Creating an Automated Spotlight Security System for Examining the Effect of Automation of Employability (Technical Topic)

The Balancing Agents between Humanity and Robotic Process Automation (STS Topic)

A Thesis Prospectus in STS 4500 Presented to the School of Engineering and Applied Sciences at the University of Virginia in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Engineering

Joshua Yu

October 2nd, 2023

Technical Project Team Members: Joshua Yu, Kiki Wong, Ethan Cha, Minsol Kim, Iijima Kousuke

On my honor as a university student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Advisors

S. Travis Elliott, Department of Engineering and Society

Adam Barnes, Department of Electrical and Computer Engineering

Capstone Final Project Report

Loose Screws

December, 2023

ECE 4991 Fall 2023

1 Statement of Work

1.1 Ethan Cha

My main role on the project was the embedded software, specifically, the functionality of the motors which plays an integral role in the success of the system. My role included setting up and testing the motors in the early stages which lead to the ability for us to send simple commands to our motors. I worked collaboratively with Kousuke Tapia to come up with the algorithm for communication between the Raspberry Pi and the MSP 432 using a UART signal. I also played a significant role in the integration of the motor system and the image processing system. This included testing and debugging of issues as well as making it more efficient to allow for a smoother tracking experience.

I was also responsible for the creation of the live streaming web application. I set up all the back end that deals with the display of the processed images frames. I worked collaboratively with Kiki Wong to a develop a clean and functional UI for our website. I also developed the back end functionality of most of the manual control buttons featured on the site. This includes the toggle communication between the MSP 432 and Raspberry Pi button and all the manual adjustment buttons that allow the user to move the camera when in manual mode using the up, down, left, and right buttons. I was also responsible for the creation and maintenance of the Gantt chart we used throughout the semester.

1.2 Minsol Kim

My contributions primarily revolved around the software aspects of the project, specifically focusing on the design, implementation, then testing and refinement of the object detection and tracking algorithm. My responsibilities began with the development of software capable of interpreting output from the thermal camera into a meaningful and usable format, then preprocessing the frames for analysis. Following the completion of this task, my focus shifted to iteratively improving the tracking algorithm over the course of the project to more reliably detect the desired

objects while reducing the number of false positives.

I also contributed to building the communication interface between the tracking software and the motor control software, specifically generating motor commands based off the locations of detected objects. Given the processing power constraints of our device, I worked on optimizing the algorithm to provide users with a smoother experience while still streaming the output feed at a high resolution. Finally, I helped design the livestream and video save functionalities on the website.

1.3 Kousuke Tapia

In navigating the multifaceted dimensions of our collaborative project, my role spanned across many categories including the design of the electrical, embedded, and structural systems, as well as crucial administrative responsibilities, encompassing tasks from circuit design and 3D modeling to facilitating team discussions and conserving the momentum of the project. However, my main responsibility was the electrical system, which included the design of circuits and their implementation on a printed circuit board that would provide power to all the components in the system. Subsequently, I made preparations for the fabrication of our PCB and once received, I undertook the meticulous process of soldering the electrical components, followed by rigorous testing and troubleshooting to ensure optimal functionality. As for the embedded sub-system, I played a crucial role in devising the communication protocol between the Raspberry Pi and MSP micro-controller by configuring the UART module with the proper settings. Collaboratively, with Ethan Cha and Minsol Kim, I assisted in the implementation of motor controls and the integration of the imaging software with the motorized components. This interdisciplinary coordination facilitated the seamless convergence of hardware and software functionalities. Lastly, I also assessed our parts needs and adeptly managed the weekly ordering process, thereby ensuring the availability of crucial components for our ongoing endeavors.

Another one of my contributions lies in the structural aspects of the project, in which I designed and 3D modeled the entire housing system including the hardware bay and the motorized

mount for the camera and flashlight. To optimize the layout of components within the housing, I went through multiple design iterations to ensure the development of a functional and aesthetically pleasing structure. Rigorous test fittings were conducted to guarantee the cohesive integration of all components. Finally, I completed orientations at the University's maker spaces to gain access to its printers and proceeded to 3D print the entire housing, ensuring the tangible realization of our envisioned design.

1.4 Kiki Wong

In the course of this project, I played a pivotal role in the software development aspect, spearheading the creation and design of the comprehensive user interface (UI) for our web server. My responsibilities extended to collaboration with Ethan Cha to seamlessly integrate the UI with the motorized controls essential for live streaming. This collaborative effort ensured a cohesive and user-friendly interface, enhancing the overall functionality of our web server.

Furthermore, I actively participated in the early stages of the project's design alongside Josh Yu, contributing to the conceptualization and design of the overall housing. Several weeks were spent trying to design an appropriate inner and outer housing design that would work with the many components implemented into the system. This collaborative effort during the project's initial phase laid the foundation for a robust and integrated system.

Throughout the entirety of the project life cycle, I also assumed the responsibility of designing all of the deliverables presented, ensuring a professional and cohesive visual representation. This encompassed diverse mediums, including presentations, poster sessions, and video presentations. By undertaking the design of these elements, I ensured a consistent, professional, and visually cohesive representation of our project across various platforms. This comprehensive approach aimed not only to showcase our technical achievements but also to effectively communicate the project's value and innovation to diverse audiences.

1.5 Joshua Yu

Throughout the development of this project, my role consisted of the design of the structural hardware, user-facing software, and research of past and future implementations for our design. During the early phases of our project, Kiki Wong and I researched and designed the exterior model of our device alongside its budgetary and physical requirements and limitations. These tasks concern the most cost-effective 3D printed material to construct our prototype with alongside a future design consisting of more expensive materials such as glass and sapphire. These resources would allow both visible light and infrared wavelengths to penetrate the casing while maintaining the structural integrity needed to protect the delicate electronics from weather or tampering.

Once we completed the structural hardware, I worked alongside Ethan Cha and Kiki Wong to construct the web server our live-stream and recorded videos display on. I was involved with creating the HTML pages in Django and Flask that would be used to test the display features. Once the pages were developed, I aided in the prototype functionality of storing video clips and sending the thermal camera's feed to a storage library. Due to storage limitations of our internal Raspberry Pi, a separate method of video storage was needed to keep mp4 files available for retrieval and display in the aforementioned HTML pages.

Finally, during the construction of our final report, I took on the responsibility of balancing all the budgetary calculations in determining how much of our budget was spent alongside the cost of development. These costs are different from the cost of production required in the costs section of this report, which I researched by calculating bulk discounts and eliminating the need to testing components used in the development life cycle. My last contribution was the Intellectual Property concerns regarding previously existing patents and how their existence might impact our product's ability to succeed in a real market. By researching three different patents relating to the lighting, thermal, and security aspects of our project, I was able to determine that our product would be able to thrive without fear of lawsuits for copyright infringement.

Contents

List of Figures

List of Tables

2 Abstract

In the expanding realm of digital advancements, safety has become increasingly accessible to people. Our objective was to contribute to this trend by developing a product aligned with enhancing the security of communities worldwide. Our solution to that problem is Radiance, a surveillance and intruder detection system that uses thermal imaging to detect any life within a designated perimeter, automatically illuminating the entity and tracking its position as it moves. A Raspberry Pi will analyze images from the thermal camera in real-time, identifying potential intruders using their heat signatures and silhouettes. This software analysis will then serve as input to the microcontroller, which will precisely drive motors to orient the system to point at the identified intruder. The light will receive a signal as soon as the intruder is within the illumination zone. The system will also include a web server which will allow users to view the thermal feed overlayed with information from the analysis, as well as be able to manually control the system to illuminate any points of interest inside the perimeter. This project is an example of computer vision and autonomous robotics in security, a field that has expanded into the consumer market from military technology [2].

3 Background

Automated and reliable surveillance technology has a wide range of applications. Government and military institutions have systems that utilize a myriad of sensors and detection systems capable of finding anything from intruders to intercontinental ballistic missiles entering a designated perimeter. Unfortunately, systems such as these are prohibitively expensive for the average consumer. Looking at the commercial market, the options available often struggle with accurate intruder identification under all lighting conditions or are not autonomous and require continuous monitoring in order to achieve real-time detection. Radiance addresses these issues by providing an affordable system capable of automatically and accurately detecting intruders. Furthermore, the ability to illuminate potential targets allows users to quickly locate and engage with intrud-

ers. The task of accurately detecting intruders, whether they be people or animals, requires clear imaging under all lighting conditions. Most commercial surveillance systems such as Simplisafe's Wireless Outdoor Security Camera use a visible light camera with a motion activated floodlight for nighttime surveillance [3]. This is a simple strategy, but often runs into false activations as the motion sensor can be easily triggered by anything from a car driving by or a bird flying through the yard. Night vision cameras such as the Lorex Nocturnal 4 Smart IP Bullet Cameras [4] are another popular option. They are infrared (IR) cameras which work by shining a short wavelength IR light over the entire field of view and detecting reflections off objects[5]. Unfortunately, due to its lack of color, objects are often difficult to identify off of IR video. Furthermore, for real-time intruder detection, the user would have to continuously monitor the video feeds as there is no alert system. Researchers at the Glasgow Caledonian University used image processing techniques to differentiate people and other moving objects from a monochrome video feed [6]. The researchers recognized that most of the time, security personnel are watching nothing happen on the dozens of video feeds they need to monitor. This quickly leads to fatigue and ultimately leads to errors. The goal of this project was to create a tool that would identify potential threats for guards to further investigate such that the personnel wouldn't have to continuously watch the feeds manually. Radiance will build upon the findings of this research project and will apply the same principles in a product more applicable to home security. The use of a more advanced camera as well as the lighting hardware will provide an all-in-one system which will equip users to quickly and precisely detect intruders. This project will draw on countless topics covered throughout the Electrical and Computer Engineering (ECE) coursework at the University of Virginia. PCB and circuit design principles from the Fundamentals series will be critical to the development of this project. Analytical, numerical, and experimental verification of our design will be a significant part of our testing process. The Embedded Computing and Robotics series will also be the foundation for much of the project's design as experience with the MSP432 microcontroller and interfacing hardware with software will be essential during the development phase. On top of the ECE coursework all team members have completed, Minsol has experience with computer vision and object tracking from

a research internship. Kousuke also brings experience designing power supply circuits and Josh completed an internship focusing on security and professional software development practices.

4 Societal Impact Constraints

4.1 Environmental Impact

Radiance exhibits a moderate environmental impact, primarily stemming from the disposal of the device and the manufacturing of its components. Many individual components can be recycled for use in other projects. The device is powered predominantly through a stable power supply, minimizing concerns about batteries. Therefore, the device contains no lithium-ion batteries or solar-powered components, addressing significant concerns for environmental safety. Efforts have been made to configure it with a low-power design. While electronic device disposal lacks federal regulation, guidelines from the United States Environmental Protection Agency (EPA) are available for safe disposal in each state [7].

4.2 Sustainability

Radiance embodies a sustainable design by incorporating a low-power system built for durability and an extended lifetime. Mechanical parts are designed for easy replacement, ensuring long-term repairability. For electrical components, sensors and computing boards such as MSP or Raspberry Pi can be replaced if damaged. The outer housing and motor shelf are constructed using PLA filament, a biodegradable, non-toxic, and recyclable bioplastic [8]. Derived from renewable raw materials, PLA filament is considered more sustainable than traditional fossil fuel-based plastics and will be relatively sustainable to manufacture or recycle.

4.3 Health and Safety

Routine checks are recommended for users to ensure the device functions as expected, as failure to do so may compromise home or staff protection. The current housing design features a top cover that locks into place, minimizing debris ingress. Although not weatherproof at present due to material costs and time constraints, future iterations aim to use more durable materials and secure housing to prevent electrical circuit shorts and weather damage. Measures have been taken to encase all wires securely, mitigating the risk of potential electrocution.

4.4 Ethical, Social, and Economic Concerns

Accessibility is a key consideration in the design of Radiance, aiming for usability by individuals of various ages and abilities. The user interface is designed to be visually pleasing and straightforward to ensure broad adoption in homes, schools, businesses, and other settings. An ethical concern revolves around the continuous operation of the device unless intentionally turned off, potentially capturing individuals passing by who do not wish to be recorded. Furthermore, this may cause some issues pertaining to data privacy.

A notable social concern involves the possibility of Radiance replacing traditional security guards who actively monitor surroundings. Unlike the human presence of security guards, Radiance operates continuously, sparking discussions about the societal implications of this shift in surveillance dynamics.

Moreover, there is a substantial ethical consideration regarding the environmental costs and resource consumption associated with the device's creation. The incorporation of certain costly components, notably the thermal camera, prompts a critical evaluation of whether the benefits of enhanced security justify the environmental impact. Striking a balance between technological advancements and environmental responsibility is imperative for the ethical deployment of Radiance.

5 Physical Constraints

5.1 Cost Constraints

Starting our project with a budget of $500 prompted careful consideration of resource utilization, especially since we needed to be prudent in navigating the inherent challenges posed by the elevated costs associated with thermal imaging devices. Anticipating the substantial expenditure caused by procuring a thermal camera, our strategic resolution for this challenge consisted in reallocating the entire budget for housing components. This calculated decision was aimed at mitigating the financial strain posed by the purchase of a core component. A more cost-effective measure was adopted in the procurement of housing components. Instead of traditional parts acquisition, we leveraged the University's 3D printing facilities, an economical alternative that incurred no additional financial costs.

5.2 Design and Manufacturing Constraints

In terms of software, we did not run into any major direct challenges, as we had availability of all necessary software resources at no expense. Given that the Raspberry Pi handled all imaging processing, the focal point of our concern was with CPU limitations. As discussed in the subsequent sections of this report, numerous challenges arose due to the Pi's lack of processing prowess, forcing us to address them through several software optimizations, ranging from minor algorithmic changes to entire conceptual rework.

On the PCB front, we successfully acquired everything that we needed. However, we did encounter certain hurdles, notably when the manufacturers we engaged lacked the capability to drill specific holes in our design due to their minuscule dimensions. While this predicament was easily resolved by adjusting the PCB layout, it necessitated additional considerations. Another noteworthy constraint surfaced in the form of certain components. The through-hole version of capacitors in the nano Farad range were considerably more expensive compared to their surface-mounted counterparts. This proved to be less than ideal, as the required surface-mounted components were

only available in sizes as small as 0.4 mm, posing a considerable challenge during the soldering process.

5.3 Tools Utilized

In order to successfully complete the project, we used various programs. The following breaks down which tools were used to design each of the sub-systems of our device.

- **Electrical Hardware** - To design the circuits, we consulted the component data sheets and employed Matlab to perform iterative calculations that informed our component selection. This methodology afforded us the flexibility to effortlessly adjust specific parameters within our circuit and observe the impact on the overall design. To facilitate the design of the printed circuit board (PCB) for subsequent fabrication, we used the EasyEDA layout software, which was selected for its advantageous learning curve and the inclusion of an extensive online library of components. Although EasyEDA is capable of running simulations, the absence of simulation models for the chips we used led us to favor testing the circuit on the breadboard directly when possible. Our utilization of EasyEDA was thus solely for the purpose of PCB layout. Several of the electrical components employed lacked readily available footprints. In response, we undertook the acquisition of skills in generating precise footprints based on component data sheets. Furthermore, in order to satisfy the design rule requirements for manufacturing, we acquired an enhanced familiarity with PCB terminology, encompassing terms such as via, annular rings, holes, and other related attributes. Finally, throughout the electrical design process, we consistently relied on NI Virtual Bench for its versatile capabilities, serving as both a multi-meter and oscilloscope. This tool played a pivotal role in seamlessly debugging and validating the functionality of our circuits.

- **Software** - The imaging software employed in our project was exclusively written in Python, leveraging several of its prominent libraries, including OpenCV, numpy, and Pyserial. OpenCV and numpy played integral roles in implementing the image processing capabilities necessary

for tracking, while Pyserial was instrumental in establishing UART communication with the micro-controller. Our chosen integrated development environment (IDE) was Visual Studio Code (VSCode), which was selected for its extensive features conducive to debugging and its seamless integration with Github for version control. In parallel, the software dedicated to motor control was entirely coded in embedded C, utilizing Code Composer Studio—an IDE proprietary to Texas Instruments (TI). The latter offers comprehensive support tailored for development with the MSP432 micro-controller. For the web application, we also wrote it using Python utilizing the Flask framework. This proved to be simple to use while powerful enough for our purposes. For the front end, we used HTML, CSS, and Javascript. While our team possessed fundamental coding knowledge in the aforementioned environments, a familiarization process was still required to adeptly navigate the libraries, particularly OpenCV. Furthermore, the meticulous configuration of serial communication through the UART protocol constituted a comprehensive undertaking, requiring a thorough consultation of the MSP Technical Reference Manual.

- **Structural Hardware** - Most of the components for the housing underwent 3D modeling in Fusion360, and subsequently, Ultimaker Cura was utilized for the slicing process in preparation for printing. As none of us possessed prior proficiency in computer-aided design (CAD), we embarked on the acquisition of skills through the execution of various simple projects. This approach allowed us to familiarize ourselves with the available tools within the software. The predominant elements primarily comprised of simple shapes, thus demanding a relatively modest level of technical expertise. However, the construction of the camera holder presented more complexity. This intricacy arose from the irregular curves and inconsistent geometry inherent in the flashlight to which it latches. Consequently, additional techniques were needed to achieve a successful modeling outcome for this particular component.

All software was readily available for use at no cost since most of them were either free or available through an educational license.

6 External Standards

When designing the whole system, the following standards were taken into consideration for both the hardware and software side of things:

1. *IPC-2221* – The design process of the PCB followed the IPC-2221 standard which is the ”Generic Standard on Printed Board Design” [9]. This standard provides generic requirements to meet industry best practices including considerations in conductor sizes, spacing, impedance control, and thermal.

2. *NEMA Enclosure Standard* – The structural hardware was initially designed to be a NEMA 3R enclosure which is intended for indoor or outdoor use, protects against solid foreign objects and forms of water including rain, sleet, or snow, and is supposed to remain undamaged by the formation of ice on the outside of the enclosure [10]. This type of classification is needed to be able to operate outside as intended but for the purposes of creating a working prototype in the time constraint, the project scope was limited to a NEMA 1 type enclosure which is intended for indoor use and provides a degree of protection against solid foreign objects.

3. *Barr Embedded C Coding Standard* – The Barr Embedded C Coding Standard is a standard that encourages the writing of reliable, maintainable, and portable embedded software in the C programming language [11]. The embedded code in this project closely aligns with these guidelines including suggestions about style, documentation, spacing, etc.

4. *JEDEC Surface Mount Technology Standards* – Because most of the components on our PCB are surface mounted, we used parts that follow the JEDEC SMT package standards [12]. These standards refer to size constraints of surface mounted components.

7 Intellectual Property Issues

There exist several patents that encompass various aspects of this project. For instance, a patent for an Automatic Stage Lighting Tracking System has claims to "multiple detection devices (pressure sensors) on the stage to detect the position of the target in real time, control the illumination direction of the light, and realize the automatic follow of the stage light" [13]. The claims stated previously are outside the realms of our project due to the method of information gathering being through pressure sensors rather than through thermal imaging and motion detection. Also, the patent specifies the usage of stage lights, which are not used in our project in favor of a generic flashlight as the source of illumination in non-stage areas. The patent's claims are independent due to not referring to any prior work that is limited by the claims mentioned in its entirety.

Other patents such as the Sensor System Augmented with Thermal Sensor Object Confirmation also claim innovations that overlap with our project. The aforementioned patent mentions "A thermal camera generates a thermal image of the scene within a field of regard, and a modeling processor coupled to the sensor system and to the thermal camera correlates a position of a selected object in the scene to the field of regard of the thermal camera and queries the thermal camera to confirm a classification of the selected object" [14]. It is worth noting that these claims are also independent due to the intention of the thermal camera and intended software is to capture and identify the classification of a person or object in the thermal camera's field of view. Our project only pertains to the detection and tracking of entities within the thermal camera's field of view with no consideration as to the entity's classification or identity. The patent's claims are independent due to not referring to any prior work that is limited by the claims mentioned in its entirety.

In regards to the intended use of our project, a patent concerning a Mesh-based Home Security System relates to the automated security aspect for our invention. This patent pertains to "a method of receiving, at a network management system and from a first camera of multiple cameras of a home security system, a request to connect the first camera to a communication network for streaming video captured by the first camera to a user device, wherein one or more of the cameras are connected to a base station of the home security system, which streams video captured by the

one or more of the cameras to the user device" [15]. The quote above presents a potential issue with our project's ability to acquire a patent due to its claim over a camera connected to a base station which streams video, similar to how our thermal camera is mounted to a base platform and streams the thermal feed it captures to an external web server. Despite this similarity, these claims may be marked as independent due to the thermal nature of our camera system as well as the patent specifying multiple cameras and sensors strung together into a network management system while our project only utilizing one thermal camera independent of any other home security system. The patent's claims are independent due to not referring to any prior work that is limited by the claims mentioned in its entirety.

8 Project Description

8.1 Performance Objectives and Specifications

The performance of our system depends on many external and internal factors that will limit the capabilities of our system. Based on those considerations, the overarching baseline of our performance goals will be as follows: given that the system is monitoring a 5 m x 5 m x 5 m volume (as depicted in Figure 1), it will be able to shine a light on a single subject that is moving at a top speed of 1.9 m/s across any direction of the room with a precision of 2 cm in either direction in the worst case scenario (refer to technical details for calculations). The chosen speed parameter corresponds to a medium human speed, as defined in a study on various human running speeds [16]. In the event that another subject is recognized by the software, our original intention was to make the light stay on the object that was first recognized, but as further discussed in this report, the system is limited to tracking one subject at a time. As described previously, the user will also be able to access a live feed of the thermal camera through the use of a web server, which will provide a user interface that will allow for the control of certain elements such as the light and movement of the camera.

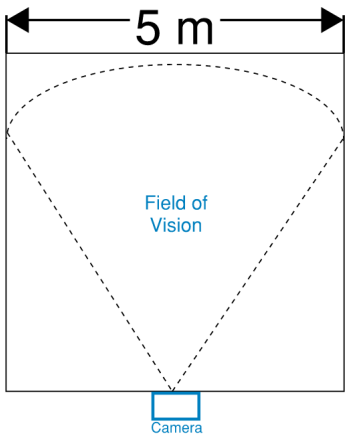

Figure 1: Baseline Performance Camera Setup (Top view)

8.2 How It Works

The final vision of our design consists of several components of equal importance that work together to achieve the performance goals that are stated above. The following diagram shows the high level structure of the system that will be implemented.

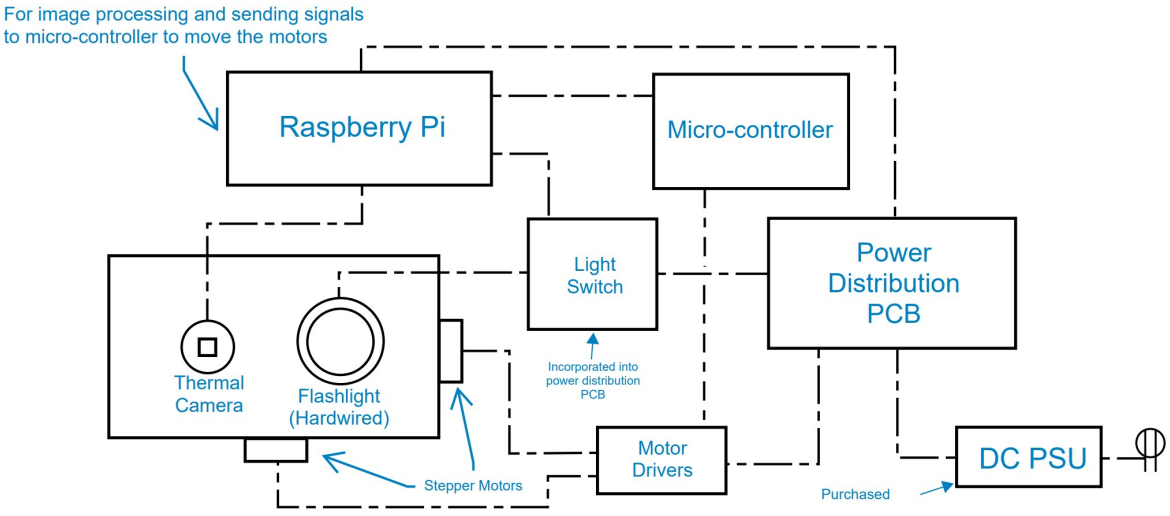

Figure 2: Full System Block Diagram

The central component of our system is a Raspberry Pi directly linked to both the thermal camera and the MSP 432 micro-controller via USB. As the Raspberry Pi processes the video feed from the camera, it will send signals to the micro-controller in real time once a subject is detected,

which will move the motors accordingly using the drivers. At the same time, the Pi will drive the gate of an NMOS transistor high which will be hardwired to the flashlight and allow power to go trough when appropriate. This operation will continue until no heat signature is detected. In order to power all of these processes, a DC power supply will be plugged into a custom power distribution PCB and will supply the appropriate voltages to the Raspberry Pi, the motors, and the flashlight. Both the camera and micro-controller will be powered by the USB ports in the Raspberry Pi. In the following subsections we provide more detailed descriptions of each of the subsystems.

8.2.1 Electrical and Embedded Hardware

This subsystem is comprised of the power distribution PCB, a DC power supply, an MSP 432 micro-controller and two NEMA 17 stepper motors. The DC power supply acquires 120 V from a standard outlet and yields an output of 12 V. Subsequently, the 12 V side is connected to the PCB to power two motor drivers and two buck converters, with the latter efficiently stepping down the voltage to 5 V. This is better illustrated in the block diagram shown in Figure 3.

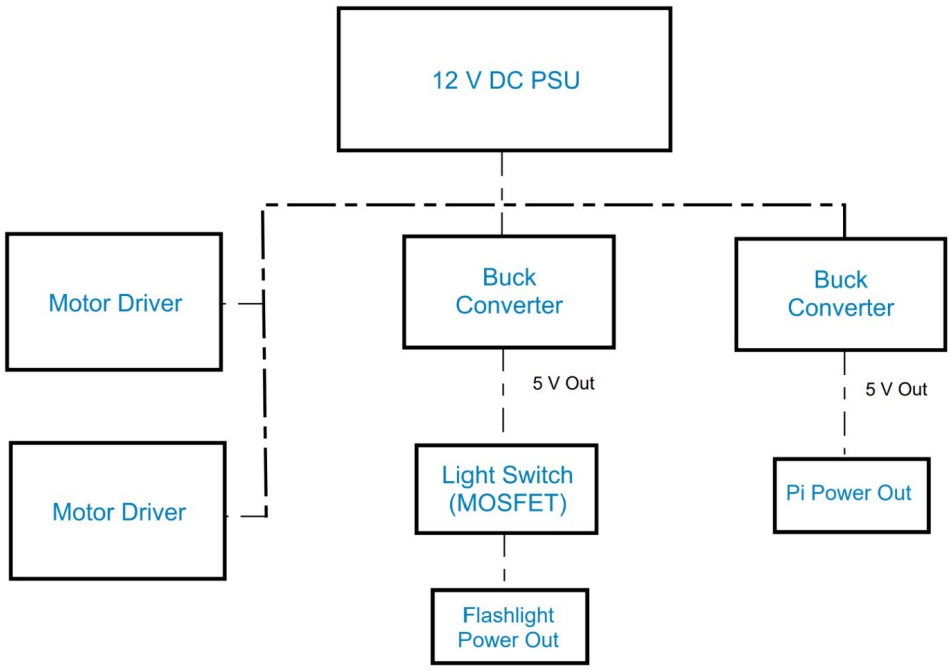

Figure 3: PCB Layout Block Diagram

Upon exploring different motor drivers, we opted for the DRV8825 chip. The external circuitry was meticulously configured in adherence to the data sheet specifications, and the comprehensive schematics are shown in Figure 4 and Figure 6. Each input of the chip is seamlessly interfaced with external components through header pins (Figure 5). The organization of input pins responsible for motor control follows a logical arrangement, facilitating practical control over the motors. To achieve this, the drivers utilize a straightforward PWM signal for speed modulation and a distinct signal for directional changes. These signals, coming from the MSP 432 GPIO pins, are precisely applied to pins 4 and 5, as seen in Figure 5. It is worth noting that the period of the PWM step signal correlates with the motor speed. Additionally, the remaining pins on the headers afford the capability to micro-step the motors (MODE pins), enable them (Reset, Enable, Sleep), and establish connections to the winding of the motors (A1, A2, B1, B2).

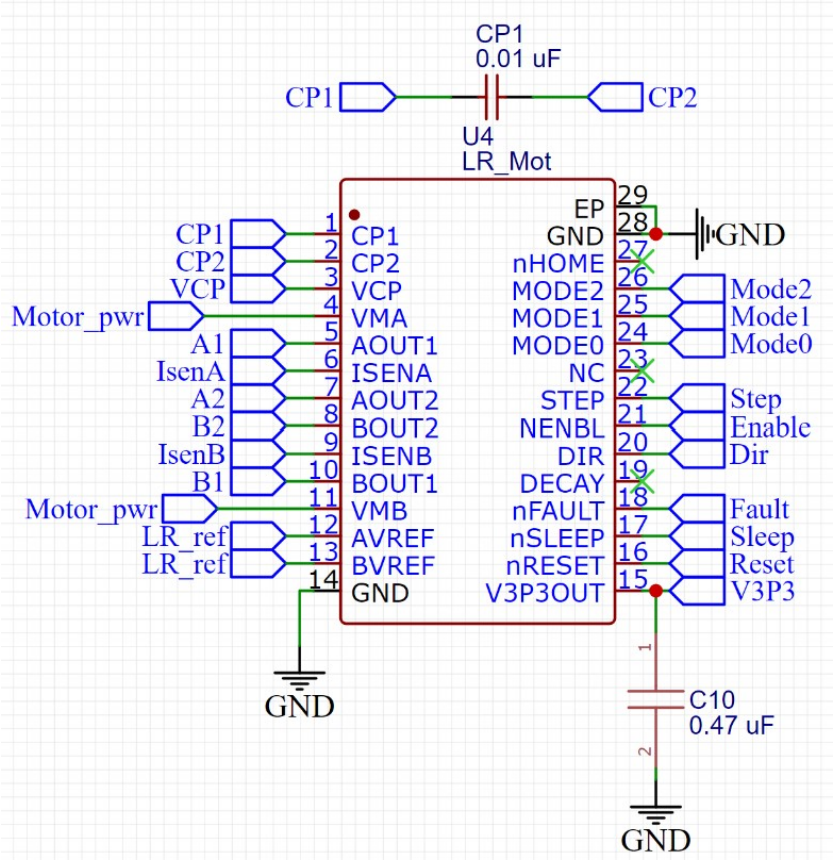

Figure 4: Stepper Motor Driver Schematic

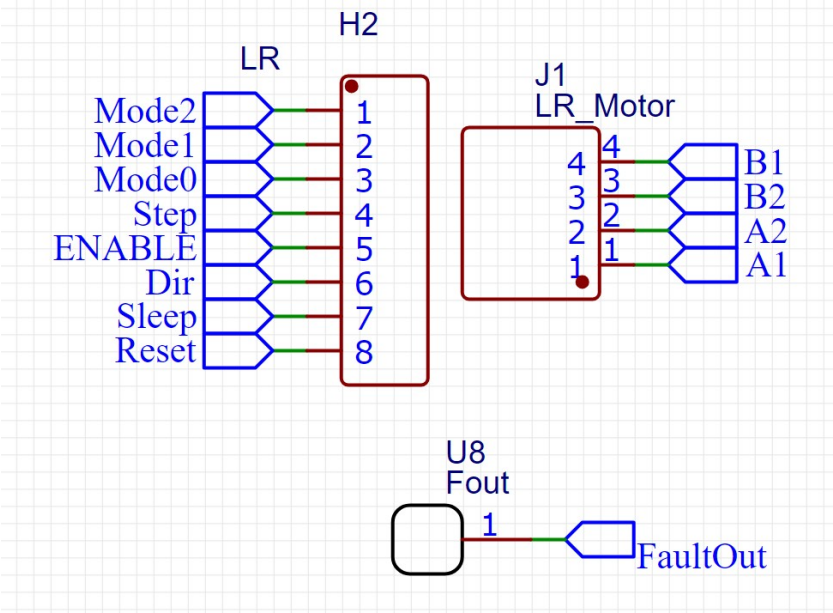

Figure 5: Stepper Motor Driver Inputs (Left) and Outputs (Right) Schematic

The external circuitry plays a dual role, primarily serving to incorporate decoupling capacitors aimed at diminishing noise and safeguarding the chip against voltage spikes. Simultaneously, it functions to restrict the current entering the motor winding. The inclusion of decoupling capacitors is evident in Figure 6 (C3 - C4). For current control, two sense resistors and a potentiometer, illustrated in Figure 6 as R5, modulate a reference voltage which is utilized to set the current limit. Based on the characteristics of the motor being used, some calculations are imperative to determine the appropriate reference voltage, ensuring motor protection. Furthermore, additional resistors in Figure 6 (R9 and R10) serve as external pull-up resistors, facilitating both motor activation under standard conditions and an automatic transition to sleep mode in the event of over-current. Essentially, the Falt_Out signal is driven high to activate the motor during normal operations; however, in the occurrence of excessive current draw, the Fault pin is driven low, concurrently with the Sleep pin, leading to the deactivation of the motor.

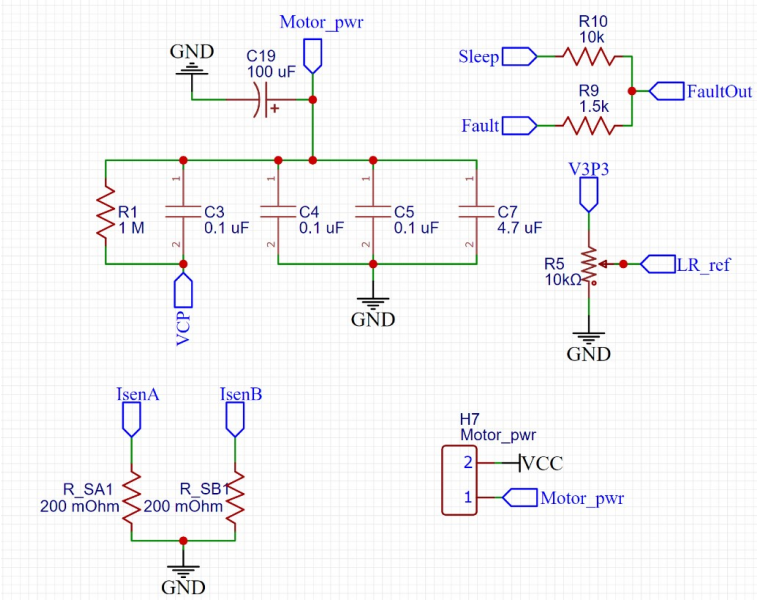

Figure 6: Stepper Motor Driver Schematic (External Circuitry)

Table 1 summarizes the configuration of all the input pins under normal operation conditions. Note that the mode pins are set so that we could micro step the motors 1/8 th step at a time, a decision made to improve the precision and smoothness of the motion.

Pin Number and Function	Status	Signal Source
Mode2 (Pin 1)	Low	MSP 432
Mode1 (Pin 2)	High	MSP 432
Mode0 (Pin 3)	High	MSP 432
Step (Pin 4)	Varies	MSP 432
Enable (Pin 5)	Varies	R Pi
Dir (Pin 6)	Varies	MSP 432
Sleep (Pin 7)	High	MSP 432
Reset (Pin 8)	High	MSP 432

Table 1: Motor Driver Pin by Pin Configuration at Default State

Aside from the motor drivers, the other main components of the PCB consist of two buck converters that simply take in 12 V and output 5 V in order to power the Raspberry Pi and the flashlight. There were numerous choices for switching regulator chips, but we decided to use Rohm's BD86120EFJ-E2 and TI's LMR3364. Figures 7 and 8 show their schematics, respectively.

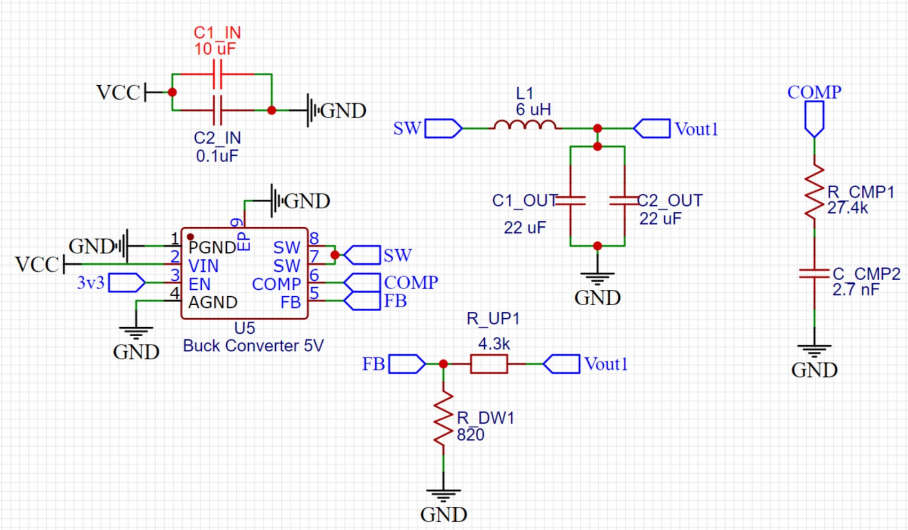

Figure 7: Rohm Buck Converter Schematic

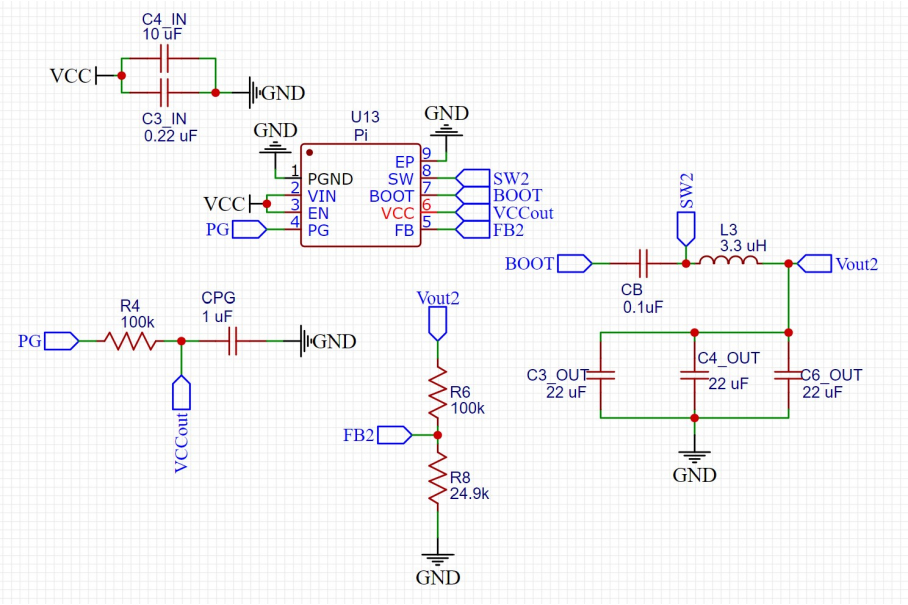

Figure 8: TI Buck Converter Schematic

Despite the utilization of two distinct models of buck converters within our system, a comprehensive examination of their external circuitry reveals much similarity in terms of functionality. A fundamental element in this design is the integration of decoupling capacitors, much like they were used for the motor drivers, they are strategically positioned to proactively mitigate potential instabilities at the inputs of the chips. Both buck converters have properly sized inductors at their

outputs, guaranteeing a sufficiently good current regulation and the output capacitors help to maintain minimal ripples in voltage. It is worth noting that these converter chips offer adjustable output voltage, which requires the implementation of a resistor network on the feedback pin (pin 5 for both converters) to establish a consistent 5 V output. Detailed pin configurations can be referenced in Figures 7 and 8. Finally, both models of buck converters use a dedicated pin for enabling the chip with an external signal, providing a control mechanism capable of selectively cutting power to specific segments of the system. Subsequent to the activation of these chips, the Raspberry Pi and flashlight components are seamlessly powered through header pins, as illustrated in Figure 9. Notably, the negative terminal of the flashlight is attached to the drain of an NMOS transistor whose gate is controlled by an external signal. This strategic configuration of the light source introduces an element of sophistication, effectively creating a software-driven switch. To control that switch, we employed one of the GPIO pins on the Raspberry Pi. Upstream of the terminals for both devices, we placed a physical power disconnect in the form of two header pins. That was done with the purpose of streamlining troubleshooting procedures, so under normal operation a jumper must be placed.

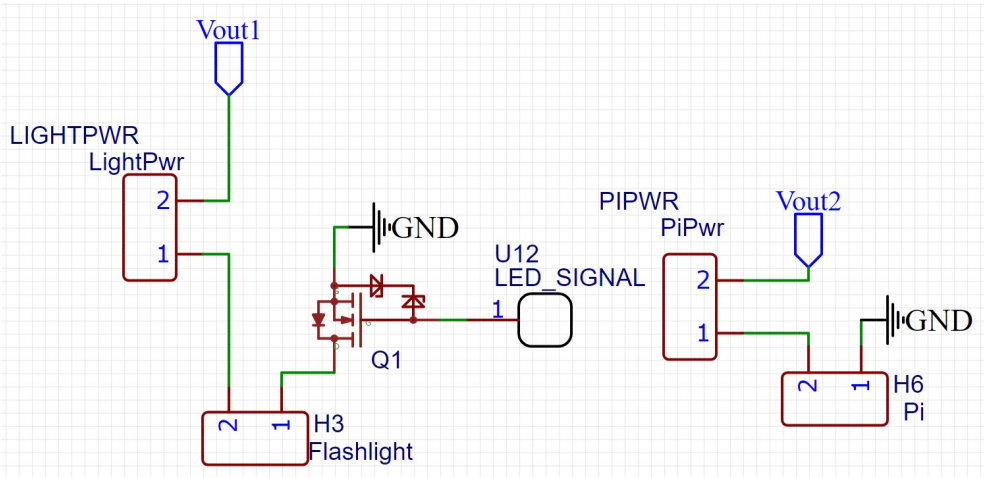

Figure 9: Buck Converter Outputs Schematic

With the completion of the circuit design processes, we proceeded to laying out all the components on the PCB. In Figure 10, the final layout of our board is depicted, providing a clear visual representation of the previously described subsystems. In the top-left corner, there's a DC barrel

female connector for the incoming 12 V power. Along the left side, one will find the two motor drivers with their associated input pins for control and output pins for the motor winding. On the right side, there are the two buck converters. The top one powers the Raspberry Pi, with its terminals in the top-right corner while the bottom one is dedicated to the flashlight since it incorporates a MOSFET for power control.

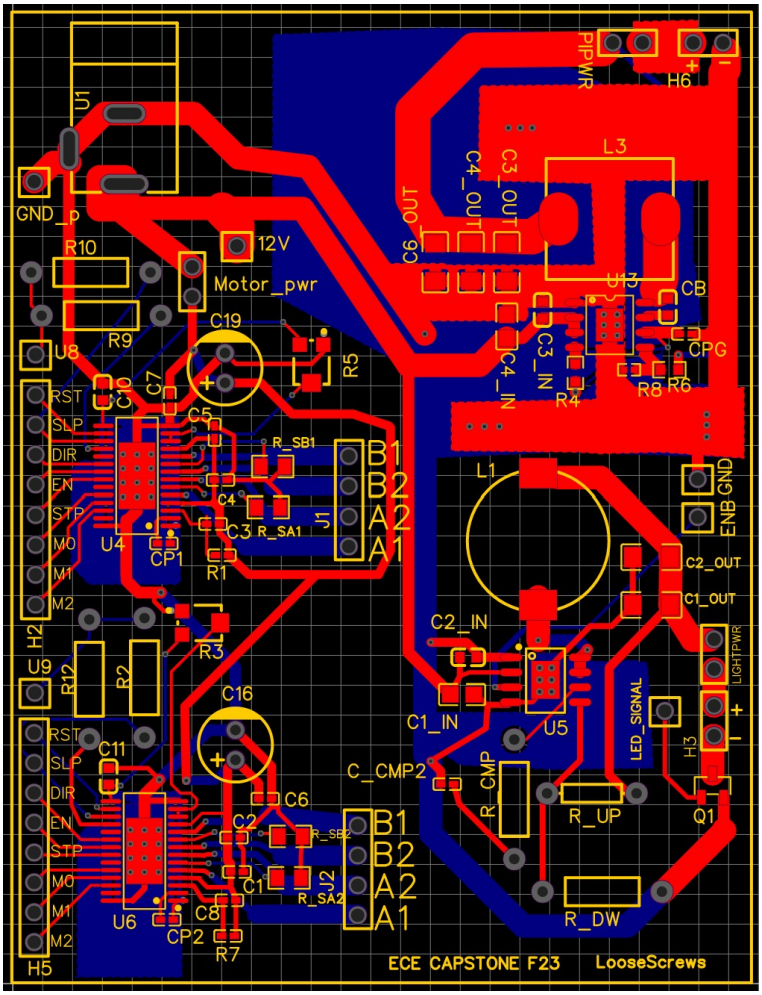

Figure 10: PCB Final Layout

With the completion of the board, the final integration phase for this sub-system involved connecting all the components that require power. Given that the flashlight was initially designed for battery operation, a modification was implemented by soldering wires onto each terminal. These wires, equipped with female headers, were then seamlessly linked to the relevant pins at the output of one of the buck converters. Similarly, the Raspberry Pi's power connection underwent adap-

tation. Utilizing a USB-C cable, we stripped the wires on one end and affixed female headers, ensuring compatibility with the corresponding pins on the board. Any remaining exposed wire was thoroughly wrapped in electrical tape to ensure proper insulation and avoid shorts. For the motor drivers, we started by wiring the input pins that needed to be permanently driven high (as indicated in Table 1) to the 3.3 V output of the MSP 432. Concurrently, the remaining pins were allocated their respective GPIOs on the MSP 432 to facilitate motor control. The motor windings were connected to the header pins depicted on the right side of Figure 6. Given the presence of two motors, a distinct driver was assigned to each, emphasizing the parallel nature of the motor control setup. Figure 11 provides a simplified diagram demonstrating the proper wiring configuration for a stepper motor [17]. Once the correct wire pairs for each motor were identified, they were effortlessly inserted into the header pins in adherence to the labels on the PCB, ensuring a seamless integration into the overall system.

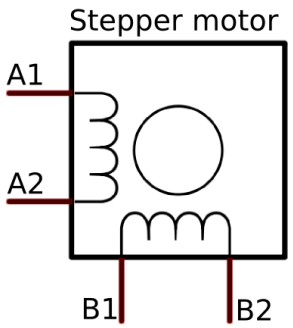

Figure 11: Stepper Motor Diagram

8.2.2 Software

The system detects objects by analyzing a thermal camera feed. It begins by picking out pixels brighter than a specified intensity, which indicate regions of heat. The software then expands these blobs of heat to more closely resemble a body heat signature. These regions are then contoured and bounding rectangles are drawn around them. The program then takes a new copy of the original frame, upscales the resolution by a factor of 2, and draws the rectangular perimeters found in the previous step. All of the image processing is done using OpenCV, an open source computer vision

library [18]. Once objects are detected, the software picks one to track. If multiple are in frame, the first to appear is tracked by default. The object being tracked is outlined in red, while all others are outlined in green. The motors are issued commands that reorient the system to center the object in the camera's field of view. Once the object is centered, the light is powered on, and the target is illuminated.

The MSP 432 is responsible for controlling the two stepper motors that move the flashlight toward the target. The two motors are initialized to two different timers so that both of them can be controlled individually. There is a simple while loop that is constantly looking for the UART signal from the Raspberry Pi. The main function utilizes bit masking and shifting to determine which motors to turn on and the direction of them. It is set up so that the UART signal can send instructions for the "left-right" motors and the "up-down" motors simultaneously. For example, the MSP-432 is able to interpret a "up and left" instruction and execute it at the same time.

The live streaming server utilizes all of the algorithms that the image processing software has already done and simply displays those frames to a web application so that the user can see what the system sees. This includes all the filtering and processing, the boxes that show potential targets, the colors of each of the boxes which indicates whether it is being tracked or not, and the cross hair that marks the middle of the camera. The back end is done using the Flask framework in Python and the front end is done using HTML, CSS, and Javascript. There are several buttons on the web application that perform manual adjustments to the system so that we can control it through the web interface. There are enable and disable motor buttons which activate and deactivate the motors just in case there is malfunction when testing or demoing. There is a toggle communication button that allows the Raspberry Pi to send instructions to the MSP 432. There is also a toggle light which just turns the flashlight on and off manually. Finally, there are four manual control buttons that are able to move the flashlight when the automatic mode is turned off: up, down, left, and right. The web application serves two main purposes. It allows the user to see what the system is seeing and it also allows the user to interact with a user interface to control the system while in use.

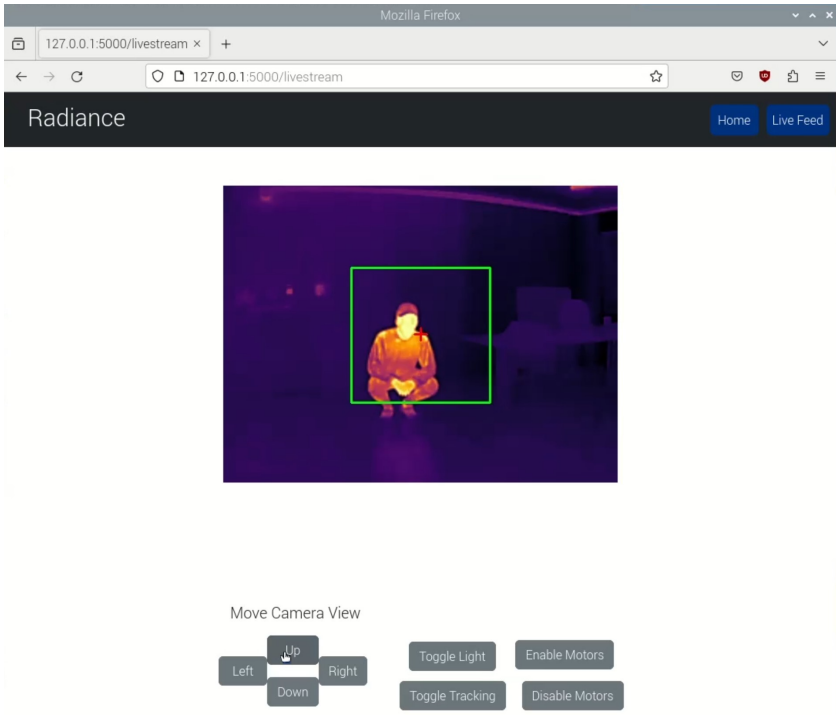

Figure 12: Live Streaming Page in Web Application

8.2.3 Structural Hardware

The main function of the structural hardware in this project was to provide a practical and secure enclosure for all of the components as well as facilitating the two dimensional motion that is essential for tracking subjects with the motors. We began our design by focusing on the turning mechanism first since that presented the most significant challenge. As seen in Figure 13, the design is comprised of a rectangular base that holds two L supports together. Each L support is designed to hold up the platform where the flashlight will be placed, except one of them is distinct so that it can also accommodate the motor that will take care of the y axis movement (up and down). Positioned beneath the rectangular base, is another motor that will cover the x axis movement (left and right). Both motors are attached to their respective structures with a flange that fits comfortably on shaft of the motor. As for the platform, it has a simple design with a curvature in the middle that matches that of the flashlight. This deliberate design choice enhances the attachment to the platform, reinforced by the flashlight fastener located at the top, ensuring a robust and secure fit.

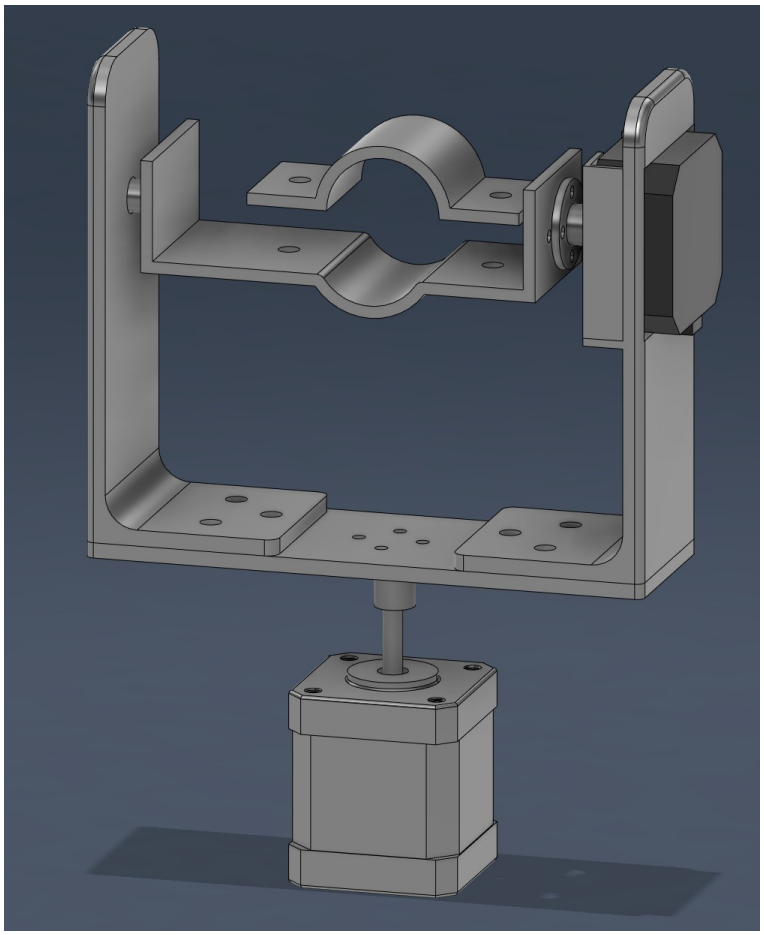

Figure 13: Housing Mount Mechanism 3D Model

Once we finalized the design for the motorized mount, we moved onto devising the hardware bay of the housing, which would contain the PCB, Raspberry PI, and MSP 432. As illustrated in Figure 14, the overall structure is quite simple, yet a substantial amount of thoughtful consideration was required. The design revolved around the motor slot in the middle and all the other components were secured onto the bay as indicated in Figure 15 using circular supports that slot tightly into the mounting holes of all the boards. To enhance the overall functionality and thermal regulation of the housing, two strategically placed fan slots were incorporated at the back. Additionally, the exhaust opening in the front was embossed with our product's name (Radiance), adding a distinct aesthetic touch to the structural design. Finally, the bay also provides the necessary openings for external connectors (video output and power).

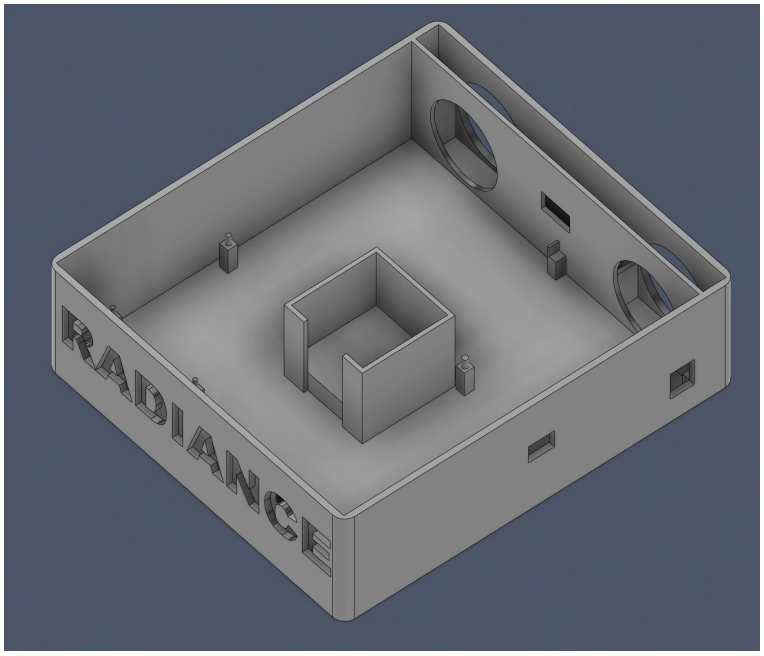

Figure 14: Housing Hardware Bay 3D Model

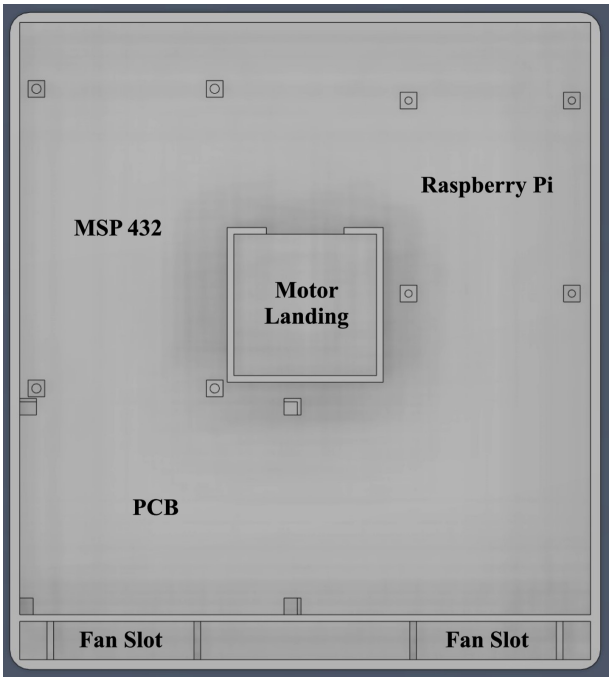

Figure 15: Housing Hardware Bay 3D Model (Top View)

The complete assembly can be seen in Figure 16 in addition to the camera mount that is secured around the head of the flashlight. Note that this design is not meant to allow full 360° rotation in either direction.

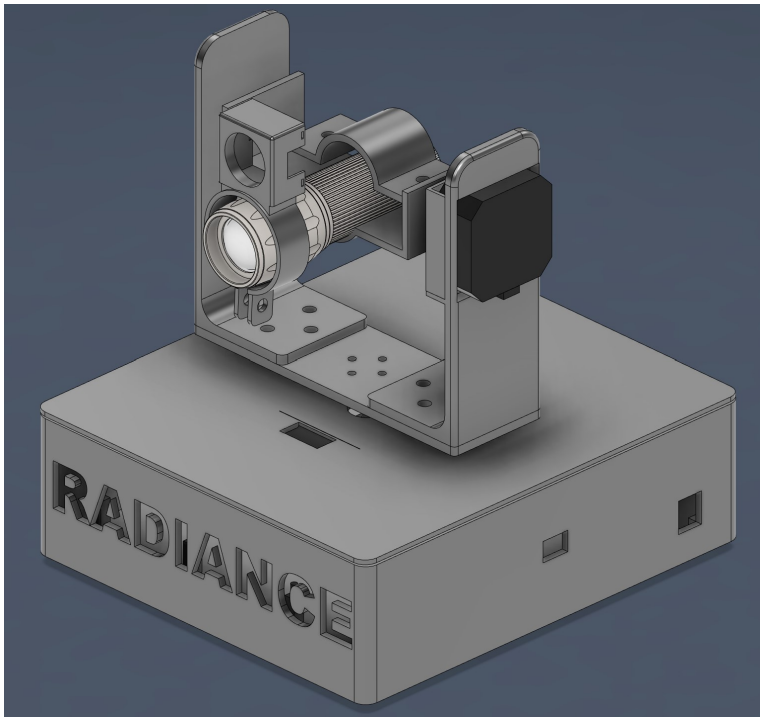

Figure 16: Final Housing Design 3D Model

8.3 Technical Details

Within this section, we provide an elaborate description of the technical design process for each subsystem. This encompasses the reasoning behind our choices, any modifications undertaken, optimizations implemented, and the careful considerations made in navigating trade-offs.

8.3.1 Electrical and Embedded Hardware

To begin our design of this sub-system, we started by carefully assessing the estimated total power consumption of the entire device in order to establish a foundational framework for the subsequent design considerations. More specifically, we aimed to properly size our power supply unit and ensure compatibility with the rest of the components. To determine this, the power draw of each major component was obtained. The Raspberry Pi 4 Model B was our device of choice for image processing and it requires 3 A at 5 V to function [19]. The micro-controller will be powered by one of the Pi's USB 2.0 ports, so assuming that maximum power draw is achieved, it

will consume about 2.5 Watts of power [20]. The selected horizontal motor is rated at a voltage of 3.4 V with 1.7 A of current draw while the vertical motion motor requires 1 A at 3.7 V [21][22]. Both add up to a total of of 9.48 W. The flashlight that is going to be used draws about 2.5 W, conservatively (500 mA at 5 V - measured value). Finally, the cooling fans draw 38 mA at 12 V (0.45 W) [23]. Table 2 provides a summary of these values, with the total power consumption calculated. Notably, the MSP 432 and camera are excluded from the total as they will be powered by the Raspberry Pi, so they are indirectly included in the final calculation.

Component	**Power Draw [W]**
Raspberry Pi	15.00
Motor (horizontal)	5.78
Motor (vertical)	3.70
Flashlight	2.50
Fans	0.91
Total	27.89

Table 2: System-wide Load Calculation

Table 2 used very conservative values and it indicated that a power supply of more than 28 W would be required. Since we had to acquire this power supply in the early stages of the project, we erred on the side of caution and opted for a 12 V supply rated for 60 W. This strategic choice aimed to provide flexibility throughout the design process, minimizing the risk of needing to purchase a different power supply in case we decided to add components or features that might require more power.

Having done a comprehensive assessment of the power requirements of the system, we could now move onto designing the specific circuitry that would distribute the appropriate voltage to each component discussed in Table 2. Since we decided to opt for a 12 V power supply the overall criteria for designing the circuits was to make their inputs be compatible with 12 V. From there, other more specific factors were considered

Regarding the motor drivers, our selection of the DRV8825 was based on several factors, including its 2 A current output, which proved sufficient for our motors. Additionally, the DRV8825

offers a multitude of features that provide extensive control over the motors. These features encompass chip enabling/disabling, microstepping capabilities, overcurrent protection, and a straightforward step and direction interface.

All component values as seen on Figures 4 and 6 were take directly from the application notes found in the data sheet. The exception to that were the sense resistors that were used to set the current output limit that goes into the motors. In order to determine those resistances and calculate the appropriate reference voltage we employed the following expression found in the datasheet[1]:

$$I_{max} = \frac{V_{ref}}{5 * R_{sense}}$$

Where I_{max} is the maximum current allowed, V_{ref} is the reference voltage applied to pins 12 and 13 as seen in Figure 4, and R_{sense} is the sense resistor value chosen. The DRV8825 has a 3.3 V output pin, so that was used along with a resistance to set the appropriate reference voltage. Instead, of using a voltage divider with fixed values, we opted for using a 10 kΩ potentiometer to accommodate motors with a wide range of current capabilities. Given that the rated current for the horizontal motion motor is 1.7 A, the appropriate reference voltage needed is:

$$V_{ref} = I_{max} * 5 * R_{sense} = 1.7 \text{ A} * 5 * 200 \text{ m}\Omega = 1.7 \text{ V}$$

Similarly for the vertical motion motor with a rating of 1 A, the reference voltage is:

$$V_{ref} = I_{max} * 5 * R_{sense} = 1 \text{ A} * 5 * 200 \text{ m}\Omega = 1 \text{ V}$$

Taking a closer look at those calculations reveals the reasoning behind choosing a 200 mΩ resistor. The $5 * R_{sense}$ factor becomes 1, which allows the rated current to always be equal to the reference voltage needed. This avoids the need to make a calculation every time. Theoretically, with this setup, the maximum current allowed is 3.3 A, yet it is crucial to note that the drivers are specifically rated up to 2 A. Consequently, meticulous attention is necessary to ensure that the

reference voltage is set within the prescribed limits and does not surpass 2 V.

The second major components of the electrical system are the buck converters that would be used to power the Raspberry Pi and Flashlight. This choice of switching regulator was guided by our objective to make the system as efficient as possible. According to a study published by the IEEE, buck converters were found to have and efficiency as high as 92%, making it an excellent choice to step down the voltage from 12 V to 5 V [24]. As the Raspberry Pi is a more sensitive computing device requiring a good quality supply, we began the design taking its requirements into account. As previously mentioned, considering a worst-case scenario where the Raspberry Pi utilizes 3 A at 5 V, it became imperative to select a converter with the appropriate current rating. After careful consideration, we chose Rohm's BD86120EFJ-E2, which boasts a maximum rating of 5 A—well beyond our anticipated requirements. Given that alternative chips with lower current ratings were available at a similar price point, the decision to opt for a component with a higher capacity was made strategically to proactively enhance the system's robustness. Two critical characteristics that demanded assessment were the current and voltage ripple requirements to ensure stable operation. Lacking specific details in the datasheet, we employed the official Raspberry Pi 4 power supply as a benchmark, which specifies a 120 mV peak-to-peak ripple [25].

Now that we obtained all the necessary information we started designing the component values using the following formulas taken directly from the datasheet [26]:

$$L = V_{out} * (V_{out} - V_{in}) \frac{1}{V_{in} * F_{osc} * \Delta I}$$

$$V_{RPL} = \Delta I * (R_{ESR} + \frac{1}{8 * C_{out} * F_{osc}})$$

$$V_{out} = \frac{R_1 + R_2}{R_2} * 0.8$$

$$R_{cmp} = \frac{2\pi * V_{out} * F_{crs} * C_{out}}{V_{FB} * G_{MP} * G_{MA}}$$

$$C_{cmp} = \frac{V_{out * C_{out}}}{I_{out} R_{cmp}}$$

Several values were taken directly from the datasheet as a good and balanced recommendation. Those include the input capacitors, feedback resistors, current ripple, crossover frequency, internal feedback voltage, current sense gain, and error amplifier trans-conductance. As for the rest, the appropriate values were used to calculate the final values of the remaining components. All variable definitions as well as component values are shown in Table 3.

Variable	**Definition**	**Value**
L	Output inductor	6 µH
C_{OUT}	Output capacitance	44 µF
V_{RPL}	Output ripple voltage	13.6 mV
V_{IN}	Input Voltage	12 V
V_{OUT}	Output voltage	5 V
R_{CMP}	Phase compensation resistor	27.4 kΩ
C_{CMP}	Phase compensation capacitor	2.7 nF
I_{OUT}	Output current	3 A
ΔI	Ripple current	0.9 A
F_{OSC}	Switching Frequency	550 kHz
R_{ESR}	Capacitor Effective Series Resistance	0.01 Ω
R_1 and R_2	Output feedback Resistors	4.3 kΩ and 0.8 kΩ
F_{CRS}	Crossover frequency	55 kHz
V_{FB}	Internal feedback voltage	0.8 V
G_{MP}	Current Sense Gain	8.6 A/V
G_{MA}	Error Amplifier Trans-conductance	400 µA/V

Table 3: Rohm Buck Converter Variable and Component Definitions

The evaluation of the buck converter, as outlined in Table 3, shows a ripple voltage of 13.6 mV. This outcome not only meets but surpasses the 120 mV specification of the official Raspberry Pi power supply. Notably, in the expression for V_{RPL}, the impact of R_{ESR} is significant. To optimize our design, we implemented two 22 μF capacitors in parallel instead of a single 44 μF capacitor (see Figure 7). This strategic choice effectively halves the effective series resistance, resulting in a substantially lower ripple voltage.

While a single buck converter theoretically possesses the capacity to power both the Pi and flashlight due to its 5 A rating, we opted for an additional converter for the flashlight to ensure the stable operation of the Raspberry Pi. The constant switching of the light could potentially lead to voltage drops that may impact the Pi. Although the power quality required for the flashlight is lower compared to the Pi, the decision to employ a separate converter was based on the reliability it adds to the system. Leveraging the existing design, we integrated the same buck converter for the flashlight, with the addition of an NMOS transistor at the output terminals (refer to Figure 9). This configuration enables the flashlight to be controlled through software using the Raspberry Pi's GPIO pins. For this purpose, we selected a transistor with sufficient current capacity and a gate threshold voltage achievable by the GPIO pins. The chosen transistor has a maximum current of 500 mA and a gate-to-source threshold voltage of 0.9 V [27], ensuring effective control of the switch.

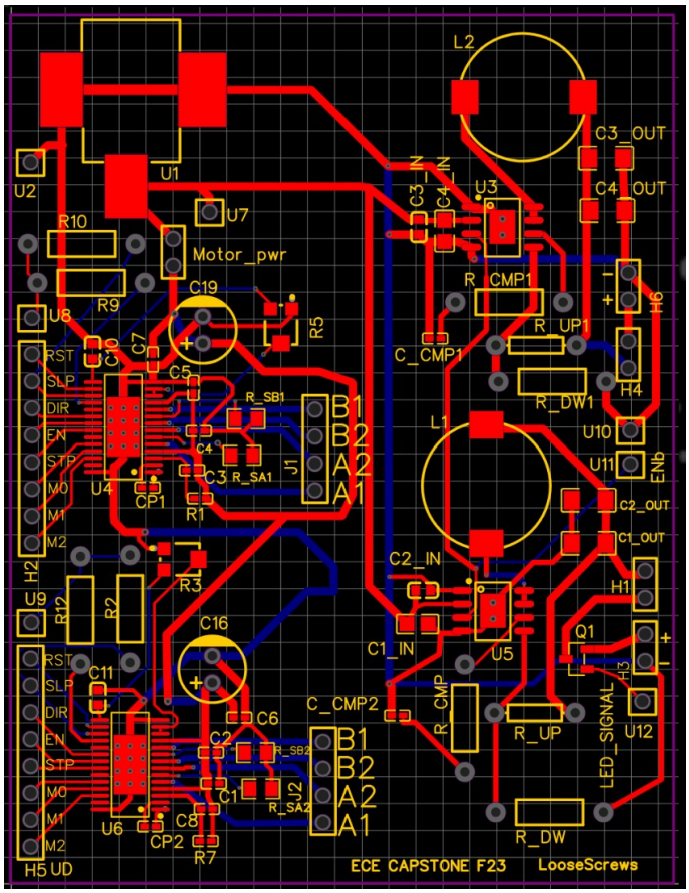

Figure 17: PCB Initial Layout

Due to the diminutive size of the chips and external components, it was hard to test them on a breadboard, so we had to wait for the PCB to arrive. As mentioned before, the optimal layout of each component consisted of the motor drivers being on the left side, while having two buck converters on the right side. Figure 17 shows what the first iteration of our board looked like.

Upon soldering several components, we successfully confirmed the functionality of the motor drivers. However, we encountered challenges with the buck converter as it outputted the correct voltage, but under a load exceeding 1 A, it experienced a voltage drop to mitigate the current draw. Troubleshooting steps were initiated, including a meticulous review of the schematics to ensure alignment with the PCB's electrical connections. The datasheet was revisited, and all component values were recalculated, yet no obvious issues were identified until the realization that the chip was overheating. To address this, we added a heat sink, allowing us to draw approximately 1.3 A of current—an improvement, albeit still below the required specifications. Nonetheless, that lead us to identify that it was a temperature-related issue with the chip, stemming from the initial PCB layout's suboptimal design. Recognizing the thermal inefficiency, we researched ways to enhance the board's thermal performance. The one critical improvement that was identified was incorporating copper areas on the top and bottom layers as heat sinks and connecting them through vias. The changes implemented in the second iteration of the board, as illustrated in Figure 18, reflected these enhancements.

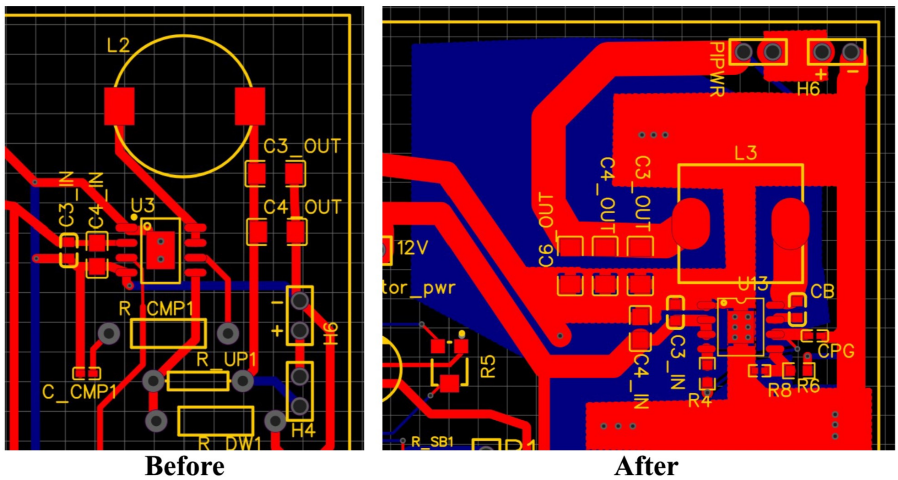

Figure 18: PCB Initial vs Final Buck Converter Layout

To address the thermal concerns with the buck converter, we strategically connected both copper areas to the ground pad beneath the chip. Additionally, we introduced extra vias to enhance the thermal transfer of heat from the top layer to the substantially larger bottom copper layer. This modification aimed to optimize the dissipation of heat, rectifying the initial thermal issues. Notably, the revision involved a transition to Texas Instrument's LMR33640 from the previous Rohm chip, as the former offered similar capabilities, but the manufacturer provided more comprehensive guidance on the recommended PCB layout. As a precautionary measure, this switch was made, with both the component values and layout drawn directly from the datasheet application notes [28]. The final component values can be observed in the schematic provided in Figure 7.

The final consideration for the PCB design was the placement and width of the traces. Since each of the components draws a relatively high current, we made sure to use separate power rails for each subsystem. As for determining the trace width necessary, we simply used the trace width calculator provided by Advanced Circuits [29]. Signal traces were uniformly set at 10 mil, while the width of the power traces varied based on the results of the calculator. Most of the power traces required a width that was greater than the pin separation of the chips, which meant that it would be physically impossible to fit the traces within that confined space. To address this, we employed a strategic approach: commencing with thin traces at the pin connections and gradually widening them as the traces expanded.

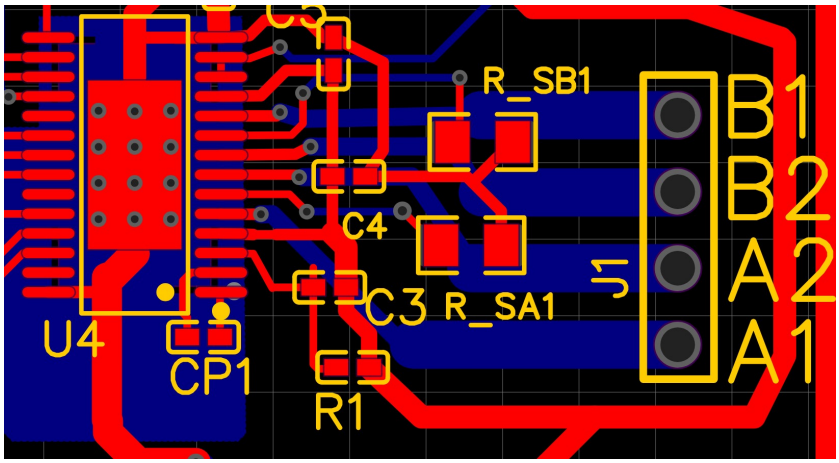

Figure 19: PCB Motor Driver Winding Connections

This technique served a dual purpose, mitigating both spatial constraints and potential thermal issues arising from thin traces. An exemplification of this approach is evident in Figure 19 (blue traces connecting pins B1, B2, A1, and A2), showcasing the traces connecting to the motor windings. These traces initiate with a thin configuration at the pin connections and progressively widen as available space allows.

In parallel with the PCB design process we also assessed some of the key factors that would ultimately impact the performance of our system. Those include motor precision, speed, and torque. When evaluating motor precision, we referred to the data sheet, which specifies a stepping angle for the motor of 1.8 degrees [21]. This implies that the motors can only move in increments of 1.8 degrees. Leveraging this information, we can derive an expression to quantify the level of precision with which the motors can orient the flashlight at a given distance:

$$L = \text{Distance from subject}$$

$$X = \text{Linear stepping distance}$$

$$\theta = \text{Stepping angle}$$

$$\tan\theta = \frac{X}{L}$$

Given that our performance goal is at a distance of L = 5 m, the linear stepping distance is:

$$X = L * \tan\theta = 0.156m$$

This implies that, in the worst-case scenario, the flashlight will be oriented with a precision of about 16 cm. While this level of precision is acceptable for human-sized subjects, tracking smaller objects may result in ineffective illumination. Initial testing of the motors at a standard stepping rate revealed excessive vibration in the mounting structure, accompanied by an unpleasant noise. This vibration adversely affected the camera's ability to capture steady frames for processing.

To address this issue, we adjusted the motor drivers to operate at a reduced rate of 8 micro-steps per step. In order to do that we simply set the pins to the appropriate level as outlined in Table 4. This modification significantly diminished vibrations, ensuring smoother operation and minimizing disruptions to the camera's frame capture. Furthermore, this adjustment enhanced the precision of object tracking, as the angle per step was reduced to one-eighth of its normal value. Applying the previous expression for linear stepping distance, the final precision of our motors at a distance of 5 meters then becomes:

$$X = L * \tan\frac{\theta}{8} = 0.02m$$

MODE2	MODE1	MODE0	STEP MODE
0	0	0	Full step (2-phase excitation) with 71% current
0	0	1	1/2 step (1-2 phase excitation)
0	1	0	1/4 step (W1-2 phase excitation)
0	1	1	8 microsteps/step
1	0	0	16 microsteps/step
1	0	1	32 microsteps/step
1	1	0	32 microsteps/step
1	1	1	32 microsteps/step

Table 4: DRV8825 Stepping Format [1]

In order to evaluate the RPM requirement of the motor, one must consider the worst case in which the camera will need to rotate to track a subject that is very close to the camera and is moving from one bottom corner (as shown in Figure 1) to the other at S = 1.9 m/s. As a conservative approximation, one can assume that the required angle of rotation for this scenario is θ = 180 $^\circ$ since the subject is very close to the camera. With this information we perform the following calculation:

$$t = \frac{L}{S} = 2.63s \text{ (Time required to traverse length of the surveyed area)}$$

The RPM required is then:

$$\text{RPM} = \frac{\theta}{t} * \frac{60s}{360^\circ} = 15.8 \text{ RPM}$$

While the relatively low RPM was not anticipated to pose significant issues, as it comfortably falls within the physical limits of our chosen motors, a critical aspect required verification. We needed to assess the capabilities of both the motor drivers and the micro-controller to ensure they could effectively drive the motor at this speed. To determine this, our first step involved calculating the required number of steps per second to achieve the desired angular speed of 15.8 RPM:

$$\omega = 15.8 \text{ RPM} = 94.8^\circ/s$$

$$\theta = \frac{1.8^\circ}{8} \text{ (Micro-stepping angle)}$$

$$\text{Steps per second } = \frac{\omega}{\theta} = 422$$

Considering that with the DRV8825 drivers, each pulse of a PWM signal corresponds to one step, we can deduce that a PWM frequency of 422 Hz is necessary to attain a speed of 15.8 RPM. As stated in the datasheet, the maximum step frequency that the driver can handle is 250 kHz [1]. Moreover, given that the MSP432's maximum clock frequency is 48 MHz, we surpass the required capabilities by many orders of magnitude [30]. This analysis ensured that the motors can operate at the intended speed without any constraints. The last important motor characteristic that we had to account for was the torque, the selected motor for horizontal movement is rated for a holding torque of 3.5 kg * cm while the one for vertical movement is rated at 1.6 kg-cm [22][21]. Both comfortably accommodate the torque needed to rotate the camera and flashlight.

8.3.2 Software

The object detection software begins by thresholding the frames that are outputted by the thermal camera for pixels above a specific intensity. After extensive testing, a value of 236 was found to be the most suitable for reliably isolating the heat signatures of people, assuming someone was in the frame. These bright spots indicate locations of heat, but due to a normalization feature built into the camera, all readings are relative. The thresholding process returns a binary image, meaning all pixels above the given threshold are set to 255, the maximum value, while all other pixels

are set to zero.

The external temperature of a human body isn't uniform, with limbs typically being significantly cooler than the head or chest. Furthermore, clothes also significantly dampen the heat given off by a person's skin. In order to most accurately detect the heat signatures of people, the thresholding value was set high to only pick up regions of the core body temperature. To compensate for regions with lower heat, the blobs of heat isolated in the preceding step go through a dilation process using 9x12 tiles over 7 iterations. As a result, these heat blobs are expanded to now cover most of the body's area.

The next step involves contouring these blobs, then finding the rectangular regions bounding each one. If any of the rectangular perimeters overlap, they are merged. Since the output of the dilation step is still a binary image, the rectangles aren't immediately drawn onto the frame being processed. Instead, a copy of the original frame from the camera is used. This frame is first upscaled from 256x192 to 512x384 using linear interpolation then a color map is applied to aid in delineating details. The locations and sizes of the rectangles found earlier are also upscaled to match the new resolution before being drawn.

```python
def detect_heat_signatures(frame, max_rect_area):
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    if should_detect_objects(gray):
        threshold_value = 236
        _, thresh = cv2.threshold(gray, threshold_value, 255, cv2.THRESH_BINARY)
        contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        cv2.imshow("Thresholding", thresh)

        for contour in contours:
            x, y, w, h = cv2.boundingRect(contour)
            rect_area = w * h
            blob_area = cv2.contourArea(contour)
            aspect_ratio = w / h if h > 0 else 0

            # Check for size and aspect ratio criteria
            if blob_area / rect_area < 0.3 or aspect_ratio > 6:
                cv2.drawContours(thresh, [contour], 0, (0, 0, 0), -1)
        cv2.imshow("Adjusted", thresh)

        kernel = np.ones((12, 9), np.uint8)
        adjusted_threshold = cv2.dilate(thresh, kernel, iterations=7)
        cv2.imshow("Dilation", adjusted_threshold)
        contours, _ = cv2.findContours(adjusted_threshold, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        rects = [cv2.boundingRect(contour) for contour in contours]

        return merge_overlapping_rectangles(rects)
    return []
```

Figure 20: Detecting Heat Signatures in a Frame

The particular thermal camera used was outputting images in RGB format, though the output was grayscale. All processing steps were done at the native 256x192 resolution using grayscale encoding for images, which prevented any data loss while still maintaining a low enough resolution for the Raspberry Pi to process in real-time. The upscaling is only done in the final step, right before streaming output to the UI.

As mentioned above, the normalization function of the camera made it impossible to assign specific pixel intensities to set temperatures. This quickly became problematic as a pixel in the frame with an intensity of 236 could represent anything from 90°F to 50°F, depending on what was in the field of view. To resolve this, a check was written into the program to take the average of all pixels below the set threshold and compare this value to a lower-bound value. If this averaged value was high, that indicated that no heat source was present, and the camera was normalizing a much smaller temperature range (just the background) across the entire 0-255 range. After extensive testing, this lower-bound value was set to 64.

Figure 21: Intermediate Steps of Image Processing Algorithm

The object tracking portion of the algorithm starts by first selecting one of the detected objects to track. If more than one are in frame, the first detected will be tracked by default. The location of the object is first stored. Once the next frame is processed and all the objects have been located

again, the object closest to the last recorded position is assumed to be the same object. This process is done continuously, tracking each object until it is lost, at which point, the program looks for a new target.

The object being tracked is outlined in red while all others are outlined in green. If the center of the object aligns with the center of the camera (within 15 pixels), the light is pointed at the object, at which point the light is powered on. Instructions for motor control are determined using the location of the target object relative to the center of the frame. Unlike the light control, there is no 15 pixel buffer zone, meaning that the motors will continuous move to keep the object centered, even while the light is on.

The chosen communication protocol for this project was UART, providing asynchronous two-way communication. While not the most advanced protocol available, it proved sufficient for transmitting motor instructions, requiring a simpler scheme. The module was configured with a baud rate of 9600, and detailed settings for the UART module can be found in Figure 39 in the Appendix.

For the set up of the motors, each of the stepper motors were connected to the GPIO pins on the MSP 432. The ”left-right” stepper motor step lead is connected to P2.5 and the direction lead is connected to P1.6 on the MSP. The ”up-down” stepper motor step lead is connected to P2.6 and the direction lead is connected to P6.7. All four are set as outputs. TimerA0 is set to the ”left-right” motor and TimerA1 is set to the ”up-down” motor. As seen in the figure below, the UART signal is masked and shifted to determine which instruction was sent by the Raspberry Pi. 0x00001000 corresponds to an ”up” command, 0x00000100 corresponds to a ”down” command, 0x00000010 corresponds to a ”left” command, and 0x00000001 corresponds to a ”right” command. The instructions were implemented with the Raspberry Pi continuously sending the necessary movement commands. Upon achieving the centered position of the subject, the corresponding stop command was then transmitted to either the horizontal or vertical motion motor.

```
while (TRUE) {

    if(EUSCI_A0->IFG & 0x01){ // if data is received then the following is executed

                command = ReceiveChar(); // reads instruction from python program

                if ((command & 0x02) >> 1) { //bit mask for left bit and shift 1
                    LR_MOTOR_DR_HIGH;
                    LR_START; // start pulse
                }
                else if ((command & 0x01)){ //bit mask for right bit no shift
                    LR_MOTOR_DR_LOW;
                    LR_START; // start pulse
                }
                else {
                    //stop
                    LR_STOP; // stop pulses
                }
                if ((command & 0x08) >> 3) { //bit mask for left bit and shift 1
                    UD_MOTOR_DR_HIGH;
                    UD_START; // start pulse
                }
                else if ((command & 0x04) >> 2){ //bit mask for right bit no shift
                    UD_MOTOR_DR_LOW;
                    UD_START; // start pulse
                }
                else {
                    //stop
                    UD_STOP; // stop pulses
                }
                //confirmation
                SendString("Changing command...\n");
            }
}
```

Figure 22: Motor Control Software

8.3.3 Structural Hardware

Leveraging additive manufacturing through 3D printing yields numerous advantages, notably affording complete control over the final product's design. However, the ultimate factor steering our decision towards 3D printing was rooted in a pragmatic assessment of our budgetary considerations. Foreseeing that allocating funds for construction materials might strain our financial resources, we strategically opted to utilize the cost-free 3D printing facilities available at the University's maker spaces. This strategic choice not only conserved our budget but also granted us flexibility and margin for error in the development of our other sub-systems. Nonetheless, we still experienced certain challenges that required multiple trial and error iterations of design.

Once we decided to move forward with 3D printing, we had to assess the capabilities of the printers that would be used to create our components. The maker spaces at the Chemistry building are equipped with Ultimaker S5 printers, which are on the low end of industrial level printing [31]. For a prototype level project like ours, the S5 printers display amazing capabilities that provide everything we would need. The ones available at the maker space came equipped with a 0.4 mm nozzle width and an ability to print layers as small as 0.1 mm [32]. Those two parameters are relevant when considering the scale of the components that need to be printed. For the majority

of our components, these printing capabilities proved more than sufficient, ensuring the accurate reproduction of our model. However, a notable challenge emerged in the creation of the camera mount lid, where having precise slots and tabs was paramount for secure attachment. Initially, we encountered difficulty achieving a secure fit for the lid, attributing it to discrepancies in dimensions. The issue stemmed from one of the clips having dimensions that were physically impossible to print, due to the layer height that was utilized. Swift resolution came in the form of adjusting the dimensions to be a multiple of the layer height, effectively mitigating the discrepancy and ensuring the successful printing of the camera mount lid, albeit after several trials.

Another important parameter to consider was the total build volume of the printer, which is 330 x 240 x 300 mm for the S5. That is a relatively large volume so we had no issue printing any of our components in one piece. In pursuit of a highly stable structure, our initial design incorporated notably thick L supports, as depicted in Figure 23. However, as we attempted to print it, we quickly realized that it would be impractical and wasteful to do so. While thicker supports inherently offer enhanced structural integrity, within the dynamics of 3D printing, the rigidity is primarily governed by factors such as infill percentage and wall count rather than the total thickness of the supports. To optimize both printing speed and material usage, 3D-printed components are predominantly composed of air on the inside, relying on infill patterns to support the external walls that define the structure. Given that insight, we decreased the thickness of our supports, aligning our design with better 3D printing practices. Guided by a reference from the School of the Art Institute of Chicago, which recommends a 10% infill for visually oriented prints and a 20% infill for structural purposes, we initially adopted a 15% infill during the prototyping phase [33]. This choice prioritized printing speed, considering the likelihood of iterative prototyping. Upon finalizing the design, we strategically increased the infill to 30%, a decision aimed at further enhancing the build quality and structural robustness of the components. Note that the base that holds the L supports also changed from a circular disk to a rectangle as show in Figure 13 since the circular geometry used more material but had no benefit on the structure. This adaptive approach reflects our commitment to optimizing the printing process at each phase of development.

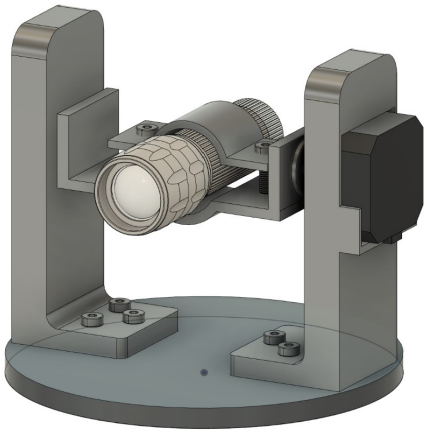

Figure 23: Housing Mount Initial Design

To ensure a robust assembly of all components, we opted for the reliable fastening mechanism provided by metal bolts and nuts. This choice guaranteed a secure and stable connection between the various pieces of our system. Additionally, we took the precaution of incorporating metal flanges to link the motor shafts to the rotating components, further enhancing the structural integrity of our design.

The next consideration that required amelioration was the stability of the flashlight and camera. While the design that has been shown so far is theoretically capable of holding those two components securely, we encountered a lot of issues with slipping since the body of the flashlight has a very smooth and metallic finish. Tightening the screws to their maximum did not help so we had to look for a material that had a high coefficient of friction against plastic and metal. In a study regarding the friction of rubber, this material was described as having a coefficient of friction that is load dependent [34]. Leveraging this property, we introduced a strategically placed piece of rubber in two different places: one where the flashlight sat on top of the platform and another one where the camera mount latches onto the head of the flashlight. By tightening the screws significantly, we essentially created a high-friction interface between all surfaces, effectively mitigating all concerns of slipping.

The design process for the hardware bay was deferred until after the successful testing of the motorized mount. As highlighted earlier, the horizontal movement motor was directly attached be-

neath the entire structure, supporting its entire weight by itself (1:1 ratio). We had initial concerns that the motor would not have enough torque to move effectively, so we planned to explore the possibility of implementing different gear ratios in order to extract more torque out of our motor. Fortunately, the motor performed well during testing, thus eliminating the need for implementing a gear mechanism and allowing us to proceed with the hardware bay design.

To optimize the structure's footprint, we strategically positioned the motor slot at the center and arranged all other boards around it. Given the multitude of variables influencing board placement, including connector clearances, board dimensions, and thermal considerations, several arrangements were explored. Ultimately, a layout was reached that optimized all these factors, as illustrated in Figure 15. This configuration ensures that both the Raspberry Pi and PCB receive fresh airflow from the intake fans, which is crucial for efficient operation of the circuits. While refining the design through multiple prints to achieve precise dimensions, challenges emerged with cable fitment inside the bay, but this issue was promptly resolved by increasing the overall height of the bay. Additionally, considering the limited shaft length of the motor, we also had to elevate the slot in which the motor sat. That raised platform can be observed in Figure 14

Finally, once the initial assembly of the structure was initiated, we had issues with the weight of the motor responsible for the up and down motion. Since it is mounted relatively high up at the top of one of the L supports (see Figure 13), it caused the whole mount structure to sag to one side, destabilizing the whole setup as a result. Fortunately, due to the relatively low cost of stepper motors, we were able to quickly find another NEMA 17 motor that had a lower profile and therefore a much lower weight. The torque was comparable to the original motor, so we had no issues with rotating the flashlight

8.4 Test Plan

Testing of the overall project thoroughly occurred at all phases of development. More specifically, testing was performed once every sub-system was completed as well as during the development of that particular sub-system. That provided reassurance that every component worked

appropriately before they were all integrated together and also made it easier to debug. In the subsequent subsections, we break down all the testing methodology that was employed for each of the sub-systems and lastly how the integrated system's performance was evaluated.

8.4.1 Electrical and Embedded Hardware

As mentioned previously, most of the circuits for this sub-system were not tested until soldered onto the PCB. Due to the diminutive size of some of the components, attempting to interface them all on a breadboard would be a challenging and impractical approach. However, this presented a challenge, as power was a prerequisite to testing other vital components in the system, including the motors, flashlight, and Raspberry Pi. The solution came in a variety of ways. For the motors, we acquired a DC power jack compatible with breadboards and used the 12 V power supply to power motor driver carrier boards that we purchased. In the case of the flashlight, we were able to use the NI Virtual Bench power supply since it is a component that doesn't consume much current. Lastly, the Raspberry Pi worked with a 15 W USB C phone charger. This makeshift setup enabled us to conduct preliminary tests on each device while awaiting the completion of the PCB.

Upon receiving the PCB, our initial step was to conduct a comprehensive examination, checking for continuity in each trace while simultaneously identifying potential shorts. This initial test served as a critical checkpoint, ensuring the absence of any manufacturing defects and validating that the board conformed electrically to our designed schematics. This precautionary measure was instrumental, since trying to rectify issues after soldering would exponentially complicate the troubleshooting process, potentially rendering prior efforts futile.

Following the confirmation of trace integrity, we methodically soldered each chip and its external circuitry individually, conducting functional tests at each stage. Throughout the soldering process, an ample amount of flux was applied to facilitate smooth solder flow and ensure optimal connections. However, as a precautionary measure, we conducted another thorough examination for shorts post-soldering. Given the tightly spaced pins on the chips, this careful scrutiny aimed to preemptively identify any potential issues that could damage the components. With a validated

electrical setup, we progressed to system functionality testing. The motor drivers and buck converters underwent comprehensive yet straightforward assessments, ensuring their suitability for the project's objectives. For the motor drivers, one of the stepper motors was wired, and a simple C program was employed to rotate it both clockwise and counterclockwise at the calculated angular speed of 15.9 RPM, as detailed in the technical section of this report. Additionally, we conducted tests at higher speeds and saw no noticeable issues.

In the case of the buck converters, a network of resistors in parallel was employed to achieve an equivalent resistance of 1.8 Ω, the lowest achievable with available components (see Figure 24 for setup). At 5 V, this setup yielded an output current of $I = \frac{5\,\text{V}}{1.8\,\Omega} = 2.78$ A. This specific configuration had previously exposed temperature issues, as discussed in the technical section. However, following the chip replacement and layout improvements, the system now successfully handled this current draw. Measurement of the current was conducted by incorporating a multimeter in series with the circuit. Once the Raspberry Pi arrived, we were able to power it with the PCB and run our program on it with no noticeable issues.

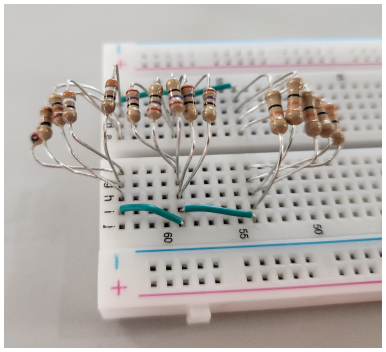

Figure 24: Buck Converter Test Set up

Upon confirming the functionality of the buck converters, we proceeded to test the flashlight. In this test, we connected the light terminals to the PCB, and wired the transistor gate to one of the GPIO pins on the MSP 432. By running a 1 Hz PWM signal, this setup validated the system's capability to control the flashlight, turning it on and off through software.

8.4.2 Software

The process of developing the object detection and tracking software began before the thermal camera had been purchased. During this preliminary phase, thermal video of two deer roaming in a forest was used. The software was assessed on how accurately it detected and tracked the deer, if it ever lost either of them, and how accurately it could recognize them as two separate objects.

Once the camera was available, the software was continuously tested in various environments, including the inside of an apartment, the lab space, and outdoors. Various lighting conditions were also employed including everything from complete darkness to broad daylight. Additionally, the software was tested with a variety of available targets including a single person, multiple people, people only partly in frame, small objects with heat, and objects with extremely high temperatures. Throughout the testing process, the software was constantly evaluated for the qualities listed above, and continuously refined to yield the best possible results.

A crucial step in the testing process was monitoring the algorithm at various intermediate stages of its processing. In cases where the software wasn't behaving as desired, substantial amounts of time were spent analyzing and studying frames right after thresholding or right after dilation in order to narrow down what the issue was.

In order to test the motor software in the early stages, a simple testing program was created: this featured a while loop that moved the motor clockwise 100 steps and then counterclockwise 100 steps. This did a couple of things for the testing process; it proved that the motors were properly set up with the pins and timer, and the correct direction was also verified. This process was done for both the "left-right" and the "up-down" motors individually and eventually simultaneously.

To simulate the communication between the Raspberry Pi and the MSP 432, a simple Python program was created that that takes the user input 'u', 'd', 'l', or 'r' for the commands "up", "down", "left", "right", respectively. The Python program then sends an instruction accordingly which is interpreted by bit masking and shifting by the MSP 432. The expected outcome of this is that the correct motor starts moving in the correct direction. This was a preliminary step to testing the integration of the Raspberry Pi and the MSP. This verified that the MSP was able to

take commands from a Python program and correctly move the motors in the desired direction.

8.4.3 Structural Hardware

Testing this subsystem followed a straightforward procedure, focusing on its ability to accommodate hardware and effectively rotate both vertically and horizontally. After printing the components, the initial step involved assembling them and verifying their dimensions. The assembly of the motorized mount, depicted in Figure 13, was followed by the printing of a test bay for the bottom motor, as shown in Figure 25. The open base was printed in order to give the whole structure stability while the hardware bay was still under the development.

The final component of the housing was the hardware bay and it did not need much testing either since the design was pretty straightforward. All we did to confirm functionality was a test fit of all the components (including wires) and confirmed that it allows for the full functionality of the device.

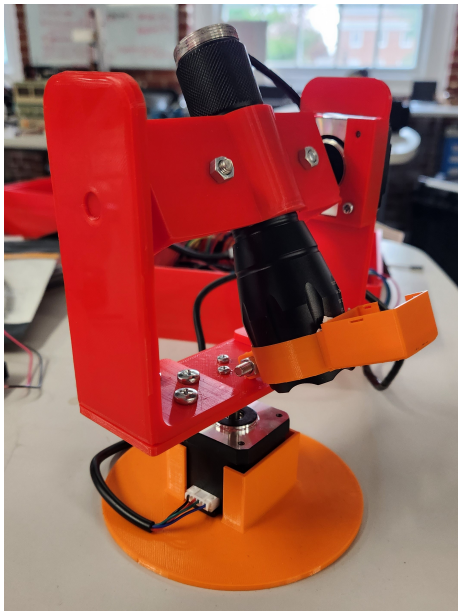

Figure 25: Housing Mount Test Set Up

Once the motors were mounted securely with screws, we used a basic C program to rotate the motors in either direction. During that initial testing, we observed the whole structure sag to one side due to the weight of the vertical motion motor. As described previously, our solution for this

was to simply buy a lower profile motor that weighed a lot less. Once we mounted that on, we started rotating the motors again at 15.8 RPM and observed no noticeable issues. We even stress tested it at 100 RPM and the mount showed resilience even at those speeds. The second test for the mount consisted on mounting both the flashlight and camera to the whole structure to see how the motors would perform under load. This is where we noticed that the flashlight was not fastened well enough and we had to mitigate that by adding rubber as described in the technical section of this report. After this adjustment, the system operated normally at the required speed.

8.4.4 Integrated System Testing

The integration testing phase presented unique challenges as it required seamless interaction among various components, necessitating careful coordination. The initial test focused on evaluating UART communication between the Raspberry Pi and MSP 432. A simple Python program was written to send data from the Pi to the MSP 432 via USB, and the MSP 432, equipped with a corresponding program, sent a response once the data was received. This program was executed in the terminal of the Pi and the received messages were being displayed successfully, paving the way for subsequent motor instructions.

Since we had established solid communication between the Raspberry Pi and MSP 432, the next obvious step was to interface the camera tracking software with the motors. For this, we manually pointed the camera at a subject using our hands and observed if the motors made the movements that would be necessary to center the camera on the subject. Additionally, we made sure to print out every command that the Pi was sending to add another layer of verification. Similarly for the flashlight, we observed if it would turn on once the camera was centered on the subject and turn off when the frame had no subjects in it.

Following successful validation of these functionalities, the camera, motors, and flashlight were mounted on the 3D-printed housing. We let the tracking software run again and instead of moving the camera with our hands, we let the motors do the work and observed that it was successfully tracking a subject. For that test, the speed had to be halved to avoid any excessive pulling of

the wires in case the motors spun out of control. At this testing stage, all components were still powered independently, and to conclude the integration process, the final step involved wiring all components onto the PCB. Each component had undergone individual testing with the PCB, so we encountered no issues when everything was running off the PCB. The tracking software was initiated once more, and successful subject tracking affirmed the seamless integration of all components.

In the conclusive testing phase, we conducted a qualitative assessment of the overall system's performance under diverse conditions. This involved placing a single subject within the camera's field of view and observing the system's responses to various movements by the subject. Key parameters assessed during this phase included response speed, precision in camera pointing, and the tracking speed.

We scrutinized the system's response speed, evaluating how effectively it adapted to changes in the subject's motion. The precision of camera pointing was assessed by confirming whether the light accurately illuminated the subject. Additionally, the tracking speed was examined by having the subject run in front of the camera at various speeds while iteratively increasing the tracking speed until the system's performance started to diminish. Further insights into the outcomes of this testing phase will be detailed in the subsequent final results section.

9 Timeline

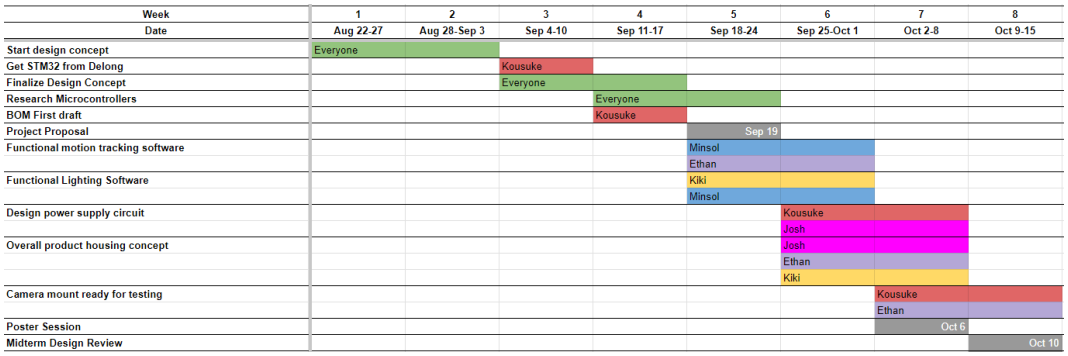

Figure 26: Initial Gantt Chart Part 1

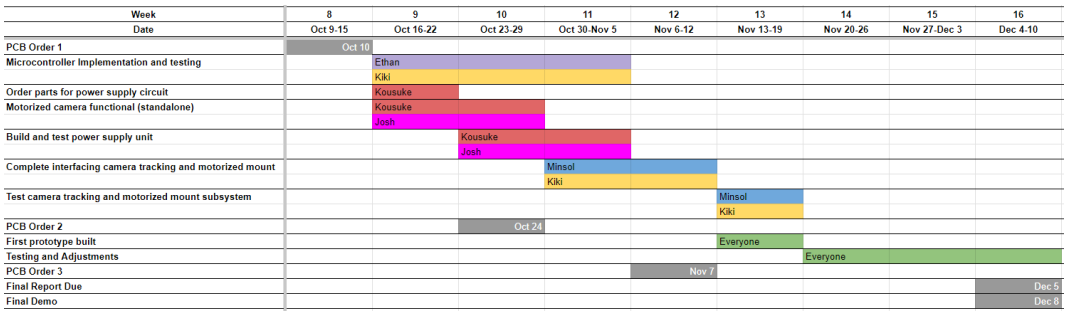

Figure 27: Initial Gantt Chart Part 2

The initial timeline charts can be seen above in Figure 24 and Figure 25, respectively. These charts served as foundational guides to delineate our intended milestones and deadlines. These charts were instrumental in establishing a structured framework, aiding us in determining the projected completion timelines for various project components. The inclusion of deadlines in the Gantt Charts underscores our commitment to project management discipline and adherence to scheduled milestones. By utilizing these charts as template guidelines, we effectively organized our workflow to align with our collective goals, fostering a proactive and methodical approach to project management.

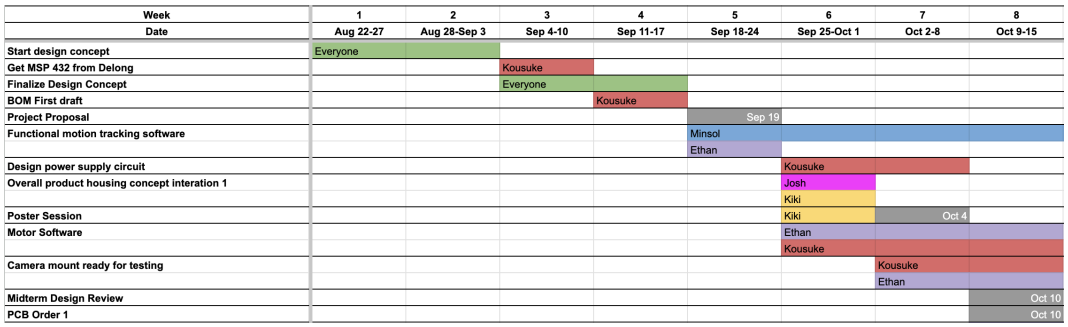

Figure 28: Updated Gantt Chart Part 1

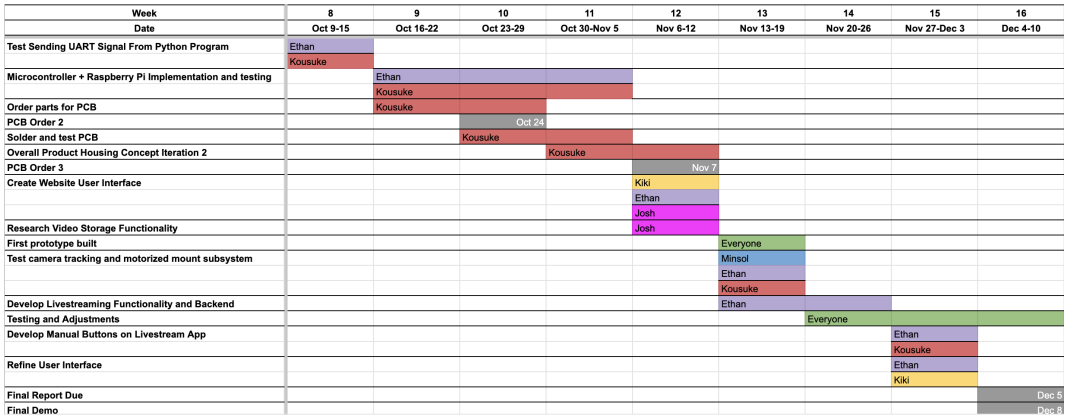

Figure 29: Updated Gantt Chart Part 2

Figure 26 and Figure 27 present the updated Gantt Charts, offering a comprehensive view of the dynamic evolution of our project timeline throughout the semester. These charts vividly reflect the pragmatic adjustments made to roles and responsibilities based on individual capabilities and the imperative to meet designated timelines. The flexibility demonstrated in reallocating tasks underscores our commitment to adapting our approach to optimize efficiency and ensure timely completion. By tracking these changes, we not only acknowledge the dynamic nature of the project but also showcase our team's agility and collaborative spirit in responding to evolving challenges.

10 Costs

The total cost overview for the development of this project is displayed in Table 5. $569.42 of the $500.00 class budget was used, requiring external funding to complete the project. This includes costs not associated with the construction of an individual unit such as testing phases, spare parts, PCB redesigns, and other miscellaneous expenses. Hardware includes the outer casing, thermal camera, and other physical components needed to maintain structural integrity. PCBs relate to the cost of designing and manufacturing the relevant boards. Motors concern the two rotating motors that allow the camera and light source to illuminate different areas of interest. Processing & control pertains to the Raspberry Pi, web server, and other software used to distinguish heat signatures and analyze the camera's feed. Electronics encompass the power source and wiring used

to power our device. Finally, the miscellaneous expenses are any small components that do not relate to any of the other categories. Table 7 and Table 8 in the Appendix show the detailed list of all components purchased throughout the entirety of development.

Category	Price
Hardware	$325.44
PCBs	$53.90
Motors	$27.90
Processing & Control	$86.45
Electronics	$65.82
Misc.	$9.91
Total	$569.42

Table 5: General Cost of Development

If this were to be manufactured as a product, the cost of producing an individual unit without any testing expenses is shown in Table 6 and Table 9 in the Appendix. The cost to manufacture 10,000 units of the current design is estimated to be $433.49 per unit. The primary source of savings originates from the mass purchasing of certain parts and materials such as the thermal cameras and PCBs, which would not require multiple iterations and could be purchased in bulk at a significant discount. The custom 3D printed parts could be mass produced via a plastic mold, significantly reducing the time and energy required to manufacture such intricate details onto the outer casing. All the aforementioned factors would contribute to an increase in production efficiency, leading to a cheaper price per unit.

Category	Price
Hardware	$248.95
PCBs	$18.90
Motors	$27.90
Processing & Control	$86.45
Electronics	$48.98
Misc.	$2.31
Total	$433.49

Table 6: Cost of One Unit

11 Final Results

The final integration testing phase showcased a functional product capable of accurately tracking a subject and directing the flashlight precisely. However, certain limitations currently prevent the device from being suitable for widespread use. The complete assembly of the final device is illustrated in Figure 30.

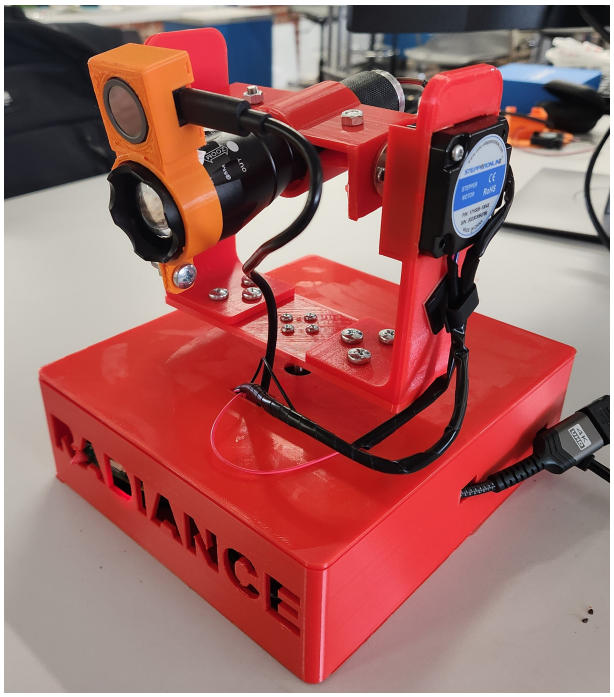

Figure 30: Fully Assembled Device

We begin this section by discussing the system's full capabilities. Upon a subject entering the camera's field of view, the system promptly detects and directs the light towards the subject with notable speed and precision. When the subject is actively tracked and begins to move, a slight yet discernible delay is observed in the motor movements. This results in the light trailing behind the subject momentarily, rather than continuously illuminating it. However, once the subject ceases movement, the system adeptly centers itself with precision once again.

As previously mentioned, various tests involved the subject moving in both horizontal and vertical directions, confirming the system's proper functionality. Different motor speeds were also tested, although the target speed of 15.9 RPM was not attainable due to hardware limitations. The camera's frame rate tops out at 25 frames per second and there is also all the added delay of our image processing, so at greater angular speeds, the camera could not produce a steady image for processing. Nevertheless, the system demonstrated functional results at a capped speed of around 8.5 RPM. The supplementary features integrated into the user interface (UI) functioned seamlessly as well. Users can connect remotely to the system and exercise control over various aspects, including camera movement, light adjustment, and the option to disable the entire system if needed.

Moving on to addressing the system's limitations that could render it unsuitable for widespread use, a major constraint lies in its software. The software is restricted to tracking a single subject as long as there isn't anything else with a greater temperature in the camera's field of view. The absence of robust machine learning models, due to hardware limitations, led to the reliance on thresholding techniques, limiting its ability to discern between living and non-living entities. While effective for one subject, it struggles with multiple subjects. However, under the right conditions, the image processing software will effectively ignore any objects that are below the typical temperature of a human being. Another limitation pertains to speed since someone who was sprinting fast enough would be picked up by our system, but no meaningful tracking would come as a result (due to latency and frame rate limitations). Upon revisiting our proposed grading criteria outlined in Figure 31 of the Appendix, we have successfully fulfilled the functionality criteria by

accomplishing the objective of moving the camera, illuminating the subject in a designated area, and maintaining continuous tracking. However, we did not entirely meet our second accuracy criterion, which aimed to differentiate between living and non-living subjects. Despite this limitation, the system remains effective in tracking a single subject.

In conclusion, although the basic functionality has been achieved, the system's performance outside of a controlled environment is limited, hindering the goal of full automation. Nonetheless, considerable progress has been made, and the entire process has provided valuable insights into the challenges of implementing this technology.

12 Future Work

Ideally, any future work done on this project should address the many limitations that have been discussed throughout this report. Starting with structural hardware improvements, the implementation of a weatherproof housing is a top priority since this is a product that would most likely be placed outside. For that reason, the enclosure should at least be resistant against water in the event of rain. Although this feature was beyond the project's scope due to challenges in achieving water-tight seals with external moving parts, it remains a key consideration for future work. Another structural improvement could involve enabling a 360° rotation, as the current design allows only for 180° in both directions. Due to budgetary constraints, we could not afford the slip ring necessary to implement that capability. Finally, to account for both visible light and infrared wavelengths needing to pass through any external casing, at least one pane of the case could be constructed out of sapphire. This material is extremely prohibitive due to its high cost, but it fulfills all the requirement of a strong, reliable housing by allowing all visible and infrared wavelengths through, being hard enough to resist bludgeoning force, and sealing the electronic components from water.

Regarding electrical and embedded hardware, while no major issues were encountered, careful attention to PCB layout is crucial. That could determine whether or not your circuit will work

properly. Good research and understanding should be acquired before taking on the task of laying out components on a board.

As evidenced by the final performance of our device, the most relevant improvement will come from approaching the image processing in a different manner. Our attempts to implement a machine learning model and run it on the Raspberry Pi were met with failure due to the lack of processing power. The resulting video feed could only run anywhere between 2 and 4 frames per second. However, upon further research, AI accelerating hardware can be purchased and be physically interfaced with edge devices such as a Raspberry Pi. A notable example of this is Google's Coral, which is a Tensor Processing Unit (TPU) that comes in a variety of form factors including a USB module [35]. We did not foresee the need for such hardware early in the project, so we did not have enough budget left to purchase it. So for future iterations of this project, that would definitely be a good place to start and it could unlock the full potential of the device. A machine learning model, facilitated by dedicated AI hardware, could accurately track humans and other living beings while disregarding inanimate objects.

References

[1] "Drv8825 stepper motor controller ic," Apr 2010, Accessed: 2023-09-05. [Online]. Available: https://www.ti.com/lit/ds/symlink/drv8825.pdf

[2] K. Sage and S. Young, "Security applications of computer vision," *IEEE Aerospace and Electronic Systems Magazine*, vol. 14, no. 4, pp. 19–29, 1999, Accessed: 2023-09-15.

[3] SimpliSafe, "Wireless outdoor security camera: Simplisafe home security," Accessed: 2023-09-15. [Online]. Available: https://simplisafe.com/outdoor-security-camera

[4] "Lorex 4k (16 camera capable)," Accessed: 2023-09-15. [Online]. Available: https://www.lorex.com/products/lorex-4k-16-camera-capable-4tb-wired-nvr-system-with-nocturnal-4-smart-ip-dome-cameras-featuring-motorized-varifocal-lens-listen-in-audio-and-30fps?variant=42640888103062

[5] A. Rogalski, "Infrared detectors: an overview," *Infrared Physics & Technology*, vol. 43, no. 3, pp. 187–210, 2002, Accessed: 2023-09-15. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1350449502001408

[6] J. Freer, B. Beggs, H. Fernandez-Canque, F. Chevrier, and A. Goryashko, "Automatic video surveillance with intelligent scene monitoring and intruder detection," *Proceedings of IEEE International Carnahan Conference on Security Technology*, Accessed: 2023-09-15.

[7] J. Jackson, "No e-wasteland for electronic waste disposal: effective legislation to protect communities surrounding landfills," pp. 499+, 2015.

[8] E. R. G. et al., "The life cycle assessment for polylactic acid (pla) to make it a low-carbon material," Accessed: 2023-12-03. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8199738/

[9] "Generic standard on printed board design." Accessed: 2023-12-01. [Online]. Available: https://www.ti.com.cn/cn/lit/ug/slau723a/slau723a.pdf

[10] "Nema ratings for enclosures - nema enclosure ratings chart," Accessed: 2023-09-15. [Online]. Available: https://www.nemaenclosures.com/enclosure-ratings/nema-rated-enclosures.html

[11] M. Barr, "Embedded c coding standard," 2018, Accessed: 2023-09-15. [Online]. Available: https://barrgroup.com/sites/default/files/barr_c_coding_standard_2018.pdf

[12] "Smd packages: Sizes dimensions details," Accessed: 2023-09-15. [Online]. Available: https://www.electronics-notes.com/articles/electronic_components/surface-mount-technology-smd-smt/packages.php

[13] W. Jiang, "Automatic stage lighting tracking system and control method therefor," Apr 2019, U.S Patent 11324095, Jun. 21, 2018.

[14] R. Hicks, "Sensor system augmented with thermal sensor object confirmation," U.S Patent 10445599, Jun. 21, 2018.

[15] P. Amini and J. A. Emmanuel, "Mesh-based home security system," U.S Patent 11576127, Dec. 21, 2018.

[16] L. Zhao, Q. Sun, B. Wang, and X. Wang, "Research on energy cost of human body exercise at different running speed," in *Proceedings of the 11th International Conference on Computer Engineering and Networks*, Q. Liu, X. Liu, B. Chen, Y. Zhang, and J. Peng, Eds. Singapore: Springer Nature Singapore, 2022, pp. 430–436, Accessed: 2023-09-15.

[17] Stan, "Main navigation menu home tutorials resources electronics fundamentals showcase about," Nov 2014, Accessed: 2023-12-03. [Online]. Available: https://42bots.com/tutorials/stepper-motor-wiring-how-to/

[18] "Open source computer vision documentation," Accessed: 2023-09-25. [Online]. Available: https://docs.opencv.org/4.x/

[19] "Raspberry pi 4 model b datasheet," Jun 2019. [Online]. Available: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf

[20] T. L. by Eaton, "Usb charging and power delivery," 2023, accessed on 2023-09-19. [Online]. Available: https://tripplite.eaton.com/products/usb-charging

[21] "Hybrid stepper motor for 3d printer," 2023, accessed on 2023-09-15. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/2263/FIT0278_Web.pdf

[22] "Nema 17 Bipolar stepper motor," (Accessed: 2023-09-18). [Online]. Available: https://www.omc-stepperonline.com/nema-17-bipolar-1-8deg-16ncm-22-6oz-in-1a-3-7v-42x42x20mm-4-wires-17hs08-1004s

[23] "Sunon - magnetic levitation motor fan," Sep 2017. [Online]. Available: http://portal.sunon.com.tw/pls/portal/sunonap.sunon_html_d_pkg.open_file?input_file_name=7264646

[24] Z. Leng, Q. Liu, J. Sun, and J. Liu, "A research of efficiency characteristic for buck converter," in *2010 The 2nd International Conference on Industrial Mechatronics and Automation*, vol. 1, 2010, pp. 232–235, Accessed: 2023-12-04.

[25] "Raspberry pi 15w usb-c power supply," Nov 2021, Accessed: 2023-09-15. [Online]. Available: https://datasheets.raspberrypi.com/power-supply/15w-usb-c-power-supply-product-brief.pdf

[26] "Synchronous buck converter," Sep 2012, Accessed: 2023-10-07. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/408/BD86120EFJ.pdf

[27] "Nx3008nbk," Nov 2022, Accessed: 2023-10-15. [Online]. Available: https://assets.nexperia.com/documents/data-sheet/NX3008NBK.pdf

[28] "Lmr33640 simple switcher® 3.8-v to 36-v, 4-a synchronous step-down converter," Oct 2019, Accessed: 2023-11-08. [Online]. Available: https://www.ti.com/lit/ds/symlink/lmr33640.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwets=1701567354539

[29] B. Suppanz, "Pcb trace width calculator," 2018, Accessed: 2023-10-07. [Online]. Available: https://www.4pcb.com/trace-width-calculator.html

[30] "Technical reference manual," Mar 2015, Accessed: 2023-09-15. [Online]. Available: https://www.ti.com.cn/cn/lit/ug/slau723a/slau723a.pdf

[31] "The maker open lab (mos)," Accessed: 2023-12-03. [Online]. Available: https://mol.as.virginia.edu/about/

[32] "Ultimaker s5," Nov 2023, Accessed: 2023-12-03. [Online]. Available: https://ultimaker.com/3d-printers/s-series/ultimaker-s5/

[33] "Guide for ultimaker 3 / s5," Accessed: 2023-10-27. [Online]. Available: https://sites.saic.edu/aoc/wp-content/uploads/sites/68/2018/05/Ultimakerguide.pdf

[34] A. Schallamach, "The load dependence of rubber friction," *Proceedings of the Physical Society. Section B*, vol. 65, no. 9, p. 657–661, 1952, Accessed: 2023-11-20.

[35] "Usb accelerator," 2020, Accessed: 2023-09-15. [Online]. Available: https://coral.ai/products/accelerator

13 Appendix

Grade	Functionality	Accuracy
A	The system is able to move to and illuminate the entire preset area and can keep pace with all moving objects within acceptable bounds.	The system can near-perfectly distinguish between living and non-living objects and can rotate the motors with speed and precision according to the information
B	The system is able to move to and illuminate the entire preset area, but can mostly keep pace with slowly to moderately moving objects	The system can regularly distinguish between living and non-living objects better than random chance and can rotate the motors according to the information
C	The system is able to move to and illuminate the entire preset area, but can not keep pace with moving objects	The system can occasionally distinguish between living and non-living objects better than random chance and can rotate the motors according to the information
D	The system is able to slightly move to and illuminate, but not the entire preset area	The system can occasionally distinguish between living and non-living objects better than random chance, yet can not rely information to the motors
F	The system is unable to move to or illuminate any of the preset area	The system is unable to distinguish between living and non-living objects

Figure 31: Grading Criteria from Project Proposal

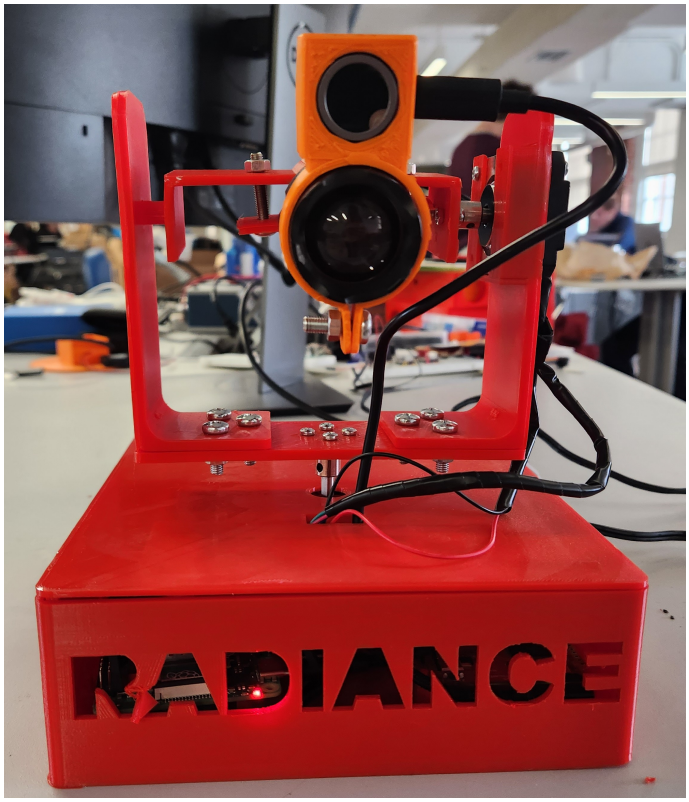

Figure 32: Final Product Front View

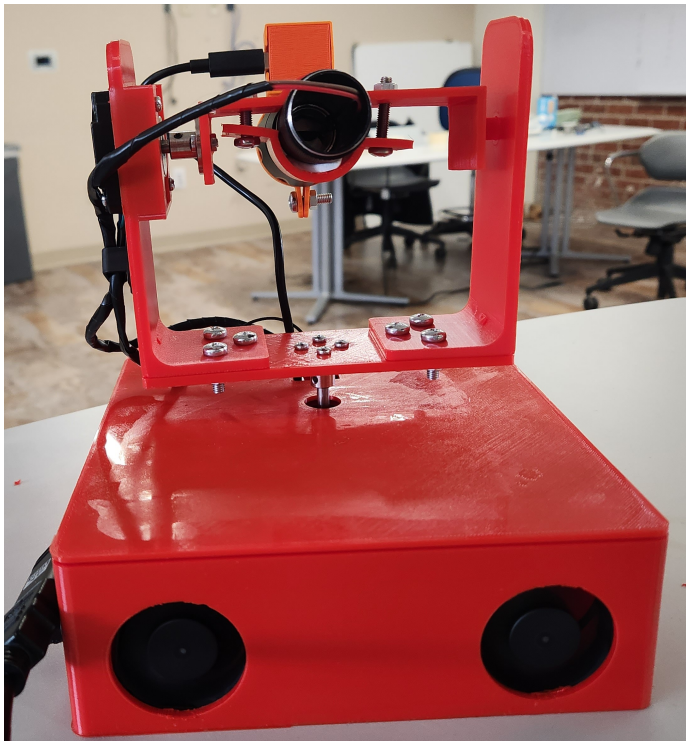

Figure 33: Final Product Rear View

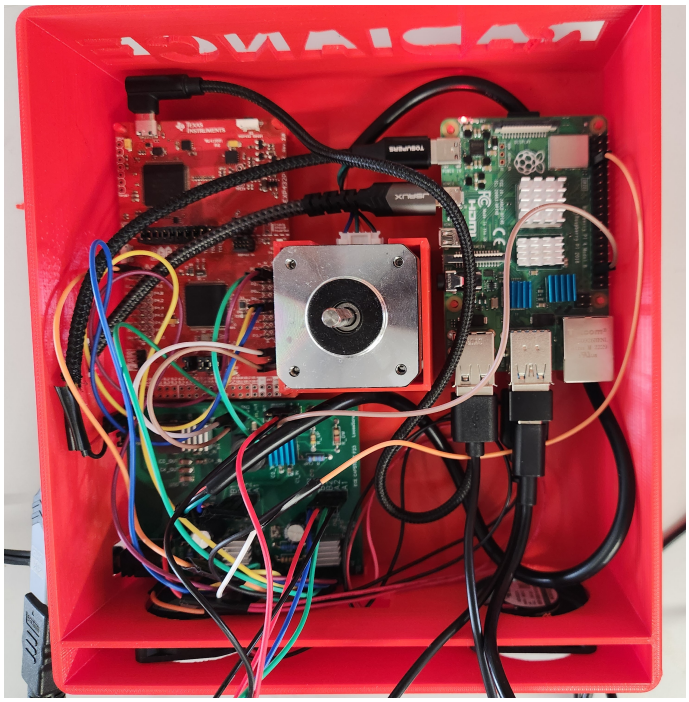

Figure 34: Final Product Hardware Bay Top View

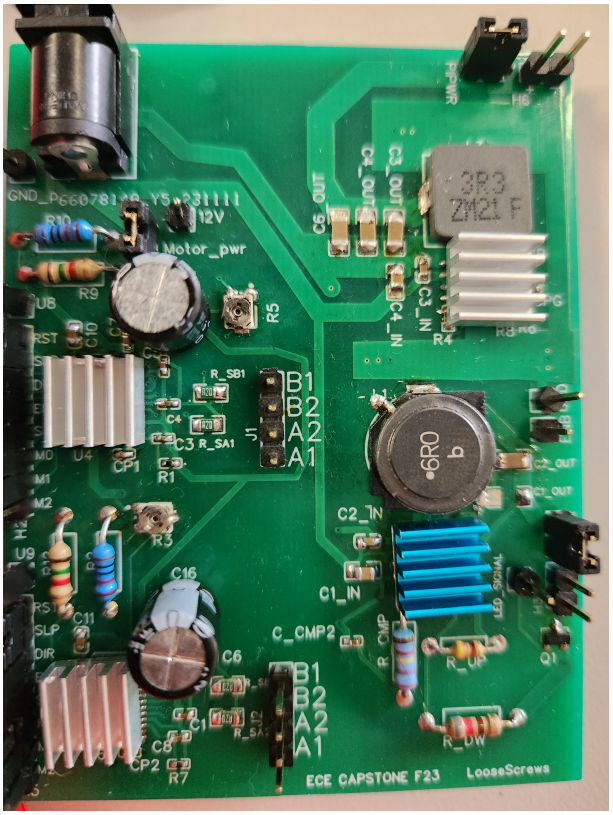

Figure 35: Fully Assembled PCB (Front)

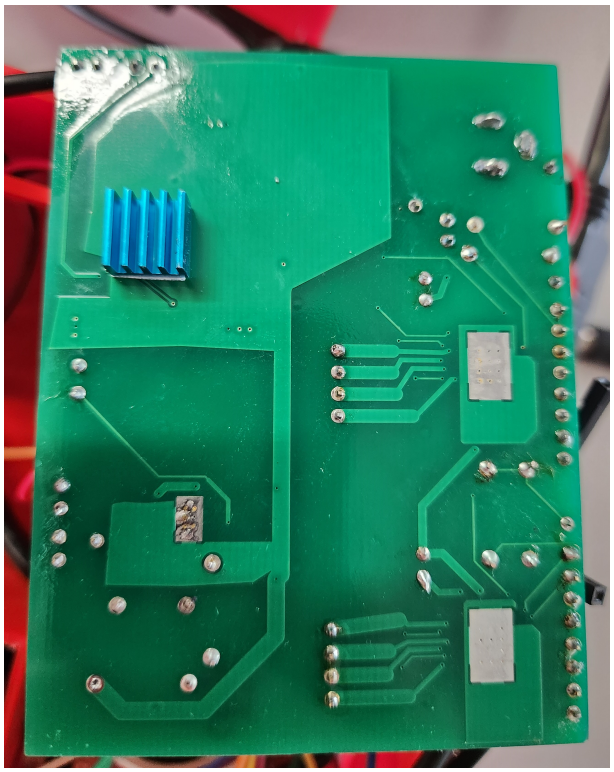

Figure 36: Fully Assembled PCB (Back)

```
#ifndef Motor_H
#define Motor_H

#include "msp.h"

////////Motor for left and right movement////////////

//Step signal. Choosing port 2 pin  5
#define LR_MOTOR_STEP_BIT 5
#define LR_MOTOR_STEP_MASK (0x01 << LR_MOTOR_STEP_BIT)
#define LR_MOTOR_STEP_OUT P2->OUT
#define LR_MOTOR_STEP_DIR P2->DIR
#define SET_LR_MOTOR_STEP_AS_AN_OUTPUT LR_MOTOR_STEP_DIR |= LR_MOTOR_STEP_MASK
#define LR_MOTOR_STEP_HIGH LR_MOTOR_STEP_OUT |= LR_MOTOR_STEP_MASK
#define LR_MOTOR_STEP_LOW LR_MOTOR_STEP_OUT &= ~LR_MOTOR_STEP_MASK
#define TOGGLE_LR_MOTOR_STEP LR_MOTOR_STEP_OUT ^= LR_MOTOR_STEP_MASK

// Dir signal. Choosing port 1 pin 6
#define LR_MOTOR_DR_BIT 6
#define LR_MOTOR_DR_MASK (0x01 << LR_MOTOR_DR_BIT)
#define LR_MOTOR_DR_OUT P1->OUT
#define LR_MOTOR_DR_DIR P1->DIR
#define SET_LR_MOTOR_DR_AS_AN_OUTPUT LR_MOTOR_DR_DIR |= LR_MOTOR_DR_MASK
#define LR_MOTOR_DR_HIGH LR_MOTOR_DR_OUT |= LR_MOTOR_DR_MASK
#define LR_MOTOR_DR_LOW LR_MOTOR_DR_OUT &= ~LR_MOTOR_DR_MASK
#define TOGGLE_LR_MOTOR_DR LR_MOTOR_DR_OUT ^= LR_MOTOR_DR_MASK

// Start and stop commands
#define LR_START TIMER_A0->CTL = (TIMER_A_CTL_SSEL__SMCLK | TIMER_A_CTL_MC__UP | TIMER_A_CTL_ID__8)
#define LR_STOP TIMER_A0->CTL = (TIMER_A_CTL_MC__STOP | TIMER_A_CTL_CLR)
```

Figure 37: Horizontal Movement Motor Setup MSP 432 Software

```
33 ////////Motor for up and down movement////////////
34
35 //Step signal. Choosing port 2 pin  6
36 #define UD_MOTOR_STEP_BIT 6
37 #define UD_MOTOR_STEP_MASK (0x01 << UD_MOTOR_STEP_BIT)
38 #define UD_MOTOR_STEP_OUT P2->OUT
39 #define UD_MOTOR_STEP_DIR P2->DIR
40 #define SET_UD_MOTOR_STEP_AS_AN_OUTPUT UD_MOTOR_STEP_DIR |= UD_MOTOR_STEP_MASK
41 #define UD_MOTOR_STEP_HIGH UD_MOTOR_STEP_OUT |= UD_MOTOR_STEP_MASK
42 #define UD_MOTOR_STEP_LOW UD_MOTOR_STEP_OUT &= ~UD_MOTOR_STEP_MASK
43 #define TOGGLE_UD_MOTOR_STEP UD_MOTOR_STEP_OUT ^= UD_MOTOR_STEP_MASK
44
45 // Dir signal. Choosing port 6 pin 7
46 #define UD_MOTOR_DR_BIT 7
47 #define UD_MOTOR_DR_MASK (0x01 << UD_MOTOR_DR_BIT)
48 #define UD_MOTOR_DR_OUT P6->OUT
49 #define UD_MOTOR_DR_DIR P6->DIR
50 #define SET_UD_MOTOR_DR_AS_AN_OUTPUT UD_MOTOR_DR_DIR |= UD_MOTOR_DR_MASK
51 #define UD_MOTOR_DR_HIGH UD_MOTOR_DR_OUT |= UD_MOTOR_DR_MASK
52 #define UD_MOTOR_DR_LOW UD_MOTOR_DR_OUT &= ~UD_MOTOR_DR_MASK
53 #define TOGGLE_UD_MOTOR_DR UD_MOTOR_DR_OUT ^= UD_MOTOR_DR_MASK
54
55 // Start and stop commands
56 #define UD_START TIMER_A1->CTL = (TIMER_A_CTL_SSEL__SMCLK | TIMER_A_CTL_MC__UP | TIMER_A_CTL_ID__8)
57 #define UD_STOP TIMER_A1->CTL = (TIMER_A_CTL_MC__STOP | TIMER_A_CTL_CLR)
```

Figure 38: Vertical Movement Motor Setup MSP 432 Software

```
16 void InitializeUART0(void) {
17     EUSCI_A0->CTLW0 |= EUSCI_A_CTLW0_SWRST;             // Reset mode to set up
18     EUSCI_A0->MCTLW = 0x0201;               // settings on page919 TRM
19     EUSCI_A0->CTLW0 = 0x0081;          // TRM pg 924
20     EUSCI_A0->BRW = 78;
21     P1->SEL0 |= 0x0C;                  // Use P1.3, P1.2 for UART. pg 7 and 138 in datasheet
22     P1->SEL1 &= ~0x0C;
23     EUSCI_A0->CTLW0 &= ~1;             // Disable reset mode
24 }
25 void SendChar(char c){
26
27     while(!(EUSCI_A0->IFG & 0x02)) { }  // waiting for transfer buffer to empty
28         EUSCI_A0->TXBUF = c;               // load next char
29
30 }
31
32 void SendString(char* str){
33     while (*str != 0)            //sends characters one by one
34     SendChar(*str++);
35 }
36
37 char ReceiveChar(){
38
39     while(!(EUSCI_A0->IFG & 0x01)) { }  //waiting for buffer to receive 1 char
40         rec = EUSCI_A0->RXBUF;              // storing received char
41
42         return rec;
43 }
```

Figure 39: UART Setup MSP 432 Software

Part Name	Manufacturer Part #	Digikey Part #	Mouser Part #	Qty Req'd	Per Unit Price	Cost
Thermal Camera				1	199.99	199.99
NEMA 17 Stepper Motor	FIT0278	1738-1036-ND		2	13.95	27.9
60 W DC PSU	TR9CE5000LCP-N(R6B)	1939-1237-ND		1	18.61	18.61
Outlet Connector for PSU	PWCD-515PC13-10A-01F-BLK	2830-PWCD-515PC13-10A-01F-BLK-ND		1	2.64	2.64
DC Power Jack (SMT)	RASM712BKZ	SC3835-ND		1	2.24	2.24
NMOS	NX3008NBK,215	1727-1220-1-ND		1	0.28	0.28
10 kOhm Potentiometer	TC33X-2-103E	TC33X-103ECT-ND		2	0.28	0.56
200 mOhm Resistor	RL1220S-R20-F	RL12S.20FCT-ND		4	0.29	1.16
4.3 kOhm Resistor	CF18JT4K30	CF18JT4K30CT-ND		2	0.1	0.2
Motor Drivers	DRV8825PWPR	296-29503-1-ND		2	5.65	11.3
22 uF Capacitor	CL31A226MOCLNNC	1276-2728-1-ND		5	0.23	1.15
10 uF Capacitor	GRM21BR61E106KA73K	490-16824-1-ND		2	0.17	0.34
0.1 uF Capacitor	CL10B104KB8NNNC	1276-1000-1-ND		2	0.12	0.24
0.47 uF Capacitor	0603B474K6R3CT	1292-1453-1-ND		2	0.1	0.2
4.7 uF Capacitor	CL05A475MO5NUNC	1276-6836-1-ND		2	0.44	0.88
10 nF Capacitor	CL05B103KB5NNNC	1276-1028-1-ND		10	0.015	0.15
0.1 uF Capacitor	GRM155R71C104KA88J	490-6328-1-ND		10	0.027	0.27
2.7 nF Capacitor	GRM155R71H272KA01J	490-6361-1-ND		2	0.1	0.2
1 MOhm Resistor	RC0402FR-071ML	311-1.00MLRCT-ND		10	0.014	0.14
Inductor	NS12555T6R0NN	587-2726-1-ND		2	1.16	2.32
Rohm Buck Converters	BD86120EFJ-E2	BD86120EFJ-E2CT-ND		2	2.5	5
Male Header Pins	61300811121	732-5321-ND		3	0.4	1.2
Pin Jumpers	QPC02SXGN-RC	S9337-ND		10	0.05	0.5
Thermal Tape	19MM-19MM-10-8810	3M10313-ND		1	1.86	1.86
27.4 kOhm	MRS25000C2742FRP00		594-MRS25000C2742FRP	2	0.29	0.58
Motor Shaft Flanges				1	8.49	8.49
Ribbon Cables				1	19.99	19.99
Raspberry Pi 4 Model B (4 GB)	SC0194(9)	2648-SC0194(9)-ND		1	55	55

Table 7: Detailed Breakdown of Costs of Development (1/2)

1 uF Capacitor	GRM155R61E105KA12D	490-10017-1-ND		1	0.1	0.1
0.22 uF Capacitor	GRM188R71E224KA88D	490-3290-1-ND		1	0.1	0.1
24.9 kOhms Resistors	CR0402-FX-2492GLF	CR0402-FX-2492GLFCT-ND		1	0.1	0.1
Inductor	HCM1103-3R3-R	283-4134-1-ND		1	2.16	2.16
TI Buck Converter	LMR33640DDDAR	296-LMR33640DDDARCT-ND		1	1.49	1.49
100 kOhm	RMCF0603FT100K	RMCF0603FT100KCT-ND		2	0.1	0.2
DC Power Jack (through hole)	PJ-002BH	CP-002BH-ND		1	0.81	0.81
40 mm Fans (12 V)	HA40101V4-1000U-A99	259-1790-ND		2	3.83	7.66
Micro HDMI Adapter				1	7.64	7.64
Motor Driver Carrier Boards				1	14.99	14.99
128 GB MicroSD Card				1	14.99	14.99
Micro USB Cable - right angle				1	7.99	7.99
Heatsinks				1	11.99	11.99
USB C Connector for Camera				1	9.98	9.98
PCB 1				1	35	35
PCB 2				1	18.9	18.9
					Total	497.49
					Available Funds	2.51
Additional Items Over Budget						
Low Profile Stepper Motor	11.99					
Nuts and Bolts	10					
Buck Conv 5 pack	15.78					
Boot USB for PI	12.41					
Breadboard power jack	9					
Motor Driver Carrier Boards	15.26				**Grand Total**	569.42
Total Over Budget	71.93					

Table 8: Detailed Breakdown of Costs of Development (2/2)

Part Name	Manufacturer Part #	Digikey Part #	Mouser Part #	Qty Req'd	Per Unit Price	Cost
Thermal Camera				1	199.99	199.99
NEMA 17 Stepper Motor	FIT0278	1738-1036-ND		2	13.95	27.9
60 W DC PSU	TR9CE5000LCP-N(R6B)	1939-1237-ND		1	18.61	18.61
Outlet Connector for PSU	PWCD-515PC13-10A-01F-BLK	2830-PWCD-515PC13-10A-01F-BLK-ND		1	2.64	2.64
NMOS	NX3008NBK,215	1727-1220-1-ND		1	0.28	0.28
10 kOhm Potentiometer	TC33X-2-103E	TC33X-103ECT-ND		2	0.28	0.56
200 mOhm Resistor	RL1220S-R20-F	RL12S.20FCT-ND		4	0.29	1.16
4.3 kOhm Resistor	CF18JT4K30	CF18JT4K30CT-ND		2	0.1	0.2
Motor Drivers	DRV8825PWPR	296-29503-1-ND		2	5.65	11.3
22 uF Capacitor	CL31A226MOCLNNC	1276-2728-1-ND		5	0.23	1.15
10 uF Capacitor	GRM21BR61E106KA73K	490-16824-1-ND		2	0.17	0.34
0.1 uF Capacitor	CL10B104KB8NNNC	1276-1000-1-ND		2	0.12	0.24
0.47 uF Capacitor	0603B474K6R3CT	1292-1453-1-ND		2	0.1	0.2
4.7 uF Capacitor	CL05A475MO5NUNC	1276-6836-1-ND		2	0.44	0.88
10 nF Capacitor	CL05B103KB5NNNC	1276-1028-1-ND		10	0.015	0.15
0.1 uF Capacitor	GRM155R71C104KA88J	490-6328-1-ND		10	0.027	0.27
2.7 nF Capacitor	GRM155R71H272KA01J	490-6361-1-ND		2	0.1	0.2
1 MOhm Resistor	RC0402FR-071ML	311-1.00MLRCT-ND		10	0.014	0.14
Inductor	NS12555T6R0NN	587-2726-1-ND		2	1.16	2.32
Rohm Buck Converters	BD86120EFJ-E2	BD86120EFJ-E2CT-ND		2	2.5	5
Male Header Pins	61300811121	732-5321-ND		3	0.4	1.2
Pin Jumpers	QPC02SXGN-RC	S9337-ND		10	0.05	0.5
Thermal Tape	19MM-19MM-10-8810	3M10313-ND		1	1.86	1.86
27.4 kOhm	MRS25000C2742FRP00		594-MRS25000C2742FRP	2	0.29	0.58
Motor Shaft Flanges				1	8.49	8.49
Ribbon Cables				1	19.99	19.99
Raspberry Pi 4 Model B (4 GB)	SC0194(9)	2648-SC0194(9)-ND		1	55	55
DC Power Jack (through hole)	PJ-002BH	CP-002BH-ND		1	0.81	0.81
40 mm Fans (12 V)	HA40101V4-1000U-A99	259-1790-ND		2	3.83	7.66
128 GB MicroSD Card				1	14.99	14.99
Micro USB Cable - right angle				1	7.99	7.99
Heatsinks				1	11.99	11.99
PCB 2				1	18.9	18.9
Nuts and Bolts				1	10	10
					Grand Total:	**433.49**

Table 9: Detailed Breakdown of Costs of One Unit

Introduction of STS Topic

Throughout recent years, the advancement in the realms of mechanical automation and artificial intelligence has accelerated our worldwide reliance on non-human production methods. Several necessities of everyday life now stem from companies with sophisticated automation pipelines that produce everything from doors and refrigerators to diapers and mobile phones (JR Automation, 2023). While this shift from human to mechanical production may bring some benefit, the overwhelming detriment to consumers' employment and earning prospects must be counterbalanced with an influx of new employment opportunities in the relevant fields before mechanical automation techniques integrate too deeply into our production methods.

Unlike innovations of the past, a wave of New Automation brings about a vast array of tasks and jobs that become obsolete in the face of an unprecedented mechanical competitor. New Automation refers to an expansion to the capabilities of automated mechanical systems and Artificial Intelligence (AI) that threatens to outperform its human competition. In previous years, only those with the most simple, uneducated positions were concerned about automation replacing their positions, but New Automation threatens even professions with very high levels of education in healthcare, law, engineering, and more (Zia Qureshi, 2022). As automation becomes capable of filling more areas of employment, working age

citizens from all sectors now face harsher requirements for employability, culminating in an overall increase in necessary skills and decrease of employable personnel (Bughin, 2018). Despite the depletion of employment opportunities for working age people, there is no denying the overwhelming economic incentives mechanical automation provides for corporations that employ automation. Among numerous other advantages, the primary economic incentive of automation resides in a reduction of human-related costs such as time-wasting meetings, arduous, repetitive tasks, and worker's compensation, which only exacerbates the destruction of human employability (Tekla, 2023).

STS Framework

In this analysis, the Diffusion of Innovations (DOI) model highlights the importance of considering human detriment in the face of inevitable mechanical automation. Popularized in 1962 by sociologist Everett Rogers, DOI argues that diffusion represents the processes by which an innovation is communicated over time among the participants in a social system (Rogers, 1982). Rogers asserts there exist five main elements that impact the spread of new ideas: the invention itself, types of adopters, communication channels in which to spread, time length, and relevant social systems surrounding the innovation.

Under DOI, every new invention is evaluated by determining how each category will influence the way people adapt to the invention's impacts. In order to sustain itself, each innovation must be widely adopted such that it reaches a critical mass and maintains its place in society. First, the innovation itself pertains to a society's current understanding of any relevant fields of research. In the case of automation, the recent craze of Artificial Intelligence such as ChatGPT has revolutionized the information gathering process for many people, highlighting the vast audience in which automation influences everyday life. Next, adopters represent not only the groups and individuals that may take interest in an innovation but also the speed at which various groups adopt the invention. For automation, companies wishing to reduce costs and unions fighting for workers' rights may adapt to mechanical automation at different rates and be classified as innovators, early adopters, early majority, late majority, or laggards. In addition, communication channels allow information to flow throughout a society with varying groups such as older and younger generations each coveting different communication methods. As such, the speed of communication and the relevant audiences those channels attract all impact how certain innovations are spread and perceived by members of society. In the case of automation, the attraction of easier work and less expenses draw people from all walks of life, which explains the rampant expansion of automation and AI in the last few years (Marr, 2023).

Furthermore, the passage of time is integral for the adoption of innovation since society very rarely accepts new inventions in a timely manner. For instance, Walmart, one of the world's largest supermarket chains, took eight years before going public in spite of its efficient payroll and supply chain system (Tikkanen, 2023). As such, the widespread adoption of automation and AI in just the last year further emphasizes the speed and scale at which automation affects society and relevant job markets (Marr, 2023). Finally, the social system surrounding an invention combines the external and internal influences such as the mass media and social connections respectively. Due to the explosion of tech-related companies such as Apple and Amazon, software engineering and other technological degrees have seen a massive spike in popularity, a harbinger of automation and its impacts on society's job market (CollegeChoice.net, 2023).

Plan for Thesis

To further my thesis on automation and how to counterbalance its effects, my thesis group and I will design and construct a fully automated spotlight security system. The system will allow completely automated detection and illumination of any living creature within a specified area, which eliminates the need for human guards while establishing employment opportunities for electricians and automation engineers. Data for this experiment will be collected by testing human

subjects against the spotlight system and determining whether the automated design replaces a human subject within reasonable parameters. Should the automated system prove reliable enough to take over the role, the destruction of a guard position and the creation of an electrical technician position will be complete. The purpose of this experiment is to demonstrate the concept of how automation and human employment can and must work together to improve upon the current state of technology while also maintaining the level of employability around the automated innovations.

References

Bughin, J., Hazan, E., Lund, S., Dahlström, P., Wiesinger, A., & Subramaniam, A. (2018, May 23). *Skill shift: Automation and the future of the workforce*. McKinsey & Company. https://www.mckinsey.com/featured-insights/future-of-work/skill-shift-automation-and-the-future-of-the-workforce

CollegeChoice.net. (2023, January 13). *What is a technology degree?*. CollegeChoice. https://www.collegechoice.net/technology/#:~:text=Technology%20degrees%20have%20grown%20in,technology%20occupations%20from%202019%2D2029.

JR Automation. (2023, October 8). *Automation systems for Consumer Product Goods (CPG)*. JRAutomation. https://www.jrautomation.com/industries/consumer-products

Marr, B. (2023, October 5). *A short history of chatgpt: How we got to where we are Today*. Forbes. https://www.forbes.com/sites/bernardmarr/2023/05/19/a-short-history-of-chatgpt-how-we-got-to-where-we-are-today/?sh=20bf8933674f

Rogers, E. M. (1982). *Diffusion of innovation*. The Free Press.

Tekla. (2023, June 19). *Nine ways automation helps you drive down costs while increasing quality*. https://www.tekla.com/resources/articles/nine-ways-automation-helps-you-drive-down-costs-while-increasing-quality

Tikkanen, A. (Ed.). (2023, October 8). *Walmart*. Encyclopædia Britannica. https://www.britannica.com/topic/Walmart

Zia Qureshi, C. W., Qureshi, Z., Eswar Prasad, C. S., Fugle, J., & Emily Gustafsson-Wright, E. P. (2022, March 9). *Understanding the impact of automation on workers, jobs, and wages*. Brookings. https://www.brookings.edu/articles/understanding-the-impact-of-automation-on-workers-jobs-and-wages/