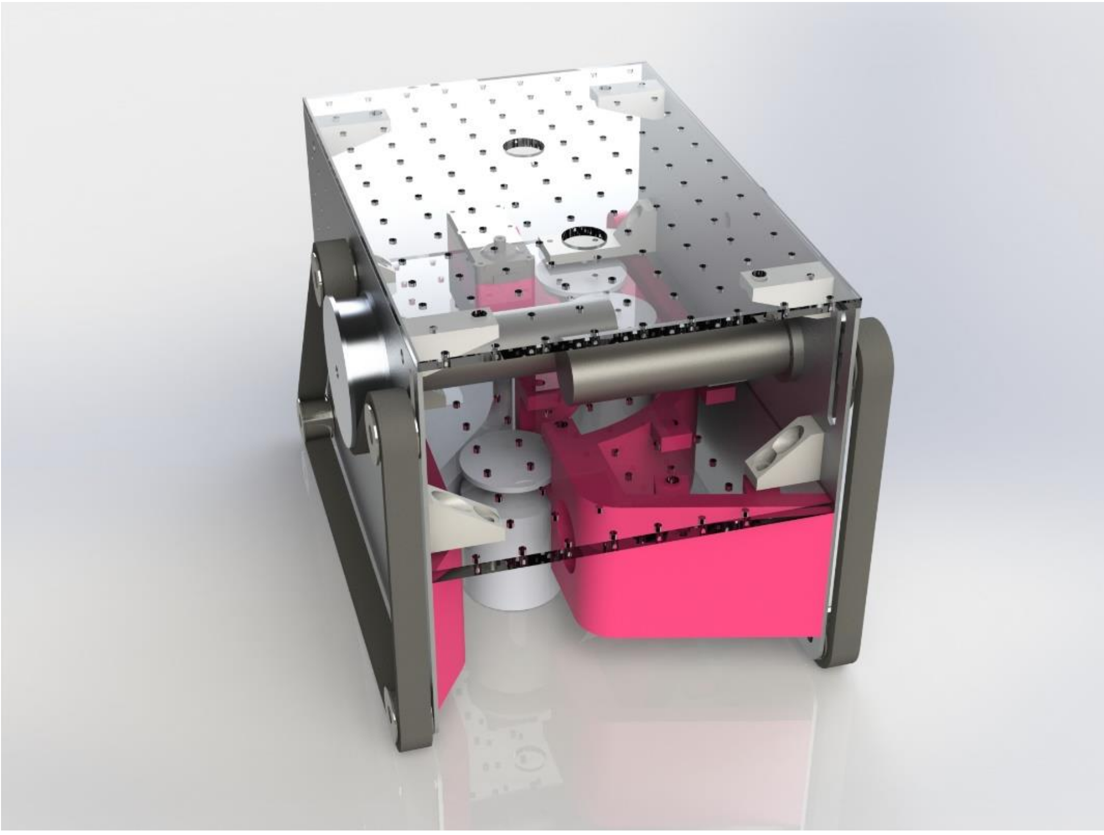
ENMT301 Group 10 Robocup Detailed Design Report



23/09/2024

Digby Eele, Eric Kleiner, Jack Edwards

Exec Summary

A sensor driven robot utilizing a teensy 4.0 was constructed to compete in a competition to pick up turned cylindrical metal weights. Rounds consisted of competing with another robot for two minutes with the goal of collecting or returning the most weights to the starting location of the match. A 50$ budget was provided for items not in a provided parts kit. Three initial design concepts were amalgamated and modified via a decision matrix to produce a final concept based on minimalism and compactness. The side walls were made from 2 mm aluminium. Plexiglass of 6 mm and 4.5 mm was used for inner mounting platforms, whilst the remaining custom parts were 3D printed from PLA.

We decided on our current collection mechanism because it can drive over fake weights and is a collection and storage all in one, making it simple to package in our small chassis. The system uses a combination of a door and a set of rails to push the weights onto for storage, lifting them off the ground.

The robot will follow a search pattern that is still being developed until a weight is detected by an array of three-time of flight sensors taking a difference in height between vertical SPAD arrays to locate weights. At this point, a PID control will navigate to the weight until detected by the inductive proximity sensor. If this sensor is triggered, the door will open to accept the target weight and complete a collection. The robot will then return to its search pattern to find a new weight if it has not reached capacity.

Our robot has achieved its speed goals and can travel at 0.34 m/s for at least 18 minutes at a theoretical max load that is much lower in practice. We still need to improve the sensor reliability to fix edge cases that cause false weight detections due to the electrical implementation of the SPAD arrays. After we have improved the sensor detection, we will focus on general navigation and chassis design, as this is the final aspect of our robot that is incomplete.

No options for sabotage have been installed, one of the options under consideration is a deployable wall. Results from testing, an FTA analysis and part drawings are included in the report and appendix.

Table of Contents

1.0 Introduction

The goal of this project is to create a robot to compete in the 2024 Robocup challenge. The competition is completed in rounds where two robots go head-to-head to collect as many target weights as possible and/or the snitch. After two minutes, the robot with the greatest score wins the round. The competition occurs within an arena containing obstacles, target weights, fake target weights and the snitch. The score is calculated based on the weight of target weights on the robot, twice the weight of any target weights returned to the robot's home base, 3 points per snitch and a quarter the weight of any fake weights is removed from your score. A robot can carry a max of 3 weights at once. The design of the robot is constrained by the competition rules, such as the robot being autonomous, running on the supplied teensy 4.0 and having a budget of $50 over the provided parts.

This report provides a detailed account of our development of a chosen design, justifying and explaining design choices. Each system will be described in its present state and its intended final state if it is incomplete.

The robot is most similar to concept 1 from our concept design report as it borrows the chassis design from this concept. We have developed a unique collection mechanism to use in combination with this chassis design. We decided to move forward with concept 1 instead of concept 2, which scored highest in our evaluation, as concept 1 was more straightforward, and we wanted to use the compact chassis of concept 1. We decided to change the collection mechanism as it was proving difficult to package the electromagnet and separate weight storage within the small chassis. In addition, we learned that fake weights with an embedded metal plate had been created. This would prevent the electromagnet from effectively isolating the real target weights for collection. Using the electromagnet would require a separate system to confirm if a target weight was real. This made the electromagnet less attractive as a solution as it lost its inherent advantage, and its disadvantages became more apparent. In testing, we found that the electromagnet required to be firmly placed flat on the top of the weight to achieve a successful collection. This would require a gimbal system to pick up the weight from an uneven surface. We did not have the space for this system.

After discussing different options, we decided to develop the current collection mechanism. This system uses a passageway for weights to pass under the robot, where a servo-actuated door would push them onto a rail if they were determined to be real by the inductive proximity sensor. This system functions as weight collection and storage as the rail is long enough to store three weights, the maximum amount that is permitted to be carried. The weights are retained at the end of the rail with another servo and arm to implement, dropping the weights at the base if deemed feasible. This system was chosen for its simplicity and ability to drive over fake weights without having to reverse removing complexity from navigation.

2.0 Design Description

The robot, as displayed in Figure 1, is designed to search for, detect, navigate to, and collect target weights during the specified competition. It uses a belt drive, a servo-actuated door with a rail system to collect and store weights, and a combination of sensors and logic to control the drive and collection system.

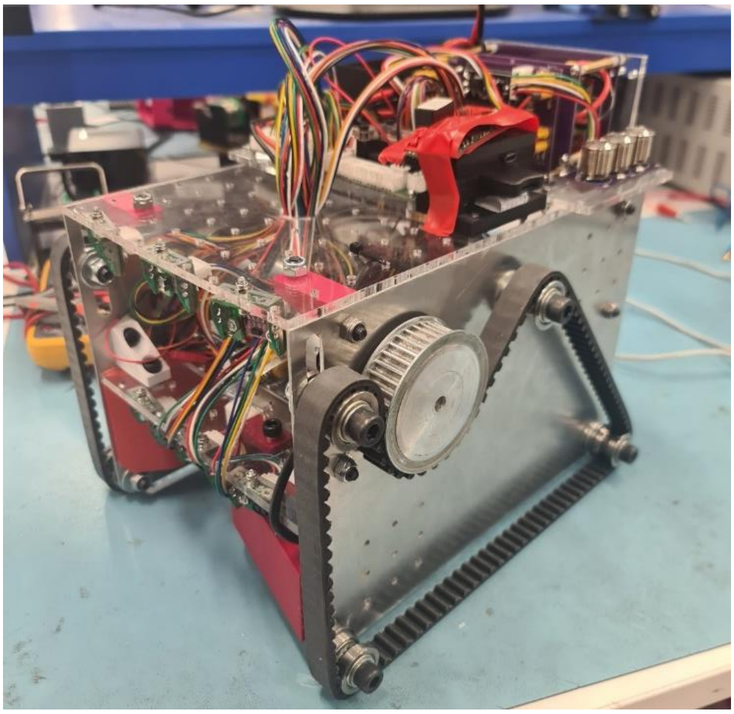


Figure 1 Robocup Robot

Extra points are awarded for returning the weights to the home base; therefore, releasing the weights has been included mechanically to be implemented if the robot is able to collect weights within the specified time and retain relative positional accuracy to make returning weights effective.

The robot's logic will be determined by the finite state machine shown in Figure 2. The robot will follow its search pattern until a weight is detected. At this point, a PID control will navigate until detected by the inductive proximity sensor. If this sensor is triggered, the door will open to accept the target weight and complete a collection. The robot will then return to its search pattern to find a new weight if it has not reached capacity.

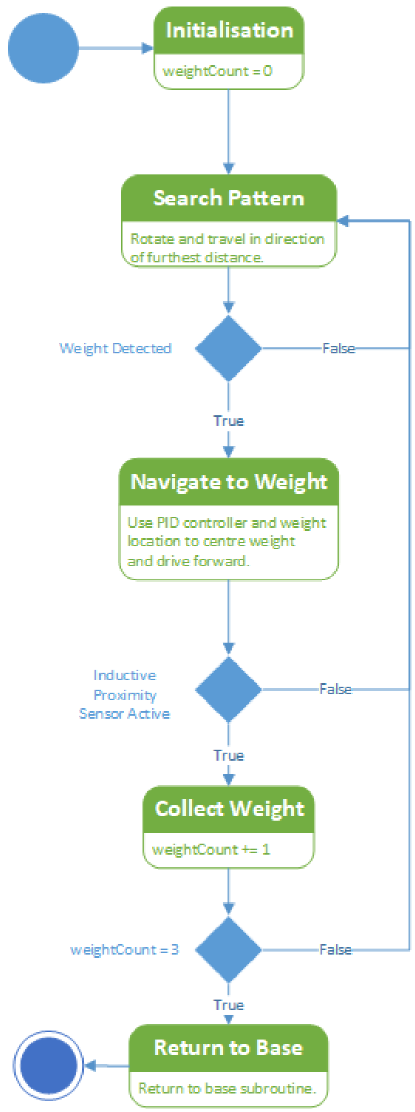


Figure 2 Finite State Machine

It has been determined that searching for target weights and arena navigation will be the most challenging aspect of this design for this reason we have attempted to design a compact robot as this will allow for less accurate navigation as the positioning accuracy required to navigate smaller spaces is proportional to the robot's dimensions.

For this reason, we decided to skip assembling and testing the supplied chassis and focus on our mechanical design. Due to the compactness of our design, all mechanical and electrical components had to be carefully packaged. This led to us being unable to start assembly until we completed the testing of our collection mechanism. After we completed the testing and received our custom side panels, we had basic driving and collection functionality within two days. Our other core design focus was simplicity. This led to the development of our current collection system, which uses a door to push the weights onto an angled rail to lift the weights off the ground. This system was chosen for its smaller size as it functions as collection and storage in one system. Additionally, it can be very consistent due to its deterministic nature the position of the weights is precisely known and controlled, leaving little possibility for jams or failed collections. The system checks target weights using the inductive proximity sensor; this is desirable as relying entirely on the electromagnet could be fooled with fake weights with embedded metal plates.

2.1 Drive Method and Chassis Design

As discussed in the overview, we opted to design a custom chassis for our robot primarily to reduce its size. To do this, the overall length is reduced, and the motors are offset by using unique side plates for each side. This allows the motors to overlap, reducing the width of the robot. This required calculating the position of the tensioners and drive pulley based on the fixed belt size supplied, which was done in SolidWorks. The side plates also include mounting for the horizontal middle and top plates. The custom side plates were waterjet cut from 2 mm aluminium plate drawings are shown in Appendix A1 and A2.

We decided on a belt drive as the parts supplied best suited this drive system; we had initially wanted to design a 3-wheel robot, but with only 2 drive motors provided, this would require using the stepper motors or other geared DC motors. The supplied DC drive motors are far superior as they produce a more useable torque and rpm while including an encoder. The disadvantage of using a belt drive is that turning inherently requires slip, which makes the encoder data far less accurate for interpolating position.

We decided to use the given drive pulley to reduce complexity in the initial drive system design. The teeth on the pulley do not match the supplied belt teeth. After some testing, it was found that the flat side of the belt produced the most grip on the arena floor and the pulley. For this reason, we decided to wrap the belt around the pulley with the flat side of the belt contacting the floor and pulley. Two belt tensioners are placed above the motor position to create a large surface for the belt to grip the pulley, as shown in Figure 3.

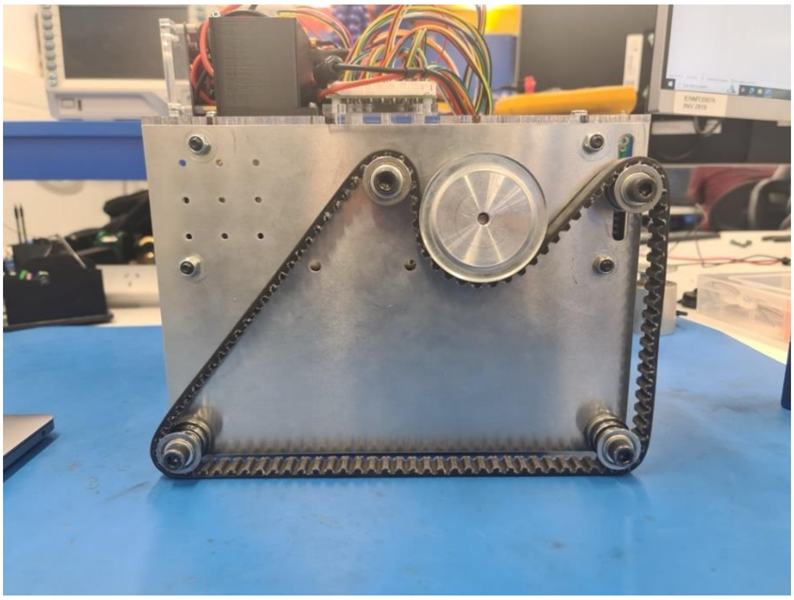


Figure 3 Custom Side Plate on Robot

The rest of the chassis is composed of 2 flat horizontal plates used for rigidity. The middle plate is laser cut from 6 mm acrylic and is positioned to secure the collection mechanism and locate sensors low down to detect weights. The rest of the space is used for mounting electrical boards. This leaves plenty of space on the top plate laser cut from 4.5 mm acrylic for the CPU, power supply, battery and navigation sensors, as shown in Figure 4.

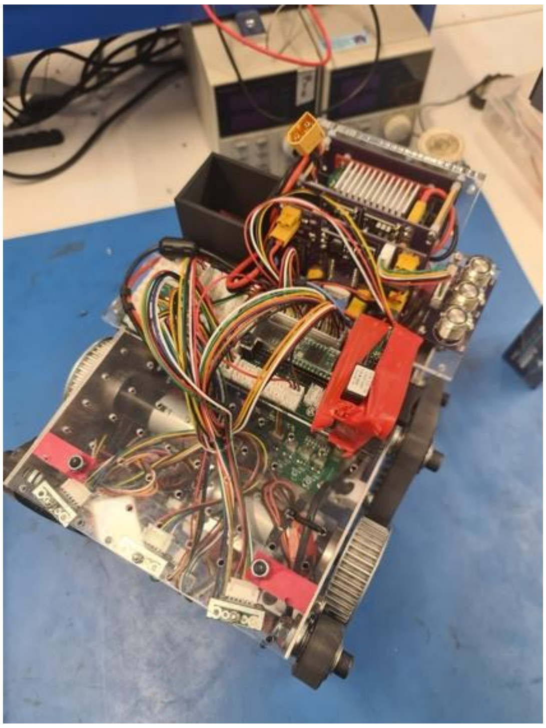


Figure 4 Top View of Robot

The implementation of the motor driver uses a PI speed controller written in C. This is common for motor drivers, as a zero error does not correspond to a zero output; therefore, the controller relies heavily on the integral term to maintain the goal speed. The speed controller is useful in ensuring consistent movement, ensuring the robot can travel in a straight line and produce repeatable movement when loaded and unloaded, independent of battery voltage. This will prevent inconstancy in navigation and provide reliable speed data.

2.2 Weight Detection and Navigation

Weight detection has been a primary focus of our testing up to this point. The time of flight or TOF sensors are highly configurable. This, in combination with their small size, has made them our first choice for weight detection. Three each of two models of TOF sensor were provided, the VL53L0X and VL53L1X. We have chosen to use the VL53L1X for weight detection due to its high range. The VL53L0X's have been placed higher for use in general navigation and are not used for weight detection. The sensor's positions are shown in Figure 5. Both models use single-photon avalanche diodes (SPAD). This allows for a smaller array to be defined to reduce and position the sensor's field of view. Testing has gone into determining the best way to use this feature; currently, we are alternating between sensing the bottom and top half of each sensor and taking the difference if it is greater than some tolerance a weight is present, its position out of the three sensors gives its location.

Initially, I developed code to alternate between all possible SPAD array locations to provide a 4 x 2 grid per senor. This allowed for a resolution of 12 horizontal positions. This additional precision was not deemed effective as it did not improve our overall field of view or improve our ability to navigate to weights after they were found. This is due to the use of a motor speed controller. When the robot turns and positions the weight in the middle sensor, it can drive forward straight enough to not require further correction to its path.

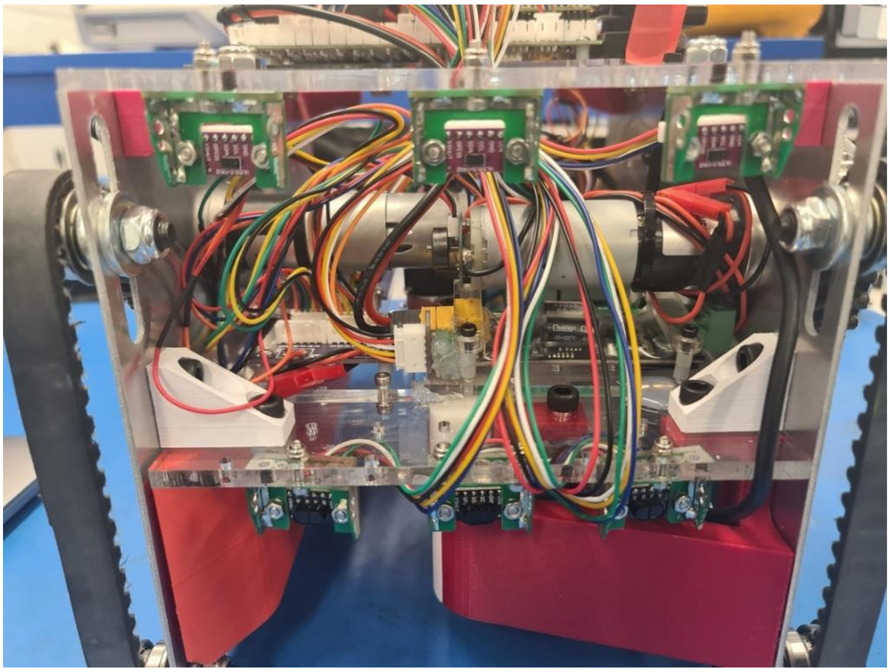


Figure 5 TOF Sensor Array VL53L1X Low for Weight Detection

Two primary issues still impact the accuracy of this system. The first is due to the implementation of the SPAD array. This array works more by focusing on a specific area in the sensor instead of completely isolating it, leading to inconsistent results. For example, when the target weight gets close, it starts to be detected by the top and bottom SPAD arrays of a sensor, which results in the weight being mistaken for an obstacle as a difference between the values it no longer measured.

The second issue is caused by the inconsistencies between SPAD arrays within a sensor. This causes issues when pointing at the edge of a box. The bottom SPAD array may detect the weight, and the top SPAD array may not. This results in a false positive when using the difference to detect the location of target weights.

To navigate towards a target weight, the data from the sensor is interpolated into an error from -2 to 2, as displayed in Table 1. This error is then run through a PID controller to adjust the relative speed of the left and right tracks to turn, centring the weight. This has proven to be effective with just proportional control the robot is able to turn to align the weight in the centre of the robot and drive forward without requiring further corrections to the path due to the use of a precise controller for the motor driving.

Table 1 Target Weight Relative Heading Error

Active Sensors	Error Value
Left	-2
Left and Middle	-1
Middle	0
Right and Middle	1
Right	2

2.3 Weight Collection and Drop-off

Once a weight has entered the robot's bounding box the weight gets checked with an inductive proximity sensor shown in Figure 6 as a green box. The weight is then found if it is target weight for collection or a dummy weight which will not be collected. Using the inductive proximity sensor means that the checking for target weights is flawless as the dummy weights will not have metallic cores at the height the sensor is checking the weights at. This means the robot can safely assume that any weight which passes the inductive proximity sensor check is a weight for collection.

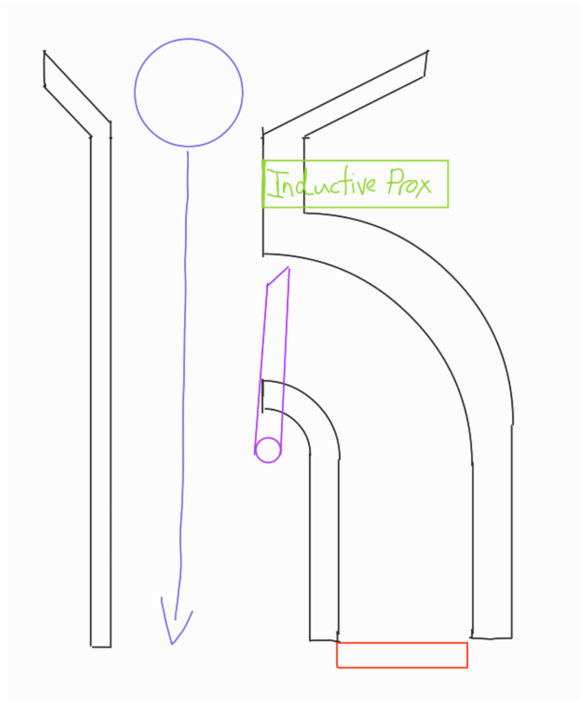


Figure 6 Non-target Weight Path Through Robot (Not to Scale)

If the weight does not pass the inductive proximity check the robot continues to drive over the weight letting the pass through the robot through the open passage shown in Figure 6, with a scaled drawing for the internal passages shown in Appendix A3. This method for discarding dummy weights leads to a very small time that the robot spends on a fake weight once it has got it within the robot by not having to reverse away from the weight and turn away.

If the weight is determined to be a target the door controlled by the large servo (RDS5160) will swing out to open another passage as shown in Figure 6. The robot will then continue to drive over the weight which will start going through the second passage which has rails along the side to lift the weight and store it within the robot. This is done by having the door slowly push the weight up the curved railing to the flat section as shown by the purple arrow in Figure 7. Once the weight is collected the door will return to the default position, shown by the purple door in Figure 6, ready to open for another weight to be collected. The weights are held in the robot from the back by another door shown as a red block in Figure 6 and Figure 7. This door is controlled by a smart servo (DRS-0101) which will only open when the robot has returned to home base. The robot will continue looking for weights until it has collected three target weights which will fill up the storage rails to the point that the weight nearly goes back down the rails and rests on the passageway door. The robot will then proceed home to drop the weights off in the base, the way the robot navigates is discussed more in 2.4 General Navigation.

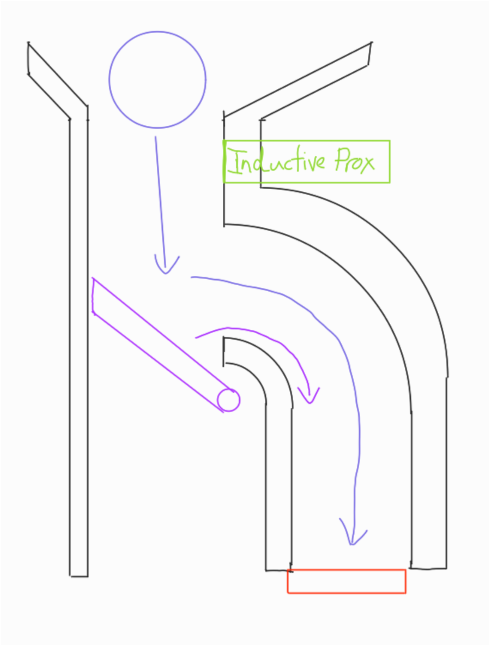


Figure 7 Target Weight Path to Robot Storage (Not to Scale)

Once the robot has returned home by finding the home base floor colour using the colour sensor (TCS34725) stored under the inductive proximity sensor. The robot will then proceed to position itself to have the backend of it facing the home base corner. Once this has been achieved the door controlled by the smart servo will open so the weights can be pushed off the rails. The door which originally separated the target and dummy weights from each other and pushed the weights onto the collection rails is then used to push the stored weights off the rails into the home base. This door was designed to be long enough to just push the third target weight off the rails.

Both doors were 3d printed to be strong enough to hold the maximum weights the robot could hold, being three 1kg weights, when angled to have the door be the only thing keeping the weights on the robot's rails.

2.4 General Navigation

As the robot is currently constructed there is only the three TOF sensors (VL53L0X) located under the top plate at the front which would be used for general navigation. These sensors are located to be symmetrical around the centre of the robot with the two side sensors being angled outwards to increase the detection of walls. The final robot will not rely on just these three TOF sensors, the further development for the general navigation is discussed more in section 5.1 General Navigation Development.

3.0 Results

3.1 Robot Speed

To ensure that the robot met its stated speed objective, a distance of three meters was measured out and marked on the floor with tape. One-meter increments were also marked. A cell phone was used as a stopwatch. The robot was commanded to move forwards for 5 trial runs. After averaging the trial runs, it was found that it took an average of 8.82 seconds to traverse the 3 meter distance. This equated to 2.94 seconds to travel one meter and gives a total travel speed of 0.34 m/s, thus meeting the speed objective.

3.2 Battery Life

Battery life is an important result to know as the robot needs to be able to run for a full round without running out of charge. This means ensuring the robot can run continuously for at least 2 minutes without running out of charge. The theoretical minimum battery life was found by taking all the components max current consumption with all the actuators being at their stalled current given by each component's datasheet. This was tabulated in Appendix E with the calculation process also specified there. The battery life was also tested by finding the peak current draw of the robot in general use by using a clamp meter to measure the current from the battery. These results for max current draw would then be used in the last equation in Appendix E to find the battery.

A peak current draw was found at 4 Ah through testing and was calculated to use 12.9 Ah by summing the current components max power consumption as specified by their datasheets. This found a tested result of the robot lasting for 1 hour of more realistic use and 18 minutes and 37 seconds from the max current on datasheet theory, as shown in Table 2, which well over the required 2-minute round even at peak consumption with everything going at once which would not occur within a real match. This means the battery consumption will not be a problem for lasting a match as off one full battery charge the robot can compete in 30 rounds before it would run out of charge at peak power consumption from testing and would last 9 rounds with everything at max power consumption which would not happen during a real round.

Table 2 Current Draw and Battery Life results

Max Current draw	Battery Life
4 A	1 Hour
12.9 A	18 minutes 37 seconds

3.3 Weight Detection Range and Reliability

Scoring within the competition requires collecting weights to achieve this first the weights must be detected. For the detection of weights, we are taking the difference between the upper and lower section of three time of flight sensors (VL53L1X). The accuracy and range of this technique is tested inside of the area using the supplied target weights.

Initial testing of the TOF sensors involved using them in long-range mode. This mode can provide up to 4 metres of range but only under ideal conditions, such as darkness. Due to the arena's high ambient light, we decided to use short-range mode, as the datasheet specified it would produce the best range in high ambient light, as shown in Table 3.

Table 3 VL53L1X Distance Mode Data (From Datasheet)

Distance Mode	Max Distance (Dark) [m]	Max Distance (Bright) [m]
Short	1.36	1.35
Medium	2.9	0.76
Long	3.6	0.73

A weight is placed in front of the robot to test the range, and the result of the sensor array is observed. The weight is then moved from across the front of the robot at a fixed distance to confirm there are no blind spots and that all sensors can detect the weight. If this is successful, the weight is moved back, and the process is repeated. With the current configuration, the robot stopped detecting the target weight at 1.34 metres.

Testing the accuracy of the sensor array was completed by manually driving the robot around the arena and recording any false weight detections. False detections with the current setup occur when encountering the corner of an obstacle, such as a box, as discussed in section 2.2. Improvements for this will be discussed, but within a 2-minute round of constant driving, three false detections occurred.

The inductive proximity sensor was also tested and was 100% accurate over 25 tests against target and fake weights.

4.0 Fault Tree Analysis

The fault tree in Appendix B was designed and modified with two modes of focus: the layout of various system domains, and the major objective failures that the robot could fail. The first page of the fault tree displays the two path options for fault analysis. The first breaks down into a full multi domain system overview. The second option was organised by specific major objective failures. This second option was broken up into flags for quick tracing. The flags were all colour coded, identified and broken down by system domain in the remaining pages.

The tree was created by first mapping out by each major system domain using the initial concept that the team decided on. It was then backfilled with flags for easier tracing of specific objective failures. Should this be used as a quick reference diagnostic tool, the prompts at the top of the page allow the reader to quickly reference the tree for troubleshooting routs. Since refinements and restructuring to the software code are still being made, the software part of the tree includes underdeveloped events to allow for future clarification. This permits for the expansion of the tree as the code approaches it final form.

Due to time constraints and other project demands during development, probability weights and failure timing occurrences were not incorporated for end events. This could be integrated into a future edition of the FTA should the need arise for a more rigorous analytical approach to failure modes.

The tree has been a useful tool for mapping out issues, and possible improvements to the robot's physical components, as well as the development of software code. A Wi-Fi module was used for debugging motor control and was included in the software fault tree. It will be removed for the final competition. An example would the mapping out the possibility of a software driven adaptation, via an inhibit gate in the FTA should damage or malfunction occur. The tree has evolved into a self-refining diagnostic tool

5.0Evaluation / Further Development

5.1 General Navigation Development

General Navigation is currently not properly implemented as the robot is currently constructed this will be the main part to be further developed as the three TOF sensors (VL53L0X) will not be enough to have a fully autonomous robot. To fix this more distance sensors will be added on top of the robot along the sides of the robot, meaning the robot can figure out where walls are in different directions. This will hopefully lead to the robot being able to drive around the arena without running into walls autonomously.

The robot in its current state only has autonomous driving when it is located and navigating to a weight as discussed in section 2.2. An algorithm will be required to search the area until a weight has been found to do this more sensors will be required at present the robot is only able to detect walls 80 degrees and 1.34 metres in front of the robot, a common strategy for searching for weights is to follow along the walls of the area to do this side sensors would be required such as the ultrasonic sensors. This would allow keeping a constant distance from the wall and driving around until a corner is met and turning away from the wall it is following. Another possible technique is moving in the direction of this the furthest distance. This would cause the robot to always be making progress into further areas. This could increase the chance of locating a weight by travels the area. We have decided not to focus on locating the robot within the area as this will be a very difficult task without a reliable encoder source as the tracks require slip to turn inaccuracy's will be introduced at every turn. This encoder data could prove useful if fused with IMU data this could be implemented using a Kalman filter.

5.2 Chassis Design

The current chassis design has been found to have an issue which was not considered when designing it. The issue with the current chassis design is when the track is fully tensioned the chassis will bend at the corners leading to the robot wobbling. This can be solved by either lessening the tension or redesigning the internal passageway so it can mount to the chassis walls which will get rid of the bending. The issue with redesigning the internal passageway is that three of the corners would need complete redesign whereas lessening the tension introduces a new but easier to solve problem. When the tension gets lessened to robot will bottom out when it tries to go over bumps as the belt does not have the tension to support the weight of the robot, this can be fixed by adding more track supporting hardware parts which will decrease to distance that the track has no support. This can also be solved by making the bottom of the robot have curves which will follow just above where the track will go. Another issue found with lessening the belt tension was slipping of the belt which can be solved by designing a custom driving gear which will fit the belt teeth in the gear reducing the chance of slipping as the current driving gear does not fit the belt teeth.

5.3 Potential Sabotage Method

Currently the robot has no method to try sabotage the opponent's robot. An idea that was thought of was to create side guards for the robot, which if it found a small gap between two walls within the arena could drop off these side guards in an attempt to block the opponent's robot from a section of the arena. This would be done once the robot was returning home to drop off weights as was considered that the robot would not have enough time to collect three target weights, drop them off at home and go to collect more reliably before the two-minute round was up. This would hopefully cause major issues for the opponent's robot as it would either not be able to get past it as it would not run into the dropped side guard as it looked like a wall or would run into the side guard and get stuck on the side guard as it would have stabilising feet at the bottom. Due to time constraints and other more important further development needed for the robot this would likely not be developed for the competition but was a plausible idea at a non-damaging sabotage against the opponent's robot.

5.4 Weight Detection

Improvements need to be made to the weight detection to avoid the false positives that are occurring, as discussed in section 3.3. To achieve this, we have discussed multiple solutions, such as using separate sensors instead of the SPAD array, this would allow the vertical sections to be aligned manually. This poses some challenges as we only have three long-range TOF sensors, so it would require using some of our budgets to get three more or mixing the short and long-range TOF sensors. This would reduce the range of weight detection. Another possible solution is to increase the filtering by using the top navigation sensors to detect the edge case where a target weight is detected when looking at the edge of a box and reject these results. In addition to fixing these issues, the sensors require remounting to avoid detection of the floor or top of a target weight when it's too close to the top SPAD array.

6.0 Contribution Statement

Digby Eele:

My contribution to the robot's mechanical systems includes completing almost all assembly, testing and design. This consists of the chassis, drive system, and collection system, which I designed, produced, fitted, and tested the following parts:

Left and right custom side plates

Belt and motor (supplied parts only assembled and tested)

Top and middle plates

Right angle brackets to attach the side and top plates

the collection passageway and door

I completed all the electrical work for the robot, including a custom wifi to serial bridge for remote control and serial output during testing. I wired all the sensors, servos, motors and relevant drive boards back to the CPU.

I have completed all the software development, testing and tuning for all modules this included:

Task scheduler

PI motor driver

SPAD Sensor array driver

Sensor Filtering

Weight navigation PID controller

Weight collection

Serial to WIFI bridge and remote control support

All this code was/is available on a git repository throughout development.

I have completed the testing for all the modules I made. Most of this time has been spent on sensor testing; I completed 12 iterations of different sensor configurations. I assisted Jack in the current measurement for the battery life testing and assisted Eric in completing the speed tests.

For the report I completed the following sections individually:

1.0 Introduction

2.0 Design Description

2.1 Drive Method and Chassis Design

2.2 Weight Detection and Navigation

3.3 Weight Detection Range and Reliability

5.4 Weight Detection

Appendix A Solidworks Drawings

The following sections were completed with another group member:

Executive Summary

5.1 General Navigation Development

I also helped with general report structure and formatting.

Eric Kleiner:

Report: Executive Summary 3.1, 4.0, Appendix B

Due to medical circumstances outside of university, my contribution to the project was reduced. I helped with initial concept design contributions and options for potential sabotage by trying to provide out of the box thinking. I also helped research and produce CAD models for frame and servo mounting brackets, sensors, and did some modifications to an iteration of the main pusher door. I drafted the fault tree analysis included in this report and contributed to the speed test runs.

Jack Edwards:

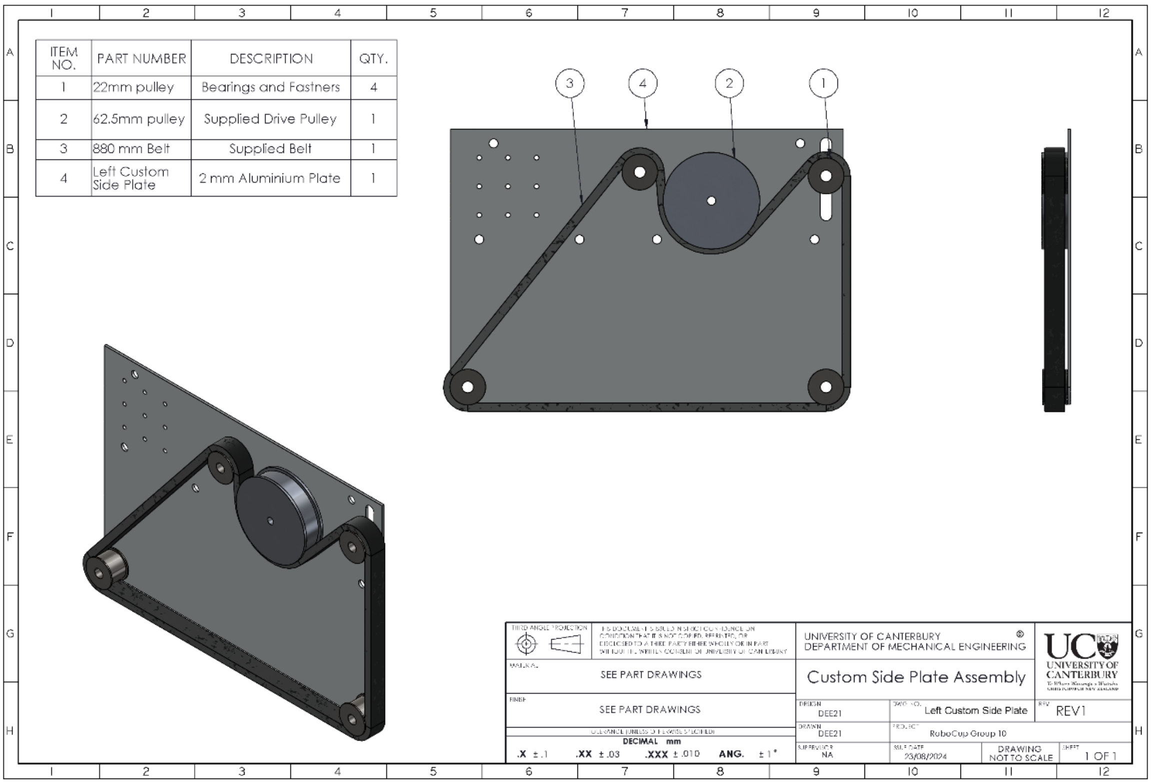
My contribution to the robot includes helping figure out how the robot would work and what robot design we will go ahead with after completing the CDR. I helped design the collection rails in the passage and did testing to see if they work with actual weights. I designed the mount for the smart servo and the gate door to hold the weights on the rails. I also designed the battery holder on top of the robot and modified the front guide piece to help guide the weights into the robot's internal passages. I also looked into the further development of the robot looking into possible sabotage methods. I did the battery life testing and wrote up and checked the bill of materials was accurate to the robot current construction.

For this report I set up the overall for format that report has used, and I wrote sections 2.3, 2.4, 3.2, 5.2, 5.3, Appendix D, E and helped write sections 5.1 with other team members.

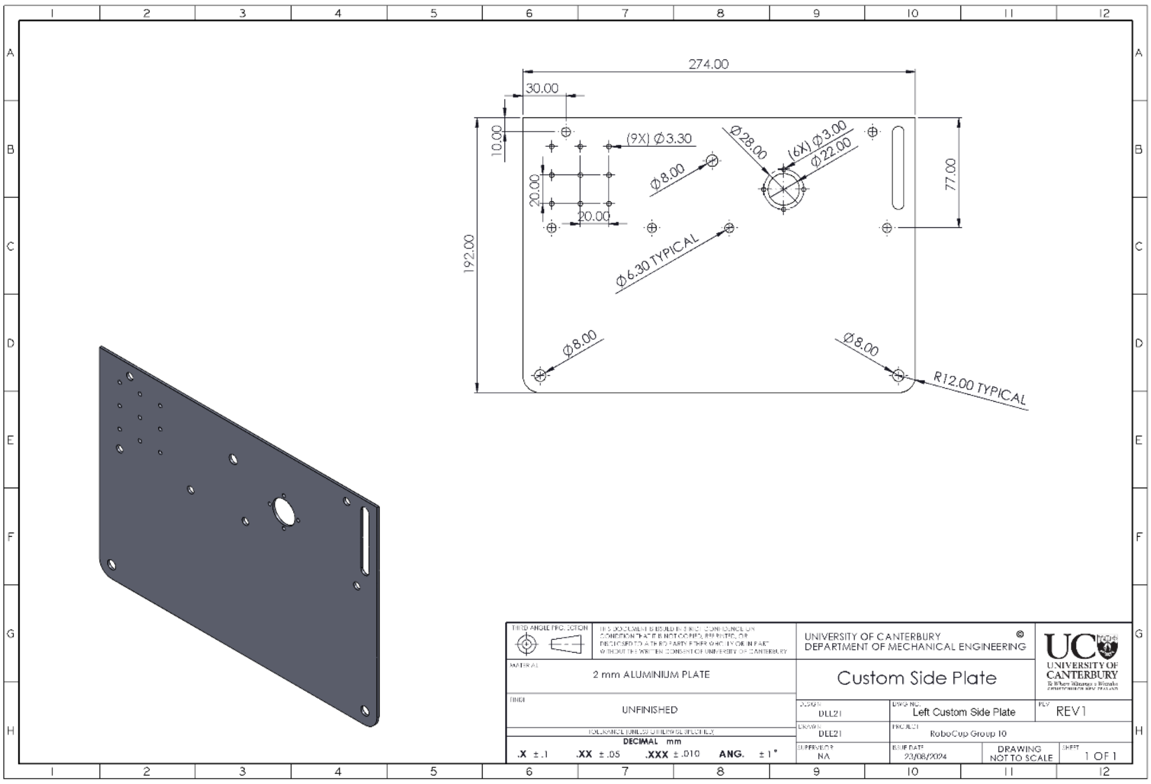
Appendices

Appendix A – Design Drawings

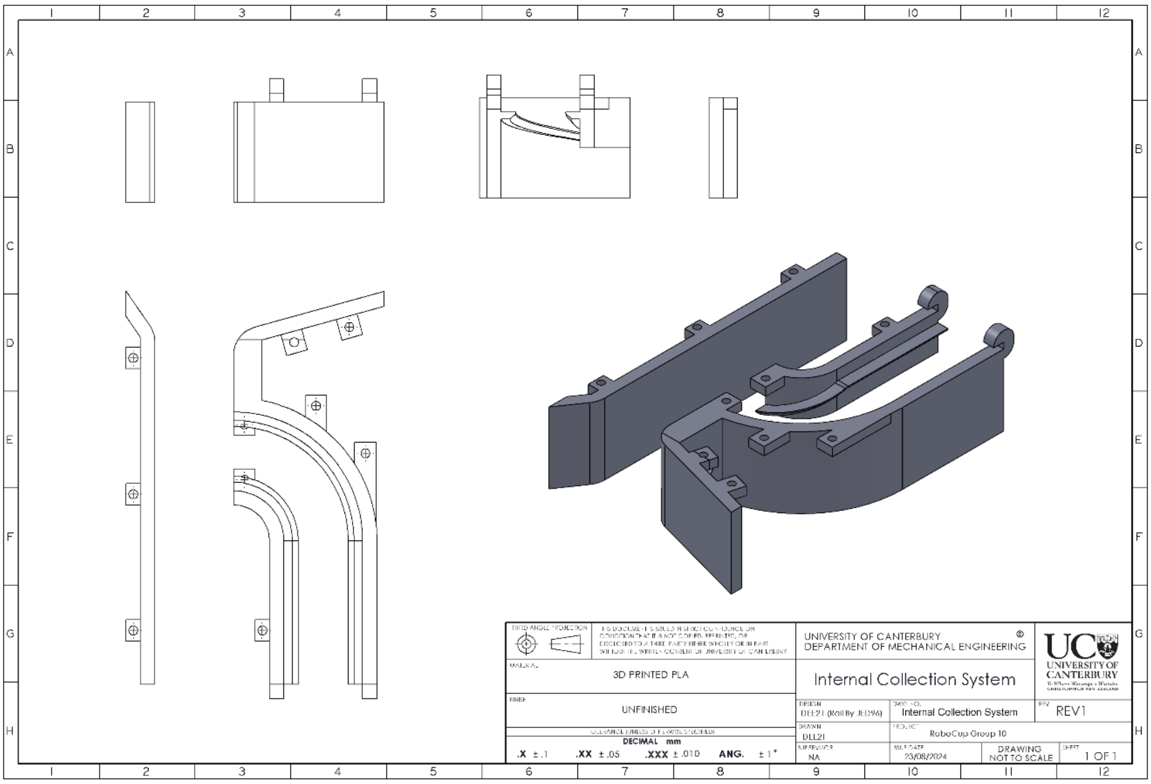
A1 Side Plate Assembly



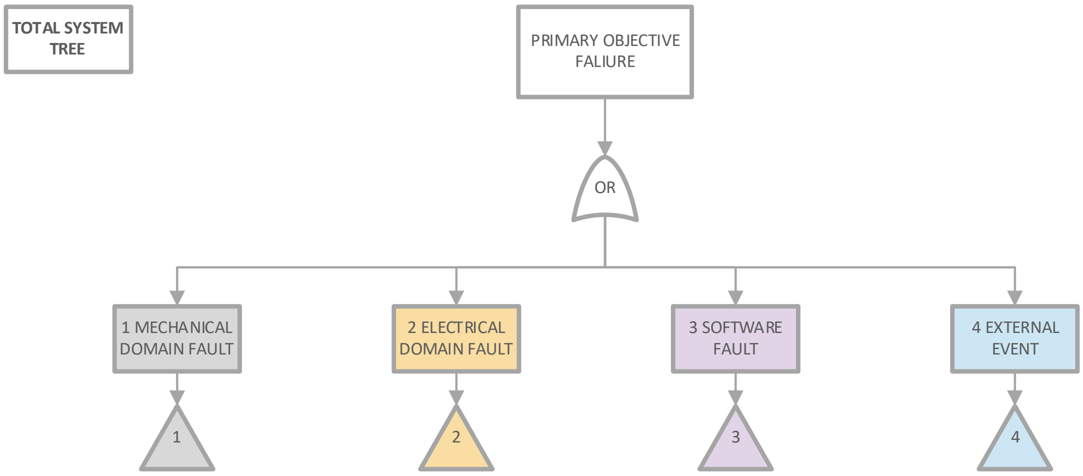
A2 Side Plate Part

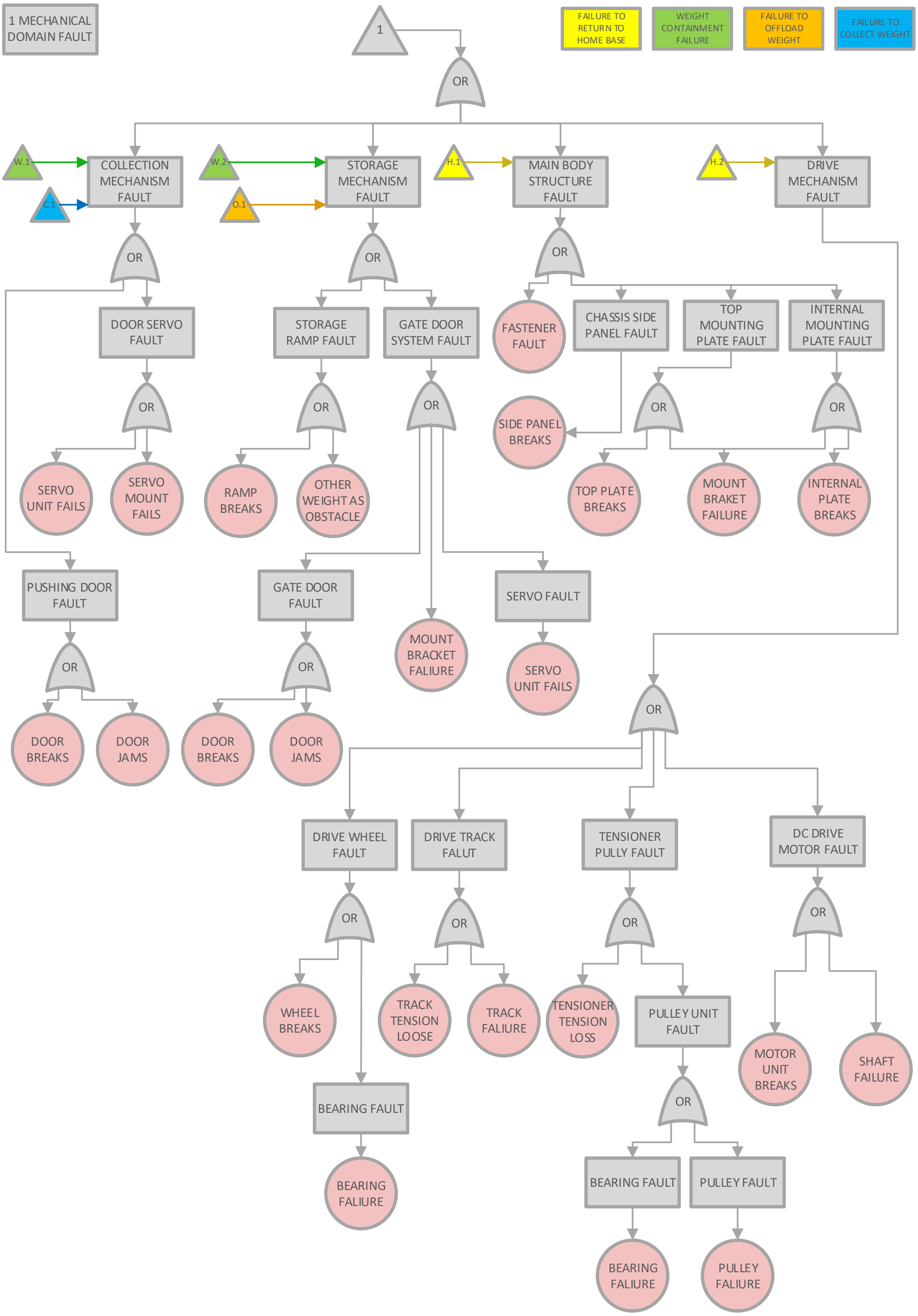


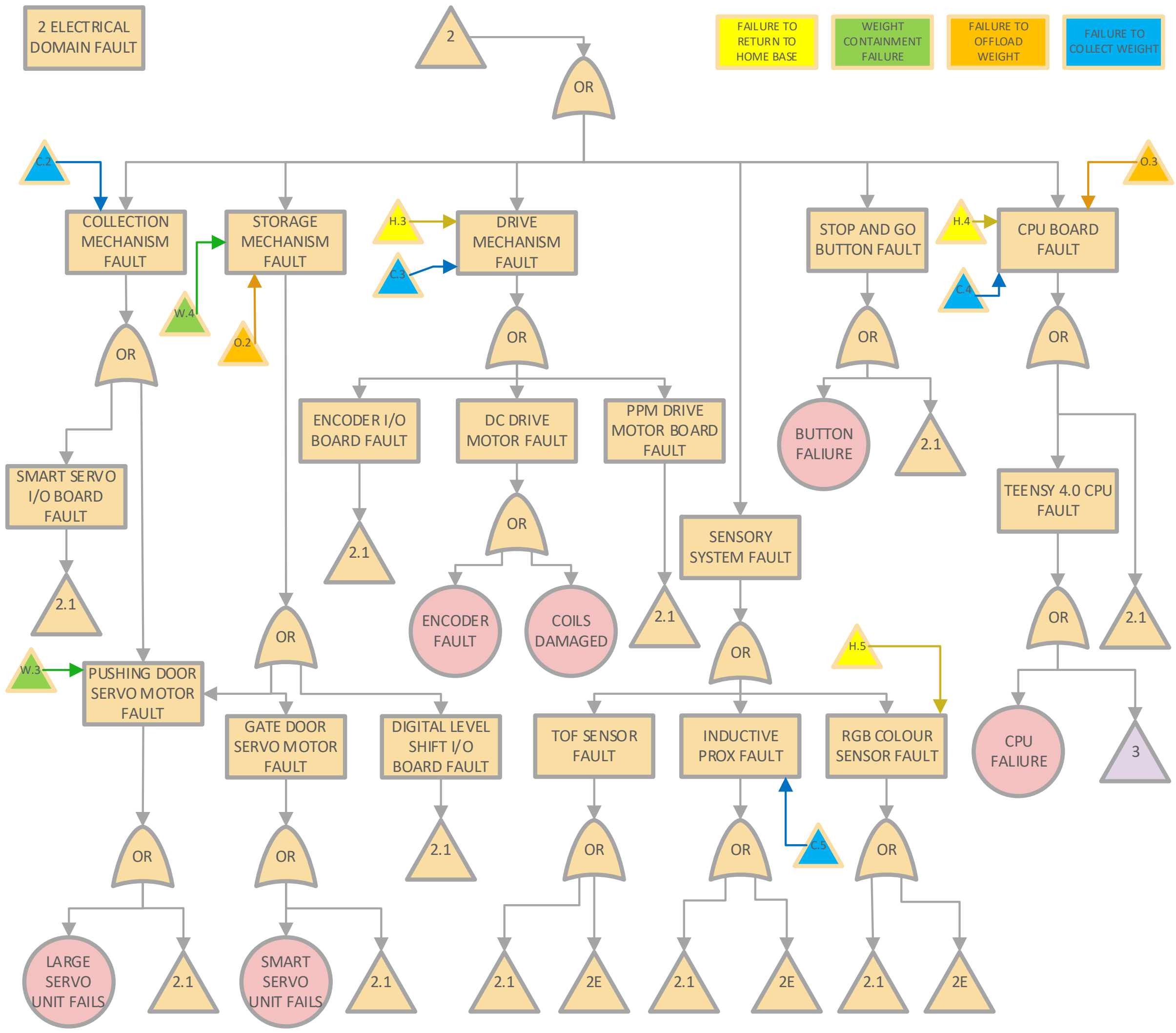
A3 3D Printed Collection System

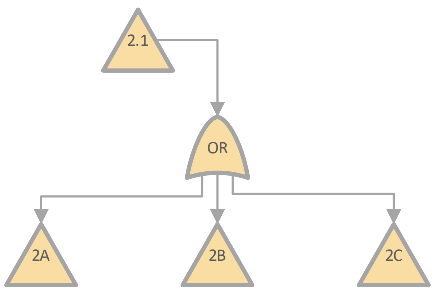


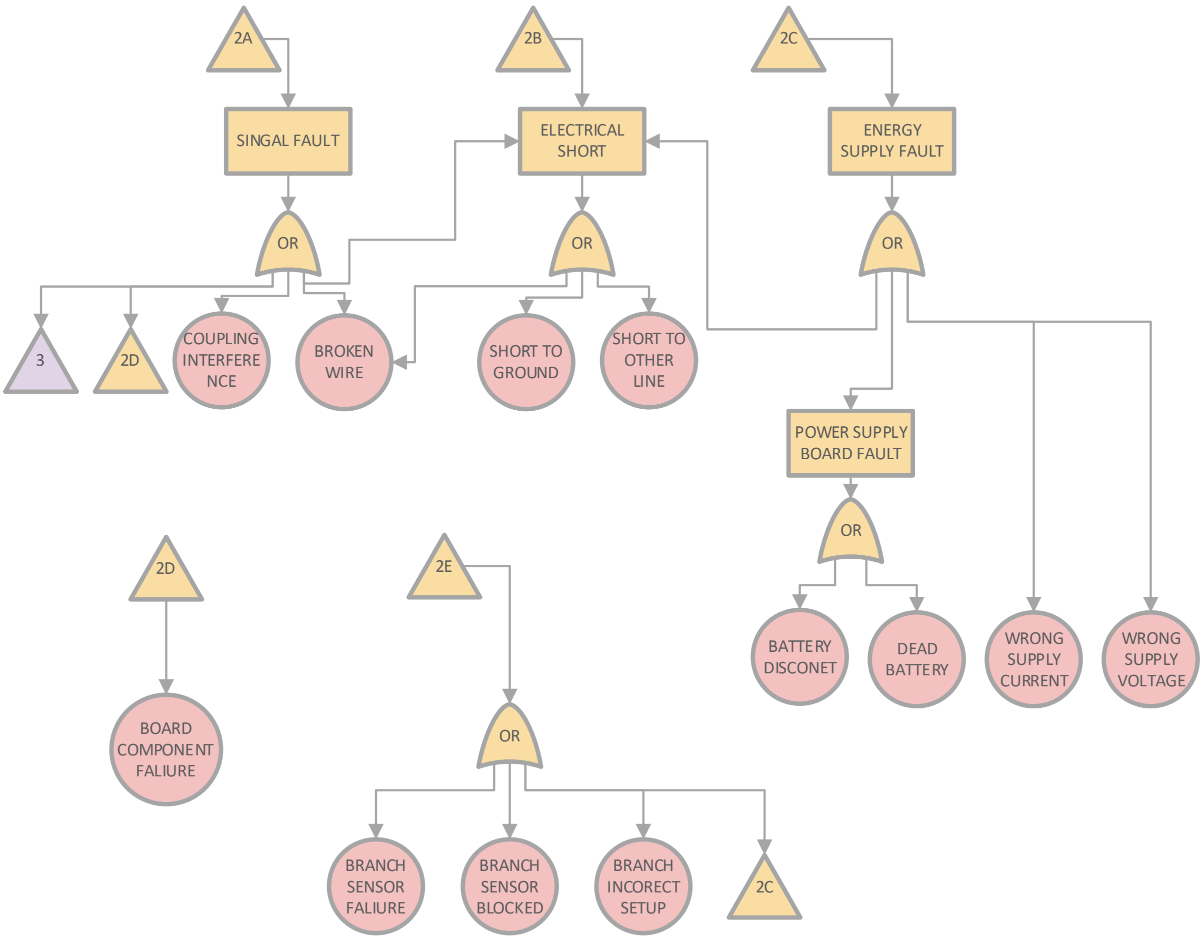
Appendix B – Fault Tree Analysis



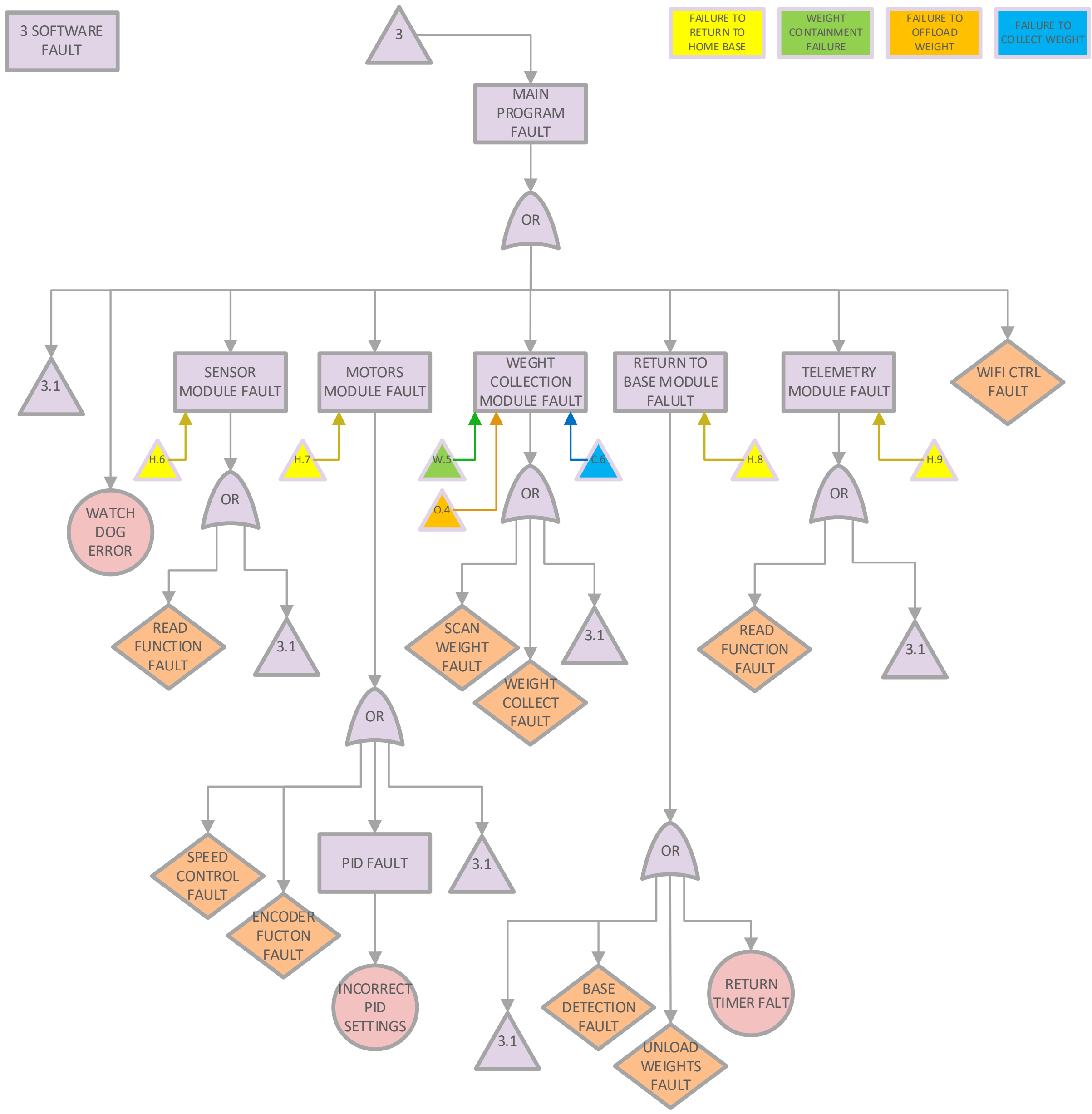

1 MECHANICAL DOMAIN FAULT
1
FAILURE TO RETURN TO HOME BASE
WEIGHT CONTAINMENT FAILURE
FAILURE TO OFFLOAD WEIGHT
FAILURE TO COLLECT WEIGHT
OR
W.1
C.1
COLLECTION MECHANISM FAULT
W.2
O.1
STORAGE MECHANISM FAULT
H.1
MAIN BODY STRUCTURE FAULT
H.2
DRIVE MECHANISM FAULT
DOOR SERVO FAULT
STORAGE RAMP FAULT
GATE DOOR SYSTEM FAULT
FASTENER FAULT
CHASSIS SIDE PANEL FAULT
TOP MOUNTING PLATE FAULT
INTERNAL MOUNTING PLATE FAULT
SIDE PANEL BREAKS
SERVO UNIT FAILS
SERVO MOUNT FAILS
RAMP BREAKS
OTHER WEIGHT AS OBSTACLE
TOP PLATE BREAKS
MOUNT BRAKET FAILURE
INTERNAL PLATE BREAKS
PUSHING DOOR FAULT
GATE DOOR FAULT
SERVO FAULT
MOUNT BRACKET FALIURE
SERVO UNIT FAILS
DOOR BREAKS
DOOR JAMS
DOOR BREAKS
DOOR JAMS
DRIVE WHEEL FAULT
DRIVE TRACK FALUT
TENSIONER PULLY FAULT
DC DRIVE MOTOR FAULT
WHEEL BREAKS
TRACK TENSION LOOSE
TRACK FALIURE
TENSIONER TENSION LOSS
PULLEY UNIT FAULT
MOTOR UNIT BREAKS
SHAFT FAILURE
BEARING FAULT
BEARING FALIURE
BEARING FAULT
PULLEY FAULT
BEARING FALIURE
PULLEY FALIURE

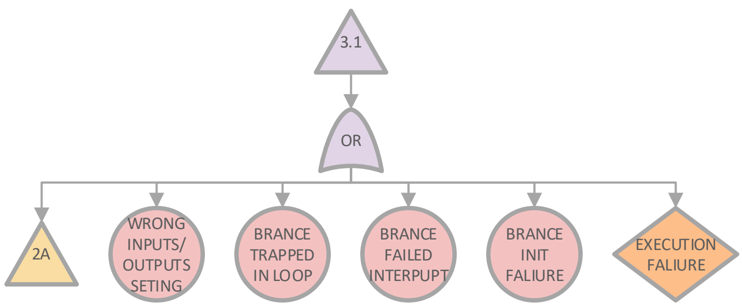

2 ELECTRICAL DOMAIN FAULT
2
OR
FAILURE TO RETURN TO HOME BASE
WEIGHT CONTAINMENT FAILURE
FAILURE TO OFFLOAD WEIGHT
FAILURE TO COLLECT WEIGHT
C.2
COLLECTION MECHANISM FAULT
STORAGE MECHANISM FAULT
H.3
C.3
DRIVE MECHANISM FAULT
STOP AND GO BUTTON FAULT
H.4
C.4
CPU BOARD FAULT
O.3
W.4
O.2
SMART SERVO I/O BOARD FAULT
ENCODER I/O BOARD FAULT
DC DRIVE MOTOR FAULT
PPM DRIVE MOTOR BOARD FAULT
BUTTON FALIURE
TEENSY 4.0 CPU FAULT
SENSORY SYSTEM FAULT
ENCODER FAULT
COILS DAMAGED
W.3
PUSHING DOOR SERVO MOTOR FAULT
H.5
GATE DOOR SERVO MOTOR FAULT
DIGITAL LEVEL SHIFT I/O BOARD FAULT
TOF SENSOR FAULT
INDUCTIVE PROX FAULT
RGB COLOUR SENSOR FAULT
CPU FALIURE
3
C.5
LARGE SERVO UNIT FAILS
SMART SERVO UNIT FAILS
2.1
2E

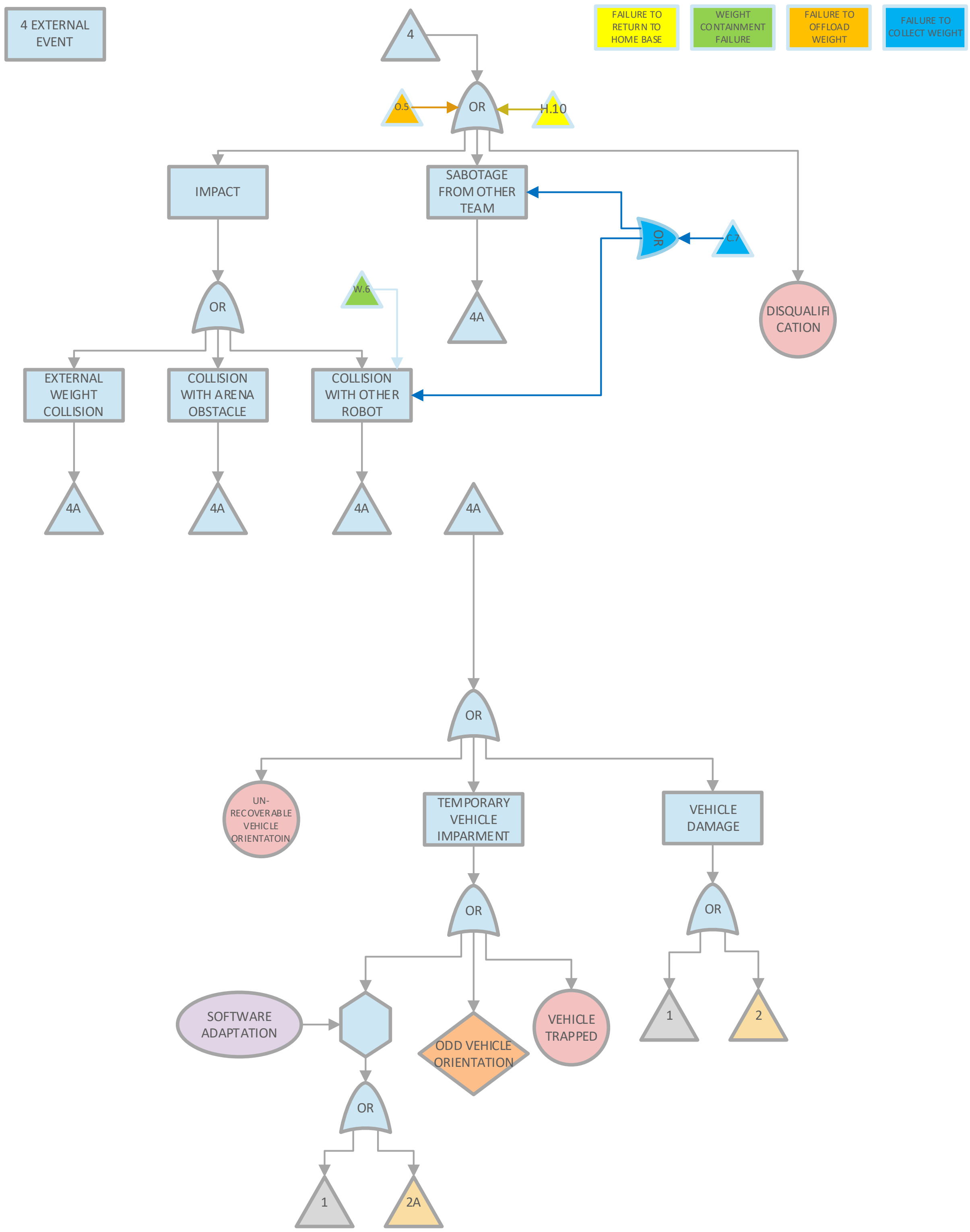

2.1
OR
2A
2B
2C



2A
SINGAL FAULT
OR
3
2D
COUPLING INTERFERE NCE
BROKEN WIRE
2B
ELECTRICAL SHORT
OR
SHORT TO GROUND
SHORT TO OTHER LINE
2C
ENERGY SUPPLY FAULT
OR
POWER SUPPLY BOARD FAULT
OR
BATTERY DISCONET
DEAD BATTERY
WRONG SUPPLY CURRENT
WRONG SUPPLY VOLTAGE
2D
BOARD COMPONENT FALIURE
2E
OR
BRANCH SENSOR FALIURE
BRANCH SENSOR BLOCKED
BRANCH INCORECT SETUP
2C


3 SOFTWARE FAULT
3
FAILURE TO RETURN TO HOME BASE
WEIGHT CONTAINMENT FAILURE
FAILURE TO OFFLOAD WEIGHT
FAILURE TO COLLECT WEIGHT
MAIN PROGRAM FAULT
OR
3.1
SENSOR MODULE FAULT
MOTORS MODULE FAULT
WEGHT COLLECTION MODULE FAULT
RETURN TO BASE MODULE FALULT
TELEMETRY MODULE FAULT
WIFI CTRL FAULT
H.6
H.7
W.5
O.4
C.6
H.8
H.9
WATCH DOG ERROR
OR
OR
OR
READ FUNCTION FAULT
3.1
SCAN WEIGHT FAULT
WEIGHT COLLECT FAULT
3.1
READ FUNCTION FAULT
3.1
OR
SPEED CONTROL FAULT
ENCODER FUCTON FAULT
PID FAULT
3.1
INCORRECT PID SETTINGS
OR
3.1
BASE DETECTION FAULT
UNLOAD WEIGHTS FAULT
RETURN TIMER FALT


3.1
OR
2A
WRONG INPUTS/ OUTPUTS SETING
BRANCE TRAPPED IN LOOP
BRANCE FAILED INTERPUPT
BRANCE INIT FALIURE
EXECUTION FALIURE



4 EXTERNAL EVENT
FAILURE TO RETURN TO HOME BASE
WEIGHT CONTAINMENT FAILURE
FAILURE TO OFFLOAD WEIGHT
FAILURE TO COLLECT WEIGHT
4
OR
O.5
H.10
IMPACT
SABOTAGE FROM OTHER TEAM
OR
C.7
DISQUALIFI CATION
OR
W.6
4A
EXTERNAL WEIGHT COLLISION
COLLISION WITH ARENA OBSTACLE
COLLISION WITH OTHER ROBOT
4A
4A
4A
4A
OR
UN-RECOVERABLE VEHICLE ORIENTATOIN
TEMPORARY VEHICLE IMPARMENT
VEHICLE DAMAGE
OR
OR
SOFTWARE ADAPTATION
ODD VEHICLE ORIENTATION
VEHICLE TRAPPED
1
2
OR
1
2A

Appendix C – Design Requirements

1. **TARGET CAPTURE AND MANAGEMENT**
    1.1. The robot shall collect assorted 0.5 kg, 0.75 kg, and 1 kg target weights.
        1.1.1. The robot should be aware of the quantity of targets weight onboard carrying at most three at once.
        1.1.2. The robot should identify metal target weights as opposed to plastic dummy weights to at least 80% accuracy.
    1.2. The robot should not be negatively impacted by catching the snitch.
    1.3. Nor should it actively attempt to catch the snitch.
    1.4. Up to 3 weights will be stored securely to prevent loss during movement.
    1.5. The storage mechanism shall allow weights to be removed from the robot for delivery to the home base.
    1.6. The robot should be aware of its location in the area.
        1.6.1. To distinguish its home base for weight deposit.
        1.6.2. To avoid picking up target weights from its own home base or opponent's base.
2. **IDENTIFICATION AND NAVIGATION**
    2.1. The robot shall effectively identify and navigate.
        2.1.1. Around walls and pipes as per rule specifications.
        2.1.2. Over speed bumps, ramps and home base rim as per rule specifications.
        2.1.3. Around opponent robots.
        2.1.4. Towards upright weights target or dummy.
    2.2. The robot should return to its home base within 20 seconds when required.
    2.3. The robot shall be able to travel at a speed of 0.3 m/s continually for 2 minutes.
    2.4. The robot shall remain functional after an upset event.
        2.4.1. Operation shall be possible after driving off platforms up to 100 mm in height.
        2.4.2. Operation shall be possible after collision with another robot or obstacle.
        2.4.3. The robot should not lose any components during operation unless they are designed to do so.
3. **STRUCTURAL AND ELECTRICAL ARCHITECTURE**
    3.1. The robot should be assembled and disassembled using hand tools only.
    3.2. All components should be replaceable or repairable within 20 minutes.
    3.3. The robot shall fit within a 400mm diameter bounding circle.
    3.4. The Teensy 4.0 should have its ports accessible when in an operational state for programming and serial debugging.
    3.5. The robot should have accessible controls for changing operation states.
4. **SAFTEY FEATURES**
    4.1. The robot shall have no sharp edges.
    4.2. The robot shall have a clearly accessible power cut-off button.

The robot shall not have design features that are intentionally crafted to cause harm to an opponent's robot.

Appendix D – Bill of Materials

In table 2 below the 3D Printed parts from PLA were costed at 5 cents per gram.

Table 4 Bill of Materials

Part	Part number	Quantity	Material for Manufactured parts	Cost (NZD)
TOF Sensor	VL53L1	3	N/A	10
TOF Sensor	VL53L0X	3	N/A	5
Colour Sensor	TCS34725	1	N/A	14
Smart servo	DRS-0101	1	N/A	58
Large Servo	RDS5160	1	N/A	60
DC Motor 143RPM	28PA51G	2	N/A	70
Internal Passage part 1	N/A	1	3D Printed (PLA)	123.22 (grams)
Internal Passage part 2	N/A	1	3D Printed (PLA)	29.97 (grams)
Internal Passage part 3	N/A	1	3D Printed (PLA)	83.07 (grams)
Internal Passage part 4	N/A	1	3D Printed (PLA)	33.21 (grams)
Gate door	N/A	1	3D Printed (PLA)	10.23 (grams)
Pushing door	N/A	1	3D Printed (PLA)	12.84 (grams)
Smart servo mount (aluminium cutout)	N/A	1	Water Jet cut Aluminium	2.5
Smart servo mount (3d printed part)	N/A	1	3D Printed (PLA)	12.17 (grams)
Chassis left side	N/A	1	Water Jet cut Aluminium	4
Chassis right side	N/A	1	Water Jet cut Aluminium	4
Inductive Proximity Sensor	LJ18A3-8-Z/BY	1	N/A	25
Drive track support hardware	N/A	8	N/A	3
Robot tracks	880-8M	2	N/A	Supplied in kit without cost
Main drive wheel	N/A	2	N/A	Supplied in kit without cost

Teensy 4.0 CPU	N/A	1	N/A	Supplied in kit without cost
Internal mounting plate	N/A	1	Laser Cut Perspex (6mm depth)	9
Top mounting plate	N/A	1	Laser Cut Perspex (4.5mm depth)	6
Structural bracket 1	N/A	2	3D Printed (PLA)	4.52 (grams)
Structural bracket 2	N/A	2	3D Printed (PLA)	3.94 (grams)
Structural bracket 3	N/A	4	3D Printed (PLA)	3.44 (grams)
Battery holder	N/A	1	3D Printed (PLA)	32.97 (grams)
4000mAh 3 cell LiPo battery	N/A	1	N/A	Supplied in kit without cost
Stop Go button	N/A	1	N/A	Supplied in kit without cost
Power supply board	N/A	1	N/A	Supplied in kit without cost
Motor drive board	N/A	1	N/A	Supplied in kit without cost
Smart Servo IO Board	N/A	1	N/A	Supplied in kit without cost
Encoder IO Board	N/A	1	N/A	Supplied in kit without cost
Digital level shift IO Board	N/A	1	N/A	Supplied in kit without cost
Inductive level shift IO Board	N/A	1	N/A	Supplied in kit without cost
Assorted fasteners (nuts, bolts, washers)	N/A	100+	N/A	Supplied in kit without cost
Total Cost				409.60

Appendix E – Battery Life Equations

Table 5 Current consumption of each component and quantity at max current draw

Component	Quantity	Current Draw (Max) [A]	Total Current Draw (Max) [A]
TOF VL53L1	3	0.040	0.120
TOF VL53L0X	3	0.040	0.120
TCS34725	1	152e-6	152e-6
DRS-0101	1	0.450	0.450
RDS5160	1	5 (When stalled)	5 (When stalled)
28PA51G	2	3.6 (When stalled)	7.2 (When stalled)

$$I_{Total}[A] = \sum_{components} I_{max} \times quantity$$

$$T_{Battery\ Life} = \frac{4Ah}{I}$$