Vehicle Name: Kaizen

Prof. Anjana Rodrigues – Faculty Mentor – anajana.rodrigues@nmims.edu
Divyansh Garg – Team Captain – divyansh10062004@outlook.com

Date Submitted: 12th May 2024

Team Members

Bhavi Mistry – Core Member – bhavimistry2005@gmail.com
Dhruv Parmar – Core Member – dhruvparmar3840@gmail.com
Pranav Lavande – Co-Captain – pranav.lavande04@gmail.com
Rishabh Bhangale – Core Member – rishabhbhangale@gmail.com
Siddharth Hooda – Core Member – sid.r.hooda@gmail.com
Sunil Idnani – Core Member – sunil.idnani0701@gmail.com
Veer Bafna – Core Member – veerbafna05@gmail.com

I, Professor Anjana Rodrigues, hereby declare that the work done by Team DARVIN under my guidance for the IGVC competition 2024 has been significant and equivalent to what might be awarded credit in a senior design course.

Prof. Anjana Rodrigues
EXTC Department
NMIMS MPSTME

Table of Contents

1. Introduction

1.1 Overview

With the ambition to break past our limitations, Team DARVIN underwent a reboot with a new roster and new plans for IGVC 2024. This brought about the creation of Kaizen, our vehicle that aims to go beyond the challenges faced by Socrates 2.0, with the involvement of a better wheel layout for dynamic balancing, a singular chassis piece that contains the components within rather than having a separate box for the same, making it more secure. It also involves an improved software approach that aims to reduce the workload on the NUC while providing the expected results with ease.

1.2 Organisation

Team Darvin is the robotics club of NMIMS MPSTME, composed of 8 team members that have been involved in the 4 departments of the team: Manufacturing, Electronics, Software and Documentation. The roles of individual members and a brief information about them is as follows:

Table 1. Team Constitution

Name	Graduation Year	Major	Role
Divyansh Garg	2026	Electronics and Telecommunication	Team Captain
Pranav Lavande	2026	Electronics and Telecommunication	Co-captain, Electronics and Manufacturing Lead
Sunil Idani	2027	Electronics and Telecommunication	Electronics Member
Veer Bafna	2027	Electronics and Telecommunication	Electronics Member
Siddharth Hooda	2025	Cybersecurity	Documentation Lead and Software Member
Bhavi Mistry	2026	Computer Science	Documentation and Software Member
Rishabh Bhangale	2026	Computer Science	Software Lead and Manufacturing Member
Dhruv Parmar	2026	Computer Science	Software Member

1.3 Design Process

For the vehicle, we implemented a Rapid Application Development strategy (RAD), check Figure 1 for information about the basics of this approach, avoiding traditional ideologies (like waterfall) to have room for errors than to avoid facing all the errors at the testing phase, which falls during the end of the process. Applying RAD also helps focusing on individual functionalities rather than focusing on the term end goal, which is crucial for ideation and improvement. For RAD, our approach included the following:

- **Requirements**: Initially, we meticulously reviewed the constraints specified for the current edition of the IGVC competition. This involved a comprehensive reassessment of the conditions and a careful examination of necessary adjustments based on prior assumptions.
- **Analysis:** To surpass previous shortcomings and incorporate innovative methodologies, we established specific parameters to gauge the efficacy of implemented components. This analytical phase was crucial for identifying areas of improvement and refining our approach.
- **Prototyping:** To ascertain the optimal feasibility and functionality of our proposed concepts, we engaged in the creation of 3D CAD models, simulated the operations of physical components, and conducted controlled tests. Prototyping served as a valuable tool for visualising and evaluating the practicality of our design ideas.
- **Integration**: Following individual component testing to ensure compliance with predefined standards, we proceeded to integrate these components, including boundary manipulation, camera input/output handling, and the power distribution system. This phase unearthed latent issues, enabling us to implement further enhancements and improvements for the overall functionality of the vehicle.

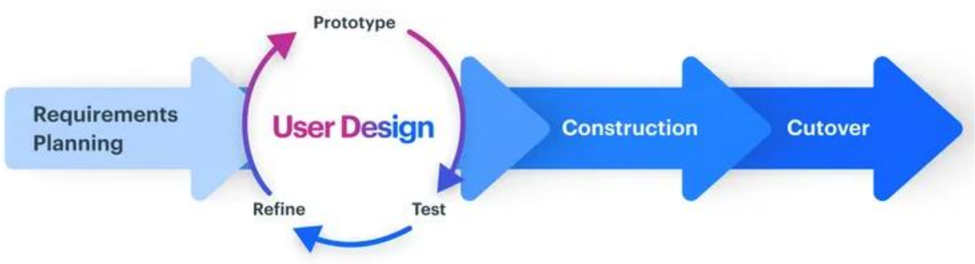


Figure 1. Development process utilized for Kaizen

The design process was centred on creating a vehicle tailored to the specific needs of the IGVC competition, with a focus on innovation, reliability, and optimization. The team pursued continuous iteration, testing, and refinement to achieve an optimal design, aiming to maximise the vehicle's capabilities and increase its likelihood of success in the competition.

2. Vehicular Requirements

2.1 Assumptions

Our main goal for this year's vehicle was to improve the mechanical failures of our previous year's vehicle which included unbalanced weight distribution and stability of the vehicle with an aim to improve our angle of attack on steeper slopes and ramp situations. See Section 4.3 and Table 5, for results obtained. We also aimed to implement less tasking software code to make the auto-nav process smoother along with a significantly larger battery backup which in turn increased the range of the vehicle significantly.

2.2 Costing

The vehicle costs were widespread with most of the budget being allocated to selective few top-of-the-line industry standard components which included the onboard computers (NUC) and the camera module. The rest of the budget was used in the mechanical aspects of the vehicle. To ensure the quality of the PCBs used in the vehicle we custom designed them. Below are the costs incurred for the major components and where we received them from:

Table 2. Expenses for Major Components

Sr No.	Item	Cost (In INR)	Source
1	DC Planetary Motors	8,500 per motor	University
2	Motor Driver	3,000	University
3	ZED 2	37,500	Sponsored
4	NUC	70,000	University
5	Display Screen	25,000	University
6	All Terrain Wheels	4,500 per wheel	Sponsored

2.3 Structure Quality

To ensure the quality of the chassis, we built it purely via aluminium extrusions, with metal bolts and brackets to hold them together in forming one uniform mass that encompasses the entirety of the vehicle rather than having it as separate components combined. The bottom side of the vehicle is reinforced with 2mm thick stainless steel which house all the electronics in the vehicle. The stainless steel will provide mechanical protection to the underbody of the vehicle from any debris that could be potentially flung off from the ground while the vehicle is in motion. The sides of the vehicle are secured with a combination of aluminium and acrylic plates. The top half which is also the only entry point into the vehicle is also secured with a layer of acrylic. See Figure 2.

2.4 Safety Mechanisms

To keep the vehicle, secure in all contexts, we implemented multiple safety protocols to reduce the attack surfaces and their impacts. We implement a remote e-kill that can safely stop the vehicle when required that does not shut off the whole vehicle, allowing quick restarts. We also implemented software protocols like three-way handshaking and heartbeat protocols to reinforce the safety of the device from any thefts. See Section 5.5

3. Innovations

3.1 Mechanical Innovations

1. Swappable panels: To improve the durability of the vehicle, Kaizen is equipped with swappable side of aluminium and acrylic. These protect the interior of the vehicle, comprising of the electronic components from any mechanical shock or debris.
2. Easy transportation: Rather than having a large singular piece for the chassis, we constructed the vehicle with non-singular aluminium pieces to allow for easier transportation.

3.2 Electronic Innovations

1. Multiple kill options: To protect the integrity and availability of the vehicle, Kaizen contains multiple kill options, ranging from mechanical switches, to turn off the electric

supply to the motors or the NUC, to electronic kill options that can be executed remotely to stop the motors in case of any emergency or derailment from the expected route. See Section 5.5
2. Heavy duty wiring: Kaizen employs 18 & 20 AWG silicon wires to ensure the power supply occurs seamlessly even at high voltages.
3. Monitoring system: Mini displays are employed to show the voltage flowing across the electronic suite in real time. This allows us to check the battery voltages easily.

3.3 Software Innovations

1. Heartbeat protocol: Kaizen includes a heartbeat protocol in its software that allows the user to continuously monitor the working of the vehicle even in case of the vehicle being outside the vision of the vehicle. The vehicle sends an alive message at regular intervals while operating and sends a death message in case of its systems not working. See Section 7.4
2. Cheaper alternative: To reduce the cost of constructing an autonomous vehicle, which usually employs a LIDAR to detect obstacles and path navigation, we wrote a novel code that does the job of object navigation using YOLO V8n, and lane detection with the help of Canny edge detector. See Section 6
3. Remote start: For ease access of the vehicle, we have a remote shell that can be accessed via an Android phone to execute the codes or to monitor the output and heartbeat messages from the protocol mentioned above.

4. Mechanical Design

4.1 Overview

The vehicle's design, reminiscent of the Socrates 2.0, features a 3-wheel setup for stability, with 2 driving wheels upfront and a castor wheel at the rear. Its modular aluminium chassis allows for versatile attachment configurations. Robust shielding, including stainless steel, acrylic, and aluminium, safeguards internal electronics. Remarkably resistant to body roll, the vehicle ensures stability even under heavy loads. Constructed with 52.49 ft. of aluminium, it boasts heightened rigidity and a 35lbs payload capacity. An improved 30° angle of attack enhances traversal across various terrains. Its aluminium extrusion construction enables swift shell replacement for adaptability. Exterior and interior mounting points facilitate easy accessory installation, enhancing versatility.

4.2 Chassis and build:

Kaizen, as shown in Figure 2, features a sturdy aluminium chassis constructed from 30x30 mm extrusion profiles, reinforced with aluminium L-brackets for stability. Its weight distribution across three wheels ensures stability, with a centre-heavy design preventing rollover in unloaded conditions. To prevent tipping under full load, a counterweight of 10-15% of the payload is recommended. The chassis houses electronics secured by a 2mm stainless steel plate and utilizes various methods to secure components and batteries. The electronics bay is sealed with aluminium extrusions, acrylic, and aluminium sheets, with access restricted to the top. A mesh rear end provides ventilation, while silicone seals enhance water resistance.

Figure 2. Chassis build

4.3 Capabilities

1. **Stability:** With a 3-wheel design and having a wide wheelbase of 34 inches and a length of 31 inches there is minimal body roll and the chances of Kaizen toppling over is almost none.
2. **Speed:** In the current motor and wheel configuration, we can achieve a top speed of 3.57 MPH.
3. **Weight capacity:** The payload carrying capacity is tested up to 30 lbs. The payload can be secured using the aluminium extrusions present in the front end of Kaizen.
4. **Incline:** We have also tested the vehicle to climb up to 30 ° inclined. We have ground clearance of up to 4 inches which can traverse over uneven terrain easily.
5. **Sturdiness:** Kaizen's side panel are a mixture of 2 materials which includes 6mm Acrylic sheets and 1.5 mm aluminium sheet. The electronics are screwed into a 2 mm solid stainless stell sheets which can also help and protect Kaizen from any underbody debris deflected off while the vehicle is in operation.

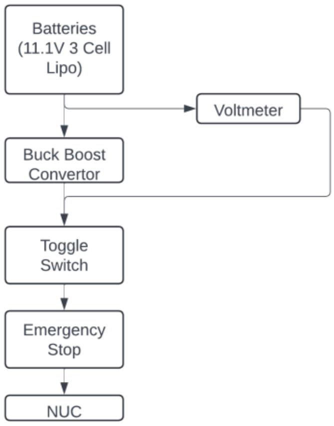
5. Electrical and Power Design

5.1 Overview

The electronic system of the entire vehicle consists of 2 separate systems working together which allows smooth operation of the vehicle. These systems include one system for the on-board computer (NUC) and one system for the driving of the motors and other peripherals, mentioned in Figures 3 and 4. It is also equipped with multiple redundancy systems in case one of the components fail.

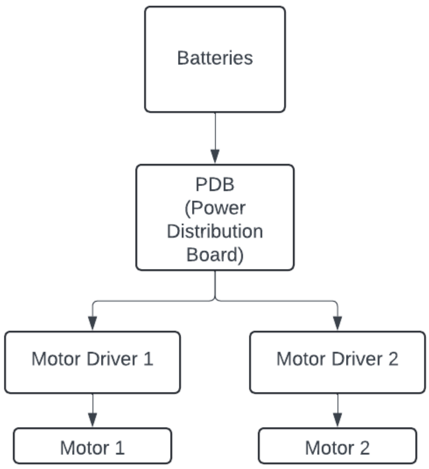
5.2 Power Distribution

Kaizen is powered by an ASUS NUC which is a compact computing unit which runs all the programs and is responsible for interfacing with all the microcontrollers. System A is the system, which is isolated for the NUC, and system B is the system which takes care for all the other electronics.



System A: This system uses a 11.1V 3 cell LiPo which is boosted up to 20V using a 150W buck boost which is a DC-DC converter. Their output is then stabilised before it reaches the NUC.

Figure 3. Power supply path for System A covering from batteries to the NUC



System B: This system is responsible for providing the required voltages to all the microcontrollers and motors. This contains 2 LiPo batteries of 11.1V 3 cells connected in series giving it a base voltage of 22.2V, which at full charge can go up to 26V. This is passed directly through a power distribution board which splits this voltage into multiple outputs. These are then used to drive the motors. An 11.1V lead acid or LiPo battery can be used to power the relay and the emergency light. This ensures that even if all the other systems are dead the emergency light will remain working.

Figure 4. Power supply path for System B, from the batteries to the motors

5.3 Motors and Motor Drivers:

Kaizen uses 2 Planetary Geared DC motors which have an operating voltage of 24V. The motors have a gear ratio of 1:23 with an output of 120 RPM. The motors are rated for a torque of 7.8 Nm. We have also used tyres of 10 inches this helps us reach a top speed of 3.57 MPH (5.75 kmph). We can increase the speed and performance of the vehicle by increasing or reducing the size of the wheels used.

5.4 Electronics Suite

Kaizen uses industry grade motor drivers which include Rhino MD20A motor drivers which support up to 30V and 60A of current. The on-board sensors include an IMU which is a Sparton module and for the GPS we are using a Spark fun Module with an antenna for a precise location tracking. We have also used an ASUS NUC equipped with an intel i5 12th gen for all the computing and processing of all the information which is received from the cameras and other peripherals. We have used an Arduino Mega as a microcontroller as there are no pins in the ASUS NUC unlike the jetson TX2 used previously in Socrates 2.0. The microcontroller is used to transmit the necessary PWM control to the motors which help the car drive in our desired direction. The Arduino and the NUC are interfaces using a ROS bridge. We are also using some smaller electronics like a DC-DC voltage regulator (buck boost converter). Which regulates and stabilizes the voltage for the NUC. Figure 5 expands on the various components utilized in the electronic suite of Kaizen.

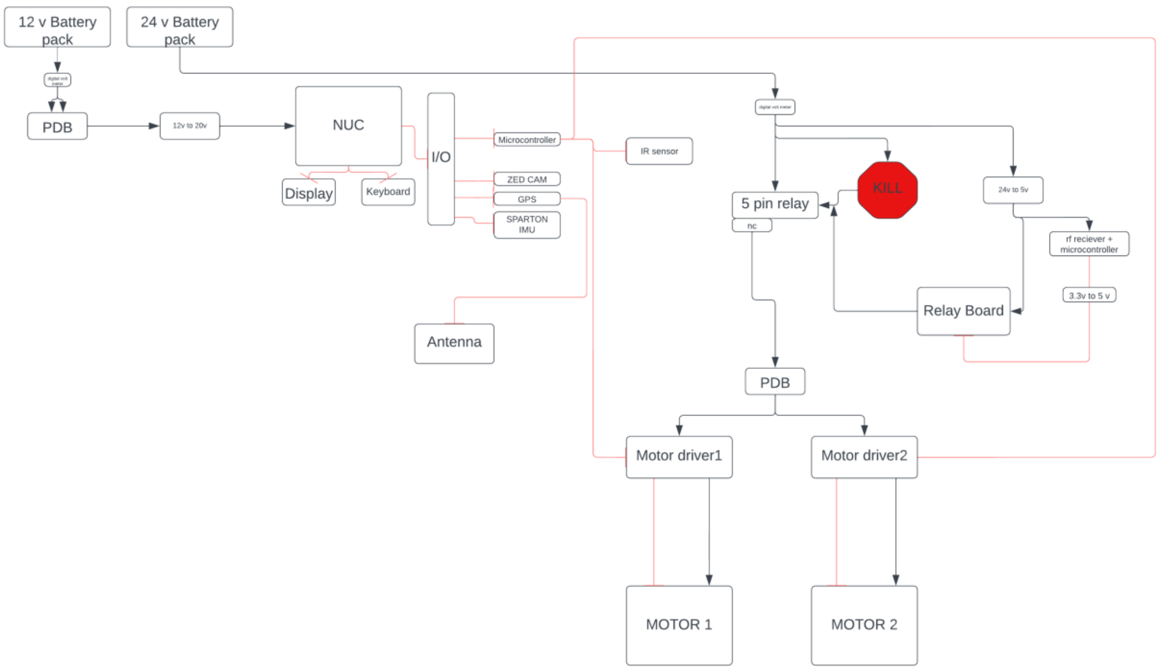


Figure 5. Electronic Suite of Kaizen

5.5 Safety Mechanisms:

Kaizen is equipped with various safety mechanisms including 2-point authentication to start any electronic system. We have 2 switches per system which has one emergency kill switch and one toggle switch. Once both switches are in the ON position, only then the electronic system is activated. The emergency kill is accompanied by a E-kill which is a wireless version of the emergency kill and the functionality remains the same of both the kill switches as they are linked to the same relay which kills the battery supply to the power distribution board. We have also integrated a soft kill which is a button mapped on the controller which kills all the currently running codes on the system.

6. Software Logics

6.1 Overview

Kaizen utilizes 4 sets of publishers and subscribers in the ROS for values related to /lane_daf, /object_daf, /bot_gps and /bot_imu, shown in figure 6 below. These broadcast float data, bounding box data (the nearest box point to the vehicle), latitude and longitude points and the z-axis angle between the waypoint and current position of the vehicle. These point information are then used to correct the current path to navigate through the oncoming obstacles within the NUC, with the help of ROS. The corrected path is relayed through the microcontrollers to the motor drivers, which control the PWM of the motors accordingly to navigate according to the data received.

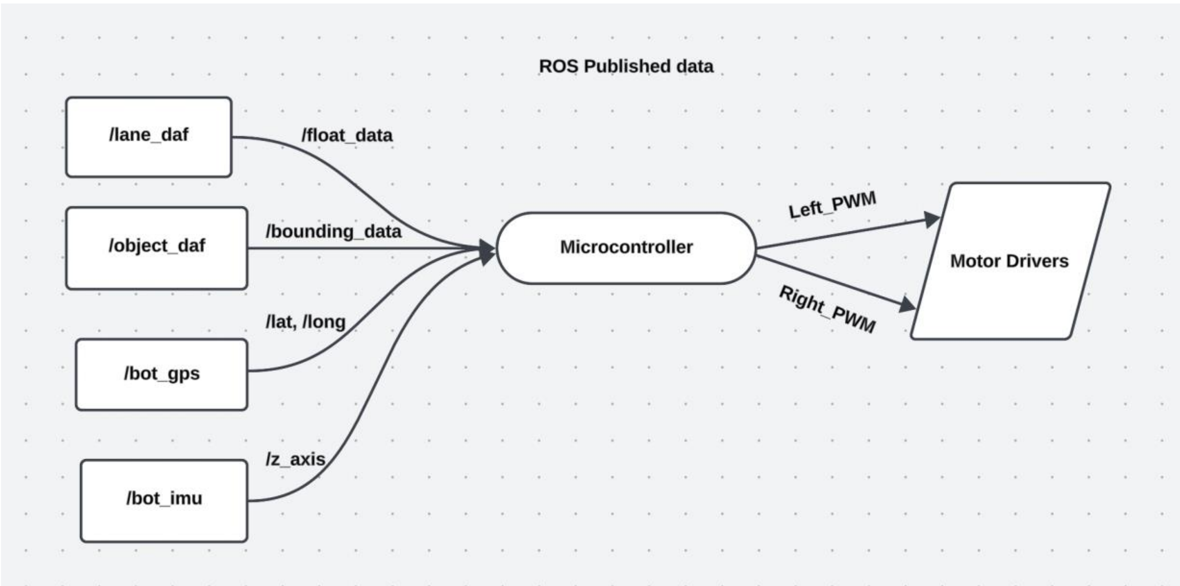


Figure 6. ROS data published and subscribed in Kaizen

6.2 Lane Detection Code

The vehicle employs a Python code which involves multiple checkpoints throughout its lifecycle to ensure smooth operation. The first condition is for ensuring the camera input is constant and visible to the system to ensure it can convert the input to grayscale. A gaussian blur is employed on the grayscale image for easy identification of the track lines. This input is then given to the Canny edge detector to identify the edges in the path. Further, the code defines a region of interest, which would be the area within which the path is to lie in the system. The region of interest's edges is mapped into an array, which allows it to define the farthest and nearest points (2 each) and the centre points between the two distant points. Another centre point, one for the vehicle, more specifically, the camera, is measured and the three centre points are mapped together to identify the map to be followed by the vehicle in the region of interest. A turn would be indicated by a shift in the edges, which would then change the centre points accordingly.

6.3 Object Navigation Code

The object tracking code, on the other hand, employs the camera input in a different manner. This camera input is then divided into a set number of grids. The neural network of the algorithm has been pre-trained to detect traffic cones. Each grid cell predicts a percentage of how likely it is that the image inside that specific grid cell is part of a traffic cone. Through this a few bounding boxes. All the bounding boxes have their own confidence grade. Higher the confidence grade, higher the accuracy of its prediction. Two thresholds are defined one for permissible difference between the wheel of the vehicle and the nearby bounding box, and one for the extreme case of a difference between the wheel and the bounding box, along with a constant value for slightly more than the difference between the centre of the vehicle and the wheel. In case of a singular object being avoided, is the distance between the wheel and the bounding box within the permissible threshold. Until the object does not enter the extreme case it does not affect the vehicle's path. As soon as the obstacle is within the extreme threshold, the vehicle moves itself to a magnitude that the centre of the vehicle and the extreme edge of the bounding box is well within the permissible range.

6.4 GPS Navigation

The GPS Navigation code takes values from the GPS module connected to it through a python script. These co-ordinates are the present co-ordinates of the vehicle up-to 6 digits after decimal. Then the co-ordinates of the GPS points present on the track are considered as centre of a circle to which we create a parameter of certain length. The vehicle heads towards the GPS modules constantly checking whether the current co-ordinates of the vehicle are not within the parameter created around the GPS points. As soon as the vehicle detects that it has entered the parameter it switches from the lane detection and object detection code to GPS Navigation code and object detection code. After the GPS point has been crossed the vehicle checks the co-ordinates of the current heading of the vehicle and tries to match it with the next GPS point while avoiding objects using the object detection code. After crossing the second GPS point the vehicle again switches to lane detection and object detection. This goes on for all the alternate GPS points.

7. Attack Surfaces and Countermeasures

7.1 NIST RMF (Risk Management Framework) Overview

The NIST Risk Management Framework (RMF) is a critical cybersecurity process designed to assist organisations in establishing and maintaining secure information systems. It offers a structured and standardised methodology to identify, assess, and manage risks associated with IT infrastructure. The primary goal of the RMF is to facilitate the development of secure and resilient systems by implementing essential security controls and promoting continuous monitoring. This proactive approach, as shown in Figure 6, aims to counter the evolving cyber threat landscape and ensure the robustness of information systems. It involves:

1. **Categorise**: In this initial step, organisations identify and classify their information systems based on the potential impact levels of a security breach. This categorization lays the foundation for subsequent risk management activities.
2. **Select**: Organisations choose and implement appropriate security controls based on the categorised systems. These controls are tailored to address specific risks and vulnerabilities identified in the earlier step.
3. **Implement**: The selected security controls are put into practice across the organisation's information systems. This step involves integrating security measures seamlessly into the existing infrastructure to enhance overall resilience.
4. **Assess**: Regular assessments are conducted to evaluate the effectiveness of implemented security controls. This step ensures that the controls are performing as intended and identifies areas for improvement or adjustment.

5. **Authorise:** Following successful control implementation and assessment, the organisation authorises the operation of its information systems. This step involves a thorough review to confirm that the security measures are in place and operational.
6. **Monitor:** Continuous monitoring is a crucial aspect of the RMF, involving ongoing surveillance of information systems to detect and respond to emerging threats. This step ensures that security measures remain effective in the face of evolving risks.

Figure 6. Risk Management Framework used in Kaizen

7.2 Threat Cases and Impact Analysis

A scenario with the threat to the vehicle being how it could be compromised software-wise by rival teams at the pit can involve multiple vulnerabilities, with varying degrees of threat.

Table 3. Vulnerability Control Codes and Threat Modelling

Vulnerability	NIST Cyber Control Code	Confidentiality	Integrity	Availability	Overall Impact Rating
Unauthorised Code Repository Access	AC-16, AC-19, IR-4	High	Low	Low	Low
Unauthorised Hardware Insertion	PE-3, PE-6, PE-9, PE-5	Medium	Medium	Medium	Medium
Unauthorised Remote System Access	AC-17, AC-7, AC-18	Medium	High	High	High
Insider Threat	IA-2, IA-3, IA-4, AC-5	Medium	Medium	Medium	Medium
Sensor Spoofing/Tampering	IA-5, IA-8, IA-9	Low	Low	Medium	Low

7.3 Implementable Security Control Measures

1. **Strong password policy (PS-1):** Regardless of the content being protected, one must always follow a strict password policy to ensure no system or information access is obtained via weak passwords for crucial accounts. We can implement policies followed by Apple for example where they do not allow personal identifiable information related to the user being in the password, strengthening it against social engineering attacks.
2. **Alert systems (SI-7):** Including alarms, notifications for unauthorised actions to alert the system owners that someone is trying to break into the system. This can dissuade any potential attacks on the vehicle, especially in the pit area where everyone would be present and proving malpractice would be easy.
3. **Password Rotation (PS-1):** Implementing a program that would be required to be executed on a regular basis, preferably weekly, which would generate a new password for the root user that would be then encrypted before being stored in the shadow file.
4. **Emergency wipe (AC-7(2)):** Having a hard kill that could encrypt all the system files and delete its encryption key to permanently hide the information of the system from any attacker as a last stage measure would protect the integrity and confidentiality of the system at the cost of the system's availability.

7.4 Implemented Security Control Measures

To harden our system, we implemented multiple security measures including:

1. **Maintaining a private repository (AC-6):** A protected repository that has limited access depending on the user id allows us to prevent any competing team from gaining knowledge and understanding of how our vehicle works, protecting the confidentiality of the team's work. By following the Principle of Least Privilege, we ensure that not any authorised id can gain access to the entirety of the system's underlying code. This also includes a honeypot that when tried to open will lead to the attacker's system being Dossed till it hangs, deterring any future attacks if possible.
2. **Physical hardening (PE-3(5)**: To prevent any unauthorised access via any attacker, we have disabled all unused ports of our device with the help of tamper proof seals, allowing only the ports to be utilised by our systems, like visual input device, microcontrollers, and input output system.
3. **Establishing an RBAC (AC-2):** By implementing a Role Based Access Control System on the Linux system with each file having different permissions for different users, we protect the integrity and confidentiality of the system. This involves making distinct roles: *Lead, Devs, Other* with a decreasing privilege across the roles, with *Lead* having full access meanwhile *Other* having limited access.
4. **Heartbeat function (PE-20):** To let the user know that the vehicle is functioning when in the case of being out of sight, we implemented a heartbeat function as present in TCP/IP protocol to ensure the vehicle is functioning with no problems. This is displayed on a phone via SSH shells to show the outputs live from the computing system.

8. Failures faced and resolutions made

Table 4. Failure points faced across Software, Electrical and Mechanical fronts

Failure Points	Cause	Resolution
Early mapping of lanes	Camera alignment with respect to height and view	Re-calibration of camera mount angle and height to

	input it received led to it being able to see further and move earlier to react according to input.	ensure the input it can see and react to is at a nearer distance to avoid premature reactions.
Loss of camera data	Harsh weather conditions or undetected obstacles can be glossed over in the unlikely case, causing vehicle crash.	IR sensors placed at the front of the vehicle can always detect any obstacle coming too close to the vehicle, which will lead to the vehicle being stopped instantly.
Components not functioning due to improper connections	Open wires were inserted directly into microprocessors, motor drivers and buck boosts, causing short circuits between neighbouring wires.	With the help of ferrules, we were able to make a single point of contact for input wires connected to motor drivers, microprocessors, and buck boosts.
Power surge can damage connected components	Conductive points were exposed to a conducted surface, causing amplification of current flowing through the components.	Sectioning and insulation of the conductive surface was implemented to avoid exposure for the conductive points.
Chance of wheels wobbling while running	Exposure to torque for the wheel can lead to the coupler being loosened.	We built a custom wheel axel which integrates the coupler to ensure the coupler does not get loose.
Excessive heat can be entrapped in the vehicle	While operating in high temperature environment, the full metal body can entrap heat causing thermal throttling.	We have kept the back side of Kaizen in a mesh pattern to allow air circulation in the interior of the vehicle, allowing the vehicle to cool down as much as possible.

9. Simulation

9.1 Mechanical performance

Since Kaizen is based on the same wheelbase of Socrates2.0 the stability was never an issue when designing the body for Kaizen. We have implemented a novel approach for housing the electronics this year making it a unibody design rather than 2 separate boxes stacked on top of each other. This design change has allowed us to increase the weight carrying capacity from 25lbs to about 35lbs.

9.2 Electrical system review

Keeping the core design similar and shifting back from the BLDC motors to brushed DC motors we have some changes in the electronics involved. The motor drivers were stress tested up to their peak limits. While ensuring new and updated components were used in the building of Kaizen. The motors were also tested at their maximum RPM with full load for 10 minutes. We have also ensured to test all the switches used in Kaizen for their peak ratings.

9.3 Software review

Testing the effectiveness of our software logic, we simulated expected scenarios for the vehicle for which, via gazebo, a path of obstacles was created. The camera input was successfully able to dictate the path to be followed by the vehicle and obstacles were avoided by looking for the nearest bounding box points.

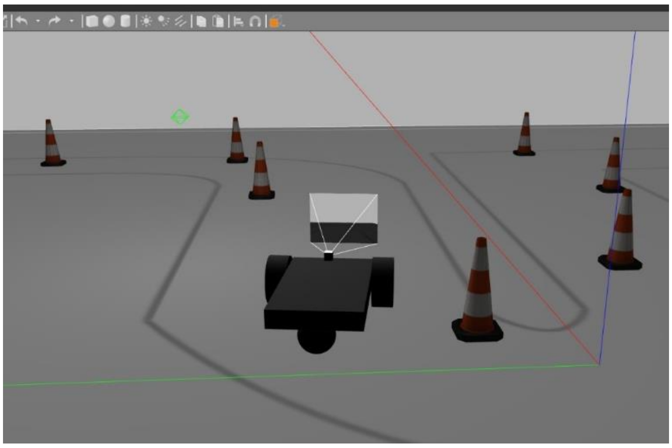


Figure 7. Gazebo simulation for Kaizen's lane navigation

10. Performance Assessment

The key difference between full load vs ideal scenarios for each element is a different case. As such, for the NUC, the full load would involve having to run graphical tasks like YOLO. The motors would face the full load when carrying the base weight of the vehicle, the payload and a counterweight. The runtime of the vehicle is measured by the time NUC runs on the given batteries, with the motors having a lesser consumption rate than the NUC. The E-Kill range does not change under external situations, and as such remains untouched. The controller's signal range, on the other hand, can vary depending on the panels being used in the vehicle.

Table 5. Evaluation elements at varying conditions

Evaluating Element	Full Load Analysis	Ideal Analysis
On Board Computer (NUC)	120W power draw	85W power draw
Motors (motor1 + motor 2)	4A + 4A (3.5MPH full load)	2.75A + 2.75A (2MPH)
Runtime	40 mins	75 mins
Speed	3.5 MPH	1.5 MPH
Payload Capacity	40 lbs (with 5% counterweight)	20 lbs
Range (E-Kill)	100 feet	100 feet
Range (Controller)	65 feet (No side panels)	40 feet