(Add Group 34) Sentry Tracking Attack Turret

Code name: *S.T.A.T*

Nicholas Martucci	*Electrical Engineer*
Carlos Garcia	*Electrical Engineer*
Devak Sharma	*Computer Engineer*
Christian Solanet	*Computer Engineer*

Review Committee

Dr. Wei Sun	*ECE*	Professor
Dr. Sonali Das(TBD)	*ECE*	Lecturer
Dr. Nazanin Rahnavard(TBD)	*ECE*	Professor
Dr. Qifeng Li	*ECE*	Associate Professor
Dr. Suboh Suboh	*ECE*	Associate Professor

Mentor

Sreeram Sundaresh

2.0 Project description

Our project is not ground-breaking or something no one has thought of before. This project is to show the wide range and thought that engineers must take into account. We all need each other to move forward, and one wrong step could mean the difference between keeping our country safe or tumbling down around us.

Our group decided to build a system that can track a target in a set coordinate system and be able to take it out using a turret. This is no easy task as multiple sensors need to be in communication along with needing to know what exactly to look out for. Should it look for a sudden movement or a certain color? Should it be if anything is sensed in this area, we should start tracking it?

How is this accomplished? Let's start with the subsystems we have. First is the power system/access box which will take a 120V/12V AC/DC converter from a wall plug in and then step the voltage down for further use. Voltage will then be distributed from there to the MCU, LEDs, buttons and all other peripherals. We will have separate access panel which will house the power system along with status LEDs, buttons, kill switches, etc. for this project. All human inputs and status updates will be on this box. We will have LEDs for status such as friendly or enemy target as well as to know if our turret has locked on. We will have switches to turn on our devices as well as emergency kill switches in case something happens that is unexpected. Another subsystem we have is the Turret substation which will house the MCU due to the fact that the turret with its motors and servos will need most of the wiring connected to it. We plan on having this project be able to break down, which will be explained in a later section so having less wired connections externally will help with our requirements to stay portable and user friendly. Our last subsystem is the Sentry Camera set up along with our HuskyLens sensor on the turret itself. The Sentry Cameras will be used for early detection of an object and will allow our system plenty of time to read the target, determine if it's friend or foe, and give ample time to respond to this object. The Sentry Cams will send over the target's location to the HuskyLens which is attached to the turret. The turret will lock on to the objects position and will first turn on the laser diode to show we have locked on to said target then we will fire a dart at the target.

To help with the setting up of our system, allowing it to be repositioned, we have our systems connected by some external wires. As stated before, our access box will serve as power to the whole system. We will have wires that come out of this box that will connect and power the turret as well as the sentry cameras. We will want to position the turret and cameras however to minimize not only latency but also offset of the camera coordinates from the turret. This can be done by hardware, however, to avoid mistakes with our control, placement of the turret and camera should be closed and allow transition and translation of data to be easy.

2.1 Project background/motivation

The design of our project is set up to mimic how a battlefield would work in a real-life military situation. We have the Sentry cameras to sense our target. This could be representative as a pilot giving coordinates to the naval ship which would have the artillery to shoot the target in question a thousand miles away. For simplicity our information is communicated by wires. The pilot would give coordinates in which case the ship would have to adjust to those coordinates. Once the coordinates are locked on by the turret the ship would be able to shoot, taking out the target. Just like our project which requires data from the Sentry cameras to get the initial location. Then our turret must lock-on shown by the laser diode and be able to shoot the target as well.

2.2 Current technology/Past projects

Military technology – C-RAM

The C-Ram stands for Counter-Rocket, Artillery, and Motor. This technology will sense incoming projectiles and be able to counter them using predictive planning and missiles to stop them. This relates to our project in the sense that we will want to be able to see our target and stop them in their tracks by intercepting them. C-RAM and our STAT require the use of sensors to detect and guess how the target is moving to accurately suppress the threat.

Traffic Stop Watchdog

This project is taking slight inspiration from one of our own engineer's brothers and everyone's mutual interest in this type of technology. A past project called Traffic Stop Watchdog had a camera mounted on a movable base that would be able to scan a room and use image detection to determine if there was an officer present in the room based off the color of their shirt. The camera will then keep the officer in the center frame while they perform their duties. The scanning function of the Traffic Stop Watchdog served as inspiration for the two stationary sentry cameras that we will be using. The tracking functionality served as inspiration for the pan-tilt mount which will ultimately follow our target object. These cameras will work together to detect, locate, and follow an object of our choosing.

Nerf Dart Turret

Other projects have tried to build automated Nerf style turrets, but most stop short of full autonomy. One example is the Nerf Dart Turret posted on Printables. That design uses a computer and WiFi link to control the turret, letting the user aim and fire remotely. It's more about showing off the mechanical launcher than tracking a target, since the system doesn't detect or follow moving objects on its own.

Our design builds on this idea but adds several important layers. Instead of relying on manual control, the turret uses two stationary cameras to constantly watch the area. When motion is detected, the turret wakes up and its own onboard camera takes over for precise aiming. IMU provides orientation feedback to smooth turret motion and prevent overshoot. These additions make the system capable of detecting, tracking, and firing automatically, without operator steering.

2.3 Objectives and goals

The main goal is to be able to acknowledge and track a ground target using the 2D coordinate system set up by the sentry cameras on the housing station. We also want the turret to enter a fully "locked on" state. Other goals can be found below.

Main goals

- Sentry cameras identify a target on ground
- Turret responds to data from sentry cameras by following movement of target on ground
- Turret locks on to target on ground with data from sentry cameras with laser
- Access box signals that an object has been detected
- Turret launches projectile towards target

Advanced goals

- Turret hits target with nerf dart with above 70% accuracy
- Be able to differentiate between friend and foe at the same time

Stretch goals

- Be able to track aerial targets moving in x, y and z axis
- Be able to lock on to aerial targets based on data from sentry cameras
- Be able to see laser from diode on aerial targets before firing
- Be able to hit target in air with nerf dart

Objectives

(Rewrite these similar to goals

Should include something that we are planning to achieve: NUMBERS

Accuracy, distance, etc etc

Update main goals to be stuff we can for sure to make

Update accuracy to be a main goal, and improve accuracy with advanced goals)

3D Printed Housing Station – The housing station for the sentry cameras and to stabilize the base of the turret must be 3D printed. Due to needing the cameras at a certain height and needing to attach turret base to something stable we must also have customization available to us. We plan on having everything connected to this be able to be detached to avoid any wiring issues with input and outputs. Sentry cameras need to be able to be secured and stable at a high height so that they don't fall which could damage circuitry and lenses. Turrets wiring and PCB must be able to be detached for safe storage but must also be able to be secured to housing station securely.

3D Printed turret – The 3D printed turret allows for customized hardware and software allowing for us to customize reaction time of sensors and data transfer along with optimized start-up of motors instead of constant need to be ready. Also, turret needs to include a shooting mechanism, HuskyLens sensor, and laser diode. Those three components should be able to move as one entity. This requires a base that can house all three of these components at the same time without losing reaction time or falling apart. 3D printing the base and turret itself allows extra customization that we can't get from pre-built tilt kits.

AC/DC PCB – This PCB is needed for power. Battery would be sufficient, however we are focusing on motion tracking and targeting, efficiency is a secondary goal. To allow us to work and worry about power at a later time we are using a wall plug in to power our system. This requires us to take a 120V/12V AC/DC converter wall plug into a PCB on the access board where we will then further regulate the voltage for other uses such as LEDs, buttons, MCU, and motors.

Sentry Cameras – These cameras need to be able to detect that an object has entered its field of view, pinpoint where the object is on the screen, and relay this information to the MCU. Extensive research is needed to choose the correct camera for this operation. We need a camera that can take in the software and also give us back information. They need to be able to communicate with each other as well, if not the theoretical 2D coordinate system will not work as intended.

Huskylens sensor – Motion tracking sensor that allows for full lock on attached to the turret. Using this sensor allows us to track the target after we are given the initial coordinates from the sentry cameras.

Laser Diode – This is a visual verification that the turret has locked on to the target in case any failure happens when it comes to shooting the target.

Software Specific Objectives

- The system must be able to generate a world coordinate system that each camera can transform its local coordinate system into to move the pan-tilt mount accurately.
- The two stationary cameras must be sampled one after the other and take precedence of whichever camera detects an object first.
- The two stationary cameras must determine an estimated distance based off the object being a known size
- The two stationary cameras must be able to ignore objects too large or too small to avoid hitting humans or objects to far away
- The system shall store the detected objects with a flag marking if they have been previously detected by the sentry cameras
- The software shall use the distance detected to determine what external indicators and peripherals should be activated or deactivated.

2.3.1 Hardware *Basic Goals*

The turret operates on a simple principle. Detect moving targets, track them, and fire safely at close range. Two stationary cameras monitor the area, waking up the turret when motion is detected. These cameras act as the first line of detection, they're always watching but don't drain power since they're not constantly processing. Once triggered, the turret's own camera takes over for aiming making sure it only fires when targets are within about five feet.

The mechanical system needs different motor types working together, a stepper motor handling horizontal rotation, giving smooth sweeping motion across the target area without the jerky movement you'd get from a regular DC motor. Vertical tilting uses a servo motor, which provides enough precision for elevation adjustments without getting too complex. The servo can't rotate continuously like the stepper, but it doesn't need to for simple up and down movement. The projectile system uses dual flywheels driven by a brushless motor, keeping dart speed consistent for better accuracy. The flywheels spin up to speed and stay there, so when a dart gets fed between them, it launches with the same velocity every time.

Power and safety controls sit in a separate access panel, positioned where users can reach it safely without getting in front of the turret. The system runs off a 12-volt wall adapter that is certified to take advantage of its safety benefits like overvoltage and overcurrent protection. Internal circuits then step-down voltages for the electronics, motors, and cameras since they all need different power levels. The control panel has the key safety features: an emergency stop button for instant shutdown, and LEDs showing if the system is powered, armed, ready to fire, or has a fault. A buzzer gives audio alerts too, which helps during demonstrations.

Safety drove most of the design decisions from the start. The launcher only fires soft foam darts like Nerf darts with motor speeds capped and measured so they don't hit too hard and have higher accuracy. This keeps the demo safe for a classroom setting, which was always the main goal. The emergency stop provides backup protection if something goes wrong.

Splitting the design into two parts (control panel and turret) made everything safer and more reliable. It's easier to build this way since you can work on each part separately, and demonstrations can't accidentally become dangerous because all the safety controls are physically separated from the shooting mechanism. The modular approach also makes troubleshooting easier when something inevitably goes wrong during development.

2.3.2 Software *Basic Goals*

The software is built on the ESP-IDF/Arduino framework for ESP32-S3, which provides access to FreeRTOS multitasking, hardware timers, and communication devices. On the stationary (sentry) cameras, the built-in camera driver library is used for low-resolution frame capture, while lightweight OpenCV-style image routines (i.e. frame differencing or blob detection) run locally. Each node uses the ESP-NOW wireless library to send and receive data between devices with low latency and high throughput. The HuskyLens on the pan-tilt mount is accessed through its dedicated HuskyLens library (UART/I2C) to read bounding boxes, object IDs, and lock state. The ESP32-S3 microcontroller uses the FreeRTOS to separate the responsibilities of communication, motor control, safety monitoring, and ensuring the turret remains responsive while processing vision data.

The laser and camera are coordinated in a hierarchical fashion. The stationary cameras on the left and right side of the housing station perform coarse detection of motion or color and transmit only target coordinates rather than full images either wirelessly or through UART/I2C. The ESP32-S3 moves the pan-tilt mount to point the main camera at the suspected target region. Once in view, the HuskyLens takes over for fine tracking and object/image recognition. A PID control loop continuously adjusts the turret until the target is centered. When the HuskyLens confirms a stable lock, the laser is enabled through GPIO controlled driver circuit. For safety measures, the software enforces interlocks and mechanical limits before activation.

A PID controller will be used to stabilize the mount's horizontal movements (driven by stepper motor) and vertical movements (driven by the servo motor). This controller utilizes the IMU (gyroscope and accelerometer) to measure the angular velocity and acceleration, while the PID algorithm computes the proportional, integral, and derivative errors to adjust the motor outputs, ensuring stable and accurate orientation of the mount. A Complementary or Kalman Filter is applied to fuse the sensor data, so that the noise from the accelerometer and drift from the gyroscope is reduced, yielding an accurate and stable angle estimate.

The display design provides real time feedback to the operator. To indicate different stages of operation, multiple LEDs will be used. When the system is powering on, a green LED will light up to indicate that the system has received power. A blue LED will light up to indicate that the stationary cameras are operating and are currently scanning objects. If an object has been spotted by the stationary cameras, the orange LED will light up. Once the pan-tilt mount locates the object, if it detects an enemy, there are three possible LED outputs. If an enemy is detected but it is beyond the 10-foot range, the orange LED will stay on. If an enemy is detected and it is inside the 10-foot range, the red LED will turn on. If the enemy is within 5 feet of the pan-tilt mount, the red LED will start rapidly blinking.

The HuskyLens camera used for object detection comes equipped with a built-in 2-inch screen that displays the camera visuals in real time. This screen will be used as secondary verification that the camera is correctly detecting and identifying the objects in its FOV. The primary verification that the camera and mount are working properly will be the laser diode that pinpoints the target when identified.

2.3.2.1 Why didn't we use Arduino or Raspberry Pi?

The ESP32-S3 was specifically chosen because of its dual-core processor which allows it to handle multiple functions simultaneously. One core can run vision and communication, while the other core handles peripherals and control loops. With 45 programmable GPIO pins, the ESP32 can support many types of interfaces such as displays, cameras, and sensors through UART, SPI, or I2C. The ESP32-S3 is the smallest and lightest microcontroller out of the three options, so the mount won't be weighed down by it, and provides leeway when choosing the cameras and other components. It costs around $10-$15, making it cost effective.

Table 1 Comparison of MCU selection

ESP32-S3	Raspberry Pi Pico 2 (RP2350)	Arduino Uno R3 (ATMega328P)
Dual Core 32-bit Microprocessor at 240MHz	Dual Core Microprocessor at 150MHz	Single Core 8-bit Microprocessor at 16MHz
512KB SRAM	520KB SRAM	2KB SRAM
45 programmable GPIOs	26 programmable GPIO pins	Supports audio
Low power management	Supports audio	1 USART, 1 SPI, 1 I2C interface options
Supports multiple interface options	Supports multiple interface options	Brown out detection
4MB-64MB of flash memory	4MB of flash memory	39 GPIO pins
Dimensions: 21.7 x 17.8mm, 1.2g	Dimensions: 51 x 21mm, 3g	Dimensions: 68.6 x 53.4mm, 25g
Input voltage of 3.3V	Input voltage of 3.3V	Input voltage of 5V
Ranges from $10-$20	Ranges from $5-$15	Ranges from $20-$30

2.4 Requirement Specifications

Below you will find our Specification requirements for demo purposes (highlighted in orange) along with other requirements we have set for ourselves to further our knowledge in real world engineering aspects during this project.

Table 2: Specification Requirements

Parameter	Description	Target Value and Unit(s)	Market requirement corresponding #
Sentry Range Detection	Sentry cameras must be able to detect an object that has entered its desired range and create a bounding box around said object.	Greater than eight feet	1,3

Success Rate of Detection	This would be the success rate of sentry camera detection of object along with the turret detection and laser diode lock on.	Greater than 90%	1,3,5
Turret Reaction Time	The time that it takes the turret to wake up, position itself towards the object, detect the object, and determine if it should lock onto it must be quick.	Less than or equal to 2 seconds	1,3
Success Rate of shooting dart	After the turret is locked on with the laser diode this would be the percentage of accuracy of being able to hit the target with a nerf dart.	Greater than 70%	1,3
Size	The dimensions of the housing station combined with the dimensions of the turret must be able to fit comfortably on a school desk.	Size of desk is about... Length less than equal to 30 inches Width less than equal to 24 inches	2,4,5
Weight	We want this to be portable so the weight must be manageable by the average person	Less than 25lbs	4,5
Connectivity	The delay between cameras taking a sample and sending it to the MCU which then sends the data to the HuskyLens attached to the turret	Less than or equal to one second	1,3,5
Setup/Tear down time	The device does not need to be set up quickly because it is not used in high-risk scenarios with a strict timing requirement. The device does not need to be taken down as it is designed to be installed and left alone.	Less than 5 minutes	2,4,5
Cost	This is a project design by college students so the cost should not be very high	Less than $1000	all

Table 3 – Marketing Requirements

Below can be found the marketing requirements our group has settled on for this project.

Marketing Requirements
1. **Fast detection and fast lock on times** 2. **Ease of physical set up for user** 3. **Ease of use for software, only need to change what target to detect**

4. **We want to make sure this system is small enough to move around to reposition if needed (aka Portable)**
5. **Cost efficient**

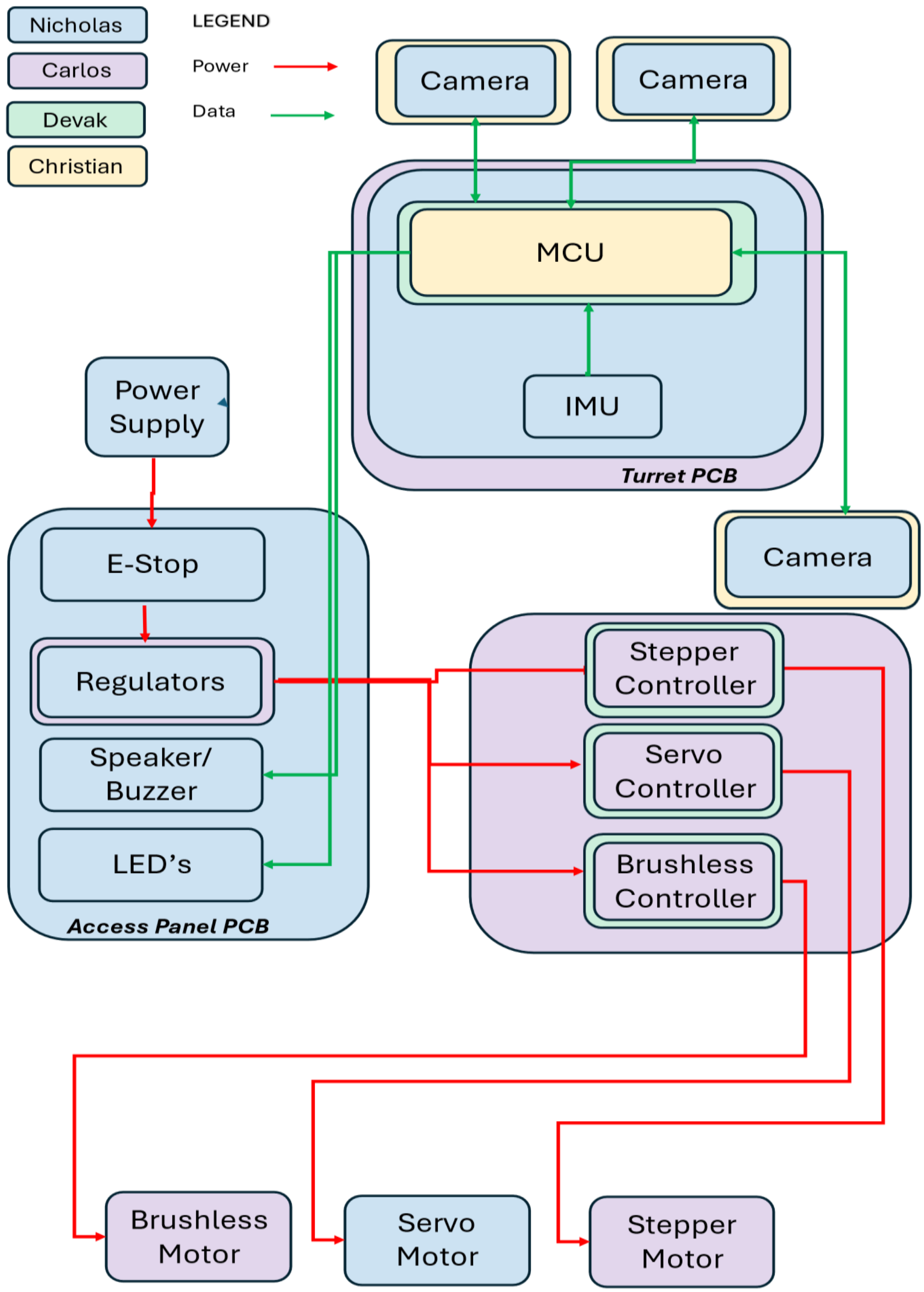
The table below is meant for showing more specifics on both the hardware side and the software side of quantitative values we expect during demonstration and test.

Table 4 - System Requirements

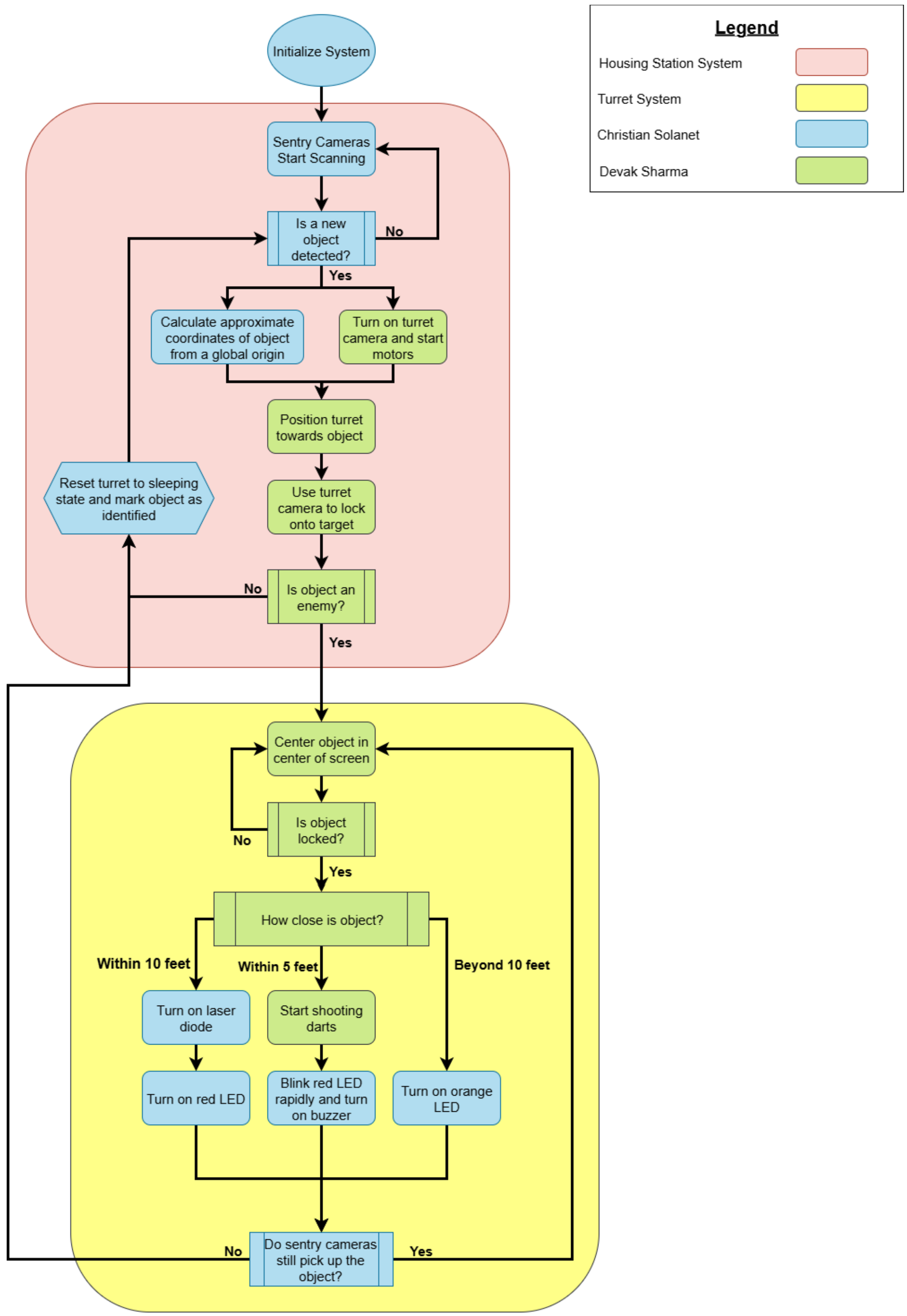
Component	Parameter	Subsystem and description	Target Value and Unit(s)
Cameras	Field of view	Software – The two stationary cameras need to provide an overall FOV that is sufficient to detect objects around the area.	Greater than 90 degrees
Cameras	Sentry Detection Time	Software – Two cameras need to be able to detect an object that has entered the desired range to then send a signal to wake the turret up	Detect target with at least 10 feet from cameras
Cameras	Turret Detection Time	Software – The turret camera needs to be able to detect an object and aim.	Detect/aim in < 2 seconds
Servo Motor / Stepper	Pan/Tilt angles	Hardware- The turret will be able to pan and tilt to the minimum angles that the fixed cameras are capable of detecting.	Pan- 270° Tilt- 90°
IMU (accelerometer / gyro)	Raw data processing	Hardware – Take in raw data to input into PID controller to stabilize signal to send over to the turret for further use.	Stabilize
Access box	User input and status updates	This box will act as the user input for the system. We will have buttons to turn on the system and	LEDs and buttons turn on with in 500ms of expected turn on

		emergency stop as well. LEDs indicate status such as friendly object or enemy object when a target is detected.	

Hardware diagram – Figure 1



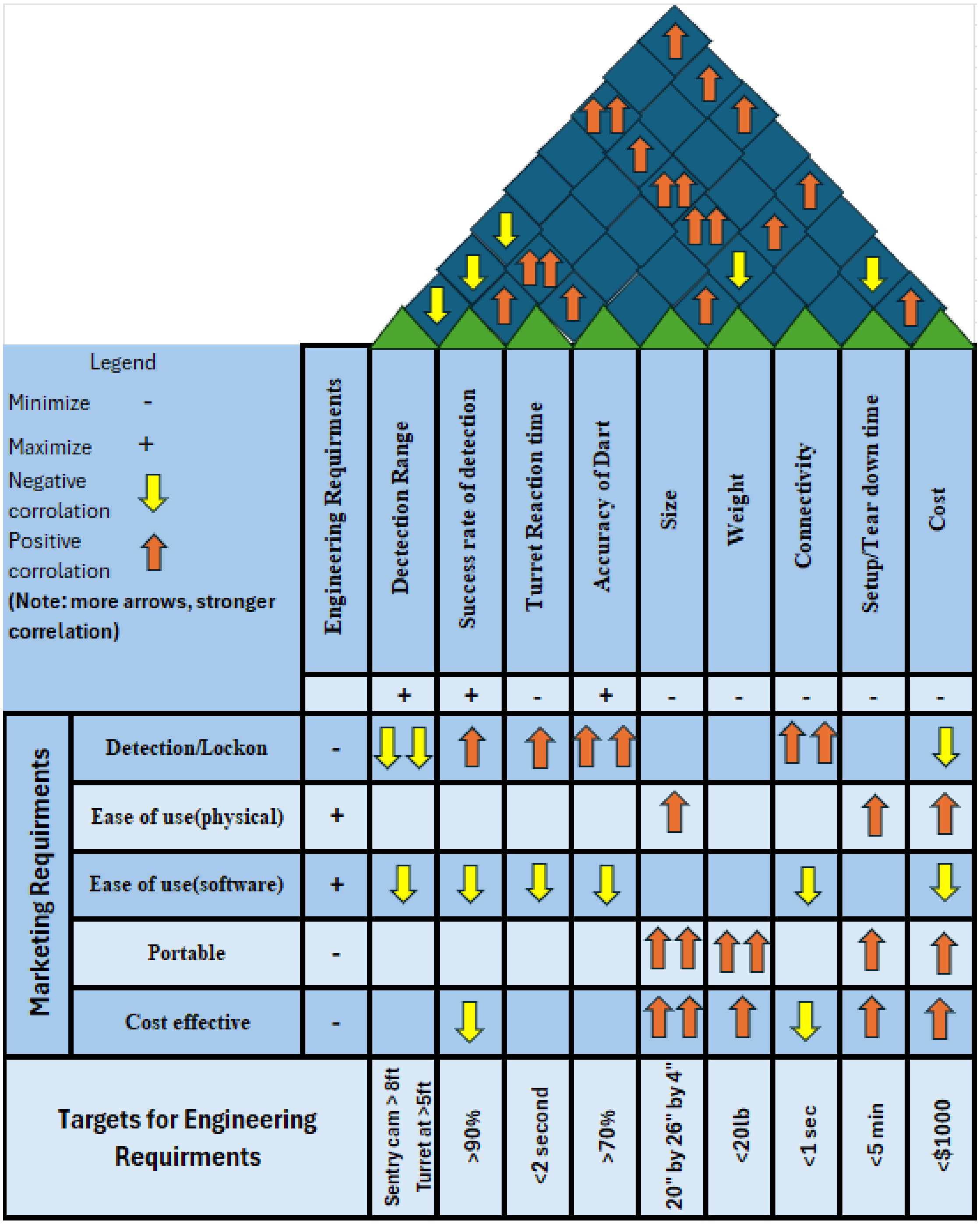
Software diagram – Figure 2



2.5 House of Quality

Below can be found our House of Quality for this project. As we can see there are plenty of interconnecting parts that all depend on each other such as connectivity and detection range of our object. However, cost as we see, can be saved a little from the few changes we have made in this project such as 3D printing when needed.

Figure 1



3.0 Administrative content

This section serves as guidance for how to navigate the roles of this project and the cost of the project as well. This is important for such a project as to see how the balance of workload, not just for the individual person but for the whole team, will look and fair out.

3.1 Bill of Materials (Include budget, PCB costs, high – low, what can cause it to go over or under

Below is our Bill of Materials for this project and the expected expenses for each item.

Table 5 – BOM

Subsystem	Part Name	Quantity	Unit	Estimated Price per [$]
Hardware	Microcontroller	1	1	$5.06
Hardware	Stepper Motor Driver Module	1	1	$10.30
Hardware	High Torque Servo	1	1	$15.90
Hardware	Electronic Speed Controller	1	1	$20.00
All	IMU	1	1	$12.95
Hardware	LEDs	1	50	$7.10
Hardware	Buzzer	1	1	$0.54
Hardware	Laser Diode	1	1	$11.07
Hardware	Stepper Motor	1	1	$22.00
Hardware	Brushless DC motors	2	1	$15.99
Hardware	AC wall power adapter	1	1	$14.99
Hardware	Emergency Button	1	1	$5.90
Hardware	5V Regulator	1	1	$0.44
Hardware	3.3V Regulator	1	1	$0.44
Hardware	PLA Filament	1	1	$17.99
Software	Husky Lens	1	1	$54.90
Software	Cameras (OV5640)	2	1	$11.99
Total				**$255.54**

3.2 Project job description per person

Table 5 – Job Descriptions

Description	Name
Modeling of power PCB and access box controls	*Nick Martucci*
Modeling of 3D printed turret and turret PCB	*Carlos Garcia*

Functionality and compatibility of PCBs working	*Nick Martucci and Carlos Garcia*
Choosing MCU	*Devak Sharma*
Data transfer and saving between sentry cameras and HuskyLens	*Devak Sharma and Christian Solanet*
Motion tracking algorithm	*Devak Sharma and Christian Solanet*
Encoding of LEDs and button on access box and any other peripherals	*Christian Solanet*
Modeling of hardware to hold cameras	*Nick Martucci and Carlos Garcia*
Modeling of access box	*Nick Martucci and Carlos Garcia*
Motor Controls	*Devak Sharma*

3.3 Milestones for SD1 and SD2

Table 6 – SD1 milestones

Project idea and initial research	**Fall Week 1 to Week 3**
Divide and Conquer	Fall Week 2 to Week 3
Deep Research into parts and specs	Fall Week 3 to Week 5
Start drawing schematics and working on initial code generation	Fall Week 5 to Week 8
Midterm report	Fall Week 3 to Week 8
Start Testing physical breadboard and initial code	Fall Week 8 to week 12
Start PCB design work	Fall Week 12 to Week 14
Final Report	Fall Week 8 to Week 16

Table 7 – SD2 milestones

Fully working Breadboard prototype with at least initial laser diode detection of target	**Spring Week 1**
Order initial PCBs	Spring Week 1
Start soldering on components	Spring Week 3
Test all stress cases see how far range can work and redesign PCBs if needed	Spring Week 4 to Week 8
Final PCB Design	By Spring Week 10
Final Demo	Spring Week 16

4.0 References

References

C-RAM research - Project Convergence: Revolutionizing Targeting in Large-Scale Combat Operations

UCF Logo Picture - Ucf Logo, Identity, University, Crest, Emblem PNG Clipart HD

Espressif Systems, “ESP32-S3 Series Datasheet,” Version 2.0. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf

Microchip Technology, “ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash,” Datasheet. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

Raspberry Pi, “RP2350 A Microcontroller by Raspberry Pi,” Datasheet, 2025-07-29 [Online]. Available: https://datasheets.raspberrypi.com/rp2350/rp2350-datasheet.pdf

DFRobot, “HuskyLens AI Machine Vision Sensor,” DFRobot Wiki. [Online]. Available: https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336

Past Project Nerf dart turret – brushless – WiFi – WIP. Printables. , from https://www.printables.com/model/338574-nerf-dart-turret-brushless-wifi-wip

KEB America, “Stepper motor vs. servo motor: Which is right for your application?,” *KEB America Blog*, 2023 https://www.kebamerica.com/blog/stepper-motor-vs-servo-motor-which-is-right-for-your-application/

OMC-StepperOnline, “Brushless DC motors vs. stepper motors,” *StepperOnline Support*. https://www.omc-stepperonline.com/support/brushless-dc-motors-vs-stepper-motors

Anaheim Automation, “Top 3 motor technologies for compact machines: Stepper vs. servo vs. BLDC,” *Anaheim Automation Blog*, 2025. https://anaheimautomation.com/blog/post/top-3-motor-technologies-for-compact-machines-stepper-vs-servo-vs-bldc