Submitted by

VALERIE LEE SZE ENG (U1920166B)
LEE YUN FAI (U1820770G)
JEE YOU WANG JUSTIN (U1922692A)
MUHAMMED IZZAT BIN FADZLON (U1821785G)
JANESSA LIOW HUI TING (U1922928G)
LI BOWEN (U1822029G)
NYAMTSOGT MUNKHBILGUUN (U1923471A)
ONG ZHEN YU (U1921046A)
RAY PRITHVIRAJ (U1822808G)
YAP ZHI HAN DARREN (U1923981F)

School of Mechanical and Aerospace Engineering
Nanyang Technological University

AY2021/2022

Abstract

Building a robot necessitates a multidisciplinary approach, whether its mechanical for hardware construction or control systems, which delves into the realm of electronics. Because robotics is becoming increasingly popular, experts in the field will be in high demand. Mechatronics engineers are professionals who possess this knowledge in the field. Mechatronics engineers, or mechatronics in general, have grown in prominence around the world as a result of globalization and digitization. MA4012 Mechatronics Engineering Design exposes students to real-world and industrially relevant mechatronics applications in this subject.

A group of ten people must work together to develop a robot with a specific goal in mind. The ultimate goal was to gather the payload, which was a tennis ball in this example, and transport it to the delivery location. For the physical criteria, a robot must be built within the dimensions of 300*300*300(mm). The physical form factor was a set variable; however, members were free to use their imagination to work on the internal components and program so long as the goal is completed. In this report, members also logged down their thinking process and steps taken during the design of this robot as part of the building process. Members made use of their engineering knowledge and applied it directly into this build process. Certain aspect to be part of the team's consideration such the conceptual design, embodiment and principles, diagrams, and as well as the robot physical build.

The ultimate product was a reasonably strong structure with a fortress-like design that ensured the sensors and cargo were adequately protected. Although the design was not perfect, it will have the upper hand in terms of structural strength with some fine tuning and more work. The build process was noteworthy because all the team members put out their best effort to contribute to making this robot completely functional and robust through rigorous brainstorming and hour-long conversations. The robot ultimately fulfilled its objective in the competition where it could collect and deliver a total of 2 balls in one round.

Table of Content

1 INTRODUCTION 8

1.1 OVERVIEW 8

1.2 COMPETITION DESCRIPTION 8

1.3 OBJECTIVES 9

2 DESIGN DEFINITION 9

2.1 IDENTIFYING USER NEEDS 9

2.2 DESIGN SPECIFICATION 11

3 CONCEPTUAL DESIGN 14

3.1 FUNCTION ANALYSIS 14

3.2 MORPHOLOGICAL CHART 15

3.3 DECISION MAKING 25

4 EMBODIMENT DESIGN 30

4.1 CLARITY 30

4.2 SIMPLICITY 32

4.3 SAFETY 32

4.4 PRINCIPLES OF EMBODIMENT DESIGN 33

5 DETAILED DESIGN OF ROBOT 35

5.1 CHASSIS 35

5.2 MAIN HOUSING 36

5.3 COLLECTOR 36

5.4 RAMP 37

5.5 EXPECTED BALL PROJECTILE MOTION 38

6 DESIGN CALCULATIONS 41

6.1 POWER TRANSMISSION CALCULATION 41

6.2 WHEEL 41

6.3 COLLECTOR – DESIGN CALCULATIONS 44

7 CONTROL SYSTEM 49

7.1 NON-CONTACT SENSORS: DISTANCE AND LINE SENSORS 49

7.2 CONTACT SENSOR: OMRON SS-5GL LIMIT SWITCH 52
7.3 DIRECTIONAL SENSOR: DIGITAL COMPASS 53
7.4 PROCESSING UNIT: VEX MICROCONTROLLER 54
7.5 OUTPUT SYSTEM: 3 – WIRE SERVO MOTOR 55
7.6 WIRING DIAGRAM 56

8 PROGRAMMING LOGIC 57

8.1 SOFTWARE ARCHITECTURE 57
8.2 CODE EXPLANATION 60
8.3 ISSUES FACED AND ITERATIVE IMPROVEMENTS 66
8.4 PERFORMANCE REVIEW AND IMPROVEMENTS TO BE MADE 68

9 VALUABLE LESSONS LEARNT 73

10 REFERENCES 74

11 APPENDICES 75

11.1 OVERALL ASSEMBLY DRAWINGS 75
11.2 PART DRAWINGS 78

Table of Figures

Figure 1-1 – Layout of Arena 8
Figure 3-1 - Overall Function Diagram 15
Figure 3-2 - Function Analysis Diagram 15
Figure 5-1 - Drive system and Chassis 35
Figure 5-2 - Main structure for top attachments 36
Figure 5-3 - Ball within the collection boundary 37
Figure 5-4 – Ramp with Built in Pressure Sensor 37
Figure 5-5 - Pressure sensor under holding platform 38
Figure 5-6 - Front gate with flapper in up position 38
Figure 5-7 – Balls is successfully collected 39
Figure 5-8 – Ball in Holding area 39
Figure 5-9 – Pressure sensor triggered when ball in holding area 40
Figure 5-10 – 2 Limit switch sensor and 2 Bottom facing line sensor 40
Figure 6-1 - Free Body Diagram of Ball on Back Ramp 44
Figure 6-2 - Design Calculation of Length of Collector 45
Figure 7-1 – GP2Y0A21 Sharp distance sensor 49
Figure 7-2 – Graph of Analog output voltage VS Distance to reflective object 50
Figure 7-3 – IR Line Sensor 51
Figure 7-4 - Omron SS-5GL Limit Switch 52
Figure 7-5 – Digital Compass Navigation PC Board 53
Figure 7-6 – Digital Compass Lighting Sequence. 53
Figure 7-7 – VEX Microcontroller 54
Figure 7-8 – 3 Wire Servo Motor 55
Figure 7-9 – Wiring Diagram of the Robot 56
Figure 8-1 – Program Flow 57
Figure 8-2 – Functions and main variables used 60
Figure 8-3 – Function to align orientation with delivery area 62
Figure 8-4 – Numeric coding of different directions with the digital compass 63
Figure 8-5 – Task running through out to detect ball or robot in front of proximity sensor ... 64
Figure 8-6 – Function which implements the searching movement of the robot 65
Figure 8-7 – Searching process illustrated 66
Figure 8-8 – The ballClose method before testing 67

Figure 8-9 – Changes made to the ballClose method 67
Figure 8-10 - Collector collecting a ball during the competition....................................68
Figure 8-11 - Aligning with the ball with offset to optimise for ball collection......................69
Figure 8-12 - Ball searching and detection process 69
Figure 8-13 - Edge sensor on the back left giving wrong values.....................................70
Figure 8-14 - Robot collecting 2 balls at once 70
Figure 8-15 - Ball moving in front of the robot, unsuccessful alignment offset 71
Figure 8-16 - Ball stuck behind robot near delivery area 72

Table of Tables

Table 1 - Established relative importance of the needs 11
Table 2 – Morphological Chart 16
Table 3 - Collector system solutions 17
Table 4 - Drive system solutions 18
Table 5 - Vehicle Chassis Design Solutions 20
Table 6 - Delivery system solutions 21
Table 7 - Edge detection solutions 22
Table 8 - Software Program Design Solutions 23
Table 9 – Design Criteria 25
Table 10 – Decision Matrix 29
Table 11 – Functions of each components 31
Table 12 – Weight of Collector and Chassis 41
Table 13 – List of Assumptions when operating the Wheels 41
Table 14 – List of Assumptions when operating Collector 47
Table 15 – Specifications of Sharp Analog Distance Sensor (Long Distance) 50
Table 16 – Specifications of IR Line Sensor 51
Table 17 – Specifications of Omron SS-5GL Limit Switch 52
Table 18 – I/O Ports of VEX Microcontroller 54
Table 19 – Function Descriptions 60

1 Introduction

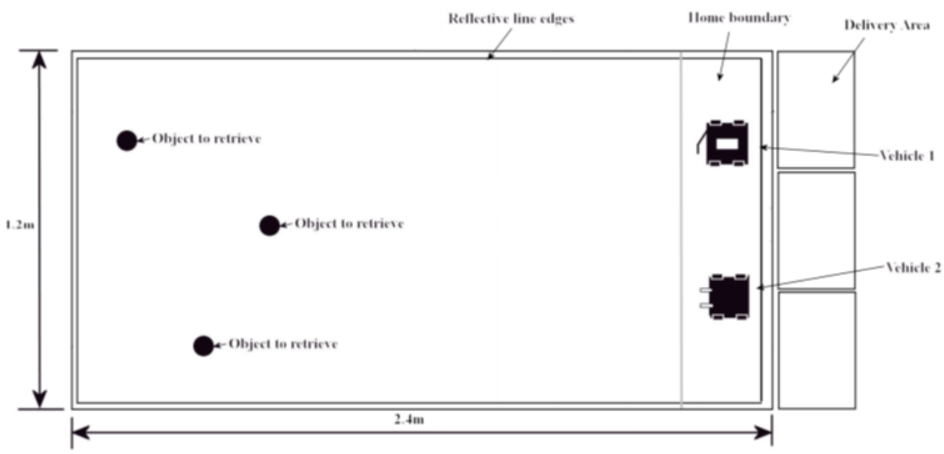
1.1 Overview

Mechatronics is a multidisciplinary field that requires the skill sets in the mechanical, electrical and computerized technologies to develop a simpler and smarter system needed in the advanced automated manufacturing industry.

MA4012 allows students to experience ‘Industrial-like’ mechatronics engineering through a mechatronics design project. The project task requires individual teams to use the approach of the systematic engineering design process which gives students the opportunity to acquire employable mechatronics design skills set.

The assignment was to design and build a compact autonomous robot that can perform basic functions such as edge detection, target tracking, searching and retrieval. The success of the team’s robot design was tested by pitting against other teams in a competition *(Description stated in Section below)*

This report details the product design specifications, systematic design process, design calculations, detailed drawings, and programming logic. The report also includes the group’s reflection and thought process in this project.



[1]

Figure 1-1 – Layout of Arena

1.2 Competition Description

Two robots will be pitted against each other in the arena starting at the home boundary. The robots are tasked to search and retrieve 3 balls that will be randomly placed around the track arena. Only 1 ball is allowed to be retrieved at any one time by a robot and the robots are not to cross the reflective line edges which act as the boundaries of the arena. The robot that retrieves and successfully delivers the most balls into the delivery area wins.

1.3 Objectives

The objective of this mechatronics design project is to allow the students to employ a practical and systematic design approach based on what they learnt in the module, to challenge the students' ability to develop the best autonomous vehicle with the provided equipment and resources within specified constraints. Students are to exhibit their engineering knowledge and project management skills to complete the given assignment. The outcome allows students to acquire relevant mechatronics skillsets in preparation for future career prospects.

2 Design Definition

2.1 Identifying User Needs

The autonomous robot must be designed based on the given set of requirements as follows:

- Dimensions of the autonomous robot are not to exceed 30cm x 30cm x 30cm before the start of the competition
- The footprint of the robot is not allowed to increase after the start of event
- The robot should comprise at least 50% of the given materials
- Power supply to the vehicle is limited to a 6-cell AA rechargeable battery pack provided.
- Boundary detection mechanism needed to prevent robot from exiting arena
- Robot can only collect one ball at any one time
- Robot is to display good manoeuvrability
- Ability to locate the position of the balls
- Retrieval mechanism to collect ball
- Ability to detect the ball once picked up
- Stowing space for ball while robot is moving towards delivery area
- Capability to identify delivery area
- Able to identify difference between ball and opponent
- Ease of assembly, maintenance and placement of electrical wire connections
- Robust structure to cushion opponent's collision impact
- No additional sensors, servos, motors, batteries are allowed over what is provided
- No limitation to the number of limit switches to be mounted on the robot

- Smooth delivery of ball (No throwing or rolling from a distance back to the delivery area)

After identifying the requirements as well as conducting research on previous years' robots used in the competition, it was decided that the robot should have a fast collection speed, easy delivery mechanism and well-placed sensors for ball detection efficiency. As such, these needs were organised into a hierarchical list based on primary or secondary needs.

- Good navigation system
    - Appropriate placement of sensors (edge detection, digital sensors, SHARP sensors)
    - Able to distinguish between various objects
    - Software design able to handle various events
- Reliable ball collection mechanism
    - Ability to detect the ball
    - Smooth ball retrieval process
    - Ability to store the ball
- Reliable ball delivery mechanism
    - Ability to move to the delivery area once ball is collected
    - Smooth ball delivery process
- Ease of hardware maintenance
- Ease of Software troubleshooting
- Compact design
    - Within specified dimensions
    - Not more than 50% of the material is outsourced
- Stable chassis design
    - Low center of gravity and high base area
    - All components are to be tightly fixed in position
- Aesthetically pleasing
    - Colorful
    - Symmetrical

The relative importance of the primary needs is established as in Figure 21.

Table 1 - Established relative importance of the needs

No.	Primary needs	Relative Importance
1	Good navigation system	4
2	Reliable ball collection mechanism	5
3	Reliable ball delivery mechanism	5
4	Ease of hardware maintenance	3
5	Ease of Software troubleshooting	3
6	Compact design	5
7	Stable chassis design	2
8	Aesthetically pleasing	1

2.2 Design Specification

After considerations of the project scope, the different processes of design stages were broken down into individual components and were completed according to their ranking of importance. The key tasks involved were Ball Collection, Ball Delivery, Robot Navigation System, Robot Chassis. The key activity that the robot should be able to perform firstly, is the successful collection of a tennis ball, secondly to transport the ball to the delivery area, and finally to release the ball for a successful delivery. The navigation and autonomous searching for the ball plays a secondary role in aiding the above-mentioned key activities.

2.2.1 Ball collection

Utilizing the sensors to detect the balls, the robot should be able to move autonomously towards the ball after a ball has been detected in range. The robot should slow its speed and creep towards the ball at 0.08m/s, and when the ball reaches a distance of 0.5m from the sensor, the robot will rotate slightly and move forward to align the collection mechanism with the ball. The motors will then begin spinning the collector mechanism to allow the ball to be collected and stored within the chassis of the robot.

The points below outline the key considerations during the collection process:

- The ball collection process should only require one motor.

- The tennis ball must be detected by a sensor, within a range of 0.5m for the robot to begin the collection process. The sensor position should be placed at a suitable height which allows for detection of the tennis ball on the ground.
- The robot must creep towards the ball at 0.08m/s whilst spinning the collector mechanism at a speed of 0.15m/s to allow the ball to be collected smoothly.
- The width of the collector must be larger than the diameter of the tennis ball. In order to increase succession of collection, the collector should be designed to have a width of 90% pertaining to the width of the ramp.
- The robot should feature a mechanism to allow the collector to eject the balls if more than one ball was accidentally collected at the same time. In order to cater for a single ball storage, the storage area of the ramp should be small enough to only fit 1 ball.
- The ramp should be designed in a way that allows the ball to be securely stored in case of any unforeseen impacts or collisions with the opponent's robot. This can be done by installation of a gate using a motor that allows for blockage and release of the ball.
- The height of the ramp should not exceed 70mm to avoid a situation where the tennis ball collides with the top of the robot during the collection process.
- The ramp should feature a limit switch that allows the robot to detect if the ball has been successfully collected before commencement of the delivery process.

2.2.2 Ball Delivery

- The back of the ramp where the ball exits the robot upon successful delivery, should not be placed lower than the height of the delivery area of 70mm.
- Before the start of the delivery process, the limit switches should allow the robot to detect that the ball was collected successfully. The delivery process should be allowed to continue only if the ball is still detected in the robot's storage area.
- After the ball had been collected, the robot should use the programmed compass to align itself towards the direction of the delivery area. Thereafter, the robot should reverse until it reaches the delivery area.
- The robot should account for compass reading inaccuracies which may result in a slanted return path to the delivery area. If this event occurs, the edge sensors of the robot should be able to play a part in re-aligning the robot.
- If the ball was accidentally released during the delivery process, the limit switches should be able to detect this change and halt the delivery process.

- The robot should be able to align itself to the delivery area to allow both limit switches located at the back of the robot to be activated when robot arrives and touches the delivery area. If only 1 limit switch were to be activated, the gate should not release the ball as this event may be triggered during a collision with the opponent’s robot.
- If the robot were to collide with another robot during the delivery process, the robot should be able to re-align itself to continue its delivery process.

2.2.3 Robot Navigation

- The robot should be able to move back and forth in a straight path and make turns in the left and right direction, consistently with a deviation limit of 0.05m per 1m travelled.
- A compass should be used and recalibrated during each round. The compass should allow the robot to navigate around the arena without having a preferred starting position to allow for flexibility, especially for its delivery process.
- The robot should feature sensors at the front for the detection and avoidance of an opponent’s robot. The sensors should be placed at a suitable height to prevent the tennis ball to be mistaken for the opponent's robot.
- The robot will continuously run a 180-degree sweep to check its surroundings. If a ball is detected, the robot will move towards the ball, otherwise the robot will move forward for 1 second to perform another 180-degree sweep.
- Line detection sensors should be placed beneath the robot, and at 4 different corners. This ensures that the robot stays within the yellow tapes, representing the boundaries of the arena. They should be placed no more than 2cm from the ground as they only have a limited detection range of 4cm.
- If the robot was detected to be at the boundary, the robot should move backwards and away from the boundary.

2.2.4 Robot Chassis

- The dimensions of the robot should be within the constraints of 30cm x 30cm x 30cm. The constraints should also be imposed on any extension the collector mechanism may have during the collection process, as well as the gate during the delivery process.
- The chassis should be designed to fit the ramp and collector mechanism. Not the other way round.
- Excessive weight at the top of the robot should be avoided to prevent a higher centre of gravity which may lead to undesirable implications during the competition.

- All sensitive components such as sensors, wirings, battery and microcontrollers should be placed within the chassis and protected to avoid the possibility of damage during a collision with the opponent's robot.
- Cables should be secured with cable ties and labelled near the components to allow for easy maintenance.
- The wheels of the robot should also be protected, if possible, to avoid any navigation issues in case of a collision.

2.2.5 Conditions for materials to be used or manufactured

- Component parts were to be fabricated manually or with a 3D printer when necessary.
- The robot should be constructed with no more than 40% of 3D printed components.
- No glue/epoxy to be used to join components together.

2.2.6 Reliability

- The robot should be able to operate under normal conditions in the area for at least 30 minutes to cater for any restarts during the round.
- The workability of the sensors of the robot should be tested and verified by all members of the team before each round of the competition.
- The workability of the collector and delivery mechanism of the robot should be tested and verified by all members of the team before each round of the competition.
- Testing should proceed by isolating the involved components.
- Bolts and nuts should be retightened before the start of each round of the competition.

3 Conceptual Design

Conceptual design is the second stage of the product design process. It is the stage where we generate potential solutions to the user needs.

3.1 Function Analysis

A Function Analysis Diagram helped to identify the various functions that should be addressed in order to make an autonomous vehicle. The main functions were further divided into several sub-functions design which helped to achieve the main function. Figure 31 shows the Overall Function Diagram. Figure 32 shows the Function Analysis Diagram. The material flow is represented by a black arrow, the signal flow is represented by a dotted line arrow and the energy flow is represented by a dash line arrow.

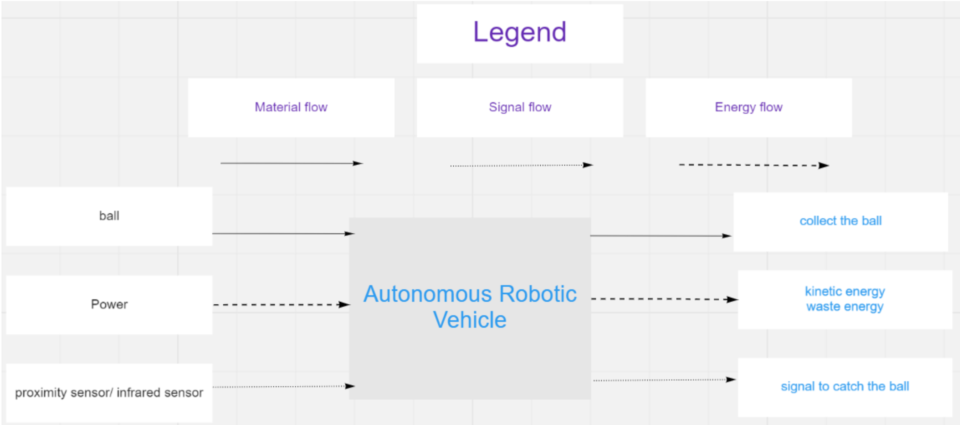


Figure 3-1 - Overall Function Diagram

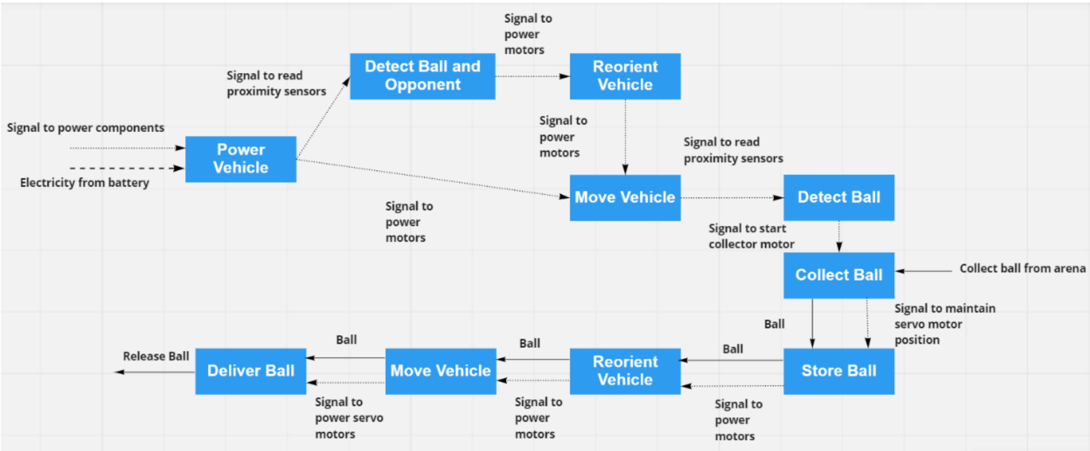
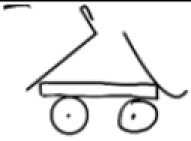
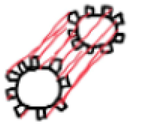
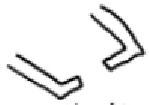
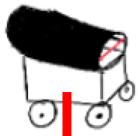
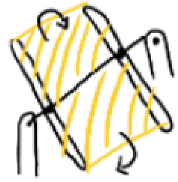
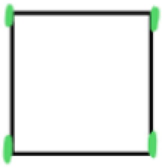
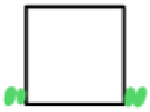
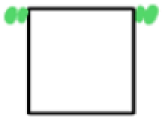
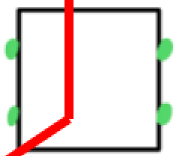
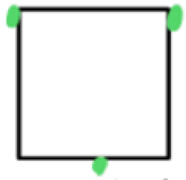
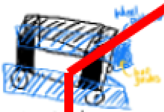
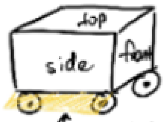
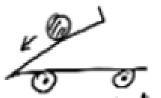
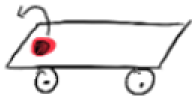
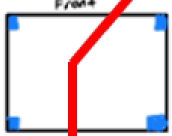
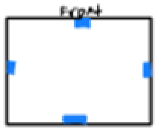
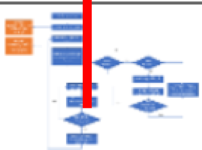
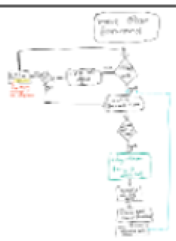
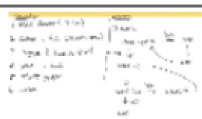


Figure 3-2 - Function Analysis Diagram

3.2 Morphological Chart

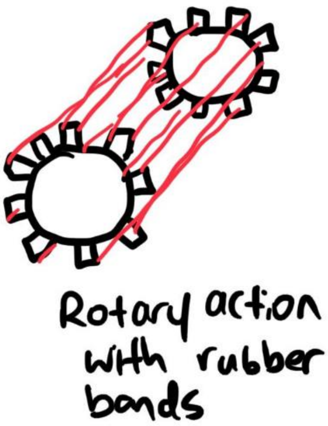
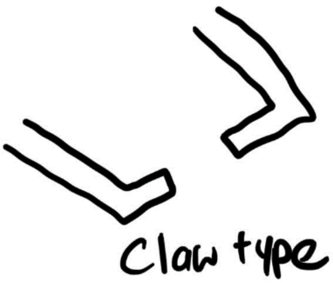
A morphological chart is a visual depiction of a product's required functionality that allows for the study of alternative techniques and combinations to achieve that functionality. It is also a method of idea generation in an analytical and systematic manner. For each product function element, there may be several practical solutions.

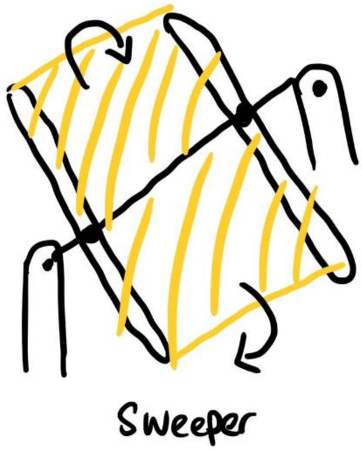
Table 2 – Morphological Chart

Sub-functions	Solution 1	Solution 2	Solution 3	Solution 4	Solution 5
Collector System	 See-saw method.	 Rotary action with rubber bands	 Claw type	 Cylinder with rubber band.	 Sweeper
Drive System	 Conventional 4-wheel drive	 double back wheel drive	 double front wheel drive	 Wheels nearer to center	 tri-wheel drive
Vehicle Chassis design	 rectangular chassis	 4-pillar design	 top side front ↑ to protect wheel	 Car-design	
Delivery System	 truck unloading mechanism	 simple ball stopper	 cylinder	 back-flip	 ejection system (spring loaded)
Edge Detection	 Front	 Front			
Software Program Design					

3.2.1 Collector System

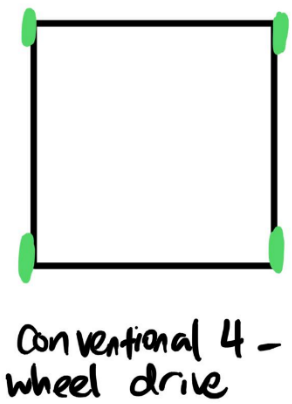
Table 3 - Collector system solutions

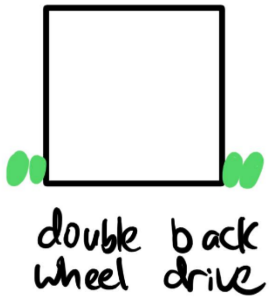
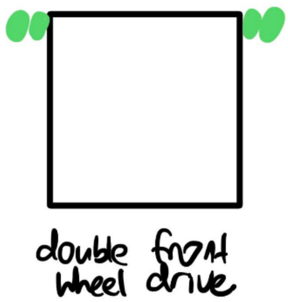
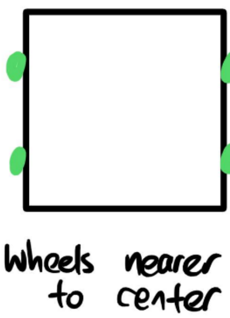
1.		<u>See-Saw Method</u> Pros: • Light weighted. • Simple-structured. Cons: • Higher centre of gravity. • Ball holding platform may not be stable. • Collection zone must be precise.
2.		<u>Rubber Band Roller Method</u> Pros: • High chance of successful collection. • Low design precision needed for collecting the ball. Cons: • Rubber band may snap during the collection process.
3.		Pros: • Can grip and land the ball precisely to specific location. Cons: • High precision for collection within the claw required. • Claw mechanism may become loose over time, leading to unsuccessful collection. • Slow collecting ability.

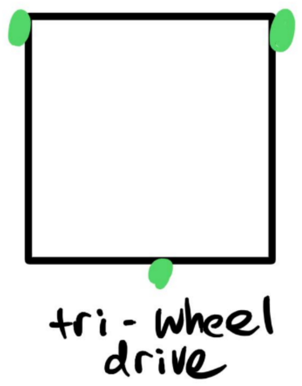
4.		Pros: • Simple-structured. Cons: • Requires robot to be precisely in line with ball. • Slight misalignments can cause ball to be stuck or pushed away.
5.		Pros: • Motor only required to rotate at a constant speed and direction. • Wide collection area (High collection success rate). • Vehicle can move while collecting. • Easy to design and fabricate. Cons: • Ball must be within a certain distance. • Ball may be pushed away instead of collected.

3.2.2 Drive System

Table 4 - Drive system solutions

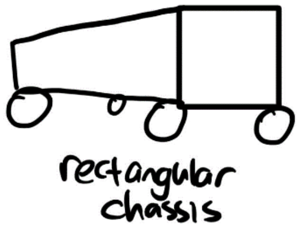
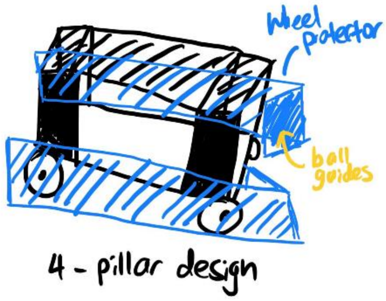
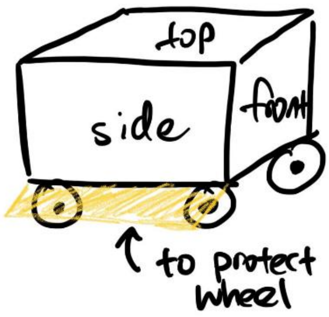
1.		Pros: • Stable base as stress is uniformly distributed among the 4 wheels. Cons: • Need to use more gears due to the distance between the wheels. • May result in turning difficulties.

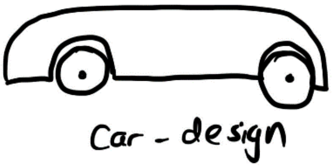
2.		Pros: • Able to overcome frictional forces easily. Cons: • Requires castor wheels at the front (not allowed). • May slip when turning.
3.		Pros: • Able to overcome frictional forces easily. Cons: • Requires castor wheels at the front (not allowed).
4.		Pros: • Most stable as stress will be uniformly distributed among the 4 wheels. • May be able to carry heavier weight due to wheels being nearer to the centre of weights. Lesser forces acting on the wheels due to a shorter distance from the load to the wheels. •

5.		Pros: • Less wheels used. • Lighter. Cons: • Stability issues. • Heavy components cannot be placed at back which will result in instability.

3.2.3 Vehicle Chassis Design

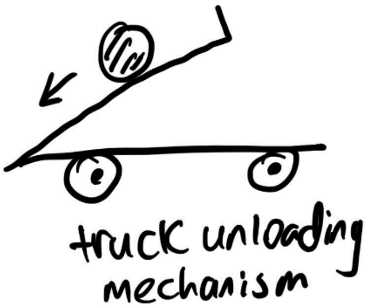
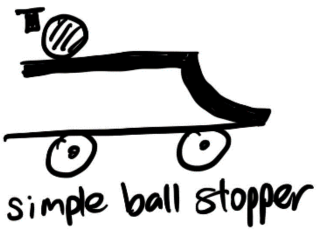
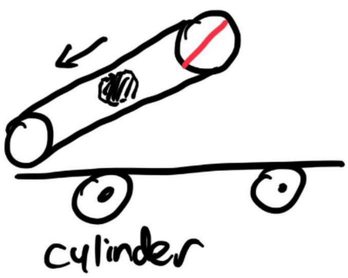
Table 5 - Vehicle Chassis Design Solutions

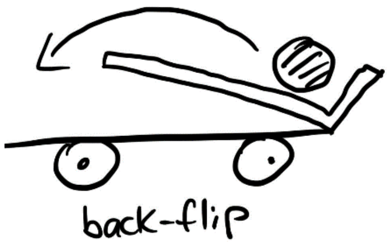
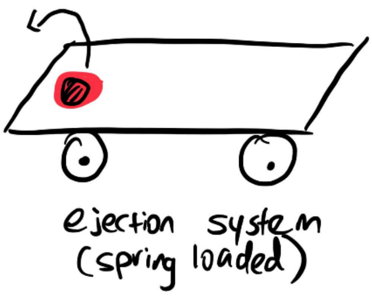
1.		Pros: • More surface area to lay internal components. Cons: • Not compact. • Wider turning angle required.
2.		Pros: • Wheels protected from external impact. • Able to guide tennis ball into the collection area. • Requires lesser material to construct. Cons: • Less compact.
3.		Pros: • Able to shield weak internal components from external impact. Cons:

		• May be too heavy, resulting in more power consumption to drive the system.
4.		Pros: • Aesthetic looking. Cons: • Limits the placements of the internal components due to its shape.

3.2.4 Delivery System

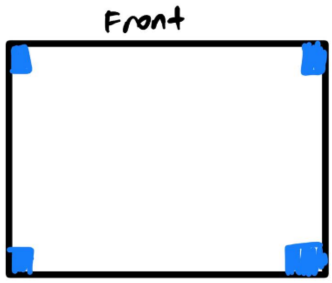
Table 6 - Delivery system solutions

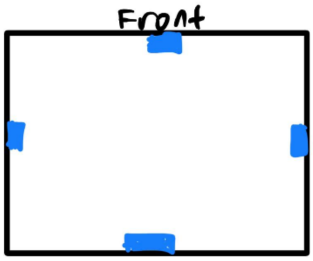
1.		Pros: • Ball can be delivered easily. Cons: • Slower process of unloading as the platform must be raised to a certain angle.
2.		Pros: • Less materials used. • Simple mechanism. Cons: • Requires rubber band to prevent ball from slipping off, which may need regular replacements.
3.		Pros: • Simple tilting will allow ball to be released. Cons: • Requires an extra mechanism to stop ball from rolling out the back end while vehicle in motion.

4.		Pros: • Simple to design and fabricate. Cons: • Motor may not be able to produce enough torque for the back-flip motion. • Poor delivering accuracy, ball may overshoot delivery area.
5.		Pros: • Do not require additional platform, hence saves space. Cons: • Requires a spring and extra electrical system for ejection. • Many calculations involved. • Complicated to design. • Poor delivering accuracy, ball may overshoot delivery area.

3.2.5 Edge Detection

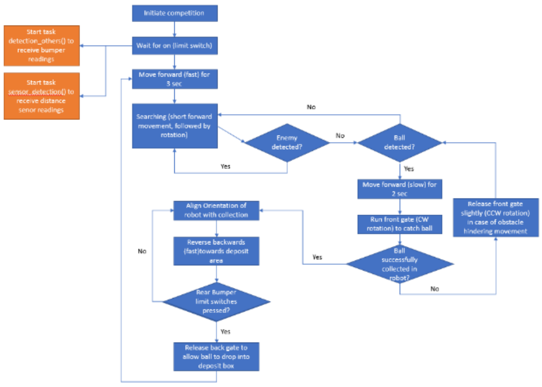
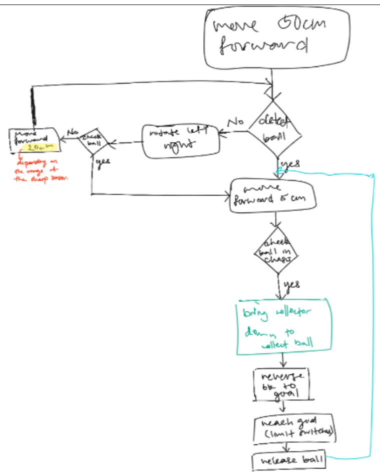
Table 7 - Edge detection solutions

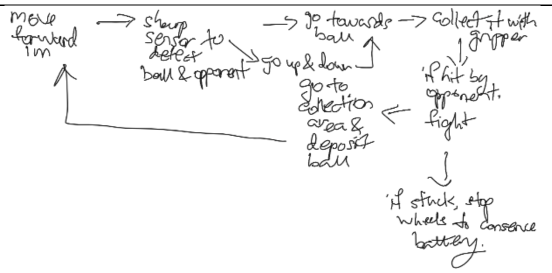
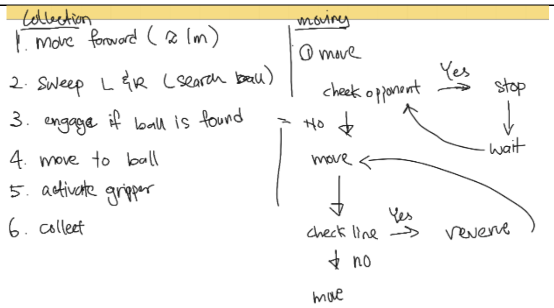
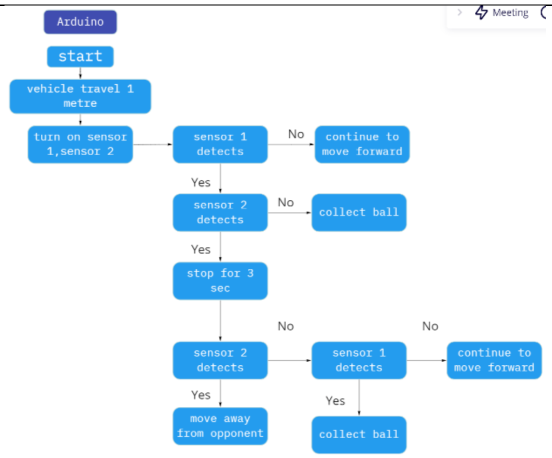
1.		Pros: • Can detect ball near the edge corners. • Edges of the robot will not overshoot the boundaries.

2.		Pros: • Simple design Cons: • Cannot detect ball near the edge corner. • Robot's edges may overshoot the boundaries during turning.

3.2.6 Software Program Design

Table 8 - Software Program Design Solutions

1.		Pros: • Complete logic that considers collector rotation to increase success of delivery. Cons: • Might take longer to find the right angle for delivery.
2.		Pros: • Fast and simple process Cons: • Unable to detect obstacles that may cause collision with other robots.

3.		Pros: • May cause obstruction to opponent's robot during competition, lowering their chance of a successful collection. Cons: • Incomplete logic • Takes very long to deliver ball as waste time on fighting the opponent.
4.		Pros: • Quicker time for robot to search and deliver the ball. Cons: • If the opponent's robot is stuck at the same place, the robot may not be able to remove itself from the obstruction.
5.		Pros: • Fast logical process. • Avoids opponent's robot, resulting in lesser chances of collisions. Cons: • If path is blocked by opponent's robot, the

		program may be trapped in a loop.

3.3 Decision Making

After a few combinations of designs were selected, the best solution had to be selected among them. A decision matrix is an appropriate structured method to evaluate multiple design concepts and select the best, taking multiple factors into account. In this concept evaluation method, evaluation criteria were to be developed, weights and values were to be assigned, and concepts were to be ranked based on their total weighted values.

In order to fulfil the design requirements and to ease the workload of the team members, the following 10 evaluation criteria were generated. Weights were also subsequently assigned to each criterion, adding up to a total of 100%.

Table 9 – Design Criteria

Criteria	Percentage (%)			
	5%	10%	15%	20%
1. Reliability of ball collection				
2. Ease of maintenance & troubleshooting				
3. Size of collection area				
4. Speed of the robot (collection time, travel speed)				
5. Ease of programming & assembling				
6. Consistency of motion				
7. Stability of the robot				
8. Appropriate space for sensor placement				
9. Power consumption				
10. Aesthetics				
TOTAL (%) :	**100%**			

"Reliability of ball collection" included the reliability of detecting the ball once it was collected, ensuring only 1 ball was collected at a time, without losing the ball during transportation. This criterion was assigned one of the highest weights as collecting 2 balls at a time would result in an immediate disqualification. Moreover, should the robot lose the ball upon collision with the opponent, a lot of time would be wasted to collect the ball again.

Another criterion that got the highest weight was “Ease of maintenance & Troubleshooting”. Building the final robot require many design iterations and edits. In each iteration, all components should be easily accessible and made simple, in order to change in order to ensure quick operation. Also, if any fault is detected from the robot on the day of the competition, any difficulty to troubleshoot the robot would result in high probability of a loss.

“Size of collection area” and “Speed of the robot” are supporting factors to satisfy the design specifications and the higher their value the better it was for the robot. “Ease of programming & assembling” helped the team members to build the robot in a simple and efficient manner. “Consistency of motion” includes the motors running at the designed speed and torque all the time and ensures the wheels are aligned correctly so that the motion of the robot matches with the corresponding motor output. “Stability of the robot” is a criterion to check how stable would the robot be when colliding with the opponent or when accelerating. In this criterion, the robot’s centre of mass should be low and the base area should be as high as possible. “Appropriate space for sensor placement” would help the robot to sense the parameters precisely and accurately. Even though abovementioned factors support the robot’s main functionalities, the robot can still operate without some of them. Therefore, each of these criteria is given 10% of weightage.

“Power consumption” and “Aesthetics” are selected as criteria as well, because the overall design quality is one of the assessments set by the course. But they are not compulsory components to affect the robot’s operation. In fact, the robot can be charged in between the matches and robot’s score in the competition is purely based on its performance. Hence, these two criteria are given the lowest percentages.

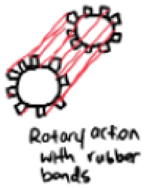
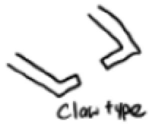
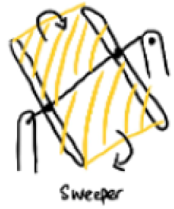
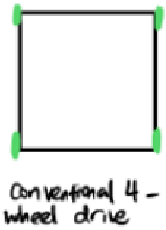
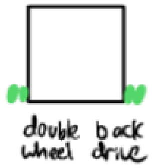
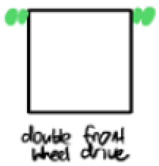
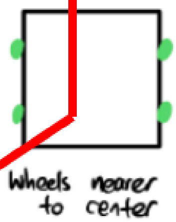
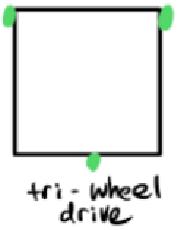
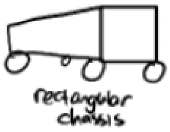
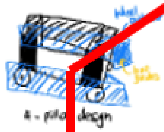
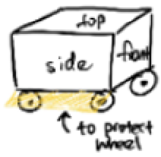
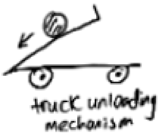
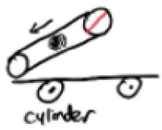
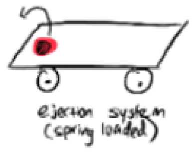
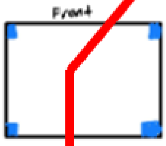
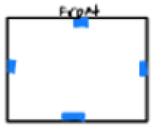
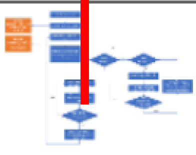
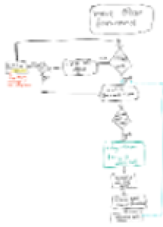
3.3.1 Decision Matrix

A decision matrix is a list of values in rows and columns that allows an analyst to systematically identify, analyse, and rate the performance of relationships between sets of values and information. Elements of a decision matrix show decisions based on certain decision criteria. The matrix is useful for looking at large masses of decision factors and assessing each factor's relative significance.

After thorough considerations, our team narrowed down to two designs:

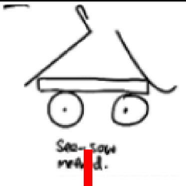
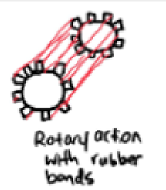
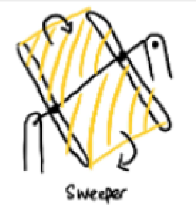
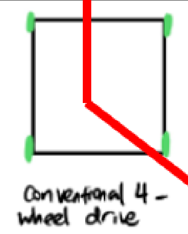
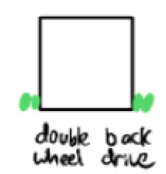
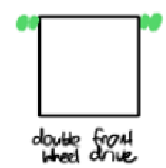
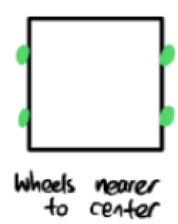
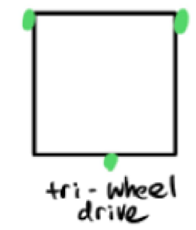
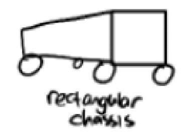
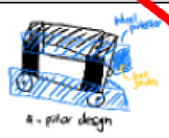
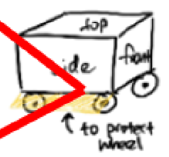
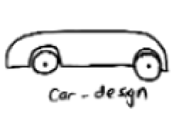
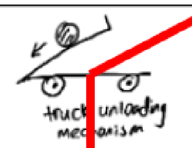
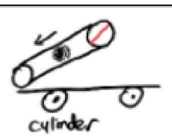
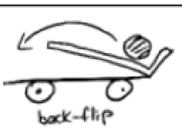
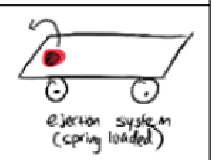
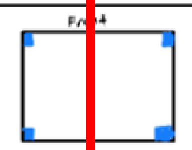
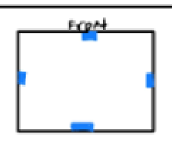
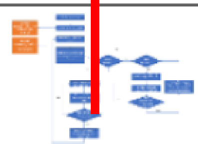
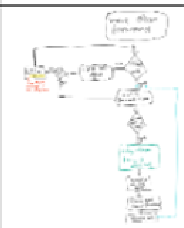
Design A:

Table 10 – Design A

Sub-functions	Solution 1	Solution 2	Solution 3	Solution 4	Solution 5
Collector System	 See-saw method.	 Rotary action with rubber bands	 Claw type	 Cylinder with rubber band.	 Sweeper
Drive System	 Conventional 4-wheel drive	 double back wheel drive	 double front wheel drive	 Wheels nearer to center	 tri-wheel drive
Vehicle Chassis design	 rectangular chassis	 4-pillar design	 top side front to protect wheel	 Car-design	
Delivery System	 truck unloading mechanism	 single ball stopper	 cylinder	 back-flip	 ejection system (spring loaded)
Edge Detection	 Front	 Front			
Software Program Design					

Design B:

Table 11 – Design B

Sub-functions	Solution 1	Solution 2	Solution 3	Solution 4	Solution 5
Collector System	 See-saw method.	 Rotary action with rubber bands	 Claw type	 Cylinder with rubber band.	 Sweeper
Drive System	 Conventional 4-wheel drive	 double back wheel drive	 double front wheel drive	 Wheels nearer to center	 tri-wheel drive
Vehicle Chassis design	 rectangular chassis	 4-pillar design	 top side front to protect wheel	 Car-design	
Delivery System	 truck unloading mechanism	 simple ball stopper	 cylinder	 back-flip	 ejection system (spring loaded)
Edge Detection	 Front	 Front			
Software Program Design					

Value is a rating value for the performance of a particular design against another design for a given criteria. In our case, we rated it from 1 to 5, with 1 being poor and 5 being ideal.

For individual weighted value (W.V.), it was obtained by multiplying the value with its weight for each criterion.

Table 12 – Decision Matrix

Criteria	Concept Variants			
	Design A		Design B	
	Value	W.V.	Value	W.V.
1. Reliability of ball collection (15%)	4	60	2	30
2. Ease of maintenance & troubleshooting (15%)	4	60	5	75
3. Size of collection area (10%)	3	30	2	20
4. Speed of the robot (collection time, travel speed) (10%)	3	30	2	20
5. Ease of programming & assembling (10%)	3	30	4	40
6. Consistency of motion (10%)	4	40	3	30
7. Stability of the robot (10%)	4	40	2	20
8. Appropriate space for sensor placement (10%)	4	40	2	20
9. Power consumption (5%)	4	20	5	25
10. Aesthetics (5%)	2	10	3	15
Total W.V.		**360**		**295**

Design A stood out as the superior design due to a few aspects. Firstly, with the 4-wheel closer to the centre of vehicle motion arrangement compared to design B, it could travel with stability and could conduct turns more reliably. This also allowed design A to take an advantage on the speed of delivery.

Secondly, design A featured a more stable collector mechanism. The see-saw design from design B might be unstable due to the nature of the asymmetrical components involved, hence the chance of collection might be lower.

Thirdly, the design chassis of design A requires lesser material to construct due to its exposed sides, thus reducing the weight of the robot. In design B, more material would have to be used to cover the sides, also it may make troubleshooting difficult.

Based on the overall weighted values obtained in the assessment matrix, Design A was deemed considerably superior to Design B and was thus chosen as the team's final design.

4 Embodiment Design

Embodiment design is one of the primary stages of the product design process. Its purpose is to allow users to create the best design arrangement, which will serve as a starting point for a more detailed design. According to design research by McGill University [1], the embodiment design is also known as preliminary design or system-level design. During the embodiment design stage, the product was developed further from the selected idea, targeting adopted product design specification (PDS) and cost constraints, so that a clear check of function, durability, production, assembly, operation, and costs could be performed. After carrying out the embodiment design, the detailed design will be next, leading directly to production.

To ensure a better mechanical assembly, embodiment design features three universal rules that applies to every design – clarity, simplicity, safety.

4.1 Clarity

Clarity is an important basic rule where the function of components must be unambiguous to minimise failure. In addition, it also ensures time and cost efficiencies. Each component was made sure to serve its purpose by reducing the complexity of rapidly identifying the failure during testing, hence reducing the amount of damage done to the robot.

The function of each robot's component is listed below as follows:

Table 13 – Functions of each components

1. Gear and wheel	The foundation of the vehicle's driving mechanism.
2. Mesh metal Frame	Used to protect the gears, wheels and wires from potential external damages caused by collisions. It is also bent at an angle with two bending edge pointed out from the front of the robot serving to extend the range of ball collection by guiding the ball into the collection zone.
3. Front Ramp	Used to roll the ball up to holding platform after retrieval.
4. Ball stopper (at Back Gate)	To hold the ball and release the ball into the delivery box.
5. Guided rails (red ones at platform)	Ensure the ball does not fall out from the back of the platform by narrowing the range of the platform.
6. Rubber band traps at the guided rails	Ensure the ball is trapped inside the delivery platform and will not roll down the ramp.
7. Metal collector	Rotates to sweep the ball up the front ramp to the delivery platform.
8. Delivery Platform	Holding area where the ball is kept after being swept. The ball waits to be delivered to and will be released at the delivery station.
9. Reflective sensors (at 4 edge corners of the robot)	To identify the yellow tape surrounding around the competition zone.
10. Distance sensor (lower)	To detect the ball.
11. Distance sensor (upper)	To detect opponent robot cars.
12. Limit switch (back)	To detect the wall at delivery area.
13. Limit switch (on ball holding platform)	Use to detect if the ball is on the delivery platform.

14. Digital Compass	Used to reorientate the robot and help in collecting and delivering the ball successfully.
15. Roof platform	This is where the VEX controller and battery are placed.
16. Motor (2 attached to each of the back wheels)	To transfer torque to drive the wheels.
17. Gear mechanism	To transmit torque and rotary motion from the back wheels to drive the front wheels.
18. Motor (Ball Stopper)	To ensure the metal stopper is rotated at an angle to release the ball when the limit sensor detects the wall at the ball delivery area.

4.2 Simplicity

Simplicity under embodiment design refers to designing to the minimum complexity level while still achieving its functions. The design should be easily understood and produced.

In this approach, the robot featured a simple design while ensuring that it could perform the basic functions such as boundary edge detection, opponent detection, ball collection and delivery to the delivery zone.

Furthermore, the components used to construct the robot were modular. The main skeleton of the robot was built using the rigid metal parts provided by the lab. The electrical components used in the robot were also standard. This enabled the team to set up a foundation structure of the robot design, leading to further innovations to achieve the main goal of the robot. The robot's program was also kept simple to ensure smooth troubleshooting.

4.3 Safety

Safety should be guaranteed by direct design, not by secondary methods (labels, guards, etc.). Safety measures can be separated into internal and external, where the internal allows for the highest safety. The external on the other hand, mainly warns or indicates danger. Safety considerations affect both the reliable fulfilments of technical function, not excluding the protection of humans and the environment.

4.3.1 Direct safety

Direct safety involves design approaches that prevent accidents from happening. Direct safety method achieve safety through systems or components actively involved in the performance particular task.

First, it was made sure that the robot will not topple over easily when hit by external impacts. It was ensured that the robot had a sturdy base by attaching four heavy wheels to the robot structure's base, powered by 2 motors with torque transmitted to the wheels.

In addition, it was made sure that the robot components were tightly attached to the structure. The electrical and metal piece components were secured, and wire entanglements were prevented by mainly using cable tie. Collars were used at both the ends of the wheels and the back gate flapper, to ensure they were kept in a fixed position without failing the system's mechanism. The metal sweeper structure was secured with both collar and blue tack. Soldering was also done for electrical connections.

4.3.2 Indirect safety

Indirect safety method involves the use of special protective systems & protective equipment.

Firstly, it was ensured that the robot was safe for the operator to use. The sharp metal edge of the metal sheets and mesh metals were covered with masking tapes. The wire connection areas were also secured with black tape to ensure no electric leakage.

Secondly, the robot's weak electric components were protected from external impacts. Metal mesh sheets wrapped around the base were used to protect exposed elements such as wires, compass, sensors, wheels, and limit switches.

Finally, it was ensured that the operator could pick up the robot with ease in the event of a technical fault. This was done by allocating enough space at the top of the robot's flat roof for the operator to hold.

4.4 Principles of embodiment design

Out of the 4 main principles of embodiment design, the following were the three key ideas that guided the embodiment design of this project:

1. Force Transmission
2. Division of Task
3. Stability or instability

4.4.1 Force Transmission

4.4.1.1 Minimise stress concentration

The metal mesh surrounding the robot features 2 fillet corners hence minimising stress concentration. This ensures that the protection mesh is less likely to fail and serves as a strong protection from external impacts.

4.4.1.2 Direct and shortest transmission path

If a force or moment is to be transmitted from one place to another with the minimum possible deformation, then the shortest and most direct force transmission path is the best.

The team picked the shortest force transmission line from the motor to the wheels. Two motors were used to power four wheels through six gears. The torque produced by the motor was sent directly to a gear and wheel through a shaft. The gears were then used to transmit torque and velocity to the other wheels using a 1:1 gear ratio.

4.4.2 Division of Tasks

A division of tasks made the system overall more efficient. It made the designing and building process faster and it was also easy to perform repairs or maintenance.

It was ensured that each component had a specific function and will not interfere with one another. For instance, the ball sweeper mainly performed the collection of the ball while the mesh metal frames were used to protect the internal electrical components of our robot.

4.4.3 Stability or Instability

A smooth curving slope was designed to ensure that the ball will be swept to the holding platform easily with momentum. Using the principle of stability, the team created a ball blockage system from a motor-powered metal ball stopper with rubber bands, to keep the ball at an equilibrium state at the delivery flat platform. The ball would only be deposited to the delivery area by lifting the ball stopper when the robot’s back limit switches were activated by contacting against the delivery area’s wall.

Furthermore, a low and wide heavy base was designed to improve the robot’s stability. During the competition, collisions occurred, creating large impacts on the robot. A broad base cushions the impact by providing a relative strong counteracting force due to the weight of the vehicle, hence the robot did not topple easily.

5 Detailed Design of Robot

The master design of a robot includes multiple sub-assemblies with their own purpose to ensure that the components are built with no redundant components. Below are the breakdowns of the subassemblies:

- Chassis
- Main Housing
- Collector
- Ramp

5.1 Chassis

Being the foundation of the robot, it must be strong and sturdy to withstand the movement of the motor and torsion twist during directional change. Thus, making the decision to choose which drive system was crucial as part of the chassis design. Looking the **figure below,** the decision was clear that the team chose to have a 4-wheel drive approach rather than 2-wheel drive, because having all 4 wheels engaged in rotation will drastically improve the traction, motion, controls, and overall drive performance. The chassis was also responsible to hold the entire weight of the robot including the main housing on the top of the chassis. A full metal base was preferred over 3D printed parts as it would provide flexibility in case of any minor changes required to be made to the design.

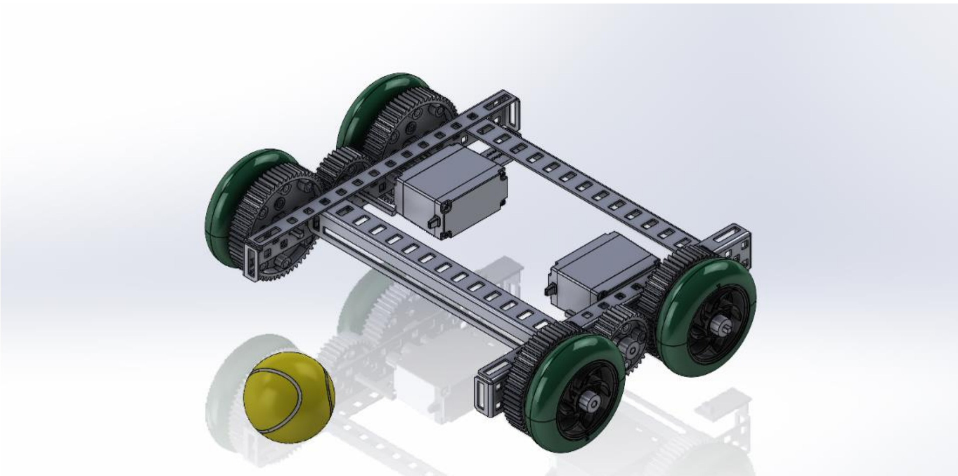


Figure 5-1 - Drive system and Chassis

5.2 Main Housing

All the components required a secured location to be stored and held down. The housing features a few aims to be fulfilled:

1. Creates a structural foundation for the components to mount
2. Provides a secured area to store the payload (tennis ball)

Within the housing, a metal ramp was fixed in a way that it clears the swinging motion with the correct radius. The housing was mounted directly onto the chassis so that it will further enhance the rigidity of the structure, resulting in better overall stability for the robot.

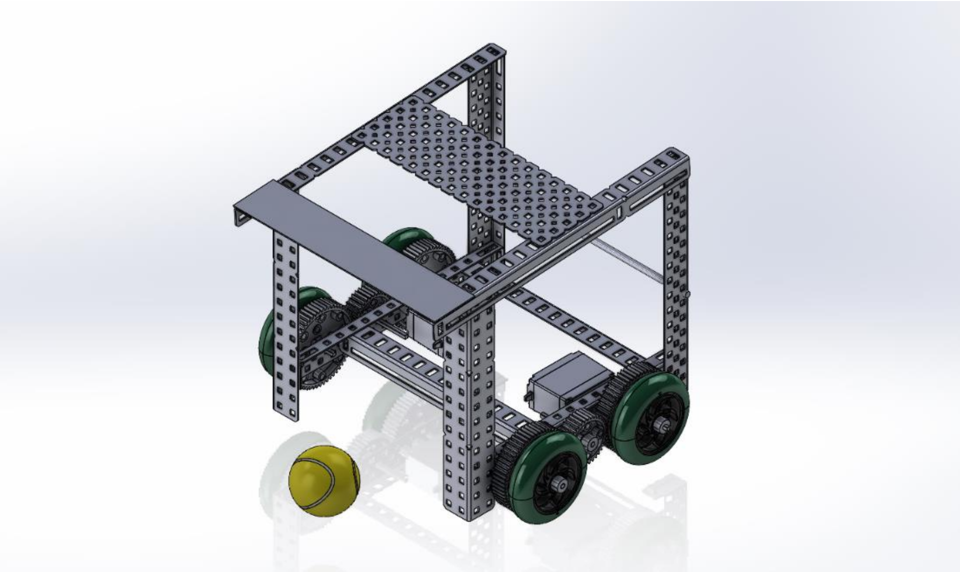


Figure 5-2 - Main structure for top attachments

5.3 Collector

Reiterating the aim of this robot - to collect the payload (tennis ball) and deliver it to the delivery point. Thus, to satisfy the above-mentioned aim, a mechanism was needed to be built to carry out the task of ball collection. Below shows a clear and concise view of the collector system.

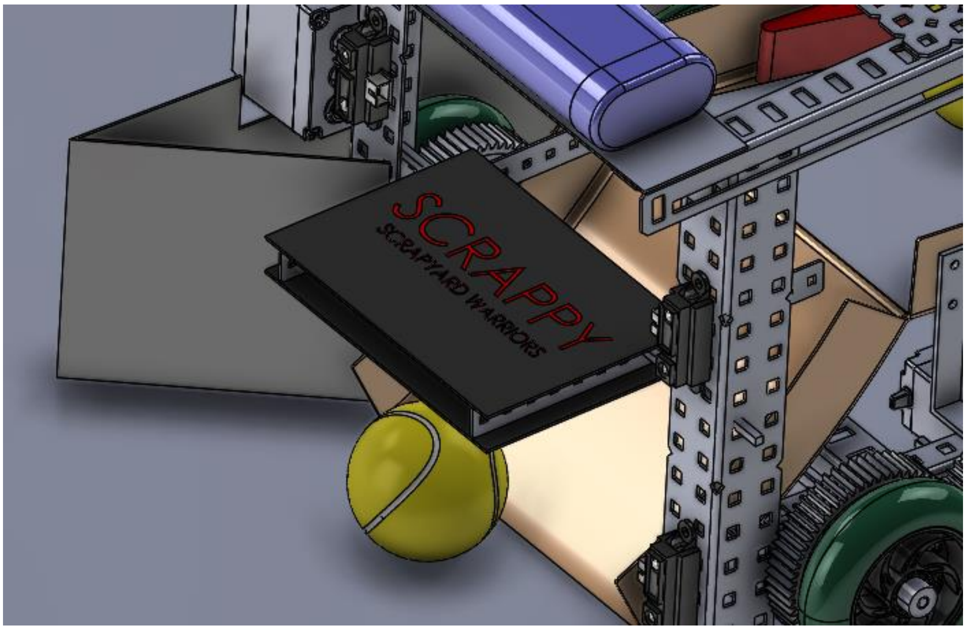


Figure 5-3 - Ball within the collection boundary

5.4 Ramp

After successfully collecting the ball, the ball must be protected and delivered to the delivery point easily. As such, the ramp was designed such that it could serve as a guide for the ball to reach the top of the ramp and to the holding area for the ball to await delivery. The holding area features a platform pressure sensor that detects if a ball was collected or not. To build this platform pressure, a limit switch was mounted on the bottom of the platform and covered with a light sheet of metal.

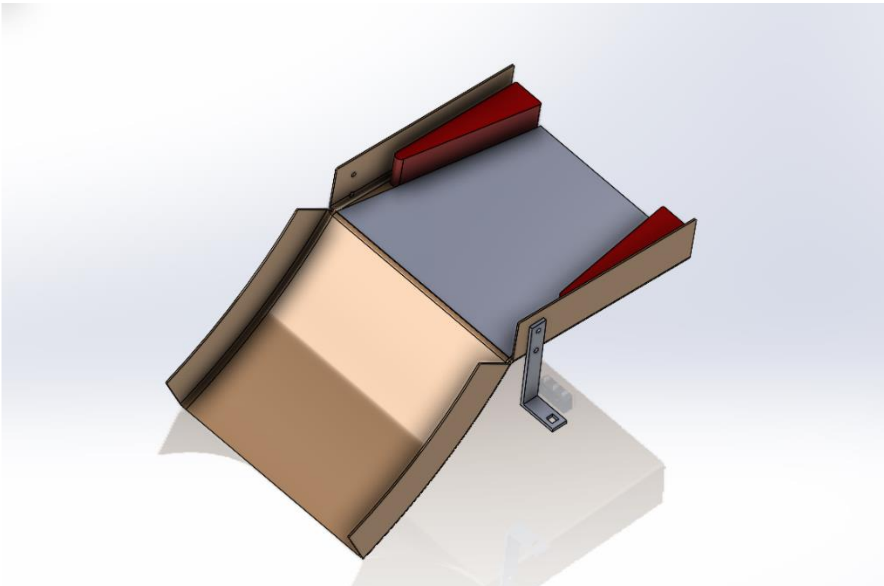


Figure 5-4 – Ramp with Built in Pressure Sensor

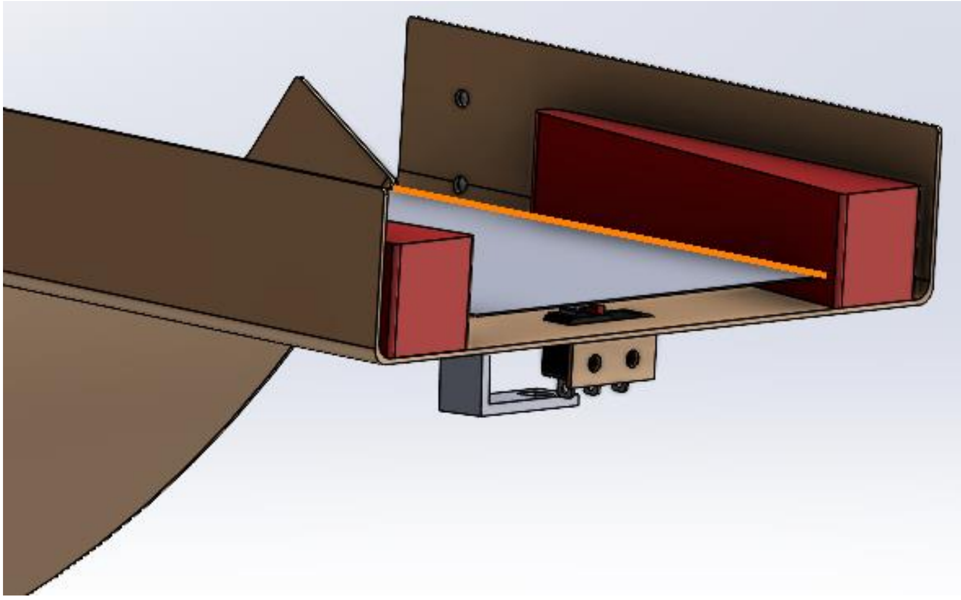


Figure 5-5 - Pressure sensor under holding platform

5.5 Expected ball projectile motion

During the design phase, considerations were made in diverse design concepts. The final design and conclusion from the team’s discussion included using a “sweeper” to collect the ball, and also a trap door to release it. The purpose was to increase the chances of collecting the ball with a wide sweeper. These can be depicted in the cross sectional views below.

Firstly, the ball would be guided into the collection area using the guides from the side bumpers.

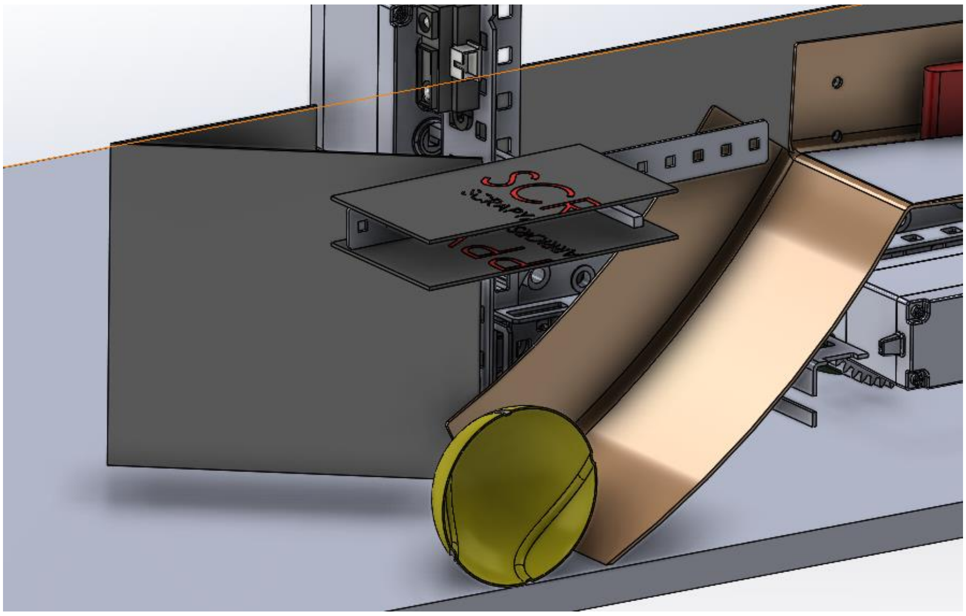


Figure 5-6 - Front gate with flapper in up position

Next, the collection mechanism would activate the sweeper to rotate and collect the ball:

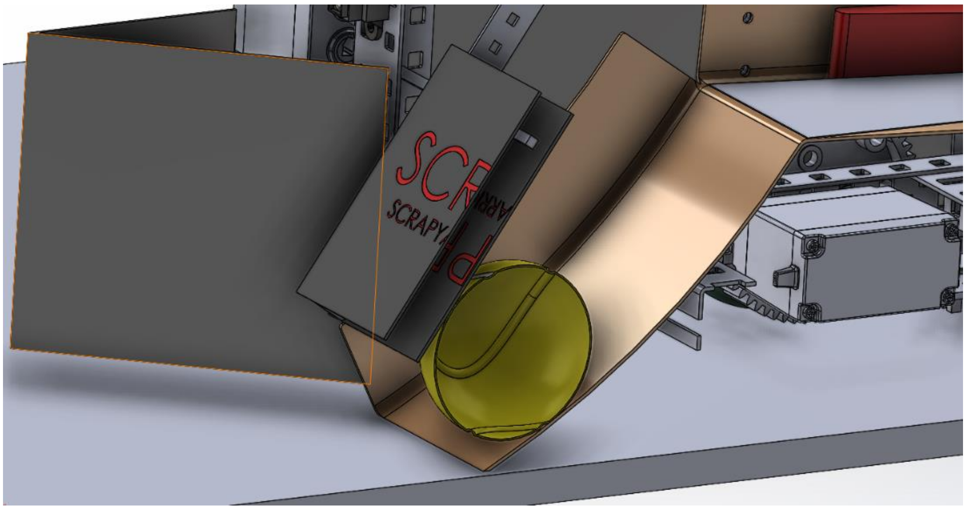


Figure 5-7 – Balls is successfully collected

The ball will finally climb onto the ramp with a successful catch, after which the door will act as a mechanism to prevent the ball from being released from the holding area, where it awaits to be delivered at the delivery point.

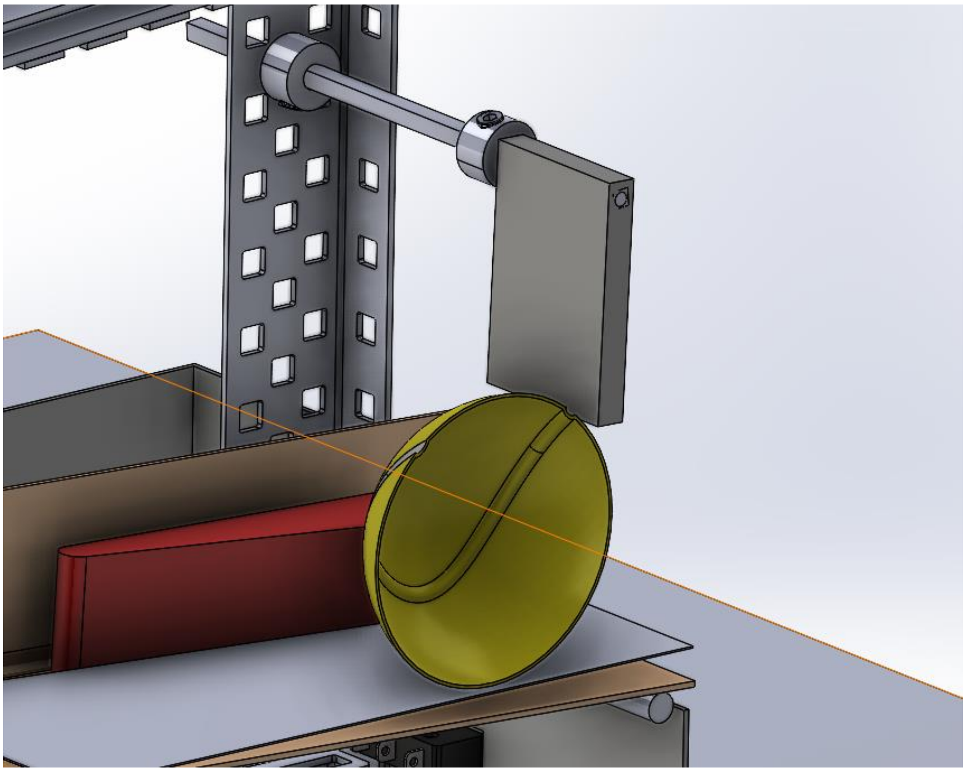


Figure 5-8 – Ball in Holding area

The platform which the ball rest upon, is equipped with a high-end pressure sensor to provide feedback into a VexController to allow the program to recognize that a ball was successfully collected.

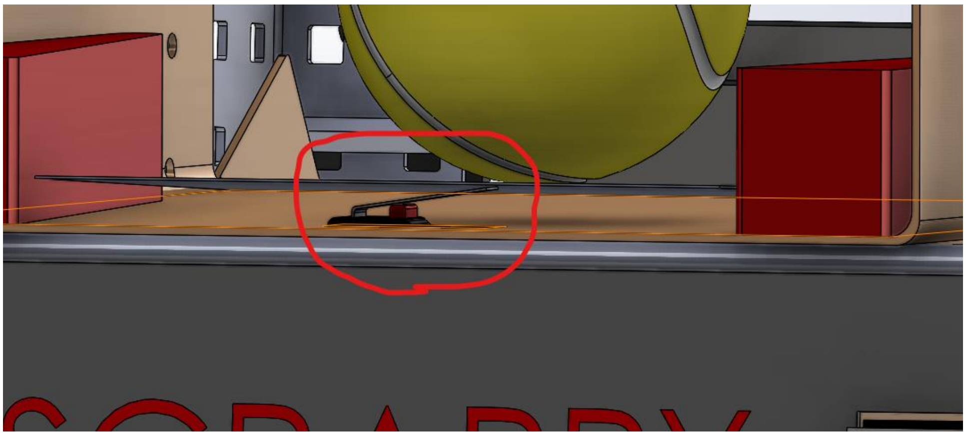


Figure 5-9 – Pressure sensor triggered when ball in holding area

The robot will continue to the delivery point to deliver the ball by reversing into the walls of the delivery area. In order for the program to proceed with the releasing of the ball, both of the rear bumper sensors would have to be depressed and activated together.

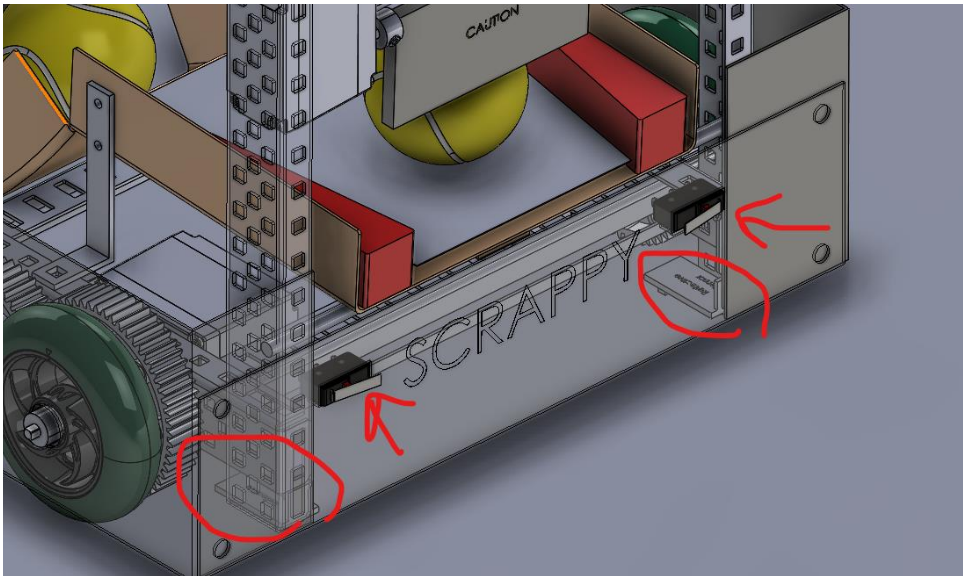


Figure 5-10 – 2 Limit switch sensor and 2 Bottom facing line sensor

6 Design Calculations

6.1 Power Transmission Calculation

The wheels and the collector are important mechanisms in the robot, powered by motors. This report discusses the dynamic and kinematic calculations of these mechanisms and will be shown in the subsequent sections of this report. The table below reflects the list of key components found in the robot and their corresponding mass.

Table 14 – Weight of Collector and Chassis

Components	**Mass (kg)**
Collector	0.122
Chassis	2.092
Total	2.214

Gear teeth for driven back wheels: 60

Intermittent Gear teeth: 36

Gear teeth for driven front wheels: 60

Gear ratio: 1:1

6.2 Wheel

The list of assumption while operating the wheels of the robot:

Table 15 – List of Assumptions when operating the Wheels

1.	Coefficient of static friction of the wheel μ_s is assumed to be 0.06 for rubber against a wooden ground
2.	Coefficient of dynamic friction of the wheel, μ_r= 0.02 (Assumed)
3.	Stall Torque of 2 Wire Motor 393, $T_{stall\ (motor)}$ = 1.68Nm (obtained from VEX reference sheet)
4.	Max continuous speed of 2 Wire Motor 393 (non-high-speed option), ω_m = 100rpm (10.47 rad/s) (Obtained from VEX reference sheet)
5.	The desired speed of the vehicle was 0.15 m/s

The motor allows for alteration of power levels from 0 to 127 or –127 to 0. Since the desired speed of the wheel was 0.15 m/s, it was found that the motor's power setting had to be tuned to a value of 60. The calculations involving the relationship between the power setting and the velocity of the wheel will be shown at chapter 6.2.1.3.

The calculations for continuous torque required to move the wheels at a constant velocity are as follows:

6.2.1 Minimum Continuous Torque Required (Wheel)

Normal force on each wheel

$$R_N = \frac{Weight}{No.\ of\ Wheels} = \frac{(2.214 \times 9.81)}{4} = 5.43N$$

Rolling frictional force to be overcome on each wheel,

$$F_f = \mu_d \times R_N = 0.02 \times 5.43 = 0.1086N$$

Radius of the wheel, r = 0.03m

Minimum Continuous Torque required to move back wheel,

$$T_{f(back\,wheel)} = F_f \times r = 0.1086 \times 0.03 = 0.00326Nm$$

Minimum Continuous Torque required to move front wheel,

$$\text{Gear Ratio} = \frac{T_{f(front\,wheel)}}{T_{f(back\,wheel)}}$$

$$T_{f(front\,wheel)} = T_{f(back\,wheel)} \times Gear\ Ratio = 0.00326 \times 1 = 0.00326Nm$$

6.2.2 Torque produced by motor at a power setting of 60

$$V_{motor} = 7.2\text{ V}$$

Current required to produce continuous torque:

$$I_{continuous} = 0.15\text{A}$$

Power drawn by motor at a power setting of 60

$$P_{motor} = V \times I \times Power\ Setting = 7.2 \times 0.15 \times \frac{61}{128} = 0.515W$$

Angular speed of the motor at power setting of 60

$$\omega = 10.47 \times \frac{61}{128} \text{ rad/s} = 4.99 \text{ rad/s}$$

Output torque of motor at power setting of 60

$$T_{motor} = \frac{P}{\omega} = \frac{0.515}{4.99} = 0.103Nm$$

Supplied Torque to front wheels.

$$T_{front\,wheel} = T_{motor} \times Gear\ Ratio \times Efficiency = 0.103 \times 1 \times 0.98 = 0.101Nm$$

The minimum torque required to drive each wheel was 0.00326Nm. Hence the motor was suitable and capable of delivery this amount of torque to these wheels.

6.2.3 Speed Output (Wheel)

The motor allows for alteration of power levels from 0 to 127 or –127 to 0. This can be beneficial in situations whereby moving at slower speeds may help the robot perform more consistently. Slowing down the speed of the motors can be achieved by adjusted the power level of the motor to a level lower than the full power.

The desired speed of the wheels was 0.15m/s, hence it was found that the motor had to be turned to a power setting of 60 to achieve this speed. The calculations include:

$$V_{motor} = 7.2 \text{ V}$$

$$I_{free} = 0.15\text{A}$$

$$P_{motor} = V \times I \times Power\ Setting = 7.2 \times 0.15 \times \frac{61}{128} = 0.515W$$

Resulting angular speed of the motor

$$\omega_{motor} = \frac{P}{Tmotor} = \frac{0.515}{0.103} = 5 \text{ rad/s}$$

Since the back wheels are located on the same motor's shaft,

$$\omega_{motor} = \omega_{back\,wheel} = 5 \text{ rad/s}$$

Since a Gear Ratio = 1 was defined previously, the speed of the front wheel is as follows:

$$\text{GR} = \frac{w_1}{w_2} = 1$$

$$w_2 = w_1 = 5 \text{ rad/s}$$

$$V = \omega r = 5 \times 0.03 = 0.15 \text{ m/s}$$

6.3 Collector – Design Calculations

6.3.1 Find Maximum Angle to Overcome Static Friction

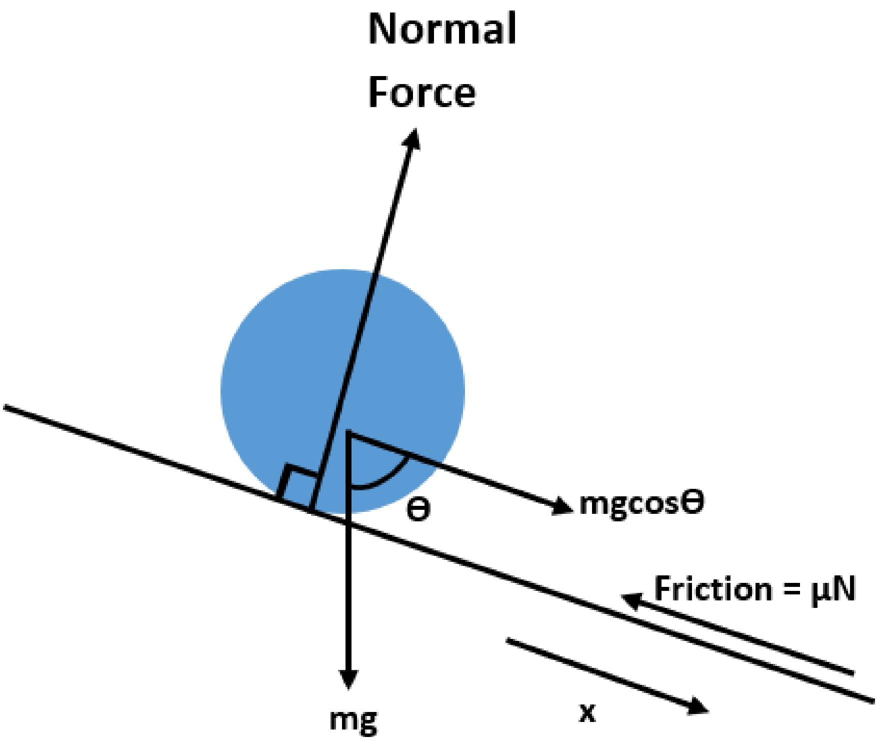


Figure 6-1 - Free Body Diagram of Ball on Back Ramp

To overcome static friction, the force component from $mg\cos\theta$ must be bigger than the frictional force.

$$mg\cos\theta > \mu N$$

From diagram, $N = mg\sin\theta$

Hence, $mg\cos\theta > mg\sin\theta$

$$\frac{mg\cos\theta}{mg\sin\theta} > \mu$$

$$\frac{1}{\tan\theta} > \mu$$

$$tan\theta < \frac{1}{\mu}$$

$$\theta < \tan^{-1}\left(\frac{1}{\mu}\right)$$

Assuming μ = 0.1,

$$\theta < \tan^{-1}\left(\frac{1}{0.1}\right)$$

$$\theta < 84.29°$$

Assuming μ = 0.05,

$$\theta < \tan^{-1}\left(\frac{1}{0.05}\right)$$

$$\theta < 87.13°$$

Hence, the angle of the back ramp must be lower than 84.29° in order overcome frictional forces to roll down the ramp. In this project, the angle of the back ramp was set to be 83°.

6.3.2 Find Length of Collector l

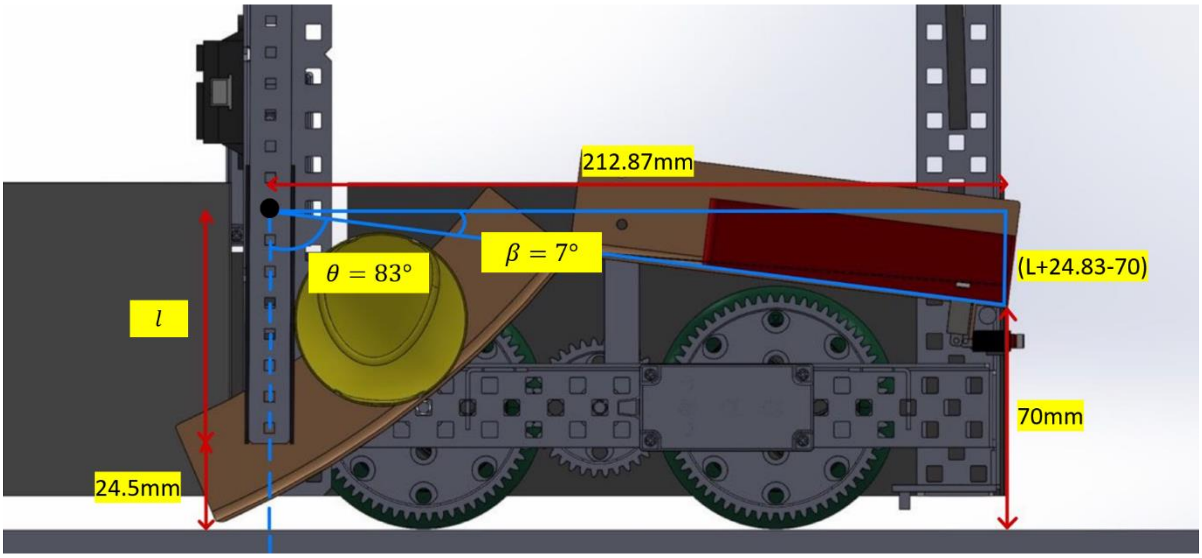


Figure 6-2 - Design Calculation of Length of Collector

With an angle of $\theta = 83°$, the minimum height at the end of the ramp also had to be at least 70mm due to the height of the delivery area. The total length of the ramp and storage area was designed to be a total length of 207.72mm. Based on the ball's diameter of 68mm, it was found that the optimal distance from the ground to the lowest point of the collector was less than half the diameter of the ball. In this case, this value was set to be 24.5mm.

$$\tan\beta = \frac{l + 24.5 - 70}{212.87}$$

$$l = 212.87\ \tan\beta + 45.5$$

$$l = 212.87\ \tan 7 + 45.5$$

$$l = \ 71.64\ mm$$

Therefore, collector arm length was taken to be 71.64 mm.

The list of assumption while operating the collector mechanism:

Table 16 – List of Assumptions when operating Collector

1.	Length of the collector $l = 71.64$ mm
2.	Mass of collector $m = 0.122\ kg$
3.	Stall Torque of 2 Wire Motor 393, $T_{stall\ (motor)}$ = 1.68Nm (obtained from VEX reference sheet)
4.	Max continuous speed of 2 Wire Motor 393 (non-high-speed option), ω_m = 100rpm (10.47 rad/s) (Obtained from VEX reference sheet)
5.	Desired angular speed of collector = 0.15 m/s

Since the desired speed of the collector was 0.15 m/s, it was found that the motor's power setting had to be tuned to a value of 60. The calculations involving the relationship between the power setting and the angular velocity of the collector will be shown at chapter 6.2.2.3.

The collection of the tennis ball involves a constant rotation of the collector, hence the calculations to define the continuous torque required for this operation is as follows:

6.3.3 Minimum Continuous Torque Required (Collector)

Minimum Torque Required to Collect Ball = Torque Required to Rotate Sweeper, hence:

$$T_{collector} = mgl = 0.122 \times 9.81 \times \tfrac{71.64}{1000} = 0.086\text{Nm}$$

$$T_{collector} = mgl = 0.122 \times 9.81 \times \frac{71.64}{1000} = 0.086Nm$$

6.3.4 Torque produced by motor at a power setting of 60

From the above calculations, Power drawn by motor at a power setting of 60:

$$P_{collector\,motor} = V \times I \times Power\ Setting = 7.2 \times 0.15 \times \frac{61}{128} = 0.515W$$

Angular speed of the motor at power setting of 60:

$$\omega = 10.47 \times \tfrac{61}{128}\ \text{rad/s} = 4.99\ \text{rad/s}$$

Output torque of motor at power setting of 60

$$T_{collector\ motor} = \frac{P}{\omega} = \frac{0.515}{4.99} = 0.103Nm$$

The minimum torque required by the collector was 0.086Nm. Hence the motor was suitable and capable of delivery this amount of torque to the collector.

6.3.5 Speed Output (Collector)

As mentioned, the relationship between the motor's power setting and the speed of the collector

From the above calculations,

Power drawn by motor at a power setting of 60

$$P_{collector\ motor} = V * I * \text{Power Setting} = 7.2 \times 0.15 \times \frac{61}{128} = 0.515\ W$$

$$P_{collector\ motor} = V \times I \times Power\ Setting = 7.2 \times 0.15 \times \frac{61}{128} = 0.515W$$

Resulting angular speed of the motor

$$\omega_{collector\ motor} = \frac{P}{T\ collector\ motor} = \frac{0.515}{0.103} = 5\ \text{rad/s}$$

Since the collector is located on the same motor's shaft, collector speed is given as follows:

$$\omega_{collector\ motor} = \omega_{collector} = 5\ \text{rad/s}$$

7 Control System

A good system must be able to react well according to the ever-changing conditions it will be subjected to. Hence, it is important to have a closed loop feedback system that can react accordingly whenever working conditions fall within their pre-set ranges. It is preferable as compared to an open loop feedback system that acts without any “intelligence” and executes its tasks without any regard to the changing conditions around the system. In a closed loop feedback system, every action taken by the robot will have a meaning behind it, and the act itself is indicative of the purpose that the robot is trying to fulfil. Overall, a closed feedback system will result in a better system even though calibration of the sensors and tuning of the trigger values in the programme requires more time to complete. The following sections of this chapter investigates the sensors, processing unit and wiring used in this robot system.

7.1 Non-Contact Sensors: Distance and Line Sensors

7.1.1 Sharp Analog Distance Sensor (long distance sensor)

Figure 7-1 – GP2Y0A21 Sharp distance sensor

The GP2Y0A21 Sharp distance sensor is a sensor that detects objects within a range of 10 cm to 80 cm. As shown in Figure 7-1, it uses a 3-pin JST PH cable that connects the 3-pin JST PH to the VEX microcontroller. 2 sharp sensors were used in this project, one for the detection of the opponent vehicle and the other for the detection of the ball. With this sensory data, the robot was able to avoid opponent vehicles while moving towards the ball. Both were mounted to the robot in a vertical arrangement.

Table 2 shows the specifications of the sharp analog distance sensor.

Table 17 – Specifications of Sharp Analog Distance Sensor (Long Distance)

Parameter	**Value**
Measurement Range	10 – 80 cm
Power Supply Voltage	4.5 to 5.5 V
Average Power Consumption	30 mA
Response Time	38 ms
Exit	Voltage-analog signal
Dimensions	29.5 x 13.0 x 13.5 mm
Mass	3.5 g

In the figure below, it shows the graph of the sensor's analog output voltage(V) vs the distance of the reflective object (cm). There is a minimum distance for every sensor and the minimum distance for this sensor is 10 cm with a voltage output of approximately 2.3V. The maximum distance is 80 cm with a voltage output of approximately 0.4V. The graph is an exponentially decreasing curve, therefore, the voltage output may decrease a lot despite a minor change in distance.

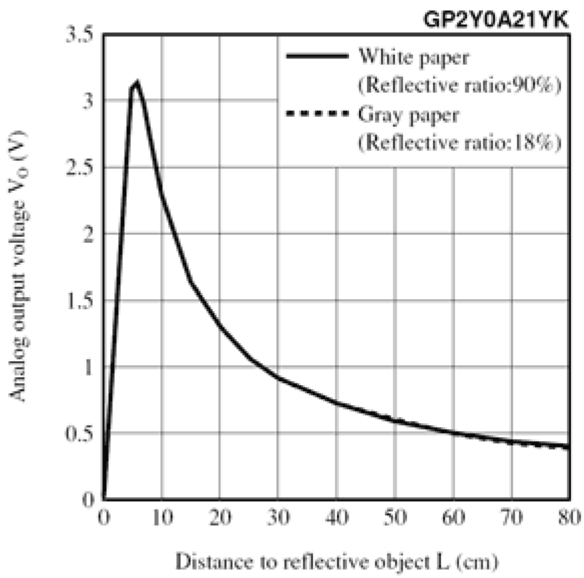


Figure 7-2 – Graph of Analog output voltage VS Distance to reflective object

7.1.2 IR Line Sensor

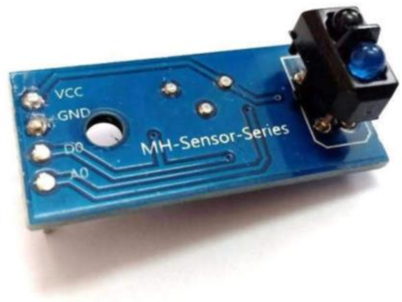


Figure 7-3 – IR Line Sensor

IR line sensor uses infrared reflection for line detection. It uses an infrared emitter and receiver, which means that the colour and texture of the surface affects detection rate. For example, a reflective surface such as yellow surface, reflects the infrared which the receiver then receives the reflected infrared then sends a signal to the VEX microcontroller.

In this project, 4 IR Line sensors were attached to the four corners of the robot, all facing downwards, to detect the yellow boundary of the arena.

This IR Line Sensor is based on the principle of detecting infrared (IR) reflection, using a pair of infrared emitter and receiver mounted together at the bottom. Due to this design, it is ideal for use in a mobile line detection robot. In infrared reflection, colour and surface texture are key factors that affect the detection rate by this sensor. For example, it was certain that when this sensor was placed in front of a yellow reflective tape, this sensor would pick up greater values with the higher levels of reflected infrared light than if placed in front of a black matte surface. Under different light conditions, the levels of reflected infrared light would also be different. Hence, to toggle it to the appropriate value for use, the onboard potentiometer was used to manually adjust the sensitivity.

In this project, 4 IR Line sensors were attached to the four corners of the robot, all facing downwards, to detect the yellow boundary of the arena.

Table 18 – Specifications of IR Line Sensor

Parameter	**Value**
Operating voltage	3.3V to 5.5V
Obstacle detection range	5mm to 10mm
Dimensions	3.2cm x 1.4cm

The sensor has 4 pins interfaces, which are analog output, digital output, ground, and supply voltage. The analog output pin was not utilised and only the digital output pin from this sensor module was used in this project.

7.2 Contact Sensor: Omron SS-5GL Limit Switch

Figure 7-4 - Omron SS-5GL Limit Switch

A limit switch is an electromechanical device that is activated by an object exerting physical force on it. It is used to determine whether an object is present.

Table 19 – Specifications of Omron SS-5GL Limit Switch

Parameters	**Values**
Current Rating	5 A
Voltage Rating AC	125 V
Operating Force	0.49 N
Release Force	0.04 N
Actuator Type	Hinge Lever
Contact Form	SPDT
Operating Temperature	-25 degrees to +85 degrees
Electrical Life	20,000 cycles
Mechanical Life	30,000,000 cycles

Like any other limit switches, the users have the option to choose whether they want the limit switch to operate as a normally closed or a normally open circuit. In this project, soldering was done to ensure that all were always in a normally open circuit.

In total, 3 limit switches were used in this project. 1 was used for detecting if the ball is collected and 2 were for detecting if the robot has reached the goal to trigger the release of the ball.

7.3 Directional Sensor: Digital Compass

Figure 7-5 – Digital Compass Navigation PC Board

The 1490 digital compass sensor from Robson Company was utilized on the PC Board. This sensor is a solid-state Hall effect device. It can detect the Earth's feeble magnetic field which allows it to display the direction when it rotates. Generally, the digital navigation PC board can be used for multiple varying navigation purposes. In this project, it was used as a simple digital compass for robot navigation.

The digital compass has a total of 8 lighting sequence:

- North (N)
- South (S)
- East (E)
- West (W)
- North-West (NW)
- North-East (NE)
- South-West (SW)
- South-East (SE)

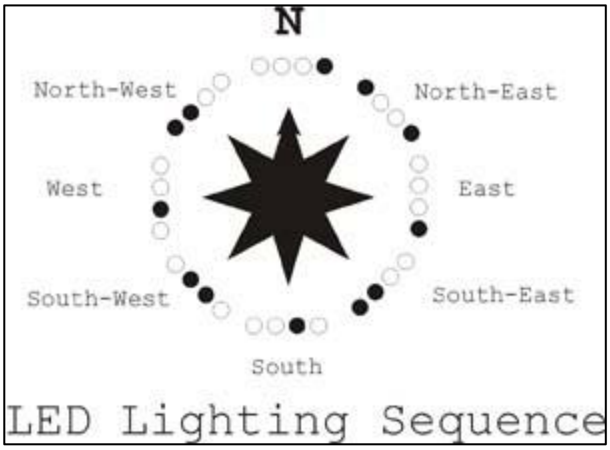


Figure 7-6 – Digital Compass Lighting Sequence.

Each direction has an allocated binary as well as a decimal number that is transmitted from the digital compass to the microcontroller, to indicate the front-facing direction. This is further illustrated within the table below.

Directions	**Binary**	**Decimal**
North	1110	14
South	1011	11
East	1101	13
West	0111	7
North – East	1100	12
North – West	0110	6
South – East	1001	9
South – West	0011	3

7.4 Processing Unit: VEX microcontroller

Figure 7-7 – VEX Microcontroller

The VEX ARM® Cortex®-based Microcontroller acts as the brain of the robot, coordinating the flow of instructions and power on the robot. All the other electrical components send signals to the microcontroller which will then coordinate the information given and execute tasks like the movements of the robot, delivering of the ball and opponent avoidance.

Table 20 – I/O Ports of VEX Microcontroller

I/O ports	Specifications
3 wire motor output (8 pins)	• For 3-wire VEX motors, Motor Controller 29s or servos • Type: Hobby standard PWM • Refresh: Every 18.5 ms
2 wire motor output (1 pins)	• For 2-wire VEX motors • Type: H-Bridge • Refresh: Every 1 mSec

I2C "Smart Sensor" Port (1 pins)	• For Future I2C Products • 0 to 3.3v logic I/O • Data rate, word width, etc set by compiler
UART Serial Ports (2 pins)	• Usage: VEX LCD Module • 0 to 3.3v logic I/O, 5V tolerant • Data rate, word width, etc set by compiler • Factory default: 19.2k baud, 8 data bits, no parity, talks with VEX LCD
12-bit Analog Inputs (8 pins)	• 12-bit resolution. • 10 µSec access time.
Fast Digital I/O (12 pins)	• 150 kHz input frequency • Can Be used as interrupt
DAC Speaker Output (1 pin)	• Usage: Output sound, voice, and music to an external speaker
Rx1 & Rx2 (2 pins)	• Connects to 75 MHz receivers • Pin 1 - Power +5 volts (the right most pin when looking at the front connector on the VEX Controller) • Pin 2 - Receiver • Pin 3 - Gnd • Pin 4 - Tether Detect (the left most pin when looking at the front connector on the VEX Controller)

The VEX Microcontroller uses a STMicroelectronics ARM® Cortex® M3 user processor which have specifications as follows:

1. Speed – 90 million instructions per seconds
2. RAM – 64KB
3. Flash – 384KB program space

7.5 Output System: 3 – Wire Servo Motor

Figure 7-8 – 3 Wire Servo Motor

The 3 wire servo motor was used for the ball collection as well as the ball delivery gate.

Parameters	Values
Free Speed	100 rpm at 7.5V
Rotation	100 degrees
Stall Torque	6.5 in-lbs.
Voltage	4.4 – 9.1 V
PWM Input	1ms – 2ms (full reverse to full forward) 1.5ms (Neutral state)
Current Draw	20mA to 1.5A per Servo
Max Power	4.9 W
Voltage	6V

7.6 Wiring Diagram

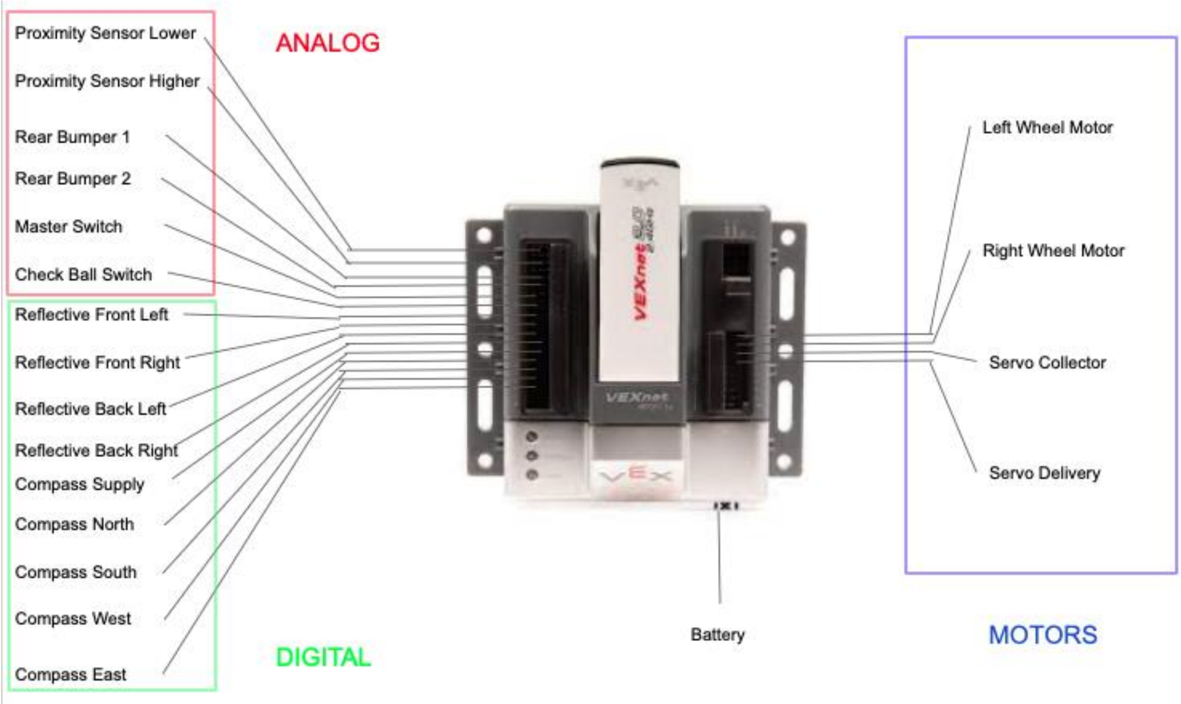


Figure 7-9 – Wiring Diagram of the Robot

8 Programming logic

8.1 Software Architecture

The figure below shows the software flowchart that the robot will undergo during the competition. The boxes in orange indicate tasks, in which the robot will continuously run throughout the entirety of the competition in separate threads that do not interfere with the main thread. This allows the robot to simultaneously run multiple algorithms in parallel.

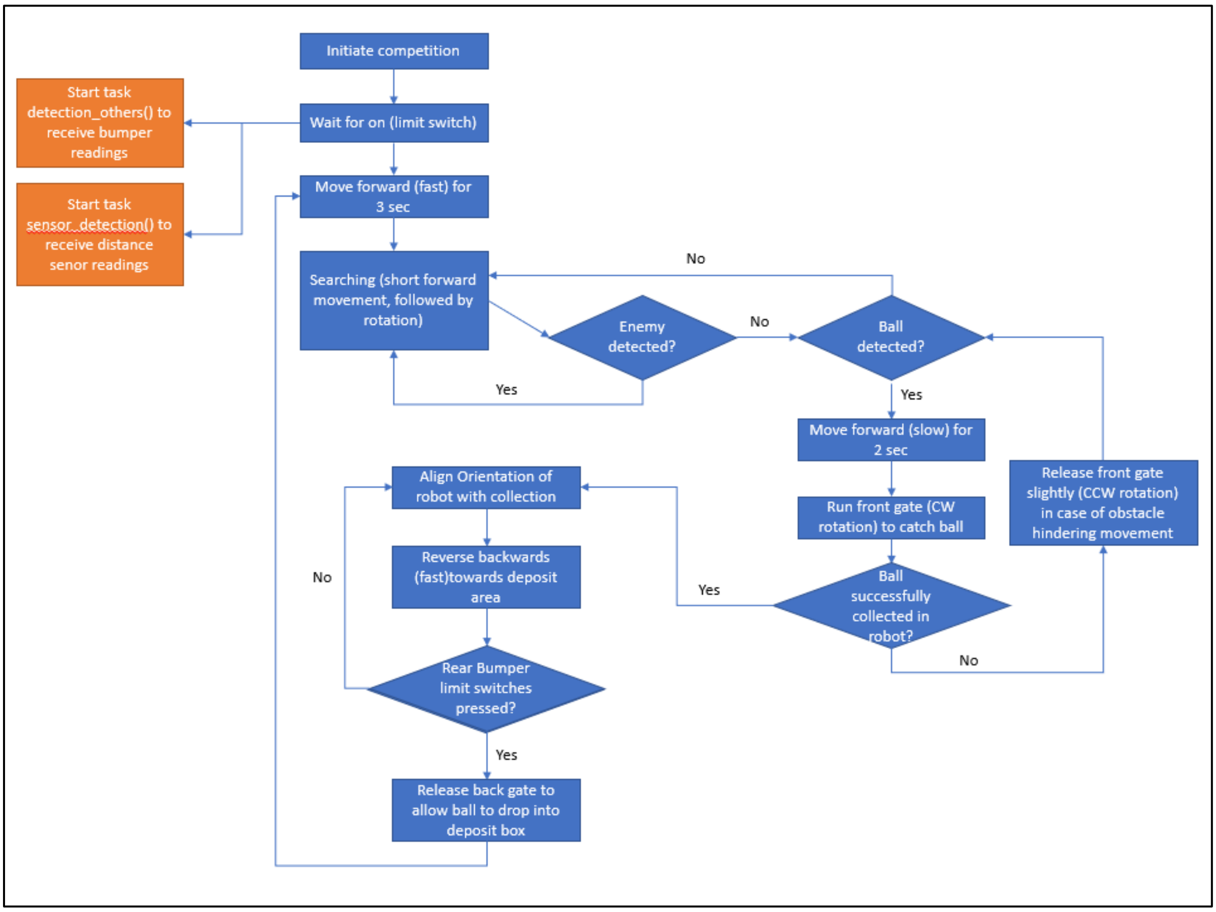


Figure 8-1 – Program Flow

When the robot is switched on, all sensors will start taking in values. However, the software sequence will only be initiated when the master limit switched is pressed.

To prevent any conflict of which algorithm will run first, the tasks were only written to update global variables in the program without actuating any physical movements on the robot itself. Physical movements would be performed by functions within the main program to ensure a smooth continuous flow in the robot's movements. The code will initiate two tasks (orange) to run concurrently with the main program:

1. <u>Bumper/Orientation Detection:</u> The task detection_others() continuously checks the orientation of the robot as well as whether the rear bumper was being depressed at any point in time. This was important in situations where the robot might reverse into an opponent's robot during the delivery phase. Used concurrently with the rear line sensors, the bumper detection was able to ensure that the ball will be released **only** at the delivery area.
2. <u>Sensor Detection:</u> The task sensor_detection() continuously checks the two front long-distance sensors to detect whether our robot is either facing an opponent or a ball. In brief, if the bottom long-distance sensor detects an obstacle while the top distance sensor does not, it can be concluded that the object detected is a ball. However, if both were detected, then it results in the only possibility is that the object is a robot. To ensure that this algorithm works properly, the height of the bottom sensor had to be precisely positioned to prevent false positives (eg: detecting boundary as ball).

The initial planned flow for the program is as listed below:

8.1.1 Detection

1) Limit switch on
2) Check compass reading and store in variable first_dir (integer variable temp)
3) Move forward for time t1 (1m)
4) Check in straight direction: if obstacle in front of left sensor, then go straight, align with short range sensor and go to collect stage; Else if in front of right sensor, align with left sensor, then same logic as above.
5) Rotate 90 degrees left and 180 degrees right - while running task obstacle_detection (robot or ball)
   a. If ball, stop rotate and move forward:
   b. If in rotate_left_stage, use left sensor array to detect ball and move forward until in range of long-range sensor (+-5) then turn theta degrees (map theta to distance) to allow ball to come in front of short-range sensor then use that to get to best distance with ball to collect (depends on dimensions)
   c. If in rotate_right_stage, use right sensor first to detect obstacle, turn in place to get obstacle in line with left sensor array, detect ball or robot, if ball use same logic as rotate left stage to get to ball collection stage
   d. Else if robot: continue sweep

e. Else nothing: move forward 0.1m, reset position first, then do 270 sweep CW and run algorithm 5a, setting rotate_right_stage as true.
    a. If still nothing, reset position then run program from [4] again.

8.1.2 Collection

6) Rotate scoop motor for time t2 and wait for t3 milliseconds
    a. If limit switch returns true within t3 , then continue with flow and set codeState=STATE_GO_TO_COLLECTION
    b. Else reset scoop motor for time t2, check ball with short range sensor:
    c. If detected, then move to collection stage
    d. Else run program from [4].

8.1.3 Reversing

7) Check compass direction dir, if dir!=first_dir, then use func rotate_initial to straighten robot:
    a. In rotate_initial, based on temp value rotate until current temp = initial temp
8) Move backward until 4 back limit switches pressed and yellow tape sensor(BL and BR) detected.
9) Rotate back gate for time t4 to release ball into deposit box, reset back gate and run program from [3].

8.2 Code Explanation

```
/*--------------------------------global variables----------------------------------*/
bool ballClose=false;
bool robotDetected=false;
bool ballCollected=false;
bool rearBumperPressed=false;
int global_orientation;
int codeState=0;
bool already_in_collection_place=false;

/*--------------------------------function definitions------------------------------*/

void align_orientation_with_collection();
void start_move();
bool is_limit_ball();
void read_orientation();
void servo_to_angle(int port, int direction, int time);
bool catch_ball();
void move(int direction, int speedMode);
void rotate(int direction, int speedMode);
void reset_servo();
bool go_to_collection_place();
bool move_to_ball();
bool search_ball();
void wait_for_on();
bool line_detection();

/*---------------------------------task definitions----------------------------------*/

task competition();
task sensor_detection();
task detection_others();
```

Figure 8-2 – Functions and main variables used

8.2.1 Function Description

Table 21 – Function Descriptions

Function/Task Name	**Description**
void align_orientation_with_collection();	Aligns robot with collection point, rotating the robot until it faces south based on compass readings
void start_move();	Moves the robot forward for 3 seconds at high speed. This was done to get the robot to the middle of playing field as fast as possible before searching starts.
void read_orientation();	Reads the robots current orientation

void servo_to_angle(int port, int direction, int time);	Controls the motors that are responsible for collection and deposit of ball from the robot.
bool catch_ball();	Runs the front collection motor to collect the ball into the robot
void move(int direction, int speedMode);	Allows precise control of speed and direction during forward/reverse movement of robot.
void rotate(int direction, int speedMode);	Allows precise control of speed and direction during Clockwise (CW)/Counter Clockwise(CCW) rotation of robot.
void reset_servo();	Resets the collection and depositing motors after successfully scoring a ball.
bool go_to_collection_place();	Robot will continually reverse until it reaches the collection place with rear bumper pressed and yellow line detected on rear line sensors.
bool search_ball();	Runs search algorithm until ball is detected. Search algorithm includes rotating 90 degrees CCW, 180 degrees CW and then 90 degrees CCW again to face its initial direction.
bool move_to_ball();	After ball is detected, robot will move slowly towards the ball until it is within range of the catcher motors.
void wait_for_on();	After robot is powered on, this function allows starting the whole software algorithm with the flick of the master limit switch
bool line_detection();	Detects the yellow tape at the boundary of the playing field. This

	function is called in a while loop whenever the robot is running a move/rotate function to prevent overshooting the boundary.
task competition();	Main task that starts other sub tasks and main software algorithm for searching, collection, aligning and deposit of ball.
task sensor_detection();	Continually receives readings from the top and bottom distance sensors to determine if robot or ball detected.
task detection_others();	Continually receives compass readings and rear bumper readings to determine if robot has reached deposit area or hit another robot.

8.2.2 Function Details

Several specific functions which were deemed important to the robot's overall movements will be discussed here in detail to display how specific strategies were implemented.

8.2.2.1 void align_orientation_with_collection()

```
void align_orientation_with_collection()
{
    //rotate to align the orientation with the collection place
    if(global_orientation==0||global_orientation==1||global_orientation==2||global_orientation==3)
    {
        while(global_orientation!=4)
        {
            rotate(-1,2);//CW
        }
    }
    else if(global_orientation==5||global_orientation==6||global_orientation==7||global_orientation==4)
    {
        while(global_orientation!=4)
        {
            rotate(1,2);//CCW
        }
    }
    move(1,0);
    rotate(1,1); //rotate CCW to orientate properly to deposit box
    wait1Msec(500); //for 500Msec
    move(1,0);
}
```

Figure 8-3 – Function to align orientation with delivery area

The global variable, global_orientation, was defined in the function read_orientation(). The robot was first aligned to face South. Since the precision of the compass was not high enough for accurate reversal, and the board also did not correspond to any of the compass's pre-set 8 directions, further rotations were required to align the robot with the deposit area. A CCW rotate function is implemented after the compass alignment, as shown by lines 553 to 556 in Figure 83. A further adjustment, by calling the rotate() function, was required to align the robot perfectly to the deposit box during reversal. This is illustrated in Figure 8-4.

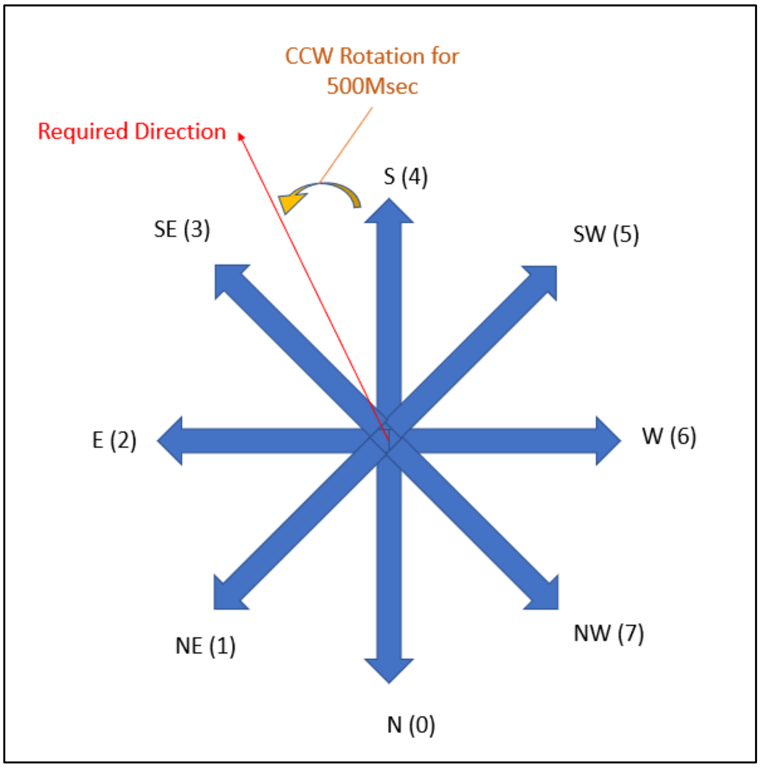


Figure 8-4 – Numeric coding of different directions with the digital compass

8.2.2.2 task sensor_detection();

```
361 task sensor_detection()
362 {
363     while(true)
364     {
365         if(600<SensorValue[B] && SensorValue[B]< 3000)          //lower sensor detects something
366         {
367             if (SensorValue[C] >600)                            //upper sensor detects something
368             {
369                 ballDetected=true;
370                 robotDetected=true;
371             }
372             else                                                //upper sensor does not detect anything
373             {
374                 ballDetected=true;
375                 robotDetected = false;
376             }
377         }
378         else
379         {
380             ballDetected=false;
381             robotDetected=false;
382         }
383         //If ball is detected, Check if the ball is close enough
384         if(ballDetected)
385         {
386             if(SensorValue[B] >900)                             //side sensor detects within the range acceptable for front sweep
387             {
388                 ballClose =true;                                //ball is close enough for the front sweeper
389             }
390             else
391             {
392                 ballClose = false;                              //not close enough
393             }
394         }
395     }//end of while
396 }//end of task
```

Figure 8-5 – Task running through out to detect ball or robot in front of proximity sensor

This task was vital to the workability of the robot. It acts as the eyes of the robot in detecting whether the obstacle in front of it is either a ball or an opponent. A forever while loop is called within this task to ensure the task runs for the entirety of the competition. SensorValue[B] corresponds to the lower sensor, B, while SensorValue[C] corresponds to the upper sensor, C. Referring to Figure 8-5, the values 600 and 3000 in line 365 was determined through experimentation of what works best for accurate and robust results. If the sensor B detects an object while the top sensor does not, the object is a ball. If both sensors detect an object, then the object is an obstacle. This could only be implemented if the lower sensor's position was perfectly aligned to be just above the boundary of the play area to prevent detecting the boundary as a ball. ballClose is another global variable that was continuously updated as a prerequisite to run catch_ball algorithm to sweep the ball up.

8.2.2.3 bool search_ball()

```
bool search_ball()
{
    codeState=STATE_SEARCH_BALL; //indicate the code is now at search_ball function

    while(true)
    {
        debugVar=1;
        clearTimer(T1);
        while(time1[T1]<2*600)  //set timer for 1200Msec
        {
            line_detection();
            if(ballDetected && !robotDetected)
            {
                move(1,0);              //stop
                return true;
            }
            rotate(1,2);                //rotate CCW
        }

        clearTimer(T1);
        while(time1[T1]<4*600)  //change the value 2400 TODO
        {
            line_detection();
            if(ballDetected && !robotDetected)
            {
                move(1,0);              //stop
                return true;
            }
            rotate(-1,2);               //rotate CW
        }
        debugVar=2;

        clearTimer(T1);
        while(time1[T1]<2*600)  //set timer for 1200Msec
        {
            line_detection();
            if(ballDetected && !robotDetected)
            {
                move(1,0);              //stop
                return true;
            }
            rotate(1,2);                //rotate CCW
        }
        move(1,0);

        //if ball is not detected during rotation
        clearTimer(T1);
        while(time1[T1]<1000)   //set timer for 1800Msec
        {
            move(1,2);
            line_detection();
        }
        return false;
    }
```

Figure 8-6 – Function which implements the searching movement of the robot

Figure 8-6 represents the main algorithm that the robot undergoes prior to ball collection. Visually, this can be represented by the image shown below in Figure 8-7:

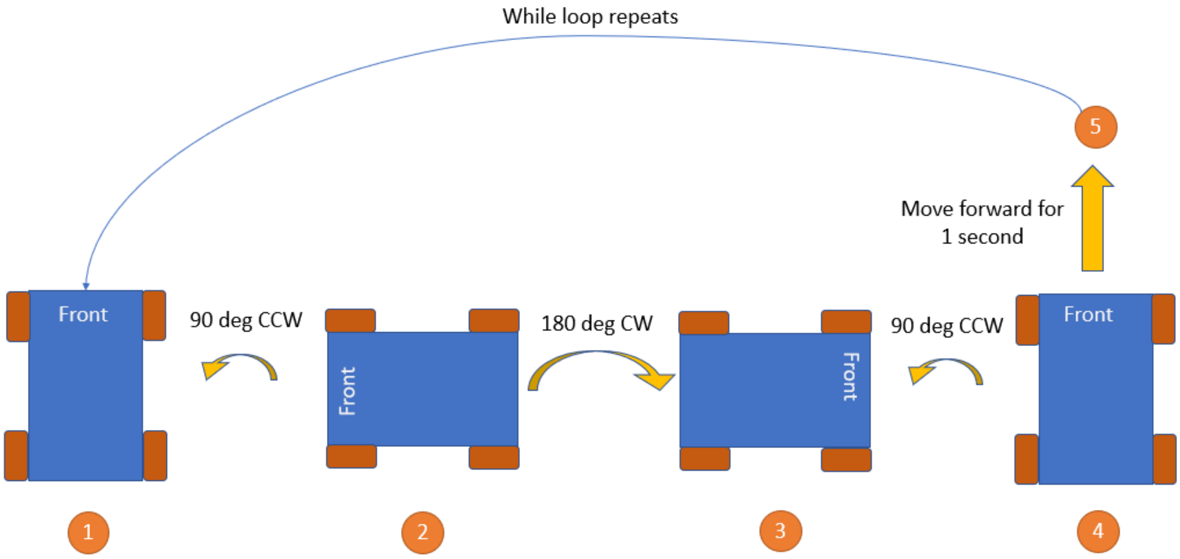


Figure 8-7 – Searching process illustrated

The robot rotates on the spot using a 90-degree CCW rotation, followed by a 180-degree CW rotation, and then a 90-degree CCW rotation to face forward again. While this rotation is happening, the robot will be continuously searching for the ball as seen in the code where the conditional statement include “if(ballDetected && !robotDetected)”, which corresponds to the global variables discussed in task sensor_detection() above. If no ball is detected, the robot will continue with its forward movement for 1 second, and then the loop will repeat itself. If a ball is detected, the function returns true and exits the loop, moving on to the move_to_ball() function for eventual collection.

8.3 Issues Faced and Iterative Improvements

1. Short distance sensor initially used to detect ballClose was not reliable
    - Removed the short range sensor and coded algorithm based on that
2. Orientation did not correspond with board
    - Added a rotational element after using the compass to get the correct orientation
3. Rotation algorithm did not have line_detection function called initially, causing robot to go over boundary during search.
    - Added a line_detection algorithm and search worked without exceeding boundary.

4. Ball was not detected in previous algorithm due to initial plan of using short range sensor to detect the ball.
    - ballClose is a variable that is updated as the robot moves towards the ball, but the location of the ball is not changed as robot moves towards it.

```
        if(ballDetected)
        {
            distanceB = (10000.0/SensorValue[B]-0.6685)/0.4255;

            if(SensorValue[A] > 910 && SensorValue[A] <1500)        //side sensor detects within the range acceptable for grabber (Sensor A is short range se
            {
                ballClose =true;    //ball is close enough for the grabber
            }
            else
            {
                ballClose = false; //not close enough
            }
        }
    }//end of while
}//end of task
```

Figure 8-8 – The ballClose condition before testing

    - However, in this case, the “funnel” on our robot changes the location of the ball hence ballClose will return false every time the ball location was changed by the funnel.
    - This means that move_to_ball() will never return true and the algorithm will never move on to the catch_ball() function.
    - Solution was to remove ambiguity, making the code simpler but only using sensor B to detect the ball’s distance from the robot.

```
        if(ballDetected)
        {
            if(SensorValue[B] >900)                         //side sensor detects within the range acceptable for front sweep
            {
                ballClose =true;                            //ball is close enough for the front sweeper
            }
            else
            {
                ballClose = false;                          //not close enough
            }
        }
    }//end of while
}//end of task
```

Figure 8-9 – Changes made to the ballClose condition to increase reliability

8.4 Performance Review and Improvements to be made

The collector movement strategy worked well to ensure that multiple (3) rotations would ensure a ball collected and with good enough speed to make the ball reach the collected region.

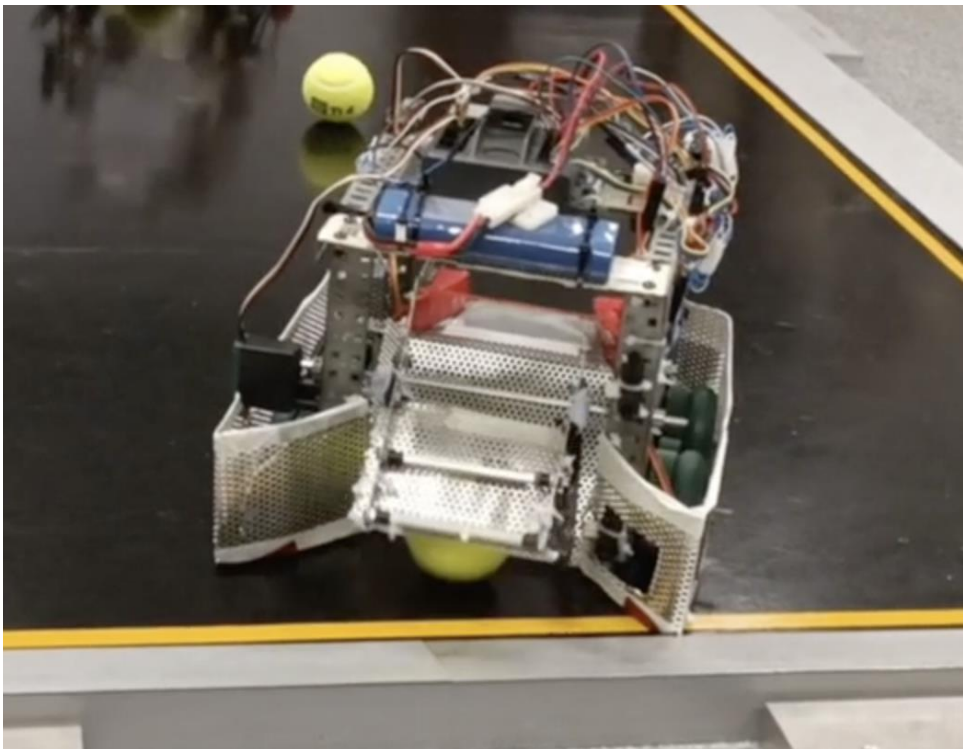


Figure 8-10 - Collector collecting a ball during the competition

Given that the ball collecting strategy was not a roller, the searching and navigating strategy would have to be very precise, which worked out (in the detection ranges assigned to the sensor dismissing noise) in all matches except one wherein the issue was about an edge sensor which disrupted the flow of the collection code state. This in comparison to that of a roller collector where the robot would just have to move towards the ball, required much more tweaking and testing. However, this performed decently within the team's expectations, resulting in the robot performing well in 5 out of 6 test matches.

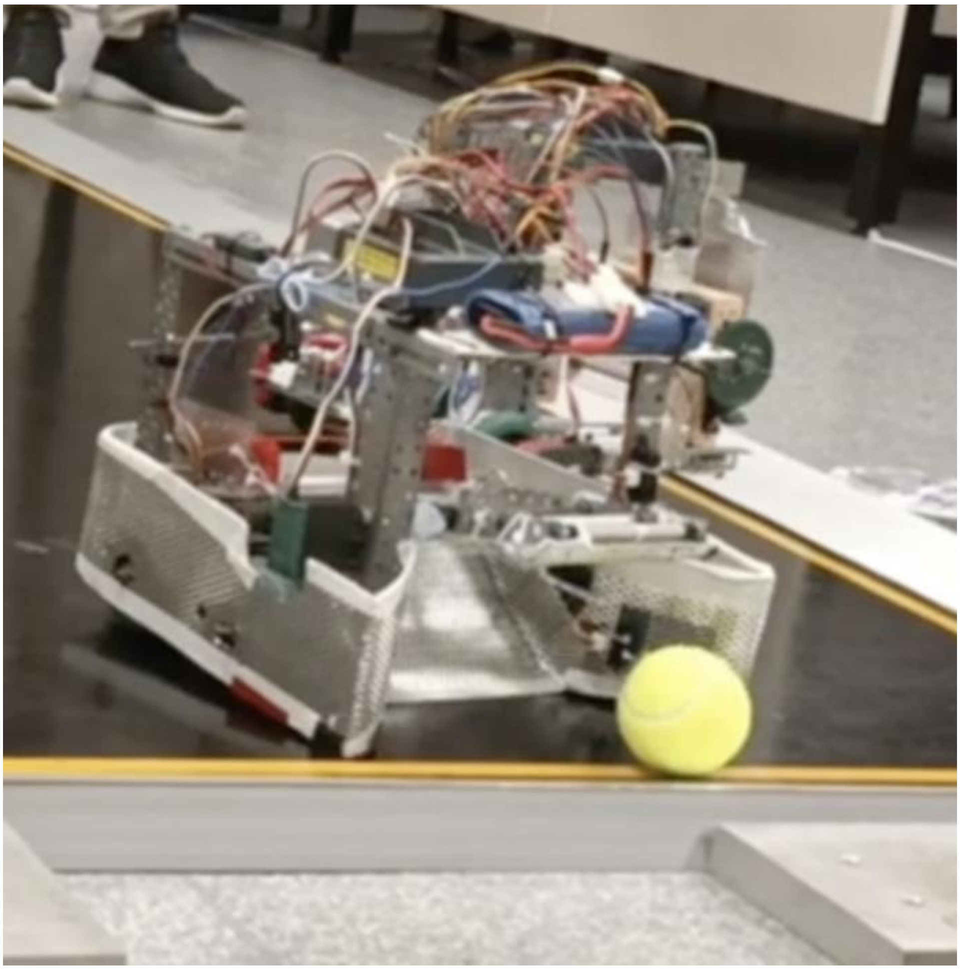


Figure 8-11 - Aligning with the ball with offset to optimise for ball collection

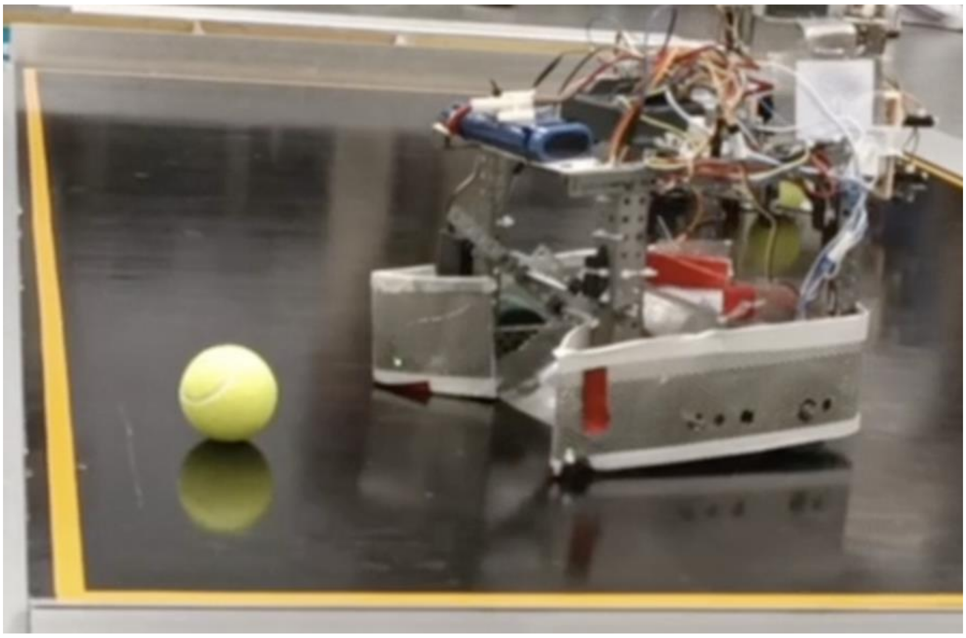


Figure 8-12 - Ball searching and detection process

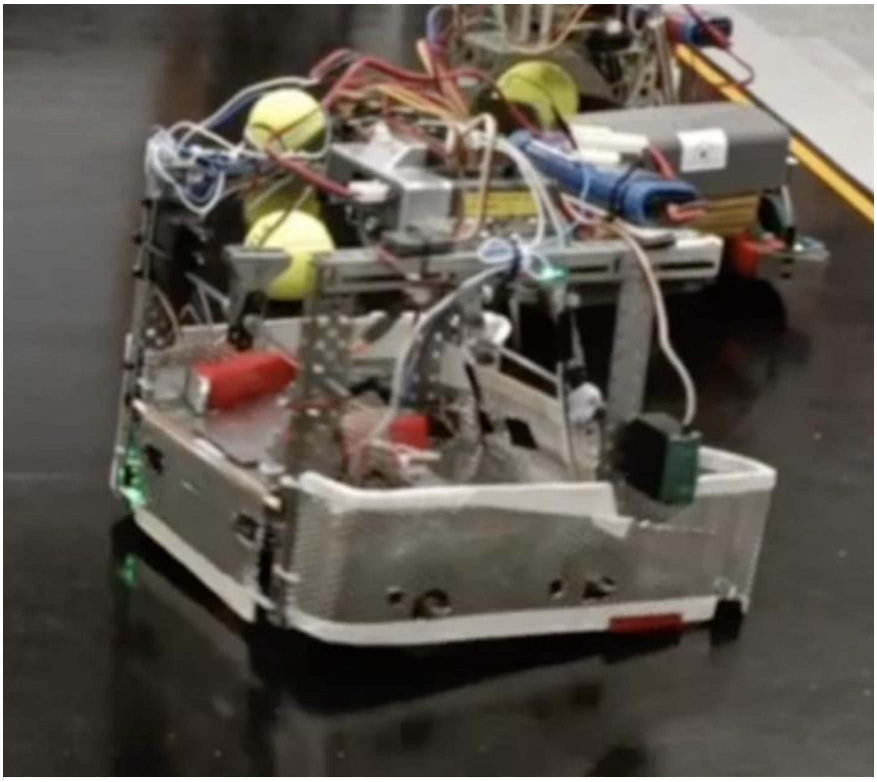


Figure 8-13 - Edge sensor on the back left giving wrong values

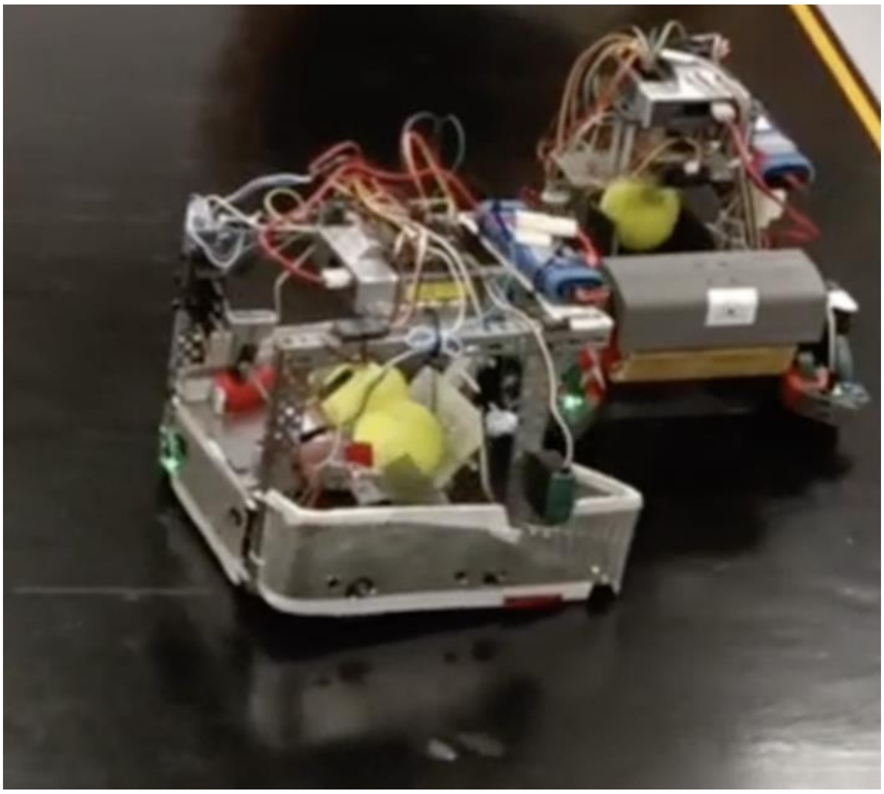


Figure 8-14 - Robot collecting 2 balls at once

The above situation occurred thrice during the competition journey, disqualifying the team thrice. This was due to a loophole in the double ball collection strategy. The ball collection region had place only for one ball and once the ball was collected based on the value of the collected limit switch at the back, the collector flap was to rotate outwards to allow extra balls to be pushed out. However, the path from the mouth of the robot to the holding area in the ramp featured corners for 2 balls to get stuck if both were collected together. This causes the outward sweeping movement, which was intended to release the second ball, did not work.

In another case of ball collection, 2 balls were collected and both reached the holding area in the robot, but both were stuck above the limit switch, hence no ball was detected by the pressure sensor.

To improve the performance in this regard, the ramp should have been narrowed and the space in which the balls could get stuck, would be minimised.

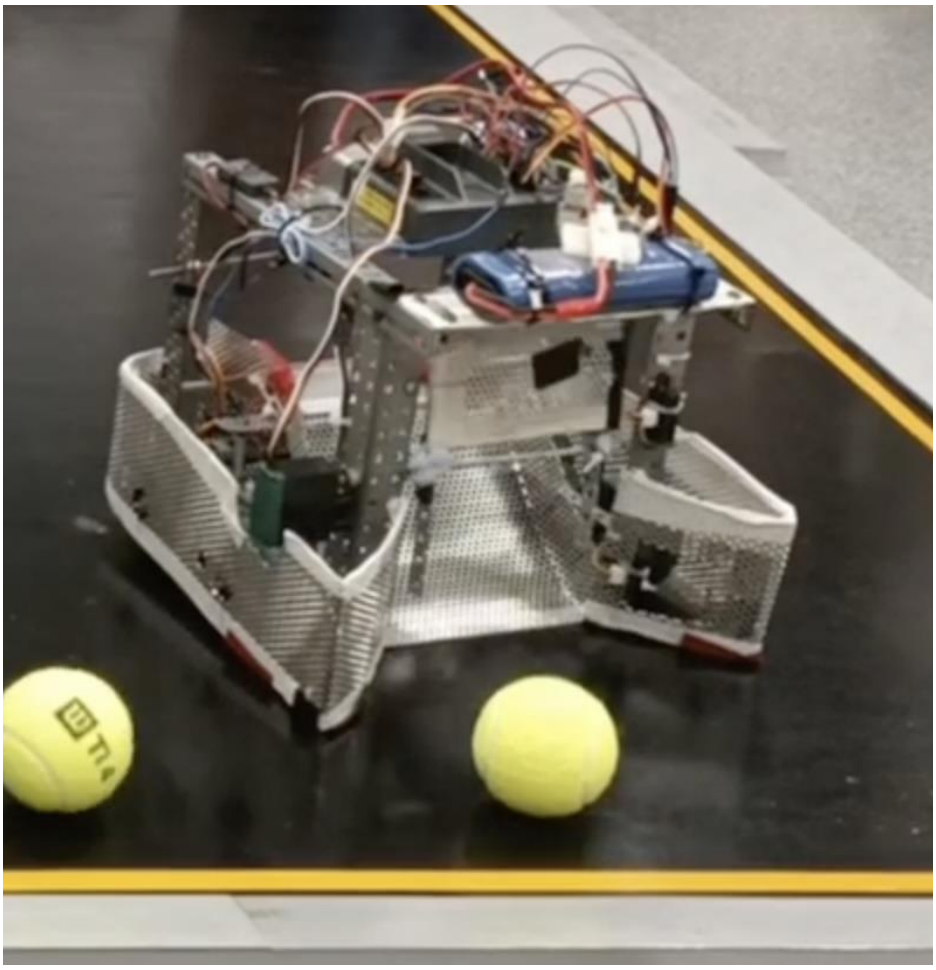


Figure 8-15 - Ball moving in front of the robot, unsuccessful alignment offset

The movements of the balls did not allow proper navigation of the robot to the ball as the sensor placements on one side allowed detection of the ball once only on that side. This made the robot move towards it while aligning itself to the ball. There was a case whereby the ball bounced off the boundary and changed position, going further away out of reach of the collector. Hence the collector movement was not able to capture any balls. To solve this issue, more sensors could have been utilised to increase the range of ball detection, and also perhaps with a more elaborate setup to make the cameras move together with the ball, to ensure reliability in the case of collecting moving balls.

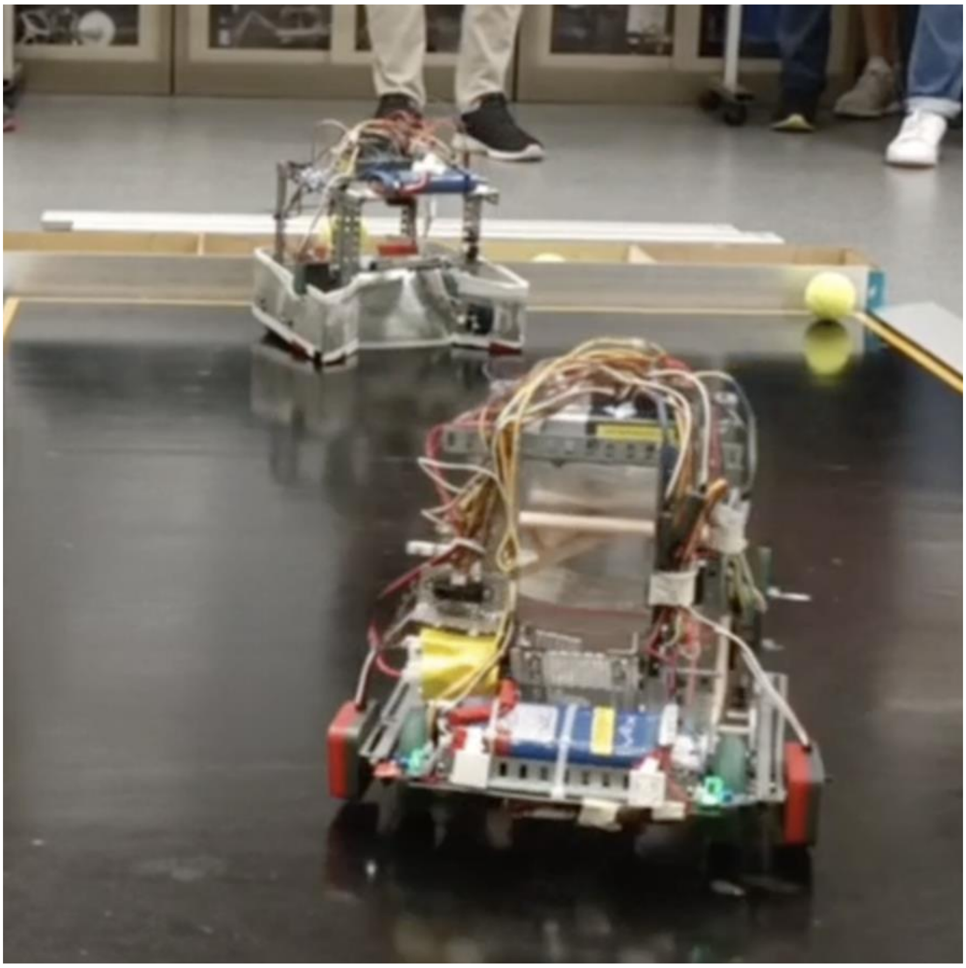


Figure 8-16 - Ball stuck behind robot near delivery area

Also, if a ball was behind the robot while the robot was reversing into the delivery area, it created situations where the robot's delivery process was interrupted. In this case the robot took a lot of time to realign and remove itself from the path of the hindering ball, only then could the robot continue the delivery process. To do this better, a system could have been implemented in the code, to avoid the balls during the delivery process.

9 Valuable Lessons Learnt

An autonomous robot was successfully constructed and was capable of collecting and transporting tennis balls to a predetermined place in a predetermined region. The team finished the robot development from 0 to 1 by following the design process of an autonomous driving robot from an original concept creation to product design, and then to the final product performance test.

The team also effectively employed theoretical information learnt in classrooms, to develop a design with practical importance. Furthermore, they recognized the value of collaboration to successfully complete the project within a team of ten. In this project’s collaborative work, the team members utilized and improved on many skills, such as CAD drawing, hardware building and programming. Communication, effective role splitting, and teamwork also played a huge importance in their success.

Even though some of the team members have diverse views and solutions towards the problems faced, they managed to compile all their interesting ideas and sought a common ground for fulfilling a common purpose in the project. It was amazing to see every member providing each other with morale and mental support to overcome the various challenges throughout the project.

The most fruitful part of this journey was that the team was able to successfully design an autonomous robot vehicle capable of fulfilling its mission. Despite not winning the competition, the process leading up to the successful completion of the project was profoundly enriching experience for everyone involved.

This module allowed the members to work with people of various characters, combining everyone with different strengths to leverage upon, and different weaknesses to overcome as a team. Putting work aside, everyone agreed that this was the most interesting and exciting module and during the competition, they had a fun and memorable time cheering for their robot to win.

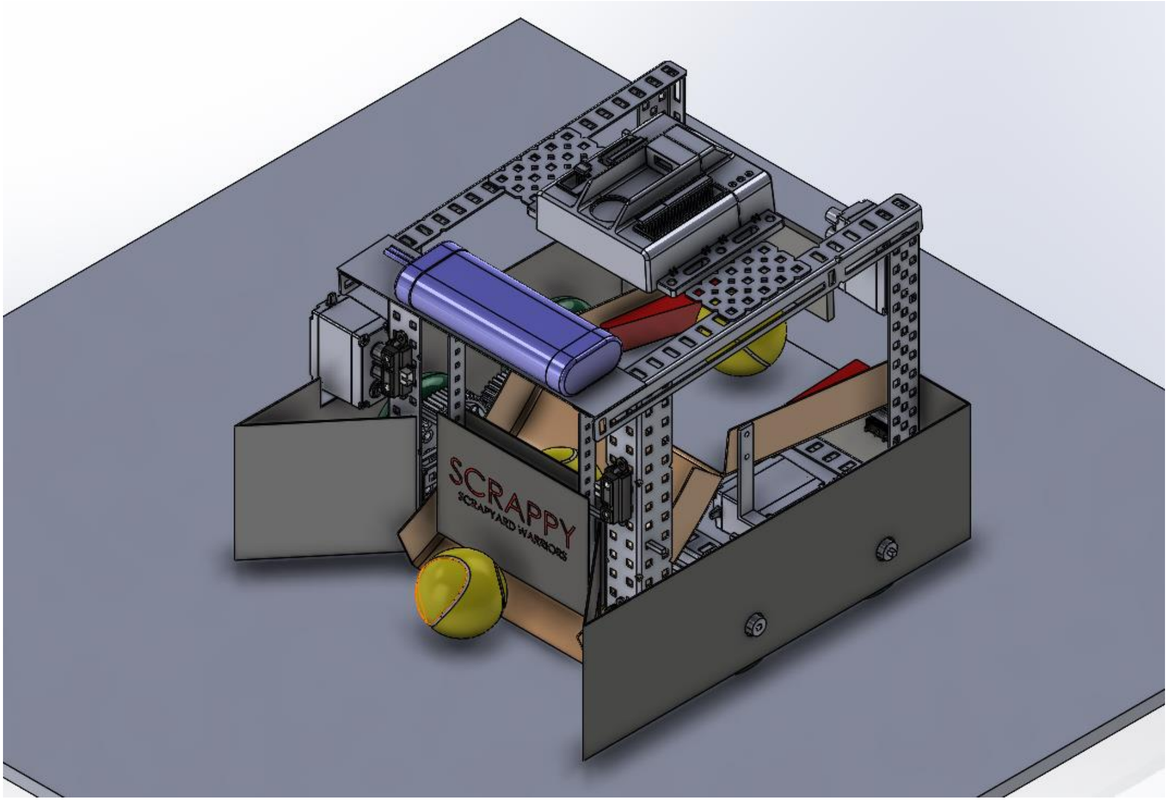
In brief, the team successfully constructed an autonomous robot vehicle that satisfied the specifications and fulfilled its purpose in the competition.

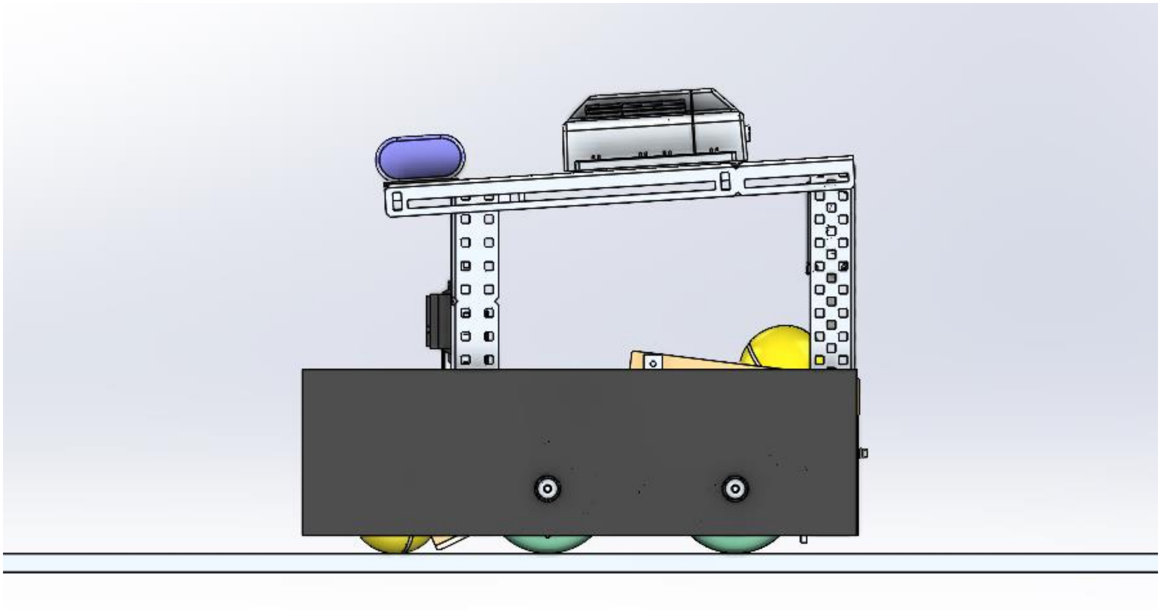
10 References

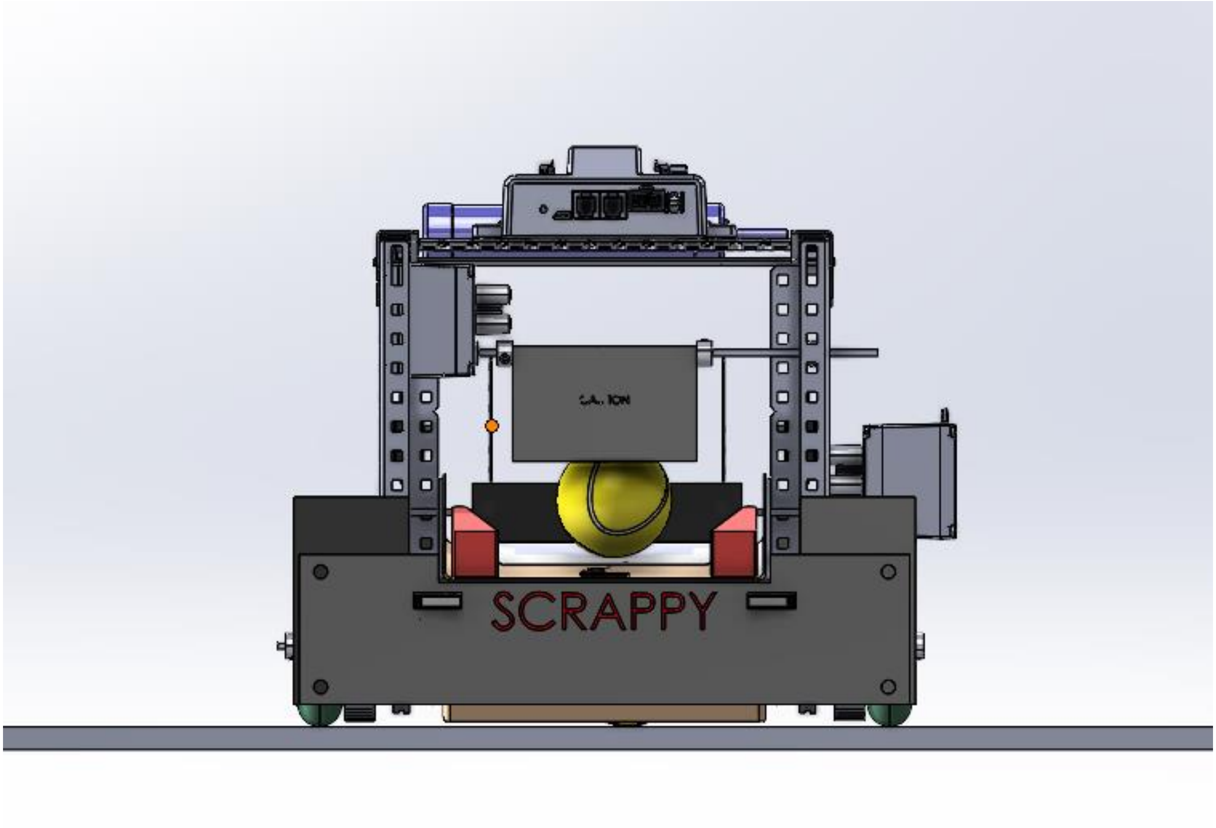
[1] “McGill University,” *Engineering Design*, Nov. 04, 2021. https://www.mcgill.ca/engineeringdesign/engineering-design-process/design-phases-practice/embodiment-design (accessed Apr. 14, 2022).

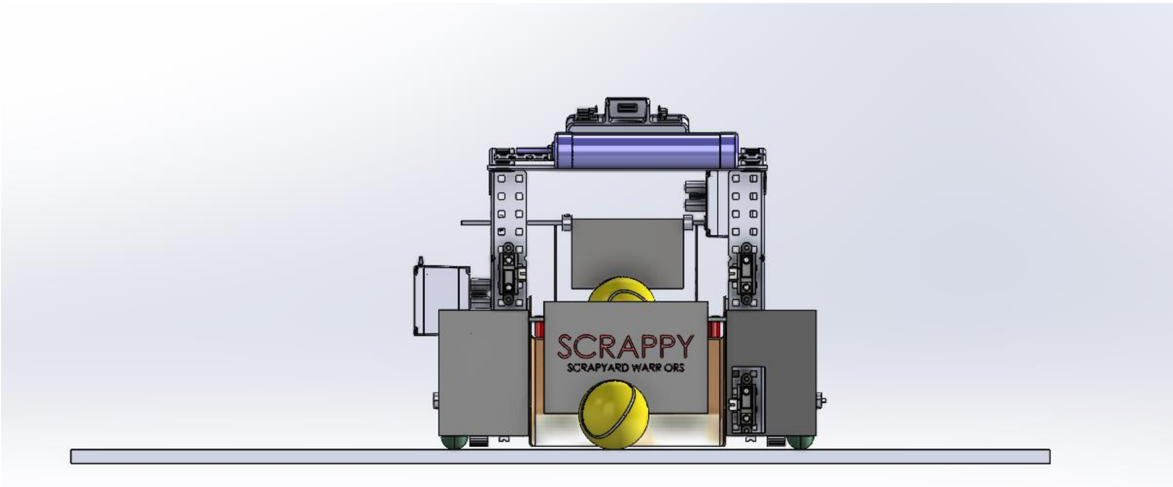
11 Appendices

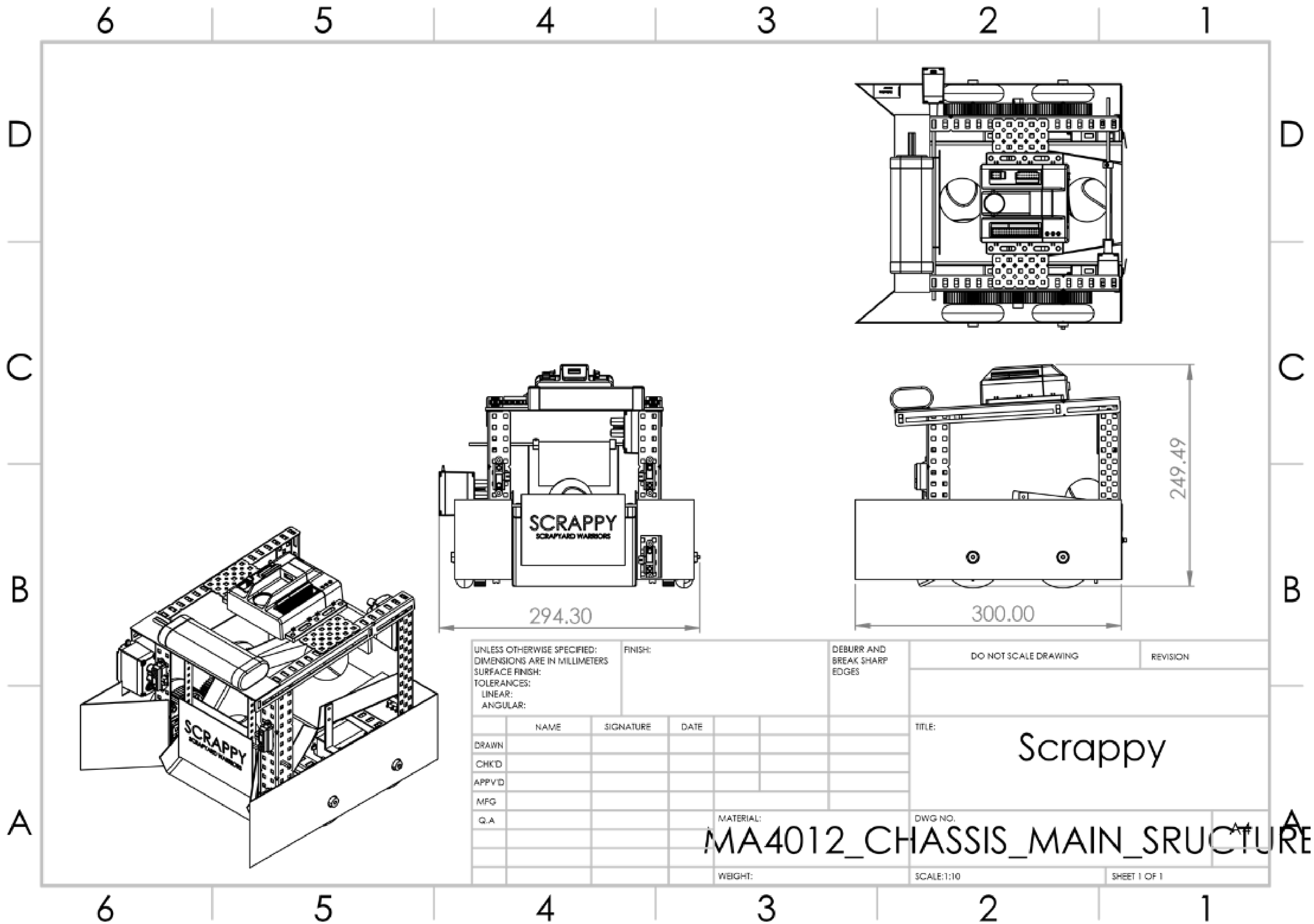
11.1 Overall Assembly Drawings



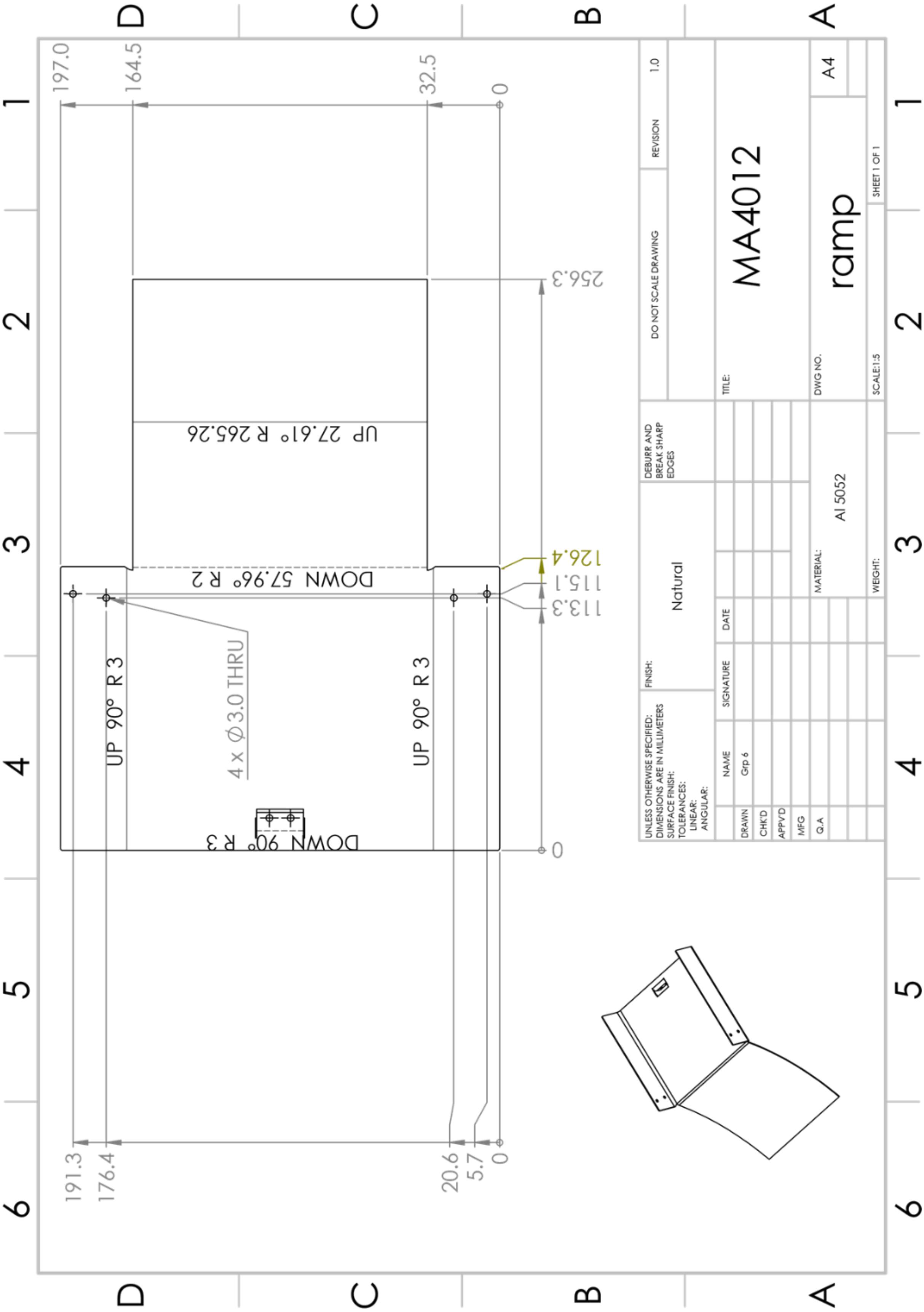


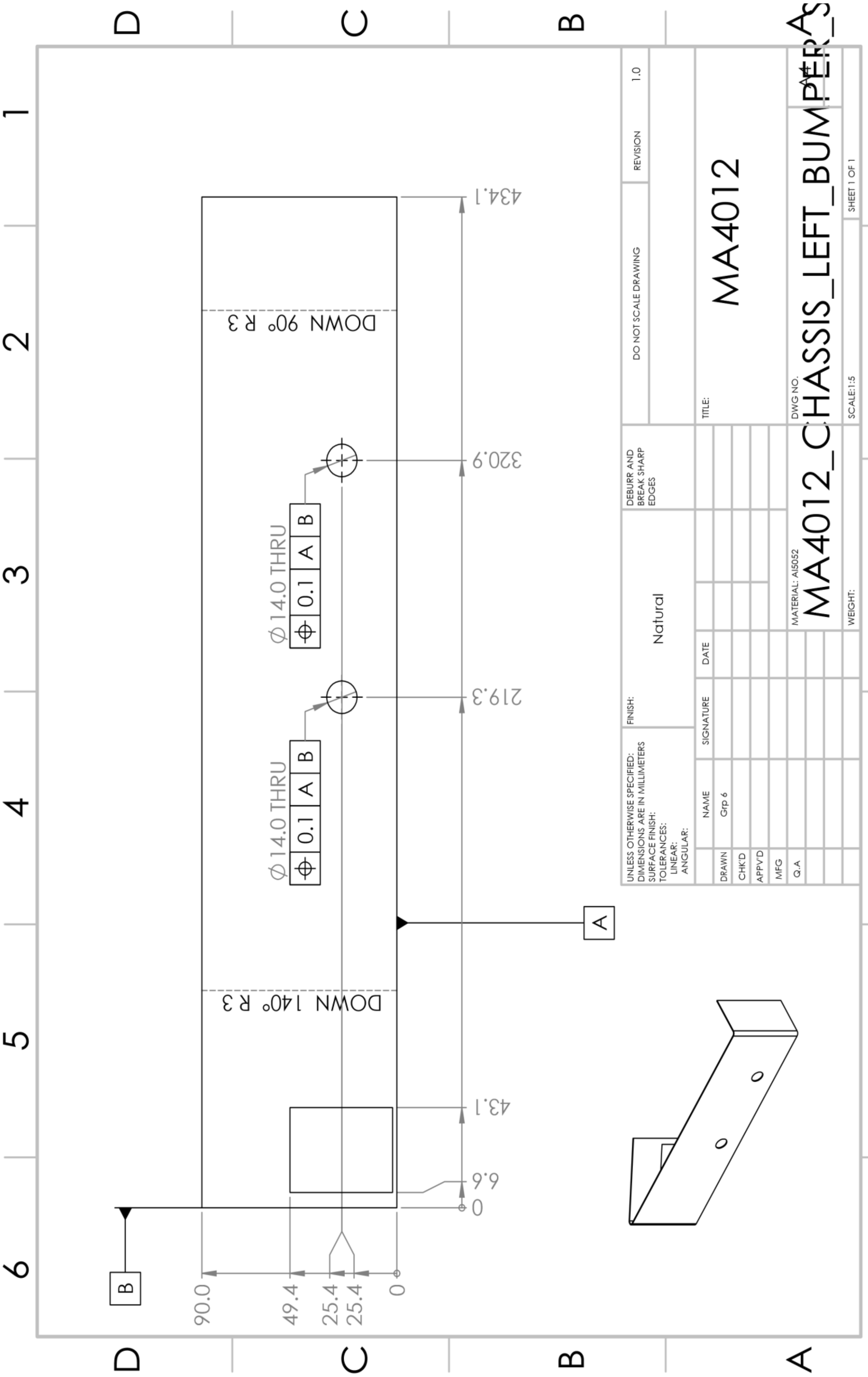

SCRAPPY

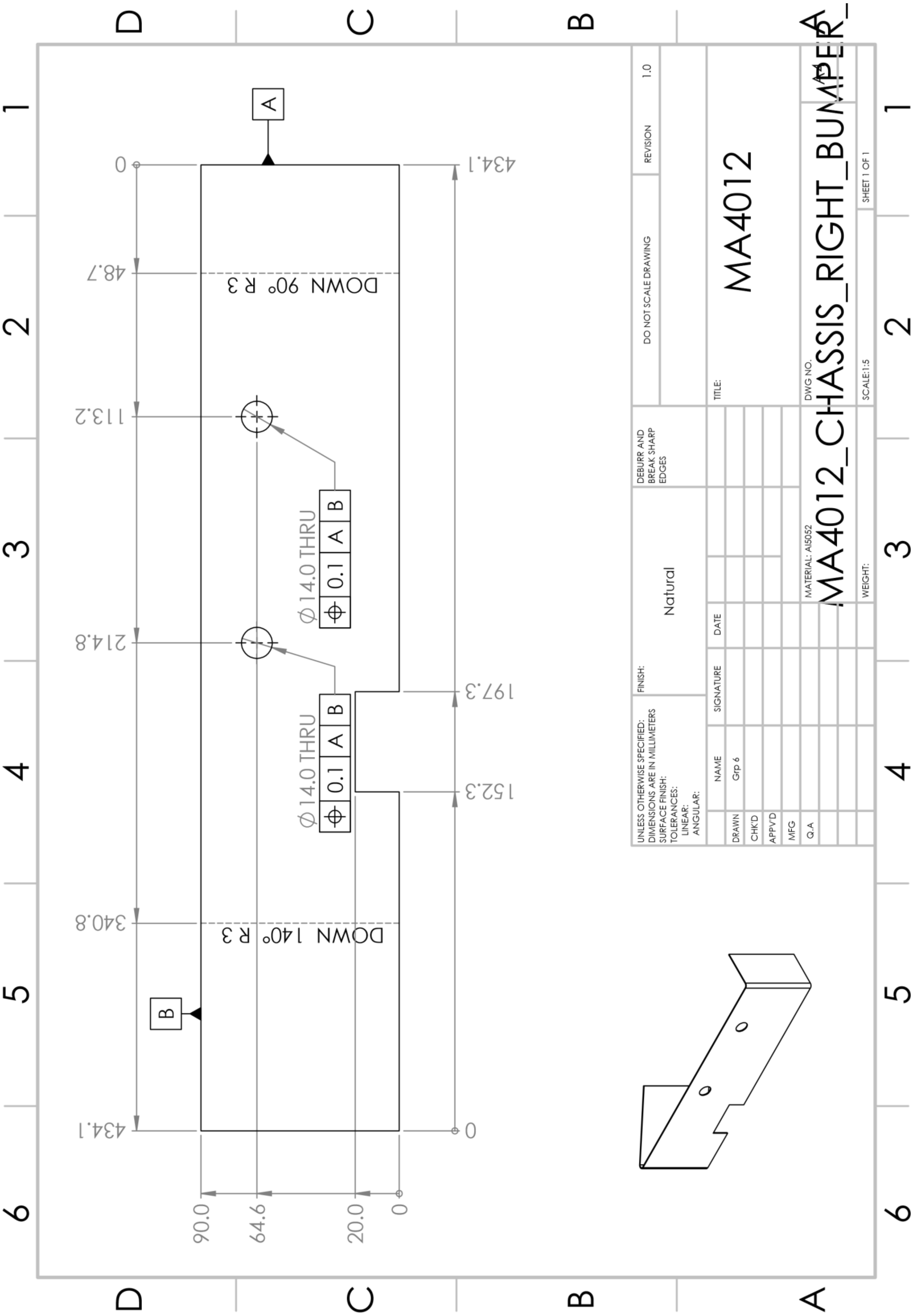

SCRAPPY
SCRAPYARD WARRIORS

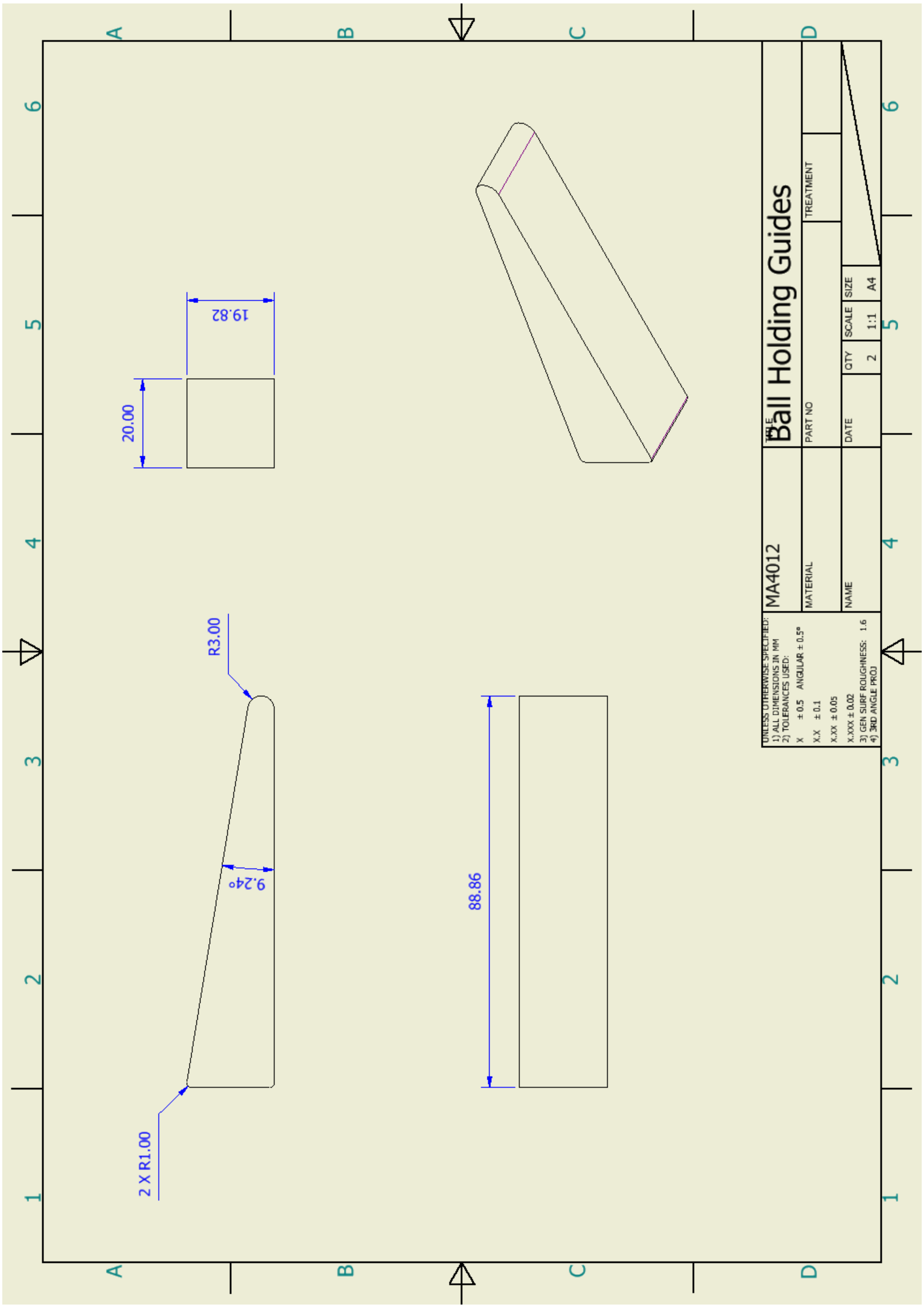

SCRAPPY
SCRAPYARD WARRIORS
294.30
300.00
249.49
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Scrappy
MATERIAL:
DWG NO.
MA4012_CHASSIS_MAIN_SRUCTURE
A4
WEIGHT:
SCALE:1:10
SHEET 1 OF 1

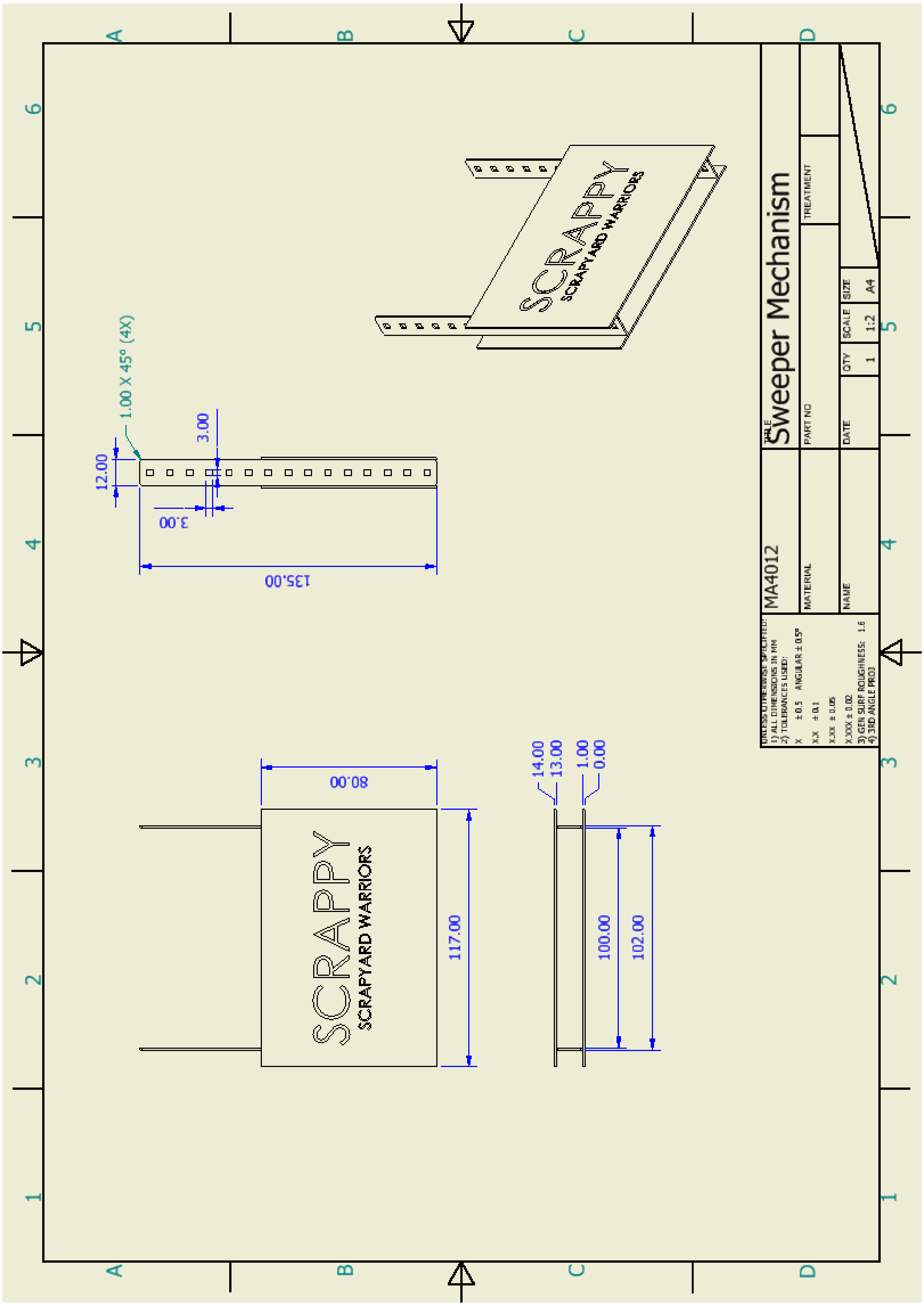
11.2 Part Drawings

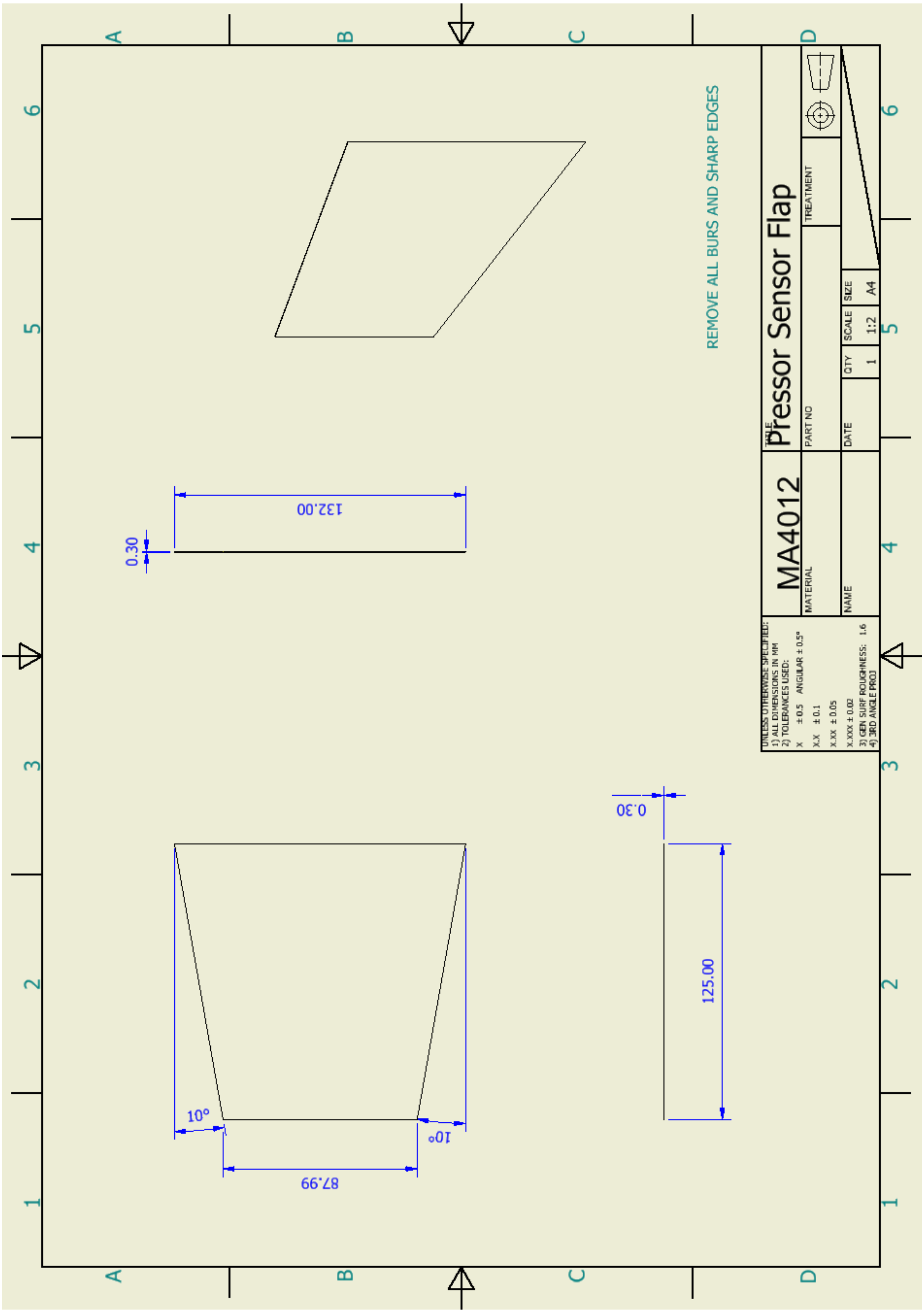



MA4012_CHASSIS_LEFT_BUMPER_S
MA4012
TITLE:
DWG NO.
SCALE:1:5
SHEET 1 OF 1
WEIGHT:
MATERIAL: Al5052
REVISION
1.0
DO NOT SCALE DRAWING
DEBURR AND BREAK SHARP EDGES
FINISH:
Natural
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
NAME
SIGNATURE
DATE
DRAWN
Grp 6
CHK'D
APPV'D
MFG
Q.A
DOWN 90° R 3
DOWN 140° R 3
Ø14.0 THRU
0.1 A B
434.1
320.9
219.3
43.1
6.6
0
90.0
49.4
25.4
25.4
A
B


MA4012_CHASSIS_RIGHT_BUMPER_
MA4012
DOWN 90° R 3
DOWN 140° R 3
Ø14.0 THRU
0.1 A B
A
B
0
48.7
113.2
214.8
340.8
434.1
152.3
197.3
90.0
64.6
20.0
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
Natural
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
1.0
NAME
SIGNATURE
DATE
DRAWN
Grp 6
CHK'D
APPV'D
MFG
Q.A
TITLE:
MATERIAL: Al5052
DWG NO.
A4
WEIGHT:
SCALE:1:5
SHEET 1 OF 1


19.82
20.00
R3.00
9.24°
88.86
2 X R1.00
UNLESS OTHERWISE SPECIFIED:
1) ALL DIMENSIONS IN MM
2) TOLERANCES USED:
X ± 0.5 ANGULAR ± 0.5°
X.X ± 0.1
X.XX ± 0.05
X.XXX ± 0.02
3) GEN SURF ROUGHNESS: 1.6
4) 3RD ANGLE PROJ
MA4012
TITLE
Ball Holding Guides
MATERIAL
PART NO
TREATMENT
NAME
DATE
QTY
2
SCALE
1:1
SIZE
A4


TITLE
Sweeper Mechanism
MA4012
PART NO
MATERIAL
TREATMENT
DATE
NAME
QTY
1
SCALE
1:2
SIZE
A4
UNLESS OTHERWISE SPECIFIED:
1) ALL DIMENSIONS IN MM
2) TOLERANCES USED:
X ± 0.5 ANGULAR ± 0.5°
X.X ± 0.1
X.XX ± 0.05
X.XXX ± 0.02
3) GEN SURF ROUGHNESS: 1.6
4) 3RD ANGLE PROJ
SCRAPPY
SCRAPYARD WARRIORS
1.00 X 45° (4X)
12.00
3.00
3.00
135.00
80.00
117.00
14.00
13.00
1.00
0.00
100.00
102.00



REMOVE ALL BURS AND SHARP EDGES
TITLE
Pressor Sensor Flap
MA4012
PART NO
MATERIAL
TREATMENT
DATE
NAME
QTY
1
SCALE
1:2
SIZE
A4
UNLESS OTHERWISE SPECIFIED:
1) ALL DIMENSIONS IN MM
2) TOLERANCES USED:
X ± 0.5 ANGULAR ± 0.5°
X.X ± 0.1
X.XX ± 0.05
X.XXX ± 0.02
3) GEN SURF ROUGHNESS: 1.6
4) 3RD ANGLE PROJ
132.00
0.30
0.30
125.00
87.99
10°
10°