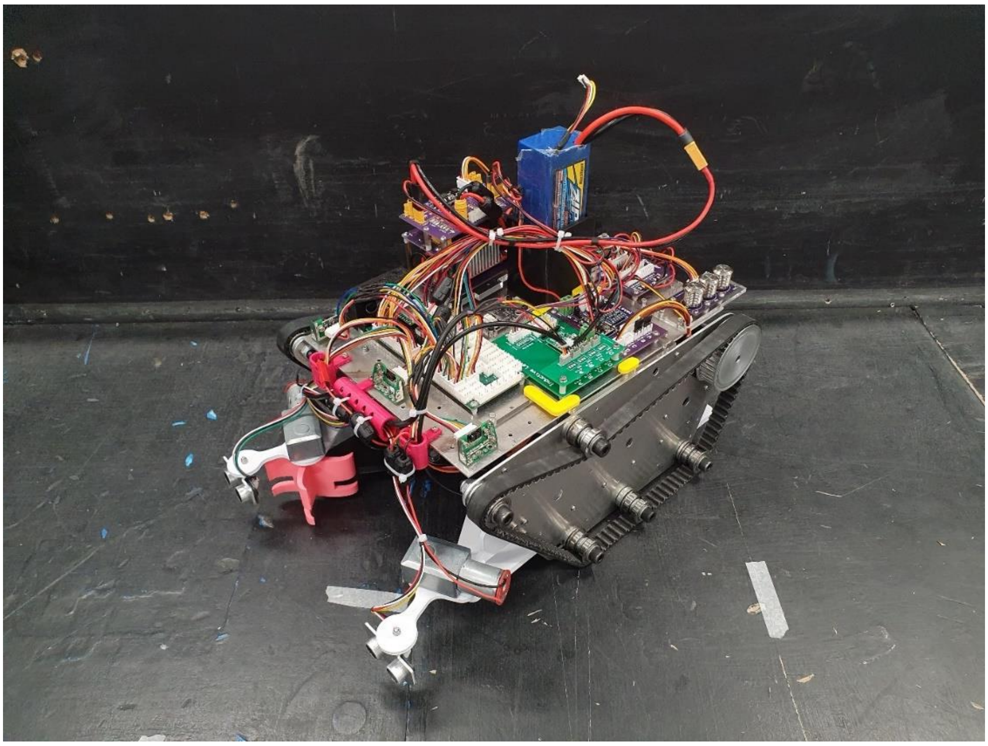


Detailed Design Report

ENMT301 09/09/24 Group 4

Mark day

Kieran James

Jack Willson

Executive Summary

To achieve desired outcomes the robot is divided into subsystem.

- Chassis and Drivetrain
- Target Detection
- Target Collection
- Navigation

The robot used a tank drive, drivetrain with the original gear. This fell short of the robot's speed requirement 0.5 m/s. The actual speed achieved was 0.33 m/s. the chassis and drivetrain also produced a drift of -12%. Custom gears or software bias could improve speed and correct drift.

The target detection uses ToF and ultrasound sensors. A Boolean logic system detects targets and reduces noise with a moving average filter. This was tested using a best of 20 system by placing weights and recording the how successful the detection was. The target detection could be improved by consider replacing ultrasound sensors with more reliable ToF sensors.

The target collection uses a funnel and tray system that replaced the original combine harvester design. Weights are sorted based on material using an inductive proximity sensor and an IR sensor. The new tray system is reliable but does not meet the requirement for collecting weights in any orientation. Removing the combine motors could reduce weight and simplify the robot.

The robot navigation system works by detecting walls with ToF sensors and stores then position using odometry and gyro data. The system works inconsistently, with the robot failing to return home in three out of five runs. Additional ToF sensors and acceleration data from the IMU could enhance reliability.

An FTA was used to identify key components and likely causes of faults. From the FTA, Simplicity was prioritized to reduce risk, especially by replacing moving parts with static guides. The wiring was optimized for reliability, and code was modularized for flexibility. The gate was made from alloy to improve durability.

Table of Contents

1. Introduction

This report will detail the design and manufacturing of the RoboCup robot. It will detail each subsystem of the robot. Each subsystem will be assessed against the requirement set up in the conceptual design report (CDR). The subsystems will also be tested in a results section to improve the robot and recommendations made where the robot fails to meet the requirements.

The goal of the robot is to move out from home base into an area with obstacles and weights placed throughout. The robot must collect weights and bring them home to score points. The area layout is unknown, and the robot must not have any communication from the user. This makes the task difficult. To achieve this the following subsystems have been implemented

- Chassis and Drivetrain
- Target Detection
- Target Collection
- Navigation

The robot target collection has been changed from the CDR. This was due to reliability issues. The detail of the change is explained in section 2.2.

2. Design Description

2.1 Overview

The robot has been designed to maximise simplicity and reliability. The current robot is shown in Figure 1.

It uses sensors to gather information from the world. From this information it moves throughout area to gather weights and returns them to base.

To do this the robot must have primary motion system, the system implemented is a 'tank drive' system. The robot has ToF sensors that return distances from walls and obstacles which allow it to navigate throughout the area.

The robot uses ultrasound sensors to identify all weights, it then uses the navigation system to rotate and drive towards the weight. The weight is collected then is kept on board with a "sheep gate" system using an inductive proximity sensor and servo. The inductive proximity sensor determines if the weight is a target or dummy. The robot then continues forward searching for more weights until it holds 3 on board where it returns home.

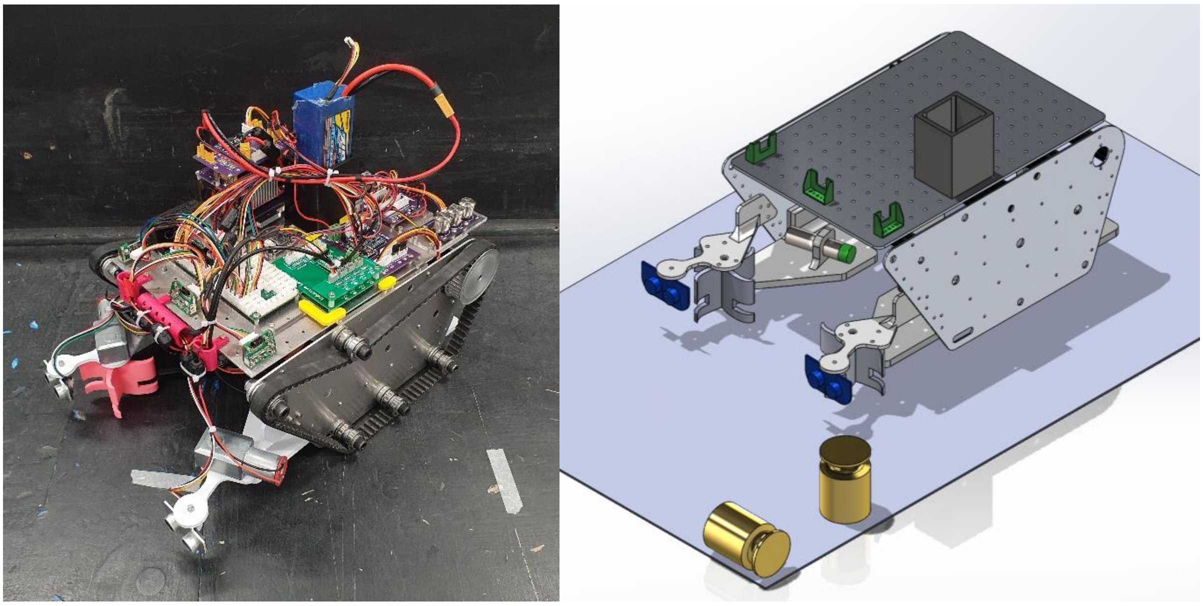


Figure 1: Group Four's robot as of 08/09/24 (Left), CAD model (Right).

2.2 Chassis and Drivetrain

The chassis and drivetrain designs were taken from the CDR and used with minor modifications. The default chassis plates were used along with the standard configuration of tracks and bearings for the drivetrain (Figure 2) rather than printing new track sprockets as described in the CDR. The decision to use the standard sprocket was based on the fact the robot travelled at the desired speed with the alloy sprocket. The motors powering the tracks were 2x 28PA51G DC geared motors with built-in encoders. The motors were powered using a 2-channel DC motor driver and a PWM input. The extrusions were arranged to give the chassis sufficient stiffness and strength while allowing enough space at the front of the robot for the required weight collection components (Figure 3). Using the default chassis plates and extrusions also provided a significant amount of flexibility that was later found useful as the collection mechanism went through multiple further optimisations.

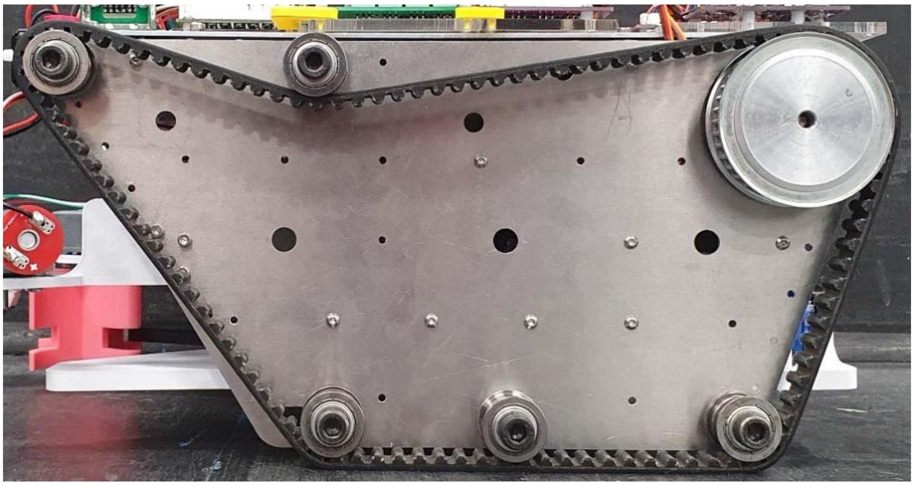


Figure 2: Chassis plate and drivetrain layout.

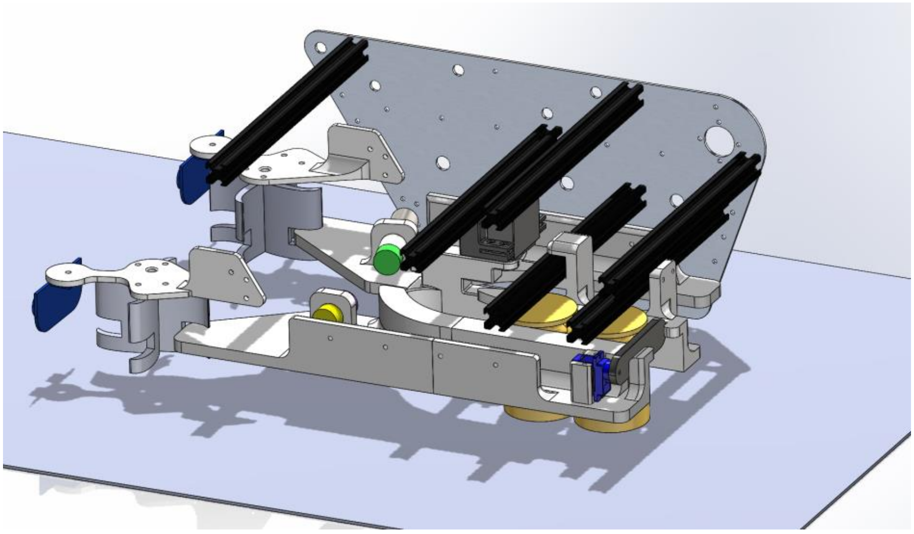


Figure 3: Arrangement of aluminium extrusions.

For mounting the electronics, the provided drilled plate was mounted to extrusions on the topside of the robot. This provided adequate space required for most of the supporting electrical circuits, the Teensy MCU and the battery (Figure 4). To fit some of the circuit boards, small brackets were printed as not all the boards had their own mounting holes. This made the electrical board more compact and discarded the need for a second mounting plate. The ultrasonic driver board and the second DC motor driver were attached to the underside of the electrical board for cable management and space. To help keep cables tidy, manageable and away from moving parts, cable organisers were printed and attached to the front. These also doubled as mounting points for cable ties.

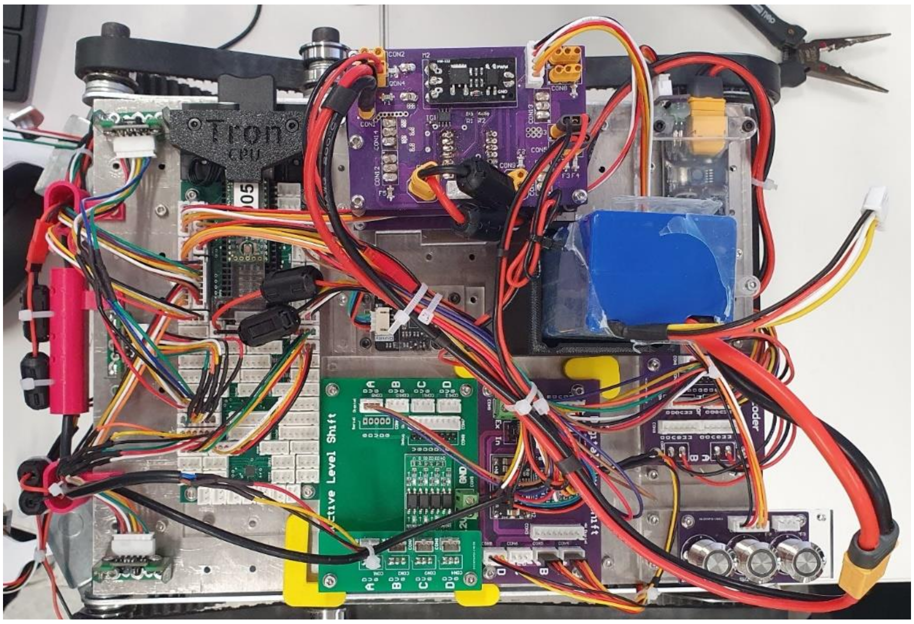


Figure 4: Electronics Plate

2.3 Target Detection

Time of Flight Sensors (VL530X) and ultrasound sensors (HC-SR04) are used for target detection. The TOF's are mounted at the top of the chassis meaning they detect walls and other robots while being above the height of target weights. There are two ultrasound sensors placed lower to the ground at the front of the robot. In code a Boolean, "is_target_found", is assigned a true value if the digital TOF read is larger than the digital ultrasound read. The TOF is likely to detect a wall while if the reading is the same for the ultrasounds, they must be detecting the same wall. Additionally, a moving average filter is applied to reduce noise, smoothing the output signals.

2.4 Target Collection

After the CDR was completed and the concept was selected, once the design process had begun it became apparent that the chosen design would be difficult to implement and manufacture. Further investigation also found some reliability issues around the 'combine harvester' design for collecting weights. One of which was difficulty directing the weight to a specific location to be examined if the weight is metallic or plastic. A solution to this was to have a tray that funnels weights into a channel (Figure 5). The technical drawing for the right-hand tray can be found in appendix C. The funnel would position the weight in the channel (without knocking the weight over) where a gate would then stop the weight in the channel next to an inductive proximity sensor.

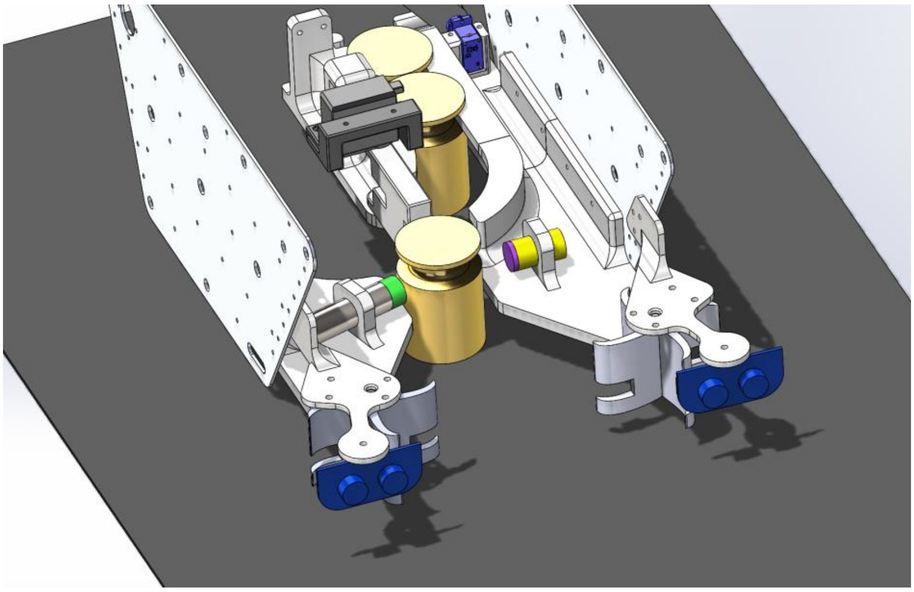


Figure 5: The trays make up the funnel, the channels and brackets for sensors.

The gate would then draft the weight to be kept or discarded depending on if it was metallic or not. Plastic weights are free to be discarded from the rear of the robot and target weights are directed into a closed off channel and held in the robot for the remainder of the round or until the robot travels to it’s home base. The holding bay (Figure 6) was designed to fit and carry three target weights of any weight. A separate servo with a gate prevents them from exiting the bay at the rear. The left-hand tray and the centre partition create the holding bay channel. The channel is designed to have a lip that fits into the slot of the weight preventing it from falling out the bottom of the robot.

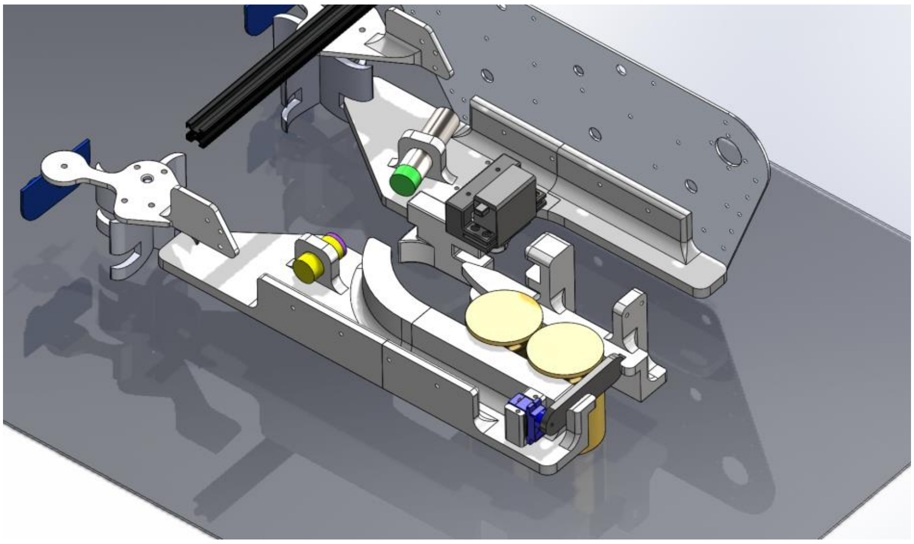


Figure 6: The weights are held upright and contained in the robot in the holding bay.

To keep parts of the original concept in the new design, the ‘combine harvester’ was remodelled into two vertical sweepers. The purpose of these were to increase the width of possible collection area by sweeping weights from a wider span into the funnel. Two DC geared right-angle motors are mounted to the two front brackets and drive the sweepers.

The gate has been specifically designed to be able to rotate 90 degrees into the holding bay (Figure 7). A technical drawing for the gate can be found in appendix C. This is so the gate can push the weights further down the channel to prevent a weight from travelling back out the front of the robot in the occasion that the robot must reverse. The hook in the gate also provides higher chance of gate pushing the weight into the holding bay rather than jamming against the weight. The gate is driven by a HX12K servo that can provide up to 9.4kg/cm of torque, which is adequate for pushing and holding up to three weights. The servo has a 3D printed bracket that is mounted to an extrusion, as seen in Figure 7 and 8.

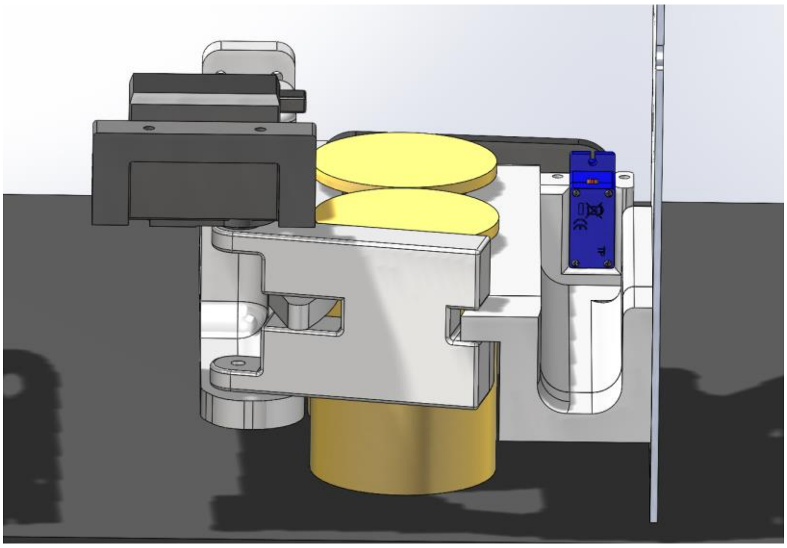


Figure 7: Gate securing two target weights in the holding bay.

To support the bottom of the gate a M4 cap screw is threaded into the bottom of the gate acting as a removable pin and the head of the bolt rotates in the centre partition. The reason for having a removable pin is so there is easy access to the cap screw holding the gate to the servo and so the partition can slide backwards to remove. The centre partition provides the necessary support to hold the target weights and creates one side of the holding bay. To mount the partition to the robot, a slot at the front fits over one extrusion and two bolts hold the rear to a second extrusion (Figure 8). The design allows for significantly simpler assemble as access to the front extrusion is limited and this results in a total of only 3 bolts holding the partition in. An advantage of mounting the servo and partition to the extrusions is the ability to adjust the width of the channel for fine tuning.

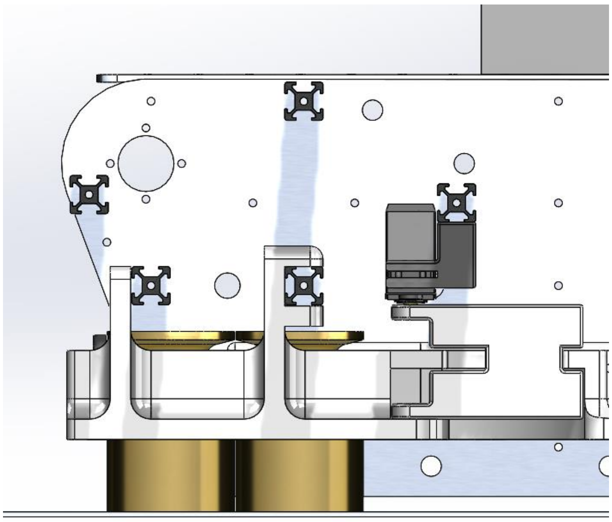


Figure 8: The centre partition mounted on two extrusions.

2.5 Navigation

The robot detects walls using three ToF sensors (VL53L1X). One centre sensor detecting walls in front of the robot and the other two at the front looking left and right to detect walls next to the robot.

From the data sheet the FoV of the ToF is 27 degrees. This means that the robot should be able to detect any obstacle in front of it at a distance of 450mm or greater as shown in Figure 9. The sensors can detect walls from 40 to 4000 mm.

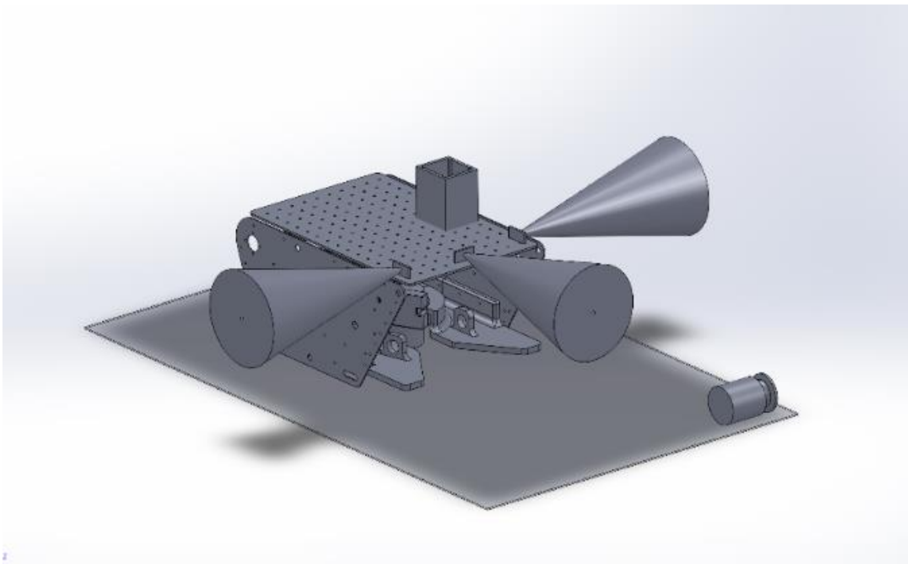


Figure 9: ToF Field of View

The robot searches for weights based moving into the most free space. The robot moves forwards until it detects a wall. Once detected it uses the left and right sensors to determine which direction has walls further away and turn towards that direction. Moves until the it detects a wall and repeats the process.

The robot navigates by remembering where it is and certain locations it has been. It does this by storing an x, y position where 0,0 is home base but also it can then store approximate position of obstacles and weights as it comes across them. The x, y positions are calculated by combining the angle from the gyro and the odometry recorded from the encoder.

To move the robot to a given x, y position the function "MoveTo" can be called. This function turns the robot until it is facing the correct heading. From there it will start to move forward till it reaches a wall. Once there is a wall or obstacle in its way it will turn towards the most free space and move forwards. Once it no longer sees the obstacle it will turn back towards the given x, y heading and move forward again repeating the process if another wall is encountered.

2.6 Software Design

The robot is controlled by a Teensy 4.0 microcontroller with an IO expansion board. The teensy has 40 pins but 24 in the given configuration. Having a IO expander allows for the additional IO to be plugged in and multiplexed to. It also allows for standard connector types, serial, digital, i2c etc.

The code is developed and built in VS code as a platformIO project. This allows for greater functionality as the code written is C++ not Arduino C++.

The code is broken up into modules:

- main.cpp: task scheduler set up and run.
- finite_state_machine.cpp: all states and logic to switch between them.
- return_to_base.cpp: calculates where home is, if home and return home.
- weight_collection.cpp: detects weights and functions to pick weight up.
- smartmove.cpp: moves the robot based on sensors, move while avoiding walls etc.
- motors.cpp: call to set motor/actuators values.
- sensors.cpp: takes in data from sensors.

The module dependency tree is shown in Figure 10.

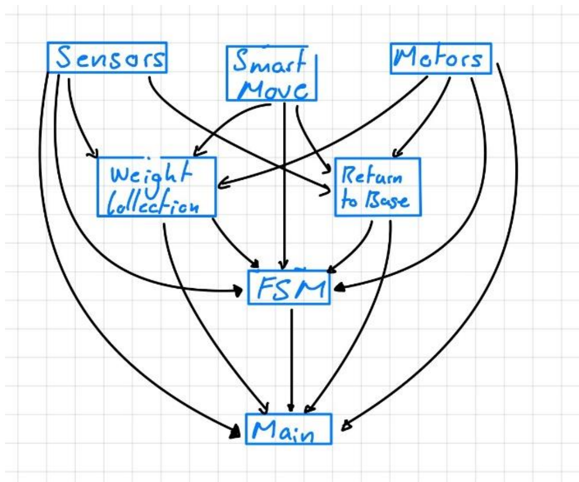


Figure 10: Module Dependency Tree

The modules are designed such that there is low level logic which only deal with getting inputs and outputs from the real world. Moving down the tree more algorithms and less input and output is found in the modules. The modules are based on subcomponents. This means algorithms can be changed without needing to change the entire code base.

The Finite State Machine is the powerhouse of the program. It sets what functionally is allowed and ensures that each task is carrying out in the order it should be. Figure 11 shows the states and how they interact with each other.

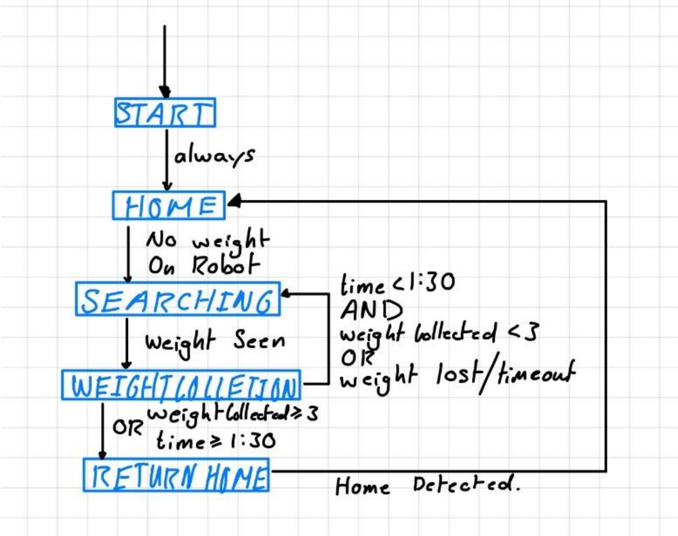


Figure 11: State Machine Diagram

- Start State: ensures all actuators are in position.
- Home State: unloads weights.
- Searching State: moves the robot based on a most free space algorithm.
- Weight collection State: move towards weight and determines fakes.
- Return Home State: returns the robot to home avoiding obstacles.

3. Results

3.1 Chassis and Drivetrain

The taking the average speed of 3 runs the speed of the robot was determined to be 0.32 m/s this falls short of the required speed 0.5 from requirement 2.4.

Another factor measured was the motor drift this was not explicated included in the requirements but affects the ability of the to effectively drive. The drift was measured to be -12% or in other words, the robot would move to the left 120mm every meter moved.

3.2 Target Detection

The ultrasounds must be tested to see if they contribute to the design requirement 1.8. If the Robot detects a weight in the arena, it shall be able to move towards it successfully. To test the ultrasound sensors, the robot was placed in the practice arena with a target weight placed in its view within 15 degrees of either side. The serial monitor was used to determine if the robot had successfully detected the weight. The target weight was placed at different positions including both near and far with far weights being placed within the sensor's detection limit of 400cm. Tests were also completed with multiple weights in view with a successful test being decided if the robot identifies the closest weight and correctly moves towards this weight. Table 1 shows the score for each category being the number of times out of 20 tests a successful detection occurs.

Table 1: Ultrasound Detection Test

TEST	NEAR TARGET	FAR TARGET	NEAR TARGETS	FAR TARGETS
1 ULTRASOUND	16	12	12	10
2 ULTRASOUNDS	20	16	17	16
3 ULTRASOUNDS	20	15	4	18
4 ULTRASOUNDS	20	19	16	17

The results show a reduction in accuracy when detecting multiple weights as more ultrasound sensors are added. This is likely due to the complexity of the code required as more sensors are added. The near target was the most accurate however with multiple targets weights this accuracy dropped off significantly. To further increase the accuracy there should be further coding improvements.

3.3 Target Collection

The new changes in the weight collection mechanism resulted in failure of one the requirement proposed in the CDR. The tray system does not allow weights to be collected in any orientation as laid out in requirement 1.9. The tray was designed to act low on the weight to prevent toppling, this proved useful as driving towards a weight was tested 10 times and not once was a weight tipped over. After some fine adjustments of the position of the inductive proximity and digital IR sensor, the robot was able to distinguish accurately when a weight was in position to be tested. The inductive proximity then successfully recognised if the weight is metallic or plastic for 100% of the 10 tests. These tests were carried out by driving the robot towards the weight with external weight detection turned off.

Separate testing was used when examining the effectiveness of the channel and holding bay. The gate was set in the open position for the holding bay and weights were placed in the channel as if they were detected as metallic. The robot would then drive forward, and the weight would slide into the holding bay. This was successful 9 of 10 times where one time the weight got stuck at the intersection of the gate and the partition. This is easily resolved by closing the gate towards the holding bay and the weight would be pushed into the holding bay. The release mechanism was tested with a full holding bay and opening the rear gate and driving forward. All three weights were deposited out the robot.

3.4 Navigation

The navigation system works well when it works and when it doesn't it causes it to get stuck on obstacles without being able to recover or can no longer find home which fails the requirement of 2.3 for the concept report. It never hit any walls.

Table 2: Navigation Results

RUN	OBSTACLES HIT	STUCK	QUADRANTS VISITED	HOME
1	2	0	4	1
2	3	0	4	0
3	2	1	3	0
4	4	2	2	0
5	3	0	4	1

Table 2 shows the performance of the robot moving from home into the field then back to home after 60 seconds. It should be noted that it only made it home twice out of 5 runs.

It was observed that if the robot hit obstacles or gets stuck the wheels would spin without it going anywhere. This would mean that the robot thinks it has travelled without doing so and hence its map no longer reflects reality and cannot find real home again.

Two main factors that were observed to cause the poor performance. One is the front Tof missing obstacles as shown in Figure 12.

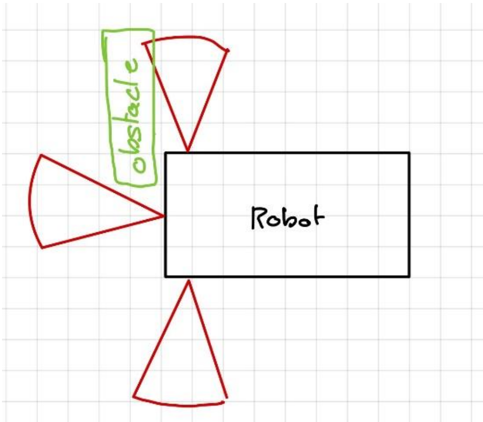


Figure 12: Obstacle Missed Diagram

The other lesser cause was the motor drift causing the robot to always turn left when its meant to go forward. Although this shouldn't cause the navigation system to fail. It causes it to stop redirect itself and then start again.

3.5 Cost Analysis

We used 386 grams of PLA for 3D printing, but 300 grams of PLA was included in the budget. This resulted in an extra cost in PLA of $4.30 NZD. All parts designed were manufactured using 3D print except the main gate, this will be milled out of aluminium for strength as it provides a stronger mounting point to the servo. The aluminium has an approximate extra cost of $11.40 NZD. We used 1 extra ToF sensor (long-range) than what was provided, this had an extra cost of $10 NZD. Everything else is included in the budget. The total additional cost of the robot came out to $25.70 NZD which is well within the additional costs budget of $50. A full bill of materials can be found in appendix 7.1.

3.6 Requirement Analysis

Table 3 shows what requirement has been meet. Explanation of fail and recommendation can be found in their respective result/design sub system or further development. Stars refer to inconsistency.

Table 3: Requirements Specification

Functional + Performance Requirements	*Description*	*Pass/fail*
1.1	The Robot shall be able to recognise if it is at 'home' or in the playing arena	PASS
1.2	The Robot shall be able to distinguish between its home and its opponent's home.	PASS
1.3	If the Robot is positioned at its home and contains weights, it must place the weights on home.	PASS
1.4	The Robot should return home once it contains three weights.	PASS
1.5	The Robot shall be able to recognise walls in relation to itself.	PASS*
1.6	The Robot shall be able to traverse all the following obstacles:	PASS
1.6.1	Speedbumps of minimum 25mm high.	PASS
1.6.2	Ramps of minimum 100mm high and 30% gradient.	PASS
1.7	The Robot shall be no greater than 400 mm wide.	PASS
1.8	If the Robot detects a weight in the arena, it shall be able to move towards it successfully.	PASS
1.9	The Robot shall be able to collect weights placed in any orientation.	FAIL
1.10	The Robot shall only collect weights in the playing arena.	PASS
1.11	The robot should not collect fake weights.	PASS
1.12	The robot shall be able to collect weights against arena walls and obstacles.	PASS
1.13	Once identified The Robot should not interact with a dummy weight consecutively.	PASS
1.14	The Robot shall cash in weights before the end of the round.	PASS
2.1	The Robots battery shall last for at minimum, 2 minutes.	PASS
2.2	The Robot shall be able to pick up a weight once it enters a picking up mode within 5 seconds.	PASS
2.3	The Robot shall not be stuck on in a subsection of the arena for more than 20 seconds	FAIL*
2.4	The Robot shall be able to move at least 0.5 m/s whilst fully loaded.	FAIL

4. FTA

In general from the FTA, it was decided that all subcomponents should be kept as simple as possible. For example, the moving combine was replaced with a guide rail system and moved to a less important sub system. This was due the added risks involved with moving parts.

From the FTA a common fault was wiring issues as this could lead to failure of the drivetrain, weight collection, and detection system. Due to this the importance of designing for reliability in the wiring system, Figure 13 was prioritised.

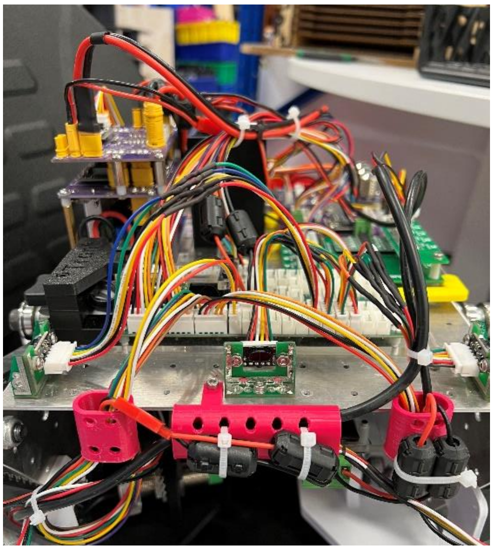


Figure 13: Photo of Wiring System

The Number of wires were minimised, and tip tied together. This increased the manufacturing time but also increased the reliability. Wire conduits were made to protect the wires from hard edges and organise wires. Connections were kept as close as possible, and wires shorten where they could be.

The code was determined from the FTA to be a common fault. Due to this coding style and naming standards were implemented. The code was modularised such that the logic/computation and real-world inputs/outputs were separated. This allows the code to be more reliable to changes. Camel case naming was implemented to improve readability.

From the FTA poor collection design was a critical fault. Due to this the gate will be made out of alloy. Doing this decreases the chance of the gate breaking and improves the reliability when other robots collide with it.

5. Evaluation / Further Development

5.1 Drivetrain

The drive train could stand to have a custom gear. This would allow for a higher ratio gearing and thus a faster speed meeting requirement 2.4. This would also allow for an opportunity to address drift as a bias could be put on one gear. Otherwise, a software bias can be used to reduce drift.

5.2 Navigation

Increasing the number of ToF sensors on the front of the robot to at least two and then taking the minimum value between them and considering that the front ToF sensor value. This would eliminate the robot clipping obstacles and meet requirement 2.3.

Adding the acceleration data from the imu into the navigation system could improve the reliability. Normally acceleration data is integrated for displacement, and this could replace the need for encoders that miss interpret information when the wheels slip. This was found to be a suboptimal solution due to sensor drift. Even though there are methods to eliminate drift. A more interesting solution would be to use the imu to detect if an obstacle has been struck (a large spike in acceleration) and then throw away encoder values whilst stuck or make the robot back up.

5.3 Weight Detection

The ultrasound sensors for target detection work fine but could be improved. An issue that was discovered was that the ultrasound sensors were taking a very long time to process when objects are further than 4m away. This was causing the CPU to hang and cause unexpected behaviour. This may not be an issue in the arena as object would be within 4m most of the round, but it is possible. One idea is to use TOF sensors instead of ultrasounds as these have been very simple to implement so far. Further testing would be required to determine if these would be a successful alternative.

5.3 Weight Collection

The combine motors could be removed entirely as the effectiveness of them have been reduced as the weight collection system includes a 3D printed “funnel” to collect weights. The combine and motors add unnecessary weight and overcomplicates the robot while also not improving the weight collection a significant amount. Power consumption is not currently an issue for the robot however removing the dc motors running the combines would reduce a lot of weight from the robot while also halving the number of DC motors running reducing power consumption.

6. Contribution statement

6.1 Mark

For the robot, I contributed to the software design, moving it to a platformIO project and creating modules. I implemented the navigation system in code and electronically. I also contributed to the electronic layout and cable management. I assisted in the development of the target collection combine wheels. For the report I wrote: The design and result for the navigation, the effect of FTA on design, design of software and the summary.

6.2 Kieran

I contributed mostly to the coding of the target weight detection and collection. Designed CAD models for a few minor parts. I also assisted in the setting up and testing of sensors such as time of flight, ultrasound, colour and inductive proximity. For the report I completed the target detection overview and FTA as well as the results section for target detection.

6.3 Jack

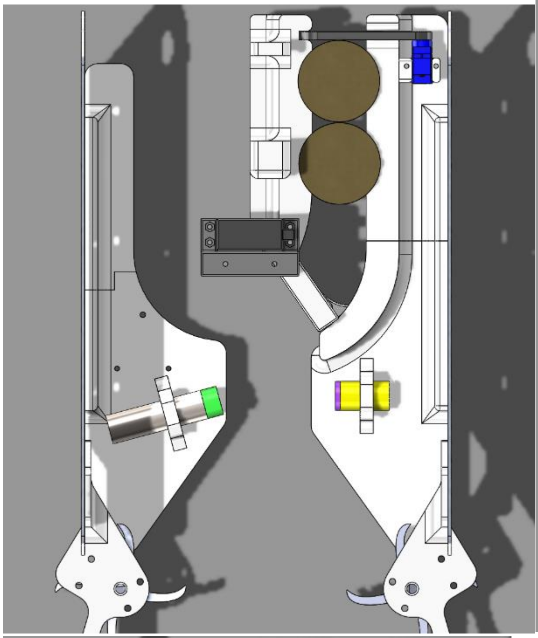
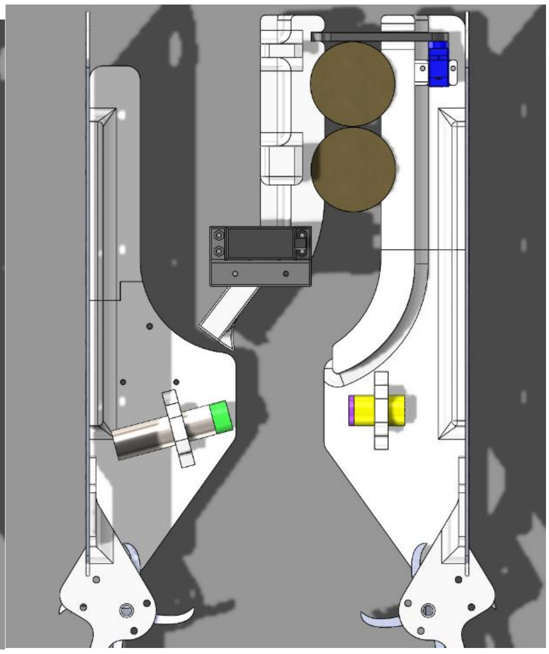
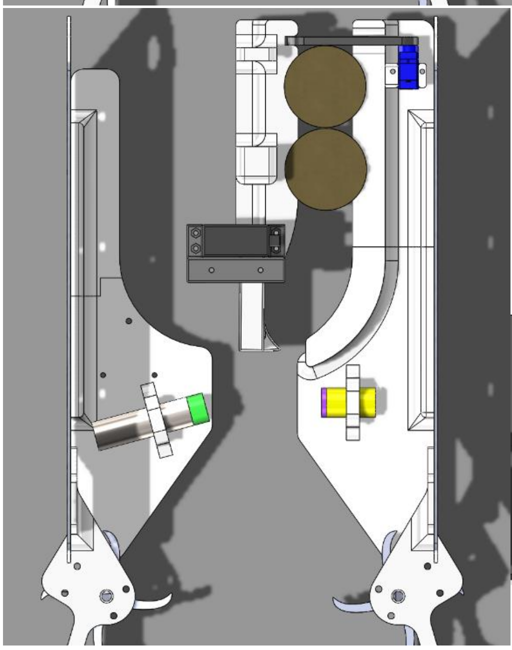
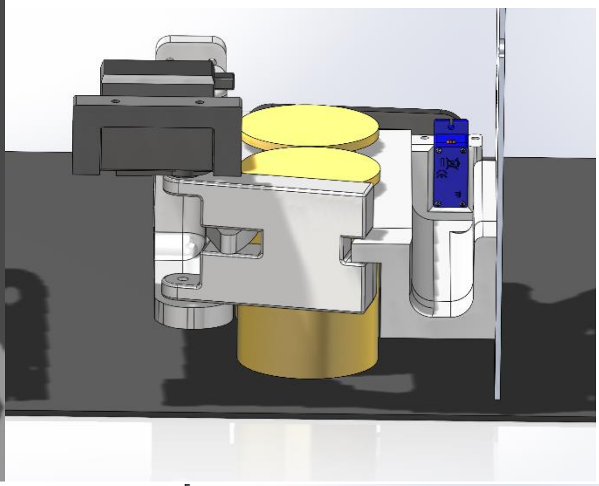
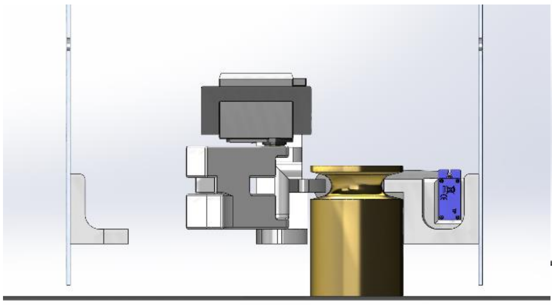
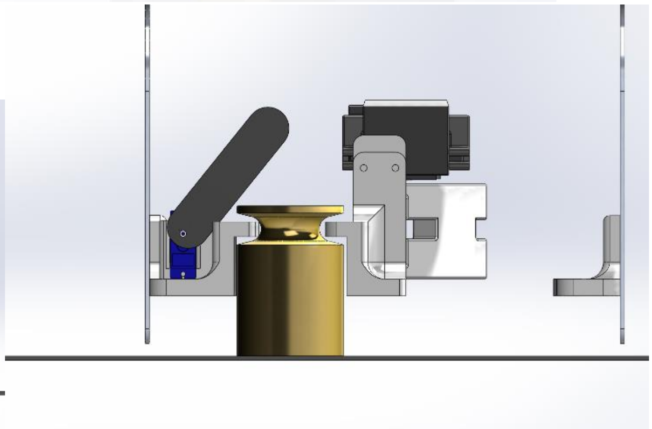
I have contributed to the design, manufacture and assembly of the weight collection system including the trays, partition, gates, servo mounts. I transferred all the designs to CAD models except the ‘combine’ wheels and brackets. I created the technical drawings for two of the 3D printed parts and assembled the entire robot in CAD. I created the circuit board brackets and contributed to the electrical layout. I contributed to the chassis, drivetrain and weight collection sections of the report, as well as the costing section and bill of materials.

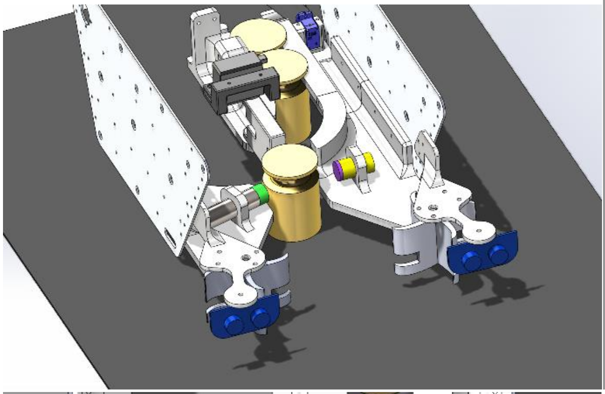
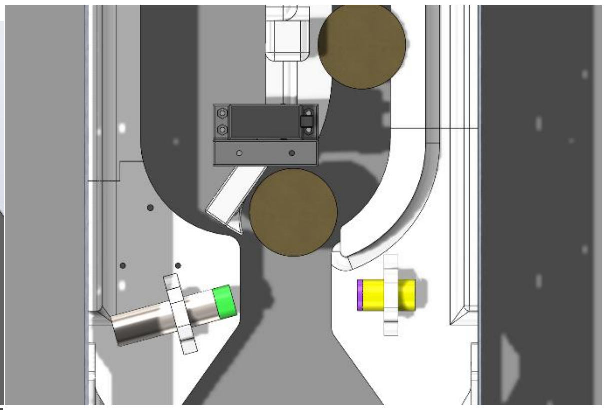
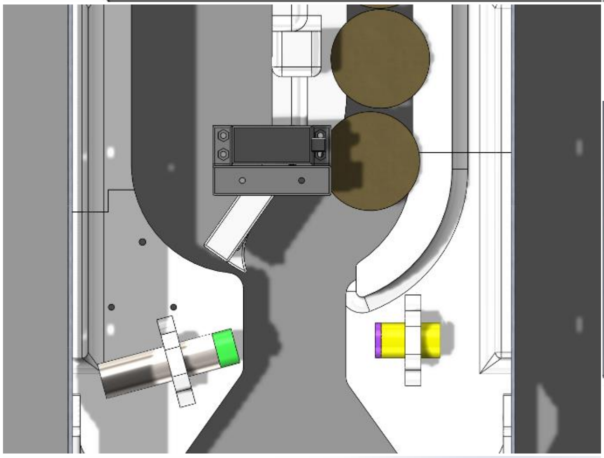
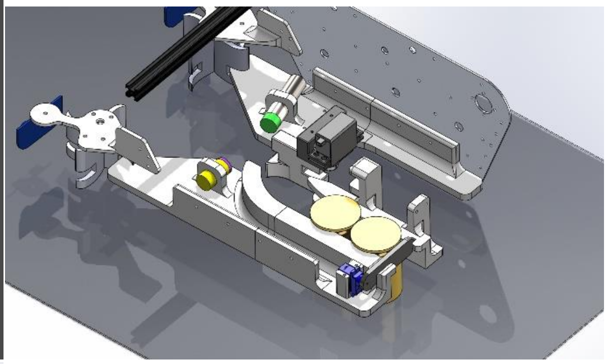
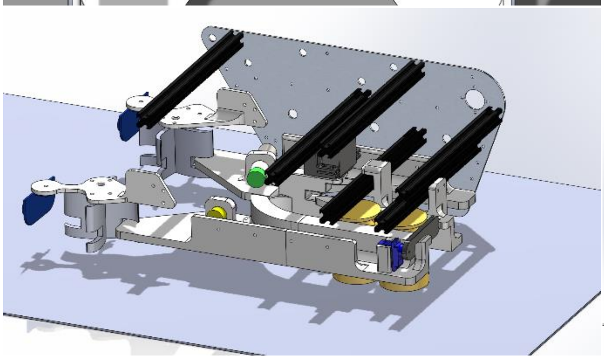
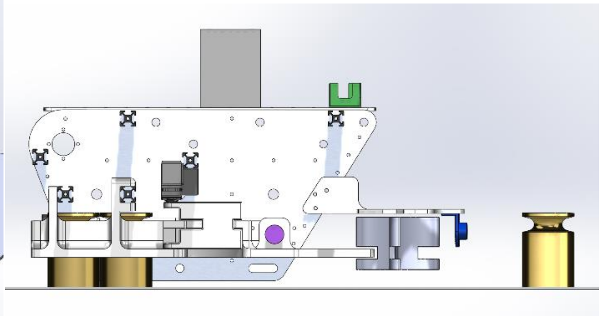
7. Appendix

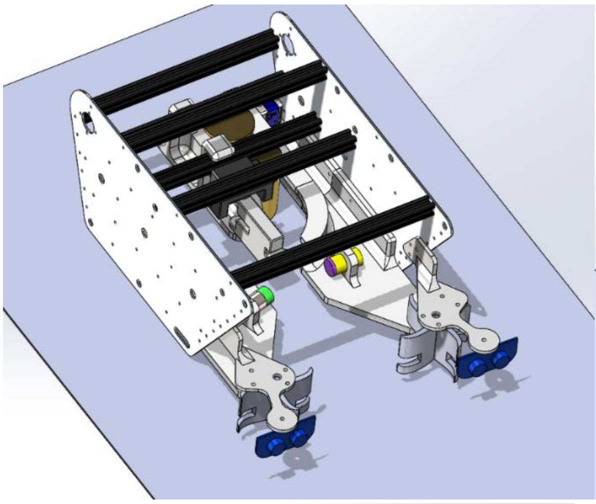
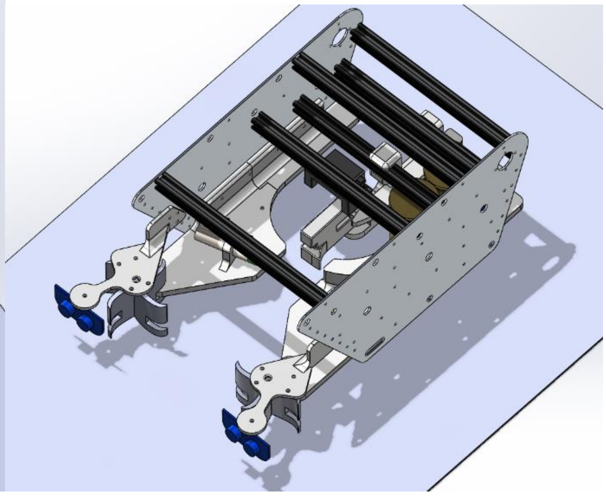
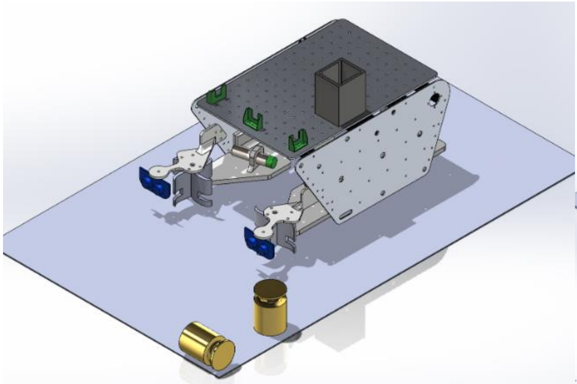
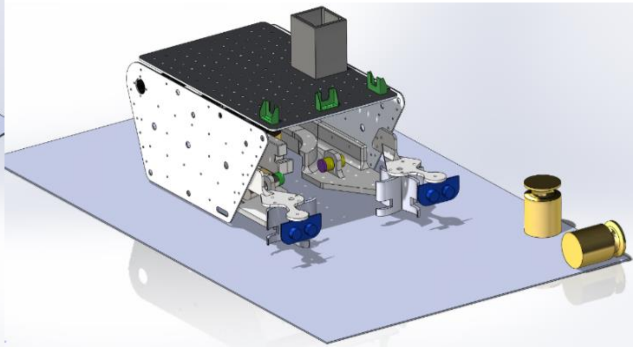
A. Bill of Materials

Part Description	Part Number	Cost per Unit ($ NZD)	Quantity
Teensy 4.0 MCU	TEENSY40	23.80	1
Chassis Plates			2
Robot tracks	880-8M		2
Drive Pulley			2
Track Bearings			10
Electrical Plate			1
Aluminium Ex. 223.5mm			6
Digital Level Shift			1
Inductive Level Shift			1
Encoder Board			1
DC Motor Driver			2
Power Supply Module			1
Power Buttons			1
Colour Sensor	TCS34725	14.00	1
Ultrasound Board		1.00	1
Ultrasound Sensors	HC-SR04	1.80	2
TOF sensor (Long)	VL53L1XV2	10.00	3
IMU Sensor	SEN0253	36.00	1
Inductive Proximity	LJ18A	25.00	1
Digital IR Sensor	SEN0019	6.00	1
Servo	HX12K	14.00	1
Servo	SG90	2.50	1
DC Motor with encoder	28PA51G	70.00	2
DC Motor 200RPM	SKU505979	15.00	2
Aluminium Billet		$11.40 per 100x100x15mm	75x45x15mm
PLA		0.05 per g	386 g

B. CAD Photos

C. Engineering Drawings (Right-hand Tray, Gate)

D. Fault Tree Analysis