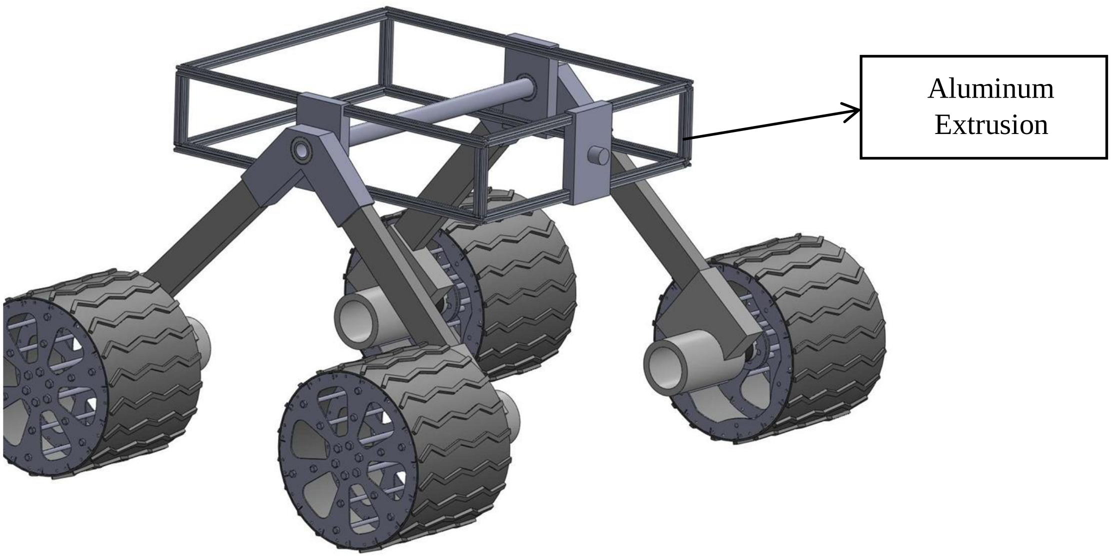
- **Type of Robot: Autonomous Rover[IGV]**
- **Link of CAD :** https://drive.google.com/drive/folders/1Fbs--0POYBewVTIGlXbTmj3GZvEJPO7q
- **Robot Assembly Design** (Proposed Diagram): Drawing each part of the robot is preferred as an attachment. (CAD drawings are preferred) (400 words)

  NASA's Mars missions demonstrated the versatility of the rocker-bogie in Curiosity and Perseverance, and by following the detailed guidelines and tasks provided by **GUJCOST**, we re-engineered the concept into a 4-wheel rover with a rocker-type suspension and a **differential stabilizer bar**. This configuration reduces weight and kinematic complexity while preserving terrain adaptability, enabling traversal over hard rock, loose soil, and slopes up to 45°. The stabilizer minimizes chassis roll, ensuring continuous wheel–ground contact, higher traction efficiency, and a stable platform for autonomous sensing and navigation.

  - The rocker mechanism consists of **two rocker arms joined at a central pivot**, designed at a **90° orientation** to minimize induced stresses during terrain traversal.
  - The overall **dimensions of the rover are 83 × 68 × 60 cm (L × W × H)**, making it compact yet stable for IGVC field navigation.
  - This geometry distributes loads evenly, reducing **strain concentration** and chances of permanent deformation in the chassis.




  - Each rocker arm supports a pair of wheels, forming a **4-wheel locomotion system** capable of negotiating ramps, bridges, and uneven terrain.
  - The two rockers are connected via a **differential stabilizer bar**, which balances the motion between left and right sides.
  - The stabilizer ensures that when one side rises over an obstacle, the opposite side compensates, keeping the **chassis level**
  - This enhances **stability** on slopes up to 45°, critical for ramps and dual incline sections in the IGVC arena.
  - The mechanism enables **alternate working of both rockers**, distributing impact forces and maintaining continuous wheel contact with the ground.
  - Reduced chassis oscillation improves **sensor accuracy** for localization, obstacle detection, and QR/AR code scanning during autonomous navigation.

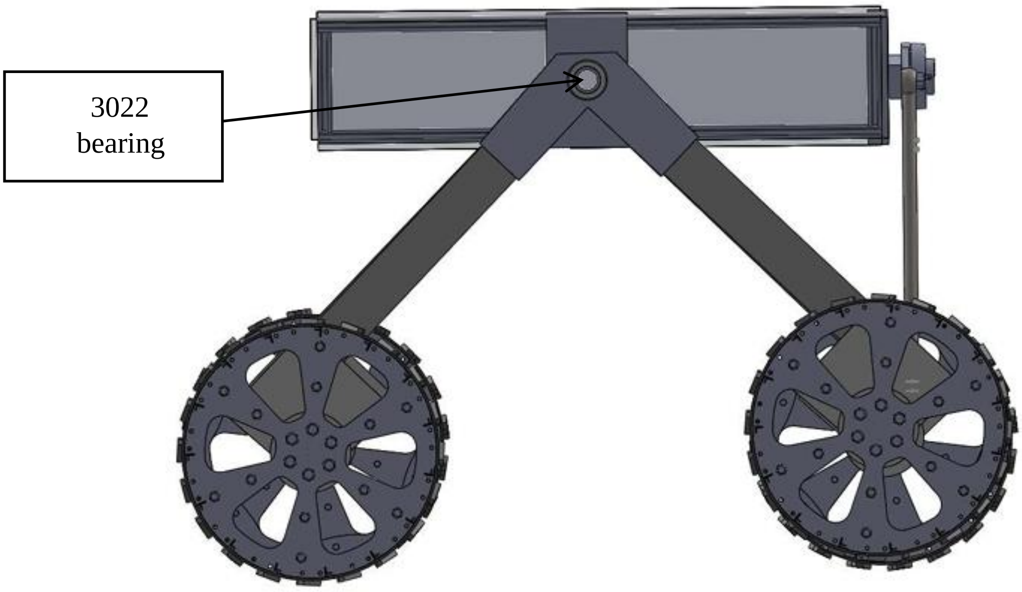



Fig: Side View

- The rocker system is compact, lightweight, and energy-efficient, meeting IGVC **design expectations** of mobility and endurance.
- Overall, it provides an **adaptive suspension** solution that enhances rover performance in the IGVC challenge, especially in Stage 3 (Terrain Challenge).

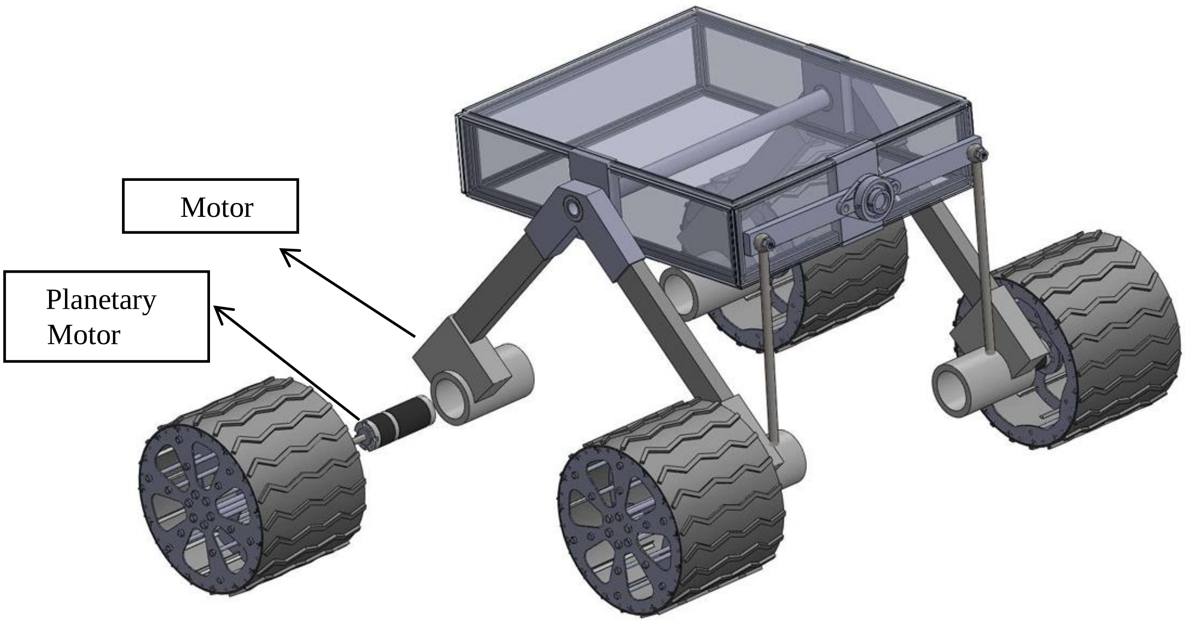
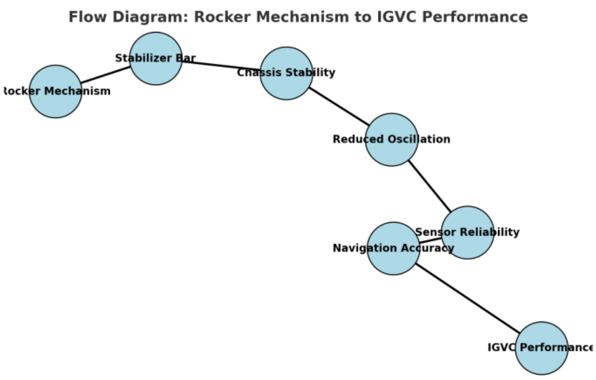
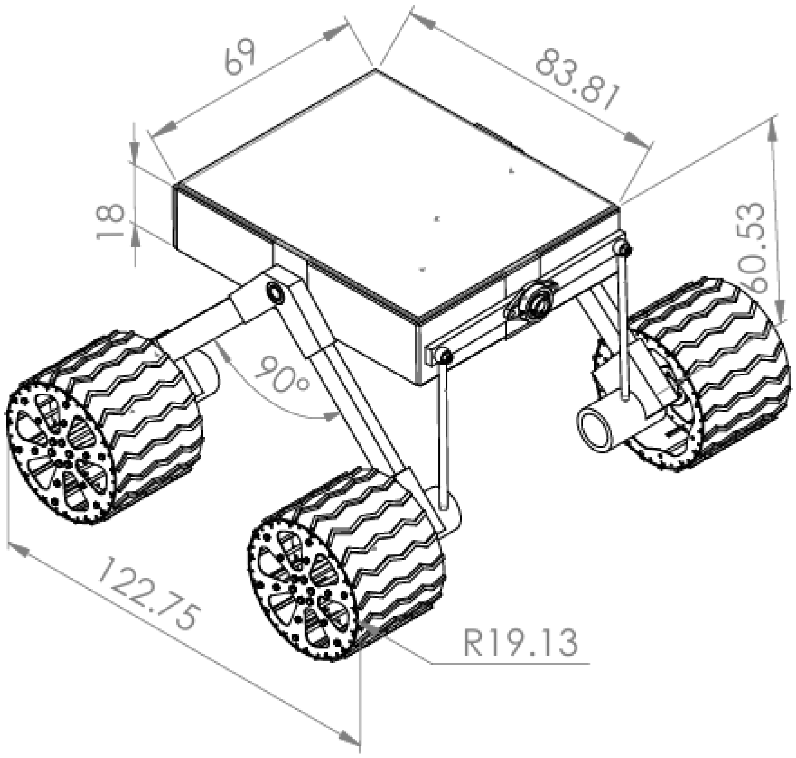



Fig: Wheel Assembly

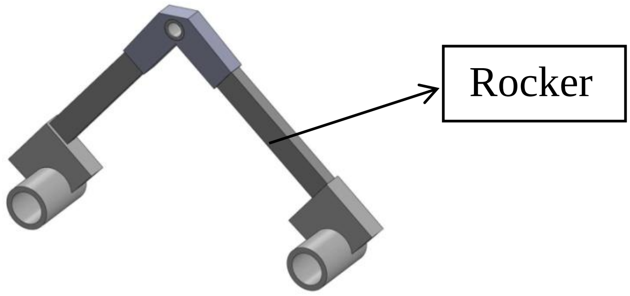



- **Design specifications**

  - The rover is designed with compact dimensions to maintain stability while fitting within IGVC arena constraints.
  - Length (83.81 cm) ensures longitudinal balance, Width (69 cm) prevents toppling, and Height (60 cm) allows mounting of sensors at optimal levels.




  - Constructed using a lightweight **aluminum alloy frame**, which provides strength-to-weight efficiency. Integrated with a **rocker-stabilizer suspension system** for improved terrain adaptability.
  - Suspension Mechanism Rocker arms are joined at **90° orientation**, connected via a **differential stabilizer bar**. This ensures one side compensates for the other, reducing oscillations and keeping the chassis balanced.
  - The rover can climb and descend ramps with inclines up to **45°**, aligning with IGVC terrain requirements. Achieved through optimized torque–weight ratio and suspension design.




- **Differential Stabilizer**

  - The stabilizer ensures that when one rocker arm lifts (due to terrain), the opposite rocker compensates, keeping the chassis relatively level, It transfers motion from one side to the other using a simple **mechanical linkage.**
  - At the top of the chassis (center), added a **horizontal shaft** running laterally across this shaft acts as the **pivoting differential bar**, mounted on bearings to allow free rotation.
  - From each rocker arm (left and right), connected a **linkage rod** to the ends of the differential bar. These linkages are pinned/bolted to allow angular motion.
  - When the **left rocker rises** due to an obstacle, its linkage **pushes/rotates the differential bar**, this rotation **pulls down the right linkage**, lowering the right rocker slightly to balance, this equal and opposite motion keeps the main chassis box stable.
  - The stabilizer bar and linkages were likely made of **mild steel rods**, chosen for **torsional strength** while being lightweight.

- **Wheel**

  - The wheels have an **outer diameter of 190 mm** and an estimated **tread width of 150–180 mm,** designed with **multi-layer rim rings** connected via circumferential plates. A **radial spoke arrangement** links the hub to the rim, ensuring structural rigidity and even stresses distribution.
  - Manufactured from **Aluminum Alloy 6063-T6** for high strength-to-weight ratio, corrosion resistance, and machinability.

| Property | Value | Units |
|---|---|---|
| Elastic Modulus | 703599.9068 | kgf/cm^2 |
| Poisson's Ratio | 0.33 | N/A |
| Tensile Strength | 917.7389722 | kgf/cm^2 |
| Yield Strength | 421.8391776 | kgf/cm^2 |
| Tangent Modulus | | kgf/cm^2 |
| Thermal Expansion Coefficient | 2.3e-05 | /ºC |
| Mass Density | 0.0027 | kg/cm^3 |
| Hardening Factor | 0.85 | N/A |

  - Each wheel is estimated to weigh **1.5–2.0 kg,** giving a combined mass of **6–8 kg for all four wheels** (~15–20% of the rover's 40 kg). This lightweight design reduces inertia and power consumption, improving overall mobility efficiency.

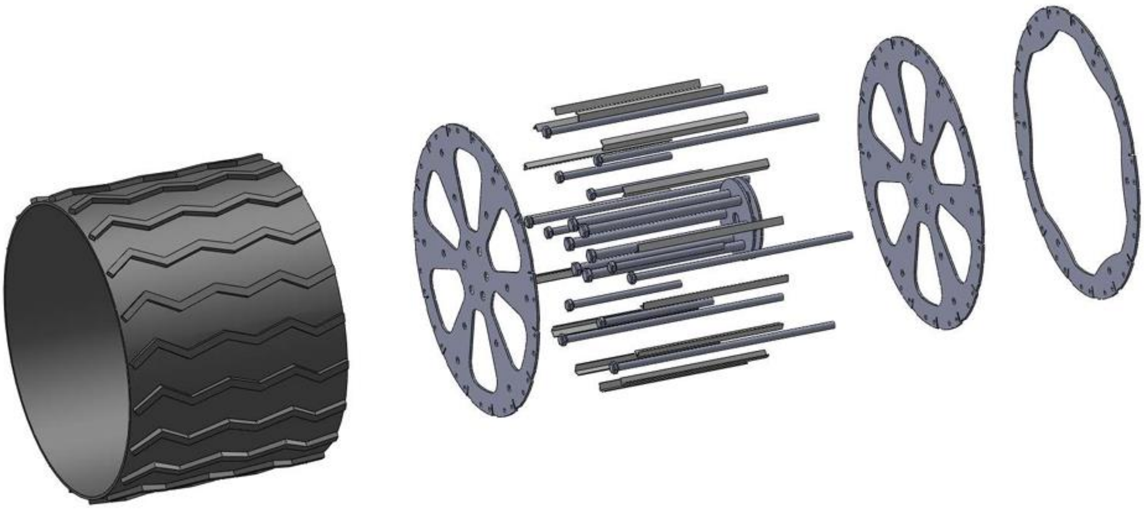
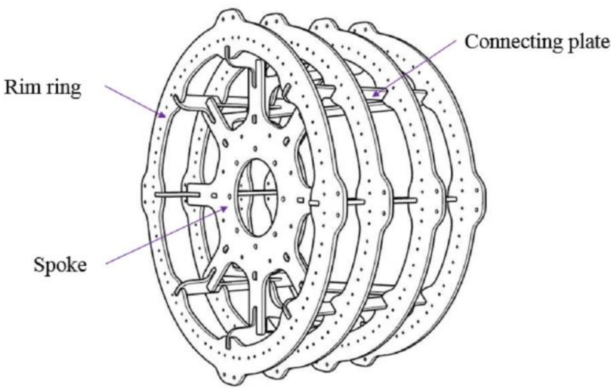


Fig: Exploded View

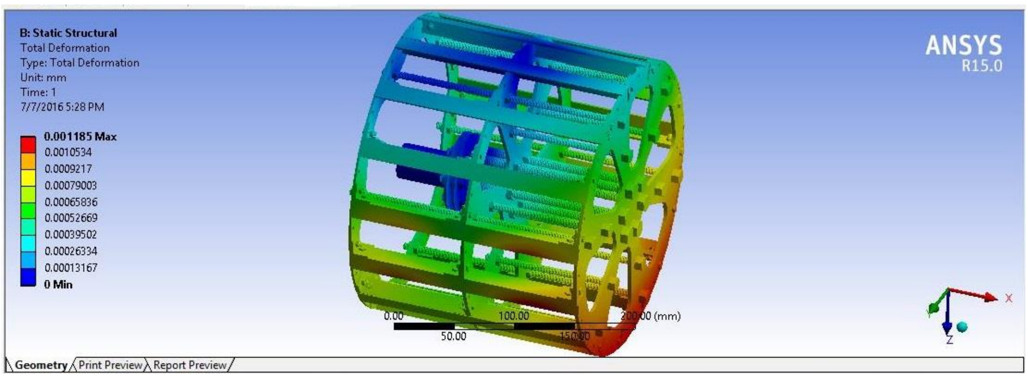
- **ANSYS** simulations indicate a **maximum deformation of only ~0.0012 mm** under static loading. Each wheel can safely sustain **15–20 kg load,** maintaining a **Factor of Safety (FoS) of 2.0–2.5** against yielding, making them suitable for rugged planetary environments.
- The **cut-outs in rim rings** enhance **traction and grip** on sandy or rocky terrain while reducing slippage. The **open-spoke, hollow structure** prevents soil/mud accumulation (self-cleaning) and minimizes weight, while the distributed rim design improves **shock absorption** on uneven surfaces.




- **Calculations**
  - Per-wheel loads
    For flat ground
    $$\mathbf{N(flat)=mg/4= (40\times9.81)/4= 98.1\ N}$$
  - On 45° slope
    $$\mathbf{N(45)=(mg\cos45)/4= 69.37\ N} \qquad \mathbf{F(\parallel,45)=(mg\sin45)/4= 69.37\ N}$$

- Wheel torque requirement (45° climb)
  
  Basic torque (per wheel): **T(basic)=($F_{\parallel,45}$XRadius) = 69.37×0.125=8.67 N\m**
  
  **Therefore Motor/gearbox sizing: with a 30:1 gearbox, motor torque needed**

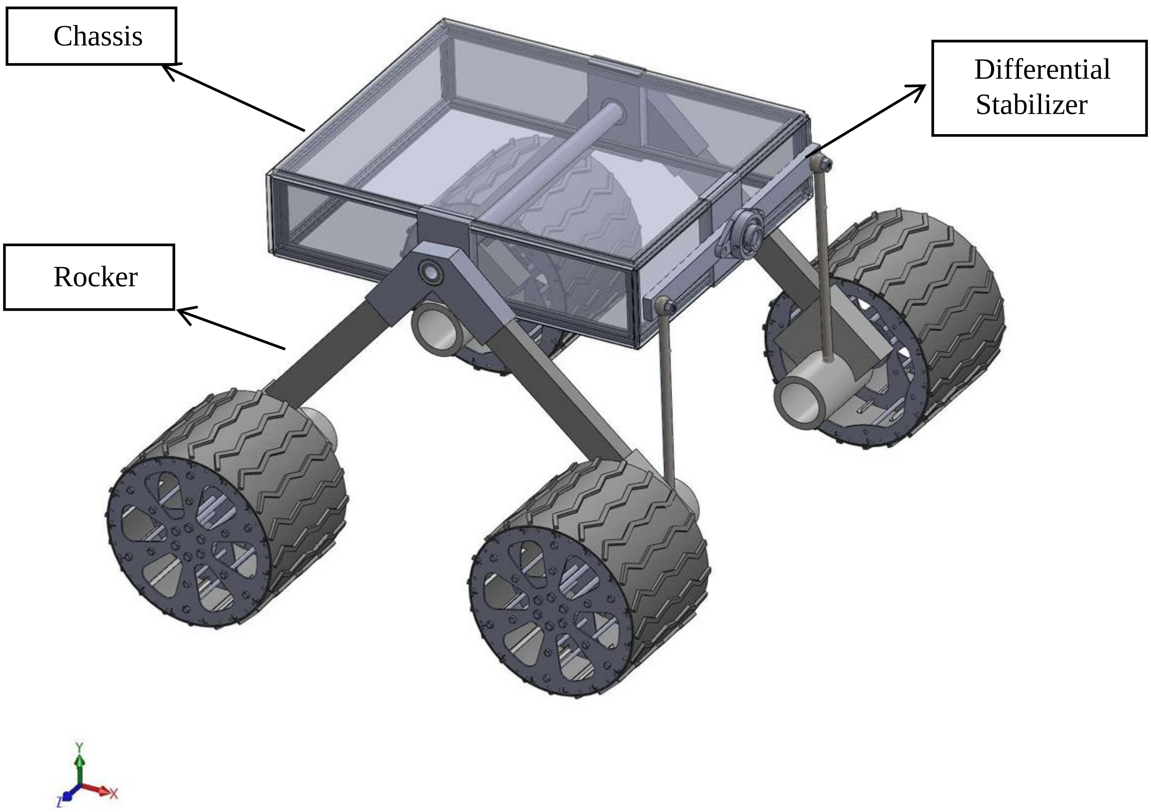
- Traction requirement for 45°
  
  No-slip condition: **μ≥(N45)/($F_{\parallel,45}$) =** tan**45∘=1.0**
  
  Design target with margin: **μ≥1.2**→ justify chevron lugs or rubber tread.



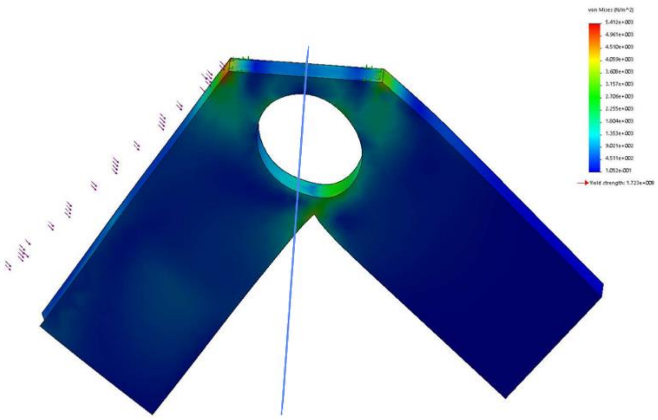

- **FEA** shows **max deformation ≈ 0.0012 mm** under the applied static load — essentially rigid. With **Al 6061-T6** (yield **≈ 275 MPa**) the peak von Mises stress is **< 120 MPa**, giving a **factor of safety ≈ 2–2.5.** This confirms the wheel is **stiff, lightweight, and structurally safe** for 40 kg rover loads and 45° slope climbing.

## ♦ Steps to manufacture wheel

- **Process:** Laser/waterjet cut all rings & plates, deburr, then TIG weld spacers or fasten with M4/M5 screws + Loctite 243
- **Jigs:** Maintain concentricity using a Ø250 mm dowel-pin jig, keeping TIR ≤0.2 mm.
- **Hub:** Machine aluminum hub with Ø20 mm bore + 6 mm keyway, and press-fit dual 6004-2RS bearings (H7/g6).
- **Finish:** Shim plates if axial runout exceeds spec; final balance with M4 trim screws, bead-blast and hard-anodize (mask bores/keyway).

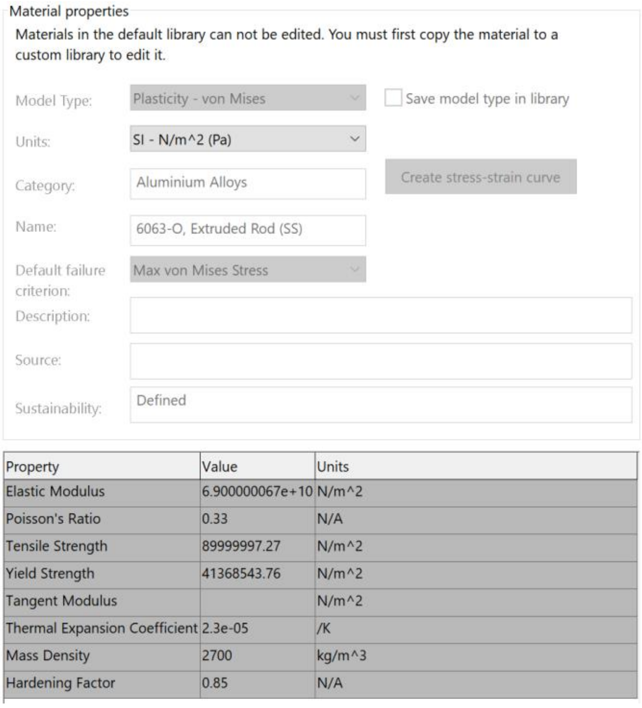



- ♦ Selection of material is an important step in designing of any component The main advantages of material selection are:
  - It increases the reliability of product
  - It reduces the cost of product
  - It can also optimize the weight of product
  - For this design, **Aluminum Alloy (Extrusion grade)** has been selected due to the following reasons
    1. **Lightweight** compared to steel, which reduces overall system mass and improves mobility
    2. **Good machinability and weldability**, making fabrication easier
    3. **Corrosion resistance**, suitable for outdoor and dusty terrain condition
    4. **High strength-to-weight ratio** (Yield strength ≈ 275 MPa, UTS ≈ 310 MPa)



- ♦ The material test data of the aluminum extrusion confirms the above properties. Finite Element Analysis (FEA) performed on the rocker mechanism (with plate thicknesses 3/4/3 mm) showed:
    - **Maximum deformation ≈ 0.0012 mm** under static loading → excellent stiffness
    - **Stress values well below 120 MPa**, ensuring safe design against yielding

Thus, Aluminum 6063-T6 extrusion is an optimal balance between **strength, weight, and manufacturability** for the rover design.




| Property | Value | Units |
|---|---|---|
| Elastic Modulus | 6.900000067e+10 | N/m^2 |
| Poisson's Ratio | 0.33 | N/A |
| Tensile Strength | 89999997.27 | N/m^2 |
| Yield Strength | 41368543.76 | N/m^2 |
| Tangent Modulus | | N/m^2 |
| Thermal Expansion Coefficient | 2.3e-05 | /K |
| Mass Density | 2700 | kg/m^3 |
| Hardening Factor | 0.85 | N/A |

- ## Components to be used:

    - ▪ List of Structure components: Like beams, bushes, shafts, belts, plates, pins, pulleys, wheel, connectors, batteries, motors, etc.

| No. | Hardware details | Subsystem | Category | Quantity Need | Specification | Justification for Chosen Type |
|---|---|---|---|---|---|---|
| 1 | Motor | Roving Mechanism | Planetary Gear motors | 4 | 12v,96rpm,121.7N-cm | Ease of control Availability |
| 2 | Motor | Roving Mechanism | | 4 | Metal gear<br>Stall torque = 18 kg-cm<br>Operating voltage = 4.8v~6.6v | Ease of control Direction |
| 5 | Chassis material | Chassis | Aluminum extrusion | 15 | Profile size: 2020 Extrusion<br>Length:500mm<br>Material: Aluminum Alloy 6063 | Lightweight and strong material and easy constructible |

- List of Motion Components: Like chain, sprockets, flaps, etc.

  a. Differential
  b. Rocker bogie
  c. Bearing

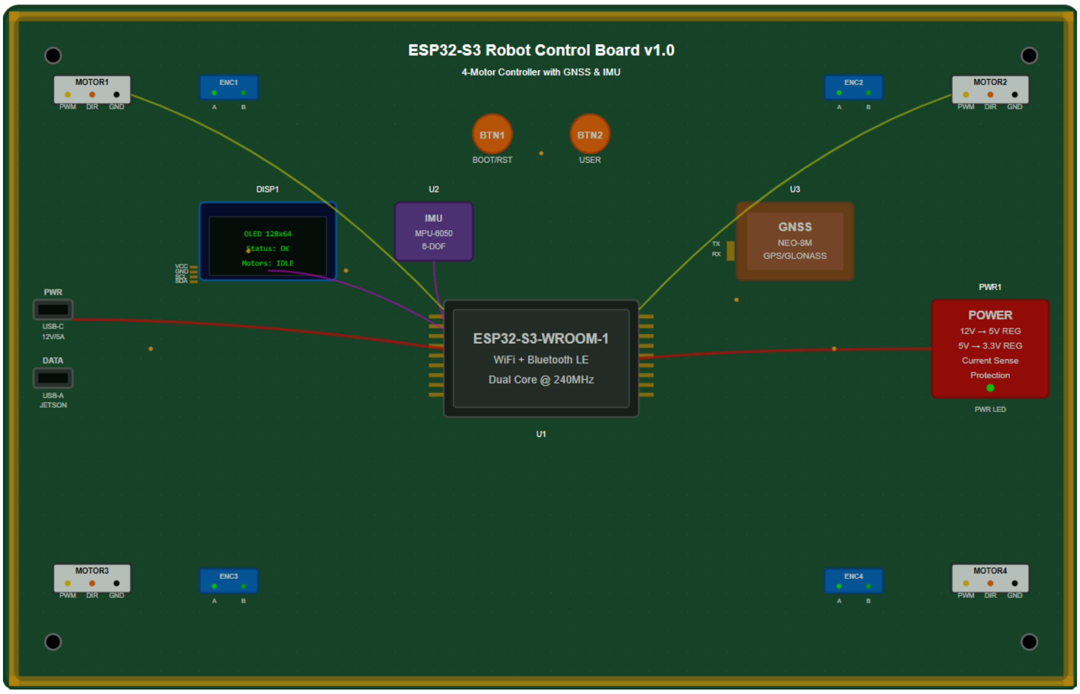
- List of Electronics Components: Like smart ports, switches, joysticks, controllers, LED/LCD screen, power supply, programming components, etc.

| Sr. No. | **Component** | **Qty** | **Price** | **Vendor** |
|---|---|---|---|---|
| 1 | ESP32-S3-DevKitC Development Board | 1 | ₹1,494 | Robu.in |
| 2 | MPU-9250 9-axis IMU Module | 1 | ₹429 | Robu.in |
| 3 | RushFPV GNSS Pro (u-blox M10 GPS) | 1 | ₹1,299 | Quadkart.in |
| 4 | YDLIDAR G2 360° LiDAR (12 m) | 1 | ₹13,124 | Robu.in |
| 5 | Intel RealSense D435i (DepthCam) | 1 | ₹39,783 | Robu.in |
| 6 | 12V DC Geared Motor (250 RPM, 100 N·cm) | 4 | ₹6,657 each | Robu.in |
| 7 | Incremental Rotary Encoder (600 PPR, magnetic) | 4 | ₹3,022 each | Robu.in |
| 8 | Cytron MD10A 10A Motor Driver | 4 | ₹2,500 each | Cytron / Robu (dual channel MDD10A) |
| 9 | NVIDIA Jetson Nano 4GB Module | 1 | ₹13,158 | Robu.in |
| 10 | 7.5″ E-Ink Display (640×384) | 1 | ₹5,459 | Robu.in |
| 11 | Misc electronics (headers, PCB, etc.) | 1 | ₹2,000 | – |

- List of Other Accessories: Clothes, plastic eyes/ear/feeling like real all external components which are for the look.

- **The methodology of Making Robot**: (Please write Entire Technical Specifications of Proposed Robot with brief notes and diagrams) (400 words)

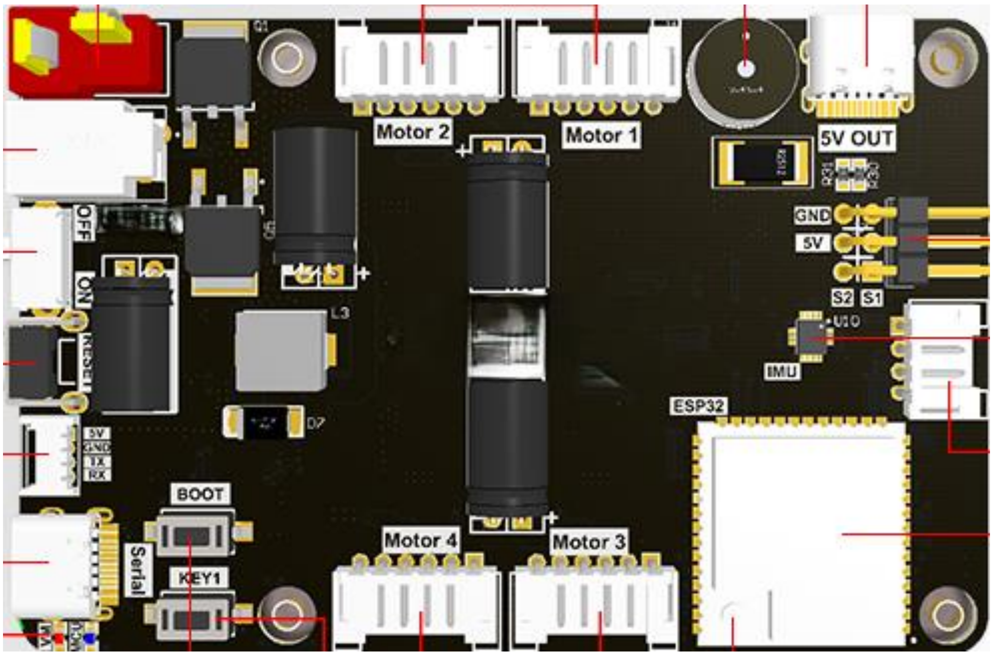
  - ♦ **Custom ESP32-S3 MCU Board**




  - ♦ We will design a compact custom PCB around the ESP32-S3 chip to integrate all low-level electronics. The PCB is approximately 10×5 cm (2-layer) and includes:

    - **Motor Driver Connectors (4×):** Each provides 12 V, two PWM signals, two DIR signals, and ground to drive one MD10A channel.
    - **Encoder Inputs (4×):** Each has VCC, GND, and two signal lines (A, B) for quadrature encoders.
    - **GNSS/UART Header:** A 6-pin connector (VCC, GND, TX, RX, SCL, SDA) for the RushFPV GPS (UART for NMEA, I²C for compass).
    - **I²C Header:** 4-pin header (VCC, GND, SCL, SDA) for the IMU and OLED (shared bus).
    - **Button Inputs (2×):** Two GPIO pins with pull-ups to read user pushbuttons (mode select, emergency stop).
    - **Power & Boot:** A 5 V regulator (buck) feeds the ESP32 and sensors; a connector for the 12 V battery input; jumpers/solder pads select battery charging vs. direct power. The board can also supply 5 V to the Jetson USB power if needed.
    - The board includes an OLED display showing system status (GPS fix, mode, battery voltage) and indicator LEDs for boot/BLE status. The ESP32 firmware runs PID loops for motor velocity control using encoder feedback (tuned for ramp ascent in Stage 3). All sensor inputs are timestamped and sent over UART to the Jetson (to minimize wiring).

- **PCB Layout & Integration**




- The PCB layout was designed for robust routing of power and signals. Motor power traces (12 V) are 1 mm wide copper to handle ~10 A. Logic level traces use 0.25 mm width. We placed the ESP32-S3 module centrally, with peripherals grouped logically: motor headers on one edge, encoders on another, power connectors on top, and the OLED/BLE antenna area clear of interference. All high-speed signals (encoder edges, PWM) have short, shielded paths.

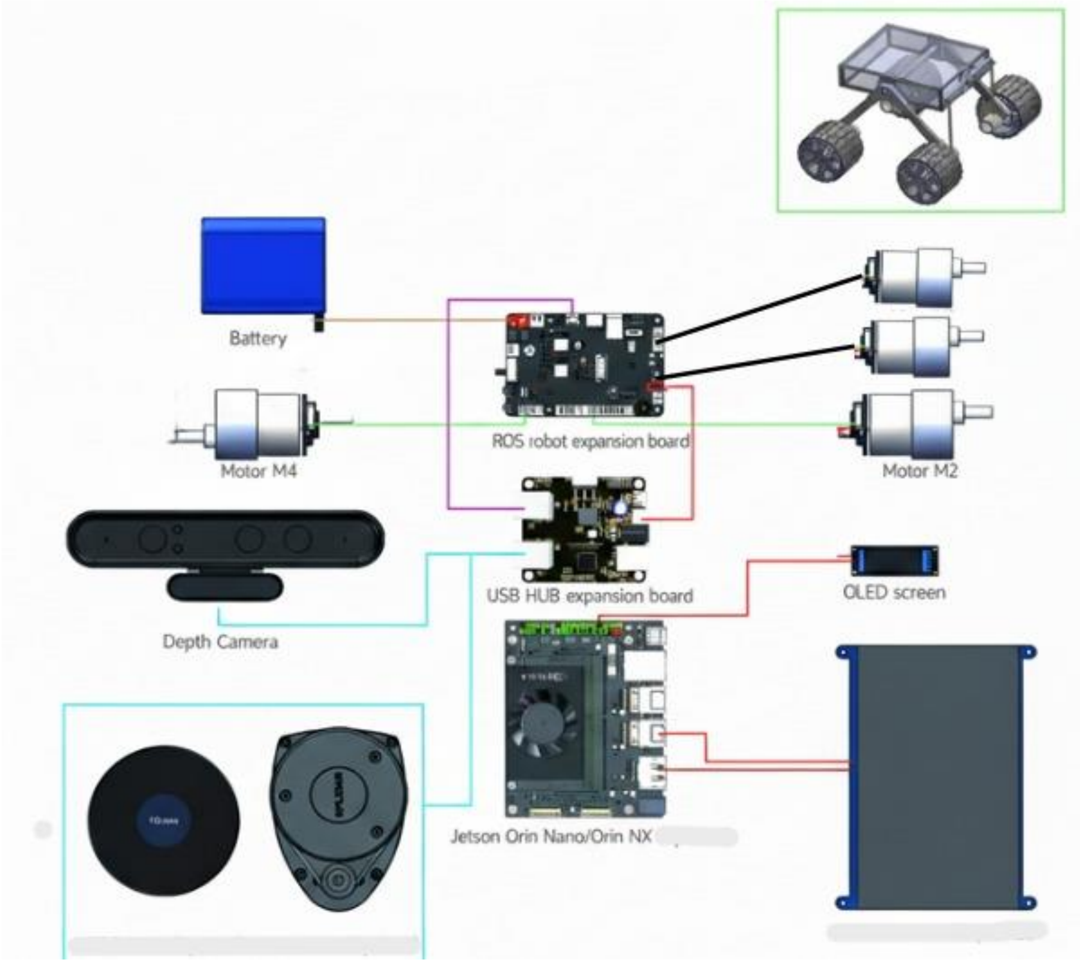
- Pinout summary: The board breaks out the following interfaces to headers/connectors:

  - **Motor Driver 1–4** (5 pins each: PWM1, PWM2, DIR1, DIR2, GND, VIN)
  - **Encoder 1–4** (4 pins each: VCC, GND, A, B)
  - **GNSS** (6 pins: VCC, GND, SCL, SDA, TX, RX)
  - **I²C Bus** (4 pins: VCC, GND, SCL, SDA to IMU/OLED)
  - **Buttons** (2 pins with GND)
  - **Power Input** (12 V, GND) and **5 V Out** for Jetson. This layout allows the Jetson Nano (now Orin Nano) to connect via a short cable to the board's UART and USB ports for data/power.

- Citing the ESP32 board functionality: it indeed "gathers sensor data and runs a motor-control PID," and it "provides connectors for MPU-9250 IMU, GNSS Pro GPS, wheel encoders, motor drivers, an OLED, and push-button inputs. The power section provides 12 V to motors and regulated 5 V to the logic

.

- **Electronics Hardware and Sensor Selection**




- **Custom ESP32-S3 MCU board:** The on-board microcontroller is an ESP32-S3 (chosen for its built-in Wi-Fi, Bluetooth and AI acceleration). It gathers sensor data and runs a motor-control PID. The board provides connectors for:
  - **MPU-9250 IMU (×1):** 9-axis inertial sensor (gyro/accel/magnetometer) with I²C interface (VCC, GND, SCL, SDA).
  - **Rush FPV GNSS Pro (×1):** GPS/GNSS module with u-blox M10 chip (supports GPS, GLONASS, Galileo, BeiDou concurrently at up to 10 Hz). This multi-constellation support ensures robust outdoor positioning. (Its connector has VCC, GND, SCL, SDA, TX, RX.)

- **Wheel encoders (×4):** Each DC motor has an incremental encoder; board hosts 4 connectors (VCC, GND, A, B) to read encoder A/B channels. High-resolution (600 PPR) encoders yield precise odometry.

- **Motor driver headers (×4):** For each wheel, 5-pin connector to Cytron MD10A (GND, PWM1, PWM2, DIR1, DIR2) for bidirectional PWM drive.

- **OLED display:** A small onboard OLED (VCC, GND, SCK, SDA) shows status.

- **User buttons (×2):** Two push-button inputs for mode/control.
- **Power supply:** Onboard DC-DC regulator converts battery 12 V to 5 V for MCU and sensors; also provides unregulated 12 V out to motor drivers (bypassing regulator) as part of the BMS design.
- **Single-board computer:** NVIDIA Jetson Orin Nano 4 GB (supports Linux + ROS2, GPU-accelerated AI). It runs ROS2 and high-level processing.
- **Intel RealSense D435i depth camera (×1):** Provides synchronized depth (to ~10 m) and color images; includes a built-in IMU. The IMU in D435i assists SLAM by giving motion data. USB3 interface to Jetson.
- **YDLIDAR G2 2D LiDAR (×1):** 360° laser scanner, 12 m range, 360° FOV, up to 5 kHz update. Chosen for full-plane obstacle detection with good range and ambient-light resistance 9 . Connects via UART (USB/serial).
- **7.5″ Display (e-ink):** Low-power screen (640×384) for status/GUI.
- **Compass (HMC5883) on GPS module:** Provides heading.

- **Component Justification:**
  - *ESP32-S3:* Integrates dual-core MCU plus Wi-Fi/Bluetooth, enabling wireless debugging/logs and future IoT connectivity. On-chip tensor accelerator could be used for lightweight ML.
  - *Rush GNSS Pro (M10):* Latest u-blox M10 chipset supports multiple GNSS (GPS/GLONASS/Galileo/ BeiDou) and up to 10 Hz fixes, giving improved accuracy and faster updates than older modules.
  - *MPU-9250 IMU:* Established low-cost 9-DoF IMU for inertial data (complementing depth camera IMU).
  - *Intel RealSense D435i:* A high-fidelity RGB-D camera with global shutter and on-board IMU, ideal for outdoor 3D mapping and visual SLAM. It offers up to 1280×720 depth at 90 fps, with wide FOV.
  - *YDLIDAR G2:* Affordable 360° planar LiDAR with 12 m range 9 , suitable for both indoor/outdoor obstacle detection. Its class-1 laser is eye-safe, and robust motor/optics yield stable scans. 360° coverage is useful for localization.
  - Cytron MD10A motor driver: Capable of continuous 10 A per channel, easily driving 12 V high-torque motors.

- In addition to the ESP32 board, the robot uses a **NVIDIA Jetson Orin Nano (8 GB)** single-board computer for high-level processing, replacing the earlier Jetson Nano. The Orin Nano's Ampere GPU delivers up to **67 TOPS** of AI performance (Sparse INT8)[7], vastly outpacing the Nano, enabling real-time neural networks on camera data. Connected to the Jetson (via USB/serial) are: - **Intel RealSense D435i (RGB-D camera):** Provides synchronized depth (to ~10 m) and RGB images, plus a built-in IMU. This will handle visual tasks: line detection, QR decoding, and scene understanding.
    - **YDLIDAR G2 360° (2D LiDAR):** A 360° planar laser scanner (12 m range, up to 5 kHz) for obstacle detection and mapping. Its full horizontal coverage and ambient-light immunity make it ideal for SLAM and obstacle avoidance outdoors.
    - **7.5″ E-Ink Display:** Low-power status display (640×384) mounted on the chassis, driven by the ESP32 (for mode/status readout).
- The D435i's RGB camera will serve double duty: detecting the course line in Stage 1 and reading QR/AR markers in Stage 4 (using on-board neural nets or OpenCV). An additional narrow-field camera could be added for higher-speed line following if needed, but the RealSense already provides a wide-angle color image. A small compass (HMC5883) on the GPS module gives a coarse heading if required.
- Our sensor suite (IMU, GPS, encoders, LiDAR, depth/RGB camera) covers all navigation needs. For example, indoor-outdoor SLAM can fuse LiDAR and vision; outdoors the GPS provides a global fix. The ESP32's on-chip neural engine could also run lightweight ML (e.g. gesture recognition or sensor anomaly detection) ifdesired.

- ## Software Architecture & Data Flow

  - **Data flow:** On ESP32-S3, sensor readings (IMU, wheel encoders, GPS, OLED status) are timestamped and packaged into serial packets. These packets are sent over UART to the Jetson Nano. On the Jetson, a custom ROS2 node (Python/C++) reads UART and publishes sensor topics: /imu/data (sensor_msgs/Imu), /odom (nav_msgs/Odometry from encoders), /gps/fix (NavSatFix), /scan (LaserScan from YDLidar), and /depth/points or /depth/image_raw (PointCloud2 from D435i).

  - **Sensor fusion & TF:** A ROS2 robot_localization EKF node fuses wheel odometry and IMU data to produce smooth odometry (/odometry/filtered) [10] . Static transforms ( /tf_static ) are broadcast from fixed frames: chassis to sensor mounts. The robot_state_publisher (with joint states, if any) publishes all link transforms on /tf (dynamic) and /tf_static (fixed) [11] . This yields a full TF tree (e.g. map  odom  base_link  sensors).

  - **Visualization:** Topics /imu/data , /odom , /scan , and /depth/points can be visualized in RViz2. IMU and odometry give orientation and position, LiDAR scan is shown as 2D obstacles, and depth camera data as pointcloud.

  - **SLAM (2D vs 3D):** Initially, standard 2D SLAM algorithms will be used to build maps:

    - **Cartographer**, **Gmapping**, **slam_toolbox** (ROS2-compatible), which operate on 2D LiDAR ( /scan ). For 3D mapping with the depth camera:

      - **RTAB-Map** (RGB-D SLAM) can use the D435i's depth+RGB+IMU to build 3D maps (pointclouds).

    - **ORB-SLAM2/3** or **LSD-SLAM** (visual SLAM) could use the RealSense images for sparse SLAM (monocular/stereo). These are "vSLAM" approaches mentioned (we would research ORB-SLAM3 for RGB-D use).

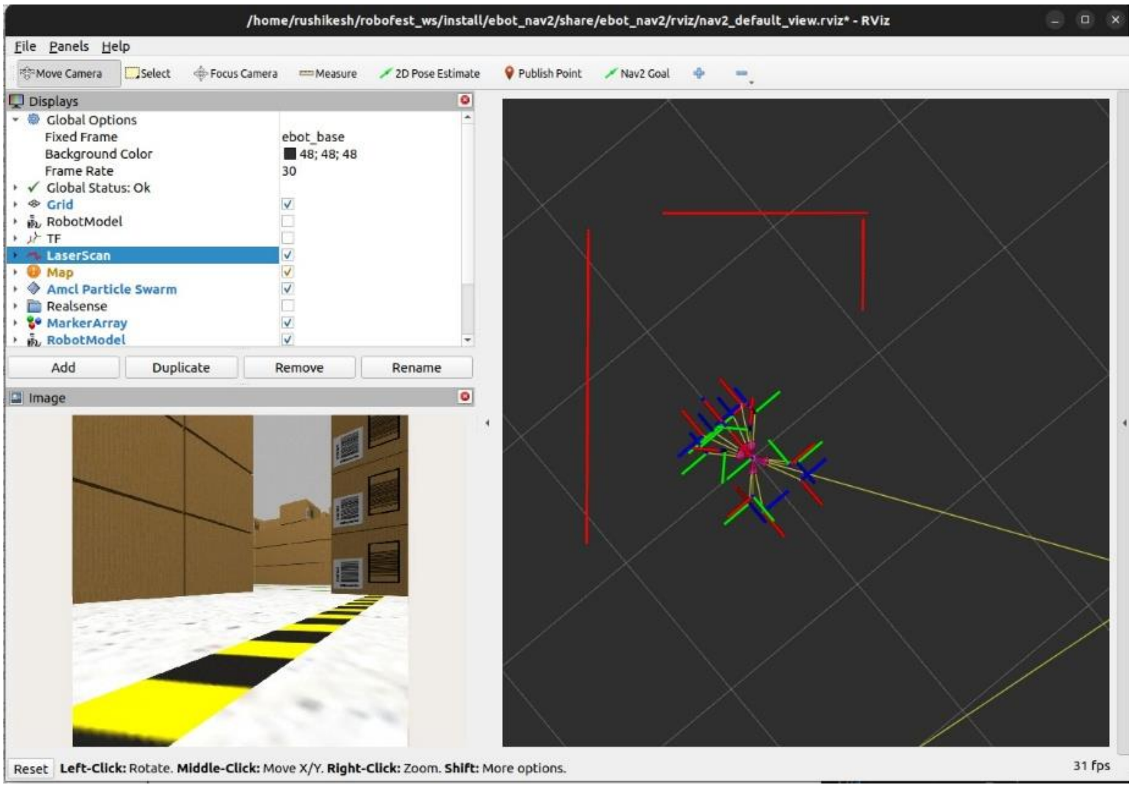
  - **Localization:** During navigation, the fused odom & IMU and recent SLAM map yield a pose (via amcl on 2D or visual-odometry). A navsat_transform node can fuse GPS fixes to correct drift and publish a /fix or a transform to global (geo) coordinates.

  - **Autonomous Navigation:** For 2D nav on a flat map, ROS2 **Nav2** (Navigation Stack) will plan and follow paths using /scan and /odom . For 3D point-cloud navigation (outdoor/open), one could research extensions like Move3D or using OctoMap from /depth/points . (This is an active research area; potential methods include costmaps generated from heightmaps or using the depth data as obstacle warnings).

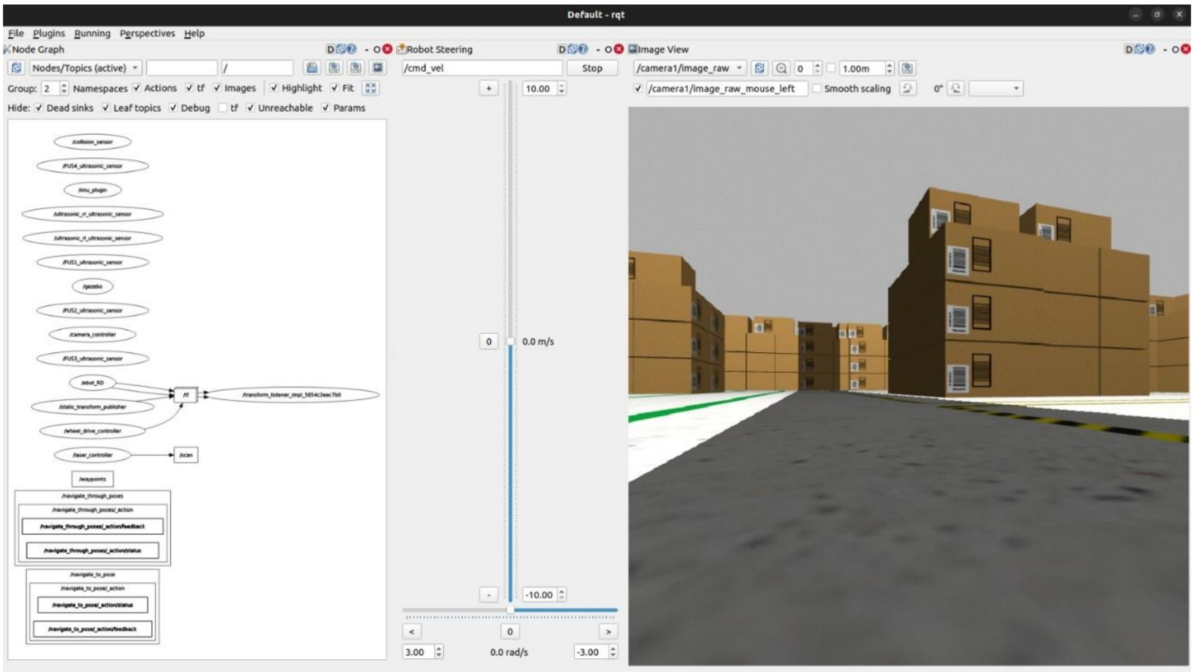
  - **AI/Computer Vision:** The Jetson Nano enables onboard AI: running a neural network (YOLO/ tensorRT) on camera feed to detect obstacles or targets. Machine learning (point-cloud clustering, ground plane segmentation) could enhance obstacle detection beyond raw SLAM. (With ROS2 on Jetson, vision pipelines for object recognition or terrain classification can be integrated.)

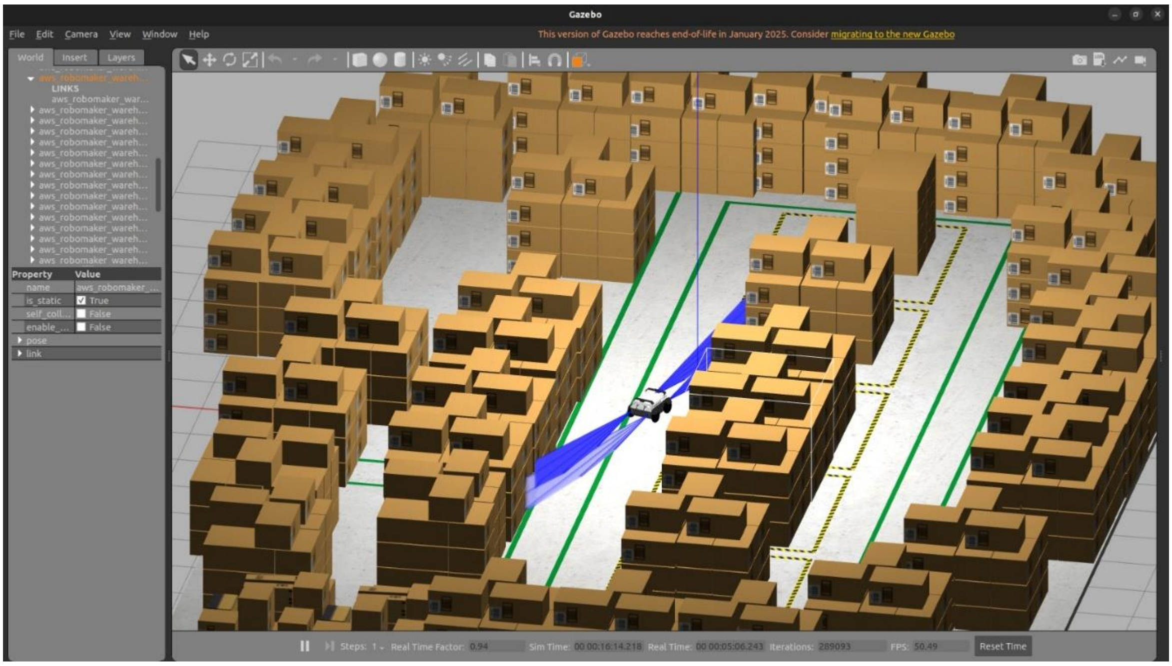
- The data flow in **Manual mode** is direct: joystick commands from a BLE-connected PS5 controller (handled by the ESP32) bypass most autonomy and drive the wheels. In **Semi-autonomous mode**, camera video streams (compressed) and telemetry are sent to a base station, where a user may teleoperate or monitor. In **Autonomous mode**, goal waypoints and courses are fed into Nav2. All modes use the same sensor pipeline, but behavior differs.

- Key SLAM & Nav Choices



- **Robot Visualization on Rviz**




**RQT Node Sensor Graph and Camera View**
**Custom Gazebo World with Robot**

- **SLAM algorithms:**

  - **2D:** Cartographer and Gmapping (2D lidar SLAM), and SlamToolbox (incremental and offline 2D mapping) are candidates for planar mapping.
  - **3D**: RTAB-Map (RGB-D SLAM with loop closure) is a mature choice for depth-camera 3D maps. For purely visual SLAM, ORB-SLAM2/3 or VINS-Mono (monocular + IMU) can be explored.
  - **Nav2*:*** ROS2 Nav2 (the successor of move_base) will be used for standard 2D waypoint navigation on an occupancy map.
  - **Topic outputs:** All sensor outputs (/imu, /odom, /scan, /depth/points, /fix) are published on ROS2 topics. TF frames (mapodombase_link) are broadcast (static transforms from robot_state_publisher ). Rviz2 can then display: sensor data, robot model, and map.
  - **RQt Graphs:** The ROS node graph (RQt) will show nodes/publishers/subscribers for SLAM nodes, publisher of /tf , joint_state_publisher , and Navigation planners. *(Placeholder: RQt graph images for 2D, 3D, localization, Nav2 workflows.)*

  - For mapping and localization, we plan to leverage existing ROS2-compatible SLAM packages: **Cartographer** or **GMapping** (lidar-based 2D SLAM) and **slam_toolbox** (incremental/online mapping) for building a 2D occupancy map. Simultaneously, we can run **RTAB-Map** (RGB-D SLAM) using the D435i's depth+RGB+IMU to build 3D pointcloud maps (with loop closure). Visual SLAM options (ORB-SLAM2/3 or VINS-Mono) could be explored if needed. During navigation, the fused odometry provides the robot's pose, and AMCL (Adaptive Monte Carlo Localization) can localize on the 2D map. A **navsat_transform** node can optionally fuse GPS fixes into the pose to correct drift.

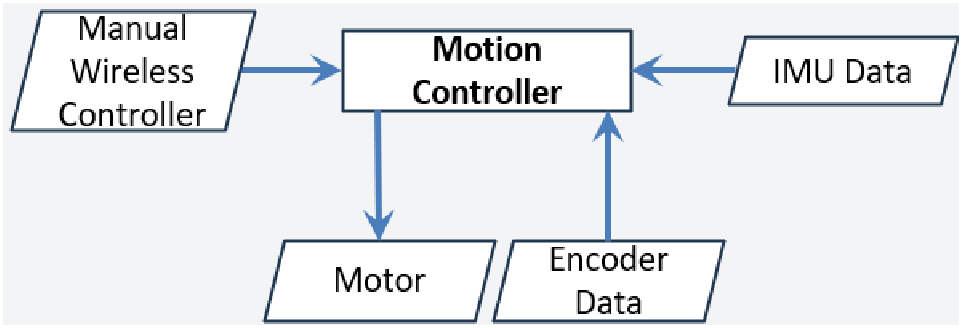
  - For path planning, we use the ROS2 **Nav2 stack**. Given the 2D map, Nav2 will compute and follow paths using laser scans (/scan) and odometry (/odom)[15]. We will configure behavior tree planners for smooth paths around obstacles. Since the course is mostly flat, 2D navigation suffices; for open outdoor segments we may explore OctoMap (3D occupancy) or height-based costmaps to account for rough terrain. In summary, Nav2 handles the core waypoint navigation (covering Stages 1–4), while our higher-level logic handles stage-specific tasks (e.g. stopping for dynamic obstacles).

- **Additional AI / CV Features**

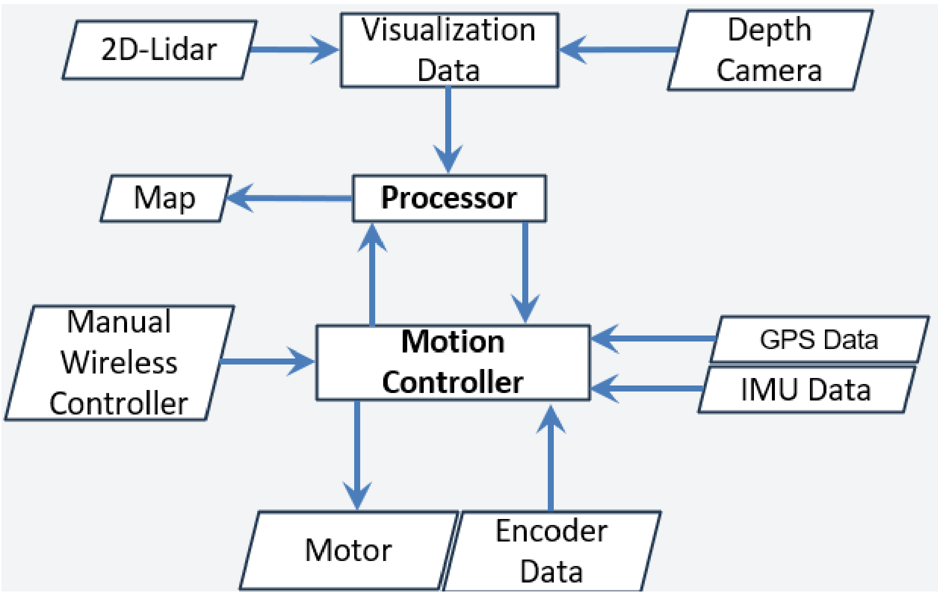
  - **Object Detection:** Using the D435i RGB camera and Jetson GPU, run real-time deep learning (YOLO, MobileNet) for object/person detection. Detected obstacles or targets could be published as extra topics (/detections ).
  - **Terrain Classification:** Apply an ML model to camera or lidar data to classify ground vs obstacles (e.g. via a CNN on image or a point-cloud segmentation network).
  - **Autonomy Enhancement:** AI could identify safe paths or goal objects. TensorRT acceleration on Jetson can run lightweight neural nets without hindering SLAM.

- **Operating Modes**

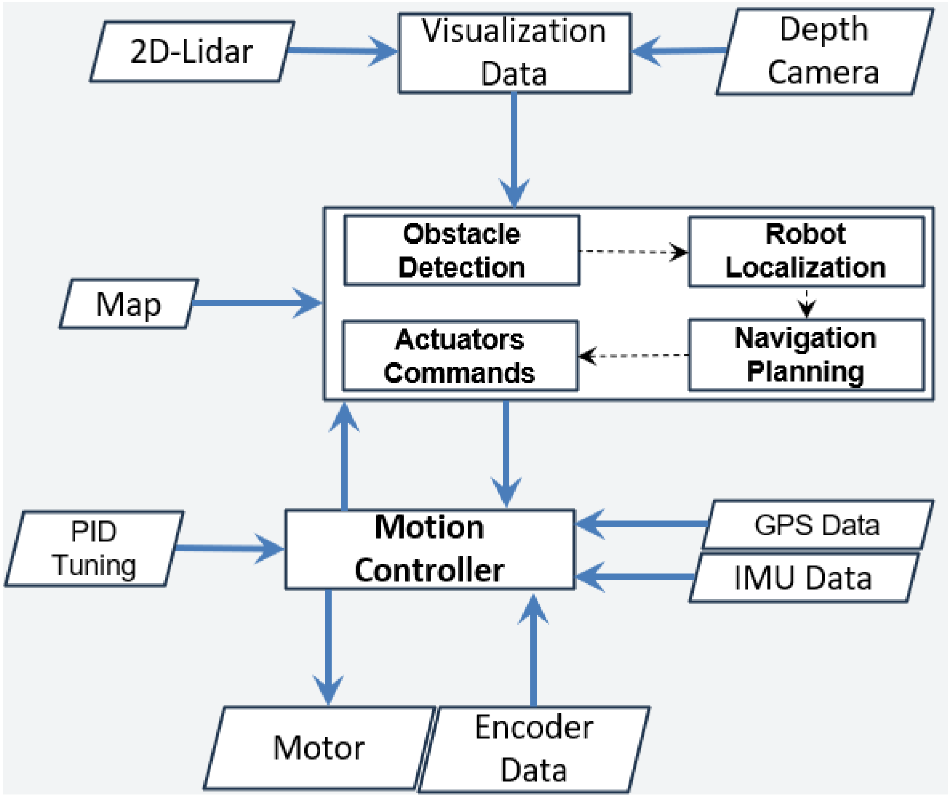
  - **Manual Mode (Remote Teleop):** The robot can be driven entirely by a human operator using a **PS5 DualSense controller** connected via Bluetooth to the ESP32-S3. The ESP32 runs a BLE client that receives controller inputs (joystick axes, buttons) and translates them into motor commands (via PWM/DIR pins). This allows full-speed manual control for debugging or hand-run practice. In this mode, the Jetson runs minimal autonomy (just relaying power/status), and the ESP32 directly controls wheel velocities according to the gamepad.




  - **Semi-Autonomous Mode (Assisted Remote):** The system streams live sensor data (camera view, lidar scan, odometry) to a remote station (via ROS2 topics over Wi-Fi) where a human can monitor progress. The operator can intervene via joystick commands if needed. This is essentially teleoperation with situational awareness. ROS2 topics like /camera/image are published to a remote RViz/GUI. The user can toggle between manual steering or allow autopilot to run, blending modes as necessary.

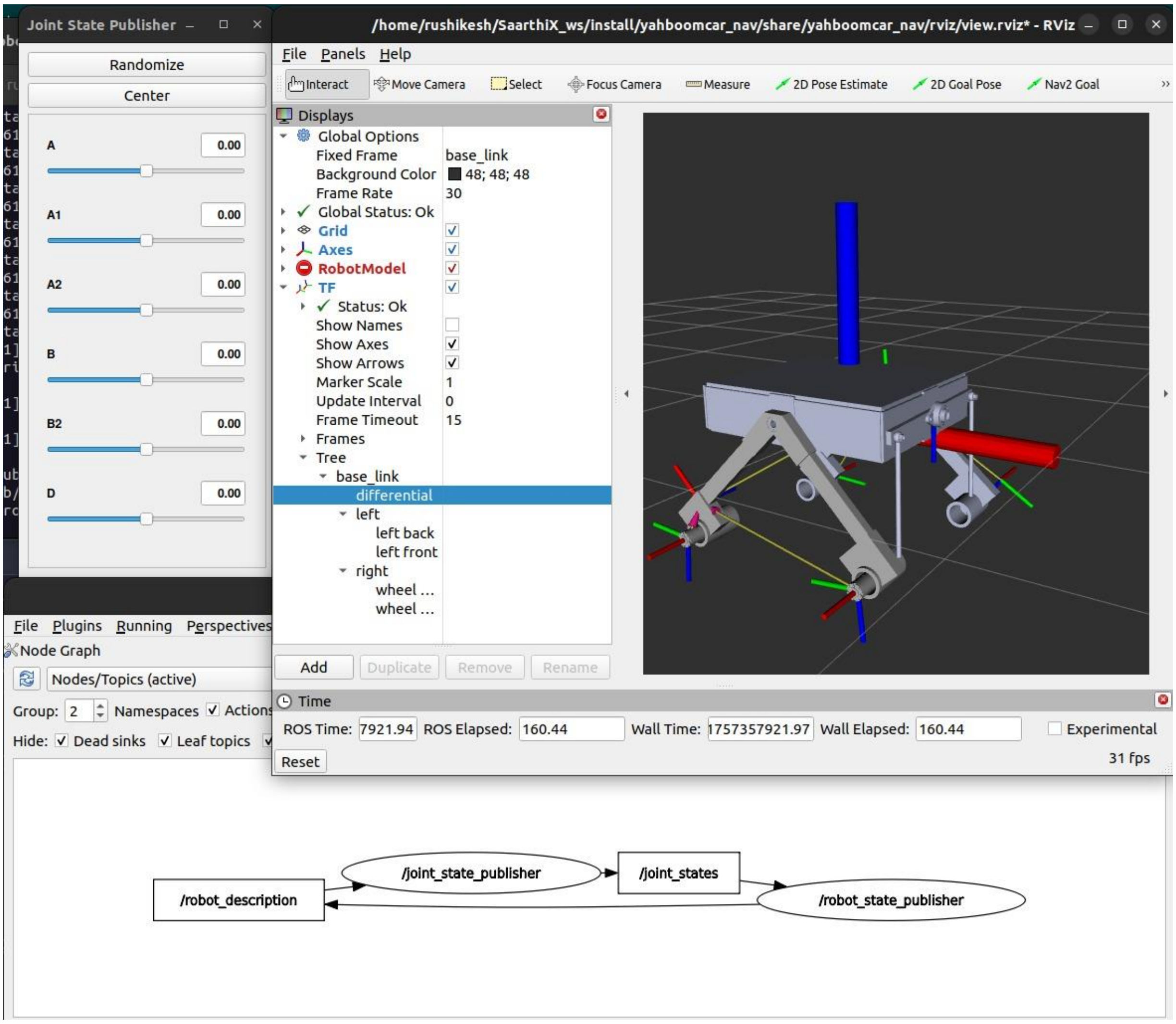



  - **Fully Autonomous Mode:** In full autonomy, the robot uses **Nav2-based navigation** to traverse the course. High-level waypoints (preloaded map corners or auto-generated goals) are sent to Nav2, which plans paths around static obstacles. For each stage, the autonomy behavior is specialized: Stage 1 uses vision to lock onto the line, Stage 2 engages obstacle avoidance if lidar sees an obstacle, etc. Additionally, we implement **Custom Dynamic Obstacle Handling**: if the lidar or camera detects a moving obstacle crossing the path, a ROS2 node commands an immediate stop, monitors until the object clears, then re-issues the goal to resume. Thus, Autonomous mode is Nav2 + stage-specific logic (e.g. stopping logic for the bonus stage).




The data flow in each mode follows ROS2 conventions: sensor drivers → perception/planning nodes → actuator commands. (Diagrams of these workflows would show the sensor topics feeding into ROS2 nodes differently per mode but are omitted here.)

- **Simulation & Digital Twin**



### ♦ Digital Twin of Robot

We employ a Gazebo-based **digital twin** of the vehicle to simulate the entire system before deploying on hardware. The digital model includes realistic dynamics (wheel slip, inertia) and sensor models (simulated RealSense camera and 2D lidar).We launch Gazebo with a custom world (terrain, obstacles, ramps) and run the same ROS2 stack as on the real robot. This allows debugging the navigation and vision pipelines safely. Rviz is run alongside Gazebo to visualize the robot's perception (mapped obstacles, path plans). Using the Gazebo twin, we can test Stage 1 line following (with a virtual floor line), Stage 3 ramp climbing (with an incline model), and the bonus dynamic obstacle (simulating a rolling ball) without risk.

- **Research And Development:** We initially designed a conventional rocker–bogie suspension and tested the rover in semi-autonomous mode. However, instability of the main body caused unstable sensor data, leading to errors in autonomous navigation. Major issues also arose in machining and assembly due to the complexity of rocker–bogie joints. To overcome this, we modified the design and developed a new suspension system that is machining-free, requires only basic tools, and allows quick assembly and disassembly. This improved stability, reduced fabrication effort, and enhanced sensor reliability

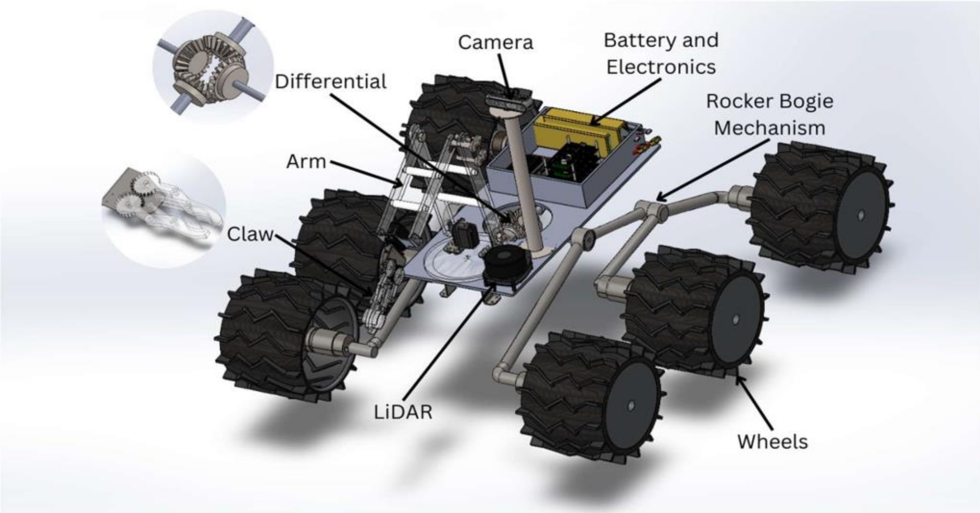
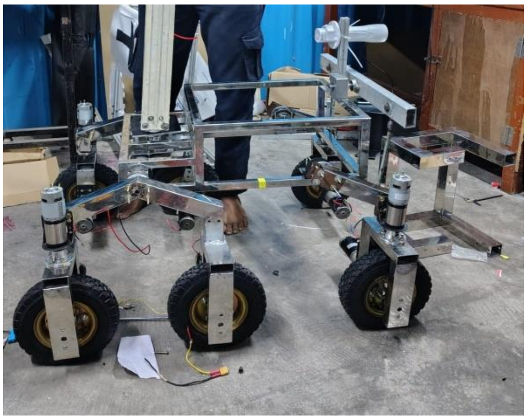



Fig: 1st Rover

We have Arranged the rocker and bogie in the perpendicular manner so this problem can get solve and one more advantage we get the wheels which were going away from the body are now on their own position this was because of only the modified rocker bogie


Fig: Modified Rover

This rover has the capability to traverse terrains with rocks, irregular surfaces, and slopes without transmitting shocks to the main body, ensuring accurate sensor data. We further tested it on rocks with diameters larger than its wheels, muddy surfaces, and for tasks such as soil testing, where the rover was not capable of performing effectively. Therefore, multiple iterations, calculations, and design modifications were carried out, which led to the development of an improved rover system.



This rover was capable of traversing varied terrains, slopes, and conditions without getting stuck. However, during a 360° turn, high stresses were generated on the rover segments, limiting turning speed and response. To address this, we explored several solutions and, through a research paper, studied the Double Rocker Mechanism originally designed for four wheels. After simulations and analysis, we developed a modified Double Rocker Mechanism, which resolved the issue and enabled smooth turning. The simulations also confirmed that the rover could traverse 45° slopes, fully satisfying the GUJCOST guidelines

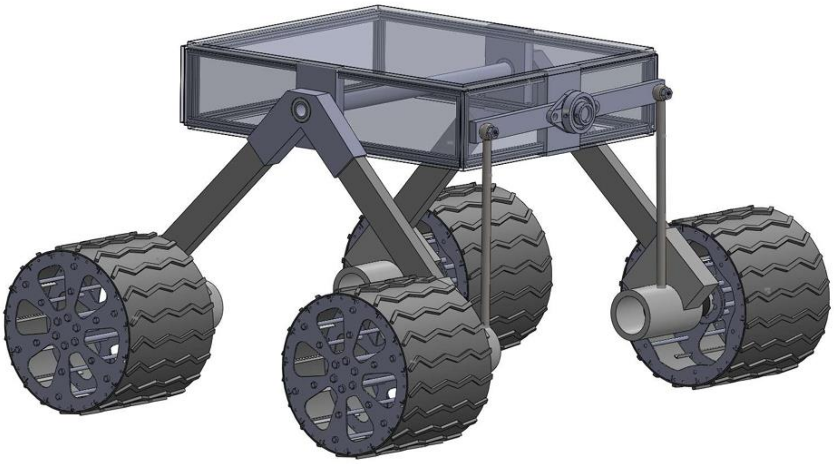


Fig: Final 4 Wheeled Rover

1. Application of proposed Robot in a societal context: (Not more than 100 words)
   he proposed versatile robotic platform addresses critical societal needs. In **disaster management**, it aids in search, rescue, and debris navigation. For **environmental monitoring**, it collects real-time terrain and pollution data. In **agriculture**, it supports precision farming through crop monitoring and automation. Its scope extends to **military and defense**, enabling surveillance and logistics in hazardous zones. For **urban utilities**, it assists in infrastructure inspection and maintenance. In **remote exploration**, it operates in terrains inaccessible to humans, such as mines or extraterrestrial surfaces. Equipped with advanced sensors, GNSS navigation, **double rocker suspension**, and ROS2, it ensures reliable, autonomous, and adaptable performance.

2. Size of robot proposed for Proof of concept (Small version):

   a. Length in 60 cm b. Width in 55cm c. Height in 30cm

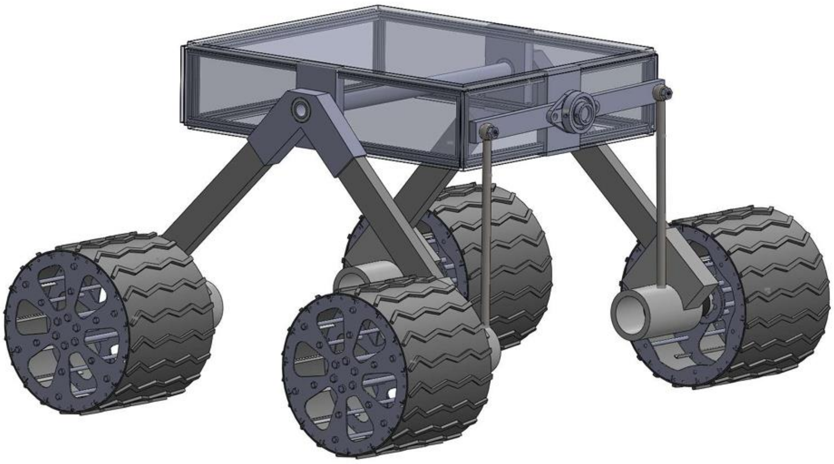
3. Size of robot proposed as Proto type (Actual Version)

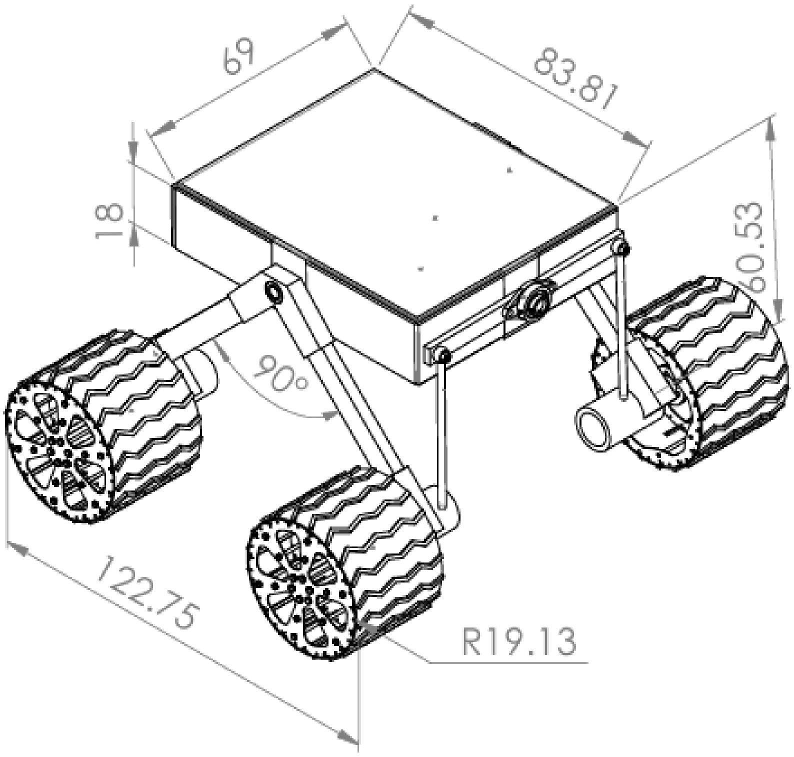
   a. Length in 122.75cm b. Width in 110cm c. Height in 60.cm

4. Timeline for Robot making with milestones. (Divided in activities vs. no. of days)

| Task | Start Week | End Week | Duration (weeks) | Notes |
|---|---|---|---|---|
| Requirements& CAD Design | 1 | 3 | 3 | CAD replication, sensor mounts |
| Chassis Fabrication & Assembly | 4 | 5 | 2 | Cutting, welding, finishing |
| PCB Design &Fabrication | 3 | 4 | 2 | Schematic, PCB layout, prototype fab |
| Electronics Assembly & Testing | 5 | 6 | 2 | Populate PCB, verify connections |
| ESP32 Firmware & ROS2 Interface | 4 | 7 | 4 | UART comms, sensor data formats |
| Sensor Integration (ROS2 nodes) | 6 | 8 | 3 | Publishing /imu, /odom, /scan, / points etc. |
| SLAM & Localization Setup | 7 | 9 | 3 | Test 2D SLAM (Carto/Gmap), RTAB-Map |
| Navigation (Nav2) Deployment | 9 | 10 | 2 | 2D Nav on map, obstacle avoidance |
| Simulation and Validation | 8 | 10 | 3 | Gazebo scenario tests, Rviz demos |

5. Please attach the proposed outline (photograph) for understanding of the evaluation committee.






6. YES I agree to file a patent of the proposed robot in ROBOFEST-GUJARAT 5.0 from the PIC Cell of GUJCOST.

-x-x-x-