Tracking Recyclables Automated Sorting Hand (T.R.A.S.H.)

Group 7 Authors:

Dylan Bowers	Sayed Almousawi	Jayden Lumas	Tristan Koopman
Photonics Engineering	*Electrical Engineering*	*Computer Engineering*	*Mechanical Engineering*

Advisors:

Dr. Lei Wei	Dr. Chung Yong Chan	Dr. Arthur Weeks	Dr. Aravinda Kar

Committee Members:

Dr. Behal	ECE	*Professor*
Dr. Delfyett	CREOL	*Professor*
Dr. Mark Maddox	ECE	*Professor*

UNIVERSITY OF CENTRAL FLORIDA

Table of Contents

1.0 Executive Summary 1

2.0 Project Description 2

2.1 Project Motivation and Background 2

2.2 Project Goals 2

2.2.1 Basic Goals 2

2.2.2 Advanced Goals 3

2.2.3 Stretch Goals 4

2.3 Project Objectives 4

2.4 Specifications and Engineering Requirements 5

2.5 Function of Project 6

2.6 Illustration 6

2.7 Hardware Diagram 7

2.8 Software Diagram 8

2.9 House of Quality 9

2.10 Existing Products and Past Projects 10

2.10.1 Annin Robotics AR4 6DOF Robot Arm 10

2.10.2 AMP Robotics: AI-Powered Robotic Sorters 11

2.10.3 ZenRobotics: Heavy Duty Waste Sorting 11

2.10.4 MIT CSAIL: RecyBot Research Project 11

2.10.5 Max-AI: Autonomous Quality Control 12

2.10.6 Fraunhofer Institute: Spectroscopy in Recycling 12

2.10.7 A.P.A.L.E.S. Automatic Material Identification 12

2.10.8 Key Takeaways for T.R.A.S.H. 13

2.11 Marketing Requirements 13

2.11.1 Pricing and Affordability 13

2.11.2 Usability and Accessibility 14

2.11.3 Target Audience 14

2.11.4 Summary of Marketing Requirements 14

3.0 Research 15
3.1 Torque Estimations 15
3.1.1 Procedure 15
3.1.2 Calculations 17
3.2 Material Selection 21
3.2.1 Aluminum 21
3.2.2 Carbon Fiber 21
3.2.3 Steel 22
3.2.4 PETG 22
3.3 Computer Vision (CV) 22
3.3.1 Comparison with Alternative Detection Methods 23
3.3.2 Advantages of Computer Vision 25
3.3.3 Object Detection Technology Selection 26
3.4 Microcontroller 27
3.4.1 ATmega328P 28
3.4.2 ESP32-S3-WROOM-1-N8 28
3.4.3 STM32F401 29
MCU Selection 30
3.5 Single Board Computer (SBC) 30
3.5.1 Raspberry Pi 5 31
3.5.2 Nvidia Jetson Nano 32
3.5.3 Orange Pi 33
3.5.4 SBC Selection 34
3.6 Power Components 35
3.6.1 Power Supply 35
3.6.2 Voltage Regulator 35
3.6.3 DC-to-DC Buck Converter 36
3.7 Motors 37
3.7.1 Servo Motors 37
3.7.2 Stepper 37
3.7.3 Brushless DC 38

3.7.4 Motor Selection....38
3.8 Motor Encoders....40
3.8.1 Optical Encoders....40
3.8.2 Magnetic Encoders....40
3.8.3 Potentiometers....40
3.8.4 Encoder Selection....41
3.9 Motor Drivers....41
3.9.1 Stepper Driver Selection....41
3.9.2 Servo Driver Selection....42
3.10 Speed Reducer....42
3.10.1 Strain Wave Gear....43
3.10.2 Cycloidal Drive....44
3.10.3 Capstan Drive....44
3.10.4 Planetary Gears....45
3.11 Camera....45
3.11.1 Raspberry Pi Camera Module V2....47
3.11.2 Raspberry Pi Camera Module V3/ArduCam 12MP....48
3.11.3 Raspberry Pi Camera High Quality....49
3.11.4 ArduCam 16MP....50
3.11.5 ArduCam 64MP....51
3.11.6 Camera Selection....52
3.12 ROS & ROS2....53
3.12.1 Robot Operating System 1 (ROS)....54
3.12.2 Robot Operating System 2 (ROS2)....55
3.12.3 ROS Selection....55
3.13 Spectroscopy Methods....56
3.13.1 Raman Spectroscopy....57
3.13.2 Reflection Based Spectroscopy....58
3.13.3 Laser Inducted Breakdown Spectroscopy....59
3.13.4 Transmission Based Spectroscopy....59
3.14 Linear CCD Sensor....61

3.14.1 Performance of Linear CCD 62
3.15 Illumination Sources 62
3.15.1 Incandescent Source 64
3.15.2 Light Emitting Diode (LED) Source 66
3.16 Resolving Position with Optics System 70
3.16.1 Triangulation 71
3.16.2 Deblurring method 73
3.17 Dispersion Techniques: Diffraction Grating vs Prisms 75
3.17.1 Principle of Operation 75
3.18 Spectral Characteristics of Recycled Materials 78
3.18.1 Distinguishable Materials 78
3.18.2 The Usefulness of The Spectrometer 79
4.0 Standards and Design Constraints 80
4.1 Relevant Standards 80
4.1.1 Universal Asynchronous Receiver-Transmitter (UART) 80
4.1.2 Inter-Integrated Circuits (I2C) 81
4.1.3 Serial Peripheral Interface (SPI) 82
4.1.4 Universal Serial Bus (USB) 82
4.1.5 PCB Design Standards 82
4.1.6 Robot Operating System (ROS) Compliance 83
4.1.7 ISO 10218 83
4.2 Design Constraints 83
4.2.1 Economic Constraints 84
4.2.2 Safety Constraints 84
4.2.3 Environmental Constraints 85
4.2.4 Manufacturability Constraints 85
4.2.5 Time Constraints 85
4.2.6 Optical Safety Constraints 86
5.0 Comparison of ChatGPT and other Similar Platforms 87
5.1 ChatGPT – Open AI 87
5.2 Gemini – Google DeepMind 88

5.3 DeepSeek 89
5.4 Microsoft Co-Pilot 89
5.5 Broader Impact of AI Tool Use in Senior Design 90
5.5.1 AI Example 1 90
5.5.2 AI Example 2 91
5.5.3 AI Example 3 91
6.0 Hardware Design 92
6.1 First Subsystem: Power Management 92
6.1.1 12V to 5V DC/DC Buck Converter Schematic 93
6.1.2 5V to 3.3V LDO Regulator Schematic 94
6.2 Second Subsystem: Spectrometer TCD1304DG Sensor 95
6.3 Third Subsystem: ESP32-S3-WROOM and Motor Control 96
6.3.1 MCU Schematic 97
6.3.2 Stepper Motor Connections 97
6.3.3 Servo Motor Driver Schematic 98
6.4 Fourth Subsystem: Raspberry Pi and Other Peripherals 100
6.5 Hardware Design Summary 101
6.6 Optical Design 103
6.6.1 Design Overview 103
6.6.2 Zemax Design 104
7.0 Software Design 106
7.1 Software Logic Overview 107
7.2 Libraries and Tools 109
7.3 Subsystems 110
7.3.1 Computer Vision 110
7.3.2 Inverse Kinematics and Simulations 112
7.3.3 Spectrometer 113
7.3.4 Motor Control 114
7.3.5 User Interface 115
7.3.6 Communications 117
7.4 Summary 118

8.0 Structure Design 119
8.1 Linkages 119
8.1.1 Linkage Cross-Section 120
8.1.2 Internal Loadings 121
8.2 Joints 123
8.2.1 Linkage Coupling 123
8.2.2 Speed Reducer (with transmission) 124
8.3 Differential Wrist 125
8.3.1 Motion 125
9.0 System Fabrication/Prototype Construction 127
9.1 PCB Design and Layout 127
9.1.1 Main PCB 127
9.1.2 Spectrometer PCB 130
9.2 Electronics Enclosure 131
10.0 System Testing and Evaluation 132
10.1 Hardware Testing 133
10.1.1 Servo Motor Testing 133
10.1.2 Stepper Motor Testing 133
10.1.3 CCD Testing 134
10.1.4 Camera Testing 135
10.2 Software Testing 136
10.3 Spectrometer Feasibility Test 138
11.0 Administrative Content 140
11.1 Budget and Financing 140
11.2 Bill of Materials 141
11.3 Work Distribution 142
11.4 Milestones 145
12.0 Conclusion 145
13.0 Appendix B: List of Tables 149
14.0 Appendix C: List of Figures 151
15.0 Appendix D: List of Equations 153

16.0 Appendix E: AI Prompts 154
17.0 Appendix F: Design Competence Evaluation 157

1.0 Executive Summary

A common fear among United States of America citizens is that the rise in robotics technology spells the end of the common laborer. The speculation among most is that with industrial robots fulfilling menial tasks, many hard-working citizens will be left without jobs. However, the more significant threat to the United States is not a labor shortage, but rather a laborer shortage. Since 1963, the world's births per woman have steadily declined from 5.32 to 2.27 in 2022. In the United States specifically, the number of births per woman has gone from 3.32 to 1.66 . This is concerning because to sustain a stable population, the total fertility rate needs to be 2.1 births per woman [2].

With severe laborer shortages and an aging population expected to hit the United States, automation is needed to avoid an economic disaster. One area that stands to benefit greatly from automation is the waste management industry. Historically, plastic waste needed to be sorted by hand from a conveyor line into designated recycling bins, requiring significant manual labor, which is becoming difficult to sustain. With the advancements in industrial robotics, the industry has seen some integration of automated sorting systems. However, these systems are generally large and hazardous machinery, moving in areas that are not safe for human workers, and can be quite expensive. The TRASH project aims to provide a safe, affordable, and efficient automated solution for identifying, acquiring, and sorting recyclable waste products from other than recyclable waste.

The TRASH project focuses on the development of a robotic arm capable of sorting recyclable waste. The 6-degree-of-freedom (6DOF) robotic arm will be integrated with a system that utilizes computer vision and optical sensors to identify and acquire recyclable materials from waste. This innovation will provide a safer alternative to traditional sorting methods, reducing the need for manual labor and increasing the efficiency of waste processing at an affordable price, available to local waste management companies.

The integration of computer vision and spectroscopy systems will contribute to making our robot fully automated by adding the ability to detect the material and obtain its location without any human interaction needed. This will be in addition to a touch display user interface used to force any commands wanted or check the status of the machine.

2.0 Project Description

In this section, the motivation and background of the project are discussed.

2.1 Project Motivation and Background

The primary motivation behind the TRASH project is the growing laborer shortage and the efficiency of waste management. As the American population steadily decreases with a fertility rate that is unable to sustain its population size, robotics is needed more than ever. With the implementation of robotics, human resources can be allocated more efficiently as fewer workers are needed to take care of a larger population unable to work. While there are many advances in the realm of home care robotics, there are a slew of safety concerns, and more advancements are needed before implementation can even be considered a widespread practice. However, the industrial environment requires little to no human interaction and much simpler environments, with many industrial robotics systems using fixed, programmed systems.

Additionally, as America has become sustainment-oriented, the waste industry requires more workers in order to maintain the green initiative. Therefore, there exists an increased number of workers whose only purpose is to pick up and sort recyclables from garbage. These simple tasks can and have begun to be automated in order to free up human resources for more complex tasks that require human engagement.

2.2 Project Goals

TRASH aims to address these challenges by developing a 6DOF pick and place robot arm capable of identifying, acquiring, and sorting recyclables from non-recyclable waste. TRASH will be as lightweight as possible in order to decrease the cost of materials and actuators. It will utilize a combination of computer vision and optical systems for onboard material identification, allowing ease of integration for the robot into an already-existent system and item positioning.

2.2.1 Basic Goals

For our robotic arm, we have a set of basic goals that our project will achieve. As a group, we have determined these goals to be fundamental capabilities required for TRASH. At its most basic level, TRASH will be able to detect the positional data of a plastic bottle among trash using optical and computer vision techniques. After identifying the position and material data, the robot arm will be able to grab and relocate the recyclable item to its respective bin.

Basic Goals		
Identification		
Identify 12oz Green Glass Bottle	Yes	Y/N
Detect Item Position Accuracy	80	%
Detect 5 Items Presented to Robot	Yes	Y/N
Material Identification Accuracy	80	%
Identification Time	30	sec
Motion		
Minimum Payload	100	grams
End Effector Target Positioning Tolerance	5	mm
Motion Time	1	min
Voltage Input	120	VAC

Table 1: Basic Goals

2.2.2 Advanced Goals

Our Advanced Goals reflect the ideal outcome for TRASH, demonstrating enhanced functionality and performance. For our advanced goals, TRASH has better performance criteria and a larger library of identifiable materials, not limited to just plastic bottles, which is achieved by increasing the spectrometer spectrum resolution. Additionally, TRASH will incorporate an interface to give users a real-time camera feed.

Advanced Goals		
Identification		
Identify Cardboard and Aluminum	Yes	Y/N
Detect Plastic Bottle Position	90	%
Detect 6+ Items Presented to Robot	Yes	Y/N
Material Identification Accuracy	90	%
Identification Time	20	sec
Motion		
Minimum Payload	500	grams
End Effector Target Positioning Tolerance	3	mm
Motion Time	45	sec
Max Power Consumption	150	W
Added Functionality		

UI with Real-Time Camera Feed	Yes	Y/N

Table 2: Advanced Goals

2.2.3 Stretch Goals

For TRASH to exceed all expectations, it will achieve its stretch goals. The stretch goals are best-case scenario circumstances that we will try to achieve. For our stretch goals, we aim to improve upon some criteria related to time and accuracy. We also look to integrate systems that improve the efficiency and overall package of the robotic arm. Include the optical fiber integrated into the end effector, allowing for quick material identification without the need to move an item to the spectrometer, only to return nonrecyclables back to where they were picked.

Stretch Goals		
Identification		
Identify 4+ Recyclable Items	Yes	Y/N
Detect Item Position Accuracy	80	%
Inventory Items into Database	Yes	Y/N
Material Identification Accuracy	80	%
Identification Time	10	sec
Motion		
Minimum Payload	1	kg
Motion Time	30	sec
PCB Layout Footprint	432	in^3
Added Functionality		
UI Displays Number of Recyclables	Yes	Y/N
Conveyor Belt System	Yes	Y/N
Integrated Optical Fiber	Yes	Y/N

Table 3: Stretch Goals

2.3 Project Objectives

The objective of the TRASH project is to design and implement a cost-effective, compact, and self-contained robotic sorting system that efficiently identifies and separates recyclable materials from mixed waste using computer vision, optics, and Artificial Intelligence. The system aims to improve waste sorting efficiency for small-scale recycling operations by providing high detection accuracy, precise pick-and-place capabilities, and an automated

classification process while ensuring scalability, energy efficiency, and ease of use. The main objectives of the TRASH project include

1. **Development of a Robotic Arm:** Design and build a lightweight, cost-effective robotic arm that can accurately perform waste sorting tasks with high precision.
2. **Integration of Computer Vision and Optical Systems:** Use computer vision and optical sensors to allow the robot to identify recyclable materials based on their light reflectance characteristics and machine learning, ensuring accurate sorting.
3. **Increased Efficiency:** Improve the efficiency of waste management by automating a process that is typically labor-intensive, freeing human workers for more complex tasks.
4. **Cost-Effective Solution:** Minimize costs by using lightweight materials for the robot's construction and choosing efficient actuators and components.
5. **Seamless Integration:** Ensure the TRASH robot can be easily integrated into current waste management workflows and systems without requiring major infrastructure changes.

2.4 Specifications and Engineering Requirements

As a part of our design requirements, we have presented below in Table 4 and Table 5 our deliverable specifications.

Specification	Target Value	Units
Object Detection Time	20	sec
Total Pickup Time	30	sec
Pickup Accuracy	± 5	mm
Min. Payload	100	g
Arm Degrees of Freedom	6	DOF
Power Consumption	≤ 350	W
Power Supply	120	VAC
Computer Vision	1	camera
Material Identification	At least 2	Materials

Table 4: Engineering Requirements

Specification	Target Value	Units
Slit Width	$8 \leq x \leq 40$	µm
Spectral Resolution	< 20	nm
Lens Focal Distance	100	mm

Table 5: Optical Specifications

2.5 Function of Project

TRASH will be able to orientate its position on the initial startup. From there, TRASH will locate all available items. It will then identify what the material of the items seen are and classify the items as recyclable or not recyclable. If an item is recyclable, the item will be placed in a designated bin. All other items will be left for waste disposal. TRASH will use 6 degrees of freedom in order to allow full manipulation of all items regardless of orientation.

2.6 Illustration

Figure 1 represents a simple 3D illustration of the TRASH concept design we would like to have by the end of this project. The design is not finalized or accurate at this point. The robot arm used in this illustration does not represent the actual design of the arm. This was just a simple free robot arm design downloaded from the internet to support our illustration. The spectrometer system shown is the basic idea of what it would be like. A spectrometer box is mounted next to the arm while a bunch of LED lights are shined on its lens from the opposing side of the base. The robot arm would then place an item in between the LED lights and the box to determine the item's material.

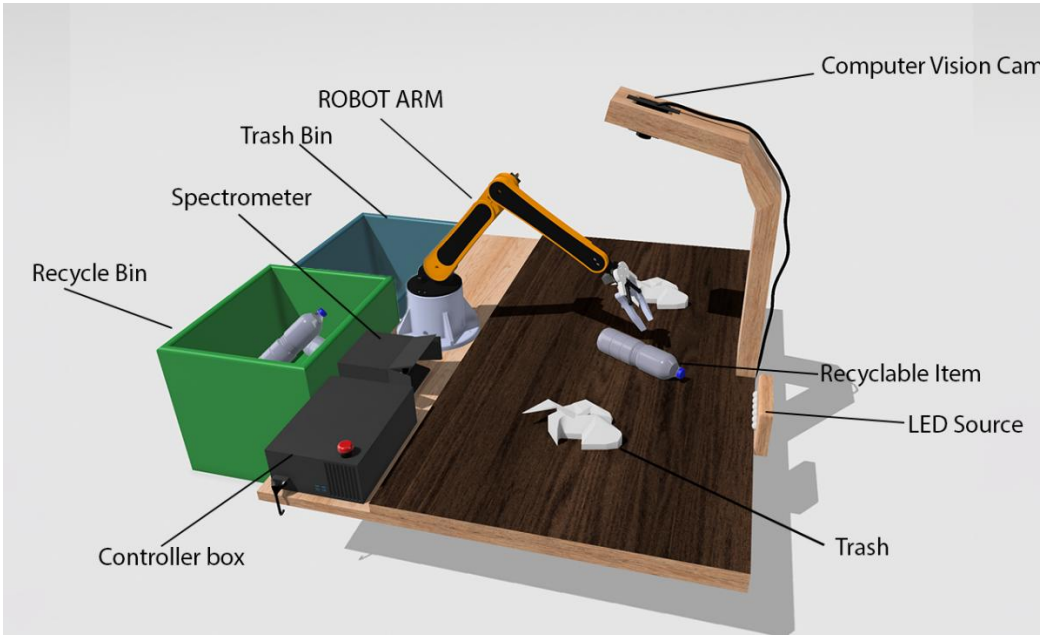


Figure 1: TRASH Concept Illustration

Figure 2 serves as a simple reduction of the final design. The samples is illuminated by a array of LEDs which can be a combination of WLEDs and IR LEDs to ensure to illuminate the spectral range from 400nm to 1100nm. A reflection-based diffraction grating will be

used with a blazed wavelength of 1000nm to compensate for the drop of responsivity of the silica-based detector. The samples that are going to be tested are discussed in length in the recycled material portion of the paper; however, sample can include, glass bottles, aluminum cans, and plastic materials.

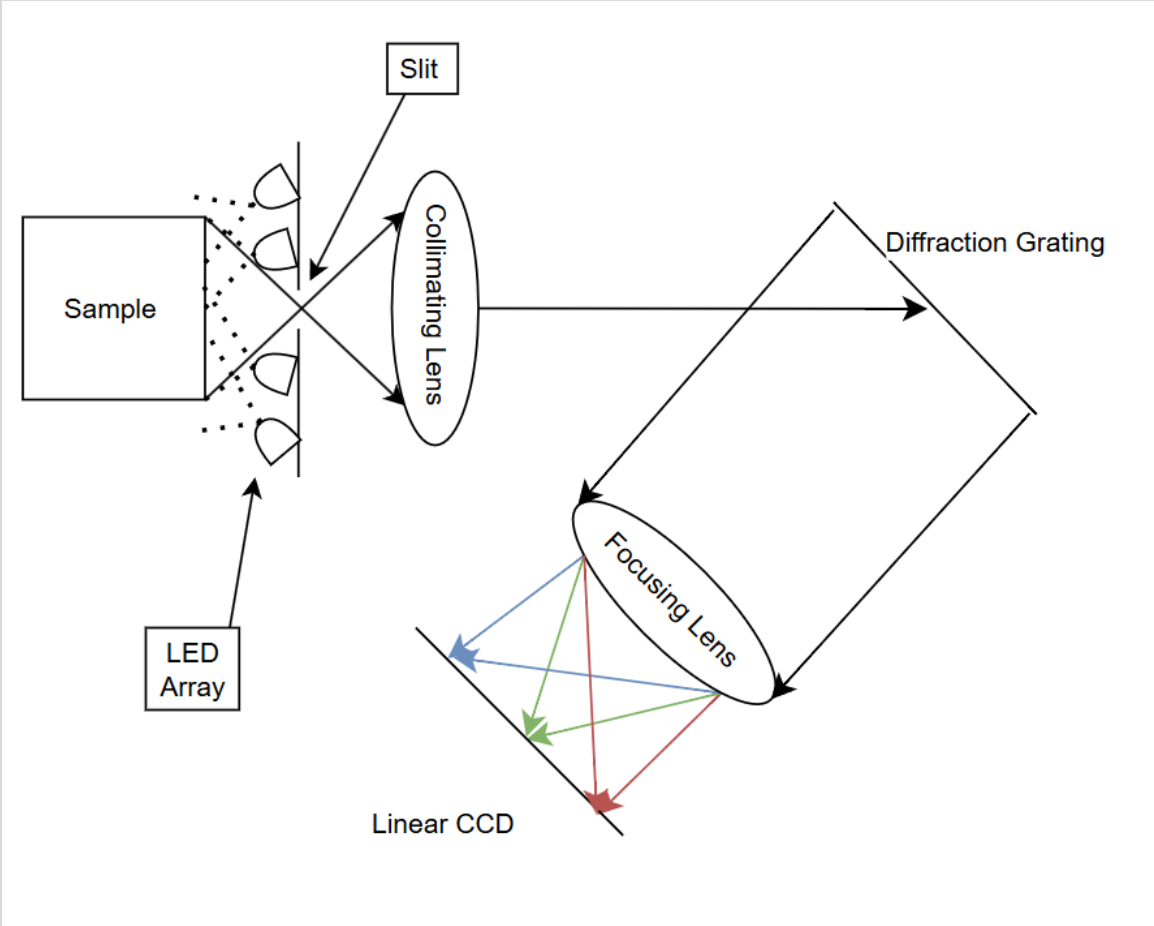


Figure 2: Spectrometer Diagram

2.7 Hardware Diagram

Figure 3 details the work breakdown of the hardware subsystems between the team members.

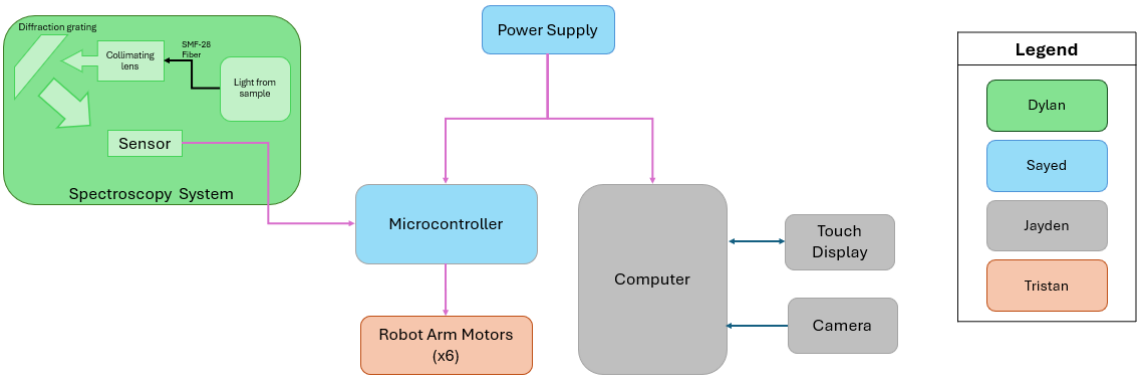


Figure 3: Hardware Block Diagram

Status of each block:

- Power Supply: Researched and designed
- Microcontroller: Researched and acquired
- Computer: Researched and acquired
- Motors and drivers: Researched, acquired, and tested
- Touchscreen Display: Researched and acquired
- Camera: Researched, and acquired
- Spectrometer: Researched and designed (sensor acquired)

2.8 Software Diagram

Figure 4 details the work breakdown of the hardware subsystems between the team members.

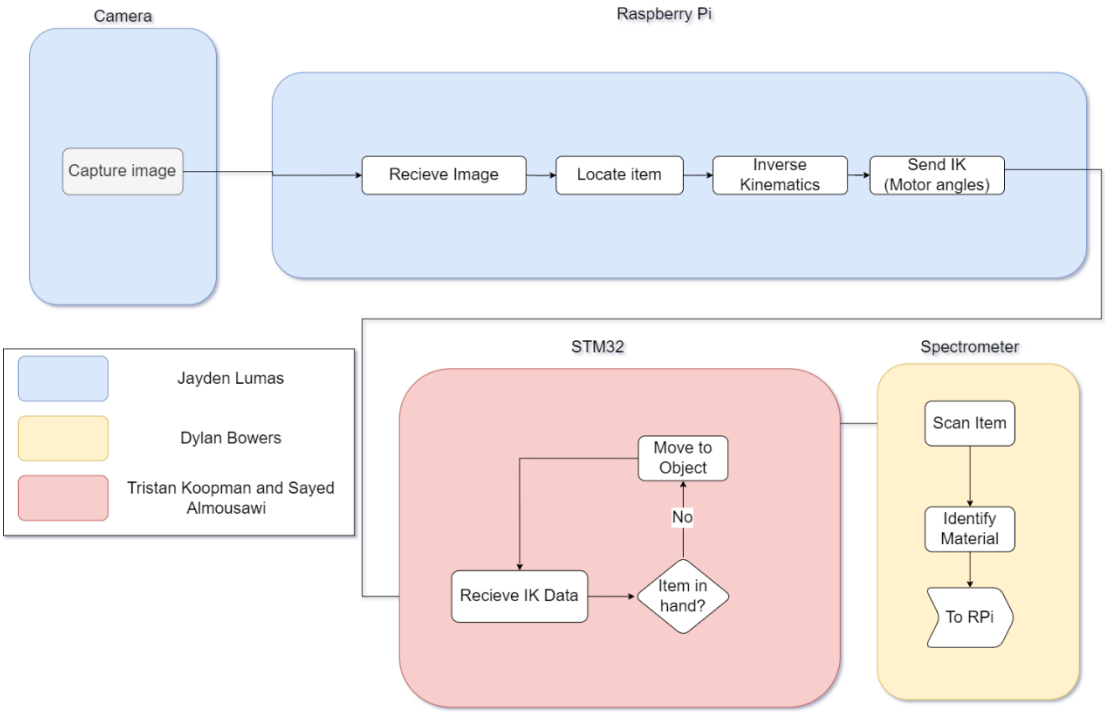


Figure 4. High-Level Software Block Diagram

2.9 House of Quality

This section explains the technical design elements and customer expectations, showing how the House of Quality (HoQ) shown in Figure 5: House of Quality can be a helpful tool for teams to communicate better. The HoQ acts as a central point, making it clear how design features connect with customer needs

, which makes decision-making more effective. By ensuring the final product or service meets customer expectations, the HoQ is important for keeping time and resources are focused on areas of importance.

House of Quality

Design Requirements / Customer Requirements	Importance	Specterum Resolution (nm)	Exposure Duration (msec)	Material Classification Accuarcy (%)	Object Classification Accuarcy (%)	Time to identify (sec)	Hits target within 10% tolerance (Y/N)	Time to calculate position to target (sec)	Time for EE to reach goal (m/s)	User Operation time (sec)	Cost($)	Weight(lbs)	Friction factor	Emergency Button (Y/N)
Can accurately identify multiple objects	19	9	1	9	9	-1					-3			
Can quickly identify multiple objects	10.6		-3	-1	-1	9		3			-1			
Can accurately move different objects	15						9	3	9	1	-1	3		
Can quickly move different objects	9						9	9	9	1	-3	1	1	
Easy to operate	14								3	9	-1			9
Is affordable	7	-9			-1	-3		-1	-3		9	3	-1	
Can be moved around	3										-3	9		
Can lift multiple types of objects	12										-3		9	
Safe to work around	10.4	-1					3		-3		-3	3		9

Figure 5: House of Quality

2.10 Existing Products and Past Projects

In order to better understand the capabilities, limitations, and design decisions of similar robotic systems, we reviewed several existing products and past projects. This section highlights relevant systems that provided insight into our own design, particularly in terms of mechanical structure, control methodology, and hardware implementation.

2.10.1 Annin Robotics AR4 6DOF Robot Arm

The Annin Robotics AR4 is an open-source, 6-degree-of-freedom (6DOF) robotic arm that has become a popular choice among hobbyists and educators due to its accessible design and detailed documentation. The AR4 is fully 3D-printable and built using commonly available components, including NEMA stepper motors and harmonic drive gearboxes for high precision.

One of the key features of the AR4 is its use of industrial-style motion control through the use of Mach4 software and Arduino-based G-code interpreters. This allows the system to perform repeatable, accurate movements suitable for automation tasks.

Although the AR4 is a strong example of a DIY robotic arm, it is more focused on mechanical repeatability and industrial simulation than material detection or sorting tasks. Our project draws inspiration from the modular mechanical design of the AR4, but diverges in terms of application. Instead of preprogrammed movements, we aim to identify real-time material and integrate sensors customized for recycling automation.

2.10.2 AMP Robotics: AI-Powered Robotic Sorters

One of the most well-known developments in this space is by AMP Robotics, a company that has led the charge in artificial intelligence and robotic systems in material recovery facilities (MRFs). AMP’s robotic arms are equipped with computer vision and machine learning algorithms that enable them to identify and sort various types of recyclable materials such as plastics, metals, and cardboard.

AMP's systems rely on a combination of visual data and shape recognition, with high-speed robotic arms capable of making over 80 picks per minute. Their platform, known as the AMP Cortex, is designed to be retrofitted onto existing conveyor belts in MRFs, making it a flexible and scalable solution. Although AMP primarily uses visual identification, the concept of integrating additional sensors, like spectrometers, has been proposed to further improve classification accuracy. [3]

2.10.3 ZenRobotics: Heavy Duty Waste Sorting

ZenRobotics, a Finnish company, has developed industrial-scale robotic waste sorting systems. ZenRobotics' Heavy Picker robots are capable of sorting bulky construction and demolition waste using a combination of advanced sensors, including near-infrared (NIR) spectrometers, 3D scanning, and metal detectors. These systems utilize a multi-sensor fusion approach, combining the benefits of each sensor to identify different material types.

ZenRobotics demonstrates the feasibility and effectiveness of integrating spectrometers for waste identification. Their use of NIR spectrometers allows the robots to distinguish between various types of plastics, textiles, and organics—capabilities that are crucial in achieving high-purity sorting.[4]

2.10.4 MIT CSAIL: RecyBot Research Project

On the academic side, the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT has conducted extensive research in robotic waste sorting. Their project, nicknamed "RecyBot," focuses on a low-cost, adaptive sorting robot that learns from human guidance. Although their system is still in the experimental phase, it highlights a growing interest in combining machine learning with tactile and visual sensing for sorting tasks.

The MIT team's robot is designed to work alongside humans, improving its classification skills over time using reinforcement learning. While the current version uses visual and force feedback, the researchers have discussed the potential benefits of adding spectroscopic sensing to further refine material classification, especially in cases where objects look visually similar but are made from different substances.[5]

2.10.5 Max-AI: Autonomous Quality Control

Max-AI, developed by Bulk Handling Systems (BHS), is another commercially available product that integrates artificial intelligence for robotic waste sorting. It functions as an "autonomous quality control" system, using deep learning algorithms to identify materials with a high level of precision. Though Max-AI is primarily camera-based, it shows the industrial viability of AI-guided sorting arms on conveyor systems.[6]

2.10.6 Fraunhofer Institute: Spectroscopy in Recycling

The Fraunhofer Institute for Process Engineering and Packaging IVV in Germany has been actively researching the application of spectroscopy in recycling. Their work includes the use of Hyperspectral Imaging (HSI) to separate food packaging materials and distinguish between recyclable plastics. The integration of HSI into robotic sorting systems is a promising area that supports our project's goal.

Fraunhofer's researchers have developed systems where HSI sensors are mounted above conveyor belts, capturing detailed spectral data of each passing item. The robot arm then uses this data to determine the material composition before executing the sorting action. [7]

2.10.7 A.P.A.L.E.S. Automatic Material Identification

The APALES project's built-in spectrometer introduces a novel capability to laser engraving: automatic material identification using visible light spectroscopy [8]. This compact module leverages a broadband visible LED to illuminate the workpiece and collects the reflected light through a fiber-optic sensor. By analyzing the unique spectral fingerprint of materials like wood, leather, and cardboard, the system can determine the material type in real time. This eliminates the need for the user to manually select or remember material settings – a first for laser engravers. Once identified, APALES automatically adjusts the laser's engraving parameters (power, speed, focus) optimized for that material, ensuring consistent, high-quality results without trial and error.

The APALES spectrometer can distinguish between materials by their visible spectral response. Wood, leather, and cardboard each reflect a different pattern of wavelengths and intensities when illuminated by white light. For example, wood may show distinct absorption bands or color tints in the spectrum compared to leather. The system's software classifies these spectral patterns with high confidence (exceeding 85–90% accuracy in

identification tests). Importantly, this identification is accomplished rapidly – often in under five seconds – so as not to slow down the engraving workflow.

2.10.8 Key Takeaways for T.R.A.S.H.

From the examples above, several important trends emerge that validate our design concept:

1. **Integration of AI and sensor technologies:** Most leading systems combine traditional computer vision with AI to identify materials. Incorporating a spectrometer, particularly an NIR or HSI system, is a natural evolution in this field and offers improved classification for look-alike materials.
2. **Conveyor-based operation:** Existing solutions often use robotic arms operating above conveyor belts, which aligns with our design. These setups allow for continuous, real-time sorting and can be scaled easily. This matches the intent of our project. However, it has been listed as a stretch goal for the purposes of our Senior Design project.
3. **Adaptability and modularity:** Many of the systems discussed are designed to retrofit into existing facilities. Our robot should be designed with similar flexibility, allowing it to be added to different industrial setups without significant infrastructure changes.

2.11 Marketing Requirements

In order for the T.R.A.S.H. system to succeed beyond its technical specifications, it must also meet a robust set of marketing requirements. These criteria determine how appealing the product will be to its target audience, how competitive it will be in the market, and how easily end users with diverse backgrounds can adopt it. The primary marketing considerations include pricing strategy, ease of use, and advertising reach.

2.11.1 Pricing and Affordability

The affordability of T.R.A.S.H. will play a pivotal role in its market viability. The waste management industry, particularly small-scale and local recycling centers, often faces strict budget constraints. Therefore, pricing the product competitively without compromising performance is essential.

Our development philosophy includes component-level cost efficiency, favoring modular, off-the-shelf parts and 3D-printed custom housings where applicable. Design decisions, including lightweight material selection and integration of low-cost microcontrollers like the ESP32-S3 and STM32F401, support a low bill of materials. The project team aims to keep the total system cost below a threshold while still delivering robust performance in real-time recycling environments.

By minimizing production costs through careful component selection and lean manufacturing techniques, T.R.A.S.H. can be offered at a lower price point than existing industrial robotic sorting systems. This makes the product more accessible to a broader

range of consumers, including community-run recycling programs, educational institutions, and municipal waste facilities.

2.11.2 Usability and Accessibility

Another key marketing requirement is ease of use. Our target demographic includes users with limited or no technical background. While many automated sorting systems require specialized training, T.R.A.S.H. is being designed with user-friendly operation as a priority.

This includes the integration of an intuitive touchscreen interface, simplified setup procedures, and the use of a clear visual feedback system, such as real-time camera feeds. These design choices will reduce the learning curve and encourage user confidence during daily operation. A plug and play architecture also helps make deployment seamless for facilities with minimal IT infrastructure.

Accessibility is further emphasized by minimizing the need for routine maintenance or troubleshooting. The system is designed to be largely self-contained and self-correcting, with diagnostic indicators and easy access points for minor service tasks. All of these usability enhancements help broaden our potential market reach.

2.11.3 Target Audience

The marketing strategy for T.R.A.S.H. begins with identifying the primary user base. We will focus on waste management companies, recycling centers, and sustainability-focused organizations, especially those operating at a small to medium scale and looking for ways to automate without investing in large industrial infrastructure.

Given that our current prototype is optimized for use in stationary environments with conveyor systems (as described in our stretch goals), we will primarily advertise to clients with fixed-position, indoor sorting needs. This includes facilities that process curbside recyclables, e-waste disassembly lines, and even educational labs seeking real-world robotics.

2.11.4 Summary of Marketing Requirements

In summary, our target audience is recycling centers, sustainability organizations, schools, and local facilities. Therefore, our product will be priced so that it is easily accessible for small to medium-sized waste processors. Additionally, our system must be operable with minimal training and have integrated intuitive controls.

3.0 Research

In the following sections, we will discuss the foundational research developed that influenced our final designs.

3.1 Torque Estimations

By providing the Torque Estimations for the TRASH robot we are able to get a better understanding of the requirements and limitations of the robot. From these baseline estimations, we are able to provide a holistic understanding of the cost breakdown required for the robot and insight into the weight requirements needed throughout the robot arm. These estimations are directly used in 3.7 Motors and 8.1 Linkage Structure.

3.1.1 Procedure

Firstly, determining a preliminary estimate of the torque requirements involves making a slew of assumptions. The purpose of these assumptions is to provide a baseline for calculating torque requirements. These assumptions are designed to be an over-exaggeration of what the real prototype design will be. This is done to simplify calculations and to allow room for design. Additionally, by creating these assumptions with outstanding values, we are able to make practical purchasing decisions that require us to assume and investigate more expensive and robust technologies. However, after our prototype designs are finalized and we revisit the necessary torque requirements for the robot arm, we will either be on track with our initial estimates or we will be well below our margins and in a situation where we are under budget rather than over budget.

The assumptions that were made take the loadings in the worst-case scenario. (1) We assume that the arm is purely horizontal, creating the largest moments at the base of each linkage. In practice, the arm will rarely be fully extended. (2) To account for the linkage weight, we assume that the support structure is a solid aluminum rod with a ½ inch diameter and a length of 7 inches. In actuality, an aluminum rod is an overkill for accounting for the internal loadings flowing through the support structure, and if an aluminum rod is being used, it will likely be hollowed out. (3) To account for the connection mechanism connecting the linkages to each other, we assume the connector is a solid cube of 2 inches at 100% infill PETG at each end of the 2 linkages. In actuality, the connector will be far more complicated than a cube and will likely span a larger surface area than 2x2x2. However, the connector will be at most 20% infill and will comprise of mechanisms that have spacings between them. (4) To account for possible gears to adjust gear ratios, we assume a singular cylinder of 1in diameter, ½ in thickness, and 100% infill PETG. In actuality, the gears in use, if used, will be at most 30% infill and will likely be much smaller. (5) In order to account for the weight of the end effector at the end of the robot, we assume the End Effector (EE) is 3 linkages of 2in x ½in x 1in at 100% infill PETG. In actuality, the EE will be much more complicated than 3 rectangular linkages and has the possibility of spanning a larger workspace. However, the EE will have at most a 20% infill. (6) In order to account for the weight of the motors, we assume that each motor is placed at the

intersection of the linkages and weighs 0.2 lbs. In actuality, the weight of the motors will decrease further from the base we travel. (7) to account for the workspace of the wrist, we assume that the length of the wrist is ½ in. However, the size of the wrist is unknown. (8) The final assumption is that the arm is holding 2 lbs. at the end of the EE. In actuality, we are only required to hold 1 lb of weight with the EE. Additionally, it would be poor practice for the robot to grab items with the tips of the EE rather than at the center of the gripper. This excess weight serves to round off the numbers, providing an additional Factor of Safety (FS), and allows the option of including other, more complicated means of acquiring objects.

Name	Accounting	Assumption	Reality
(1) Arm Position	Worst Case position	Worst-case position is horizontal & fully extended	Arm will rarely be at this position.
(2) Support Structure	Weight of the Support Structure	Solid aluminum rod ½in ø x 7in. for 2 linkages	Lighter material; ifelse: rod will be hollow
(3) Connectors	Weight of connection mechanism of linkages	2in x 2in x 2in 100% infill PETG at each end of 2 linkages	Larger area, more spacing, ≤ 20% infill
(4) Gears	Weight of gear trains	1in ø x ½in 100% infill PETG at each joint	May not be used, if used, likely smaller, ≤ 30% infill
(5) End Effector	Weight of EE	3 linkages of 2in x ½in x 1in 100% infill PETG	More complicated, larger workspace, ≤ 20% infill
(6) Motors	Weight of motors	Placed at each joint, weighs 0.2lbs	Weight of motors will decrease further from base
(7) Wrist	Size of the wrist workspace	Length is ½in	Length is unknown
(8) Object	Weight of item being acquired	Object weighs 2lbs	Only required to lift a 1lb object, not at tip of EE

Table 6 Robot Arm Weight Assumptions

Linkage Lengths		
x1	7	in
x2	7	in
x3	0.5	in
x4	2	in

Table 7: Prototype Arm Dimensions: Length of linkages

Linkage Connectors	
l	2 in
w	2 in
h	2 in

Table 8: Prototype Arm Dimensions: Connector

Motor Weights	
M1	0.2 lb
M2	0.2 lb
M3	0.2 lb
M4	0.2 lb
M5	0.2 lb
M6	0.2 lb

Table 9: Prototype Arm Dimensions: Weight of Motors

Rods	
Radius1	0.25 in
Radius2	0.25 in

Table 10: Prototype Arm Dimensions: Rod

End Effector Dimensions	
l	2 in
w	0.5 in
t	1 in

Table 11: Prototype Arm Dimensions: EE

Gear	
Radius	0.5 in
Depth	0.5 in

Table 12: Prototype Arm Dimensions: Gear

3.1.2 Calculations

Step 1: Calculate the different loads being acted upon the robot arm.

For calculating the weight of the rod, first we need to calculate the volume of the rods. The formula for calculating the volume of a rod is given in (3.1.1).

$$V_r = \pi r^2 h \tag{3.1.1}$$

Substituting the values of the linkage from Table 10: Prototype Arm Dimensions: Rod into (3.1.1) gives

$$V_{r1} = V_{r2} = \pi(0.25in)^2(7in) = 1.37in^3$$

The aluminum rod being used is an Everbilt 1/2 in. x 3 ft. Aluminum Solid Round Rod Extruded with aluminum alloy 6063-T5. Aluminum alloy 6063-T5 has a material density of $\rho = 0.0975\ lb/in^3$ [9].

Using the calculated volume and the provided density, we calculate the weight of each rod using (3.1.2).

$$W_r = V_r \rho_A \tag{3.1.2}$$

$$W_r = (1.37in^3)\left(0.0975\frac{lb}{in^3}\right) = 0.134lb$$

By repeating these steps and using Table 12: Prototype Arm Dimensions: Gear we are able to calculate the volume and weight of the gear.

$$V_G = \pi r^2 h \tag{3.1.1}$$

$$V_G = \pi(0.5)^2(0.5) = 0.393in^3$$

PETG is used as it has a higher density compared to PLA, providing an additional increase in FS. The density for Overture PETG (a common and reliable brand) is given as $\rho = 1.25\ g/cm^3$, which converts to $0.04516\ lb/in^3$ [10].

$$W_G = V_G \rho_{PETG} \tag{3.1.2}$$

$$W_G = (0.393in^3)\left(0.04516\frac{lb}{in^3}\right) = 0.0177lb$$

Next, using Table 8: Prototype Arm Dimensions: Connector we will calculate the connectors that will be located at the end of each linkage using Overture PETG,

$$V_C = lwh \tag{3.1.3}$$

$$V_C = (2)(2)(2) = 8in^3$$

$$W_C = V_c \rho_{PETG} \tag{3.1.2}$$

$$W_C = (8in^3)\left(0.04516\frac{lb}{in^3}\right) = 0.361lb$$

Repeating the same steps with Table 11: Prototype Arm Dimensions: EE and the same material, we are able to calculate the weight of the EE;

$$V_{EE} = lwh \tag{3.1.3}$$

$$V_{EE} = (2)(0.5)(1) = 1in^3$$

$$W_{EE} = V_{EE} \rho_{PETG} \tag{3.1.2}$$

$$W_{EE} = (1in^3)\left(0.04516\frac{lb}{in^3}\right) = 0.04516lb$$

After calculating the applied loads, we can now apply the calculated loadings along the arm. Because the weight of each segment is a uniformly distributed loading applied equally across the surface area, we are able to take the resulting load and calculate it at the centroid of the segment. Thus simplifying our calculations.

Step 2: Calculate Moments

In order to calculate the standing torque required of the motors, we must calculate the moments created at the motor positions by the applied loadings. To do this, we will apply the sum of moments **Error! Reference source not found.** at each motor location to calculate the resulting moment acted upon by the motor by the various loadings starting from the motor furthest from the base.

$$\sum M = 0 = Fx \tag{3.1.4}$$

There are 6 motors in this design. One providing rotational movement of the entire arm assembly, one motor providing motion to each of the two linkages, and 3 motors comprising the wrist and gripper motion. For the purposes of determining static torque, only motors 2-6 need to be considered. It is important to note that for simplicity of calculations, Motors 5&6 are coincident with each other at the centroid of the wrist mechanism.

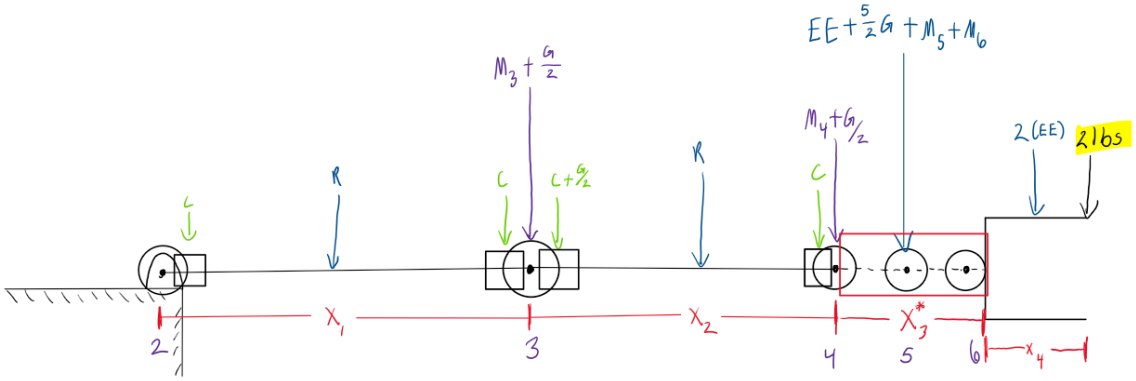


Figure 6 Free Body Diagram

For Motors 5&6, these motors must support the weight of the object being held, along with the two pincers of the EE. Thus, the Moment equation is as follows:

$$\sum M_{5,6} = 0 = -(2lb)\left(\frac{x_3}{2} + x_4\right) - 2(W_{EE})\left(\frac{x_4}{2} + \frac{x_3}{2}\right) \tag{3.1.5}$$

Motor 4 includes the same loadings as Motors 5&6 with increased offset distance, along with the inclusion of the full wrist mechanism.

$$\begin{aligned}\sum M_4 = 0 = &-(2lb)(x_3 + x_4) - 2(W_{EE})\left(x_3 + \frac{x_4}{2}\right) \\ &- (W_{EE} + \frac{5}{2}W_G + W_{m_5} + W_{m_6})(\frac{x_3}{2})\end{aligned} \tag{3.1.6}$$

Motor 3 includes all of motors 4-6, along with the second rod and respective connectors.

$$\begin{aligned}\sum M_3 = 0 = &-(2lb)(x_2 + x_3 + x_4) \\ &- 2(W_{EE})\left(x_2 + x_3 + \frac{x_4}{2}\right) \\ &- \left(W_{EE} + \frac{5}{2}W_G + W_{m_5} + W_{m_6}\right)\left(x_2 + \frac{x_3}{2}\right) \\ &- \left(W_{m_4} + \frac{W_G}{2}\right)(x_2) - (W_C)\left(x_2 - \frac{C_l}{2}\right) \\ &- (W_R)\left(\frac{x_2}{2}\right) - (W_C + \frac{W_G}{2})(\frac{C_l}{2})\end{aligned} \tag{3.1.7}$$

Motor 2 includes all of motors 3-6, along with the first rod and respective connectors.

$$\begin{aligned}\sum M_3 = 0 = &-(2lb)(x_1 + x_2 + x_3 + x_4) \\ &- 2(W_{EE})\left(x_1 + x_2 + x_3 + \frac{x_4}{2}\right) \\ &- \left(W_{EE} + \frac{5}{2}W_G + W_{m_5} + W_{m_6}\right)\left(x_1 + x_2\right. \\ &\left. + \frac{x_3}{2}\right) - \left(W_{m_4} + \frac{W_G}{2}\right)(x_1 + x_2) \\ &- (W_C)\left(x_1 + x_2 - \frac{C_l}{2}\right) - (W_R)\left(x_1 + \frac{x_2}{2}\right) \\ &- \left(W_C + \frac{W_G}{2}\right)\left(x_1 + \frac{C_l}{2}\right) \\ &- \left(W_{m_3} + \frac{W_G}{2}\right)(x_1) - (W_C)\left(x_1 - \frac{C_l}{2}\right) \\ &- (W_R)\left(\frac{x_1}{2}\right) - (W_C)(\frac{C_l}{2})\end{aligned} \tag{3.1.8}$$

Plugging in values to equations (3.1.5), (3.1.6), (3.1.7) and (3.1.8) from Table 7, Table 9, and Table 11 along with the weights calculated previously, we get the following minimum static torque requirements:

$$T_6 = T_5 = 52.12\ Ncm$$

$$T_4 = 59.41\ Ncm$$

$$T_3 = 313.93\ Ncm$$

$$T_2 = 653.32\ Ncm$$

3.2 Material Selection

A critical factor in the design of a robot arm structure is the material. For any robot weight is a limiting factor directly affecting the resulting success or failure of the robot. In the case of a robot arm, this fact is painfully obvious. After calculating the Torque Estimations in Calculations we see that any weight acted at the end of a moment arm has cascading loads affecting the motors. For this reason, it is critical that we choose a material that is extremely lightweight.

The materials being compared will be investigated based on their weight, cost, and bending stiffness and strength. Weight will be evaluated based on density (g/cm^3) as it is not dependent on the volume of the material. This is important because for materials with higher Bending Stiffness and Strength, the volume will decrease. Bending stiffness will be determined based on Young's Modulus, which is "The ratio of stress to strain when deformation is totally elastic; also a measure of the stiffness of a material" [9]. Bending Strength can be represented as either Tensile Strength or Yield Strength. Tensile Stress involves plastic deformation, whereas Yield Strength is the highest point before deformation [9] we will use Yield Strength for our application. The cost will be represented in US dollars.

3.2.1 Aluminum

Aluminum is an obvious candidate for T.R.A.S.H. as it is an abundant resource commonly used in engineering for its lightweight and adequate material properties. There are multiple types of Aluminum Alloys used in manufacturing. However, we will be comparing them with Annealed 6063-T5 as it is common and readily available.

Aluminum is one of the lighter metals that is commonly used in manufacturing, with a density of only around 2.70 (g/cm^3) [9]. Aluminum also has decent strength with a Modulus of Elasticity of 69 GPa and a Yield Strength of 55 MPa [9]. It is due to its lightweight properties, decent strength, and simple manufacturing that it is commonly used in transportation.

3.2.2 Carbon Fiber

Carbon fiber has become a trend in the materials industry. It is rare to find college and home projects alike that do not use some form of carbon fiber element. Carbon fiber has become a buzzword in the same way Artificial Intelligence has, and for good reason due to its adjustable loading capabilities and lightweight.

Carbon fiber is an extremely challenging material to quantify and measure. Therefore, for simplicity, we will be using "Materials Science and Engineering" by Callister and Rethwisch [9] analysis of "Standard Carbon Fiber". Carbon Fiber is composed of layers of interwoven strands of carbon filament cured in resin prepreg, creating a rigid structure from molds or casts. Carbon fiber is relatively lightweight with a density of 1.6 (g/cm^3).

However, with a strength comparable to steel, much less material is needed to perform engineering applications. In comparison, steel, with a Modulus of Elasticity of 207 GPa has a density of 7.85(g/cm^3). Standard Carbon Fiber has a Modulus of Elasticity of 145 GPa, meaning Carbon Fiber has more than a 3 times strength-to-weight ratio. Carbon fiber is a brittle material and therefore does not have a yield strength. However, its Tensile Strength is 1240 GPa, making it an extremely effective and weight-efficient material. However, with so many benefits there comes a cost, and that cost is monetary. While manufacturing practices for carbon fiber have improved drastically, it still costs more for less material.

3.2.3 Steel

Steel is a common material used in engineering applications for its pure strength. Very common material for robotics, still can be seen in most industrial robotic arms, a field with which our project is looking to enter. For simplicity, we will be making our comparison with Hot-rolled Steel alloy A36 [9].

For industrial robot arms that are doing precise motion, a common issue arises in the realm of controls. To simply fix this issue, roboticists increased the weight of the robot arms in order to decrease the overshoot without compromising rise time. Therefore, most industrial robot arms use titanium and steel, which are both heavy and costly to manufacture. However, we are not aiming to solve any control problems for our robot arm. Steel has a density of 7.85 (g/cm^3), a Modulus of Elasticity of 207 GPa, and a yield strength of 220-250 MPa. Due to its high weight-to-strength ratio, Steel is not an option for our robot.

3.2.4 PETG

PETG, or Polyethylene Terephthalate, is a common filament used for FDM printing for most hobbyists. Compared to other filaments used, PETG has a higher Yield and Tensile strength with a slightly increased weight. For purposes of comparison to the other materials, we will be referencing Overture PETG [10] as it is a well-known and used brand.

PETG is a lightweight material with a density of 1.25 (g/cm^3). FDM manufacturing is also an extremely easy and efficient method for manufacturing custom and complex pieces. However, with a Youngs Modulus of 1986 MPa and Tensile strength of 44.3 MPA, it is orders of magnitude weaker than other materials. It is also a very unreliable material as slight alterations to nozzle temperature, bed temperature, humidity, and speed can all affect the quality of the material. Meaning the material can likely fail at a much lower loading.

3.3 Computer Vision (CV)

Computer Vision is an essential component of modern robotic systems, enabling machines to perceive and interpret visual information for enhanced automation and efficiency. By utilizing advanced image processing techniques, computer vision allows robots to detect, classify, and interact with their environment with high precision. In waste recycling, this

capability is crucial for accurately identifying and sorting materials, reducing reliance on manual labor, and improving overall system sustainability.

For the T.R.A.S.H. project, computer vision serves as the foundation for workspace visualization and object detection. Traditional sorting methods are often limited by the need for physical contact or constrained sensing mechanisms. In contrast, a computer vision system provides a holistic view of the sorting environment, allowing the robot to identify, track, and categorize objects before they reach the end effector. This proactive approach streamlines the sorting process and enhances the overall efficiency of our system.

3.3.1 Comparison with Alternative Detection Methods

3.3.1.1 Ultrasound Sensors

Ultrasound sensors operate by emitting high-frequency sound waves and measuring the time delay of the echo after bouncing off an object. These sensors are widely used for distance measurement and obstacle detection due to their low cost and robustness in different lighting conditions. However, despite these benefits, ultrasound sensors present major challenges when applied to our sorting system:

Limited Field of View & Continuous Scanning Requirement: Since ultrasound sensors can only detect objects directly in front of them, the robotic arm would need to continuously scan the entire workspace to locate objects. This is problematic because the arm must actively sweep across the workspace to search for height and distance discrepancies. Additionally, if multiple objects are present, the system may accidentally knock over previously scanned and unscanned items, forcing re-scans and making object tracking unreliable.

Edge Detection and Object Centering Requirement: One of the key limitations of using an ultrasonic sensor is the lack of shape and edge detection. To accurately grasp an object, especially when the end effector and the target object have precise dimensions, the robot would need to scan around the object to determine its edges and overall size. Since ultrasonic sensors only provide single-point distance measurements rather than a full outline, the system would require additional movement to estimate the center of the object. This adds complexity and reduces the efficiency of the sorting process.

Environmental Interference: Ultrasonic sensors are highly sensitive to environmental conditions that can affect the speed and reliability of sound wave propagation. Variables such as temperature, humidity, and air pressure directly influence the velocity of sound in air, leading to drift in distance measurements over time. When operating over longer periods, these fluctuations can compound, causing the sensor to deliver inconsistent readings across the same workspace. This creates challenges in maintaining accuracy, especially when precision is required for repetitive sorting tasks.

Material Absorption and False Readings: The performance of ultrasonic sensors also depends on how well sound waves reflect off surfaces. Certain materials, such as rubber, foam, and fabric, can absorb sound energy rather than reflect it. When this happens, the sensor may receive weakened or no echo at all, resulting in false, delayed, or entirely missed readings. This poses a serious issue for object detection and sizing, particularly when dealing with mixed-material waste streams where surface properties vary significantly.

Surface Geometry and Precision Limitations: Curved or irregular surfaces, like cylindrical cans or plastic bottles, present another obstacle. Sound waves can be deflected away from the sensor at unpredictable angles, which causes errors in calculating the object's position or shape. Combined with the sensor's inherently low spatial resolution, this limits the robot's ability to reliably identify object boundaries or centers. In applications where precise alignment between the end effector and the object is necessary, these inconsistencies can lead to frequent pickup failures or misclassifications.

Due to these limitations, ultrasound sensors are not ideal for our workspace, as they restrict flexibility, require continuous movement, and struggle with complex object geometries.

3.3.1.2 LiDAR Sensors

LiDAR (Light Detection and Ranging) is a sensing technology that uses laser pulses to measure the distance to objects and generate detailed 3D point cloud maps of the environment. It is commonly used in autonomous vehicles, surveying, industrial automation, and robotic mapping because of its ability to deliver high-resolution spatial data with accuracy. However, despite these capabilities, LiDAR presents several limitations for the T.R.A.S.H. system:

High Cost: LiDAR units are significantly more expensive than other sensing solutions. Even entry-level LiDAR systems often cost hundreds to thousands of dollars, which exceeds the budget constraints of this project. Since cost-efficiency is a major requirement for T.R.A.S.H., LiDAR is not a financially practical option.

Memory and Processing Demands: LiDAR systems generate dense 3D point clouds that require substantial memory and processing power to analyze. This presents a major issue for T.R.A.S.H., which is designed to operate on lightweight, embedded processing platforms. Attempting to run real-time detection and classification from LiDAR data would result in delays, inefficiencies, or system overload.

Limited Material Identification: While LiDAR excels at identifying object shape and spatial position, it does not detect color, texture, or material composition. For a sorting

system where material type is the defining sorting parameter, this makes LiDAR unsuitable without pairing it with additional hardware such as spectrometers or vision systems.

Environmental Sensitivity: Although LiDAR is less affected by lighting than traditional cameras, it can still struggle in certain indoor conditions. In our case, the system will operate indoors, where reflective surfaces such as aluminum cans or glossy plastics, as well as translucent materials like plastic bags or clear bottles, may be present. These materials can scatter or absorb the laser pulses, leading to incomplete point clouds, false readings, or missing object data. These challenges reduce the reliability of LiDAR in accurately detecting recyclable items within the T.R.A.S.H. workspace.

Overhead in Multi-Object Environments: LiDAR systems are often optimized for mapping or vehicle-scale environments rather than small, cluttered workspaces with many overlapping or touching objects. Distinguishing individual recyclables in a dense pile using point cloud data alone is computationally complex and would require additional algorithms to parse object boundaries and determine grasp points.

While LiDAR provides excellent 3D spatial awareness, its high cost, processing demands, and inability to detect material properties make it an inefficient choice for T.R.A.S.H. Given the nature of our indoor sorting environment, where reflective and translucent materials are common, the reliability of LiDAR further decreases. For a system that prioritizes cost-effectiveness, accuracy, and material classification, LiDAR does not meet the practical or technical requirements of our design.

3.3.2 Advantages of Computer Vision

To overcome the limitations of ultrasound and LiDAR, we propose using a computer vision camera for object detection. This method offers a balanced combination of accuracy, adaptability, and affordability, making it the most suitable option for the T.R.A.S.H. system.

Affordability: Computer vision cameras are significantly more cost-effective than LiDAR systems while still providing dependable object detection. This makes them ideal for budget-conscious applications without sacrificing essential functionality.

Consistent and Wide-Area Coverage: Unlike ultrasound sensors that require constant movement to scan the workspace, a single camera can capture a broad and stable view of the entire area in one frame. This reduces the need for mechanical scanning and allows the system to track multiple objects efficiently.

Flexible Object Identification: Computer vision excels in identifying a wide range of objects based on shape, color, texture, and material appearance. Unlike ultrasound, which

struggles with curved or soft surfaces, and LiDAR, which only maps shape and distance, computer vision provides meaningful visual context. This is essential for distinguishing between different types of recyclables, such as plastic bottles, aluminum cans, and paper products.

Adaptability with Machine Learning: Computer vision systems can be enhanced through integration with machine learning and AI models, allowing the system to continuously improve in recognizing and classifying objects. Over time, as more data is gathered, these models can be retrained to increase accuracy and adapt to changes in object appearance, orientation, or environmental conditions. Additionally, there is a wide range of existing cloud-based solutions and frameworks that can accelerate development and improve detection performance. Services such as Amazon Web Services (AWS) Rekognition and Microsoft Azure Computer Vision provide robust, pre-trained models that can be adapted or integrated into our system. These platforms offer tools for object detection, image classification, and anomaly recognition, reducing the development time and increasing reliability through proven architectures.

3.3.3 Object Detection Technology Selection

In summary, this section evaluated three object detection technologies: ultrasound sensors, LiDAR, and computer vision, to determine the most suitable solution for the T.R.A.S.H. system. Ultrasound sensors, while low-cost and reliable in certain environments, were ruled out due to their limited scanning capabilities, environmental sensitivity, low precision, and poor performance with curved or sound-absorbing surfaces. LiDAR sensors offer high spatial accuracy but were dismissed because of their high cost, large processing requirements, difficulty operating with translucent or reflective indoor materials, and inability to classify materials.

Computer vision emerged as the most effective and scalable solution. It provides broad workspace coverage, real-time classification, and the flexibility to detect recyclables based on shape, color, texture, and material appearance. When paired with machine learning, computer vision enables continuous improvement in object recognition accuracy. The availability of cloud-based AI tools such as AWS Recognition and Microsoft Azure Computer Vision further enhances development efficiency and detection reliability.

By integrating computer vision into the T.R.A.S.H. system, we achieve a solution that is cost-effective, precise, and well-suited for handling dynamic recycling environments. This technology supports our goal of building an automated sorting system that can accurately identify, classify, and process recyclables with minimal hardware overhead and maximum adaptability.

3.4 Microcontroller

Microcontrollers (MCUs) are essential components in embedded systems and robotics. They manage all of the hardware components, process sensor data, and execute control tasks. An MCU has a central processing unit (CPU), memory, and I/O peripherals on a single chip, designed for efficient embedded applications.

For our project, the MCU is responsible for automating various tasks, including processing sensor data, controlling the robotic arm motors, and managing system communication. Given the need for precise real-time control of the robotic arm, the selected MCU must offer sufficient computational power, peripheral support, and energy efficiency.

Additionally, the spectroscopy system in our design uses a Linear CCD sensor which requires an MCU to generate very specific and complex clock signals, acquire sensor output signals, and transmit data to other subsystems. These tasks require efficient resource management to ensure accuracy and reliability. The Linear CCD Sensor requirements are discussed in more detail in section 6.2 but basically, it requires 4 timers (at least one 32-bit timer), ADC, DMA, and a fast serial communication protocol.

To meet these requirements, we will evaluate whether a single high-performance MCU is sufficient or if multiple MCUs operating in parallel would provide better efficiency. If multiple MCUs are used, they must communicate effectively using a reliable communication protocol to facilitate seamless data exchange. Since MCUs are generally cost-effective, this approach could optimize performance without significantly increasing manufacturing costs.

The choice of an MCU depends on several critical specifications. Computational power is determined by the CPU architecture and clock speed, which affects the ability to process data in real-time. Memory resources, including RAM and Flash storage, must be sufficient to handle real-time data processing and program execution. Peripheral support is another crucial factor, as the MCU needs to include essential features such as GPIO pins, analog-to-digital converters (ADC), pulse-width modulation (PWM), direct memory access (DMA), and communication interfaces like I^2C, SPI, and UART for integration with sensors and actuators. Power consumption must also be optimized for efficiency, particularly for continuous operation. Additionally, the development ecosystem is another critical consideration. An ideal MCU should be well-documented, have strong community support, and offer robust development tools and software libraries.

Another key consideration is the data size, as the architecture of an MCU varies between 8-bit, 16-bit, and 32-bit designs. An 8-bit MCU is suitable for basic control tasks with minimal processing demands, while a 32-bit MCU provides higher data throughput and better computational capabilities and is ideal for applications requiring real-time

processing. Considering these criteria, the following sections will provide a detailed analysis of each MCU we are considering before concluding with our final selection based on our project requirements.

3.4.1 ATmega328P

The first MCU under consideration is the ATmega328P, an 8-bit AVR RISC-based microcontroller developed by Atmel. It is widely used in Arduino platforms due to its low cost, simplicity, and power efficiency. The ATmega328P has a maximum CPU frequency of 20 MHz, 32 KB of Flash memory, 2 KB of SRAM, and 1 KB of EEPROM. It features 23 GPIO pins and supports communication interfaces such as UART, I²C, and SPI. It also includes three timers: two 8-bit and one 16-bit, along with six PWM channels.

While the ATmega328P is a reliable MCU for basic embedded applications, it presents several limitations for our project. The 8-bit architecture and 20 MHz clock speed are insufficient for handling the complex real-time control tasks required by the robotic arm and spectroscopy system. Additionally, its limited peripheral support, particularly the availability of only three timers, may not be adequate for generating motor control signals and meeting the precise timing requirements of the spectroscopy system, which requires at least one high-resolution 32-bit timer and multiple PWM outputs. The microcontroller also features only one UART/USART port, which could become a bottleneck when interfacing with multiple serial devices simultaneously. Another significant limitation is the lack of a DMA controller, which is essential for efficient data acquisition from the spectroscopy sensor. Without DMA, the CPU would need to handle data transfers manually, leading to processing delays and inefficiencies.

Due to these constraints, the ATmega328P is most likely not enough to meet our project's requirements. While it remains a viable option for simpler control tasks, our project demands a more capable MCU with better peripherals support and development features. Further comparison with other microcontrollers will help determine the best choice for our specific needs.

3.4.2 ESP32-S3-WROOM-1-N8

Our next consideration is the ESP32-S3-WROOM-1-N8, it is a powerful 32-bit microcontroller designed by Espressif, featuring a dual-core Xtensa LX7 processor with a maximum clock speed of 240 MHz. It is becoming a popular microcontroller due to its built-in Wi-Fi and Bluetooth capability. It includes 8 MB of Flash memory, 384 KB of ROM, and 512 KB of SRAM. It has 36 GPIO pins and supports various communication protocols, including three UART interfaces, two I²C interfaces, and four SPI interfaces. The microcontroller features four timers, all with 54-bit resolution based on a 16-bit prescaler. For pulse-width modulation (PWM), it provides eight LED PWM channels and two motor control pulse-width modulation (MCPWM) channels. It consists of two 10-channel 12-bit SAR ADCs, supporting up to 20 channels in total.

While the ESP32-S3 is capable of the tasks for this project and despite its high processing power and integrated Wi-Fi/Bluetooth capabilities, it is not the best option for the linear CCD sensor due to a lack of sufficient timer configurations needed for driving it. Additionally, we found that its ADC performance is very poor according to the reviews of people who tried it. While the ESP32-S3 does support direct memory access (DMA), its timing precision and peripheral integration are less suited to the high-speed, real-time operations required in our system. Due to these limitations, we have decided to look for an alternative microcontroller that is more suitable for our linear CCD sensor, but still keep the ESP32-S3 for the motor controls since it is very easy to program, and supports many communication protocols.

3.4.3 STM32F401

After evaluating multiple options, we have selected the STM32F401 as the microcontroller for our project. This MCU, part of the STM32F4 series developed by STMicroelectronics, features a 32-bit ARM Cortex-M4 core running at a maximum clock speed of 84 MHz. It has 256 KB of Flash memory, 512 bytes of one-time programmable (OTP) memory, and 64 KB of RAM. It includes 36 GPIO pins and supports a variety of communication protocols, including three UART interfaces, three I²C interfaces, and three SPI interfaces.

The STM32F401 is particularly well-suited for our application due to its extensive timer support. It features two 32-bit timers and six 16-bit timers, providing more than enough timing resources for the spectroscopy system. The STM32 series also offers clear and detailed documentation, particularly regarding the timer trigger registers, enabling precise synchronization and advanced control using the STM32CubeIDE development environment. Additionally, its ADC system consists of 10 channels with 12-bit resolution and a much faster sampling rate compared to the ESP32-S3.

Another advantage of the STM32F401 is its support for DMA, which allows for efficient data transfer without overloading the CPU. Unlike the ESP32-S3, which includes additional processing overhead due to its wireless connectivity, the STM32F401 provides a more streamlined and deterministic approach to real-time control. Given these advantages, the STM32F401-CCU6 is the optimal microcontroller for our spectrometer, offering a combination of great computational power, timer precision, and ADC performance.

MCU Selection

Feature	ATmega328P	ESP32-S3-WROOM-1	STM32F401
CPU Core	AVR 8-bit	Xtensa Dual-core 32-bit LX7	ARM Cortex-M4 32-bit
Max Clock Frequency	20 MHz	240 MHz	84 MHz
GPIOs	23 pins	36 pins	36 pins
USB	NA	OTG FS	OTG FS
UART	1xUSART	3	3xUSART
I2C	1	2	3
Timers	2xTimer 8-bit 1xTimer 16-bit	4xTimer 54-bit (16-bit prescaler)	2xTimer 32-bit 6xTimer 16-bit
PWM	6 channels	8 channels LED PWM, 2 MCPWM	4 channels/timer
ADC	8 channels 10-bit	2xSAR ADC 10 channels 12-bit	10 channels 12-bit
DMA	NA	Yes	Yes
Wi-Fi/Bluetooth	NA	Yes	NA
Cost	~$3	~$5	~$5

Table 13: MCU Selection

As explained in the sections above, we chose two microcontrollers from Table 13. Since each one has its own capabilities that we would need for our project. The three UARTs, I2C, and ease of programming capabilities of the ESP32-S3 are very useful for our motor control system. Meanwhile, the STM32 timers' capabilities are exactly what we need for the spectrometer system.

3.5 Single Board Computer (SBC)

Single Board Computers (SBCs) are compact, fully integrated computing systems that incorporate a processor, memory, storage, and I/O peripherals on a single circuit board. Unlike microcontrollers (MCUs), which are optimized for real-time control of hardware, SBCs are designed for high computational tasks, advanced processing, and efficient communication between multiple components in an embedded system. They run full-fledged operating systems, such as Linux, allowing for software flexibility, modularity, and networking capabilities.

For our project, an SBC is essential due to the high computational demands that an MCU alone cannot handle. The system must process real-time data from multiple sources, including a camera for object identification, a spectrometer for material classification, and motor control for actuation. The SBC must support high-speed communication protocols that allow efficient data transfer between these modules and ensure smooth integration with other hardware components. Additionally, the requirement for modular parts necessitates an adaptable computing platform capable of handling software and hardware updates without significant reconfiguration.

Another key factor is the necessity of Robot Operating System 2 (ROS2) for seamless integration and control. ROS2 provides a robust middleware framework that enables efficient communication between different subsystems, such as computer vision, motion planning, and sensor fusion. A Single Board Computer is the only device that can efficiently host ROS2 while managing concurrent tasks, such as machine learning inference, sensor data aggregation, and real-time motor control.

The selection of an SBC for our project depends on several key factors, including processing power, memory, I/O capabilities, software compatibility, power efficiency, and expandability. Given that our system must handle image processing, machine learning, and real-time control, the CPU and GPU performance of the SBC are critical considerations. Additionally, the availability of high-speed interfaces like USB 3.0, MIPI CSI for camera modules, and GPIO for motor and sensor control ensures compatibility with all required hardware components.

The following sections will analyze different SBCs, comparing their performance and capabilities before finalizing our selection based on our project requirements.

3.5.1 Raspberry Pi 5

One of the Single Board Computers (SBCs) we are considering for our project is the Raspberry Pi 5 (8GB) due to its high computational performance, extensive I/O capabilities, and flexibility for real-time processing. The Broadcom BCM2712 quad-core Arm Cortex-A76 processor (64-bit) clocked at 2.4GHz is a significant upgrade over previous models, with improved out-of-order execution, larger caches, and enhanced instruction pipelines, making it well-suited for parallel processing of sensor data, motor control commands, and real-time vision processing for robotic applications. The LPDDR4X-4267 SDRAM (8GB) ensures a high memory bandwidth, allowing for efficient data transfer between the CPU and peripherals, particularly crucial for handling multiple high-resolution image streams from cameras and storing large datasets for real-time processing. Additionally, the 8GB of SDRAM allows us to allocate more memory to ROS2 nodes, reducing latency and improving task scheduling efficiency.

For our computer vision system, the dual 4-lane Mobile Industry Processor Interface (MIPI) Camera Serial Interface (CSI) camera connections provide high-bandwidth support for stereo vision or multiple camera feeds, enabling real-time object identification and spatial analysis for robotic arm manipulation. The RP1 I/O controller offloads all USB, GPIO, PCIe, and MIPI CSI/DSI communication from the CPU, enabling independent high-speed data transfer for cameras, sensors, and actuators without system bottlenecks. This allows for full-speed USB 3.0 (5Gbps), optimized GPIO for real-time motor control, and dedicated camera interfaces, ensuring low-latency coordination between vision processing and robotic actuation while maximizing overall system efficiency. Another notable addition is the PCIe 2.0 interface, which, while not essential, provides valuable expansion options such as NVMe SSD storage for high-speed access to image datasets, machine learning models, and logged sensor data. Similarly, the UART debug port, though not a requirement, enhances low-level debugging and system monitoring, making it a useful feature for firmware updates and diagnostics.

While the Raspberry Pi 5 does not support RAM expansion, its 8GB of LPDDR4X memory is sufficient for our workload, enabling us to run ROS2, OpenCV-based vision processing, and AI models simultaneously without major memory bottlenecks. If additional memory bandwidth is required under extreme workloads, we have the option to utilize swap memory on an NVMe SSD via the PCIe interface, providing a potential solution for handling high-resolution image processing and large dataset operations. Overall, the combination of high processing power, fast memory, dual-camera support, and expandable storage options makes the Raspberry Pi 5 a compelling choice for our project, delivering the computational efficiency needed for real-time robotic vision and motor control.

3.5.2 Nvidia Jetson Nano

One of the Single Board Computers (SBCs) we are considering for our project is the NVIDIA Jetson Nano due to its dedicated AI acceleration, powerful GPU, and optimized software stack for machine learning and computer vision applications. The Jetson Nano is powered by a quad-core Arm Cortex-A57 processor clocked at 1.43GHz and an NVIDIA Maxwell GPU with 128 CUDA cores, allowing for parallel execution of AI inference tasks, real-time image processing, and deep learning-based object detection. The 4GB of LPDDR4 memory provides sufficient bandwidth for handling ROS2 processes, computer vision algorithms, and real-time sensor fusion while maintaining low power consumption. While the Jetson Nano has a lower clock speed than other SBCs, its GPU acceleration compensates by enabling faster neural network inference and optimized CUDA-based computations for robotic applications.

For our computer vision system, the Jetson Nano features a single 2-lane MIPI CSI camera interface, which supports real-time video streaming and AI-based object recognition. Although it lacks dual-camera support, it leverages hardware-accelerated image processing using NVIDIA’s Vision Programming Interface (VPI) and TensorRT to efficiently handle deep learning models for object detection and classification. The GPIO and I2C interfaces allow for sensor integration and motor control, while four USB 3.0 ports provide high-

speed data transfer for peripherals such as stereo cameras, LiDAR, or external storage. Additionally, the Gigabit Ethernet port ensures low-latency networking, which is useful for remote operation or distributed robotic control.

Although the Jetson Nano is limited to 4GB of RAM and lacks PCIe NVMe storage support, its GPU acceleration makes it a strong candidate for AI-intensive tasks such as image classification, material identification, and real-time decision-making. The ability to optimize deep learning models using TensorRT and deploy them efficiently on CUDA cores makes it an attractive choice for applications where machine learning inference is critical. Overall, the combination of GPU-accelerated computing, optimized software ecosystem, and high-speed connectivity makes the Jetson Nano a compelling option for our project, particularly if AI-based vision processing is a priority.

3.5.3 Orange Pi

One of the Single Board Computers (SBCs) we are considering for our project is the Orange Pi 5 (8GB) due to its high-performance Rockchip RK3588S processor, versatile I/O capabilities, and cost-effectiveness. The Rockchip RK3588S features an octa-core CPU with four Arm Cortex-A76 cores clocked at 2.4GHz and four Cortex-A55 cores at 1.8GHz, providing a significant increase in multi-threaded performance compared to other SBCs. Additionally, the Mali-G610 MP4 GPU offers GPU acceleration for graphics processing and AI inference, making it capable of handling computer vision and deep learning-based robotic applications. The 8GB of LPDDR4X RAM ensures high memory bandwidth, allowing for efficient multitasking between ROS2, OpenCV-based image processing, and real-time motor control tasks.

For our computer vision system, the Orange Pi 5 provides a dual MIPI-CSI camera interface, supporting multiple high-resolution camera modules for stereo vision or real-time image capture. Unlike the Jetson Nano, which relies heavily on its GPU for AI acceleration, the Orange Pi 5 integrates a Neural Processing Unit (NPU) capable of 6 TOPS (Tera Operations Per Second) of AI performance, allowing for efficient deep learning inference on edge devices. This makes it a strong choice for autonomous robotic applications where AI-powered vision processing is required. The USB 3.0 ports and high-speed GPIOs provide seamless integration with additional sensors, actuators, and external storage devices, while the Gigabit Ethernet ensures low-latency communication for networking applications.

While the Orange Pi 5 lacks native PCIe expansion for NVMe storage, its USB 3.0 interface allows for high-speed external storage options if additional swap memory or dataset storage is required. Its NPU acceleration gives it an advantage over standard CPU-based SBCs, making it well-suited for AI-driven material identification, robotic vision, and autonomous navigation tasks. Overall, the Orange Pi 5 balances high computational power, dedicated AI acceleration, and versatile I/O options, making it a strong alternative for real-

time processing and robotics applications that require machine learning and multi-threaded CPU performance.

3.5.4 SBC Selection

Overall, after evaluating the Raspberry Pi 5 (8GB), NVIDIA Jetson Nano, and Orange Pi 5 (8GB), we have decided to select the Raspberry Pi 5 (8GB) as the most suitable SBC for our project. Based on our research, the Raspberry Pi 5 offers the best balance of community support, ease of use, affordability, and sufficient performance to meet our project requirements.

While the Orange Pi 5 provides greater raw computational capabilities, it lacks consistent performance, strong software support, and affordability, making it less viable for long-term development. Similarly, while the Jetson Nano excels in AI-based tasks and GPU-accelerated computing, it falls short in storage expandability, ease of integration, and cost-effectiveness. In conclusion, the Raspberry Pi 5 delivers the necessary performance while maintaining broad compatibility, strong community support, and cost efficiency, making it the optimal choice for our system.

Feature	Raspberry Pi 5 (8GB)	NIVIDA Jetson Nano	Orange Pi 5 (8GB)
CPU Core	Broadcom BCM2712 Quad-Core Cortex-A76	Quad-Core Cortex-A57	Rockchip RK3588S (4x Cortex-A76 + 4x Cortex-A55)
Clock Speed	2.4 GHz	1.43 GHz	2.4 GHz (A76) / 1.8 GHz (A55)
GPU	VideoCore VII	128-core Maxwell GPU	Mali-G610 MP4
RAM	8GB SDRAM	4GB LPDDR4	8GB LPDDR4X
Storage	MicroSD, PCIe NVMe (optional)	MicroSD	eMMC, NVMe SSD (via M.2 slot) 26
MIPI CSI	Dual 4-lane MIPI CSI	Single 2-lane MIPI CSI	Dual 4-lane MIPI CSI
Networking	Gigabit Ethernet, Wi-Fi 6, Bluetooth 5.0	Gigabit Ethernet (No Wi-Fi or Bluetooth)	Gigabit Ethernet, Wi-Fi (via external module)
AI Acceleration	No (External Module needed)	Yes (CUDA, TensorRT)	Yes (6 TOPS NPU)
Power Consumption	Medium	High	Medium

Availability	High	Low	High
Cost	~$80 (without peripherals)	~$250 (with peripherals)	~$110 (Without peripherals)
Software Support	High (Raspberry Pi OS, Ubuntu, Debian, ROS2)	Medium (JetPack SDK, Ubuntu-based, ROS2)	Medium (Debian, Ubuntu, Android)

Table 14: SBC Selection

3.6 Power Components

In this section, we will research and select the power system components that we need to power our subsystems and peripherals. Ensuring that we have enough current supply, low noise, and high efficiency is important to consider when developing any project. Our project is especially sensitive to noise due to the linear CCD sensor, which requires clean and specific timing signals to get a clean output.

3.6.1 Power Supply

A generic AC/DC switching power supply unit was selected for the primary power source to convert 120VAC wall power to a stable 12VDC output, which then feeds the rest of the power delivery system. The choice to obtain a unit instead of designing one was made due to safety issues when dealing with such high voltages and currents.

The key feature when choosing the power supply is the amount of power or current supplied and the cost. A comparison is shown in Table 15. We chose to go with the option with the maximum current and 12V output.

Feature	**Mean Well LRS-35-12**	**Mean Well LRS-35-15**	**Amazon Universal Switching Power Supply**
Voltage Output	12V	15V	12V
Max Current	3A	2.4	30A
Power	36W	36W	360W
Circuit Protection	Yes	YES	Yes

Table 15: Switching Power Supply Comparison

3.6.2 Voltage Regulator

Linear Voltage Regulators, especially Low Dropout Voltage (LDO) ones, are great options when we need to generate a specific voltage level with very minimal noise. However, they

are not efficient since they dissipate most of the lost energy as heat. For that reason, we chose to use LDOs for regulating output with a small difference between input and output, such as 5V to 3.3V or 4V. We will utilize switching regulators for larger applications, which will be discussed in the next section. Table 16 shows a comparison of multiple options we could use to generate the 3.3V rail required by our logic-level components and sensors.

Feature	AMS1117-3.3	MIC5205-3.3	LD1117-3.3
Voltage Output	3.3V	3.3V	3.3V
Max Current	1A	150mA	800mA
Dropout Voltage	1.1V	0.35V	1.1V

Table 16. 3.3V Linear Voltage Regulator Comparison

The reason we chose the AMS1117-3.3 is it has a sufficient current supply for our logic components, it is widely available and it is a simple implementation with minimal external components. Even though it has a higher dropout voltage, the 5V input from the buck converter ensures stable operation

3.6.3 DC-to-DC Buck Converter

This project requires a high-efficiency step-down converter from 12V to 5V to supply both the linear regulators and any 5V components directly. Buck converters are a great option for such tasks. They consist of switching voltage regulators that use transistor switching to regulate the voltage output, which increases efficiency significantly compared to linear regulators.

Feature	LM2678-5.0	MP1584	XL4015
Voltage Output	5V	Adjustable	Adjustable
Max Current	5A	3A	5A
Efficiency	85%-92%	80%-90%	85%

Table 17. 5V Buck Converter Comparison

The choice made in Table 17 was made due to multiple reasons. It has a High current output, a fixed 5V version that simplifies feedback design, and it is a reliable regulator tested and used in many projects we have seen.

3.7 Motors

The motor selection process is critical to ensuring precise and efficient motion control in our robotic system. Motors must be chosen based on factors such as accuracy, torque, speed, power efficiency, and ease of control. This section discusses different motor technologies, compares their advantages and disadvantages, and then selects the appropriate motor type for our project. We then analyze different motor drivers and encoders to ensure optimal control and feedback integration.

Different motor types are used in robotics and automation systems, each with unique characteristics that make them suitable for different applications. The primary motor technologies considered for our project are servo motors, stepper motors, and brushless DC (BLDC) motors.

3.7.1 Servo Motors

Servo motors, particularly the inexpensive RC hobby servos, are commonly used in robotic applications where position control is necessary. These motors consist of a DC motor, a gear train, a potentiometer for position feedback, and an internal controller. They typically operate on pulse-width modulation (PWM) signals, where the pulse width determines the target position of the motor shaft.

Hobby servos, such as the SG90 and MG995, are low-cost and simple to use. They provide reasonable position accuracy, making them suitable for lightweight robotic arms, pan-tilt camera mounts, and simple automation projects. However, they have limitations in torque output, continuous rotation capability, and precision when compared to industrial-grade servo motors. Additionally, they lack direct torque feedback, making them less suitable for high-load applications.

3.7.2 Stepper

Stepper motors are widely used in robotics and automation due to their precise positioning capabilities. They operate by moving in discrete steps, meaning they do not require an internal feedback system for position control. Stepper motors can be bipolar or unipolar, with bipolar motors being the more common choice due to their higher torque efficiency.

One major advantage of stepper motors is their ability to hold a position without requiring continuous power input, making them ideal for applications where stability is essential. However, stepper motors can suffer from missed steps under high loads unless an external feedback system, such as an encoder, is used. They also tend to consume power continuously, even when idle, which can generate excess heat.

Stepper motors are commonly available in standard sizes, such as NEMA 17, NEMA 23, and NEMA 34, each offering varying levels of torque and power requirements. Their ease of use, precise step control, and availability make them a strong candidate for our project.

3.7.3 Brushless DC

Brushless DC motors are highly efficient and offer smooth motion control. Unlike brushed motors, BLDC motors use electronic commutation, eliminating the need for physical brushes, which results in a longer lifespan and reduced maintenance.

BLDC motors are commonly used in high-speed applications, such as drones, electric vehicles, and industrial automation. They provide superior efficiency and torque output compared to stepper motors and hobby servos. However, they require specialized motor controllers, such as ESCs (Electronic Speed Controllers)**,** to operate correctly. Additionally, BLDC motors typically rely on Hall-effect sensors or encoders for position feedback, adding to system complexity and cost.

While BLDC motors are excellent for high-speed and efficiency-focused applications, they are not the best fit for our robotic arm, which requires precise, incremental motion rather than continuous high-speed operation.

3.7.4 Motor Selection

After evaluating the different motor technologies as shown in Table 18, we have chosen stepper motors for our main joints due to their high precision, range of motion, and wide variety of holding torque options available at a low cost. Stepper motors provide an ideal balance between affordability and performance for robotic applications.

For the end-effector joints, we decided to use servo motors since they are cheaper, smaller, and lighter. Additionally, we do not need high torque or continuous rotation at these joints, so servo motors would be sufficient.

Features	Servo	Stepper	BLDC
Position Control	Closed Loop	Open Loop	Open Loop
Cost	~$2 - $10	~$10 - $50	~$80 - $200
Durability	~15000 cycles	~10000 hours	~20000 hours
Range of Motion	180°-270°	360°	360°
Precision	1°	1.8° or 0.9°/step	Controller dependent

		(micro stepping available)	
Holding Torque	~0.5 - 3 Nm	~0.1 - 8 Nm	~0.1 - 100 Nm
Driver	PWM Signal	Stepper Driver (e.g., A4988)	ESC (Electronic Speed Controller)

Table 18: Motor Technology Comparison

For the servo motors, we decided to go with the MG996R since it provides the most torque, precision, and durability among the three options shown in Table 19.

Features	**SG90**	**MG90s**	**MG996R**
Position Control	Closed Loop	Closed Loop	Closed Loop
Cost	~$2–3	~$4–5	~$7–10
Durability	~100,000 cycles	~150,000 cycles	~200,000 cycles (metal gear)
Precision	±10° (approx.)	±8° (approx.)	±5° (approx.)
Holding Torque	1.8 kg·cm @ 4.8V	~2.2 kg·cm @ 4.8V	~9–11 kg·cm @ 6V

Table 19: Servo Motor Selection: End Effector

Stepper Motors come in a variety of standard sizes, such as NEMA 14, 17,23,34, etc. Each standard size has a range of holding torque it can handle,, which depends on the length of the motor used. For our project, we chose to go with the NEMA 17 as shown in Table 20, it provides a good balance of cost and size to torque. Although NEMA 23 motors have greater holding torque, they are significantly larger and can be more expensive.

Features	**Nema 14**	**Nema 17**	**Nema 23**
Step count	200 steps/rev (1.8°/step)	200 steps/rev (1.8°/step)	200 steps/rev (1.8°/step)
current	~0.4 - 1.0 A/phase	~1.2 - 2.0 A/phase	~2.0 - 3.5 A/phase
torque	~0.08 - 0.12 N·m	~0.12 - 0.8 N·m	~1.2 - 4.0 N·m
Phases	2	2	2

Table 20: Stepper Motor Selection: Main Joints

3.8 Motor Encoders

Encoders play a crucial role in motion control by providing feedback on position, speed, and direction. Encoders are necessary in our robotic system to ensure accurate stepper motor positioning and prevent missed steps. This section explores different encoder types, compares their advantages and disadvantages, and selects the best option for our application.

3.8.1 Optical Encoders

Optical encoders use a disk with alternating transparent and opaque sections, along with a light source and a sensor, to detect motion and position changes. When the disk rotates, the light passes through or is blocked, generating a signal that can be processed to determine the position and speed of the motor. Optical encoders are known for their high accuracy and resolution, making them suitable for applications requiring precise motion control. They provide fast response times and can deliver both absolute and incremental position feedback. However, they are highly sensitive to dust and dirt, which can obstruct the sensor and affect performance. Additionally, optical encoders tend to be more expensive than other types and require careful alignment during installation to function correctly.

3.8.2 Magnetic Encoders

Magnetic encoders use a magnetized disk and a magnetic sensor to detect changes in position. The sensor measures variations in the magnetic field to determine motion and position. Unlike optical encoders, magnetic encoders are more resistant to environmental interference and can function reliably in dusty or vibrating conditions. They are compact and durable, making them suitable for industrial applications where robustness is a priority. While magnetic encoders offer better resistance to harsh conditions, they have slightly lower resolution compared to optical encoders and may introduce small errors due to magnetic interference. Shielding is required to prevent external magnetic fields from affecting their accuracy.

3.8.3 Potentiometers

Potentiometers are simple analog position sensors that measure resistance changes as a shaft rotates. They provide a continuous voltage output corresponding to the shaft position, making them suitable for low-precision applications. Their affordability and simplicity make them widely available and easy to integrate into basic motion control systems. However, potentiometers suffer from mechanical wear over time due to their contact-based design, which reduces their lifespan and reliability in continuous operation. They also have limited rotation range, typically restricted to less than 360 degrees, and do not offer the precision required for high-accuracy applications.

3.8.4 Encoder Selection

For our robotic system, we require an encoder that balances accuracy, durability, and cost-effectiveness. Given the need for robust operation in varying conditions and reasonable precision for stepper motor feedback, we select magnetic encoders. These provide reliable feedback while maintaining durability in an environment where dust and vibrations are present. Optical encoders, while highly accurate, are too sensitive to environmental factors and require careful installation. Potentiometers, while inexpensive, do not offer the resolution or reliability needed for precise motion control.

Feature	Potentiometer	Magnetic	Optical
Accuracy	Around ±3–5% of full range	Around ±1–3%	Can be better than ±1%
Resolution	Limited by ADC (e.g., 10-bit)	12–14-bit digital or analog	Very fine (e.g., 1024+ steps)
Durability	Wears over time (mechanical contact)	Long-lasting (no contact)	Durable, but can be affected by dust
Cost	<$2	~$5 - ~$10	~$10 - ~$100
Precision	Moderate, affected by noise	Decent with good calibration	Very precise for position tracking

Table 21: Stepper Motor Encoder Selection

The comparison in Table 21 was generated using ChatGPT, and the prompt used can be found in *Appendix E-2*. After reviewing relevant datasheets to verify the information, we chose to use magnetic encoders for our stepper motors due to their balance of accuracy, durability, and ease of integration. For servo motors, the use of potentiometer-based encoders is built-in and not optional, so we continued using them as provided.

3.9 Motor Drivers

In this section we compare and select the stepper motor drivers as well as the servo motor drivers.

3.9.1 Stepper Driver Selection

When deciding which stepper motor driver to use for our project, we looked at three main options: the A4988, TMC2209, and MKS Servo42C. Each one has its own pros and cons depending on what you need. The A4988 is a basic and cheap driver that works fine for simple open-loop control. The TMC2209 adds some extra features like quieter movement and higher microstepping, but it’s still open-loop unless you use some stall detection. The

MKS Servo42C, on the other hand, is a full closed-loop driver that mounts right on the motor and includes an encoder for feedback.

Here's a comparison of the three drivers and what they offer:

Feature	A4988	TMC2209	MKS Servo42C
Control Type	Open-loop	Open-loop	Closed-loop with encoder
Feedback Support	None	None	Built-in encoder
Interface	Step/Dir	Step/Dir	UART, FOC, Step/Dir
Mounting	External to motor	External to motor	On motor
Size	0.6" x 0.8"	0.6" x 0.8"	1.65" x 1.65"
Features	Basic stepper control	Quiet stepping, stealthChop	PID control, auto-tuning, feedback
Cost	~$1	~$3	~$15

Table 22: Stepper Driver Comparison

In the end, we picked the MKS Servo42C. The main reason is that it makes wiring and setup easier since it goes right on the stepper motor. It also gives us encoder feedback and closed-loop control using PID, which helps with accuracy and reliability. On top of that, it uses UART to send position commands, which fits perfectly with the rest of our control system.

3.9.2 Servo Driver Selection

Servo motors can be controlled directly using PWM signals from a microcontroller, so technically we don't need a separate driver. But we chose to use the PCA9685 I2C PWM driver to keep the wiring cleaner and avoid using too many CPU resources.

3.10 Speed Reducer

In order to limit the amount of spending on high-torque motors, it makes more logistical sense to use speed reducers to increase the amount of torque produced by the motors. Therefore, it is important that we select gearboxes that create the most torque reliably. Additionally, unlike motors related to the end effector and spherical wrist, these motors control the linkages of the robot arm and, therefore, can be integrated further from the joint, reducing weight on the robot arm.

		Concept Alternatives							
		Strain Wave Gear		Cycloidal Drive		Capstan Drive		Planetary Gears	
Criteria	Importance Weight %	Rating	Weighted Rating	Rating	Weighted Rating	Rating	Weighted Rating	Rating	Weighted Rating
High Torque Ratio	40%	4	1.60	4	1.60	3	1.20	3	1.20
High Reliabilty	35%	3	1.05	4	1.40	3	1.05	4	1.40
Low Backlash	10%	4	0.40	4	0.40	4	0.40	2	0.20
Low Cost	10%	4	0.40	3	0.30	3	0.30	4	0.40
Small Footprint	5%	3	0.15	3	0.15	3	0.15	4	0.20
Totals	100%		3.60		3.85		3.10		3.40

Rating	Value
Unsatisfactory	0
Just Tolerable	1
Adequate	2
Good	3
Very Good	4

1. Importance weights assigned
2. Team rates each concept 0 to 4
3. Multiply rating X importance
4. Sum weighted ratings

Table 23: Concept Alternatives – Speed Reducer

3.10.1 Strain Wave Gear

Strain Wave Gears (SWGs) are in high demand in the industrial robotics industry for their precise control of angular positioning and high-power transmission. Additionally, SWGs are compact in size, have a high reduction ratio, and have virtually zero backlash [11], making them perfect for robotics applications. SWGs utilize a unique operating principle based on the mechanics of materials. It is comprised of three parts: the Wave Generator, Flexspline, and Circular Spline [12].

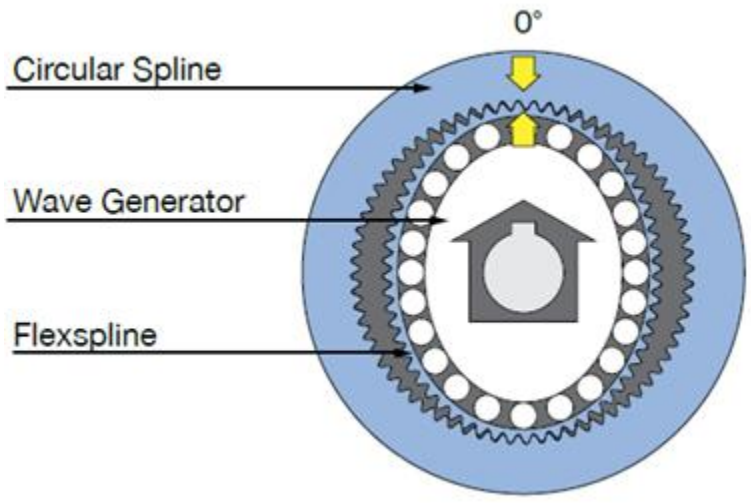


Figure 7: Strain Wave Gear Components [13]

The issue with using an SWG is that they rely on the mechanical properties of the material in order to operate smoothly. For the purposes of our project, our speed reducers will be made using FDM manufacturing. As discussed in *3.2.4,* PETG is unreliable due to its

inconsistency in manufacturing. Therefore, we would not be able to manufacture our own SWG and would have to spend an unrealistic amount of our budget to acquire a speed reducer.

3.10.2 Cycloidal Drive

Very similar to the Strain Wave Gear, the Cycloidal Drive is a fan favorite in the robotics industry for its compact design, reduction ratios of up to ten times that of planetary or gear drives, and virtually zero-backlash [14]. The cycloidal drive is composed of a high-speed input shaft, an eccentric bearing, two cycloidal disks, a ring gear, rollers, and a slow-speed output shaft.

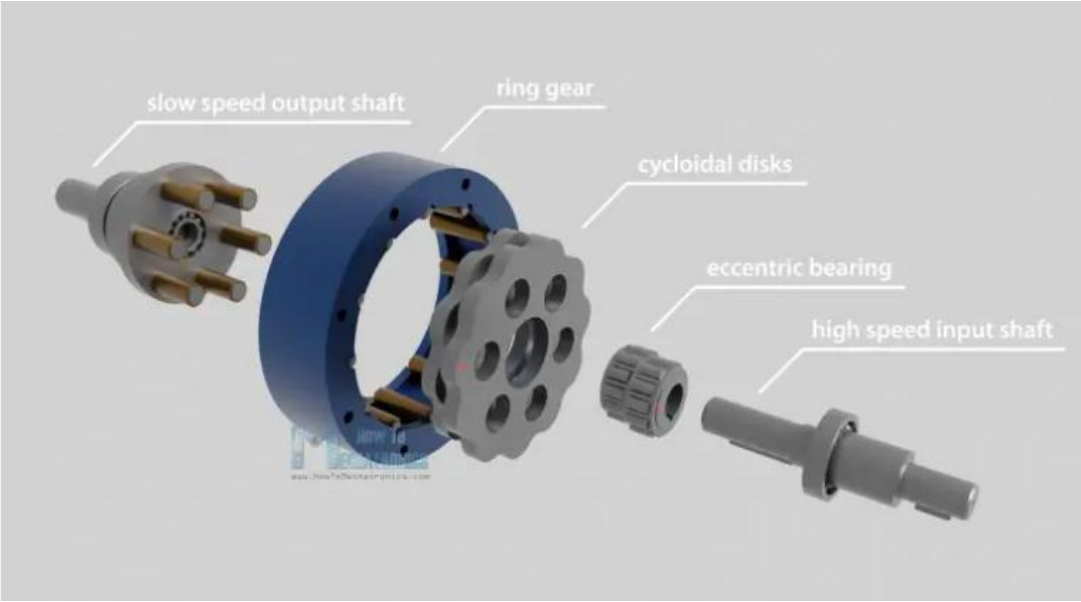


Figure 8: Cycloidal Drive with main component labels [15]

Unlike the strain wave gear, the cycloidal drive does not rely on the mechanics of material, but rather on the motion of the cycloidal disks driving the output shaft. This design has an increase of hardware components, which will raise the cost. However, it is far more reliable than the strain wave gear.

3.10.3 Capstan Drive

A Capstan drive is a quasi-direct-drive (QDD) speed reducer that utilizes cables wrapped in round drums commonly found in medieval machines. The Capstan cable drive is low weight/inertia, has a simple design, and has a high strength-to-weight ratio with haptic feedback, making it perfect for robotics applications [16].

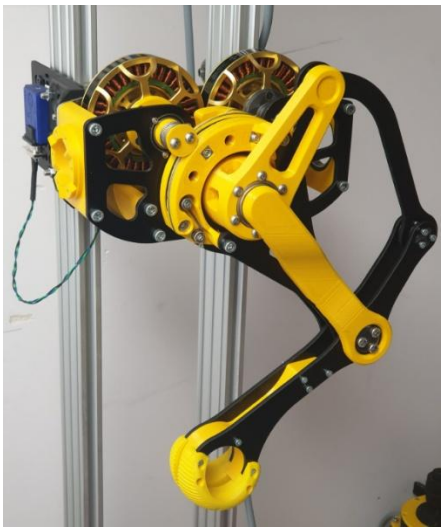


Figure 9: Robot Quadruped Leg Using Capstan Drive [17]

Capstan drives have high torque ratios and are reliable dependent on the cord used to link the two drums together. They also have a reasonably small footprint and weight for the amount of work that the speed reducer is achieving. However, the main limiting factor of a capstan drive is the reduced range of motion. This will further complicate the engineering requirements for the robot arm and would not be an applicable choice for our case.

3.10.4 Planetary Gears

Planetary gears are one of the most common forms of speed reducers. Composing of an outer ring gear, planet gears, and a central sun gear, planetary gears can be found in most automobiles for their compactness and high torque ratios.

For our design, our planetary gears would require little to no hardware making it low-cost and lightweight. However, the torque-to-footprint ratio is much lower than other options, including harmonic and cycloidal drives. Due to the amount of friction and lack of efficiency, it is also much more challenging to make planetary gears back drivable.

3.11 Camera

For the T.R.A.S.H. system to achieve high-accuracy object identification and precise localization, the selection of an appropriate camera is critical. The camera serves as the primary input for the computer vision pipeline, and its capabilities directly affect the system's accuracy, responsiveness, and integration flexibility. Several core requirements were established to ensure compatibility with the processing platform and to align with the project's goals of modularity, affordability, and performance.

The first fundamental requirement is seamless integration with the Raspberry Pi 5 through the native Camera Serial Interface (CSI) port. Using the CSI port allows direct hardware-level communication between the camera and the Pi without occupying USB bandwidth. This not only frees up USB ports for other essential peripherals but also minimizes latency and allows for compact, efficient hardware layouts on the limited space on the Raspberry Pi board. Native integration ensures that the imaging system can be fully synchronized with image processing and actuation routines, providing real-time responsiveness for sorting operations. Within our selection of cameras, they will all be supporting CSI port,

In addition to hardware compatibility, the camera must support High Dynamic Range (HDR) imaging. HDR enables the camera to maintain visual clarity across bright and dark regions of the scene. This is especially important when capturing the contours and surfaces of reflective or translucent recyclables like plastic bottles, aluminum cans, or coated paper. With HDR enabled, computer vision algorithms can better distinguish between edges, colors, and material transitions, improving both detection accuracy and classification robustness.

Cost is another driving factor in camera selection. One of the core principles of the T.R.A.S.H. system is affordable modularity. The solution must be accessible and easy to replicate using commercial off-the-shelf (COTS) components. Therefore, while advanced imaging capabilities are desired, the selected camera must remain within a reasonable price range to ensure that the system can be produced and scaled cost-effectively. Cameras priced up to $100 were considered acceptable based on the overall budget allocated to sensors.

Image resolution is also critical for detailed object analysis. Higher resolution allows the system to identify smaller or partially obstructed objects within the field of view. It improves the system’s ability to track shapes and contours accurately, especially during post-processing techniques like segmentation or object recognition. However, a balance must be maintained between resolution and processing requirements. Excessively high resolutions can introduce unnecessary computational overhead and reduce real-time performance unless paired with aggressive optimization and hardware acceleration.

An often overlooked but important consideration is lens distortion. Lower distortion is preferred, as it reduces the need for camera calibration and image correction, streamlining the vision pipeline. However, distortion can occasionally provide functional benefits. A slightly distorted lens, for instance, may offer a wider Field of View (FOV), which could reduce the number of scanning motions required by the robotic arm to cover the workspace. In such cases, distortion and FOV exist in a trade-off: a higher FOV may lead to faster object coverage, while lower distortion simplifies positional accuracy.

The system also benefits from higher frame rates whenever possible. Increased frames per second (FPS) allows the T.R.A.S.H. robot to process more data in real time, enabling better object tracking, faster updates to position estimates, and more accurate sorting decisions. A camera capable of capturing 30 FPS or higher provides sufficient resolution for the Pi's processing capabilities and the arm's pick-and-place requirements.

Equally important is the camera's performance under low-light conditions. In realistic environments, lighting may not always be consistent or controllable. Materials such as aluminum and transparent plastics can reflect or refract light unpredictably, causing sections of the workspace to fall into shadow or become overexposed. A camera with strong low-light sensitivity ensures that object features remain visible and detectable even when ambient light is reduced. This allows the system to operate reliably across different installation scenarios, including dimly lit rooms, shaded workspaces, or enclosed environments. Cameras that exhibit high sensitivity and low noise under minimal lighting reduce the burden on artificial lighting systems and improve the system's resilience to environmental variability.

Lastly, library compatibility and ease of software integration play a significant role. The camera must be supported by popular, well-maintained Python and C++ libraries like OpenCV, libcamera, or Picamera2. This ensures that real-time image capture, frame processing, and feature extraction can be easily integrated into the existing codebase without low-level driver development. Library support is crucial for maintaining development velocity, debugging, and enabling future upgrades to the system.

To evaluate potential imaging solutions, six camera modules were selected that meet the outlined technical requirements and interface constraints: the Raspberry Pi Camera Module V2, Raspberry Pi Camera Module V3, Raspberry Pi High Quality (HQ) Camera, and three Arducam models—the Arducam 12MP, Arducam 16MP, and Arducam 64MP. These cameras represent a broad range of capabilities from 8MP to 64MP, including modules with autofocus, HDR, and varying levels of low-light sensitivity. All selected cameras are natively compatible with the Raspberry Pi 5 CSI interface, ensuring direct integration into the system architecture. The Raspberry Pi V3 and Arducam 12MP are nearly identical in specifications and will be evaluated together in a shared subsection.

3.11.1 Raspberry Pi Camera Module V2

The Raspberry Pi Camera Module V2 is one of the most established and widely used camera options for Raspberry Pi platforms. Priced at approximately $15, it offers a highly affordable solution for basic imaging applications and benefits from extensive community support due to its long-standing availability since April 2016. Its low cost and widespread availability make it particularly well-suited for COTS integration, allowing for simple repairs, replacements, and future upgrades. This makes it a great choice for modular systems such as T.R.A.S.H., where long-term maintainability and ease of part sourcing are critical design goals.

This camera is built around the Sony IMX219 CMOS sensor, which is an 8-megapixel image sensor paired with a fixed-focus lens. In terms of performance, it is capable of capturing video at 1080p60, 720p180, and even 4Kp30 frame rates. However, due to the Raspberry Pi's internal processing limitations, the maximum usable capture resolution on most systems is 1080p30, which remains sufficient for standard computer vision tasks such as object detection and tracking.

The V2 camera does not support HDR, which can limit its effectiveness in high-contrast environments or in scenes with variable lighting conditions. Additionally, its performance in low-light environments is limited, making it less ideal for applications where ambient illumination may be inconsistent or reduced. Under well-lit conditions, however, it provides predictable image clarity and color fidelity for general-purpose use.

Another advantage of the Camera Module V2 is its broad compatibility with Raspbian and mature library support, including seamless integration with tools such as Picamera, libcamera, and OpenCV. Its long-standing presence in the Raspberry Pi ecosystem ensures reliable documentation, stable drivers, and ease of troubleshooting—factors that continue to make it a dependable baseline solution for developers prioritizing cost, simplicity, and compatibility over advanced imaging features.

3.11.2 Raspberry Pi Camera Module V3/ArduCam 12MP

The Raspberry Pi Camera Module V3, released in January 2023, and the Arducam 12MP, introduced shortly after as a third-party equivalent, represent the next step up in imaging performance for Raspberry Pi platforms, offering enhanced features such as autofocus and High Dynamic Range (HDR) at an accessible price point. Both modules are priced at approximately $25 and utilize the Sony IMX708 CMOS sensor, delivering 12-megapixel resolution with a significant improvement in image sharpness and adaptability compared to earlier generation models. Given their near-identical sensor architecture and operational specifications, these two cameras are evaluated jointly in this subsection.

Both the V3 and the Arducam 12MP support autofocus functionality, allowing them to dynamically adjust focus for objects at varying distances. This feature is particularly valuable in scenarios involving cluttered workspaces or variable object heights, as it improves overall clarity and ensures consistent edge definition for computer vision tasks. With a Minimum Object Distance (MOD) of 5 cm, these cameras are well-suited for close-range applications such as material classification and detailed feature detection.

In terms of imaging performance, both modules support video capture up to 1080p at 50 frames per second, which falls just under the Raspberry Pi 5’s hardware decoding limit of 1080p60, making full use of the available processing capabilities. The inclusion of HDR

support—paired with enhanced low-light sensitivity—significantly improves their ability to handle scenes with strong lighting contrasts, dark shadows, or reflective materials. These attributes are especially useful for identifying objects under variable ambient conditions and maintaining reliable detection performance across different environments. With a 66-degree horizontal Field of View (FOV) in the base model, these cameras provide a balanced frame coverage that reduces the number of repositioning movements required from the robotic arm while maintaining sufficient pixel density for accurate object segmentation.

Crucially, both the Raspberry Pi V3 and the Arducam 12MP are fully compatible with Raspbian OS, ensuring seamless integration with the rest of the T.R.A.S.H. system. Supported libraries such as libcamera, Picamera2, and OpenCV offer a stable software interface for development and testing, reducing the effort needed for camera control, frame acquisition, and real-time processing. The V3 module, being an official Raspberry Pi product, benefits from robust community and developer support. The Arducam 12MP, while third-party, closely mirrors the V3 in design and performance, offering a near drop-in alternative.

In summary, both the Raspberry Pi Camera V3 and Arducam 12MP present a highly capable imaging solution with modern features like autofocus, HDR, and strong low-light performance, without departing from the project's commitment to affordability and compatibility. Their shared architecture and reliable operation make them strong candidates for mid-tier vision requirements within the T.R.A.S.H. platform.

3.11.3 Raspberry Pi Camera High Quality

The Raspberry Pi High Quality (HQ) Camera, released in April 2020, is a premium imaging module designed for developers and applications that demand advanced optical performance and greater flexibility in lens selection. Priced at approximately $50, the HQ Camera offers a significant step up in customization and image fidelity compared to standard camera modules, making it a strong candidate for controlled environments where precision and clarity are prioritized.

The module features a Sony IMX477 CMOS sensor with a 12.3-megapixel resolution and a 7.9 mm diagonal sensor size, offering a larger imaging area compared to standard modules. It is paired with a C/CS-mount lens system and also supports M12 lenses through widely available adapters, enabling compatibility with a broad range of optical configurations. This allows the camera's field of view, focus range, and aperture to be tailored to specific use cases—an important advantage for applications requiring tightly framed imaging, high-zoom inspection, or adjustable focal lengths. The flexibility in optics also opens the door to specialized setups involving filters, varifocal lenses, or manual iris control.

While the HQ Camera does not include HDR support, and its low-light performance is not as optimized out of the box compared to sensors with onboard HDR and larger pixel sizes, its larger 1.55µm pixel pitch does provide inherently better light sensitivity and signal-to-noise ratio than smaller, lower-cost modules. In well-lit or optically controlled setups, the image clarity and depth of field it provides are unmatched among Raspberry Pi-compatible camera modules. However, for environments with limited ambient lighting or high dynamic range needs, the absence of native HDR can be a limiting factor without careful lighting control or external enhancements.

The camera can deliver high-quality video capture at up to 1080p30, and while the sensor itself supports higher resolutions (including 4K), the Raspberry Pi 5's real-time processing capabilities are best utilized at 1080p, particularly when balancing frame rate with computational demands for image processing and classification tasks.

The HQ Camera is fully supported under Raspbian OS, and integrates smoothly with major libraries including libcamera, Picamera2, and OpenCV. As an official Raspberry Pi accessory, it benefits from extensive documentation, stable firmware support, and broad community engagement. Its modular lens system is also natively supported through the Raspberry Pi ecosystem, with mounting brackets and accessories readily available.

In summary, the Raspberry Pi HQ Camera offers exceptional optical quality, lens flexibility, and system-level compatibility. While it lacks HDR and enhanced low-light modes found in some newer compact modules, it excels in applications that prioritize resolution, depth of field, and controllable imaging conditions. For the T.R.A.S.H. system, it is best suited for use cases where custom optics and high clarity are required, particularly in well-lit environments with minimal spatial constraints.

3.11.4 ArduCam 16MP

The Arducam 16MP Autofocus Camera, released in Jan 2022, offers a balance of high-resolution imaging and modern autofocus features at a competitive price point of approximately $25. Built around the Sony IMX519 CMOS sensor, this camera delivers a resolution that is double that of the Raspberry Pi Camera Module V2 and approximately 40% higher than the Raspberry Pi High Quality Camera. This increase in pixel density makes it especially appealing for applications requiring fine detail, such as precision sorting, inspection, and monitoring. Arducam has positioned this series of cameras to support a wide range of industrial and consumer-grade applications, making them versatile tools for embedded systems and smart vision projects.

Although it is not an official Raspberry Pi Foundation product, this module is engineered to be compatible with recent Raspberry Pi software frameworks and is tuned using similar algorithms found in the official camera modules. This results in improved image quality and consistency while maintaining integration with standard pipelines.

The camera supports autofocus functionality, allowing it to adjust dynamically to subjects at different depths, with a MOD of 10 cm. This capability is especially useful in robotic environments where object distances can vary unexpectedly. The module supports video capture at 1080p30, which aligns with the real-time processing capabilities of the Raspberry Pi 5 and is suitable for most visual classification and sorting tasks.

Despite its high resolution, the Arducam 16MP does not include HDR support, which affects its ability to capture detail in scenes with strong contrast between light and dark regions. Consequently, its performance in low-light environments is also limited, making it better suited for consistent, well-lit settings or systems equipped with supplemental lighting.

A significant drawback to note is that this module does not natively support the Raspbian OS, which introduces additional configuration overhead for developers aiming to use it in a standard Raspberry Pi environment. Alternative driver support and adjusted Linux distributions are often required for full functionality. Similarly, while software libraries are available—typically through Arducam's own SDK—some customization may be necessary for integration with libcamera and OpenCV workflows.

In summary, the Arducam 16MP camera presents an affordable, high-resolution imaging option with autofocus and flexible usage across advanced applications. However, its limited HDR and low-light support, combined with the lack of native Raspbian compatibility, means it is best suited for controlled environments where resolution is the primary requirement and platform flexibility is less critical.

3.11.5 ArduCam 64MP

The Arducam 64MP Autofocus Camera, introduced in late 2022, represents the highest-resolution imaging solution currently available for Raspberry Pi platforms. Priced at approximately $60, it brings flagship smartphone camera performance to embedded systems, making it ideal for applications where ultra-high-definition imaging and precision matter most.

This module features a 64-megapixel sensor, capable of capturing still images at an impressive 9152 x 6944 resolution, placing it in a class well beyond typical embedded camera modules. The integration of autofocus, combined with enhanced software features such as electronic pan-tilt-zoom (ePTZ) and continuous autofocus, expands its capabilities in dynamic vision systems. Since September 2023, support for Phase Detection Autofocus (PDAF) has further improved its real-time focusing speed and accuracy—beneficial in fast-moving environments where subject distance may change rapidly.

Designed for the latest generations of Raspberry Pi boards, the Arducam 64MP camera leverages a modern image sensor architecture derived from premium mobile devices. This makes it particularly suitable for projects requiring DSLR-like clarity without the cost, size, or complexity of standalone imaging systems. The sensor's resolution enables detailed material classification, texture recognition, and wide area scanning with minimal need for mechanical repositioning.

In terms of hardware integration, the module retains compatibility with the v1/v2 form-factor, MIPI CSI-2 interface, and standard tripod mounting systems, ensuring it fits seamlessly into existing Raspberry Pi setups. It is also fully compatible with the latest libcamera software stack, making it easier for developers to transition without rewriting camera pipelines or rebuilding workflows. Despite its third-party origin, the camera is designed to align closely with Raspberry Pi's ecosystem to provide a smooth development experience for advanced users.

However, it is important to note that while the camera offers cutting-edge resolution, it does not include native HDR support, which can limit its ability to balance exposure in high-contrast scenes. Similarly, its low-light performance, while enhanced by resolution and optics, may not fully compensate for the absence of dedicated dynamic range enhancement. For best results, this camera performs optimally in well-lit environments or with the assistance of external lighting systems.

Library support includes full integration with libcamera, and with some adjustments, it works well alongside OpenCV for advanced image processing. As a newer module, community support is still growing but actively maintained by Arducam.

In summary, the Arducam 64MP is an exceptionally powerful imaging tool that brings ultra-high resolution and autofocus features to the Raspberry Pi platform. While its lack of HDR and limitations in low-light sensitivity must be considered, its clarity, modern interface compatibility, and support for advanced focus modes make it a leading choice for vision applications where detail and precision are paramount.

3.11.6 Camera Selection

After evaluating six Raspberry Pi-compatible camera modules across multiple technical dimensions—including resolution, field of view, low-light capability, HDR support, and software compatibility—the Raspberry Pi Camera Module V3 and the Arducam 12MP stand out as the most balanced and effective choices for the T.R.A.S.H. platform.

Both modules utilize the Sony IMX708 12MP CMOS sensor, support HDR, and offer autofocus with a Minimum Object Distance of 5 cm, making them well-suited for close-

range inspection and object classification tasks. Their 66° field of view is wide enough to reduce mechanical repositioning while still maintaining clarity, and they perform well in low-light conditions, which is vital for consistent performance in indoor and varied lighting environments.

Importantly, both cameras offer full Raspbian OS support, ensuring seamless integration with the Raspberry Pi 5 and compatibility with widely used libraries like libcamera, Picamera2, and OpenCV. Their $25 price point also aligns well with the project's focus on affordability and COTS-based modularity.

Given these strengths, the Raspberry Pi Camera Module V3 is the primary choice for T.R.A.S.H. due to its official support, wide availability, and superior developer documentation. However, should the V3 become unavailable due to its high demand and frequent stock shortages, the Arducam 12MP will serve as the direct alternative. With nearly identical hardware and software support, it ensures uninterrupted development without compromising system performance or compatibility.

Feature	RPi Cam V2	RPi Cam V3 and ArduCam 12MP	RPi HQ	ArduCam 16MP	ArduCam 64MP
Cost	$15	$25	$50	$25	$60
MP Count	8MP	12MP	12.3MP	16MP	64MP
FOV	62.2	66	Lens Dependent	66	Lens Dependent
MOD (cm)	5	5	Lens Dependent	10	Lens Dependent
Performance	1080p30	1080p50	1080p30	1080p30	1080p30
HDR	No	Yes	No	No	No
RaspBian	Yes	Yes	Yes	No	Yes
Support	Excellent	Excellent	Excellent	Moderate	Good

Table 24: Camera Comparison

3.12 ROS & ROS2

Robot Operating System (ROS) is an open-source framework designed to facilitate the development and integration of robotic systems. It provides a structured way for multiple components, such as sensors, actuators, and control systems, to communicate efficiently. Rather than developing custom communication protocols for every hardware and software module, ROS allows developers to utilize pre-built libraries, messaging systems, and well-established communication methods. This significantly reduces development complexity while ensuring system flexibility and scalability.

For our project, ROS is a critical component in managing communication between different hardware and software subsystems. Our robotic system involves multiple interconnected components, including a vision system for object detection, motion planning algorithms, and motor controllers for actuation. ROS provides the necessary framework to integrate these elements into a unified system. By using ROS, we can break down the project into modular nodes, each responsible for a specific task, improving maintainability and system stability.

Additionally, ROS has a large and active community, which provides extensive documentation, troubleshooting support, and a vast collection of pre-existing packages. This strong support network accelerates development and ensures that we can incorporate industry-tested solutions rather than developing every component from scratch.

3.12.1 Robot Operating System 1 (ROS)

ROS has been the dominant framework for robotic applications for over a decade. It has powered a wide range of robotic systems, from research projects to industrial automation. One of its key advantages is its modularity, allowing developers to create independent software packages (nodes) that communicate using a publish-subscribe model. This enables seamless integration of different components without requiring direct dependencies between them.

Despite its strengths, ROS has some fundamental limitations. One of its primary drawbacks is that it relies on a centralized master node for communication between components. This creates a single point of failure, if the ROS Master crashes, the entire system loses communication. Furthermore, ROS lacks real-time processing support, making it less suitable for applications requiring precise timing, such as robotic motion control and high-speed sensor data processing.

Another limitation of ROS is its security. Since it was primarily designed for research and prototyping, security features such as encrypted communication and authentication were not a priority. This makes ROS less viable for commercial and industrial applications where data security and reliability are critical.

Additionally, ROS was developed primarily for Linux-based systems, limiting its cross-platform compatibility. Although some workarounds exist to run ROS on Windows, they are not officially supported or optimized for real-world applications.

Due to these limitations, ROS reached its end of life on May 31st, 2025, and is no longer actively maintained. While it still has a large user base, its long-term viability has diminished, making ROS2 the preferred option for modern robotic development.

3.12.2 Robot Operating System 2 (ROS2)

ROS2 was designed as the next-generation replacement for ROS, addressing many of its predecessor's limitations while introducing new features that make it more robust and scalable. One of the most significant improvements in ROS2 is the transition from a centralized ROS Master to a decentralized communication architecture based on the Data Distribution Service (DDS). Unlike ROS, where all nodes had to communicate through a master node, ROS2 allows for direct peer-to-peer communication, eliminating single points of failure and enabling more scalable multi-robot systems.

Another major advantage of ROS2 is its real-time processing capabilities. It includes improved scheduling and execution models, making it better suited for applications that require precise timing, such as robotic motion control and sensor fusion. This makes ROS2 a superior choice for real-time robotics applications compared to ROS.

Security has also been significantly improved in ROS2. Unlike ROS, which lacked encryption and authentication mechanisms, ROS2 includes built-in security features, such as secure node-to-node communication and access control policies. This makes ROS2 more suitable for commercial and industrial applications where data security and integrity are crucial.

Additionally, ROS2 introduces cross-platform support, allowing it to run not only on Linux but also on Windows and macOS. This provides greater flexibility for integrating ROS2 into a wider range of embedded systems and industrial environments.

3.12.3 ROS Selection

By adopting ROS2 for our project, we gain access to better performance, reliability, and future-proof software support. Its real-time capabilities, improved communication framework, and security features make it the ideal choice for building a robust and scalable robotic system.

Feature	ROS	ROS2
End Of Life	May 31st, 2025	Actively Supported and being Developed
Middleware	Centralized ROS Master	Decentralized DDS (Data Distribution Service)
Real-Time Performance	Not designed for real-time applications	Improved execution timing and scheduling
Multi-Robot & Distributed Systems	Limited support	Built-in support for multi-robot and distributed applications

Security	Minimal security features	Encrypted communication and authentication
Cross-Platform Support	Primarily Linux-based	Supports Linux, Windows, and macOS
Community & Package Availability	Large community with many legacy packages	Growing community with most major ROS1 packages ported over

Table 25: Ros Comparison

3.13 Spectroscopy Methods

Spectroscopic techniques serve as powerful analytical tools for material identification and characterization, each leveraging distinct principles of light-matter interactions. Reflection-based spectroscopy, transmission-based spectroscopy, Laser-Induced Breakdown Spectroscopy (LIBS), and Raman spectroscopy represent prominent methods, each uniquely suited to different types of material analysis. As an optical engineer, my goal is to select a spectroscopic system that is straightforward to construct, safe to operate, and sufficiently precise to enable accurate material identification.

The subsequent discussion systematically outlines and compares these spectroscopy methods, emphasizing their suitability for particular sample types, sensitivity levels, and instrumentation complexities Table 26. This comparative framework aims to elucidate why specific techniques, such as Raman spectroscopy for molecular precision or LIBS for elemental detection, emerge as optimal choices under defined analytical contexts. Through such detailed analysis, this section provides a comprehensive basis for selecting appropriate spectroscopic methodologies to achieve accurate material identification and characterization objectives.

Parameter	Reflection-Based Spectroscopy	Transmission-Based Spectroscopy	Laser-Induced Breakdown Spectroscopy (LIBS)	Raman Spectroscopy
Principle	Measures light reflected from a sample	Measures light transmitted through a sample	Analyzes light emitted by a plasma formed from the sample	Measures light scattered by molecules; energy shift or Raman effect
Information Obtained	Surface properties, surface chemical composition	Bulk properties, concentration of absorbing spectrum	Elemental composition, trace elements analysis	Molecular structure, chemical composition, crystallinity
Material States	Mostly solids, or reflective surface	Suitable for gases, liquids,	Solids, liquids, and gases	Solids, liquids, and gases

		and thin or transparent solids		
Samples Preparation	Minimal to none	Can requires careful sample preparation for path length	Minimal to none, though the sample is often destroyed	Minimal to none
Destructive	Non-destructive	Non-destructive	destructive	Non-destructive
Typical Application	Material surface analysis, paint characterization, thin films	Environmental monitoring, solution chemistry, medical diagnostics	Metal analysis, environmental monitoring, forensic analysis	Pharmaceutical analysis, materials science, mineralogy
Sensitivity	Dependent on surface smoothness and optical properties	High sensitivity to changes in concentration and path length	High sensitivity to all elements, capable of detecting trace levels	High sensitivity to molecular vibrations
Advantages	Quick and adaptable to in-line manufacturing processes	Accurate concentration measurements, clear and direct data	Fast, requires no sample prep, provides immediate results	Provides detailed molecular information, versatile
Limitation	Limited to surface analysis, can be influenced by surface roughness	Limited by sample transparency, thickness, and homogeneity	Requires high-powered lasers, safety considerations	Requires specialized equipment, fluorescence can interfere
Instrumentation Complexity	Moderate	Moderate to high, depending on the setup	High, due to the need for laser and spectrometer	High, due to the need for lasers and sensitive detectors

Table 26: Spectroscopy Methods Comparison

3.13.1 Raman Spectroscopy

Raman spectroscopy is a spectroscopic technique used to observe vibrational, rotational, and other low-frequency modes in a system. The Raman effect is a from of inelastic scatted light. Incident light from a laser strikes a molecule, most of the scattered light scatters elastically not resulting in an energy change. However, a small portion of the light scatters inelastically these photons have an energy change. The energy change is associated with molecular vibrations to rotations based on the material type. The energy change are classified as Stokes Raman Scattering (SRS) or Anti-Stokes Raman Scattering (ASRS). SRS is defined as if the scattered light has a lower energy than the incident light because

the photon has transferred some of its energy to the molecule. The molecule experiences a exciting vibrational mode resulting in a shift to a longer wavelength. ASRS is defined as if the scattered light has higher energy than the incident light because the molecule was initially in an excited vibration state and has lost energy to the incident photon. This results in a higher energy photon with a shorter wavelength.

Material identification is made possible by SRS and ASRS by providing a unique spectral fingerprint based on the vibration energy level of the material. Making Raman spectroscopy a powerful tool for studying the molecular composition of materials. Which is key to achieving material identification for our project.

3.13.2 Reflection Based Spectroscopy

Reflection-based Spectroscopy is a technique that analyzes the spectrum of light reflected from a sample to determine its properties. A broadband light source is incident on the sample. The reflected light interacts with the samples' surface some wavelengths will be absorbed while others are reflected dependent on the material properties. For spectral analysis the reflected light is collected, and the intensity of the light is measured based on wavelengths. The spectral data provided a fingerprint of the sample. Data Interpretation provides a way to classify the spectrum of an unknow sample to reference spectra.

3.13.2.1 Diffusion Reflectance Spectroscopy

Reflection based spectroscopy has three main methods depending on the type the material we are taking samples from: Diffuse Reflectance Spectroscopy, Specular Reflectance Spectroscopy, and Angle-resolved Spectroscopy. Diffusion Reflectance Spectroscopy (DRS) measures the light that is scattered in many directions after interacting with the sample. Samples for these methods are in the categories of powers, rough surfaces and opaque materials that produce low intensity reflections. Unlike specular reflection, diffuse reflections does not follow the law of reflection; instead, light is scattered spectrally at a various angles due to the irregularities or granularities in the surface.

3.13.2.2 Specular Reflectance Spectroscopy

Specular Reflectance Spectroscopy involves measuring the reflected light that obeys the law of reflection or Snell's Law. Meaning the angle of incidence is equal to the angle of reflection. Suitable samples include smooth and shiny surfaces (e.g. metal, glass, and polished stones). Common application for specular reflectance is used for characterizing thin films, coatings and the semiconductor industry. For our use case this would limit the sample choice of what we could identify; however, is also the simplest optical setup.

3.13.2.3 Angle-Resolved Spectroscopy

Angle-Resolved Spectroscopy modus operandi involves changing the angles of incidence and/or observation to measure how much the reflective properties of the surface change

with angle. Application is used most commonly in optical engineering for designing and analyzing optical components (e.g. lens and mirrors). In addition, it is used in photonics for quality control in layered materials manufacturing. Out of the three methods for reflection-based spectroscopy I believe that specular reflectance offers the most option for out sample choice.

3.13.3 Laser Inducted Breakdown Spectroscopy

Laser Inducted Breakdown Spectroscopy is an analytical technique that is used for determining the elemental compositional of materials. A high-powered pulsed laser is focused at the samples to ablate the surface of the material. Pulse times in this study application of LBIS were 4 ns to 8 ns [18]. The ablation process vaporizes the material creating plasma. As the excited ions in the plasma return to a lower state, they emit light. Since each element has its own characteristic wavelengths. Thus materials can be resolved. Human's fingerprints are akin to materials characteristic wavelength it's a unique identifier. To achieve Spectral Analysis the emitted light is collected and analyzed using a spectrometer.

Advantages to LIBS is its application across a wise variety of sample types. Unlike reflection-based spectroscopy which has a variety of set up to cover a limited number of samples. LIBS has applications across various fields. LIBS is used in material science, Environmental Science, Archaeology, Forensic Science and Space Exploration. LIBS is highly versatile, but it comes at a cost both financially and safely.

Most of the down sides of LIBS come from the Instrumentation complexity and safety concerns. The use of laser with nano second pulse times which on Thor labs can cost $4750. The choice of lasers and it's characters, wavelength and pulse width are critical for plasma generation. To take full account of the plasma generation a broad spectral range is needed to resolve low energy photons. Timing Systems are required to synchronize and laser pulse with the exposure time of the sensor.

Safety considerations due to the use of higher-powered laser risk eye and skin damage. Proper safety measures will have to be taken when showcasing or developing the laser system. High-Voltage systems used to drive the laser pose risk pose a safety factor as well. Plasma Generation depending on material type may generate ultraviolet light and harmful byproducts. The operation and testing of such a hazardous system in the shared lab environment of senior design

3.13.4 Transmission Based Spectroscopy

Transmission-based spectroscopy is the fundamental principle of measuring how much light passes thought a sample. By analyzing the transmitted light, it provides valuable information about the bulk properties of a material, including the concentration of

absorbing species and structural characteristics. This method is suitable for analysis on gases, liquid, and thin or transparent solids.

Variable	Definition
$\boldsymbol{I_t(\lambda)}$	Transmitted light intensity
$\boldsymbol{I_0(\lambda)}$	Relative to the incident intensity
$\boldsymbol{T}$	The transmittance (3.13.1)
$\boldsymbol{\varepsilon}$	Molar absorptivity (L/mol·cm), a material-specific constant.
$\boldsymbol{c}$	Concentration of the absorbing species (mol/L).
$\boldsymbol{l}$	Path length (cm), the distance the light travels through the sample.
$\boldsymbol{A}$	Absorbance, a measure of how much light is absorbed. Equation (2)

Table 27: Specular Reflectance Variables

$$T = \frac{I_t(\lambda)}{I_0(\lambda)} \tag{3.13.1}$$

$$A = -\log(T) \tag{3.13.2}$$

$$A = \varepsilon c l \tag{3.13.3}$$

Molar absorptivity (ε) in spectroscopy quantifies how strongly a substance absorbs light at a given wavelength. It is a material-specific constant that can be used to resolve unknown materials. Higher molar absorptivity values result in deeper absorption features in the transmitted spectrum. Transmission-based spectroscopy exhibits high sensitivity to even minor changes in both concentration and path length. This sensitivity arises from the Beer-Lambert Law (3.13.3). The thickness of the sample directly affects the amount of light absorbed. To work around this, the sensor can be exposed to longer exposure times.

In conclusion, For high-accuracy material identification, Raman spectroscopy is the most effective due to its ability to provide distinct molecular fingerprints through vibrational analysis, ensuring precise differentiation of chemical compositions. LIBS excels in elemental analysis, offering rapid, high-sensitivity detection across diverse materials, though it is destructive and requires a high-power laser. Transmission-based spectroscopy is ideal for bulk analysis of transparent or thin materials, leveraging high sensitivity to concentration and path length changes but is constrained by sample transparency and uniformity. Reflection-based spectroscopy is best suited for surface characterization but is highly influenced by surface roughness, limiting its application to bulk identification. For

molecular-level accuracy, Raman spectroscopy is superior, while LIBS is optimal for elemental composition analysis, making them the most viable techniques for high-precision material identification.

An alternative is to enhance material identification is the integration of Reflection-Transmission Hybrid Spectroscopy (RTHS), combining the strengths of both reflection-based and transmission-based techniques. This method simultaneously analyzes surface and bulk properties by capturing both reflected and transmitted spectra, significantly improving accuracy in heterogeneous or semi-transparent materials. By incorporating dual-path optical configurations, RTHS enables comprehensive spectral analysis, overcoming the limitations of individual techniques—reflection's surface dependence and transmission's transparency constraints. This hybrid approach is particularly beneficial for multi-layered materials, thin films, and biological tissues, where both surface interactions and internal composition are critical for precise characterization.

3.14 Linear CCD Sensor

In this section, we compare TCD1501D and TCD1304DG as they relate to the needs of T.R.A.S.H.

Specification	TCD1501D	TCD1304DG
Number of Elements	5000 elements	3648 elements
Pixel Size	Low: 7 µm by 7 µm	High: 8 µm by 200 µm
Sensitivity	Low: 10.4 to 15.6 V/lx·s	High: 110 to 160 V/lx·s
Dark Signal Voltage	2 mV	Up to 5 mV
Dynamic Range	3000	Up to 300 (no unit specified)
Saturation Output Voltage	3 V	450 to 600 mV
Output Impedance	0.5 to 1 kΩ	0.5 to 1 kΩ
Output Signal Voltage	Not specified	1.5 to 3.5 V
Operating Temperature	-20°C to +60°C	-25°C to +60°C
Package Type	22 pin CERDIP	22 pin CERDIP

Table 28: Linear CCD Sensor Comparison

3.14.1 Performance of Linear CCD

The TCD1304DG's much larger pixel area (i.e., 8 × 200 µm vs 7 × 7 µm) translates to significantly higher photon collection per pixel, yielding a dramatically greater sensitivity (~160 V/(lx·s) for TCD1304DG compared with only ~13 V/(lx·s) for TCD1501D). Both sensors are silicon-based and have a special response like in **Error! Reference source not found.**. This high sensitivity means the TCD1304DG can produce strong signals even with low light levels, which is critical in spectroscopy.

Moreover, the TCD1304DG has a very low dark signal (~2 mV at 10 ms integration), ensuring minimal noise contribution during long exposures. Its dynamic range (~300:1) is well-suited for capturing both weak and moderately strong spectral lines without saturation , and the built-in electronic shutter helps manage any extremely bright inputs. The output signal of the TCD1304DG is a small (~0.6 V) analog swing on a ~2.5 V DC level, which is easily handled by standard low-voltage ADCs and circuitry. In contrast, the TCD1501D's output can swing several volts (up to ~6.5 V), necessitating higher supply voltages and more complex interfacing. In summary, the TCD1304DG's combination of high sensitivity, low dark noise, adequate dynamic range, and convenient output voltage makes it ideal for spectroscopy applications, clearly outperforming the less sensitive TCD1501D for detecting faint spectral signals and providing a cleaner, more easily processed output.

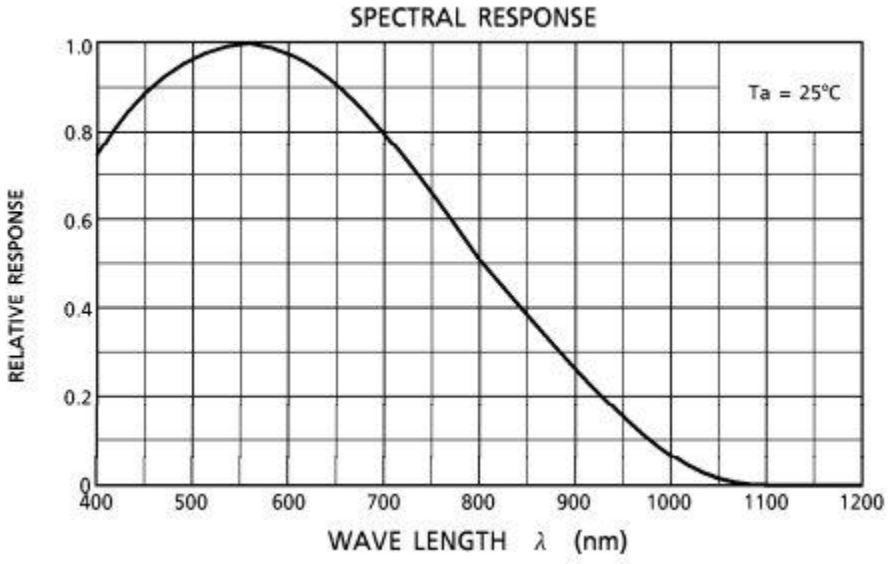


Figure 10: Spectral Response of Silicon-Based Detectors

3.15 Illumination Sources

The goal of the illumination source is to provide sufficient incident optical power across the wavelength range of 400nm to 1100nm. This range is due to the responsivity of a silicon detector.

Characteristic	Incandescent	White LED (WLED)	RGB LED	IR LED
Illumination Power	1–10 mW/mm²	10–30 mW/mm²	~5–20 mW/mm²	~2–15 mW/mm²
Power Requirements	5–50 W	0.5–5 W	~1–3 W	~0.2–2 W
Cost	<$1	$1–$3	$3–$8	$0.50–$2
Spectral Range	≈400–1100+ nm	~400–700 nm	R: ~620–630 nm G: ~520–540 nm B: ~450–470 nm	850–940 nm (typ.)
Spectral Uniformity	Excellent	~Uniform in visible range	Varies per channel; non-uniform blend	Narrowband (~20–50 nm FWHM)
Thermal Output	>80% energy lost as heat (~200–400 °C bulb temp)	Minimal (surface ~40–60 °C)	chip ~40–70 °C	Moderate (~50–80 °C)
Suitability for PETG End Effector (max 373 K)	Unsuitable (bulb >373 K)	Suitable	Suitable, not idea	Suitable
Driving Circuit Complexity	Resistor or AC line	Constant current driver (~3–5 components)	PWM driver or IC (~5–10 components)	Simple current limiter(~2–4 components)

Table 29: Illumination Source Comparison

3.15.1 Incandescent Source

Incandescent lamps consist of a filament typically tungsten inside of an evacuated glass bulb with the necessary wired connections. The tungsten filaments are excited with electrical power thus creating heat. Once the filament has reached Approximately 850 Kelvin a red glow will appear [19]. The emitted light for a given element when heated in this matter can be approximated by black body radiation.

$$I(\lambda, T) = \frac{2\pi hc^2}{\lambda^5} \frac{1}{e^{\left(hc/\lambda kT\right)} - 1} \tag{3.15.1}$$

$$\lambda_{max} T = 2.898 \times 10^{-3} m \cdot K \tag{3.15.2}$$

Using (3.15.2) we can solve for the peak wavelength and T, the resulting temperature necessary to heat the filament. With a center wavelength of 750 nm, the temperature needed to heat is 3864 K (3.15.2). The blue line in Figure 11 shows the intensity of the source across our interested wavelength range. For context compared to the sun with an average temperature of 6600 K, which defines the wavelengths of light we are accustomed to seeing [19].

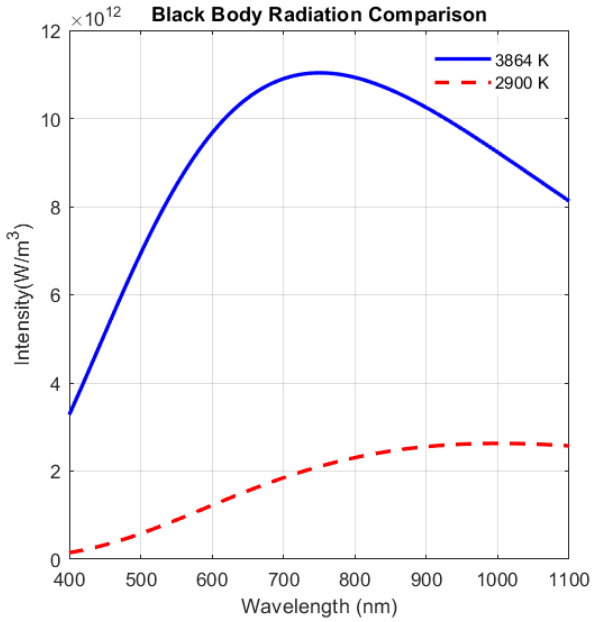


Figure 11: Black Body Radiation Comparison Theoretical

Of all possible solid filament tungsten has the highest known melting temperature of 3680 Kelvin and the lowest rate of evaporation of pure metals. In contrast carbon can withstand higher temperatures without melting however it evaporates too quickly. It is because of tungsten high melting point and low vaporization pressure that makes it the ideal case for

a incandescent source. To ensure the stable operation of a tungsten incandescent source the useful maximum temperature is about 3000 K with an average of 2900 K [19]. From the ideal temperature of 3864 K to a realistic operating temperature of 2900 K. This temperature shift greatly affects the output intensity of the visible range as seen in Figure 12.Figure

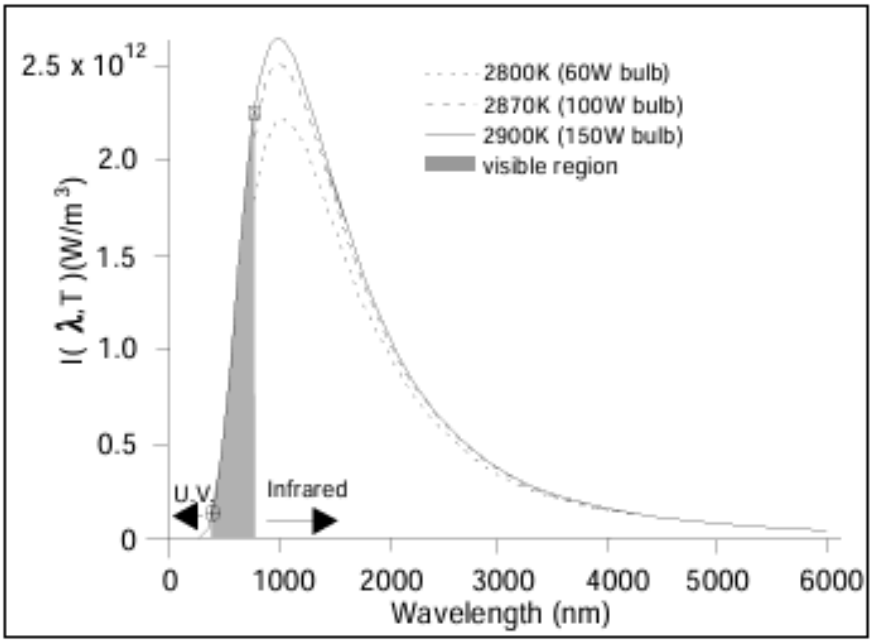


Figure 12: Black Body Radiation Comparison Actual [19]

With the stretch goal of embedding the illumination source and light capturing system into the end effector of the robot we have a limitation. The end effector is going to be 3D printed in PETG My mechanical engineer has placed the constraint The light bulb cannot heat this material above 373 K. In Figure 12 all of the white space under the curve is not contributing to useful optical power. Considering that all of the light that is outside the visible range will be considered as heat production an incandescent source is not suitable with the end effector.

The driving circuitry necessary for an incandescent source is also cumbersome. Incandescent bulbs are a resistive load that requires a controlled current or voltage source to operate safely and effectively. Traditionally, incandescent bulbs were designed for AC line voltage; However, DC power supplies can be used in modern incandescent sources. Components of the driving circuit include current limiters, series resistors, power supply requirements, and control and monitoring systems. When an incandescent source is first turned on, it has low resistance, which causes a large inrush of current to prevent damage. Current limiting resistors are necessary. The current limiting or series resistor must be chosen to provide appropriate power dissipation, as well as to keep the current within safe

limits. If an incandescent source is chosen because of its uniform illumination, across the desired wavelength range, control and monitoring will be essential to limit the thermal transfer into the PETG end effector.

3.15.2 Light Emitting Diode (LED) Source

The output spectrum of an cannot be described as simply as the black body distribution. Thus, a more in-depth explanation of how LED have their spectrum is required to make justified decision on the choice of LED's. The wavelength of an emitted photon is not simply bandgap the of the semiconductor material. Because electrons in the conduction band of the semiconductor material are a distribution of energies that are greater than the bandgap energy. The purpose of this section 3.12.2 is to justify the design choices in the following section about WLEDs, RGB LEDs and IR LEDs.

To extract light out of an LED the diode device must be forward biased. Now, the P-type material must be active thus excess electrons are being injected into that region. The excess electrons or the electron distribution $n_E(E)$, in the conduction band of the *p*-type material is a function of the density if states $\boldsymbol{g}(E)$, and the fermi-Dirac probability, $f(E)$ (3.15.3)***Error! Reference source not found.*** [20]. The peak energy hv_0 of the electron distribution can be approximated by [20]. The term $\frac{1}{2}k_BT$, descried the approximate peak of the distribution $n_E(E)$ above the conduction band and E_g is the energy difference between the valance band and the conduction band [20]. T being the temperature of the material and k_B the Boltzmann constant. There is a similar distribution p_E for the holes in the valance band of the semiconductor; however, the p_E distribution is much bigger than then that electron distribution $n_E(E)$ we can assume the rate of recombination will depend primarily on the injected electron distribution [20].

The purpose of understanding the electron distribution is to better understand how this effects photon energy distribution and thus the emission spectrum.

$$n_E(E) = \boldsymbol{g}(E)f(E) \tag{3.15.3}$$

$$hv_0 \approx E_g + \frac{1}{2}k_BT \tag{3.15.4}$$

Now that we have defined the electron distribution lets understand how this effects the emission spectrum. When these electrons recombine with holes near the valence band edge Ev, they emit photons. The photon energy is the difference between the electron and hole energy levels. Since most recombination occurs from electrons near the peak of n_E because they have the highest energy state (3.15.4) and visualized in Figure 13.

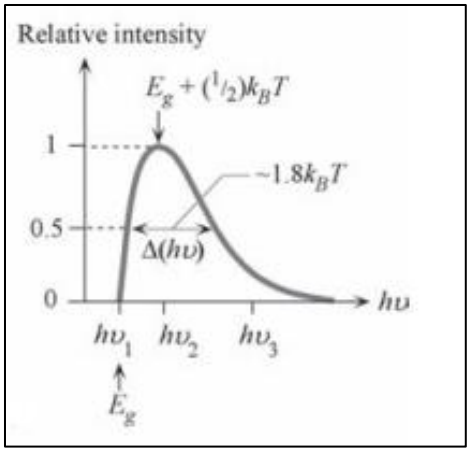


Figure 13: Relative Intensity of Light as a Function of Photon Energy [20]

The relative intensity of light as a function of photon energy based on a direct bandgap semiconductor and the electron distribution described in Next, wavelength and photon are inversely related to each other defined in (3.15.5).

$$\lambda = \frac{hc}{E} = \frac{hc}{h\nu} \tag{3.15.5}$$

h is planks constant, *c* is the speed of light and E is energy is how to connect the wavelength to the electron distribution. Using *hν* (3.15.5) from and finding the relative spread by (3.15.6). The result of is the range is the emitted wavelength from the LED. The peak of this range is the results of (3.15.7).

$$\Delta(hc) \approx E_g + 2 \times \frac{1}{2} k_B T = E_g + k_B T \tag{3.15.6}$$

$$\lambda_0 = \frac{hc}{E_g + \frac{1}{2} k_B T} \tag{3.15.7}$$

To summarize the purpose of the previous discussion, the goal was to build a foundational understanding of how LEDs emit light and to evaluate the suitability of various illumination sources for integration into the robotic end effector. The analysis clearly indicates that LEDs offer substantial advantages for this application. Specifically, mechanical engineering constraints impose a temperature limit of 373 K on the end effector, making thermal management a critical consideration. Given that the project's stretch goal is to incorporate a spectrometer into the robotic hand, thoughtful and proactive design decisions must be made to minimize integration challenges.

LED illumination sources are favorable primarily due to their low thermal output, making them suitable for the temperature constraints mentioned above. Additionally, LEDs only require a low-voltage (typically 5 V DC or less) power source, simplifying the driving circuitry significantly compared to incandescent sources. While it is true that incandescent sources offer relatively uniform optical emission across the targeted wavelength spectrum (400–1100 nm), their practical challenges, including high thermal emission and complex power requirements, outweigh this benefit.

Consequently, the subsequent section of this research will focus on addressing the challenges associated with achieving comprehensive spectral coverage from 400 nm to 1100 nm using LED-based solutions.

3.14.1.1 White LED (WLED)

A blue LED with phosphor (WLED) operates by emitting high-energy blue photons through direct recombination, as described in the previous section 3.16.2. These photons, centered around 450 nm, excite a phosphor layer that re-emits light at longer wavelengths, producing a broader spectrum across the visible range. The resulting white light combines the sharp blue LED emission with the broad, down-converted phosphor emission. Unlike direct band-to-band recombination, the phosphor spectrum does not follow the electron distribution described $n_E(E)$ (3.15.3), but rather reflects the material's fluorescence response, which broadens the overall spectral output like Figure 14. The broad and wide range of emissions makes a favorable selection for a light source.

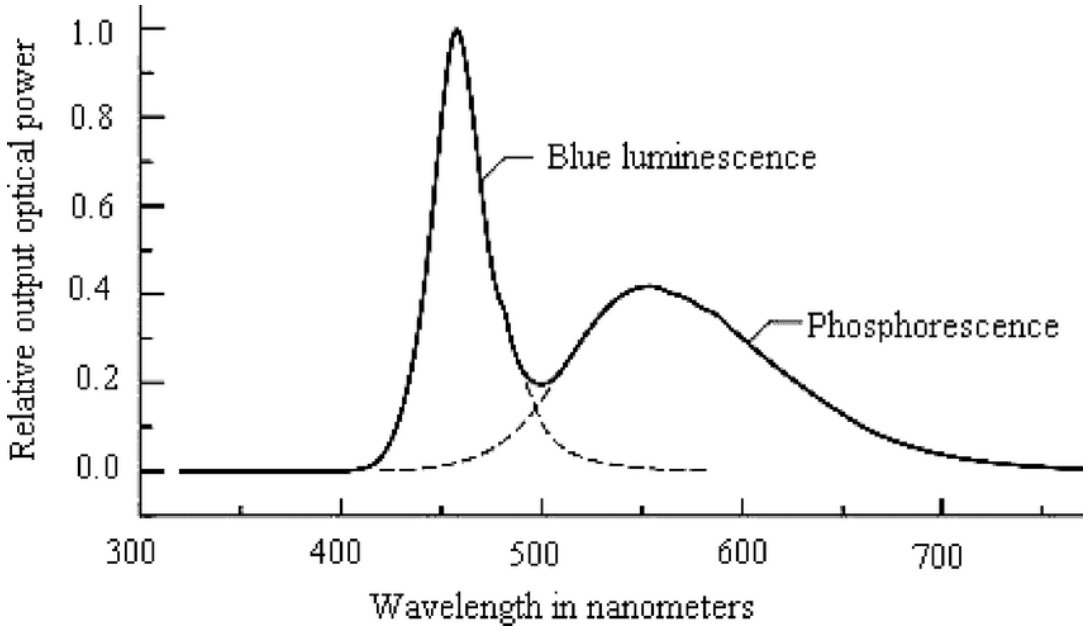


Figure 14: Emission Spectrum of WLED

3.14.1.2 RGB LED

An RGB LED consists of three individual LEDs — red, green, and blue — each with its own semiconductor material and corresponding bandgap. The emission from each sub-LED follows the electron distribution behavior explained earlier, with a spectral peak at defined by (3.15.6). These emissions are relatively narrow, and when combined, the LED outputs a multi-peak spectrum as seen in Figure 15. The overall color can be adjusted by controlling the current through each LED, allowing for dynamic color mixing. Unlike phosphor-based white LEDs, RGB LEDs produce color directly from semiconductor recombination, making them ideal for display applications.

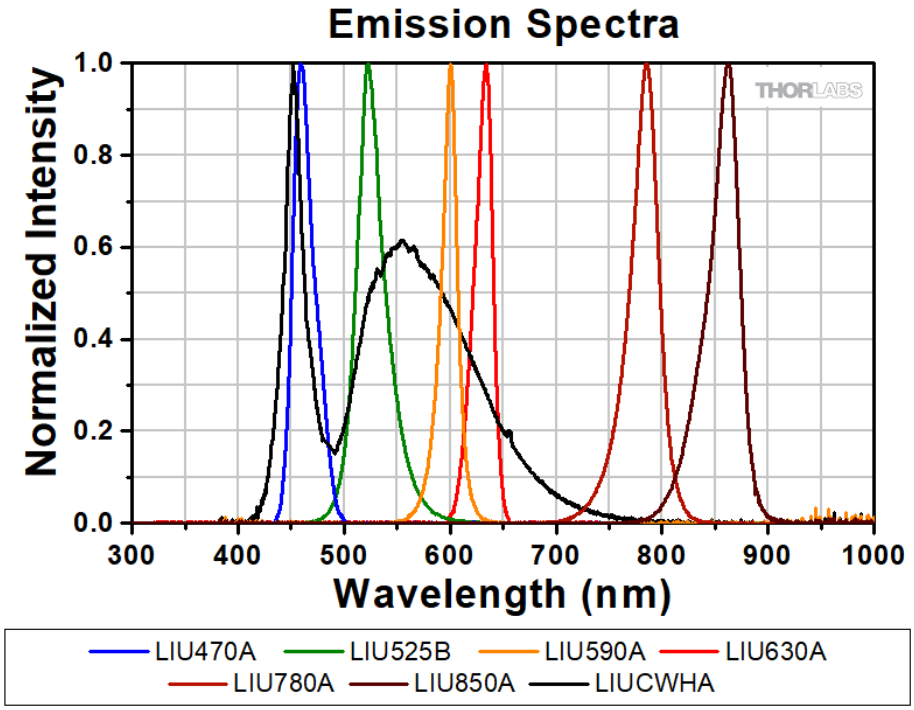


Figure 15: Spectral Information of a Variety of LEDs from Thor Labs.

3.14.1.3 Infrared LED (IR LED)

Infrared LEDs emit photons in the near-IR region, typically around 850–950 nm. These devices use semiconductor materials with smaller bandgaps, such as GaAs or AlGaAs. The emission spectrum follows the same principles discussed earlier, with the peak wavelength determined by (3.15.7). However, the observed IR LED spectra often show greater asymmetry and broader linewidths than predicted by the ideal model. This discrepancy arises from heavy doping effects, which introduce impurity bands that effectively lower E_g , as well as from photon reabsorption and recombination-induced energy redistribution. With that said illumination for WLEDs drop off at 800nm. So, to cover the entire spectrum LED's from 800nm to 1100nm are required. it would be idea to have some LEDs to cover the missing spectra. Thor labs have just the LEDs we need. The LED930L has a FWHM cover from 890nm to 960nm. In addition, the LED 1050E which covers the range from

995nm to 1105nm. The LEDs do have some overlap between these values which will be helpful for full spectrum coverage.

For the final system, an array of these three LED types: LED930L, LED1050E and a WLED. Is a favorable array because to replace the WLEDs spectrum requires four to five LEDs types. To put frank, it's easier to drive an array of WLED and have them in parallel. Then the IR LEDs cover for the WLED in the NIR range.

3.16 Resolving Position with Optics System

For the computer vision with a single camera setup resolving depth can be quite difficult. Knowing the depths from the end effector of the robot is the first step in the process of achieving the final goals of the computer vision, resolving the position data. I am going to address the fundamentals of triangulation to understand how optics can assist the computer vision in resolving position.

In a computer vision system utilizing a single camera, resolving depth information is inherently challenging due to the loss of three-dimensional context in a two-dimensional image. For robotic applications, however, accurately determining the depth of objects relative to the robot's end effector is a crucial first step toward reliable position estimation and spatial awareness.

Method	Configuration	Principle	Pros	Cons	Hardware Complexity	Depth Accuracy
Monostatic	Single source & detector (collocated)	Time-of-flight	Simple, compact, easy integration	Limited to single-point depth; ambiguities with multiple scatterers	Low	Moderate limited in complex scenes
Bistatic	Source and detector separated	Time-of-flight	Greater flexibility; diverse measurement geometries	ambiguities remain for multiple scatterers	Medium	Moderate still limited for complex scenes
Range Tomography	Multiple sources and/or sensors at multiple positions	Intersection of multiple spherical or ellipsoidal constraints	Resolves multiple scatterers; precise 3D localization; robust reconstruction	Requires multiple measurement points	High	High; accurate 3D reconstruction
Depth from	Single camera with known PSF	Image blur analysis & deconvoluti	Single camera setup: no additional	Requires precise calibration;	Low	Moderate; depends heavily on

Defocus (DfD)		on of defocus PSF	viewpoints needed; computationally efficient	sensitive to optical parameters and noise		calibration and blur-model accuracy

Table 30: Comparisons of Optical Positioning Systems

3.16.1 Triangulation

The fundamentals of triangulation via a practical example. Using probes that are both source and the sensor know as a monostatic configuration [21]. The source emits a pulse as the pulse hits the object then creating scatter. That scatter then received by the sensor using their relationship in (3.16.1) we can solve for the range of the object to the detector [21]. Using probes that are both source and the sensor know as a monostatic configuration [21]. The source emits a pulse as the pulse hits the object then creating scatter. That scatter then received by the sensor using their relationship in (3.16.2) we can solve for the range of the object to the detector [21]. The fundamentals of triangulation are via a practical example. Using probes that are both source and the sensor know as a monostatic configuration [21]. The source emits a pulse as the pulse hits the object then creating a scatter. That scatter then received by the sensor using their relationship in (3.16.1) we can solve for the range of the object to the detector [21]. The variables for**Error! Reference source not found.** τ is the time of arrival, R is the range or distance from object to detector, and v is the velocity of the pulse (3.16.1).

$$\tau = \frac{2R}{v}, \tag{3.16.1}$$

While the basic triangulation principle using a single source/sensor provides valuable depth information for a single point, a practical computer vision system, especially one intended for use in robotics requires the ability to resolve multiple scatterers or features in a scene. This introduces the need for additional measurements and system complexity.

To stay consistent with the design of the robot where we may only want one camera in the end effector, I will discuss a system to resolve position. Then discuss why adding additional detectors with the goal of improving the spatial resolution is necessary. The first system is classified as a multi-static setup, where the sensor passively receives signals from various emitters after they are reflecting or scatter from objects in the workspace. Since the setup involves the separation of source and sensor. What are the fundamental principles that resolve the range in that type of system.

The separation of the source and sensor is known as the Bistatic Configuration [21]. To solve the projection surface in the case is an ellipsoid. The source and sensor are at the foci. r_s is the position of the source, r_d is the position of the detector, r is the position of the

scatter, *v* is the velocity in the medium, and *t* is the pulse time of flight (3.16.2)**Error! Reference source not found.**.

$$|r - r_s| + |r - r_d| = vt \tag{3.16.2}$$

Now we have discussed both monostatic and bistatic configuration to resolve range. Each of these methods in their simple forms are unable to identify multiple scatters at unknow locations. By gathering a sufficient number of measurements from various locations (which provide multiple views), the positions of the scatterers can be determined. This approach calls for a tomographic method, specifically known as range tomography [21].

Range tomography does require more than one detector. Meaning having only one sensor in the end effector is not viable for this solution. However, if tomography can affect the best solution to 3D positioning implementing more sensors is viable. As previously discussed, monostatic and bistatic configurations each provide valuable but limited spatial information due to ambiguities when multiple scatterers are present [21].

By acquiring multiple range measurements from distinct locations, one can intersect numerous spherical or ellipsoidal constraints, effectively pinpointing the positions of the scatterers. This method, known specifically as range tomography, emerges as an ideal solution to the challenges faced by single-view triangulation approaches. Therefore, while incorporating tomography would necessitate deploying additional detectors or emitters, its substantial improvement in spatial resolution and accuracy makes this increased complexity both justified and strategically beneficial for robust 3D positioning in robotic vision systems.

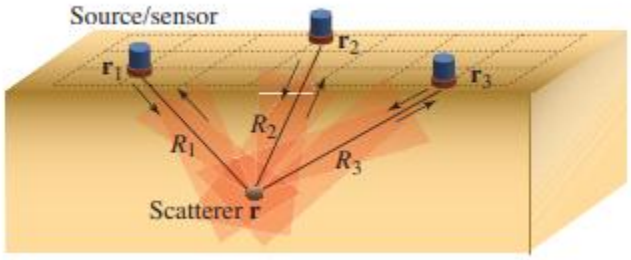


Figure 16: Range Tomography [21]

The purpose of Figure 16: Range Tomography [21] is to visualize the method of range tomography. It is three monostatic units to resolve the location of a single scatter.

Triangulation serves as the fundamental principle for resolving depth and spatial position using time-of-flight measurements as seen in Figure 16. In a monostatic configuration,

where the source and sensor are collocated, range is determined based on the round-trip time of a pulse reflected from a scatterer. This setup is simple and compact—ideal for robotic systems with limited hardware capacity, such as a single camera mounted on an end effector. In contrast, a bistatic configuration separates the source and detector, enabling more diverse measurement geometries. Each bistatic measurement constrains the scatterer's position to an ellipsoidal surface, offering greater flexibility and potential accuracy in complex scenes. While both configurations can determine the range to a single point, they are limited when multiple scatterers are present. To resolve full spatial distributions of unknown scatterers, these configurations must be extended into a tomographic system, where multiple measurements from different locations are combined. This process, known as range tomography, enables the reconstruction of a 3D scene by interpreting the overlapping constraints from each measurement. In robotic vision, such tomographic approaches—rooted in triangulation—allow a single mobile sensor, paired with multiple emitters, to build a depth-aware understanding of the environment for accurate positioning and navigation.

If the addition of depth sensing is necessary, range tomography is the most comprehensive system to solve the distance problem.

3.16.2 Deblurring method

If you have a single camera (or lens) and can measure how large the defocus blur spot is, along with calibrations of lens aperture and focal length, then, in principle, you can infer the distance of the underlying object plane. Range tomography requires the implementation of an additional optical system the deblurring method can be implemented with the initial hardware design.

One way to recover three-dimensional information from a single camera is by exploiting the phenomenon of defocus blur. When an optical system images a scene that extends beyond a specific focal plane, points away from that plane appear blurred. By modeling this blur—often with a known point spread function (PSF)—you can systematically "invert" the blur effect and estimate how far each point is from the camera. This approach, commonly referred to as Depth from Defocus (DfD), effectively derives depth by measuring and interpreting the size of the blur circle in the recorded image[21].

$$g(x,y) = h(x,y) \otimes f(x,y), \tag{3.16.3}$$

where $f(x, y)$ is the true scene and $h(x, y)$ is the PSF associated with the blur (3.16.3). Because this blur is linear and shift-invariant (i.e. often well-approximated by a Gaussian for defocus), one can perform deconvolution or deblurring to recover $f(x,y)$ from $g(x, y)$. In principle, once the shape or width of the blur kernel is known at each pixel or region, the resulting variations can be mapped back to scene depth, since objects further from the lens's focal plane will produce larger blur spots. Furthermore, by leveraging the kinematics of a robot arm, multiple images can be acquired at different distances from the object; in

each image, the deblurring effect is measured and used to refine the relationship between blur radius and focal plane offset, ultimately enabling precise localization of the object in three-dimensional space.

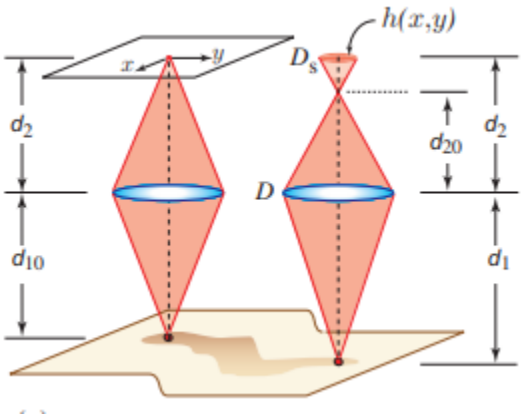


Figure 17: Deblurring Effect [21]

This image was taken from source [14]. To illustrate how a change is distance results in deblurring. As well as serve as a reference of the variable location for the following equations (3.16.4) and (3.16.5).

$$h_{in}(x,y) \propto \left| P_1\left(\frac{2\pi}{\lambda d_2}, \frac{2\pi}{\lambda d_2}\right)\right|^2 \tag{3.16.4}$$

$$\frac{1}{d_e} = \frac{1}{d_1} + \frac{1}{d_2} - \frac{1}{f} \tag{3.16.5}$$

The underlying geometry in Equations (3.16.4) and (3.16.5) shows that moving an object away from the plane of best focus increases the blur spot, in the recorded image. By measuring this blur diameter and calibrating it against known lens parameters (such as the aperture size and focal length), one can systematically infer how far each part of the scene is from the lens by the variable d_e in (3.17.2). (3.17.1) is to help solve the PSF associated with the blur.

Hence, even with a single camera, the amount of defocus provides valuable depth cues. After acquiring one or more images at specific focus settings, deblurring algorithms estimate the PSF and its variation, allowing them to reconstruct both a sharper image and an approximate depth map of the scene. This technique, depth from defocus, does not require stereo vision or multiple viewpoints and can be implemented on ordinary hardware, provided the blur model is well defined and the system is adequately calibrated.

3.17 Dispersion Techniques: Diffraction Grating vs Prisms

Spectrometers rely on dispersive elements to separate light into its component wavelengths. In dispersive spectrometers – including modern imaging spectrometers – this is achieved by either a prism or a diffraction grating, which spreads the incident spectrum across a detector array. Imaging spectrometers (used in hyperspectral remote sensing) form a two-dimensional image while simultaneously dispersing light in the orthogonal direction to encode spectral information[22].

Aspect	Diffraction Gratings	Prisms
Dispersion Principle	Diffraction and interference from periodic structures	Refraction due to wavelength-dependent refractive index
Spectral Range	Broad coverage (UV to IR) achievable through careful design	Limited by prism material transmission window; typically visible to short-wave IR
Resolving Power	High resolving power, easily tunable via grating geometry	Moderate resolving power, constrained by material and geometry
Spectral Efficiency	High efficiency and control over dispersion	Lower efficiency; significant reflection losses, especially at IR
Size/Form Factor	Compact design; flexible geometry	Usually bulkier due to material thickness and geometry constraints
Cost	Cost-effective, especially for broad spectral ranges	Higher cost, especially for IR-specific materials (CaF_2, ZnSe, alkali halides)
Complexity	More complex design and manufacturing	Simpler, straightforward optical design
Stray Light	Potentially higher stray light; requires careful design	Lower stray light; advantageous for sensitive applications
Application Suitability	Preferred for high-resolution and imaging spectrometers, versatile spectral analysis	Ideal for simpler setups requiring moderate resolution and high spectral purity

3.17.1 Principle of Operation

Optical gratings are precisely engineered optical components that produce dispersion through the physics of diffraction and interference. A grating typically consists of a surface or volume with a periodic structure. When light strikes this periodic structure, it is diffracted into multiple directions, corresponding to different diffraction orders *m.*(3.17.1)

which quantify how each wavelength is sent into a different angle. This means that as wavelength changes, the outgoing angle of a given order changes, leading to spatial separation of colors. $\Delta\lambda_{min}$ (3.17.2) quantifies the minimum resolvable spectral separation. Notice that it's not constant across the system it's also wavelength dependent.

$$sin\theta_d = sin\theta_i + m\frac{\lambda}{\Lambda} \tag{3.17.1}$$

$$\Delta\lambda_{min} = 1.22\frac{\Lambda}{D}\frac{cos\theta_d}{m}\lambda \tag{3.17.2}$$

Symbol	Meaning
Λ	grating period
D	beam diameter
$\Delta\lambda_{min}$	minimum-resolvable spectral separation
m	diffraction order
θ_i	incident angle
θ_d	dispersion angle

Table 31: Relevant terms for (3.17.1) and (3.17.2) from [22].

Gratings are especially favored in imaging spectrometers because they can provide the required spectral resolution and range while maintaining a compact form factor [22]. Gratings are used in spectrometers over prisms in many cases because of their superior dispersion control and flexibility in peak transmission point. A well-designed grating spectrometer can achieve much higher resolving power than a prism-based design of similar size [22]. Equation (3.17.2) helps to approximate the minimum-resolvable spectral separation I can expect to get on the final design. The resulting dispersion calculated using (3.17.2). Gratings can also be used over a wider range of wavelengths including ultraviolet and infrared bands where prism materials become impractical [22]. These advantages make diffraction gratings ubiquitous in modern spectrophotometers, monochromators, and imaging spectrometers.

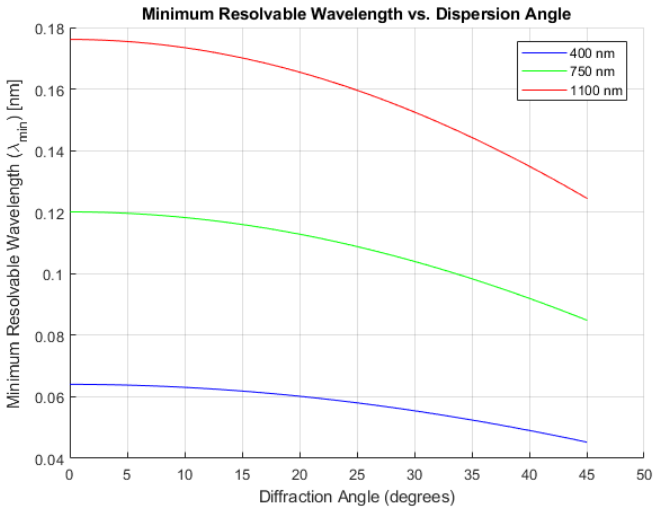


Figure 18: $\Delta\lambda_{min}$ vs Dispersion Angle

For the end-of-the-semester demo, I will be using a diffraction grating with the following specs: 600 grooves/mm, blazes wavelength of 1000 nm, 12.7 mm by 12.7 mm. The reason I chose this diffraction grating is for the following reasons. The blazed wavelength is 1000nm, and since the responsivity of a silicon detector drops off, the choice is limited to the NIR. To have the peak reflective at 1000nm will allow a greater response of the system in the NIR range. As evident in Figure 18 the $\Delta\lambda_{min}$ is well below the spectral resolution stated in the goals.

Next, a deeper dive into prisms and their spectral cover. Prisms have a limited spectral coverage dictated by the transmission window of the prism material[22]. For example, common optical glasses can cover the visible and part of the near-infrared (up to ~1.5 µm), and fused silica can extend into the short-wave IR (~2.5 µm)[22]. To reach further into the infrared, one must use specialized IR prism materials (such as CaF_2, ZnSe, or alkali halide crystals), which often come with drawbacks like high refractive indices leading to large reflection losses[22]. Using Prisms as the Dispersion mechanism can be cost-prohibitive due to the material choice. As well as increasing the footprint of the spectrometer.

In conclusion, diffraction gratings have become the dispersive element of choice in most modern spectrometers, especially imaging spectrometers, due to their high dispersion efficiency, tunable resolving power, and broad spectral usability[22]. They enable compact, high-performance spectrometers that can operate from the UV through the IR with appropriate configurations. Prisms, being the older technology, still offer advantages in simplicity and spectral purity, and are well-suited for moderate-resolution applications over limited wavelength ranges [22]. The choice between a grating and a prism ultimately hinges on the instrument requirements: for high resolution or extended spectral coverage, gratings typically outperform, whereas for low stray light and simplicity in a restricted band, a prism spectrometer may be preferable.

In summary, we have outlined the fundamental working principles of optical gratings and justified their prevalent use in spectrometers, while also comparing them with prism-based approaches in terms of dispersion mechanism, resolving power, spectral range, and optical performance.

3.18 Spectral Characteristics of Recycled Materials

A critical aspect of the project is the successful identification of the materials, so the robot arm knows where to place them. The goal of this section is to see which materials can be identified in the 400 nm to 1100 nm range. The goal of this section is to find a set of materials that have sufficient peak wavelengths to successfully identify the materials.

3.18.1 Distinguishable Materials

Materials that lack distinctive characteristic wavelengths can be greatly enhanced by applying the first derivative transformation to the Spectra, which can greatly enhance the subtle specular differences. Recalling from calculus, the first derivative accentuates inflection points in the spectrum and turns small absorption features into pronounced features.

Identifying common plastics (e.g., PET, HDPE, PP, PVC, PS). Fewer reflective spectroscopy in the defined range is challenging because their raw reflectance spectra often look similar.

I found a source that can identify PET from PVC with spectral information that is less than 1100 nm. For the samples of the plastics, they used a variety of colors and ages of the plastic. The separation of these two plastics is critical for recycling efforts. The presence of PVC in plastics waste may generate hazardous chlorinated compounds[23]. This accuracy in needed for successful material identification using a spectrometer; however, this accuracy is overkill for our goals of identifying trash against recycling.

Paper and cardboard are cellulose-based, giving them a different spectral signature than plastics. Dry paper (especially white office paper) tends to have high diffuse reflectance throughout the visible and NIR, resulting in a bright, relatively flat spectrum from 400 nm up to about 1300 nm. Cardboard (brown fiberboard) contains lignin and other impurities, which absorb more in the blue and violet end of the spectrum – this is why brown cardboard appears dull brown. No sharp absorption features occur in paper's reflectance below 1100 nm – the chemical bonds in cellulose (C–H, O–H) do have overtone absorptions, but the strongest are at ~1450 nm and 1940 nm (water/OH) and ~2100 nm (cellulose combination band), outside our range. The lack of strong absorption bands below 1100nm makes it harder to distinguish from other objects.

Some of the only materials found where the material is distinguishable are glass. Glass behaves very differently from plastics or paper in this spectral range. Clear glass (e.g. flint glass bottles) is mostly transparent from 400–1100 nm, meaning it does not reflect or absorb much – instead, most light passes through. Thus, a light source will have to be placed in order to use transmission spectroscopy. Colored glass, however, has deliberate additives that absorb specific wavelengths: Green glass (e.g. wine bottles colored with iron chromite) strongly absorbs in the red and NIR part of the spectrum, while transmitting green wavelengths. amber glass effectively filters out <500 nm "high-energy" light almost completely, giving it a dark brown appearance.

Material	Wavelengths	Unit
PET (Polyethylene terephthalate)	1130, 1170, 1420, 1660	nm
PVC (Polyvinyl chloride)	1200, 1420, 1480	nm
HDPE (High-density PE)	1210	nm
PP (Polypropylene)	1180, 1190	nm
PS (Polystyrene)	1675	nm
Paper	High reflectance from 400 to 1100	nm
Aluminum	High reflectance from 400 to 1100	nm
Copper	High reflectance in the 600 to 700	nm
Green Glass	600 to 700	nm
Brown Glass	400 to 500	nm

Table 32: Materials and Their Characteristic Wavelengths

Table 32: Materials and Their Characteristic Wavelengths is a collection of data from studies [24] and [23]. Notice the lack of materials that have discernible features in the 400nm to 1100nm range. It's going to be challenging to differentiate material using just the visible spectrum.

3.18.2 The Usefulness of The Spectrometer

After evaluating the spectral characteristics of plastic, paper, cardboard, and glass. Each common recyclable would be suitable for the scope of the project. It is evident that almost all of these common recyclables have at least some measurable features within the wavelength range of interest. The near-infrared reflect data from 900 nm to 1100 nm, which falls within our sensitivity range. Even if these points are often subtle output changes, I will raise the concern that the responsivity of our sensor also drops off in these longer wavelengths. If the spectrometer makes it into the final design, optimization will have to be done in the sensors' lowest responsivity area. For the materials that fall within the more visible higher responsivity of the sensor in the case of paper and aluminum, there's a lack

of distinguishable features. Of the possible items that I would consider distinguishable, it leaves us with copper, green glass, and brown glass.

There is a lackluster of distinguishable features that reside inside the wavelength of our silicon-based detector. It begs the question of whether a spectrometer limited to this range is a suitable identification method. A silicon-based spectrometer may not be sufficient to reduce the ambiguity in identifying a wider range of recyclables.

Since this project was already going to use computer vision to solve location, challenges were faced in resolving distance and position. There is a question that an optical-based time of flight system can help with finding the locations of the objects. Then, for computer vision, use object identification. Since the accuracy in which the trash and recycling will be defined as blue and yellow bins used in home recycling.

4.0 Standards and Design Constraints

This section contains all the information on standards used as well as all constraints that needed to be considered

4.1 Relevant Standards

The following standards have been found to have relevance to our project. At this moment, core components are yet to be fully decided making more detailed discussion not possible at this time. As more exact components are chosen and deceloped, more and more exact standard discussions will be expanded as necessary. Efforst have been made to create specifications that encapsulate the standards in a much more easily verifiable mode. This will allow fro a much broader view of the following standards that play a key part in the project in hadn.

4.1.1 Universal Asynchronous Receiver-Transmitter (UART)

UART is a serial communication protocol that enables asynchronous serial data transfer between devices. It is based on the industry standard TL16C550 asynchronous communications element [25]. It works with just two wires—transmit (TX) and receive (RX) and doesn’t need a shared clock between devices. This makes it simple to implement and ideal for basic communication between microcontrollers or for debugging purposes.

In our project, we use UART in multiple places. The main one is the connection between the Raspberry Pi and the ESP32 that controls our motors. This allows us to send commands from the Pi to the ESP32 and get feedback, like motor position or system status, back in return. We also used UART to talk to our MKS Servo42C stepper motor drivers. These drivers accept specific command packets through UART that let us move to a certain angle, set the speed, or even read back the current angle from the motor.

Another reason we utilize UART is that it makes testing and debugging a lot easier. Since most development boards (like the ESP32) already have built-in UART support, we can just plug them into USB and open a serial monitor to test the system. We will use UART a lot during development to print debug messages or watch how our commands are being processed.

On the hardware side, UART is easy to wire up, but we had to be careful about voltage levels. The Raspberry Pi and ESP32 both use 3.3V logic, but if we ever wanted to connect to something using 5V, we added a level shifter to avoid damaging the chips.

Overall, UART is important in our system for simple, fast communication between devices. It keeps things lightweight and flexible, which is great when we're building and testing parts of the project separately.

4.1.2 Inter-Integrated Circuits (I2C)

I2C is a de facto world standard for low-speed serial communication, initially developed by Philips (now NXP)[26]. It uses just two lines, SDA for data and SCL for clock, and supports multiple devices on the same bus by using addresses for communication with a specific device, which makes it perfect for systems with limited pins or space. It's especially useful for connecting multiple devices efficiently without a lot of wiring or extra complexity.

We will use I2C in our system to control the PCA9685 servo driver. This chip generates multiple PWM outputs, which is super helpful for driving our servo motors. Instead of generating PWM signals directly from the ESP32, which would take up a lot of processing power, we let the PCA9685 do the hard work. The ESP32 just sends simple I2C commands to tell each servo what angle to move to.

We plan to use I2C for any sensors we add later on, like temperature or voltage monitors. That way, we don't have to deal with a bunch of extra UART connections or SPI lines. I2C keeps our wiring clean and makes it easier to expand the system later if we need to.

Sometimes, pull-up resistors on the SDA and SCL lines can help ensure the signals stay clean. On the PCB, we try to keep those lines short and away from any high-power motor wires to avoid interference.

Because it's a widely adopted standard, I2C ensures that our system remains compatible with various devices and development tools. It's well-supported in libraries and tools, and

most microcontrollers (like the ESP32 and STM32) have I2C built in. That's why it's one of the main protocols we rely on.

4.1.3 Serial Peripheral Interface (SPI)

SPI is a more complex serial communication protocol, mainly used for short-distance wired communication between ICs. It's a de facto standard designed by Motorola during the 1980s, and it's known for being faster and more reliable than I2C in many use cases. It uses four wires—MOSI, MISO, SCLK, and CS—and supports full-duplex communication. We plan to use SPI in areas of the system where speed is critical.

Even though it uses more pins, SPI is great for data-heavy applications, which makes it a solid choice for performance-sensitive parts of our robotic system, and it is a great interface to integrate into a PCB.

4.1.4 Universal Serial Bus (USB)

USB is a standard for connecting and transferring data between devices, and it's basically everywhere in modern electronics. In our project, we use USB mainly for programming microcontrollers, powering development boards, and sending data from the STM32 to a computer.

One important use is with the STM32-based spectrometer board. We send the CCD data over USB using the built-in USB OTG feature, which acts like a virtual COM port on the computer. This lets us read raw sensor data in real time for testing and debugging. USB is also used to power both the ESP32 and STM32 boards during early development.

Because USB is standardized and well-supported, it makes testing and communication much easier. We don't have to worry about special drivers or hardware interfaces. It's also fast enough for most of the data we need to send, especially for plots or logs.

4.1.5 PCB Design Standards

To ensure our system is reliable and safe, we followed common industry PCB design guidelines, including recommendations from IPC standards like IPC-2221 (generic PCB design standard). These guidelines help with trace width selection, spacing between signals, via size, and overall layout rules to prevent issues like short circuits or interference.

For example, we used proper trace widths based on the current each signal carries — thicker traces for motor power lines and thinner ones for logic signals. We also separated high-power areas (like stepper motor drivers and 12V power rails) from sensitive logic components to avoid noise. Decoupling capacitors were placed close to ICs, as recommended in IPC-2221, to stabilize voltage and reduce ripple.

Following these PCB design standards helped us reduce errors, improve signal integrity, and make the board easier to debug. It also ensures that the board could be manufactured by most PCB fabs without modification.

4.1.6 Robot Operating System (ROS) Compliance

For our robotic software framework, we selected ROS 2 due to its strong presence in both industry and academia, as well as its compatibility with compact hardware like the Raspberry Pi. ROS 2 allows us to build a modular system where each part of the robot—such as vision, motion, and control—can operate independently but communicate seamlessly. This structure supports efficient debugging, future expansion, and clean integration across all subsystems.

ROS 2 also provides access to a wide range of open-source tools essential to our development. We are using MoveIt 2 for motion planning and manipulation of the robotic arm, Nav2 for potential navigation applications, OpenCV for real-time object detection and classification, RVIz for live system visualization, and Gazebo for simulating the complete robotic workflow in a controlled environment. These tools help us streamline both testing and deployment without the need for costly third-party software.

By leveraging ROS 2 and its ecosystem, we ensure that our system is scalable, cost-effective, and aligned with modern robotics standards. It enables us to simulate, visualize, and control the robot smoothly, all while meeting key design constraints related to performance, power, and ease of development.

4.1.7 ISO 10218

ISO 10218 is an international standard that outlines safety requirements for industrial robots. While our project isn't an industrial robot in the traditional sense, we're still referencing this standard to guide how we handle safety. For example, we're considering things like emergency stop buttons, safe operating speeds, and minimizing risk during human interaction.

Following ISO 10218 helps us build a system that's not only functional but also safe to operate in a lab or demo setting. It also prepares us in case this system ever needs to meet stricter safety regulations for more professional use.

4.2 Design Constraints

In this section, we will look at the individual constraints that have shaped the project.

4.2.1 Economic Constraints

When it comes to designing a robot arm, there are many aspects to take into account. For most industrial robots, the main area of concern relates to accuracy and precision. Both of these criteria relate to the field of controls, which is a highly complex mathematical field. As the environments these industrial robots operate in become more complex creating more variables, the robots require ever more complex control systems to maintain proper functionality. One way that many roboticists in the industrial automation sector combat the complexities that deal with control systems is by increasing the weight of these robots. Increasing the weight of the robot makes robot harder to move, reducing the amount of backlash and overshoot. This then creates a new issue of settling time with the robot end effector taking a considerably larger amount of time to reach its desired goal. To address this new challenge, roboticists use large speed reducers and higher torque motors to be able to move the now thousand-pound robot arm. With higher density material increasing the weight, along with complicated speed reducers, custom machine milled, and actuators capable of moving multi-ton payloads, industrial robot arms are an expensive price tag for manufacturing plants and are completely unreasonable for small-scale operation.

In the waste management industry, a significant amount of revenue is earned through the processing and recycling of waste material. Some waste is sorted for waste-to-energy facilities to generate electricity. Other items are collected for composting. Some items are collected and sold as recycled material. Therefore, it is of great interest for waste management facilities to efficiently organize and collect different materials.

4.2.2 Safety Constraints

Safety is always a major concern when it comes to robotics, especially in environments where people and machines operate in the same space. Traditional industrial robots are often large, heavy, and powered by high-torque motors, which makes it difficult, if not impossible, for them to safely share a workspace with humans. In most industrial setups, robots are kept in isolated zones with safety cages or emergency stop systems to reduce the risk of injury.

In contrast, the T.R.A.S.H. system is designed with safety in mind from the beginning. One of our goals was to keep the entire system lightweight and use less powerful motors. This not only lowers the risk of injury in case of contact but also reduces overall system cost. Because the robotic arm doesn’t require high-powered motors to operate, it’s much safer to run in environments where people may be nearby, such as waste sorting facilities or demonstration settings.

We also selected our motor drivers with safety in mind. For example, the MKS Servo42C stepper drivers we’re using include built-in PID control and real-time feedback. This gives us the ability to fine-tune the arm’s movement and limit how fast or aggressively it responds. If someone accidentally touches the arm while it's moving, the PID parameters

can be configured to respond smoothly and reduce impact, rather than continuing at full force.

By focusing on a lightweight build, using lower-power components, and implementing control features like PID tuning, our system is much safer to interact with than typical industrial setups. This makes it a better fit for environments where isolated zones aren't practical and allows workers or operators to be around the robot without requiring strict isolation.

4.2.3 Environmental Constraints

One of the main concerns in the realm of robotics is energy. While robotics are more effective than manual laborers, they are not as energy efficient. Human beings are extremely energy efficient. However, machinery requires a substantial amount of power in order to operate as desired. This power comes at an environmental cost. While there are many green forms of energy, the industry is not advanced enough to cover an industrial robotics revolution. Therefore, all required energy for these industrial robots will utilize an increasing amount of non-renewable energy, negatively affecting the environment.

Because T.R.A.S.H. utilizes weaker motors than conventional industrial robots it will not have as much of a power draw as other industrial robots. It will, therefore, be more environmentally friendly. It is also possible that with enough optimization and efficiency, waste management plants will be able to generate their own electricity from waste products, helping combat the increasing power requirements generated by the facilities. Someday, waste facilities may even reach a net positive in power consumption.

4.2.4 Manufacturability Constraints

Common industrial robots are made completely in-house with little purchasing of parts. Some robotics industries go so far as to manufacture their own motors to fit their design requirements. While these companies do create beautiful and innovative products, they also require a lot of funding to manufacture and tend to deviate a company's resources to areas it does not need to focus.

T.R.A.S.H. will not be taking a component design approach. Instead, our project will use readily available resources and components to reduce the necessary time and resources spent on fitting design requirements. There will be a number of custom-designed components. However, these components will be easily manufactured using FDM printing. We will use this manufacturing process as the 3D printing industry as become more available at higher quality.

4.2.5 Time Constraints

Time is one of the most limiting factors throughout the development of the T.R.A.S.H. system. Since our team decided to complete Senior Design 1 and 2 during the Spring and Summer semesters, we do not have the usual timeline most groups get. The Summer

semester is also four weeks shorter than the Fall, which means we have to move faster through critical phases like testing, debugging, and final integration.

We knew from the start that time would be tight, so we created a Gantt chart early in the semester to lay out our tasks and deadlines. This helped us break the project into manageable phases and keep track of what needed to be done each week. Even with that, unexpected setbacks still happened. A few team members had to deal with major family events during the semester, which affected how much time and energy we could dedicate at certain points.

Because of these time constraints, we made intentional design choices to stay on schedule. Instead of designing custom motor drivers or making some mechanical parts from scratch, we used reliable off-the-shelf components like the MKS Servo42C stepper drivers. This saved us a lot of time on both the hardware and software sides, allowing us to focus on testing and integration. We also leaned heavily on 3D printing to speed up mechanical prototyping. Being able to redesign quickly and reprint parts helps us stay on track and respond to changes fast enough.

Time limitations also pushed us to think realistically about essential features and what could be saved for later. For example, we are focused on getting a stable, working robotic arm with basic sorting functionality before diving into advanced features like optimization or complex machine learning integration. This way, we ensure we have a functioning system before spending time on stretch goals.

4.2.6 Optical Safety Constraints

The use of visible white LEDs (WLEDs) and infrared (IR) LEDs as illumination sources in the spectrometer introduces specific optical safety considerations. Because the T.R.A.S.H. system is designed for use in industrial environments where operators may be in proximity to the light source, care must be taken to minimize exposure to potentially hazardous optical radiation.

White LEDs used in the system emit broadband visible light, typically peaking in the blue region around 450 nm. Prolonged exposure to intense blue light can pose a photochemical retinal hazard, commonly referred to as “blue light hazard.” Similarly, the IR LEDs emit radiation beyond 780 nm, which is invisible to the human eye but can still be focused onto the retina and cause thermal damage without triggering an aversion response. Even though the system operates in the tens of milliwatts range—below Class 3B laser thresholds—the lack of enclosure and close working distance elevates the need for caution.

To address this, we imposed several design constraints. First, the LEDs are angled away from the operator line of sight and recessed within the mechanical housing to reduce direct

exposure. Second, we limit the drive current to ensure the radiant intensity remains within safe exposure limits defined by the IEC 62471 standard for photobiological safety of lamps and lamp systems. Additionally, visible warning labels and training guidelines are provided to operators. In future iterations, we plan to include passive optical filters and possibly enclosures with interlocks for high-power variants.

By carefully managing power levels, orientation, and user awareness, the spectrometer's optical system is designed to be safe for prolonged industrial use without requiring protective eyewear—ensuring functionality without compromising safety.

5.0 Comparison of ChatGPT and other Similar Platforms

As artificial intelligence tools rapidly evolve, their role in academic and engineering environments continues to expand. In Senior Design, tools such as ChatGPT, Gemini, DeepSeek, and Microsoft Copilot serve as intelligent assistants—streamlining tasks such as code debugging, hardware integration, documentation, and concept explanation. While these tools offer tremendous benefits in accelerating productivity and understanding, their overuse or misuse can hinder the development of critical thinking, engineering judgment, and self-reliance.

This chapter evaluates these AI tools through both a technical and educational lens, beginning with an overview of their underlying language models and culminating in their practical strengths, limitations, and ethical considerations.

5.1 ChatGPT – Open AI

ChatGPT is built on OpenAI's GPT-4 architecture, a large multimodal transformer trained on an extensive dataset of internet text, books, and code repositories. It uses self-attention mechanisms to understand context, generate responses, and provide coherent reasoning across long prompts. GPT-4 is fine-tuned using reinforcement learning from human feedback (RLHF), enhancing its ability to maintain conversational logic, emulate technical reasoning, and generate structured text.

In our project, ChatGPT helped identify issues in both hardware timing logic and Python-based UI development. It was particularly effective at debugging errors related to serial communication and giving quick overviews of hardware protocols like UART or I2C.

Pros:

- Excellent for explaining technical concepts, identifying logic flaws, and guiding research organization.
- Capable of formatting full documentation drafts with correct structure and grammar.
- Provides iterative problem-solving support in software and hardware debugging.

Cons:

- May hallucinate technical information if prompts are vague.
- Over-reliance can discourage independent troubleshooting and lead to superficial comprehension.
- Not specialized for code execution—suggested snippets must be verified and tested.

5.2 Gemini – Google DeepMind

Gemini, developed by DeepMind, is a multimodal model designed to synthesize text, image, and audio input. The Gemini 1.5 series builds upon Google's internal Pathways Language Model (PaLM) infrastructure with enhanced real-time access to Google Search, making it especially powerful for gathering current events and literature. It excels in summarization and factual recall rather than hands-on code reasoning.

Gemini could be used to locate recent publications on image sensors and optical detection algorithms during our literature review phase. Its integration with search tools made it effective for discovering credible sources related to spectrometry and embedded systems.

Pros:

- Integrates live web access for real-time information gathering and cross-referencing.
- Useful for finding academic references and identifying industry trends.
- Ideal for summarizing datasets, manuals, and datasheets.

Cons:

- Less reliable for in-depth technical reasoning or syntax-level debugging.
- Interface lacks context persistence, making follow-up queries more difficult to refine.
- Output may oversimplify complex systems when compared to engineering-specific models.

5.3 DeepSeek

DeepSeek is a code-centric language model based on the DeepSeek-V2 architecture. It is trained extensively on public code repositories (e.g., GitHub) with a focus on low-level and high-level programming languages. Its tokenizer and model layers are optimized for interpreting syntax, generating efficient code blocks, and tracing logical relationships between functions.

However, unlike ChatGPT or Copilot, DeepSeek operates more like a static model—it lacks strong context chaining and real-time validation. Additionally, its training sources have raised concerns about secure data handling, particularly when models are trained on proprietary or open-source but non-permissively licensed code.

Pros:

- Highly optimized for code structure generation, especially in embedded languages (C/C++).
- Helps detect hardware register misuse or missed configuration steps.
- Accelerates setup of boilerplate code for microcontroller firmware.

Cons:

- Security risk: potential use of data from unknown or non-permissive repositories.
- Limited contextual awareness; struggles with multi-step problem-solving across files.
- Cannot validate correctness—generated code must be manually reviewed and tested.

5.4 Microsoft Co-Pilot

Copilot is powered by OpenAI's Codex model, a derivative of GPT-3 fine-tuned specifically on public source code. Integrated directly into IDEs like Visual Studio Code, Copilot offers real-time autocomplete and contextual code suggestions based on existing project files. Its advantage lies in situational awareness: the more complete your codebase, the more accurate its suggestions.

Pros:

- Seamless integration with coding environments.
- Excellent for real-time code generation, syntax completion, and logic inference.
- Saves time during repetitive firmware and frontend development tasks.

Cons:

- Code quality is dependent on existing context—errors in the base code can propagate.
- Limited outside IDE use; ineffective for documentation or system design tasks.
- Occasionally generates inefficient or unsafe code without warnings.

5.5 Broader Impact of AI Tool Use in Senior Design

I tools like ChatGPT, Gemini, DeepSeek, and Copilot have redefined the workflow of engineering design. They enable rapid prototyping, real-time debugging, and deeper exploration of hardware-software interactions. In our own experience, AI played a crucial role in catching timing mismatches in our UART communication, formatting our spectrometer interface logic, and generating ROS2 node templates.

However, the increased dependence on these tools raises serious questions:

- Are students still learning how to debug from first principles?
- Do teams verify suggestions through testing, or are they blindly trusting outputs?
- Is AI complementing or replacing engineering intuition?

These concerns are not hypothetical. Misapplication of AI in critical areas—such as signal timing, safety checks, or library dependencies—can result in significant system failures. Furthermore, tools like DeepSeek must be vetted for code security, as unverified sources may introduce vulnerabilities.

Ultimately, AI should act as a co-pilot, not an autopilot. Students must remain the system architects, fully accountable for understanding the logic, functionality, and safety of every element in their designs.

5.5.1 AI Example 1

During the early Research phase, one of the first big challenges we faced was choosing a microcontroller. With so many options available, each with different specs, prices, and features. It felt overwhelming. We weren't sure where to even begin. So, we asked ChatGPT as a starting point.

The first thing we asked was something along the lines of:

"The first thing on my agenda is to research MCU chips and decide what we would use…" Instead of giving us an answer right away, ChatGPT asked a follow-up: *"What factors do you have in mind?"* At that point, we realized we didn't know what factors even mattered

yet, so we told it that. It then gave us a helpful list of things to consider when selecting an MCU, including I/O requirements, communication protocols, power consumption, clock speed, and cost.

This list gave us a clear path to follow. We used it to compare chips like the ESP32, STM32F401, and ATmega328P, and eventually narrowed down our decision based on our system's specific needs. This shows how AI helped not by giving us a final answer, but by helping us ask the right questions and start thinking like engineers. (See Appendix E.1 for full prompt and ChatGPT response.)

5.5.2 AI Example 2

As we evaluated encoder options for our stepper and servo motor systems, we needed a clear comparison between potentiometer-based, magnetic, and optical encoders. While we had a basic idea of each type, we weren't confident about the accuracy, resolution, and cost trade-offs. We used ChatGPT to generate a summary table after asking: *"Give me a table summary comparing potentiometer, magnetic, and optical encoders. Key features I would like to know are accuracy, resolution, durability, cost, and precision."*

ChatGPT responded with a well-organized table that clearly compared all three types across those features. The result was so helpful and clearly written that we ended up including the table directly in our report after reviewing it for accuracy against the datasheets.

This saved us significant time and helped our team make a decision to go with magnetic encoders for the stepper motors, due to their good balance of precision, durability, and moderate cost. *(See Appendix E.2 for the prompt and full output.)*

5.5.3 AI Example 3

While testing our STM32 firmware for the TCD1304 linear CCD sensor, we needed to verify the contents of a buffer array called CCDPixelBuffer[]. We were using STM32CubeIDE but weren't sure how to view the contents of this array during debugging.

We asked ChatGPT:

"CCDPixelBuffer[] how do I show those on the debugger"

It walked us through the full process step-by-step from entering debug mode, to opening the "Expressions" tab, and finally, how to manually inspect buffer contents by adding expressions like CCDPixelBuffer[0], or even &CCDPixelBuffer[0]. The explanation was clear, direct, and immediately applicable to our IDE setup.

Using this guidance, we were able to view the pixel buffer live while stepping through our code. This helped us confirm that our DMA and ADC setup was capturing values correctly and that our buffer wasn't misaligned or empty. This is a good example of how ChatGPT filled in knowledge gaps related to tools, not just code logic or theory. *(See Appendix A.3 for the full prompt and output.)*

6.0 Hardware Design

In this chapter, we provide a detailed explanation of the hardware components used in our project. Each subsystem is made up of several key parts, and when combined, they form an organized and functional setup that powers the T.R.A.S.H. system. Our goal is to present the hardware in a clear and organized way, making it easy to understand how each part contributes to the overall operation and control of the project.

6.1 First Subsystem: Power Management

The power management system is designed to efficiently distribute power to various components in the robotic system while ensuring stability and reliability. the system will be powered using a wall outlet through a 12V power supply unit, which serves as the primary voltage source for the stepper motors and is also used to generate the lower voltages required by other subsystems. The voltage conversion process involves a 5V DC/DC buck converter and a 3.3V low-dropout voltage regulator (LDO), each serving specific roles in powering different parts of the system.

The 12V power supply is primarily used to drive the stepper motors, as they require a higher voltage to operate. Stepper motors draw significant amounts of current, and all other devices will also be powered from the same source, so the power source must be capable of supplying enough current to run the whole system. The Voltage regulators will regulate the power delivery and prevent sudden fluctuations from affecting the components.

Since the Raspberry Pi, servo motors, and TCD1304DG linear CCD sensor require a stable 5V power source, a 5V buck converter is used to step down the 12V input efficiently. A buck converter is chosen instead of an LDO for this step due to its significantly higher efficiency, minimizing power loss and heat dissipation, especially when having a large difference between the input voltage and the desired output voltage. The servo motors used for the end effector in the robotic arm and the Raspberry PI will be connected to the buck converter directly since they require a consistent 5V input to ensure smooth operation. Additionally, the TCD1304DG linear CCD sensor operates at 4V and the STM32 and ESP32 microcontrollers operate at 3.3V, so the Buck converter will serve as a pre-regulator for the LDO regulators, deriving power from the 5V rail instead of the 12V supply to improve efficiency.

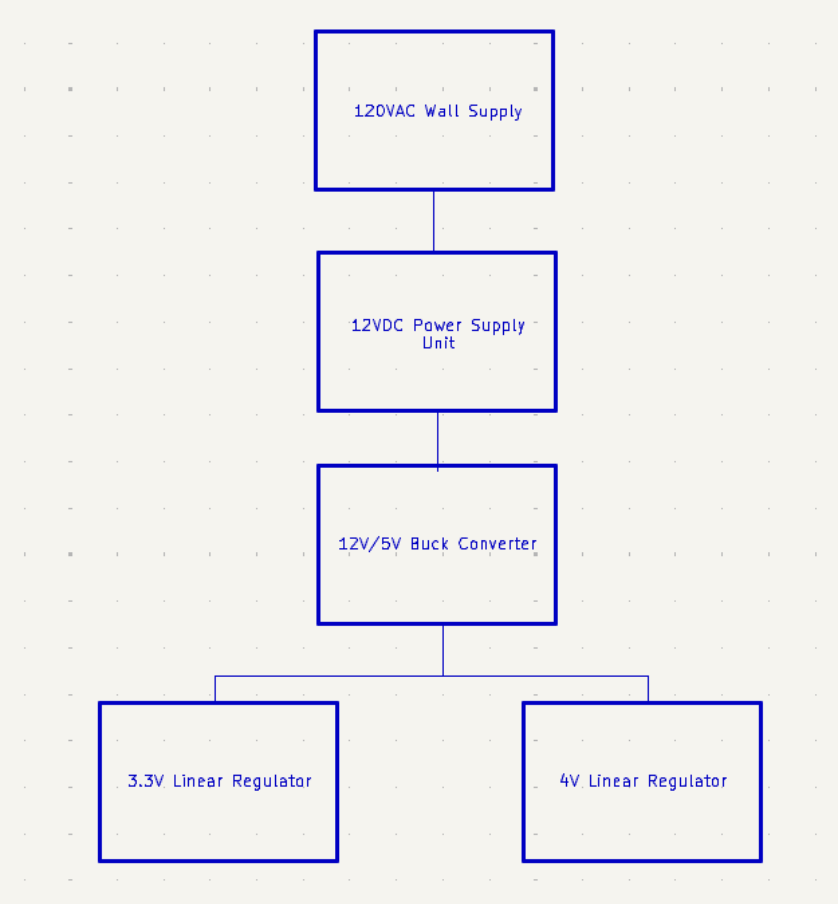


Figure 19: Power System Architecture

In summary, as illustrated in Figure 19, the 12V power supply serves as the main supply of the system, directly powering the stepper motors while also feeding the 5V buck converter to support other components. The power distribution system is designed to optimize efficiency, minimize heat dissipation, and ensure reliable operation across all subsystems. The block diagram provided below illustrates the complete power management architecture, showing how each voltage rail is derived and distributed throughout the system.

6.1.1 12V to 5V DC/DC Buck Converter Schematic

The schematic in Figure 20 shows the schematic diagram of the 12V to 5V Buck Converter. We have designed this circuit using the TI WEBENCH tool, which created a customized power supply circuit based on system requirements. This circuit outputs 5V, up to 5A of current. If more power is needed for our system, we could either use two of these buck converters, one for the servos' power input and one for all other devices, which could possibly help with efficiency as well. Alternatively, we could design a more complex power supply circuit with a different switching regulator that can handle the extra power needed to be delivered.

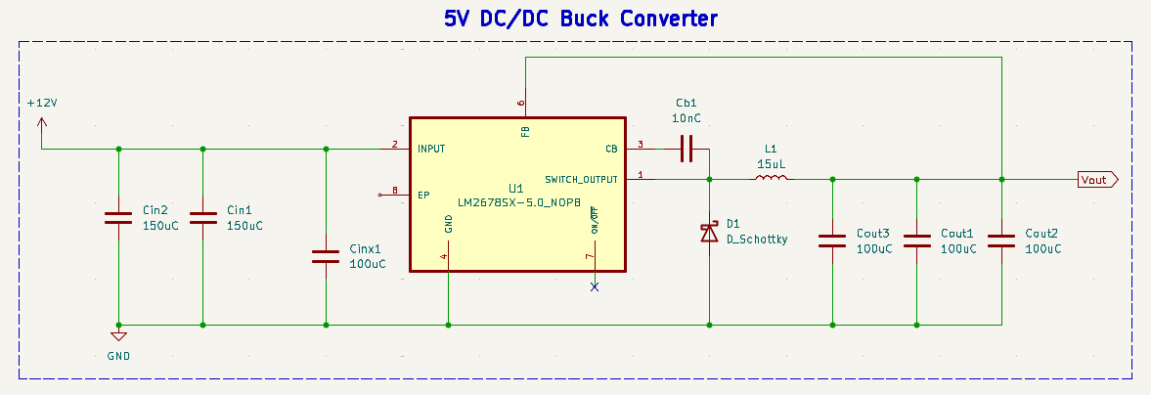


Figure 20: 5V Buck Converter Schematic

6.1.2 5V to 3.3V LDO Regulator Schematic

This schematic in Figure 21 shows the circuit for the 3.3V low dropout voltage linear regulator that is used to power the STM32F401 and the ESP32-S3-WROOM microcontrollers, it is a very simple and generic regulator design using the AMS1117-3.3 IC with decoupling capacitors at the input and output pins. The AMS1117-3.3 can output up to 1A of current, which is plenty for our microcontroller. The dropout voltage is around 1V to 1.3V according to the datasheet. So using a 5V as an input to the regulator wouldn't be an issue.

Additionally, a red LED was added to debug and verify that the LDO is working as expected. The 1k ohm resistor limits the current going into the LED to prevent it from burning out. Red LEDs have a forward voltage of 1.8V and a max current of 20mA. The voltage across the resistor would be 3.3V-1.8V = 1.5V. With a 1k ohm resistor, the current through this loop is 1.5mA, which is below the maximum allowed.

Currently, the regulator is designed to take the input from the USB connection (VBUS) used for the microcontroller. However, a direct connection to the +5V rail is also an option.

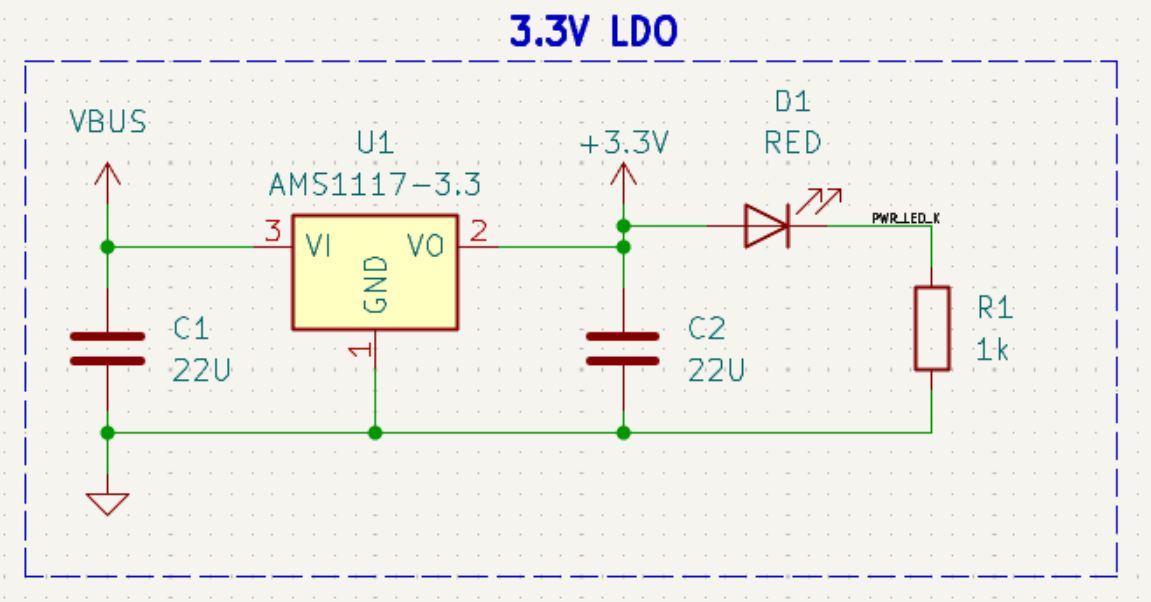


Figure 21: 3.3V LDO Regulator Schematic

6.2 Second Subsystem: Spectrometer TCD1304DG Sensor

This subsystem focuses on driving the TCD1304DG linear CCD used in our spectrometer system. Linear CCDs collect photons in pixels, convert them into electrical charges, and shift them to the output pin sequentially. The sensor operation relies on precise timing control for charge transfer and readout. The driving circuit and timing requirements provided in the TCD1304DG datasheet [27] ensure that the sensor receives the correct clock signals for charge transfer, pixel shifting, and analog signal output. The design includes a PNP transistor that acts as a buffer for the output signal (OS) before sending it to the ADC to digitize it. A 4V power supply, along with some decoupling capacitors to minimize noise, is recommended.

The circuit requires three essential clock signals:

1. ICG (Integration Clear Gate):
   - This pulse defines the integration period, during which the sensor accumulates charge based on incident light.
   - A high pulse of at least 1 µs is required to reset and start a new exposure cycle.
2. SH (Shift Gate):
   - The SH pulse controls the charge transfer from the sensor's photodiodes to the CCD shift register.
   - It should have a pulse width of at least 1 µs, ensuring proper charge movement.

3. Master Clock (MCLK):

- A continuous high-frequency clock is required to shift the pixel charges sequentially.
- This signal determines the readout speed and pixel clocking rate.

This subsystem will be implemented on a separate PCB mounted on the spectrometer. It will interface with the main PCB via a USB Micro port, providing 5V power input and MCU to Raspberry Pi Communication.

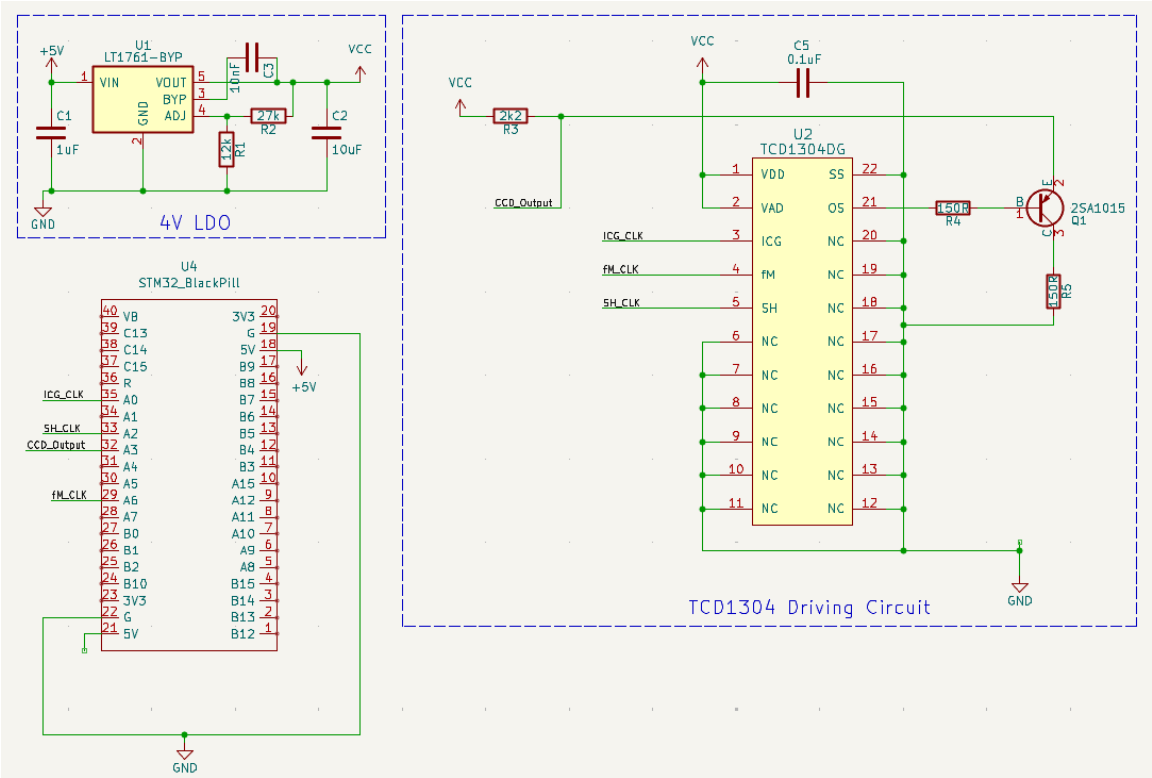


Figure 22: TCD1304DG Linear CCD Sensor Schematic

The schematic shown in Figure 22 follows the typical driving circuit in the datasheet, with an additional 4V low dropout voltage regulator to step down the 5V power coming from the main PCB.

6.3 Third Subsystem: ESP32-S3-WROOM and Motor Control

This subsystem includes most of the major parts of our project, which will be on the main PCB. It consists of an MCU dedicated to controlling the motors, servo motors, PWM drivers, stepper motors, and their drivers and encoders.

6.3.1 MCU Schematic

The schematic shown in Figure 23 shows the ESP32 microcontroller used along with its bare minimum components needed to function. Such as decoupling capacitors, flashing buttons and pins, and connecting the GPIO pins needed for our project. One I2C Bus and three UARTs are all we need from the microcontroller. The I2C is used to communicate with the PCA9685 PWM Driver. One UART is for stepper drivers, another UART is for interfacing with the Raspberry Pi, and the last UART is for programming the MCU.

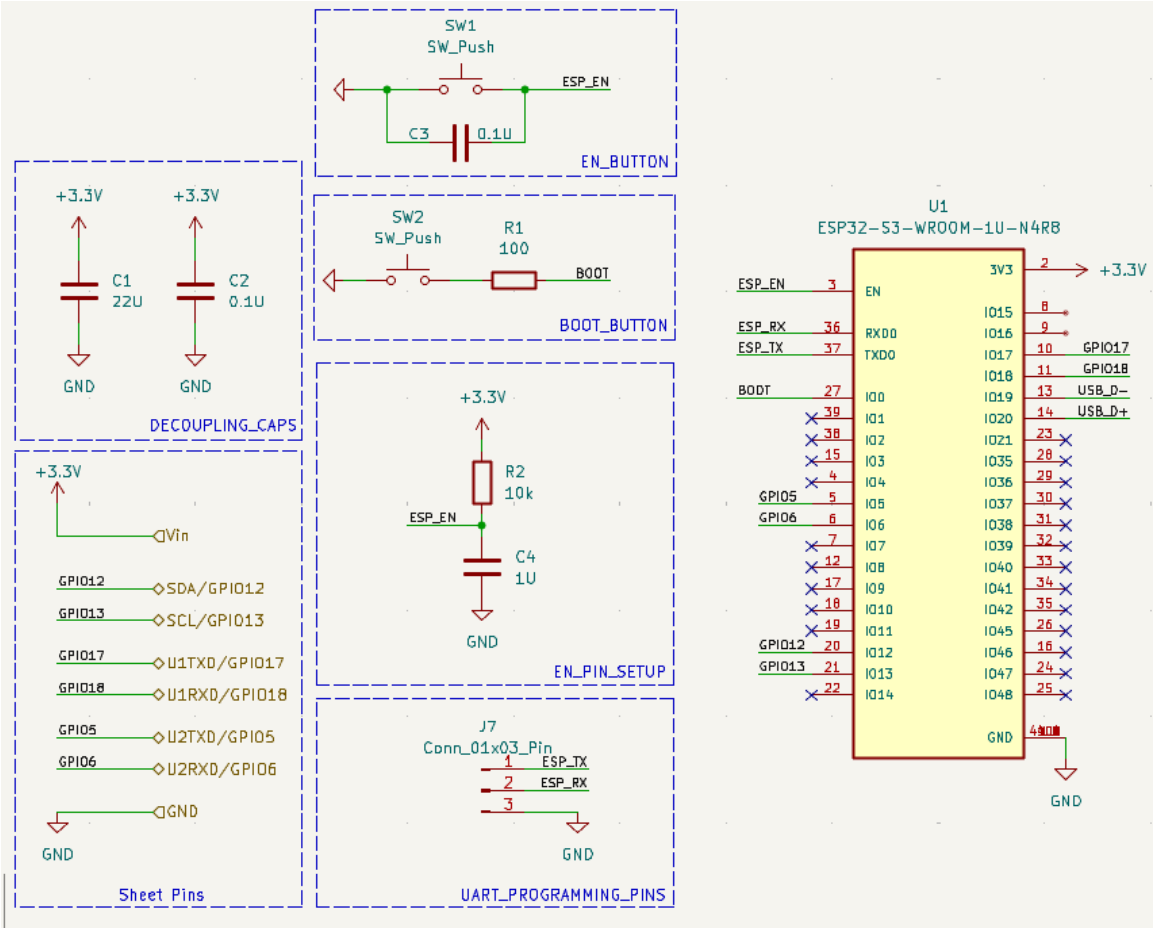


Figure 23: ESP32-S3-WROOM MCU Schematic

6.3.2 Stepper Motor Connections

The diagram in Figure 24 is an illustration of the MKS servo42C stepper motor encoder and driver connections. Each motor requires five wires, two for power and the other three

for UART communication. It is a very simple setup that provides a lot of functionality to our project.

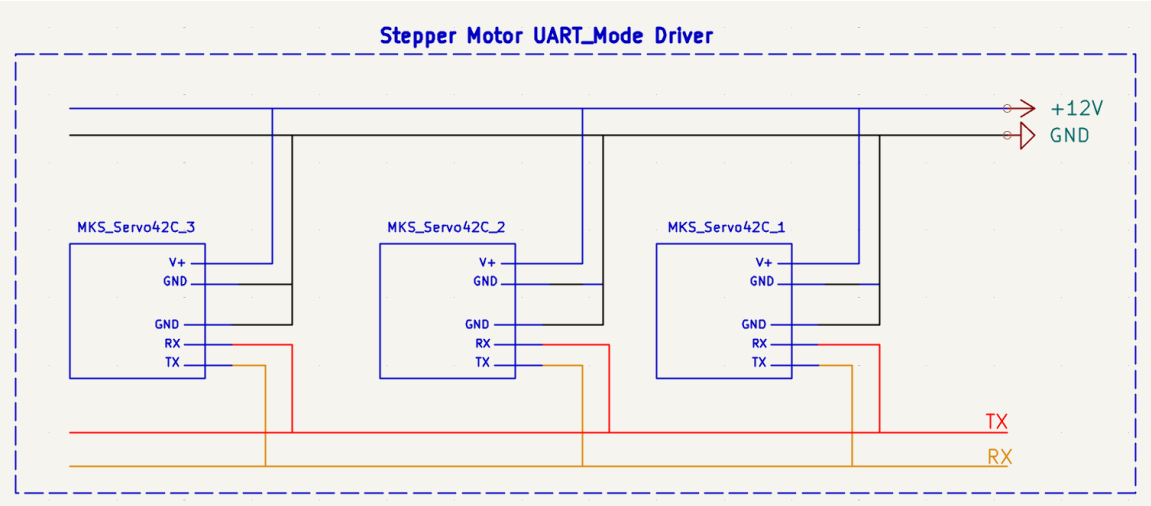


Figure 24: Stepper Motor Wiring Diagram

6.3.3 Servo Motor Driver Schematic

The PCA9685 I2C-controlled PWM Driver schematic shown in Figure 25 is provided by Adafruit, the designer of this module. We will integrate this driver circuit into our PCB design to drive the servo motors. Since this design is a 16-channel PWM driver and we only have three servo motors, we can remove unnecessary channels in our custom design to reduce the footprint.

The schematic consists of a terminal block for power input, reverse polarity protection, an LED to verify power connection, multiple 3-pin connectors to connect the motors, and some resistors. Other than removing the extra channel connectors, we can remove the terminal block since we will power the circuit directly from the PCB. Additionally, since we do not need multiple PCA9685 chips, we can hard-set the I2C address by connecting the A0 to A5 pins together. The final schematic can be seen in Figure 26.

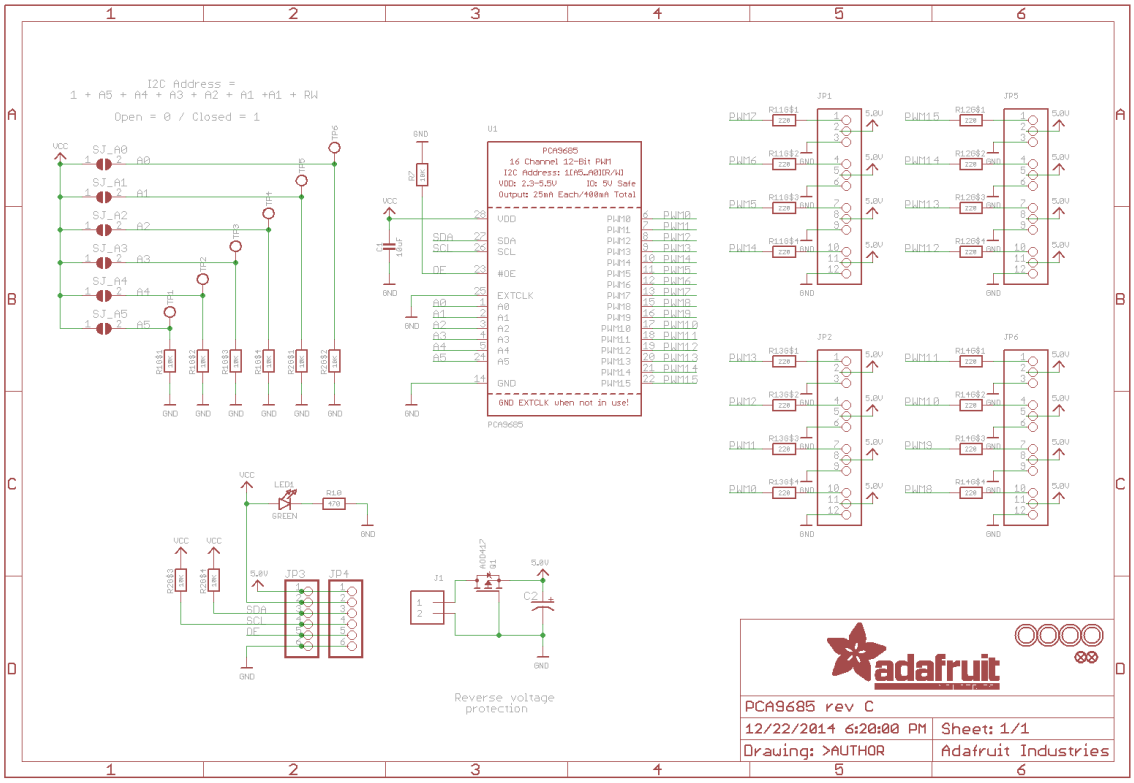


Figure 25: Adafruit PCA9685 PWM Driver Reference Schematic

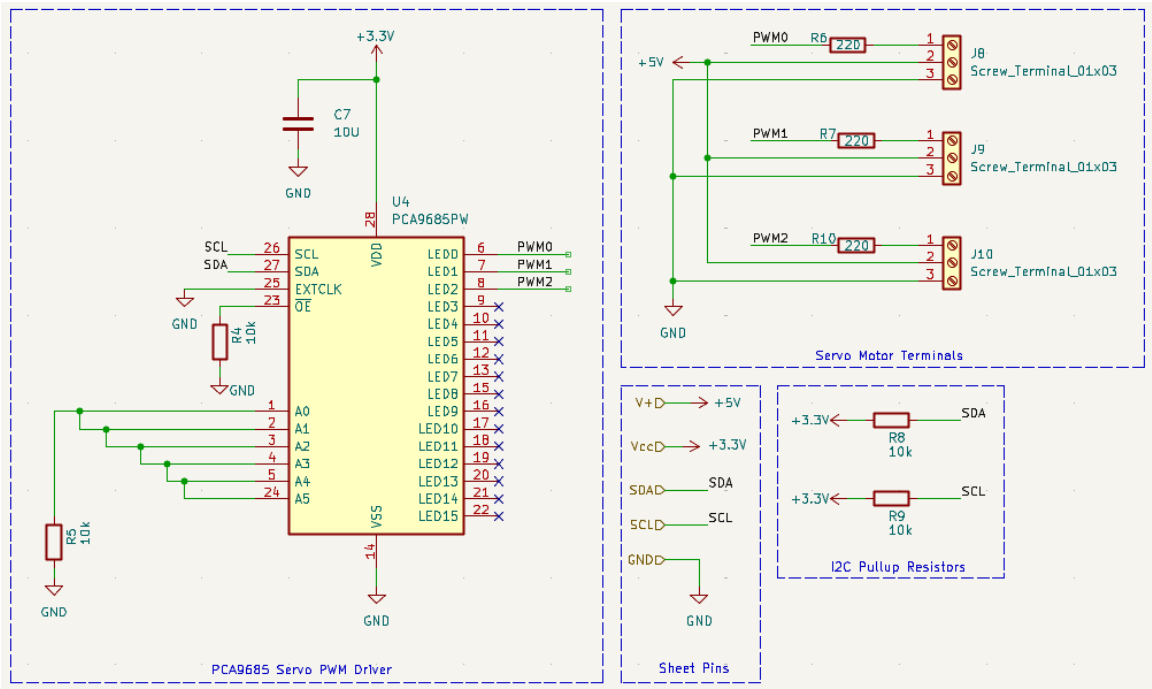


Figure 26: PCA9685 PWM Driver Final Schematic

6.4 Fourth Subsystem: Raspberry Pi and Other Peripherals

The final subsystem includes the Raspberry Pi along with various peripherals such as the computer vision camera and an optional touchscreen display. Most peripherals in this subsystem connect externally via USB or HDMI, and therefore do not require internal schematics. The only components requiring schematic-level connections are the ESP32 UART connections that interface with the Raspberry Pi through its GPIO pins, power, and ground. Figure 27 illustrates these GPIO connections in detail.

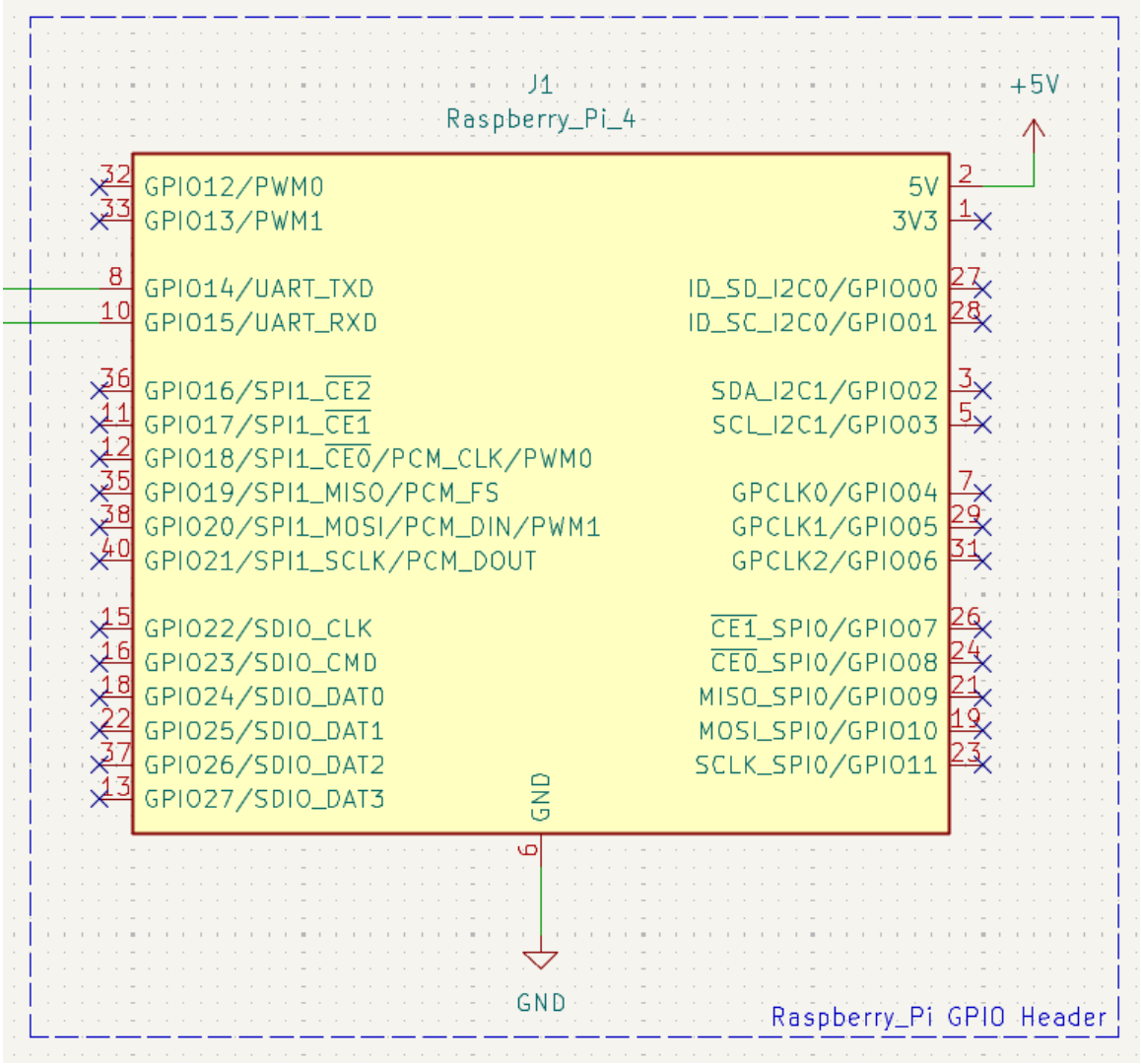


Figure 27: Raspberry Pi GPIO Connections Schematic

6.5 Hardware Design Summary

The block diagram shown in Figure 28 provides an overview of all subsystems and how they interconnect to form the complete T.R.A.S.H. system. Additionally, Figure 29 displays the schematic of the main PCB, which integrates most of these subsystems and includes terminal blocks for connecting external components.

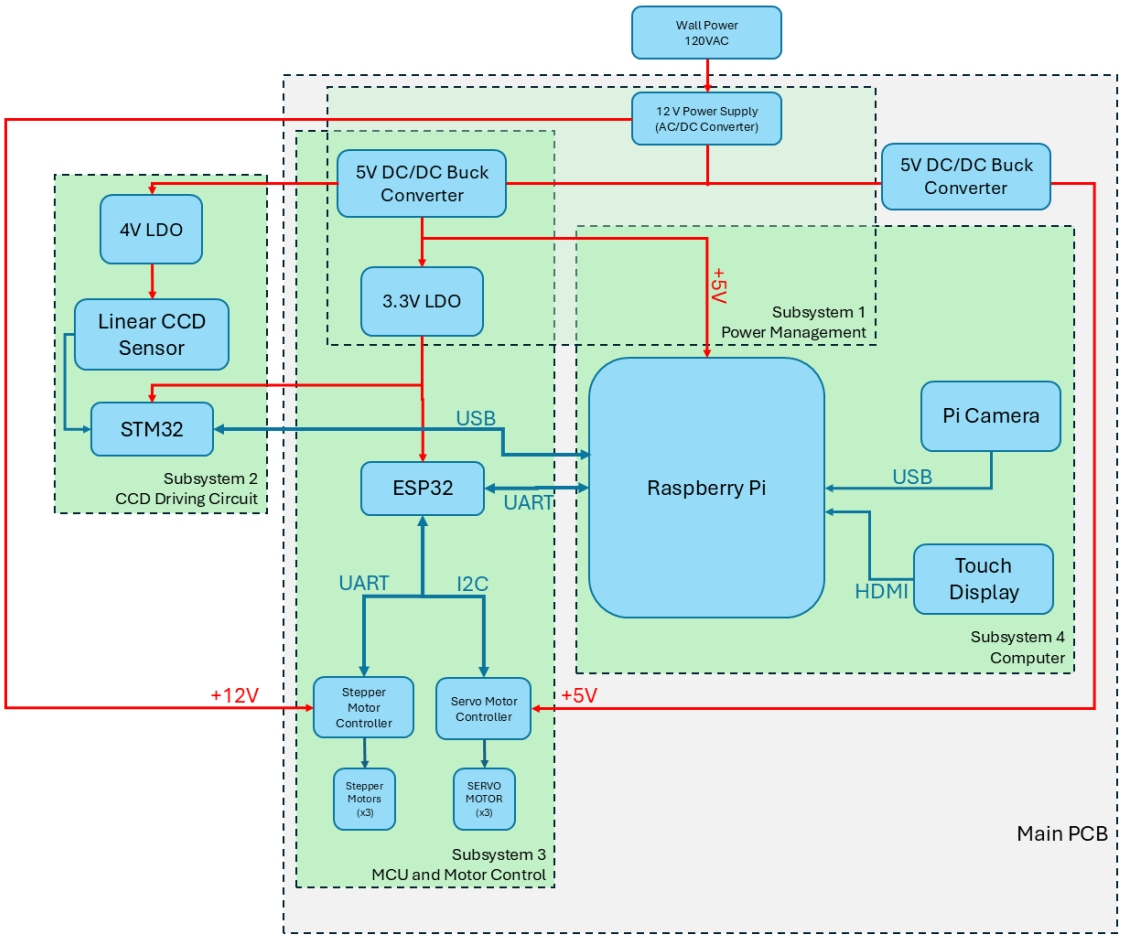


Figure 28: Hardware Design Summary Block Diagram

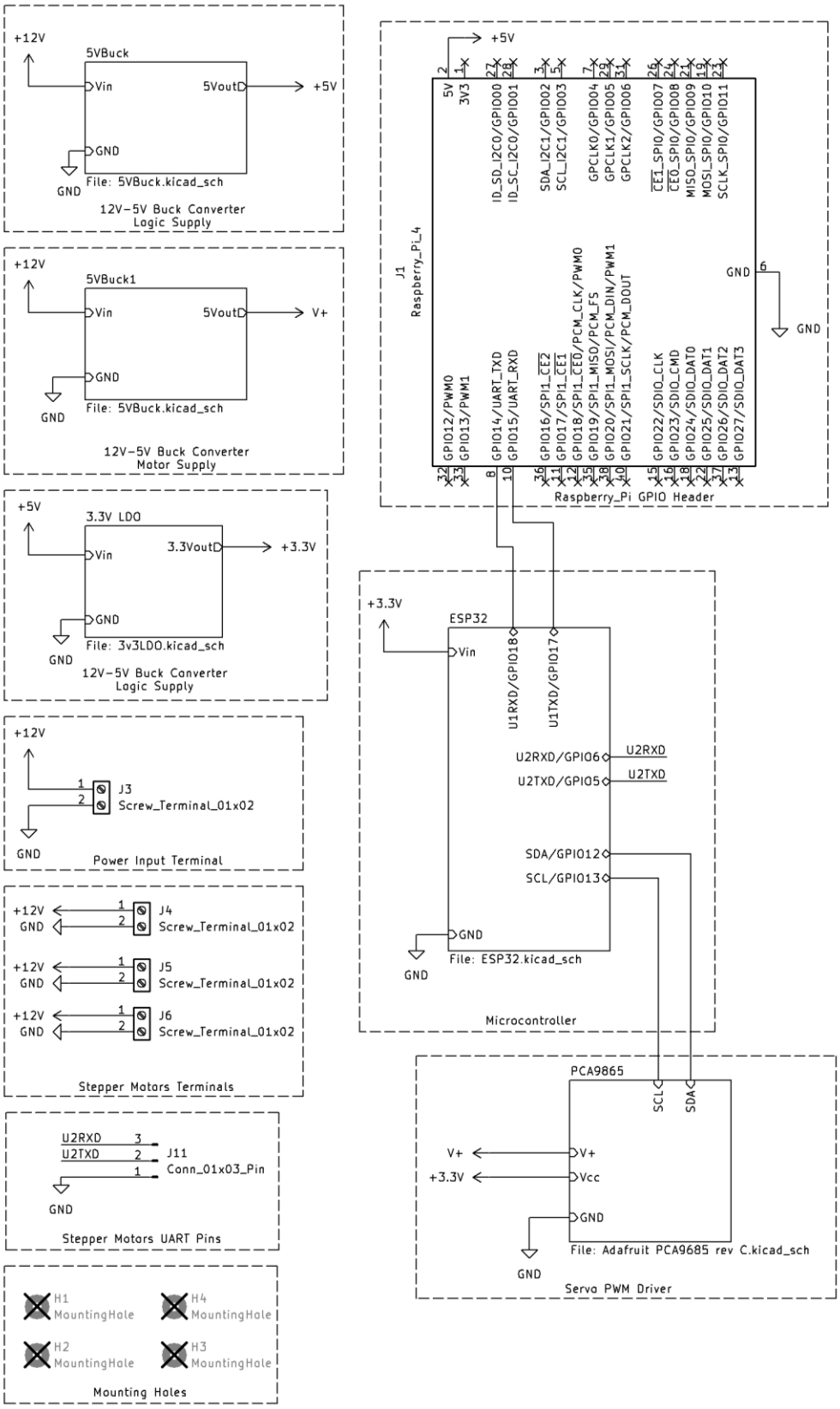


Figure 29: Main PCB Schematic

6.6 Optical Design

Designing a spectrometer is comprised of 5 optical components: slit, collimating lens, grating, focusing lens and an imaging sensor.

6.6.1 Design Overview

Starting with the choice of the sensor as it will give the designer the first constraint on the size of the pixel. The sensor is a TCD1304DG with a pixel size of 8µm by 200 µm with a pitch of 8 µm. It would be ideal to have a 1:1 imaging system and have a slit of 8µm. Employing a 1:1 imaging configuration in optical systems provides superior spatial resolution, enhances radiometric efficiency, minimizes optical aberrations, simplifies alignment procedures, and facilitates straightforward calibration, thus establishing it as a widely favored design approach. However, Thorlabs closest slit size was (10 ± 1) µm the S10HK. So, the system will need to have desired magnification of 0.8 to image the slit onto a single pixel. The only way to get the desired magnification is the ratio of focal length of the lens choice (6.6.1).

$$M_{Spec} = \frac{f_1}{f_2} = \frac{Collimating\ Lens}{Focusing\ Lens} \tag{6.6.1}$$

For my original Zemax design I was using only BK7 singlet lenes. This led to chromatic aberration that resulted in chromatic focal shifts that could not addressed with singlet lenses. Thus, achromat where need to correct for the chromatic focal shifts. Before diving into this next design iteration, I wanted to see what achromatic lenses where available on Thorlabs before designing a system where the parts could be sourced.

Fortunately, there was a set of achromatic lenses that could exactly match my desired magnification. The AC254-100-AB or focusing lens was an Achromatic doublet with a focal distance of 100mm and a diameter of 1 inch with an anti-reflectance coating for 400nm to 1100 nm. The AC508-080-AB or collimating lens was an Achromatic doublet with a focal distance of 80mm and a diameter of 2 inch with an anti-reflectance coating for 400nm to 1100 nm. Applying (6.6.1) with the two lenes above got the desired M_{Spec} of 0.8.

For the diffraction grating which already discussed in section 3.18 and analysis in Figure 18: $\Delta\lambda_{min}$ vs Dispersion Angle. Although additional analysis needs to be done to understand its dispersion angle based on incident angle and which incident angle is the most optimal. The optimal angle is the Littrow conditional meaning when operating at this Littrow configuration, the diffraction efficiency in the targeted diffraction order is maximized for the blaze wavelength.

6.6.2 Zemax Design

First, when setting up the aperture type I used object cone angle. The 0.9 multiplied by the R which is the radius of the lens is to adjust for a 90% clear aperture. R is 12.7 mm and L is 100 mm (6.6.2)

$$\theta_{cone} = \arctan\left(\frac{R \times 0.9}{L}\right) \cong 6.6 \tag{6.6.2}$$

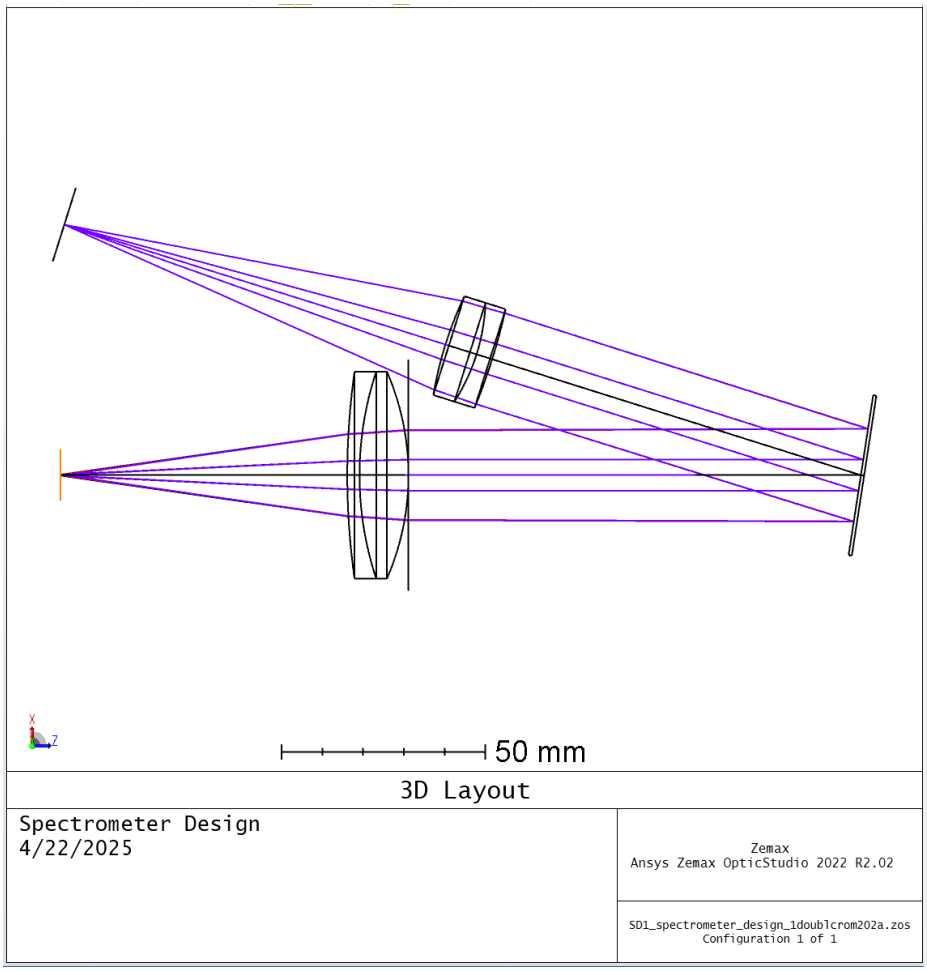
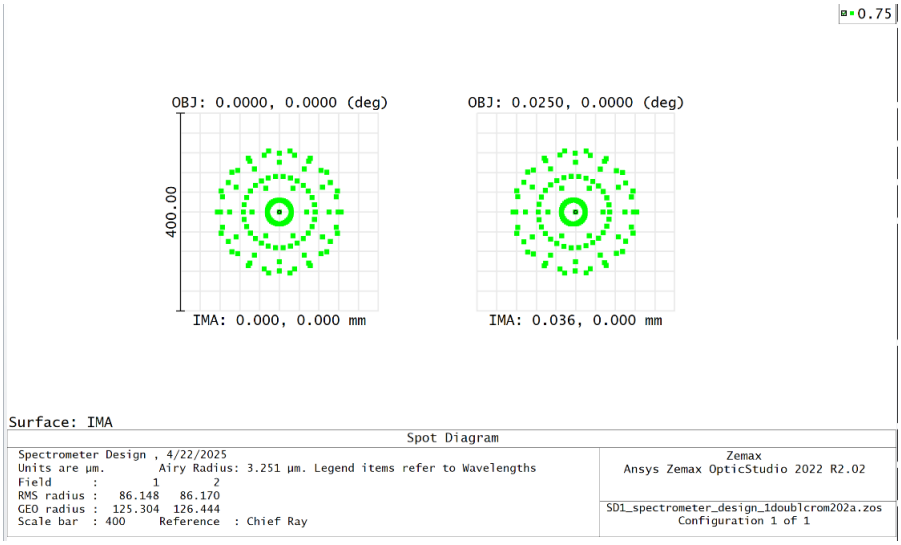


Figure 30: Lens Layout

After inputting all the lens data and optimizing the distance between the lenses we get the lens layout Figure 30. When simulating surface as a first order the dispersion angles don't alight with the MATLAB simulated dispersion angles



. So, I input an off-axis field angle to simulate the off-axis light that would be dispersed by the grating. The justification of choose 14.7 degrees for the simulation is it matches meets the Littrow condition for the grating.

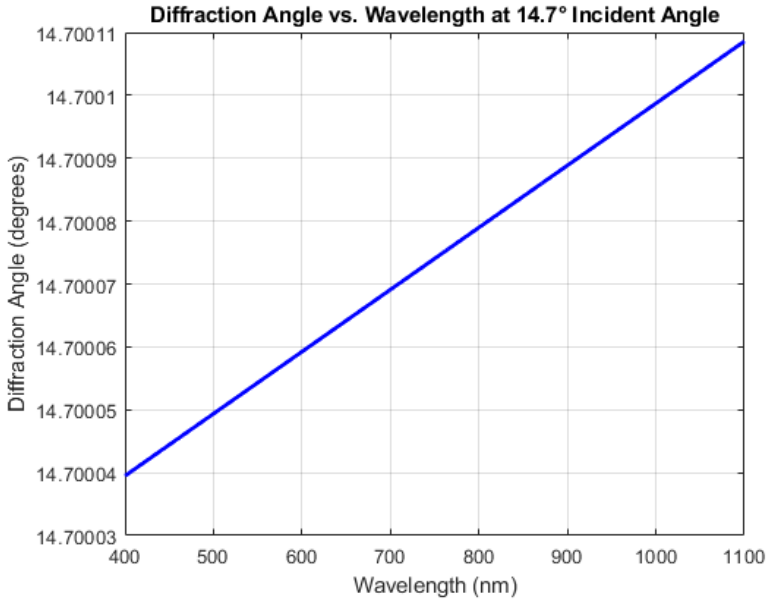


Figure 31: Diffraction Angle

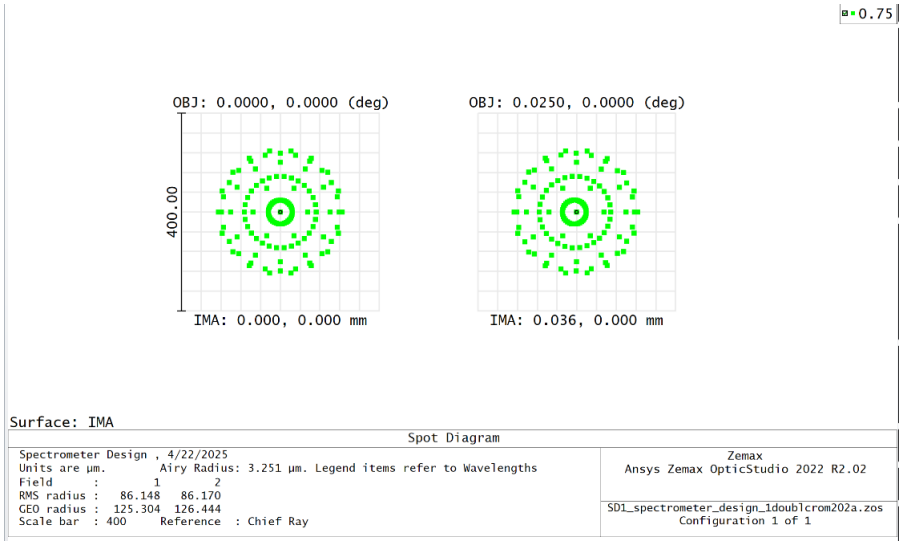


Figure 32: Spot Diagram

As expected, the spot diagrams show the spectrometer is aberration limited Figure 32. Being aberration-limited is not a flaw because if the airy disk is smaller than the pixel size of 8µm, light is dispersed over the length of the sensor and the sensor is sensitive enough to pick up the incident light. In the Final demo video, we verified that the sensor is sensitive enough and the light is dispersed over the sensor. Even though RMS radius is much larger than the airy disk it doesn't impact the performance of the system.

7.0 Software Design

The software design of the T.R.A.S.H. system focuses on modularity, real-time performance, and seamless integration across all major subsystems. The architecture is divided between the Raspberry Pi 5 and the STM32F401 microcontroller. The Raspberry Pi is responsible for running high-level processes such as object detection, inverse kinematics, material classification, and user interface management. In contrast, the STM32 handles time-sensitive operations including spectrometer control, motor actuation, and direct communication with low-level hardware.

The overall system was built to operate autonomously with minimal latency, supporting dynamic object recognition and sorting in a constrained environment. All subsystems were written with scalability in mind, allowing for the future integration of additional peripherals or sensors. The software leverages industry-standard communication protocols (I^2C, UART) and open-source libraries (OpenCV, Ultralytics, ROS2) to maintain compatibility, reduce development time, and allow for long-term maintainability.

7.1 Software Logic Overview

The control logic of the T.R.A.S.H. system is structured around a modular runtime sequence designed for continuous recycling operations. The system's behavior is governed by a repeatable control loop split between high-level perception and low-level actuation, executed on two independent but synchronized platforms: the Raspberry Pi 5 and the STM32F401 microcontroller. This dual-processing approach enables real-time responsiveness, parallel tasking, and efficient coordination between software logic and physical actuation.

The overall software flow of the T.R.A.S.H. system is illustrated in Figure 33. This diagram presents a high-level overview of how the system responds to detected objects, beginning with visual identification and ending with successful sorting. It highlights the interaction between the Raspberry Pi and STM32, showing how data flows through computer vision, trajectory planning, material classification, and motor control. The flowchart serves as a visual reference for the modular, event-driven nature of the runtime loop.

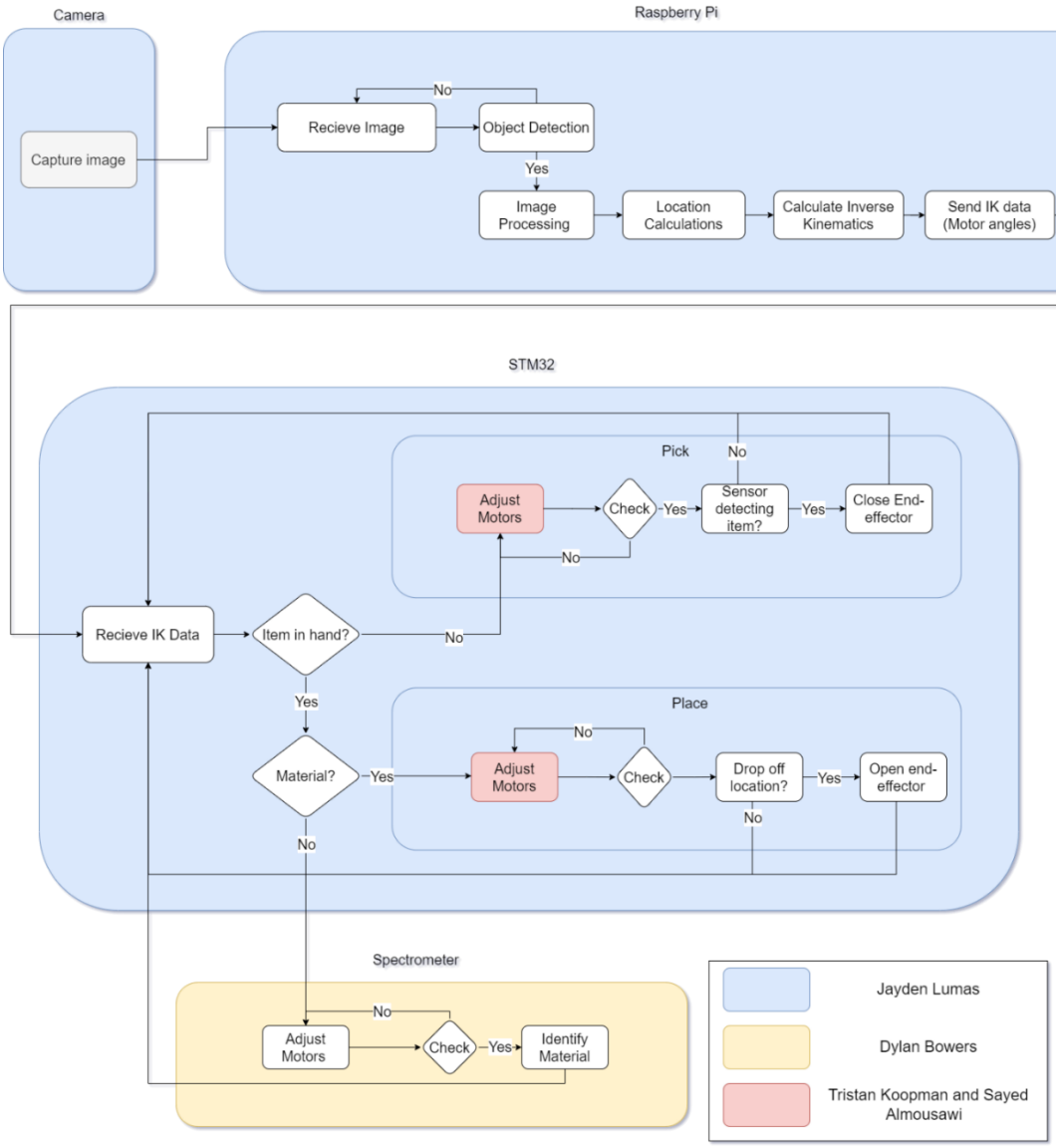


Figure 33: Software Design Logic Flowchart

At system startup, both processors initialize their respective subsystems. The Raspberry Pi loads the object detection model (YOLOv8) and prepares optional offloading to AWS Rekognition for enhanced object identification in low-confidence scenarios. The STM32 configures the spectrometer's clocking signals, activates the ADC and DMA for data acquisition, and initializes the motor control drivers. Once initialized, the software transitions into its primary operational loop.

A camera mounted onto the end-effector and above the workspace feeds image data into the Raspberry Pi. Detected objects are processed in real time using YOLOv8, with optional assistance from AWS to handle difficult lighting or crowded object arrangements. Both systems work in tandem to identify and track multiple objects simultaneously. Once an

object is confirmed, the Raspberry Pi transforms the object's pixel position into robot-relative coordinates using a predefined transformation matrix based on camera calibration.

This data is passed into the inverse kinematics solver to calculate the required joint angles. These angles are transmitted to the STM32, which executes the movement sequence to position the end-effector over the object. Once the arm is in position, the STM32 initiates spectrometer sampling and sends the resulting reflectance data back to the Raspberry Pi.

The Pi compares the spectrum against a database of known recyclable materials and identifies the object's material classification. The corresponding bin location is then selected, and a second set of joint angles is calculated and sent to the STM32 to complete the sorting operation.

System status, detection results, and classification outcomes are displayed in real time through the user interface. Manual overrides and safety checks are built into each stage, with error-handling routines in place to manage unexpected behaviors.

This control flow is continuously repeated while the system is active. Each stage includes conditional validation to ensure proper execution. The modular structure also supports subsystem upgrades and independent testing without requiring changes to the overall architecture.

7.2 Libraries and Tools

The software for the T.R.A.S.H. system is built using a hybrid programming model that combines high-level scripting for perception and planning with embedded development for real-time control. Each language and library was selected to meet the specific demands of its role in the system, supporting efficient development, robust communication, and seamless integration between components.

High-level logic runs on the Raspberry Pi 5 using Python 3. Python was selected for its fast development cycle, compatibility with robotics frameworks, and broad support for computer vision, machine learning, and hardware interfacing. It is used to implement the computer vision pipeline, inverse kinematics calculations, GUI logic, and ROS2 node communication. OpenCV is used for image processing, Ultralytics YOLOv8 is used for object detection and model training, and Tkinter (or a comparable tool) is used to build the graphical user interface.

Simulation and motion planning are handled using MoveIt2, Gazebo, and RViZ. MoveIt2 generates and validates motion plans using the robot's 6-DOF configuration. Gazebo simulates the robotic arm in a physics-based environment, allowing the team to test

grasping, classification, and sorting routines before deployment. RViZ is used to visualize trajectories, object positions, and system feedback during simulation and live operation.

The STM32F401 microcontroller is programmed in C using the STM32 Hardware Abstraction Layer (HAL). It is responsible for executing motor control, generating spectrometer signals, and managing real-time responses. The microcontroller communicates with the Raspberry Pi using UART and receives commands encoded in a lightweight custom protocol. To connect with the ROS2 network, the STM32 runs Micro-ROS, which allows it to subscribe to trajectory messages and publish system status updates. This connection enables the embedded system to operate as a native ROS2 node, while still maintaining the low-latency, deterministic behavior required for hardware control.

At the core of the entire system is ROS2 Jazzy, the middleware responsible for managing communication between all subsystems. ROS2 introduces several key improvements over ROS1, including native support for real-time data exchange, distributed processing, and more reliable communication. It uses a modular publish-subscribe architecture, where software components (nodes) can share data through topics, services, and actions. In T.R.A.S.H., ROS2 runs on both the Raspberry Pi and the STM32, ensuring full system automation, reliable coordination, and centralized logging. This architecture improves fault isolation, scalability, and responsiveness, making the overall system more robust and maintainable than its ROS1-based counterparts.

7.3 Subsystems

The software stack is divided into several subsystems, each responsible for a specific function within the sorting operation. These components are designed to operate asynchronously and communicate through standardized interfaces. This approach reduces interdependencies between modules and ensures that failures in one component do not propagate throughout the system.

7.3.1 Computer Vision

The computer vision subsystem serves as the primary method for identifying, localizing, and tracking recyclable items within the workspace. This module is essential to initiating the sorting pipeline, as all subsequent motion planning, grasping, and classification depends on accurate object detection and position estimation.

The main detection framework is built around YOLOv8, which is deployed locally on the Raspberry Pi 5. Unlike pre-trained models or limited cloud-based services, the project team plans to train a custom YOLOv8 model using a dataset specifically curated for the types of recyclable materials expected in the system. This approach allows for detection performance to be optimized based on the unique size, texture, shape, and labeling conditions of aluminum cans, PET bottles, cardboard, and other selected targets. The team

will use the YOLOv8 training pipeline as a tool to handle data augmentation, labeling, model tuning, and evaluation.

Although Amazon Web Services (AWS) Rekognition was originally explored as a detection backend, its object recognition capability is limited to general-purpose classes and lacks the fine-grained specificity required for this application. As a result, AWS Rekognition is retained in the system solely as a fallback for redundancy. If YOLOv8 returns a low-confidence detection or experiences inference failure, cropped object regions may be offloaded to AWS to improve robustness in uncertain scenes.

Feature	YOLOv8	AWS Rekognition
Deployment Location	On-Device	Cloud-based(AWS Server)
Latency	Low (Local Hardware)	High (Network Speed)
Model Customization	Fully Trainable	Limited to AWS object classes
Detection Flexibility	Tunable for specific shapes and objects	General purpose, lacks tuneability
Scalability	Easily re-trainable	Only supports label matching

Table 33: Computer Vision Tools

The camera system includes two views: a primary camera mounted on the end effector and a secondary overhead camera. The end-effector camera is used for close-range detection, alignment, and real-time visual feedback as the object is approached. It allows the system to refine object position, confirm proximity before pickup, and assist with final grasp alignment. The overhead camera provides a wide-angle view of the workspace, allowing for initial detection and general localization before the arm is moved into position.

Detected bounding boxes are converted to physical coordinates using a calibrated transformation matrix and then passed to the inverse kinematics module for targeting. The entire detection stack runs as a ROS2 node, communicating its outputs to downstream subsystems including trajectory planning and GUI display.

The flexibility of YOLOv8's architecture allows the team to continuously retrain and fine-tune the model as more real-world training data becomes available during prototyping and deployment phases, ensuring high adaptability and continued improvement in detection accuracy.

7.3.2 Inverse Kinematics and Simulations

The Inverse Kinematics (IK) and simulation subsystem is responsible for computing the joint angles required for the robotic arm to reach a specific target in space and validating that motion through a virtual test environment before execution on hardware.

The T.R.A.S.H. arm is a 6-DOF system, which requires a robust and flexible IK framework. Rather than relying on custom geometry solvers alone, the team uses ROS2 Jazzy as the middleware platform for simulation and trajectory validation. This allows for the integration of high-level planning and low-level motion logic using a well-documented and community-supported robotics stack.

The IK solution and motion planning are implemented using MoveIt2, a powerful tool designed for robotic trajectory generation. MoveIt2 enables the robotic arm to calculate motion plans that obey joint limits, avoid collisions, and reach specific Cartesian targets with millimeter-level accuracy. The framework provides configuration tools and planners that help define the robot's model, kinematics, and motion constraints. These configurations are used to validate collision-free trajectories before any physical commands are sent to the arm.

For visualization and motion verification, RViZ is used to display the robot’s real-time pose, target positions, and detected objects within the workspace. To model the system in a realistic environment, Gazebo provides a physics-based simulation that includes the arm, workspace, and surrounding objects. These tools work together to allow complete system validation in software before testing on actual hardware.

Before any motion plan is executed, the system first simulates the movement in Gazebo and checks its validity through MoveIt2. If the planned trajectory is unreachable, exceeds joint limits, or results in a potential collision, the plan is rejected and flagged in the system logs. This safeguard prevents unsafe or undefined arm behavior.

By using simulation during development and deployment, the team can identify design flaws early, test performance under controlled conditions, and confidently implement new features. This process also supports future scalability, as the simulation environment can be easily updated to reflect design changes or new modules.

To manage these motion sequences in a structured and traceable way, the inverse kinematics subsystem follows a defined state machine. Each state represents a specific behavior or decision point for the robotic arm, ensuring the system transitions logically from detection to pickup, classification, and sorting. This structure provides clarity, safety, and a basis for debugging or scaling the motion logic.

State	Action	Trigger to Next State	Next State
Idle	Arm moves to high observation point and looks down at workspace	System startup or return from last cycle	Searching
Searching	System waits for object detection from vision module	Object detected by YOLOv8	Object Detected
Object Detected	Convert image coordinates to robot frame, compute IK with MoveIt2	Valid trajectory generated	Moving to Pick
Moving to Pick	Arm moves to object location	Arm reaches pick coordinates	Picking Up
Picking Up	End effector closes to grasp object	Grasp confirmed (timeout or feedback)	Material Classifying
Material Classifying	Arm moves above spectrometer, STM32 samples, Pi classifies material	Classification result received	Sorting
Sorting	Arm moves to appropriate bin based on material type	Arm reaches bin location	Letting Go
Letting Go	End effector opens to release object	Object released and arm returns to idle position	Idle

Table 34: State Machine for Inverse Kinematics and Simulation Subsystem

7.3.3 Spectrometer

The spectrometer subsystem is responsible for identifying the material composition of objects after they have been picked up by the robotic arm. This process is essential to the classification logic of the T.R.A.S.H. system, as it determines the final bin assignment for each item. The spectrometer operates using a reflection-based optical system controlled by the STM32F401 microcontroller.

The STM32 generates the precise clocking signals required to drive the TCD1304DG linear CCD sensor, which captures light intensity values reflected from the object surface. These signals are produced using internal timers and controlled with hardware-level accuracy. The sensor’s analog output is sampled using the STM32's ADC and stored through DMA for minimal CPU overhead, allowing efficient and synchronized signal acquisition.

Once sampling is complete, the STM32 transmits the collected data to the Raspberry Pi using UART. The spectral data is structured as a sequence of digitized intensity values representing the reflected light signature. On the Pi, this data is received by a dedicated ROS2 node responsible for spectral classification. The classification process compares the incoming spectrum against a library of preloaded reference profiles, each corresponding to a known recyclable material such as aluminum, PET, HDPE, or cardboard.

Similarity scores are computed using a vector-based comparison method, and the highest-confidence match is selected as the material label. If the confidence is below a threshold or the spectrum appears inconclusive, the object is flagged for reprocessing or manual review. Valid classifications are stored in the system log and passed to the motion planner to determine the appropriate sorting trajectory.

The STM32 operates as a Micro-ROS node, enabling it to publish sampling status and timing metrics to the ROS2 system. This allows other nodes, including the GUI and data logger, to receive real-time updates on classification results and spectrometer performance. All classification events are timestamped and indexed in the GUI's settings panel, allowing operators to review historical performance and confirm correct sorting behavior.

This subsystem integrates low-level hardware control with high-level decision-making, and its design ensures that sampling is accurate, efficient, and traceable across the full system pipeline.

7.3.4 Motor Control

The motor control subsystem is managed entirely by the STM32F401 microcontroller. This subsystem is responsible for executing joint angle commands received from the Raspberry Pi and directly controlling the motion of the robotic arm.

Communication between the Pi and STM32 is established through a custom UART protocol. Unlike off-the-shelf ROS-compatible motor drivers, the team manually developed an embedded firmware layer on the STM32 to receive joint angle targets and handle actuation logic. When the Raspberry Pi sends a command containing desired joint positions, the STM32 interprets the data and initiates rotation of the respective joints.

Servo motors are used for each joint, and one key advantage of this configuration is that PWM generation is already integrated into the servo hardware. As a result, the STM32 does not need to generate or tune raw PWM signals at the software level, simplifying the low-level control logic. Instead, the microcontroller focuses on parsing incoming commands, validating them, and issuing angle updates to the servos based on the embedded protocol.

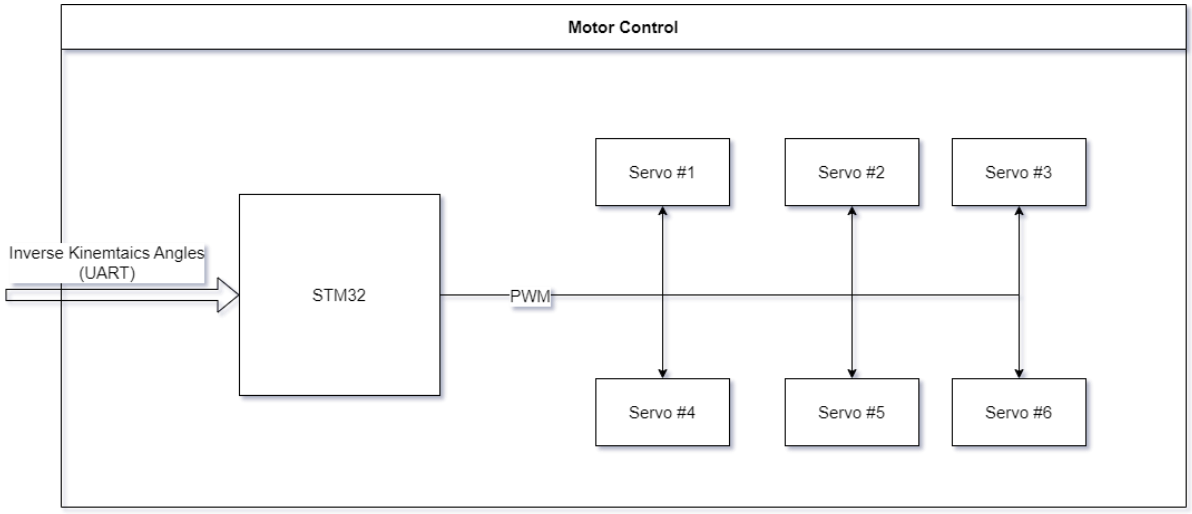


Figure 34: Simple Motor Control Block Diagram

This approach gives the team full control over motor behavior while maintaining a lightweight and responsive firmware footprint. It also enables direct debugging, real-time system feedback, and tighter synchronization with upstream planning nodes. Any deviation from expected behavior, such as a joint not reaching its target angle, is flagged and logged through serial communication.

7.3.5 User Interface

The user interface for T.R.A.S.H. is designed to always provide operators with complete control and visibility in the robotic system. Hosted on the Raspberry Pi, the GUI (Figure 35) offers an intuitive layout that allows users to initiate system operations, monitor performance in real time, and interact with key hardware subsystems.

The interface is structured around four primary elements: a D-pad for manual arm positioning, an open/close toggle for end effector control, a system power button, and a dedicated settings panel. Each of these elements connects directly to the underlying ROS2 nodes and hardware interfaces to ensure seamless real-time execution of user commands.

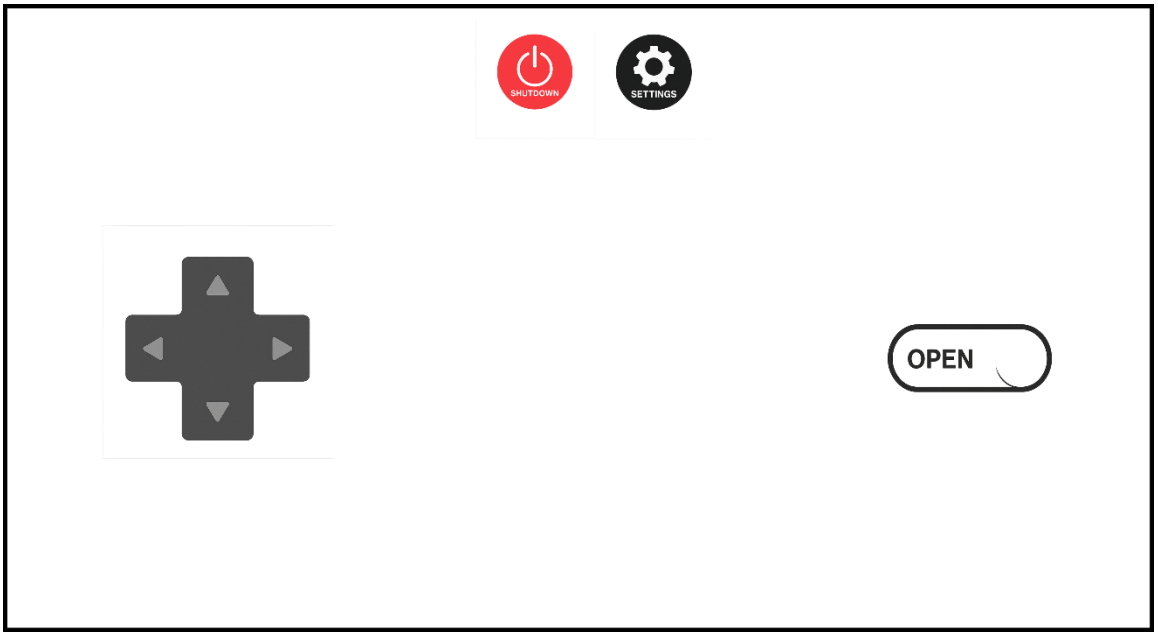


Figure 35: User Interface GUI Layout

The D-pad enables manual override of the robotic arm's position by sending incremental coordinate shifts to the STM32. This allows the user to reposition the arm precisely, especially useful for calibration or fine-tuned corrections. The open/close toggle allows direct control of the end effector, giving the user the ability to pick up or release an object on command. These manual controls are especially valuable during system validation, troubleshooting, or when automatic detection sequences need human oversight.

The power button, located at the top of the interface, provides a clean and reliable method for shutting down the system safely. This helps prevent data loss or unintended motor behavior and ensures that all critical processes are terminated in the correct sequence.

A standout feature of the interface is its built-in error handling and notification system. If any fault or unexpected condition is detected during operation, such as a missed detection, communication timeout, or invalid motor state, a real-time popup alert is displayed. These alerts provide clear descriptions of the issue and suggest the next steps or recovery options. This feature allows users to quickly diagnose and resolve problems without needing to access backend logs or restart the system blindly.

The settings panel offers deeper insight into system performance and sorting analytics. Within this panel, users can access a full record of recent system activity, including:

- The number of items detected and picked up

- Material classifications for each object processed
- Bin assignment history
- Timestamped logs of sorting cycles
- System warnings or error flags

This data can be used for auditing, performance tuning, and validating the system's classification accuracy. It also gives the operator the ability to track system efficiency and material throughput over time.

Designed for mouse and keyboard interaction, the GUI is lightweight, responsive, and continuously updated by underlying ROS2 nodes. It serves as a critical layer between the user and the autonomous pipeline, balancing automation with manual oversight and enabling direct engagement with every stage of the sorting process.

7.3.6 Communications

The communication architecture of the T.R.A.S.H. system is built around a layered structure that links high-level perception and planning modules with low-level motor control and sensor feedback. This design allows each subsystem to operate independently while maintaining synchronized behavior through well-defined communication interfaces.

At the hardware level, all communication between the Raspberry Pi 5 and the STM32F401 microcontroller is handled using UART. A custom serial protocol was developed to transmit joint angle commands, classification results, and spectrometer data. On the STM32 side, the team implemented their own embedded system that directly receives these commands, parses them, and actuates the appropriate servo motors. These servo motors benefit from built-in PWM control, which reduces the need for additional signal generation logic. The STM32 continuously monitors the incoming data stream and provides confirmation or error messages back to the Pi for logging and recovery.

I2C is used within the STM32 for communication with onboard peripherals such as ADCs or expanders, while GPIO is reserved for simple hardware toggling and low-latency signaling. Each channel includes error detection, timeouts, and safety checks to prevent unintended execution in the event of dropped or corrupted packets.

At the system level, ROS2 Jazzy is used on the Raspberry Pi as the central middleware framework. It manages communication between all high-level nodes, including the computer vision pipeline, inverse kinematics planner, GUI system, and range-sensing modules. These components are separated into ROS2 nodes such as EE_Camera_Node, Birds_Eye_Camera_Node, Range_Node, IK_Node, Gui_Node, and Controller_Node.

Messages are exchanged using topics and services that provide reliable, asynchronous communication between components. (Figure 36)

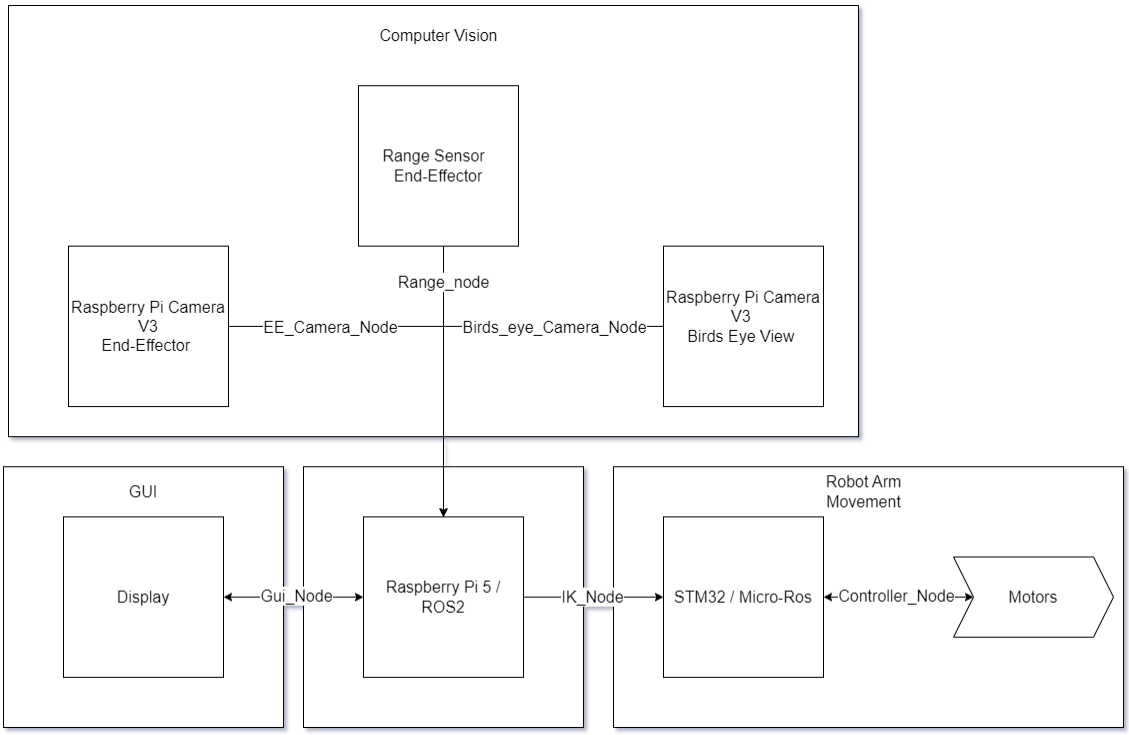


Figure 36: ROS2 – Based System Communication

To bridge the STM32 with the ROS2 ecosystem, Micro-ROS is used on the embedded side. This allows the STM32 to act as a ROS2 node (Controller_Node) capable of subscribing to trajectory messages and publishing state updates. Micro-ROS enables the same modular communication flow used in high-level planning to be extended into the embedded layer, unifying the control structure under a consistent architecture.

All motion plans are calculated and verified in simulation using MoveIt2 and Gazebo before being sent to hardware. Visual outputs, system feedback, and log information are displayed through the GUI and can be visualized in real time using RViZ. The complete system flow is shown in Figure 33, detailing the communication pathways between the Raspberry Pi, ROS2 nodes, STM32, and motors.

7.4 Summary

The software system for T.R.A.S.H. was developed to support real-time performance, modularity, and full integration between high-level perception logic and low-level

hardware execution. Each core subsystem—computer vision, inverse kinematics, motion control, spectrometer classification, user interface, and communication—was designed with clear responsibilities and built using tools optimized for those roles.

ROS2 Jazzy serves as the foundation for all system-level communication and modular node development, while Micro-ROS bridges the embedded STM32 firmware into the ROS2 ecosystem. Python was chosen for rapid development and compatibility with tools such as YOLOv8, MoveIt2, OpenCV, RViZ, and Gazebo. C was used on the embedded side for performance and direct hardware control.

By combining real-time UART communication, robust simulation frameworks, and intuitive user control, the system achieves a balance between automation and human oversight. The software design not only supports the current sorting workflow but also establishes a flexible foundation for future improvements in detection, classification, control, and scalability.

8.0 Structure Design

In this section, we discuss the structural design and limitations of T.R.A.S.H.

8.1 Linkages

For a robotic arm, the linkages serve multiple purposes. The most obvious function of the linkages is to serve as an intermediary offset between joints. This offset dictates the available workspace within which the robot is able to work. Linkages are studied based on their geometry, and as such, links are considered to be rigid [28]. These rigid links are additionally the transmission system of power from the robot actuators. Therefore, the linkages are required to be load-bearing devices. Finally, it is important to note that mechanical linkages are used to transform a given input force and motion into a desired output force and motion. The ratio of the output force to the input force is known as the mechanical advantage of the linkage, and while the ratio of the input speed to the output speed is known as the speed ratio, the speed ratio and mechanical advantage are defined as yield the same number in an ideal linkage [28].

When analyzing the force acting on our robot arm, we realize that the only stress acting upon the linkages is bending, as there are rare instances when the robot's arms will experience high compression, tension, or torsional stress. However, by analyzing Figure 6 Free Body Diagram, we see that bending stress is a crucial case we need to consider. Cost is the final consideration, as seen in Table 35: Concept Alternatives – Material Selection.

		Concept Alternatives							
		Aluminum		Carbon Fiber		Steel		PETG	
Criteria	Importance Weight %	Rating	Weighted Rating	Rating	Weighted Rating	Rating	Weighted Rating	Rating	Weighted Rating
Light Weight	35%	2	0.70	4	1.40	2	0.70	4	1.40
Low cost	25%	4	1.00	1	0.25	3	0.75	4	1.00
High Bending Stiffness	20%	2	0.40	3	0.60	4	0.80	1	0.20
High Bending Strength	20%	2	0.40	4	0.80	3	0.60	1	0.20
Totals	100%		2.50		3.05		2.85		2.80

Rating	Value
Unsatisfactory	0
Just Tolerable	1
Adequate	2
Good	3
Very Good	4

1. Importance weights assigned
2. Team rates each concept 0 to 4
3. Multiply rating X importance
4. Sum weighted ratings

Table 35: Concept Alternatives – Material Selection

As seen from the Table above, based on the Importance Weight given the previously mentioned criteria, Carbon Fiber material will be used for the robot arm linkages.

8.1.1 Linkage Cross-Section

Now that we have determined Carbon Fiber is the best choice for material, the next question is what cross-section will be best for our robot arm. Because carbon fiber is very costly to get custom-shaped, the only options being considered are cross sections that are most commonly produced. These cross-sections are rods, tubes, and plates/brackets.

		Concept Alternatives					
		Rod		Plates		Tube	
Criteria	Importance Weight %	Rating	Weighted Rating	Rating	Weighted Rating	Rating	Weighted Rating
Hardware	30%	4	1.20	2	0.60	3	0.90
Volume	25%	2	0.50	4	1.00	4	1.00
Moularity	25%	4	1.00	2	0.50	4	1.00
Joint Interface	20%	3	0.60	4	0.80	3	0.60
Totals	100%		3.30		2.90		3.50

Rating	Value
Unsatisfactory	0
Just Tolerable	1
Adequate	2
Good	3
Very Good	4

1. Importance weights assigned
2. Team rates each concept 0 to 4
3. Multiply rating X importance
4. Sum weighted ratings

Table 36: Concept Alternatives – Linkage Cross-Section

8.1.1.1 Rod

Rods are very commonly manufactured; however, because of their manufacturing method and increased volume, they are much more expensive compared to other cross-sections like plates or tubes. The circular design does allow for cusps to be screwed onto the outside of the rod. Because these cusps are friction fit, it also allows for easy swapping of different modules. The joint interface of the linkage is not ideal. Adhesive would be the best method for securing the linkage to the joint adaptor. However, this would be permanent and would not allow for a redesign without the need to purchase more rods. Therefore, friction-fit adaptors will be needed in order to allow retrofits.

8.1.1.2 Plates and Brackets

Plates are much easier to manufacture and, therefore, much cheaper. They can also be manufactured very thin and, therefore, have a much lower volume compared to rods. The commonly manufactured brackets also provide a very simple and replaceable method for connecting to the joint interface. However, for modularity, in order to add mounts for cables or motors, adaptors will need to be bolted in the plates. This not only degrades the overall strength of the linkage but requires hardware, adding additional weight to the robot arm. Additional hardware is to be avoided.

8.1.1.3 Tubes

Tubes provide the same benefits as rods with the reduced volume. Tubes are another very commonly manufactured structure and have a much more customizable surface area without additional weight or unneeded strength. It is for these reasons that the linkages will be made using Carbon Fiber tubes as the base structure.

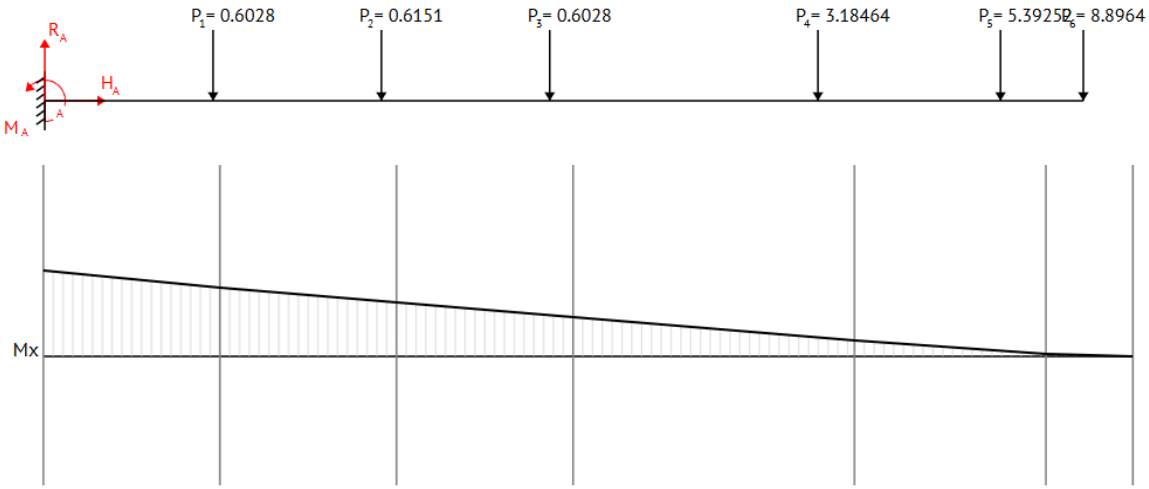
8.1.2 Internal Loadings

Based on Table 36: Concept Alternatives – Linkage Cross-Section we find that for our linkage cross-section, the tube is the best choice. In this section, we will find the location of the highest bending moment and use this value to calculate the maximum bending stress.

The equation for calculating the maximum bending stress shown in ((8.1.1) where M is the maximum bending moment, c is the perpendicular distance from the neutral axis, and I is the area moment of inertia.

$$\sigma_b = \frac{Mc}{I} \tag{8.1.1}$$

Because the robot arm can be considered a fixed cantilever beam, we know that the maximum bending moment is at the base.



Therefore, we will use the maximum expected torque output of our motor with the expected gear reduction ratio as the maximum bending moment.

$$M = 590Nmm * 15$$

After finding the maximum bending moment, the next step is to calculate the area moment of inertia. The area moment of inertia is the distribution of the cross-section as it is distributed across the bending axis.

$$I_y = \int y^2 dA \tag{8.1.2}$$

Where y is the distance from the reference axis. For a hollow circle or pipe, the equation for the area moment of inertia is

$$I_y = \frac{\pi}{4}(r_o^4 - r_i^4) \tag{8.1.3}$$

Using Rockwest Composites T-RND-472-L12 pultruded uni carbon fiber tube, we get the following values,

$$I_y = \frac{\pi}{4}((12mm)^4 - (9.8mm)^4)$$

$$I_y = 9{,}041.75mm^4$$

Using the bending moment, area moment of inertia calculated in the previous steps, we will complete (8.1.3)

$$\sigma_b = \frac{(8850Nmm)(6mm)}{(9{,}041.75mm^4)}$$

$$\sigma_b = 5.873MPa$$

Due to carbon fiber strength calculations being to complex, a generous assumption of 145 GPa will be used. As we can clearly see, the amount of bending stress that can possibly be applied to the carbon fiber tubes is nowhere near approaching the maximum strength of the tubes, and therefore, is not a concern.

8.2 Joints

The joints of the robot arm are the sources of force and motion. Whether direct or indirect drive, the joints of the articulated robot create the compounded resultant motion of the end effector. Along with the transmission of motion, the joints also serve as the connection point between the mechanical linkage chain. These joints can house electronics like motors or cables or speed reducers like gearboxes or belt drives.

8.2.1 Linkage Coupling

For our robotic arm we want to reduce the amount of man hours required for modifications and repairs. Therefore, it is imperative that all pieces and components are not only affordable, but also easily replaceable. Practically, what this means is that for binding the linkages to the joints will want to avoid any adhesives and other means of bonding that is irreversible or can cause damage to the components. For this reason, friction fit designs, or screwed assemblies are to be sought after.

Carbon fiber is a very resilient material. However, it can become severely weakened when drilled or cut due to its unique structural properties [29]

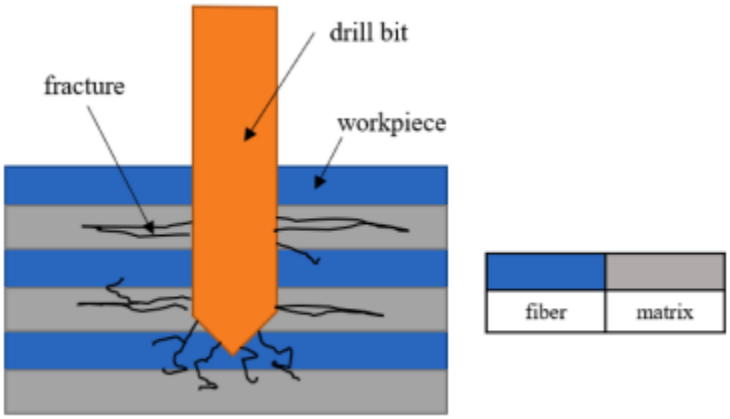


Figure 37: Drilling of CFRP composite [29]

Additionally, while carbon fiber dust is not toxic, when machining carbon fiber the dust is so fine and can cause major irritation if in contact with eyes or inhaled. Therefore, to bypass these issues, all bonding to carbon fiber will additionally avoid requiring machining.

The current design for bonding the linkages with the rest of the mechanical structure is seen in Figure 38. The design utilizes the structural property of carbon fiber resistance to

deformation. A single untapped M3 screw will be used to pinch the carbon fiber rod in place.

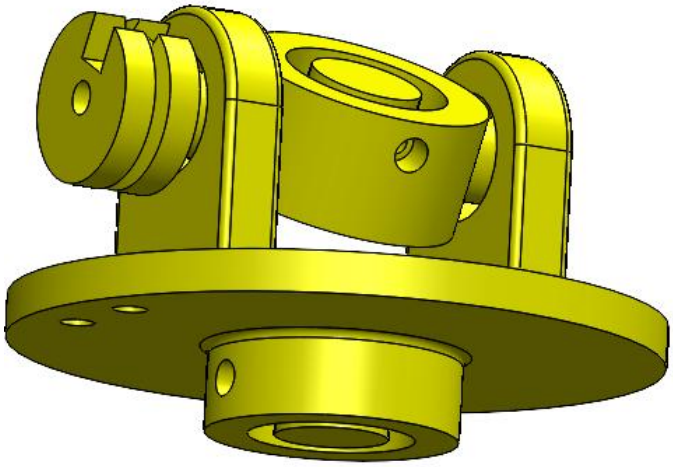


Figure 38: Joint Assembly SolidWorks

8.2.2 Speed Reducer (with transmission)

As discussed in Speed Reducer the Cycloidal Drive is the best choice when it comes to precision and control in a speed reducer. In the realm of robotics, precision and accuracy are critical, therefore, cycloidal drive is the obvious choice for increasing motor torque. An aspect that will need to be considered, as it is intended for the cycloidal drives integrated into the base of the robot arm, if there are inconsistencies in the manufacturing of the cycloidal drive there is a risk of vibration due to the nature of the cycloidal drive requiring the cycloidal disks being offset from the center axis of the speed reducer. To counteract this imbalance, cycloidal drives are designed with at least two cycloidal disks offset 180deg to each other. Vibration in the base of the arm where the cycloidal drives are integrated will have cascading effects on the other motors further down the robotic arm, creating a risk for control management.

Regarding transmission, most industrial robots use direct driven joints with the motors installed superimposed onto the joints they are providing motion to. This however, puts heavy pieces of hardware on the lever arm creating additional moment requiring increasingly stronger and more expensive motors the further from the end effector. Not only does this needlessly increase the price, but also the weight of the robot, causing it to be increasingly more dangerous to work around. For this reason, we will be actuating the individual joints away from the joint themselves transmitting the power from a distance.

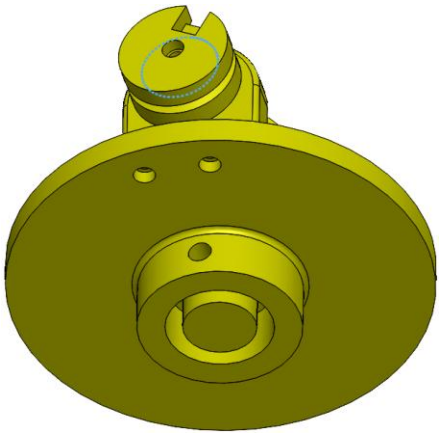


Figure 39: Joint Assembly Tendon Path

In the design shown above PTFE tubes will be used to route the internal tendons to the joints from the cycloidal drives to the joint assembly base. The tendons themselves will be made from UHMWPE Micro Cord due to its extremely low creep value.

8.3 Differential Wrist

The spherical wrist for the precision and flexibility of a robot arm. The wrist of a robot arm provides fine-tuned motion to position the end effector and payload in the correct position and orientation with little error. By necessity, the wrist system of the robot also has load-bearing capabilities. The spherical wrist allows rotation in all directions, making them suitable for tasks requiring flexible orientation of the end-effector [30]. Our robot's application requires that it can pick items that may be in less-than-ideal orientations. Therefore, it is critical that the robot end effector can be orientated in any direction.

8.3.1 Motion

As a part of our projects emphasis on weight mitigation the design of our wrist will use a differential drive as the governing design. A differential wrist allows for compact motion reducing the moment arm otherwise created by having motors in series.

Figure 40: Differential Wrist Design [31]

The differential wrist is comprised of two bevel gears mirrored across the center line of the linkage. Both bevel gears are connected to the output bevel gear as well as a coupler that secures the output bevel gear in place. When both motors are spinning in the same direction it creates rotation on the output bevel gear. When both motors are spinning in opposite directions it creates rotation about the axis that passes through the center of the input bevel gears offset by the bevel coupler.

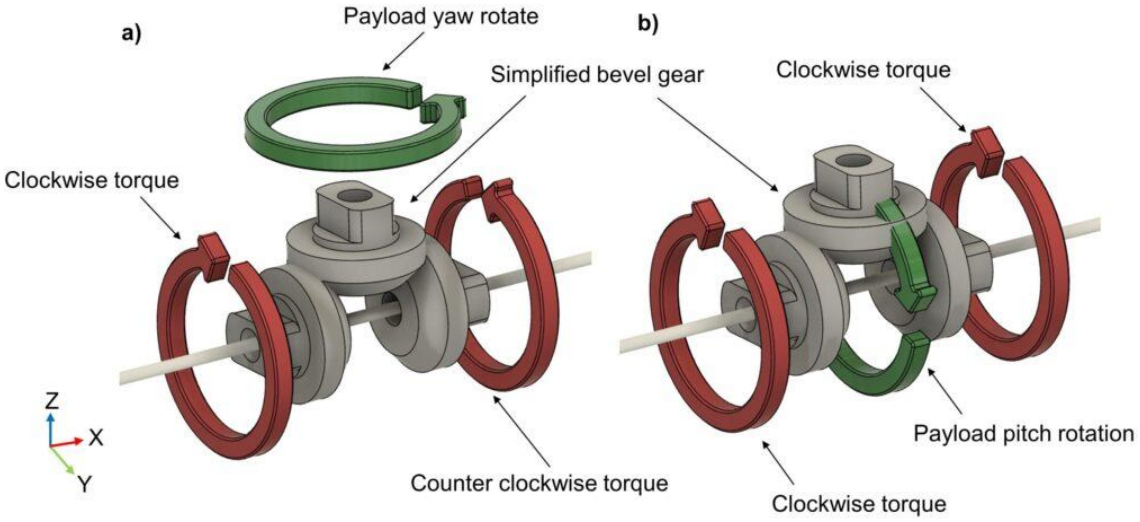


Figure 41:Bevel gear motion [32]

9.0 System Fabrication/Prototype Construction

9.1 PCB Design and Layout

As shown in the schematic diagrams section, the hardware architecture for this project is divided across two PCBs developed using KiCad. A main control and power board and a secondary spectrometer sensor board. This layout improves system flexibility, simplifies debugging, and allows for cleaner separation between main controls and the optoelectronic spectrometer subsystem. Which can be very sensitive to noise.

9.1.1 Main PCB

The main PCB is the central control and power distribution board for our project. It's built around an ESP32-S3 microcontroller, which handles real-time control of both stepper and servo motors. The ESP32 communicates with the MKS Servo42C stepper drivers using UART and controls the PCA9685 servo driver over I2C. It also connects to the Raspberry Pi through UART for system-level coordination. To keep wiring organized, all motor connections go through clearly labeled terminals. The board will power all major components from the included 12V input terminal, along with 5V buck converters and a 3.3V LDO regulator. A 2×20 GPIO header is included to connect the main PCB to the Raspberry Pi using a ribbon cable, making integration easy and clean.

From a layout perspective, the board was carefully designed with electrical performance and practical usability in mind. Decoupling capacitors were placed close to each major IC and voltage rail to filter out noise and handle transient power demands. Additional headers and test LEDs were added throughout the board to support debugging and provide access to MCU programming. Power circuitry and regulators are intentionally placed away from the ESP32 to reduce high-frequency noise coupling into logic and communication lines. The Raspberry Pi header is placed along the board's edge to minimize cable length and maintain a compact form factor. The current version of the PCB layout, shown in the figure below, serves as a functional prototype and can be refined and improved in Senior Design II based on testing and system performance.

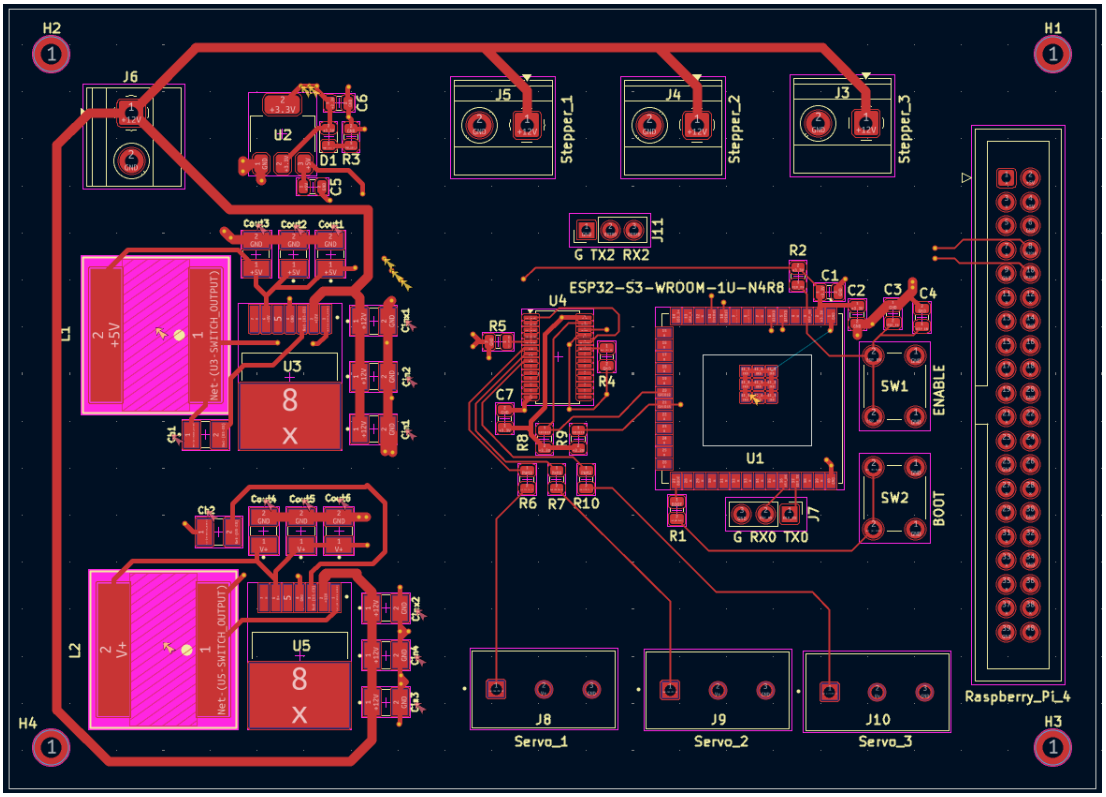


Figure 42: Main Board Layout (Top)

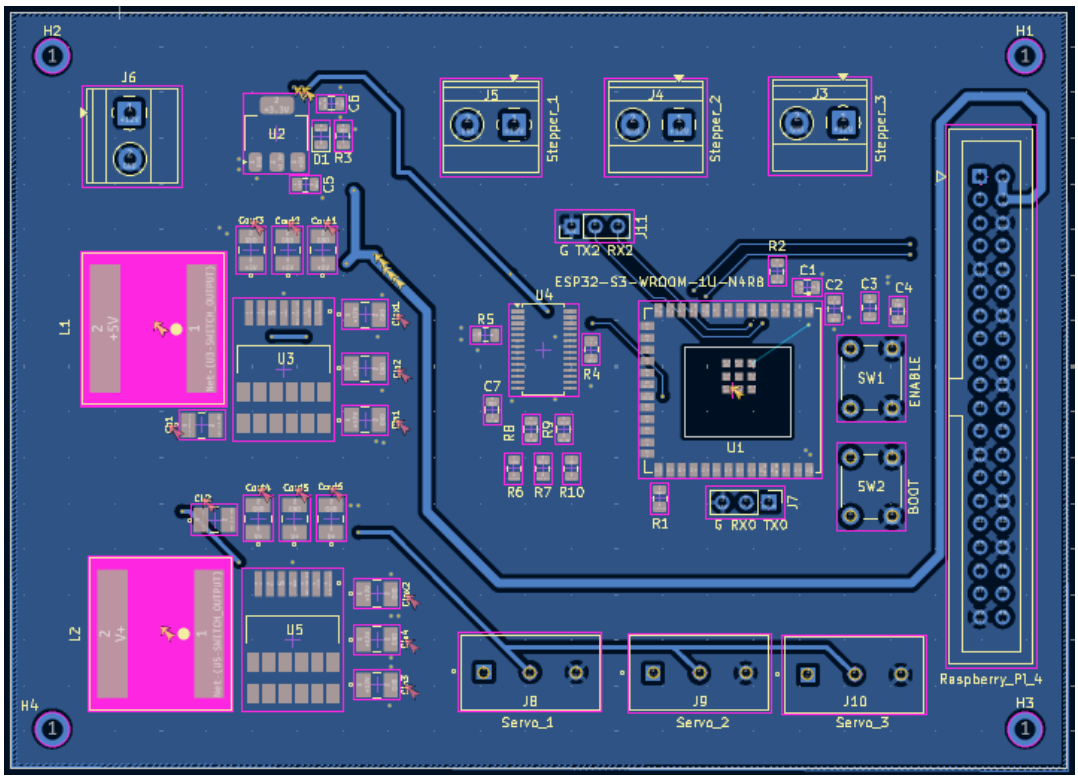


Figure 43: Main Board Layout (Bottom)

KiCad also includes a 3D viewer that makes it easier to see how the PCB will look once it's fully assembled. This feature helps with checking component placement and overall layout, and it's useful for spotting any design issues early. The figures below show a 3D view of the main PCB.

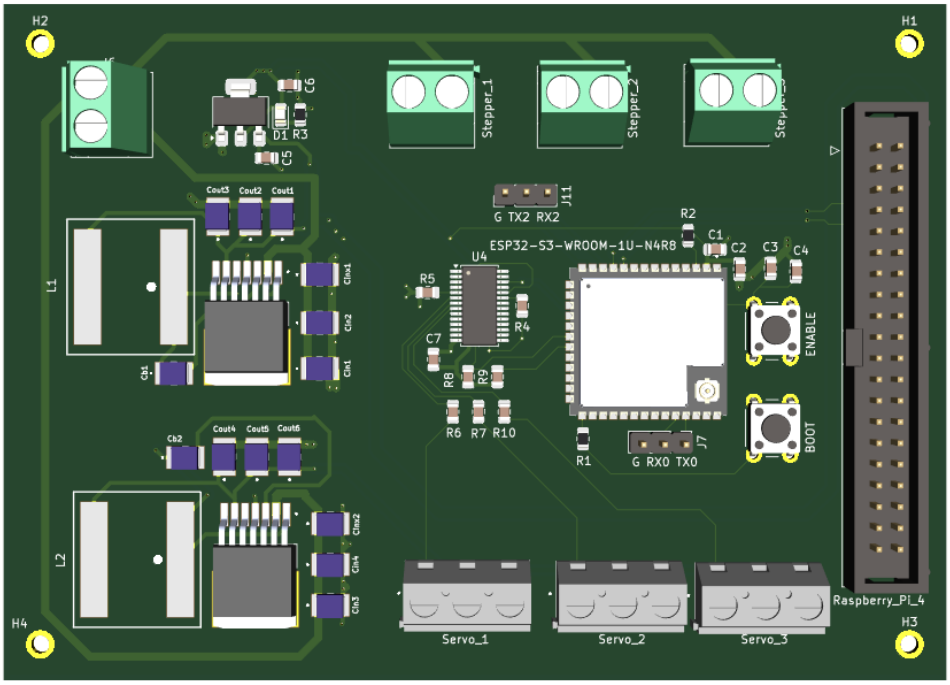


Figure 44: Main Board 3D (Top)

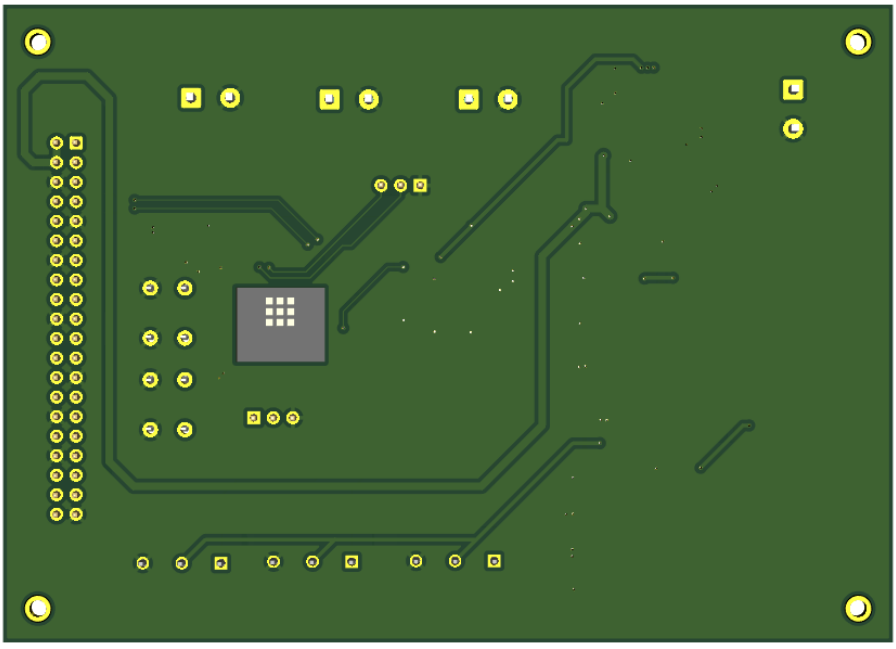


Figure 45: Main Board 3D (Bottom)

9.1.2 Spectrometer PCB

A second, smaller PCB is dedicated to driving the TCD1304 linear CCD sensor and is built around an STM32F401 microcontroller. This board is responsible for generating the precise timing signals required by the CCD, as well as amplifying and sampling its analog output. It includes a 4V LDO regulator to power the CCD, an amplifier circuit to boost the sensor's analog signal, and decoupling capacitors to reduce ripple and maintain signal integrity. Communication and power are provided through a USB connection to the Raspberry Pi, with the STM32 operating in USB CDC mode.

Keeping the spectrometer system on a separate PCB offers several advantages. It allows the CCD module to be detached and moved independently, which is helpful when aligning the sensor within the spectrometer subsystem. The microcontroller and sensor will be attached using header pins to easily replace faulty parts. Additionally, isolating the STM32 makes it easier to program and debug without affecting the main control board or motor systems. The Linear CCD PCB layout and 3D model are shown in the figures below.

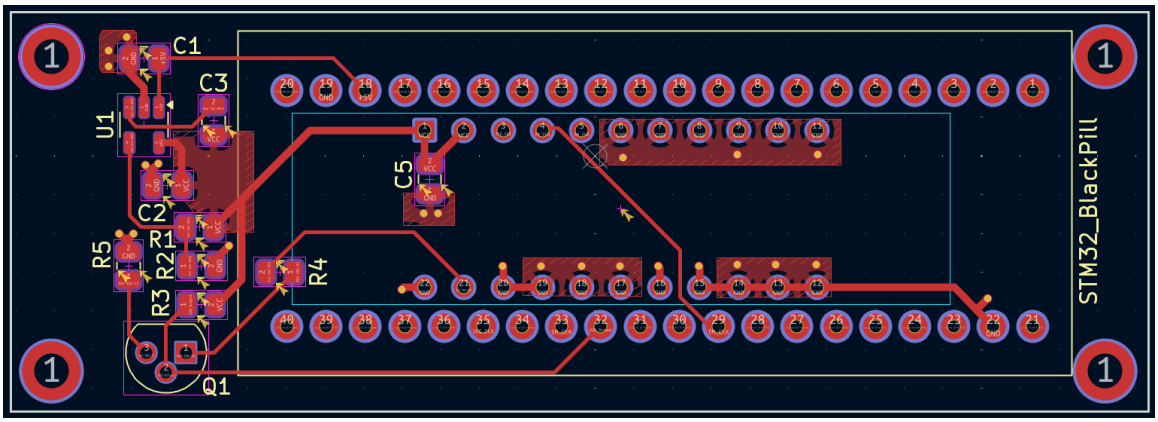


Figure 46: Linear CCD PCB Layout (Top)

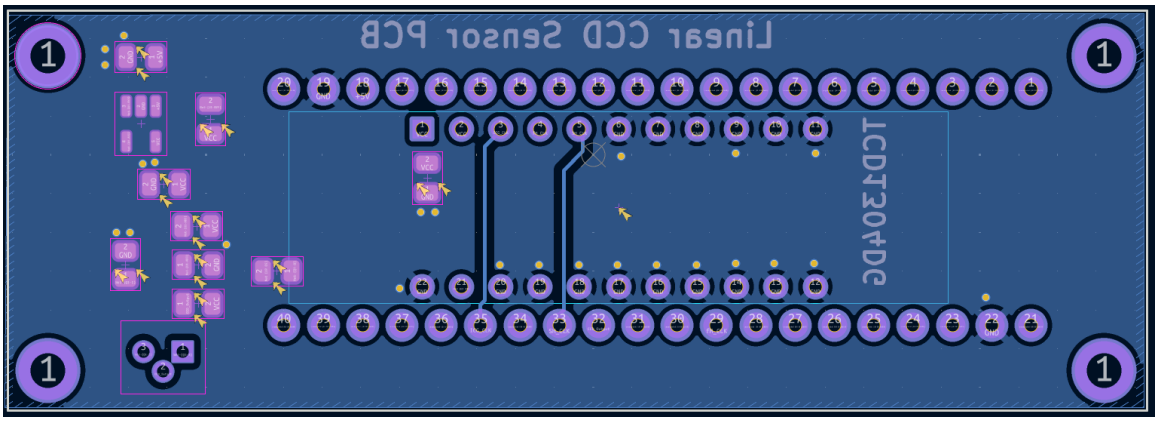


Figure 47: Linear CCD PCB Layout (Bottom)

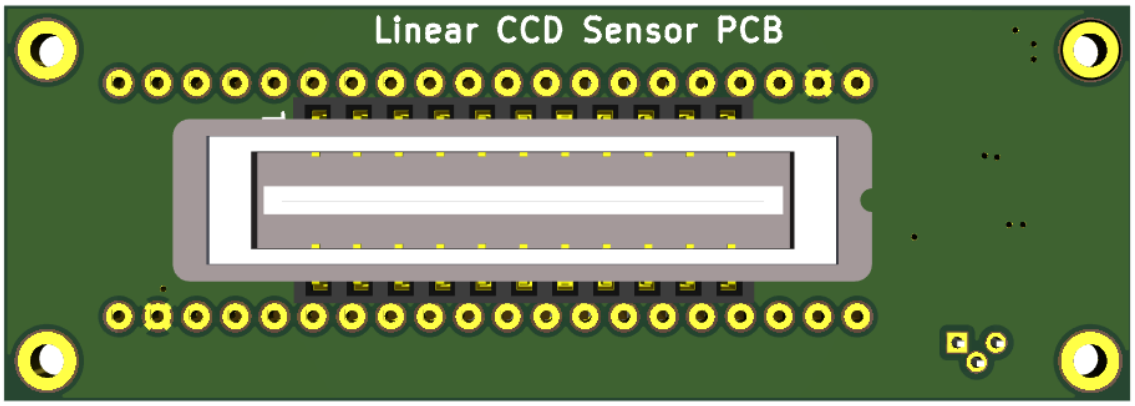


Figure 48: Linear CCD PCB 3D (Top)

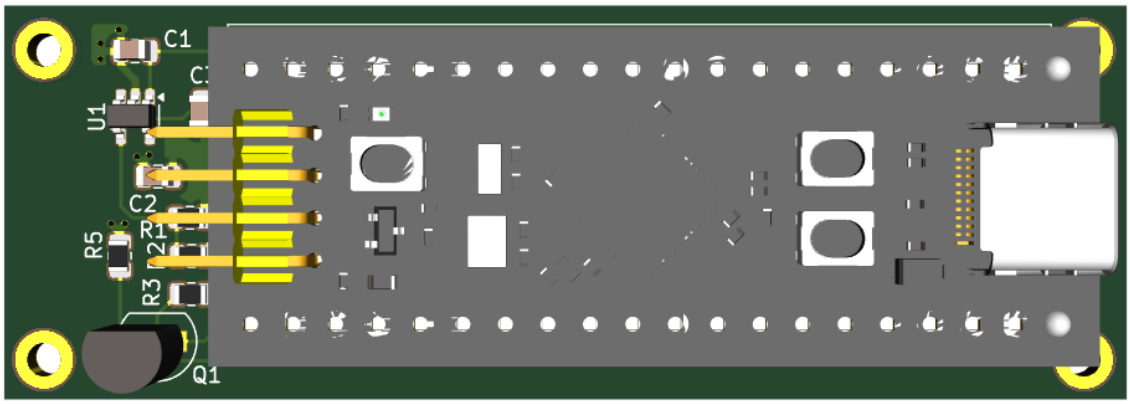


Figure 49: Linear CCD PCB 3D (Bottom)

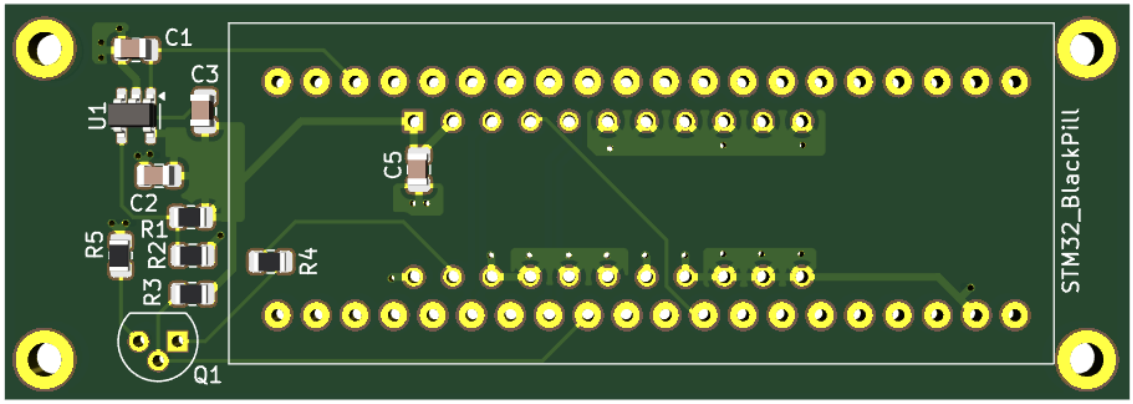


Figure 50: Linear CCD PCB 3D Without THT (Bottom)

9.2 Electronics Enclosure

The electronics enclosure holds all the major components of the T.R.A.S.H. control system in one compact and organized box. It was designed to keep wiring clean, protect the electronics, and make the system look neat.

The main PCB is mounted inside the box next to the **Raspberry Pi**. This layout keeps all connections short and easy to manage. Wires for motors, power, and sensors go through labeled terminals on the sides of the box as shown in Figure 51.

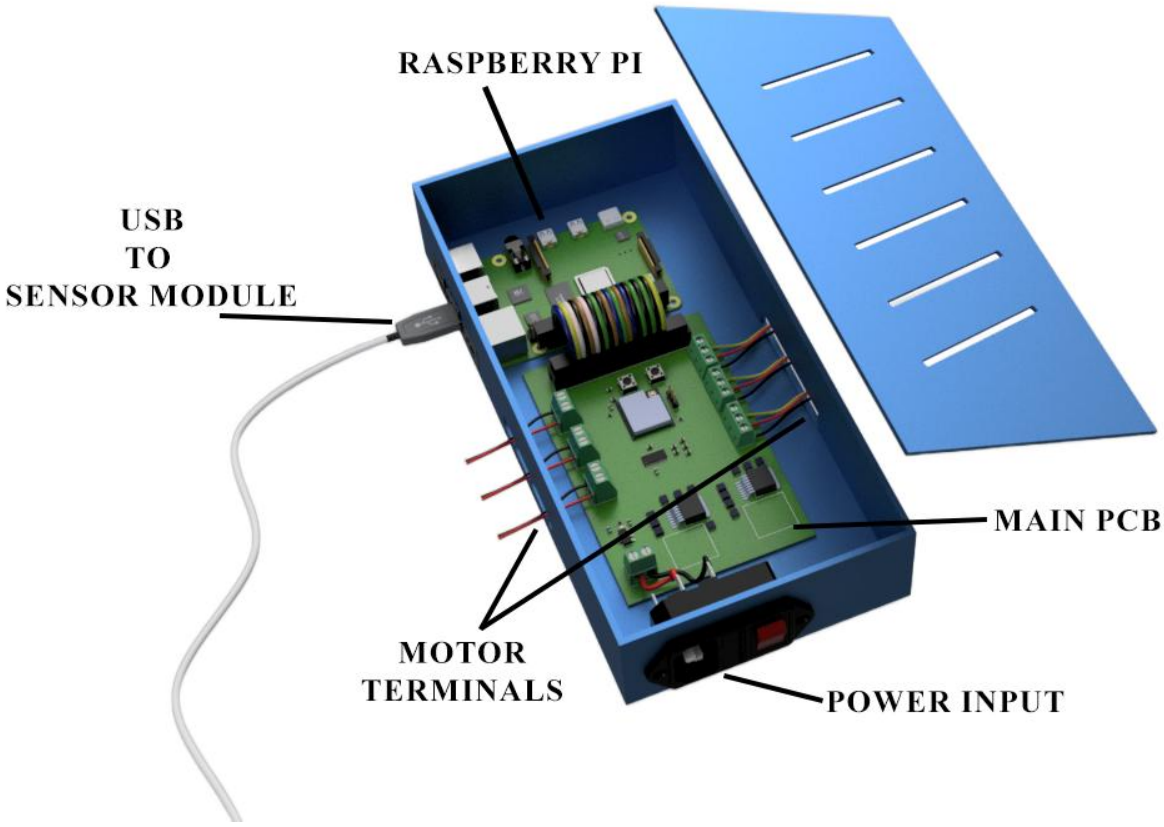


Figure 51: Controller Box and Enclosure Design

10.0 System Testing and Evaluation

This chapter covers the testing we've done so far on both the hardware and software components of the T.R.A.S.H. system. The goal was to make sure each part functions properly and does what we expect before full system integration. Testing early on helped us catch issues, confirm performance, and make any needed adjustments before moving forward.

We began by testing the hardware components separately—like the servo motors, stepper motors, CCD spectrometer, power supply, and camera. Each test included setup details, what we were looking for, and the results we observed. We also ran basic tests on the software to confirm that the core features were working.

After the individual tests, we started combining components to track integration progress and check how well different parts work together. At the end of this chapter, we include a short evaluation of the spectrometer's performance and a quick overview of what's planned for Senior Design II.

10.1 Hardware Testing

10.1.1 Servo Motor Testing

We began testing the servo motors individually by connecting them directly to the ESP32's PWM output pins. This bare setup allowed us to quickly verify basic functionality without using any external drivers. We ran simple code that swept the servos from 0° to 180° in steps, confirming that they moved smoothly and held their positions accurately. This helped us confirm the servos were working correctly before moving forward with the full setup.

After successful direct control testing, we switched to using the PCA9685 16-channel PWM driver, controlled by the ESP32 over I2C. This setup allowed us to manage multiple servos at once with more precise and stable PWM signals. We repeated the sweep and angle tests to verify correct operation with the driver and ensure the servos responded well to the new control signals.

In both setups, all servos moved smoothly without noticeable delay, jitter, or overheating. Power was initially supplied using a buck converter set to 5V for the servos, while the ESP32 was powered via USB. However, when we attempted to power both the servos and the ESP32 from the same buck converter, moving multiple servos at once caused the current draw to spike, which occasionally reset the microcontroller. To resolve this issue, we added a second buck converter dedicated to the servos in our PCB design to isolate the loads and prevent brownouts.

10.1.2 Stepper Motor Testing

The stepper motors in our system are controlled using the MKS Servo42C closed-loop stepper drivers, which communicate over UART using a custom command protocol. This protocol allows us to set the motor's direction, speed, and target angle.

For initial testing, we used the manufacturer's GUI to send commands to the stepper drivers via UART. This setup, powered by a 12V supply, allowed us to verify basic movement functionality and ensure that the motors responded correctly to position and speed commands, as shown in Figure 52.

Once basic operation was confirmed, we tested the built-in PID control by manually applying resistance to the motor during movement to see how well it held its position. The motors handled these stress tests without any overheating, jitter, or electrical issues.

After confirming stable performance, we moved on to integrating the stepper motors into our custom firmware. The ESP32 was programmed to send UART commands directly to the MKS drivers, allowing full control of speed, direction, and position through our own code. More details on this integration are covered in the software testing section.

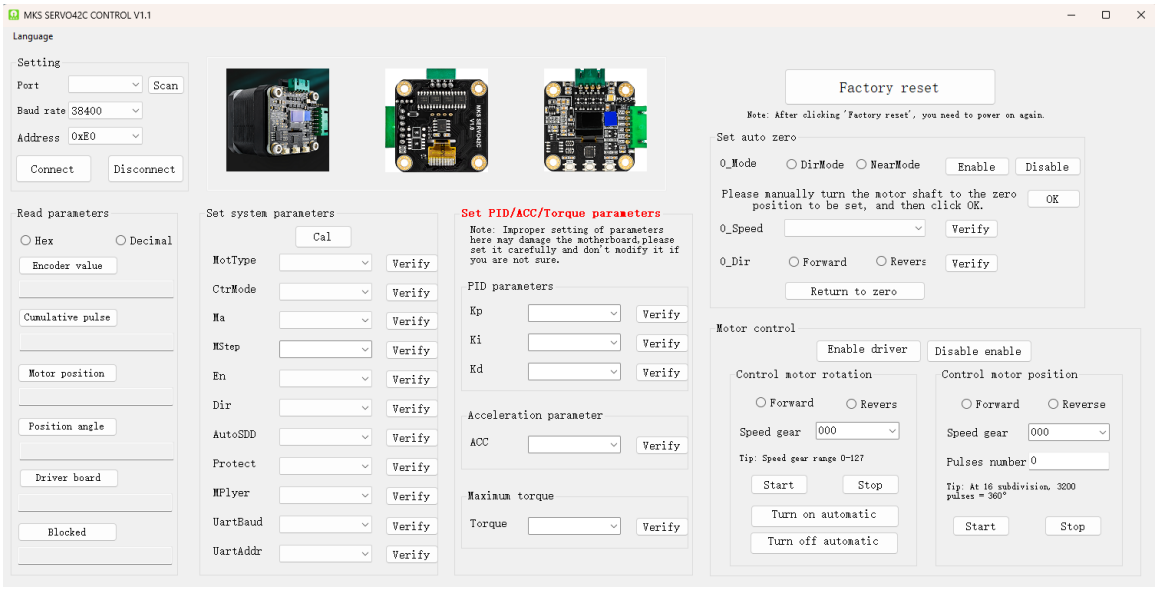


Figure 52: MKS Servo42C Control GUI

10.1.3 CCD Testing

The spectrometer subsystem uses an STM32F401 microcontroller to generate the required timing signals for the TCD1304 linear CCD sensor, including ICG, SH, and fM, all produced using hardware timers.

Initial hardware testing began by building the CCD driving and readout circuit on a breadboard as shown in Figure 53. A basic power supply was set up using an LM317 adjustable regulator, configured to output 4V for the CCD as recommended by manufacturer. The CCD's analog output was amplified using a 2N3906 PNP transistor in a simple common-emitter configuration to raise the signal level before feeding it into the STM32's ADC.

After verifying correct timing and signal quality on the oscilloscope as shown in Figure 54, the circuit was transferred to a perf board for easier integration and testing with the lens assembly. A USB OTG connection was used to stream ADC data to a PC, with the STM32 acting as a USB CDC (virtual COM port) device.

During testing, we observed that the CCD is highly sensitive to ambient light. Excessive light exposure caused visible leakage between adjacent pixels, which was confirmed on the oscilloscope. Because of this, integration time needs to be carefully controlled to avoid signal distortion and pixel bleeding, especially under strong lighting conditions.

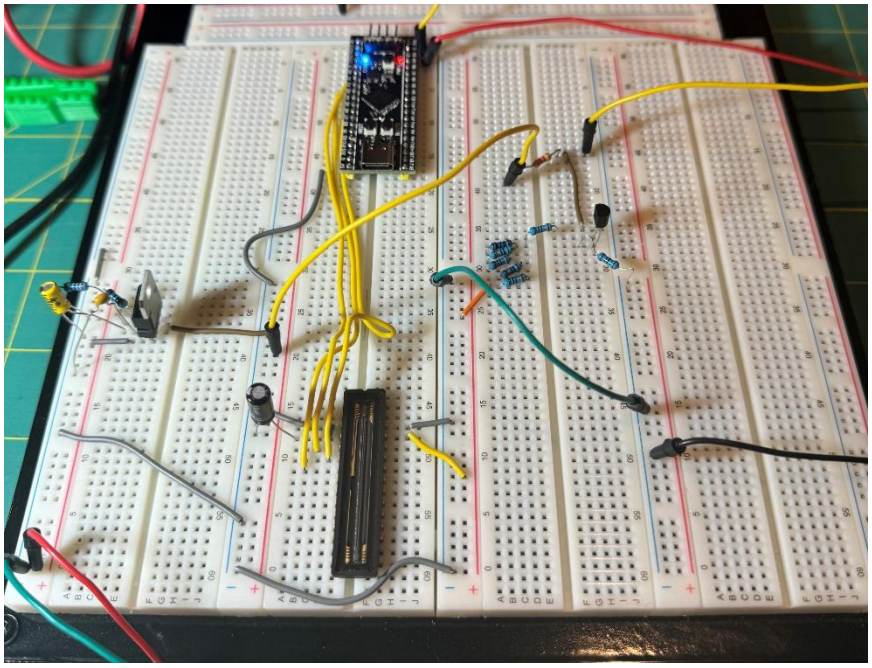


Figure 53: CCD Sensor Testing Circuit

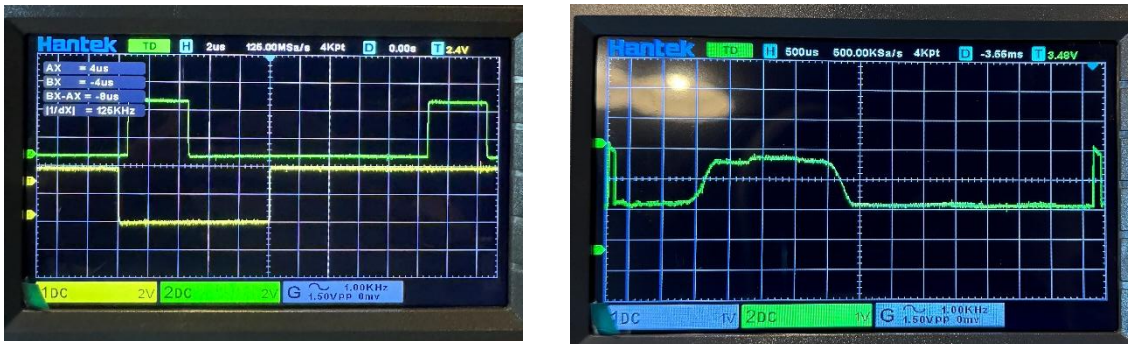


Figure 54: CCD Timing (left) and output (right) Signals.

10.1.4 Camera Testing

For the camera system, we originally planned to use the Raspberry Pi Camera Module V3, but due to stock issues, we replaced it with the Arducam Camera Module 3. This camera offers similar resolution and functionality, and it works well with the Raspberry Pi 5 using the built-in CSI port. The camera setup was simple, and it integrates cleanly with our computer vision software.

The camera is mounted directly above the sorting workspace to provide a full view of the items placed beneath it. During testing, the camera feed was clear, consistent, and provided enough detail for our detection software to recognize and track objects. This makes it possible for the robotic arm to know where each item is located.

The software we use for object detection includes tools like YOLOv8 and AWS Recognition, which help us identify items such as cans or bottles based on shape and appearance. These tools have been tested with our setup and perform as expected. The camera provides a clear and reliable image feed, which is essential for the software to work correctly.

Although we considered using additional lighting, we found that normal room lighting is enough for the camera to detect objects. Any extra lighting is reserved for the spectrometer system, which has its own lighting requirements during material scanning.

Overall, the camera and software work together as intended. The camera provides a live video feed, the software detects what's in the workspace, and the system quickly sends that data to the robotic arm. This makes the sorting process fast, accurate, and automated.

10.2 Software Testing

For software testing, we focused on getting each major part of the system up and running through custom code and then making sure everything could work together smoothly. Most of the work was done on the ESP32 for motor control and on the STM32 for the CCD sensor, with testing and debugging done using both serial commands and visual tools. Computer Vision was tested with a webcam connected to the PC using Python.

We started with servo motor control, writing a simple program that sent angle commands to the PCA9685 over I²C. Once we confirmed the servos were moving correctly, we moved on to the MKS Servo42C stepper drivers, which required custom UART commands based on the manufacturer's protocol. This included sending commands for angle, speed, direction, and reading angle feedback.

After testing both separately, we combined them into one program so the ESP32 could control both servos and steppers at the same time through serial commands. To make testing easier and more user-friendly, we also built a web-based GUI, shown in Figure 55, that lets us send motor commands through buttons and sliders instead of typing them in a terminal. This showed that the ESP32 is ready to take commands from the Raspberry Pi in Senior Design II.

On the CCD side, we created firmware for the STM32F401 that handles timing signal generation and reads the CCD output using the built-in ADC and DMA. At first, we just sent raw binary data to the PC, then wrote a simple Python program to plot the data. The initial results were glitchy because the ADC sampling wasn't synced properly with the CCD shutter, but we fixed this by changing the STM32 timer setup so the ADC would trigger at the right time.

Once we had clean plots, we added more features like adjustable integration time, signal averaging, and even gradient visualization as shown in Figure 56. These tools helped us debug and understand the CCD output better. In SD2, we'll convert this data into a format that machine learning can use to identify different materials.

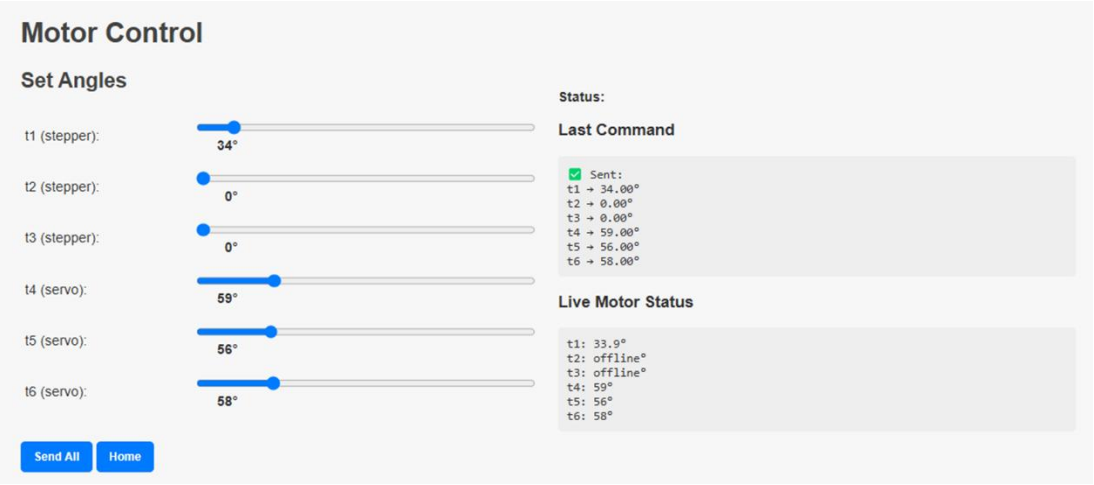


Figure 55: Motors Testing Interface

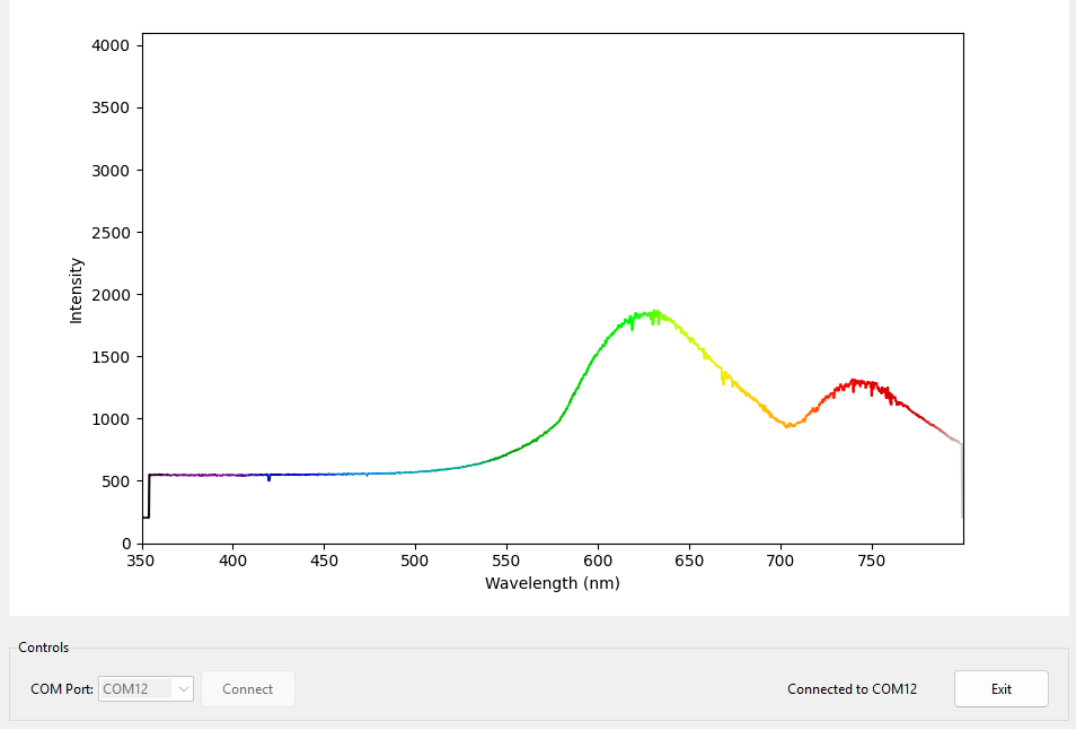


Figure 56: CCD Sensor Serial Plot

Computer Vision was tested using a standard webcam and later the Arducam Module 3, both connected to a Raspberry Pi and a development PC. We wrote Python scripts to capture live video and used both YOLOv8 and AWS Rekognition for object detection. YOLOv8 was run locally with a pretrained model, while AWS Rekognition was tested as a cloud-based service by uploading test images and analyzing the response data.

To supplement detection, we also used Canny edge detection to help identify the outlines of objects in the workspace. This allowed us to estimate the size of each object and determine if it was within reach of the robotic arm. Edge detection was especially helpful when YOLO detection alone was not sufficient for identifying the object's full dimensions. It also provided visual confirmation when the object was close enough to be picked up.

After verifying both YOLOv8 and AWS worked independently, we created a combined script to handle real-time detection with edge outlining. The results included bounding boxes and edge highlights, which were used to send object type, location, and size estimates to the robotic arm system. This showed that the computer vision system is functioning correctly and ready to support object sorting in Senior Design II.

10.3 Spectrometer Feasibility Test

The spectrometer system was successfully assembled, aligned, and tested, comprising a 10 µm slit, a 100 mm focal length collimating achromatic lens, a diffraction grating, an

80 mm focal length focusing achromat, and a linear CCD sensor. A set of calipers was used to measure and accurately position the spacing between each optical component. To verify collimation, I used a card to observe the beam diameter after the collimating lens; the diameter remained relatively constant, confirming that the light emerging from the lens was collimated. Given that the collimating lens has a 1-inch diameter and the grating is a ½-inch square, the grating was fully illuminated.

The diffraction grating was angled to prevent vignetting and ensure that no diffracted light was blocked by other optical elements. The 80 mm focusing lens, with a 2-inch diameter, collected all the diffracted light. Since the incoming beam was approximately collimated, the image plane was positioned 80 mm from the focusing lens, corresponding to its focal length.

For the test, an incandescent light source was directed at a Coke can to provide broadband reflective illumination. The resulting spectrum was successfully captured by the sensor and recorded as part of a live demonstration.

This test represented a critical milestone in Senior Design I, confirming that the optical system could deliver sufficient irradiance to the sensor to generate a usable signal for the electrical subsystem. Based on preliminary calculations, the required optical power on a single pixel at 633 nm was estimated to be 2.49 fW. The successful capture of spectral data validated that this sensitivity level was achievable with the current optical configuration.

11.0 Administrative Content

11.1 Budget and Financing

Sub-System	Description	Budget
Development	Dev board, jumper wires, and breadboards	$50
PCB and Electricals	PCBs, ICs, resistors, switches, etc.	$120
Motor System	Motors, drivers, and encoders	$220
Frame/Robotic Arm	Construction/assembly components, PLA filaments, mounting hardware	$180
Optics and Sensors	Sensors, lenses, light sources,	$700
Computer Vision System	Raspberry Pi, Camera, etc.	$140
Total		$1410

Table 37: Budget Estimates

11.2 Bill of Materials

Item	Quantity	Price	Total Cost
STM32F401 MCU	1	$6.69	$6.69
LT1761-BYP LDO	1	$5.07	$5.07
LT1761-3.3 LDO	1	$4.71	$4.71
LM2678SX-5.0/NOPB Buck Switching Regulator	2	$5.67	$11.34
STM32F401 Dev Kit	1	$14.99	$14.99
TCD1304DG	1	$6.95	$6.95
0.59NM NEMA 17 Stepper Motor	3	$22.00	$66.00
MKS_Servo42C Encoder/Driver	3	$11.00	$33.00
25KG Servo Motor	3	$13.99	$41.97
PCA9685 PWM Driver	1	$14.95	$14.95
Raspberry Pi LCD - 7" Touchscreen	1	$60.00	$60.00
12V 30A AC/DC Power Supply	1	$18.99	$18.99
Main PCB	1	$40.00	$40.00
CCD Sensor PCB	1	$20.00	$20.00
Electronic Components	NA	$30.00	$30.00
Raspberry Pi 5	1	$135.99	$135.99
Camera	2	$35.00	$70.00
Carbon Fiber Rods	3	$21.29	$63.87
PLA Filament	2	$19.99	$39.98

Bearings Pack	1	$17.59	$17.59
Assorted Hardware	NA	$60.00	$60.00
Ø1/2"N-BK7 Plano-Convex Lenses f = 50mm	1	$21.98	$21.98
Ø2"N-BK7 Plano-Convex Lenses f = 50mm	1	$34.75	$34.75
LEDW25E	5	$1.81	$9.09
IRLED (LED930L)	2	$8.28	$16.56
IRLED 2 (LED1050E)	2	$21.14	$42.28

Table 38: Bill of Materials

11.3 Work Distribution

Component	Primary Member	Secondary Member
Robot Arm	Tristan	Jayden
Frame		
Inverse Kinematics		Jayden
Motors/Drivers	Sayed	Tristan
Power Source		
Hardware and PCB		
Software Integration		Dylan
Camera	Jayden	
Raspberry pi		Sayed
Software Integration		Sayed
Computer Vision		
Lenses	Dylan	

CCD Sensor		Sayed
Spectrometer Design		

Table 39: Work Distribution

The table above outlines the primary and secondary members responsible for each major project component. Below, we provide a more detailed description of each team member's individual contributions to the project throughout the semester.

Sayed Almousawi

Sayed served as the primary hardware and electronics engineer on the team. His main responsibilities included researching and acquiring motors, drawing all schematic diagrams, designing the custom PCB that housed the ESP32 motor controller, power regulation circuitry, and terminal connections for external peripherals. He led the development of the embedded systems code for motor control, including UART communication with the MKS Servo42C stepper drivers and I2C communication with the PCA9685 servo controller. In addition, Sayed TCD1304 linear CCD circuitry, tested it, and developed the STM32 firmware for the TCD1304 spectrometer, which involved generating timing signals with multiple timers and acquiring analog data using DMA. He also created a Python interface for visualizing CCD output and debugging integration timing. His contributions were crucial in ensuring the electrical system was functional.

In terms of report writing, Sayed contributed to all hardware-related sections, including MCU and motor selection, and spectrometer implementation. He also wrote the full sections for Ch6: Hardware Design, Ch9: PCB Layout, and Ch10: Hardware Testing. Additionally, Sayed contributed to Ch5 by providing the AI examples, to Ch4 standards and constraints, and to Ch2's existing technology, and 3d illustration.

Dylan Bowers

Dylan was primarily responsible for the optical design and research behind the spectrometer system. He began by researching which spectroscopy methods would be feasible and accurate enough to differentiate recyclables from non-recyclables. During the optical design process, Dylan addressed practical challenges, adjusting the system to accommodate components that did not match the original design calculations. He also conducted research to select the most suitable linear CCD for the spectrometer. As an additional challenge, Dylan opted to implement an LED-based illumination source, requiring careful consideration of spectral output and power. Expanded research into depth scanning technologies to help the CE solve depth sensing issues if they arise. Created Zemax designs over 15 iteration most of which not in the paper due it their poor performance. Although each one offer an insight on how to better use Zemax and getting closer to a viable solation that was tested at the end of the semester with positive results.

In terms of report writing, all aspects of chapter 3 have to do with optics 3.14 to 3.19, chapter 5 sections introduction, pros, and cons of AI. Chapter 4 Optical safety constrains. Chapter 10 section 3 and contributed to some but not all Chapter 2 Existing products.

Tristan Koopman

Tristan took the lead on the mechanical design and structural planning of the system. This included leading the conversation that defined the project scope, goals, and system requirements. Tristan also took the lead in designing the full mechanical architecture of the robotic arm. He was responsible for the structural layout of the linkages, joints, and wrist, including torque analysis and internal load calculations for each major joint. Tristan calculated all required motor torques under worst-case loading scenarios and evaluated multiple speed reducer options to select the most suitable drivetrain design. He also analyzed mechanical constraints such as manufacturability, material limits, and time-based design tradeoffs. In addition to the mechanical system, he coordinated early-stage system planning with Sayed for motor selection. Tristan's work also covered the physical integration of components, ensuring the mechanical design was compatible with electrical and optical subsystems.

In terms of report writing, Tristan authored Sections 2.0 through 2.5 and created the House of Quality. He also wrote Section 3.1 (torque estimations), Section 3.10 (speed reducers), Section 4.2 (design constraints), Section 8.0 (structure design), and Section 11.4 (milestones).

Jayden Lumas

Jayden was responsible for leading the development of the computer vision pipeline and simulation environment using ROS2. He implemented real-time object detection using YOLOv8 and Python, integrating Canny edge detection to determine object proximity and dimensions for accurate robotic pickup. Jayden tested and compared both AWS Rekognition and YOLOv8 for object detection performance. Based on results, YOLOv8 was selected as the system's primary object detection model due to its local execution speed and higher accuracy, while AWS will be used as a backup cloud-based classifier for redundancy and remote support. Jayden also designed and tested the robotic end effector in Fusion 360, optimizing the geometry for workspace constraints and cable routing. He built the ROS2 simulation environment using Gazebo and MoveIt2 to test motion planning, joint control, and camera-based targeting. While the core inverse kinematics algorithms were handled by Tristan, Jayden assisted with integration and testing in ROS2 to ensure system functionality. His work formed the foundation for intelligent detection, decision-making, and robotic control.

In terms of documentation, Jayden authored Chapter 7: Software Design and contributed to Chapter 3.3: Computer Vision, Chapter 3.12: Camera, and Chapter 3.13: ROS & ROS2. His writing covered vision pipeline architecture, object detection model selection, camera integration, and ROS2 node structure. He also supported documentation anywhere involving software and computer vision.

11.4 Milestones

Task Description	Start Date	End Data
D&C	1/7/25	1/24/25
Revised D&C	1/24/25	2/7/25
Midterm Report	1/24/25	3/24/25
Revised Report	3/24/25	4/4/25
Final Report Draft / Demo Video	4/4/25	4/14/25
Final Report	4/14/25	4/22/25

Table 40. SD1 Milestones

Task Description	Start Date	End Data
Subsystems integration	TBD	TBD
PCB Assembly	TBD	TBD
Testing and Adjustments	TBD	TBD
Final Testing	TBD	TBD
Final Presentation/Demo	TBD	TBD
Final Paper	TBD	TBD

Table 41. SD2 Milestones

12.0 Conclusion

Senior Design I gave us the chance to take our idea for an automated sorting robot and begin turning it into a real, working system. The goal of our project, T.R.A.S.H., is to build a robotic system that can detect and separate recyclable materials from mixed waste using sensors, motors, and embedded control. While the full system hasn't been built yet, this semester was focused on designing each part, choosing the right components, testing

critical subsystems, and making sure everything is ready for full implementation in Senior Design II.

A major part of this phase was research and planning. We spent time comparing different microcontrollers and components before settling on the ones that made the most sense for our project. We chose the ESP32-S3 to handle all motor control because of its strong processing power and built-in Wi-Fi, and the STM32F401 for driving the spectrometer because of its timer and ADC capabilities. We also selected MKS Servo42C stepper drivers with UART control and encoder feedback, along with a PCA9685 board to control our servo motors. All of these choices were based on testing, technical features, and how they fit into our system.

One of the most important and challenging subsystems we worked on was the spectrometer. We used the TCD1304 linear CCD sensor to detect material types based on how they reflect light. This required generating precise timing signals and capturing the output with DMA on the STM32. After several iterations, we got consistent and stable data from the sensor, which we then visualized using a Python program we wrote. While we haven't implemented the material classification system yet, the data is clean enough that we'll be able to move forward with that next semester.

Besides electronics, we worked on planning the structure of the system. We designed the controller box layout, started mechanical designs for the arm and sorting setup, and made sure that all the parts talk to each other. We mapped out communication using UART, I2C, and USB, and tested basic messages between devices to confirm it's all feasible.

Even though we haven't built the full system yet, we made solid progress and cleared many of the difficult technical issues early on. We now have working prototypes of our most critical subsystems, and we've created a design we feel confident about. In Senior Design II, we'll move into the final build phase, connecting all subsystems, fabricating the mechanical parts, and developing the full control logic and material sorting process.

By the end of the next phase, our goal is to have a fully functional system that can detect materials, classify them, and sort them into the correct bins. Everything we've done this semester, from research, design, prototyping, and testing, has laid the groundwork for that.

Appendix A: References

1. Indicators, W.D., *World Births per Woman*. 2025, Data Commons: World Bank Group Data Catalog.
2. Evaluation, I.f.H.M.a., *Fertility forecasts and their implications for population growth*. 2024: healthdata.org.
3. Robotics, A. *AMP Cortex - Robotic Sorting System*. 2024; Available from: https://www.amprobotics.com/amp-cortex.
4. Ltd., Z. *Heavy Picker*. 2024; Available from: https://zenrobotics.com/heavy-picker.
5. Gordon, R. *Trash sorting robot learns by watching humans*. 2019; Available from: https://news.mit.edu/2019/trash-sorting-robot-learner-0816.
6. Systems, B.H. *Max-AI® – Intelligent Robotic Sorters*. 2024; Available from: https://www.max-ai.com.
7. IVV, F.I.f.P.E.a.P. *Hyperspectral Imaging for Material Detection in Sorting Systems*. 2024; Available from: https://www.ivv.fraunhofer.de/en/services/hyperspectral-imaging.html.
8. Felipe Mosquera, A.M., Kavinaash Jesurajah, Jackson Gilliland, Nolan McGinle, Amir Mohd, *Automated Parameter Adjusting Laser Engraving System*. 2024.
9. Callister Jr, W.D. and D.G. Rethwisch, *Materials science and engineering: an introduction*. 2020: John wiley & sons.
10. Overture, *Overture PETG Technical Data Sheet*. 2024: www.overture3d.com.
11. Singh, A., *Long-Term Strategies to Address Feedback Control Challenges in Robotics*. AZO Robotics, 2024.
12. LLC, H.D. *Harmonic Gear*. 2022.
13. LLC, H.D., *How it Works*. 2022, HarmonicDrive: https://www.harmonicdrive.net/technology.
14. Dejan *What is Cycloidal Driver? Designing, 3D Printing and Testing*.
15. Dejan, *What is Cycloidal Driver? Designing, 3D Printing and Testing*. How to Mechatronics: https://howtomechatronics.com/how-it-works/what-is-cycloidal-driver-designing-3d-printing-and-testing/.
16. Ridgway, J., *Towards the Design of Small, Inexpensive, Cable-Driven Dynamic Quadrupedal Robots*. 2023, University of Toronto (Canada).
17. Lickindorf, D., *2Dof leg assembled on the test stand*, in *Stanley - the capstan based quadruped kit*. 2021: hackday.io.
18. Samek, O., et al., *Application of laser-induced breakdown spectroscopy to in situ analysis of liquid samples.* Optical Engineering, 2000. **39**(8).
19. Dan Maclsaac, G.K., and Graydon Anderson *Basic Physics of the Incandescent Lamp (Lightbulb)*. 1999, Physics Teacher. p. 520-523.

20. Kasap, S.O., *Optoelectronics and Photonics: Principles and Pratices*. Person International Edition ed. S. Edition. 2013, Harlow, Essex, England: Pearson.
21. Saleh, B., *Introduction To Subsurface Imaging*. 2011, United Kingdom, Cambridge: Cambridge University Press.
22. Eismann, M., *Dispersive Spectrometer Design and Analysis*. 2012. p. 313-362.
23. Moroni, M., et al., *PET and PVC separation with hyperspectral imagery*. Sensors (Basel), 2015. **15**(1): p. 2205-27.
24. Hamed Masoumi, S.M.S., and Zahra Khani, *Identification and classification of plastic resins using near infrared reflectance spectroscopy*. International Journal of Mechanical and Industrial Engineering 2012. **6**: p. 213-220.
25. *KeyStone Architecture Universal Asynchronous Receiver/Transmitter (UART) User Guide*. 2010, Texas Instruments: Dallas, TX
26. *UM10204: I2C-bus Specification and User Manual*. 2021, NXP Semiconductors.
27. *TCD1304DG: CCD Linear Image Sensor Datasheet*. Toshiba Corporation.
28. *Robotic Mechanisms – LINKAGES – Simple Planar Linkages 51009*. Learn Robotics, 2013.
29. Ze, G.K., et al., *Challenges associated with drilling of carbon fiber reinforced polymer (CFRP) composites-A review*. Composites Part C: Open Access, 2023. **11**: p. 100356.
30. *What is a robot wrist and why does it matter in automation?* 2025.
31. Robotics, S., *Differential robot wrist*. 2024, Source Robotics: Maker World.
32. Loon, B., *How an Innovative Wrist Design with Built-In Planetary Gearbox Enhances the Performatnce of 6DOF Robotic Arms*. 2023, The White Owls.

13.0 Appendix B: List of Tables

Table 1: Basic Goals 3
Table 2: Advanced Goals 4
Table 3: Stretch Goals 4
Table 4: Engineering Requirements 5
Table 5: Optical Specifications 5
Table 6 Robot Arm Weight Assumptions 16
Table 7: Prototype Arm Dimensions: Length of linkages 17
Table 8: Prototype Arm Dimensions: Connector 17
Table 9: Prototype Arm Dimensions: Weight of Motors 17
Table 10: Prototype Arm Dimensions: Rod 17
Table 11: Prototype Arm Dimensions: EE 17
Table 12: Prototype Arm Dimensions: Gear 17
Table 13: MCU Selection 30
Table 14: SBC Selection 35
Table 15: Switching Power Supply Comparison 35
Table 16. 3.3V Linear Voltage Regulator Comparison 36
Table 17. 5V Buck Converter Comparison 36
Table 18: Motor Technology Comparison 39
Table 19: Servo Motor Selection: End Effector 39
Table 20: Stepper Motor Selection: Main Joints 39
Table 21: Stepper Motor Encoder Selection 41
Table 22: Stepper Driver Comparison 42
Table 23: Concept Alternatives – Speed Reducer 43
Table 24: Camera Comparison 53
Table 25: Ros Comparison 56
Table 26: Spectroscopy Methods Comparison 57
Table 27: Specular Reflectance Variables 60
Table 28: Linear CCD Sensor Comparison 61
Table 29: Illumination Source Comparison 63
Table 30: Comparisons of Optical Positioning Systems 71
Table 31: Relevant terms for (3.17.1) and (3.17.2) from [22]. 76
Table 32: Materials and Their Characteristic Wavelengths 79
Table 33: Computer Vision Tools 111
Table 34: State Machine for Inverse Kinematics and Simulation Subsystem 113

Table 35: Concept Alternatives – Material Selection....120
Table 36: Concept Alternatives – Linkage Cross-Section....120
Table 37: Budget Estimates....140
Table 38: Bill of Materials....142
Table 39: Work Distribution....143
Table 40. SD1 Milestones....145
Table 41. SD2 Milestones....145

14.0 Appendix C: List of Figures

Figure 1: TRASH Concept Illustration 6
Figure 2: Spectrometer Diagram 7
Figure 3: Hardware Block Diagram 8
Figure 4. High-Level Software Block Diagram 9
Figure 5: House of Quality 10
Figure 6 Free Body Diagram 19
Figure 7: Strain Wave Gear Components [13] 43
Figure 8: Cycloidal Drive with main component labels [15] 44
Figure 9: Robot Quadruped Leg Using Capstan Drive [17] 45
Figure 10: Spectral Response of Silicon-Based Detectors 62
Figure 11: Black Body Radiation Comparison Theoretical 64
Figure 12: Black Body Radiation Comparison Actual [19] 65
Figure 13: Relative Intensity of Light as a Function of Photon Energy [20] 67
Figure 14: Emission Spectrum of WLED 68
Figure 15: Spectral Information of a Variety of LEDs from Thor Labs. 69
Figure 16: Range Tomography [21] 72
Figure 17: Deblurring Effect [21] 74
Figure 18: $\Delta\lambda min$ vs Dispersion Angle 77
Figure 19: Power System Architecture 93
Figure 20: 5V Buck Converter Schematic 94
Figure 21: 3.3V LDO Regulator Schematic 95
Figure 22: TCD1304DG Linear CCD Sensor Schematic 96
Figure 23: ESP32-S3-WROOM MCU Schematic 97
Figure 24: Stepper Motor Wiring Diagram 98
Figure 25: Adafruit PCA9685 PWM Driver Reference Schematic 99
Figure 26: PCA9685 PWM Driver Final Schematic 99
Figure 27: Raspberry Pi GPIO Connections Schematic 100
Figure 28: Hardware Design Summary Block Diagram 101
Figure 29: Main PCB Schematic 102
Figure 30: Lens Layout 104
Figure 31: Diffraction Angle 105
Figure 32: Spot Diagram 106
Figure 33: Software Design Logic Flowchart 108
Figure 34: Simple Motor Control Block Diagram 115

Figure 35: User Interface GUI Layout 116
Figure 36: ROS2 – Based System Communication 118
Figure 37: Drilling of CFRP composite [29] 123
Figure 38: Joint Assembly SolidWorks 124
Figure 39: Joint Assembly Tendon Path 125
Figure 40: Differential Wrist Design [31] 126
Figure 41:Bevel gear motion [32] 126
Figure 42: Main Board Layout (Top) 128
Figure 43: Main Board Layout (Bottom) 128
Figure 44: Main Board 3D (Top) 129
Figure 45: Main Board 3D (Bottom) 129
Figure 46: Linear CCD PCB Layout (Top) 130
Figure 47: Linear CCD PCB Layout (Bottom) 130
Figure 48: Linear CCD PCB 3D (Top) 131
Figure 49: Linear CCD PCB 3D (Bottom) 131
Figure 50: Linear CCD PCB 3D Without THT (Bottom) 131
Figure 51: Controller Box and Enclosure Design 132
Figure 52: MKS Servo42C Control GUI 134
Figure 53: CCD Sensor Testing Circuit 135
Figure 54: CCD Timing (left) and output (right) Signals 135
Figure 55: Motors Testing Interface 137
Figure 56: CCD Sensor Serial Plot 138

15.0 Appendix D: List of Equations

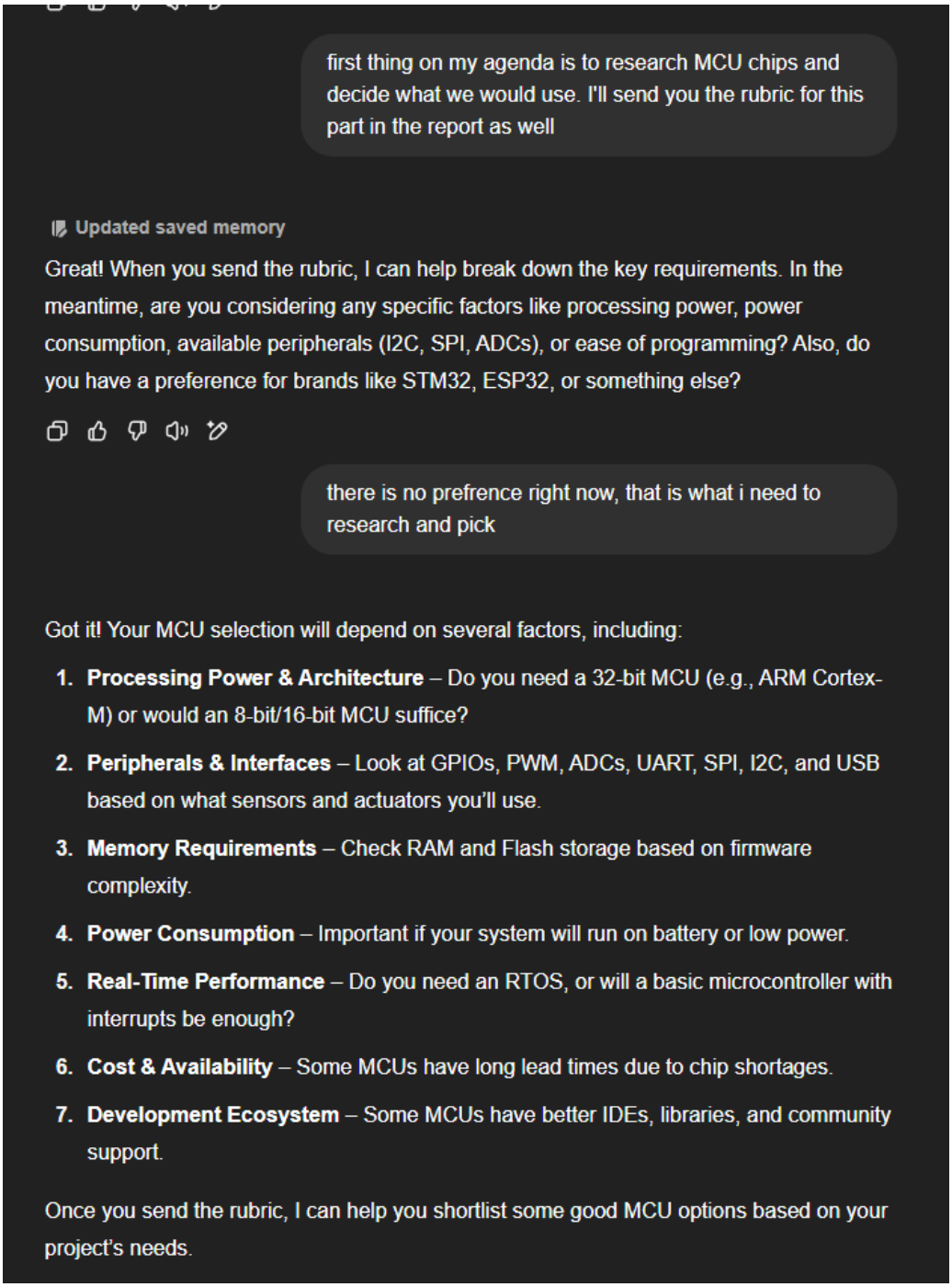
(3.1.1) **17**
(3.1.2) **17**
(3.1.3) **18**
(3.1.4) **19**
(3.1.5) **19**
(3.1.6) **19**
(3.1.7) **20**
(3.1.8) **20**
(3.13.1) **60**
(3.13.2) **60**
(3.13.3) **60**
(3.15.1) **64**
(3.15.2) **64**
(3.15.3) **66**
(3.15.4) **66**
(3.15.5) **67**
(3.15.6) **67**
(3.15.7) **67**
(3.16.1) **71**
(3.16.2) **72**
(3.16.3) **73**
(3.16.4) **74**
(3.16.5) **74**
(3.17.1) **76**
(3.17.2) **76**
(6.6.1) **103**
(6.6.2) **104**
(8.1.1) **121**
(8.1.2) **122**
(8.1.3) **122**

16.0 Appendix E: AI Prompts

E.1 - Microcontroller Selection Help

Platform: ChatGPT

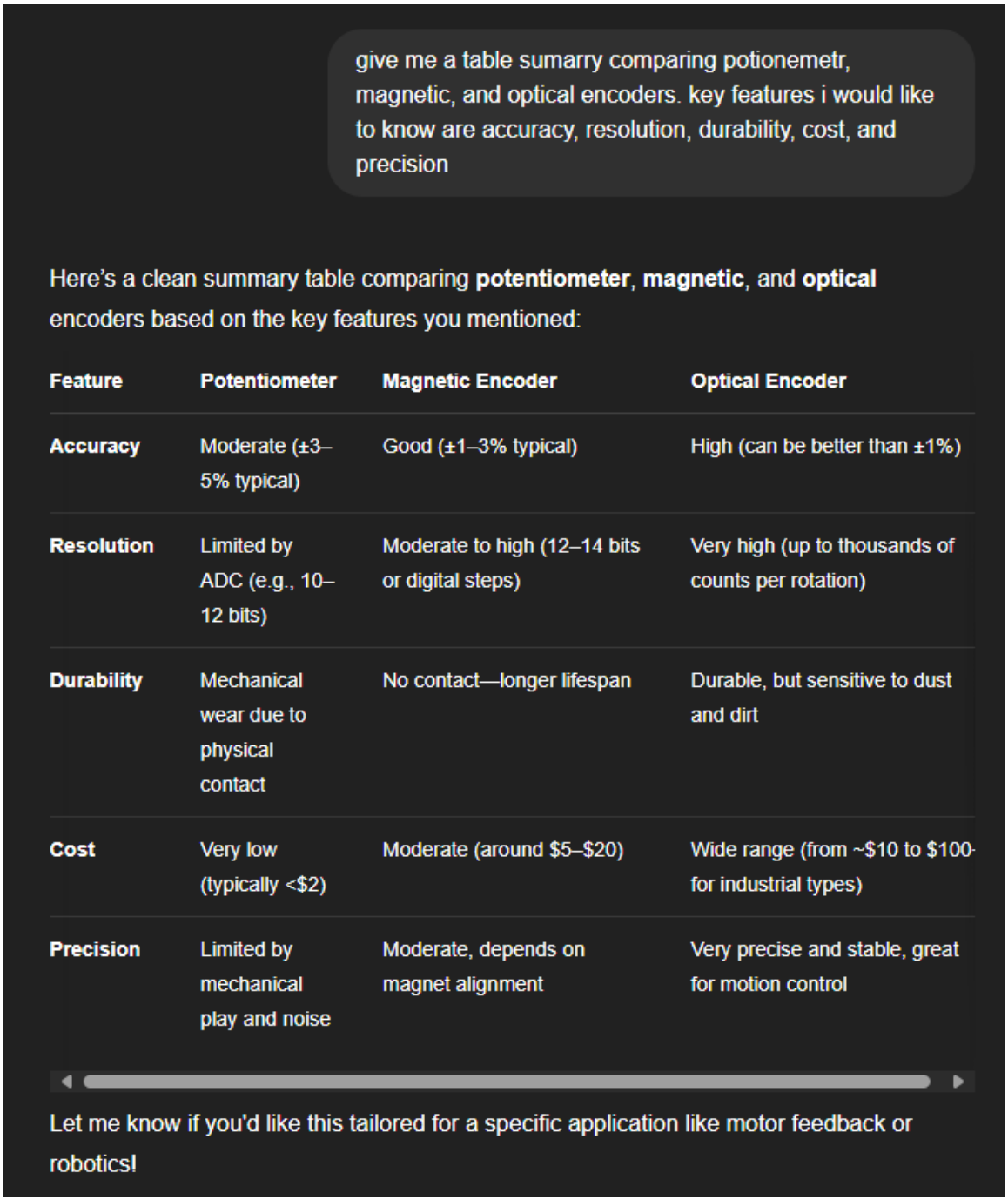
Screenshot:



E.2 - Encoder Comparison Table

Platform: ChatGPT

Screenshot:



give me a table sumarry comparing potionemetr, magnetic, and optical encoders. key features i would like to know are accuracy, resolution, durability, cost, and precision

Here's a clean summary table comparing **potentiometer**, **magnetic**, and **optical** encoders based on the key features you mentioned:

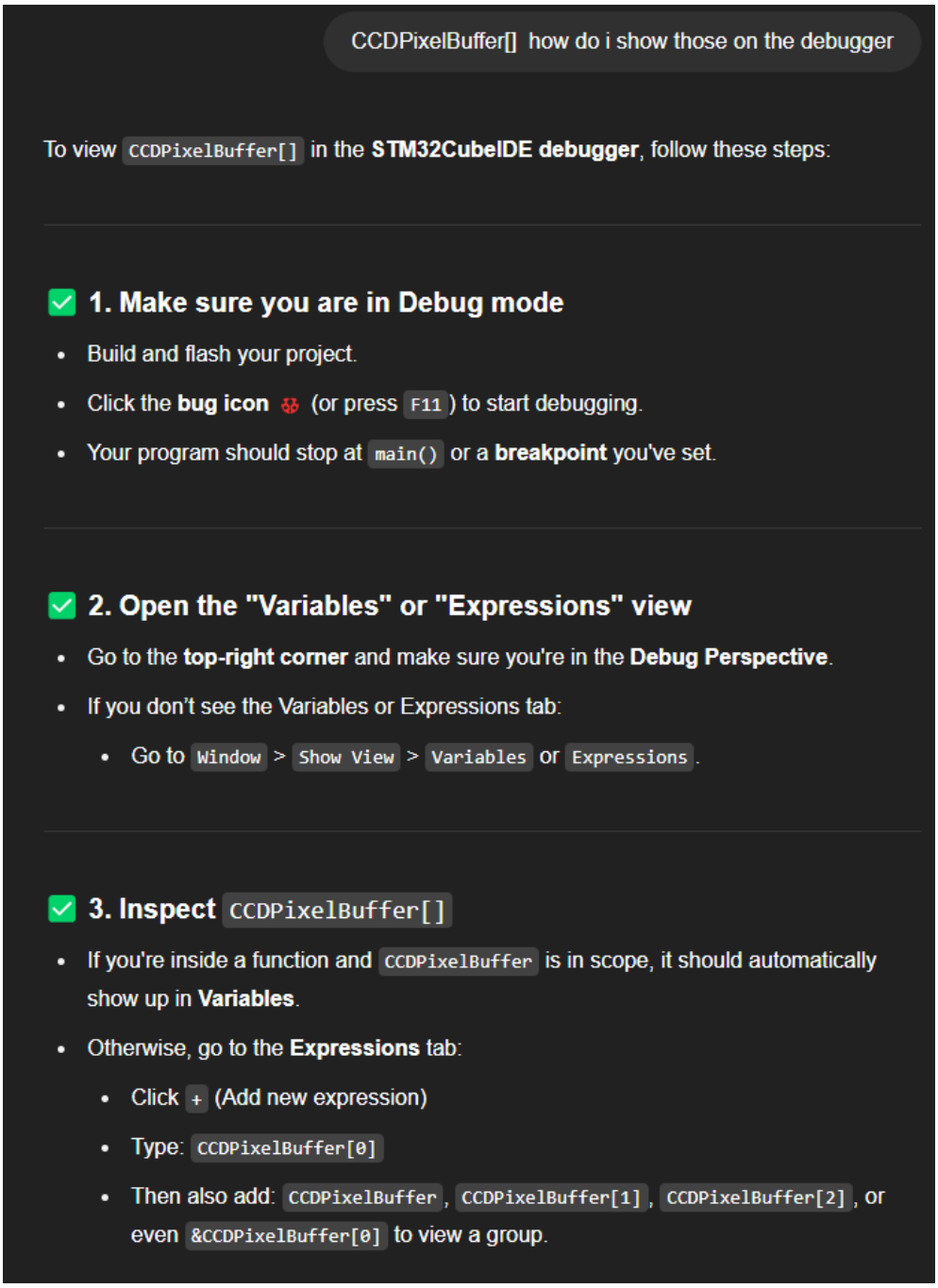
Feature	Potentiometer	Magnetic Encoder	Optical Encoder
Accuracy	Moderate (±3–5% typical)	Good (±1–3% typical)	High (can be better than ±1%)
Resolution	Limited by ADC (e.g., 10–12 bits)	Moderate to high (12–14 bits or digital steps)	Very high (up to thousands of counts per rotation)
Durability	Mechanical wear due to physical contact	No contact—longer lifespan	Durable, but sensitive to dust and dirt
Cost	Very low (typically <$2)	Moderate (around $5–$20)	Wide range (from ~$10 to $100+ for industrial types)
Precision	Limited by mechanical play and noise	Moderate, depends on magnet alignment	Very precise and stable, great for motion control

Let me know if you'd like this tailored for a specific application like motor feedback or robotics!

E.2 - STM32 Debugging Help

Platform: ChatGPT

Screenshot:



17.0 Appendix F: Design Competence Evaluation

ME Design Areas	Critical/Main contributor	Strong contributor	Necessary but not a primary contributor	Necessary but only a minor contributor	Only a passing reference	Not included in this Design Project
Thermal-Fluid Energy Systems						X
Machines & Mechanical Systems	X					
Controls & mechatronics	X					
Materials Selection			X			
Modeling & Measurement Systems				X		
Manufacturing				X		

Mechanical Topics Utilized in this Senior Design Project:

Topic	Criticality	Section and Page(s)	Comments
Machines & Mechanical Systems	Main	8.0	SD2
Controls & mechatronics	Main		SD2
Materials Selection	Necessary	3.2	SD1
Modeling & Measurement Systems	Minor	3.1.1, 3.1.2, 8.1.1	SD1&2
Manufacturing	Minor	8.1, 3.10, 8.2, 0, 8.3	SD1&2