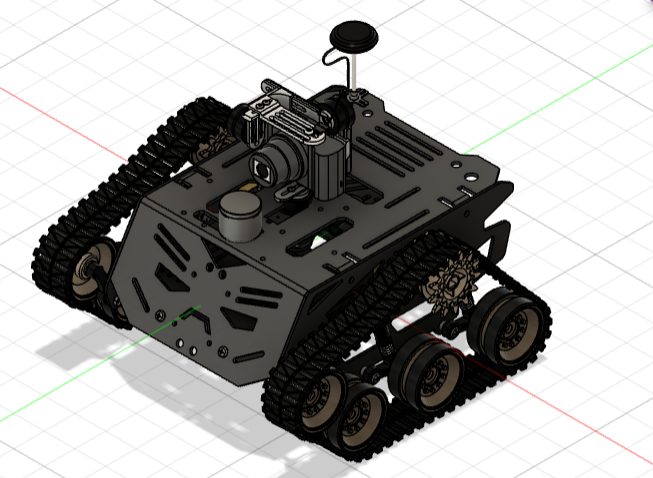
Autonomous Military Rover:

**We are developing a rugged, high-performance rover designed specifically for military applications. This solution features a 360-degree night vision camera, audio feed capabilities, and GNSS location broadcasting with an integrated GNSS navigation system. The rover is capable of reaching any location autonomously through path planning and Computer Vision, making it ideal for gathering information in situations or locations too dangerous for human personnel.**

**By integrating advanced GNSS navigation and camera feed technologies, the rover provides real-time situational awareness and reconnaissance capabilities across diverse terrains and environments. Additionally, it offers wireless control via RF communication along with autonomous capabilities, allowing operators to view the live feed of camera and Location Coordinates, crucial for maintaining situational awareness.**

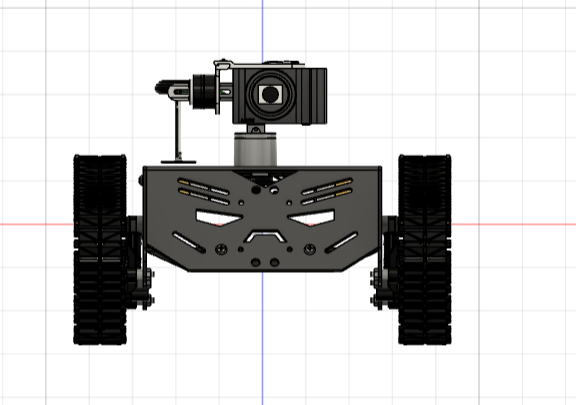
**Furthermore, the rover utilizes computer vision and machine learning for obstacle and object detection and reaching the location safely without any external control. This enables the identification of potential threats like fires, water bodies, and even human presence, enhancing overall environmental awareness.**

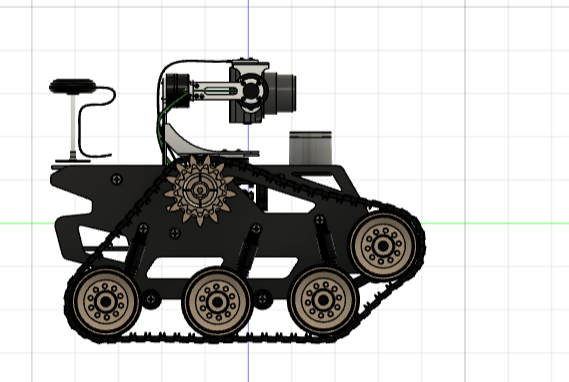
**There will be present a mountable section where mounts can be installed for various applications. The modular design allows for customization like Fire Fighting, Mine Detection, Robotic Arm, Grenade launcher making the rover adaptable for various missions.**



# Robot Assembly Design:

Link to the CAD Model: https://a360.co/3VgOUcQ





# Components to be used:

1. List of Structure components:
2. Rugged Iron Chassis
3. Wheels
4. Belt
5. Batteries
6. List of Motion Components:
7. Chain
8. Suspension
9. List of electronics components:

a. **2-channels ESC:** Motor speed controller.

b. **NVIDIA Jetson Nano (SBC):** AI computing platform.

c. **Pixhawk (Ardupilot):** Autopilot flight controller.

d. **12V DC gear motor x4:** Drive motors.

e. **Battery 18V 6Ah:** Power supply.

f. **DC-DC regulator 5V 10A:** Voltage converter.

g. **75mm GPS pole:** GPS antenna mount.

h. **Microphone Module:** Sound input device.

i. **LoRa Module:** Long-range communication.

j. **RF Module:** Radio frequency communication.

k. **RF controller Remote:** Remote control unit.

l. **Camera Module:** Image capturing device.

m. **Microcontroller:** Control unit.

n. **MPU-6050:** Motion sensing (accelerometer and gyroscope).

o. **Obstacle Avoider Sensor:** Detects obstacles.

p. **LEDs:** Light indicators.

q. **Buzzers:** Audio alerts.

r. **LiDAR Sensor:** Light detection and ranging.

1. List of other Accessories:
2. Mounts
3. Screws and Nuts
4. Cooling Fan
5. Data Storage (SSD/SD Card)
6. Wiring and Connecters

# The methodology of Making Robot:

**Overall Function:** Autonomous reconnaissance and situational awareness rover for military applications using Autonomous capabilities of Deep Learning and Neural networks for path planning and obstacle avoidance. The Wheels of the Rover are tracks that can easy help us move the rover at **45 degrees incline** as the **surface of contact** in case of tracks are way more than round wheels which will help in **more friction** and **glide easily** **through tall grasses** as well because of the **powerful motors driving the Track of the Rover**.

**Dimensions (estimated):**

* Length: 450mm
* Width: 350mm
* Height: 200mm

**Weight (estimated):** 15kg (44 lbs)

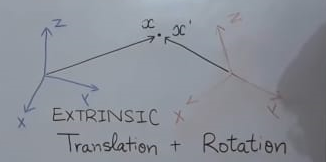
# Structure:

* Material: Rugged Iron Chassis (Provides high durability for harsh environments)
* Tracks : For increased traction in challenging terrains (Improves manoeuvrability)
* Suspension: For Independent suspension system (Improves stability and handling)

# Autonomous Software:

**Theory:**

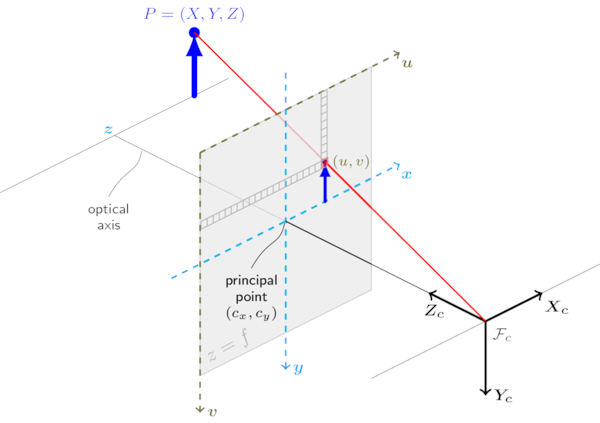
* We’ll have to superimpose the velodyne (LiDAR sensor) and the Camera on top of each other to get one single image.
* As we can see the blue axes identify the LiDAR and the red axes identifies the camera on top, and we’ll be superimposing the LiDAR image on top of the camera image.
* The matrix transformation consists of two parts (translation + rotation).
* X(T) = X + ; X(R) = X
* Both the transformations can be combined into 1 four-dimensional transformation.
* =
* Now that we have our transformed axes where the LiDAR co-ordinates have been superimposed onto the camera axes, we can now take the 3d image and project it onto a 2d image plane.
* We have taken the z-axis as going from our point of view towards the 2d-plane.
* Given a transformation matrix M, we can do the same homogenous transformations as above and get X’’.
* X’’ = MX’
* We’ll divide the x and y-axis with the z-axis to get a point on the 2d-plane.
* r = ; s = ; where (r,s) are arbitrary points on the 2d-plane.
* After the above operations, we can get the LiDAR point clouds and see how the point cloud looks like after we put it on an image which is generated by our camera.



**Building the application:**

**Step 1: Calibration, converting axes, and labelling data**

* We’ll be using the KITTI dataset.
* After reading the calibration file, we’ll extract the camera projection matrix and the transformation matrix from LiDAR to camera coordinates.
* Then converting the LiDAR 3D points to camera coordinates using the provided camera projection matrix and LiDAR-to-camera transformation matrix.
* Then, after reading an image from the disk and the detection labels from a file, it extracts the bounding box coordinates along with the label name.
* We’ll organize the data into a dictionary where each frame contains a list of objects with their bounding box coordinates and IDs.
* We’ll be using vedo for 3D visualizations, and download blobs for the 3d model of a car, and 3d model of an ego vehicle (represents the camera or LiDAR sensor mounted on a vehicle).



**Step 2: Generating bounding boxes**

* We’ll convert the input bounding boxes to a common format (to formats such as OpenPCDet and Waymo).
* Convert the bounding box data into mesh box objects for visualization.
* Generate line-based visualizations for bounding boxes.
* Then we’ll generate transformed box points based on the input bounding box data.

**Step 3: Generate the colours to be used in visualizations**

* Generate a list of random colours based on a specifies colormap, and convert them to RGB format.
* Given a list of object indices, map each object to a colour from the colormap list.
* Map the scatter points to colours based on a specified colormap. The output is a list of RGB colours with an alpha channel.

**Step 4: Visualization**

* We have the bounding box representation and background color.
* Set up lighting with VTK library.
* Set the colour map for objects.
* Add points, spheres, 3D boxes, and 3D cars to the visualization scene.
* Set the camera matrices (intrinsic and extrinsic).
* Add the image to the visualization.
* Set up functions for showing and saving the visualizations.

**Step 5: Additional features**

**i) Detecting water bodies:**

* As the LiDAR will be unable to collect data from a water body, it’ll show up without a point cloud at the area.
* We can divide the 2D image from the LiDAR into an cell grid, and calculate the average RGB colour, any cell having greater than 0 colors, are visitable.

**ii) Detecting hills and mountains:**

* Using the same grid as proposed above, we can get the average inclination of the scenery at any cell.
* Thus any surrounding structure with a steep slope can be detected.

# Mechanism of Motion of the Rover:

The Rover is using the Pixhawk Module. The Pixhawk is an advanced autopilot system used in various autonomous vehicles, including rovers. It provides robust hardware and software support for controlling motors, sensors, and executing complex algorithms mentioned in the Autonomous Software.

# Application of proposed Robot in a societal context:

Beyond military use, this rugged rover's design could be adapted for search and rescue in hazardous areas (fires, collapsed buildings). Its mobility, sensors, and cameras could help locate survivors and transmit real-time data to aid teams. This technology could be valuable for disaster response and civilian applications as well.

1. Size of Robot proposed for Proof of Concept (Small Version):
2. Length in cm: 30
3. Width in cm: 20
4. Height in cm: 20
5. Size of Robot proposed as prototype (Actual Version):
6. Length in cm: 45
7. Width in cm: 30
8. Height in cm: 20

# Timeline for Robot Making with milestones. (Divided in activities Vs. no. of days)

**Phase 1: Design and Development (45 Days)** In this phase, the focus is on designing the rover’s structure, selecting the appropriate components, and developing the initial software. This includes creating the chassis, integrating the motors, and setting up the navigation and control systems.

**Phase 2: Prototyping and Testing (90 Days)** This phase involves building a prototype of the rover and conducting extensive testing. The rover’s performance is evaluated under various conditions, and any issues or shortcomings are identified.

**Phase 3: Refinement and Validation (90 Days)** In the final phase, the feedback from the testing phase is used to refine the rover’s design and software. Further testing is conducted to validate the improvements. The rover is finalized and prepared for deployment.

**Total Estimated Timeline:** 225 Days