-9750H (6 cores @ 2.6-4.5 GHz)	Windows 10	16GB	1x USB Type-C 2x USB Type-A 3.2 Gen 2 (10 Gbps) 1x USB Type-A 2.0 1 HDMI 2.0	2.3kg	~1.5-2 hours
Kaitlyn's 2017 HP Notebook	Intel Core i7	Windows 10	8GB	3 USB Ports; 1HDMI port	2.1kg	~2 hours

Table 23: Laptop Comparisons

The chosen laptop is the Predator PH315-52. This one was chosen because it has the best processor and more than enough RAM, which will hopefully minimize computation times for running the code. This choice also has enough USB ports for us to be able to plug in the camera and PCB to the computer. The battery life is also about what we need to be, for our purposes as well.

6.0 Research

This section will go into prototypes that have already been created. While many systems have been created in recent years, expanding on previous designs and creating more advanced and improved systems with new approaches will allow this project to go above and beyond.

6.1 Existing Products

When searching for non-lethal turrets available for private use that could be purchased, not much was available. One product that was similar to this project comes from a company called Sublethal. Their product is a remotely accessible gun, as shown in **Figure 24**, on a rotating stand with a camera. Their system uses a civilian paintball gun with nylon bullets.

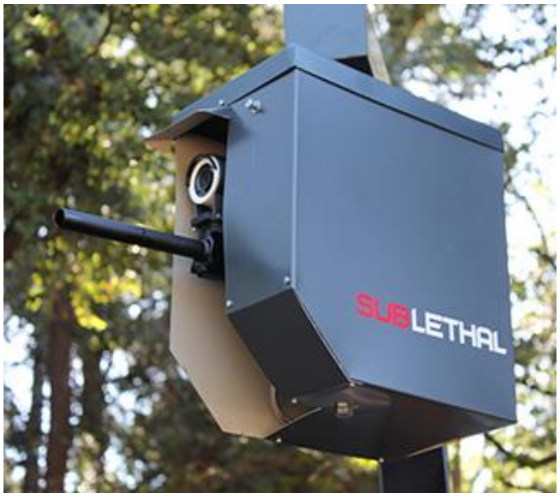


Figure 24: Sublethal's remote accessible gun

They chose to not use paintballs as they are more likely to cause blockages in the system. However, in the case of this project nylon balls will do far more damage than what the aim of this project hopes to achieve. Past this product, not much could be found in terms of purchasable products the majority of the inspiration for this project comes from online hobbyists documenting their own systems.

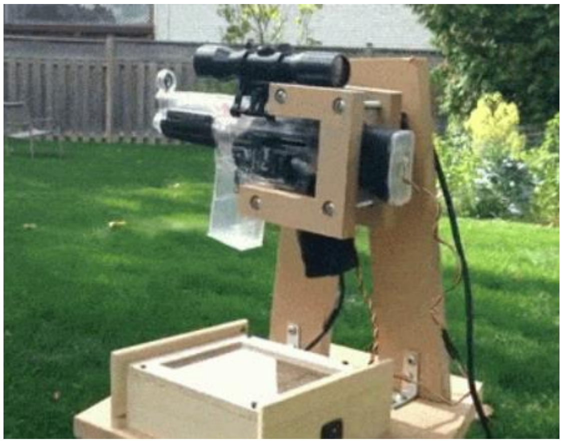


Figure 25: James Hobson's airsoft sentry

One such product comes from a website called Hackaday where James Hobson created an airsoft sentry gun, as seen in **Figure 25**, that was, in theory, to be used for home security. This project was developed over 5 years ago now and there is much that can be improved upon specifically the targeting system. However, his system was inspired by another system called Project Sentry Gun. That is a longstanding project which aims to provide a basis for people to create their own sentry projects.

Project Sentry Gun has developed a product called the Gladiator II, as can be seen in **Figure 26**, that they sell that uses a paintball gun and is marketed towards paintball teams. One interesting feature of their product is that it has a color that disables the targeting system, so if someone is wearing the color of choice then the system will not target them. However, their system is very old, and the setup, while durable, is very bulky.

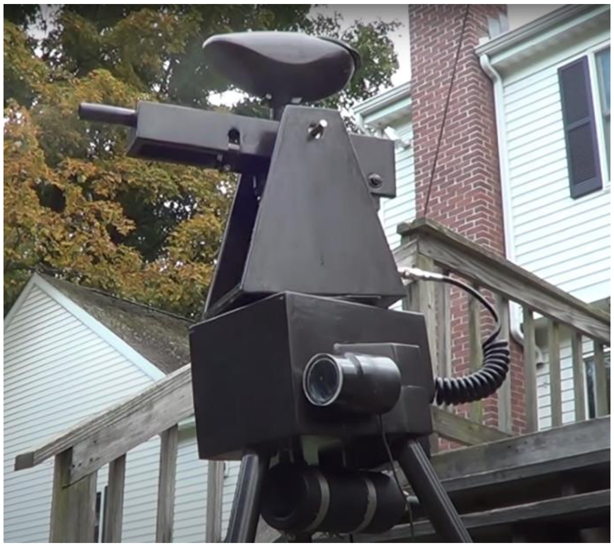


Figure 26: The Gladiator II

6.2 Market Analysis

The Motion-Detecting Sentry has a wide range of possible groups that can be marketed to. The main group being those that want to use this product for recreational purposes. This product will be perfectly outfitted for paintball tournaments as each unit can be configured to target certain colors, or in this case jerseys or paintball vests, making it perfect for games of paintball where each team would have use of one unit to add to the field. This product could also be modified to use a laser rather than a paintball gun, making it perfect to be used in laser tag games as well.

Another possible market could include those that could use it as a deterrent for home defense in less stable parts of the world. As can be seen from the company Sublethal there is a market for this type of product in South Africa, but their product is much more damaging than what we have designed, therefore our product would be much more applicable in environments where people are more looking for something to deter possible threats than do more serious damage. That being said due to the nature of our design it would be simple to modify it to use a different type of ammunition to fit a customer's needs.

The final market that we will consider is the possibility of this product being able to be used in more criminal situations. The main course of thinking is for it to be used similar to how fire alarm pull units will splatter ink onto the hand of the person who pulled the alarm in the case that the person pulled it without the correct intent. So, the system would be implemented in places where if someone were to try to trespass or steal the person would be marked with brightly colored paintballs that would make them easily identifiable.

For this product we have identified multiple groups of people that we could likely market to with success. So, while this is a slightly niche and odd product there are groups of people that would be interested and willing to purchase such a product.

7.0 Design Constraints and Standards

This section covers the constraints and standards that will affect the design of this project. Standards are documents that establish uniform engineering methods and practices. Standards will play an important role in the design and manufacturing components of this project. The constraints relating to this project will primarily focus on the ethical and safety concerns related to the sentry given the legality of having an automatic targeting system on even a nonlethal weapon.

7.1 Related Standards

This section will focus on explaining the industry standards that will be applicable to our project. Standards are considered critical in the world of engineering design, as they allow us to design efficiently, stay in line with other new technological designs and practices. Standards will also help us to create something reliable and safe for whomever the end user may be.

7.1.1 Programming Language – Python

The Python programming language has not yet been standardized by an international or national standard, even though it is one of the most popular programming languages in the world. The closest thing to a standard that Python has its syntax and the Python Enhancement Proposal 8 (or PEP 8) style guide.

Python syntax, like any other syntax, is mandatory; otherwise, the code simply will not function.

PEP 8's aim is to bring all Python together under one style, increasing readability and overall understanding of Python code, but isn't meant to be followed in every circumstance. For this project, following PEP 8 is important, since multiple group members will be working on the code. If PEP 8 is adhered to, on member can make a contribution, and another can easily understand that contribution, and quickly add upon that contribution without conflicts. The PEP 8 style concerns Code layout, whitespace, trailing commas, comments (which are particularly important for making code understandable by others), naming conventions, and strings.

7.1.3 PCB Design

https://www.ipc.org/TOC/IPC-2221A.pdf

https://www.protoexpress.com/blog/ipc-2221-circuit-board-design/

For our PCB design we will be following the IPC-2221 standard. This standard covers acceptable circuit board design, interconnections and how to correctly mount components. The most significant topics covered in this standard include how to properly space conductors and how large the traces on the board should be.

When placing conductors on the PCB the distance between the two components would need to be spaced a certain distance. The way these components are spaced are based on two measurements, clearance and creepage, which can be seen on the figures below. Ideally the space between the conductors will be as much as possible without becoming redundant. These are defined in international standards IEC 950 and EN 60950.

Figure 27 shows the clearance between two traces. This is determined by the peak value of the DC voltage. The clearance between traces is also adjust to account for dry pollution and condensation

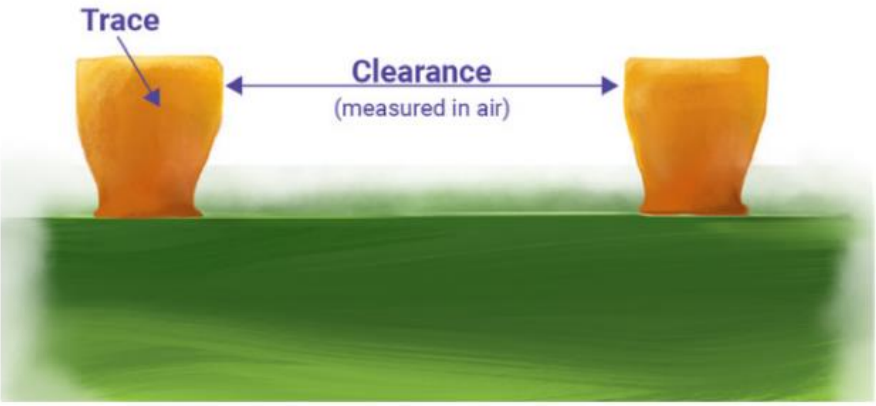


Figure 27: Clearance between traces

Figure 28 below shows an acceptable amount of creepage between two traces. This is just an overview as the exact amount of creepage needs to be determined. This is determined by the root-mean-square, RMS value of the AC voltage. Creepage distances are also adjusted to ensure PCB safety standards. This is includes protection from dry pollution and condensation in the air.

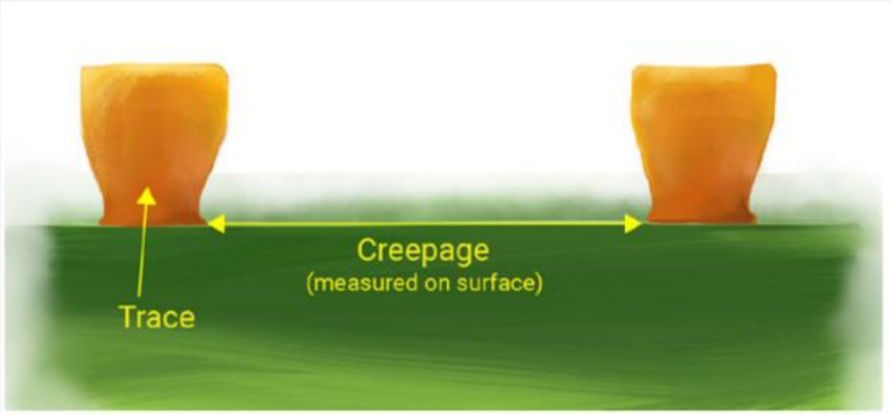


Figure 28: Creepage between traces

There are also standards for varying degrees of pollution called IEC 60947-1 which has four categories.

Pollution degree 1: No dry pollution, similar to the environment in a sealed room.

Pollution degree 2: Non-conductive pollution with a possibility of temporary conductive pollution from condensation. The area in a laboratory is an example.

Pollution degree 3: Conductive pollution or contamination occurs because of humidity or dust. This is seen in industrial environments.

Pollution degree 4: Persistent conductivity because of excessive humidity and dust. This is due to either rain or snow.

For determining the size of the traces on the board we will need to calculate what the maximum amount of current going through them will be, we will also need to look at the signal characteristics and the allowable temperature, these are included in IPC-2141 and IPC-4562. In a perfect case the traces will be just big enough to not burn out as the current moves through it. A trace needs to have an appropriate amount of thickness as shown in **Figure 29 (below)**.

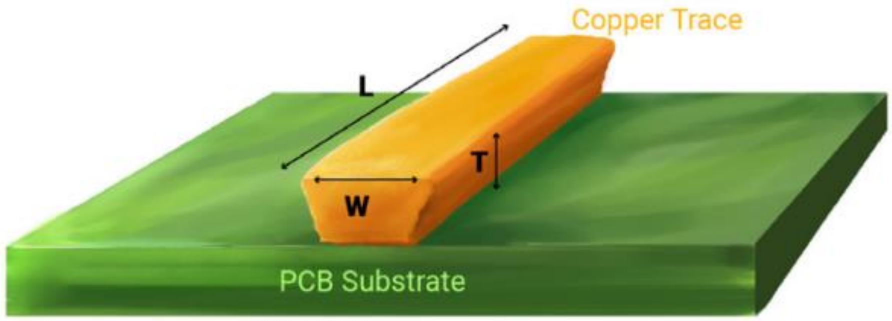


Figure 29: Thickness of a trace

The final part of this standard that we will talk about is insulation resistance tests. This will help us to protect the PCB against short circuits if any contact between the conductors does happen to occur. The insulation resistance test is when a voltage is applied to the PCB to allow the current to be measured to determine the resistance of the board's insulation. Another test used to determine if the PCB has a high enough insulation is done by again applying a voltage, this time much higher, and the leakage current is measured using a HiPot tester, and if the insulation can withstand the high voltage, then the insulation will be considered strong enough.

7.1.4 Motors

NEMA is an abbreviation for the National Electrical Manufacturers Association. Although based in the United States of America, this is actually an international standards committee, although being American the specifications were all originally created using the imperial system instead of the metric system. The NEMA standard ensures that replacement parts will fit and be readily available, and that motors from different machines will be interchangeable. By utilizing NEMA standard motors, we will in effect by following this standard.

In 1984 the NEMA committee set out some standards for motor sizes, based upon the faceplate size of the motor. This standard is still in use today and results in motors designated "NEMA 17" or "NEMA 23". These designations can cause confusion at times as the name only really concerns the size of the motor, and not its other specifications such as voltage, current, step angle, or even if it's bipolar or unipolar.

The primary NEMA number specifies the position and size of the mounting face. For example, the "17" in "Nema 17" would indicate the motor has a faceplate approximately 1.7 inches wide. The full NEMA standard also describes other features of the motor and is written as: NEMA *DDMMLLL-CCCIVVVSSSW.*

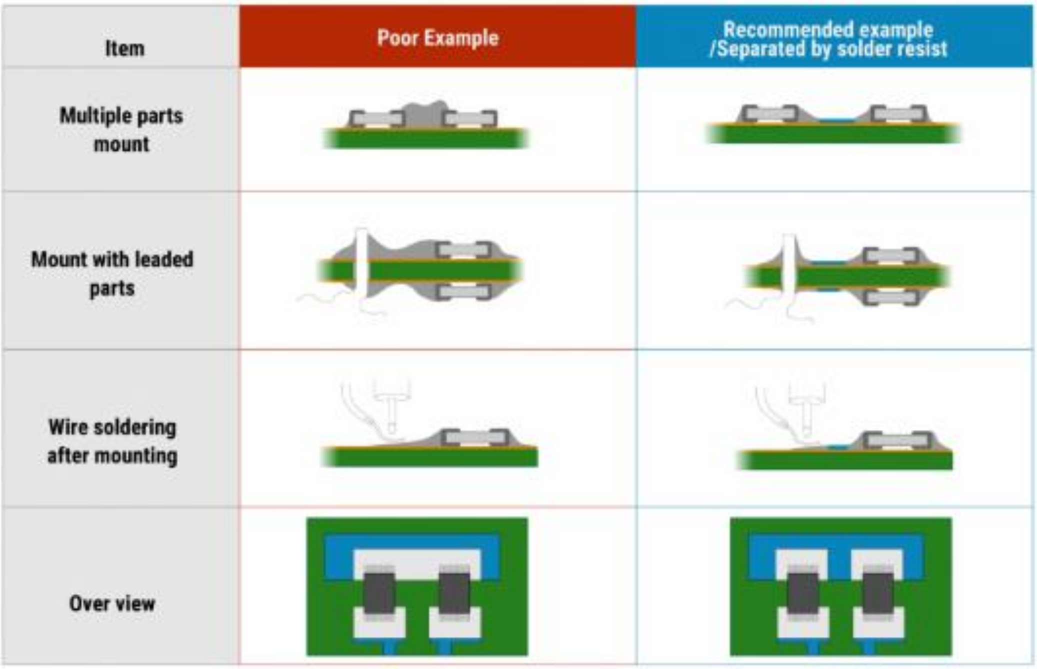
- *DD* is diameter (inches x 10). For square stepper motors, the length of a side is used instead of the diameter.
- *MM* is mount type code (inches x 10), including none, one, or both of these letter codes
    - "C" if there are holes tapped in the face of the motor
    - "D" if there is a flange on the back end of the motor with slots missed for bolts to pass through
- *LLL* is length (inches x 10)
- The "-" at this point separates mounting information from electric characteristics
- *CCC* is phase current (amps x 10)
- *I* is insulation class and defines the maximum allowable operating temperature
    - Class A is 221 Fahrenheit
    - Class B is 266 Fahrenheit
    - Class F is 311 Fahrenheit
    - Class H is 356 Fahrenheit
- *VVV* is phase voltage (rating x 10)
- *SSS* is steps per revolution

- *W* is a winding code
    - A is two wires
    - B is 3
    - C is 4
    - D is 5
    - E is 6
    - F is 8

For example, a 3.4” diameter stepper motor with a flange that is 1.6 inches long, has a phase current of 1.2 Amps, class B insulation, 5.3 phase voltage, 200 steps per revolution, and 8 wires is: NEMA 34D016-012B053200F.

7.1.5 Soldering

https://www.protoexpress.com/blog/ipc-j-std-001-standard-soldering-requirements/
Given that the design of this system will require us to have components to be soldered we must consider the necessary standards in place. Standards such as IPC J-STD-001 and IPC-A-610 have been developed to specify what the industry requires to be sufficient. These standards are issued by IPC, a trade association that deals with standardizing the assembly and production of electronic equipment. The IPC also accredited by the American National Standards Institute (ANSI).



Item	Poor Example	Recommended example /Separated by solder resist
Multiple parts mount		
Mount with leaded parts		
Wire soldering after mounting		
Over view		

Figure 30: Mounting examples

IPC J-STD-001 or Joint industry standard been used globally for soldered electronics and electronic assembly. In 1992 it was released as J-STD-001 A and has been continuously revised and improved to the current version of J-STD-001 H. This standard focuses on specifications on electronics and electrical assemblies. It is split into three classes focusing on manufacturability, performance requirements, process control regulations, and verification testing.

- Class 1: General electronic products
- Class 2: Service electronic products
- Class 3: High-performance electronic products

These classes focus on the highest quality of assembly and have a precedence for best soldering practices. This also takes safety and environmental conditions into consideration. Firstly, cleanliness must be practiced in order to prevent any contamination of materials, tools, and surfaces. Solder can either be lead or lead-free, and these practices are important to prevent any possible lead contamination. Second heating and cooling rates need to be equal. This should be according to the manufacturer's instructions. This will avoid ruining any components due to temperature. Third, wires are not to be damaged. To avoid this solder needs to wet the tinned part of the wire. Fourth, cleanliness needs to be continuously checked before applying conformal coating and stacking. This will help with avoiding any contamination.

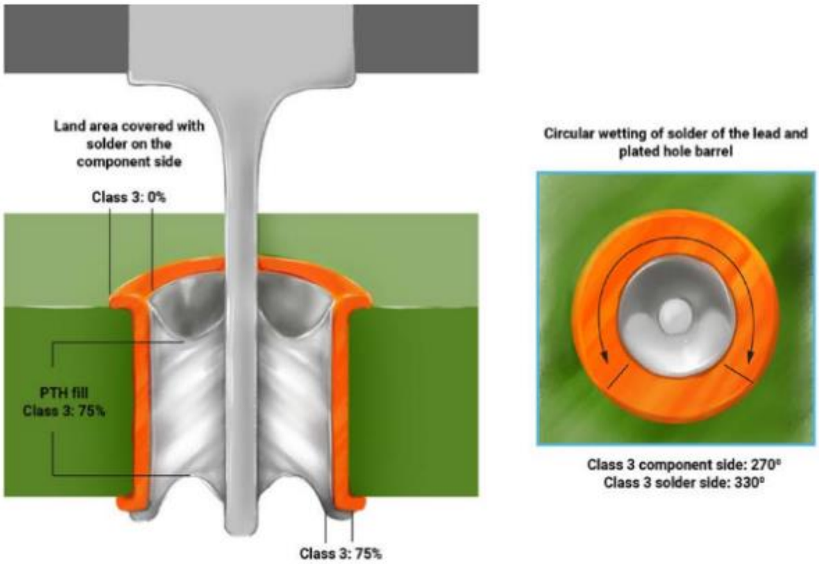

Land area covered with solder on the component side
Class 3: 0%
PTH fill
Class 3: 75%
Class 3: 75%
Circular wetting of solder of the lead and plated hole barrel
Class 3 component side: 270°
Class 3 solder side: 330°

Figure 31: Through hole soldering example

7.1.6 Battery

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6762806

The power supply of our project will be the Duracell Ultra WKDC12-55P Deep Cycle AGM SLA battery. An important standard for batteries is "IEEE Recommended Practice for Installation Design and Installation of Valve-regulated Lead-Acid Batteries for Stationary Applications", or IEEE Standard 1187-2013. This standard is a revision of IEEE standard 1187-2002. As its title implies, this standard's purpose is to recommend proper installation and design procedures for users of VRLA batteries. Battery sizing, maintenance, capacity testing, charging equipment, battery protection, and monitoring are all beyond the scope of this standard.

The standard outlines installation design criteria that we can use for the integration of the Duracell battery into our turret. General considerations are as follows:

- Space allocated for the battery and associated equipment should allow for present and future needs
- Floor loading capabilities of the battery location should be established
- Load limitations of transport equipment and access routes should be considered
- The location should be as free from vibration as practical
- The general battery area should be clean, dry, and ventilated. Provide adequate space and illumination for inspection, maintenance, testing, and cell/battery replacement.
- Spill containment is not necessary for VRLA battery installations
- For personnel safety, fresh portable eye wash devices should be readily accessible when handling and connecting batteries
- Provisions for the safe handling and recycling of VRLA batteries should be in accordance with environmental regulations.
- For smaller installations, portable lighting might be necessary to provide adequate illumination. To the extent possible, lighting fixtures should be located to minimize the effects of debris falling onto the battery in the event of a luminaire failure. If batteries are installed in rows of open racks, lighting should be over the aisles, not over the batteries.
- The battery should be protected from direct sunlight to prevent case material degradation
- The battery should be protected from spot heating and cooling

Design for maintainability:

- A maintainable design is one in which the terminals of all cells/units rae accessible during normal float operation for periodic maintenance and interconnection resistance checks. Examples of less maintainable designs include uninterruptable power supply (UPS) systems in which each battery is sealed inside a cabinet without any provision for access or cells with fully insulated terminal covers that cannot be removed without also disconnecting the cell from the circuit.
- Whenever possible, avoid installations containing series-parallel connections within a string of cells because the voltage and internal ohmic measurements may not represent the actual condition of each cell.

Some of these recommendations are not as useful to us as others. Since we are only using two batteries, we likely won't need any transport equipment. We also won't need to follow the provisions concerning recycling of VRLA batteries, as we don't intend on completely using up the batteries' power.

In accordance with these standards, the batteries will be placed on the ground below the turret tripod in a well-ventilated space protected from the elements (such as rain). This space will be made easily accessible for maintenance.

7.1.7 Software Testing

https://standards.ieee.org/project/29119-1.html
Standard ISO/IEC/IEEE 29119 "define[s] a generic process model for software testing that can be used by any organization when performing any form of software testing". The standard is used as an aid to ensure proper test processes, documentation, and techniques are used. Standard 29119 is split into four sections, and we will be using three of them (because the first section, Concepts and Definitions, is not included in UCF's subscription). 29119-2 groups the testing activities that may be performed during the life cycle of a software system into three process groups. These three process groups are the organizational test process, test management processes, and dynamic test processes, as can be seen in the **Figure 32** below.

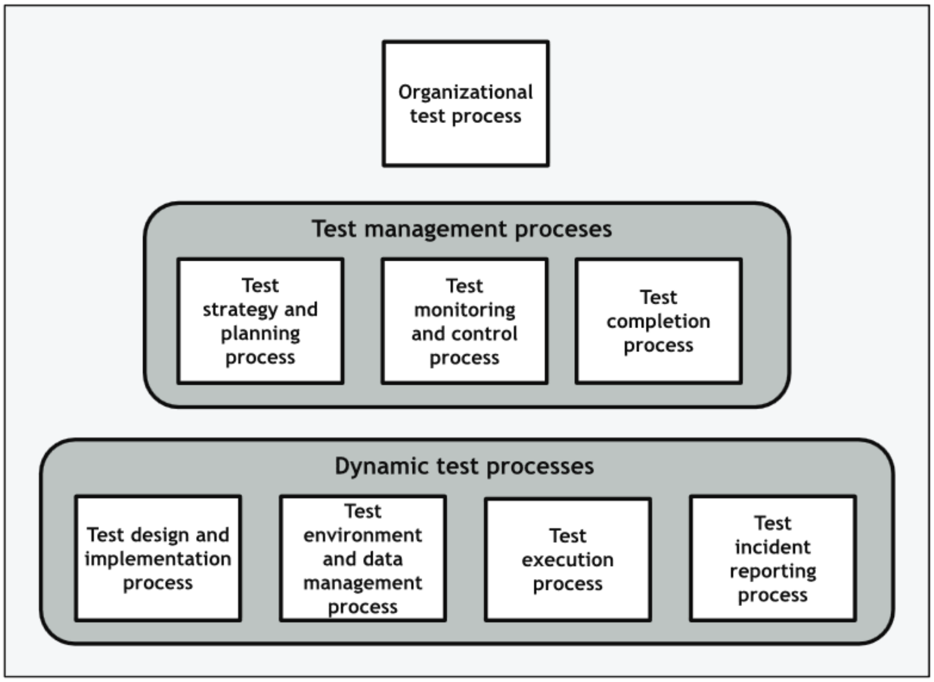

Organizational test process
Test management proceses
Test strategy and planning process
Test monitoring and control process
Test completion process
Dynamic test processes
Test design and implementation process
Test environment and data management process
Test execution process
Test incident reporting process

Figure 32: The multi-layer testing process

The organizational test process comprises activities for the creation, review, and maintenance of the organizational test specifications. It also covers the monitoring of organizational compliance with them. The results of the successful implementation of the organizational test process include:

- The requirements for organizational test specifications are identified.
- The organizational test specifications are developed.
- The organizational test specifications are agreed to by stakeholders.
- The organizational test specifications are made accessible.
- Conformance to the organizational test specifications is monitored.
- Updates to organizational test specifications are agreed to by stakeholders.
- Updates to the organizational test specifications are made

The test management processes, shown in **Figure 33** below, include test strategy and planning; test monitoring and control; and test completion. These generic test management processes may be applied at the project level, for test management at different test levels, and for managing various test types. When applied at the project test management level, these test management processes are used to manage the testing for the whole project, based on a project test plan. There may be multiple different test plans, and these test plans may be tested separately or combined into a single overall test. Either way, the following procedure will remain unchanged. After a test plan is established and testing begins, test monitoring and control scrutinizes whether testing progresses in accordance with the test plan and the organizational test specifications. If there are significant departures from planned progress, planned activities, or other aspects of the test plan, activities will be initiated to correct or compensate for the resultant variances. Once this is done, the test moves onto the test completion process. The test completion process archives test assets, cleans up the test environment, identifies lessons learned, and finally reports test completion.

Integrated into the test completion process are the dynamic test processes. The four dynamic test processes are: test design and implementation; test environment and data management; test execution; and test incident reporting. The role of these processes is to ensure the test completion process is carried out correctly, and that all contributing test members are aware of testing results with respect to the set guidelines of the process and final result.

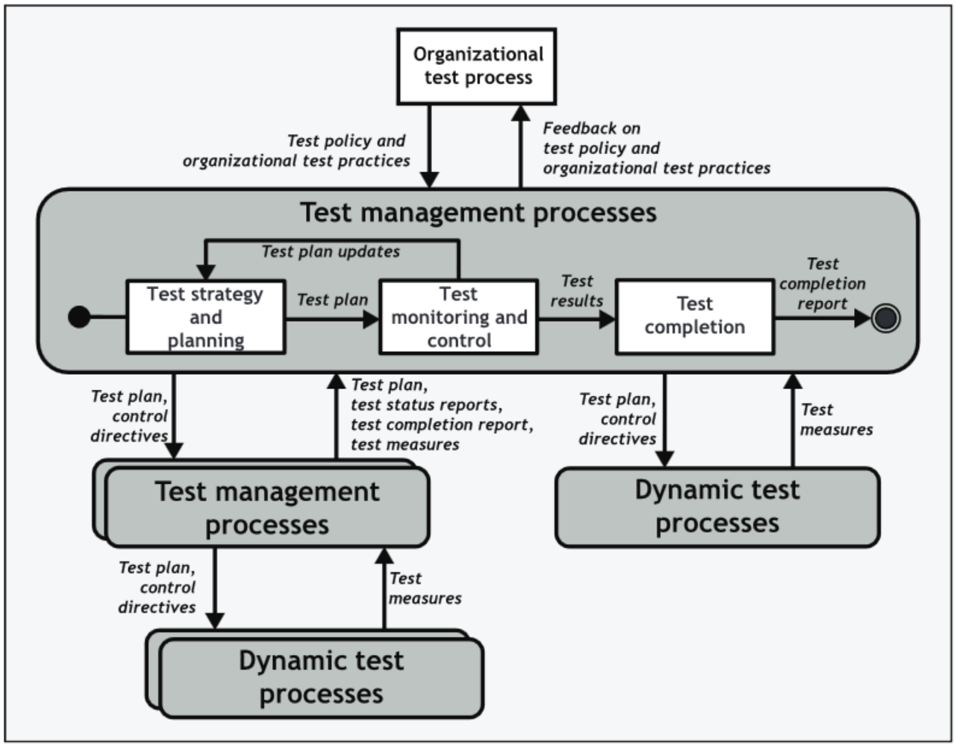

Organizational test process
Test policy and organizational test practices
Feedback on test policy and organizational test practices
Test management processes
Test plan updates
Test strategy and planning
Test plan
Test monitoring and control
Test results
Test completion
Test completion report
Test plan, control directives
Test plan, test status reports, test completion report, test measures
Test plan, control directives
Test measures
Test management processes
Dynamic test processes
Test plan, control directives
Test measures
Dynamic test processes

Figure 33: Test management process relationships

Standard 29119-4 concerns test techniques, defining “test design techniques that can be used during the test design and implementation process that is defined in ISO/IEC/IEEE 29919-2". Standard 29119-4 provides test design techniques for three separate classes of testing: specification-based testing, structure-based testing, and experience-based testing. Each of the three classes include multiple different possibilities that may be applied in order to complete testing. Each case is defined in terms of derivation of test conditions, test coverage items and test cases.

In specification-based testing, the test basis is used as the main source of information to design test cases. This test basis is gathered from the requirements, specifications, models, and user preferences. In structure-based testing, the structure of the test item is used as the primary source of information to design the test cases. In experience-based testing, the knowledge and experience of the tester is used as the primary source of information to design test cases. The three classes of test design techniques are complementary, and their combined application typically results in more effective testing.

By following standard ISO/IEC/IEEE 29119 and showing that all of the standard's requirements have been met, our group will be able to claim full conformance to this standard. However, not every part of this standard is necessary for the turret's software design. Therefore, we will be claiming tailored conformance by demonstrating that the requirements chosen by our group have been met.

7.2 Design Constraints

The following will outline constraints for our project. One of the biggest constraints will be budget. We will not be able to afford the newest technologies on the market that would be optimal for our prototype. Time will also be another important constraint as we only have a semester to test our prototype and have it working for our panel to critique. With our prototype using computer vision to track people we must consider ethical, social, and political constraints. People could be against our idea or think there is not a need for it. Environmental, health and safety constraints will be a concern as it will fire on people. Manufacturing is a big concern as there are shortages of materials as many companies were impacted by Covid-19. All these will be addressed in the following sections.

7.2.1 Economic Constraints

The most limiting constraint for the team would be the budget. For our original budget it was agreed that everyone would contribute $100 towards the project. Due to this, much of our design is focused around keeping the cost of materials low. With a higher budget this project could be greatly improved but keeping to the initial $400 has caused us to consider alternative options. Choosing used and older model paintball gun was one of the first cost reductions considered. By choosing an older used model allowed us to save over $150 while still getting a comparable unit. Another tradeoff occurred when choosing motors, rather than using motors with smaller steps, to keep costs down we instead choose to use a more cost-effective option and including a gear reduction, which will still cost less than the more expensive options. Similarly, the battery was also a component that needed through research into as it can be very easy to spend the majority of the budget on one. Meaning that we would have to carefully determine exactly the load that is going to be required from the battery, which will allow us to pick the cheapest and best option, saving up to $100 on the project budget.

As simple as the components list is for this project one of the more difficult constraints is being able to keep the system outside, and because of this many of the components will require additional water-resistant housings. Given that the components and housings will still cost less than the water-resistant variations of the components and any premanufactured housings. But having to make these accommodations will put more strain on the budget. Therefore, by buying less weather resistant components that can be modified to fit our circumstances will allow us to more easily stay within the intended budget.

Another issue our group is facing is the recent chip shortage, starting in the spring of 2020 and continuing through to the writing of this paper, the winter of 2021. Experts expect the chip shortage to continue throughout 2022, which means it will be inescapable throughout the construction of the turret. The chip shortage was initially started by the shutdown of multiple factories due to the COVID-19 pandemic, coupled with the ever-rising demand of computer chips. Supply chain issues faced by both the United States (where we are developing this project) and China (where most of these computer chips are coming from) have only exacerbated the issue. As a result, the prices of many components have increased, and our budget is further restrained.

Overall, so long as each component of the system is carefully specified and chosen keeping relatively close to the budget will be manageable. Based on the projections for this project none of the tradeoffs will cause issues with the final systems.

7.2.2 Time Constraints

The timeframe of this project is extremely important due to how tight the deadlines are. The project has a total of 8 months to complete, this timeframe includes everything from the design process to the manufacturing and assembling the final product. Time was a large part of determining the scale of the project and the features that will be implemented into the product. In the case of the Motion-Detecting Sentry some of the additional features that will require extra time and research include having a GUI on the associated computer, implementing any presets for the end user or having a customized and compact gun housing. These features, among others, will be on a secondary list that will be reassessed once the primary objectives have been implemented and perfected. This being the case if an additional feature is going to be considered for implementation, then we will also have to take into account how much time is left in the overarching timeline and if that will be enough time to not only complete it but also have the time to order and get in any additional components. When we design the PCB, we will consider the additional features and if those features will need components on PCB so that in the worst-case scenario, we will have unused ports or components on the PCB rather than having to give up on implementing a feature simply because reordering a new PCB would likely be unachievable. Another concern for us will be ordering the PCB and all of the necessary electrical components. Ideally by the time we have to order all the components, everything will be available. In order to give us the best chance of having all the electronics arrive on time ordering the PCB will be done at the start of senior design 2, before almost anything else is begun.

To help us understand the time that will be needed for each prioritized feature and additional feature, a list of how long it will take for each feature to be implemented will be kept helping better schedule out overarching timelines as well as help each team member better budget their time and stay on track by being able to prioritize everything they will need to do, allowing us to meet every deadline. For any additional features we will also set maximum amounts of time that will be allocated to those so that if a certain feature is not working and taking up too much time then we will have a good idea of where to stop and move on to other higher priority tasks. This will prevent us from wasting time on a feature that will not be able to be implemented by the final submission. We will also allocate a fair bit of time to testing the final system and working out any issues on the computer side, which can be done once the hardware and physical system is manufactured.

Due to the tight timeline and having a hard deadline at the end of senior design 2, being able to stay on schedule and meet every deadline will be one of our highest priorities. Any deadline missed could put the team farther behind than just that deadline as it would cause us to miss future deadlines.

7.2.3 Ethical, Social and Political Constraints

When it comes to automated weapons platforms such as this turret, there are a lot of ethical, social, and political issues to consider. The primary of these is the amount of force used by the turret to deter trespassers. There is a fine distinction between “Lethal”, “Less than Lethal” and “Non-Lethal” weapons, and this distinction is controversial.

A lethal weapon attempts to defeat an adversary’s ability to resist, while a less-than-lethal device attempts to defeat the adversary’s will to resist. Ability to resist is quite measurable and concrete – death or severe injury is usually the final end to any ability to resist. However, an adversary’s will to resist is intangible, and defies measurement. An adversary may surrender if a non-lethal device was fired and missed, or they may continue to attack with complete disregard for personal injury.

Lethal weapons are defined by their capability to cause death, while non-lethal devices are defined by their intent not to cause death. However, “nonlethal” is a term that has grown to be rejected by military and law enforcement institutions, as it is misleading to the lethal capabilities of these weapons. For example, a taser is popularly considered “non-lethal”, when it is quite capable lethal harm. A 2014 study published in the American Heart Association’s journal *Circulation* found that tasers can cause cardiac arrest and even death.

(https://www.ahajournals.org/doi/full/10.1161/CIRCULATIONAHA.113.005504).

When determining the legality of owning a turret, nonlethal or otherwise, the laws tend to vary greatly. In the United States especially, the laws concerning using something of this nature can be complex. If this product were to be hidden on a property, then it would be wholly illegal but if the product were to be used in either a non-civilian scenario or a scenario where everyone involved was fully aware of the device then it would be acceptable for use.

An AI shouldn’t be given a lethal weapon due to the threat of a software bug causing the turret to unleash lethal force against innocent bystanders. Less than lethal is also considered off-limits for this project, as it is a demonstration for a college course, and not a real military or law enforcement tool.

To completely avoid causing any serious injury to anyone, we are building the turret with a completely nonlethal weapon – a paintball gun. The only way this weapon could cause serious injury is if it were directly fired into the eyes of the target, and to assist in avoiding this, the robot is being programmed to fire at the target's center-of-mass. Theoretically, the paintball gun and the accompanying trigger mechanism could simply be replaced with a "less than lethal" option, such as a taser or riot shotgun, but in the interest of everyone's safety, we will be using the paintball gun.

7.2.4 Environmental, Health and Safety Constraints

For our project the environment will play a big role in how our system runs given the turret will primarily be outside for most cases. Thus, we will need to be diligent in making sure the system will be able to be outside in high humidity for long periods of time and that all the electronics will have proper water-resistant casings to protect them from any rain or bad weather. The system will not be able to be left out in any heavy weather conditions given that this will be a mostly open system that will not be fully waterproof to the extent that it will be able to hold up to any heavy downpours.

The computer vision program will be carefully set to target the midsection of the target person and aim well below the face and neck. In order to make sure that the system will never hit anyone in such a way that it will cause serious harm, the system will be rigorously stress-tested until we can be sure that it will never accidentally aim at or above a person's neck.

7.2.5 Manufacturability Constraints

During the design process for this project, we had to keep in mind the availability of the necessary electronic components and confirming at the time of design that the required components are available and have a large amount in stock, meaning if there are less than a few hundred in stock then it will be easy to assume that the parts may not be available when we go to order the PCB and the electronic components.

Another constraint we need to address is if we have all the necessary tools to construct the structure of the project. In our case we have all of the tools necessary to manufacture everything either through the school or our personal tools. To fully assemble our project, we will need to be able to cut and weld together the metal stand, which requires either a grinder or a chop saw to accurately cut the metal bars at the correct angles, as well as a welder to assemble the base. For the PCB we will need a soldering iron. And finally for the acrylic housing we will need a laser cutter. Another manufacturability constraint is that of the cost of having materials, in our case metal, can get quite costly in a limited budget such as ours.

7.2.6 Testing and Presentation Constraints

For the Motion-Detecting Sentry being able to test and present this project will be quite difficult given the fact that we will need both ample room and permission to use an automated paintball gun with automatic target detection. Testing the robot will also require an area that supplies contrast between targets and their background for the human detection algorithm. This means that the testing area will require ample (and preferably uniform) lighting.

In order to perform adequate testing, we will seek permission from a landowner or the university to test our system in a large open space where the chance of anyone being in the way will be minimal. If it is an area with the possibility of pedestrians, we purchase caution tape to make a barricade. This will help block off the area we will need for testing and alert any pedestrians nearby for safety measures.

If we are required to have a working model at the senior design showcase, then we will unload the paintballs and remove the compressed air from the gun to eliminate any actual possibility of the paintball gun being able to actually fire. We will instead attach a laser pointer to the barrel of the gun to show where it will point. This will allow us to show the targeting system in a crowded area without needing people to all be fully dressed in protective gear around the gun.

8.0 Design

In the following sections, the members of Group 33 will outline the design features and functions which may be implemented for the Motion-Detecting Sentry Turret Project. These design ideas will be the layout of our initial ideas and specifications of the prototype, as well as many of components needed to for it to function properly.

8.1 Design Overview

This section will go into all the details of the design for the Motion-Detecting Sentry Turret Project. This project can be broken down into two major sections. The sections are the software portion of the design and the hardware portion. The software section will detail how the project will identify and target humanoid entities within range of the device. That section will also detail the exact functions of the program and the algorithms used. The hardware section will discuss every aspect of the printed circuit board, as well as the motors used in the project, and overall structure of the sentry turret product. This section will detail the physical aspects of Group 33's prototype, such as the motion of the device and the reasoning for the layout and organization of parts.

The image below, **Figure 34: Initial Flow Diagram**, shows a diagram which outlines the flow of information and power throughout the systems implemented in the sentry turret project. The flow of the prototype will generally follow this model. The battery will be used to power the microcontroller and all of the other peripherals connected to it. The camera will be connected directly to the computer and receive power. The computer will handle the computer vision and through our algorithm it will communicate with the chip on our microcontroller. When a target is identified the process of activating the warning light and engaging the motors will begin.

The software part of our project will ultimately be the driving force for our computer vision. It will take input from our camera and process it using computer vision. From there, it identifies targets trained from the OpenCV library to mark. Then our written algorithm will move the paintball gun into firing position of the target. This is essentially telling the turret when to turn on the warning light, how to move, and when to fire.

The hardware side will focus on building all the physical components needed for the Motion Detecting Sentry. This will be part of our main structure and everything we will need for a physical build to power the prototype. The chip on the microcontroller will be the bridge to connection between software and hardware. This will allow the software to communicate with its physical components such as the warning light, motors, and turret. The battery will provide power to all these components by connection by the microcontroller.

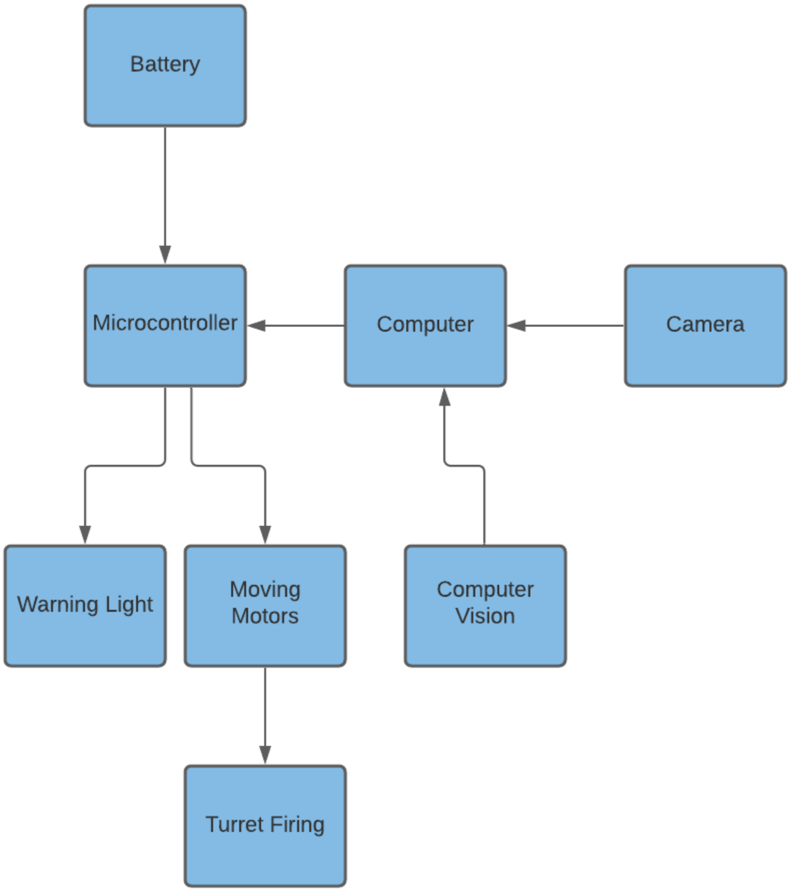


Figure 34: Initial Flow Diagram

8.2 Software Design

The software portion of the turret will be partly implemented on the ATMega328P microprocessor and partly implemented on the laptop which is connected to the camera and the microcontroller. The laptop's primary purpose will be to take a video stream as input from the camera, detect human targets in this video stream, and then calculate orders for the turret's motors which are relayed through the microcontroller. The microcontroller takes input from the laptop and outputs pulse width modulation signals to control pulses going to the motors. The program flows from the laptop to the microcontroller as seen in **Figures 35 and 36** below.

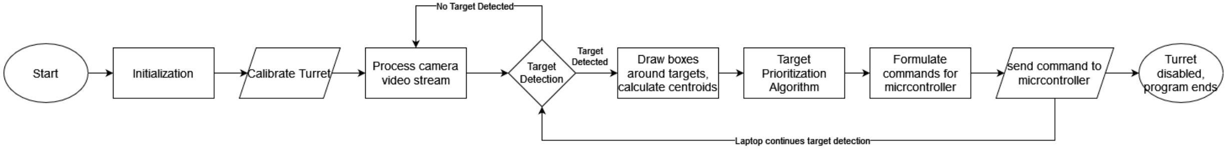

No Target Detected
Start
Initialization
Calibrate Turret
Process camera video stream
Target Detection
Target Detected
Draw boxes around targets, calculate centroids
Target Prioritization Algorithm
Formulate commands for micrcontroller
send command to micrcontroller
Turret disabled, program ends
Laptop continues target detection

Figure 35: Flowchart for Laptop Software

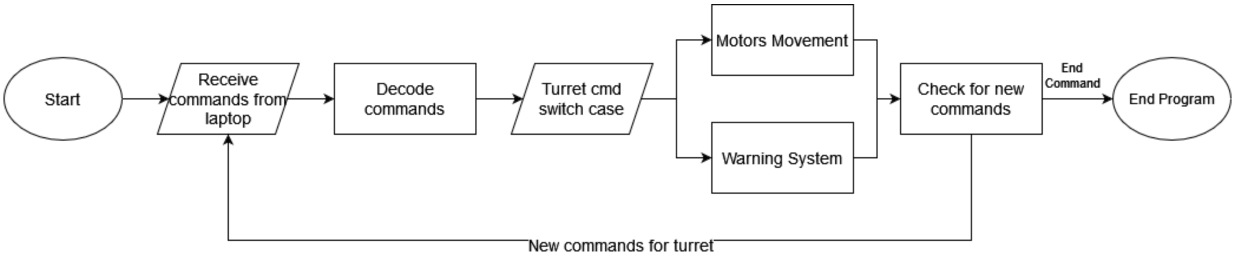


Figure 36: Flowchart for Microcontroller Software

It is somewhat difficult to plan out the entirety of the project's software without yet having started the prototyping process, and as such the software explained in this document is largely pseudo code which will likely vaguely resemble the software of the finished product.

8.2.1 Computer

OpenCV has a built-in method to detect pedestrians. It has a pre-trained **HOG** (Histogram of Oriented Gradients) and Linear SVM (Support Vector Machine) model to detect pedestrians in images and video streams.

Before doing anything else, the turret must be calibrated. This is done by through direct control over the turret, facilitated by the Interactive class. The turret is test fired, and then adjusted so that the turret is firing at the center of the camera's video feed. This becomes the "crosshair" of the turret. Instead of test firing, a laser pointer can also be attached to the turret gun with tape to see where the turret is pointing, and align it this way. This method would save on paint ball ammunition.

After calibration, our python program will first initialize the HOG descriptor/person detector. Then it will receive a video stream from the camera, captured frame-by-frame. These frames can be resized (e.g. to 640x480) and converted to greyscale for faster detection. Then, the program gets the bounding boxes of detected humans using the pre-trained HOG. These bounding boxes can be stored into a NumPy array. Each bounding box has x and y coordinates for its four corners. We can then derive the center of the target from these coordinates, and then calculate the required steps for our stepper motors to take to aim the turret gun at this point based on the turret's current position. This is done by calculating the distance between the target point and the crosshairs, and measuring the distance changed by a single step of the stepper motors.

These steps are relayed through a USB connection to the ATmega328 microcontroller, which coordinates the turret's stepper motors. The ATmega328 will be running its own code with the Arduino IDE to convert the input form the laptop into commands for the stepper motors themselves. The laptop converts the command into a binary number before sending it to the ATmega328, which decodes the number into commands for the turret's components.

Another possible method for target detection is background subtraction. This method requires a constant background, so it may run into difficulties where the turret's environment is changing. We grab the current frame, resize it, change it to greyscale, and then apply a bilateral filter. The bilateral filter reduces noise while retaining edges. We could also use a gaussian blur to reduce noise, but it does not preserve edges like the bilateral filter does, which can result in objects being more difficult to detect. However, the tradeoff is that Gaussian blurring is faster to compute than bilateral filtering. Bilateral filters use a nested for loop (for each pixel, look at every pixel), and are non-linear, depending on image content. Both factors slow it down significantly. Fortunately, there is a way to get a fast approximation of a bilateral filter, although it does result in a loss of accuracy. Further testing is required to determine which noise reduction method is most efficient for our computer vision algorithm.

Next, we apply the cv2.absdiff method, which subtracts the current frame from the previous frame. This leaves us with an image containing the positions the foreground object had over time.

First apply a morphological operation. Here we will first dilate and then erode the image which is called a “close” operation. We shall then threshold the image to make the object more prominent. To remove the small objects in the image we'll proceed with a median filter which helps to remove 'salt and pepper' noise.

Next, we find the contours in the object using cv2.findcontours. This outlines the shape of the detected movement, which we then draw a bounding box around using cv2.boundingRect().

Like the other method, we use this bounding box and its coordinates to calculate the x and y steps we need to undertake to move the turret’s gun to the target. This part of the process can be multithreaded for greater efficiency – one thread for the x and y dimension each.

One we have calculated that the turret is within a certain small number of steps from the target’s x & y coordinates, we can send the order to fire to the microcontroller, which relays this order to the servo motor controlling the paintball gun’s trigger.

How to account for movement?

The turret will be constantly comparing where the gun is currently aimed, and where its target is, and then moving the turret to aim at the target. When the target moves, the turret will follow. The turret’s ability to keep up with moving targets relies on the efficiency of the computer vision algorithm (e.g. how many frames the algorithm can process per second). The faster the algorithm can detect a target and calculate the necessary movements to aim at this target, the more accurately the turret can fire upon fast moving targets. However, since we are always calculating the current frame received from the camera, and that the program & the communication between devices is not instantaneous, we will always be lagging behind by a little bit. Hopefully, we can make the software efficient enough that the lag is small enough for the turret to still reliably hit the target. The only other solution is to estimate the target’s future position and aim the turret accordingly.

We could use a Kalman Filter to follow and predict a target’s path to lead them. This requires that the targets are moving at a constant velocity or constant acceleration, which may not always be the case. A Kalman Filter would also require a significant amount of training data in order for it to predict the movement of targets reliably. There are no pre-trained Kalman Filters available for our project, and we do not have the time to go through the process of training a Kalman Filter.

Another possible solution is to calculate the target's velocity and acceleration, and then calculate the turret's required position to hit the target while keeping the traverse time of the turret itself in mind. This would be difficult and run into efficiency/speed problems as the lead would have to be recalculated every single frame.

We also need to calibrate the turret's position in relation to the camera and stepper motors before starting the program, so that when the algorithm has to calculate the necessary steps to aim the turret at a detected target, it will do so accurately.

The final hurdle is how to handle multiple targets. There are several methods for target acquisition in this case.

1. Fire at the target closest to the gun's crosshairs. We simply calculate the distance from the gun to the centroids of all detected targets, and then command the turret to fire at the closest one. To avoid a problem similar to method 2, a buffer can be used to make the turret "stick" to one target for a short period, even if relative distances change.
2. Fire at the target closest to the turret itself. This is more difficult as the turret has no reliable way to measure distance on the z axis. Without purchasing an extra part like an ultrasonic sensor to detect distance, the closest target can be estimated by bounding box size. A target closer to the turret will have a larger bounding box, therefore the target with the largest bounding box can be marked as the closest and fired upon. This method does have a serious problem. The turret does not have an insignificant traversal time as it switches from one target to another. Therefore, two targets can take turns stepping towards and away from the turret, causing it to constantly switch between the two. The turret will be too busy switching targets to actually fire on either.
3. First in, first out targeting. The turret will prioritize targets based on the order of detection. The turret fires on the first target it sees until that target retreats, and then moves to the next. If somebody is willing to be shot at repeatedly, or is wearing protective equipment, then targets detected after them can act with impunity as long as the first target distracts the turret.

Method 1, firing at the target closes to the turret's crosshairs, is the most efficient option available.

8.2.2 Python Pseudo Code

Python pseudo code:
Main:

```
        Beginning of the program

        Calls Turret's calibration & initialization functions, then starts target detection
```

Raw_mode:

```
    Function for taking in user input through the keyboard
Class VideoUtils:
    Helper functions for video utilities
    Live_video:
        captures live video from the camera, frame by frame. Display to the laptop monitor.
    Find_target:
            Our target detection algorithm goes here.
            Prioritize_target:
                Decide which target to prioritize according to the algorithm decided upon earlier
Class Turret:
    Initialize:
        set up stepper motors. Set their speed, current x & y steps, and their ports/pins
    Calibrate:
        calibrate x:
            user calibrates the tilt of the gun so it is level. (w) moves up, (s) moves down, (enter) to finish.
        calibrate y:
            User directly calibrates the yaw of the gun so it aligns with the camera. (a) moves left, (d) moves right, (enter) to finish
    Target_detection:
        VideoUtils.find_target(self.move_axis)
    Move_axis:
        (x, y, w, h) = bounding rectangle found from find_target
        Calculate center of target
        Calculate movement necessary from crosshair (found in calibration) to target
        Move_x:
        Add move left, right, or neither to command
        Move_y:
        Add move up, down, or neither to command
        If within acceptable distance to target:
        Add fire gun to command
        (the acceptable distance has to be found through trial and error testing)
        Send command
    Interactive:
        Interactive mode for troubleshooting. Pivot with (a) and (d), tilt with (w) and (s), fire with (enter), exit with (e)
        While loop that reads keyboard input and if-else statement transforms input into commands for the motors/gun using the fire & move functions below
    Deactivate:
        turns off motors, good for automatically disabling the motors on shut down
```

A class diagram of the above pseudo-code is provided below in **Figure 37.**

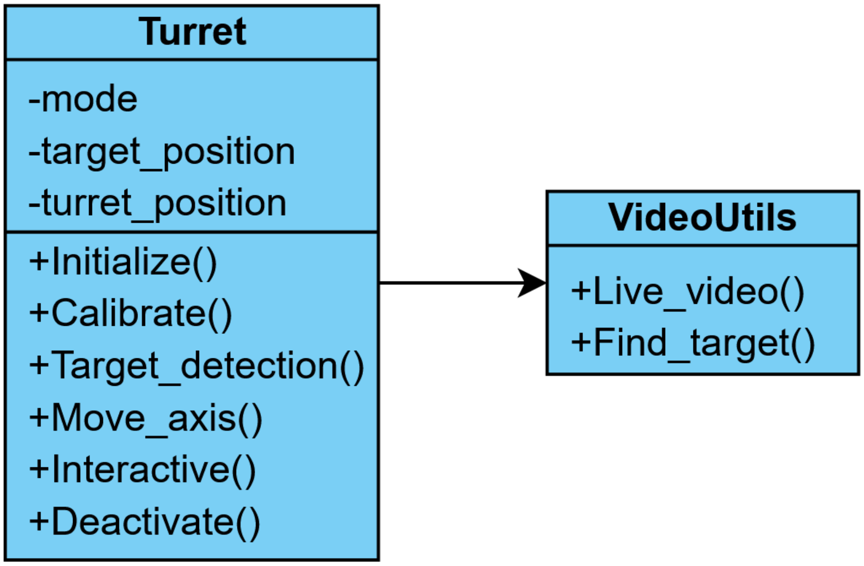


Figure 37: Class Diagram of Python pseudo-code

8.2.3 Microcontroller

The software on the microcontroller will be relatively simple and will be programmed using the Arduino IDE. However, getting an ATMega328 microprocessor with the Arduino

IDE pre-installed is quite difficult, especially since we are using a surface-mounted chip on a PCB rather than a separate chip on its own. There are two methods of burning the Arduino bootloader to the PCB: either through a USB to Serial/TTL adapter or using another Arduino board as an in-system programmer. To use the Arduino board as an in-system programmer, we just have to connect the right pins to each other and upload the code from the programmer board to the ATMega328 microcontroller. Since this process can be time consuming, we will likely be using the Arduino development board for prototyping the software, and then upload the finished code to the PCB. Our laptop will communicate with the microcontroller through a USB 2.0 port. The USB data will be converted to serial with a F232RL chip.

After setting up the microcontroller's pins, inputs, and outputs, the microcontroller will start communication with the laptop. The microcontroller is kept in a low power mode until the motion detector is triggered. When the motion detector is triggered, the microcontroller will check its connection with the laptop for new commands. These commands are then decoded from binary byte by byte, being converted into integers. These integers are our commands for the turret, interpreted through a series of if statements.

There is no real communication from the microcontroller to the laptop. One might think we would use the motion detector to interrupt/halt the computer vision program on the laptop as well the program on the microcontroller, but we do not. The motion detector sends no messages to the laptop through the microcontroller. Instead, as long as the motion detector hasn't been triggered, the microcontroller is placed into low power mode and ignores commands from the laptop. After the motion detector is triggered, the microcontroller enters into high power mode, and a timer begins. Once in high power mode, the microcontroller begins taking commands from the laptop again. If the turret goes five minutes without detecting any targets with its computer vision algorithm (and not detecting anything with the motion detector), then the turret will go back into low power mode.

There has been some consideration of removing the motion detector from the turret altogether. This is because its range may be too limited for the turret to be useful. The PIR sensor's limited range of 30 feet may never actually catch any targets, resulting in the turret permanently remaining in low power mode and generally being useless. If the motion detector feature does turn out to be a failure, the manner in which the software was designed will make the remove of this feature painless. The motion detector feature is basically limited to a single condition check in a while statement that contains the primary code loop of the microcontroller's program. Removing the motion detector will just require changing this while loop to an ordinary loop statement.

8.2.4 Arduino Pseudo Code

Arduino Pseudo Code:

```
Setup
        Assign the pins for these functions/parts:
                Pan motor
                Tilt motor
                Trigger servo
                Warning light
                Motion Sensor
        Set up inputs and outputs
Start communication with the laptop
If (motion detector triggered)
        Loop
        Check for new commands
        Read first bytes in buffer, checking for the indicator for the start of a message
        Once a message is detected, read the message byte by byte
        Then decode the message bytes into integers
        These integers are our commands for the turret, which are interpreted through a series of if statements or a switch case. For example, if the fire integer is 1, then we fire the turret. If it isn’t, we don’t fire.
        If powered off or reset, break
```

8.3 Hardware Design

The hardware components for this project will be comprised of a PCB, camera, warning light and battery. The camera itself will be directly communicating with the computer to relay the images that will need to be processed by the computer vision software. The battery will serve to power stepper motors and the PCB will be powered by the computer via the USB connection. In the following subsections each section of the PCB will be discussed and the connections. In this section it will also be discussed how the microcontroller will be powered.

8.3.1.1 PCB Block Diagram

The PCB contains the connections necessary to allow the microcontroller to communicate with the computer via USB and then relay the coordinates derived by the computer vision software and send it to the stepper motor drivers which in turn will direct the stepper motors to the correct placement. Pictured in **Figure 38** below, is the PCB Block Diagram. It shows how the system of our prototype is supposed to flow. Our input will first be obtained from computer vision for image processing. Then when it has successfully identified a target, the microcontroller will then be alerted to turn on the warning light and activate the motors. While this whole process takes place the battery will be supply power to the warning light and microcontroller.

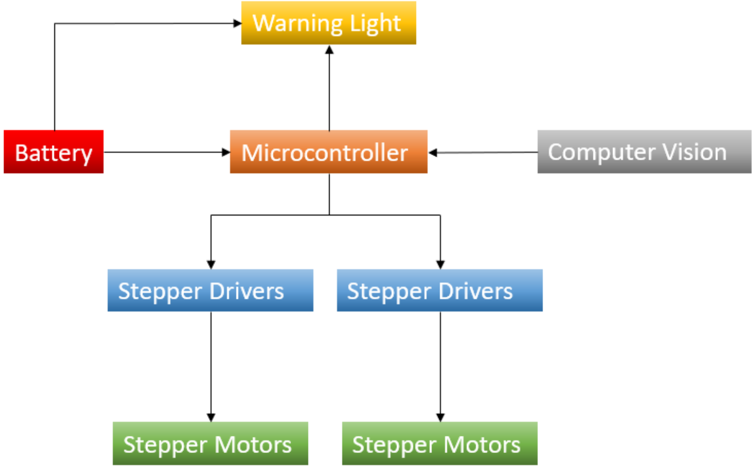


Figure 38: PCB Block Diagram

Pictured below in **Figure 39**, is the total schematic of what will be included on the final printed board. As we continue to design and test into senior design 2 the schematic will change as we add and shift what the board will need to include and how it will need to function as we come across more issues. It shows the connections for our motors, microcontroller, voltage regulators, and serial to USB. Going further into the paper there will be sections that will show individual schematics of the parts mentioned.

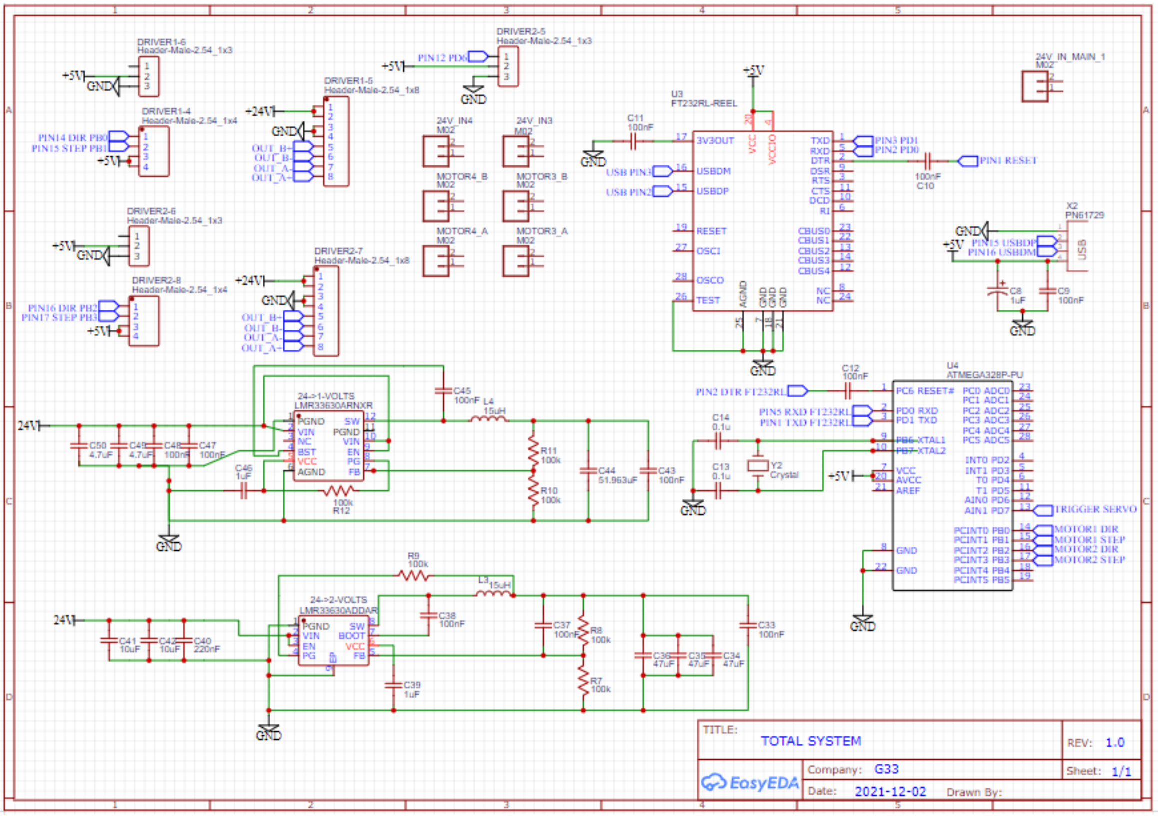


Figure 39: PCB Schematic

8.3.1.2 Microcontroller

The microcontroller chosen for this project was the AtMega328P. For the hardware side of the microcontroller, it was the best choice, given both the sheer amount of data available about the microchip, as well as the ability to use the Arduino environment to assist with prototyping and troubleshooting. This microcontroller was chosen also because of the ease of being able to control both stepper motors and the servo with a single microcontroller.

From the diagram below in **Figure 40**, it has a 28-Pin structure that will fit the needs of connecting all of our other peripherals. We will need to have drivers for our 3 motors as well as configurations for our warning light. It will be programmed using a pre-owned Arduino Uno, but we will be integrating the chip onto our PCB to make a custom microcontroller. This way we can have a better understanding of how a microcontroller communicates with hardware and software.

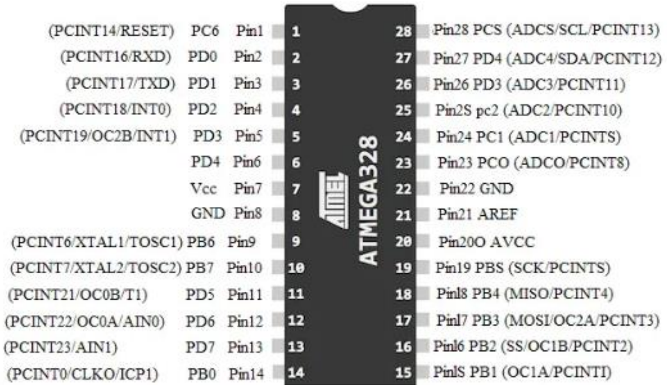

(PCINT14/RESET) PC6 Pin1 1
(PCINT16/RXD) PD0 Pin2 2
(PCINT17/TXD) PD1 Pin3 3
(PCINT18/INT0) PD2 Pin4 4
(PCINT19/OC2B/INT1) PD3 Pin5 5
PD4 Pin6 6
Vcc Pin7 7
GND Pin8 8
(PCINT6/XTAL1/TOSC1) PB6 Pin9 9
(PCINT7/XTAL2/TOSC2) PB7 Pin10 10
(PCINT21/OC0B/T1) PD5 Pin11 11
(PCINT22/OC0A/AIN0) PD6 Pin12 12
(PCINT23/AIN1) PD7 Pin13 13
(PCINT0/CLKO/ICP1) PB0 Pin14 14
28 Pin28 PCS (ADCS/SCL/PCINT13)
27 Pin27 PD4 (ADC4/SDA/PCINT12)
26 Pin26 PD3 (ADC3/PCINT11)
25 Pin2S pc2 (ADC2/PCINT10)
24 Pin24 PC1 (ADC1/PCINTS)
23 Pin23 PCO (ADCO/PCINT8)
22 Pin22 GND
21 Pin21 AREF
20 Pin20O AVCC
19 Pin19 PBS (SCK/PCINTS)
18 Pinl8 PB4 (MISO/PCINT4)
17 Pinl7 PB3 (MOSI/OC2A/PCINT3)
16 Pinl6 PB2 (SS/OC1B/PCINT2)
15 PinlS PB1 (OC1A/PCINTI)
ATMEGA328

Figure 40: ATMEGA328P Pins

8.3.1.3 Microcontroller to USB

In order to convey the information provided by the computer to the microcontroller a serial to USB converter was implemented. The FT232RL chip is going to be used as it is inexpensive and commonly used with this series of microcontrollers. This chip will be added to the PCB such that we can have a cost effective and reliable method for computer to microcontroller communication. Below in **Figure 41**, shows the footprint of the FT23RL with connections.

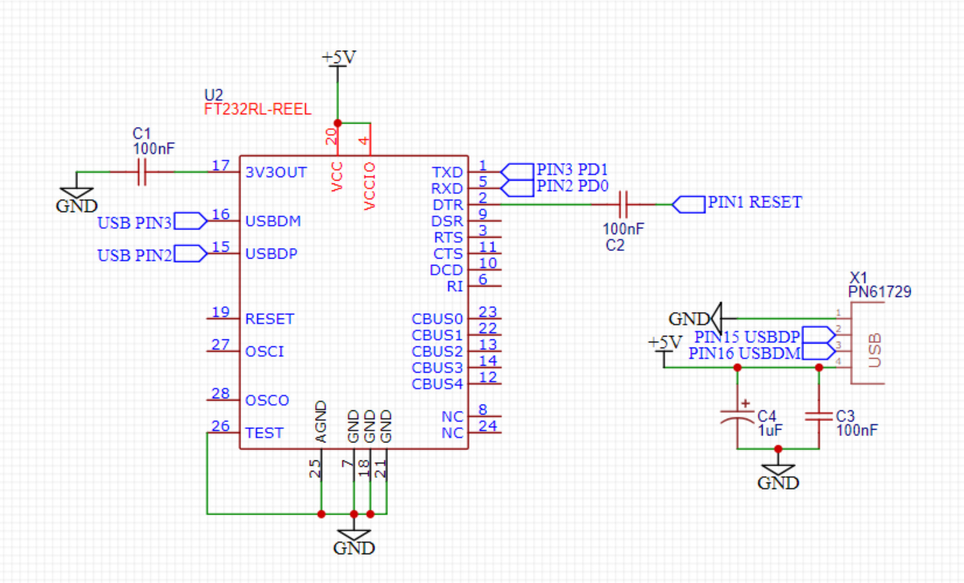


Figure 41: Serial to USB Schematic

8.3.1.4 Prototype Build

Once the motors and drivers are ordered the PCB will have been implemented on the breadboard. This will begin our initial phase of testing our parts. Once we have determined all the parts are working our first build will begin. The camera will be first hooked up to our laptop (computer) to test the OpenCV algorithm. Once we have determined a satisfactory percentage of detection, we will then move onto programming the microcontroller chip.

We will first start with building the top of the structure of MDF. This part will sit on top of the tripod and will stabilize and balance our paintball gun. This part must be built first as we need to test if it can hold the weight of the attached paintball gun, the servo and stepper motors, and the CO2 tank for firing. Once we have tested the durability of the MDF we move onto building the tripod.

Next work can be started on the tripod and the base where the top half will be secured onto. This will be a difficult task to build as it will require welding 3 metal poles with equivalent distance from each other. Also, we must factor the cost of getting our sheet of metal cut for the top of the tripod. An alternative option would be to price a pre-made tripod to fit our needs, but this is solely dependent on the budget and the products available to us.

The top structure will then be secured onto the tripod, and then be ready to be hooked up to the additional electrical components. Our PCB will be hooked up to our motors, warning light, computer through a USB cable, and most importantly, our battery. The PCB and all the wiring will be contained in housing for protection that will be placed under the tripod for accessibility and testing. The camera housing will be positioned and clamped directly under the paintball gun.

When all the components are secured and bolted then we can commence stress testing. First the computer vision will be tested again to adjust to the new position of the camera. It will be adjusted until the computer vision is satisfactory. The communication between the microcontroller chip and computer can now be tested. This will determine if the code is satisfactory or needs to be reworked. The firing mechanism and the movement can also be tested at this time. This will determine the amount of recoil the prototype will face, and if we will need to accommodate for more stability.

8.3.1.5 System Integration

For our prototype we have decided to migrate the microcontroller chip onto our PCB. This will help cut down on the budget, and also give us experience of designing our own microcontroller. The chip we will be integrating onto our PCB will be the ATmega328P. The chip will be bought unprogrammed and will need to have a bootloader installer. We will use a pre-owned Arduino Uno board as an In-System Programmer (ISP) to load the bootloader on the chip. This will enable us to upload our program onto the ATmega328P via the USB-to-serial converter.

On its own, the microcontroller is not able to run OpenCV, but it can interface via a written program. Our program will be written in the Python coding language that is capable of working with OpenCV. This code will handle the computer vision and decide what is a target. Through the computer working with OpenCV it can process the images captured by our camera. This will help our prototype detect targets and properly aim.

The program uploaded on the microcontroller chip will communicate with the code on the computer. This will tell our turret what to do. When OpenCV identifies a target, the computer will signal the microcontroller to turn on the warning light and after a few seconds if the target has not left the firing range, it will engage the motor to fire.

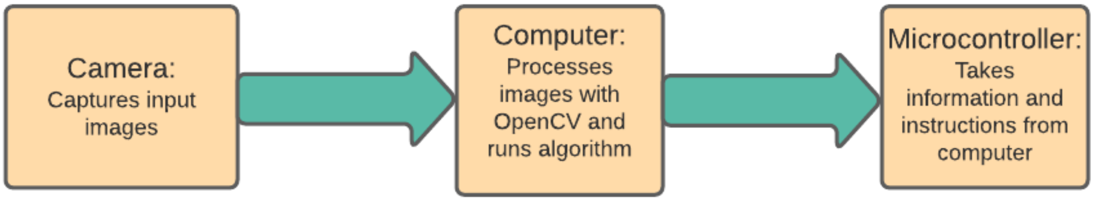

Camera:
Captures input images
Computer:
Processes images with OpenCV and runs algorithm
Microcontroller:
Takes information and instructions from computer

Figure 42: Integration Flow

Figure 42 shows the cascade of communication between camera, computer, and microcontroller. This solution is optimal when using OpenCV with Arduino microcontroller chips. The chip will be able to execute all the hardware functions we need for our prototype while communicating with the computer. Using OpenCV with a computer will provide superior processing power, so the combination of the chip interacting with the computer will cover both support for software and hardware functions.

8.3.2 Power Supply

For our power supply we will be using two 12-Volt batteries in series as shown in **Figure 43**, to reach the 24-Volts needed to run the stepper motors. In practice it would be better to have a single 24-Volt battery, for the simple reason weight reduction and packageability of the unit. However, due to budget constraints it was decided that for the project it would be best to use what was available to save on expenses. This battery will then be connected to the PCB and fed to the drivers and two voltage regulators. The battery will be connected to the PCB via screw terminal for easy connection and disconnection for transportation of the system. The first one will be to provide power to the microcontroller, the chip for the serial to USB converter, the logic chips of the stepper motor drivers and the servo motor, set to 5V. The second voltage regulator will be dropping the 24-Volt input down to 12-Volts to power the warning lights.

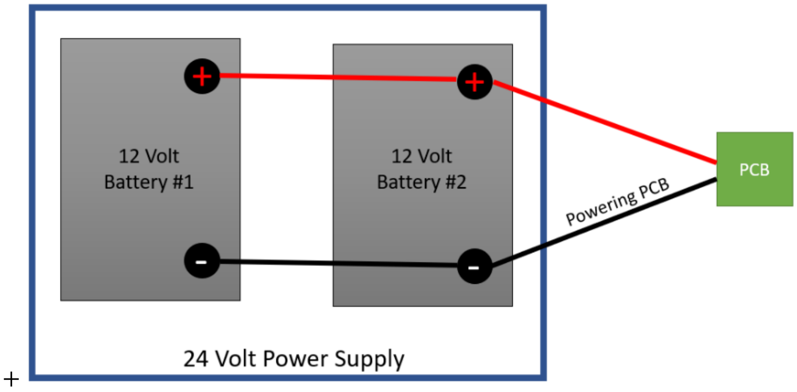


Figure 43: Batteries in Parallel

8.3.2.1 Voltage Regulation

We will be using two different voltage regulators to ensure that all the different components attached to the PCB receive the correct voltage at a constant rate. This area will be necessary for our system so that we don't overload the microcontroller and logic chips as that would lead to total system failure. In the **Figure 44,** below there are two separate voltage regulators. The first one is going to be explicitly used for the warning light system as they require 12 volts. While this voltage regulator will provide a steady 12 volts it won't need to be as accurate given that the warning lights have a much wider voltage range for fluctuations before failing. However, the second voltage regulator pictured below will be stepping 24 volts down to 5 volts, and this will need to be quite accurate as all of the logic chips and microcontrollers do not allow for much fluctuation in the voltage before failure.

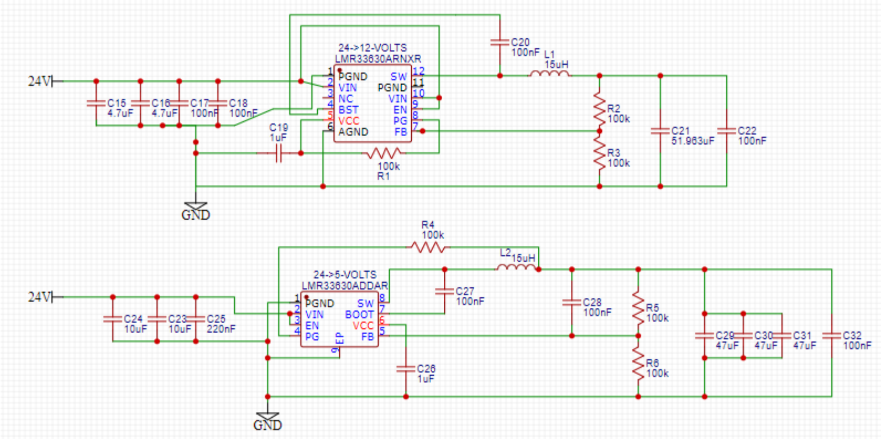


Figure 44: Voltage Regulation

The voltage regulators pictured above are variations of what were produced by the WEBENCH application provided by Texas Instruments. That software allowed us to easily find and choose which voltage regulator chips would work best for our project, as stepping from 24 to 5 volts is quite a large step.

8.3.2.2 Power Calculations

The power needed to run the system will come from multiple sources. The components that will require power include the computer running the computer vision, the camera, the PCB and microcontroller and the stepper motors. The camera will be directly connected to the computer that will be used to run the computer vision software; the computer being used for this project will be a laptop meaning it will be able to run off of its own battery for the 2-hour life of the system between charges. The camera, as it will need to feed images directly to the computer will be plugged into the computer via USB connection and because of this it will also pull from the computer's battery.

The next part of the system will be powered by the 24-Volt source. **Table 24** show all the components that will be supplied power from the 24-Volt source and the individual requirements. This will primarily be powering the stepper motors as these will require most power out of all the components. The drivers that will be used to control the stepper motors require an input voltage of 5-Volts. Finally, the Atmega328P also requires an input voltage of 5-Volts. Both the microcontroller and the drivers will be powered with an external 5-Volt battery and a voltage regulator will be used to make sure nothing goes awry.

Load on the 24-Volt Source	
Microcontroller	5V
Stepper Motors (2)	24V
Servo Motor	5V
Warning Light	12V
Stepper Motor Drivers	5V

Table 24: Voltage Requirements

8.3.3 Hardware Components and Implementation

For our PCB we will need to be able to connect the power supply to the motors and to the microcontroller and stepper motor drivers. Since our voltage source will be 24 volts, we will implement a voltage regulator to reduce the voltage down to 5V which will be sufficient to power the microcontroller and the stepper motor drivers. To connect our 24-volt power supply to our board I will be using screw terminal block connectors, these connectors will also be used to connect the 24-volt batteries to the motors themselves. We will also be running power through the PCB to the warning light system which will require 12-volts for which we will implement a second voltage divider to take the input voltage down from 24 to 12 volts. For attaching the microcontroller, because we will be burning the bootloader onto it separately, I am going to use 2 rows of 14 female pin headers as that will allow us to remove the microcontroller and manipulate it, if necessary.

The current generalized layout for our PCB is shown below in **Figure 45**. Once all of the components are tested together and working then the layout for our PCB will be finalized and sent out to be manufactured.

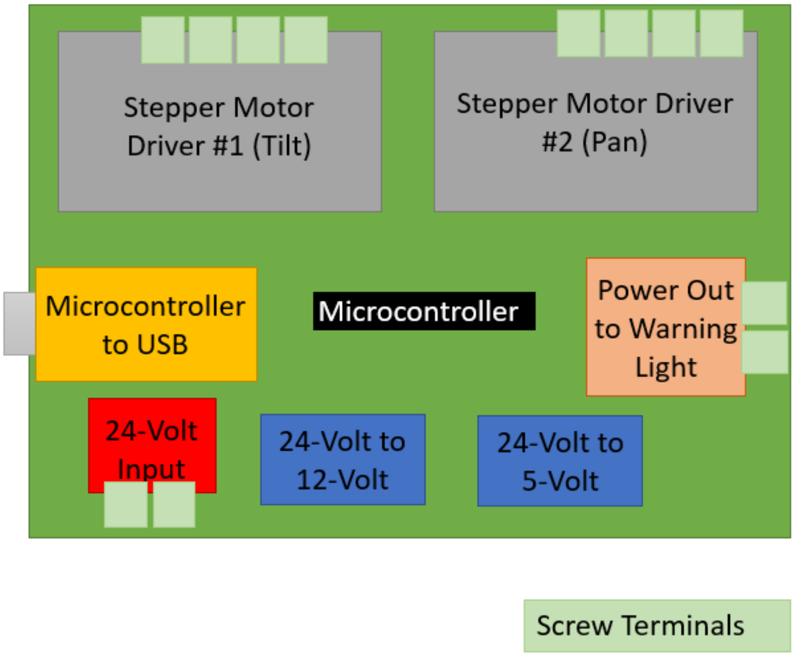


Figure 45: PCB Simplified Layout

8.4 Structure Design

This section will cover how the system will fit together and how each system will be housed and mounted. We will discuss the base of the structure, the mounting of the gun and camera, and the placement of the motors. This section will also discuss how the system will move along the two axes.

8.4.1 Base Design

The base of the structure is modeled such that it can be cut from either MDF or sheet metal. This allows for easier and more cost-effective prototyping. For the first complete build of our design, we will use quarter inch MDF and cut it using the laser cutter. Once the MDF model is successfully fitted and assembled, then it can be affirmed that the metal structure will work. Due to the cost of metal and needing to send it off to be laser cut will cause the base structure to be much more expensive than its MDF counterpart meaning that any mistake will be quite costly. **Figure 46** shows a complete proposed design of our prototype. The structure with the gun sits firmly atop the base for stabilization.

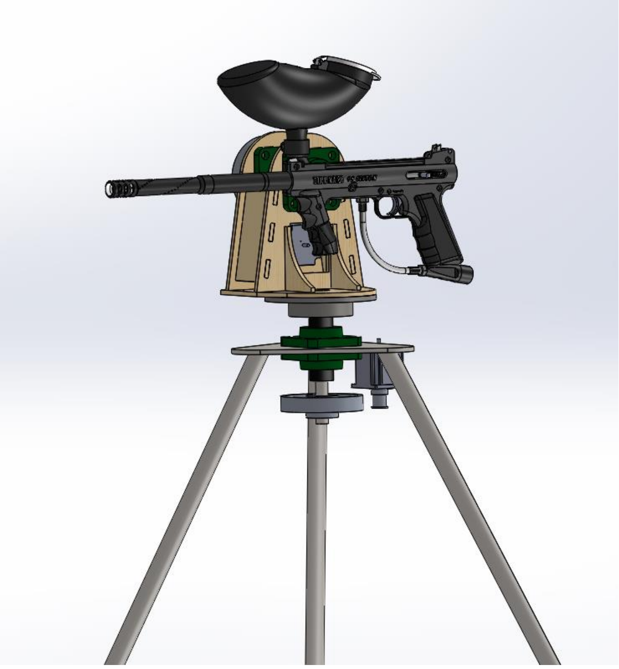


Figure 46: Base Design

8.4.2 Electrical Components Housing

In order to best contain the majority of the components on the structure it was determined that the best course of action was to either create or purchase a water resistant clear acrylic box to house the PCB, drivers and fan. Once the PCB is finalized and printed and the board can be fully assembled, and accurate measurements can be taken from there it will be determined if the prefabricated housings will suffice and if not then one can be sized and cut from a sheet of acrylic.

8.4.3 Camera Housing

Computer vision is going to be an integral part of the project. For this reason, we must carefully consider how to protect the camera while keeping it in a stable position. Our prototype will need to be in an outdoor setting so there is a possibility of it coming into contact with water. We have taken into consideration the camera and its electrical components must have some form of waterproofing. It must also be able to protect the camera from any dirt or debris that may affect the quality of image capturing.

To best hold the camera to the structure it will be housed separately from the rest of the electrical components. The biggest concern is the possibility of the camera getting wet. To best avoid this, we will be using a clear structure for the camera to mount in and still be able to utilize its function of image capture. A suitable material to cover the protection of the camera would be a clear acrylic paired with a 3-D printed casing, as shown below in **Figure 47**.

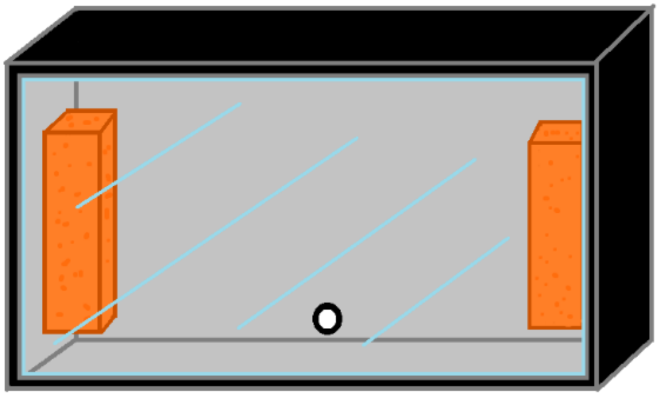


Figure 47: Camera Housing (May be later replaced with solidworks CAD model)

Acrylic is known to be lighter than glass and is very durable. This is particularly important as we want to keep the camera housing as light as possible. The front will also be transparent, so the camera is able to capture images without any assistance. The maintenance of it will also be very simple as we will need a microfiber cloth with soap and water to keep the acrylic surface free of debris and dust. The 3-D printed case will also be durable plastic and can be maintained with similar methods.

The camera will also be snugly fitted with foam on both sides, this will be from the foam padding that is adhered to the sides of the encloser. This will add a layer of protection and stability to the camera as we want to decrease as much motion and physical interference. The only area that will be exposed will be for the camera wiring that will be a small hole in the back (see **Figure 47**), surrounding the hole for the wire will be lines of rubber that will securely hold the wire in place and help with the water resistance. This is necessary for communication between the camera and computer for computer vision. This opening will only be small enough for the cable and will not affect the overall protection the case provides to the camera.

8.4.4 Camera Mounting and Positioning

The position of the camera will be the last parts of the structure to be finalized. Once the structure and motor system are completed, choosing a suitable location to complement the firing of the paintball gun will be tested then agreed upon. We must ensure that our camera will be placed in a stable position to get an uninterrupted feed so are images are without jitters or distortions. We also want to test the imaging range of the camera, and the effect it will have on targeting depending on where it will be placed.

Due to the camera being made for a stationary desktop computer we will have to create a small structure to mount the camera to the rest of the system. It was decided that the best place to have the camera mounted was close to directly below the paintball gun as possible. From the base of the structure below the gun is a protruding member to which the camera and housing will sit. This will provide the stability the camera would need to accurately track a target and a suitable position in relation to the paintball gun for firing. We will be attaching a L bracket to the base of the structure to mount the camera housing to, as modeled below in **Figure 48**.

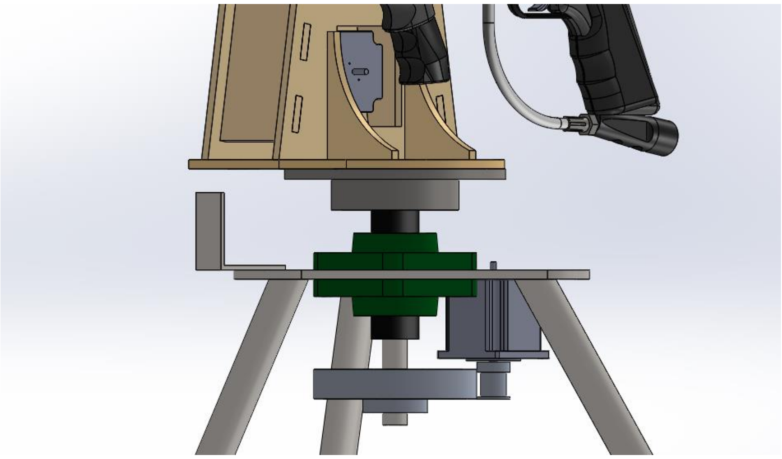


Figure 48: Camera Housing Arm Mount

To mount the camera housing, we will use a clamp. This will give us the ability to securely affix camera housing for testing. This mechanism is also ideal as we will have to test various positions for mounting. The clamp will also be very flexible for our needs as it will allow us to move the camera housing or grant the ability to take off and clean or replace if necessary. While the clamp will be affixed to the camera housing, it will not cause any damage to the camera or the area it is affixed to. This will serve as an exceptionally reliable method for testing various mounting areas without causing damage to our prototype.

When our prototype is near completion, we will extensively test if the position of our camera housing is sufficient. The prototype will experience a small amount of recoil from the firing of the paintball gun, but with the design of the housing and the clamp, the stability of our camera should adequate. The foam padding inside the camera housing will absorb some of the recoil keeping the camera in place and safe. The clamp will also keep the camera housing in place when firing and limit some movement also.

8.4.5 Warning Light Structure

We will be fabricating our own warning light structure. It will be 3d printed and made from plastic. The lights will be front facing with no obstructions so it will be visible. The structure will be a flat with 12mm holes for the lights to screw into. This will be then securely attached to the front of the prototype. This is the most cost-effective method than buying both the light and the structure. The light will then blink red and orange to signify a warning to targets within the range of the turret.

8.4.6 Position of Paintball Gun

The position of the gun will be determined by two stepper motors, one handling the pan, moving the point of the gun in a horizontal direction and another motor tiling the paintball gun, which will move vertically. For our design, pictured below in **Figure 49**, we choose to have the paintball gun side mounted to the gear that is connected to the stepper motor controlling the tilt of the gun.

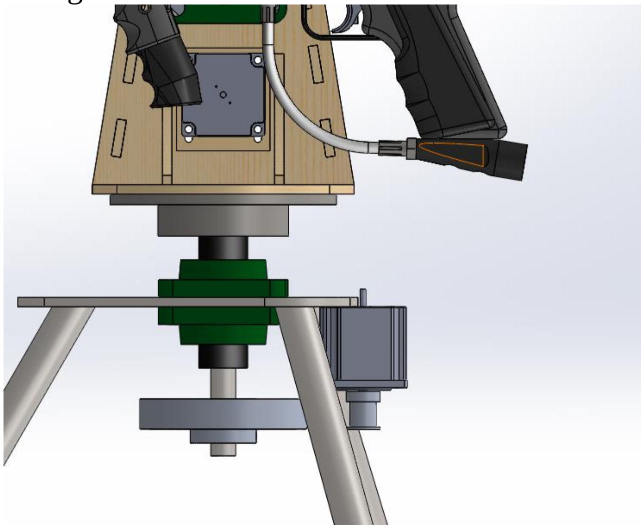


Figure 49: Mount for tilt and pan stepper motor

The gear that is closest to the gun will be connected to stepper motor with a reduction system between, the gear reduction system will be further discussed in the motor design section. The mount holding the first motor and gun will have its base mounted to the underside main panel of the base to best save space and keep the system relatively compact as visualized in **Figure 50**.

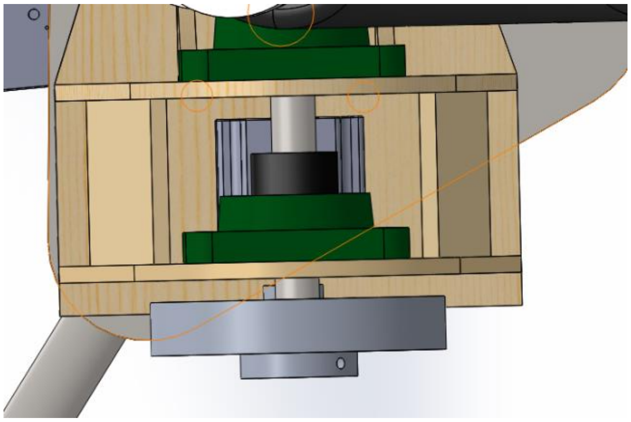


Figure 50: Mount for tilt and pan stepper motor

8.4.4.2 Paintball Gun Mounting

To mount the paintball gun to the structure we chose to create a clamping piece around the gun at its center of gravity. This will easily allow us to mount the gun to the stepper motor that will move the gun along the z- axis.

The only modification that will be made to the stock paintball gun will be to the compressed air tank. Due to the length of the gun when the compressed air tank is attached, it would've been much more difficult to mount the gun to the mountings that will lead to the motor. Having the air tank attached will also affect the center of gravity of the paintball gun. By removing the compressed air tank, the center of gravity of the unit will be moved further up the length of the gun which will allow for a more centered mounting point.

Due to the fact that the air tank cannot be left at the end of the gun from a total system standpoint the only options that are available involve mounting the tank elsewhere on the structure. The better option would be to run a longer hose from the end of the gun down to the leg of the structure where the tank could easily be clamped. In **Figure 51**, the red outlined area shows what it would look like to have the air tank properly mounted on the end of the paintball gun, which would cause the center of gravity to move drastically closer to the butt of the gun which would lead to greater accuracy issues as the stepper motor would require more torque to adjust the gun which could lead to inaccuracy in the calculations. However, if the air tank were to be mounted onto the leg of the structure, outlined below in blue, then balancing the gun on the motor structure would be much easier and more accurate. This being said, the air tank could also be mounted below the base of the structure.

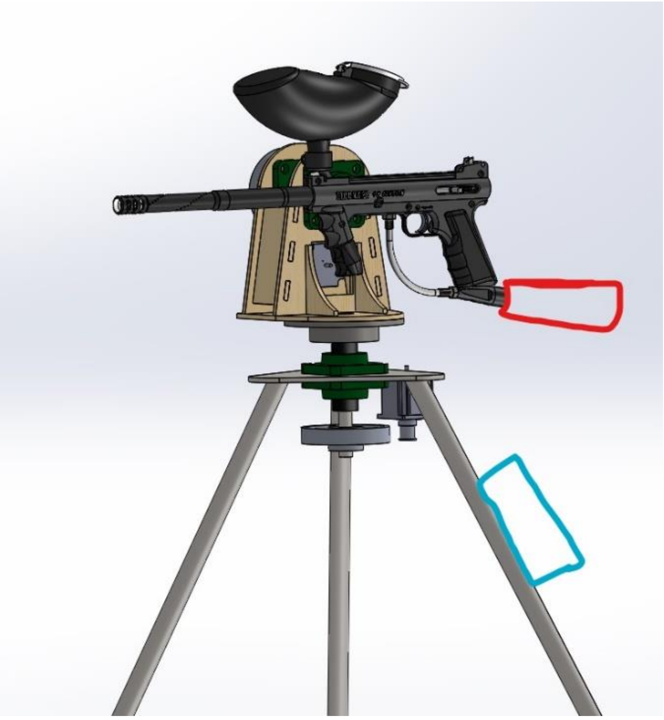


Figure 51: Mount for compressed air tank

The better option is clamping the tank to the leg of the structure due to the fact that this will require less bends in the tube and a more direct line from the output of the tank to the input of the paintball gun, this will cause less ware on the connections between the hose, tank and gun.

8.5 Motor Design

When choosing which motors would best fit into the design of the sentry turret as depicted in Section 8.4, there were multiple factors that had to be considered. The most important factor being precision. At shorter distances, having larger step sizes would not affect the overall precision of the motor by a significant degree, but because the sentry turret system is to be accurate a distance of 50 feet or more, the differences between step sizes become increasingly impactful. Being able to both have a stepper motor with smaller steps—around 0.9 degrees—and the ability to take advantage of micro-stepping, is very important. Another factor that had to be considered was how much load the motor would need to move. In the case of the current system being designed, that load will be approximately 10 pounds. All factors taken into consideration, the Nema 23 was the obvious choice, as it has the capability to provide the necessary torque required to move the load and aim the hit indication device. Also, the Nema 23 stepper motor is widely available, so the members of Group 33 should not expect to encounter supply chain issues when attempting to secure this part.

8.5.1 Driver

In order to move the stepper motors, the microcontroller will need stepper motor drivers to communicate between the microcontroller and the motors themselves. These drivers will act as an intermediary bridge between hardware and software. The driver being used is set up for a voltage anywhere between 6.5 and 44 Volts and an amperage of about 2.1 amps, up to 5 amps. For our motors we will require an average amperage between 3 and 4 amps. For this driver in particular we will need a forced air flow if we are to go above 2.1 amps, overheating is not expected to be too much of an issue due to the fact that there will not be a constant current supplied. Rather, the current supplied will sharply increase and sharply decrease each time the computer detects a new target. When compared to the heat that would be generated if the drivers needed to hold a steady 4 to 5 amps, the the heat generated by the drivers through these short spikes in current would be less hot for a longer period of time. If the drivers were required to maintain the current steadily, heat would be generated much more quickly. That being said, to cover all bases we will be implementing a cooling system by adding slots for airflow and monitoring the temperature during prototyping to determine if additional measures will need to be taken.

8.5.2 Board Implementation

In **Figure 52: Connections for Stepper Motors** (below), the layout of the drivers and their connections between the motors and the microcontroller can be observed. The drivers will also be connected to the same 5-volt source that the microcontroller is going to be connected to. It was possible to make this decision because of the overlap between the operating voltages of the drivers and the microcontroller. This overlap occurred at 5 volts.

Eliminating the necessity of adding another voltage divider to the printed circuit board will result in a cleaner and more efficient design overall.

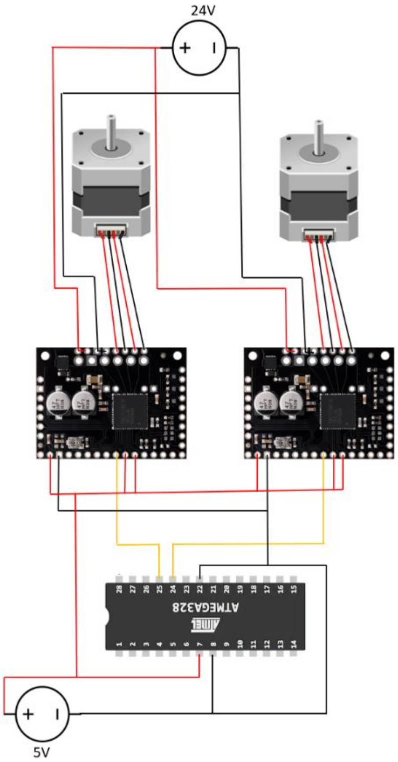


Figure 52: connections for stepper motors

The next decision made which would impact the design of the printed circuit board was to avoid having the 24-volts that will need to be run to the stepper motor drivers. Rather than running the 24-volts through the PCB, we will instead be running a separate line straight to the drivers. This decision eases the process of prototyping the circuit board for this project. Once the prototyping of the board is complete this decision may change based on the necessities of the project and of the members of Group 33. For now, the connections will be as shown in **Figure 53: Schematic for Stepper Motors**, below.

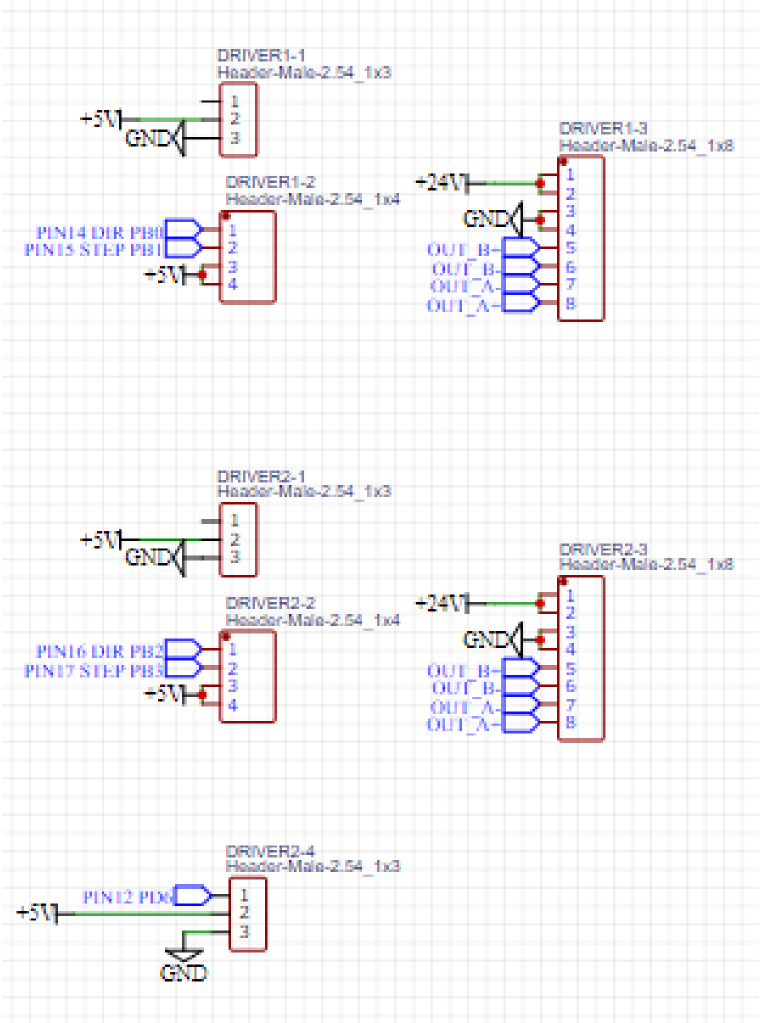


Figure 53: schematic for stepper motors

8.5.3 Motor Reduction System

To determine what the optimal step size in terms of what the accuracy of the system will be a few calculations can be done to determine what the best number of micro-steps will be best. To do the calculations we will have to make a few assumptions, the first being what we want to call the average human running speed, for these calculations I chose to pick the average sprinting speed of humans, about 10 mph, for my calculations and then I chose for the systems average distance for the target to be about 50 feet away, while still testing the max distance of 75 feet. The last part that needed chosen was the reduction ratio, this was mostly determined by cost and availability, thus by choosing a reduction ratio of 3 or 6 was easiest and most cost effective, I chose 6 because the higher gear ratio allows us to have greater accuracy. The motor we chose has a step size of 0.9 degrees.

The outputs of these calculations are after the gear reduction has been applied can be seen in **Table 25: Gear Reduction Calculations**, below.

Target Distance (feet)	Reduction Ratio	Number of Micro-Steps	Motor Speed (RPM)	Motor Step Size (degrees)	Turret Step Size (degrees)	Accuracy (inches)
50	3	16	8.4034	0.3	0.01875	0.19635
50	3	32	8.4034	0.3	0.009375	0.098175
50	6	16	16.8068	0.15	0.009375	0.098175
50	6	32	16.8068	0.15	0.004688	0.049087
75	3	16	5.602267	0.3	0.01875	0.294524
75	3	32	5.602267	0.3	0.009375	0.147262
75	6	16	11.20453	0.15	0.009375	0.147262
75	6	32	11.20453	0.15	0.004688	0.073631

Table 25: Gear Reduction Calculations

This confirms the final calculations and design decisions are decent. Hence, we will be using a reduction ratio of 6 paired with 32 micro steps. Using the 32 micro steps will also allow for smoother transitions. Having the higher gear ratio paired with the micro steps will also allow for higher accuracy at further distances. In the figures below where the motor shaft is in line with the shaft connected to the turret, there is empty space but in reality, there will be a belt added to fully connect the system.

8.5.4 Modifications to Trigger

The paintball marker selected will require modifications to be affixed to the sentry turret project in such a way that it is capable of firing automatically. **Figure 54: Firing Mechanism Side View** (below) and **Figure 55: Firing Mechanism Bottom View** (below) presents a crude drawing of the firing mechanism that will be mounted to the handle of the paintball gun. In the drawing, the red/pink colored sections represent the handle of the paintball gun. The black sections represent one face of the housing for the firing mechanism. The beige and green sections represent a motor and ovular attachment, respectively. The motor will be connected to the PCB, which will activate the motor when the sentry turret is programmed to fire the paintball gun. As the motor rotates, the ovular attachment will press against the trigger of the paintball gun to fire a projectile from the device once per rotation.

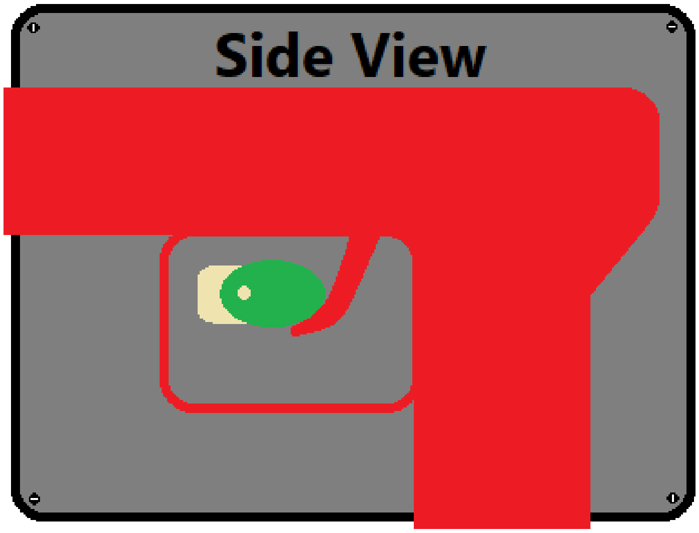


Figure 54: Firing Mechanism Side View

The housing for the attachment depicted in **Figure 54** and **Figure 55** will be 3D printed using a model specifically designed to fit the paintball marker chosen for the project. This attachment must fit tightly to the paintball gun to ensure the rotation of the motor causes sufficient contact with the trigger to result in the emission of a paintball projectile from the barrel of the gun.

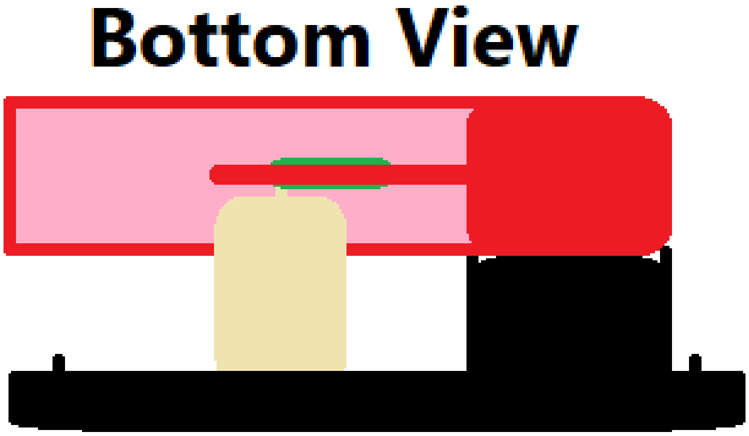


Figure 55: Firing Mechanism Bottom View

8.6 Trigger Mechanism Design

The sentry turret project will require a modified paintball gun to automatically fire at targets without human intervention past the point of arming the device. This modified paintball gun will be achieved through the attachment of a motorized mechanism to the handle/trigger of the Tippman Model 98 Paintball Gun visible in **Figure 56**. Construction of the mechanism will begin by designing a digital three-dimensional model of the mechanism's frame. The model will be composed of two halves which will interlock with one another around the trigger of the paintball gun. A servo motor will be fitted to the inside of the housing and glued into place in such a way that a force directed perpendicular to the motor will not result in the motor becoming dislodged from its proper place within the housing. Next, a plastic disk will be attached to the motor with an offset such that rotation of the motor will allow the disk to press against and release the trigger of the paintball gun. The motor will be programmed to respond to the recognition of a suitable target by the camera. After recognition and target acquisition, the paintball gun will be aimed at the target, and the motor will perform a full rotation once every second while the target is still within range of the sentry turret. The rotation of the motor will begin from rest, a position in which the disk attached to the motor makes no contact with the trigger of the paintball gun; the use of a servo motor for this task will provide enough precision in this movement to ensure that the disk ends its rotation in exactly the same position as it started. This factor is extremely important in ensuring the continued functionality of the sentry turret for any shots following the first, as a misalignment in the disk's position could result in misfires or lack of firing altogether.

The trigger activation mechanism described above must fit tightly to the handle/trigger of the paintball marker to be effective and durable when firing the device. Because of this, the mechanism had to be designed after Group 33 obtained the Tippman Model 98 Paintball Gun which would be used for the project. The Group determined the most appropriate size and structure for the mechanism given the shape of the paintball gun, then took precise measurements of the relevant portions of the marker. These precise measurements allowed the 3D model to be created with the necessary dimensions to be attached to the paintball gun.

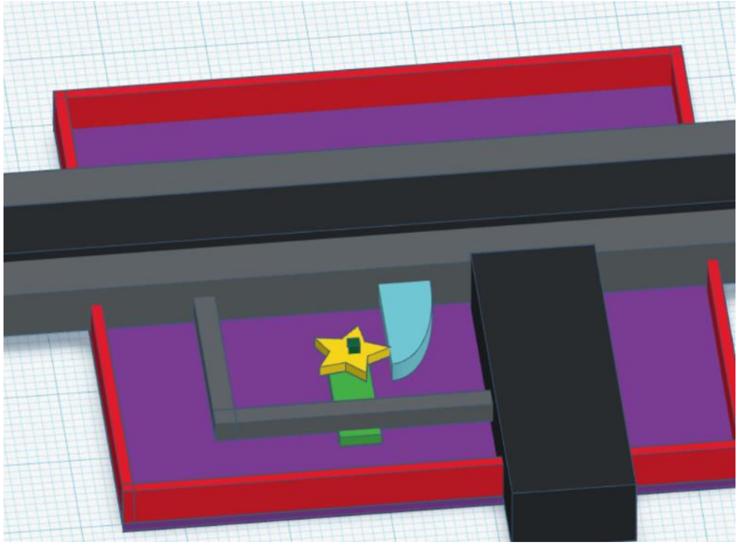


Figure 56: Trigger Activation Mechanism Concept Design

A conceptual design of this trigger mechanism is included in **Figure 56**, above. In this image, shapes colored purple/red represent the housing for the mechanism, while gray/black shapes represent the Tippman Model 98 Paintball Gun. The blue shape attached to the paintball gun in this image is used to show the trigger, and the green shapes model the Tower Pro MG995 Servo Motor. Attached to the motor, the yellow star shape represents the motor attachment which, when the motor is activated, will collide with and activate the trigger of the paintball marker. Note that the star shape was chosen based on available attachments that come stock with the motor, as seen in **Figure 18: Tower Pro MG995 with attachments** (in section 4.2.3).

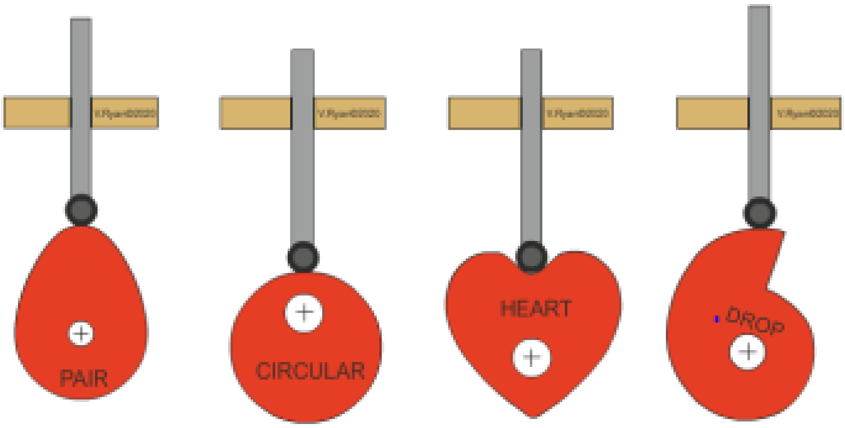


Figure 57: CAM Mechanisms

For the triggering mechanism, as can be seen pictured as a star above, is a placeholder for what will be the one of the CAMs pictured in **Figure 57: CAM Mechanisms** (above). The Pair is a widely used and effective shape for pushing a lever, such as a trigger. The pair shape, however, may not be the best shape for a fast release. For our system we would want a shape more akin to the drop design which would allow for us to fully push in the trigger and a fraction of a second later the trigger would be released and ready to be pushed again. From there we would be have multiple drops on one piece. Our design would essentially be that of a rachet with less drops and the trigger would be the pinion. In **Figure 58: Rachet and Pinion** (below) the triggering wheel would be labeled 1 and would have fewer and deeper teeth than what is shown, the trigger, acting as the pinion, is labeled as 2. The part labeled as 3 would be mounted into the trigger guard, so that we will be able to measure once the paintball gun is delivered and be able to accurately size.

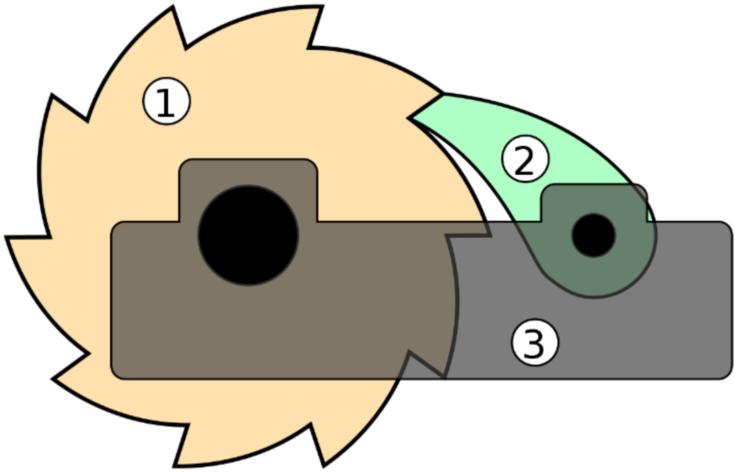


Figure 58: Rachet and Pinion

8.6.1 Motor Selection and Calculations

The trigger mechanism will use the Tower Pro MG995 servo motor. The Tippmann 98 paintball gun has a trigger pull of 2.5 pounds. Of course, pounds don't convert directly into pound-inches, although we can estimate that the trigger would require at least about 2.5 pound-inches to pull. 2.5 pound-inches is equivalent to 2.88 kg-cm, or 40 oz-in. The Tower Pro MG995 has a torque of 8.5kg-cm at 4.8V, which is enough to pull the gun's trigger. The MG995 also has a speed of 60 degrees in 0.2 seconds at 4.8V, which practically means that there will be a delay of at least 0.2 seconds between the microcontroller activating the trigger mechanism, and the trigger actually being pulled.

8.6.2 Triggering Mechanism Fastening

The fastening for the trigger activation mechanism will go through many stages of development. First, for prototyping and development, the mechanism will be fastened to the paintball gun using grooves included within the 3D model's design, then secured using elastic bands. These grooves will prove to the members of Group 33 whether the measurements taken for this purpose were, in fact, accurate and sufficient for designing this attachment. The elastic bands will pull the two halves of the 3D printed housing together when placed around the trigger of the gun. The bands will be removable to allow for removal of the attachment for the purpose of making further adjustments when developing the product.

After the functionality of the mechanism has been tested using these fastenings, Group 33 will employ more permanent fixtures to ensure that the mechanism cannot be jostled out of place. The three-dimensional model will be modified to include ridges within the structure to interlock with one another when the halves are pressed together. These interlocking pieces will be glued together when the final form of the trigger activation mechanism has been reached, making the mechanism a permanent part of the sentry turret device (unless broken apart through sheer force).

Once we have determined the sizing of the trigger guard and position of the trigger within the guard and the angle at which the trigger triggers then we can determine the final size of the CAM and from there the placement of the servo, initially we will set up a temporary fastener to the guard of the gun. But once we have all of the measurements finalized then we will size out the housing that will hold the servo and wrap around the trigger guard and if extra stability is needed then we will have the mount go up and sit above the top of the gun where we can clamp it into place.

9.0 Testing & Prototypes

This section will go into detail on how each part of the project will be tested to ensure the turret is functioning according to its specifications and requirements. Testing each component individually will make integrating the entire system easier.

9.1 Hardware Testing

Hardware testing of the turret system will ensure that the PCB, motors, paintball gun, camera, warning light, motion sensor and power supply are all functioning correctly. The testing environment of individual hardware components will be done at the UCF TI Innovation lab. The final test of the completed turret will be done outdoors at one of the group members' homes.

9.1.1 PCB / Microcontroller Testing

For the PCB fabrication, we will first prototype most components using the Arduino and we will be able to connect the drivers and motors through Arduino, breadboard and straight to the drivers. So, the 24-volt source will only be able to go straight to the drivers and to power the microcontroller and lights we will simply use small batteries in place of the voltage regulators. Once the PCB breadboard prototype is fully functional, we will create a rough prototype PCB layout, such that we will be able to test out the USB converter chip and the voltage regulators. From there we will ideally have a working prototype that we will be able to base the final PCB design and layout off of. By having a rough printed circuit board, it will be easy to test for failures and see where improvements can be made before we order the finalized board.

The microcontroller will be tested on the Arduino and once it has been tested it has been proven to function correctly, we will use the Arduino to burn the bootloader onto it. The Microcontroller will be one of the last items placed onto the PCB prototype to minimize the chances of malfunction.

9.1.1.1 PCB Prototype

During the time between the semesters all the necessary parts will be ordered. Each of the chips will initially be double ordered in through pin versions. This will allow us to have 2 chances to fully assemble a rough prototype PCB prior to having the PCB professionally printed.

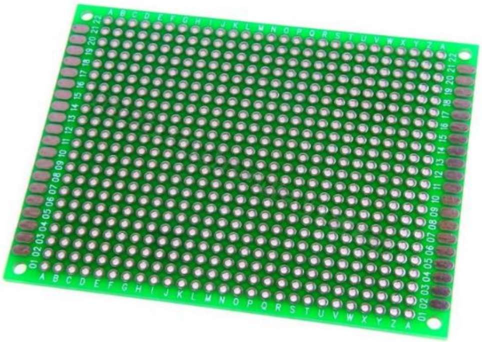


Figure 59: PCB Testing

To have the best chance of success, and to avoid having to spend exorbitant amounts on just rebuying all of the different components, each sub system will be prototyped separately first. This means that each of the voltage regulators will be assembled and tested prior to being connected to anything. And the microcontroller to USB set up will be on a separate prototype board which will be powered by 5 Volt battery such that we can ensure that it will not be compromised by a possibly faulty voltage regulator. But once the two voltage regulators are both insured to work then the board will be attached to the board where the USB 2 microcontroller prototype is, and we will make sure that those attachments are secure and working from there we will add the lights to the board coming off of the 12 Volt regulator. Once the lights are fully working we will move on to connecting the motors and their respective drivers to the board those will be the last added because we can already ensure that the drivers will work as they are going to be purchased premade tested separately prior to attaching to our system once all the boards are attached and all the connections are secure and the system works we will move on to creating a rough prototype and making sure that everything will fit together neatly period from there we will export to EAGLE where we will finalize our board layout and send it off to be printed.

9.1.2 Power Supply Testing

Before attaching the two 12-Volt batteries in parallel both will be tested, as shown in **Figure 60**, separately to ensure that they are both running at the estimated voltage. Once both units are properly working and charged, they will be attached in parallel and retested before being connected to the PCB. This will ensure that the correct voltage and current is being output such that the voltage regulators will work correctly. This step is important as we will need to ensure our battery will be able to last for testing and demonstration. This step will be done quite really in the design process to ensure that the batteries are outputting in a voltage range that will work with our voltage regulators. If they are operating out of the expected range, then we will first double check that the battery is fully charged, then that is not corrupted, and that all the connections are good.

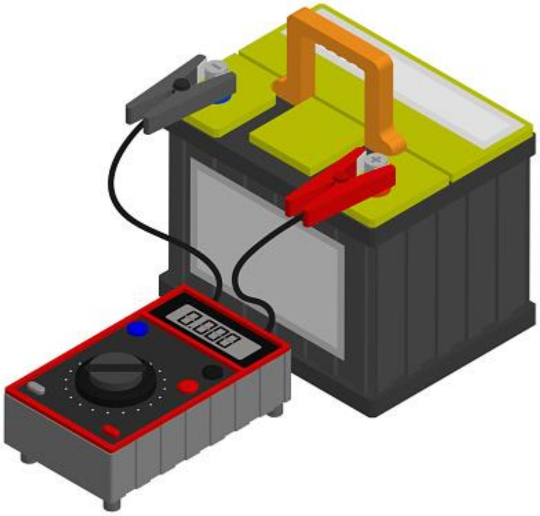


Figure 60: Battery Testing

9.1.3 Motor Testing

Each motor and driver will be tested individually such that we can avoid failure. Once we are sure the motors are working correctly, this will be done with a smaller voltage source and capacitors, to avoid having to test both the drivers and motors together initially, such that if one isn't working it won't cause problems with the other.

1. Hook up the 24-Volt power supply
2. Temporarily connect the power to the Motor driver
3. Manually touch the motors to the ports on the drivers
4. Ensure that the motors are moving in the direction of the inputs

This method of testing will ensure that once the motors are attached to the system and they fail it will not be because of a faulty motor, which will allow us to more quickly pinpoint where the error could be in the system.

9.1.4 Camera Testing

The testing of the camera is done to ensure the camera is functioning and properly sending a video feed to the laptop without interruption or failure.

1. Connect the camera to the laptop
2. Turn on the power
3. Begin the camera's stream to the laptop
4. Observe the video stream, checking for failures, lag, low framerate, etc.

Errors may result from faulty connections, software drivers, or malfunctioning hardware in the camera or laptop.

9.2 Software Testing

Software testing will ensure that the laptop and microcontroller software successfully carry out their respective roles. The turret system must power up correctly, respond to the motion sensor, receive camera feed, run the target detection & acquisition algorithms, communicate commands from the laptop to the microcontroller, carry out these commands for the motors to aim and fire the turret. Software testing will be carried out after hardware testing has been completed.

9.2.1 Computer Vision Testing

The most integral software component is the laptop's computer vision algorithm. If this component is successful, then it will receive a video stream from the turret's camera, detect any humans in the camera's view, then generate commands for the microcontroller which orchestrates the aiming and firing of the turret at the human. The procedure for testing the laptop's software is as follows:

1. Turn on the system (laptop, camera, etc.)
2. Start the program

3. While one group member watches the laptop, another walks in front of the camera and then stands still.
4. Observe the results displayed on the laptop, and check if they meet the software's specifications. If they do, continue the test. If they do not, the program has failed, and fixes are required.
5. Have the group member in front of the camera walk from side to side and back and forward to test the program's ability to track targets.
6. Repeat step 4.
7. Have another group member stand in front of the camera and have the two group members slowly change position back and forward so that one group member is closer to the camera than the other. This is to test multiple target acquisition and prioritization.

The laptop should draw a bounding box around the detected humans. Then it should be a calculating priority if more than one human is detected, based on the distance of the target to the turret. Then it should be creating a binary number which decodes into instructions for the microcontroller. Debug printouts should be visible that list the required adjustments for the turret accompanied by "translations" for the binary commands. These binary commands should match the debug information, and should make sense for the target's current location (e.g., if the target is to the left of the turret, the command should be telling the microcontroller to activate the pan stepper motor in order to turn the turret to the left).

The testing environment for the software will be at UCF's Senior Design Laboratory. Personal computer (laptops) will be running the python program on the Windows Visual Studio Code IDE.

9.2.2 Microcontroller Testing

Send a test command to the microcontroller through the laptop, commanding it to turn the turret in each direction, turn on the warning light, then fire, and finally power down. This test command will ensure that every function of the microcontroller is working according to the command given.

1. Power on system
2. Send a pre-made command from the laptop to the microcontroller
3. Observe the results and compare them to the intended results of the command. If the actual results do not match the expected results, then the microcontroller software has an error.
4. Repeat this process for every motion of the turret (panning, tilting, firing, etc.)

The testing environment for the software will be at UCF's Senior Design Laboratory. The same laptop used for the computer vision test will be connected to the microcontroller through a USB connection. The laptop will be running the Arduino IDE to monitor and edit the microcontroller's program code.

9.3 Potential Problems

Any project being designed from the ground up will experience its share of problems in the course of its development. With this in mind, the members of Group 33 have prepared for a number of potential problems which they anticipate encountering at some point.

9.3.1 Component Communication

With our project having different sections that must talk to each other, there is a possibility we may have hang ups in our design. One of our biggest concerns is the code from the computer will not communicate with the microcontroller. If this happens the whole cascade of the computer talking to the microcontroller telling it to activate the lights and motors will not happen. These problems can be caused by the FT232RL USB to Serial UART interface, or more specifically, our misuse of the FT232RL. Nobody in our group has actually put together our own PCB, so our inexperience may cause us to build the PCB incorrectly in a way in which the FT232RL interface fails to function.

Another issue could be no communication from USB to microcontroller. It could be an issue with the code, and we would have to troubleshoot and look up ways to fix it. On the other hand, the code could be working perfectly fine, but if the USB module does not work, we would have to order a new part. Similarly, the microcontroller could not communicate with the motors and the turret would not move or fire. Here there would be two reasons also. The motors could have malfunctioned, or code is not working as intended. The code for the motors would be similar to each other so that could rule out if the code was faulty.

9.3.2 Target Tracking

There could be problems with OpenCV not sufficiently tracking targets that are moving in view of the camera. The impact of this issue could result in projectiles being fired in unintended directions and at unintended objects, risking damage to nearby property. If this happens there could be errors within the code or there could be a library that was not installed. To fix this first check the required files that are required to make OpenCV function. If everything is installed go back and ensure that the code has the correct logic or is not missing a step.

This issue would present itself as the system not correctly hitting the target. If we were to learn everything on the software side is correct, then we'd have to move on to the hardware and embedded side of the system to find the issue. The most likely cause would either be functionality I think microcontroller, which could then be the issue in the bootloader which could simply be reburned. Or the issue could be in the motors themselves, however if we had previously tested that we could rule it out. The last area that we could find the issue in is communication with the drivers from the microcontroller.

9.3.3 Warning Light Failure

Our warning light could also not trigger on. It could possibly be how we connected the light as a switch between the 12V and output, faulty code, not receiving enough power, or part malfunction. For this part, troubleshooting will be very tricky. For the malfunctioned part option, we would unscrew the lights from the screw terminal and hook up a new one. We also test the light to see if it is not getting power. Lastly, we could manipulate the code to see if the setup is wrong.

If we further find that everything around the warning light failure is properly working though to be safe to assume that the warning lights themselves were burnt out by something from there, we could go on to test if the voltage regulators were functioning correctly because if there happened to be a spike in the voltage regulation and a voltage over 12 volts were to get supplied to the warning lights then they could burn out.

9.3.4 Errant Calculations

If our voltage stepdown was incorrect it could cause our prototype to malfunction and not receive proper power. The only option to fix this would be to check our PCB schematics and ensure everything was put in correctly. If it was not done correctly, we would have to send off for a new PCB to be made which would cost us time and money. Ideally, this error is going to be avoided by using a multimeter once on the board has been printed and using a multimeter to compare out inputs to outputs, once we have verified that voltage step-downs are what was calculated we will add the microcontroller and USB to Serial chip. Another way we could avoid this is by pre-ordering a few extra voltage regulators to verify they will act as they should. If this problem were to occur against all precautions, we would simply have to redesign that section of the board.

9.3.5 Motion Sensor

As seen in the parts selection section of the document, the ranges of motion sensors within our budgetary range are all quite small. The sensor we are most likely to try is the PIR sensor, which has an effective range of 30 feet. This is smaller than the proposed operating range of the turret, meaning that potential targets will have to move a significant distance into the turret’s range to activate it. This would help keep the turret power efficient, ensure the turret fires at targets its actually capable of hitting, and wouldn’t be a problem at all in a small indoors area. However, it is likely to cause a serious problem.

The turret may never be activated if nobody comes within the relatively short 30-foot range of the PIR motion sensor. This would result in people being able to easily bypass the turret by remining just outside of the motion sensor’s range.

9.3.6 Problem Management

The uncertainty of what might happen is a reality of development. The biggest issue is if we will have the time and budget to resolve these problems as they occur, or if we will need to redesign important parts of our project. Therefore, it is imperative for us to start building early so we can work out any bugs or redesigns as they appear. Table * below outlines some potential problems and suggest some solutions we could take to fix them. As stated earlier the members in our group have not attempted a project on this scale before so this will be learning from trial and error.

Problem	**Solutions**
No communication between the computer and microcontroller	• Check connection between computer and microcontroller • Check code with a test setup on a breadboard • Possible faulty USB module
Program not tracking targets	• Ensure that OpenCV has all required files • Check to make sure logic of the code is correct and there are no errors
Microcontroller not communicating with motors	• Malfunctioned motor • Check code if all motors are not working
Warning light not triggering	• Check connection • Remove LED and test • Check code
Voltage stepdown incorrect	• New PCB

Table 26: Possible problems with components

10.0 Project Operation

The following sections are steps we will take to operate our prototype. It will involve interactions with targets, safety precautions for users, and troubleshooting for user problems that may arise. It is similar to the function of a user manual and outlines some steps for operation and safety.

The primary purpose of the paintball turret is to be used recreationally in paintball tournament games. The way the paintball turret is implemented in these games is up to the game's organizers and/or players. It could conceivably be used as an environmental hazard that all players must avoid – used to corral players into specific areas for more intense firefights. It poses a new challenge of maneuvering around the turret's firing range to the players. It could also be used specifically for team battles, with the turret's software being modified to fire only at targets wearing a jersey of a specific color. For example, with two teams Red and Blue, each team will get a turret which they can place in selection of spots predesignated by the game organizers to provide support to their team. The team's turret will be programmed to fire only at targets wearing jerseys of the enemy team's color. Our group is sure that players could come up with even more ways to use the turret.

10.1 User Interaction

Given that our system will be used by paintball teams we will be configuring it to target certain jerseys specific to each unit such that eventually if there are multiple units on one field each will be able to have a separate color to target. There will also be a section for jersey color selection to be assigned to the referees such that no unit on the field will ever target them. This setup will allow the system to be seamlessly integrated into any paintball field.

The turret is to be placed into an open area on solid, flat ground. Before operating the turret, confirm that all users are abiding by safety guidelines. Connect the battery to the turret's circuitry, then connect the laptop to the microcontroller's USB. Turn on the turret and the laptop. Begin running the software for the laptop and microcontroller. Calibrate the turret, and then set it to automatic mode. Load the paintball gun's ammo hopper. The turret should now be fully operational.

10.2 Safety Precautions

Prior to testing on anyone we will be using a cardboard cutout of a person to ensure that the targeting system is accurate and will not aim above the shoulders of the target. Once we can ensure that the system is correctly aiming, we will then move on to testing with volunteers. Everyone will be in full paintball gear as well as a motorcycle helmet to best protect everyone in case of anything going astray with the system. We will be conducting the tests in a large private field to negate the issue of having the system target anyone not a part of the tests.

Safety precautions for users of the turret:

1. Safety precautions pertaining to the turret are to be included in the safety orientation given to players before games.
2. When playing with the paintball gun turret, paintball players should be following the same safety precautions they would any other time they play paintball.

3. Protective paintball masks and armor are to be worn at all times within the designated shooting areas.
4. The turret will use field paint for its paintballs, to avoid using balls that could cause harm to the players.
5. The turret is only to be set up and turned on within designated shooting areas, to avoid the possibility of opening fire on people not wearing protective equipment.
6. Given that the turret's human detection algorithm cannot detect when a player has been eliminated, any player eliminated by the turret is to first immediately leave the turret's line of sight / firing range.
7. Players should not intentionally fire at the turret with the intent to damage it or blind the camera – doing this may affect its target acquisition algorithm in unexpected ways.
8. If the turret is to be turned on outside of the designated shooting area for any reason (demonstration, maintenance, etc.) it is not to be loaded with ammunition unless those around it are wearing protective equipment.
9. Keep the turret's power supply away from water or any other liquid to prevent damage to the equipment.
10. Ensure that no wiring is exposed to the elements. Wires should be covered by a jacket or insulation.
11. During operation of the turret, the enclosure for the electronic components (the PCB) should be kept closed to protect the electronics from the elements or other environmental hazards.
12. Do not turn on the device if there are any loose or torn wires.
13. If opening the enclosure to the electrical components, make sure the power is turned off, and wait a few minutes for the circuits to discharge electricity.
14. If unsure the electrical components still have electricity stored in the circuit, check for power using a multimeter.

10.2 General Information and Troubleshooting

If you are having difficulty getting the turret to work, use the following **Table 27,** to assist in troubleshooting your problem:

Problem	Solutions
Device isn't Powering Up	• Check the battery's charge • Check for any loose wires or connections
Device Targets Wrong Colors	• Restart System • Double check color selections
Device Constantly Misses Targets	• Clean the camera lens • Check structure stability • Check if motors or belts are obstructed
Device Doesn't Shoot	• Is the trigger servo working • Is the servo attachment hitting • Are all the wires secured

Table 27: User Problems

The first issue that may arise when the user encounters our system is that the device will not power on. The first solution to l game. The solutions for this are to restart the system and double check the color selections within the system. If the device misses all its targets, then this could mean an issue with the camera lens being dirty against needs cleaned. Or that the structure is not on stable ground so the simplest fix to this is to make sure that all three legs of the structure are securely grounded in that it is not moving when it fires and turns. Or the issue could be within the motors and belts an if they get too dirty or muddy then that could cause them to become stuck and immobile. The last issue that we could foresee would be that the device just doesn't shoot this would probably because that something happened with the servo the first suggestion would be to check the connections from the PCB to the servo. The other solution to this issue could be that the servo attachment is not hitting the trigger which could be because it either became dislodged or broke. This should cover all of the potential issues that the user may face when using our system. As the system continues to be tested and prototyped more issues may arise that we can add to this section.

Our prototype is only in the testing phase, so occurrences of malfunctions or loose wires are a common place. Problems may arise for users when operating the prototype. It is best to be prepared and think ahead for solutions. Instead of assuming a component or code is broken it is best to think of possible errors. It may not in fact be the prototype that is malfunctioning, but something that the user has done.These user problems may even help to develop new method to implement in our prototype or help future engineering students build their own prototype.

9.0 Administrative Content

The following two sections of Milestones and Budget will serve as guidelines towards our prototype. Milestones will outline the timeframe for both Senior Design I and II. It will show where we need to be going forward with the project or if we need to restructure with the allotted time on our table. Here we will plan how many weeks to spend on design, testing, and the final prototype. In the budget section there will be a current list of specific components and materials as we need for the prototype. This section will be updated as we move forward to the final result.

11.1 Milestones

Senior Design I		
Number	**Milestone**	**Planned Completion Week**
1	Project Ideas	August 27th, 2021
2	Final Idea and Roles	September 9th, 2021
3	Divide and Conquer V1	September 17th, 2021
4	Divide and Conquer V2	October 1st, 2021
5	Add Sections: Part Comparison, Research, Standards	October 8th, 2021
6	Microcontroller Analysis	October 15th, 2021
7	60 Page Draft	November 5th, 2021
8	Meeting With Dr. Wei	November 8th, 2021
9	Continue Developing Report	November 8th, 2021
10	Beginning PCB Design	November 12th, 2021
11	100 Page Draft	November 19th, 2021
12	Order Parts	December 1st, 2021
13	Finalizing and Editing Report	December 4th, 2021
14	Final Document Due	December 7th, 2021

Table 3: Senior Design I Milestones and Weekly Breakdown

Senior Design II		
Number	**Milestone**	**Planned Completion Week**
15	Prototyping Components	Winter Break
16	Start building	TBD
17	Testing	TBD
18	PCB fabrication	TBD
19	Finish Prototype	TBD
20	Final Testing	TBD
21	Presentation	TBD

Table 4: Senior Design II Milestones and Weekly Breakdown

This section is a planned breakdown of the timeframe of both Senior Design I and II. **Table 3** and **Table 4 is** our projected plan of how much time to spend on each phase. It details our phases and the projected weeks. This will serve as a planned overview of our schedule. It will be updated as many of these phases have not occurred. In the Planned Completion Week column those will be adjusted depending on the amount of time actually spent on the project.

Senior Design I will largely be focused on planning our prototype. It is where we will agree upon our initial idea and set forth our ideas of building an actual prototype. Documentation for the focus of our project will be started and expanded upon to show our progress. Research is also a major focus at this time as we will find what already exists in the market or if there have been other prototypes completed. We will also discuss what parts will be necessary, and research if they will be sufficient enough to build a functioning prototype. Our research will also include technology and parts. First technology will be analyzed to see which would be best for our project. This will include both hardware and software. Hardware will take most of the time as it will contain the majority of comparisons. Many parts will have different methods of implementation and we will have to decide which is best based on cost versus features. Later along this timeline design ideas will take place along with initial schematics. Our PCB design will also be started during this phase, and we will be able to define it further as the semester progresses. Finally, we will order parts we have agreed upon, and test them for functionality. This is going to be a semester of fine tuning our ideas into a project that can be achieved considering our budget and all the resources available. It is subject to change as we find new information and technology.

Senior Design II will execute the building of our prototype. This phase will be heavily focused on building and testing. With our parts ordered and tested we will begin with the initial design ideas outlined in the Senior Design I document. We will ensure structural integrity of all the working parts before assembling. During Senior Design I we would have completed a plan of what would be sufficient for our project, but Senior Design II will put form to our plan. During this phase we will find out what will work from testing. Parts may have to be reordered or replaced to find what will work best for the prototype. For our prototype to be demonstrated successfully this phase must consist of a finalized plan.

9.2 Budget

Our project will be self-funded, with a maximum budget of $400 ($100 for each member). The estimates of price breakdown based on project sections and parts are given below in **Table 5**, excluding replacements for malfunctions or damages.

Item	**Quantity**	**Price Estimate**
Camera	1	$40 – $80
Sensor	1	$25
Power Cord	1	$10
Internal Power Supply	1	$28
Gun	1	$20 – $100
Motors	2	$20
Microcontroller	1	$35 – $75
Jumper Wires	1	$5
Paint	1	$10
Screws and Washers	2	$20
Adhesive	1	$12
Red Warning Light	1	$5
Materials for Structure	1	$50
Total	**-**	**$280 – $475**

Table 5: Parts and Budget

10.0 Project Summary and Conclusions

We have designed and re-designed our project multiple times throughout this semester. There have been many automated turret systems made before, but there is a plethora of different ways to approach the same idea. A lot of research was spent deciding on a singular, practical design for our purposes. Since there has been so much previous work in this area, there were a lot of examples to draw inspiration from. We went through multiple preliminary designs, starting with a catapult, then moving to a cannon, and finally to a paintball gun turret.

The purpose of the design also changed with the design of the turret's gun. The initial purpose of the catapult was cornhole, the cannon for personal defense, and the paintball gun is for recreational use in paintball games. We had to narrow down our purpose for this project, but in theory many other concepts can be designed from our concept. With more engineering it could help put out fires by tossing water cans onto a fire. It could be programmed to assist with watering plants at a certain time of day. Or it could play a game of cornhole with you and keep score.

Many of our design choices were made with our budget foremost in our minds, and budget limits have required significant changes to our overall design as we progressed. The most significant of these was the constant degrading of our central computer for the turret – going from an SBC to a microcontroller paired with one of our laptops. This would in turn cause changes to the rest of the design, particularly that of the software.

The paintball turret can be used in paintball tournaments as a sort of environmental hazard or tactical equipment available to the teams. The turret can be set up in different locations to make the paintball arenas more dynamic and engaging for the players. The turret isn't completely limited to paintball either – the turret gun can be changed out and modified for different purposes (although the greater the difference in gun, the more changes to the rest of the turret's structure would be necessary). For example, the paintball gun could be changed out for a laser diode for use in laser tag. It could even be given laser tag targets attached to the structure and wired up to the microcontroller, programmed to temporarily disable the turret when hit.

There are more implementations that could be considered for our prototype. In consideration, if this project was used for home defense, we could have an early alert for the homeowner. If we had additional time and funding, we could develop a phone app that could prompt a message saying the turret was activated. For recreational use we could set up an app for remote access and players could control where and when to fire at targets. As we have mentioned before, the firing tool could be switched out for other things such as nerf guns, water guns, or airsoft guns.

Paintball is not really a particularly important issue in the world, and only has a niche consumer/fan base. However, there is not a requirement for senior design projects to be world-changing panaceas, or applicable to the majority of the human race. We wanted to do this project because we thought it would be fun and a challenge for our engineering skills. We hope this project results in a finished product that achieves our goals and is fun for paintball players to play with.

11.0 Project Roles

This section will break down what each member will contribute to the team and their responsibilities. We will work together so we will be directly involved in the entire process of building our prototype. This breakdown does not mean that members will be solely responsible for what is listed, but rather active contribution and helping each other complete tasks toward the final product.

11.1 Liderma Guerry Roles

Liderma has split experience hardware and software. She will be assisting with building on the hardware side and assisting with coding as needed. She will work with the leads to complete and inspect to ensure each task is completed.

- Building and implementing parts
- Making sure LEDs are programmed correctly
- Making sure parts are mounted correctly
- Assisting hardware and software leads on their tasks

11.2 Quintin Jimenez Roles

Quintin has the most experience with coding. He will take the lead in programming and software troubleshooting. He will be using Python with the OpenCV portion and C++ with the embedded portion. He will take direction supervision on programming decisions.

- Programming for OpenCV on the computer
- Programming for communication with the microcontroller
- Troubleshooting for code
- Integration of code between computer and microcontroller

11.3 Michael Macallister Roles

Michael has experience in both computer vision and some Arduino projects. He will take the lead on the computer vison side and assist with the transition of the code from the computer interacting with the microcontroller. He will have some supervision over the motors.

- Checking and assisting coding with OpenCV
- Checking and assisting with microcontroller integration
- Overseeing servo and stepper motors
- Direction over the camera for proper computer vision

11.4 Kaitlyn Martin Roles

Kaitlyn has the most experience in the area of electrical engineering. For this reason, she will be the lead when it comes to selecting components for PCB and fabrication. She will be guiding us in the following tasks:

- Working on PCB design and fabrication

- Power Regulation of the system
- Checking if each component is connected
- Checking the integrity of electrical components

12.0 References

Admin, “Types of motion sensors: Working and Their Applications,” *WatElectronics.com*, 12-Oct-2021. [Online]. Available: https://www.watelectronics.com/types-of-motion-sensors-working-and-applications/.

D. Primbet, “Logitech C920 Webcam Review,” *Tom's Guide*, 08-Mar-2021. [Online]. Available: https://www.tomsguide.com/reviews/logitech-c920-webcam.

ElectroPeak, “PIR motion sensor: How to use PIRS W/ Arduino & Raspberry Pi,” *Arduino Project Hub*, 08-Apr-2019. [Online]. Available: https://create.arduino.cc/projecthub/electropeak/pir-motion-sensor-how-to-use-pirs-w-arduino-raspberry-pi-18d7fa.

HackerHouse, “Raspberry pi motion tracking Airsoft Turret,” *Instructables circuits*, 22-Sep-2017. [Online]. Available: https://www.instructables.com/Raspberry-Pi-Motion-Tracking-Airsoft-Turret/.

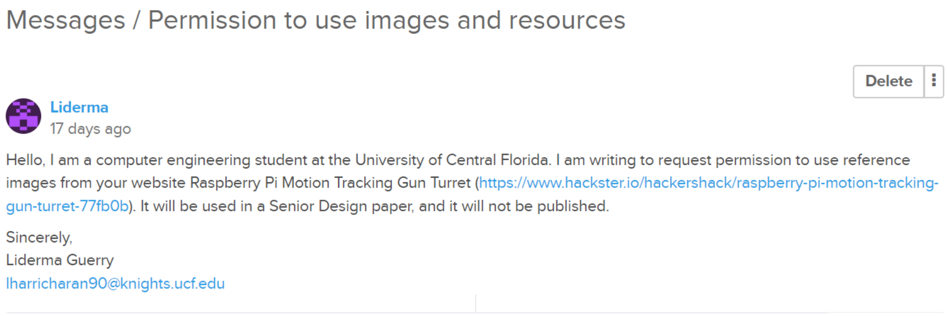
“RPI USB webcams,” *RPi USB Webcams - eLinux.org*, 15-Sep-2020. [Online]. Available: https://elinux.org/RPi_USB_Webcams.

T. Williams, “Logitech C270 HD webcam review,” *Tom's Guide*, 07-May-2021. [Online]. Available: https://www.tomsguide.com/reviews/logitech-c270-hd-webcam.

T. Williams, “Logitech C922 Pro HD Stream Webcam Review,” *Tom's Guide*, 27-Jul-2021. [Online]. Available: https://www.tomsguide.com/reviews/logitech-c922-pro-webcam.

Z. Wendt, “Arduino compatible sensors: Types of Arduino sensors: Arrow.com,” *Arrow.com*, 06-Mar-2020. [Online]. Available:https://www.arrow.com/en/research-and-events/articles/sensor-prototyping-motion-detection-sonar-and-more.

13.0 Copyright Permissions

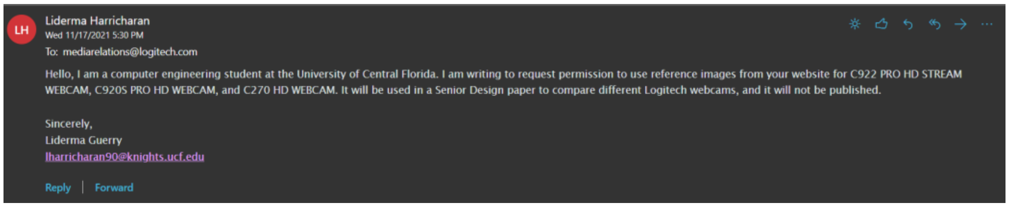


Messages / Permission to use images and resources

Liderma
17 days ago

Hello, I am a computer engineering student at the University of Central Florida. I am writing to request permission to use reference images from your website Raspberry Pi Motion Tracking Gun Turret (https://www.hackster.io/hackershack/raspberry-pi-motion-tracking-gun-turret-77fb0b). It will be used in a Senior Design paper, and it will not be published.

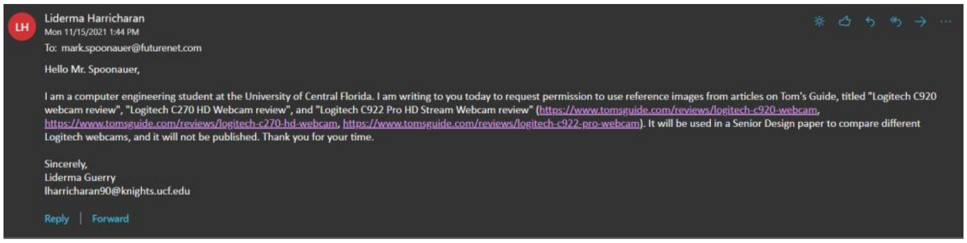
Sincerely,
Liderma Guerry
lharricharan90@knights.ucf.edu



Liderma Harricharan
Wed 11/17/2021 5:30 PM
To: mediarelations@logitech.com

Hello, I am a computer engineering student at the University of Central Florida. I am writing to request permission to use reference images from your website for C922 PRO HD STREAM WEBCAM, C920S PRO HD WEBCAM, and C270 HD WEBCAM. It will be used in a Senior Design paper to compare different Logitech webcams, and it will not be published.

Sincerely,
Liderma Guerry
lharricharan90@knights.ucf.edu

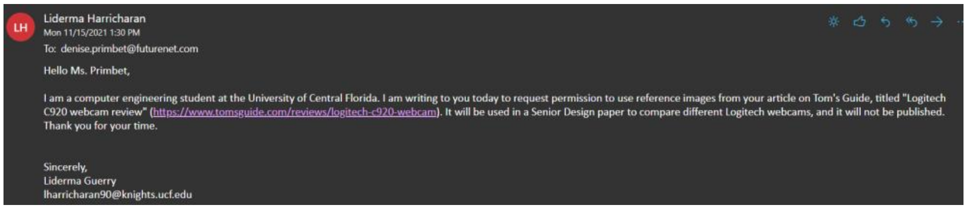


Liderma Harricharan
Mon 11/15/2021 1:44 PM
To: mark.spoonauer@futurenet.com

Hello Mr. Spoonauer,

I am a computer engineering student at the University of Central Florida. I am writing to you today to request permission to use reference images from articles on Tom's Guide, titled "Logitech C920 webcam review", "Logitech C270 HD Webcam review", and "Logitech C922 Pro HD Stream Webcam review" (https://www.tomsguide.com/reviews/logitech-c920-webcam, https://www.tomsguide.com/reviews/logitech-c270-hd-webcam, https://www.tomsguide.com/reviews/logitech-c922-pro-webcam). It will be used in a Senior Design paper to compare different Logitech webcams, and it will not be published. Thank you for your time.

Sincerely,
Liderma Guerry
lharricharan90@knights.ucf.edu



Liderma Harricharan
Mon 11/15/2021 1:30 PM
To: denise.primbet@futurenet.com

Hello Ms. Primbet,

I am a computer engineering student at the University of Central Florida. I am writing to you today to request permission to use reference images from your article on Tom's Guide, titled "Logitech C920 webcam review" (https://www.tomsguide.com/reviews/logitech-c920-webcam). It will be used in a Senior Design paper to compare different Logitech webcams, and it will not be published. Thank you for your time.

Sincerely,
Liderma Guerry
lharricharan90@knights.ucf.edu