Product Development File

Lab Section 002
Group 03

Robert Brunet
Kai Laugesen
Sheel Patel
Eric Wehlend

Specification and Planning Phase

Overview:
The goal of this project is to design and build an autonomous "scavenger" robot that can traverse an area up to 25 m², identify valuable objects based on their color, collect them, and return them to a base station—all within a 120-second time limit. The robot must operate fully autonomously using an ESP32-based control system and must be constructed using components from the MSE 2202 Lab Kit (along with additional approved parts).

Problem Statement:
Modern industrial and commercial environments sometimes require robotic systems to perform tasks in hazardous or hard-to-access areas. Our scavenger robot addresses this need by autonomously navigating an unmarked, obstacle-free area (except for perimeter boundaries), efficiently collecting objects (e.g., marbles or similar items) identified by their color reflectance/absorption properties, and depositing only the high-value items into a collection container.

(b) Product Design Requirements (understanding the problem)

Functional Requirements

- **Autonomous Navigation:**
  - Traverse a 25 m² area without external guides.
  - Use onboard sensors (IR, ultrasonic, encoders) to detect obstacles and navigate.
- **Object Detection & Collection:**
  - Identify valuable objects based on color.
  - Collect objects using a collection mechanism (e.g., rock picker, claw, or vacuum system).
- **Sorting & Drop-Off:**
  - Sort objects onboard using a color sensor–based system.
  - Deposit valuable items into a collection bin via a controlled dropper system.
- **Return-to-Home Function:**
  - Once the collection time nears the 120-second limit, the robot must autonomously return to base guided by a beacon system.

Performance Requirements

- **Task Completion Time:**
    - Complete the entire collection and drop-off within 120 seconds.
- **Precision:**
    - Accurate object collection and deposit with minimal error.
- **Reliability:**
    - No collisions with obstacles during autonomous operation.
- **Weight & Dimensions:**
    - Entire system must fit within the designated MSE locker dimensions.
- **Energy Efficiency:**
    - Operate within the energy budget provided by the lab kit's rechargeable battery pack.

Design Constraints

- **Budget:**
    - Additional components must not exceed a $50 budget.
- **Component Availability:**
    - Use primarily the MSE 2202 Lab Kit parts, supplemented with approved additional components (e.g., the CH-N20-3 encoder, etc.).
- **Autonomy & Safety:**
    - The robot must be self-powered and safe to operate autonomously without human intervention.

(c) Program Plan (planning and scheduling)

Key Due Dates

- **Concept Generation and Selection (Due: February 24, 2025):**
    - Brainstorm and develop multiple design concepts.
- **Detail Design Documentation (Due: March 7, 2025):**
    - Complete CAD models and the finalized bill of materials.
- **Drive System Implementation (Due: March 17, 2025):**
    - Functional demonstration of the robot's drive system with supporting code.
- **Pickup & Sorting System Implementation (Due: March 24, 2025):**
    - Demonstrate the complete pickup and sorting mechanisms.
- **Final Prototype & Demonstration (Due: March 30, 2025):**
    - Video demonstration of the fully operational system and submission of the prototype.
- **Product Development File (PDF) (Due: April 5th, 2025 before 5 PM)**
    - Collection of documents that cover the entire history of the design.
- **Product Design Report (Due: April 5th, 2025 before 5 PM)**
    - A technical report that clearly describes a single refined solution for the design problem.
- **Final Prototype (Due: April 5th, 2025 before 5 PM):**
    - Final physical and technical robot design.

Program Plan

We felt the best way to tackle this project is by building our schedule around the key due dates, setting weekly goals to ensure we stay on top of the assignment, and leaving room for delegating in case any of the group members get busy. The timeline reflects our task structure and the integration of necessary milestones to ensure that everything is progressing as planned.

February 17-22:

- **Define Goals**
  Kick off the project by setting clear, achievable objectives and the expected outcomes.

- **Identify Subsystems Required**
  Define all the subsystems that will be required for the robot, including the drive system, collection system, sorting mechanism, and sensor integration.

- **Generate Concepts for Each Subsystem**
  Over reading week, generate as many concepts for each subsystem as possible. Write a brief paragraph for each idea and create rough sketches to visualize the concepts.

February 23 - March 1:

- **Ideate Full Robot Design**
  Using the concepts generated earlier, ideate and refine the full design of the robot, combining the best features of each subsystem into a cohesive plan.

- **Format and Submit First Milestone Report**
  Finalize and format the design report, including the robot's overall structure, functionality, and key details. Submit the report as the first milestone.

- **Work on Product Development File**
  Begin populating the product development file with all relevant information, including the concepts, rough sketches, and design decisions made so far. Continue to update it throughout the term.

- **Bill of Materials (BoM) and CAD Model**
  Continue working on the bill of materials and the CAD model. Ensure that the CAD model aligns with the proposed design. Order the omni-directional mecanum wheels as part of this process.

March 2-8:

- **Finalize Report Format and Write-Up**
  Complete the formatting of the report and write a bill of materials, CAD model, and the functionality of the robot. This will be part of the second milestone.

- **Gather SolidWorks Files (Parts and Assemblies)**
  Gather all SolidWorks files (parts and assemblies) and zip them for submission.

- **Submit Second Milestone**
  Submit the second milestone, which includes the formatted report, the bill of materials, CAD files, and other relevant documents.

March 9-15:

- **3D Print Robot Chassis**
  Start the 3D printing process for the robot chassis. This will be the physical foundation for the assembly of the robot.

- **Receive Omni-Directional Mecanum Wheels**
  Expect the arrival of the omni-directional mecanum wheels and begin integrating them into the robot chassis.

March 16-22:

- **Wire DC Motor Encoders to ESP-32**
  Wire the DC motor encoders connected to each wheel to the ESP-32 microcontroller. This will allow for motor control and feedback for the robot's movement.

- **Develop and Upload Drive System Code**
  Develop and upload the initial code to the microcontroller during the lab session. Ensure that the basic drive functionality is working.

- **Record Drive System Demo Video**
  Record a demo video of the drive system in action, showing the functionality of the robot's wheels and movement.

- **Push Code to GitHub**
  Push the code to GitHub, ensuring that all changes and updates are properly version-controlled and shared with the team.

- **Submit Third Milestone**
  Submit the third milestone, including the demo video, code on GitHub, and any relevant progress updates.

- **3D Print Other Components (Collection Device and Sorter)**
  Continue 3D printing the components for the collection system and sorter. Attach them to

the chassis as they are completed.

- **Begin Product Design Report**
  As the full design takes shape, begin compiling the product design report to document the design process, challenges, and decisions.

March 23-29:

- **Implement Colour Sensor to Sorter**
  Integrate a color sensor into the sorting mechanism to detect the color of objects for sorting purposes.

- **Write Code for Colour Sensor**
  Develop the code to interface with the color sensor, and determine the threshold for detecting green objects.

- **Write Report on Threshold Determination**
  Document the threshold determination process, explaining how the color sensor distinguishes between green and non-green objects.

- **Record Collection and Sorting System Demo Video**
  Record a demo video showing the functionality of both the collection and spiral sorting systems.

- **Submit Fourth Milestone**
  Submit the fourth milestone, which will include the demo video, code for the color sensor, and the report on threshold determination.

- **Continue Working on Product Design Report and Product Development File**
  Continue refining the product design report and updating the product development file with additional information and milestones.

- **Finalize Wiring of Robot**
  Complete any remaining wiring for the robot's subsystems, ensuring that everything is fully connected and ready for testing.

- **Calibrate Robot's Drive and Path**
  Calibrate the robot's drive system and its path for use during the upcoming showcase. Test the robot on the surface where it will be demonstrated.

March 30 - April 4:

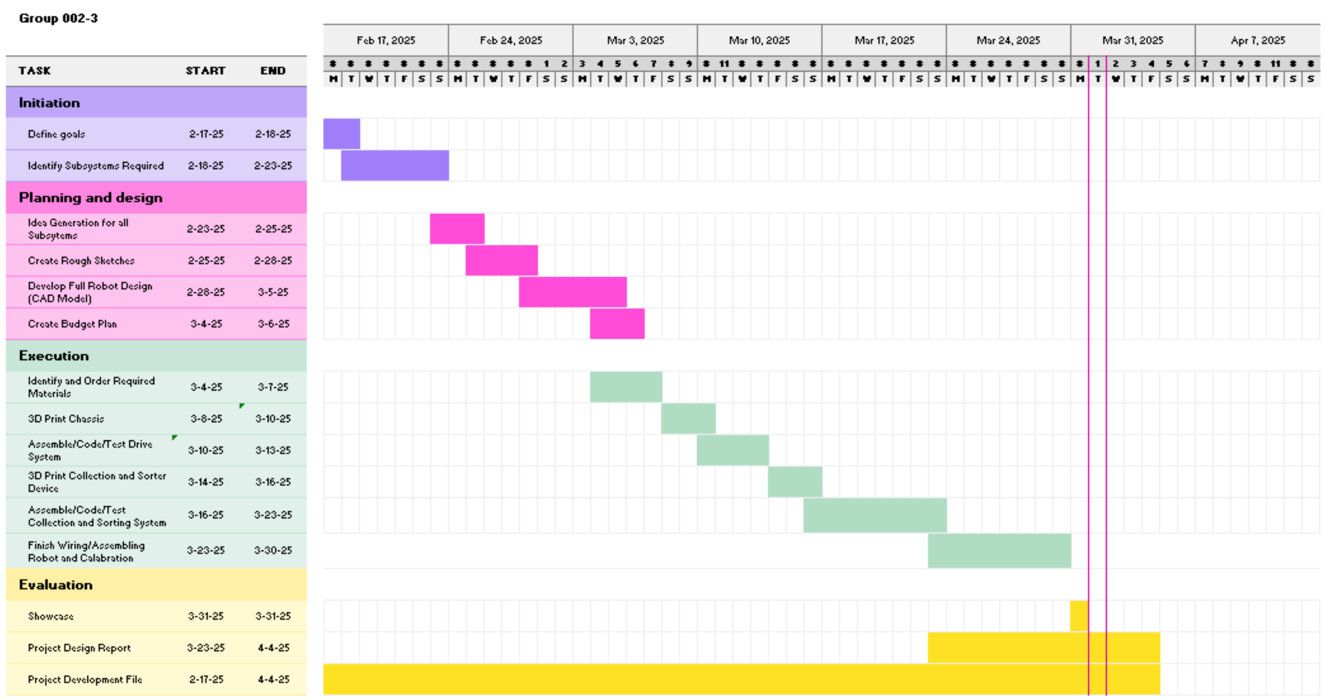
- **Record Fully Functional Robot Demo Video**
  Record a final demo video that showcases all subsystems working together, including

the drive system, collection system, sorting mechanism, and color sensor.

- **Finalize Product Development File and Product Design Report**
  Complete and finalize the product development file, ensuring it includes all documentation, designs, code, and progress reports throughout the project.

- **Put Final Prototype in Locker**
  Place the completed robot prototype in the locker, ensuring it is stored safely and ready for the showcase.

Gantt Chart Overview:



<u>Concept Generation and Selection</u>

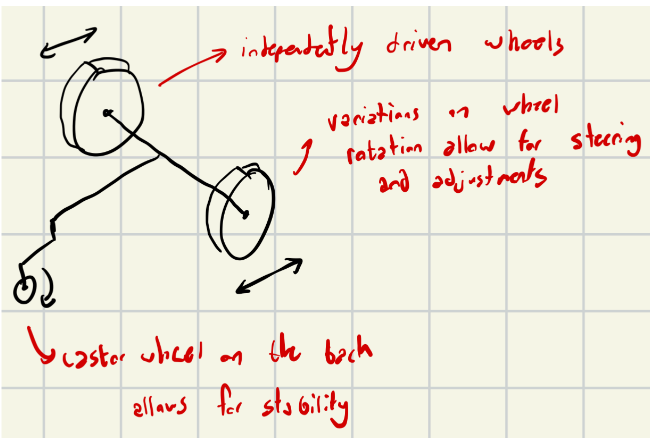
To begin this project, as a group we broke the design and development of this robot into its different subsystems, and generated concepts for them. Our robot will consists of 7 subsystems:

1. Mobility System
2. Collection System
3. Sorting System
4. Dropper System
5. Beacon System
6. Electrical System
7. Control and Navigation System

Mobility System

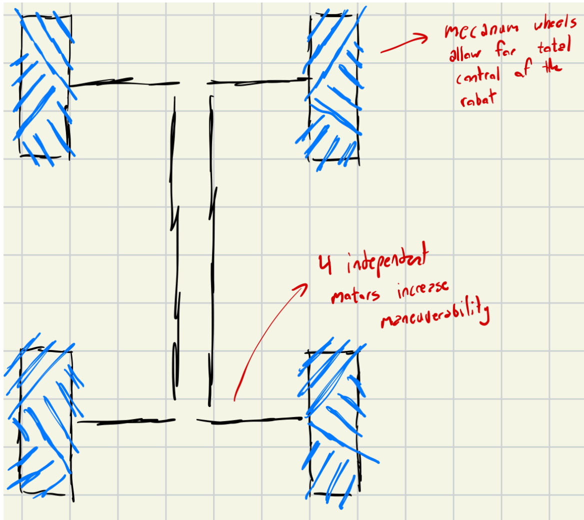
<u>Differential Drive</u>

A differential drive system is a practical and efficient choice for the scavenger robot, offering simple control and high maneuverability with two independently powered wheels and a free-spinning caster for stability. This design allows for zero-radius turns, making it ideal for maneuvering within the 25 m² competition area without the need for external guidance. Its low complexity and energy efficiency fit well within the project's constraints, helping to keep the robot lightweight and within power limits while ensuring smooth movement. However, wheel slip on low-friction surfaces and drifting due to motor speed imbalances could affect accuracy, making feedback mechanisms like encoders necessary for precise navigation. Additionally, if the robot carries too much weight, the caster could become unstable, increasing the risk of tipping. Careful design of the marble collection and sorting system will be important to maintain balance. Despite these challenges, a well-tuned differential drive system provides a reliable and efficient way for the robot to move autonomously, collect marbles, and return to base within the 120-second limit.



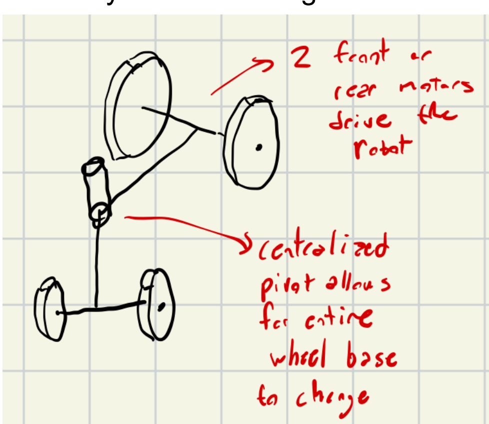
Omni Directional Wheels

To optimize the robot's movement and positioning parameters, we are looking into the application of omni-directional wheels, that is, a set of four mecanum wheels. The wheels allow for lateral movement in addition to forward and backward motion, which helps in navigating, collecting, and sorting the marbles without any external markers. This much maneuverability is useful to accurately position the robot around the marbles and to steer clear of other objects while performing the task. On the other hand, the mecanum wheels do add additional control complications because each wheel can be controlled independently, necessitating a more sophisticated motion control algorithm. Moreover, the project's budget of $50 over the supplied components also poses a challenge when considering the mecanum wheels because they are considerably more expensive. Regardless of this setback, improving efficiency in navigation and increasing the precision of tasks performed makes using these wheels viable, as the overall design of the robot is changed.



Articulated Steering

Another concept we've considered is a steering system with a centralized pivot mechanism. This design uses two powered wheels at either the front or rear of the robot, while the pivot allows the entire wheelbase to rotate relative to the chassis. By letting the robot's body articulate, this setup could offer better maneuverability than a standard differential drive system, making it easier to navigate tight spaces and execute smoother turns. One advantage of this system is that it could simplify steering control, as the robot would function more like a car, rather than relying on complex calculations for turning in place. However, designing a strong and stable pivot mechanism adds a layer of mechanical complexity, and the robot's stability during movement would need to be carefully addressed. With only 120 seconds to complete the scavenger task, this system's ability to enable fluid, continuous movement could help maximize efficiency when covering the 25 m^2 competition area to collect and sort marbles.



Wheel Shroud

A design idea for the scavenger robot is the use of a wheel shroud to cover the sides of the wheels when in motion, particularly in lateral movement. The primary role of a wheel shroud is to prevent things, such as marbles or garbage, from being trapped between the wheels, which can obstruct the movement of the robot or damage its components. This shielding cover could be especially beneficial in using omni-directional or mecanum wheels, as their side motion would draw gems into the wheel assembly from the side. A shroud may also be utilized to direct marbles into the collection system by preventing them from slipping between the wheels. However, installation of a wheel shroud would be difficult because it adds weight, potential encroachment upon wheel clearance space, and demands careful design so as not to restrict the motion of wheels or airflow. Due to potential for preventing obstruction and improving efficiency in collection, consideration of incorporating a wheel shroud is taken as long as the robot is not restricted by this addition nor by space or weight constraints.

Stepper Motor

In the process of choosing the right motors for the scavenger robot's drive system, one that comes to mind is the employment of stepper motors. Stepper motors travel in fixed steps, giving highly accurate positioning without the necessity for supplemental feedback systems. This position makes them ideal for applications where accurate and incremental motion is needed, such as aligning the robot in exactly the right position with respect to marbles when collecting them or positioning it close to the sorting bin. The precision of stepper motors would improve the robot's performance in covering the 25 m^2 area. However, a major disadvantage is the risk of stepping delays, which would impede smooth and rapid movement. This weakness could hurt the robot's overall speed and responsiveness, which is crucial given the 120-second time limit to complete the scavenger task. Therefore, the application of stepper motors is a compromise between precise positioning and the requirement for fast, smooth motion over the field.

Servo Motor

Servo motor is another motor option for the drive mechanism of the scavenger robot. Standard servo motors are designed to provide precise angular positioning, typically in a limited range of 180°, and are best suited for applications where controlled rotational motion is necessary. Even though there are continuous rotation servos, they are actually standardized servos that have been modified and lack control precision compared to other motors. The main advantage of servo motors is that they may be designed to position exactly and this would suit non-drive usage such as lift mechanisms or arms for gripping. However, due to their narrower range of operation and lesser accuracy in repeated turnings, they cannot best fit as primary drive motors in the said application. Because there is the need for smooth, continuous movement along the 25 m^2 area within the 120-second time limit, servo motors may not provide the even velocity and control needed for effective navigation and object pickup.

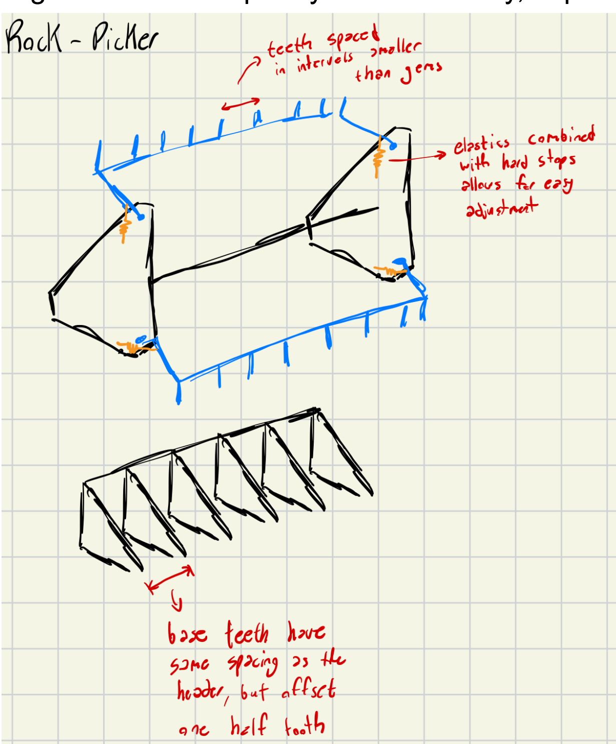
Encoded DC Motor

Another motor option that was taken into consideration for the scavenger robot's drive system is the use of DC motors with encoders. DC motors generate continuous rotation and thus are very appropriate for uses where constant motion is required, such as moving the robot across the 25 m^2 competition area. When paired with encoders, they provide convenient feedback for closed-loop speed and position control, allowing greater accuracy for navigation and object fetching. This setup enables the robot to monitor distance covered and maintain speeds steady, which is helpful in adhering to the 120-second time constraint. There are some drawbacks to this motor choice, however. The utilization of a closed-loop control system makes the programming more complex and must be precisely calibrated to receive exact readings from the encoders. DC motors also suffer from torque issues at low speeds, making low-speed maneuvers challenging. Mechanical wear and tear over a period of time and potential slippage between wheels and surface can also result in inaccuracies in distance travel in spite of encoder feedback. These have to be considered while deciding if DC motors with encoders are the most suitable for the robot drive system.

Collection System

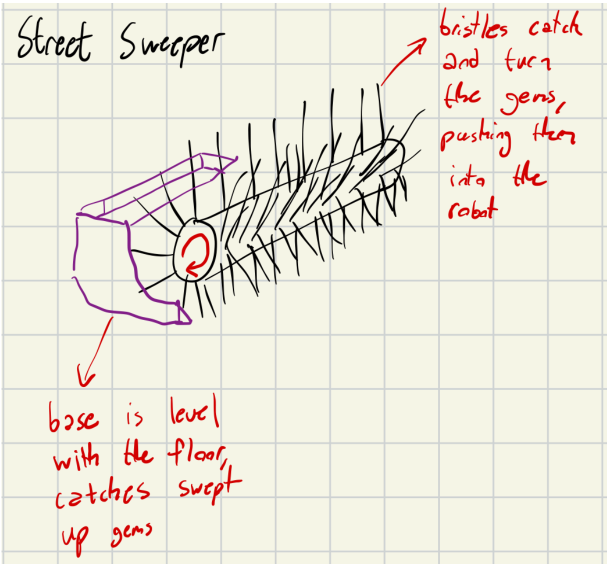
Rock Picker

A rock picker-style collection system offers an efficient solution for gathering marbles as it allows for bulk collection without requiring precise alignment. This mechanism, inspired by agricultural rock pickers, typically uses rotating tines or a conveyor system to lift marbles off the ground and deposit them into a collection bin. The ability to collect multiple marbles at once makes it well-suited for covering a 25 m² competition area within the 120-second time limit. Additionally, its automated and continuous operation can maximize efficiency, ensuring more valuable marbles are gathered within the time constraints. However, the design involves multiple moving parts, which increase power consumption and add weight to the robot, a critical factor given the requirement to fit within a locker and maintain a reasonable mass-to-score ratio. Careful optimization of materials, motor selection, and battery efficiency is necessary to ensure the system remains lightweight while maintaining high collection performance. Despite these challenges, a well-designed rock picker mechanism can significantly enhance the robot's ability to gather marbles quickly and effectively, improving overall competition performance.



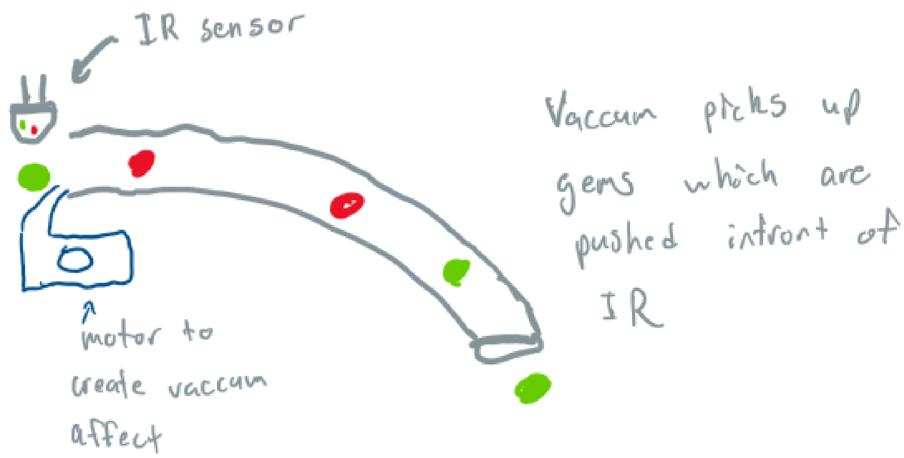
Street Sweeper

A street sweeper-style collection system utilizes a rotating brush mechanism to sweep marbles into a collection bin. Soft bristles or rubber tines help guide marbles without bouncing them away, allowing for the simultaneous collection of multiple marbles. This method is simple to operate and requires only one rotating brush, making it a straightforward approach to gathering marbles. However, the effectiveness of this system depends on the bristle stiffness and rotation speed, as marbles may be pushed away rather than collected if not properly configured. Additionally, insufficient bristle strength may limit the ability to direct marbles into a chute.



__Vacuum Sucker__

A vacuum pickup system uses suction to lift marbles from the ground and moves them in front of an IR sensor for colour detection. It provides a rapid and touchless method of marble pickup, reducing the risk of unwanted movement or bouncing. It can also be set up to selectively collect single marbles, which can simplify sorting. But the system's efficiency is based on suction strength and energy consumption, as low vacuum will fail to lift marbles, but too strong a vacuum will drain the battery very quickly. Moreover, clogging or non-uniform airflow will impact performance, and design must be well handled to ensure smooth operation throughout the collection process.

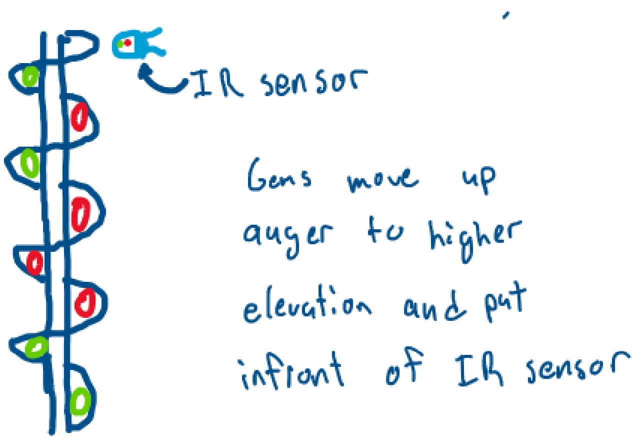


Claw
A claw-based collection system uses two moving jaws to securely grab marbles for transport, similar to the servo-controlled claw mechanism previously built. This method offers high precision, allowing the robot to selectively pick up specific marbles, potentially reducing the need for a separate sorting mechanism. However, this approach is more complex due to the multiple moving parts and servo motor control, requiring precise coordination for effective operation. Additionally, the slower pickup process, where only one or two marbles can be collected at a time, may limit efficiency compared to bulk collection methods.

Sorting System

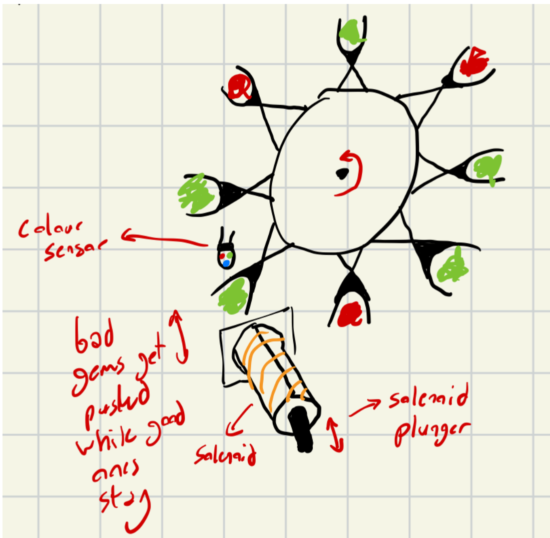
Auger
An auger sorting system uses a rotating screw mechanism to transport marbles through a tube, where they are scanned and sorted based on color before reaching the collection bin. As

marbles move through the auger, they pass an IR color sensor positioned along the tube, which detects their reflectance properties. Once a marble is identified, a microcontroller processes the sensor's data and determines its path. A servo-controlled gate located at a designated point along the auger changes position based on the detected color, redirecting marbles into different collection bins. This system ensures continuous and controlled sorting, allowing for precise classification without requiring additional post-sorting mechanisms. Pros include consistent movement, compact design, and reduced risk of misalignment, as marbles remain in the auger throughout sorting. However, accurate sensor placement and response timing are crucial to prevent misclassification, and servo actuation must be quick enough to adjust before the next marble arrives. Additionally, constant auger motion increases power consumption, requiring an efficient control system.



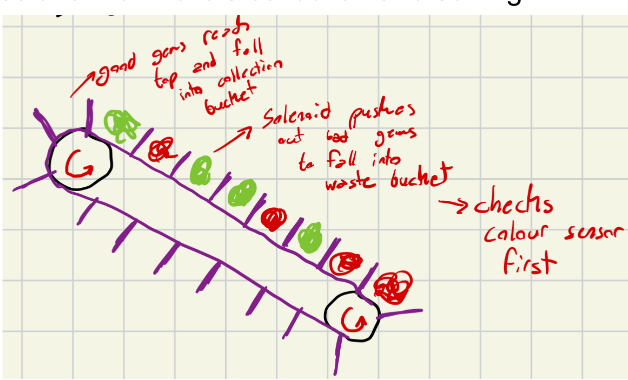
Spiral Sorter

Rotating wheel sortation applies the use of a color sensor and a solenoid plunger to sort marbles based on their classification. The sensor detects the moving marbles' color along the rotating wheel as the solenoid displaces unwanted marbles while allowing wanted ones to get into the collection area. It accommodates constant sorting without stoppage, thus appropriate for continuous high-speed usage. The benefits are rapid and mechanical sorting, minimal human interference, and ability to sort more marbles in succession. One drawback is careful timing in triggering the solenoid properly, with incorrect sorting from misalignment or mechanical lag. The presence of a turning device also makes the setup complex and may require special calibration for continuous use.



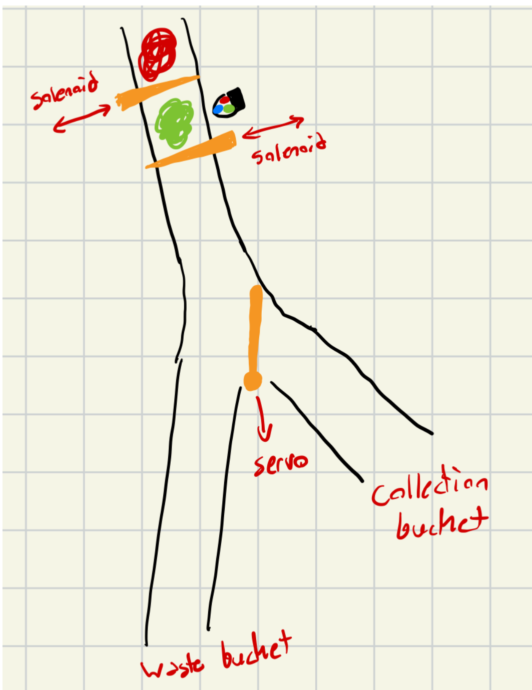
Conveyor Sorter

The conveyor sorter system uses a moving belt, a color sensor, and a solenoid to separate marbles by color. As the marbles move along the belt, the sensor identifies each marble's color. Marbles that match the desired criteria continue to the collection bucket, while the solenoid diverts the others into a waste bin. This system provides continuous, high-speed sorting, making it effective for processing multiple marbles within the 120-second time limit. Pros include automated operation, efficient handling of multiple marbles, and precise separation, reducing the need for additional sorting. However, alignment issues may arise if marbles are not properly spaced, and the solenoid timing must be precise to avoid incorrect sorting. Additionally, the mechanical complexity of the conveyor system increases power consumption and potential maintenance requirements. With proper calibration, this system can offer a fast and reliable solution for marble collection and sorting.



Zipper funnel

A zipper funnel sorting system provides an efficient and accurate method for classifying marbles based on color, ensuring that each marble is individually processed for precise sorting. It consists of three gates and two exit paths, where the first gate directs the marble into an IR sensor chamber to identify its color. Once scanned, the second gate opens, allowing the marble to proceed while a servo motor activates to guide it down the appropriate chute to its designated collection area. This structured approach minimizes sorting errors and improves organization by ensuring that only valuable marbles reach the final collection bin. The system's controlled and step-by-step process eliminates the need for post-collection filtering, making it a streamlined and effective solution. However, the use of multiple servos increases control complexity, requiring precise timing and synchronization to prevent errors. If the gates do not operate quickly enough, there is a risk of jamming or bottlenecks, which could slow the sorting process. Additionally, servo malfunctions or misalignments may impact accuracy, requiring calibration and optimized programming to maintain reliability. Despite these challenges, a well-designed zipper funnel system offers a highly accurate and efficient sorting method, making it a practical solution for the marble collection and classification requirements within the 120-second time limit.



Dropper System

Front Bucket

A front bucket system offers a simple and controlled method for depositing collected marbles at the drop-off location. Positioned at the front of the robot, it features a servo-controlled trap door that releases marbles when the robot reaches the beacon. The trap door hinges outward, allowing the marbles to fall into the container below. This system enables quick and efficient unloading, reducing the time needed for depositing marbles within the 120-second limit. Pros include low mechanical complexity, reliable operation, and fast marble release. However, precise alignment with the drop-off location is crucial, as misalignment could result in marbles missing the container. Additionally, the trap door must function smoothly to avoid partial dumping or jamming, and the bucket's capacity may limit collection efficiency. With proper calibration, this system provides a straightforward and effective unloading solution.

Rear Bucket

A rear bucket system functions similarly to the front bucket but is positioned at the back of the robot to help balance weight distribution. Instead of unloading directly upon reaching the beacon, the robot would first turn 180 degrees before activating the servo-controlled trap door, allowing marbles to fall into the collection container. This design helps prevent front-heavy imbalance, improving overall stability while maintaining a controlled and efficient unloading process. Pros include better weight distribution, reduced risk of tipping, and a reliable dumping mechanism. However, the extra turning maneuver adds time to the process, and alignment with the drop-off point must still be precise to ensure marbles land correctly. Additionally, trap door calibration remains important to prevent jamming or incomplete unloading. If weight balance becomes an issue, the rear bucket provides a practical alternative for the marble collection and deposit system.

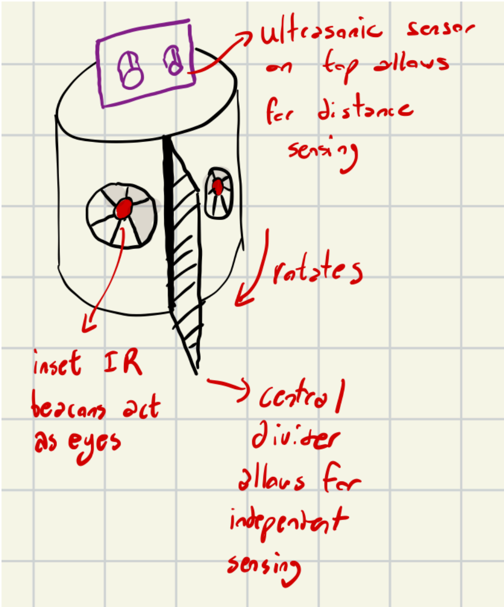
Side Door

A side door system provides an alternative method for unloading marbles by using a trap door mechanism on the side of the robot rather than the front or rear. When the robot reaches the beacon, a servo activates the side door, allowing marbles to exit the collection area and fall into the designated container. This design eliminates the need for additional turning maneuvers, making unloading faster and more direct. Pros include simplified navigation, reduced unloading time, and even weight distribution, as the robot does not need to shift its balance significantly. However, precise alignment remains crucial, as the side door must be positioned correctly relative to the drop-off point to prevent marbles from missing the container. Additionally, marbles may roll unpredictably when released from the side, requiring careful control over door opening speed and angle. With proper calibration, a side door system can provide a quick and efficient unloading solution while maintaining stability and simplicity in operation.

Beacon System - Locators

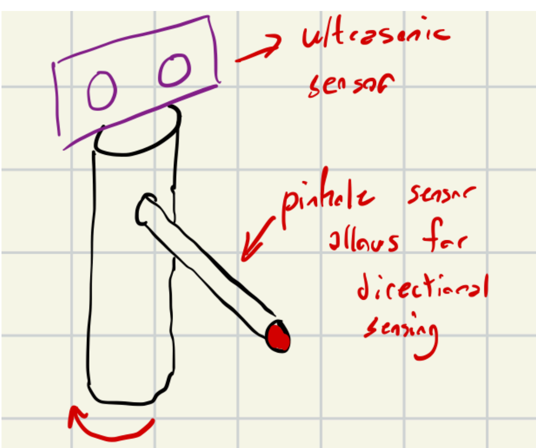
Eyes

The beacon locator system is designed to guide the robot back to its home base, and one promising concept is the “eyes” design. In this setup, two IR sensors are mounted side-by-side with a substantial divider between them to prevent interference. This divider ensures that the sensor facing directly toward the IR beacon registers a higher reading than its counterpart. The robot rotates the sensor assembly until both IR sensors report the same value, indicating they’re perfectly aligned with the beacon. At that point, an ultrasonic sensor mounted atop the swiveling “eye” assembly kicks in to measure the distance to the beacon. This blend of directional IR sensing and ultrasonic distance measurement allows the robot to precisely orient itself and navigate efficiently back to the base. To avoid issues with tangled wires during continuous rotation, a slip ring is incorporated, enabling the sensor assembly to pivot indefinitely.



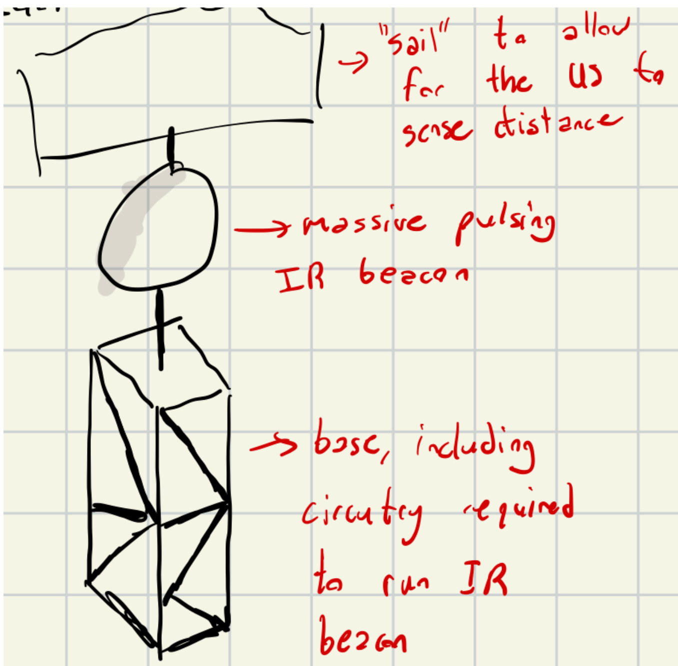
__Pinhole__

Another concept for the beacon locator system employs the pinhole method for enhanced directional sensing. In this design, an IR sensor is housed within a narrow tube—akin to a pinhole aperture—which constrains its field of view. This limited angle of detection allows the sensor to focus on the IR beacon with a high degree of angular precision, similar to the method demonstrated in Lab 4. The sensor assembly rotates until it registers a peak IR signal, confirming direct alignment with the beacon. Once aligned, an ultrasonic sensor mounted atop the rotating platform activates to measure the distance to the target. This dual-sensor approach—combining focused IR detection with precise ultrasonic ranging—ensures accurate localization of the home base. Additionally, the lightweight design of the pinhole setup, coupled with the use of a slip ring for continuous rotation without twisting wires, underscores the system's technical robustness and practicality.



Beacon System - Beacon

The beacon system is key to guiding the scavenger robot back to its base. Instead of using the standard lab kit's weaker IR components, this design uses a large, high-intensity pulsing IR LED to send a much stronger signal—one that the robot can detect from farther away. Right above the IR LED, there's a large cardboard "sail" that acts as a reflective surface for the robot's ultrasonic sensor. This setup gives the sensor a clear target, making it easier to measure the distance between the robot and the beacon once the IR signal is locked on. The beacon's base houses all the necessary circuitry to power and control the pulsing LED, ensuring a steady and reliable signal. By pairing a robust IR signal with a clearly detectable ultrasonic target, this beacon design boosts both detection range and reliability, helping the robot navigate accurately within its 25 m² operating area.



Electrical System

MSE Arduino Board

The MSE Arduino Board is a practical and efficient choice for the electrical system since it is already available and familiar, reducing the need for additional hardware or extensive learning. Since we have previous experience working with it, integration into the robot's design will be straightforward, allowing more time to focus on functionality rather than troubleshooting unfamiliar components. The board provides reliable performance, easy programming, and compatibility with various sensors and actuators, making it well-suited for controlling the collection, sorting, and navigation systems. Additionally, it offers built-in communication capabilities for interfacing with components like ultrasonic sensors, IR beacons, servos, and motor controllers, ensuring smooth operation.

Custom PCB

A custom PCB would provide the cleanest and most reliable electrical design for the project, integrating all necessary circuits onto a compact and organized board. This minimizes wiring clutter, reduces connection failures, and enhances durability. Additionally, it allows for precise routing and optimization of power distribution, improving overall system efficiency. However, a custom PCB is costly and requires significant lead time, making modifications difficult if design changes are needed later. While it offers a professional and space-saving solution, it may not be the most flexible option for rapid prototyping.

Custom Perfboard

A custom perfboard provides a practical middle ground between a breadboard and a PCB, allowing for permanent soldering of circuits while still offering some flexibility for modifications. This option eliminates loose connections and wiring issues commonly found in breadboards, ensuring a more stable and reliable electrical system. Since perfboards can be manually assembled, there is no lead time required, making them a cost-effective and adaptable choice for the project. Additionally, errors can be easily corrected by desoldering and reconfiguring the layout, making it a suitable option for transitioning from prototype to a more durable, long-term solution.

Breadboard

A breadboard is the simplest and most adaptable option for testing and prototyping the electrical system. It allows for quick modifications, troubleshooting, and iterative design changes, making it ideal for early development and circuit experimentation. However, breadboards do not provide the most secure connections, and frequent handling can cause loose wires and unreliable contact points, leading to potential malfunctions. While useful for initial circuit testing, a more permanent solution like a perfboard or PCB would be necessary to ensure long-term reliability for the final robot design.

Control and Navigation System

Triangulation

The robot uses triangulation-based navigation to follow a set path with impressive accuracy, which is key for efficient marble collection and precise movement. In real-world settings, though, factors like uneven surfaces, slight motor glitches, or drift can throw it off course. To handle these challenges, the robot relies on a closed-loop feedback system that pulls data from both an IR beacon and ultrasonic sensors. At strategic points along its route, it uses inverse kinematics to calculate the beacon's angle and determine the distance from obstacles in real time, allowing it to adjust its course on the fly. This constant tweaking helps keep the robot reliably on track and minimizes movement errors.

Ultrasonic Collision Prevention

The robot uses ultrasonic sensors on its front, left, and right sides to help it navigate safely. These sensors keep an eye on the surroundings, ensuring the robot stays a safe distance from walls and other obstacles. This setup minimizes unexpected collisions and promotes smooth movement. Although the sensors might occasionally miss objects with unusual surfaces or at sharp angles, proper placement and calibration can boost their accuracy. Overall, this real-time detection system is crucial for the robot to operate independently.

Return to Home

The Return to Home function ensures that the robot successfully completes its collection process and returns to deposit the marbles before the 120-second time limit expires. Initially, the robot will navigate freely, collecting marbles until the remaining time reaches 30 seconds. At this point, the Return to Home procedure activates, guiding the robot back to the beacon for a precise drop-off. This feature prevents the risk of failing to unload collected marbles on time, ensuring maximum efficiency in the collection process. However, the effectiveness of this method depends on accurate localization and reliable sensor feedback, as errors in path planning or beacon tracking could delay the return process.

By integrating precise navigation, obstacle avoidance, and a structured return strategy, the robot will be able to operate efficiently, avoid collisions, and complete its collection and drop-off process within the required 120-second limit.

Concept Evaluation

Mobility System				
Constraint	**Acceptance Criteria**	**Omni-Directional Wheels**	**Differential Drive**	**Articulated Steering**
Maneuverability	Enables lateral, forward, backward, and rotational movement	GO	NO-GO	GO
Speed/Time Efficiency	Covers a 25 m² area within 120 seconds	GO	GO	GO
Weight & Size	Lightweight and compact; fits within locker dimensions	GO	GO	GO
Sensor Integration	Provides reliable encoder feedback and supports sensor mounting	GO	GO	GO
Safety & Control	Allows precise control to avoid obstacles	GO	NO-GO	GO

Rationale:

- **Omni-Directional Wheels:**
  - *Pros:* Offer excellent maneuverability with the ability to strafe (move laterally) and execute precise turns, which is critical for navigating a confined 25 m² area.
  - *Cons:* Tend to be bulkier and may push the overall design outside of locker size limits.
- **Differential Drive:**
  - *Pros:* Typically more compact and simpler in construction, which is beneficial for fitting within size constraints.
  - *Cons:* Lacks lateral movement; can result in less precise control (especially during zero-radius turns), increasing the risk of collisions.
- **Articulated Steering:**
  - *Pros:* Offers good maneuverability with a controlled pivot mechanism while remaining relatively compact.
  - *Cons:* Although viable, the added mechanical complexity requires precise calibration for smooth turns.

Collection System				
Constraint	**Acceptance Criteria**	**Rock Picker**	**Vacuum Collector**	**Claw-Based Pickup**
Collection Efficiency	Rapidly collects multiple marbles in one pass	GO	GO	NO-GO
Reliability (No Jamming)	Mechanism resists jamming during operation	GO	NO-GO	GO
Integration & Complexity	Integrates seamlessly with other systems; minimal complexity	GO	NO-GO	NO-GO
Cost	Additional cost remains under budget (≤ $50)	GO	NO-GO	GO
Maintenance	Requires minimal calibration and upkeep	GO	NO-GO	GO

Rationale:

- **Rock Picker:**
  - *Pros:* Efficiently gathers multiple objects simultaneously, is relatively simple, and fits within the cost constraints.
  - *Cons:* Must be carefully designed to avoid jamming with debris; overall, it received “GO” if designed with proper guards.
- **Vacuum Collector:**

- *Pros:* Offers non-contact collection, which might be gentle on objects.
    - *Cons:* Typically requires high-powered components that can push costs above $50, has higher risks of jamming, and may complicate sensor integration (especially for inline color sensing).
- **Claw-Based Pickup:**
    - *Pros:* Provides precise control when picking up objects.
    - *Cons:* Often slower (collecting one or two objects per cycle), which may risk exceeding the 120-second limit, and struggles with handling multiple marbles simultaneously.

Sorting System					
Constraint	**Acceptance Criteria**	**Spiral Sorter**	**Auger Sorter**	**Conveyor Sorter**	**Zipper**
Sorting Accuracy	Reliably differentiates valuable vs. non-valuable objects	GO	GO	GO	GO
Speed	Processes objects quickly to maintain a 120-second cycle	GO	NO-GO	GO	GO
Reliability (Avoid Jamming)	Mechanism minimizes risk of jamming	GO	NO-GO	NO-GO	NO-GO
Sensor Integration	Easily integrates with the color sensor for classification	GO	GO	GO	GO
Cost & Complexity	Remains simple and cost-effective	GO	NO-GO	NO-GO	NO-GO

Rationale:

- **Spiral Sorter:**
    - *Pros:* Provides continuous sorting in a compact design, and its integration with a color sensor can be relatively straightforward.
    - *Cons:* Must be engineered to minimize jamming, but overall its speed and accuracy make it the preferred option.
- **Auger Sorter:**
    - *Pros:* Can sort objects based on continuous rotation.
    - *Cons:* More prone to jamming if objects get misaligned, and its slower operation may not meet the 120-second constraint.

- **Conveyor Sorter:**
  - *Pros:* Offers continuous movement and sorting.
  - *Cons:* Generally involves more complex mechanical systems, increasing both cost and risk of jamming.

Dropping System				
Constraint	**Acceptance Criteria**	**Front Bucket**	**Rear Bucket**	**Side Door**
Drop-Off Efficiency	Quickly and reliably releases collected objects	GO	GO	GO
Safety	Prevents spillage and ensures safe deposit	GO	GO	NO-GO
Beacon Integration	Activates in sync with the beacon signal without delays	GO	NO-GO	GO
Ease of Use	Simple mechanism requiring minimal calibration	GO	GO	NO-GO
Cost	Remains within budget	GO	GO	GO

Rationale:

- **Front Bucket:**
  - *Pros:* Offers rapid and controlled object release and is straightforward to integrate with the beacon signal for drop-off.
  - *Cons:* Minimal issues, hence rated “GO” across most criteria.
- **Rear Bucket:**
  - *Pros:* Can balance weight distribution by placing the drop-off at the rear.
  - *Cons:* May require additional maneuvers (such as a 180° turn), potentially delaying the process and complicating beacon integration.
- **Side Door:**
  - *Pros:* Provides an alternative drop-off method that could simplify some routing.
  - *Cons:* Has potential safety concerns, as lateral drop-off might lead to spillage or misalignment with the collection container.

Beacon System			
Constraint	**Acceptance Criteria**	**IR "Eyes"**	**Pinhole IR**

		Configuration	Setup
Detection Range	Detects beacon from a sufficient distance	GO	GO
Accuracy	Accurately determines angle and distance	GO	GO
Integration with Ultrasonic	Easily pairs with an ultrasonic sensor for distance measurement	GO	GO
Calibration Ease	Requires minimal alignment adjustments	GO	NO-GO
Cost	Cost-effective solution	GO	GO

Rationale

- **IR "Eyes" Configuration:**
  - *Pros:* Provides robust detection with a wider field of view, easier calibration, and reliable integration with an ultrasonic sensor.
  - *Cons:* May slightly increase component count but still fits within cost constraints.
- **Pinhole IR Setup:**
  - *Pros:* Offers high directional accuracy by limiting the field of view.
  - *Cons:* Requires precise calibration and alignment, which can complicate setup and maintenance.

Electrical System					
Constraint	**Acceptance Criteria**	**Custom PCB**	**Custom Perfboard**	**Breadboard**	**MSE Arduino Board**
Durability	Provides robust, long-term connections	GO	GO	NO-GO	GO
Organization	Ensures clear, tidy wiring and circuit layout	GO	GO	NO-GO	GO
Cost	Additional cost remains under budget (≤ $50)	GO	GO	GO	GO
Flexibility	Allows for easy modifications/upgrades	NO-GO	GO	GO	GO
Compactness	Fits within the overall	GO	GO	NO-GO	GO

	design constraints				

Rationale:

- **Custom PCB:**
    - *Pros:* Offers a compact, durable solution with neat routing and reliable connections, ideal for long-term deployment.
    - *Cons:* Less flexible for modifications once fabricated.
- **Custom Perfboard:**
    - *Pros:* Provides a balance between durability and flexibility, allowing for adjustments during prototyping.
    - *Cons:* Slightly bulkier than a PCB but still acceptable.
- **Breadboard:**
    - *Pros:* Highly flexible and useful during early prototyping.
    - *Cons:* Not durable or compact enough for final deployment, making it unsuitable for long-term use.
- **MSE Arduino Board:**
    - *Pros:* Already well built, contains all needed electrical components, and no cost, as it has already been built and paid for in the past.
    - *Cons:* May be a little large.

Control/Navigation System			
Constraint	**Acceptance Criteria**	**Triangulation (IR/Ultrasonic)**	**Alternative (e.g., SLAM)**
Navigation Accuracy	Reliably guides the robot to the beacon and avoids obstacles	GO	GO
Real-Time Processing	Processes sensor data quickly enough for dynamic adjustments	GO	NO-GO
Implementation Complexity	Integrates with ESP32 with minimal additional complexity	GO	NO-GO
Robustness	Performs reliably under varying environmental conditions	GO	GO
Cost	Remains cost-effective	GO	GO

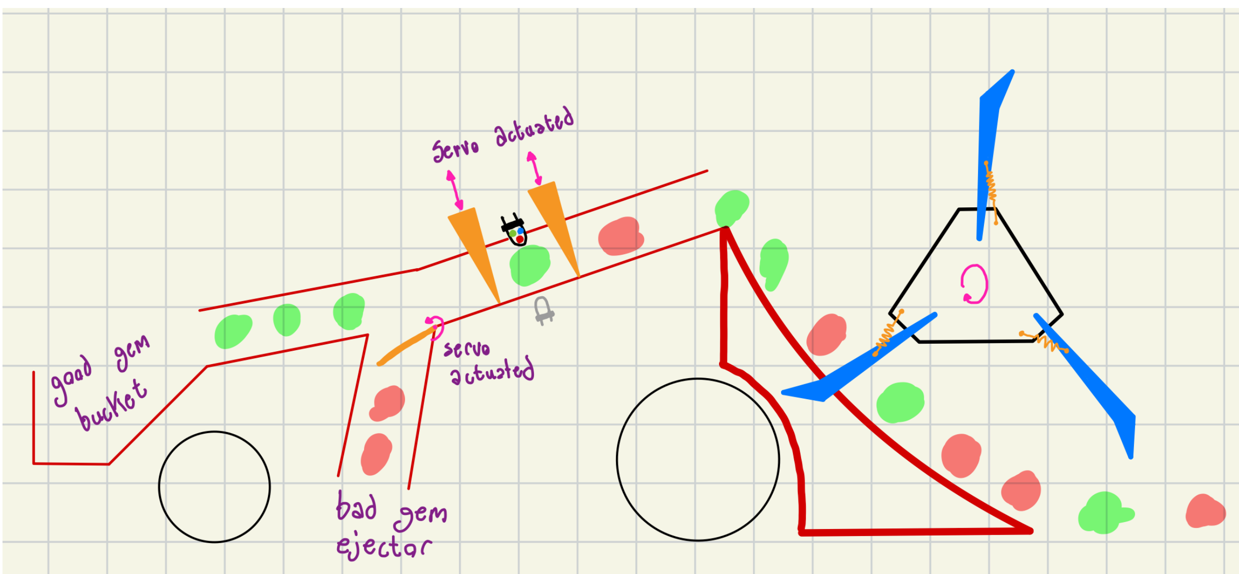
Rationale:

- **Triangulation (IR/Ultrasonic):**

    - *Pros:* Provides a straightforward, low-complexity method to accurately determine the beacon's location. Its reliance on well-understood sensor technology (IR and ultrasonic) makes it robust and cost-effective.
    - *Cons:* Requires proper sensor calibration but is generally easier to implement in real time.
- **Alternative (e.g., SLAM):**
    - *Pros:* Offers advanced navigation capabilities and could potentially adapt to more complex environments.
    - *Cons:* Requires more processing power and increases implementation complexity, which may not be feasible given the ESP32's resources and the need for real-time performance.

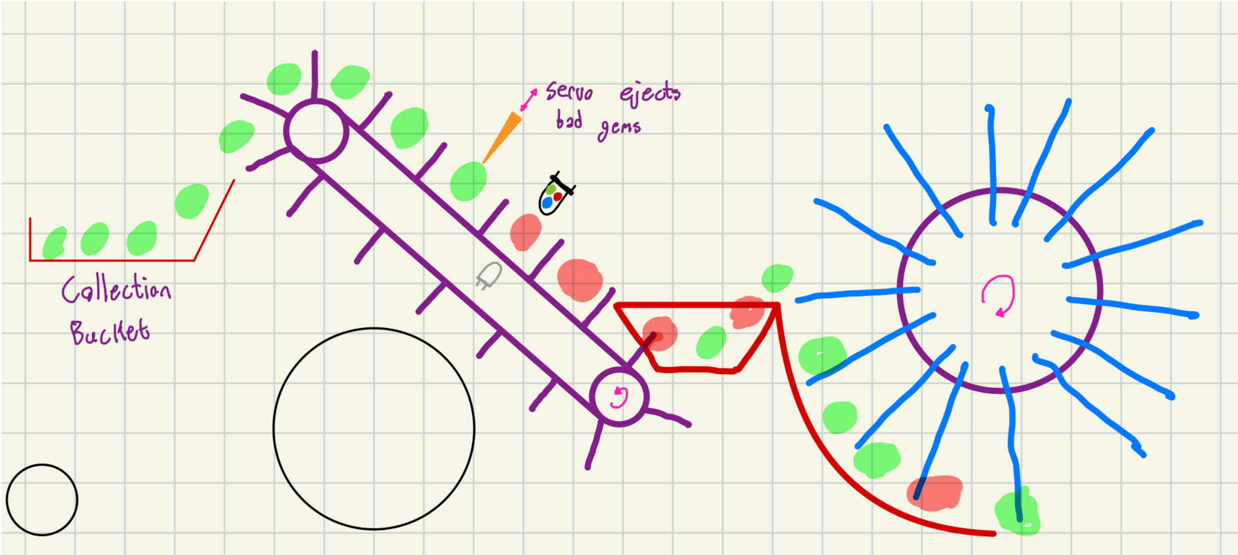
Full Designs

1 - Omni Directional Wheels, Zipper Funnel , Rock Picker



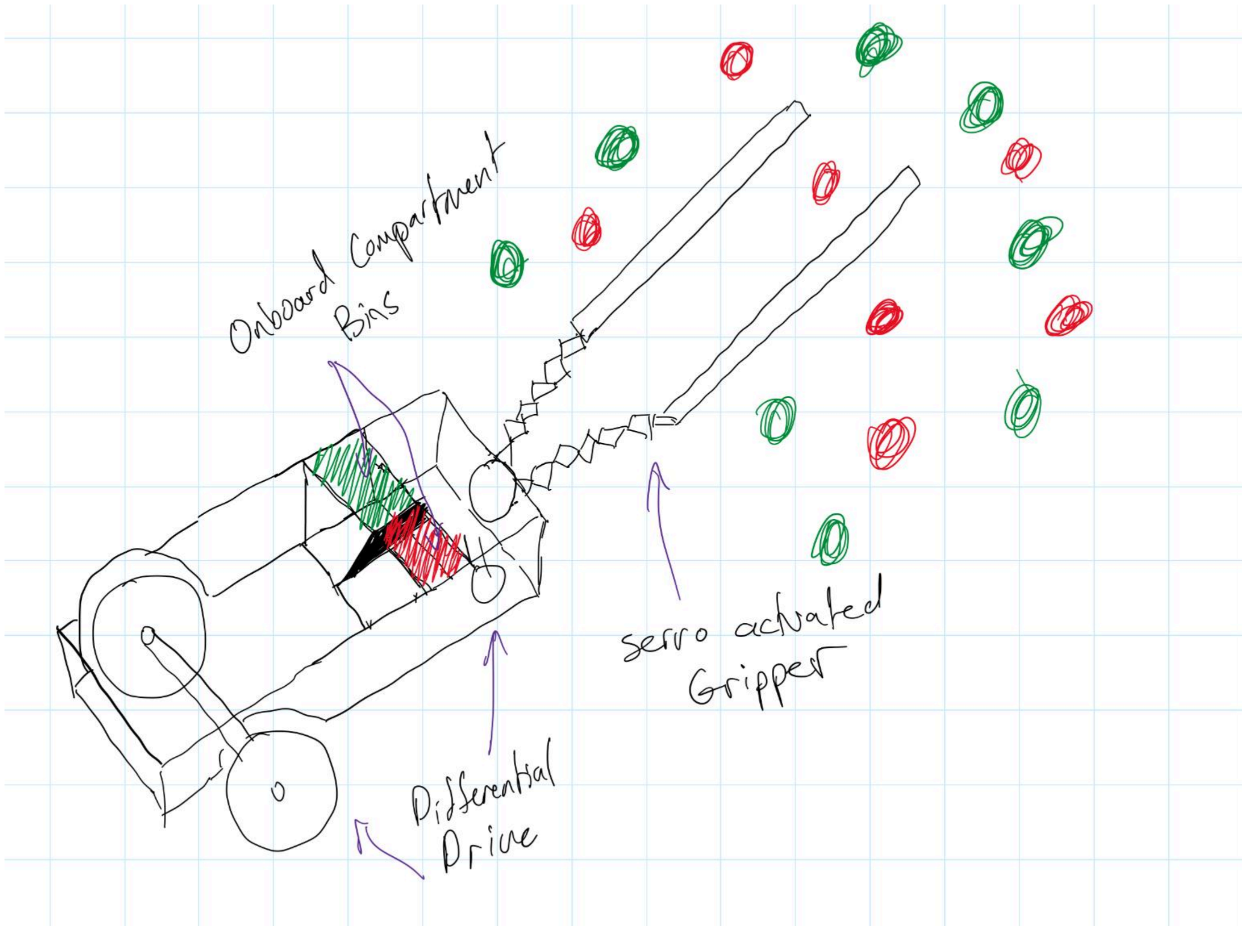
The robot employs omni-directional wheels for agile, multi-directional movement, enabling quick navigation within a 25 m² area. A rock picker mechanism rapidly gathers multiple objects simultaneously, while a zipper funnel system sorts them based on color, ensuring only high-value items are collected. This integrated design meets the constraints for autonomous operation, time efficiency, and cost-effectiveness.

2 - Street Sweeper, Conveyer Sorter, Differential Drive



This design uses a differential drive for straightforward, robust navigation in confined spaces. A street sweeper mechanism with a rotating brush efficiently gathers objects, while a conveyor sorter directs items based on color for continuous sorting. The integrated system is simple, cost-effective, and meets the operational requirements for autonomous performance.

3 - Differential Drive, Servo-Actuated Gripper, Onboard Dual-Compartment Sorter



This design employs a differential drive for reliable, straightforward navigation. A servo-actuated gripper selectively collects objects, while an onboard dual-compartment sorter efficiently separates high-value items from others. Together, these subsystems enable precise and autonomous operation within the 120-second task window.

Product Design Phase

Bill Of Materials:

Bill Of Materials			
Item	Website	Quantity	Price
Mecanum Wheel Omni-Directional	[AliExpress]	4	$4.93
Prusament Filament (1,195g)	Estimated Cost by program shown in images	1,195 g	$11.83
DC Motors	Provided	8	Provided

Servo Motors	Provided	2	Provided
Battery	Provided	1	Provided
Arduino	Provided	1	Provided
Colour Sensor	Provided	1	Provided
Wires	Provided	Unknown	Provided

Product Selection Reasoning:

Omni-Directional Wheels:

Omni-directional wheels are the best choice because they allow for instant movement in any direction, unlike articulated or differential drive systems. Articulated steering relies on a centralized pivot, adding mechanical complexity and potential stability issues, while differential drive requires a turning radius, limiting agility. In contrast, omni-wheels enable seamless strafing and precise control, allowing the robot to navigate tight spaces and reposition effortlessly without wasted movement. Additionally, omni-wheels provide greater accuracy, as other systems depend on precise motor control and encoder feedback to counteract drift. This makes omni-directional wheels the most efficient and reliable option for smooth, controlled movement.

Wheel Shroud

A wheel shroud will be implemented to protect the wheels and avoid any objects getting caught.

Setup/Coding

Each wheel will be set up at a 45 degree angle and powered by one DC motor with an encoder. We will require a motor driver and use arduino boards to power all four wheels. Each wheel will follow an equation breakdown.

The equations below illustrate the movement patterns for moving forward, strafing right, and rotating clockwise:

Front Left = Y+X+Z
Front Right = Y−X−Z
Back Left = Y−X+Z
Back Right = Y+X−Z

Y = Forward/Backward Movement
X = Left/Right Strafing
Z = Rotation (CCW/CW)
+ = Wheels move forward
- = Wheels move backward

Motion	X	Y	Z	Front Left	Front Right	Back Left	Back Right

Forward	0	100	0	100 + 0 + 0 = **100**	100 - 0 - 0 = **100**	100 - 0 + 0 = **100**	100 + 0 - 0 = **100**
Backward	0	-100	0	-100 + 0 + 0 = **-100**	-100 - 0 - 0 = **-100**	-100 - 0 + 0 = **-100**	-100 + 0 - 0 = **-100**
Strafe Right	100	0	0	0 + 100 + 0 = **100**	0 - 100 - 0 = **-100**	0 - 100 + 0 = **-100**	0 + 100 - 0 = **100**
Strafe Left	-100	0	0	0 - 100 + 0 = **-100**	0 + 100 - 0 = **100**	0 + 100 + 0 = **100**	0 - 100 - 0 = **-100**
Rotate CW	0	0	100	0 + 0 + 100 = **100**	0 - 0 - 100 = **-100**	0 - 0 + 100 = **100**	0 + 0 - 100 = **-100**
Rotate CCW	0	0	-100	0 + 0 - 100 = **-100**	0 - 0 + 100 = **100**	0 - 0 - 100 = **-100**	0 + 0 + 100 = **100**

Triangulation:
Our method uses two IR sensors mounted at the front of the chassis to detect a beacon's signal. Each sensor measures the distance to the beacon, and with the known fixed separation between them, we apply triangulation using the law of cosines. This calculation determines both the position and the angle of the beacon relative to the robot's center. Calibration ensures that the sensor readings accurately translate into physical distances. With the computed angle and distance, the robot can steer directly toward the beacon for efficient navigation and alignment within its operating area.

Setup/Coding
The code is structured to continuously read analog inputs from the two IR sensors, convert these readings into physical distances using calibrated conversion factors, and then compute the beacon's position relative to the robot. Using the known distance between the sensors, the code applies trigonometric formulas (specifically, the law of cosines and the Pythagorean theorem) to calculate the x and y coordinates of the beacon. Finally, it determines the beacon's angle using the atan2 function and outputs this value to the serial monitor for further processing, such as guiding the robot toward the beacon.

Spiral Sorter
Our sorting mechanism consists of a spiral sorter that transports marbles along a circular path vertically in elevation. As the marbles move along the wheel, they pause in front of a color sensor, which determines the marble's color. Based on the detected color:

- If the marble is green, a corresponding trap door opens, allowing the marble to be collected in a bin inside the robot.

- If the marble is red, it is dropped off the robot through an exit chute.

The system is controlled using an ESP32 microcontroller, with a DC motor with an encoder driving the wheel. A color sensor is used to detect marble colors. The trapdoor mechanism is controlled using a servo motor.

Setup/Coding

The Arduino code controls a spiral sorter system that moves marbles, detects their color, and sorts them accordingly. The motor advances in steps, positioning each marble in front of the color sensor . The sensor reads the RGB values of the marble, and based on predefined thresholds, determines whether the marble is green or red. If the marble is green, a servo motor-controlled trap door opens to drop it into a bin inside the robot. If the marble is red, another trap door opens fully, allowing the marble to be discarded off the robot. The wheel then moves to align the next marble, and the process repeats. The system ensures precise positioning using stepper motor control, while delays are used to allow marbles to settle before scanning and sorting. Serial prints help with debugging by displaying real-time sensor readings and sorting decisions.

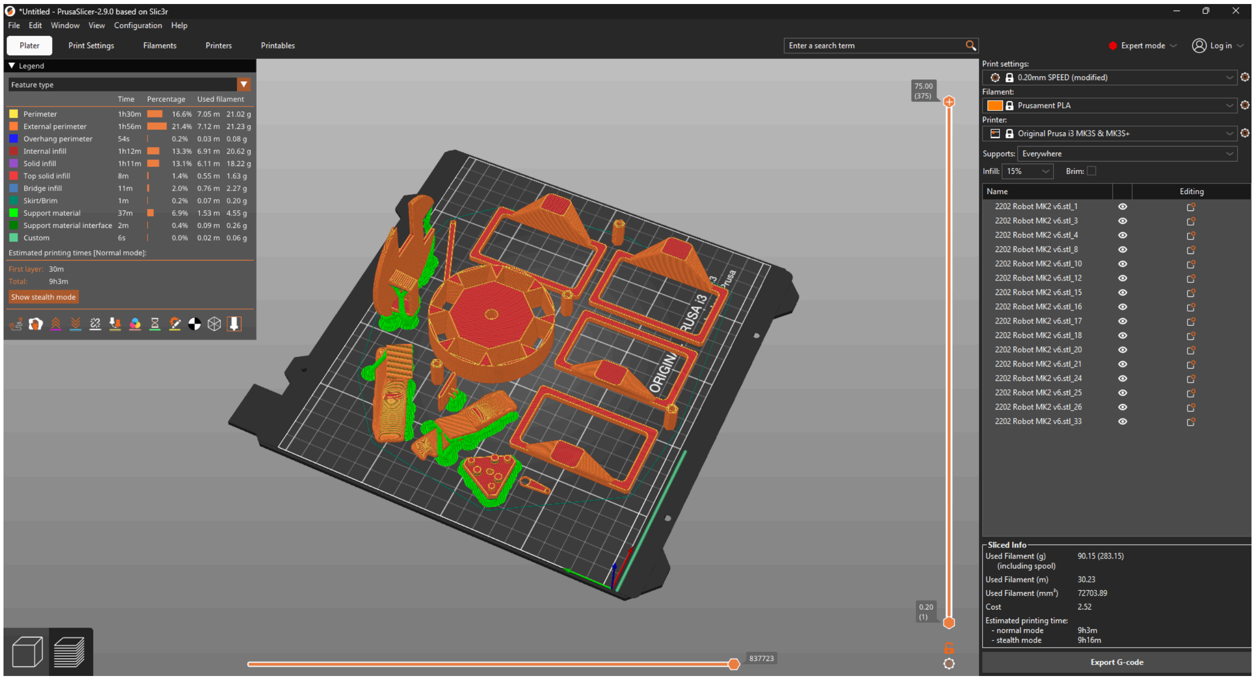
Object Transport - Funnel

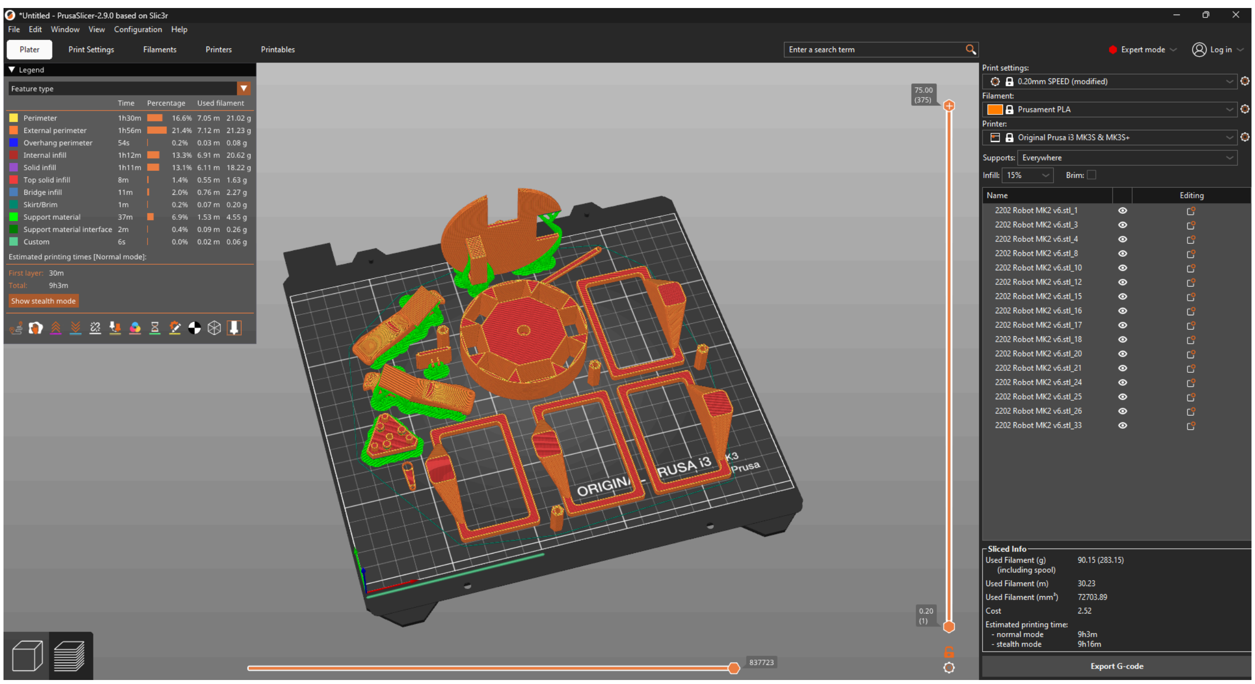
For the funnel design, we iterated through 3D printed test pieces to refine its shape, ensuring a smooth, efficient transition of objects into the collection container. Based on our tests, we optimized the funnel's geometry to minimize blockages. In addition, we plan to incorporate a vibration motor at the base of the funnel to help dislodge any objects in case of a jam, ensuring the system remains reliable during operation.

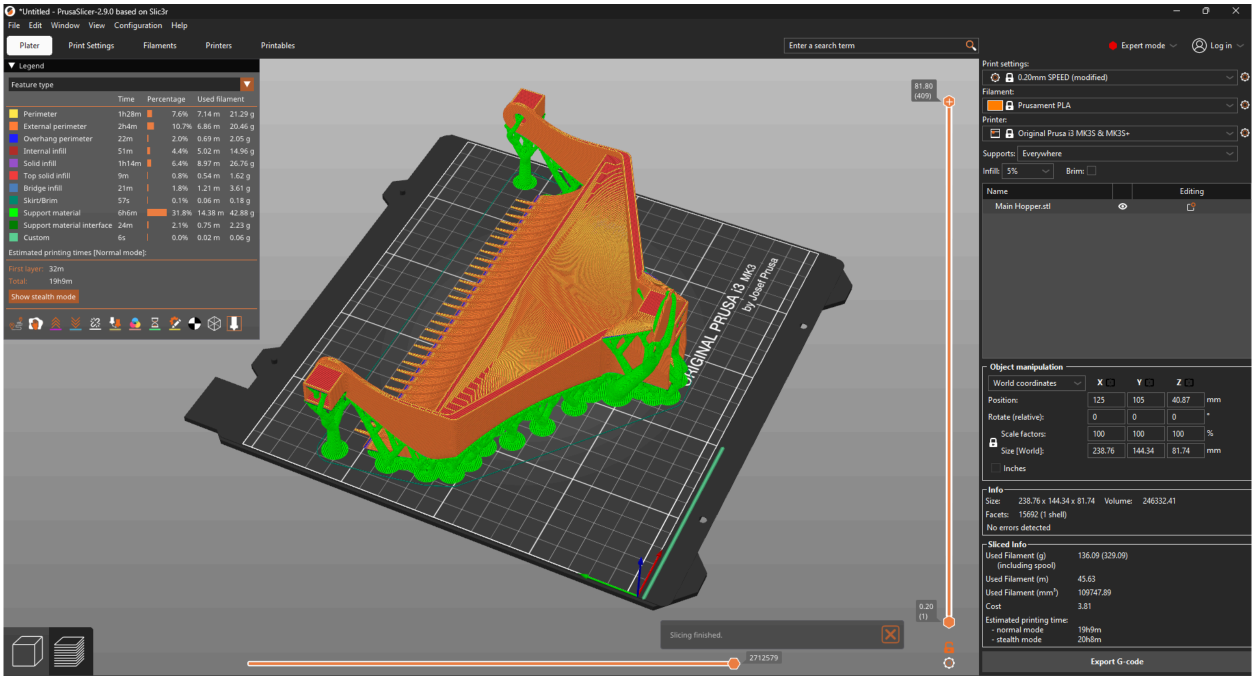
3D Printing Info

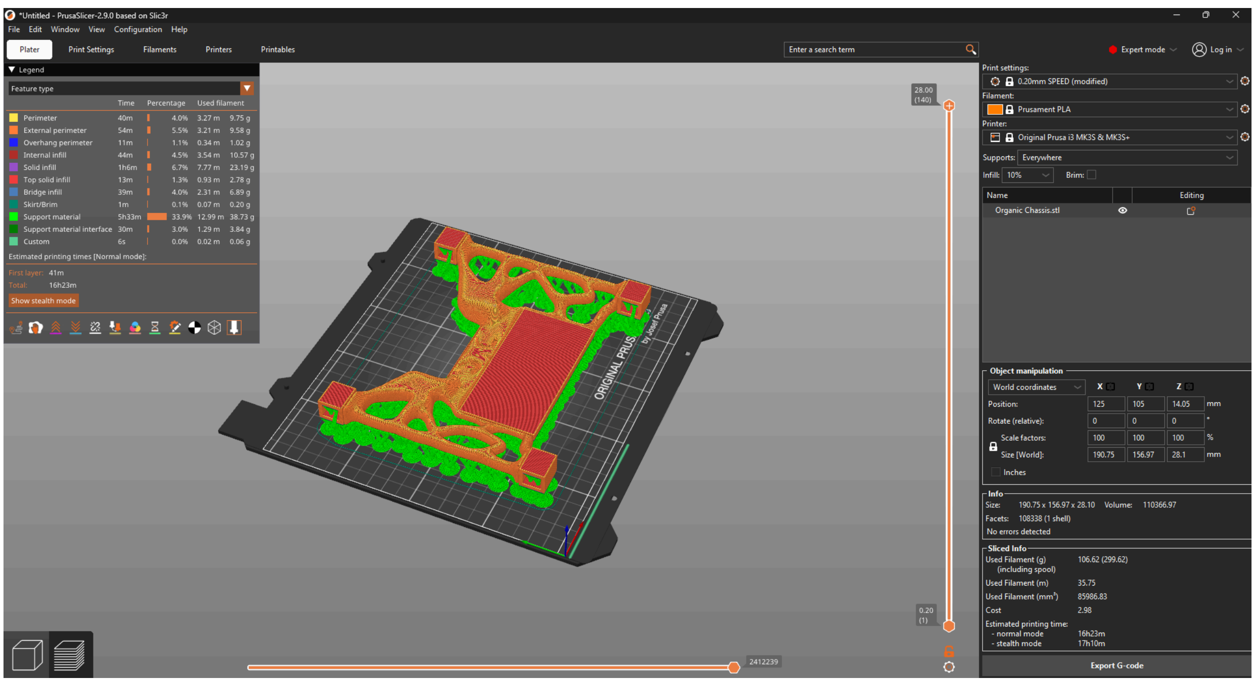
We plan to 3D print our design using an FDM printer, carefully optimizing the print settings to balance strength, precision, and efficiency. To achieve a smooth surface finish in critical areas, we will implement variable layer heights, allowing for higher resolution where necessary while maintaining faster print speeds in less detailed regions. Given the complexity of our design, we also plan to take advantage of a beta feature: organic support. This will allow us to print the chassis as a single cohesive piece rather than assembling multiple parts, reducing weak points and ensuring better structural integrity. By minimizing the need for post-processing and assembly, this method will improve the overall durability and quality of our final product.

3D Printing Images:









After finalizing our design choice, and 3D printing the chassis, attaching the omni-directional mecanum wheels, and wiring the motor on ESP-32, we successfully implemented our drive system.

The drive system uses a holonomic control algorithm based on omni-directional (mecanum) wheels to achieve precise movement in all directions. The code includes routines for motor initialization, encoder-based distance measurement, and movement functions (forward/backward, strafing, and rotation) to enable a square-pattern motion test.

Key features include:

- **Motor & Robot Structures:** Data structures to store GPIO pin assignments, PWM channels, and encoder positions.
- **Initialization Functions:** Routines to configure GPIOs, attach PWM channels, and set up encoder interrupts.
- **Movement Functions:** Implementations of moveRobotDistance(), strafeRobotDistance(), and rotateRobotAngle() using encoder feedback.
- **Control Loop:** A main loop that executes a series of movements (forward, strafe, rotate) in a square pattern with pauses for stabilization.

Explanation

- **Initialization:**
  The code initializes the motor and robot structures, sets up PWM for motor control, and attaches encoder interrupts.
- **Movement Functions:**
  Functions like moveRobotDistance(), strafeRobotDistance(), and rotate() use encoder feedback to move the robot accurately for a specified distance or angle.
- **Holonomic Control:**
  The setRobotMovement() function uses the mecanum wheel equations to calculate individual wheel speeds based on desired X (strafe), Y (forward/backward), and Z (rotation) inputs.
- **Control Loop:**
  In the loop() function, the robot executes a series of movements (square pattern) with delays between actions and uses a state variable (motionStage) to track progress.

This code represents the core drive system implementation for Milestone 3. It has been integrated into the Product Development File to demonstrate our approach to achieving autonomous motion within the project constraints. What made this implementation and testing so successful was that the robot was incredibly reliable, robust, and could consistently repeat specified maneuvers. All this testing was done before beacon implementation, so the fact that the drive system showed such great results made us optimistic going forward, and presented a backup solution, should the beacon system be unsuccessful. We would have the robot follow a set path by simply hard coding the route, and because it was so reliable in getting back to its original spot, we would not have to worry about any mechanical constraints.

Threshold Determination for the Colour Sens

The goal is to reliably detect green objects using the TCS34725 colour sensor mounted on our spiral sorter. Our approach involved iterative testing—observing the sensor's output on the serial monitor as objects were passed through the sorter. This qualitative method allowed us to quickly adjust thresholds based on real-time performance, ensuring that our system could confidently differentiate green objects from others.

Methodology

<u>Experimental Setup</u>

- Hardware:
    - Sensor: TCS34725 colour sensor
    - Microcontroller: ESP32 (custom I2C pins: SDA on GPIO17, SCL on GPIO18)
    - Sorter: Spiral sorter integrated into the robot chassis
- Operating Conditions:
    - Testing was performed in the lab with controlled lighting.
    - Objects of known colors (green, blue, red etc.) were used to evaluate sensor performance.

<u>Iterative Testing Approach</u>

- Real-Time Observation:
  The sensor was mounted on the spiral sorter, and objects were launched through it. The serial monitor provided real-time readings from the sensor's red, green, blue, and clear channels.
- Threshold Adjustment:
    - Initially RGB values were recorded from gems to try and and create appropriate thresholds for the amount of green
    - During testing, we observed that a blue object with a greenish shade triggered a false positive, as it showed higher green values than expected. Values of green and blue were close to identical with a slightly higher amount of green colour
    - Through continuous adjustments based on real-time serial feedback, we set a color difference threshold to ensure a sufficient distinction between green and blue. This prevented blue objects from being mistaken for green while still allowing green marbles to be correctly identified.
- Final Successful Code:
  The final code uses a green intensity threshold and a color difference threshold to ensure that the green channel value is sufficiently higher than the red and blue channels. This code is the version that yielded reliable performance in our system.

Design, Assumptions, and Adjustments

<u>Design Considerations:</u>

- ◆ Sensor Placement:
    - The sensor is fixed at a known distance from the objects in the sorter to ensure consistency.
- ◆ Controlled Testing:
    - By testing in a controlled lighting environment, we minimized external light interference.
- ◆ Assumptions:
    - The sensor's response is consistent under the controlled conditions.
    - The range of object colors used in testing is representative of the actual operating environment.
- ◆ Adjustments Made:

- We iteratively modified the thresholds using live serial data until the green detection became robust against variations, including cases where a blue object exhibited unexpectedly high green readings.

Limitations

- Ambient Lighting Effects:
  While testing was done under controlled lighting, the actual operating environment may feature variable lighting conditions.
- Darker Objects:
  The objects in the sorter are generally darker, which means they may reflect less light. This could affect sensor readings and potentially lead to misclassification if the thresholds are not adjusted for lower intensity levels.
- Sensor Drift:
  Over time, sensor performance might change, necessitating periodic recalibration.

Implications

- Reliability in Controlled Settings:
  The final thresholds have been fine-tuned to provide reliable green detection within our controlled test environment.
- Need for Future Adjustments:
  In real-world deployments, where lighting and object reflectivity may vary, additional calibration or dynamic threshold adjustments might be required.
- Confidence Level:
  Despite the qualitative nature of our approach, the iterative testing process has provided a high degree of confidence in the system's performance under our test conditions.

Results and Conclusion

Through continuous testing and real-time adjustments, we have established a set of thresholds that reliably detect green objects in our spiral sorter system. The final code version incorporates these thresholds and has proven successful during demonstrations. This iterative, observation-based approach, although not based on formal statistical analysis, ensured that the sensor operates reliably under our specific conditions.

Engineering Drawings

Electrical Schematics

Testing and Adjustments

Drive System:

Milestone 3 was all about the drive system, and making sure it works. Testing was very successful as the robot was able move in straight lines, make 90 degree turns, and successfully make it back to its original position, only off by a few centimetres, indicating the design was very robust and reliable.

Collecting and Sorting System:

Testing for the collection and sorting system was done around the time milestone 4 was due. The main difficulty in testing these two systems was getting the threshold determination for green (valuable) objects right. After some trial and error, our spiral sorter and colour sensor was able to correctly identify the valuable objects, and discard invaluable objects. The discarding of objects uncovered new challenges, as the objects would get caught on the wheels, despite the wheel shrouds being in place. As the objects come from an elevated surface, the shrouds are rendered useless as they are in place to ward off objects in the wheel's path. Cardboard modifications were added to essentially force the invaluable objects to be discarded in a way that it would not get on top of the wheel shroud.

Return to Home:

The goal was to use three ultrasonic sensors, two at the front of the robot, and one on the beacon, to triangulate the position of the robot. However this proved to be a very challenging task, as the two sensors on the robot would not connect at the same time, and would generate many errors. We modified this task, by instead using an IR sensor, however even that was being difficult to work with, and as the showcase coming up left us in a time constraint, we had to go with our last case scenario plan of hardcoding a path for the robot.

Product Evaluation

Design Summary

Our final design integrates several advanced subsystems:

Drive System:

Utilizes mecanum/omni-directional wheels powered by DC motors with encoder feedback. These motors are controlled via DC motor drivers and managed by an ESP32 on a PCB for precise, omnidirectional movement.

Chassis:

Developed using generative design techniques that incorporate AI-based simulations. This process evaluates stresses and loads to strategically remove non-essential material, reducing the overall weight while maintaining structural integrity.

Collection:

A loader mechanism scoops objects from the ground and transfers them into a collector. The collected items are then passed into the sorting subsystem.

Sorting:

Implements a spiral sorter equipped with a color sensor. When a detected object matches the desired color, a trap door opens, depositing the object into an internal bin. Non-valuable items continue through the sorter and are released back onto the ground at the end of the rotation.

Navigation:

<u>Ideal</u>

The robot uses triangulation based on data from two ultrasonic sensors at the front, combined with a beacon and an additional ultrasonic sensor, to accurately locate and approach objects for collection.

<u>Actual</u>

Robot was hard programmed to a straight line route and then drove backwards to return back to the bucket.

What We Actually Did

Code
When developing the code we did it on a segmented basis

Sorting

Drive