IRoC-U

ISRO Robotics Challenge - URSC

Let's build a space robot

Proposal Report

by

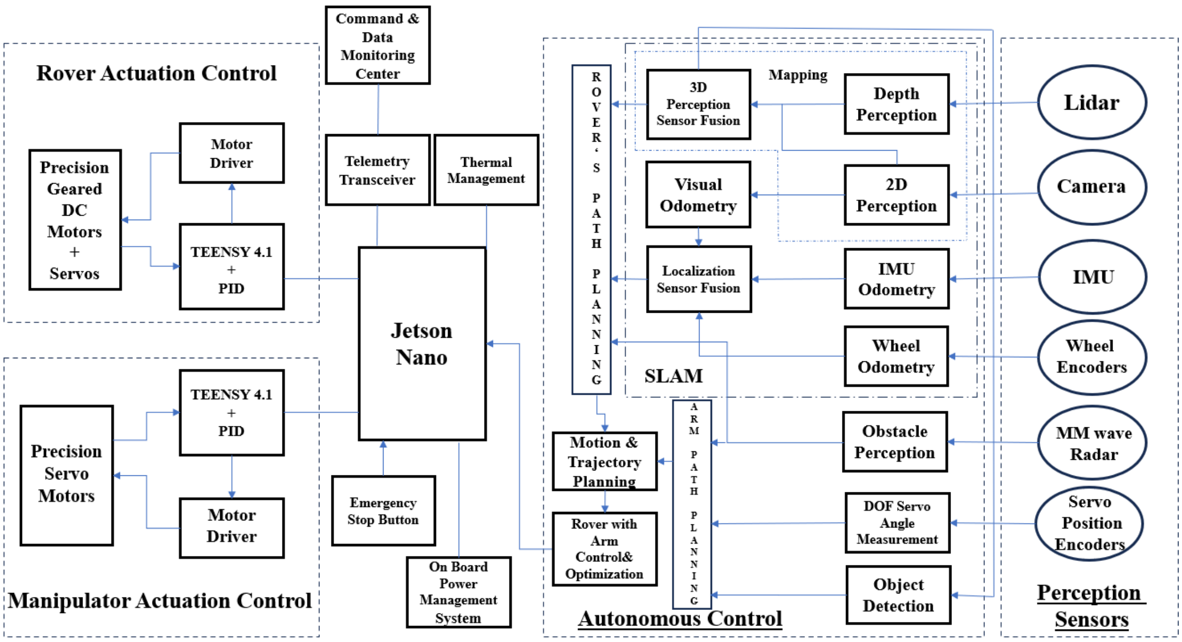
TEAM: ANADI
10089

December 2023

Lovely Professional University

Table of Contents

System Architecture



The proposed schematic uses a jetson nano as the main component which is pivotal with edge computing and artificial intelligence tasks. The primary role will be to process data from onboard sensors. The overview of the processes and components used are:

1. **Rover Actuation Control:** In the actuation control system of our rover, an TINZY assumes a central role in orchestrating the movement of the vehicle. Specifically, the Arduino Mega interfaces with a motor driver, acting as the control hub for precision geared DC motors responsible for driving the rover's wheels. This integration allows for precise and programmable control of motor speed, direction, and torque. The Arduino Mega executes control algorithms that interpret higher-level commands, enabling the rover to navigate, turn, and adjust its speed autonomously. The use of precision geared DC motors enhances the rover's manoeuvrability, ensuring accurate and controlled movements.
2. **Manipulator Actuation Control:** In the Manipulator Actuation Control of our rover, the manipulator arm is equipped with the TINZY, acting as the control unit, collaborating with a PID (Proportional-Integral-Derivative) controller to manage the positioning and motion of the arm. A motor driver interfaces with the Arduino Mega to control precision servo motors embedded in the manipulator arm. The PID controller facilitates feedback control, continuously adjusting the motor inputs to achieve the desired arm positions with accuracy. This integrated system allows for fine-tuned manipulation capabilities, enabling the rover to execute intricate tasks such as picking up objects, adjusting tools, or interacting with its environment. The combination of the Arduino Mega, PID controller, motor driver, and precision servo motors ensures a responsive and reliable manipulator arm, enhancing the rover's versatility in performing a wide range of tasks during its exploratory missions.

3. **Autonomous Control (ACS):**

a. The **ACS of our rover** leverages a multi-sensor approach, integrating Lidar, Camera, Inertial Measurement Unit (IMU), and Wheel Encoders to achieve comprehensive environmental perception and navigation capabilities. The Lidar provides depth perception, enabling the rover to understand the 3D structure of its surroundings. Simultaneously, the Camera captures 2D visual information for scene interpretation. The IMU contributes inertial odometry, while Wheel Encoders provide precise wheel motion data. The system employs sensor fusion techniques to merge the depth information from Lidar with the 2D data from the Camera, creating a robust 3D perception model. Additionally, visual odometry is derived from the 2D images for motion estimation. Further, the system performs localization sensor fusion, combining visual odometry with IMU and wheel odometry data. These integrated sensor inputs form the basis of a Simultaneous Localization and Mapping (SLAM) system, crucial for rover path planning. By employing these sensor fusion techniques, our rover achieves a high level of autonomy, navigating through dynamic environments, and enabling efficient and adaptive path planning.

b. The **ACS for our manipulator arm** employs a sophisticated sensor suite, including MMwave (millimetre-wave radar) radar, servo position encoders, and object detection capabilities, to enhance its perception and decision-making abilities. The MMwave radar serves as a key component for obstacle detection, providing real-time information about the surroundings to prevent collisions and ensure safe manipulation. Servo position encoders contribute to precise measurements of the degrees of freedom (DOF) in the manipulator arm, enabling accurate control and feedback during movement. Object detection technology further refines the system's perception, allowing the manipulator to identify and locate objects within its workspace. By integrating these sensors, the system achieves a comprehensive understanding of its environment, facilitating obstacle avoidance, and informing the path planning of the manipulator arm. This holistic approach ensures that the manipulator arm operates autonomously and efficiently, adapting to dynamic scenarios in a wide range of tasks and environments.

4. **Security Measures:** Our rover is equipped with an Emergency Power System designed to ensure continuous operation and critical functionality in unforeseen circumstances. The mechanism for which will be given later in this report.

The integration of Rover Path Planning and Arm Path Planning data forms the basis for Motion and Trajectory Planning, ensuring cohesive rover and arm coordination. The optimized motion plan, derived from these planning processes, is fed back to the Jetson Nano for real-time execution. The Jetson Nano orchestrates the rover's movements and manipulator arm actions based on the unified plan. This streamlined process enhances the rover's autonomy and adaptability, allowing it to execute tasks efficiently and navigate its environment seamlessly during exploration missions.

Roving Mechanism

The proposed roving mechanism is a compact wheeled platform equipped with a Rocker bogie differential suspension system with manipulator arm, designed for versatile operations in a specified arena with M-sand, sloped surfaces, cubes, craters, and other challenges. With dimensions under 1m x 1m x 0.8m and powered by a standalone battery in a non-GPS environment, the rover prioritizes agility, stability, and efficient traversal. The Rocker bogie mechanism, chosen for its advantages over other options, enhances the rover's ability to navigate slopes with inclinations of at least 15 degrees and overcome obstacles up to 150mm in height

Some of the points considered while selecting this mechanism are,

1. **Ground contact on uneven terrain**: Rocker bogie suspension mechanism contains multiple pivot points, due to these points, wheels maintain continuous ground contact throughout the terrain. This helps the rover in maintaining stability by distributing weight evenly.
2. **Mobility**: Compared to other roving mechanisms, rocker bogie is highly advantageous while traversing over hills, rocks and uneven terrains. By employing this mechanism, a rover can travel over objects which are twice the size of its wheel.
3. **Vibrations damping**: When a rover is travelling over rough and uneven surfaces, it experiences vibrations which can eventually damage electronics onboard the rover. The Rocker bogie mechanism helps in actively dampening these vibrations by allowing each wheel to move independently, this minimizes shock transfer.

Considering the advantages of Rocker bogies mechanism over other mechanisms, this mechanism has been chosen. While designing the Rocker bogie different parameters such as wheel design and dimensions, suspension geometry, load distribution has been considered to optimize the design and ease the travel of rover in M-sand, over cube and through craters. A over body differential has been implemented which connects rocker bogie mechanism on both sides of the rover and balances the entire rover weight.

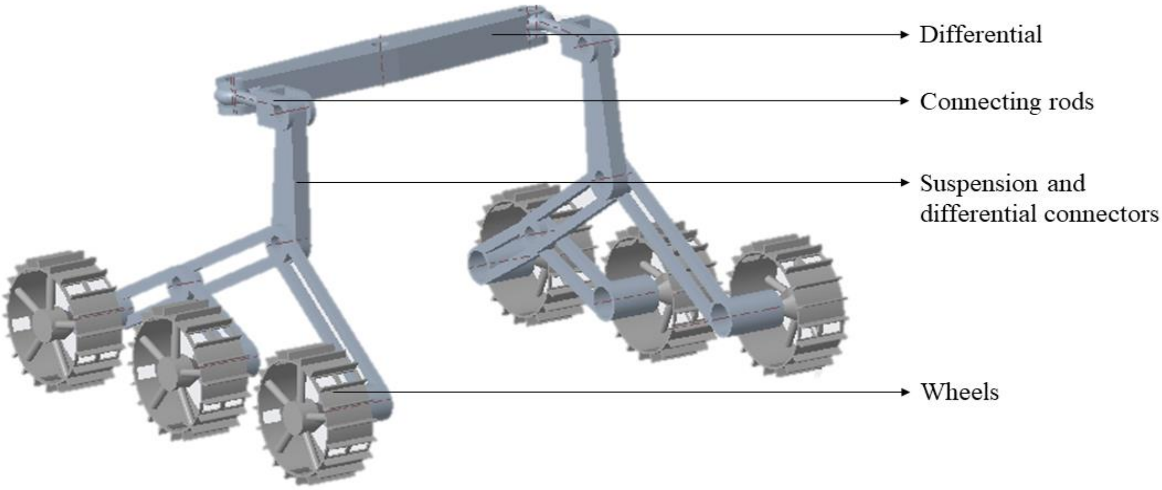


Fig-1: Rocker bogie with differential mechanism

In Fig-1, a rocker bogie connected with 3 wheels on each side can be seen. Each wheel to wheel there is a distance of 250 mm. These rocker bogies are connected to connectors and connecting rods which enable a link with differential. Differential is mounted over the body of the rover and its rotates around its center, thus balancing the center of gravity of rover. The over body differential has a dimension of around 420 mm which is a little extended than the body dimensions which is around 400 mm.

Wheels of the rover has been specially designed, analyzed and tested to travel over uneven terrains like rocks, slopes and sand. As the specifications of the terrain is specifically M sand, the rover wheels need to have the capability to travel even if its wheels are submerged in sand. In particular, sandy terrain, there is a high potential risk that sand and small rocks can enter inside of the wheels and eventually damage them. Just like in the case of NASA's Curiosity and Perseverance rovers. So, in order to tackle these problems, the wheel is designed in such a way that wheel has an increased traction to travel in sandy terrain and holes to remove the sand which enters into the wheel.

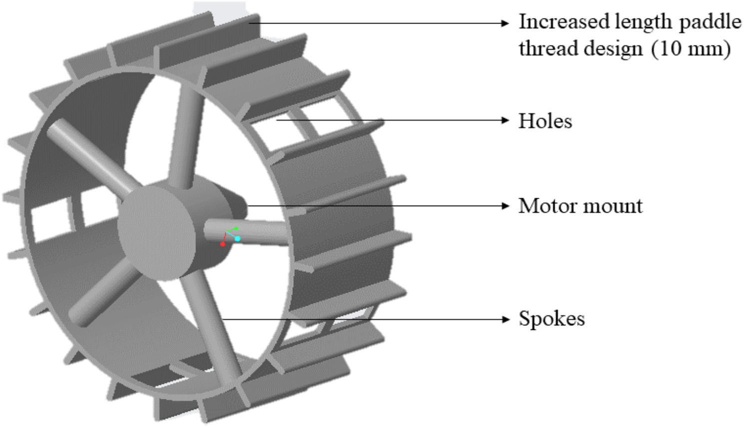


Fig-2: Wheel design

The Overall dimensions of the wheel including the paddle thread design are around 180 mm, where the paddle thread dimensions are 10 mm, diameter of the outer rim of the wheel is 160 mm and diameter of inner rim is around 155 mm. 5 spokes have been provided to give structural strength to the wheel and bare the loads sustained. The width of each wheel is around 60 mm. The design of the wheels has been 3D printed to conduct tests on different terrains and check if the design is practical.

Fig-3: Wheels tested on sandy terrain with slopes.

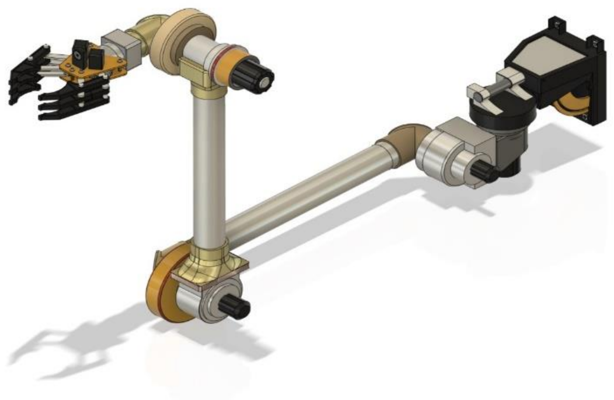
4 wheels have been printed and attached onto 4 motors to a fixed base with no suspension mechanism. Based on the tests conducted and observations made though those tests, it shows that these wheels can provide better performance compared to other wheel designs in sandy and rocky terrain in particular than on smooth surface.

Mechanism for Sample Pick-and-Place Activity

The rover arm plays a crucial role in conducting detailed exploration, sample picking and placing and analysis of the Martian landscape. It is a multifunctional apparatus designed to manipulate and interact with the environment on Mars. Its features include a pick and place mechanism, scientific tools for collecting samples, and high-definition cameras for defined movements and movement points.

As rover arm plays a crucial role in the activity, we followed some parameters to make it more accurate and precision and they are

* FUNCTIONALITY
* RANGE OF MOTION
* DESIGN
* SOFTWARE APPLICATION



Geometrical view of manipulator designed in Fusion 360

The type of manipulator is beneficial on the terrains like M sand where rough movements and heavy tractions, deflections take place. The design of manipulator we chose contains 6 DOF and a horizontal two fingered gripper which is especially cylindrical grooved for precise holding of the defined object of given dimensions.

Components of the Rover Arm

The rover arm comprises a set of articulate joints, enabling it to reach various positions with dexterity and accuracy. It also houses a sophisticated collection of servos and actuators, allowing for real-time feedback and adjustments during operations.

Each component is carefully selected to withstand the harsh conditions of the Martian terrain demands.

Steps for Picking Up an Object

*Approach:

The rover arm manoeuvres into position, aligning the gripper with the targeted sample under the guidance of visual feedback**.**

*Gripping:

The gripper delicately secures the object, r`**.**

*Retraction:

Once the object is secured, the arm retracts, allowing for further examination without interference from the gripping mechanism.

Steps for Placing an Object

*Approach:

The rover arm navigates to the designated placement area, using visual sensors to position the gripper accurately.

* Release:

The gripper gently releases the object, monitoring its placement to ensure it aligns precisely with the targeted location.

*Confirmation:

Upon successful release, the arm verifies the object's stability and completes the assignment.

Inverse Kinematics:

In our robotic arm we are using inverse kinematics because we should determine the angles of 6 degree of freedom manipulator, if the angles are determined we conclude the movement of manipulator.

Why I choose Inverse Kinematics: Because we need to know about angles for movement of manipulator.

Formulae used for above arm

Terms

A = a = Links (ex-link 1, link 2….)

P = Distance between manipulator Base to End effector

W = Distance between ground to 4th Link

R = Orientation

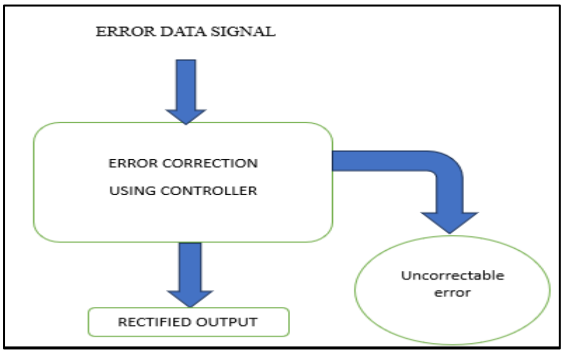
Inverse Dynamics:

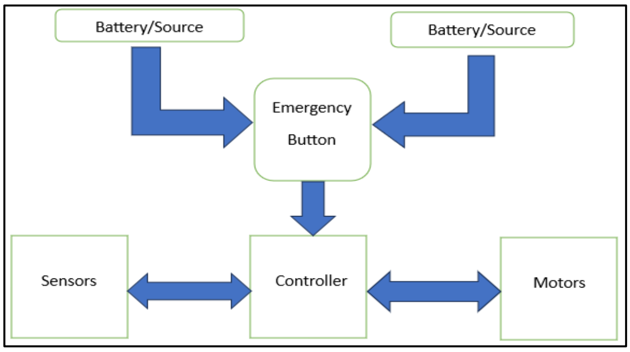
In inverse Dynamics we determined torque generated by using mass, centre of mass, gravity and force. M= mass matrix= g

Challenges: The rover arm's performance is subject to the extreme rough terrain conditions of M sand, including differential point loads and Bending Moments. Ensuring reliable operation under such conditions requires continuous updating, maintenance, and innovative engineering is done.

Emergency Response System

Emergency response system is desined along with a push button for stopping manually on a bot or using the electrically by some of the faulty conditions given below





	Problem statement	**Remedy**	**Terminal action**
BATTERIES PROBLEM (batteries)	Battery Overcharging and Over-discharging can cause serious consequences as it can cause a risk of high potential to the working and functionality of the whole rover.	* BMS. * Apply a proper set of algorithms provide most appropriate charging procedure.	If the battery got damaged eventually, it will be cut-off, and backup battery come into action.
OVERHEATING (motors, Powerlines, microcontrollers)	Overheating can occur in space rovers due to prolong exposure of rover to harsh environment, motor overloading, increasing temperature due to faulty mechanism design.	*Generate an interrupt to the microcontroller which will shut down or will put the whole rover in sleep mode.	*When software fails to deal with overheating manual emergency button is used.
COMMUNICATION FAILURE (transmitter,receiver, micro-controller, antenna)	A communication failure due to any reason or an over extension for re-establishment of communication, it can cause an inability to give command and data transmission to rover by mission control.	* The rover will stop its operation and go into stand-by mode. * For internal system failures, then whole rover will shut-down Diagnosis system will do internal check for recovery.	*If the re-establishment period is crossed then all systems of rover will be shut downed. *Communication system will work on least battery consumption mode.
UNEXPECTED TERRAIN OR OTHER SAFETY CONCERS (actuation	Rover can face issues like unexpected terrain where a terrain is detected by sensors which can cause tipping over or getting struck, collision risks with big obstacles like boulders, etc.	*Below the setpoint it'll cross the obstacle without avoiding,	*Above the set point it'll avoid the obstacle like pits or mountainous. Rover movement system be powered off or will be kept at halt if it keeps going towards it.

motors)			
SENSOR FAILURE (sensors)	Sensors are used to collect data of mostly every part of the rover, traction, path, as well as the environment around it. Failure of them can will harm rover functioning and give wrong data. The failure can occurs due to mechanical damage, thermal stress, radiation exposure, dust and contamination, etc.	*Sensor failure can be detected by using redundant sensors and comparing the data of both. *Sensors can also do an internal Diagnostics and provide reports of their health. *Based on these reports the power management system will disconnect the target sensor.	*If the fundamental sensors failed, the whole power of the rover will be switched down. *Communication system will send the diagnostic report to the mission control and will wait for the mission controls decision.
SOFTWARE FAILURE (micro-controller)	software like OS, Navigation software, communication software, data processing software, etc work together to carry out the whole space mission. These are crucial part for rover. Failure of them means failure of everything.	*Instrument calibration as well as diagnostic test will keep the software in check and will send data to mission control continuously.	*If any software fails the diagnostic test, the rover will enter in a safe mode and the whole power will shut down. *Rover will contact mission control and wait. Till the time, the rover will be in a partial sleep mode.
UNCONTROLLABLE SPEED AND DIRECTION	Uncontrollability can occur in rover due to software glitches, hardware malfunctions, power system failure, etc.	PID can be used to detect traction deviation as well as speed control.	If PID control system fails to control manual emergency shutdown is preferred

Hardware Identification:

No.	Hardware Details	Subsystem	Category	Quantity Needed	Justification for Chosen Type
1	Precision Geared DC Motor	Rover Actuation Control	Planetary Geard Servo	6	High torque, precise control
2	Precision Servo motors	Manipulator Actuation Control	Servos	4	Angular position precision

3	Motor Drivers with Feedback	Actuation Control & Perception Sensor	H-bridge Drivers with feedback	6+4	H-bridge, precise control, feedback
4	Rover Chassis and Suspension	Mechanical & Structures	Wheeled Rocker Bogie Differential	1	Rocker bogie, stable mobility
5	Wheels	Mechanical & Structures	Customised Wheels	6	Customized for traction, durability
6	Manipulator Structure with gripper	Mechanical & Structures	Aluminium Fabricated	1	Aluminium structure, versatile gripper
7	Rp02 Lidar	Perception Sensor	Mechanical 2D Lidar	1	2D Lidar, accurate perception
8	RGBD Camera	Perception Sensor	TOF Based	1	TOF-based depth sensing
9	IMU sensor (9Axis)	Perception Sensor	Mems Based Inertial Measurement	1	MEMS-based stability, orientation
10	MM wave Radar sensor	Perception Sensor	MEMS Device	1	Long-range object detection
11	Jetson nano	On Board Computation	Gpu Based Computer	1	GPU-based onboard computation
12	Controllers	On Board Computation Control	Controller Board	2	Versatile, programmable control boards
13	Power Distribution and Management Board	Power Management Subsystem	Power Electronics Circuitry	1	Efficient power distribution, management
14	24v LI ion Battery of 20AH	Power Management Subsystem	LI-ION	1	Reliable, rechargeable power source
15	3DR Radio Telemetry 433MHz	Communication Subsystem	Telemetry Control	1	Wireless communication, telemetry control
16	Miscellaneous	-	-	-	Additional, components

1. Precision Geared DC Motor:

The chosen precision geared DC motors are sourced from the market, offering a combination of high torque, speed, voltage, and current capabilities. These motors are crucial for rover actuation, providing the necessary power and precision required for controlled mobility in various terrains.

2. Precision Servo Motors:

Sourced from the market, these precision servo motors are selected for their accuracy and responsiveness in angular position control. They deliver the required torque, speed, voltage, and current, making them ideal for precise manipulative tasks within the robotic system.

3. Motor Drivers with Feedback:

The motor drivers with feedback are procured from the market and are essential for controlling both DC motors and servos. Utilizing H-bridge technology, these drivers ensure bidirectional control and incorporate feedback mechanisms to enhance precision and real-time monitoring of motor performance.

4. Rover Chassis and Suspension:

The rover chassis is fabricated from aluminium, combining strength with a lightweight design. The suspension system, crucial for stability and mobility, is tailored to handle the rover's mass effectively. The aluminium fabrication ensures durability and structural integrity in challenging environments**.**

5. Wheels:

The wheels are customized and manufactured through 3D printing, tailored to provide optimal traction and durability. Designed to distribute mass effectively, these wheels enhance the rover's ability to traverse different terrains, contributing to overall mobility and performance.

6. Manipulator Structure with Gripper:

The manipulator structure is a combination of 3D printed and aluminium-fabricated components. This hybrid design ensures strength and versatility in the manipulator's construction. The gripper, an integral part of the structure, is engineered for precise and adaptable handling of objects, adding functionality to the robotic system**.**

7. Rp02 Lidar A2M12:

The Rp02 Lidar A2M12 is sourced from the market and plays a pivotal role in perception sensing. This 2D Lidar boasts a scanning range of 18m and a detection range of 12m, coupled with a high sample rate of 16000x. With a 360-degree field of view, it provides accurate and comprehensive environmental data, contributing to the robust perception capabilities of the robotic system.

8. RGBD Camera (Intel RealSense D435i):

The RGBD camera, specifically the Intel RealSense D435i, is procured from the market. With a range of 0.3-3 meters for depth estimation, this camera enhances the vision capabilities of the robotic system. It incorporates advanced RealSense technology, providing accurate and reliable depth perception for various applications.

9. IMU Sensor (10Axis) GY-91:

The GY-91 IMU sensor, sourced from the market, integrates MPU-9250 and BMP280 technologies. Operating within a voltage range of 3.0-5.0V and featuring a 16-bit ADC, this 10-axis sensor is essential for maintaining stability and tracking orientation, contributing to precise motion sensing within the robotic system.

10. MM Wave Radar Sensor:

The MM Wave Radar Sensor is procured from the market and operates at 24GHz with an impressive range of 20 meters. Leveraging MEMS technology, it provides long-range object detection capabilities, enhancing the robotic system's awareness and safety in its surroundings**.**

11. Jetson Nano:

The Jetson Nano, sourced from the market, serves as the onboard computation unit for the robotic system. Equipped with a quad-core ARM A57 processor running at 1.43GHz and a 128-core Maxwell GPU, it enables high-performance computation, making it well-suited for complex tasks and data processing.

12. Teensy 4.1 Controllers:

The Teensy 4.1 controllers, procured from the market, feature an ARM Cortex-M7 processor operating at 600MHz. With 1024k RAM and 55 total IO ports each, these controllers provide a powerful and versatile platform for real-time control and customization within the robotic system.These components collectively form a sophisticated hardware architecture, combining precision, adaptability, and advanced sensing capabilities to create a versatile and capable robotic system.

13. Power Distribution and Management Board:

The Power Distribution and Management Board is sourced both from the market and through customization. This crucial component oversees the efficient distribution of power throughout the robotic system. Tailored circuitry ensures optimal power consumption rates, contributing to the overall energy efficiency and reliability of the system.

14. 24V Li-Ion Battery:

The 24V Li-Ion Battery, procured from the market, serves as the primary power source for the robotic system. With a capacity of 20AH, operating at 24V, and capable of delivering the required current, these batteries provide a reliable and rechargeable power solution, ensuring sustained operation during missions or tasks.

15. Telemetry Transceiver:

The Telemetry Transceiver, sourced from the market, operates at 433MHz. This component facilitates wireless communication and telemetry control, allowing for remote monitoring and control of the robotic system. It plays a crucial role in establishing communication links for real-time data transmission.

16. Miscellaneous:

The Miscellaneous category encompasses additional components sourced based on specific project requirements. These components, amounting to an estimated cost of ₹10,000, provide flexibility and adaptability to the robotic system. They may include sensors, actuators, or modules tailored to unique use cases and project needs.

This comprehensive hardware setup integrates advanced technologies, precision engineering, and thoughtful customization to create a robotic system capable of handling diverse tasks with reliability and efficiency. The combination of mobility, perception, computation, and control components forms a cohesive and versatile platform for various applications in robotics

Software Identification

```
# Algorithm for Rover Navigation with ROS 2
# Initialization
initialize_sensors()  # Initialize Lidar, RGBD camera, IMU, Encoders
initialize_perception()  # Initialize perception components (YOLO, VL SLAM, etc.)
initialize_navigation()  # Initialize navigation components (Path planning, obstacle avoidance, etc.)
# Perception Loop
while not rospy.is_shutdown():
    lidar_data = read_lidar_data()
    rgb_image, depth_map = read_rgbd_data()
    imu_data = read_imu_data()
    encoder_data = read_encoder_data()
    # Object Detection using YOLO
    detected_objects = detect_objects(rgb_image)
    # SLAM using VL SLAM
    pose_estimation = run_vl_slam(lidar_data, rgb_image)
    # Sensor Fusion
    fused_data = fuse_sensor_data(lidar_data, rgb_image, depth_map, imu_data, encoder_data)
    # Task-Specific Processing
    process_detected_objects(detected_objects)
    process_slam_data(pose_estimation)
    process_fused_sensor_data(fused_data)
# Navigation Loop
while not rospy.is_shutdown():
    # Path Planning
    waypoint_path = plan_path_to_waypoint()
    # Autonomous Navigation
    navigate_autonomously(waypoint_path)
    # Obstacle Avoidance
    avoid_obstacles()
    # Sample Pickup and Drop
    if reached_waypoint():
        identify_and_pickup_sample()
        transport_sample_to_drop_position()
        identify_sample_drop_position()
        autonomously_transfer_sample()
    # Autonomous Final Positioning
    if sample_dropped():
        exit_from_final_position()
        position_nearer_to_final_position()
```

Along with Autonomous navigation of the rover and Manipulator arm that must fulfil the required sequence of processes mentioned in the challenge that should traverse and perform Automated pickup and to navigate to destination to place the payload. Autonomous Navigation will be done in three connected phases.

1. SLAM (Simultaneous Localization and Mapping)
2. Path Planning & Optimization
3. Motion and trajectory planning

Description Type	**SENSOR Fusion**	**SLAM**	**Path Planning**

Purpose	Combines data from different sensors to provide more accurate, reliable and comprehensive understanding of the system.	Used by robots to simultaneously create a map of an unknown environment while locating themselves in it.	Path planning for a rover which has to navigate through diverse terrain involves determining the optimal route from starting point to destination and deciding which obstacle to avoid and which to traverse.
Function	Integrates information from different sensors to get accurate parameters from the required data format	Involves a robot to construct a map of the environment using sensor data such as 3d point clouds, meshing and soon	Main function is to generate optimal path using best algorithm that suites our problem statement, path optimization and real-time adaptation by constantly updating the path based on real-time sensor data.
Process	Filter raw sensor data from different sensors and align this data in a common coordinate frame, then combine this information from different sensors using various techniques like Kalman filters, giving a unified representation of the environment that incorporates the data from different sensors into one.	Collect data from sensors like LiDAR, use this data to identify the features of the environment. Then associate these features with sensor measurements to track their location over time and determine the position of the robot based on this map information and update the map continuously using the new sensor information as the robot moves.	First of all, choose an appropriate algorithm based on environment and rover capabilities, then implement the selected algorithm to calculate an initial path from starting point to final point. Optimize the path considering factors like efficiency and minimum travel time. At the end, continuously update the planned path based real-time sensor data.

Path Planning & Optimization

Path planning is used to select the shortest distance of the path from the starting point to the target point. According to the degree of environmental freedom the method can be classified as two types: Global path planning and Local path planning

Description Type	**Global Path planning**	**Local Path planning**
Description	High-level planning for finding an optimal path from start to destination in the entire environment	Low-level planning for navigating around immediate obstacles in the robot's proximity
ALGOS	Dijkstra's Algorithm, A* Algorithm, Visibility, Graphs, Probabilistic Roadmaps (PRM)	Dynamic Window Approach, Velocity Obstacle, Reactive Approaches (e.g., Bug Algorithms)

Hardware and Software Realization Plan

No.	**Hardware Details**	**Procurement Source (Market/Fabrication/3D Printing/…..)**	**Specifications/ Realization Plan**	**Quantity**	**Estimated Cost**

1	Precision Geared DC Motor	Market	Torque, Speed, Voltage, Current	6	24000
2	Precision Servo motors	Market	Torque, Speed, Voltage, Current	4	12000
3	Motor Drivers with Feedback	Market	Torque, Speed, Voltage, Current	6+4	10000
4	Rover Chassis and Suspension	Aluminium fabrication	Strength, mass	1	10000
5	Wheels	3D printed	Strength, mass		5000
6	Manipulator with gripper	3D Printing and Aluminium fabrication	Strength mass	1	10000
7	Rp02 Lidar A2M12	Market	Scanning-18m, Detection-12m,16000x sample rate,360degree	1	30000
8	RGBD Camera (Intel Real sense D435i)	Market	0.3-3meters, Depth estimation	1	41000
9	IMU sensor (10Axis) GY-91	Market	MPU-9250 + BMP280,3.0-5.0v,16bit ADC	1	1000
10	MM wave Radar sensor	Market	24GHz,20meters	1	8000
11	Jetson nano	Market	Quadcore ARM A57@1.43GHz , 128-core Maxwell	1	20000
12	Teensy 4.1 Controllers	Market	ARM cortex-M7 @ 600MHz,1024k Ram,55 total IO ports	2	5000
13	Power Distribution Board	Market and Customised	Power consumption rate	1	5000
14	24v LI ion Battery	Market	Voltage, current	2	10000
15	Telemetry Transceiver	market	433MHz	1	8000
16	Miscellaneous	-	-	-	10000

The operational efficiency of robots is mainly determined by efficiency of controlling software. To obtain that level of efficiency a strong firmware should be utilized. A firmware is used to run user programs on the device and can be thought of as the software that enables hardware to run.

Parameters	ROS2(Robotic Operating System)	Rtos (Real Time Operating System)	Hybridization of ROS2 and Rtos

Objective	ROS2 is an operating system for robots and robot-like devices, with a focus on collecting sensor input and separating that from motor control, allowing a simulation environment and a good UI for overseeing it all.	Rtos is a class of operating systems with well-defined timing characteristics for controlling physical objects where timing is critical. a RTOS based program can let the processor multitask while still ensuring that an interrupt handler (which might represent a limit switch on a moving piece of machinery) will get picked up and dealt with in a predictable number of microseconds.	The objectives are to incorporate the ROS 2 and RTOS strengths. It aims to leverage ROS 2 for high level communication and coordination, while using an RTOS for real time control tasks.
Communication Model	Adheres to the publish-subscribe communication paradigm. It enables the exchange of messages between different nodes in a distributed system.	Offers low-level real-time communication techniques, like shared memory and message passing. Many times, communication is straightforward and predictable.	Integrates the RTOS's real-time communication methods for time-sensitive tasks with the ROS 2 publish-subscribe model for communication between non-real-time components.
Flexibility	Its modular and flexible design makes it appropriate for a variety of robotic applications. It may operate on multiple platforms and supports a variety of programming.	Customized for real-time use and could not have the same level of modularity and flexibility as general-purpose operating systems. Predictability is given precedence over features with multiple uses.	Provides flexibility by enabling the use of an RTOS for operations requiring exact and predictable timing, and ROS 2 for tasks that don't require hard real-time guarantees.
Real-Time Capabilities	Although ROS 2 has some real-time control features, it is not a real-time operating system per se. It might not	Has features built in to ensure that tasks are completed in a timely and predictable manner. Applications requiring exact control and timing are a good fit for it.	Seeks to provide real-time performance by utilizing the RTOS's capabilities for important operations and ROS 2's sophisticated communication features.

	adhere to rigorous real-time specifications.		

The optimal approach to enhance the speed of the software system involves the integration of a real-time operating system with a non-real-time counterpart, forming a hybrid system. This hybrid configuration is designed to concurrently manage both real-time and non-real-time tasks by automatically segregating their corresponding interrupts.

Simulation And Real-Time Visualisation Tools:

Simulators are software's or platforms that simulate the behaviour of a robotic system in a virtual environment. The simulators must be able to perform the following task:

1. **Simulating Robot Dynamics:**

Robotics' movements, interactions with their surroundings, and reactions to outside factors can all be simulated.

2. **Simulated Sensors:**

Simulate the actions of sensors like IMUs, lidar, radar, and cameras. To test perception algorithms and sensor fusion methodologies, simulate sensor data.

3. **Modelling the Environment:**

Construct and simulate virtual worlds that are suitable for robot operation. This covers the depiction of barriers, topography, and other pertinent environmental elements.

4. **Practical Physics:**

To simulate the effects of forces, gravity, friction, and other physical phenomena on robot behaviour, provide a realistic physics simulation.

5. **Testing Control Algorithms:**

Make it easier for control algorithms to be tested and validated so that developers may assess how well various control strategies work in a controlled setting.

6. **Virtual Multi-Robot Systems:**

Encourage the multiple robot simulation.

Parameters	**Gazebo**	**Rviz**	**Rqt**
Purpose	A 3D simulation environment for robots. It allows users to simulate and visualize the dynamics and interactions of robots in	A 3D visualization tool for robotic systems. It provides a real-time, 3D views of robot sensor data, robot models, and other relevant	A set of plugins and tools for GUI development in ROS. It provides a graphical interface for monitoring and interacting with

	a virtual environment.	information.	ROS nodes and topics
Visualization Capabilities	3D simulation with realistic physics and environmental interactions. Allows the visualization of robot models and sensor data in a simulated environment.	3D visualization of robot models, sensor data, and other relevant information. Supports various visualization plugins for different types of data.	Provides a framework for creating custom plugins, including visualizers for different types of data. Allows the creation of custom graphical interfaces.
Simulation vs. Real-time Visualization	**Focus:** Simulation of robots and their interactions in a 3D environment. **Real-Time:** Real-time simulation with physics.	**Focus:** Real-time visualization of robot data during runtime. **Real-Time:** Typically used for real-time debugging and monitoring.	**Focus:** Real-time monitoring and interaction with ROS nodes. **Real-Time:** Primarily used for real-time introspection and control.
Integration with ROS	Often used with ROS for simulation. Gazebo can be launched through ROS and integrated with ROS node	A core part of the ROS visualization system. Often launched as part of a ROS workspace and integrated with ROS nodes	A set of ROS tools for GUI development. RQt plugins are used to interact with and monitor ROS systems.

Gazebo, RViz, and RQt play complementary roles in the ROS ecosystem, each serving unique functions. Gazebo specializes in 3D simulation, RViz excels in real-time visualization and debugging, and RQt provides a modular GUI framework for ROS development. By combining these tools, developers can leverage their specific strengths across the entire lifecycle of robotic software development.

In this case it is best to use Gazebo for simulation during development and testing, RViz for real-time visualization during rover operation, and RQt for monitoring and interaction with ROS nodes.

Test Plan

System Level	Testing
1. Navigation and Obstacle Avoidance	-Create different Obstacle courses of varying complexity -Measure Navigation accuracy
2. Remote Control Operation	-Use remote control to drive the rover through predefined routes. -Examine different control methods (manual, semi-autonomous, autonomous) -Evaluate Time and Accuracy
3. Endurance and Reliability	-Run the rover continuously for a long period -Monitor system Performance and health.

	-Simulate in various environmental conditions
4. Communication Range	-Increase the distance between the rover and the control system. -Observe the signal strength -Note down the operational range

Sub-System Level	**Testing**
1. Power Management System	-Simulate varying power conditions -Observe Power Consumption
2. Sensor Calibration	-Expose the rover to various environments -Validate sensor outputs against known outputs
3. Motor Control and Actuation	-Execute movement commands -Examine under various terrain conditions -Verify response time and accuracy
4. Communication Module	-Test signal strength, range, and reliability -Examine data integrity during transmission

Suspension Mechanism	**Testing**
1. Stability	- Terrain Simulation (rocky, sandy, inclined surfaces) - Obstacle Simulation
2. Load bearing	- Varied Weights - Stress Testing (Heavy Loads) - Monitor suspension behaviour
3. Durability	- Continuously operating the suspension system on simulated terrain for long periods
4. Traction and Performance	- Test on various surfaces like sand, rock and measure the wheel grip. - Observe and measure the speed on flat and inclined surfaces. - Collect all the data of speed, radius, wheel skidding.
5. Environmental	- Test on extreme cold and hot temperatures. - Move the rover on dusty and harsh environments to observe effects on suspension system
6. Functional	- Check functionality of Joints, pivots.

Rover Structure	**Testing**
1. Visual Inspection	-Check for any visible damages or cracks in the chassis, frame, or body. -Check for correct alignment of components. -Check whether the bolts and screws fitted correctly.
2. Dimensional Accuracy	-Use callipers to measure dimensions specified in documentation. -Verify alignment of wheels, arm etc.,
3. Load Testing	-Apply different payload objects and test them in which condition is that working properly.
4. Vibration and Shock	-Test the rover by vibrations and shocks.
5. Terrain Simulation	-Drive the rover over simulated rough arena -Check whether any noises, vibrations are getting while traversing.
6. Environmental Exposure	-Expose the rover to different environmental conditions (humidity, temperature, dust, water) as per the given guidelines
7. Impact Testing	-Do the tests using specified forces or throwing the rover from different heights.
8. Documentation and reporting	-Document detailed test procedures, observations, findings, and any other issues discovered during testing

Test plan for Motors:

DC Geared Motors	Testing
1. Voltage and Current	-Apply different voltages within the motor range, measure current and record performance of speed and torque.
2. No-Load	-Run the motor with no load in it, measure speed.
3. Load	-Apply different loads on it and measure the speed, torque and current.
4. Efficiency	-Calculate input and output power in specified loads then calculate its efficiency
5. Temperature	-Run motor continuously at different conditions measure and monitor the temperature whether it stays safe
6. Stall	-Apply heavy load until the motor stops rotating, measure the stall torque.

Servo Motors	Testing
1. Positioning Accuracy	-Test the servo in various positions, measure and monitor the accuracy of the position using in-built encoders.
2. Speed and Torque response	- Measure the motor's response time to reach the predefined values.
3. Overload	-Apply heavy load capacity to the motor and observe the accuracy and response of motor.
4. Temperature and thermal stability	-Run the motor continuously, monitor and measure the temperature at various points.
5. Feedback System	-Provide some random errors, observe the motor's response and accuracy, and monitor the feedback by provided by encoders.

Test Plan for Controllers:

Jetson Nano	Testing
1. Functional Testing	-Peripheral Connectivity: Test USB, HDMI, Ethernet, and other peripherals. -GPU: Test GPU functionality
2. Performance Testing	-Heat and Thermal Testing: Monitor heat dissipation.
3. Compatibility Testing	-Operating System Compatibility: Test compatibility with different operating systems and software libraries. -Peripheral Compatibility: Test compatibility with various peripherals, sensors, and devices
4. Power and Energy Testing	-Power Consumption: Measure power consumption under different load conditions. -Energy Efficiency: Check energy efficiency for optimized conditions.

Arduino Uno	Testing
1. Functional Testing	-Board Initialization: Ensure the Arduino Uno initializes properly. -Digital and Analog I/O: Test digital and analog pins for input and output functions
2. Compatibility Testing	-IDE Compatibility: Test compatibility with Arduino IDE and programming environment. -Peripheral Compatibility: Verify compatibility with sensors, modules etc

3. Performance Testing	-Blink Test: Test basic LED blink functionality. -Analog-to-Digital Conversion: Test accuracy of analog inputs using varying voltages
4. Communication Testing	-Serial Communication: Test serial communication between the Arduino Uno and a computer. -I2C and SPI Communication: Verify communications with I2C SPI, can, uart devices.

Battery Management System	**Testing**
1. Performance Testing	1.Capacity and Energy Testing: -Use a charge/discharge analyser to measure the battery capacity and energy output. 2. Voltage and Current Measurements: -Use a mustimeter to continuously monitor voltage and current during charge and discharge cycles. -Plot a graph for charge/discharge to analyse the battery's behaviour.
2. Safety and Abuse Testing	1.Thermal Runaway: -Conduct controlled heating tests using environment chamber or heat source, do observe the battery's response. 2.Over charge and Over discharge: -Apply excessive charging or discharging to test the battery's safety limits
3. Environmental Testing	1. Temperature Sensitivity: -Monitor performance of batteries under varying temperatures. 2.Vibration and Mechanical Stress: - Apply controlled vibrations on batteries so that we can check the assess the battery's durability.

System Specifications

S.NO	**System**	**Specification**
1	Rover Type	Wheeled
2	Rover Dimension (Alone)	Length: < 1 m, Breadth: 1 m, Height: 0.8 m
3	Rover Mass (with Manipulator Arm)	< 50 kg
4	Slope Climbing Capability	Minimum 15˚
5	Obstacle Climbing Capability	Minimum 150 mm
6	Power Source	Battery operated only
7	Communication	RF radiation mode only
8	Soil Type for Mobility	M sand
9	Gravity	Under Earth's Gravity
10	Manipulator Type	To be specified by the team
11	Manipulator Payload Mass	200 gm
12	Navigation Systems	On-board sensors and programming only
13	Battery Capacity	Enough for continuous navigation tasks
14	Use of Robotics Kits	Prohibited
15	Kill Switch	Required on rover's exterior
16	Rover Construction	own rover
17	Recommended Components	COTS components
18	Rover Speed	Minimum: 1 cm/s, Maximum: ≤ 50 cm/s
19	Real-time Control	Via radio link with legally available frequencies

20	Maximum Distance to Antenna Mast	< 25 m
21	Manipulator Arm Type	Universal robotic arm
22	Sample Tube Handling	Cylinder tube
23	Rover Stability	Should be stable during operation using rocker bogie suspension

Project management

S.No.	Task	Main Responsibility	Deadline for Completion	Estimated Cost	Secondary Responsibility
1	Project Initiation	Prashanth	15-01-2024	5000	Vinay
2	Requirements Analysis	Vinay Reddy	15-01-2024	5000	Prashanth Devarahatti
3	Hardware Procurement	John	30-01-2024	1,60,000	Prashanth Devarahatti
4	Software Development	Adi	20-3-2024	0	Bhavya
5	Sensor Integration	Vishnu	15-3-2024	0	Prashanth Devarahatti
6	Rover Assembly	Vinay Reddy	28-2-2024	20000	Prashanth Devarahatti
7	Simulation and Testing	Adi	15-2-2024	-open source-	Bhavya, Prashanth
8	Algorithm Development	Bhavya	30-1-2024	0	Adi, Prashanth Devarahatti

9	Telemetry Integration	Prashanth	20-3-2024	-open source-	John
10	System Integration	Prashanth	15-4-2024	0	John, Vishnu
11	Documentation	Hasri	15-01-2024	0	John, Vishnu
12	Presentation Preparation	Hasri	15-01-2024	0	Prashanth, Vinay

The project management strategy has been meticulously devised to ensure the successful execution of our rover development project. This involves a comprehensive breakdown of tasks, responsibilities, and strategic planning for schedule and budget management.

System Breakdown Structure

Key Responsibilities

1. *Project Initiation (Prashanth)
   - Initiated the project on 15-01-2024 with a budget allocation of 5000.
   - Collaborated with Vinay Reddy for a smooth project kick-off.

2. *Requirements Analysis (Vinay Reddy)
   - Conducted a detailed requirements analysis by 15-01-2024.
   - Coordinated closely with Prashanth Devarahatti for a comprehensive understanding of project needs.

3. *Hardware Procurement (John)
   - Oversaw the procurement of hardware components, budgeted at 160,000.
   - Collaborated with Prashanth Devarahatti to ensure hardware compatibility.

4. *Software Development (Adi)
   - Led the software development phase, leveraging in-house expertise.
   - Bhavya provided secondary support for seamless software integration.

5. *Sensor Integration (Vishnu)
   - Integrated sensors into the rover's system by 15-03-2024.
   - Prashanth Devarahatti provided additional support to optimize sensor functionality.

6. *Rover Assembly (Vinay Reddy)
   - Led the assembly process with a budget of 20,000.
   - Prashanth Devarahatti supported the assembly phase.

7. *Simulation and Testing (Adi)
   - Conducted simulation and testing activities by 15-02-2024.
   - Bhavya and Prashanth collaborated to ensure robust testing procedures.

8. *Algorithm Development (Bhavya)
   - Developed algorithms with in-house expertise by 30-01-2024.
   - Adi and Prashanth Devarahatti provided support for algorithm integration.

9. *Telemetry Integration (Prashanth)
   - Integrated telemetry systems using open-source tools by 20-03-2024.
   - John collaborated to ensure effective telemetry communication.

10. *System Integration (Prashanth)
    - Ensured seamless integration of all subsystems by 15-04-2024.
    - John and Vishnu provided secondary support for comprehensive system integration.

11. *Documentation (Hasri)
    - Documented project details by 15-01-2024.
    - John and Vishnu contributed to documentation efforts.

12. *Presentation Preparation (Hasri)
    - Prepared presentations for effective communication by 15-01-2024.
    - Prashanth and Vinay supported the preparation process.

Schedule Management

The schedule has been meticulously planned to meet project deadlines. Each task is aligned with its respective completion date, allowing for seamless progression through the project phases.

Cost Estimation

A detailed cost estimation has been provided for each task, taking into account budgetary considerations. Open-source tools and in-house expertise have been leveraged to minimize costs wherever possible.

Additional Technical Notes

Software Development: Leveraged in-house expertise to develop the required software, minimizing external costs.
Sensor Integration: Leveraged existing sensors where possible to reduce costs and streamline integration efforts.
Simulation and Testing: Used open-source simulation tools to simulate and test rover functionalities, reducing the need for expensive proprietary software.
Algorithm Development: Capitalized on in-house expertise for algorithm development, minimizing external costs.
Telemetry Integration: Utilized open-source telemetry tools to integrate communication systems, reducing direct costs.
System Integration: Conducted in-house integration efforts to ensure seamless collaboration among various subsystems.
Documentation: Implemented internal documentation processes to capture project details and progress.
Presentation Preparation: Conducted in-house preparation of presentations to effectively communicate project status and achievements.

This refined project management section provides a detailed insight into the strategic planning, responsibilities, and technical aspects of the project. It emphasizes the team's expertise and the efficient use of resources for successful project execution.

Novelty in the overall proposal

Our team believes that our approach is unique and created our own system Architecture for our bot, where our team comprises from all the departments of engineering having required skills to build robots for every indigenous subsystem required from scratch with passionately
We found Novelty in

1. System Architecture
2. Wheel Design
3. Autonomous motion
4. Pick and place mechanism
5. Rover Design
6. Power management
7. Emergency response System
8. Software part